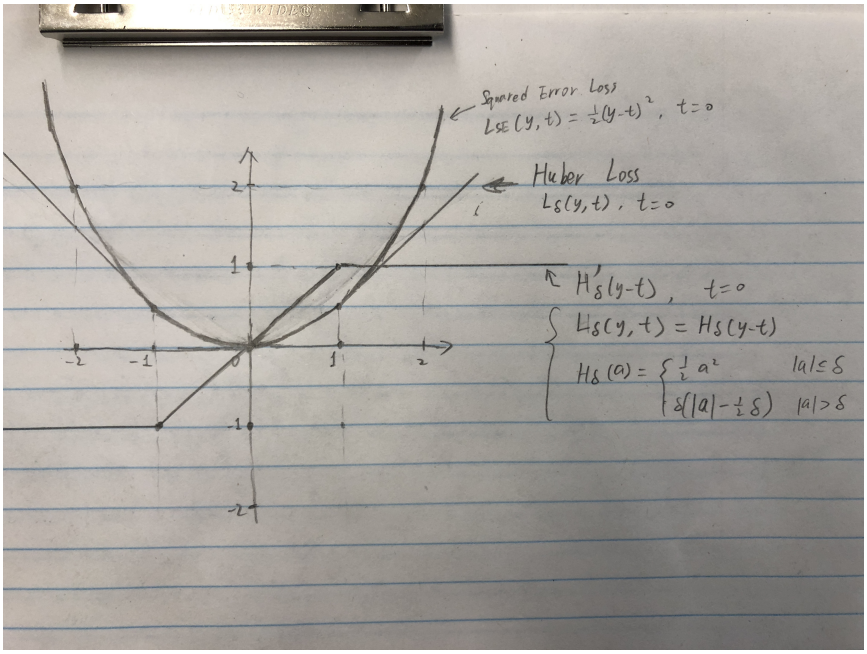
CSC411 A3

Zhihong Wang 1002095207

2018-10-08

Q1:

(a).
Sketched: Select $\delta = 1$



Why we expect the Huber loss to be more robust to outliers:

Huber Loss $L_\delta(y,t) = H_\delta(y-t) = \begin{cases} \frac{1}{2}(y-t)^2, |y-t| \le \delta \\ \delta(|y-t| - \frac{1}{2}\delta), |y-t| > \delta \end{cases}$

We know δ is the hyper parameter of Huber loss, y is the real/true value, and t is the predicted value based on the model. By checking the sketched from above, we can see the horizontal axis is the difference between the true value and the predicted value (i.e. $y - t$), and the vertical axis is the loss value.
For Huber loss, if the prediction is less than or equal to δ, we use squared error; if the prediction is greater than δ, we use linear error (i.e. absolute error).

Based on the graph, we know the squared error loss is very sensitive to outliers. The squared error loss is easy to over value the difference between the true value and the predicted value (i.e. it punishes outliers hardly, not robust enough). However, the Huber loss is able to control the loss value due to outliers (i.e. it punishes outliers lighter, less sensitive, more robust than squared error loss). Actually, Huber loss is the combination of squared error loss and absolute error loss, which improves the drawback (i.e. not robust) of squared error loss.

(b).
Note: $\delta > 0$

$$H_\delta(a) = \begin{cases} \frac{1}{2}a^2, |a| \leq \delta \\ \delta(|a| - \frac{1}{2}\delta), |a| > \delta \end{cases}$$

$$H_\delta^{'}(a) = \begin{cases} a, |a| \leq \delta \\ \delta, |a| > \delta, a > 0 \\ -\delta, |a| > \delta, a < 0 \end{cases}$$

$y = w^\top x + b$

$\frac{\partial L_\delta}{\partial w} = \frac{\partial L_\delta(y,t)}{\partial w} = \frac{\partial L_\delta(w^\top x+b,t)}{\partial w} = \frac{\partial H_\delta(w^\top x+b-t)}{\partial w} = \frac{\partial H_\delta(w^\top x+b-t)}{\partial (w^\top x+b-t)} \times \frac{\partial (w^\top x+b-t)}{\partial w}$
$= \frac{\partial H_\delta(w^\top x+b-t)}{\partial (w^\top x+b-t)} \times x = H_\delta^{'}(y-t) \times x$

$\frac{\partial L_\delta}{\partial b} = \frac{\partial L_\delta(y,t)}{\partial b} = \frac{\partial L_\delta(w^\top x+b,t)}{\partial b} = \frac{\partial H_\delta(w^\top x+b-t)}{\partial b} = \frac{\partial H_\delta(w^\top x+b-t)}{\partial (w^\top x+b-t)} \times \frac{\partial (w^\top x+b-t)}{\partial b}$
$= \frac{\partial H_\delta(w^\top x+b-t)}{\partial (w^\top x+b-t)} \times 1 = H_\delta^{'}(y-t)$

$$H_\delta^{'}(y-t) = \begin{cases} y-t, |y-t| \leq \delta \\ \delta, |y-t| > \delta, y-t > 0 \\ -\delta, |y-t| > \delta, y-t < 0 \end{cases}$$

Q2:

(a).

Let vector $r = y - Xw$
then $\langle r, Ar \rangle = r^T Ar = \sum_j r_j a^{(j)} r_j = \sum_{j=1}^{N} r_j^2 a^{(j)}$

$L(w) = \frac{1}{2}(y - Xw)^T A(y - Xw) + \frac{\lambda}{2}\|w\|^2$
$= \frac{1}{2}(A^{\frac{1}{2}}(y - Xw))^2 + \frac{\lambda}{2}\|w\|^2$
$= \frac{1}{2}\|A^{\frac{1}{2}}y - A^{\frac{1}{2}}Xw\|^2 + \frac{\lambda}{2}\|w\|^2$
$= \frac{1}{2}(A^{\frac{1}{2}}y - A^{\frac{1}{2}}Xw)^T(A^{\frac{1}{2}}y - A^{\frac{1}{2}}Xw) + \frac{\lambda}{2}\|w\|^2$
$= \frac{1}{2}(y^T A^{\frac{1}{2}} - w^T X^T A^{\frac{1}{2}})(A^{\frac{1}{2}}y - A^{\frac{1}{2}}Xw) + \frac{\lambda}{2}\|w\|^2$
$= \frac{1}{2}(y^T Ay + w^T X^T AXw - y^T AXw - w^T X^T Ay) + \frac{\lambda}{2}\|w\|^2$
$= \frac{1}{2}(y^T Ay + w^T X^T AXw - 2w^T X^T Ay + \lambda w^T w)$

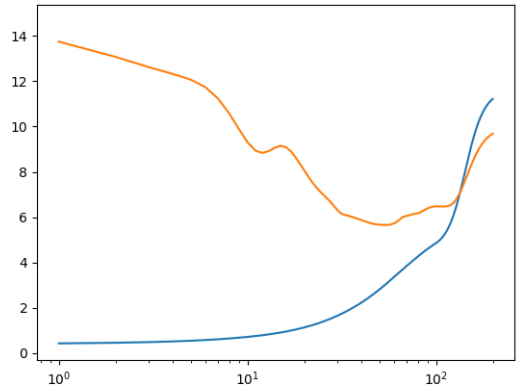
$\nabla L(w) = \frac{1}{2}\nabla_w(y^T Ay + w^T X^T AXw - y^T AXw - w^T X^T Ay + \lambda\|w\|^2)$
$= \frac{1}{2}\nabla_w(y^T AXw + w^T X^T AXw - w^T X^T Ay) + \lambda Iw$
$= -\frac{1}{2}\nabla_w tr(y^T AXw) + \frac{1}{2}\nabla_w tr(w^T X^T AXw) - \frac{1}{2}\nabla_w tr(w^T X^T Ay) + \lambda Iw$
$= -\frac{1}{2}X^T Ay + X^T AXw - \frac{1}{2}X^T Ay + \lambda Iw$
$= -X^T Ay + X^T AXw + \lambda Iw$
$= -X^T Ay + X^T AXw + \lambda w$

Then $-X^T Ay + X^T AXw^* + \lambda w^* = 0$
So, $(X^T AX + \lambda I)w^* = X^T Ay$

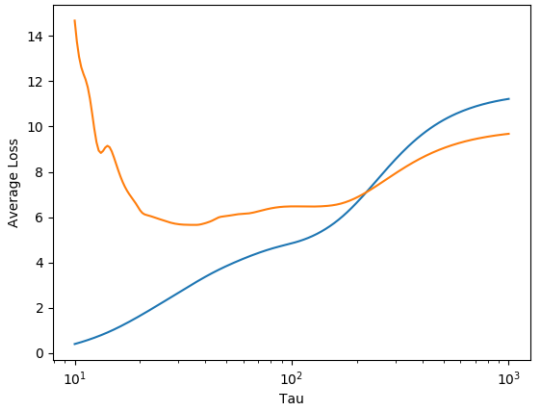
Therefore, $w^* = (X^T AX + \lambda I)^{-1} X^T Ay$

(c).
Blue Line: Training, Orange Line: Validation
The random seed is 0.
idx is a global random index array with range N given by the starter code.
We can also generate local *idx* to overwrite the global one for different outputs.

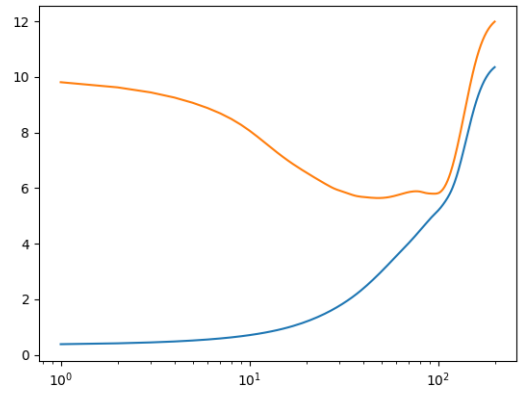
Graph without modifying the *plt* part from the starter code (local idx):



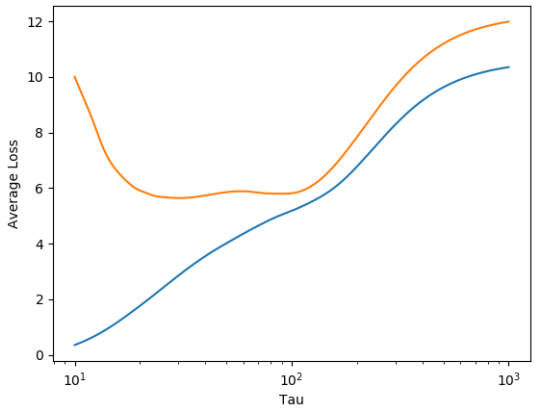
Graph with added labels and correct range (local idx):



Graph without modifying the *plt* part from the starter code (global idx):



Graph with added labels and correct range (global idx):



(d). How would you expect this algorithm to behave as $\tau \to \infty$? When $\tau \to 0$? Is this what actually happened?
It was expected to look like the blue training line (i.e. average training loss): when $\tau \to \infty$, the loss $\to$ a fixed number (This number depends on the random data splitting. If there is no fixed random seed, the results will be different every time), and when $\tau \to 0$, the loss $\to 0$.
However, the orange validation line (i.e. average validation loss) is different from the training. When $\tau \to \infty$, the loss $\to$ a specific number. Also, when $\tau \to 0$, the loss $\to$ a fixed number too.