LDA with Gibbs Sampling

Group 3

Introduction

Evolution of topic models:
TF-IDF → LSA → *pLSA* → LDA

TF-IDF is word-based, not topic-based

LSA leverages TF-IDF to identify weights of words, which are then compressed into set number of topics

Latent Dirichlet Allocation - probabilistic approach

LDA is a probabilistic topic model, departing from LSA and TF-IDF

	LSA	**LDA** (Blei et al. (2003))
Approach	Matrix factorization	Probabilistic
Generative?	No, not able to apply to “unseen” documents	Yes, can be applied to “unseen” documents
Algorithm	Harder to implement	Easier to implement

LSA & LDA shared assumptions:

1) bag of words
2) order of documents is not significant

LDA origins

The paper "Latent Dirichlet Allocation" was published by Blei, Ng and Jordan in the Journal of Machine Learning Research in 2003

Journal of Machine Learning Research 3 (2003) 993-1022 Submitted 2/02; Published 1/03

Latent Dirichlet Allocation

David M. Blei BLEI@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720, USA

Andrew Y. Ng ANG@CS.STANFORD.EDU
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA

Editor: John Lafferty

Abstract

We describe *latent Dirichlet allocation* (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. We present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation. We report results in document modeling, text classification, and collaborative filtering, comparing to a mixture of unigrams model and the probabilistic LSI model.

1. Introduction

In this paper we consider the problem of modeling text corpora and other collections of discrete data. The goal is to find short descriptions of the members of a collection that enable efficient processing of large collections while preserving the essential statistical relationships that are useful for basic tasks such as classification, novelty detection, summarization, and similarity and relevance judgments.

Significant progress has been made on this problem by researchers in the field of information retrieval (IR) (Baeza-Yates and Ribeiro-Neto, 1999). The basic methodology proposed by IR researchers for text corpora—a methodology successfully deployed in modern Internet search engines—reduces each document in the corpus to a vector of real numbers, each of which represents ratios of counts. In the popular *tf-idf* scheme (Salton and McGill, 1983), a basic vocabulary of "words" or "terms" is chosen, and, for each document in the corpus, a count is formed of the number of occurrences of each word. After suitable normalization, this term frequency count is compared to an inverse document frequency count, which measures the number of occurrences of a

LDA was independently invented for use in population genetics research by Pritchard, Stephens and Donnelly in 2000

Inference of Population Structure Using Multilocus Genotype Data

Jonathan K. Pritchard, Matthew Stephens and Peter Donnelly
Department of Statistics, University of Oxford, Oxford OX1 3TG, United Kingdom

Manuscript received September 23, 1999
Accepted for publication February 18, 2000

ABSTRACT

We describe a model-based clustering method for using multilocus genotype data to infer population structure and assign individuals to populations. We assume a model in which there are K populations (where K may be unknown), each of which is characterized by a set of allele frequencies at each locus. Individuals in the sample are assigned (probabilistically) to populations, or jointly to two or more populations if their genotypes indicate that they are admixed. Our model does not assume a particular mutation process, and it can be applied to most of the commonly used genetic markers, provided that they are not closely linked. Applications of our method include demonstrating the presence of population structure, assigning individuals to populations, studying hybrid zones, and identifying migrants and admixed individuals. We show that the method can produce highly accurate assignments using modest numbers of loci—*e.g.*, seven microsatellite loci in an example using genotype data from an endangered bird species. The software used for this article is available from http://www.stats.ox.ac.uk/~pritch/home.html.

IN applications of population genetics, it is often useful to classify individuals in a sample into populations. In one scenario, the investigator begins with a sample of individuals and wants to say something about the properties of populations. For example, in studies of human evolution, the population is often considered to be the unit of interest, and a great deal of work has focused on learning about the evolutionary relationships of modern populations (*e.g.*, Cavalli *et al.* 1994). In a second scenario, the investigator begins with a set of predefined populations and wishes to classify individuals of unknown origin. This type of problem arises in many contexts (reviewed by Davies *et al.* 1999). A standard approach involves sampling DNA from members of a number of potential source populations and using these samples to estimate allele frequencies in each population at a series of unlinked loci. Using the estimated allele frequencies, it is then possible to compute the likelihood that a given genotype originated in each population. Individuals of unknown origin can be assigned to populations according to these likelihoods Paetkau *et al.* 1995; Rannala and Mountain 1997).

In both situations described above, a crucial first step is to define a set of populations. The definition of populations is typically subjective, based, for example, on linguistic, cultural, or physical characters, as well as the geographic location of sampled individuals. This subjective approach is usually a sensible way of incorporating diverse types of information. However, it may be difficult to know whether a given assignment of individuals to populations based on these subjective criteria represents a natural assignment in genetic terms, and it would be useful to be able to confirm that subjective classifications are consistent with genetic information and hence appropriate for studying the questions of interest. Further, there are situations where one is interested in "cryptic" population structure—*i.e.*, population structure that is difficult to detect using visible characters, but may be significant in genetic terms. For example, when association mapping is used to find disease genes, the presence of undetected population structure can lead to spurious associations and thus invalidate standard tests (Ewens and Spielman 1995). The problem of cryptic population structure also arises in the context of DNA fingerprinting for forensics, where it is important to assess the degree of population structure to estimate the probability of false matches (Balding and Nichols 1994, 1995; Foreman *et al.* 1997; Roeder *et al.* 1998).

Pritchard and Rosenberg (1999) considered how genetic information might be used to detect the presence of cryptic population structure in the association mapping context. More generally, one would like to be able to identify the actual subpopulations and assign individuals (probabilistically) to these populations. In this article we use a Bayesian clustering approach to tackle this problem. We assume a model in which there are K populations (where K may be unknown), each of which is characterized by a set of allele frequencies at each locus. Our method attempts to assign individuals to populations on the basis of their genotypes, while simultaneously estimating population allele frequencies. The method can be applied to various types of markers [*e.g.*, microsatellites, restriction fragment length polymorphisms (RFLPs), or single nucleotide polymorphisms (SNPs)], but it assumes that the marker

Corresponding author: Jonathan Pritchard, Department of Statistics, University of Oxford, 1 S. Parks Rd., Oxford OX1 3TG, United Kingdom. E-mail: pritch@stats.ox.ac.uk

Genetics **155:** 945–959 (June 2000)

What is LDA?

- **Latent Dirichlet Allocation**
  - Words are the only observable variables, all others are **latent** variables
  - Leverages **Dirichlet** distributions
  - **Allocates** the words of the document to different topics
- Generative Statistical Model for Topic Modeling
  - Imagine how the documents were created and reverse engineer generation
- Documents contain multiple topics, but probably not all of them

Generative Process

Each **topic** is represented as a distribution over a fixed vocabulary

e.g. a *genetics* topic would have a high probability of containing words about genetics

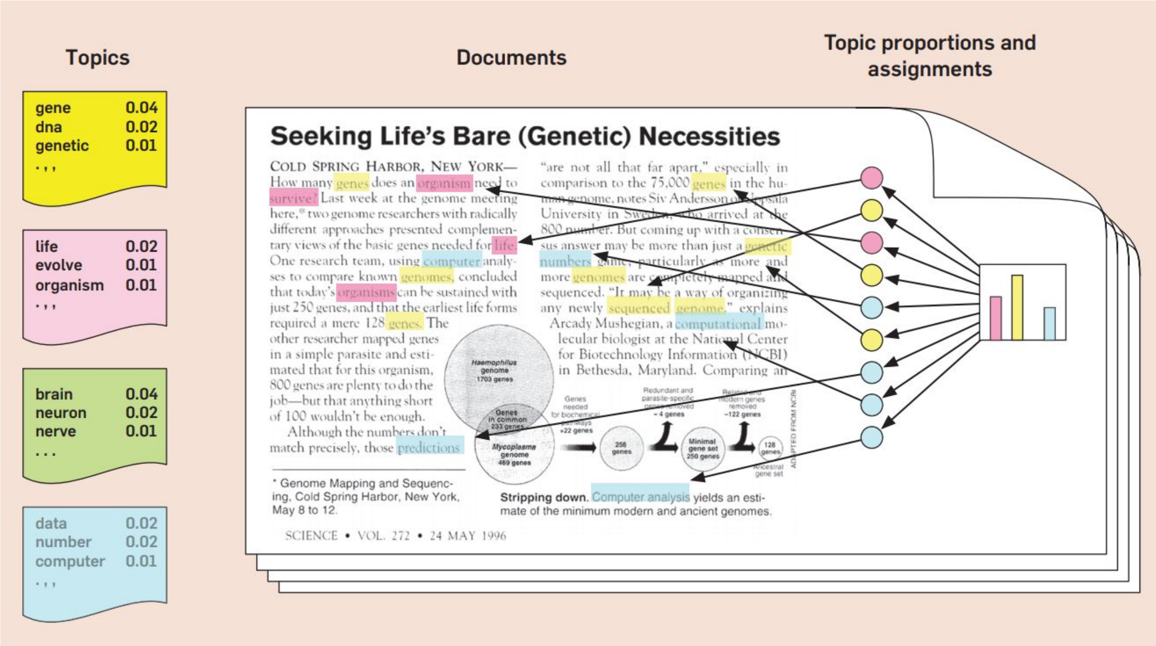
Each **document** has a topic distribution

e.g. an article about genetic data analysis would have a high probability for the topics *genetics* and *data analysis*

Each **word** in a document is chosen from a topic

- the topic *genetics* might be chosen with a high probability from the topic distribution for the document
- the word *gene* might be chosen from the topic *genetics* with high probability

Generative Process

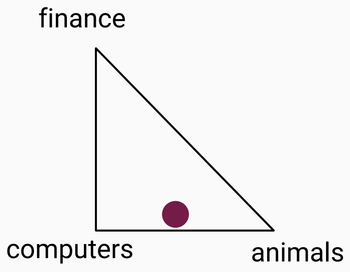


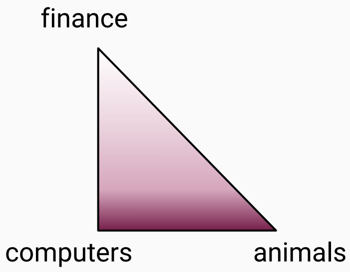
What kind of distribution?

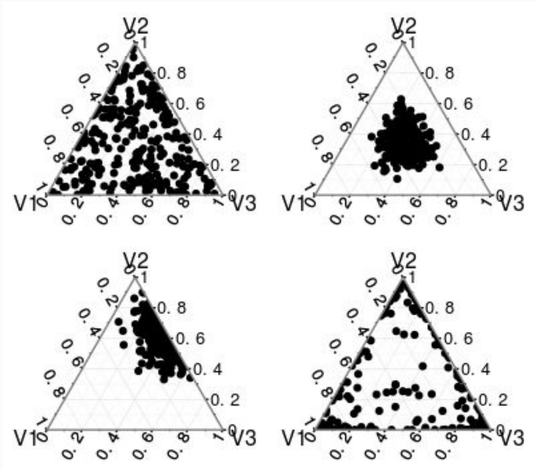
Dirichlet Distribution

Distribution over multinomial distributions

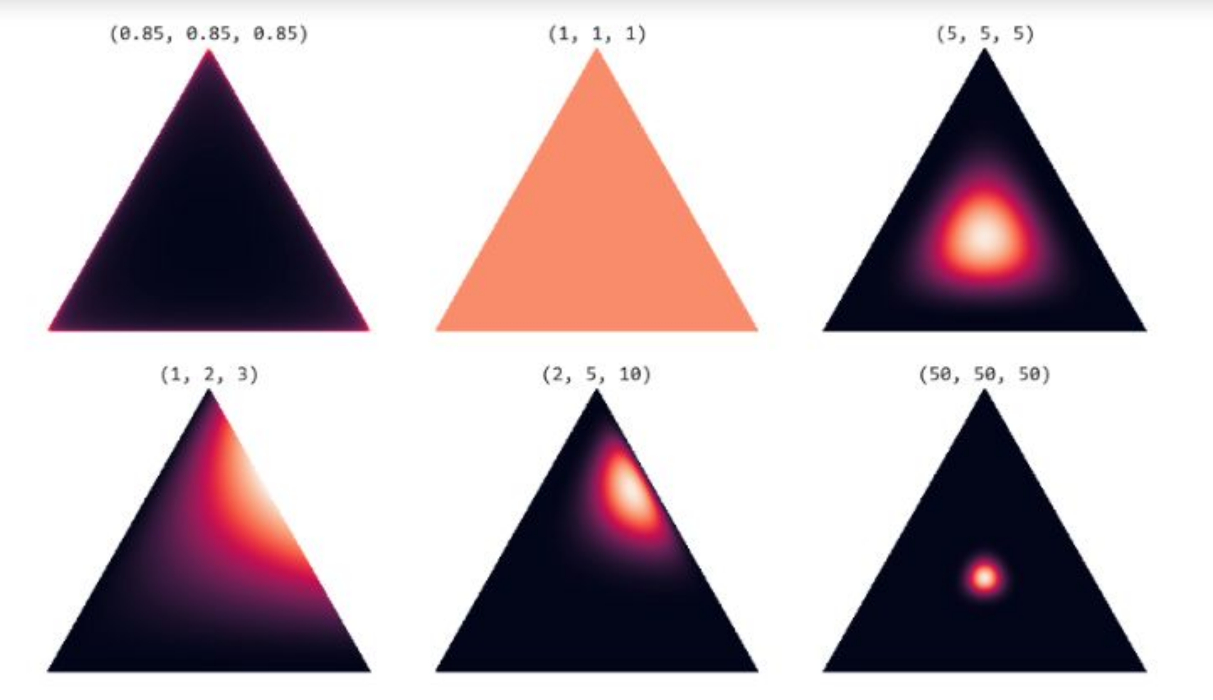
P(word = mouse) = (½, ½, 0)







Concentration parameter



Inference

To “generate” a new document, we need to know:

1. **Topic distribution** for the document
2. **Word distributions** for each topic

We don’t know the distributions, so we have to **infer** them from training data

That’s computationally expensive!

Sample from the distribution to iteratively approximate the values

How do we sample?

Gibbs Sampling

Approximate joint distributions for latent variables to sample from

Remove one value for a latent variable, then calculate new joint distribution conditioned on other values, then randomly sample from that distribution

E.g. For each word, unassign a topic, compute a new joint distribution for that word and each topic based on all other words with topic assignments in the document, choose a new topic assignment from the distribution

Update priors based on observations

High Level Overview of Algorithm

Input:

- Set of documents, made up of words
- Number of topics to find

Learning:

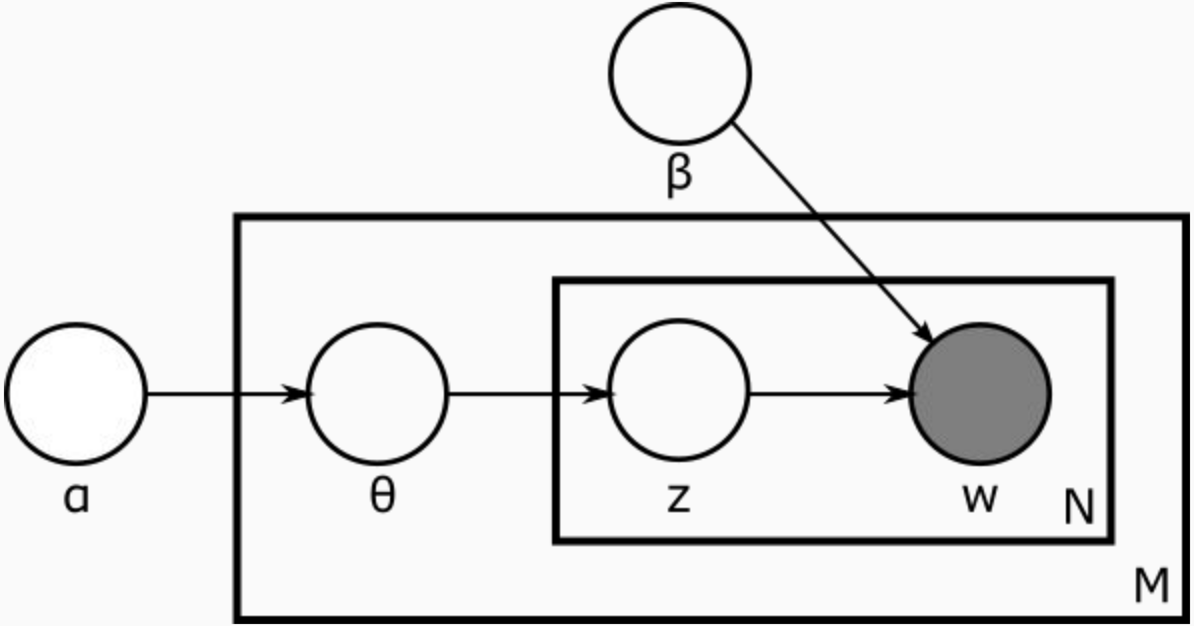
- Initialize topic distributions and topic-word distributions (usually randomly)
- Using a Gibbs Sampler, iteratively sample and update
- Update priors until stop condition

Output:

- Topic distributions and topic-word distributions

Algorithm

LDA Generation Plate Notation



Generative Process: the assumption

Document
word word word word word word word
word word word word word word word
word word word word word word word
word word word word word word word

If we have a document of a certain length:

And we want it to be 60% about fashion

and 40% about business:

Generative Process: the assumption

If we have a list of fashion words and a list of business words:

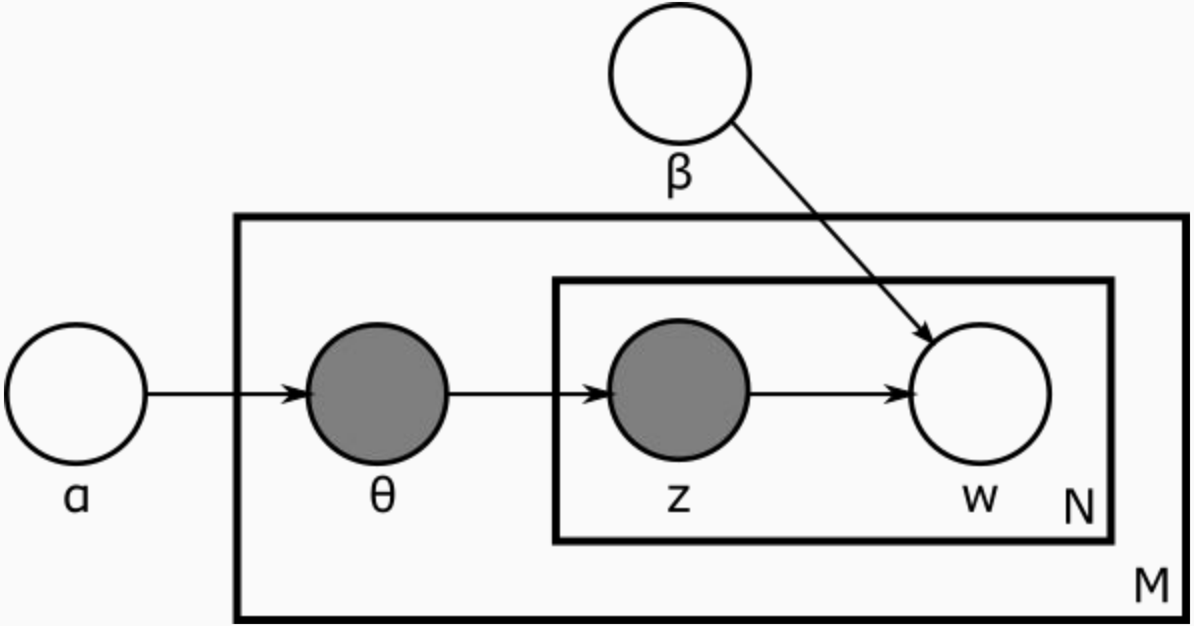
Fashion	Business
clothes shoes	market value
design style	company profit
vogue trend	trend merger
popular hip	revenue sell

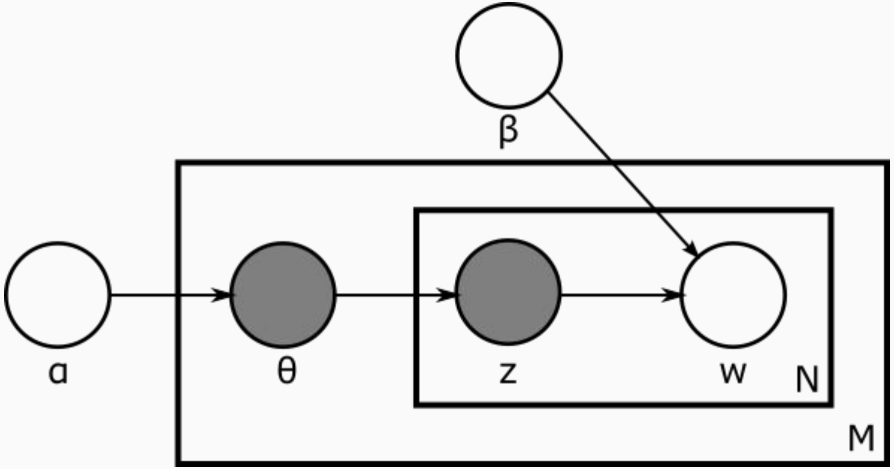
We could generate a document using 60% fashion words and 40% business words:

Document

trend hip clothes clothes profit hip merger hip market clothes trend popular trend style trend hip design profit company style

LDA Inference Plate Notation





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

Gibbs Sampling

- Markov Chain Monte Carlo (MCMC) algorithm
- Sample from distributions with two or more dimensions
- When the conditional probabilities can be calculated, estimate joint probabilities.

Simple 2-D Gibbs example

	mouse	horse	money	market
finance	0	1	4	5
computers	5	0	2	1
animals	3	3	0	2

	mouse	horse	money	market
finance	0	$\frac{1}{10}$	$\frac{2}{5}$	$\frac{1}{2}$
computers	$\frac{5}{8}$	0	$\frac{1}{4}$	$\frac{1}{8}$
animals	$\frac{3}{8}$	$\frac{3}{8}$	0	$\frac{1}{4}$

$P(word|topic)$

	mouse	horse	money	market
finance	0	$\frac{1}{4}$	$\frac{2}{3}$	$\frac{5}{8}$
computers	$\frac{5}{8}$	0	$\frac{1}{3}$	$\frac{1}{8}$
animals	$\frac{3}{8}$	$\frac{3}{4}$	0	$\frac{1}{4}$

$P(topic|word)$

	mouse	horse	money	market
finance	0	$\frac{1}{4}$	$\frac{2}{3}$	$\frac{5}{8}$
computers	$\frac{5}{8}$	0	$\frac{1}{3}$	$\frac{1}{8}$
animals	$\frac{3}{8}$	$\frac{3}{4}$	0	$\frac{1}{4}$

$$P(topic|word)$$

	mouse	horse	money	market
finance	0	$\frac{1}{10}$	$\frac{2}{5}$	$\frac{1}{2}$
computers	$\frac{5}{8}$	0	$\frac{1}{4}$	$\frac{1}{8}$
animals	$\frac{3}{8}$	$\frac{3}{8}$	0	$\frac{1}{4}$

$$P(word|topic)$$

	mouse	horse	money	market
finance	0	$\frac{1}{4}$	$\frac{2}{3}$	$\frac{5}{8}$
computers	$\frac{5}{8}$	0	$\frac{1}{3}$	$\frac{1}{8}$
animals	$\frac{3}{8}$	$\frac{3}{4}$	0	$\frac{1}{4}$

$$P(topic|word)$$

	mouse	horse	money	market
finance	0	$\frac{1}{10}$	$\frac{2}{5}$	$\frac{1}{2}$
computers	$\frac{5}{8}$	0	$\frac{1}{4}$	$\frac{1}{8}$
animals	$\frac{3}{8}$	$\frac{3}{8}$	0	$\frac{1}{4}$

$$P(word|topic)$$

1. Randomly initialize each x_i

2. For $t = 1, ..., T$:

   2.1 $x_1^{t+1} \sim p(x_1 | x_2^{(t)}, x_3^{(t)}, ..., x_m^{(t)})$

   2.2 $x_2^{t+1} \sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, ..., x_m^{(t)})$

   2.m $x_m^{t+1} \sim p(x_m | x_1^{(t+1)}, x_2^{(t+1)}, ..., x_{m-1}^{(t+1)})$

Inference

For all the words in every document, start out by assigning each one to a topic randomly:

Document
cat dog animal dog animal cat cat

Document
apple pie ingredient apple pie flour dough

Document
market analyst invest invest price market

Topic 1	cat cat apple apple dough invest market
Topic 2	dog animal cat pie pie market invest
Topic 3	animal dog ingredient flour analyst price

Inference

Count the number of times each word occurs with each topic...

	cat	dog	animal	apple	pie	market	...
Topic 1	2	0	0	2	0	1	
Topic 2	1	1	1	0	2	1	
Topic 3	0	1	0	0	0	0	

Inference

And count the number of times words from each document occur with each topic.

	Document 1 words	Document 2 words	Document 3 words
Topic 1	2	3	2
Topic 2	3	2	2
Topic 3	2	2	2

The Algorithm

For every document *d*:

For every word *w* in the document:

For every topic *t*:

$$P(t) \sim \frac{\text{\# of times } w \text{ occurs in that topic } (+\beta)}{\text{\# of times } w \text{ occurs in that topic } + \text{\# of unique words} * \beta} * \frac{\text{\# of words in document in that topic } (+\alpha)}{\text{\# of words} + \text{\# of topics} * \alpha}$$

The Algorithm

For every document *d*:

Document
cat dog animal dog animal cat cat

For every word *w* in the document: cat

For every topic *t*:

Topic 1	~~cat~~ cat apple apple dough invest market

$$P(t) \sim \frac{\text{\# of times } w \text{ occurs in that topic } (+\beta) \quad 1+\beta}{\text{\# of times } w \text{ occurs in that topic + \# of unique words} * \beta \quad 1 + 12 * \beta} * \frac{\text{\# of words in document in that topic } (+\alpha) \quad 1+\alpha}{\text{\# of words + \# of topics} * \alpha \quad 12 + 3 * \alpha}$$

=0.05555

The Algorithm

Document

cat dog animal dog animal

cat cat

cat

Alpha = 0.5
Beta = 0.1

Topic 1 ~ 0.05555
Topic 2 ~ 0.12963
Topic 3 ~ 0.01543

$P(t = 1 \mid w, d, z) = (0.05555) / (0.05555 + 0.12963 + 0.01543)$
$= 0.27691$
$P(t = 2 \mid \ldots) = 0.64618$
$P(t = 3 \mid \ldots) = 0.07692$

The Algorithm

Randomly sample from this distribution:

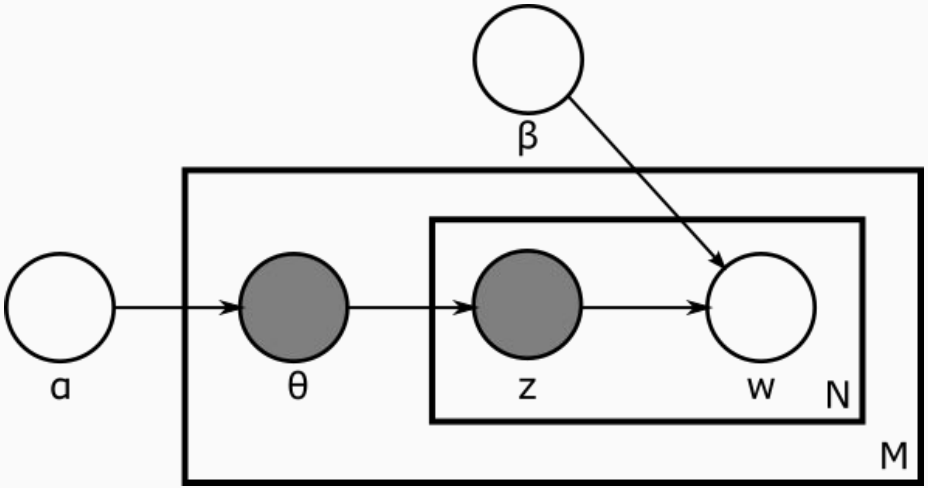
The Algorithm

Reassign the word to the new topic. Do this over and over and over again.

Topic 1	~~cat~~ cat apple apple dough invest market
Topic 2	**cat** dog animal cat pie pie market invest
Topic 3	animal dog ingredient flour analyst price

Resulting Text Representation

Documents, which are a bag of words, are represented as a mixture of topics from which words can be sampled from multinomial distributions.



Performance

Intrinsic evaluation metrics

Hold-out perplexity

$$\hat{H} = -\frac{1}{m}\log_2 P(w_1, w_2, \ldots, w_m)$$

$$PP = 2^{\hat{H}}$$

Coherence (PMI, NPMI)

$$\mathrm{pmi}(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)}$$

$$\mathrm{npmi}(x;y) = \frac{\mathrm{pmi}(x;y)}{h(x,y)}$$

Extrinsic evaluation metrics

If your gold standard data are labeled, accuracy can be measured directly by use of downstream algorithms (SVM, clustering, etc).

Multiclass classification:
cross-entropy

Clustering: B-cubed, F-measure, etc.

Computational complexity

Inference is $\boldsymbol{\Theta}$(k * |d| * |V|) where k is number of topics, d is the set of documents, V is the vocab.

For a large number of topics, some research suggests that the problem is NP-hard.

Demo

Go to Jupyter notebook

Application

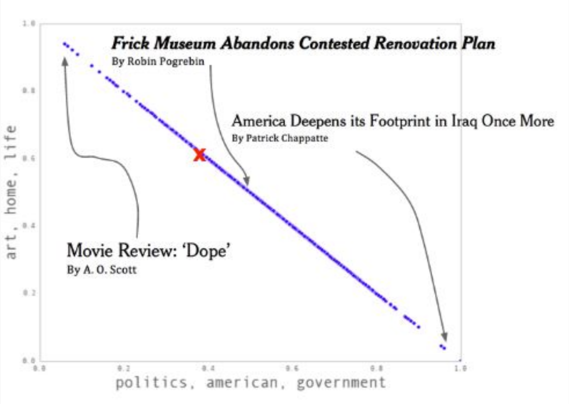
LDA for recommending NYT articles

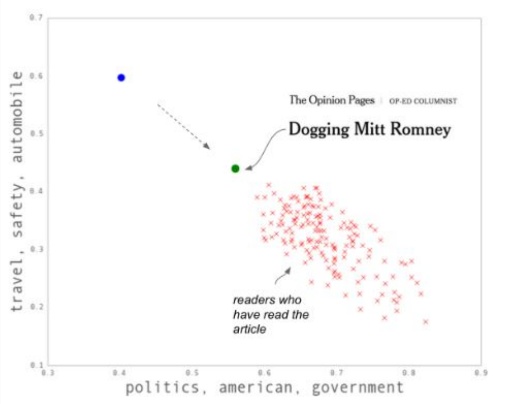
User: distribution of topics they're interested in

Article: distribution of topics/words

Adjust topic distributions based on reader preferences

- Add offsets to model topic error, incorporate reading patterns
- Iteratively adjust offsets and then recalculate reader scores





Grade of Membership (GoM) Models: Population genetics equivalent to LDA

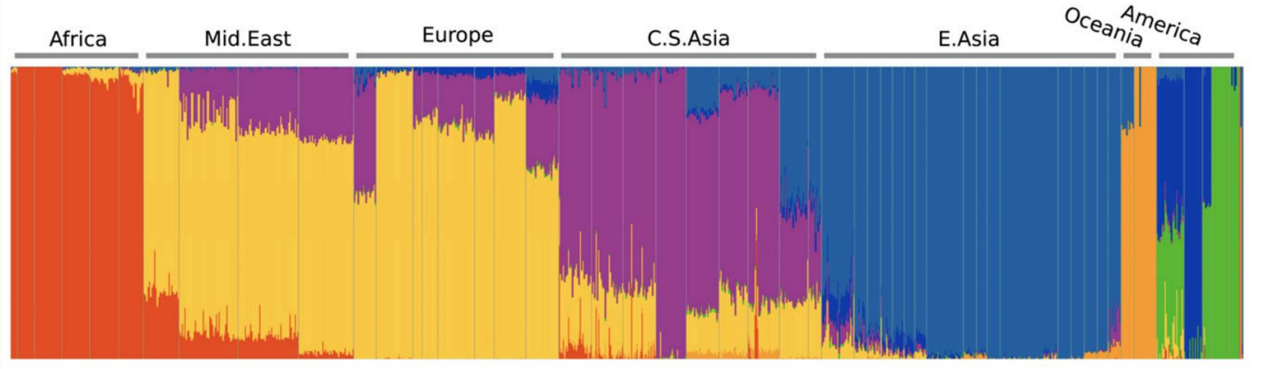


Image from Raj, Stephens and Pritchard, "fastSTRUCTURE: Variational Inference of Population Structure in Large SNP Data Sets" Genetics (2014)

populations
=
topics

DNA segments
=
documents

genes
=
words

Questions?

References

- https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
- https://en.wikipedia.org/wiki/Gibbs_sampling
- http://www.cs.columbia.edu/~blei/papers/WangBlei2011.pdf
- https://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/
- https://wiseodd.github.io/techblog/2017/09/07/lda-gibbs/
- http://www.cs.columbia.edu/~blei/papers/Blei2012.pdf
- http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf
- http://www.genetics.org/content/197/2/573
- https://stats.stackexchange.com/questions/244917/what-exactly-is-the-alpha-in-the-dirichlet-distribution
- https://www.youtube.com/watch?v=yK7nN3FcgUs&feature=youtu.be
- https://www.youtube.com/watch?v=FkckgwMHP2s
- https://people.cs.umass.edu/~wallach/publications/wallach09evaluation.pdf
- http://cseweb.ucsd.edu/~dhu/docs/exam09.pdf

References

- http://www1.icsi.berkeley.edu/Speech/docs/HTKBook3.2/node188_mn.html
- https://en.wikipedia.org/wiki/Pointwise_mutual_information
- https://towardsdatascience.com/dirichlet-distribution-a82ab942a879

Credits

Aidan: Generative process, inference, algorithm slides

Elijah: Initial intro, performance, demo, presentation

Julia: Intro, Applications - general NLP & applications beyond linguistics

Kevin: Gibbs algorithm example and algorithm overview slides

Zoe: Intro/General Overview, NLP applications, further applications

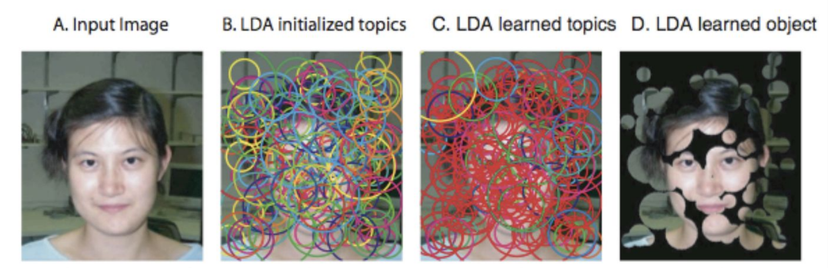
Appendix

NLP Applications for LDA

NLP Task	Explanation
Similarity/recommendations	Find related documents by comparing topic weight vectors (cf. NYT example above)
Search	In search engines, return more "topical" results first; SEO
Word sense disambiguation	In cases where multiple word meanings are possible, use LDA to suggest most likely meaning based on text/document topics
Machine translation	Similar to WSD - when multiple translations are possible for one word, use LDA to suggest most likely translation based on text/document topics
Corpus exploration	Find topic clusters in large corpora of literary texts, archives, etc. Popular in digital humanities research (e.g. this blog)

Additional applications for LDA - Images

document = image   word = codeword (patch of image)   topic = object



In A - D above, the object learned is a person's face, but multiple objects could be learned in a single image, much like multiple topics could be learned in a single document

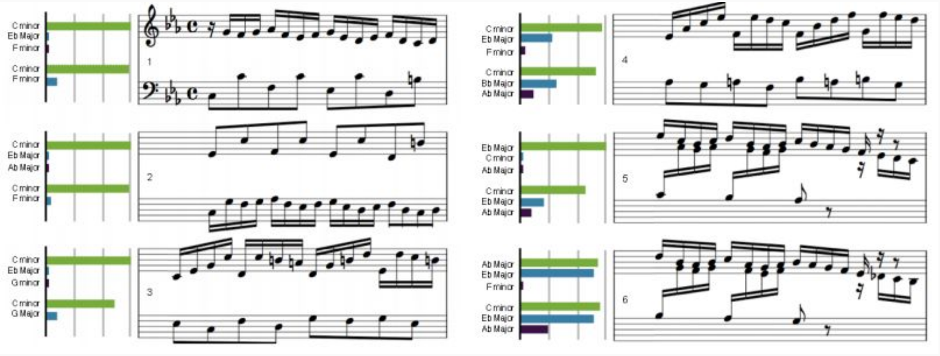
Before NN / deep learning era, LDA was a popular approach for image clustering, image retrieval and image relevance ranking

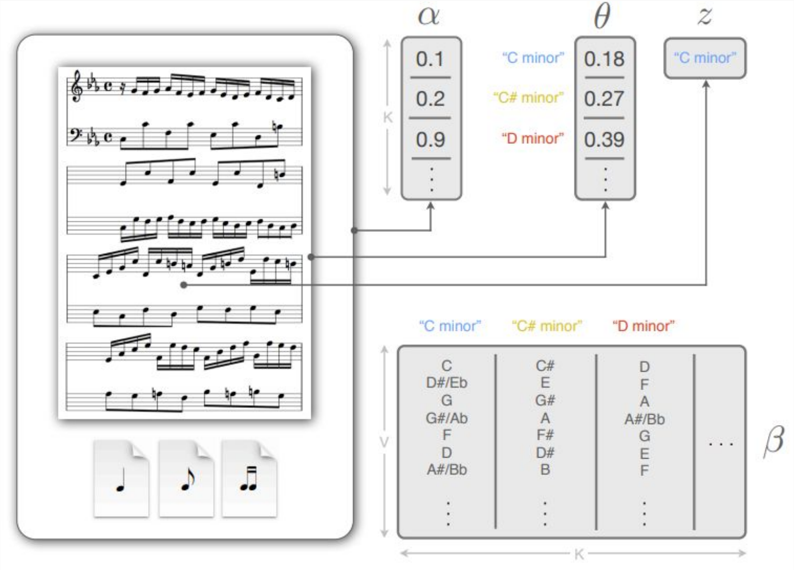
Sources: http://cseweb.ucsd.edu/~dhu/docs/exam09.pdf, http://pages.cs.wisc.edu/~pradheep/Clust-LDA.pdf

Additional applications for LDA - Music

document = **song** **word** = **note** **topic** = **key**

Key finding: find a key for a song
Modulation Tracking: find a key for a segment





Source: http://cseweb.ucsd.edu/~dhu/docs/exam09.pdf