1. (a)

I did this homework with Luning Zhao.

We work on our homework seperately, but we discuss when meeting problems.

Homework is fine.

(b) I certify that all solutions are entirely in my words. and that I have not looked at another student's solutions. I have credited all external sources in this write up.

Siyao Jia

2. (a) When $k=n$, $\mu_i = \frac{1}{|C_i|} \sum_{x_i \in C_i} x_i = \frac{1}{|C_i|} x_i = x_i$

$$\sum_{i=1}^{k} \sum_{x_i \in C_i} \| x_j - \mu_i \|_2^2 = \sum_{i=1}^{n} (\mu_i - x_i) = 0$$

So the minimum value of object is 0 when $k=n$.

(b) $f(\mu_i) = \lambda \|\mu_i\|^2 + \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$

$= \lambda \mu_i^T \mu_i + \sum (x_j - \mu_i)^T (x_j - \mu_i)$

$$\frac{\partial f(\mu_i)}{\partial \mu_i} = 2\lambda \mu_i^T + \sum -2 x_j^T + 2 \mu_i^T = 2\lambda \mu_i^T + 2|C_i| \mu_i^T + \sum -2 x_j^T$$

Let $\frac{\partial f(\mu_i)}{\partial \mu_i} = 0 \Rightarrow \mu_i = \frac{\sum x_j}{\lambda + |C_i|}$

(c) Assume the k meet locations are μ_i where $i \in [1, k]$

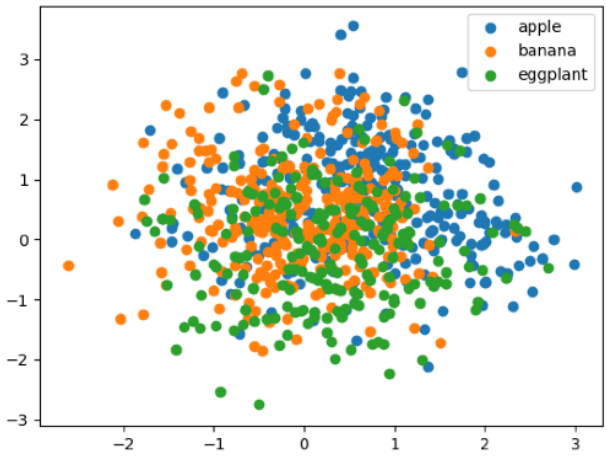
The distance for students near μ_i to drive to μ_i is $\sum_{x_j \in C_i} \|x_j - \mu_i\|^2$

The distance for vehicle to drive from μ_i to $(0,0)$ is $\|\mu_i\|^2$.

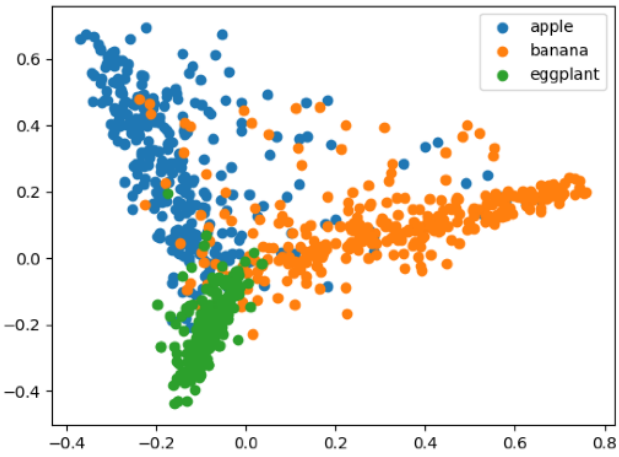
So the objective function is $\min_{C_1 \cdots C_k} \sum_{i=1}^{k} (\lambda \|\mu_i\|^2 + \sum_{x_j \in C_i} \|x_j - \mu_i\|^2)$

(d) Set $i(j) = \text{argmin}_i \frac{1}{|C_i|} \sum_{x_i \in C_i} k(x_j, x_i)$

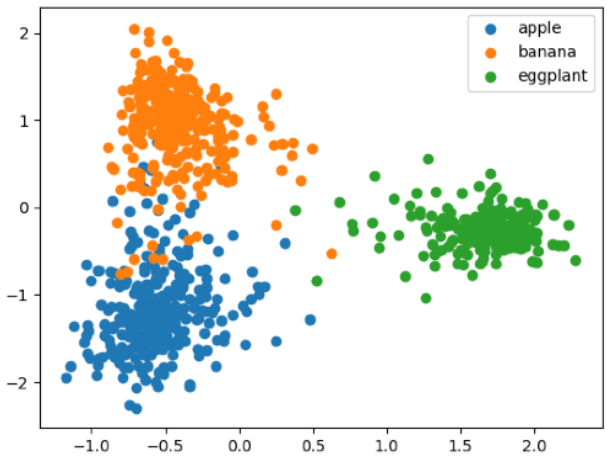
3 (a)



(b)

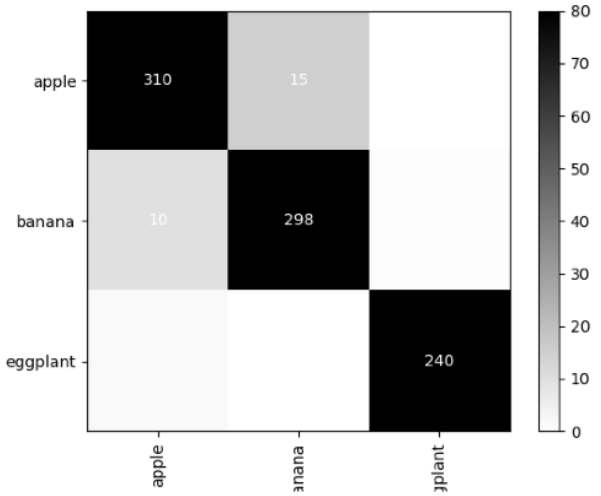


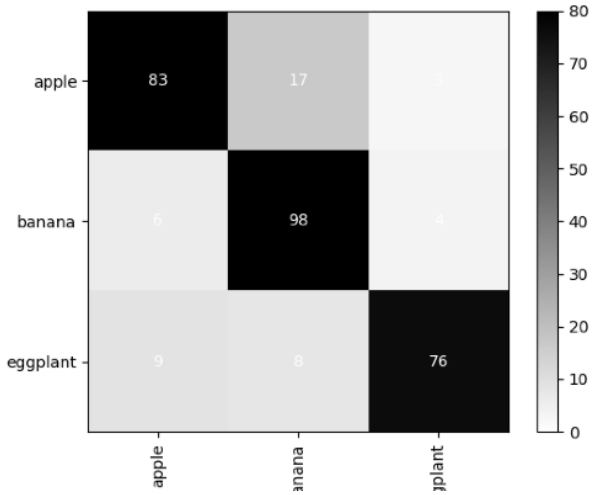
(c)



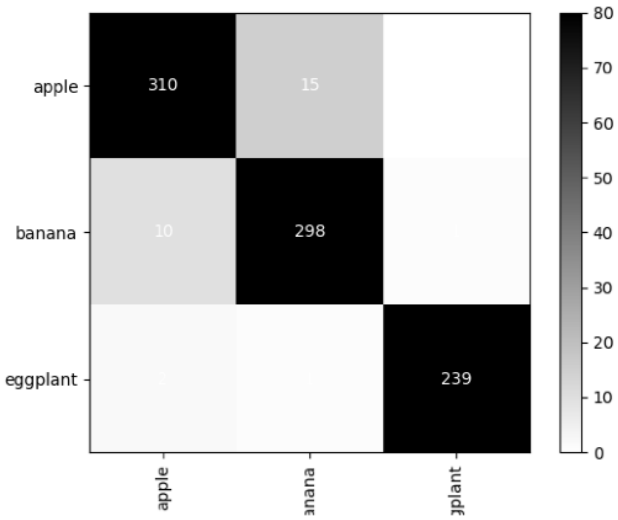
CCA is the best for separation among classes. Random is the worst. Because CCA finds the dimension that correlates most with the class.

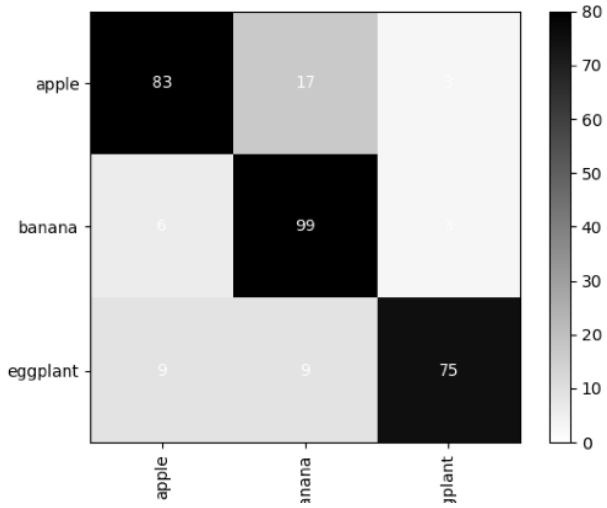
(d) training: validation:





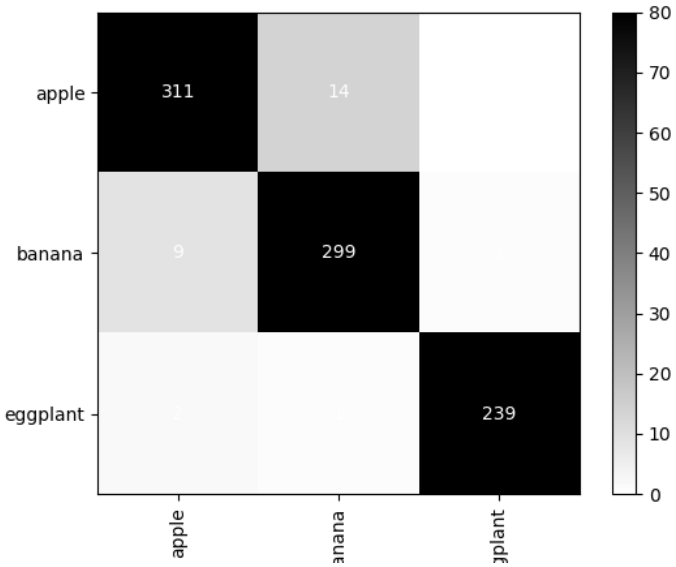
(e) training: validation:

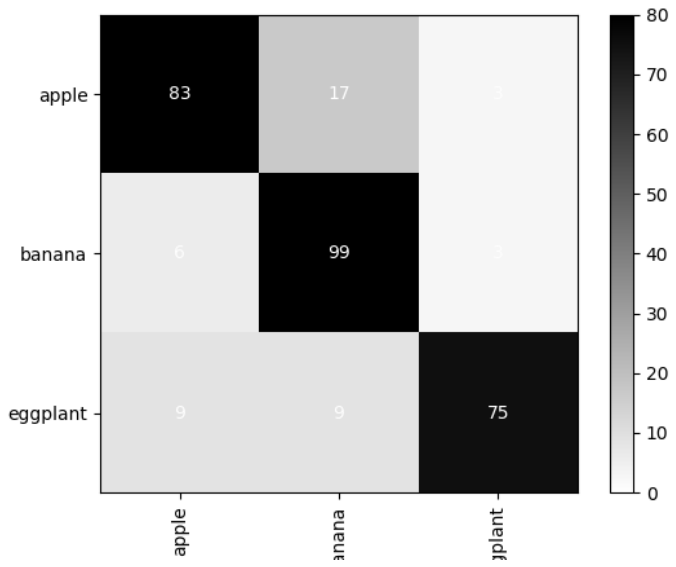





```
class LDA_Model():

    def __init__(self,class_labels):
        ###SCALE AN IDENTITY MATRIX BY THIS TERM AND ADD TO COMPUTED COVARIA
        self.reg_cov = 0.001
        self.NUM_CLASSES = len(class_labels)



    def train_model(self,X,Y):
        ''''
        FILL IN CODE TO TRAIN MODEL
        MAKE SURE TO ADD HYPERPARAMTER TO MODEL
        '''
        self.u = []
        for i in range(self.NUM_CLASSES):
            idx = np.where(np.array(Y)==i)[0]
            self.u.append(np.mean(np.array(X)[idx], axis=0))
        self.sig = np.cov(np.array(X).T)
        self.sig += np.identity(len(self.sig))*self.reg_cov


    def eval(self,x):
        ''''
        Fill in code to evaluate model and return a prediction
        Prediction should be an integer specifying a class
        '''
        y = []
        for i in range(self.NUM_CLASSES):
            y.append((x-self.u[i]) @ inv(self.sig) @ (x-self.u[i]).T)
        pdb.set_trace()
        return np.argmin(y)
```

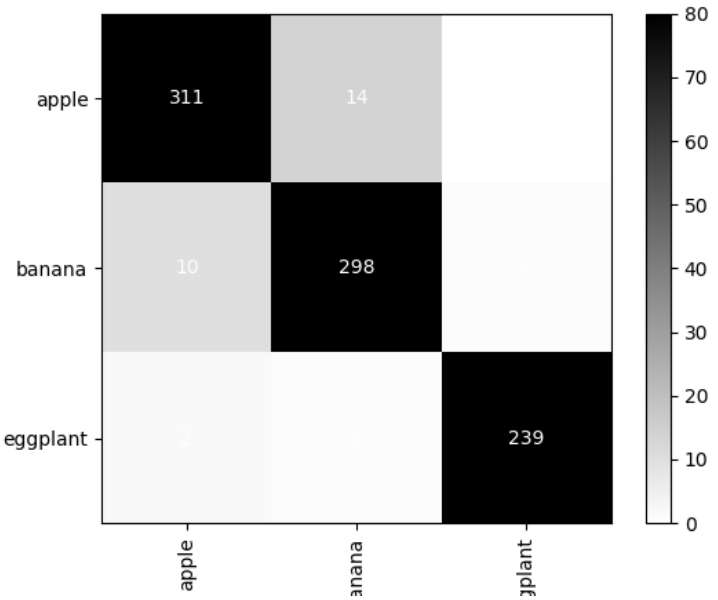
(f) training: validation:

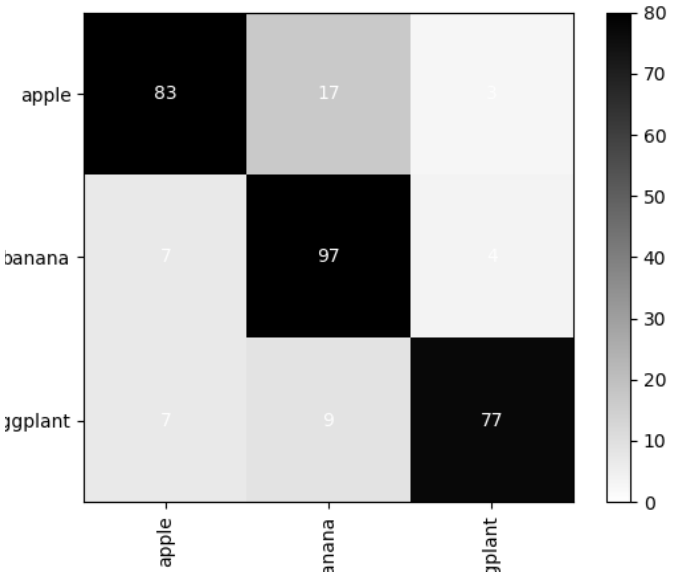




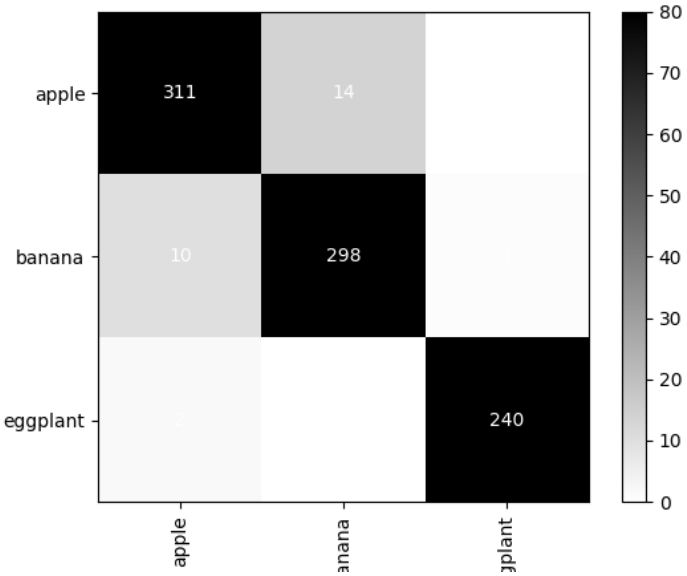

```python
class QDA_Model():
    def __init__(self,class_labels):
        ###SCALE AN IDENTITY MATRIX BY THIS TERM AND ADD TO COMPUTED COVARIANCE MATRIX TO PREVENT IT B
        self.reg_cov = 0.01
        self.NUM_CLASSES = len(class_labels)



    def train_model(self,X,Y):
        ''''
        FILL IN CODE TO TRAIN MODEL
        MAKE SURE TO ADD HYPERPARAMTER TO MODEL

        '''
        self.u = []
        self.sig = []
        for i in range(self.NUM_CLASSES):
            idx = np.where(np.array(Y)==i)[0]
            self.u.append(np.mean(np.array(X)[idx], axis=0))
            sig = np.cov(np.array(X)[idx].T)
            sig += np.identity(len(sig))*self.reg_cov
            self.sig.append(sig)

    def eval(self,x):
        ''''
        Fill in code to evaluate model and return a prediction
        Prediction should be an integer specifying a class
        '''
        y = []
        for i in range(self.NUM_CLASSES):
            y.append((x-self.u[i]) @ inv(self.sig[i]) @ (x-self.u[i]).T + np.log(det(self.sig[i])))
        return np.argmin(y)
```

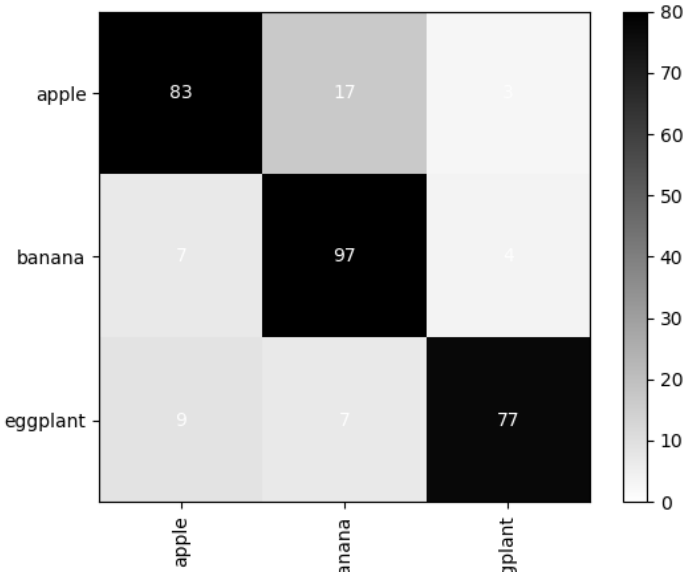
(g) training: validation:



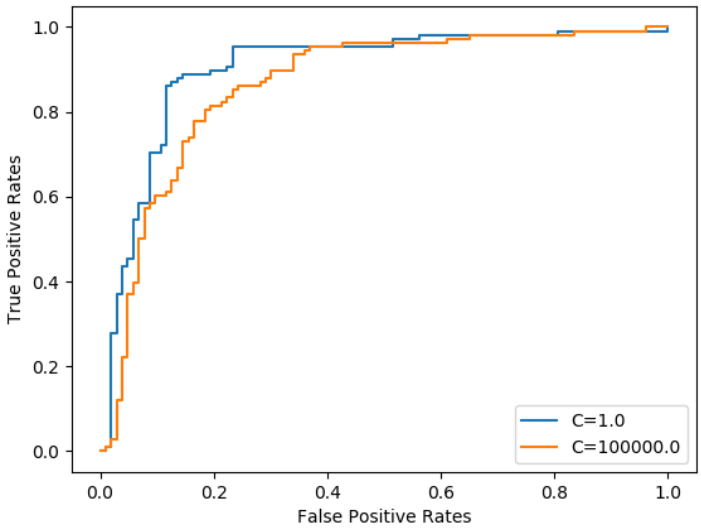


(h) training: validation:





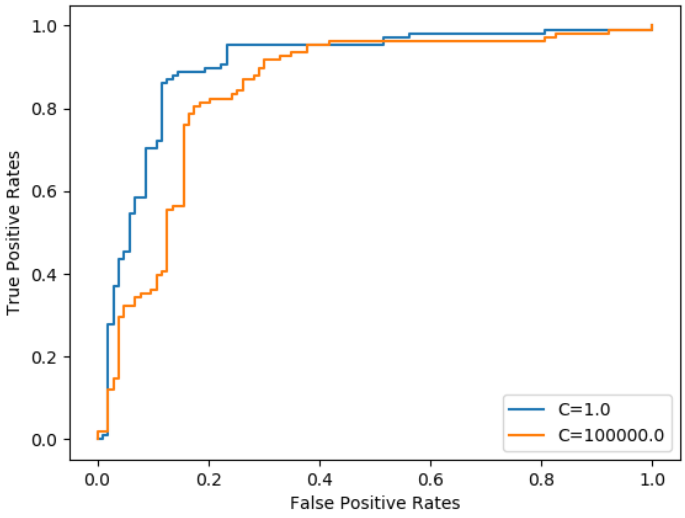
(i)



C=1 is better

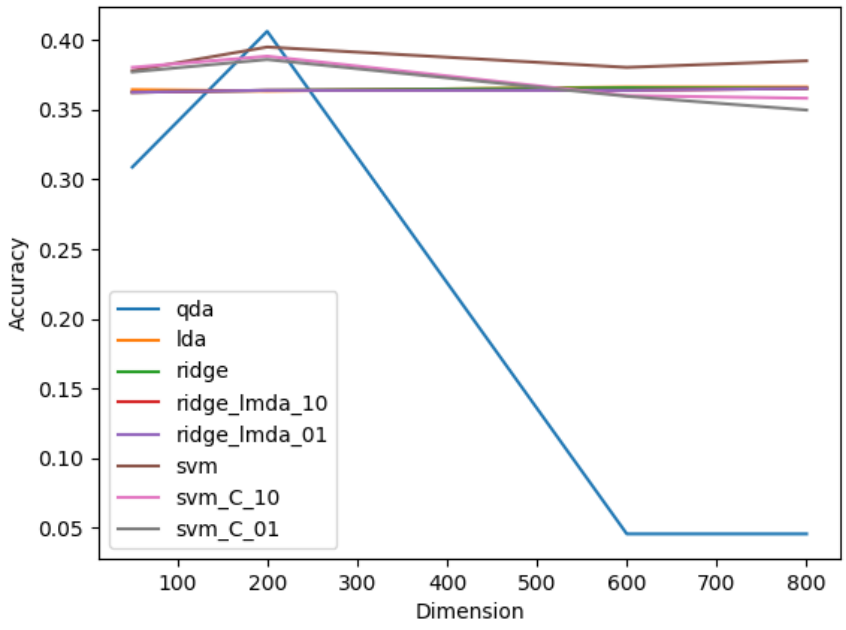
```
def ROC(scores, labels):
        thresholds = sorted(np.unique(scores))
        thresholds = [-float("Inf")] + thresholds + [float("Inf")]
        tps = []
        fps = []

        # student code start here
        # TODO: Your code
        # student code end here
        for thres in thresholds:
            tpr, fpr = compute_tp_fp(thres, scores, labels)
            tps.append(tpr)
            fps.append(fpr)
        return tps, fps
```

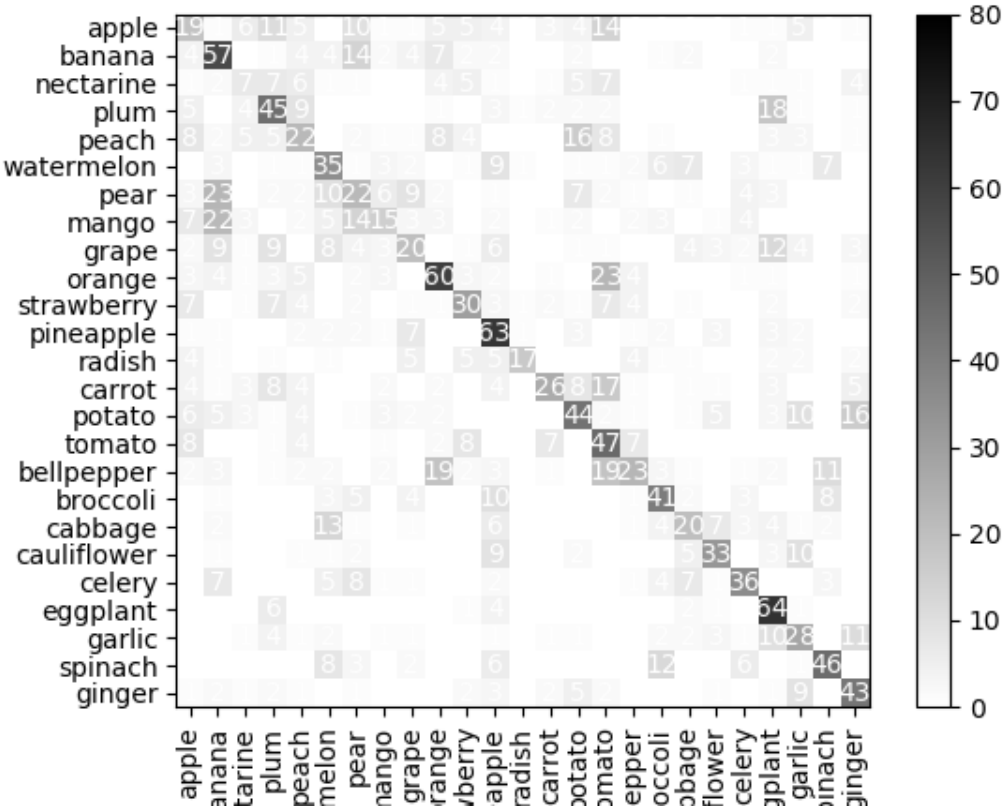
(j)



C=100000
becomes worse

(k)


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
Accuracy
qda
lda
ridge
ridge_lmda_10
ridge_lmda_01
svm
svm_C_10
svm_C_01
100
200
300
400
500
600
700
800
Dimension


apple
banana
nectarine
plum
peach
watermelon
pear
mango
grape
orange
strawberry
pineapple
radish
carrot
potato
tomato
bellpepper
broccoli
cabbage
cauliflower
celery
eggplant
garlic
spinach
ginger
80
70
60
50
40
30
20
10
0

4. (a) $P(X=x, Z=1|\theta) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \times 0.5$

$P(X=x, Z=2|\theta) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \times 0.5$

$P(X=x;\theta) = \frac{1}{2\sqrt{2\pi}} \left(\frac{1}{\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + \frac{1}{\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \right)$

$\ell(X=x;\theta) = -\log\sigma_1 - \frac{(x-\mu_1)^2}{2\sigma_1^2} - \log\sigma_2 - \frac{(x-\mu_2)^2}{2\sigma_2^2} - \log(2\sqrt{2\pi})$

(b) $\ell(X_1=x_1, \cdots X_n=x_n) = \sum_{i=1}^{n} \ell(X_i=x_i)$

$= \sum_{i=1}^{n} \left[-\log\sigma_1 - \frac{(x_i-\mu_1)^2}{2\sigma_1^2} - \log\sigma_2 - \frac{(x_i-\mu_2)^2}{2\sigma_2^2} \right] + c$

(c) $\ell(x_i|\theta) = -\log P(x_i;\theta)$

$= \log\left(\sum_{k=1}^{2} P(x_i, k;\theta)\right)$

$= \log\left(\sum_{k=1}^{2} \frac{P(x_i,k;\theta)}{q_i(k)} q_i(k)\right)$

$\geq \sum_{k=1}^{2} q_i(k) \log\frac{P(x_i,k;\theta)}{q_i(k)}$

$= \sum_{k=1}^{2} q_i(k) \log P(x_i,k;\theta) + \sum_{i=1}^{2} q_i(k) \log\left(\frac{1}{q_i(k)}\right) = F_i(\theta; q_i)$

$\ell(\{x_i\};\theta) = \sum_i \log P(x_i;\theta)$

$\geq \sum_{i=1}^{n} F_i(\theta; q_i) \quad = F(\theta; q)$

(d) $q(z_1=z_1, \cdots, z_n=z_n) = \prod_{i=1}^{n} P(z=z_i|x=x_i;\theta^t) = \prod_{i=1}^{n} q_i(z_i=z_i)$

$\Rightarrow q_i(z_i=z_i) = P(z=z_i|x=x_i;\theta^t)$

$F_i(\theta^t; q_i) = \sum_{k=1}^{2} q_i(k) \log\frac{P(x_i,k;\theta^t)}{q_i(k)}$

$= \sum_{k=1}^{2} q_i(k) \log\frac{P(x_i,k;\theta^t)}{P(z=z_i|x=x_i,\theta^t)}$

$= \sum_{k=1}^{2} q_i(k) P(x=x_i|\theta^t) = \ell(x_i|\theta)$

So $\ell(\{x_i\};\theta^t) = \sum_i F_i(\theta^t; q_i) = F(\theta^t; q)$

So $q^{t+1}(z_1=z_1, \cdots, z_n=z_n) = \prod_{i=1}^{n} P(z=z_i|x=x_i;\theta^t)$ is a valid maximizer for E-step.

(e) $q_i^{t+1}(z_i=1) = P(z=1 \mid x=x_i; \theta^t)$

$$= \frac{P(z=1 \mid x=x_i;\theta^t)}{P(z=1 \mid x=x_i;\theta^t) + P(z=2 \mid x=x_i;\theta^t)}$$

$$= \frac{\frac{1}{\sigma_1}\exp\left(-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}\right)}{\frac{1}{\sigma_1}\exp\left(-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}\right) + \frac{1}{\sigma_2}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)}$$

Similarly $q_i^{t+1}(z_i=2) = \dfrac{\frac{1}{\sigma_2}\exp\left(-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}\right)}{\frac{1}{\sigma_1}\exp\left(-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}\right) + \frac{1}{\sigma_2}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)}$

Intuitively, we want the distribution of $P(z=k)$ proportional to the probability $P(x \mid z, \theta^t)$. So this update makes sense

(f) $L((\mu_1, \mu_2, \sigma_1, \sigma_2; q^{t+1})$

$$= \sum_{i=1}^{n} L(x_i; \theta, q_i^{t+1})$$

$$= \sum_{i=1}^{n} \left(q_i^{t+1} \log P(x_i; k=1, \theta) + (1-q_i^{t+1}) \log P(x_i; k=2, \theta) \right)$$

$$= \sum_{i=1}^{n} \left(q_i^{t+1} \left(-\log\sigma_1 - \frac{(x_i-\mu_1)^2}{2\sigma_1^2}\right) + (1-q_i^{t+1}) \left(-\log\sigma_2 - \frac{(x_i-\mu_2)^2}{2\sigma_2^2}\right)\right) + C$$

$$= C - \sum_{i=1}^{n} \left[q_i^{t+1} \left(\frac{(x_i-\mu_1)^2}{2\sigma_1^2} + \log\sigma_1\right) + (1-q_i^{t+1}) \left(\frac{(x_i-\mu_1)^2}{2\sigma_2^2} + \log\sigma_2\right)\right]$$

(g) $\dfrac{\partial L}{\partial \mu_1} = -\sum_{i=1}^{n} q_i^{t+1} \dfrac{2(x_i-\mu_1)(-1)}{2\sigma_1^2} = -\dfrac{\sum_{i=1}^{n} q_i^{t+1}(\mu_1 - x_i)}{\sigma_1^2}$

Similarly, $\dfrac{\partial L}{\partial \mu_2} = -\dfrac{\sum_{i=1}^{n} (1-q_i^{t+1})(\mu_2 - x_i)}{\sigma_2^2}$

$$\frac{\partial L}{\partial \sigma_1} = -\sum_{i=1}^{n} q_i^{t+1}\left(\frac{(x_i-\mu_1)^2\cdot(-2)}{2\sigma_1^3} + \frac{1}{\sigma_1}\right) = \frac{\sum_{i=1}^{n} q_i^{t+1}(x_i-\mu_1)^2}{\sigma_1^3} - \frac{\sum_{i=1}^{n} q_i^{t+1}}{\sigma_1}$$

Similarly, $\dfrac{\partial L}{\partial \sigma_2} = \dfrac{\sum_{i=1}^{n} (1-q_i^{t+1})(x_i-\mu_2)^2}{\sigma_2^3} - \dfrac{\sum_{i=1}^{n} (1-q_i^{t+1})}{\sigma_2}$

(h) $\frac{\partial L}{\partial \mu_1} = -\frac{\sum_{i=1}^{n} q_i^{t+1}(\mu_1 - x_i)}{\sigma_1^2} = 0 \Rightarrow \sum_{i=1}^{n} q_i^{t+1}(\mu_1 - x_i) = 0$

$$\Rightarrow \mu_1^{t+1} = \frac{\sum_{i=1}^{n} q_i^{t+1} x_i}{\sum_{i=1}^{n} q_i^{t+1}}$$

Similarly, $\frac{\partial L}{\partial \mu_2} = 0 \Rightarrow \mu_2^{t+1} = \frac{\sum_{i=1}^{n} (1 - q_i^{t+1}) x_i}{\sum_{i=1}^{n} (1 - q_i^{t+1})}$

$$\frac{\partial L}{\partial \sigma_1} = \frac{\sum_{i=1}^{n} q_i^{t+1}(x_i - \mu_1)^2}{\sigma_1^3} - \frac{\sum_{i=1}^{n} q_i^{t+1}}{\sigma_1} = 0 \Rightarrow \frac{\sum_{i=1}^{n} q_i^{t+1}(x_i - \mu_1)^2}{\sigma_1^2} = \sum_{i=1}^{n} q_i^{t+1}$$

$$\Rightarrow (\sigma_1^2)^{t+1} = \frac{\sum_{i=1}^{n} q_i^{t+1}(x_i - \mu_1^{t+1})^2}{\sum_{i=1}^{n} q_i^{t+1}}$$

Similarly, $\frac{\partial L}{\partial \sigma_2} = 0 \Rightarrow (\sigma_2^2)^{t+1} = \frac{\sum_{i=1}^{n} (1 - q_i^{t+1})(x_i - \mu_2^{t+1})^2}{\sum_{i=1}^{n} (1 - q_i^{t+1})}$

Intuitively, the new μ_i is the weighted mean of all the points, and σ_i is the weighted variance of all points, so it makes sense.

(i) $l(\{x_i\}; \mu) = \sum_{i=1}^{n} \log(P(x_i; \mu))$

$$= \sum_{i=1}^{n} \log\left(\frac{1}{2}\frac{e^{-\frac{1}{2}(x-\mu)^2}}{\sqrt{2\pi}} + \frac{1}{2}\frac{e^{-\frac{1}{2}(x+\mu)^2}}{\sqrt{2\pi}}\right)$$

(j) $q^{t+1}(z_1 = z_1, \cdots z_n = z_n) = \prod_{i=1}^{n} P(z = z_i | x = x_i; \theta^t)$ Eq(4)

$$\Rightarrow q_i^{t+1}(z=1) = P(z=1 | x = x_i; \theta^t) = \frac{P(z=1 | x = x_i; \theta^t)}{P(z=1 | x = x_i; \theta^t) + P(z=2 | x = x_i, \theta^t)}$$

$$= \frac{\exp\left(-\frac{(x_i - \mu_1^t)^2}{2}\right)}{\exp\left(-\frac{(x_i - \mu_1^t)^2}{2}\right) + \exp\left(-\frac{(x_i + \mu_1^t)^2}{2}\right)}$$

(k) $L(\mu; q^{(t+1)}) = C - \sum_{i=1}^{n} \left[q_i^{(t+1)} \left(\frac{(x_i-\mu_1)^2}{2\sigma_1^2} + \log \sigma_1 \right) + (1-q_i^{(t+1)}) \left(\frac{(x_i-\mu_2)^2}{2\sigma_2^2} + \log \sigma_2 \right) \right]$

plug in $\mu_1 = \mu$, $\mu_2 = -\mu$, $\sigma_1 = \sigma_2 = 1$

$$L(\mu; q^{(t+1)}) = C - \sum_{i=1}^{n} \left[q_i^{(t+1)} \frac{(x_i-\mu)^2}{2} + (1-q_i^{(t+1)}) \frac{(x_i+\mu)^2}{2} \right]$$

$$\frac{dL(\mu; q^{(t+1)})}{d\mu} = -\sum_{i=1}^{n} \left[q_i^{(t+1)} \frac{2(x_i-\mu)(-1)}{2} + (1-q_i^{(t+1)}) \frac{2(x_i+\mu)}{2} \right]$$

$$= -\sum_{i=1}^{n} \left[-q_i^{(t+1)} (x_i-\mu) + (1-q_i^{(t+1)}) (x_i+\mu) \right]$$

$$= -\sum_{i=1}^{n} \left[-q_i^{(t+1)} x_i + q_i^{(t+1)} \mu + (1-q_i^{(t+1)}) x_i + (1-q_i^{(t+1)}) \mu \right]$$

$$= -\sum_{i=1}^{n} \left[(1-2q_i^{(t+1)}) x_i + \mu \right] = 0$$

$$\Rightarrow \quad \mu^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} (2q_i^{(t+1)} - 1) x_i$$

(l) $$\frac{d(l(\{x_i\}; \mu)/n)}{d\mu} = \frac{\frac{1}{n} d\left(\sum_{i=1}^{n} \log \left(\frac{1}{2} \frac{e^{-\frac{1}{2}(x-\mu)^2}}{\sqrt{2\pi}} + \frac{1}{2} \frac{e^{-\frac{1}{2}(x+\mu)^2}}{\sqrt{2\pi}} \right) \right)}{d\mu}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{e^{-\frac{1}{2}(x-\mu)^2}(-2)(x-\mu) + e^{-\frac{1}{2}(x+\mu)^2}(-1)(x+\mu)}{e^{-\frac{1}{2}(x-\mu)^2} + e^{-\frac{1}{2}(x+\mu)^2}}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{-\left(e^{-\frac{1}{2}(x-\mu)^2} + e^{-\frac{1}{2}(x+\mu)^2} \right) \mu + x \left(e^{-\frac{1}{2}(x-\mu)^2} - e^{-\frac{1}{2}(x+\mu)^2} \right)}{e^{-\frac{1}{2}(x-\mu)^2} + e^{-\frac{1}{2}(x+\mu)^2}}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(\frac{e^{-\frac{1}{2}(x-\mu)^2} - e^{-\frac{1}{2}(x+\mu)^2}}{e^{-\frac{1}{2}(x-\mu)^2} + e^{-\frac{1}{2}(x+\mu)^2}} x - \mu \right)$$

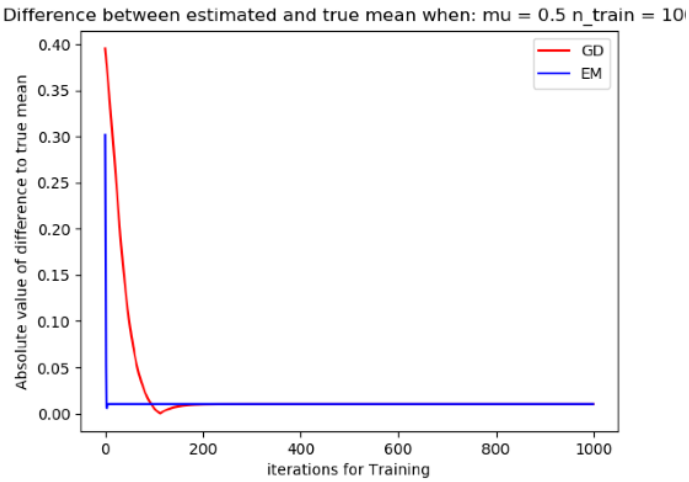
$$= \left[\frac{1}{n} \sum_{i=1}^{n} (2w_i - 1) x_i \right] - \mu \qquad \text{where } w_i(\mu) = \frac{e^{-\frac{(x_i-\mu)^2}{2}}}{e^{-\frac{(x_i-\mu)^2}{2}} + e^{-\frac{(x_i+\mu)^2}{2}}}$$

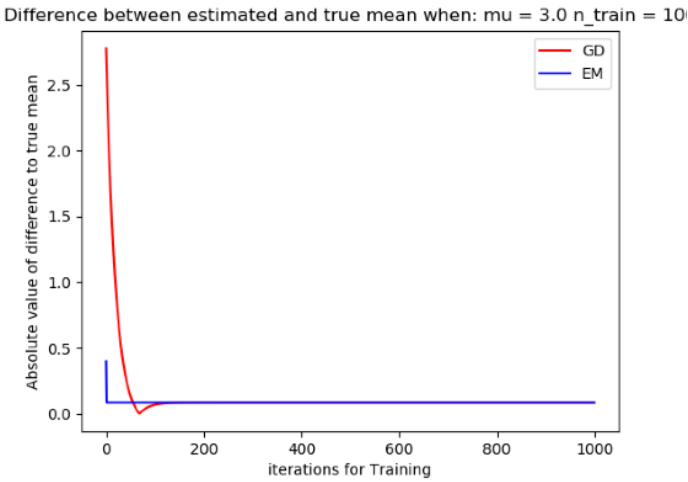
$$\mu_{t+1}^{GA} = \mu_t^{GA} + \alpha \left. \frac{d(l(\{x_i\}; \mu))}{d\mu} \right|_{\mu=\mu_t^{GA}}$$

$$= \mu_t^{GA} + \alpha \left[\left[\frac{1}{n} \sum_{i=1}^{n} \left((2w_i(\mu_t^{GA}) - 1) x_i \right] - \mu_t^{GA} \right] \right.$$

$$= (1-\alpha) \mu_t^{GA} + \alpha \left[\frac{1}{n} \sum_{i=1}^{n} (2w_i(\mu_t^{GA}) - 1) x_i \right]$$

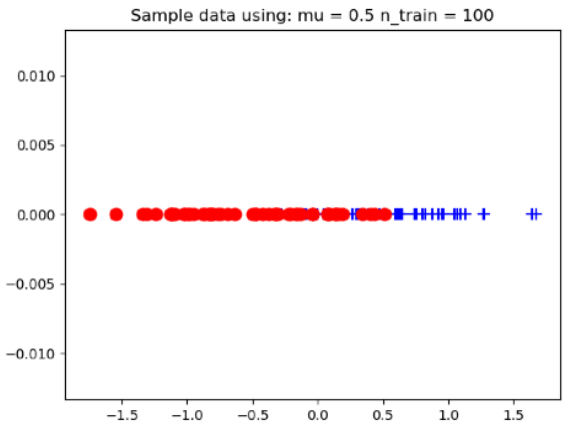
4 (m)
EM update doesn't depend on u_t directly, while GA does depend on u_t.
EM and GA all depend on xi linearly, where EM has coefficients of 2qi and GA has coefficients of 2*alpha*wi.

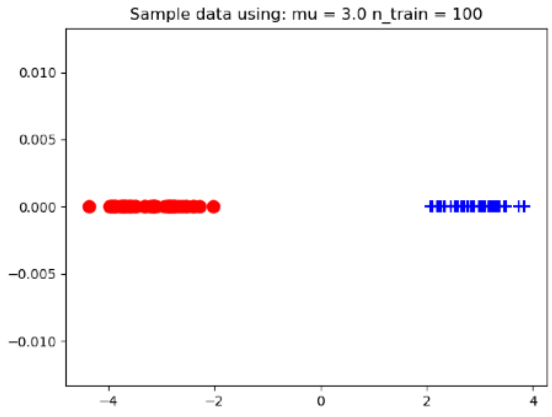




EM converges faster than GD in both cases.
Yes, they matches the updates derived previously.

4(n)
K means will work if two groups are well separated when u are large.
However, it does depend on if u is small or large.

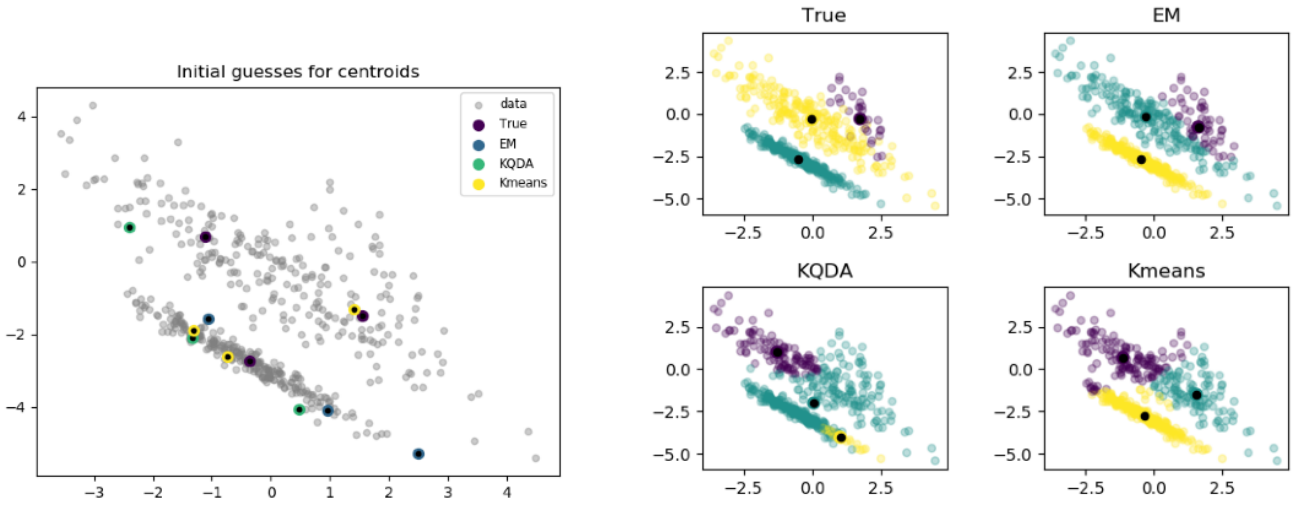





```
------------
True mean:0.500, GA (final) estimate:0.510, EM (final) estimate:0.510, K-Means (final) estimate:0.591
------------
------------
True mean:3.000, GA (final) estimate:3.085, EM (final) estimate:3.085, K-Means (final) estimate:3.085
```

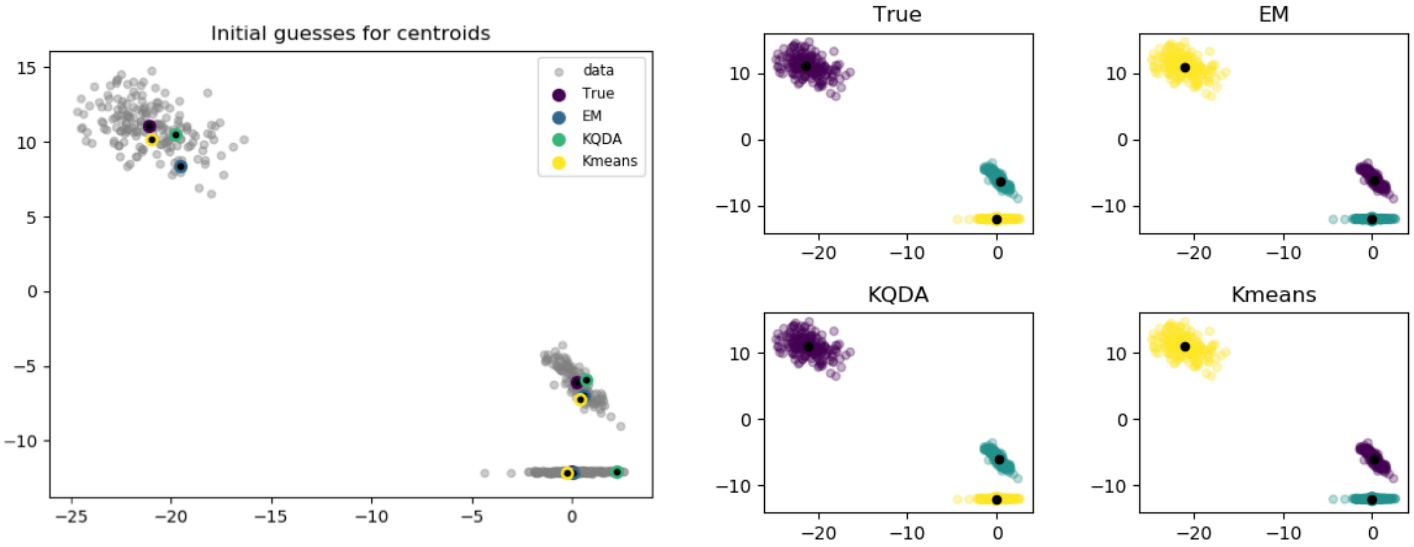
if u is small, two groups will overlap, so k means will not work well.

5. (a)



There is some overlap between groups, so Means and KQDA doesn't work very well, but EM gives a pretty satisfying classification.

(b)



When groups are well separated, all methods work pretty well.

6. Q: Suppose we want to maximize $f(x,y) = x+y$, subject to constraint $x^2+y^2=1$. Find the optimal x and y using Lagrangian multiplier.

A: max $f(x,y) = x+y$, given $g(x,y) = x^2+y^2-1 = 0$

Let $L(x,y,\lambda) = f(x,y) + \lambda\, g(x,y)$

$= x+y+\lambda(x^2+y^2-1)$

$\frac{\partial L}{\partial x} = 1+2\lambda x = 0 \Rightarrow x = -\frac{1}{2\lambda}$

$\frac{\partial L}{\partial y} = 1+2\lambda y = 0 \Rightarrow y = -\frac{1}{2\lambda}$

$\frac{\partial L}{\partial \lambda} = x^2+y^2-1 = 0 \Rightarrow (-\frac{1}{2\lambda})^2 + (-\frac{1}{2\lambda})^2 - 1 = 0 \Rightarrow \lambda = \pm\frac{1}{\sqrt{2}}$

When $\lambda = \frac{1}{\sqrt{2}}$ $x = -\frac{1}{\sqrt{2}}$, $y = -\frac{1}{\sqrt{2}}$, $f(x,y) = -\sqrt{2}$

When $\lambda = -\frac{1}{\sqrt{2}}$ $x = \frac{1}{\sqrt{2}}$, $y = \frac{1}{\sqrt{2}}$, $f(x,y) = \sqrt{2}$

So $f(x)_{max} = \sqrt{2}$ when $x = y = \frac{1}{\sqrt{2}}$.