Homework 5

Tailin Lo tl1720 N15116873

1 section 2.1.1

(1) prove huberized hinge loss is differentiable.
let $\mu = yt$, rewrite the equation,

$$l_{huber-hinge} = \begin{cases} 0 \text{ if } \mu > 1\text{+h} \\ \dfrac{(1+h-\mu)^2}{4h} \text{ if } |1-\mu| \leq \text{h} \\ 1-\mu \text{ if } \mu < \text{1-h} \end{cases} \tag{1}$$

Differentiate each segment, we can get,

$$\frac{dl_{huber-hinge}}{d\mu} = \begin{cases} 0 \text{ if } \mu > 1\text{+h} \\ \dfrac{-(1+h-\mu)}{2h} \text{ if } |1-\mu| \leq \text{h} \\ -1 \text{ if } \mu < \text{1-h} \end{cases} \tag{2}$$

Now, consider the margin
(i) $\mu = 1-h$
the left of the derivative is -1
the right of the derivative is $\frac{-(1+h-(1-h))}{2h} = -1$
(ii) $\mu = 1+h$
the left of the derivative is $\frac{-(1+h-(1+h))}{2h} = 0$
the right of the derivative is 0
Thus, the derivative is continuous in R
Proof.

(2) write the analytic expression of the gradient of the huberized hinge loss

$$\frac{dl_{huber-hinge}}{d\mu} = \begin{cases} 0 \text{ if } \mu > 1\text{+h} \\ \dfrac{-(1+h-\mu)}{2h} \text{ if } |1-\mu| \leq \text{h} \\ -1 \text{ if } \mu < \text{1-h} \end{cases} \tag{3}$$

(3) give an upper bound of Lipschitz-continuous From the above, it's easy to observe the largest derivative is 0. From the definition of an upper bound of Lipschitz-continuous, $|f(x)-f(y)| \leq c|x-y| => \frac{|f(x)-f(y)|}{|x-y|} \leq c$, we can say that the upper bound of Lipschitz-continuous is actually find the upper bound of the derivative. And because the largest derivative is 0, the upper bound of Lipschitz-continuous is 0. Thus, the gradient of huberized hinge loss is continue over R.

2 section 2.1.2

$F(\omega) = \omega^T \cdot \omega + \frac{C}{n}\mathbf{1} \cdot \mathbf{L}$ where $\mathbf{1}, \mathbf{L} \in \mathbf{R_{n \times 1}}$, L is
$[L_1, L_2, ..., L_n] = [l_{huber-hinge}(y_1, \omega^T x_1), ..., l_{huber-hinge}(y_n, \omega^T x_n)]$.
$\nabla_\omega F = (\frac{dF}{d\omega_1}, \frac{dF}{d\omega_2}, ..., \frac{dF}{d\omega_{d+1}})$
$= (2\omega_1 + \frac{C}{n}\mathbf{1} \cdot \nabla_{\omega_1}\mathbf{L}, 2\omega_2 + \frac{C}{n}\mathbf{1} \cdot \nabla_{\omega_2}\mathbf{L}, ..., 2\omega_{d+1} + \frac{C}{n}\mathbf{1} \cdot \nabla_{\omega_{d+1}}\mathbf{L})$
$= 2\omega + \frac{C}{n}\mathbf{Q} \cdot \mathbf{1}$, where $\mathbf{Q} =$

$$\begin{bmatrix} \nabla_{\omega_1}L_1 & \nabla_{\omega_1}L_2 & \nabla_{\omega_1}L_3 & \dots & \nabla_{\omega_1}L_n \\ \nabla_{\omega_2}L_1 & \nabla_{\omega_2}L_2 & \nabla_{\omega_2}L_3 & \dots & \nabla_{\omega_2}L_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nabla_{\omega_{d+1}}L_1 & \nabla_{\omega_{d+1}}L_2 & \nabla_{\omega_{d+1}}L_3 & \dots & \nabla_{\omega_{d+1}}L_n \end{bmatrix}$$

=

$$\begin{bmatrix} \nabla_\omega L_1 & \nabla_\omega L_2 & \nabla_\omega L_3 & \dots & \nabla_\omega L_n \end{bmatrix}$$

$$\nabla_\omega L_i(\mathbf{x}, y) = \begin{cases} \mathbf{0} \text{ , if } y(\omega^T \cdot \mathbf{x}) > \text{1+h} \\ \dfrac{(1+h-y(\omega^T \cdot \mathbf{x}))}{2h} \times (-y\mathbf{x}) \text{ , if } |1-y(\omega^T \cdot \mathbf{x})| \leq \text{h} \\ -y\mathbf{x} \text{ , if } y(\omega^T \cdot \mathbf{x}) < \text{1-h} \end{cases} \tag{4}$$

3 section 3

In this section, we have two parts needed to be implemented, i.e., one is mini-batch kmeans, and the other is VLAD representation.
In MiniBatchKMeans.py, there are two simple tests for this algorithm. The first test is check whether the algorithm can have precise centroids. The second test is to sweep the minimum batch size versus objection value. This test can decide what the mini-batch is.
In the folder of project, the file mainly is used as the test of VALD. According to the paper, we can implement a simple VLAD extraction and test function. The prediction still used the KNN algorithm to compare with the result of the last assignment. The following is the result,