manipal PROlearn

Topic Models

By

Mirza Rahim Baig

Agenda

- Motivating: making sense of text
- Topic modeling and applications
- Some very high level intuition
- Input and output of a topic model
- The simplest model: a term as a topic; limitations of this approach
- Probabilistic topic models
- PLSA
- LDA
  - Dirichlet process
  - LDA parameters
  - Inference
- LDA hands on
- Practical considerations
- Summary

Topics and their discovery, and its utility

Topic: main idea being discussed in the text

Topic discovery: identifying the topics discussed in the text

Just that!

Is it useful?

What could be the applications?

1. What do people like or dislike about a product?
2. What are the main topics in customer survey responses?
    a. What were the pain points?
    b. What went well?
3. What twitter users are talking about?
4. Assessing document similarity and retrieval
5. Automatic labeling of documents (to multiple categories)
    a. News article to multiple categories
6. Map documents to new latent/hidden topic space, follow up with ML algorithm
    a. Or just dimensionality reduction

Some very, very high level intuition

I’ve been told I give a lot of advice (aka. gyaan), both solicited and unsolicited

- About a few limited, recurrent topics

Let’s say some disciple of mine transcribed all the gyaan I delivered to humankind, so far

Guess the topic from some key words -

Topic 1	Statistics, model, learning, language, text
Topic 2	happiness, satisfaction, reflection, validation, priorities
Topic 3	stoic, virtue, existentialism, radical

- Looking at a few terms was enough to give a fair idea of what the topic is

Some very, very high level intuition continued

We have identified our topics.

Topic 1	statistics, model, learning, language, text
Topic 2	happiness, satisfaction, reflection, validation, priorities
Topic 3	stoic, virtue, existentialism, radical

With this identification, given any text, we can estimate what topics are being covered.

Guess the topics being discussed in the following statement:

“Working with text and modeling the complexities of natural language is something high on my priorities as it provides me with great satisfaction and happiness”

- Text is about topics 1 and 2 in equal proportion, topic 3 is absent (0.5, 0.5, 0)

This is the core idea of topic modeling. Humans are very good at it!

Topic models: Input and output / defining the tasks

Given a corpus (set of documents), there are 2 outputs from a topic model -

	Word1	Word2	Word3	Word4	Word5	.	.	WordM
Topc1	0.3	0.1	0.2	0.3	0.1	0	0	0
Topic2	0	0.2	0.1	0	0.2	0	0	0.5
Topic3	0.05	0.1	0.2	0	0.1	0.3	0	0
Topic4	0	0.1	0.1	0	0.1	0	0.3	0.4
Topic5	0.1	0.05	0.3	0.05	0	0	0.5	0

Topic - Words i.e. topic ‘definition’ or ‘composition’

- A certain mix of words is a ‘topic’
- A probability distribution over terms
- Words can appear across topics

	Topic1	Topic2	Topic3	Topic4	Topic5
Doc1	0.1	0	0.3	0	0.6
Doc2	0.6	0.2	0	0.1	0
Doc3	0.15	0.15	0.65	0.05	0
Doc4	0.1	0	0.4	0	0.5
Doc5	0	0	0	0	1

Doc - Topic i.e. topic coverage by each document

- A document can talk of more than 1 topic
  - A blog about statistics and medicine
  - About science and technology
- Mixed membership assignment (in clustering terms)
- A probability distribution over topics

Topic models: Input and output / defining the tasks

The two major tasks in topic modeling:
1. defining topic
2. estimating coverage

Input: a collection of **M** documents with **K** topics
Output:

1. Topics $\{\theta 1, \theta 2, .., \theta k\}$,
2. Topic Coverage in each document $\{\pi_{i1}, \pi_{i2}, .., \pi_{ik}\}$, wit $\sum_{j=1}^{k} \pi_{ij} = 1$

Hold on! Let's not get too ahead of ourselves. Let's begin at the beginning instead of the end, to really appreciate topic models as they are used today.

Let's first ask the foremost question: ***How do we define a topic?***

The most basic approach - a term as a topic

Let's first look at this simplest approach

The two major tasks in topic modeling:
1. defining topic: easy, each term is a topic
2. estimating coverage: ?

How do we go about task 2?

$$\pi_{ij} = \frac{\text{count}(\theta_j, d_i)}{\sum_{L=1}^{k} \text{count}(\theta_L, d_i)}$$

Example: Two topics

- **Astronomy: 'stars'**
- **Movies: 'film'**

*"The **stars** are out tonight at the 19th annual **film** awards"*

Topic as a distribution over terms

Re-defining a topic: distribution over a vocabulary

What issues does this solve?

- Different weights to same word in different topics allow for subtle differences in topics
- Word sense disambiguation depending on the topic

θ_1: Magic

P (w | θ_1)
magic: 0.09
trick: 0.08
card: 0.02
black: 0.015
.
art: 0.007
science: 0.005
.
travel: 0.001

θ_2: Science

P (w | θ_2)
science: 0.05
experiment: 0.05
lab: 0.03
research: 0.02
.
art: 0.003
budget: 0.003
.
magic: 0.001

S

Possible Approach: Matrix Factorization (LSA)

Applying SVD on the Document - Term matrix

$$A \approx U \, S \, V^T$$

MxV | MxK | KxK | KxV

M: number of documents
V: vocabulary (unique words)
K: number of topics

Document - Topic coverage (U)

Topic - Term distributions (V^T)

Drawback?

LSA Hands-on

Probabilistic modeling

1. Assume a data model

- Data arrived from some generative probabilistic process
- Sequence of probabilistic steps
- Includes hidden variables
    - structures in the data we don’t have access, want to find

2. Infer the hidden structure using posterior inference

- We’ll compute the conditional distribution of the hidden variables

Probabilistic modeling

Lambda = all parameters of the model

Lambda* = argmax(P (data | model, lambda)

Probability vs lambda curve?

Maximum likelihood estimation

Task: given some data, estimate the probabilities in the model

What would your estimate be?

Document

anxiety	: 40
help	: 5
medical	: 15
depression	: 15
consult	: 5

Language model

anxiety	:
help	:
medical	:
depression	:
consult	:

Likelihood maximized when estimate is same as observed empirical probability

- Gives our observed text data the best probability
- So this is our maximum likelihood estimate

Maximum likelihood vs Bayesian

Maximum likelihood estimation -

- Defined best as “data likelihood reaches maximum”
- Problem: small sample; will trust data entirely

$$\hat{\theta} = \arg\max_{\theta} P(X \mid \theta)$$

Bayesian estimation -

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid X) = \arg\max_{\theta} P(X \mid \theta) P(\theta)$$

- Best means
  - being consistent with our ‘prior’ knowledge, and
  - explaining data well
- Theta treated as random variable

Problem: how do we define prior?

MAP estimation derivation

Note: MLE is special case of MAP where the priors are uniform

One simple language model

The Unigram model

- Words drawn one at a time
- Words in a sentence are independent of the others

Accurate?
Useful?

Greatly simplifies our process and calculations

Question: But is this sufficient?

- Insufficient for sarcasm detection

- Sufficient for topic modeling

Activity - The (Imaginary) Generative Process

Author 1	**love, harmony, happiness, joy, priorities, satisfaction, life, benefit**
Author 2	**music, loud, party, drinks, fun, happiness, shots, love, crazy, booze**
Author 3	**statistics, model, learning, pattern, language, machine, accuracy**

Activity - The (Imaginary) Generative Process - Updat

High — Medium — Low

High

Author 1 — love, harmony, happiness, joy, priorities, satisfaction, life, benefit

Medium

Author 2 — music, loud, party, drinks, fun, happiness, shots, love, crazy, booze

Low

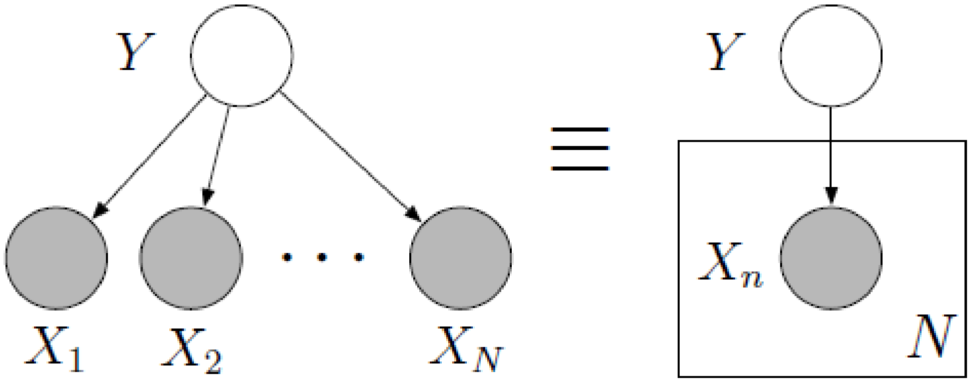
Author 3 — statistics, model, learning, pattern, language, machine, accuracy

Aside - Plate Notation

A concise way of visually representing the dependencies among the model parameters

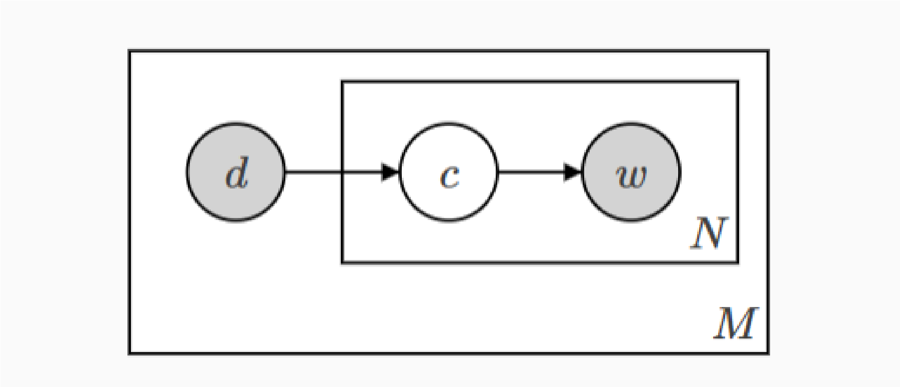
A basic intro to graphical models

- Nodes are random variables
- Edges (here) are dependencies: X1 depends on Y, X2 depends on Y etc.
- Observed variables are shaded, unobserved are not
- Plates represent repetitive structure



PLSA

A simple, but very useful, formulation



d = document index

c = word's topic drawn from P(c|d)

w = word drawn from P(w|c)

$$P(w,d) = \sum_c P(c)P(d|c)P(w|c) = P(d)\sum_c P(c|d)P(w|c)$$

Both P(c|d) and P(w|c) are modeled as multinomial distributions:

- P(c|d) is the topic distribution given the document
- P(w|c) is the term distribution given the topic

These distributions are parameters

- Large number of parameters. Parameters for P(c|d) increase linearly with the number of documents

Aside: The EM algorithm

Quick note on the Expectation Maximization Procedure:

- Hidden assignment variable used
  - Augmenting data by predicting values of useful hidden variable in E-step
- Initialize randomly
- Iterate using E-step and M-step
- Stop when likelihood doesn’t change

E-step: calculate expected value of likelihood given all other parameters

M-step: update parameters to maximize the likelihood

PLSA: Finding the parameters via. EM

Applying EM to PLSA:

Assignment variables and model parameters updated in different steps

E-step: Calculate Posterior probability P (c|d) with each word

$$P(c|d,w) = \frac{P(c)P(d|c)P(w|c)}{\sum_{\forall c \in C} P(c)P(d|c)P(w|c)}$$

M-step:

$$P(w|c) \propto \sum_{\forall d \in D} n(d,w)P(c|d,w)$$
$$P(d|c) \propto \sum_{\forall w \in W} n(d,w)P(c|d,w)$$
$$P(c) \propto \sum_{\forall d \in D} \sum_{\forall w \in W} n(d,w)P(c|d,w)$$

LSA Hands-on

LDA

‘Latent Dirichlet Allocation’ is an extension over PLSA

- Specified as a Bayesian model
    - Parameters are random variables with some known prior distributions
- Generalized form, where the distributions are inferred
- Account for uncertainty in parameters when making predictions
    - An averaged prediction using the probable values

Activity - The (Imaginary) Generative Process - Updat

High Medium Low

High

Author 1

love, harmony, happiness, joy, priorities, satisfaction, life, benefit

Medium

Author 2

music, loud, party, drinks, fun, happiness, shots, love, crazy, booze

Low

Author 3

statistics, model, learning, pattern, language, machine, accuracy

Aside: The ‘Dirichlet’ in LDA

Multinomial distribution:

- Distribution over discrete outcomes
- Represented by a non-negative vector that sums to 1 (simplex)
- Our topic-terms, and doc-topics are multinomial distributions

These multinomial distributions themselves are random outcomes drawn from a distribution.

- **A Dirichlet distribution!**

Parameters of a Dirichlet Distribution

In the most general form:

$$\mathrm{Dir}(\alpha_1, \ldots, \alpha_T) = \frac{\Gamma\left(\sum_j \alpha_j\right)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{T} p_j^{\alpha_j - 1}$$

Each α_j is a prior, before observing any actual data

A symmetric Dirichlet distribution used commonly, characterized by a single *α*.

$$f(x_1, \ldots, x_{K-1}; \alpha) = \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \prod_{i=1}^{K} x_i^{\alpha - 1}.$$

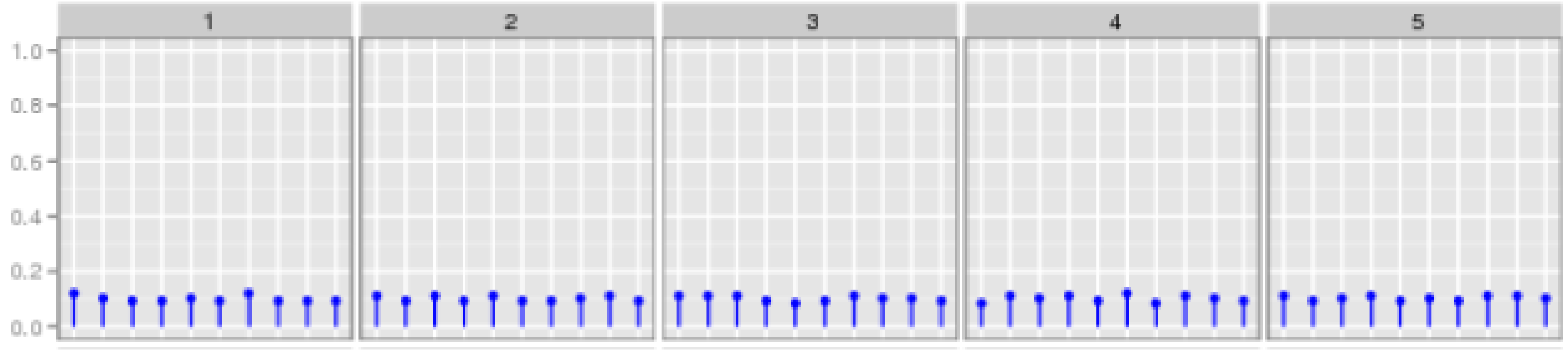
α is the concentration parameter

- Determines how spread your topics are
- Low alpha would mean sparse topics

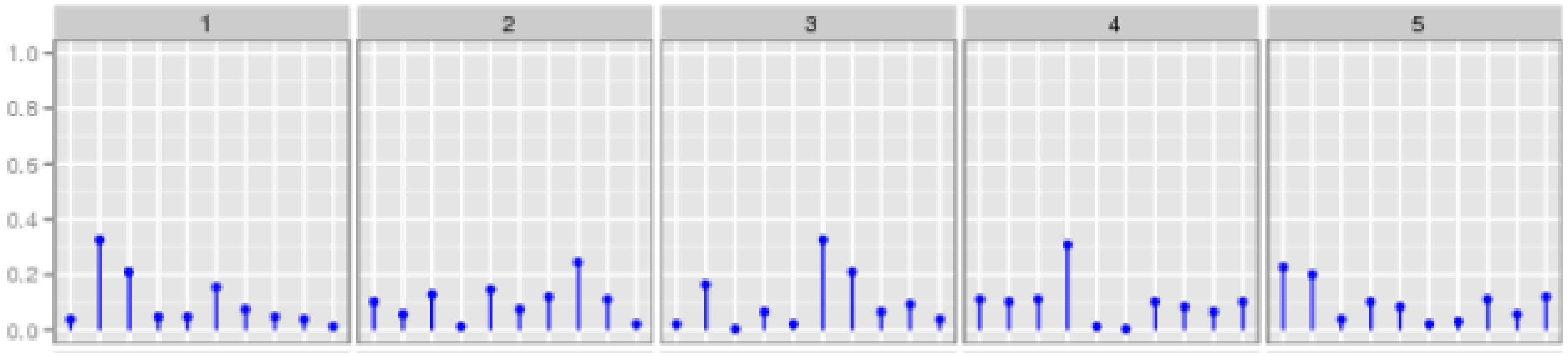
Let's examine the effect of the parameters in LDA

Effect of alpha

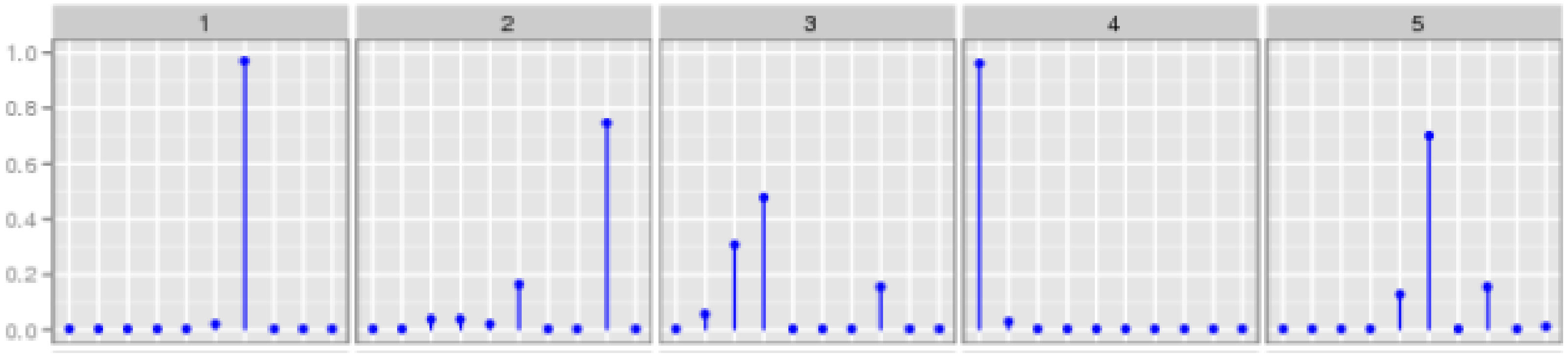
100



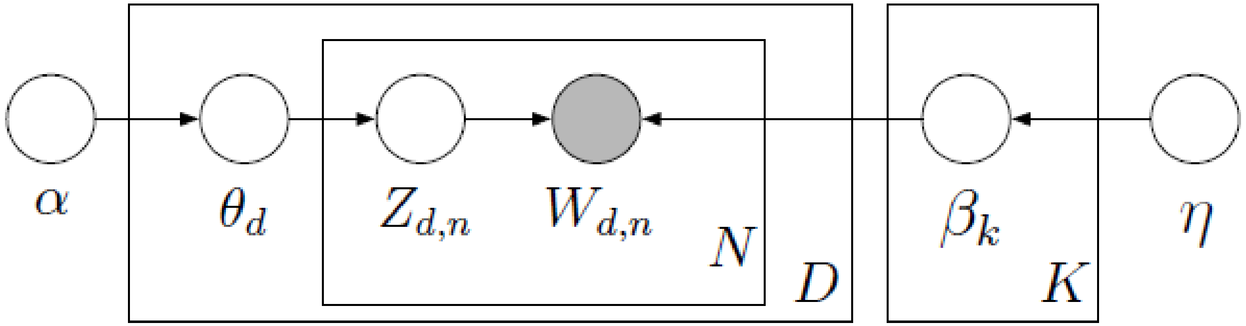
1



0.01



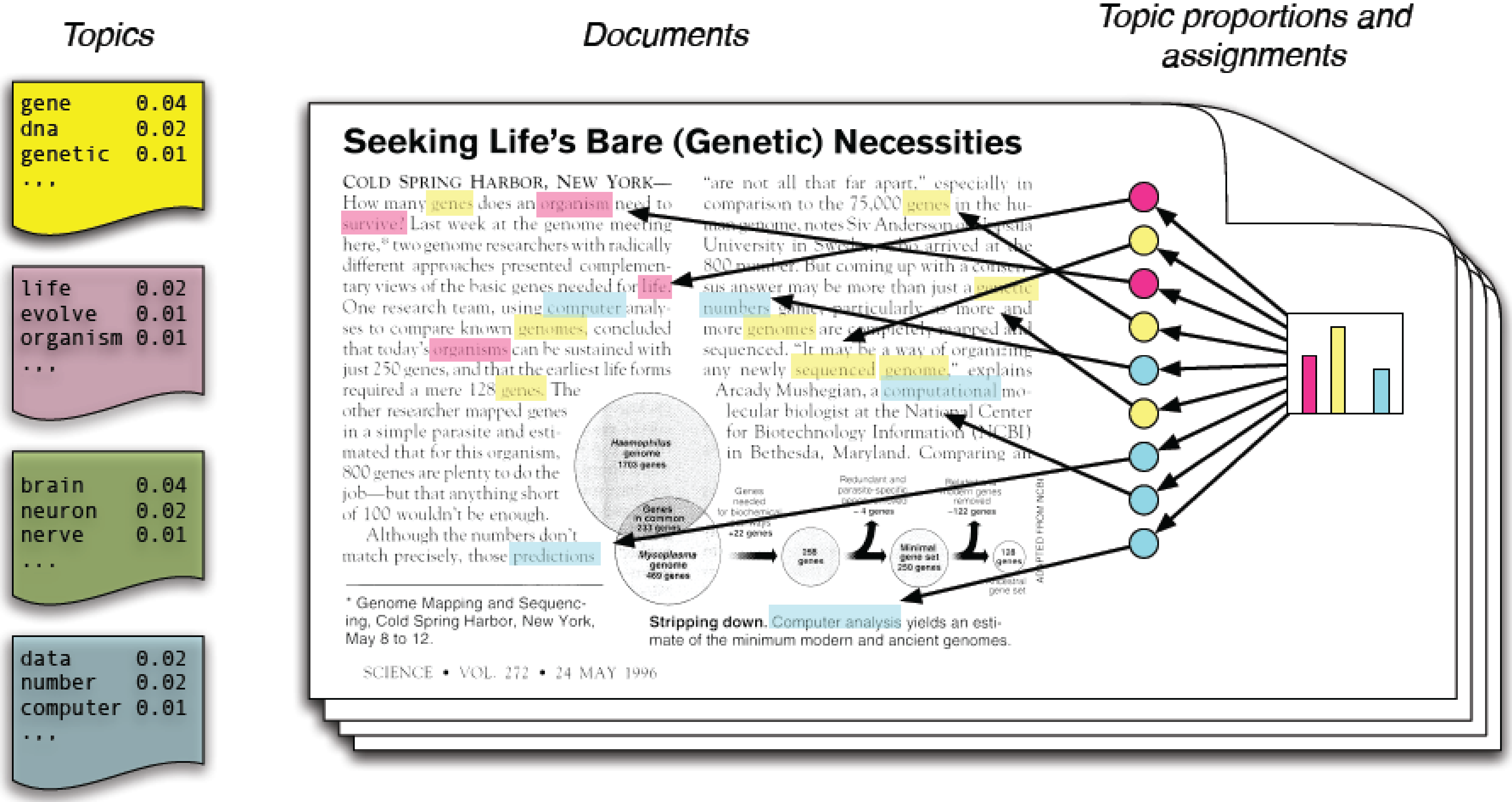
LDA parameters



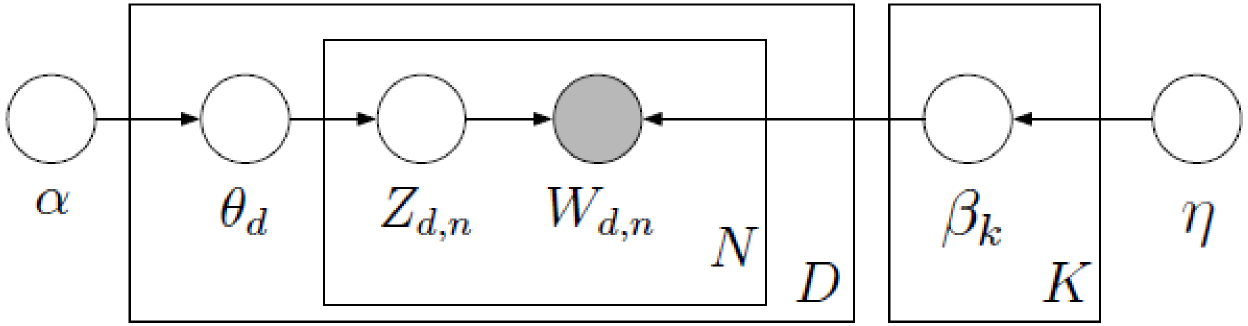
- Each beta is a distribution of terms (K distributions)
- Each theta is distribution of topics (D of them)
- Zdn is the topic assignment for the word in word position in that document
  - Zdn is a number from 1 to k
  - Depends on theta (theta sparse if low alpha)

Note: placing a Dirichlet prior on the distributions leads to smoothing determined by the parameter alpha/eta

Document generation



Generating our corpus from the model



STEPS:

1. For each topic, draw a multinomial distribution *beta* (term distribution)
2. For each document, draw multinomial distribution *theta (topic distribution)*
3. For each word position, select a single topic $Z_{d,n}$ from distribution given by *theta*
4. Select word w_n from the term distribution given by *beta*

Need to infer:

- Per-word topic assignment
- Per-document topic proportions
- Per-corpus topic distributions

Inference for parameter estimation

Exact inference using EM style algos is not tractable for the LDA formulation

- E step has the intractable expectation

$$p(\vec{\theta}_{1:D}, z_{1:D,1:N}, \vec{\beta}_{1:K} \mid w_{1:D,1:N}, \alpha, \eta) = \frac{p(\vec{\theta}_{1:D}, \vec{z}_{1:D}, \vec{\beta}_{1:K} \mid \vec{w}_{1:D}, \alpha, \eta)}{\int_{\vec{\beta}_{1:K}} \int_{\vec{\theta}_{1:D}} \sum_{\vec{z}} p(\vec{\theta}_{1:D}, \vec{z}_{1:D}, \vec{\beta}_{1:K} \mid \vec{w}_{1:D}, \alpha, \eta)}.$$

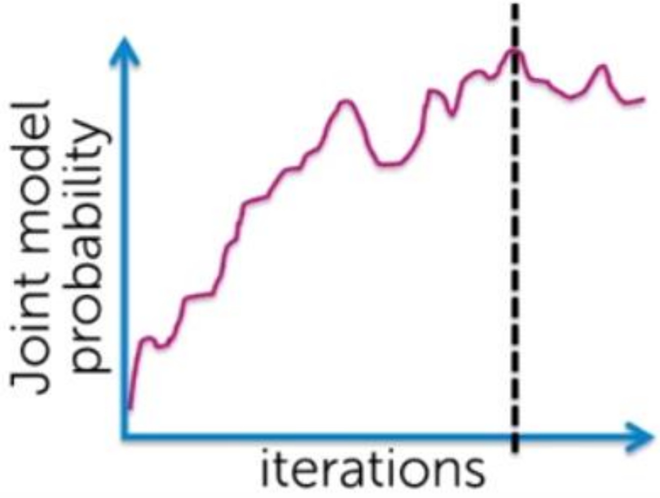
There are various approximation methods -

- Variational EM
- Expectation propagation
- Collapsed variational inference
- Gibbs sampling
- Collapsed Gibbs sampling

Parameter estimation using collapsed Gibbs sampling

A form of MCMC (Markov Chains Monte Carlo)

- Assignment variables and model parameters treated same
- Iteratively perform hard assignment
- Keep updating the parameters with each iteration
- Joint model probability stabilizes (mostly) after some iterations
- Average the predictions for the final results



$$P(z_i = j \mid \mathbf{z}_{-i}, w_i, d_i, \cdot) \propto \frac{C^{WT}_{w_i j} + \beta}{\sum_{w=1}^{W} C^{WT}_{wj} + W\beta} \frac{C^{DT}_{d_i j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d_i t} + T\alpha}$$

$$\phi'^{(j)}_i = \frac{C^{WT}_{ij} + \beta}{\sum_{k=1}^{W} C^{WT}_{kj} + W\beta}$$

$$\theta'^{(d)}_j = \frac{C^{DT}_{dj} + \alpha}{\sum_{k=1}^{T} C^{DT}_{dk} + T\alpha}$$

Measuring document similarity

- Cosine similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- KL divergence

$$D_{\mathrm{KL}}(P\|Q) = -\sum_i P(i) \log \frac{Q(i)}{P(i)},$$

- Jensen Shannon where

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M)$$

$$M = \frac{1}{2}(P + Q)$$

- Hellinger distance

$$\text{document-similarity}_{d,f} = \sum_{k=1}^{K} \left(\sqrt{\widehat{\theta}_{d,k}} - \sqrt{\widehat{\theta}_{f,k}} \right)^2$$

LDA hands on

More approaches to topic modeling

NMF - Non-negative Matrix Factorization

Extensions to LDA -

- Hierarchal - num of topics inferred
- Infinitely nested - gives topic hierarchy as a tree
- Dynamic LDA (let topics evolve over time)

Practical considerations, endnotes

Unsupervised learning, so human judgement essential

- Often a good idea to check with business if topics make intuitive sense
- There are ‘metrics’ that try to capture human interpretability

PLSA vs LDA

- LDA more generalised form of PLSA, and more elegant
- PLSA overfits on small data (too many parameters!)
- PLSA and LDA perform similarly on large datasets

Ideas discussed

- The notion of a topic
- Topic discovery and its utility
- Unsupervised learning: human evaluation needed
- High level intuition: words enough to assess topic
- Defining the tasks of a topic model: input and output
- Simplest model: topic as term, drawbacks. Need for topic as a distribution over terms
- Matrix factorization approach
- Probabilistic topic models
    - Unigram model
- Plate notation: concise way of visually representing models
- PLSA formulation
- EM algorithm, application to PLSA
- Multinomial distributions and the Dirichlet distribution

Summarizing our learning

- Topic models an important approach to understanding the text
- Simple unigram language models sufficient for topic discovery
- Probabilistic models are a necessary and powerful representation/construct
- PLSA and LDA are two methods
- Unsupervised learning; best to include humans in the loop
    - Several attempts towards metrics that capture human judgement, but nothing better than to include humans

Appendix

THANK YOU