Machine Learning

8. Expectation maximization

Yannick Le Cacheux

CentraleSupélec - Université Paris Saclay

September 2024

Outline

The 1D normal distribution

- Probability density function (p.d.f.) for a Gaussian with mean μ and standard deviation σ:

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

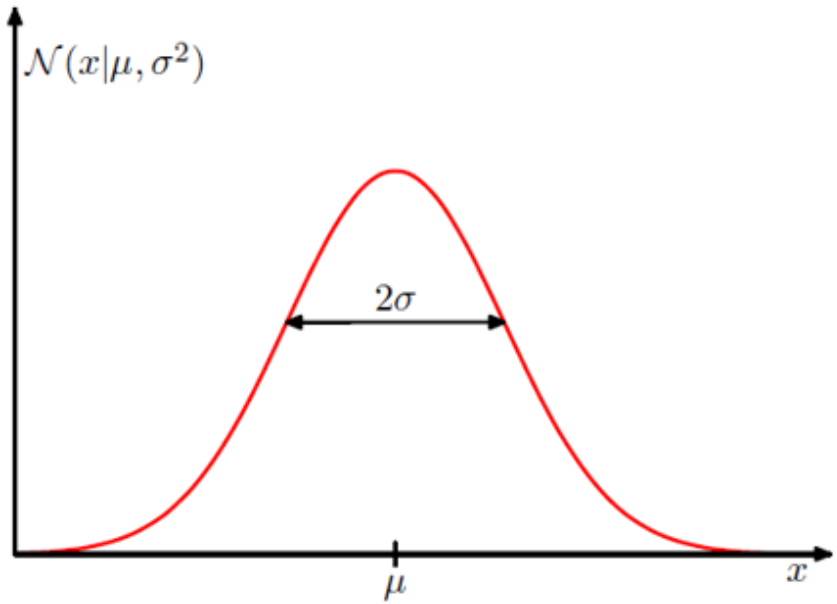


Figure: Gaussian probability density function[1]

[1] From C. Bishop, Patter Recognition and Machine Learning, as are all figures in this section.

Likelihood of a Gaussian

- For a dataset of N identically and independently distributed (i.i.d.) observations $\mathbf{x} = (x_1, \ldots, x_N)^\top$, the likelihood is given by

$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \sigma^2)$$

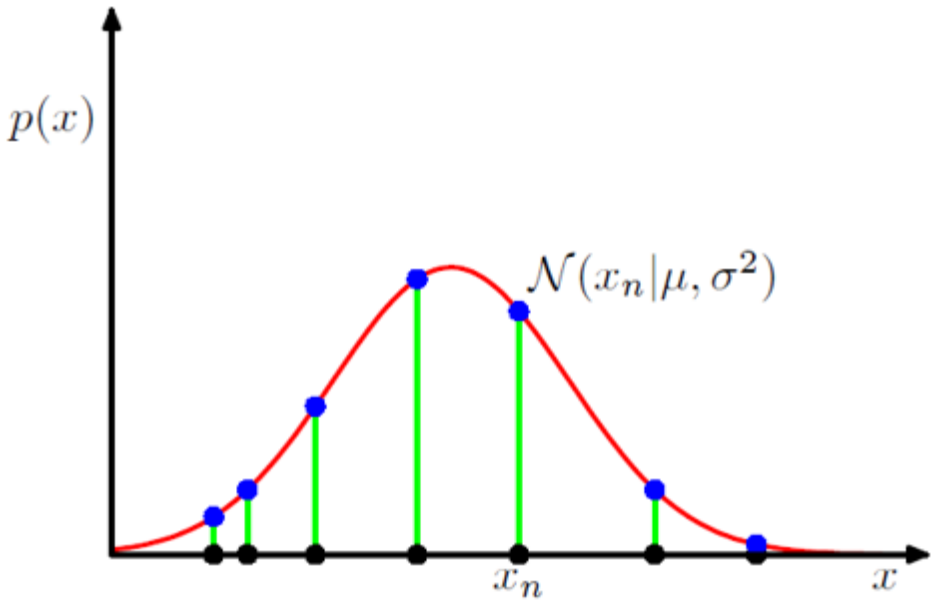


Figure: Likelihood of different samples x_n

Maximum likelihood estimation

- The log-likelihood is

$$\log p(\mathbf{x}|\mu, \sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

- Maximizing this quantity w.r.t. μ by setting $\dfrac{\partial \ln p(\mathbf{x}|\mu, \sigma^2)}{\partial \mu}$ to 0, we get the maximum likelihood estimator of μ:

$$\hat{\mu}_{\mathsf{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

- Similarly, the maximum likelihood estimator of σ^2 is

$$\hat{\sigma^2}_{\mathsf{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

The D-dimensional normal distribution

- The p.d.f. of a D-dimensional Gaussian with mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ is

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

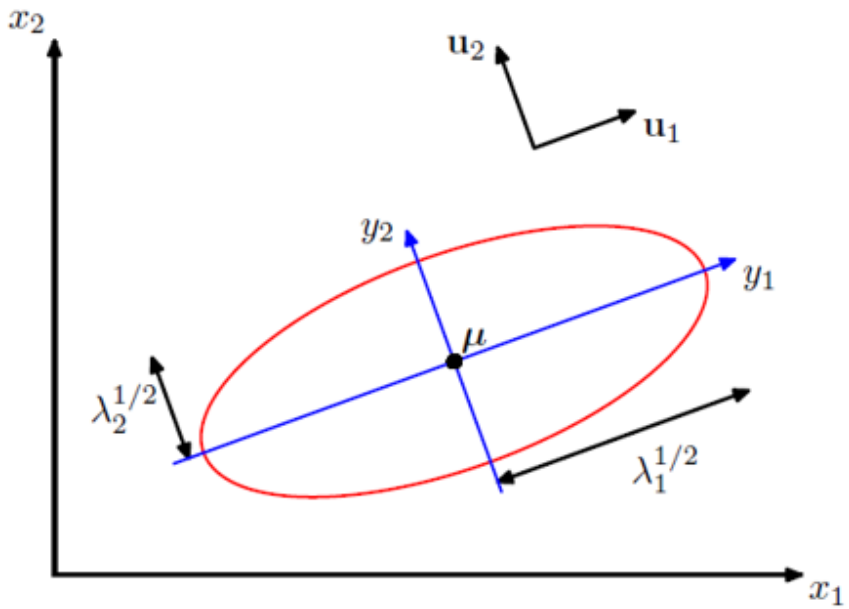
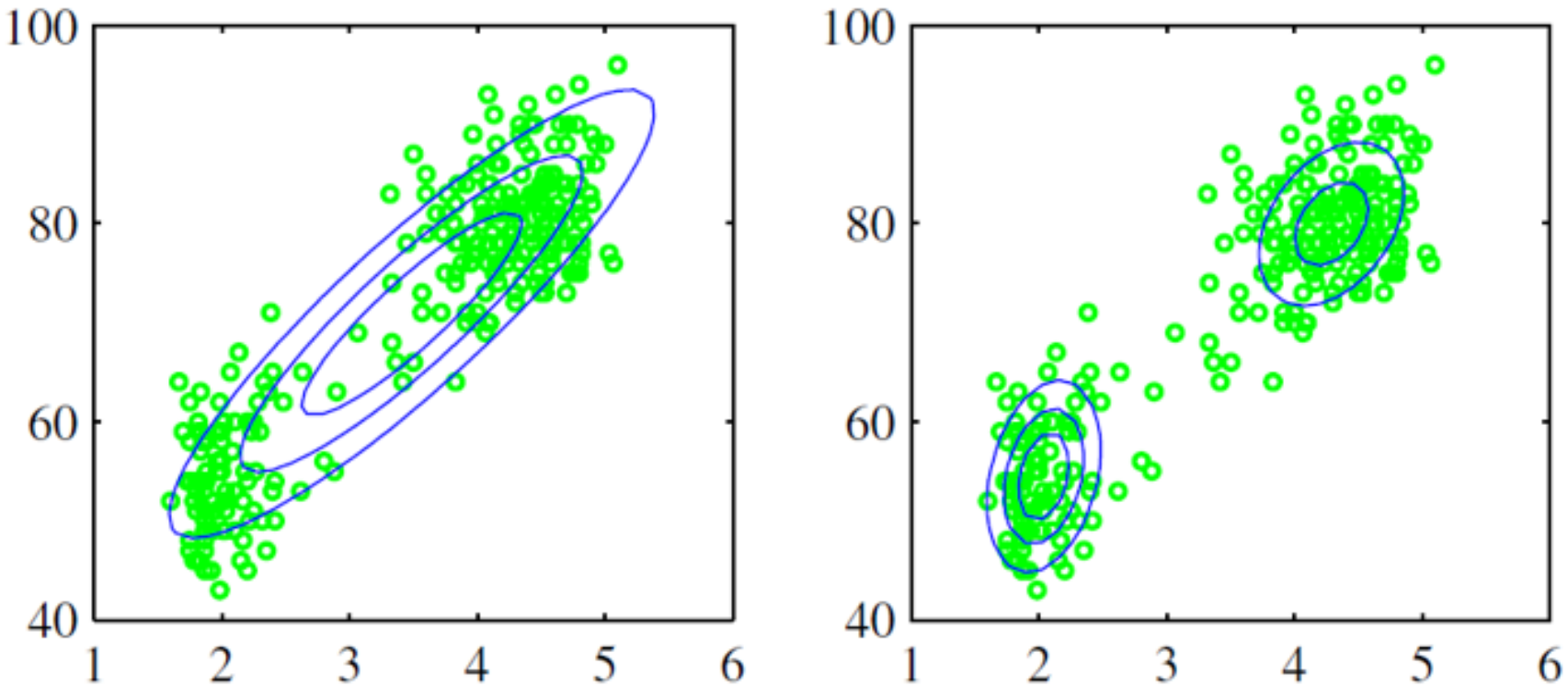


Figure: 2D Gaussian with principal components $\mathbf{u}_1$ and $\mathbf{u}_2$, and eigenvalues λ_1 and λ_2

Mixtures of Gaussians

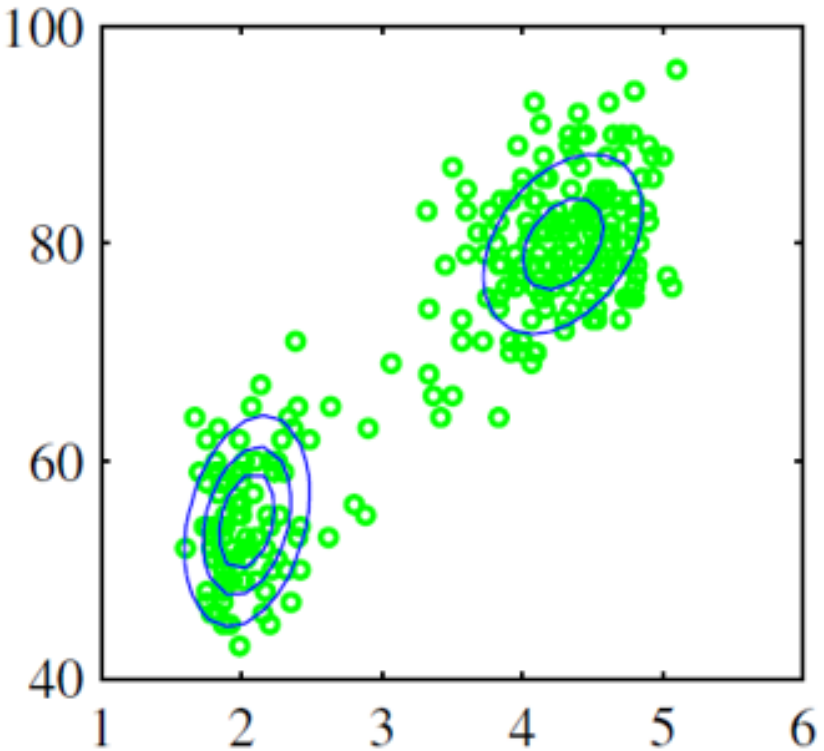
- A single Gaussian (left) may not be suitable to approximate some distributions

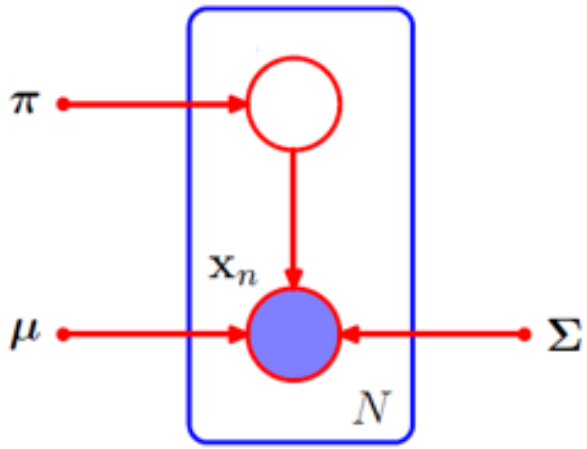


- Instead, we can use a *mixture* of K Gaussians (right)

Sampling from mixtures of Gaussians

- We can consider that samples are produced by the following process:
  - We select a cluster k among K clusters with probability π_k
  - Then we sample a point $\mathbf{x}$ based on the distribution of this cluster: a D-dim Gaussian with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$





Probability density function

- The resulting probability density function (p.d.f.) is

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

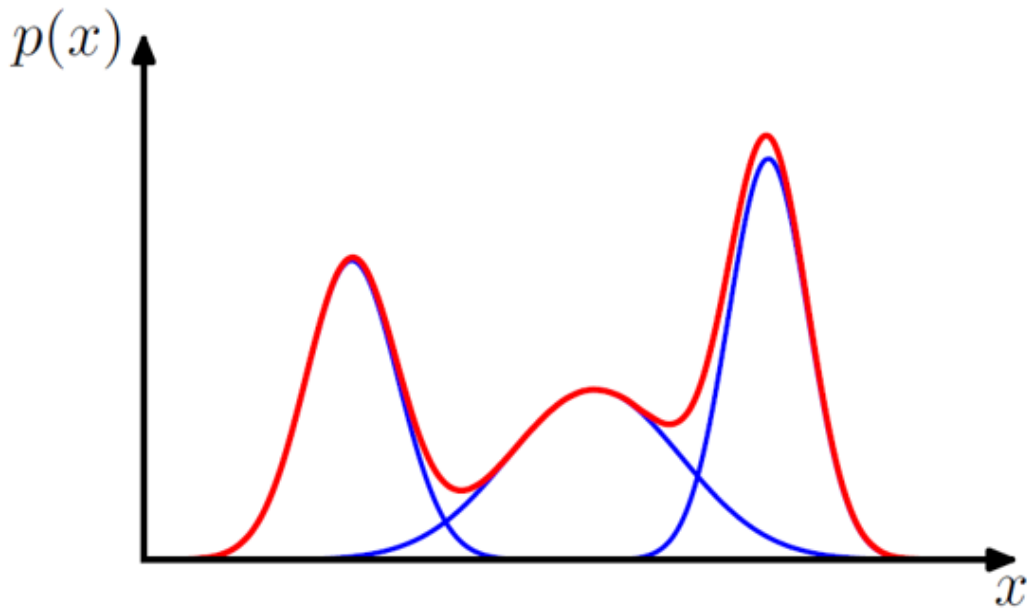
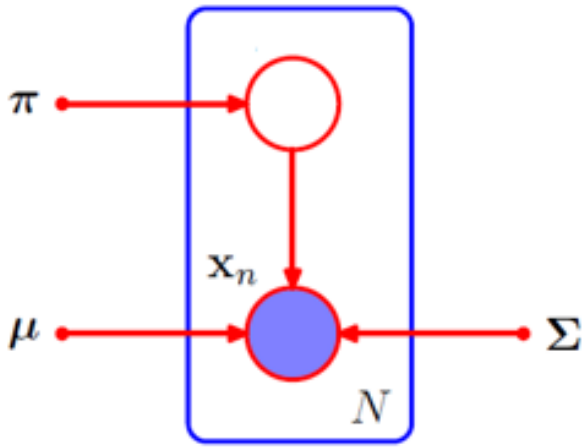


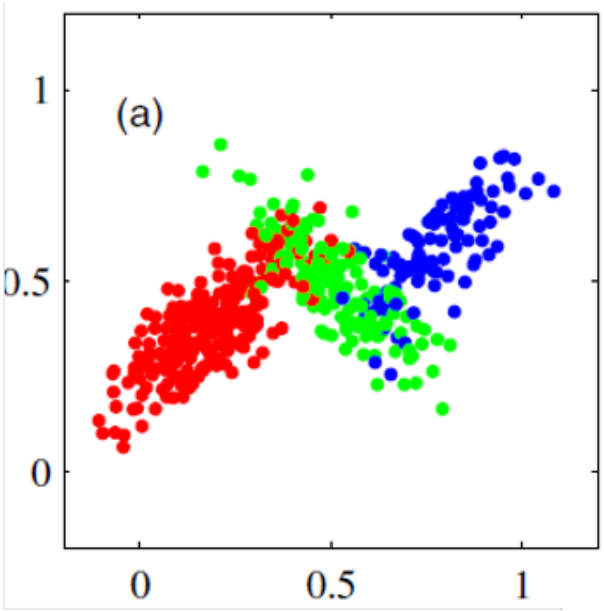
Figure: Probability density function of a 1D mixture of Gaussians

Modeling data as a mixture of Gaussians

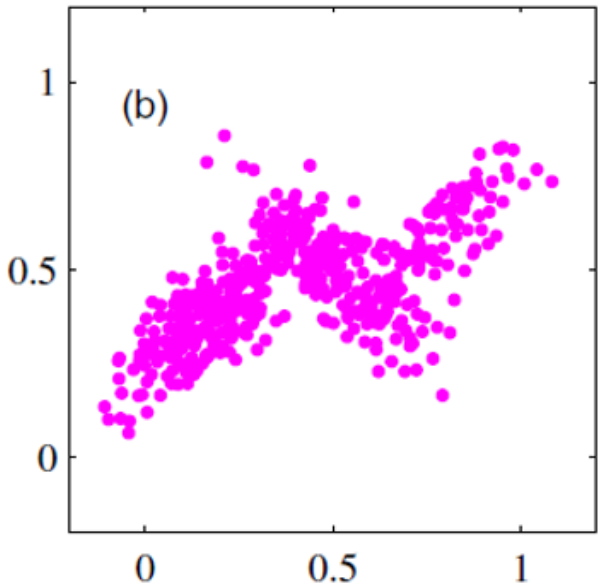
- Suppose we have N training points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and we want to model data as a mixture of K Gaussians
- We want to find the parameters π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ of each Gaussian
- We also want to know from which Gaussian each data point $\mathbf{x}_n$ was most likely sampled



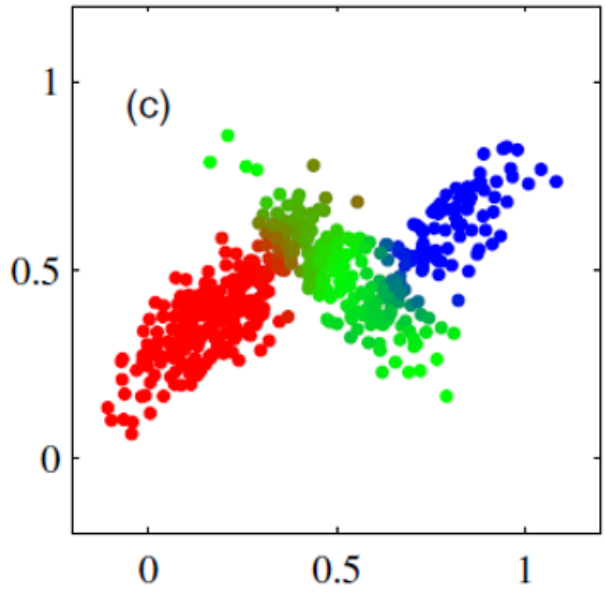
Modeling data as a mixture of Gaussians



(a) (Unknown) ground truth



(b) What we have



(c) What we want

We want to estimate π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ for $k \in \{1, \ldots, K\}$

Maximum likelihood approach?

- Our probability density function is

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

- The corresponding log-likelihood for dataset $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top \in \mathbb{R}^{N \times D}$ is

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

  where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top \in \mathbb{R}^K, \mathbf{M} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)^\top \in \mathbb{R}^{K \times D}, \boldsymbol{S} = (\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)^\top \in \mathbb{R}^{K \times D \times D}$

- Unfortunately, we cannot maximize this analytically

Additional notations

- We introduce a random variable $\mathbf{z} \in \{0,1\}^K$ defined such that

$$z_k \in \{0,1\} \quad \text{and} \quad \sum_k z_k = 1 \quad \text{(exactly 1 element of } \mathbf{z} \text{ is 1)}$$

$$\text{so that} \quad p(z_k = 1) = \pi_k$$

- We also introduce the conditional probability $\gamma(z_k)$ given the corresponding point $\mathbf{x}$:

$$\gamma(z_k) = p(z_k = 1|\mathbf{x})$$

- From Bayes theorem, we have:

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

Maximizing w.r.t. $\boldsymbol{\mu}_k$

- Let us see what must happen at a maximum of the likelihood function
- Setting $\dfrac{\partial \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = 0$ we obtain

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- From which we can get (multiplying by $\boldsymbol{\Sigma}_k^{-1}$)

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Maximizing w.r.t. $\boldsymbol{\Sigma}_k$ and π_k

- Similarly, setting $\dfrac{\partial \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} = 0$ we can obtain

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top$$

- With some slightly more cumbersome math[2] to maximize $\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ w.r.t. π_k we can get

$$\pi_k = \frac{N_k}{N}$$

[2] We need to introduce Lagrange multipliers to ensure that $\sum_k \pi_k = 1$

Are we done?

- Do we have a closed-form solution?

Are we done?

- Do we have a closed-form solution?
- Unfortunately, no: in

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

  $\gamma(z_{nk})$ depends on $\boldsymbol{\mu}_k$ (and on $\boldsymbol{\Sigma}_k$)
- Same thing in

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top$$

The Expectation-Maximization algorithm: E step

- But we can use the following iterative algorithm:
- Assuming we know π_k $\boldsymbol{\mu}_k$ $\boldsymbol{\Sigma}_k$, we estimate the probability that each point n was generated by a given cluster k

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

- This is the E step, or *Expectation* step

The Expectation-Maximization algorithm: M step

- Assuming we know the $\gamma(z_{nk})$, we evaluate the π_k $\boldsymbol{\mu}_k$ $\boldsymbol{\Sigma}_k$ with

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \quad \text{with} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top$$

$$\pi_k = \frac{N_k}{N}$$

- This is the M step, or *Maximization* step

The Expectation-Maximization algorithm: summary

In the Expectation-Maximization algorithm, we alternate between

- E step: given fixed parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k \quad \forall k$, estimate *responsibilities* $\gamma(z_{nk})$
- M step: given fixed responsibilities $\gamma(z_{nk}) \quad \forall n, k$, estimate parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k \quad \forall k$

until we reach convergence.

- *Side question: do we obtain an optimal solution to our initial problem?*

Illustration of the EM algorithm

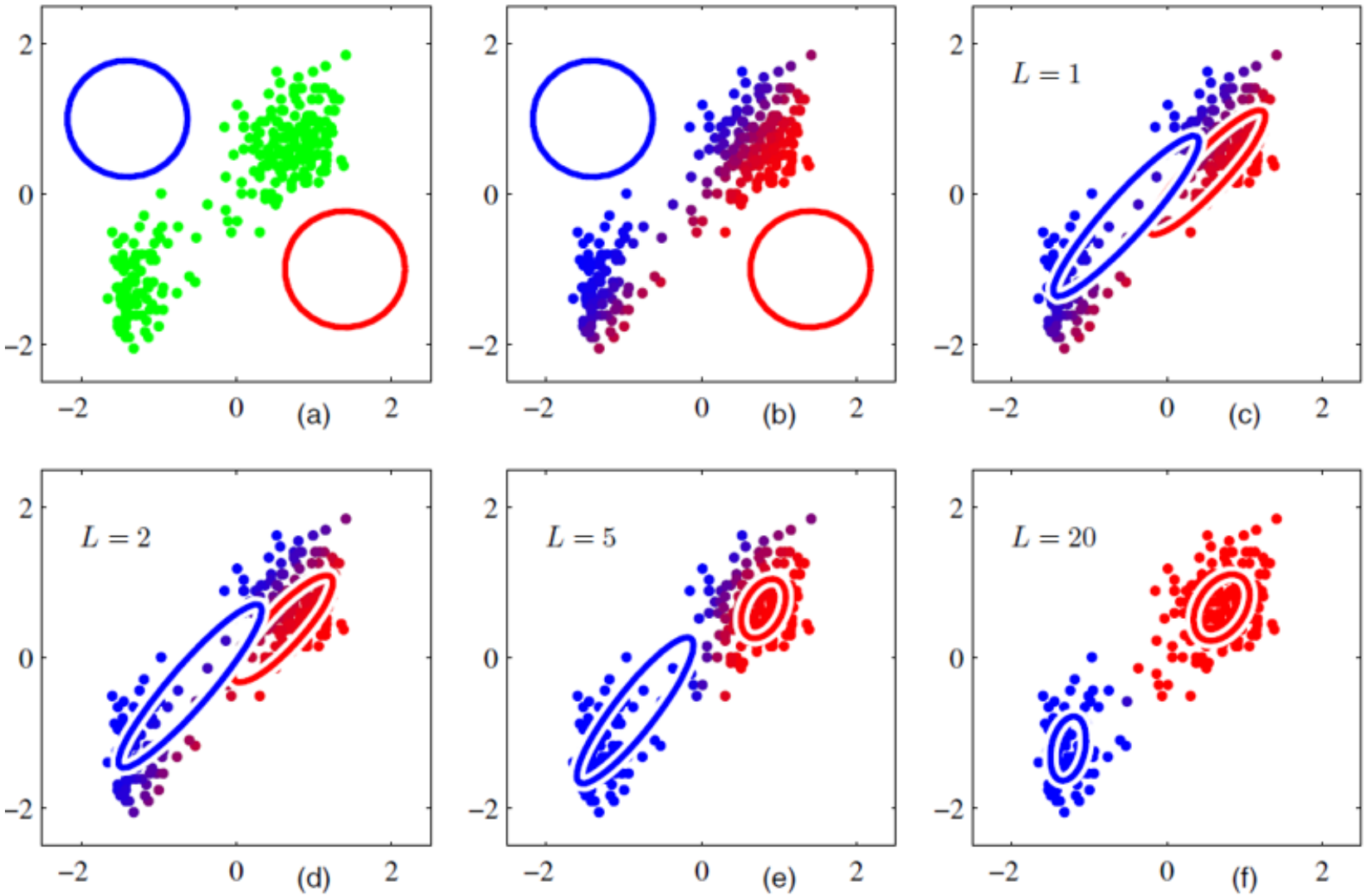


Figure: Illustration of the steps of the EM algorithm

Link with K-Means

- Notice anything?

Link with K-Means

- Notice anything?
- This looks very much like the iterations of the K-Means algorithm!
- K-Means can actually be considered as a "special case" of the EM algorithm where
  - $\gamma(z_{nk}) = 1 \ \forall n, k$: "hard" cluster assignements $\in \{0, 1\}$ instead of "soft" responsabilities $\in [0, 1]$
  - $\mathbf{\Sigma}_k = \sigma^2 \mathbf{I}_D \ \forall k$: isotropic Gaussians: distances are the same in all directions and for all clusters.
- Conversely, EM can be seen as a generalization of K-Means allowing for the two nuances above
- In practice, K-Means is often used as an initialization for the EM algorithm

Outline

EM as a general algorithm

- The Expectation Maximization algorithm can be applied to many other distributions beyond Gaussian mixtures
- General setting:
  - We have observed variables $\mathbf{X}$ and hidden variables $\mathbf{Z}$
  - We want to find parameters $\boldsymbol{\theta}$ maximizing $p(\mathbf{X}|\boldsymbol{\theta})$
  - But direct optimization is intractable

- Key idea: alternate between
  - E-step: Compute $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t) = \mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}_t}[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})]$
  - M-step: Find $\boldsymbol{\theta}_{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t)$

Example: Mixture of Bernoulli distributions

- Consider binary data $x_n \in \{0, 1\}$ from K different sources
- Each source k has 1 parameter p_k (probability of observing 1)
- Model parameters: $\boldsymbol{\theta} = \{\pi_k, p_k\}_{k=1}^K$

E-step: Compute responsibilities $\gamma(z_{nk})$

$$\gamma(z_{nk}) = \frac{\pi_k p_k^{x_n}(1-p_k)^{1-x_n}}{\sum_j \pi_j p_j^{x_n}(1-p_j)^{1-x_n}}$$

M-step: Update parameters p_k π_k

$$p_k^{\text{new}} = \frac{\sum_n \gamma(z_{nk}) x_n}{\sum_n \gamma(z_{nk})} \quad \text{and} \quad \pi_k^{\text{new}} = \frac{\sum_n \gamma(z_{nk})}{N}$$

- Compare with Gaussian mixture:
  - Same form for responsibilities but simpler likelihood
  - p_k replaces $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
  - Same update for mixing proportions π_k

Other applications of EM

- Hidden Markov Models (HMM)
  - Involves observations relying on a fixed number of hidden states with transition matrices
  - Hidden states are the latent variables $\mathbf{Z}$
  - Observed sequences are the visible variables $\mathbf{X}$
- Factor Analysis
  - Involves continuous latent variables, where Gaussian Mixture Model (GMM) involves discrete clusters
  - Latent factors are the hidden variables
  - Observed features are the visible variables
- Probabilistic Principal Component Analysis
  - Special case of GMM with a single Gaussian
  - Principal components are treated as latent variables
  - Data points are the observed variables
  - More robust to noise than standard PCA
- ...