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

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION 1

In logistic regression model,

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

And the loss function for MAP is

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

Taking derivative of $\mathcal{L}(\mathbf{w})$ w.r.t $\mathbf{w}$ and setting to 0.

$$-\sum_{n=1}^{N} \frac{\partial}{\partial \mathbf{w}} log(p(y_n|\mathbf{x}_n, \mathbf{w})) + \frac{\lambda}{2}\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^T\mathbf{w} = 0$$

Plugging the value of $p(y_n|\mathbf{x}_n, \mathbf{w})$

$$\sum_{n=1}^{N} \frac{\partial}{\partial \mathbf{w}} log(\frac{1}{1 + exp(-y_n \mathbf{w}^T \mathbf{x}_n)}) = \frac{\lambda}{2}\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^T\mathbf{w}$$

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

Using $\frac{\partial(\mathbf{w^T x})}{\partial \mathbf{w}} = \mathbf{x}$ and $\frac{\partial(\mathbf{w^T K w})}{\partial \mathbf{w}} = (\mathbf{K} + \mathbf{K^T})\mathbf{w}$

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

$\implies \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$ with $\alpha_n = \frac{1}{\lambda}\frac{exp(-y_n \mathbf{w}^T \mathbf{x}_n)}{1+exp(-y_n \mathbf{w}^T \mathbf{x}_n)} = \frac{1-\mu_n}{\lambda}$
More the value of μ_n, more is the probability of y=1 for that point. So if a point lies close to the boundary or on the opposite side, value of (1-μ_n) will be large and hence it will contribute more to the weights and simlarly for points which are correctly classified and far from boundary, it will be small and hence acts as regulariser.
Since α_n assigns weightage to different points, $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$ makes sense as w is being learned based on points close to the boundary more as compared to points far from boundary and its expression is similar to any other linear classifier.

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

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION 2

In generative classification model for binary classification,

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

Here, $p(y=1) = \pi$ and $p(\mathbf{x}|y=1) = \prod_{d=1}^{D} \mu_{d,1}^{x_d}(1-\mu_{d,1})^{1-x_d}$

$$p(y=1|\mathbf{x}) = \frac{\prod_{d=1}^{D} \mu_{d,1}^{x_d}(1-\mu_{d,1})^{1-x_d} \cdot \pi}{\prod_{d=1}^{D} \mu_{d,1}^{x_d}(1-\mu_{d,1})^{1-x_d} \cdot \pi + \prod_{d=1}^{D} \mu_{d,0}^{x_d}(1-\mu_{d,0})^{1-x_d} \cdot (1-\pi)}$$

$$\implies p(y=1|\mathbf{x}) = \frac{1}{1 + \frac{\prod_{d=1}^{D} \mu_{d,0}^{x_d}(1-\mu_{d,0})^{1-x_d} \cdot (1-\pi)}{\prod_{d=1}^{D} \mu_{d,1}^{x_d}(1-\mu_{d,1})^{1-x_d} \cdot \pi}}$$

$$\implies p(y=1|\mathbf{x}) = \frac{1}{1 + exp(log(\frac{\prod_{d=1}^{D} \mu_{d,0}^{x_d}(1-\mu_{d,0})^{1-x_d} \cdot (1-\pi)}{\prod_{d=1}^{D} \mu_{d,1}^{x_d}(1-\mu_{d,1})^{1-x_d} \cdot \pi}))}$$

Now calculating value of the exponential term, Let

$$K = log(\frac{\prod_{d=1}^{D} \mu_{d,0}^{x_d}(1-\mu_{d,0})^{1-x_d} \cdot (1-\pi)}{\prod_{d=1}^{D} \mu_{d,1}^{x_d}(1-\mu_{d,1})^{1-x_d} \cdot \pi})$$

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

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

$$\implies K = \sum_{d=1}^{D} x_d w_d + k = \mathbf{w}^T\mathbf{x} + k$$

where,

$$w_d = log(\frac{\mu_{d,0}(1-\mu_{d,1})}{\mu_{d,1}(1-\mu_{d,0})}) \ , \ k = log(\frac{1-\pi}{\pi}) + \sum_{d=1}^{D} log(\frac{1-\mu_{d,0}}{1-\mu_{d,1}})$$

Hence,

$$p(y=1|\mathbf{x}) = \frac{1}{1 + exp(\mathbf{w}^T\mathbf{x} + k)}$$

which is equivalent to a probabilistic discriminative classifier.
For decision boundary, $p(y=1|\mathbf{x}) = p(y=0|\mathbf{x}) = 1 - p(y=1|\mathbf{x})$
$\implies \mathbf{w}^T\mathbf{x} + k = c$, which is a **linear** decision boundary.

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

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION 3

Given:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} \left(y_n - \mathbf{w^T x_n}\right)^2 \; s.t. \; ||\mathbf{w}|| \leq c$$

Or, the condition can be modified as $||\mathbf{w}||^2 \leq c^2$
Using Lagrangian,

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} \left(y_n - \mathbf{w^T x_n}\right)^2 + \arg\max_{\alpha \geq 0} \alpha(||\mathbf{w}||^2 - c^2)$$

In Dual form,

$$\hat{\mathbf{w}} = \arg\max_{\alpha \geq 0} \; \arg\min_{\mathbf{w}} \sum_{n=1}^{N} \left(y_n - \mathbf{w^T x_n}\right)^2 + \alpha(||\mathbf{w}||^2 - c^2)$$

First differentiating w.r.t $\mathbf{w}$,

$$\frac{\partial}{\partial \mathbf{w}} (\sum_{n=1}^{N} \left(y_n - \mathbf{w^T x_n}\right)^2 + \alpha(||\mathbf{w}||^2 - c^2)) = 0$$

Using $\frac{\partial(\mathbf{w^T x})}{\partial \mathbf{w}} = \mathbf{x}$ and $\frac{\partial(\mathbf{w^T K w})}{\partial \mathbf{w}} = (\mathbf{K} + \mathbf{K^T})\mathbf{w}$ and $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$

$$\implies -2 \sum_{n=1}^{N} \mathbf{x_n}(y_n - \mathbf{x_n^T w}) + 2\alpha \mathbf{w} = 0$$

$$\implies \mathbf{w} = (\alpha \mathbf{I} + \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T)^{-1} (\sum_{n=1}^{N} y_n \mathbf{x}_n)$$

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

Plugging this expression of $\mathbf{w}$ in Lagrangian, we get

$$\hat{\alpha} = \arg\max_{\alpha \geq 0} \mathcal{L}(\alpha)$$

Substituting this $\hat{\alpha}$ in $\mathbf{w}$,

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

Which is similar to the solution of l_2 regularized least squares linear regression model with hyperparameter $= \hat{\alpha}$

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

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION 4

In Softmax,

$$p(y_n = k|\mathbf{x}_n\mathbf{W}) = \frac{exp(\mathbf{w}_k^T\mathbf{x}_n)}{\sum_{l=1}^{K} exp(\mathbf{w}_l^T\mathbf{x}_n)} = \mu_{nk} \quad and \quad \sum_{l=1}^{K} \mu_{nl} = 1$$

And the likelihood is given by

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

where $y_{nl} = 1$ if true class of example n is l, and 0 otherwise.
Finding Negative-Log-Likelihood,

$$NLL(\mathbf{W}) = -\sum_{n=1}^{N}\sum_{l=1}^{K} y_{nl} log(\mu_{nl})$$

$$\implies NLL(\mathbf{W}) = \sum_{n=1}^{N}\sum_{l=1}^{K} y_{nl}(log(\sum_{t=1}^{K} exp(\mathbf{w}_t^T\mathbf{x}_n)) - \mathbf{w}_l^T\mathbf{x}_n)$$

To obtain MLE solution for $\mathbf{w}_i$ differentiate NLL($\mathbf{W}$) w.r.t $\mathbf{w}_i$ and set it to 0.

$$\frac{\partial}{\partial \mathbf{w}_i} NLL(\mathbf{W}) = 0$$

$$\implies \sum_{n=1}^{N}(\sum_{l=1}^{K} y_{nl} \frac{exp(\mathbf{w}_i^T\mathbf{x}_n)}{\sum_{t=1}^{K} exp(\mathbf{w}_t^T\mathbf{x}_n)}\mathbf{x}_n - y_{ni}\mathbf{x}_n) = 0$$

$$\implies \sum_{n=1}^{N}(\frac{exp(\mathbf{w}_l^T\mathbf{x}_n)}{\sum_{t=1}^{K} exp(\mathbf{w}_t^T\mathbf{x}_n)}\mathbf{x}_n - y_{ni}\mathbf{x}_n) = 0$$

$$\implies \sum_{n=1}^{N}(\mu_{ni} - y_{ni})\mathbf{x}_n = 0$$

which does not give closed form solution.
Now gradient for any i is

$$g_i = \frac{\partial}{\partial \mathbf{w}_i} NLL(\mathbf{W}) = \sum_{n=1}^{N}(\mu_{ni} - y_{ni})\mathbf{x}_n$$

So the Gradient Descent Update Rule will be

$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \eta g_k^{(t)}$$

$$\implies \mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \eta \sum_{n=1}^{N} (\mu_{nk}^{(t)} - y_{nk}) \mathbf{x}_n$$

$$\implies \mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \sum_{n=1}^{N} (\mu_{nk}^{(t)} - y_{nk}) \mathbf{x}_n$$

For Stochastic Gradient Descent, Update Rule will be

$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - (\mu_{nk}^{(t)} - y_{nk}) \mathbf{x}_n$$

Overall Sketch:

1. Initialize $\mathbf{w}_k$ as $\mathbf{w}_k^{(0)}$ for all k
2. Pick a random n $\in \{1, 2, ..., N\}$.
   Update for all $\mathbf{w}_k$ as follows
   $$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - (\mu_{nk}^{(t)} - y_{nk}) \mathbf{x}_n$$
3. Repeat until convergence

Special Case (Soft assignments replaced by hard assignments):
$\mu_{nk} = 1 \; for \; k = \arg\max_l \{\mu_{nl}\}_{l=1}^{K}$ and $\mu_{nk'} = 0$ otherwise.

The Update rule will be

$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - g_k^{(t)}$$

where the value of $g_k^{(t)}$ will be

$$g_k^{(t)} = \begin{cases} 0 & \mu_{nk}^{(t)} = y_{nk} \\ \mathbf{x}_n & \mu_{nk}^{(t)} = 1, y_{nk} = 0 \\ -\mathbf{x}_n & \mu_{nk}^{(t)} = 0, y_{nk} = 1 \end{cases}$$

Overall Sketch:

1. Initialize $\mathbf{w}_k$ as $\mathbf{w}_k^{(0)}$ for all k
2. Pick a random n $\in \{1, 2, ..., N\}$.
   Update for all $\mathbf{w}_k$ as follows
   $$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - g_k^{(t)}$$
   where value of $g_k^{(t)}$ is given above
3. Repeat until convergence

Updates in this case are case dependent, i.e. they depend on the values of $\mu_{nk}^{(t)}$ *and* y_{nk}, and when they are equal, $\mathbf{W}$ is not updated, whereas in previous case, $\mathbf{W}$ is updated each time.

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

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION 5

To show : the set of $\mathbf{x}'s$ and the set of $\mathbf{y}'s$ are linearly separable if and only if their convex hulls do not intersect.
Part 1: If $\mathbf{X}$ and $\mathbf{Y}$ are linearly separable then their convex hulls do not intersect.
Applying the defintion of linear separability,

$$\forall \mathbf{x}_i \in \mathbf{X},\ \mathbf{w}^T\mathbf{x}_i + b > 0$$

$$\forall \mathbf{y}_i \in \mathbf{Y},\ \mathbf{w}^T\mathbf{y}_i + b < 0$$

Let's suppose their convex hull intersect, so there exists $\mathbf{z}$ such that

$$\mathbf{z} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n = \sum_{m=1}^{M} \beta_m \mathbf{y}_m$$

such that $\alpha_i > 0, \beta_i > 0$ and $\sum_{i=1}^{N} \alpha_i = 1, \sum_{i=1}^{M} \beta_i = 1$
Using the condition obtained from linear separability,

$$\mathbf{w}^T\mathbf{x}_i + b > 0 \implies \alpha_i\mathbf{w}^T\mathbf{x}_i + \alpha_i b > 0$$

$$\implies \sum_{i=1}^{N} \alpha_i\mathbf{w}^T\mathbf{x}_i + \sum_{i=1}^{N} \alpha_i b > 0 \implies \mathbf{z} + b > 0$$

Also,

$$\mathbf{w}^T\mathbf{y}_i + b < 0 \implies \beta_i\mathbf{w}^T\mathbf{y}_i + \beta_i b < 0$$

$$\implies \sum_{i=1}^{M} \beta_i\mathbf{w}^T\mathbf{y}_i + \sum_{i=1}^{M} \beta_i b < 0 \implies \mathbf{z} + b < 0$$

Hence Contradiction. So the assumption was wrong.
Part 2: If $\mathbf{X}$ and $\mathbf{Y}$ have non intersecting convex hulls then they are linearly separable.
Let the minimum distance between the convex hulls be d and this distance would always exist for 2 non-overlapping convex hulls. And let the points be $\mathbf{x}$ and $\mathbf{y}$ such that $\mathrm{d} = |\mathbf{x} - \mathbf{y}|$. So, Claim is that there exists a hyperplane separating these and it passes through midpoint($\mathbf{m}$) between $\mathbf{x}$ and $\mathbf{y}$, and is perpendicular to line joining $\mathbf{x}$ and $\mathbf{y}$.
So distance of $\mathbf{x}$ from the plane is d/2.
Suppose there exists some other $\mathbf{x}_1$ whose perpendicular distance from plane is less than d/2.
Now the line joining $\mathbf{x}_1$ and $\mathbf{x}$ can't be parallel to hyperplane.
Since it is convex hull, all points on the line joining $\mathbf{x}_1$ and $\mathbf{x}$ have distance less than d/2 from the plane. Take a point $\mathbf{x}_2$ on this line whose distance from $\mathbf{m}$ is less than d/2.
Now this point has distance from $\mathbf{y}$ less than d which is absurd since nearest point to $\mathbf{y}$ is $\mathbf{x}$ whose distance is d. Hence there can't be any point whose perpendicular distance from plane is less than d/2.
So we have a hyperplane.
Hence proved.

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION
6

The new condition is

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq m$$
$$\implies y_n(\frac{\mathbf{w}^T}{m}\mathbf{x}_n + \frac{b}{m}) \geq 1$$
$$\implies y_n(\mathbf{w}_1^T\mathbf{x}_n + b_1) \geq 1$$

where $\mathbf{w}_1 = \frac{\mathbf{w}}{m}$ and $b_1 = \frac{b}{m}$
The Lagrangian becomes,

$$\mathcal{L}(\mathbf{w}_1, b_1, \alpha) = \frac{||\mathbf{w}_1||}{2} + \sum_{n=1}^{N} \alpha_n(1 - y_n(\mathbf{w}_1^T\mathbf{x}_n + b_1))$$

which learns the hyperplane given by

$$\mathbf{w}_1\mathbf{x}_n + b_1 = 0$$

Now since $\mathbf{w}_1 = \frac{\mathbf{w}}{m}$ and $b_1 = \frac{b}{m}$,

$$\implies \frac{\mathbf{w}^T}{m}\mathbf{x}_n + \frac{b}{m} = 0$$
$$\implies \mathbf{w}^T\mathbf{x}_n + b = 0$$

which is same as the hyperplane learned with original conditions. So changing the condition does not change the effective hyperplane learned by SVM.

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

Student Name: Sahil Dhull
Roll Number: 160607
Date: September 30, 2018

QUESTION 7

Note: For all the plots, red points are in positive class and blue points are in negative class. SVM part is implemented using Scikit Learn.

Part 1

(On binclass.txt)

1. Different Variances σ_+ and σ_-
   $\sigma_+ = 7.30$ and $\sigma_- = 4.36$

$$\mu_+ = \begin{bmatrix} 10.01 \\ 19.55 \end{bmatrix}, \ \mu_- = \begin{bmatrix} 20.32 \\ 9.69 \end{bmatrix}$$

   Plot:

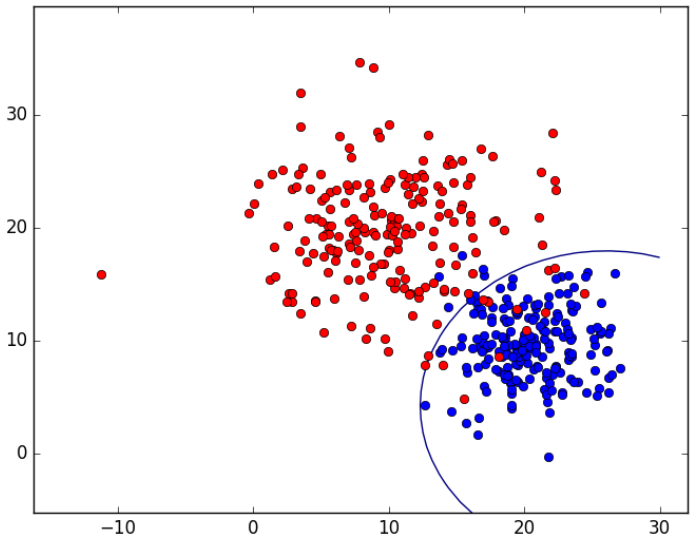

Figure 1: Generative with quadratic boundary for binclass.txt

2. Same Variance σ
   Plot:

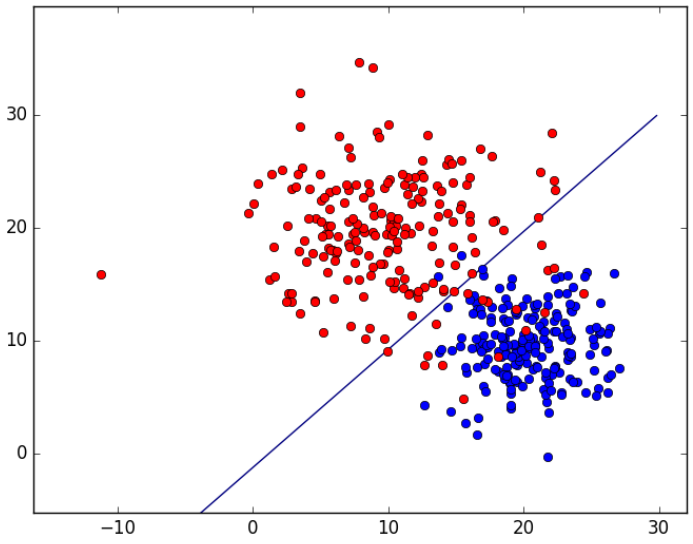

Figure 2: Generative with linear boundary for binclass.txt

3. Using SVM classifier with linear kernel
   Plot:

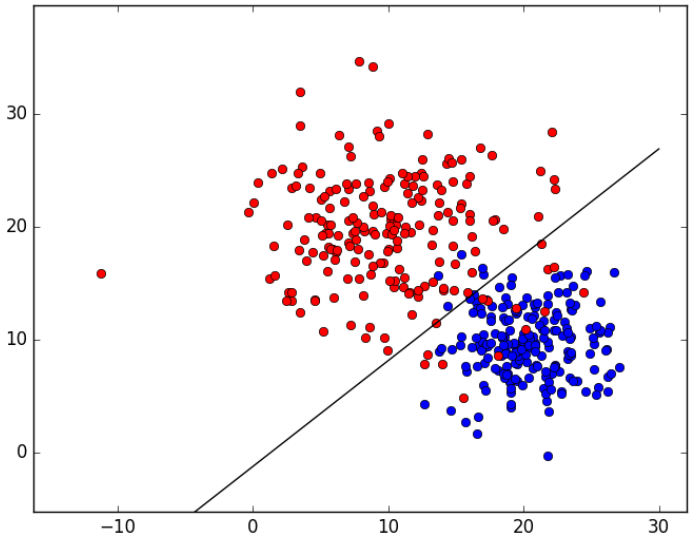

Figure 3: SVM with linear kernel for binclass.txt

Part 2

(On binclassv2.txt)

1. Different Variances σ_+ and σ_-

$\sigma_+ = 10.71$ and $\sigma_- = 4.36$

$$\mu_+ = \begin{bmatrix} 10.58 \\ 18.56 \end{bmatrix}, \ \mu_- = \begin{bmatrix} 20.32 \\ 9.69 \end{bmatrix}$$

Plot:

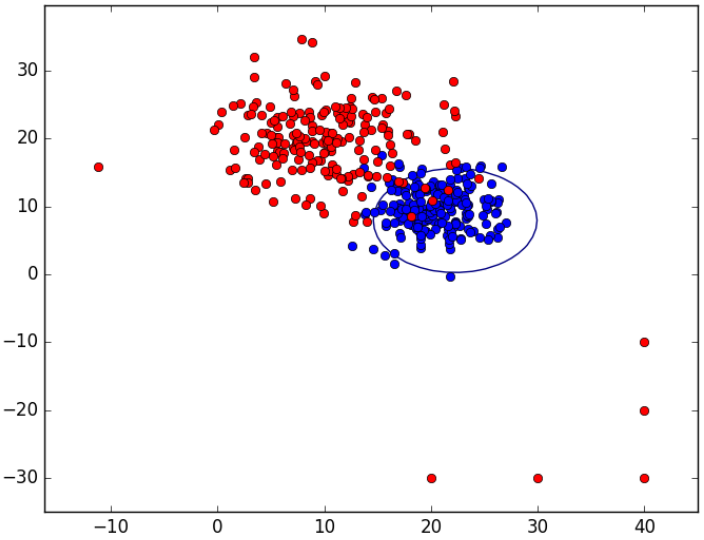

Figure 4: Generative with linear boundary for binclassv2.txt

2. Same Variance σ
   Plot:

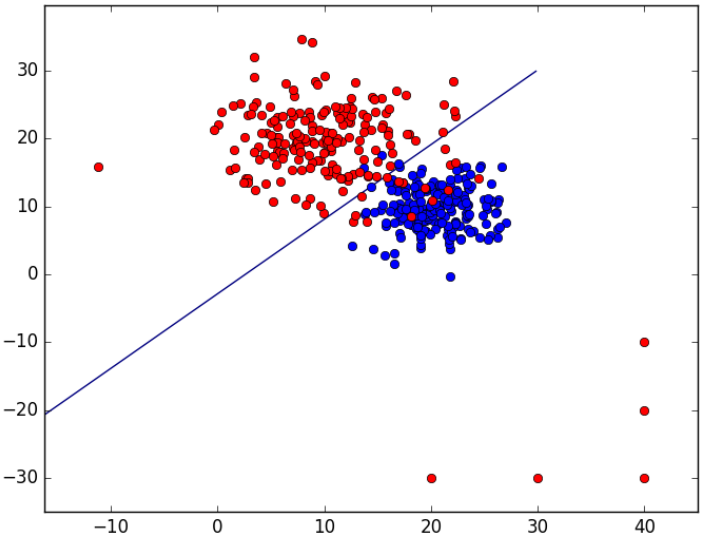

Figure 5: Generative with linear boundary for binclassv2.txt

3. Using SVM classifier with linear kernel
   Plot:

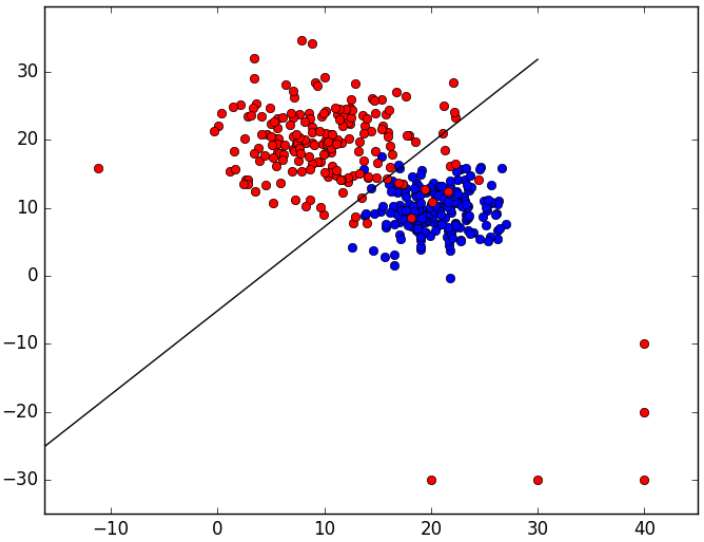

Figure 6: SVM with linear kernel for binclassv2.txt

Conclusion:

Generative classification with Gaussian class conditional works better for 1st dataset and SVM works better for 2nd dataset as their were outliers in the second case.
In general, SVM works better even with outliers.