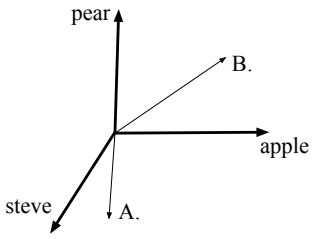
Topic Modelling

Charles Sutton
Data Mining and Exploration
Spring 2012

Goal

- Want a representation of documents for
  - Clustering
  - Retrieval
  - Detecting outliers (e.g., new events)
  - Visualisation
- Embed documents in a vector space

Vector Space (Geometrically)



Two news articles:

A. **Apple** unveils iPad tablet computer
BBC News - 2 hours ago
Apple has put an end to weeks of speculation by unveiling its tablet device, which it has called the iPad. Steve Jobs, **Apple's** chief executive unveiled the ...

B. **Apple**, **pear** crop to shrink 15pc
Stuff.co.nz - 18 hours ago
This season's **apple** and **pear** export crop is expected to be down 15 per cent nationally on last year's, but only 3 per cent in Nelson. ...
NZ **apple** forecast down Freshinfo

$$\theta = (n_{v_1}, n_{v_2}, \ldots, n_{v_m})$$

continuous representation of document

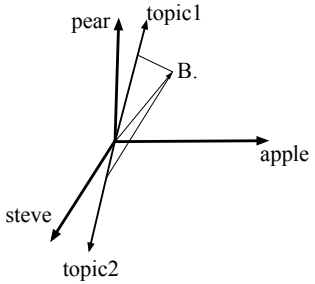
number of times word 2 in the vocabulary occurs in the document

(Could also incorporate idf, if you know about that.)

Latent Semantic Analysis

IDEA: Embed the documents in a lower-dimensional space.

Like a change of basis in linear algebra, i.e.,

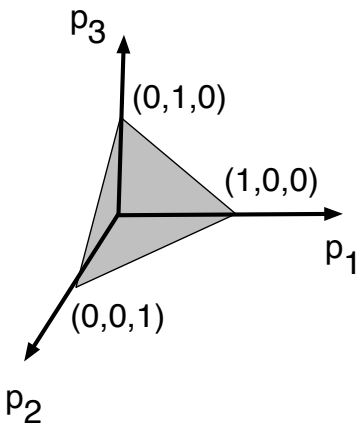


Each of the "basis vectors" is a point in the vocabulary space.

So we can view them as a "topic", i.e., a weighted combination of words that tend to co-occur.

To represent a document, we project it onto each of the topics.

The Simplex



The probability simplex is the set of points

$$(p_1, p_2, \ldots, p_n)$$

such that

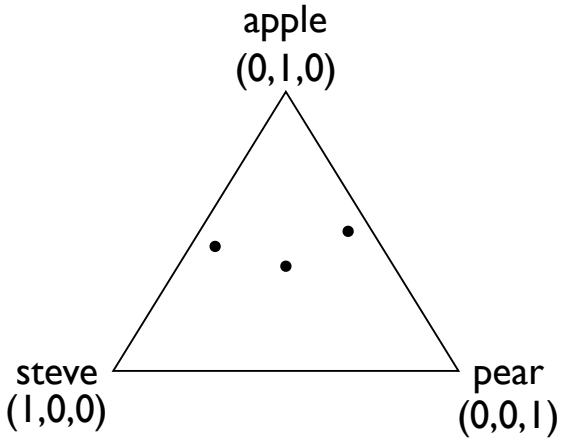
$$\sum_i p_i = 1$$ **hyperplane**

and

$$p_i \geq 0, \forall i$$ **positive orthant**

Every point on the simplex represents a probability distribution with n outcomes.

The Simplex (for Words)

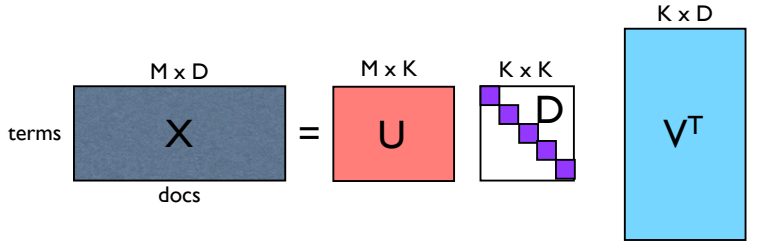


Problems

- This simple vector space representation is useful for information retrieval
- But not as useful for lots of other things
  - Visualisation
  - Event detection
- Want to reduce the dimensionality

To find basis vectors: use Principal Components Analysis.

$$X = UDV^T$$



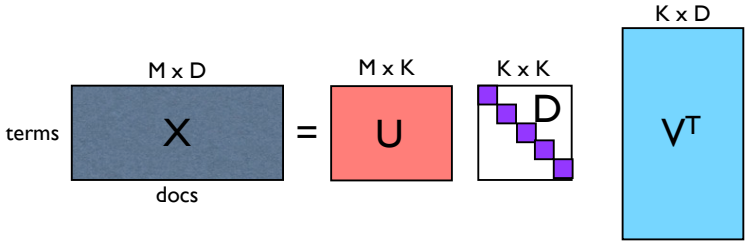
M: Number of terms
D: Number of documents
K: Number of "topics" (size of latent space)

U and V are orthonormal, i.e. $U^TU = I$

The columns of UD are the principal components

To find basis vectors: use Principal Components Analysis.

$$X = UDV^T$$



Columns of UD: "location" of a term in the latent space

Columns of DV^T: "location" of a doc in the latent space

e.g., Call $S = DV^T$ with columns $S = [s_1 s_2 \ldots s_D]$

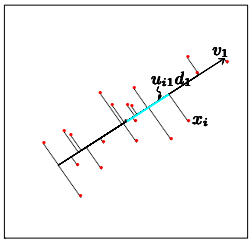
Then for each document $x_i = U s_i$

Why this factorisation?

$$X = UDV^T$$

(where U,V singular vectors, D singular values)

minimises reconstruction error



i.e., finds the best low-dimensional plane for projecting the data

Terminology

Singular value decomposition: The factorisation $X=UDV^T$

Principal components analysis: application of SVD to data matrices

Latent semantic analysis: application of PCA to term-document matrices

Latent semantic indexing: application of LSA in a search engine

Square matrices have *eigenvectors* and *eigenvalues*

Non-square matrices have *singular vectors* and *singular values*

Special bonus fact: SVD is the same as eigenvalue-finding.
The columns of U are the eigenvectors of X^TX.
(Easy to show: Use fact that V is orthonormal.)

Technical Memo Example

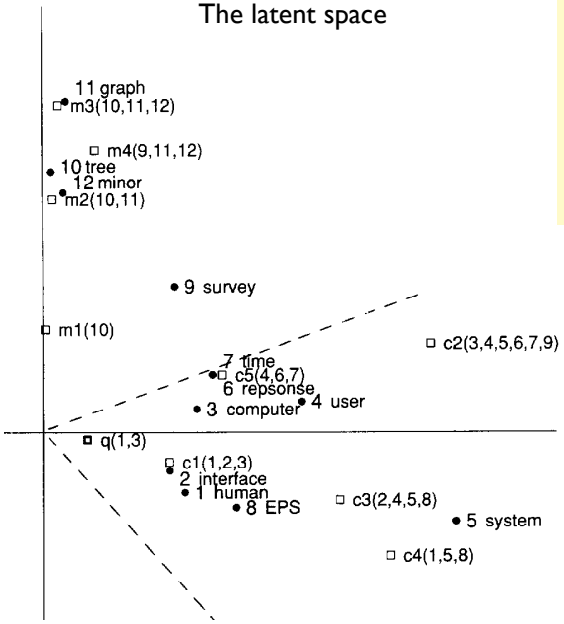
Titles

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

[Deerwester et al, J of Am Soc for Information Science, 1990]

The latent space



Filled circles: Terms

Open squares: Documents

q: query "human computer"

dashed lines: cosine distance of 0.9 from q

Problems with LSA

- Possible that the "reconstruction" of a document has negative term frequencies
- Implicit Gaussian assumption in the minimisation problem (i.e., from PCA)
- Not compositional: How do I incorporate other types of information

Probabilistic LSA

We can avoid these difficulties using a probabilistic model. To do this, we need to:

1. Define a model
2. Figure out how to do inference
3. Figure out how to do parameter estimation

[Hoffman, UAI 1999]

Probabilistic Latent Semantic Analysis

For each document d

Generate N_d from Geometric(p_d)

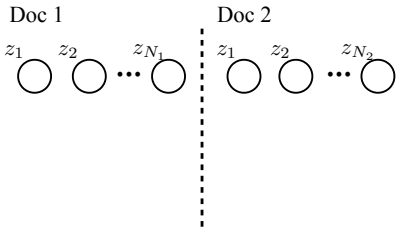
Probabilistic Latent Semantic Analysis

For each document d

Generate N_d from Geometric(p_d)

For word index $i = 1, 2, \ldots N_d$

Sample z_i from a discrete distribution with parameters θ_d



Each z_i is a cluster label, i.e., $z_i \in \{\text{Cluster0}, \text{Cluster1}, \ldots, \text{ClusterK}\}$

So θ_d a distribution over the cluster labels, i.e., a point on the K-dimensional simplex

$$\theta_{dk} = p(z = \text{Cluster}_k | d)$$

Probabilistic Latent Semantic Analysis

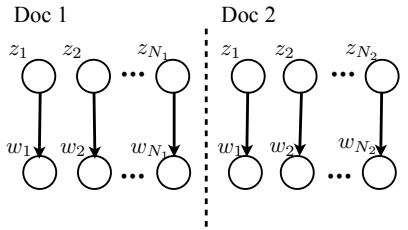
For each document d

Generate N_d from Geometric(p_d)

For word index $i = 1, 2, \ldots N_d$

Sample z_i from a discrete distribution with parameters θ_d

Sample w_i from a discrete distribution with parameters ψ_{z_i}



ψ_{z_i} a distribution over words, i.e., a point on the M-dimensional simplex

One of these distributions for each possible cluster.

$$\psi_{k,v} = p(w_j = v | z_j = \text{Cluster}_k)$$

Use Plates

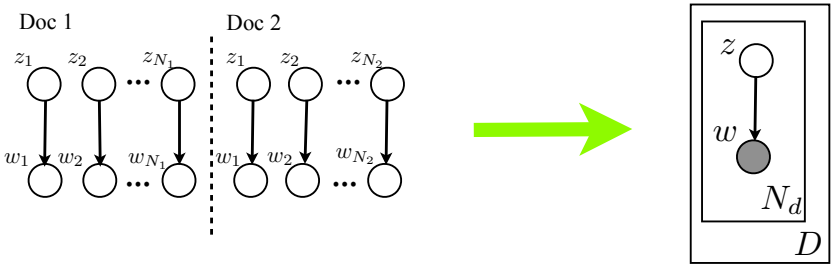


Plate means "copy this structure over and over"

The Parameters

$\theta = (\theta_1, \theta_2, \ldots, \theta_D)$ $\quad D(K-1)$

The distributions that generate z's

Each one of these guys is a distribution over topics

$\psi = (\psi_1, \psi_2, \ldots, \psi_K)$ $\quad K(M-1)$

The distributions that generate w's

Each one of these guys is a distribution over words

How do we estimate all of these?

(This is a lot.)

Analogy to LSA

Terms to topics

M x K

U

$\psi = (\psi_1, \psi_2, \ldots, \psi_K)$ $\qquad K(M-1)$

$\sim$ The distributions that generate *w*'s

Each one of these guys is a distribution over words

Documents to topics

K x D

V^T

$\theta = (\theta_1, \theta_2, \ldots, \theta_D)$ $\qquad D(K-1)$

$\sim$ The distributions that generate *z*'s

Each one of these guys is a distribution over topics

Now what?

1. Define a model

$$p(w, z, N_d) = (1-p_d)^{N_d} p_d \prod_{i=1}^{N_d} \theta_{d,z_i} \psi_{z_i,w_i}$$

$p(N_d)$ $\quad p(z_i|N_d)$ $\quad p(w_i|z_i, N_d)$

Now what?

1. Define a model

$$p(w, z, N_d) = (1-p_d)^{N_d} p_d \prod_{i=1}^{N_d} \theta_{d,z_i} \psi_{z_i,w_i}$$

2. Figure out how to do inference

3. Figure out how to do parameter estimation

1. Inference

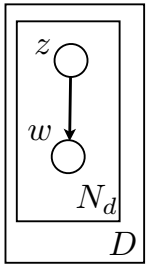
Inference here means:

Fix θ, ψ

For each document, word, compute

$p(w_j; \theta, \psi)$

in PLSA model:



1. Inference

Inference here means:

Fix θ, ψ (assume they're known for certain)

For each document, word, compute

$$p(w_{dj};\theta,\psi) = \sum_{z_{dj}=1}^{K} p(w_{dj}, z_{dj};\theta,\psi)$$

$$= \sum_{z_{dj}=1}^{K} p(w_{dj}|z_{dj};\psi)p(z_{dj};\theta)$$

$$= \sum_{z_{dj}=1}^{K} \psi_{w_{dj},z_{dj}}\theta_{d,z_{dj}}$$

2. Parameter estimation

Use maximum likelihood. The logarithm of the likelihood is

$$\mathcal{L}(\theta,\psi) = \log p(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D;\theta,\psi)$$

This is the marginal distribution we just looked at.

$$= \log \prod_{d=1}^{D}\prod_{i=1}^{N_d}\sum_{z_{dj}=1}^{K} p(z_{dj}|\theta_d)p(w_{dj}|z_{dj},\psi)$$

$$= \sum_{d=1}^{D}\sum_{i=1}^{N_d}\log \sum_{z_{dj}=1}^{K} \theta_{d,z_{dj}}\psi_{w_{dj},z_{dj}}$$

We want to maximise this distribution with respect to

$$\theta = (\theta_1, \theta_2, \ldots, \theta_D)$$
$$\psi = (\psi_1, \psi_2, \ldots, \psi_K)$$

subject to $\sum_{z=1}^{K}\theta_{dk} = 1, \forall d$ $\quad \theta_{dk} \geq 0$

$\sum_{v=1}^{M}\psi_{vk} = 1, \forall k$ $\quad \psi_{vk} \geq 0$

Expectation Maximisation

E-step: "Fill in" distribution over missing data using the current parameters

Here: Missing data are the z_{dj}

$$q_{dj} := p(z_{dj}|w_{dj},\theta,\psi)$$

$$= \frac{p(z_{dj}, w_{dj}|\theta,\psi)}{p(w_{dj}|\theta,\psi)}$$

→ this is just $\psi_{z_{dj},w_{dj}}\theta_{d,z_{dj}}$

→ eqn for this on "Inference" slide

M-step: Find new parameters by maximising likelihood with "filled in" distribution

$$\theta^*, \psi^* = \arg\max_{\theta,\psi} \sum_d \sum_j \sum_{z_{dj}} q_{dj} \log p(w_{dj}|z_{dj})$$

The solution looks the same as ML with multinomial data, e.g.,

Iterate: E-step and M-step until converged

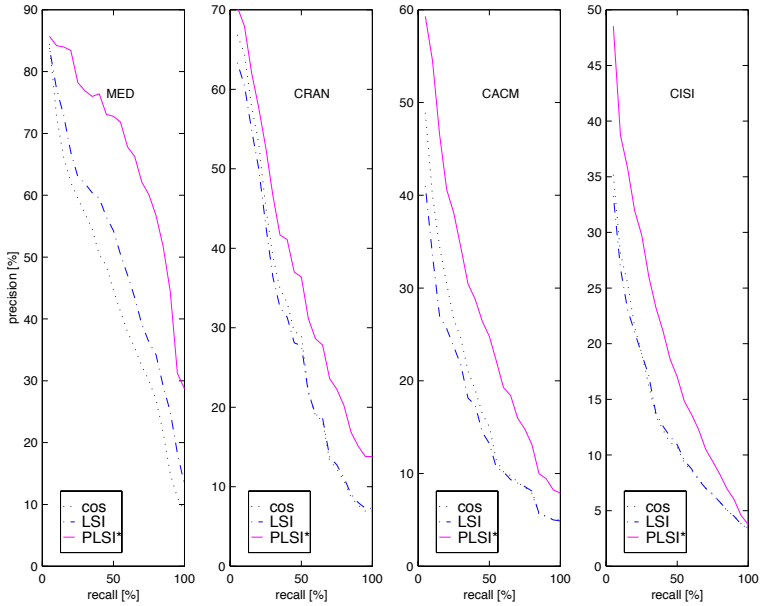
(Detail: Hoffman uses a "tempered EM" to get this to work. You don't need to know about this.)

Topics

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

Figure 3: Eight selected factors from a 128 factor decomposition. The displayed word stems are probable words in the class-conditional distribution $P(w|z)$, from top to bottom in descending orc

Use in IR



What's wrong with PLSA

- Too many parameters [Practical]
- Not really a probabilistic model at the document level [Aesthetic]

Solution to both: Be Bayesian!

Remember Bayesianism

Maximum likelihood: Maximize over parameters, Bayes: Integrate over them

Example: Flipping a coin. Want to estimate probability θ of coin coming up heads.

Number of heads: N_h Number of flips: N

Likelihood:

$$p(N_h|\theta) = \binom{N}{N_h} \theta^{N_h}(1-\theta)^{N-N_h}$$

Parameter estimate (ML):

$$\hat{\theta} = \arg\max_{\theta \in [0,1]} \log p(N_h|\theta) = \frac{N_h}{N}$$

Bayesian:

Prior (ex: uniform):

$$p(\theta) = 1 \text{ if } \theta \in [0,1], 0 \text{ otherwise}$$

Joint model:

$$p(N_h, \theta) = p(N_h|\theta)p(\theta)$$

Posterior:

$$p(\theta|N_h) = \frac{p(N_h|\theta)p(\theta)}{p(N_h)}$$

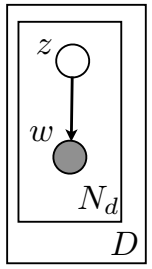
In this example:

$$\theta|N_h \sim \text{Beta}(1+N_h, 1+N-N_h)$$

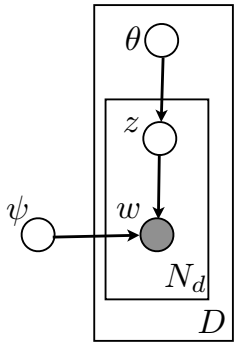
$$p(\theta|N_h) = C(N_h)\theta^{N_h}(1-\theta)^{N-N_h}$$

We can do the same thing using the pLSA likelihood. This give a Bayesian pLSA, which is called latent Dirichlet allocation (LDA).

Latent Dirichlet Allocation



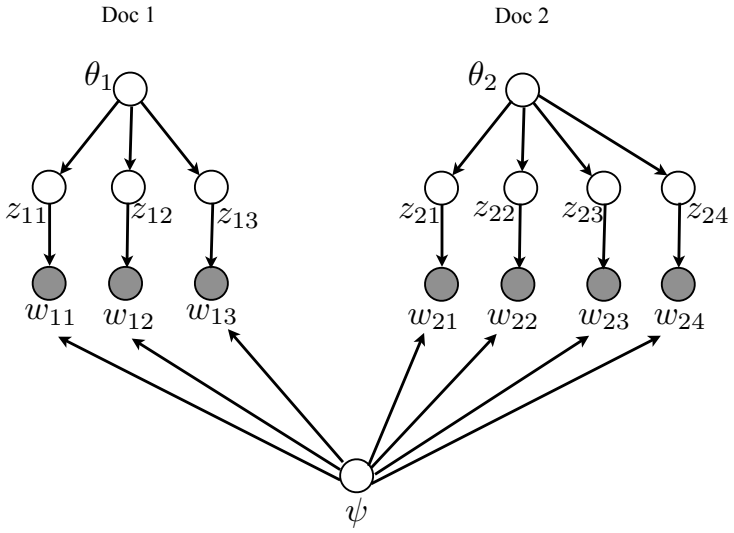
PLSA



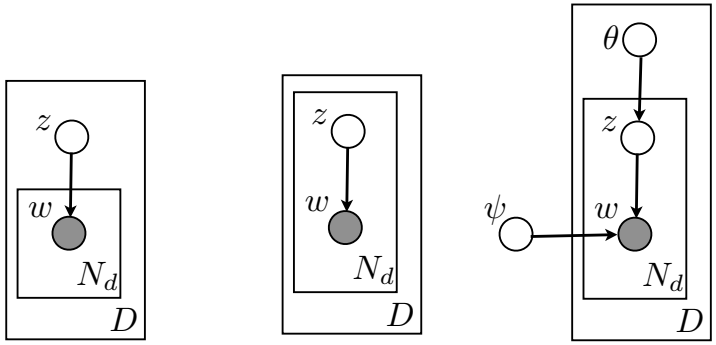
LDA

[Blei, Ng, and Jordan, 2003]

LDA, unrolled

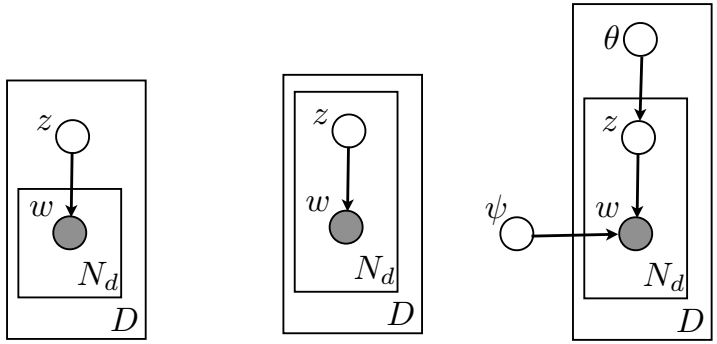


Digression



??? PLSA LDA

Digression



Document clustering PLSA LDA

Dirichlet Distribution

Dirichlet is a generalization of the beta distribution for n-dimensional probability vectors rather than 2-d

Dirichlet is conjugate to multinomial just as beta is to binomial

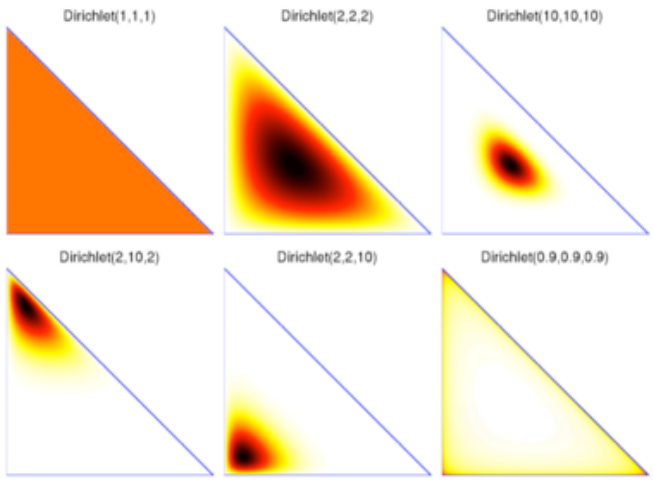
$$p(\theta|\alpha) = C \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

where θ is a point on the unit simplex, i.e.,

$$\theta_k \geq 0, \qquad \sum_k \theta_k = 1$$

C is some number that does not depend on θ and that you don't have to think about for our purposes.

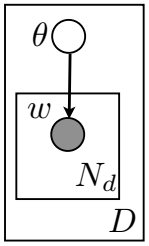
Dirichlet Examples



[Source: https://projects.csail.mit.edu/church/wiki/Models_with_Unbounded_Complexity]

Exercise

If the idea of Bayes for pLSA is in the clouds, consider this model



$$p(\theta|\alpha) = C\prod_{k=1}^{K}\theta_k^{\alpha_k-1}$$

$$p(\mathbf{w}_1 \ldots \mathbf{w}_D|\theta) = \prod_{d=1}^{D}\prod_{j=1}^{N_d}\theta_{w_{dj}}$$

Q: What is the formula for the posterior

$$p(\theta|\mathbf{w}_1 \ldots \mathbf{w}_D)$$

The answer is very similar to the coin flip case.

Think of it like a die with one word from the vocabulary on each face.

Latent Dirichlet Allocation

For each topic *k*

Generate ψ_k from Dirichlet(β)

Latent Dirichlet Allocation

For each topic *k*

Generate ψ_k from Dirichlet(β)

For each document *d*

Generate θ_d from Dirichlet(α)

Latent Dirichlet Allocation

For each topic *k*

Generate ψ_k from Dirichlet(β)

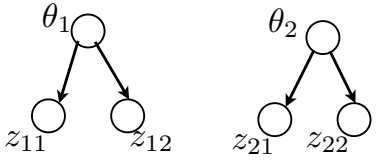
For each document *d*

Generate θ_d from Dirichlet(α)

Generate N_d from Geometric(p_d)

For word index $i = 1, 2, \ldots N_d$

Sample z_i from a discrete distribution with parameters θ_d



Latent Dirichlet Allocation

For each topic *k*

Generate ψ_k from Dirichlet(β)

For each document *d*

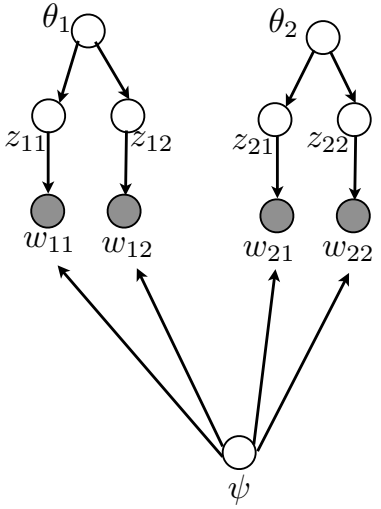
Generate θ_d from Dirichlet(α)

Generate N_d from Geometric(p_d)

For word index $i = 1, 2, \ldots N_d$

Sample z_i from a discrete distribution with parameters θ_d

Sample w_i from a discrete distribution with parameters ψ_{z_i}



OK. Now what?

There are two things that you want to do with any probabilistic model:

1. Probabilistic inference

2. Estimate parameters

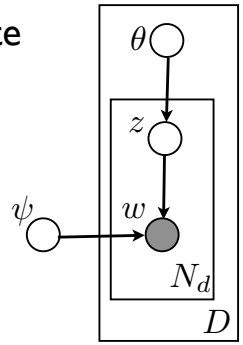
1. Inference

Inference here means:

For each document, word, compute

$$p(\theta, \psi, \mathbf{z}_1, \ldots, \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

in LDA model:



For each document, word, compute

$$p(\theta, \psi, \mathbf{z}_1, \ldots, \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

Two main ways to approximate this:

1. Variational methods
2. Markov chain Monte Carlo

You'll learn both in MLPR, but for now....

Monte Carlo

As a Bayesian, I care about the posterior:

$$p(\theta, \psi, \mathbf{z}_1, \ldots, \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

Suppose I want to know the posterior mean, i.e.:

$$E[\theta | \mathbf{w}_1 \ldots \mathbf{w}_D] = \int_\theta \int_\psi \sum_{\mathbf{z}_1 \ldots \mathbf{z}_D} \theta p(\theta, \psi, \mathbf{z}_1 \ldots \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

Can't compute this analytically. But suppose that I can draw samples:

$$\theta_m \sim p(\theta, \psi, \mathbf{z}_1 \ldots \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D) \quad \text{for} \quad m = 1, 2, \ldots M$$

Monte Carlo

Suppose I want to know the posterior mean, i.e.:

$$E[\theta | \mathbf{w}_1 \ldots \mathbf{w}_D] = \int_\theta \int_\psi \sum_{\mathbf{z}_1 \ldots \mathbf{z}_D} \theta p(\theta, \psi, \mathbf{z}_1 \ldots \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

Can't compute this analytically. But suppose that I can draw samples:

$$\theta_m \sim p(\theta, \psi, \mathbf{z}_1 \ldots \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D) \quad \text{for} \quad m = 1, 2, \ldots M$$

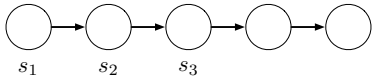
Then can approximate

$$E[\theta | \mathbf{w}_1 \ldots \mathbf{w}_D] \approx \frac{1}{M} \sum_{m=1}^{M} \theta_m$$

This is called Monte Carlo integration

Markov Chain

A Markov chain is any probability distribution



We write it as $p(s_1, s_2, \ldots s_N) = \prod_{i=1}^{N} p(s_i | s_{i-1})$

transition distribution
transition kernel

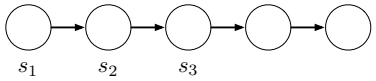
Marginal distributions

$$p(s_1), p(s_2), p(s_3), \ldots$$

can be computed by the forward algorithm...

Markov Chain

A Markov chain is any probability distribution



Marginal distributions

$$p(s_1), p(s_2), p(s_3), \ldots$$

can be computed by the forward algorithm...

Question: Does this limit exist:

$$p^*(s) = \lim_{T \to \infty} p(s_T)$$

In many cases, yes. This is called a stationary distribution.

Stationary Distributions

Example: $s_T \in \{0, 10\}$

$$s_1 = 3$$

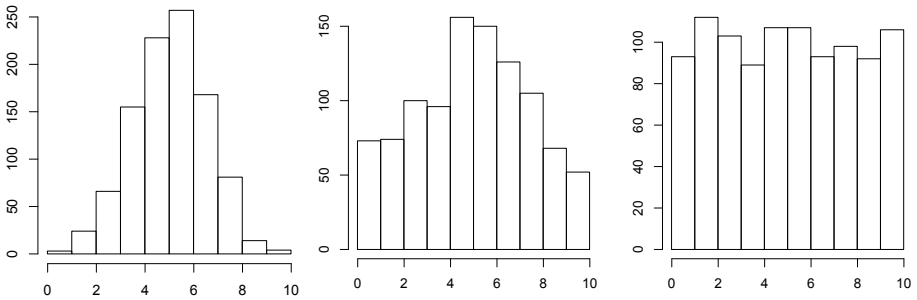
$$p(s_T | s_{T-1}) = \begin{cases} s_{T-1} + 1 & \text{with probability } 0.5 \\ s_{T-1} - 1 & \text{with probability } 0.5 \end{cases}$$

This is called a "random walk".

Then it's actually easy to see that:

$$p^*(s) = \tfrac{1}{11} \text{ for any } s \in \{0, 10\}$$

Example



Markov Chain Monte Carlo

But I don't know how to sample

$$\theta_m \sim p(\theta, \psi, \mathbf{z}_1 \ldots \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

MCMC: Create a Markov chain whose state is

$$s_m = (\theta_m, \psi_m, \mathbf{z}_{1m} \ldots \mathbf{z}_{Dm})$$

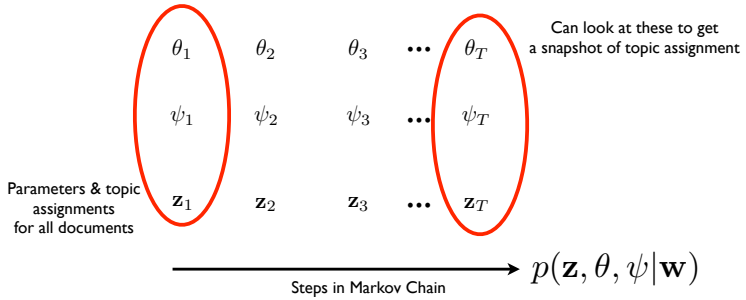
and whose transition kernel is chosen cleverly so

$$p^*(s) = p(\theta, \psi, \mathbf{z}_1 \ldots \mathbf{z}_D | \mathbf{w}_1 \ldots \mathbf{w}_D)$$

How do you pick the transition kernel?

MCMC output

You get a sequence of parameter settings:



2. Parameter Estimation

Actually, we're being Bayesian. So we don't do this.

Once we've sampled from the posterior

$$p(\theta, \psi|\mathbf{w}_1, \ldots \mathbf{w}_N)$$

We're happy. This gives us

1. Distribution over embedding of each document
2. Distribution over topics for each word
3. Distribution over topic-word probabilities

"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
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

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. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Summary

Methods for dimensionality reduction in documents

1. Latent semantic analysis
2. Probabilistic latent semantic analysis
3. Latent Dirichlet allocation

Really, main uses: visualisation, as building blocks

Interesting things to note:

* Win from Bayesianism
  LDA is simply Bayesian PLSA

* Move from a linear algebra approach to graphical modeling
  (cf. factor analysis and beyond from PMR)