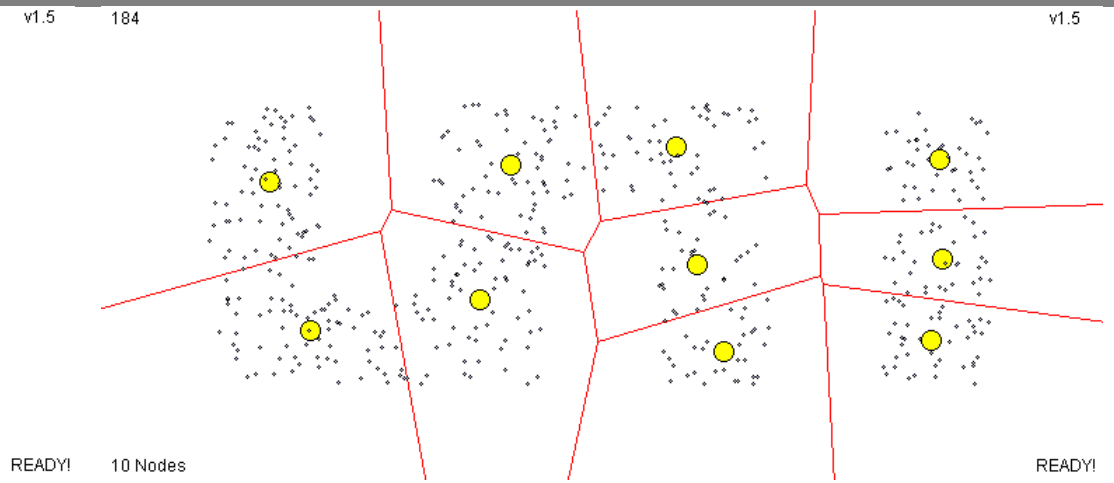
Unsupervised Learning

Marco Piastra

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

Vector quantization

Original data



Quantization (compression via prototypes)

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

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:

$$w(\boldsymbol{x}) := \text{argmin}_{\boldsymbol{w}_j} \|\boldsymbol{x} - \boldsymbol{w}_j\|$$

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

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

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

is minimized.

***k-means* algorithm:**

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

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

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

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

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

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

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

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

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

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

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

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

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

$$= \frac{\partial}{\partial \boldsymbol{w}_j}\frac{1}{2}\sum_{D(\boldsymbol{w}_j)} (\boldsymbol{x}_i^2 + \boldsymbol{w}_j^2 - 2\,\boldsymbol{x}_i^T\,\boldsymbol{w}_j) = \sum_{D(\boldsymbol{w}_j)} (\boldsymbol{w}_j - \boldsymbol{x}_i)$$

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

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

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*)

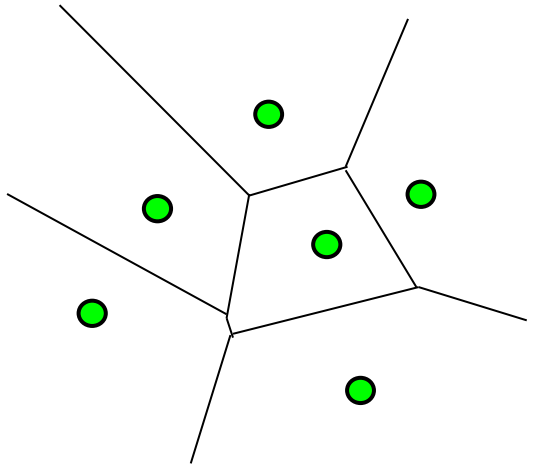
k-means (*Generalized Lloyd's Algorithm – Vector quantization*)

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

Voronoi cell:

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

Voronoi tesselation: the complex of all Voronoi cells of θ



Algorithm (*rewritten*)**:**

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

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

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

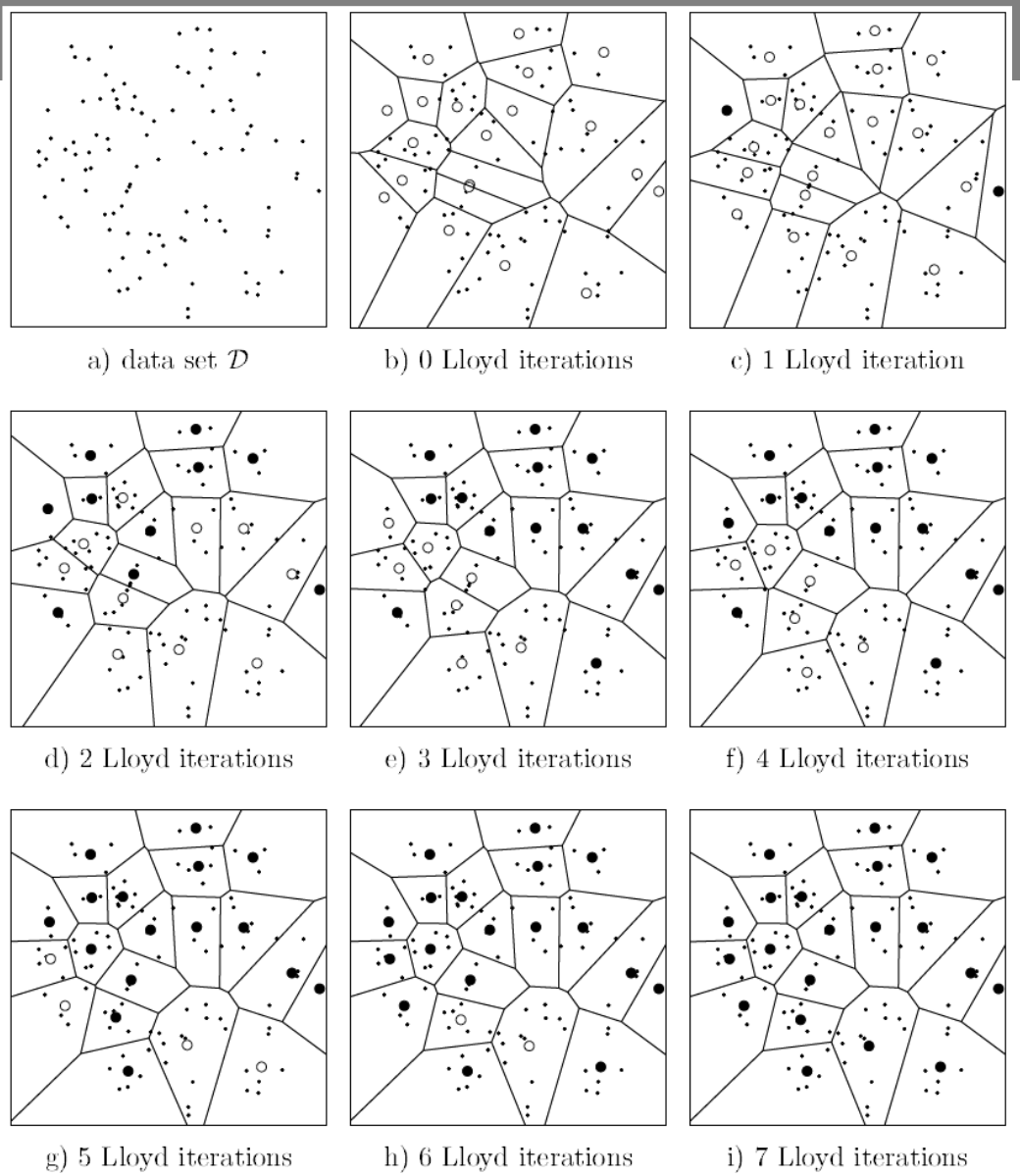
$$\boldsymbol{w}_j = \frac{1}{|\{\boldsymbol{x}_i \in V(\boldsymbol{w}_j)\}|} \sum_{\{\boldsymbol{x}_i \in V(\boldsymbol{w}_j)\}} \boldsymbol{x}_i$$

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

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

Continuous case

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

A linear operator

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

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

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 $\{X_i\}$ is

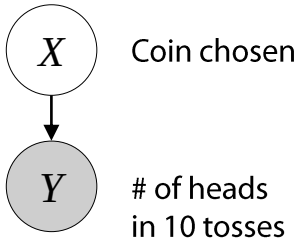
$$\mathbb{E}[f(\{X_i\})] := \sum_{\{X_i\}} f(\{X_i\})\, P(\{X_i\})$$

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

Expectation Maximization: a preliminary example



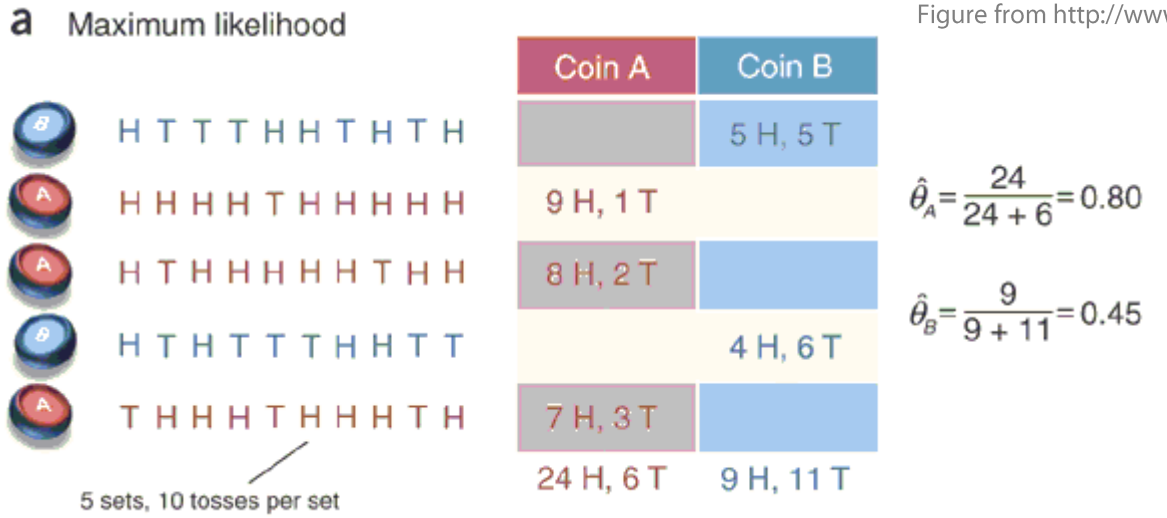


Figure from http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html

- An experiment with two coins

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

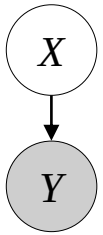
  Random variables: Y *number of heads,* X *selected coin (i.e A or B)*

  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:

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

Expectation Maximization: a preliminary example



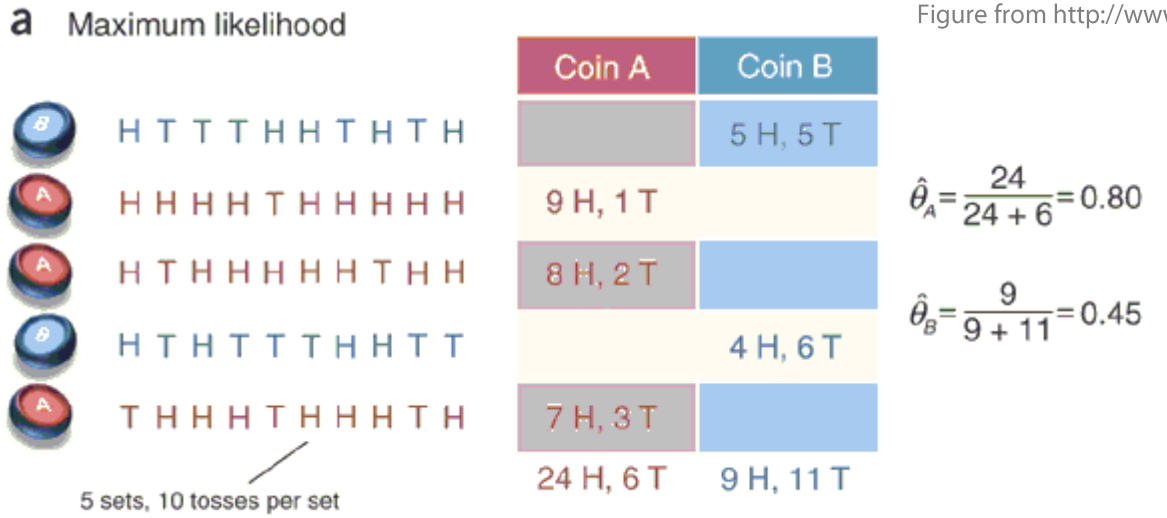


Figure from http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html

- An experiment with two coins

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

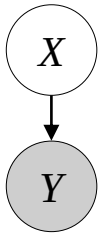
  Random variables: Y *number of heads,* X *selected coin (i.e A or B)*

  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



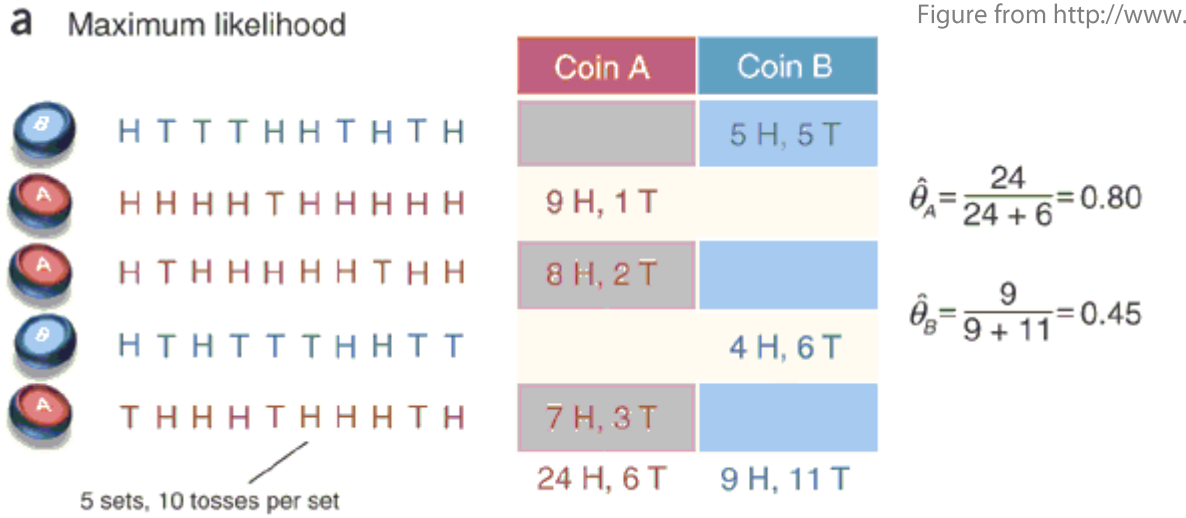


Figure from http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html

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

  Likelihood

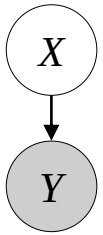
$$P(D \mid \theta) = P(\{Y^{(i)}\} \mid \theta) = \sum_{\{X^{(i)}\}} P(\{(Y^{(i)}, X^{(i)})\} \mid \theta)$$

  MLE

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

  ** This optimization is intractable, in general*

Expectation Maximization: a preliminary example



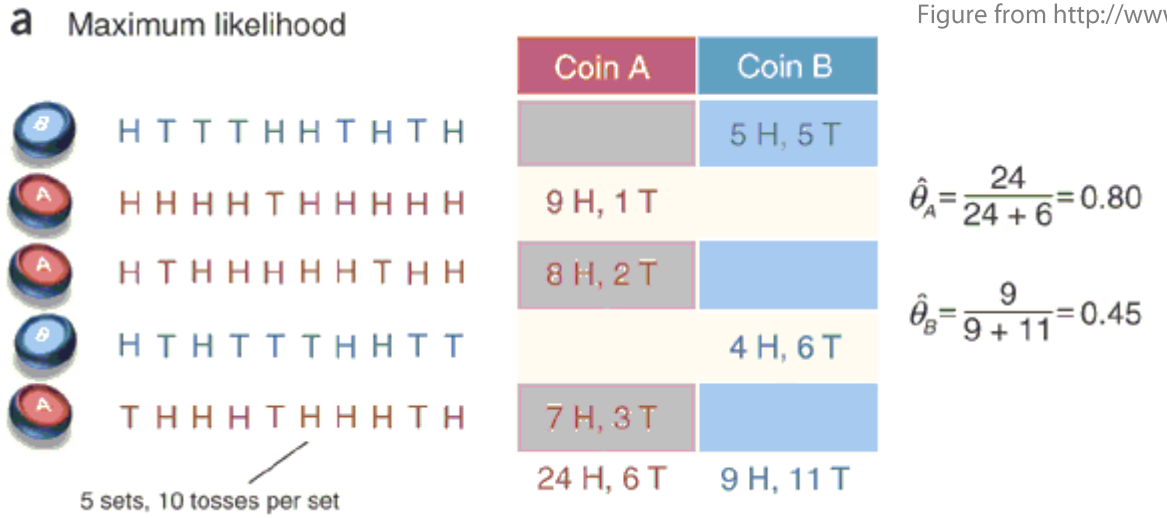


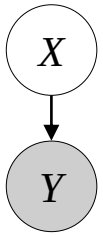
Figure from http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html

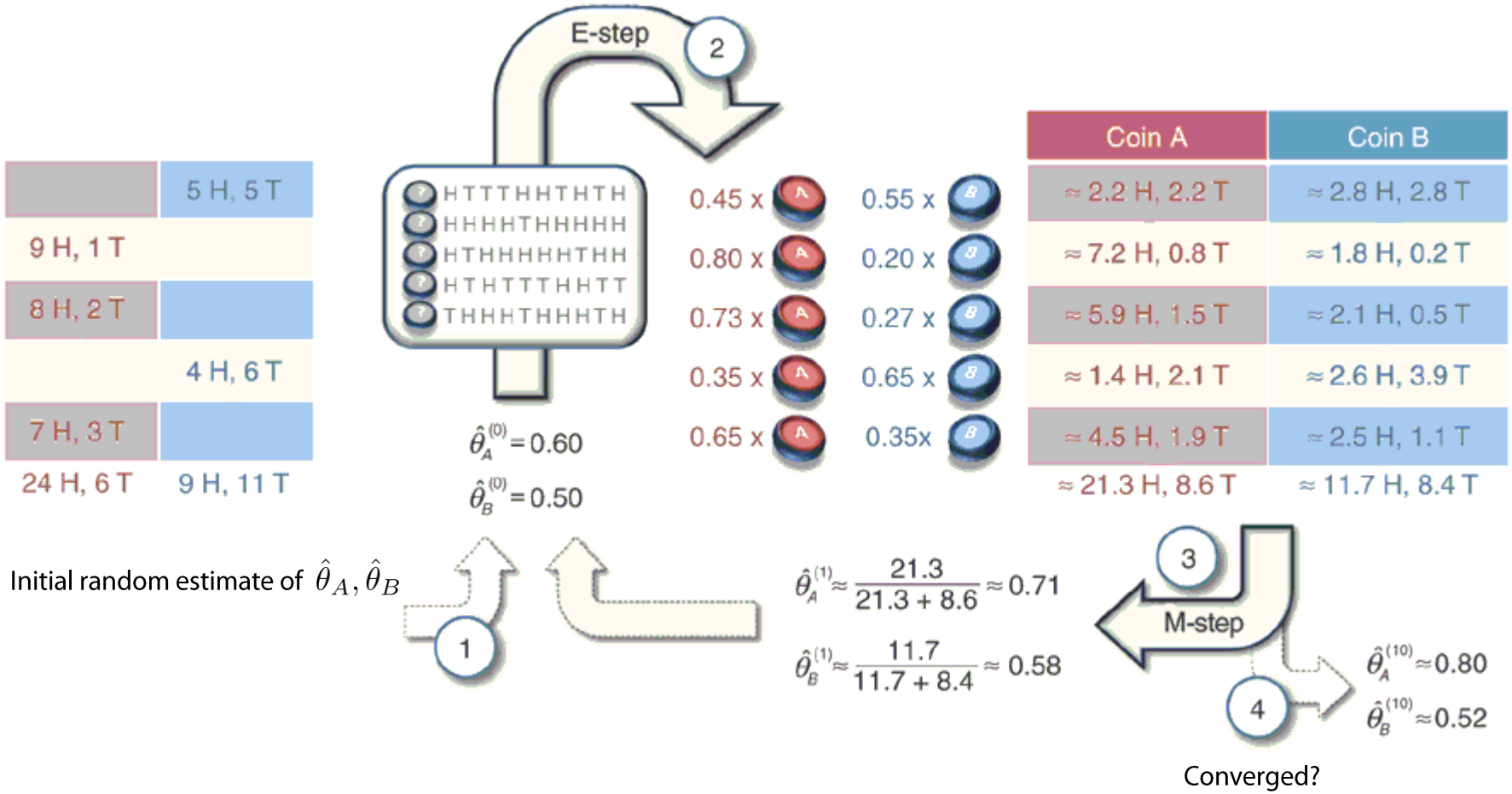
- What if X is *hidden* (= *latent*, = *unobserved*)?

  Intuitive idea: use expected values for unobserved variables

  1. Define an initial (random) guess $\hat{\theta}^{(0)}$
  2. Compute $Q_i(X^{(i)}) := P(X^{(i)} \mid Y^{(i)}; \hat{\theta}^{(t)})$ *E-step*
  3. Maximize
     $$\hat{\theta}^{(t+1)} = \text{argmax}_\theta \sum_i \mathbb{E}_{Q_i(X^{(i)})}[Y^{(i)} \mid X^{(i)}; \hat{\theta}^{(t)}]$$ *M-step*
  4. Unless some convergence criterion has been met, go to step 2.

Expectation Maximization: a preliminary example



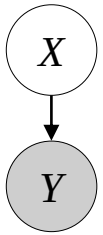


Initial random estimate of $\hat{\theta}_A, \hat{\theta}_B$

Converged?

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

Expectation Maximization: a preliminary example



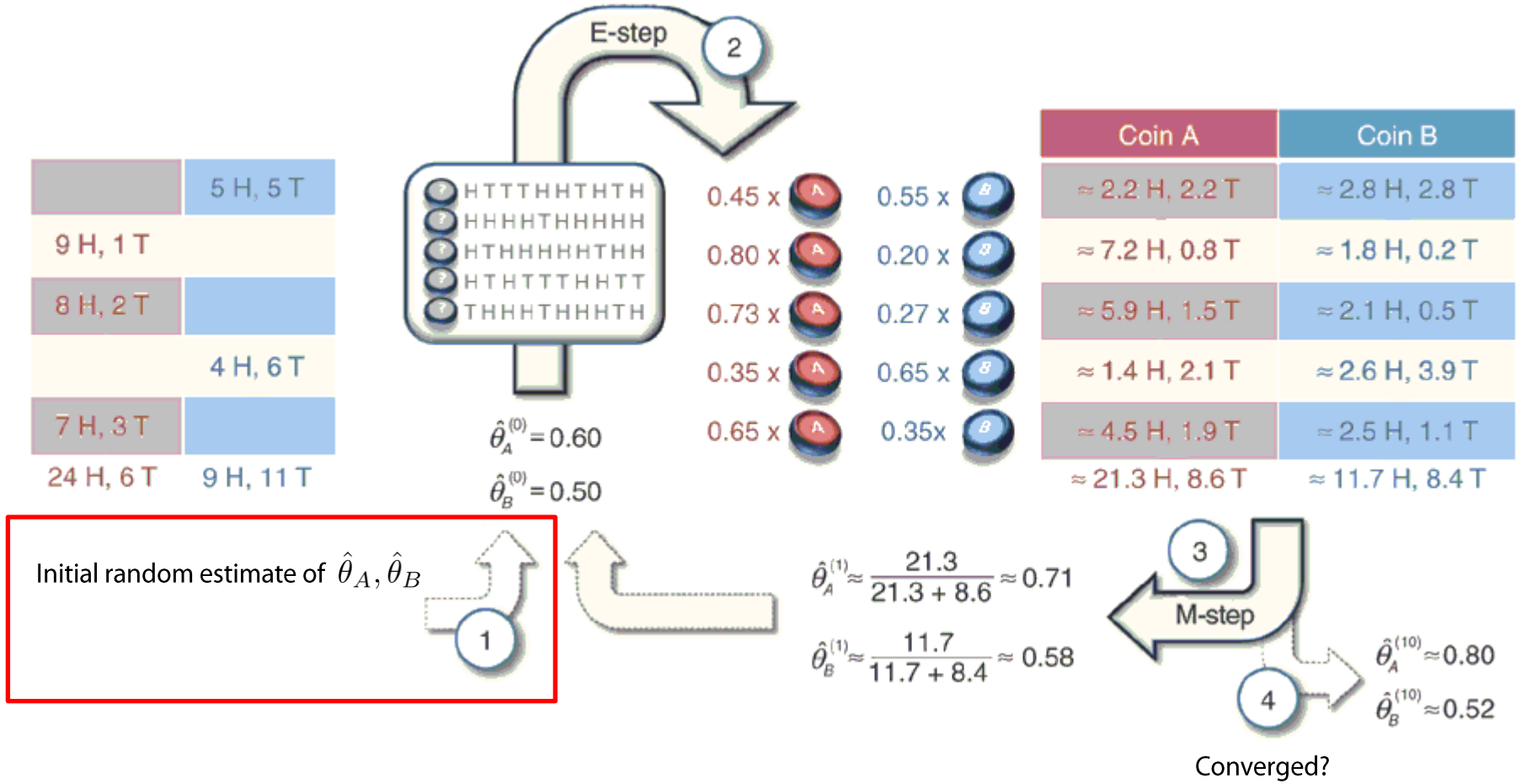
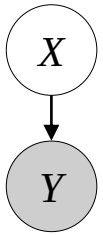


Figure from http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html

Expectation Maximization: a preliminary example

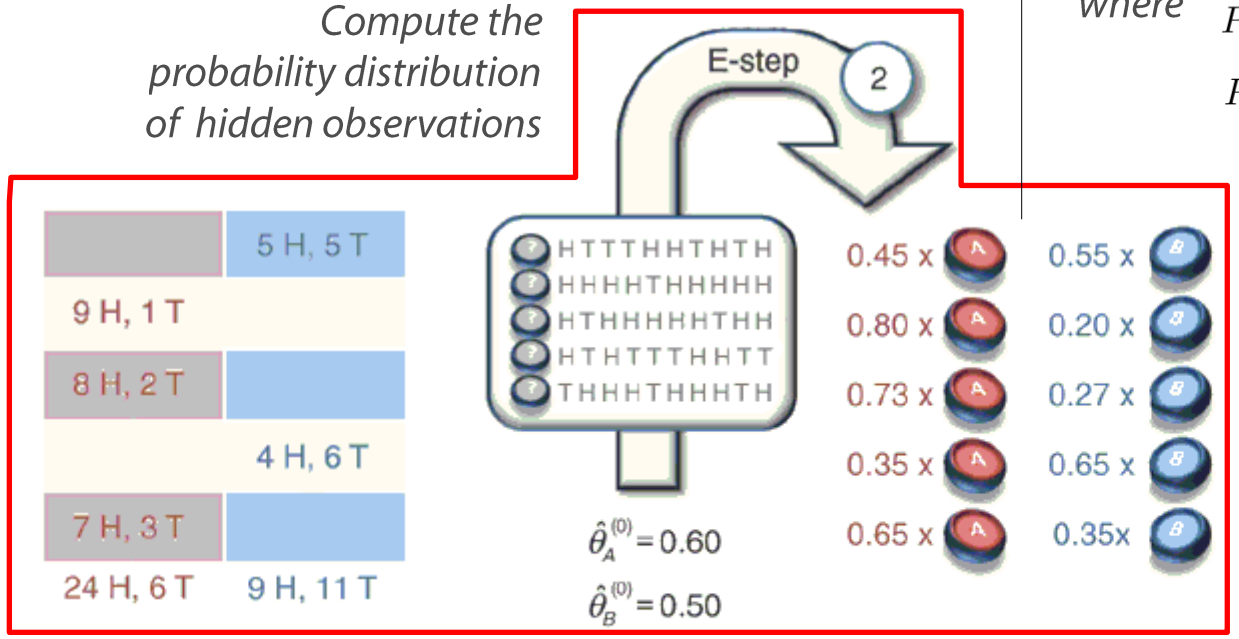


$$Q_i(X^{(i)}) := P(X^{(i)} \mid Y^{(i)}; \hat{\theta}^{(t)})$$

Compute the probability distribution of hidden observations

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

$P(X = x \mid \theta) = 0.5$



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 $\hat{\theta}_A, \hat{\theta}_B$

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

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

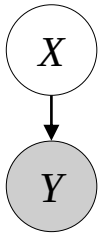
$\hat{\theta}_A^{(10)} \approx 0.80$

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

Converged?

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

Expectation Maximization: a preliminary example

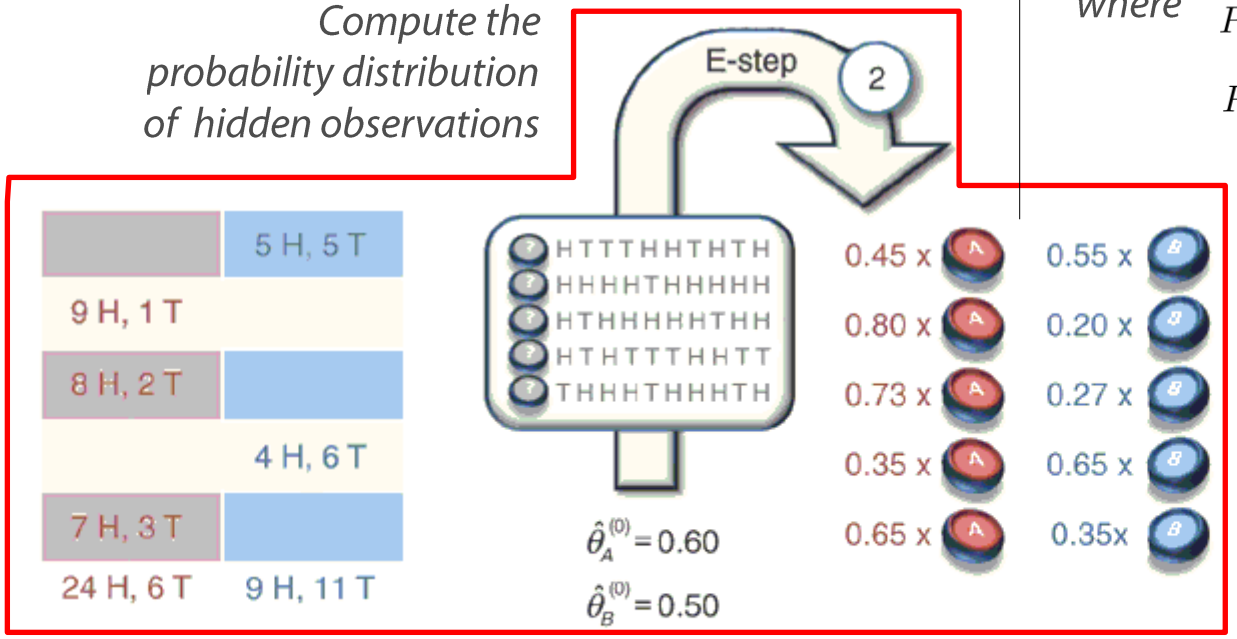


$$Q_i(X^{(i)} = x) = \frac{P(Y^{(i)} = y \mid X^{(i)} = x; \theta) P(X^{(i)} = x \mid \theta)}{\sum_x P(Y^{(i)} = y \mid X^{(i)} = x; \theta) P(X^{(i)} = x \mid \theta)}$$

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

$P(X^{(i)} = x \mid \theta) = 0.5$

Compute the probability distribution of hidden observations

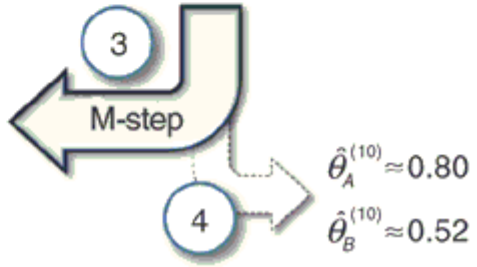


Coin A	Coin B
≈ 2.2 H, 2.2 T	≈ 2.8 H, 2.8 T
≈ 7.2 H, 0.8 T	≈ 1.8 H, 0.2 T
≈ 5.9 H, 1.5 T	≈ 2.1 H, 0.5 T
≈ 1.4 H, 2.1 T	≈ 2.6 H, 3.9 T
≈ 4.5 H, 1.9 T	≈ 2.5 H, 1.1 T
≈ 21.3 H, 8.6 T	≈ 11.7 H, 8.4 T

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

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

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



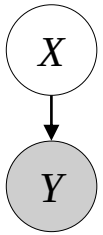
$\hat{\theta}_A^{(10)} \approx 0.80$

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

Converged?

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

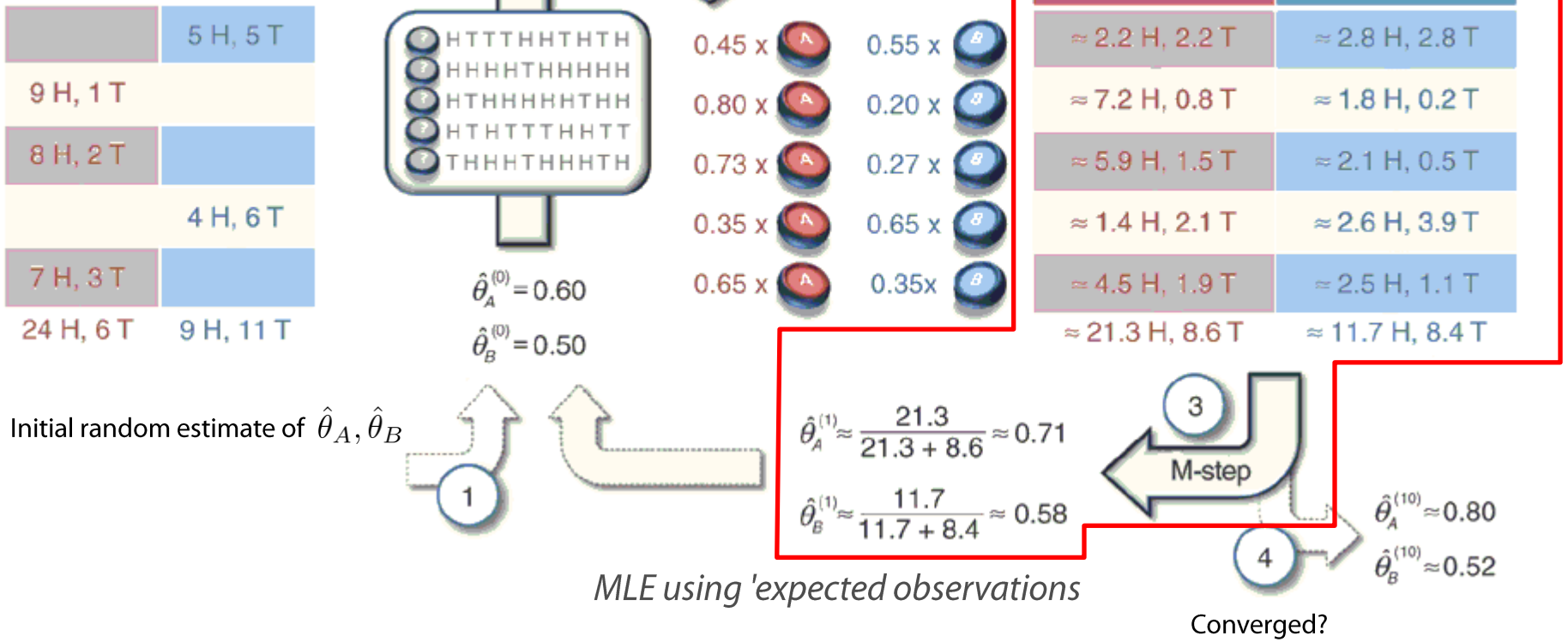
Expectation Maximization: a preliminary example



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

$$= \sum_x y\, P(X^{(i)} = x \mid Y^{(i)} = y; \theta)$$

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



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

MLE using 'expected observations

Converged?

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

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*

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

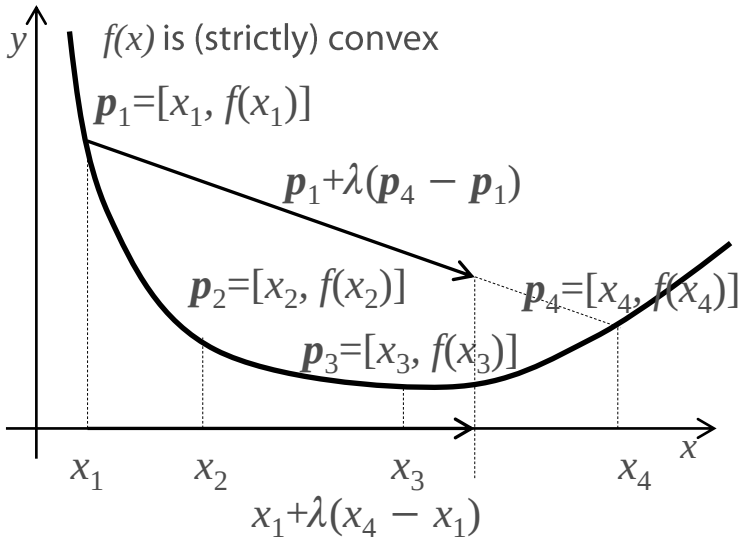
*f is **convex** when for any two points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ the segment $(\boldsymbol{p}_i - \boldsymbol{p}_j)$ is not below f*

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*

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



Corollary:

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

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

Dual results also hold for <u>*concave*</u> functions

An aside: Jensen's inequality

A relationship between probability and geometry

When f is *convex function*

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

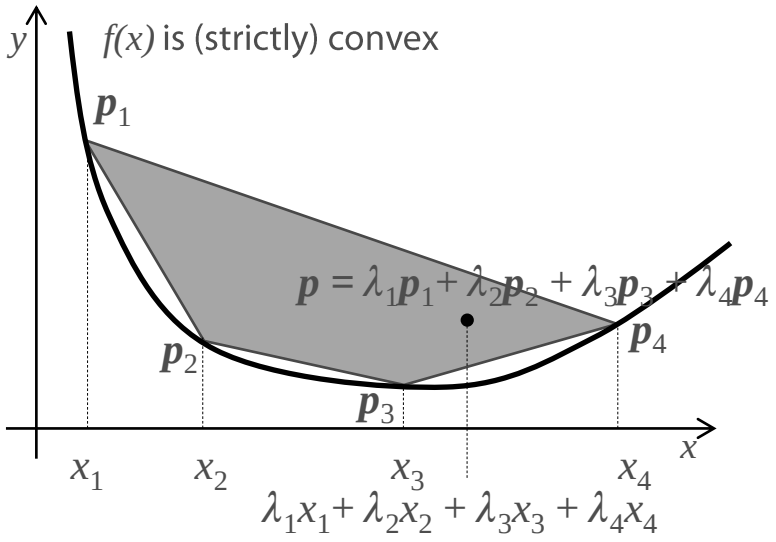
To see this, consider

$$\boldsymbol{p} = \lambda_1 \boldsymbol{p}_1 + \lambda_2 \boldsymbol{p}_2 + \lambda_3 \boldsymbol{p}_3 + \lambda_4 \boldsymbol{p}_4$$

i.e. a ***linear combination*** of $\boldsymbol{p}_i$ points

This is an ***affine*** combination if $\sum \lambda_i = 1$

and it is a ***convex*** combination if also $\lambda_i \geq 0, \ \forall i$



When the λ_i define a probability, then $\boldsymbol{p}$ is a *convex combination* of $\boldsymbol{p}_i$ points

Any convex combination of $\boldsymbol{p}_i$ points lies inside their ***convex hull*** (*see figure*) and therefore above f :

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

Corollary: the only way to make the convex hull be <u>on</u> 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

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

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

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

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

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

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

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

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

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

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 θ at random
2) (*E-step*) Compute the probabilities

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

3) (*M-step*) Compute a new estimate of θ

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

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

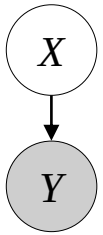
The algorithm converges to a local maximum of the log-likelihood

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 $D = \{y^{(1)}, \ldots, y^{(N)}\}$ *of points in* $\mathbb{R}^d$

Algorithm:

1) For each value *k*, assign ϕ_k, μ_k and Σ_k at random
2) (*E-step*) For all the $y^{(m)}$ in D compute the probabilities

$$p_k^{(m)} = P(X = k|y^{(m)}, \phi_k, \mu_k, \Sigma_k) = \phi_k \cdot \mathcal{N}(x^{(m)}; \mu_k, \Sigma_k)$$

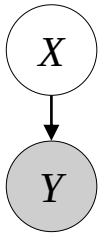
3) (*M-step*) Compute the new estimates for the parameters

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

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

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 $D = \{y^{(1)}, \ldots, y^{(N)}\}$ *of points in* $\mathbb{R}^d$

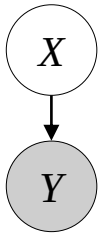
Proof (of the M-step):

$$\sum_m \sum_k p_k^{(m)} \log P(Y^{(m)}, X = k|\phi_k, \mu_k, \Sigma_k)$$

$$= \sum_m \sum_k p_k^{(m)} \log P(Y^{(m)}|X = k, \mu_k, \Sigma_k) P(X = k|\phi_k)$$

$$= \sum_m \sum_k p_k^{(m)} \left(\log\left(2\pi^{-d/2}(\det \Sigma_k)^{-1/2}\right) + \left(-\frac{1}{2}(y - \mu_k)^T \Sigma_k^{-1}(y - \mu_k)\right) + \log \varphi_k\right)$$

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 $D = \{y^{(1)}, \ldots, y^{(N)}\}$ *of points in* $\mathbb{R}^d$

Proof (of the M-step):

$$\frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left(\log\left((2\pi)^{-d/2}(\det \Sigma_k)^{-1/2}\right) + \left(-\frac{1}{2}(y^{(m)} - \mu_k)^T \Sigma_k^{-1}(y^{(m)} - \mu_k)\right) + \log \phi_k\right)$$

$$= \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left(-\frac{1}{2}(y^{(m)} - \mu_k)^T \Sigma_k^{-1}(y^{(m)} - \mu_k)\right)$$

$$= \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left(-\frac{1}{2}(y^{(m)T} \Sigma_k^{-1} y^{(m)} + \mu_k^T \Sigma_k^{-1} \mu_k - 2y^{(m)T} \Sigma_k^{-1} \mu_k)\right)$$

$$= \sum_m p_j^{(m)} \left(x^T \Sigma_j^{-1} - \mu_j^T \Sigma_j^{-1}\right)$$

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

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

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

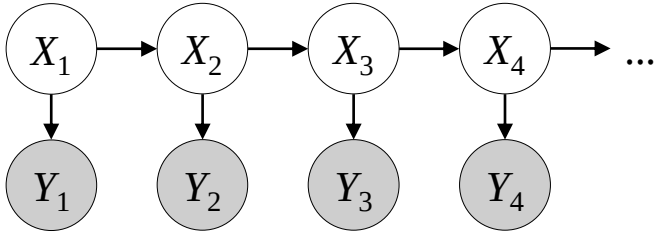
The EM algorithm for learning with missing data

Missing Values

- *Hidden Variables*

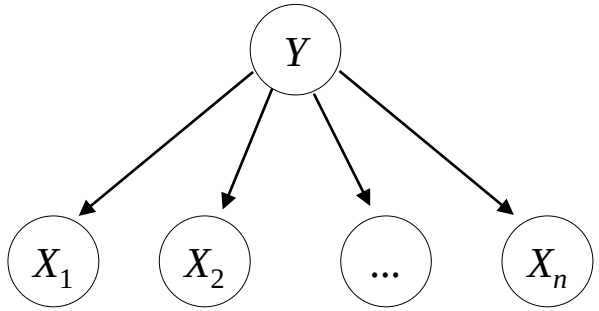
  Some of the variables may be hidden, i.e., non observable 'by design'

  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:

  $$\boldsymbol{X} := \{X_1, \ldots, X_n\}$$

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

  $$\boldsymbol{X}^{(m)} := \{X_1^{(m)}, \ldots, X_n^{(m)}\}$$

  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 $\boldsymbol{X}$ is a set of random observability variables

  $$\boldsymbol{O_X} := \{O_{X_1}, \ldots, O_{X_n}\}$$

  with probability distribution

  $$P_{missing}(\boldsymbol{X}, \boldsymbol{O_X}) = P(\boldsymbol{X})\, P_{missing}(\boldsymbol{O_X} | \boldsymbol{X})$$

Missing Values: *MCAR*

- *Missing Completely at Random (MCAR)*

  Fundamental assumption

  $$\langle \boldsymbol{X} \perp \boldsymbol{O_X} \rangle$$

  which entails that:

  $$P_{missing}(\boldsymbol{X}, \boldsymbol{O_X}) = P(\boldsymbol{X})\, P_{missing}(\boldsymbol{O_X})$$

  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

$$\boldsymbol{X}^{(m)} := \boldsymbol{X}^{(m)}_{obs} \cup \boldsymbol{X}^{(m)}_{hid}$$

observed / *missing*

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

  Fundamental assumption: in each data item

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

  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

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

  This entails:

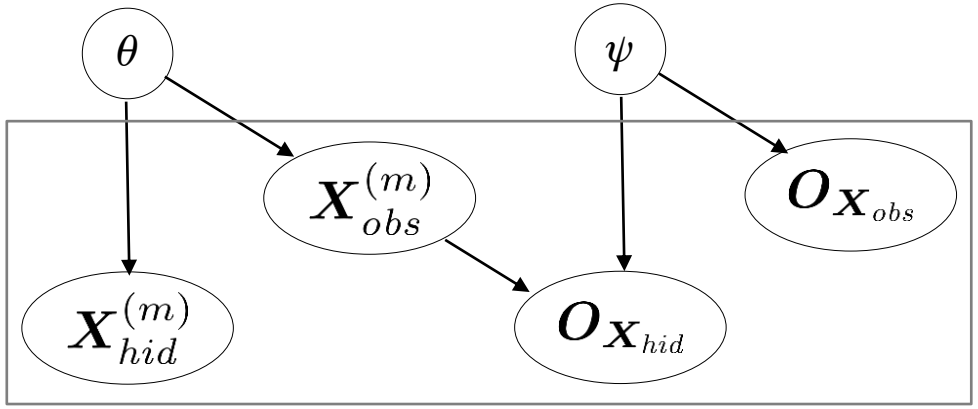
$$P_{missing}(\boldsymbol{X}^{(m)}, \boldsymbol{O}_{\boldsymbol{X}}) = P_{missing}(\boldsymbol{X}_{obs}^{(m)}, \boldsymbol{X}_{hid}^{(m)}, \boldsymbol{O}_{\boldsymbol{X}})$$
$$= P(\boldsymbol{X}_{obs}^{(m)}, \boldsymbol{X}_{hid}^{(m)})\, P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}_{obs}^{(m)}, \boldsymbol{X}_{hid}^{(m)})$$
$$= P(\boldsymbol{X}_{obs}^{(m)}, \boldsymbol{X}_{hid}^{(m)})\, P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}_{obs}^{(m)})$$

$$P_{missing}(\boldsymbol{X}_{obs}^{(m)}, \boldsymbol{O}_{\boldsymbol{X}}) = \sum_{\boldsymbol{X}_{hid}} P(\boldsymbol{X}_{obs}^{(m)}, \boldsymbol{X}_{hid}^{(m)})\, P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}_{obs}^{(m)})$$

$$P_{missing}(\boldsymbol{X}_{obs}^{(m)}, \boldsymbol{O}_{\boldsymbol{X}}) = P(\boldsymbol{X}_{obs}^{(m)})\, P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}_{obs}^{(m)})$$

This is the relevant property

Likelihood under MAR



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

Variables and parameters in an observability model as a graphical model

Likelihood

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

$$l(\theta, \psi \mid D) = \sum_{m=1}^{N} \log P(D^{(m)} \mid \theta, \psi)$$

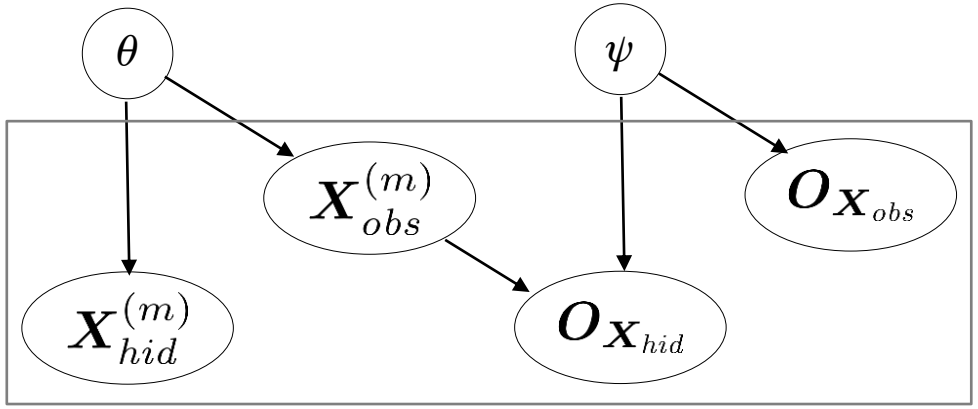
$$= \sum_{m=1}^{N} \log \left(P(\boldsymbol{X}^{(m)}_{obs}, \boldsymbol{X}^{(m)}_{hid} \mid \theta, \psi)\, P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}^{(m)}_{obs}, \boldsymbol{X}^{(m)}_{hid}, \theta, \psi) \right)$$

$$= \sum_{m=1}^{N} \log \left(P(\boldsymbol{X}^{(m)}_{obs}, \boldsymbol{X}^{(m)}_{hid} \mid \theta)\, P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}^{(m)}_{obs}, \psi) \right)$$

$$= \sum_{m=1}^{N} \log P(\boldsymbol{X}^{(m)}_{obs}, \boldsymbol{X}^{(m)}_{hid} \mid \theta) + \sum_{m=1}^{N} \log P_{missing}(\boldsymbol{O}_{\boldsymbol{X}} | \boldsymbol{X}^{(m)}_{obs}, \psi)$$

We are interested in optimizing θ ... *yet we have only observed values*

Likelihood under MAR



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

Variables and parameters in an observability model as a graphical model

Likelihood (for observed values)

$$l(\theta \mid D) := \sum_{m=1}^{N} \log P(\boldsymbol{X}^{(m)}_{obs} \mid \theta)$$

$$= \sum_{m=1}^{N} \log \sum_{\boldsymbol{X}^{(m)}_{hid}} P(\boldsymbol{X}^{(m)}_{obs}, \boldsymbol{X}^{(m)}_{hid} \mid \theta)$$

$$= \sum_{m=1}^{N} \log \sum_{\boldsymbol{X}^{(m)}_{hid}} \left(P(\boldsymbol{X}^{(m)}_{obs} \mid \theta) P(\boldsymbol{X}^{(m)}_{hid} \mid \boldsymbol{X}^{(m)}_{obs}, \theta) \right)$$

Looks promising: using probabilities instead of missing values ...

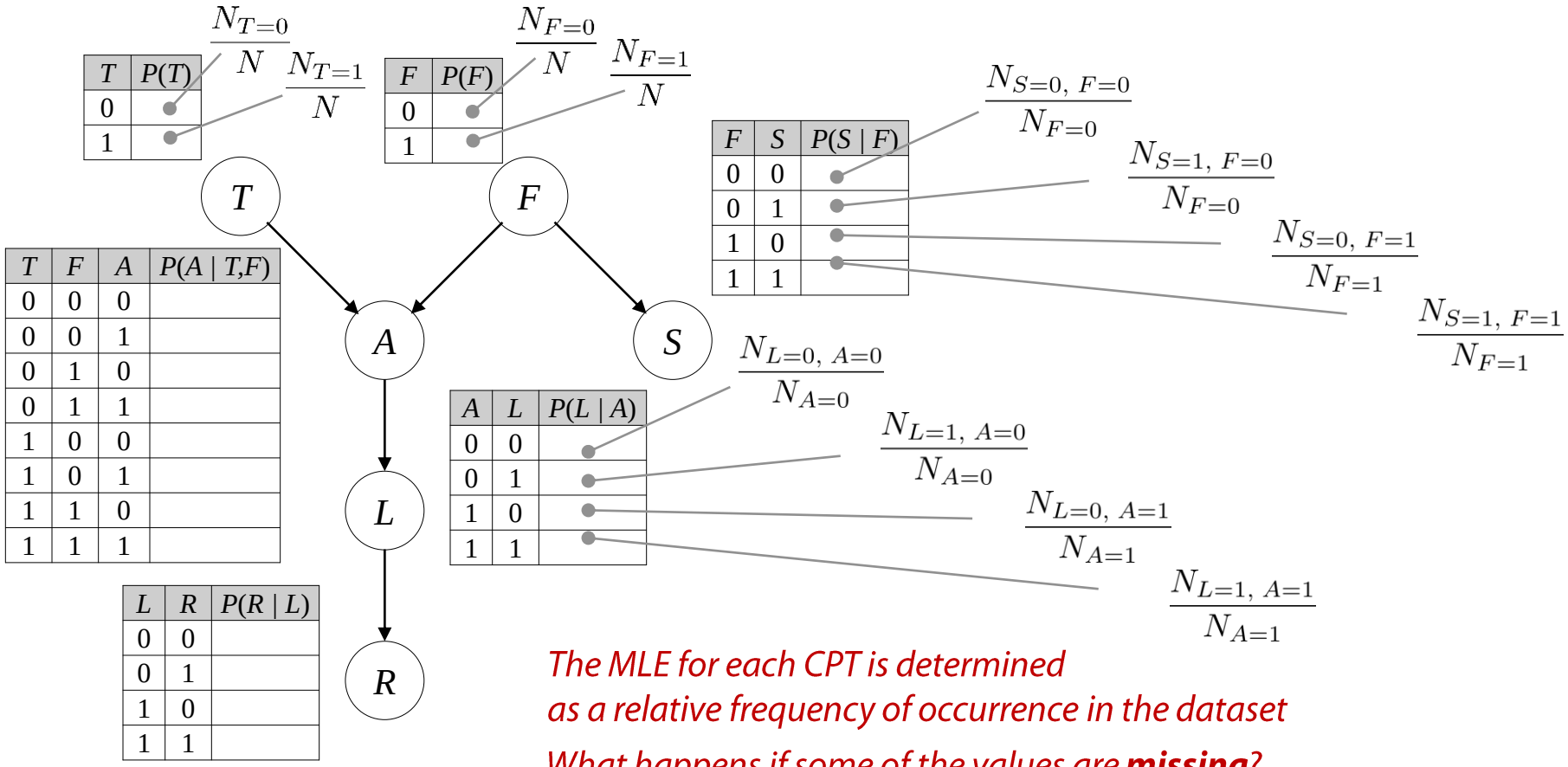
but this may be very hard to optimize in general

Learning CPTs for a graphical model via 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)*



T	P(T)
0	
1	

F	P(F)
0	
1	

F	S	P(S | F)
0	0	
0	1	
1	0	
1	1	

T	F	A	P(A | T,F)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A	L	P(L | A)
0	0	
0	1	
1	0	
1	1	

L	R	P(R | L)
0	0	
0	1	
1	0	
1	1	

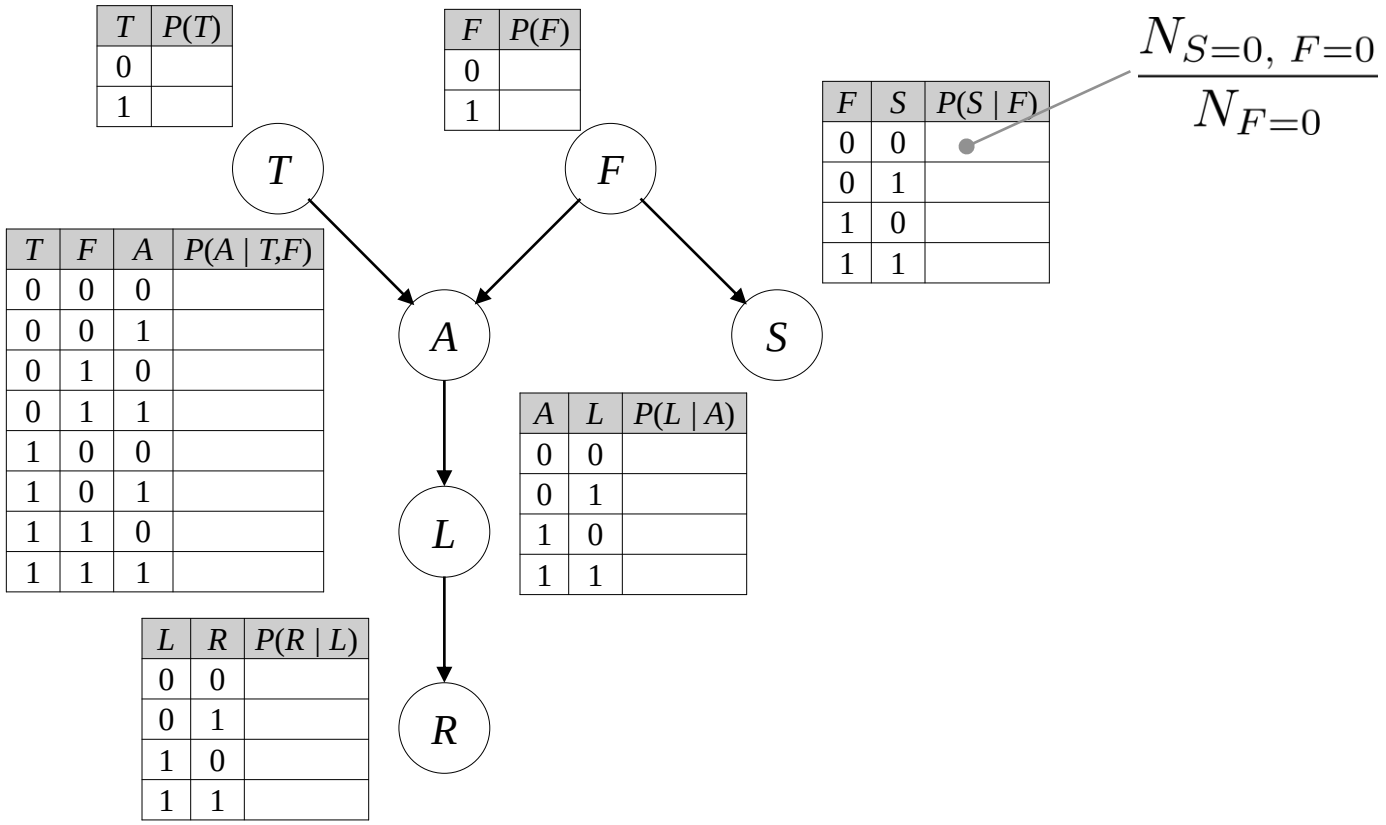
The MLE for each CPT is determined
as a relative frequency of occurrence in the dataset

*What happens if some of the values are **missing**?*

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



T	P(T)
0	
1	

F	P(F)
0	
1	

F	S	P(S | F)
0	0	
0	1	
1	0	
1	1	

T	F	A	P(A | T,F)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A	L	P(L | A)
0	0	
0	1	
1	0	
1	1	

L	R	P(R | L)
0	0	
0	1	
1	0	
1	1	

Learning CPTs for a graphical model via EM

Fundamental idea: using probabilities of missing observations

Let's consider a dataset $D = \left\{ \boldsymbol{X}^{(i)} \right\}_{i=1}^{N}$

Each data item $\boldsymbol{X}^{(i)}$ may contain some missing data

Example: $\boldsymbol{X}^{(i)} = (X_1^{(i)} = x_1, X_2^{(i)} = x_2, X_3^{(i)} = ?)$

Define $\tilde{\boldsymbol{X}}^{(i)}$ as one possible ***completion*** of $\boldsymbol{X}^{(i)}$, namely one in which there are no missing data

Example: assuming that $X_3 \in \{0, 1\}$, $(X_1^{(i)} = x_1, X_2^{(i)} = x_2, X_3^{(i)} = 0)$ $(X_1^{(i)} = x_1, X_2^{(i)} = x_2, X_3^{(i)} = 1)$ *are the two possible completions of* $\boldsymbol{X}^{(i)}$

Note that there will be as many completions of $\boldsymbol{X}^{(i)}$ *as there are combinations of possible values for the missing data*
For any complete observation, $\tilde{\boldsymbol{X}}^{(i)} = \boldsymbol{X}^{(i)}$ *: there is only one possible completion that coincides with the data item itself*

Likewise, $\boldsymbol{X}_{obs}^{(i)}$ is the part of $\boldsymbol{X}^{(i)}$ which contain the actual observations

Example: $\boldsymbol{X}_{obs}^{(i)} = (X_1^{(i)} = x_1, X_2^{(i)} = x_2)$

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:

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

In the EM algorithm, use *estimated* occurrences:

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

Sum extended to all possible completions

In words, any incomplete observations 'splits up' and contributes with the probabilities of possible completions
Note that, when all observations are complete:

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

Learning CPTs for a graphical model via EM

Fundamental idea: using probabilities of observations

Algorithm:

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

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

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