

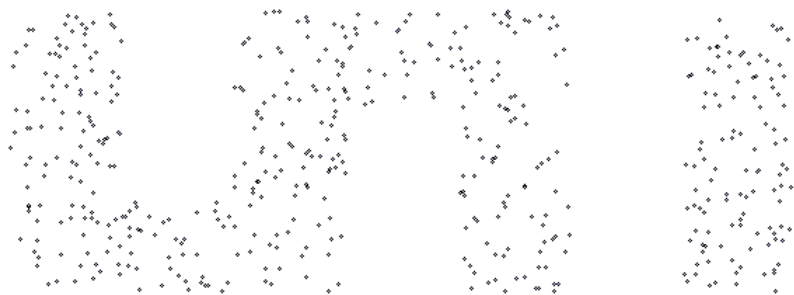
Artificial Intelligence

Unsupervised Learning

Marco Piastra

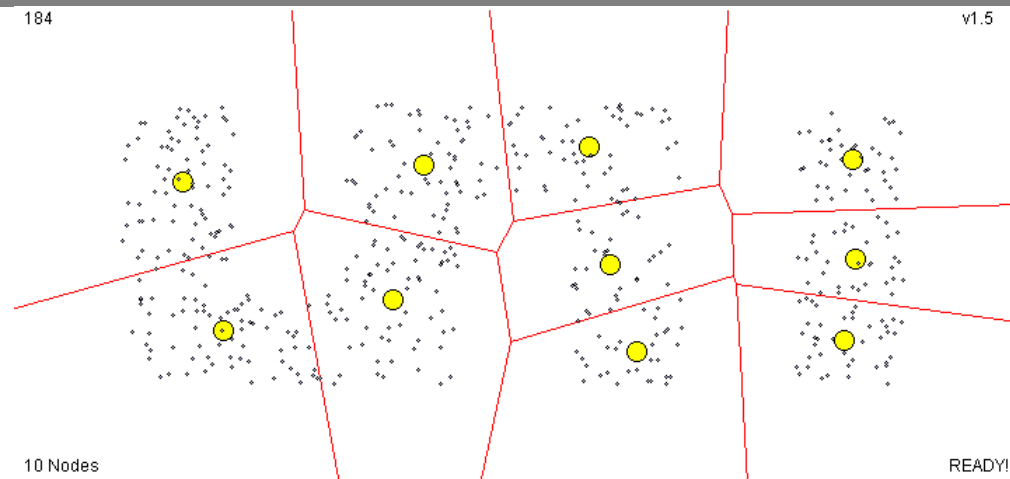
Vector quantization

184



v1.5

184



v1.5

10 Nodes

READY!

10 Nodes

READY!

Original data

Quantization (compression via prototypes)

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

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

- 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)

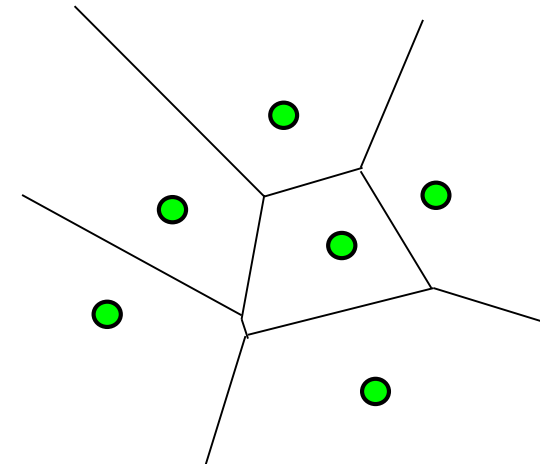
k-means (Generalized Lloyd's Algorithm – Vector quantization)

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 θ



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

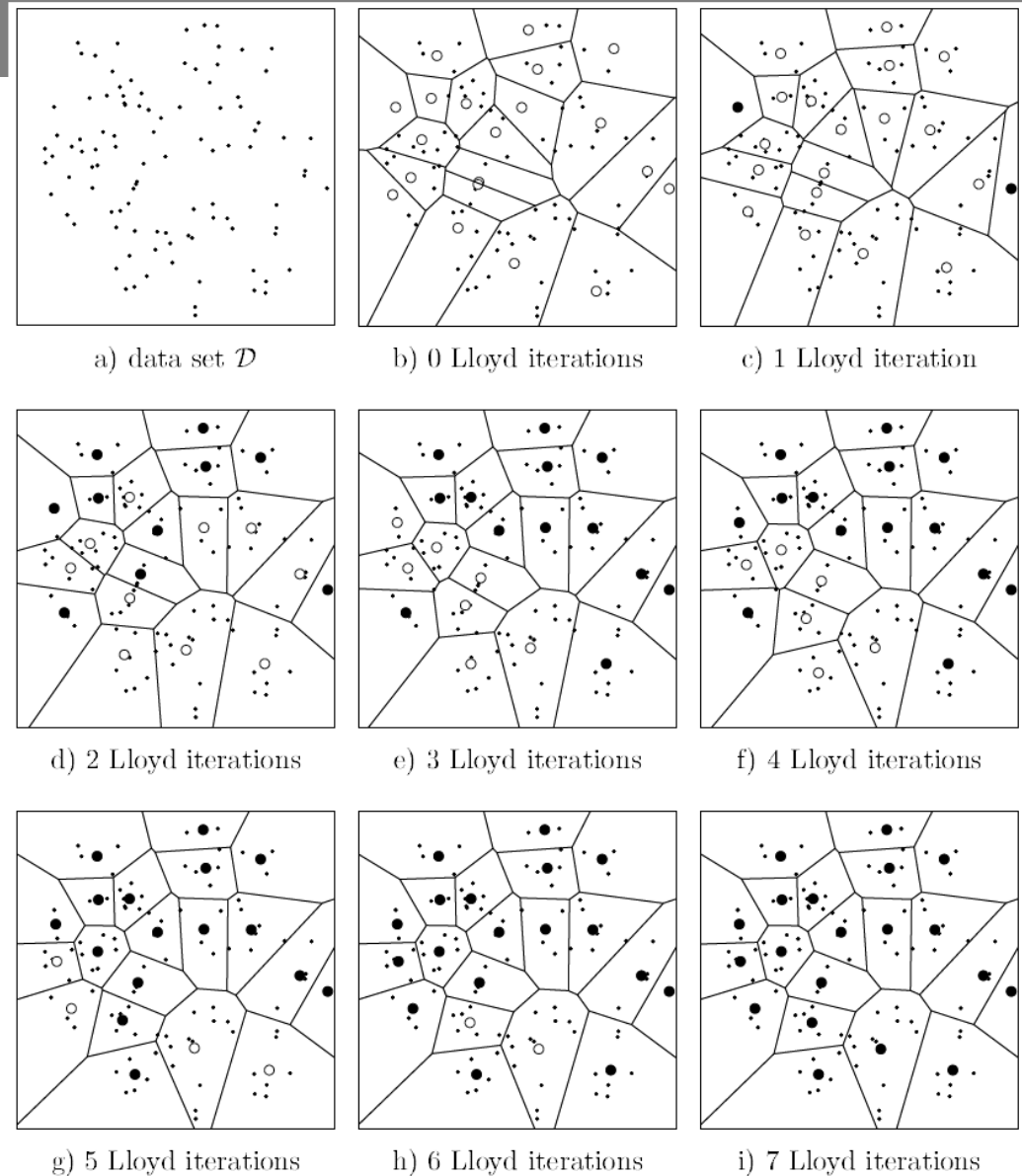
$$\mathbf{w}_j = \frac{1}{|\{\mathbf{x}_i \in V(\mathbf{w}_j)\}|} \sum_{\{\mathbf{x}_i \in V(\mathbf{w}_j)\}} \mathbf{x}_i$$

- 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



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 $\{X_i\}$ is

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

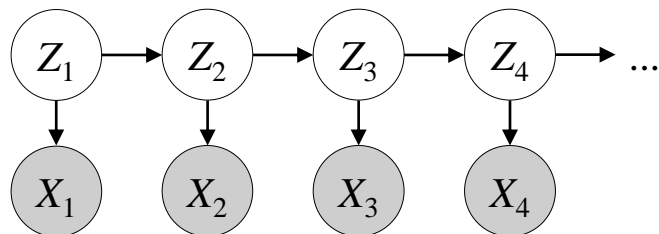
the sum is over all possible combinations of values of the random variables

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

The extension to the continuous case is obvious

Incomplete observations

Example: 'Hidden Markov' model



Terminology:

hidden = latent = always unobserved

missing = unobserved (in a data set)

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

$$P(\{X_i\}, \{Z_j\}) = P(Z_1) P(X_1 | Z_1) \prod_{i=2}^n P(Z_i | Z_{i-1}) P(X_i | Z_i) \quad \text{Joint distribution}$$

■ Problem

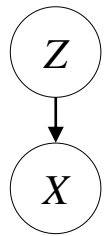
MLE of parameters θ starting from *partial* observations of the $\{X_i\}$ variables only

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

$$L(\theta | D) = P(D | \theta) = \sum_{\{Z_j\}} P(D, \{Z_j\} | \theta)$$

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

Expectation Maximization: a preliminary example



a Maximum likelihood

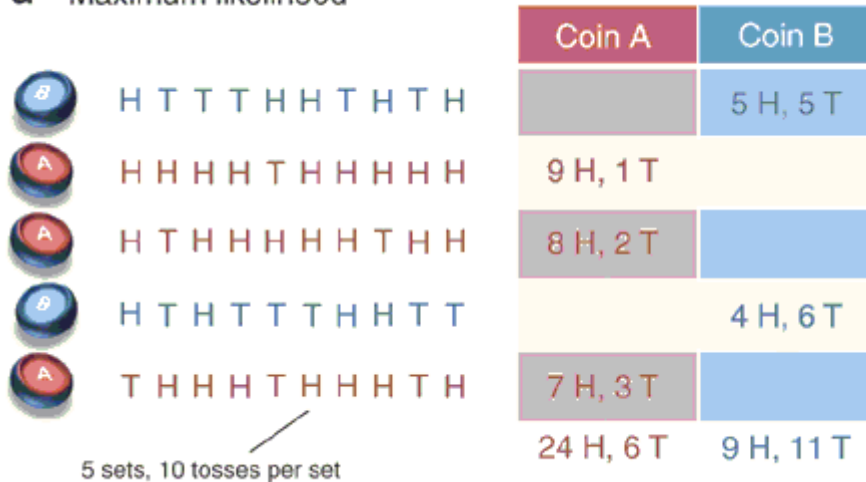


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

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

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

- An experiment with two coins

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

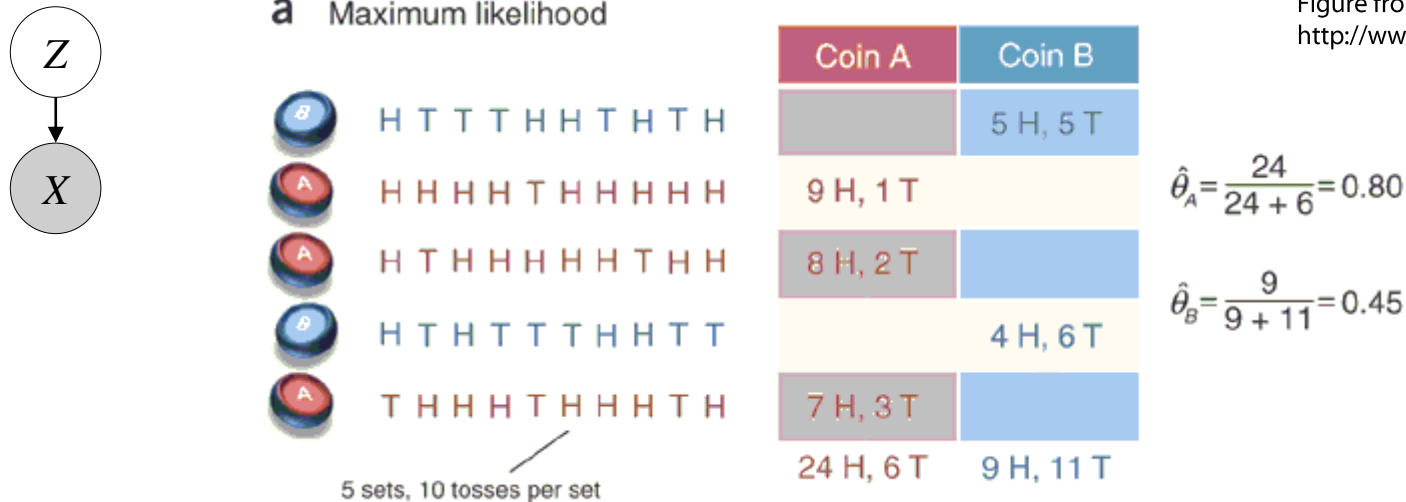
Random variables: X number of heads, Z selected coin (i.e A or B)

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

When the results are fully observable, by MLE:

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

Expectation Maximization: a preliminary example



- An experiment with two coins

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

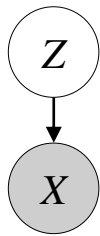
Random variables: X number of heads, Z selected coin (i.e A or B)

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

- What if Z 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

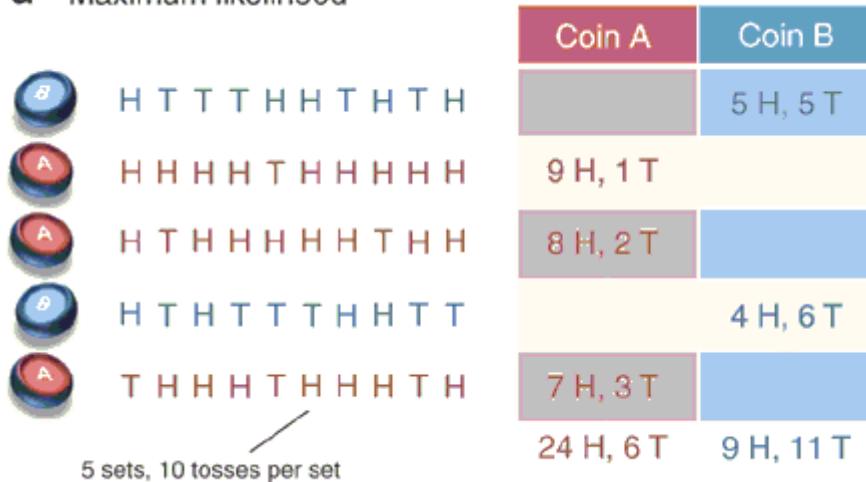


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

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

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

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

Likelihood

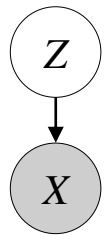
$$P(D | \theta) = P(\{x^{(i)}\} | \theta) = \sum_{\{Z^{(i)}\}} P(\{\langle x^{(i)}, Z^{(i)} \rangle\} | \theta)$$

MLE

$$\theta^* := \operatorname{argmax}_{\theta} \sum_{\{Z^{(i)}\}} P(\{\langle x^{(i)}, Z^{(i)} \rangle\} | \theta)$$

* This optimization is intractable, in general

Expectation Maximization: a preliminary example



a Maximum likelihood

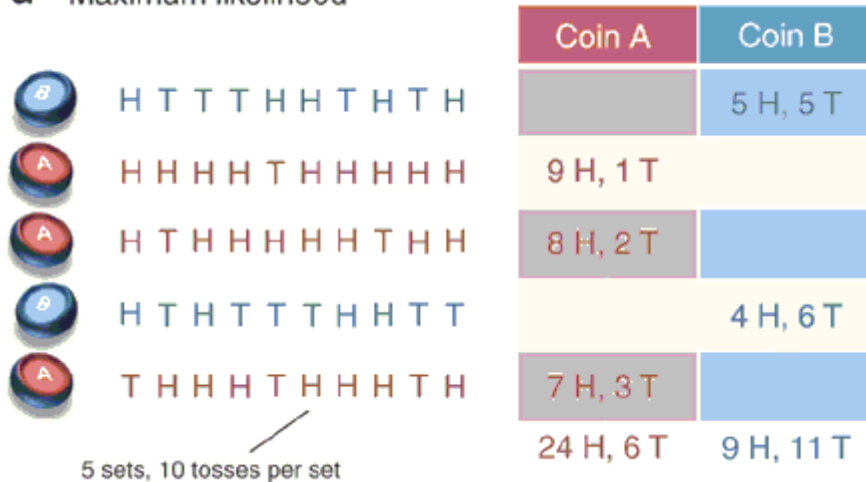


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

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

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

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

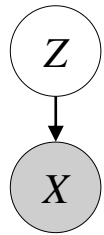
Intuitive idea: use expected values for unobserved variables

- Define an initial (random) guess $\hat{\theta}^{(0)}$
- Create an intermediate function $Q(\{\langle x^{(i)}, Z^{(i)} \rangle\} | \hat{\theta}^{(t)})$ based on $\mathbb{E}_{Z^{(i)}} [X^{(i)} | \hat{\theta}^{(t)}]$
- Maximize

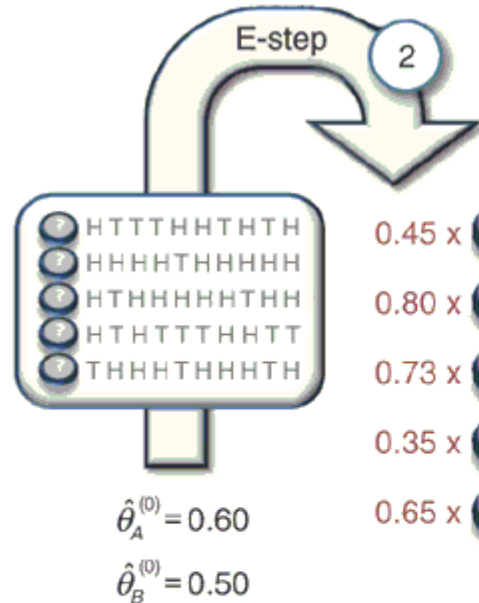
$$\hat{\theta}^{(t+1)} = \operatorname{argmax}_{\theta} \sum_i Q(\{\langle x^{(i)}, Z^{(i)} \rangle\} | \hat{\theta}^{(t)})$$

- 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



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

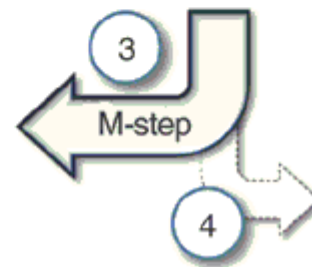
Initial random estimate of $\hat{\theta}_A, \hat{\theta}_B$



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

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

MLE given expected observations

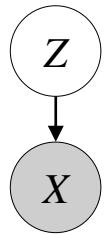


$$\hat{\theta}_A^{(10)} \approx 0.80$$

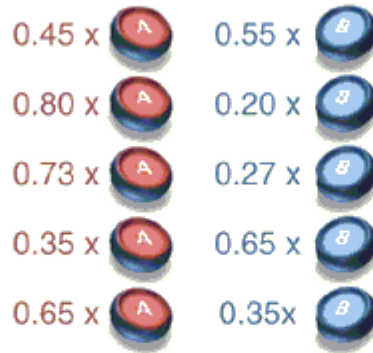
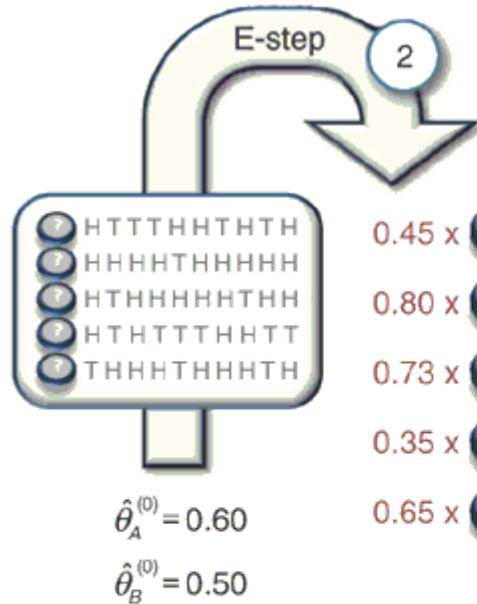
$$\hat{\theta}_B^{(10)} \approx 0.52$$

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

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

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

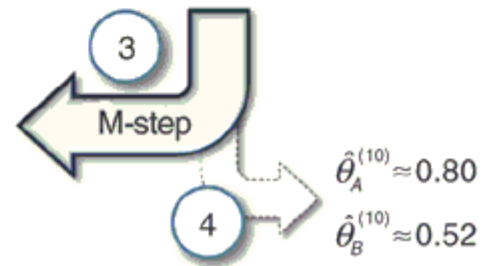
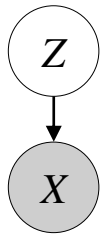


Figure from <http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html>

MLE given expected observations

Converged?

Expectation Maximization: a preliminary example



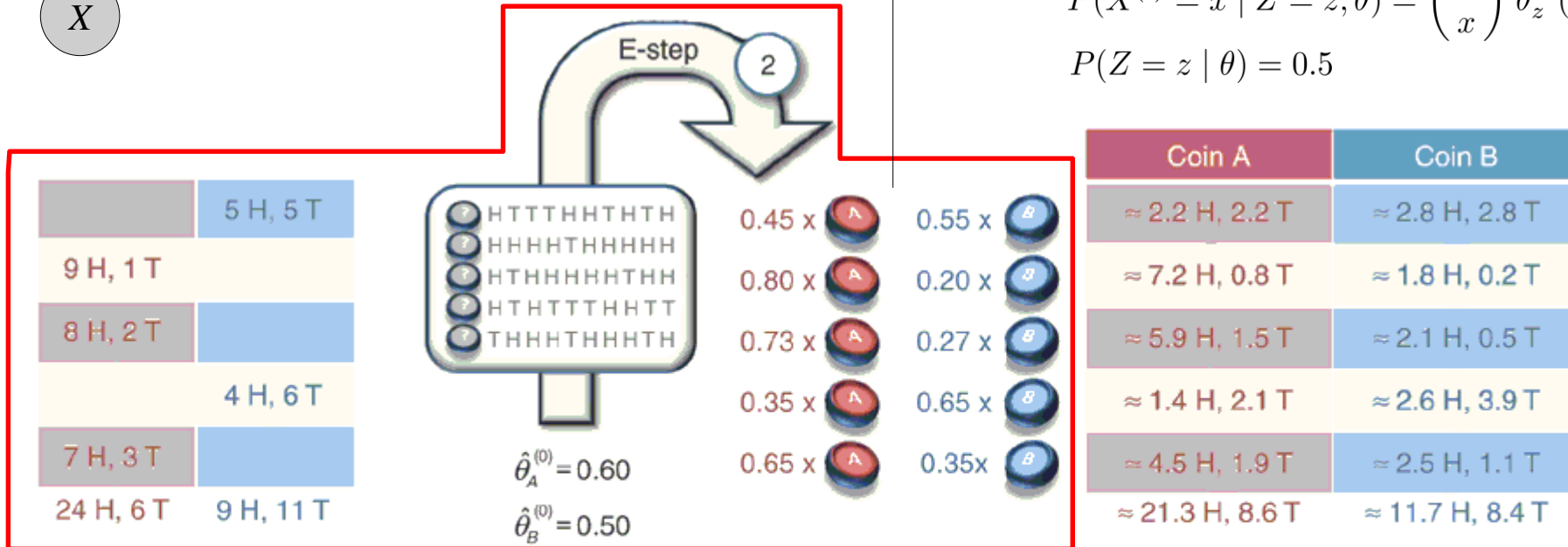
Compute the probability distribution of hidden observations

$$P(Z = z | X^{(i)} = x, \theta) = \frac{P(X^{(i)} = x | Z = z, \theta)P(Z = z | \theta)}{\sum_z P(X^{(i)} = x | Z = z, \theta)P(Z = z | \theta)}$$

where

$$P(X^{(i)} = x | Z = z, \theta) = \binom{10}{x} \theta_z^x (1 - \theta_z)^{10-x}$$

$$P(Z = z | \theta) = 0.5$$

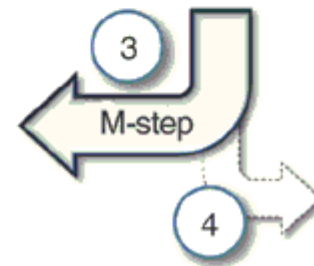


Initial random estimate of $\hat{\theta}_A, \hat{\theta}_B$



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

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

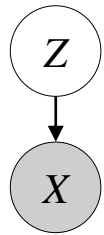


$$\hat{\theta}_A^{(10)} \approx 0.80$$

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

Converged?

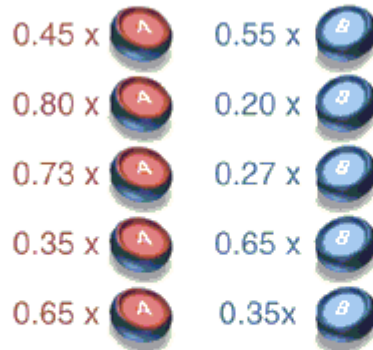
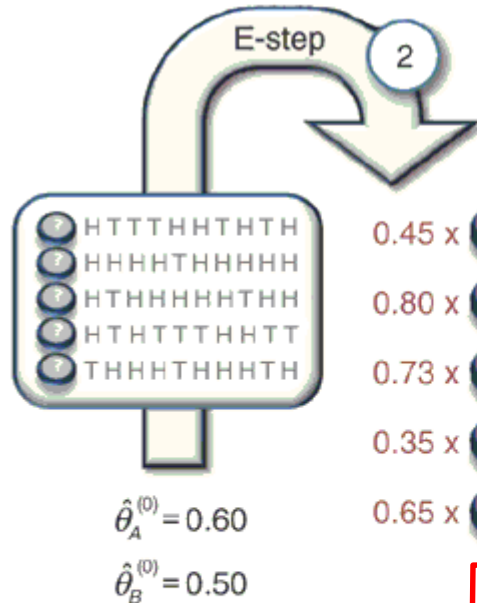
Expectation Maximization: a preliminary example



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

$$\mathbb{E}_Z[x \mid X = x, \theta] = \sum_z x P(Z = z \mid X = x, \theta) = x$$

	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

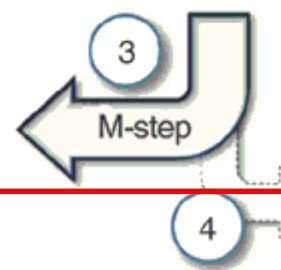
Initial random estimate of $\hat{\theta}_A, \hat{\theta}_B$



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

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

MLE using 'expected observations'



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

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

Converged?

An aside: Jensen's inequality

A relationship between probability and geometry

When f is convex function

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

f is **convex** when for any two points p_i and p_j the segment $(p_i - p_j)$ is not below f

That is, when

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

Furthermore, f is **strictly convex** when

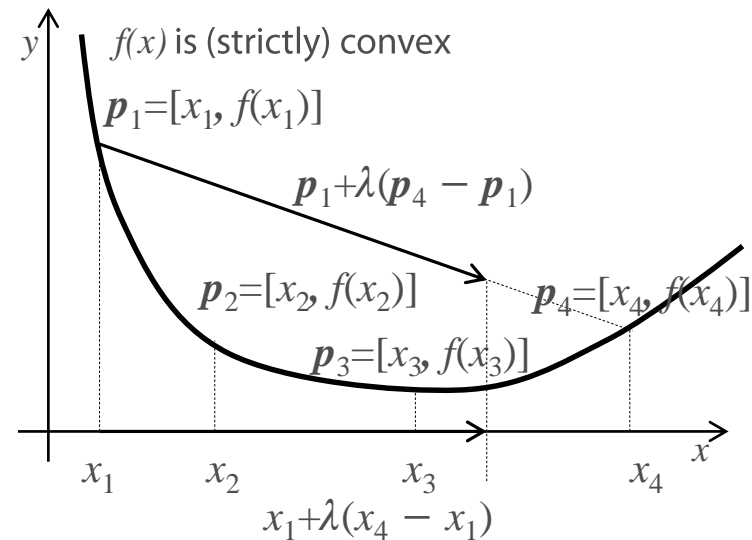
$$\lambda f(x_i) + (1-\lambda)f(x_j) > f(\lambda x_i + (1-\lambda)x_j) \quad \forall \lambda \in (0,1)$$

Corollary:

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

$$f(E[\{X_i\}]) = E[f(\{X_i\})]$$

Dual results also hold for concave functions



An aside: Jensen's inequality

A relationship between probability and geometry

When f is convex function

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

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

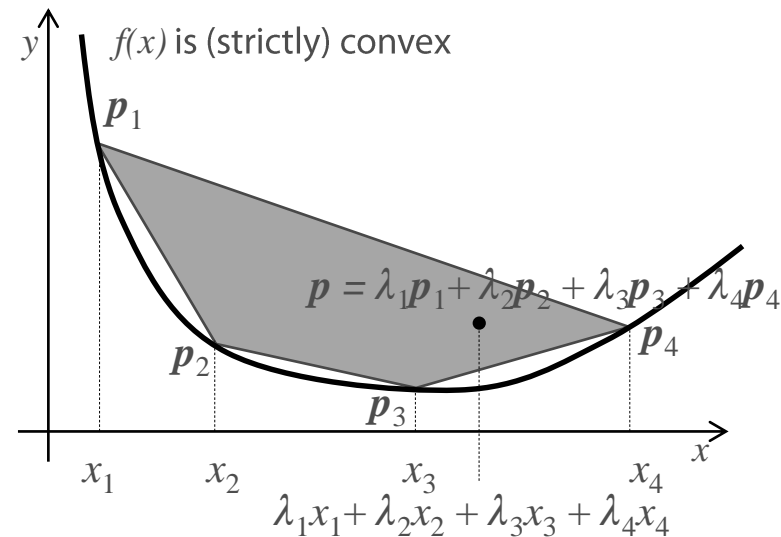
This is an **affine** combination if $\sum \lambda_i = 1$
and it is a **convex** combination if also $\lambda_i \geq 0, \forall i$

When the λ_i define a probability, then 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

$$L(\theta | D) = P(D | \theta) = \prod_m P(D_m | \theta)$$

$$\ell(\theta | D) = \sum_m \log P(D_m | \theta) = \sum_m \log \sum_{\{Z_i\}} P(D_m, \{Z_i\} | \theta)$$

Arbitrary probability distributions

$$= \sum_m \log \sum_{\{Z_i\}} Q_m(\{Z_i\}) \frac{P(D_m, \{Z_i\} | \theta)}{Q_m(\{Z_i\})}$$

Jensen's inequality: log is concave

$$= \sum_m \log E_{Q_m(\{Z_i\})} \left[\frac{P(D_m, \{Z_i\} | \theta)}{Q_m(\{Z_i\})} \right] \geq \sum_m E_{Q_m(\{Z_i\})} \left[\log \frac{P(D_m, \{Z_i\} | \theta)}{Q_m(\{Z_i\})} \right]$$
$$= \sum_m \sum_{\{Z_i\}} Q_m(\{Z_i\}) \log \frac{P(D_m, \{Z_i\} | \theta)}{Q_m(\{Z_i\})}$$

Expectation–Maximization (EM) Algorithm

Alternate optimization (coordinate ascent)

Log-likelihood function:

$$\ell(\theta | D) \geq \sum_m \sum_{\{Z_i\}} Q_m(\{Z_i\}) \log \frac{P(D_m, \{Z_i\} | \theta)}{Q_m(\{Z_i\})}$$

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

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

$$Q_m(\{Z_i\}) := \frac{P(D_m, \{Z_i\} | \theta)}{\sum_{\{Z_i\}} P(D_m, \{Z_i\} | \theta)} = \frac{P(D_m, \{Z_i\} | \theta)}{P(D_m | \theta)} = P(\{Z_i\} | D_m, \theta) =: p_{\{Z_i\}}^{(m)}$$

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

Then maximize the log-likelihood while keeping $Q_m(\{Z_i\})$ constant

$$\begin{aligned} \theta^* &= \arg \max_{\theta} \sum_m \sum_{\{Z_i\}} p_{\{Z_i\}}^{(m)} \log \frac{P(D_m, \{Z_i\} | \theta)}{p_{\{Z_i\}}^{(m)}} && \text{This is also called the } \underline{\text{entropy}} \text{ of } Q_m(\{Z_i\}) \\ & && \text{(i.e. a constant measure of the distribution)} \\ &= \arg \max_{\theta} \sum_m \left(\sum_{\{Z_i\}} p_{\{Z_i\}}^{(m)} \log P(D_m, \{Z_i\} | \theta) - \sum_{\{Z_i\}} p_{\{Z_i\}}^{(m)} \log p_{\{Z_i\}}^{(m)} \right) \\ &= \arg \max_{\theta} \sum_m \sum_{\{Z_i\}} p_{\{Z_i\}}^{(m)} \log P(D_m, \{Z_i\} | \theta) \end{aligned}$$

Expectation– Maximization (EM) Algorithm

Alternate optimization (coordinate ascent)

Log-likelihood function and its estimator:

$$\ell(\theta | D) \geq \sum_m \sum_{\{Z_i\}} Q_m(\{Z_i\}) \log \frac{P(D_m, \{Z_i\} | \theta)}{Q_m(\{Z_i\})}$$

Algorithm:

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

$$p_{\{Z_i\}}^{(m)} = Q_m(\{Z_i\}) = P(\{Z_i\} | D_m, \theta)$$

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

$$\theta^* = \arg \max_{\theta} \sum_m \sum_{\{Z_i\}} p_{\{Z_i\}}^{(m)} \log P(D_m, \{Z_i\} | \theta)$$

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

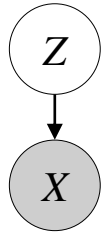
The algorithm converges to a local maximum of the log-likelihood

The effectiveness of algorithm depends on the form of the distribution (see step3):

$$P(D_m, \{Z_i\} | \theta)$$

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

EM Algorithm: mixture of Gaussians



Model:

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

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

Multivariate normal distribution

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

i.e. the condition probabilities are normal distributions

The observations are a set $D = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ 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 x_i in D compute the probabilities
 $p_k^{(m)} = P(Z = k | x^{(m)}, \phi_k, \mu_k, \Sigma_k) = \phi_k \cdot N(x^{(m)}; \mu_k, \Sigma_k)$
- 3) (M-step) Compute the new estimates for the parameters

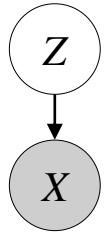
$$\phi_k = \frac{1}{n} \sum_m p_k^{(m)}$$

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

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

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

EM Algorithm: mixture of Gaussians



Model:

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

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

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

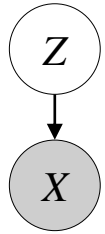
i.e. the condition probabilities are normal distributions

The observations are a set $D = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ of points in \mathbb{R}^d

Proof (of the M-step):

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

EM Algorithm: mixture of Gaussians



Model:

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

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

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

i.e. the condition probabilities are normal distributions

The observations are a set $D = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ of points in \mathbb{R}^d

Proof (of the M-step):

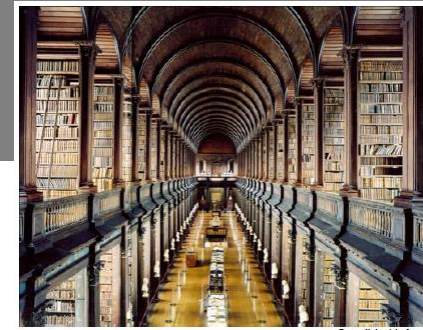
$$\begin{aligned} & \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left(\log\left((2\pi)^{-d/2} (\det \Sigma_k)^{-1/2}\right) + \left(-\frac{1}{2}(x^{(m)} - \mu_k)^T \Sigma_k^{-1} (x^{(m)} - \mu_k)\right) + \log \phi_k \right) \\ &= \frac{\partial}{\partial \mu_j} \sum_m \sum_k p_k^{(m)} \left(-\frac{1}{2}(x^{(m)}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

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

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

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

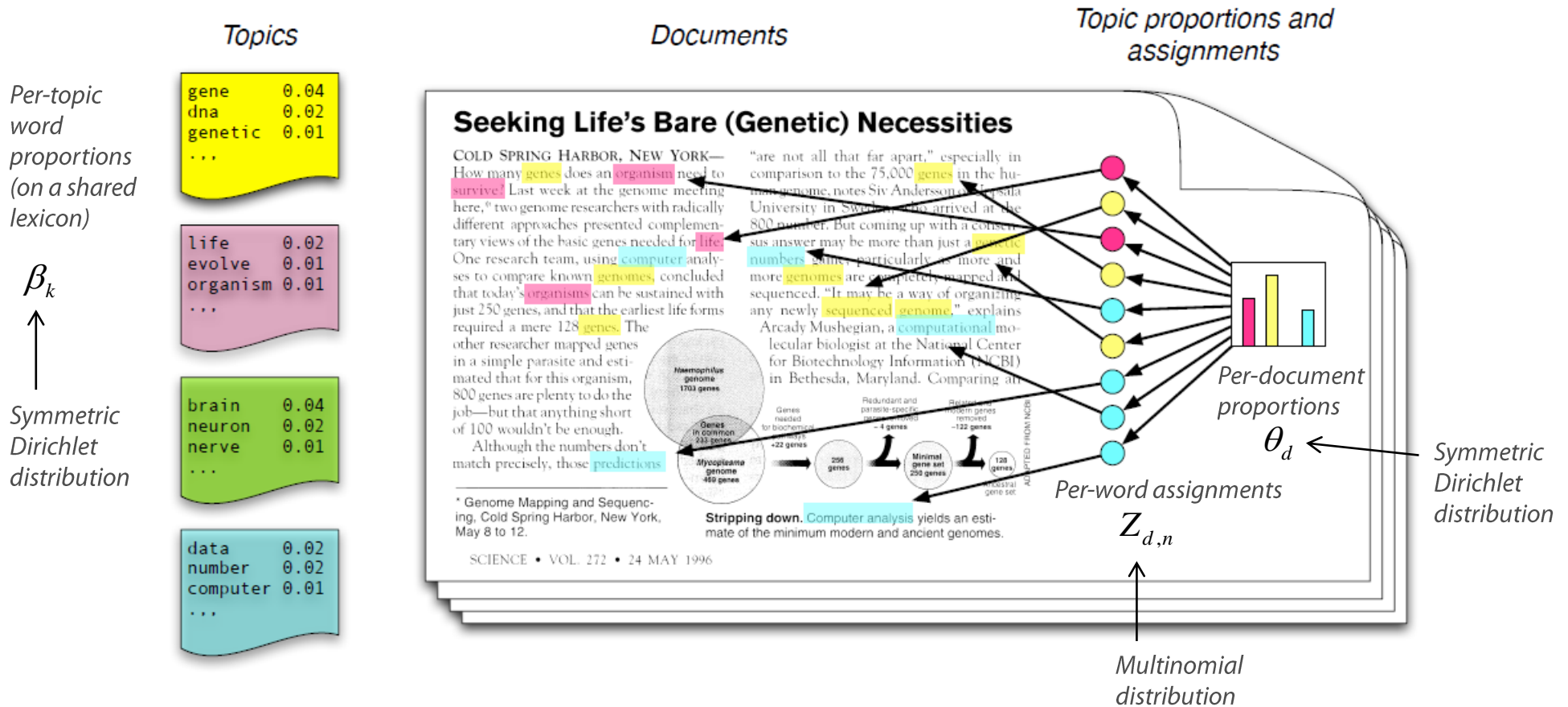
Topic modeling



Candida Hofer

Topic modeling

Classifying a (large) corpus of digital documents relying on word counting only



Multinomial distribution

- Bernoulli

Head or Tail?

$$P(X = 1) = \theta, \quad P(X = 0) = 1 - \theta$$

- Binomial

n heads out of N coin tosses

$$P(X = n) = \binom{N}{n} \theta^n (1 - \theta)^{(N-n)}$$

- Categorical

The result of throwing a dice with k faces

$$P(X = 1) = \theta_1, \quad P(X = k) = \theta_k, \quad \sum_{i=1}^k \theta_i = 1$$

- Multinomial

Obtaining an outcome combination x_1, \dots, x_k in N throws of a k -faced dice, with $\sum_{i=1}^k x_i = N$

$$P(X_1 = x_1, \dots, X_k = x_k) = \frac{N!}{x_1! \dots x_k!} \prod_{i=1}^k \theta_i^{x_i}$$

Dirichlet distribution

■ Beta distribution

What do you think about a coin after obtaining $(\alpha_1 - 1)$ heads and $(\alpha_2 - 1)$ tails?

$$\text{Beta}(x_1, x_2; \alpha_1, \alpha_2) := \frac{x_1^{\alpha_1-1} \cdot x_2^{\alpha_2-1}}{\text{B}(\alpha_1, \alpha_2)}, \quad x_1 + x_2 = 1$$

This is just a re-writing of the 'standard' formula:

$$\text{Beta}(x; \alpha, \beta) := \frac{x^{\alpha-1} (1-x)^{\beta-1}}{\text{B}(\alpha, \beta)}$$

■ Dirichlet distribution

What do you think about a k -faced dice after obtaining $(\alpha_1 - 1), (\alpha_2 - 1) \dots (\alpha_k - 1)$ outcomes?

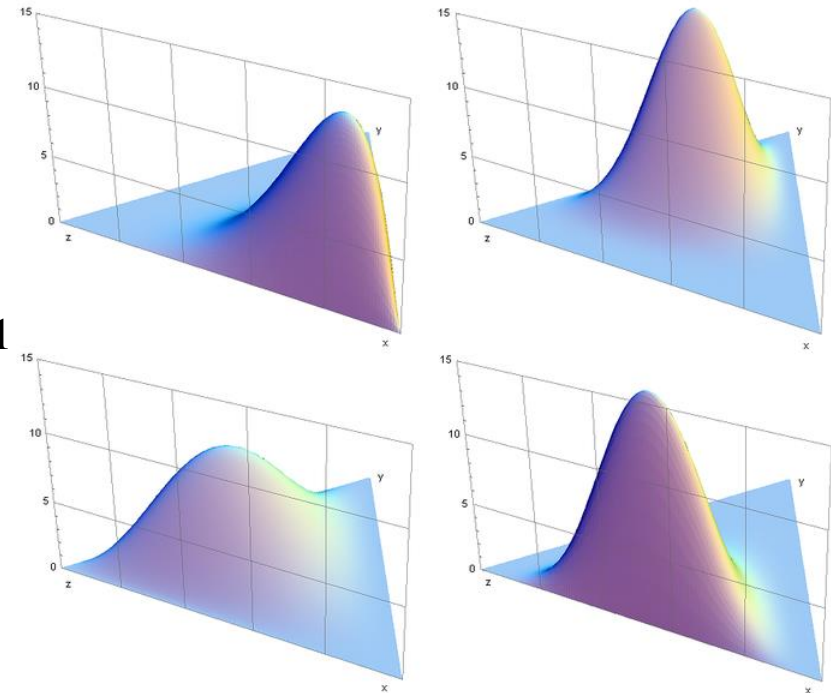
$$\text{D}(x_1, \dots, x_k; \alpha_1, \dots, \alpha_k) := \frac{\prod_{i=1}^k x_i^{\alpha_i-1}}{\text{B}(\alpha_1, \dots, \alpha_k)}, \quad \sum_{i=1}^k x_i = 1$$

where

$$\text{B}(\alpha_1, \dots, \alpha_k) := \frac{\prod_{i=1}^k \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}$$

is the *multivariate Beta function*.

The Dirichlet distribution is the **conjugate prior** of the Multinomial distribution



examples of Dirichlet distributions, for $k = 3$
(from Wikipedia)

Dirichlet distribution

- Symmetric Beta distribution

i.e. when $\alpha = \beta$

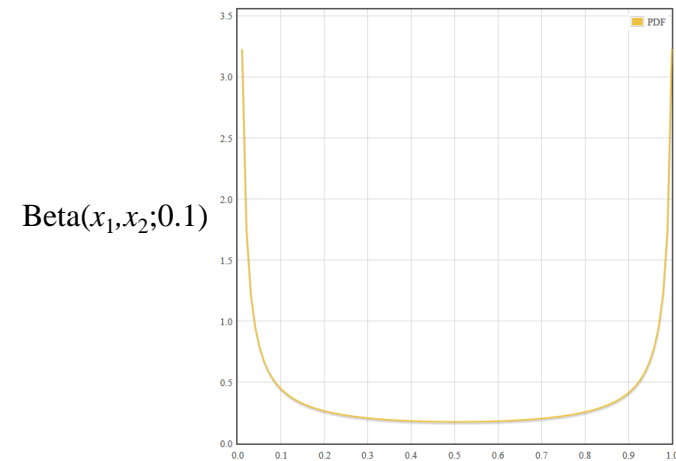
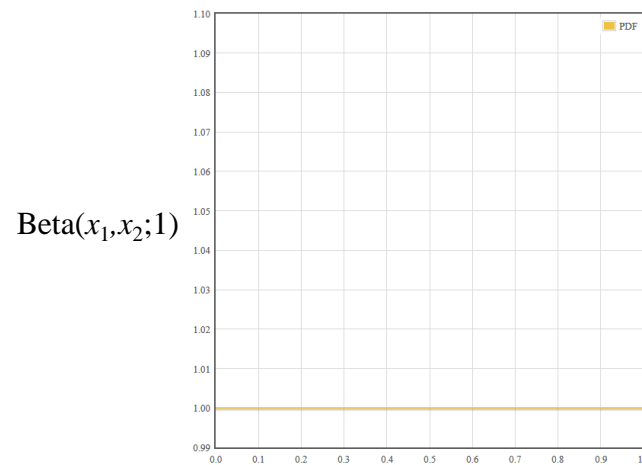
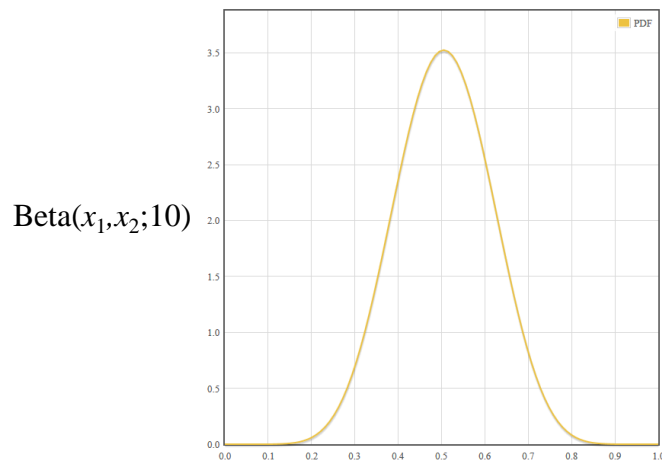
$$\text{Beta}(x_1, x_2; \alpha) := \frac{x_1^{\alpha-1} \cdot x_2^{\alpha-1}}{B(\alpha, \alpha)}, \quad x_1 + x_2 = 1$$

- Symmetric Dirichlet distribution

i.e. when $\alpha_1 = \alpha_2 = \dots = \alpha_k$

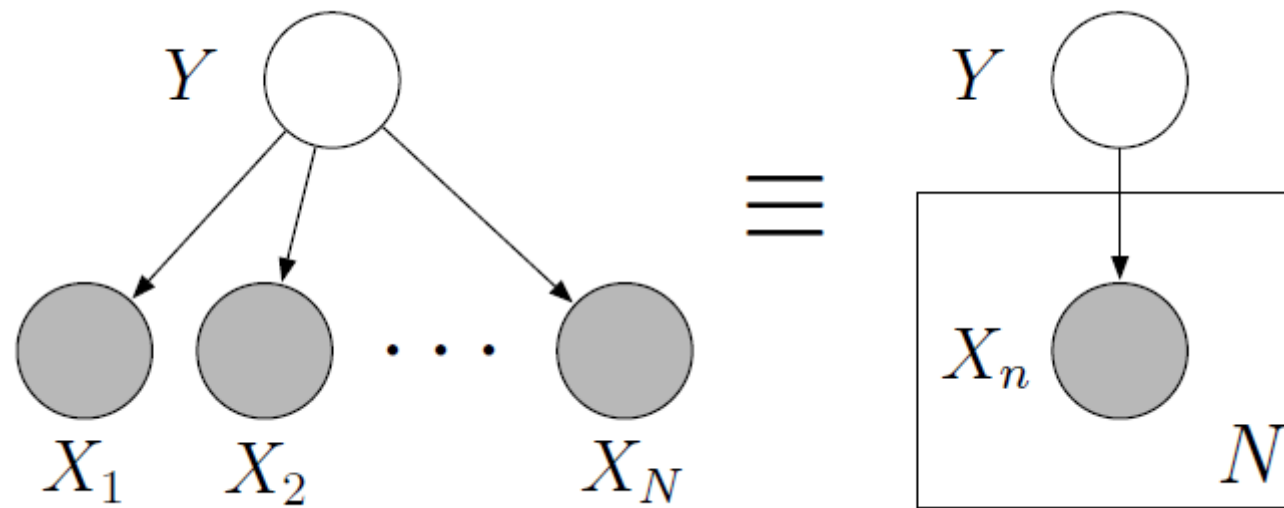
$$D(x_1, \dots, x_k; \alpha) := \frac{\prod_{i=1}^k x_i^{\alpha-1}}{B(\alpha, \dots, \alpha)}, \quad \sum_{i=1}^k x_i = 1$$

*Note: in both distributions, the parameters can be < 1
(this is true of the non-symmetric versions as well)*



An aside: plate notation

A shorthand notation for graphical models

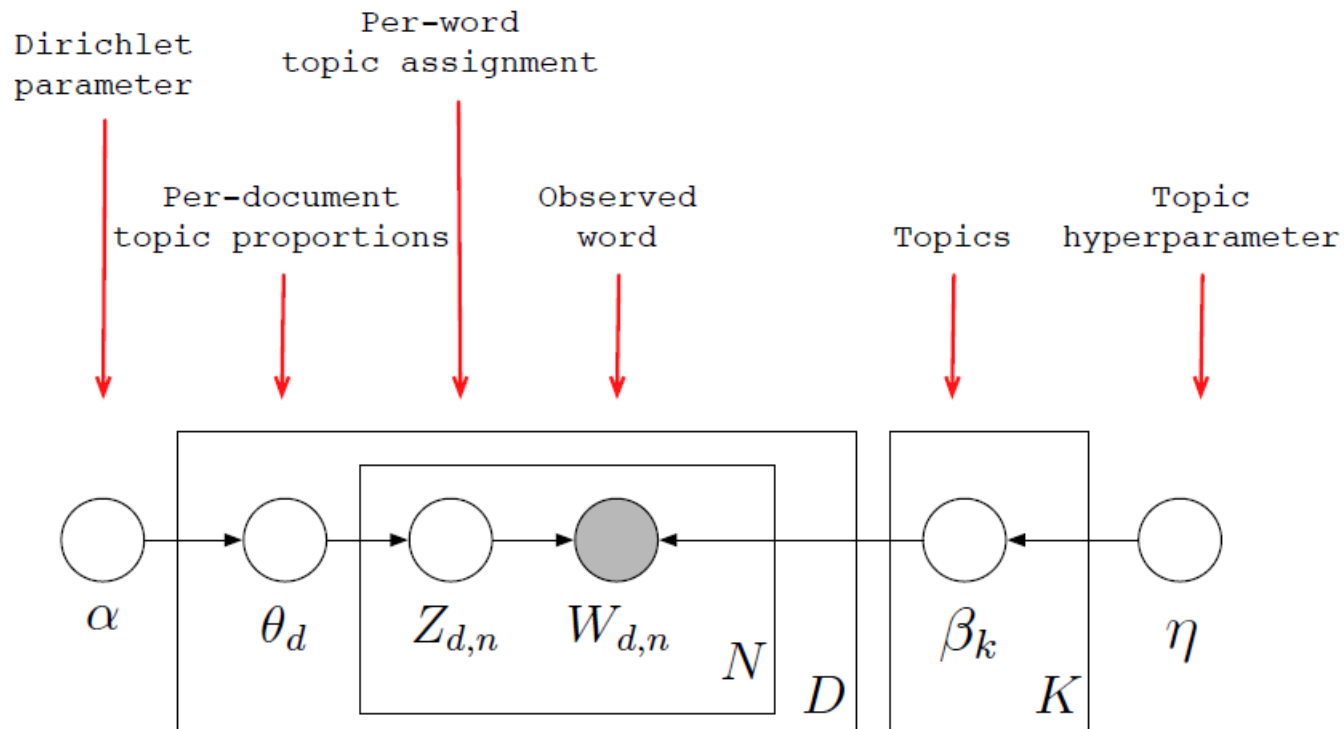


An example: Probabilistic Topic Models (Blei & Lafferty, 2009)

Classifying a corpus of documents with k (unknown) topics
when the only observable variables is the multiple occurrence of words

A *mixture model*:

each document belongs to multiple topics, with different probabilities

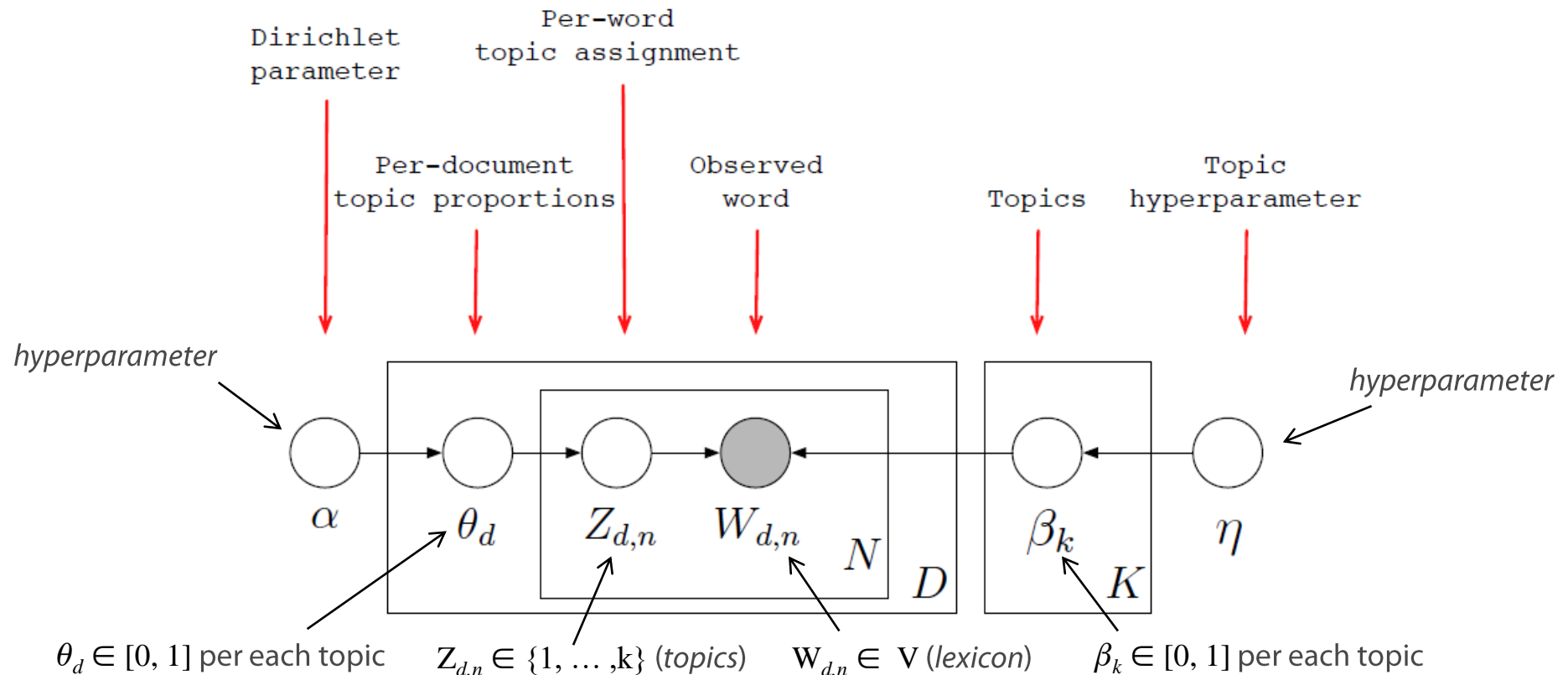


An example: Probabilistic Topic Models (Blei & Lafferty, 2009)

Classifying a corpus of documents with K (unknown) topics
when the only observable variables is the multiple occurrence of words

A mixture model:

each document belongs to multiple topics, with different probabilities

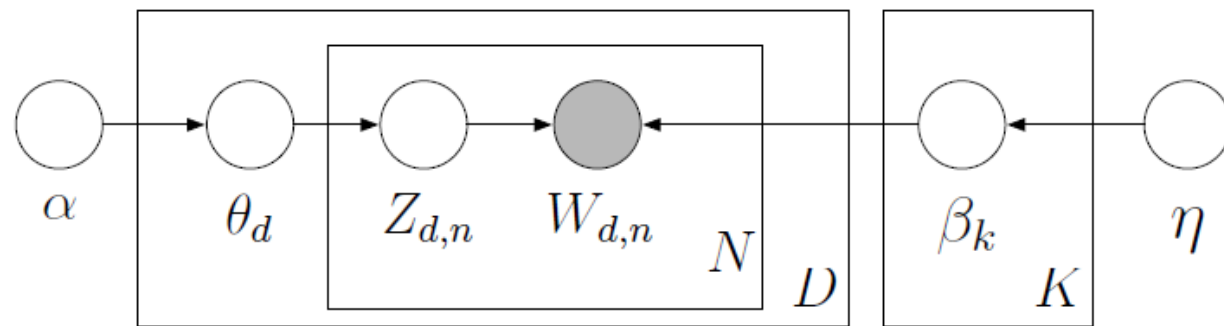


An example: Probabilistic Topic Models (Blei & Lafferty, 2009)

Classifying a corpus of documents as *mixtures* of K (unknown) topics when the only observable variables is the multiple occurrence of words

A three-level, mixture model:

each document belongs to multiple topics, with different probabilities



$$\prod_{i=1}^K p(\beta_i | \eta) \prod_{d=1}^D p(\theta_d | \alpha) \left(\prod_{n=1}^N p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

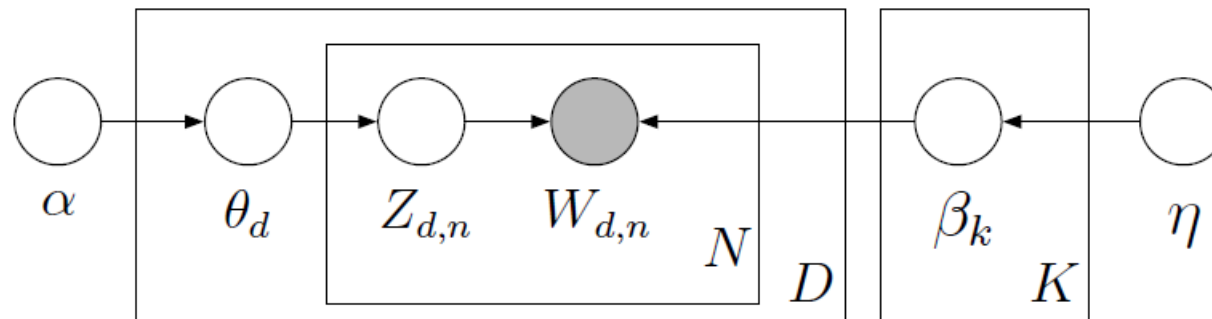
↑ Symmetric Dirichlet distributions
 ↑ Multinomial distributions

Latent Dirichlet Allocation (LDA)

Classifying a corpus of documents as *mixtures* of K (unknown) topics when the only observable variables is the multiple occurrence of words

A three-level, mixture model:

each document belongs to multiple topics, with different probabilities



A generative procedure:

- 1 Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, \dots, K\}$.
- 2 For each document:
 - 1 Draw topic proportions $\theta_d \sim \text{Dir}(\alpha)$.
 - 2 For each word:
 - 1 Draw $Z_{d,n} \sim \text{Mult}(\theta_d)$.
 - 2 Draw $W_{d,n} \sim \text{Mult}(\beta_{Z_{d,n}})$.

LDA: which results?

Identifying topics:
relative frequencies
of words that define a class

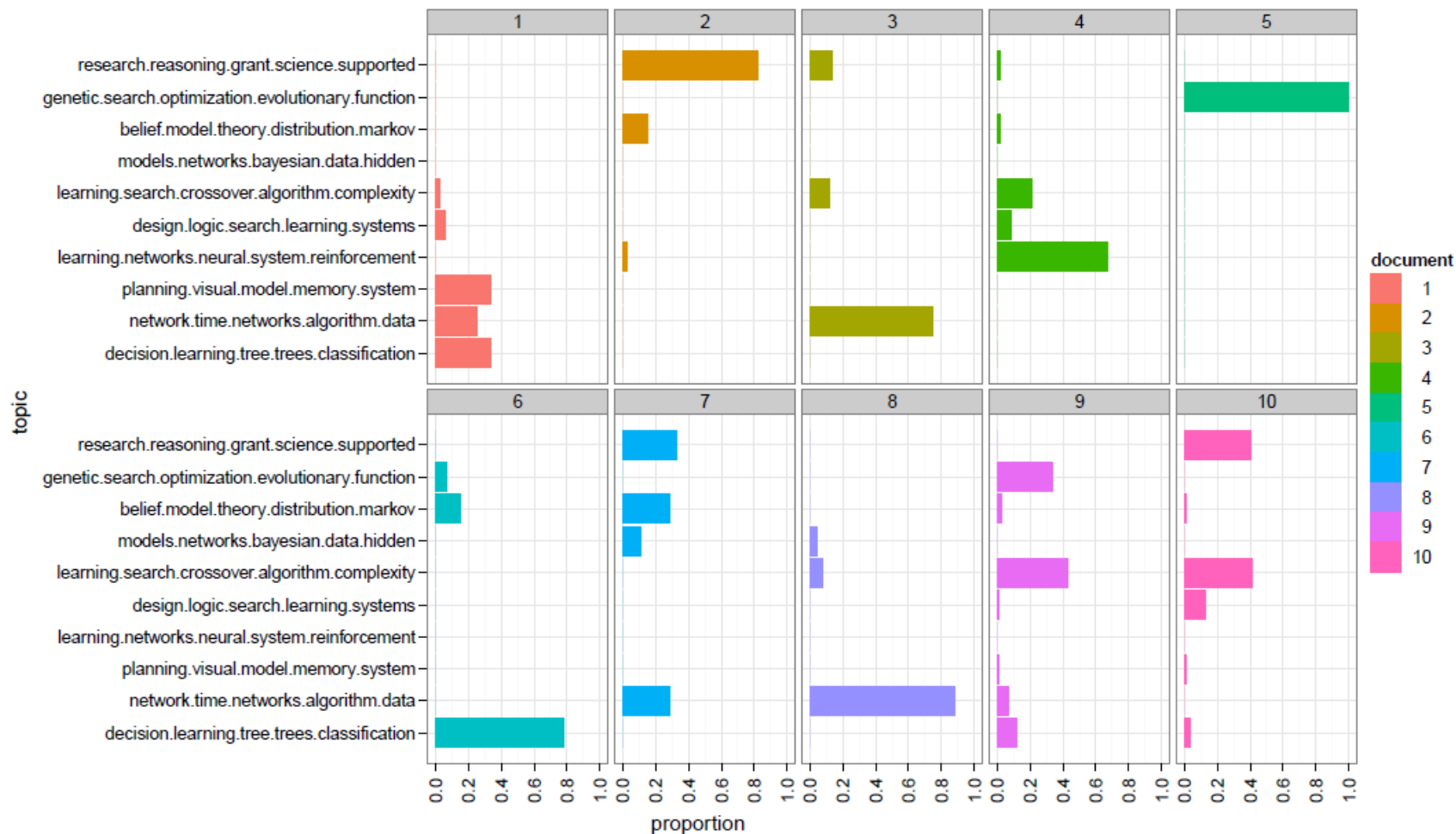
*Each box represents a topic
The size of words in a box
represents its relative proportion*



LDA: which results?

Classifying documents: *relative topic assignment proportions*

Each topic is represented by a list of most relevant words



LDA in practice

There exist multiple methods

Mean-Field Variational Inference (Blei et al. 2003)

(not discussed here – see links to the literature)

(It is a sort of generalization of the EM algorithm)

Many software implementations around: e.g. Apache Mahout

Real-world examples

The OCR'ed collection of Science from 1990-2000 [2009]

- *17K documents*
- *11M words*
- *20K unique terms (stop words and rare words removed)*

Model: 100 Topics

The New York Times online recommender system [2015]

See <http://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/>