CSE250B Homework 4

Qiao Zhang

February 12, 2016

1 regression problem

1.1

Given L(w), we can calculate the H(w)
$H_{jk} = \frac{\partial^2 L}{\partial w_k \partial w_j}$
$= \frac{\partial}{\partial w_k}(-2\sum_{i=1}^{n} x_j^{(i)}(y^{(i)} - w \cdot x^{(i)}))$
$= -2\sum_{i=1}^{n} x_j^{(i)} \frac{\partial}{\partial w_k}(y^{(i)} - w \cdot x^{(i)})$
$= 2\sum_{i=1}^{n} x_j^{(i)} x_k^{(i)}$
$= 2x_j \cdot x_k$
where $x_j = [x_j^{(1)}, x_j^{(2)}, ..., x_j^{(n)}]$, $x_k = [x_k^{(1)}, x_k^{(2)}, ..., x_k^{(n)}]$
We can apply matrix decomposition to H(w) by

$$H = 2\begin{bmatrix} --x_1-- \\ --x_2-- \\ ... \\ --x_p-- \end{bmatrix}\begin{bmatrix} | & | & ... & | \\ x_1 & x_2 & ... & x_p \\ | & | & ... & | \end{bmatrix} = VV^T$$

$$V = \sqrt{2}\begin{bmatrix} --x_1-- \\ --x_2-- \\ ... \\ --x_p-- \end{bmatrix}$$

Thus H is positive semi-definite, which implies that L(w) is convex.

1.2

$$H_{jk} = \frac{\partial L}{\partial w_j} \qquad = -2\sum_{i=1}^{n} x_j^{(i)}(y^{(i)} - w \cdot x^{(i)})$$

$$\nabla L(w) = -2\sum_{i=1}^{n} x^{(i)}(y^{(i)} - w \cdot x^{(i)})$$

$$w_{t+1} = w_t - \eta_t \nabla L(w_t)$$

$$w_{t+1} = w_t + 2\eta_t \sum_{i=1}^{n} x^{(i)}(y^{(i)} - w_t \cdot x^{(i)})$$

1.3

$$w_{t+1} = w_t - \eta_t H^{-1}(w_t) \nabla L(w_t)$$

Because H(w) is positive semi-definite, H(w) can be written as

$$H(w) = Q \Lambda Q^T$$

Thus,

$$H^{-1}(w) = Q \Lambda^{-1} Q^T$$

where $H^{-1}(w_t)$ can be easily obtained.

$$w_{t+1} = w_t - \eta_t \frac{L(w)}{\nabla L(w_t)} = w_t - \eta_t \frac{L(w)}{2 \sum_{i=1}^{n} x^{(i)} (y^{(i)} - w_t \cdot x^{(i)}}$$

2 Convexity

2.1

$H_{jk} = \frac{\partial^2 f}{\partial x_j \partial x_k}$
$= \frac{\partial}{\partial x_j}(\frac{\partial x^T M x}{\partial x_k})$
$= \frac{\partial}{\partial x_j}(\frac{\partial \sum_{i,j} M_{ij} x_i x_j}{\partial x_k})$
$= \frac{\partial}{\partial x_j}(\sum_i M_{ik} x_i + \sum_i M_{ki} x_i)$
$= M_{jk} + M_{kj}$
Therefore,

$$H = M + M^T$$

Because M is positive semi-definite, M^T and H are also positive semi-definite. Thus f is convex.

2.2

$H_{jk} = \frac{\partial^2 f}{\partial x_j \partial x_k}$
$= \frac{\partial}{\partial x_j}(\frac{\partial e^{u \cdot x}}{\partial x_k})$
$= \frac{\partial}{\partial x_j}(u_k e^{u \cdot x})$
$= u_k u_j e^{u \cdot x}$

$$H = \begin{bmatrix} - - u_1 e^{u \cdot x/2} - - \\ - - u_2 e^{u \cdot x/2} - - \\ \dots \\ - - u_p e^{u \cdot x/2} - - \end{bmatrix} \begin{bmatrix} | & | & \dots & | \\ u_1 e^{u \cdot x/2} & u_2 e^{u \cdot x/2} & \dots & u_p e^{u \cdot x/2} \\ | & | & \dots & | \end{bmatrix} = VV^T$$

$$V = \begin{bmatrix} - - u_1 e^{u \cdot x/2} - - \\ - - u_2 e^{u \cdot x/2} - - \\ \dots \\ - - u_p e^{u \cdot x/2} - - \end{bmatrix}$$

Thus H is positive semi-definite, which implies that f is convex.

2.3

$\because$ g and h are convex
$\therefore \forall\ a, b \in R^p,\ \theta \in [0,1] \quad g(\theta a + (1-\theta)b) \le \theta g(a) + (1-\theta) g(b)$
$\therefore \forall\ a, b \in R^p,\ \theta \in [0,1] \quad h(\theta a + (1-\theta)b) \le \theta h(a) + (1-\theta) h(b)$
$\therefore \forall\ a, b \in R^p,\ max[g(\theta a+(1-\theta)b), h(\theta a+(1-\theta)b)] \le \theta max[g(a), h(a)] + (1-\theta) max[g(b), h(b)]$
$\therefore \forall\ a, b \in R^p,\ \theta \in [0,1] \quad f(\theta a + (1-\theta)b) \le \theta f(a) + (1-\theta) f(b)$
$\therefore$ f is convex

3 Logistic regression using gradient descent

3.1 b

For part b, a function named logistic is implemented with parameters of input data, input label, max_iteration and step size.
Four samples are shown below in order to show the influence of step size and maximum iteration.

w = logistic(X, Y, 100000, 0.1)
iteration = 29513
w = [24.28620473 2.59508653 -89.11172933]

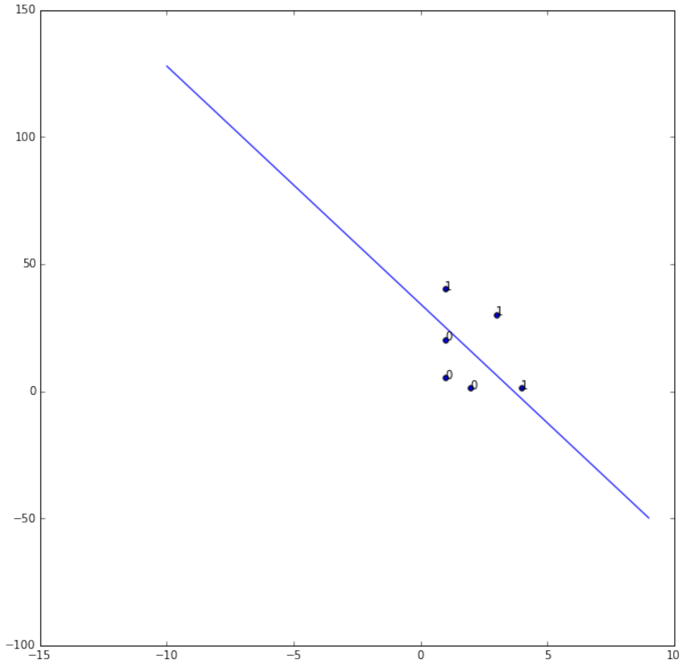

Figure 1: decision boundary and six points

w = logistic(X, Y, 100000, 0.05)
iteration = 100000
w = [8.81024328 0.64992294 -27.95754635]

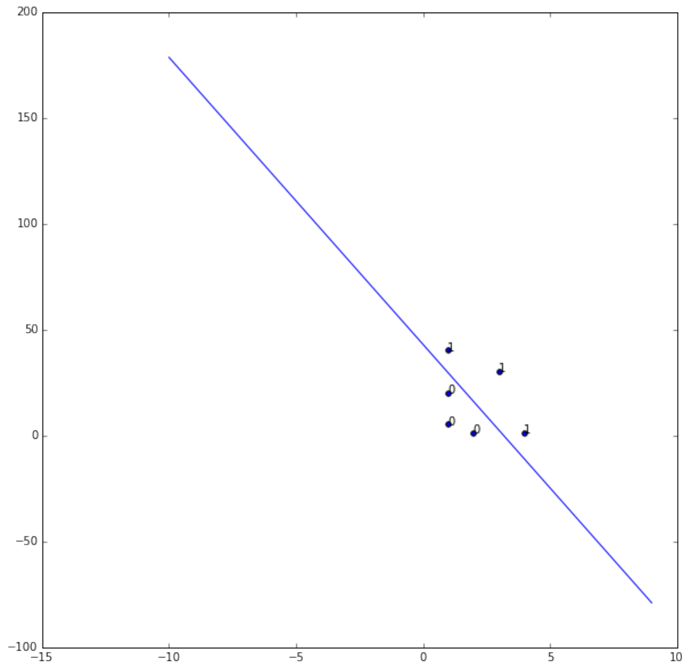

Figure 2: decision boundary and six points

w = logistic(X, Y, 100000, 0.2)
iteration = 26769
w = [50.02437432 4.63249848 -193.38130935]

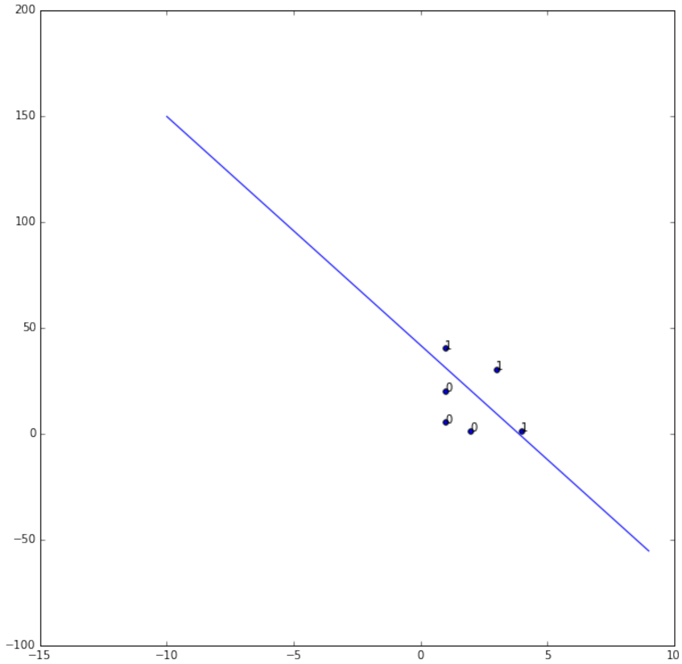

Figure 3: decision boundary and six points

w = logistic(X, Y, 500, 0.2)
iteration = 500
w = [16.61027093 5.03188882 -65.61743524]

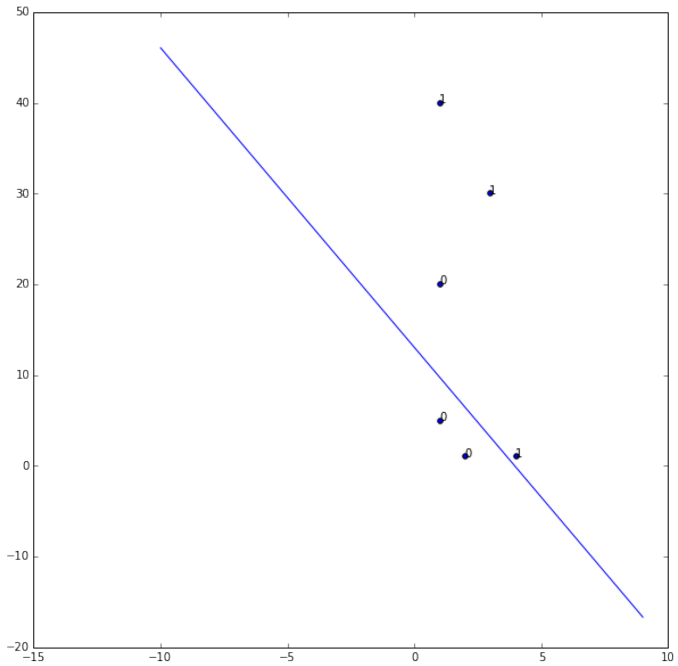

Figure 4: decision boundary and six points

3.2 c

After scaling down the x2-axis, the number of iterations needed for convergence increases. But the corresponding margin also increases, in other words, the decision boundary divides the points in a better way.
X = [[2,0.1,1], [1,2,1], [1,0.5,1], [4,0.1,1], [1,4,1], [3,3,1]]
Y = [-1, -1, -1, 1, 1, 1]
w = logistic(X, Y, 100000, 0.1)
iteration = 100000
w = [9.55167574 7.04368379 -30.3522845]

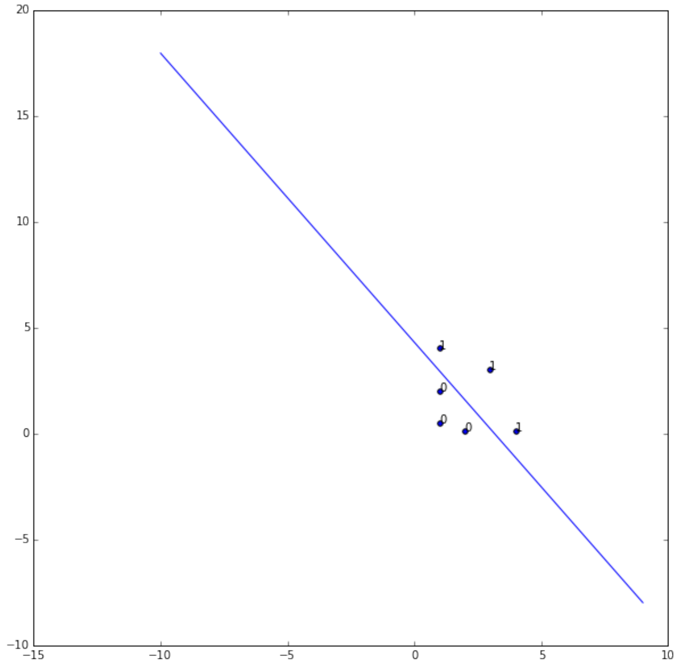

Figure 5: decision boundary and six points

3.3 d

I used two bi-variate Gaussians, each generate 50 random samples.
a = np.random.multivariate_normal([0,0], [[1,0],[0,2]], 50)
b = np.random.multivariate_normal([3,3], [[3,1],[1,2]], 50)
w = logistic(X, Y, 100000, 0.07)
iteration = 134
w = [-1.1394483 -1.55500097 3.77205524]
There are many mis-classified points as a result of overlapping classes.

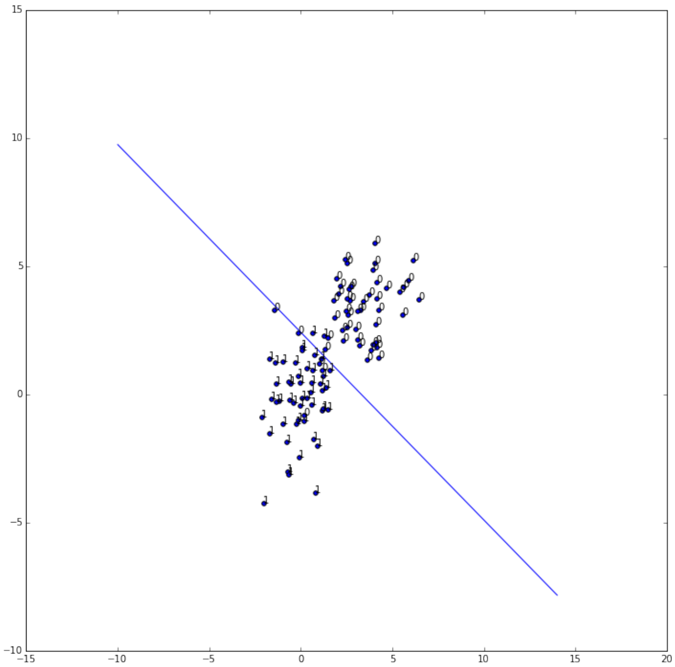

Figure 6: decision boundary and six points