CSE 404 Introduction to Machine Learning

Python Lab Report for Logistic Regression.

HW 5

By Sayem Lincoln

PID - A54207835

Problem 1

(a) If we are learning from ±1 data to predict a noisy

target P(y | x) with candidate hypothesis h, then the maximum

likelihood method reduces to the task of finding h that minimizes cross entropy error where $p = [\![y = +1]\!]$ and $q = h(x_n)$:

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

$$= \sum_{n=1}^{N} p \ln \frac{1}{q} + (1-p) \ln \frac{1}{1-q} \qquad (1.1)$$

Because minimizing the in sample error above is equivalent to minimizing the maximum likelihood in sample error:

$$E_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} \ln \frac{1}{\theta(y_n w^t x_n)} = \frac{1}{N} \sum_{n=1}^{N} \ln(1 + e^{-y_n w^t x_n}) \qquad (1.2)$$

(b) According to the definition of the entropy of a random variable, we can see that the entropy is related to the expectation of the random variable:

$$H(X) = \sum_{x} p(x) * \ln p(x) = \mathbb{E}_{X \sim p(x)} \left[\ln \frac{1}{p(x)} \right]$$

According to the task, we have true distribution $\{p, 1-p\}$ and approximated distribution $\{q, 1-q\}$. The inefficiency of assuming that the true distribution is $\{q, 1-q\}$, not $\{p, 1-p\}$ can be measured with relative entropy or Kullback-Leibler distance. In other words, relative entropy is a measure of the distance between two distributions:

$$D(p||q) = \mathbb{E}_{X \sim p(x)} \left[\ln \frac{p(x)}{q(x)} \right] = \mathbb{E}_{X \sim p(x)} \left[\ln \frac{1}{q(x)} - \ln \frac{1}{p(x)} \right]$$

The cross-entropy defined as

$$H(p,q) = H(p) + D(p||q) = \mathbb{E}_{X\sim p(x)}\left[\ln\frac{1}{q(x)}\right] \quad (1.3)$$

According to the task, $h(x) = \theta(w^t x)$, hence minimizing the in sample cross-entropy error (1.1) is equivalent to minimizing the one in (1.2) because (1.3) have the same sign and proportional to (1.2).

Problem 2

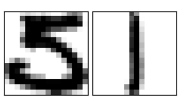
$$E_{in}(w) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n x_n}{1+e^{y_n w^t x_n}} = \frac{1}{N}\sum_{n=1}^{N} -y_n x_n \theta(-y_n w^t x_n)$$

Because $\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$ hence $\theta(-y_n w^t x_n) = \frac{1}{1+e^{y_n w^t x_n}}$

'misclassified ' example contributes more to the gradient than a correctly classified one because 'misclassified' example will make gradient to change his direction and weights will be updated accordingly.

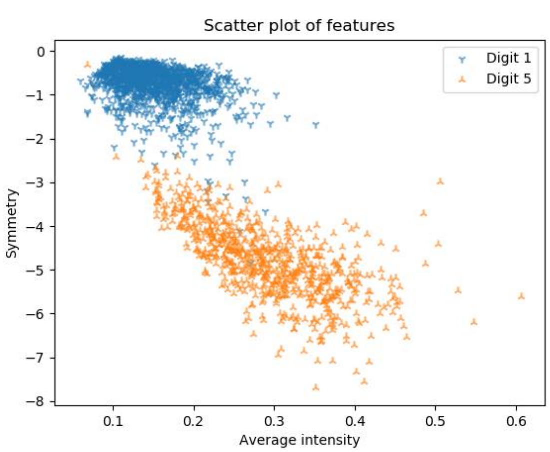
Problem 3

(a)



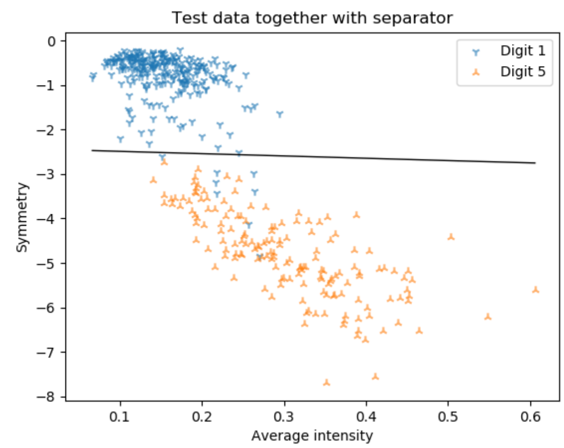
(b) I've used symmetry and average intensity (as discussed in class)

(c)



Problem 4

(a)



(b)

```
Accuracy on train dataset: 99.55%
Ein: 1.94%
Accuracy on test dataset: 98.11%
Etest: 8.15%
```

(c)

```
Accuracy on train dataset: 99.49%
Ein: 1.79%
Accuracy on test dataset: 98.11%
Etest: 8.85%
```

(d) I would use model without 3rd order polynomial transform because model with 3rd order polynomial transform may suffer from underfitting or overfitting if you will set max number of iteration to the wrong value (it may drastically affect E_{out} in a bad way).