Assignment5

Arpita Agrawal, USC ID: 8100884538

November 2016

1 Clustering

a) $D = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \mu_k||_2^2$, where r_{nk} is either 1 or 0 depending on if the point n is in the cluster k or not and μ_k is the prototype of the cluster k.

Our cost function J is $D = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \mu_k||_2^2$, which is basically the distance of each point from there clusters prototype μ_k. To minimise the distance equation, we will differentiate the equation w.r.t μ_k

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

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

Assuming N_k= Number of points in the cluster k, we have $\sum_{n=1}^{N} r_{nk} = N_k$

Therefore, $\mu_k = \frac{\sum_{n=1}^{N} r_{nk}x_n}{N_k}$, which shows μ_k is the mean of all data points assigned to the cluster k, for any k, when the objective D is minimized and hence justifies the iterative procedure of k-means

b) Changing the distortion measure to $D = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \overline{\mu_k}||_1$, i.e the distance measure has been changed from L_1 norm to L_2 norm. $(||z||_1 = \sum_d)||z_d||$

To minimise the above distance equation differentiating w.r.t μ_k,

$$\frac{\partial}{\partial \mu_k} D = \frac{\partial}{\partial \mu_k}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \overline{\mu_k}||_1 = \sum_{n=1}^{N}\frac{\partial}{\partial \mu_k} r_{nk}||x_n - \overline{\mu_k}||_1 = \sum_{n=1}^{N} r_{nk} sign(x_n - \overline{\mu_k}) = 0$$

Solving for μ_k

$\sum_{n=1}^{N} r_{nk} sign(x_n - \overline{\mu_k})$ is going to be zero when the sum is going to 0. sign() function returns +1 / -1, so for the summation to be zero, half the elements should be on the left side of μ_k and the rest half should be on the right side of μ_k. And this summation can be broken down into feature wise summation i.e. :

$\sum_{n=1}^{N} r_{nk}\sum_{j=1}^{N} sign(x_{nj} - \overline{\mu_{kj}}) = 0$, for this equation to be zero, each element j of the vector μ_k has to be the median of the j elements of all vectors x_n

This proves if we change from L_1 norm to L_2 norm, the protypes of our cluster will have to be median for the minimisation of our distortion function.

c) Now assuming that we apply a mapping $\phi(x)$ to map data points into feature space our distortion function will transform into :

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

c -i) Show D can be represented in terms of only K(x_i, x_j) = $\phi(x_i)^T \phi(x_j)$

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

Replacing $\mu_k = \frac{\sum_{i=1}^{N} r_{ik}\phi(x_i)}{\sum_{i=1}^{N} r_{ik}}$ in the above equation, and substituting $\sum_{i=1}^{N} r_{ik} = N_k$, we get,

$$D = \sum_{i=1}^{N} \phi(x_n)^T\phi(x_n) - \frac{2\sum_{i=1}^{N} r_{ik}\phi(x_i)^T\phi(x_n)}{N_k} + \frac{2\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}\phi(x_i)^T\phi(x_j)r_{jk}}{N_k^2}$$

$$D = \sum_{i=1}^{N}\sum_{k=1}^{K} K(x, x_n) - \frac{2\sum_{i=1}^{N} K(x_i, x_n)}{N_k} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}r_{jk}K(x_i, x_j)}{N_k^2}$$

c -ii)

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

OR, in terms of $K(x_i, x_j)$ as below,

$$D = min \sum_{i=1}^{N}\sum_{k=1}^{K} K(x, x_n) - \frac{2\sum_{i=1}^{N} K(x_i, x_n)}{N_k} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ik}r_{jk}K(x_i, x_j)}{N_k^2}$$

c -iii) Pseudo Code for KMeans algorithm::

function kernelkMeans(data, k, kernel):

data = transform_data_using_kernel(data, kernel)
Initialise **mu**[k] = Pick k unique random points from the transformed dataset.

for i=1 to N:
    for j=1 to N:
        compute $K(x_i, x_j)$ by multiplying $\phi(x_i)\phi(x_j)$

while prev_mu_arr ! = mu_arr :

    prev_mu_arr = mu_arr
    for i=1 to N:
        for k=1 to K:
            dist[i][k] = compute distance of point i from prototype k using the equation formed in the previous section.

        min_dist_index[i] = argmin(dist[i][k]) row wise, cluster id for which the distance was minimum for data point i
        min_dist_index[i] will have the cluster ids of all the data points

        Recompute the mu_arr[i] array based on the cluster ids given in the previous statement

end while loop

2 EM algorithm

$$X = \begin{cases} \pi + (1-\pi)e^{-\lambda} & , \quad \text{if } x_i = 0 \\ (1-\pi)\frac{\lambda^{xi}e^{-\lambda}}{x_i!}, & \quad \text{if} x_i > 0 \end{cases} \tag{1}$$

To define the above equation, lets take a latent variable z, which will define the probabbility when x = 0, because when x=0, the probability given to us is $\pi + (1-\pi)e^{-\lambda}$ which is basically $(1-\pi)$ times poisson + something. So, that diversion is going to be described by our latent variable z, when $x_i = 0$

$z_i = 1$, when the distribution diverges from poisson, when x=0,
$z_i = 0$, when the distribution follows poisson distribution, when x=0

$P(X_i = 0; Z_i = 1) = P(Z_i = 1)P(X_i = 0|Z_i = 1) = \pi * 1$

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

likelihood $L = \prod_{x_i=0} \pi^{z_i}((1-\pi)e^{-\lambda})^{1-z_i} \prod_{x_i>0}(1-\pi)\frac{\lambda^{xi}e^{-\lambda}}{x_i!}$

Taking log on both sides:

$logL = \sum_{x_i=0} z_i log\pi + (1 - z_i)(log(1-\pi) - \lambda) + \sum_{x_i>0} z_i log(1-\pi) + x_i log(\lambda) - \lambda - log(x_i!)$

Now, Applying EM algorithm to find the optimal parameters,

E-step : Compute Q$(\theta|\theta^{(t)})$: $E_{p(z|x,\theta^{(t)})}[logp(x, z|\theta)]$

M-step :$\theta^{(t+1)} = argmax_\theta E_{p(z|x,\theta^{(t)})}[logp(x, z|\theta)]$

E- Step::
Using the log likelihood derived in part 1 and the E step definition defined before,

$Q(\theta|\theta^{(t)}) = E_{p(z|x,\theta^{(t)})}[z_i]log\pi + (1 - E_{p(z|x,\theta^{(t)})}[z_i])(log(1-\pi) - \lambda) + log(1-\pi) + x_i log(\lambda) - \lambda - log(x_i!)$

$E_{p(z|x,\theta^{(t)})}[z_i] = z_i P(z_i = 0|X) + z_i P(z_i = 1|X_i = 0) = 0 * P(z_i = 0|X) + 1 * P(z_i = 1|X_i = 0)$

Applying Bayesian theorem::

$= P(z_i = 1|X_i = 0) = \frac{P(x_i=0|z_i=1)P(z_i=1)}{P(x_i=0|z_i=0)P(z_i=0)+P(x_i=0|z_i=1)P(z_i=1)} = \frac{\pi_t}{\pi_t+(1-\pi_t)e^{-\lambda}}$

Substituting the above value back into the $Q(\theta|\theta^{(t)})$ equation, we get,

$Q(\theta|\theta^{(t)}) = \sum_{x=0} \frac{\pi_t}{\pi_t+(1-\pi_t)e^{-\lambda}} log\pi + (1 - \frac{\pi_t}{\pi_t+(1-\pi_t)e^{-\lambda}})(log(1-\pi) - \lambda) \quad +$

$\sum_{x_i>0} log(1-\pi) + x_i log(\lambda) - \lambda - log(x_i!)$

M Step::

$\frac{\partial}{\partial\lambda}Q = \sum_{x_i=0} -(1 - E[z_i]) + \sum_{x_i>0} \frac{x_i}{\lambda} - 1 = 0$

$= \sum_{x_i=0}(E[z_i] - 1) + \sum_{x_i>0} 1 = \sum_{x_i>0} -\frac{x_i}{\lambda}$

$\lambda = \frac{\sum_{x_i>0} -x_i}{\sum_{x_i=0} E[z_i] - 1 - \sum_{x_i>0} 1}$

$\hat{\lambda} = \frac{\sum_{x_i>0} x_i}{n - \sum_{x_i=0} E[z_i]}, \qquad E[z_i] = \frac{\pi_t}{\pi_t+(1-\pi_t)e^{-\lambda}}$

Differentiating w.r.t π

$\frac{\partial}{\partial\pi}Q = \sum_{x_i=0} \frac{E[z_i]}{\pi} - \frac{1}{1-\pi}(1 - E[z_i]) + \sum_{x_i>0} \frac{1}{1-\pi} = 0$

$\sum_{x_i=0} \frac{(1-\pi)E[z_i]+\pi E[z_i]}{\pi(1-\pi)} - \frac{n}{1-\pi} = \sum_{x_i=0} \frac{E[z_i]}{\pi(1-\pi)} - \frac{n}{1-\pi} = \sum_{x_i=0} E[z_i] - n\pi = 0$
$\hat{\pi} = \sum_{x_i=0} \frac{E[z_i]}{n}$

Final updated paramters are:

$\hat{\pi} = \sum_{x_i=0} \frac{E[z_i]}{n}$

$\hat{\lambda} = \frac{\sum_{x_i>0} x_i}{n - \sum_{x_i=0} E[z_i]}$,

$\hat{z}_i = \frac{\pi_t}{\pi_t + (1-\pi_t)e^{-\lambda}}$

3 Gaussian Mixture Model

$$f(x|\theta_1) = \mathcal{N}(\mu_1, \sigma_1^2), \qquad f(x|\theta_2) = \mathcal{N}(\mu_2, \sigma_2^2)$$

Given $\mu_1 = 0, \sigma_1 = 1, \mu_2 = 0, \sigma_2 = 0.5$

Likelihood function for $p(x_n|\alpha)$

$$p(x_n|\alpha) = \frac{\alpha}{\sqrt{2\pi}} e^{\frac{-(x_n)^2}{2}} + \frac{(1-\alpha)}{\sqrt{\pi}} e^{-(x_n)^2} = \alpha\left(\frac{1}{\sqrt{2\pi}} e^{\frac{-(x_n)^2}{2}} \frac{-1}{\sqrt{\pi}} e^{-(x_n)^2}\right) + \frac{1}{\sqrt{\pi}} e^{-(x_n)^2}$$

So, the above equation can be visualised in terms of y= mx + c, i..e a linear equation.

As m increase y should increase, and as y decrease, if we decrease the value of alpha take it close to zero, it can still increase depending on our constant term.

If we look at the slope, m = $\frac{1}{\sqrt{2\pi}} e^{\frac{-(x_n)^2}{2}} \frac{-1}{\sqrt{\pi}} e^{-(x_n)^2}$

The above slope term is going to be greater than 0 when:

$\frac{1}{\sqrt{2\pi}} e^{\frac{-(x_n)^2}{2}} \frac{-1}{\sqrt{\pi}} e^{-(x_n)^2} > 0$

$\frac{1}{\sqrt{2}} e^{\frac{-(x_n)^2}{2}} > e^{-(x_n)^2}$, Taking log on both sides

$\frac{-(x_n)^2}{2} > log(\sqrt{2}) - x_1^2)$

$\frac{(x_n)^2}{2} > log(\sqrt{2}), \quad (x_n)^2 > log2$, at this point we should set $\alpha = 1$ else, we should set $\alpha = 0$, so that the slope term minimised and constant term determines the value of likelihood.

So, when $(x_n)^2 > log2$, we should set $\alpha = 1$ else we should set $\alpha = 0$

4 Programming

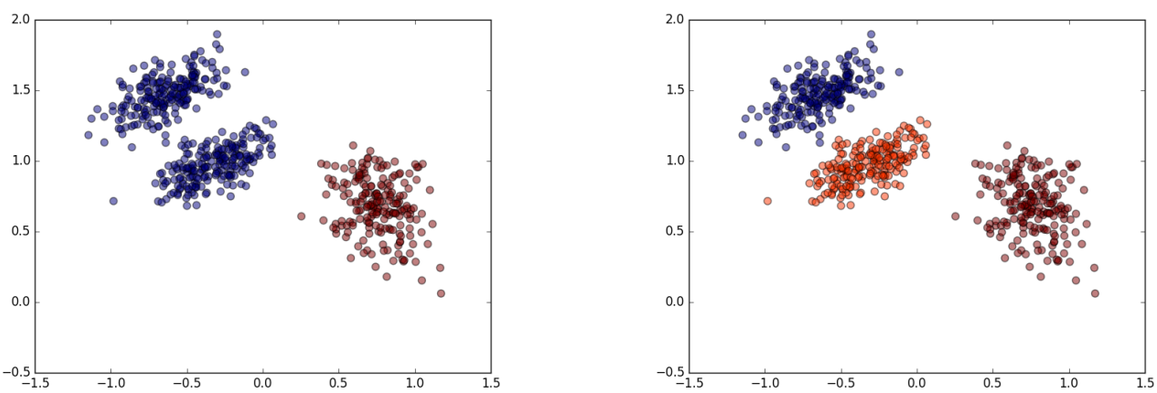


Figure 1: a) KMeans with k=2 (blob data) b) KMeans with k=3 (blob data)

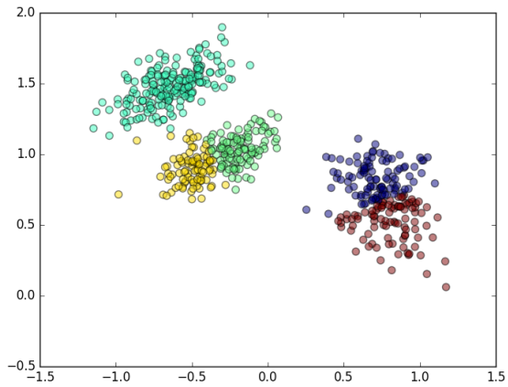
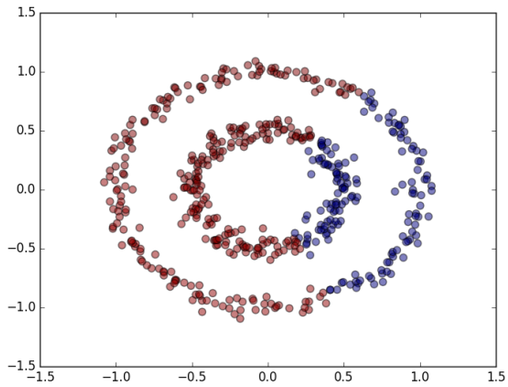


Figure 2: a) KMeans with k=5 (blob data)



b) KMeans with k=2 (cirlce data)

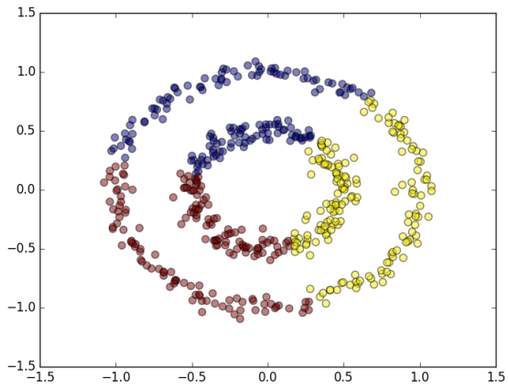
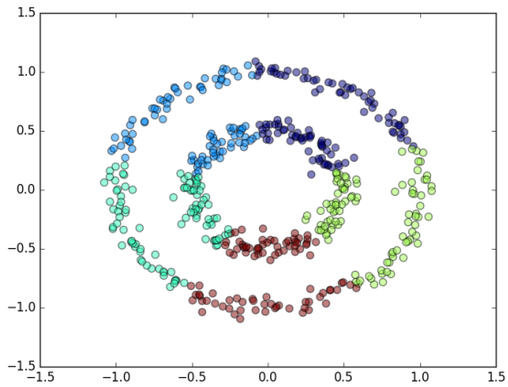


Figure 3: a) KMeans with k=3 (cirlce data)



b) KMeans with k=5 (cirlce data)

As we can see from the above figures, blob data could be easily separated with k=2 in our 2 dimensional space. But, we were unable to separate the rings data present in the circle data. That happened because there is no line present which can divide the circle data into two different clusters in the linear space.KMeans tries to seperate the data using the protype point at the centre of the cluster, it grows in an outward direction and does this seperation circularly. That type of seperation is not possible here on this data. So, to clusterise the ring data present in the circle csv we will have to user kernel k means, transform our data into some other dimension, cluster the data and present it back into the 2 dimensional space.

5 Implement kernel k-means

a) Kernel used for seperating the circle data :

$$[\quad x_1^2 \qquad x_1^2 + x_2^2 \qquad x_2^2 \quad]$$

‘

The above kernel is a valid kernel. It can be visualised as a summation of two polynomial kernels, Polynomial kernels are themselves valid kernel functions, and when two valid kernel functions are added together the resulting kernel function is also valid. As proved in earlier assignments.

End

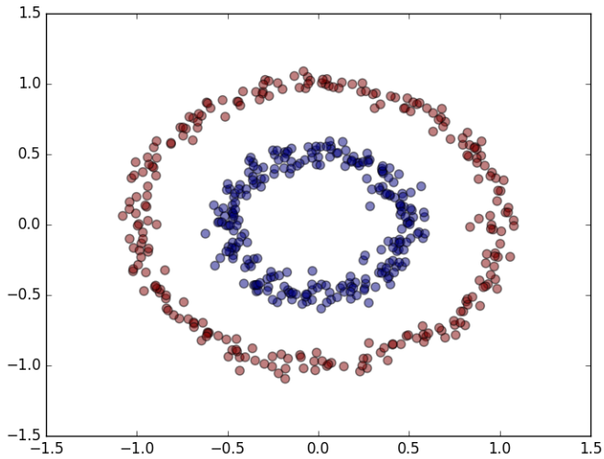


Figure 4: Kernel K means executed on circle data. With the kernel transformation, now we are able to seperate the data into two seperate rings.

6 Implement Gaussian Mixture Model

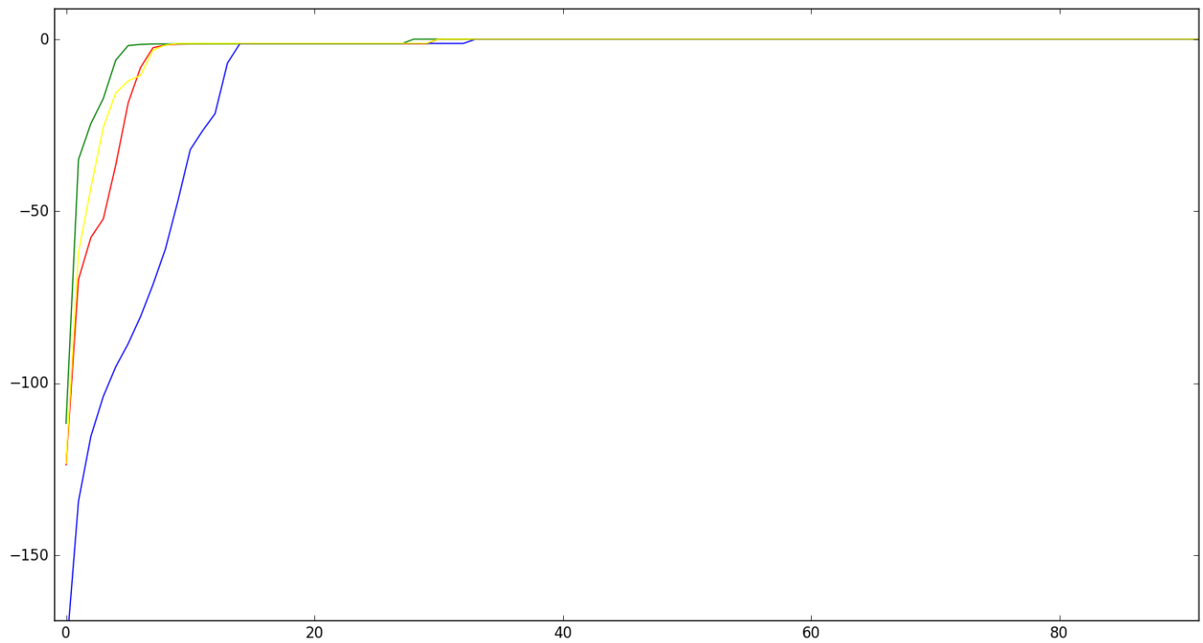


Figure 5: Log Likelihood of the data over iterations of EM for 5 consecutive runs

Final parameters for μ, σ, ϕ are:

Final Mean for Best run :::

[[-0.32605638 0.97141657]
[-0.63945809 1.47639628]
[0.7589834 0.67975196]]

Final Sigma for run 4 :::

[[[0.03607098 0.01460963]
[0.01460963 0.01629659]]

[[0.03613339 0.01557092]

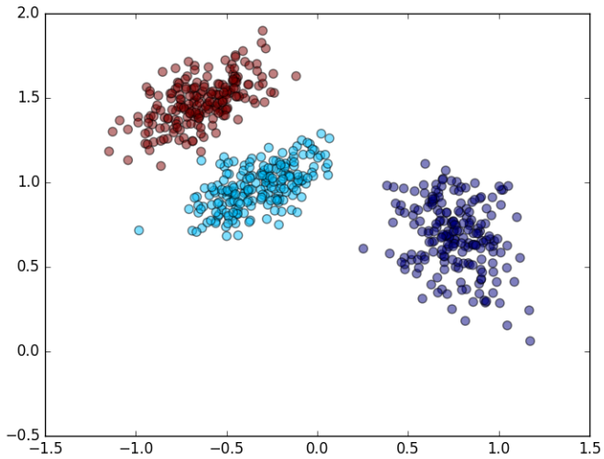


Figure 6: Log Likelihood of the data over iterations of EM for 5 consecutive runs

[0.01557092 0.01885686]]

[[0.02716186 -0.00839328]
[-0.00839328 0.04043837]]]