

# *Artificial Intelligence*

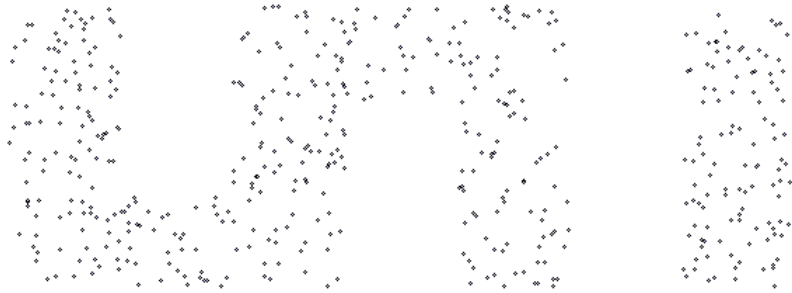
## *Unsupervised Learning*

Marco Piastra

An aside:  
The K-means algorithm  
*(i.e. alternate optimization)*

# Vector quantization

184

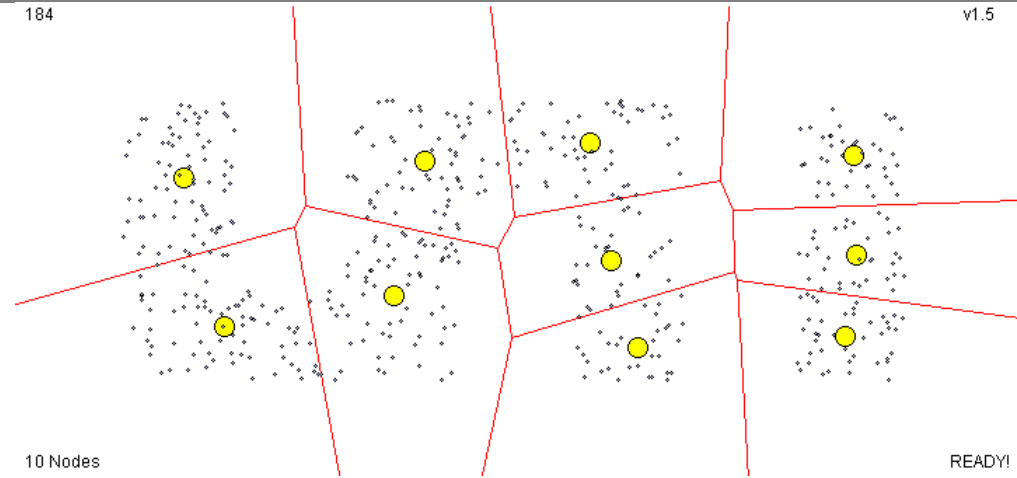


10 Nodes

Original data

v1.5

184



READY! 10 Nodes

Quantization (compression via prototypes)

v1.5

READY!

The basic idea is to replace each real-valued vector  $\mathbf{x} \in \mathbb{R}^d$  with a discrete symbol  $\mathbf{w}_j \in \mathbb{R}^d$  which belongs to a codebook of  $k$  prototypes  $\theta := \{\mathbf{w}_1, \dots, \mathbf{w}_k\}$

Each data vector is encoded by using the index of the most similar prototype, where similarity is measured in terms, for instance, of Euclidean distance:

$$w(\mathbf{x}) := \operatorname{argmin}_{\mathbf{w}_j} \|\mathbf{x} - \mathbf{w}_j\|$$

For instance, part of mpeg4 and QuickTime (Apple) video compression algorithms work in this way and so does the Ogg Vorbis audio compression algorithm

# *k-means* (Generalized Lloyd's Algorithm – Vector quantization)

Given a set  $D := \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  of observations (i.e. vectors in  $\mathbb{R}^d$ )  
and a set  $\theta := \{\mathbf{w}_1, \dots, \mathbf{w}_k\}$  of  $k$  prototypes (i.e. vectors in  $\mathbb{R}^d$ )

Clustering problem: find an assignment function  $w : D \rightarrow \theta$   
such that the objective (loss) function:

$$J(D, \theta) := \frac{1}{2} \sum_{i=1}^N \|\mathbf{x}_i - w(\mathbf{x}_i)\|^2$$

is minimized.

# *k-means* (Generalized Lloyd's Algorithm – Vector quantization)

## **k-means algorithm:**

- 1) Position the  $k$  prototypes at random
- 2) Assign each observation to its closest prototype

$$w(\mathbf{x}_i) := \operatorname{argmin}_{\mathbf{w}_j} \|\mathbf{x}_i - \mathbf{w}_j\|$$

- 3) Position each prototype at the *centroid* of the observations assigned to it

$$\mathbf{w}_j = \frac{1}{|D(\mathbf{w}_j)|} \sum_{\mathbf{x}_i \in D(\mathbf{w}_j)} \mathbf{x}_i \quad \text{where } D(\mathbf{w}_j) := \{\mathbf{x}_i \in D \mid w(\mathbf{x}_i) = \mathbf{w}_j\}$$

- 4) Unless no prototype was moved in step 3), go back to step 2)

This algorithm converges to a local minimum of  $J(D, \theta)$

# *k-means* (Generalized Lloyd's Algorithm – Vector quantization)

Why does the algorithm work: *alternate optimization* (also 'coordinate descent')

Step 2): Assign observations while keeping the  $k$  prototype fixed

$$w(\mathbf{x}_i) := \operatorname{argmin}_{\mathbf{w}_j} \|\mathbf{x}_i - \mathbf{w}_j\|$$

which minimizes each of the terms in  $J(D, \theta) := \frac{1}{2} \sum_{i=1}^N \|\mathbf{x}_i - w(\mathbf{x}_i)\|^2$

Step 3): Reposition the  $k$  prototypes while keeping the assignments fixed

$$J(D, \theta) := \frac{1}{2} \sum_{i=1}^N \|\mathbf{x}_i - w(\mathbf{x}_i)\|^2 = \frac{1}{2} \sum_j \sum_{D(\mathbf{w}_j)} (\mathbf{x}_i - \mathbf{w}_j)^2$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}_j} J(D, \theta) &= \frac{\partial}{\partial \mathbf{w}_j} \frac{1}{2} \sum_{D(\mathbf{w}_j)} (\mathbf{x}_i - \mathbf{w}_j)^2 = \frac{\partial}{\partial \mathbf{w}_j} \frac{1}{2} \sum_{D(\mathbf{w}_j)} (\mathbf{x}_i - \mathbf{w}_j)^T (\mathbf{x}_i - \mathbf{w}_j) \\ &= \frac{\partial}{\partial \mathbf{w}_j} \frac{1}{2} \sum_{D(\mathbf{w}_j)} (\mathbf{x}_i^2 + \mathbf{w}_j^2 - 2 \mathbf{x}_i^T \mathbf{w}_j) = \sum_{D(\mathbf{w}_j)} (\mathbf{w}_j - \mathbf{x}_i) \end{aligned}$$

then, by imposing  $\frac{\partial}{\partial \mathbf{w}_j} J(D, \theta) = 0$  we obtain

$$\mathbf{w}_j = \frac{1}{|D(\mathbf{w}_j)|} \sum_{D(\mathbf{w}_j)} \mathbf{x}_i$$

# *k-means* (Generalized Lloyd's Algorithm – Vector quantization)

## Discussion of the *k-means* algorithm

- a) At each step of the algorithm  $J(D, \theta)$  cannot increase: only decrease or stay equal
- b) The algorithm is a variant of a *gradient descent*, in which at each step the *gradient descent* is performed on one subset of variables only
- c) It must reach a *fixed point*, where both gradients vanish
- d) But the only guarantee is that the algorithm reaches a local minimum  
(*unless it gets stuck in a saddle point*)

# *k-means* (Generalized Lloyd's Algorithm – Vector quantization)

Given a set  $D := \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  of observations (i.e. vectors in  $\mathbb{R}^d$ )  
and a set  $\theta := \{\mathbf{w}_1, \dots, \mathbf{w}_k\}$  of  $k$  prototypes (i.e. vectors in  $\mathbb{R}^d$ )

## **Voronoi cell:**

$$V(\mathbf{w}_j) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x} - \mathbf{w}_j\| \leq \|\mathbf{x} - \mathbf{w}_l\|, \forall l \neq j\}$$

**Voronoi tessellation:** the complex of all Voronoi cells of  $\theta$

## **Algorithm** (rewritten):

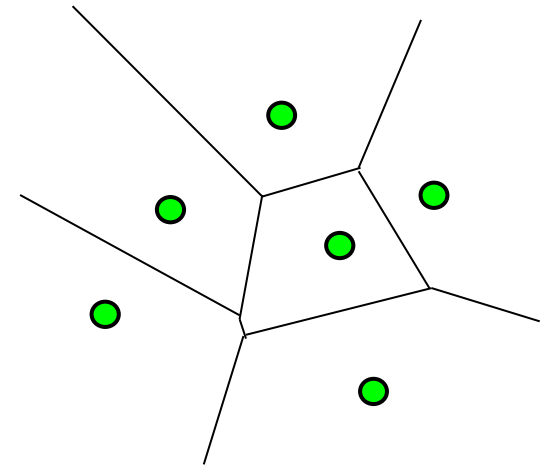
- 1) Position the  $k$  prototypes at random
- 2) Assign each observation to its Voronoi cell

$$w(\mathbf{x}_i) := \mathbf{w}_j \mid \mathbf{x}_i \in V(\mathbf{w}_j)$$

- 3) Position each prototype at the *centroid* of the observations in its Voronoi cell

$$\mathbf{w}_j = \frac{1}{|\{\mathbf{x}_i \in V(\mathbf{w}_j)\}|} \sum_{\{\mathbf{x}_i \in V(\mathbf{w}_j)\}} \mathbf{x}_i$$

- 4) Unless no prototype was moved in step 3), go back to step 2)

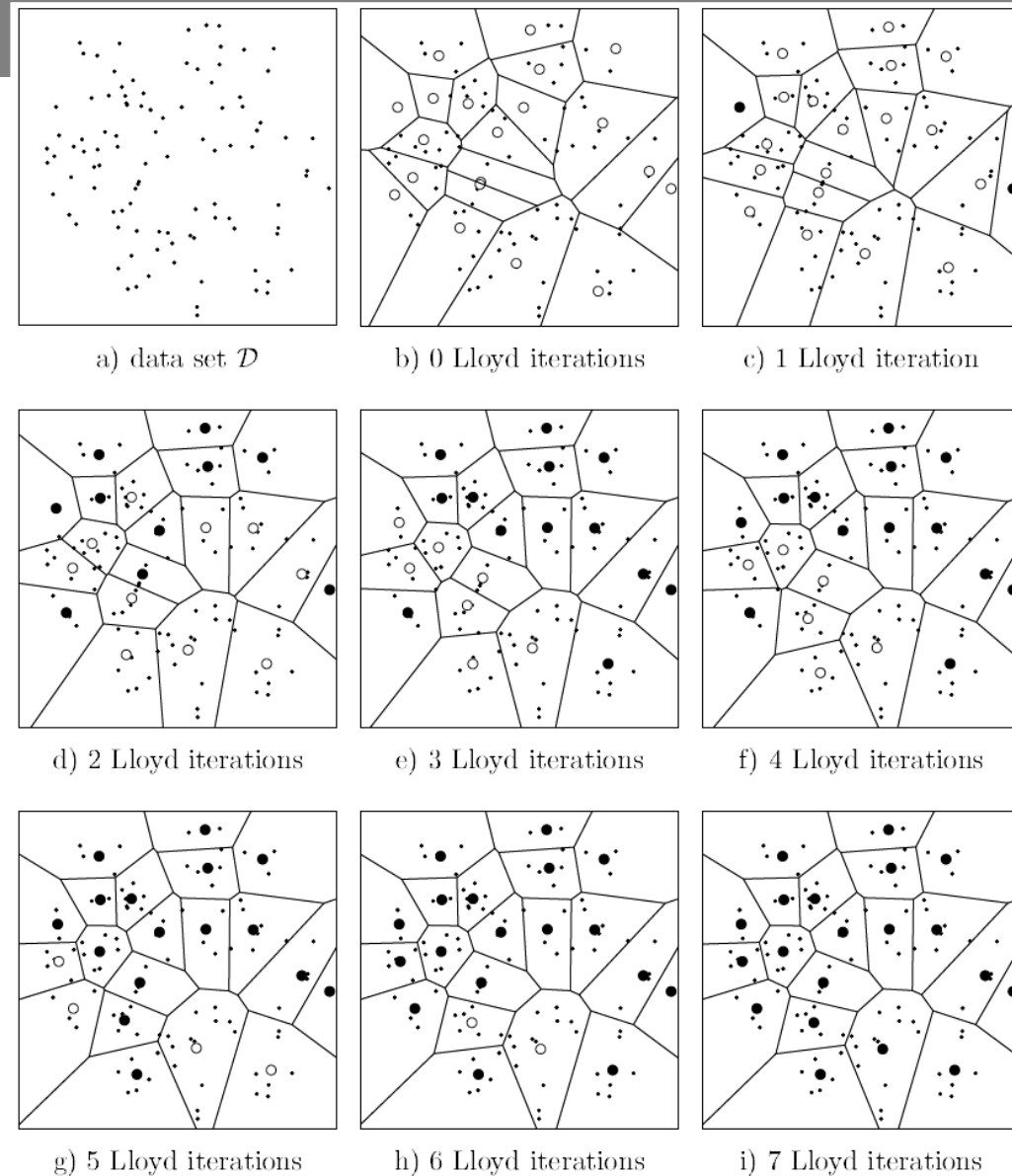




# *k*-means

An example run of the algorithm

The landmarks (empty circles) become black when they cease to move



# The Expectation–Maximization (EM) algorithm

# Expected value of a random variable

(also *expectation*)

Basic definition

$$\mathbb{E}_X[X] := \sum_{x \in \mathcal{X}} x P(X = x)$$

More concise notation

$$\mathbb{E}[X] := \sum_{x \in \mathcal{X}} x P(x)$$

A linear operator

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

$$\mathbb{E}[cX] = c\mathbb{E}[X]$$

Continuous case

$$\mathbb{E}[X] := \int_{x \in \mathcal{X}} x p(x) dx$$

Conditional expectation

$$\mathbb{E}_X[X|Y = y] = \mathbb{E}[X|Y = y] := \sum_{x \in \mathcal{X}} x P(X = x|Y = y)$$

Iterated expectation (*see Wikipedia*)

$$\mathbb{E}_X[X] = \mathbb{E}_Y[\mathbb{E}_X[X|Y]]$$

# Joint Expected Value

The **expected value** of a function  $f$  of a set of random variables is  $\{X_i\}$

$$\mathbb{E}[f(\{X_i\})] := \sum_{\{X_i\}} f(\{X_i\}) P(\{X_i\})$$

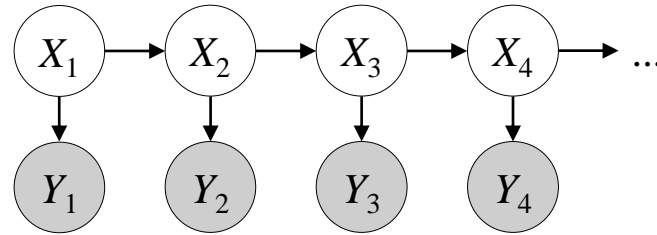
*the sum is over all possible combinations of values of the random variables*

(Unless specified otherwise, the  $\mathbb{E}$  operator acts over *all* the random variables enclosed)

*The extension to the continuous case is obvious*

# Incomplete observations

## Example: 'Hidden Markov' model



Terminology:

*hidden = latent = always unobserved*

*missing = unobserved (in a data set)*

Typically,  $X_i$  nodes are *hidden*,  
i.e. *non-observables*

$$P(\{Y_i\}, \{X_j\}) = P(X_1)P(Y_1 | X_1) \prod_{i=2}^n P(X_i | X_{i-1})P(Y_i | X_i) \quad \text{Joint distribution}$$

### ■ Problem

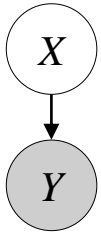
*MLE* of parameters  $\theta$  starting from *partial* observations of the  $\{X_i\}$  variables only

In other terms, this is the *MLE* of the *likelihood function*

$$L(\theta|D) = P(D|\theta) = P(\{\{Y_j^{(i)}\}\}|\theta) = \sum_{\{X_j^{(i)}\}} P(\{\{Y_j^{(i)}, X_j^{(i)}\}\}|\theta)$$

Note that the model (= the probability function) and the (partial) observations are known,  
while the values of some variables are hidden

# Expectation Maximization: a preliminary example



**a** Maximum likelihood

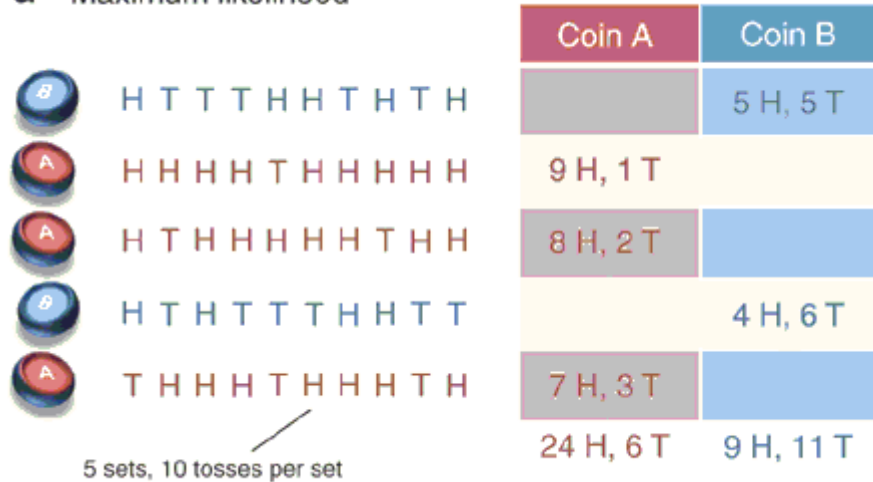


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

$$\hat{\theta}_A = \frac{24}{24 + 6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9 + 11} = 0.45$$

- An experiment with two coins

At each step, one coin is selected at random (*with equal probability*) and then tossed ten times

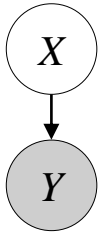
Random variables:  $Y$  number of heads,  $X$  selected coin (i.e A or B)

Parameters to be learnt:  $\theta = \{\theta_A, \theta_B\}$  probabilities of landing on head of A and B

When the results are fully observable, by MLE:

$$\theta_A^* = \frac{N_{Y=1, X=A}}{N_{X=A}} \quad \theta_B^* = \frac{N_{Y=1, X=B}}{N_{X=B}}$$

# Expectation Maximization: a preliminary example



a Maximum likelihood

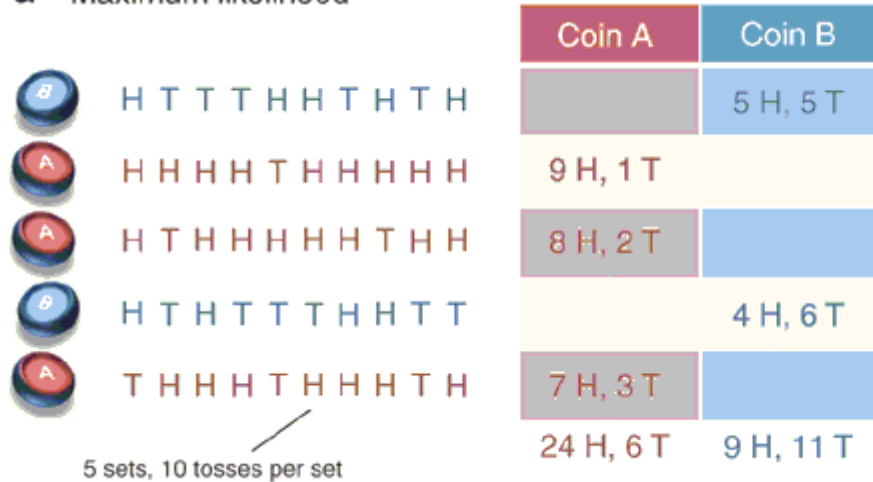


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

$$\hat{\theta}_A = \frac{24}{24 + 6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9 + 11} = 0.45$$

- An experiment with two coins

At each step, one coin is selected at random (*with equal probability*) and then tossed ten times

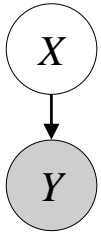
Random variables:  $Y$  number of heads,  $X$  selected coin (i.e A or B)

Parameters to be learnt:  $\theta = \{\theta_A, \theta_B\}$  probabilities of landing on head of A and B

- What if  $X$  is *hidden* (= latent, = unobserved)?

*The results of each sequence of coin tosses are known, but not the coin selected*

# Expectation Maximization: a preliminary example



**a** Maximum likelihood

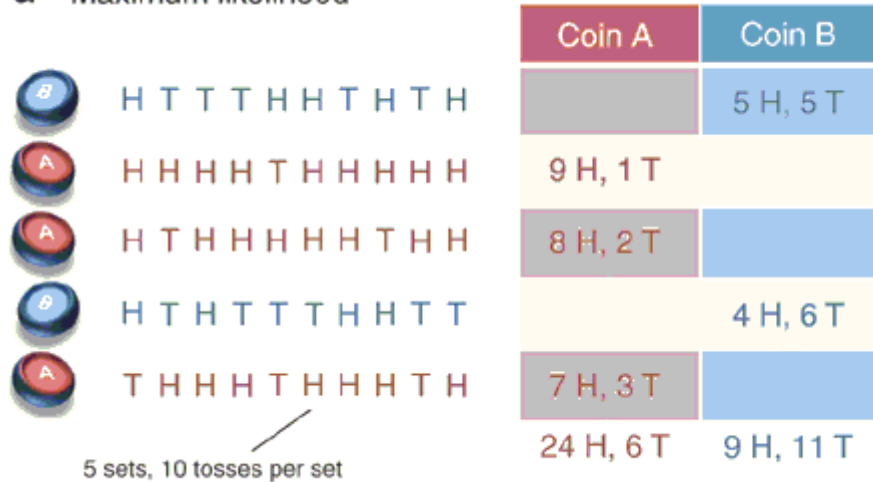


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

$$\hat{\theta}_A = \frac{24}{24 + 6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9 + 11} = 0.45$$

- What if  $X$  is *hidden* (= *latent*, = *unobserved*)?

*Likelihood*

$$P(D | \theta) = P(\{Y^{(i)}\} | \theta) = \sum_{\{X^{(i)}\}} P(\{(Y^{(i)}, X^{(i)})\} | \theta)$$

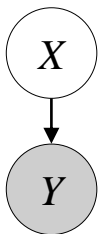
*MLE*

$$\theta^* := \operatorname{argmax}_{\theta} \sum_{\{X^{(i)}\}} P(\{(Y^{(i)}, X^{(i)})\} | \theta)$$

*\* This optimization is intractable, in general*



# Expectation Maximization: a preliminary example



**a** Maximum likelihood

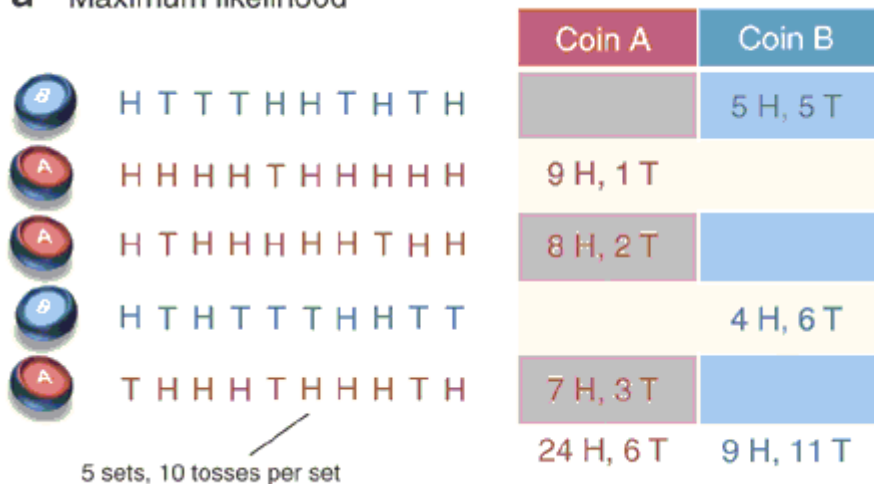


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

$$\hat{\theta}_A = \frac{24}{24 + 6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9 + 11} = 0.45$$

- What if  $Z$  is *hidden* (= *latent*, = *unobserved*)?

*Intuitive idea: use expected values for unobserved variables*

- Define an initial (random) guess  $\hat{\theta}^{(0)}$
- Compute  $Q_i(X^{(i)}) := P(X^{(i)} | Y^{(i)}; \hat{\theta}^{(t)})$
- Maximize

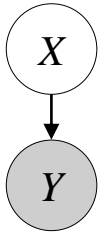
*E-step*

$$\hat{\theta}^{(t+1)} = \operatorname{argmax}_{\theta} \sum_i \mathbb{E}_{Q_i(X^{(i)})} [Y^{(i)} | X^{(i)}; \hat{\theta}^{(t)}]$$

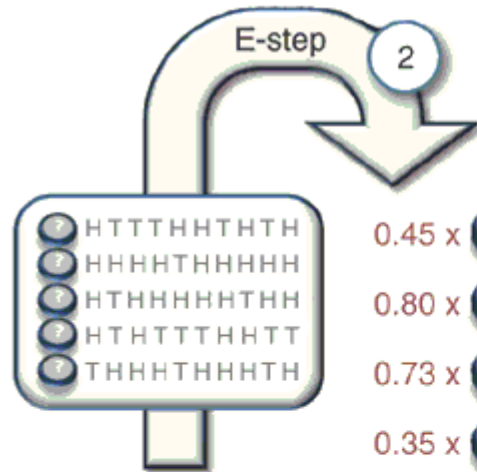
*M-step*

- Unless some convergence criterion has been met, go to step 2.

# Expectation Maximization: a preliminary example



	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T



$$\hat{\theta}_A^{(0)} = 0.60$$

$$\hat{\theta}_B^{(0)} = 0.50$$

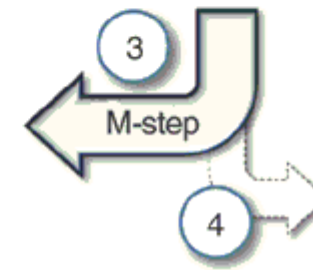
Coin A	Coin B
$\approx 2.2$ H, $2.2$ T	$\approx 2.8$ H, $2.8$ T
$\approx 7.2$ H, $0.8$ T	$\approx 1.8$ H, $0.2$ T
$\approx 5.9$ H, $1.5$ T	$\approx 2.1$ H, $0.5$ T
$\approx 1.4$ H, $2.1$ T	$\approx 2.6$ H, $3.9$ T
$\approx 4.5$ H, $1.9$ T	$\approx 2.5$ H, $1.1$ T
$\approx 21.3$ H, $8.6$ T	$\approx 11.7$ H, $8.4$ T

Initial random estimate of  $\hat{\theta}_A, \hat{\theta}_B$



$$\hat{\theta}_A^{(1)} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71$$

$$\hat{\theta}_B^{(1)} \approx \frac{11.7}{11.7 + 8.4} \approx 0.58$$

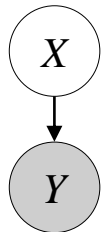


$$\hat{\theta}_A^{(10)} \approx 0.80$$

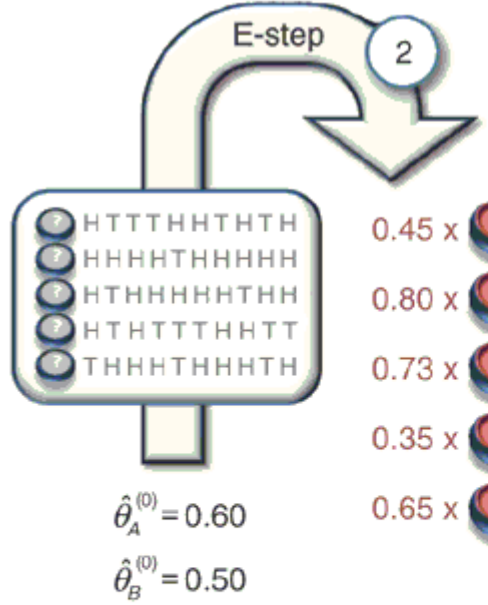
$$\hat{\theta}_B^{(10)} \approx 0.52$$

Converged?

# Expectation Maximization: a preliminary example



	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T



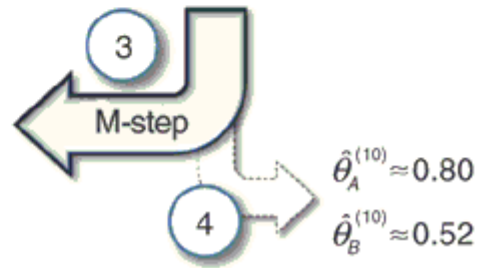
Coin A	Coin B
≈ 2.2 H, 2.2 T	≈ 2.8 H, 2.8 T
≈ 7.2 H, 0.8 T	≈ 1.8 H, 0.2 T
≈ 5.9 H, 1.5 T	≈ 2.1 H, 0.5 T
≈ 1.4 H, 2.1 T	≈ 2.6 H, 3.9 T
≈ 4.5 H, 1.9 T	≈ 2.5 H, 1.1 T
≈ 21.3 H, 8.6 T	≈ 11.7 H, 8.4 T

Initial random estimate of  $\hat{\theta}_A, \hat{\theta}_B$

1

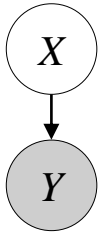
$$\hat{\theta}_A^{(1)} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71$$

$$\hat{\theta}_B^{(1)} \approx \frac{11.7}{11.7 + 8.4} \approx 0.58$$



Converged?

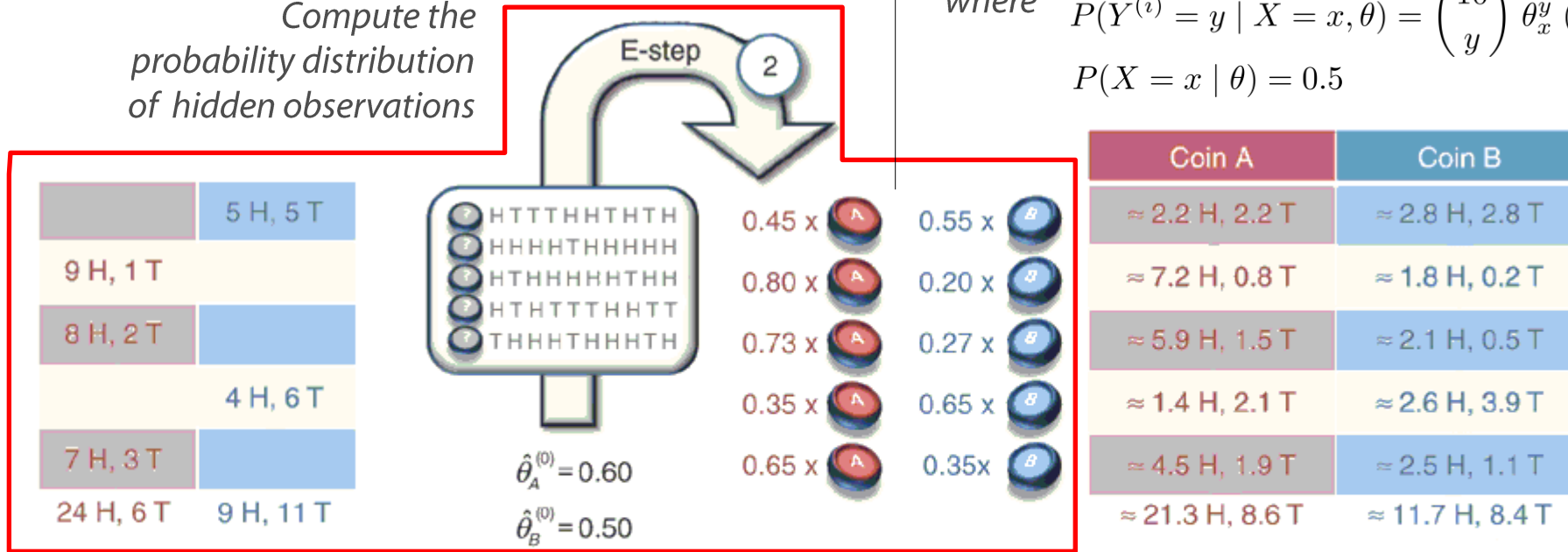
# Expectation Maximization: a preliminary example



$$Q_i(X = x \mid Y^{(i)} = y, \theta) := \frac{P(Y^{(i)} = y \mid X = x, \theta)P(X = x \mid \theta)}{\sum_y P(Y^{(i)} = y \mid X = x, \theta)P(X = x \mid \theta)}$$

where  $P(Y^{(i)} = y \mid X = x, \theta) = \binom{10}{y} \theta_x^y (1 - \theta_x)^{10-y}$   
 $P(X = x \mid \theta) = 0.5$

Compute the probability distribution of hidden observations

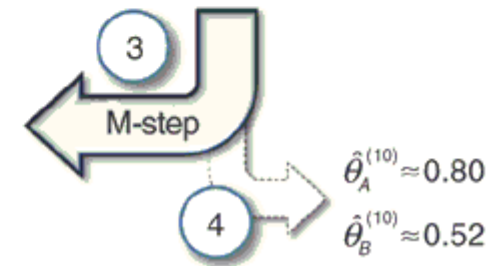


Initial random estimate of  $\hat{\theta}_A, \hat{\theta}_B$



$$\hat{\theta}_A^{(1)} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71$$

$$\hat{\theta}_B^{(1)} \approx \frac{11.7}{11.7 + 8.4} \approx 0.58$$

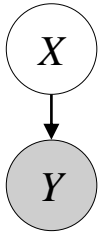


$$\hat{\theta}_A^{(10)} \approx 0.80$$

$$\hat{\theta}_B^{(10)} \approx 0.52$$

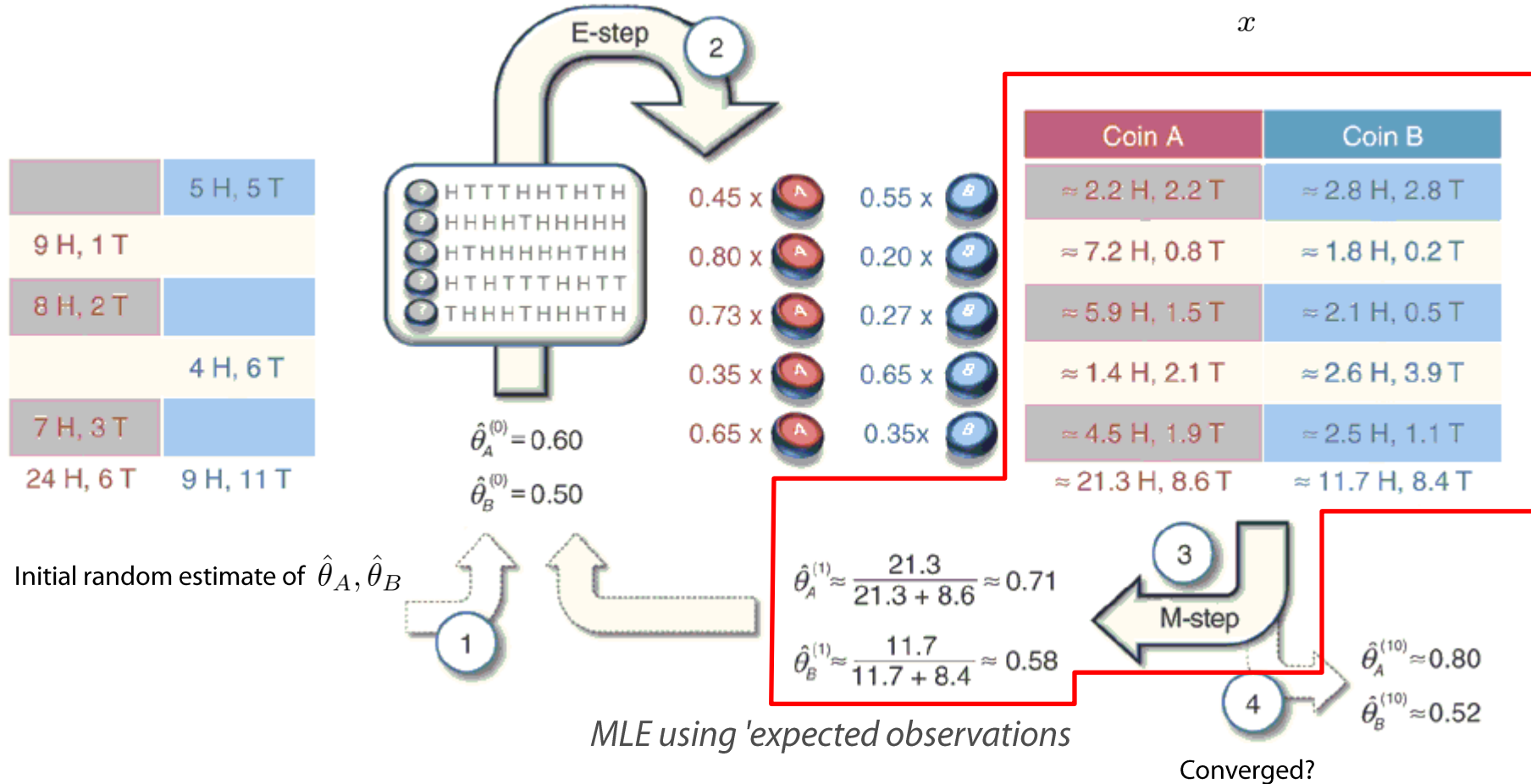
Converged?

# Expectation Maximization: a preliminary example



Use 'expected observations' instead of actual observations to update ML estimations

$$\mathbb{E}_X [ y \mid Y^{(i)} = y, X, \theta ] = \sum_x y Q_i(X = x \mid Y^{(i)} = y, \theta) = y$$



# Convergence of the EM algorithm (in the discrete case)

# An aside: Jensen's inequality

A relationship between probability and geometry

When  $f$  is convex function

$$f(\mathbb{E}[\{X_i\}]) \leq \mathbb{E}[f(\{X_i\})]$$

$f$  is **convex** when for any two points  $\mathbf{p}_i$  and  $\mathbf{p}_j$  the segment  $(\mathbf{p}_i - \mathbf{p}_j)$  is not below  $f$

That is, when

$$\lambda f(x_i) + (1 - \lambda)f(x_j) \geq f(\lambda x_i + (1 - \lambda)x_j), \forall \lambda \in [0, 1]$$

Furthermore,  $f$  is **strictly convex** when

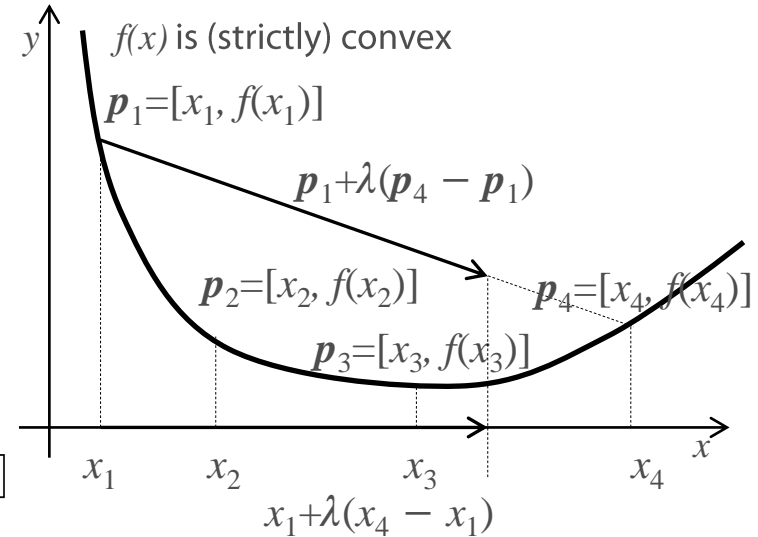
$$\lambda f(x_i) + (1 - \lambda)f(x_j) > f(\lambda x_i + (1 - \lambda)x_j), \forall \lambda \in (0, 1)$$

Corollary:

when  $f$  is *strictly convex*, if and only if all the variables in  $\{X_i\}$  are constant it is true that

$$f(\mathbb{E}[\{X_i\}]) \leq \mathbb{E}[f(\{X_i\})]$$

Dual results also hold for concave functions



# An aside: Jensen's inequality

A relationship between probability and geometry

When  $f$  is convex function

$$f(\mathbb{E}[\{X_i\}]) \leq \mathbb{E}[f(\{X_i\})]$$

To see this, consider

$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3 + \lambda_4 p_4$$

i.e. a **linear combination** of  $p_i$  points

This is an **affine** combination if  
and it is a **convex** combination if also

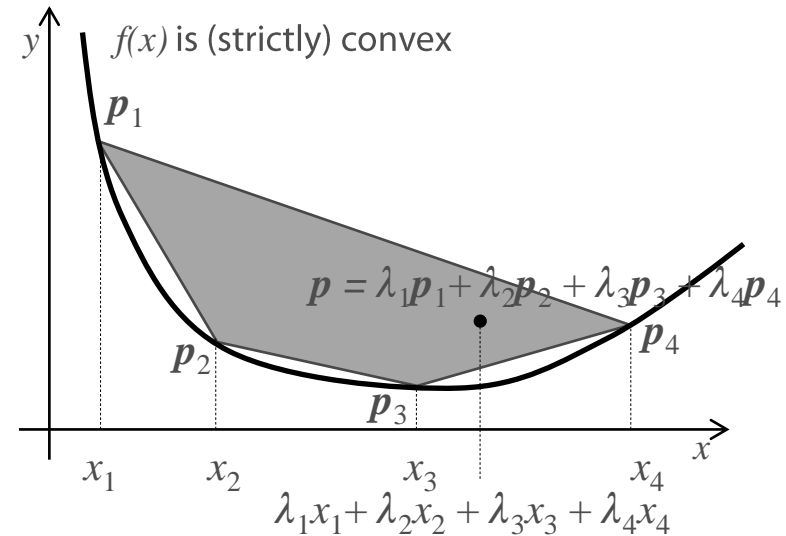
$$\begin{aligned} \sum \lambda_i &= 1 \\ \lambda_i &\geq 0, \forall i \end{aligned}$$

When the  $\lambda_i$  define a probability, then  $p$  is a **convex combination** of  $p_i$  points

Any convex combination of  $p_i$  points lies inside their **convex hull** (see figure)  
and therefore above  $f$  :

$$\sum_i \lambda_i f(x_i) \geq f\left(\sum_i \lambda_i x_i\right)$$

*Corollary: the only way to make the convex hull be on  $f$  is to shrink it to a single point (i.e. the Jensen's corollary)*





# Incomplete observations

*Likelihood function with hidden random variables*

$$L(\theta|D) = P(D|\theta) = \prod_m P(D^{(m)}|\theta)$$

$$\ell(\theta|D) = \sum_m \log P(D^{(m)}|\theta) = \sum_m \log \sum_{\{Z_i\}} P(D^{(m)}, \{X_i\}|\theta_k)$$

*Arbitrary probability distributions*

$$= \sum_m \log \sum_{\{X_i\}} Q^{(m)}(\{X_i\}) \frac{P(D^{(m)}, \{X_i\}|\theta)}{Q^{(m)}(\{X_i\})}$$

*Jensen's inequality: log is concave*

$$= \sum_m \log \mathbb{E}_{Q^{(m)}(\{X_i\})} \left[ \frac{P(D^{(m)}, \{X_i\}|\theta)}{Q^{(m)}(\{X_i\})} \right] \geq \sum_m \mathbb{E}_{Q^{(m)}(\{X_i\})} \left[ \log \frac{P(D^{(m)}, \{X_i\}|\theta)}{Q^{(m)}(\{X_i\})} \right]$$

$$= \sum_m \sum_{\{X_i\}} Q^{(m)}(\{X_i\}) \log \frac{P(D^{(m)}, \{X_i\}|\theta)}{Q^{(m)}(\{X_i\})}$$

# Expectation–Maximization (EM) Algorithm

*Alternate optimization (coordinate ascent)*

Log-likelihood function:

$$\ell(\theta|D) \geq \sum_m \sum_{\{X_i\}} Q^{(m)}(\{X_i\}) \log \frac{P(D^{(m)}, \{X_i\}|\theta)}{Q^{(m)}(\{X_i\})}$$

*This inequality becomes equality when this term is constant (see Jensen's corollary)*

1) Keep  $\theta$  constant, define  $Q^{(m)}(\{Z_i\})$  so that the right side of the inequality is maximized

$$Q^{(m)}(\{X_i\}) := \frac{P(D^{(m)}, \{X_i\}|\theta)}{\sum_{\{X_i\}} P(D^{(m)}, \{X_i\}|\theta)} = \frac{P(D^{(m)}, \{X_i\}|\theta)}{P(D^{(m)}|\theta)} = P(\{X_i\}|D^{(m)}, \theta) =: p_{\{X_i\}}^{(m)}$$

*These numbers can be computed from the graphical model (i.e. as an inference step)*

2) Then maximize the log-likelihood while keeping  $Q^{(m)}(\{Z_i\})$  constant

$$\theta^* = \operatorname{argmax}_{\theta} \sum_m \sum_{\{X_i\}} p_{\{X_i\}}^{(m)} \log \frac{P(D^{(m)}, \{X_i\}|\theta)}{p_{\{X_i\}}^{(m)}}$$

*This is also called the entropy of  $Q^{(m)}(\{X_i\})$  (i.e. a constant measure of the distribution)*

$$= \operatorname{argmax}_{\theta} \sum_m \left( \sum_{\{X_i\}} p_{\{X_i\}}^{(m)} \log P(D^{(m)}, \{X_i\}|\theta) - \sum_{\{X_i\}} p_{\{X_i\}}^{(m)} \log p_{\{X_i\}}^{(m)} \right)$$

$$= \operatorname{argmax}_{\theta} \sum_m \sum_{\{X_i\}} p_{\{X_i\}}^{(m)} \log P(D^{(m)}, \{X_i\}|\theta)$$

# Expectation– Maximization (EM) Algorithm

*Alternate optimization (coordinate ascent)*

Log-likelihood function and its estimator:

$$\ell(\theta|D) \geq \sum_m \sum_{\{X_i\}} Q^{(m)}(\{X_i\}) \log \frac{P(D^{(m)}, \{X_i\}|\theta)}{Q^{(m)}(\{X_i\})}$$

**Algorithm:**

- 1) Assign the  $\theta$  at random
- 2) (*E-step*) Compute the probabilities

$$p_{\{X_i\}}^{(m)} = Q^{(m)}(\{X_i\}) = P(\{X_i\}|D^{(m)}, \theta)$$

- 3) (*M-step*) Compute a new estimate of  $\theta$

$$\theta^* = \operatorname{argmax}_{\theta} \sum_m \sum_{\{X_i\}} p_{\{X_i\}}^{(m)} \log P(D^{(m)}, \{X_i\}|\theta)$$

- 4) Go back to step 2) until some convergence criterion is met

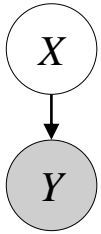
*The algorithm converges to a local maximum of the log-likelihood*

*The effectiveness of algorithm depends on the form of  $P(\{X_i\}|D^{(m)}, \theta)$  (see step3)*

*In particular, when this distribution is exponential... (e.g. Gaussian – see next slide)*

An aside:  
the EM algorithm in the continuous case  
(Mixture of Gaussians)

# EM Algorithm: mixture of Gaussians



## Model:

The hidden variable  $X$  has  $k$  possible values, the observable variable  $Y$  is a point in  $\mathbb{R}^d$

$$P(X = k) := \phi_k$$

Multivariate normal distribution

$$P(Y = y|X = k) = \mathcal{N}(y; \mu_k, \Sigma_k) := (2\pi)^{-d/2} (\det \Sigma_k)^{-1/2} \exp\left(-\frac{1}{2}(y - \mu_k)^T \Sigma_k^{-1} (y - \mu_k)\right)$$

i.e. the condition probabilities are normal distributions

The observations are a set  $D = \{y^{(1)}, \dots, y^{(N)}\}$  of points in  $\mathbb{R}^d$

## Algorithm:

- 1) For each value  $k$ , assign  $\phi_k, \mu_k$  and  $\Sigma_k$  at random
- 2) (*E-step*) For all the  $y^{(m)}$  in  $D$  compute the probabilities
$$p_k^{(m)} = P(X = k|y^{(m)}, \phi_k, \mu_k, \Sigma_k) = \phi_k \cdot \mathcal{N}(y^{(m)}; \mu_k, \Sigma_k)$$
- 3) (*M-step*) Compute the new estimates for the parameters

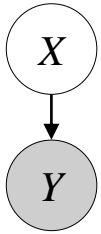
$$\phi_k = \frac{1}{n} \sum_m p_k^{(m)}$$

$$\mu_k = \frac{\sum_m p_k^{(m)} y^{(m)}}{\sum_m p_k^{(m)}}$$

$$\Sigma_k = \frac{\sum_m p_k^{(m)} (y - \mu_k)(y - \mu_k)^T}{\sum_m p_k^{(m)}}$$

- 4) Go back to step 2) until some convergence criterion is met

# EM Algorithm: mixture of Gaussians



## Model:

The hidden variable  $X$  has  $k$  possible values, the variable  $Y$  is a point in  $\mathbb{R}^d$

$$P(X = k) := \phi_k$$

$$P(Y = y|X = k) = \mathcal{N}(y; \mu_k, \Sigma_k) := (2\pi)^{-d/2} (\det \Sigma_k)^{-1/2} \exp\left(-\frac{1}{2}(y - \mu_k)^T \Sigma_k^{-1} (y - \mu_k)\right)$$

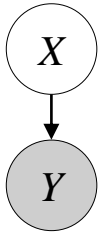
i.e. the condition probabilities are normal distributions

The observations are a set  $D = \{y^{(1)}, \dots, y^{(N)}\}$  of points in  $\mathbb{R}^d$

## Proof (of the M-step):

$$\begin{aligned} & \sum_m \sum_k p_k^{(m)} \log P(Y^{(m)}, X = k | \phi_k, \mu_k, \Sigma_k) \\ &= \sum_m \sum_k p_k^{(m)} \log P(Y^{(m)} | X = k, \mu_k, \Sigma_k) P(X = k | \phi_k) \\ &= \sum_m \sum_k p_k^{(m)} \left( \log \left( 2\pi^{-d/2} (\det \Sigma_k)^{-1/2} \right) + \left( -\frac{1}{2} (y - \mu_k)^T \Sigma_k^{-1} (y - \mu_k) \right) + \log \phi_k \right) \end{aligned}$$

# EM Algorithm: mixture of Gaussians



## Model:

The hidden variable  $X$  has  $k$  possible values, the variable  $Y$  is a point in  $\mathbb{R}^d$

$$P(X = k) := \phi_k$$

$$P(Y = y|X = k) = \mathcal{N}(y; \mu_k, \Sigma_k) := (2\pi)^{-d/2} (\det \Sigma_k)^{-1/2} \exp\left(-\frac{1}{2}(y - \mu_k)^T \Sigma_k^{-1} (y - \mu_k)\right)$$

i.e. the condition probabilities are normal distributions

The observations are a set  $D = \{y^{(1)}, \dots, y^{(N)}\}$  of points in  $\mathbb{R}^d$

## Proof (of the M-step):

$$\begin{aligned} & \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left( \log\left((2\pi)^{-d/2} (\det \Sigma_k)^{-1/2}\right) + \left(-\frac{1}{2}(y^{(m)} - \mu_k)^T \Sigma_k^{-1} (y^{(m)} - \mu_k)\right) + \log \phi_k \right) \\ &= \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left( -\frac{1}{2}(y^{(m)} - \mu_k)^T \Sigma_k^{-1} (y^{(m)} - \mu_k) \right) \\ &= \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left( -\frac{1}{2}(y^{(m)T} \Sigma_k^{-1} y^{(m)} + \mu_k^T \Sigma_k^{-1} \mu_k - 2y^{(m)T} \Sigma_k^{-1} \mu_k) \right) \\ &= \sum_m p_j^{(m)} (x^T \Sigma_j^{-1} - \mu_j^T \Sigma_j^{-1}) \end{aligned}$$

$$\text{By imposing: } \sum_m p_j^{(m)} (x^T \Sigma_j^{-1} - \mu_j^T \Sigma_j^{-1}) = 0 \quad \Rightarrow$$

$$\mu_j = \frac{\sum_m p_j^{(m)} y^{(m)}}{\sum_m p_j^{(m)}}$$

See the link in the web page for the derivations of other parameters ...

# The EM algorithm for learning with missing data

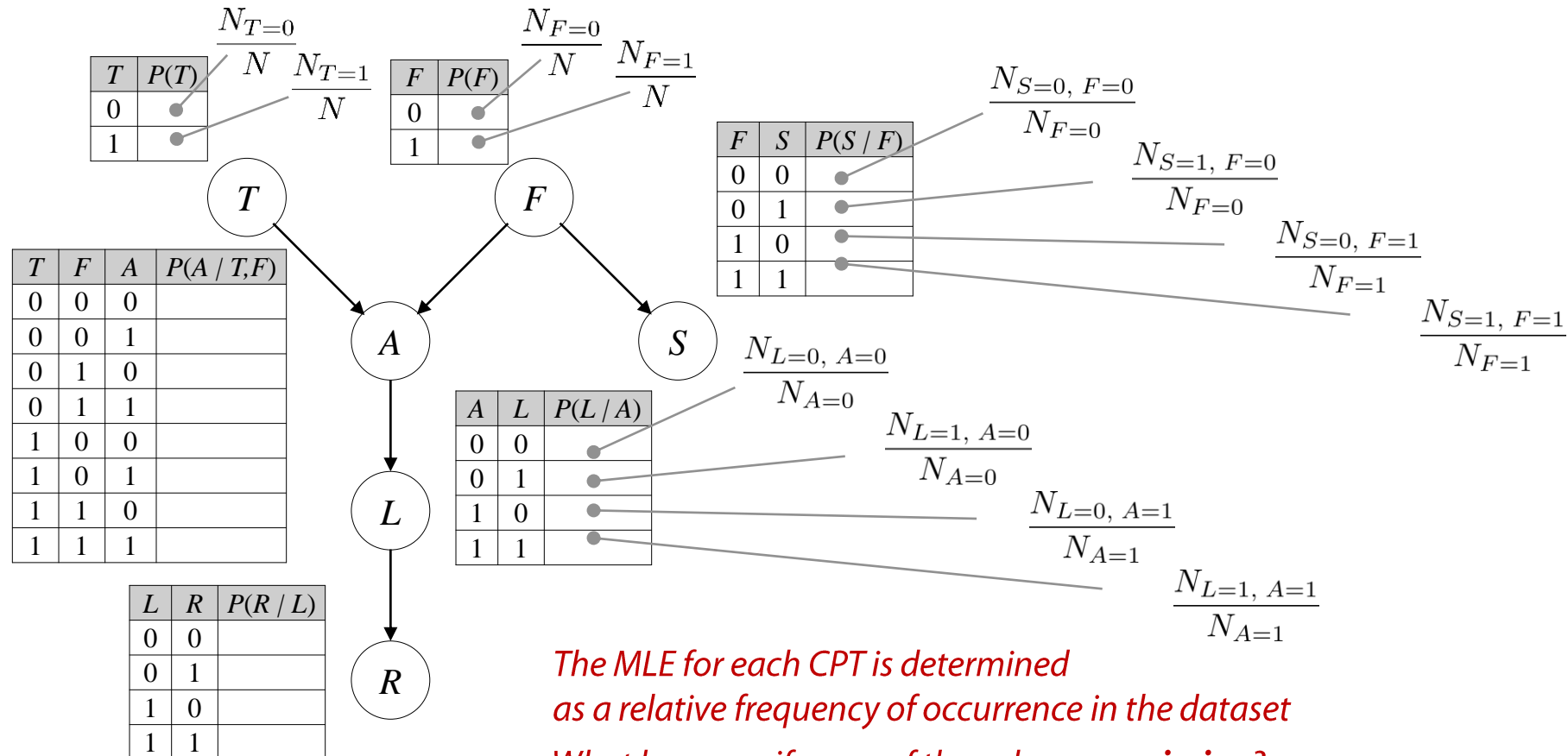


# Learning CPTs for a graphical model via MLE

**Model:** random variables plus the graph of dependencies

**Observations:** dataset of values, from completely observed outcomes

**Parameters (to be determined):** all conditional probabilities (i.e. all CPTs)



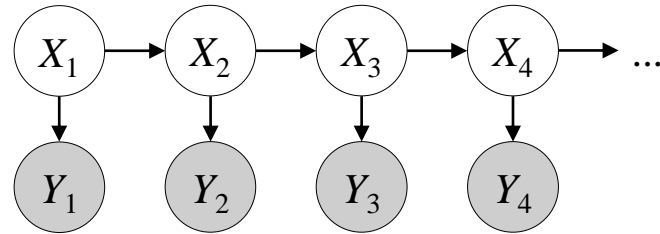
The MLE for each CPT is determined as a relative frequency of occurrence in the dataset  
 What happens if some of the values are **missing**?

# Missing Values

## ■ *Hidden Variables*

Some of the variables may be hidden, i.e., non observable 'by design'

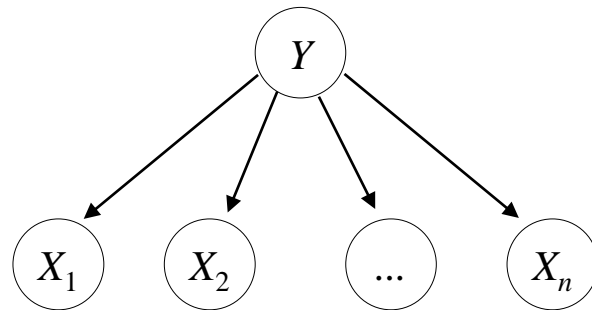
*Example: 'Hidden Markov Model'*



## ■ *Incomplete Observations*

Sometimes, however, observations may be missing 'by accident' and not 'by design'

*Example: 'Naïve Bayesian Classifier'*



What if some classifications  $Y$  are missing, or a few features  $X_i$  are not available?

# Missing Values: *Observability Model*

## ■ *Observed and Unobserved Variables*

Let's consider a graphical model with a set of random variables:

$$\mathbf{X} := \{X_1, \dots, X_n\}$$

In each actual observation (i.e., a data item)

$$\mathbf{X}^{(m)} := \{X_1^{(m)}, \dots, X_n^{(m)}\}$$

each value  $X_i^{(i)}$  may be either *observed* or *unobserved* (i.e., *missing*)  
determined by a *random variable*  $O_{X_i} \in \{0, 1\}$

An ***observability model*** for a graphical model with random variables  $\mathbf{X}$   
is a set of random observability variables

$$\mathbf{O}_{\mathbf{X}} := \{O_{X_1}, \dots, O_{X_n}\}$$

with probability distribution

$$P_{missing}(\mathbf{X}, \mathbf{O}_{\mathbf{X}}) = P(\mathbf{X}) P_{missing}(\mathbf{O}_{\mathbf{X}} | \mathbf{X})$$

# Missing Values: *MCAR*

- *Missing Completely at Random (MCAR)*

Fundamental assumption

$$\langle \mathbf{X} \perp \mathbf{O}_{\mathbf{X}} \rangle$$

which entails that:

$$P_{missing}(\mathbf{X}, \mathbf{O}_{\mathbf{X}}) = P(\mathbf{X}) P_{missing}(\mathbf{O}_{\mathbf{X}})$$

This is tempting and it could ease all subsequent computations...

*... but it is too strong, and hardly enforceable in many practical cases*

Moral: we need a weaker assumption

# Missing Values: *MAR*

- *Missing at Random (MAR)*

Consider a generic data item, with possibly missing values

$$\mathbf{X}^{(m)} := \underset{\substack{\text{observed} \\ \swarrow}}{\mathbf{X}_{obs}^{(m)}} \cup \underset{\substack{\text{missing} \\ \searrow}}{\mathbf{X}_{hid}^{(m)}}$$

*Missing values  
need NOT be for the same  
variables in each data item*

Fundamental assumption: in each data item

$$\langle \mathbf{X}_{hid}^{(m)} \perp \mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)} \rangle$$

Namely, the *values* of the missing variables are independent from their *observability* given the *values* of the observed variables

*It is still a strong assumption, yet much more realistic...*

# Missing Values: *MAR*

- *Missing at Random (MAR)*

Fundamental assumption: in each data item

$$\langle \mathbf{X}_{hid}^{(m)} \perp \mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)} \rangle$$

This entails:

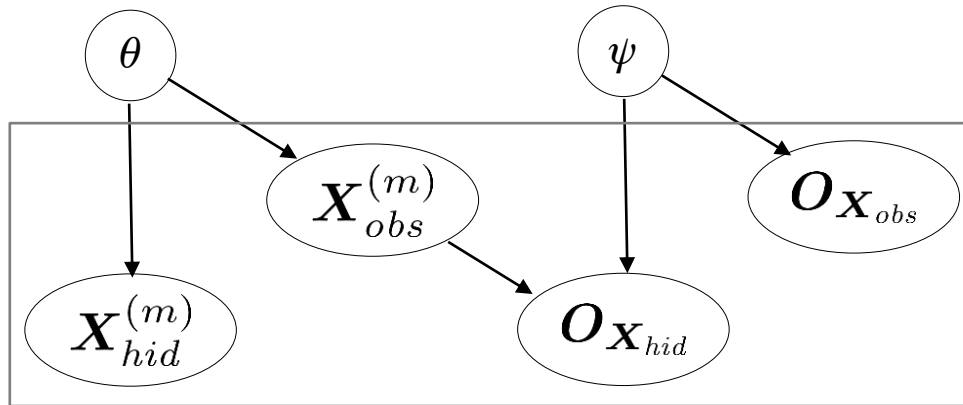
$$\begin{aligned} P_{missing}(\mathbf{X}^{(m)}, \mathbf{O}_X) &= P_{missing}(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}, \mathbf{O}_X) \\ &= P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}) P_{missing}(\mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}) \\ &= P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}) P_{missing}(\mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)}) \end{aligned}$$

$$P_{missing}(\mathbf{X}_{obs}^{(m)}, \mathbf{O}_X) = \sum_{\mathbf{X}_{hid}} P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}) P_{missing}(\mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)})$$

$$P_{missing}(\mathbf{X}_{obs}^{(m)}, \mathbf{O}_X) = P(\mathbf{X}_{obs}^{(m)}) P_{missing}(\mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)})$$

————— This is the relevant property

# Likelihood under MAR



$$\langle \mathbf{X}_{hid}^{(m)} \perp \mathbf{O}_{\mathbf{X}} \mid \mathbf{X}_{obs}^{(m)} \rangle$$

Variables and parameters in an observability model as a graphical model

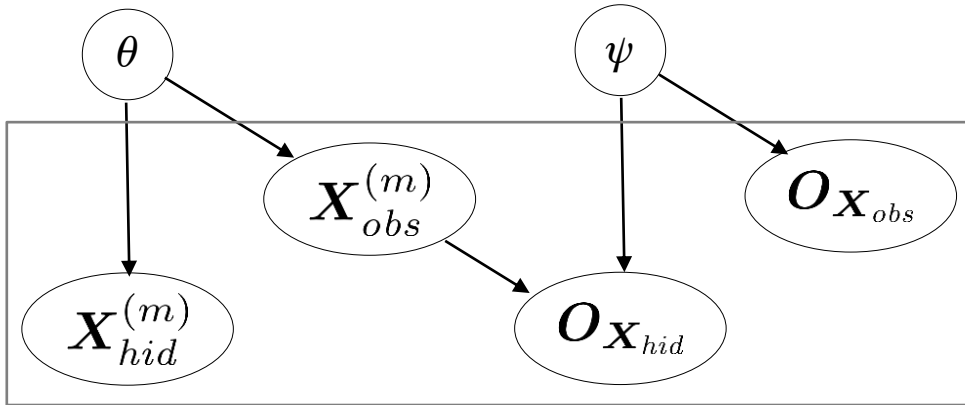
## Likelihood

$$L(\theta, \psi \mid D) = \prod_{m=1}^N P(D^{(m)} \mid \theta, \psi) \quad \text{where:} \quad D^{(m)} := \{\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}\}$$

$$\begin{aligned} l(\theta, \psi \mid D) &= \sum_{m=1}^N \log P(D^{(m)} \mid \theta, \psi) \\ &= \sum_{m=1}^N \log \left( P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)} \mid \theta, \psi) P_{missing}(\mathbf{O}_{\mathbf{X}} \mid \mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)}, \theta, \psi) \right) \\ &= \sum_{m=1}^N \log \left( P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)} \mid \theta) P_{missing}(\mathbf{O}_{\mathbf{X}} \mid \mathbf{X}_{obs}^{(m)}, \psi) \right) \\ &= \sum_{m=1}^N \log P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)} \mid \theta) + \sum_{m=1}^N \log P_{missing}(\mathbf{O}_{\mathbf{X}} \mid \mathbf{X}_{obs}^{(m)}, \psi) \end{aligned}$$

We are interested in optimizing  $\theta$  ... yet we have only observed values

# Likelihood under MAR



$$\langle \mathbf{X}_{hid}^{(m)} \perp \mathbf{O}_X \mid \mathbf{X}_{obs}^{(m)} \rangle$$

Variables and parameters in an observability model as a graphical model

Likelihood (for observed values)

$$\begin{aligned} l(\theta \mid D) &:= \sum_{m=1}^N \log P(\mathbf{X}_{obs}^{(m)} \mid \theta) \\ &= \sum_{m=1}^N \log \sum_{\mathbf{X}_{hid}^{(m)}} P(\mathbf{X}_{obs}^{(m)}, \mathbf{X}_{hid}^{(m)} \mid \theta) \\ &= \sum_{m=1}^N \log \sum_{\mathbf{X}_{hid}^{(m)}} \left( P(\mathbf{X}_{obs}^{(m)} \mid \theta) P(\mathbf{X}_{hid}^{(m)} \mid \mathbf{X}_{obs}^{(m)}, \theta) \right) \end{aligned}$$

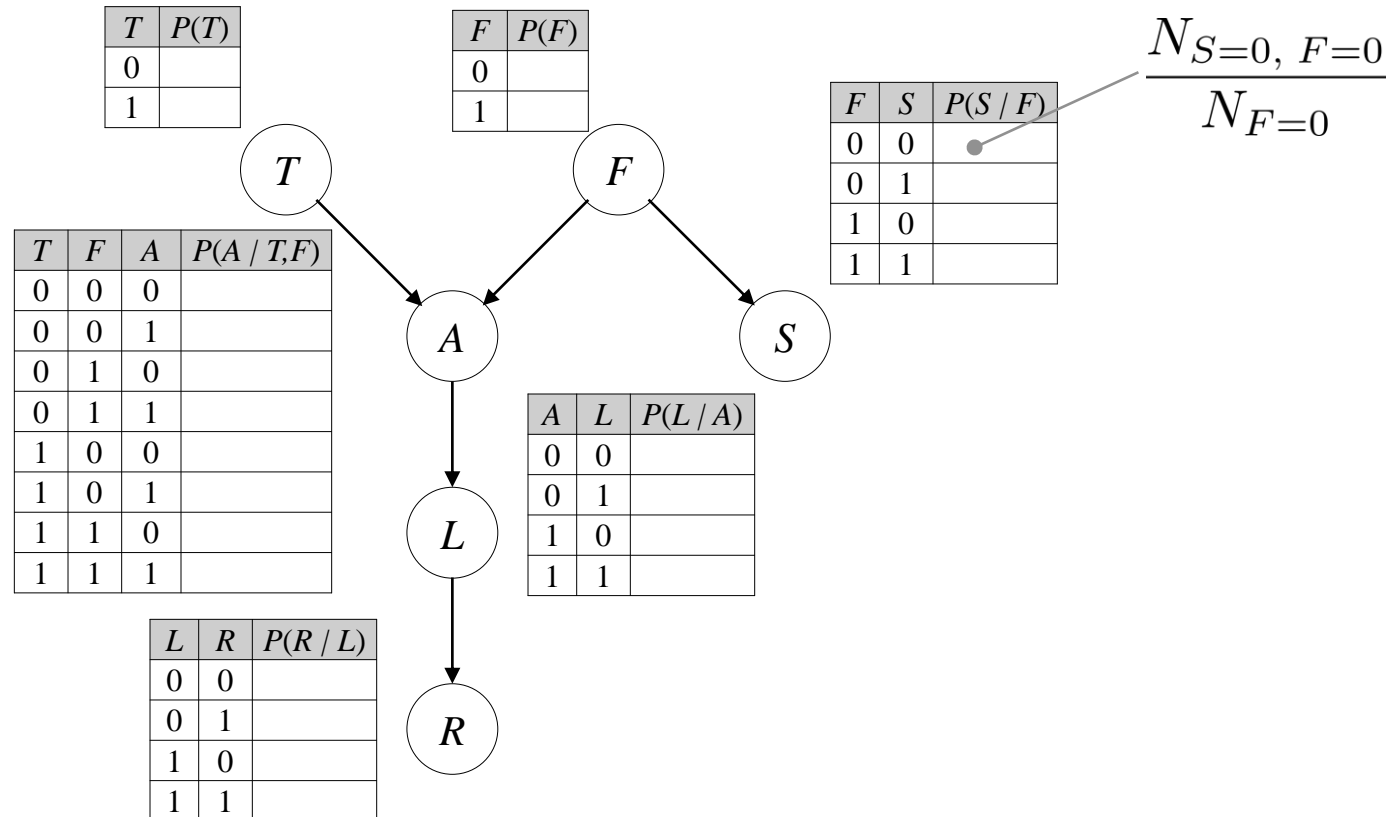
*Looks promising: using probabilities instead of missing values ...  
but this may be very hard to optimize in general*



# Learning CPTs for a graphical model via EM

**Fundamental idea:** using probabilities of observations

In the completely observed case: probabilities are estimated as frequencies of occurrence



# Learning CPTs for a graphical model via EM

**Fundamental idea:** using probabilities of observations

In the completely observed case: probabilities are estimated as frequencies of occurrence

More in general:

$$\frac{N_{X_i, \mathbf{Z}}}{N_{\mathbf{Z}}} \quad \text{where:} \quad \mathbf{Z} = \text{parents}(X_i)$$

In the EM algorithm, use *estimated* occurrences:

$$\frac{\tilde{N}_{X_i, \mathbf{Z}}}{\tilde{N}_{\mathbf{Z}}} \quad \text{where:} \quad \tilde{N}_{\mathbf{X}} := \sum_{m=1}^N P(\mathbf{X} \mid \mathbf{X}_{obs}^{(m)}, \theta)$$

*In words, any incomplete observations 'splits up' and contributes with the probabilities of the (possible) missing values*

*Note that, when all observations are complete:*

$$\tilde{N}_{\mathbf{X}} = N_{\mathbf{X}}$$

# Learning CPTs for a graphical model via EM

**Fundamental idea:** using probabilities of observations

## Algorithm:

- 1) Assign parameters  $\theta^{(0)}$  at random
- 2) Compute  $P(\mathbf{X} \mid \theta^{(t)})$
- 3) Update all parameters using estimated *occurrences*:

$$\theta_{X_i \mid \mathbf{Z}}^{(t+1)} = \frac{\tilde{N}_{X_i, \mathbf{Z}}}{\tilde{N}_{\mathbf{Z}}} \quad \text{where all estimations are made using } P(\mathbf{X} \mid \theta^{(t)})$$

- 3) Go back to step 2) until some convergence criterion is met