Unsupervised Learning

Marco Piastra

k-means (Generalized Lloyd's Algorithm - Competitive learning)

Given a set $D = \{x_1, x_2, \ldots, x_n\}$ *of observations (i.e. points in* $\mathbf{R}^d$*)*
and a set $W = \{w_1, w_2, \ldots, w_k\}$ *of* k *landmarks (i.e. points in the same space)*

Clustering problem: position the k landmarks and assign each observation to a landmark so that the objective function is minimized:

$$J(D,W) := \sum_i \left\| x_i - w(x_i) \right\|^2$$

where $w(x_i)$ is the function that assign each observation to a landmark

k-means (*Generalized Lloyd's Algorithm - Competitive learning*)

Given a set $D = \{x_1, x_2, \ldots, x_n\}$ *of observations (i.e. points in* $\mathbf{R}^d$*)*
and a set $W = \{w_1, w_2, \ldots, w_k\}$ *of* k *landmarks (i.e. points in the same space)*

Clustering problem: position the k landmarks and assign each observation to a landmark so that the objective function is minimized:

$$J(D,W) := \sum_i \left\| x_i - w(x_i) \right\|^2$$

where $w(x_i)$ is the function that assign each observation to a landmark

Algorithm:

1) Position the k landmarks at random
2) Assign each observation to its closest landmark

$$w(x_i) := w_k \mid k = \operatorname{argmin}_j \left\| x_i - w_j \right\|$$

3) Position each landmark at the centroid (i.e. the geometric *mean*) of its observations

$$w_j := \frac{1}{|\{x_i \mid w(x_i) = w_j\}|} \sum_{\{x_i \mid w(x_i) = w_j\}} x_i$$

4) Go back to step 2) until unless no landmark was moved in step 3)

This algorithm converges to a local minimum of J

k-means (Generalized Lloyd's Algorithm - Competitive learning)

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

Step 2): Assume that the k landmarks have been positioned

The assignment

$$w(x_i) := w_k \mid k = \operatorname{argmin}_j \|x_i - w_j\|$$

minimizes each of the terms in $J(D,W) := \sum_i \|x_i - w(x_i)\|^2$

Step 3) Reposition the k landmarks while keeping the assignment $w(x_i)$ fixed

$$J(D,W) := \sum_{w_j} \sum_{\{x_i \mid w(x_i) = w_j\}} \|x_i - w_j\|^2$$

$$\frac{\partial}{\partial w_j} J(D,W) = \frac{\partial}{\partial w_j} \sum_{\{x_i \mid w(x_i) = w_j\}} \|x_i - w_j\|^2 = \frac{\partial}{\partial w_j} \sum_{\{x_i \mid w(x_i) = w_j\}} (x_i - w_j)^T \cdot (x_i - w_j)$$

$$= \frac{\partial}{\partial w_j} \sum_{\{x_i \mid w(x_i) = w_j\}} (x_i^T \cdot x_i + w_j^T \cdot w_j - 2x_i^T \cdot w_j) = 2 \sum_{\{x_i \mid w(x_i) = w_j\}} (w_j - x_i)$$

then, by imposing $\frac{\partial}{\partial w_j} J(D,W) = 0$

$$w_j := \frac{1}{|\{x_i \mid w(x_i) = w_j\}|} \sum_{\{x_i \mid w(x_i) = w_j\}} x_i$$

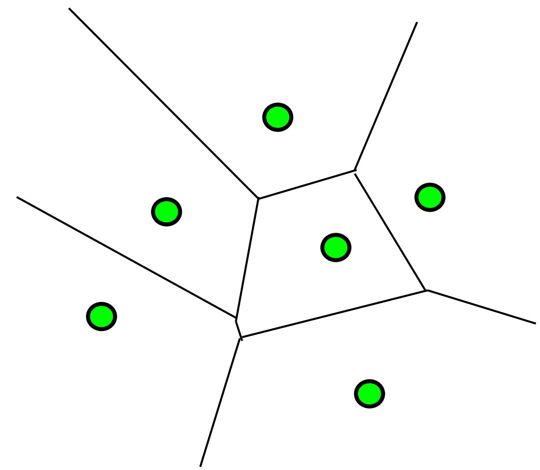
An alternative formulation

Given a set $D = \{x_1, x_2, \ldots, x_n\}$ *of observations (i.e. points in* $\mathbf{R}^d$*)*
and a set $W = \{w_1, w_2, \ldots, w_k\}$ *of* k *landmarks (i.e. points in the same space)*

Voronoi cell:

$$\boldsymbol{V}_i := \left\{ x \in \boldsymbol{R}^d \;\middle|\; \|x - w_i\| \le \|x - w_j\|, \forall j \ne i \right\}$$

Voronoi tesselation: the complex of all Voronoi cells of W



Algorithm:

1) Position the *k* landmarks at random
2) Assign observations in each Voronoi cell

$$\text{forall } x_i \in \boldsymbol{V}_j, \quad w(x_i) := w_j$$

3) Position each landmark at the centroid (i.e. the geometric *mean*) of its observations

$$w_j := \frac{1}{|\{x_i \mid w(x_i) = w_j\}|} \sum_{\{x_i \mid w(x_i) = w_j\}} x_i$$

4) Go back to step 2) until unless no landmark was moved in step 3)

k-means

An example run of the algorithm

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

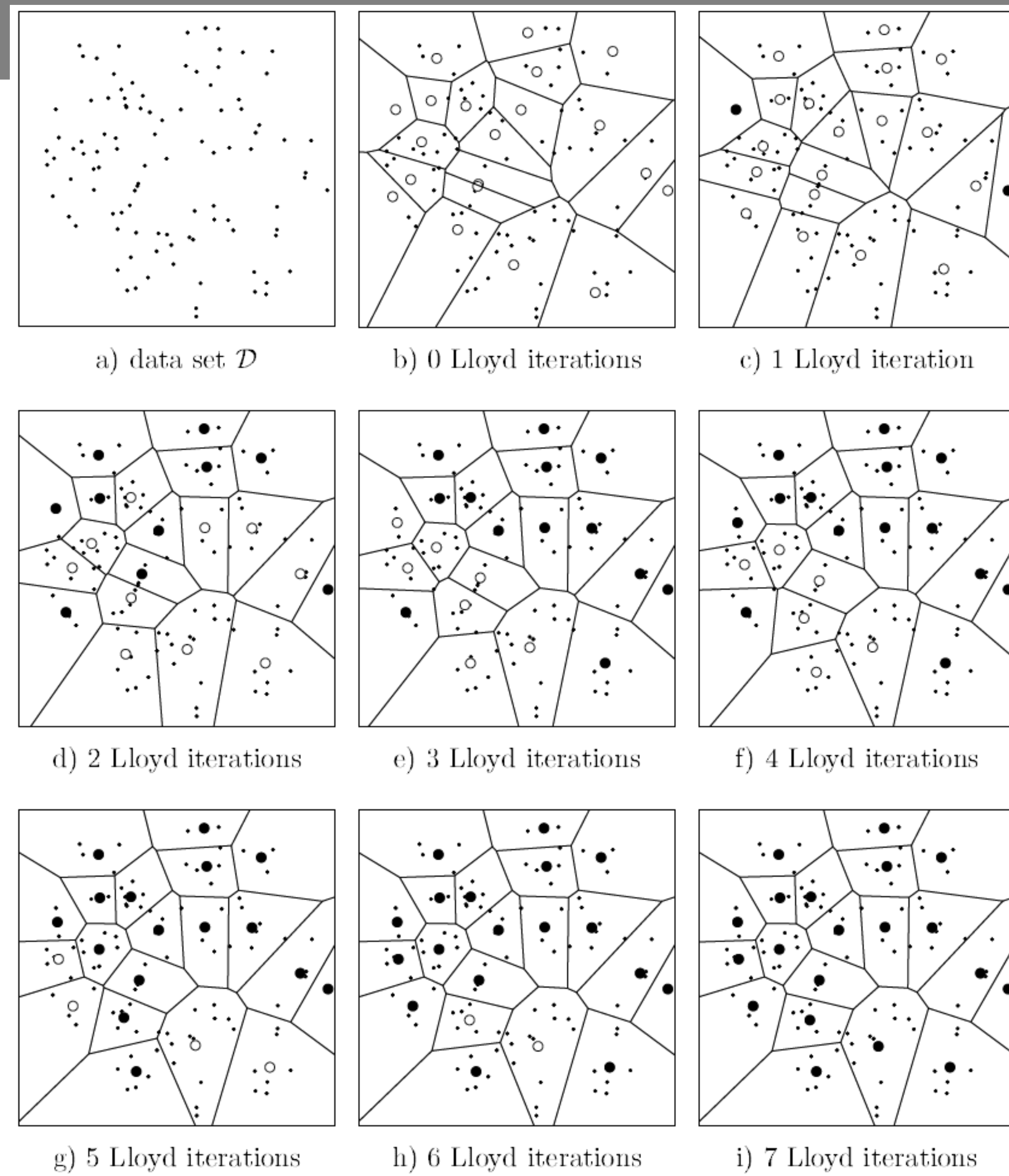
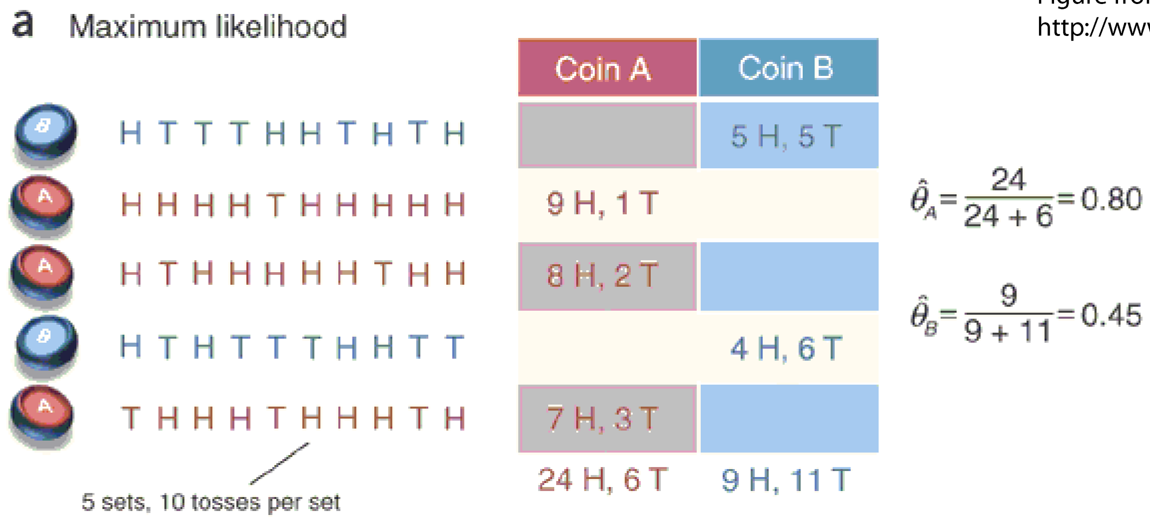


Figure from
http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html



- An experiment with two coins

  At each step, one coin is selected at random and is tossed ten times

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

  Parameters: $\theta = [\theta_A, \theta_B]$ probability of landing on head of A and B, resp.

When it is known which coin has been used at each step, by MLE:

$$\hat{\theta}_A = \frac{N_{A=1}}{N_A} \qquad \hat{\theta}_B = \frac{N_{B=1}}{N_B}$$

- What if Z is *hidden = latent = unobserved?*

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

$$P(Z_1 = A \mid X_1, \theta_A) \propto \binom{10}{5} \cdot 0.60^5 \cdot (1-0.60)^5$$

$$P(Z_1 = B \mid X_1, \theta_B) \propto \binom{10}{5} \cdot 0.50^5 \cdot (1-0.50)^5$$

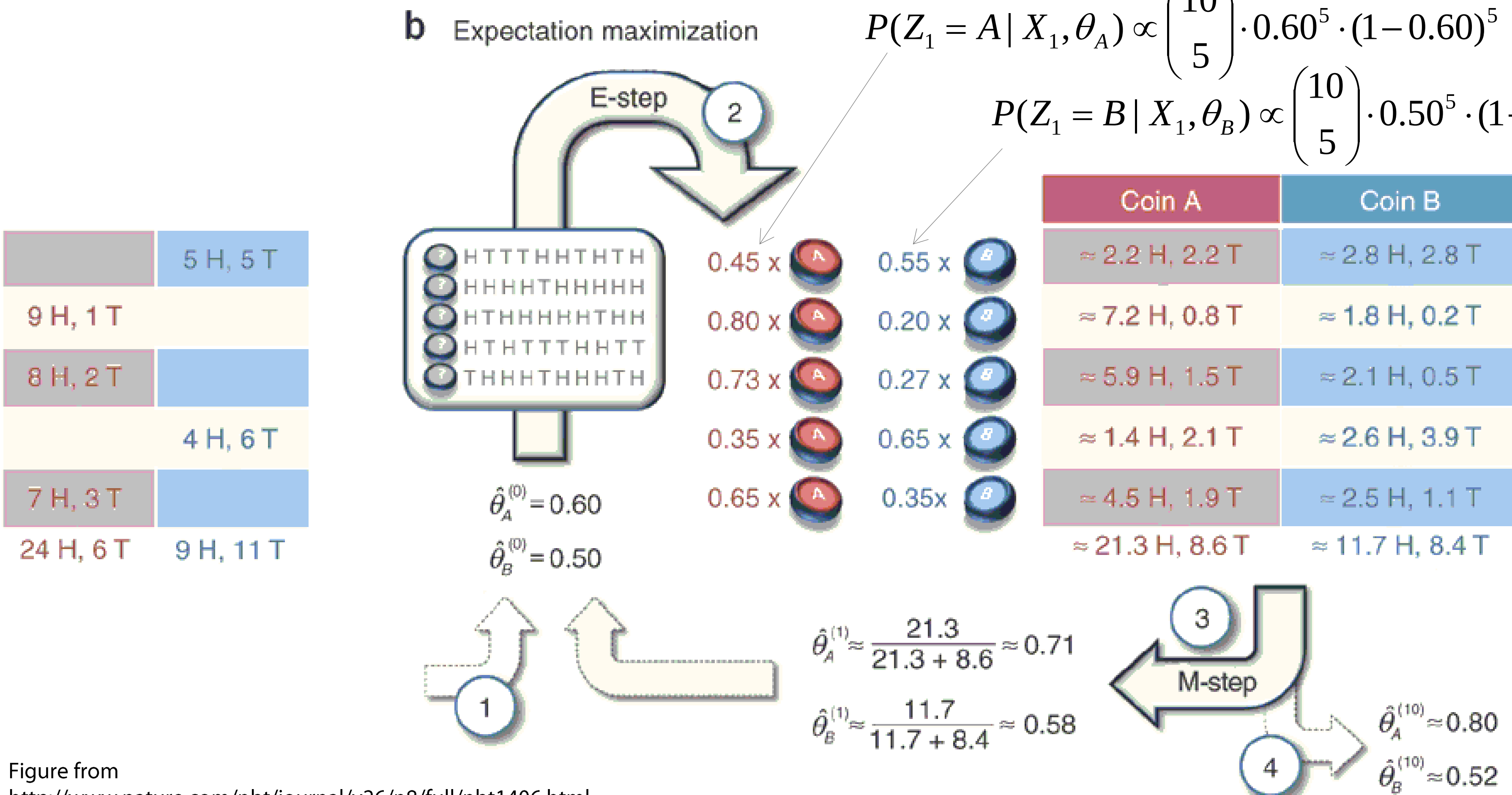
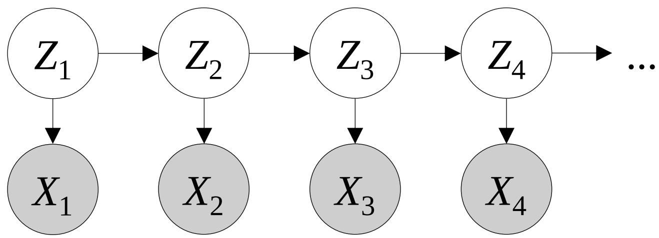


Figure from
http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html

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*

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

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*

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

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

Expected value

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

$$E[f(\{X_i\})] \coloneqq \sum_{\{X_i\}} P(\{X_i\}) \cdot f(\{X_i\})$$

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

Special case:

$$E[\{X_i\}] \coloneqq \sum_{\{X_i\}} P(\{X_i\}) \cdot \{X_i\}$$

the expectation is also an ordered set of values (i.e. some abuse of notation here…)

An aside: Jensen's inequality

A relationship between probability and geometry

When f is *convex function*

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

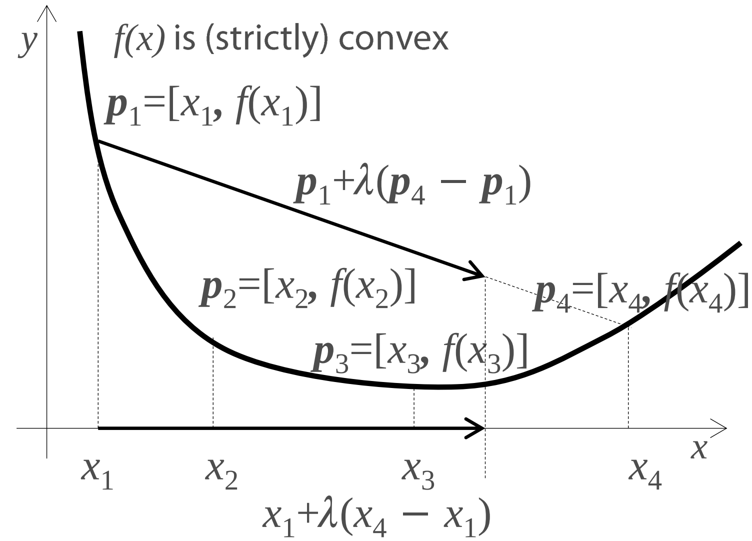
*f is **convex** when for any two points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ the segment ($\boldsymbol{p}_i - \boldsymbol{p}_j$) is not below f*

That is, when

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

*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 <u>constant</u> it is true that

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

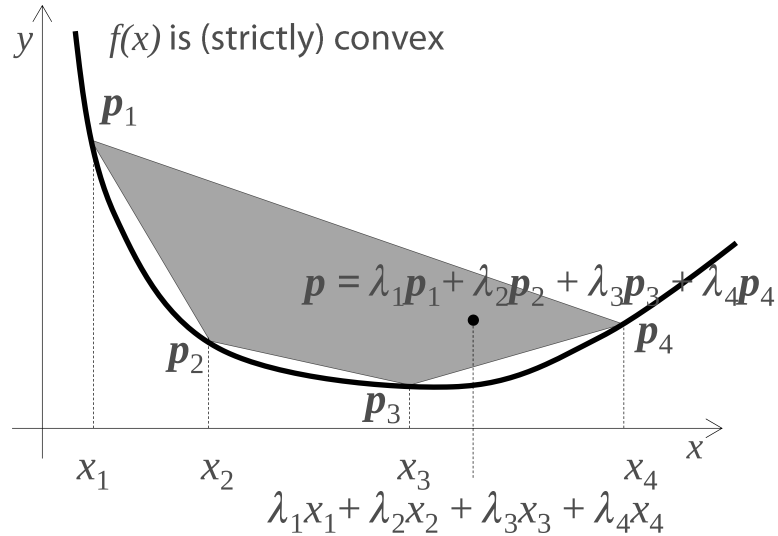
Dual results also hold for *<u>concave</u>* functions

An aside: Jensen's inequality

A relationship between probability and geometry

When f is *convex function*

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



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

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

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

Arbitrary probability distributions

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

Jensen's inequality: log is concave

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

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

Expectation- Maximization (EM) Algorithm

Alternate optimization (coordinate ascent)

Log-likelihood function:

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

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

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

$$Q_m(\{Z_i\}) \coloneqq \frac{P(D_m, \{Z_i\} \mid \theta)}{\sum_{\{Z_i\}} P(D_m, \{Z_i\} \mid \theta)} = \frac{P(D_m, \{Z_i\} \mid \theta)}{P(D_m \mid \theta)} = P(\{Z_i\} \mid D_m, \theta) \eqqcolon p_{\{Z_i\}}$$

These <u>numbers</u> can be computed from the graphical model (i.e. as an <u>inference</u> step)

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

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

$$= \arg\max_\theta \sum_m \left(\sum_{\{Z_i\}} p_{\{Z_i\}} \log P(D_m, \{Z_i\} \mid \theta) - \sum_{\{Z_i\}} p_{\{Z_i\}} \log p_{\{Z_i\}} \right)$$

This is also called the <u>entropy</u> of $Q_m(\{Z_i\})$ *(i.e. a constant measure of the distribution)*

$$= \arg\max_\theta \sum_m \sum_{\{Z_i\}} p_{\{Z_i\}} \log P(D_m, \{Z_i\} \mid \theta)$$

Expectation- Maximization (EM) Algorithm

Alternate optimization (coordinate ascent)

Log-likelihood function and its estimator:

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

Algorithm:

1) Assign the θ at random
2) (*E-step*) Compute the probabilities
$$p_{\{Z_i\}} = Q_m(\{Z_i\}) = P(\{Z_i\} \mid D_m, \theta)$$
3) (*M-step*) Compute a new estimate of θ
$$\theta^* = \arg\max_\theta \sum_m \sum_{\{Z_i\}} p_{\{Z_i\}} \log P(D_m, \{Z_i\} \mid \theta)$$
4) Go back to step 2) until some convergence criterion is met

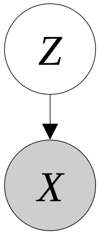
The algorithm converges to a local maximum of the log-likelihood

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

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

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 $\mathbf{R}^d$

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

Multivariate normal distribution

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

i.e. the condition probabilities are normal distributions

The observations are a set $D = \{x_1, x_2, \ldots, x_n\}$ of points in $\mathbf{R}^d$

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

$$p_{mk} = P(Z=k \mid x_m, \phi_k, \mu_k, \Sigma_k) = \phi_k \cdot N(x_m;\mu_k,\Sigma_k)$$

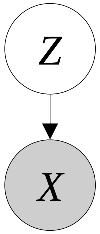
3) (*M-step*) Compute the new estimates for the parameters

$$\phi_k = \frac{1}{n}\sum_m p_{mk}$$

$$\mu_k = \frac{\sum_m p_{mk}x_m}{\sum_m p_{mk}} \qquad \Sigma_k = \frac{\sum_m p_{mk}(x-\mu_k)(x-\mu_k)^T}{\sum_m p_{mk}}$$

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 $\mathbf{R}^d$

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

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

i.e. the condition probabilities are <u>normal</u> distributions

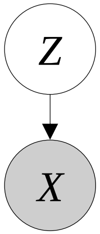
The observations are a set $D = \{x_1, x_2, \dots, x_n\}$ of points in $\mathbf{R}^d$

Proof (of the M-step):

$$\sum_m\sum_k p_{mk}\log P(X_m, Z=k \mid \phi_k,\mu_k,\Sigma_k) = \sum_m\sum_k p_{mk}\log P(X_m \mid Z=k,\mu_k,\Sigma_k)P(Z=k\mid\phi_k)$$

$$= \sum_m\sum_k p_{mk}\left(\log\left((2\pi)^{-d/2}(\det\Sigma_k)^{-1/2}\right) + \left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right) + \log\phi_k\right)$$

EM Algorithm: mixture of Gaussians



Model:

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

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

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

i.e. the condition probabilities are *<u>normal</u>* *distributions*

The observations are a set $D = \{x_1, x_2, \ldots, x_n\}$ *of points in* $\mathbf{R}^d$

Proof (of the M-step):

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

$$=\frac{\partial}{\partial\mu_j}\sum_m\sum_k p_{mk}\left(-\frac{1}{2}(x_m-\mu_k)^T\Sigma_k^{-1}(x_m-\mu_k)\right)=\frac{\partial}{\partial\mu_j}\sum_m\sum_k p_{mk}\left(-\frac{1}{2}(x_m^T\Sigma_k^{-1}x_m+\mu_k^T\Sigma_k^{-1}\mu_k-2+x_m^T\Sigma_k^{-1}\mu_k)\right)$$

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

By imposing: $\sum_m p_{mj}\left(x^T\Sigma_j^{-1}-\mu_j^T\Sigma_j^{-1}\right)=0$

$$\mu_j=\frac{\sum_m p_{mj}x_m}{\sum_m p_{mj}}$$

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

Multinomial distribution

- Bernoulli

  Head or Tail?

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

- Binomial

  n heads out of m coin tosses

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

- Categorical

  The result of throwing a dice with k *faces*

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

- Multinomial

  Obtaining an outcome combination $x_1, \ldots, x_k$ in m throws of a k *–faced dice*, *with* $\sum_{i=1}^{k} x_i = m$

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

Dirichlet distribution

- Beta distribution

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

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

same expression as before, after renaming the parameters…

- Dirichlet distribution

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

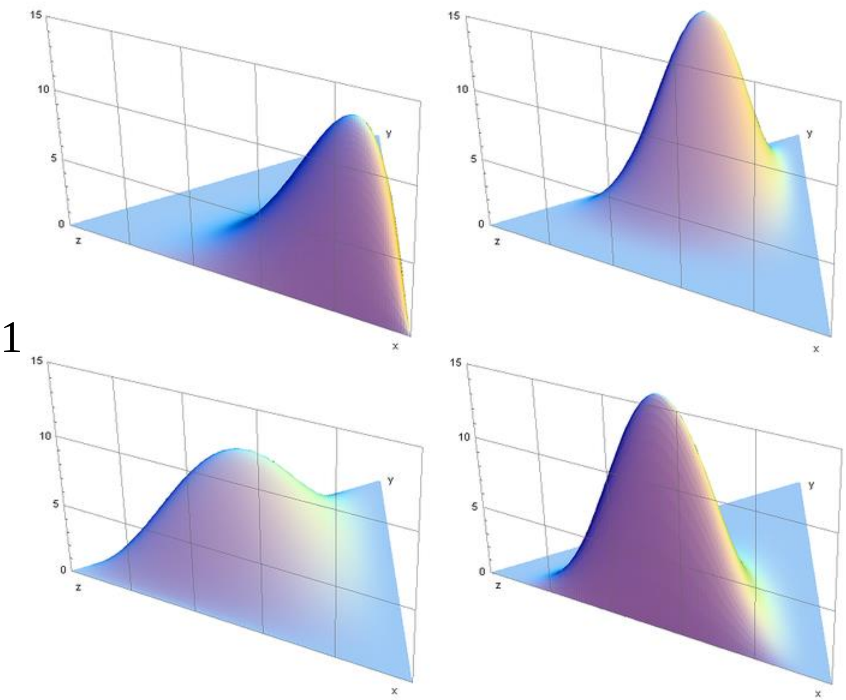
$$\mathrm{D}(x_1, \ldots, x_k; \alpha_1, \ldots, \alpha_k) := \frac{\prod_{i=1}^{k} x_i^{\alpha_i - 1}}{\mathrm{B}(\alpha_1, \ldots, \alpha_k)}, \qquad \sum_{i=1}^{k} x_i = 1$$

where

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

is the *multivariate Beta function*.

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



examples of Dirichlet distributions, for $k = 3$

Dirichlet distribution

- Symmetric Beta distribution

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

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

- Symmetric Dirichlet distribution

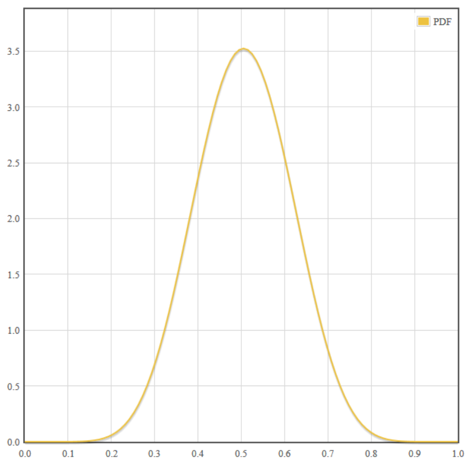
  i.e. when $\alpha_1 = \alpha_2 = \ldots = \alpha_k$

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

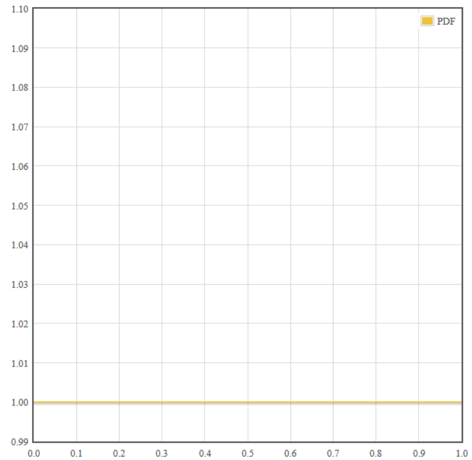
Note: in both distributions, the parameters can be < 1

(this is true of the non-symmetric versions as well)

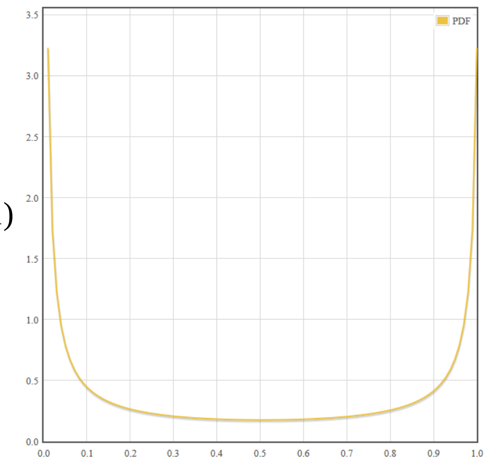
$\mathrm{Beta}(x_1,x_2;10)$



$\mathrm{Beta}(x_1,x_2;1)$

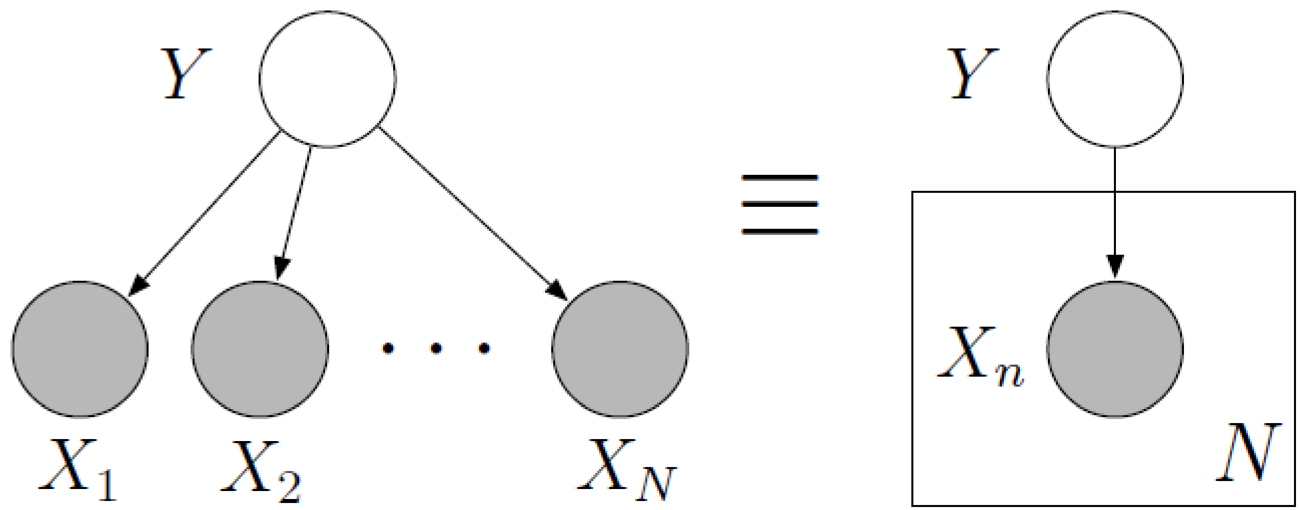


$\mathrm{Beta}(x_1,x_2;0.1)$



An aside: plate notation

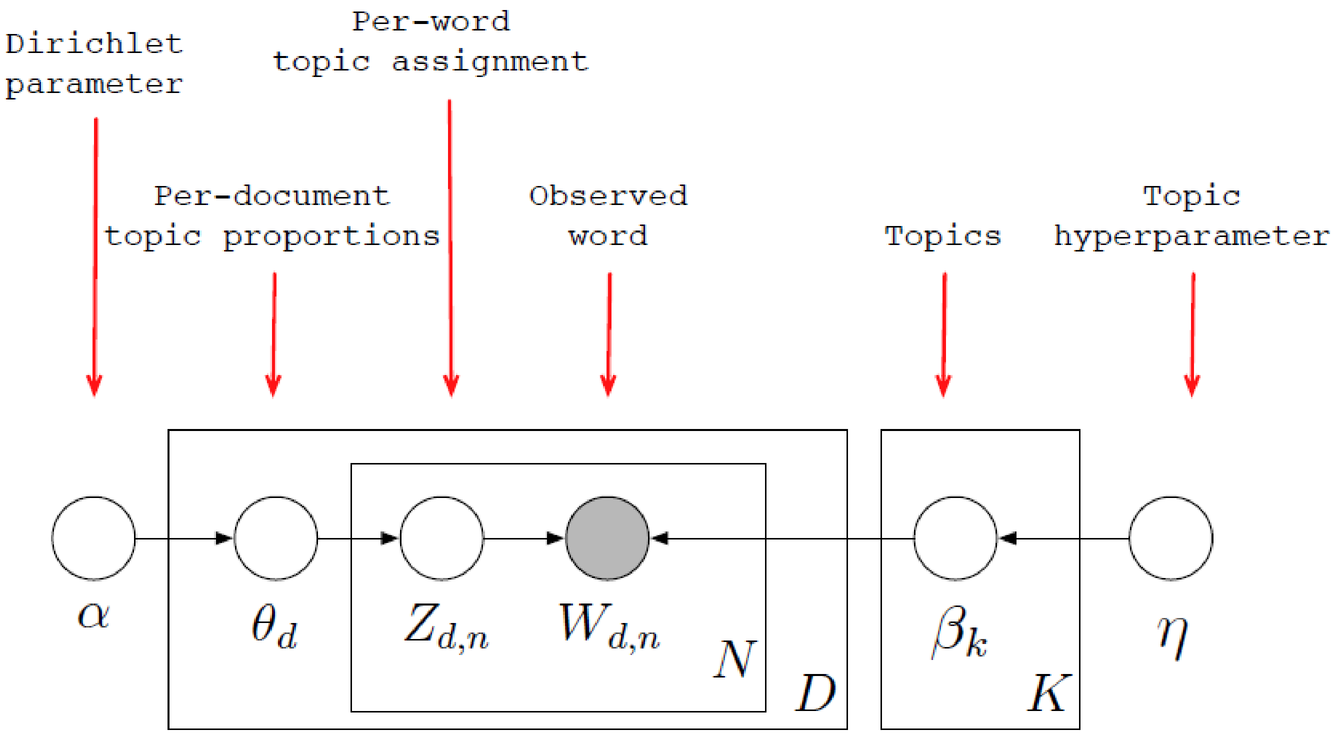
A shorthand notation for graphical models



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

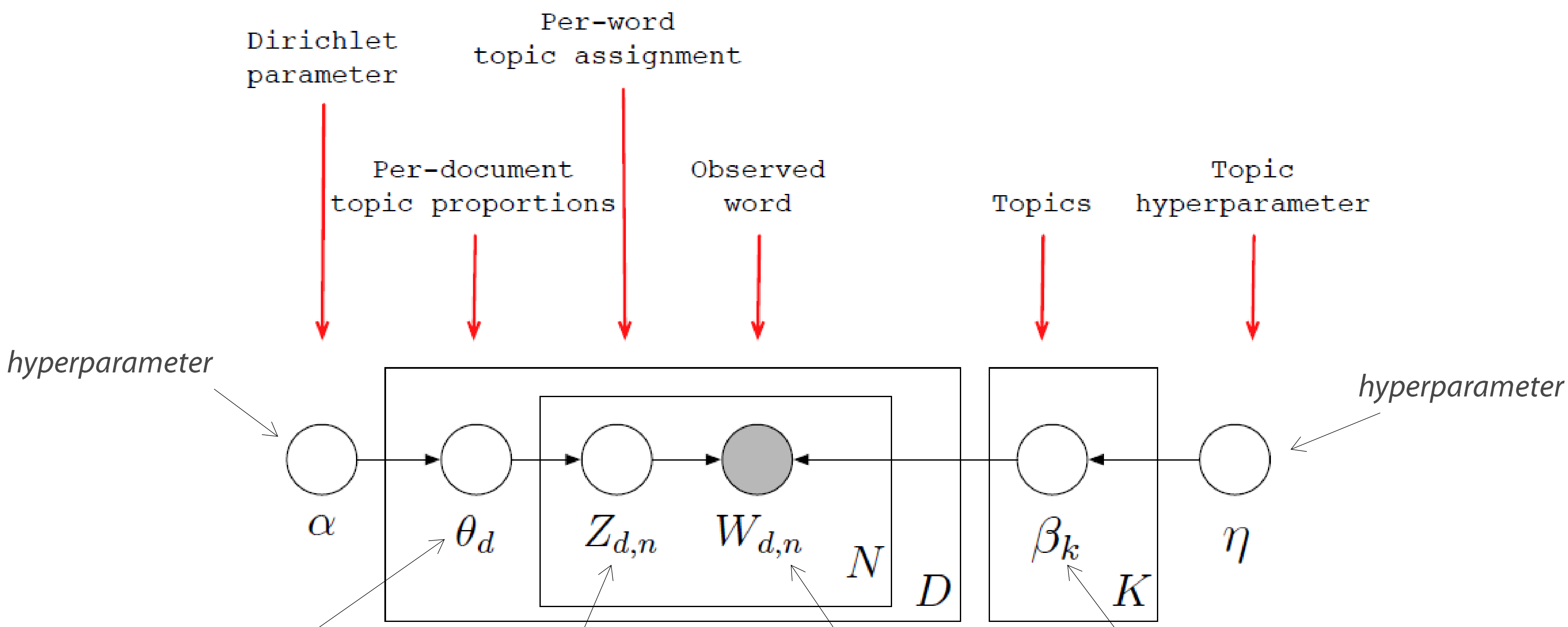
A mixture model:
each document belongs to multiple topics, with different probabilities



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



$\theta_d \in [0, 1]$ per each topic

$Z_{d,n} \in \{1, \dots, k\}$ (*topics*)

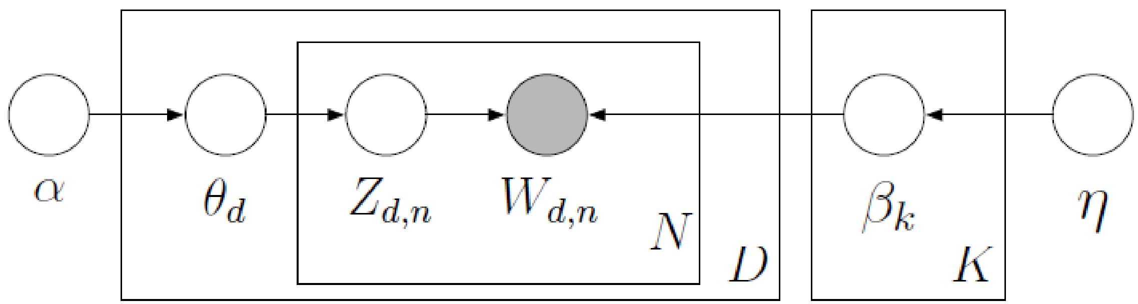
$W_{d,n} \in$ V (*lexicon*)

$\beta_k \in [0, 1]$ per each word in V (*lexicon*)

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 <u>mixture</u> model:
each document belongs to <u>multiple topics</u>, with different probabilities

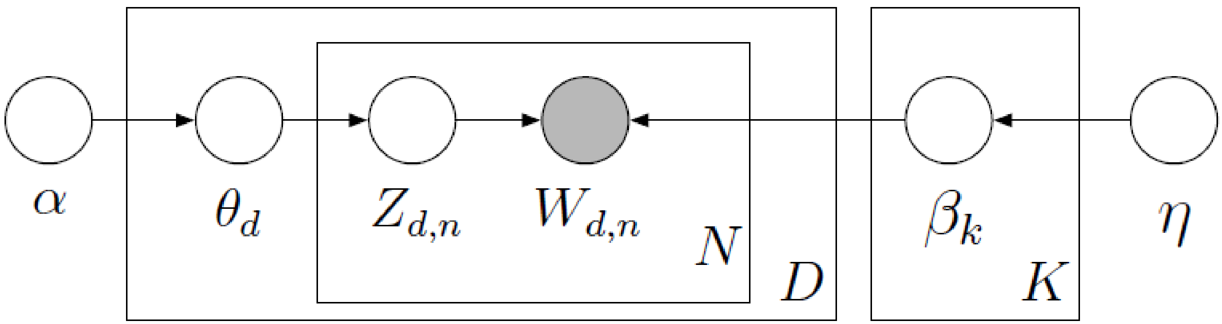


$$\prod_{i=1}^{K} p(\beta_i \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right)$$

Latent Dirichlet Allocation (LDA)

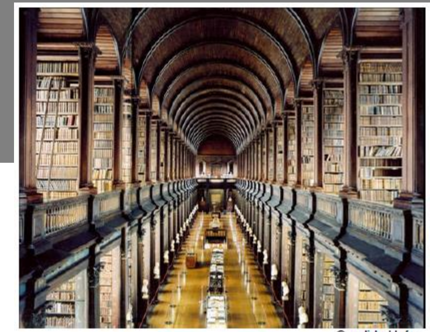
Classifying a corpus of documents with k (unknown) topics
when the only observable variables is the multiple occurrence of words

Generative model: multinomial + Dirichlet



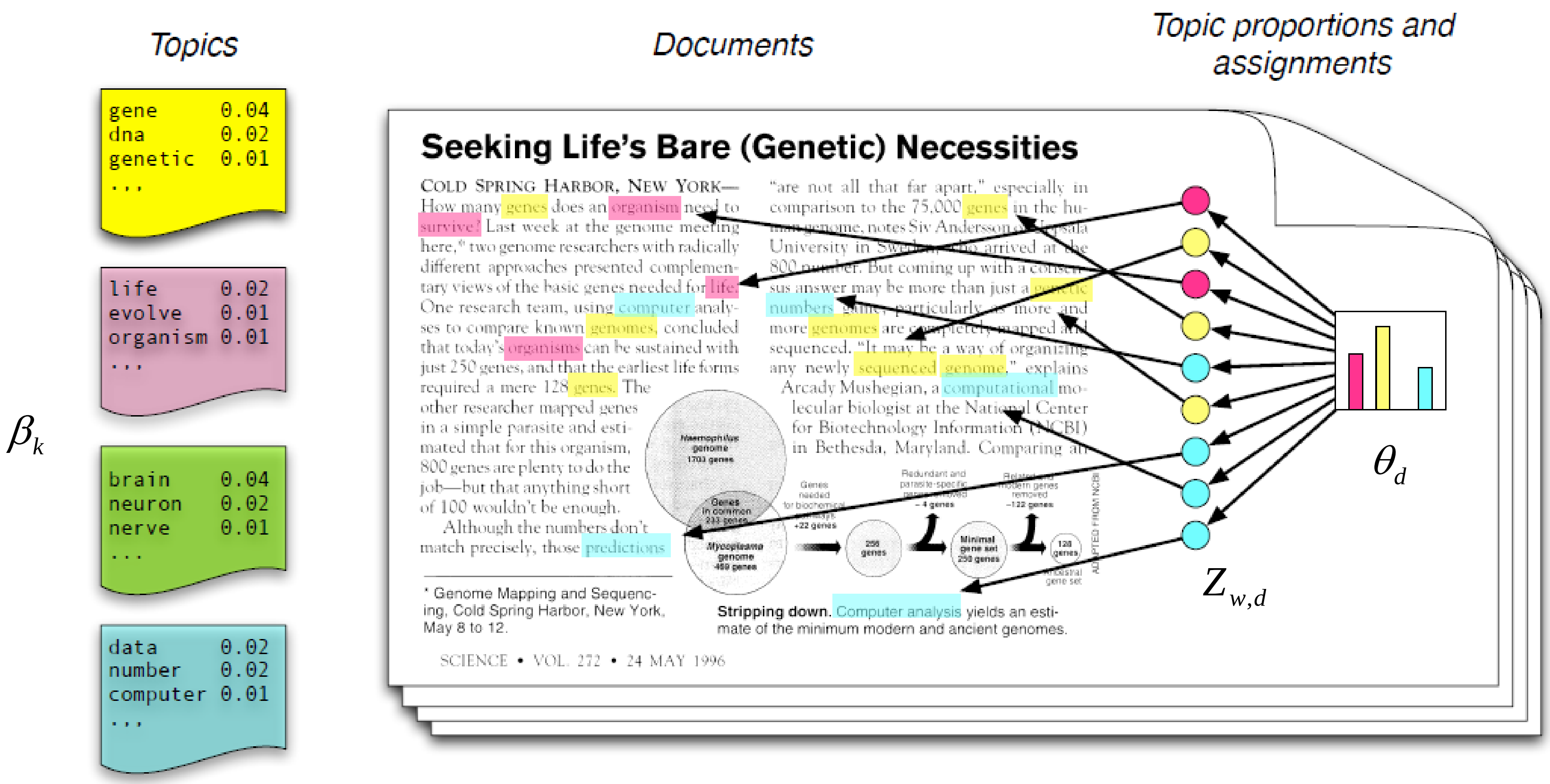
1. Draw each topic $\beta_i \sim \mathrm{Dir}(\eta)$, for $i \in \{1, \dots, K\}$.
2. For each document:
   1. Draw topic proportions $\theta_d \sim \mathrm{Dir}(\alpha)$.
   2. For each word:
      1. Draw $Z_{d,n} \sim \mathrm{Mult}(\theta_d)$.
      2. Draw $W_{d,n} \sim \mathrm{Mult}(\beta_{Z_{d,n}})$.

LDA: what is this for?



Candida Hofer

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



LDA: which results?

Identifying topics:
relative frequencies
of words that define a class

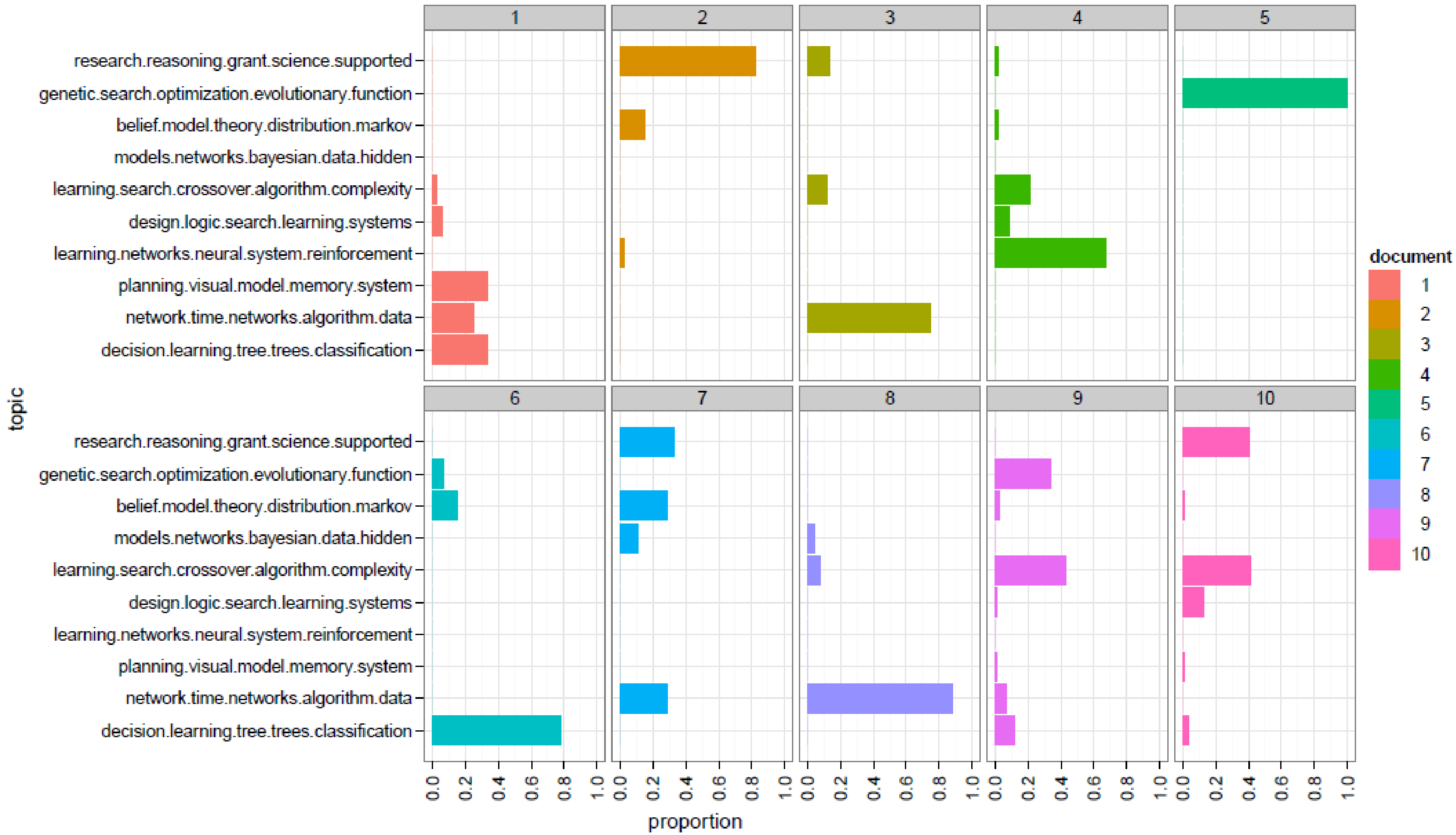
Each box represents a topic

The size of words in a box represents its relative proportion

LDA: which results?

Classifying documents: *relative assignment proportions*

Each topic is represented by a list of most relevant words



LDA: how does it work?

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