IN4320: Machine Learning Assignment 1

Zheng Liu (4798406)

February 23, 2019

1 Question 1

Proof:
let $=(\mathrm{x}_1, x_2, ..., x_n)^T,\ \vec{b} = (y_1, y_2, ..., y_n)^T,\ (x_i, y_i \in R,\ \vec{a},\ \vec{b} \in R^d))$
$(|x_i + y_i|)^2 = |x_i^2 + y_i^2 + 2x_iy_i| = x_i + y_i + 2x_iy_i$
$\leq x_i^2 + y_i^2 + 2|x_iy_i| = (|x_i| + |y_i|)^2,$
$|x_i + y_i| \leq |x_i| + |y_i|,$
$so,\ \sum_{i=1}^{n} |x_i + y_i| \leq \sum_{i=1}^{n}(|x_i| + |y_i|) = \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |y_i|$
$so,\ ||\vec{a} + \vec{b}||_1 \leq ||\vec{a}||_1 + ||\vec{b}||_1,$
$let\ f(\vec{x}) = ||\vec{x}||_1,\ f(c\vec{x}) = ||c\vec{x}||_1 = |c|||\vec{x}||_1,$
$especially,\ c \in [0, 1]\ (c\ is\ weight),\ f(c\vec{x}) = c||\vec{x}||_1 = cf(\vec{x}),$
$f(c\vec{x} + (1 - c)\vec{y}) \leq f(c\vec{x}) + f((1 - c)\vec{y})$
$= cf(\vec{x}) + (1 - c)f(\vec{y}),\ it\ holds\ for\ \forall\ \vec{x}, \vec{y} \in R^d\ that\ norm\_1\ is\ convex,$
$so\ ||x_i - m_cy_i||_1\ and\ \lambda||m_+ - m_- + a||\ hold\ that\ they\ are\ convex,$
$then\ use\ the\ property\ that\ if\ f_1, f_2, ..., f_n\ are\ convex,\ then\ \sum_i f_i\ is\ convex,$

$$\sum_{c\in\{+,-\}} \sum_{i=1}^{N_c} ||x_i - m_cy_i||_1\ holds\ that\ it\ is\ convex,$$

$$so,\ L(m, m_+, a) = \sum_{c\in\{+,-\}} \sum_{i=1}^{N_c} ||x_i - m_cy_i||_1 + \lambda||m_+ - m_- + a||\ is\ convex.$$

End

2 Question 2

Based on the setting, the problem is transformed into solve the corner points of the following equation and the plot is shown in figure1.

$$|1 - m_-| + |3 - m_-| + |a - m_-| = \pi \tag{1}$$

For these three parts of absolute value, when they are equal to 0 respectively, the derivative does not exist, so they are the corner points respectively. So, for example, set $|1 - m_-| = 0$, $|3 - m_-| + |a - m_-| = \pi$ corresponds to two corner points, it is easy to calculate that they are

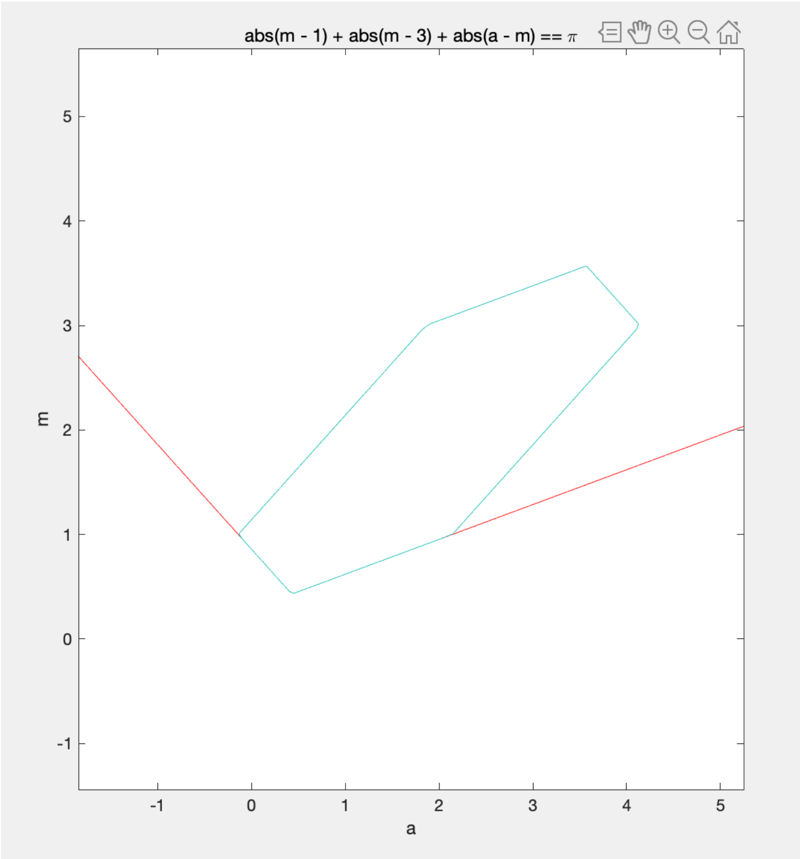

Figure 1: Plot of question 2

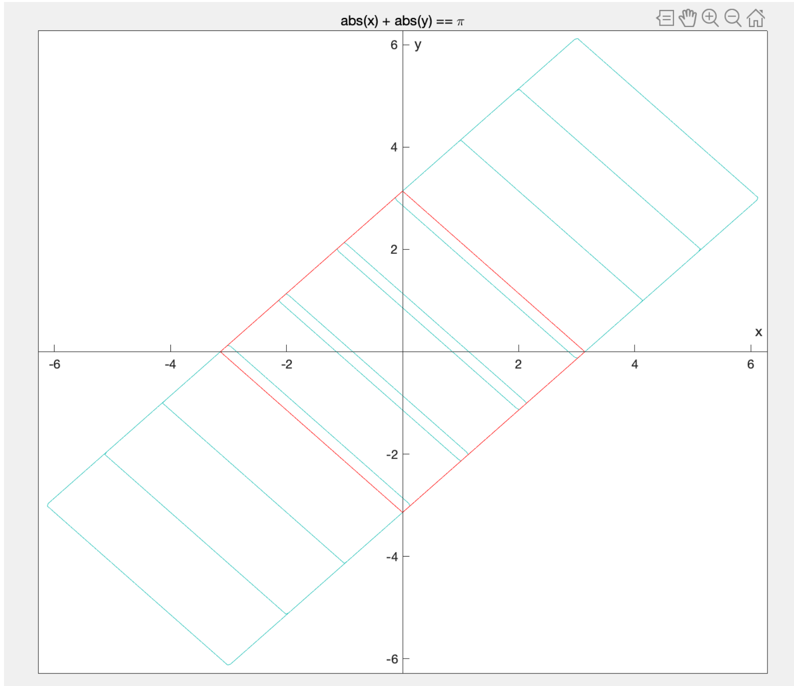

Figure 2: Set threshold value equal to π, only $a = 0$ it has sparsity solution

$(1,\pi-1)$ and $(1,\pi-1)$, also it is easy to calculate the rest four corner points are $(3,1+\pi)$, $(3,5-\pi)$, $(\frac{4+\pi}{2})$ and $(\frac{4-\pi}{2})$.

3 Question 3

This term is the regularisation for the loss function, it enforces and limits the parameter m_+ and m_- not to be too large that overfits on the training dataset.

It has sparsity solution if and only if a equals to 0. Take 2-dimensions as example, let $x = m_+^{(1)} - m_-^{(1)}$ and $y = m_+^{(2)} - m_-^{(2)}$, x and y are the representation of $m_+ - m_-$ in two dimensions respectively. Set threshold equals to π, the function is presented as $|x + a| + |y + a| = \pi$ shown in figure2 of function of x and y by changing value of a from a negative value to a positive value. The red square is when $a = 0$ the figure of $|x| + |y| = \pi$.

Assume m_+ and m_- are fixed, each dimension of these two vectors is also fixed. It is obviously that to minimise the regularisation term is to minimise the function $||m_+ - m_- + a||_1$, it equals to minimise $\sum_{i=1} |m_+^{(i)} - m_-^{(i)} + a|$.I took the dimensions of m_+ and m_- equal to a odd number, 3, as example, setting each dimension of $m_+ - m-$ a value and plot the result, there exists a single point that has minimum value and the related value of a is equal to the median of values of all the dimensions $-|m_+^{(i)} - m_-^{(i)}|$. Similarly, set the dimension to an even number, there does not exist a single minimum point but a minimum curve, it starts and end on two middle value of $-|m_+^{(i)} - m_-^{(i)}|$,

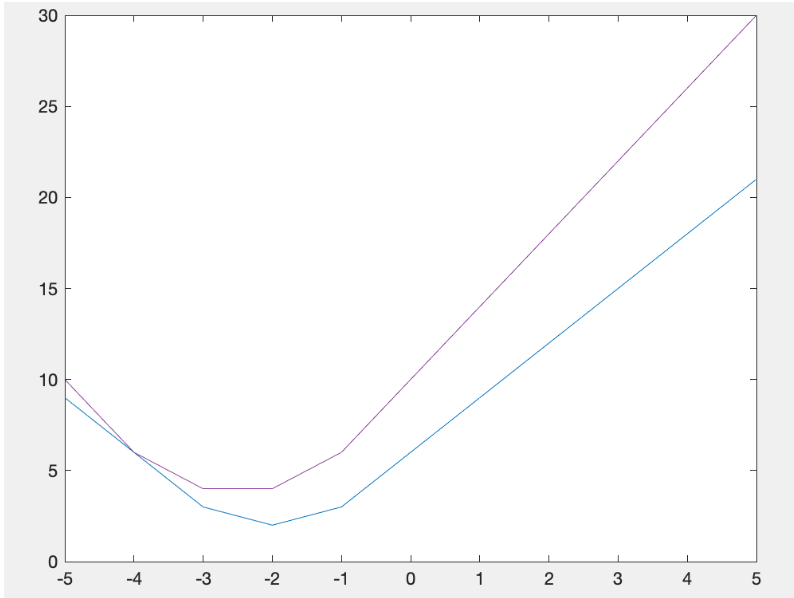

Figure 3: $Dimension = 3$, $m_+^{(i)} - m_-^{(i)}$ concludes {1,2,3}, the minimum value exist on -2, $Dimension = 4$, $m_+^{(i)} - m_-^{(i)}$ concludes {1,2,3,4}, the minimum value exist between -2 and -3

obviously, the median of all the dimensions is in the curve. The result is shown in figure3.

4 Question 4

I initialise the learning rate to a small value, 0.00001 and the turns of learning into 1000. Because the data scale is relatively small. I would rather each to move a small step each turn.

I used the gradient descent method to optimise the parameter of the loss function. I did manually program for gradient descent (not using the library function). The basic thing to do is to calculate the optimised m_+, m_- and a by taking the partial derivative and applying the gradient descent function. As there are the absolute functions, I consider it cannot be done directly to take the derivative, there are three situation for a single absolute function, the inner term is positive, negative and zero. So I used *if*-condition in my code to separate these three situations, after which, the derivative is easy to calculate and the rest of the function is easy to apply.

Then I apply the current parameters achieved from gradient descent function into the loss function and obtain the value. I did the loop of applying "loss function-gradient descent" until the loss value changes in a small range, I regard the steady statue reach the lowest value (as proved in question 1, there exists only one local minima which is also the global minima).

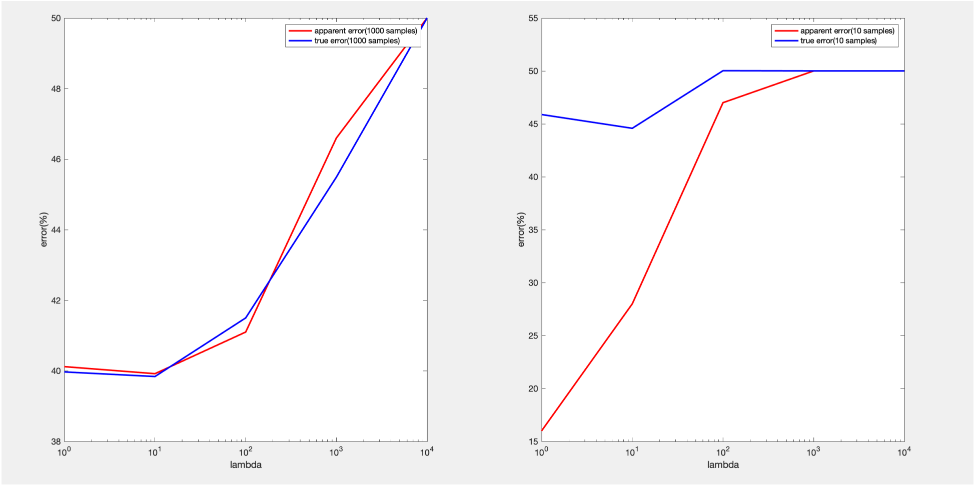

Figure 4: Error rate curves of λ based on different sample size

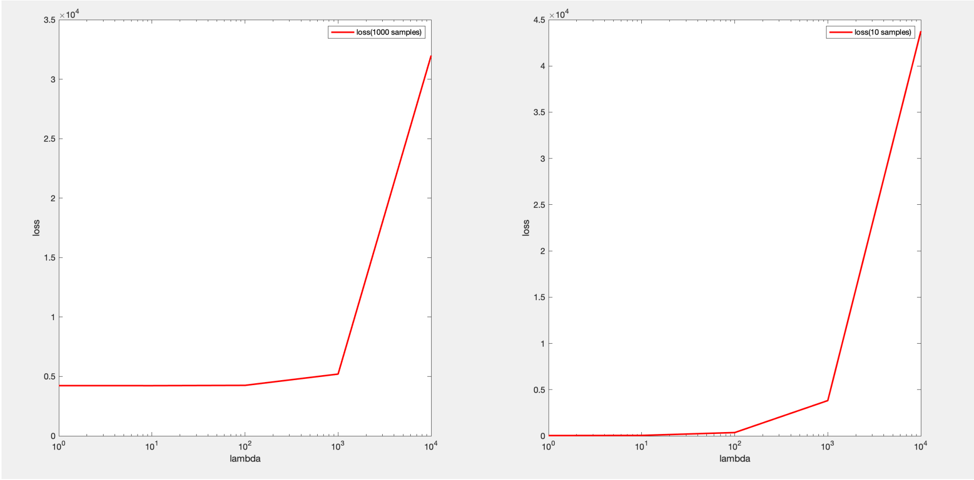

Figure 5: Two loss function curves of λ based on different sample size

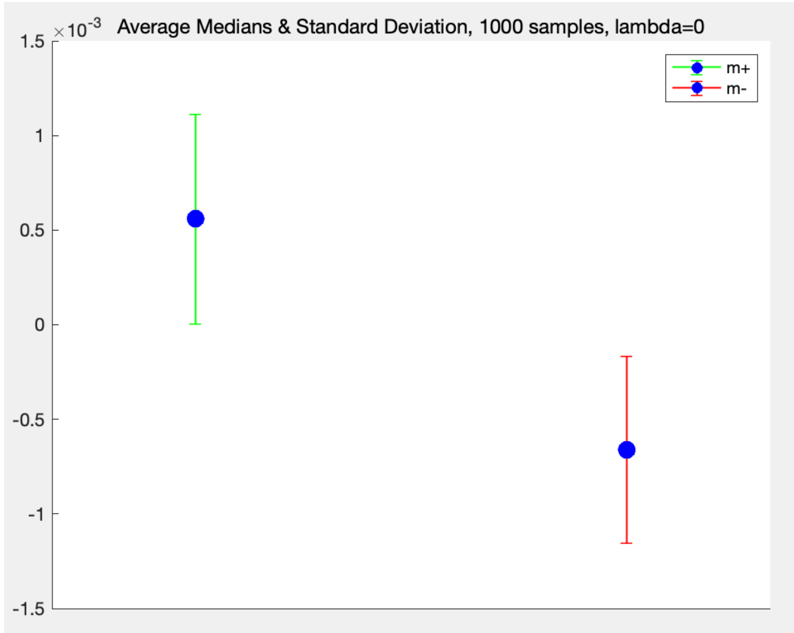

Figure 6

5 Question 5

The error rate curves, including apparent error and true error, for both when sample number is 1000 and 10 respectively are shown in figure4, with $\lambda \in \{0,1,10,100,1000,10000\}$.

The loss curves shown in figure5 illustrate the trend of loss when λ grows large. It is obvious that $\lambda = 10^3$ is a turning point, after which the loss grows immediately.

6 Question 6

Figure6, 7, 8 and 9 show the average median and standard deviation of m_+ and m_-.

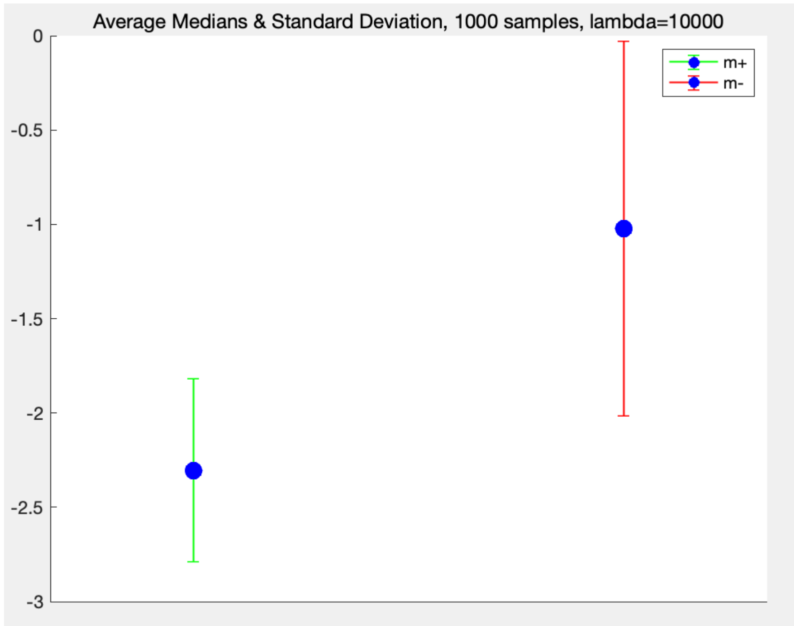

Figure 7

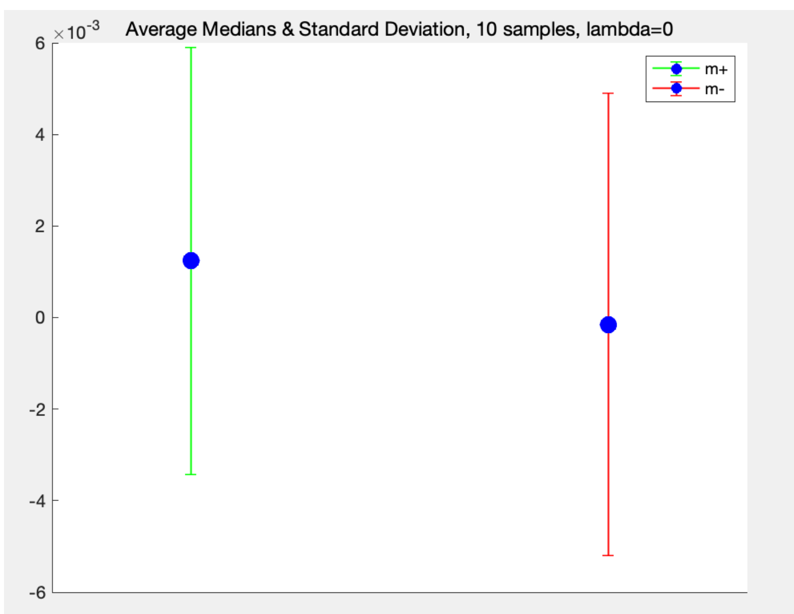

Figure 8

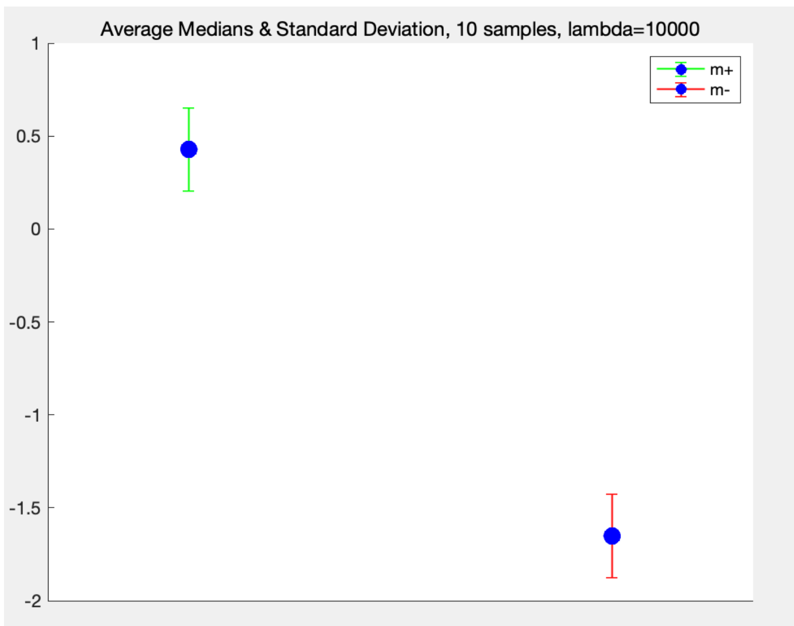
Average Medians & Standard Deviation, 10 samples, lambda=10000
m+
m-
1
0.5
0
-0.5
-1
-1.5
-2

Figure 9