

LATENT DIRICHLET ALLOCATION

Giuseppe Di Benedetto, Jack Jewson, Kaspar Märtens and Qinyi Zhang

29th January 2016

TOPIC MODELING

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to **survive**? Last week at the genome meeting here,* two genome researchers with radically different approaches complemented views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough. Although the numbers don't match precisely, those **predictions**

"are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson, a physicist at Uppsala University in Sweden who arrived at the 800 number. But coming up with a consensus answer may be more than just a **genetic** numbers game, particularly as more and more **genomes** are completely mapped and sequenced. "It may be a way of organizing any newly **sequenced genome**," explains Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



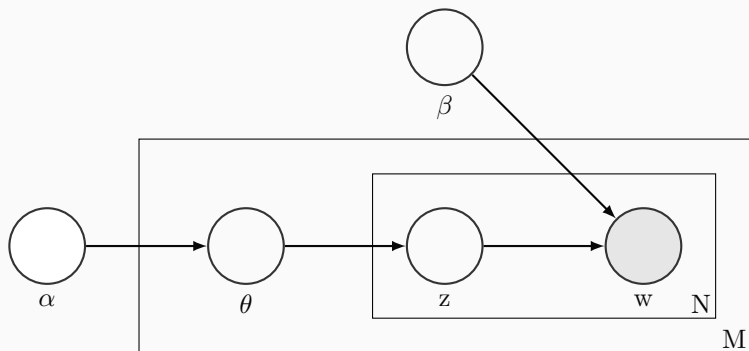
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

Topic proportions and assignments

GRAPHICAL MODEL FOR LDA



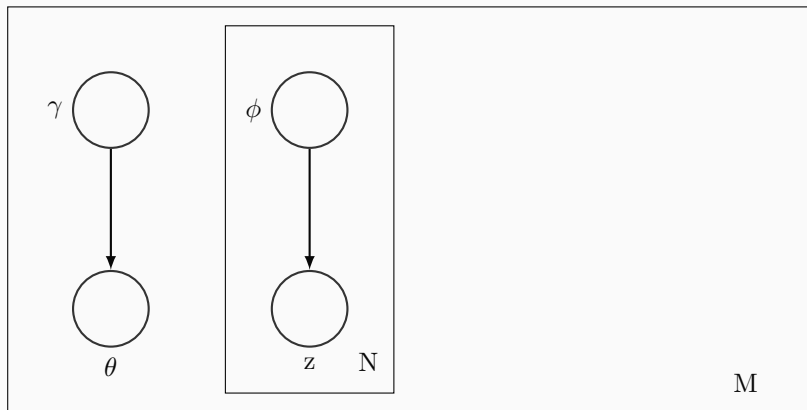
THE “BAG OF WORDS” AND EXCHANGABILITY

Assume documents are a “Bag of Words” leading to exchangeability in the distribution of words in a document

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

Documents within the corpus are also assumed to be exchangeable

VARIATIONAL EM



Likelihood, $p(\mathbf{w} | \alpha, \beta)$, required for inference on hidden parameters but intractable in this context.

Use approximation from statistical physics to bound the log-likelihood

$$\begin{aligned}\log p(\mathbf{w} | \alpha, \beta) &\geq E_q[\log p(\theta, z, \mathbf{w} | \alpha, \beta)] - E_q[\log q(\theta, z | \gamma, \phi)] \\ &=: \mathcal{L}(\gamma, \phi; \alpha, \beta)\end{aligned}$$

$$\log p(\mathbf{w} | \alpha, \beta) = \mathcal{L}(\gamma, \phi; \alpha, \beta) + D(q(\theta, z | \gamma, \phi) || p(\theta, z | \mathbf{w}, \alpha, \beta))$$

```
repeat
  # E-step
  for  $d$  in documents do
    repeat
      update  $\phi_d$  (loop over all words and topics)
      update  $\gamma_d$ 
      compute  $\mathcal{L}_d$ 
    until convergence (i.e. relative change in  $\mathcal{L}_d$  is less than  $\varepsilon$ )
  # M-step
  update  $\alpha$  (Newton's method)
  update  $\beta$ 
until convergence
```

Blei et al (2003) describe a scheme for updating α with Newton's method in linear time

$$\alpha^{t+1} = \alpha^t - \mathbf{H}(f(\alpha^t))^{-1} \nabla f(\alpha^t)$$

Blei et al (2003) describe a scheme for updating α with Newton's method in linear time

$$\alpha^{t+1} = \alpha^t - \mathbf{H}(f(\alpha^t))^{-1} \nabla f(\alpha^t)$$

However,

1. They provide implementation with $\alpha_i = \alpha_j$
2. It is better to optimize α on log-scale
3. On the log-scale, we cannot invert the Hessian in linear time

Dataset of 1500 NIPS papers from 1988 to 1999, split into 90% train and 10% test portions.

Dataset of 1500 NIPS papers from 1988 to 1999, split into 90% train and 10% test portions.

Our vocabulary: 7605 words (out of total 12419 words, appearing more than 20 times).

Dataset of 1500 NIPS papers from 1988 to 1999, split into 90% train and 10% test portions.

Our vocabulary: 7605 words (out of total 12419 words, appearing more than 20 times).

We constructed a *document* \times *term* matrix, used as input to our R function.

	network	model	learning	function	input	neural	set	algorithm	system	data
[1,]	4	56	4	1	6	1	5	6	24	26
[2,]	25	71	6	5	1	15	15	0	3	40
[3,]	0	11	20	9	4	1	4	6	15	10
[4,]	18	0	36	8	5	1	1	18	3	1
[5,]	6	39	67	12	7	7	1	1	10	0

EXAMPLE IMPLEMENTATION

Topic 1	Topic 2	Topic 4	Topic 6	Topic 8
network	function	model	network	object
system	learning	data	training	visual
model	network	distribution	unit	model
learning	weight	gaussian	input	image
neural	error	parameter	hidden	motion
control	algorithm	algorithm	set	field
input	result	function	error	direction
dynamic	set	method	output	unit
output	neural	mean	neural	map
recurrent	number	component	data	position
rules	parameter	probability	weight	system
rule	case	likelihood	learning	eye
attractor	input	density	performance	images
point	bound	mixture	layer	representation
trajectory	training	matrix	model	view

EXAMPLE IMPLEMENTATION

$$\gamma(w_{18}) =$$

(601.97, 556.37, 0.06, 0.06, 0.06, 366.60, 0.06, 0.06, 99.42, 32.93)

This paper provides a systematic analysis of the recurrent backpropagation (RBP) algorithm, introducing a number of new results. The main limitation of the RBP algorithm is that it assumes the convergence of the network to a stable fixed point in order to backpropagate the error signals. We show by experiment and eigenvalue analysis that this condition can be violated and that chaotic behavior can be avoided. Next we examine the advantages of RBP over the standard backpropagation algorithm. RBP is shown to build stable fixed points corresponding to the input patterns. This makes it an appropriate tool for content addressable memories, one-to-many function learning, and inverse problems.

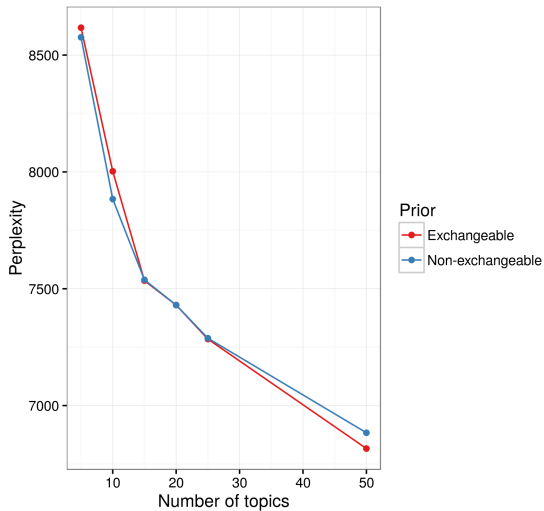
DOCUMENT MODELING - PERPLEXITY

For an unseen set of M documents,

$$\text{perplexity}(\mathcal{D}_{test}) = \exp \left\{ - \frac{\sum_{d=1}^M \log p(w_d | \alpha, \beta)}{\sum_{d=1}^M N_d} \right\} = \exp \left\{ - \frac{\mathcal{L}(\mathbf{w}_d)}{\sum_{d=1}^M N_d} \right\}$$

where N_d is the number of words in document d .

DOCUMENT MODELING - PERPLEXITY



COLLABORATIVE FILTERING

At random remove one word from an unseen document

Find 'optimal' ϕ and γ using the $(N_d - 1)$ words left in the unseen document

Use these to calculate the likelihood of the removed word based on the rest of the document

Using likelihood

$$p(w|w_{obs}) = \sum_z \beta_{z,w} \frac{\gamma(w_{obs})_z}{\sum_i \gamma(w_{obs})_i} \quad (1)$$

COLLABORATIVE FILTERING - AN EXAMPLE

e.g.

Random removed word "hand"

36th most probable word in topic 7 and the 51st most probable word in topic 1

$\gamma(w_{obs}) = (458.57, 0.06, 152.18, 0.06, 71.04, 4.13, 0.06, 530.90, 0.06, 55.51)$

92nd most likely word to be the final word

Number of intersections between 20 most likely words to be removed word and 50 most likely words for each topic

(12, 8, 6, 5, 10, 9, 8, 15, 7, 5)

STOCHASTIC VARIATIONAL EM

Define $\rho_t = (\tau_0 + t)^{-k}$, with $k \in (0.5, 1]$

Initialize λ randomly

for $t = 0$ to ∞ **do**

 # *E-step*:

 Initialize γ

repeat

 update ϕ (loop over all words and topics)

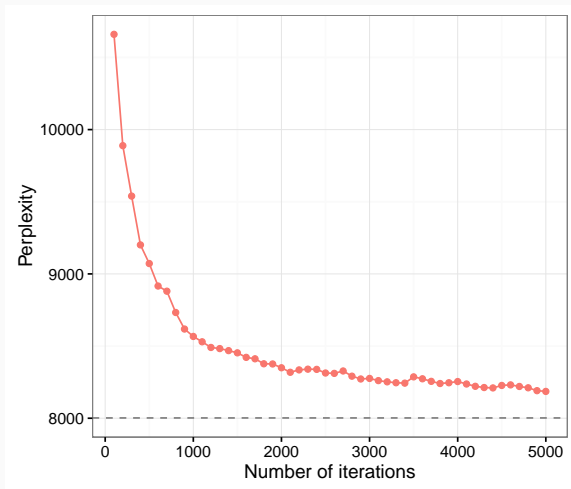
 update γ

until $\frac{1}{k} \sum_k |\text{change in } \gamma_{tk}| < \epsilon$

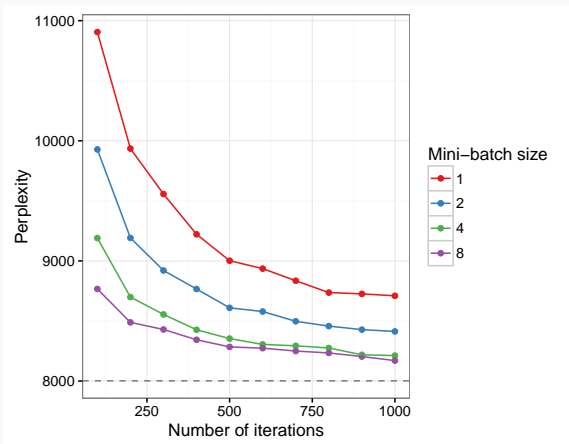
 # *M-step*:

 update λ

STOCHASTIC VARIATIONAL EM



STOCHASTIC VARIATIONAL EM



DISCUSSION AND FURTHER WORK

1. Choice for number of topics: Hierarchical Dirichlet Processes (Teh et al. 2006)
2. Measures of human interpretability for topic models (Chang et al. 2009)