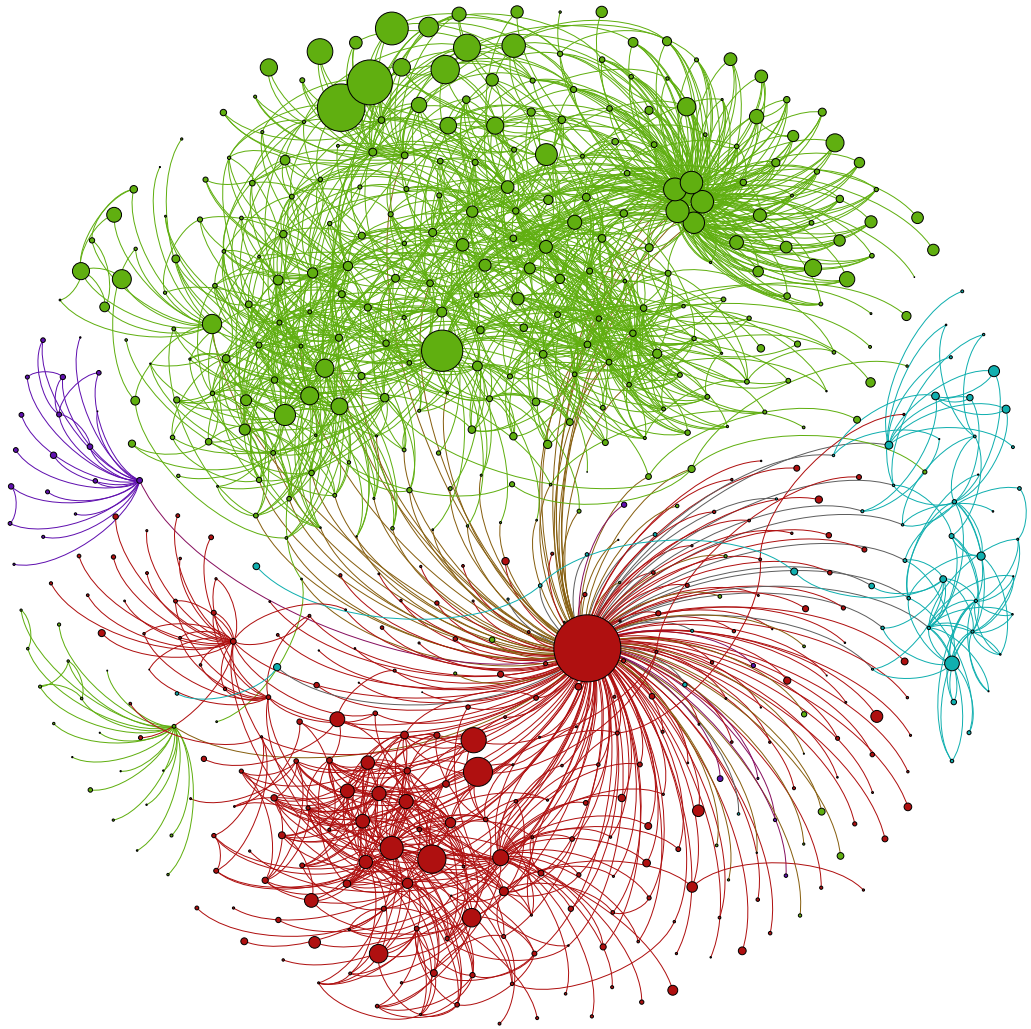
VARIATIONAL INFERENCE: FOUNDATIONS AND INNOVATIONS

David M. Blei
Departments of Computer Science and Statistics
Columbia University

with Rajesh Ranganath (Princeton) and Shakir Mohamed (DeepMind)

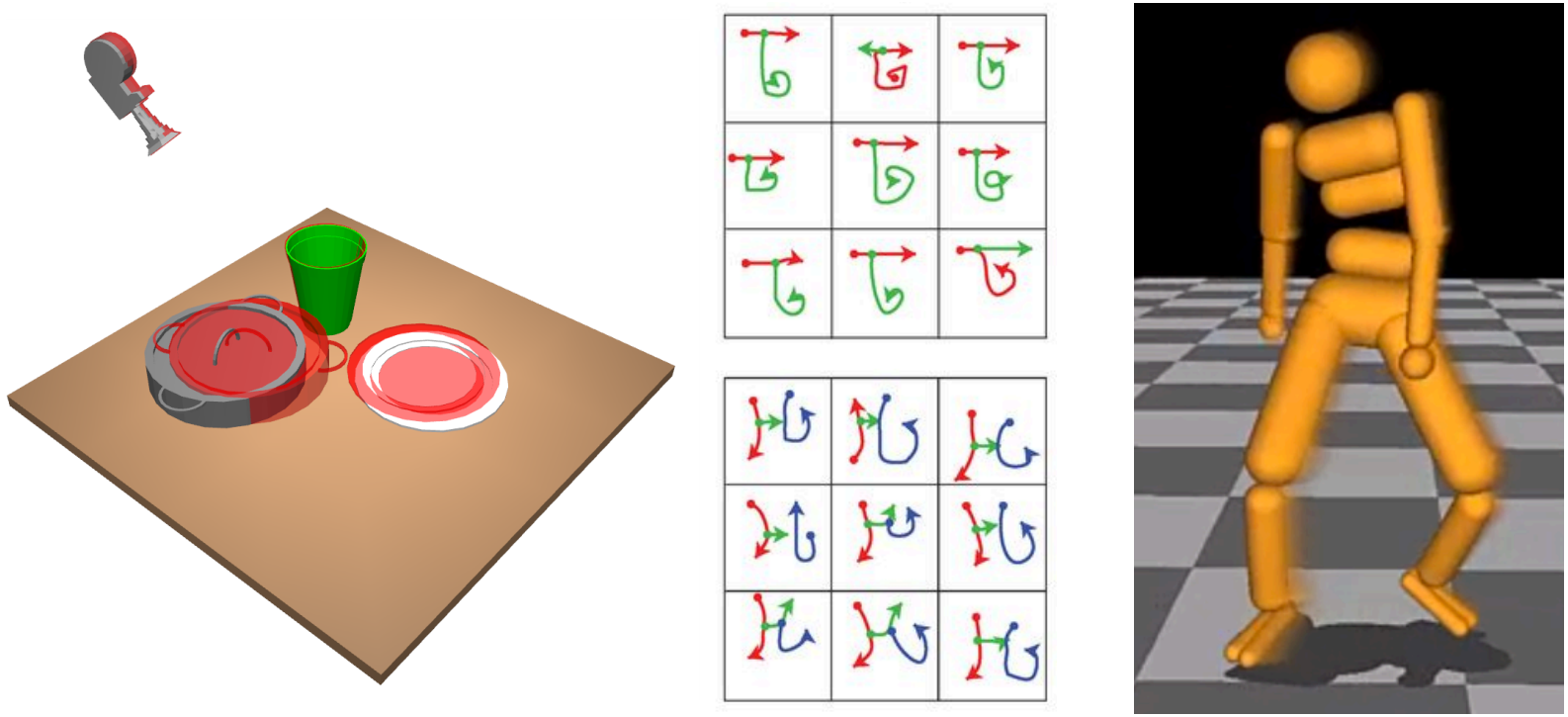


Communities discovered in a 3.7M node network of U.S. Patents

[Gopalan and Blei, PNAS 2013]

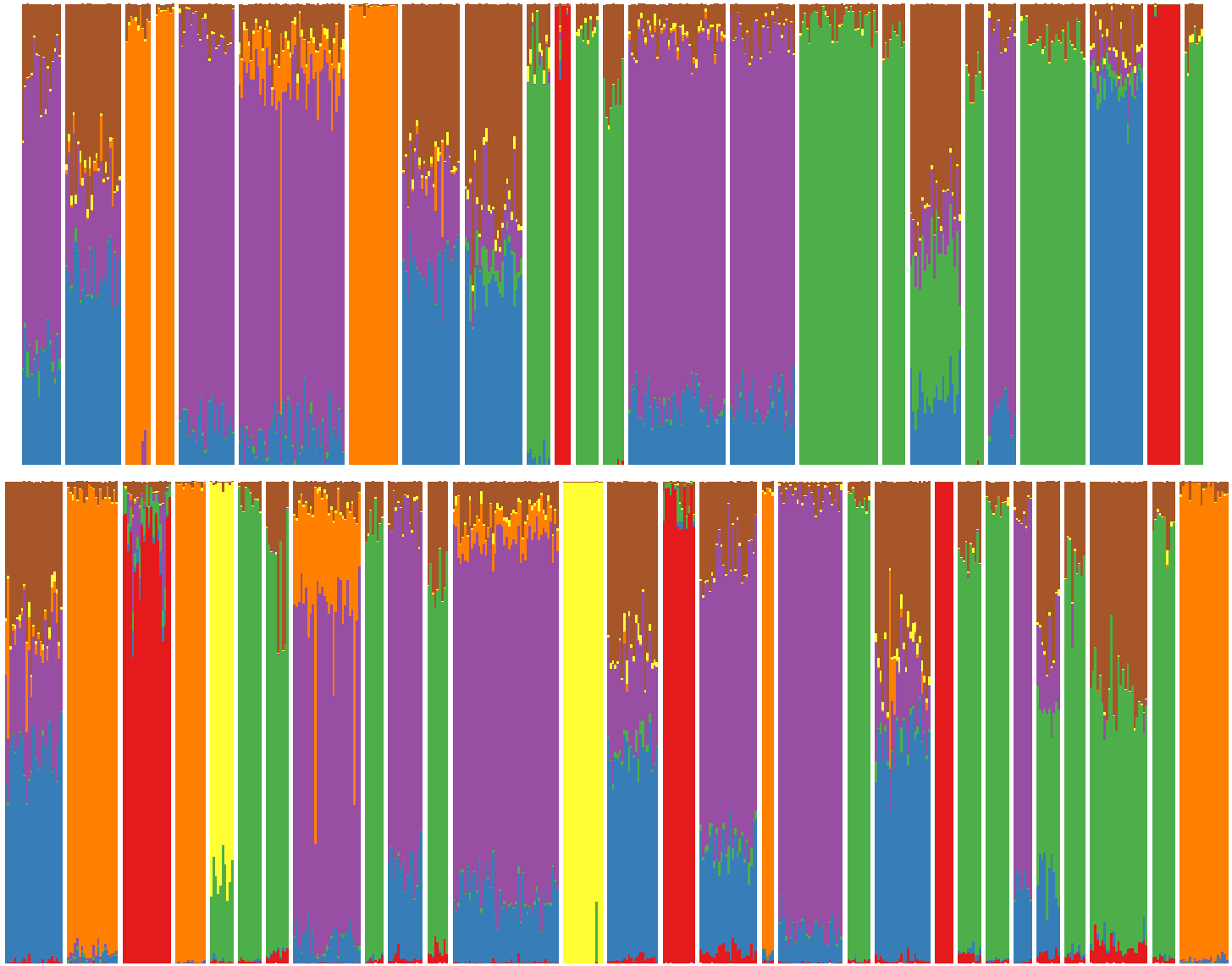
Topics found in 1.8M articles from the New York Times

[Hoffman, Blei, Wang, Paisley, JMLR 2013]



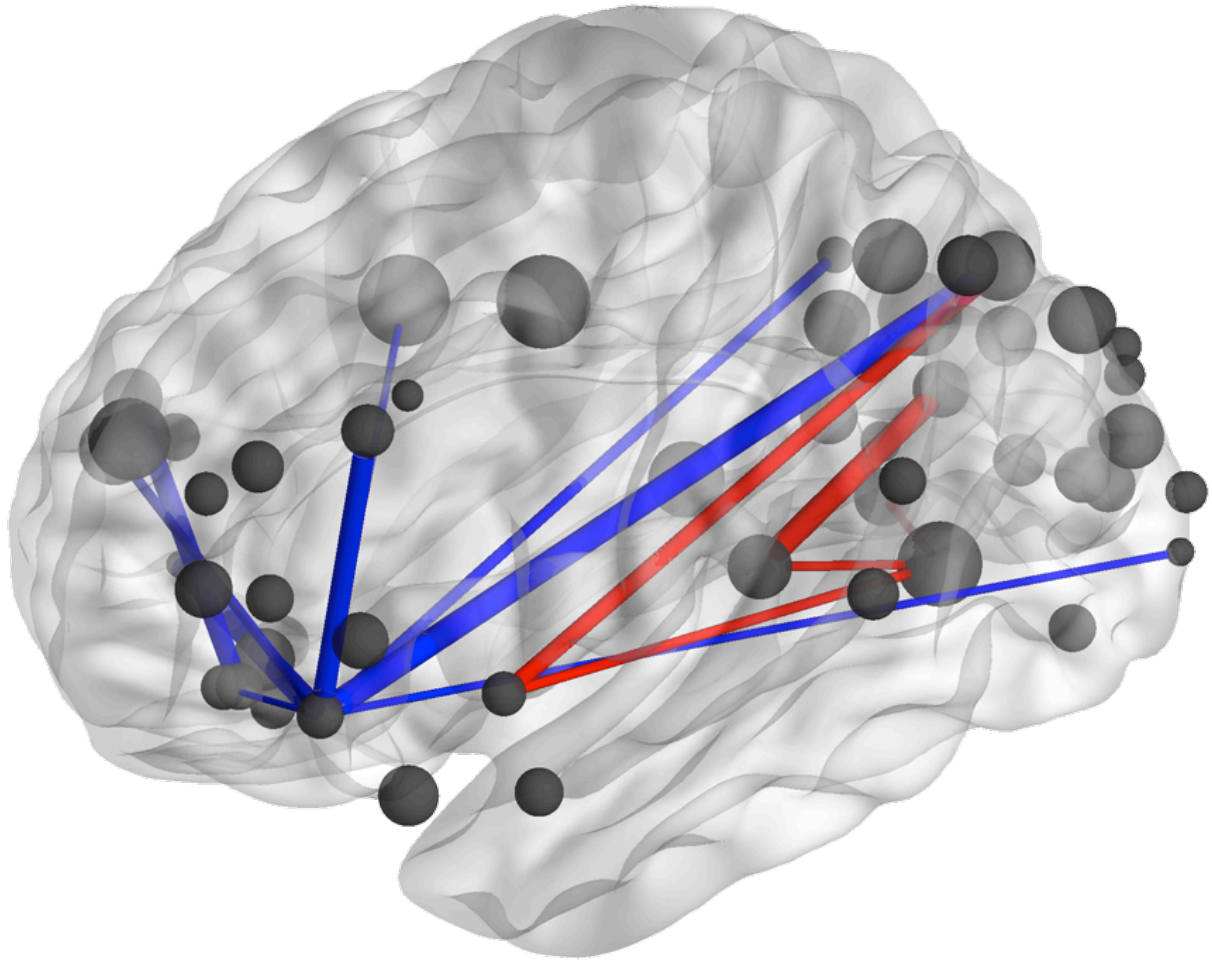
Scenes, concepts and control.

[Eslami+ 2016, Lake+ 2015]



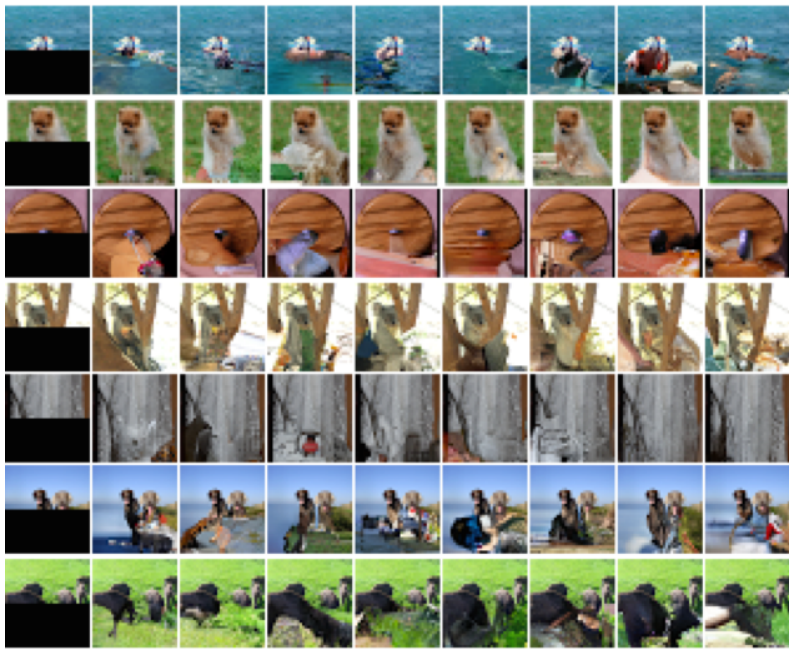
Population analysis of 2 billion genetic measurements

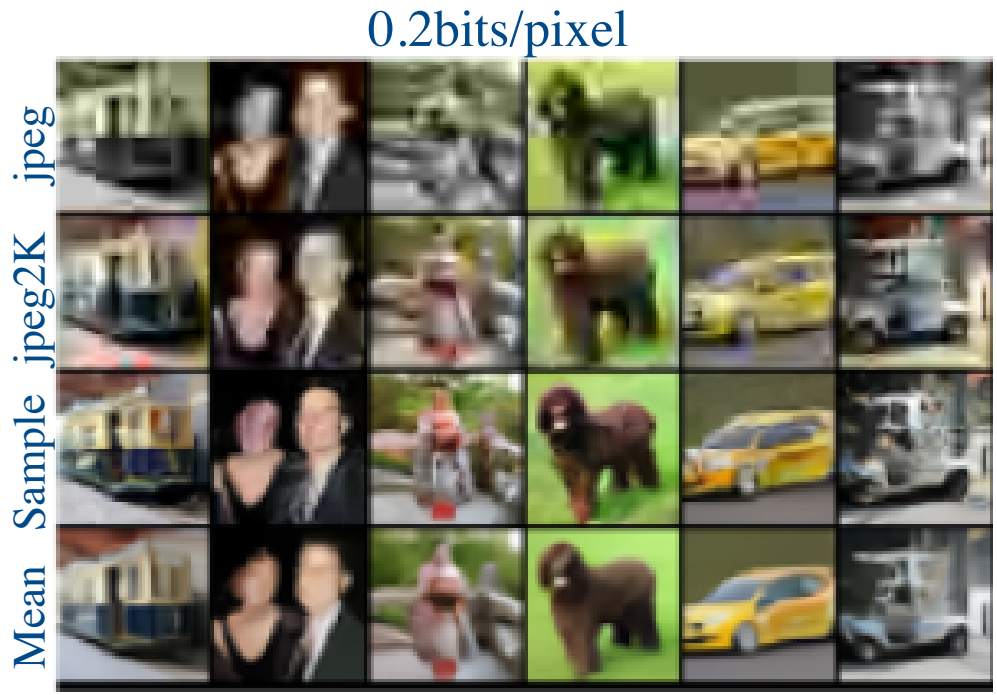
[Gopalan, Hao, Blei, Storey, Nature Genetics (in press)]



Neuroscience analysis of 220 million fMRI measurements

[Manning+ PLOS ONE 2014]





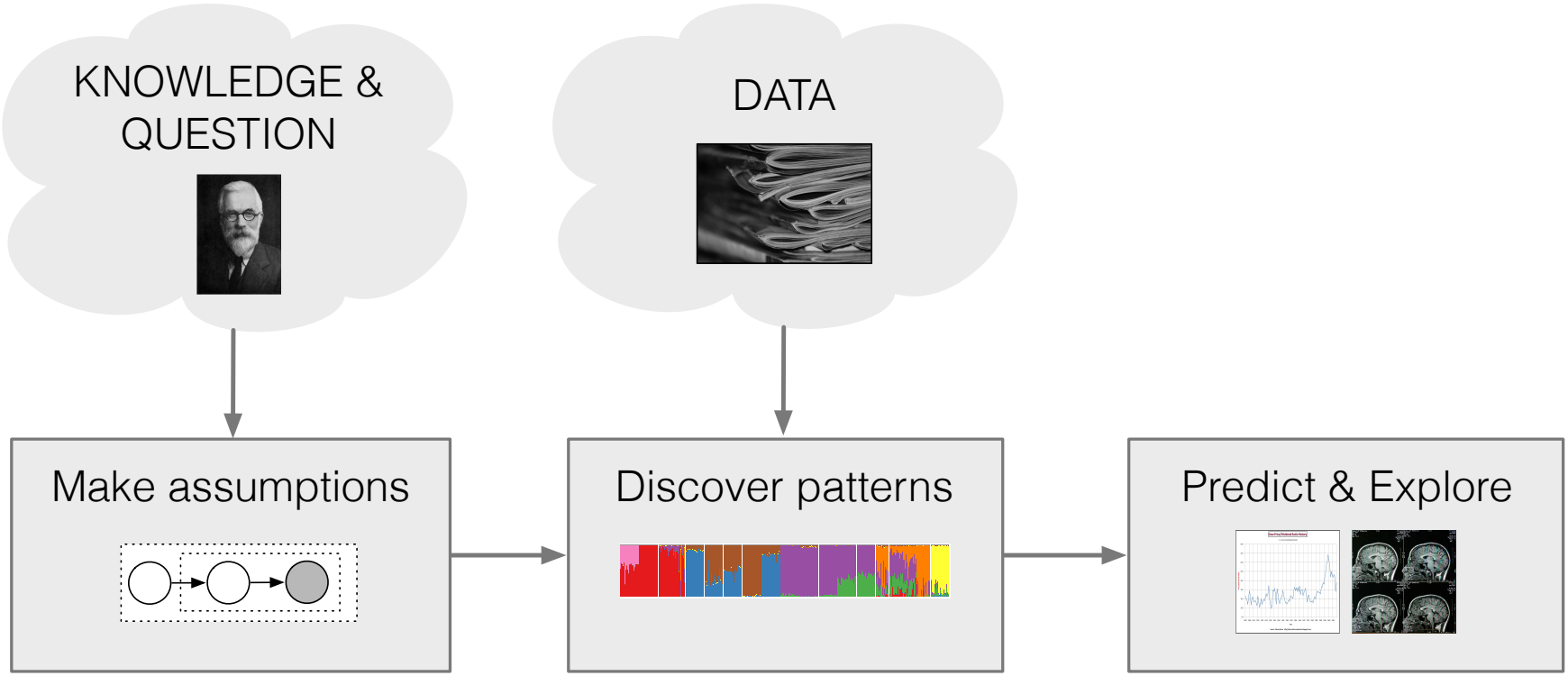
Compression and content generation.

[Van den Oord+ 2016, Gregor+ 2016]

Analysis of 1.7M taxi trajectories, in Stan

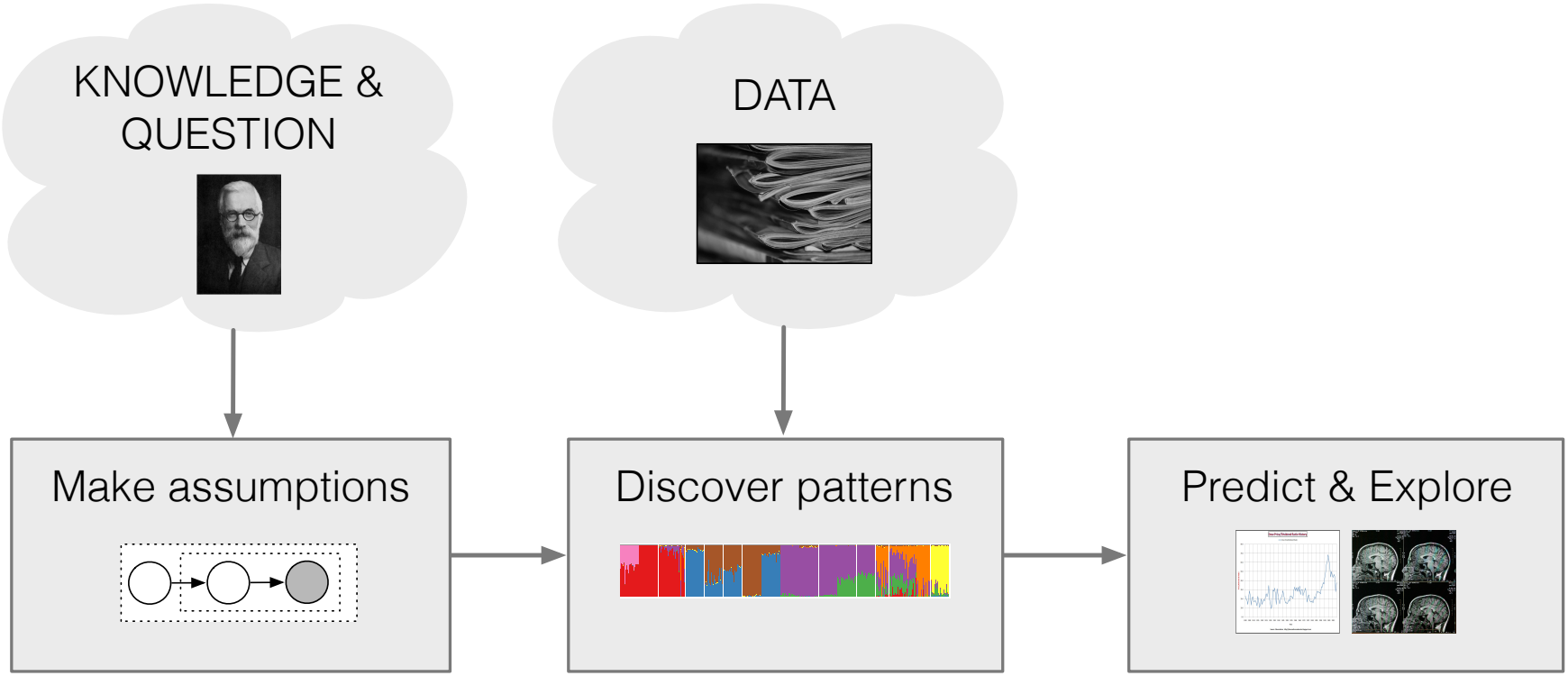
[Kucukelbir+ 2016]

The probabilistic pipeline

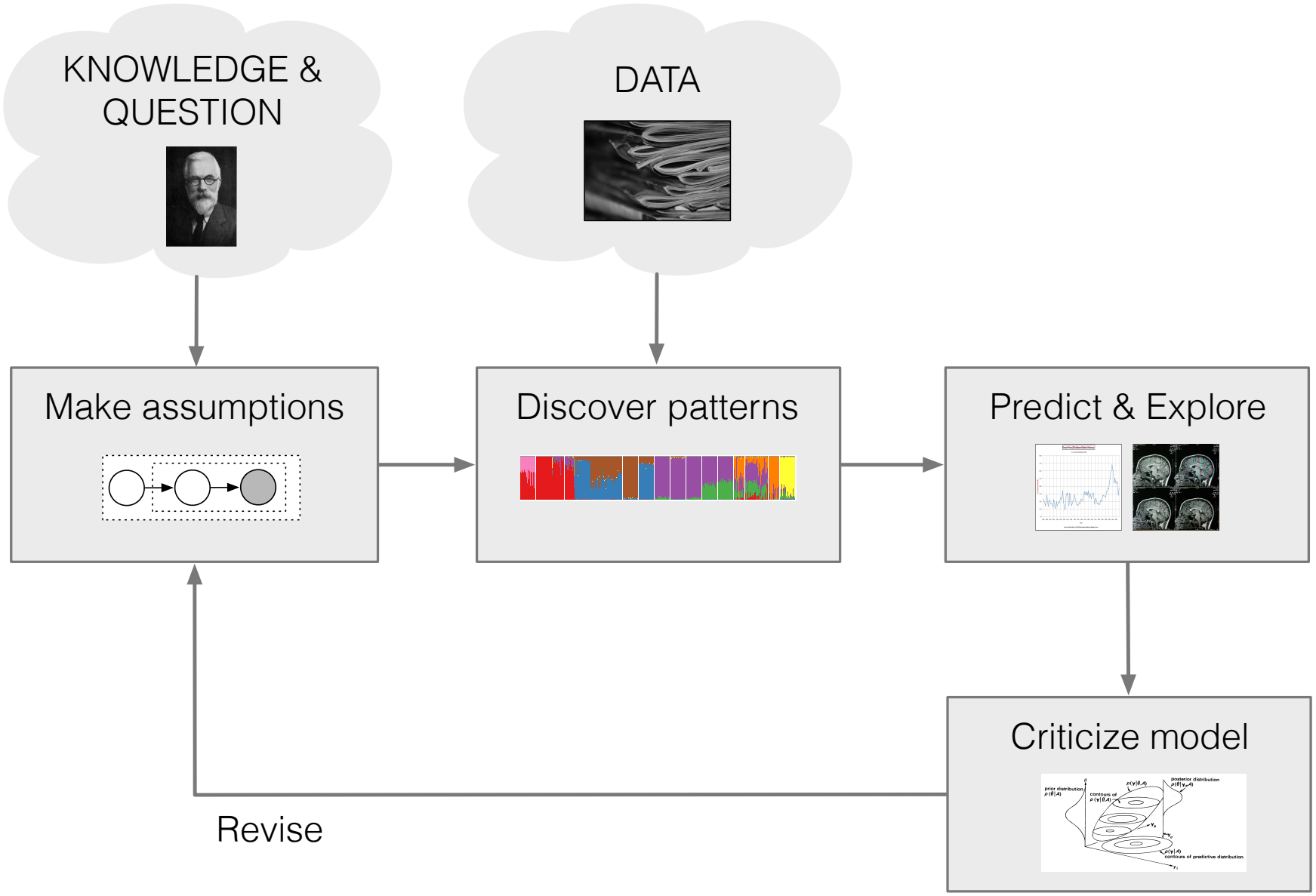


- Customized data analysis is important to many fields.
- Pipeline separates **assumptions**, **computation**, **application**
- Eases collaborative solutions to statistics problems

The probabilistic pipeline



- **Posterior inference** is the key algorithmic problem.
- Answers the question: What does this model say about this data?
- Our goal: **General** and **scalable** approaches to posterior inference



[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

PART I

Main ideas and historical context

Probabilistic machine learning

- A probabilistic model is a joint distribution of hidden variables $\mathbf{z}$ and observed variables $\mathbf{x}$,

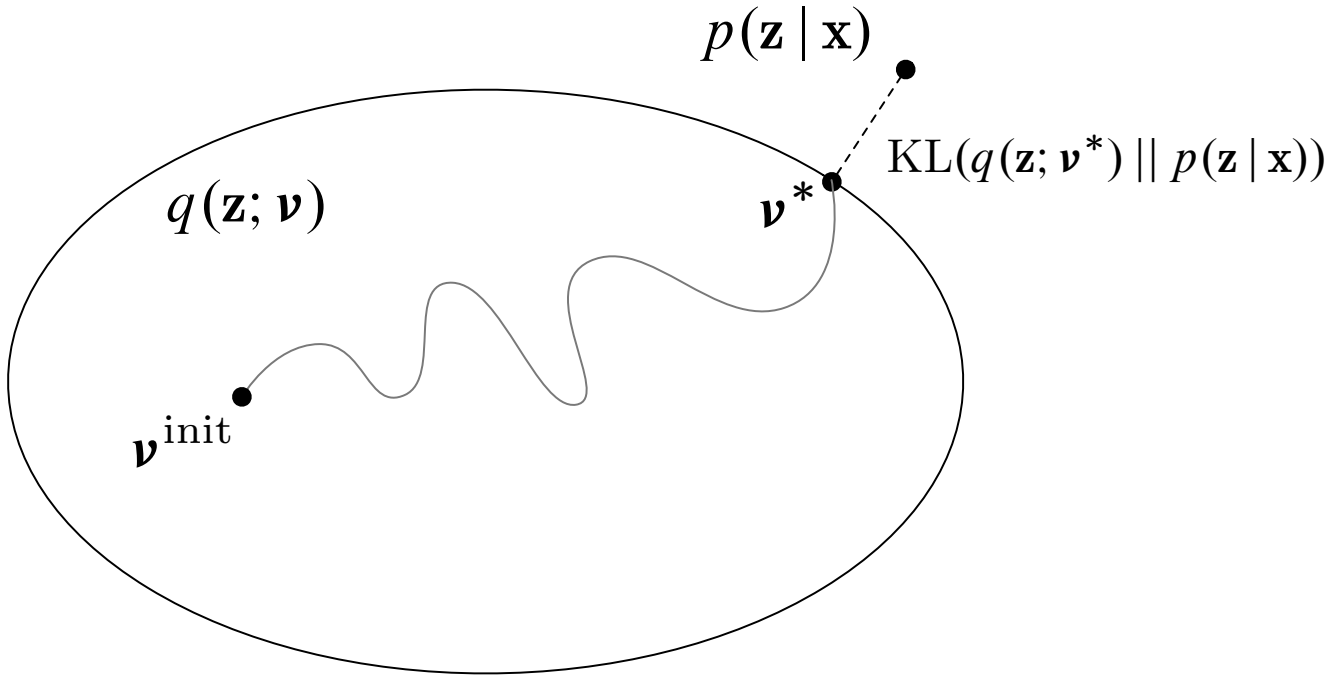
$$p(\mathbf{z}, \mathbf{x}).$$

- Inference about the unknowns is through the **posterior**, the conditional distribution of the hidden variables given the observations

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})}.$$

- For most interesting models, the denominator is not tractable. We appeal to **approximate posterior inference**.

Variational inference

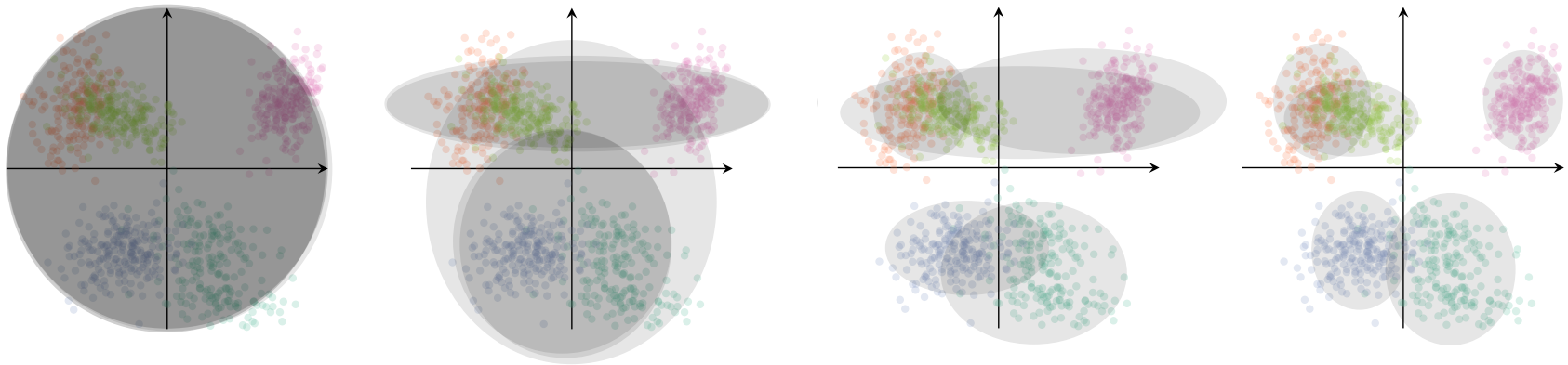


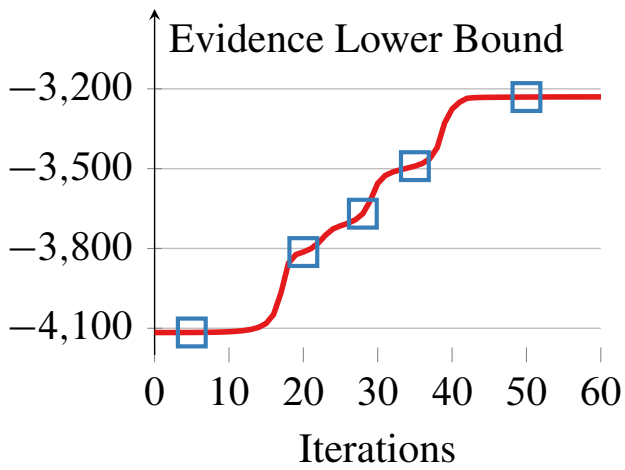
- VI turns **inference into optimization**.
- Posit a **variational family** of distributions over the latent variables,

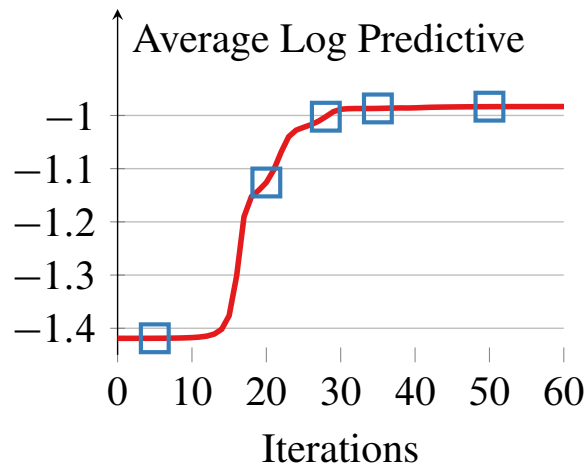
$$q(\mathbf{z}; \boldsymbol{\nu})$$

- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.
  (There are alternative divergences, which connect to algorithms like EP, BP, and others.)

Example: Mixture of Gaussians

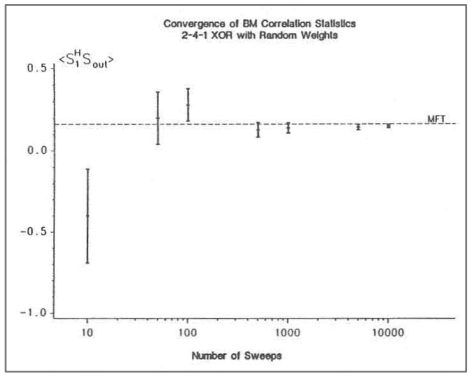




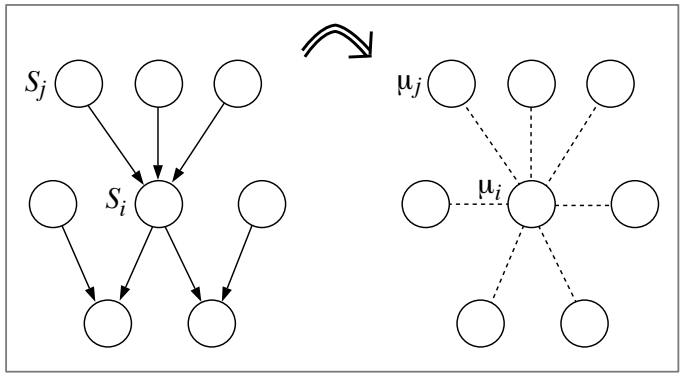


[images by Alp Kucukelbir; Blei+ 2016]

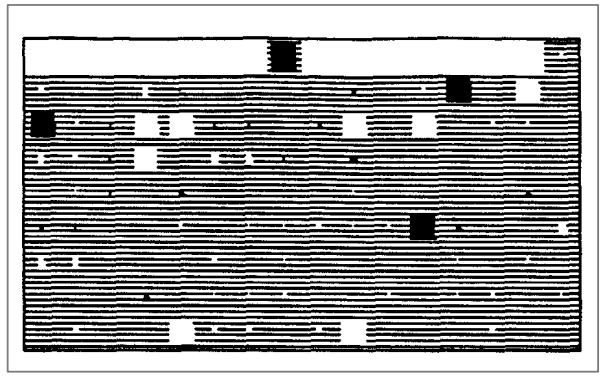
History



[Peterson and Anderson 1987]



[Jordan et al. 1999]

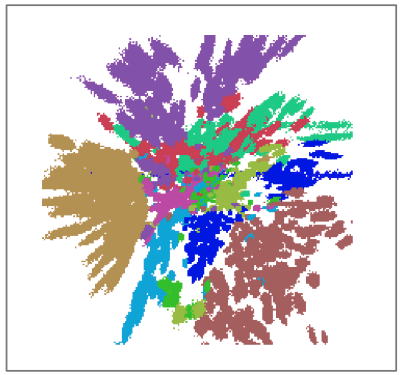


[Hinton and van Camp 1993]

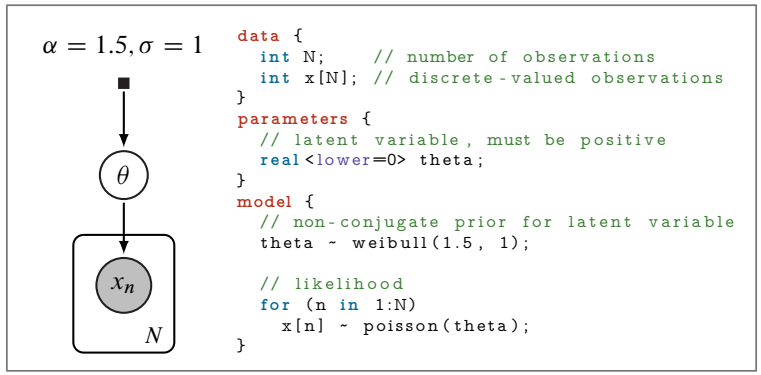
- Variational inference adapts **ideas from statistical physics** to probabilistic inference. Arguably, it began in the late eighties with Peterson and Anderson (1987), who used mean-field methods to fit a neural network.
- This idea was picked up by Jordan's lab in the early 1990s—Tommi Jaakkola, Lawrence Saul, Zoubin Gharamani—who **generalized it to many probabilistic models**. (A review paper is Jordan et al., 1999.)
- In parallel, Hinton and Van Camp (1993) also **developed mean-field for neural networks**. Neal and Hinton (1993) connected this idea to the EM algorithm, which lead to further variational methods for mixtures of experts (Waterhouse et al., 1996), HMMs (MacKay, 1997), and neural networks (Barber and Bishop, 1998).

This tutorial

[Kingma and Welling 2013]



[Rezende et al. 2014]



[Kucukelbir et al. 2015]

- There is now a flurry of new work on variational inference, making it scalable, easier to derive, faster, more accurate, and applying it to more complicated models and applications.
- Modern VI touches many important areas: probabilistic programming, reinforcement learning, neural networks, convex optimization, Bayesian statistics, and myriad applications.
- Our goal is to teach you the basics, explain some of the newer ideas, and to suggest open areas of new research.

Variational inference

Part I: Main ideas and historical context

Jordan+, *Introduction to Variational Methods for Graphical Models*, 1999

Part II: Mean-field VI and stochastic VI

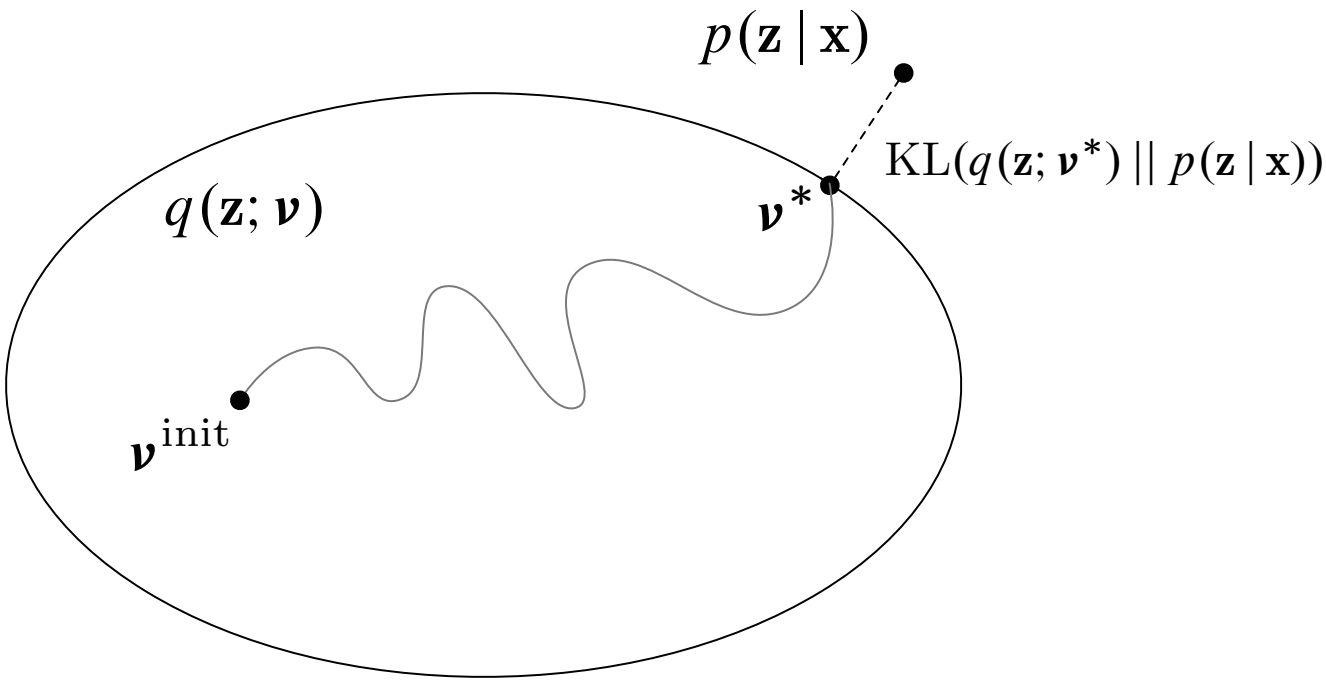
Ghahramani and Beal, *Propagation Algorithms for Variational Bayesian Learning*, 2001
Hoffman+, *Stochastic Variational Inference*, 2013

Part III: Stochastic gradients of the ELBO

Ranganath+, *Black Box Variational Inference*, 2014
Rezende+, *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*, 2014
Kucukelbir+ *Automatic Differentiation Variational Inference*, 2016

Part IV: Summary

Variational inference



VI approximates difficult quantities from complex models.

With **stochastic optimization** we can

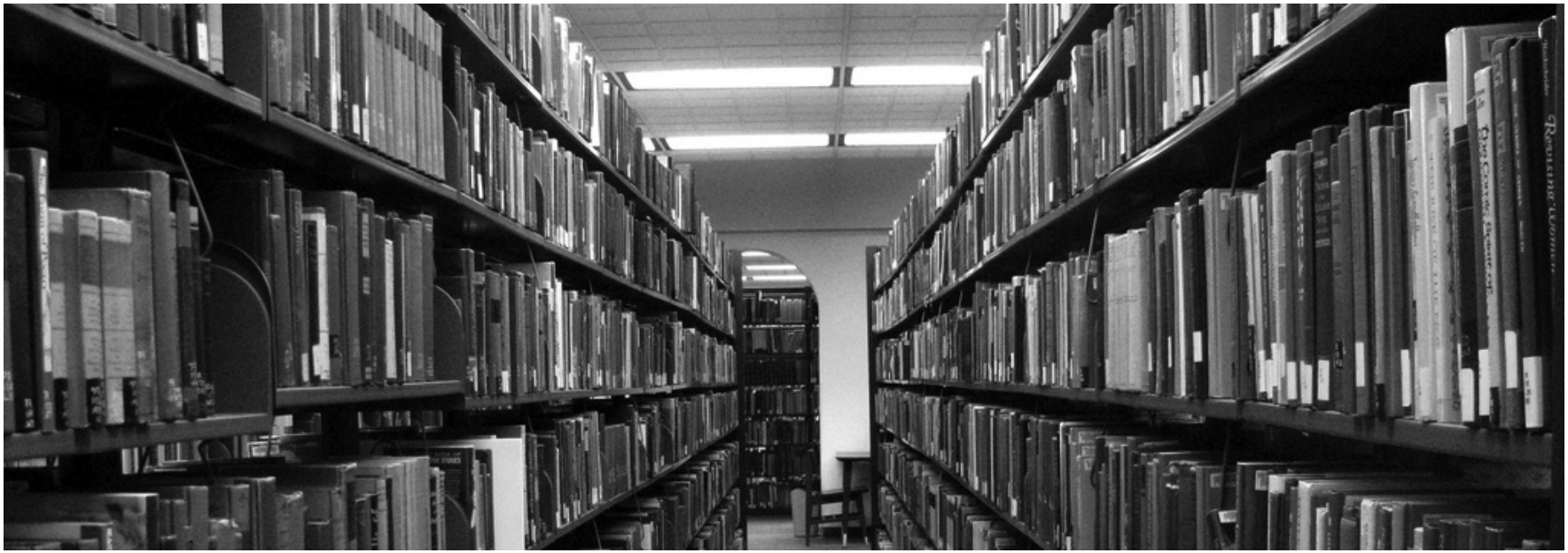
- scale up VI to massive data
- enable VI on a wide class of difficult models
- enable VI with elaborate and flexible families of approximations

PART II

Mean-field variational inference and stochastic variational inference

Mean-field variational inference casts Bayesian computation as optimization. *Stochastic variational inference* scales to massive data.

Motivation: Topic Modeling



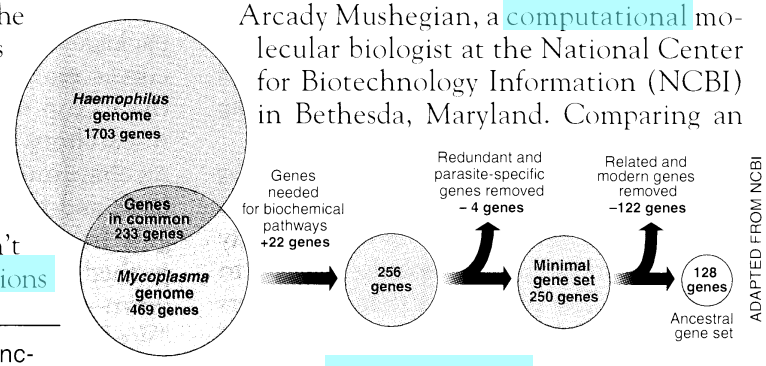
Topic models use posterior inference to discover the hidden thematic structure in a large collection of documents.

Example: Latent Dirichlet Allocation (LDA)

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 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 Siv Andersson of 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



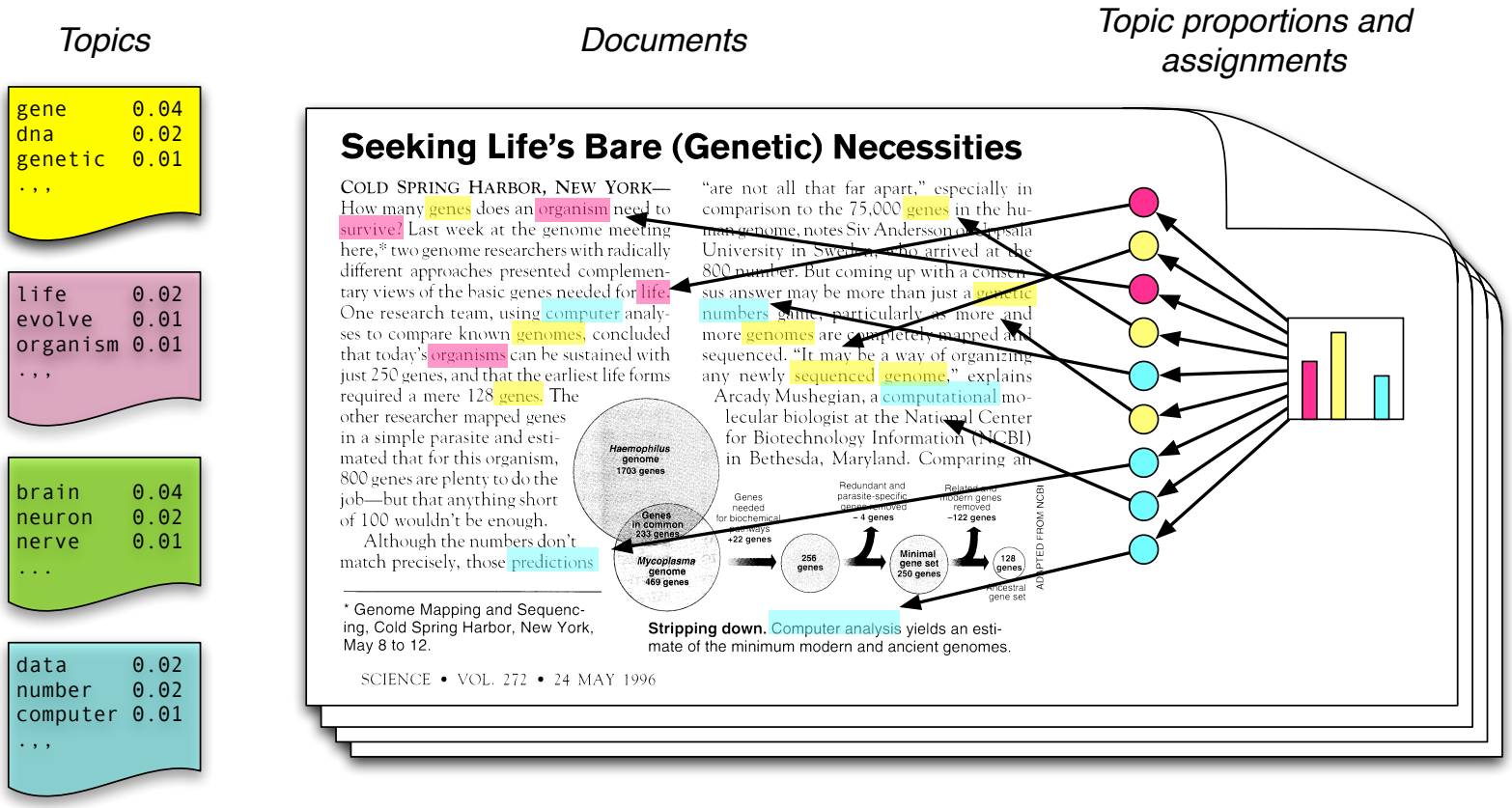
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

SCIENCE • VOL. 272 • 24 MAY 1996

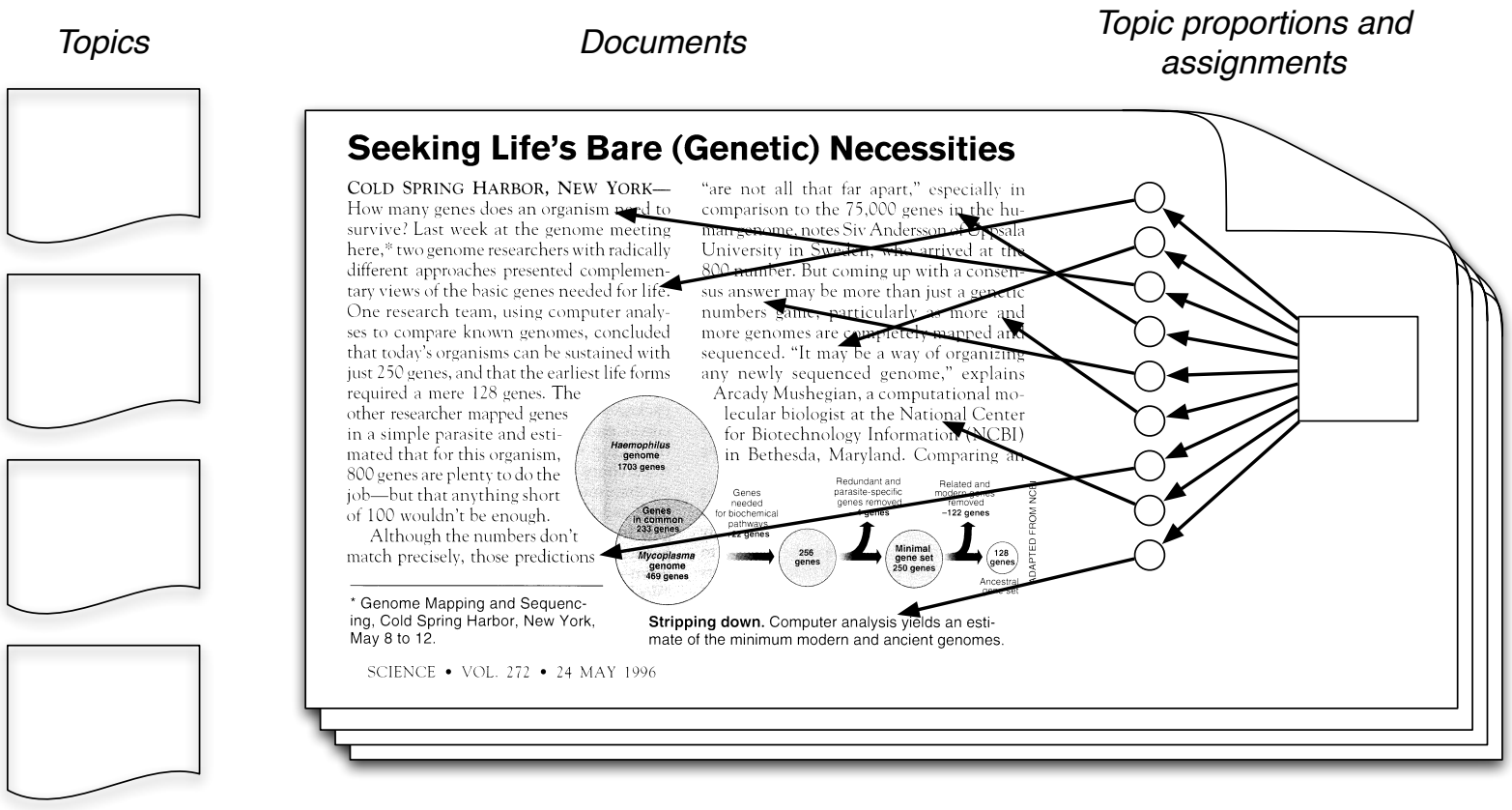
Documents exhibit multiple topics.

Example: Latent Dirichlet Allocation (LDA)



- 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

Example: Latent Dirichlet Allocation (LDA)

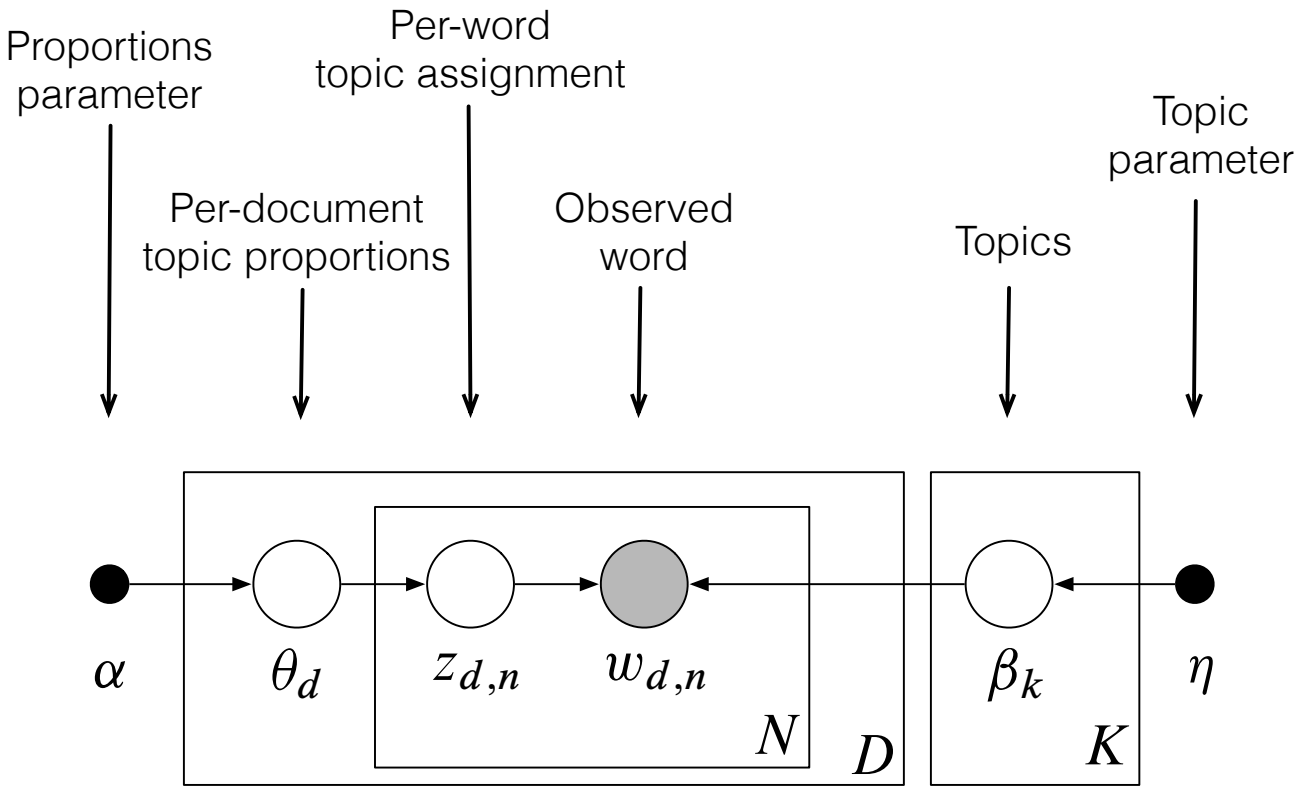


- But we only observe the documents; everything else is hidden.
- So we want to calculate the posterior

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

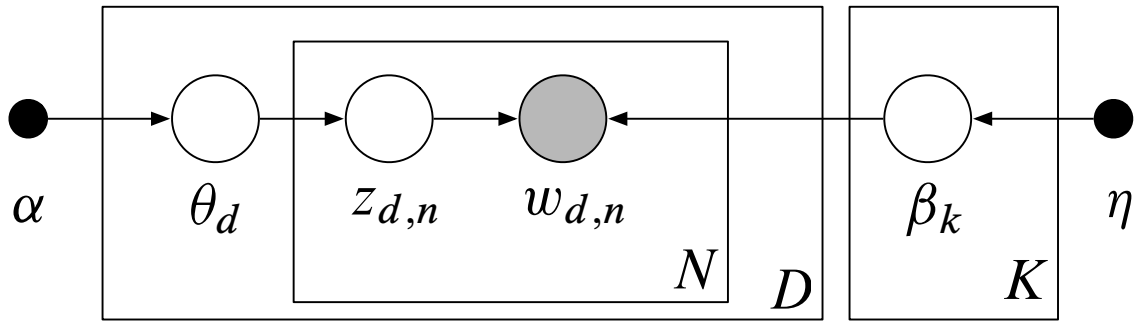
(Note: millions of documents; billions of latent variables)

LDA as a Graphical Model



- Encodes **assumptions** about data with a factorization of the joint
- Connects assumptions to **algorithms** for computing with data
- Defines the **posterior** (through the joint)

Posterior Inference



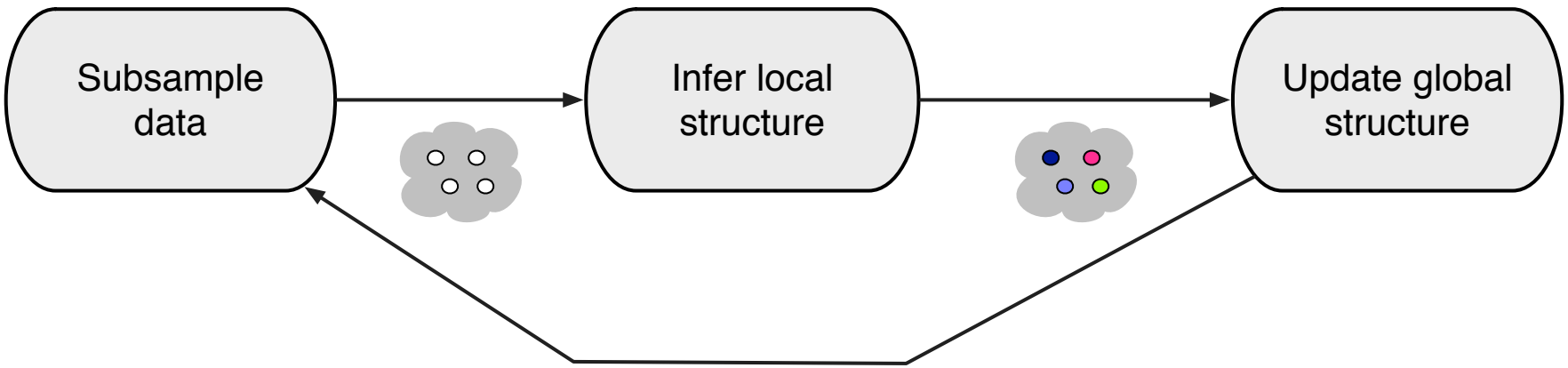
- The posterior of the latent variables given the documents is

$$p(\beta, \boldsymbol{\theta}, \mathbf{z} \mid \mathbf{w}) = \frac{p(\beta, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w})}{\int_\beta \int_{\boldsymbol{\theta}} \sum_{\mathbf{z}} p(\beta, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w})}.$$

- We can't compute the denominator, the marginal $p(\mathbf{w})$.
- We use approximate inference.

Topics found in 1.8M articles from the New York Times

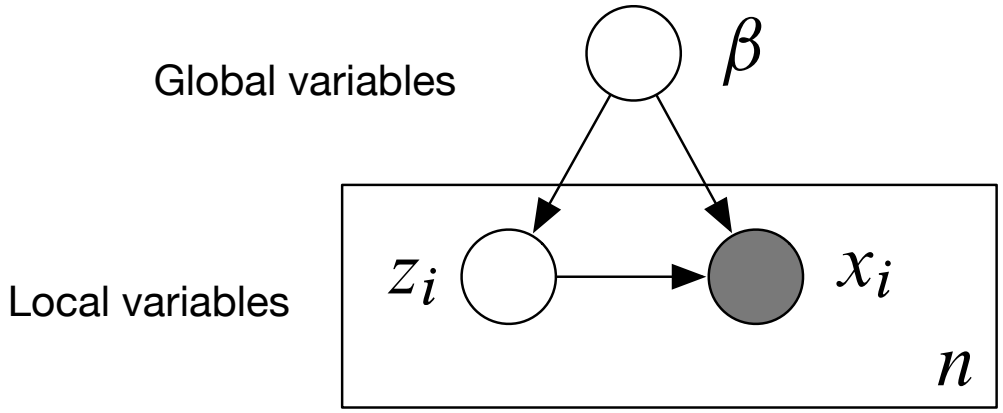
Mean-field VI and Stochastic VI



Road map:

- Define the generic class of conditionally conjugate models
- Derive classical mean-field VI
- Derive stochastic VI, which scales to massive data

Conditionally conjugate models

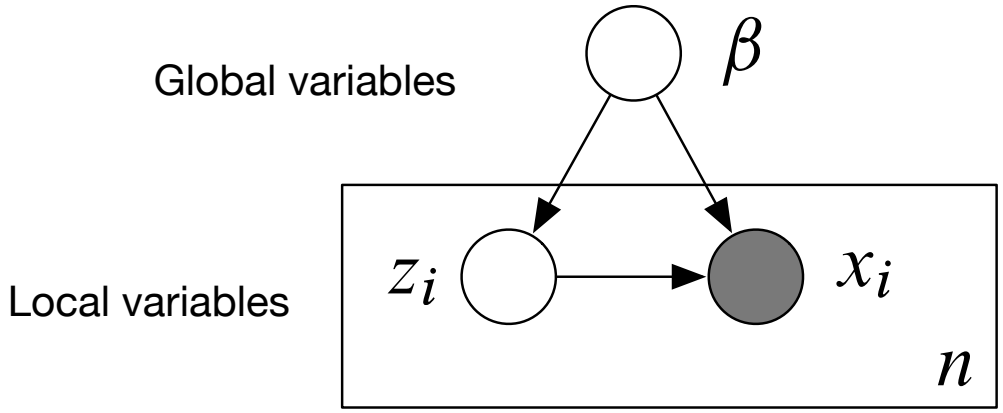


$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- The observations are $\mathbf{x} = x_{1:n}$.
- The **local** variables are $\mathbf{z} = z_{1:n}$.
- The **global** variables are β.
- The ith data point x_i only depends on z_i and β.

Compute $p(\beta, \mathbf{z} \mid \mathbf{x})$.

Conditionally conjugate models

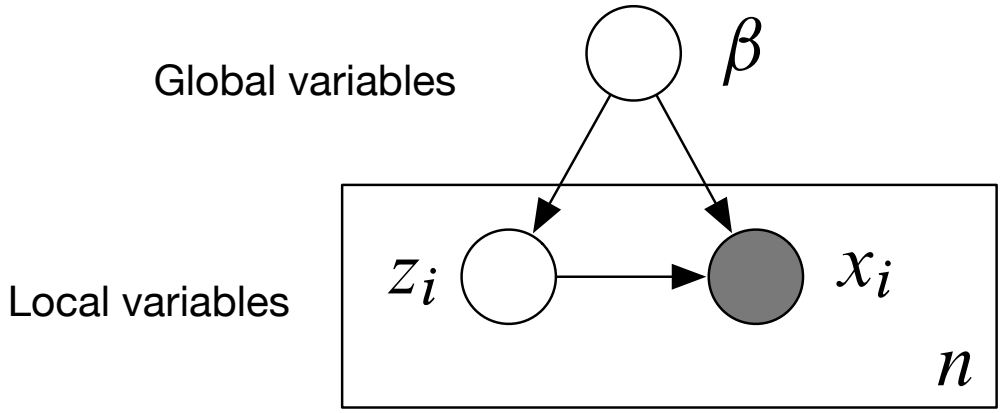


$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- A **complete conditional** is the conditional of a latent variable given the observations and other latent variables.
- Assume each complete conditional is in the exponential family,

$$p(z_i \mid \beta, x_i) = h(z_i) \exp\{\eta_\ell(\beta, x_i)^\top z_i - a(\eta_\ell(\beta, x_i))\}$$
$$p(\beta \mid \mathbf{z}, \mathbf{x}) = h(\beta) \exp\{\eta_g(\mathbf{z}, \mathbf{x})^\top \beta - a(\eta_g(\mathbf{z}, \mathbf{x}))\}.$$

Conditionally conjugate models



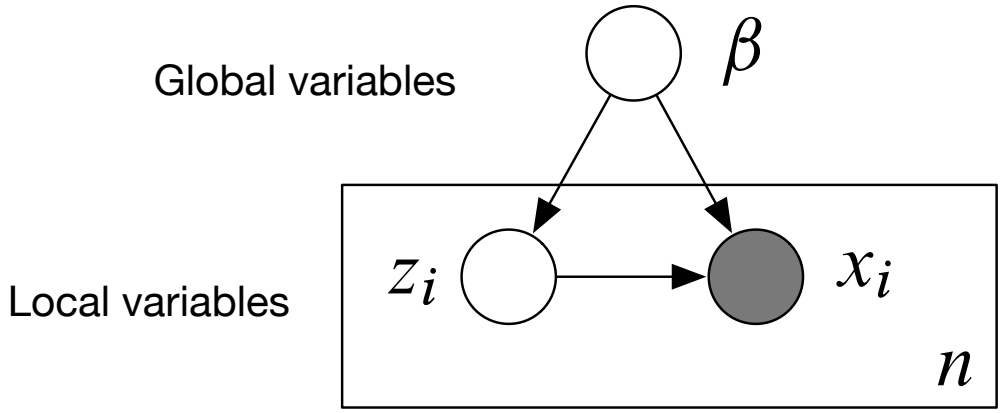
$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- A **complete conditional** is the conditional of a latent variable given the observations and other latent variable.
- The global parameter comes from conjugacy [Bernardo and Smith, 1994]

$$\eta_g(\mathbf{z}, \mathbf{x}) = \alpha + \textstyle\sum_{i=1}^{n} t(z_i, x_i),$$

where α is a hyperparameter and $t(\cdot)$ are sufficient statistics for $[z_i, x_i]$.

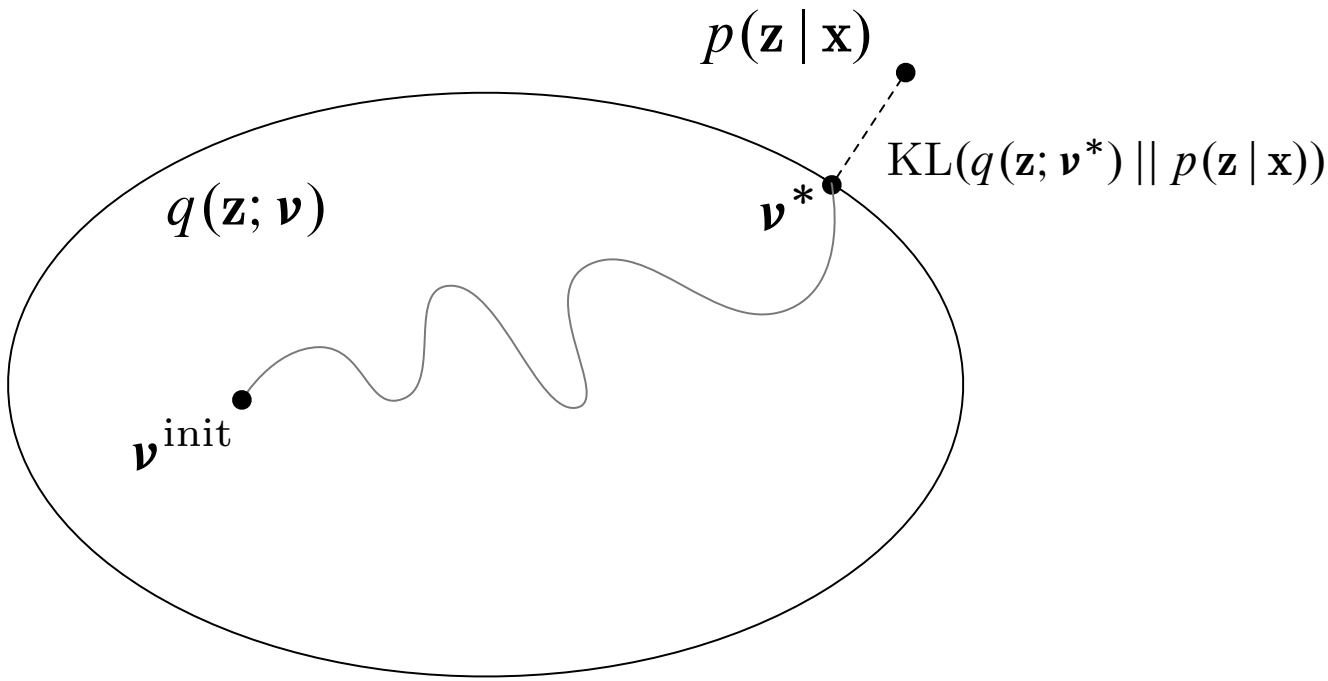
Conditionally conjugate models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Bayesian mixture models
- Time series models
  (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization
  (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression
  (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models
  (LDA and some variants)

Variational inference



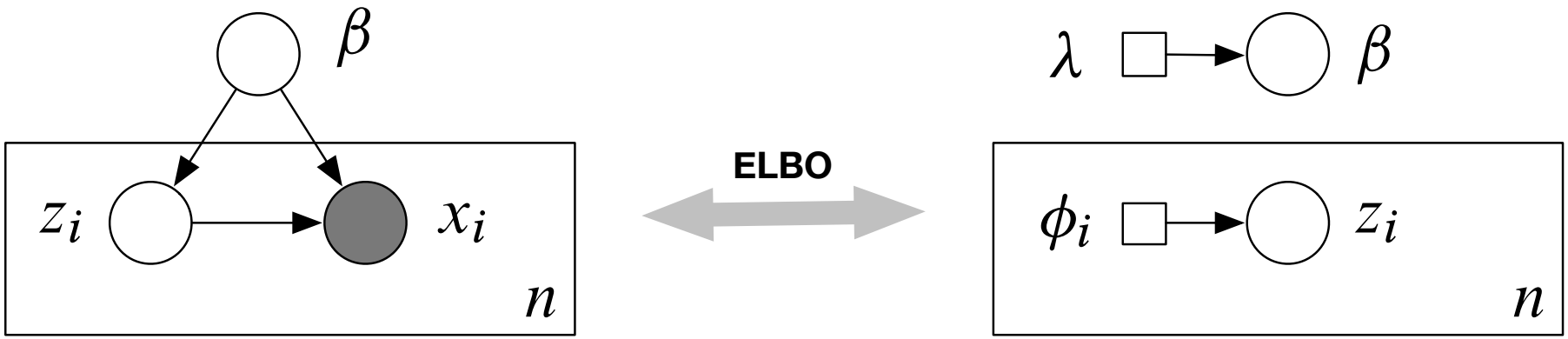
Minimize KL between $q(\beta, \mathbf{z}; \boldsymbol{\nu})$ and the posterior $p(\beta, \mathbf{z} \mid \mathbf{x})$.

The evidence lower bound

$$\mathcal{L}(\boldsymbol{\nu}) = \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\beta, \mathbf{z}; \boldsymbol{\nu})]$$

- KL is intractable; VI optimizes the **evidence lower bound** (ELBO) instead.
  - It is a lower bound on $\log p(\mathbf{x})$.
  - Maximizing the ELBO is equivalent to minimizing the KL.
- The ELBO trades off two terms.
  - The first term prefers $q(\cdot)$ to place its mass on the MAP estimate.
  - The second term encourages $q(\cdot)$ to be diffuse.
- Caveat: The ELBO is not convex.

Mean-field variational inference



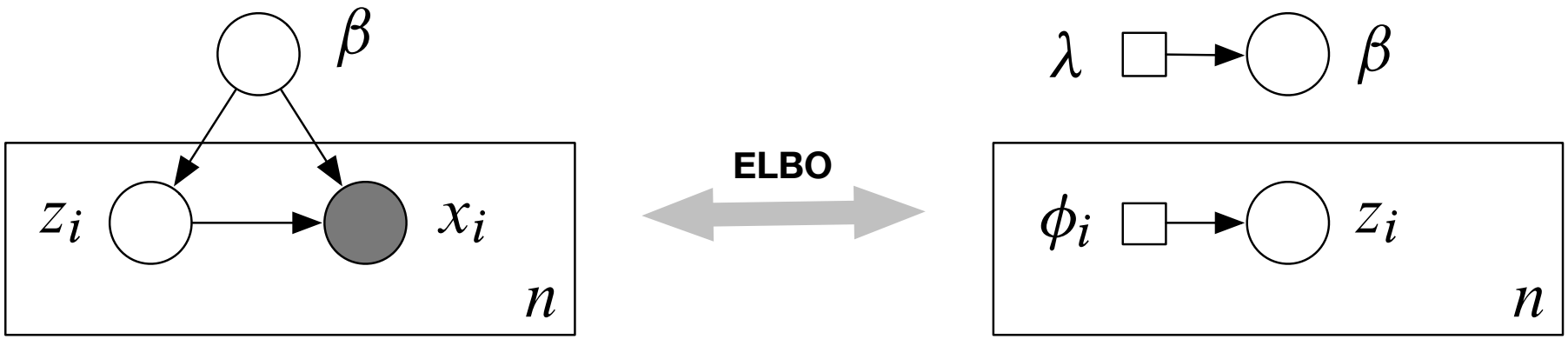
- We need to specify the form of $q(\beta, \mathbf{z})$.
- The **mean-field family** is fully factorized,

$$q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi}) = q(\beta; \lambda) \prod_{i=1}^{n} q(z_i; \phi_i).$$

- Each factor is the same family as the model's complete conditional,

$$p(\beta \mid \mathbf{z}, \mathbf{x}) = h(\beta) \exp\{\eta_g(\mathbf{z}, \mathbf{x})^\top \beta - a(\eta_g(\mathbf{z}, \mathbf{x}))\}$$
$$q(\beta; \lambda) = h(\beta) \exp\{\lambda^\top \beta - a(\lambda)\}.$$

Mean-field variational inference



- Optimize the ELBO,

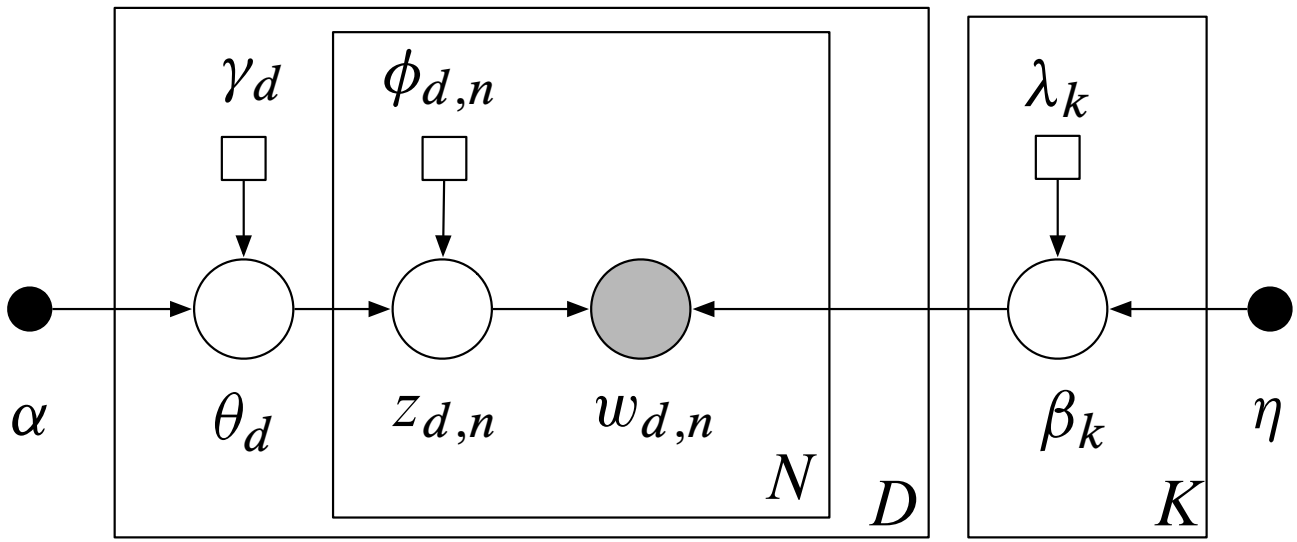
$$\mathcal{L}(\lambda, \boldsymbol{\phi}) = \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\beta, \mathbf{z})].$$

- Traditional VI uses coordinate ascent [Ghahramani and Beal, 2001]

$$\lambda^* = \mathbb{E}_\phi\left[\eta_g(\mathbf{z}, \mathbf{x})\right];\ \phi_i^* = \mathbb{E}_\lambda[\eta_\ell(\beta, x_i)]$$

- Iteratively update each parameter, holding others fixed.
  - Notice the relationship to Gibbs sampling [Gelfand and Smith, 1990] .
  - Caveat: The ELBO is not convex.

Mean-field variational inference for LDA



- The local variables are the per-document variables θ_d and $\mathbf{z}_d$.
- The global variables are the topics $\beta_1, \ldots, \beta_K$.
- The variational distribution is

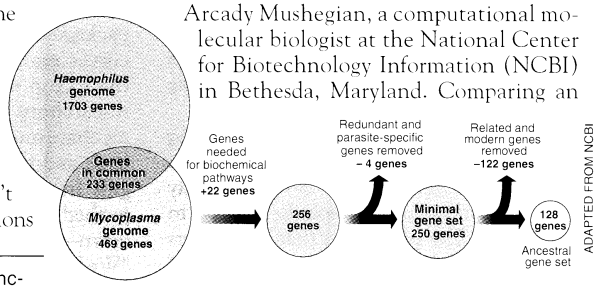
$$q(\beta, \boldsymbol{\theta}, \mathbf{z}) = \prod_{k=1}^{K} q(\beta_k; \lambda_k) \prod_{d=1}^{D} q(\theta_d; \gamma_d) \prod_{n=1}^{N} q(z_{d,n}; \phi_{d,n})$$

Mean-field variational inference for LDA

Seeking Life's Bare (Genetic) Necessities

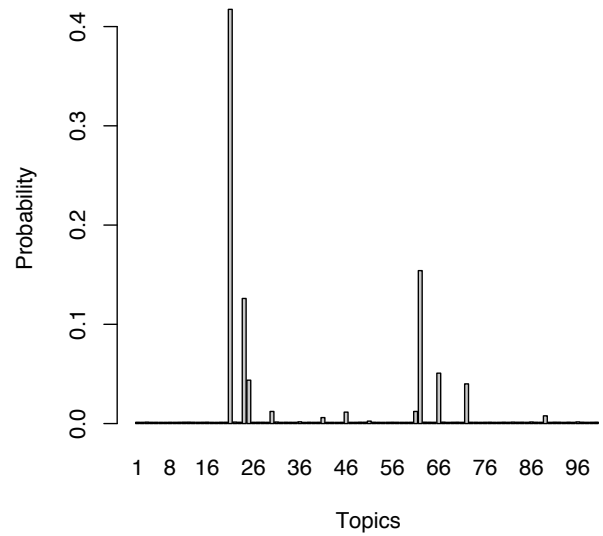
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known 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 Siv Andersson of 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.

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



Mean-field variational inference for LDA

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Classical variational inference

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly.

repeat

  for *each data point* i **do**

    Set local parameter $\phi_i \leftarrow \mathbb{E}_\lambda[\eta_\ell(\beta, x_i)]$.

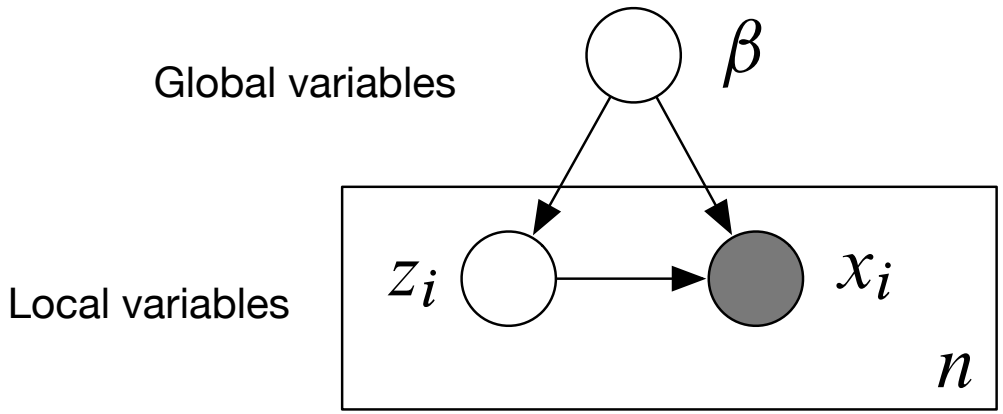
  end

  Set global parameter

$$\lambda \leftarrow \alpha + \sum_{i=1}^{n} \mathbb{E}_{\phi_i}[t(Z_i, x_i)].$$

until *the ELBO has converged*

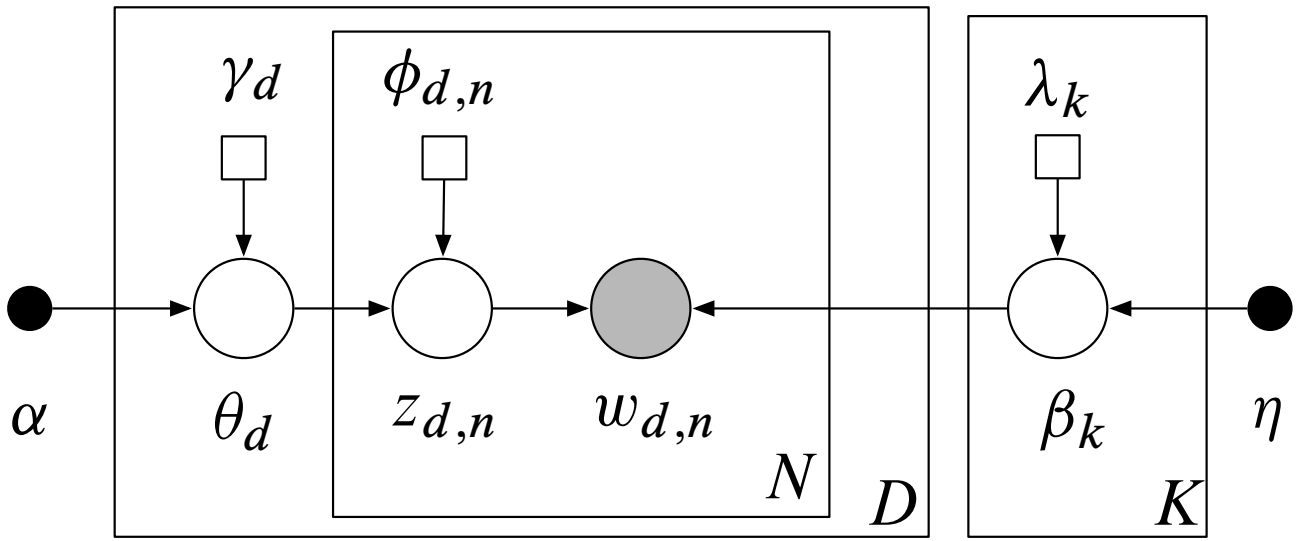
Conditionally conjugate models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

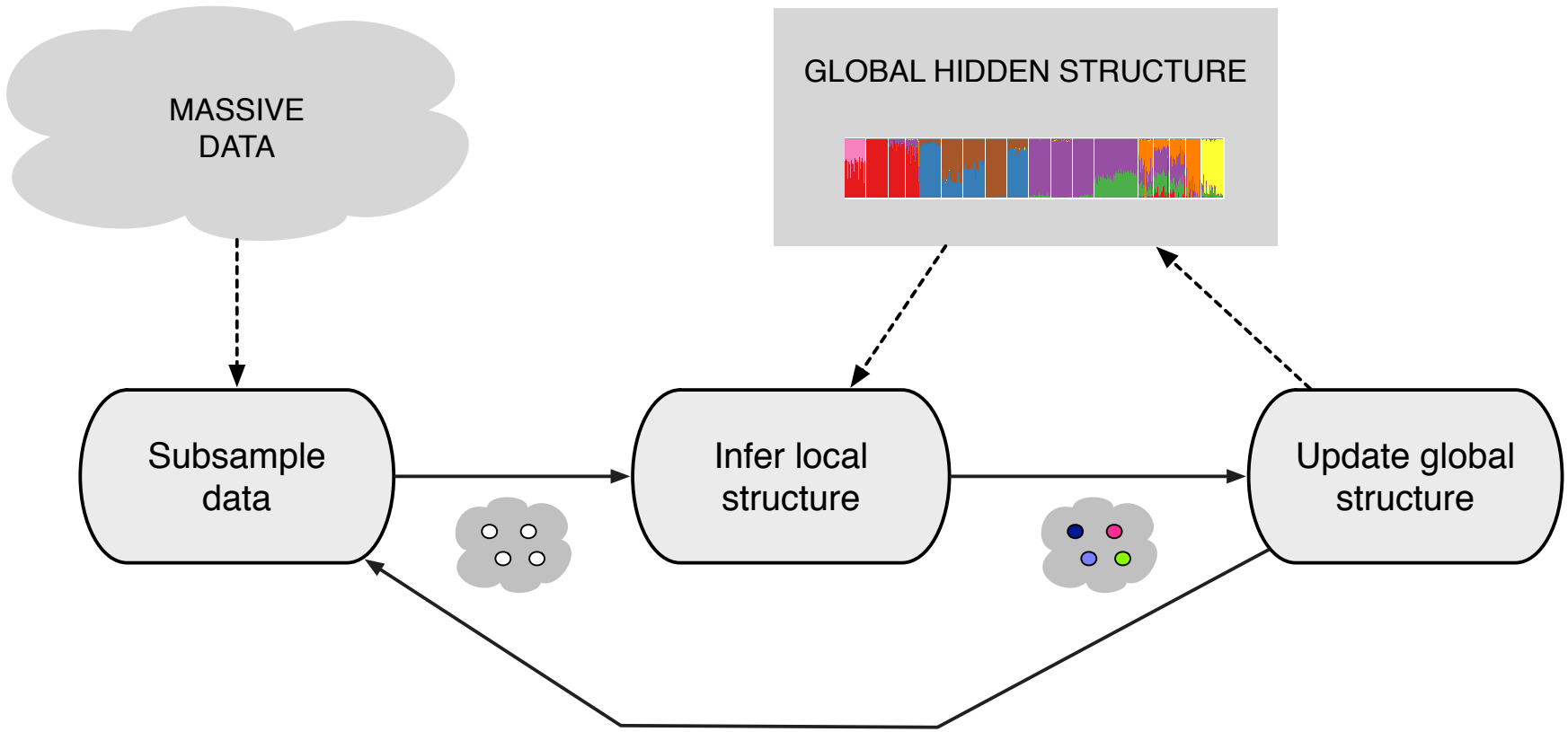
- Bayesian mixture models
- Time series models
  (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization
  (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression
  (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models
  (LDA and some variants)

Stochastic variational inference



- Classical VI is inefficient:
  - Do some local computation *for each data point*.
  - Aggregate these computations to re-estimate global structure.
  - Repeat.
- This cannot handle massive data.
- **Stochastic variational inference** (SVI) scales VI to massive data.

Stochastic variational inference



Stochastic optimization

A STOCHASTIC APPROXIMATION METHOD[1]

By Herbert Robbins and Sutton Monro

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.

- Replace the gradient with cheaper noisy estimates [Robbins and Monro, 1951]
- Guaranteed to converge to a local optimum [Bottou, 1996]
- Has enabled modern machine learning

Stochastic optimization

A STOCHASTIC APPROXIMATION METHOD[1]

By Herbert Robbins and Sutton Monro

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.

- With noisy gradients, update

$$\nu_{t+1} = \nu_t + \rho_t \hat{\nabla}_\nu \mathscr{L}(\nu_t)$$

- Requires unbiased gradients, $\mathbb{E}\left[\hat{\nabla}_\nu \mathscr{L}(\nu)\right] = \nabla_\nu \mathscr{L}(\nu)$
- Requires the step size sequence ρ_t follows the Robbins-Monro conditions

Stochastic variational inference

- The **natural gradient** of the ELBO [Amari, 1998; Sato, 2001]

$$\nabla_\lambda^{\text{nat}} \mathcal{L}(\lambda) = \left(\alpha + \sum_{i=1}^n \mathbb{E}_{\phi_i^*}[t(Z_i, x_i)]\right) - \lambda.$$

- Construct a **noisy natural gradient**,

$$j \sim \text{Uniform}(1, \ldots, n)$$
$$\hat{\nabla}_\lambda^{\text{nat}} \mathcal{L}(\lambda) = \alpha + n\mathbb{E}_{\phi_j^*}[t(Z_j, x_j)] - \lambda.$$

- This is a good noisy gradient.
  - Its expectation is the exact gradient (*unbiased*).
  - It only depends on optimized parameters of one data point (*cheap*).

Stochastic variational inference

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly. Set ρ_t appropriately.

repeat

Sample $j \sim \text{Unif}(1, \ldots, n)$.

Set local parameter $\phi \leftarrow \mathbb{E}_\lambda\left[\eta_\ell(\beta, x_j)\right]$.

Set intermediate global parameter

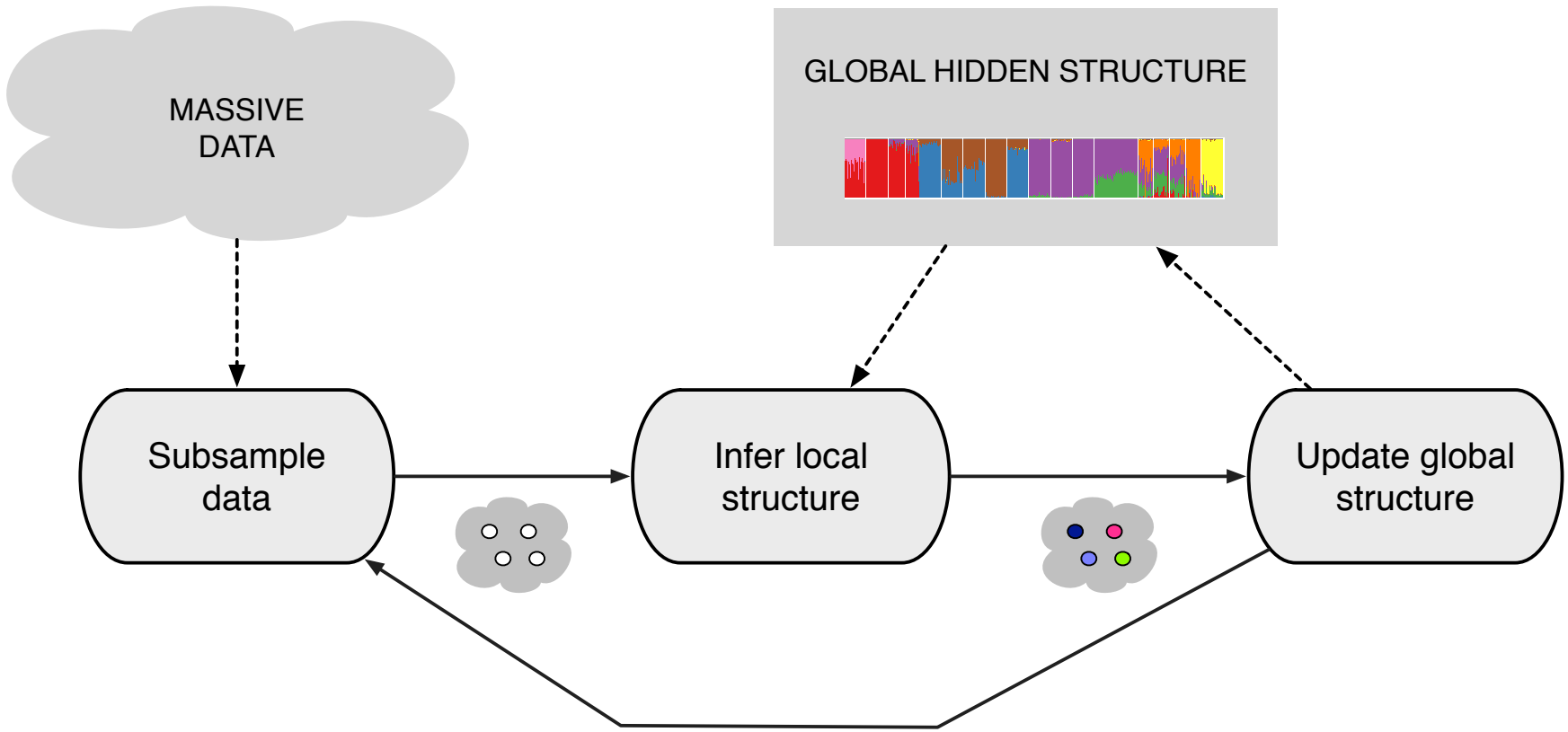
$$\hat{\lambda} = \alpha + n\mathbb{E}_\phi[t(Z_j, x_j)].$$

Set global parameter

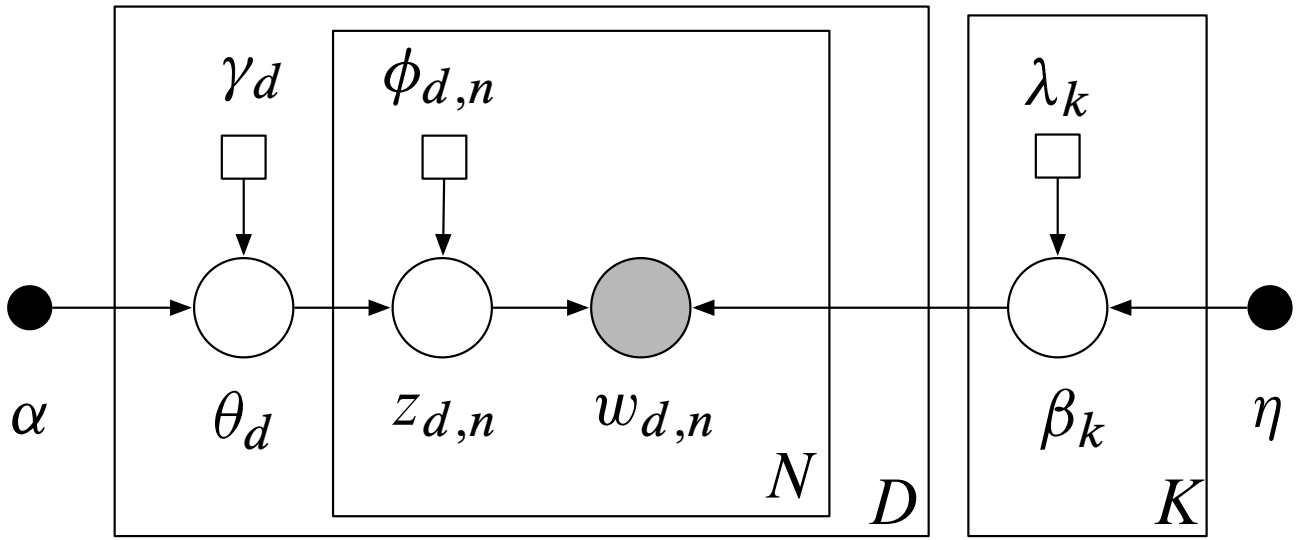
$$\lambda = (1 - \rho_t)\lambda + \rho_t\hat{\lambda}.$$

until *forever*

Stochastic variational inference

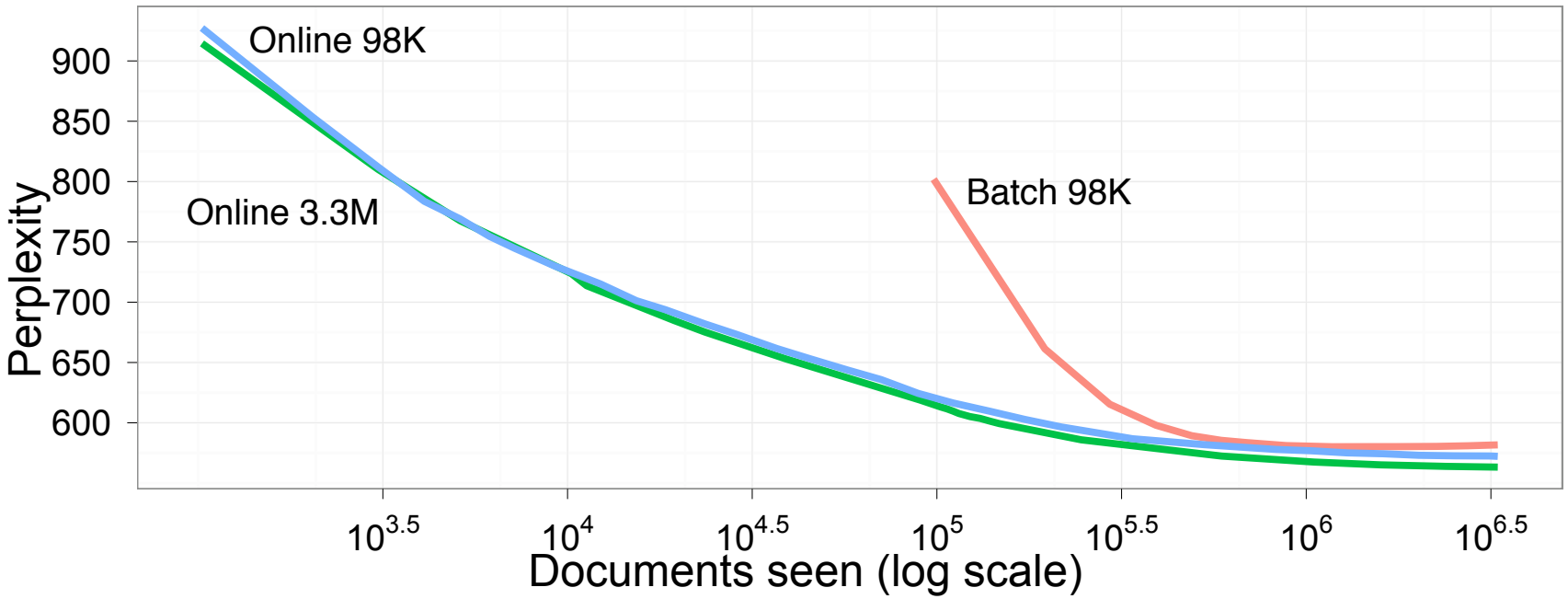


Stochastic variational inference for LDA



- Sample a document
- Estimate the local variational parameters using the current topics
- Form intermediate topics from those local parameters
- Update topics as a weighted average of intermediate and current topics

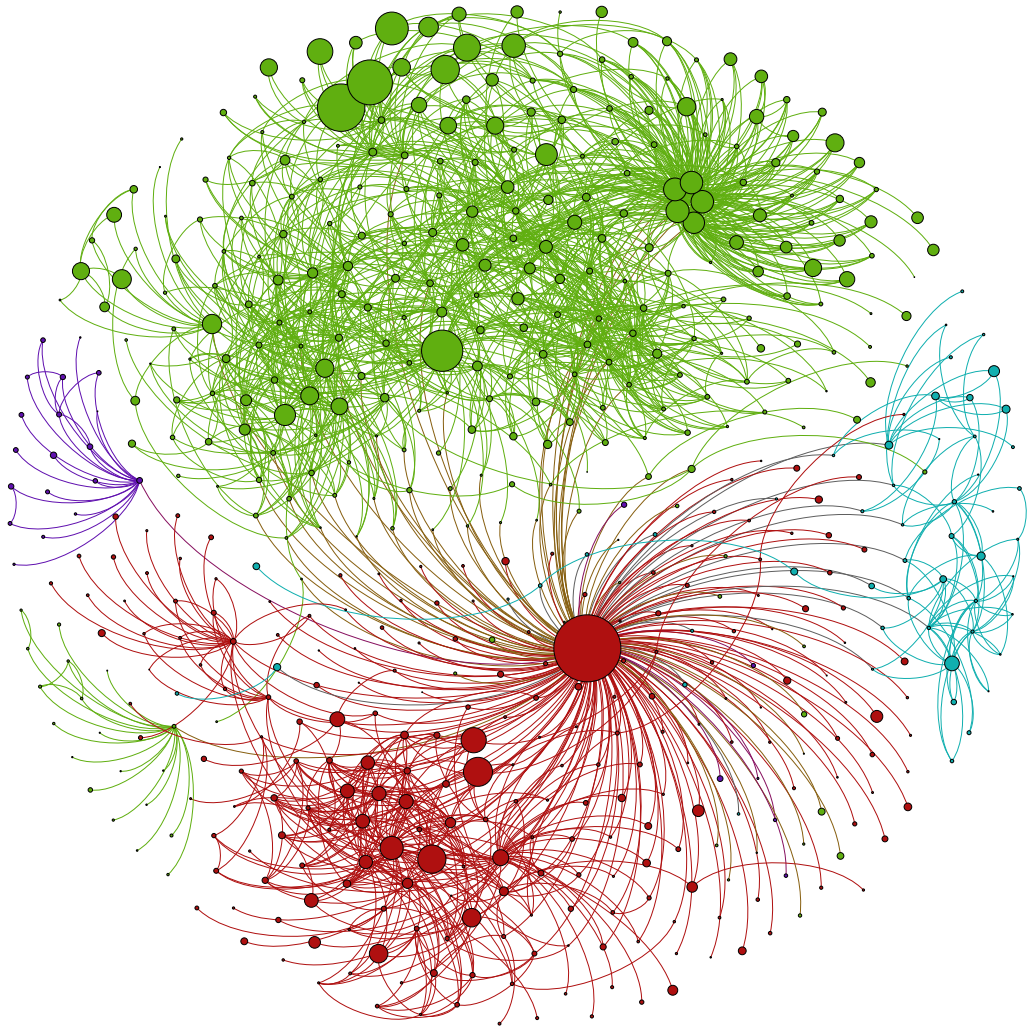
Stochastic variational inference for LDA



Documents analyzed	2048	4096	8192	12288	16384	32768	49152	65536
Top eight words	systems	systems	service	service	service	business	business	business
	road	health	systems	systems	companies	service	service	industry
	made	communication	health	companies	systems	companies	companies	service
	service	service	companies	business	business	industry	industry	companies
	announced	billion	market	company	company	company	services	services
	national	language	communication	billion	industry	management	company	company
	west	care	company	health	market	systems	management	management
	language	road	billion	industry	billion	services	public	public

[Hoffman et al., 2010]

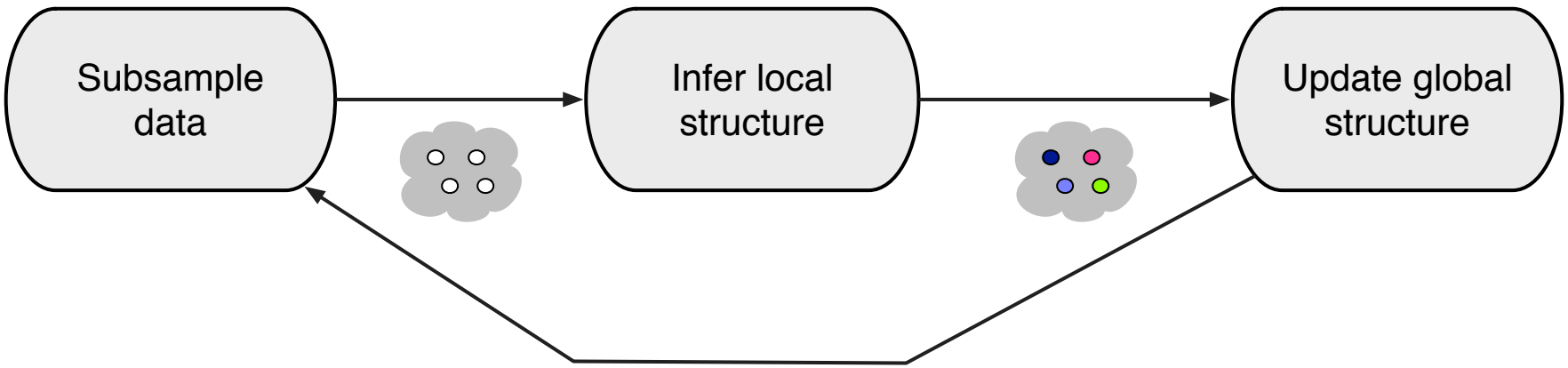
Topics using the HDP, found in 1.8M articles from the New York Times



Communities discovered in a 3.7M node network of U.S. Patents

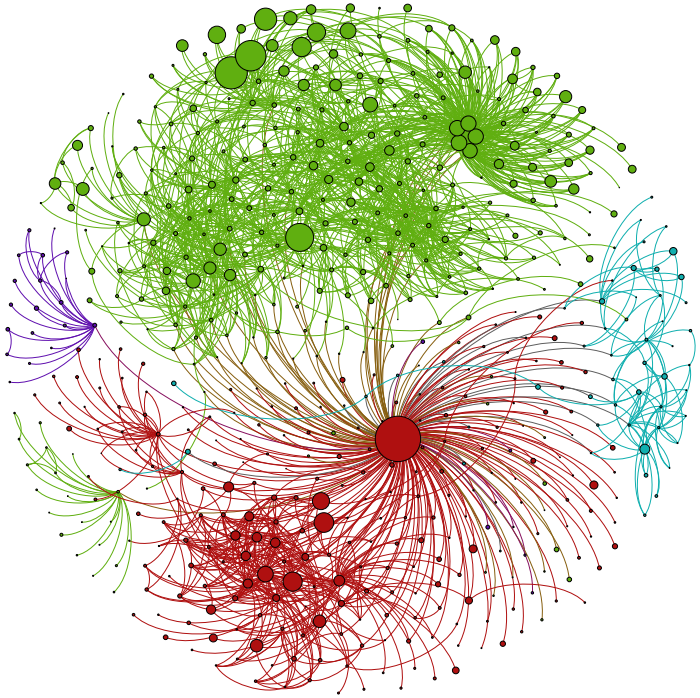
[Gopalan and Blei, PNAS 2013]

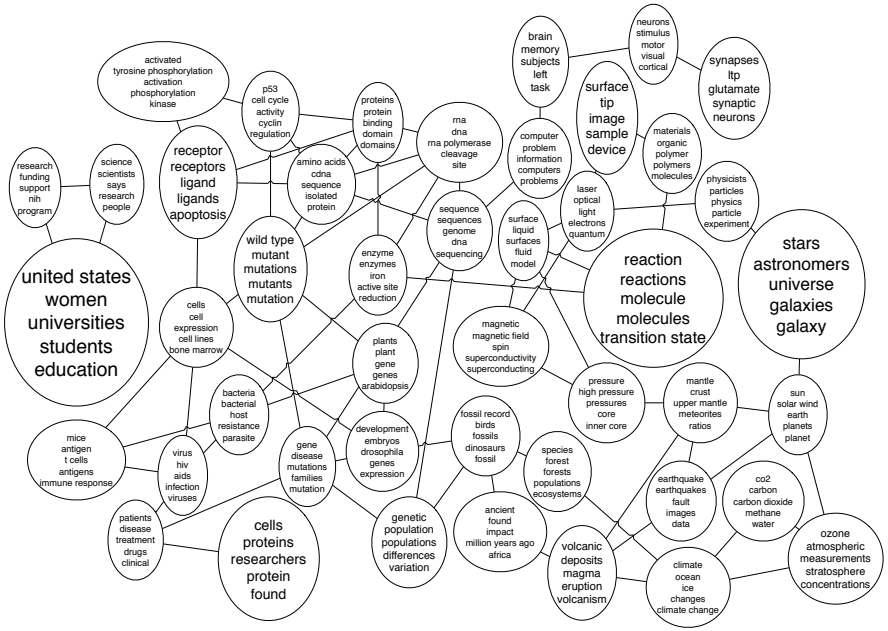
SVI scales many models

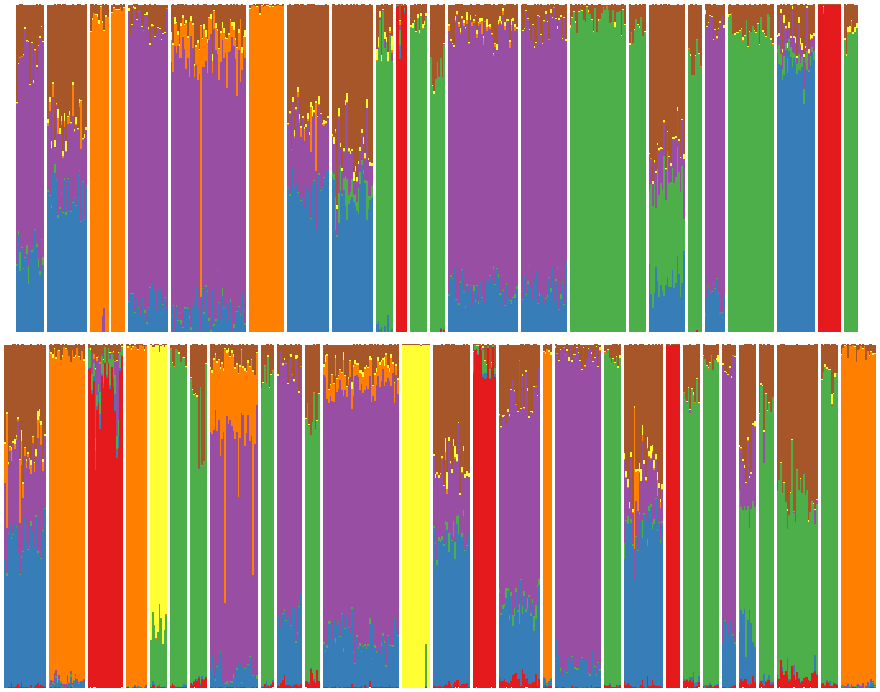


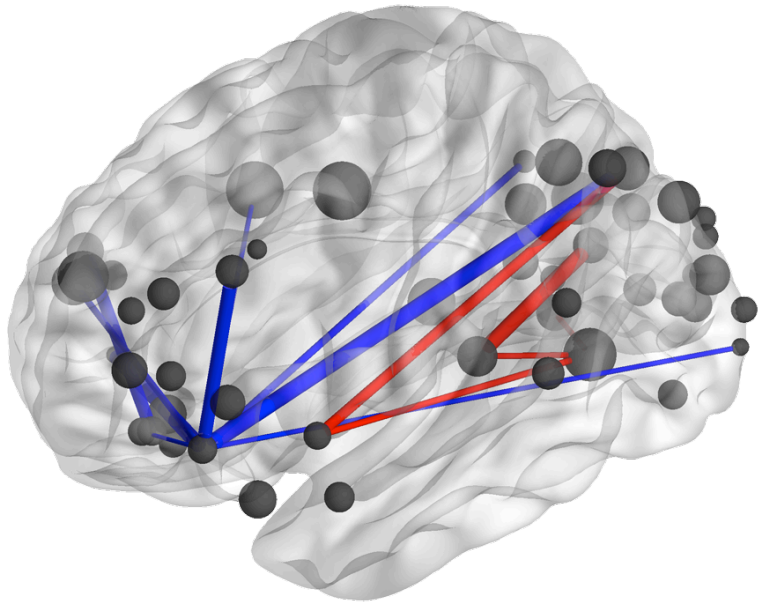
- Bayesian mixture models
- Time series models (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models (LDA and some variants)

Mean-field variational inference casts Bayesian computation as optimization. *Stochastic variational inference* scales to massive data.




activated tyrosine phosphorylation activation phosphorylation kinase
p53 cell cycle activity cyclin regulation
proteins protein binding domain domains
rna dna rna polymerase cleavage site
brain memory subjects left task
neurons stimulus motor visual cortical
synapses ltp glutamate synaptic neurons
surface tip image sample device
computer problem information computers problems
materials organic polymer polymers molecules
physicists particles physics particle experiment
laser optical light electrons quantum
research funding support nih program
science scientists says research people
receptor receptors ligand ligands apoptosis
amino acids cdna sequence isolated protein
sequence sequences genome dna sequencing
surface liquid surfaces fluid model
wild type mutant mutations mutants mutation
enzyme enzymes iron active site reduction
reaction reactions molecule molecules transition state
stars astronomers universe galaxies galaxy
united states women universities students education
cells cell expression cell lines bone marrow
plants plant gene genes arabidopsis
magnetic magnetic field spin superconductivity superconducting
pressure high pressure pressures core inner core
mantle crust upper mantle meteorites ratios
sun solar wind earth planets planet
bacteria bacterial host resistance parasite
mice antigen t cells antigens immune response
virus hiv aids infection viruses
gene disease mutations families mutation
development embryos drosophila genes expression
fossil record birds fossils dinosaurs fossil
species forest forests populations ecosystems
earthquake earthquakes fault images data
co2 carbon carbon dioxide methane water
patients disease treatment drugs clinical
cells proteins researchers protein found
genetic population populations differences variation
ancient found impact million years ago africa
volcanic deposits magma eruption volcanism
climate ocean ice changes climate change
ozone atmospheric measurements stratosphere concentrations



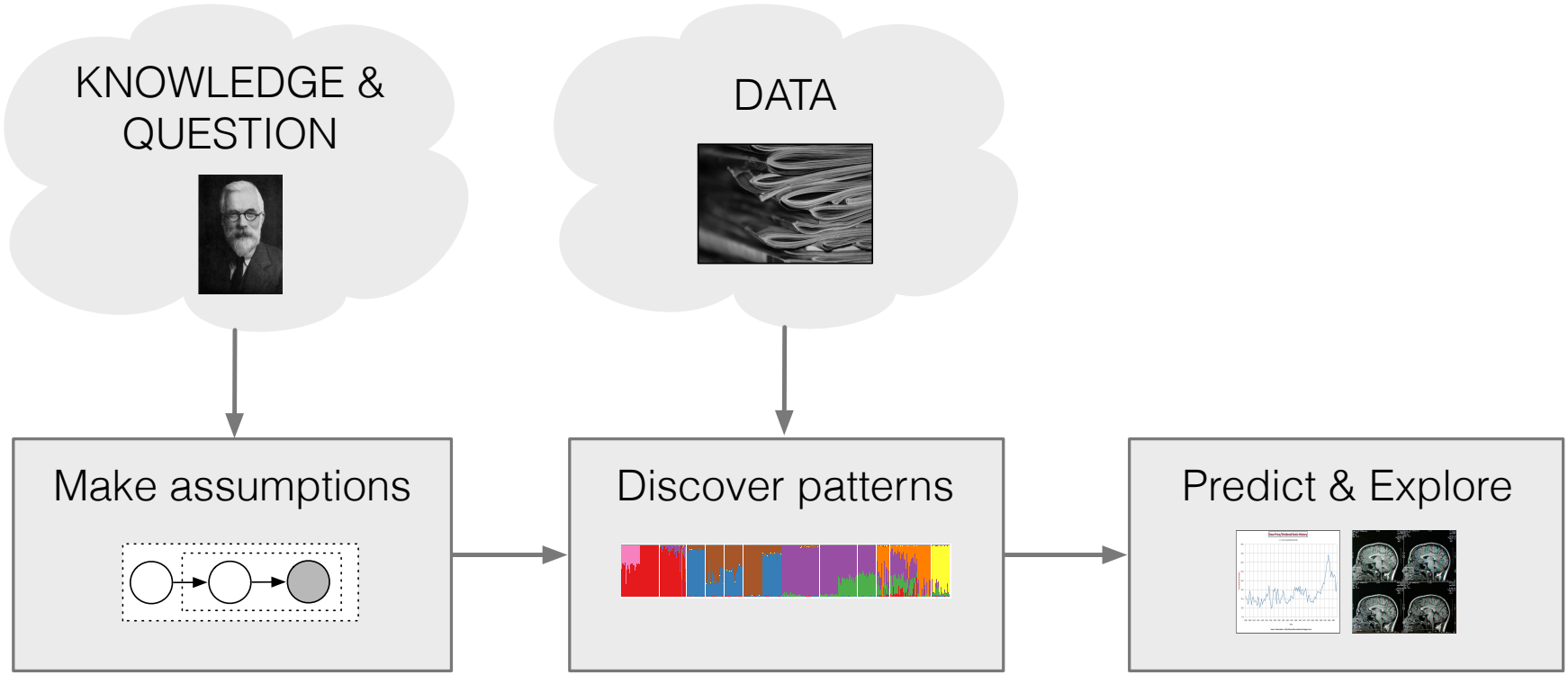


PART III

Black box variational inference

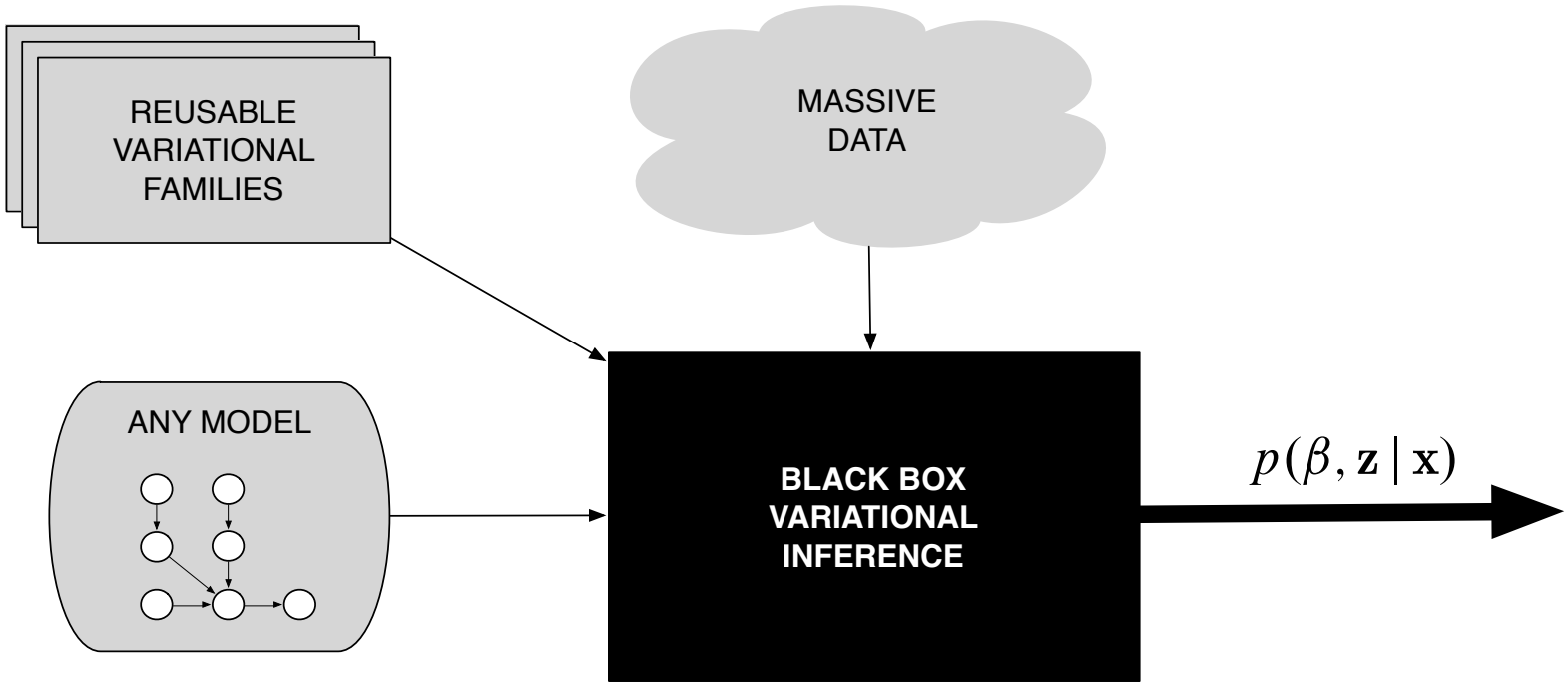
Monte Carlo gradients enable *black box variational inference*, algorithms that efficiently perform Bayesian computation in any model.

Black box variational inference



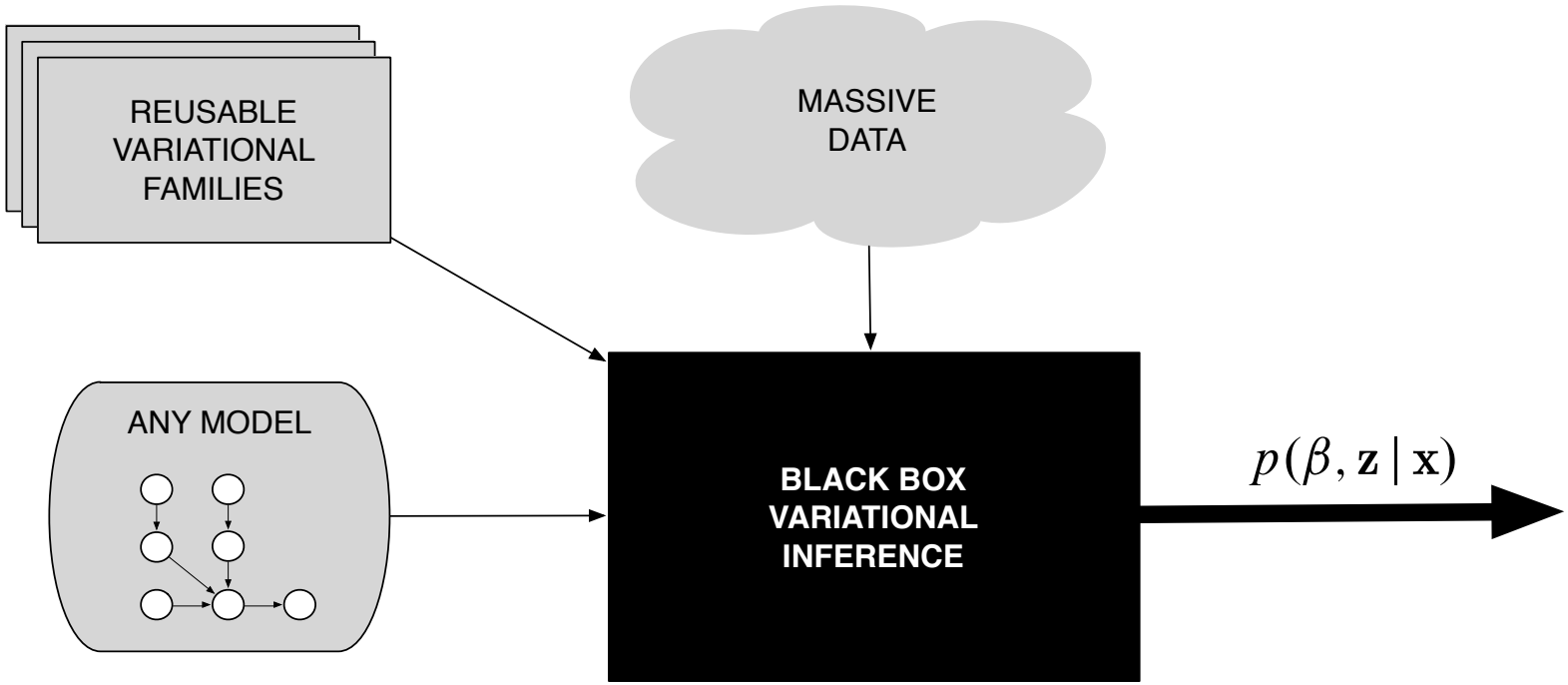
- Approximate inference can be difficult to derive.
- Especially true for models that are not conditionally conjugate
- E.g., discrete choice models, Bayesian generalized linear models, ...

Black box variational inference



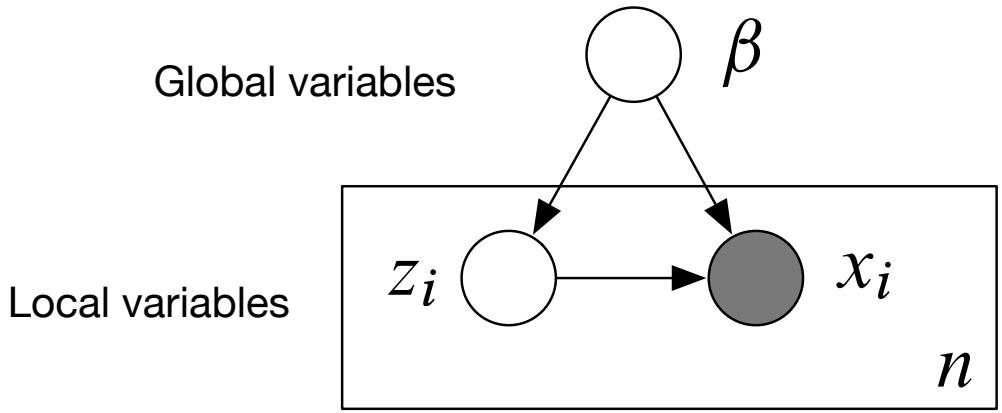
- Easily use variational inference with *any model*
- No exponential family requirements
- No mathematical work beyond specifying the model

Black box variational inference



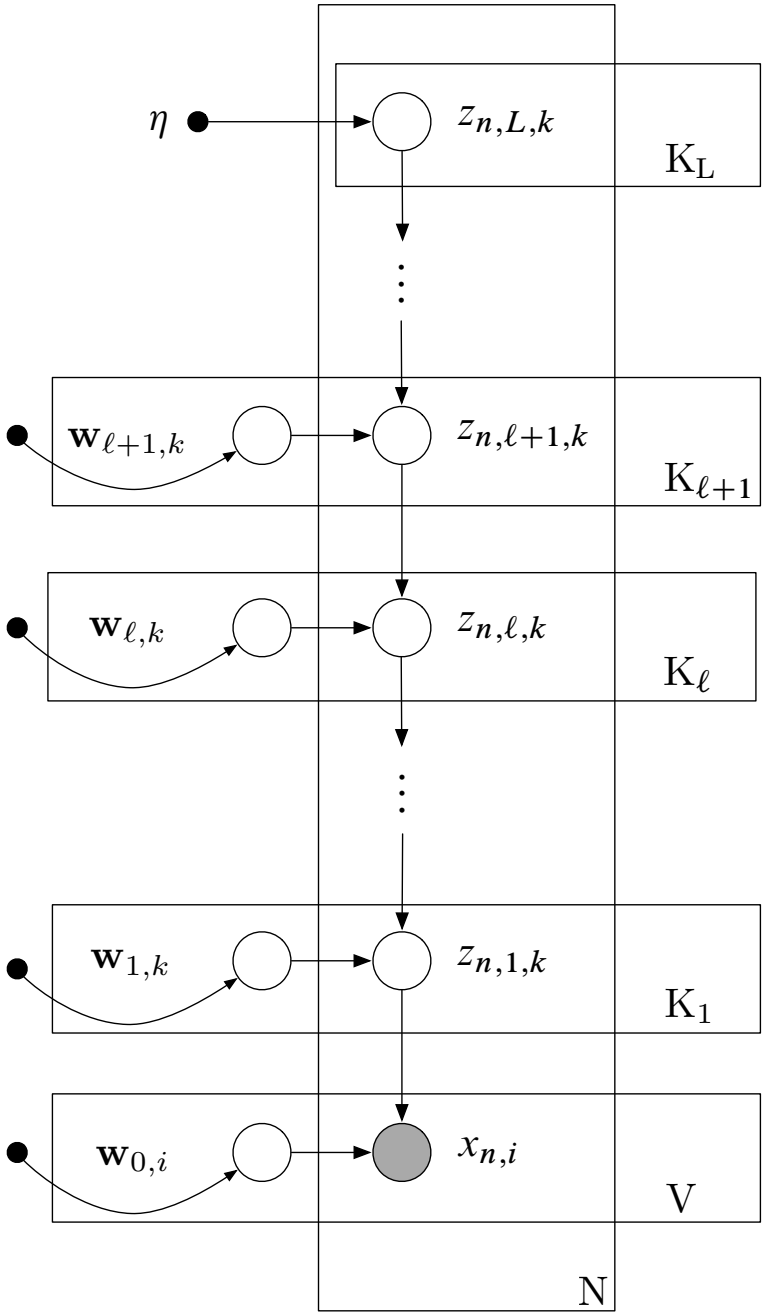
- Sample from $q(\cdot)$ (or a related distribution)
- Form noisy gradients without model-specific computation
- Use stochastic optimization

Nonconjugate models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Nonlinear time series models
- Deep latent Gaussian models
- Models with attention
- Generalized linear models
- Stochastic volatility models
- Discrete choice models
- Bayesian neural networks
- Deep exponential families
- Correlated topic models
- Sigmoid belief networks



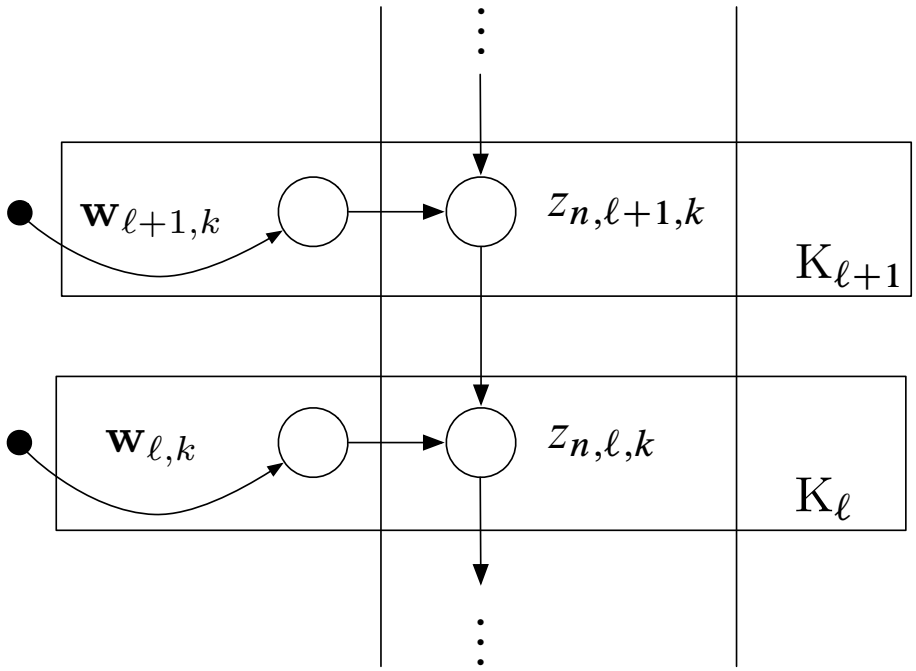
$z_{n,L,k} \sim \textsc{Exp-Fam}(\eta)$

$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$

$x_{n,i} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{0,i}^\top \mathbf{z}_{n,1}))$

[Ranganath+ 2015]

Deep exponential families



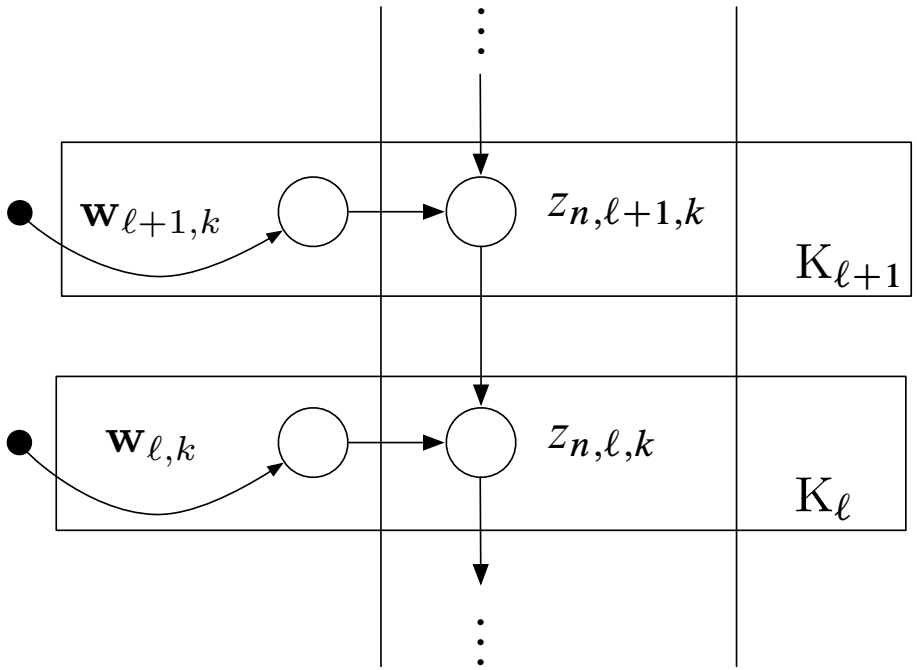
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

All distributions are in canonical exponential family form

$$p(z_{n,\ell,k} \mid \mathbf{z}_{n,\ell+1}, \mathbf{w}_{\ell,k}) = \exp\{\eta(\cdot)^\top t(z_{n,\ell,k}) - a(\eta(\cdot))\}$$
$$\eta(\cdot) = g(\mathbf{z}_{n,\ell+1}^\top \mathbf{w}_{\ell,k}).$$

(Note: Inner product is not strictly necessary.)

Deep exponential families

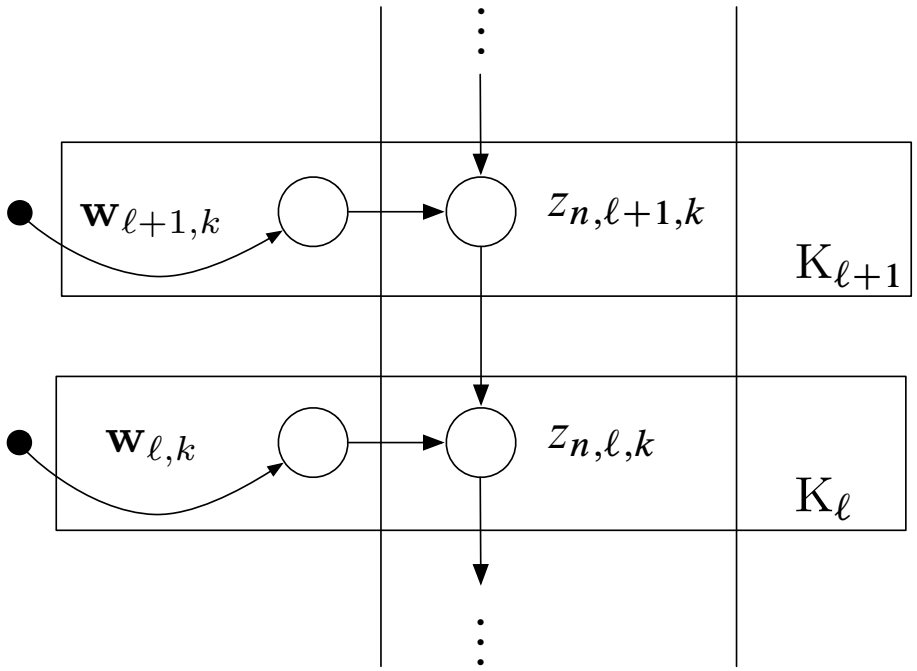


$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^{\top}\mathbf{z}_{n,\ell+1}))$$

Possibilities for the hidden layers

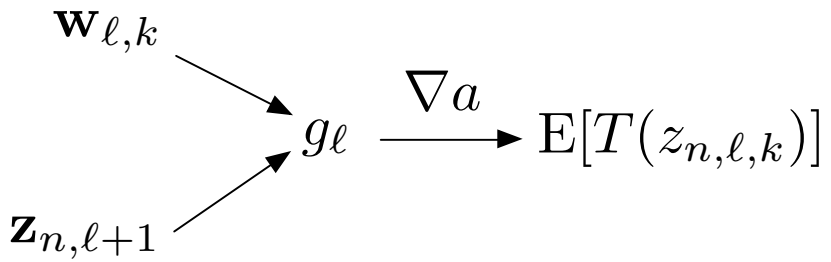
- Non-negative (and sparse) : Gamma
- Binary : Bernoulli
- Count : Poisson
- Real-valued : Gaussian

Deep exponential families



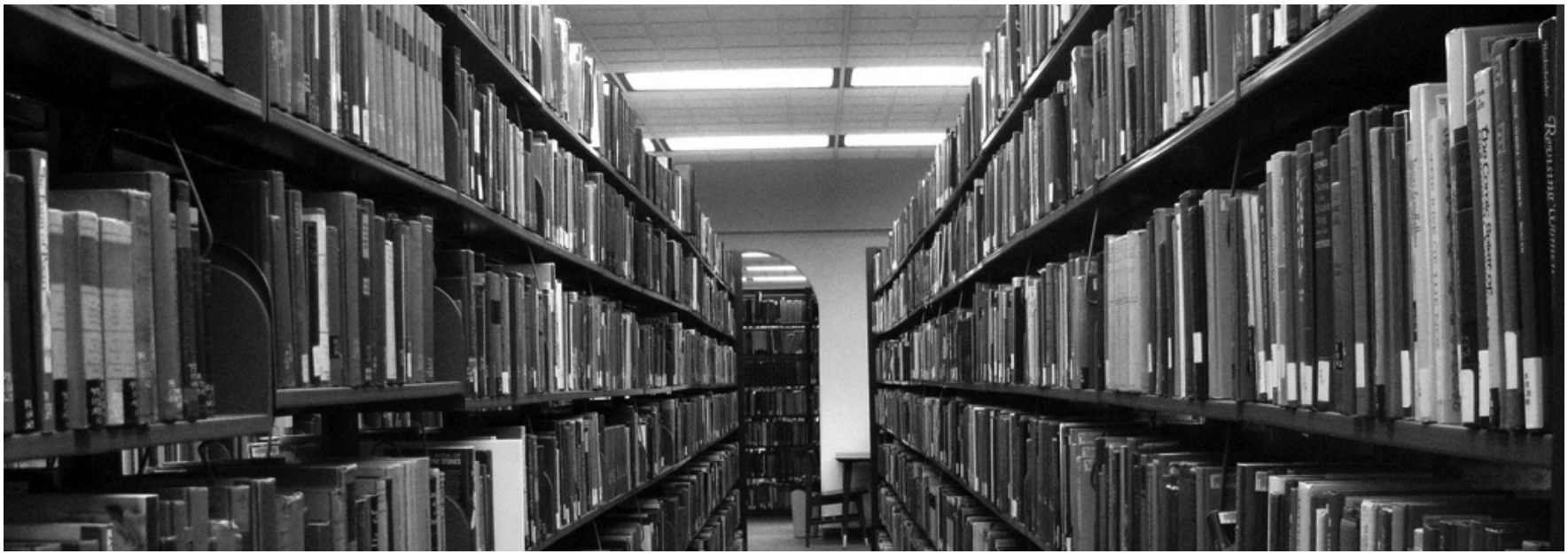
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

Nonlinearities arise naturally. There are two sources.



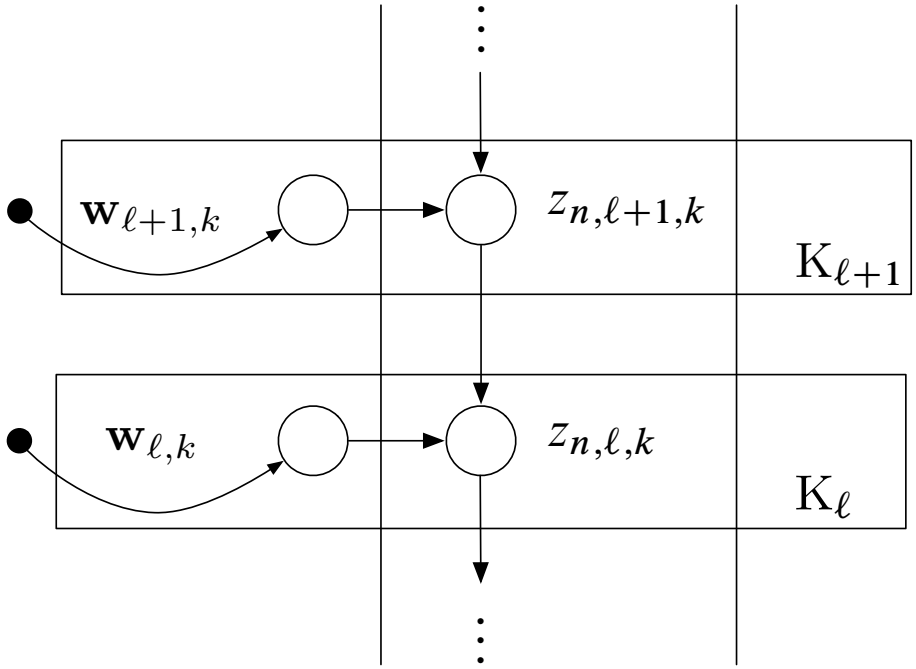
E.g., when $\mathbf{z}$ are Bernoulli then ∇a is the sigmoid function.

Example: Text data



- In discrete data, $x_{n,i}$ is a count, e.g. of word i in document n
- Use a Poisson likelihood

$$x_{n,i} \sim \text{Poisson}(g(\mathbf{w}_{0,i}^\top \mathbf{z}_{n,1}))$$



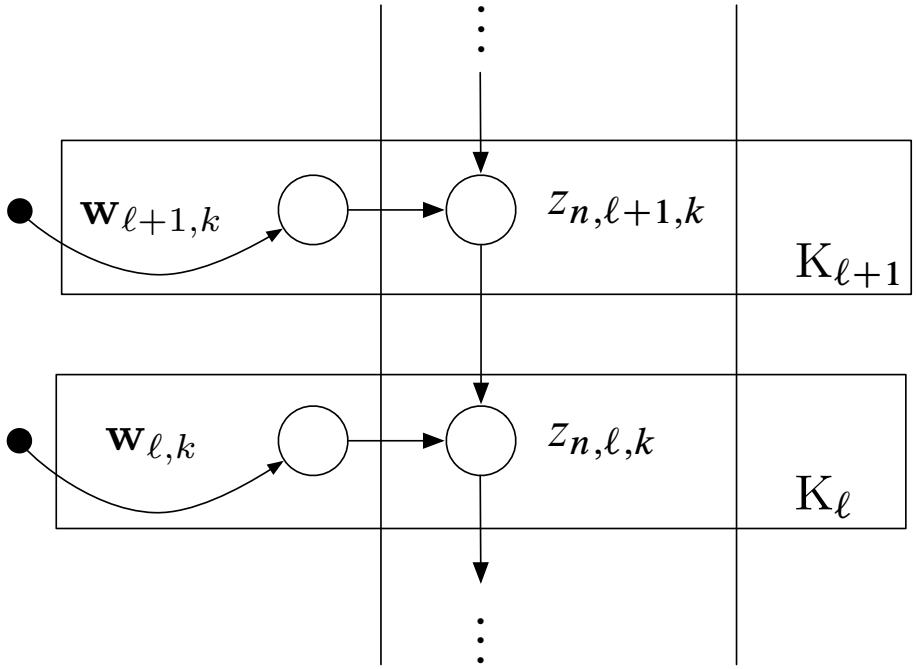
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

Bernoulli DEF, aka sigmoid belief network [Neal, 1990]

- Bernoulli layers; identity link; Gaussian prior on the weights
- Expectation comes from the logistic function

$$\mathbb{E}[z_{n,\ell,k}] = \sigma(\mathbf{z}_{n,\ell+1}^\top \mathbf{w}_{\ell+1,k})$$

- Variable $z_{n,\ell,k}$ represents whether component (ℓ, k) is "on."



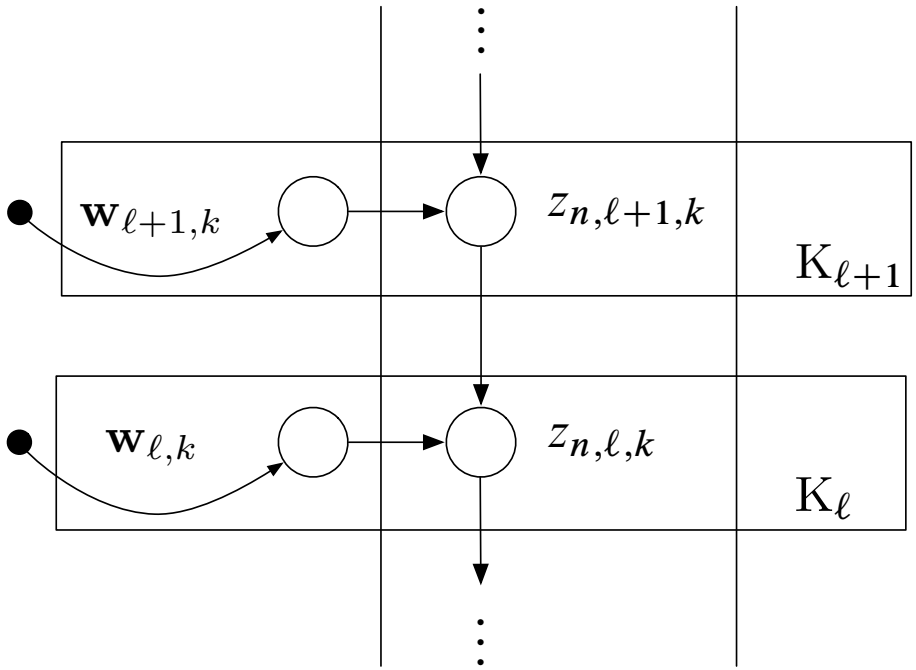
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

Poisson DEF

- Poisson layers; identity link; Gaussian prior on the weights
- Expectation is the exponentiated dot product

$$\mathbb{E}[z_{n,\ell,k}] = \exp(\mathbf{z}_{n,\ell+1}^\top \mathbf{w}_{\ell+1,k})$$

- Variable $z_{n,\ell,k}$ represents the "count" of component (ℓ, k).



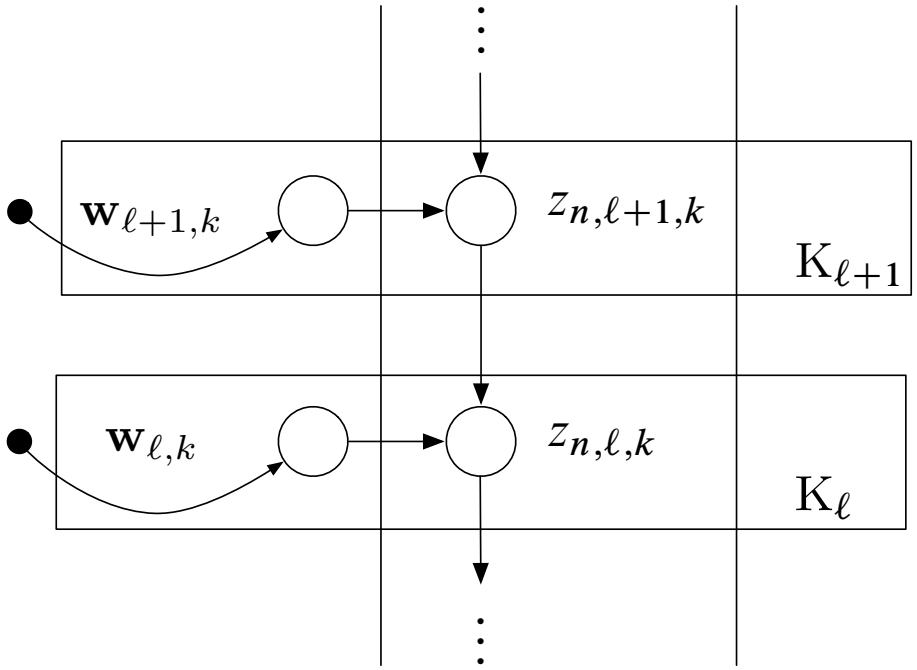
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

Poisson DEF (with log link)

- Poisson layers; log link $g(\cdot) = \log(\cdot)$.
- Expectation is dot product

$$\mathbb{E}[z_{\ell,k}] = \mathbf{z}_{n,\ell+1}^\top \mathbf{w}_{\ell+1,k}$$

- Here we place a Gamma prior on $\mathbf{w}$.



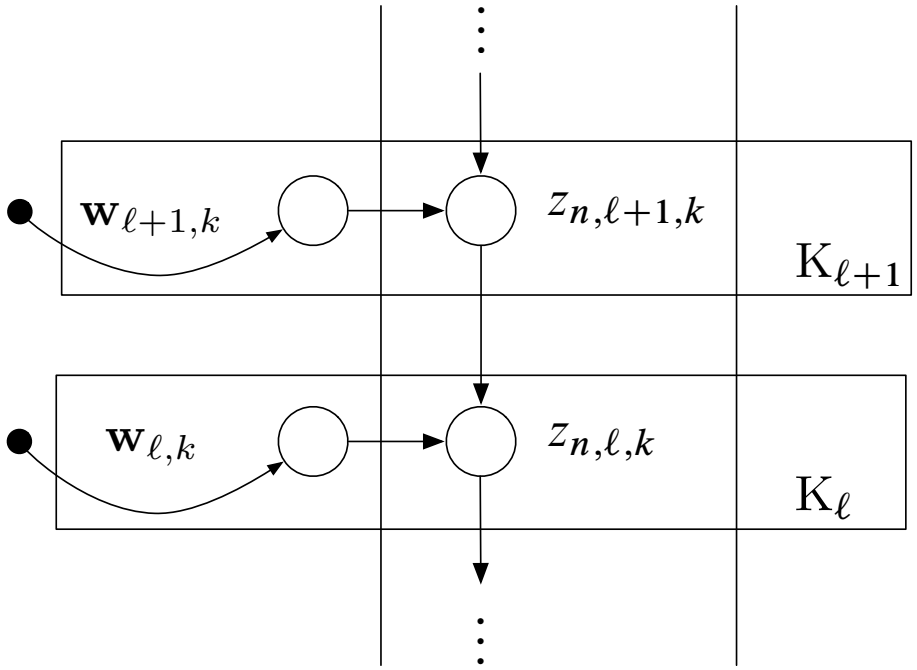
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

Sparse Gamma DEF

The Gamma is a two-parameter distribution over the positive reals

$$p(z) = z^{-1} \exp(\alpha \log(z) - \beta z - \log \Gamma(\alpha) - \alpha \log(\beta)).$$

- Sufficient statistics are z and $\log z$
- When $\alpha < 1$ the mass concentrates around zero.



$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^{\top}\mathbf{z}_{n,\ell+1}))$$

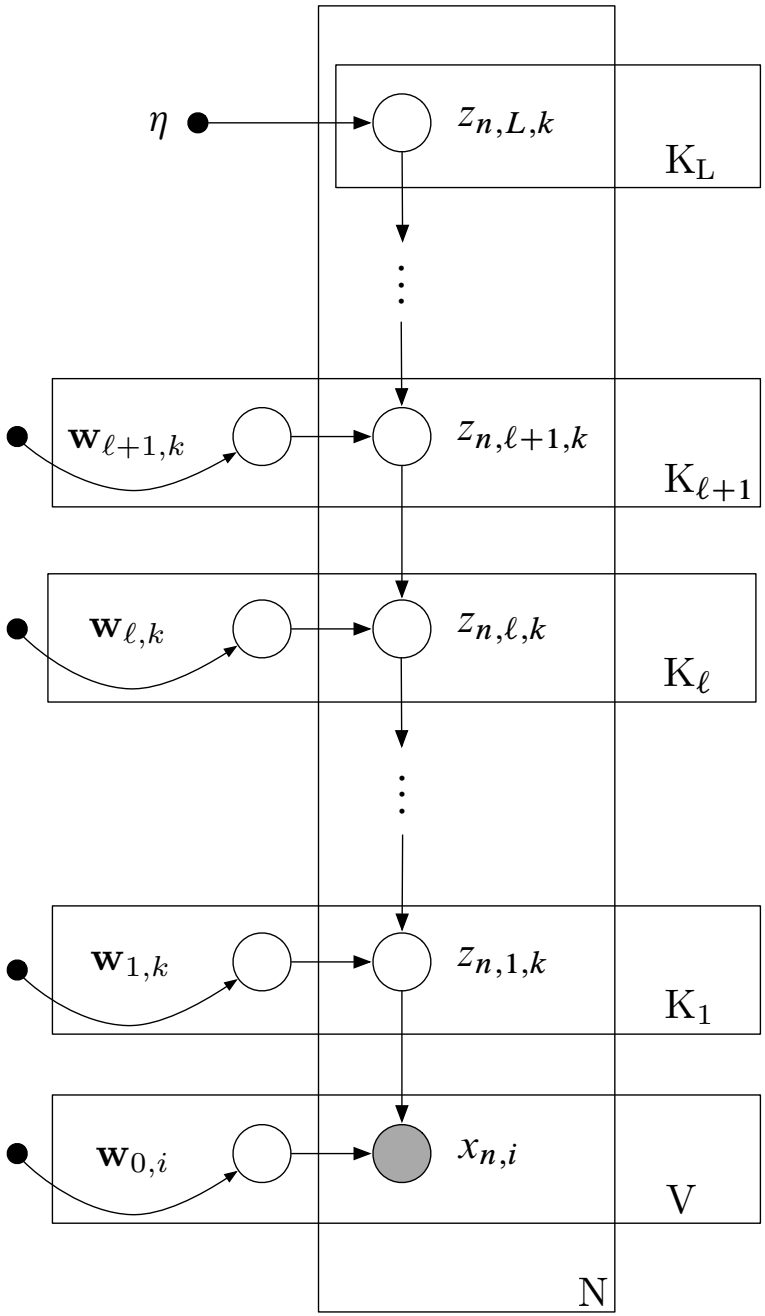
Sparse Gamma DEF

- Use sparse gamma distributions for the latent variables.

$$g_\alpha = \alpha_\ell \quad ; \quad g_\beta = \frac{\alpha_\ell}{\mathbf{z}_{n,\ell+1}^{\top}\mathbf{w}_{\ell,k}}$$

- Place a gamma prior on $\mathbf{w}_{\ell,k}$
- The expected value is $\mathbb{E}[z_{n,\ell,k}] = \mathbf{z}_{n,\ell+1}^{\top}\mathbf{w}_{\ell,k}$.

Deep exponential families

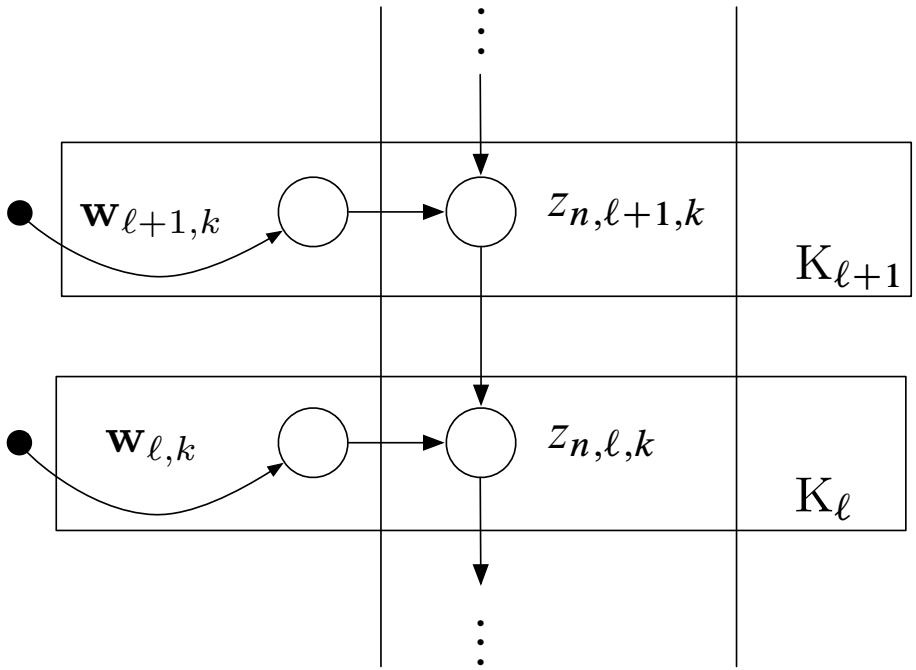


$z_{n,L,k} \sim \text{Exp-Fam}(\eta)$

$z_{n,\ell,k} \sim \text{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$

$x_{n,i} \sim \text{Exp-Fam}(g(\mathbf{w}_{0,i}^\top \mathbf{z}_{n,1}))$

Posterior inference



$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

- Goal: Try out many DEFs on a data set
  - Explore distributions, link functions, number of layers
- Solution: *black box variational inference* (BBVI)
- Let's derive BBVI in general; we will get back to DEFs later.

A recipe for variational inference

$$p(\mathbf{z}, \mathbf{x})$$

Posit a model, a joint distribution of hidden and observed variables.

A recipe for variational inference

$$q(\mathbf{z}; \boldsymbol{\nu})$$

Choose the variational family, distributions of the hidden variables.

A recipe for variational inference

$$\mathcal{L}(\boldsymbol{\nu}) = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu})]$$

Write the ELBO, the objective function for finding a $q(\mathbf{z};\boldsymbol{\nu})$ close to $p(\mathbf{z}|\mathbf{x})$.

A recipe for variational inference

$$\mathscr{L}(v) = x v^2 + \log v \quad \text{(example)}$$

Calculate the resulting expectation, i.e., take the integral.

A recipe for variational inference

$$\nabla_v \mathcal{L}(\boldsymbol{v}) = 2x\boldsymbol{v} + 1/\boldsymbol{v} \quad \text{(example)}$$

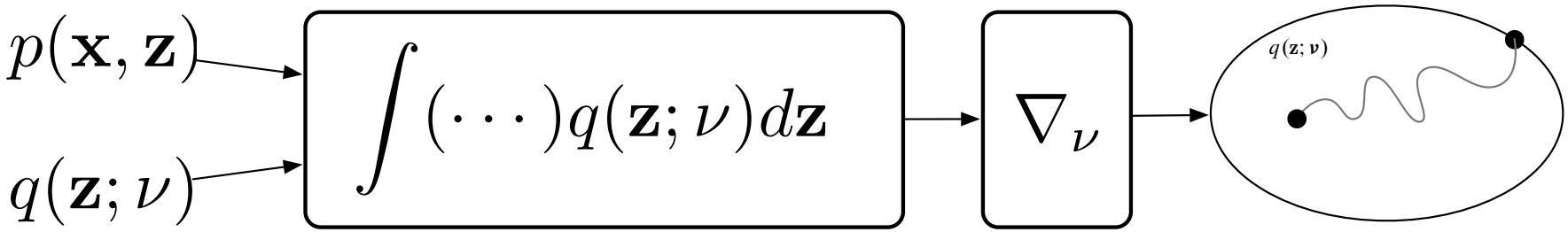
Take derivatives.

A recipe for variational inference

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \rho_t \nabla_{\boldsymbol{\nu}} \mathcal{L}$$

Optimize.

A recipe for variational inference



1. Posit a model
2. Choose a variational family
3. Integrate (calculate the ELBO)
4. Take derivatives
5. Optimize

Simplest example: Bayesian logistic regression

- Data are pairs (x_i, y_i)
  - x_i is a covariate
  - $y_i \in \{0, 1\}$ is a binary label
  - z are the regression coefficients

- Conditional on covariates, Bayesian LR posits a generative process of labels

$$z \sim N(0, 1)$$
$$y_i \,|\, x_i, z \sim \text{Bernoulli}(\sigma(z x_i)),$$

  where $\sigma(\cdot)$ is the logistic function, mapping reals to $(0, 1)$.

VI for Bayesian logistic regression

- Consider one data point (x, y).
- Our goal is to approximate the posterior coefficient $p(z \mid x, y)$.
- The variational family $q(z; \boldsymbol{\nu})$ is a normal; $\boldsymbol{\nu} = (\mu, \sigma^2)$
- The ELBO is

$$\mathcal{L}(\mu, \sigma^2) = \mathbb{E}_q[\log p(z) + \log p(y \mid x, z) - \log q(z)]$$

VI for Bayesian logistic regression

$$\mathcal{L}(\mu, \sigma^2) \quad = \quad \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)]$$

VI for Bayesian logistic regression

$$
\begin{aligned}
\mathcal{L}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C
\end{aligned}
$$

VI for Bayesian logistic regression

$$
\begin{aligned}
\mathcal{L}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \exp(xz))]
\end{aligned}
$$

VI for Bayesian logistic regression

$$
\begin{aligned}
\mathcal{L}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \exp(xz))] \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))]
\end{aligned}
$$

VI for Bayesian logistic regression

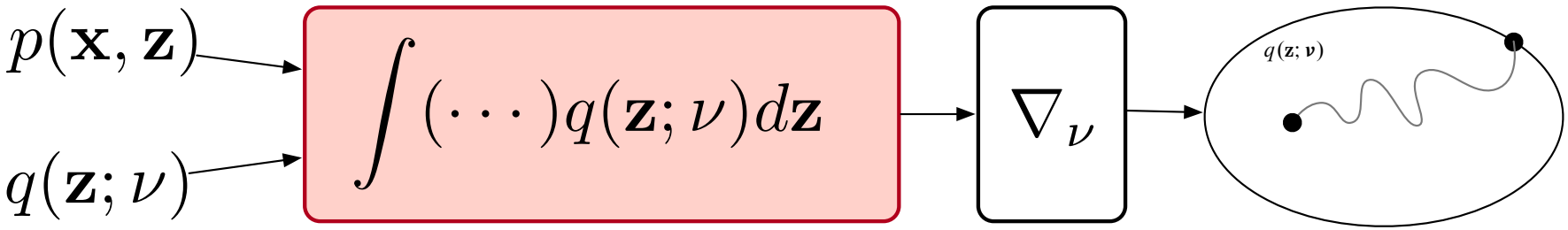
$$
\begin{aligned}
\mathcal{L}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \exp(xz))] \\
&= -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))]
\end{aligned}
$$

We are stuck—we cannot analytically take the expectation.

Options?

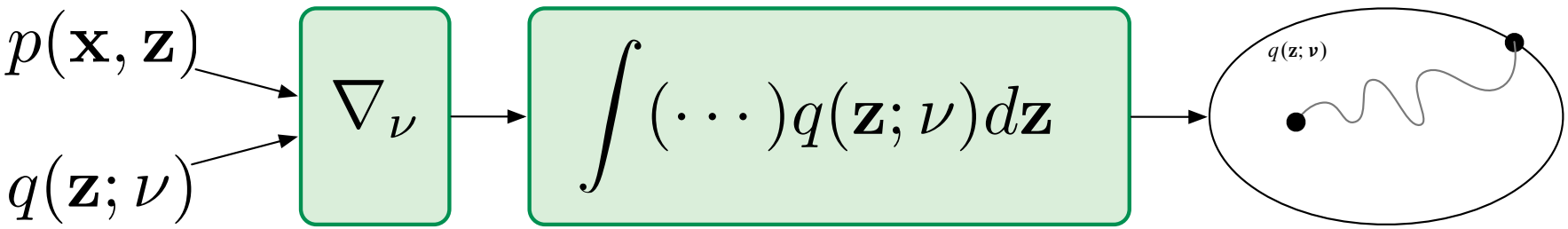
- Derive a model-specific bound
  [Jordan and Jaakola 1996], [Braun and McAuliffe 2008], others
- Use other approximations (that require model-specific analysis)
  [Wang and Blei 2013], [Knowles and Minka 2011]
- But neither satisfies our criteria for *generic inference*.

The problem with the VI recipe



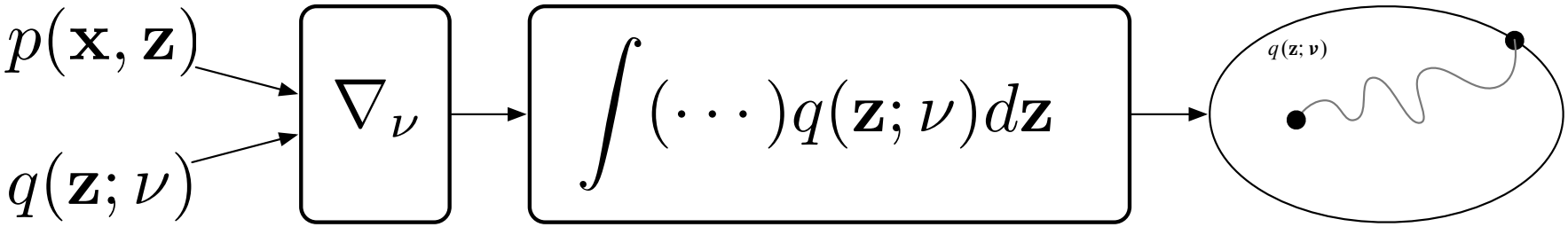
The integral is hard to take.

Solution: Swap integration and differentiation



Now we can use Monte Carlo gradients and stochastic optimization.

The new recipe



- This is the key idea behind modern methods in variational inference
- It has enabled score gradients, reparameterization gradients, amortized inference, probabilistic programming, complex variational families, and alternative divergences.
- Next: The general mathematics and some specific examples

Reversing the gradient and the expectation

- Define the "instantaneous ELBO"

$$g(\mathbf{z}, \boldsymbol{\nu}) = \log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\nu}).$$

- The ELBO is

$$\mathscr{L} = \mathbb{E}_q \left[g(\mathbf{z}, \boldsymbol{\nu}) \right] = \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z}$$

- We want to calculate $\nabla_{\boldsymbol{\nu}} \mathscr{L}$.

Reversing the gradient and the expectation

Recall the fact

$$\nabla_{\boldsymbol{\nu}} q(\mathbf{z}; \boldsymbol{\nu}) = q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu}).$$

Reversing the gradient and the expectation

Recall the fact

$$\nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) = q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}).$$

With this,

$$\nabla_{\nu} \mathcal{L} \quad = \quad \nabla_{\nu} \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z}$$

Reversing the gradient and the expectation

Recall the fact

$$\nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) = q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}).$$

With this,

$$\begin{aligned} \nabla_{\nu} \mathcal{L} &= \nabla_{\nu} \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\ &= \int \nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \end{aligned}$$

Reversing the gradient and the expectation

Recall the fact

$$\nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) = q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}).$$

With this,

$$\begin{aligned}
\nabla_{\nu} \mathcal{L} &= \nabla_{\nu} \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int \nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z}
\end{aligned}$$

Reversing the gradient and the expectation

Recall the fact

$$\nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) = q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}).$$

With this,

$$\begin{aligned}
\nabla_{\nu} \mathcal{L} &= \nabla_{\nu} \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int \nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z}; \boldsymbol{\nu})} [\nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu})]
\end{aligned}$$

Reversing the gradient and the expectation

Recall the fact

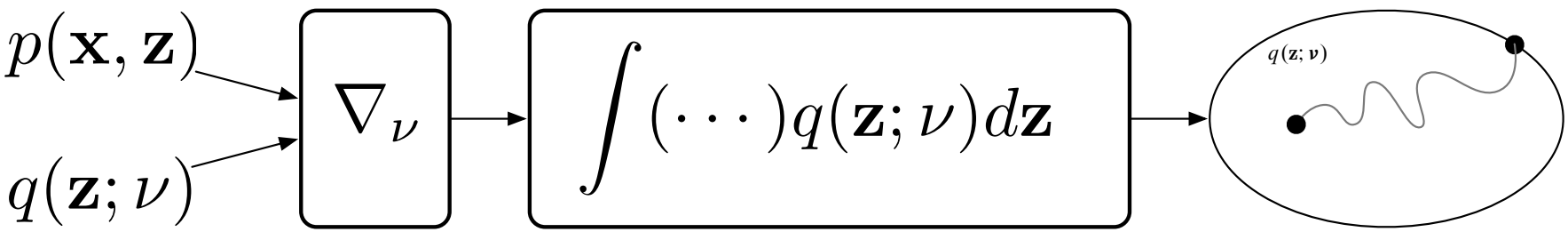
$$\nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) = q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}).$$

With this,

$$\begin{aligned}
\nabla_{\nu} \mathcal{L} &= \nabla_{\nu} \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int \nabla_{\nu} q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z}; \boldsymbol{\nu})} [\nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + \nabla_{\nu} g(\mathbf{z}, \boldsymbol{\nu})]
\end{aligned}$$

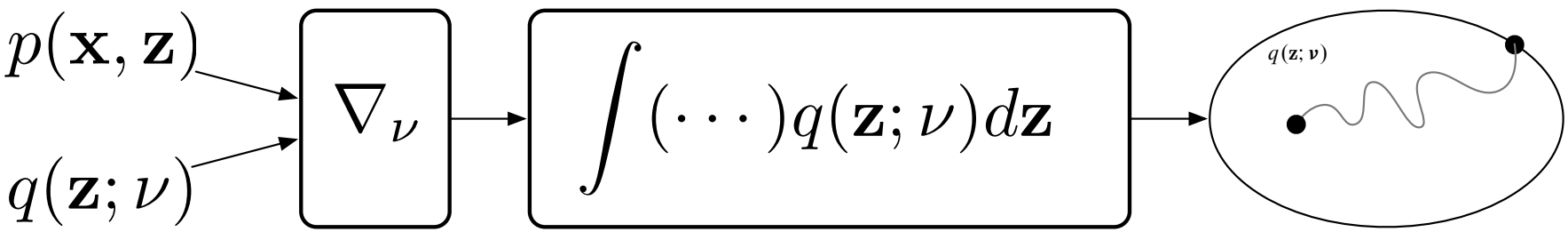
We have written the gradient as an expectation.

Roadmap



- Score function estimator and basic black box variational inference
- Reparameterization gradient
- Autodifferentiation VI and probabilistic programming

Roadmap



- **Score function estimator and basic black box variational inference**
- Reparameterization gradient
- Autodifferentiation VI and probabilistic programming

The score function and black box variational inference

- Recall

$$\nabla_{\nu}\mathcal{L} = \mathbb{E}_{q(\mathbf{z};\nu)}[\nabla_{\nu}\log q(\mathbf{z};\nu)g(\mathbf{z},\nu) + \nabla_{\nu}g(z,\nu)]$$

  Simplify the second term

$$\mathbb{E}_q[\nabla_{\nu}g(\mathbf{z},\nu)] = \mathbb{E}_q[\nabla_{\nu}\log q(\mathbf{z};\nu)] = 0$$

- This gives the score gradient

$$\nabla_{\nu}\mathcal{L} = \mathbb{E}_{q(\mathbf{z};\nu)}[\nabla_{\nu}\log q(\mathbf{z};\nu)(\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\nu))]$$

- Sometimes called the likelihood ratio or REINFORCE gradient
  [Glynn 1990; Williams, 1992; Wingate+ 2013; Ranganath+ 2014; Mnih+ 2014]

Noisy unbiased gradients

- We construct noisy unbiased gradients with Monte Carlo,

$$\hat{\nabla}_{\boldsymbol{\nu}} = \frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}_s; \boldsymbol{\nu})(\log p(\mathbf{x}, \mathbf{z}_s) - \log q(\mathbf{z}_s; \boldsymbol{\nu})),$$

$$\text{where } \mathbf{z}_s \sim q(\mathbf{z}; \boldsymbol{\nu})$$

- To compute a noisy gradient of the ELBO, we need to
  - sample from $q(\mathbf{z})$
  - evaluate $\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu})$
  - evaluate $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$

- Satisfies the "black box criteria" — no model-specific analysis needed.

Algorithm 1: Basic Black Box Variational Inference

Input: data $\mathbf{x}$, model $p(\mathbf{z}, \mathbf{x})$.

Initialize $\boldsymbol{\nu}$ randomly.
Set ρ_t appropriately.

while *not converged* **do**

Take S samples from the variational distribution

$$\mathbf{z}[s] \sim q(\mathbf{z}; \boldsymbol{\nu}) \quad s = 1 \ldots S$$

Update the variational parameters

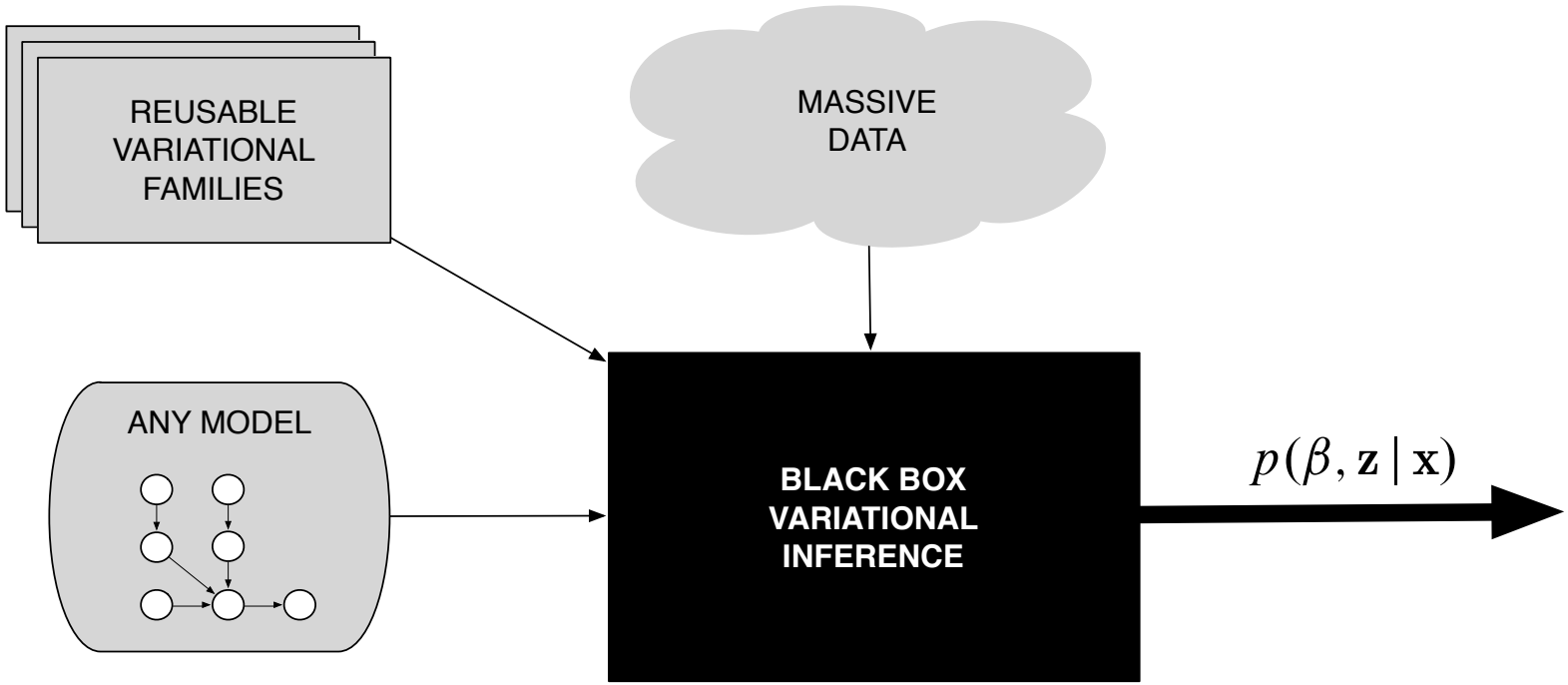
$$\boldsymbol{\nu} = \boldsymbol{\nu} + \rho_t \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}[s]; \boldsymbol{\nu}) (\log p(\mathbf{x}, \mathbf{z}[s]) - \log q(\mathbf{z}[s]; \boldsymbol{\nu}))$$

end

Black box criteria

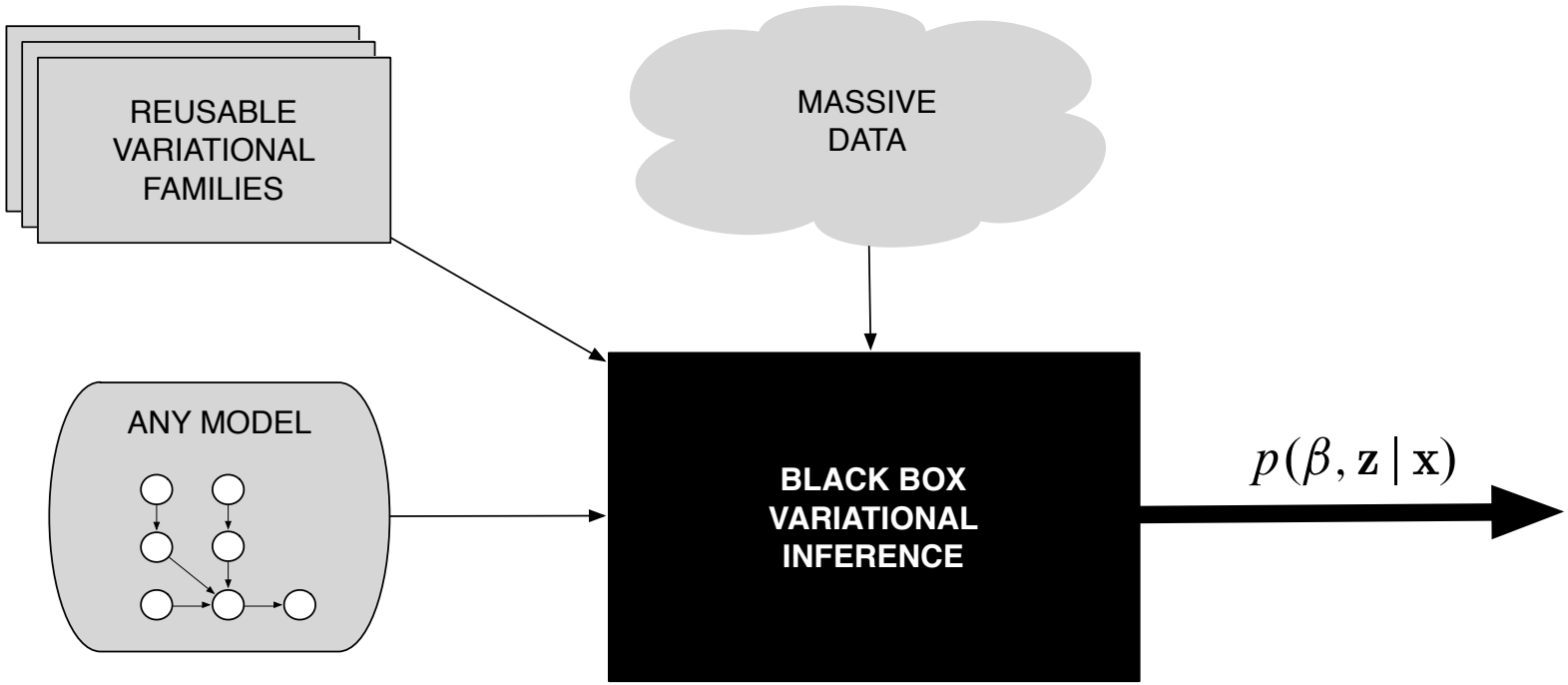
Variable z	Log density $\log q(z;\lambda)$	Score $\nabla_\lambda \log q(z)$
Poisson(λ)	$-\lambda + z\log\lambda - \log z!$	$-1 + \frac{z}{\lambda}$
Bernoulli($\sigma(\lambda)$)	$z\lambda - \log(1+\exp(\lambda))$	$z - \sigma(\lambda)$
Gamma($\lambda, 1$)	$-\log\Gamma(\lambda) + (\lambda-1)\log z - z$	$-\Psi(\lambda) + \log(z)$
Exponential($\frac{1}{\lambda}$)	$-\log\lambda - \frac{z}{\lambda}$	$-\frac{1.0}{\lambda} + \frac{z}{\lambda^2}$
Normal($\lambda, 1$)	$-\frac{1}{2}\log(2\pi) - \frac{1}{2}(z-\lambda)^2$	$z - \lambda$

Black box variational inference



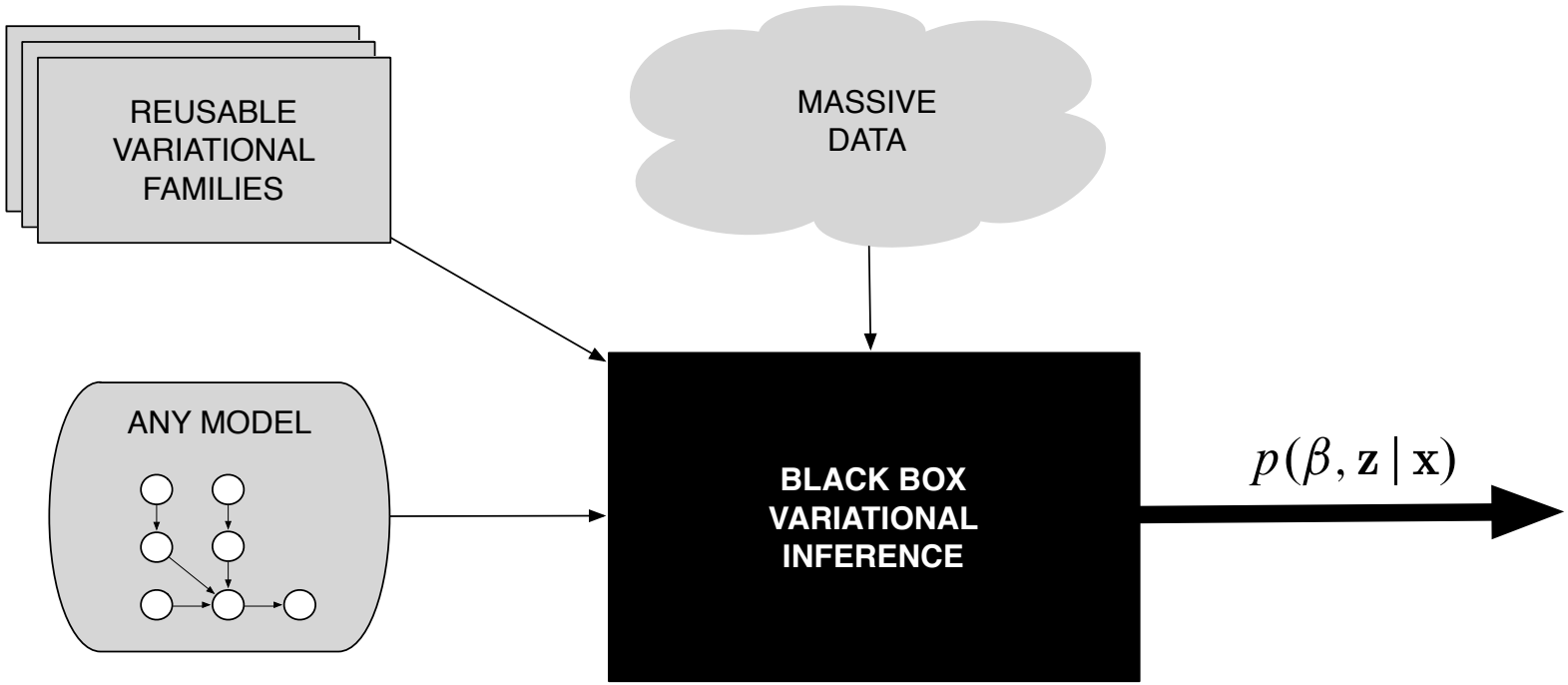
- Easily use variational inference with *any model*
- No exponential family requirements
- No mathematical work beyond specifying the model

Black box variational inference



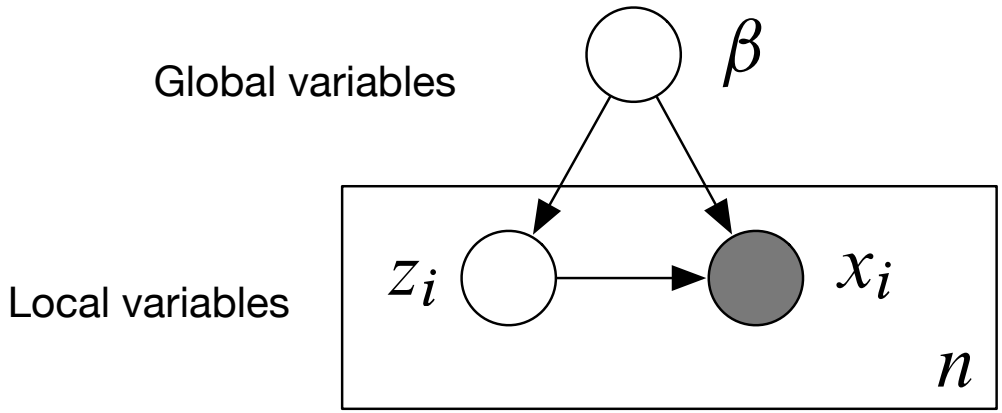
- Sample from $q(\cdot)$ (or a related distribution)
- Form noisy gradients without model-specific computation
- Use stochastic optimization

Black box variational inference



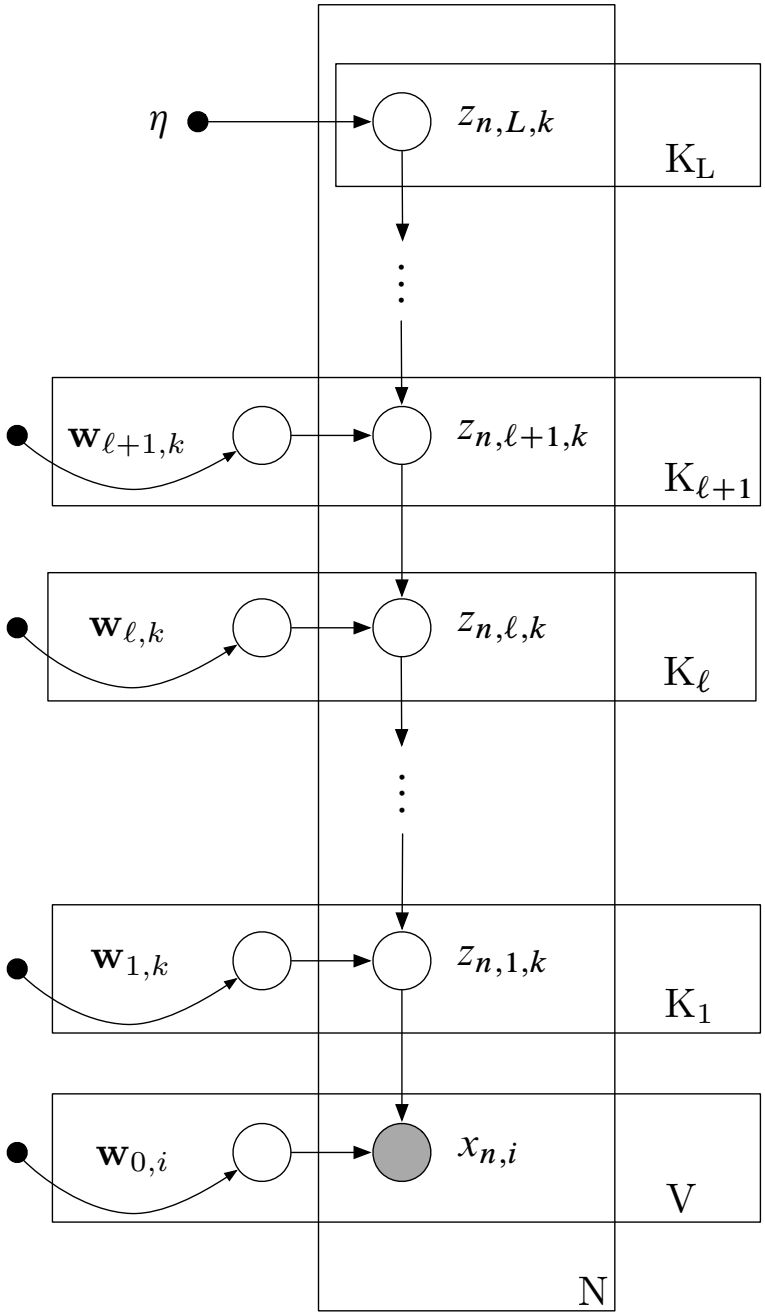
- We must control the variance of the gradient
  - Rao-Blackwellization, control variates, importance sampling, ...
- Adaptive learning rates [Duchi+ 2011; Tieleman and Hinton 2012]
- Stochastic variational inference, for handling massive data

Nonconjugate models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Nonlinear time series models
- Deep latent Gaussian models
- Models with attention
- Generalized linear models
- Stochastic volatility models
- Discrete choice models
- Bayesian neural networks
- Deep exponential families
- Correlated topic models
- Sigmoid belief networks



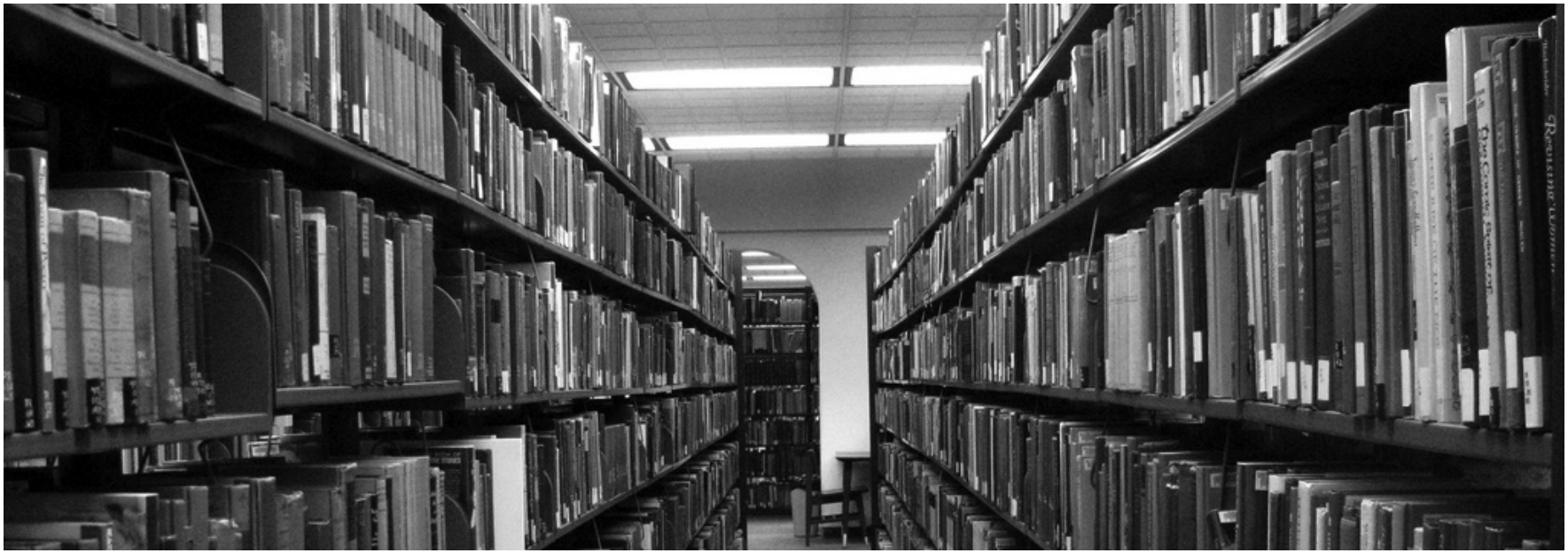
$z_{n,L,k} \sim \textsc{Exp-Fam}(\eta)$

$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$

$x_{n,i} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{0,i}^\top \mathbf{z}_{n,1}))$

[Ranganath+ 2015]

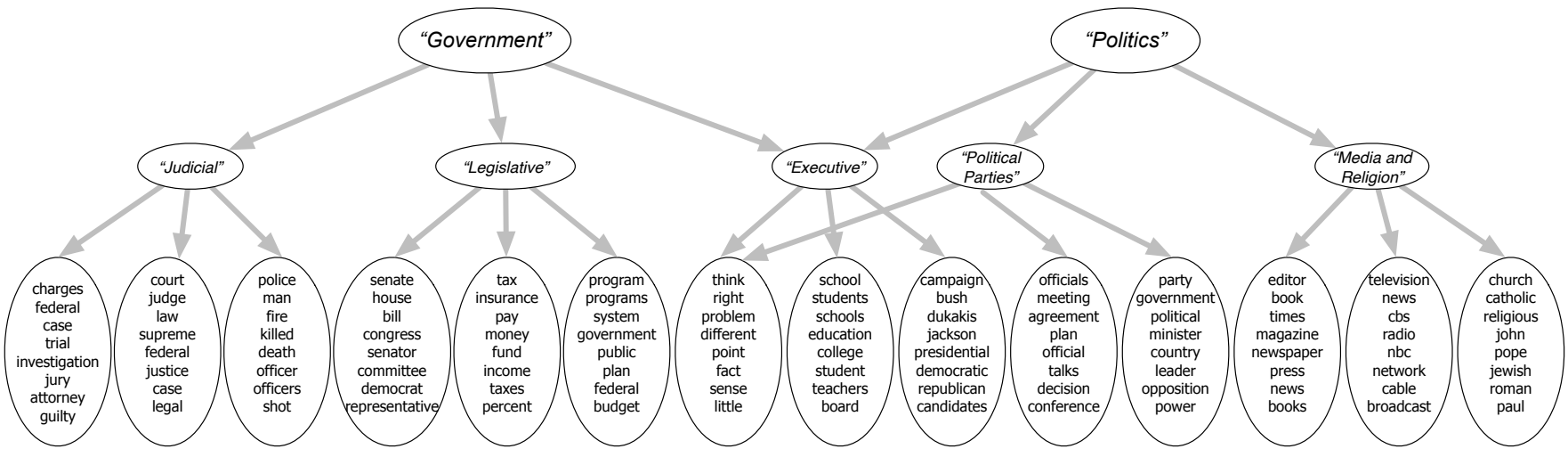
Empirical study

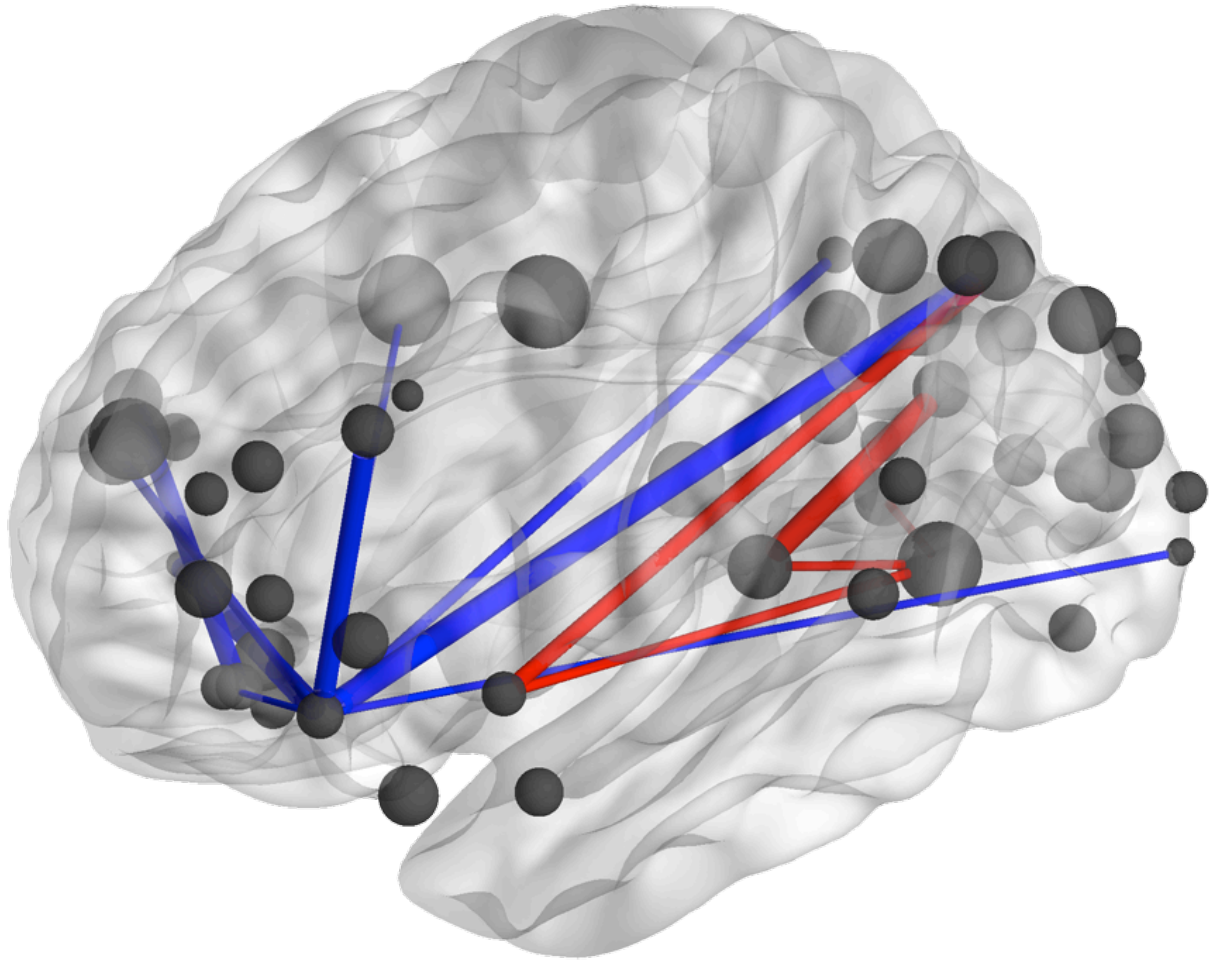


- NYT and Science (about 150K documents in each, about 7K terms)
- Evaluation: Document prediction perplexity (lower is better)
  [Wallach et al., 2009]
- Used BBVI for all runs, varied depth, distributions, and link functions.

DEF evaluation

Model	$p(\mathbf{w})$	*NYT*	*Science*
LDA [Blei+ 2003]		2717	1711
DocNADE [Larochelle+ 2012]		2496	1725
Sparse Gamma 100	$\emptyset$	2525	1652
Sparse Gamma 100-30	Γ	2303	1539
Sparse Gamma 100-30-15	Γ	**2251**	1542
Sigmoid 100	$\emptyset$	2343	1633
Sigmoid 100-30	$\mathcal{N}$	2653	1665
Sigmoid 100-30-15	$\mathcal{N}$	2507	1653
Poisson 100	$\emptyset$	2590	1620
Poisson 100-30	$\mathcal{N}$	2423	1560
Poisson 100-30-15	$\mathcal{N}$	2416	1576
Poisson log-link 100-30	Γ	2288	**1523**
Poisson log-link 100-30-15	Γ	2366	1545


"Government"
"Politics"
"Judicial"
"Legislative"
"Executive"
"Political Parties"
"Media and Religion"
charges federal case trial investigation jury attorney guilty
court judge law supreme federal justice case legal
police man fire killed death officer officers shot
senate house bill congress senator committee democrat representative
tax insurance pay money fund income taxes percent
program programs system government public plan federal budget
think right problem different point fact sense little
school students schools education college student teachers board
campaign bush dukakis jackson presidential democratic republican candidates
officials meeting agreement plan official talks decision conference
party government political minister country leader opposition power
editor book times magazine newspaper press news books
television news cbs radio nbc network cable broadcast
church catholic religious john pope jewish roman paul



Neuroscience analysis of 220 million fMRI measurements

[Manning+ 2014]

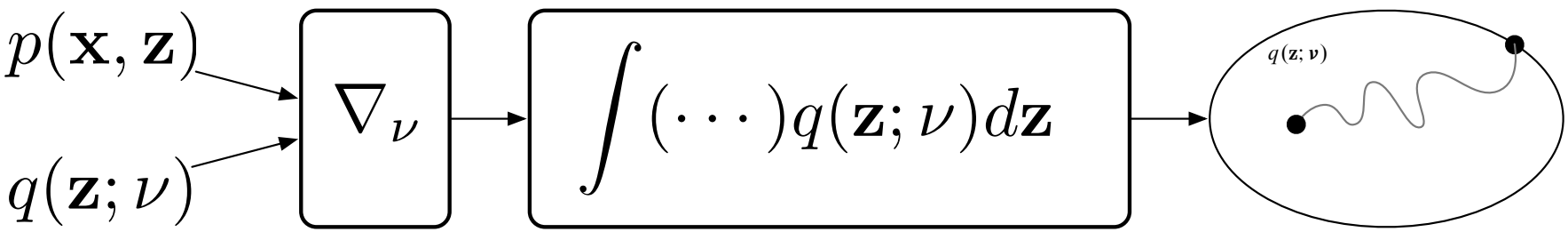
More assumptions?

The black box criteria are to

- sample from $q(\mathbf{z})$
- evaluate $\nabla_{\nu} \log q(\mathbf{z}; \boldsymbol{\nu})$
- evaluate $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$

Can we make additional assumptions that are not too restrictive?

Roadmap



- Score function estimator and basic black box variational inference
- **Reparameterization gradient**
- Autodifferentiation VI and probabilistic programming

The reparameterization gradient

- Assume that we can express the variational distribution with a transformation, where

$$\begin{aligned}\boldsymbol{\epsilon} &\sim s(\boldsymbol{\epsilon}) \\ \mathbf{z} &= t(\boldsymbol{\epsilon}, \boldsymbol{\nu}) \\ &\rightarrow \mathbf{z} \sim q(\mathbf{z}; \boldsymbol{\nu})\end{aligned}$$

- For example,

$$\begin{aligned}\epsilon &\sim \text{Normal}(0, 1) \\ z &= \epsilon\sigma + \mu \\ &\rightarrow z \sim \text{Normal}(\mu, \sigma^2)\end{aligned}$$

- Also assume $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$ are differentiable with respect to $\mathbf{z}$

The reparameterization gradient

- Recall

$$\nabla_{\boldsymbol{\nu}} \mathscr{L} = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu}) g(\mathbf{z},\boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} g(\mathbf{z},\boldsymbol{\nu})]$$

The reparameterization gradient

- Recall

$$\nabla_{\nu}\mathcal{L} = \mathbb{E}_{q(\mathbf{z};\nu)}[\nabla_{\nu}\log q(\mathbf{z};\nu)g(\mathbf{z},\nu) + \nabla_{\nu}g(\mathbf{z},\nu)]$$

- Rewrite using using $\mathbf{z} = t(\epsilon, \nu)$,

$$\nabla_{\nu}\mathcal{L} = \mathbb{E}_{s(\epsilon)}[\nabla_{\nu}\log s(\epsilon)g(t(\epsilon,\nu),\nu) + \nabla_{\nu}g(t(\epsilon,\nu),\nu)]$$

The reparameterization gradient

- Recall

$$\nabla_{\nu}\mathcal{L} = \mathbb{E}_{q(\mathbf{z};\nu)}[\nabla_{\nu}\log q(\mathbf{z};\nu)g(\mathbf{z},\nu) + \nabla_{\nu}g(\mathbf{z},\nu)]$$

- Rewrite using using $\mathbf{z} = t(\epsilon, \nu)$,

$$\nabla_{\nu}\mathcal{L} = \mathbb{E}_{s(\epsilon)}[\nabla_{\nu}\log s(\epsilon)g(t(\epsilon,\nu),\nu) + \nabla_{\nu}g(t(\epsilon,\nu),\nu)]$$

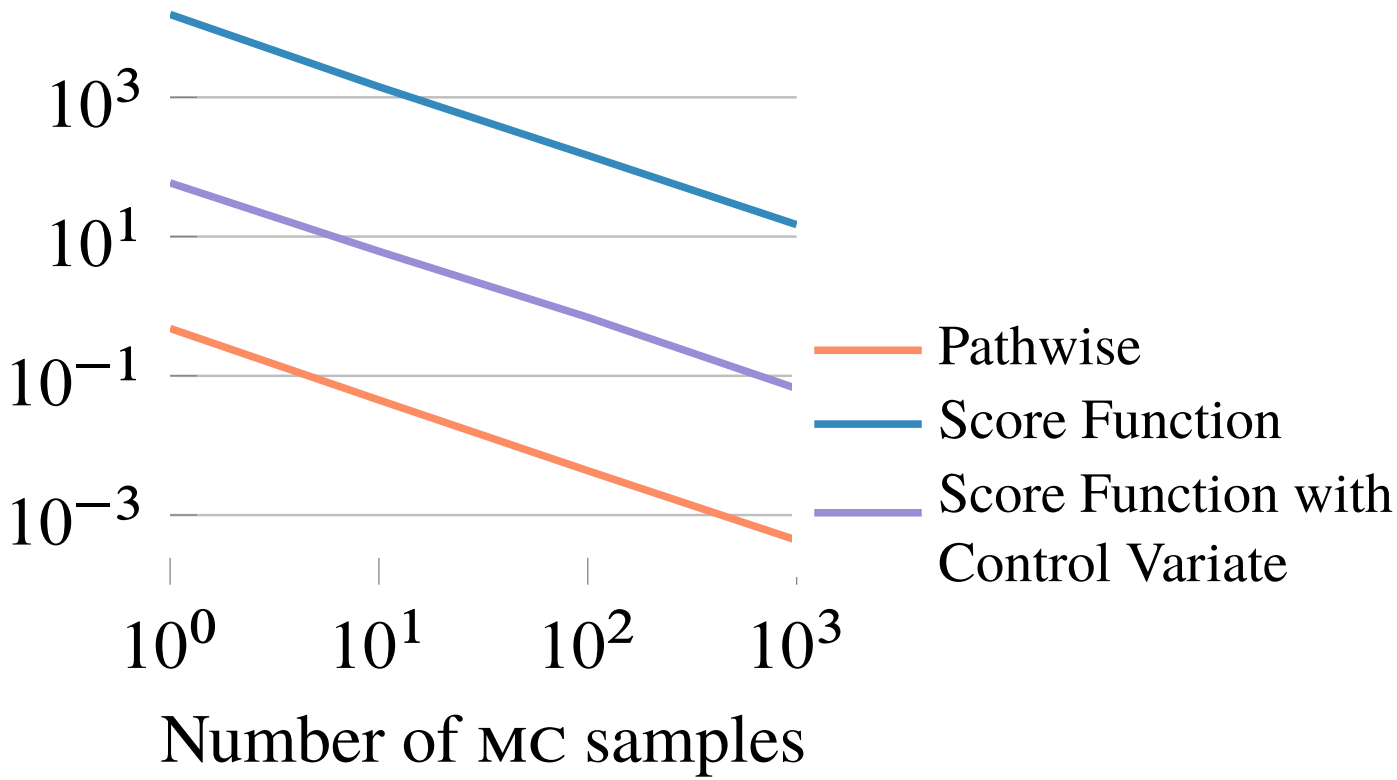
- Note that $\nabla_{\nu}\log s(\epsilon) = 0$. Thus,

$$\begin{aligned}\nabla\mathcal{L} &= \mathbb{E}_{s(\epsilon)}[\nabla_{\nu}g(t(\epsilon,\nu),\nu)] \\ &= \mathbb{E}_{s(\epsilon)}[\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\nu)]\nabla_{\nu}t(\epsilon,\nu) - \nabla_{\nu}\log q(\mathbf{z};\nu)] \\ &= \mathbb{E}_{s(\epsilon)}[\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\nu)]\nabla_{\nu}t(\epsilon,\nu)]\end{aligned}$$

This is also known as the reparameterization gradient.

[Glasserman 1991; Fu 2006; Kingma+ 2014; Rezende+ 2014; Titsias+ 2014]

Variance Comparison



[Kucukelbir+ 2016]

Score vs. reparameterization gradients

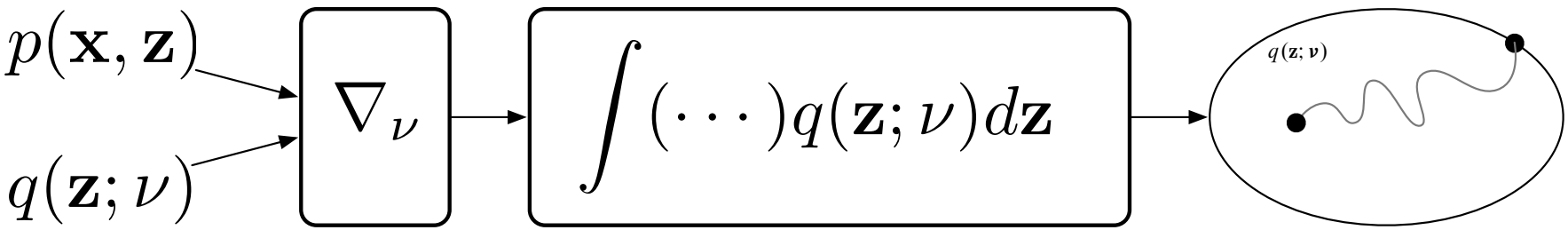
Score: $\mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu})(\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu}))]$

- Differentiates the variational density
- Works for discrete and continuous models
- Works for large class of variational approximations
- Variance can be a problem

Reparameterization: $\mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu})]\nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon},\boldsymbol{\nu})]$

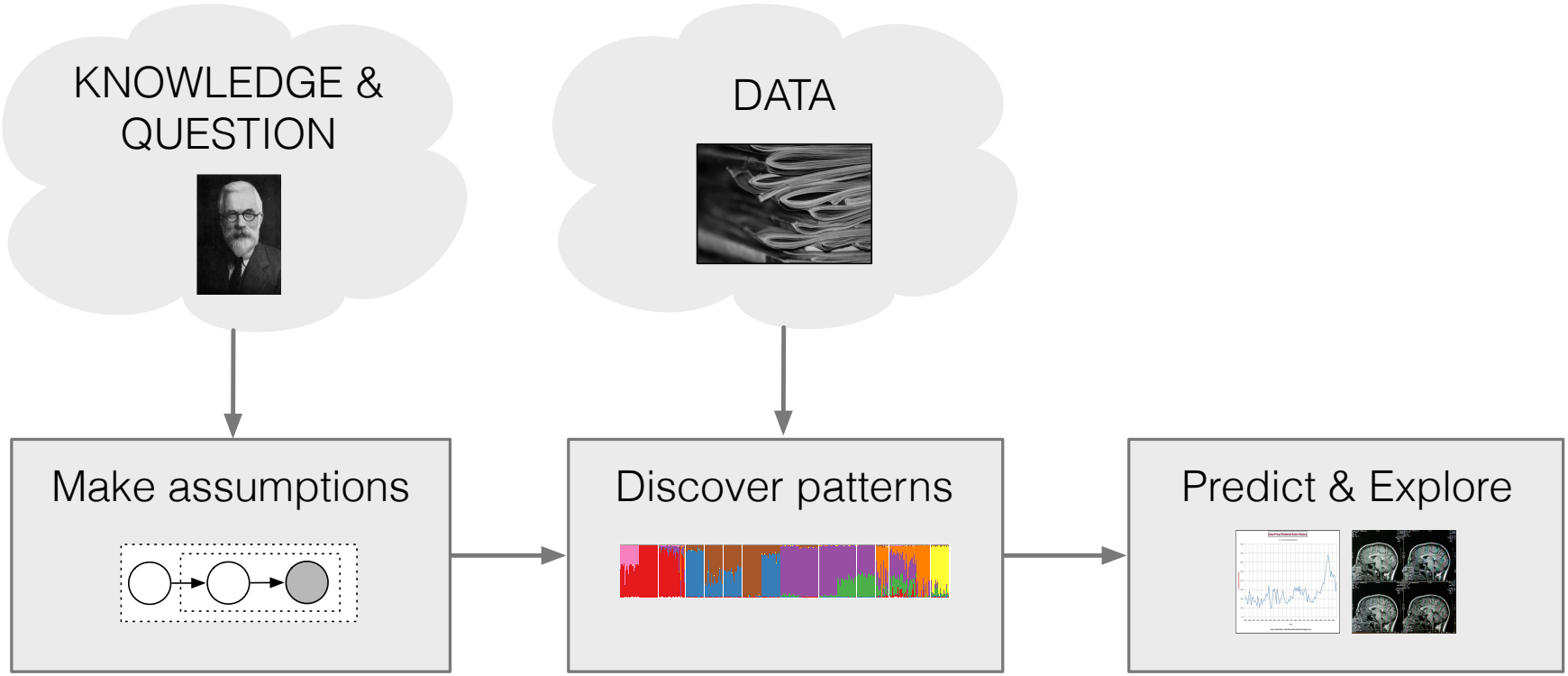
- Differentiates the instantaneous ELBO
- Requires differentiable models
- Requires variational approximation to have form $\mathbf{z} = t(\boldsymbol{\epsilon},\boldsymbol{\nu})$
- Better behaved variance

Roadmap



- Score function estimator and basic black box variational inference
- Reparameterization gradient
- **Autodifferentiation VI and probabilistic programming**

Probabilistic programming



- *Probabilistic programming languages* automate inference.
- Write a model down as a program; "compile" it to an inference procedure.
- We deployed reparameterization gradients in Stan;
  10,000 modelers can use variational inference. [Kucukelbir+ 2016]

Supervised pPCA with ARD

```
data {
  int<lower=0> N;              // number of data points in dataset
  int<lower=0> D;              // dimension
  int<lower=0> M;              // maximum dimension of latent space to cinsider

  vector[D] x[N];
  vector[N] y;
}

parameters {
  matrix[M,N] z;               // latent variable
  matrix[D,M] w_x;             // weights parameters
  vector[M]   w_y;             // variance parameter
  real<lower=0> sigma;
  vector<lower=0>[M] alpha;    // hyper-parameters on weights
}

model {
  // priors
  to_vector(z) ~ normal(0,1);
  for (d in 1:D)
    w_x[d] ~ normal(0, sigma * alpha);
  w_y ~ normal(0, sigma * alpha);

  sigma ~ lognormal(0,1);
  alpha ~ inv_gamma(1,1);

  // likelihood
  for (n in 1:N) {
    x[n] ~ normal(w_x * col(z, n), sigma);
    y[n] ~ normal(w_y' * col(z, n), sigma);
  }
}
```


Maia
Guelfães
Ermesinde
Santa Cruz do Bispo
Leça do Balio
Aguas Santas
Guifões
São Mamede de Infesta
Matosinhos
Rio Tinto
Gondomar
Barrosos
Alumiara
Lavadores
Vila Nova de Gaia
Valbom
Canidelo
Madalena
Avintes

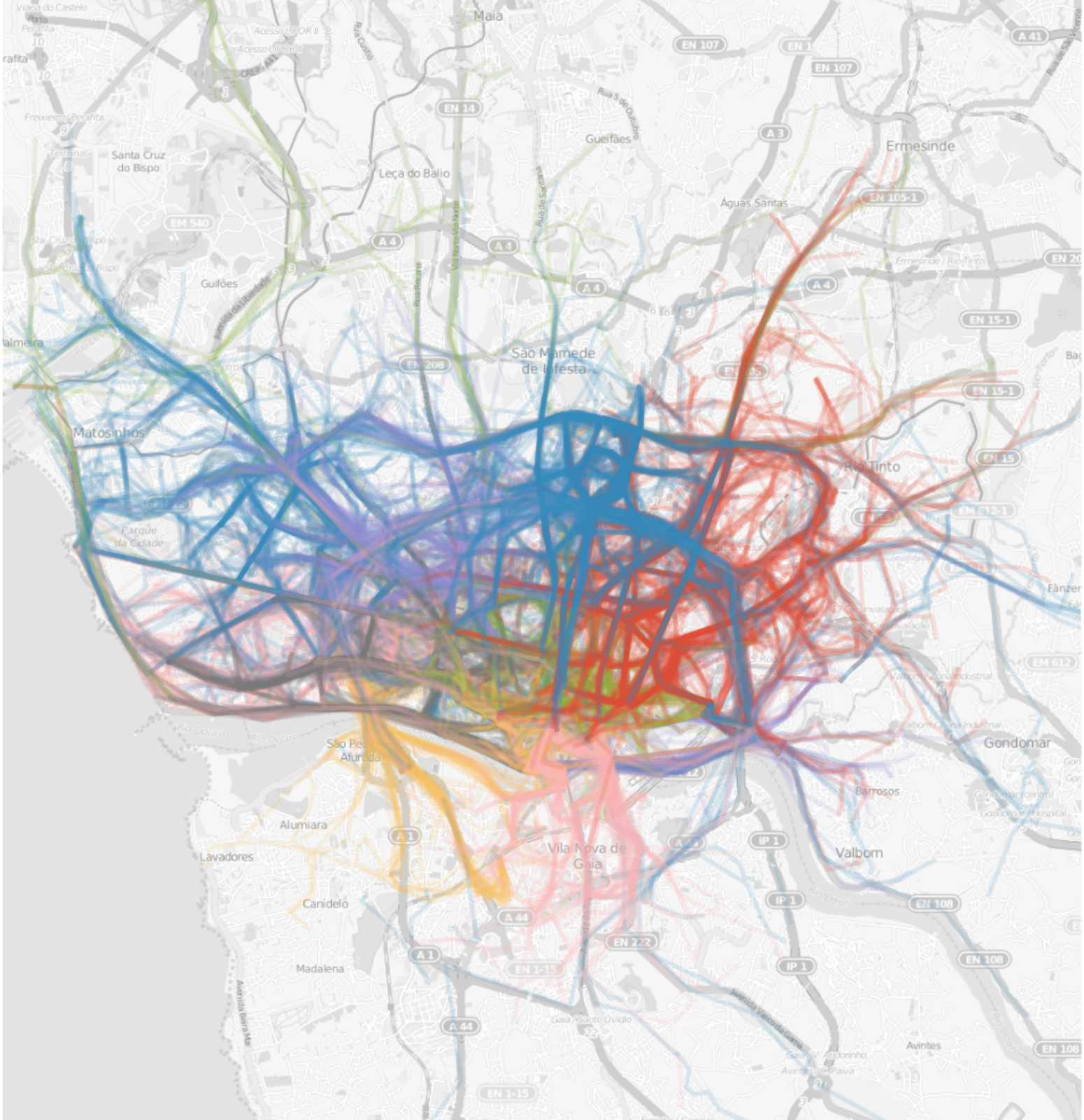
Exploring Taxi Trajectories

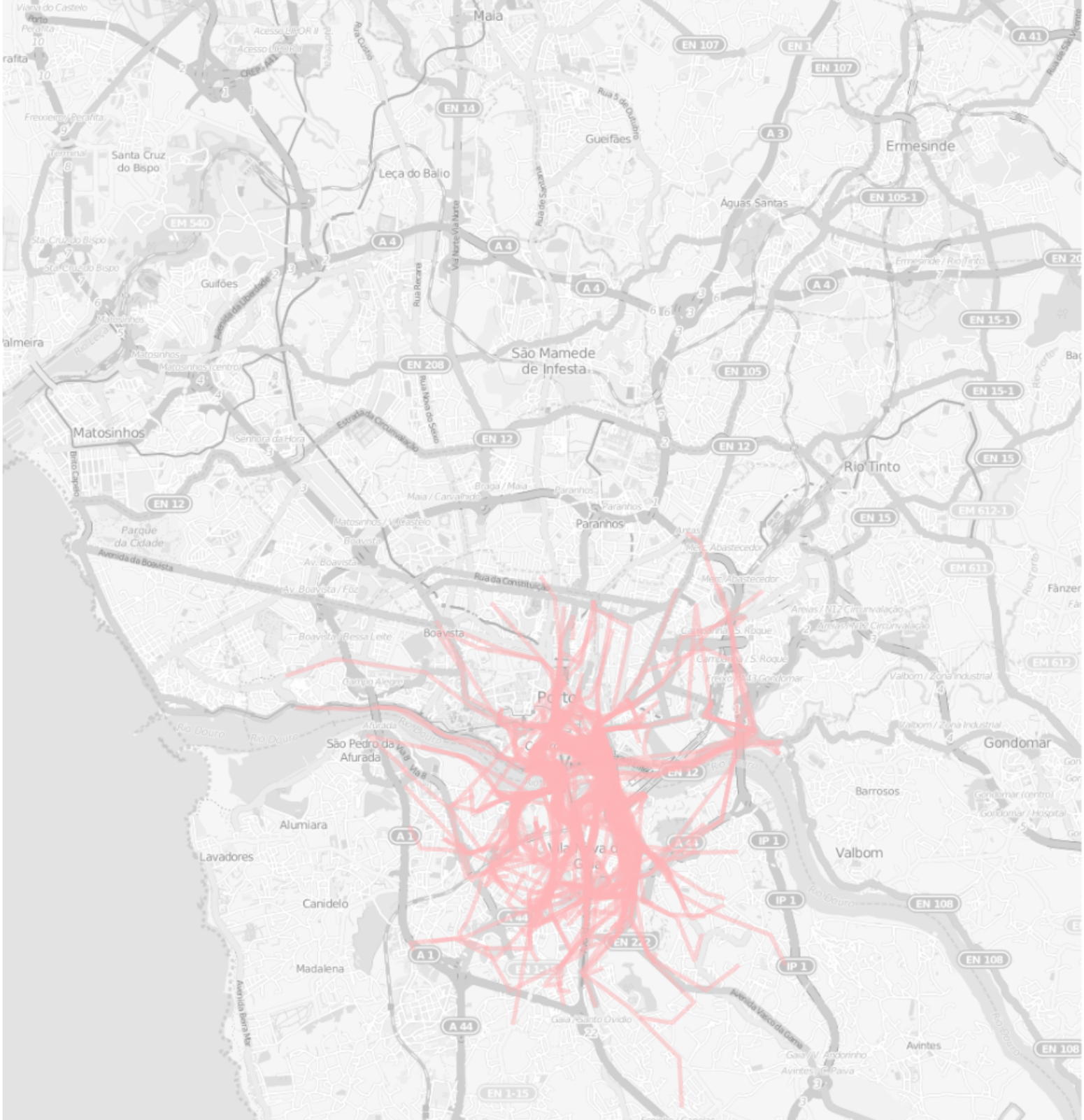
Data: 1.7 million taxi rides from Porto, Portugal

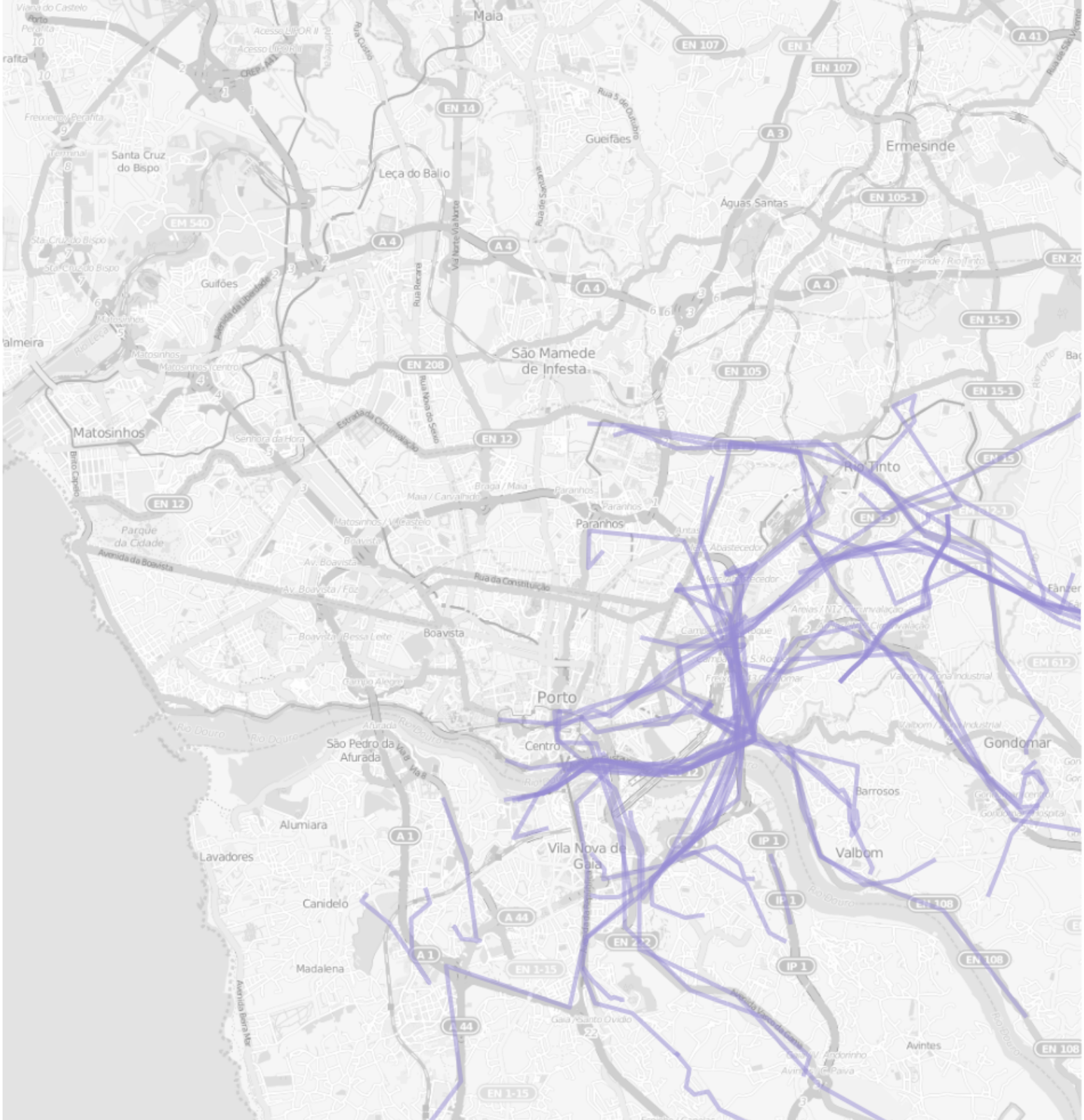
- Write down a supervised pPCA model (∼minutes).
- Use ADVI to fit model (∼hours).
- Project data into pPCA subspace (∼minutes).

- Write down a mixture model (∼minutes).
- Use ADVI to find patterns (∼minutes).

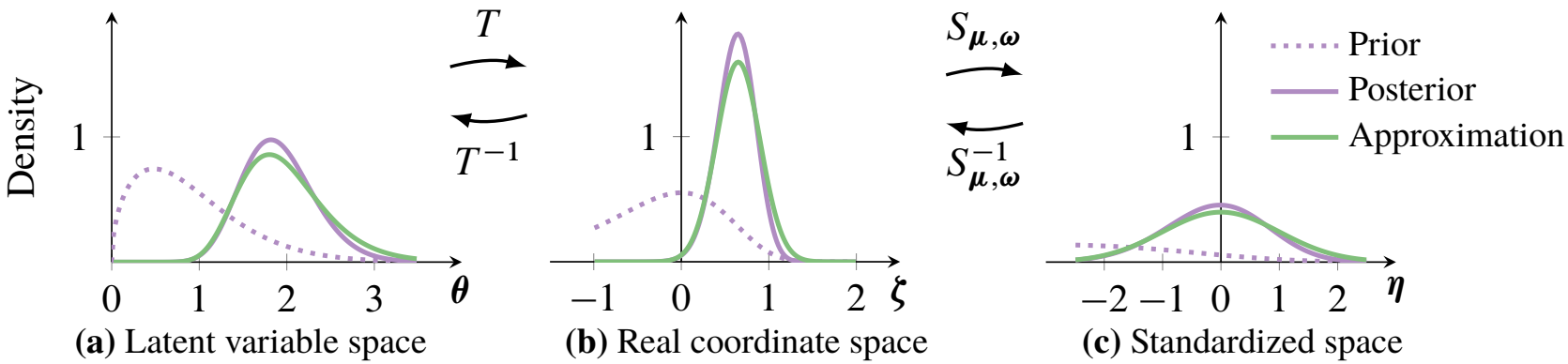
What would take us weeks → a single day.


Maia
Ermesinde
Guelfães
Leça do Balio
Santa Cruz do Bispo
Águas Santas
Guifões
São Mamede de Infesta
Matosinhos
Rio Tinto
Parque da Cidade
Gondomar
Barrosos
Alumiara
Lavadores
Vila Nova de Gaia
Valbom
Canidelo
Madalena
Avintes
EN 107
EN 14
A 3
A 41
EN 105-1
A 4
EM 540
EN 15-1
EN 208
EN 15
EM 612-1
EM 612
IP 1
EN 108
A 1
A 44
EN 222
EN 1-15
Rua 5 de Outubro
Rua de Santa Maria
Avenida da Liberdade
Rua Roberto
Via Norte-Via Norte
Avenida Barra Mar
Avenida Vieira de Castro
Rio Douro


Maia
Ermesinde
Santa Cruz do Bispo
Leça do Balio
Guelfães
Águas Santas
Guifões
São Mamede de Infesta
Matosinhos
Rio Tinto
Paranhos
Boavista
Parque da Cidade
São Pedro da Afurada
Alumiara
Lavadores
Canidelo
Madalena
Gondomar
Barrosos
Valbom
Avintes
EN 107
EN 14
A 3
A 4
EN 105-1
EN 540
EN 208
EN 105
EN 15-1
EN 12
EN 15
EM 612-1
EM 613
EM 612
A 1
IP 1
EN 108
A 44
EN 222
EN 1-15


Maia
Ermesinde
Santa Cruz do Bispo
Leça do Balio
Guelfães
Águas Santas
Guifões
São Mamede de Infesta
Matosinhos
Rio Tinto
Paranhos
Boavista
Porto
São Pedro da Afurada
Gondomar
Barrosos
Alumiara
Lavadores
Vila Nova de Gaia
Valbom
Canidelo
Madalena
Avintes
EN 14
EN 107
A 3
EN 105-1
A 4
EN 540
EN 208
EN 105
EN 15-1
EN 12
EN 15
A 1
IP 1
A 44
EN 108
EN 1-15
EN 612

ADVI details



(a) Latent variable space **(b)** Real coordinate space **(c)** Standardized space

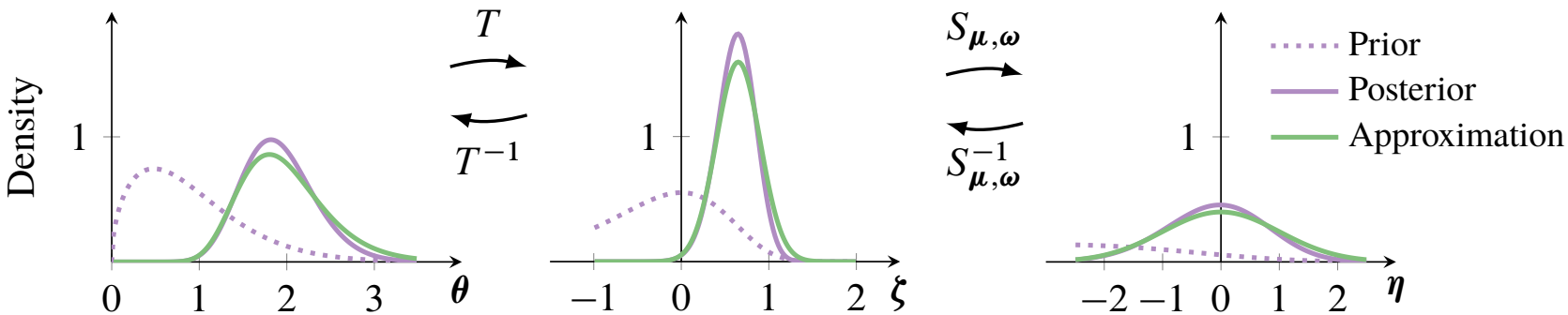
1. **Transform the model** from $p(\mathbf{z}, \mathbf{x})$ to $p(\zeta, \mathbf{x})$, where $\zeta \in \mathbb{R}^d$.

The mapping is in the joint,

$$p(\zeta, \mathbf{x}) = p\left(\mathbf{x}, T^{-1}(\zeta)\right) \left| \det J_{T^{-1}}(\zeta) \right|.$$

Stan provides a library for transforming probabilistic programs.

ADVI details



(a) Latent variable space **(b)** Real coordinate space **(c)** Standardized space

2. Redefine the variational problem with a Gaussian variational distribution.

The variational family is

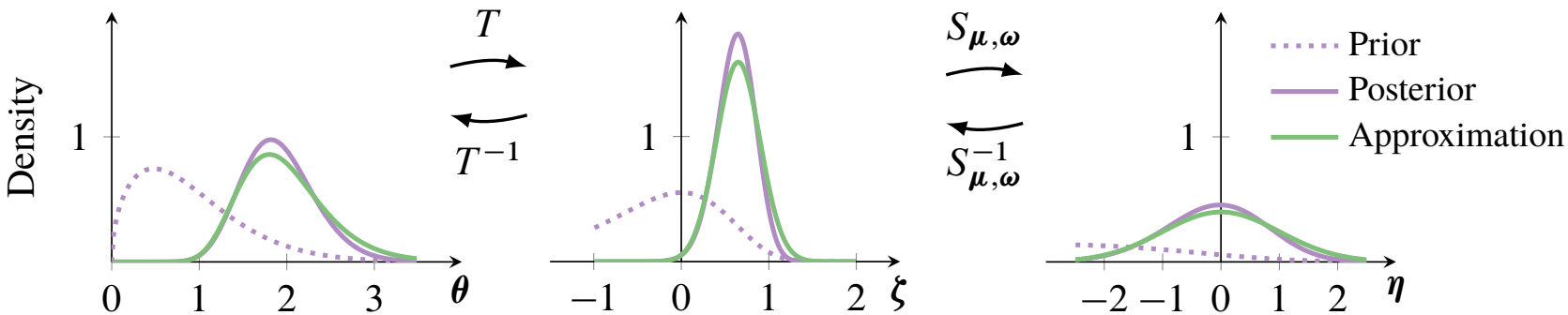
$$q(\boldsymbol{\zeta}\,;\,\boldsymbol{\nu}) = \mathcal{N}\left(\boldsymbol{\zeta}\,;\,\boldsymbol{\mu},\boldsymbol{\sigma}\right) = \prod_{k=1}^{K} \mathcal{N}\left(\zeta_k\,;\,\mu_k,\sigma_k\right).$$

The evidence lower bound is

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

We can use this variational problem **across transformable models**.

ADVI details



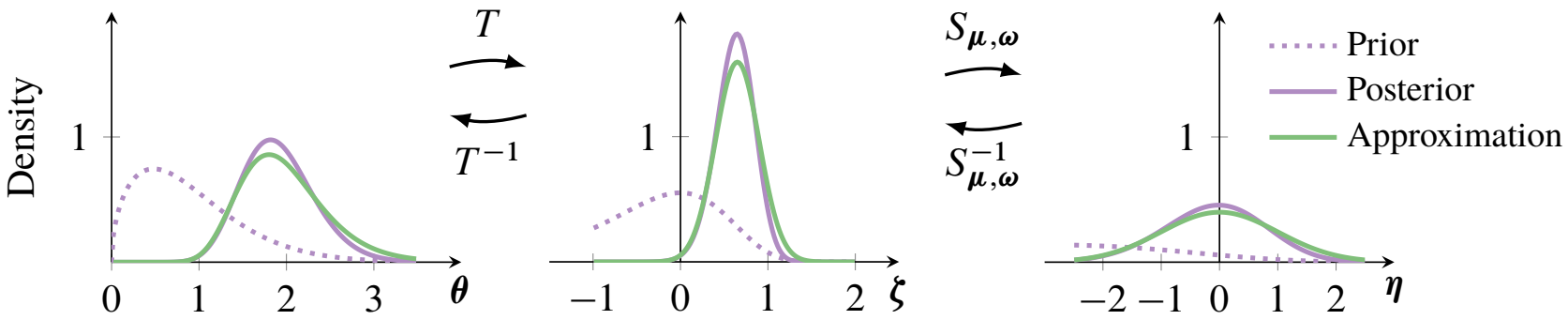
(a) Latent variable space **(b)** Real coordinate space **(c)** Standardized space

3. Use the reparameterization trick, where $\boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{I})$,

$$\mathscr{L} = \mathbb{E}\Big[\log p\left(\mathbf{x}, T^{-1}(S^{-1}_{\mu,\omega}(\boldsymbol{\eta}))\right) + \log\left|\det J_{T^{-1}}\left(S^{-1}_{\mu,\omega}(\boldsymbol{\eta})\right)\right|\Big] + \sum_{k=1}^{K}\omega_k.$$

This is a second transformation, of the already transformed variable.

ADVI details

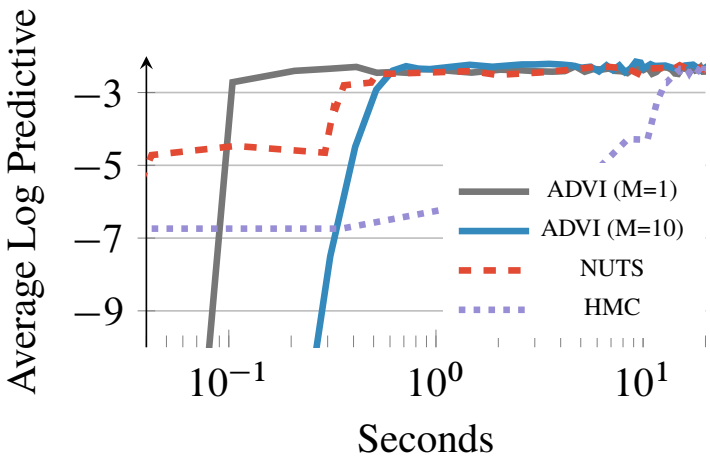


(a) Latent variable space **(b)** Real coordinate space **(c)** Standardized space

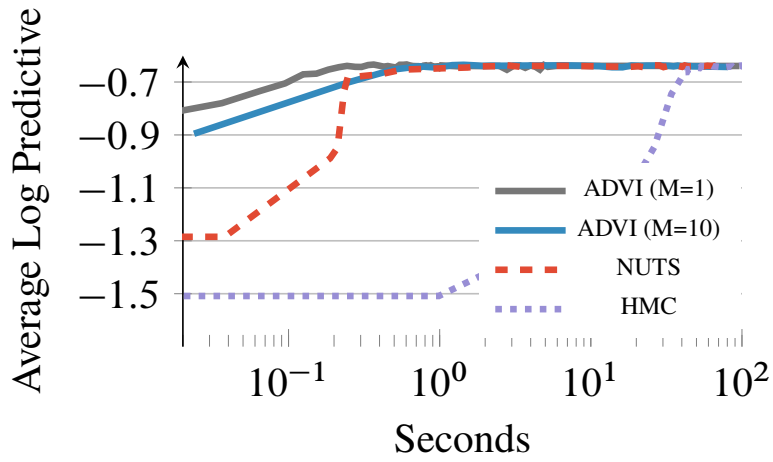
4. ***Optimize***.

- Form MC estimates of the gradient and fit with stochastic optimization.
- For large data, use stochastic variational inference too.
- Autodiff handles the reparameterization gradient.

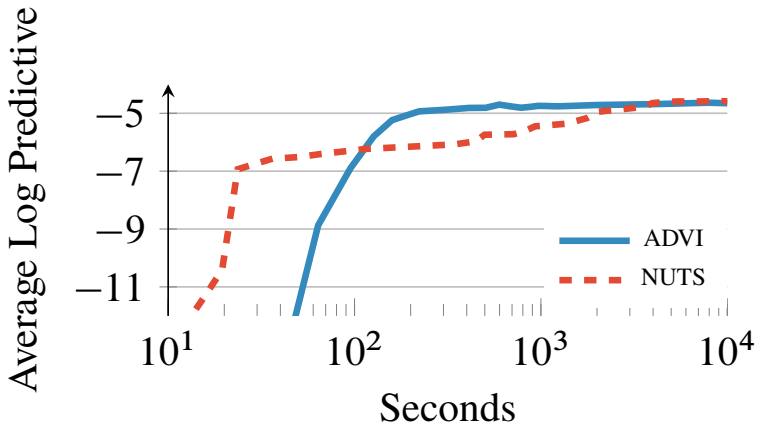
Other benchmarks



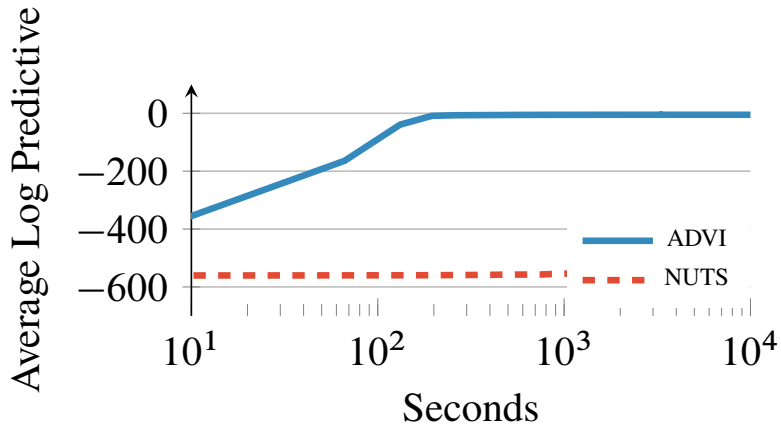
(a) Linear Regression with ARD



(b) Hierarchical Logistic Regression



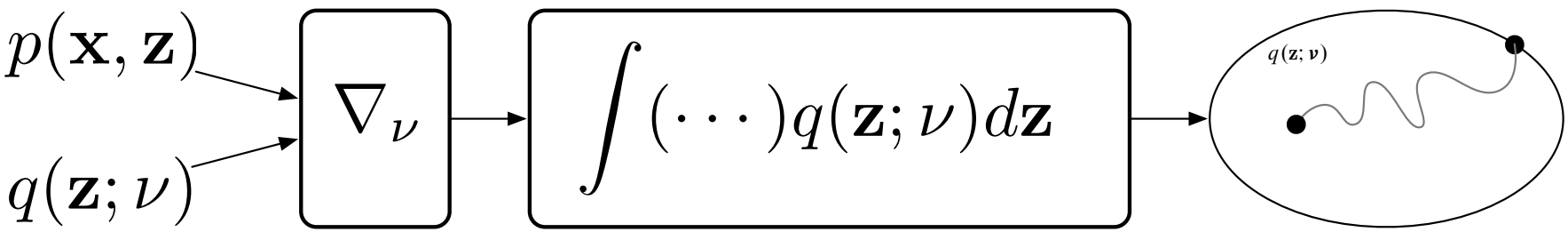
(a) Gamma Poisson Predictive Likelihood



(b) Dirichlet Exponential Predictive Likelihood

[Kucukelbir+ 2016]

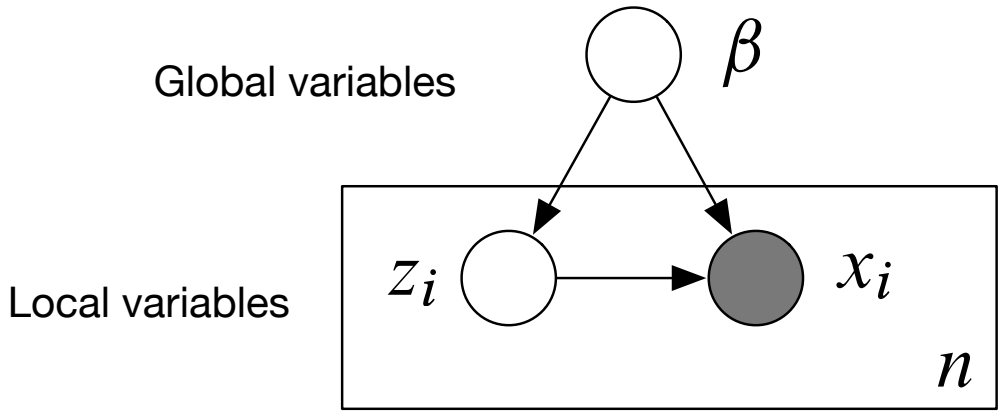
Roadmap



- Score function estimator and basic black box variational inference
- Reparameterization gradient
- Autodifferentiation VI and probabilistic programming

Monte Carlo gradients enable *black box variational inference*, algorithms that efficiently perform Bayesian computation in any model.

Nonconjugate models



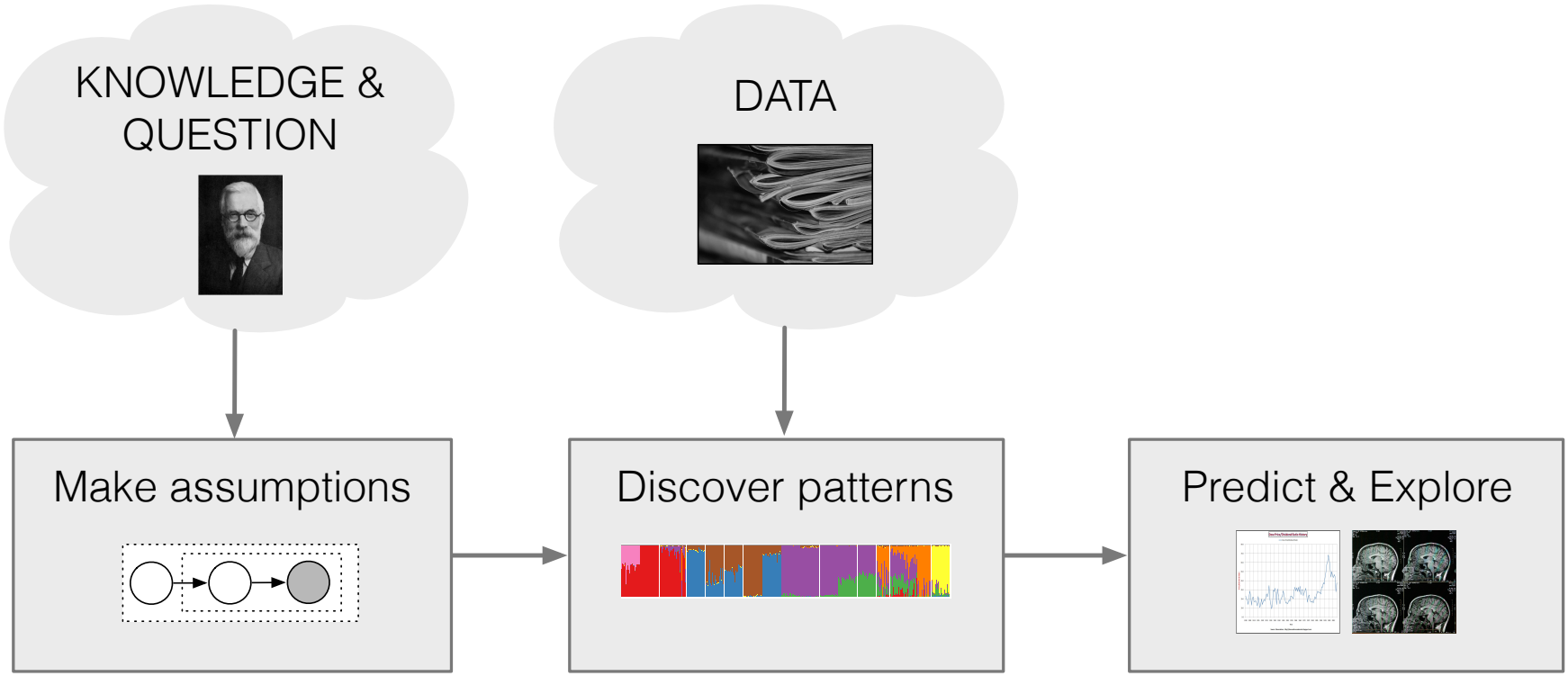
$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Nonlinear time series models
- Deep latent Gaussian models
- Models with attention
- Generalized linear models
- Stochastic volatility models
- Discrete choice models
- Bayesian neural networks
- Deep exponential families
- Correlated topic models
- Sigmoid belief networks

PART IV

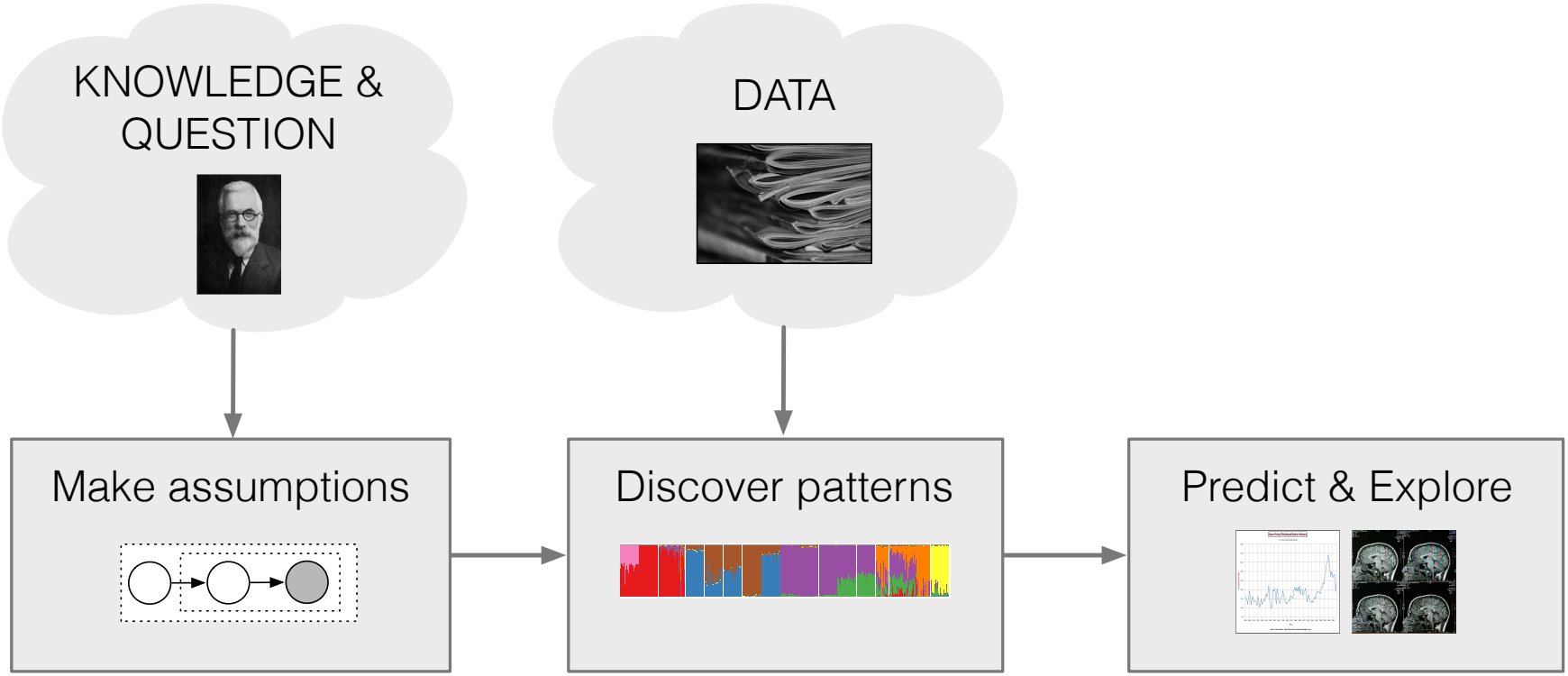
Summary

The probabilistic pipeline



- Customized data analysis is important to many fields.
- Pipeline separates **assumptions**, **computation**, **application**
- Eases collaborative solutions to statistics problems

The probabilistic pipeline



- **Posterior inference** is the key algorithmic problem.
- Answers the question: What does this model say about this data?
- Our goal: **General** and **scalable** approaches to posterior inference

Probabilistic machine learning

- A probabilistic model is a joint distribution of hidden variables $\mathbf{z}$ and observed variables $\mathbf{x}$,

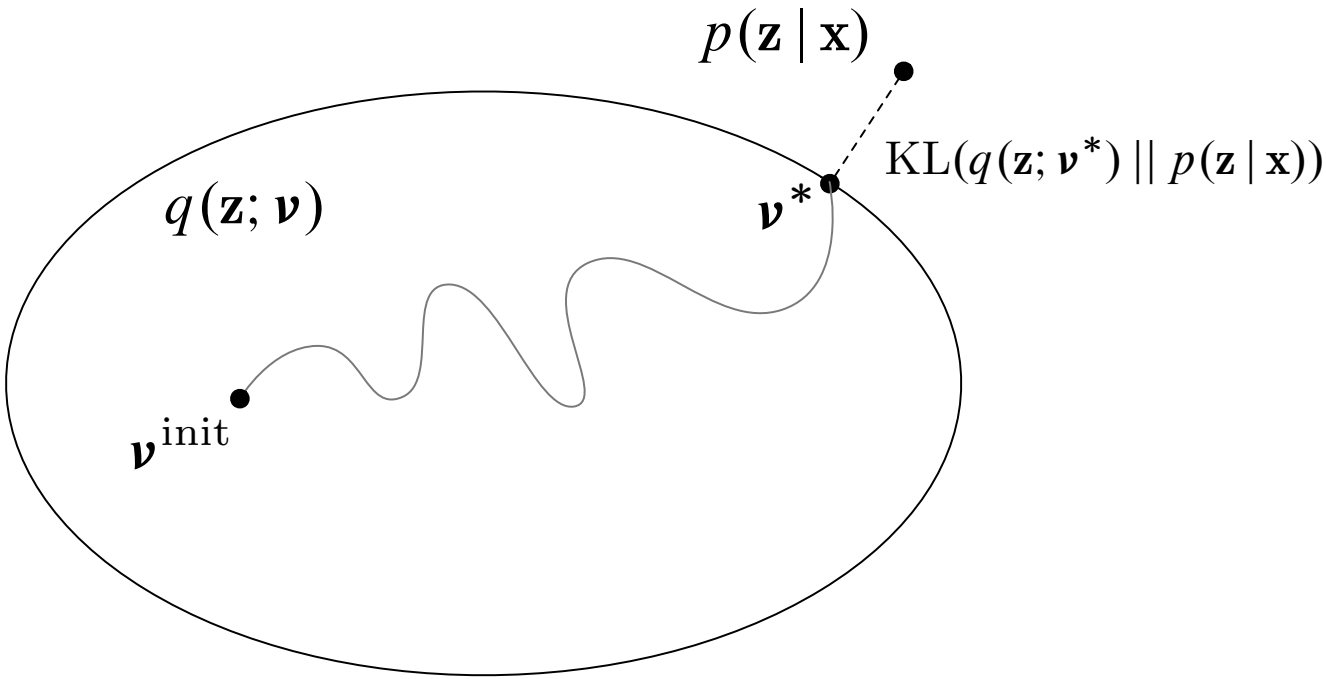
$$p(\mathbf{z}, \mathbf{x}).$$

- Inference about the unknowns is through the **posterior**, the conditional distribution of the hidden variables given the observations

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})}.$$

- For most interesting models, the denominator is not tractable. We appeal to **approximate posterior inference**.

Variational inference

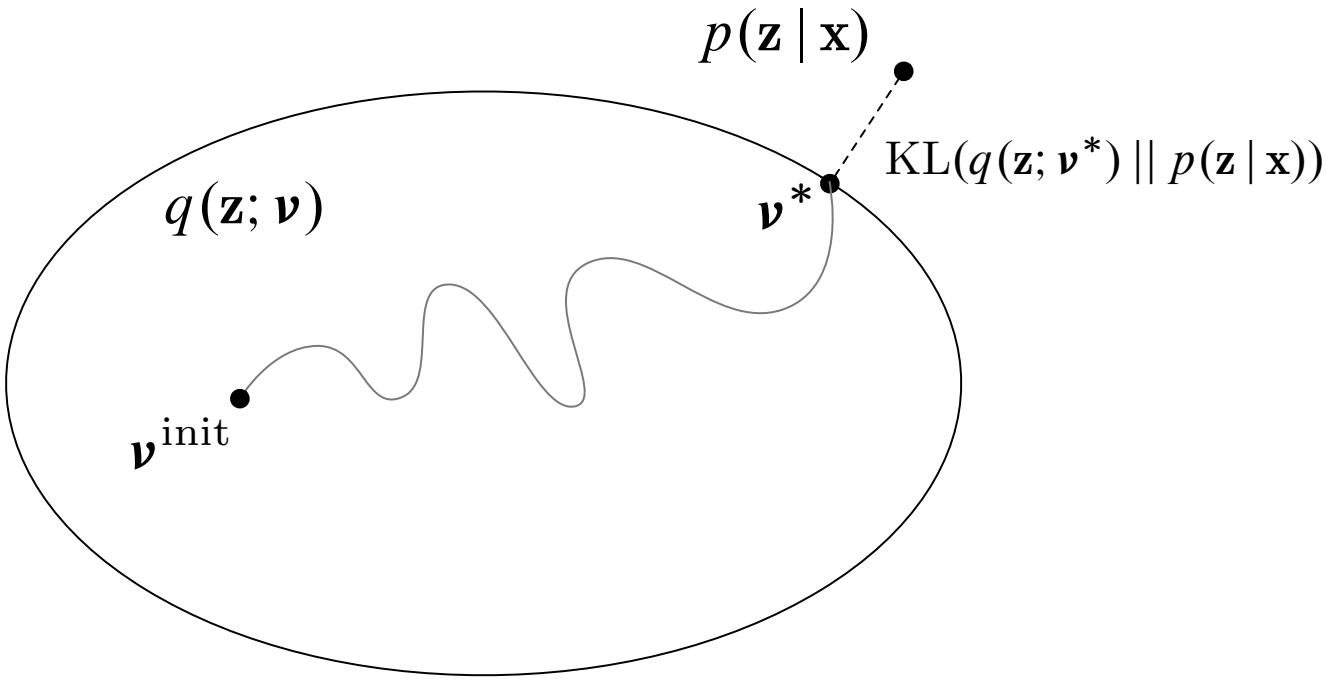


- VI turns **inference into optimization**.
- Posit a **variational family** of distributions over the latent variables,

$$q(\mathbf{z}; \boldsymbol{\nu})$$

- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.
  (There are alternative divergences, which connect to algorithms like EP, BP, and others.)

Variational inference



With **stochastic optimization** we can

- scale up VI to massive data
- enable VI on a wide class of difficult models
- enable VI with elaborate and flexible families of approximations

Variational inference

Part I: Main ideas and historical context

Jordan+, *Introduction to Variational Methods for Graphical Models*, 1999

Part II: Mean-field VI and stochastic VI

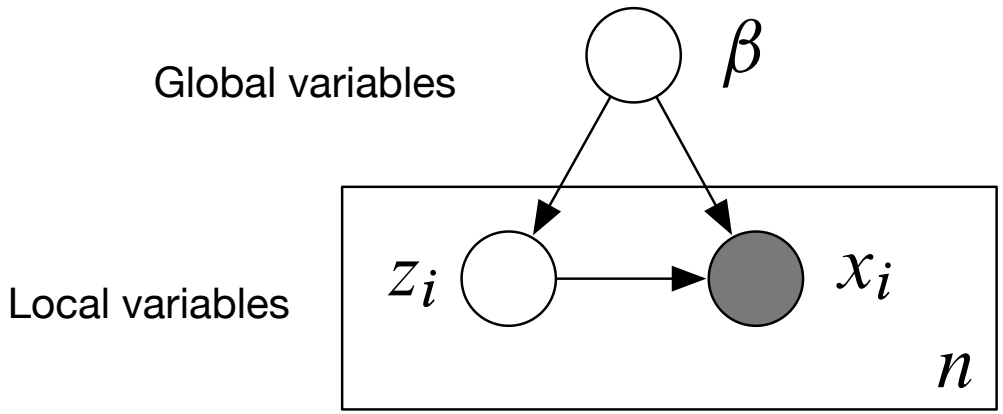
Ghahramani and Beal, *Propagation Algorithms for Variational Bayesian Learning*, 2001
Hoffman+, *Stochastic Variational Inference*, 2013

Part III: Stochastic gradients of the ELBO

Ranganath+, *Black Box Variational Inference*, 2014
Rezende+, *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*, 2014
Kucukelbir+ *Automatic Differentiation Variational Inference*, 2016

Part IV: Summary

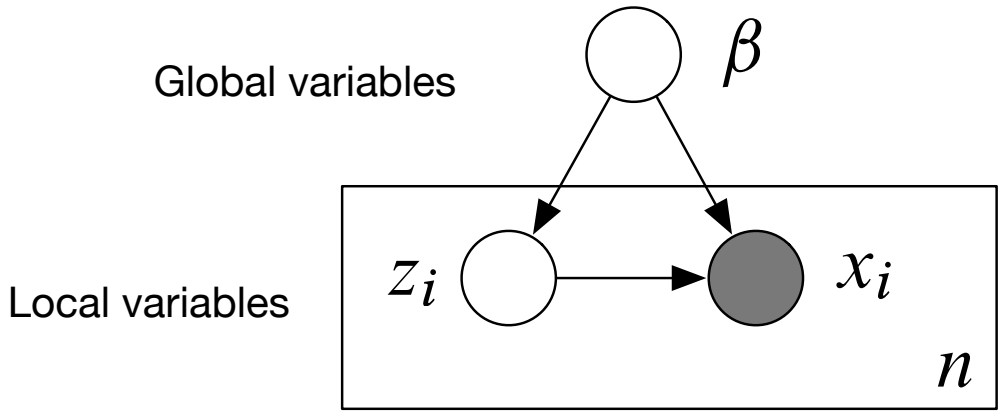
Conditionally conjugate models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Bayesian mixture models
- Time series models
  (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization
  (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression
  (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models
  (LDA and some variants)

Nonconjugate models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Nonlinear time series models
- Deep latent Gaussian models
- Models with attention
- Generalized linear models
- Stochastic volatility models
- Discrete choice models
- Bayesian neural networks
- Deep exponential families
- Correlated topic models
- Sigmoid belief networks

Edward: Probabilistic modeling, inference, and criticism

`github.com/blei-lab/edward`

(lead by Dustin Tran)

- Theory
  - MCMC has been widely analyzed and studied
  - The theoretical properties of VI are far less explored.
    (But see work by Hall, Bickel, others.)
  - E.g., we are working on variational Bernstein-von-Mises

- Optimization
  - Can we find better local optima?
  - Can we accelerate convergence?

- Alternative divergences
  - KL is chosen for its convenient properties, but it has some undesirable properties (e.g. zero-forcing)
  - Can we use other divergences?

- Better estimates of posterior variance
  - E.g., full-rank ADVI [Kucukelbir+ 2016]
  - posthoc correction [Girodano and Broderick 2016]