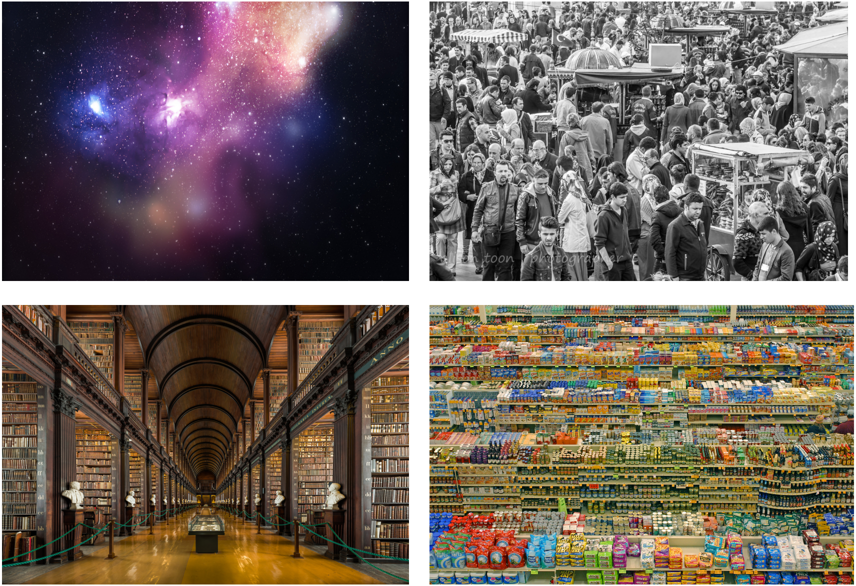
VARIATIONAL INFERENCE:
FOUNDATIONS AND INNOVATIONS

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



We have ***complicated data***; we want to ***make sense*** of it.

What is ***complicated data***?

- many data points; many dimensions
- elaborate structures and relationships (e.g., text)
- different interconnected modalities (e.g., images, links, text, clicks)

What is ***making sense of data***?

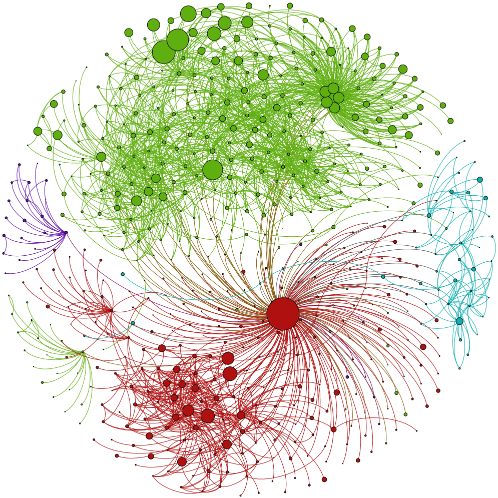
- make predictions about the future
- identify interpretable patterns
- do science: confirm, elaborate, form causal theories

PROBABILISTIC MACHINE LEARNING

- ML methods that **connect domain knowledge to data**.
- Provides a computational methodology for analyzing data
- Goal: A methodology that is **expressive, scalable, easy to develop**

APPLIED BAYESIAN STATISTICS

- Statistical methods that **connect domain knowledge to data**.
- Provides a computational methodology for analyzing data
- Goal: A methodology that is **expressive, scalable, easy to develop**



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

[Gopalan and Blei PNAS 2013]

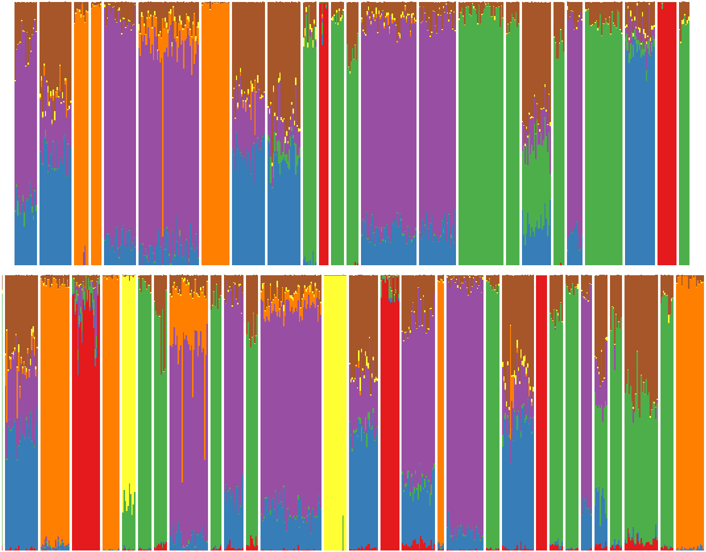
1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

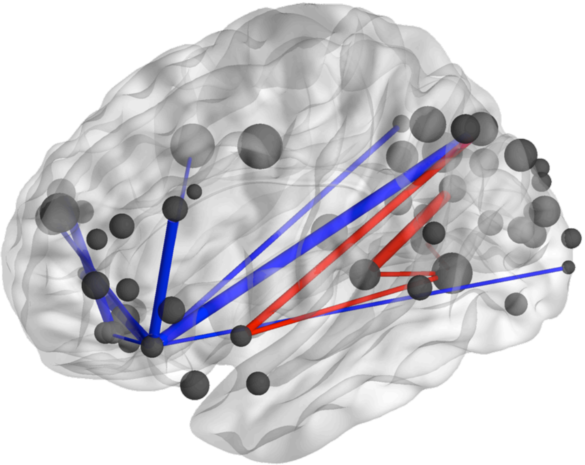
Topics found in 1.8M articles from the New York Times

[Hoffman+ JMLR 2013]



Population analysis of 2 billion genetic measurements

[Gopalan+ Nature Genetics 2016]

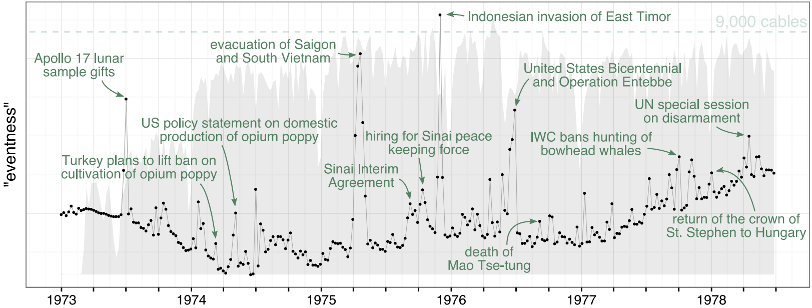


Neuroscience analysis of 220 million fMRI measurements

[Manning+ PLOS ONE 2014]

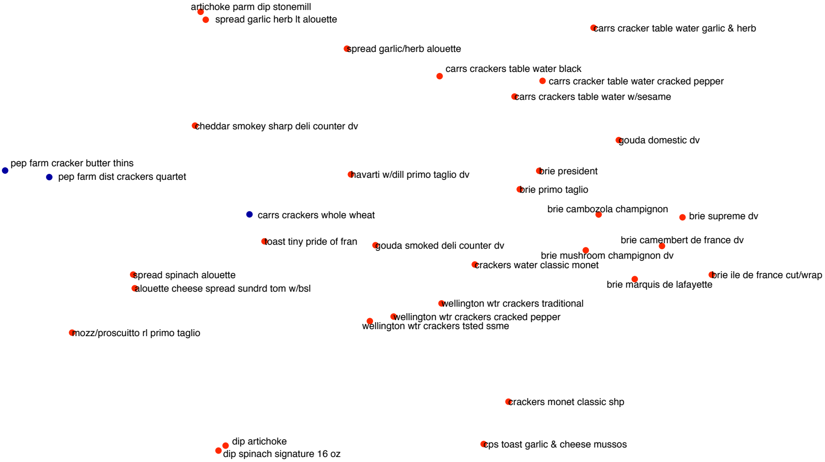
Analysis of 1.7M taxi trajectories, in Stan

[Kucukelbir+ JMLR 2016]



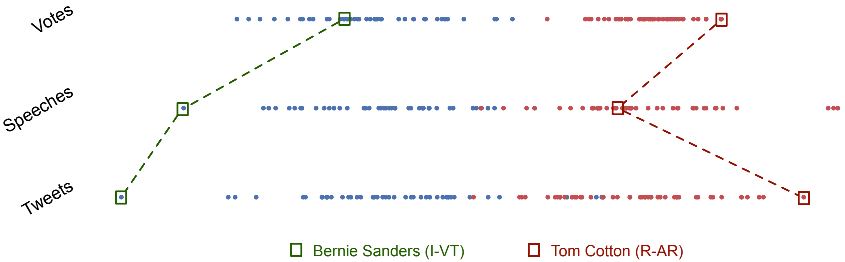
Analysis of 2M declassified cables from the State Dept

[Chaney+ EMNLP 2016]



(Fancy) discrete choice analysis of 5.7M purchases

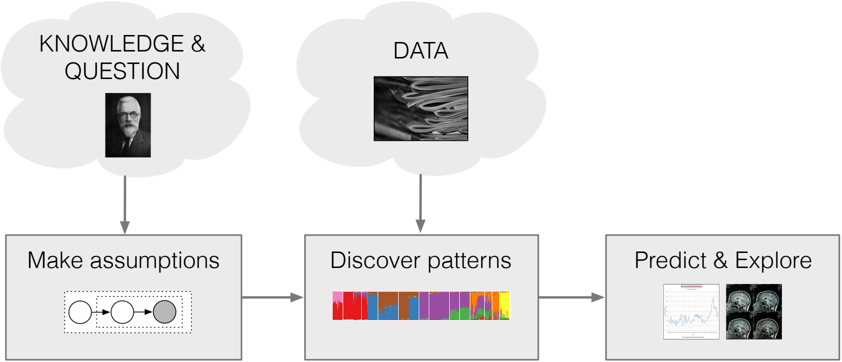
[Ruiz+ 2017]



Estimating political sentiment based on text

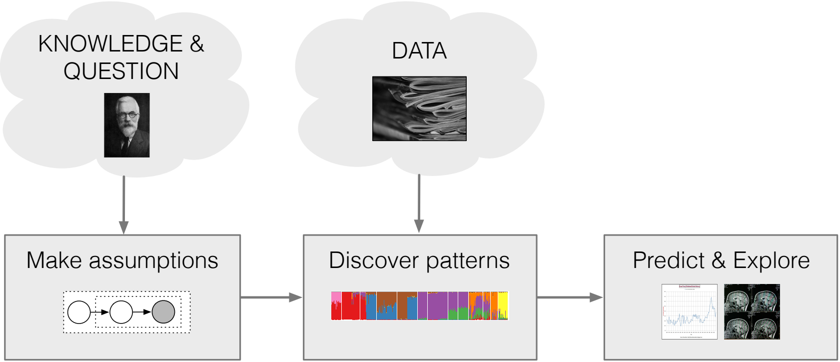
[Vafa+ in progress]

The probabilistic pipeline



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

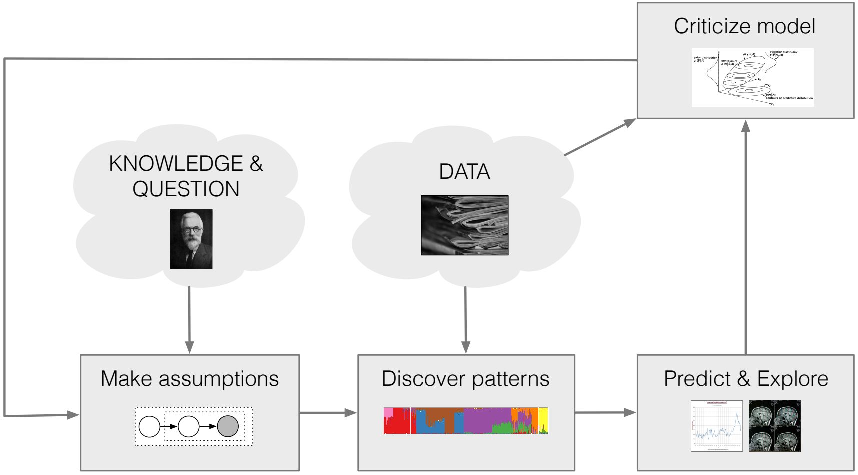
The probabilistic pipeline



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

The story I'm trying to tell

- **1950's and 1960's**: Bayesian inference is nearly impossible.
- **1970's and 1980's**: Early work on approximate Bayesian inference—Metropolis-Hastings, importance sampling.
- **1990's**: The Gibbs sampler and better computation. We can fit many Bayesian models. (Early work on variational inference.)
- **2000's**: VI in practice; we can fit Bayesian models faster.
- **2010's**: VI is scalable and general; robust probabilistic programming. We can explore large classes of models and with large data sets.



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

Introduction

Probabilistic machine learning

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

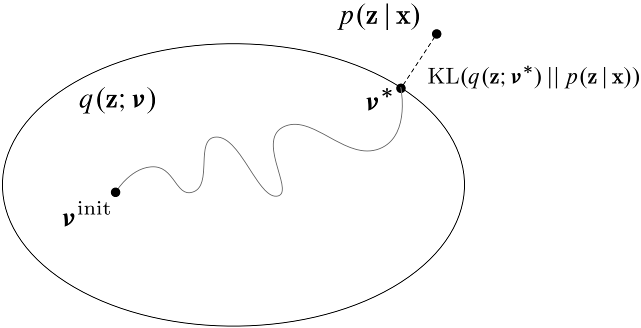
$$p(\mathbf{z}, \mathbf{x}).$$

- 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



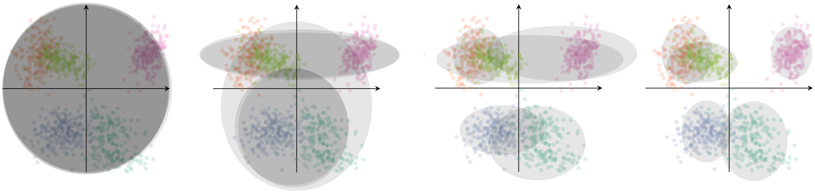
- VI solves **inference** with **optimization**.
  (Contrast this with MCMC.)

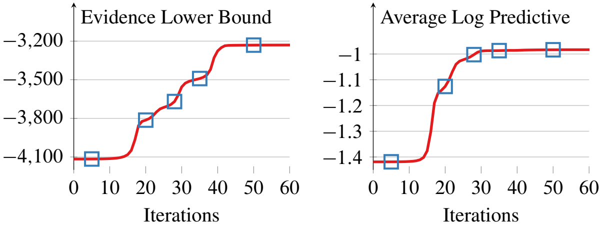
- Posit a **variational family** of distributions over the latent variables,

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

- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.

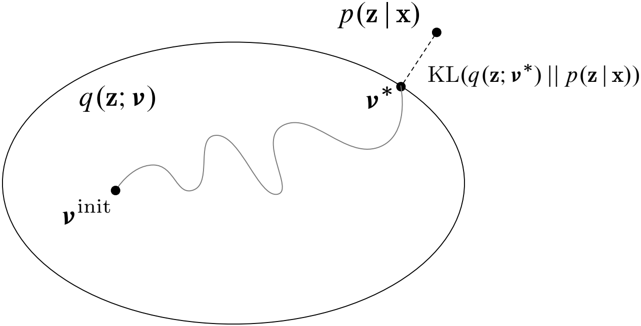
Example: Mixture of Gaussians





[images by Alp Kucukelbir; Blei+ 2016]

Variational inference



VI solves **inference** with **optimization**.

In this tutorial:

- the **basics** of VI
- VI for **massive data**
- VI for a wide class of **difficult models**

“Prerequisites”

- A little probability
  - joint distribution, conditional distribution
  - expectation, conditional expectation

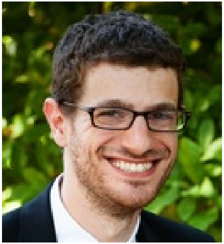
- A little optimization
  - the main idea
  - gradient-based optimization
  - coordinate-ascent optimization

- A little Bayesian statistics (but you don’t have to be a Bayesian!)

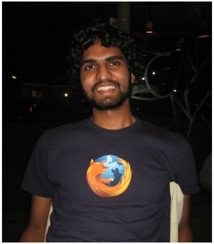
What you will learn about

- The basics of variational inference (VI)
  - Mean-field variational inference
  - Coordinate ascent optimization for VI
- Stochastic variational inference for massive data
- Black box variational inference
  - Score gradients
  - Reparameterization gradients
  - Amortized variational families, the variational autoencoder
- Models, along the way
  - Latent Dirichlet allocation and topic models
  - Deep exponential families
  - Embedding models of consumer behavior
  - Deep generative models

Collaborators



Matt Hoffman
(Google)

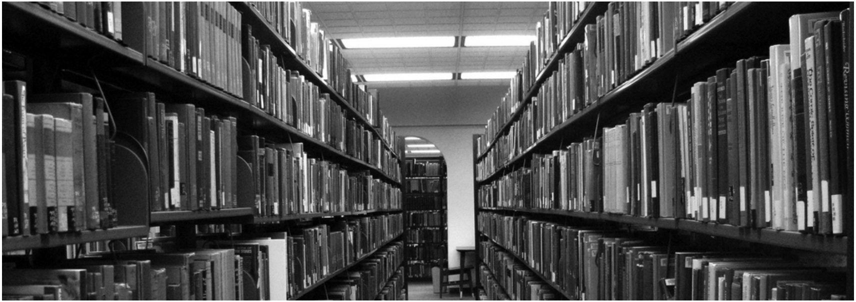


Rajesh Ranganath
(NYU)

Alp Kucukelbir
(Fero Labs)

Variational Inference
& Stochastic Variational Inference

Motivation: Topic Modeling



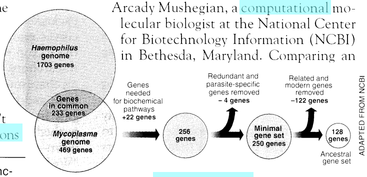
Topic models use posterior inference to discover the hidden thematic structure in a large collection of documents.

Model: 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

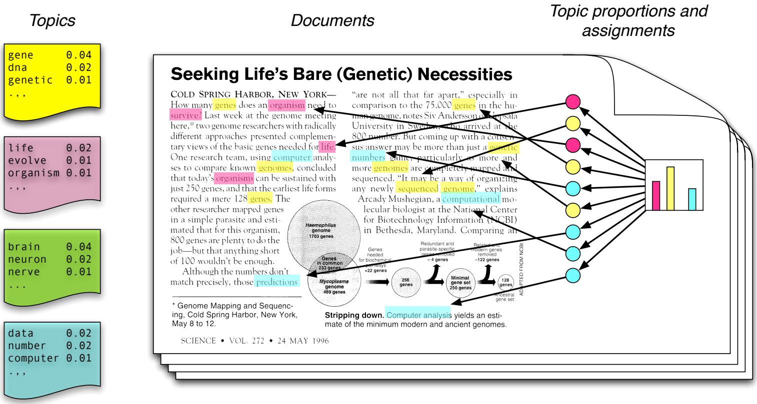


Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

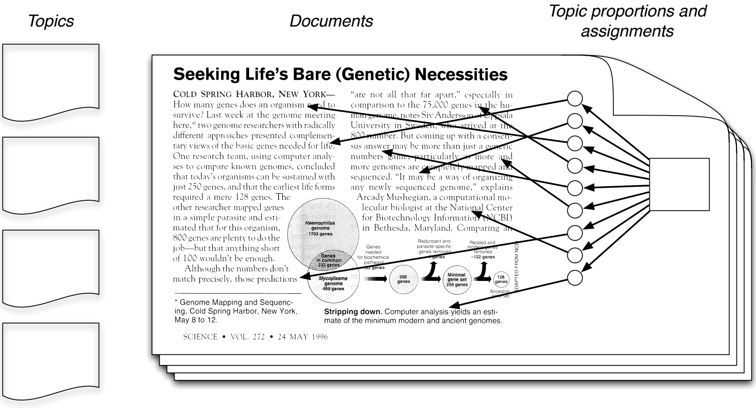
Documents exhibit multiple topics.

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

Latent Dirichlet Allocation (LDA)

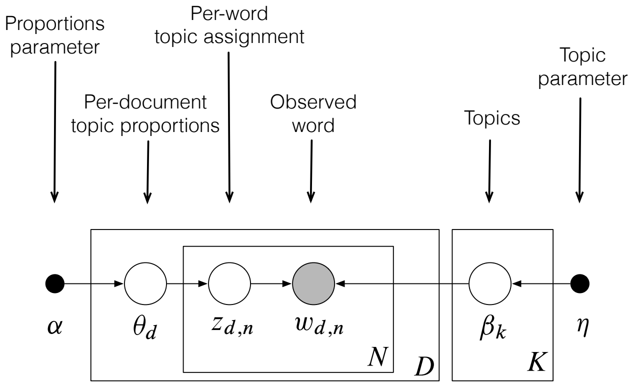


- But we only observe the documents; everything else is hidden.
- So we want to calculate the posterior

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

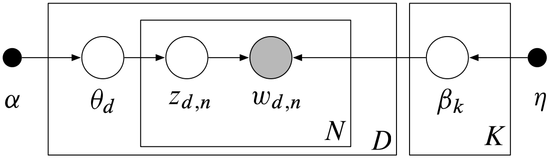
(Note: millions of documents; billions of latent variables)

LDA as a Graphical Model



- A schematic of the **generative process**
- Defines a **factorization of the joint distribution**
- Connects to **assumptions** and **algorithms**

Posterior Inference

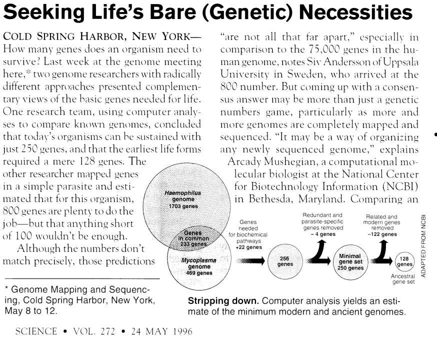


- The posterior of the latent variables given the documents is

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

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

Mean-field variational inference for LDA



Seeking Life's Bare (Genetic) Necessities

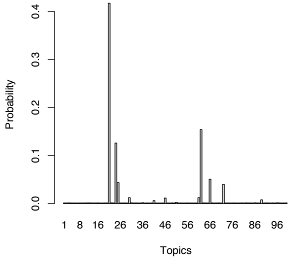
COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known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

SCIENCE • VOL. 272 • 24 MAY 1996



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

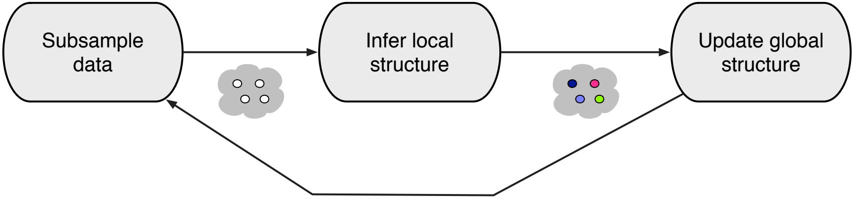
1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

Topics found in 1.8M articles from the New York Times

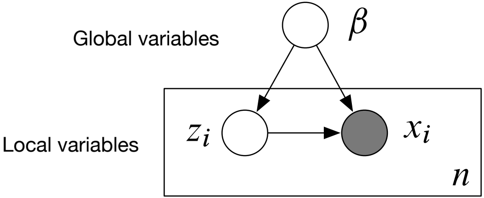
Mean-field VI and Stochastic VI



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

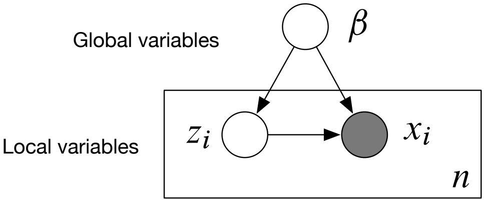


$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- 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,

$$\begin{aligned} p(z_i \mid \beta, x_i) &= \text{expfam}(z_i \,;\, \eta_\ell(\beta, x_i)) \\ p(\beta \mid \mathbf{z}, \mathbf{x}) &= \text{expfam}(\beta \,;\, \eta_g(\mathbf{z}, \mathbf{x})), \end{aligned}$$

where $\text{expfam}(z\,;\,\eta) = h(z)\exp\{\eta^\top z - a(\eta)\}$.

Aside: The exponential family

$$p(x) = h(x)\exp\{\eta^\top t(x) - a(\eta)\}$$

Terminology:

η	the natural parameter
$t(x)$	the sufficient statistics
$a(\eta)$	the log normalizer
$h(x)$	the base density

Aside: The exponential family

$$p(x) = h(x) \exp\{\eta^\top t(x) - a(\eta)\}$$

- The log normalizer is

$$a(\eta) = \log \int \exp\{\eta^\top t(x)\} dx$$

- It ensures the density integrates to one.
- Its gradient calculates the expected sufficient statistics

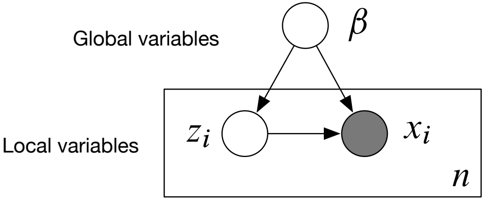
$$\mathbb{E}[t(X)] = \nabla_\eta a(\eta).$$

Aside: The exponential family

$$p(x) = h(x) \exp\{\eta^\top t(x) - a(\eta)\}$$

- Many common distributions are in the exponential family
  - Bernoulli, categorical, Gaussian, Poisson, Beta, Dirichlet, Gamma, etc.
- Outlines the theory around conjugate priors and corresponding posteriors
- Connects closely to variational inference [Wainwright and Jordan, 2008]

Conditionally conjugate models



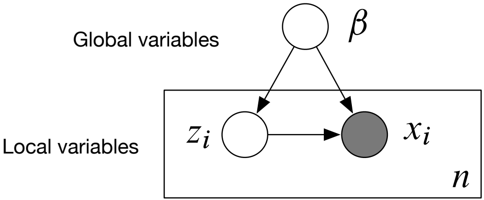
$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Each **complete conditional** is in the exponential family.
- The global parameter comes from conjugacy [Bernardo and Smith, 1994]

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

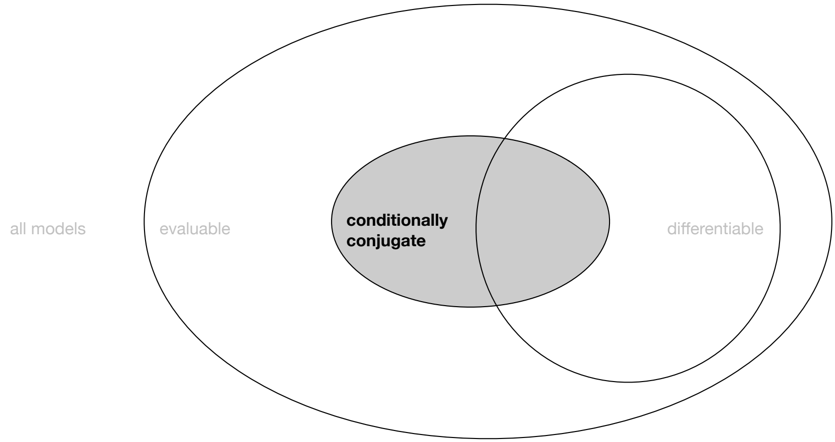
  where α is a hyperparameter and $t(\cdot)$ are sufficient statistics for $[z_i, x_i]$.

Conditionally conjugate models

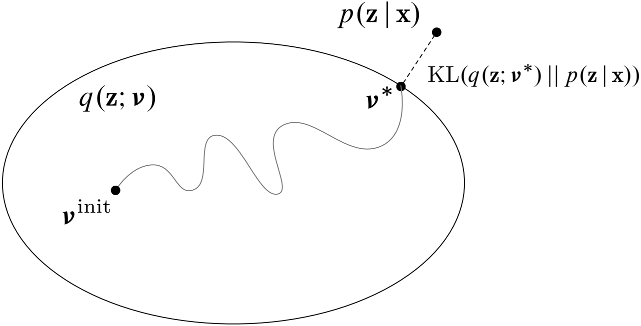


$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- 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)


all models
evaluable
conditionally conjugate
differentiable

Variational inference



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

The Kullback-Leibler divergence

- The KL divergence between two distributions is

$$\mathrm{KL}(q \,\|\, p) = \mathbb{E}_q\left[\log \frac{q(\beta, \mathbf{z})}{p(\beta, \mathbf{z} \,|\, \mathbf{x})}\right]$$

- It is **non-negative**, $\mathrm{KL}(q \,\|\, p) \geq 0$. It **equals zero** when $q = p$.

- It is **not symmetric**, $\mathrm{KL}(p \,\|\, q) \neq \mathrm{KL}(q \,\|\, p)$

The evidence lower bound

$$\mathscr{L}(\boldsymbol{\nu}) = \underbrace{\mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{Expected complete log likelihood}} - \underbrace{\mathbb{E}_q[\log q(\beta, \mathbf{z};\ \boldsymbol{\nu})]}_{\text{Negative entropy}}$$

- 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.

The evidence lower bound

$$\mathscr{L}(\boldsymbol{\nu}) = \underbrace{\mathbb{E}_q[\log p(\mathbf{x} \mid \beta, \mathbf{z})]}_{\text{Expected log likelihood of data}} - \underbrace{\mathrm{KL}(q(\beta, \mathbf{z};\, \boldsymbol{\nu}) \,||\, p(\beta, \mathbf{z}))}_{\text{KL between variational and prior}}$$

- 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 MLE.
  - The second term encourages $q(\cdot)$ to be close to the prior.
- Caveat: The ELBO is not convex.

Two properties of the ELBO

- Consider the KL divergence between q and $p(\beta, \mathbf{z} \mid \mathbf{x})$. Write it as

$$\mathrm{KL} = \underbrace{\mathbb{E}_q[\log q(\beta, \mathbf{z}; \boldsymbol{\nu})] - \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{negative ELBO}} + \underbrace{\log p(\mathbf{x})}_{\text{not a function of } \boldsymbol{\nu}}.$$

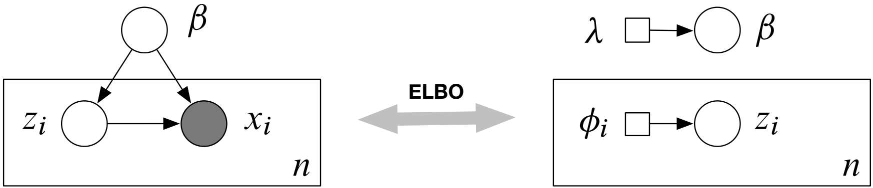
  Maximizing the ELBO minimizes the KL.

- Rearrange to write the log marginal as

$$\log p(\mathbf{x}) = \underbrace{\mathrm{KL}}_{\text{a positive value}} + \underbrace{\mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{the ELBO}}.$$

  The ELBO is a lower bound, $\mathscr{L}(\boldsymbol{\nu}) \leq \log p(\mathbf{x})$.

Mean-field variational inference



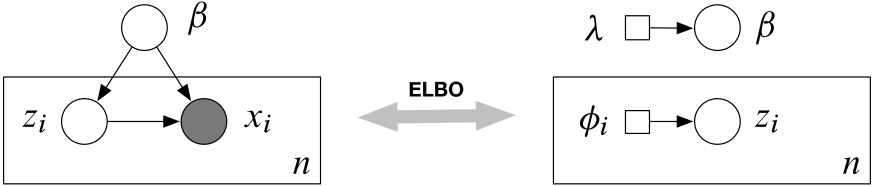
- We need to specify the form of $q(\beta, \mathbf{z})$; it defines the **variational family**.
- The **mean-field family** is fully factorized,

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

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

$$p(\beta \,|\, \mathbf{z}, \mathbf{x}) = \text{expfam}(\beta \,;\, \eta_g(\mathbf{z}, \mathbf{x}))$$
$$q(\beta \,;\, \lambda) = \text{expfam}(\beta \,;\, \lambda)$$

Mean-field variational inference



- Optimize the ELBO,

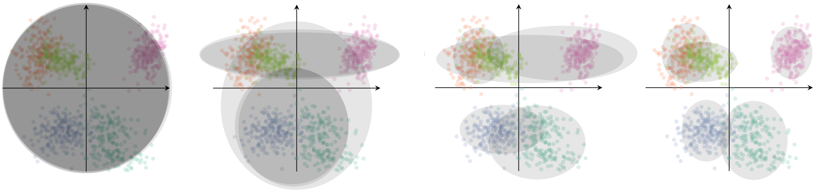
$$\mathscr{L}(\lambda, \boldsymbol{\phi}) = \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\beta, \mathbf{z})].$$

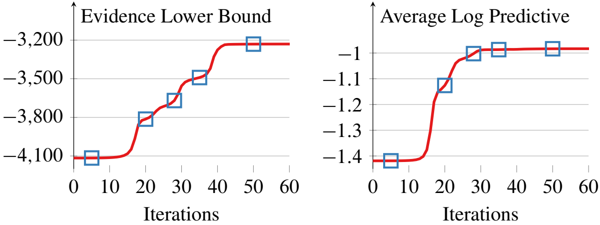
- Traditional VI uses coordinate ascent [Ghahramani and Beal, 2001]

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

- Iteratively update each parameter, holding others fixed.
  - Notice the relationship to Gibbs sampling [Gelfand and Smith, 1990] .

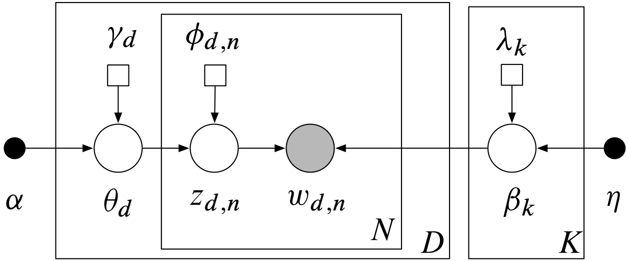
Example: Mixture of Gaussians





[images by Alp Kucukelbir; Blei+ 2016]

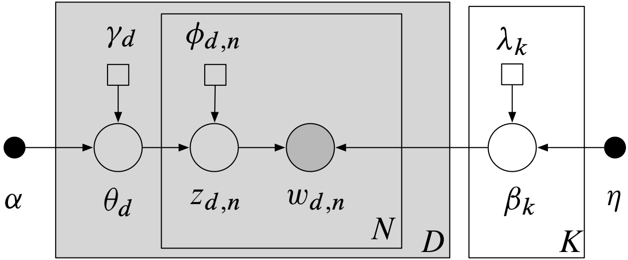
Mean-field variational inference for LDA



- 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

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

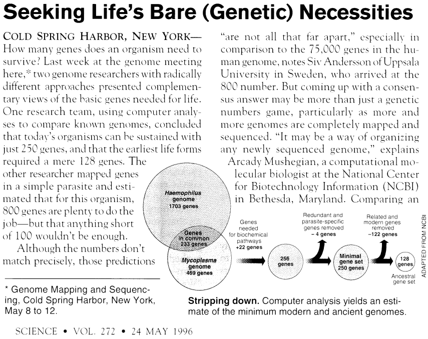
Mean-field variational inference for LDA



- In the "local step" we iteratively update the parameters for each document, holding the topic parameters fixed.

$$
\begin{aligned}
\gamma^{(t+1)} &= \alpha + \textstyle\sum_{n=1}^{N} \phi_n^{(t)} \\
\phi_n^{(t+1)} &\propto \exp\{\mathbb{E}[\log \theta] + \mathbb{E}[\log \beta_{.,w_n}]\}.
\end{aligned}
$$

Mean-field variational inference for LDA



Seeking Life's Bare (Genetic) Necessities

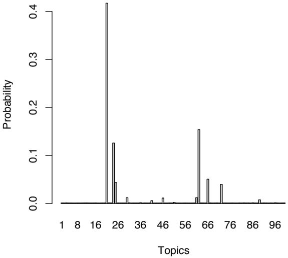
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known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

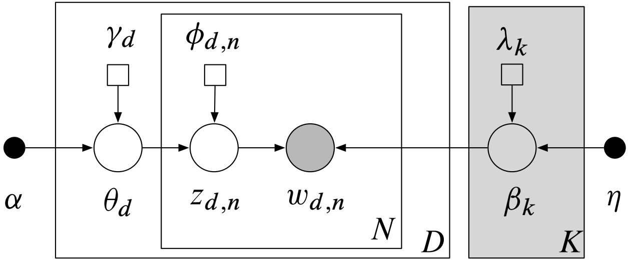
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

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

SCIENCE • VOL. 272 • 24 MAY 1996



Mean-field variational inference for LDA



- In the "global step" we aggregate the parameters computed from the local step and update the parameters for the topics,

$$\lambda_k = \eta + \sum_d \sum_n w_{d,n} \phi_{d,n}.$$

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

Coordinate ascent variational inference (CAVI)

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

Coordinate ascent variational inference (CAVI)

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

Initialize λ randomly.

Coordinate ascent variational inference (CAVI)

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

Initialize λ randomly.

while *not converged* **do**

Coordinate ascent variational inference (CAVI)

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

Initialize λ randomly.

while *not converged* **do**

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

Set local parameter

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

end

Coordinate ascent variational inference (CAVI)

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

Initialize λ randomly.

while *not converged* **do**

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

    Set local parameter

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

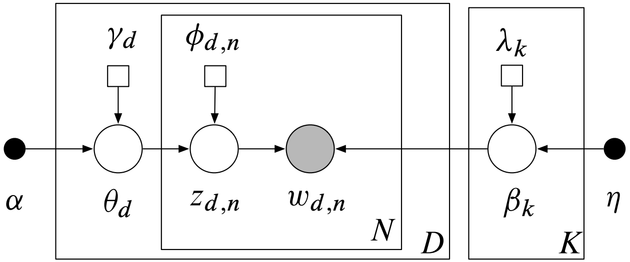
  end

  Set global parameter

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

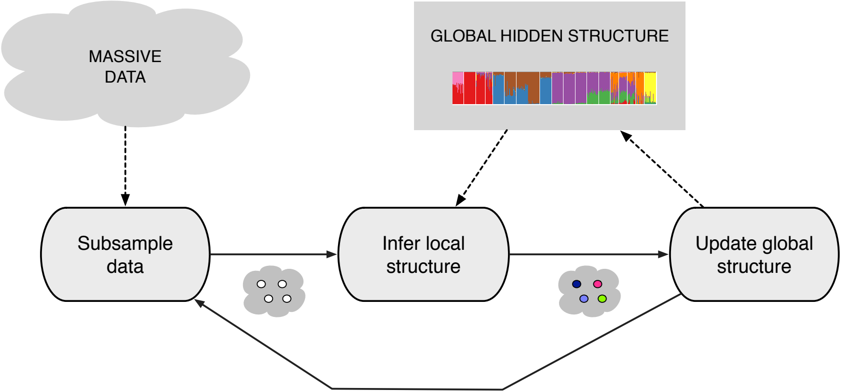
end

Stochastic variational inference (SVI)



- 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; Hoffman+ 2013]

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

- Construct a **noisy natural gradient**,

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

- It is **good for stochastic optimization.**
  - Its expectation is the exact gradient (*unbiased*).
  - It only depends on optimized local parameters of one data point (*fast*).
  - We don't need to store those optimized parameters (*cheap*).

Stochastic variational inference

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

Stochastic variational inference

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

Initialize λ randomly. Set ρ_t appropriately.

Stochastic variational inference

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

Initialize λ randomly. Set ρ_t appropriately.

while *not converged* **do**

Stochastic variational inference

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

Initialize λ randomly. Set ρ_t appropriately.

while *not converged* **do**

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

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

Stochastic variational inference

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

Initialize λ randomly. Set ρ_t appropriately.

while *not converged* **do**

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

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

Stochastic variational inference

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

Initialize λ randomly. Set ρ_t appropriately.

while *not converged* **do**

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

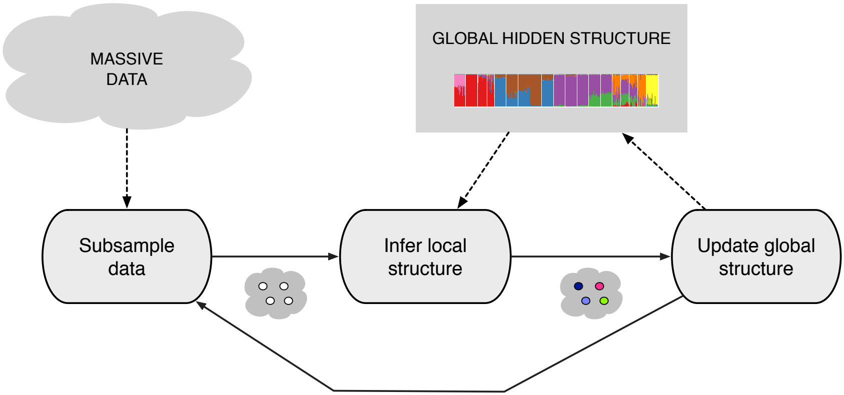
$$\hat{\lambda} = \alpha + n\mathbb{E}_\phi[t(Z_j, x_j)].$$

Set global parameter

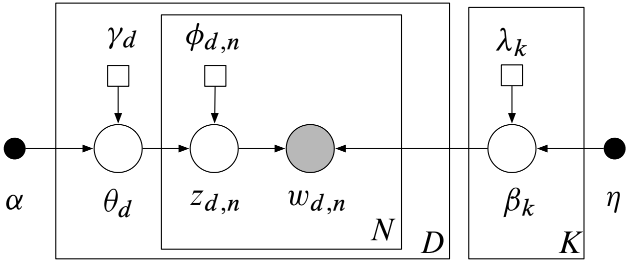
$$\lambda = (1 - \rho_t)\lambda + \rho_t \hat{\lambda}.$$

end

Stochastic variational inference

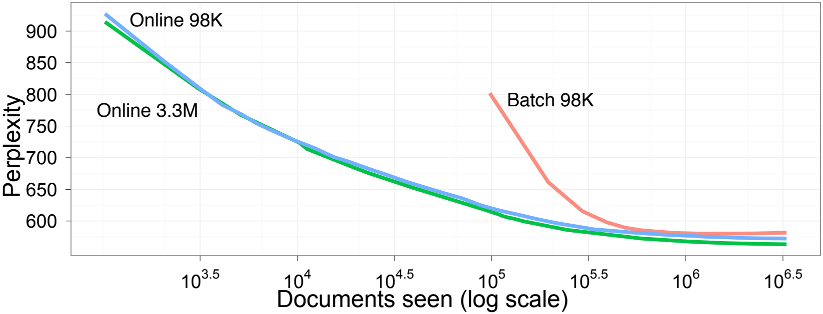


Stochastic variational inference for LDA



- 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**
	systems	systems	service	service	service	business	business	business
	road	health	systems	systems	companies	service	service	industry
	made	communication	health	companies	systems	companies	companies	service
Top eight words	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

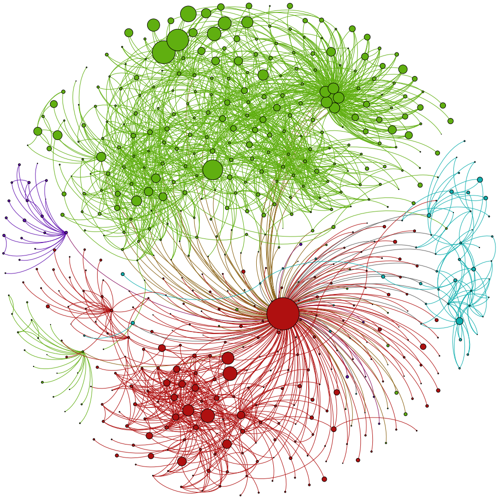
[Hoffman+ 2010]

1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

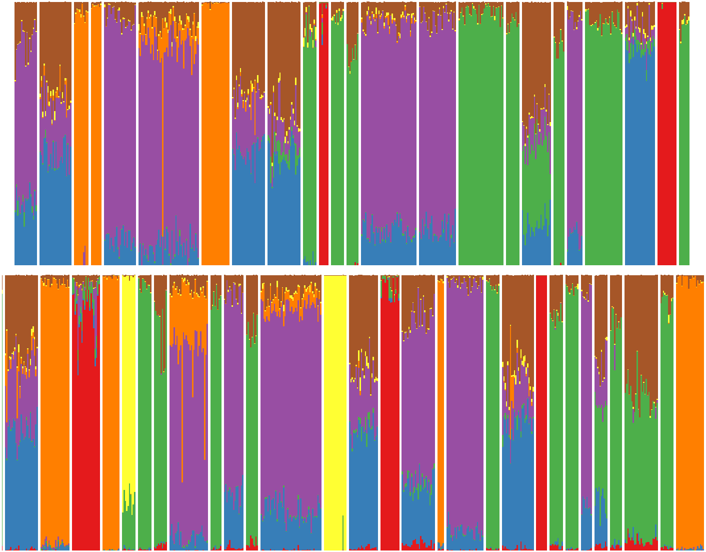
11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

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

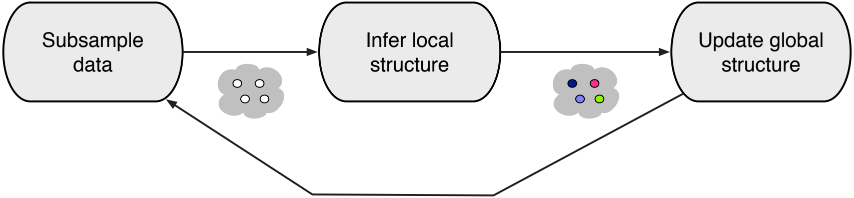
[Gopalan and Blei, PNAS 2013]



Population analysis of 2 billion genetic measurements

[Gopalan+ Nature Genetics 2016]

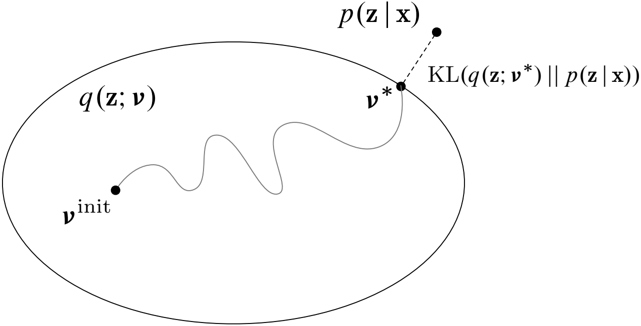
SVI scales many models



- 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)

Black Box Variational Inference

Variational inference



- VI solves **inference** with **optimization**.
- Posit a **variational family** of distributions over the latent variables.
- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.

A.1 Computing $\mathbf{E}[\log(\theta_i \mid \alpha)]$

The need to compute the expected value of the log of a single probability component under the Dirichlet arises repeatedly in deriving the inference and parameter estimation procedures for LDA. This value can be easily computed from the natural parameterization of the exponential family representation of the Dirichlet distribution.

Recall that a distribution is in the exponential family if it can be written in the form:

$$p(x \mid \eta) = h(x) \exp\left\{\eta^T T(x) - A(\eta)\right\},$$

where η is the natural parameter, $T(x)$ is the sufficient statistic, and $A(\eta)$ is the log of the normalization factor.

We can write the Dirichlet in this form by exponentiating the log of Eq. (1):

$$p(\theta \mid \alpha) = \exp\left\{\left(\sum_{i=1}^k (\alpha_i - 1)\log\theta_i\right) + \log\Gamma\left(\sum_{i=1}^k \alpha_i\right) - \sum_{i=1}^k \log\Gamma(\alpha_i)\right\}.$$

From this form, we immediately see that the natural parameter of the Dirichlet is $\eta_i = \alpha_i - 1$ and the sufficient statistic is $T(\theta_i) = \log\theta_i$. Furthermore, using the general fact that the derivative of the log normalization factor with respect to the natural parameter is equal to the expectation of the sufficient statistic, we obtain:

$$\mathrm{E}[\log\theta_i \mid \alpha] = \Psi(\alpha_i) - \Psi\left(\sum_{j=1}^k \alpha_j\right)$$

where Ψ is the digamma function, the first derivative of the log Gamma function.

A.3.2 Variational Dirichlet

Next, we maximize Eq. (15) with respect to γ_i, the ith component of the posterior Dirichlet parameter. The terms containing γ_i are:

$$\begin{aligned} L_{[\gamma]} = &\sum_{i=1}^k (\alpha_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right) + \sum_{n=1}^N \phi_{ni}\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right) \\ &- \log\Gamma\left(\sum_{j=1}^k \gamma_j\right) + \log\Gamma(\gamma_i) - \sum_{i=1}^k (\gamma_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right). \end{aligned}$$

This simplifies to:

$$L_{[\gamma]} = \sum_{i=1}^k \left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right)\left(\alpha_i + \sum_{n=1}^N \phi_{ni} - \gamma_i\right) - \log\Gamma\left(\sum_{j=1}^k \gamma_j\right) + \log\Gamma(\gamma_i).$$

We take the derivative with respect to γ_i:

$$\frac{\partial L}{\partial \gamma_i} = \Psi'(\gamma_i)\left(\alpha_i + \sum_{n=1}^N \phi_{ni} - \gamma_i\right) - \Psi'\left(\sum_{j=1}^k \gamma_j\right)\sum_{j=1}^k \left(\alpha_j + \sum_{n=1}^N \phi_{nj} - \gamma_j\right).$$

Setting this equation to zero yields a maximum at:

$$\gamma_i = \alpha_i + \sum_{n=1}^N \phi_{ni}. \tag{17}$$

Since Eq. (17) depends on the variational multinomial ϕ, full variational inference requires alternating between Eqs. (16) and (17) until the bound converges.

Finally, we expand Eq. (14) in terms of the model parameters (α, β) and the variational parameters (γ, ϕ). Each of the five lines below expands one of the five terms in the bound:

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta) = &\log\Gamma\left(\sum_{j=1}^k \alpha_j\right) - \sum_{i=1}^k \log\Gamma(\alpha_i) + \sum_{i=1}^k (\alpha_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right) \\ &+ \sum_{n=1}^N \sum_{i=1}^k \phi_{ni}\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right) \\ &+ \sum_{n=1}^N \sum_{i=1}^k \sum_{j=1}^V \phi_{ni} w_n^j \log\beta_{ij} \\ &- \log\Gamma\left(\sum_{j=1}^k \gamma_j\right) + \sum_{i=1}^k \log\Gamma(\gamma_i) - \sum_{i=1}^k (\gamma_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right) \\ &- \sum_{n=1}^N \sum_{i=1}^k \phi_{ni}\log\phi_{ni}, \end{aligned} \tag{15}$$

where we have made use of Eq. (8).

In the following two sections, we show how to maximize this lower bound with respect to the variational parameters ϕ and γ.

A.3.1 Variational multinomial

We first maximize Eq. (15) with respect to ϕ_{ni}, the probability that the nth word is generated by latent topic i. Observe that this is a constrained maximization since $\sum_{i=1}^k \phi_{ni} = 1$.

We form the Lagrangian by isolating the terms which contain ϕ_{ni} and adding the appropriate Lagrange multipliers. Let β_{iv} be $p(w_n^v = 1 \mid z^i = 1)$ for the appropriate v. (Recall that each w_n is a vector of size V with exactly one component equal to one; we can select the unique v such that $w_n^v = 1$):

$$L_{[\phi_{ni}]} = \phi_{ni}\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right) + \phi_{ni}\log\beta_{iv} - \phi_{ni}\log\phi_{ni} + \lambda_n\left(\sum_{j=1}^k \phi_{ni} - 1\right),$$

where we have dropped the arguments of L for simplicity, and where the subscript ϕ_{ni} denotes that we have retained only those terms in L that are a function of ϕ_{ni}. Taking derivatives with respect to ϕ_{ni}, we obtain:

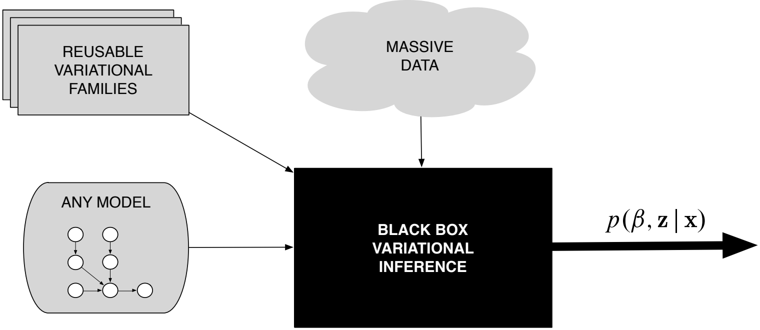
$$\frac{\partial L}{\partial \phi_{ni}} = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right) + \log\beta_{iv} - \log\phi_{ni} - 1 + \lambda.$$

Setting this derivative to zero yields the maximizing value of the variational parameter ϕ_{ni} (cf. Eq. 6):

$$\phi_{ni} \propto \beta_{iv}\exp\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right)\right). \tag{16}$$

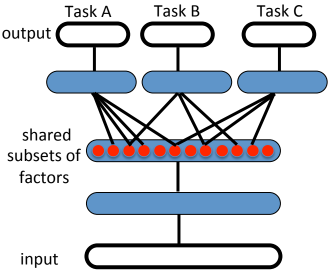
[from Blei+ 2003]

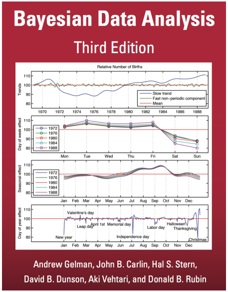
Black box variational inference



- Easily use variational inference with **any model**; no more appendices!
- Perform inference with **massive data**
- **No mathematical work** beyond specifying the model

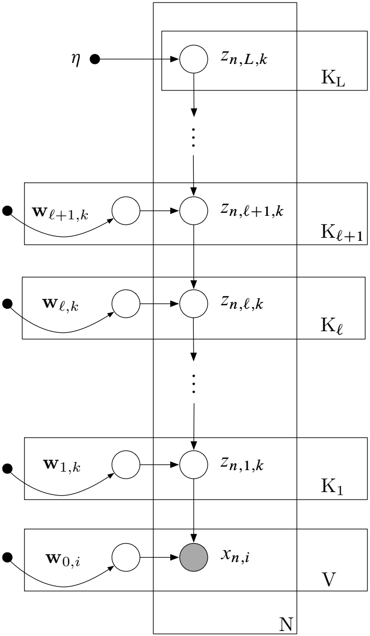
Motivation: Bayesian deep learning





- ***Deep learning*** [Bengio+ 2013]
  Discover layered representations of high-dimensional data.
- ***Bayesian statistics*** [Gelman+ 2014]
  Cast inferences of unknown quantities as probability calculations.
- ***Bayesian deep learning*** [Ranganath+ 2015]
  Posterior inference of layered representations of high-dimensional data.

Model: Deep exponential families

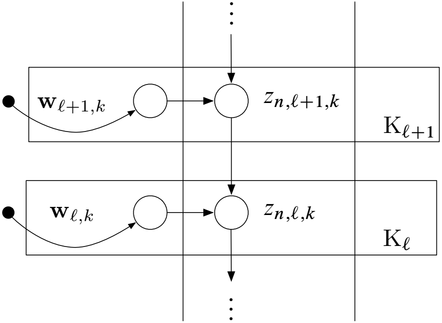


$$z_{n,L,k} \sim \textsc{Exp-Fam}(\eta)$$

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

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

Deep exponential families



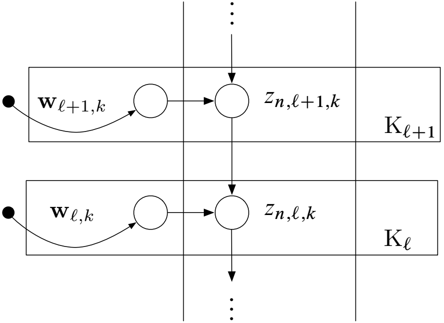
$$z_{n,\ell,k} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{\ell,k}^\top \mathbf{z}_{n,\ell+1}))$$

- All distributions are in an exponential family

$$\begin{aligned} p(z_{n,\ell,k} \mid \mathbf{z}_{n,\ell+1}, \mathbf{w}_{\ell,k}) &= \text{expfam}(z\,;\,\eta) \\ \eta &= g(\mathbf{z}_{n,\ell+1}^\top \mathbf{w}_{\ell,k}). \end{aligned}$$

- Design choices:
  - number of layers; number of units per layer
  - type of representation; link function g

Deep exponential families

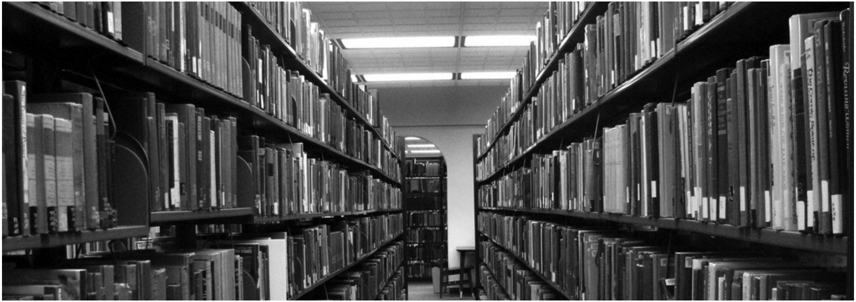


$$z_{n,\ell,k} \sim \text{EXP-FAM}(g(\mathbf{w}_{\ell,k}^{\top}\mathbf{z}_{n,\ell+1}))$$

The hidden layers define the type of latent representation:

- *Bernoulli*, binary [Sigmoid Belief Net, Neal 1992]
- *Gaussian*, real-valued [Deep Latent Gaussian Models, Rezende+ 2014, Kingma+ 2014]
- *Poisson*, count
- *Gamma*, positive (and sparse)

Example: Text data

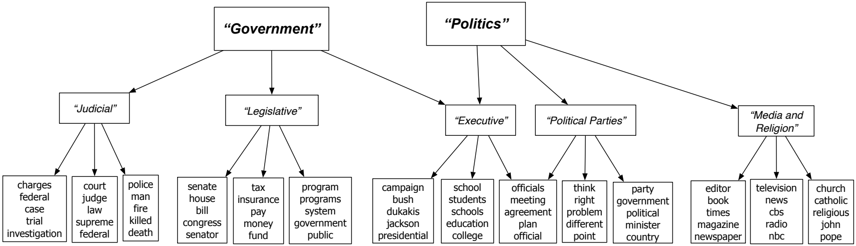


- The data layer defines the type of observations, e.g., $x_{n,i}$ can be the count of word i in document n.
- Use a Poisson likelihood, conditional on the last layer of hidden variables,

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

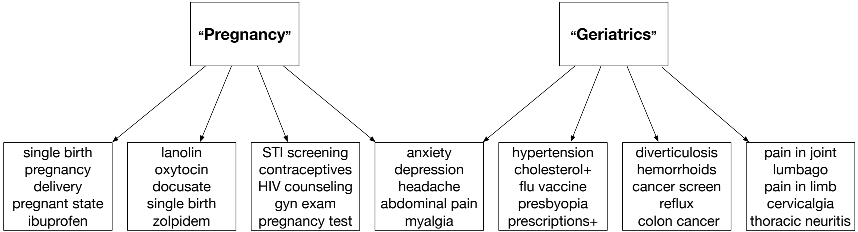
- (This is an alternative to LDA that finds layers of topics.)

New York Times



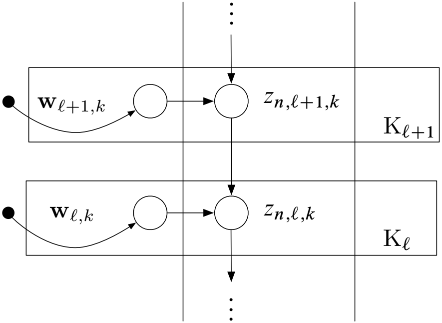
- 160,000 documents; 8,500 vocabulary terms; 10M observed words
- The posterior weights provide topics and topics-of-topics.

Medical Diagnoses



- 300,000 patients; 18,000 diagnoses; 1.5M observed diagnoses
- The posterior weights provide topics and topics-of-topics.

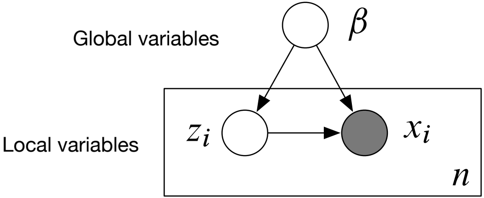
Deep exponential families



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

- We want to try lots of types of DEFs. But how to do inference?
- DEFs contain cascades of latent variables.
- DEFs are ***not conditionally conjugate***.

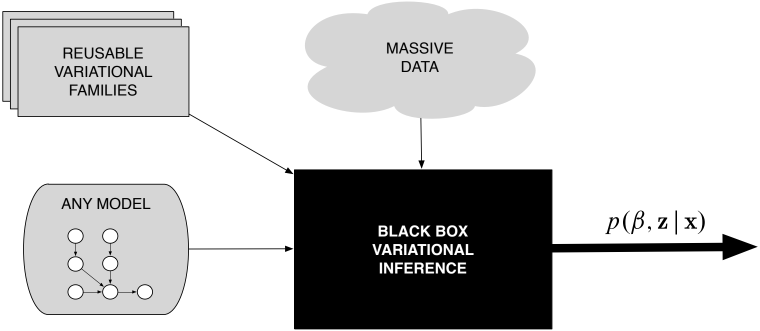
Nonconjugate models



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

- 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

Black box variational inference



- Easily use variational inference with **any model**
- Perform inference with **massive data**
- **No mathematical work** beyond specifying the model

Black box variational inference

$$\mathscr{L}(\boldsymbol{\nu}) = \underbrace{\mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{Expected complete log likelihood}} - \underbrace{\mathbb{E}_q[\log q(\beta, \mathbf{z}; \boldsymbol{\nu})]}_{\text{Negative entropy}}$$

- Recall: 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.

Black box variational inference

$$\mathcal{L}(\nu) = \underbrace{\mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{Expected complete log likelihood}} - \underbrace{\mathbb{E}_q[\log q(\beta, \mathbf{z}; \nu)]}_{\text{Negative entropy}}$$

The main idea behind BBVI:

- Write the **gradient of the ELBO as an expectation**
- Sample from $q(\cdot)$ to form a **Monte Carlo estimate of the gradient**
- Use the MC estimate in a **stochastic optimization**

Black box variational inference

$$\mathscr{L}(\nu) = \underbrace{\mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{Expected complete log likelihood}} - \underbrace{\mathbb{E}_q[\log q(\beta, \mathbf{z}; \nu)]}_{\text{Negative entropy}}$$

Keep in mind the **black box criteria**. We should only need to:

- sample from $q(\beta, \mathbf{z})$
- evaluate things about $q(\beta, \mathbf{z})$
- evaluate $\log p(\beta, \mathbf{z}, \mathbf{x})$

Black box variational inference

$$\mathscr{L}(\nu) = \underbrace{\mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})]}_{\text{Expected complete log likelihood}} - \underbrace{\mathbb{E}_q[\log q(\beta, \mathbf{z}; \nu)]}_{\text{Negative entropy}}$$

- Research in BBVI is about how to write the gradient as an expectation.
- While SVI uses stochastic optimization to overcome large datasets, BBVI uses it to overcome difficult objective functions.
- There are two main strategies:
  - **Score gradients**
  - **Reparameterization gradients**

The score gradient

$$\nabla_\nu \mathscr{L} = \mathbb{E}_{q(\mathbf{z};\nu)}[\underbrace{\nabla_\nu \log q(\mathbf{z};\nu)}_{\text{score function}} \; \underbrace{(\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\nu))}_{\text{instantaneous ELBO}}]$$

- Use the score function to write the gradient as an expectation.
  [Ji+ 2010; Paisley+ 2012; Wingate+ 2013; Ranganath+ 2014; Mnih+ 2014]
- Also called the likelihood ratio or REINFORCE gradient
  [Glynn 1990; Williams 1992]
- Optimizes ν to have high probability on $\mathbf{z}$ that give a big ELBO

The score gradient

$$\nabla_\nu \mathcal{L} = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\underbrace{\nabla_\nu \log q(\mathbf{z};\boldsymbol{\nu})}_{\text{score function}}\ \underbrace{(\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu}))}_{\text{instantaneous ELBO}}]$$

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

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

Score-gradient BBVI

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

Score-gradient BBVI

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

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

Score-gradient BBVI

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

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

while *not converged* **do**

Score-gradient BBVI

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

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

while *not converged* **do**

Take S samples from the variational distribution

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

Score-gradient BBVI

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

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

while *not converged* **do**

Take S samples from the variational distribution

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

Calculate the noisy score gradient

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

Score-gradient BBVI

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

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

while *not converged* **do**

Take S samples from the variational distribution

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

Calculate the noisy score gradient

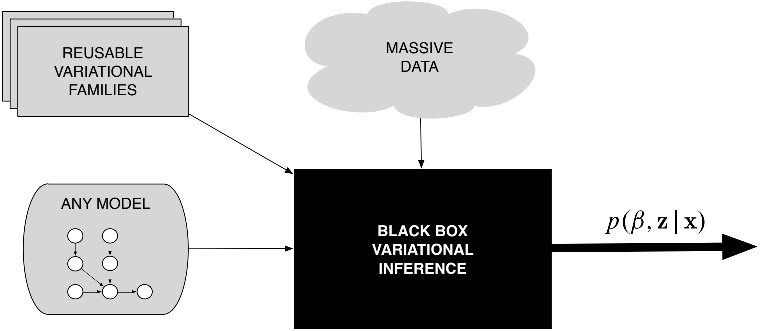
$$\tilde{g}_t = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}[s]; \boldsymbol{\nu}_t)(\log p(\mathbf{x}, \mathbf{z}[s]) - \log q(\mathbf{z}[s]; \boldsymbol{\nu}_t))$$

Update the variational parameters

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \rho_t \tilde{g}_t$$

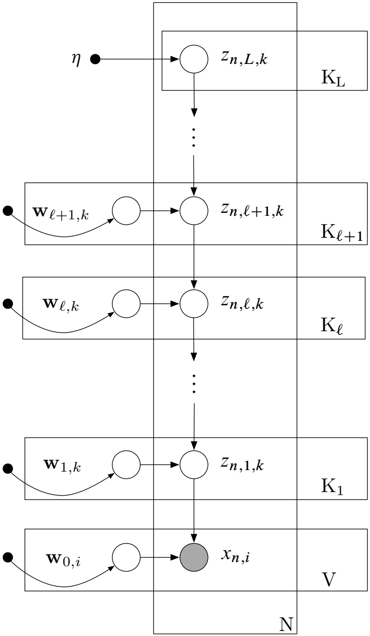
end

BBVI: Making it work



- Control the variance of the gradient [e.g., Paisley+ 2012; Ranganath+ 2014]
  - Rao-Blackwellization, control variates, importance sampling
- Adaptive step sizes [e.g., Duchi+ 2011; Kingma and Ba 2014; Kucukelbir+ 2016]
- SVI, for massive data [Hoffman+ 2013]

Deep exponential families

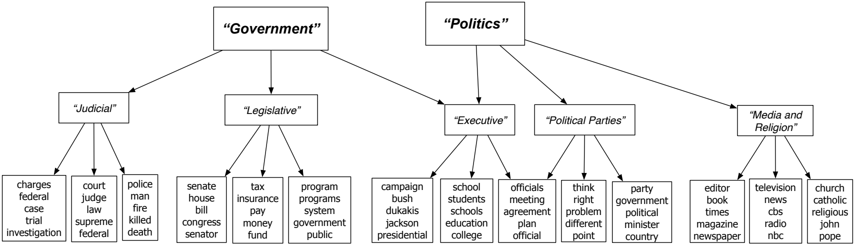


$$z_{n,L,k} \sim \textsc{Exp-Fam}(\eta)$$

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

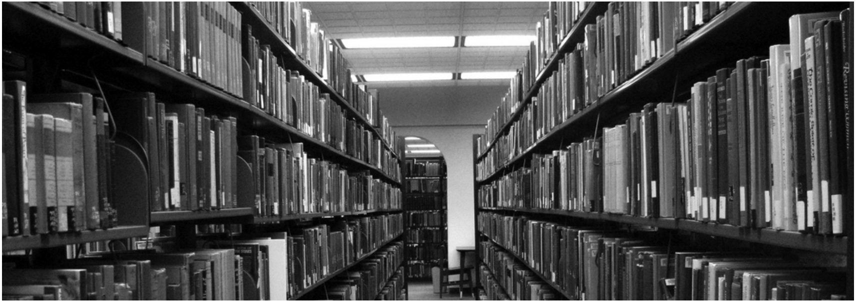
$$x_{n,i} \sim \textsc{Exp-Fam}(g(\mathbf{w}_{0,i}^\top \mathbf{z}_{n,1}))$$

New York Times



- 160,000 documents
- 8,500 vocabulary terms
- 10M observed words

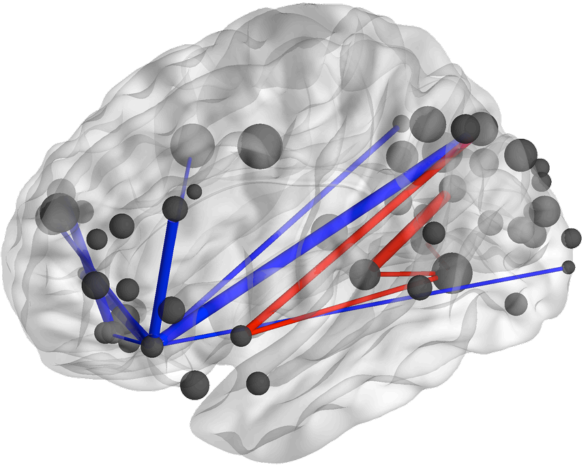
Empirical study of DEFs



- NYT and Science (about 150K documents in each, about 7K terms)
- Adjusted the depth, prior on weights, and link function
- Used BBVI for all analyses
- Held-out perplexity (lower is better) [Wallach+ 2009]

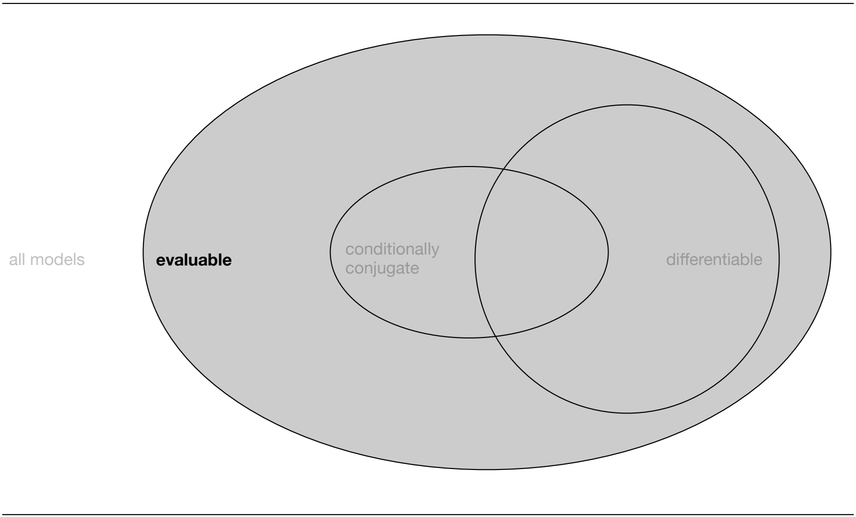
DEF evaluation

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



Neuroscience analysis of 220 million fMRI measurements

[Manning+ 2014]


all models
evaluable
conditionally conjugate
differentiable

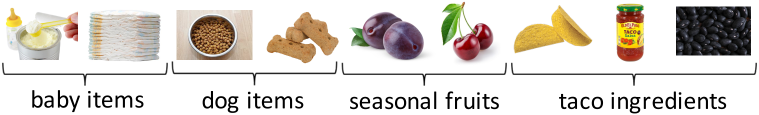
The reparameterization gradient

- One BB criterion is to calculate $\log p(\mathbf{z}, \mathbf{x})$.
  But what if we can also calculate the gradient $\nabla_{\mathbf{z}} \log p(\mathbf{z}, \mathbf{x})$.
- This information might help optimize the ELBO; and it can still be "black box," thanks to automatic differentiation.
- This is the class of models that can use the **reparameterization gradient**.

Shopper: A Bayesian model of consumer choice

- Economists want to understand how people shop
- **Shopper** is a Bayesian model of consumer behavior [Ruiz+ 2017] .
- Use it to understand patterns of purchasing behavior and estimate the effects of interventions (e.g., on price)

Shopper

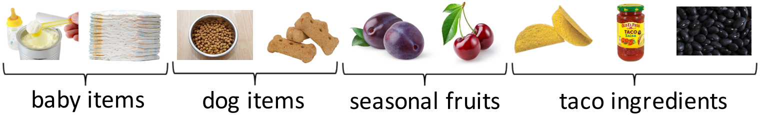


- Each customer walks into the store and sequentially chooses items, each time maximizing utility. This leads to a joint:

$$p(\mathbf{y}_t) = p(y_{t1})p(y_{t2} \mid y_{t1}) \cdots p\left(y_{tn} \mid \mathbf{y}_t^{[n-1]}\right).$$

- The customer picks each item conditional on features of the other items. These features capture that, e.g.,
  - taco shells and beans go well together
  - a customer doesn't need to buy four different types of salsa
  - people who buy dog food also usually buy dog treats
- But these features are latent!

Shopper



- The conditional probability of picking item c is a log linear model

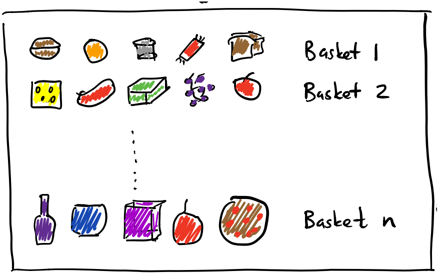
$$p(y_{ti} = c \,|\, \text{previously selected items}) \propto \exp\{\Psi_{tc}\}.$$

- The parameter is

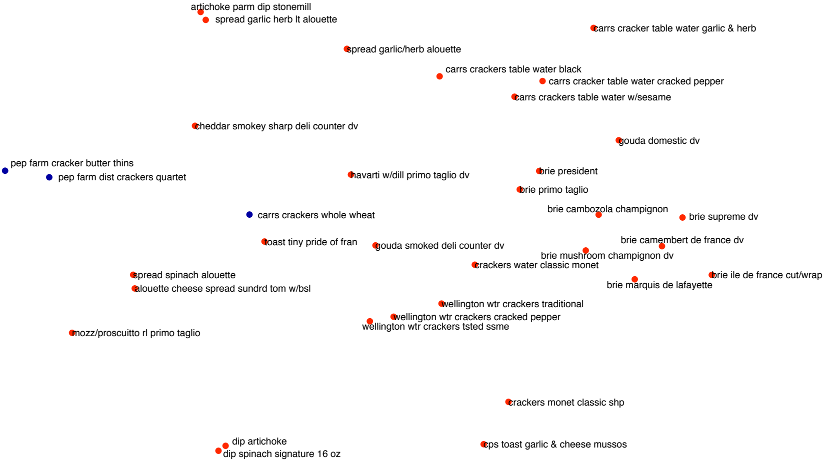
$$\Psi_{tc} = \rho_c^\top \left(\sum_{j=1}^{i-1} \alpha_{y_{tj}} \right)$$

- This is an *embedding method* [Bengio+ 2003; Rudolph+ 2016] .
  - α_{taco} : (latent) attributes of taco shells
  - ρ_{salsa} : attributes that go well with salsa

The Shopper posterior



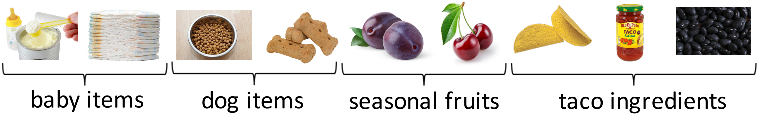
- From a dataset of shopping trips, infer the posterior $p(\boldsymbol{\alpha}, \boldsymbol{\rho} \mid \mathbf{x})$.
- Posterior of per-item attributes and per-item interaction coefficients
- 3,200 customers; 5,600 items; 570K trips; 5.7M purchased items



Shopper on 5.7M purchases.

[Ruiz+ 2017]

Shopper



- Shopper is not conditionally conjugate
- But we can evaluate $\log p(\boldsymbol{\alpha}, \boldsymbol{\rho}, \mathbf{x})$ and its gradient $\nabla_{\boldsymbol{\alpha},\boldsymbol{\rho}} \log p(\boldsymbol{\alpha}, \boldsymbol{\rho}, \mathbf{x})$.
- We can use BBVI with **the reparameterization gradient**.

Differentiable models and transformable variational families

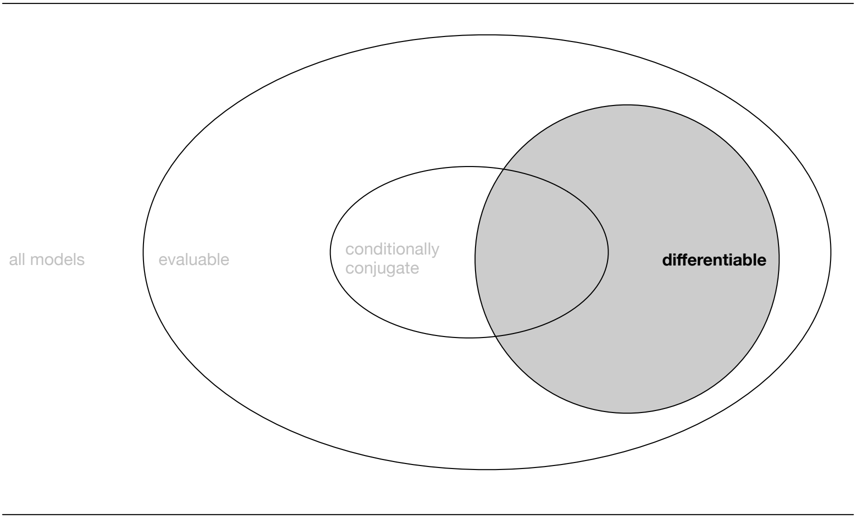
- Suppose $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$ are differentiable with respect to $\mathbf{z}$.
- Suppose the variational distribution can be written with a transformation,

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

  For example,

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

- Note: The variational parameters are part of the transformation, but not the "noise" distribution.


all models
evaluable
conditionally
conjugate
differentiable

The reparameterization gradient

$$\nabla_{\nu} \mathscr{L} = \mathbb{E}_{s(\epsilon)} \left[\underbrace{\nabla_{\mathbf{z}} [\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \nu)]}_{\text{gradient of instananeous ELBO}} \quad \underbrace{\nabla_{\nu} t(\epsilon, \nu)}_{\text{gradient of transformation}} \right]$$

- This is the reparameterization gradient.
  [Glasserman 1991; Fu 2006; Kingma+ 2014; Rezende+ 2014; Titsias+ 2014]
- Can use **autodifferentiation** to take gradients (especially of the model)
- Can use and reuse different transformations [e.g., Naesseth+ 2017]

Reparameterization BBVI

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

Reparameterization BBVI

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

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

Reparameterization BBVI

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

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

while *not converged* **do**

Reparameterization BBVI

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 auxillary variable

$$\boldsymbol{\epsilon}_s \sim s(\epsilon) \quad s = 1 \ldots S$$

Reparameterization BBVI

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 auxillary variable

$$\boldsymbol{\epsilon}_s \sim s(\epsilon) \quad s = 1 \ldots S$$

Calculate the noisy gradient

$$\tilde{g}_t = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\mathbf{z}} [\log p(\mathbf{x}, t(\boldsymbol{\epsilon}_s, \boldsymbol{\nu}_n)) - \log q(t(\boldsymbol{\epsilon}_s, \boldsymbol{\nu}_n); \boldsymbol{\nu}_n)] \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon}_s, \boldsymbol{\nu}_n)$$

Reparameterization BBVI

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 auxillary variable

$$\boldsymbol{\epsilon}_s \sim s(\epsilon) \quad s = 1 \ldots S$$

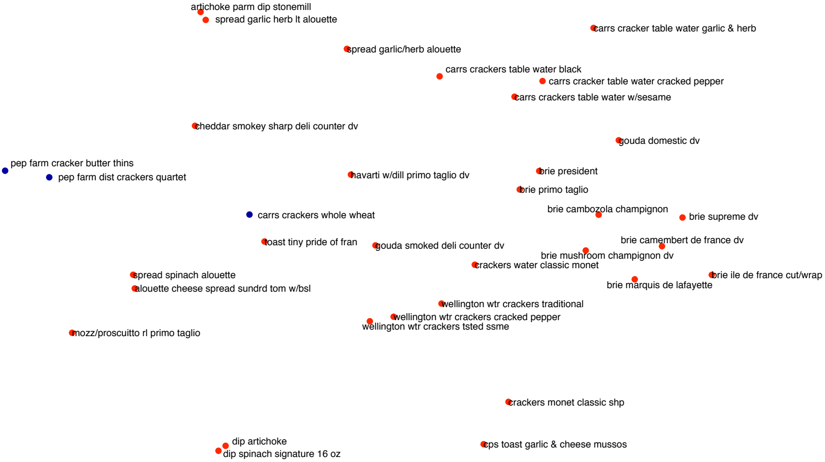
Calculate the noisy gradient

$$\tilde{g}_t = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\mathbf{z}} [\log p(\mathbf{x}, t(\boldsymbol{\epsilon}_s, \boldsymbol{\nu}_n)) - \log q(t(\boldsymbol{\epsilon}_s, \boldsymbol{\nu}_n); \boldsymbol{\nu}_n)] \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon}_s, \boldsymbol{\nu}_n)$$

Update the variational parameters

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \rho_t \tilde{g}_t$$

end



Shopper on 5.7M purchases.

[Ruiz+ 2017]

Analysis of 1.7M taxi trajectories, in Stan

[Kucukelbir+ 2017]

Score gradient

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

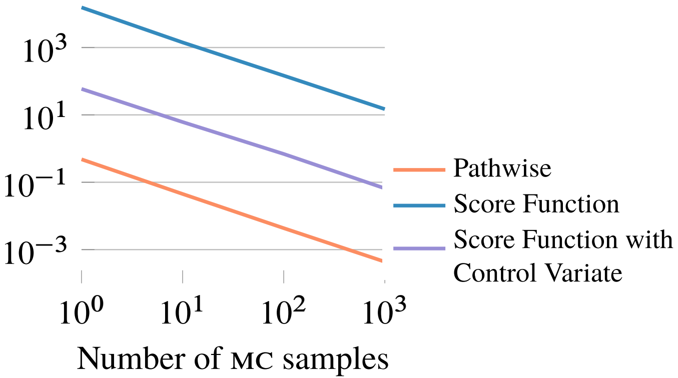
- Works for discrete and continuous models
- Works for a large class of variational approximations
- But the variance of the noisy gradient can be large

Reparameterization gradient

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

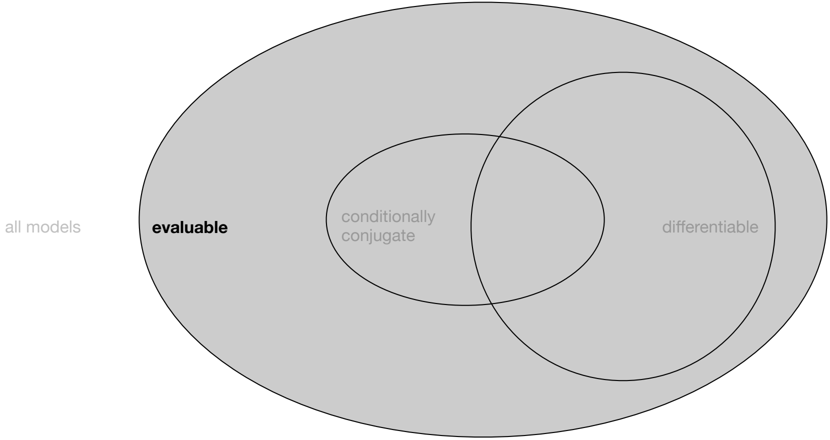
- Requires differentiable models, i.e., no discrete variables
- Requires variational approximation to have form $\mathbf{z} = t(\boldsymbol{\epsilon},\boldsymbol{\nu})$
- Better behaved variance

Variance comparison

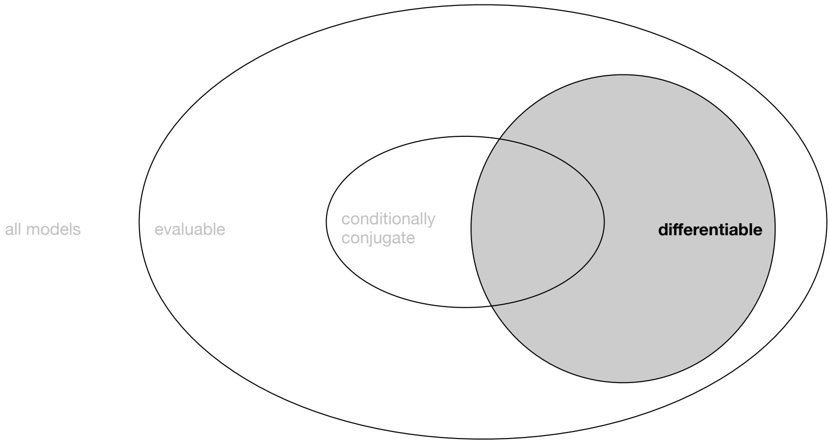


[Kucukelbir+ 2017]

Models that can use the score gradient

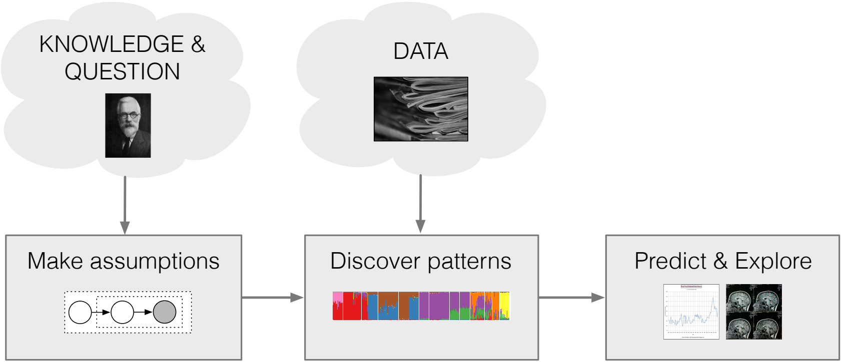


Models that can use the reparameterization gradient



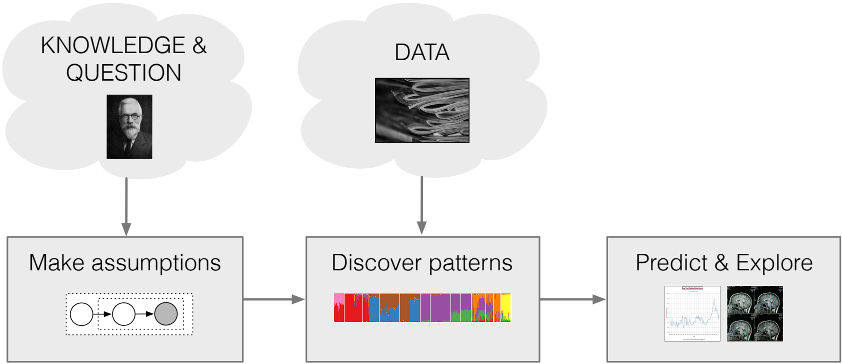
(A pause to look again at the big picture)

The probabilistic pipeline



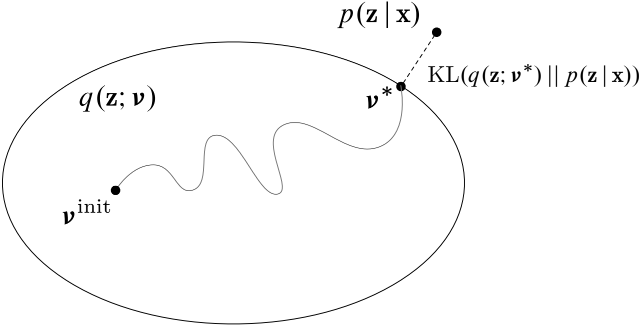
- Box's loop separates the activities involved in probabilistic ML
- We design models for applications, e.g., LDA, DEFs, Shopper.
  But what about computation?
- **Posterior inference** is the key algorithmic problem.

The probabilistic pipeline



- VI provides **scalable** and **general** approaches to posterior inference.
- **Stochastic VI** scales up VI
- **Black box VI** generalizes VI to many models

Stochastic optimization makes VI better



- SVI and BBVI both rely on **stochastic optimization**.
- SVI forms noisy gradients by *subsampling data*.
- BBVI forms noisy gradients with *Monte Carlo* (and can use *automatic differentiation*).

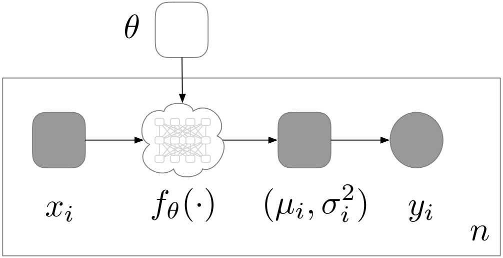
(Now let's continue)

Next topics

- What is a variational autoencoder?
- Let's derive BBVI!
- A tour of variational inference with one picture
- Open questions and references

What is a variational autoencoder?

Deep learning (a probabilistic perspective)

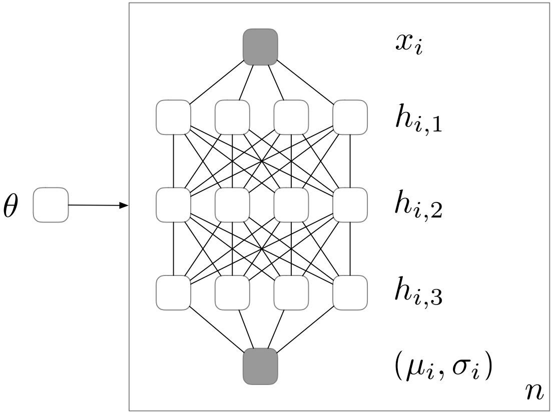


- The features are x_i (e.g., an image); the response is y_i (e.g., a category).
- Deep learning fits a conditional model θ,

$$
\begin{aligned}
(\mu_i, \sigma_i^2) &= f_\theta(x_i) \\
y_i &\sim \mathcal{N}(\mu_i, \sigma_i^2),
\end{aligned}
$$

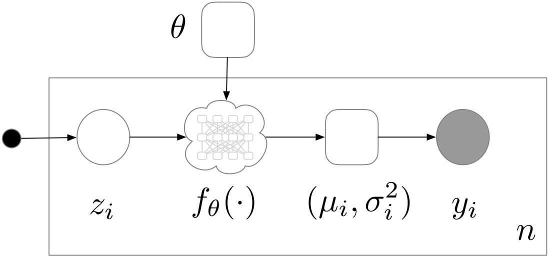
  where $f_\theta(\cdot)$ is a neural network.
- Different distributions of the response lead to different loss functions.

Deep learning (a probabilistic perspective)



- The NN forms the response parameter as a function of the features.
- It is a composition of non-linear functions, where $h_{i,\ell,k} = \sigma(h_{i,\ell-1}^\top \theta_\ell)$.
- The non-linear function can be logit, probit, softplus, etc.

Deep generative models

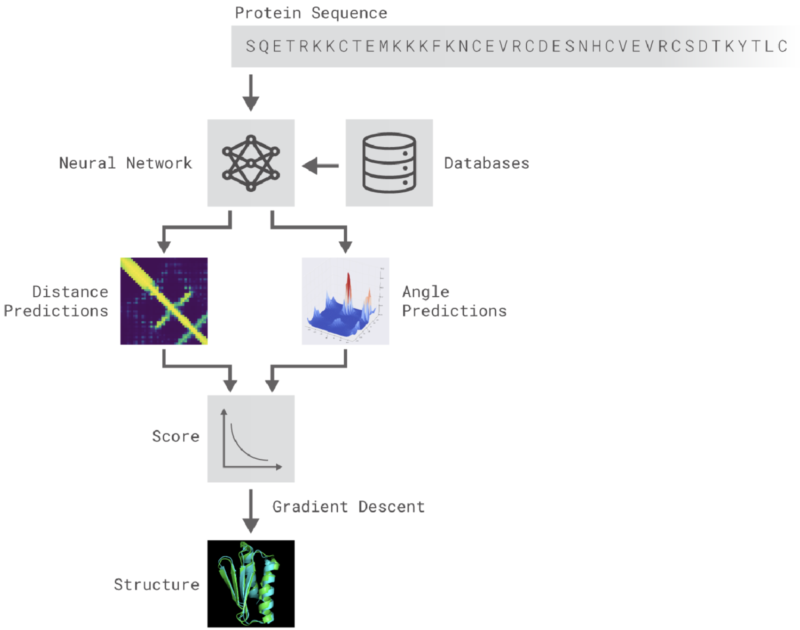


- Deep generative model [Kingma+ 2013; Rezende+ 2014]

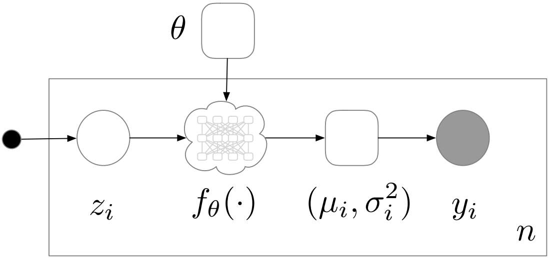
$$\begin{aligned} z_i &\sim \mathcal{N}(0, I) \\ (\mu_i, \sigma_i^2) &= f_\theta(z_i) \\ y_i &\sim \mathcal{N}(\mu_i, \sigma_i^2) \end{aligned}$$

- The input is latent; it is no longer a conditional model.
- The likelihood $p(y \mid z\,;\,\theta)$ is complicated—it involves a NN.

AlphaFold (from DeepMind)



Inference and estimation

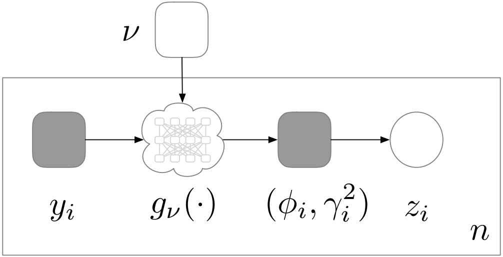


- Inference of z_i and estimation of θ are hard.
- Fix the global parameters θ. The conditional of the local variable z_i is

$$p(z_i \mid y_i, \theta) = \frac{p(z_i)p(y_i \mid z_i, \theta)}{\int p(z_i')p(y_i \mid z_i', \theta)\, \mathrm{d}z_i'}.$$

- The integral in the denominator is intractable; let's use VI.

Amortized inference

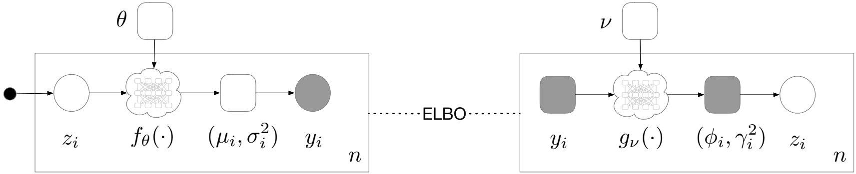


- Use VI at the local level, aiming to fit a $q(z_i)$ that is close to the posterior.
- **Amortization**: the variational family $q(z_i; y_i, \nu)$ is a function of the data point y_i and *shared* variational parameters ν [Gershman and Goodman 2014] .
- Surprise! We use a neural network,

$$(\phi_i, \gamma_i^2) = g_\nu(y_i)$$
$$q(z_i) = \text{normal}(z_i\,;\, \phi_i, \gamma_i^2).$$

It is called the **inference network**.

The variational autoencoder



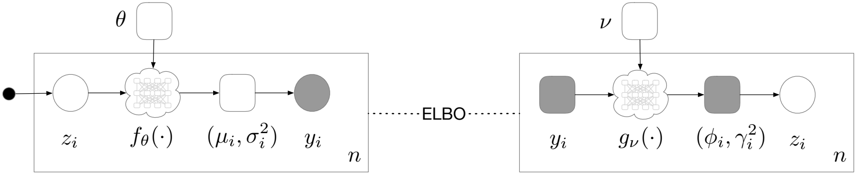
- Amortization ties together the ELBO for each z_i. The objective is

$$\mathscr{L}(\nu) = \sum_{i=1}^{n} \mathbb{E}_{q(z_i; y_i, \nu)} \left[\log p(z_i) + \log p(y_i \,|\, z_i, \theta) - \log q(z_i; y_i, \nu) \right].$$

- The expectation in the ith term uses $q(z_i\,; y_i, \nu)$.

- Amortization is about "learning to infer."
(Open research: there seems to be more to this story.)

The variational autoencoder

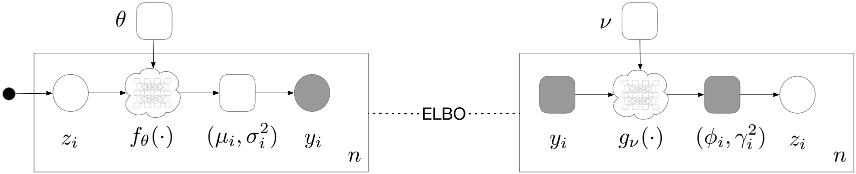


- Use the reparameterization gradient.
- First write z_i down as a transformation,

$$\varepsilon \sim \mathcal{N}(0,1)$$
$$t(\varepsilon, y_i, \nu) = \varepsilon + g_\nu(y_i).$$

- This transformation involves variational parameters ν and datapoint y_i.

The variational autoencoder



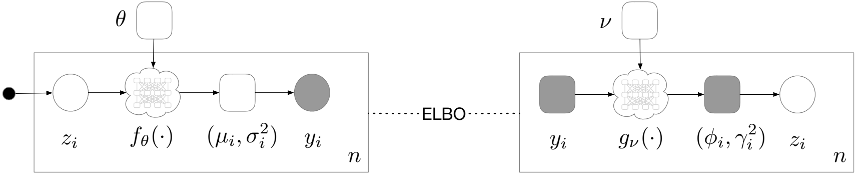
- With the amortized family, the reparameterization gradient is

$$\nabla_\nu \mathscr{L} = \sum_{i=1}^{n} \mathbb{E}_{s(\varepsilon)} \left[\nabla_{z_i} (\log p(z_i) + \log p(y_i \mid z_i, \theta) - \log q(z_i\,; y_i, \nu)) \nabla_\nu t(\varepsilon, \nu, y_i) \right].$$

- We can calculate this gradient with Monte Carlo.

- The gradients involved—of the log likelihood, log variational factor, and transformation—involve standard NN calculations (i.e., backprop) of either the model's NN (θ) or the variational NN (ν).

Fitting the model



- The ELBO is a bound on the log likelihood $\log p(\mathbf{x}; \theta)$.

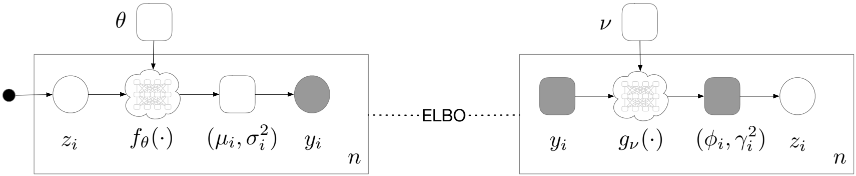
- Fit the model by following its gradient with respect to θ,

$$\nabla_\theta \mathscr{L} = \sum_{i=1}^{n} \mathbb{E}_{s(\varepsilon)} \left[\nabla_\theta \log p(y_i \,|\, z_i, \theta) \right].$$

- Here again

$$z_i = t(\varepsilon, y_i, \nu).$$

So what is a VAE?



- Simultaneously optimize the variational family ν and model θ.

- A VAE
  - samples ε_i for each datapoint and calculates z_i.
  - uses these samples to calculate noisy gradients with respect to ν and θ.
  - follows those gradients in a stochastic optimization

- (VAEs are implemented as a stochastic computational graph of the MC approximation of the ELBO; backprop takes care of the rest.)

The variational autoencoder

Input: data $\mathbf{x}$.

The variational autoencoder

Input: data $\mathbf{x}$.

Initialize $\boldsymbol{\nu}$ randomly; set ρ_t appropriately.

The variational autoencoder

Input: data $\mathbf{x}$.

Initialize $\boldsymbol{\nu}$ randomly; set ρ_t appropriately.

while *not converged* **do**

The variational autoencoder

Input: data $\mathbf{x}$.

Initialize $\boldsymbol{\nu}$ randomly; set ρ_t appropriately.

while *not converged* **do**

for *each datapoint i* **do**

Draw the noise variable and calculate the latent variable

$$\varepsilon_i \sim \mathcal{N}(0, 1); \quad z_i = t(\varepsilon_i, x_i, \nu)$$

end

The variational autoencoder

Input: data $\mathbf{x}$.

Initialize $\boldsymbol{\nu}$ randomly; set ρ_t appropriately.

while *not converged* **do**

for *each datapoint i* **do**

Draw the noise variable and calculate the latent variable

$$\varepsilon_i \sim \mathcal{N}(0,1); \quad z_i = t(\varepsilon_i, x_i, \nu)$$

end

Calculate the noisy gradients

$$\tilde{g}_\nu = \sum_{i=1}^n \nabla_{z_i}(\log p(z_i) + \log p(x_i \mid z_i, \theta_t) - \log q(z_i; x_i, \nu_t))\nabla_\nu t(\varepsilon_i, x_i, \nu_t)$$
$$\tilde{g}_\theta = \sum_{i=1}^n \nabla_\theta \log p(x_i \mid z_i, \theta_t)$$

The variational autoencoder

Input: data $\mathbf{x}$.

Initialize $\boldsymbol{\nu}$ randomly; set ρ_t appropriately.

while *not converged* **do**

for *each datapoint i* **do**

Draw the noise variable and calculate the latent variable

$$\varepsilon_i \sim \mathcal{N}(0,1); \quad z_i = t(\varepsilon_i, x_i, \nu)$$

end

Calculate the noisy gradients

$$\tilde{g}_\nu = \textstyle\sum_{i=1}^n \nabla_{z_i}(\log p(z_i) + \log p(x_i \,|\, z_i, \theta_t) - \log q(z_i; x_i, \nu_t))\nabla_\nu t(\varepsilon_i, x_i, \nu_t)$$
$$\tilde{g}_\theta = \textstyle\sum_{i=1}^n \nabla_\theta \log p(x_i \,|\, z_i, \theta_t)$$

Update the variational parameters and model parameters

$$\nu_{t+1} = \nu_t + \rho_t \tilde{g}_\nu$$
$$\theta_{t+1} = \theta_t + \rho_t \tilde{g}_\theta$$

end

Let’s derive BBVI!

Why do we need black box variational inference?

- Here is a recipe for variational inference
  - Posit a model
  - Choose a variational family
  - Take an integral (i.e., calculate the ELBO)
  - Take derivatives
  - Optimize
- What can go wrong?

A simple failure

- Take the simplest machine learning model, 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

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

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

The problem with nonconjugate models

- Consider just one data point (x,y). Set $y = 1$, so the datapoint is $(x,1)$.
- The 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

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

The problem with nonconjugate models

- Try to calculate the ELBO:

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

The problem with nonconjugate models

- Try to calculate the ELBO:

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

The problem with nonconjugate models

- Try to calculate the ELBO:

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

The problem with nonconjugate models

- Try to calculate the ELBO:

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

The problem with nonconjugate models

- Try to calculate the ELBO:

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

- We are stuck—we cannot analytically take the expectation.

The problem with nonconjugate models

- Try to calculate the ELBO:

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

- We are stuck—we cannot analytically take the expectation.

- Q: Why not take gradients of MC estimates of the ELBO?

  A: It's complicated to take gradients when the samples depend on the variable you are optimizing, here the variational parameters

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 the **black box criteria**.

Let's derive BBVI

- Define the **instantaneous ELBO**

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

  The ELBO is its expectation

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

Let's derive BBVI

- Define the **instantaneous ELBO**

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

  The ELBO is its expectation

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

- We want to calculate $\nabla_{\boldsymbol{\nu}} \mathscr{L}$ **as an expectation**.
  (Then we can use Monte Carlo and stochastic gradients.)

Let's derive BBVI

- Define the **instantaneous ELBO**

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

  The ELBO is its expectation

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

- We want to calculate $\nabla_{\boldsymbol{\nu}} \mathcal{L}$ **as an expectation**.
  (Then we can use Monte Carlo and stochastic gradients.)

- Fact:

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

  You can confirm it in your mind.

Derive the score gradient

- With this fact,

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

Derive the score gradient

- With this fact,

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

Derive the score gradient

- With this fact,

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

Derive the score gradient

- With this fact,

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

Derive the score gradient

- With this fact,

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

- **We have written the gradient as an expectation.**

- The second term vanishes,

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

- The second term vanishes,

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

- What's left is the score gradient,

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

- The second term vanishes,

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

- What's left is the score gradient,

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

- Aside: Why is the expectation of the score function equal to zero?

$$\begin{aligned}
\mathbb{E}_q[\nabla_\nu \log q(\mathbf{z}; \boldsymbol{\nu})] &= \int q(\mathbf{z}; \boldsymbol{\nu}) \nabla_\nu \log q(\mathbf{z}; \boldsymbol{\nu}) \mathrm{d}\mathbf{z} \\
&= \int \nabla_\nu q(\mathbf{z}; \boldsymbol{\nu}) \mathrm{d}\mathbf{z} \\
&= \nabla_\nu \int q(\mathbf{z}; \boldsymbol{\nu}) \mathrm{d}\mathbf{z} = \nabla_\nu 1 = 0.
\end{aligned}$$

Derive the reparameterization gradient

- Assume $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$ are differentiable with respect to $\mathbf{z}$.
- Also assume that we can express the variational distribution with a transformation, where

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

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

$$
\mathscr{L} = \mathbb{E}_{s(\boldsymbol{\varepsilon})}[g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu})].
$$

- Now take the gradient of the ELBO with respect to $\boldsymbol{\nu}$.

- Now take the gradient of the ELBO with respect to $\boldsymbol{\nu}$.

- The gradient easily goes into the expectation. Then use the chain rule,

$$\nabla_{\boldsymbol{\nu}} \mathscr{L} = \mathbb{E}_{s(\boldsymbol{\varepsilon})}[\nabla_{\mathbf{z}}(\log p(\mathbf{z}, \mathbf{x}) - \log q(\mathbf{z}; \boldsymbol{\nu}))\nabla_{\boldsymbol{\nu}} t(\boldsymbol{\varepsilon}, \boldsymbol{\nu}) - \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu}))].$$

  Here we expanded the instantaneous ELBO and used chain rule for functions of two variables,

$$\frac{\mathrm{d}f(x(t),y(t))}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\mathrm{d}f}{\mathrm{d}y}\frac{\mathrm{d}y}{\mathrm{d}t}.$$

- Now take the gradient of the ELBO with respect to $\boldsymbol{\nu}$.

- The gradient easily goes into the expectation. Then use the chain rule,

$$\nabla_{\nu}\mathscr{L} = \mathbb{E}_{s(\boldsymbol{\varepsilon})}[\nabla_{\mathbf{z}}(\log p(\mathbf{z},\mathbf{x}) - \log q(\mathbf{z};\boldsymbol{\nu}))\nabla_{\nu} t(\boldsymbol{\varepsilon},\boldsymbol{\nu}) - \nabla_{\nu}\log q(\mathbf{z};\boldsymbol{\nu}))].$$

  Here we expanded the instantaneous ELBO and used chain rule for functions of two variables,

$$\frac{\mathrm{d}f(x(t),y(t))}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\mathrm{d}f}{\mathrm{d}y}\frac{\mathrm{d}y}{\mathrm{d}t}.$$

- The second term vanishes as above, $-\mathbb{E}[\nabla_{\nu}\log q(\mathbf{z},\boldsymbol{\nu})] = 0$.

  The first terms provide the reparameterization gradient,

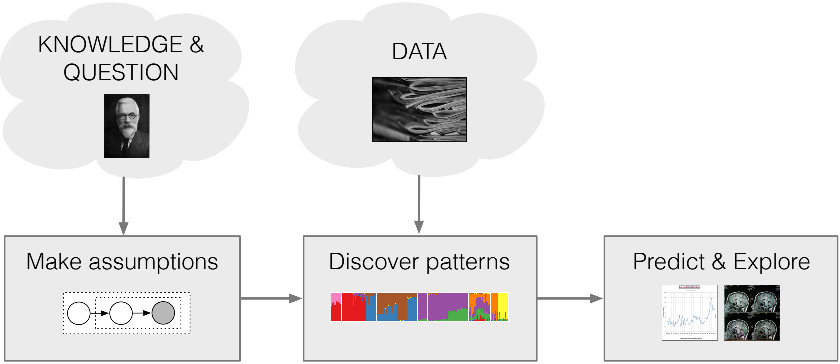
$$\nabla_{\nu}\mathscr{L} = \mathbb{E}_{s(\boldsymbol{\varepsilon})}[\nabla_{\mathbf{z}}(\log p(\mathbf{z},\mathbf{x}) - \log q(\mathbf{z};\boldsymbol{\nu}))\nabla_{\nu} t(\boldsymbol{\varepsilon},\boldsymbol{\nu})].$$

A Tour of Variational Inference (with one picture)

PROBABILISTIC MACHINE LEARNING

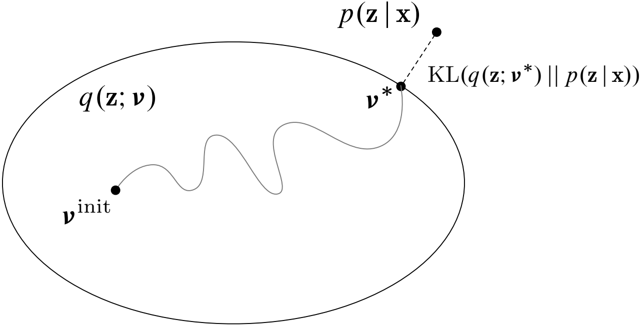
- ML methods that **connect domain knowledge to data**.
- Provides a computational methodology for analyzing data
- Goal: A methodology that is **expressive, scalable, easy to develop**

The probabilistic pipeline



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

Stochastic optimization makes VI better

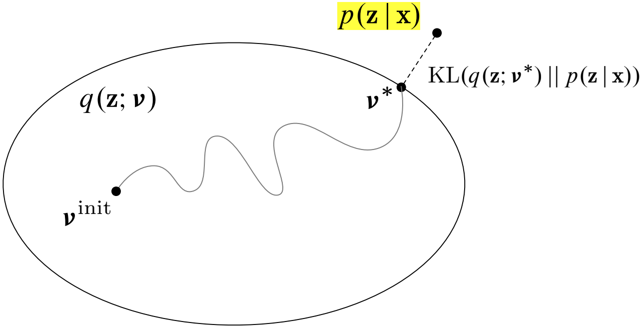


- **Stochastic VI** scales up VI to massive data.
- **Black box VI** generalizes VI to a wide class of models.

What we learned about

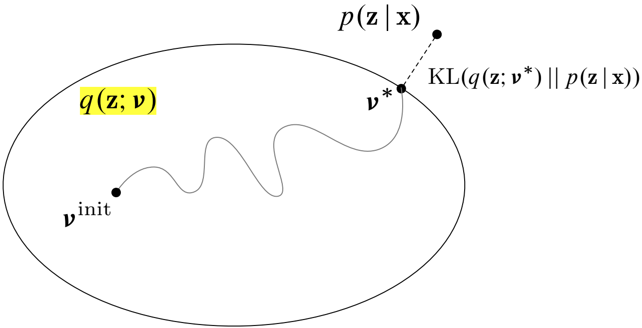
- The basics of variational inference (VI)
  - Mean-field variational inference
  - Coordinate ascent optimization for VI
- Stochastic variational inference for massive data
- Black box variational inference
  - Score gradients
  - Reparameterization gradients
  - Amortized variational families, the variational autoencoder
- Models, along the way
  - Latent Dirichlet allocation and topic models
  - Deep exponential families
  - Embedding models of consumer behavior
  - Deep generative models

The class of models



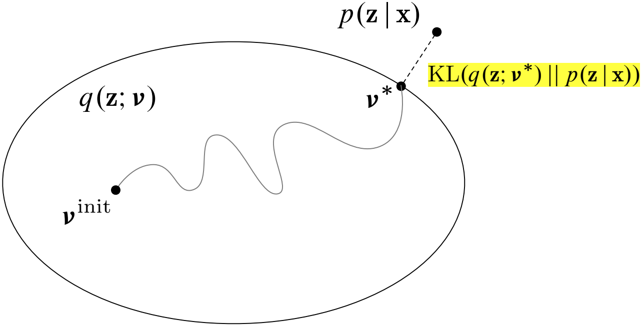
- Conditionally conjugate [Gharamani and Beal 2001; Hoffman+ 2013]
- Not ↑, but can differentiate the log likelihood [Kucukelbir+ 2015]
- Not ↑, but can calculate the log likelihood [Ranganath+ 2014]
- Not ↑, but can sample from the model [Ranganath+ 2017]

The family of variational approximations



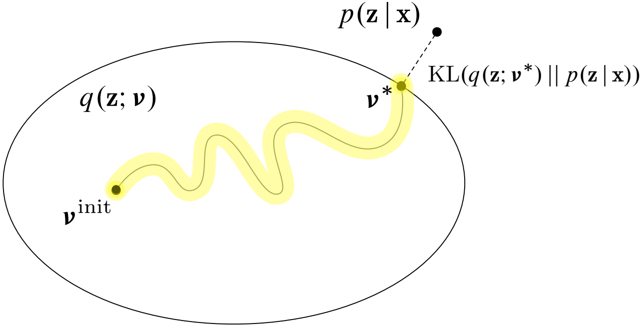
- Structured variational inference [Saul and Jordan 1996; Hoffman and Blei 2015]
- Variational models [Lawrence 2001; Ranganath+ 2015; Tran+ 2015]
- Amortized inference [Kingma and Welling 2014; Rezende+ 2014]
- Sequential Monte Carlo [Naesseth+ 2018; Maddison+ 2017; Le+ 2017]

The distance function



- Expectation propagation [Minka 2001]
- Belief propagation [Yedidia 2001]
- Operator variational inference [Ranganath+ 2016]
- χ-variational inference [Dieng+ 2017]

The algorithm



- SVI and structured SVI [Hoffman+ 2013; Hoffman and Blei 2015]
- Proximity VI [Altosaar+ 2018]
- SGD as VI [Mandt+ 2017]
- Adaptive rates, averaged and biased gradients, etc. [Many papers]

Some open problems in VI

- **Theory**
  MCMC has been widely analyzed and studied; VI is less explored.
- **Optimization**
  Can we find better local optima? Can we accelerate convergence?
- **Alternative divergences**
  KL is chosen for convenience; can we use other divergences?
- **Better approximations**
  VI underestimates posterior variance. Can we do better?

References (from our group)

- D. Blei, A. Kucukelbir, J. McAuliffe. **Variational inference: A review for statisticians**. Journal of American Statistical Association, 2017.

- M. Hoffman, D. Blei, C. Wang, J. Paisley. **Stochastic variational inference**. Journal of Machine Learning Research, 2013.

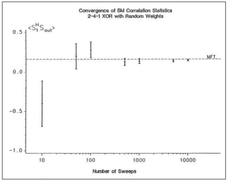
- R. Ranganath, S. Gerrish, D. Blei. **Black box variational inference**. Artificial Intelligence and Statistics, 2014.

- A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, D. Blei. **Automatic differentiation variational inference**. Journal of Machine Learning Research, 2017.

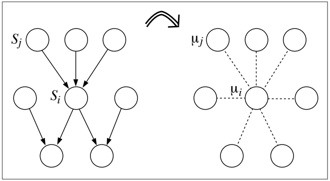
- Y. Wang and D. Blei. **Frequentist consistency of variational Bayes.** Journal of the American Statistical Association, to appear.

Extra slides

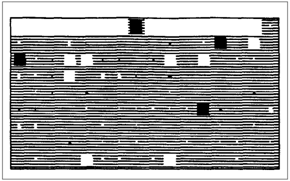
History



[Peterson and Anderson 1987]



[Jordan et al. 1999]

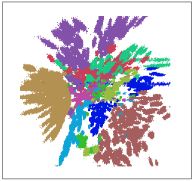


[Hinton and van Camp 1993]

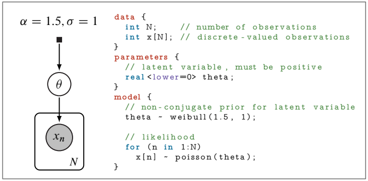
- Variational inference (VI) adapts **ideas from statistical physics** to probabilistic inference. Arguably, it began in the late eighties with Peterson and Anderson (1987), who fit a neural network with mean-field methods.
- This idea was picked up by Jordan's lab in the early 1990s—Tommi Jaakkola,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

Today

[Kingma and Welling 2013]



[Rezende et al. 2014]



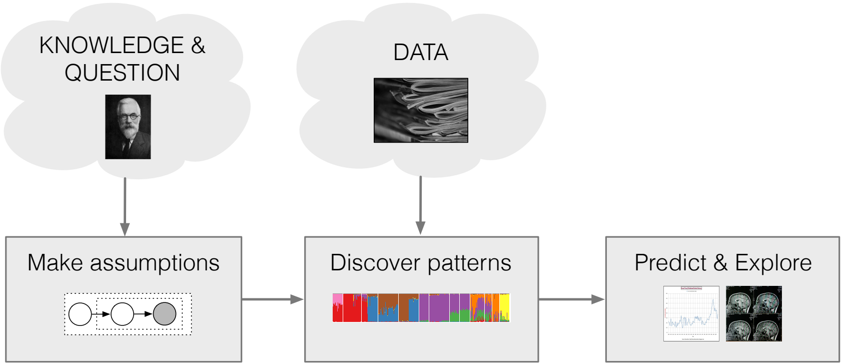
[Kucukelbir et al. 2015]

- There is now a flurry of new work on variational inference, making it **scalable**, **easier to derive**, **faster**, and **more accurate.**

- VI touches many areas: probabilistic programming, reinforcement learning, neural networks, convex optimization, and Bayesian statistics.

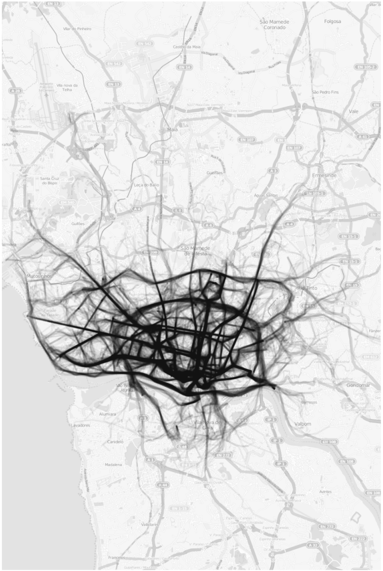
Probabilistic programming and autodifferentiation VI

Probabilistic programming



- Generative models are programs.
  Probabilistic programming takes this idea seriously.
- Languages for expressing models as programs
  Engines to compile models/programs to an inference executable.
- We can do this with BBVI.
  Key ideas: **autodifferentiation** and **stochastic optimization**.

Example: Taxi rides in Portugal



Example: Taxi rides in Portugal

- Data: 1.7M taxi rides in Porto, Portugal
- Multimodal probabilistic PCA with automatic relevance determination

$$\sigma \sim \text{log-normal}(0, 1)$$

$$\begin{aligned} \alpha_j &\sim \text{inv-gamma}(1, 1) && j = 1 \dots k \\ w_{x,j} &\sim \mathcal{N}(0, \sigma \cdot \alpha_j) \\ w_{y,j} &\sim \mathcal{N}(0, \sigma \cdot \alpha_j) \end{aligned}$$

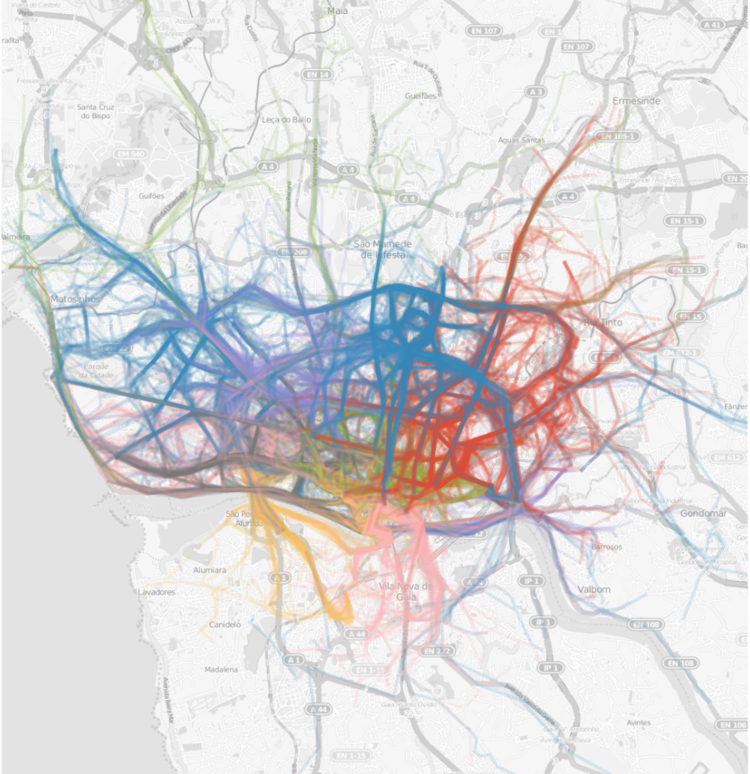
$$\begin{aligned} z_i &\sim \mathcal{N}(0, I) && i = 1 \dots n \\ x_i &\sim \mathcal{N}(w_x^\top z_i, \sigma) \\ y_i &\sim \mathcal{N}(w_y^\top z_i, \sigma) \end{aligned}$$

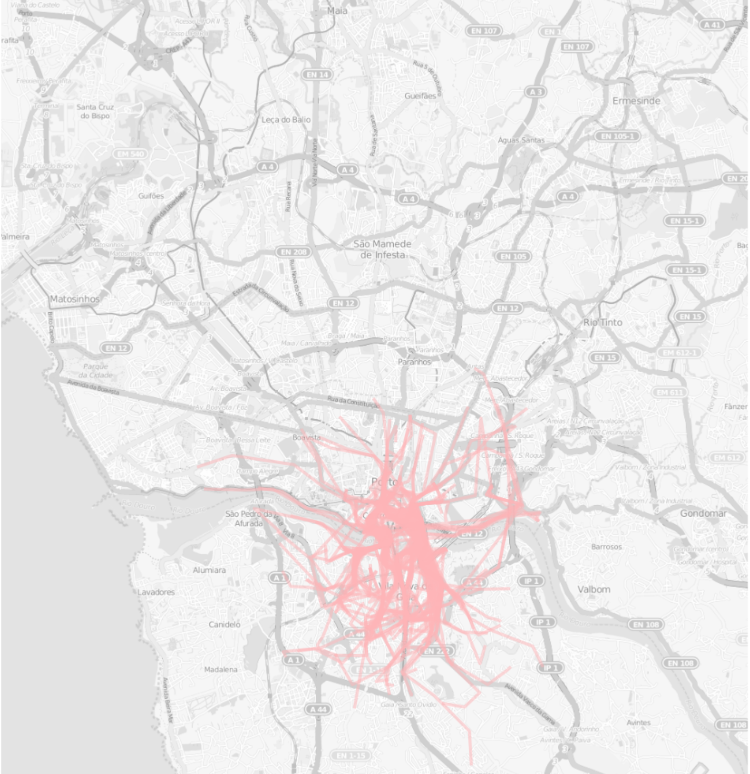
- The generative process looks like a program.

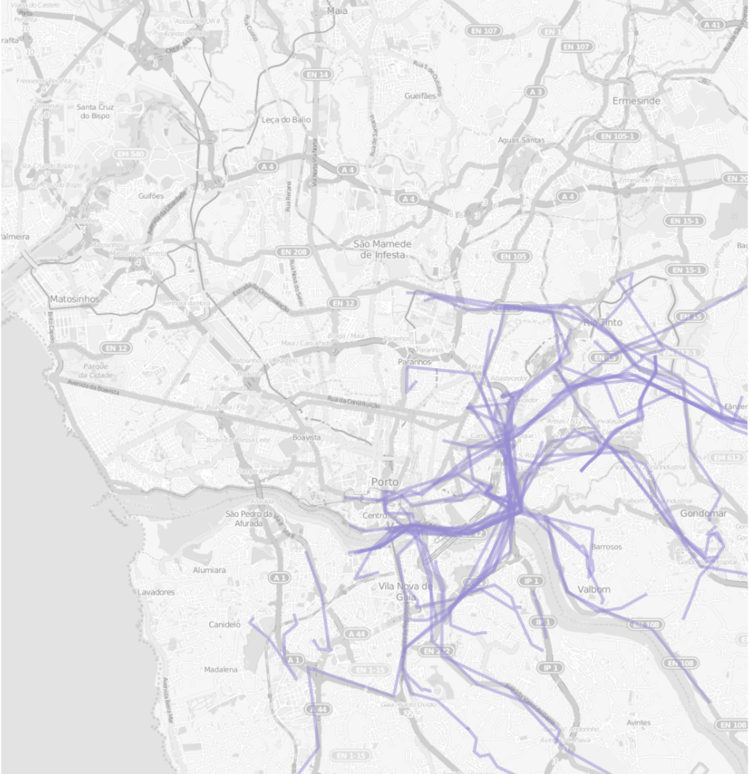
Supervised pPCA with ARD (Stan)

```
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
Ermesinde
Gueifães
Leça do Balio
Santa Cruz do Bispo
Águas Santas
Guifões
São Mamede de Infesta
Matosinhos
Rio Tinto
Parque da Cidade
São Pedro da Afurada
Gondomar
Barrosos
Alumiara
Lavadores
Vila Nova de Gaia
Valbom
Canidelo
Madalena
Avintes
Fânzeres
EN 107
EN 14
A 3
A 41
EN 105-1
A 4
EM 540
EN 208
EN 105
EN 15-1
EN 15
EM 612-1
EM 612
A 1
IP 1
EN 108
A 44
EN 222
EN 1-15


Maia
Ermesinde
Gueifães
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
Fânzeres
EN 107
A 41
EN 14
A 3
EN 105-1
EM 540
A 4
EN 15-1
EN 208
EN 15
EM 612-1
EM 612
A 1
IP 1
EN 108
A 44
EN 222
EN 1-15


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



Maia
Ermesinde
Gueifães
Águas Santas
Santa Cruz do Bispo
Leça do Balio
Guifões
São Mamede de Infesta
Matosinhos
Rio Tinto
Paranhos
Parque da Cidade
Boavista
Porto
Centro
São Pedro da Afurada
Gondomar
Barrosos
Valbom
Vila Nova de Gaia
Alumiara
Lavadores
Canidelo
Madalena
Avintes
EN 107
EN 14
A 3
A 41
EN 105-1
A 4
EN 208
EN 105
EN 15-1
EN 12
EN 15
EN 612
A 1
IP 1
A 44
EN 222
EN 1-15
EN 108
Avenida da Boavista
Rua da Constituição
Estrada da Circunvalação

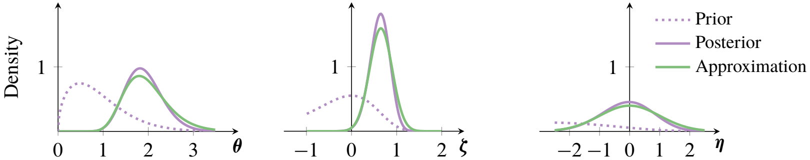
Exploring Taxi Trajectories

- Write down a supervised pPCA model ($\sim$minutes).
- Use VI to fit the model in Stan ($\sim$hours).
- Estimate latent representation z_i of each taxi ride ($\sim$minutes).

- Write down a mixture model ($\sim$minutes).
- Use VI to cluster the latent representations ($\sim$minutes).

What would take weeks $\rightarrow$ a single day.

Automatic differentiation variational inference



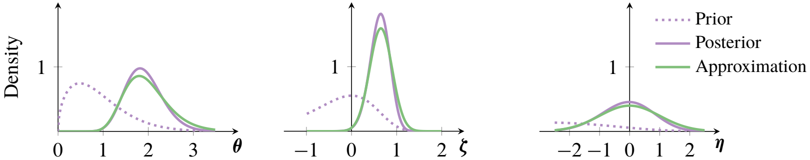
(a) Latent variable space **(b)** Real coordinate space **(c)** Standardized space

1. Transform the model.

- Transform from $p(\mathbf{z}, \mathbf{x})$ to $p(\boldsymbol{\zeta}, \mathbf{x})$, where $\boldsymbol{\zeta} \in \mathbb{R}^d$.
- The mapping is in the joint,

$$p(\zeta, \mathbf{x}) = p\left(\mathbf{x}, s(\zeta)\right) \left| \det J_s(\zeta) \right|.$$

Automatic differentiation variational inference



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

2. Redefine the variational problem.

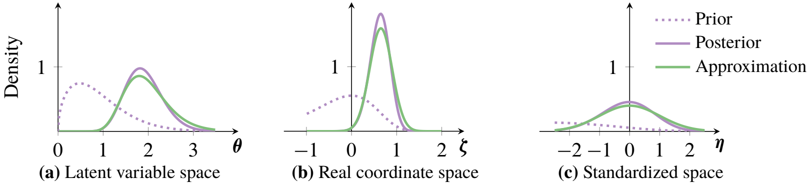
- The variational family is mean-field Gaussian

$$q(\boldsymbol{\zeta}\,;\,\boldsymbol{\nu}) = \prod_{k=1}^{K} \varphi\left(\zeta_k\,;\,\nu_k\right),$$

- The ELBO is

$$\mathscr{L} = \mathbb{E}_{q(\zeta)}\Big[\log p\left(\mathbf{x}, s(\boldsymbol{\zeta})\right) + \log\left|\det J_s(\boldsymbol{\zeta})\right|\Big] + \mathbb{H}(q)$$

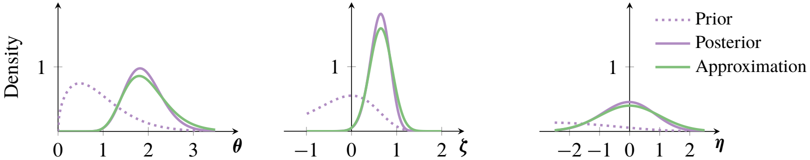
Automatic differentiation variational inference



3. Use the reparameterization gradient

- Transform $\boldsymbol{\zeta}$ using a standard normal $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ to a general normal.
- This is a second transformation of the original latent variable.
- Autodifferentiation handles the reparameterization gradient.

Automatic differentiation variational inference



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

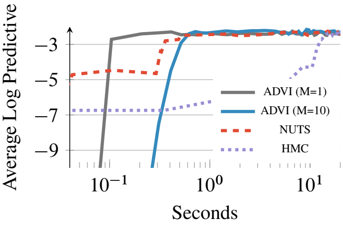
Implementation

- Stan automates going from $\log p(\mathbf{x}, \mathbf{z})$ to

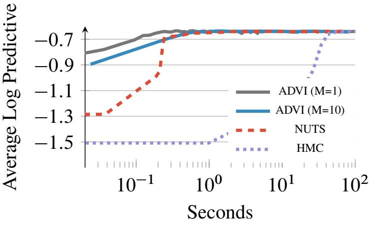
$$\log p(\mathbf{x}, s(\zeta)) + \log \left| \det J_s(\zeta) \right|$$
$$\nabla_{\zeta}(\log p(\mathbf{x}, s(\zeta)) + \log \left| \det J_s(\zeta) \right|)$$

- Use reparameterization BBVI (with the Gaussian transformation)
- Can incorporate SVI and other innovations

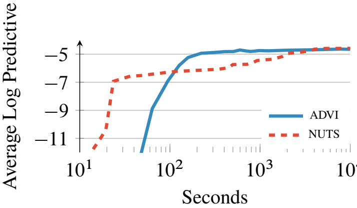
Some benchmarks



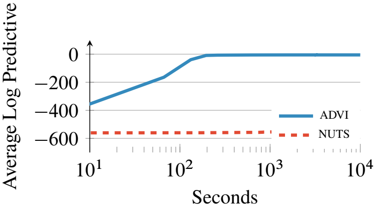
(a) Linear Regression with ARD



(b) Hierarchical Logistic Regression



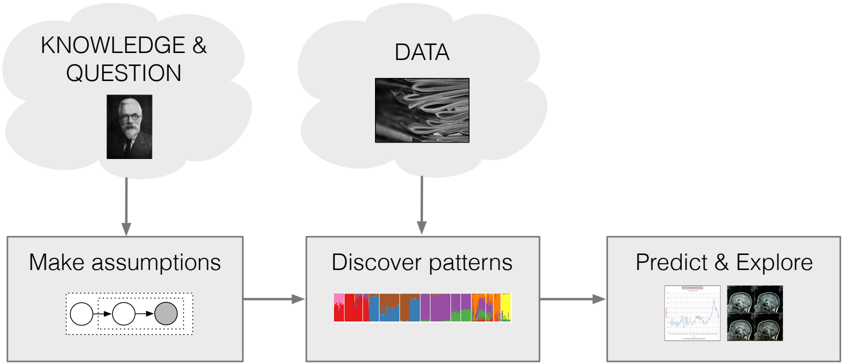
(a) Gamma Poisson Predictive Likelihood



(b) Dirichlet Exponential Predictive Likelihood

[Kucukelbir+ 2016]

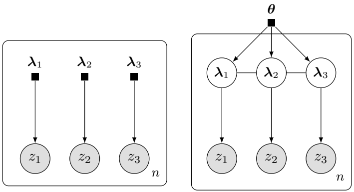
Probabilistic programming



VI is part of several probabilistic programming systems:

- **Edward**: edwardlib.org
- **PyMC3**: github.com/pymc-devs/pymc3
- **Stan**: mcstan.org
- Others: **Anglican**, **Pyro**, ...

Hierarchical variational models [Ranganath+ 2016; Tran+ 2016]



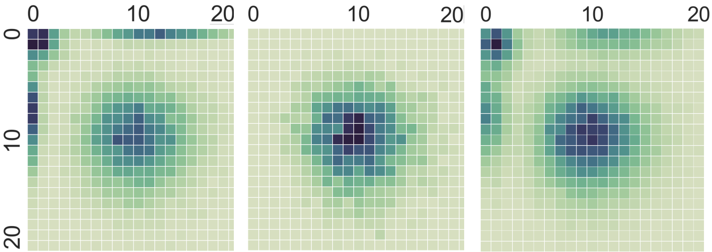
(a) MEAN-FIELD MODEL **(b)** HIERARCHICAL MODEL

- Throughout this talk we assumed the mean-field family
- A **hierarchical variational model**: is

$$q(\mathbf{z}) = \int \left(\prod_i q(z_i \mid \nu_i) \right) q(\boldsymbol{\nu}; \lambda) \, \mathrm{d}\boldsymbol{\nu},$$

- This induces dependence between z_i, e.g.,
  - A mixture over the set of mean-field parameters
  - An amortized distribution of the mean-field parameters

Hierarchical variational models



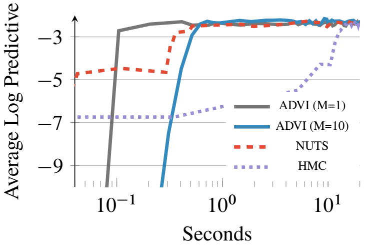
- Takes $q(\mathbf{z})$ beyond the mean field
- Provides high fidelity approximations to the posterior
- Can handle discrete variables (because of the score gradient)

Hierarchical variational models

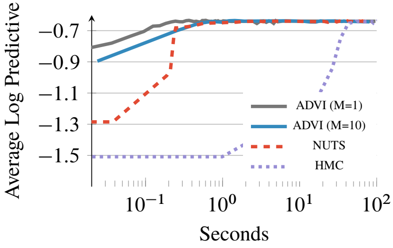
	Model	HVM	Mean-Field
Poisson	100	**3386**	3387
	100-30	**3396**	3896
	100-30-15	**3346**	3962
Bernoulli	100	**3060**	3084
	100-30	3394	**3339**
	100-30-15	**3420**	3575

- Takes $q(\mathbf{z})$ beyond the mean field
- Provides high fidelity approximations to the posterior
- Can handle discrete variables (because of the score gradient)

Should I be skeptical about variational inference?



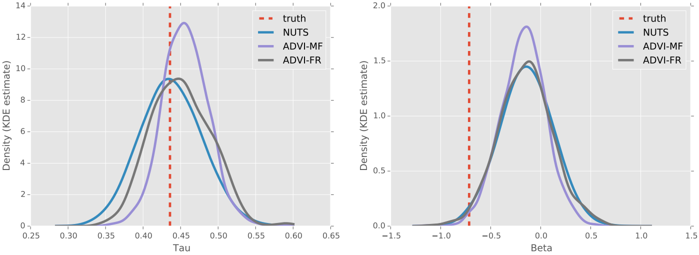
(a) Linear Regression with ARD



(b) Hierarchical Logistic Regression

- **MCMC enjoys theoretical guarantees.**
- But they usually get to the same place. [Kucukelbir+ 2016]
- We need more theory about variational inference.

Should I be skeptical about variational inference?



- **Variational inference underestimates the variance of the posterior.**
- Relaxing the mean-field assumption can help.
- Here: A Poisson GLM [Giordano+ 2015]