

MACHINE LEARNING

Chao Wang (*Frank*)

wangchao@nankai.edu.cn

CHAPTER 7:

CLUSTERING

Semiparametric Density Estimation

3

- Parametric: Assume a single model for $p(\mathbf{x} | C_i)$ (Chapters 4 and 5)
- Semiparametric: $p(\mathbf{x}|C_i)$ is a mixture of densities
Multiple possible explanations/prototypes:
Different handwriting styles, accents in speech
- Nonparametric: No model; data speaks for itself (Chapter 8)

Mixture Densities

4

$$p(\mathbf{x}) = \sum_{i=1}^k p(\mathbf{x} | G_i) P(G_i)$$

where G_i the components/groups/clusters,

$P(G_i)$ mixture proportions (priors),

$p(\mathbf{x} | G_i)$ component densities

Gaussian mixture where $p(\mathbf{x}|G_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$

parameters $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^k$

unlabeled sample $\mathbf{X} = \{\mathbf{x}^t\}_t$ (unsupervised learning)

Classes vs. Clusters

5

□ Supervised: $X = \{\mathbf{x}^t, \mathbf{r}^t\}_t$

□ Classes C_i $i=1, \dots, K$

$$p(\mathbf{x}) = \sum_{i=1}^K p(\mathbf{x} | C_i) P(C_i)$$

where $p(\mathbf{x} | C_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

□ $\Phi = \{P(C_i), \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^K$

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N} \quad \mathbf{m}_i = \frac{\sum_t r_i^t \mathbf{x}^t}{\sum_t r_i^t}$$

$$\mathbf{S}_i = \frac{\sum_t r_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T}{\sum_t r_i^t}$$

□ Unsupervised : $X = \{\mathbf{x}^t\}_t$

□ Clusters G_i $i=1, \dots, k$

$$p(\mathbf{x}) = \sum_{i=1}^k p(\mathbf{x} | G_i) P(G_i)$$

where $p(\mathbf{x} | G_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

□ $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^k$

Labels \mathbf{r}_i^t ?

k-Means Clustering

6

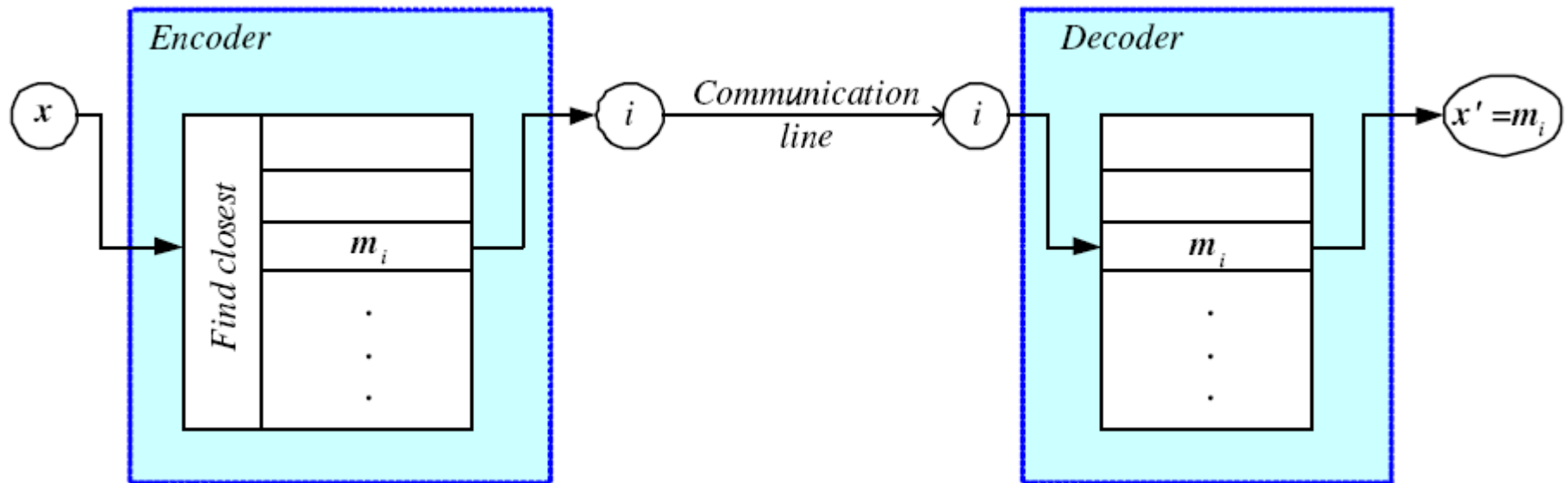
- Find k reference vectors (prototypes/codebook vectors/codewords) which best represent data
- Reference vectors, $\mathbf{m}_j, j = 1, \dots, k$
- Use nearest (most similar) reference:

$$\|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\|$$

- Reconstruction error $E(\{\mathbf{m}_i\}_{i=1}^k | \mathcal{X}) = \sum_t \sum_i b_i^t \|\mathbf{x}^t - \mathbf{m}_i\|$
$$b_i^t = \begin{cases} 1 & \text{if } \|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

Encoding/Decoding

7



k-means Clustering

8

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

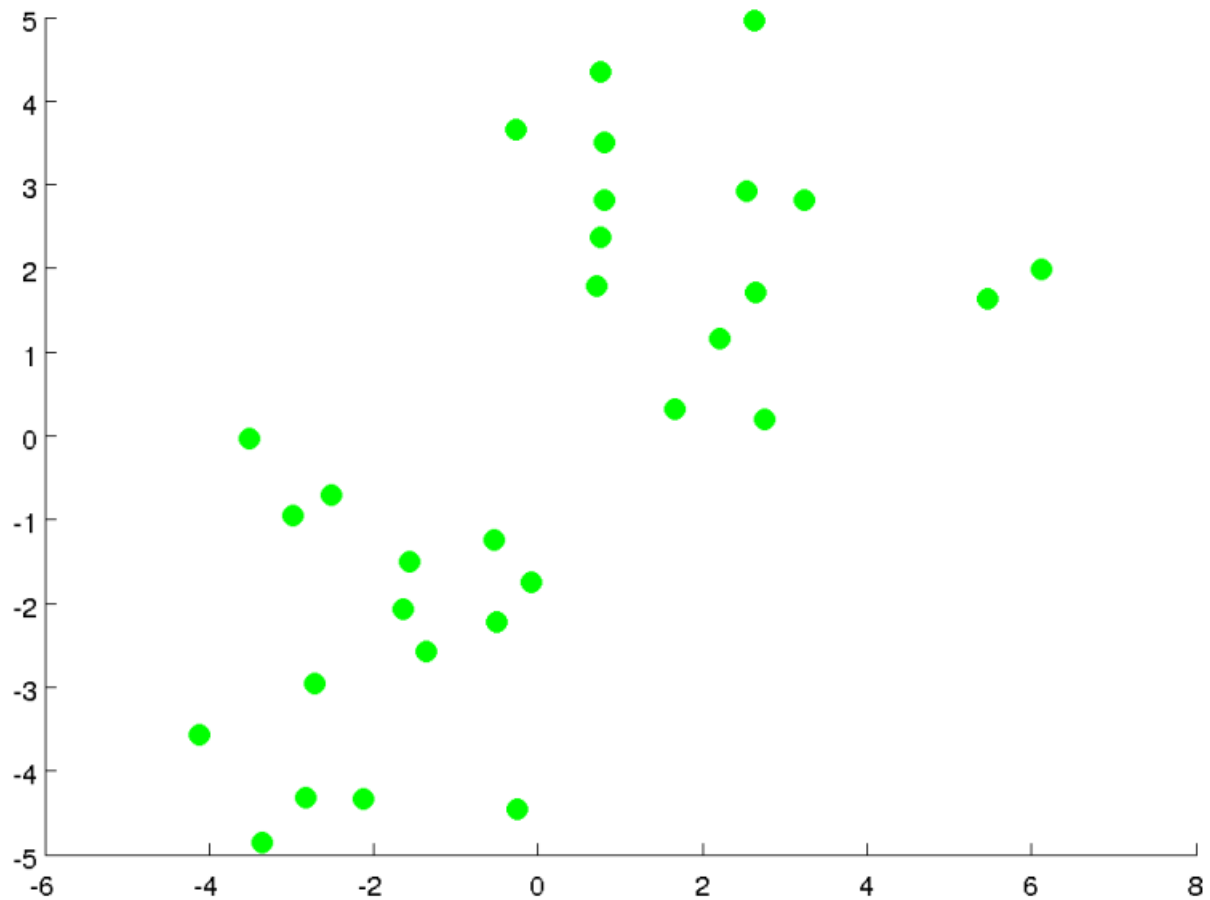
For all $\mathbf{x}^t \in \mathcal{X}$

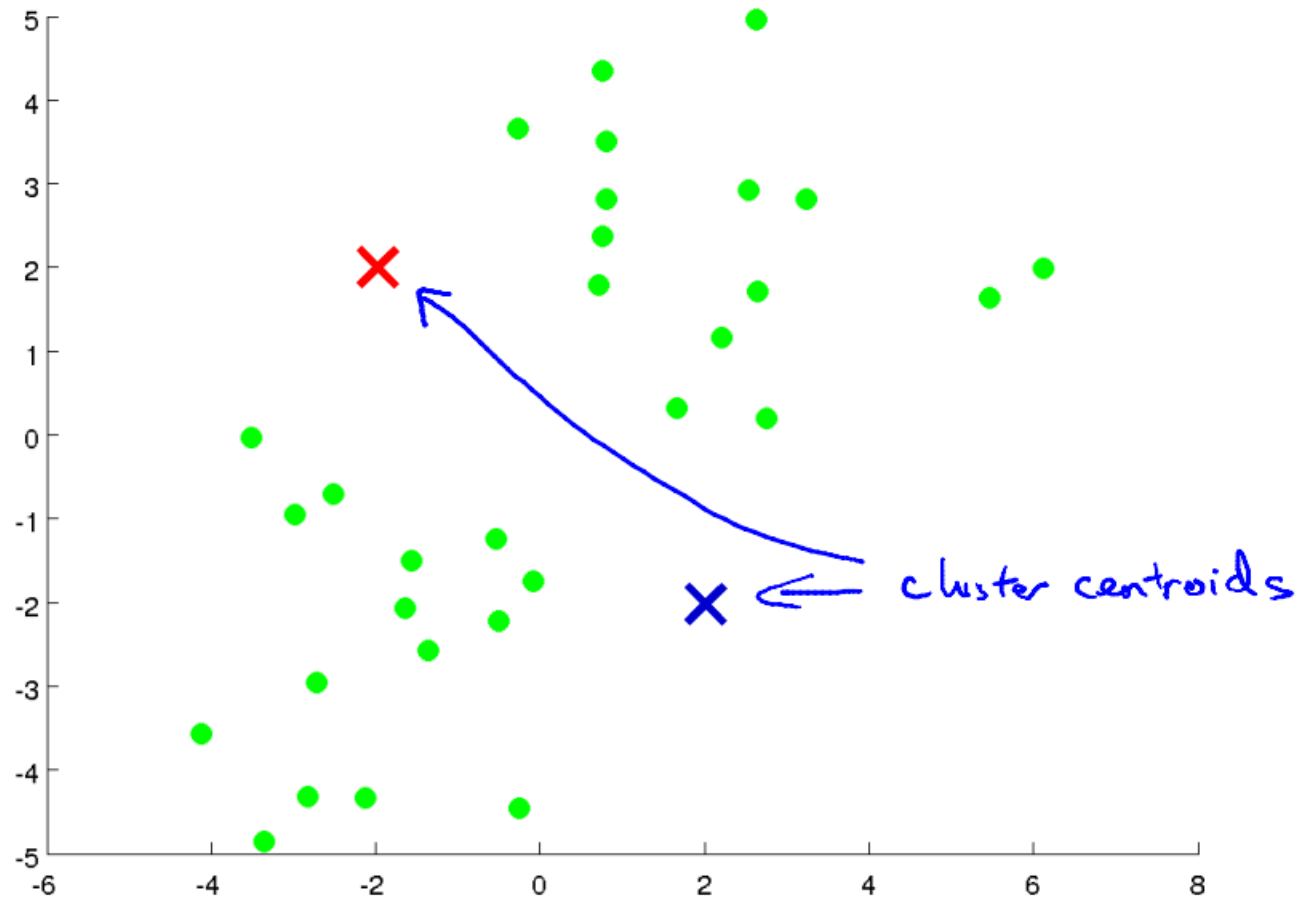
$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

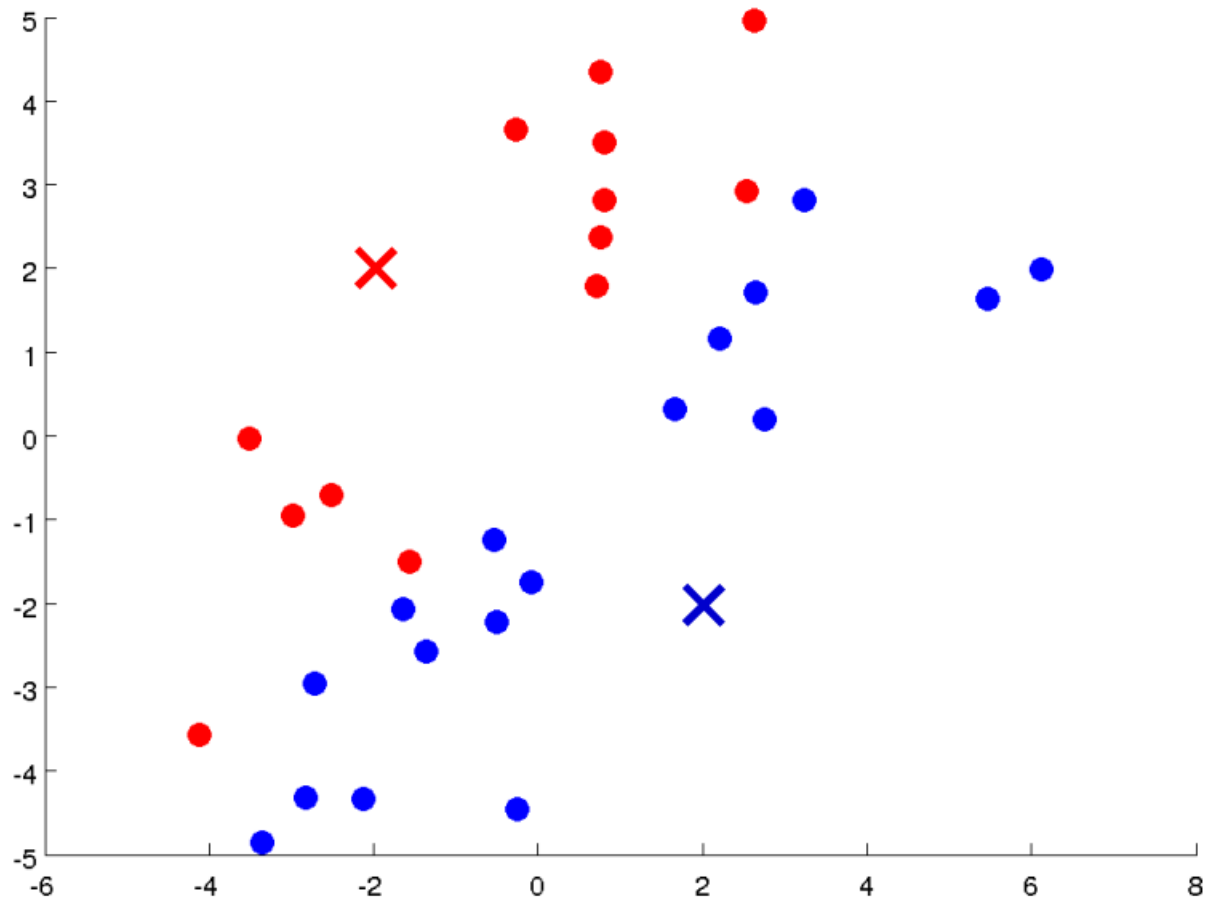
For all $\mathbf{m}_i, i = 1, \dots, k$

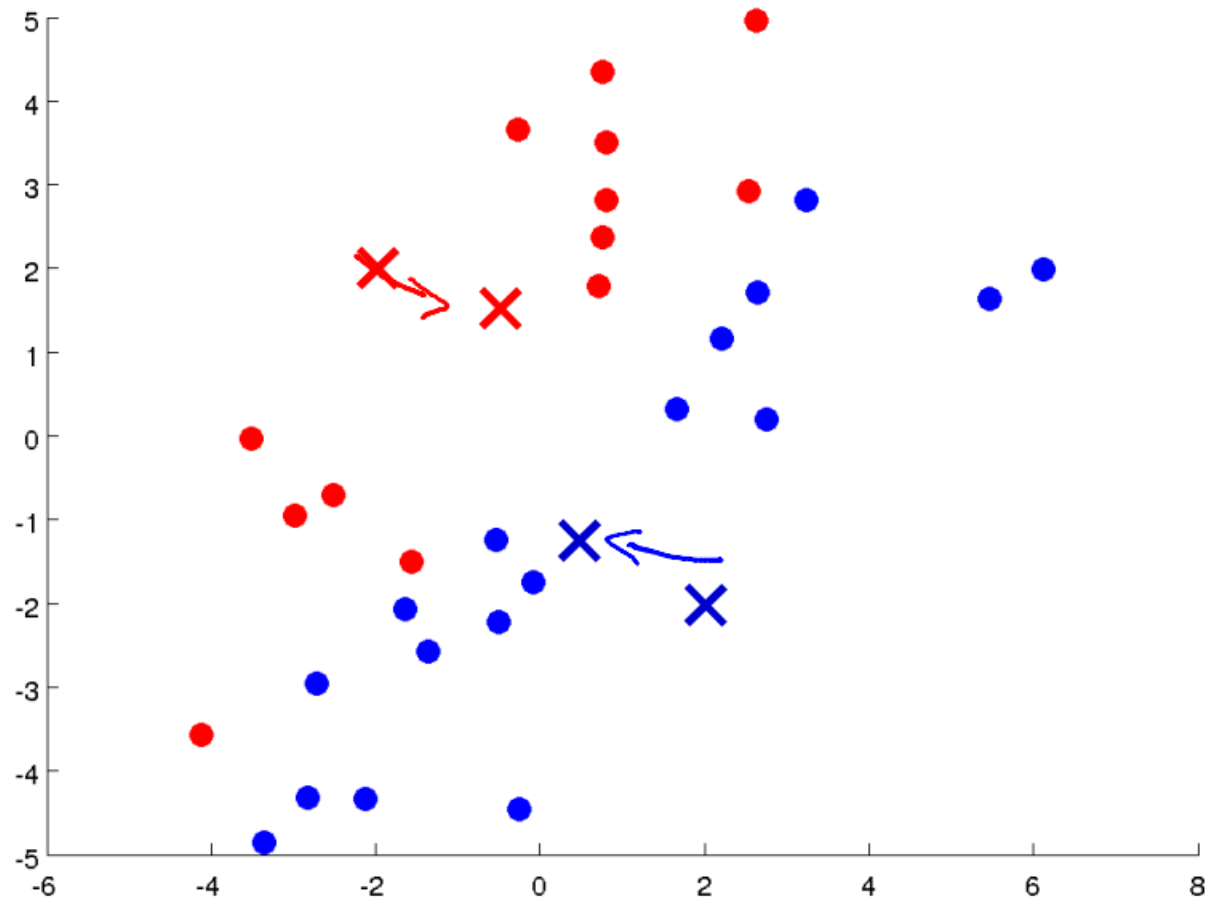
$$\mathbf{m}_i \leftarrow \sum_t b_i^t \mathbf{x}^t / \sum_t b_i^t$$

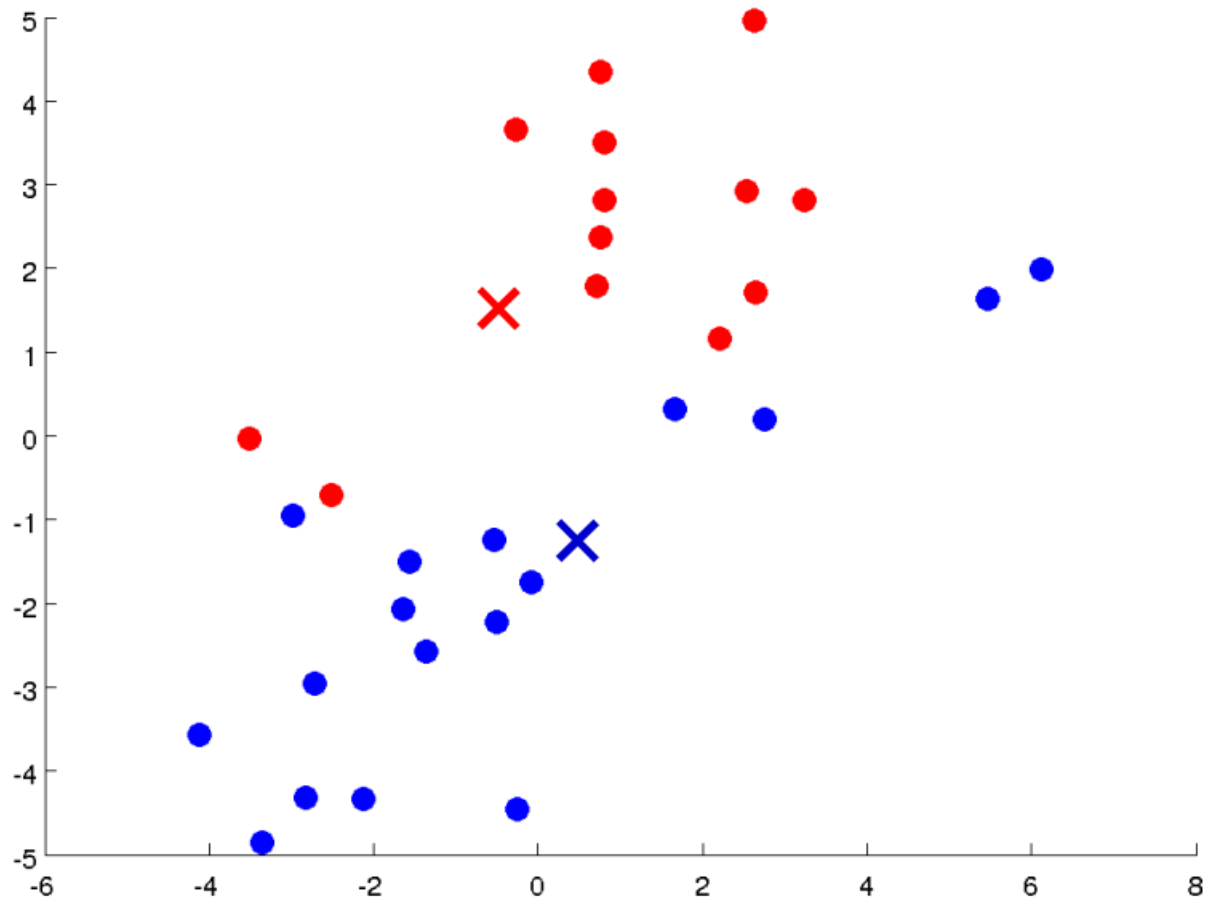
Until \mathbf{m}_i converge

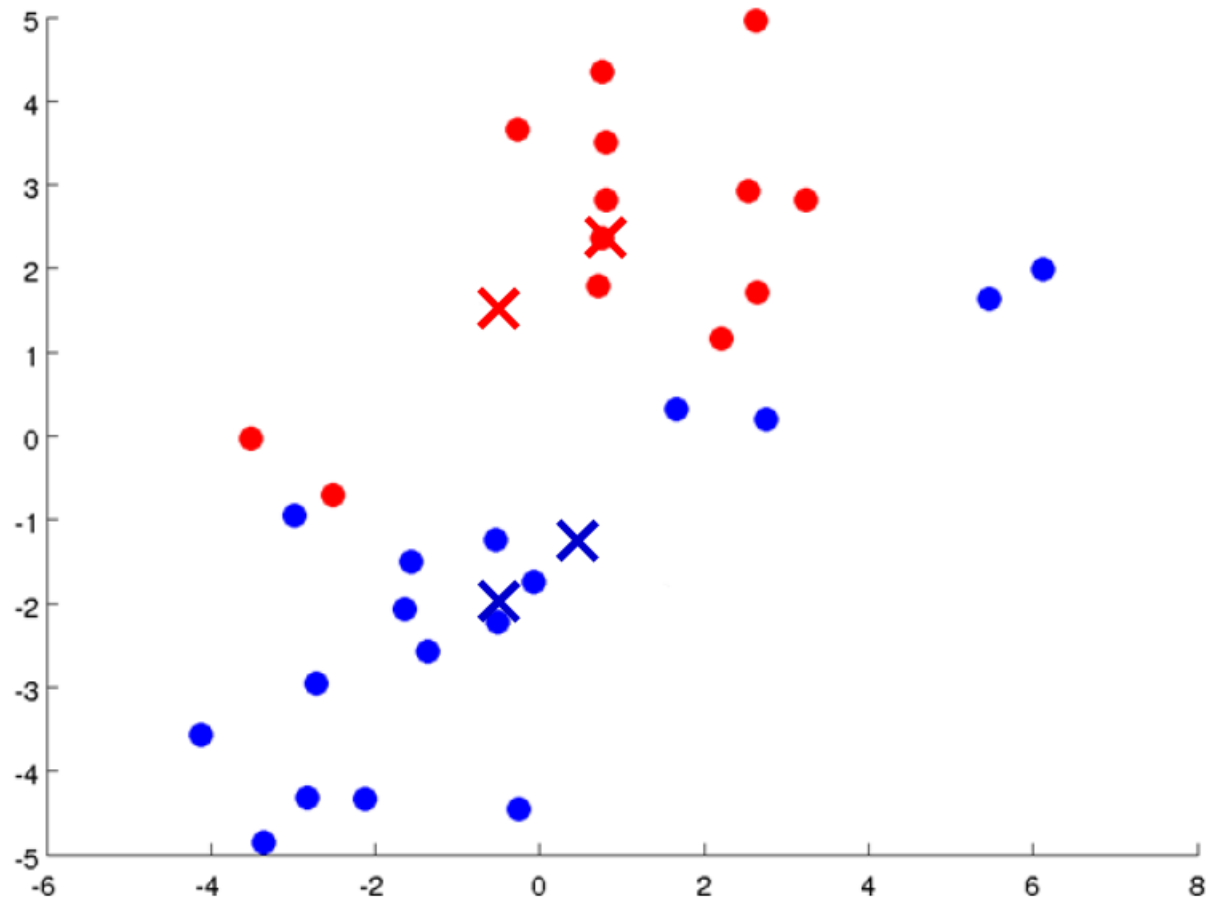


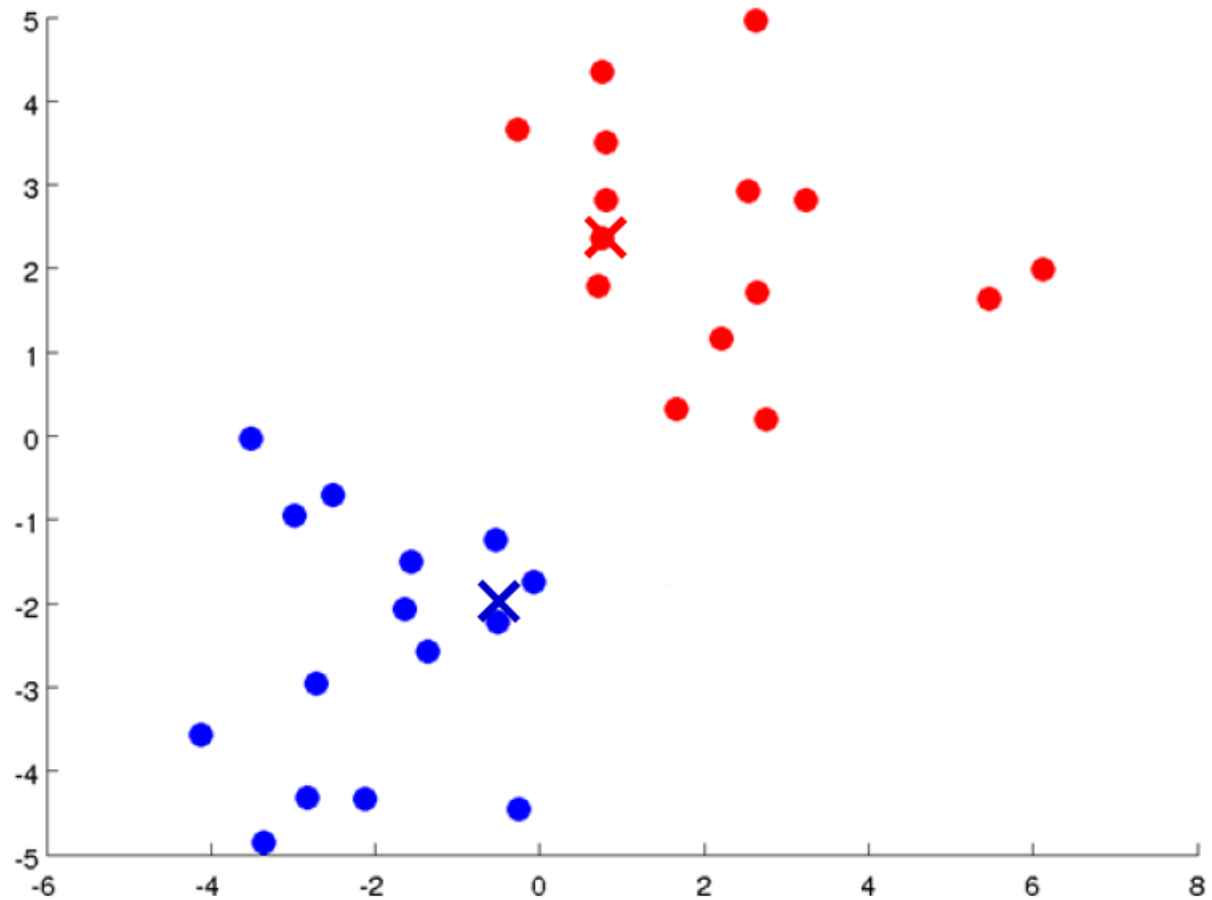


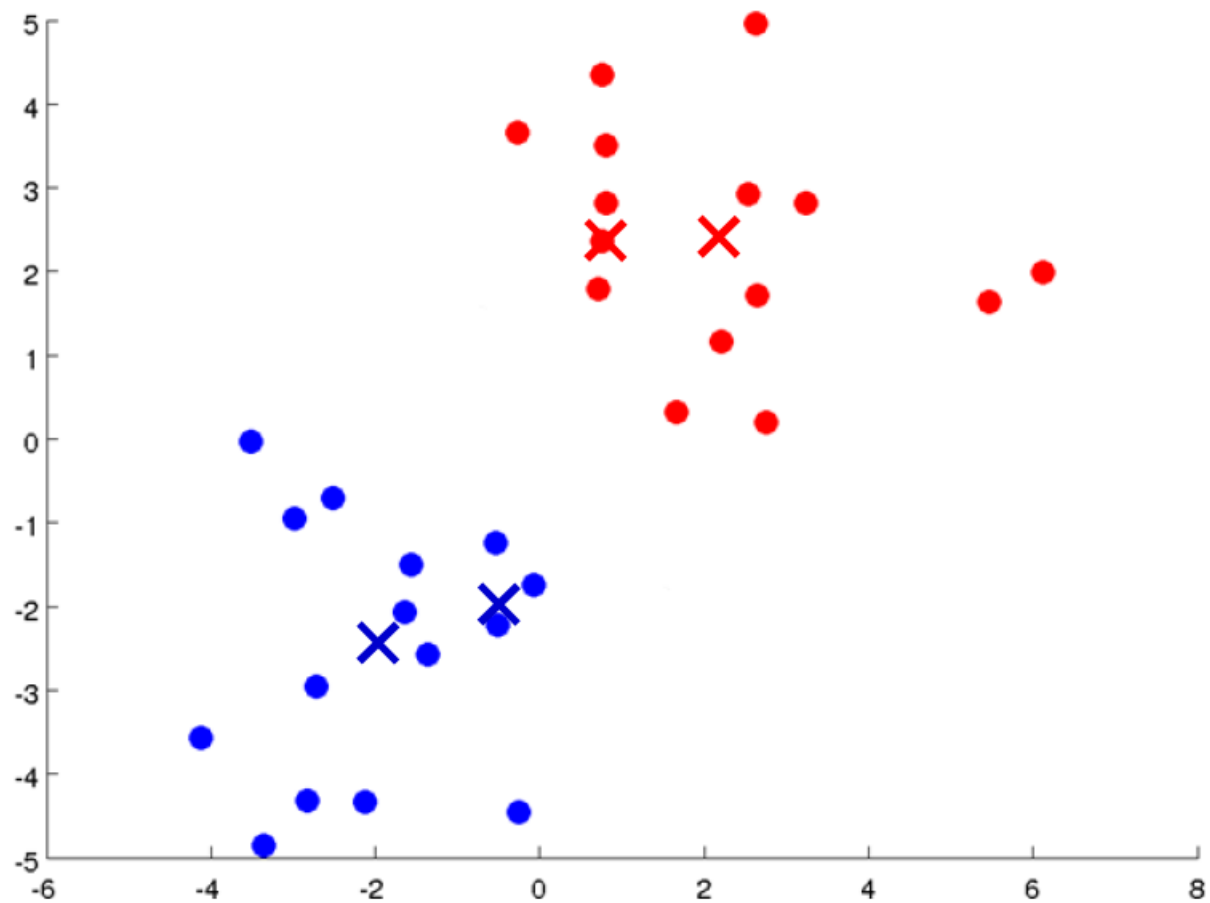


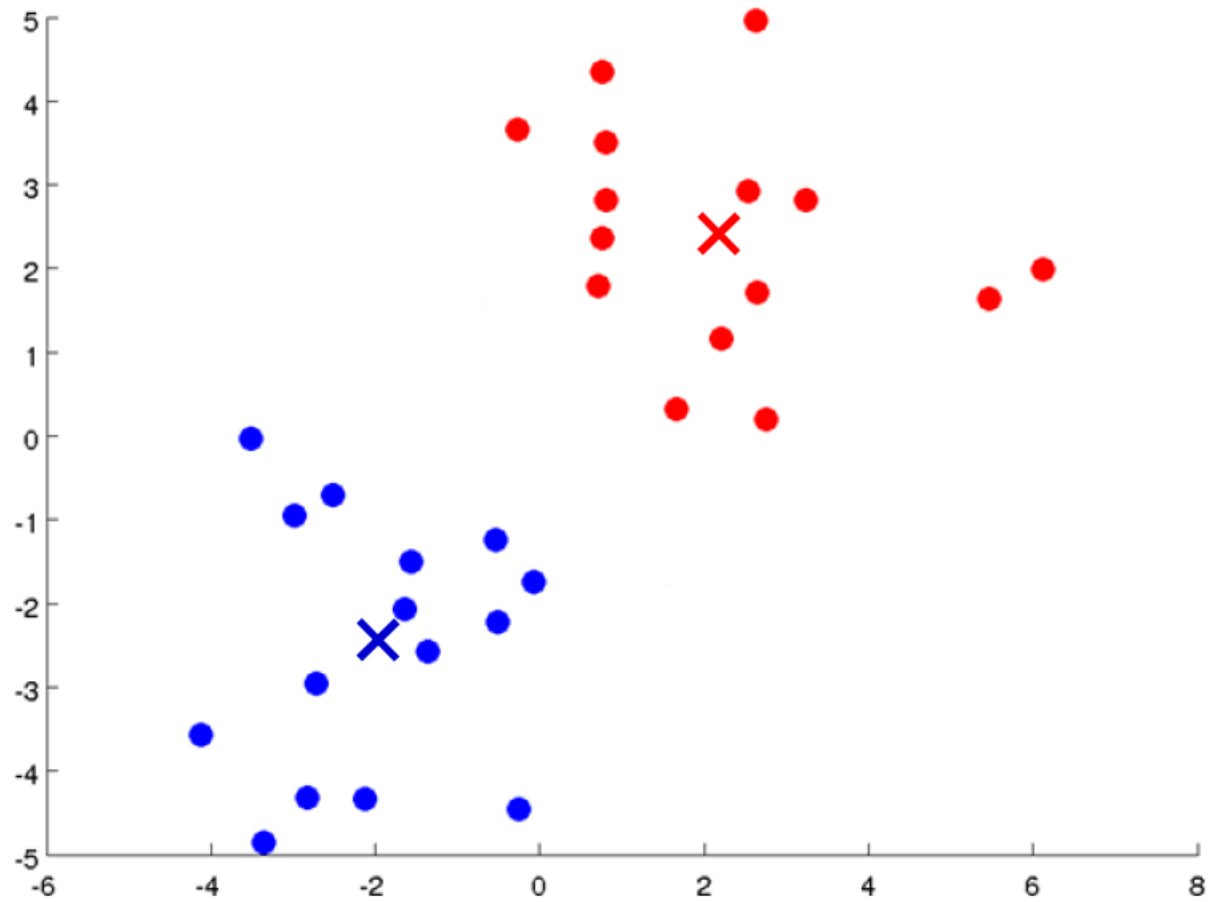


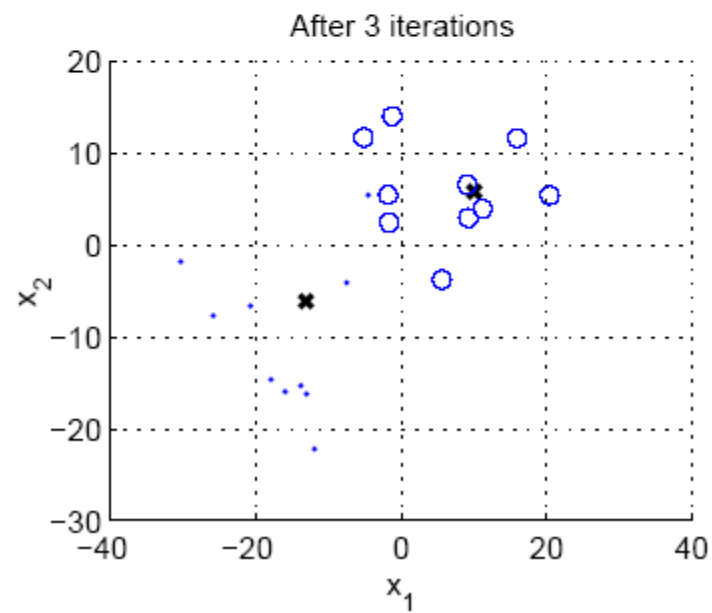
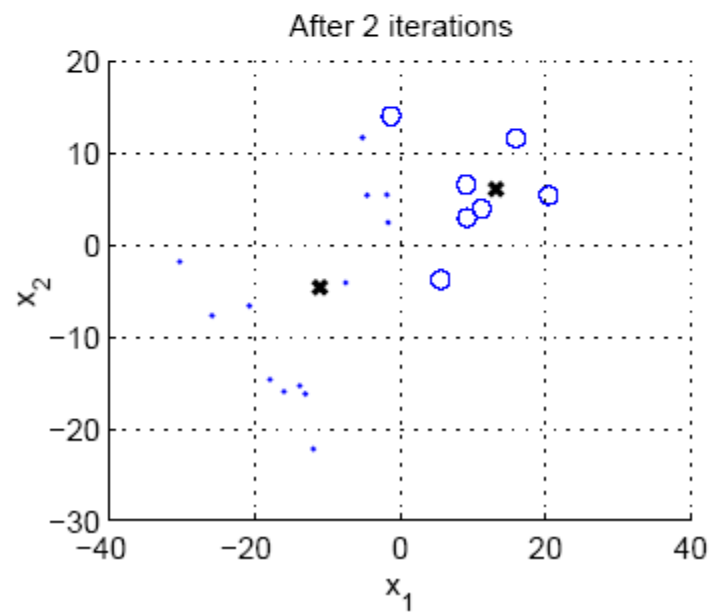
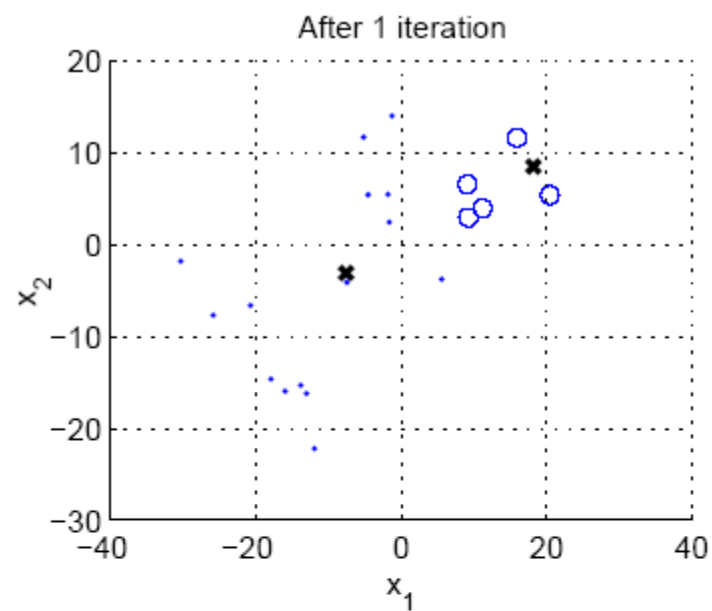
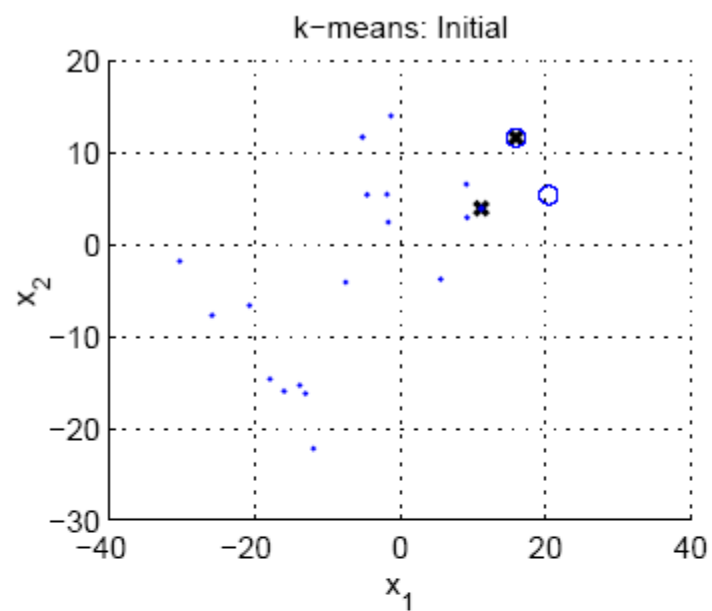












Expectation-Maximization (EM)

19

- Log likelihood with a mixture model

$$\begin{aligned}\mathcal{L}(\Phi | \mathcal{X}) &= \log \prod_t p(\mathbf{x}^t | \Phi) \\ &= \sum_t \log \sum_{i=1}^k p(\mathbf{x}^t | G_i) p(G_i)\end{aligned}$$

- Assume hidden variables z , which when known, make optimization much simpler
- Complete likelihood, $L_c(\Phi | X, Z)$, in terms of \mathbf{x} and \mathbf{z}
- Incomplete likelihood, $L(\Phi | X)$, in terms of \mathbf{x}

E- and M-steps

20

Iterate the two steps

1. E-step: Estimate z given X and current Φ
2. M-step: Find new Φ' given z , X , and old Φ .

$$\text{E-step: } \mathcal{Q}(\Phi | \Phi') = E[\mathcal{L}_c(\Phi | \mathcal{X}, Z) | \mathcal{X}, \Phi']$$

$$\text{M-step: } \Phi'^{+1} = \arg\max_{\Phi} \mathcal{Q}(\Phi | \Phi')$$

An increase in Q increases incomplete likelihood

$$\mathcal{L}(\Phi'^{+1} | \mathcal{X}) \geq \mathcal{L}(\Phi' | \mathcal{X})$$

EM in Gaussian Mixtures

21

- $z_i^t = 1$ if \mathbf{x}^t belongs to G_i , 0 otherwise (labels \mathbf{r}^t_i of supervised learning); assume $p(\mathbf{x}|G_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$

- E-step:
$$E[z_i^t | \mathcal{X}, \Phi'] = \frac{p(\mathbf{x}^t | G_i, \Phi') p(G_i)}{\sum_j p(\mathbf{x}^t | G_j, \Phi') p(G_j)}$$
$$= P(G_i | \mathbf{x}^t, \Phi') \equiv h_i^t$$

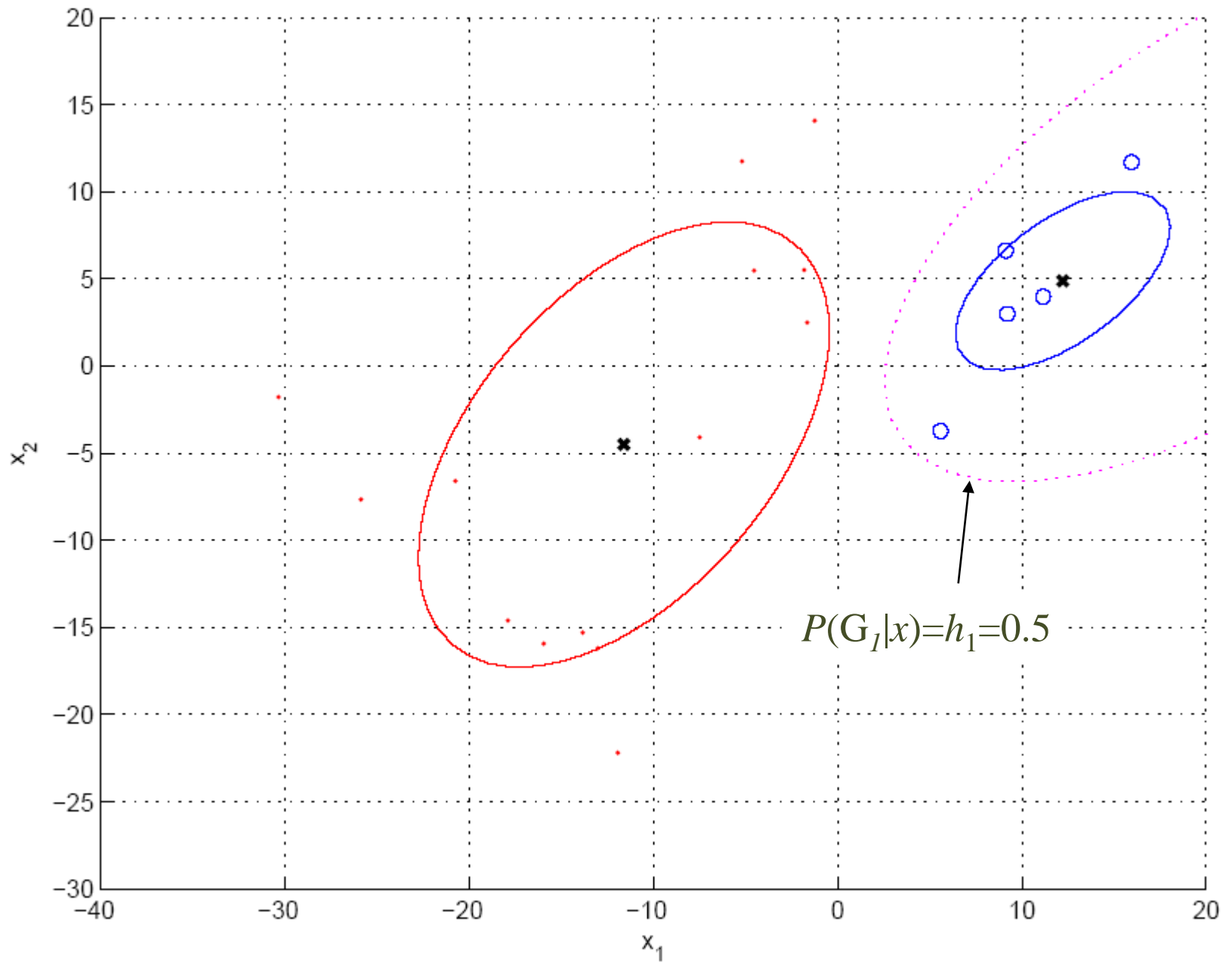
- M-step:

$$P(G_i) = \frac{\sum_t h_i^t}{N} \quad \mathbf{m}_i^{l+1} = \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t}$$

$$\mathbf{S}_i^{l+1} = \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i^{l+1})(\mathbf{x}^t - \mathbf{m}_i^{l+1})^T}{\sum_t h_i^t}$$

Use estimated labels in place of unknown labels

EM solution



Mixtures of Latent Variable Models

23

Regularize clusters

1. Assume shared/diagonal covariance matrices
2. Use PCA/FA to decrease dimensionality: Mixtures of PCA/FA

$$p(\mathbf{x}_t | G_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{V}_i \mathbf{V}_i^T + \boldsymbol{\Psi}_i)$$

Can use EM to learn \mathbf{V}_i (Ghahramani and Hinton, 1997; Tipping and Bishop, 1999)

After Clustering

24

- Dimensionality reduction methods find correlations between features and group features
- Clustering methods find similarities between instances and group instances
- Allows knowledge extraction through
 - number of clusters,
 - prior probabilities,
 - cluster parameters, i.e., center, range of features.

Example: CRM, customer segmentation

Clustering as Preprocessing

25

- Estimated group labels h_j (soft) or b_j (hard) may be seen as the dimensions of a new k dimensional space, where we can then learn our discriminant or regressor.
- **Local** representation (only one b_j is 1, all others are 0; only few h_j are nonzero) vs **Distributed** representation (After PCA; all z_j are nonzero)

Mixture of Mixtures

26

- In classification, the input comes from a mixture of classes (supervised).
- If each class is also a mixture, e.g., of Gaussians, (unsupervised), we have a mixture of mixtures:

$$p(\mathbf{x} | C_i) = \sum_{j=1}^{k_i} p(\mathbf{x} | G_{ij}) P(G_{ij})$$

$$p(\mathbf{x}) = \sum_{i=1}^K p(\mathbf{x} | C_i) P(C_i)$$

Spectral Clustering

27

- Cluster using predefined pairwise similarities B_{rs} instead of using Euclidean or Mahalanobis distance
- Can be used even if instances not vectorially represented
- Steps:
 - I. Use Laplacian Eigenmaps (chapter 6) to map to a new \mathbf{z} space using B_{rs}
 - II. Use k -means in this new \mathbf{z} space for clustering

Hierarchical Clustering

28

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

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

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

City-block distance

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

Agglomerative Clustering

29

- Start with N groups each with one instance and merge two closest groups at each iteration
- Distance between two groups G_i and G_j :

- Single-link:
$$d(G_i, G_j) = \min_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

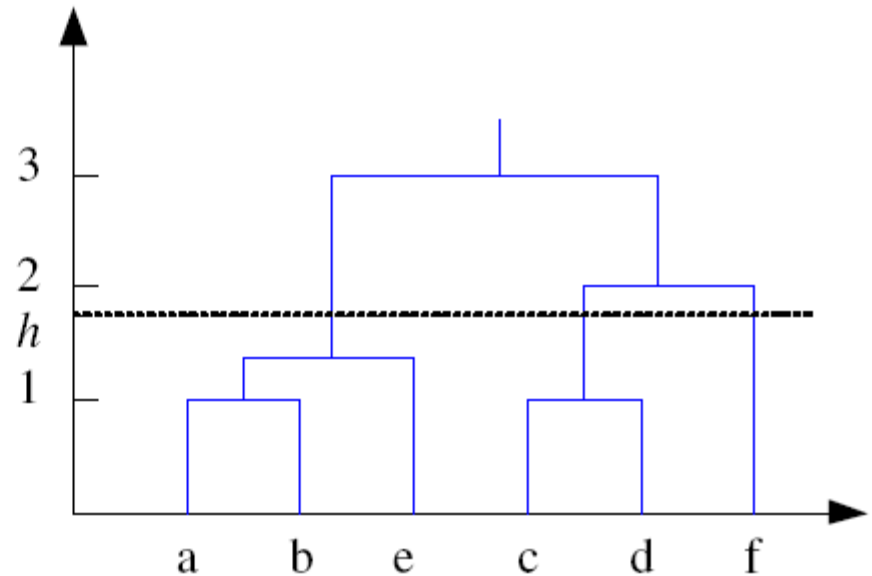
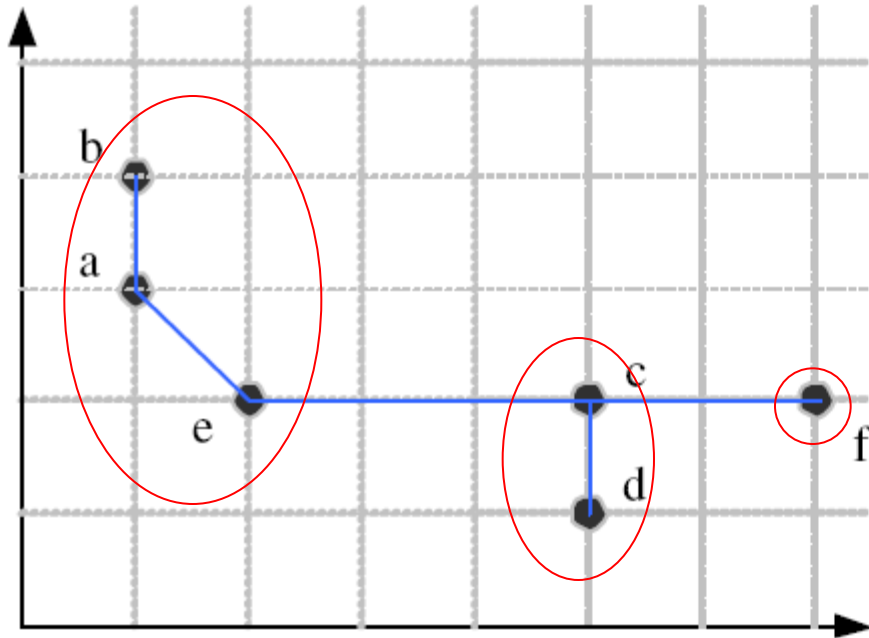
- Complete-link:
$$d(G_i, G_j) = \max_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

- Average-link, centroid

$$d(G_i, G_j) = \text{ave}_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

Example: Single-Link Clustering

30



Dendrogram

Choosing k

31

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