# 1 Clustering

*[25 points]*

## a)

*[10 points]*

First, note that $r_{nk}$ is simply an indicator function for the $k^{th}$ cluster (if k is fixed). Thus, for any cluster with index $k$, the loss function is

$$D_k = \sum_{i=1}^{N} r_{nk} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2$$

where $D = \sum_k^K D_k$. Assuming the labels are fixed, $D_k$ is convex. Thus, we take the derivative with respect to $\mu_k$ and set it to zero.

$$\begin{aligned}\nabla_{\mu_k} D_k &= \sum_{i=1}^{N} r_{nk} 2(\boldsymbol{x}_i - \boldsymbol{\mu}_k) = 0 \\ \sum_{i=1}^{N} r_{nk}\boldsymbol{\mu}_k &= \sum_{i=1}^{N} r_{nk}\boldsymbol{x}_i \\ \boldsymbol{\mu}_k &= \frac{1}{\sum_{i=1}^{N} r_{nk}} \sum_{i=1}^{N} r_{nk}\boldsymbol{x}_i\end{aligned}$$

This is exactly the mean of the data points that are in cluster $k$.

## b)

*[5 points]*

Using the $L_1$ loss, the objective function becomes separable by coordinate. More specifically, for each $k = 1, \ldots, K$, we have that

$$\min_{\boldsymbol{\mu}_k} D_k = \min_{\boldsymbol{\mu}_k} \sum_{i=1}^{N} r_{nk} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_1 = \min_{\{\mu_{k,j}\}_j^P} \sum_{j=1}^{P}\sum_{i=1}^{N} r_{nk}|x_{i,j} - \mu_{k,j}|$$

This is exactly $P$ one dimensional sub-problems of the form

$$\min_{\mu_{k,j}} \sum_{i=1}^{N} r_{nk}|x_{i,j} - \mu_{k,j}|$$

which is exactly the $L_1$ norm for a vector in $\mathbb{R}^N$.

**Proof by contradiction, Preferred approach** We need to prove that the median minimizes the $L_1$ norm distortion function. To simplify the notation, consider $n$ data points $y_1, \ldots, y_n$ with the median $m$. Suppose that the minimizer of the $L_1$ distortion is at another point $m'$ which is not a median, i.e., the number of data points that are less than $m'$ is not equal to the number of data points that are greater than it. Without loss of generality, let's assume that there are $K_l$ data points smaller than $m'$ and $K_r$ data points in larger than $m'$ and $K_l > K_r$. We can see that moving $m'$ to $m$ (which is smaller than $m$) will decrease the loss by $(K_l - K_r)|m' - m|$ which means $m'$ is not the minimizer of the objective function. This is a contradiction and the proof is complete. Thus, we showed that unless $K_r = K_l$, $m'$ cannot be the minimizer of the $L_1$ distortion loss.

**Proof by subgradients, Less preferred approach** Norms are convex. Thus, this is a convex problem, and the solution must have a subgradient containing zero.

$$\partial \sum_{i=1}^{N} r_{nk}|x_{i,j} - \mu_{k,j}| = \sum_{i=1}^{N} \partial |x_{i,j} - \mu_{k,j}| = \sum_{i=1}^{N} \operatorname{sign}(x_{i,j} - \mu_{k,j})$$

Supposing that there are an even number of datapoints, the subgradient contains zero in between the rank order $\frac{P-1}{2}$ and $\frac{P}{2}$ points, i.e. at a point where there are equal number of $x_{i,j}$ on either side of our chosen point. However, we still must choose which point inbetween these two points. Noting that there are equal numbers of points on either side of this interval, we can choose any point! This is because for any $\epsilon$ perturbation of our chosen point, so long as we do not move outside of the interval, the $x_{i,j}$ that are now further away increase the objective by $\frac{P\epsilon}{2}$ but the points that are closer reduce it by exactly the same amount.

However, for an odd number of points, the point at which there are equal number of points on either size of $\mu_{k,j}$ is exactly the median point by definition.

## c)

*[10 points]*

(1) *[4 points]* First, we can represent the center of the cluster as follows:

$$\tilde{\boldsymbol{\mu}}_k = \frac{\sum_{n=1}^{N} r_{nk}\phi(\boldsymbol{x}_n)}{\sum_{n=1}^{N} r_{nk}}$$

With this result, we can write $\tilde{D}$ as

$$\tilde{D} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||\phi(\boldsymbol{x}_n) - \frac{\sum_{i=1}^{N} r_{ik}\phi(\boldsymbol{x_i})}{\sum_{i=1}^{N} r_{ik}}||_2^2$$

Let $R_k = \{\boldsymbol{x}_n | r_{nk} = 1\}$, we have

$$\begin{aligned} ||\phi(\boldsymbol{x}_n) - \frac{\sum_{\boldsymbol{x}\in R_k}\phi(\boldsymbol{x})}{|R_k|}||_2^2 &= ||\phi(\boldsymbol{x}_n)||_2^2 - 2\frac{\sum_{\boldsymbol{x}\in R_k}\phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{x}_n)}{|R_k|} + \frac{1}{|R_k|^2}||\sum_{\boldsymbol{x}\in R_k}\phi(\boldsymbol{x})||_2^2 \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) - 2\frac{\sum_{\boldsymbol{x}\in R_k} k(\boldsymbol{x}, \boldsymbol{x}_n)}{|R_k|} + \frac{1}{|R_k|^2}\sum_{\boldsymbol{x}_i\in R_k}\sum_{\boldsymbol{x}_j\in R_k} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned}$$

(2) *[4 points]* Assume that in previous round, we have the cluster center

$$\tilde{\boldsymbol{\mu}}_k = \frac{\sum_{n=1}^{N} r_{nk}\phi(\boldsymbol{x}_n)}{\sum_{n=1}^{N} r_{nk}}.$$

For a data point $\boldsymbol{x_n}$, we update its cluster assignment to $k$ such that

$$k = \arg\min_k \left[ k(\boldsymbol{x}_n, \boldsymbol{x}_n) - 2\frac{\sum_{\boldsymbol{x}\in R_k} k(\boldsymbol{x}, \boldsymbol{x}_n)}{|R_k|} + \frac{1}{|R_k|^2}\sum_{\boldsymbol{x}_i\in R_k}\sum_{\boldsymbol{x}_j\in R_k} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \right]$$

(3) *[2 points]* Here is the pseudo code for the Kernel K-means algorithm:

```
• Randomly choose $K$ points $x_1, \ldots, x_K$ and assume $\phi(\boldsymbol{x}_i)$ as the center for each cluster.
• Loop until convergence:
    – Assign each data point to its nearest cluster center by the question above.
    – Update cluster centers by tracking $R_k = \{\boldsymbol{x}_n | r_{nk} = 1\}$.
```

## 2 Gaussian Mixture Model

*[10 points]*
We can write the likelihood function as

$$\begin{aligned} L(\alpha) &= p(x_1|\alpha) \\ &= \frac{\alpha}{\sqrt{2\pi}}\exp(-\frac{1}{2}x^2) + \frac{1-\alpha}{\sqrt{\pi}}\exp(-x^2) \\ &= [\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}x^2) - \frac{1}{\sqrt{\pi}}\exp(-x_1^2)]\alpha + \frac{1}{\sqrt{\pi}}\exp(-x_1^2) \end{aligned} \tag{1}$$

*[3 points]* for the likelihood function.

Thus, we see that the likelihood is simply a linear function of *alpha* where the sign of the slope is determined by which Gaussian produces the larger response. Since we know that $0 \le \alpha \le 1$, this tells us that if the slope is positive that we should choose $\alpha = 1$ and otherwise if the slope is negative we should choose $\alpha = 0$. Using straightforward algebra one can show that the slope is positive whenever $x_1^2 \ge \log 2$ and we should set $\alpha = 1$; otherwise set $\alpha = 0$.

(Alternatively, one could also apply Expectation maximization for this problem (not an efficient solution). Starting with $\alpha = 0.5$ and applying EM, you would observe that in each iteration, your estimate of $\alpha$ will strictly increase or decrease depends on which of the two Gaussians fit $x_1$ better, eventually lead to 1 or 0 accordingly.)
*[7 points]* for the estimation of $\alpha$.

## 3 EM algorithm

*[15 points]*

(a) *[5 points]* We can think that the data are generated as a mixture as a singleton 0 and Poisson distribution with parameter $\lambda$. As a result, we define one hidden variable $z_i$ for each $x_i = 0$.

$$z_i = \begin{cases} 0, \text{if } x_i \text{ is generated from Poisson} \\ 1, \text{if } x_i \text{ is generated from singeton 0} \end{cases}$$

The complete log likelihood is

$$\log P(X, Z|\lambda, \pi) = \log\left[\prod_{i:x_i>0}(1-\pi)\frac{\lambda^{x_i}e^{-\lambda}}{x_i!}\cdot\prod_{i:x_i=0}\pi^{z_i}[(1-\pi)e^{-\lambda}]^{1-z_i}\right]$$

(b) *[10 points]* For the E-step, we need to compute the posterior distribution on the hidden variables $z_i$.

$$p(z_i = 0|X, \lambda, \pi) = \frac{(1-\pi)e^{-\lambda}}{\pi + (1-\pi)e^{-\lambda}} = \gamma_{i0}$$

$$p(z_i = 1|X, \lambda, \pi) = \frac{\pi}{\pi + (1-\pi)e^{-\lambda}} = \gamma_{i1}$$

For the M-step, we need to compute the $Q(\theta, \theta^{\text{old}})$ function, where $\theta = (\lambda, \pi)$.

$$\begin{aligned} Q(\theta, \theta^{\text{old}}) &= E_{p(Z|X,\lambda^{\text{old}},\pi^{\text{old}})}[\log P(X, Z|\lambda, \pi)] \\ &= \sum_{i:x_i>0}[\log(1-\pi) + x_i\log\lambda - \lambda] + \sum_{i:x_i=0}[E[z_i]\log\pi + E[1-z_i][\log(1-\pi) - \lambda]] + C \\ &= \sum_{i:x_i>0}[\log(1-\pi) + x_i\log\lambda - \lambda] + \sum_{i:x_i=0}[\gamma_{i1}\log\pi + \gamma_{i0}[\log(1-\pi) - \lambda]] + C \end{aligned}$$

*[5 points]* for the expression of $Q$ function.

To get the update equation, we take derivative of $L$ with respect to $\lambda$ and $\pi$. Let $N_{\sim 0} = \sum_{i=1}^{N} I[x_i > 0]$ and set it to zero.

$$\frac{\partial Q}{\partial \pi} = -\frac{\sum_{i=1}^{N} I[x_i > 0]}{1-\pi} - \frac{\sum_{i:x_i=0} \gamma_{i0}}{1-\pi} + \frac{\sum_{i:x_i=0} \gamma_{i1}}{\pi} = 0.$$

$$\pi^{\text{new}} = \frac{\sum_{i:x_i=0} \gamma_{i1}}{\sum_{i:x_i=0} \gamma_{i0} + \sum_{i:x_i=0} \gamma_{i1} + \sum_{i=1}^{N} I[x_i > 0]} = \frac{\sum_{i:x_i=0} \gamma_{i1}}{N}$$

$$\frac{\partial Q}{\partial \lambda} = \sum_{i:x_i>0} \left[\frac{x_i}{\lambda} - 1\right] - \sum_{i:x_i=0} \gamma_{i0} = 0$$

$$\lambda^{\text{new}} = \frac{\sum_{i:x_i>0} x_i}{\sum_i I[x_i > 0] + \sum_{i:x_i=0} \gamma_{i0}} = \frac{\sum_{i=1}^{N} x_i}{\sum_i I[x_i > 0] + \sum_{i:x_i=0} \gamma_{i0}}$$

*[5 points]* for the update equation of $\pi$ and $\lambda$.

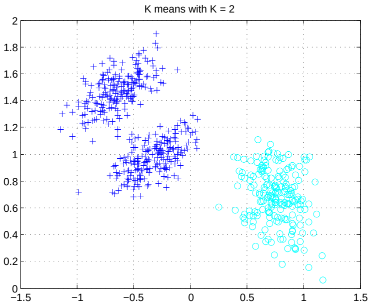
# 4 Programming

*[50 points]*

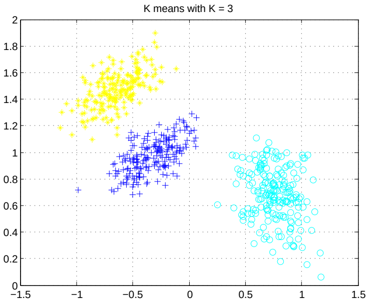
**4.2***[15 points]*

**a)***[10 points]*

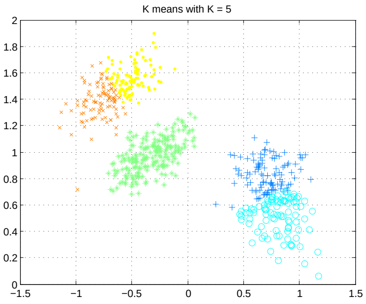
The linear K-means results for the blob.csv data is shown below. Note that the results should not be exactly as such and some variations are acceptable.



K = 2*[3 points]*

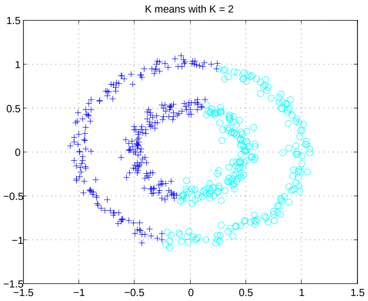


K = 3 *[3 points]*

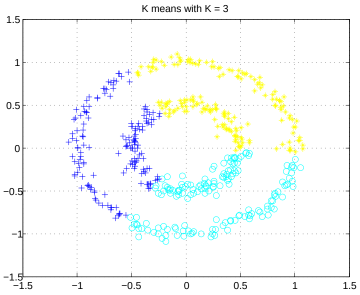


K = 5 *[1 points]*

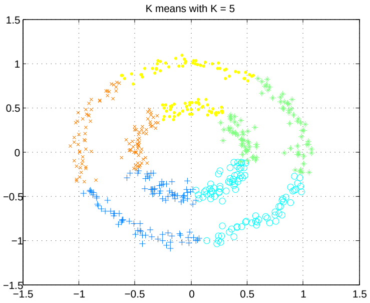
The linear K-means results for the circular.csv data is shown below. Note that the results should not be exactly as such and some variations are acceptable.



K = 2*[1 points]*



K = 3*[1 points]*



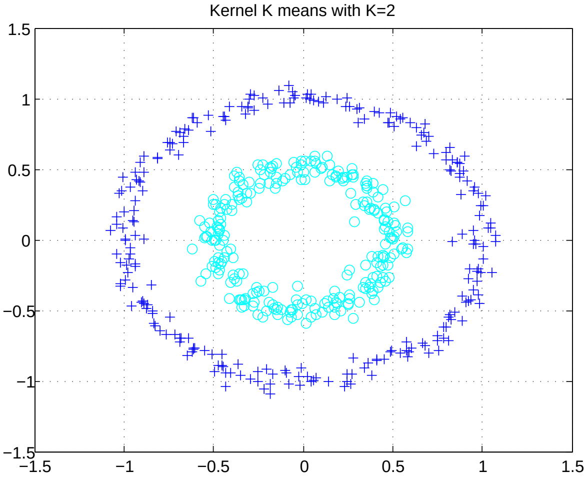
K = 5*[1 points]*

**b)** *[5 points]* The regular k-means can only separate data that are linearly separable otherwise some nonlinear kernels need to be used to map the data into a new space in which they are linearly separable.

**4.3***[15 points]*

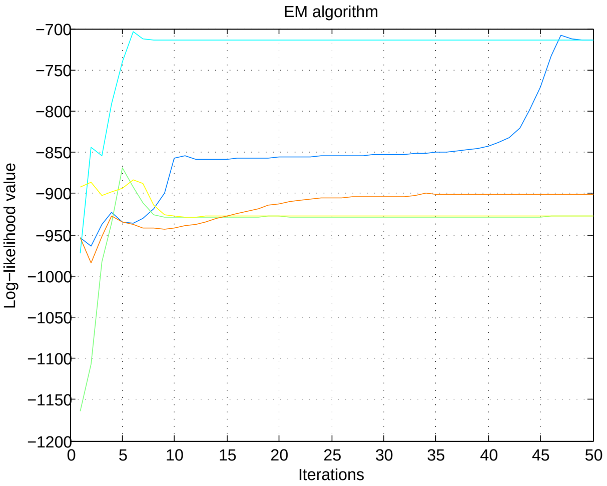
**a)** *[5 points]* Many different kernels can be used, all the polynomial kernels with Odd degree and Gaussian Kernels can be used. Check the nonlinear function to make sure it is capable of actually take the data to a space that is linearly separable.

**b)** *[10 points]* the resulted graph should be able to separate the data as follows:



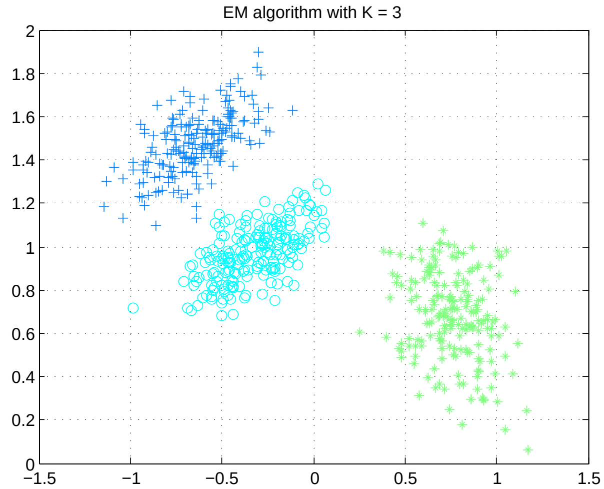
**4.4***[20 points]*

**a)** *[10 points]* The 5 tries of log-liklihood function can have different forms and shaped based on the initialization but the general trend should be as follows. Note that both log2 and log10 are acceptable.



**b)** *[10 points]*

The best of the above 5th try should be able to separate the data as follows:



And below are the means and covariance for the 3 different Gaussian components:
Cluster 1: mean $= [-0.639, 1.47]$, Cov $= [0.036, 0.015; 0.155, 0.019]$
Cluster 2: mean $= [-0.33, 0.97]$, Cov $= [0.036, 0.15; 0.015, 0.16]$
Cluster 3: mean $= [0.76, 0.68]$, Cov $= [0.027, -0.008; -0.008, 0.04]$
Note that the order of these three clusters does not matter.