Probabilistic Latent Semantic Indexing (PLSI)

CS 290D Paper Presentation

Presenter: Sina Miran

Outline

As more information becomes available, it becomes more difficult to find and discover what we need.

We need tools to help us organize, search and understand these vast amount of information.

Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives:

1. Discover the hidden themes in the collection
2. Annotate the documents according to these themes
3. Use annotations to organize, summarize and search

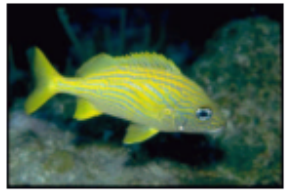
Today, the large collection of data calls for **unsupervised** probabilistic models.

Example Applications:

1. Summarizing Collections of Images

SKY WATER TREE
MOUNTAIN PEOPLE

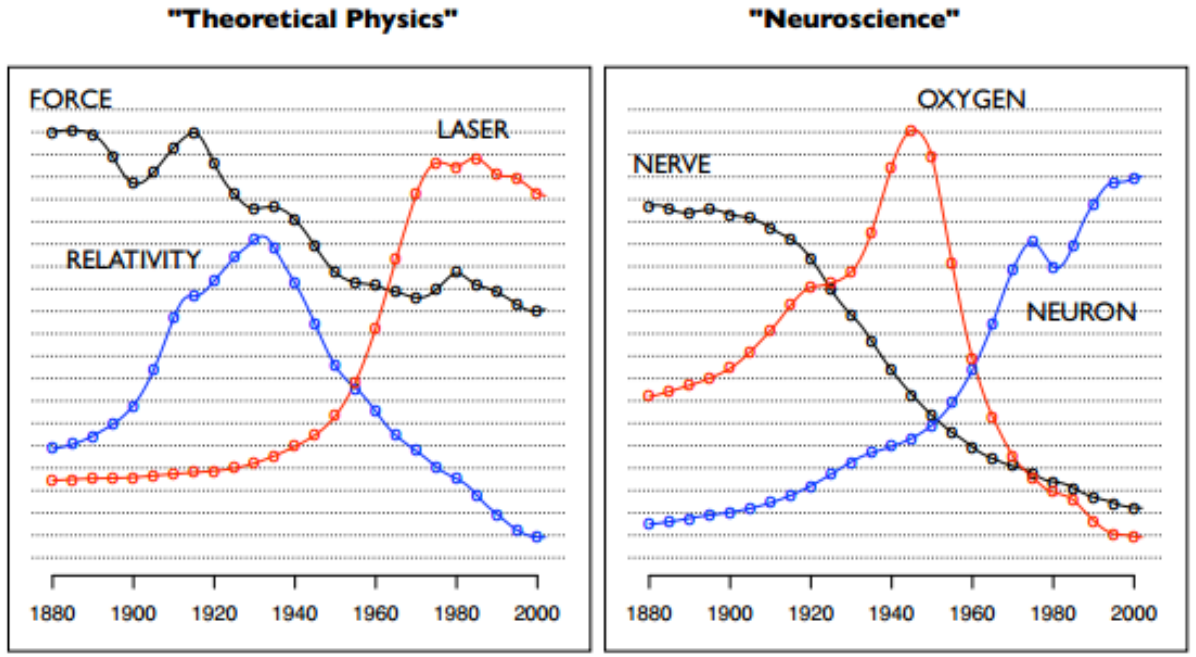
SCOTLAND WATER
FLOWER HILLS TREE



FISH WATER OCEAN
TREE CORAL

PEOPLE MARKET PATTERN
TEXTILE DISPLAY

2. Evolution of Pervasive Topics by Time



Some Assumptions:

- We have a set of documents $d_1, d_2, \dots, d_N$.
- Each document is just a collection of words or a "bag of words". Thus, the order of the words and the grammatical role of the words (subject, object, verbs, …) are **not** considered in the model.
- Words like am/is/are/of/a/the/but/… (stop words) can be eliminated from the documents as a preprocessing step since they don't carry any information about the "topics".
- In fact, we can eliminate words that occur in at least %80 ~ %90 of the documents!

Document-Term Matrix:

Each row represents a document
Each column includes the count of the corresponding term in each of the documents

$$\begin{pmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \dots & w_{tn} \end{pmatrix}$$

	Above	Abrupt	Absent	Absolute	Absorb	Accentuate	Acceptable	Access
Beige Book - National Summary - 1970-05-20	1	0	0	0	0	0	1	0
Beige Book - National Summary - 1970-06-17	1	0	0	0	0	0	0	0
Beige Book - National Summary - 1970-07-15	0	0	0	0	0	0	0	0
Beige Book - National Summary - 1970-08-12	1	0	1	1	0	0	0	0
Beige Book - National Summary - 1970-09-09	0	0	2	0	2	0	0	0
Beige Book - National Summary - 1970-10-14	0	0	1	0	0	0	0	0
Beige Book - National Summary - 1970-11-11	0	0	1	0	0	0	0	0
Beige Book - National Summary - 1970-12-09	1	0	1	0	0	0	1	0

LSI: Perform a low-rank approximation of document-term matrix (typical rank 100-300)

General Idea:

- Map documents (and terms) to a low-dimensional representation (PCA).
- Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space).
- Compute document similarity based on the **inner product** in the **latent semantic space**.

Goals:

- Similar terms and documents map to similar location in low-dimensional space

Singular Value Decomposition (SVD) Review:

$$A = U\Sigma V' \in \mathbb{R}^{n\times m}$$

$$U \in \mathbb{R}^{n\times k} \qquad \Sigma \in \mathbb{R}^{k\times k} \qquad V \in \mathbb{R}^{m\times k}$$

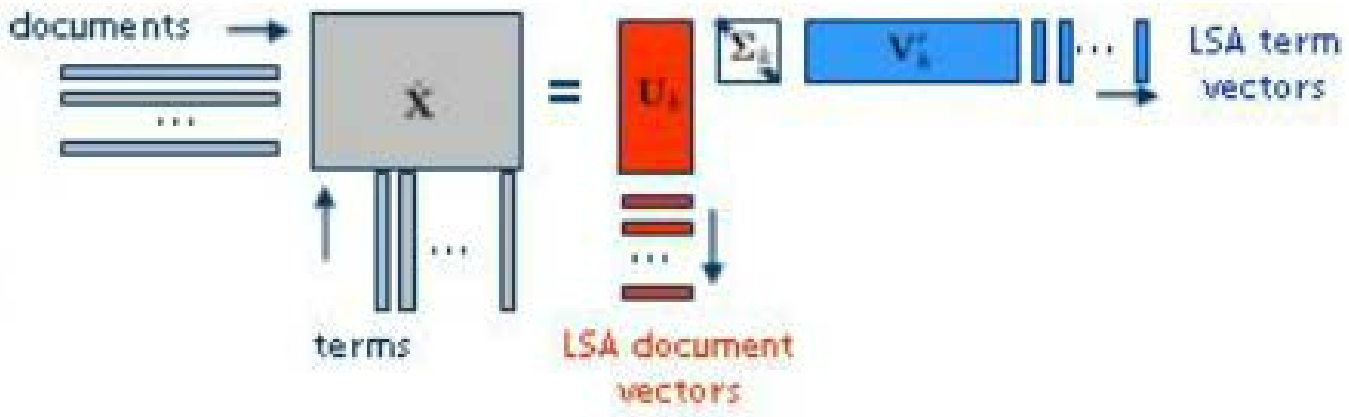
$$U'U = I \qquad V'V = I \qquad \Sigma = diag(\sigma_1, \dots, \sigma_k), \sigma_i \geq \sigma_{i+1} \qquad k = rank(A)$$

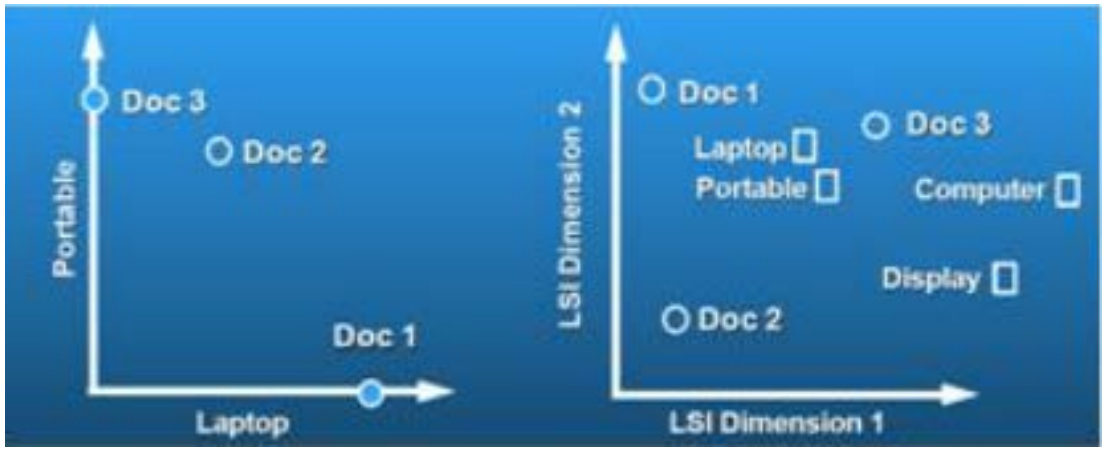
Approximation Problem:

$$X^* = \underset{\hat{X}:rank(\hat{X})=q}{\operatorname{argmin}} \left\|X - \hat{X}\right\|_F, \qquad Ferbenius\ Norm\ \|A\|_F \stackrel{\text{def}}{=} \sqrt{\sum\sum |a_{ij}|^2}$$

$$X^* = Udiag(\sigma_1, \dots, \sigma_q, 0, \dots, 0)V' \qquad X^* = \sum_{r=1}^{q} \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r'$$

Similarity measure: inner products





Vocabulary Mismatch Problem:

- One concept can be represented by several different words!
- Two documents might not contain similar terms (for instance due to writing styles) but refer to a single concept.
- Queries can contain words not present in a document and still be very relevant to that document!

We're somehow looking for $P(a\ word\ or\ a\ query \mid the\ context)$:

$$P(R_d = 1|q) = \frac{P(q|R_d = 1)P(R_d = 1)}{P(q)} \propto \underbrace{P(q|R_d = 1)}_{\substack{Given\ a\ document \\ how\ probable\ is \\ a\ query}} \quad \underbrace{P(R_d = 1)}_{\substack{Uniform\ or \\ relevant\ to\ the \\ popularity\ of \\ the\ document}}$$

$R_d \in \{0,1\}$: relevance of a document

q: a query, set of words

How to calculate $P(q|R_d = 1)$:

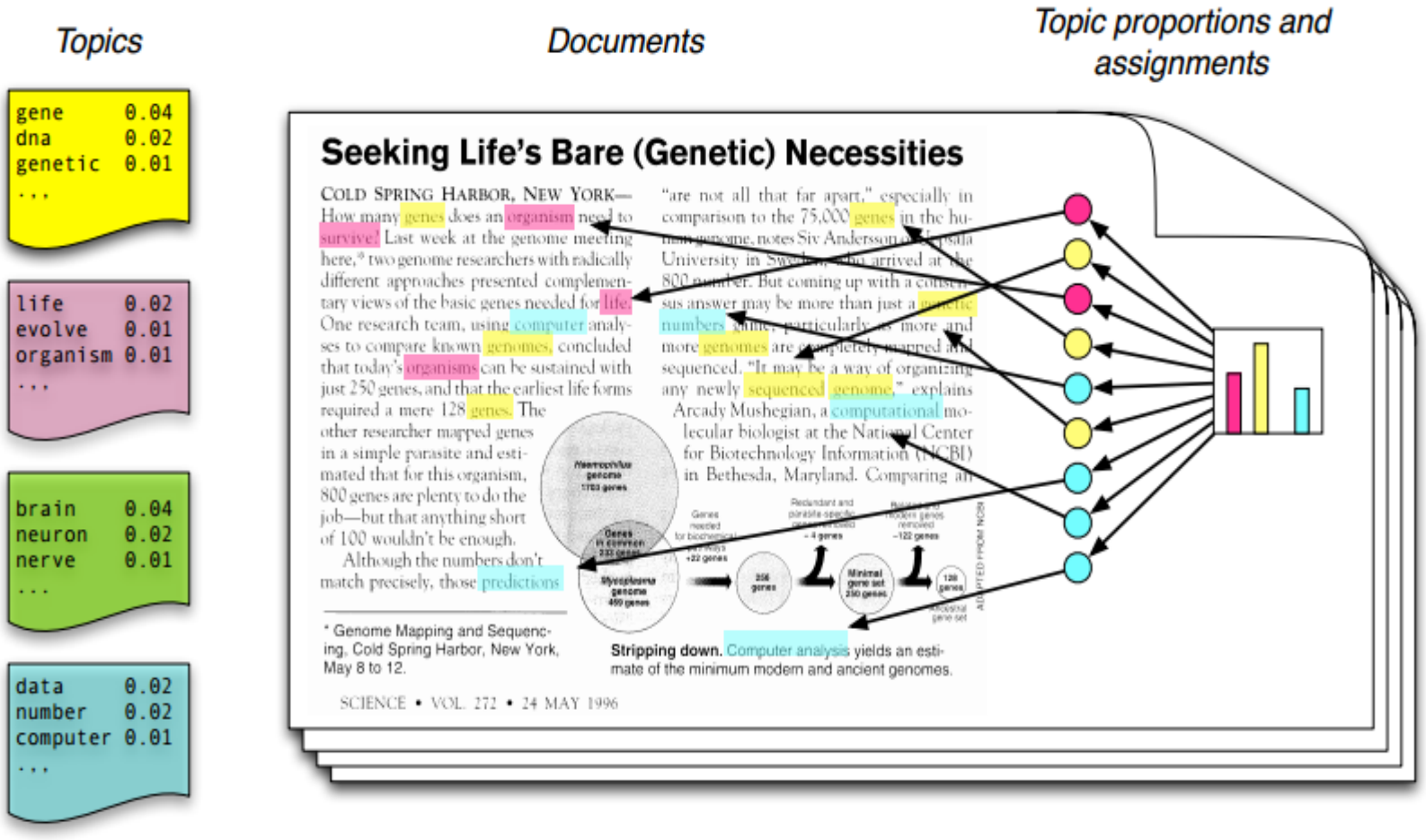
- For each document calculate the probability of each word w coming from (or being relevant to) that document i.e. $P(w|R_d = 1)$
- Calculate the conditional probability of the words in q

PLSI Model Elements:

- A set of documents $\{d_1, \dots, d_N\}$
- A set of concepts, classes or topics $\{z_1, \dots, z_K\}$
- A set of words $\{w_1, \dots, w_M\}$

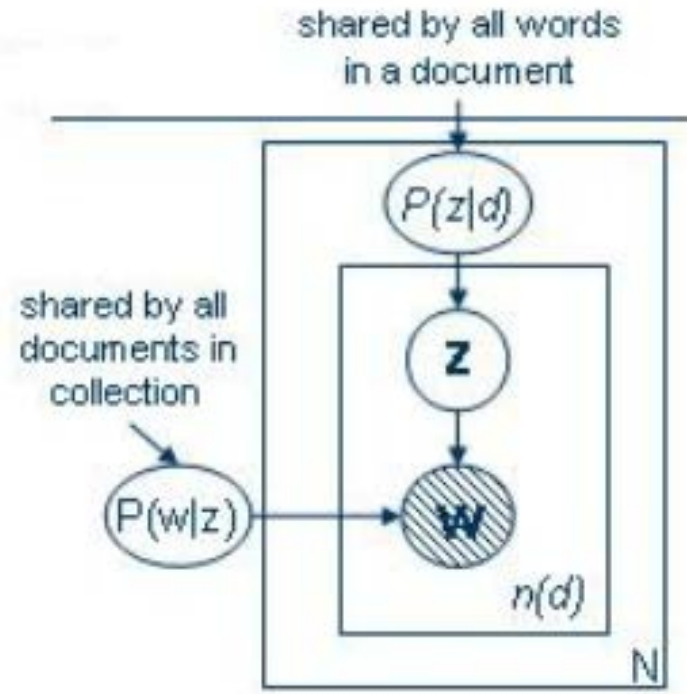
Problem: We only have the realizations of the words, and concepts are not observed (they are latent). How can we infer the probabilities of different words and concepts from the at hand documents???

The generative process:



- Each **concept** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of these topics
- We only observe the words within the documents and the other structures are **hidden variables**.

- Select a document with probability $P(d)$
- Pick a latent class z with probability $P(z|d;\theta)$
- Generate a word w with probability $P(w|z;\pi)$



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

$$\hat{P}_{LSA}(w|d) = \sum_{z \in Z} P(w|z;\theta)P(z|d;\pi)$$

$$\hat{P}_{LSA}(d,w) = P(d)\sum_{z \in Z} P(w|z)P(z|d) = \sum_{z \in Z} P(d|z)P(z)P(w|z)$$

This can be demonstrated as a matrix factorization

$$\hat{P}_{LSA}(d,w) = \sum_{z \in Z} P(d|z)P(z)P(w|z)$$

Contrast to SVD:

- No orthonormality condition for U and V here.
- The elements of U and V are non-negative.

Maximum Likelihood Estimation (ML):

Find all the parameters such that the probability of observing the corpus is maximized.

Likelihood function to be maximized: $L = \prod_{i=1}^{N} \prod_{j=1}^{M} P(d_i, w_j)^{n(d_i, w_j)}$

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

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

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

Estimated directly from data:

- $P(d_i)$: uniform or related to popularity of the document d_i
- $n(d_i)$: number of words in d_i
- $n(d_i, w_j)$: count of word w_j in d_i

The coupling effects of z_k makes this a hard optimization problem!

We use the standard Expectation Maximization (EM) algorithm to find an optimal solution.

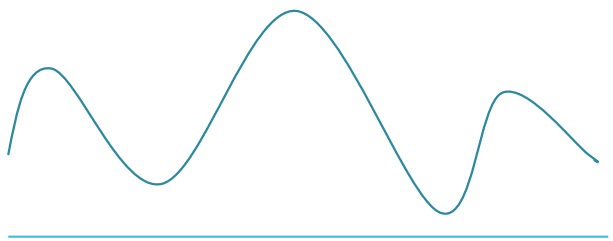
EM Algorithm:

Finding the solution of ML or MAP when some data is missing i.e. we have **latent** variables not observed.

Given: observations $x = (x_1, \dots, x_n)$ of the random variable X.
Model: $(X, Z) \sim p_\theta$ for some unknown parameter θ.
Goal: $\theta_{ML} = \arg\max_\theta p_\theta(x)$ since we only observe X

Issue: we have a marginal probability $p_\theta(x) = \sum_z p_\theta(x, z)$ which is difficult to maximize analytically (mainly because of the sum)

Also, you will probably have local maxima!



EM Algorithm improves $p_\theta(x)$ in two iterative steps namely E step and M step such that $p_{\theta_{t+1}}(x) \geq p_{\theta_t}(x)$

Note: Since you may have local maxima, EM might not give you the global optimum i.e. θ_{ML}

Back to our problem:

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

EM for our problem (repeat until convergence):

1. **E-step:** Calculate posterior probabilities for latent variables given the observations and current estimates
2. **M-step:** Update parameters using the posterior probabilities in E-step to increase $\log L$

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

1. **E-step:** Calculating posterior probabilities using the current estimates

$$P(z_k|d_i, w_j) = \frac{P(w_j, z_k|d_i)}{P(w_j|d_i)} = \frac{P(w_j|z_k, d_i) P(z_k|d_i)}{\sum_{i=1}^{K} P(w_j|z_i, d_i) P(z_i|d_i)}$$

2. **M-step:** Maximizing $\log L$ having the posterior probability

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

Data Set: Topic Detection and Tracking corpus (TDT-1)

- Approximately 7 million words
- 15863 documents
- K=128

Two most probable topics that generate
The terms “flight” and “love.

probability

“plane”	“space shuttle”	“family”	“Hollywood”
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

Data Set: Topic Detection and Tracking corpus (TDT-1)

- Approximately 7 million words
- 15863 documents
- K=128

Four additional topics from the 128 topic-Decomposition of the TDT-1 corpus.

probability ↑

"Bosnia"	"Iraq"	"Rwanda"	"Kobe"
un	iraq	refugees	building
bosnian	iraqi	aid	city
serbs	sanctions	rwanda	people
bosnia	kuwait	relief	rescue
serb	un	people	buildings
sarajevo	council	camps	workers
nato	gulf	zaire	kobe
peacekeepers	saddam	camp	victims
nations	baghdad	food	area
peace	hussein	rwandan	earthquake

Data Set: CLUSTER generated by the author

- Abstracts of 1568 documents **on clustering**
- K=128

Eight selected topics from the 128 topic-Decomposition of CLUSTER

probability ↑

"segment 1"	"segment 2"	"matrix 1"	"matrix 2"	"line 1"	"line 2"	"power 1"	power 2"
imag	speaker	robust	manufactur	constraint	alpha	POWER	load
SEGMENT	speech	MATRIX	cell	LINE	redshift	spectrum	memori
texture	recogni	eigenvalu	part	match	LINE	omega	vlsi
color	signal	uncertainti	MATRIX	locat	galaxi	mpc	POWER
tissue	train	plane	cellular	imag	quasar	hsup	systolic
brain	hmm	linear	famili	geometr	absorp	larg	input
slice	source	condition	design	impos	high	redshift	complex
cluster	speakerind.	perturb	machinepart	segment	ssup	galaxi	arrai
mri	SEGMENT	root	format	fundament	densiti	standard	present
volume	sound	suffici	group	recogn	veloc	model	implement

We want $P(z|d_i)$ to be sparse over $z_1, \dots, z_k$ i.e. each document should be related to a small number of topics.

Also, we want $P(w|z_k)$ to be sparse over $w_1, \dots, w_M$ i.e. each topic should be associated with a small proportion of the words.

Note that no conditions have been enforced on $p(z|d)$ and $p(w|z)$ in the PLSI model.

Enforcing the sparsity conditions with a Dirichlet distribution on $p(z|d)$ and $p(w|z)$

Latent Dirichlet Allocation (LDA) Model which will be presented next

[1] Hofmann, Thomas. "Probabilistic latent semantic indexing." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.

[2] Video Lectures of Thomas Hofmann and David Blei on videolectures.net:
http://videolectures.net/slsfs05_hofmann_lsvm/
http://videolectures.net/mlss09uk_blei_tm/

[3] Homepage of Thomas Hofmann, Assistant Professor of CS at Brown University:
http://cs.brown.edu/~th/

[4] Homepage of David Blei, Associate Professor of CS at Princeton University:
http://www.cs.princeton.edu/~blei/topicmodeling.html

Questions?!