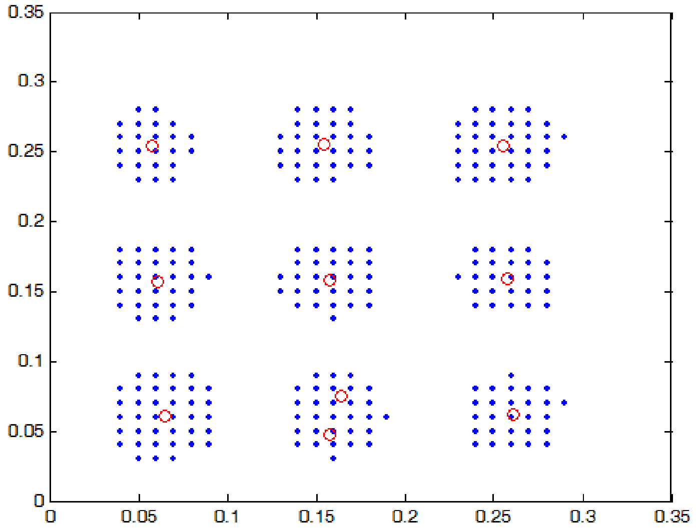
Lecture 9 Classification

- Two mappings from position to color
  - Linear combination of distances to centers
  - RBF: Linear combination of exponential distances to centers
- Four computational steps
  - (A) K-means
  - (B) Cross Distances
  - (C) Optimal Coefficients
  - (D) Coloring
- Five flow charts: OC (optimal coefficients), Coloring, ER (error rate), TRAINING & TESTING :

Kmeans: Data and MATLAB codes

data_9.zip
demo_kmeans.m

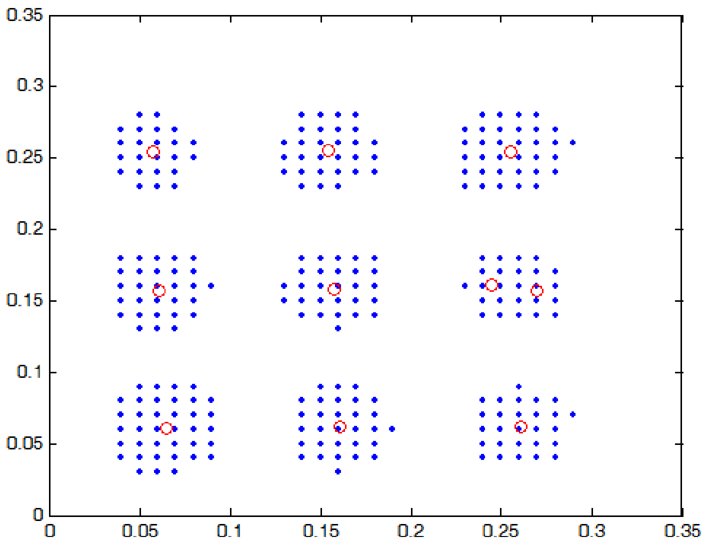
```
load data_9.mat
plot(X(:,1),X(:,2),'.');
[cidx, Y] = kmeans(X,10);
hold on;
plot(Y(:,1),Y(:,2),'ro');
```



My_kmeans: Data and MATLAB codes

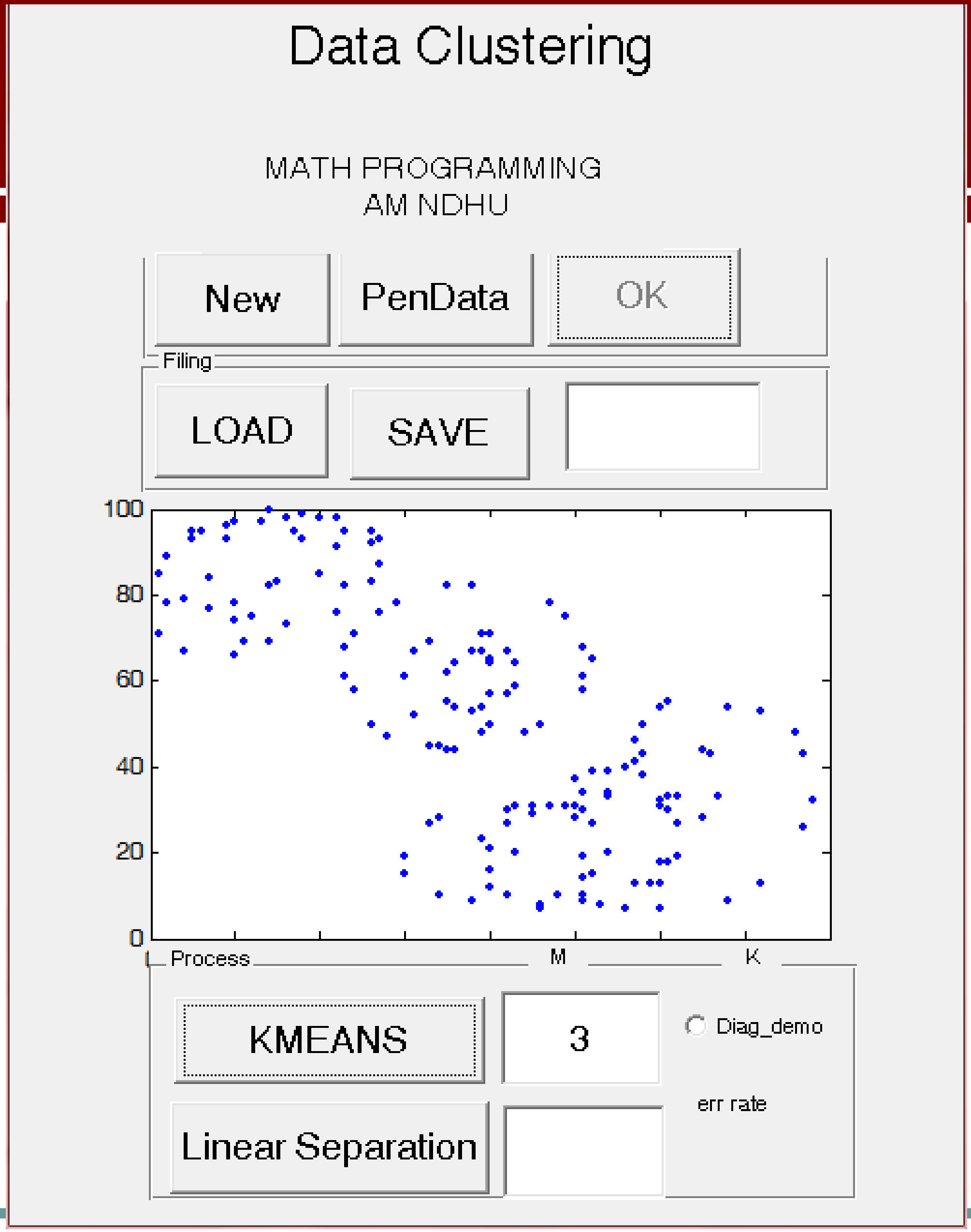
data_9.zip
demo_my_kmeans.m
my_kmeans.m

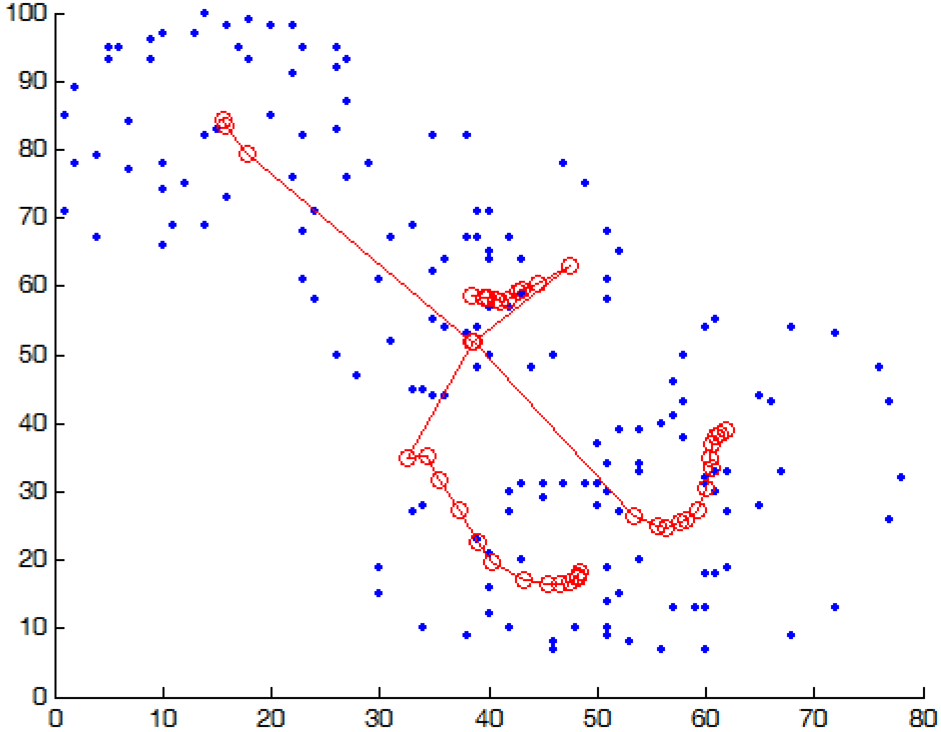
```
load data_9.mat
plot(X(:,1),X(:,2),'.');
[Y] = my_kmeans(X,10);
hold on;
plot(Y(:,1),Y(:,2),'ro');
```



GUI

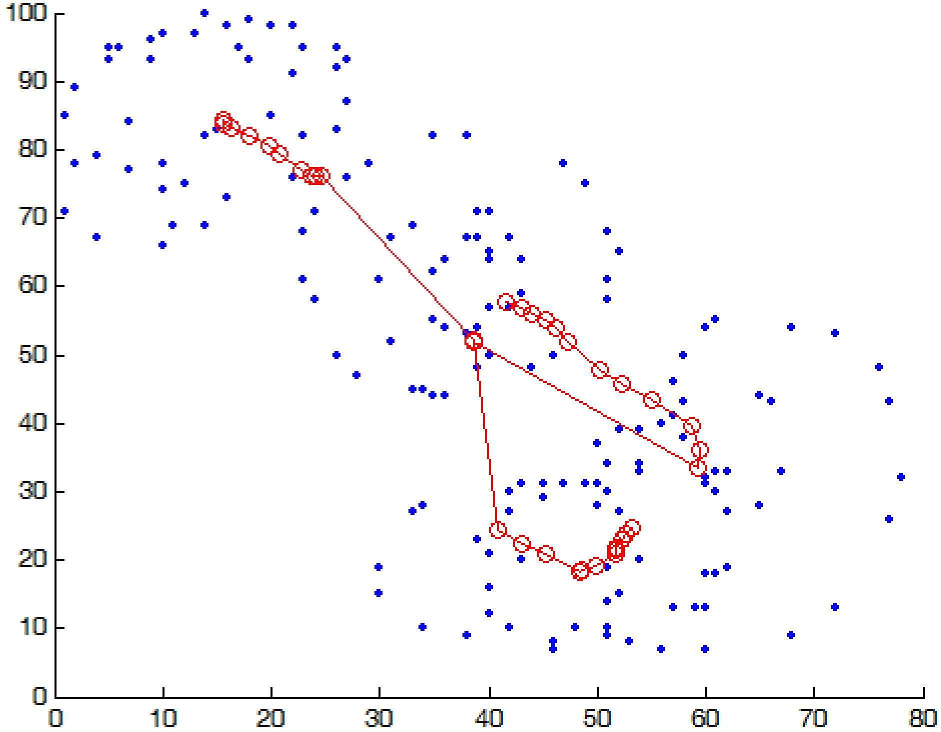
ClusteringTest.fig
ClusteringTest.m
my_kmeans.m


Data Clustering
MATH PROGRAMMING
AM NDHU
New
PenData
OK
Filing
LOAD
SAVE
100
80
60
40
20
0
Process
M
K
KMEANS
3
Diag_demo
err rate
Linear Separation



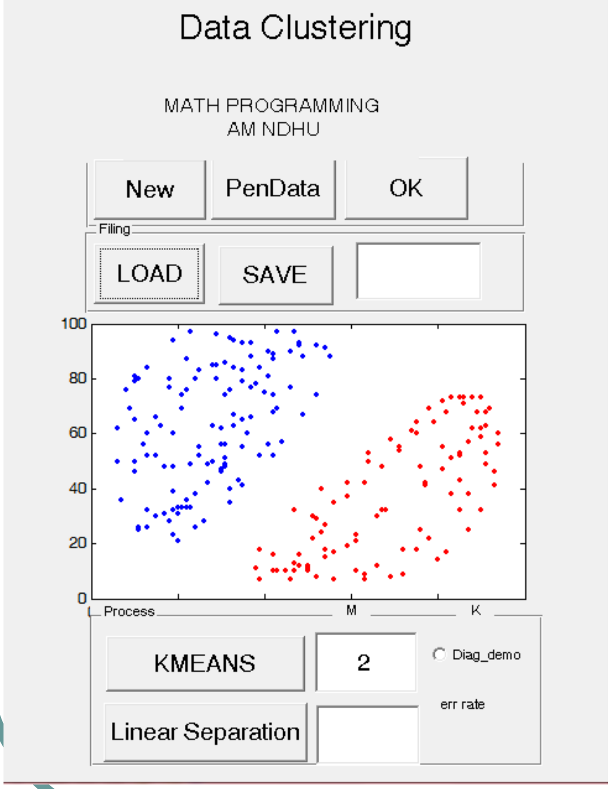
Path of 4 means to centers of clusters

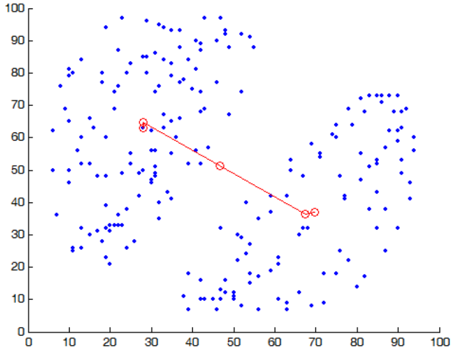
Tracking Convergence of K means

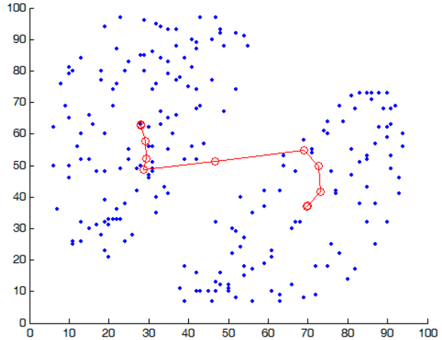


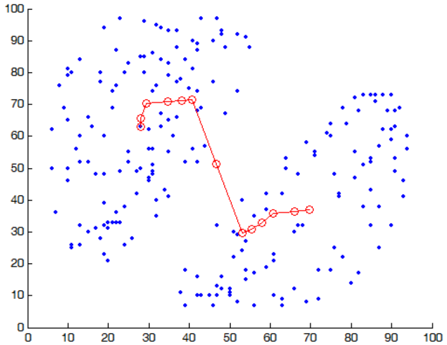
Path to converge

data_2c.zip



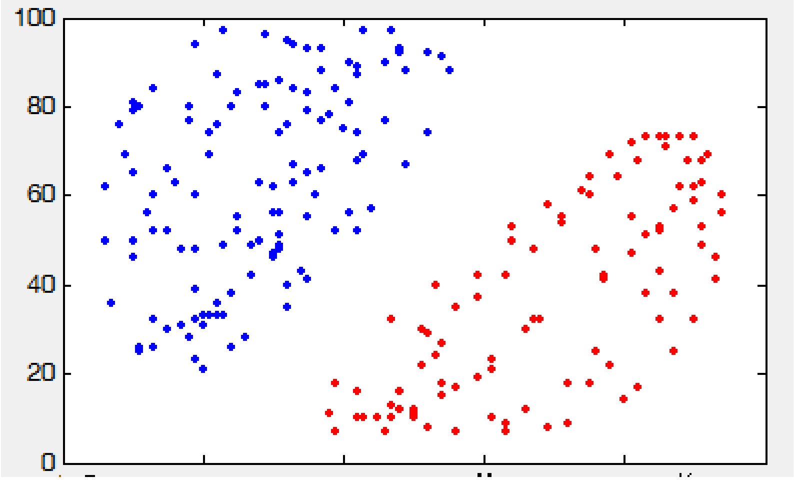






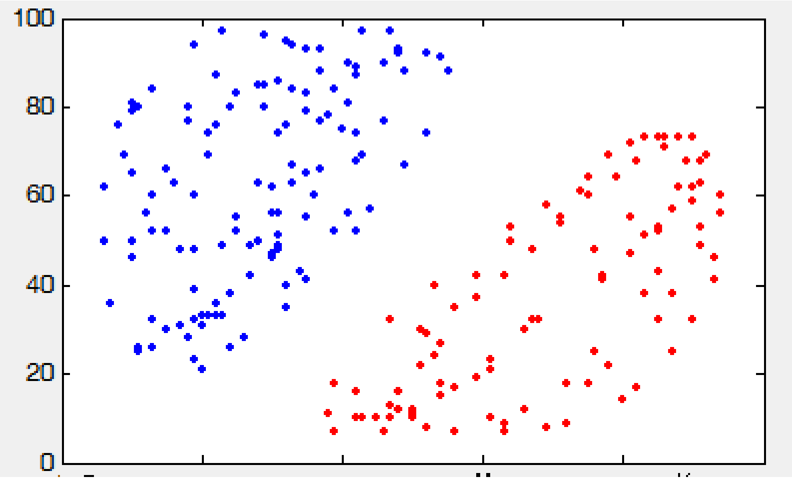
Color points

- A color point
  - POSITION
  - COLOR
- A mapping from position to color



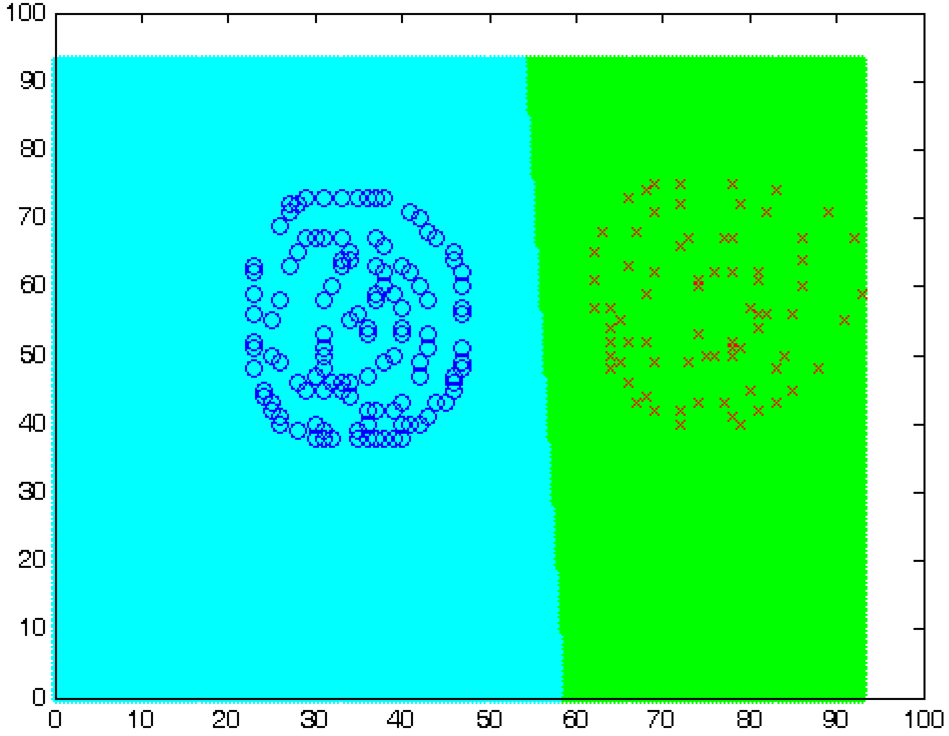
Classification of color points

- A mapping from position to color
- A rule for discriminating red points from blue points



A mapping from position to color

Each point has its **membership** to partitioned regions
Blue points and red points are well separated



A mathematical mapping

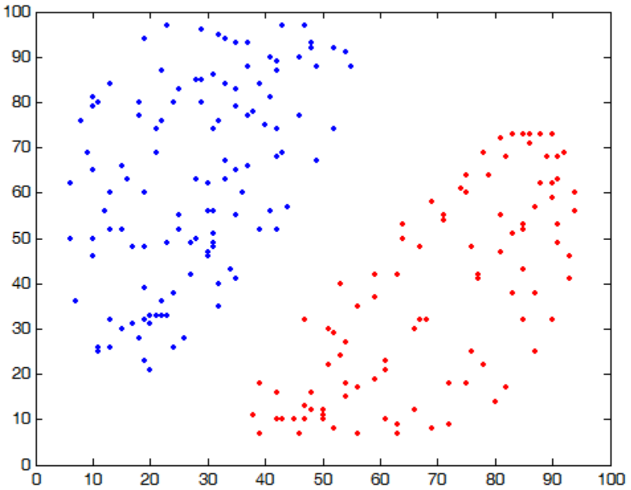
NO
TABLE LOOK-UP !!

Color points

data_2c.zip

load data_2c.mat

```
X=data_2c(:,1:2);
Y=data_2c(:,3);
```



X : positions of points

Y : 0 or 1 (blue or red color)

Two-dimensional color points

- ```
  load data_2c
  X=data_2c(:,1:2);
  Y=data_2c(:,3);
  ```
  ↓
  ```
  show_color_data(X,Y)
  ```

- Load data for classification

```
function  show_color_data(X,Y)
          figure;
          ind = find(Y==0);
          plot(X(ind,1), X(ind,2), 'b. ');
          hold on
          ind = find(Y==1);
          plot(X(ind,1), X(ind,2), 'r.')
```
```

Position & Color

- X and Y are paired data
- X collects positions of points
- Y collects colors of points
- A mapping from position to color is derived for classification of color points

Sampling for training

- Sampling one half as training points

```
N=size(X,1);
ind=randperm(N);
n = floor(N/2);
x_train=X(ind(1:n),:);
y_train=Y(ind(1:n),:);
```

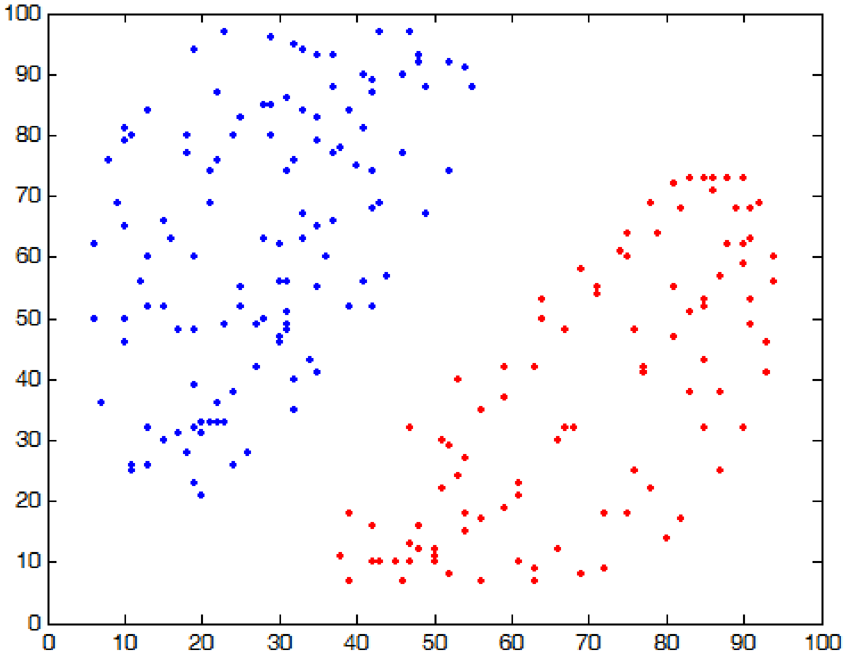
↓

```
show_color_data(x_train,y_train)
```

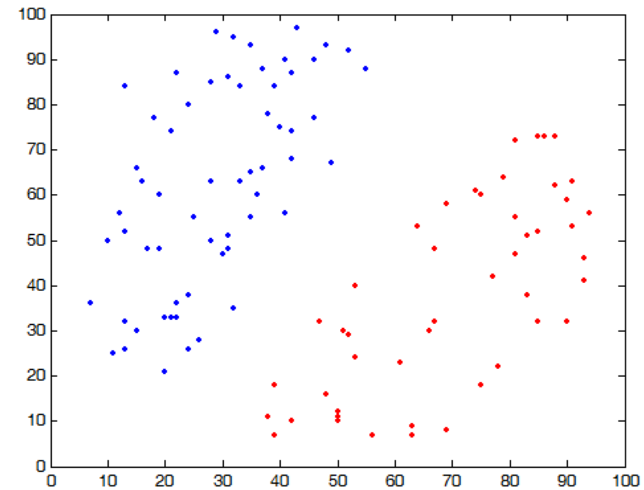
Sampling for training

X and Y

Sampling



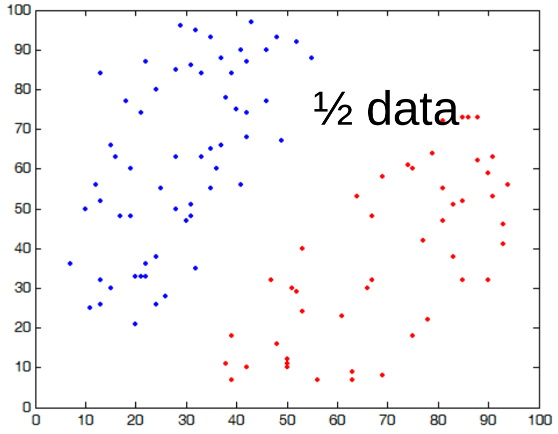
x_train and y_train



n=N/2

TRAINING

x_train and y_train



OC(optimal coefficients)
STEPS A,B,C

centers, a

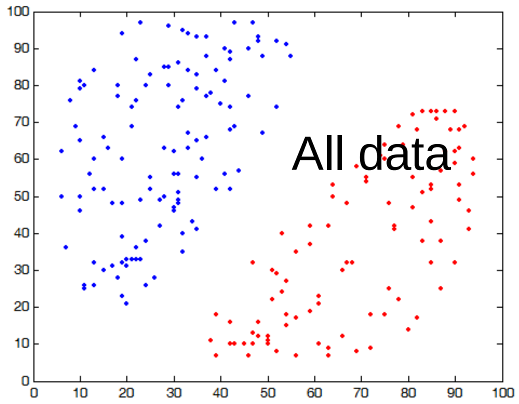
x_train → STEP D: coloring → y_hat y_train

ER (error rate)

Error rate for training

TESTING

X and Y



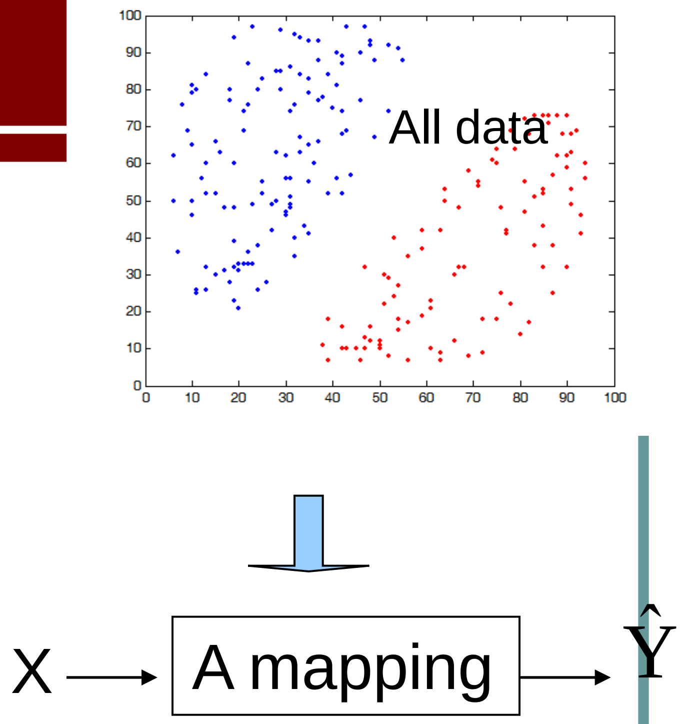
centers, a

X → STEP D: coloring → Y_hat Y

Y_hat, Y → ER (error rate)

→ Error rate for training

Error Rate



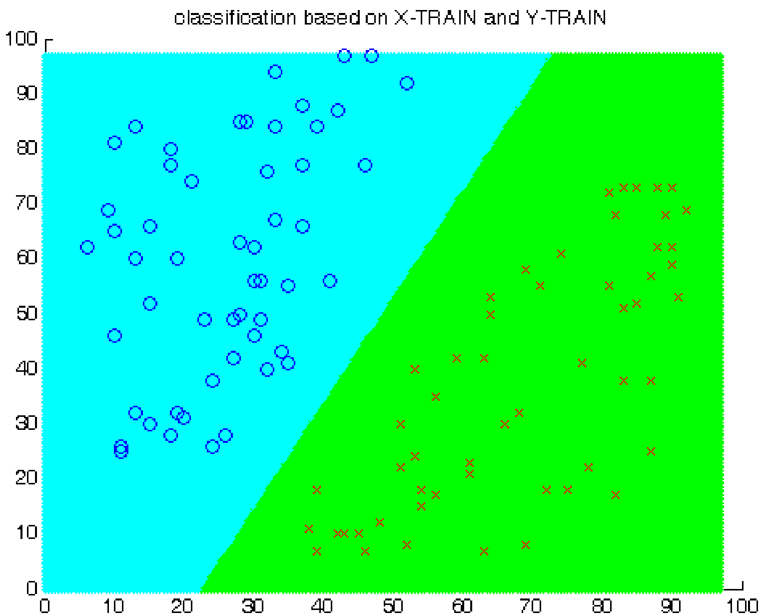
$$e_i = abs(y_i - round(\hat{y}_i))$$

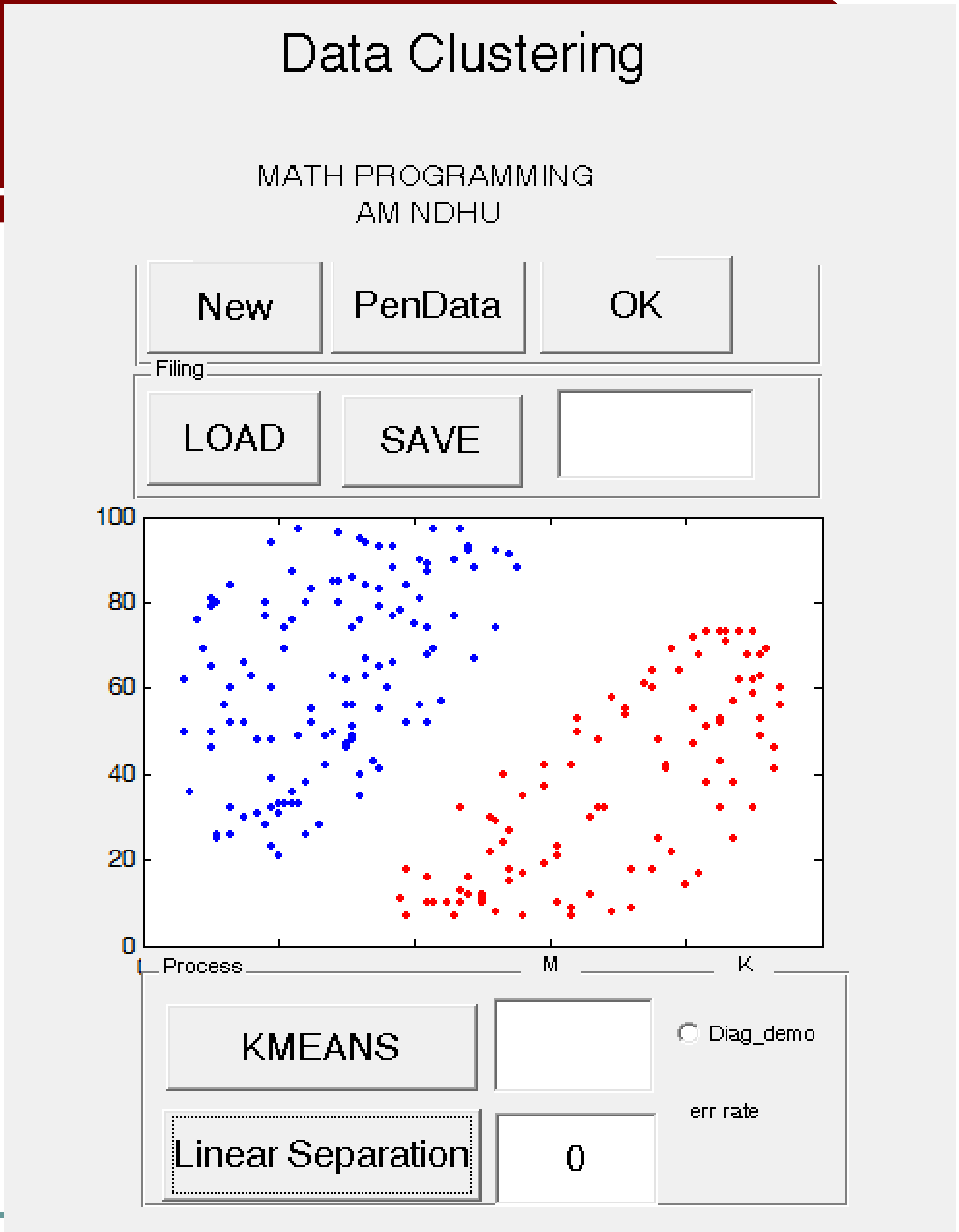
$$\frac{1}{N}\sum_i e_i$$

- Lower error rate for better testing

Linear cut

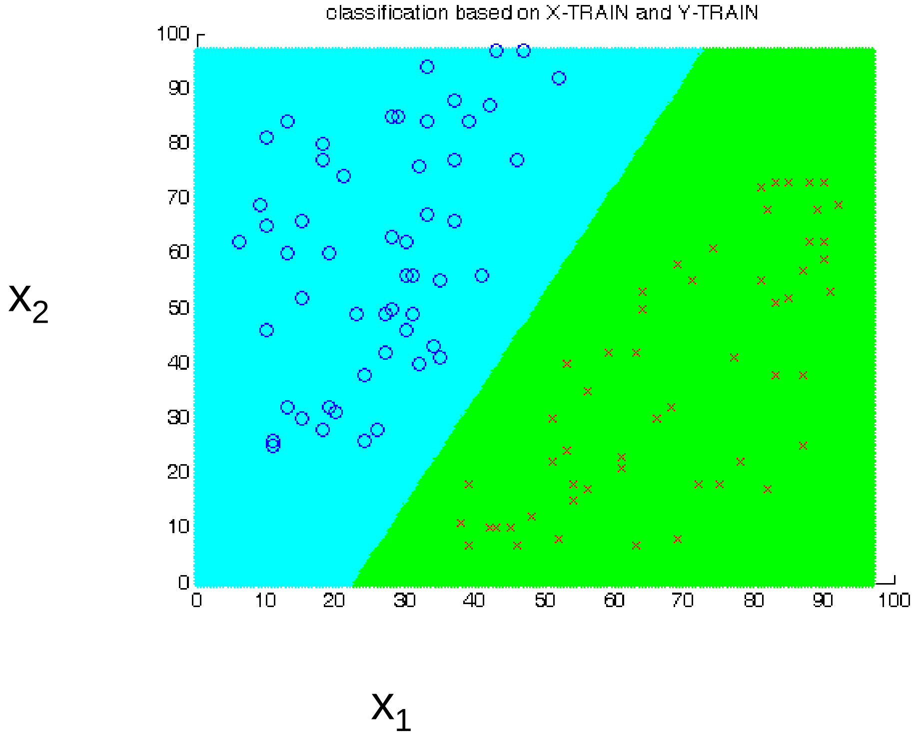
- load data_2c.txt
- error rate : zero





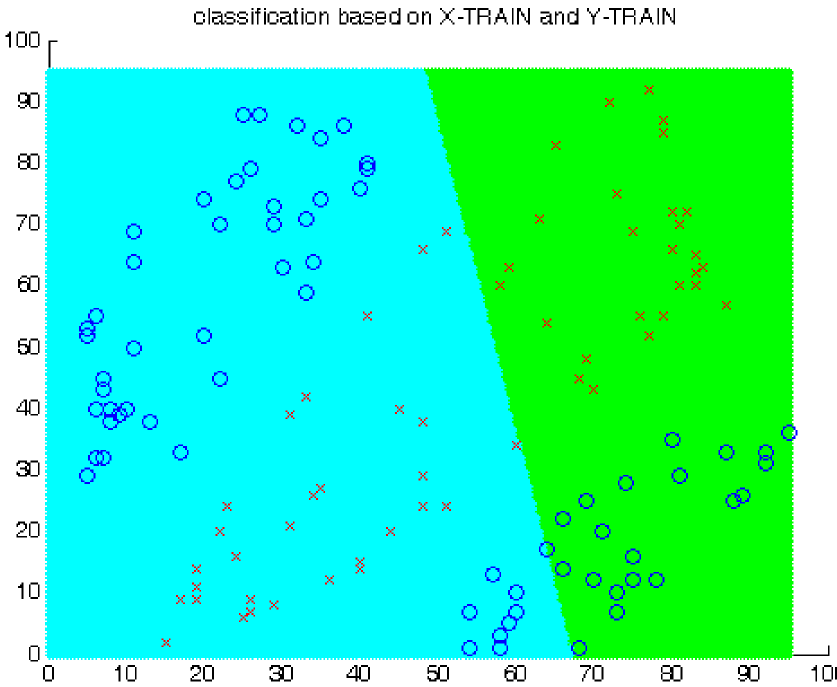
Linear cut

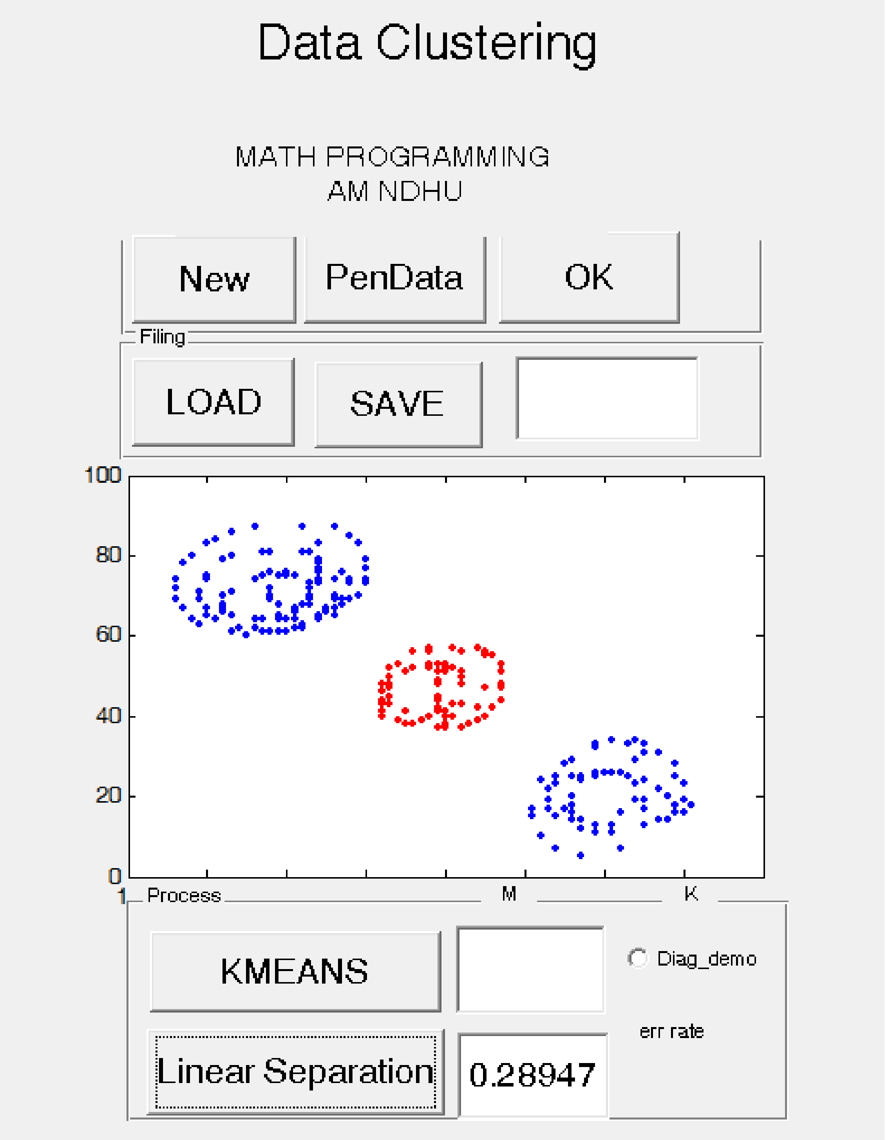
$$\hat{y} = f(x) = a_1 x_1 + a_2 x_2 + b$$



Linear cut fails

- load data_3c.txt
- error rate : 0.289





Computations

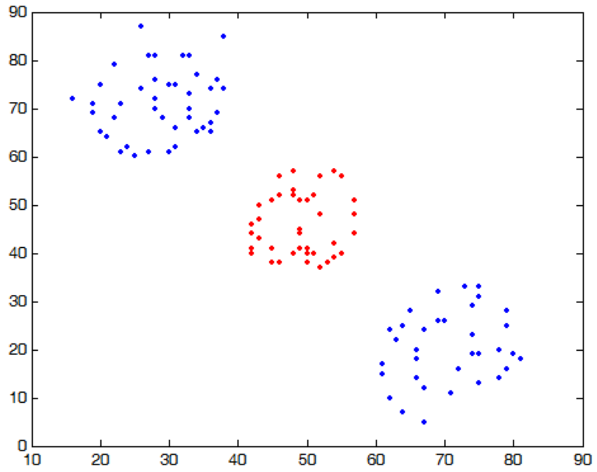
- STEP A: kmeans
  - Apply K-means to find K centers
- STEP B: cross distances
  - Calculate cross distances between K means and N data points
- STEP C: optimal coefficients
  - Determine a mapping from position to color

Sampling of training data

data_3c.zip

```
load data_3c
X=data_3c(:,1:2);
Y=data_3c(:,3);
```

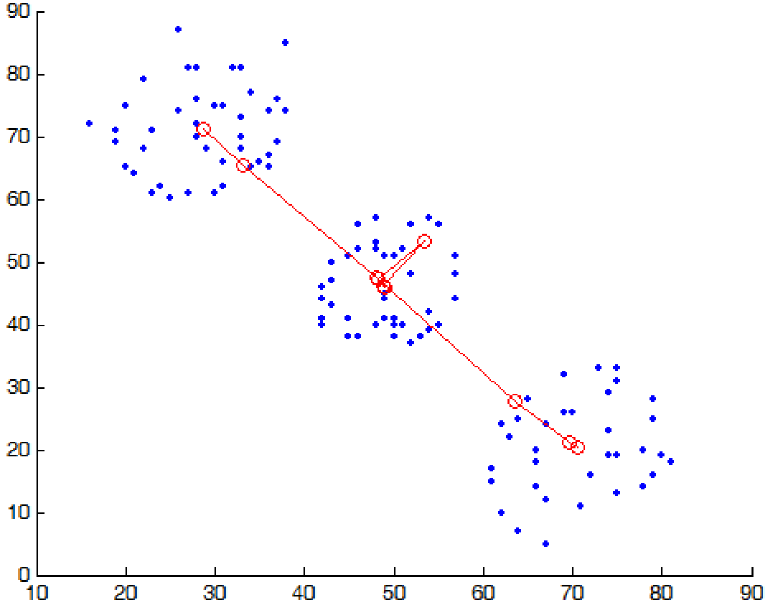
```
N=size(X,1);
ind=randperm(N);
n=floor(N/2);
x_train=X(ind(1:n),:);
y_train=Y(ind(1:n),:);
```

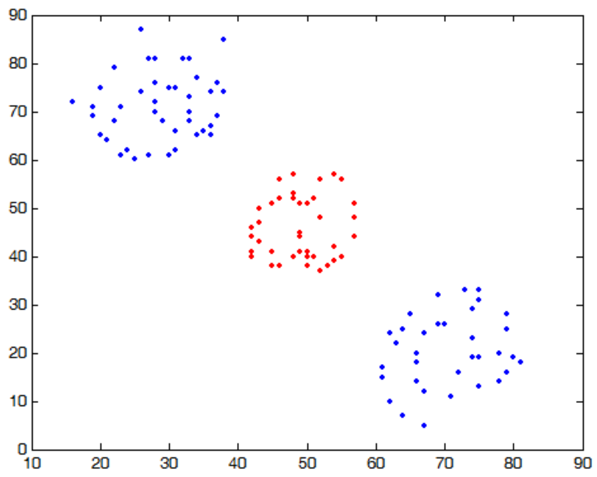



```
show_color_data(x_train, y_train)
```

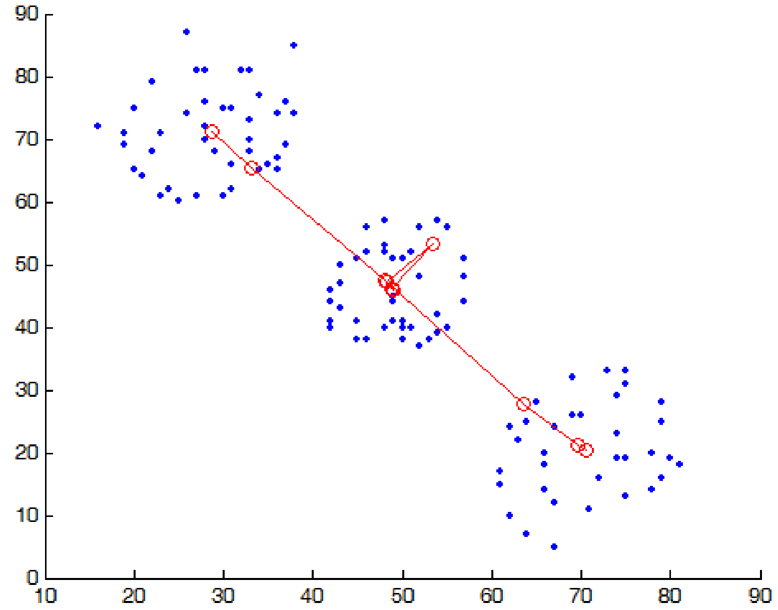
Step A

- K-means





centers = my_kmeans(x_train,3);



Step B

- Distances between centers and data points

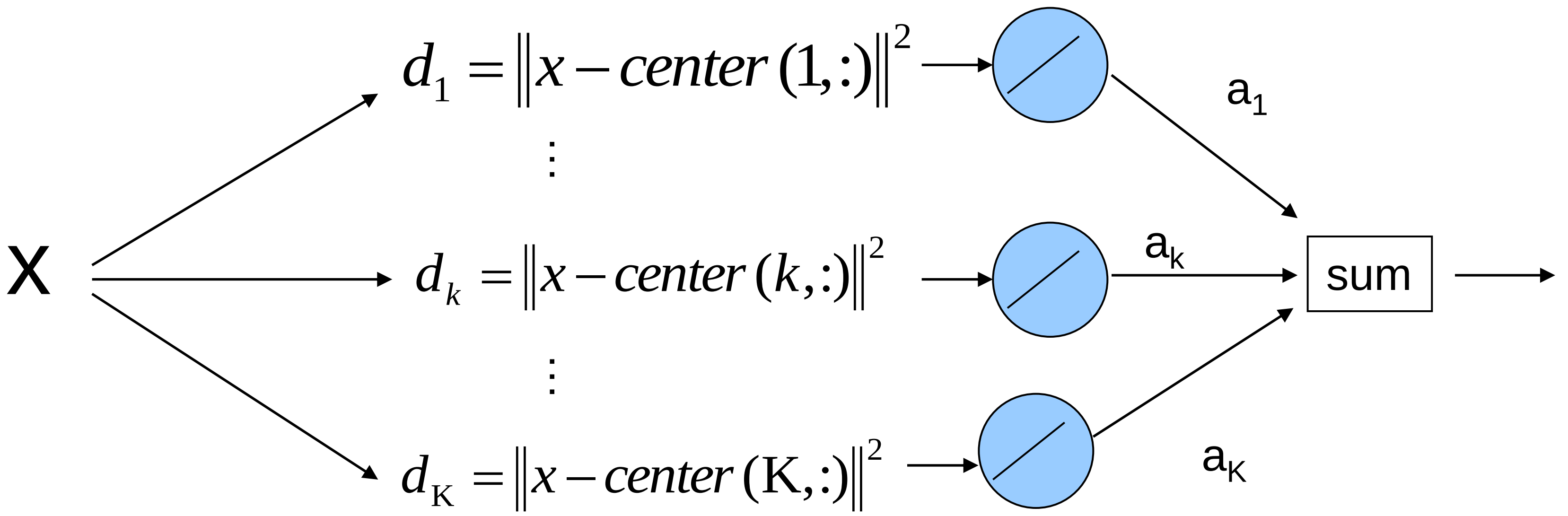
  cross_distance.m

  D = cross_distance(x_train,centers)

- D(i,j) stores the distance between the ith point and the jth center

Step C: optimal coefficients

- Linear combination of distances to centers



A mapping for coloring (mapping I)

$$\hat{y} = f(x)$$

$$= \sum_{k=1}^{K} a_k \left\| x - center(k,:) \right\|^2$$

$$= \sum_{k=1}^{K} a_k d_k$$

Linear combination of distances to K centers

Coloring one point

$$\text{Substitute the ith data point}$$

$$\hat{y}(i) = f(x(i,:))$$

$$= \sum_{k=1}^{K} a_k \left\| x(i,:) - center(k,:) \right\|^2$$

$$= \sum_{k=1}^{K} a_k d_{ik}$$

⇨

$$\hat{y}(i) = D(i,:) * \mathbf{a}$$

$\hat{\mathrm{y}}$: generated colors

D : distance matrix

$\mathbf{a}$: coefficients

$\mathbf{y}$: true colors

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_k \\ \vdots \\ \hat{y}_n \end{bmatrix}, D = [d_{ik}], \mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_K \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_n \end{bmatrix}$$

STEP D: Coloring n points

Substitute the ith data position

$$\hat{y}(i) = f(x(i,:))$$

$$= \sum_{k=1}^{K} a_k \left\| x(i,:) - center(k,:) \right\|^2 \iff \hat{\mathbf{y}} = D\mathbf{a}$$

$$= \sum_{k=1}^{K} a_k d_{ik}$$

STEP C: Optimal coefficients

$\hat{\mathbf{y}}$ is expected identical to $\mathbf{y}$

$$\hat{\mathbf{y}} \approx \mathbf{y}$$

$$\hat{\mathbf{y}} = D\mathbf{a} \approx \mathbf{y}$$

⇓

$$\mathbf{a} = pinv(D) * \mathbf{y}$$

Flow chart I : Optimal coefficients

x_train, y_train　　funtion [centers,a]=OC(x_train,y_train)

↓

STEP A　centers = my_kmeans(x_train,3);

↓

STEP B　D = cross_distance(x_train,centers)

↓

STEP C　a = pinv(D)*y_train;

↓

centers, a

Flow chart II : coloring

function y_hat=coloring(x_train,centers,a)

x_train, centers, a

↓

STEP B

D = cross_distance(x_train,centers)

↓

STEP D
coloring

y_hat=D*a;

↓

y_hat

Flow chart III: Error rate for training

function er=err_rate(y_hat,y_train)

y_hat ,y_train

↓

```
e=abs(y_train-round(y_hat));
```

↓

```
n= length(y_train);
er=sum(e)/n
```

↓

Error rate
ans =

0

Flow chart IV : TRAINING

x_train,y_train

↓

FLOW CHART I — [centers,a]=OC(x_train,y_train)

↓

FLOW CHART II — y_hat=coloring(x_train,centers,a)

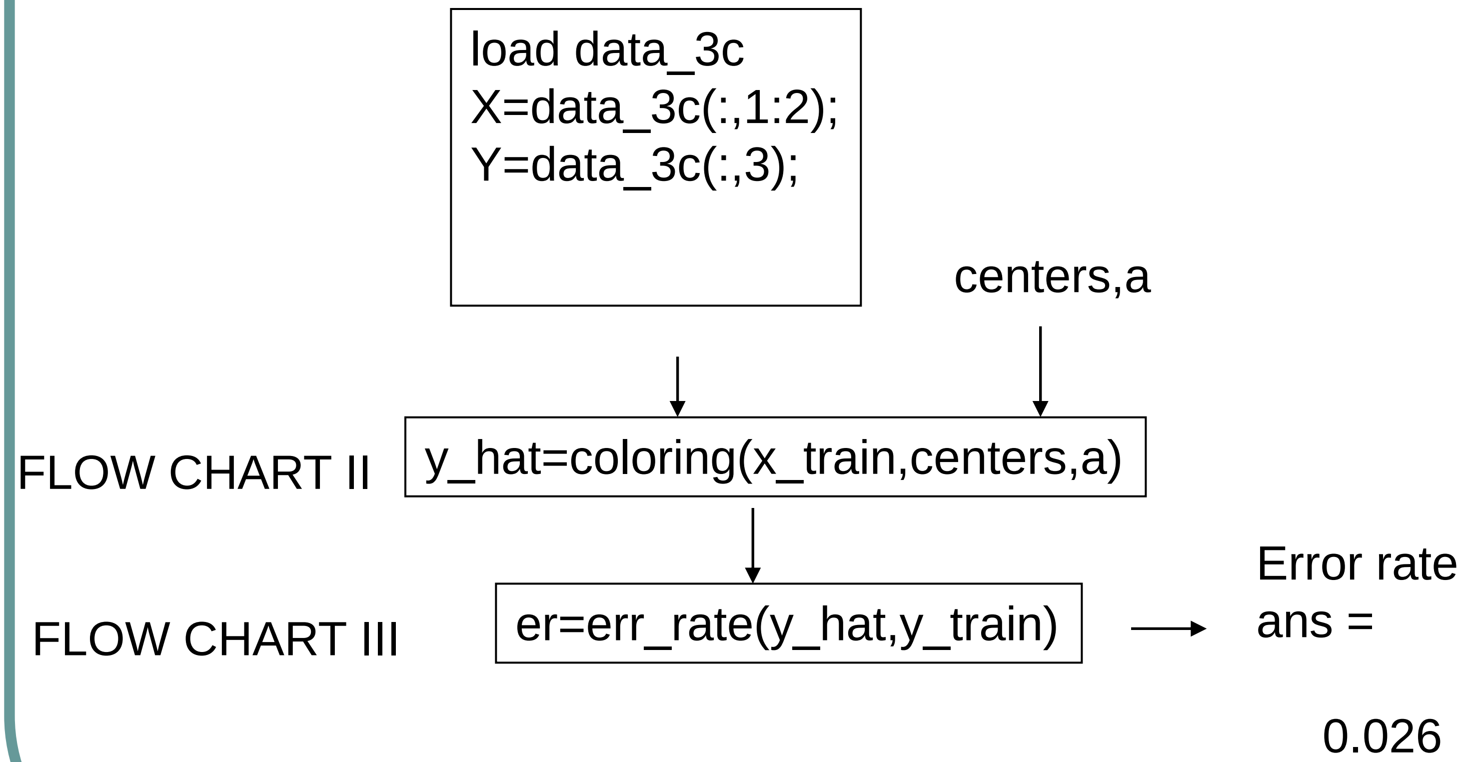
↓

FLOW CHART III — er=err_rate(y_hat,y_train)

↓

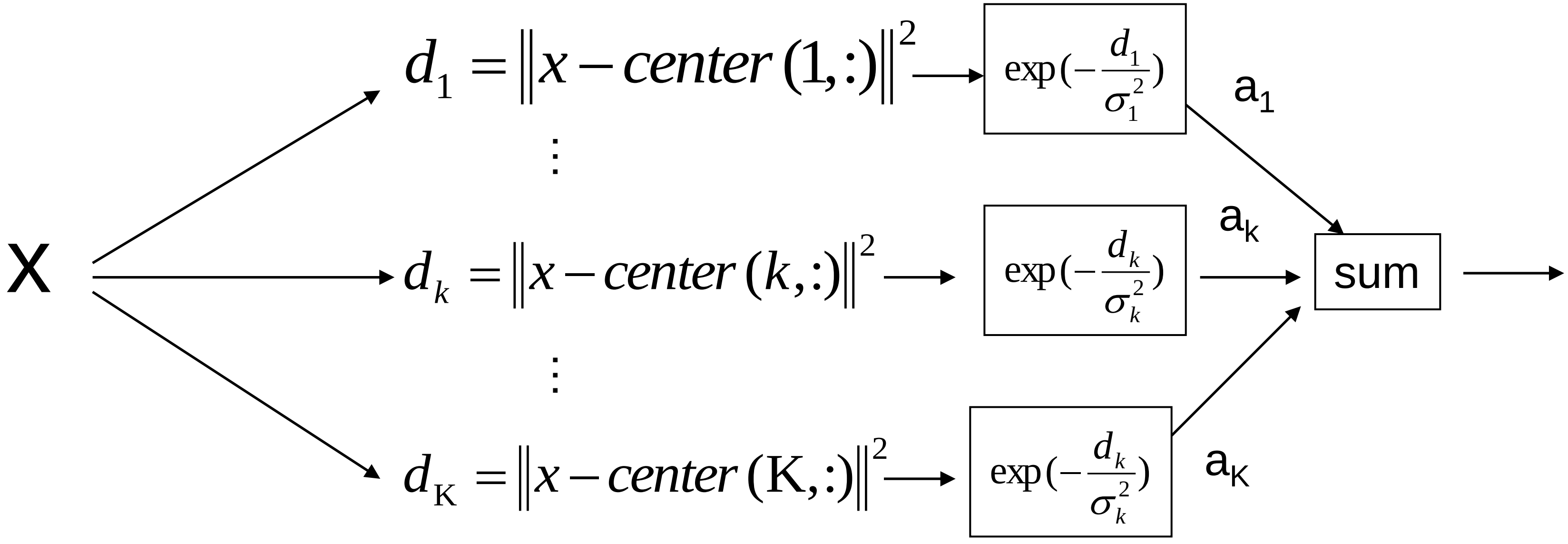
er

Flow Chart V: TESTING



Mapping II from position to color

Linear combination of exponential distances to centers



Discriminating rule

$$Substitute\ \text{the ith data point}$$

$$y(i) = f(x(i,:))$$

$$= \sum_{k=1}^{K} a_k \exp\left(-\frac{\left\| x(i,:) - center(k,:) \right\|^2}{\sigma_k^2}\right)$$

$$= \sum_{k=1}^{K} a_k \exp\left(-\frac{d_{ik}}{\sigma_k^2}\right)$$

$$D = [d_{ik}], \mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_K \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_K \end{bmatrix}$$

Step D: Coloring

Substitute the ith data point

$$\hat{y}(i) = f(x(i,:))$$

$$= \sum_{k=1}^{K} a_k \exp\left(-\frac{\left\| x(i,:) - center(k,:) \right\|^2}{\sigma_k^2}\right)$$

$$= \sum_{k=1}^{K} a_k \exp\left(-\frac{d_{ik}}{\sigma_k^2}\right)$$

$$\Rightarrow \quad \hat{\mathbf{y}} = \exp(-D/h)\mathbf{a}$$

Set all variances to h for simplicity

Optimal coefficients

$\hat{\mathbf{y}}$ is expected identical to $\mathbf{y}$

$$\hat{\mathbf{y}} \approx \mathbf{y}$$

$$\hat{\mathbf{y}} = \exp(-D/h)\mathbf{a} \approx \mathbf{y}$$

$$\downarrow$$

$$\mathbf{a} = pinv(\exp(-D/h)) * \mathbf{y}$$

Flow chart I : Optimal coefficients

funtion [centers,a]=OC(x_train,y_train)

x_train, y_train

↓

STEP A

```
centers = my_kmeans(x_train,3);
```

↓

STEP B

```
D = cross_distance(x_train,centers)
```

↓

STEP C

```
h=200;
a = pinv(exp(-D/h))*Y_TRAIN;
```

↓

centers, a

Flow chart II : coloring

function y_hat=coloring(x_train,centers,a)

x_train, centers, a

↓

STEP B

D = cross_distance(x_train,centers)

↓

STEP D
coloring

y_hat=exp(-D/h)*a;

↓

Y_hat

Flow chart III: Error rate for training

function er=err_rate(y_hat,y_train)

y_hat ,y_train

↓

```
e=abs(y_train-round(y_hat));
```

↓

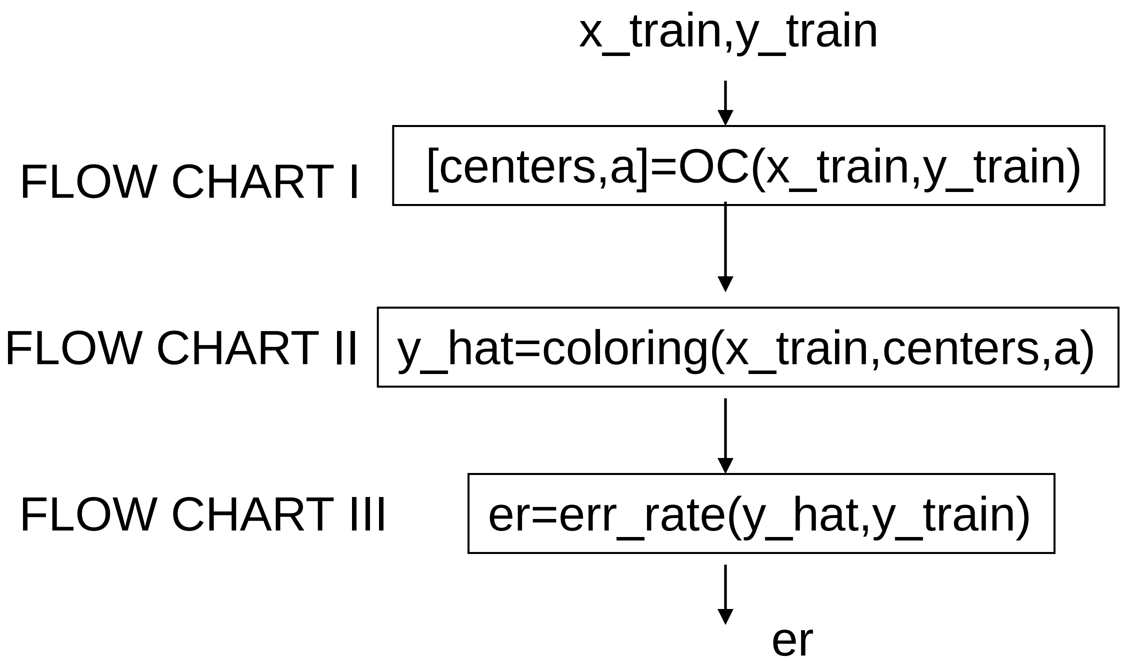
```
n= length(y_train);
er=sum(e)/n
```

↓

Error rate
ans =

0

Flow chart IV: TRAINING



Flow Chart V: TESTING

```
load data_3c
X=data_3c(:,1:2);
Y=data_3c(:,3);
```

centers,a

FLOW CHART II

```
y_hat=coloring(x_train,centers,a)
```

FLOW CHART III

```
er=err_rate(y_hat,y_train)
```

Error rate

ans =

0.000