Machine Learning

Lecture 6 – Expectation Maximization (EM) and clustering

UPPSALA
UNIVERSITET

Thomas Schön
Division of Systems and Control
Department of Information Technology
Uppsala University.

Email: thomas.schon@it.uu.se,
www: `user.it.uu.se/~thosc112`

Contents – lecture 6

1. Summary of lecture 5
2. Expectation Maximization (EM)
   - General derivation
   - Example - identification of a linear state space model
   - Example - identification of a Wiener system
3. Gaussian mixtures
   - Standard construction
   - Equivalent construction using latent variables
   - ML estimation using EM
4. Connections to the K-means algorithm for clustering

(Chapter 9)

Summary of lecture 5

A **Gaussian process** is a collection of random variables, any finite number of which have a joint Gaussian distribution.

By assuming that the considered system is a Gaussian process, predictions can be made by computing the conditional distribution $p(y(x^*)|\text{all the observations})$, $y(x^*)$ being the output for which we seek a prediction. This regression approach is referred to as **Gaussian process regression**.

The **suppor vector machine** (SVM) is a discriminative classifier that gives the maximum margin decision boundary.

Latent variables – example

A **latent variable** is a variable that is not directly observed. Other common names are hidden variables, unobserved variables or missing data.

An example of a latent variable is the state x_t in a state space model.

Consider the following linear Gaussian state space (LGSS) model

$$\begin{aligned} x_{t+1} &= \theta x_t + v_t, \\ y_t &= \frac{1}{2} x_t + e_t, \end{aligned} \qquad \begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right).$$

Expectation Maximization (EM) – strategy and idea 5(35)

The **Expectation Maximization (EM)** algorithm computes ML estimates of unknown parameters in probabilistic models involving latent variables.

Strategy: Use the *structure* inherent in the probabilistic model to separate the original ML problem into *two closely linked subproblems*, each of which is hopefully in some sense more tractable than the original problem.

EM focus on the joint log-likelihood function of the observed variables X and the latent variables $Z \triangleq \{z_1, \ldots, z_N\}$,

$$L_\theta(X, Z) = \ln p_\theta(X, Z).$$

Expectation Maximization algorithm 6(35)

Algorithm 1 Expectation Maximization (EM)

1. **Initialise:** Set $i = 1$ and choose an initial θ_1.
2. **While** not converged **do:**
   (a) **Expectation (E) step:** Compute

$$\begin{aligned} \mathcal{Q}(\theta, \theta_i) &= \mathrm{E}_{\theta_i}\left[\ln p_\theta(Z, X \mid X)\right] \\ &= \int \ln p_\theta(Z, X) p_{\theta_i}(Z \mid X) \mathrm{d}Z \end{aligned}$$

   (b) **Maximization (M) step:** Compute

$$\theta_{i+1} = \arg\max_\theta \mathcal{Q}(\theta, \theta_i)$$

   (c) $i \leftarrow i + 1$

EM example 1 – linear system identification 7(35)

Consider the following scalar LGSS model

$$\begin{aligned} x_{t+1} &= \theta x_t + v_t, \\ y_t &= \frac{1}{2} x_t + e_t, \end{aligned} \qquad \begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right).$$

The initial state is fully known ($x_1 = 0$) and the true θ-parameter is given by $\theta^\star = 0.9$.

The identification problem is now to determine the parameter θ on the basis of the observations $Y = \{y_1, \ldots, y_N\}$, using the EM algorithm.

The latent variables Z are given by the states
$Z = X \triangleq \{x_1, \ldots, x_{N+1}\}$.

Note the difference in notation compared to Bishop! The observations are denoted Y and the latent variables are denoted X.

EM Example 1 – linear system identification 8(35)

The expectation (E) step:

$$\mathcal{Q}(\theta, \theta_i) \triangleq \mathbf{E}_{\theta_i}\left\{\ln p_\theta(X, Y) \mid Y\right\} = \int \ln p_\theta(X, Y) p_{\theta_i}(X \mid Y) dX.$$

Let us start investigating $\ln p_\theta(X, Y)$. Using conditional probabilities we have,

$$\begin{aligned} p_\theta(X, Y) &= p_\theta(x_{N+1}, X_N, y_N, Y_{N-1}) \\ &= p_\theta(x_{N+1}, y_N \mid X_N, Y_{N-1}) p_\theta(X_N, Y_{N-1}), \end{aligned}$$

According to the Markov property we have

$$p_\theta(x_{N+1}, y_N \mid X_N, Y_{N-1}) = p_\theta(x_{N+1}, y_N \mid x_N),$$

resulting in

$$p_\theta(X, Y) = p_\theta(x_{N+1}, y_N \mid x_N) p_\theta(X_N, Y_{N-1}).$$

EM Example 1 – linear system identification

Repeated use of the above ideas straightforwardly yields

$$p_\theta(X, Y) = p_\theta(x_1) \prod_{t=1}^{N} p_\theta(x_{t+1}, y_t \mid x_t).$$

According to the model, we have

$$p_\theta\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} \mid x_t\right) = \mathcal{N}\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix}; \begin{pmatrix} \theta \\ 1/2 \end{pmatrix} x_t, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}\right).$$

EM example 1 – linear system identification

The resulting $\mathcal{Q}$-function is

$$\begin{aligned}\mathcal{Q}(\theta, \theta_i) &\propto -\mathbf{E}_{\theta_i}\left\{\sum_{t=1}^{N} x_t^2 \mid Y\right\}\theta^2 + 2\mathbf{E}_{\theta_i}\left\{\sum_{t=1}^{N} x_t x_{t+1} \mid Y\right\}\theta \\ &= -\varphi\theta^2 + 2\psi\theta,\end{aligned}$$

where we have defined

$$\varphi \triangleq \sum_{t=1}^{N} \mathbf{E}_{\theta_i}\left\{x_t^2 \mid Y\right\}, \qquad \psi \triangleq \sum_{t=1}^{N} \mathbf{E}_{\theta_i}\left\{x_t x_{t+1} \mid Y\right\}.$$

There exist explicit expressions for these expected values.

EM example 1 – linear system identification

The maximization (M) step,

$$\theta_{i+1} = \arg\max_\theta \mathcal{Q}(\theta, \theta_i).$$

simply amounts to solving the following quadratic problem,

$$\theta_{i+1} = \arg\max_\theta -\varphi\theta^2 + 2\psi\theta.$$

The solution is given by

$$\theta_{i+1} = \frac{\psi}{\varphi}.$$

EM example 1 – linear system identification

Algorithm 2 EM – example 1

1. **Initialise:** Set $i = 1$ and choose an initial θ_1.
2. **While** not converged **do:**
   (a) **Expectation (E) step:** Compute

$$\varphi = \sum_{t=1}^{N} \mathbf{E}_{\theta_i}\left\{x_t^2 \mid Y\right\}, \qquad \psi = \sum_{t=1}^{N} \mathbf{E}_{\theta_i}\left\{x_t x_{t+1} \mid Y\right\}.$$

   (b) **Maximization (M) step:** Find the next iterate according to

$$\theta_{i+1} = \frac{\psi}{\varphi}.$$

   (c) If $|L_{\theta_i}(Y) - L_{\theta_{i-1}}(Y)| \geq 10^{-6}$, update $i := i + 1$ and return to step 2, otherwise terminate.

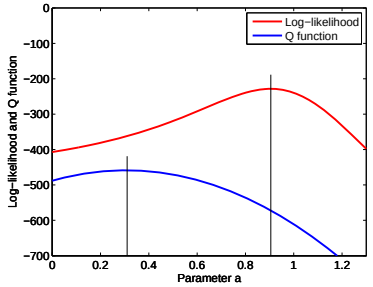
EM example 1 – linear system identification

- Different number of samples N used.
- Monte Carlo studies, each using $1\,000$ realisations of data.
- Initial guess $\theta_0 = 0.1$.

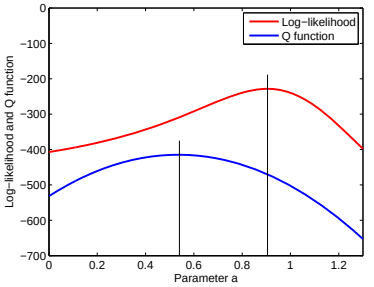
N	100	200	500	1 000	2 000	5 000	10 000
$\widehat{\theta}$	0.8716	0.8852	0.8952	0.8978	0.8988	0.8996	0.8998

No surprise, since ML is asymptotically efficient.

EM example 1 – linear system identification

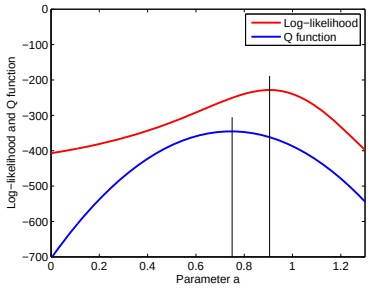


(a) Iteration 1

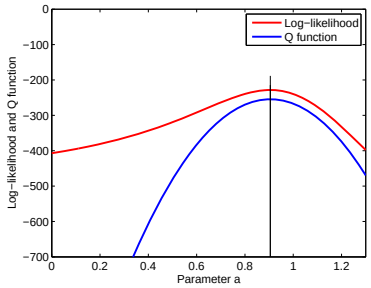


(b) Iteration 2

EM example 1 – linear system identification



(c) Iteration 3



(d) Iteration 11

All details (including MATLAB code) are provided in

Thomas B. Schön, **An Explanation of the Expectation Maximization Algorithm**. Division of Automatic Control, Linköping University, Sweden, *Technical Report nr: LiTH-ISY-R-2915*, August 2009.

`http://user.it.uu.se/~thosc112/pubpdf/schonem2009.pdf`

Nonlinear system identification using EM (I/VI)

A general state space model (SSM) consists of a Markov process $\{x_t\}_{t\geq 1}$ and a measurement process $\{y_t\}_{t\geq 1}$, related according to

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h_{\theta,t}(y_t \mid x_t, u_t), \\ x_1 &\sim \mu_\theta(x_1). \end{aligned}$$

Identification problem: Find θ based on $\{u_{1:T}, y_{1:T}\}$.

According to the above, the first step is to compute the $\mathcal{Q}$-function

$$\mathcal{Q}(\theta, \widehat{\theta}_k) = \mathbf{E}_{\theta_k}\left\{\ln p_\theta(Z, Y) \mid Y\right\}$$

Nonlinear system identification using EM (II/VI) 17(35)

Applying $\mathbf{E}_{\theta_k}\{\cdot \mid Y\}$ to

$$\begin{aligned}\ln p_\theta(X,Y) &= \ln p_\theta(Y \mid X) + \ln p_\theta(X) \\ &= \ln p_\theta(x_1) + \sum_{t=1}^{N-1} \ln p_\theta(x_{t+1} \mid x_t) + \sum_{t=1}^{N} \ln p_\theta(y_t \mid x_t).\end{aligned}$$

This results in $\mathcal{Q}(\theta,\theta_k) = I_1 + I_2 + I_3$, where

$$\begin{aligned}I_1 &= \int \ln p_\theta(x_1) p_{\theta_k}(x_1 \mid Y) \mathrm{d}x_1, \\ I_2 &= \sum_{t=1}^{N-1} \int\int \ln p_\theta(x_{t+1} \mid x_t) p_{\theta_k}(x_{t+1}, x_t \mid Y) \mathrm{d}x_t \mathrm{d}x_{t+1}, \\ I_3 &= \sum_{t=1}^{N} \int \ln p_\theta(y_t \mid x_t) p_{\theta_k}(x_t \mid Y) \mathrm{d}x_t.\end{aligned}$$

Nonlinear system identification using EM (III/VI) 18(35)

This leads us to a nonlinear state smoothing problem, which we can solve using a particle smoother (PS).

The PS provides us with the following approximation of the joint smoothing density

$$p(X \mid Y) \approx \frac{1}{M} \sum_{t=1}^{M} \delta\left(X - X^i\right),$$

which allows for the following approximations of the marginal smoothing densities that we need,

$$p_{\theta_k}(x_t \mid Y) \approx \widehat{p}_{\theta_k}(x_t \mid Y) = \frac{1}{M} \sum_{i=1}^{M} \delta(x_t - x_t^i),$$

$$p_{\theta_k}(x_{t:t+1} \mid Y) \approx \widehat{p}_{\theta_k}(x_{t:t+1} \mid Y) = \frac{1}{M} \sum_{i=1}^{M} \delta(x_{t:t+1} - x_{t:t+1}^i).$$

Nonlinear system identification using EM (IV/VI) 19(35)

Inserting the above approximations into the integrals straightforwardly yields the approximation we are looking for,

$$\begin{aligned}\widehat{I}_1 &= \int \ln p_\theta(x_1) \sum_{i=1}^{M} \frac{1}{M} \delta(x_1 - x_1^i) \mathrm{d}x_1 = \frac{1}{M} \sum_{i=1}^{M} \ln p_\theta(x_1^i), \\ \widehat{I}_2 &= \sum_{t=1}^{N-1} \int\int \ln p_\theta(x_{t+1} \mid x_t) \sum_{i=1}^{M} \frac{1}{M} \delta\left(x_{t:t+1} - x_{t:t+1}^i\right) \mathrm{d}x_{t:t+1} \\ &= \frac{1}{M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} \ln p_\theta(x_{t+1}^i \mid x_t^i), \\ \widehat{I}_3 &= \sum_{t=1}^{N} \int \ln p_\theta(y_t \mid x_t) \sum_{i=1}^{M} \frac{1}{M} \delta(x_t - x_t^i) \mathrm{d}x_t = \frac{1}{M} \sum_{t=1}^{N} \sum_{t=1}^{M} \ln p_\theta(y_t \mid x_t^i)\end{aligned}$$

Nonlinear system identification using EM (V/VI) 20(35)

It is straightforward to make use of the approximation of the Q-function just derived in order to compute gradients of the Q-function,

$$\frac{\partial}{\partial\theta} \widehat{\mathcal{Q}}(\theta,\theta_k) = \frac{\partial \widehat{I}_1}{\partial\theta} + \frac{\partial \widehat{I}_2}{\partial\theta} + \frac{\partial \widehat{I}_3}{\partial\theta}$$

For example (the other two terms are treated analogously),

$$\begin{aligned}\widehat{I}_3 &= \frac{1}{M} \sum_{t=1}^{N} \sum_{t=1}^{M} \ln p_\theta(y_t \mid x_t^i), \\ \frac{\partial \widehat{I}_3}{\partial\theta} &= \frac{1}{M} \sum_{t=1}^{N} \sum_{t=1}^{M} \frac{\partial \ln p_\theta(y_t \mid x_t^i)}{\partial\theta}\end{aligned}$$

With these gradients in place there are many algorithms that can be used in order to solve the maximization problem, we employ BFGS.

Nonlinear system identification using EM (VI/VI) 21(35)

Algorithm 3 Nonlinear System Identification Using EM

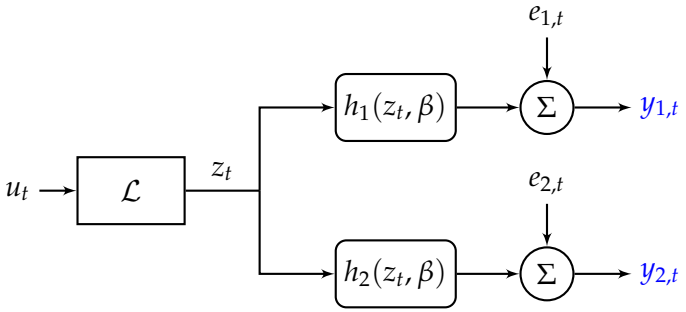
1. **Initialise:** Set $i = 1$ and choose an initial θ_1.
2. **While** not converged **do:**
   (a) **Expectation (E) step:** Run a FFBS PS and compute

$$\widehat{\mathcal{Q}}(\theta, \theta_k) = \widehat{I}_1(\theta, \theta_k) + \widehat{I}_2(\theta, \theta_k) + \widehat{I}_3(\theta, \theta_k)$$

   (b) **Maximization (M) step:** Compute $\theta_{k+1} = \arg\max_\theta \widehat{\mathcal{Q}}(\theta, \theta_k)$ using an off-the-shelf numerical optimization algorithm.
   (c) $k \leftarrow k + 1$

Thomas B. Schön, Adrian Wills and Brett Ninness. **System Identification of Nonlinear State-Space Models**. *Automatica*, 47(1):39-49, January 2011.

EM example 2 – blind Wiener identification (I/III) 22(35)



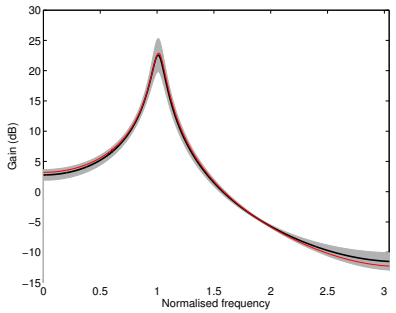
$$x_{t+1} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix}, \qquad u_t \sim \mathcal{N}(0, Q),$$

$$z_t = Cx_t, \qquad y_t = h(z_t, \beta) + e_t, \qquad e_t \sim \mathcal{N}(0, R).$$

Identification problem: Find A, B, C, β, Q, and R based on $\{y_{1,1:T}, y_{2,1:T}\}$ using EM.

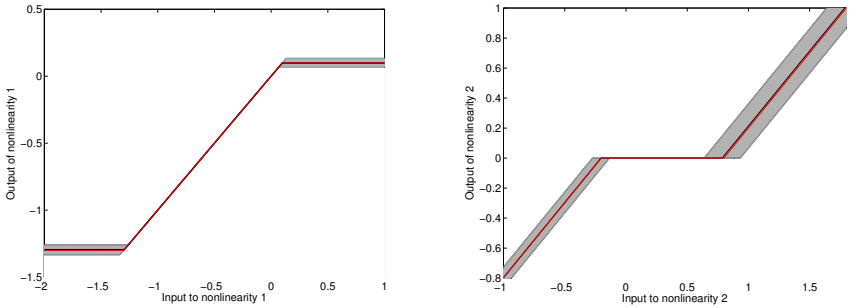
EM example 2 – blind Wiener identification (I/III) 23(35)

- Second order LGSS model with complex poles.
- Employ the EM-PS with $M = 100$ particles.
- EM-PS was terminated after 100 iterations.
- Results obtained using $T = 1\,000$ samples.
- The plots are based on 100 realizations of data.
- Nonlinearities (dead-zone and saturation) shown on next slide.



Bode plot of estimated mean (black), true system (red) and the result for all 100 realisations (gray).

EM example 2 – blind Wiener identification (I/III) 24(35)



Estimated mean (black), true static nonlinearity (red) and the result for all 100 realisations (gray).

Adrian Wills, Thomas B. Schön, Lennart Ljung and Brett Ninness. **Identification of Hammerstein-Wiener Models**. *Automatica*, 49(1): 70-81, January 2013.

Gaussian mixture (GM) – standard construction

A linear superposition of Gaussians

$$p(x) = \sum_{k=1}^{K} \underbrace{\pi_k}_{p(k)} \underbrace{\mathcal{N}\left(x_n \mid \mu_k, \Sigma_k\right)}_{p(x_n|k)}$$

is called a **Gaussian mixture (GM)**. The mixture coefficients π_k satisfies

$$\sum_{k=1}^{K} \pi_k = 1, \qquad 0 \leq \pi_k \leq 1.$$

Interpretation: The density $p(x \mid k) = \mathcal{N}(x \mid \mu_k, \Sigma_k)$ is the probability of x, given that component k was chosen. The probability of choosing component k is given by the prior probability $p(k)$.

GM – example

Consider the following GM,

$$p(x) = \underbrace{0.3}_{\pi_1} \mathcal{N}\left(x \mid \underbrace{\begin{pmatrix} 4 \\ 4.5 \end{pmatrix}}_{\mu_1}, \underbrace{\begin{pmatrix} 1.2 & 0.6 \\ 0.6 & 0.5 \end{pmatrix}}_{\Sigma_1}\right) + \underbrace{0.5}_{\pi_2} \mathcal{N}\left(x \mid \underbrace{\begin{pmatrix} 8 \\ 1 \end{pmatrix}}_{\mu_2}, \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\Sigma_2}\right) + \underbrace{0.2}_{\pi_3} \mathcal{N}\left(x \mid \underbrace{\begin{pmatrix} 9 \\ 8 \end{pmatrix}}_{\mu_3}, \underbrace{\begin{pmatrix} 0.6 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}}_{\Sigma_3}\right)$$

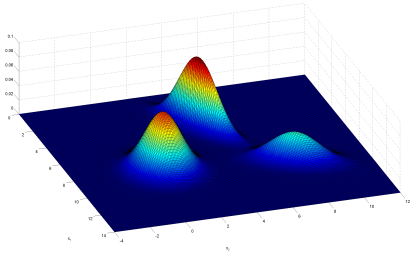


Figure: Probability density function.

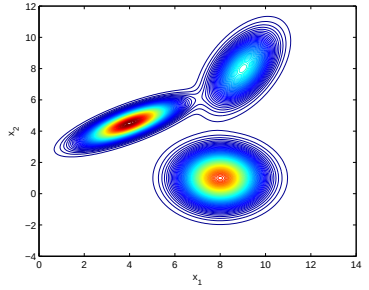


Figure: Contour plot.

GM – problem with standard construction

Given N independent observations $\{x_n\}_{n=1}^{N}$, the log-likelihood function if given by

$$\ln p(X; \pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(x_n \mid \mu_k, \Sigma_k\right) \right)$$

There is no closed form solution available (due to the sum inside the logarithm).

Let us now see how this problem can be separated into two simple problems using the EM algorithm.

First we introduce an **equivalent** construction of the Gaussian mixture by introducing a latent variable.

EM for Gaussian mixtures – intuitive preview

Based on

$$p(z_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}} \qquad \text{and} \qquad p(x_n \mid z_n) = \prod_{k=1}^{K} \mathcal{N}\left(x_n \mid \mu_k, \Sigma_k\right)^{z_{nk}}$$

we have (for independent observations $\{x_n\}_{n=1}^{N}$)

$$p(X, Z) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}\left(x_n \mid \mu_k, \Sigma_k\right)^{z_{nk}},$$

resulting in the following log-likelihood

$$\ln p(X, Z) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\ln \pi_k + \ln \mathcal{N}\left(x_n \mid \mu_k, \Sigma_k\right) \right). \qquad (1)$$

Let us now use wishful thinking and assume that Z is known. Then, maximization of (1) is straightforward.

EM for Gaussian mixtures – explicit algorithm

Algorithm 4 EM for Gaussian mixtures

1. **Initialise:** Initialize $\mu_k^1, \Sigma_k^1, \pi_k^1$ and set $i = 1$.
2. **While** not converged **do:**

(a) **Expectation (E) step:** Compute

$$\gamma(z_{nk}) = \frac{\pi_k^i \mathcal{N}(x_n \mid \mu_k^i, \Sigma_k^i)}{\sum_{j=1}^K \pi_j^i \mathcal{N}(x_n \mid \mu_j^i, \Sigma_j^i)}, \quad n = 1, \ldots, N, k = 1, \ldots, K.$$

(b) **Maximization (M) step:** Compute

$$\mu_k^{i+1} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n, \quad \pi_k^{i+1} = \frac{N_k}{N}, \quad N_k = \sum_{n=1}^N \gamma(z_{nk})$$

$$\Sigma_k^{i+1} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(x_n - \mu_k^{i+1})(x_n - \mu_k^{i+1})^T$$

(c) $i \leftarrow i + 1$

Example – EM for Gaussian mixtures (I/III)

Consider the same Gaussian mixture as before,

$$p(x) = \underbrace{0.3}_{\pi_1} \mathcal{N}\Big(x \mid \underbrace{\begin{pmatrix} 4 \\ 4.5 \end{pmatrix}}_{\mu_1}, \underbrace{\begin{pmatrix} 1.2 & 0.6 \\ 0.6 & 0.5 \end{pmatrix}}_{\Sigma_1}\Big) + \underbrace{0.5}_{\pi_2} \mathcal{N}\Big(x \mid \underbrace{\begin{pmatrix} 8 \\ 1 \end{pmatrix}}_{\mu_2}, \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\Sigma_2}\Big) + \underbrace{0.2}_{\pi_3} \mathcal{N}\Big(x \mid \underbrace{\begin{pmatrix} 9 \\ 8 \end{pmatrix}}_{\mu_3}, \underbrace{\begin{pmatrix} 0.6 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}}_{\Sigma_3}\Big)$$

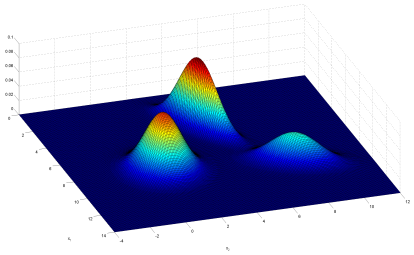


Figure: Probability density function.

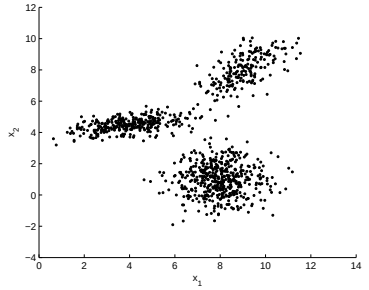


Figure: $N = 1\,000$ samples from the Gaussian mixture $p(x)$.

Example – EM for Gaussian mixtures (II/III)

- Apply the EM algorithm to estimate a Gaussian mixture with $K = 3$ Gaussians, i.e. use the $1\,000$ samples to compute estimates of $\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3$.
- 200 iterations.

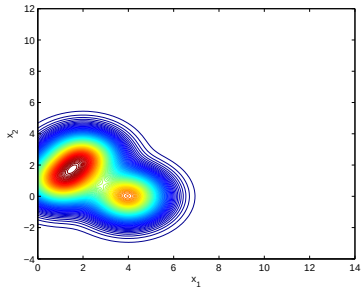


Figure: Initial guess.

Example – EM for Gaussian mixtures (III/III)

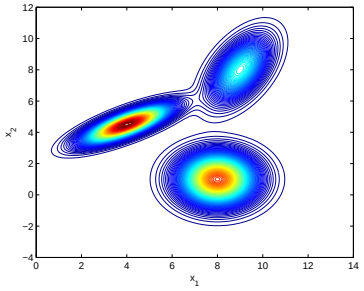


Figure: True PDF.

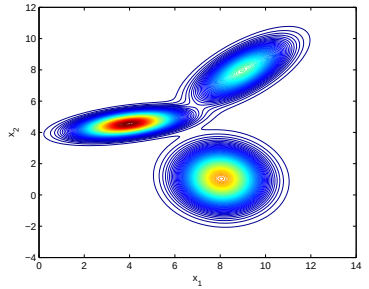


Figure: Estimate after 200 iterations of the EM algorithm.

The K-means algorithm (I/II)

Algorithm 5 K-means algorithm, a.k.a. Lloyd's algorithm

1. Initialize μ_k^1 and set $i = 1$.
2. Minimize J w.r.t. r_{nk} keeping $\mu_k = \mu_k^i$ fixed.

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

3. Minimize J w.r.t. μ_k keeping $r_{nk} = r_{nk}^{i+1}$ fixed.

$$\mu_k^{i+1} = \frac{\sum_{n=1}^N r_{nk}^{i+1} x_n}{\sum_{n=1}^N r_{nk}^{i+1}}.$$

4. If not converged, update $i := i + 1$ and return to step 2.

The K-means algorithm (II/II)

The name K-means stems from the fact that in step 3 of the algorithm, u_k is give by the mean of all the data points assigned to cluster k.

Note the **similarities** between the K-means algorithm and the EM algorithm for Gaussian mixtures!

K-means is deterministic with "hard" assignment of data points to clusters (no uncertainty), whereas EM is a probabilistic method that provides a "soft" assignment.

If the Gaussian mixtures are modeled using covariance matrices

$$\Sigma_k = \epsilon I, \quad k = 1, \ldots, K,$$

it can be shown that the EM algorithm for a mixture of K Gaussian's is **equivalent** to the K-means algorithm, when $\epsilon \to \infty$.

A few concepts to summarize lecture 6

Latent variable: A variable that is not directly observed. Sometimes also referred to as hidden variable or missing data.

Expectation Maximization (EM): The EM algorithm computes maximum likelihood estimates of unknown parameters in probabilistic models involving latent variables.

Jensen's inequality: States that if f is a convex function, then $\mathbf{E}(f(x)) \geq f(\mathbf{E}(x))$.

Clustering: Unsupervised learning, where a set of observations is divided into clusters. The observations belonging to a certain cluster are similar in some sense.

K-means algorithm (a.k.a. Lloyd's algorithm): A clustering algorithm assigning N observations into K clusters, where each observation belongs to the closest (Eucledian sense) cluster.