Feras Khemakhem

CSCE 489-500

HW#2

1) (a) The maximum likelihood starts when

$$P(y|x) = \begin{cases} h(x) & y=+1 \\ 1-h(x) & y=-1 \end{cases}$$

To find maximum likelihood, we want to find the minimum of

$$\frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{P(y_n|x_n)}\right)$$

Using the first definition and substitution, we want to minimize

$$\frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{h(x_n) \text{ if } y_n=+1}\right) + \ln\left(\frac{1}{1-h(x_n) \text{ if } y_n=-1}\right)$$

This, along with the probability distributions mentione at the end of the problem, equates to

$$\frac{1}{N}\sum_{n=1}^{N} [\![y_n=+1]\!] \ln\frac{1}{h(x_n)} + [\![y_n=-1]\!] \ln\frac{1}{1-h(x_n)}$$

which is $E_{in}(w)$. Since $h(x_n)$ is the only varying value in this equation, $h(x_n)$ must be adjusted to minimize $E_{in}(w)$.

$\therefore$ QED

(b) $E_{in}(w)$ from part (a) can be simplified to

$$E_{in}(w) = \sum_{n=1}^{N} \ln \frac{1}{P(y_n \mid x_n)}$$

According to 3.8, $P(y \mid x) = \theta(y w^T x)$, so this is

$$E_{in}(w) = \sum_{n=1}^{N} \ln \frac{1}{\theta(y_n w^T x_n)}$$

Finally, since $\theta(s) = \frac{1}{1+e^{-s}}$, then

$$E_{in}(w) = \sum_{n=1}^{N} \ln \left(1 + e^{-y_n w^T x_n}\right)$$

This in N times as large as $E_{in}(w)$ in (3.9), and since N does not change when minimizing the $E_{in}(w)$ deducted, it should also be the same as $E_{in}(w)$ in (3.9). The min w of the sum in $E_{in}(w)$ deducted is the same w min in (3.9), since both have the same input nature.

2) (a)

$$E_2(w) = ||Xw - y||^2 + \lambda ||w||^2$$

$$\nabla E_2(w) = 2(X^T X w - X^T y) + 2\lambda w$$

Set $\nabla E_2(w) = 0$

$$0 = (X^T X w - X^T y) + \lambda w$$

$$X^T X w + \lambda w = X^T y$$

$$\boxed{w = (X^T X + \lambda)^{-1} X^T y}$$

(b) The matrix being inversed in this w is no longer $X^T X$, so $X^T X$ no longer needs to be nonsingular. $(X^T X + \lambda)$ now has to be nonsingular, but since $\lambda > 0$, this will always be nonsingular!

3) $E(w) = \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + e^{-y_n w^T x_n}\right)$

(a) $\nabla E(w) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{1+e^{-y_n w^T x_n}} \cdot (-y_n x_n)$

$\theta(-y_n w^T x) = \frac{1}{1+e^{-y_n w^T x}}$

$\therefore \quad \boxed{\nabla E(w) = \frac{1}{N} \sum_{n=1}^{N} [\theta(-y_n w^T x_n)] \cdot (-y_n x_n)}$

(b) Linear regression can be uncertain, so values between 0 and 1 for $\theta(s)$ are understanable and acceptable θ is a soft threshhold. However, the classification is still a hard threshold (ie, it needs to be -1 or 1). So, this is why the boundry is linear yet the θ function outputs nonlinear results.

(c) The decision boundary is still linear because it comes two discriminant functions. The decision surface is a hyperplane

(d) The z in $\theta(z)$ is always a linear function of x, making the logostic regression to always result in linear decision boundary.

4) (d) The 3rd order polynomial transform is what I would recommend to a customer. This is because the test accuracies are all a little higher for third order. The better in-sample fit returned the better out-sample performance.

```
Testing with Linear Logistic Regression:
Max iteration testcase 0: Max iter: 100 Train accuracy: 0.834721,
Test accuracy: 0.827830
Max iteration testcase 1: Max iter: 200 Train accuracy: 0.924407,
Test accuracy: 0.900943
Max iteration testcase 2: Max iter: 500 Train accuracy: 0.966047,
Test accuracy: 0.941038
Max iteration testcase 3: Max iter: 1000 Train accuracy: 0.973735,
Test accuracy: 0.950472
Learning rate testcase 0: Learning rate: 0.10 Train accuracy: 0.966047,
Test accuracy: 0.941038
Learning rate testcase 1: Learning rate: 0.20 Train accuracy: 0.973735,
Test accuracy: 0.950472
Learning rate testcase 2: Learning rate: 0.50 Train accuracy: 0.978860,
Test accuracy: 0.962264
Testing with 3rd order polynomial Logistic Regression:
Max iteration testcase 0: Max iter: 100 Train accuracy: 0.924407,
Test accuracy: 0.898585
Max iteration testcase 1: Max iter: 200 Train accuracy: 0.958360,
Test accuracy: 0.941038
Max iteration testcase 2: Max iter: 500 Train accuracy: 0.970532,
Test accuracy: 0.948113
Max iteration testcase 3: Max iter: 1000 Train accuracy: 0.975016,
Test accuracy: 0.955189
Learning rate testcase 0: Learning rate: 0.10 Train accuracy: 0.970532,
Test accuracy: 0.948113
Learning rate testcase 1: Learning rate: 0.20 Train accuracy: 0.975016,
Test accuracy: 0.955189
Learning rate testcase 2: Learning rate: 0.50 Train accuracy: 0.978219,
Test accuracy: 0.964623
```