CSCI5521: Introduction to Machine Learning (Spring 2020)

# Clustering

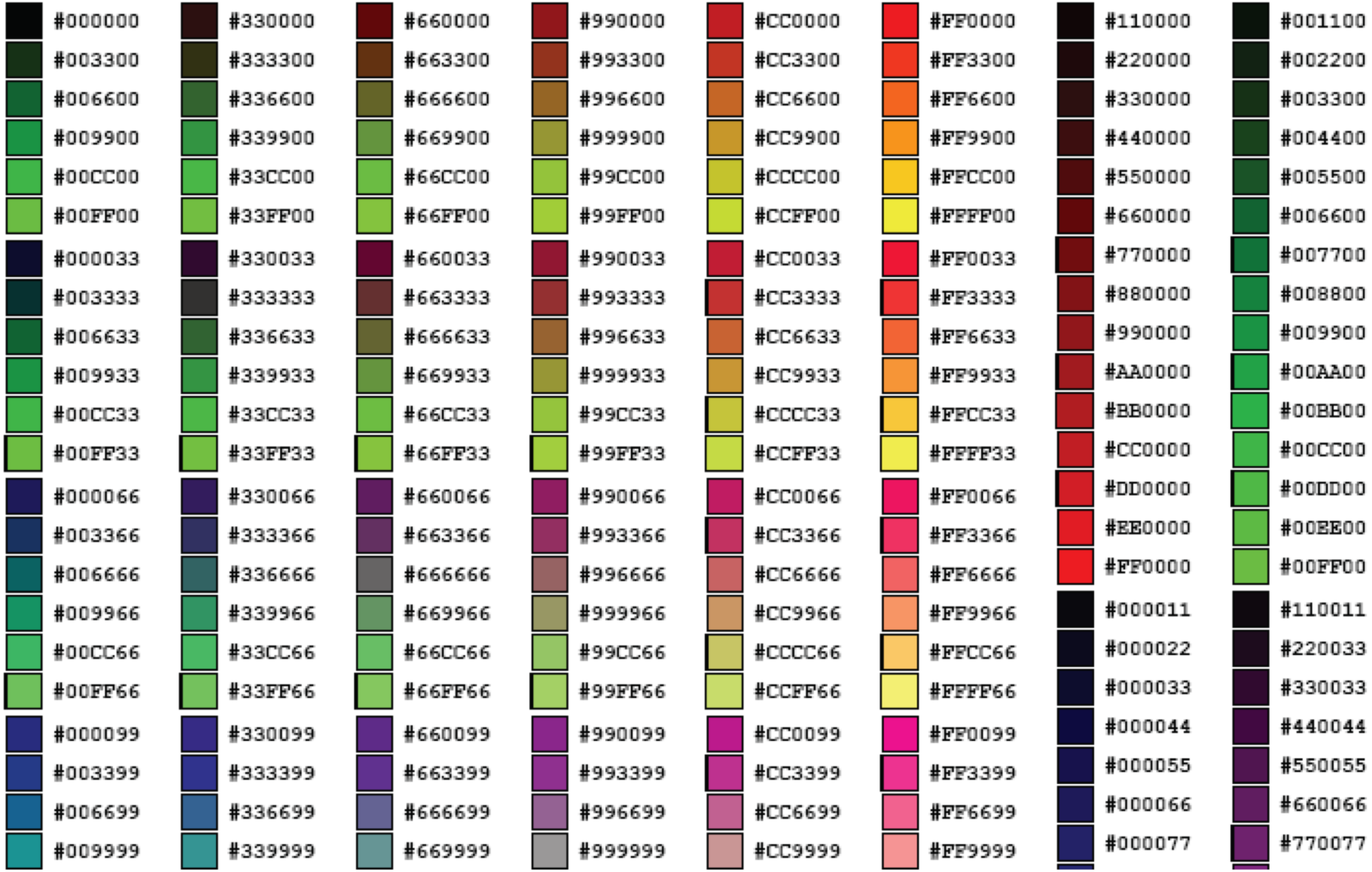
**Rui Kuang**

**Department of Computer Science and Engineering**

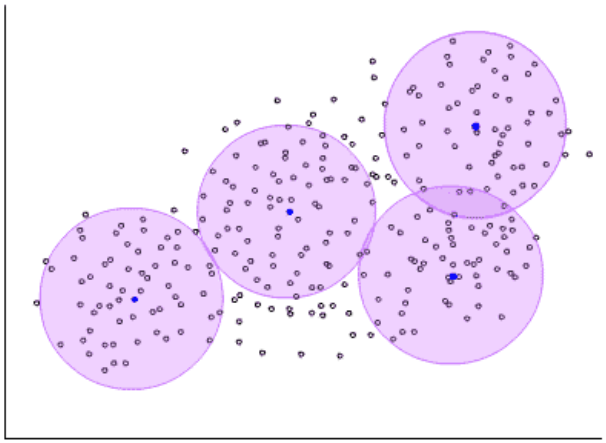
**University of Minnesota**

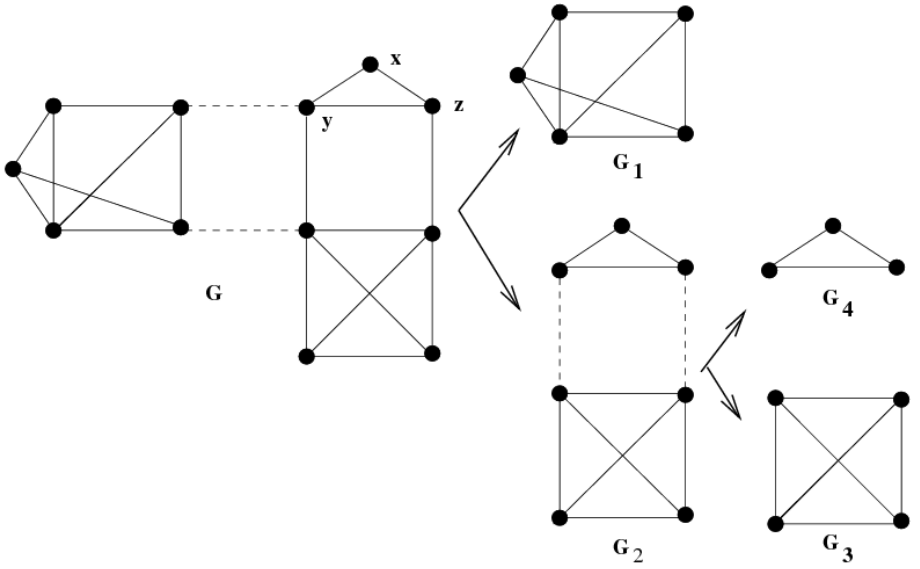
# How to Compress a Image?

# RGB Colour Codes



# How to Do Clustering?






# *k*-Means Clustering

- Find *k* reference vectors (prototypes/codebook vectors/codewords) which best represent data $\left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\|$
- Reference vectors, $\boldsymbol{m}_j$, $j = 1,\ldots,k$
- Reconstruction error:

$$E\left(\left\{\mathbf{m}_i\right\}_{i=1}^{k} | \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$$

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

# *k*-Means Clustering

- Minmizing reconstruction error:

$$E\left(\{\mathbf{m}_i\}_{i=1}^{k} | \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$$

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

$$\frac{\partial \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2}{\partial m_i} = 2 \sum_t b_i^t (\mathbf{x}^t - \mathbf{m}_i) = 0$$

$$\mathbf{m}_i = \frac{\sum_t b_i^t \mathbf{x}^t}{\sum_t b_i^t}$$

# k-means Clustering

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

Repeat

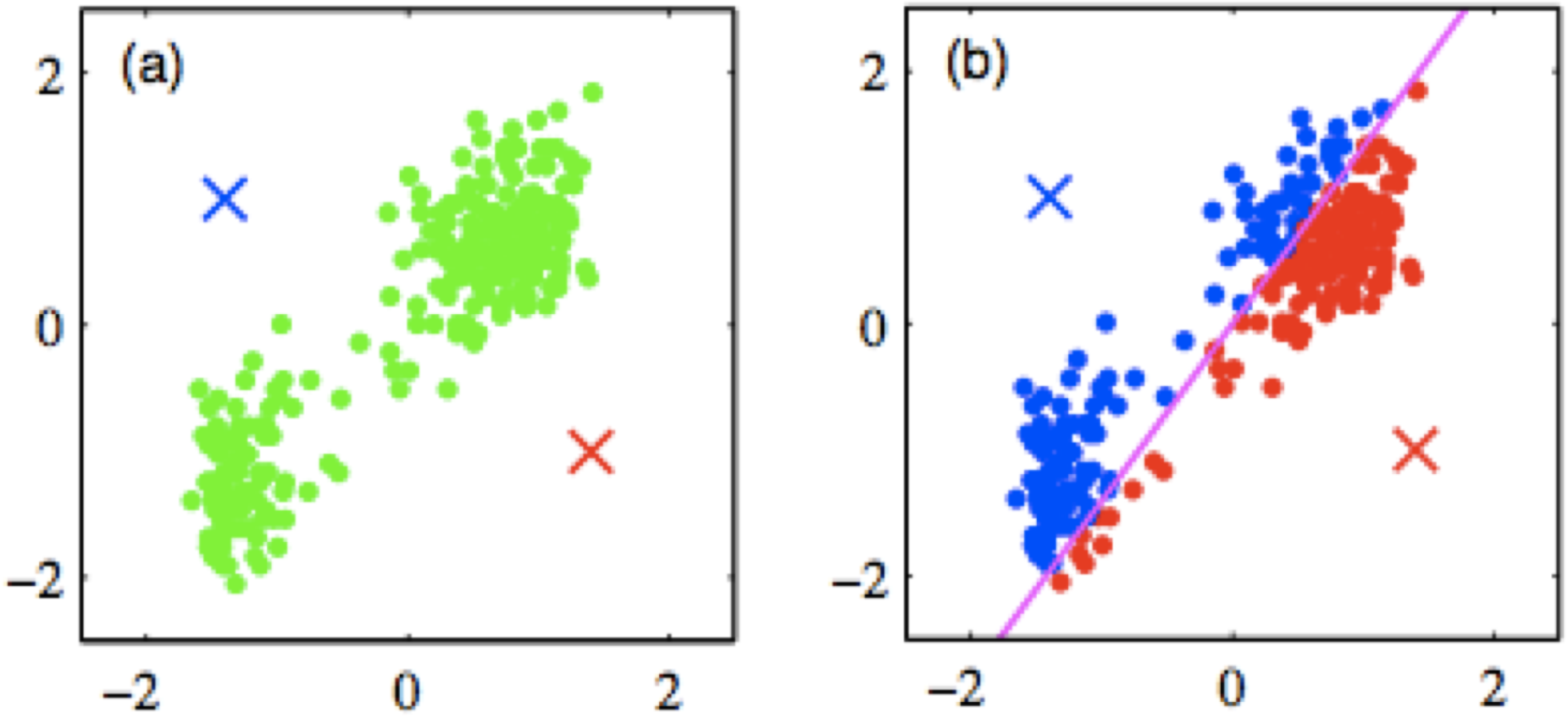
For all $\boldsymbol{x}^t \in \mathcal{X}$

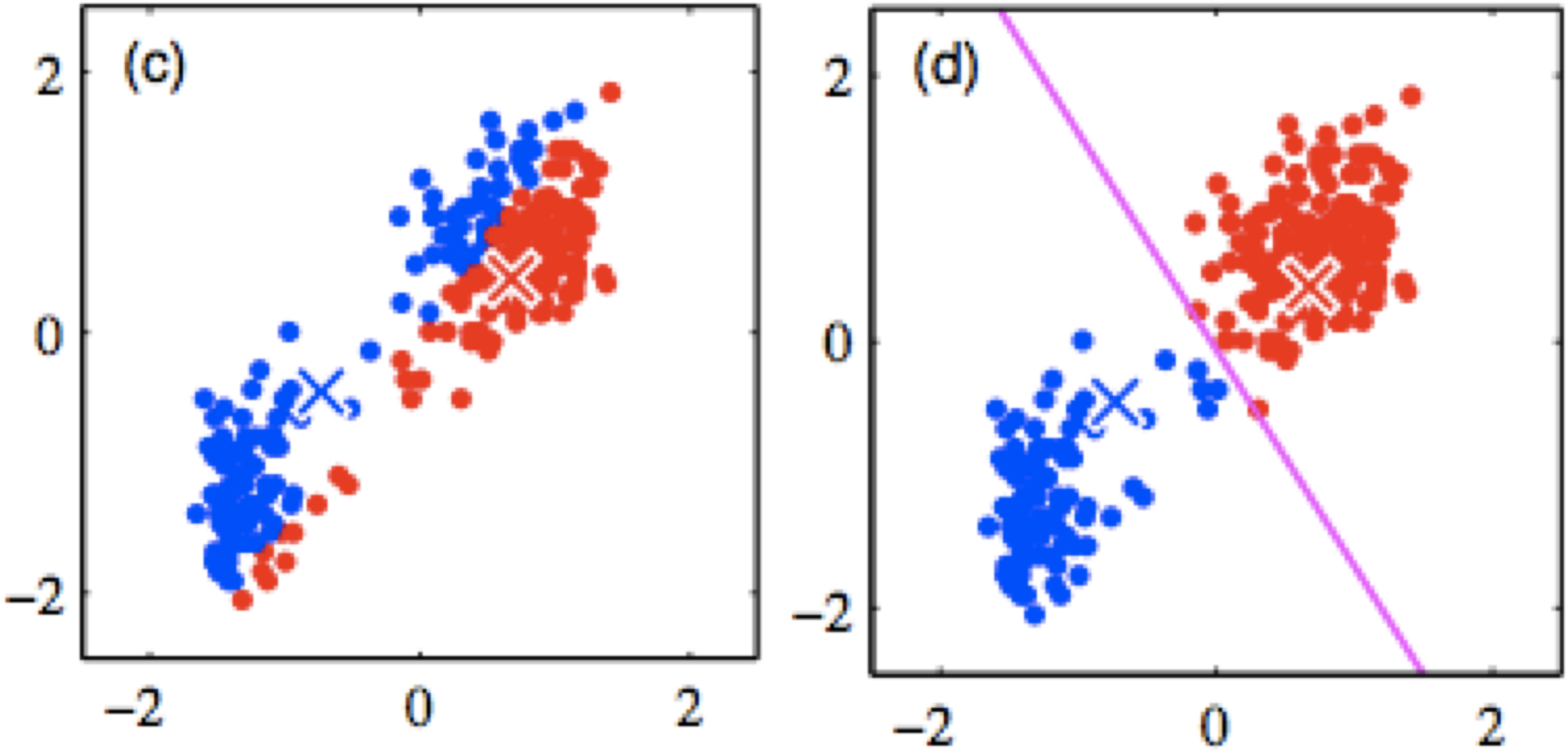
$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

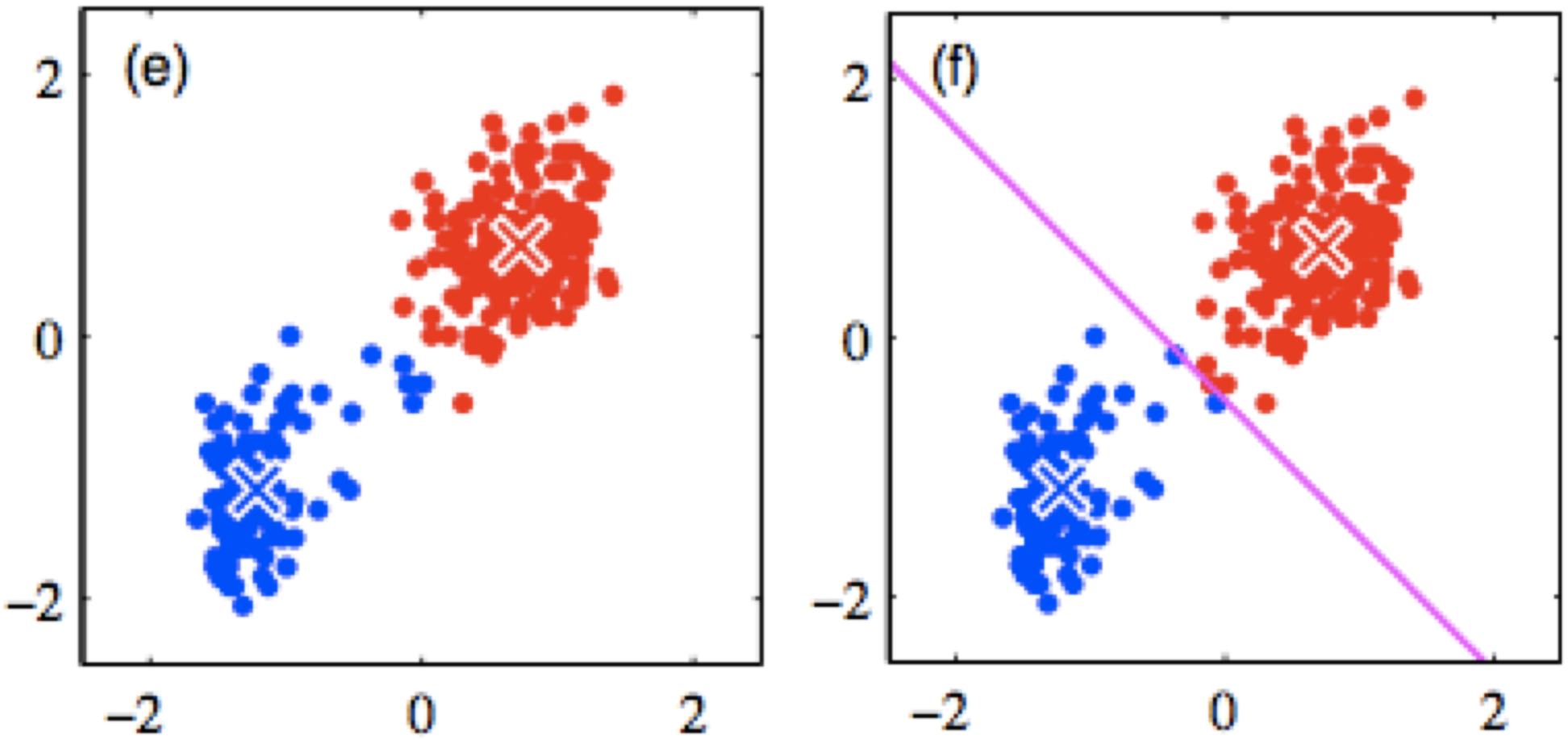
For all $\boldsymbol{m}_i, i = 1, \ldots, k$

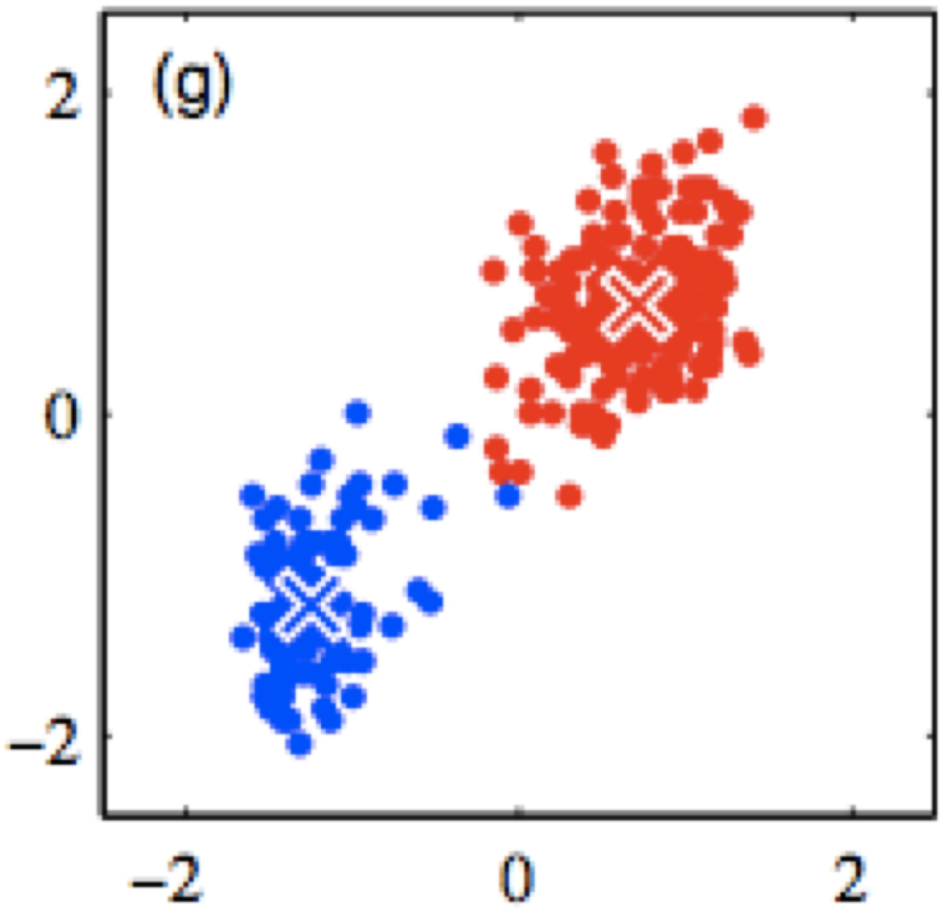
$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

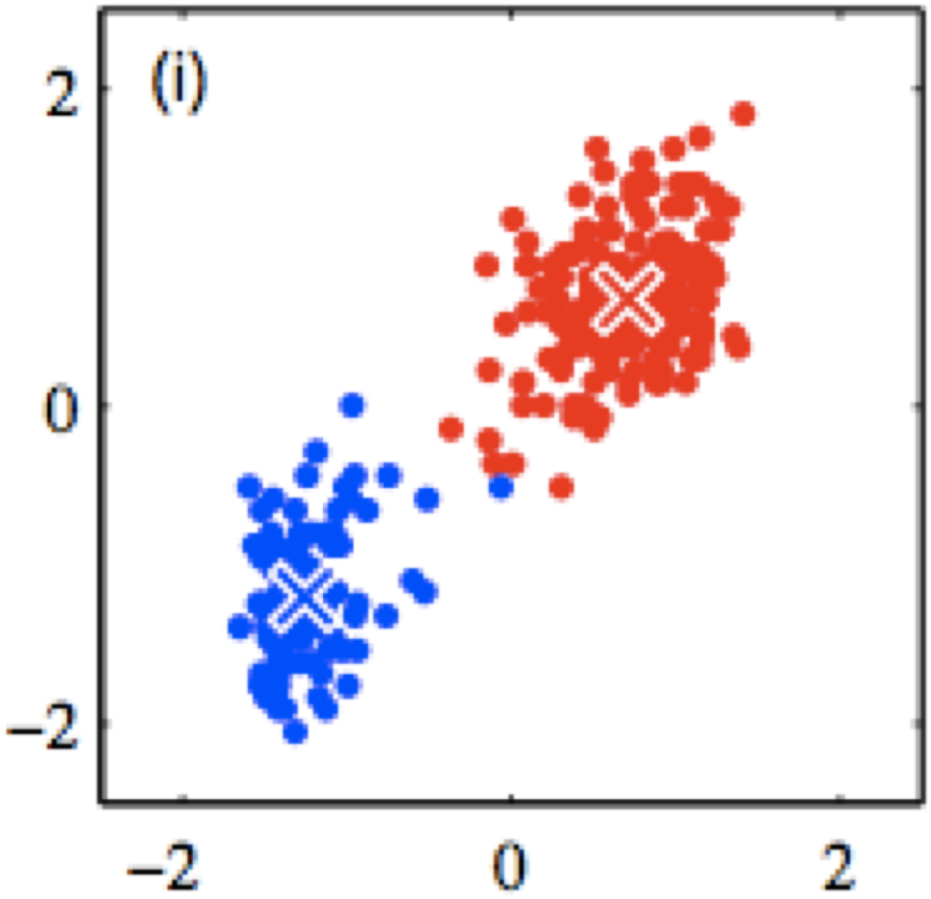
Until $\boldsymbol{m}_i$ converge


(a)
2
0
−2
−2
0
2
(b)
2
0
−2
−2
0
2

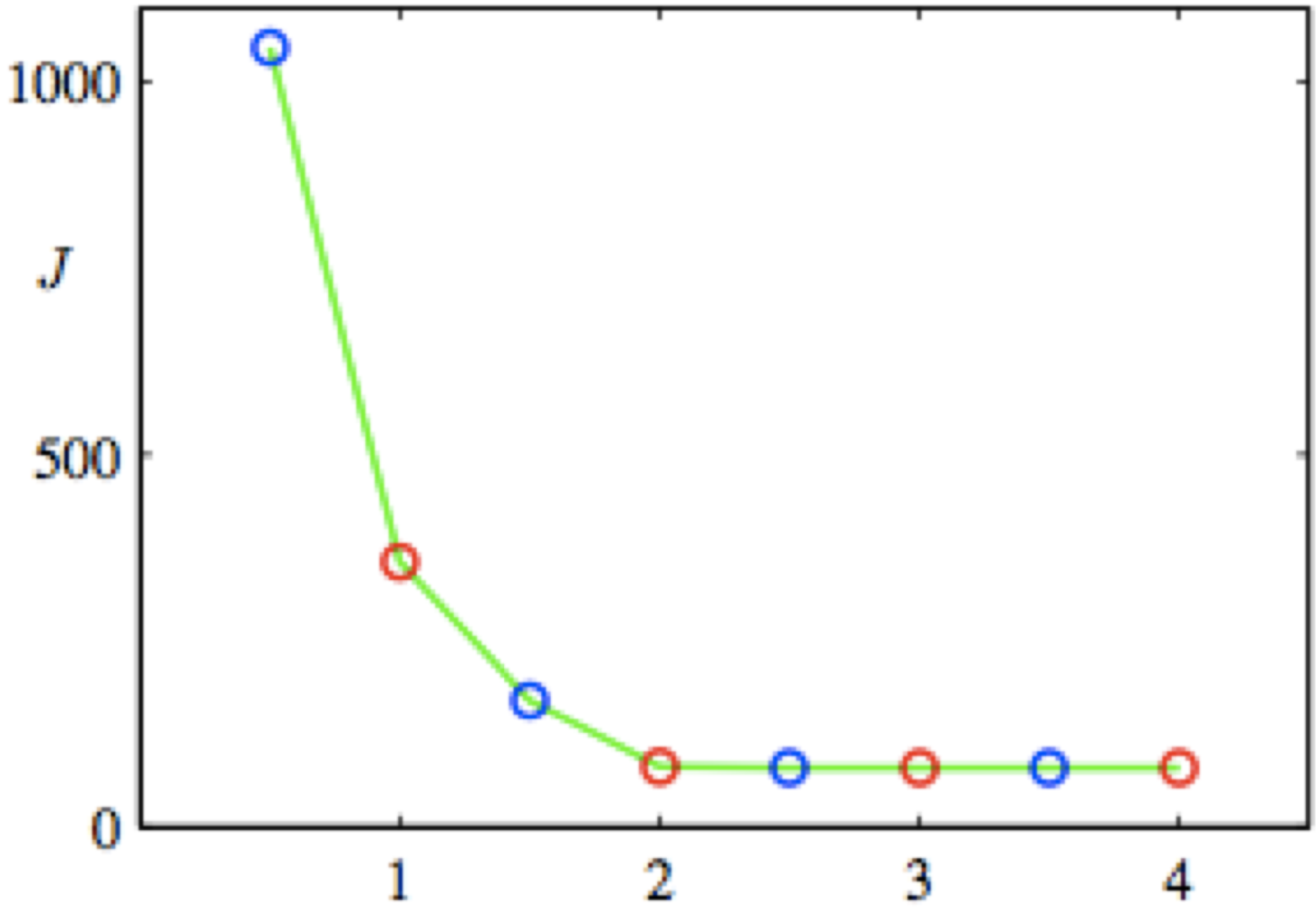

(c)
2
0
−2
−2
0
2
(d)
2
0
−2
−2
0
2


(e)
2
0
−2
−2
0
2
(f)
2
0
−2
−2
0
2


(g)
2
0
−2
−2
0
2


(i)
2
0
−2
−2
0
2

# Reconstruction Error

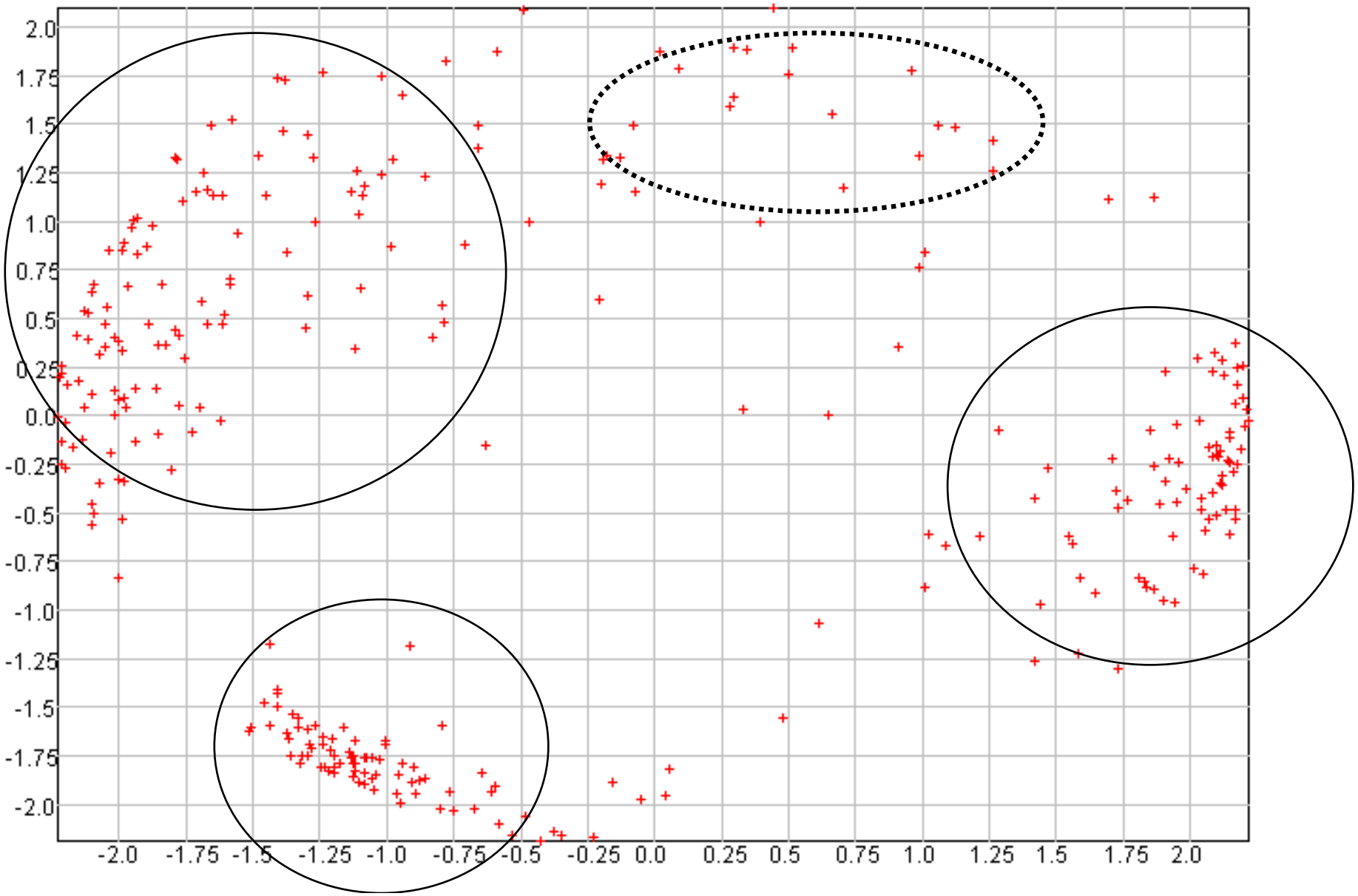


# Choosing *k*

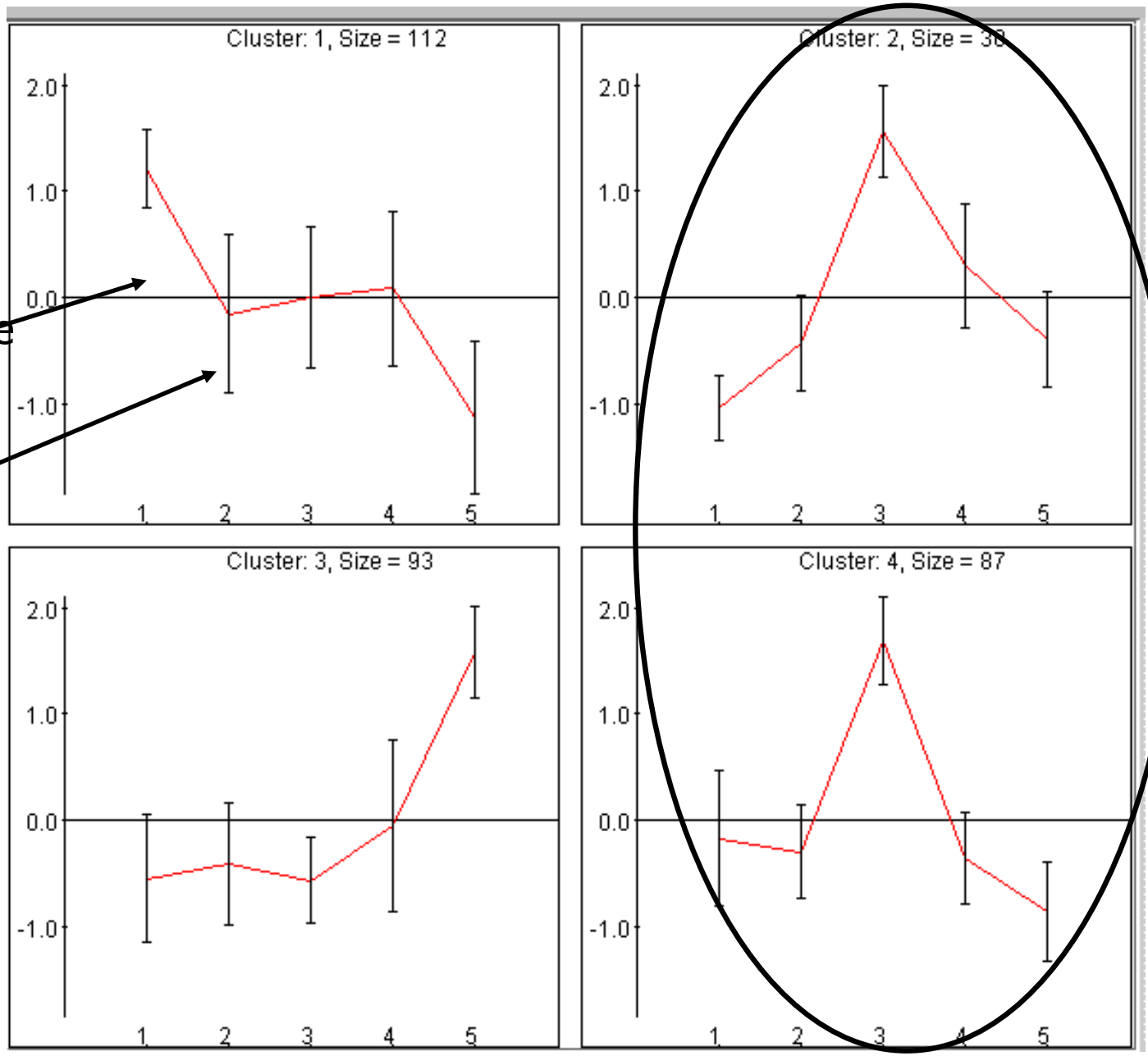
- Defined by the application, e.g., image quantization
- Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)
- Plot data (after PCA) and check for clusters

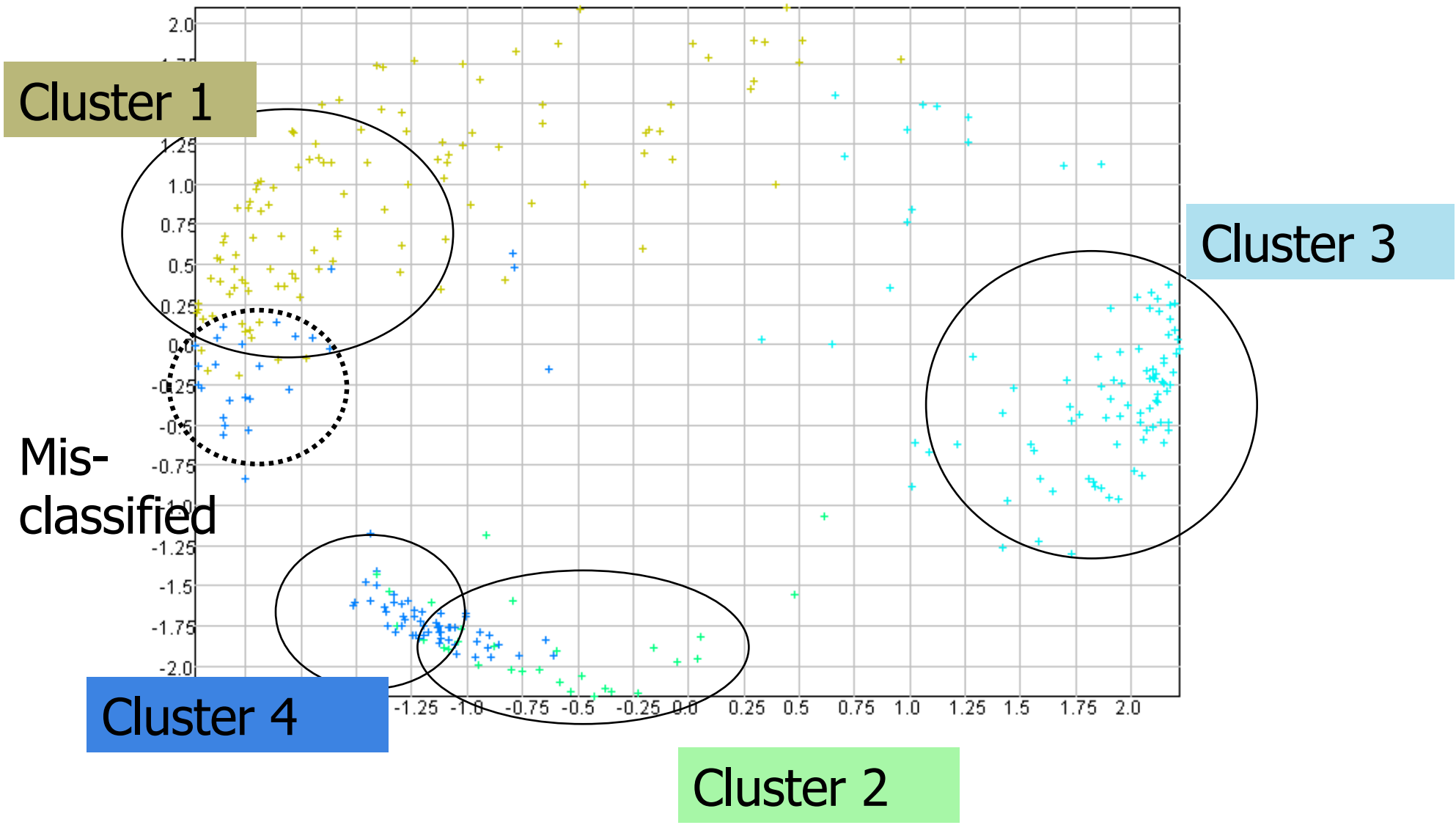
# PCA – Example

Visual estimation of the number of clusters in the data

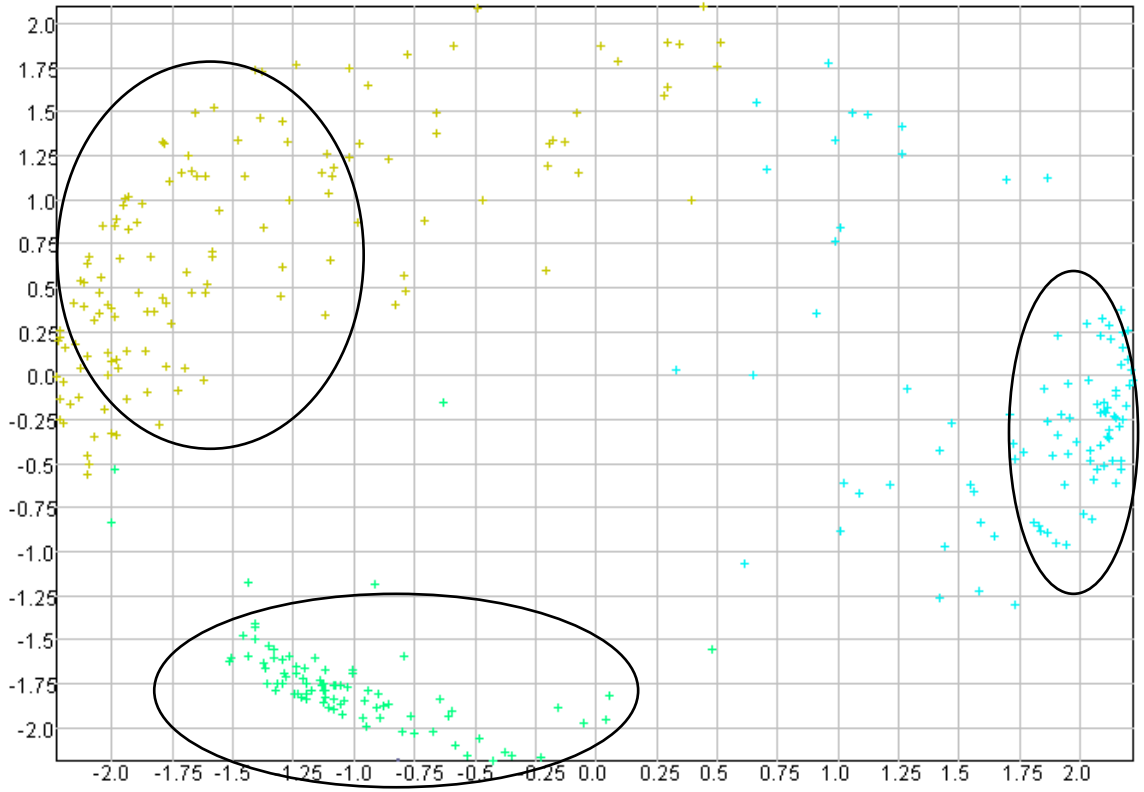


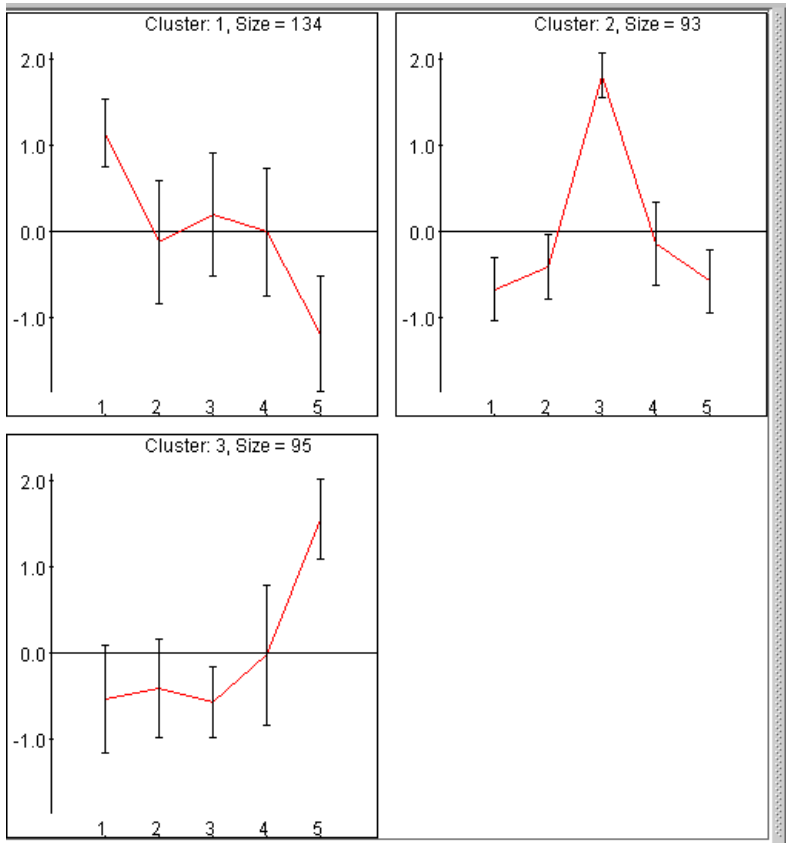
# K-MEANS example: 4 clusters



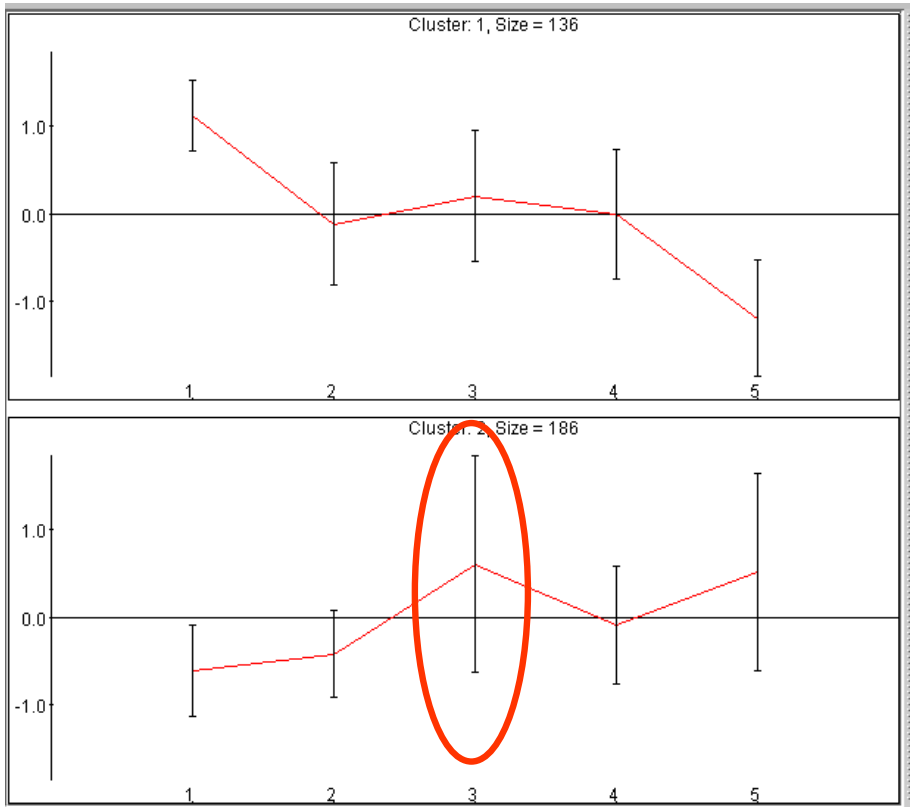

Cluster 1
Cluster 3
Mis-
classified
Cluster 4
Cluster 2
2.0
1.25
1.0
0.75
0.5
0.25
0.0
-0.25
-0.5
-0.75
-1.0
-1.25
-1.5
-1.75
-2.0
-1.25 -1.0 -0.75 -0.5 -0.25 0.0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0

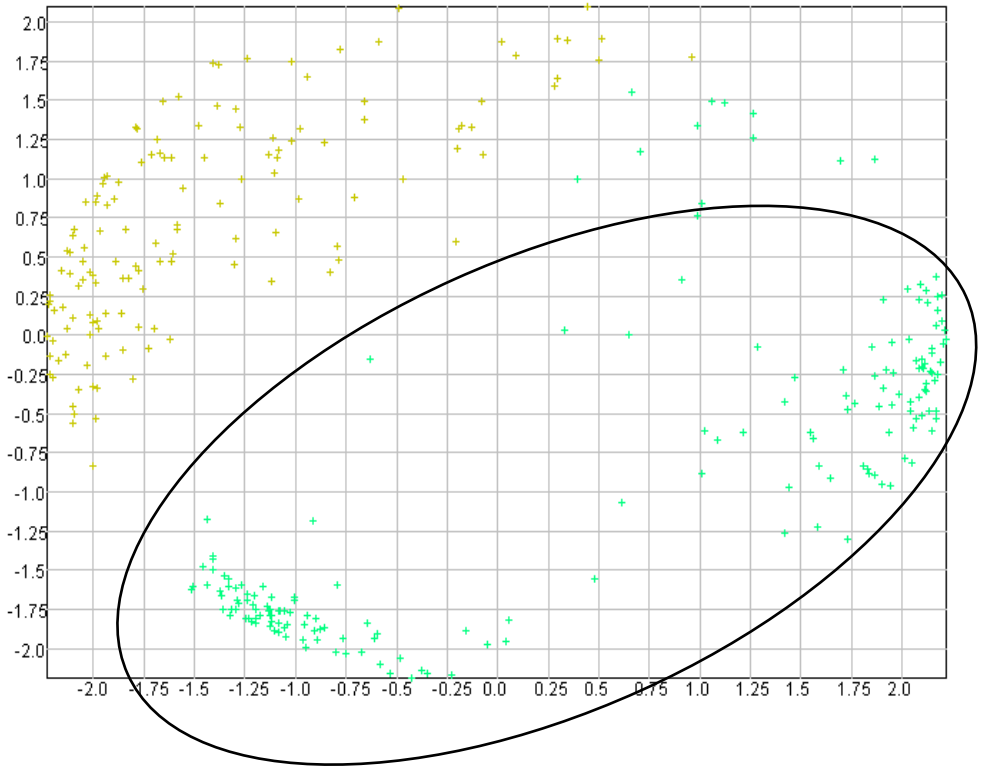
# K-means example: 3 clusters



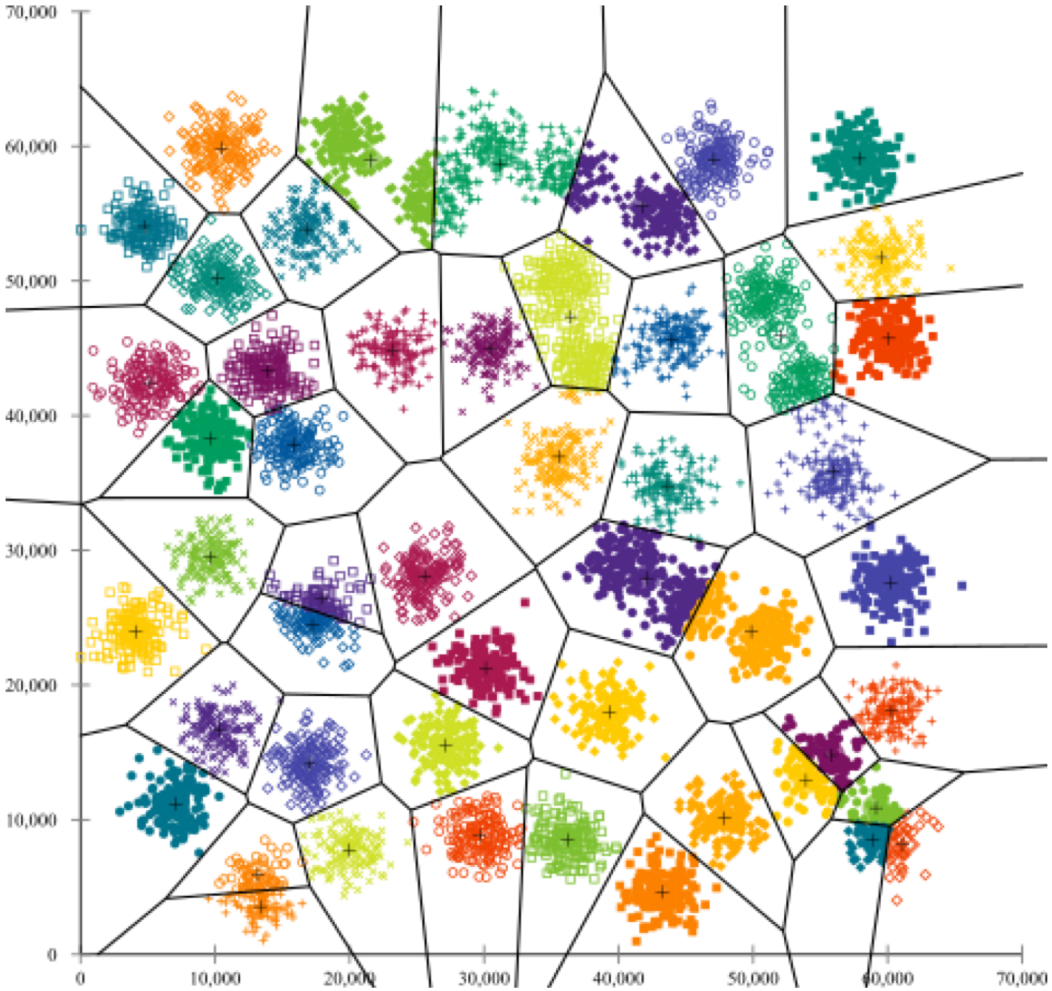



# Too few clusters: K=2



# *K*-means Local Minimum

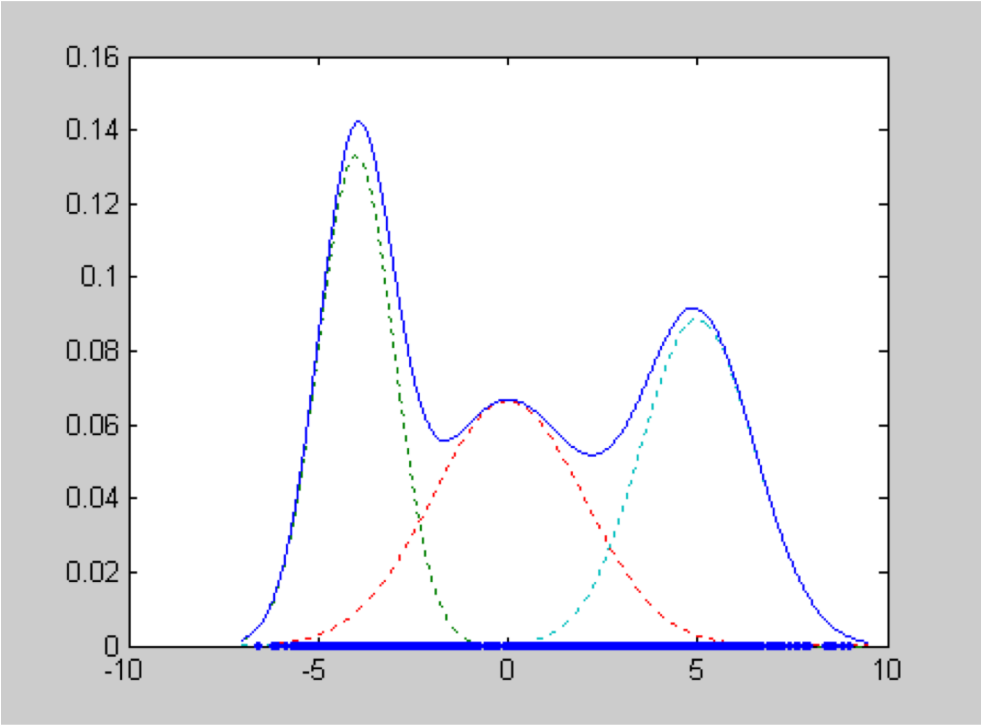


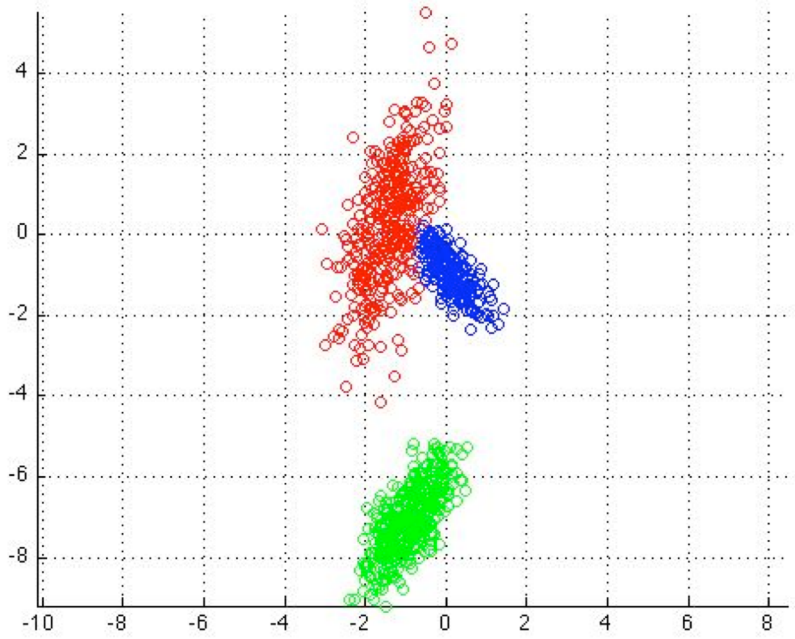

https://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means

# *K*-means Tips

- Local minimum: run multiple times and choose the one with the lowest reconstruction error.
- Collapsing on one data point and empty clusters.
- SSE (sum of squared errors) assumes spherical variances.

# Mixture of Gaussians




$$p(\mathbf{x} \mid \Phi) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x} \mid \mu_i, \Sigma_i),$$

$$\text{with } 0 \le \pi_i \le 1 \text{ and } \sum_{i=1}^{K} \pi_i = 1.$$

# Difficult ML Problem

- Maximum Likelihood

$$\mathcal{L}(\Phi|\mathcal{X}) = \sum_t \log \sum_{i=1}^{k} p\left(\mathbf{x}^t|G_i\right) P\left(G_i\right) = \sum_t \log \sum_{i=1}^{k} \pi_i \mathcal{N}\left(\mathbf{x}^t \mid \mu_i, \Sigma_i\right)$$

- Take derivative wrt $\mu_i$

$$0 = -\sum_{t=1}^{N} \frac{\pi_i \mathcal{N}\left(x^t \mid \mu_i, \Sigma_i\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(x^t \mid \mu_j, \Sigma_j\right)} \Sigma_i^{-1}(x^t - \mu_i)$$

$\gamma(z_i^t)$

$$\mu_i = \frac{1}{N_i} \sum_{t=1}^{N} \gamma(z_i^t)\mathbf{x}^t, \quad N_i = \sum_{t=1}^{N} \gamma(z_i^t)$$

# Mixture of Gaussians

- Maximum Likelihood

$$\mathcal{L}(\Phi \mid \mathcal{X}) = \sum_t \log \sum_{i=1}^{k} p\left(\mathbf{x}^t \mid G_i\right) P\left(G_i\right) = \sum_t \log \sum_{i=1}^{k} \pi_i \mathcal{N}\left(\mathbf{x} \mid \mu_i, \Sigma_i\right)$$

- Take derivative wrt $\Sigma_i$

$$\Sigma_i = \frac{1}{N_i} \sum_{t=1}^{N} \gamma(z_i^t)(x^t - \mu_i)(x^t - \mu_i)^T$$

- Take derivative wrt $\pi_i$

$$\frac{\partial \sum_t \log \sum_{i=1}^{k} \pi_i \mathcal{N}\left(\mathbf{x} \mid \mu_i, \Sigma_i\right) + \alpha\left(\sum_{i=1}^{K} \pi_i - 1\right)}{\partial \pi_i} = 0 \Leftrightarrow \pi_i = \frac{N_i}{N}$$

# EM on Gaussian Mixtures

- E-step:

$$\gamma(z_i) = \frac{\pi_i \mathcal{N}\left(\mathbf{x} \mid \mu_i, \Sigma_i\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x} \mid \mu_j, \Sigma_j\right)}$$

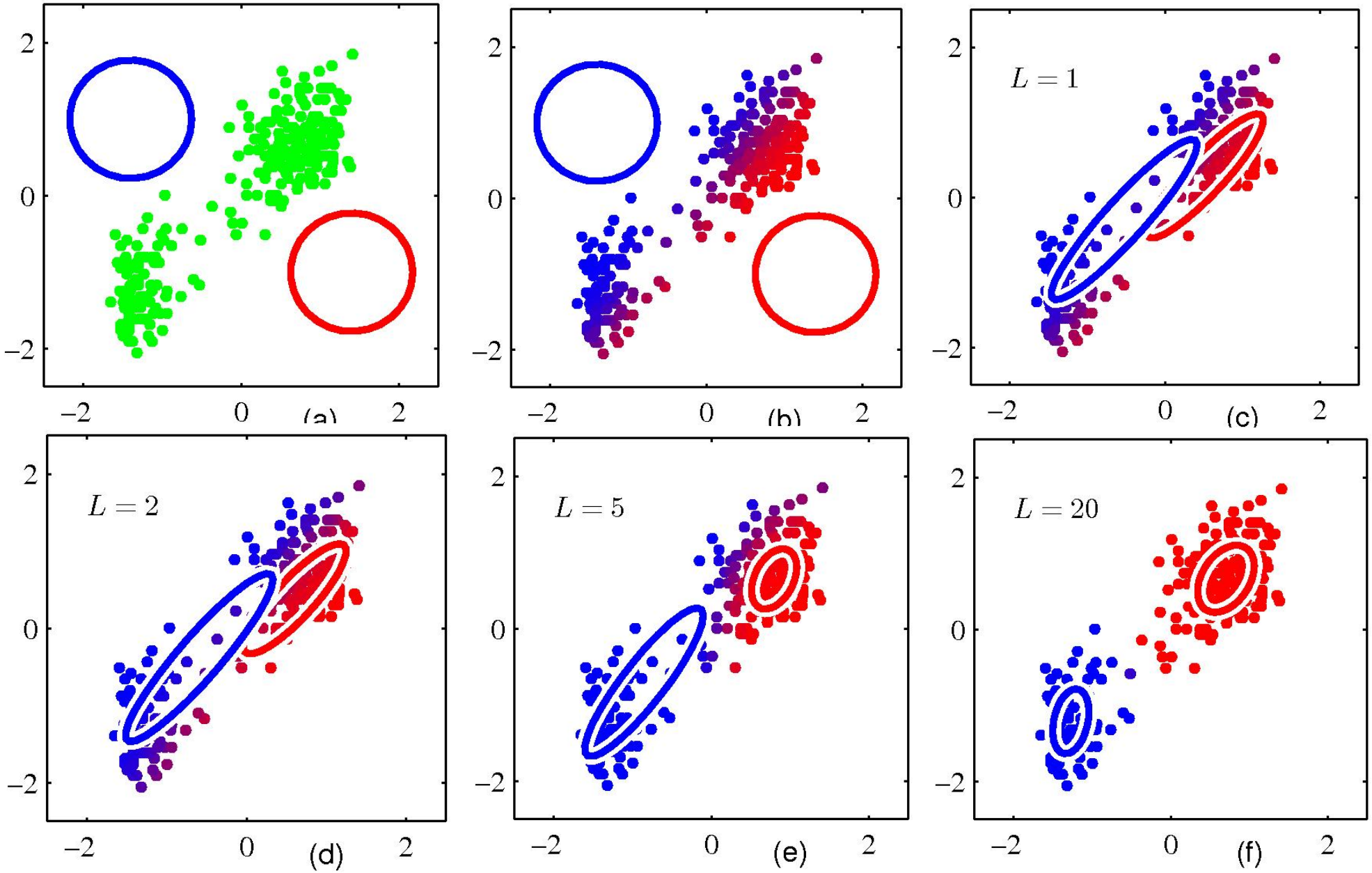
- M-step:

$$\mu_i = \frac{1}{N_i} \sum_{t=1}^{N} \gamma(z_i^t) x_i, \quad N_i = \sum_{t=1}^{N} \gamma(z_i^t)$$

$$\pi_i = \frac{N_i}{N}$$

$$\Sigma_i = \frac{1}{N_i} \sum_{t=1}^{N} \gamma(z_i^t)(x^t - \mu_i)(x^t - \mu_i)^T$$

# EM on Gaussian Mixtures Example



# The Role of Hidden Variables

- Consider coupling a multinomial variable with Gaussian. $z_i$ = 1 if ***x*** belongs to $G_i$, 0 otherwise;

$$p(z) = \prod_{i=1}^{K} \pi_i^{z_i} \qquad p(x \mid z) = \prod_{i=1:K} \mathcal{N}(\mathbf{x} \mid \mu_i, \Sigma_i)^{z_i}$$

- Joint distribution

$$p(x,z) = p(z) * p(x \mid z) = \prod_{i=1}^{K} (\pi_i \mathcal{N}(\mathbf{x} \mid \mu_i, \Sigma_i))^{z_i}$$

- Work with complete likelihood on *p(x,z)*.

# Complete Likelihood

- Complete likelihood, $\mathcal{L}_c(\Phi \mid X,Z)$, in terms of ***x*** and ***z***

$$\mathcal{L}_c\left(\Phi|\mathcal{X}\right) = \log \prod_t p(x^t, z^t|\Phi) = \sum_t \log p\left(x^t, z^t|\Phi\right)$$

$$= \sum_t \log p\left(z^t|\Phi\right) + \log p\left(x^t|z^t, \Phi\right)$$

$$= \sum_t \sum_i z_i^t \left[\log \pi_i + \log p(x^t \mid \Phi_i)\right]$$

- Work with the expectation of $\mathcal{L}_c(\Phi \mid X)$

$$\mathcal{Q}(\Phi \mid \Phi^l) = E\left[\mathcal{L}_c\left(\Phi \mid \mathcal{X}, \mathcal{Z}\right) \mid \chi, \Phi^l\right]$$

$$= \sum_t \sum_i E\left[z_i^t \mid \chi, \Phi^l\right]\left[\log \pi_i + \log p(x^t \mid \Phi_i^l)\right]$$

# Expectation (EM)

- E-Step: Expectation

$$\mathcal{Q}(\Phi \mid \Phi^l) = E\left[\mathcal{L}_c(\Phi \mid \mathcal{X}, \mathcal{Z}) \mid \chi, \Phi^l\right]$$

$$= \sum_t \sum_i E\left[z_i^t \mid \chi, \Phi^l\right]\left[\log \pi_i + \log p(x^t \mid \Phi_i^l)\right]$$

$$E\left[z_i^t \mid \chi, \Phi^l\right] = E\left[z_i^t \mid x^t, \Phi^l\right] = p\left(z_i^t = 1 \mid x^t, \Phi^l\right)$$

$$= \frac{p\left(z_i^t = 1 \mid \Phi^l\right) p(x^t \mid z_i^t = 1, \Phi^l)}{p\left(x^t \mid \Phi^l\right)}$$

$$= \frac{\pi_i p(x^t \mid \Phi_i^l)}{\sum_j \pi_j p\left(x^t \mid \Phi_j^l\right)} = P(z_i^t = 1 \mid x^t, \Phi^l) \equiv \gamma(z_i^t)$$

# Maximization (EM)

- M-Step: Maximization

$$\Phi^{l+1} = \arg\max_{\Phi} \mathcal{Q}(\Phi \mid \Phi^{l})$$

$$\mathcal{Q}(\Phi \mid \Phi^{l}) = \sum_{t}\sum_{i} \gamma(z_i^t)\left[\log \pi_i + \log p(x^t \mid \Phi_i^l)\right]$$

- Maximum likelihood learning, e.g. gaussian

$$\pi_i = \frac{N_i}{N} \qquad \mu_i = \frac{1}{N_i}\sum_{t=1}^{N} \gamma(z_i^t) x_i, \quad N_i = \sum_{t=1}^{N} \gamma(z_i^t)$$

$$\Sigma_i = \frac{1}{N_i}\sum_{t=1}^{N} \gamma(z_i^t)(x^t - \mu_i)(x^t - \mu_i)^T$$

# EM summary

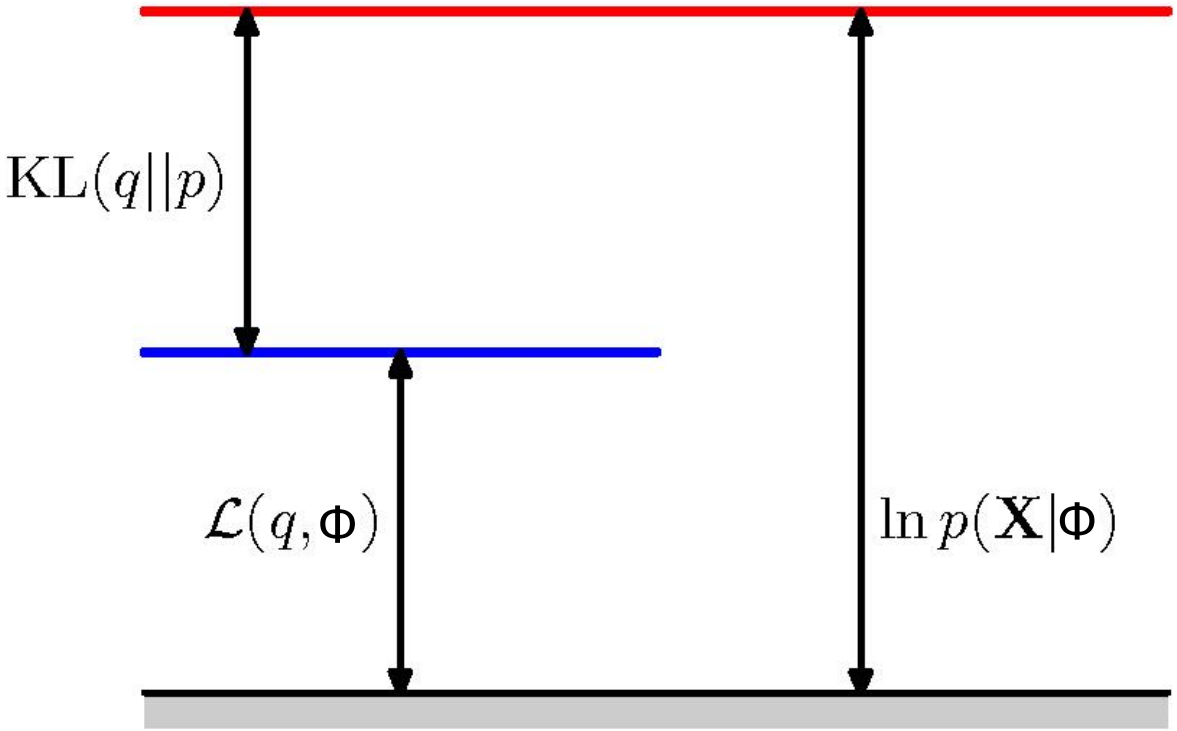
- The likelihood function of mixture of Gaussians is intractable (log sum problem).
- The complete likelihood function is a product of Gaussians selected by $z_i$ (log product).
- EM aims to maximize the expectation (over $z$) of the complete likelihood wrt the parameters of mixture of Gaussians.
- The expectation of the complete likelihood relies on the expectation of each $z_i$, which is re-estimated in each iteration.

# Why EM Converges?

$$\ln p(X \mid \Phi) = \sum_Z q(Z) \ln p(X \mid \Phi) = \sum_Z q(Z) \ln \frac{p(X, Z \mid \Phi)}{p(Z \mid X, \Phi)}$$

$$= \sum_Z q(Z) \ln \frac{p(X, Z \mid \Phi)}{q(Z)} - \sum_Z q(Z) \ln \frac{p(Z \mid X, \Phi)}{q(Z)}$$

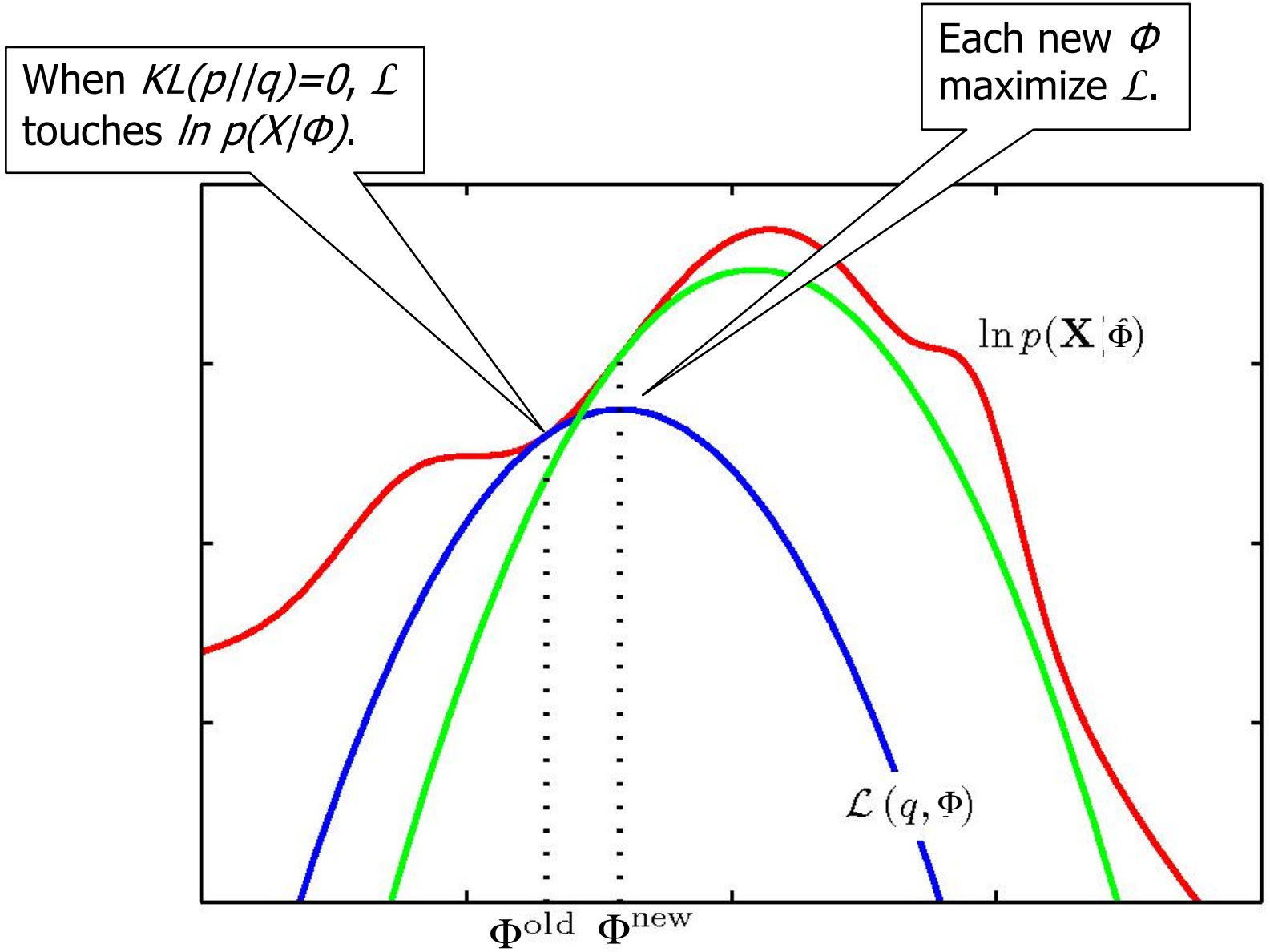
$$= \mathcal{L}(q, \Phi) + KL(q \,\|\, p)$$




Since $KL(q \,\|\, p) \geq 0$,

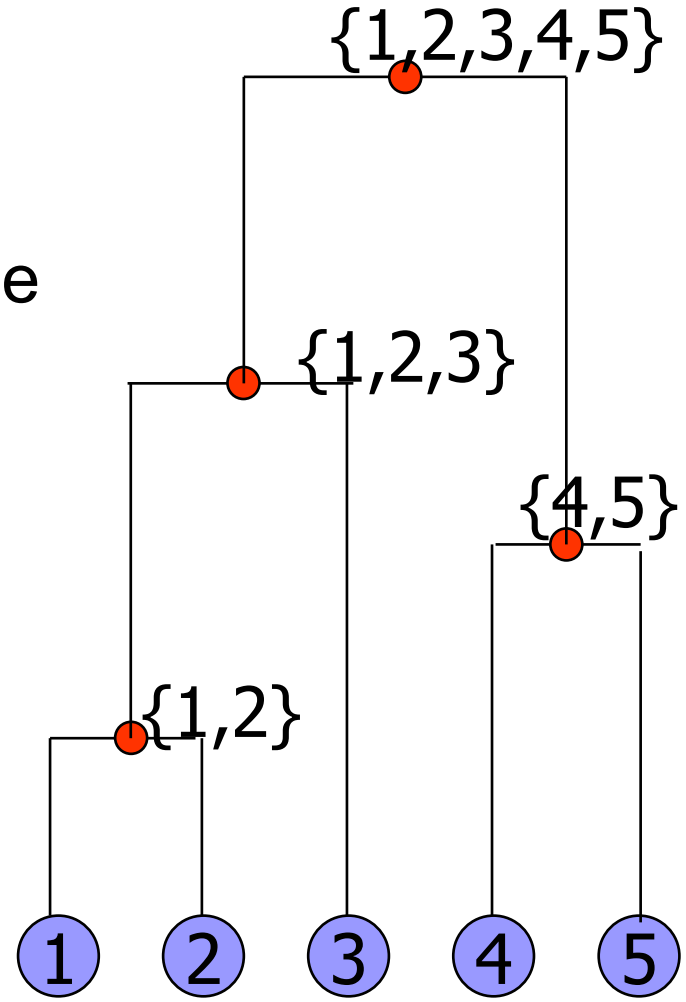
$\ln p(X \mid \Phi) \geq \mathcal{L}(q, \Phi)$.

# Analysis of EM



# Hierarchical Clustering

- Organize the samples in a structure of a hierarchical tree
- Agglomerative (bottom up)
- Divisive (top down)

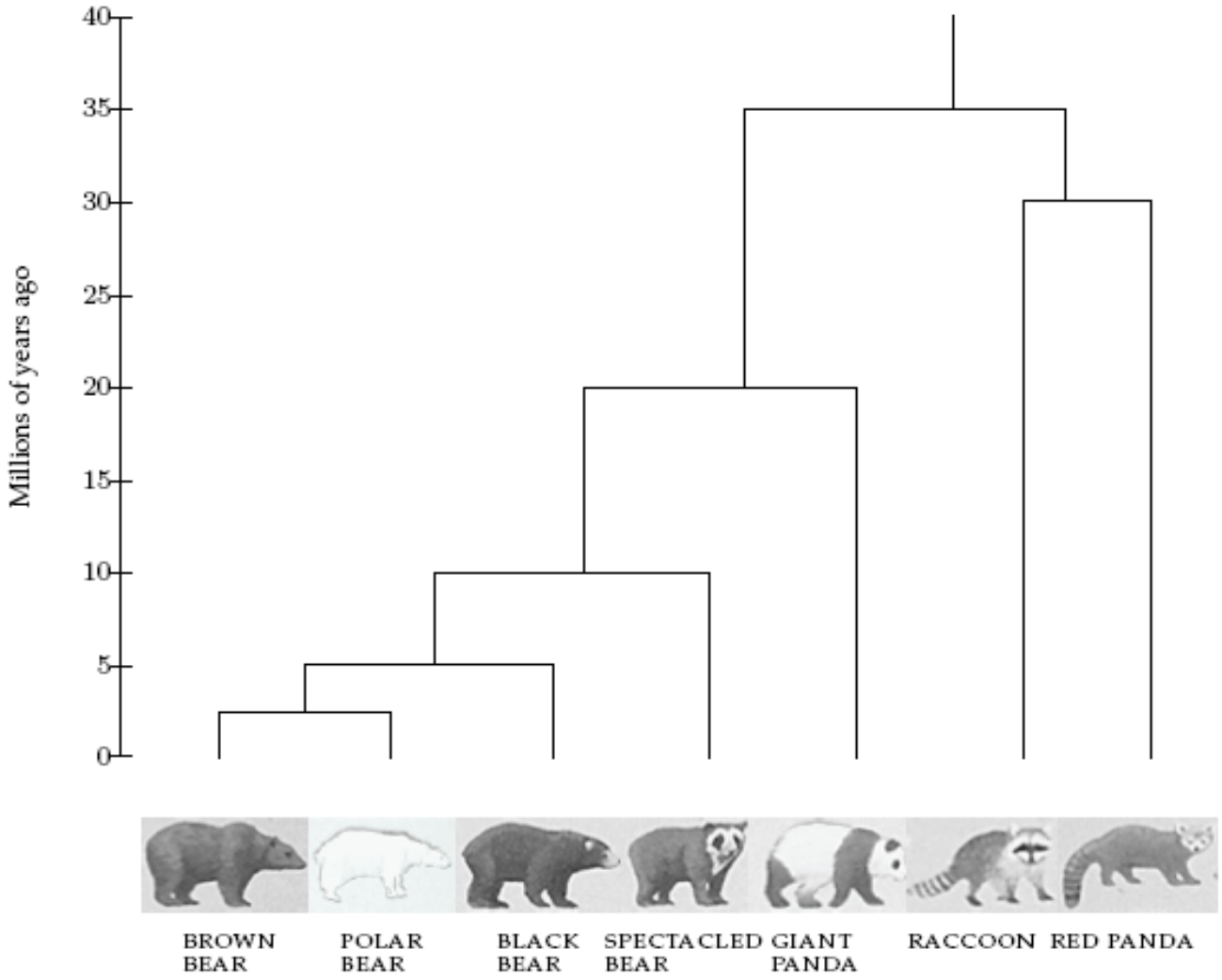


# Evolution and DNA Analysis: the Giant Panda Riddle

- For roughly 100 years scientists were unable to figure out which family the giant panda belongs to

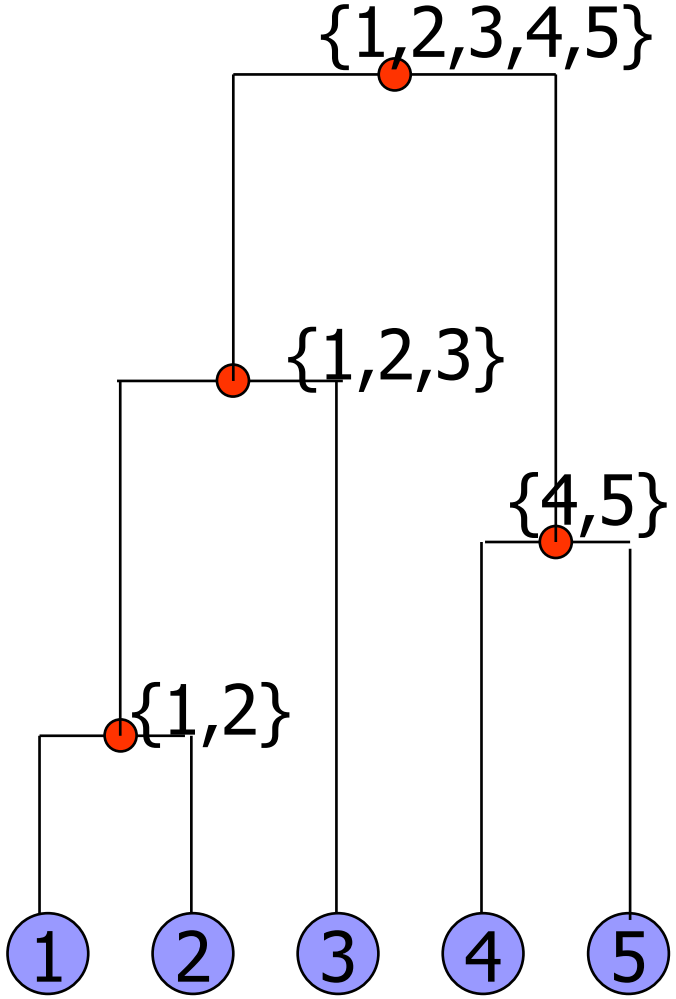

- Giant pandas look like bears but have features that are unusual for bears and typical for raccoons, e.g., they do not hibernate
- In 1985, Steven O'Brien and colleagues solved the giant panda classification problem using DNA sequences and algorithms

# Phylogenetic tree of bears and raccoons



# Hierarchical Clustering: Agglomerative

- Initial step: each sample is regarded as a cluster with one item
- Find the 2 most similar clusters and merge them into a common node
- The length of the branch is proportional to the distance
- Iterate on merging nodes until all samples are contained in one cluster- the root of the tree.



# Hierarchical Clustering

- Cluster based on similarities/distances
- Distance measure between instances $\boldsymbol{x}^r$ and $\boldsymbol{x}^s$

  Minkowski ($L_p$) (Euclidean for $p = 2$)

$$d_m\left(\mathbf{x}^r,\mathbf{x}^s\right) = \left[\sum_{j=1}^{d}\left(x_j^r - x_j^s\right)^p\right]^{1/p}$$

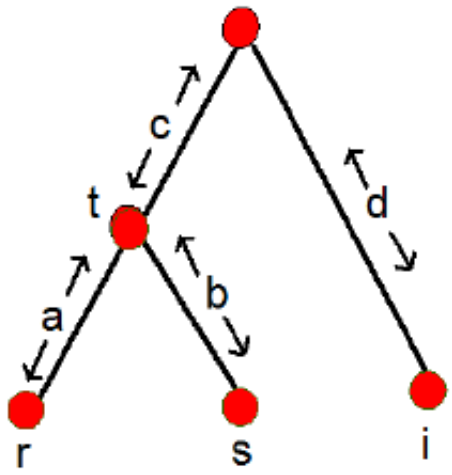
  City-block distance

$$d_{cb}\left(\mathbf{x}^r,\mathbf{x}^s\right) = \sum_{j=1}^{d}\left|x_j^r - x_j^s\right|$$

# Algorithm

- Input: The distance matrix *D*
- Find the pair *r, s* with the least distance
- Merge clusters *r, s*.
- Delete elements *r, s* and add a new element *t* with

$$D_{it} = D_{ti} = \frac{D_{ir} + D_{is} - D_{rs}}{2}$$

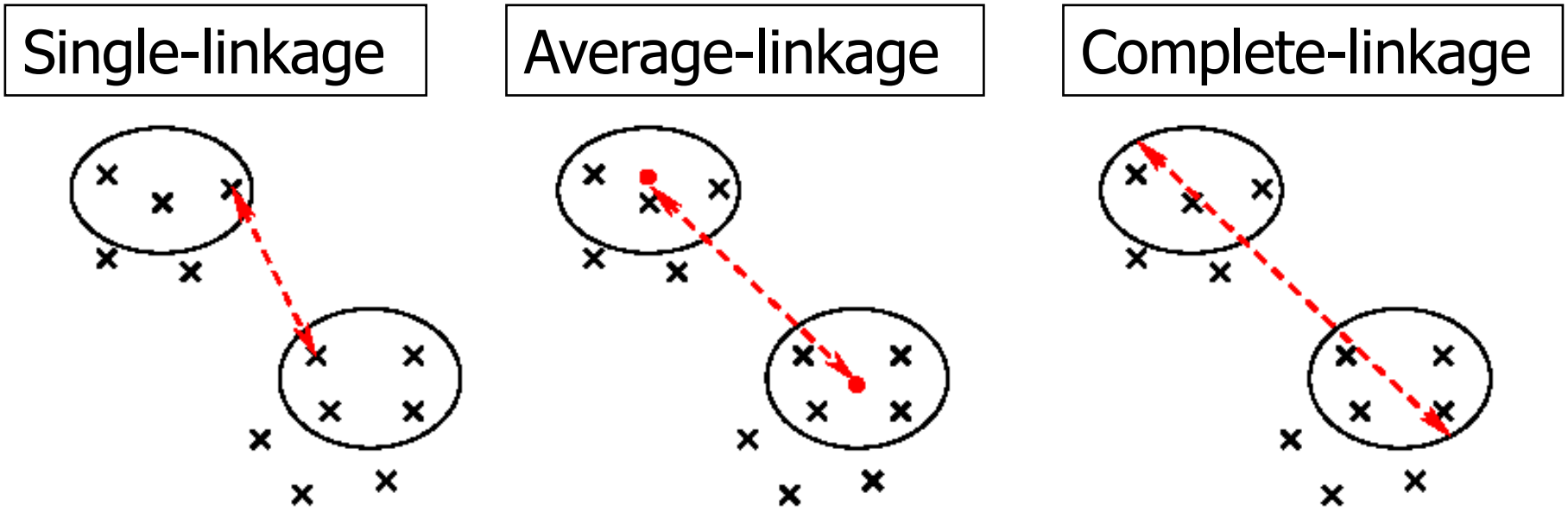



- Repeat until one element is left

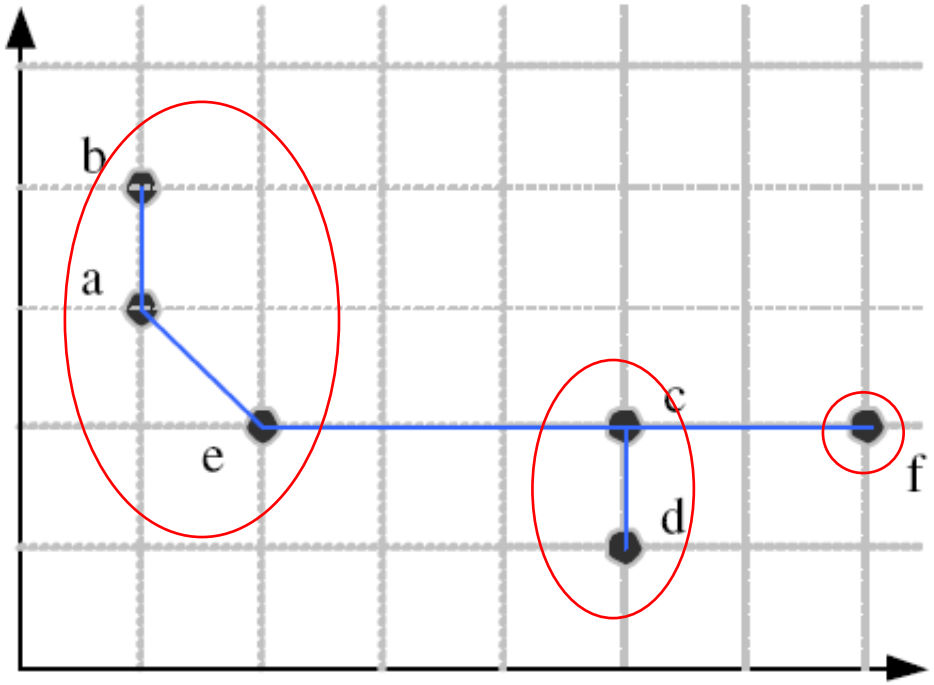
$$D_{it} = c + d = \frac{(c + d + a) + (c + d + b) - (a + b)}{2} = \frac{D_{ir} + D_{is} - D_{rs}}{2}$$

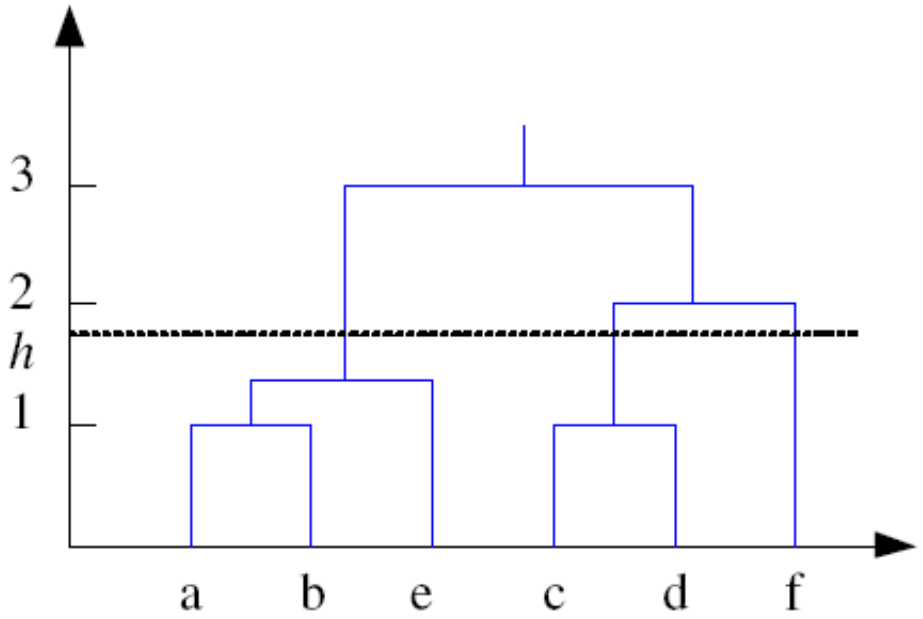
# Algorithm

- Average linkage: distance between the centroids
- Single-link: minimum distance (long and thin)
- Complete-link: maximum distance (tight)



# Example: Single-Link Clustering




*Dendrogram*