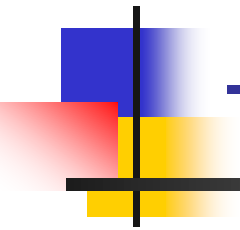


Information retrieval – LSI, pLSI and LDA

Jian-Yun Nie

Basics: Eigenvector, Eigenvalue

- Ref: http://en.wikipedia.org/wiki/Eigenvector
- For a square matrix A:

  $$A\mathbf{x} = \lambda\mathbf{x}$$

  where $\mathbf{x}$ is a vector (eigenvector), and λ a scalar (eigenvalue)
- E.g.

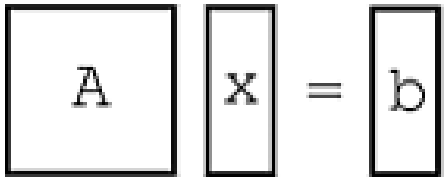
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix},$$

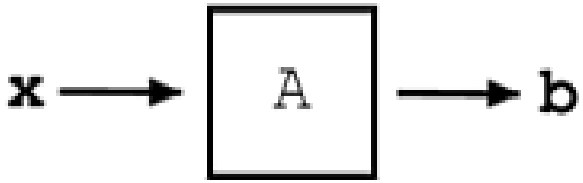
$$\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

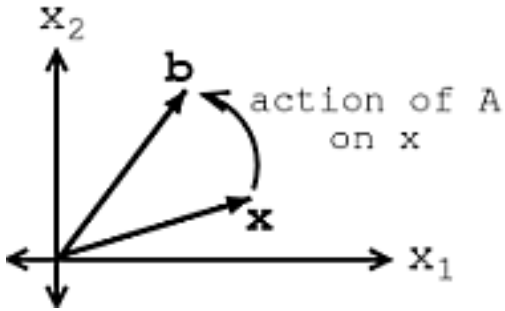
$$\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 4 \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Why using eigenvector?

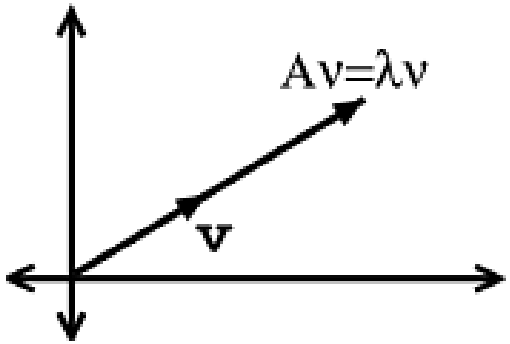
- Linear algebra: $A\,\boldsymbol{x} = b$







- Eigenvector: $A\,\boldsymbol{x} = \lambda\,\boldsymbol{x}$



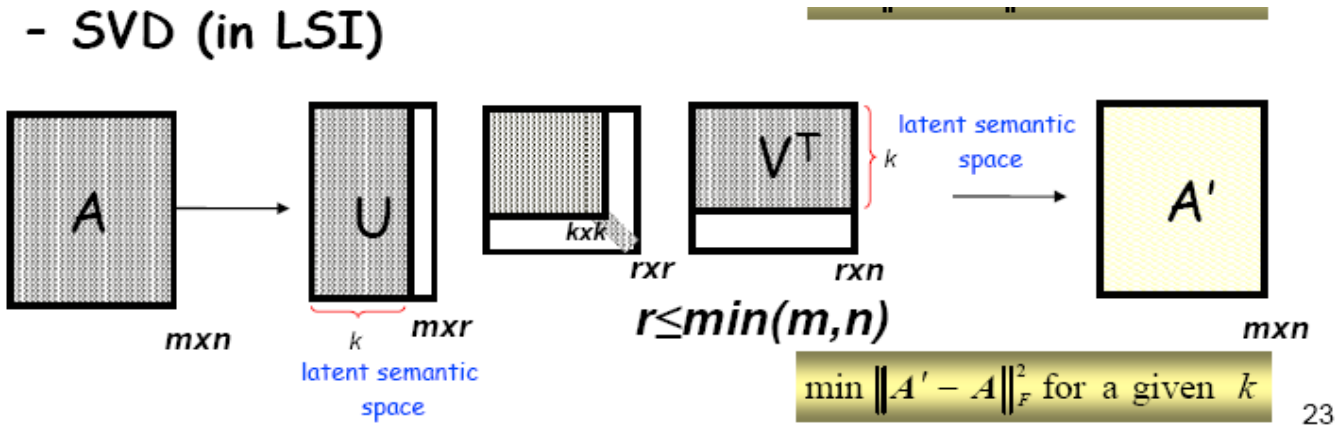
Why using eigenvector

- Eigenvectors are orthogonal (seen as being independent)
- Eigenvector represents the basis of the original vector A
- Useful for
  - Solving linear equations
  - Determine the natural frequency of bridge
  - ...

Latent Semantic Indexing (LSI)

- LSI: a technique projects queries and docs into a space with "latent" semantic dimensions
  - Co-occurring terms are projected onto the same dimensions
  - In the latent semantic space (with fewer dimensions), a query and doc can have high cosine similarity even if they do not share any terms
  - Dimensions of the reduced space correspond to the axes of greatest variation
    - Closely related to Principal Component Analysis (PCA)

Latent Semantic Analysis



LSI

- Singular Value Decomposition (SVD) used for the word-document matrix
  - A least-squares method for dimension reduction

	Term 1	Term 2	Term 3	Term 4
Query	user	interface		
Document 1	user	interface	HCI	interaction
Document 2			HCI	interaction

Classic LSI Example (Deerwester)

Titles

c1:	*Human* machine *interface* for Lab ABC *computer* applications
c2:	A *survey of user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user*-perceived *response time* to error measurement
m1:	The generation of random, binary, unordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors: A survey*

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

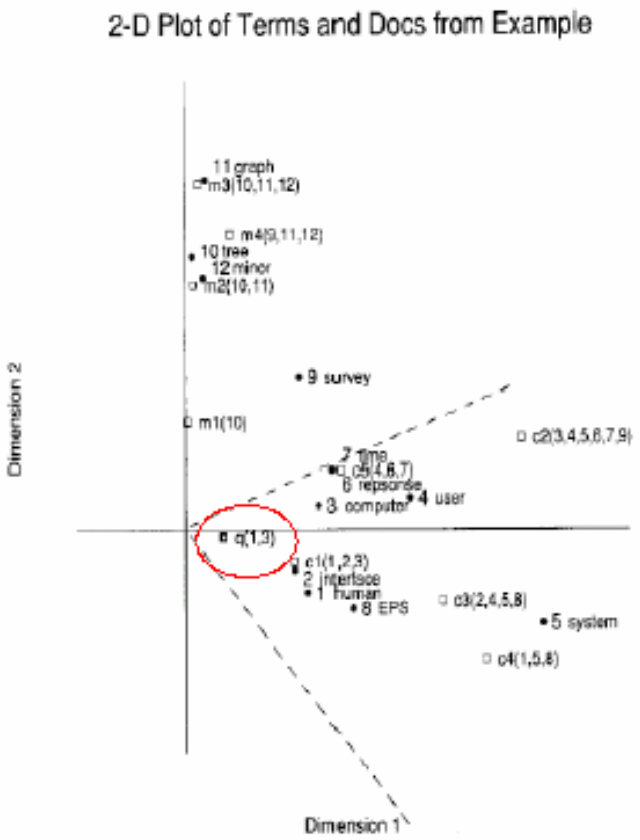


FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the sample TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes are scaled for Document-Document or Term-Term comparisons. The dotted cone represents the region whose points are within a cosine of .9 from the query q. All documents about human-computer (c1–c5) are "near" the query (i.e., within this cone), but none of the graph theory documents (m1–m4) are nearby. In this reduced space, even documents c3 and c5 which share no terms with the query are near it.

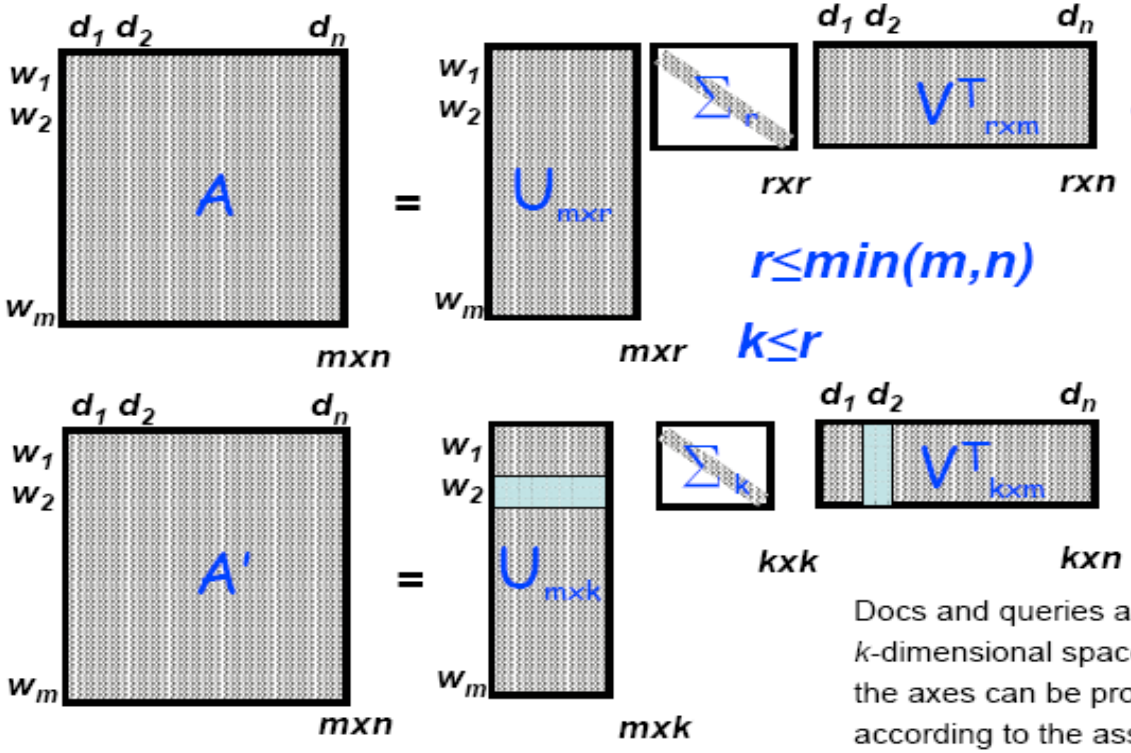
LSI, SVD, & Eigenvectors

- SVD decomposes:
  - Term x Document matrix X as
    - $X=U\Sigma V^T$
      - Where U,V left and right singular vector matrices, and
      - Σ is a diagonal matrix of singular values
    - Corresponds to eigenvector-eigenvalue decompostion: $Y=VLV^T$
      - Where V is orthonormal and L is diagonal
    - U: matrix of eigenvectors of $Y=XX^T$
    - V: matrix of eigenvectors of $Y=X^TX$
    - Σ : diagonal matrix L of eigenvalues

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

SVD: Dimensionality Reduction

- Singular Value Decomposition (SVD)



Both U and V has orthonormal column vectors

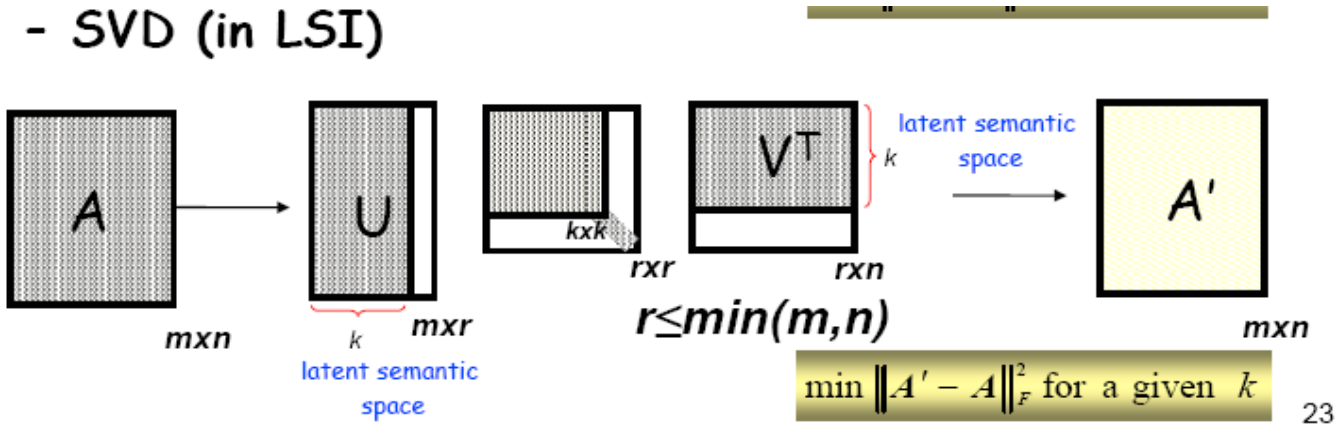
$U^TU=I_{kXk}$

$V^TV=I_{kXk}$

$||A||_F^2 \geq ||A'||_F^2$

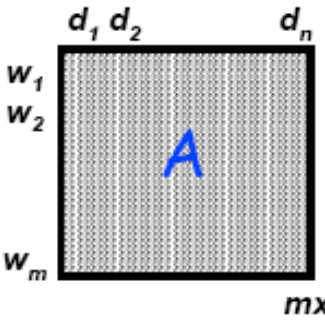
Docs and queries are represented in a *k*-dimensional space. The quantities of the axes can be properly weighted according to the associated diagonal values of Σ_k

Cutting the dimensions with the least singular values



Computing Similarity in LSI

- Fundamental comparisons based on SVD
  - The original word-document matrix (A)



- compare two terms → dot product of two rows of A
  - or an entry in AA^T
- compare two docs → dot product of two columns of A
  - or an entry in A^TA
- compare a term and a doc → each individual entry of A

  - The new word-document matrix (A')

$U'=U_{mxk}$
$\Sigma'=\Sigma_k$
$V'=V_{nxk}$

- compare two terms → dot product of two rows of $U'\Sigma'$
- compare two docs → dot product of two rows of $V'\Sigma'$
- compare a query and a doc → each individual entry of A'

$A'A'^T=(U'\Sigma'V'^T)(U'\Sigma'V'^T)^T=U'\Sigma'V'^TV'\Sigma'^TU'^T=(U'\Sigma')(U'\Sigma')^T$

$A'^TA'=(U'\Sigma'V'^T)^T(U'\Sigma'V'^T)=V'\Sigma'^TU^TU'\Sigma'V'^T=(V'\Sigma')(V'\Sigma')^T$

I

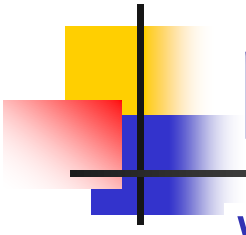
For stretching or shrinking

LSI and PLSI

- LSI: find the k-dimensions that Minimizes the Frobenius norm of A-A'.
  - Frobenius norm of A:

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 = \operatorname{trace}(A^*A) = \sum_{i=1}^{\min\{m,n\}} \sigma_i^2$$

- pLSI: defines one's own objective function to minimize (maximize)



pLSI – a generative model

What is a generative probabilistic model?

Has roughly the following procedure

1. Assume the data we see is generated by some parameterized *random* process.
2. *Learn* the parameters that best explain the data.
3. Use the model to predict *(infer)* new data, based on data seen so far.

Benefits compared to non-generative models

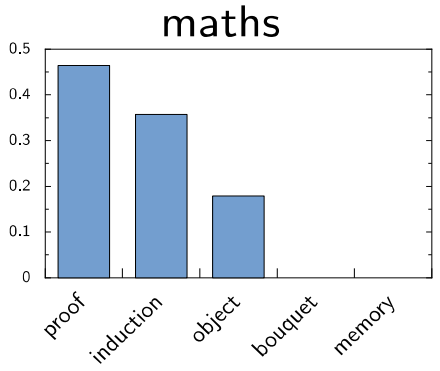
- Assumptions and model are explicit.
- For the inference and learning step we can use well-known algorithms (e.g. EM, Markov Chain Monte Carlo).

pLSI – a probabilistic approach

- Models each word in a document as a sample from a mixture model.

$$p(w) = \sum_{i=1}^{k} p(z_i)p(w|z_i) \qquad \text{s.t.} \quad \sum_{i=1}^{k} p(z_i) \stackrel{!}{=} 1$$

- Introduces the concept of a *topic*.



ed by different topics.

- Problems: (1) *not well-defined* on documents level, (2) overfitting ($kV + kM$ parameters).

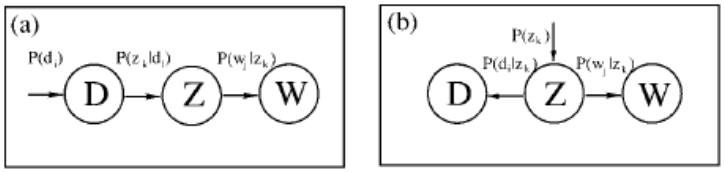
pLSI

- Assume a multinomial distribution

$$P(d_i, w_j) = P(d_i)P(w_j \mid d_i), \quad P(w_j \mid d_i) = \sum_{k=1}^{K} P(w_j \mid z_k)P(z_k \mid d_i).$$

- Distribution of topics (*z*)

$$P(d_i, w_j) = \sum_{k=1}^{K} P(z_k)P(d_i \mid z_k)P(w_j \mid z_k),$$



Question: How to determine *z* ?

Using EM

- Likelihood

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j)\,\log P(d_i, w_j)$$

$$= \sum_{i=1}^{N} n(d_i)\left[\log P(d_i) + \sum_{j=1}^{M}\frac{n(d_i, w_j)}{n(d_i)}\,\log\sum_{k=1}^{K} P(w_j \mid z_k)P(z_k \mid d_i)\right]$$

- E-step

$$P(z_k \mid d_i, w_j) = \frac{P(w_j \mid z_k)P(z_k \mid d_i)}{\sum_{l=1}^{K} P(w_j \mid z_l)P(z_l \mid d_i).}$$

- M-step

$$P(w_j \mid z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)P(z_k \mid d_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i, w_m)P(z_k \mid d_i, w_m)},$$

$$P(z_k \mid d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j)P(z_k \mid d_i, w_j)}{n(d_i)}.$$

Relation with LSI

- Relation

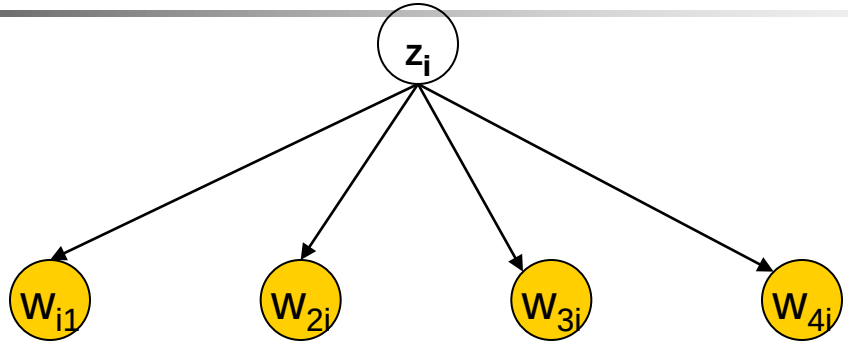
$$\mathbf{P} = \hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^t$$

$$P(d,w) = \sum_{z \in Z} P(z)P(d \mid z)P(w \mid z)$$

$$\hat{\mathbf{U}} = (P(d_i \mid z_k))_{i,k} \qquad \hat{\Sigma} = \mathrm{diag}(P(z_k))_k \qquad \hat{\mathbf{V}} = (P(w_j \mid z_k))_{j,k}$$

- Difference:
  - LSI: minimize Frobenius (L-2) norm ~ additive Gaussian noise assumption on counts
  - pLSI: log-likelihood of training data ~ cross-entropy / KL-divergence

Mixture of Unigrams (traditional)



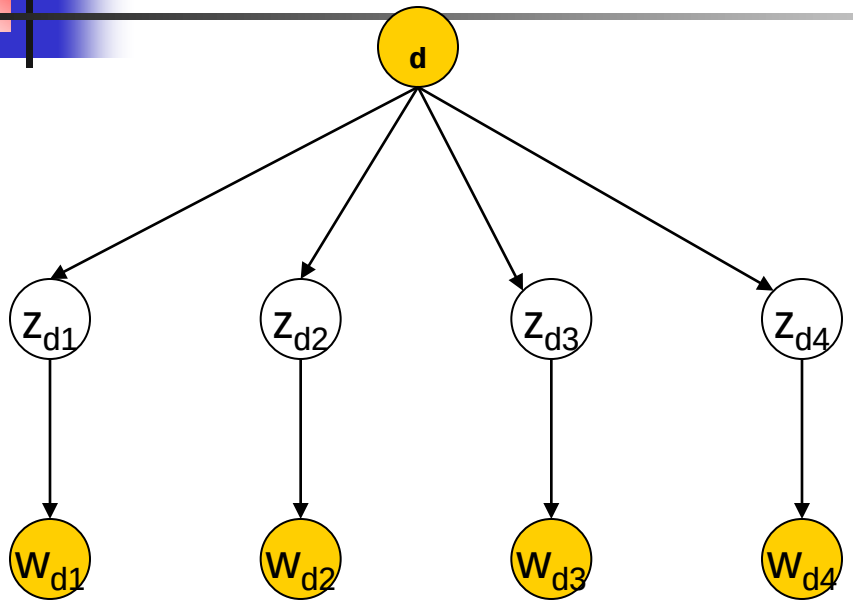
Mixture of Unigrams Model (this is just Naïve Bayes)

For each of M documents,

- Choose a topic z.
- Choose N words by drawing each one independently from a multinomial conditioned on z.

In the Mixture of Unigrams model, we can only have one topic per document!

The pLSI Model

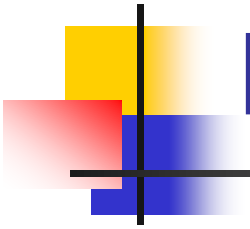


Probabilistic Latent Semantic Indexing (pLSI) Model

For each word of document d in the training set,

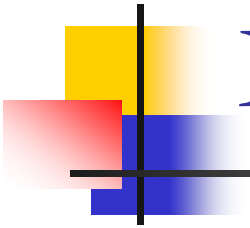
- Choose a topic z according to a multinomial conditioned on the index d.
- Generate the word by drawing from a multinomial conditioned on z.

In pLSI, documents can have multiple topics.



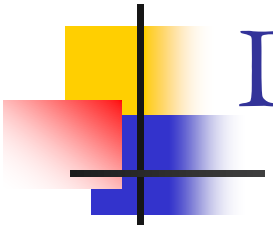
Problem of pLSI

- It is not a proper generative model for document:
  - Document is generated from a mixture of topics
- The number of topics may grow linearly with the size of the corpus
- Difficult to generate a new document



Dirichlet Distributions

- In the LDA model, we would like to say that the *topic mixture proportions* for each document are drawn from some distribution.
- So, we want to put a distribution on multinomials. That is, k-tuples of non-negative numbers that sum to one.
- The space is of all of these multinomials has a nice geometric interpretation as a (k-1)-*simplex*, which is just a generalization of a triangle to (k-1) dimensions.
- Criteria for selecting our prior:
  - It needs to be defined for a (k-1)-simplex.
  - Algebraically speaking, we would like it to play nice with the multinomial distribution.

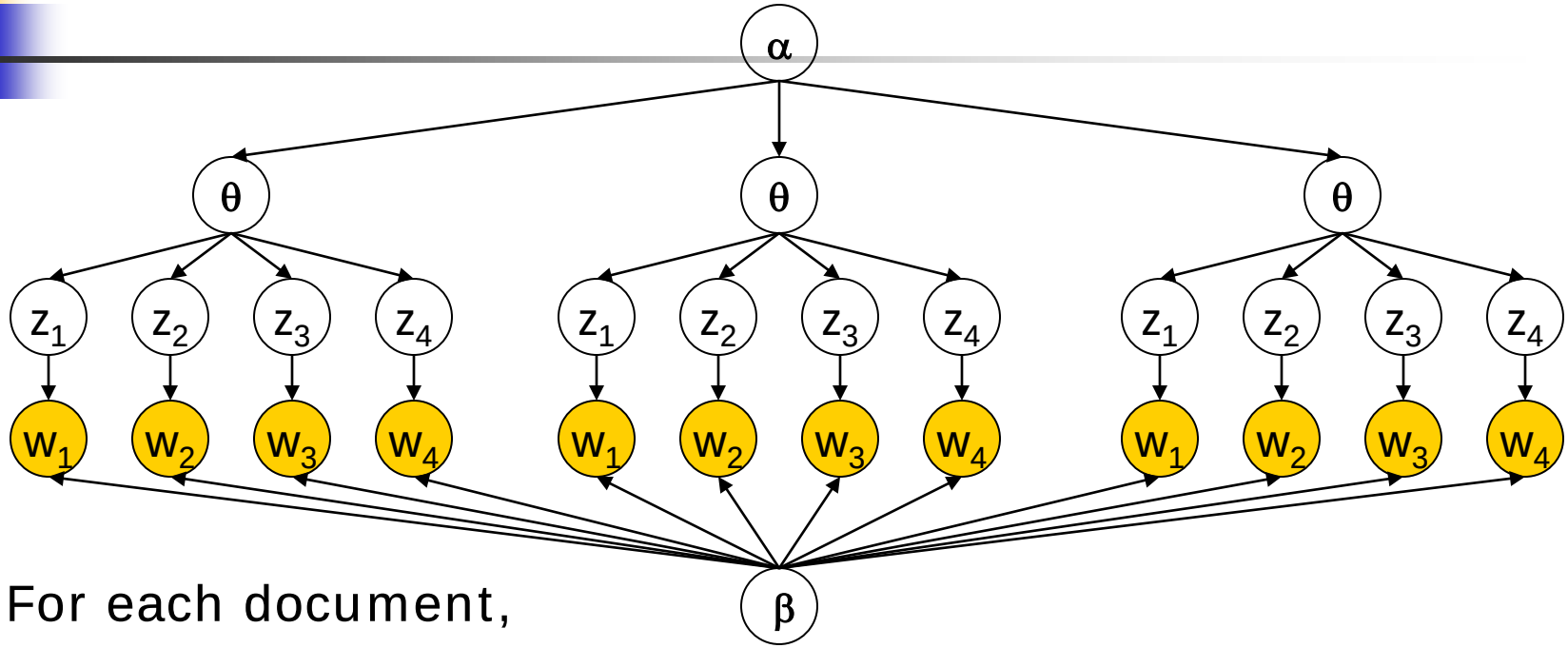


Dirichlet Distributions

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

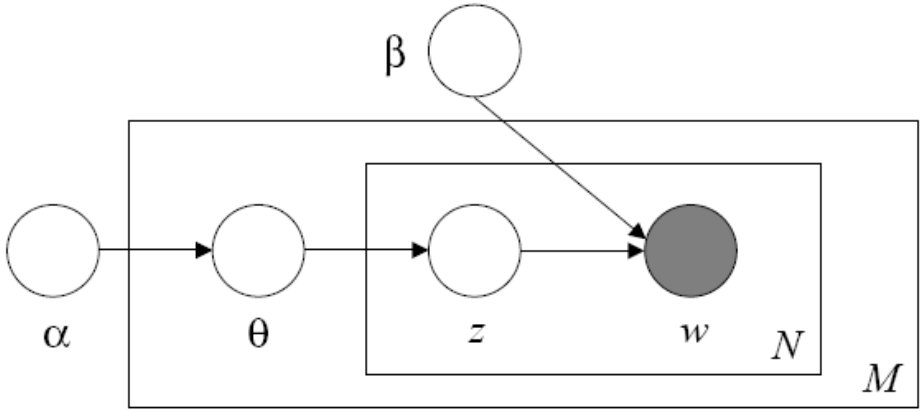
- Useful Facts:
  - This distribution is defined over a (k-1)-simplex. That is, it takes k non-negative arguments which sum to one. Consequently it is a natural distribution to use over multinomial distributions.
  - In fact, the Dirichlet distribution is the conjugate prior to the multinomial distribution. (This means that if our likelihood is multinomial with a Dirichlet prior, then the posterior is also Dirichlet!)
  - The Dirichlet parameter α_i can be thought of as a prior count of the i^{th} class.

The LDA Model



- For each document,
- Choose $\theta \sim \text{Dirichlet}(\alpha)$
- For each of the N words wn:
  - Choose a topic $z_n \text{» Multinomial}(\theta)$
  - Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

The LDA Model

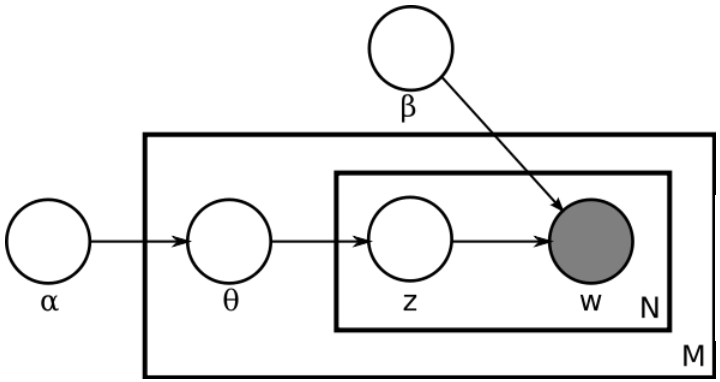


For each document,

- Choose θ» Dirichlet(α)
- For each of the N words w_n:
  - Choose a topic z_n» Multinomial(θ)
  - Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

LDA (Latent Dirichlet Allocation)

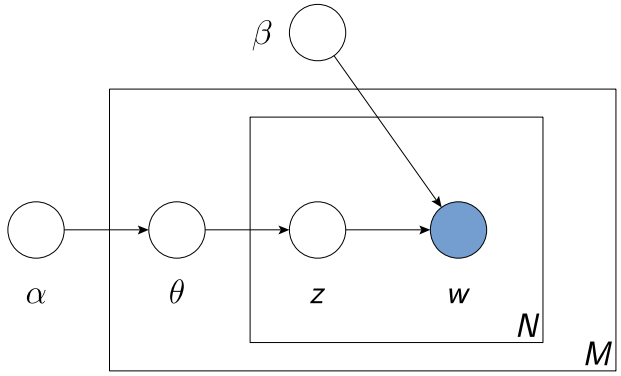
- Document = mixture of topics (as in pLSI), but according to a Dirichlet prior
  - When we use a uniform Dirichlet prior, pLSI=LDA
- A word is also generated according to another variable β: $\beta_{ij} = p(w^j = 1 \mid z^i = 1)$



$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \right) d\theta.$$

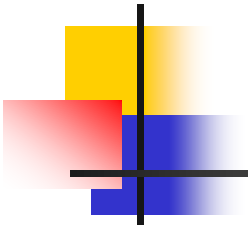
$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d.$$

LDA: The algorithm (2/2)



		0.35	0.2	0	0
comp sc.	0.23	0.17	0.35	0	0.25
wine	0	0	0.1	0.75	0.15

- α is a k-vector. Tells us how much Dirichlet prior scatters around the different topics.



Inference: The problem

To which topics does a given document belong to? Thus want to compute the posterior distribution of the hidden variables given a document:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)}$$

where

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$$

and

$$p(\mathbf{w}|\alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\theta.$$

This not only looks awkward, but is as well *computationally intractable* in general. Coupling between θ and β_{ij}. Solution: *Approximations*.

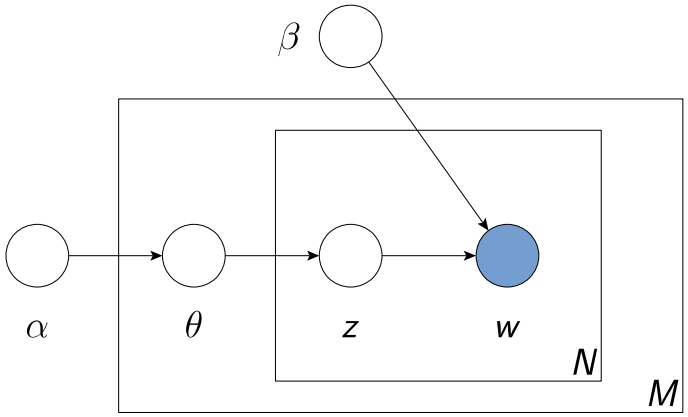
Variational Inference

•In variational inference, we consider a simplified graphical model with variational parameters γ, ϕ and minimize the KL Divergence between the variational and posterior distributions.

$$(\gamma^*, \phi^*) = \arg\min_{(\gamma,\phi)} KL(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta))$$

Variational inference

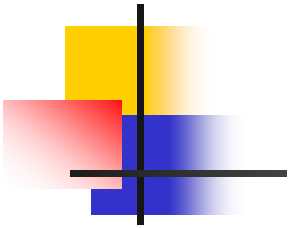
- Replace the graphical model of LDA by a simpler one.



ghtest

lower bound.

- Problematic coupling between θ and β not present in simpler graphical model.



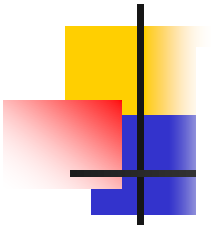
How LDA performs on Reuters data (1/2)

About the experiments

- 100-topic LDA trained on a 16'000 documents corpus of news articles by Reuters (the news agency).
- Some standard stop words removed.

Top words from some of the $p(w|z)$

"Arts"	"Budgets"	"Children"	"Education"
new	million	children	school
film	tax	women	students
show	program	people	schools
music	budget	child	education
movie	billion	years	teachers
play	federal	families	high
musical	year	work	public

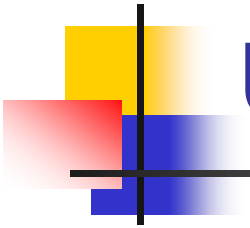


How LDA performs on Reuters data (2/2)

Inference on a held-out document

Again: "Arts", "Budgets", "Children", "Education".

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants.



Use of LDA

- A widely used topic model
- Complexity is an issue
- Use in IR:
  - Interpolate a topic model with traditional LM
  - Improvements over traditional LM,
  - But no improvement over Relevance model (Wei and Croft, SIGIR 06)

References

- **LSI**
  - Deerwester, S., et al, Improving Information Retrieval with Latent Semantic Indexing, Proceedings of the 51st Annual Meeting of the American Society for Information Science 25, 1988, pp. 36–40.
  - Michael W. Berry, Susan T. Dumais and Gavin W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, UT-CS-94-270,1994
- **pLSI**
  - Thomas Hofmann, *Probabilistic Latent Semantic Indexing*, Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), 1999
- **LDA**
  - **Latent Dirichlet allocation.** D. Blei, A. Ng, and M. Jordan. Journal of Machine Learning Research, 3:993-1022, January 2003.
  - **Finding Scientific Topics.** Griffiths, T., & Steyvers, M. (2004). Proceedings of the National Academy of Sciences, 101 (suppl. 1), 5228-5235.
  - **Hierarchical topic models and the nested Chinese restaurant process.** D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum In S. Thrun, L. Saul, and B. Scholkopf, editors, Advances in Neural Information Processing Systems (NIPS) 16, Cambridge, MA, 2004. MIT Press.
- Also see Wikipedia articles on LSI, pLSI and LDA