Machine Learning HW5

Siddharth Bhadkamkar

USC ID: 8342072533

bhadkamk@usc.edu

(213-257-4605)

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

For every n, only one r_{nk} is 1, all others are 0.

So for a cluster k' we can differentiate with respect to a particular $\mu_{k'}$ and get rid of $\sum_{k=1}^{K}$

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

$$\frac{\partial}{\partial \mu_k'}(D) = \frac{\partial}{\partial \mu_k'} \sum_{n=1}^{N} r_{nk'} ||x_n - \mu_k||_2^2 = 0$$

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

$\mu_k = \frac{\sum_{n=1}^{N} r_{nk'} x_n}{\sum_{n=1}^{N} r_{nk'}}$ So μ_k is mean of its member points

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

summing absolute values over all the dimensions d.

$$||x_n - \mu_k||_1 = \sum_{j=1}^{d} |x_{nj} - \mu_{kj}|$$

We also know that all the dimensions are independent of each other.
So we can minimize the following functions for each dimension j.

$$D_j = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} |x_{nj} - \mu_{kj}|$$

We introduce s_{nk} such that $s_{nk} = -1 \ if x_{nj} > \mu_{kj}$, $s_{nk} = 1 \ otherwise$

$$D_j = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} s_{nk} (-x_{nj} + \mu_{kj})$$

Consider a particular cluster k' and dimension j. We differentiate by $\mu_{k'j}$

$$\frac{\partial}{\partial \mu_{k'j}} D_j = \frac{\partial}{\partial \mu_{k'j}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} s_{nk} (-x_{nj} + \mu_{kj})$$

Element corresponding to k' will remain and others will become 0.

$$\frac{\partial}{\partial \mu_{k'j}} D_j = \frac{\partial}{\partial \mu_{k'j}} \sum_{n=1}^{N} r_{nk'} s_{nk'} (-x_{nj} + \mu_{k'j})$$

$$\frac{\partial}{\partial \mu_{k'j}} D_j = \sum_{n=1}^{N} r_{nk'} s_{nk'} \quad = 0$$ …. Equated to 0.

From this equation it is clear that on dimension j, half of the member points lie on left side and other half on the right side of the j-th dimension of mean ie. $\mu_{k'j}$

So $\mu_{k'j}$ is indeed median on j-th dimension. This is true for each j in 1 to d.

Hence proved

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

From 1a we know $\mu_k = \frac{\sum_{n=1}^{N} r_{nk}\phi(x_n)}{\sum_{n=1}^{N} r_{nk}}$

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

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

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

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

So D can be represented in terms of only kernel K(x i , x j)

So for a point x_n , distance from cluster k centroid is

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

We choose cluster k such that it minimizes d

Pseudocode:

1. Initialize random positions for the cluster centers in original feature space.
2. Find distance between each point and each cluster center in the mapped feature space using kernel functions and assign each point a cluster (one whose center is nearest).
3. For each cluster, update its center by bringing it to the mean position with respect to it's member points.
4. Go back to step 2 if cluster centers changed. Else exit.

2. $f(x|\theta_1) = N(\mu_1, \sigma_1{}^2)$

$f(x|\theta_2) = N(\mu_2, \sigma_2{}^2)$

$$L(x_1|\theta_1, \theta_2, \alpha) \ = \ \alpha N(\mu_1, \sigma_1{}^2) \ + (1-\alpha) N(\mu_2, \sigma_2{}^2)$$

Since $0 \le \alpha \le 1$ we know that

$$min(N(\mu_1, \sigma_1{}^2), N(\mu_2, \sigma_2{}^2)) \le L(x_1|\theta_1, \theta_2, \alpha) \le max(N(\mu_1, \sigma_1{}^2), N(\mu_2, \sigma_2{}^2))$$

We want to maximize $L(x_1|\theta_1, \theta_2, \alpha)$

So $L(x_1|\theta_1, \theta_2, \alpha) = max(N(\mu_1, \sigma_1{}^2), N(\mu_2, \sigma_2{}^2))$

If $N(\mu_1, \sigma_1{}^2) > N(\mu_2, \sigma_2{}^2)$ *then* $\alpha = 1$ *else* $\alpha = 0$

$$N(\mu_1, \sigma_1{}^2) = exp(-0.5x_1{}^2)\, for\ \mu_1 = 0\ \&\ \sigma_1{}^2 = 1$$

$$N(\mu_2, \sigma_2{}^2) = \frac{exp(-x_1{}^2)}{\sqrt{0.5}}\, for\ \mu_1 = 0\ \&\ \sigma_1{}^2 = 1$$

So $\alpha = 1\ if\ x_1{}^2 > ln(2)\ else\ \alpha = 0$

3. $p(x_i) = \pi + (1-\pi)e^{-\lambda} \; if \; x_i = 0$

$p(x_i) = (1-\pi)\frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \; if \; x_i > 0$

$\pi \; is \; probability \; of \; zero \; distribution$

$\pi \; is \; probability \; of \; poisson \; distribution$

Let us introduce hidden variable z_i such that $z_i = 1 \; if \; x_i \; from \; zero \; distribution$

$z_i = 0 \; if \; x_i \; from \; poisson \; distribution$

So we can write the likelihood function using an Indicator function I.

$$L(x_i|\pi,\lambda) = \prod_{i=1}^{N} (\pi)^{z_i I(x_i=0)}[(1-\pi)e^{-\lambda}]^{(1-z_i)I(x_i=0)}[\frac{(1-\pi)\lambda^{x_i}e^{-\lambda}}{x_i!}]^{(1-z_i)I(x_i>0)}$$

So log likelihood is:

$$l(x_i|\pi,\lambda) = \sum_{i=1}^{N} z_i I(x_i = 0)log(\pi) + (1-z_i)I(x_i = 0)log\,[(1-\pi)e^{-\lambda}]$$

$$+ (1-z_i)I(x_i > 0)log[\frac{(1-\pi)\lambda^{x_i}e^{-\lambda}}{x_i!}]$$

Estimation step:

$$z_i = \frac{p(x_i|zero\ Distribution)p(zero\ Distribution)}{p(x_i|zero\ Distribution)p(zero\ Distribution) + p(x_i|poisson\ Distribution)p(poisson\ Distribution)}$$

Use $x_i = 0$

$$z_i = \frac{\pi}{\pi+(1-\pi)e^{-\lambda}}$$

Maximization step:

Differentiate $l(x_i|\pi,\lambda)$ by π and equate to 0.

$$\frac{\partial l(x_i|\pi,\lambda)}{\partial \pi} = 0 = \frac{\sum_{i=1}^{N} z_i I(x_i=0)}{\pi} - \frac{\sum_{i=1}^{N}(1-z_i)I(x_i=0)}{1-\pi} - \frac{\sum_{i=1}^{N}(1-z_i)I(x_i>0)}{1-\pi}$$

$$\frac{\sum_{i=1}^{N} z_i I(x_i=0)}{\pi} = \frac{\sum_{i=1}^{N}(1-z_i)I(x_i=0)}{1-\pi} + \frac{\sum_{i=1}^{N}(1-z_i)I(x_i>0)}{1-\pi} = \frac{\sum_{i=1}^{N}(1-z_i)}{1-\pi}$$

$$\frac{\pi}{1-\pi} = \frac{\sum_{i=1}^{N} z_i I(x_i=0)}{\sum_{i=1}^{N}(1-z_i)}$$

$$\pi = \frac{\sum_{i=1}^{N} z_i I(x_i=0)}{\sum_{i=1}^{N} (1-z_i)+z_i I(x_i=0)} = \frac{\sum_{i=1}^{N} z_i I(x_i=0)}{N}$$

Differentiate $l(x_i|\pi,\lambda)$ by λ and equate to 0.

$$\frac{\partial l(x_i|\pi,\lambda)}{\partial\lambda} = 0 = \sum_{i=1}^{N}(1-z_i)I(x_i=0)(-1) + \sum_{i=1}^{N}(1-z_i)I(x_i>0)(\frac{x_i}{\lambda}-1)$$

$$-\sum_{i=1}^{N}(1-z_i) + \sum_{i=1}^{N}\frac{(1-z_i)I(x_i>0)x_i}{\lambda} = 0$$

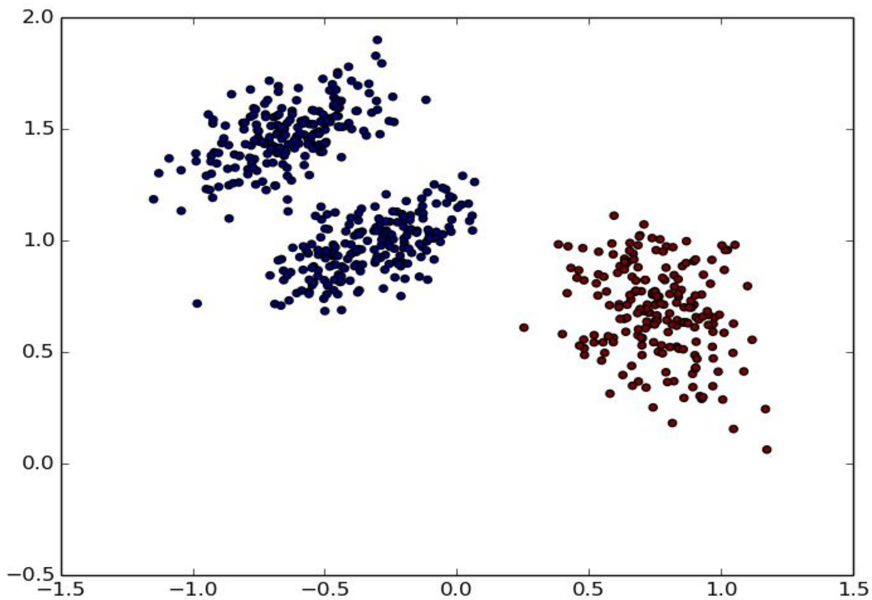
$$\lambda = \frac{\sum_{i=1}^{N} I(x_i>0)x_i}{N-\sum_{i=1}^{N} z_i I(x_i=0)}$$

4. **Programming**

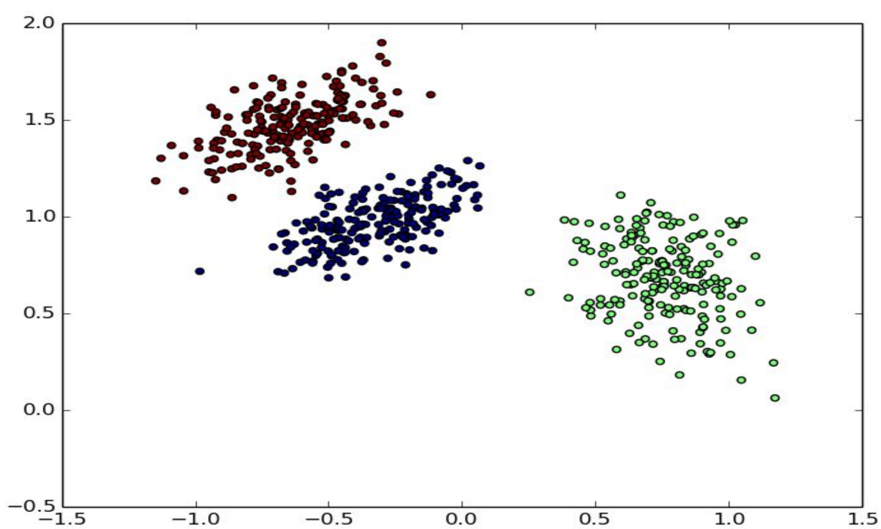
Implement k-means

a) hw5 blob.csv

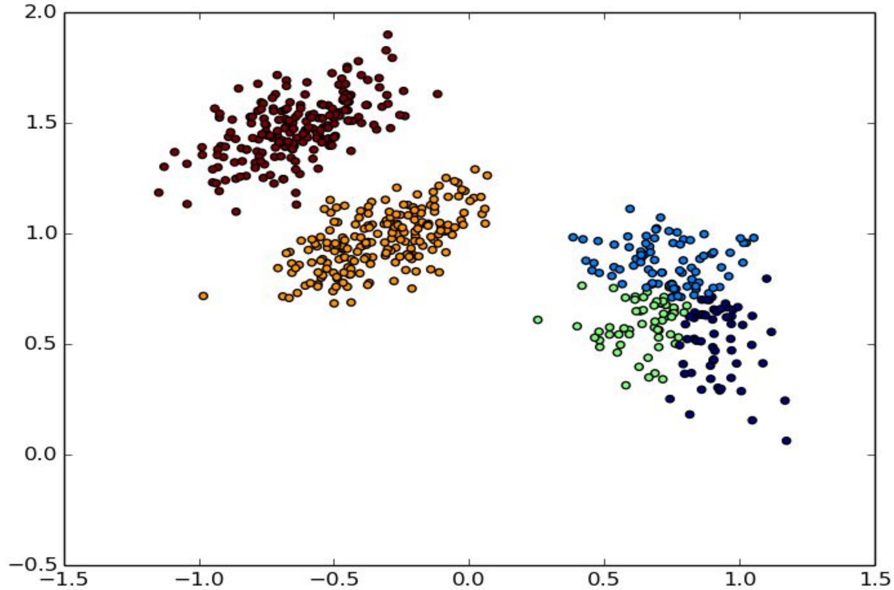
K=2



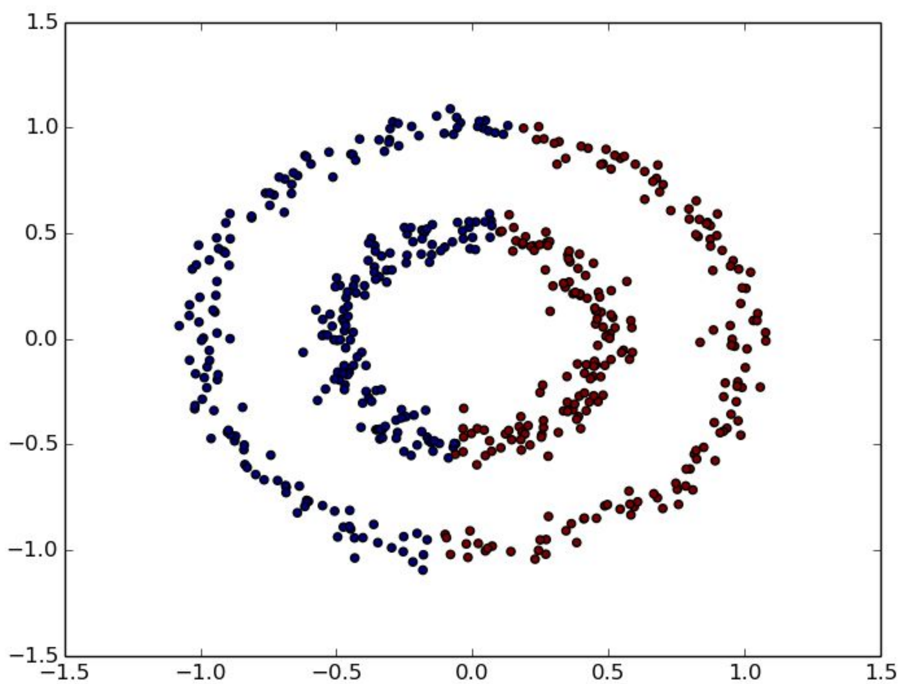
K=3



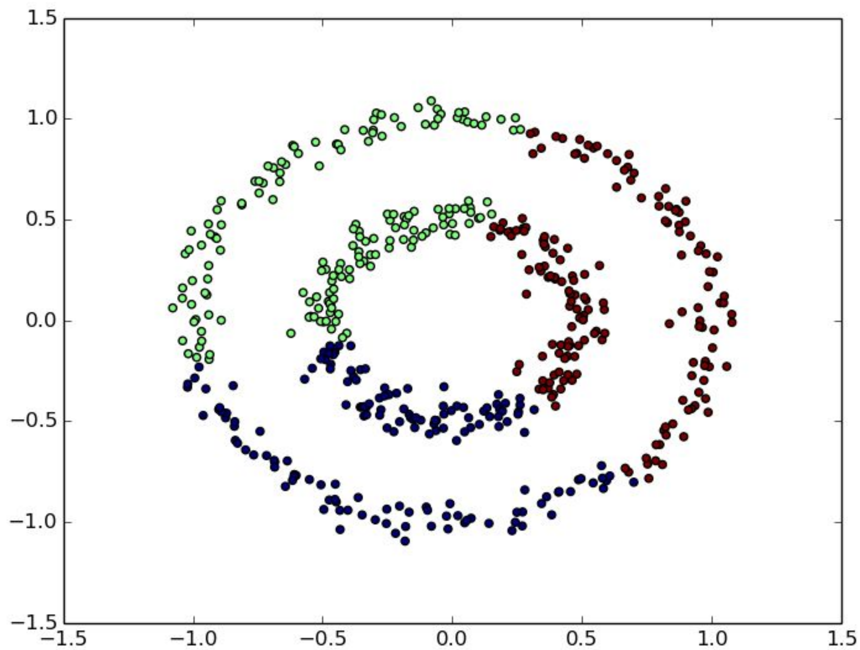
K=5



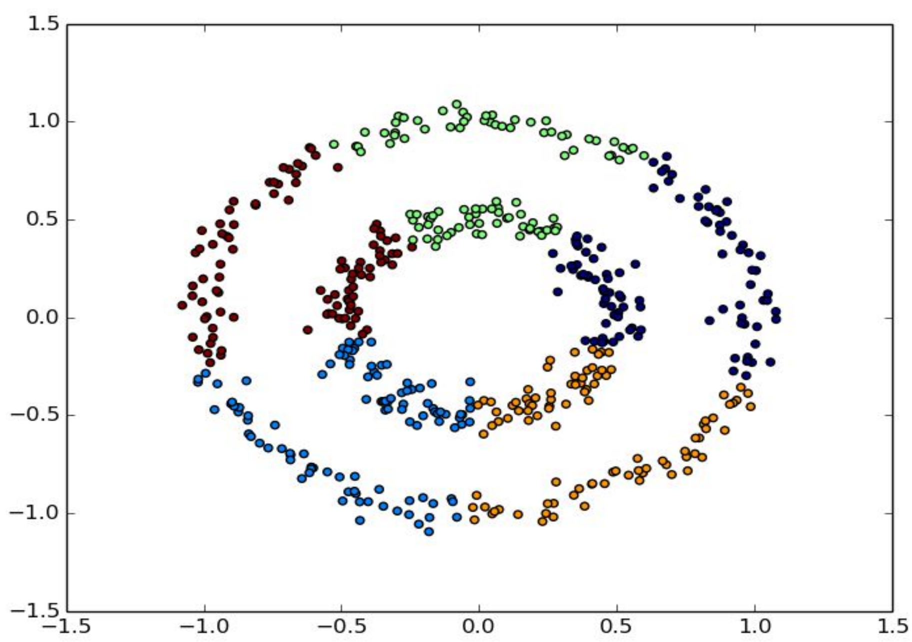
hw5 circle.csv
K=2



K=3


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

K=5


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

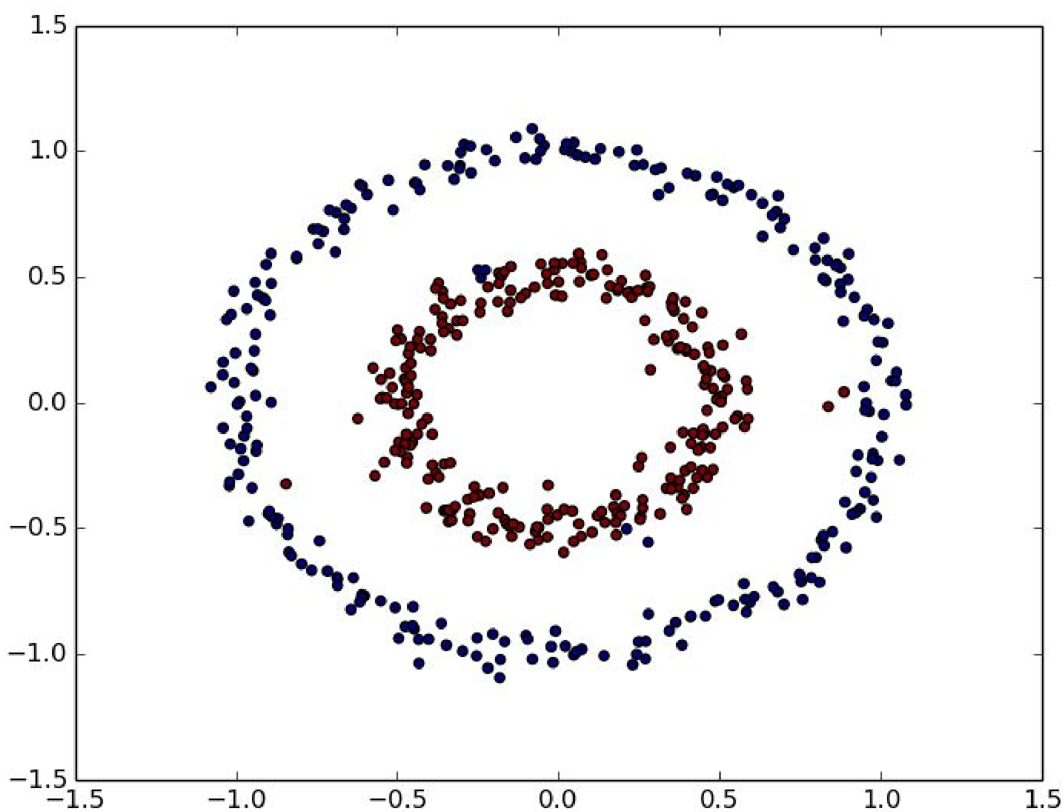
b) **k-means algorithm fails to separate the two circles in the hw5 circle.csv**

In k means clustering, each point is assigned to a cluster with the nearest centroid.
In case of concentric circles, you cannot find a point (in same feature space) that is nearer to all points in outer circle but far from inner circle. Hence k-means algorithm fails.
The only solution to this is to project our features in some higher dimensions and hope to have separate clusters in that feature space.

Implement kernel k-means

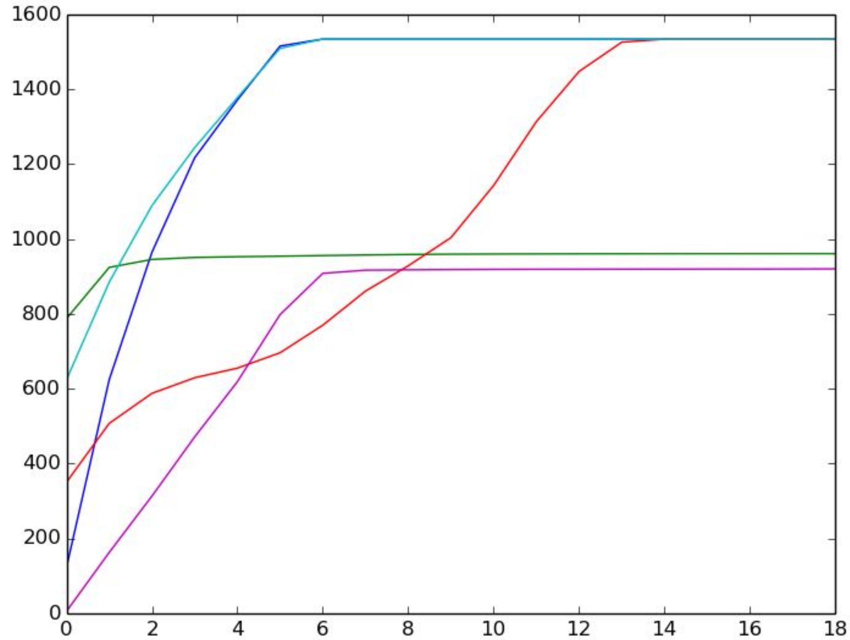
a) Used an RBF kernel.

b) K=2

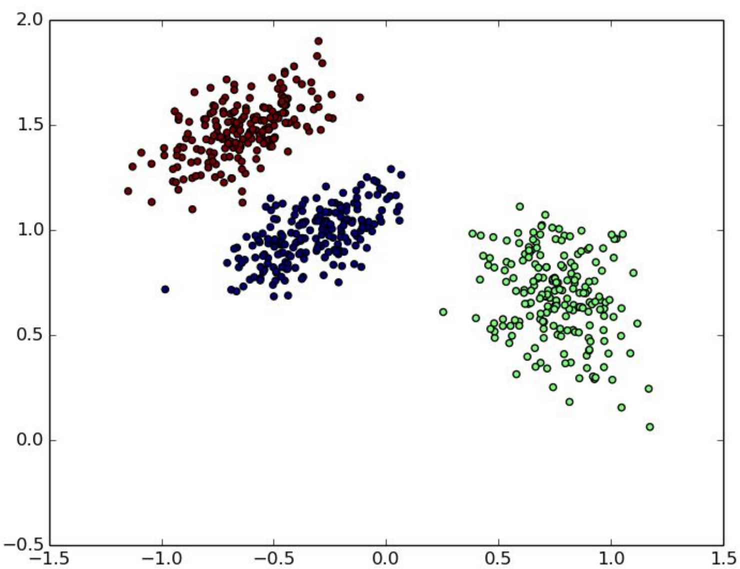


Implement Gaussian Mixture Model

a)



b)



Variance 1:
[0.0359528   0.01551689
0.01551689 0.0193642]
Variance 2:
[0.02716617 -0.00839784
-0.00839784 0.04044061]
Variance 3:
[0.03600029 0.0146632
0.0146632   0.01628779]

Mean 1:
[-0.6395396   1.47455763]
Mean 2:
[0.75896585 0.67976677]
Mean 3:
[-0.32579627 0.97130734]