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

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 1

Given, likelihood and prior distributions as:

$$\begin{aligned} p(y_n|x_n, w) &= \sigma(y_n w^T x_n) \\ &= \frac{1}{1+\exp(-y_n w^T x_n)} \end{aligned}$$

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

While estimating parameters using MAP we maximize the posterior distribution:

$$w_{MAP} = \arg\max_w p(w|Y, X)$$

$$\begin{aligned} p(w|Y, X) &= \frac{p(Y|w, X)p(w|X)}{p(Y|X)} \\ &= \frac{(\prod_{n=1}^{N} p(y_n|x_n, w))p(w|X)}{p(Y|X)} \quad \text{(assuming i.i.d)} \end{aligned}$$

Now we can take log on both sides since log is a monotonically increasing function the maximizing argument won't change:

$$\begin{aligned} w_{MAP} &= \arg\max_w p(w|Y, X) \\ &= \arg\max_w [\ \log(p(w|Y, X))] \\ &= \arg\max_w [\ C_o + \frac{-\lambda}{2} w^T w + \sum_{n=1}^{N} -\log\left(1+\exp(-y_n w^T x_n)\right)] \\ &= \arg\max_w [\ f(w)] \end{aligned}$$

where C_o is a constant with respect to w, Now we can differentiate this and equate to 0 to get MAP estimate:

$$\begin{aligned} \frac{\partial f(w)}{\partial w} &= -w\lambda + \sum_{n=1}^{N} \frac{y_n x_n}{1+\exp(y_n w^T x_n)} = 0 \\ w &= \frac{1}{\lambda} \sum_{n=1}^{N} \frac{1}{1+\exp(y_n w^T x_n)} y_n x_n \\ &= \sum_{n=1}^{N} \alpha_n y_n x_n \end{aligned}$$

where,

$$\begin{aligned}\alpha_n &= \frac{1}{\lambda(1+\exp(y_n w^T x_n))} \\ &= \frac{1}{\lambda}[1 - \frac{1}{1+\exp(-y_n w^T x_n)}] \\ &= \frac{1}{\lambda}[1 - p(y_n|w, x_n)]\end{aligned}$$

We can see the term α_n is variance of w multiplied with (1 - current likelihood) of y_n given x_n and current w which gives a estimate of how wrong our current prediction for a given y_n is.
i.e.
After every iteration if prediction is wrong then the contribution of $y_n x_n$ in calculating w increases proportionally to amount by which estimate is wrong. Also the variance decides how much the estimate should vary which is like pulling w towards current value (i.e regularization effect).

So this makes complete sense as to why the expressions are coming like this.
This concludes the answer for this question.

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

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 2

Given, the following distritbution family data

$$p(y=1) = \pi$$
$$p(x|y=1) = \prod_{d=1}^{D} p(x_d|y=1)$$
$$p(x_d|y=1) = (\mu_{d1})^{x_d}(1-\mu_{d1})^{1-x_d}$$

$$p(y=0) = 1-\pi$$
$$p(x|y=0) = \prod_{d=1}^{D} p(x_d|y=0)$$
$$p(x_d|y=0) = (\mu_{d0})^{x_d}(1-\mu_{d0})^{1-x_d}$$

To show equivalence with probabilistic discriminative classifier(i.e considering logistic regression)...

$$\begin{aligned} p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\ &= \frac{\pi \prod_{d=1}^{D} p(x_d|y=1)}{p(x)} \\ &= \frac{\pi \prod_{d=1}^{D} (\mu_{d1})^{x_d}(1-\mu_{d1})^{1-x_d}}{(\pi)\prod_{d=1}^{D}(\mu_{d1})^{x_d}(1-\mu_{d1})^{1-x_d} + (1-\pi)\prod_{d=1}^{D}(\mu_{d0})^{x_d}(1-\mu_{d0})^{1-x_d}} \\ &= \frac{1}{1+(\frac{1-\pi}{\pi})\prod_{d=1}^{D}(\frac{\mu_{d0}}{\mu_{d1}})^{x_d}(\frac{1-\mu_{d0}}{1-\mu_{d1}})^{1-x_d}} = \frac{1}{1+g(x)} \end{aligned}$$

which is similar to logistic regression where the g(x) was a sigmoid function. For decision boundary we need to equate the probabilities of two classes i.e at decision boundary:

$$p(y=1|x) = p(y=0|x)$$
$$(\pi)\prod_{d=1}^{D}(\mu_{d1})^{x_d}(1-\mu_{d1})^{1-x_d} = (1-\pi)\prod_{d=1}^{D}(\mu_{d0})^{x_d}(1-\mu_{d0})^{1-x_d}$$

Now on taking log on both sides the equation of decision boundary would be:

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

which is linear equation so the decision boundary is linear discriminator.

If i equate this to a logistic regression model where,

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

we will get,

$$(\frac{1-\pi}{\pi})\prod_{d=1}^{D}(\frac{\mu_{d0}}{\mu_{d1}})^{x_d}(\frac{1-\mu_{d0}}{1-\mu_{d1}})^{1-x_d} \;=\; \exp(-(w^T x+b))$$

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

which can be re written as,

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

so we can write,

$$w_d = \log(\frac{\mu_{d1}*(1-\mu_{d0})}{\mu_{d0}*(1-\mu_{d1})})$$

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

This concludes the answer to this question.

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

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 3

We have our optimization problem defined as:

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

We can use Lagrange approximation to convert this into uncontrained optimization problem over w as:

$$\begin{aligned}\hat{w} &= \arg\min_{w} \sum_{n=1}^{N} (y_n - w^T x_n)^2 + \max_{\alpha, \alpha > 0} \alpha(w^T w - c) \\ &= \arg\max_{\alpha, \alpha > 0} \min_{w} (Y - Xw)^T (Y - Xw) + \alpha(w^T w - c)\end{aligned}$$

Now we can differentiate this new optimization objective with respect to w and equate to 0:

$$\begin{aligned}-2X^T(Y - Xw) + 2\alpha w &= 0 \\ w &= (X^T X + \alpha I)^{-1} X^T Y\end{aligned}$$

Which is exact same solution to a l_2 regularized linear regression optimization solution where α is the regularization constant.
This concludes answer to this question.

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

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 4

We are using softmax function as class probability:

$$p(y_n = k|x_n, W) = \mu_{nk} = \frac{\exp(w_k^T x_n)}{\sum_{l=1}^{K} \exp(w_l^T x_n)}$$

So we can write our Negative log likelihood loss function as:

$$\begin{aligned} NLL(W) &= -\log(P(Y|X, W)) = -\log(\prod_{k=1}^{K}\prod_{n=1}^{N} p(y_n = k|x_n, W)^{\mathbb{I}(y_n=k)}) \\ &= -\sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{I}(y_n = k)\{w_k^T x_n - \log(\sum_{l=1}^{K} \exp(w_l^T x_n))\} \end{aligned}$$

We can now differentiate our loss function w.r.t to w to get the gradient as:

$$\begin{aligned} \Delta_{w_k} NLL(W) &= -\sum_{n=1}^{N} \{I(y_n = k)\{x_n\} - \frac{x_n \exp(w_k^T x_n)}{\sum_{l=1}^{K} \exp(w_l^T x_n)}\} \\ &= -\sum_{n=1}^{N} x_n(\mathbb{I}(y_n = k) - \mu_{nk}) \end{aligned}$$

We can now write down the gradient descent update equation as:

$$\begin{aligned} w_k^{t+1} &= w_k^t - \eta \Delta_{w_k} NLL(W) \\ &= w_k^t + \eta \sum_{n=1}^{N} x_n(\mathbb{I}(y_n = k) - \mu_{nk}) \end{aligned}$$

For stochastic gradient descent in each iteration N = 1 so we can write update equation as:

$$w_k^{t+1} = w_k^t + \eta x_n(\mathbb{I}(y_n = k) - \mu_{nk})$$

We can observe that in SGD the update will only happen for one w vector for that particular class and loss function will reduce in only one class. Also that update will be in approximated gradient direction to actual one.

So the sketch of updates will come something like this assuming x is two dimensional:

SGD Sketch:
1. Sample a object i from training data.
2. Calculate $\mu_{\mathbf{nk}}$ using softmax function.
3. Use update equation given above to update w.
4. Repeat steps 1-3 for every i.
5. Repeat steps 1-4 iteratively till convergence.

In case the μ_{nk} is hard class assignment i.e will be 1 if k is the highest probability class then two conditions on y_n and μ_{nl} can be mixed as:

$$w_k^{t+1} = w_k^t + \eta x_n (1 - \mathbb{I}(y_n = \arg\max_l \mu_n l))$$

SGD Sketch(hard):
1. Sample a object i from training data.
2. Calculate $\arg\max_{\mathbf{l}}(\mu_{\mathbf{nl}})$ using softmax function.
3. Use update equation given above to update w.
4. Repeat steps 1-3 for every i.
5. Repeat steps 1-4 iteratively till convergence.

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

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 5

We define a convex hull as any linear combination of points of in flowing manner will lie inside hull,

$$X = \sum_n \alpha_n x_n| \quad x_n, X \epsilon A\{hull\}$$
$$\text{such that}, \sum_n \alpha_n = 1, \alpha_n \geq 0$$

Two sets A,B are linearly separable if for all points in A and B there exists a W such that,

$$w^T x + b > 0, \forall x \epsilon A$$
$$w^T y + b < 0, \forall y \epsilon B$$

Lets consider that two hulls are linearly separable and intersects.
Lets say Y is a element of B and A, B intersect we can write:

$$Y = \sum_n \alpha_n x_n \mid x_n \epsilon A$$
$$w^T Y + b = \sum_n \alpha_n (w^T x_n + b)$$
$$= \sum_n \alpha_n (+ve)$$
$$> 0$$

This violates the condition that we had on set B. so due to contradiction we can say if two convex hulls intersect they are not linearly separable.

Now lets, prove the converse.
Given that two convex hulls do not intersect then there exist closest pair of points from hulls such that,

$$x*, y* = \arg\min_{x,y} |x - y|$$
$$s.t, x \epsilon A, y \epsilon B$$

we define,

$$d = |x* - y*|$$

We can say there exist a hyperplane that passes through mis point of this points from which all points on hulls are alteast d/2 distance away.

$$w^\intercal x_i > \frac{d}{2}, \forall x_i \epsilon A$$
$$w^\intercal y_i < \frac{d}{2}, \forall y_i \epsilon B$$

which proves that non intersecting hulls are linearly seperable.
This proves that the set of x_s and the set of y_s are linearly separable if and only if the convex hulls do not intersect.

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 6

We set our constraint as,

$$y_n(w^T x_n + b) \geq m$$

Our margin now can be written as:

$$\gamma = \min_n \frac{|w^T x_n + b|}{||w||} = \frac{m}{||w||}$$

we can write our optimization problem as:

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

we can define now two parameters:

$$\hat{w} = \frac{w}{m}, \hat{b} = \frac{b}{m}$$

Now our optimization problem becomes:

$$\arg \max_{\alpha, \alpha \geq 0} \min_{w,b} L(w, b, \alpha) = \frac{\hat{w}^T \hat{w}}{2} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\hat{w}^T x_n + \hat{b}))$$

which is exactly similar to Linear SVM optimization problem with m=1.
So our solution will be scaled by a factor m for both w and b but effective hyperplane will remain same.

This concludes the answer to this question.

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

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: September 29, 2018

QUESTION 7

In Generative Classification we can directly take the comparison between probabilities of each class. And classify the point to class with maximum class likelihood.
Here we are talking about classification. So we can write,

$$\begin{aligned} p(y_n = 1|x_n) &= \frac{p(x_n|y_n = 1)p(y_n = 1)}{p(x)} \\ &= \frac{0.5 * ((2\pi)^k|\Sigma_+|)^{-0.5} \exp(-0.5 * (x_n - \mu_+)^T \Sigma_+^{-1}(x_n - \mu_+))}{p(x)} \\ &= C_0(|\Sigma_+|)^{-0.5} \exp(-0.5 * (x_n - \mu_+)^T \Sigma_+^{-1}(x_n - \mu_+)) \end{aligned}$$

Where C_0 is a constant with respect to σ_+, μ_+,
We now define our objective function as:

$$\begin{aligned} L &= \log \prod_{n=1:y_n=1}^{N} p(y_n = 1|x_n) \\ &= \sum_{n=1:y_n=1}^{N} C_0 - \frac{1}{2}\log(|\Sigma_+|) - \frac{1}{2}(x_n - \mu_+)^T \Sigma_+^{-1}(x_n - \mu_+)) \end{aligned}$$

Now we differentiate this equation with σ_+, μ_+ to get optimal solution,

$$\begin{aligned} &\frac{\partial L}{\partial \mu_+} = 0 \implies \sum_{n=1:y_n=1}^{N} (x_n - \mu_+) = 0 \\ &\implies \mu_+ = \frac{1}{N_+} \sum_{n=1:y_n=1}^{N} x_n \\ &\frac{\partial L}{\partial \sigma_+^2} = 0 \implies \sum_{n=1:y_n=1}^{N} -\frac{D}{\sigma_+^2} + \frac{(x_n - \mu_+)^T(x_n - \mu_+)}{\sigma_+^4} \\ &\implies \sigma_+^2 = \frac{1}{N_+ D} \sum_{n=1:y_n=1}^{N} (x_n - \mu_+)^T(x_n - \mu_+) \end{aligned}$$

Where D is the number of dimensions of data object, N_+ is number of positive class objects.
Similarly we can obtain the results for class -1 as:

$$\begin{aligned} &\implies \mu_- = \frac{1}{N_-} \sum_{n=1:y_n=-1}^{N} x_n \\ &\implies \sigma_-^2 = \frac{1}{N_- D} \sum_{n=1:y_n=-1}^{N} (x_n - \mu_-)^T(x_n - \mu_-) \end{aligned}$$

Similarly for calculating in case where both variances are same the solution for μ_+, μ_- won't change as it does not depend on that but solution for variance will change so we need to maximize:

$$p(Y|X) = \prod_{n=1}^{N} p(y_n = 1|x_n)^{\mathbb{I}(y_n=1)} p(y_n = 0|x_n)^{\mathbb{I}(y_n=0)}$$

with respect to σ. So we take log and differentiate this and equate to zero and we get:

$$\implies \sigma^2 = \frac{1}{ND}\{\sum_{n=1:y_n=+1}^{N} (x_n - \mu_+)^T(x_n - \mu_+) + \sum_{n=1:y_n=-1}^{N} (x_n - \mu_-)^T(x_n - \mu_-)\}$$

$$\sigma^2 = \frac{1}{ND}\{N_+\sigma_+^2 + N_-\sigma_-^2\}$$

where N is total number of samples.

File 1: binclass.txt plots

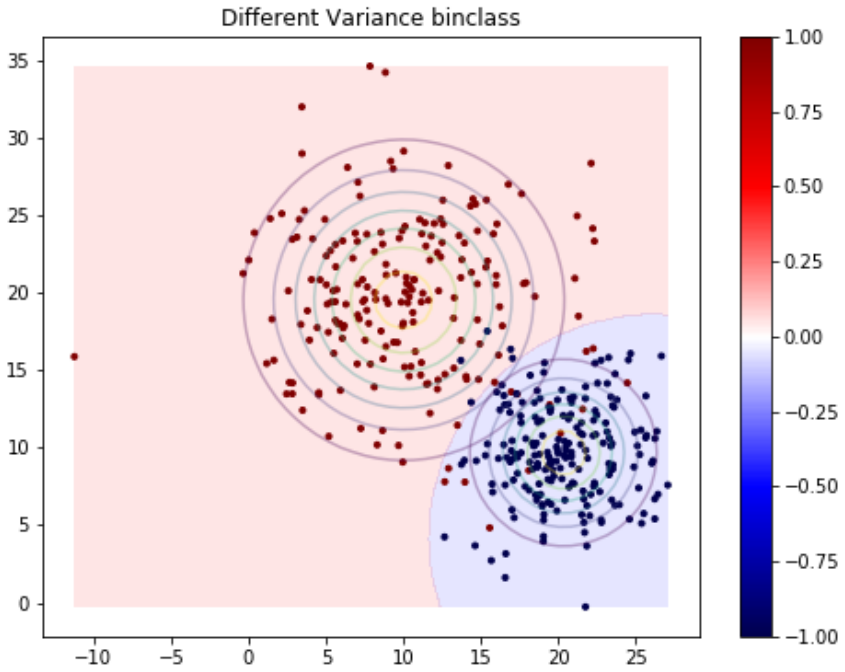

Figure 1: Gaussian Contours and Decision boundary Non linear

Observation for binclass.txt file the linear separator SVM as well as the Gaussian separator(less error) behaved almost equally as the data was almost linearly separable.

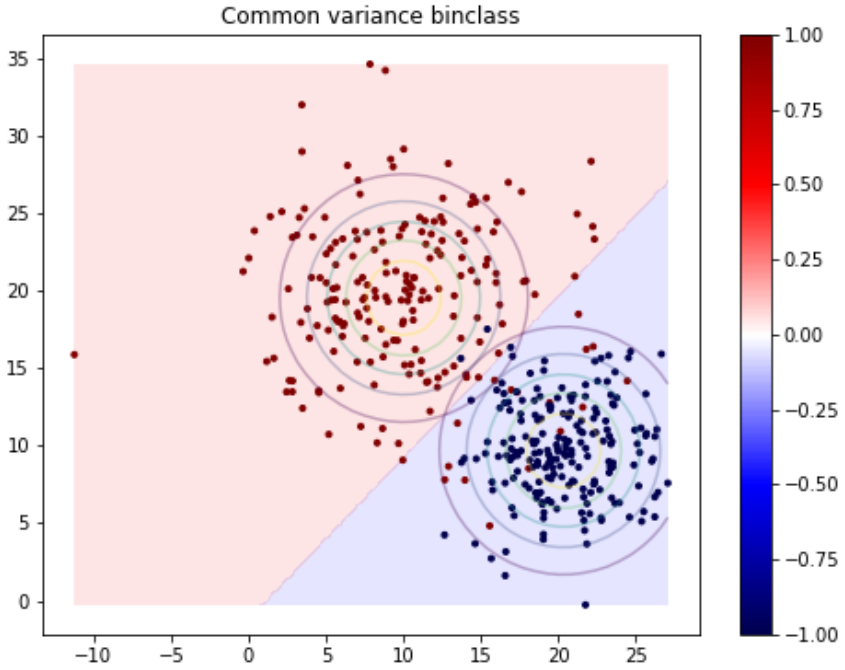

Figure 2: Gaussian Contours and Decision boundary Linear

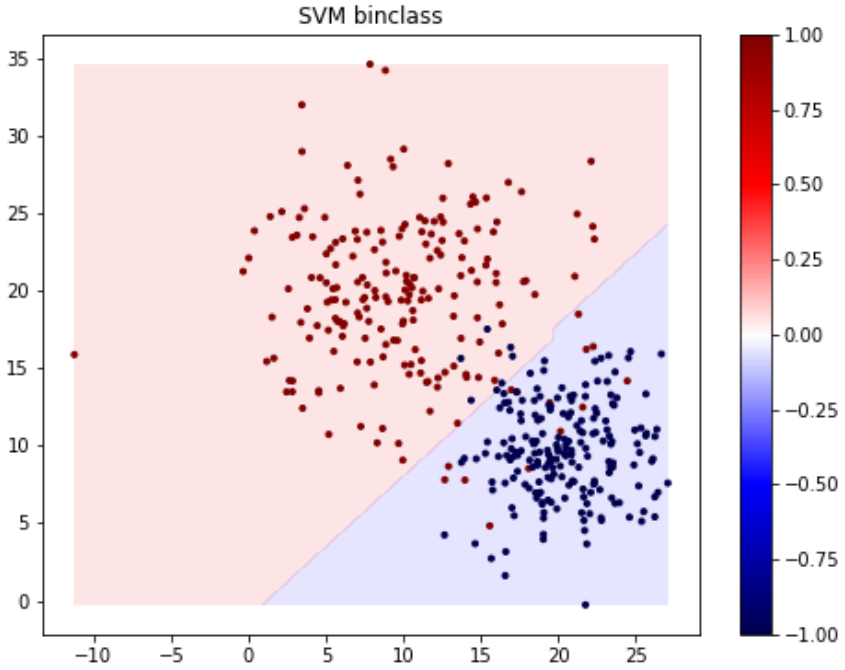

Figure 3: SVM Decision boundary

File 2: binclassv2.txt plots

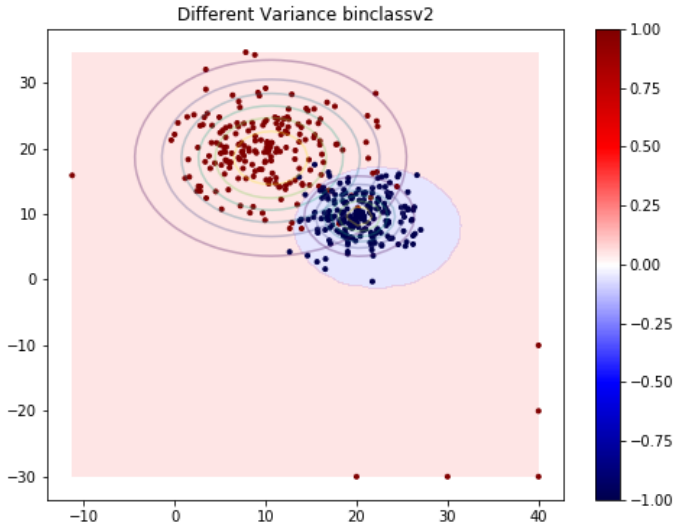

Figure 4: Gaussian Contours and Decision boundary Non linear

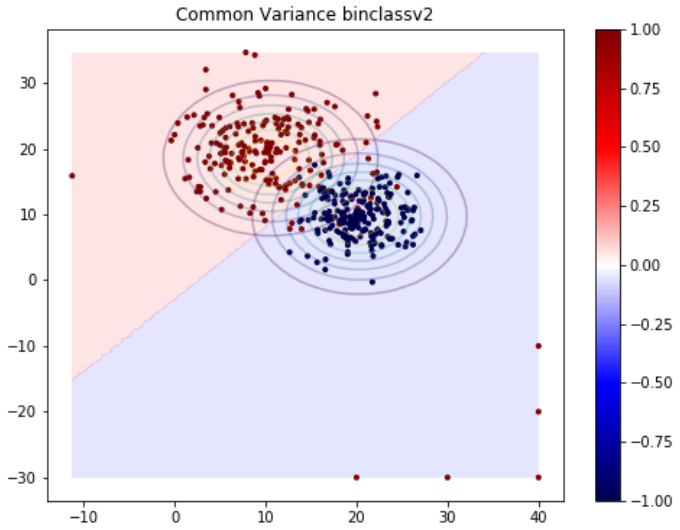

Figure 5: Gaussian Contours and Decision boundary Linear

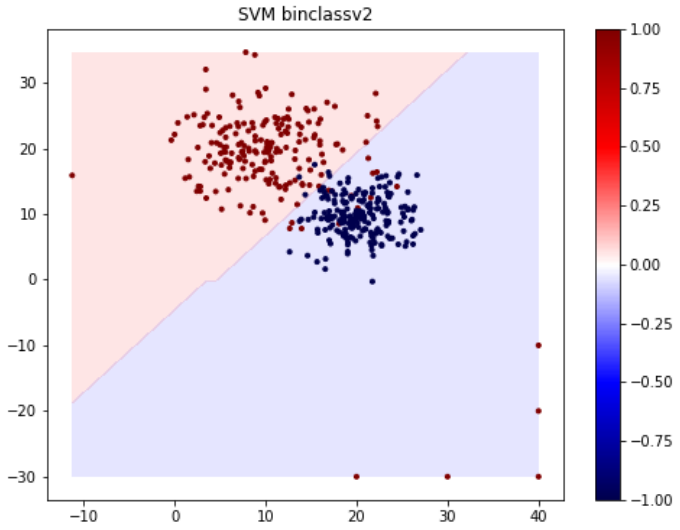

Figure 6: SVM Decision boundary

Here on the other hand the data is have some domain points after blue points so nonlinear boundary is required to be learned and to do so and generative is better as compared to SVM (assuming the points might not be outliers).

This concludes the answer to above question.