

Clustering: Models and Algorithms

Shikui Tu

2018-03-08

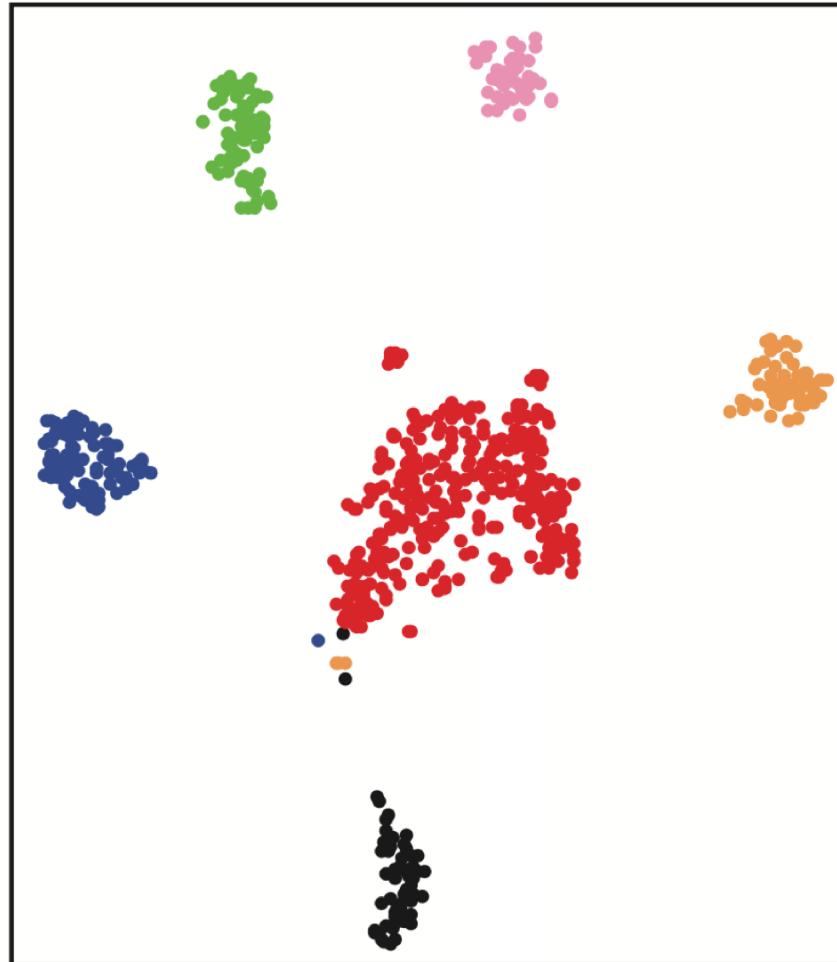
Outline

- Clustering
 - K-mean clustering, hierarchical clustering
- Adaptive learning (online learning)
 - CL, FSCL, RPCL
- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood

What is clustering?

例子：不同类型的癌细胞会各自聚在一起

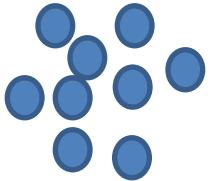
物以类聚



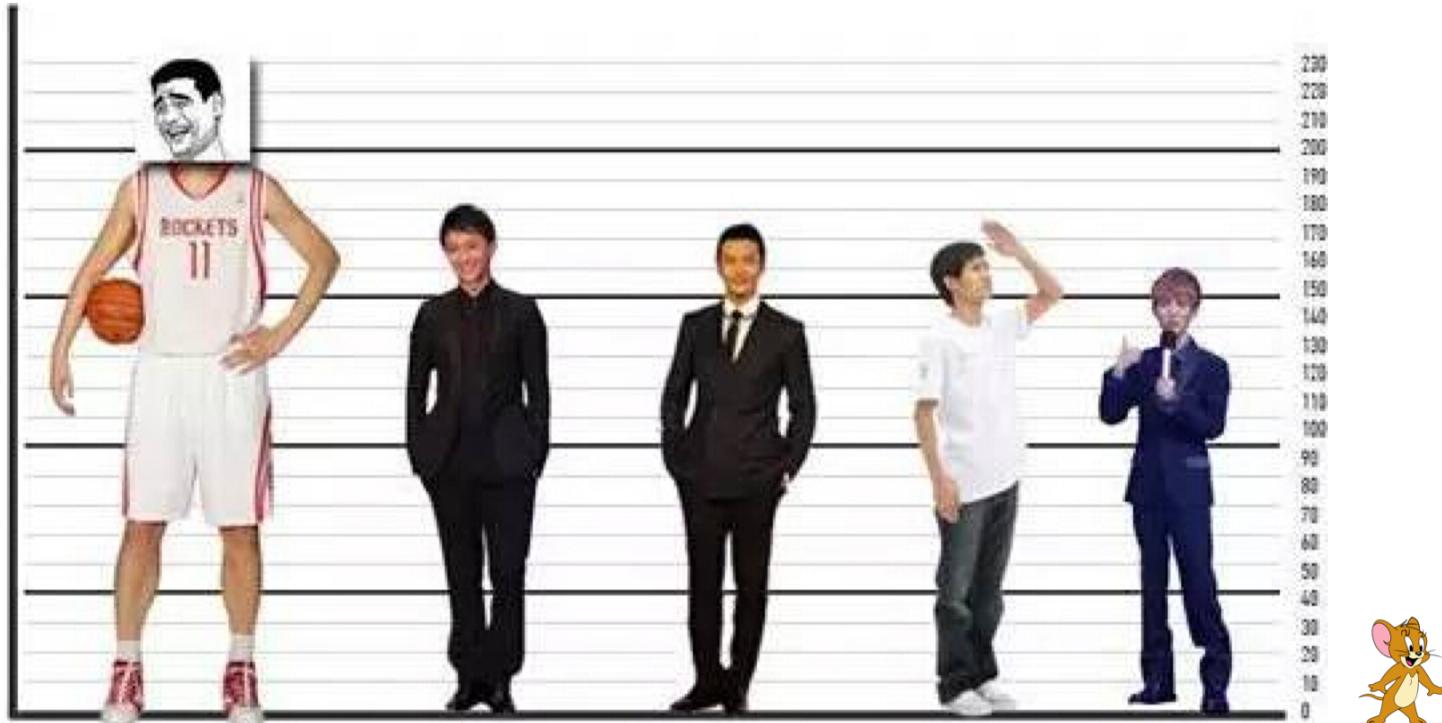
Six malignant tumors (melanoma)

8 APRIL 2016 • VOL 352 ISSUE 6282, SCIENCE

How to represent a cluster



- 例如：将每个人的身高记下来



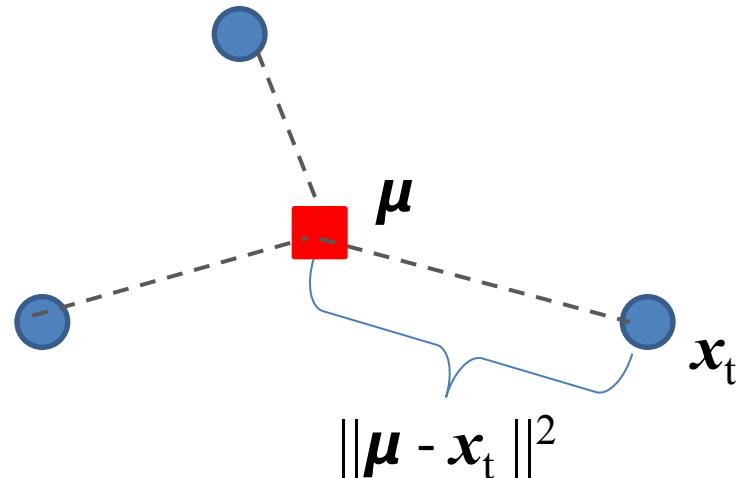
但是，如果只能记一个身高数值...

平均值

总误差最小

How to define error?

Square distance:



$$\|\mu - x_1\|^2 + \|\mu - x_2\|^2 + \|\mu - x_3\|^2$$

可以证明：当 μ 是所有数据点的均值时，平方距离和最小

Matrix derivatives

$$\left[\frac{\partial \mathbf{x}}{\partial y} \right]_i = \frac{\partial x_i}{\partial y} \quad \left[\frac{\partial x}{\partial \mathbf{y}} \right]_i = \frac{\partial x}{\partial y_i} \quad \left[\frac{\partial \mathbf{x}}{\partial \mathbf{y}} \right]_{ij} = \frac{\partial x_i}{\partial y_j}$$

$$\frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a} \tag{69}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a} \mathbf{b}^T \tag{70}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X}^T \mathbf{b}}{\partial \mathbf{X}} = \mathbf{b} \mathbf{a}^T \tag{71}$$

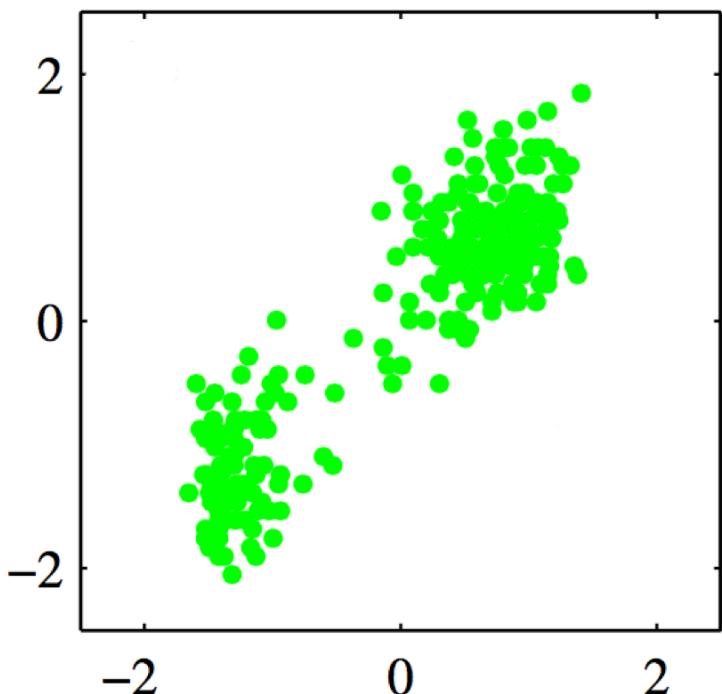
$$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T \mathbf{X}^T \mathbf{a}}{\partial \mathbf{X}} = \mathbf{a} \mathbf{a}^T \tag{72}$$

$$\frac{\partial \det(\mathbf{X})}{\partial \mathbf{X}} = \det(\mathbf{X}) (\mathbf{X}^{-1})^T \tag{49}$$

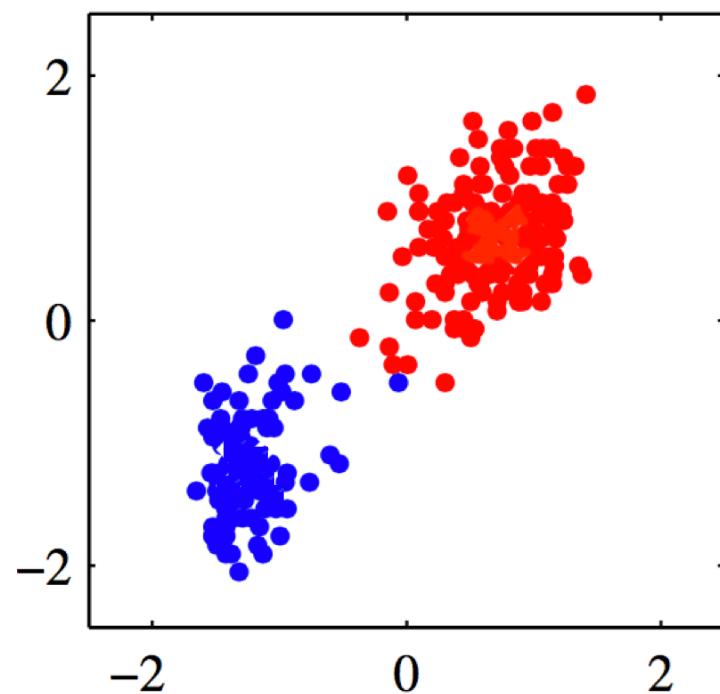
$$\frac{\partial \mathbf{Y}^{-1}}{\partial x} = -\mathbf{Y}^{-1} \frac{\partial \mathbf{Y}}{\partial x} \mathbf{Y}^{-1} \tag{59}$$

Clustering the data

We have the following data:



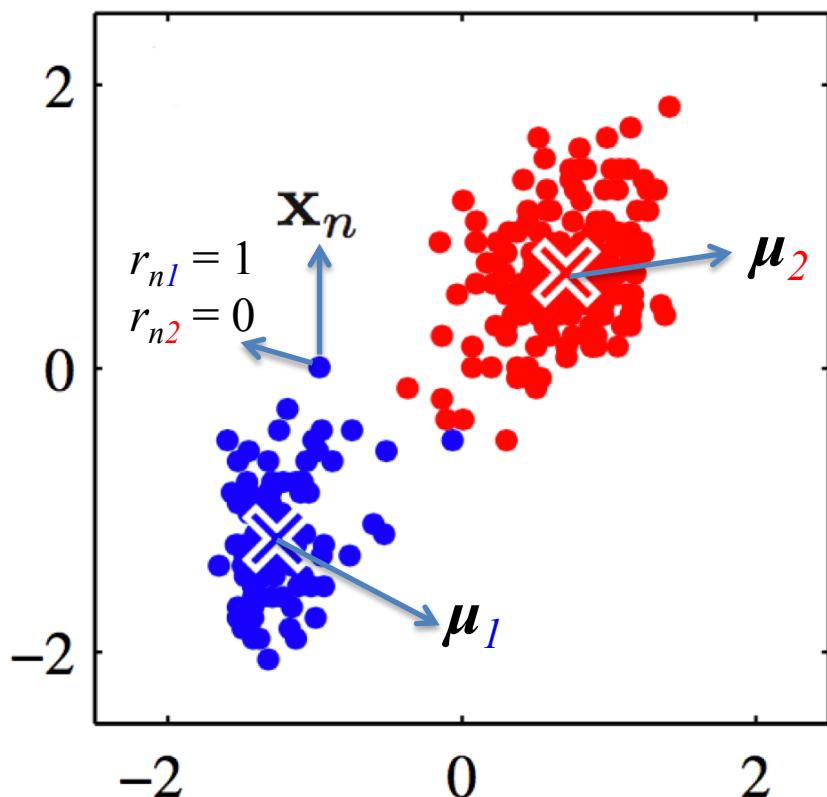
We want to cluster the data into two clusters (**red** and **blue**)



How?

Minimize the sum of square distances J

$$\text{minimize} \quad J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



$r_{nk} = 1$ if and only if data point \mathbf{x}_n is assigned to cluster k ;
otherwise $r_{nk} = 0$.

$k = 1, 2$; $K = 2$ clusters

$n = 1, \dots, N$;
 N : the total number of points.

We need to calculate $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$.

If we know r_{n1} , r_{n2} for all $n=1,\dots,N$

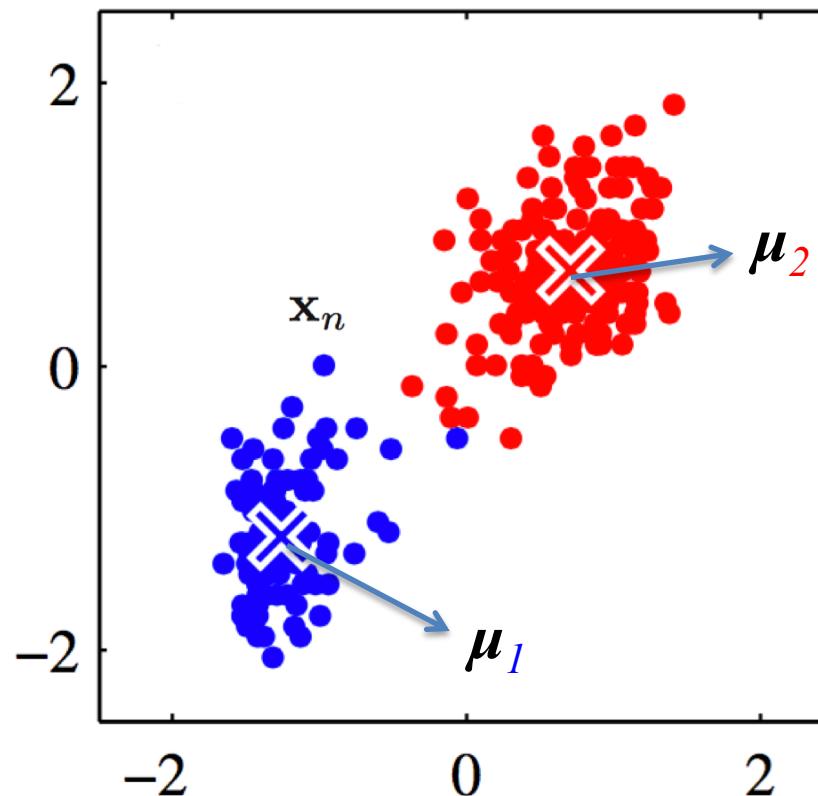
Since the points have been assigned to cluster 1 or cluster 2, we calculate

μ_1 = mean of the points in cluster 1

μ_2 = mean of the points in cluster 2

Or formally

$$\mu_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$



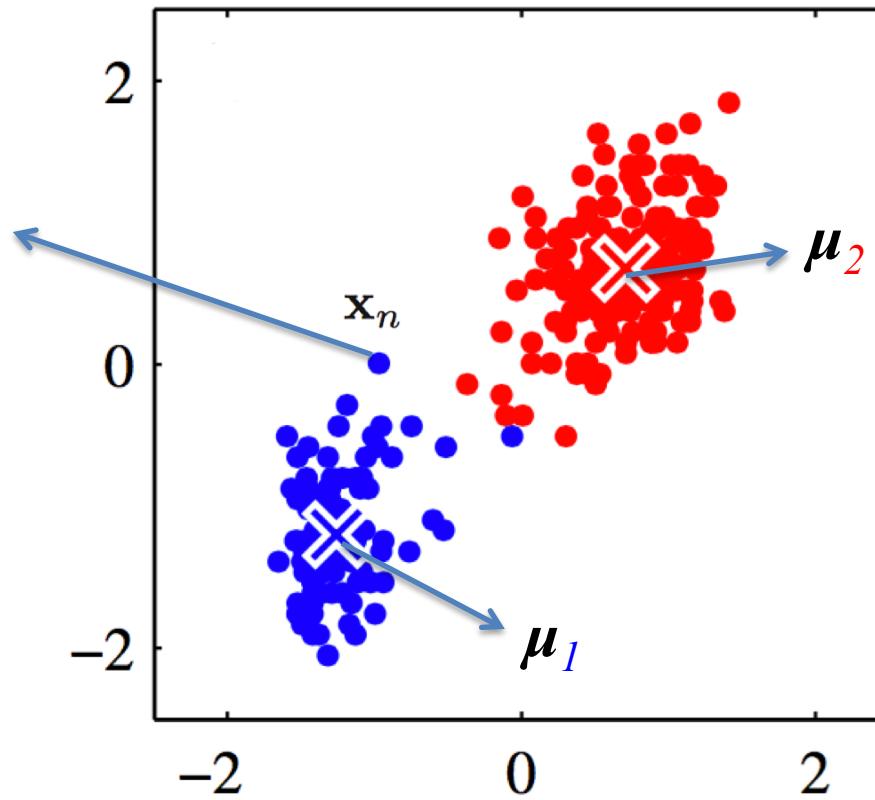
We call it the **M Step.**

If we know μ_1, μ_2

We should assign point \mathbf{x}_n to cluster 1, because

$$\|\mathbf{x}_n - \mu_1\|^2 < \|\mathbf{x}_n - \mu_2\|^2$$

Then, $r_{n1} = 1$
 $r_{n2} = 0$

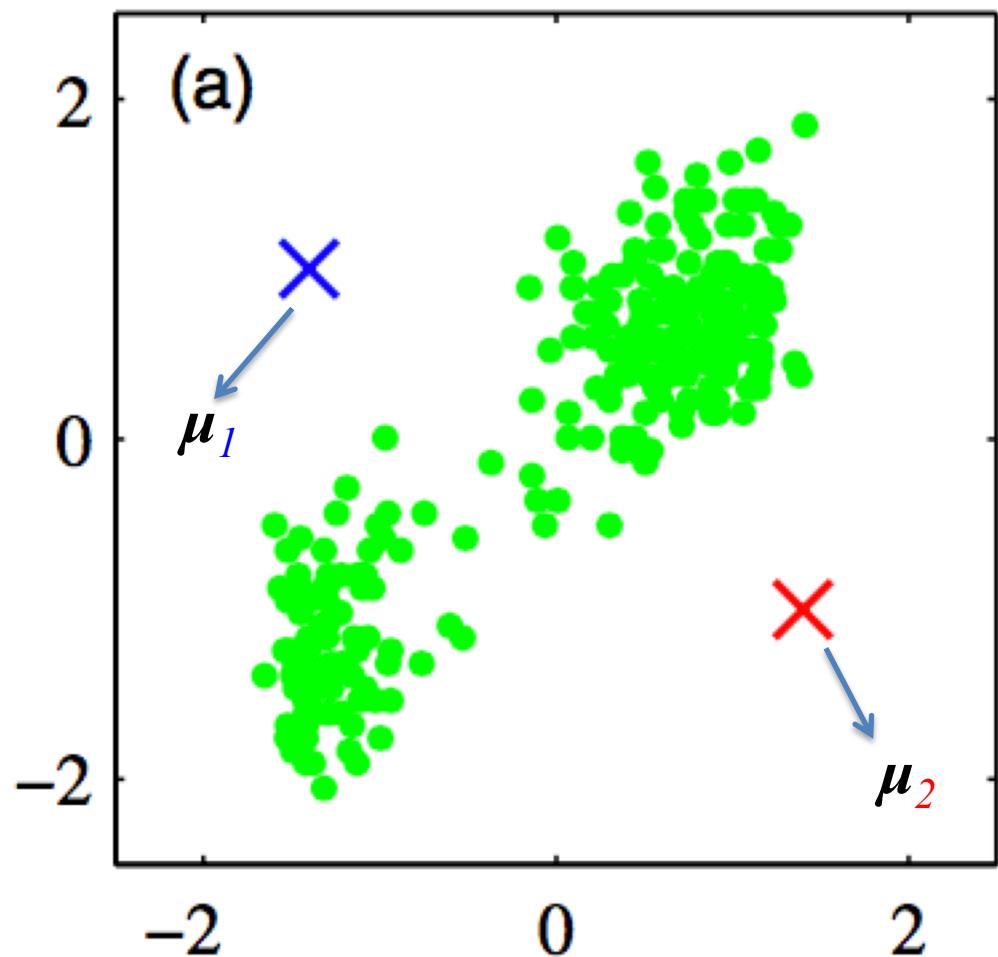


Or formally

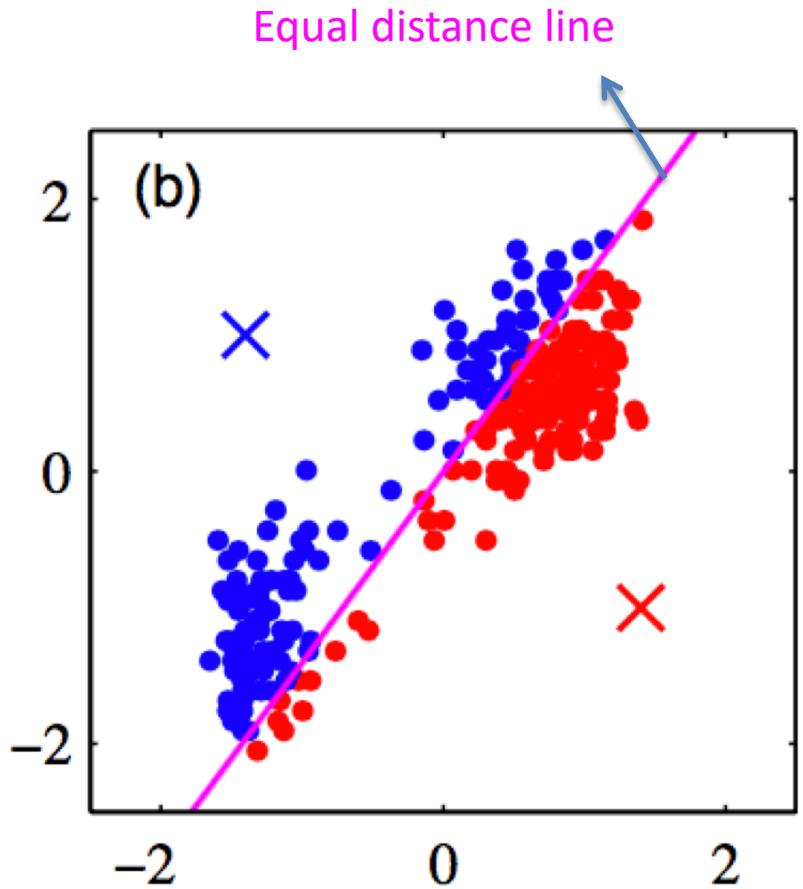
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

We call it the **E Step**

Initialization



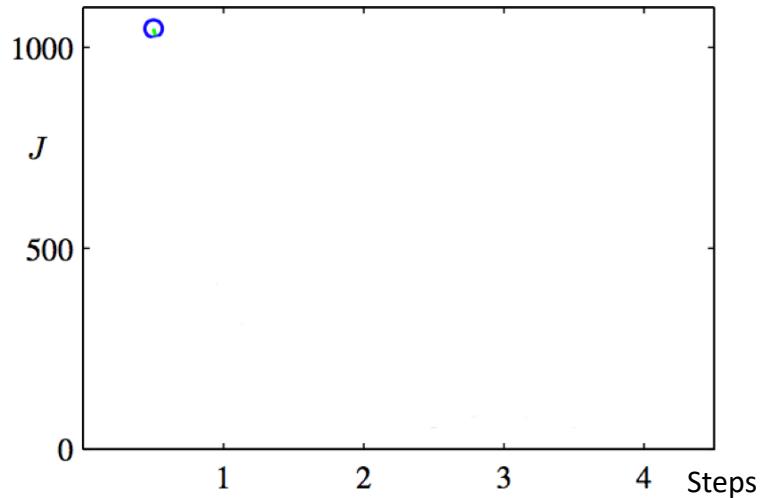
Given μ_1, μ_2 , calculate r_{n1}, r_{n2} for all
 $n=1, \dots, N$



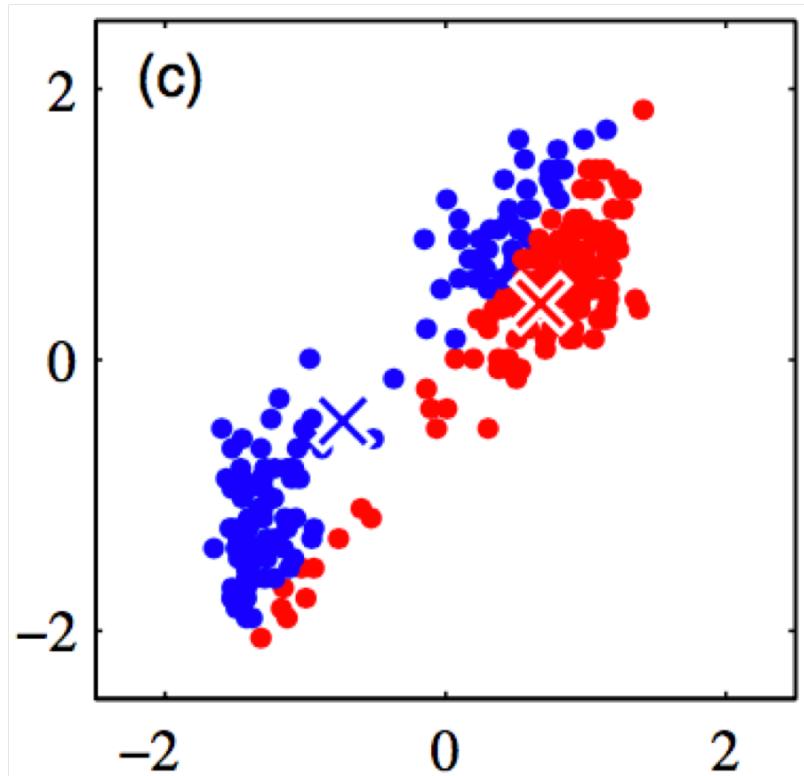
E Step

Assign the points to the nearest cluster:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$



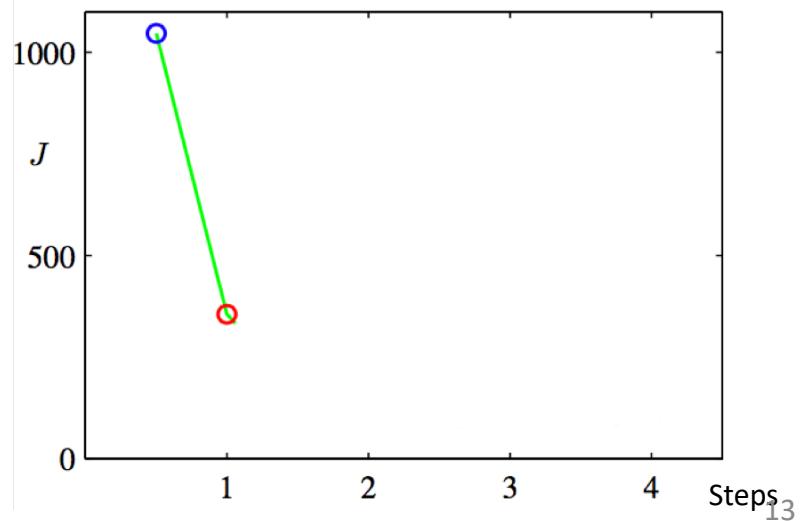
Given r_{n1} , r_{n2} , calculate μ_1 , μ_2



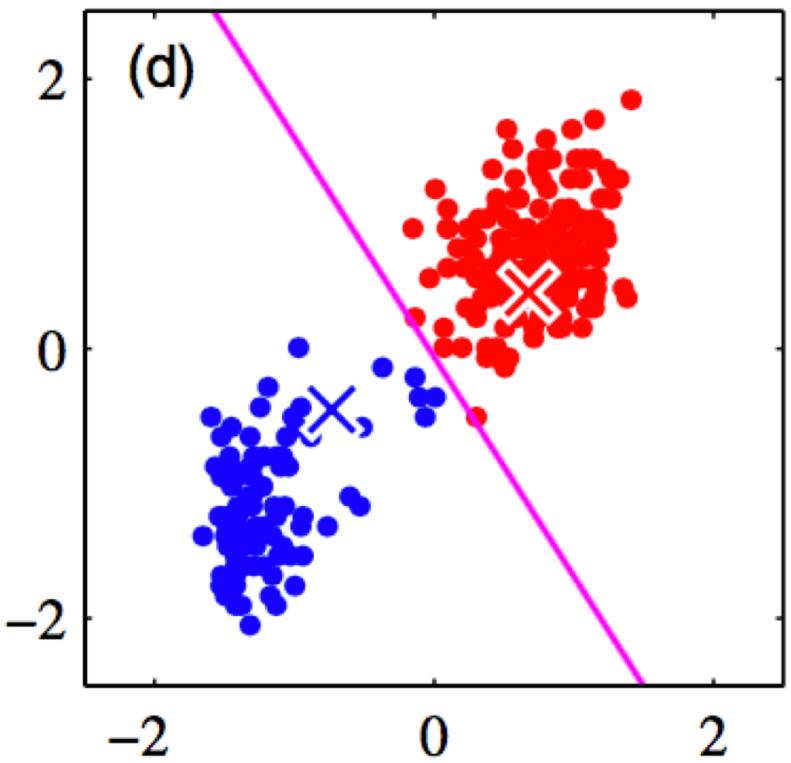
M Step

Calculate the means of the points in each cluster:

$$\mu_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$



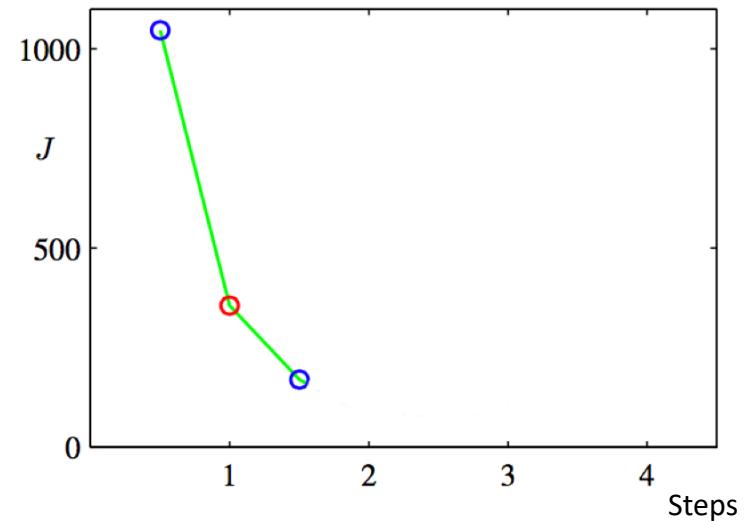
Given μ_1, μ_2 , calculate r_{n1}, r_{n2} for all
 $n=1, \dots, N$



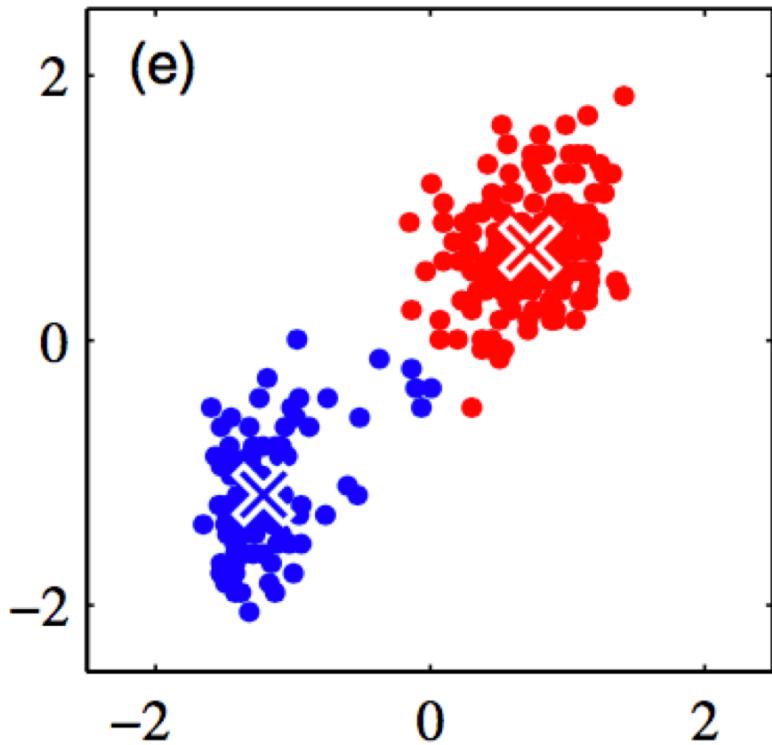
E Step

Assign the points to the nearest cluster:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \| \mathbf{x}_n - \boldsymbol{\mu}_j \|^2 \\ 0 & \text{otherwise.} \end{cases}$$



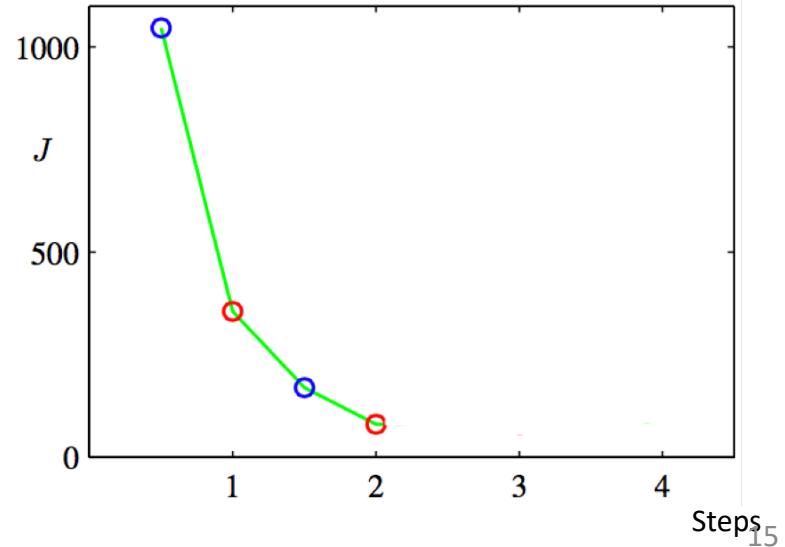
Given r_{n1} , r_{n2} , calculate μ_1 , μ_2



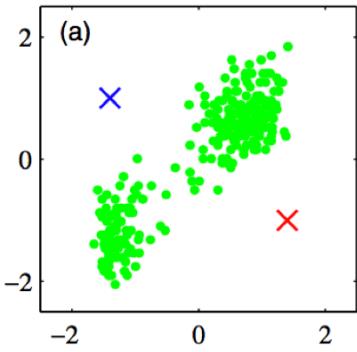
M Step

Calculate the means of the points in each cluster:

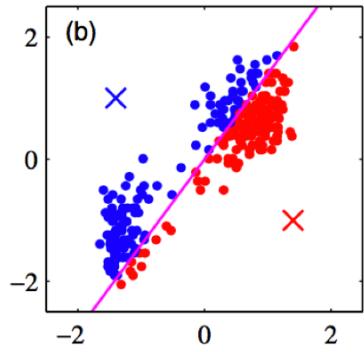
$$\mu_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$



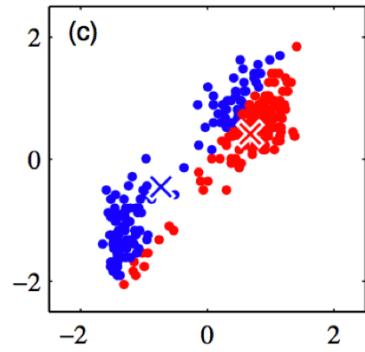
Initialization



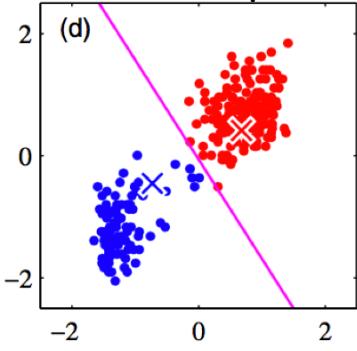
E-Step



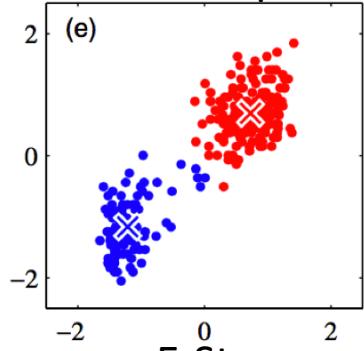
M-Step



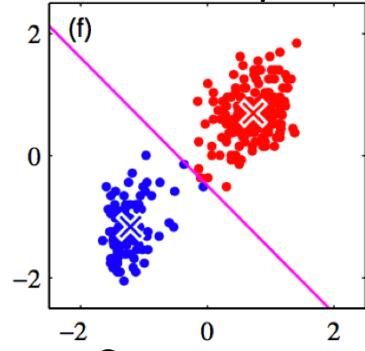
E-Step



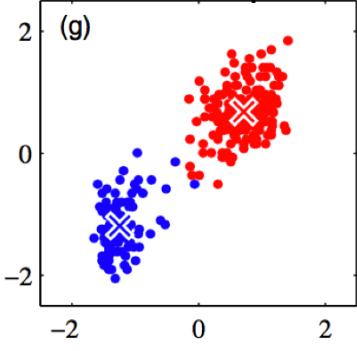
M-Step



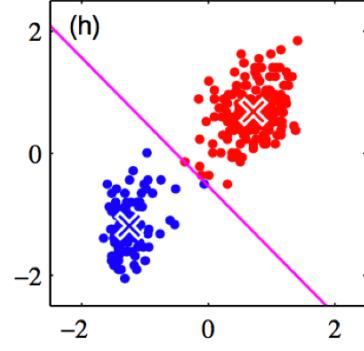
E-Step



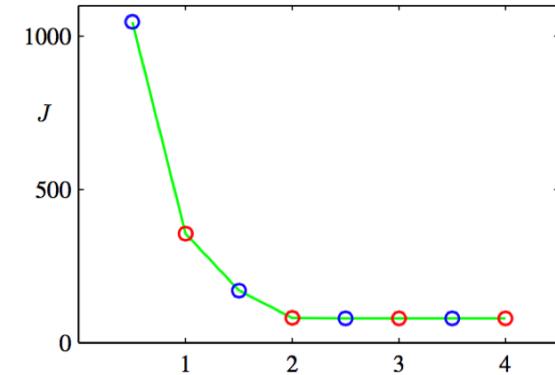
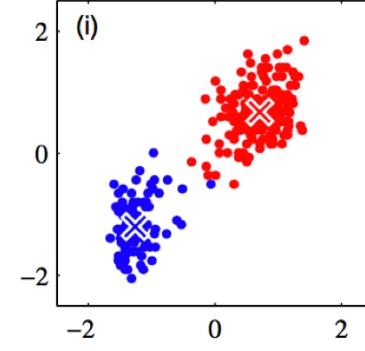
M-Step



E-Step



Convergence



If J does not change, or $\{\mu_1, \mu_2\}$ do not change, then the algorithm converges.

K均值法小结

- 初始化均值点 μ_1, \dots, μ_k
- 迭代如下
 - 把每个数据点按照就近原则分配给相应的 μ_i
 - 把 μ_i 更新为所分配的数据点的均值
- 迭代停止，如果聚类分配不变

Initialize $\mathbf{m}_i, i = 1, \dots, k$, for example, to k random \mathbf{x}^t

Repeat

For all $\mathbf{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

For all $\mathbf{m}_i, i = 1, \dots, k$

$$\mathbf{m}_i \leftarrow \sum_t b_i^t \mathbf{x}^t / \sum_t b_i^t$$

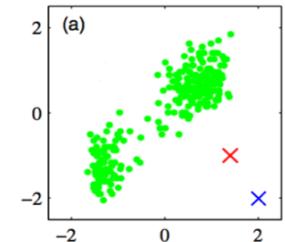
Until \mathbf{m}_i converge

Basic ingredients

- Model or structure
- Objective function
- Algorithm
- Convergence

Questions for K-mean algorithm

- Does it find the global optimum of J ?
 - No, the nearest local optimum, depending on initialization
- If Euclidean distance is not good for some data, do we have other choices?
- Can we assign each data point to the clusters probabilistically?
- If K (the total number of clusters) is unknown, can we estimate it from the data?



Outline

- Clustering
 - K-mean clustering, **hierarchical clustering**
- Adaptive learning (online learning)
 - CL, FSCL, RPCL
- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood

Hierarchical Clustering

- k -means clustering requires
 - k
 - Positions of initial centers
 - A distance measure between points (e.g. Euclidean distance)
- Hierarchical clustering requires a measure of distance between *groups* of data points

Hierarchical Clustering

- Agglomerative clustering
- A very simple procedure:
 - Assign each data point into its own group
 - Repeat: look for the two closest groups and merge them into one group
 - Stop when all the data points are merged into a single cluster

Distance Measure

- Distance between data points a and b :

- $d(a, b)$

- Group A and B

- Single-linkage

$$d(A, B) = \min_{a \in A, b \in B} d(a, b)$$

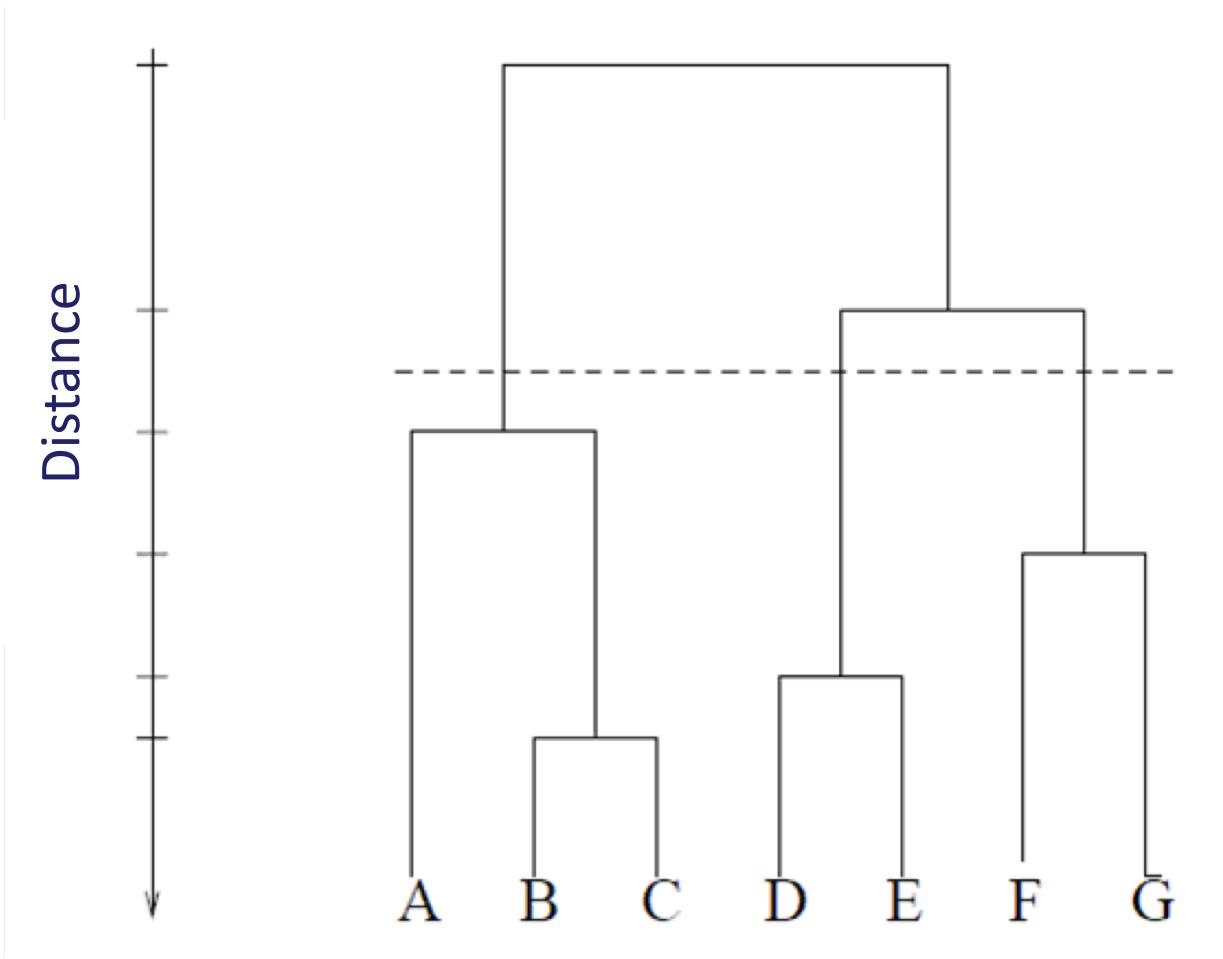
- Complete-linkage

$$d(A, B) = \max_{a \in A, b \in B} d(a, b)$$

- Average-linkage

$$d(A, B) = \frac{\sum_{a \in A, b \in B} d(a, b)}{|A| \cdot |B|}$$

Dendrogram



Outline

- Clustering
 - K-mean clustering, hierarchical clustering
- Adaptive learning (online learning)
 - CL, FSCL, RPCL
- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood

From batch to adaptive

- Given a batch of data points

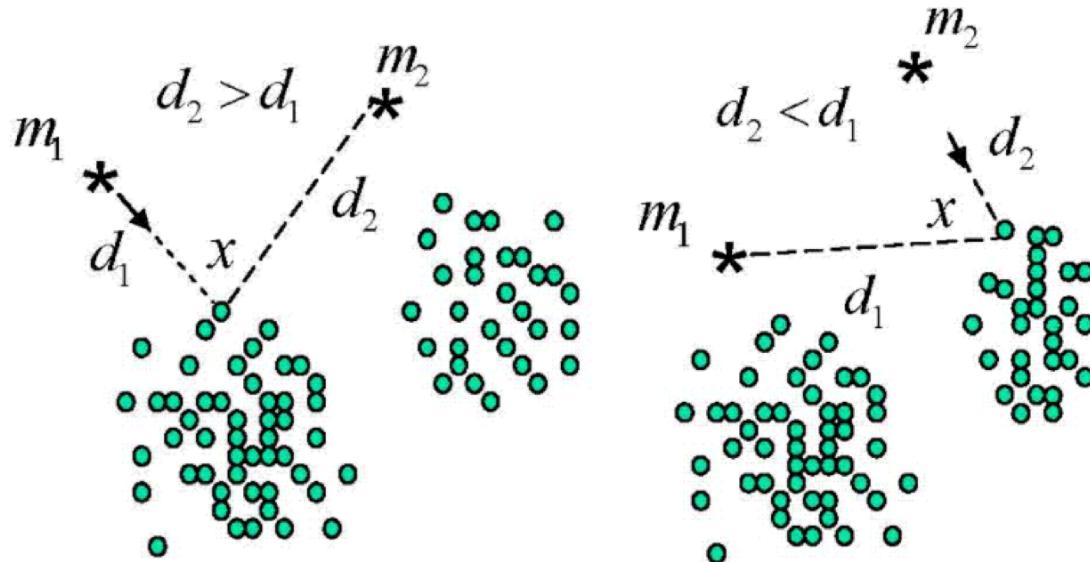


- Data points come one by one:



Competitive learning

- Data points come one by one:



(a) m_1 is the winner (b) m_2 is the winner

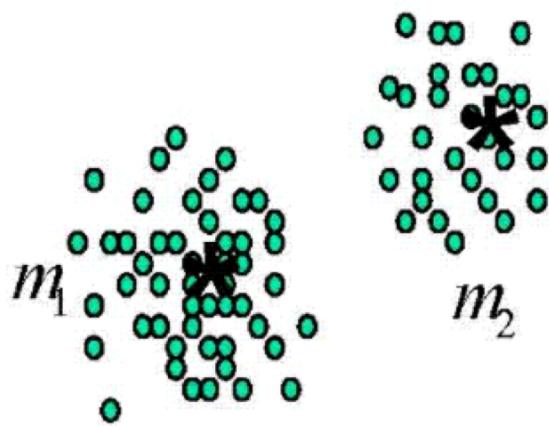
$$\varepsilon_t(\theta_j) = \|x_t - m_j\|^2$$

$$p_{j,t} = \begin{cases} 1, & \text{if } j = c, \\ 0, & \text{otherwise;} \end{cases}$$

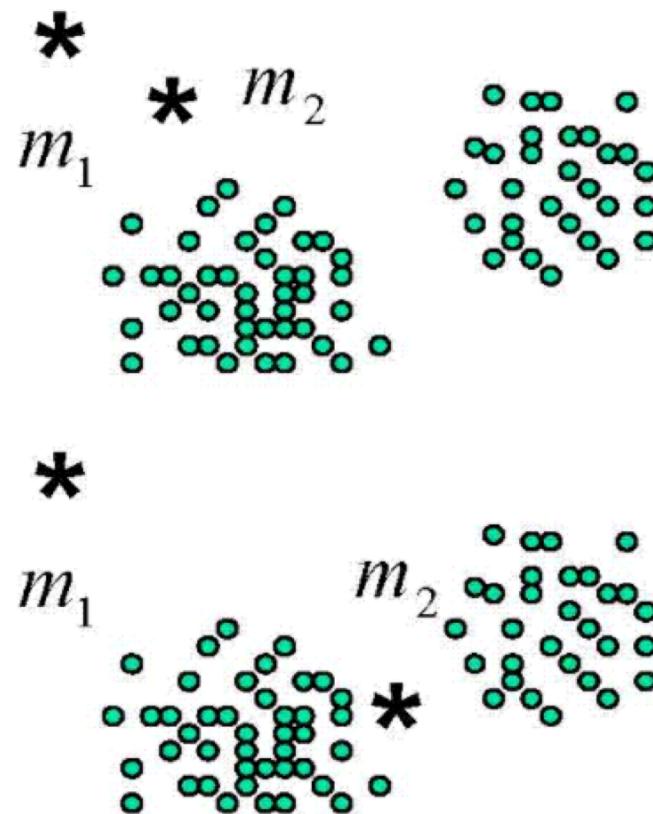
$$c = \arg \min_j \varepsilon_t(\theta_j).$$

$$m_j^{new} = m_j^{old} + \eta p_{j,t}(x_t - m_j^{old}).$$

When starting with “bad initializations”

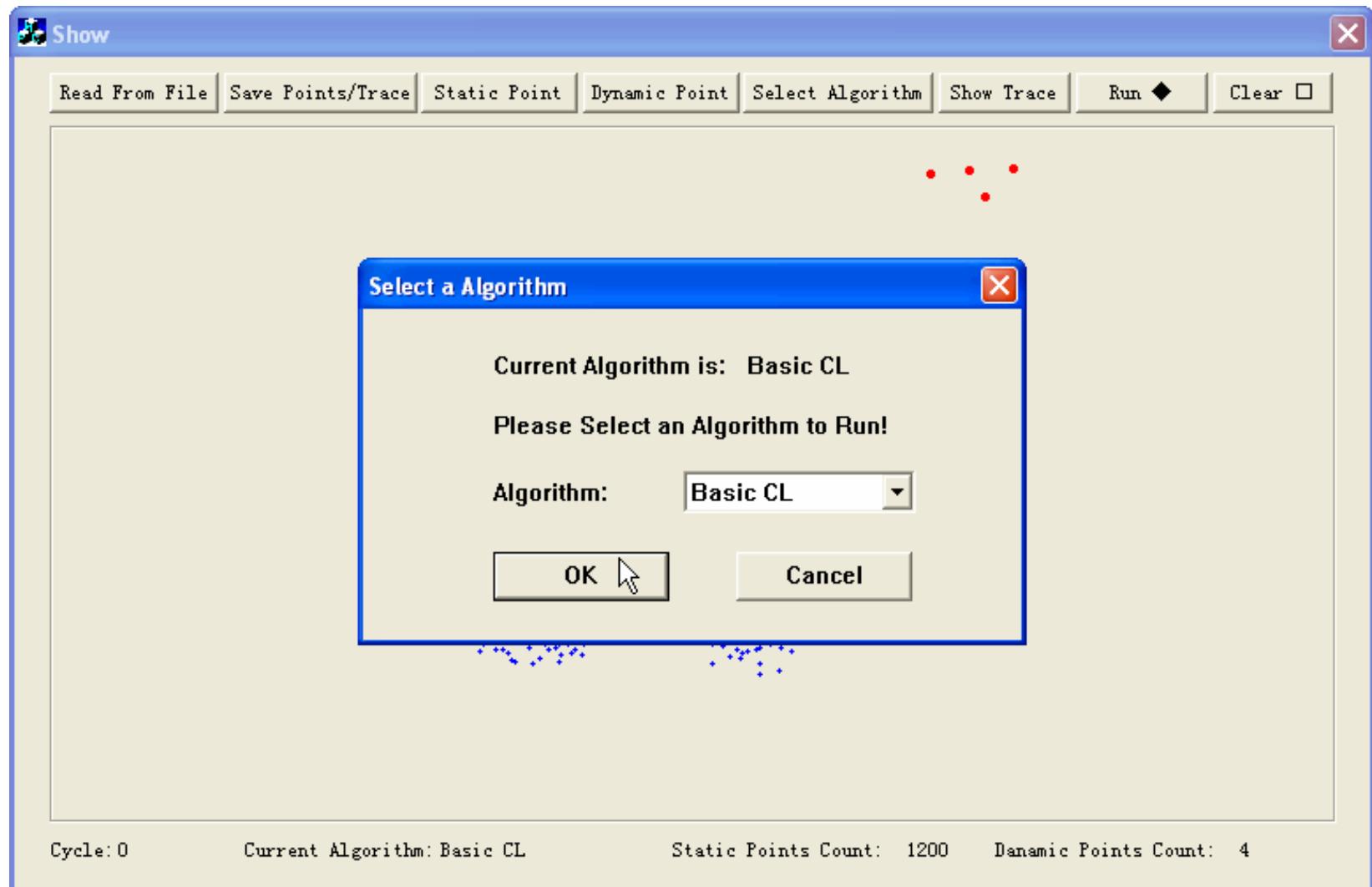


(c) converged



(d) one unit dead

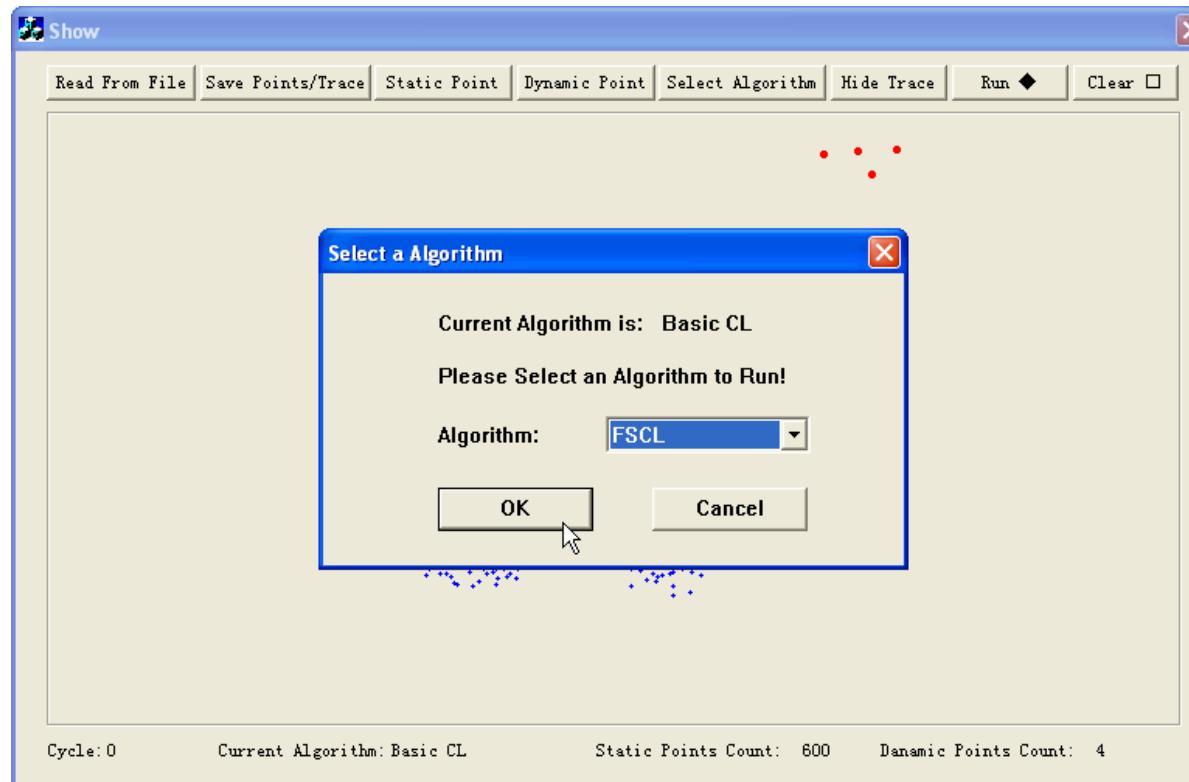
A four-cluster case



frequency sensitive competitive learning (FSCL) [Ahalt et al., 1990]

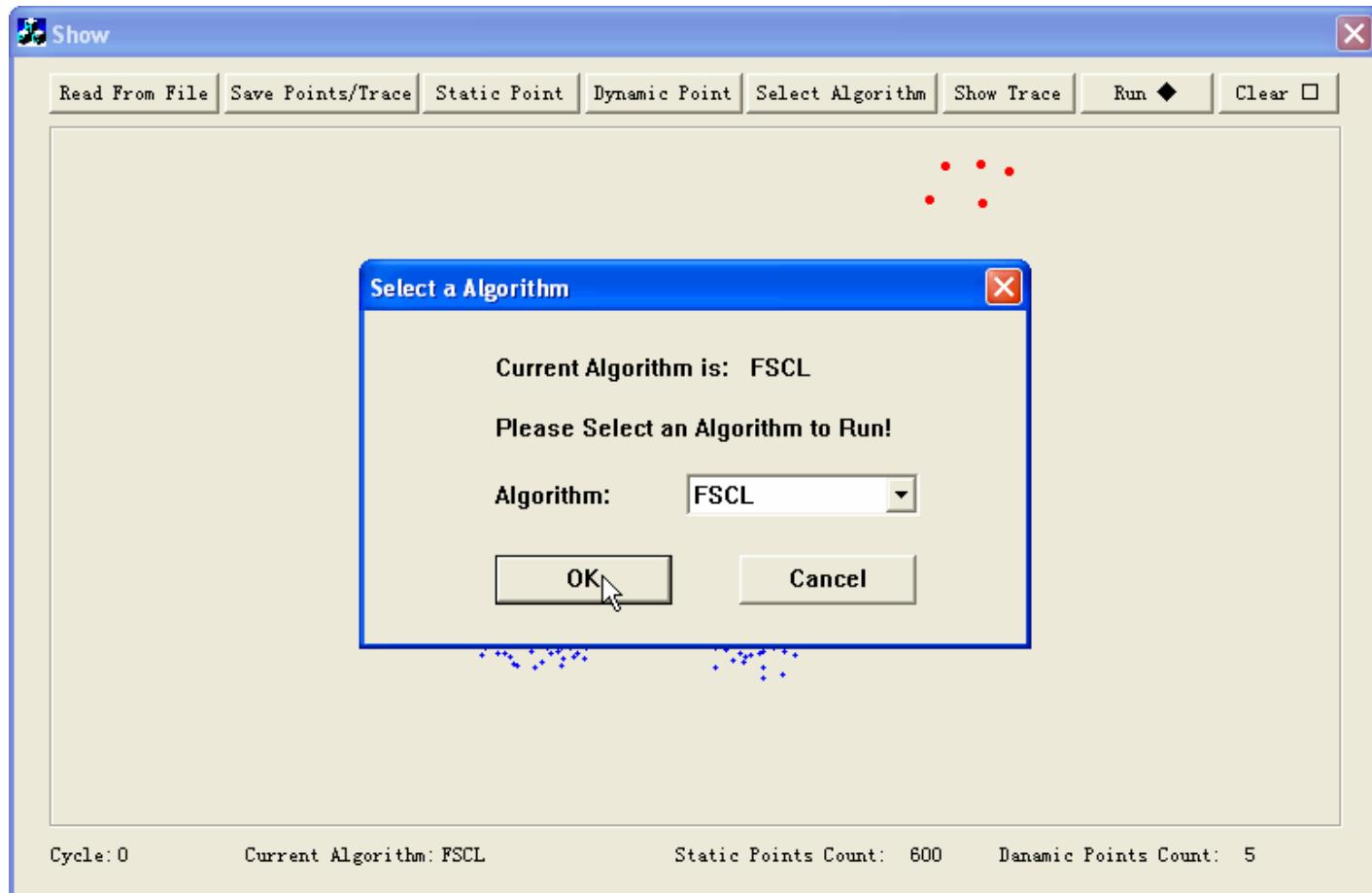
The idea is to penalize the frequent winners:

$$\varepsilon_t(\theta_j) = \alpha_j \|x_t - m_j\|^2$$



FSCL is not good when there are extra centers

When k is pre-assigned to 5. the frequency sensitive mechanism also brings the extra one into data to disturb the correct locations of others



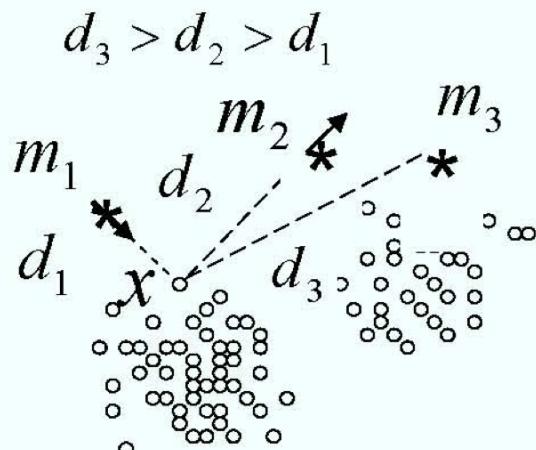
Rival penalized competitive learning (RPCL)

(Xu, Krzyzak, & Oja, 1992 , 1993)

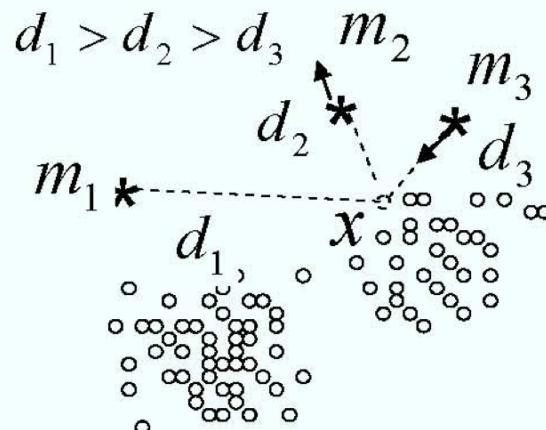
The RPCL differs from FSCL by implementing $p_{j,t}$ as follows:

$$p_{j,t} = \begin{cases} 1, & \text{if } j = c, \\ -\gamma, & \text{if } j = r, \\ 0, & \text{otherwise,} \end{cases} \quad \begin{cases} c = \arg \min_j \epsilon_t(\theta_j), \\ r = \arg \min_{j \neq c} \epsilon_t(\theta_j), \end{cases}$$

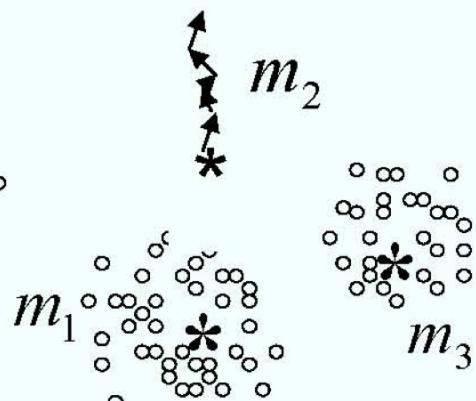
where γ approximately takes a number between 0.05 and 0.1 for controlling the penalizing strength.



(a) m_1 is the winner
 m_2 is the rival

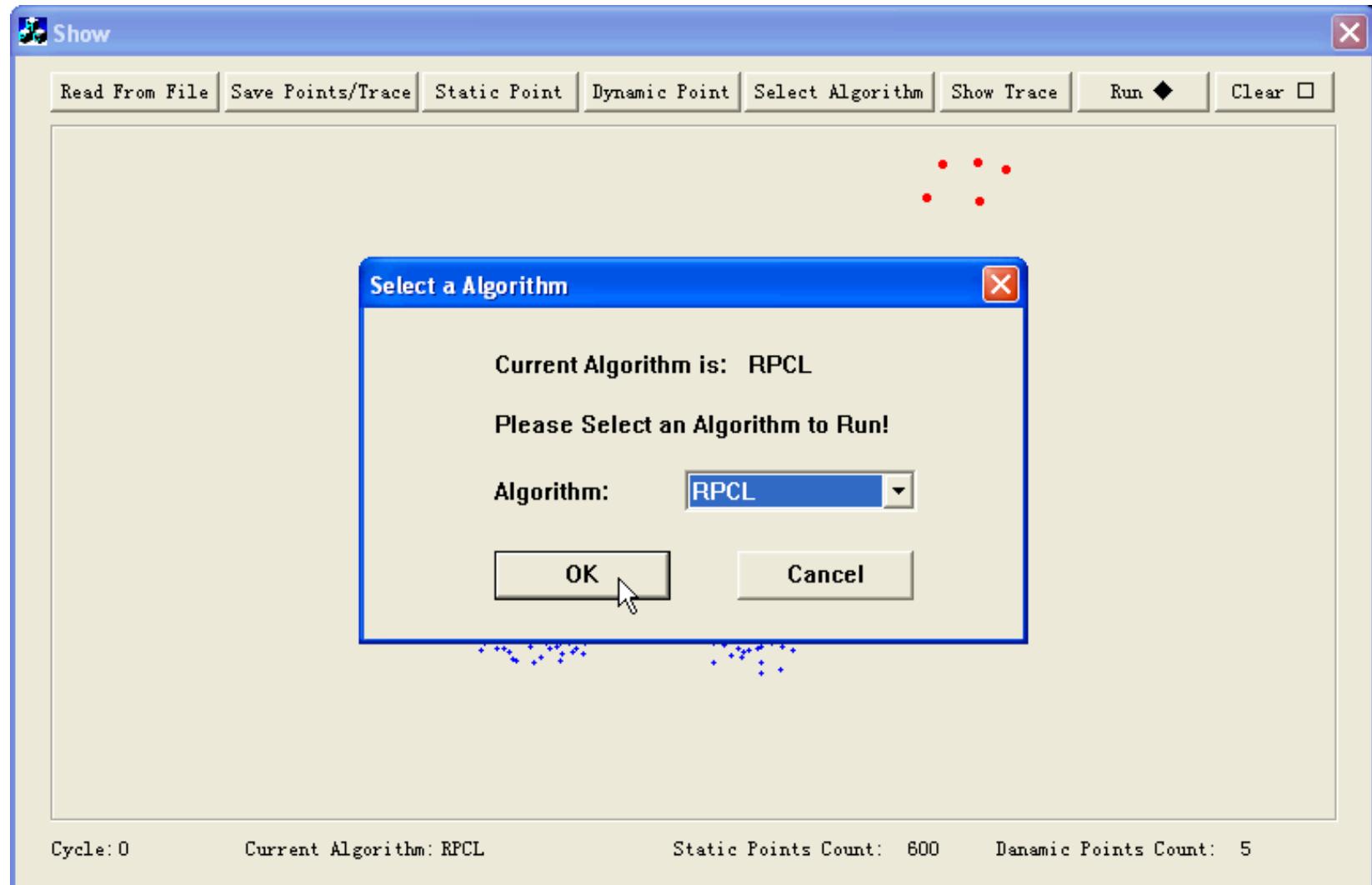


(b) m_3 is the winner
 m_2 is the rival



(c) m_1 and m_3 are converged
 m_2 is driven far away

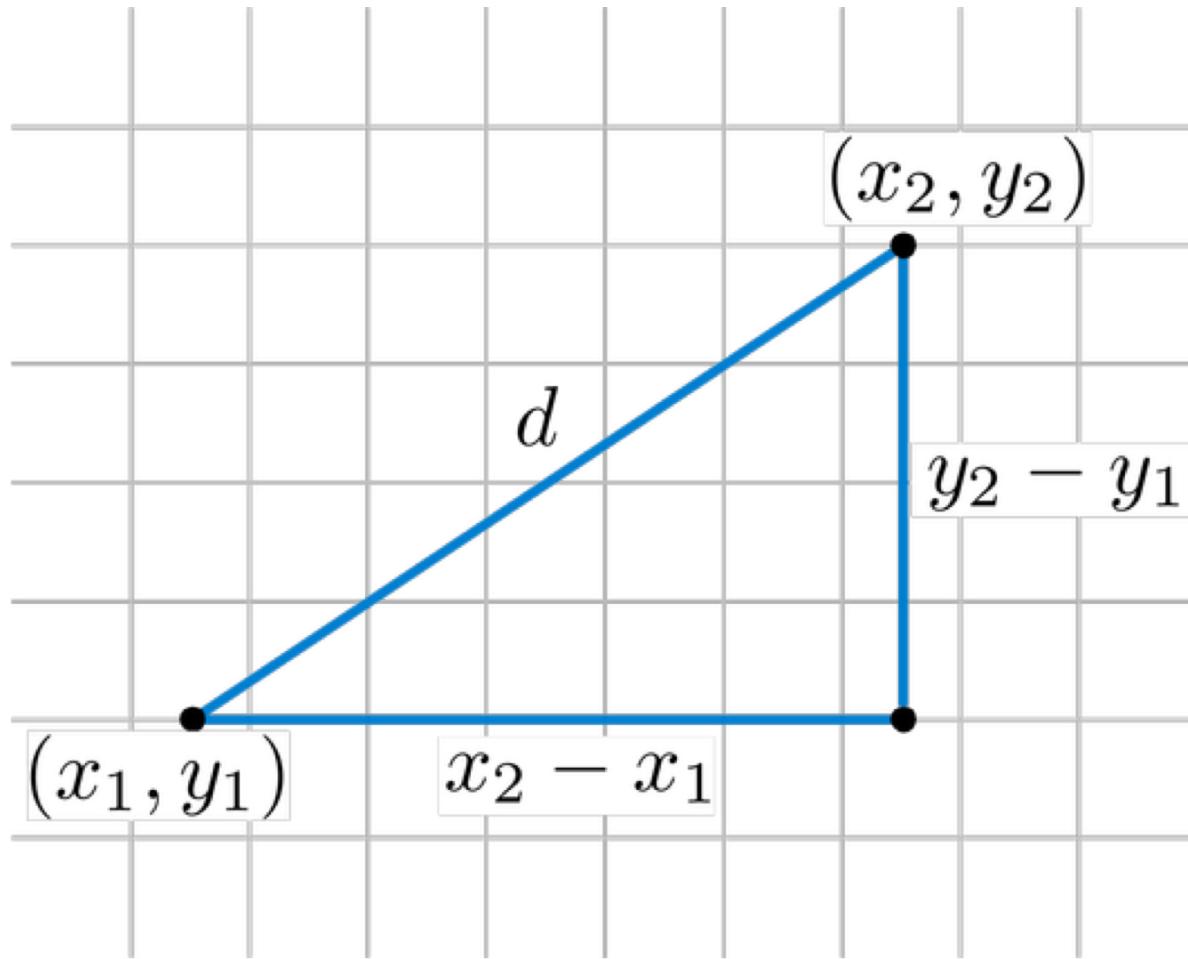
Rival penalized mechanism makes extra agents driven far away.



Outline

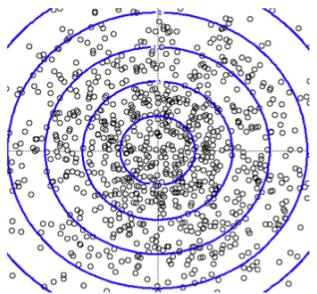
- Clustering
 - K-mean clustering, hierarchical clustering
- Adaptive learning (online learning)
 - CL, FSCL, RPCL
- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood

Euclidean Distance

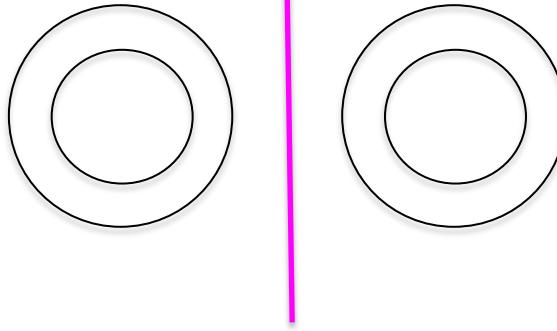


Euclidian distance may not be a good measure for some data

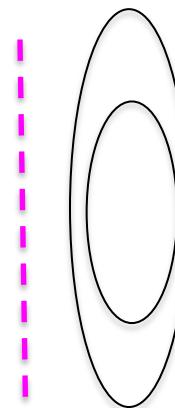
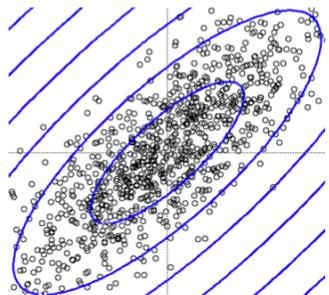
Euclidean distance



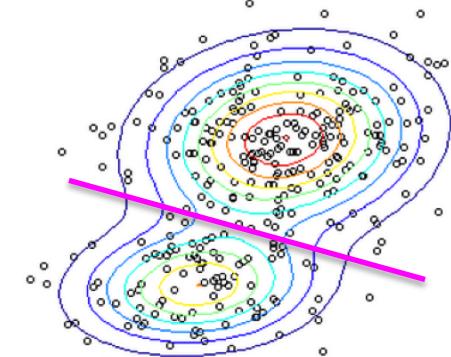
Equal distance line



Mahalanobis distance



In general



Distances at different directions could be different!

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}. \quad \Sigma \text{ is the covariance matrix}$$

More Distance Measures

Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c e_{fc} - e_{gc} $
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (e_f - e_g)^\top \Sigma^{-1} (e_f - e_g)$, where Σ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2} \sqrt{\sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2} \sqrt{\sum_c e_{gc}^2}}$
Spellman rank correlation	As Pearson correlation, but replace e_{gc} with the rank of e_{gc} within the expression values of gene g across all conditions $c = 1 \dots C$
Absolute or squared correlation	$d_{fg} = 1 - r_{fg} $ or $d_{fg} = 1 - r_{fg}^2$
<hr/> d_{fg} , distance between expression patterns for genes f and g . e_{gc} , expression level of gene g under condition c .	

From distance to probability

distance

$$\|x - \mu\|^2$$

“The closer, the more likely.”

likely

$$\exp\{-\lambda \|x - \mu\|^2\}$$

Sum or integral to
be one

Probability

It is more powerful to consider everything in probability framework!

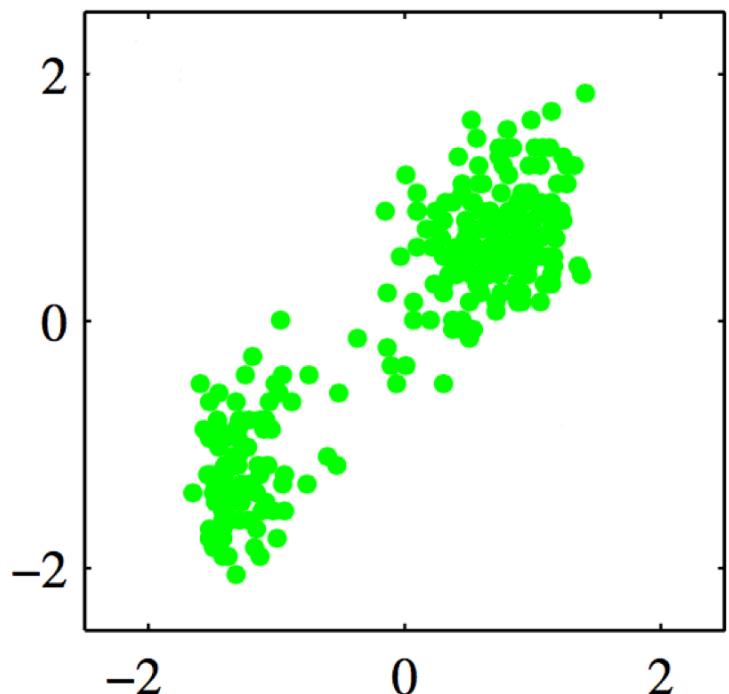
$$\frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

Gaussian distribution with the Mahalanobis distance

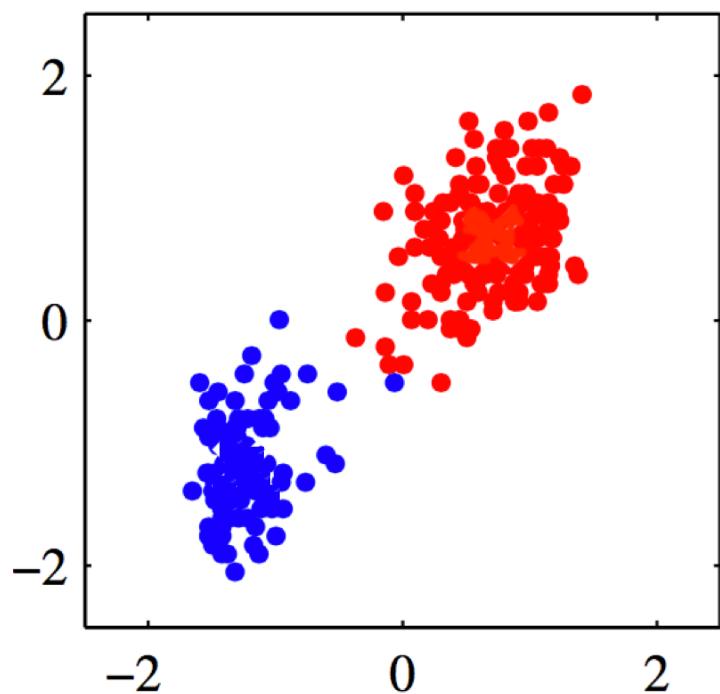
$$D_M(x) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}.$$

Review the clustering problem again

We have the following data:

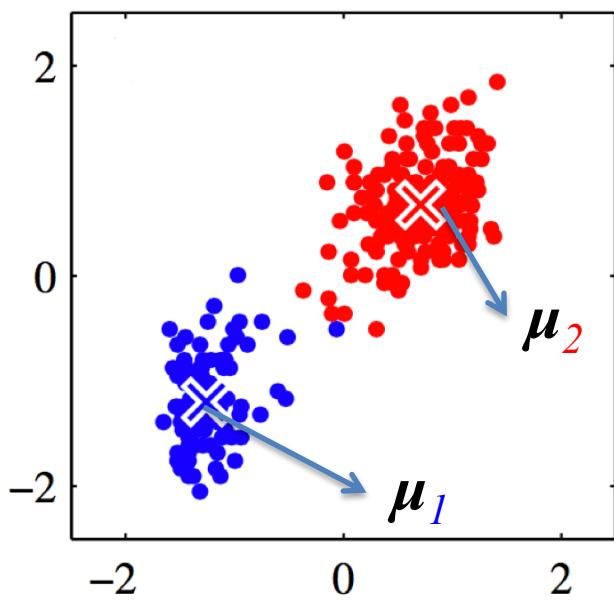


We want to cluster the data into two clusters (**red** and **blue**)

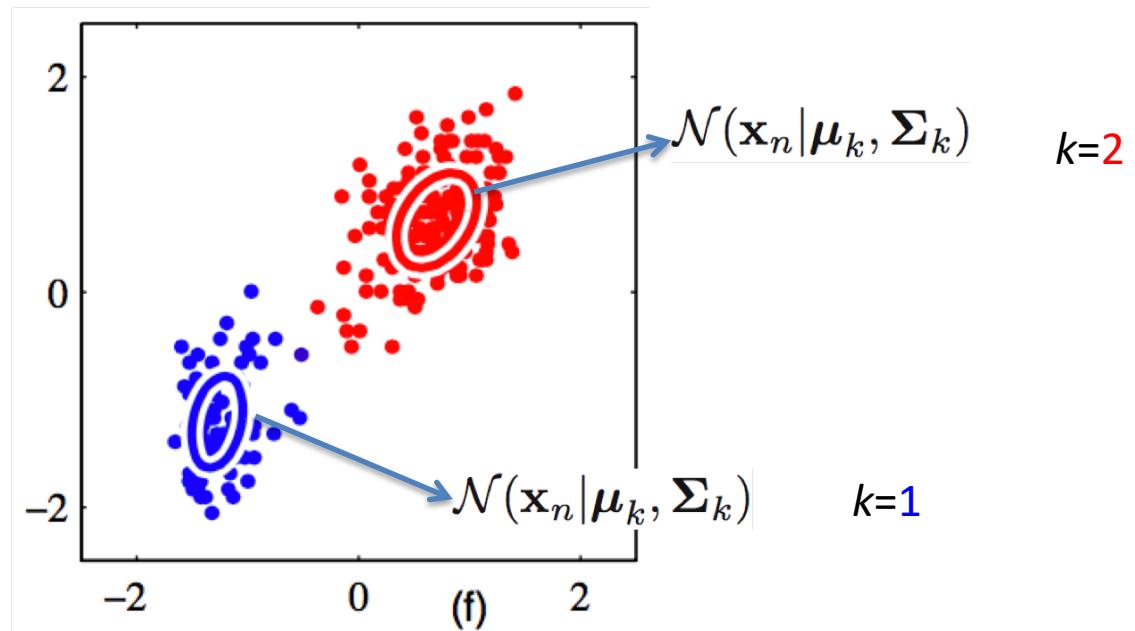


Instead if using $\{\mu_1, \mu_2\}$, each cluster is represented as a Gaussian distribution

K-means

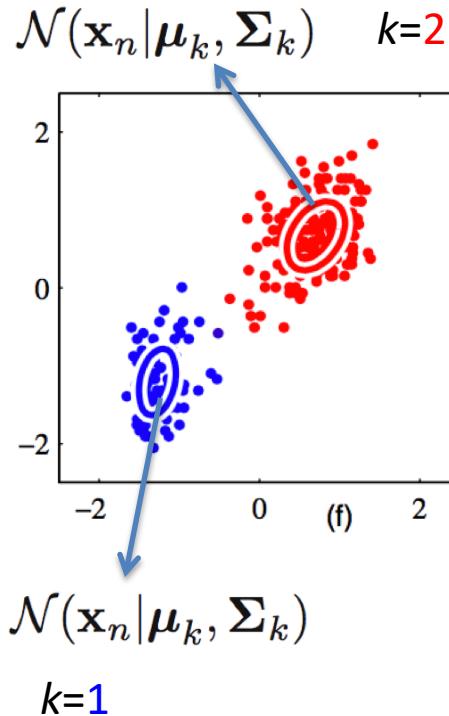


Gaussian Mixture Model (GMM)



$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right)$$

Gaussian Mixture Model (GMM)



We use $z_k = 1$ to indicate a point \mathbf{x} belongs to cluster k

$$\mathbf{z} = (z_1, \dots, z_K) \quad z_k \in \{0, 1\} \quad \sum_k z_k = 1$$

Assume the points in the same cluster follow a **Gaussian distribution**

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

A mixing weight for each cluster:

$$p(z_k = 1) = \pi_k \quad 0 \leq \pi_k \leq 1 \quad \sum_{k=1}^K \pi_k = 1$$

prior probability of point belonging to a cluster

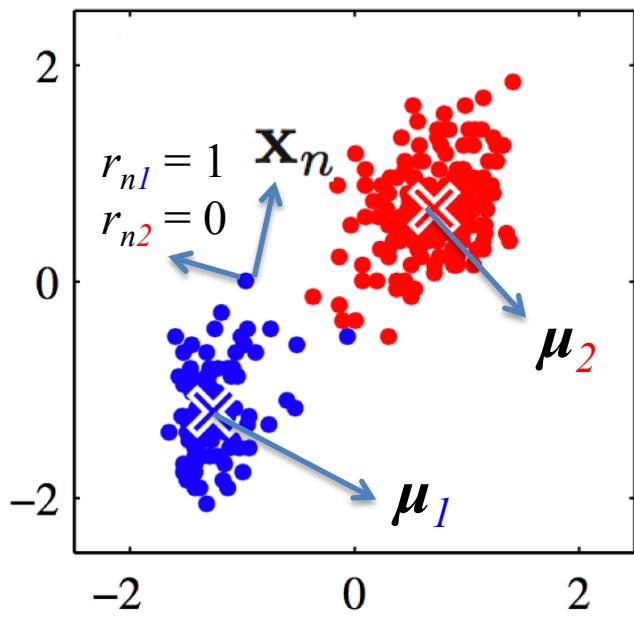
So, we get a distribution for the data point \mathbf{x} :

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

From minimizing sum of square distances to finding maximum likelihood

minimize

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



maximize likelihood

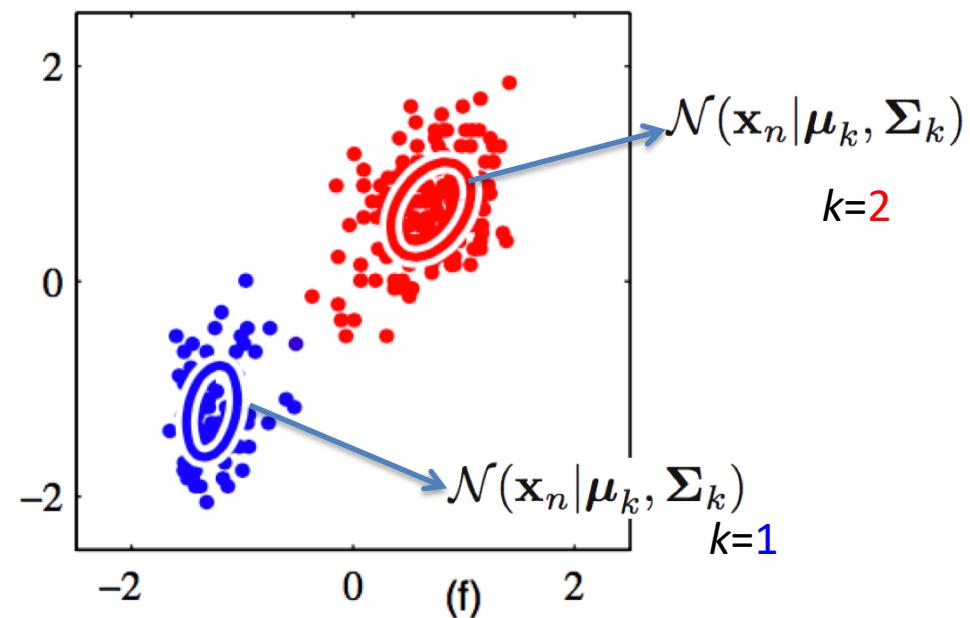
$$p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$$

$$X = \{x_1, \dots, x_N\}$$

$$\boldsymbol{\pi} = \{\pi_1, \dots, \pi_K\}$$

$$\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$$

$$\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\}$$



Remember: **The closer the distance, the more likely the probability.**

Maximum likelihood

Given a data set $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$ in which the observations $\{\mathbf{x}_n\}$ are assumed to be drawn independently from a multivariate Gaussian distribution, we can estimate the parameters of the distribution by maximum likelihood. The log likelihood function is given by

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu})$$

Maximizing the log-likelihood function:

$$\frac{\partial}{\partial \boldsymbol{\mu}} \ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}) = 0 \quad \longrightarrow \quad \boldsymbol{\mu}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$$

$$\text{Similarly we get} \quad \boldsymbol{\Sigma}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_{\text{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\text{ML}})^T$$

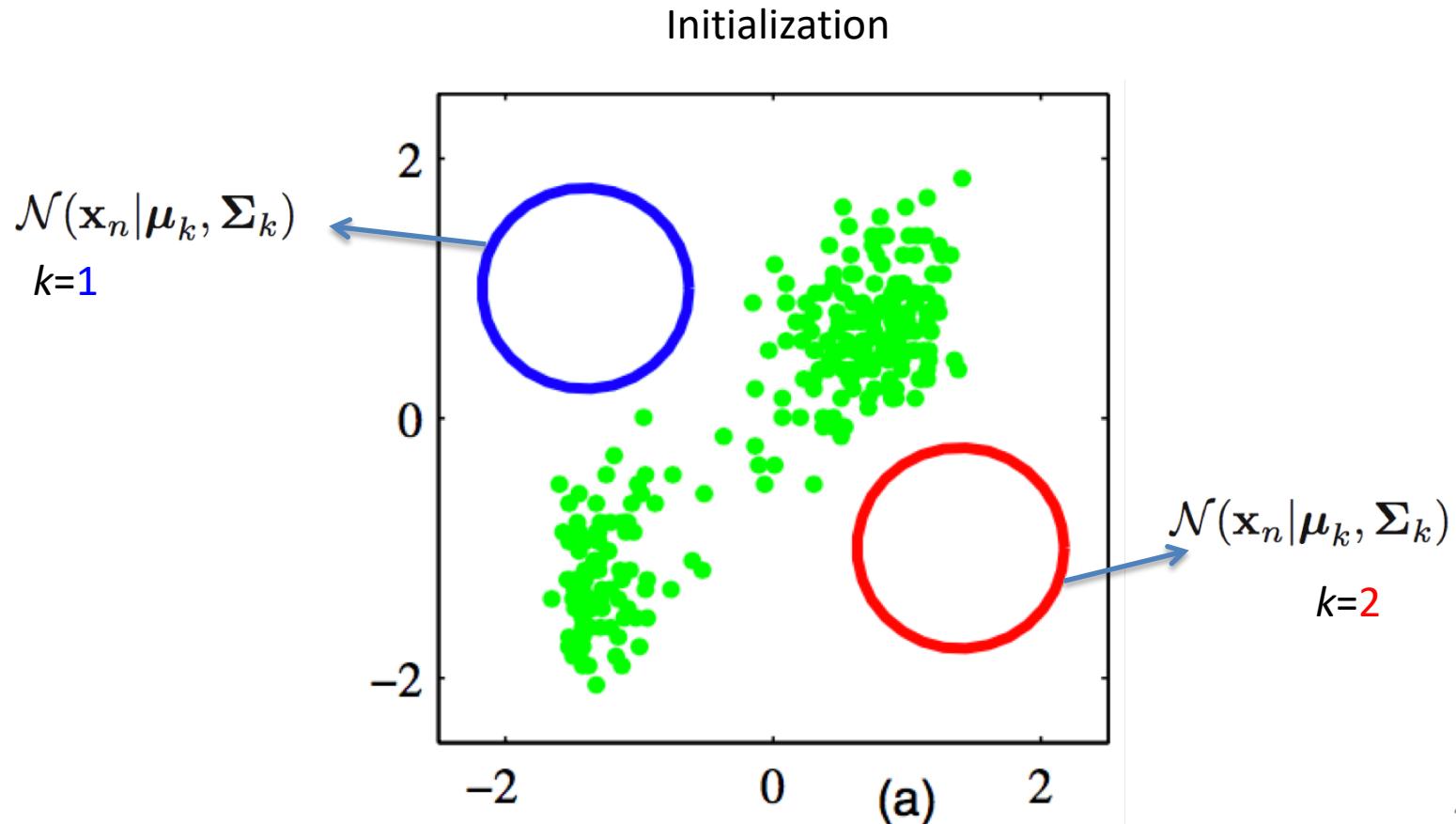
$\boldsymbol{\mu}_{\text{ML}}$ and $\boldsymbol{\Sigma}_{\text{ML}}$ are the maximum likelihood estimates of the mean and the co-variance matrix.

Outline

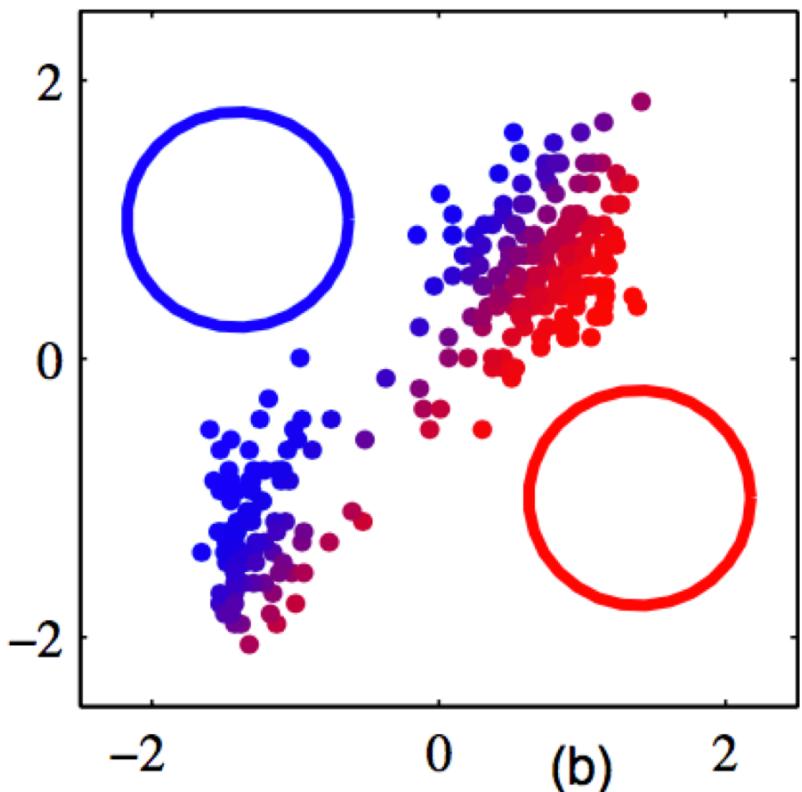
- Clustering
 - K-mean clustering, hierarchical clustering
- Adaptive learning (online learning)
 - CL, FSCL, RPCL
- Gaussian Mixture Models (GMM)
- Expectation-Maximization (EM) for maximum likelihood

Expectation-Maximization (EM) algorithm for maximum likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$



E Step



When the parameters are given, the assignments of the points can be calculated by the posterior probability, i.e., the probability of a data point belonging to a cluster once we have observed the data point.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

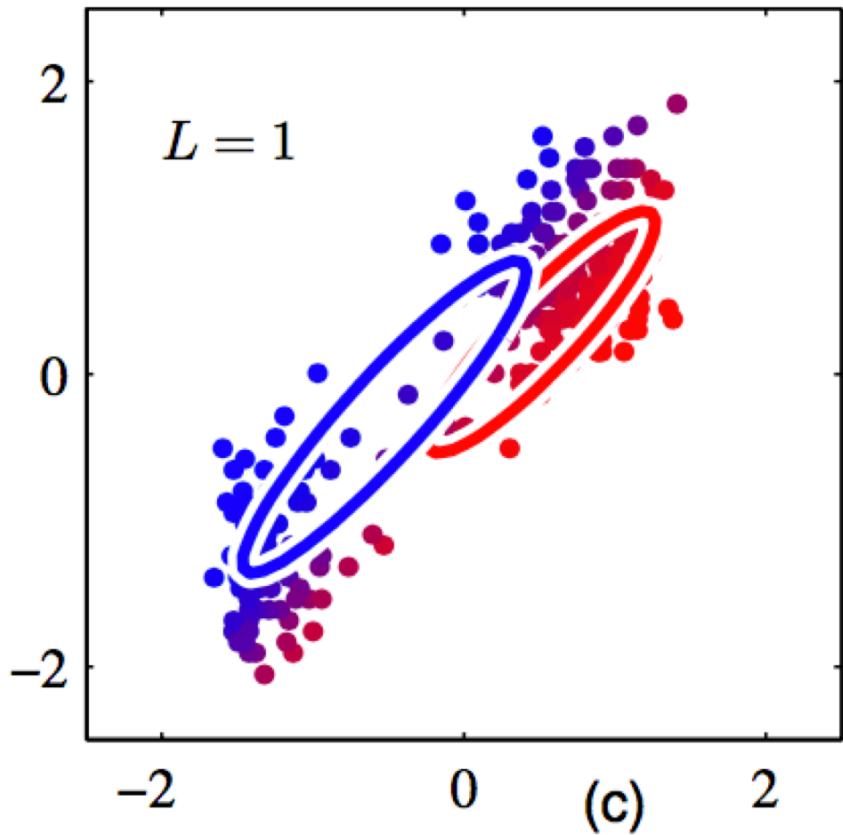
Soft assignment:
A point fractionally belongs to two clusters.

For example,

0.2 belong to cluster 1

0.8 belong to cluster 2

M Step



When the assignments $\gamma(z_{nk})$ of the points to the clusters are known, parameters could be calculated for each cluster (Gaussian) separately.

Mixing weight π_k : the proportion of number of points in cluster k within all data points

$$\pi_k = \frac{N_k}{N} ; \quad N_k = \sum_{n=1}^N \gamma(z_{nk}).$$

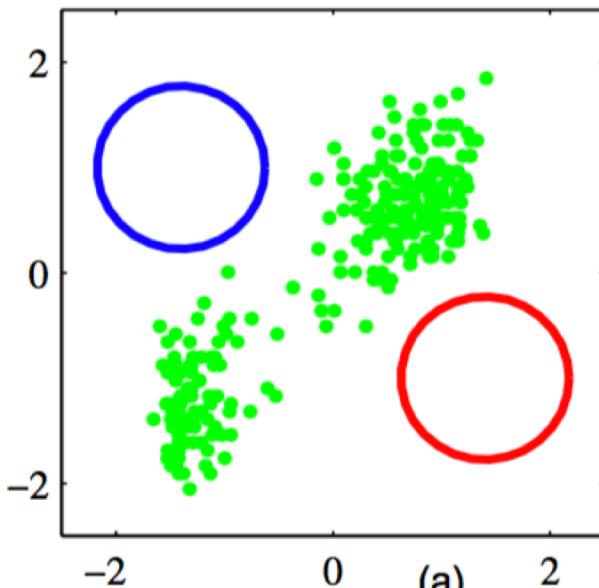
μ_k, Σ_k : the mean and the covariance matrix are calculated for each cluster

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

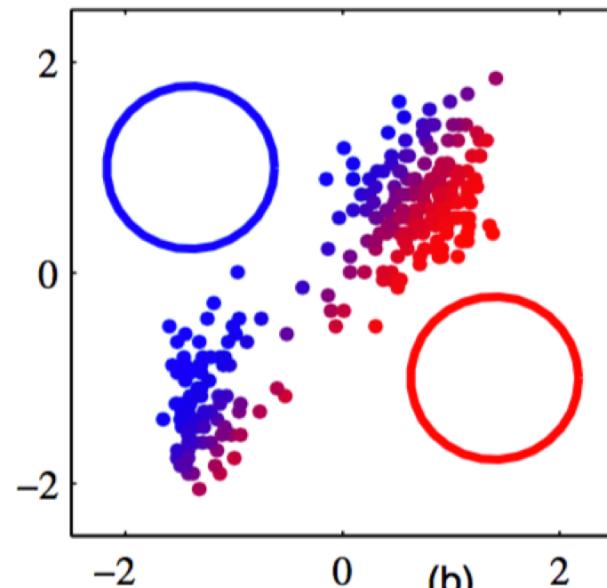
$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T$$

L denotes the number of cycles of the EM algorithm.

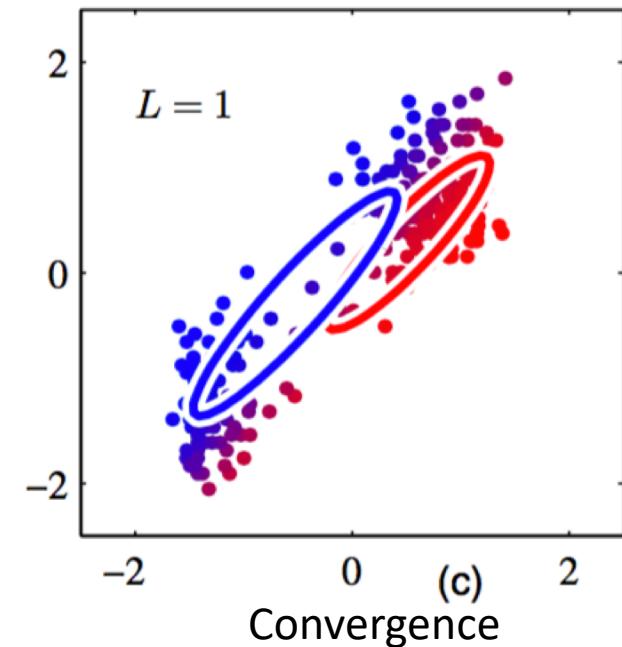
initialization



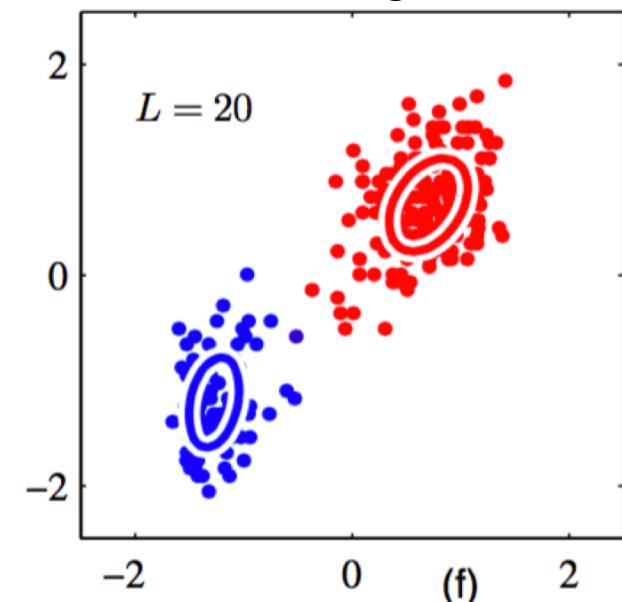
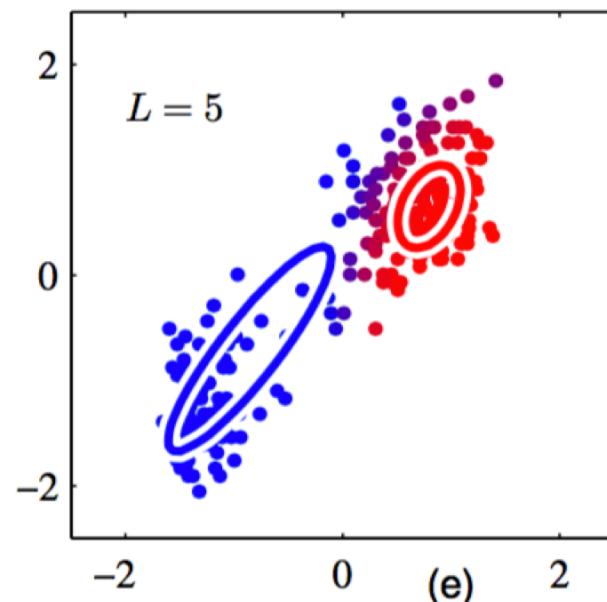
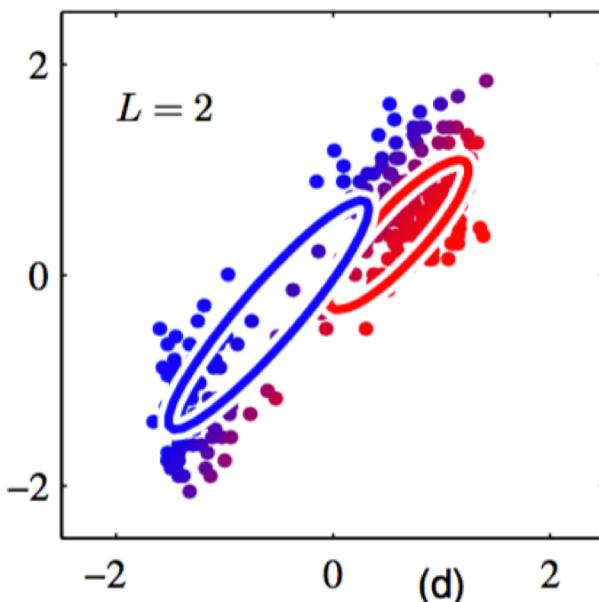
E-Step



M-Step



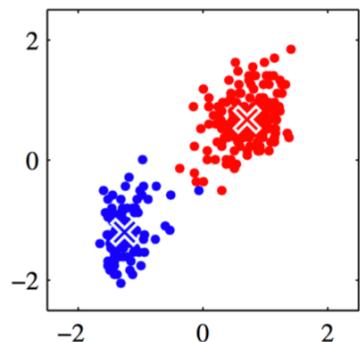
Convergence



L denotes the number of cycles of E-Step and M-Step.

Relation to K-means

$$\|\mathbf{x} - \boldsymbol{\mu}_k\|^2$$

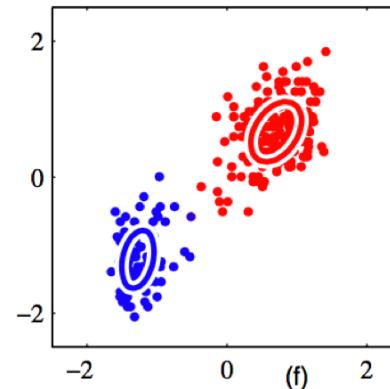


$$\{\boldsymbol{\mu}_k\}$$

One-in-K assignment

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

$$\boldsymbol{\Sigma}_k = \epsilon \mathbf{I}$$



GMM considers covariance and mixing weights.

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp \left\{ -\frac{1}{2\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \right\}$$

Soft assignment

$$\gamma(z_{nk}) = \frac{\pi_k \exp \left\{ -\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 / 2\epsilon \right\}}{\sum_j \pi_j \exp \left\{ -\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 / 2\epsilon \right\}}$$

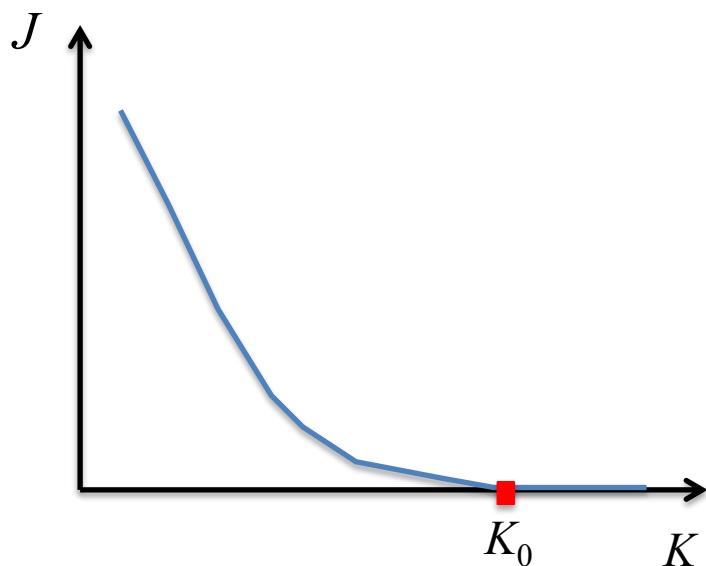
Summary for the EM algorithm for GMM

- Does it find the global optimum?
 - No, like K-means, EM only finds the nearest local optimum and the optimum depends on the initialization
- GMM is more general than K-means by considering mixing weights, covariance matrices, and soft assignments.
- Like K-means, it does not tell you the best K.

How to determine the cluster number K?

K-mean

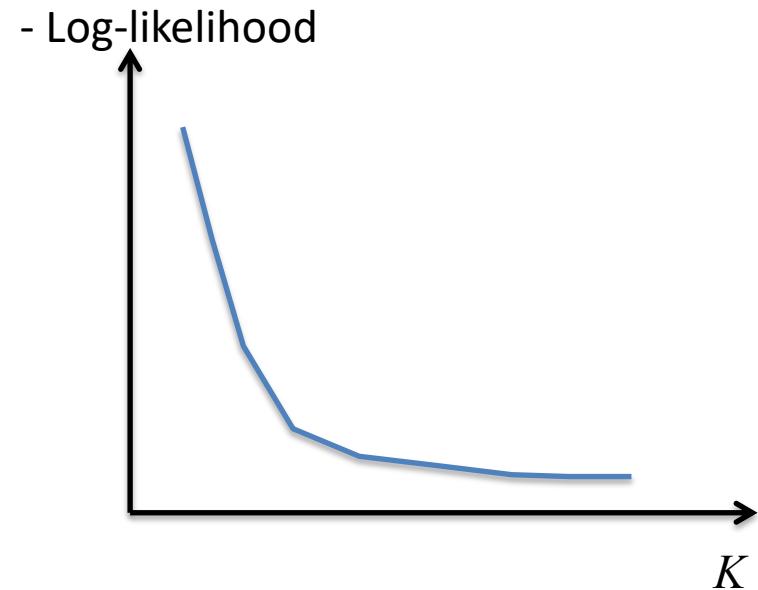
$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



J does not tell which K is better.

GMM

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$



Negative log-likelihood also decreases as K increases.

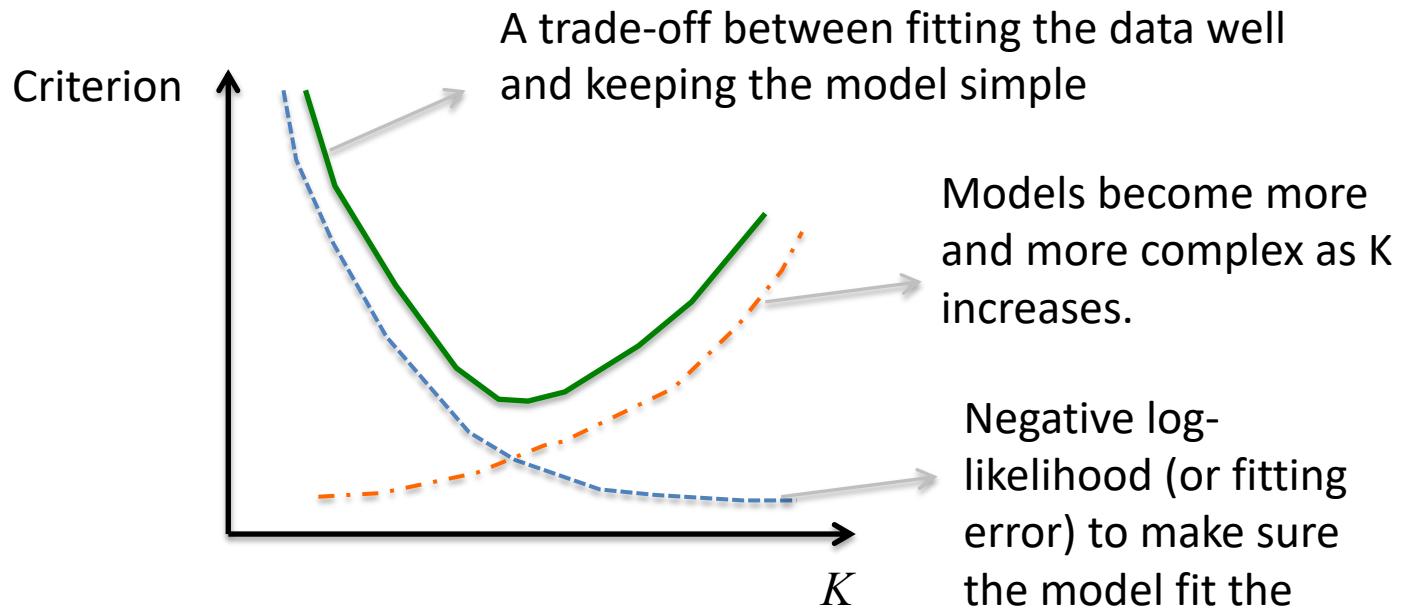
Model selection in general

Probabilistic model

$$p(X_N | \Theta_K)$$

Candidate models:

$$\Theta_1 \subseteq \Theta_2 \subseteq \dots \subseteq \Theta_K \subseteq \dots$$



Akaike's Information Criterion (AIC)

$$\ln p(X_N | \hat{\Theta}_K) - d_k$$

d_k : number of free parameters

Bayesian Information Criterion (BIC)

$$\ln p(X_N | \hat{\Theta}_K) - \frac{1}{2} d_k \ln N$$

N : sample size

Thank you!