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

Student Name: Piyush Bagad
Roll Number: 150487
Date: October 31, 2018

QUESTION 1

The soft-margin linear SVM dual objective to compute weights can be stated as

$$\arg \max_{0 \leq \alpha \leq C} f(\alpha)$$

where $f(\alpha) = \alpha^T.\mathbf{1} - \frac{1}{2}\alpha^T \mathbf{G} \alpha$. The gradient-ascent step that we consider for α_n is:

$$\alpha_n = \alpha_n + \delta_*; \quad \text{where} \quad \delta_* := \arg\max_{\delta} f(\alpha + \delta \mathbf{e}_n)$$

First, we will have to compute δ_* in order to apply the ascent step.

$$f(\alpha + \delta \mathbf{e}_n) = (\alpha + \delta \mathbf{e}_n)^T.\mathbf{1} - \frac{1}{2}(\alpha + \delta \mathbf{e}_n)^T \mathbf{G} (\alpha + \delta \mathbf{e}_n)$$

$$\therefore f(\alpha + \delta \mathbf{e}_n) = \alpha^T.\mathbf{1} + \delta \mathbf{e}_n^T.\mathbf{1} - \frac{1}{2}\alpha^T \mathbf{G} \alpha - \frac{1}{2}\alpha^T \mathbf{G} \delta \mathbf{e}_n - \frac{1}{2}\delta \mathbf{e}_n^T \mathbf{G} \alpha - \frac{1}{2}\delta \mathbf{e}_n^T \mathbf{G} \delta \mathbf{e}_n$$

$$f(\alpha + \delta \mathbf{e}_n) = f(\alpha) + \delta(\mathbf{e}_n^T.\mathbf{1} - \alpha^T \mathbf{G} \mathbf{e}_n) - \frac{1}{2}\delta^2 \mathbf{e}_n^T \mathbf{G} \mathbf{e}_n$$

Taking partial derivative w.r.t. δ, and equating it to 0, we get

$$\frac{\partial f(\alpha + \delta \mathbf{e}_n)}{\partial \delta} = (\mathbf{e}_n^T.\mathbf{1} - \alpha^T \mathbf{G} \mathbf{e}_n) - \delta \mathbf{e}_n^T \mathbf{G} \mathbf{e}_n$$

$$\Rightarrow \boxed{\delta_* = \frac{\mathbf{e}_n^T.\mathbf{1} - \alpha^T \mathbf{G} \mathbf{e}_n}{\mathbf{e}_n^T \mathbf{G} \mathbf{e}_n}} \tag{1}$$

Note that we want to ensure that $0 \leq \alpha_n^{(t)} \leq C$, thus if the update value for α_n becomes $\geq C$, we will just set it to C and if it goes less 0, we will just set it to 0. Thus, the update for α_n will become as follows:

$$\alpha_n^{(t)} = \begin{cases} 0, & \text{if } \alpha_n^{(t-1)} + \delta_* < 0 \\ C, & \text{if } \alpha_n^{(t-1)} + \delta_* > C \\ \alpha_n^{(t-1)} + \delta_* & \text{otherwise} \end{cases} \tag{2}$$

Coordinate ascent algorithm sketch

1. Inititalize $\alpha = \alpha^{(0)} = [\alpha_1^{(0)}, .., \alpha_N^{(0)}]$
2. Randomly choose $n \in \{1, 2, .., N\}$. Compute δ_* from equation 1.
3. Update $\alpha_n^{(t)}$ in terms of $\alpha_n^{(t-1)}$ and δ_* as in equation 2.
4. If not converged, go to step 2.

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

Student Name: Piyush Bagad
Roll Number: 150487
Date: October 31, 2018

QUESTION 2

Let $\hat{f}$ be the function learnt by minimizing $\mathcal{L}_W$, which is defined as the sum of squared distances between all pairs of points that are within the same cluster, i.e.,

$$\hat{f} = \arg\min_f \sum_{n,m} \mathbb{I}(f_n = f_m)\|\mathbf{x}_n - \mathbf{x}_m\|^2$$

Note that we can write the indicator function as follows:

$$\mathbb{I}(f_n = f_m) = 1 - \mathbb{I}(f_n \neq f_m)$$

Thus, replacing in the loss function, we obtain

$$\hat{f} = \arg\min_f \left[\sum_{n,m} \|\mathbf{x}_n - \mathbf{x}_m\|^2 - \sum_{n,m} \mathbb{I}(f_n \neq f_m)\|\mathbf{x}_n - \mathbf{x}_m\|^2\right]$$

We can re-write it as follows, since the first term does not involve the argument for minimization. Essentially, it remains to be a constant w.r.t the minimization.

$$\therefore \hat{f} = \sum_{n,m} \|\mathbf{x}_n - \mathbf{x}_m\|^2 - \arg\min_f \sum_{n,m} \mathbb{I}(f_n \neq f_m)\|\mathbf{x}_n - \mathbf{x}_m\|^2$$

$$\therefore \hat{f} = -\arg\min_f \sum_{n,m} \mathbb{I}(f_n \neq f_m)\|\mathbf{x}_n - \mathbf{x}_m\|^2$$

$$\boxed{\therefore \hat{f} = \arg\max_f \sum_{n,m} \mathbb{I}(f_n \neq f_m)\|\mathbf{x}_n - \mathbf{x}_m\|^2}$$

Hence, it turns out to be equivalent to maximizing the inter-cluster distances - distance between points of different clusters.

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

Student Name: Piyush Bagad
Roll Number: 150487
Date: October 31, 2018

QUESTION 3

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N\},\ \Theta = \{\mu, \Sigma\}$ and

$$p(x_n|\Theta) = \mathcal{N}(x_n|\Theta)$$

Part 1: Estimating the conditional given current parameter estimates

Let's say we have Θ as our current parameter estimates. Let us try to estimate the missing part of data for point x_n. We have

$$p([x_n^{obs}, x_n^{miss}]|\Theta) = \mathcal{N}([x_n^{obs}, x_n^{miss}]|\Theta)$$

Based on the missing and observed part of this data point, let us denote the paramters as follows:

$$\mu = \begin{pmatrix}\mu_{miss}\\ \mu_{obs}\end{pmatrix} = \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix},\ \Sigma = \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}$$

Note that this miss-observed split may be different for different examples. This also makes intuitive sense, for example, if for the first example the first two features are missing, then its estimate should be based on the means of these features of other examples and also covariances with other features. For an example with different set of missing features, the estimates will be different. Since a conditional of a gaussian is a gaussian, using result theorem 4.3.1 from MLAPP, we have

$$\boxed{p(x_n^{miss}|x_n^{obs}, \Theta) = \mathcal{N}(x_n^{miss}|x_n^{obs}, \{\mu_{miss|obs}, \Sigma_{miss|obs}\})} \text{ where}$$

$$\mu_{miss|obs} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_n^{obs} - \mu_2)$$

$$\Sigma_{miss|obs} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Part 2: Expected CLL

First, I will compute the CLL as follows and then take the expectation:

$$CLL = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\Theta = \{\mu, \Sigma\})$$

$$CLL = \sum_{n=1}^{N} \log p([\mathbf{x}_n^{obs}, \mathbf{x}_n^{miss}]|\Theta = \{\mu, \Sigma\})$$

$$\mathbb{E}[CLL] = \sum_{n=1}^{N} \mathbb{E}\left[\log p([\mathbf{x}_n^{obs}, \mathbf{x}_n^{miss}]|\mu, \Sigma)\right]$$

$$\boxed{\mathbb{E}[CLL] = \text{constant } + \frac{N}{2}(log|\Sigma|^{-1}) - \frac{1}{2}\text{Trace}(\Sigma^{-1}\mathbb{E}[S])}$$

$$S = \sum_{n=1}^{N} (\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T$$

We will need to compute the expectation of S over the posterior. Note that, we will have the following expectation values:

$$\mathbb{E}[\mathbf{x}_n] = \begin{pmatrix} \mathbb{E}[\mathbf{x}_n^{obs}] \\ \mathbb{E}[\mathbf{x}_n^{miss}] \end{pmatrix} = \begin{pmatrix} \mathbf{x}_n^{obs} \\ \mathbb{E}[\mathbf{x}_n^{miss}] \end{pmatrix} \tag{3}$$

$$\mathbb{E}[\mathbf{x}_n \mathbf{x}_n^T] = \begin{pmatrix} \mathbf{x}_n^{obs}\mathbf{x}_n^{(obs)T} & \mathbf{x}_n^{obs}\mathbb{E}[\mathbf{x}_n^{miss}]^T \\ \mathbb{E}[\mathbf{x}_n^{miss}](\mathbf{x}_n^{obs})^T & \mathbb{E}[\mathbf{x}_n^{miss}(\mathbf{x}_n^{miss})^T] \end{pmatrix} \tag{4}$$

$$\mathbb{E}[S] = \sum_{n=1}^{N} \left\{ \mathbb{E}[\mathbf{x}_n \mathbf{x}_n^T] + \mu\mu^T - 2\mu\mathbb{E}[\mathbf{x}_n]^T \right\} \tag{5}$$

Note that from part 0.2, we have got the conditional distribution for $\mathbf{x}_n^{miss}$, thus we will have the mean $\mathbb{E}[\mathbf{x}_n^{miss}] = \mu_{miss|obs}$. The expected CLL computation is over the conditional distribution. Thus, we can plug the value of $\mathbb{E}[\mathbf{x}_n^{miss}]$ in the above equations to obtain the expected CLL. The only computation that will remain is

$$\mathbb{E}[\mathbf{x}_n^{miss}(\mathbf{x}_n^{miss})^T] = \mathbb{E}[\mathbf{x}_n^{miss}]\mathbb{E}[\mathbf{x}_n^{miss}]^T + cov(\mathbf{x}_n^{miss}) = \mathbb{E}[\mathbf{x}_n^{miss}]\mathbb{E}[\mathbf{x}_n^{miss}]^T + \Sigma_{11}$$

Part 3: Estimation of parameters: M step

Let us denote $\mathbb{E}[CLL] = \mathcal{L}(\mu, \Sigma)$. In order to obtain the updates for these paramteres, we will set the partial derivatives to 0 treating terms involving $\mathbb{E}[\mathbf{x}_n^{miss}]$ as constant.

$$\frac{\partial \mathcal{L}}{\partial \mu} = -\frac{1}{2}(\Sigma^{-1})^T \frac{\partial \mathbb{E}[S]}{\partial \mu}$$

$$\frac{\partial \mathbb{E}[S]}{\partial \mu} = \sum_{n=1}^{N} \left\{ \frac{\partial \mu\mu^T}{\partial \mu} - 2\frac{\partial(\mu\mathbb{E}[\mathbf{x}_n]^T)}{\partial \mu} \right\}$$

$$\frac{\partial \mathbb{E}[S]}{\partial \mu} = N\frac{\partial \mu\mu^T}{\partial \mu} - 2\sum_{n=1}^{N} \frac{\partial([\mu\mathbf{x}_n^{obs(T)} \quad \mu\mathbb{E}[\mathbf{x}_n^{miss}]^T])}{\partial \mu}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma} = \frac{N|\Sigma|}{2}\frac{\partial |\Sigma|^{-1}}{\partial \Sigma} - \frac{1}{2}\frac{\partial \text{Trace}(\Sigma^{-1}\mathbb{E}[S])}{\partial \Sigma}$$

EM Algorithm steps:

1. Initialize $\Theta = \Theta^{(0)} = \{\mu^{(0)}, \Sigma^{(0)}\}$, set $t = 1$
2. **E step**: Now, assuming the values of $\Theta^{(t-1)}$, compute the expected CLL using expression for $\mathbb{E}[S]$ (5) which in turn needs the equations 3 and 4.
3. **M step**: Now, update the values for $\mu^{(t)}$ and $\Sigma^{(t)}$ using the equations —-. Update $t = t + 1$.
4. Repeat steps 2-4 until convergence.

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

Student Name: Piyush Bagad
Roll Number: 150487
Date: October 31, 2018

QUESTION 4

Let us fix the following notation: $X_1 = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, $X_2 = \{x_m\}_{m=N+1}^{N+M}$ and let $Z_2 = \{z_m\}_{m=N+1}^{N+M}$ be the latent variables. The set of parameters is $\Theta = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^{K}$. We assume that all the examples, whether with known labels or not are i.i.d sampled. We have the conditional distribution as follows:

$$p(z_m = k|x_m, X_1, \Theta) = p(z_m = k|x_m, \Theta) \tag{6}$$

$$= \frac{p(x_m|z_m = k, \Theta)p(z_m = k|\Theta)}{p(x_m|\Theta)} \tag{7}$$

$$= \frac{p(x_m|z_m = k, \Theta)p(z_m|\Theta)}{\sum_{l=1}^{K} p(x_m, z_m = l|\Theta)} \tag{8}$$

$$\text{Let } \gamma_{mk} = p(z_m = k|x_m, X_1, \Theta) = \frac{\pi_k \mathcal{N}(x_m|\mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(x_m|\mu_l, \Sigma_l)}$$

$$\boxed{\gamma_{mk} = \frac{\pi_k \mathcal{N}(x_m|\mu_k, \Sigma_k)}{\mathcal{N}\left(\sum_l \pi_l \mu_l, \right)}} \tag{9}$$

Expected CLL

Let us fix the notation to be (as usual): $z_{mk} = 1$ iff $z_m = k$ and $y_{nk} = 1$ iff $y_n = k$. Now, the CLL will be computed as follows, note the assumption of IID.

$$CLL = \log p(X_1, (X_2, Z_2)|\Theta) = \log p(X_2, Z_2|X_1, \Theta) + \log p(X_1|\Theta)$$

$$= \log p(X_2, Z_2|\Theta) + \log p(X_1|\Theta)$$

Now, we have

$$\log p(X_1|\Theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \left(\log \pi_k + \log \mathcal{N}(x_n|y_n = k, \mu_k, \Sigma_k)\right)$$

$$\log p(X_2, Z_2|\Theta) = \sum_{m=N+1}^{N+M}\sum_{k=1}^{K} z_{mk} \left(\log \pi_k + \log \mathcal{N}(x_m|z_m = k, \mu_k, \Sigma_k)\right)$$

Now, notice that for computing the expected CLL, we will need the expectation over the posterior: $\mathbb{E}[z_{nk}] = 1 * p(z_{nk} = 1) + 0 * p(z_{nk} = 0) = p(z_{nk} = 1) = \gamma_{mk}$.

$$\mathbb{E}[CLL] = \sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \left(\log \pi_k + \log \mathcal{N}(x_n|y_n = k, \mu_k, \Sigma_k)\right) + \tag{10}$$

$$\sum_{m=N+1}^{N+M}\sum_{k=1}^{K} \gamma_{mk} \left(\log \pi_k + \log \mathcal{N}(x_m|z_m = k, \mu_k, \Sigma_k)\right) \tag{11}$$

Updating the paramters

We have the following objective.

$$\hat{\Theta} = \arg\max_{\Theta} \mathbb{E}[CLL]$$

It basically boils down to solving the problem of gaussian generative classification with known labels:

$$\mathbb{E}[CLL] = \sum_{n=1}^{N+M} \sum_{k=1}^{K} t_{nk} \left(\log \pi_k + \log \mathcal{N}(x_n|\mu_k, \Sigma_k)\right)$$

where $t_{nk} = y_{nk}, \forall n \in \{1, 2.., N\}$ and $t_{nk} = \gamma_{nk}, \forall n \in \{N+1, N+2.., N+M\}$. With Ref: Lecture 16, Slide 19, the parameter updates will become:

$$\hat{\pi}_k = \frac{\sum_{n=1}^{N+M} t_{nk}}{N+M} = \frac{\sum_{n=1}^{N} y_{nk} + \sum_{m=N+1}^{N+M} \gamma_{nk}}{N+M} = \frac{N_k + M_k}{N+M} \tag{12}$$

$$\hat{\mu}_k = \frac{1}{N_k + M_k} \sum_{n=1}^{N+M} t_{nk} \mathbf{x}_n \tag{13}$$

$$\hat{\Sigma}_k = \frac{1}{N_k + M_k} \sum_{n=1}^{N+M} t_{nk} (\mathbf{x}_n - \hat{\mu}_k)(\mathbf{x}_n - \hat{\mu}_k)^T \tag{14}$$

The EM algorithm

1. Initialize $\Theta = \Theta^{(0)} = \{\pi_k^{(0)}, \mu_k^{(0)}, \Sigma_k^{(0)}\}_k$, set $t = 1$
2. **E step**: Now, assuming the values of $\Theta^{(t-1)}$, compute the expected CLL from equation 10 using expression for $\mathbb{E}[z_{mk}] = \gamma_{mk}$ (See 9).
3. **M step**: Now, update the values for $\pi_k^{(t)}, \mu_k^{(t)}$ and $\Sigma_k^{(t)}$ using the equations 12, 13 and 14 respectively. Update $t = t + 1$.
4. Repeat steps 2-4 until convergence.

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

Student Name: Piyush Bagad
Roll Number: 150487
Date: October 31, 2018

QUESTION 5

Let the observed data be $\{(\mathbf{x}_n, y_x)\}_{n=1}^N$ and latent variables $Z = \{z_1, z_2, .., z_K\}$. The global set of parameters be given by $\Theta = \{(\mathbf{w}_1, ..., \mathbf{w}_K), (\pi_1, .., \pi_K)\}$, we have

$$z_n \sim \text{multinoulli}(\pi_1, .., \pi_K)$$

$$y_n \sim \mathcal{N}(\mathbf{w}_{z_n}^T \mathbf{x}_n, \beta^{-1})$$

Part 1: Mixture of regression models

The model is basically fitting multiple regression models $\mathbf{w}_{z_n}$ on the input data. It is similar to doing multi-output regression. In simple probabilistic linear regression, we assume that all of the data points is sampled i.i.d. from a linear space. However, as shown in figure 1, for such kind of data, mixture of multiple regression models can help.

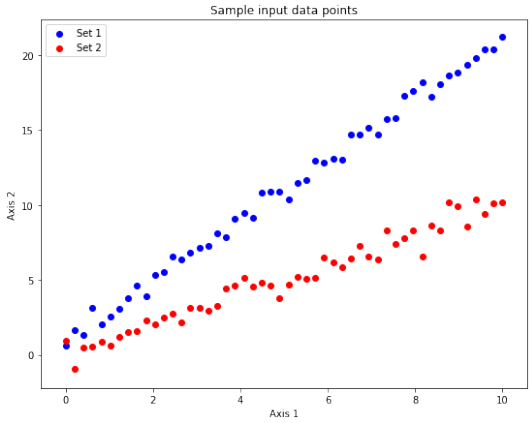


Figure 1: If the data looks somewhat like this, using multiple mixture of linear regressions would be a better idea.

Part 2: ALT-OPT Algorithm

Estimating the latent variables fixing the parameters

Suppose we have the optimal value of $\Theta = \hat{\Theta}$, then we can estimate z_n as follows:

$$\hat{z_n} = \arg\max_k p(z_n = k|y_n, \mathbf{x}_n, \hat{\Theta}) = \arg\max_k \frac{p(y_n, z_n = k|\mathbf{x}_n, \hat{\Theta})}{p(y_n|\mathbf{x}_n, \hat{\Theta})} \tag{15}$$

$$= \arg\max_k \frac{p(y_n|z_n = k, \mathbf{x}_n, \hat{\Theta})p(z_n = k|\hat{\Theta})}{\sum_{l=1}^K p(y_n|z_n = l, \mathbf{x}_n, \hat{\Theta})p(z_n = l|\hat{\Theta})} \tag{16}$$

$$= \arg\max_k \frac{\pi_k \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^K \pi_l \mathcal{N}(\mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})} \tag{17}$$

Let us denote the term in equation 17 as γ_{nk}. Notice that the denominator does not depend on k and can be dropped, but I am keeping it since the expression takes form of probability. It will not matter for the maximization. Also, let's use the notation that $z_{nk} = 1$ *iff* $z_n = k$.

Estimating the parameters fixing the latent variables

Once we pick $\hat{z}_n$'s that maximize γ_{nk}'s, we can now keep them fixed and estimate parameters.

$$\hat{\Theta} = \arg\max_{\Theta} \mathcal{L}(\Theta, \{\hat{z}_n\}) = \arg\max_{\Theta} \sum_{n=1}^{N} \log p(y_n, \hat{z}_n | \mathbf{x}_n, \Theta)$$

$$\hat{\Theta} = \arg\max_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk} \left[\log \pi_k + \log \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})\right]$$

Note that we will also have to incorporate the constraint $\sum_k \pi_k = 1$. We can then solve by constrained optimization. For π_k, since the term involving it is not coupled with any of the other paramters, we can solve for it independently. The problem for finding π_k is exactly the same as that in generative classification with known labels (Ref: Lecture 15, Slide 9), so I will use that result directly.

$$\boxed{\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \hat{z}_{nk} = \frac{N_k}{N}}$$

Now, to estimate the weights $\mathbf{w}_k$'s, we have

$$\{\hat{\mathbf{w}}_k\}_{k=1}^{K} = \arg \max_{\mathbf{w}_1,..,\mathbf{w}_K} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk} (\mathbf{w}_k^T \mathbf{x}_n - y_n)^2$$

$$\{\hat{\mathbf{w}}_k\}_{k=1}^{K} = \arg \max_{\mathbf{w}_1,..,\mathbf{w}_K} \sum_{k=1}^{K} (\mathbf{y} - \mathbf{X}\mathbf{w}_k)^T Z_k (\mathbf{y} - \mathbf{X}\mathbf{w}_k) \text{ where } Z_k = \text{Diag}(z_{1k}, .., z_{Nk})$$

Note that these terms can be maximized independently since they do not depend upon each other. Thus, we get

$$\hat{\mathbf{w}}_k = \arg\max_{\mathbf{w}_k} (\mathbf{y} - \mathbf{X}\mathbf{w}_k)^T Z_k (\mathbf{y} - \mathbf{X}\mathbf{w}_k)$$

This is the same as solving a linear regression problem with examples being weighted by $\{z_{nk}\}$'s. Thus, the solution (Ref: Practice Problem Set 1, Q.5) will be as follows:

$$\boxed{\hat{\mathbf{w}}_k = [\mathbf{X}^T Z_k \mathbf{X}] \mathbf{X}^T Z_k \mathbf{y}}$$

A Special case

For the case when $\pi_k = (1/K),\ \forall k = 1, 2, ..K$ implies that

$$\hat{z}_n = \arg\max_k \delta_{nk} = \arg\max_k \frac{\mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \mathcal{N}(\mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}$$

This kind of an update ignores the number of points that can be explained by a single model $\mathbf{w}_k$. Thus, it will only consider the probability of x_n explained by these different models giving them equal weightage.

Part 3A: EM Algorithm

A. Computing the posterior

We have already computed the conditional distribution for the latent variables in equation 17.

$$\boxed{p(z_n = k|y_n, \mathbf{x}_n, \Theta) = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}}$$

From the discrete distribution, we have the expectation as simply:

$$\mathbb{E}[z_n] = \sum_{k=1}^{K} k p(z_n = k|y_n, \mathbf{x}_n, \Theta) = \sum_{k=1}^{K} k\gamma_{nk}$$

$$\mathbb{E}[z_{nk}] = 1 * p(z_{nk} = 1) + 0 * p(z_{nk} = 0) = p(z_n = k) = \gamma_{nk} \tag{18}$$

B. Computing the expected CLL

Now, let us compute the expected CLL.

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

$$\mathbb{E}[CLL] = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}] \left[\log \pi_k + \log \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})\right] \tag{19}$$

We can get $\mathbb{E}[z_{nk}]$ from equation 18, and thus the expected CLL. The rest of the procedure for maximizing the expected CLL will remain the same as in previous part except that instead of z_{nk}, we will be using γ_{nk}. Thus, we have

$$\boxed{\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}} \tag{20}$$

$$\boxed{\hat{\mathbf{w}}_k = [\mathbf{X}^T \Gamma_k \mathbf{X}] \mathbf{X}^T \Gamma_k \mathbf{y}} \tag{21}$$

where $\Gamma_k = \text{Diag}(\gamma_{1k}, \gamma_{2k}, .., \gamma_{nk})$.

C. The Algorithm

1. Initialize $\Theta = \Theta^{(0)} = \{\pi_k^{(0)}, \mathbf{w}_k^{(0)}\}$, set $t = 1$
2. **E step**: Now, assuming the values of $\Theta^{(t-1)}$, compute the expected CLL from equation 19 using expression for $\mathbb{E}[z_{nk}] = \gamma_{nk}$.
3. **M step**: Now, update the values for $\pi_k^{(t)}, \mathbf{w}_k^{(t)}$ using the equations 20, 21. Update $t = t+1$.
4. Repeat steps 2-4 until convergence.

Part 3B: Particular case

We have to show that when $\beta \to \infty$, the EM algorithm boils down to the ALT-OPT algorithm. Basically, we want to show the following:

$$\gamma_{nk} = 1 \ \textit{iff} \ k_* = \arg\max_k \frac{\pi_k \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_n, \beta^{-1})}{\sum_{l=1}^K \pi_l \mathcal{N}(\mathbf{w}_l^T \mathbf{x}_n, \beta^{-1})}$$

We have

$$\gamma_{nk}(\beta) = \frac{\pi_k \exp\left[-\beta^2 \|y_n - \mathbf{w}_k^T \mathbf{x}_n\|^2\right]}{\sum_{l=1}^K \pi_l \exp\left[-\beta^2 \|y_n - \mathbf{w}_l^T \mathbf{x}_n\|^2\right]}$$

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

Now, consider the case when $k \neq k_*$ as defined above, the term $\|y_n - \mathbf{w}_l^T \mathbf{x}_n\|^2 - \|y_n - \mathbf{w}_k^T \mathbf{x}_n\|^2 < 0, \forall l = k_*$, thus the exponent becomes ∞ making $\gamma_{nk} \to 0$. For $k = k_*$, clearly, all other terms will go to 0, and thus $\gamma_{nk_*} = 1$. Thus, the claim is proved and hence we will have

$$\mathbb{E}[z_{nk}] = 1 \text{ only when } \hat{z}_{nk} = 1$$

This implies that the expectation step becomes the same as that of ALT-OPT and so does maximization since the expected likelihood will be the same that the paramtere update step in ALT-OPT.

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

Student Name: Piyush Bagad
Roll Number: 150487
Date: October 31, 2018

QUESTION
6

Programming Problem 1

1. Kernel Ridge Regression

Observations: In Kernel ridge regression, an important observation was that the mean squared error increases on increasing λ. This is expected since lower λ implies overfitting while increasing λ means too simple model and thus underfits. Refer to figures 2a, 2b, 3a and 3b for plots and table **??** for mean squared error.

λ	**Mean Squared Error**
0.1	0.00107
1	0.02944
10	0.37461
100	0.83259

Table 1: As we increase the regularization hyperparameter λ, the mean squared error increases. Clearly, as evident from the plots as well, $\lambda = 0.1$ coressponds to overfit model whereas $\lambda = 100$ corressponds to an underfit model. The model with $\lambda = 1$ seems to be appropriate.

2. Landmark-ridge Regression

Observations: The key observation is that the mean squared error decreases as we increase L upto a point and then remains more or less the same (look at the change from $L = 50$ to $L = 100$). Using too low value of L means that the model is based upon information of a very few data points and these datapoints may not capture all the characteristics of the global dataset. For an extremely high value of L, it will be almost similar to using all the points in the dataset, also, it is important to notice that increasing L implies higher cost of computations since the dimension of feature space increases. Refer to figures 4a, 4b, 5a, 5b and 5c for plots and table **??** for mean squared error.

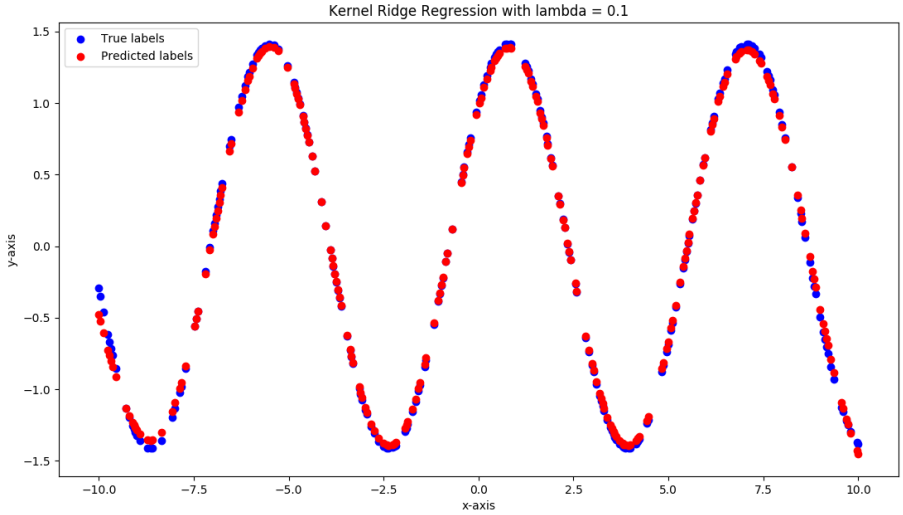
Programming Problem 2

1. Using Hand-crafted Features

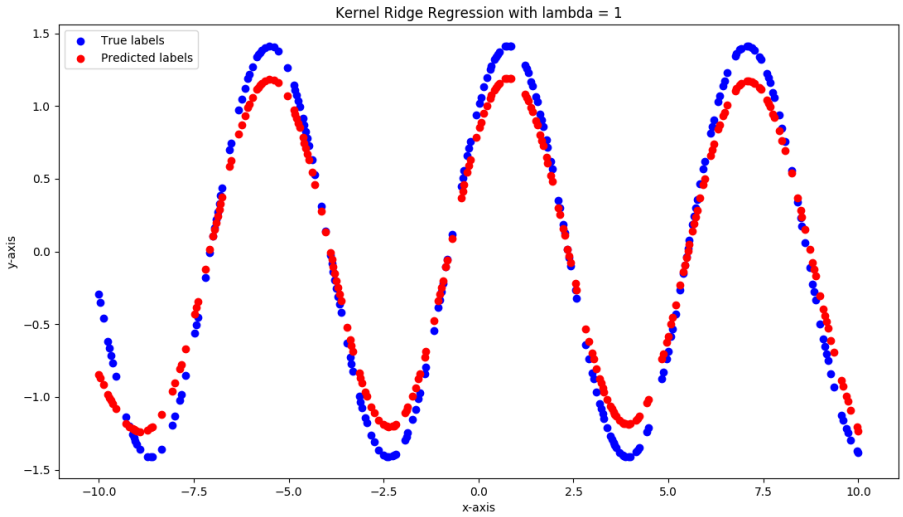
I have used the following three-dimensional feature transformation for clustering:

$$f : \mathbb{R}^2 \to \mathbb{R}^3 \Rightarrow f(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$$

I wanted a feature transformation that will split the dataset into two clusters and that the two transformed clusters will be smooth and seperated. From the data visualization, it seemed that



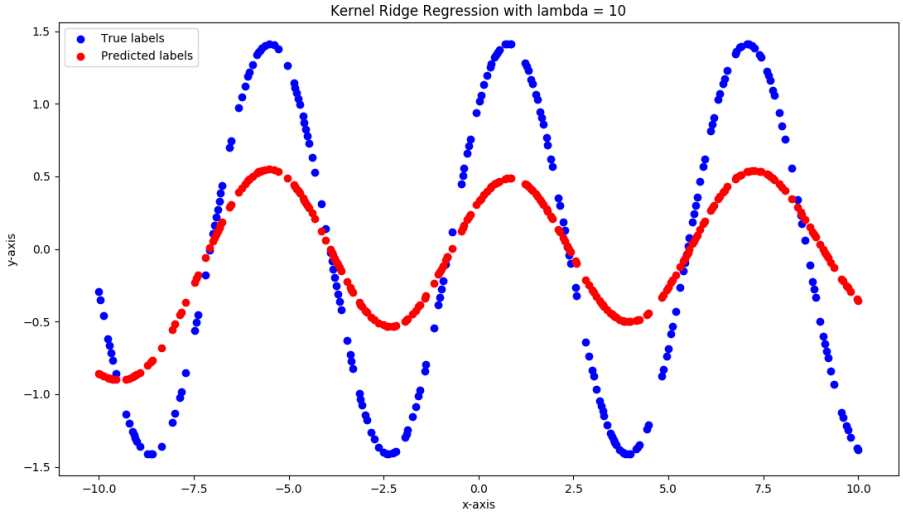
(a) 1a



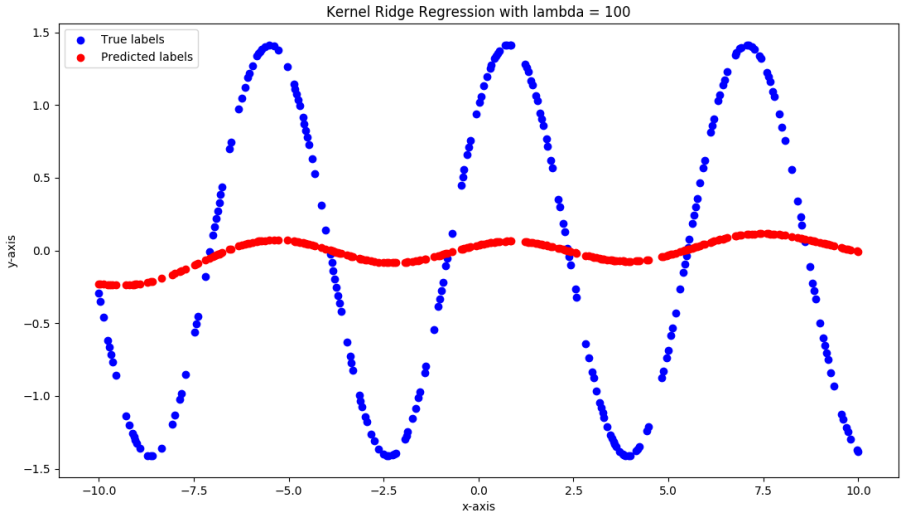
(b) 1b

L	**Mean Squared Error**
2	0.95181
5	0.39178
20	0.00905
50	0.00142
100	0.00182

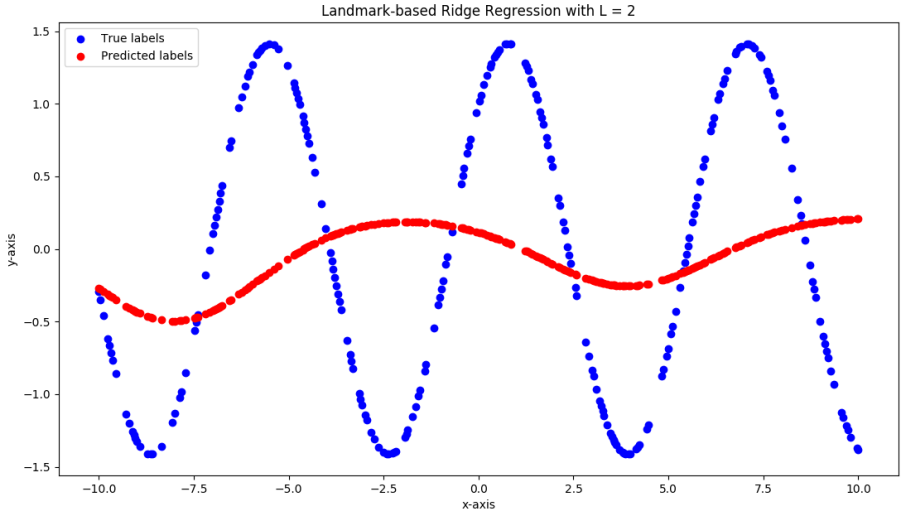
Table 2: As we increase the value of L, the mean squared error decreases. Clearly, as evident from the plots as well, $L = 100$ coressponds to perfectly fit model whereas $L = 1$ corressponds to an excessively underfit model. The model with $L = 50$ seems to be a pretty good approximation.


Kernel Ridge Regression with lambda = 10
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

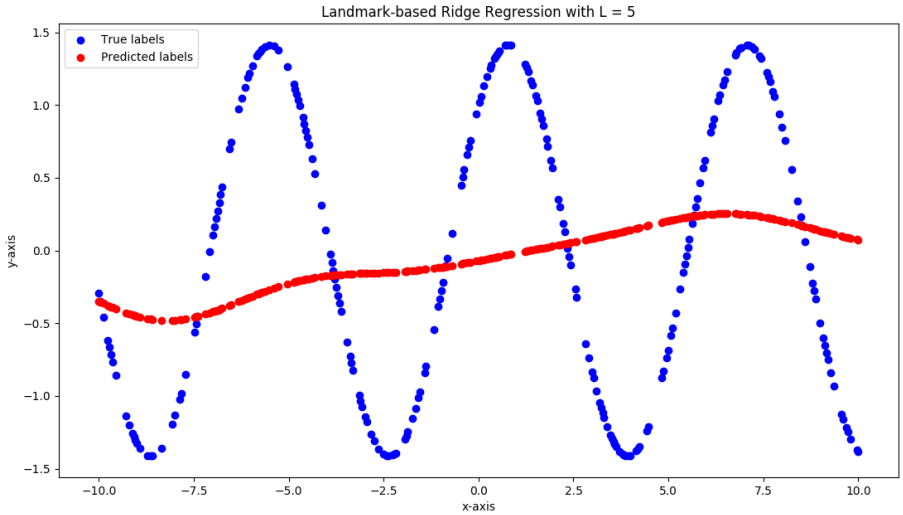
(a) 1c


Kernel Ridge Regression with lambda = 100
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

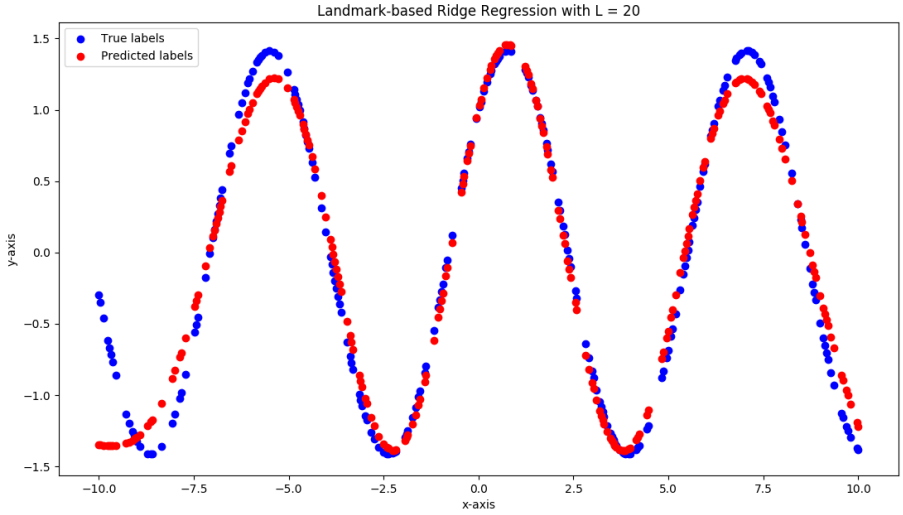
(b) 1d


Landmark-based Ridge Regression with L = 2
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

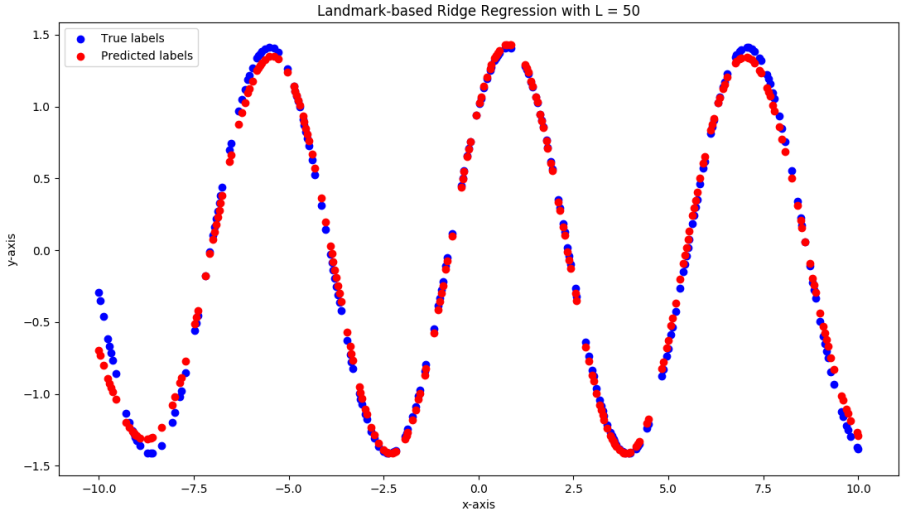
(a) 1a


Landmark-based Ridge Regression with L = 5
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

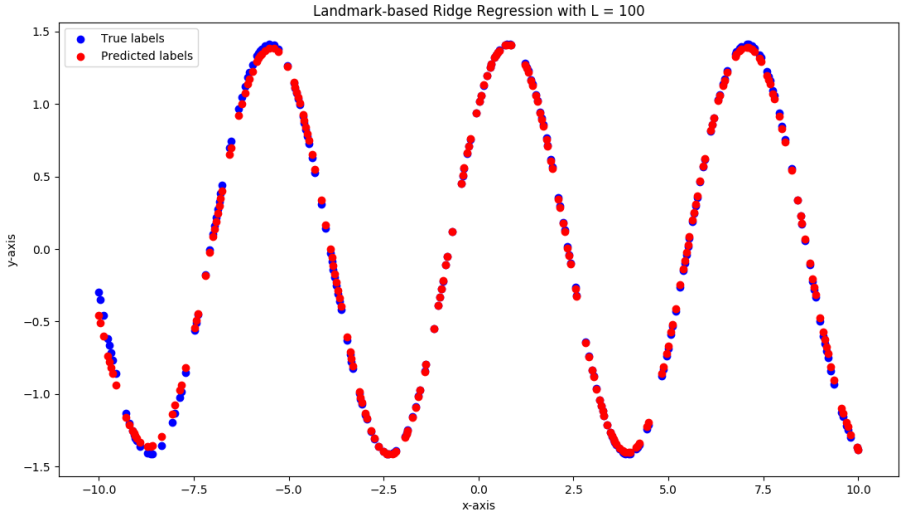
(b) 1b


Landmark-based Ridge Regression with L = 20
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

(a) 1c

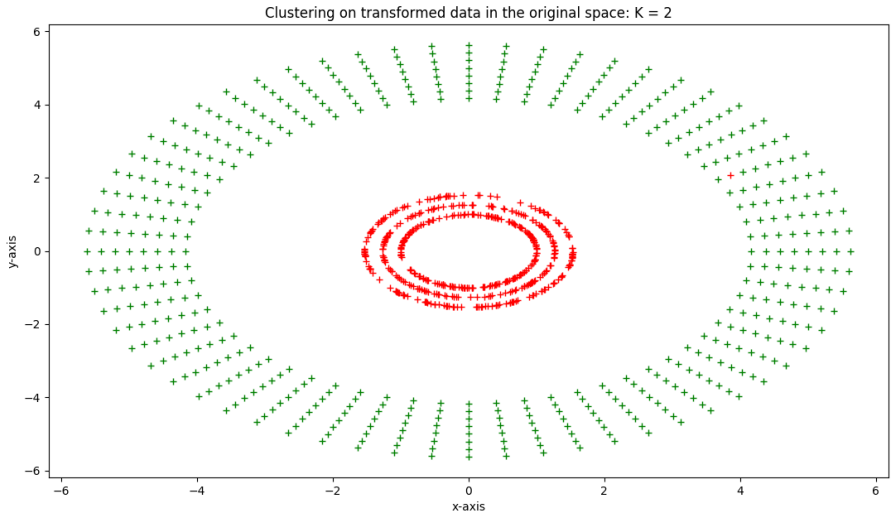

Landmark-based Ridge Regression with L = 50
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

(b) 1d


Landmark-based Ridge Regression with L = 100
True labels
Predicted labels
y-axis
x-axis
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

(c) 2e

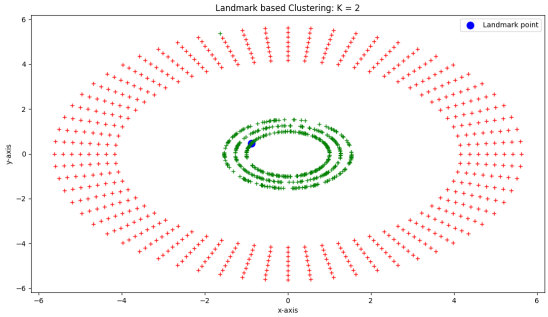
the invariant binding each of the clusters was the inner and outer radius os the two concentric clusters. Refer to figure 6a for visual results.



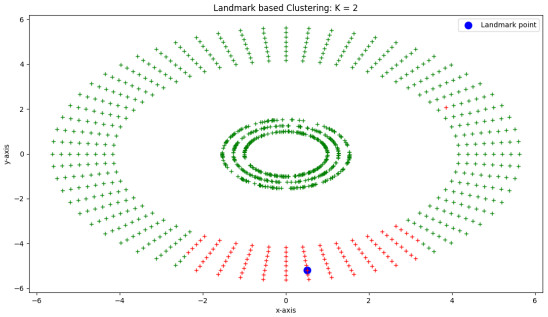
(a) Dataset in original space with clusters

2. Using Kernels

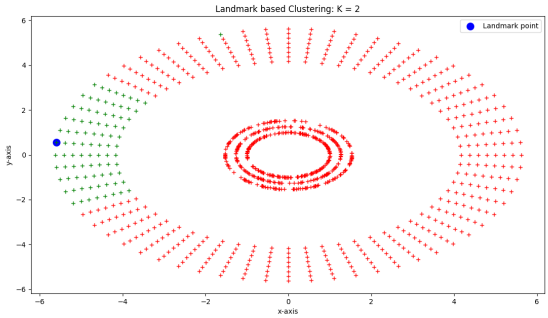
Using landmark based approach with RBF kernels, I observed that the clustering is very sensitive to the choice of landmark ($L = 1$) point. The clustering is such that one of the clusters is centered around the landmark point and the rest is the second cluster. A good cluster may be obtained if the randomly chosen landmark point lands into the innermost part of the inner cluster. Note that this is expected since clustering assumes equal spread of clusters and spherical clusters. thus if the landmark point lies in inner cluster, one of the clusters centers around it (since, the new feature space is based on distance from the landmark) and the rest becomes the other cluster. Refer to following figures for results.


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

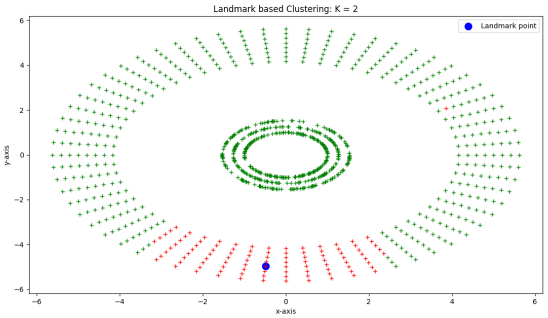
(a)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

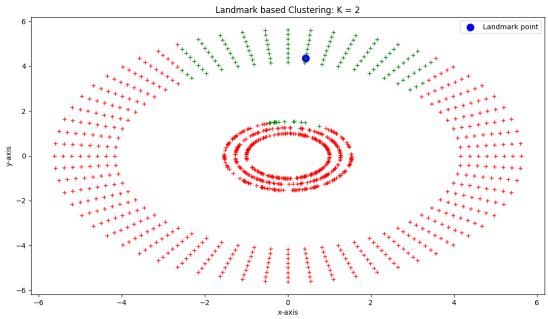
(b)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

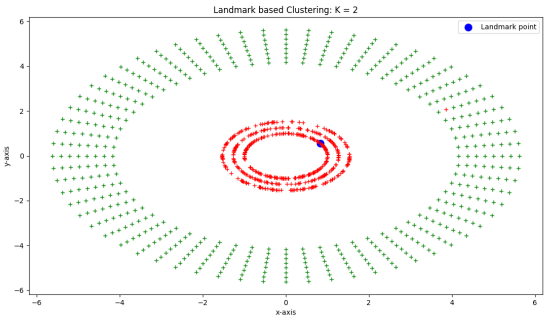
(c)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

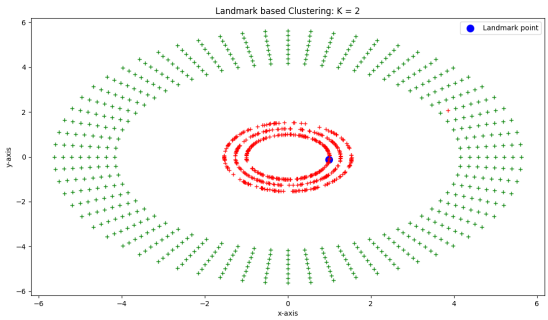
(d)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

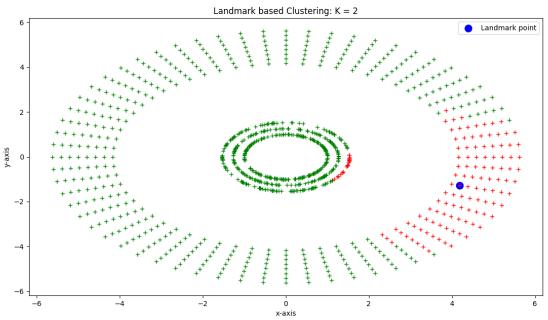
(e)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

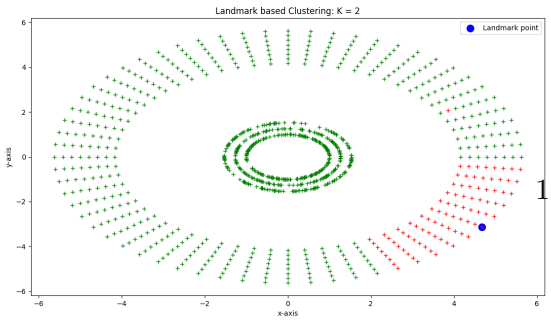
(f)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

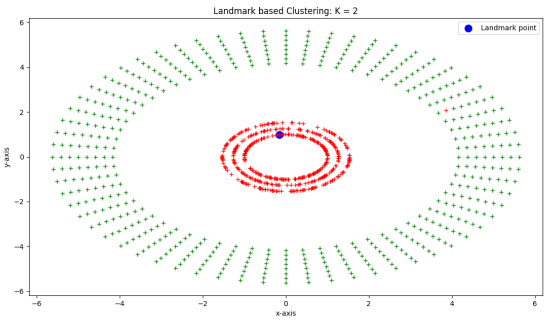
(g)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

(h)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

(i)


Landmark based Clustering: K = 2
Landmark point
x-axis
y-axis

(j)