Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 2, 2018

QUESTION 1

A vector symbol $\mathbf{b}$, a symbol in blackboard font $\mathbb{R}$, a symbol in calligraphic font $\mathcal{A}$, some colored text

$$\begin{aligned}\delta_* &= \underset{\delta}{argmax} f(\alpha + \delta \mathbf{e}_n) \\ f(\alpha + \delta \mathbf{e}_n) &= \alpha^T \check{1} + \delta - \frac{1}{2}[\alpha^T \mathbf{G}\alpha + \delta(\alpha^T \mathbf{G}\mathbf{e}_n + \mathbf{e}_n \mathbf{G}\alpha) + \delta^2 G_{nn}] \\ &= \alpha^T \check{1} + \delta - \frac{1}{2}[\alpha^T \mathbf{G}\alpha + 2\delta\alpha^T \mathbf{G}\mathbf{e}_n + \delta^2 G_{nn}]\end{aligned}$$

Now differentiating w.r.t. δ we get,

$$\delta_* = \frac{(1 - \alpha^T \mathbf{G}\mathbf{e}_n)}{Gnn}$$

But we have to also maintain the constraint that $0 \leq \alpha_n \leq C$. So whenever the update $\alpha_n = \alpha_n + \delta_*$ goes out of the bound we take it inside the bound by choosing the nearest boundary points(projected gradient ascent). In algebric terms:

$$\delta_* = max\left(-\alpha_n, min\left(C - \alpha_n, \frac{(1 - \alpha^T \mathbf{G}\mathbf{e}_n)}{Gnn}\right)\right)$$

Algorithm:

- Initialize α, set t=1.
- Randomly pick any $n \in \{1 \ldots N\}$.
- Compute $\delta_* = max\left(-\alpha_n, min\left(C - \alpha_n, \frac{(1-\alpha^T \mathbf{G}\mathbf{e}_n)}{Gnn}\right)\right)$.
- Update $\alpha_n = \alpha_n + \delta_*$
- Set t = t + 1 and repeat until convergence.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 2, 2018

QUESTION 2

Note that:

$$\sum_{n,m} I[f_n = f_m] \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2 + \sum_{n,m} I[f_n \neq f_m] \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2 = \sum_{n,m} \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2$$

Here the right side of the term will be the constant as it is the sum of distance of every pair of points. So by taking the second term on left hand side to the right hand side will give us the desired result. Therefore:

$$argmin_f \mathcal{L}_w = argmin_f \sum_{n,m} I[f_n = f_m] \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2 = argmax_f \sum_{n,m} I[f_n \neq f_m] \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2$$

This clearly shows that minimizing $\mathcal{L}_W$, which is defined as the sum of squared distances between all pairs of points that are within the same cluster, implicitly also maximizes the sum of squared distances between all pairs of points that are in different clusters.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 2, 2018

QUESTION 3

Since $\mathbf{x}_n = [\mathbf{x}_n^{miss}, \mathbf{x}_n^{obs}]$ where $\mathbf{x}_n \sim \mathcal{N}(\mu, \Sigma)$ we can also write:

$$\mu = \begin{pmatrix} \mu_o \\ \mu_m \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{oo} & \Sigma_{om} \\ \Sigma_{mo} & \Sigma_{mm} \end{pmatrix}$$

where μ_o denotes the mean value of the observed part and μ_m denotes mean value of the missing part.
We will model this problem as latent variable model where the missing values are treated as latent variables. We will use Expectation Maximization algorithm to find the parameters when the data is missing. The general steps for EM will be as follows

1. First estimate the posterior distribution of latent variable given current parameter and data.
2. Compute the complete data log likelihood function i.e. $\log(P(\mathbf{X}, \mathbf{Z}|\Theta)$
3. Compute the Expected CLL with respect to latent variable following the posterior distribution we calculated in step 1. Also if you want to estimate the latent variable then update it with the expectation of the latent variable w.r.t posterior distribution. For expected CLL replace $\mathbf{z}_n$ with $E_{p(\mathbf{z}_n|\mathbf{x}_n,\Theta)}[\mathbf{z}_n]$, $\mathbf{z}_n\mathbf{z}_n^T$ with $E_{p(\mathbf{z}_n|\mathbf{x}_n,\Theta)}[\mathbf{z}_n\mathbf{z}_n^T]$ and so on....
4. Find and update the parameters that maximizes the expected CLL.
5. Repeat the process until convergence.

Part 1
In this part of the question we were asked to find the posterior probability distribution. Using section 4.3.1 of MLAP we find that $p(\mathbf{x}_n^{miss}|\mathbf{x}_n^{obs}, \mu, \Sigma) \sim \mathcal{N}(\mathbf{x}_n^{miss}|\mu_{m|o}^n, \Sigma_{m|o}^n)$ where $\mu_{m|o}^n = \mu_m^n + \Sigma_{mo}^n\Sigma_{oo}^{-1}(\mathbf{x}_n^{obs} - \mu_o^n)$ and $\Sigma_{m|o}^n = \Sigma_{mm}^n - \Sigma_{mo}^n\Sigma_{oo}^{-1}\Sigma_{om}^n$. Note that for each point mean and covariance matrix may be different since we don't know which and how many components of the data point will be missing.

Part 2
Equation for CLL is

$$\begin{aligned}
\log(P(\mathbf{X}, \mathbf{Z}|\Theta) &= \sum_{n=1}^{N} log(P(\mathbf{x}_n^{miss}, \mathbf{x}_n^{obs}|\mu, \Sigma) \\
&= \sum_{n=1}^{N} log\left(P(\mathbf{x}_n^{miss}|\mathbf{x}_n^{obs}\mu, \Sigma)P(\mathbf{x}_n^{obs}|\mu, \Sigma)\right) \\
&= \sum_{n=1}^{N} \log \mathcal{N}(\mathbf{x}_m^n|\mu_{m|o}^n, \Sigma_{m|o}^n) + \sum_{n=1}^{N} \log \mathcal{N}(\mathbf{x}_o^n|\mu_o^n, \Sigma_{oo}^n)
\end{aligned}$$

Now I will use the following form of Normal distribution:

$$\mathcal{N}(\mathbf{x}|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} exp\left[-\frac{1}{2} trace[\Sigma^{-1}\mathbf{S}]\right]$$

where,

$$\mathbf{S} = (\mathbf{x}-\mu)(\mathbf{x}-\mu)^T$$

Using the above form we get the final form of CLL as

$$A = (\Sigma^n_{m|o})^{-1}(\mathbf{x}^n_m(\mathbf{x}^n_m)^T - \mu^n_{m|o}(\mathbf{x}^n_m)^T - \mathbf{x}^n_m(\mu^n_{m|o})^T + \mu^n_{m|o}(\mu^n_{m|o})^T)$$
$$B = (\Sigma^n_{oo})^{-1}(\mathbf{x}^n_o(\mathbf{x}^n_o)^T - \mu^n_o(\mathbf{x}^n_o)^T - \mathbf{x}^n_o(\mu^n_o)^T + \mu^n_o(\mu^n_o)^T)$$
$$CLL = -\frac{1}{2}\left(\sum_{n=1}^{N} D\log 2\pi + (\log|\Sigma^n_{m|o}| + \log|\Sigma^n_{oo}|) + trace\,(A+B)\right)$$

Now to find the expected CLL we will replace $\mathbf{x}^n_m$ with $E[\mathbf{x}^n_m]$. Similarly replace $\mathbf{x}^n_m(\mathbf{x}^n_m)^T$ with $E[\mathbf{x}^n_m(\mathbf{x}^n_m)^T]$. Note here that $\mathbf{x}^n_m$ is the latent variable and $\mathbf{x}^n_o$ is already known.

$$E[\mathbf{x}^n_m] = \mu^n_{m|o}$$
$$E[\mathbf{x}^n_m(\mathbf{x}^n_m)^T] = E[\mathbf{x}^n_m]E[\mathbf{x}^n_m]^T + cov(\mathbf{x}^n_m) = E[\mathbf{x}^n_m]E[\mathbf{x}^n_m]^T + \Sigma^n_{m|o}$$

Putting the above values we get $A = I^n_m$.

$$E[CLL] = -\frac{1}{2}\left(\sum_{n=1}^{N} D\log 2\pi + (\log|\Sigma^n_{m|o}| + \log|\Sigma^n_{oo}|) + noofelements(x^n_m) + trace\,(B)\right)$$

Part 3

Estimating μ and Σ: We will maximize ECLL to get the parameters. In previous part, we replaced $\mathbf{x}^n_m = E[\mathbf{x}^n_m]$ it means the missing values are replaced with $\mathbf{x}^{\hat{n}}_m = E[\mathbf{x}^n_m]$. It means now we have the full data points $\hat{x^n} = (x^n_o, \mathbf{x}^{\hat{n}}_m)$ which follows gaussian distribution. After applying MLE we get the following estimates:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N}\hat{x}^n$$
$$\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(\hat{x}^n - \hat{\mu})(\hat{x}^n - \hat{\mu})^T$$

EM algorithm

1. Initialize μ^0, Σ^0, set t=1
2. **E step:** For each $n = 1 \ldots N$ missing data component find $\mathbf{x}^{\hat{n}}_m = E[\mathbf{x}^n_m] = \mu^n_{m|o}$
3. Now, $\hat{x}^n = (x^n_o, \hat{\mathbf{x}_m}^n)$, using MLE find $\mu^{(t)} and \Sigma^{(t)}$ as follows:

$$\mu^{(t)} = \frac{1}{N}\sum_{n=1}^{N}\hat{x}^n$$
$$\Sigma^{(t)} = \frac{1}{N}\sum_{n=1}^{N}(\hat{x}^n - \mu^{(t)})(\hat{x}^n - \mu^{(t)})^T$$

4. Set t = t + 1 and go to step 2 if not yet converged.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 2, 2018

QUESTION 4

Here we will consider M labels of unlabelled examples as the latent variable $\mathbf{Z} = \{\mathbf{z}_n\}_{n=N+1}^{N+M}$. The CLL will be written as,

$$\begin{aligned}\log P(\mathbf{X}, \mathbf{Z}) &= \sum_{i=1}^{N} \log P(\mathbf{x}_i, \mathbf{y}_i) + \sum_{j=N+1}^{N+M} \log P(\mathbf{x}_j, \mathbf{y}_j) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}[\log \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k) + \log \pi_k] + \sum_{j=N+1}^{N+M}\sum_{k=1}^{K} z_{jk}[\log \mathcal{N}(\mathbf{x}_j|\mu_k, \Sigma_k) + \log \pi_k]\end{aligned}$$

Here $y_{nk} = 1$ if $y_n = k$ and zero otherwise. Similarly, $z_{nk} = 1$ if $z_n = k$ and zero otherwise. Also note that $\forall j = N+1 \dots N+M, \forall k = 1 \dots K$,

$$E[z_{jk}] = \frac{\pi_k \mathcal{N}(\mathbf{x}_j|\mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{x}_j|\mu_l, \Sigma_l)}$$

$$ECLL = \sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}[\log \mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k) + \log \pi_k] + \sum_{j=N+1}^{N+M}\sum_{k=1}^{K} E[z_{jk}][\log \mathcal{N}(\mathbf{x}_j|\mu_k, \Sigma_k) + \log \pi_k]$$

Define:

$$w_{nk} = \begin{cases} y_{nk} & \text{if } 0 \le n \le N \\ E[z_{nk}] & \text{if } N+1 \le n \le N+M \end{cases}$$

So we can write expected CLL as follows:

$$ECLL = \sum_{n=1}^{N+M}\sum_{k=1}^{K} w_{nk}[\log \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k) + \log \pi_k]$$

Note that above ECLL become analogous to gaussian mixture model. Since now we have all the recepie, we will write EM algorithm.

EM Algorithm:

1. Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$, as $\Theta_{(0)}$ set t=1
2. E step: Compute for all $n = N+1 \dots N+M, k = 1 \dots K$

$$E[z_{nk}^{(t)}] = \frac{\pi_k^{(t-1)} \mathcal{N}(x_n|\mu_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{l=1}^{K} \pi_l^{(t-1)} \mathcal{N}(x_n|\mu_l^{(t-1)}, \Sigma_l^{(t-1)})}$$

3. Define:

$$N_k = \sum_{n=1}^{N} y_{nk}$$

$$M_k = \sum_{n=N+1}^{N+M} E[z_{nk}]$$

Estimate Θ via MLE as follows:

$$\mu_k^{(t)} = \frac{\sum_{n=1}^{N+M} w_{nk} x_n}{N_k + M_k}$$
$$\Sigma_k^{(t)} = \frac{\sum_{n=1}^{N+M} w_{nk} (x_n - \mu_k^{(t)})(x_n - \mu_k^{(t)})^T}{N_k + M_k}$$
$$\pi_k^{(t)} = \frac{N_k + M_k}{N + M}$$

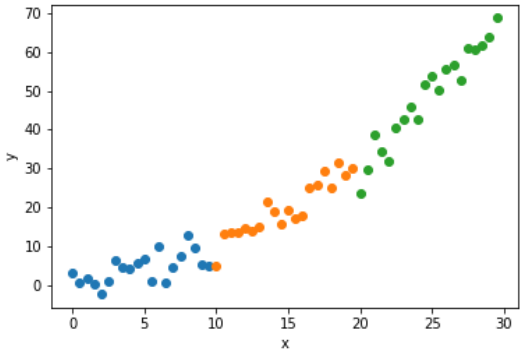
4. Set t = t + 1 and go to step 2 if not yet converged.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 2, 2018

QUESTION 5

Part 1: This model will be able to learn K different Linear hyperplanes. This is the K-fold generalisation of linear Regression model. Our previous linear regression model was able to learn only one hyperplane but it may not always be the case. For e.g. in the following figure it would be better if we learn 3 piece wise linear regression model instead of just one.



Part 2: In Alt-Opt we take the MAP estimate of posterior distribution. So $\forall n = 1 \dots N$

$$\begin{aligned}\hat{\mathbf{z}_n} &= argmax_k p(\mathbf{z}_n = k | y_n, \Theta) \\ &= argmax_k \frac{\pi_k \mathcal{N}(y_n | \mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(y_n | \mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}\end{aligned}$$

After that we maximize the CLL with respect to Θ and keeping the latent variable as calculated above. ($\Theta = \{\mathbf{w}_n\}_{n=1}^N$).

$$\begin{aligned}CLL &= \sum_{n=1}^{N} \log p(y_n, \hat{\mathbf{z}_n} | \Theta) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{\mathbf{z}}_{nk} \left(\log \pi_k + \log \beta - \frac{1}{2} \log 2\pi - \frac{\beta}{2} (y_n - \mathbf{w}_k^T \mathbf{x}_n)^2 \right)\end{aligned}$$

In the above equation $\hat{\mathbf{z}}_{nk} = I(\mathbf{z}_n = k)$. Note that this is the constrained optimization problem and solving it using lagrangian we get the following estimate $\forall k = 1 \dots K$,

$$\begin{aligned}\hat{\Theta} &= argmax_\Theta CLL \\ \hat{\pi}_k &= \frac{\sum_{n=1}^{N} \hat{\mathbf{z}}_{nk}}{N} \\ \hat{\mathbf{w}}_k &= \left(\sum_{n=1}^{N} \hat{\mathbf{z}}_{nk} \mathbf{x}_n \mathbf{x}_n^T \right)^{-1} \left(\sum_{n=1}^{N} \hat{\mathbf{z}}_{nk} y_n \mathbf{x}_n \right)\end{aligned}$$

Special Case: When $\pi_k = \frac{1}{K}, \forall k$, then the update equation for latent variable becomes,

$$\hat{\mathbf{z}_n} = argmin_k (y_n - \mathbf{w}_k^T \mathbf{x}_n)^2$$

This also makes sense intuitively as for each data point we are choosing that class that is minimizing the error. It will solely depend on the weight vector. Weight that makes the prediction as close to actual value as possible will be chosen.

Part 3:

For EM the CLL and Expected CLL is given by,

$$CLL = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\log \pi_k + \log \mathcal{N}(y_n | \mathbf{w}_k^T \mathbf{x}_n, \beta^{-1}) \right)$$

$$E[z_{nk}] = \frac{\pi_k \mathcal{N}(y_n | \mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(y_n | \mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}$$

$$ExpectedCLL = \sum_{n=1}^{N} \sum_{k=1}^{K} E[z_{nk}] \left(\log \pi_k + \log \mathcal{N}(y_n | \mathbf{w}_k^T \mathbf{x}_n, \beta^{-1}) \right)$$

EM algo:

1. Initialize $\Theta = \{\pi_k, \mathbf{w}_k\}_{k=1}^K$ as $\Theta^(0)$, set t=1.
2. E-step: Compute the expectation,

$$E[z_{nk}^{(t)}] = \frac{\pi_k^{(t-1)} \mathcal{N}(y_n | (\mathbf{w}_k^{(t-1)})^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l^{(t-1)} \mathcal{N}(y_n | (\mathbf{w}_l^{(t-1)})^T \mathbf{x}_n, \beta^{-1})}$$

3. Find MLE as follows:

$$\pi_k^{(t)} = \frac{\sum_{n=1}^{N} E[z_{nk}^{(t)}]}{N}$$

$$\mathbf{w}_k^{(t)} = \left(\sum_{n=1}^{N} E[z_{nk}^{(t)}] \mathbf{x}_n \mathbf{x}_n^T \right)^{-1} \left(\sum_{n=1}^{N} E[z_{nk}^{(t)}] y_n \mathbf{x}_n \right)$$

4. Set t = t + 1 and go to step 2 if not yet converged.

Alt-opt as special case of EM

Goal: as β goes to inf, EM-reduces to Alt-opt. First let's try to look what we get in both the cases:

$$\textbf{Alt-opt: } k' = \hat{\mathbf{z}}_n = argmax_k \frac{\pi_k \mathcal{N}(y_n | \mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(y_n | \mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}$$

$$\textbf{EM: } \hat{\mathbf{z}}_{nk} = E[z_{nk}] = \frac{\pi_k \mathcal{N}(y_n | \mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(y_n | \mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}$$

Let's say solution of alt-opt is k'. Then if we show for $\beta \to \infty$ $z_{nk} = 1$ for $k = k'$ and zero otherwise then we are done.

$$\hat{z}_{nk} = \frac{1}{1 + \sum_{l \neq k} \frac{\pi_l}{\pi_k} exp\left(-\beta (\left\| y_n - \mathbf{w}_l^T \mathbf{x}_n \right\|^2 - \left\| y_n - \mathbf{w}_k^T \mathbf{x}_n \right\|^2) \right)}$$

For $k = k'$, $(\left\| y_n - \mathbf{w}_l^T \mathbf{x}_n \right\|^2 - \left\| y_n - \mathbf{w}_k^T \mathbf{x}_n \right\|^2) > 0$ (by alt opt condition). Therefore as $\beta \to \infty$, $z_{nk} = 1$ for $k = k'$

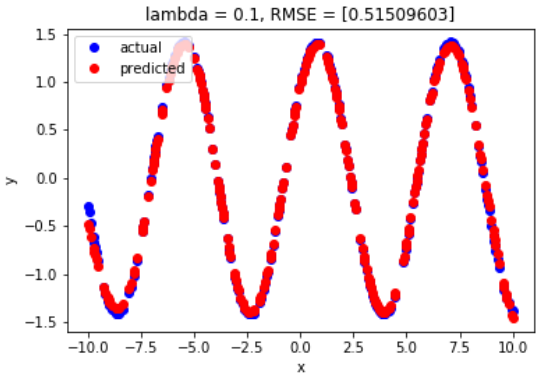
Similarly, for $k \neq k'$, $(\left\| y_n - \mathbf{w}_l^T \mathbf{x}_n \right\|^2 - \left\| y_n - \mathbf{w}_k^T \mathbf{x}_n \right\|^2) < 0$ (by alt opt condition). Therefore as $\beta \to \infty$, $z_{nk} = 0$ for $k \neq k'$.

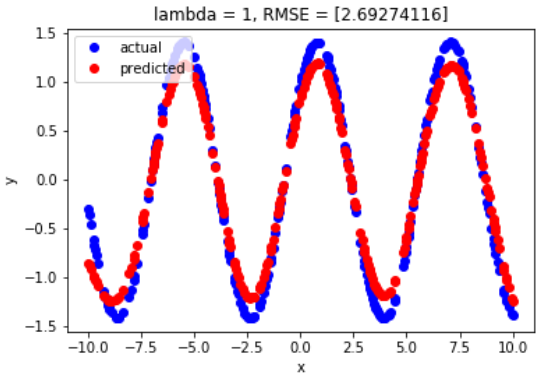
Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 3

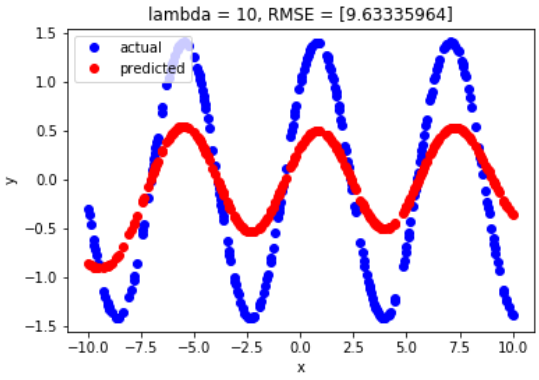
Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 2, 2018

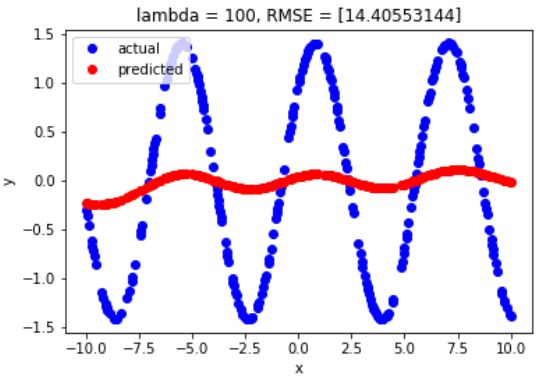
QUESTION 6

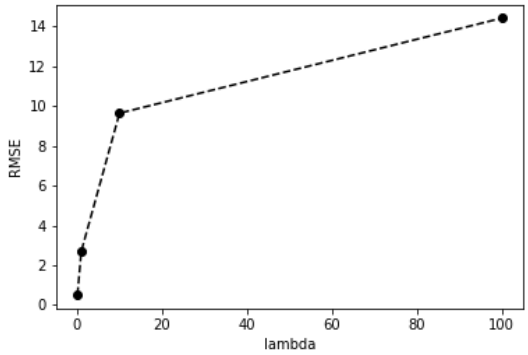
1 (a) In case of kernel ridge regression we saw that as λ increases more and more underfitting takes place. **RMSE value increases with lambda**. This also makes sense as λ acts a regulariser coeffecient whose increase in value increases the root mean square error. RMSE values corresponding to lambda [**0.1, 1, 10, 100**] are [**0.51, 2.69, 9.63, 14.40**] The following plots were obtained:



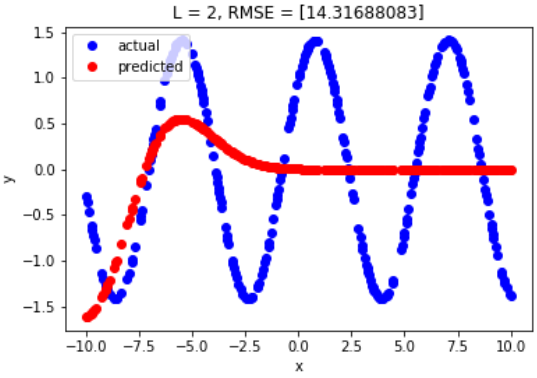


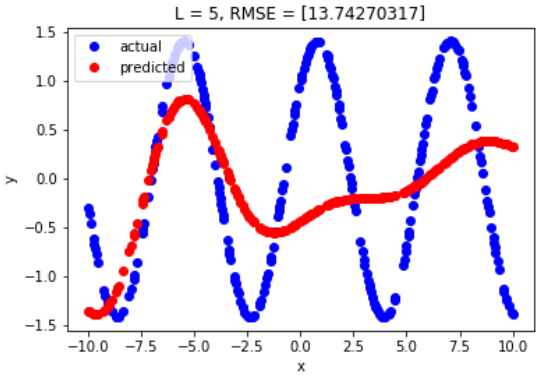


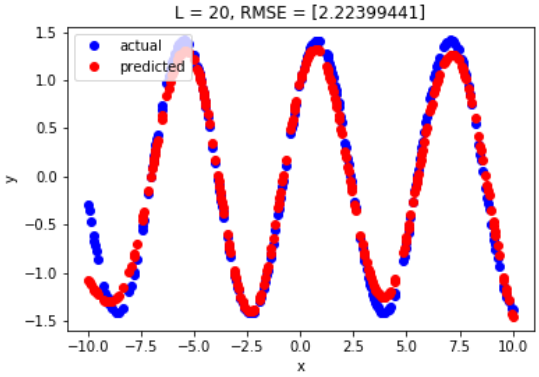


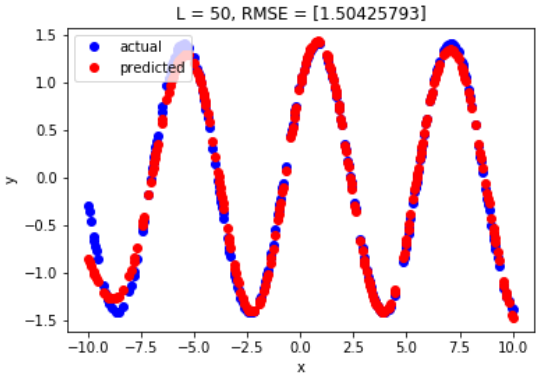


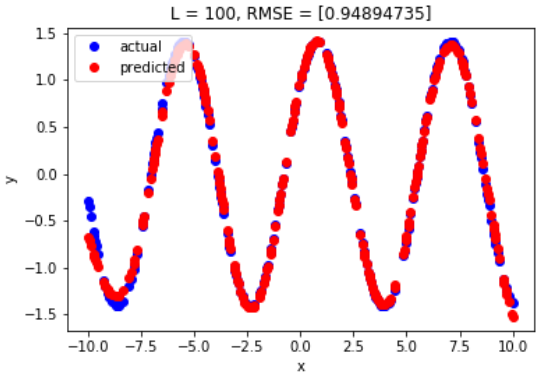
1(b) We observe that as more and more Landmarks are chosen RMSE values decreases. RMSE values corresponding to lambda **[2, 5, 20, 50, 100]** are **[14.31, 13.74, 2.23, 1.50, 0.94]**. The L=50 seems good enough as it is generalising quite well for L=100 there are chances of overfitting which may not be desirable. The following plots were obtained:

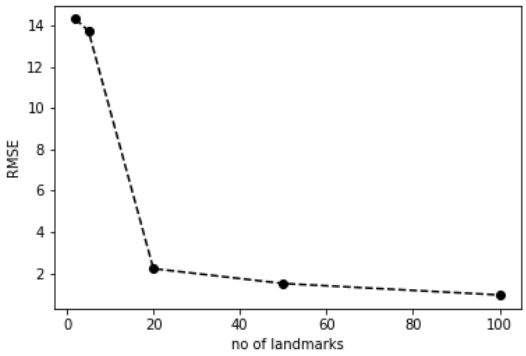



L = 5, RMSE = [13.74270317]
actual
predicted
x
y

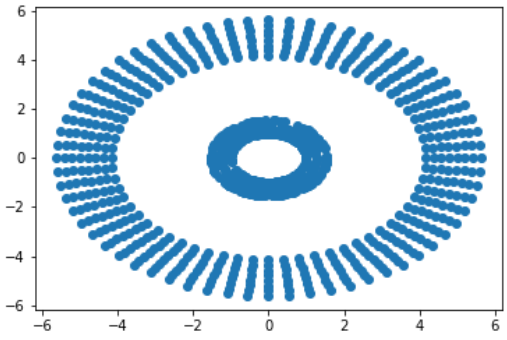

L = 20, RMSE = [2.22399441]
actual
predicted
x
y


L = 50, RMSE = [1.50425793]
actual
predicted
x
y

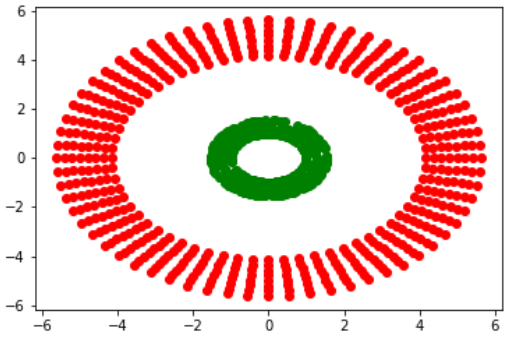

L = 100, RMSE = [0.94894735]
actual
predicted
x
y



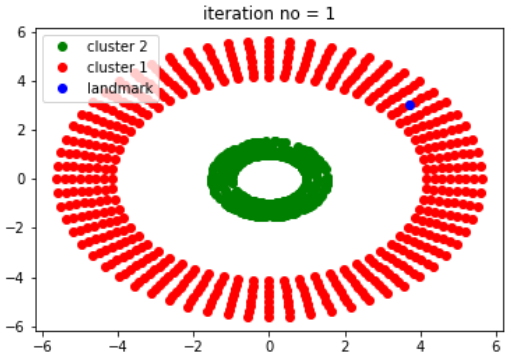
2(a) The original plot is:



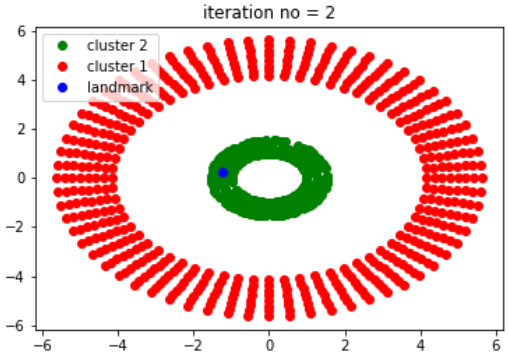
Since the dataset was not linearly separable so k means wouldn't be able to cluster them properly. I introduced a third dimension in a way such that the data points become linerly separable. For the inner circle, I took it on the $z = 10$ and for outer circle I took it to $z = -10$. After that I applied k-means.The plot after applying k-means on transformed one -



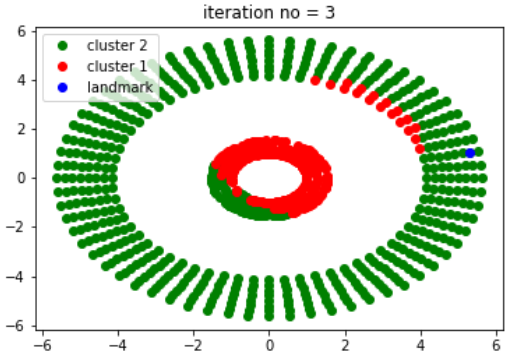
2(b) The following plots were obtained:



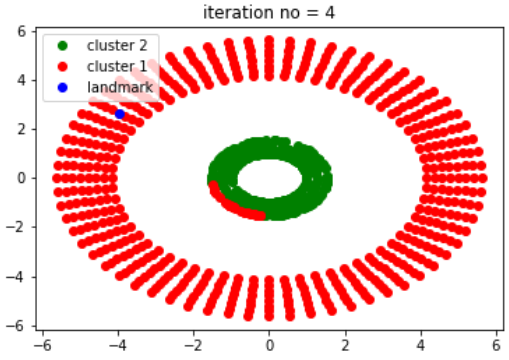
- iteration 0-landmark.png



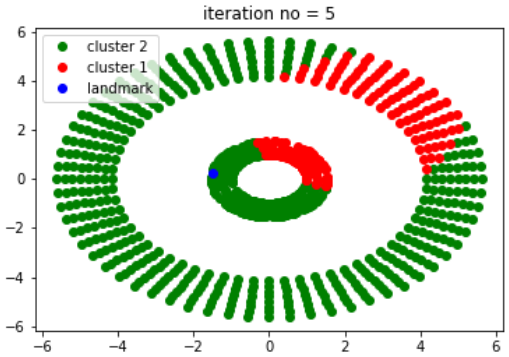
- iteration 1-landmark.png



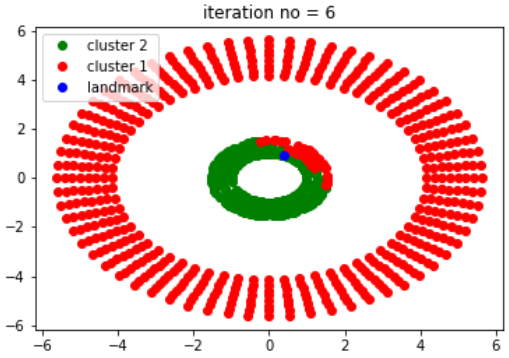
- iteration 2-landmark.png



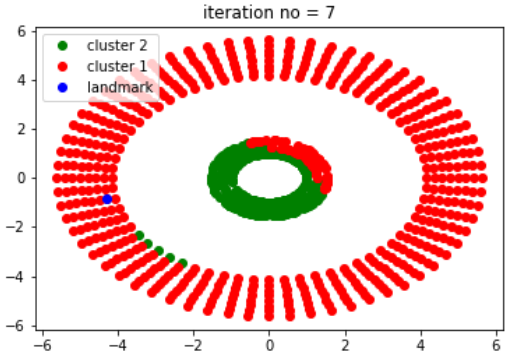
- iteration 3-landmark.png



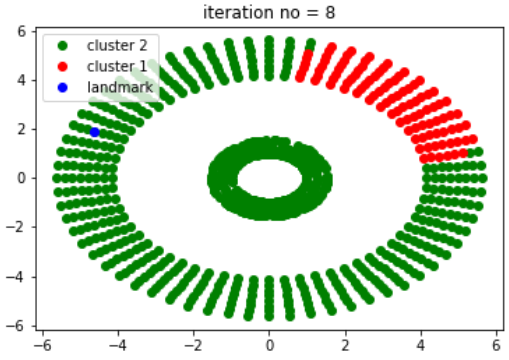
- iteration 4-landmark.png



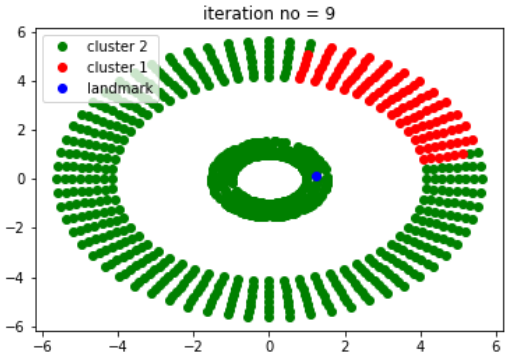
- iteration 5-landmark.png



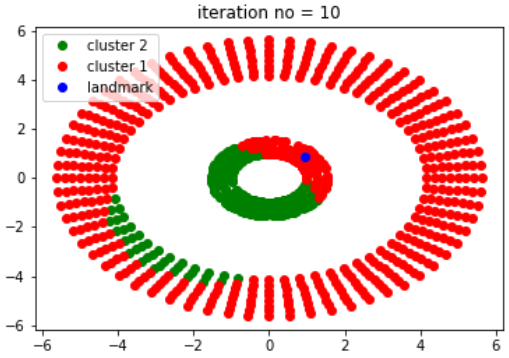
- iteration 6-landmark.png



- iteration 7-landmark.png



- iteration 8-landmark.png



- iteration 9-landmark.png