Intro to ML

October 27th, 2021

CHAPTER 7:

Clustering

Classes vs. Clusters

- Supervised: $X = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_t$
- Classes C_i $i=1,...,K$

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

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

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

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

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

- Unsupervised : $X = \{\boldsymbol{x}^t\}_t$
- Clusters G_i $i=1,...,k$

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

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

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

Labels $\boldsymbol{r}^t_i$?

Through clustering

Imagine a case

- A image, 24 bits/pixel, ~ 16 million colors
- Say we have a screen with 8 bits/pixel
    - 256 colors only

- Find the best colors among 16 million colors so that the image look as close to the original image as possible -> color quantization

- Vector quantization ->continuous value to discrete space

k-Means Clustering

- A vector quantization method
- Find k reference vectors (prototypes/codebook vectors/codewords) which best represent data
- Reference vectors, $\boldsymbol{m}_j$, j =1,…,k
- Use nearest (most similar) reference:

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

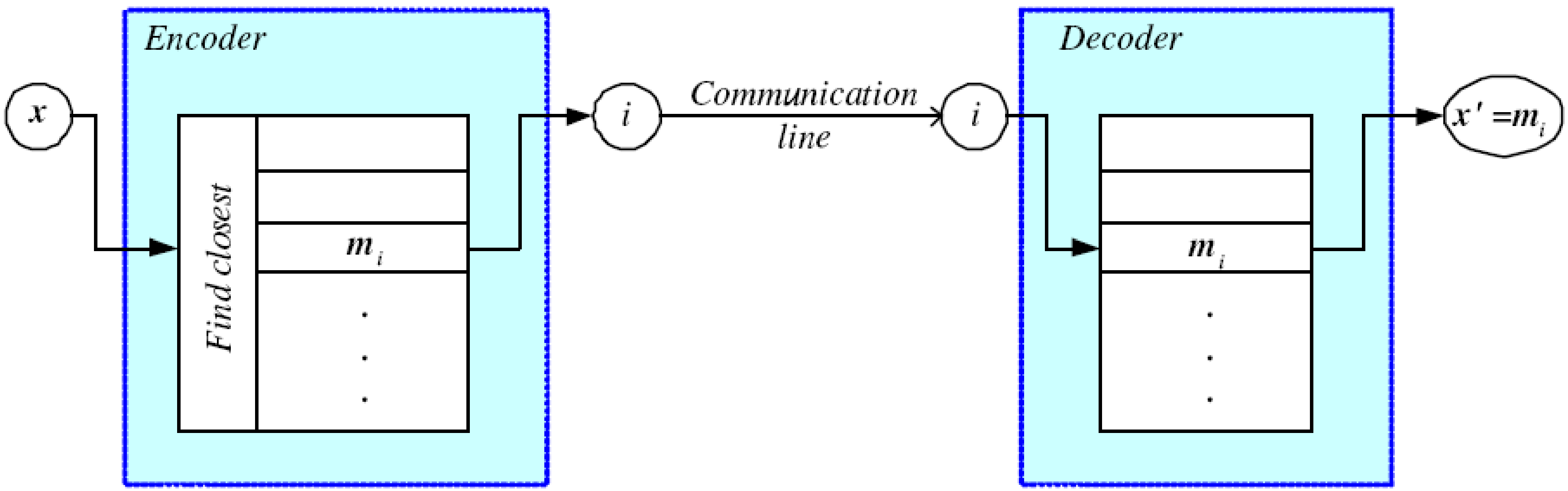
find a reference vector that is close to each of x^t

- Reconstruction error

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

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

Encoding/Decoding
Vector quantization



Vector quantization saves bit, at the receiving end, there is an error
Quantization can be imaged as clustering

k-means Clustering

No analytical solution, since it's about finding that reference vector m_i
Results in an **iterative** approach

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

Repeat

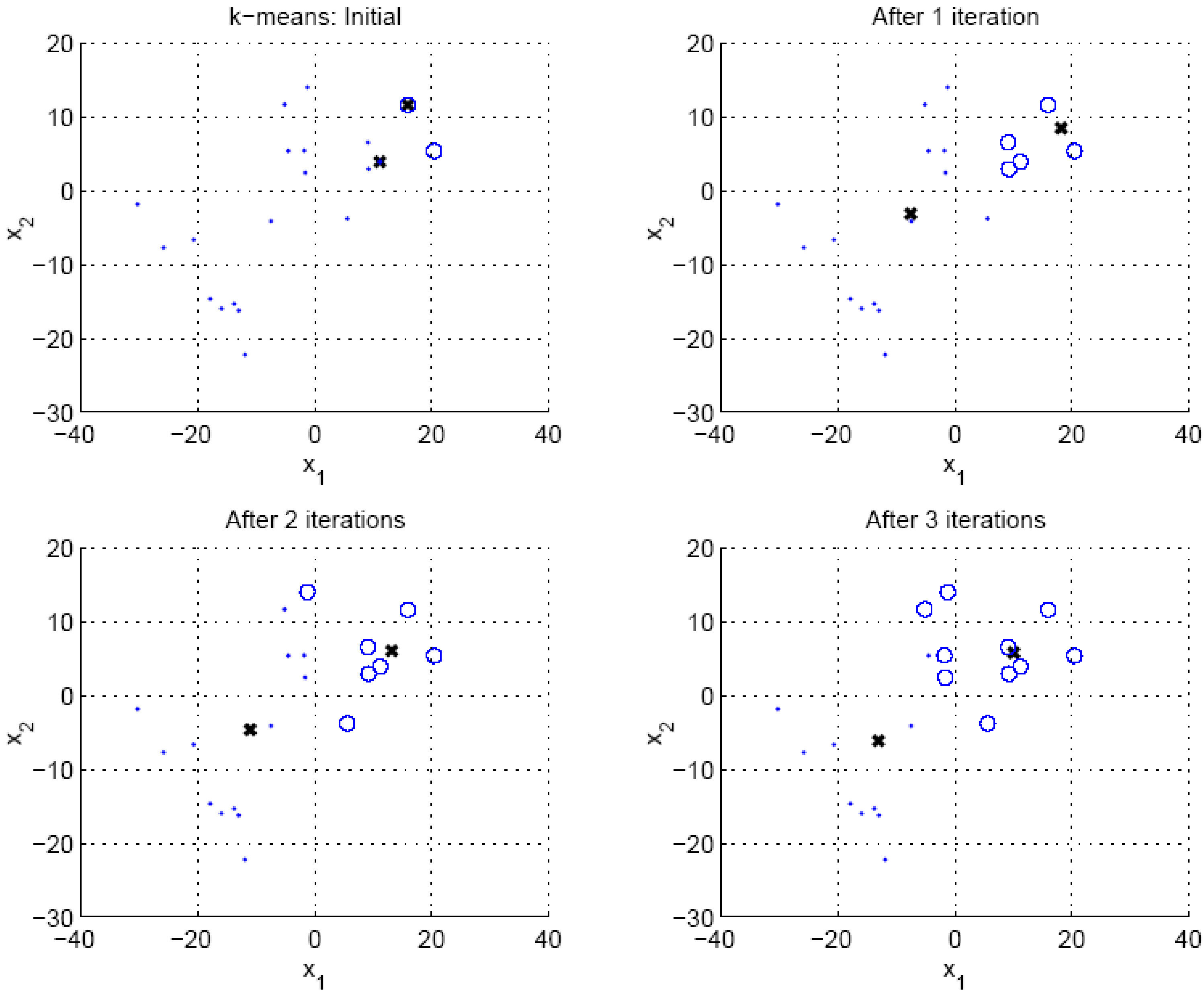
For all $\boldsymbol{x}^t \in \mathcal{X}$

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

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



K-means aim at finding k codebook that minimizes reconstruction error

Mixture model

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

where G_i the components/groups/clusters,

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

$p(\boldsymbol{x} \mid G_i)$ component densities

If known, the data sample can find it's associated 'components/ groups /cluster'

Expectation-Maximization (EM)

- Log likelihood of a mixture model

$$\mathcal{L}(\Phi \mid \mathcal{X}) = \log \prod_t p\left(\mathbf{x}^t \mid \Phi\right)$$

$$= \sum_t \log \sum_{i=1}^{k} p\left(\mathbf{x}^t \mid G_i\right) P\left(G_i\right)$$

Unknown, (we don't know which cluster the sample belongs to)

No analytical solution when learning this model

EM Algorithm, core concept

- Assume there exist hidden variables z, which when known, make optimization much simpler
- Complete likelihood, $L_C(\Phi|X,Z)$, in terms of ***x*** and ***z***
- Incomplete likelihood, $L(\Phi|X)$, in terms of ***x***

E- and M-steps

Model parameter

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.

$$\text{E-step}: \mathcal{Q}\left(\Phi \mid \Phi^{l}\right) = E\left[\mathcal{L}_C(\Phi \mid \mathcal{X}, \mathcal{Z}) \mid \mathcal{X}, \Phi^{l}\right]$$

$$\text{M-step}: \Phi^{l+1} = \arg\max_{\Phi} \mathcal{Q}\left(\Phi \mid \Phi^{l}\right)$$

An increase in Q function increases incomplete likelihood

$$\mathcal{L}\left(\Phi^{l+1} \mid \mathcal{X}\right) \geq \mathcal{L}\left(\Phi^{l} \mid \mathcal{X}\right)$$

There is proof beyond this class

EM in Gaussian Mixtures

Mixture of Guassian distribution

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

- $z^t_i = 1$ if $\boldsymbol{x}^t$ belongs to G_i, 0 otherwise; assume $p(\boldsymbol{x}|G_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

- E-step
  - Expectation, find the expected value of the current model parameters under the current parameter set
- M-step
  - Maximize these parameters, and iterate

z^t is indicator variable $(z_1^t, z_2^t \dots z_k^t)$ $z_i^t=1$ when x^t = belong to cluster G_i
z multinomial distribution
Prior distribution z_i is π_i

$$P(\mathbf{z}^t) = \prod_{i=1}^{k} \pi_i^{z_i^t} \tag{7.7}$$

The likelihood of an observation $\boldsymbol{x}^t$ is equal to its probability specified by the component that generated it:

$$p(\mathbf{x}^t|\mathbf{z}^t) = \prod_{i=1}^{k} p_i(\mathbf{x}^t)^{z_i^t} \tag{7.8}$$

$p_i(\mathbf{x}^t)$ is shorthand for $p(\mathbf{x}^t|\mathcal{G}_i)$. The joint density is

$$p(\mathbf{x}^t, \mathbf{z}^t) = P(\mathbf{z}^t)p(\mathbf{x}^t|\mathbf{z}^t)$$

and the complete data likelihood of the iid sample $\mathcal{X}$ is

$$\begin{aligned}
\mathcal{L}_c(\Phi|\mathcal{X}, \mathcal{Z}) &= \log \prod_t p(\mathbf{x}^t, \mathbf{z}^t|\Phi) \\
&= \sum_t \log p(\mathbf{x}^t, \mathbf{z}^t|\Phi) \\
&= \sum_t \log P(\mathbf{z}^t|\Phi) + \log p(\mathbf{x}^t|\mathbf{z}^t, \Phi) \\
&= \sum_t \sum_i z_i^t[\log \pi_i + \log p_i(\mathbf{x}^t|\Phi)]
\end{aligned}$$

$t \quad i$

E-step: We define

$$
\begin{aligned}
\mathcal{Q}(\Phi|\Phi^l) &\equiv E\left[\log P(X,Z)|\mathcal{X},\Phi^l\right] \\
&= E\left[\mathcal{L}_c(\Phi|\mathcal{X},\mathcal{Z})|\mathcal{X},\Phi^l\right] \\
&= \sum_t \sum_i E[z_i^t|\mathcal{X},\Phi^l][\log \pi_i + \log p_i(\mathbf{x}^t|\Phi)]
\end{aligned}
$$

E[X] = $\sum xp(x)$

where

$$
\begin{aligned}
E[z_i^t|\mathcal{X},\Phi^l] &= E[z_i^t|\mathbf{x}^t,\Phi^l] \quad \mathbf{x}^t \text{ are iid} \\
&= P(z_i^t = 1|\mathbf{x}^t,\Phi^l) \quad z_i^t \text{ is a 0/1 random variable} \\
&= \frac{p(\mathbf{x}^t|z_i^t = 1,\Phi^l)P(z_i^t = 1|\Phi^l)}{p(\mathbf{x}^t|\Phi^l)} \quad \text{Bayes' rule} \\
&= \frac{p_i(\mathbf{x}^t|\Phi^l)\pi_i^l}{\sum_j p_j(\mathbf{x}^t|\Phi^l)\pi_j^l} \\
&= \frac{p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)P(\mathcal{G}_i)}{\sum_j p(\mathbf{x}^t|\mathcal{G}_j,\Phi^l)P(\mathcal{G}_j)} \\
&= P(\mathcal{G}_i|\mathbf{x}^t,\Phi^l) \equiv h_i^t
\end{aligned}
\qquad (7.9)
$$

- E-step:
  - Expected value is the posterior probability of the sample coming from that mixture (cluster)
  - It’s between 0-1
  - We can also think about it as soft assignment of cluster for each sample

  - In E step, given data X, compute the complete data likelihood given the current model parameters

M-step: We maximize $\mathcal{Q}$ to get the next set of parameter values Φ^{l+1}:

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

which is

$$\begin{aligned}\mathcal{Q}(\Phi|\Phi^l) &= \sum_t \sum_i h_i^t [\log \pi_i + \log p_i(\mathbf{x}^t|\Phi)] \\ &= \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p_i(\mathbf{x}^t|\Phi)\end{aligned}$$

The second term is independent of π_i and using the constraint that $\sum_i \pi_i = 1$ as the Lagrangian, we solve for

$$\nabla_{\pi_i} \sum_t \sum_i h_i^t \log \pi_i - \lambda \left(\sum_i \pi_i - 1 \right) = 0$$

and get

$$\pi_i^{l+1} = \frac{\sum_t h_i^t}{N}$$

which is analogous to the calculation of priors in equation 7.2.

Similarly, the first term of equation 7.10 is independent of the components and can be dropped while estimating the parameters of the components. We solve for

$$\nabla_{\Phi} \sum_t \sum_i h_i^t \log p_i(\mathbf{x}^t|\Phi) = 0$$

Complete steps for GMM

- If assume Gaussian component (each mixture is a Gaussian distribution)

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

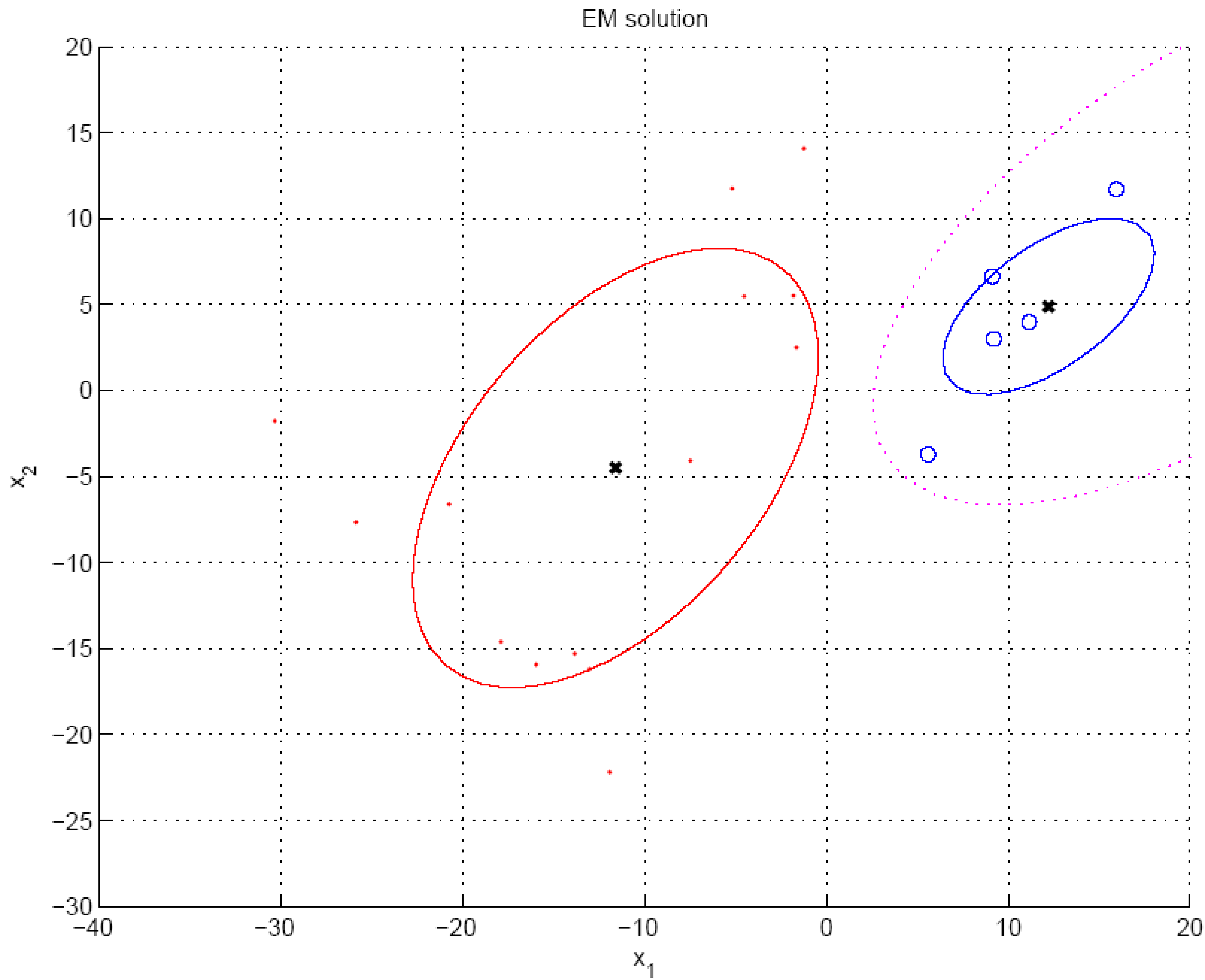
$$\mathbf{S}_i^{l+1} = \frac{\sum_t h_i^t \left(\mathbf{x}^t - \mathbf{m}_i^{l+1}\right)\left(\mathbf{x}^t - \mathbf{m}_i^{l+1}\right)^T}{\sum_t h_i^t}$$

Soft assignment of a sample to a class

$$h_i^t = \frac{\pi_i |\mathbf{S}_i|^{-1/2} \exp[-(1/2)(\boldsymbol{x}^t - \boldsymbol{m}_i)^T \mathbf{S}_i^{-1} (\boldsymbol{x}^t - \boldsymbol{m}_i)]}{\sum_j \pi_j |\mathbf{S}_j|^{-1/2} \exp[-(1/2)(\boldsymbol{x}^t - \boldsymbol{m}_j)^T \mathbf{S}_j^{-1} (\boldsymbol{x}^t - \boldsymbol{m}_j)]}$$

Practice implementation of GMM

- EM is initialized by k-means -> so you get initial parameter
- Once done k-mean, use m_i and samples associated with each cluster as to estimate the initial parameters used for mixture of Gaussian distributions
- Once done-learning, the GMM model can be used for 'clustering' of samples x^t (compute h_i^t)


EM solution
20
15
10
5
0
−5
−10
−15
−20
−25
−30
−40
−30
−20
−10
0
10
20
x1
x2

After Clustering

- Dimensionality reduction methods **find correlations between <u>features</u> and group <u>features</u>**
- Clustering methods find **similarities between <u>instances</u> and group <u>instances</u>**
- Allows knowledge extraction through

    number of clusters,

    prior probabilities,

    cluster parameters, i.e., center, range of features.

Example: CRM, customer segmentation

Clustering as Preprocessing

- 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

- 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:

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

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