

# Variational Inference

---

Michael Thomas Wojnowicz

June 11, 2021

Data Intensive Studies Center, Tufts University

# Table of contents

1. Overview
2. Variational Inference and Expectation Maximization
3. Coordinate Ascent Variational Inference (CAVI)
4. Example: Bayesian Mixture Model
5. Example: Latent Dirichlet Allocation (LDA)
6. Automatic Differentiation Variational Inference (ADVI)

## Some questions

- What is variational inference?
- When is it useful?
- Is it the same as variational bayes?
- Why is it called *variational* inference?
- What is Variational Expectation Maximization (VEM)? Variational Bayes Expectation Maximization (VBEM)?
- How can we apply VI to inference problems?

# Overview

---

# Overview

---

**The problem: marginalization**

# Parametric statistical models

## Parametric statistical models

A *parametric statistical model* posits

- $x$ : observed data
- $\theta$ : parameters
- $z$  (possibly): latent random variables

## Parameters vs. latent variables

Both  $z$  and  $\theta$  are unobserved, but only the dimensionality of  $z$  increases with the number of samples in  $x$ .

## Frequentist vs. Bayesian variants

Frequentists take parameters  $\theta$  to be fixed (but unknown) constants, whereas the Bayesians take  $\theta$  to be random variables.

# Three statistical modeling paradigms of interest

Let us consider models that present an **intractable marginal**.

## Bayesian latent variable models

Examples: Bayesian Mixture Model, Bayesian Hidden Markov Model, Latent Dirichlet Allocation, Bayesian nonparametric versions of the preceding

## Bayesian (non-latent variable) models

Examples: Non-conjugate models, Many hierarchical Bayesian models

## Frequentist latent variable models

Examples: Hidden Markov Models (although we have handled this case), Variational Autoencoders (the classical kind), Bayesian Generalized Linear Mixed Effects Models

# Statistical inference

## In general

We must compute the marginal

$$p(\mathbf{x} \mid \mathbf{c}) = \int p(\mathbf{x}, \mathbf{u} \mid \mathbf{c}) \, d\mathbf{u} \quad (1)$$

where

- $\mathbf{x}$ : observed data
- $\mathbf{u}$ : unobserved random variables
- $\mathbf{c}$ : constant values

# The need for marginalization in statistical inference

Model	Inferential goal	Target marginal
		$p(\mathbf{x}   \mathbf{c})$
Bayesian (non-latent)	$p(\boldsymbol{\theta}   \mathbf{x})$	$p(\mathbf{x}) = \int p(\boldsymbol{\theta}, \mathbf{x}) d\boldsymbol{\theta}$
Bayesian latent	$p(\mathbf{z}, \boldsymbol{\theta}   \mathbf{x})$	$p(\mathbf{x}) = \int p(\boldsymbol{\theta}, \mathbf{x}, \mathbf{z}) d\boldsymbol{\theta} dz$
Frequentist latent	$\text{argmax}_{\boldsymbol{\theta}} p(\mathbf{x}   \boldsymbol{\theta})$	$p(\mathbf{x}   \boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}   \boldsymbol{\theta}) dz$

# Problem: These marginalizations may be intractable

## Example: Hidden Markov Model

Define  $T$ : the state transition matrix

$\epsilon_j$ : the  $j$ th emission distribution,  $j = 1, \dots, k$

$\pi$ : the initial latent state distribution

$$\begin{aligned} p(\mathbf{x} \mid \theta) &= \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta) \\ &= \sum_{\mathbf{z}=(z_1, \dots, z_n)} p(\mathbf{x}, \mathbf{z} \mid \theta) \\ &= \sum_{\mathbf{z}=(z_1, \dots, z_n)} \pi_{z_1} \epsilon_{z_1}(x_1) T_{z_1, z_2} \epsilon_{z_2}(x_2) T_{z_2, z_3}, \dots, T_{z_{n-1}, z_n} \epsilon_{z_n}(x_n) \end{aligned}$$

has  $\mathcal{O}(n k^n)$  complexity. 

Consider e.g. that  $(k, n) = (5, 100) \rightarrow 10^{72}$  calculations.



# Overview

---

**The technique: functional optimization**

# Towards variational inference

We construct a lower bound on the target marginal.

## Variational Lower Bound (VLBO)

Let  $q$  be any probability density over  $\mathbf{u}$ . Then:

$$\begin{aligned}\ln p(\mathbf{x} \mid \mathbf{c}) &= \ln \int p(\mathbf{u}, \mathbf{x} \mid \mathbf{c}) d\mathbf{u} \\ &= \ln \int q(\mathbf{u}) \frac{p(\mathbf{u}, \mathbf{x} \mid \mathbf{c})}{q(\mathbf{u})} d\mathbf{u} \\ &\stackrel{\text{Jensen's}}{\geq} \int q(\mathbf{u}) \ln \left( \frac{p(\mathbf{u}, \mathbf{x} \mid \mathbf{c})}{q(\mathbf{u})} \right) d\mathbf{u} \\ &:= \text{VLBO}(q)\end{aligned}$$

# Variational Inference: Maximizing the VLBO

## Variational Inference

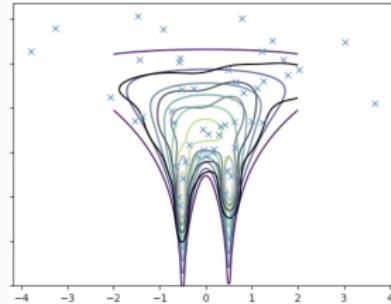
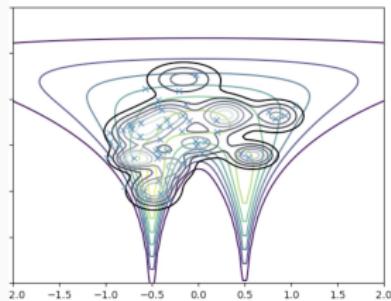
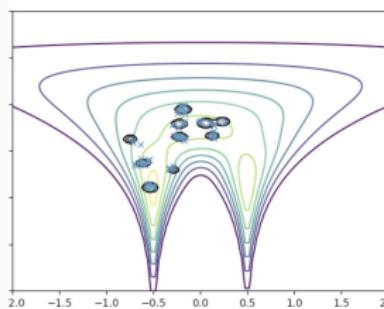
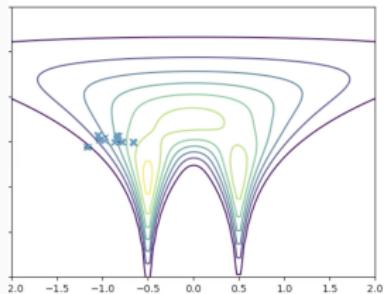
*Variational inference* (VI) proceeds by finding  $q^*$ , the variational density in tractable family  $\mathcal{Q}$  which maximizes the VLBO :

$$q^*_{\text{solution}} = \underset{\substack{q \in \mathcal{Q} \\ \text{approximating family}}}{\operatorname{argmax}} \text{VLBO}(q)$$

Rk: Note that we are trying to optimize over a function space (of a particular kind).

# Illustration

Here we approximate an probability distribution by finding the best approximation from tractable family  $\mathcal{Q} = \{10\text{-component Gaussian mixture models}\}$



# Overview

---

**Decompositions: Intuition on the cost function**

# Decompositions of the VLBO

## Energy/Entropy Decomposition of the VLBO

By simply appealing to properties of the logarithm and the definition of expectation, we obtain

$$\begin{aligned} \text{VLBO}(q) &= \int q(\mathbf{u}) \ln p(\mathbf{x}, \mathbf{u} \mid \mathbf{c}) d\mathbf{u} - \int q(\mathbf{u}) \ln q(\mathbf{u}) d\mathbf{u} \\ &= \underset{\text{energy}}{\mathbb{E}_q [\log p(\mathbf{x}, \mathbf{u} \mid \mathbf{c})]} + \underset{\text{entropy}}{\mathbb{H}[q(\mathbf{u})]} \end{aligned}$$

Q What is the effect of the entropy term?

# Decompositions of the VLBO

## Likelihood/Prior Decomposition of the VLBO

By applying the chain rule to the preceding, and then reapplying the definition of KL divergence, we obtain another nice form

$$\begin{aligned}\text{VLBO}(q) &= \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{u} \mid \mathbf{c})] + \mathbb{H}[q(\mathbf{u})] \\ &= \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{u} \mid \mathbf{c})] - \mathbb{E}_q[\log q(\mathbf{u})] \\ &= \mathbb{E}_q[\log p(\mathbf{x} \mid \mathbf{u}, \mathbf{c})] + \mathbb{E}_q[\log p(\mathbf{u} \mid \mathbf{c})] - \mathbb{E}_q[\log q(\mathbf{u})] \\ &= \mathbb{E}_q[\log p(\mathbf{x} \mid \mathbf{u}, \mathbf{c})] - \text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \mathbf{c}))\end{aligned}$$

expected log likelihood                  divergence from prior

Note that the first term grows in magnitude as the number of samples increases; thus, the prior's influence diminishes asymptotically.

# Overview

---

The posterior perspective

# Maximizing the VLBO minimizes the KL divergence (to the posterior)

By definition, the KL divergence from the target posterior to the variational density is given by

$$\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})) = \mathbb{E}_q \left[ \log \frac{q(\mathbf{u})}{p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})} \right]$$

By the chain rule, we get

$$\begin{aligned} \text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})) &= \underbrace{\mathbb{E}_q[\log q(\mathbf{u})] - \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{u} \mid \mathbf{c})]}_{\text{energy/entropy decomposition}} + \log p(\mathbf{x} \mid \mathbf{c}) \\ &= -\text{VLBO}(q) + \text{constant} \end{aligned}$$

Discuss: What is the optimal variational density?

## Maximizing the VLBO minimizes the KL divergence (to the posterior)

By definition, the KL divergence from the target posterior to the variational density is given by

$$\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})) = \mathbb{E}_q \left[ \log \frac{q(\mathbf{u})}{p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})} \right]$$

By the chain rule, we get

$$\begin{aligned} \text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})) &= \underbrace{\mathbb{E}_q[\log q(\mathbf{u})] - \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{u} \mid \mathbf{c})]}_{\text{energy/entropy decomposition}} + \log p(\mathbf{x} \mid \mathbf{c}) \\ &= -\text{VLBO}(q) + \text{constant} \end{aligned}$$

### The optimal variational density

The optimal variational density,  $q^*(\mathbf{u})$  is the target posterior density  $p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})$  when the underlying variational family  $\mathcal{Q}$  is unrestricted

# **Overview**

---

## **Summary**

# Summary

- VI is a general tool. It is useful whenever you face intractable marginals.

Model	Inferential goal	Intractable marginal	Variational density	Posterior
General case	infer about $\theta$	$p(\mathbf{x}   \mathbf{c})$	$q(\mathbf{u})$	$p(\mathbf{u}   \mathbf{x}, \mathbf{c})$
Frequentist latent	$\text{argmax}_{\theta} p(\mathbf{x}   \theta)$	$p(\mathbf{x}   \theta) = \int p(\mathbf{x}, \mathbf{z}   \theta) d\mathbf{z}$	$q(\mathbf{z})$	$p(\mathbf{z}   \mathbf{x}, \theta)$
Bayesian (non-latent)	$p(\theta   \mathbf{x})$	$p(\mathbf{x}) = \int p(\theta, \mathbf{x}) d\theta$	$q(\theta)$	$p(\theta   \mathbf{x})$
Bayesian latent	$p(\mathbf{z}, \theta   \mathbf{x})$	$p(\mathbf{x}) = \int p(\theta, \mathbf{x}, \mathbf{z}) d\theta d\mathbf{z}$	$q(\mathbf{z}, \theta)$	$p(\mathbf{z}, \theta   \mathbf{x})$

# How does VI accommodate the goal of statistical inference?

Given selection of variational family  $\mathcal{Q}$ , the optimal variational density  $q^*$

...

## The marginal perspective

- *For frequentist models:* ... makes the VLBO best approximate the target marginal likelihood,  $p(\mathbf{x} | \boldsymbol{\theta})$ , which is what we wanted to maximize.
- *For Bayesian models:* ... raises the (approximate) evidence term  $p(\mathbf{x})$  (the term used for Bayesian model comparison) as high as possible.

## The posterior perspective

- *For Bayesian models:* ... is the family member which is closest to the target posterior  $p(\boldsymbol{u} | \mathbf{x})$ .
- *For frequentist models:* ... provides the best substitution  $q^*(\mathbf{z}) \approx p(\mathbf{z} | \mathbf{x}, \boldsymbol{\theta}^{\text{curr}})$  into the E-step of the EM algorithm<sup>1</sup>

<sup>1</sup>See next section for more information.

# **Overview**

---

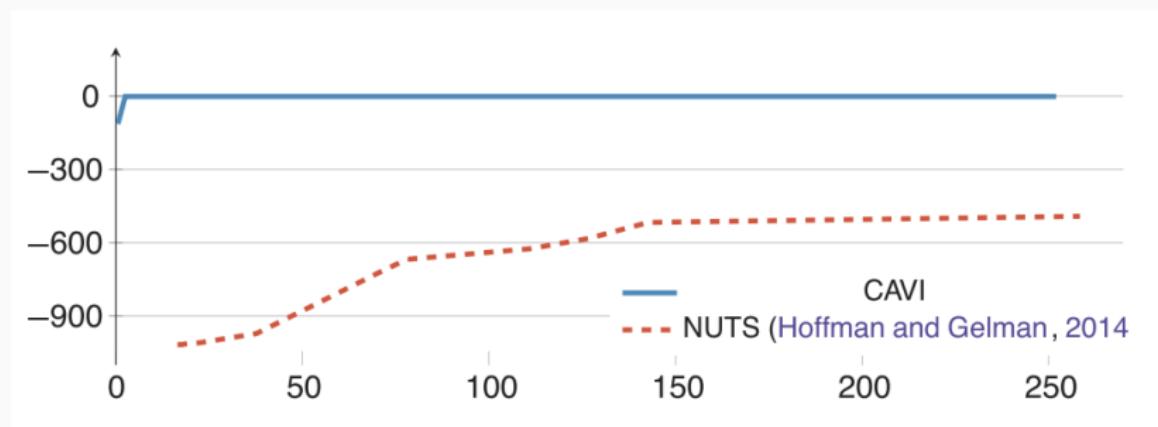
**Evaluation (in context)**

# Approximate Bayesian Inference

- The two most prominent strategies for approximating intractable posteriors are VI and Markov Chain Monte Carlo (MCMC).
- MCMC uses **sampling**. We construct a Markov chain over model parameters. The stationary distribution is the posterior. We approximate the posterior with samples.
- VI uses **approximation**. A tractable approximating family is chosen, and parameters are optimized to be close to the posterior.

# Variational Inference vs MCMC

Variational Inference scales better to large datasets.



**Figure 1:** Comparison of CAVI to a Hamiltonian Monte Carlo-based sampling technique. The plot shows log predictive test set accuracy by training time (minutes). CAVI fits a Gaussian mixture model to 10,000 images in less than a minute.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518), 859-877.

# Variational Inference vs. Expectation Propagation

Let us fix a distribution  $P$  and consider two optimization strategies

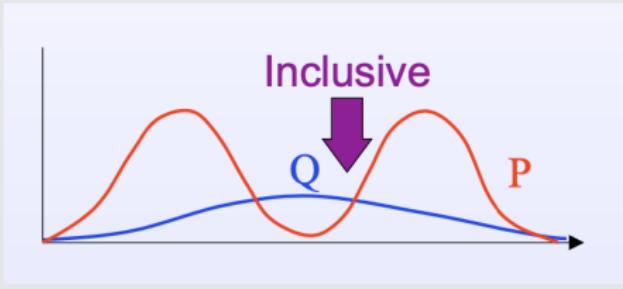
## Variational Inference

Minimizing  
 $\text{KL}(Q||P)$   
 $= \mathbb{E}_Q \left[ \log \frac{q(x)}{P(x)} \right]$



## Expectation Propagation

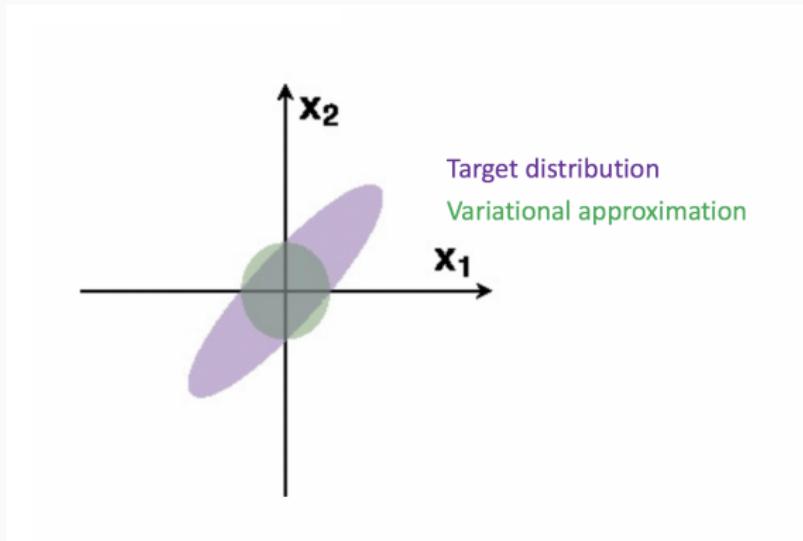
Minimizing  
 $\text{KL}(P||Q)$   
 $= \mathbb{E}_P \left[ \log \frac{P(x)}{q(x)} \right]$



Q: What does this say about VI? Which one would you prefer to use?

Image Credit: Tushar Tank

## Shortcoming: VI underestimates variance of the true posterior



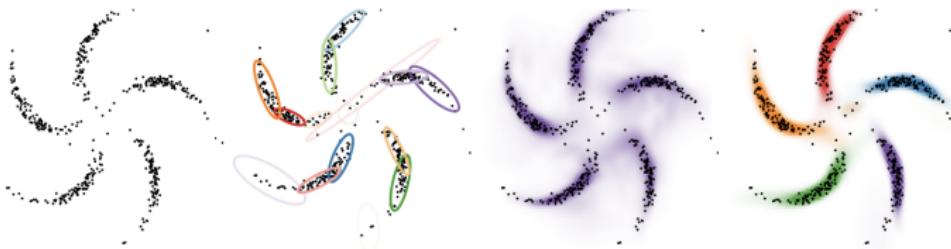
$$\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})) = \mathbb{E}_q \left[ \log \frac{q(\mathbf{u})}{p(\mathbf{u} \mid \mathbf{x}, \mathbf{c})} \right]$$

Intuition

- If  $q(\mathbf{u})$  is low, then we don't care (because of the expectation).
- If  $q(\mathbf{u})$  is high and  $p(\mathbf{x}, \mathbf{u} \mid \mathbf{c})$  is low, then we pay a price

## Modern application

We can compose probabilistic graphical models with neural networks to exploit their complementary strengths.



The resulting model is expressive, but also interpretable/decomposable.

# **Variational Inference and Expectation Maximization**

---

# Expectation Maximization (EM)

The EM algorithm refines an initial guess  $\theta^{(0)}$  via the recursion

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta} \mathbb{E}_{p(z | x, \theta^{(t)})} \left[ \ln p(x, z | \theta) \right]$$

until convergence to a local optimum.

## Example: Gaussian Hidden Markov Model

*E-step:* Compute  $p_i := p(z_i | x_i, \theta^{(t)})$  via the forward-backward algorithm.

*M-step:* Just a computation of weighted empirical means and variances:

$$\hat{\mu}_k^{(t)} = \frac{\sum_i (\textcolor{orange}{p_i = k}) x_i}{\sum_i (\textcolor{orange}{p_i = k})}, \quad \hat{\Sigma}_k^{(t)} = \frac{\sum_i (\textcolor{orange}{p_i = k}) (x_i - \hat{\mu}^{(t)}) (x_i - \hat{\mu}^{(t)})^T}{\sum_i (\textcolor{orange}{p_i = k})}$$

## EM from the perspective of VI

For a frequentist latent variable model, the VLBO is

$$\text{VLBO}(q_{\mathbf{z}}, \theta) = \mathbb{E}_q [\log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})] + \mathbb{H}[q(\mathbf{z})]$$

# EM from the perspective of VI

For a frequentist latent variable model, the VLBO is

$$\text{VLBO}(q_z, \theta) = \mathbb{E}_q [\log p(x, z | \theta)] + \mathbb{H}[q(z)]$$

Using coordinate ascent (in the sense of variational calculus), we get the following update equations:

$$\mathbf{q \; update :} \quad q_z^{(t+1)} = \operatorname{argmax}_{q_z} \text{VLBO}(q_z; \theta^{(t)}) \quad (2)$$

$$\mathbf{\theta \; update :} \quad \theta^{(t+1)} = \operatorname{argmax}_{\theta} \text{VLBO}(q_z^{(t+1)}; \theta) \quad (3)$$

# EM from the perspective of VI

For a frequentist latent variable model, the VLBO is

$$\text{VLBO}(q_z, \theta) = \mathbb{E}_q [\log p(x, z | \theta)] + \mathbb{H}[q(z)]$$

Using coordinate ascent (in the sense of variational calculus), we get the following update equations:

$$\mathbf{q \; update :} \quad q_z^{(t+1)} = \operatorname{argmax}_{q_z} \text{VLBO}(q_z; \theta^{(t)}) \quad (2)$$

$$\mathbf{\theta \; update :} \quad \theta^{(t+1)} = \operatorname{argmax}_{\theta} \text{VLBO}(q_z^{(t+1)}; \theta) \quad (3)$$

As argued earlier, we can solve the *q update* exactly by setting

$$q_z^{(t+1)} = p(z | x; \theta^{(t)})$$

in which case the  *$\theta$  update* becomes (what?)

# EM from the perspective of VI

For a frequentist latent variable model, the VLBO is

$$\text{VLBO}(q_z, \theta) = \mathbb{E}_q [\log p(x, z | \theta)] + \mathbb{H}[q(z)]$$

Using coordinate ascent (in the sense of variational calculus), we get the following update equations:

$$\mathbf{q \; update :} \quad q_z^{(t+1)} = \operatorname{argmax}_{q_z} \text{VLBO}(q_z; \theta^{(t)}) \quad (2)$$

$$\mathbf{\theta \; update :} \quad \theta^{(t+1)} = \operatorname{argmax}_{\theta} \text{VLBO}(q_z^{(t+1)}; \theta) \quad (3)$$

As argued earlier, we can solve the *q update* exactly by setting

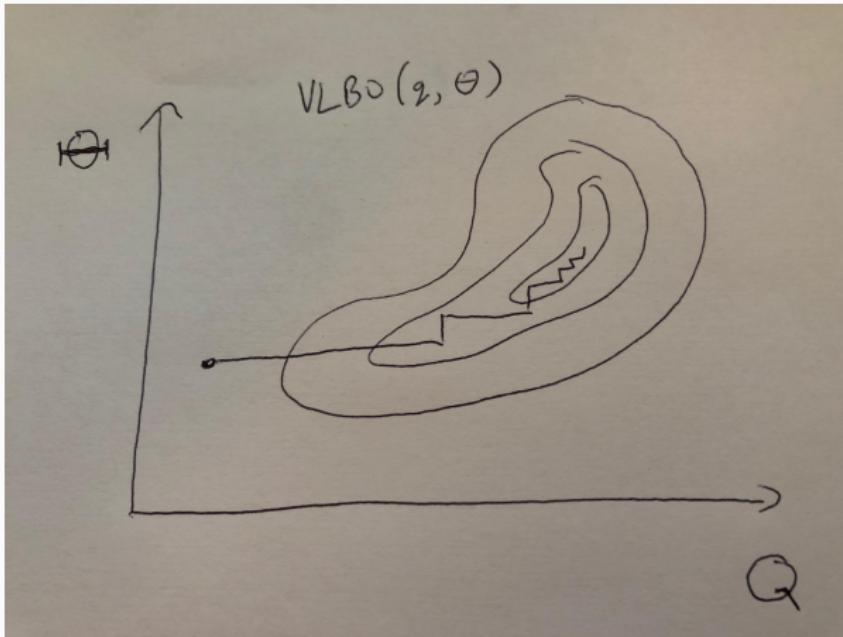
$$q_z^{(t+1)} = p(z | x; \theta^{(t)})$$

in which case the  *$\theta$  update* becomes (what?)

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta} \mathbb{E}_{p(z | x, \theta^{(t)})} \left[ \ln p(x, z | \theta) \right] \quad (4)$$

which is precisely the EM algorithm.

## EM as coordinate ascent on the VLBO



- If  $Q$  unrestricted, we have EM
- What if we restrict  $Q$  ?

# Variational Expectation Maximization (VEM)

Consider a **frequentist latent variable model**. Since we don't always have access to  $p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta})$ , we may restrict our variational family  $\mathcal{Q}$  to some convenient form. In this case, coordinate ascent on the VLBO is given by:

$$\begin{aligned} q_{\mathbf{z}}^{(t+1)} &= \operatorname{argmax}_{q_{\mathbf{z}} \in \mathcal{Q}} \text{VLBO}(q_{\mathbf{z}}; \boldsymbol{\theta}^{(t)}) \\ \boldsymbol{\theta}^{(t+1)} &= \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{q_{\mathbf{z}}^{(t+1)}} \left[ \ln p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta}) \right] \end{aligned}$$

which generalizes the EM algorithm.

# Variational Bayes Expectation Maximization (VBEM)

Consider a **Bayesian latent variable model**. So we need to swap  $p(x, z, \theta)$  for  $p(x, z | \theta)$  in the VLBO.

If we construct the variational density with the factorization

$$q(z, \theta) = q_z(z)q_\theta(\theta)$$

then the VLBO becomes

$$\text{VLBO}(q_z(z), q_\theta(\theta)) := \int \int q_z(z)q_\theta(\theta) \ln \left( \frac{p(z, \theta, x)}{q_z(z)q_\theta(\theta)} \right) d\theta \ dz \quad (5)$$

We can perform coordinate ascent on the VLBO with respect to the densities  $q_z$  and  $q_\theta$ :

$$\text{VB-E step : } q_z^{(t+1)} = \operatorname{argmax}_{q_z} \text{VLBO}(q_z; q_\theta^{(t)})$$

$$\text{VB-M step : } q_\theta^{(t+1)} = \operatorname{argmax}_{q_\theta} \text{VLBO}(q_z^{(t+1)}; q_\theta)$$

# VBEM: Derivation

See notes.

# Variational Bayes Expectation Maximization (VBEM)

The coordinate ascent equations have the form

$$\text{VB-E step : } q_z^{(t+1)} \propto \exp \left( \mathbb{E}_{q_\theta^{(t)}} [\ln p(x, z | \theta)] \right) \quad (6)$$

$$\text{VB-M step : } q_\theta^{(t+1)} \propto p(\theta) \exp \left( \mathbb{E}_{q_z^{(t)}} [\ln p(x, z | \theta)] \right) \quad (7)$$

## Prior-likelihood decomposition

Bayes' rule

$$p(\theta | x) \propto \underset{\text{posterior}}{p(\theta)} \underset{\text{prior}}{p(x | \theta)} \underset{\text{likelihood}}{}$$

VB-M update

$$\underset{\text{variational posterior}}{q_\theta^{(t+1)}} \propto \underset{\text{prior}}{p(\theta)} \underset{\text{expected likelihood under variational distribution}}{\exp \left( \mathbb{E}_{q_z^{(t)}} [\ln p(x, z | \theta)] \right)}$$

## VI and EM: Summary

Variational inference can be considered as a generalization of the expectation maximization algorithm (which is generally used by frequentists). It

- relaxes the need for tractable computation of the posterior distribution  $p(z | x, \theta)$ .
- relaxes the assumption that  $\theta$  is a deterministic variable; variational calculus lets us do coordinate ascent on the *distribution* governing  $\theta$ .

## **Coordinate Ascent Variational Inference (CAVI)**

---

# Coordinate Ascent Variational Inference (CAVI)

Coordinate ascent variational inference (CAVI) is a general approach to fitting models using VI.

This approach generalizes VBEM.

# Mean Field Coordinate Ascent Variational Inference (MF-CAVI)

## Mean field variational families

A variational family  $\mathcal{Q}$  is mean field if it factorizes

$$q(u_1, \dots, u_K) = \prod_{k=1}^K q_k(u_k) \quad (8)$$

*Mean field coordinate ascent variational inference (MF-CAVI)* is CAVI performed under the mean field assumption (8).

## Update equations for MF-CAVI

To perform coordinate ascent on the VLBO under the mean field assumption (8), we iteratively update our variational factors  $\{q_k\}_k$  via

$$q_k(u_k) \propto \exp \left\{ \mathbb{E}_{q_{-k}} \left[ \log p(u_k \mid \mathbf{u}_{-k}, \mathbf{x}, \mathbf{c}) \right] \right\} \quad (9)$$

The derivation uses variational calculus, and is nearly syntactically identical to the derivation of the VBEM updates.

## Update equations for MF-CAVI

To perform coordinate ascent on the VLBO under the mean field assumption (8), we iteratively update our variational factors  $\{q_k\}_k$  via

$$q_k(u_k) \propto \exp \left\{ \mathbb{E}_{q_{-k}} \left[ \log p(u_k \mid \mathbf{u}_{-k}, \mathbf{x}, \mathbf{c}) \right] \right\} \quad (9)$$

The derivation uses variational calculus, and is nearly syntactically identical to the derivation of the VBEM updates.

Rk: Note the connection to Gibbs sampling. In the MCMC literature, the distributions  $p(u_k \mid \text{rest})$  are known as *full conditionals* or *complete conditionals*. Gibbs sampling involves successive draws from the full conditionals. In mean-field variational inference, we take expectations of the same distributions, in order to update our posterior approximations.

## Example: Bayesian Mixture Model

---

## Example: Bayesian Mixture Model

---

Why variational Bayesian mixture models?

# Why variational Bayesian mixture models?

## Why Bayes?

All the usual reasons – exploit prior knowledge, protect against overfitting via Bayesian model averaging, can use the evidence (or ELBO) for model selection, etc.

## Why Variational Bayes?

Traditional MCMC becomes very burdensome for these types of models  
(mixture models, hidden mixture models, etc.) due to the multimodality in the posterior and the label switching.

See: A comparison of variational approximations for fast inference in mixed logit models

# VB Predictive vs. ML Solution

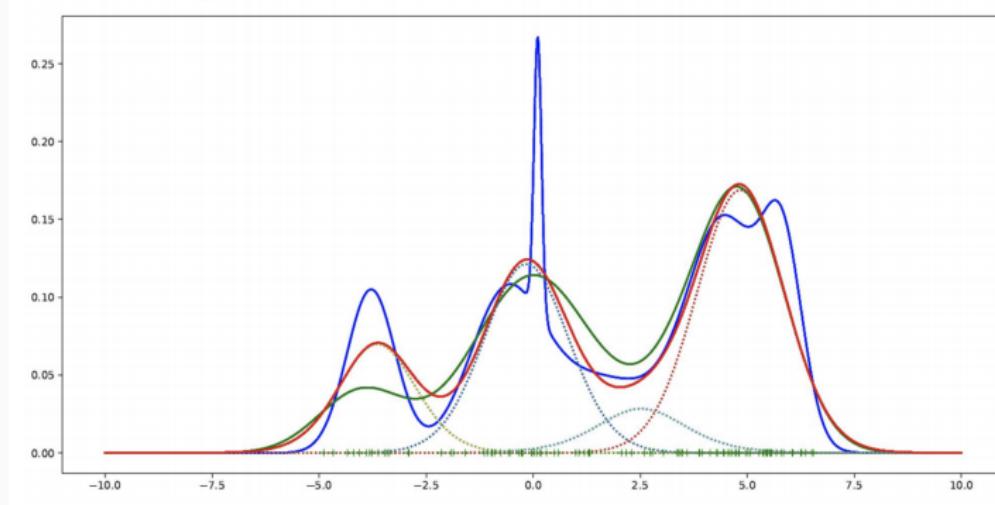


Image Credit: Lukas Burget

- VB was initialized from the ML solution
- VB recovers from ML overfitting and is closer to the true distribution for generating the training data

# Extensions of Bayesian Mixture Models

The Bayesian framework can be used to endow mixture models with many nice properties.

## Example: Dirichlet Process Mixture Models (DPMMs)

DPMM's have an unbounded number of mixture components.

The model automatically adapts its number of components.

- Click [here](#) for Demo 1.
- Click [here](#) for Demo 2.

# Example: Bayesian Mixture Model

---

Inference algorithm

## Example: Bayesian Gaussian Mixture Model

To see the mean field CAVI algorithm (9) in a concrete context, consider a version of the Bayesian Gaussian Mixture Model.

$$\mu_k \sim \text{Normal}(M_k = 0, V_k = \sigma^2) \quad k = 1, \dots, K$$

$$c_i \sim \text{Categorical}(\pi_1, \dots, \pi_K) \quad i = 1, \dots, n$$

$$x_i \mid c_i, \mu \sim \text{Normal}(\mu_{c_i}, 1) \quad i = 1, \dots, n$$

(The model is simple in that it assumes univariate observations and that each mixture component has unit variance.)

The joint density, by chain rule, is

$$p(x, c, \mu) = p(\mu) \prod_{i=1}^n p(c_i) p(x_i \mid c_i, \mu)$$

And a mean-field variational family is given by

$$q(c, \mu) = \prod_{k=1}^K q(\mu_k) \prod_{i=1}^n q(c_i)$$

## Example: Bayesian Gaussian Mixture Model

We apply (9) to determine the coordinate updates for  $q_{c_i}$ , the variational factors governing cluster assignments.

$$\begin{aligned} q(c_{ik}) &\propto \exp \left\{ \mathbb{E}_{q_{\mu_k}} \left[ \log p(c_i = k) + \log p(x_i | c_i = k, \mu) \right] \right\} \\ &\propto \exp \left\{ \mathbb{E}_{q_{\mu_k}} \left[ \log \pi_k + x_i \mu_k - \frac{1}{2} \mu_k^2 \right] \right\} \\ &\propto \pi_k \exp \left\{ x_i \mathbb{E}_{q_{\mu_k}} [\mu_k] - \frac{1}{2} \mathbb{E}_{q_{\mu_k}} [\mu_k^2] \right\} \end{aligned}$$

The next slide reveals that the  $q_{\mu_k}$  are Gaussian, and hence the above expectations are easy to compute.

**Note:** We abuse notation, and write  $q(c_{ik})$  as shorthand for  $q(c_i = k)$

## Bayesian Gaussian Mixture Model: Updates to mixture component means

Using the same strategy as when updating cluster assignments  $c_i$ , we obtain

$$\begin{aligned} q(\mu_k) &\propto \exp \left\{ \mathbb{E}_{q_{\mu_k}} \left[ \log p(\mu_k) + \sum_{i=1}^n \log p(x_i \mid c_i = k, \mu) \right] \right\} \\ &\propto \exp \left\{ -\frac{1}{2\sigma^2} \mu_k^2 + \sum_{i=1}^n \mathbb{E}_{q_c} \left[ 1_{c_i=k} \left( x_i \mu_k - \frac{1}{2} \mu_k^2 \right) \right] \right\} \\ &\propto \exp \left\{ \left( \sum_{i=1}^n q(c_{ik}) x_i \right) \mu_k + -\frac{1}{2} \left( \frac{1}{\sigma^2} + \sum_{i=1}^n q(c_{ik}) \right) \mu_k^2 \right\} \end{aligned}$$

which is an exponential family distribution with sufficient statistics  $(\mu_k, \mu_k^2)$  and base measure  $\propto 1$ ; hence it is Gaussian.

## Bayesian Gaussian Mixture Model: Updates to mixture component means

It is easy to show<sup>2</sup> that for a Gaussian with mean  $M$  and variance  $V$ , the natural parameters are given by

$$\eta_1 = \frac{M}{V}, \quad \eta_2 = -\frac{1}{2V}$$

From the last slide, the variational density  $q(\mu_k)$  has natural parameters

$$\eta_1 = \left( \sum_{i=1}^n q(c_{ik}) x_i \right), \quad \eta_2 = -\frac{1}{2} \left( \frac{1}{\sigma^2} + \sum_{i=1}^n q(c_{ik}) \right)$$

Using this, we can backsolve to determine the updates to the mean and variance of the Gaussian variational density governing the  $k$ th cluster mean:

$$M_k = \frac{\sum_{i=1}^n q(c_{ik}) x_i}{1/\sigma^2 + \sum_{i=1}^n q(c_{ik})}, \quad V_k = \frac{1}{1/\sigma^2 + \sum_{i=1}^n q(c_{ik})}$$

---

<sup>2</sup>Indeed, in this course we have seen

## Example: Latent Dirichlet Allocation (LDA)

---

## Acknowledgements

This section, especially the intro, borrows heavily from David Blei's 2012 ICML tutorial.

# Overview

LDA is a generative probabilistic model of a corpus of documents of text.

LDA assumes:

- There is a set of **topics** that describe the corpus
- Each document exhibits these topics to varying degrees.

# Overview

LDA is a generative probabilistic model of a corpus of documents of text.

LDA assumes:

- There is a set of **topics** that describe the corpus
- Each document exhibits these topics to varying degrees.

So:

- The topics and how they relate to the documents are hidden structure
- The main computational problem is to infer this hidden structure

## 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,<sup>10</sup> two genome researchers with radically different approaches presented complementary 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

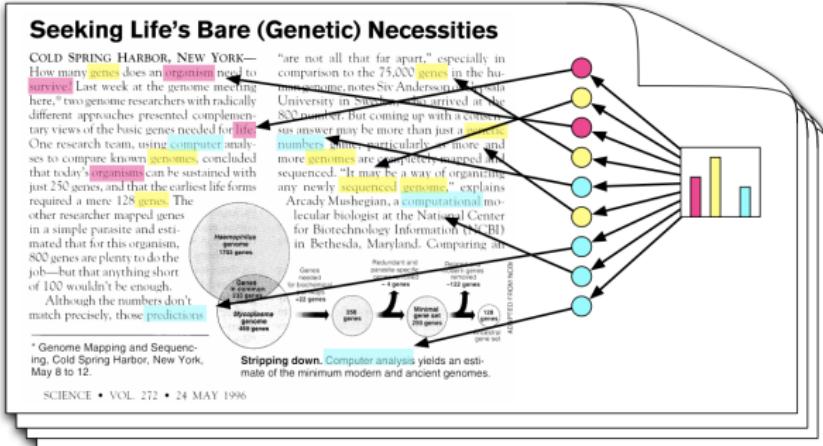
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Sv Anderson, a postdoctoral student in Svartman's lab who arrived at the 800 number. But coming up with a consensus answer may be more than just a numbers game, particularly if more and more genomes 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

<sup>10</sup> Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

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

## Topic proportions and assignments



- Each topic is a distribution over words.

## 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,<sup>10</sup> two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genetics, 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

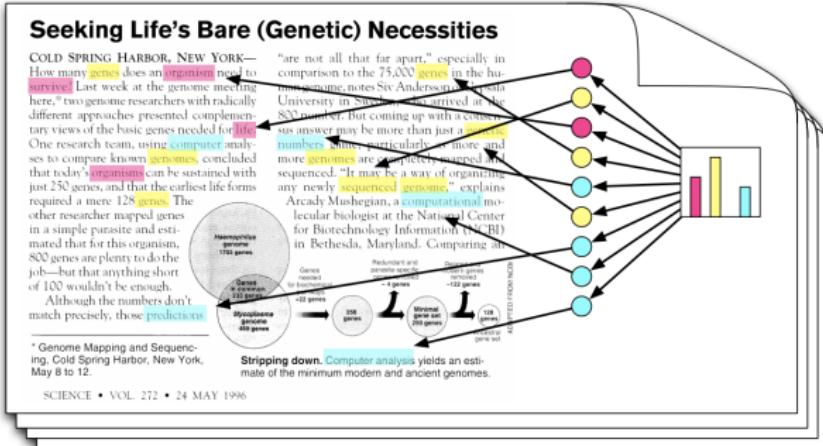
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Sv Anderson, a Russia University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a numbers game, particularly if more and more genomes are completely 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.

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

## Topic proportions and assignments



- Each **topic** is a distribution over words.
- Each **document** is a mixture of corpus-wide topics.

## 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

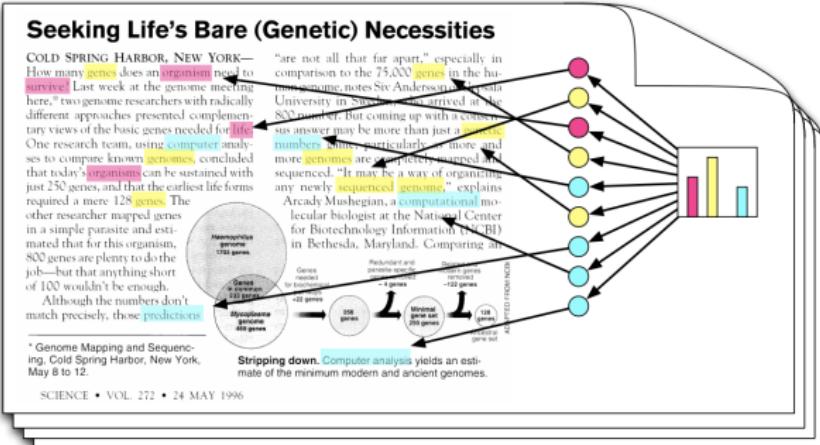
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,<sup>10</sup> two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genetics, 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 Sv Anderson, a Russia University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a numbers game, particularly if 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

## Topic proportions and assignments



- Each **topic** is a distribution over words.
- Each **document** is a mixture of corpus-wide topics.
- Each **word** is drawn from one of those topics.

## 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

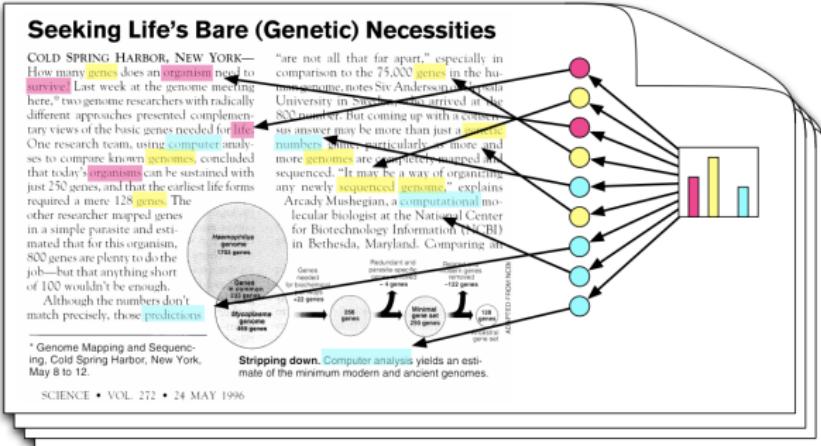
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,<sup>10</sup> two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genetics, 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 Sv Anderson, a Russia University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a numbers game, particularly if 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

## Topic proportions and assignments



- Each **topic** is a distribution over words.
- Each **document** is a mixture of corpus-wide topics.
- Each **word** is drawn from one of those topics.

## 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,<sup>10</sup> two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genetics, 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

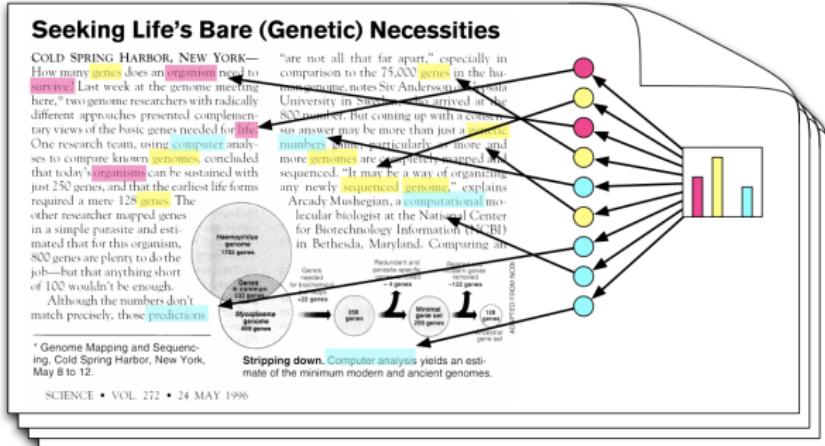
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Sv Anderson, a Russia University in St. Petersburg, who arrived at the 800 number. But coming up with a consensus answer may be more than just a numbers game, particularly if more and more genomes are 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.

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

## Topic proportions and assignments



- Each **topic** is a distribution over words.
- Each **document** is a mixture of corpus-wide topics.
- Each **word** is drawn from one of those topics.

Rk: This is a "bag of words" model.

## 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

## Topic proportions and assignments

### 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,<sup>10</sup> two genome researchers with radically different approaches presented complementary 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 Sv Anderson, a Russia University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a numbers game, particularly if 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.

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

SCIENCE • VOL. 272 • 24 MAY 1996

Figure 1: Topic modeling diagram showing topics as distributions over words and documents as mixtures of topics.

The figure illustrates the topic modeling process. At the top right, a bar chart represents a document as a mixture of topics. Below it, a legend shows the number of genes for each topic type: Minimum (250 genes), Minimal (250 genes), Intermediate (412 genes), Maximum (75,000 genes), and Human (75,000 genes). Arrows point from the document's topic proportions to the corresponding topics. Each topic is a circle containing a word and its probability. The Minimum topic contains 'genes' (0.04), 'DNA' (0.02), and 'genetic' (0.01). The Minimal topic contains 'life' (0.02), 'evolve' (0.01), and 'organism' (0.01). The Intermediate topic contains 'brain' (0.04), 'neuron' (0.02), and 'nerve' (0.01). The Maximum topic contains 'data' (0.02), 'number' (0.02), and 'computer' (0.01). The Human topic contains 'genes' (0.04), 'DNA' (0.02), and 'genetic' (0.01).

- Each **topic** is a distribution over words.
- Each **document** is a mixture of corpus-wide topics.
- Each **word** is drawn from one of those topics.

Rk: This is a “bag of words” model.

Source: David Blei, 2012 ICML Tutorial

## Topics



## Documents

## Topic proportions and assignments

### 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,<sup>6</sup> two genome researchers with radically different approaches presented complementary 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 230 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.

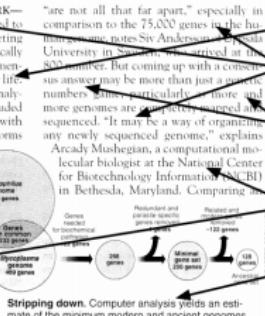
Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Steve Anderson of Cornell University in Ithaca, who arrived at this 800 number. But coming up with a consensus answer may be more than just a numbers game. As particularly as more and more genomes are fully or partially sequenced, "it may be a way of organizing any newly sequenced genome," explains

Arcadi 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.

- In reality, we only observe the documents.

## Topics



## Documents

## Topic proportions and assignments

### 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,<sup>6</sup> two genome researchers with radically different approaches presented complementary 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 230 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 Steve Anderson of Cornell University in Ithaca, who arrived at this 800 number. But coming up with a consensus answer may be more than just a numbers game. As particularly as more and more genomes are fully or partially sequenced, "it may be a way of organizing any newly sequenced genome," explains

Arcadi 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

Diagram illustrating the process of genome mapping and sequencing:

Original genome (1700 genes)

Genes needed for minimal life (128 genes)

Residual and redundant genes (1582 genes)

Minimal set of genes (230 genes)

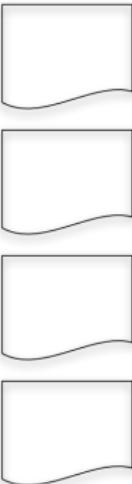
Maximal set of genes (800 genes)

Actual genome (223 genes)

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

- In reality, we only observe the documents.
- The model structure is **hidden**.

## Topics



## 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,<sup>6</sup> two genome researchers with radically different approaches presented complementary 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 350 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

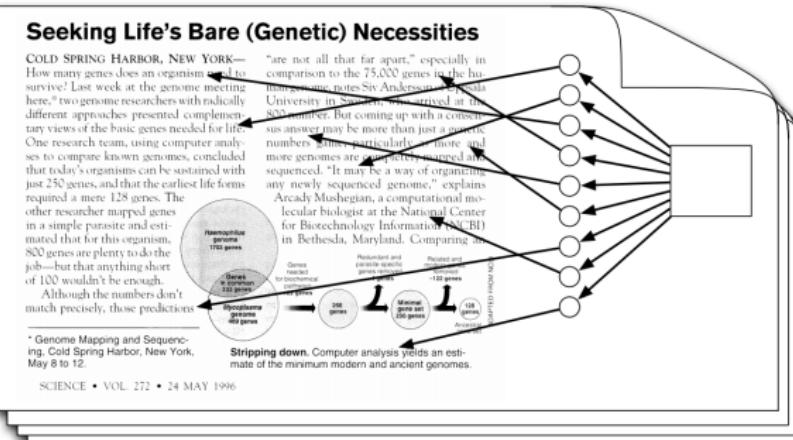
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at this 800 number. But coming up with a consensus answer may be more than just a numbers game. As more and more genomes are being fully sequenced,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

**Diagram:** A flowchart titled "Stripping down" illustrating the process of determining the minimum set of genes required for life. It starts with a large circle labeled "Homo sapiens 37,000 genes". An arrow points to a smaller circle labeled "Genes 2,221 genes". From this, another arrow points to a circle labeled "Genes needed for minimal life 128 genes". This is followed by a circle labeled "Residual and dispensable genes 350 genes". Finally, an arrow points to a circle labeled "Minimal set of genes required for life 128 genes". Arrows also point from the first three circles to a legend on the right: "Genes needed for minimal life" and "Residual and dispensable genes 350 genes".

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

SCIENCE • VOL. 272 • 24 MAY 1996

## Topic proportions and assignments



- In reality, we only observe the documents.
- The model structure is **hidden**.
- Our goal is to **infer** the hidden variables; i.e. compute

$$p(\text{topics, proportions, assignments} \mid \text{documents})$$

## Recall: Dirichlet Distribution

- Dirichlet distribution is *conjugate prior* of Categorical

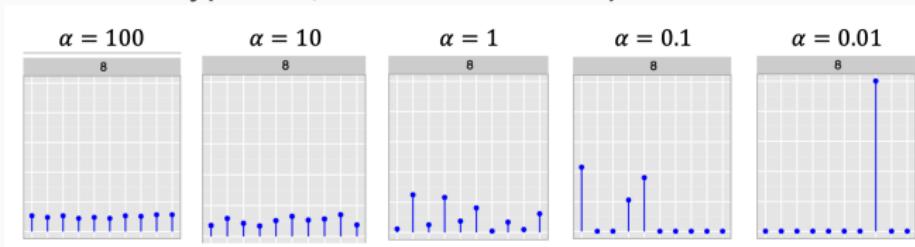
$$p(\theta \mid \alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \cdots \theta_k^{\alpha_k-1}$$

## Recall: Dirichlet Distribution

- Dirichlet distribution is *conjugate prior* of Categorical

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \cdots \theta_k^{\alpha_k-1}$$

- For symmetric Dirichlet distributions ( $\alpha_1 = \dots = \alpha_K$ ), a scalar hyperparameter  $\alpha = \sum_k \alpha_k$  controls the shape and sparsity of the  $\theta_d$ 's. (per-document topic proportions).
  - high  $\alpha$ : typical  $\theta_d$  (from the prior) will be uniform
  - small  $\alpha$ : a typical  $\theta_d$  (from the prior) will be sparse

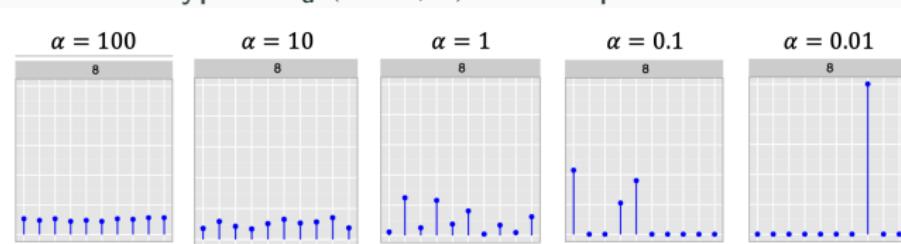


## Recall: Dirichlet Distribution

- Dirichlet distribution is *conjugate prior* of Categorical

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \cdots \theta_k^{\alpha_k-1}$$

- For symmetric Dirichlet distributions ( $\alpha_1 = \dots = \alpha_K$ ), a scalar hyperparameter  $\alpha = \sum_k \alpha_k$  controls the shape and sparsity of the  $\theta_d$ 's. (per-document topic proportions).
  - high  $\alpha$ : typical  $\theta_d$  (from the prior) will be uniform
  - small  $\alpha$ : a typical  $\theta_d$  (from the prior) will be sparse



- Likewise,  $\eta$  controls the shape and sparsity of the  $\beta_k$ 's (the topics – distribution over words)

# LDA: Generative Process

The model is described by the following **generative process**:

---

<sup>3</sup> Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup> Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

# LDA: Generative Process

The model is described by the following **generative process**:

- Fix a vocabulary of  $V$  words, and set the number of topics,  $K$ .

---

<sup>3</sup>Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup>Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

# LDA: Generative Process

The model is described by the following **generative process**:

- Fix a vocabulary of  $V$  words, and set the number of topics,  $K$ .
- Set hyperparameters  $\eta \in \mathbb{R}^V, \alpha \in \mathbb{R}^K$ .

---

<sup>3</sup>Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup>Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

# LDA: Generative Process

The model is described by the following **generative process**:

- Fix a vocabulary of  $V$  words, and set the number of topics,  $K$ .
- Set hyperparameters  $\eta \in \mathbb{R}^V, \alpha \in \mathbb{R}^K$ .
- For  $k$  in  $(1, K)$ :
  - Define the *topic* (a distribution over words),  $\beta_k \in \mathbb{R}^V \sim \text{Dir}(\eta)$ .<sup>3</sup>

---

<sup>3</sup>Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup>Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

# LDA: Generative Process

The model is described by the following **generative process**:

- Fix a vocabulary of  $V$  words, and set the number of topics,  $K$ .
- Set hyperparameters  $\eta \in \mathbb{R}^V, \alpha \in \mathbb{R}^K$ .
- For  $k$  in  $(1, K)$ :
  - Define the *topic* (a distribution over words),  $\beta_k \in \mathbb{R}^V \sim \text{Dir}(\eta)$ .<sup>3</sup>
- For  $d$  in  $(1, D)$  :
  - Choose (per-document) *topic proportions*  $\theta_d \in \mathbb{R}^K \sim \text{Dir}(\alpha)$ .<sup>4</sup>

---

<sup>3</sup>Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup>Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

# LDA: Generative Process

The model is described by the following **generative process**:

- Fix a vocabulary of  $V$  words, and set the number of topics,  $K$ .
- Set hyperparameters  $\eta \in \mathbb{R}^V, \alpha \in \mathbb{R}^K$ .
- For  $k$  in  $(1, K)$ :
  - Define the *topic* (a distribution over words),  $\beta_k \in \mathbb{R}^V \sim \text{Dir}(\eta)$ .<sup>3</sup>
- For  $d$  in  $(1, D)$  :
  - Choose (per-document) *topic proportions*  $\theta_d \in \mathbb{R}^K \sim \text{Dir}(\alpha)$ .<sup>4</sup>
  - For  $n$  in  $(1, N_d)$ :
    - Choose the *topic assignment*  $z_{d,n} \sim \text{Categorical}_K(\theta_d)$

---

<sup>3</sup>Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup>Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

# LDA: Generative Process

The model is described by the following **generative process**:

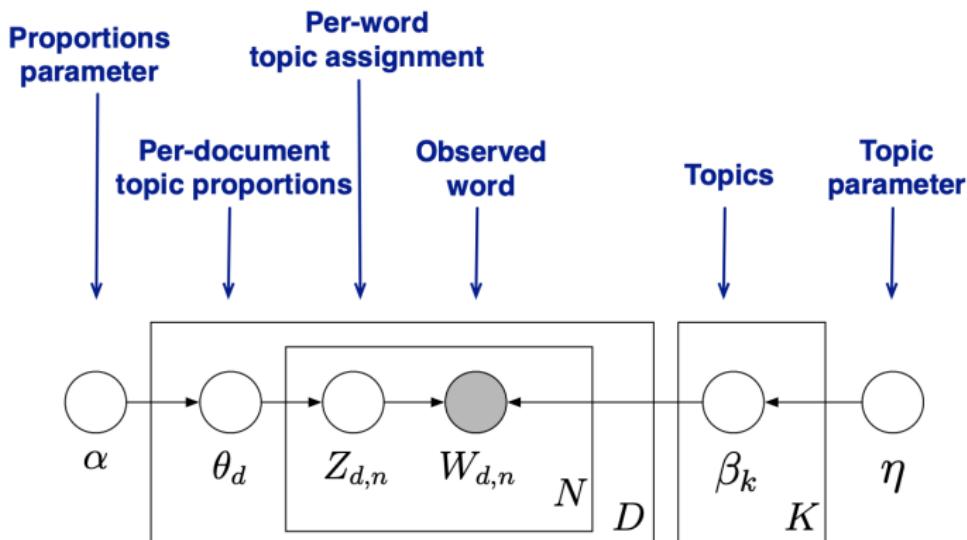
- Fix a vocabulary of  $V$  words, and set the number of topics,  $K$ .
- Set hyperparameters  $\eta \in \mathbb{R}^V, \alpha \in \mathbb{R}^K$ .
- For  $k$  in  $(1, K)$ :
  - Define the *topic* (a distribution over words),  $\beta_k \in \mathbb{R}^V \sim \text{Dir}(\eta)$ .<sup>3</sup>
- For  $d$  in  $(1, D)$  :
  - Choose (per-document) *topic proportions*  $\theta_d \in \mathbb{R}^K \sim \text{Dir}(\alpha)$ .<sup>4</sup>
  - For  $n$  in  $(1, N_d)$ :
    - Choose the *topic assignment*  $z_{d,n} \sim \text{Categorical}_K(\theta_d)$
    - Choose *word*  $w_{d,n} \sim \text{Categorical}_V(\beta_{z_{d,n}})$

---

<sup>3</sup>Interpretation of  $\eta$ : psuedocount of vocabulary words observed across prior topics.

<sup>4</sup>Interpretation of  $\alpha$ : psuedocount of topics observed across prior documents.

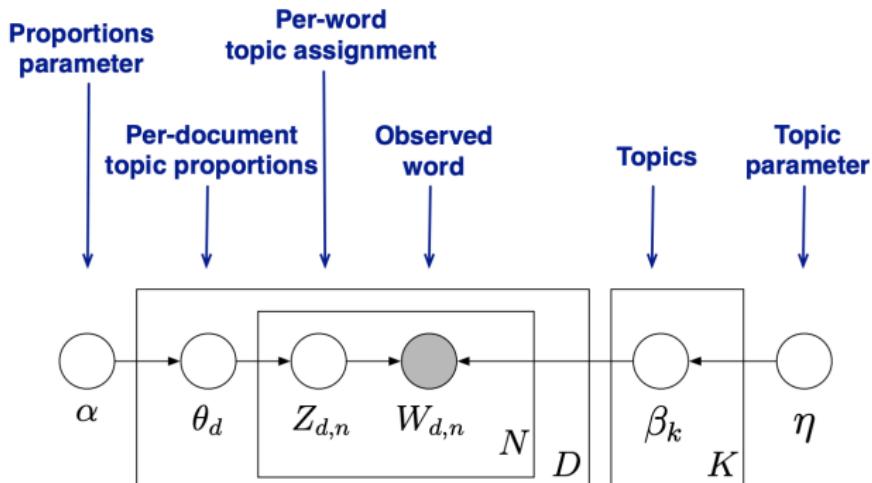
# LDA as a graphical model



Recall:

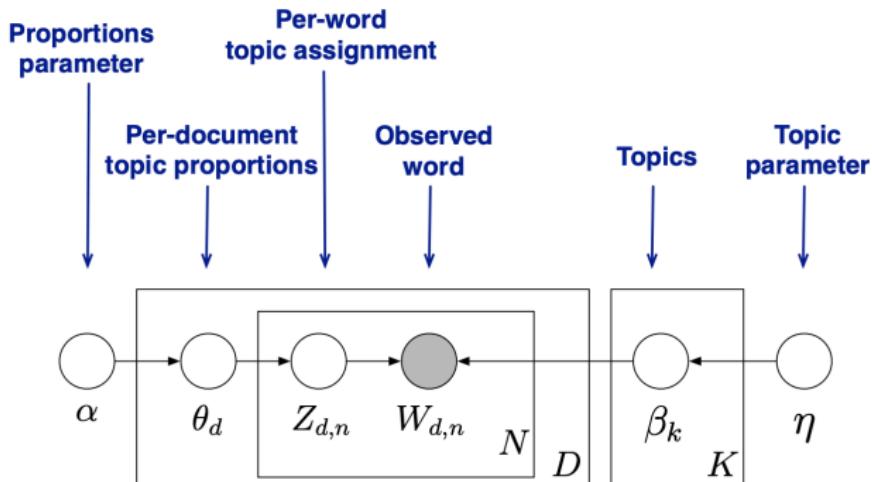
- Nodes are random variables.
- Shaded nodes are observed.
- Plates indicate replicated variables.
- Each node is conditionally independent from its non-descendants given its parents.

# Joint distribution



$$p(z, \theta, w, \beta | \alpha, \eta) = \prod_{k=1}^K p(\beta_k | \eta) \underset{Dirichlet}{\prod_{d=1}^D p(\theta_d | \alpha)} \underset{Categorical}{\prod_{n=1}^N p(z_{d,n} | \theta_d)} \underset{Categorical}{p(w_{d,n} | z_{d,n}, \beta_k)} \quad (10)$$

# Joint distribution



$$p(z, \theta, w, \beta | \alpha, \eta) = \prod_{k=1}^K p(\beta_k | \eta) \prod_{d=1}^D p(\theta_d | \alpha) \prod_{n=1}^N p(z_{d,n} | \theta_d) p(w_{d,n} | z_{d,n}, \beta_k) \quad (10)$$

**Remark.** Note that LDA is a "bag-of-words" model; i.e. the probability of a word (or document) is invariant to word order.

# Variational distribution

We approximate the posterior  $p(\theta | z, w)$  using mean field variational inference (8). In particular, we assume that the variational family  $\mathcal{Q}$  has a density which factorizes as

$$\begin{aligned} q &= q_\delta(\theta) q_\tau(z) \\ &= \prod_{d=1}^D \underbrace{q_\delta(\theta_d)}_{\text{Dirichlet}} \prod_{n=1}^{N_d} \underbrace{q_{\tau_n}(z_{d,n})}_{\text{Categorical}} \end{aligned} \tag{11}$$

**Note** We are treating the topics  $\beta_k$  as a constant for simplicity. For a fuller treatment, see Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

# Update equations

## LDA coordinate ascent update equations

$$\delta_{d,k} = \underset{\text{prior counts}}{\alpha_k} + \sum_{n=1}^{N_d} \underset{\text{var. expected assignment}}{\tau_{d,n,k}} \quad (12)$$

$$\begin{aligned} \tau_{d,n,k} &\propto \underset{\text{var. "prior" topic assignment}}{\exp \left\{ \mathbb{E}_{q_\delta(\theta)} \left[ \log \theta_{d,k} \right] \right\}} \underset{\text{likelihood}}{\beta_{k,[w_{d,n}]}} \\ &= \left( \Psi(\delta_k) - \Psi \left( \sum_j \delta_j \right) \right) \beta_{k,[w_{d,n}]} \end{aligned} \quad (13)$$

where  $\Psi(\cdot)$  is the first derivative of the log  $\Gamma$  function.

- Derivable via VBEM (see notes).
- Characteristic form: latent variable update depends on the data, global parameter update depends on the latent variable

**Note** We are treating  $\beta$  as a constant for brevity. We could fit also  $\beta$  to the data via VEM. (VEM does "empirical Bayes" for you.) More generally, we could model  $\beta$  as a random variable with VBEM. For a fuller treatment, see Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

## The role of analytical computations

The variational categorical update crucially hinges on facts about the **exponential family**.

# The role of analytical computations

The variational categorical update crucially hinges on facts about the **exponential family**.

In particular, the meat of the proof of the variational categorical update depends crucially on the fact that the Dirichlet of a single probability component is given by

$$\mathbb{E}_{q_\delta(\theta)} \left[ \log \theta_i \right] = \Psi(\delta_i) - \Psi\left(\sum_k \delta_k\right) \quad (14)$$

where  $\Psi(\cdot)$  is the first derivative of the  $\log \Gamma$  function. This fact is justified via facts about the exponential family (such as that the derivative of the log normalization factor with respect to the natural parameter is equal to the sufficient statistic).

# The role of analytical computations

The variational categorical update crucially hinges on facts about the **exponential family**.

In particular, the meat of the proof of the variational categorical update depends crucially on the fact that the Dirichlet of a single probability component is given by

$$\mathbb{E}_{q_\delta(\theta)} \left[ \log \theta_i \right] = \Psi(\delta_i) - \Psi\left(\sum_k \delta_k\right) \quad (14)$$

where  $\Psi(\cdot)$  is the first derivative of the  $\log \Gamma$  function. This fact is justified via facts about the exponential family (such as that the derivative of the log normalization factor with respect to the natural parameter is equal to the sufficient statistic).

For many more complicated models (e.g. VAE), such expectations (even the variational ones) are intractable, and so we won't be able to use CAVI.

# LDA Example Inference

- Data: 17K Science documents from 1990-2000 ( 11M words, 20K unique terms)
- Model: 100-topic LDA model, fit using variational inference

1 dna gene sequence genes sequences human genome genetic analysis two	2 protein cell cells proteins receptor fig binding activity activation kinase	3 water climate atmospheric temperature global surface ocean carbon atmosphere changes	4 says researchers new university just science like work first years	5 mantle high earth pressure seismic crust temperature earths lower earthquakes
6 end article start science readers service news card circle letters	7 time data two model to system number different with on	8 materials surface high structure temperature molecules chemical molecular to university	9 dna rna transcription protein site binding sequence proteins specific sequences	10 disease cancer patients human gene medical studies drug normal drugs
11 years million ago age university north early fig evidence record	12 species evolution population evolutionary university populations natural studies genetic today	13 protein structure proteins two amino binding acid residues molecular structural	14 cells cell virus hiv infection immune human antigen infected viral	15 space solar observations earth stars university mass sun astronomers telescope
16 fax manager science aaas advertising sales member recruitment associate washington	17 cells cell gene genes expression development mutant mice fig biology	18 energy electron state light quantum physics electrons high laser magnetic	19 research science national scientific scientists new states university united health	20 neurons brain cells activity fig channels university cortex neuronal visual

# Application: Anomaly Detection in Network Traffic Traces

LDA can be applied to *documents* that can be just about anything!

## IP Addresses

An **ip address**, like 72.194.113.177, is (roughly) an address assigned to each device connected to the Internet. My laptop has one, my iphone has one, every website (Google, Apple, etc.) has one, etc.

One application (Newton, B. D. (2012). Anomaly Detection in Network Traffic Traces Using Latent Dirichlet Allocation.)

- “Documents” = the session of a specific IP address
- “words” = the full external IP address and port number combinations.
- The “words” in each “document” are counted and then this data set is processed by LDA to yield a compact model of the data.

Q: Ok, but how to perform anomaly detection? Q: What assumptions are being made by LDA?

*I analyzed each of the anomalies detected in the last half hour of the trace [...]. The second anomaly with nearly 300 thousand messages exchanged with an SMTP server, is a bit more troubling. It is possible that this was actually a malicious client participating in a Mailbomb attack. According to the DARPA Intrusion Detection Attacks Database [8] a Mailbomb attack “is one in which the attacker sends many messages to a server, overflowing that server’s mail queue and possibly causing a system failure.”*

# **Automatic Differentiation Variational Inference (ADVI)**

---

# Coordinate ascent, and its discontents

## The ELBO (Evidence Lower Bound): Parametric View

$$L(\lambda) = \mathbb{E}_{q_\lambda(z)} \left[ \log p(x, z) - \log q(z) \right]$$

# Coordinate ascent, and its discontents

## The ELBO (Evidence Lower Bound): Parametric View

$$L(\lambda) = \mathbb{E}_{q_\lambda(z)} \left[ \log p(x, z) - \log q(z) \right]$$

Traditionally, we optimize via coordinate-ascent (CAVI).

$$\hat{\lambda}_{t+1} = \hat{\lambda}_t + \rho_t \nabla L(\hat{\lambda}_t)$$

# Coordinate ascent, and its discontents

## The ELBO (Evidence Lower Bound): Parametric View

$$L(\lambda) = \mathbb{E}_{q_\lambda(z)} \left[ \log p(x, z) - \log q(z) \right]$$

Traditionally, we optimize via coordinate-ascent (CAVI).

$$\hat{\lambda}_{t+1} = \hat{\lambda}_t + \rho_t \nabla L(\hat{\lambda}_t)$$

**Problem:** Each model's VI algorithm is<sup>5</sup> a snowflake.

- Requires *model-specific* computations
- Must pick a good  $q$ .

---

<sup>5</sup>if it exists

# Differentiable Probability Models

GOAL: Approximate the posterior  $p(\theta | x) \propto p(x | \theta)p(\theta)$

## The class of probability models that ADVI supports

- Dataset  $\mathbf{x} = x_{1:N}$  where each  $x_n$  is a realization of a discrete or continuous random variable
- Latent variables  $\theta$  are continuous
- $\nabla_\theta \log p(\mathbf{x}, \theta)$  exists within the support of the prior

$$\Theta := \text{supp}(p(\theta)) = \{\theta \mid \theta \in \mathbb{R}^K \text{ and } p(\theta) > 0\} \subseteq \mathbb{R}^K$$

### INCLUDES

*Generalized linear models*

*Mixture models*

*Topic models*

*Linear dynamic systems*

*Gaussian process models*

*Deep exponential families*

### EXCLUDES

*Ising model*

*Sigmoid belief networks*

*Bayesian nonparametric models*

## Step 1: Transforming to unbounded support

We define a differentiable bijection to give the parameters unbounded support.

$$T : \Theta \rightarrow \mathbb{R}^K$$

$$\theta \mapsto \zeta$$

We use *change of variables* to express the joint density in the new space.

$$p(\mathbf{x}, \zeta)_{\text{new space}} = p(\mathbf{x}, T^{-1}(\zeta))_{\text{transformed original space}} \left| \det J_{T^{-1}}(\zeta) \right|$$

So in the new space, the ELBO becomes

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\zeta; \lambda)} \left[ \log p(\mathbf{x}, T^{-1}(\zeta)) + \log \left| \det J_{T^{-1}}(\zeta) \right| \right] + \mathbb{H}[q(\zeta; \lambda)]$$

## Step 1: Transforming to unbounded support

We define a differentiable bijection to give the parameters unbounded support.

$$T : \Theta \rightarrow \mathbb{R}^K$$

$$\theta \mapsto \zeta$$

We use *change of variables* to express the joint density in the new space.

$$p(\mathbf{x}, \zeta)_{\text{new space}} = p(\mathbf{x}, T^{-1}(\zeta)) \left| \det J_{T^{-1}}(\zeta) \right|_{\text{transformed original space}}$$

So in the new space, the ELBO becomes

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\zeta; \lambda)} \left[ \log p(\mathbf{x}, T^{-1}(\zeta)) + \log \left| \det J_{T^{-1}}(\zeta) \right| \right] + \mathbb{H}[q(\zeta; \lambda)]$$

✓ Model-independent variational factors.

✗ Cannot compute the gradient of the cost function

## Step 2: The reparameterization trick

We *re-parameterize* the Gaussian  $\zeta \sim \mathcal{N}(\mu, \Sigma)$ , such that its dependence on the original parameter  $\lambda = (\mu, \Sigma)$  is transferred to a (deterministic) standardization function,  $S_\lambda$

- Factorize  $\Sigma = \mathbf{L}\mathbf{L}^T$ .
- The standardized random variable is

$$\epsilon := S_\lambda(\zeta) = \mathbf{L}^{-1}(\zeta - \mu), \quad \text{where } \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

- The unstandardized random variable can be recovered via

$$\zeta = S_\lambda^{-1}(\epsilon)$$

### ADVI ELBO

$$\mathcal{L}(\lambda) = \mathbb{E}_{\mathcal{N}(\epsilon; 0, \mathbf{I})} \left[ \log p(x, T^{-1}(S_\lambda^{-1}(\epsilon))) + \log \left| \det J_{T^{-1}}(S_\lambda^{-1}(\epsilon)) \right| \right] + \mathbb{H}[q(\zeta; \lambda)]$$

## Step 2: The reparameterization trick

We *re-parameterize* the Gaussian  $\zeta \sim \mathcal{N}(\mu, \Sigma)$ , such that its dependence on the original parameter  $\lambda = (\mu, \Sigma)$  is transferred to a (deterministic) standardization function,  $S_\lambda$

- Factorize  $\Sigma = \mathbf{L}\mathbf{L}^T$ .
- The standardized random variable is

$$\epsilon := S_\lambda(\zeta) = \mathbf{L}^{-1}(\zeta - \mu), \quad \text{where } \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

- The unstandardized random variable can be recovered via

$$\zeta = S_\lambda^{-1}(\epsilon)$$

### ADVI ELBO

$$\mathcal{L}(\lambda) = \mathbb{E}_{\mathcal{N}(\epsilon; 0, \mathbf{I})} \left[ \log p(x, T^{-1}(S_\lambda^{-1}(\epsilon))) + \log \left| \det J_{T^{-1}}(S_\lambda^{-1}(\epsilon)) \right| \right] + \mathbb{H}[q(\zeta; \lambda)]$$

✓ Model-independent variational factors.

✓ Can estimate the gradient of the expectation.

**Questions?**