Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 1

Given a logistic regression model, $p(y_n|x_n, w) = \dfrac{1}{1 + exp(-y_n w^T x_n)}$

Gaussian prior $p(w) = N(0, \lambda^{-1}I)$

The Negative log-likelihood is given below:

NLL(w)=$\sum_{n=1}^{N} log(1 + exp(-y_n w^T x_n) + log(exp(\lambda^T w\lambda))$

NLL(w)=$\sum_{n=1}^{N} log(1 + exp(-y_n w^T x_n) + \lambda^T w\lambda$

Using the identity:$log(a + b) = log(b) + log(\dfrac{a}{b+1})$

NLL(w)=$\sum_{n=1}^{N} log(exp(-y_n w^T x_n)) + log(\dfrac{1}{exp(-y_n w^T x_n + 1}) + \lambda^T w\lambda$

NLL(w)=$\sum_{n=1}^{N} -y_n w^T x_n + log(\dfrac{1}{exp(-y_n w^T x_n + 1)}) + \lambda^T w\lambda$

Taking the derivative of NLL(W) with respect to w and equating it to 0,

$$argmax_w = \frac{\delta}{\delta w}(\sum_{n=1}^{N} -y_n w^T x_n + log(\frac{1}{exp(-y_n w^T x_n) + 1}) + \lambda^T w\lambda) = 0$$

$$\sum_{n=1}^{N} -y_n x_n + \frac{y_n x_n}{1 + exp(y_n w^T x_n)} + 2\lambda w = 0$$

$$\sum_{n=1}^{N} y_n x_n (1 - \frac{1}{1 + exp(y_n w^T x_n)}) = 2\lambda w$$

$$\tilde{w} = \frac{1}{2\lambda} \sum_{n=1}^{N} y_n x_n (1 - \frac{1}{1 + exp(y_n w^T x_n)})$$

$$\tilde{w} = \frac{1}{2\lambda} \sum_{n=1}^{N} y_n x_n (1 - \frac{exp(y_n w^T x_n)}{1 + exp(y_n w^T x_n)})$$

$$\tilde{w} = \frac{1}{2\lambda} \sum_{n=1}^{N} y_n x_n \frac{1}{exp(y_n w^T x_n)}$$

So, we can say that $\tilde{w}$ can be expressed in the form of $\sum_{n=1}^{N} \alpha_n y_n x_n$

Here, $\tilde{w}$ is a function of w itself so, we do not have closed form solution of $\tilde{w}$.

$\alpha_n = \dfrac{1}{exp(y_n w^T x_n)}$.

$y_n w^T x_n$ is the signed distance from hyper-plane.

The value of $y_n w^T x_n$ is positive when points are correctly classified and negative when incorrectly classified.

For $y_n = 1$,$w^T x_n > 0 => y_n w^T x_n is positive$

For $y_n = -1$,$w^T x_n < 0 => y_n w^T x_n is also positive$

If $y_n w^T x_n$ is positive then $\frac{1}{exp(y_n w^T x_n)}$ is 0.

Therefore,$\alpha_n = 0$

When points are correctly classified w is not updated.

If $y_n w^T x_n$ is negative then $exp(-y_n w^T x_n)$ is some large value.

Therefore,$\alpha_n = some large value$

When points are incorrectly classified w is updated with $\frac{1}{2\lambda} \sum_{n=1}^{N} y_n x_n \alpha_n$.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 2

In generative classification model,

$P(y = k|x) = \dfrac{p(y = k) * p(x|y = k)}{p(x)}$

Given: $p(y = 1) = \Pi$ and $p(y = 0) = 1 - \Pi$

$p(x|y = k) = \Pi_{d=1}^{D} \mu_d^{x_d}(1 - \mu_d)^{1-x_d}$

$logp(x|y = k) = \sum_{d=1}^{D} x_d log(\mu_d) + (1 - x_d)log(1 - \mu_d))$

Now, plugin the values of $p(x|y = 1)$ and $p(x|y = 0)$

$logp(x|y = 1) = \sum_{d=1}^{D} x_d log(\mu_{d1}) + (1 - x_d)log(1 - \mu_{d1}))$

$logp(x|y = 0) = \sum_{d=1}^{D} x_d log(\mu_{d0}) + (1 - x_d)log(1 - \mu_{d0}))$

$logp(y = 1|x) = \sum_{d=1}^{D} x_d log(\mu_{d1}) + (1 - x_d)log(1 - \mu_{d1}) + log(\Pi)$

$logp(y = 1|x) = \sum_{d=1}^{D} x_d log(\mu_{d1}) + (1 - x_d)log(1 - \mu_{d1}) + log(\Pi)$

$logp(y = 1|x) = \sum_{d=1}^{D} x_d log(\mu_{d1}) - x_d log(1 - \mu_{d1}) + log(1 - \mu_{d1}) + log(\Pi)$

$logp(y = 1|x) = \sum_{d=1}^{D} x_d log(\dfrac{\mu_{d1}}{1 - \mu_{d1}}) + log(1 - \mu_{d1}) + log(\Pi)$

$logp(y = 0|x) = \sum_{d=1}^{D} x_d log(\mu_{d0}) + (1 - x_d)log(1 - \mu_{d0}) + log(1 - \Pi)$

$logp(y = 0|x) = \sum_{d=1}^{D} x_d log(\mu_{d0}) + (1 - x_d)log(1 - \mu_{d0}) + log(\Pi)$

$logp(y = 0|x) = \sum_{d=1}^{D} x_d log(\mu_{d0}) - x_d log(1 - \mu_{d0}) + log(1 - \mu_{d0}) + log(1 - \Pi)$

$logp(y = 0|x) = \sum_{d=1}^{D} x_d log(\dfrac{\mu_{d0}}{1 - \mu_{d0}}) + log(1 - \mu_{d0}) + log(1 - \Pi)$

At decision boundary,$p(y = 1|x) - p(y = 0|x) = 0$

Therefore, $logp(x|y=1) - logp(y=0|x) = 0$

$\sum_{d=1}^{D} x_d log(\frac{\mu_{d1}}{1-\mu_{d1}}) - x_d log(\frac{\mu_{d0}}{1-\mu_{d0}}) + log(1-\mu_{d1}) - log(1-\mu_{d0}) + log(\Pi) - log(1-\Pi) = 0$

$\sum_{d=1}^{D} log(\frac{\mu_{d1}(1-\mu_{d0})}{(1-\mu_{d1})\mu_{d0}})x_d + log(\frac{1-\mu_{d1}}{1-\mu_{d0}}) + log(\frac{\Pi}{1-\Pi}) = 0$

$\sum_{d=1}^{D} w_d x_d + b = 0$

Therefore, $w_d = log(\frac{\mu_{d1}(1-\mu_{d0})}{(1-\mu_{d1})\mu_{d0}})$ and $b = log(\frac{1-\mu_{d1}}{1-\mu_{d0}}) + log(\frac{\Pi}{1-\Pi})$.

The decision boundary learned by this model is Linear.

The equivalent discriminative model is:

$p(y=1|x,w) = p(y=0|x,w)$

$\frac{exp(w^T x)}{1+exp(w^T x)} = \frac{1}{1+exp(w^T x)}$

$exp(w^T x) = 1$

Taking log on both sides,

$w^T x = 0$

$\sum_{d=1}^{D} w_d x_d + b = 0$

$$p(y=1|x) = \frac{p(y=1) * p(x|y=1)}{p(x)}$$

$$p(y=1|x) = \frac{p(y=1) * p(x|y=1)}{p(y=1) * p(x|y=1) + p(y=0) * p(x|y=0)}$$

$$p(y=1|x) = \frac{\Pi * \sum_{d=1}^{D} x_d log(\mu_{d1}) + (1-x_d) log(1-\mu_{d1})}{\Pi * \sum_{d=1}^{D} x_d log(\mu_{d1}) + (1-x_d) log(1-\mu_{d1}) + (1-\Pi) * \sum_{d=1}^{D} x_d log(\mu_{d0}) + (1-x_d) log(1-\mu_{d0})}$$

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 3

Given: Least square Regression with constraint $||w|| <= c$ to some $c > 0$.

Now, the original problem is
$\tilde{w} = argmin_w(L(w) + argmax_{\alpha>=0}\alpha g(w))$

$\tilde{w}_P = argmin_w(argmax_{\alpha>=0}(L(w) + \alpha g(w))$

The above equation is the primal problem. The dual of the problem is:
$\tilde{w}_D = argmax_{\alpha>=0}(argmin_w(L(w) + \alpha g(w))$

Here $L(w) = \sum_{n=1}^{N}(y_n - w^T x_n)^2$ and $g(w) = w^T w - c^2$

The dual problem is:
$\tilde{w}_D = argmax_{\alpha>=0}(argmin_w(\sum_{n=1}^{N}(y_n - w^T x_n)^2 + \alpha(w^T w - c^2))$

$\tilde{w} = argmax_{\alpha>=0}(argmin_w((Y - XW)^T(Y - XW) + \alpha(w^T w - c^2))$

To derive the expression of $\tilde{w}$ we differentiate the above equation with respect to w and equating it to 0.

$$\frac{\delta}{\delta w}((Y - Xw)^T(Y - Xw) + \alpha(w^T w - c^2)) = 0$$

Using the identity $(A - B)^T = (A^T - B^T) and (AB)^T = B^T A^T$

$$\frac{\delta}{\delta w}((Y^T - w^T X^T)(Y - Xw) + \alpha(w^T w - c^2)) = 0$$

$$\frac{\delta}{\delta w}((Y^T Y - Y^T Xw - w^T X^T Y + w^T X^T Xw) + \alpha(w^T w - c^2)) = 0$$

$$0 - 2X^T Y + 2X^T Xw + 2\alpha Iw = 0$$

$$\tilde{w} = (X^T X + \alpha I)^{-1} X^T Y$$

The above expression is same as L2 regularized least square expression.α, the Lagrangian multiplier is the regularization hyper parameter.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 4

$$p(y_n = k|x_n, W) = \frac{exp(W_k^T x_n)}{\sum_{l=1} k exp(w_l^T x_n)} = \mu_n^k$$

Each likelihood is a multinoulli distribution. Therfore,

$p(y|X, W) = \Pi_{n=1}^{N} \Pi_{l=1}^{k} \mu_{nl}^{y_{nl}}$

$L(w) = log \Pi_{n=1}^{N} \Pi_{l=1}^{k} \mu_{nl}^{y_{nl}}$

$L(w) = \sum_{n=1}^{N} \sum_{l=1}^{k} y_{nl} log \mu_{nl}$

$L(w) = \sum_{n=1}^{N} [(\sum_{l=1}^{k} y_{nl} w_l^T x_n) - log(\sum_{l=1}^{k} exp(w_l^T x_n))]$

The above equation is the log likelihood. The negative log likelihood(NLL(w) is $-L(w)$.

Taking derivative of NLL(w) with respect to each w_k we get the gradient corresponding to each weights of W,

$g_k = \sum_{n=1}^{N} \frac{exp(w_k^T x_n)}{\sum_{l=1} k exp(w_l^T x_n)} x_n - y_{nk} x_n$
$g_k = \sum_{n=1}^{N} (\mu_{nk} - y_{nk}) x_n$

Gradient descent algorithm to update each w_k: where $\eta = 1$

Step 1: Initialize W as $W^{(0)}$

Step 2: for $l = 1$ to k:

$w_l^{t+1} = w^{(t)} - g_l$

$w_l^{t+1} = w^{(t)} - \sum_{n=1}^{N} (\mu_{nl} - y_{nl}) x_n$

Step 3: Repeat until Converge

Stochastic Gradient descent algorithm to update each w_k: where $\eta = 1$

Step 1: Initialize W as $W^{(0)}$

Step 2: for $l = 1$ to k:

Pick a random $i \epsilon 1, 2....., N$. Update w_l as follows:

$w_l^{t+1} = w^{(t)} - g_{il}$

$w_l^{t+1} = w^{(t)} - (\mu_{il} - y_{il})x_n$

Step 3: Repeat until Converge

When this model uses hard class probabilities instead of soft class probabilities the model will be trained similar to a perceptron algorithm.

Stochastic Gradient descent algorithm to update each w_k where $\eta = 1$ in perceptron manner:

Step 1: Initialize W as $W^{(0)}$

Step 2: for $l = 1$ to k:

Pick a random $i \epsilon 1, 2....., N$. Update w_l as follows:

Step 3: for $l = 1$ to k:

if l is not $argmax_l \{\mu_{il}\}$

$w_l^{t+1} = w^{(t)} + y_{il}x_n$

Step 4: If not converged go to step 2

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 5

Given: Two sets of datapoints, $\{x_1, x_2, ...x_n\}$ and $\{y_1, , y_2, ..., y_n\}$.

The convex hull is defined as:

$x = \sum_n \alpha_n x_n$, where $\alpha_n >= 0$ and $\sum_n \alpha_n = 1$.

The two sets are linearly separable if $w^T x_n + w_0 > 0$ for all x and $w^T y_n + w_0 < 0$ for all y.

Therefore,$g(x)$ is the linear discriminant function for x and $g(y)$ is the linear discriminant function for y.

$g(x) = w^T x_n + w_0$

$g(x) = w^T \sum_n \alpha_n x_n + w_0$

$g(x) = w^T \sum_n \alpha_n x_n + w_0$

$g(x) = \sum_n \alpha_n (w^T x_n + w_0)$

$g(y) = w^T y_n + w_0$

$g(y) = w^T \sum_m \beta_m y_n + w_0$

$g(y) = w^T \sum_m \beta_m y_n + w_0$

$g(y) = \sum_m \beta_m (w^T y_n + w_0)$

where $\beta_m >= 0$ and $\sum_m \beta_m = 1$

Linear Separability is satisfied if $g(x) > 0$ and $g(y) < 0$ If there exists a point xz which lies at the intersection of two convex hulls x and y,

$g(xz) = \sum_n \alpha_n (w^T xz + w_0) = \sum_m \beta_m (w^T xz + w_0)$.

It is not possible for the linear discriminant $g(xz)$ to be both greater than 0 and less than 0 for the same data point.

Therefore, we can say if two convex hulls intersects then there do not exists any linearly separable hyper plane.

Also, it is not possible for the point xz to be both $w^T xz + w_0 > 0$ and $w^T xz + w_0 < 0$.

Therefore, we can say that if two convex hulls are linearly separable then two convex hulls do not intersect.

Hence,for the set of two convex hulls, are linearly separable if and only if the convex hulls do not intersect.(proved)

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 6

The general convention of writing effective hyper-plane in SVM is $y_n(w^T x_n + b) >= 1$.

Instead we are writing the effective hyper-plane in SVM as: $y_n(w^T x_n + b) >= m$.

Therefore,the equation of margin maximizing hyper planes are:

Π :Margin Maximizing hyper-plane

$\Pi_+ = w^T x + b = +m$

$\Pi_- = w^T x + b = -m$

Margin=$\dfrac{2m}{||w||}$

The dual problem of Hard-Margin SVM is:

$max_{\alpha>=0} min_{w,b} L(w,b,\alpha) = \dfrac{w^T w}{2m} + \sum_{n=1}^{N} \alpha_n (m - y_n(w^T x_n + b))$

Taking Derivative of $L(w,b,\alpha)$ with respect to w and equating it to 0:

$\dfrac{w}{m} - \sum_{n=1}^{N} \alpha_n y_n x_n = 0$

$w = m * \sum_{n=1}^{N} \alpha_n y_n x_n$

Taking $+mand-m$ instead of $+1and-1$ does not change the affecting hyper plane because the hyper-plane w is still in the form of $\sum_{n=1}^{N} \alpha_n y_n x_n$ and m is a constant.Since, m is a constant it does not affect in any optimization standpoint.

Another reason is:

$W^T x + b = m$

$(w/k)^T x + (b/m) = 1$

$w'^T x + b' = 1$

Any value of m is possible because w must be perpendicular to Π and $||w||$ need not be 1.

Therefore, changing the hyper-plane from $y_n(w^T x_n + b) >= 1$ to $y_n(w^T x_n + b) >= m$ does not change the effective hyper-plane.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Aneet Kumar Dutta
Roll Number: 18111401
Date: September 30, 2018

QUESTION 7

The first 3 plots horizontally the resulting hyper planes using the data file 'binclass.txt' and the next 3 corrsponds to data file 'binclassv2.txt'.

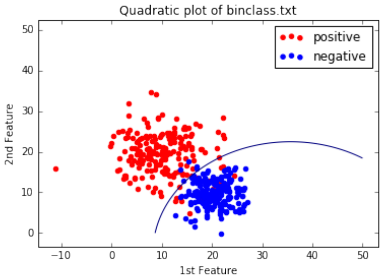

Figure 1: Quadratic Analysis

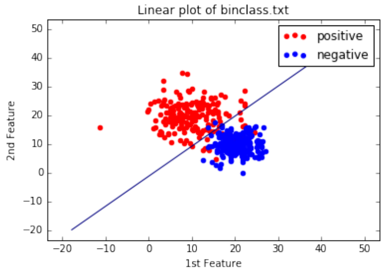

Figure 2: Linear Analysis

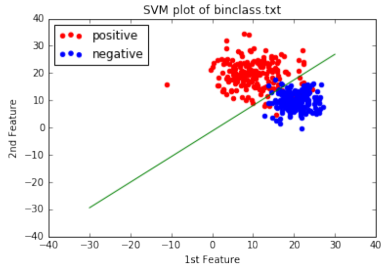

Figure 3: SVM

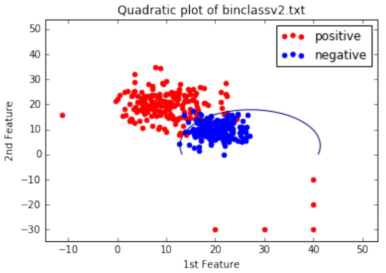

Figure 4: Quadratic Analysis

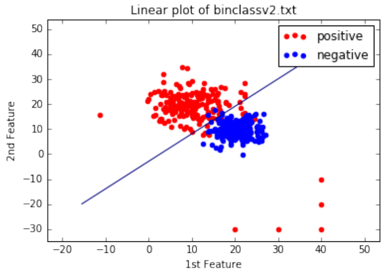

Figure 5: Linear Analysis

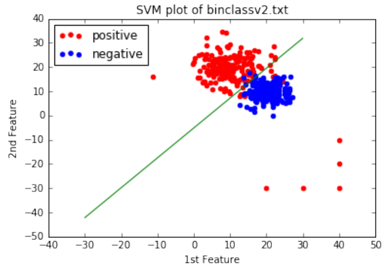

Figure 6: SVM

In both the dataset Quadratic generative classification is better than linear generative classification and linear kernel SVM.