CSE446 HW2

Runlin He

February 2024

A1

a

L1 norm function graph in 3-D is diamond shaped, in 2-D is square, so it has lots of corner. The weight at the corner is 0. While L2 norm function graph in 2-D is circle and in 3-D is a ball, and there is no corner on the ball, so the chance of weight = 0 is much lower.

b

Upside: This regularizer has corner in 2-D graph, so it may encourage sparsity (weight = 0)

Downside: This regularizer in 2-D graph is not convex, so we may not achieve the global minimum.

c

True, if the step-size is too large, it may cause overstep.

d

SGD require less computational resource relative to GD since we just need to take one random sample and do the gradient descent for the sample, while SGD is more sensitive to step size and noise relative to GD.

e

Linear regression have closed form solution, so we can apply the closed from solution directly. While Logistic regression doesn't have closed form solution, so we have to utilize gradient descent to achieve the global minimum.

A2

a

Fit on the non-negativity:

$$f(x) = \sum_{i=1}^{n} |x_i|$$
$$\because |x_i| \geq 0$$
$$\therefore \sum_{i=1}^{n} |x_i| \geq 0$$
$$\textbf{when } |x_i| = 0, \forall i \in \{1, 2, ...n\}$$
$$\sum_{i=1}^{n} |x_i| = 0$$

Fit on absolute scalability:

$$\begin{aligned} |a|f(x) &= |a| * \sum_{i=1}^{n} |x_i| \\ &= \sum_{i=1}^{n} |a||x_i| && \text{By multiplication} \\ &= \sum_{i=1}^{n} |ax_i| && \text{by absolute value multiplication} \\ &= f(ax) \end{aligned}$$

Fit on triangle inequality:

For three cases to constant a and b

i. a, b ≥ 0

$$\begin{aligned} |a| = a, |b| = b \\ |a+b| = a + b = |a| + |b| \end{aligned}$$

ii. a ≥ 0, b ≤ 0

$$\begin{aligned} |a| = a, |b| = -b \geq b \\ |a+b| \leq |a| + |b| \end{aligned}$$

a and b can be any constant so they can swap as well

iii. a ≤ 0, b ≤ 0

$$\begin{aligned} |a| = -a, |b| = -b \\ |a+b| = -a - b = |a| + |b| \end{aligned}$$

The above 3 cases have included all cases for constant a and b, so we get

$$\begin{aligned} |a+b| &\leq |a| + |b| \\ \therefore |x_1 + y_1| &\leq |x_1| + |y_1| \\ \because x_i, y_i &\in \mathbf{R} \\ |x_i + y_i| &\leq (|x_i| + |y_i|) \end{aligned}$$

So we could prove the f(x) = $\sum_{i=1}^{n} |x_i|$ is a norm

b

For $(1,2), (2,1)$ in 2-D,

$g((1,2)) = g((2,1)) = (1+\sqrt{2})^2 = 3 + 2\sqrt{2}$,

$g((1,2) + (2,1)) = g((3,3)) = (\sqrt{3} + \sqrt{3})^2 = 12$

$2 * (3 + 2\sqrt{2}) < 12$, we get proved g(x) is not a norm

A3

Part I is not convex, we could connect the point b to c, the line is out of scope to Part I. Not fit on $\lambda x + (1-\lambda)y \in A$

Part II is not convex, we could connect the point a to d, the line is out of scope to Part II. Not fit on $\lambda x + (1-\lambda)y \in A$

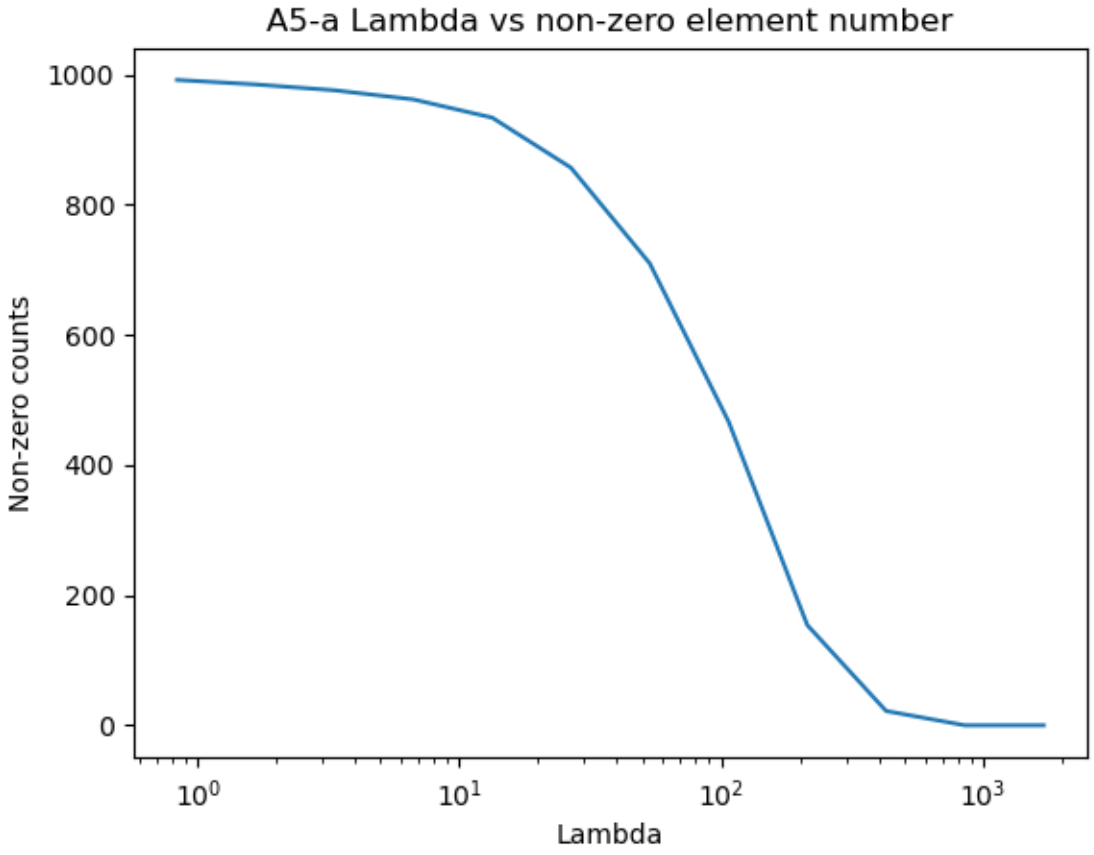
A4

Part I is convex, when we connect every point on the f(x), we found the connection line is always above the function, so we could utilize $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$ to determine it is convex.

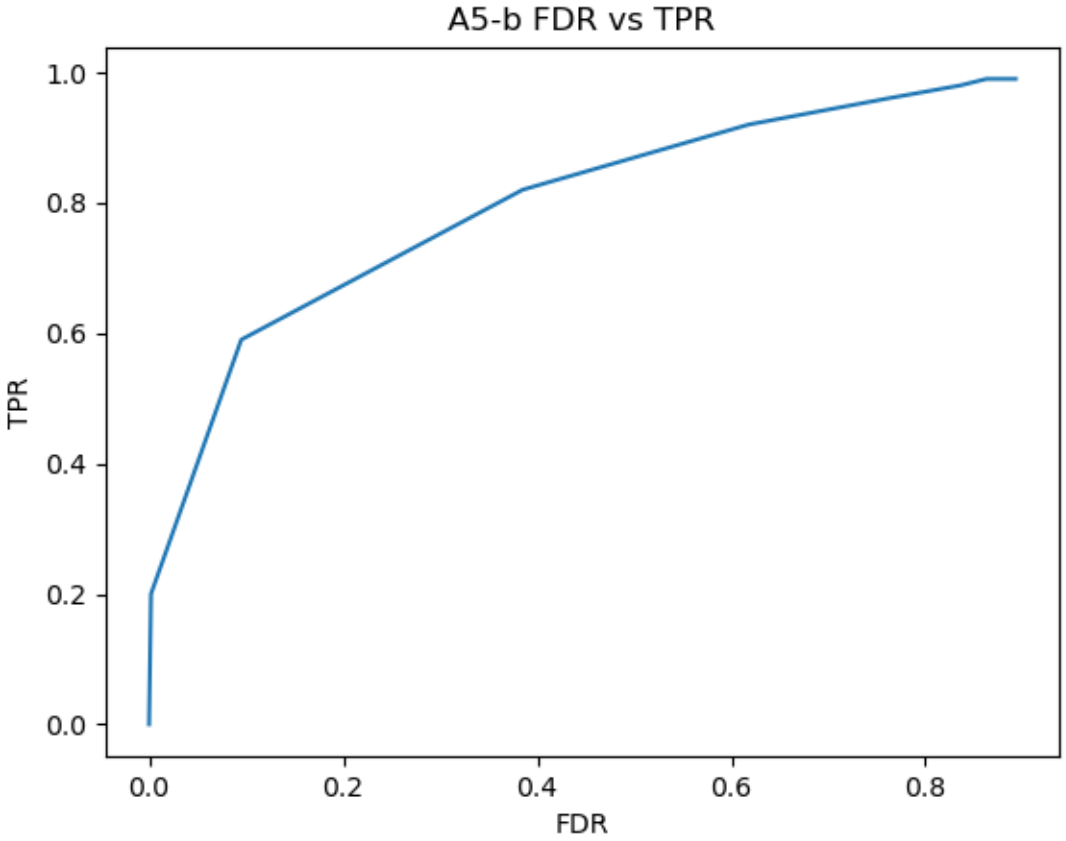
Part II is not convex, when we connect a to c, the connection line is below the function which means $f(\lambda a + (1-\lambda)c) > \lambda f(a) + (1-\lambda)f(c)$

A5

a

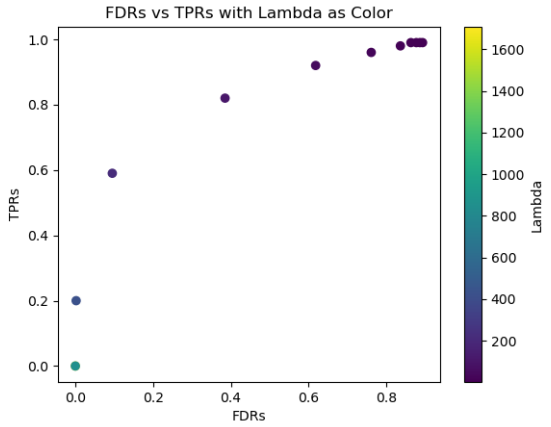


b



c

With increasing the lambda in plot a, we found the nonzero counts in the weight vector decrease to nearly 0, which means the lambda encourage the sparsity for our weight. For plot b, I want to give another plot to better explain the lambda effect



In this plot, we could found with the decrease of lambda, the FDR and TPR increase, which means lambda could encourage sparsity for the whole weight vector (which means encourage weight tend to be 0). With smaller lambda, we tend to have larger true nonzero weights and false nonzero weights.

A6

a

PersPerOccupHous: mean persons per household: Financial and housing policies, including interest rates and lending practices, impact the affordability of homes, thus affecting the mean persons per household

PolicBudgPerPop: police operating budget per population: police budgets are influenced by governing body, elections and available financial resources like tax revenues

NumStreet: number of homeless people counted in the street: vary due to historical policy choices in the U.S. such as housing affordability by financial policy, social services availability, and economic policies affecting employment.

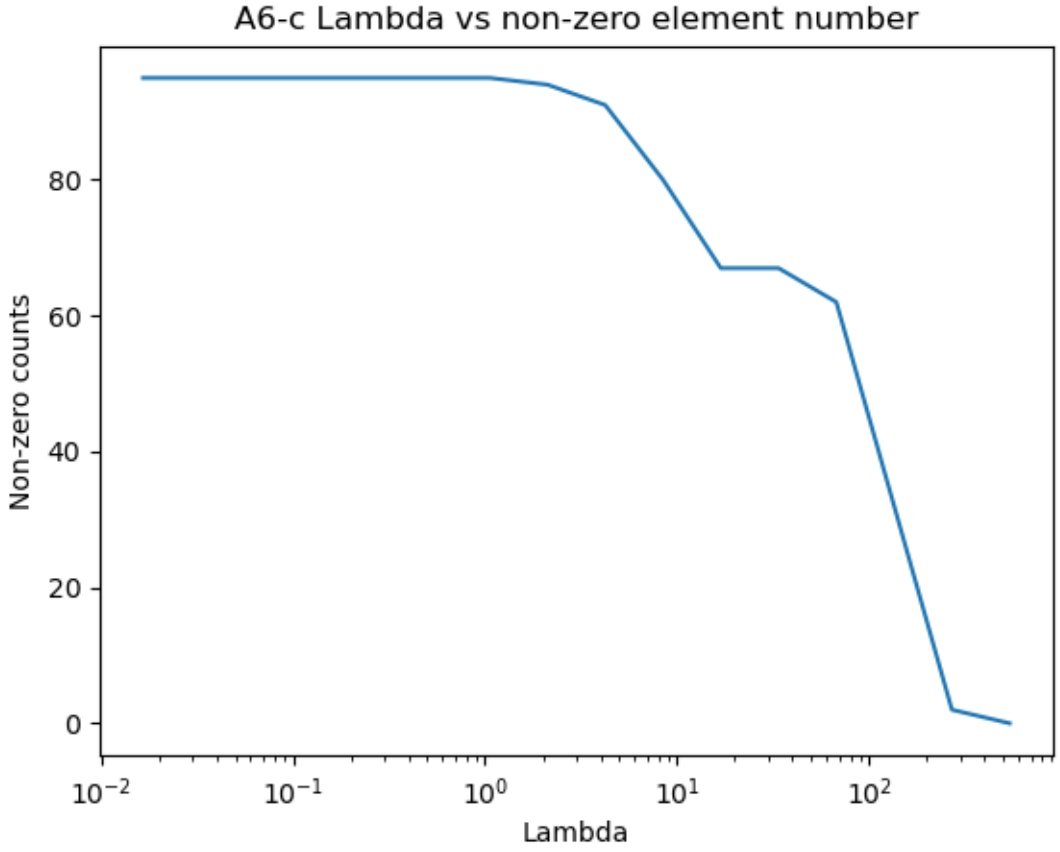
b

PctPopUnderPov: percentage of people under the poverty level may indicate the violent crime, while it is the result of crime since violent prisoner may have less probability to get a job with good salary.

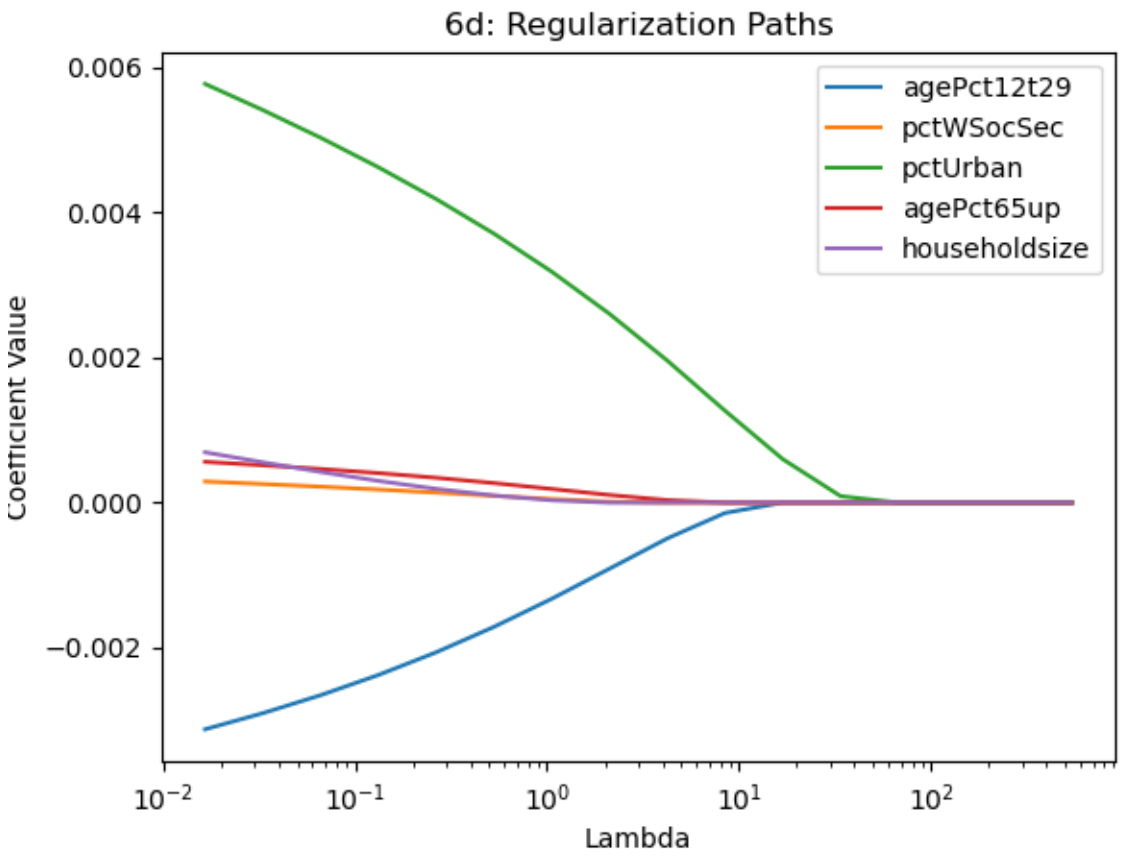
NumIlleg: NumIlleg seems to be the most relative parameter in the dataset. While this is the result of crime, since the crime increase, the number of illegal increase. The Illegal crime include the violent crime.

NumStreet: number of homeless people counted in the street may indicate the violent crime, while it is also the result of crime since violent prisoner may have less job opportunities and less salary so that they may not have home, so they have to stay at street.

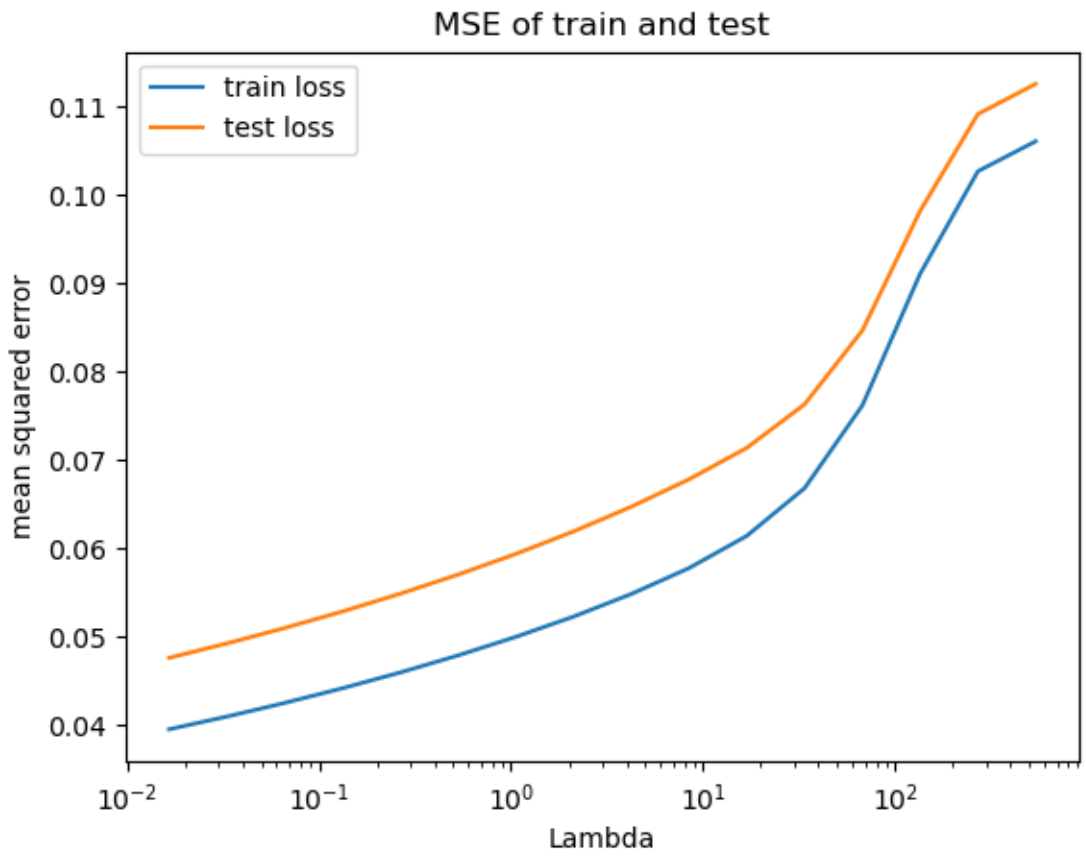
c



d


6d: Regularization Paths
agePct12t29
pctWSocSec
pctUrban
agePct65up
householdsize
Coefficient Value
0.006
0.004
0.002
0.000
−0.002
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Lambda

e



f

NumIlleg has the largest weight ≈ 0.0568. PctFam2Par has the smallest weight ≈ -0.0357.

NumIlleg has the most attribute to the voilent crime is due to the violent crime is the reason for NumIlleg to increase rather than the NumIlleg contribute the violent crime's rate.

Percentage of families (with kids) that are headed by two parents may reduce the rate of violent crime since the complete family could help cultivate a healthy mind for kids. The two parents family could also protect themselves from violent crime.

g

The flaw in this line of reasoning is reverse of cause and outcome.

The reason for agePct65up is high is the safer region will be more attractive to the old (since they may have less ability to protect them if they meet a violent crime) rather than the people older than 65 make the region safer.

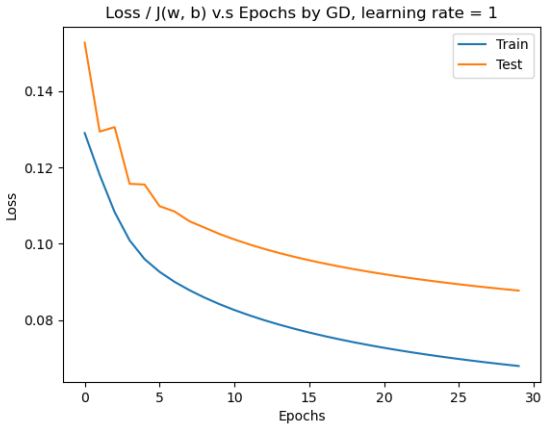
A7

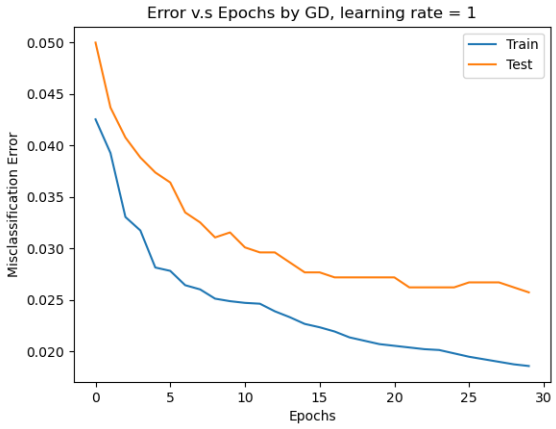
a

$$\begin{aligned}\nabla w_j &= \frac{1}{n}\sum_{i=1}^{n}(-y_i * x_i * e^{-y_i+(b+x_i^T w)} * \frac{1}{1+e^{-y_i(b+x_i^T w)}+2\lambda w}) \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{-y_i x_i * e^{-y_i(b+x_i^T w)}}{1+e^{-y_i(b+x_i^T w)}} + 2\lambda w \\ &= \frac{1}{n}\sum_{i=1}^{n} -y_i x_i(\mu_i^{-1}-1)\mu_i + 2\lambda w \\ &= \frac{1}{n}\sum_{i=1}^{n} -y_i x_i * (1-\mu_i(w,b)) + 2\lambda w\end{aligned}$$

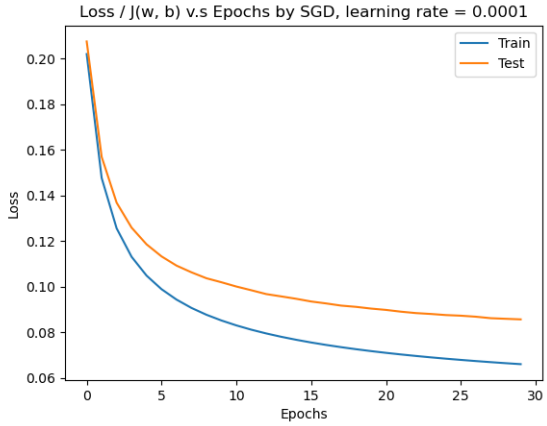
$$\begin{aligned}\nabla w_b &= \frac{1}{n}\frac{\sum_{i=1}^{n} -y_i * \exp(-y_i(b+(x_i^T w)))}{1+exp(-y_i(b+x_i T w))} \\ &= \frac{1}{n}\sum_{i=1}^{n} -y_i(\mu_i^{-1}-1)\mu_i \\ &= \frac{1}{n}\sum_{i=1}^{n} -y_i * (1-\mu_i(w,b))\end{aligned}$$

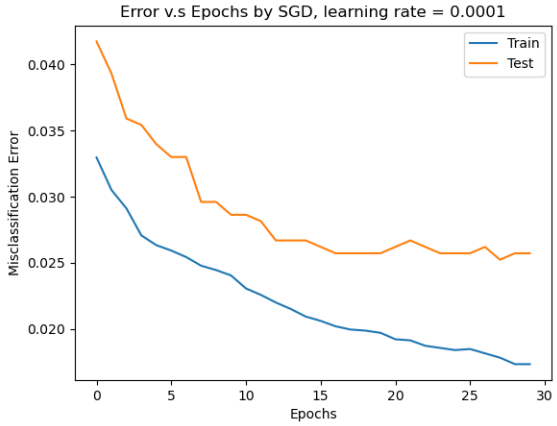
b



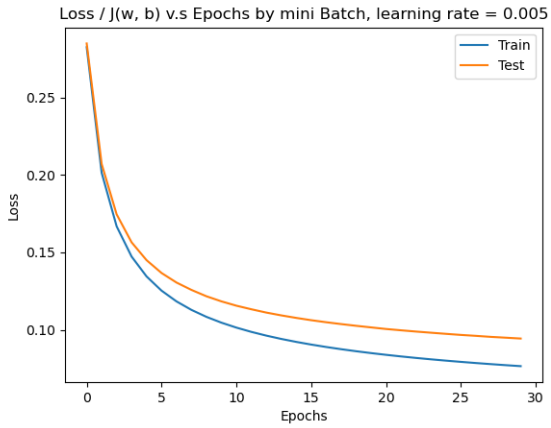


c





d



A8

I spend 30 hours on coding part for the VScode issue and vector to matrix operation.