# CS6350 - Homework/Assignment-6

Arnab Das(u1014840)

December 6, 2016

## 1: Warmup: Probabilities

**(1)** Given

$$P(A_1) = P(A_2) = P(A_1|A_2) = \frac{1}{2}$$

From conditional probability:

$$P(A_1, A_2) = P(A_1|P(A_2))P(A_2) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

Also, if we calculate the product of their individual probabilities :

$$P(A_1)P(A_2) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

Hence, from the above two formulations, we have:

$$P(A_1, A_2) = P(A_1|P(A_2))P(A_2) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} = P(A_1)P(A_2)$$

Thus, $A_1, A_2$ are independent

**(2)** Given, $A_1, A_2, A_3$ are mutually exclusive, $P(A_i) = \frac{1}{3}$ and $P(A_4|A_i) = \frac{i}{6}$ . Then, we have the following probabilities:

$$P(A_1) = P(A_2) = P(A_3) = P(A_4) = \frac{1}{3}$$

$$P(A_4|A_{i=1}) = \frac{1}{6}$$

$$P(A_4|A_{i=2}) = \frac{1}{3}$$

$$P(A_4|A_{i=3}) = \frac{1}{2}$$

Using total probability theorem to evaluate $P(A_4)$,

$$P(A_4) = P(A_1)P(A_4|A_1) + P(A_2)P(A_4|A_2) + P(A_3)P(A_4|A_3)$$

$$P(A_4) = \frac{1}{3} \times \frac{1}{6} + \frac{1}{3} \times \frac{1}{3} + \frac{1}{3} \times \frac{1}{2} = \frac{1}{3}$$

**(3)** Let n be the number at the top when a fair six-sided die is tossed. If a fair coin is tossed n times, required to find the probability of exactly two heads.

Now, we can get two heads only if the toss of the dice gives a number greater than equal to 2. For a single toss of an unbiased coin, we have the probabilites, $P(H) = \frac{1}{2}$ and $P(T) = \frac{1}{2}$ . Our problem is conditioned on the value of n that we get from the dice. So, we can write the probability of number of heads being exactly 2 as:

$$P(H = 2) = \sum_{n=2}^{6} P(H = 2|n)P(n)$$

$P(H=2) = P(H=2|n=2)P(n=2) + P(H=2|n=3)P(n=3) + P(H=2|n=4)P(n=4)$
$+P(H=2|n=5)P(n=5) + P(H=2|n=6)P(n=6)$

Now for a specific n, $n \geq 2$, the probability of exactly two heads $= \binom{n}{2}P(H)^2P(T)^{n-2} = \binom{n}{2}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^{n-2} = \binom{n}{2}\left(\frac{1}{2}\right)^n$

Also, since the we are given an unbiased fair dice, hence the probabilitities of getting a value of n between 1 to 6 will be equally probable, Hence,

$$P(n=2) = P(n=3) = P(n=4) = P(n=5) = P(n=6) = \frac{1}{6}$$

Using this formulation, in the expression for finding the probability of exactly two heads , we get:

$$P(H=2) = \binom{2}{2}\left(\frac{1}{2}\right)^2\frac{1}{6} + \binom{3}{2}\left(\frac{1}{2}\right)^3\frac{1}{6} + \binom{4}{2}\left(\frac{1}{2}\right)^4\frac{1}{6} + \binom{5}{2}\left(\frac{1}{2}\right)^5\frac{1}{6} + \binom{6}{2}\left(\frac{1}{2}\right)^6\frac{1}{6}$$

$$P(H=2) = \left(\frac{1}{2}\right)^2\frac{1}{6} + 3\left(\frac{1}{2}\right)^3\frac{1}{6} + 6\left(\frac{1}{2}\right)^4\frac{1}{6} + 10\left(\frac{1}{2}\right)^5\frac{1}{6} + 15\left(\frac{1}{2}\right)^6\frac{1}{6}$$

$$P(H=2) = \frac{1}{6}\left(\left(\frac{1}{2}\right)^2 + 3\left(\frac{1}{2}\right)^3 + 6\left(\frac{1}{2}\right)^4 + 10\left(\frac{1}{2}\right)^5 + 15\left(\frac{1}{2}\right)^6\right) = \frac{33}{128}$$

**(4)** Given $P(A_1) = a_1$ and $P(A_2) = a_2$. From conditional probability, we can write:

$$P(A_1|A_2) = \frac{P(A_1 \cap A_2)}{P(A_2)} = \frac{P(A_1) + P(A_2) - P(A_1 \cup A_2)}{P(A_2)}$$

Now, $P(A_1 \cup A_2) \leq 1$. Thus we get,

$$P(A_1|A_2) = \frac{P(A_1) + P(A_2) - P(A_1 \cup A_2)}{P(A_2)} \leq \frac{P(A_1) + P(A_2) - 1}{P(A_2)}$$

replacing the given values of the probabilities, we get:

$$P(A_1|A_2) = \frac{P(A_1) + P(A_2) - P(A_1 \cup A_2)}{P(A_2)} \leq \frac{a_1 + a_2 - 1}{a_2}$$

**(5a)** If $A_1$ and $A_2$ are independent variables, show that: $E[A_1 + A_2] = E[A_1] + E[A_2]$
**Proof:** Assuming $A_1$ and $A_2$ being independent discrete random variables, although the same holds for the continuous case. Then the expectation of $A_1$ is written as:

$$E[A_1] = \sum_{a_1} a_1 P(A_1 = a_1)$$

where $a_1$ represents the values that $A_1$ can take and $P(a_2)$ represents their probabilities. Similarly,

$$E[A_2] = \sum_{a_2} a_2 P(A_2 = a_2)$$

Then, we can write:

$$E[A_1 + A_2] = \sum_{a_1}\sum_{a_2}(a_1 + a_2)P(A_1 = a_1, A_2 = a_2)$$

or,

$$E[A_1 + A_2] = \sum_{a_1}\sum_{a_2} a_1 P(A_1 = a_1, A_2 = a_2) + \sum_{a_1}\sum_{a_2} a_2 P(A_1 = a_1, A_2 = a_2)$$

or,

$$E[A_1 + A_2] = \sum_{a_1} a_1 \sum_{a_2} P(A_1 = a_1, A_2 = a_2) + \sum_{a_2} a_2 \sum_{a_1} P(A_1 = a_1, A_2 = a_2)$$

Now, $\sum_{a_2} P(A_1 = a_1, A_2 = a_2) = P(A_1 = a_1)$, since we are summing over all the possible $a_2$ values that $A_2$ can take. Similarly, $\sum_{a_1} P(A_1 = a_1, A_2 = a_2) = P(A_2 = a_2)$. Plugging these back into the main equation, we get:

$$E[A_1 + A_2] = \sum_{a_1} a_1 P(A_1 = a_1) + \sum_{a_2} a_2 P(A_2 = a_2) = E[A_1] + E[A_2]$$

**(5b)** If $A_1$ and $A_2$ are independent variables, show that: $var[A_1 + A_2] = var[A_1] + var[A_2]$
**Proof:** Variance of a random variable X is defined as:

$$var(X) = E[(X - E[X])^2] = E[X^2 - 2XE[X] + (E[X])^2$$

using the linearity of expectation as proved in the previous question, we can separate out the terms as:

$$var(X) = E[X^2] - 2E[XE[X]] + E[(E[X])^2]$$

Now, $E[X]$ is the theoretical mean of the distribution and hence a constant. ALso, the expectation of a constant terms is the term itself. Hence:

$$var(X) = E[X^2] - 2E[X]E[X] + E[X]^2] = E[X^2] - (E[X])^2$$

Thus, the variance of the given independent random variables, $A_1, A_2$ will be:

$$var(A_1) = E[A_1^2] - (E[A_1])^2]$$

$$var(A_2) = E[A_2^2] - (E[A_2])^2]$$

Then the variance of the sum of the two independent variables will be:

$$var[A_1 + A_2] = E[(A_1 + A_2)^2] - (E[A_1 + A_2])^2 = E[(A_1 + A_2)^2] - (E[A_1] + E[A_2])^2$$

$$var[A_1 + A_2] = E[A_1^2] + E[A_2^2] + 2E[A_1A_2] - (E[A_1])^2 - (E[A_2])^2 - 2E[A_1]E[A_2]$$

$$var[A_1 + A_2] = (E[A_1^2] - (E[A_1])^2) + (E[A_2^2] - (E[A_2])^2) + 2E[A_1A_2] - 2E[A_1]E[A_2]$$

$$var[A_1 + A_2] = var[A_1] + var[A_2] + 2E[A_1A_2] - 2E[A_1]E[A_2]$$

$$var[A_1 + A_2] = var[A_1] + var[A_2] + 2\sum_{a_1}\sum a_2 a_1 a_2 P(A_1 = a_1, A_2 = a_2) - 2E[A_1]E[A_2] \tag{1}$$

In (1), since the random variables, $A_1, A_2$ are independent, so we have, $P(A_1 = a_1, A_2 = a_2) = P(A_1 = a_1)P(A_2 = a_2)$. Plcaing this back in the equation and separating out the sum with their corresponding variables we get:

$$var[A_1 + A_2] = var[A_1] + var[A_2] + 2\sum_{a_1} a_1 P(A_1 = a_1) \sum_{a_2} a_2 P(A_2 = a_2) - 2E[A_1]E[A_2]$$

$$var[A_1 + A_2] = var[A_1] + var[A_2] + 2E[A_1]E[A_2] - 2E[A_1]E[A_2]$$

$$var[A_1 + A_2] = var[A_1] + var[A_2]$$

## 2: Naive Bayes

**(1.a)** We are given the true distribution values. If we draw infinite data from this distribution, then the empirically evaluated probabilities will converge close to the true distribution according to the law of large numbers which suggests for infinitely drawn data the sampled mean and the theoretical mean converges. Hence, after seeing infinitely drawn data, the evaluated probabilities can be approximated to the true distribution, in which case we have:

$$\hat{P}(x_1 = -1|y = -1) = 0.8$$

$$\hat{P}(x_1 = 1|y = -1) = 0.2$$
$$\hat{P}(x_1 = -1|y = 1) = 0.1$$
$$\hat{P}(x_1 = 1|y = 1) = 0.9$$
$$\hat{P}(y = 1) = 0.9$$
$$\hat{P}(y = -1) = 0.1$$

**(1.b)** Using the above probability values, we evaluate the required probabilities of the given table as :

$$\hat{P}(x_1 = -1, y = -1) = \hat{P}(x_1 = -1|y = -1)\hat{P}(y = -1) = 0.08$$

$$\hat{P}(x_1 = 1, y = -1) = \hat{P}(x_1 = 1|y = -1)\hat{P}(y = -1) = 0.02$$
$$\hat{P}(x_1 = -1, y = 1) = \hat{P}(x_1 = -1|y = 1)\hat{P}(y = 1) = 0.09$$
$$\hat{P}(x_1 = 1, y = 1) = \hat{P}(x_1 = 1|y = 1)\hat{P}(y = 1) = 0.81$$

Hence, the prediction will be :

$$y'\Big|_{x_1=-1} = argmax_y \hat{P}(x_1 = -1, y) = +1$$

$$y'\Big|_{x_1=+1} = argmax_y \hat{P}(x_1 = -1, y) = +1$$

Thus, the table will be :

Table 1: Probability Table-1: Input $x_1$, $\hat{P}(x_1, y = -1)$,$\hat{P}(x_1, y = 1)$ ,Prediction: $y' = argmax_y \hat{P}(x_1, y)$

| Input $x_1$ | $\hat{P}(x_1, y = -1)$ | $\hat{P}(x_1, y = 1)$ | Prediction: $y' = argmax_y \hat{P}(x_1, y)$ |
|---|---|---|---|
| -1 | 0.08 | 0.09 | +1 |
| +1 | 0.02 | 0.81 | +1 |

**(1.c)** To find the error in predictions, we need to find $P(y' \neq y)$, Thus:

$$P(y' \neq y) = P(y' \neq y, x_1 = -1) + P(y' \neq y, x_1 = +1)$$

$P(y' \neq y) = P(y' = 1, x_1 = -1)P(y = -1, x_1 = -1) + P(y' = -1, x_1 = -1)P(y = +1, x_1 = -1)$
$+P(y' = +1, x_1 = +1)P(y = -1, x_1 = +1) + P(y' = -1, x_1 = +1)P(y = +1, x_1 = +1)$
$P(y' \neq y) = P(y' = 1, y = -1, x_1 = -1) + P(y' = -1, y = +1, x_1 = -1)$
$+P(y' = +1, y = -1, x_1 = +1) + P(y' = -1, x_1 = +1)P(y = +1, x_1 = +1)$
$P(y' \neq y) = 0.08 + 0 + 0.02 + 0 = 0.1$

**(2.a)** Given a binary classification problem with two features , $x_1, x_2$, both of which can take discrete values {-1,1}. Additionaly, the feature $x_2$, is exactly identical to first feature $x_1$.
Since, $x_1$ and $x_2$ are identical/same feature, hence:

$$P(x_1|y) = P(x_2|y)$$

and, since $x_1$ and $x_2$ will have identical values under all conditions, hence
$P(x_1 = a_1|x_2 = a_2) = 1$, if $a_1 = a_2$
$P(x_1 = a_1|x_2 = a_2) = 0$, if $a_1 \neq a_2$

Now, from conditional probability, we can write:

$$P(x_1, x_2|y) = P(x_1|x_2, y)P(x_2|y)$$

For condittional independence, we require,

$$P(x_1, x_2|y) = P(x_1|y)P(x_2|y) \tag{2}$$

However, in this case, since $x_1$ and $x_2$, are the same feature, hence the probability of $x_1$, given $x_2$ and $y$, will be same as probability of $x_2$ given y, multiplied with the gating function $P(x_1|x_2)$ which means $P(x_1|x_2, y) = P(x_1|x_2)$ such that if $x_1 = x_2$, it is one, and if $x_1 \neq x_2$, it is zero. Thus, we have,

$$P(x_1, x_2|y) = P(x_1|x_2, y)P(x_2|y) = P(x_1|x_2)P(x_2|y)$$

This differs from the requirement of conditional independence in (2). Hence, $x_1$ and $x_2$ are **not** conditionally independent, given y.

**(2.b)** Given that $\hat{P}(x_1|y)$, $\hat{P}(x_2|y)$ and $\hat{P}(y)$ represent the learned parameters of a Naive Bayes(NB) classifier on infinite data set, so the values of these probabilities will be same as in the previous question. Hence,evaluating the probabilities in the table:
1.$\hat{P}(x_1 = -1, x_2 = -1, y = -1) = \hat{P}(x_1 = -1|y = -1)\hat{P}(x_2 = -1|y = -1)\hat{P}(y = -1) = 0.8 \times 0.8 \times 0.1 = 0.064$
2.$\hat{P}(x_1 = -1, x_2 = 1, y = -1) = \hat{P}(x_1 = -1|y = -1)\hat{P}(x_2 = 1|y = -1)\hat{P}(y = -1) = 0.8 \times 0.2 \times 0.1 = 0.016$
3.$\hat{P}(x_1 = 1, x_2 = -1, y = -1) = \hat{P}(x_1 = 1|y = -1)\hat{P}(x_2 = -1|y = -1)\hat{P}(y = -1) = 0.2 \times 0.8 \times 0.1 = 0.016$
4.$\hat{P}(x_1 = 1, x_2 = 1, y = -1) = \hat{P}(x_1 = 1|y = -1)\hat{P}(x_2 = 1|y = -1)\hat{P}(y = -1) = 0.2 \times 0.2 \times 0.1 = 0.004$

5.$\hat{P}(x_1 = -1, x_2 = -1, y = 1) = \hat{P}(x_1 = -1|y = 1)\hat{P}(x_2 = -1|y = 1)\hat{P}(y = 1) = 0.1 \times 0.1 \times 0.9 = 0.009$
6.$\hat{P}(x_1 = -1, x_2 = 1, y = 1) = \hat{P}(x_1 = -1|y = 1)\hat{P}(x_2 = 1|y = 1)\hat{P}(y = 1) = 0.1 \times 0.9 \times 0.9 = 0.081$
7.$\hat{P}(x_1 = 1, x_2 = -1, y = 1) = \hat{P}(x_1 = 1|y = 1)\hat{P}(x_2 = -1|y = 1)\hat{P}(y = 1) = 0.9 \times 0.1 \times 0.9 = 0.081$
8.$\hat{P}(x_1 = 1, x_2 = 1, y = 1) = \hat{P}(x_1 = 1|y = 1)\hat{P}(x_2 = 1|y = 1)\hat{P}(y = 1) = 0.9 \times 0.9 \times 0.9 = 0.729$

Hence, the predictions will be like: $y'\Big|_{x_1=-1,x_2=-1} = argmax_y\hat{P}(x_1 = -1, x_2 = -1, y) = -1$. Similarly, filling up the table: Thus the table will be:

Table 2: Probability Table-1: Input $x_1$, $x_2$, $\hat{P}(x_1, x_2, y = -1)$,$\hat{P}(x_1, x_2, y = 1)$ ,Prediction: $y' = argmax_y\hat{P}(x_1, x_2, y)$

| Input $x_1$ | $x_2$ | $\hat{P}(x_1, x_2, y = -1)$ | $\hat{P}(x_1, x_2, y = 1)$ | Prediction: $y' = argmax_y\hat{P}(x_1, x_2, y)$ |
|---|---|---|---|---|
| -1 | -1 | 0.064 | 0.009 | -1 |
| -1 | 1 | 0.016 | 0.081 | +1 |
| 1 | -1 | 0.016 | 0.081 | +1 |
| 1 | 1 | 0.004 | 0.729 | +1 |

**(2.c)** To find the error in prediction, we need to find $P * y' \neq y)$. Thus,

$P(y' \neq y) = P(y' \neq y, x_1 = -1, x_2 = -1) + P(y' \neq y, x_1 = -1, x_2 = 1) + P(y' \neq y, x_1 = 1, x_2 = -1) + P(y' \neq y, x_1 = 1, x_2 = 1)$

$P(y' \neq y, x_1 = -1, x_2 = -1) = P(y' = 1, y = -1, x_1 = -1, x_2 = -1) + P(y' = -1, y = +1, x_1 = -1, x_2 = -1) = 0 + 0.009 = 0.009$
$P(y' \neq y, x_1 = -1, x_2 = +1) = P(y' = 1, y = -1, x_1 = -1, x_2 = +1) + P(y' = -1, y = +1, x_1 = -1, x_2 = +1) = 0.016 + 0 = 0.016$
$P(y' \neq y, x_1 = +1, x_2 = -1) = P(y' = 1, y = -1, x_1 = +1, x_2 = -1) + P(y' = -1, y = +1, x_1 = +1, x_2 = -1) = 0.016 + 0 = 0.016$
$P(y' \neq y, x_1 = +1, x_2 = +1) = P(y' = 1, y = -1, x_1 = +1, x_2 = +1) + P(y' = -1, y = +1, x_1 = +1, x_2 = +1) = 0.004$

Therefore,

$$P(y' \neq y) = 0.009 + 0.016 + 0.016 + 0.004 = 0.045$$

**(2.d)** Since Naive Bayes(NB) considers the features to be conditionally independent, hence if there exists features duplicated or strongly correlated, NB will treat them independently and give both of them strong weights such that they have double influence. On the other hand, Logistic-Regression(LR) does not have any conditional independence requirement and hence it performs better in cases where features might be duplicated or strongly correlated, since its minimization algorithm will tend to compensate for the correlation of features.

## 3: Naive Bayes and Linear Classifiers

Given inputs as d-dimensional feature vector , $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$, where each $x_j$ can be a real number. Classifier predicts 1 if:

$$P(y = 1|x) \geq P(y = 0|x)$$

$$\frac{P(y = 1|x)}{P(y = 0|x)} \geq 1$$

or equivalently:

$$\frac{P(x|y = 1)P(y = 1)}{P(x|y = 0)P(y = 0)} \geq 1$$

By the Naive Bayes assumption of conditional independence of the features, we have $P(x|y) = \prod_{j=0}^{d} P(x_j|y)$. Then we have $\frac{P(y = 1)}{P(y = 0)} \prod_{j=0}^{d} \frac{P(x_j|y = 1)}{P(x_j|y = 0)} \geq 1$ for the **decision boundary**. Suppose each $P(x_j|y)$ is defined using a Gaussian probability density function(pdf), one for each value of $y$ and $j$ with mean $\mu_{j,y}$ and variance $\sigma^2$. Thus, we can use the pdf expression for $P(x_j|y)$, such that the mean for $y = 1$ and a specific $j$ is denoted as $\mu_{j,y_1}$ and the mean for $y = 0$ and a specific $j$ is denoted as $\mu_{j,y_0}$. Also, let us denote the prior probability of $P(y = 1)$ as $p$. Then the prior probability of $P(y = 0)$ is $(1 - p)$. Hence, we can write out the decision boundary as:

$$\frac{p}{1-p} \prod_{j=0}^{d} \frac{\frac{1}{\sqrt[2]{2\pi\sigma^2}} e^{-\frac{(x_j - \mu_{j,y_1})^2}{2\sigma^2}}}{\frac{1}{\sqrt[2]{2\pi\sigma^2}} e^{-\frac{(x_j - \mu_{j,y_0})^2}{2\sigma^2}}} \geq 1$$

$$\frac{p}{1-p} \prod_{j=0}^{d} e^{\frac{1}{2\sigma^2}(\mu_{j,y_1} - \mu_{j,y_0})(2x_j - \mu_{j,y_0} - \mu_{j,y_1}} \geq 1$$

Taking natural lo on both sides and rearranging the terms to separate out $x_j$, we get:

$$\log(\frac{p}{1-p}) + \frac{1}{2\sigma^2}\sum_{j=0}^{d}(\mu_{j,y_1} - \mu_{j,y_0})(-\mu_{j,y_1} - \mu_{j,y_0}) + \frac{1}{2\sigma^2}\sum_{j=0}^{d} 2x_j(\mu_{j,y_1} - \mu_{j,y_0}) \geq 0$$

The means of each of the distributions are specific to the distributions and constant with respect to $x_j$. Furthermore, it is given the variance is same for all the distribution, hence it is also a constant with respect to $x_j$. Thus, from the above expression we can separate out the constant term as the bias, such that we can write the following:
bias = b = $\log(\frac{p}{1-p}) - \frac{1}{2\sigma^2}\sum_{j=0}^{d}(\mu_{j,y_1}^2 - \mu_{j,y_0}^2)$ and
weights, $w_j = \frac{1}{\sigma^2}(\mu_{j,y_1} - \mu_{j,y_0})$.
hence, we get the decision boundary as:

$$b + \sum_{j=0}^{d} w_j x_j \geq 0$$

Hence, our classifier is a linear classifier.

## 4: Experiment

Given minimization problem: $\min_{\mathbf{w}}\{\sum_{i=1}^{m}\log(1+\exp(-y_i\mathbf{w}^T\mathbf{x_i})) + \frac{1}{\sigma^2}\mathbf{w^T}\mathbf{w}\}$ **(1)** To find the derivative of the function: $g(\mathbf{w}) = \log(1+\exp(-y_i\mathbf{w^T}\mathbf{x_i}))$. For an n-dimensional vector, its gradient will have derivative components with respect to each feature,$w_j$, that will be:

$$\frac{\partial g(\mathbf{w})}{\partial w_j} = \frac{\exp(-y_i\mathbf{w^T}\mathbf{x_i})(-y_i x_{ij})}{1+\exp(-y_i\mathbf{w^T}\mathbf{x_i})} = \frac{-y_i x_{ij}}{1+\exp(y_i\mathbf{w^T}\mathbf{x_i})}$$

Thus, the overall gradient vector will be:

$$\nabla g(\mathbf{w}) = [\frac{\partial g}{\partial w_1}, \frac{\partial g}{\partial w_2}, \ldots, \frac{\partial g}{\partial w_n}] = \frac{-y_i\mathbf{x_i}}{1+\exp(y_i\mathbf{w^T}\mathbf{x_i})}$$

**(2)** Gradient update is the innermost step of SGD where a single example is treated as the entire dataset to compute the gradient. The objective function for a single example $(\mathbf{x_i}, y_i)$, will be:

$$J(\mathbf{w}) = \log(1+\exp(-y_i\mathbf{w^T}\mathbf{x_i})) + \frac{1}{\sigma^2}\mathbf{w^T}\mathbf{w}$$

The partial derivative with respect to weight component $w_j$ will be:

$$\frac{\partial J(\mathbf{w})}{\partial w} = \frac{-y_i x_{ij}}{1+\exp(y_i\mathbf{w^T}\mathbf{x_i})} + \frac{2}{\sigma^2}w_j$$

Thus the overall gradient will be:

$$\nabla J(\mathbf{w}) = [\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \ldots, \frac{\partial J}{\partial w_n}] = \frac{-y_i\mathbf{x_i}}{1+\exp(y_i\mathbf{w^T}\mathbf{x_i})} + \frac{2}{\sigma^2}\mathbf{w}$$

$$\nabla J(\mathbf{w}) = \frac{-y_i\mathbf{x_i}}{1+\exp(y_i\mathbf{w^T}\mathbf{x_i})} + \frac{2}{\sigma^2}\mathbf{w} \tag{3}$$

**(3)** In Stochastic Gradient Descent(SGD), we make an update after seeing an example and evaluating its gradient, and we tend to update the weight vector, $\mathbf{w}$, in the opposite direction of the gradient such that it moves closer towards the minima. Hence, the **update rule** is:

$$\mathbf{w_{t+1}} = \mathbf{w_t} - r_t\nabla J(\mathbf{w})$$

$$\mathbf{w_{t+1}} = \mathbf{w_t} - r_t\{\frac{-y_i\mathbf{x_i}}{1+\exp(y_i\mathbf{w^T x_i})} + \frac{2\mathbf{w_t}}{\sigma^2}\}$$

$$\mathbf{w_{t+1}} = \mathbf{w_t}(1 - \frac{2r_t}{\sigma^2}) + \frac{r_t y_i}{1+\exp(y_i\mathbf{w^T x_i})}\mathbf{x_i}$$

where $r_t$, is the adaptive learning rate in the t'th step, and is derived from: $r_t = \frac{r_0}{1+\frac{r_0 t}{\sigma^2}}$. Replacing the gradient from (3) , we get have the following **peusoCode** for stochastic gradient:

```
SGD(DataSet=D,Epochs=T)
    Initialize $\mathbf{w_0} = \mathbf{0}, t = 0$
    For epoch in $1 \dots T$
        permute DataSet D
        for each $< \mathbf{x_i}, y_i >$ D:
            $r_t = \frac{r_0}{1+\frac{r_0 t}{\sigma^2}}$
            $\mathbf{w_{t+1}} = \mathbf{w_t}(1 - \frac{2r_t}{\sigma^2}) + \frac{r_t y_i}{1+\exp(y_i\mathbf{w^T x_i})}\mathbf{x_i}$
            t = t+1
    return w
```

**(4)** The experiments were run on a cross-validate *sigmaSquare* and the *learningRate*. The selected list of cross-validation parameters were:
SigmaSquare = $[0.0001, 0.01, 16, 100, 169, 1000, 10000]$
Learningrate = $[10, 5, 0.1, 0.01, 0.05]$
After 5-fold cross validation, the chosen sigmaSquare and Learning Rates are: **Best-SigmaSquare =** 1000, **Best Initial Learning Rate =** 0.1.
For cross validation, 20 epochs were used, while for the final training and evaluating the objective function after every epoch, 100 epochs , were used.
**Final Training Accuracy = 84.268%**, and, **Final Test Accuracy = 84.54%**
Below table reports the accuracies seen during cross validation

Table 3: 5-fold cross validation on UCI data set

| SigmaSquare($\sigma^2$) | *LearningRate* | Avg.Training Accuracy(%) | Avg.Test Accuracy(%) |
|---|---|---|---|
| 0.0001 | 10 | 75.5416 | 75.5226 |
| 0.01 | 10 | 75.5416 | 75.5226 |
| 16 | 10 | 79.5946 | 79.5475 |
| 100 | 10 | 83.6009 | 82.9485 |
| 169 | 10 | 83.9399 | 83.5725 |
| 1000 | 10 | 84.7466 | 83.9157 |
| 10000 | 10 | 83.7295 | 82.6053 |
| 0.0001 | 5 | 75.5416 | 75.5226 |
| 0.01 | 5 | 75.5416 | 75.5226 |
| 16 | 5 | 79.5946 | 79.5787 |
| 100 | 5 | 83.6866 | 82.9797 |
| 169 | 5 | 83.9360 | 83.6661 |
| 1000 | 5 | 84.6960 | 83.9157 |
| 10000 | 5 | 82.2330 | 81.7784 |
| 0.0001 | 0.1 | 75.5416 | 75.5226 |
| 0.01 | 0.1 | 75.5416 | 75.5226 |
| 16 | 0.1 | 79.5284 | 79.4539 |
| 100 | 0.1 | 83.5736 | 83.0733 |
| 169 | 0.1 | 83.9088 | 83.3853 |

| | | | |
|---|---|---|---|
| **1000** | **0.1** | **84.3413** | **84.3369** |
| 10000 | 0.1 | 84.2283 | 83.6817 |
| 0.0001 | 0.01 | 75.5416 | 75.5226 |
| 0.01 | 0.01 | 75.5416 | 75.5226 |
| 16 | 0.01 | 79.5908 | 79.5787 |
| 100 | 0.01 | 83.5268 | 83.2449 |
| 169 | 0.01 | 83.9360 | 83.4945 |
| 1000 | 0.01 | 84.8597 | 84.1497 |
| 10000 | 0.01 | 85.0311 | 83.9625 |
| 0.0001 | 0.05 | 75.5416 | 75.5226 |
| 0.01 | 0.05 | 75.5416 | 75.5226 |
| 16 | 0.05 | 79.5752 | 79.5787 |
| 100 | 0.05 | 83.5658 | 83.0265 |
| 169 | 0.05 | 83.9594 | 83.5569 |
| 1000 | 0.05 | 84.6726 | 83.8533 |
| 10000 | 0.05 | 84.5869 | 83.5725 |

The below Figure(1) shows the plot of the objective function, and the next figure(2), shows the same objective function without connecting the plots. As figure(2) suggests, the objective function rapidly reduced within the initial epochs, however, at frequent intervals there comes to be some outliers. In our experiment, we also print the l2-norm of the weight vector after learning each epoch(Look at **finalLog** in the submission tar. The l2-norm suggests steady decrease in the weight vector, intially starting from **4.08** and finally steadying itself at **3.67**. We also experimented on the gradient descent over this data-set,figure(3) shows the plot of objective function for gradient descent that steadily decreases as expected but much slowly than compared to SGD. The gradient descent also shows few outliers which indicates that the objective function with this dataset might be weakly convex which may justify so many outliers in the SGD plot.

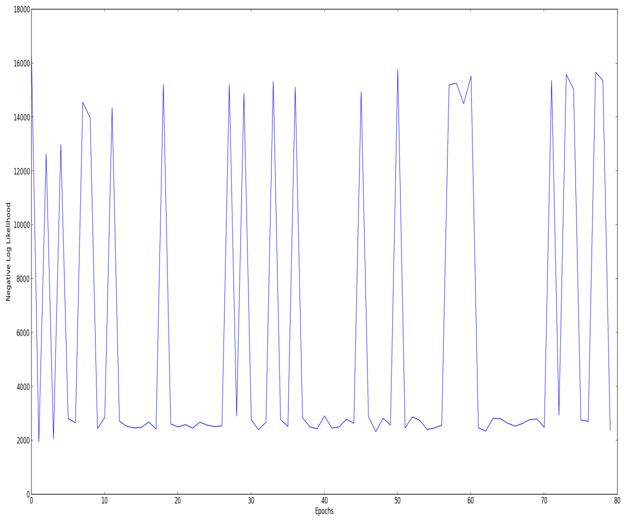

Figure 1: Objective function with straight line interpolation between points

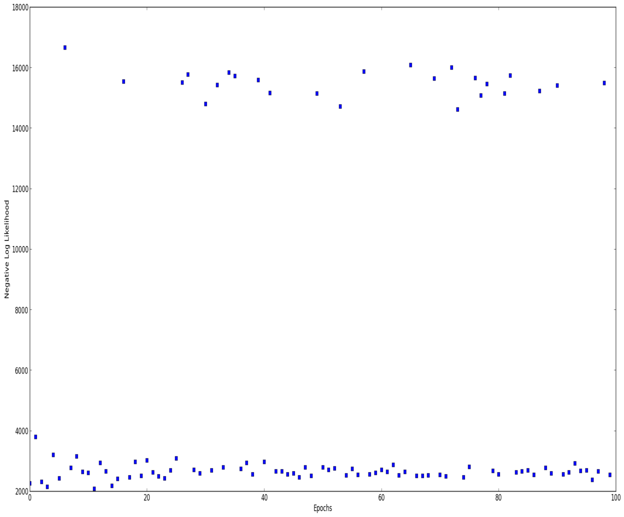

Figure 2: Objective function with straight line interpolation between points

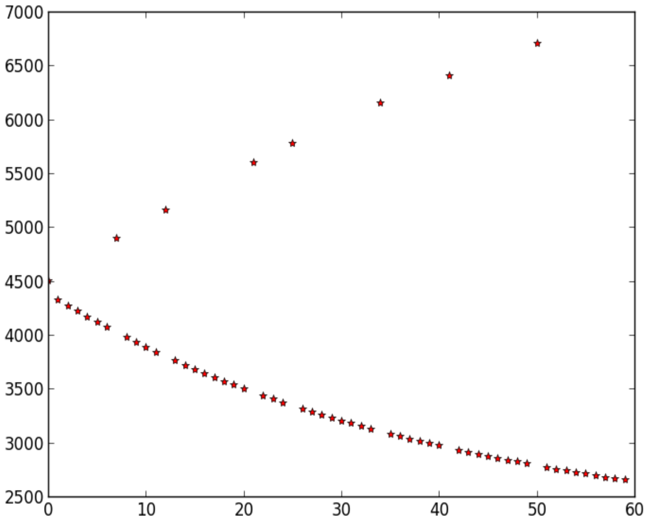

Figure 3: Objective function for gradient descent

Thats all for now!! Check out the code!!