

# The Expectation – Maximization (EM) algorithm

- ♦ The Maximum Likelihood estimation requires to optimize the Likelihood (or the log likelihood)
- ♦ Depending on the form of  $p(x|\theta)$  the maximization can be easy or difficult
  - ♦ For example: if  $p(x|\theta)$  is a Gaussian, we have seen that we can set the derivatives of  $l(\theta)$  to zero, and solve for  $\theta$
- ♦ However, for many models the analytical solution can not be retrieved, and we have to resort to more complex technique, such as the Expectation-Maximization (E-M) algorithm

# The EM algorithm

- ♦ This represents a general method of finding, from a given data set, the maximum-likelihood estimate of the parameters of a probabilistic model with **hidden variables**
- ♦ EM is an iterative algorithm which, at each iteration, is guaranteed to **increase** the likelihood
- ♦ It is also guaranteed to **converge** to a maximum of the likelihood
  - ♦ The obtained maximum is however **local**

# The EM algorithm: general formulation

## PROBLEM FORMULATION

Given a Bayesian Network with observable and hidden variables and conditional probabilities parametrized by  $\theta$ ;

Given a training set  $D = \{x_1 \dots x_N\}$  (i.i.d), the goal of the maximum likelihood estimation is to find the parameter  $\theta_{ML}$  such that

$$\theta_{ML} = \arg \max_{\theta} p(D|\theta)$$

where

$$p(D|\theta) = \prod_{i=1}^N p(x_i|\theta)$$

# The EM algorithm: general formulation

Equivalent formulation (log-likelihood formulation)

$$\theta_{ML} = \arg \max_{\theta} \log p(D|\theta)$$

where

$$\log p(D|\theta) = \sum_{i=1}^N \log p(x_i|\theta)$$

# The EM algorithm: general formulation

## PRELIMINARY CONCEPTS:

### CONCEPT 1: Expectation

Given a function  $f(a,y)$ , where “a” is a constant and “y” is a random variable governed by the marginal distribution  $p(y)$

the **expected value of  $f(a,y)$  with respect to  $y$**  is defined as

$$E_y[f(a,y)] = \int_y f(a,y)p(y)dy \quad [y \text{ continuous}]$$

$$E_y[f(a,y)] = \sum_y f(a,y)p(y) \quad [y \text{ discrete}]$$

# The EM algorithm: general formulation

## INTUITION

$E_y[f(a,y)]$  represents the “weighted average” of the function  $f(a,y)$  for all possible value of  $y$ , where the weights are given by the probabilities  $p(y)$

Example: suppose  $y$  can take the three values (10, 20 or 30)

$$E_y[f(a,y)] = f(a,y=10)p(y=10) + \\ f(a,y=20)p(y=20) + \\ f(a,y=30)p(y=30)$$

Function when  
 $y$  is 30

Probability that  
 $y$  is 30

# The EM algorithm: general formulation

## CONCEPT 2: Conditional Expectation

Given a function  $f(a, y)$ , where “a” is a constant and “y” is a random variable governed by the marginal distribution  $p(y)$

**the expected value of  $f(a, y)$  with respect to  $y$  conditioned on another variable  $z$  is defined as**

$$E_{y|z}[f(a, y)|z] = \int_y f(a, y)p(y|z)dy \quad [y \text{ continuous}]$$

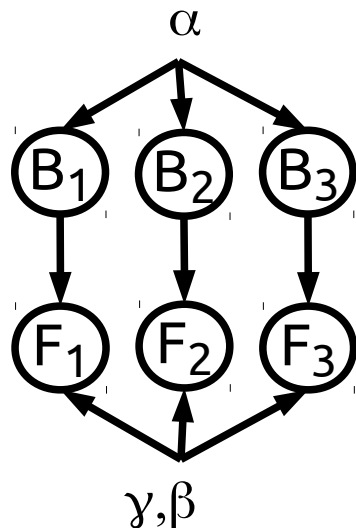
$$E_{y|z}[f(a, y)|z] = \sum_y f(a, y)p(y|z) \quad [y \text{ discrete}]$$

# The EM algorithm: general formulation

## E-M INTUITION

Let's call  $X$  the **observable data**  
(corresponding to observable variables of the  
Bayesian Network)

NOTE:  $X$  represents data for which we can have  
a training set (also called "Incomplete data")



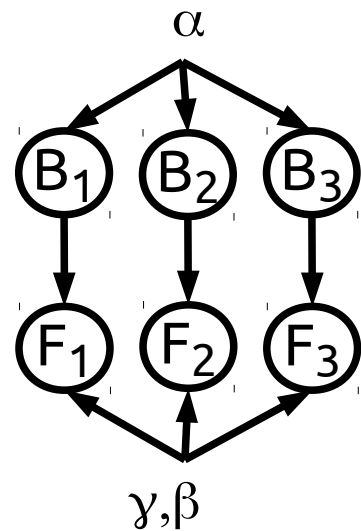
In our Two-boxes example  
(suppose we have three fruits)  
 $X = \{F_1, F_2, F_3\}$



# The EM algorithm: general formulation

## E-M INTUITION

Let's call  $Y$  the **hidden data** (corresponding to hidden variables of the Bayesian Network)



In our Two-boxes example  
(suppose we have three fruits)  
 $Y = \{B_1, B_2, B_3\}$

# The EM algorithm: general formulation

$Z = \{X, Y\}$  is called “Complete Data”

Now, we have to maximize  $p(X|\theta)$  - called the “incomplete likelihood”)

But we don't know  $Y$ , thus this maximization is impossible

Suppose we have  $F1='o'$ ,  $F2='a'$ ,  $F3='o'$ , how can we estimate  $\beta = P('o' | B = 'r')$ ??

But.... it would be easier if we would know  $Y$

$F1 = 'o'$	$F2 = 'a'$	$F3 = 'o'$
$B1 = 'r'$	$B2 = 'r'$	$B3 = 'b'$

# The EM algorithm: general formulation

In other words, it is easier to maximize the “complete likelihood”  $p(X, Y | \theta)$

## MAIN IDEA OF EM

- “guess” values for hidden variables  $Y$  (B in our two-boxes example)
- use the guesses to estimate  $\theta$
- use the new  $\theta$  to refine the guesses
- re-estimate  $\theta$  and so on

We have that with a proper “guessing” and “re-estimation” the procedure converges to an optimal  $\theta$

# The EM algorithm: general formulation

## The EM algorithm

Starting from some initial values  $\theta^0$  for the parameters  $\theta$ , the E-M algorithm iterates between these two steps until convergence

E-step: “compute guesses”

M-step: “re-estimate parameters”

# The EM algorithm: general formulation

## E-step

Compute the so-called **Q-function**, which is defined as the expected value of the complete log likelihood  $\log p(X, Y | \theta)$  with respect to  $y$  given the observed data  $X$  and the current parameter estimate  $\theta^{(i-1)}$

$$Q(\theta, \theta^{(i-1)}) = E_y \left[ \log p(X, Y | \theta) | X, \theta^{(i-1)} \right]$$

$$\longrightarrow \int_y \log p(X, y | \theta) p(y | X, \theta^{(i-1)}) dy \quad [y \text{ continuous}]$$

$$\longrightarrow \sum_y \log p(X, y | \theta) p(y | X, \theta^{(i-1)}) \quad [y \text{ discrete}]$$

# The EM algorithm: general formulation

## OBSERVATIONS

Observation 1.  $p(y|X, \theta^{(i-1)})$  represents the “guess” we are making on the hidden variables given the current estimate and the data

Example:

$$p(B = 'r' \mid X, \theta^{(i-1)}) = 0.1$$

$$p(B = 'b' \mid X, \theta^{(i-1)}) = 0.9$$

my guess is that the box should be blu (almost sure about this)

# The EM algorithm: general formulation

## Observation 2.

$\theta$  are the parameters to be estimated (i.e. they are random variables) whereas  $\theta^{(i-1)}$  are the parameters estimated at the previous iteration (i.e. they simply represent constants)

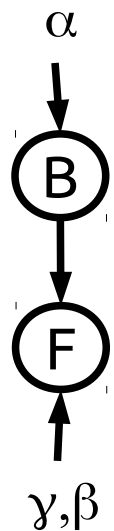
# The EM algorithm: general formulation

Observation 3.

How to compute “guesses”  $p(Y|X,\theta)$ ?

When we know the parameters  $\theta$  it is easy to compute  $p(Y|X,\theta)$

Example



$p(Y|X,\theta)$  in this case is  $p(B|F,\theta)$

Knowing  $\alpha, \beta, \gamma$  we can compute the joint probability  $p(B, F|\theta)$  and derive  $P(B|F,\theta)$

$$p(B|F) = \frac{p(B, F)}{p(F)} = \frac{p(B, F)}{\sum_B p(B, F)}$$



# The EM algorithm: general formulation

We don't know  $\theta$ , but we have an estimate of it coming from the previous step:  $\theta^{(i-1)}$

The idea is to use this estimate to compute the guesses

At the very beginning, we will have a poor estimate, but as soon as we have a better estimate we can have better guesses

# The EM algorithm: general formulation

M-step.

Obtain a novel estimate  $\theta^{(i)}$  for the parameters

$$\theta^{(i)} = \arg \max_{\theta} Q(\theta, \theta^{(i-1)})$$

# The EM algorithm: general formulation

## INTUITION

Suppose that  $y$  is discrete (and can take three values (10,20,30)), and suppose that we exactly know the true value of  $y$  (20)

Then  $p(y|X, \theta^{(i-1)})$  has all zeros and 1 in the position of the true value

$$p(y|X, \theta^{(i-1)}) = [0 \ 1 \ 0]$$

Therefore the Q-function reduces to:

# The EM algorithm: general formulation

$$\begin{aligned} Q(\theta, \theta^{(i-1)}) &= \sum_y \log p(X, y|\theta)p(y|X, \theta^{(i-1)}) \\ &= \log p(X, y = 10|\theta)p(y = 10|X, \theta^{(i-1)}) + \\ &\quad + \log p(X, y = 20|\theta)p(y = 30|X, \theta^{(i-1)}) + \\ &\quad + \log p(X, y = 30|\theta)p(y = 30|X, \theta^{(i-1)}) \\ &= \log p(X, y = 10|\theta) * 0 + \\ &\quad + \log p(X, y = 20|\theta) * 1 + \\ &\quad + \log p(X, y = 30|\theta) * 0 \\ &= \log p(X, y = 20|\theta) \end{aligned}$$

# The EM algorithm: general formulation

Now, the maximization of  $Q(\theta, \theta^{(i-1)})$  reduces to the maximization of  $\log p(X, y=10 | \theta^{(i-1)})$ , which is doable as seen before (it is the maximization of the likelihood knowing the hidden variables)

But: since we don't know the value of the hidden variable  $y$ , we maximize the "weighted average" of the likelihood

$$\sum_y \log p(X, y | \theta) p(y | X, \theta^{(i-1)})$$

Likelihood for a  
given value of  $y$

Probability of  
such value

# The EM algorithm: summary

Initialization: set  $\theta^{(0)}$

Repeat:

E-Step: compute the  $Q(\theta, \theta^{(i-1)})$  function

$$Q(\theta, \theta^{(i-1)}) = E_Y \left[ \log p(\mathcal{X}, Y | \theta) | \mathcal{X}, \theta^{(i-1)} \right]$$

M-Step: re-estimate the parameters

$$\theta^{(i)} = \arg \max_{\theta} Q(\theta, \theta^{(i-1)})$$

Until convergence (e.g.  $\theta^{(i)} - \theta^{(i-1)} < \text{epsilon}$ )

# An example: the EM algorithm for mixture of Gaussians

- Let's first recap the GMM model. Within a GMM we assume that the distribution of the points follows the following formula

$$p(x) = \sum_{j=1}^K \pi_j f_j(x|\Theta_j)$$

Where every component  $f_j$  is a Gaussian

$$f_j(x_i|\theta_j) = f_j(x_i|\mu_j, \Sigma_j) = \frac{1}{2\pi^{d/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)}$$

# An example: the EM algorithm for mixture of Gaussians

- ♦ Example: GMM in 1D with 2 components

$$p(x) = \pi_1 \mathcal{N}(x|\mu_1, \sigma_1) + \pi_2 \mathcal{N}(x|\mu_2, \sigma_2)$$

$\pi_1$  = Prior probability for the first Gaussian

$\pi_2$  = Prior probability for the second Gaussian

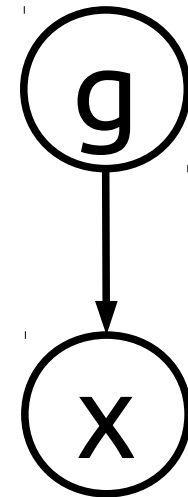
$$N(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{||x-\mu||^2}{2\sigma^2}}$$



# GMM as Bayesian Networks

Variables and edges:

- ♦  $\mathbf{x}$  = variable usable to describe the **value** of the point (**visible** variable, continuous)
- ♦  $\mathbf{g}$  = variable usable to describe which is the **component**, i.e. from which Gaussian we have sampled the point (hidden variable, discrete)
- ♦ Clearly the value of  $\mathbf{x}$  depends on the chosen Gaussian (i.e. on the value of  $\mathbf{g}$ )



# GMM as Bayesian Networks

Conditional probabilities:

- $p(g)$ : the prior probabilities ( $g$  has no parents)

$$p(g = 1) = \pi_1 \quad p(g = 2) = \pi_2$$

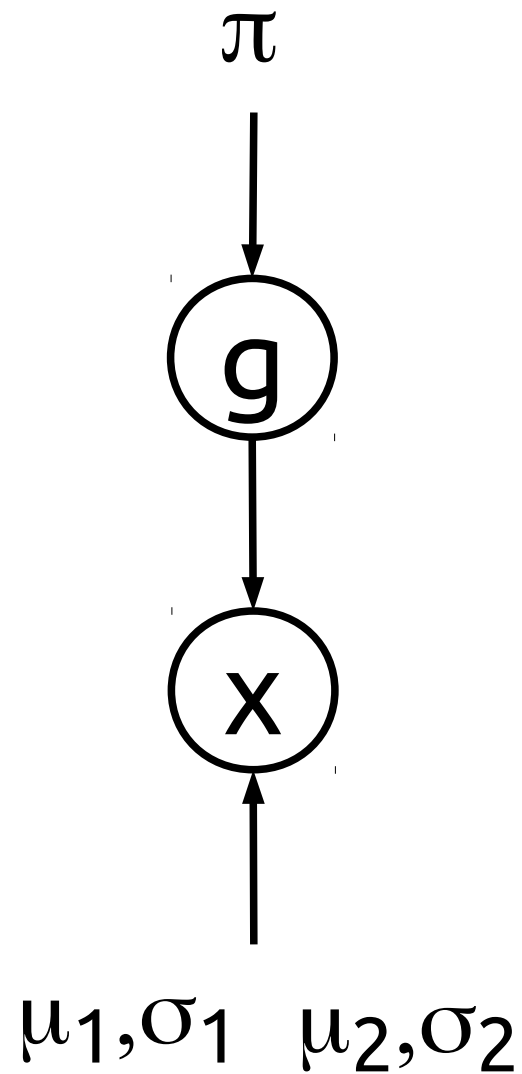
- $p(x|g)$  = the two Gaussian probabilities

$$p(x|g = 1) = \mathcal{N}(x|\mu_1, \sigma_1)$$

$$p(x|g = 2) = \mathcal{N}(x|\mu_2, \sigma_2)$$

# GMM as Bayesian Networks

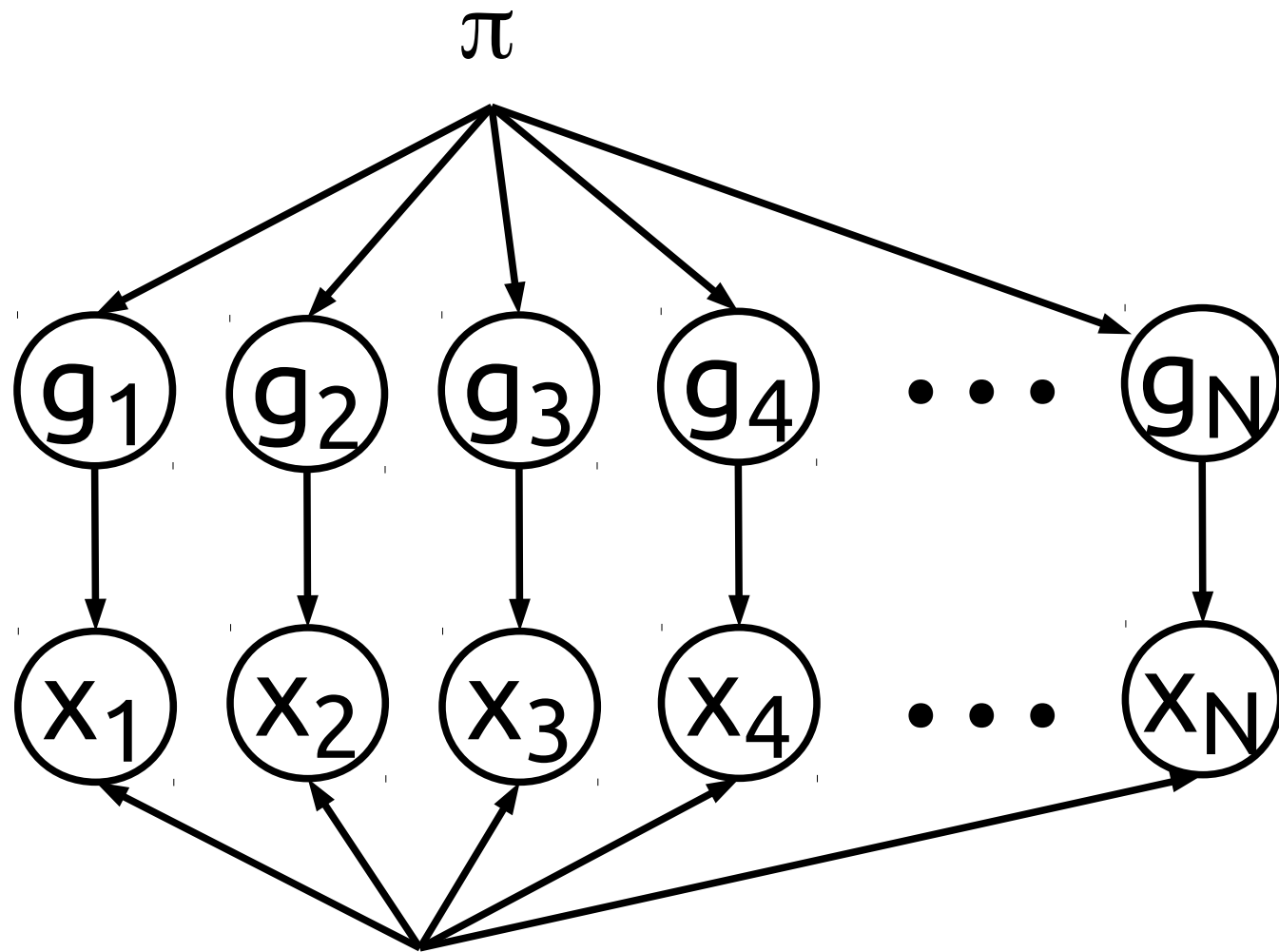
The full Bayesian Network



# GMM as Bayesian Networks

- ♦ NOTE: This represents a Bayesian Networks for **one** point
- ♦ As in the other example (the two boxes case), here we want to model **a set** of points  $x_1, \dots, x_N$ , all generated with the scheme described before
  - ♦ This means that there is a pair of variables  $(x, g)$  for every point of the dataset

# GMM as Bayesian Networks



$\mu_1, \sigma_1 \mu_2, \sigma_2$  ←

NOTE: The parameters are the same for all points

# An example: the EM algorithm for mixture of Gaussians

- EM for mixture of Gaussians (One-dimensional case)

(Board)

# The EM for Mixture of 1D Gaussians

**Initialization:**  $t = 0$ , set  $\theta^{(0)}$

$$\theta^{(0)} = \{\mu_1^{(0)}, \sigma_1^{(0)}, \pi_1^{(0)}, \dots, \mu_K^{(0)}, \sigma_K^{(0)}, \pi_K^{(0)}\}$$

**Repeat:**

$$t = t+1$$

**E-Step:** compute  $w_{ij}$  for all  $i=1..N$  and  $j=1..K$

$$w_{ij} = p(j|x_i, \theta^{(t-1)}) = \frac{\pi_j^{(t-1)} \mathcal{N}(x_i | \mu_j^{(t-1)}, \sigma_j^{(t-1)})}{\sum_{\ell=1}^K \pi_{\ell}^{(t-1)} \mathcal{N}(x_i | \mu_{\ell}^{(t-1)}, \sigma_{\ell}^{(t-1)})}$$

**M-Step:** re-estimate  $\theta^{(t)}$  (for all  $j=1..K$ )

$$\pi_j^{(t)} = \frac{1}{N} \sum_{i=1}^N w_{ij}, \quad \mu_j^{(t)} = \frac{\sum_{i=1}^N w_{ij} x_i}{\sum_{i=1}^N w_{ij}}, \quad (\sigma_j^2)^{(t)} = \frac{\sum_{i=1}^N w_{ij} (x_i - \mu_j^{(t)})^2}{\sum_{i=1}^N w_{ij}}$$

Until convergence (e.g.  $LL^{(t)} - LL^{(t-1)} < \text{epsilon}$ )

# The EM algorithm

- ♦ In summary: in the E-STEP the EM estimates the probability that every Gaussian had generated the different points

For every point  $x_i$

$$w_{i1} = p(y_i = 1 | \mathcal{X}, \theta^{(i-1)}) = p(y_i = \text{'blue'} | \mathcal{X}, \theta^{(i-1)})$$

$w_{i1}$  represents how probable is the fact that the point has been generated by the first Gaussian (the blue one) given the current parameters

$$w_{i2} = p(y_i = 2 | \mathcal{X}, \theta^{(i-1)}) = p(y_i = \text{'red'} | \mathcal{X}, \theta^{(i-1)})$$



# The EM algorithm

- in the M-STEP the EM re-estimates the parameters using the values computed in the E-STEP

$$\mu_1 = \sum_{i=1}^N w'_{i1} x_i$$

$$\left[ w'_{i1} = \frac{w_{i1}}{\sum_{j=1}^N w_{j1}} \right]$$

The new mean for the Blue Gaussian is the average of **all** points, each one weighted with the probability of having been generated by the blue Gaussian

Note: In the classical average: all points have the same weight ( $w_i=1/N$ )

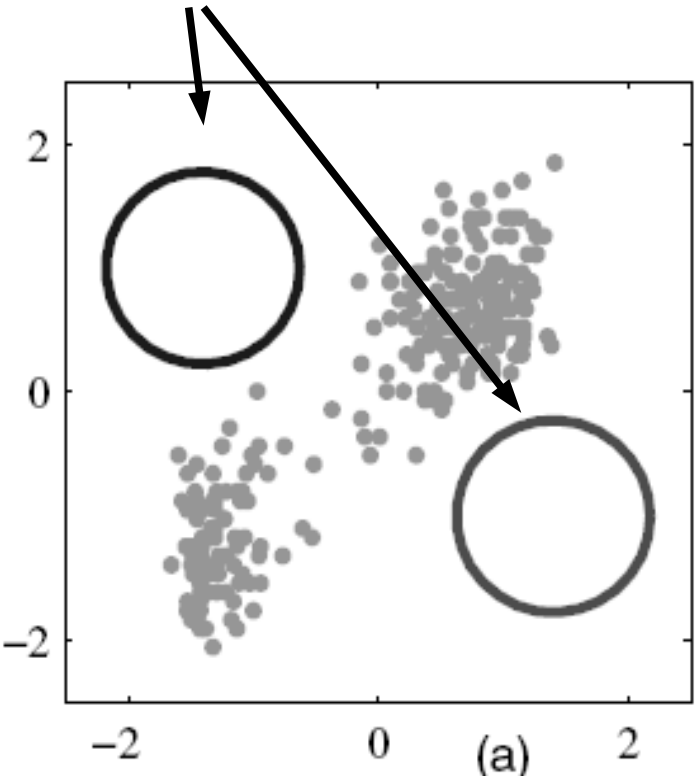
$$\text{Average} = \frac{1}{N} \sum_{i=1}^N x_i = \sum_{i=1}^N \frac{1}{N} x_i$$

Weighted average: each point contributes according to the weight

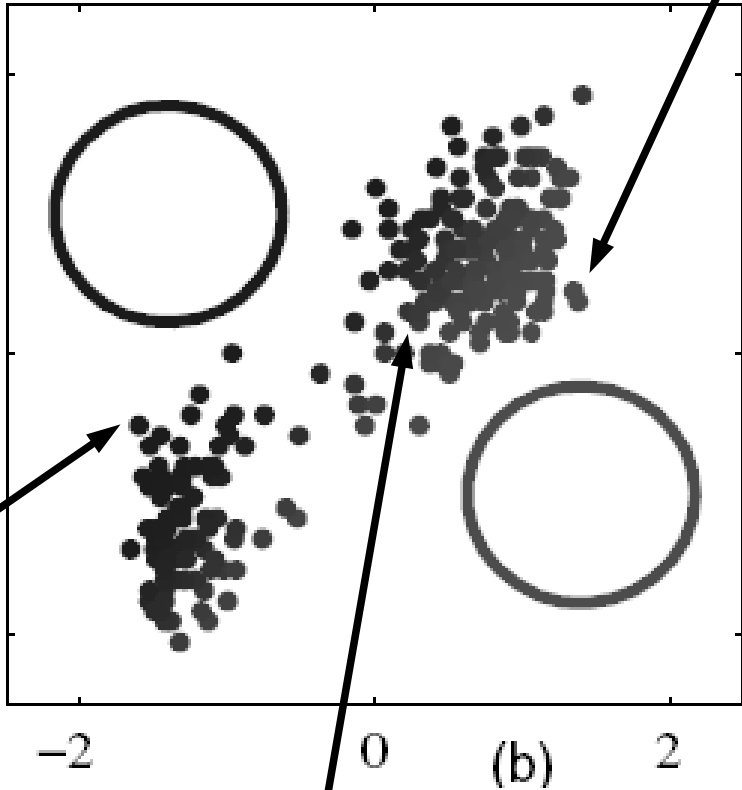
$$\text{Weighted Average} = \sum_{i=1}^N w_i x_i$$

Note:  $w_{i1} + w_{i2} = 1$

Current guesses  $\theta^{(i-1)}$



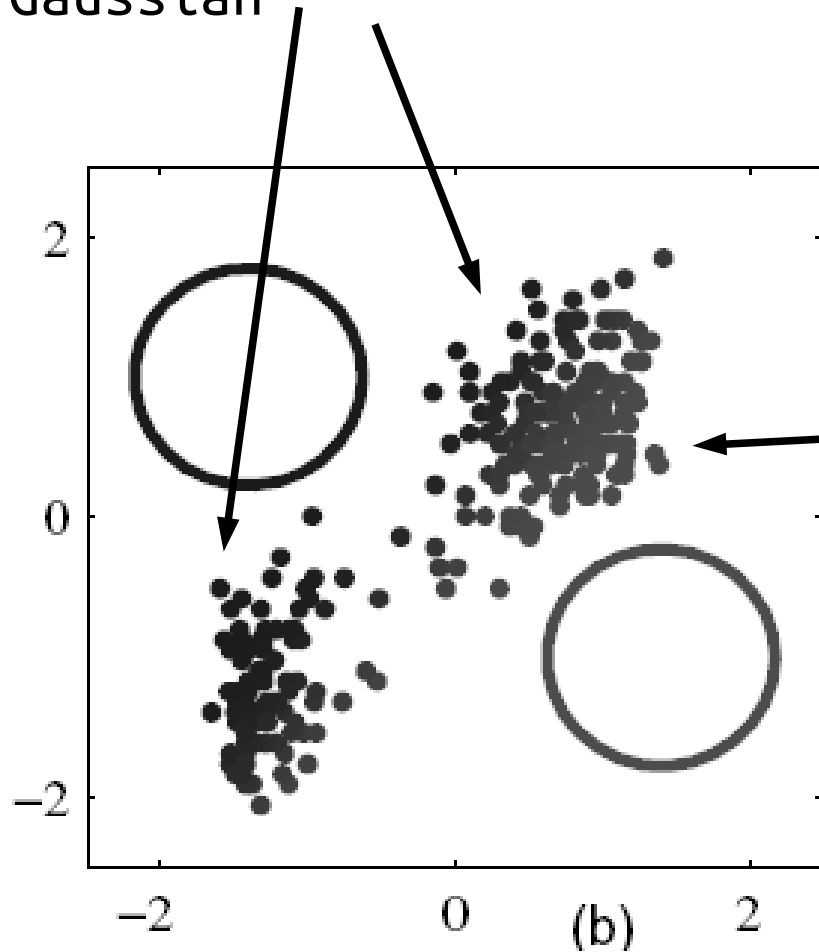
Red point:  $w_{i2} > w_{i1}$ ! It is more probable that the point has been generated by the red Gaussian



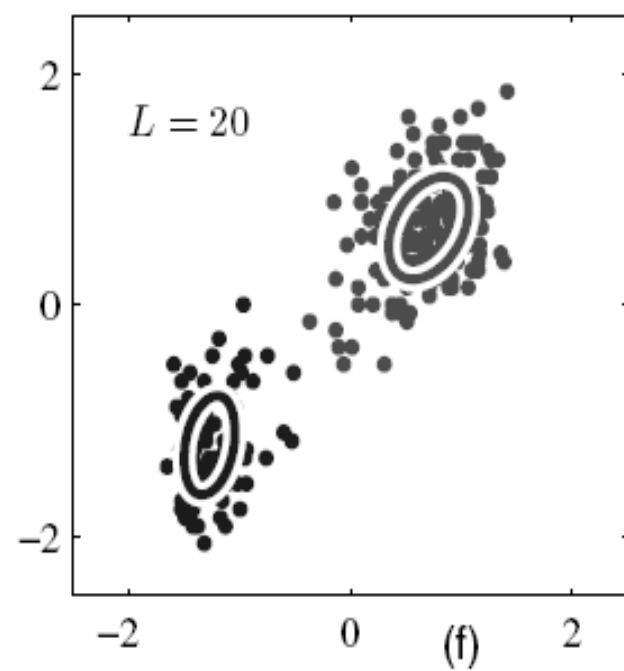
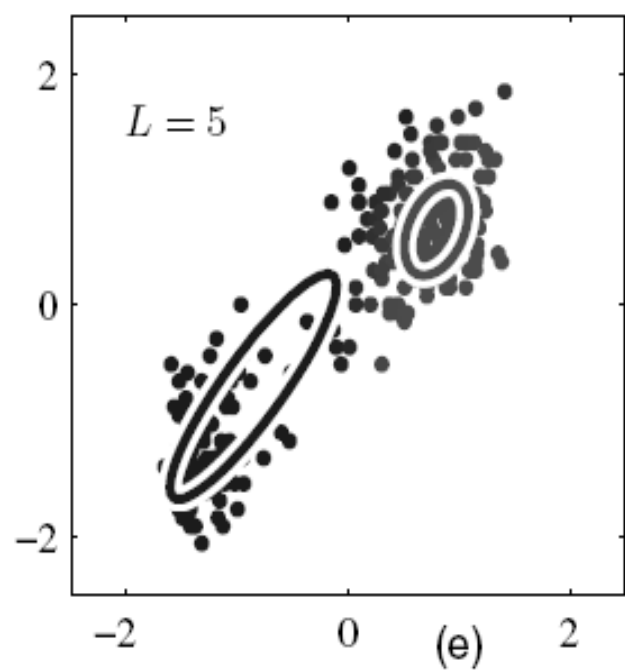
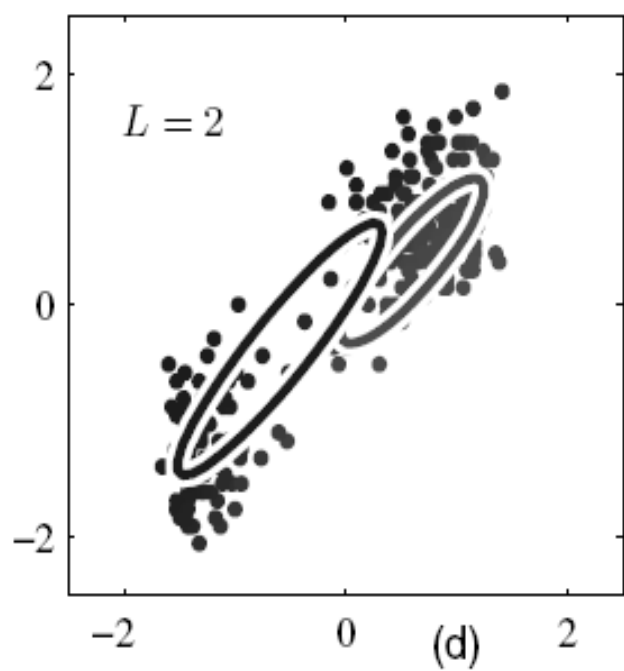
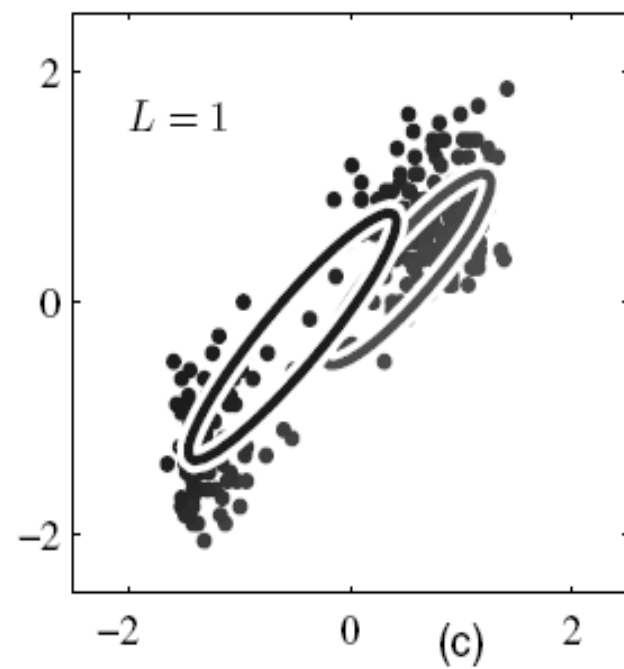
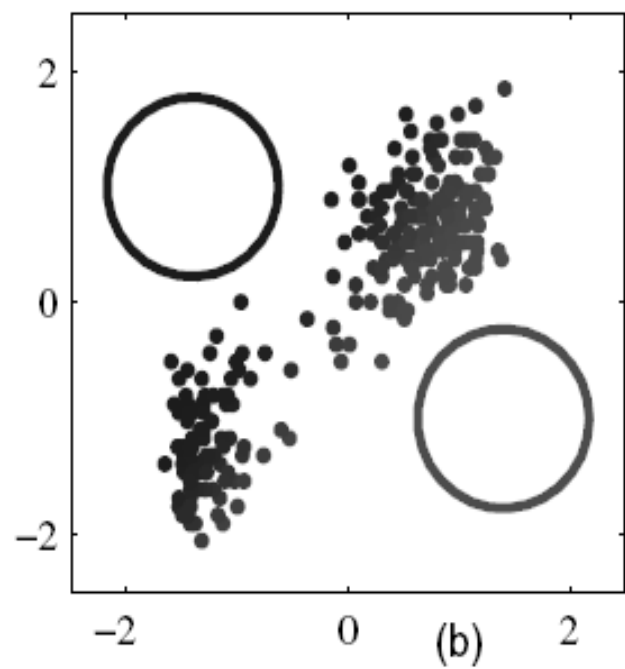
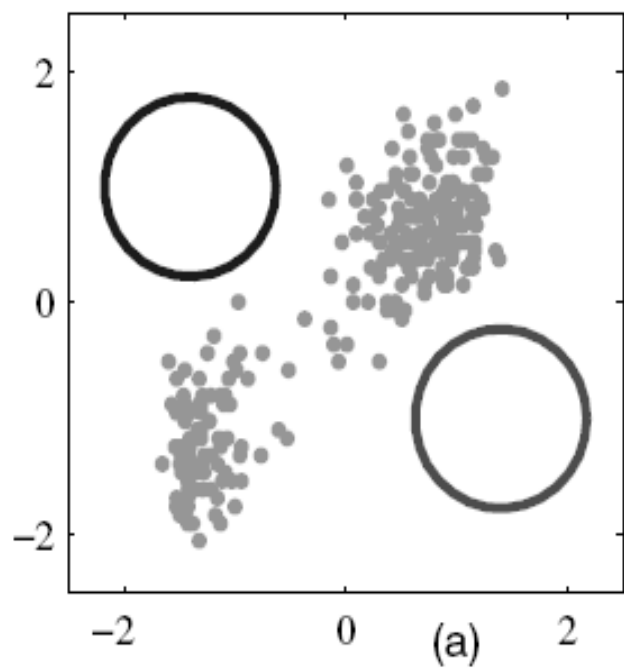
blue point:  $w_{i1} > w_{i2}$ ! It is more probable that the point has been generated by the blue Gaussian

This point is purple (a mixture of blue and red):  $w_{i1} \sim w_{i2}$ ! Both Gaussians are equiprobable

For these points,  $w_{i1}$  is larger than  $w_{i2}$ , therefore they would contribute more to the novel estimation of the blue Gaussian



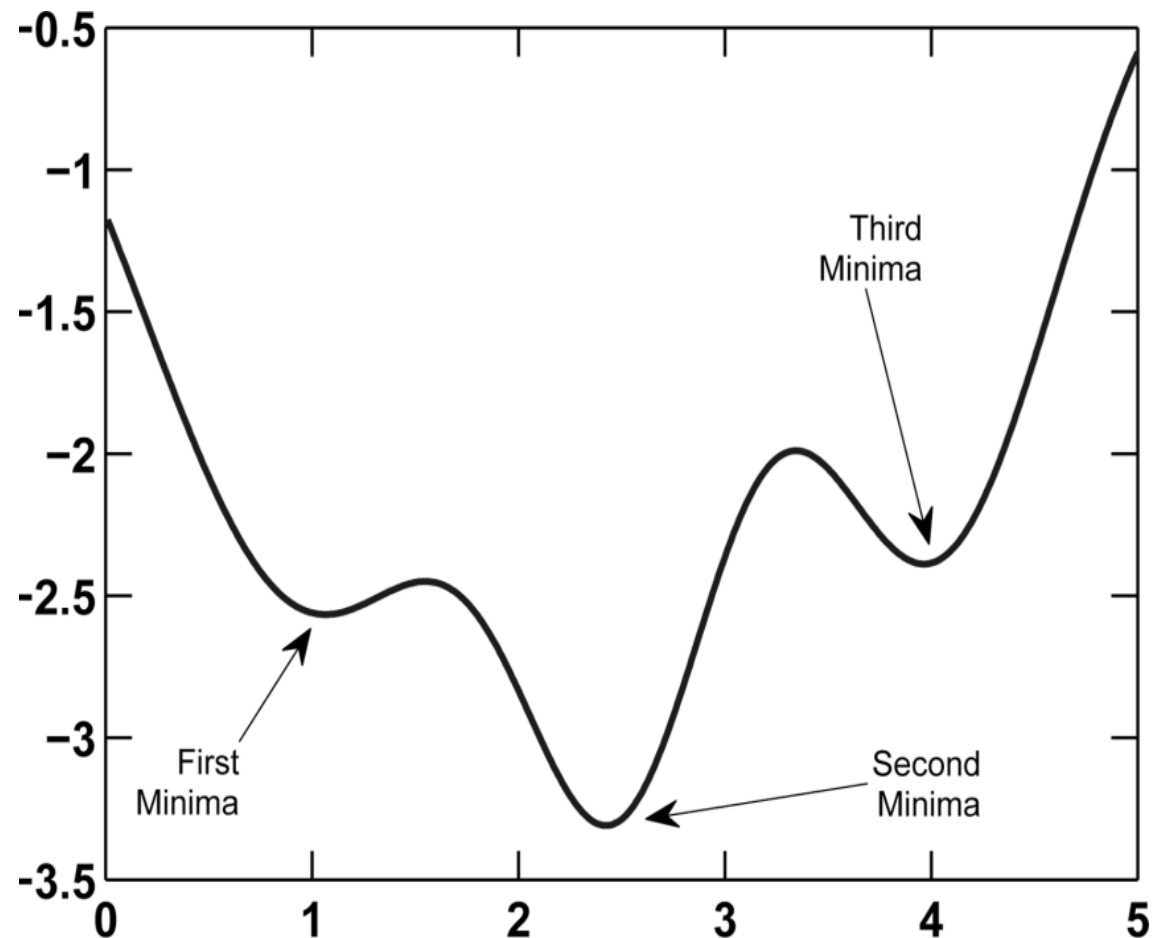
For these points, on the contrary,  $w_{i2}$  is larger, therefore they would contribute more to the novel estimation of the red Gaussian



# Initialization

- The EM algorithm, starting from an **initial estimate**, converges to a **local** optimum.

Since typically the Log likelihood is highly multimodal, finding a good starting point is crucial to get a good estimate



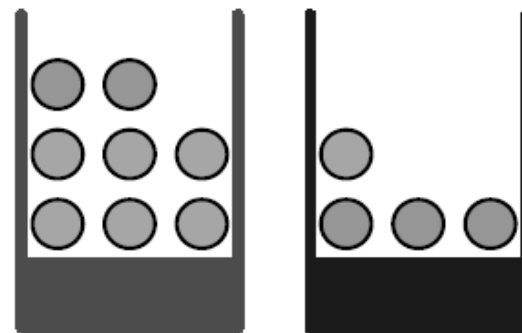
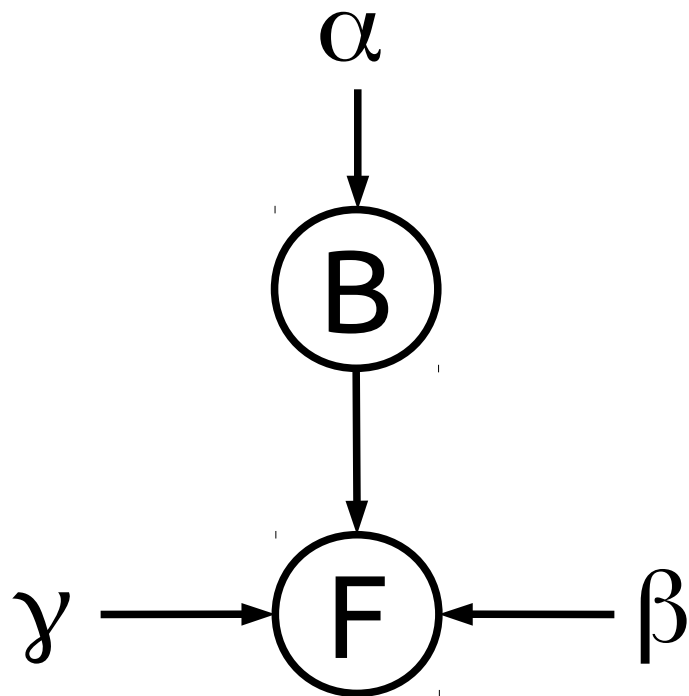
# Summary

- ♦ The EM algorithm represents a flexible tool to learn a Maximum Likelihood estimate of the parameters of a Bayesian Network
  - ♦ It can be used also in other contexts
- ♦ It requires complex mathematical derivations: depending on the complexity of the Bayes Net, it can be easy / difficult / impossible to derive analytical E-Step and M-Step
- ♦ Needed tradeoff: **computability** vs **descriptivity**

# Inference

# Inference

- Once the model is learnt, it can be interesting to **query** the model, i.e. to ask something
- Example 1: the “**Two Boxes problem**”





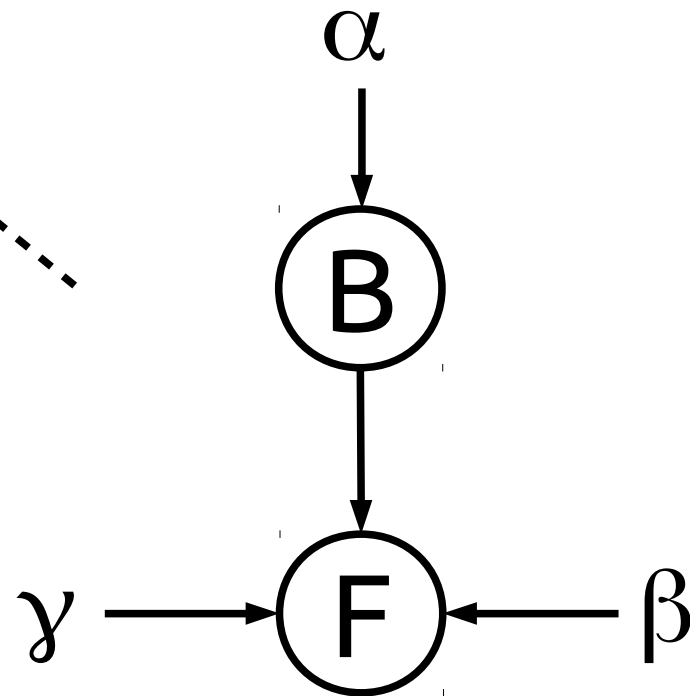
# Inference

- ♦ A possible interesting question can be: “What is the **probability** of extracting an **apple**?”
  - ♦ I.e. we are interested in  $P(F = 'a')$
- ♦ This probability is obtained by exploiting the following facts
  - ♦  $P(F)$  can be obtained from  $P(B,F)$  by **marginalization**
  - ♦  $P(B,F)$  – the joint probability – is given by the Bayesian Network, which also provides a **factorization** of it

# Inference

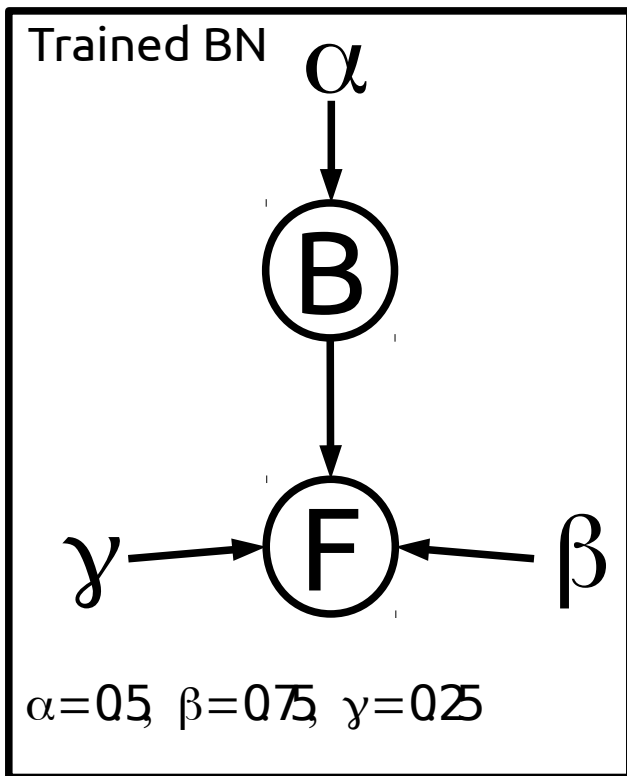
$$\begin{aligned} P(F) &= \sum_B P(F, B) \\ &= \sum_B P(B)P(F|B) \end{aligned}$$

The Bayesian Network provides a factorization of the joint probability



# Inference

$$\begin{aligned}
 P(F = 'a') &= \sum_B P(F = 'a', B) \\
 &= \sum_B P(B)P(F = 'a'|B) \\
 &= P(B = 'b')P(F = 'a'|B = 'b') \\
 &\quad + P(B = 'r')P(F = 'a'|B = 'r') \\
 &= \alpha(1 - \gamma) + (1 - \alpha)(1 - \beta) \\
 &= 0.5 \cdot 0.75 + 0.5 \cdot 0.25 \\
 &= 0.5
 \end{aligned}$$



$P(B)$

$$\begin{aligned}
 P(B = 'b') &= \alpha \\
 P(B = 'r') &= 1 - \alpha
 \end{aligned}$$

$P(F|B)$

$$\begin{aligned}
 P(F = 'o' \mid B = 'r') &= \beta \\
 P(F = 'a' \mid B = 'r') &= 1 - \beta \\
 P(F = 'o' \mid B = 'b') &= \gamma \\
 P(F = 'a' \mid B = 'b') &= 1 - \gamma
 \end{aligned}$$

# Inference

- **Exercise:** compute  $p(x = 5)$  within a Gaussian Mixture Model with 2 one-dimensional Gaussians

$$p(g = 1) = \pi_1 \quad p(g = 2) = \pi_2$$

$$p(x|g = 1) = \mathcal{N}(x|\mu_1, \sigma_1)$$

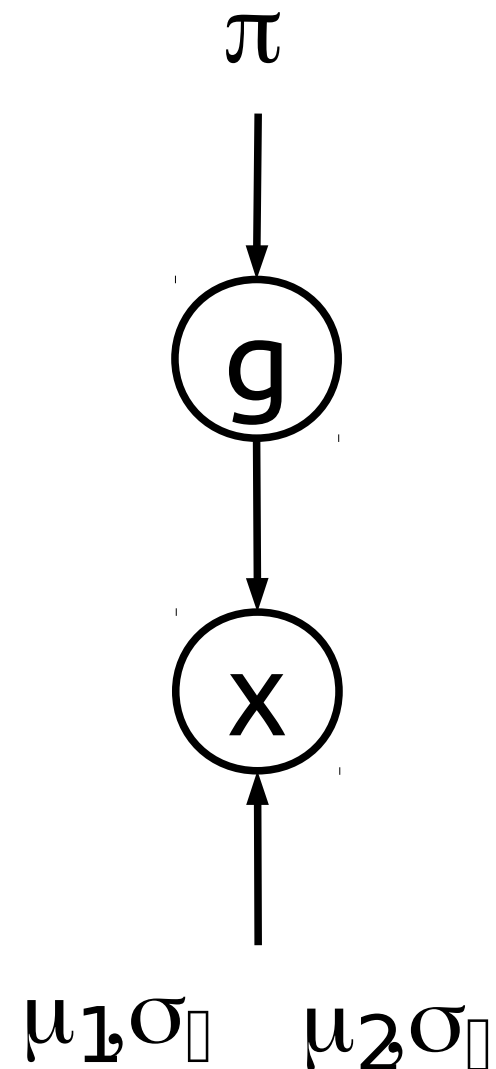
$$p(x|g = 2) = \mathcal{N}(x|\mu_2, \sigma_2)$$

## Trained GMM

$$\pi_1 = 0.3, \pi_2 = 0.7$$

$$\mu_1 = 4.2, \sigma_1 = 2$$

$$\mu_2 = 5.2, \sigma_2 = 1$$



# Inference

$$\begin{aligned}P(x = 5) &= \sum P(x = 5, g) \\&= \sum_g P(g)P(x = 5|g) \\&= P(g = 1)P(x = 5|g = 1) \\&\quad + P(g = 2)P(x = 5|g = 2) \\&= \pi_1 \mathcal{N}(x = 5|\mu_1, \sigma_1) + \pi_2 \mathcal{N}(x = 5|\mu_2, \sigma_2)\end{aligned}$$

**Knowing that**  $\mathcal{N}(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}$

$$\begin{aligned}P(x = 5) &= 0.3 \frac{1}{\sqrt{2\pi \cdot 2^2}} e^{-\frac{\|5-4.2\|^2}{2 \cdot 2^2}} + 0.7 \frac{1}{\sqrt{2\pi \cdot 1^2}} e^{-\frac{\|5-5.2\|^2}{2 \cdot 1^2}} \\&= 0.0721 + 0.2737 \\&= 0.3458\end{aligned}$$

# Inference

- ♦ Another interesting inference can be performed on the **hidden variables**
  - ♦ To understand the **causes**
- ♦ **Two boxes example:** knowing that I have extracted an apple, what is the probability that I've chosen the blue box?
  - ♦  $P(B = 'b' \mid F = 'a')$
- ♦ NOTE:  $P(B = 'b' \mid F = 'a')$  is different from the conditional of the Bayesian network –  $P(F = 'a' \mid B = 'b')$

# Inference

- ♦ The procedure is similar

$$P(B|F) = \frac{P(F, B)}{P(F)}$$

Definition of  
conditional probability

$$P(F, B) = P(B)P(F|B)$$

The joint probability  
(and its factorization)  
is given by the BN

$$P(F) = \sum_B P(F, B) = \sum_B P(B)P(F|B)$$

$P(F)$  is computed as  
before via  
marginalization

# Other inferences

- ♦ Inference needed during **learning**
  - ♦ Example: the E-M for mixture of Gaussians

posterior to be computed in the E-Step  
for the hidden variables

$$w_{i1} = p(y_i = 1 | \mathcal{X}, \theta^{(i-1)}) = p(y_i = \text{'blue'} | \mathcal{X}, \theta^{(i-1)})$$

- ♦ **Optimization:** which is the configuration of the hidden variables for which the joint probability is **maximum**? Which is the **most probable** configuration of the hidden variables?
  - ♦ Often called MAP (Maximum a Posteriori) estimation



# Inference: summary

- ♦ Inference is used to extract interesting information from the Bayesian Network
  - ♦ Summaries, causes, optimization
- ♦ In general, performing inference is not always so easy (e.g. integrals)

Example: a GMM with an infinite number of components

$$p(x) = \int_g p(x, g) dg$$

# Inference: summary

- ♦ Another aspect: the structural complexity of Bayesian Networks (e.g. cycles) may make the inference problem intractable
- ♦ (Again) Needed tradeoff: **computability** vs **descriptivity**

# Inference: summary

Many complex algorithms have been proposed to perform not trivial inference (not seen here):

- ♦ Exact inference (variable elimination, belief propagation – for trees, ..)
- ♦ Variational inference (mean field)
- ♦ Monte Carlo inference (Gibbs sampling)

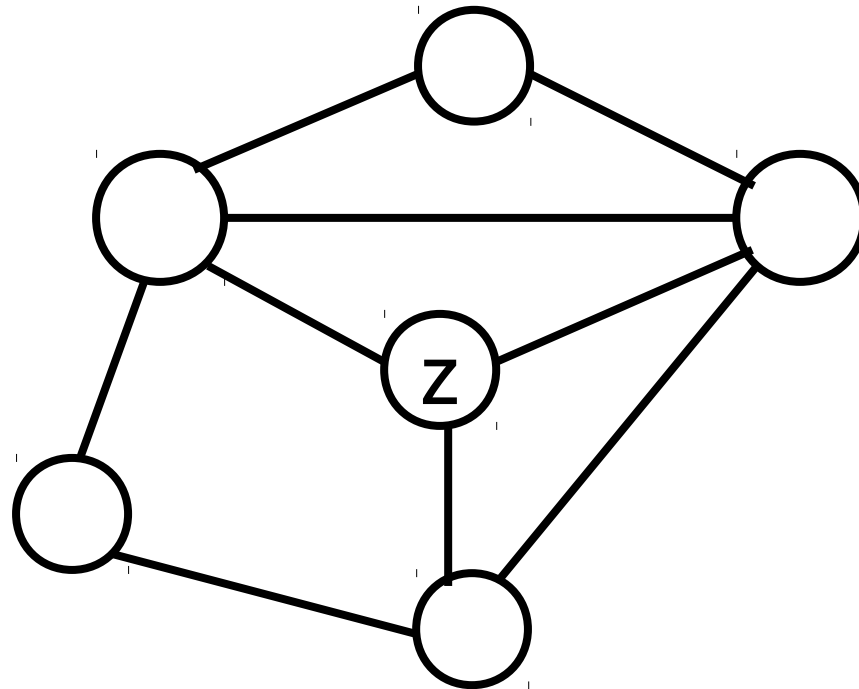
For more info see the Kevin Murphy's tutorial (further readings)

# Other Probabilistic Graphical Models

- ♦ There are other two families of Probabilistic Graphical Models:
  - ♦ Markov Random Fields
  - ♦ Factor Graphs

# Markov Random Fields

- ♦ **Undirected** graph of random variables
- ♦ Each variable is independent of all other variables, given its **neighbors**

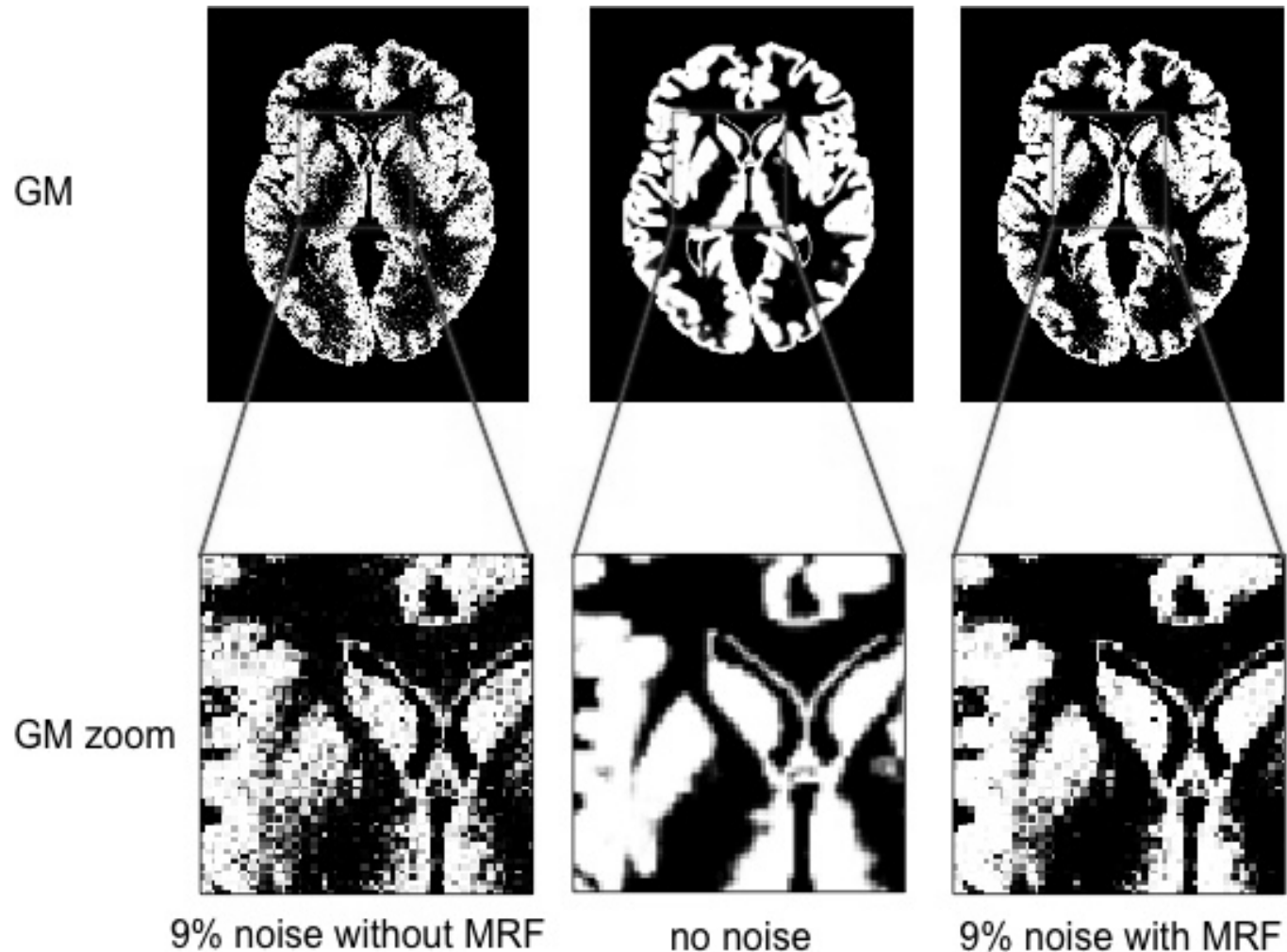


# Markov Random Fields

- ♦ Widely used to analyse **images**

They can model **continuity** (smoothness) and **spatial proximity**, crucial aspects in images

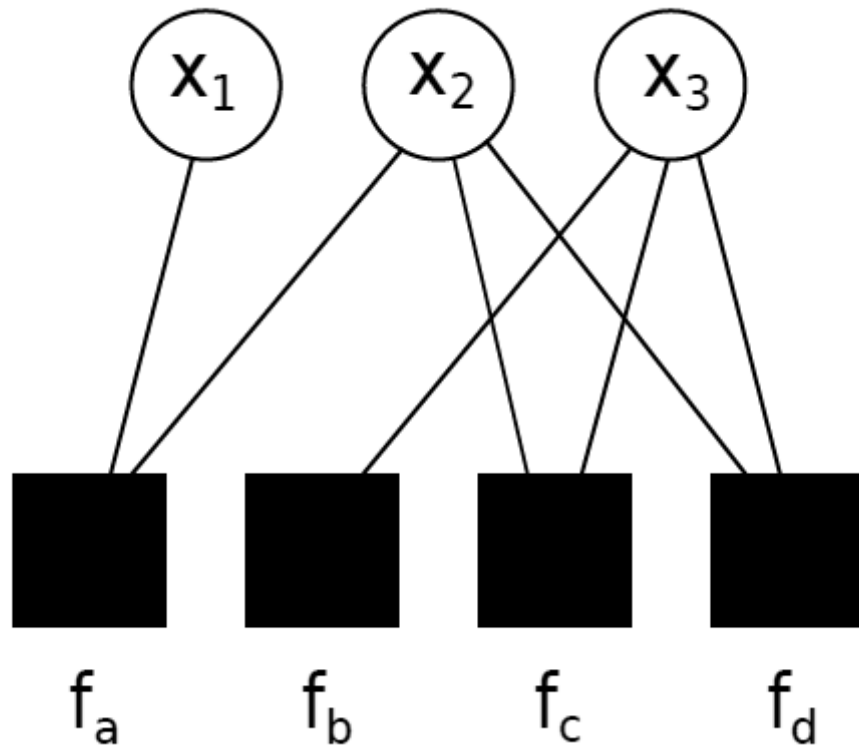
**Example:** noise removal



# Factor Graphs

- ♦ Less investigated class of probabilistic Graphical Models (introduced by Frey)
- ♦ Main idea: to express a global function of several variables as a collection of factors (local functions) over a subset of those variables
- ♦ The graph has two kinds of node:
  - ♦ Nodes for **variables**
  - ♦ Nodes for functions (called **factors**)

# Factor Graphs



$x_1, x_2, x_3$  are the **variables**

$f_a, f_b, f_c,$  and  $f_d$  are **factors**

This factor graph encodes the factorization of a function  $g(x_1, x_2, x_3)$  over all the variables

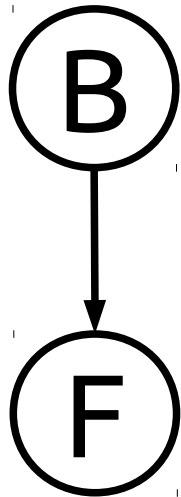
$$g(x_1, x_2, x_3) = f_a(x_1, x_2) f_b(x_3) f_c(x_2, x_3) f_d(x_2, x_3)$$



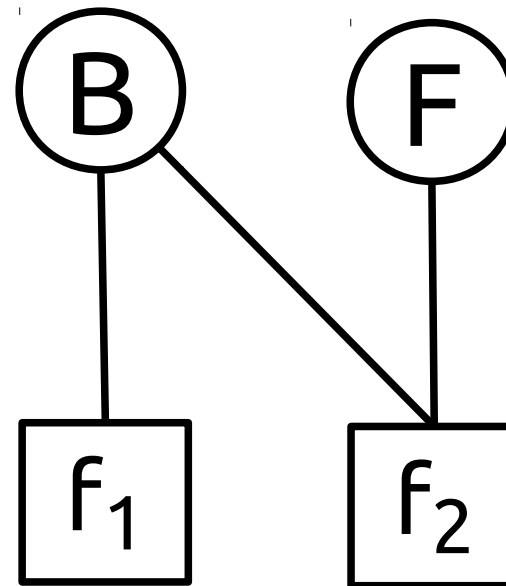
# Factor Graphs

- This formalism is more general than the Bayesian Network formalism

Every Bayesian Network can be written as a Factor Graph



$$P(B, F) = P(B)P(F|B)$$



$$f_1(B) = P(B)$$

$$f_2(B, F) = P(F|B)$$

$$g(B, F) = f_1(B)f_2(B, F)$$

$$= P(B)P(F|B)$$

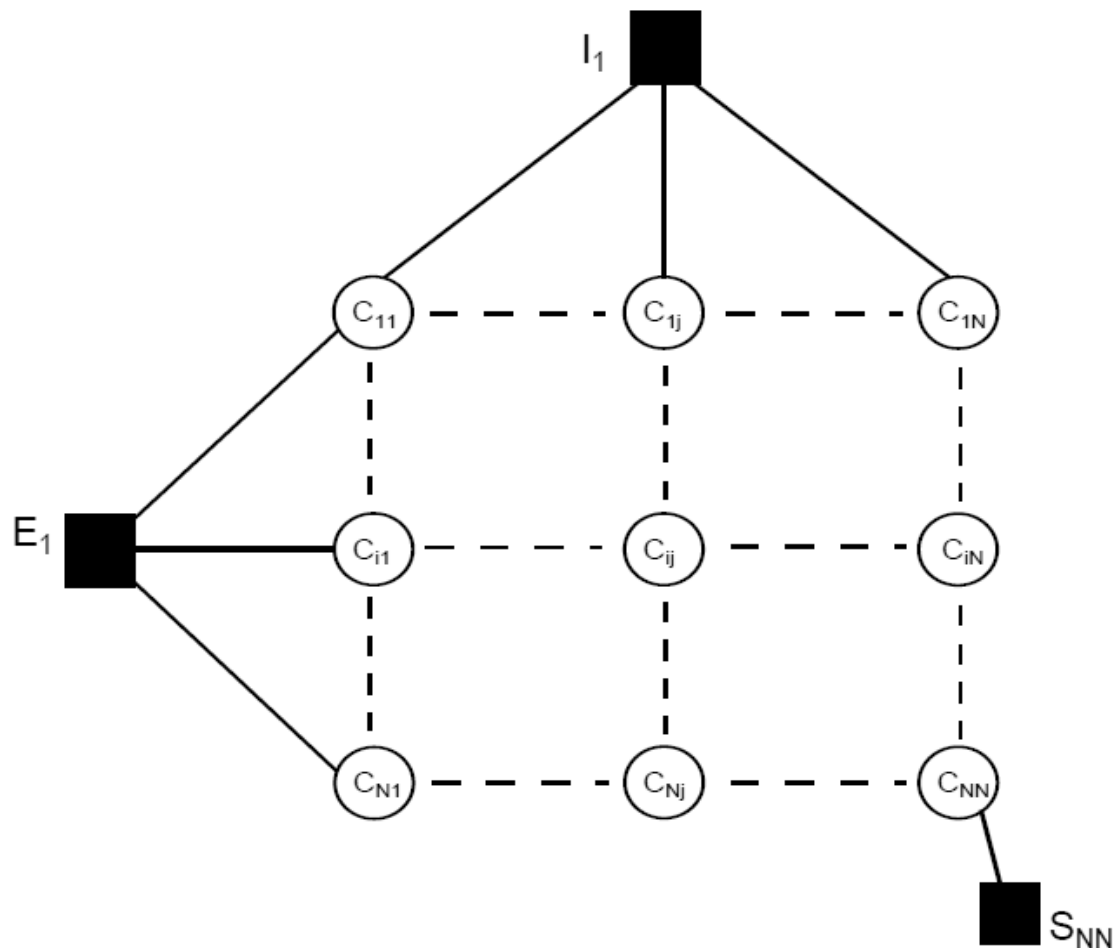
$$= P(B, F)$$

# Factor Graphs

- This formalism can be used also for “non probabilistic” functions

Example: **Affinity Propagation**  
algorithm for  
Clustering

[*Science* 315, 972–976  
(2007)]



# Conclusions

- ◆ Probabilistic Graphical Models represent a powerful tool to model structured objects
  - ◆ Capability to capture the complexity
  - ◆ Different information can be extracted
  - ◆ Many algorithms / tools to perform training, inference, optimization
- ◆ Constraint: tradeoff between **computability** and **descriptivity**

# Further Readings

- ♦ B. Frey and N. Jojic: “A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models”, IEEE Trans. On Pattern Analysis and Machine Intelligence, 27(9), 2005
- ♦ Kevin P. Murphy: “An introduction to graphical models”, available from [http://www.cs.ubc.ca/~murphyk/Papers/intro\\_gm.pdf](http://www.cs.ubc.ca/~murphyk/Papers/intro_gm.pdf)
- ♦ J.A. Bilmes: “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models”, TR-97-021