Language Information Processing, Advanced

Topic Models

mcuturi@i.kyoto-u.ac.jp

Today's talk

- Continue exploring the representation of **text as histogram of words**.
- Objective: unveil **automatically**
  - topics in large corpora,
  - distribution of topics in each text.
- These techniques are called **topic models**.
- Topic models are related to other algorithms:
  - dictionary learning in computer vision,
  - matrix factorization
- A lot of work in the previous decade
  - Start with a precursor: **Latent Semantic Indexing** ('88)
  - follow with **probabilistic Latent Semantic Indexing** ('99)
  - continue with **Latent Dirichlet Allocation** ('03)
  - and finish with **Pachinko Allocation** ('06).
- This field is still very active... generalizations to non-parametric Bayes
- Chinese Restaurant Process, Indian Buffet Process *etc.*

Reminder: The Naive Bayes Assumption

- From a factorization

$$P(\boldsymbol{C}, \boldsymbol{w_1}, \cdots, \boldsymbol{w_n}) = \prod_{i=1}^{n} P(\boldsymbol{w_i} | \boldsymbol{C}, \boldsymbol{w_1}, \cdots, \boldsymbol{w_{i-1}})$$

which handles all the **conditional** structures of text,

- we assume that each word appears **independently conditionally to** C,

$$\begin{aligned} P(\boldsymbol{w_i} | \boldsymbol{C}, \boldsymbol{w_1}, \cdots, \boldsymbol{w_{i-1}}) &= P(\boldsymbol{w_i} | \boldsymbol{C}, \cancel{\boldsymbol{w_1}, \cdots, \boldsymbol{w_{i-1}}}) \\ &= P(\boldsymbol{w_i} | \boldsymbol{C}) \end{aligned}$$

- and thus

$$P(\boldsymbol{C}, \boldsymbol{w_1}, \cdots, \boldsymbol{w_n}) = \prod_{i=1}^{n} P(\boldsymbol{w_i} | \boldsymbol{C})$$

- The only thing the Bayes classifier considers is **word histogram**

A Few Examples

Science

computer	chemistry	cortex	orbit	infection
methods	synthesis	stimulus	dust	immune
number	oxidation	fig	jupiter	aids
two	reaction	vision	line	infected
principle	product	neuron	system	viral
design	organic	recordings	solar	cells
access	conditions	visual	gas	vaccine
processing	cluster	stimuli	atmospheric	antibodies
advantage	molecule	recorded	mars	hiv
important	studies	motor	field	parasite

FIGURE 1. Five topics from a 50-topic LDA model fit to *Science* from 1980–2002.

Image Source: Topic Models Blei Lafferty (2009)

Yale Law Journal

contractual	employment	female	markets	criminal
expectation	industrial	men	earnings	discretion
gain	local	women	investors	justice
promises	jobs	see	sec	civil
expectations	employees	sexual	research	process
breach	relations	note	structure	federal
enforcing	unfair	employer	managers	see
supra	agreement	discrimination	firm	officer
note	economic	harassment	risk	parole
perform	case	gender	large	inmates

FIGURE 3. Five topics from a 50-topic model fit to the *Yale Law Journal* from 1980–2003.

Image Source: Topic Models Blei Lafferty (2009)

Single Result for Science Article

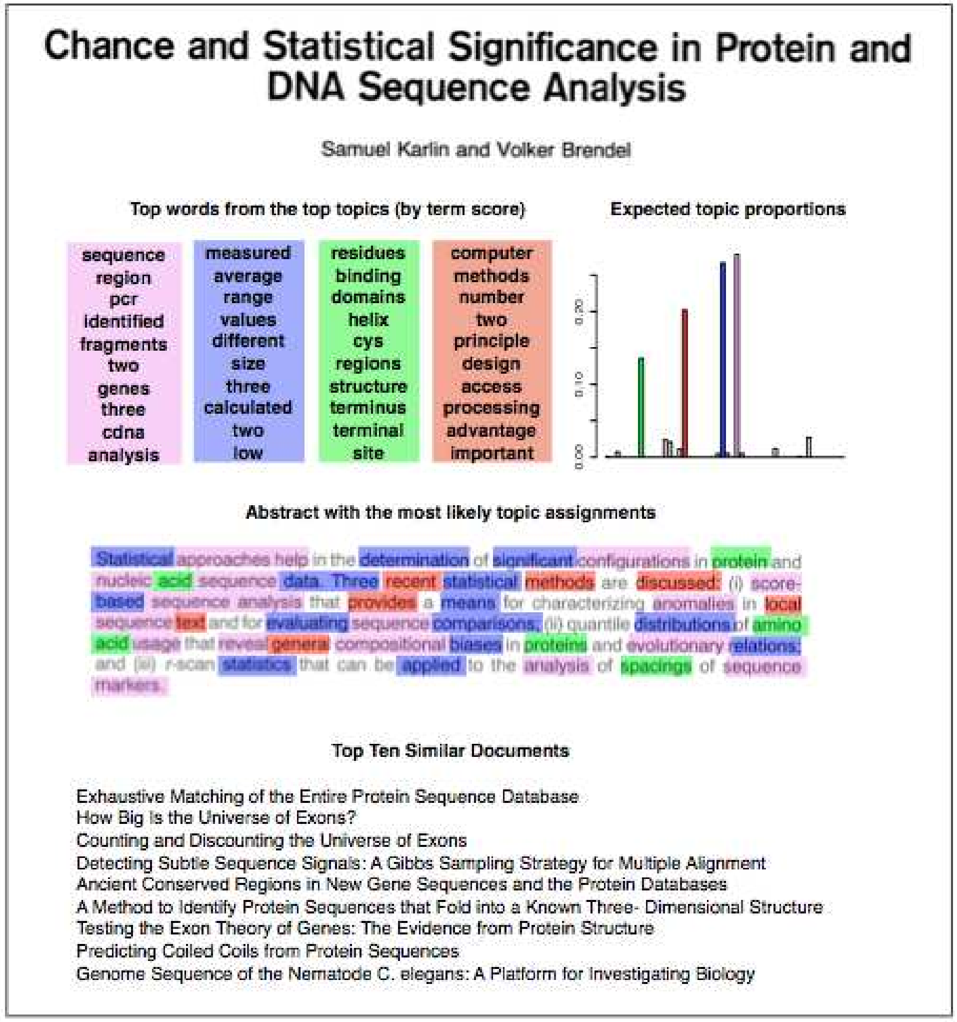
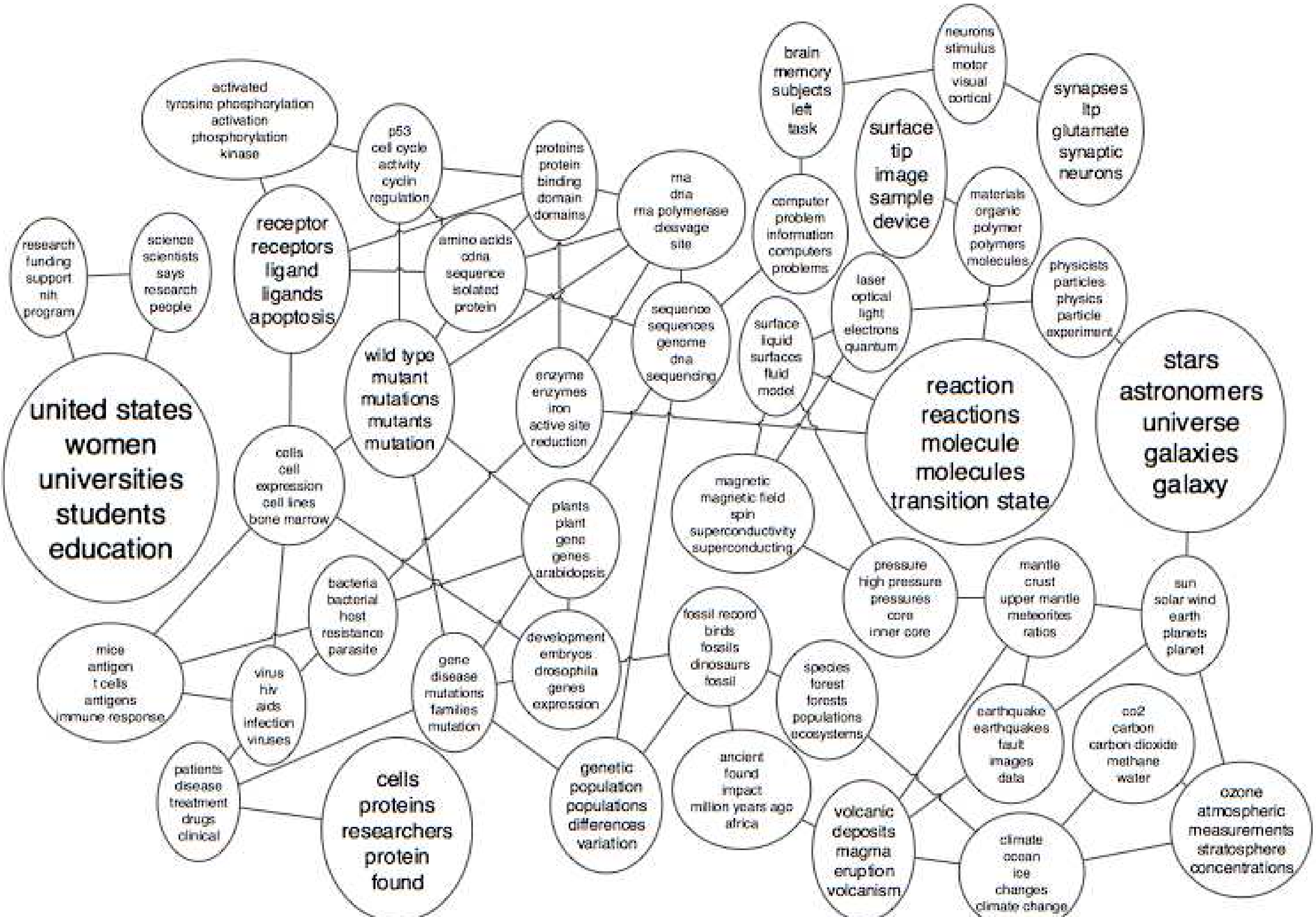


FIGURE 4. The analysis of a document from *Science*. Document similarity was computed using Eq. (4); topic words were computed using Eq. (3).

Topic Graphs



Latent Semantic Indexing

a variation of PCA for normalized word counts...

Latent Semantic Indexing [Deerwester, S., et al, '88]

- Uncover recurring **patterns** in text by considering examples.
- These patterns are **groups of words which tend to appear together**.
- To do so, given a set of n documents, LSI considers a document/word matrix

$$T = \left[\mathrm{tf}_{i,j}\right] \in \mathbb{R}^{m\times n}$$

  where $\mathrm{tf}_{i,j}$ counts the **term-frequency** of word j in text i.

- Using this information, LSI builds a set of influential **groups of words**
- This is similar in spirit to **PCA**:
  - learn **principal components** from data
  - represent each datapoint as the **sum of a few principal components**
  - use the **principal coordinates** for clustering or in supervised tasks.

Renormalizing Frequencies, Preprocesing

Rather than considering only tf_{ij},
introduce a term $x_{ij} = l_{ij} g_i$
which incorporates both **l**ocal and **g**lobal weights

- Local weights ($i.e.$relative to a term i and document j)
  - **binary weight**: $l_{ij} = \delta_{\mathrm{tf}_{ij}>0}$
  - **simple frequency** $l_{ij} = \mathrm{tf}_{ij}$,
  - **hellinger** $l_{ij} = \sqrt{\mathrm{tf}_{ij}}$
  - **log(1+)** $l_{ij} = \log(\mathrm{tf}_{ij} + 1)$
  - **relative to max** $l_{ij} = \frac{\mathrm{tf}_{ij}}{2 \max_i(\mathrm{tf}_{ij})} + \frac{1}{2}$
- Global weights ($i.e.$relative to a term i across **all** documents)
  - **equally weighted documents** $g_i = 1$
  - l_2 **norm of frequencies** $g_i = \frac{1}{\sqrt{\sum_j \mathrm{tf}_{ij}^2}}$
  - $g_i = gf_i / df_i$, where $gf_i = \sum_j \mathrm{tf}_{ij}$, and $df_i = \sum_j \delta_{\mathrm{tf}_{ij}>0}$
  - $g_i = \log_2 \frac{n}{1+df_i}$
  - $g_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$, where $p_{ij} = \frac{\mathrm{tf}_{ij}}{gf_i}$

Word/Document Representation

- typically, one can define

$$X = \left[x_{ij}\right], x_{ij} = \underbrace{\left(1 + \sum_j \frac{p_{ij}\log p_{ij}}{\log n}\right)}_{g_i} \underbrace{\log(\mathrm{tf}_{ij}+1)}_{l_{ij}}$$

- After preprocessing, consider the *normalized* occurrences of words,

$$\begin{array}{cc} & \begin{array}{c} \mathrm{d}_j \\ \downarrow \end{array} \\ \mathrm{t}_i^T \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \end{array}$$

- represents both **term vectors** t_i and **document vectors** d_j
- $\rightarrow$ normalized representation of points (documents) in variables (terms), or vice-versa.

Word/Document Representation

- Each row represents a term, described by its relation to each document:

$$\mathrm{t}_i^T = \begin{bmatrix} x_{i,1} & \dots & x_{i,n} \end{bmatrix}$$

- Each column represents a document, described by its relation to each word:

$$\mathrm{d}_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{m,j} \end{bmatrix}$$

- $\mathrm{t}_i^T \mathrm{t}_{i'}$ is the correlation between terms i, i' over **all** documents.
  - XX^T contains all these dot products.
- $\mathrm{d}_j^T \mathrm{d}_{j'}$ is the correlation between documents j, j' over **all** terms.
  - $X^T X$ contains all these dot products

Singular Value Decomposition

- Consider the **singular value decomposition** (SVD) of X,

$$X = U\Sigma V^T$$

where $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m\times n}$ is diagonal.

- The matrix products highlighting term/documents correlations are

$$\begin{aligned} XX^T &= (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V^{T^T}\Sigma^T U^T) = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T \\ X^T X &= (U\Sigma V^T)^T(U\Sigma V^T) = (V^{T^T}\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T \end{aligned}$$

- U contains the **eigenvectors** of XX^T,
- V contains the **eigenvectors** of $X^T X$.
- Both XX^T and $X^T X$ have the same **non-zero** eigenvalues, given by the non-zero entries of $\Sigma\Sigma^T$.

Singular Value Decomposition

- Let l be the number of non-zero eigenvalue of $\Sigma\Sigma^T$. Then

$$
X = \hat{X}_{(l)} \stackrel{\text{def}}{=} U_{(l)} \quad \Sigma_{(l)} \quad V^T_{(l)}
$$

$$
\begin{array}{cc} & \begin{array}{ccc} & (\mathrm{d}_j) & \\ & \downarrow & \end{array} \\ (\mathrm{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \end{array}
= (\tau_i^T) \rightarrow \begin{bmatrix} \begin{bmatrix} \\ \mathrm{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathrm{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix}\sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l\end{bmatrix} \cdot \begin{array}{c} (\delta_j) \\ \downarrow \\ \begin{bmatrix} [& \mathrm{v}_1 &] \\ & \vdots & \\ [& \mathrm{v}_l &] \end{bmatrix}\end{array}
$$

- $\sigma_1, \dots, \sigma_l$ are the **singular** values,
- $u_1, \dots, u_l$ and $v_1, \dots, v_l$ are the **left and right** singular vectors.
- The only part of U that contributes to t_i is its i'th row, written τ_i.
- The only part of V^T that contributes to d_j is the j'th column, δ_j.

Low Rank Approximations

- A property of the SVD is that for $k \leq l$

$$\hat{X}_k = \underset{X \in \mathbb{R}^{m \times n}, \mathbf{Rank}(X)=k}{\operatorname{argmin}} \|X - X_k\|_F$$

- $\hat{X}_k$ is an approximation of X with **low rank**.
- The term and document vectors can be considered as **concept spaces**
  - the k entries of τ_i provide the occurrence of term i in the k^{th} concept.
  - δ_j^T provides the relation between document j and each concept.

Latent Semantic Indexing Representation of Documents

$\boxed{\text{We can use LSI to}}$

- Quantify the relationship **between documents** j **and** j':
  - compare the vectors $\Sigma_k \delta_j^T$ and $\Sigma_k \hat{\delta}_{j'}$
- Compare **terms** i **and** i' through $\tau_i^T \Sigma_k$ and$\tau_{i'}^T \Sigma_k$,
  - provides a clustering of the terms in the concept space.
- Project a new document onto the concept space,

$$q \to \chi = \Sigma_k^{-1} U_k^T q$$

Probabilistic Latent Semantic Indexing

Latent Variable Probabilistic Modeling

- PLSI adds on LSI by considering a **probabilistic** modeling built upon a **latent** class variable.
- Namely, the joint likelihood that word w appears in document d depends on an

  unobserved variable $z \in \mathcal{Z} = \{z_1, \cdots, z_K\}$

  which defines a joint probability model over $\mathcal{W} \times \mathcal{D}$ (words $\times$ documents) as

$$p(d, w) = P(d)P(w|d), P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)$$

  which thus gives

$$p(d, w) = P(d) \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)$$

  we also have that

$$p(d, w) = \sum_{z \in \mathcal{Z}} P(z)P(w|z)P(d|z)$$

Probabilistic Latent Semantic Indexing

- The different parameters of the probability below

$$p(d,w) = P(d) \sum_{z \in \mathcal{Z}} P(w|z) P(z|d)$$

are all **multinomial** distribution, distributions on the simplex.

$$P(z), P(w|z) P(d|z)$$

- These coefficients can be estimated using maximum likelihood with latent variables.
- Typically using the **Expectation Maximization** algorithm.

Probabilistic Latent Semantic Indexing

- Consider again the formula

$$p(d,w) = \sum_{z \in \mathcal{Z}} P(z)P(w|z)P(d|z)$$

- If we define matrices
  - $U = [P(w_i|z_k)]_{ik}$
  - $V = [P(d_j|z_k)]_{jk}$
  - $\Sigma = \mathbf{diag}(P(z_k))$

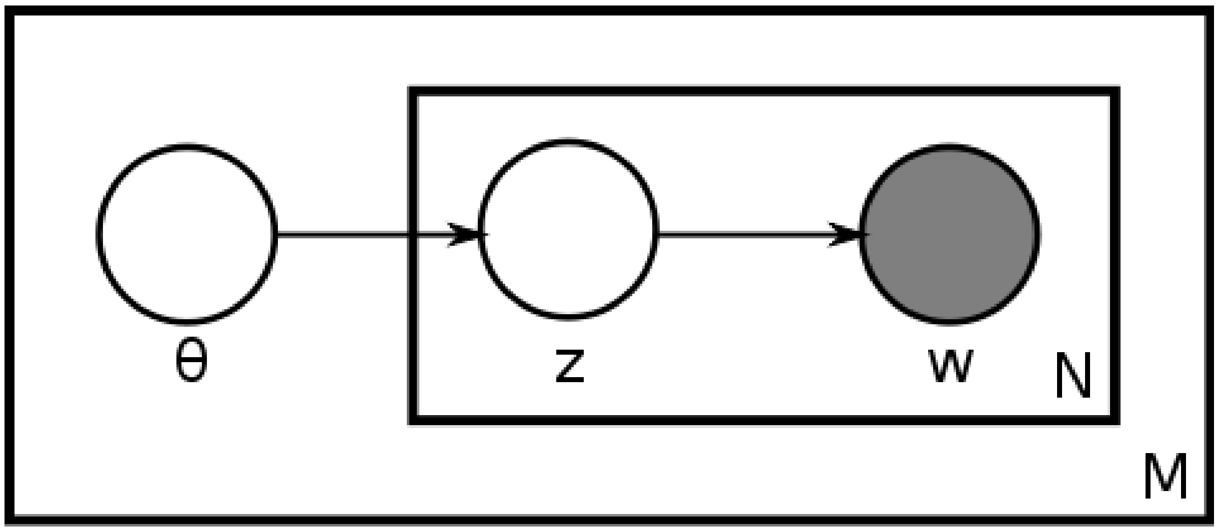
  we obtain that

$$P = [P(w_i, d_j)] = U\Sigma V^T$$

- P and X are the same matrices. We have found a **different factorization** of P (or X).
- **Difference**
  - In LSI, SVD considers the **Frobenius norm** to penalize for discrepancies.
  - in **probabilistic** LSI, we use a different criterion: likelihood function.

Probabilistic Latent Semantic Indexing

- The probabilistic viewpoint provides a **different cost function**
- The probabilistic assumption is explicitated by the following graphical model



- Here θ stands for a document d, M number of documents, N number of words in a document

Image Source: Wikipedia

- The plates stand for the fact that such dependencies are repeated M and N times.

Latent Dirichlet Allocation

Dirichlet Distribution

- Dirichlet Distribution is a distribution on the **canonical simplex**

$$\Sigma_d = \{\mathbf{x} \in \mathbb{R}_+^d \,|\, \sum_{i=1}^d x_i = 1\}$$

- The density is parameterized by a family β of d real **positive** numbers,

$$\beta = (\beta_1, \cdots, \beta_d),$$

has the expression

$$p_\beta(\mathbf{x}) = \frac{1}{\mathrm{B}(\beta)} \prod_{i=1}^d x_i^{\beta_i - 1}$$

with normalizing constant $\mathrm{B}(\beta)$ computed using the Gamma function,

$$\mathrm{B}(\beta) = \frac{\prod_{i=1}^d \Gamma(\beta_i)}{\Gamma\left(\sum_{i=1}^K \beta_i\right)}$$

Dirichlet Distribution

- The Dirichlet distribution is **widely used** to model count histograms
- Here are for instance $\beta = (6, 2, 2), (3, 7, 5), (6, 2, 6), (2, 3, 4)$.

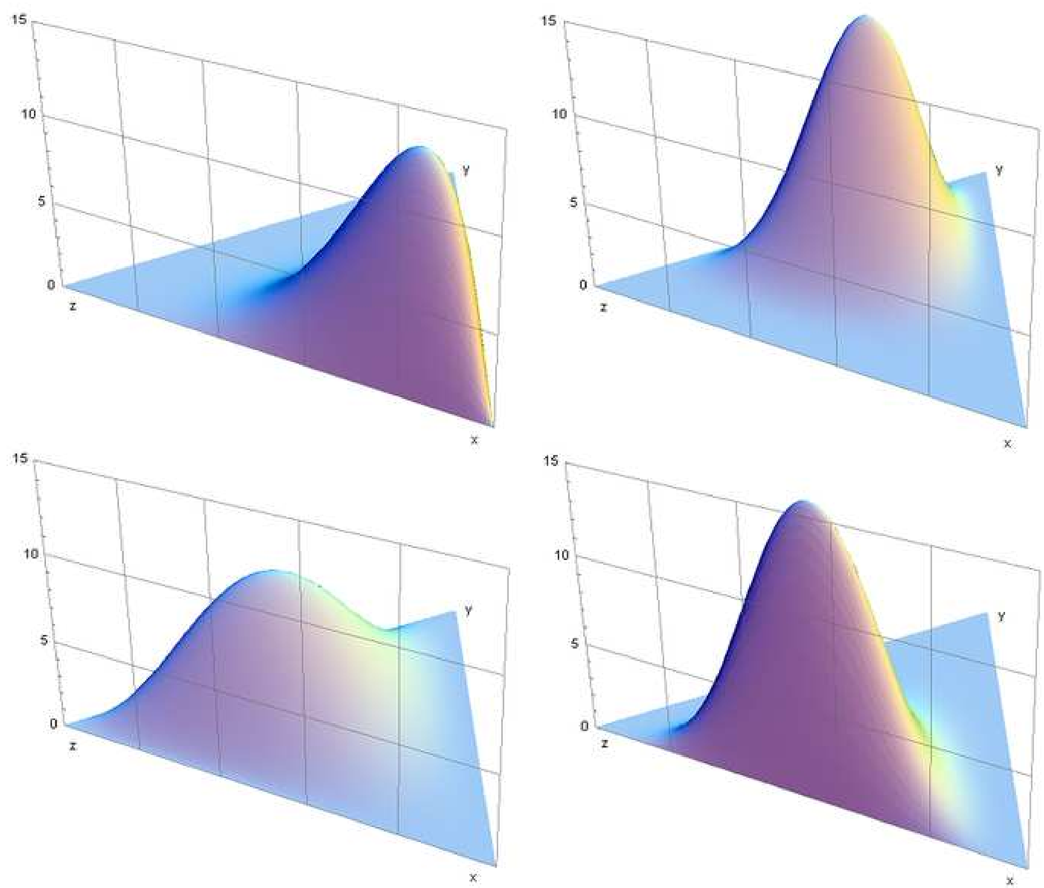


Image Source: Wikipedia

Probabilistic Modeling in Latent Dirichlet Allocation

- LDA assumes that **documents** are **random mixtures over latent topics**,
- **each topic** is characterized by a **distribution over words**.
- **each word** is generated following this distribution.
- Consider K topics,
  - a Dirichlet distribution on topics $\alpha \in \mathbb{R}^K_{++}$ for documents
  - K multinomials on V words described in a Markov matrix (rows sum to 1)

$$\varphi \in \mathbb{R}_+^{K \times V}, \varphi_k \sim \mathrm{Dir}(\beta).$$

Latent Dirichlet Allocation

> Assume that all document $\boldsymbol{d_i} = (\boldsymbol{w_{i1}, \cdots w_{iN_i}})\ j$ has been generated with the following mechanism

- Choose a distribution of topics $\boldsymbol{\theta_i} \sim \mathrm{Dir}(\alpha), j \in \{1, \ldots, M\}$ for document $\boldsymbol{d_i}$.
- For each of the word locations (i, j), where $j \in \{1, \ldots, N_i\}$
    - Choose a **topic** $\boldsymbol{z_{i,j}} \sim \mathrm{Multinomial}(\theta_i)$ at each location j in document $\boldsymbol{d_i}$
    - Choose a **word** $w_{i,j} \sim \mathrm{Multinomial}(\varphi_{z_{i,j}})$.

Latent Dirichlet Allocation

- The graphical model of LDA can be displayed as

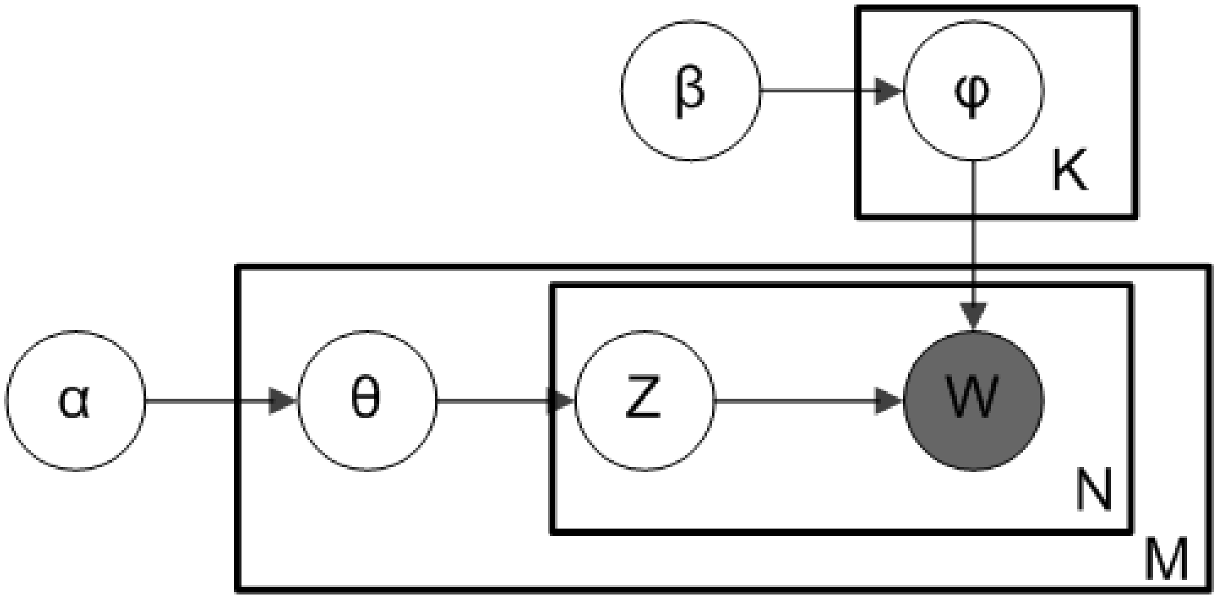


Image Source: Wikipedia

Latent Dirichlet Allocation

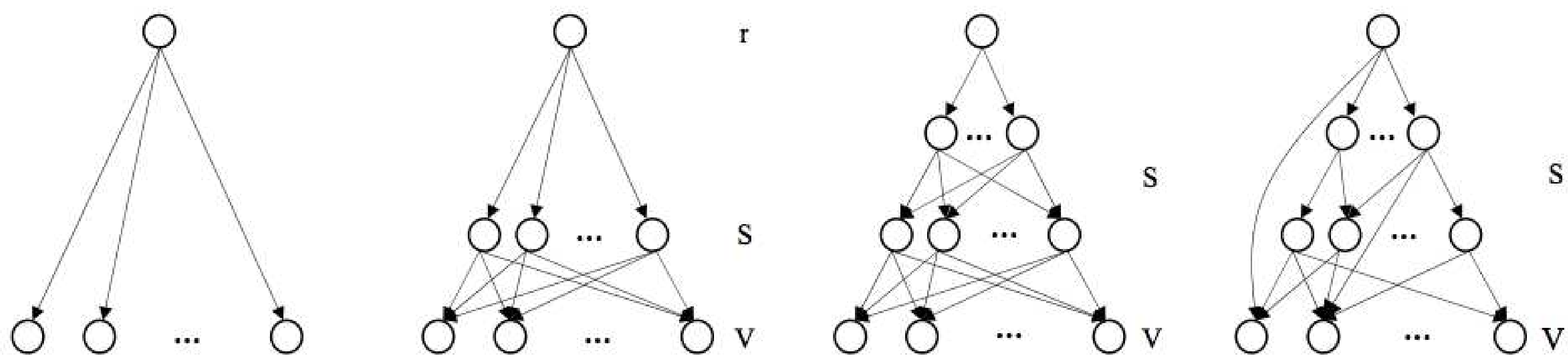
- Inferring now all parameters and latent variables
  - set of K topics,
  - topic mixture $\boldsymbol{\theta_i}$ of each document $\boldsymbol{d_i}$,
  - set of word probabilities for each topic $\boldsymbol{\phi_k}$,
  - topic $\boldsymbol{z_{ij}}$ of each word $\boldsymbol{w_{ij}}$

  is a **Bayesian inference** problem.
- Many different techniques can be used to tackle this issue.
  - See talk from Arnaud Doucet earlier last week..
  - Gibbs sampling
  - Variational Bayes
- This is, in practice, the main challenge to use LDA.

Pachinko Allocation

The idea in one image

- From a simple multinomial (per document) to the Pachinko allocation.



(a) Dirichlet Multinomial (b) LDA (c) Four-Level PAM (d) Arbitrary PAM

Image Source: Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations, Li Mc-Callum

The idea in one image

- Difference with LDA

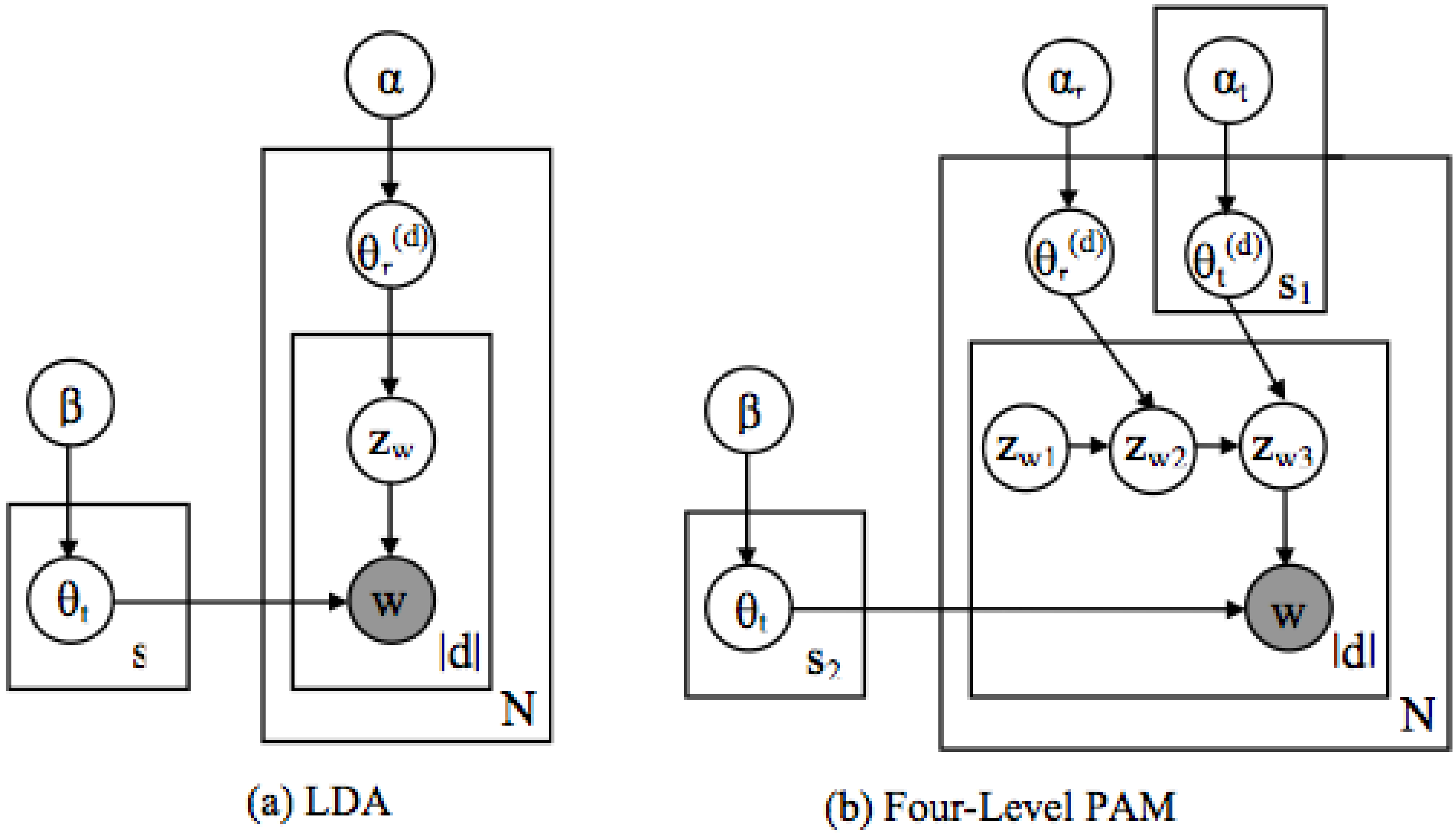


Image Source: Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations, Li Mc-Callum