Large Scale data clustering Models and codes

2021

J.M. Wu

Large scaled data clustering

- Large scaled data
- Parallel and distributed processes
- Expectation Maximization
- K-means
- Hierarchical clustering models
- Codes : annealed K-Means, Annealed EM
- Numerical simulations

Images and Sounds

- Facial images
  - http://www.face-rec.org/databases/
- Hand-writing character images
- MFCC features of speeches
  - https://sounds.bl.uk/
-

- Natural images
  - https://www.istockphoto.com/
  - http://deeplearning.net/datasets/
- Medical images
- Art Images https://www.pexels.com/search/art/

Gaussian pdf

Membership to a cluster
Exclusive or overlapping

Mahalanobis Distance

MEAN

$$P_k(x) = P(x|y_k, A_k)$$

$$= \frac{1}{(2\pi)^{d/2}\sqrt{|A_k^{-1}|}} \exp\left(\frac{-(x - y_k)^t A_k (x - y_k)}{2}\right)$$

WHY CLUSTERING

- Generative models Gaussian mixtures

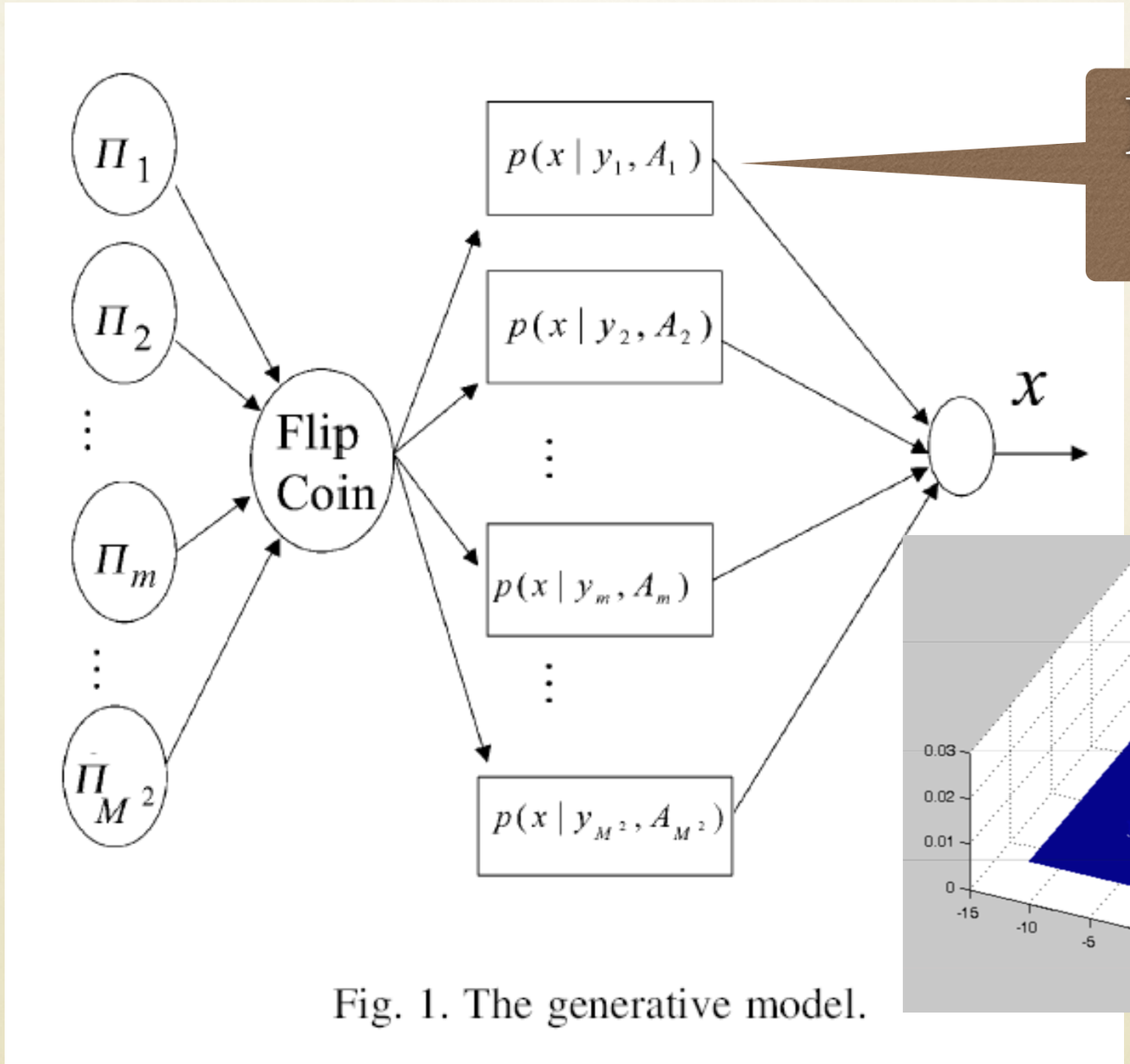
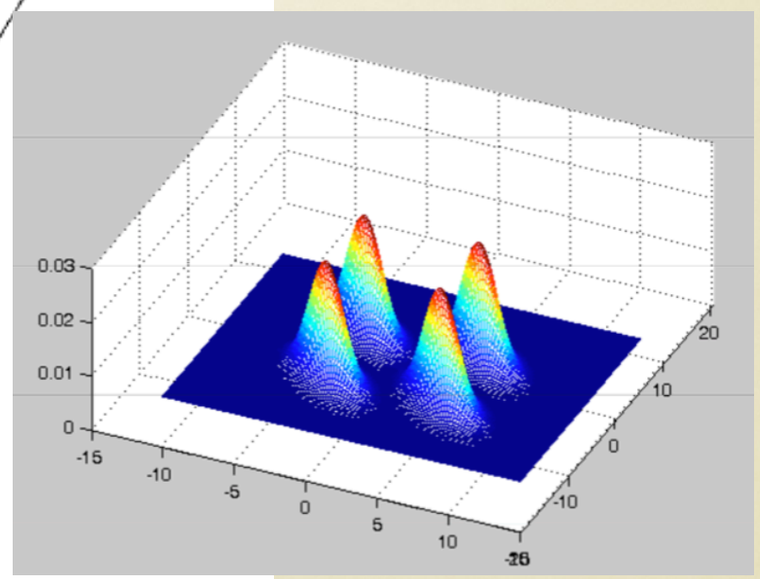
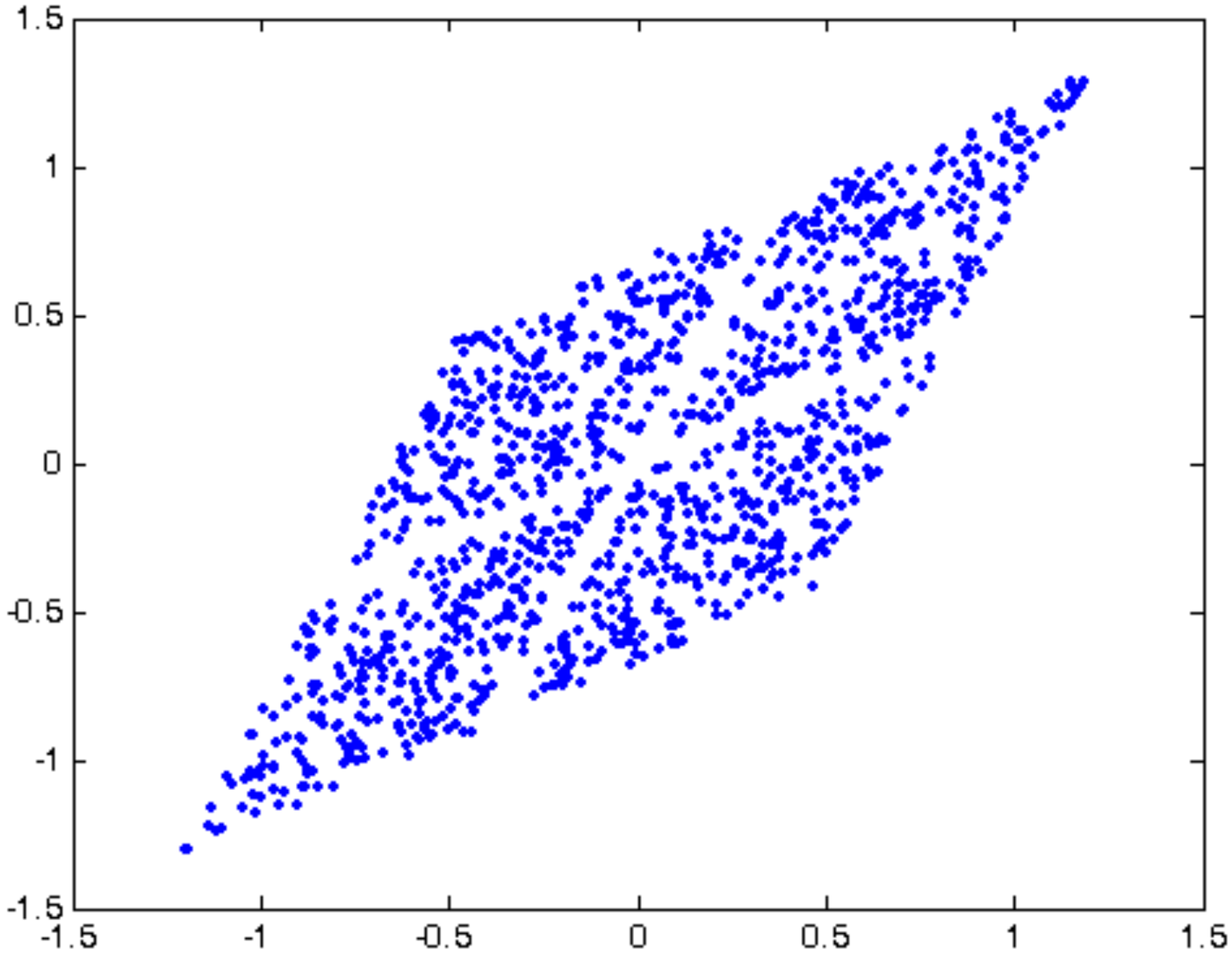


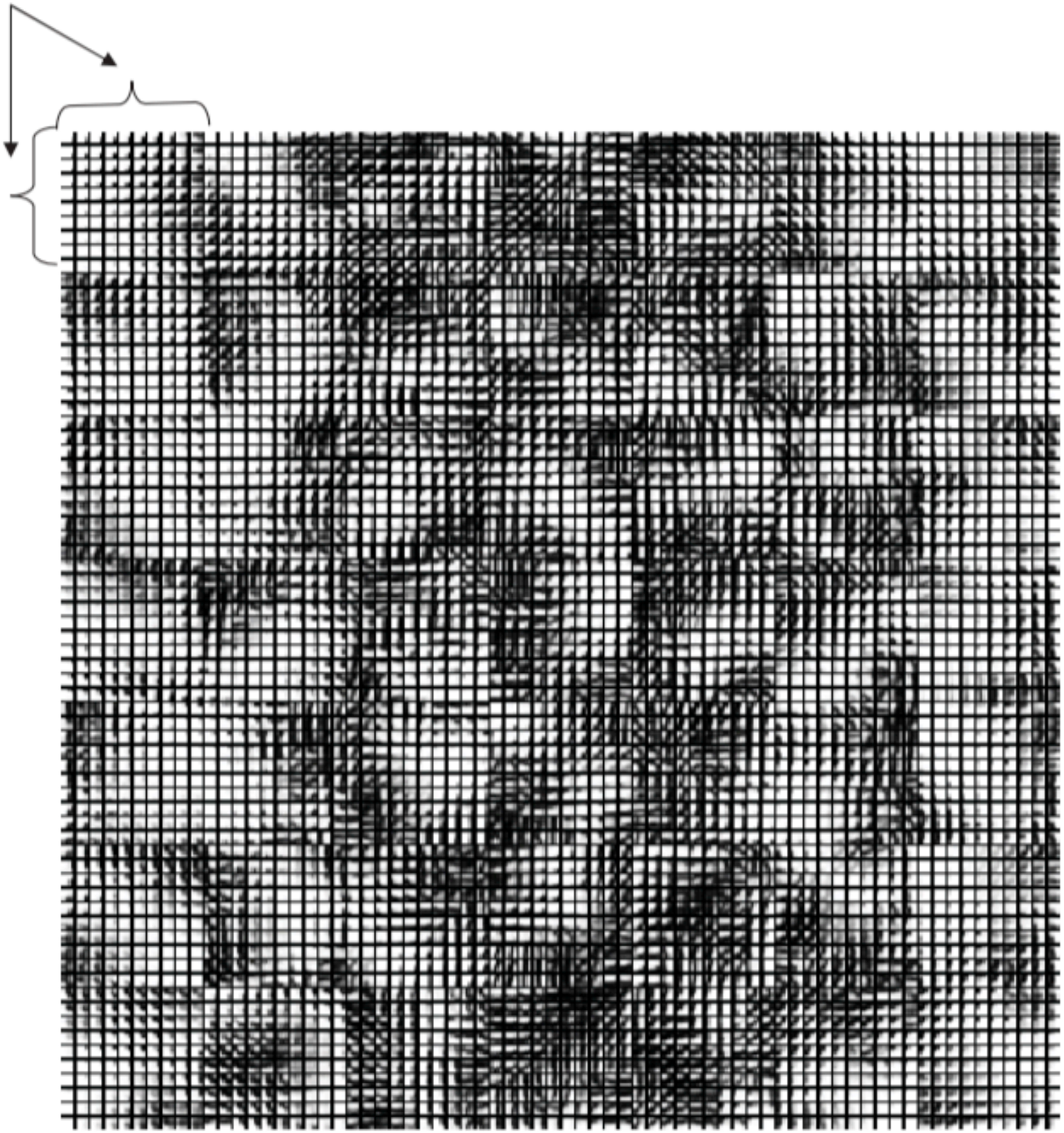
Fig. 1. The generative model.

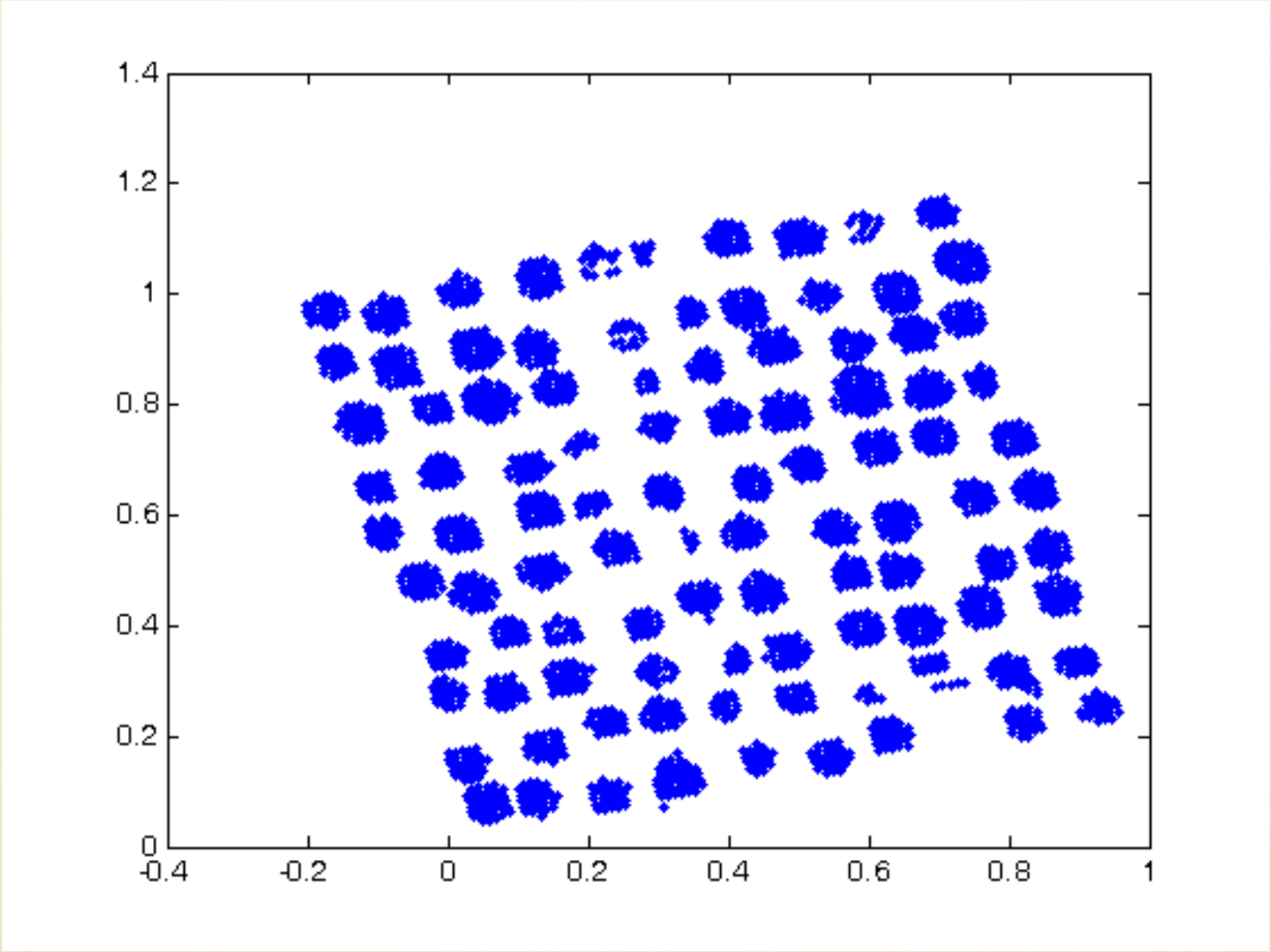
Many local means



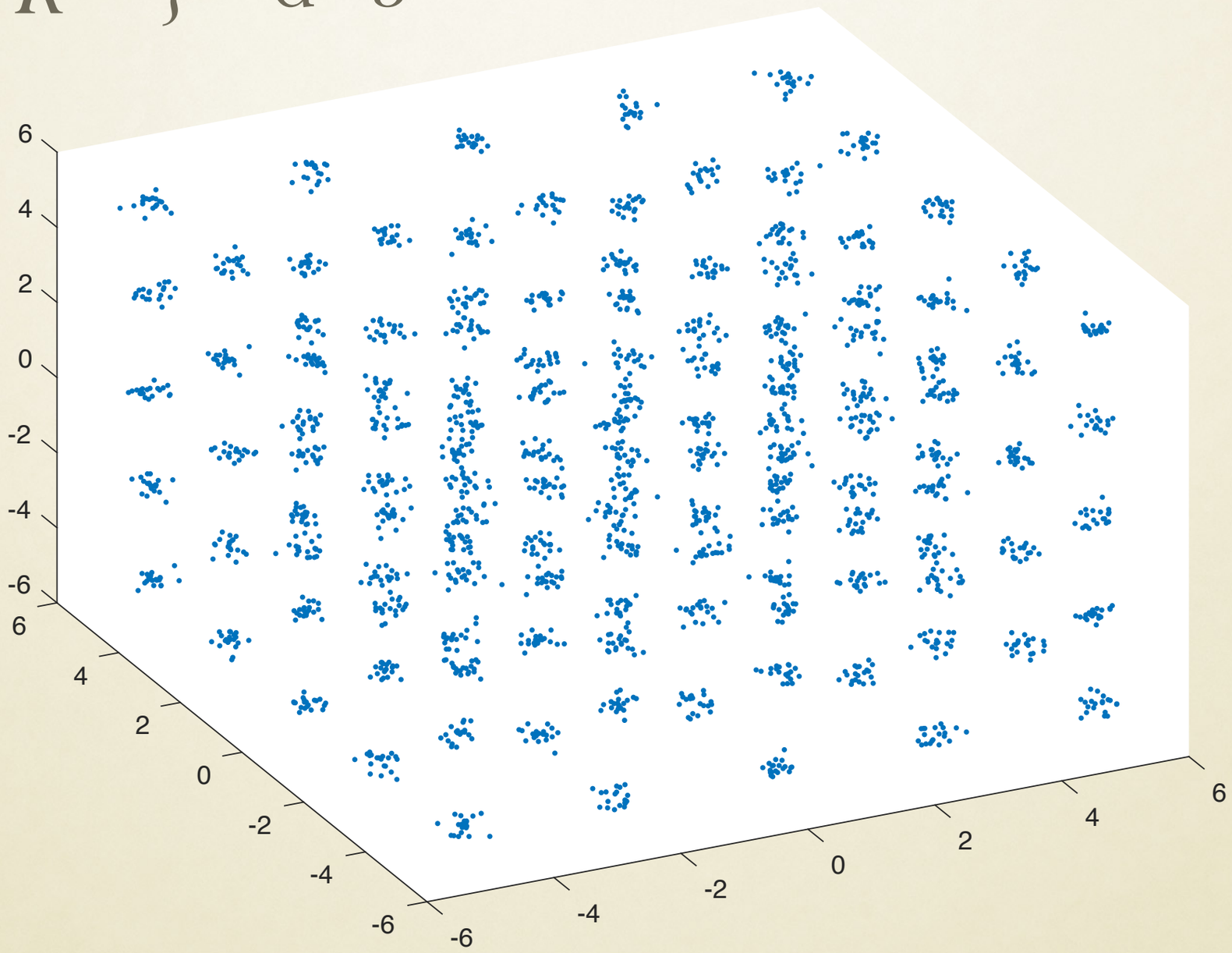

1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

A block for 10x10 cortical points of a natural elastic net




1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

$$X=\{x[t]\in R^{d}\ \}\quad d=3$$

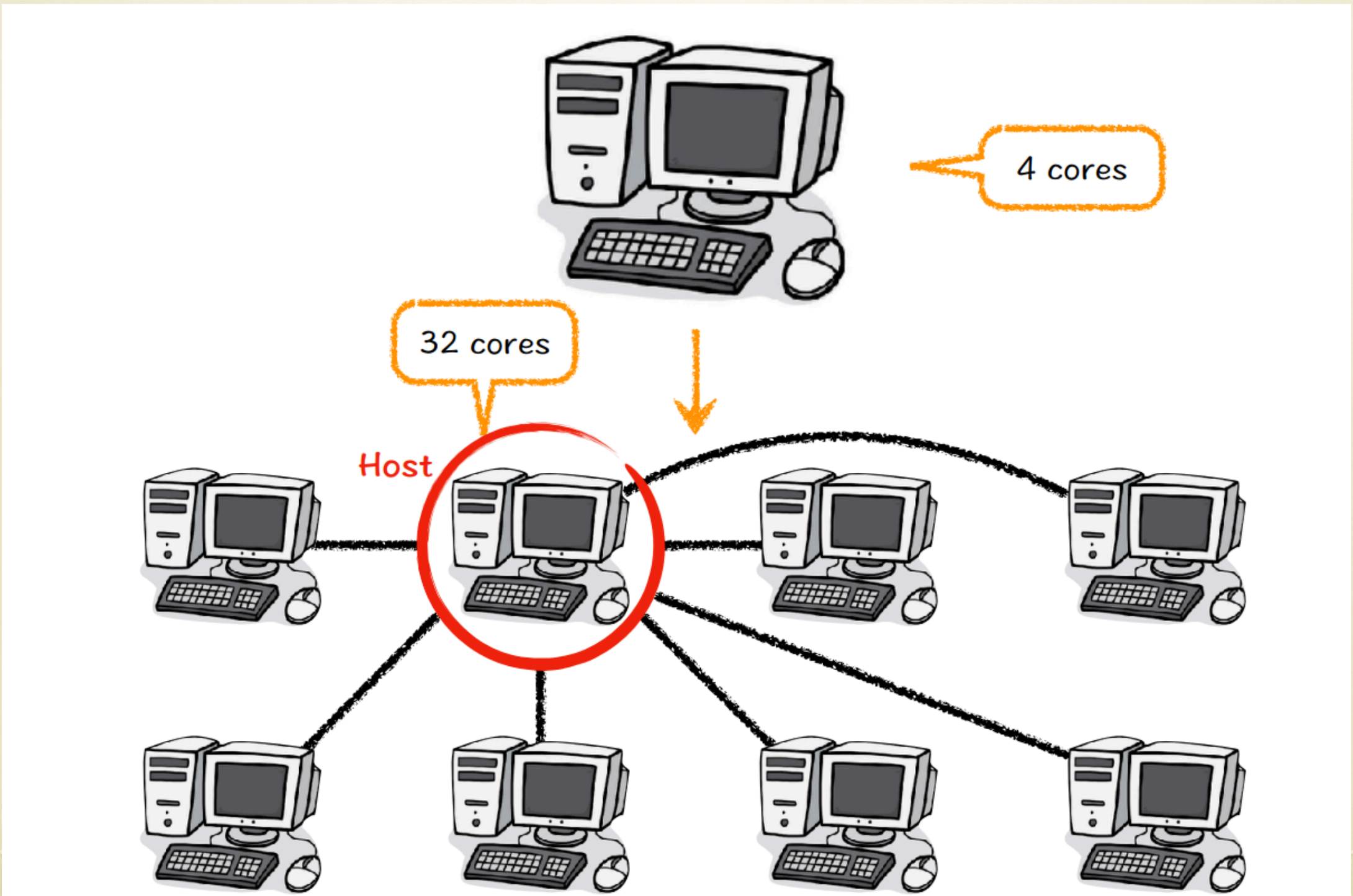


data_gen.m

```
clear all
L=5;
a(1,:)=linspace(-5,5,L);
a(2,:)=linspace(-5,5,L);
a(3,:)=linspace(-5,5,L);
X=[];
for i=1:L
    for j=1:L
        for k=1:L
            center=[a(1,i) a(2,j) a(3,k)];
            Xi=randn(20,3)*0.15+ ones(20,1)*center;
            X=[X;Xi];


        end
    end
end
plot3(X(:,1),X(:,2),X(:,3),'.');
```

Parallel and Distributed Processes



- 在其他台電腦輸入當台的 IP 與 作為 Host 的電腦 IP

Admin Center

File Hosts MJS Workers Help

Hosts

Add or Find...
Start mdce Service
Stop mdce Service...
Test Connectivity...

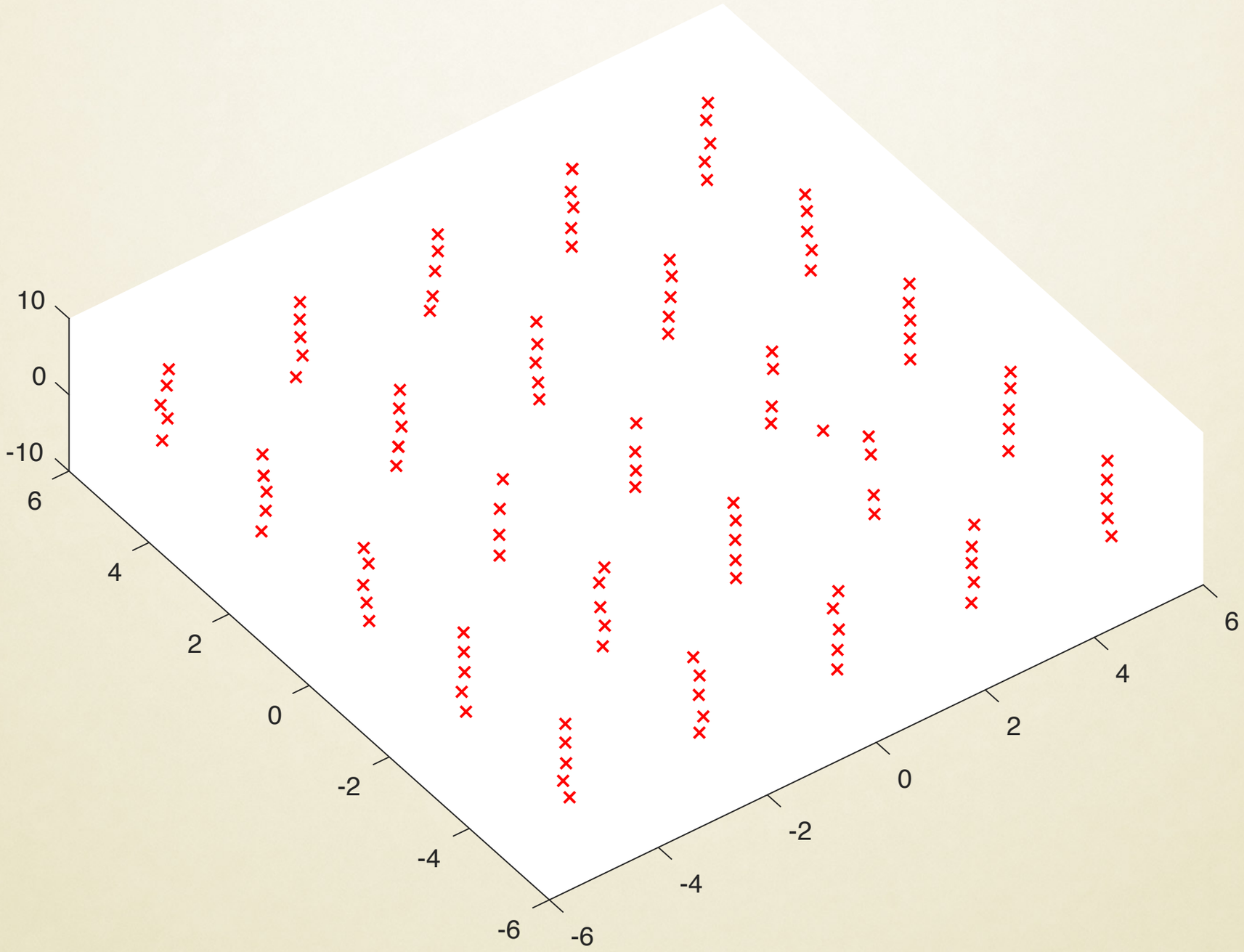
Host			MDCE Service		MJS	Work...
Hostname	Reachable	Cores	Status	Up Since	Name	Count
am5-1. (192.168.1.214)	yes	4	running	2018-07-06 16:53		4
am5-2. (192.168.1.134)	yes	4	running	2018-07-06 16:53		4
am5-3. (192.168.1.211)	yes	4	running	2018-07-06 16:52		4
am5-4. (192.168.1.181)	yes	4	running	2018-07-06 16:51		4
am5-5. (192.168.1.150)	yes	4	running	2018-07-06 16:35	TSP4000	4
am5-6. (192.168.1.240)	yes	4	running	2018-07-06 16:33		4
am5-7. (192.168.1.217)	yes	4	running	2018-07-06 16:32		4
am5-8. (192.168.1.197)	yes	4	running	2018-07-06 16:31		4

- 8 台各有 4 核心的電腦 → 一台擁有 32 核心的電腦

MATLAB Job Scheduler (MJS)

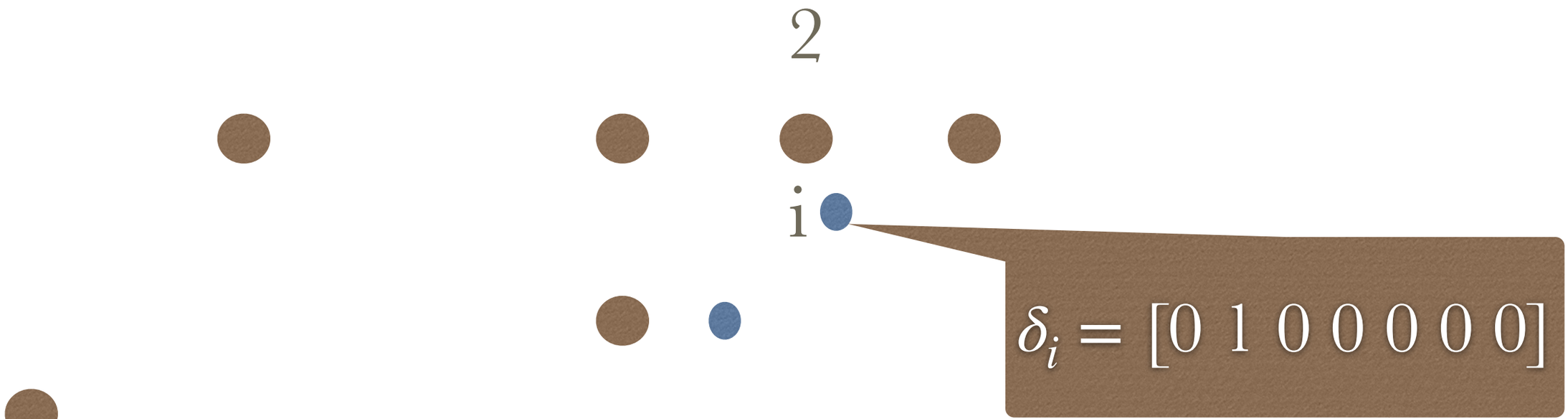
Start...
Stop...
Resume

Name	Hostname	Status	Up Since	Workers
TSP4000	am5-5.	running	2018-07-06 16:57	32


10
0
-10
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

Criteria

- High-dimensional data: sub-images
- Large-scaled data: ten millions patterns
- High Speed: parallel and distributed processes
- Accuracy, High Quality


2
i
$\delta_i = [0\ 1\ 0\ 0\ 0\ 0\ 0]$

Exclusive Membership

$\delta_i = [0\ 1\ 0\ 0\ 0\ 0\ 0] = e_2^7$

$$e_k^K = [0, 0, \cdots, 0, 1, 0, \cdots, 0, 0]^T$$

$$\text{pos} \quad 1\ 2 \cdots \text{k-1}, \text{k}, \text{k+1}, \cdots, \text{K}$$

- A vector of K binary values
- Only one active bit among K bits
- The kth bit is active and the remaining bits zeroes

Standard Basis

$\delta[t] \in \{e_1, \ldots, e_7\}$, $\delta[t] = e_2$ iff
x[t] is generated by the 2th pdf

$$\boldsymbol{\Xi} = \{\mathbf{e}_1^K, \cdots, \mathbf{e}_k^K, \cdots, \mathbf{e}_K^K\}$$

$$\boldsymbol{\delta}[t] \in \boldsymbol{\Xi} = \{e_1^K, \cdots, e_k^K, \cdots, e_K^K\}$$

$$\boldsymbol{\delta}[t] = e_k^K \Leftrightarrow \text{x[t] is generated by the kth pdf}$$

$$\boldsymbol{\delta}[t] \in \Xi = \{e_1^K, \cdots, e_k^K, \cdots, e_K^K\}$$

$$\boldsymbol{\delta}[t] = e_k^K \Leftrightarrow \text{x[t] is generated by the kth pdf}$$

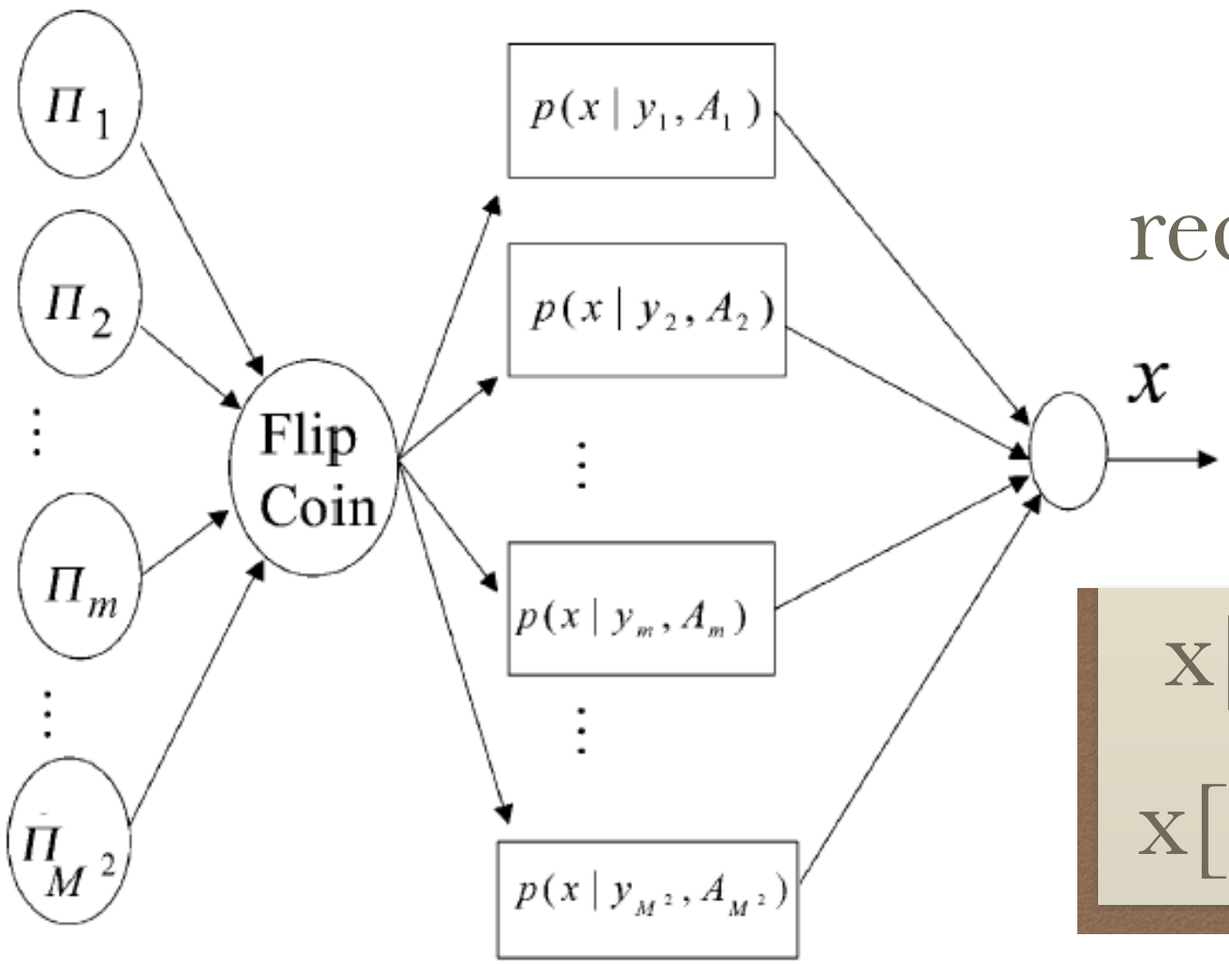


Fig. 1. The generative model.

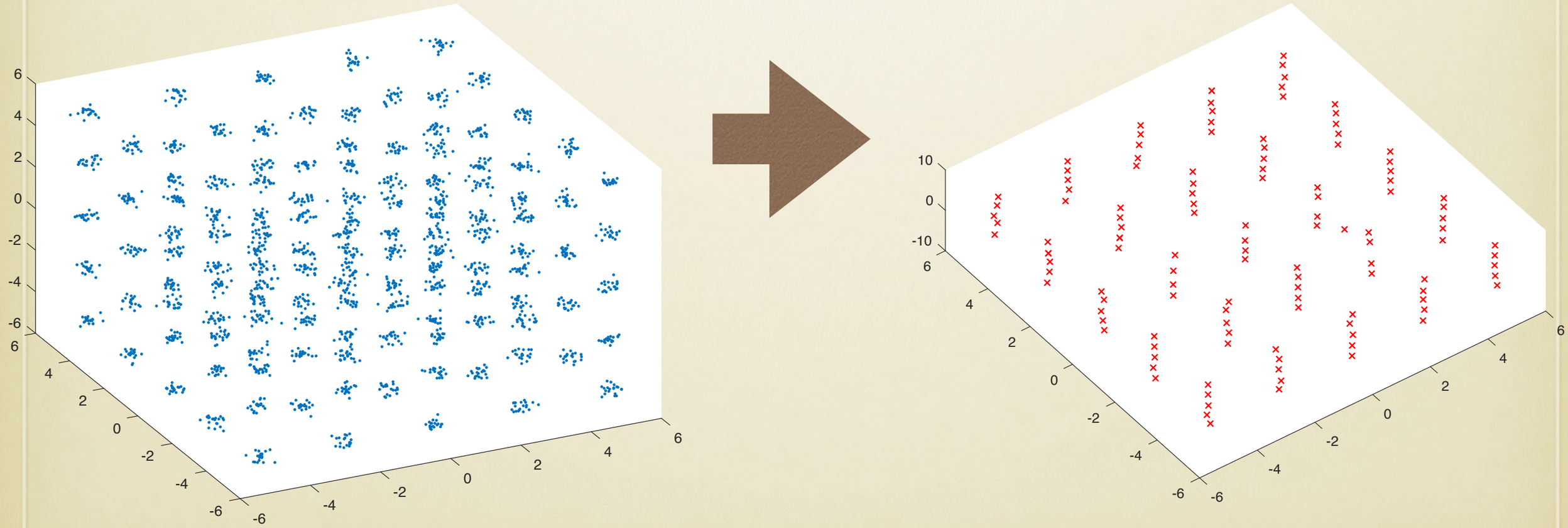
K-means:
recover exclusive memberships

x[1]

x[N]

Clustering

- input : x[t] for all t
- output : $\delta[t]$ for all t
- Representatives or local means : $\{y[m]\}_{m=1}^{K}$



$$\Pr(\xi_i = e_k^K) \propto \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)$$

$$\sum_{k=1}^{K} \Pr(\xi_i = e_k^K) = 1$$

$$\beta > 0$$

$$\Pr(\xi_i = e_k^K) = ?$$

Probabilistic Memberships

$$\Pr(\xi_i = e_k^K) \propto \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)$$

$$\sum_{k=1}^{K} \Pr(\xi_i = e_k^K) = 1$$

$$\Pr(\xi_i = e_k^K) = ?$$

$$\Pr(\xi_i = e_k^K) = C \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)$$

$$C \sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2) = 1$$

$$C = \frac{1}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}$$

$$\Pr(\xi_i = e_k^K) = \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}$$

Expectation

$$\Pr(\xi_i = e_k^K) \propto \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)$$

- Consider an exclusive membership as a random vector
- Assumption of probability
- Expectation $\langle \xi_i \rangle = ?$

Expectation

$$\left\langle \xi_i \right\rangle = \sum_{k=1}^{K} \Pr(\xi_i = \mathbf{e}_k^K)\mathbf{e}_k^K = \sum_{k=1}^{K} \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}{\sum_{h=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_h \|^2)} \mathbf{e}_k^K$$

$$= \left(\frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_1 \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}, \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_2 \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}, \cdots, \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_K \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)} \right)$$

Expectation Equation

$$\nu_{ik} \equiv \left\langle \xi_{ik} \right\rangle = \Pr(\xi_i = e_k^K) = \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)} \quad \text{(E1)}$$

EM: expectation maximization

Expectation Maximization

$$\mathrm{E}(\xi, Y) = \sum_{i=1}^{N} \sum_{k=1}^{K} \xi_{ik} (\mathbf{x}_i - \mathbf{y}_k)^T (\mathbf{x}_i - \mathbf{y}_k)$$

Mahalanobis
A=I

A: pdf

$$(x_i - y_k)^T A (x_i - y_k)$$

- Mathematical modeling
- The distance between x[t] and its representative is minimized.

One and only one active bit in $[\xi_{i1}, \ldots, \xi_{ik}, \ldots, \xi_{iK}]$
inner summation contains one non-zero term at most

$$\mathrm{E}(\boldsymbol{\xi}, Y) = \sum_{i=1}^{N}\sum_{k=1}^{K} \xi_{ik}(\mathbf{x}_i - \mathbf{y}_k)^T(\mathbf{x}_i - \mathbf{y}_k)$$

- EM minimizes $E(\langle\xi\rangle, Y)$ directly with respect to all y_k

$$\boxed{\mathrm{E}(\langle\boldsymbol{\xi}\rangle, Y) = \sum_{i=1}^{N}\sum_{k=1}^{K} \langle\xi_{ik}\rangle(\mathbf{x}_i - \mathbf{y}_k)^T(\mathbf{x}_i - \mathbf{y}_k)}$$

Maximization (minimization)

$$\mathrm{E}(\langle \xi \rangle, Y) = \sum_{i=1}^{N} \sum_{k=1}^{K} \langle \xi_{ik} \rangle (\mathbf{x}_i - \mathbf{y}_k)^T (\mathbf{x}_i - \mathbf{y}_k)$$

$$\frac{\partial \mathrm{E}(\langle \xi \rangle, Y)}{\partial \mathbf{y}_k} = -2 \sum_{i=1}^{N} \langle \xi_{ik} \rangle (\mathbf{x}_i - \mathbf{y}_k) = 0$$

$$\sum_{i=1}^{N} \langle \xi_{ik} \rangle \mathbf{y}_k = \sum_{i=1}^{N} \langle \xi_{ik} \rangle \mathbf{x}_i \Rightarrow \mathbf{y}_k = \frac{\sum_{i=1}^{N} \langle \xi_{ik} \rangle \mathbf{x}_i}{\sum_{i=1}^{N} \langle \xi_{ik} \rangle}$$

EM Formula

$$v_{ik} \equiv \left\langle \xi_{ik} \right\rangle = \Pr(\xi_i = e_k^K) = \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)} \quad \text{(E1)}$$

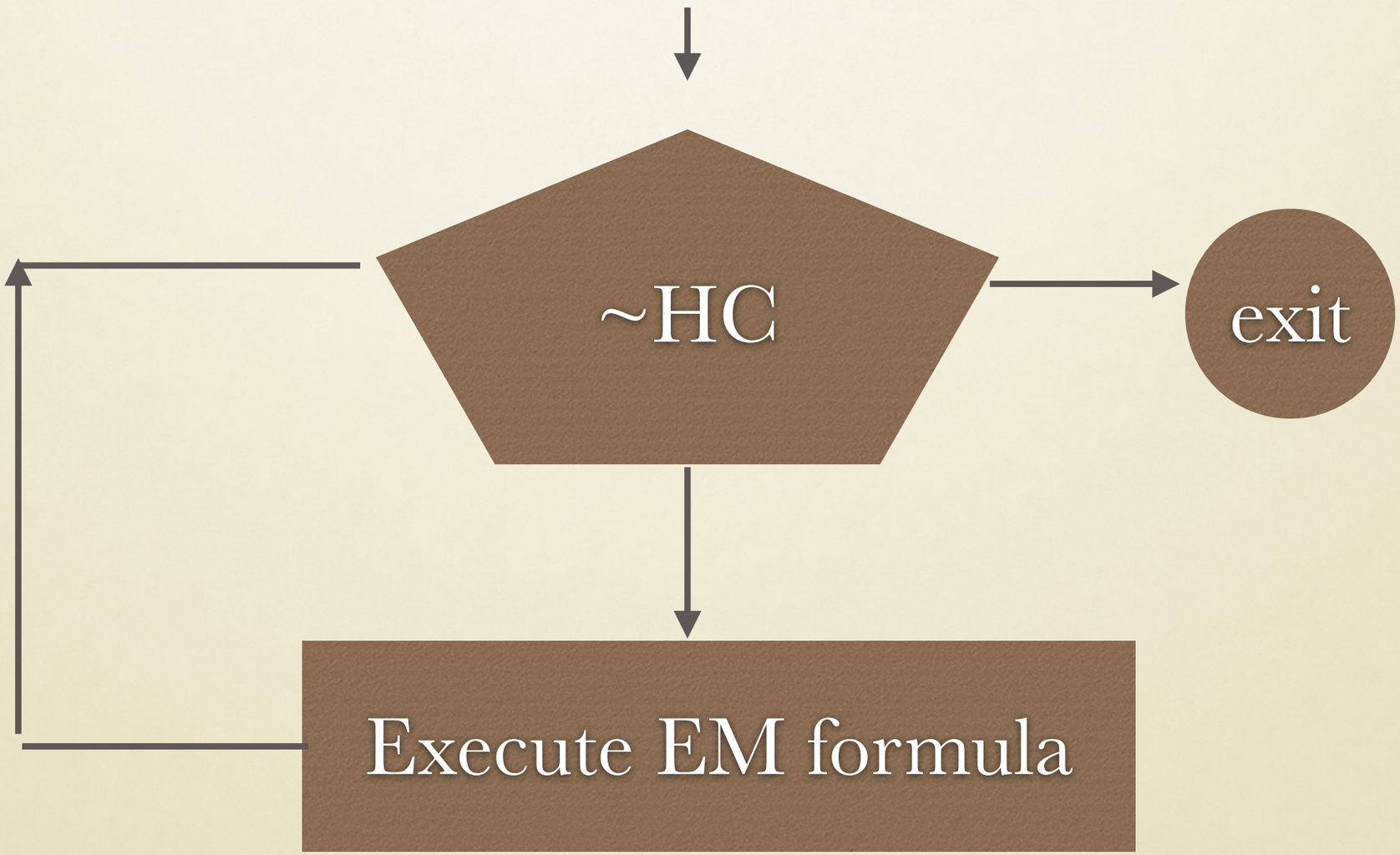
$$\mathbf{y}_k = \frac{\sum_{i=1}^{N} \left\langle \xi_{ik} \right\rangle \mathbf{x}_i}{\sum_{i=1}^{N} \left\langle \xi_{ik} \right\rangle}$$

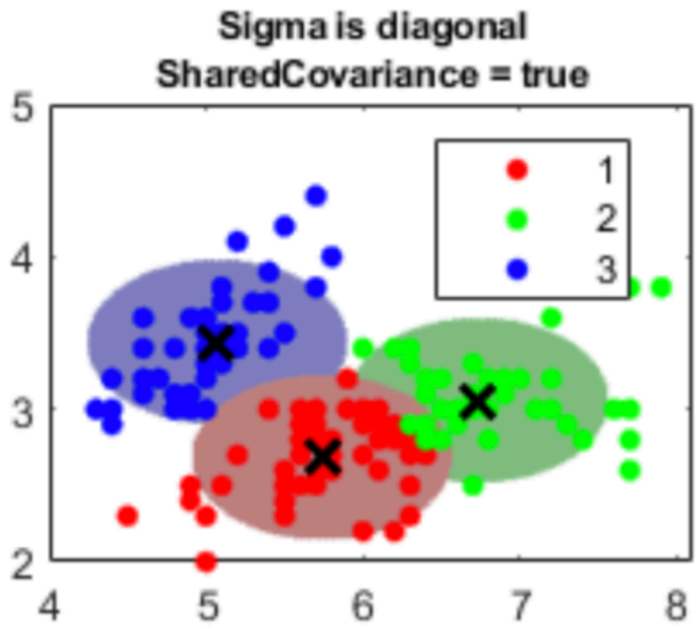
$\beta = 1$

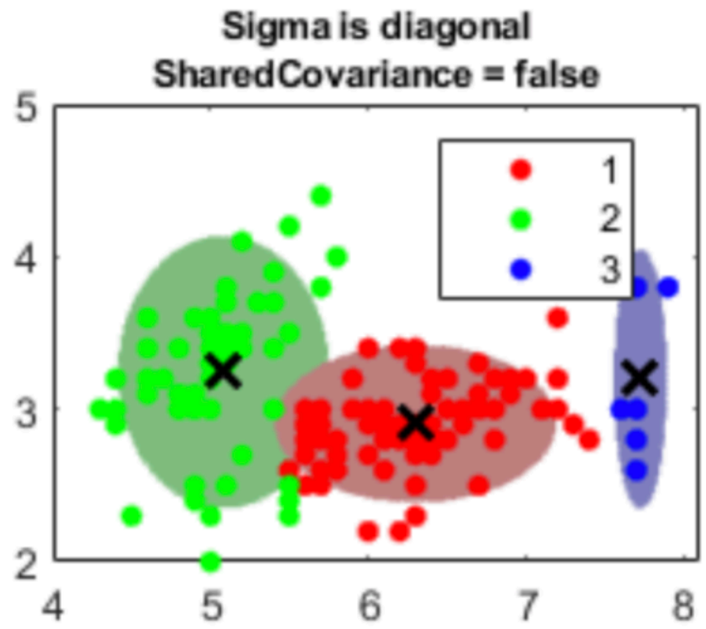
While-looping

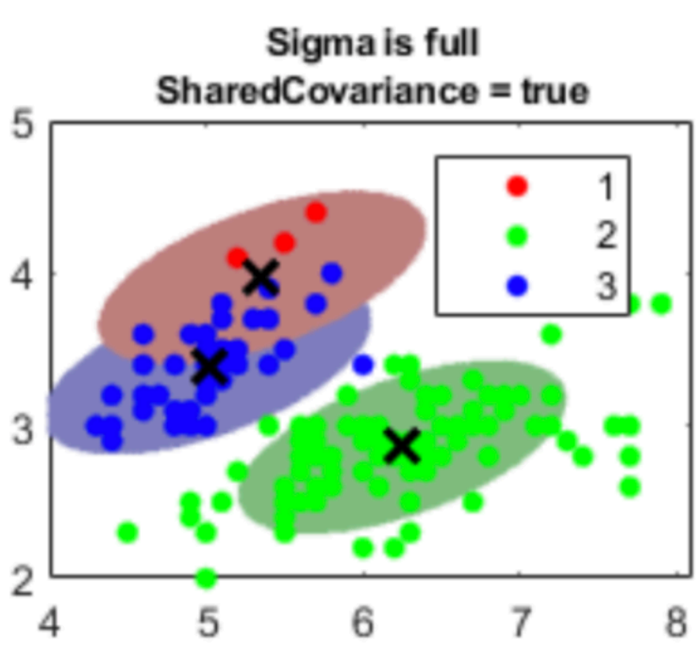
Initialize v_{ik} for all ik

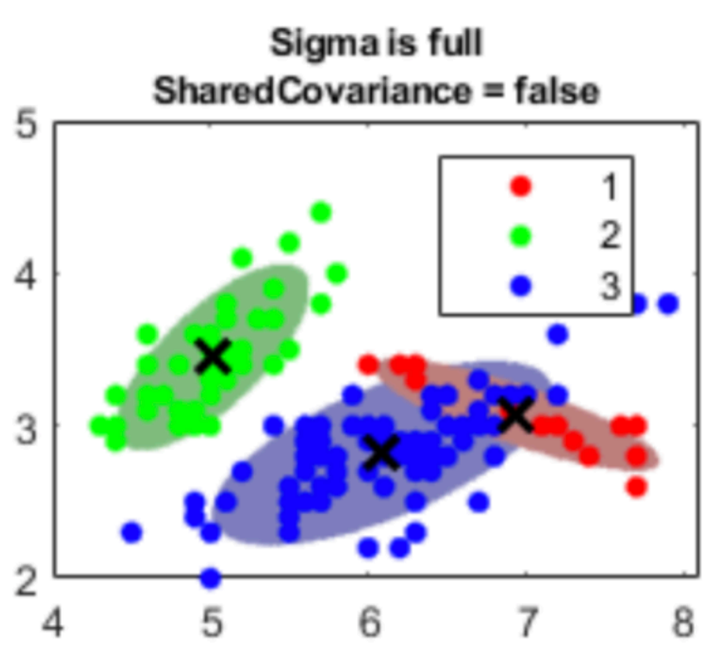
Initialize y_k for all k




Sigma is diagonal
SharedCovariance = true
1
2
3
5
4
3
2
4
5
6
7
8


Sigma is diagonal
SharedCovariance = false
1
2
3
5
4
3
2
4
5
6
7
8


Sigma is full
SharedCovariance = true
1
2
3
5
4
3
2
4
5
6
7
8


Sigma is full
SharedCovariance = false
1
2
3
5
4
3
2
4
5
6
7
8

girishvjoshi / **EM-Clustering** Public

Notificati

master 1 branch 0 tags

Go to file Code

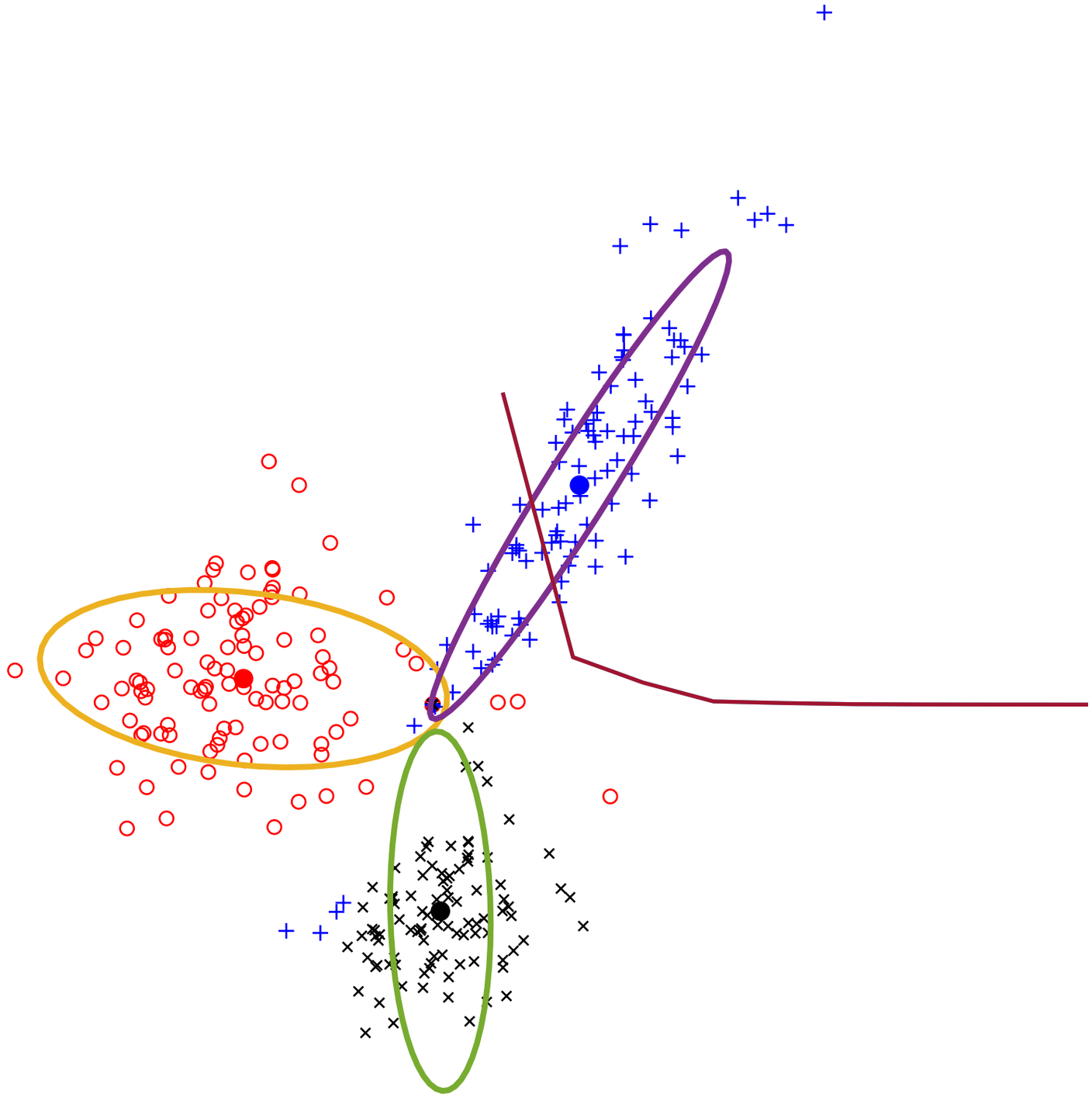
girishvjoshi moved the file to new folder 3ab5b8f on 5 Oct 2018 6 commits

@EM	moved the file to new folder	3 years ago
EM_main.m	Update EM_main.m	4 years ago
README.md	Update README.md	4 years ago

README.md

EM-Clustering

This Code Implements Expectation-Maximization Algorithm in Matlab



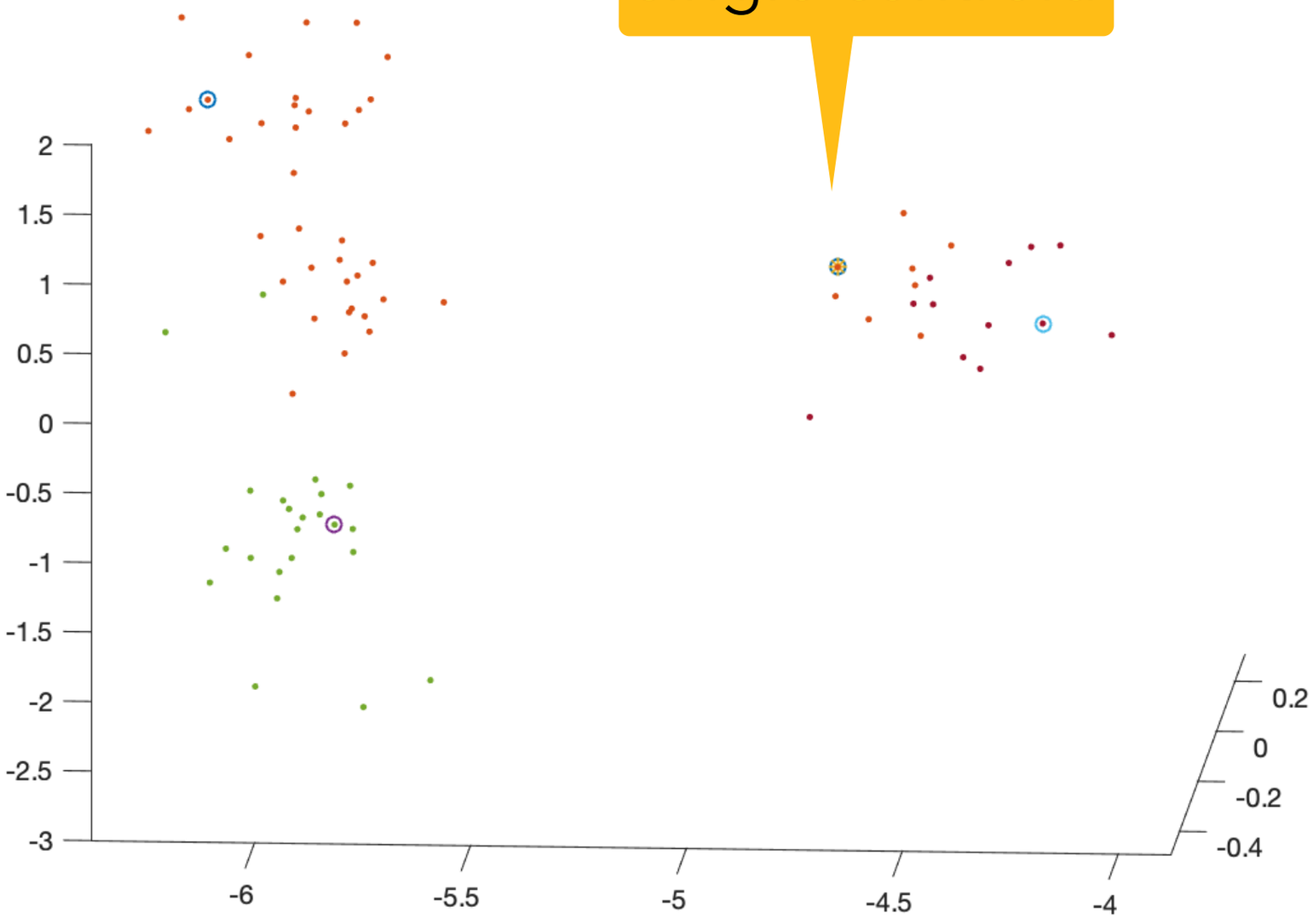
Searched for **expectation maximization**

Gaussian Mixture Models - Cluster based on Gaussian mixtu
Expectation-Maximization algorithm
Documentation > Statistics and Machine Learning Toolbox > Clus

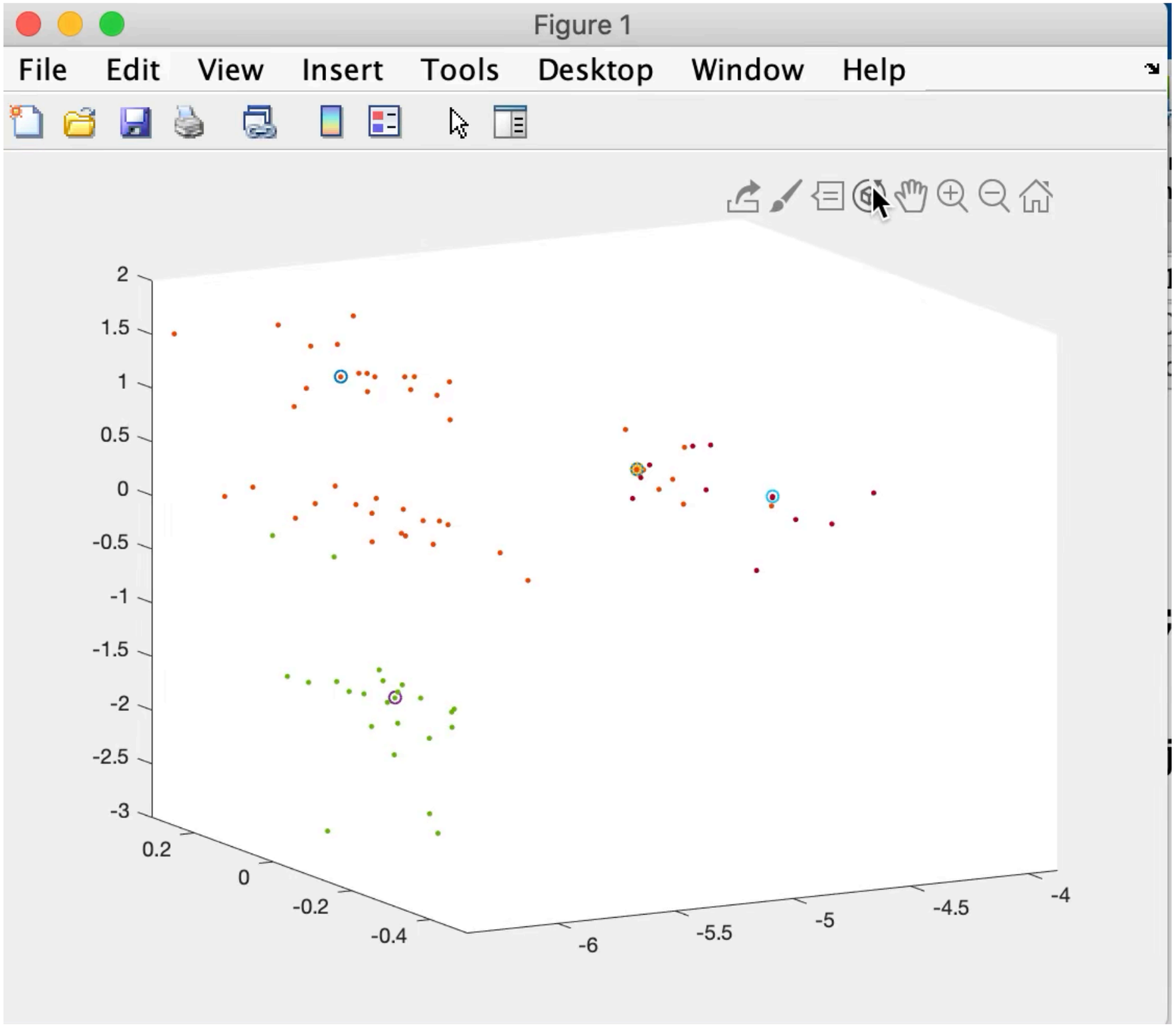
Cluster Analysis - Unsupervised learning techniques to find patterns in data
Documentation > Statistics and Machine Learning Toolbox

Clustering by K-Means

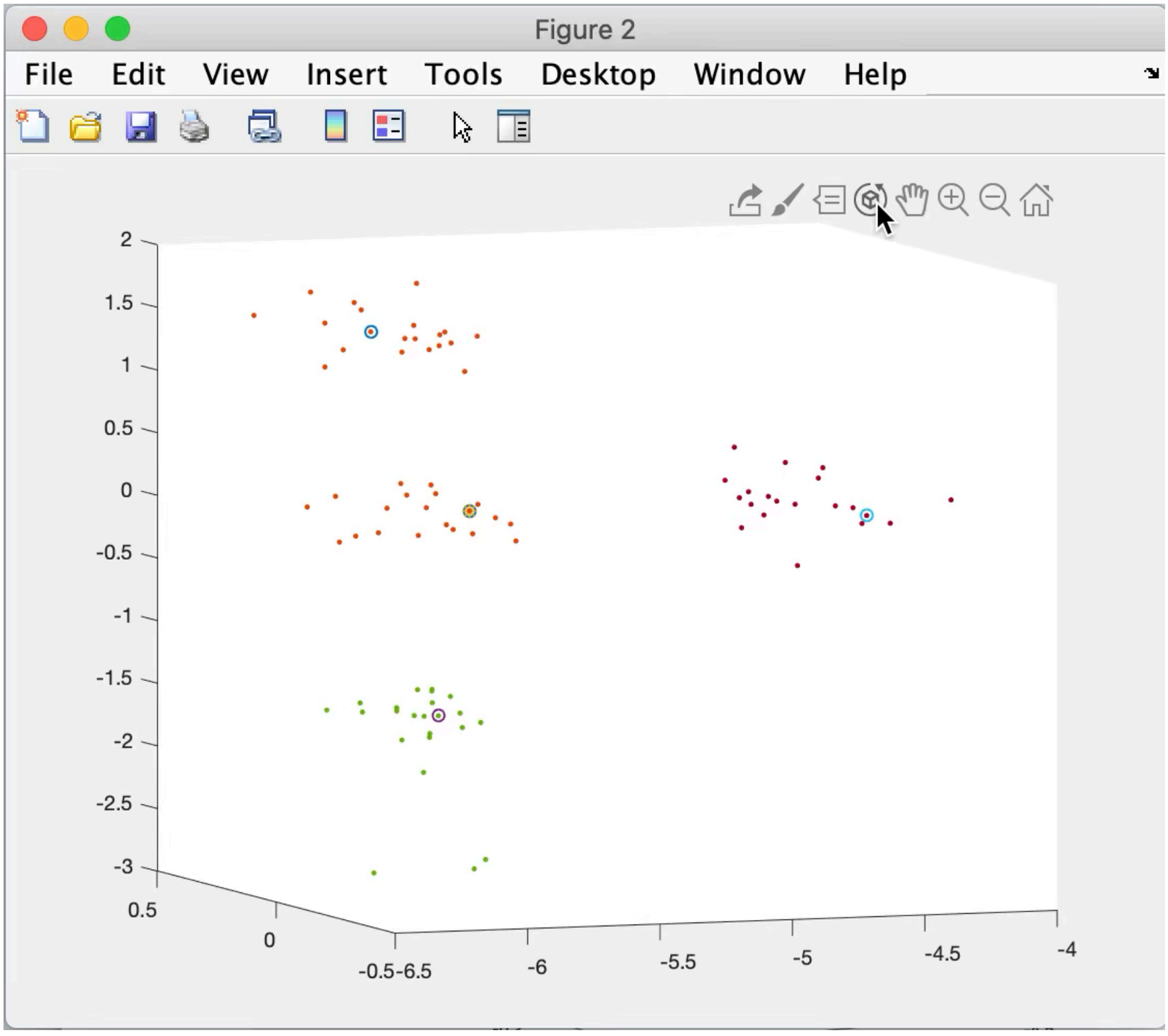
Update a single centroid



Memberships before updating

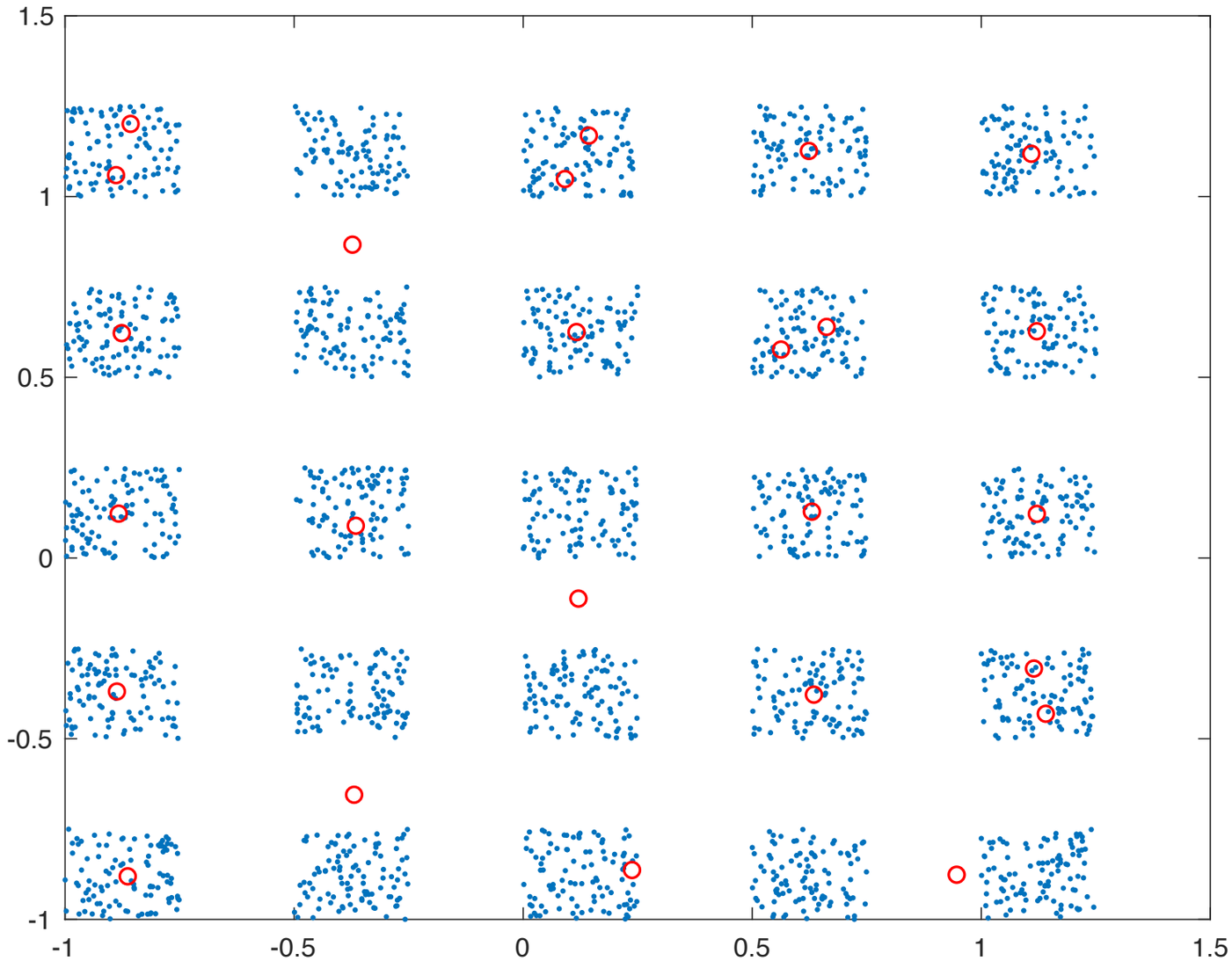


Memberships after updating



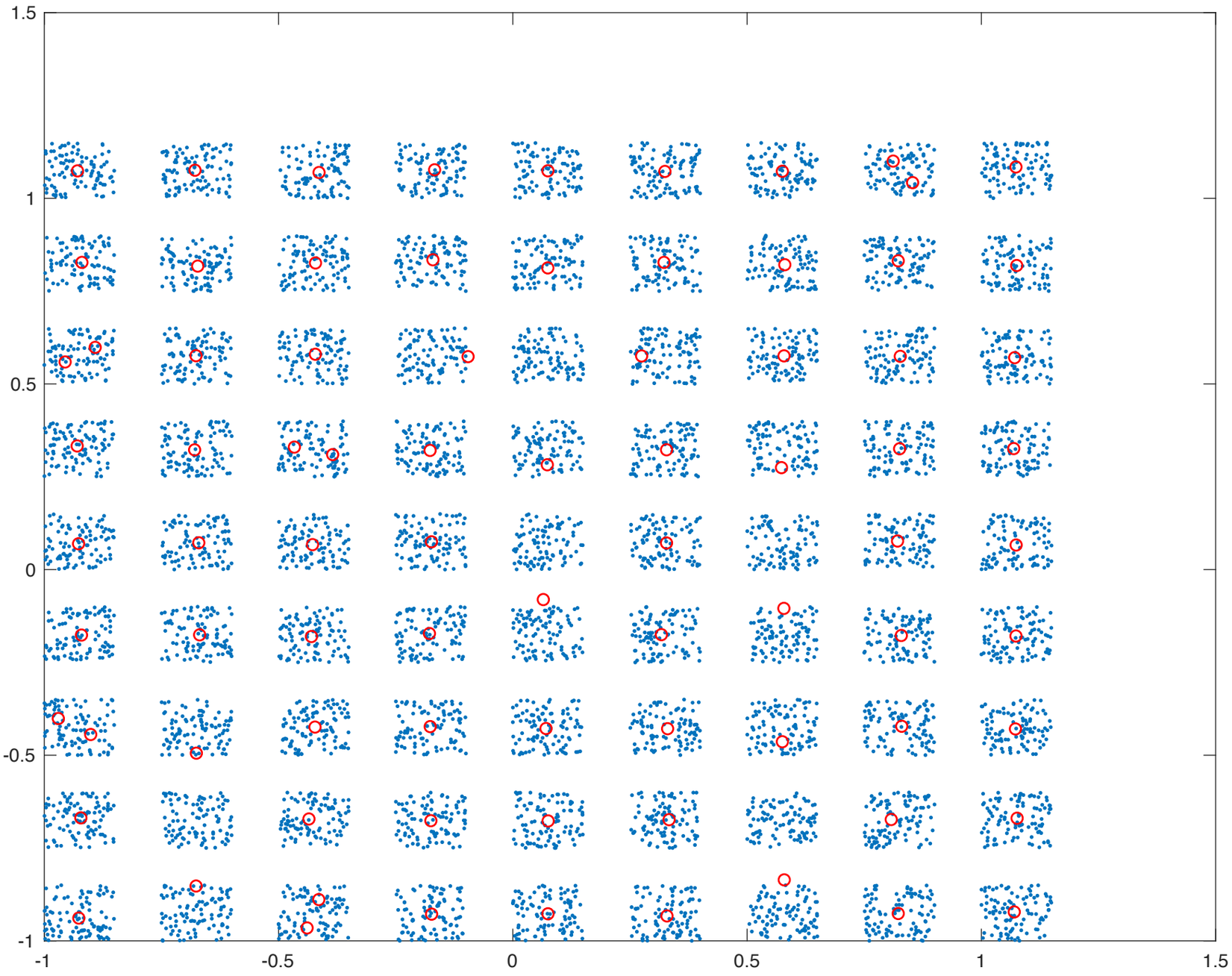
```
x1=linspace(-1,1,5);
x2=linspace(-1,1,5);
X=[];
for i=1:5
    for j=1:5
        X=[X;rand(100,2)*0.25+[ones(100,1)*x1(i) ones(100,1)*x2(j)]];
    end
end

plot(X(:,1),X(:,2),'.');
hold on;
[cidx, ctrs] = kmeans(X,25);
plot(ctrs(:,1),ctrs(:,2),'ro');
```


1.5
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5

```
K = 9;
x1=linspace(-1,1,K);
x2=linspace(-1,1,K);
X=[];
for i=1:K
    for j=1:K
        X=[X;rand(100,2)*0.15+[ones(100,1)*x1(i) ones(100,1)*x2(j)]];
    end
end

plot(X(:,1),X(:,2),'.');
hold on;
[cidx, ctrs] = kmeans(X,K^2);
plot(ctrs(:,1),ctrs(:,2),'ro');
```


1.5
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5

data_gen2.m

```
   %  Created by Jiann-Ming Wu on 10/23/18.
L = 6;
a(1,:)=linspace(-6,6,L);
a(2,:)=linspace(-6,6,L);
a(3,:)=linspace(-6,6,L);
X=[]; Y=[];
for i=1:L
   for j=1:L
      for k=1:L
         center=[a(1,i) a(2,j) a(3,k)];
         Xi=randn(20,3)*0.15+ ones(20,1)*center;
         X=[X;Xi];
         Y=[Y;center];

      end
   end
end
A=eye(3)+randn(3,3)*0.1;
X=X*A;
plot3(X(:,1),X(:,2),X(:,3),'.');
```

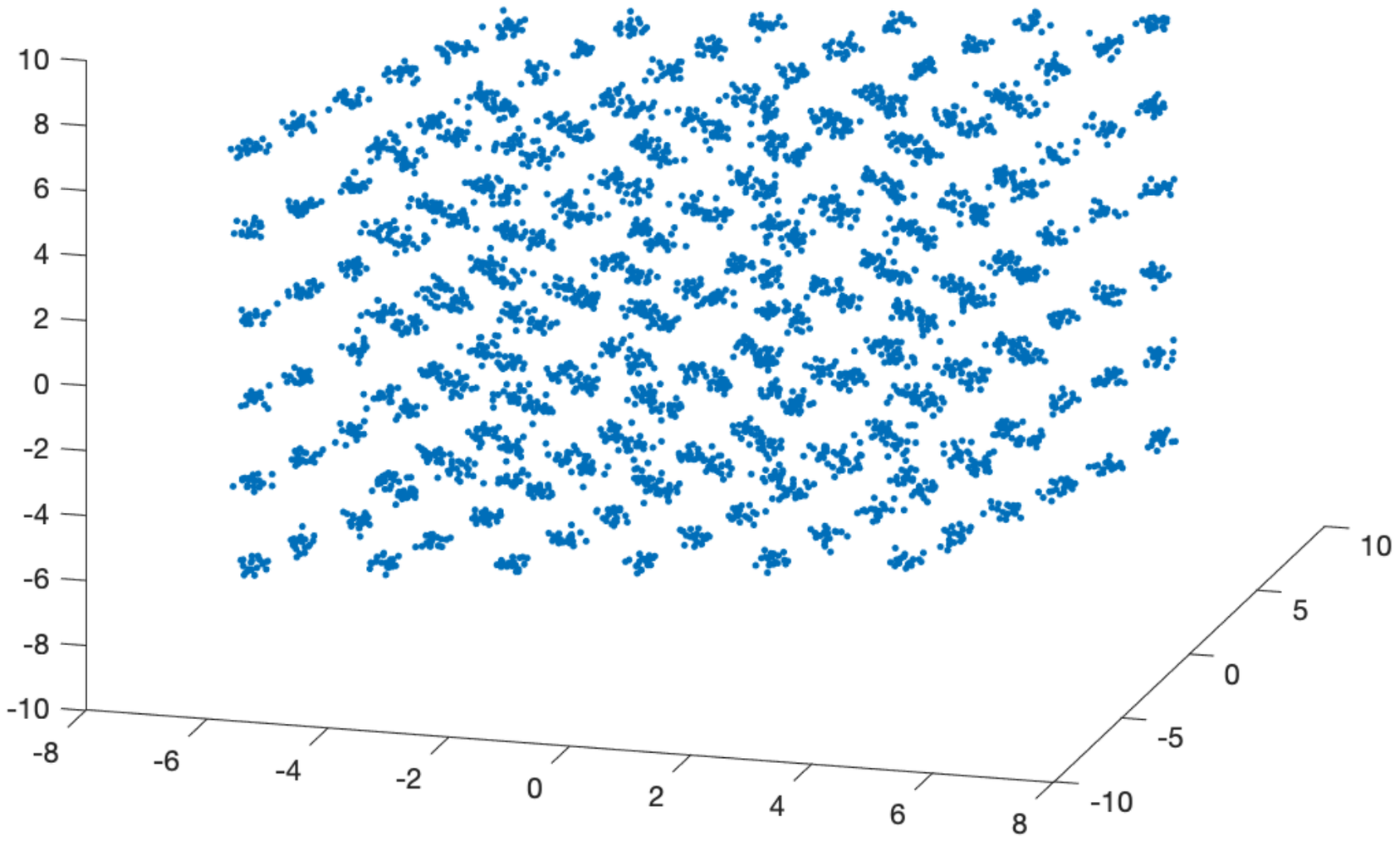
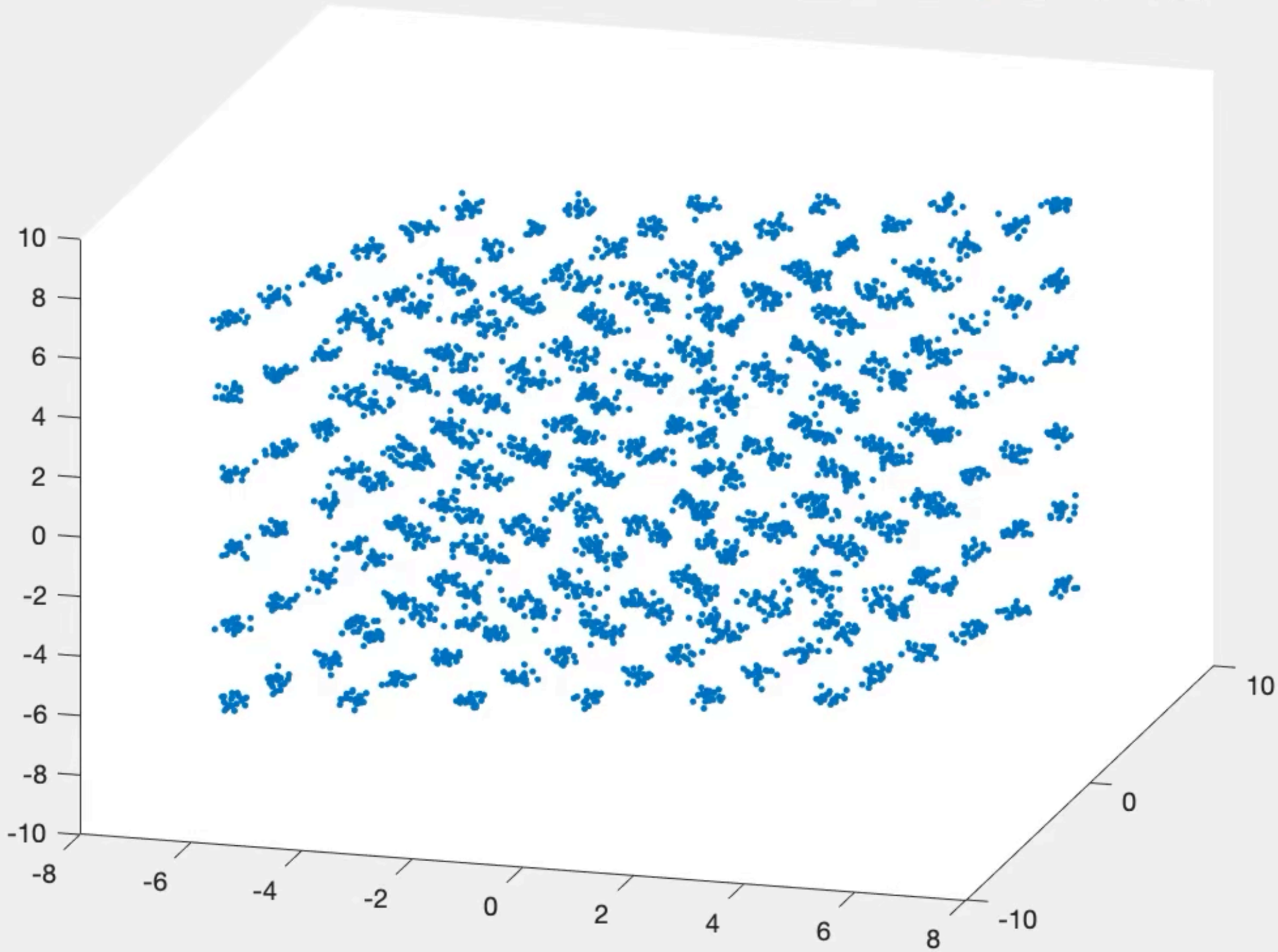

10
8
6
4
2
0
-2
-4
-6
-8
-10
-8
-6
-4
-2
0
2
4
6
8
10
5
0
-5
-10

Figure 1

File Edit View Insert Tools Desktop Window Help



```
   %  Created by Jiann-Ming Wu on 10/23/18.
L = 6;
a(1,:)=linspace(-6,6,L);
a(2,:)=linspace(-6,6,L);
a(3,:)=linspace(-6,6,L);
X=[]; Y=[];
for i=1:L
   for j=1:L
      for k=1:L
         center=[a(1,i) a(2,j) a(3,k)];
         Xi=randn(20,3)*0.15+ ones(20,1)*center;
         X=[X;Xi];
         Y=[Y;center];

      end
   end
end
A=eye(3)+randn(3,3)*0.1;
X=X*A;
plot3(X(:,1),X(:,2),X(:,3),'.');
hold on
[cidx, ctrs] = kmeans(X,L^3);
plot3(ctrs(:,1),ctrs(:,2),ctrs(:,3),'ro');
```

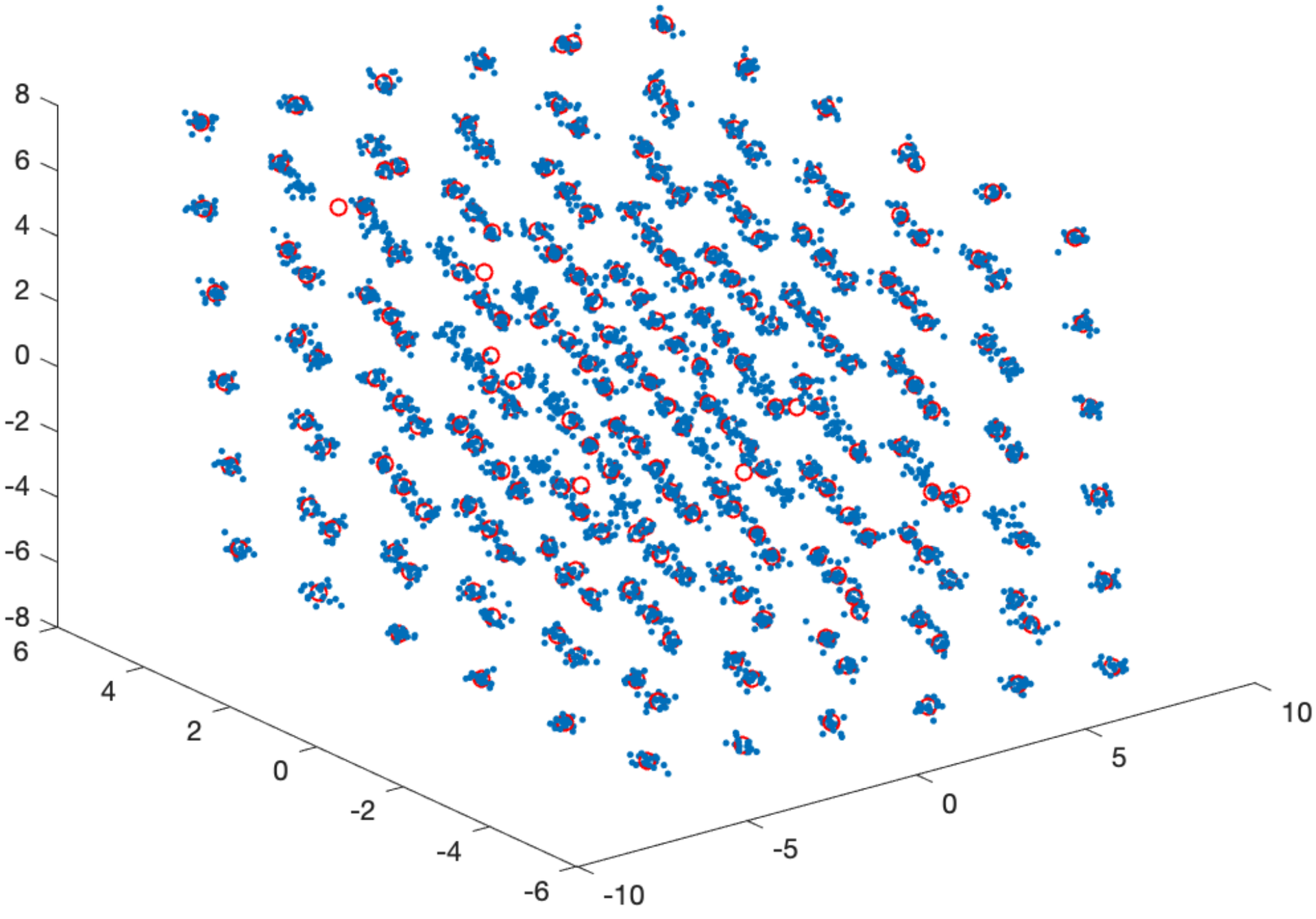
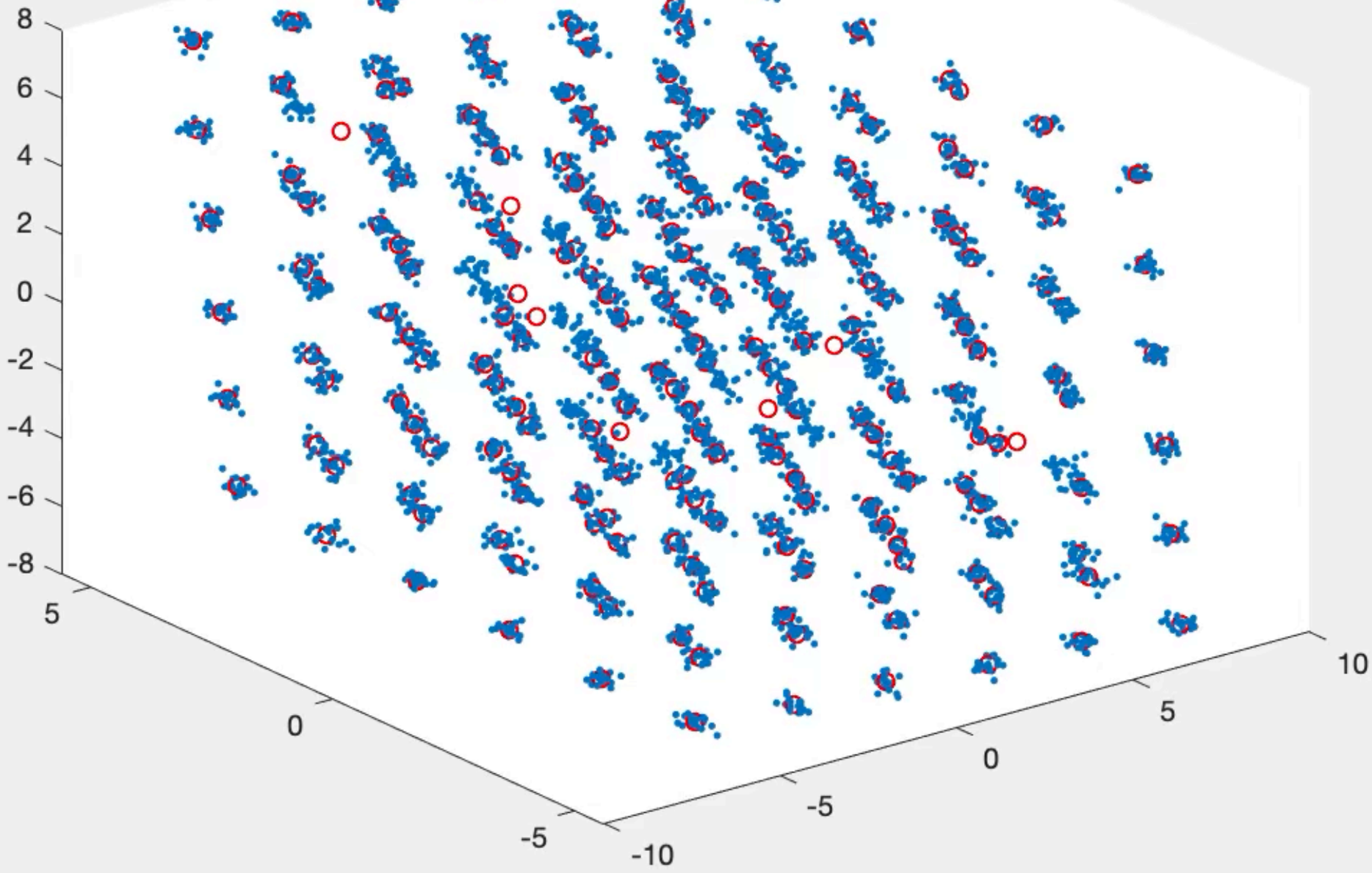

8
6
4
2
0
-2
-4
-6
-8
6
4
2
0
-2
-4
-6
-10
-5
0
5
10

Figure 1

File Edit View Insert Tools Desktop Window Help



```
function demo_my_kmeans()
    x1=linspace(-1,1,5);
    x2=linspace(-1,1,5);
    X=[];
    for i=1:5
        for j=1:5
            X=[X;rand(100,2)*0.25+[ones(100,1)*x1(i) ones(100,1)*x2(j)]];
        end
    end

    plot(X(:,1),X(:,2),'g.');
    hold on;
    Y = my_kmeans(X,25);
    plot(Y(:,1),Y(:,2),'ro');
end
```

```
function Y=my_kmeans(X,M)
    mean_X = mean(X);
    N = size(X,1);
    Y = ones(M,1)*mean_X + rand(M,2)*2-1;
    change = 1; ep = 10.^-6;
    while change > ep
        D=cross_dis(X,Y);
        [xx,v]=min(D');
        dis = [];
        for i=1:M
            ind=find(v == i);
            if length(ind) > 0
                Y_new(i,:) =mean(X(ind,:));
                dis = [dis xx(ind)];
            else
                Y_new(i,:) = rand(1,2)*2-1;
            end
        end
        change = mean(mean(abs(Y-Y_new)));
        fprintf('change %f dis %f \n',change, mean(dis));
        Y=Y_new;
    end
end
```

```
function D=cross_dis(X,Y)
    K=size(Y,1);N=size(X,1);
    A=sum(X.^2,2)*ones(1,K);
    C=ones(N,1)*sum(Y.^2,2)';
    B=X*Y';
    D=sqrt(A-2*B+C);
end
```

```
for i = 1:N
for j = 1:K
    D(i,j) = norm(X(I,:),Y(j,:)
    end
end
```

$$D_{ij} = (x_i - y_j)^T(x_i - y_j) = x_i^T x_i - 2x_i^T y_j + y_j^T y_j$$

```
X=rand(10000,10);
Y = rand(100,10);
    K=size(Y,1);N=size(X,1);
    tic
    A=sum(X.^2,2)*ones(1,K);
    C=ones(N,1)*sum(Y.^2,2)';
    B=X*Y';
    D=sqrt(A-2*B+C);
t1 = toc

tic

for i = 1:10000
for j = 1:100
      D2(i,j) = norm(X(i,:)-Y(j,:));
end
end
t2 = toc

sum(sum(abs(D-D2)))
```

K-means

- Consider fixed local means, y_k for all k
- Determine exclusive memberships for each data
- Minimize

$$E_i = \sum_k \xi_{ik} \parallel x_i - y_k \parallel^2$$

$$\sum_k \xi_{ik} = 1$$

$$\xi_{ik} \in \{0,1\}$$

ASSIGNMENT

$$E_i = \sum_k \xi_{ik} \parallel x_i - y_k \parallel^2$$

- is minimized by simply assigning one to ξ_{ik*}

$$x_i \quad is \quad closest \quad to \quad y_{k*}$$

Assignment

$x_i \quad is \quad closest \quad to \quad y_{k*}$

- is equivalent to

$$\| x_i - y_{k*} \| = min_k \| x_i - y_k \|$$

ASSIGNMENT

- x_i is simply assigned to a cluster whose representative is closest to x_i

$$\| x_i - y_{k*} \| = min_k \| x_i - y_k \|$$

- is equivalent to

```
D=cross_dis(X,Y);
[xx,v]=min(D');
```

$$k* = \arg\min_k \| x_i - y_k \|$$

Partition and Updating K-means

$$S_k = \{x_i \,|\, \xi_i = e_k \,|\, \xi_{ik} = 1\}$$

- Partition the whole data set into K non-overlapping subsets
- x_i is partitioned to S_k if $\xi_i = e_k$

```
for i=1:M
    ind=find(v == i);
    if length(ind) > 0
        Y_new(i,:) =mean(X(ind,:));
        dis = [dis xx(ind)];
    else
        Y_new(i,:) = rand(1,2)*2-1;
    end
end
```

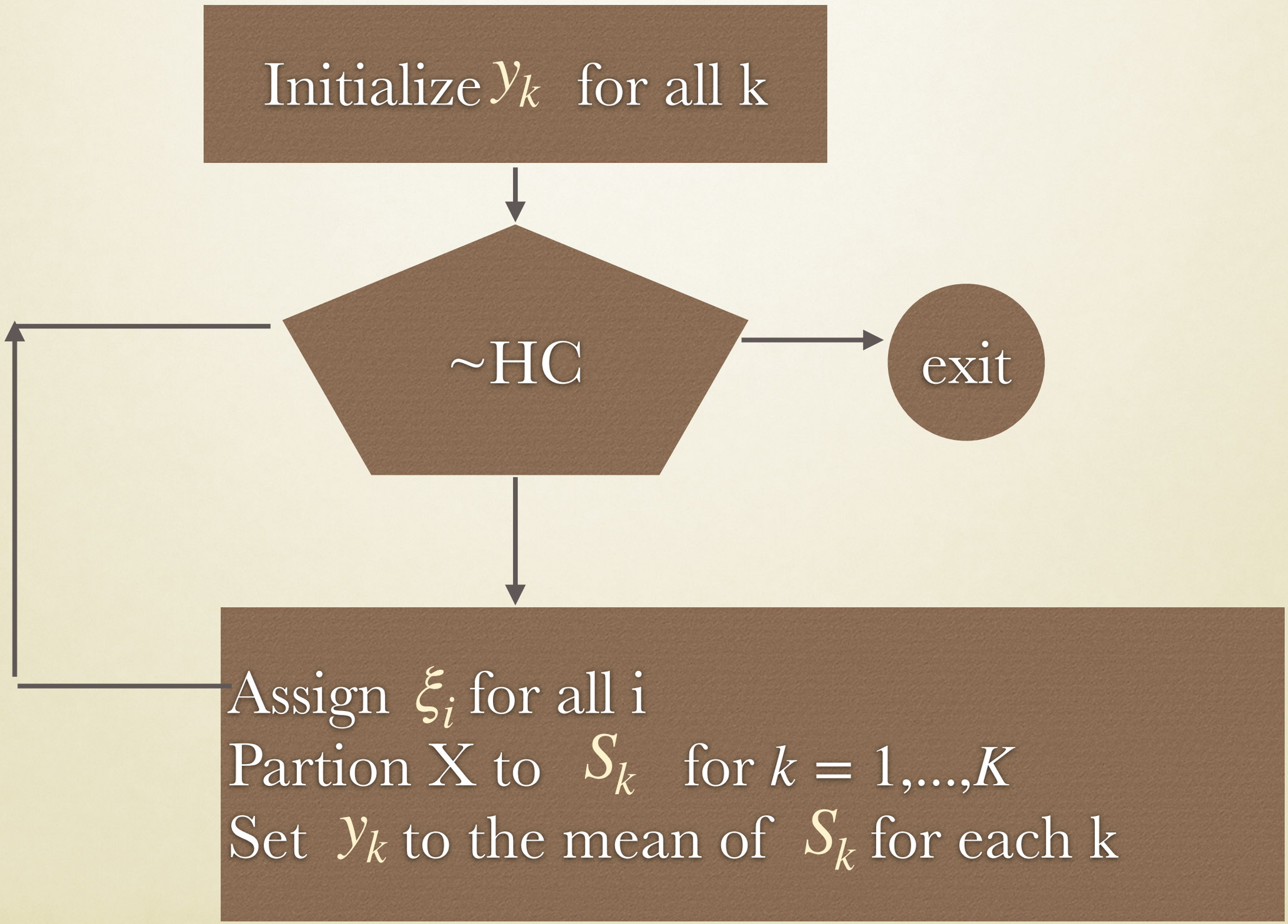
Updating K-means

$$y_k = \frac{1}{|S_k|} \sum_{i \in S_k} x_i$$

- Recalculate the mean of elements in S_k

```
Y_new(i,:) =mean(X(ind,:));
```

While-looping



Synchronous update: All centroids are updated at the same time

Annealed Expectation Maximization

Mahalanobis Distance

1. Set β to a sufficiently low value

$$A = 0.01 \times I$$

$$\mathbf{y}_k \approx \frac{1}{N}\sum_t \mathbf{x}[t], v_k[t] \approx \frac{1}{K}$$

2. E step : update v using (E1)

3. M step : update y using (M1)

update A using (M2)

4. If $\frac{1}{N}\sum_t \sum_k v_k[t]^2 \geq 0.98$, halt

else $\beta \leftarrow \frac{1}{0.98}\beta$, goto step 2

Set β to a sufficiently low value

$$\mathbf{y}_k \approx \frac{1}{N}\sum_t \mathbf{x}[t],\ v_k[t] \approx \frac{1}{K}$$

$$\frac{1}{N}\sum_t \sum_k v_k[t]^2 \geq 0.98$$

. E step : update v using (E1)

. M step : update y using (M1)

: $\beta \leftarrow \frac{\beta}{0.995}$

```
function [Y Q]=annealed_kmeans2(X,K)
[N d]=size(X);
mean_x = mean(X);
B=0.1;stability=1/K;
Y=rand(K,d)*0.2-0.1+ones(K,1)*mean_x;
HC=0; Q=ceil(rand(N,1)*size(Y,1))';
ep=10^-10;
while ~HC
   if stability < 1/K*2
      Y=Y+rand(K,d)*0.02-0.01;
   end
   D=cross_dis(X,Y);
   U= exp(-B*D);
   S=sum(U,2);
   ind_zero=find(S < ep);
   S(ind_zero)=10^-6;
   n_empty_node=length(ind_zero);
   Q=U./(S*ones(1,K));
   stability=mean(sum(Q.^2,2));
   E=mean(sum(Q.*D.^2,2));
   stability=stability*K/(K-n_empty_node);
   for k=1:K
      a=sum(Q(:,k));
      b=sum(X.*( Q(:,k)*ones(1,d)));
      if a > 0
      Y(k,:) = b/a;
      end
   end
   fprintf('B %f sta %f E %f n %d\n',B,stability,E,n_empty_node);
   if stability > 0.98
      HC=1;
   end
   B=B/0.995;
end
```

EM Formula

$$v_{ik} \equiv \left\langle \xi_{ik} \right\rangle = \Pr(\xi_i = e_k^K) = \frac{\exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)}{\sum_{k=1}^{K} \exp(-\beta \| \mathbf{x}_i - \mathbf{y}_k \|^2)} \quad \text{(E1)}$$

$$\mathbf{y}_k = \frac{\sum_{i=1}^{N} \left\langle \xi_{ik} \right\rangle \mathbf{x}_i}{\sum_{i=1}^{N} \left\langle \xi_{ik} \right\rangle}$$

```
D=cross_dis(X,Y);
U= exp(-B*D);
S=sum(U,2);
ind_zero=find(S < ep);
S(ind_zero)=10^-6;
n_empty_node=length(ind_zero);
Q=U./(S*ones(1,K));
```

Derivation from Free Energy

Standard basis K =3

$$\Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K) = \frac{\exp(\beta u_k[t])}{\sum_{j=1}^{K} \exp(\beta u_j[t])}$$

$e_1 = [1,0,0]$

$e_2 = [0,1,0]$

$e_3 = [0,0,1]$

$$\text{Expectation of } \boldsymbol{\delta}[t] = \sum_{k=1}^{K} \mathbf{e}_k^K \Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K)$$

$$\text{Entropy of } \boldsymbol{\delta}[t] = -\sum_{k=1}^{K} \Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K) \ln \Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K)$$

$$\Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K) = \frac{\exp(\beta u_k[t])}{\sum_{j=1}^{K} \exp(\beta u_j[t])} \equiv v_k[t] = \langle \delta_k[t] \rangle$$

$$\mathrm{H_t} \equiv \text{Entropy of } \boldsymbol{\delta}[t] = -\sum_{k=1}^{K} \Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K) \ln \Pr(\boldsymbol{\delta}[t] = \mathbf{e}_k^K)$$

$$= -\sum_{k=1}^{K} v_k[t](\beta u_k[t] - \ln \sum_{j=1}^{K} \exp(\beta u_j[t]))$$

$$= -\beta \sum_{k=1}^{K} v_k[t] u_k[t] + \sum_{k=1}^{K} v_k[t] \ln \sum_{j=1}^{K} \exp(\beta u_j[t])$$

Free Energy

- A combination of Mean Energy and Negative Entropy

$$\mathrm{F} = \left\langle \mathrm{E}(\boldsymbol{\delta}) \right\rangle - \frac{1}{\beta} H(\boldsymbol{\delta})$$

$$\approx \mathrm{E}(\left\{ \left\langle \boldsymbol{\delta}[t] \right\rangle \right\}) - \frac{1}{\beta} \sum_{t} H(\boldsymbol{\delta}[t])$$

Derived based on

Kullback - Leiberg(KL) divergence

$$= E(v) - \beta \sum_{k=1}^{K} v_k[t] u_k[t] + \ln \sum_{j=1}^{K} \exp(\beta u_j[t])$$

Mean Field Equations

$$\frac{\partial \mathrm{F}}{\partial v_k[t]} = 0, \frac{\partial \mathrm{F}}{\partial u_k[t]} = 0, \forall k, t$$

$$u_k[t] = -\frac{\partial \mathrm{E}(\mathbf{v})}{\partial v_k[t]},$$

$$v_k[t] = \frac{\exp(\beta u_k[t])}{\sum_j \exp(\beta u_j[t])}$$

Free Energy

$$L(\boldsymbol{\delta}, \mathbf{y}, \mathbf{A}) = \sum_k L_k$$

$$= \frac{1}{2} \sum_t \sum_k \delta_k[t](\mathbf{x}[t] - \mathbf{y}_k)^T \mathbf{A}(\mathbf{x}[t] - \mathbf{y}_k) - \frac{N}{2} \log|\mathbf{A}|$$

$$\mathrm{F}(\mathbf{v}, \mathbf{u}, \mathbf{y}, \mathbf{A})$$

$$= \mathrm{E}(\mathbf{v}, \mathbf{y}, \mathbf{A}) + \sum_t \sum_k v_k[t] u_k[t]$$

$$- \frac{1}{\beta} \sum_t \ln \sum_j \exp(\beta u_j[t])$$

$$\frac{\partial \mathrm{F}}{\partial v_k[t]} = 0, \frac{\partial \mathrm{F}}{\partial u_k[t]} = 0, \forall k, t$$

$$\frac{\partial \mathrm{F}}{\partial \mathbf{y}_k} = \frac{dL(\mathbf{y} \mid \mathbf{v}, \mathbf{A})}{d\mathbf{y}_k} = 0, \quad \text{(M1)}$$

Large Scale data clustering
Parallel and distributed codes

2021

J.M. Wu

Large scaled data clustering

- Cross Distance
- Parallel and distributed codes of Cross distances
- Hierarchical clustering models
- Codes : annealed K-Means, Annealed EM
- Numerical simulations

```
function D=cross_dis(X,Y)
K=size(Y,1);N=size(X,1);
A=sum(X.^2,2)*ones(1,K);
C=ones(N,1)*sum(Y.^2,2)';
B=X*Y';
D=sqrt(A-2*B+C);
```

$$(x-y)^T(x-y) = x^Tx - 2x^Ty + y^Ty$$

Case 1

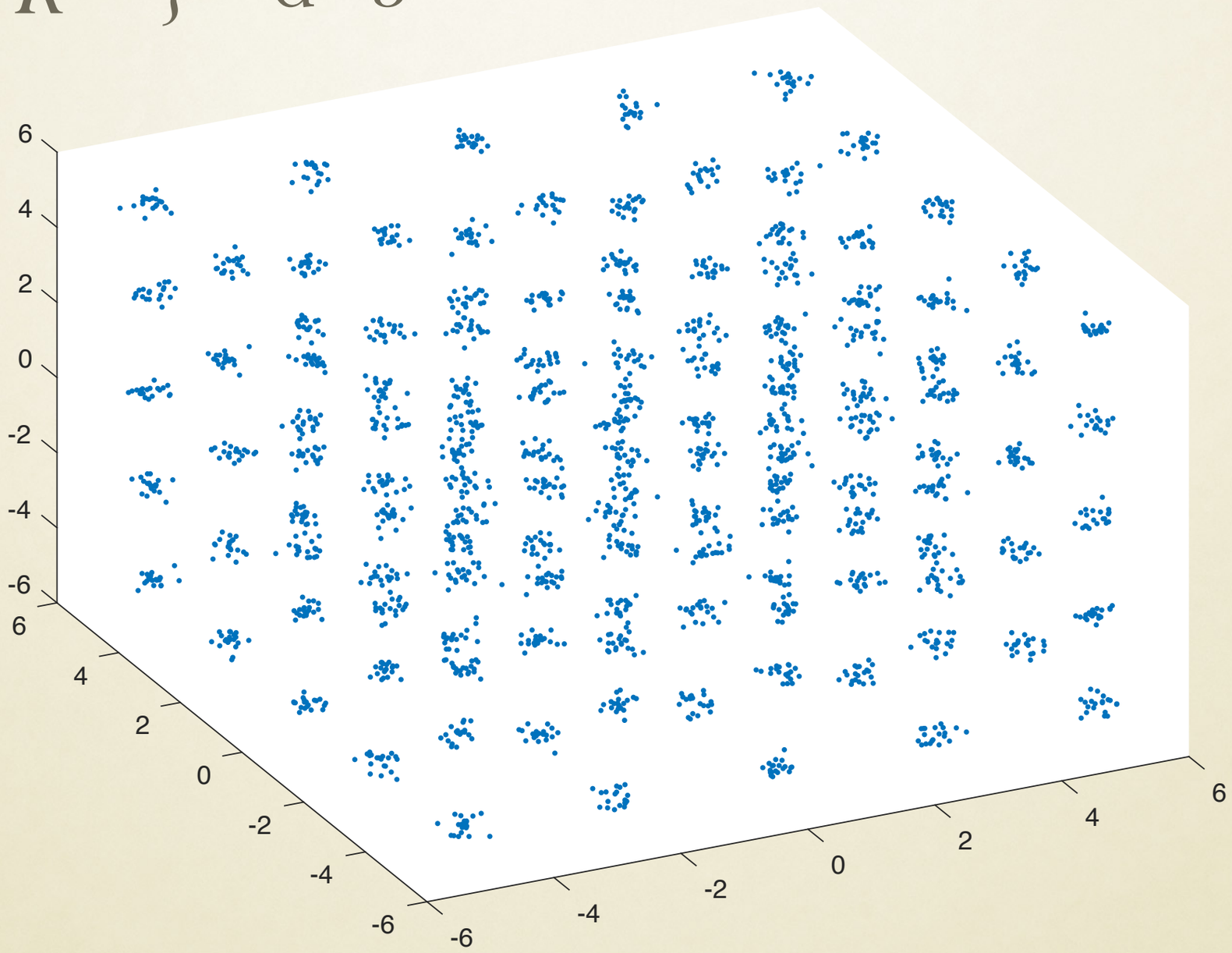
- dimension = 3
- 2500 data points
- 125 centers
- cross distances between data points and centers
- 2500x125

data_gen.m

```
clear all
L=5;
a(1,:)=linspace(-5,5,L);
a(2,:)=linspace(-5,5,L);
a(3,:)=linspace(-5,5,L);
X=[]; Y=[];
for i=1:L
    for j=1:L
        for k=1:L
            center=[a(1,i) a(2,j) a(3,k)];
            Xi=randn(20,3)*0.15+ ones(20,1)*center;
            X=[X;Xi];
            Y=[Y;center];

        end
    end
end
plot3(X(:,1),X(:,2),X(:,3),'.');
```

$$X=\{x[t]\in R^{d}\ \}\quad d=3$$

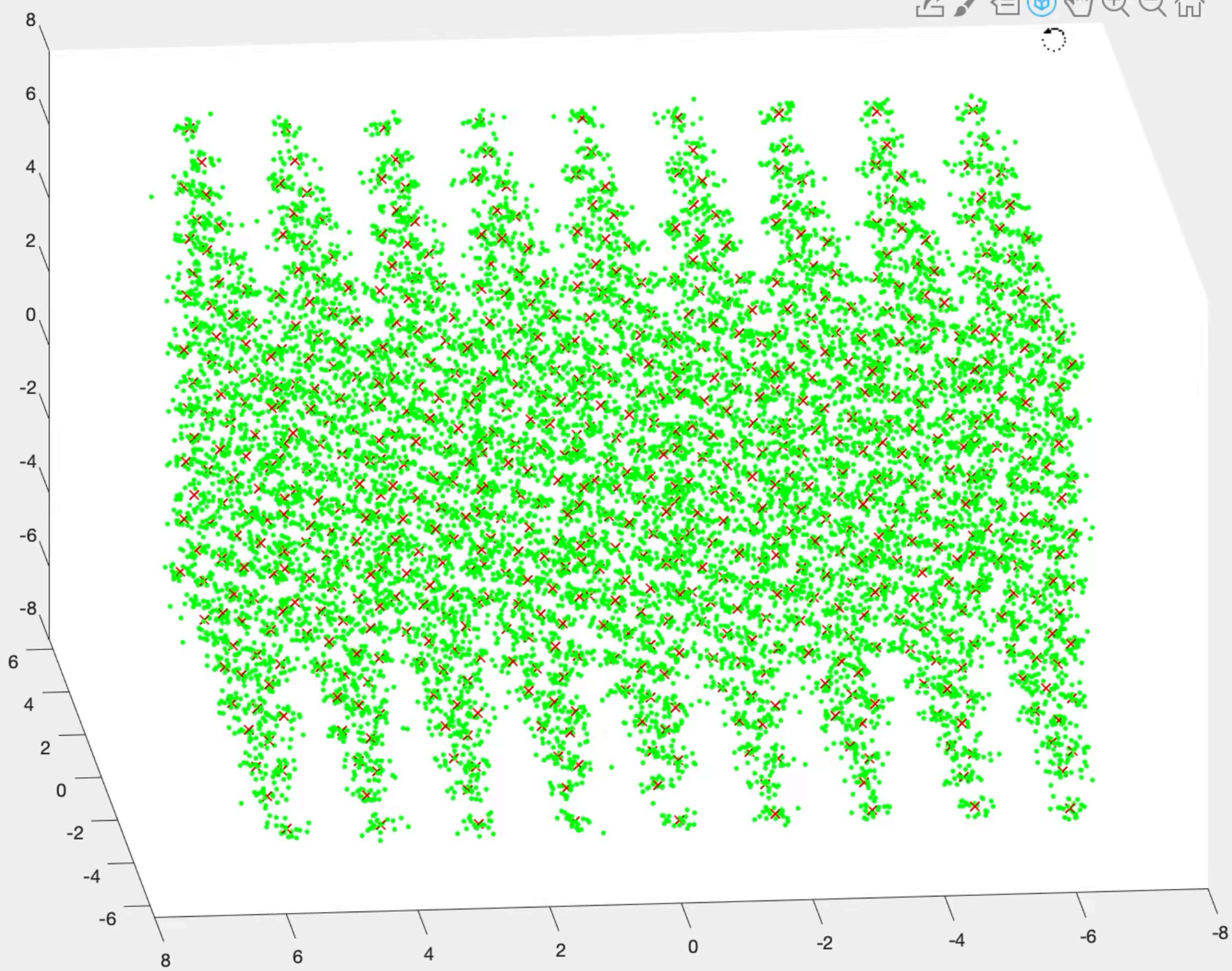


- D=cross_dis(X,Y);
- >> size(D)
- ans =
- 2500 125
- >> tic;D=cross_dis(X,Y);toc
- Elapsed time is 0.046284 seconds.

```
function [Y Q]=annealed_kmeans2(X,K)
[N d]=size(X);
mean_x = mean(X);
B=0.1;stability=1/K;
Y=rand(K,d)*0.2-0.1+ones(K,1)*mean_x;
HC=0; Q=ceil(rand(N,1)*size(Y,1))';
ep=10^-10;
while ~HC
   if stability < 1/K*2
      Y=Y+rand(K,d)*0.02-0.01;
   end
   D=cross_dis(X,Y);
   U= exp(-B*D);
   S=sum(U,2);
   ind_zero=find(S < ep);
   S(ind_zero)=10^-6;
   n_empty_node=length(ind_zero);
   Q=U./(S*ones(1,K));
   stability=mean(sum(Q.^2,2));
   E=mean(sum(Q.*D.^2,2));
   stability=stability*K/(K-n_empty_node);
   for k=1:K
      a=sum(Q(:,k));
      b=sum(X.*( Q(:,k)*ones(1,d)));
      if a > 0
      Y(k,:) = b/a;
      end
   end
   fprintf('B %f sta %f E %f n %d\n',B,stability,E,n_empty_node);
   if stability > 0.98
      HC=1;
   end
   B=B/0.995;
end
```

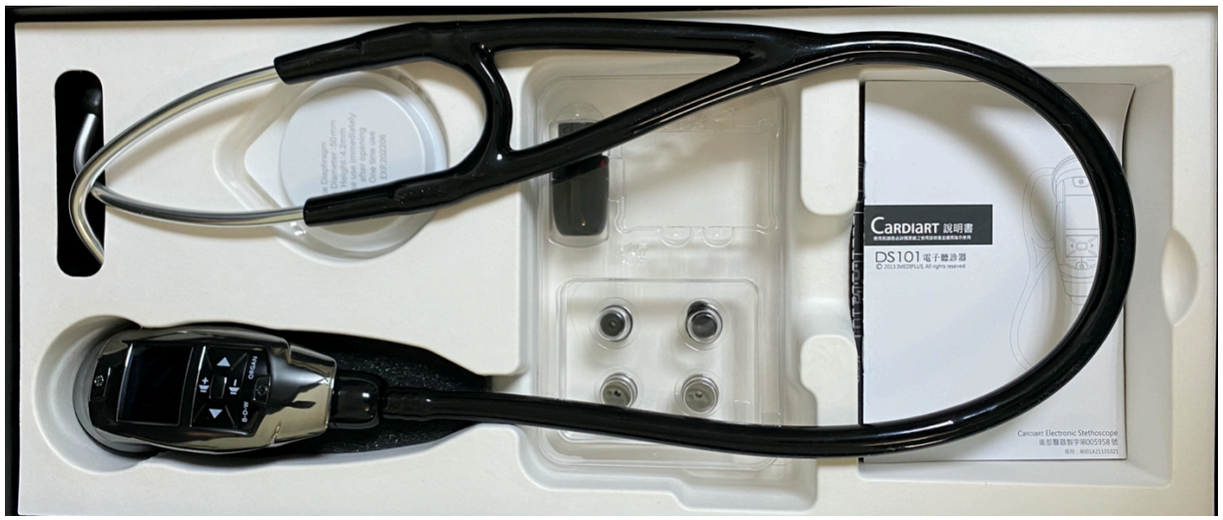
Figure 8

File Edit View Insert Tools Desktop Window Help



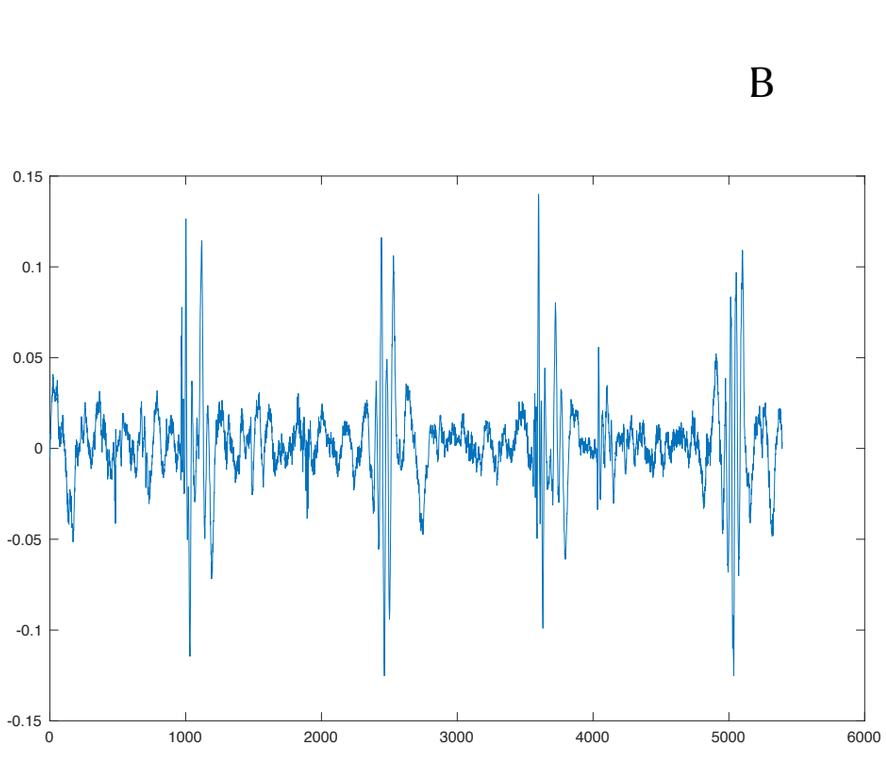
Extremely high-dimensional patterns

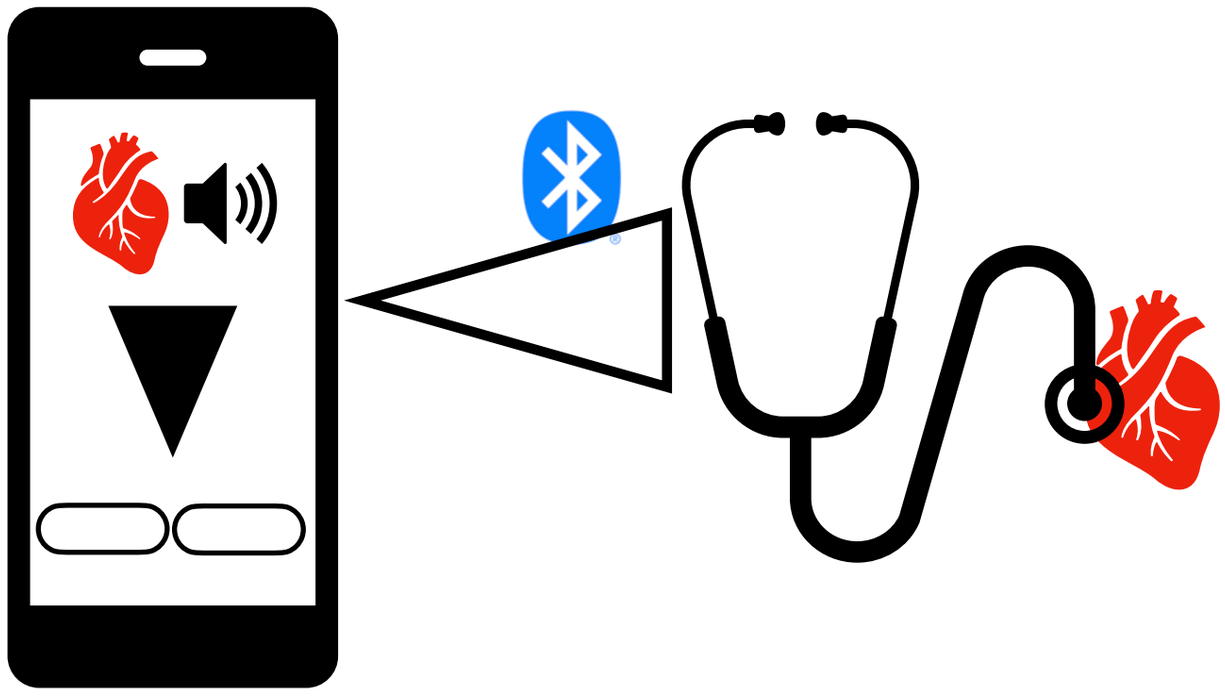
A



A stethoscope equipped with BLUETOOTH-4.0

B



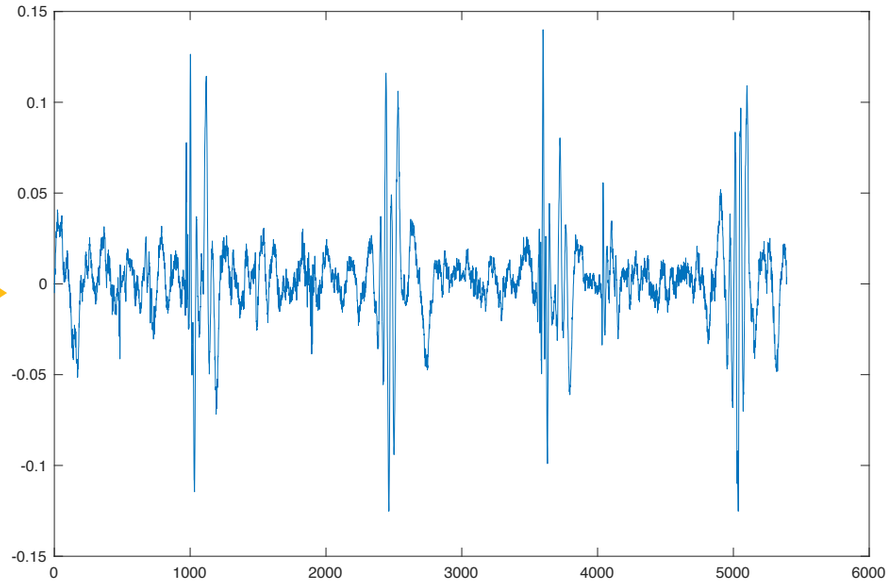


Record heart sound and transmit to a mobile device for analysis

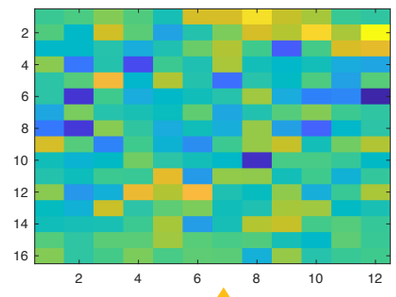
Heart sounds

There are totally N = 12900
16x12 MFCC patterns

Murmur heart sounds

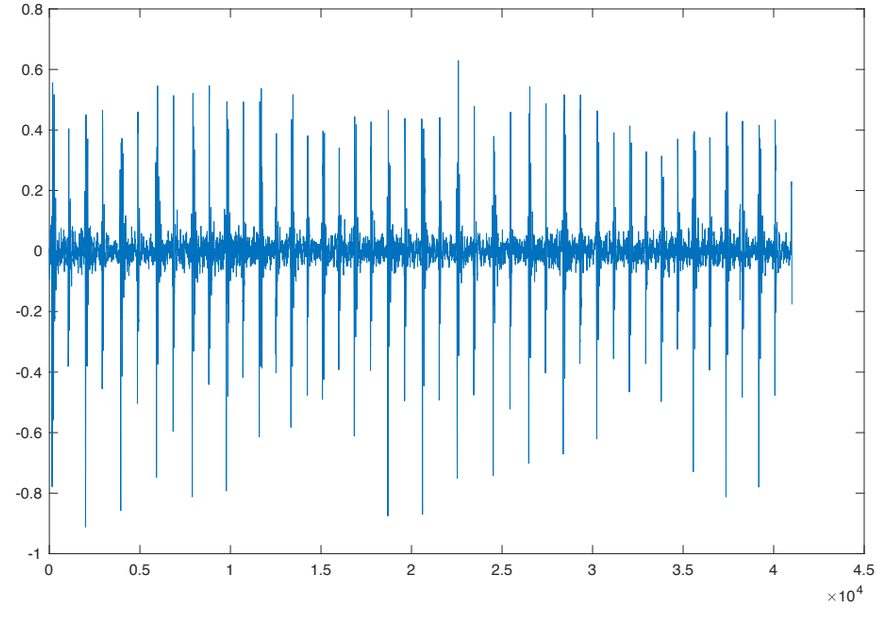


N_{MURMUR} = 9300
MFCC PATTERNS



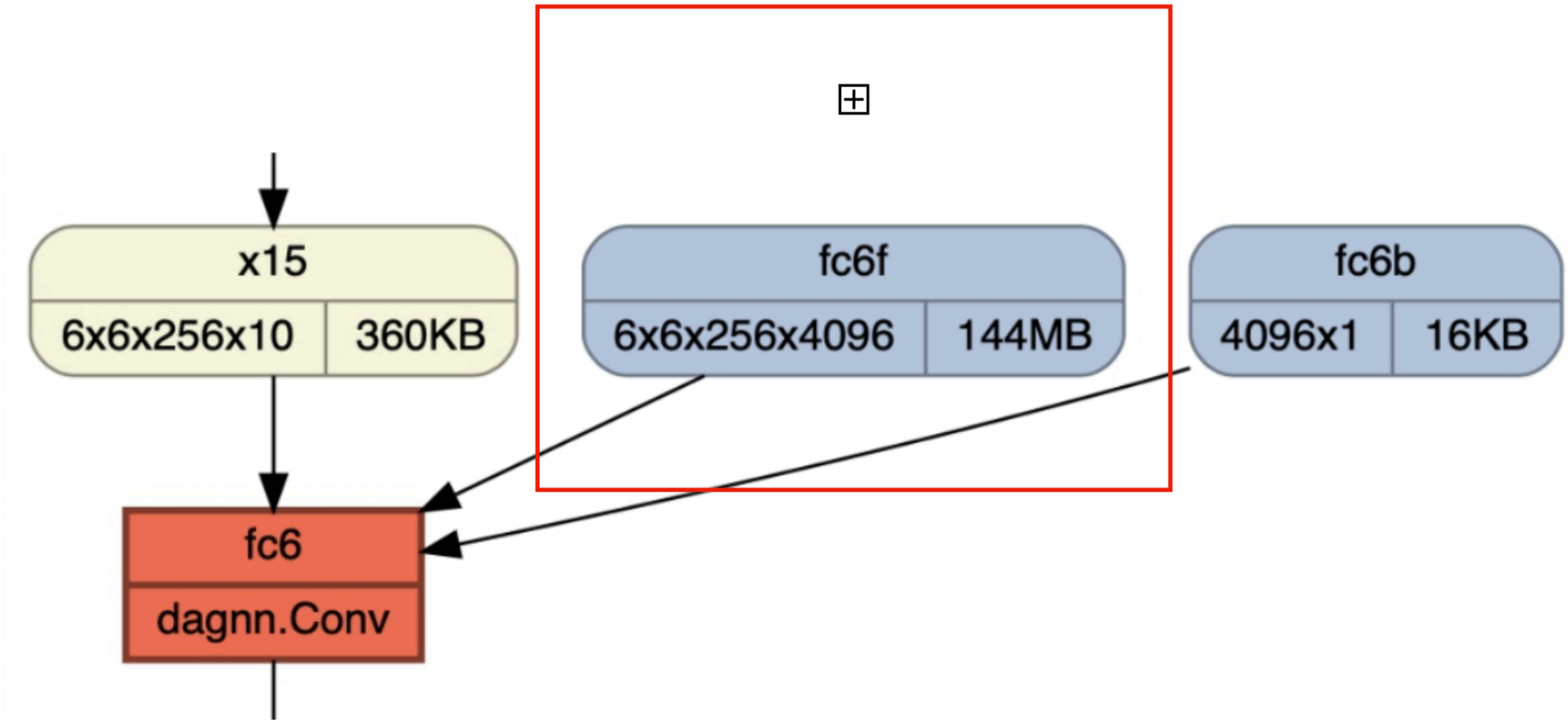
one 16x12 MFCC pattern

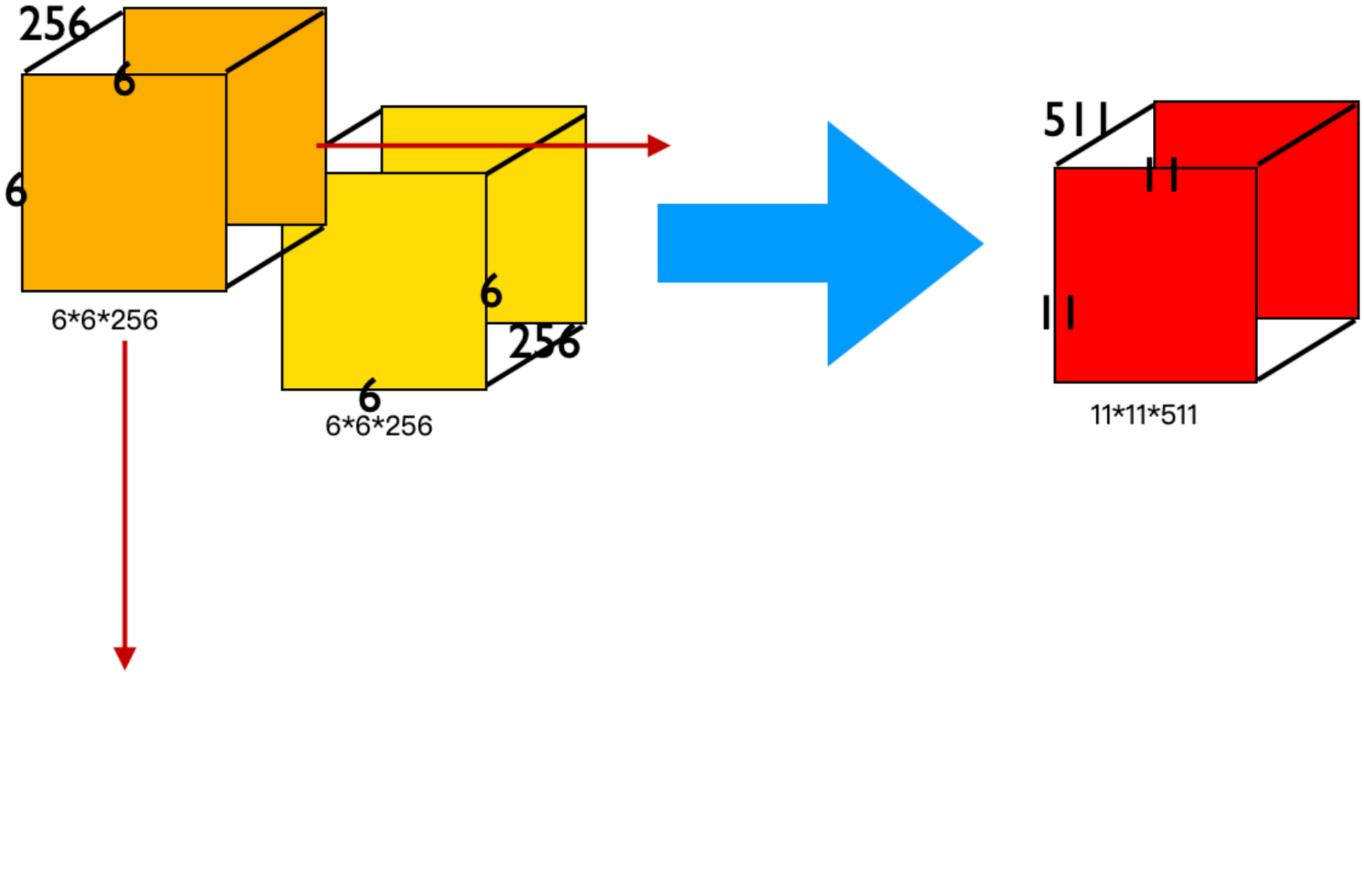
Normal heart sounds



N_{NORMAL}= 3600
MFCC PATTERNS

MFCC: Mel-Frequency Ceptral Coefficients


x15
6x6x256x10
360KB
fc6f
6x6x256x4096
144MB
fc6b
4096x1
16KB
fc6
dagnn.Conv


256
6
6
6*6*256
6
256
6
6*6*256
511
11
11
11*11*511

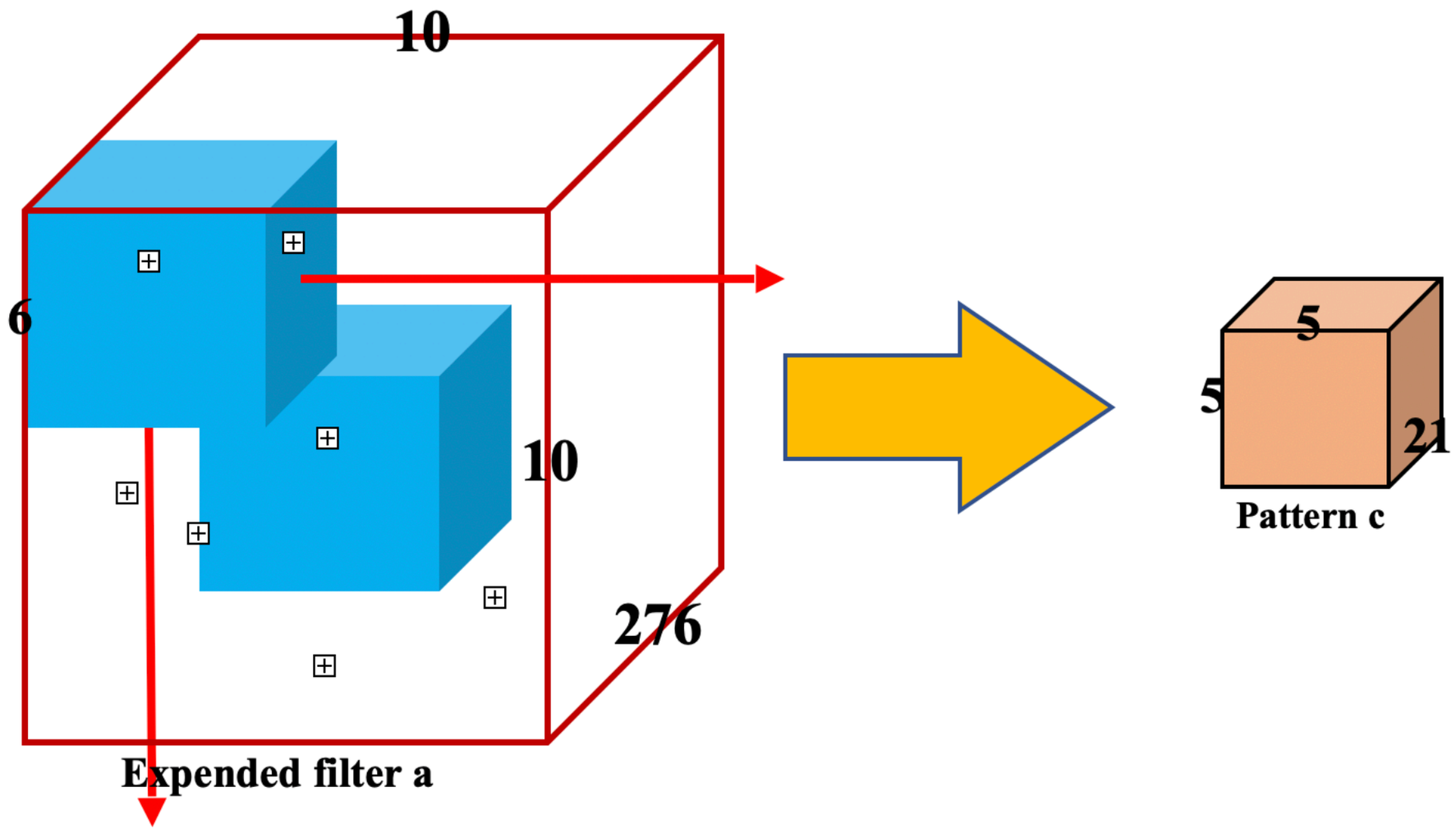


Figure 13. Convolution of two high-dimensional 3D filters.

CASE 2

- dimension = 13
- data points 1998000
- centers 12000
- X= rand(198000,13); Y=rand(12000,13);
-

- X= rand(199800,13); Y=rand(12000,13);
- tic;D=cross_dis(X,Y);toc

Error using *
Requested 199800x10000 (14.9GB) array exceeds maximum array size preference. Creation of arrays greater than this limit may take a long time and cause MATLAB to become unresponsive. See array size limit or preference panel for more information.

Error in **cross_dis** (line 3)
A=sum(X.^2,2)*ones(1,K);

BATCHES

- X= rand(100*4000,13); Y=rand(12000,13);
- x_batch =4000; y_batch=4000;
- x_batch_num = 100*4000/x_batch;
- y_batch_num =12000/y_batch;
- for i=1:x_batch_num
- XX{i}=X(1+(i-1)*x_batch:i*x_batch);
- end
- for j=1:y_batch_num
- YY{j}=Y(1+(j-1)*x_batch:j*x_batch);
- end
-

```
D = zeros(x_batch_num,x_batch,y_batch);
for i=1:x_batch_num
    D(i,:,:)=zeros(x_batch,y_batch);
end
parfor i=1:x_batch_num
   D(i,:,:)=cross_dis(XX{i},YY{1});
end
```

Images and Sounds

- Facial images
  - http://www.face-rec.org/databases/
- Hand-writing character images
- MFCC features of speeches
  - https://sounds.bl.uk/
-