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

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 30, 2018

QUESTION 1

Note that likelihood is given by:

$$P(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(y_n|\mathbf{x}_n, \mathbf{w})$$

MAP objective function can be written as combination of NLL and log of prior as follows:

$$\begin{aligned}\mathcal{L}(w) &= -\sum_{n=1}^{N} P(y_n|\mathbf{x}_n, \mathbf{w}) - log(P(w)) \\ &= \sum_{n=1}^{N} -y_n \mathbf{w}^T \mathbf{x}_n + log(1 + exp(y_n \mathbf{w}^T \mathbf{x}_n)) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}\end{aligned}$$

Now differentiating with respect to w:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \sum_{n=1}^{N} \frac{-y_n \mathbf{x}_n}{1 + exp(y_n \mathbf{w}^T \mathbf{x}_n)} + \lambda \mathbf{w}$$

Putting gradien equals to zero we get:

$$\hat{\mathbf{w}} = \sum_{n=1}^{N} \frac{1}{\lambda(1 + exp(y_n \mathbf{w}^T \mathbf{x}_n))} y_n \mathbf{x}_n$$

Clearly $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$ where:

$$\alpha_n = \frac{1}{\lambda(1 + exp(y_n \mathbf{w}^T \mathbf{x}_n))} = \frac{(1 - P(y_n|\mathbf{x}_n, \mathbf{w}))}{\lambda}$$

Note that the form of the solution $\mathbf{w}$ is simply a weighted sum of all the training inputs $\mathbf{x}_1, ..., \mathbf{x}_N$. Since P is the logistic regression model so whenever ther will be correct prediction α_n 0 because for correct prediction $P(y_n|\mathbf{x}_n, \mathbf{w}) = 1$and whenever there is incorrect prediction α_n $\frac{1}{\lambda}$.
Therefore, $\mathbf{w}$ gets updated whenvere there is wrong prediction which looks similar to the mistake driven model (Perceptron algorithm).

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

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 30, 2018

QUESTION 2

Note that:

$$P(y_n = k|\mathbf{x}) = \frac{P(\mathbf{x}|y_n = k)P(y_n = k)}{\sum_{k=0}^{1} P(\mathbf{x}|y_n = k)P(y_n = k)}$$

So,

$$P(y = 1|\mathbf{x}) = \frac{\pi \prod_{d=1}^{D} \mu_{d1}^{x_d}(1-\mu_{d1})^{(1-x_d)}}{\pi \prod_{d=1}^{D} \mu_{d1}^{x_d}(1-\mu_{d1})^{(1-x_d)} + (1-\pi)\prod_{d=1}^{D} \mu_{d0}^{x_d}(1-\mu_{d0})^{(1-x_d)}}$$

It is clearly in discriminative form classifier.
Now at the boundary probability of both the classes will be same i.e. $P(y = 1|\mathbf{x}) = P(y = 0|\mathbf{x})$. If we equate both the terms then the denominator term will cancel and only numerator term will remain. We can also write $\log P(y = 1|\mathbf{x}) = \log P(y = 0|\mathbf{x})$

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

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

The decision boundary is clearly linear as it can be written in the form of $\mathbf{w}^T\mathbf{x} + b = 0$ where dth index weight is given by $w_d = \log(\frac{\mu_{d1}(1-\mu_{d0})}{\mu_{d0}(1-\mu_{d1})})$ and the bias term is given by $b = \log(\frac{\pi}{1-\pi}) + \sum_{d=1}^{D} \log(\frac{(1-\mu_{d1})}{(1-\mu_{d0})})$.

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

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 30, 2018

QUESTION 3

We have to show that solving problem:

$$\hat{\mathbf{w}} = argmin_w \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2 s.t. \|\mathbf{w}\| \leq c$$

is equivalent to solving some l_2 regularized least squares linear regression model that will give the exact same solution.
We will write this constrained problem in terms of lagrangian.

$$\hat{\mathbf{w}} = \mathcal{L}(\mathbf{w}, \lambda) = argmin_w max_{\lambda \geq 0} \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda(\mathbf{w}^T \mathbf{w} - c^2)$$

Since it is the convex problem we will solve the dual version of this problem.

$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$ give us the solution

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

Also, $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$ implies $\mathbf{w}^T \mathbf{w} - c^2 = 0 \Rightarrow \|\mathbf{w}\| = c$.
Now, using norm value in equation 1 we get c as a function of λ. Now will take the suitable value of λ say λ_0 such that $f(\lambda_0) = c$.

$$\hat{\mathbf{w}} = \mathcal{L}(\mathbf{w}, \lambda) = argmin_w max_{\lambda \geq 0} \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda_0 \mathbf{w}^T \mathbf{w} - -\lambda_0 c^2$$

Note that $-\lambda_0 c^2$ term will not affect the minimum argument value. So the solution for both the equation will be same.

$$P(y_n|\mathbf{x}_n, \mathbf{W}) = \prod_{k=1}^{K} \mu_{nk}^{y_{nk}}$$

where y_{nk} will be equal to 1 if $y_n = k$ and 0 otherwise $\forall k = 1, 2, \ldots K$.
Likelihood Function:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{W}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mu_{nk}^{y_{nk}}$$

Now define the negative log likelihood function as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \log \mu_{nk} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}(w_k^T \mathbf{x}_n - \log(\sum_{k=1}^{K} exp(w_k^T \mathbf{x}_n)))\end{aligned}$$

Now differentiating with respect to $\mathbf{w}_l$ we get:

$$\mathbf{g}_k = \frac{\partial \mathcal{L}(\mathbf{w})}{\mathbf{w}_l} = \sum_{n=1}^{N} \left(-y_{nl} + (\sum_{k=1}^{K} y_{nk}) \frac{exp(w_l^T \mathbf{x}_n)}{\sum_{k=1}^{K} exp(w_k^T \mathbf{x}_n)} \right) \mathbf{x}_n = \sum_{n=1}^{N} \left(-y_{nl} + (\sum_{k=1}^{K} y_{nk}) \mu_{nl} \right) \mathbf{x}_n$$

Now for MLE estimate $\mathbf{g}_k = 0 \forall k = 1 \ldots K$, we can clearly see that we cann't get closed form solution for $\mathbf{W}$. So we have to use gradient descent approach to update the weight vector. The general form of **G.D.** is $\mathbf{w}^{new} = \mathbf{w}^{old} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$. For our case $\eta = 1$.

$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \mathbf{g}_k, \forall k = 1 \ldots K$$

In stochastic gradient descent we randomly choose only one point say $(\mathbf{x}_n, \mathbf{y}_n)$ and consider gradient only for that point and update weights accordingly. Define gradient for the nth point as follows:

$$\mathbf{g}_k = \left(-y_{nl} + (\sum_{k=1}^{K} y_{nk}) \mu_{nl} \right) \mathbf{x}_n$$

Stochastic gradient descent update:

$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \mathbf{g}_{nk}, \forall k = 1 \ldots K$$

PART 2: Replacing soft class probabilites with hard class assignments
In this case $\mu_{nk} = 1$ for $k = argmax_l \{\mu_{nl}\}_{l=1}^{K}$ and $\mu_{nk'} = 0, \forall k' \neq k$ For this case there are two different cases when there is correct prediction and when there is incorrect prediction.

Correct prediction:
Actual output $y_n = K_0$ and prediction is also K_0, in this gradient will be equal to zero. Therefore, $\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)}$ for all k. This also makes intuitive sense as we are not updating weights when the prediction is correct.
Incorrect Prediction:
Actual output $y_n = L_0$ and prediction is also K_0 where $K_0 \neq L_0$. In this case we get the following result:
If $k \neq K_0 and k \neq L_0$ then $\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)}$
If $k = K_0 and k \neq L_0$ then $\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \mathbf{x}_n$
If $k \neq K_0 and k = L_0$ then $\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} + \mathbf{x}_n$. This also

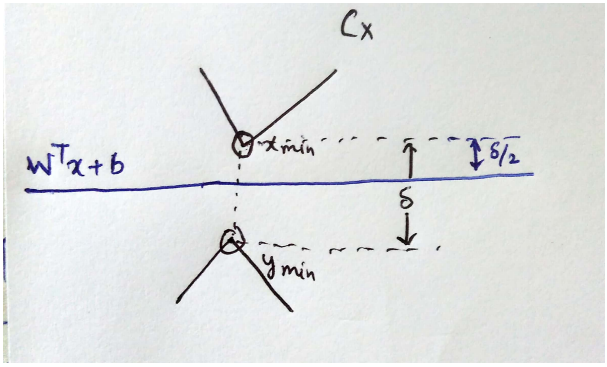
Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 30, 2018

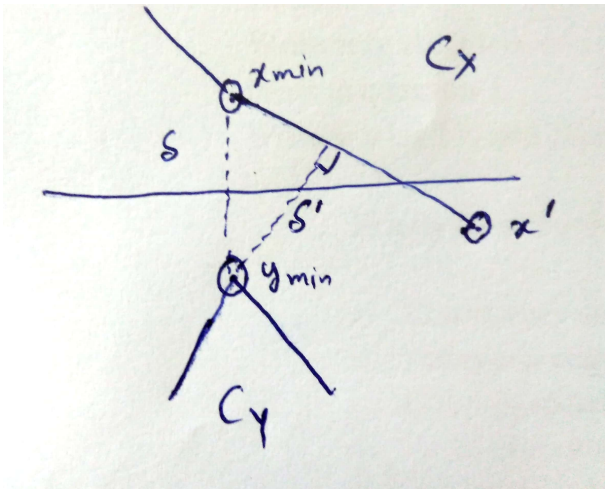
QUESTION 5

Let, $\mathbf{C}_x, \mathbf{C}_y$ denote the set of all points enclosed by the both the convex hulls.
Claim: $\mathbf{C}_x, \mathbf{C}_y$ are linearly separable if and only if both the convex hulls don't intersect with each other.
Proof:
($\Rightarrow$)
$\mathbf{C}_x$ and $\mathbf{C}_y$ are linearly separable implies that there exists a hyperplane of the form $\mathbf{w}^T\mathbf{x} + \mathbf{b}$ which separates both the convex hull.
i.e. $\forall, \mathbf{x}' \in \mathbf{C}_x, \mathbf{y}' \in \mathbf{C}_y$,
$(\mathbf{w}^T\mathbf{x}' + \mathbf{b})(\mathbf{w}^T\mathbf{y}' + \mathbf{b}) < 0$ $----(1)$
Now suppose convex hulls do intersect, which means $\exists \mathbf{z} \in \mathbf{C}_x, \mathbf{z} \in \mathbf{C}_y$ but by property (1), $(\mathbf{w}^T\mathbf{z} + \mathbf{b})^2 < 0$ which is clearly a contradiction.

($\Leftarrow$)
We will try to prove it by construction. Assume convex hull don't intersect then we have to prove that both of them are linearly separable.
Let $\delta = min_{\mathbf{x}\in\mathbf{C}_x, \mathbf{y}\in\mathbf{C}_y}\{dist(\mathbf{x}, \mathbf{y})\}$ that is δ is the minimum distance between points in convex hulls. Note $\delta > 0$ because if not then $\delta = 0$ which implies $dist(x_{min}, y_{min}) = 0$, which in turn signify that $x_{min} = y_{min}$ but which is a contradiction to our assumption that convex hulls don't intersect.



Now consider a plane $\mathbf{w}^T\mathbf{x} + \mathbf{b}$ which is a perpendicular bisector of the line joining the points $(\mathbf{x}_{min}, \mathbf{y}_{min})$. Our claim is that this hyperplane will linearly separate both the hulls. I will try to prove this by contradiction. Suppose the hyperplane is not separable which means without loss of generality $\mathbf{x}' \in \mathbf{C}_x, \mathbf{y}_{min}$ belong to the same side. (see figure on next page). Now consider the line joining between $\mathbf{x}', \mathbf{x}_{min}$ which will be the part of $\mathbf{C}_x$. Then consider the perpendicular distance from $\mathbf{y}_{min}$ call it δ'. Clearly, $\delta' < \delta$ which is clearly the contradiction to our initial assumption that δ is the minimum distance between points in convex hulls.


xmin
Cx
s
s'
x'
ymin
Cy

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

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 30, 2018

QUESTION 6

Note, that our aim is to maximize the margin between two classes subject to the condition $y_n(\mathbf{w}^T\mathbf{x}_n + \mathbf{b}) \geq 1$. We want to show that if we change the condition to $y_n(\mathbf{w}^T\mathbf{x}_n + \mathbf{b}) \geq m$ for some $m \in \mathbb{R}$. It will suffice to show that direction of $\mathbf{w}$ will not be changed it will just scale up by some constant.
Both of the problems will be constrained optimization problem i.e. maximizing margin with some constraints on it, so we will use lagrangian method to convert it into unconstrained optimization problem with $\alpha_n, \beta_n \forall n = 1 \dots N$ as the lagrange multipliers for both of the problems respectively. The two optimization problems are as follows:

$$max_{\alpha \geq 0} min_{\mathbf{w},\mathbf{b}} \mathcal{L}(\mathbf{w}, \mathbf{b}, \alpha) = \frac{\mathbf{w}^T\mathbf{w}}{2} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\mathbf{w}^T\mathbf{x}_n + \mathbf{b})) \tag{2}$$

$$max_{\beta \geq 0} min_{\mathbf{w},\mathbf{b}} \mathcal{L}'(\mathbf{w}, \mathbf{b}, \beta) = \frac{\mathbf{w}^T\mathbf{w}}{2} + \sum_{n=1}^{N} \beta_n (m - y_n(\mathbf{w}^T\mathbf{x}_n + \mathbf{b})) \tag{3}$$

Now differentiating $\mathcal{L}$ and $\mathcal{L}'$ w.r.t $\mathbf{w}, \mathbf{b}$ and setting them to zero will give the following equations.

$$\mathbf{w}_1 = \sum_{n=1}^{N} \alpha_n y_n x_n \tag{4}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{5}$$

$$\mathbf{w}_2 = \sum_{n=1}^{N} \beta_n y_n x_n \tag{6}$$

$$\sum_{n=1}^{N} \beta_n y_n = 0 \tag{7}$$

Now substituting the value of $\mathbf{w}_1, \mathbf{w}_2$ in equation $(1), (2)$ respectively we get the dual problem as follows:

$$max_{\alpha \geq 0} \mathcal{L}_D(\alpha) = \alpha^T\mathbf{1} - \alpha^T\mathbf{G}\alpha \tag{8}$$

$$max_{\beta \geq 0} \mathcal{L}'_D(\beta) = m\beta^T\mathbf{1} - \beta^T\mathbf{G}\beta \tag{9}$$

where $\mathbf{1}$ is the vector of 1s, $\mathbf{G}$ is the $N \times N$ matrix with $G_{mn} = y_m y_n \mathbf{x}_m^T \mathbf{x}_n^T$, $\alpha = [\alpha_1, \dots \alpha_N], \beta = [\beta_1, \dots, \beta_N]$. Let $\hat{\alpha}, \hat{\beta}$ be the two optimal solution of the above two equations. Note that we can also write equation 9 as follows:

$$max_{\beta \geq 0} \mathcal{L}'_D(\beta) = m^2 \left(\left(\frac{\beta}{m}\right)^T \mathbf{1} - \left(\frac{\beta}{m}\right)^T \mathbf{G} \left(\frac{\beta}{m}\right) \right) \tag{10}$$

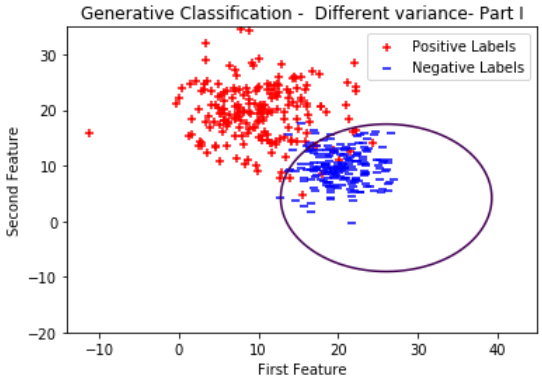
Clearly the form of above equation is infact similar to equation 7 which gives $\frac{\hat{\beta}}{m} = \hat{\alpha}$, which gives $\mathbf{w}_2 = m\mathbf{w}_1$. It is clear that solution has just been scaled by a constant, without changing the direction.

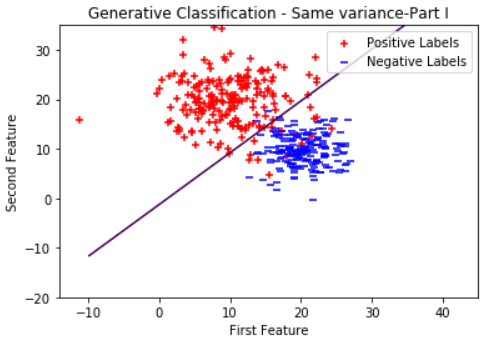
Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 2

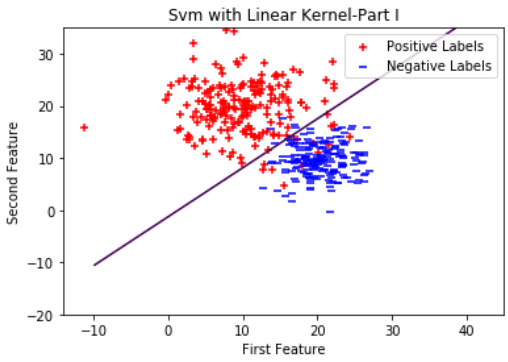
Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 30, 2018

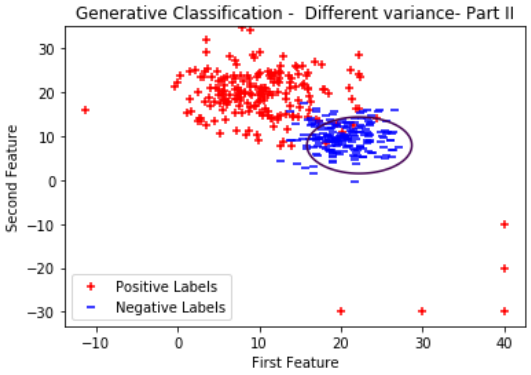
QUESTION 7

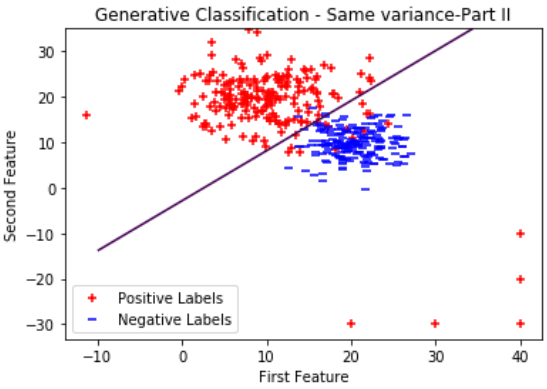
The following decision boundaries were obtained for generative classifier and support vector machine (with linear kernel). I used scikit-learn's svm module with linear kernel for implementation of Support vector Machine.
Second dataset contained a lot of outliers compared to the first one. Generative classifier with different variances for both the positive and negative classes **gave tighter quadratic (sort of elliptical) decision boundary dataset with a lot of outliers**. In case of linear decision boundary (generative classification with same variances and support vector machine) quite identical decision boundaries were obtained for both the datasets.

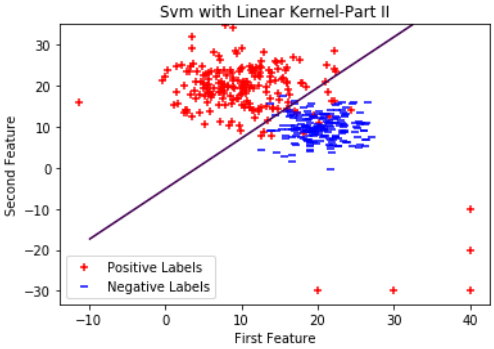







Generative Classification - Different variance- Part II
Second Feature
First Feature
Positive Labels
Negative Labels


Generative Classification - Same variance-Part II
Second Feature
First Feature
Positive Labels
Negative Labels


Svm with Linear Kernel-Part II
Second Feature
First Feature
Positive Labels
Negative Labels