1. Clustering

a.

We are given a set of data points $\{x_n\}_{n=1}^N$, and the method minimizes the following distortion measure (or objective or clustering cost):

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|_2^2$$

Where, μ_k is the prototype of the k-th cluster. r_{nk} is a binary indicator variable. If x_n is assigned to the cluster k, r_{nk} is 1 otherwise r_{nk} is 0.

Taking $\frac{\partial D}{\partial \mu_k} = 0$, we get

$$\sum_{n=1}^{N} r_{nk}[-2(x_n - \mu_k)] = 0$$

Or,

$$\sum_{n=1}^{N} r_{nk}\ \mu_k\ = \sum_{n=1}^{N} r_{nk}\ x_n$$

Therefore, we have,

$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} r_{nk}\ x_n}{\sum_{n=1}^{N} r_{nk}}}$$

b.

We are given that:

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|_1$$

Now, minimizing the D with respect to μ_k for a particular cluster M, we have

$$\frac{\partial D}{\partial \mu_k} = \sum_{m=1}^{M} F(x_n - \mu_k)$$

Where

$$F(x_n - \mu_k) = \begin{cases} 1, & \text{when } x_n > \mu_k \\ -1, & \text{when } x_n < \mu_k \end{cases}$$

Now, $\frac{\partial D}{\partial \mu_k} = 0$ when μ_k separates the points to its left and right equally. That is, $I(x_n |\ x_n < \mu_k) = I(x_n |\ x_n > \mu_k)$. Which means μ_k is the median.

c.

i.

If we apply a mapping ϕ(x) to map data points into feature space, then, we define the objective function of kernel K-means as:

$$\widetilde{D} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\phi(x_n) - \mu_k\|_2^2$$

Where

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

Now, let's consider $\|\phi(x_n) - \tilde{\mu}_k\|_2^2$

$$\|\phi(x_n) - \mu_k\|_2^2 = (\phi(x_n) - \tilde{\mu}_k)^T\ (\phi(x_n) - \tilde{\mu}_k)$$

$$\|\phi(x_n) - \mu_k\|_2^2 = \phi(x_n)^T\ \phi(x_n)\ - 2\tilde{\mu}_k{}^T\ \phi(x_n) + \tilde{\mu}_k{}^T\tilde{\mu}_k$$

$$\|\phi(x_n) - \mu_k\|_2^2 = \phi(x_n)^T\ \phi(x_n)\ - 2\frac{\sum_{n=1}^{N} r_{nk}\ \phi(x_n)^T\phi(x_n)}{\sum_{n=1}^{N} r_{nk}} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}\ \phi(x_i)^T\phi(x_j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}}$$

Now we can substitute,

$$K(x_i, x_j) = \ \phi(x_i)^T\ \phi(x_j)$$

We get,

$$\|\phi(x_n) - \mu_k\|_2^2 = K(x_n, x_n)\ - 2\frac{\sum_{n=1}^{N} r_{nk}\ K(x_n, x_n)}{\sum_{n=1}^{N} r_{nk}} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}\ K(x_i, x_j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}}$$

For simplicity of notations lets represent $R_k = \sum_{n=1}^{N} r_{nk}$. Then,

$$\|\phi(x_n) - \mu_k\|_2^2 = K(x_n, x_n)\ - 2\frac{\sum_{n=1}^{N} r_{nk}\ K(x_n, x_n)}{R_k} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}\ K(x_i, x_j)}{R_k^2}$$

Substituting this in the original equation:

$$\boxed{\widetilde{D} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\left[K(x_n, x_n)\ - 2\frac{\sum_{n=1}^{N} r_{nk}\ K(x_n, x_n)}{R_k} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}\ K(x_i, x_j)}{R_k^2}\right]}$$

ii.

For given point x_n calculate $\|\phi(x_n) - \mu_k\|_2^2$ and $\widetilde{D}$ for all possible clusters k

Assign cluster to point x_n using:

$$r_{nk} = \begin{cases} 1, & k = argmin_k \|\phi(x_n) - \mu_k\|_2^2 \\ 0, & otherwise \end{cases}$$

Where

$$\|\phi(x_n) - \mu_k\|_2^2 = K(x_n, x_n) - 2\frac{\sum_{n=1}^{N} r_{nk}\ K(x_n, x_n)}{R_k} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\, r_{jk}\ K(x_i, x_j)}{R_k^2}$$

And $R_k = \sum_{n=1}^{N} r_{nk}$

iii.

Pseudo Code:

```
1.  procedure Kernel K means
2.  C[i] = x(random(1..N)) for 1 < i < k        Initialise cluster centroids[1..k]
                                                 choosing any k points randomly of N
3.  for i : 1 to N do:
4.      for j : 1 to N do:
5.          K[i,j] ← φ(x_i)^T φ(x_j)
6.      end for
7.  end for
8.  r(n,k) ← [0]
9.  for i : 1 to N do:
10.     j ← argmin_k ||φ(x_i) − C_k||_2^2       Use the formula to calculate L2 distances
11.     r[i,j] ← 1
12.     Update C[j]                             Recalculate centroids of assigned cluster j
13. end for
14. end procedure
```

2. Gaussian mixture model

We are given the prior probabilities as

$$P(\theta_1) = \alpha$$

and therefore

$$P(\theta_2) = 1 - \alpha$$

Since we know that the data is generated from a univariate Gaussian,

We have,

$$f(x|\theta_1) \sim N(0,1)$$

$$P(x|\theta_1) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$$

And

$$f(x|\theta_2) \sim N(0,0.5)$$

$$P(x|\theta_1) = \frac{1}{\sqrt{\pi}} \exp(-x^2)$$

The likelihood can be written as

$$L(\alpha) = P(\theta_1)\, P(x|\theta_1) + P(\theta_2)\, P(x|\theta_2)$$

$$L(\alpha) = \alpha \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}) + (1-\alpha) \frac{1}{\sqrt{\pi}} \exp(-x^2)$$

Or

$$L(\alpha) = \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) - \frac{1}{\sqrt{\pi}} \exp(-x^2)\right) \alpha + \frac{1}{\sqrt{\pi}} \exp(-x^2)$$

This function is linear in α. The slope of the function when $x^2 \geq \log(2)\ and\ \alpha = 1$ is positive. We can start expectation maximization as we do regularly for GMMs or use $prior = \alpha = 0.5$. The α then gets updated at each step and eventually converges after many iterations to the global minimum. The increase or decrease in α is determined by the Gaussian that the point belongs to.

3. EM Algorithm

We are given that,

$$f(x) = \begin{cases} \pi + (1-\pi)e^{-\lambda}, & x_i = 0 \\ (1-\pi)\ \dfrac{\lambda^{x_i} e^{-\lambda}}{x_i!}\ , & x \geq 0 \end{cases}$$

We can rewrite this as

$$X_i = \begin{cases} x_i, & probability = (1-\pi)\,\dfrac{\lambda^{x_i}e^{-\lambda}}{x_i!} \\ 0, & probability = \pi + (1-\pi)e^{-\lambda} \end{cases}$$

We define a latent variable Zi for all cases where Xi = 0. (because when we observed Xi = 0 we do not know if it came out of the 'Poisson' distribution or it came out the 'degenerate' distribution). So Xi comes out of a mixture of a degenerate distribution as follows:

$$Z_i = \begin{cases} 1, & X_i \text{ is from degenerate distrbution} \\ 0, & otherwise \end{cases}$$

Therefore, we have,

$$P(X_i = 0, Z_i = 0) = P(Z_i = 1 = 0)\ \cdot P(X_i = 0|Z_i = 0) = (1-\pi)e^{-\lambda} \cdot 1$$

and

$$P(X_i = 0, Z_i = 1) = P(Z_i = 1)\ \cdot P(X_i = 0|Z_i = 1) = \pi \cdot 1$$

Now, we can write the Likelihood function as:

$$L\big((\pi,\lambda)\big|(X,Z)\big) = \prod_{x_i=0} \pi^{z_i} \cdot \Big((1-\pi)e^{-\lambda}\Big)^{1-z_i} \times \prod_{x_i>0} (1-\pi)\,\frac{\lambda^{x_i}e^{-\lambda}}{x_i!}$$

Taking the log, we get log likelihood as:

$$\log L\big((\pi,\lambda)\big|(X,Z)\big) = \sum_{I(x_i=0)} z_i \log(\pi) + (1-z_i)(\log(1-\pi) - \lambda) + \sum_{I(x_i>0)} (\log(1-\pi) + x_i \log(\lambda_i) - \lambda - \log(x_i!))$$

Notation $\theta = (\pi,\lambda)\theta_0$ represents a known parameter

E Step:

$$Q(\theta,\theta_0) = \sum_{I(x_i=0)} E_{P(Z|X)}\,(z_i)\log(\pi) + \big(1 - E_{P(Z|X)}\,(z_i)\big)(\log(1-\pi) - \lambda) + \sum_{I(x_i>0)} (\log(1-\pi) + x_i \log(\lambda_i) - \lambda - \log(x_i!))$$

Where,

$$E_{P(Z|X)}\,(z_i) = 0 \times P(Z_i = 0|X_i = 0) + 1 \times P(Z_i = 1|X_i = 0)$$
$$= \frac{P(Z_i = 1)\,P(X_i = 0|Z_i = 1)}{P(Z_i = 0)\,P(X_i = 0|Z_i = 0) + P(Z_i = 1)\,P(X_i = 0|Z_i = 1)}$$
$$= \frac{\pi_0}{(1-\pi_0)e^{-\lambda_0} + \pi_0}$$

So,

$$1 - E_{P(Z|X)}\,(z_i) = 1 - \frac{\pi_0}{(1-\pi_0)e^{-\lambda_0} + \pi_0} = \frac{\big[(1-\pi_0)e^{-\lambda_0} + \pi_0\big] - \pi_0}{(1-\pi_0)e^{-\lambda_0} + \pi_0}$$
$$= \frac{(1-\pi_0)e^{-\lambda_0}}{(1-\pi_0)e^{-\lambda_0} + \pi_0}$$

Hence,

$$Q(\theta, \theta_0) = \sum_{I(x_i=0)} \frac{\pi_0}{(1-\pi_0)e^{-\lambda_0} + \pi_0} \log(\pi) + \left(\frac{(1-\pi_0)e^{-\lambda_0}}{(1-\pi_0)e^{-\lambda_0} + \pi_0} \right) (\log(1-\pi) - \lambda) + \sum_{I(x_i>0)} (\log(1-\pi) + x_i \log(\lambda_i) - \lambda - \log(x_i!))$$

M Step:

Taking the gradient, $\frac{\partial Q}{\partial \lambda} = 0$

$$\sum_{I(x_i=0)} \left(1 - E\,(z_i)\right)(-1) + \sum_{I(x_i>0)} \left(\frac{x_i}{\lambda} - 1\right) = 0$$

$$\hat{\lambda} = \frac{\sum_{I(x_i>0)} x_i}{n - \sum_{I(x_i=0)} E\,(z_i)}$$

Or

$$\boxed{\hat{\lambda} = \frac{\sum_{I(x_i>0)} x_i}{n - \sum_{I(x_i=0)} \hat{z}_i}}$$

Where,

$$\boxed{\hat{z}_i = E\,(z_i) = \frac{\pi_0}{(1-\pi_0)e^{-\lambda_0} + \pi_0}}$$

Taking the gradient, $\frac{\partial Q}{\partial \pi} = 0$

$$\sum_{I(x_i=0)} \left(\frac{E(z_i)}{\pi} - \frac{1 - E(z_i)}{1-\pi} \right) + \sum_{I(x_i>0)} \left(\frac{1}{1-\pi} \right) = 0$$

$$\sum_{I(x_i=0)} \left(\frac{E(z_i)}{\pi} + \frac{E(z_i)}{1-\pi} \right) - \frac{n}{1-\pi} = 0$$

$$\boxed{\hat{\pi} = \frac{1}{n} \sum_{I(x_i=0)} \hat{z}}$$

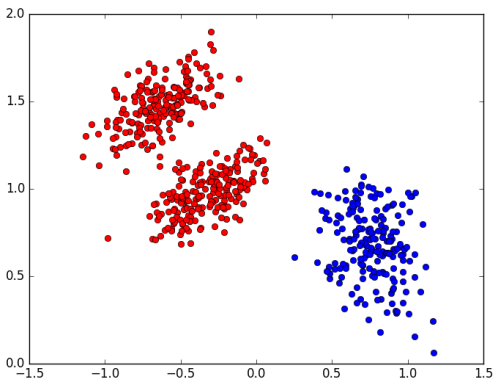
Therefore, the parameter updates will be as highlighted above (enclosed in boxes)

4. Programming

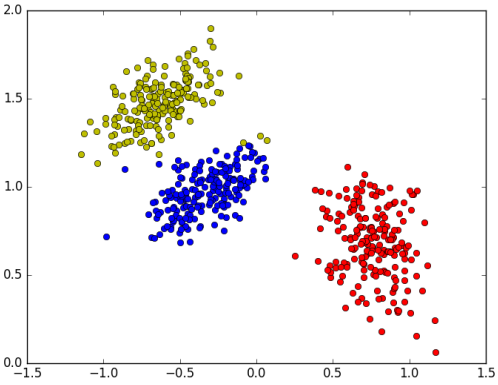
ii. Implement k-means

a.

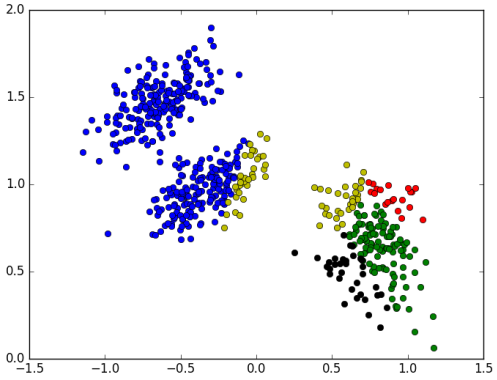
Cluster Plots for hw5_blob.csv



K = 2

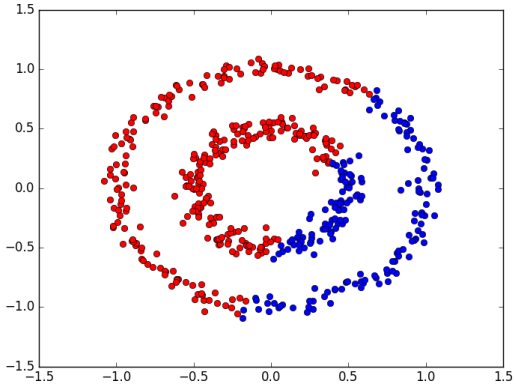


K = 3

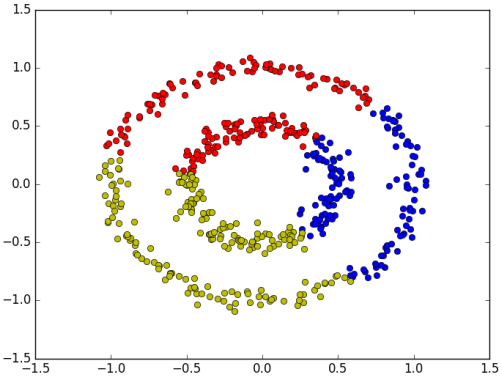


K = 5

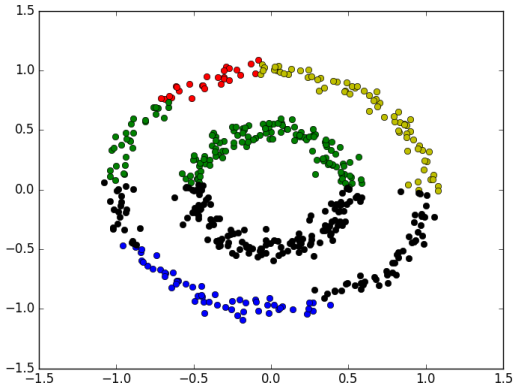
Cluster Plots for hw5_circle.csv



K = 2



K = 3



K = 5

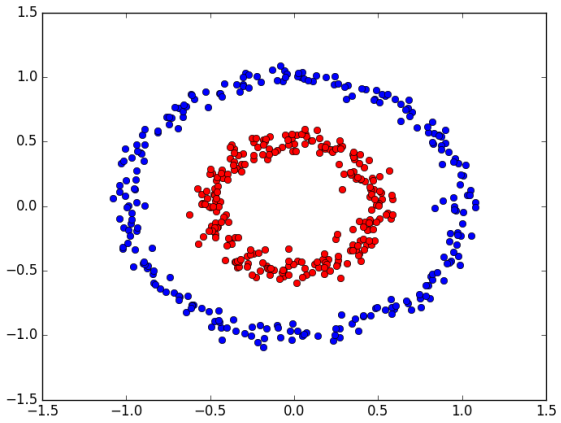
b.

The k Means algorithm uses a linear decision boundary. Hence it has trouble separating a quadratic data set like a circle as two circles. When you run k means on such a data set, we get the circle separated using a linear boundary as shown in the above figures. To correctly separate the circles, we use a kernel function to transform the features to a more linearly separable form. (like $(x^2 + y^2)$ for circles)

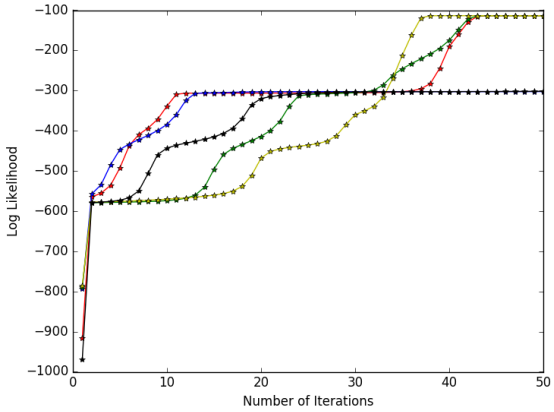
iii. Implement kernel k-means

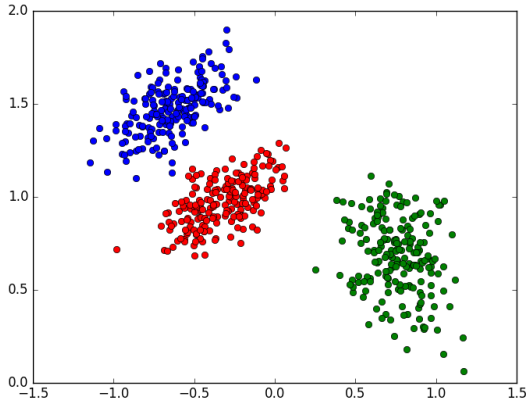
a. Kernel function used is a polynomial kernel with the kernel function $K(x_1, x_2) = [x_1^2 + x_2^2]$

b. Plot of cluster assignments



iv. Implement Gaussian Mixture Model





Best Values Selected
Log Likelihood = -113.67086782
(5 runs of 50 iterations each)

For Cluseter 1:
Mean =
[[-0.32592106 0.97133574]]
Covariance =
[[0.03604954 0.01463887]
[0.01463887 0.0162912]]

For Cluseter 2:
Mean =
[[-0.6394629 1.4746064]]
Covariance =
[[0.0359676 0.01549315]
[0.01549315 0.01935168]]

For Cluseter 3:
Mean =
[[0.75896032 0.67976982]]
Covariance =
[[0.02717056 -0.00840045]
[-0.00840045 0.040442]]

Collaboration

Collaborated on thoughts and ideas with **Adarsha Desai** and **Mahesh Pottippala Subrahmanya**