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
- unstructured (e.g. text)
- multimodal and interconnected (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
PROBABLISTIC MACHINE LEARNING

- ML methods that connect domain knowledge to data.
- Provides a computational methodology for scalable modeling.
- 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]
<table>
<thead>
<tr>
<th>1</th>
<th>Game</th>
<th>Season</th>
<th>Team</th>
<th>Coach</th>
<th>Play</th>
<th>Points</th>
<th>Games</th>
<th>Giants</th>
<th>Second</th>
<th>Players</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Life</td>
<td>Know</td>
<td>School</td>
<td>Street</td>
<td>Man</td>
<td>Family</td>
<td>Says</td>
<td>House</td>
<td>Children</td>
<td>Night</td>
</tr>
<tr>
<td>3</td>
<td>Film</td>
<td>Movie</td>
<td>Show</td>
<td>Life</td>
<td>Television</td>
<td>Films</td>
<td>Director</td>
<td>Man</td>
<td>Story</td>
<td>Says</td>
</tr>
<tr>
<td>4</td>
<td>Book</td>
<td>Life</td>
<td>Books</td>
<td>Novel</td>
<td>Story</td>
<td>Man</td>
<td>Author</td>
<td>House</td>
<td>War</td>
<td>Children</td>
</tr>
<tr>
<td>5</td>
<td>Wine</td>
<td>Street</td>
<td>Hotel</td>
<td>House</td>
<td>Room</td>
<td>Night</td>
<td>Place</td>
<td>Restaurant</td>
<td>Park</td>
<td>Garden</td>
</tr>
<tr>
<td>6</td>
<td>Bush</td>
<td>Campaign</td>
<td>Clinton</td>
<td>Republican</td>
<td>House</td>
<td>Party</td>
<td>Democratic</td>
<td>Political</td>
<td>Democrats</td>
<td>Senator</td>
</tr>
<tr>
<td>7</td>
<td>Building</td>
<td>Street</td>
<td>Square</td>
<td>House</td>
<td>Buildings</td>
<td>Development</td>
<td>Space</td>
<td>Percent</td>
<td>Real</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Won</td>
<td>Team</td>
<td>Second</td>
<td>Race</td>
<td>Round</td>
<td>Cup</td>
<td>Open</td>
<td>Game</td>
<td>Play</td>
<td>Win</td>
</tr>
<tr>
<td>9</td>
<td>Yankees</td>
<td>Game</td>
<td>Mets</td>
<td>Season</td>
<td>Run</td>
<td>League</td>
<td>Baseball</td>
<td>Team</td>
<td>Games</td>
<td>Hit</td>
</tr>
<tr>
<td>10</td>
<td>Government</td>
<td>War</td>
<td>Military</td>
<td>Officials</td>
<td>Iraq</td>
<td>Forces</td>
<td>Iraqi</td>
<td>Army</td>
<td>Troops</td>
<td>Soldiers</td>
</tr>
<tr>
<td>11</td>
<td>Children</td>
<td>School</td>
<td>Women</td>
<td>Family</td>
<td>Parents</td>
<td>Child</td>
<td>Life</td>
<td>Says</td>
<td>Help</td>
<td>Mother</td>
</tr>
<tr>
<td>12</td>
<td>Stock</td>
<td>Percent</td>
<td>Companies</td>
<td>Fund</td>
<td>Market</td>
<td>Bank</td>
<td>Investors</td>
<td>Funds</td>
<td>Financial</td>
<td>Business</td>
</tr>
<tr>
<td>13</td>
<td>Church</td>
<td>War</td>
<td>Women</td>
<td>Life</td>
<td>Black</td>
<td>Political</td>
<td>Catholic</td>
<td>Government</td>
<td>Jewish</td>
<td>Pope</td>
</tr>
<tr>
<td>14</td>
<td>Art</td>
<td>Museum</td>
<td>Show</td>
<td>Gallery</td>
<td>Works</td>
<td>Artists</td>
<td>Street</td>
<td>Artist</td>
<td>Paintings</td>
<td>Exhibition</td>
</tr>
<tr>
<td>15</td>
<td>Police</td>
<td>Yesterday</td>
<td>Man</td>
<td>Officer</td>
<td>Officers</td>
<td>Case</td>
<td>Found</td>
<td>Charged</td>
<td>Street</td>
<td>Shot</td>
</tr>
</tbody>
</table>

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]
Capsule, a probabilistic model for detecting and characterizing important events in large collections of historical communication, is illustrated in Figure 1. This model analyses two million cables from the National Archives and successfully detects real-world events such as the Indonesian invasion of East Timor (Dec. 7, 1975), the Air France hijacking and Israeli rescue operation (June 27–July 4, 1976), and the fall of Saigon (April 30, 1975). It also identifies other significant moments, including the U.S. sharing lunar rocks with other countries (March 21, 1973) and the death of Mao Tse-tung (Sept. 9, 1976).

The intuition behind Capsule is that embassies write cables throughout the year, usually describing typical business such as the visiting of a government official. However, when an important event occurs, it pulls embassies away from their typical business to write cables that discuss what happened and its consequences. Thus, Capsule effectively defines an “event” as a moment in history when embassies deviate from what each usually discusses, and when each embassy deviates in the same way.

Capsule embeds this intuition into a Bayesian model. It uses hidden variables to encode what “typical business” means for each embassy, how to characterize the events of each week, and which cables discuss those events. Given a corpus, the corresponding posterior distribution provides a filter on the cables that isolates important moments in the diplomatic history.

Capsule can be used to explore any corpora with the same underlying structure: text (or other discrete multivariate data) generated over time by known entities. This includes email, consumer behavior, social media posts, and opinion articles.

We present the model in Section 2, providing both a formal model specification and guidance on how to use its posterior to detect and characterize real-world events. In Section 3, we evaluate Capsule and explore its results on a collection of U.S. State Department cables and on simulated data.

Analysis of 2M declassified cables from the State Dept

[Chaney+ EMNLP 2016]
(Fancy) discrete choice analysis of 5.7M purchases

[Ruiz+ 2017]
The probabilistic pipeline

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

- **Posterior inference** is the key algorithmic problem.
- Answers the question: What does this model say about this data?
- Goal: **General** and **scalable** approaches to posterior inference
Figure S2: Population structure inferred from the TGP data set using the TeraStructure algorithm at three values for the number of populations $K$. The visualization of the ✓’s in the Figure shows patterns consistent with the major geographical regions. Some of the clusters identify a specific region (e.g. red for Africa) while others represent admixture between regions (e.g. green for Europeans and Central/South Americans). The presence of clusters that are shared between different regions demonstrates the more continuous nature of the structure. The new cluster from $K=7$ to $K=8$ matches structure differentiating between American groups. For $K=9$, the new cluster is unpopulated.

[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]
Introduction
A probabilistic model is a joint distribution of hidden variables $z$ and observed variables $x$, 

$$p(z, x).$$

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

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

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,
  
  \[ q(z; \nu) \]

- Fit the **variational parameters** \( \nu \) to be close (in KL) to the exact posterior.
  (There are alternative divergences, which connect to algorithms like EP, BP, and others.)
Example: Mixture of Gaussians

[Images by Alp Kucukelbir; Blei+ 2016]
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
  - Probabilistic programming
- Models, along the way
  - Latent Dirichlet allocation and topic models
  - Deep exponential families
  - Embedding models of consumer behavior
  - Deep generative models
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).
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.
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.
Example: Latent Dirichlet Allocation (LDA)

Seeking Life’s Bare (Genetic) Necessities

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

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


SCIENCE • VOL. 272 • 24 MAY 1996

Documents exhibit multiple topics.
Each topic is a distribution over words

Each document is a mixture of corpus-wide topics

Each word is drawn from one of those topics
Example: Latent Dirichlet Allocation (LDA)

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

\[ p(\text{topics, proportions, assignments} | \text{documents}) \]

(Note: millions of documents; billions of latent variables)
LDA as a Graphical Model

- Encodes **assumptions** about data with a factorization of the joint
- Connects assumptions to **algorithms** for computing with data
- Defines the **posterior** (through the joint)
The posterior of the latent variables given the documents is

\[
p(\beta, \theta, z | w) = \frac{p(\beta, \theta, z, w)}{\int_\beta \int_\theta \sum_z p(\beta, \theta, z, w)}.
\]

- We can't compute the denominator, the marginal \(p(w)\).
- We use approximate 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 St. Anderson 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 Arad Cyrus, 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.

SCIENCE • VOL. 272 • 24 MAY 1996
Mean-field variational inference for LDA

human
genome
dna
 genetic
genes
sequence
gene
molecular
sequencing
map
information
genetics
mapping
project
sequences
evolution
 evolutionary
 species
 organisms
 life
 origin
 biology
 groups
 phylogenetic
 living
 diversity
 group
 new
 two
 disease
 host
 bacteria
 diseases
 resistance
 bacterial
 new
 strains
 control
 infectious
 malaria
 parasite
 parasites
 united
 tuberculosis
 computer
 models
 information
 data
 computers
 system
 network
 systems
 model
 parallel
 methods
 networks
 software
 new
 simulations
| 1 | Game  | Life  | Film  | Book  | Wine  |
| 2 | Season| Know  | Movie | Life  | Street |
| 3 | Team  | School | Show  | Books | Hotel  |
| 4 | Coach | Street | Life  | Novel | House  |
| 5 | Play  | Man   | Television | Story | Room   |
| 6 | Points| Family | Films | Man   | Night  |
| 7 | Games | Says  | Director | House | Place  |
| 8 | Giants| House | War    | Author| Restaurant |
| 9 | Second| Children | War  | War   | Park   |
| 10 | Players| Night | Books  | Children | Garden |
| 11 | Bush | Building | Won  | Yankees | Government |
| 12 | Campaign | Street | Team  | Game  | War |
| 13 | Clinton | Square | Second | Mets | Military |
| 14 | Republican | House | Race | Season | Officials |
| 15 | House | Buildings | Round | Run | Iraq |
| 16 | Party | Development | Cup  | League | Forces |
| 17 | Democratic | Space | Open | Baseball | Iraqi |
| 18 | Political | Percent | Game | Team | Army |
| 19 | Democrats | Real | Play | Games | Troops |
| 20 | Senator | Yates | Win  | Hit  | Soldiers |

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

The observations are $x = x_{1:n}$.

The local variables are $z = z_{1:n}$.

The global variables are $\beta$.

The $i$th data point $x_i$ only depends on $z_i$ and $\beta$.

Compute $p(\beta, z | x)$. 

\[
p(\beta, z, x) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i | \beta)
\]
Conditionally conjugate models

\[ p(\beta, z, x) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i | \beta) \]

- A **complete conditional** is the conditional of a latent variable given the observations and other latent variables.

- Assume each complete conditional is in the exponential family,

\[
p(z_i \mid \beta, x_i) = h(z_i) \exp\{\eta_{\ell}(\beta, x_i)^\top z_i - a(\eta_{\ell}(\beta, x_i))\} \\
p(\beta \mid z, x) = h(\beta) \exp\{\eta_{g}(z, x)^\top \beta - a(\eta_{g}(z, x))\}.
\]

(The exponential family include most distributions that we use.)
Aside: The exponential family

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

Terminology:

- \( \eta \) the natural parameter
- \( t(x) \) the sufficient statistics
- \( a(\eta) \) the log normalizer
- \( h(x) \) the underlying measure (not important)
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

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

A complete conditional is the conditional of a latent variable given the observations and other latent variable.

The global parameter comes from conjugacy [Bernardo and Smith, 1994]

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

where \( \alpha \) is a hyperparameter and \( t(\cdot) \) are sufficient statistics for \( [z_i, x_i] \).
Conditionally conjugate models

Global variables
Local variables

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

- 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 conditionally conjugate differentiable evaluable
Minimize KL between $q(\beta, z; \nu)$ and the posterior $p(\beta, z | x)$. 
The evidence lower bound

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

- Expected complete log likelihood
- Negative entropy

- KL is intractable; VI optimizes the **evidence lower bound** (ELBO) instead.
  - It is a lower bound on \( \log p(x) \).
  - 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

\[ \mathcal{L}(\nu) = \mathbb{E}_q \left[ \log p(x | \beta, z) \right] \quad - \quad \text{KL between variational and prior} \]

- KL is intractable; VI optimizes the evidence lower bound (ELBO) instead.
  - It is a lower bound on \( \log p(x) \).
  - 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.
We need to specify the form of $q(\beta, z)$.

The **mean-field family** is fully factorized,

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

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

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

$$ q(\beta; \lambda) = h(\beta) \exp\{\lambda^\top \beta - a(\lambda)\}. $$

(If the complete conditional is Gaussian then so is the variational factor.)
Mean-field variational inference

Optimize the ELBO,

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

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

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

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

[images by Alp Kucukelbir; Blei+ 2016]
The local variables are the per-document variables $\theta_d$ and $z_{d,n}$.

The global variables are the topics $\beta_1, \ldots, \beta_K$.

The variational distribution is

$$ q(\beta, \theta, z) = \prod_{k=1}^{K} q(\beta_k; \lambda_k) \prod_{d=1}^{D} q(\theta_d; \gamma_d) \prod_{n=1}^{N} q(z_{d,n}; \phi_{d,n}) $$
Mean-field variational inference for LDA

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

\[
\gamma^{(t+1)} = \alpha + \sum_{n=1}^{N} \phi_n^{(t)}, \\
\phi_n^{(t+1)} \propto \exp\{\mathbb{E}[\log \theta] + \mathbb{E}[\log \beta_{.,wn}]\}.
\]
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 Sv 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


SCIENCE • VOL. 272 • 24 MAY 1996
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

definitions:
- human
- genome
- dna
- genetic
- genes
- sequence
- gene
- molecular
- sequencing
- map
- information
- genetics
- mapping
- project
- sequences
- evolution
- evolutionary
- species
- organisms
- life
- origin
- biology
- groups
- phylogenetic
- living
- diversity
- group
- new
- two
- common
- disease
- host
- bacteria
- diseases
- resistance
- bacterial
- new
- strains
- control
- infectious
- malaria
- parasite
- parasites
- united
- tuberculosis
- computer
- models
- information
- data
- computers
- system
- network
- systems
- model
- parallel
- methods
- networks
- software
- new
- simulations
Algorithm 1: Coordinate Ascent Variational Inference

**Input:** data $x$, model $p(\beta, z, x)$.

Initialize $\lambda$ 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

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

**end**
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.
Figure S2: Population structure inferred from the TGP data set using the TeraStructure algorithm at three values for the number of populations $K$. The visualization of the $\checkmark$'s in the Figure shows patterns consistent with the major geographical regions. Some of the clusters identify a specific region (e.g. red for Africa) while others represent admixture between regions (e.g. green for Europeans and Central/South Americans). The presence of clusters that are shared between different regions demonstrates the more continuous nature of the structure. The new cluster from $K=7$ to $K=8$ matches structure differentiating between American groups. For $K=9$, the new cluster is unpopulated.
Stochastic optimization

A STOCHASTIC APPROXIMATION METHOD

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 $\alpha$ 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 $\theta$ 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 METHOD¹
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 \( \alpha \) 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 \( \theta \) in probability.

- With noisy gradients, update

\[
\nu_{t+1} = \nu_t + \rho_t \hat{\nabla}_\nu \mathcal{L}(\nu_t)
\]

- Requires unbiased gradients, \( \mathbb{E}[\hat{\nabla}_\nu \mathcal{L}(\nu)] = \nabla_\nu \mathcal{L}(\nu) \)

- Requires the step size sequence \( \rho_t \) follows the Robbins-Monro conditions
Stochastic variational inference

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

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

- Construct a **noisy natural gradient**,\[ j \sim \text{Uniform}(1, \ldots, n)\]

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

- It is **good for stochastic optimization**.
  - Its expectation is the exact gradient (**unbiased**).
  - It only depends on optimized parameters of one data point (**cheap**).
Algorithm 2: Stochastic Variational Inference

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

Initialize $\lambda$ randomly.
Set $\rho_t$ appropriately.

**while not converged do**

- Sample $j \sim \text{Unif}(1, \ldots, n)$.
- Set local parameter
  
  $\phi \leftarrow \mathbb{E}_\lambda [\eta_\ell(\beta, x_j)]$.

- 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

Figure S2: Population structure inferred from the TGP data set using the TeraStructure algorithm at three values for the number of populations $K$. The visualization of the ✓’s in the Figure shows patterns consistent with the major geographical regions. Some of the clusters identify a specific region (e.g. red for Africa) while others represent admixture between regions (e.g. green for Europeans and Central/South Americans). The presence of clusters that are shared between different regions demonstrates the more continuous nature of the structure. The new cluster from $K=7$ to $K=8$ matches structure differentiating between American groups. For $K=9$, the new cluster is unpopulated.
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 seen (log scale)

Perplexity

Documents analyzed

Top eight words

[Hoffman+ 2010]
<table>
<thead>
<tr>
<th>Figure 5</th>
<th>Topics found in a corpus of 1.8 million articles from the New York Times. Modified from Hoffman et al. (2013).</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Game, Season, Team, Coach, Play, Points, Games, Giants, Second, Players</td>
</tr>
<tr>
<td>2</td>
<td>Life, Know, School, Street, Man, Family, Says, House, Children, Night</td>
</tr>
<tr>
<td>3</td>
<td>Film, Movie, Show, Life, Television, Films, Director, Man, Story, Says</td>
</tr>
<tr>
<td>4</td>
<td>Book, Life, Books, Novel, Story, Man, Author, House, War, Children</td>
</tr>
<tr>
<td>5</td>
<td>Wine, Street, Hotel, House, Room, Night, Place, Restaurant, Park, Garden</td>
</tr>
<tr>
<td>6</td>
<td>Bush, Campaign, Clinton, Republican, House, Party, Democratic, Political, Democrats, Senator</td>
</tr>
<tr>
<td>7</td>
<td>Building, Street, Square, Housing, House, Buildings, Development, Space, Percent, Real</td>
</tr>
<tr>
<td>8</td>
<td>Won, Team, Second, Race, Round, Cup, Open, Game, Play, Win</td>
</tr>
<tr>
<td>9</td>
<td>Yankees, Game, Mets, Season, Run, League, Baseball, Team, Games, Hit</td>
</tr>
<tr>
<td>10</td>
<td>Government, War, Military, Officials, Iraq, Forces, Iraqi, Army, Troops, Soldiers</td>
</tr>
<tr>
<td>11</td>
<td>Children, School, Women, Family, Parents, Child, Life, Says, Help, Mother</td>
</tr>
<tr>
<td>12</td>
<td>Stock, Percent, Companies, Fund, Market, Bank, Investors, Funds, Financial, Business</td>
</tr>
<tr>
<td>13</td>
<td>Church, War, Women, Life, Black, Political, Catholic, Government, Jewish, Pope</td>
</tr>
<tr>
<td>14</td>
<td>Art, Museum, Show, Gallery, Works, Artists, Street, Artist, Paintings, Exhibition</td>
</tr>
<tr>
<td>15</td>
<td>Police, Yesterday, Man, Officer, Officers, Case, Found, Charged, Street, Shot</td>
</tr>
</tbody>
</table>

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)
Variational inference

- VI solves inference with optimization.
- Posit a variational family of distributions over the latent variables.
- Fit the variational parameters $\nu$ to be close (in KL) to the exact posterior.
A.1 Computing $E[\log(\theta_i | \alpha)]$

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:

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

where $\eta$ 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):

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

From this form, we immediately see that the natural parameter of the Dirichlet is $\eta = \alpha_i - 1$ and the sufficient statistic is $T(\eta) = \log(\theta_i)$. 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:

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

where $\Psi$ 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 $\gamma_i$, the $i$th component of the posterior Dirichlet parameter. The terms containing $\gamma_i$ are:

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

$$- \log \Gamma(\sum_{j=1}^{k} \gamma_j) + \log \Gamma(\gamma_i) - \sum_{j=1}^{k} (\gamma_j - 1) (\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)).$$

This simplifies to:

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

We take the derivative with respect to $\gamma_i$:

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

Setting this equation to zero yields a maximum at:

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

Since Eq. (17) depends on the variational multinomial $\phi$, 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 ($\alpha, \beta$) and the variational parameters ($\gamma, \phi$). Each of the five lines below expands one of the five terms in the bound:

$$L(\gamma, \phi; \alpha, \beta) = \log \Gamma(\sum_{j=1}^{k} \alpha_j) - \sum_{i=1}^{k} \log \Gamma(\alpha_i) + \sum_{i=1}^{k} (\alpha_i - 1) (\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j))$$

$$+ \sum_{i=1}^{k} \sum_{n=1}^{N} \phi_{ni} (\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j))$$

$$+ \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{n=1}^{N} \phi_{ni} \phi_{nj} \log \beta_{ij}$$

$$- \log \Gamma(\sum_{j=1}^{k} \gamma_j) + \sum_{i=1}^{k} \log \Gamma(\gamma_i) - \sum_{i=1}^{k} (\gamma_i - 1) (\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j))$$

$$- \sum_{i=1}^{N} \sum_{j=1}^{N} \phi_{ni} \log \phi_{nj},$$

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 $\phi$ and $\gamma$.

A.3.3 Variational Multinomial

We first maximize Eq. (15) with respect to $\phi_{ni}$, the probability that the $n$th 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 $\phi_{ni}$ and adding the appropriate Lagrange multipliers. Let $p_{ni} = \frac{\gamma_i}{\gamma_i' + \sum_{j=1}^{k} \gamma_j'}$ for the appropriate $v$. (Recall that each $w_{ni}$ is a vector of size $V$ with exactly one component equal to one; we can select the unique $v$ such that $w_{ni}^{v} = 1$):

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

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

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

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

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

[from Blei+ 2003]
Black box variational inference

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

evaluable
conditionally conjugate
differentiable
Nonconjugate models

Global variables

Local variables

\[
p(\beta, z, x) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i | \beta)
\]

- 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
Deep exponential families (a digression)

- **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.
Deep exponential families

\[ z_{n,L,k} \sim \text{EXP-FAM}(\eta) \]

\[ z_{n,\ell+1,k} \sim \text{EXP-FAM}(g(w_{\ell,k}^T z_{n,\ell+1})) \]

\[ z_{n,\ell,k} \sim \text{EXP-FAM}(g(w_{\ell,k}^T z_{n,\ell+1})) \]

\[ x_{n,i} \sim \text{EXP-FAM}(g(w_{0,i}^T z_{n,1})) \]
Deep exponential families

\[ z_{n,\ell+1, k} \sim \text{Exp-Fam}(g(w_{\ell, k}^T z_{n,\ell+1})) \]

- All distributions are in canonical exponential family form

\[ p(z_{n,\ell, k} | z_{n,\ell+1}, w_{\ell, k}) = \exp\{\eta(\cdot)^\top t(z_{n,\ell, k}) - a(\eta(\cdot))\}. \]

- The natural parameter uses a link function, \( \eta(\cdot) = g(z_{n,\ell+1}^T w_{\ell, k}) \).

- DEF design choices:
  - number of layers; number of units per layer
  - type of representation; link function
Deep exponential families

The hidden layers:

- **Bernoulli**, for binary representations [Sigmoid Belief Net, Neal 1992]
- **Poisson**, for count representations
- **Gaussian**, for real representations
- **Gamma**, for positive representations
Example: Text data

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

$$x_{n,i} \sim \text{Poisson}(g(w_{0,i}^T z_{n,1}))$$

- This is an alternative to LDA that finds layers of topics
- 160,000 documents
- 8,500 vocabulary terms
- 10M observed words
Medical Diagnoses

- 300,000 patients
- 18,000 diagnoses and medicines
- 1.5M observed diagnoses
Deep exponential families

- DEFs can be composed in more complex models
  - Text analysis
  - Collaborative filtering ("double DEFs")
  - Survival analysis

- Open source software:
  
  https://github.com/blei-lab/deep-exponential-families
Deep exponential families

- How to do inference?
- DEFs are \textit{not} in the class of conditionally conjugate models.
- The complete conditionals do not have a closed form.
Goal: Try out many DEFs on a dataset

- Explore types of representations, link functions, number of layers

Solution: black box variational inference (BBVI)
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

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

- BBVI with the score function estimator
- BBVI with the reparameterization gradient
- Probabilistic programming and autodifferentiation VI
- How to derive BBVI
Black box variational inference

- BBVI with the score gradient
- BBVI with the reparameterization gradient
- Probabilistic programming and autodifferentiation VI
- How to derive BBVI
BBVI with the score gradient

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

- Write the gradient as an expectation.
- Use Monte Carlo to form stochastic gradients.
- Sometimes called the score, likelihood ratio, or REINFORCE gradient
  [Glynn 1990; Williams 1992; Wingate+ 2013; Ranganath+ 2014; Mnih+ 2014]
Noisy unbiased gradients

- Construct noisy unbiased gradients with Monte Carlo,

\[
\hat{\nabla}_\nu = \frac{1}{S} \sum_{s=1}^{S} \nabla_\nu \log q(z_s; \nu)(\log p(x, z_s) - \log q(z_s; \nu)),
\]

where \(z_s \sim q(z; \nu)\)

- To compute a noisy gradient of the ELBO,
  - sample from \(q(z)\)
  - evaluate \(\nabla_\nu \log q(z; \nu)\)
  - evaluate \(\log p(x, z)\) and \(\log q(z)\)

- Satisfies the black box criteria — no model-specific is analysis needed.
Algorithm 3: Basic Black Box Variational Inference

**Input:** data \( x \), model \( p(z, x) \).

Initialize \( \nu \) randomly.
Set \( \rho_j \) appropriately.

**while** not converged **do**

Take \( S \) samples from the variational distribution

\[
z[s] \sim q(z; \nu) \quad s = 1 \ldots S
\]

Calculate the noisy score gradient

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

Update the variational parameters

\[
\nu_{t+1} = \nu_t + \rho_t \tilde{g}_t
\]

**end**
Black box variational inference

- Control the variance of the gradient
  - Rao-Blackwellization, control variates, importance sampling, ...
- Adaptive learning rates [Duchi+ 2011; Tieleman and Hinton 2012]
- Stochastic variational inference, for handling massive data
- 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)
- Held-out perplexity (lower is better) [Wallach+ 2009]
- Adjusted the depth, prior on weights, and link function
- Used BBVI for all analyses
## DEF evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>$p(w)$</th>
<th>NYT</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA [Blei+ 2003]</td>
<td></td>
<td>2717</td>
<td>1711</td>
</tr>
<tr>
<td>DocNADE [Larochelle+ 2012]</td>
<td></td>
<td>2496</td>
<td>1725</td>
</tr>
<tr>
<td>Sparse Gamma 100</td>
<td>$\emptyset$</td>
<td>2525</td>
<td>1652</td>
</tr>
<tr>
<td>Sparse Gamma 100-30</td>
<td>$\Gamma$</td>
<td>2303</td>
<td>1539</td>
</tr>
<tr>
<td>Sparse Gamma 100-30-15</td>
<td>$\Gamma$</td>
<td><strong>2251</strong></td>
<td>1542</td>
</tr>
<tr>
<td>Sigmoid 100</td>
<td>$\emptyset$</td>
<td>2343</td>
<td>1633</td>
</tr>
<tr>
<td>Sigmoid 100-30</td>
<td>$\mathcal{N}$</td>
<td>2653</td>
<td>1665</td>
</tr>
<tr>
<td>Sigmoid 100-30-15</td>
<td>$\mathcal{N}$</td>
<td>2507</td>
<td>1653</td>
</tr>
<tr>
<td>Poisson 100</td>
<td>$\emptyset$</td>
<td>2590</td>
<td>1620</td>
</tr>
<tr>
<td>Poisson 100-30</td>
<td>$\mathcal{N}$</td>
<td>2423</td>
<td>1560</td>
</tr>
<tr>
<td>Poisson 100-30-15</td>
<td>$\mathcal{N}$</td>
<td>2416</td>
<td>1576</td>
</tr>
<tr>
<td>Poisson log-link 100-30</td>
<td>$\Gamma$</td>
<td>2288</td>
<td><strong>1523</strong></td>
</tr>
<tr>
<td>Poisson log-link 100-30-15</td>
<td>$\Gamma$</td>
<td>2366</td>
<td>1545</td>
</tr>
</tbody>
</table>
Neuroscience analysis of 220 million fMRI measurements

[Manning+ 2014]
all models

evaluateable

differentiable

conditionally conjugate
Black box variational inference

- BBVI with the score function estimator
- BBVI with the reparameterization gradient
- Probabilistic programming and autodifferentiation VI
- How to derive BBVI
- Economists want to understand how people choose items

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

\[ p(y_t) = p(y_{t1})p(y_{t2} | y_{t1}) \cdots p(y_{tn} | y_t^{[n-1]}) . \]

The customer picks each item conditional on features of the other items. These features capture that, e.g.,

- medialuna and coffee go well together
- a customer doesn’t need to buy four different types of dolce de leche
- chimichurri sauce goes on *everything*

But these features are latent!
The conditional probability of picking item $c$ is a log linear model
\[ p(y_{ti} = c \mid \text{previously selected items}) \propto \exp\{\Psi_{tc}\}. \]

The parameter is
\[ \Psi_{tc} = \rho_c \top \left( \sum_{j=1}^{i-1} \alpha_{y_{tj}} \right) \]

This is an embedding method \cite{Bengio+2003, Rudolph+2016}.

- $\alpha_{\text{dolce}}$ : (latent) attributes of dolce de leche
- $\rho_{\text{coffee}}$ : attributes that go well with coffee
From a dataset of shopping trips, infer the posterior $p(\alpha, \rho | y)$.

- Posterior of per-item attributes and per-item interaction coefficients
- 3,200 customers; 5,600 items; 570K trips; 5.7M purchased items
SHOPER on 5.7M purchases.

[more results in Ruiz+ 2017]
We can evaluate \( \log p(\alpha, \rho, y) \)

And we can evaluate its gradient \( \nabla_{\alpha,\rho} \log p(\alpha, \rho, y) \).

We can use the reparameterization gradient.
Differentiable models

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

\[ \epsilon \sim s(\epsilon) \]
\[ z = t(\epsilon, \nu) \rightarrow z \sim q(z; \nu) \]

- For example,

\[ \epsilon \sim \text{Normal}(0, 1) \]
\[ z = \epsilon \sigma + \mu \rightarrow z \sim \text{Normal}(\mu, \sigma^2) \]

- Also assume \( \log p(x, z) \) and \( \log q(z) \) are differentiable with respect to \( z \)
all models  evaluable  conditionally conjugate  differentiable
Nonconjugate models

\[ p(\beta, z, x) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i | \beta) \]

- 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
BBVI with the reparameterization gradient

\[
\nabla_{\nu} \mathcal{L} = \mathbb{E}_{s(\epsilon)} \left[ \nabla_z [ \log p(x, z) - \log q(z; \nu) ] + \nabla_{\nu} t(\epsilon, \nu) \right]
\]

- Write the gradient as an expectation,
- Form noisy gradients with Monte Carlo.
- This is the reparameterization gradient.

[Glasserman 1991; Fu 2006; Kingma+ 2014; Rezende+ 2014; Titsias+ 2014]
BBVI with the reparameterization gradient

\[ \nabla_{\nu} \mathcal{L} = \mathbb{E}_{s(\epsilon)} \left[ \nabla_{z} [\log p(x, z) - \log q(z; \nu)] \right] \nabla_{\nu} t(\epsilon, \nu) \]

- Can use autodifferentiation to take gradients (especially of the model)
- Can use and reuse different transformations [e.g., Naesseth+ 2017]
- Requires continuous latent variables
Algorithm 4: Reparameterization Black Box Variational Inference

**Input:** data \( x \), model \( p(z, x) \).

Initialize \( \nu \) randomly.
Set \( \rho_t \) appropriately.

**while** not converged **do**

Take \( S \) samples from the auxiliary variable

\[ \epsilon_s \sim s(\epsilon) \quad s = 1 \ldots S \]

Calculate the noisy gradient

\[ \tilde{g}_t = \frac{1}{S} \sum_{s=1}^{S} \nabla_z [\log p(x, t(\epsilon_s, \nu_n)) - \log q(t(\epsilon_s, \nu_n); \nu_n)] \nabla \nu_t(\epsilon_s, \nu_n) \]

Update the variational parameters

\[ \nu_{t+1} = \nu_t + \rho_t \tilde{g}_t \]

**end**
SHOPPER on 5.7M purchases.
Score vs. reparameterization gradients

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

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

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

- Differentiates the instantaneous ELBO
- Requires differentiable models, i.e., no discrete variables
- Requires variational approximation to have form \( z = t(\epsilon, \nu) \)
- Better behaved variance
Variance comparison

Black box variational inference \cite{ranganath2014black} takes a different approach. The gradient estimator uses the gradient of the variational approximation and avoids using the gradient of the model. For example, the following estimator

\[
\log q(x) / \log p(x) \cdot \log \frac{\log q(x)}{\log p(x)} - \log \text{det} J^{-1} \cdot \log \text{det} J
\]

and the gradient estimator in Equation (7) both lead to unbiased estimates of the exact gradient. While it is more general—it does not require the gradient of the model and thus applies to more settings—its gradients can suffer from high variance.

Figure 8 empirically compares the variance of both estimators for two models. Figure 8a shows the variance of both gradient estimators for a simple univariate model, where the posterior is a Gamma $\Gamma(10, 10)$. We estimate the variance using ten thousand recalculations of the gradient $L$, across an increasing number of samples $M$. The gradient has lower variance; in practice, a single sample succeeds. (See the experiments in Section 4.)

Figure 8b shows the same calculation for a 100-dimensional nonlinear regression model with likelihood $N(y | \tanh(x) ; I)$ and a Gaussian prior on the regression coefficients $\theta$. Because this is a multivariate example, we also show the gradient with a variance reduction scheme using control variates described in Ranganath et al. \cite{ranganath2014black}. In both cases, the gradient is computationally more efficient.

3.3 Sensitivity to Transformations

uses a transformation $T$ from the unconstrained space to the constrained space. We now study how the choice of this transformation affects the non-Gaussian posterior approximation in the original latent variable space.

Consider a posterior density in the Gamma family, with support over $R > 0$. Figure 9 shows three configurations of the Gamma, ranging from $\Gamma(1, 2)$, which places most of its mass close to $\theta = 0$, to $\Gamma(10, 10)$, which is centered at $\theta = 1$. Consider two transformations $T_1$ and $T_2$ $T_1 W \theta \& \log \theta$ and $T_2 W \theta \& \log \exp \theta / 1$;

\cite{kucukelbir16}
Amortized inference and the variational autoencoder

- **Amortization:**
  Variational parameters a parameterized function of the input \( \nu_\eta(x) \)
  [Gershman and Goodman 2014]

- This lets us “learn to infer.” Test-time inference is fast.
  (Open question: There seems to be more to the story.)

- Plays nicely with autodifferentiation and reparameterization gradients.
Amortized inference and the variational autoencoder

- **Model:** Deep generative model [Kingma et al. 2013; Rezende et al. 2014]

\[ \mathbf{z}_i \sim \mathcal{N}(0, I) \]
\[ \mathbf{x}_i \sim \mathcal{N}(f_\theta(\mathbf{z}_i), \sigma^2) \]

- **Variational parameters** \( \mathbf{v}_\eta(\mathbf{x}) \), also a deep neural network

- **Algorithm:**
  - Update \( \eta \) with reparameterization gradient
  - Update \( \theta \) with gradient
Black box variational inference

- BBVI with the score function estimator
- BBVI with the reparameterization gradient
- Probabilistic programming and autodifferentiation VI
- How to derive BBVI
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

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

\[ \alpha_j \sim \text{inv-gamma}(1, 1) \quad j = 1 \ldots k \]

\[ w_{x,j} \sim \mathcal{N}(0, \sigma \cdot \alpha_j) \]

\[ w_{y,j} \sim \mathcal{N}(0, \sigma \cdot \alpha_j) \]

\[ z_i \sim \mathcal{N}(0, I) \quad i = 1 \ldots n \]

\[ x_i \sim \mathcal{N}(w_x^T z_i, \sigma) \]

\[ y_i \sim \mathcal{N}(w_y^T z_i, \sigma) \]

- 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 consider
  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; // hyper-parameters on weights
  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);
  }
}
Exploring Taxi Trajectories

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

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

What would take weeks → a single day.
1. Transform the model.

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

$$p(\zeta, x) = p(x, s(\zeta)) \left| \det J_s(\zeta) \right|.$$
Automatic differentiation variational inference

2. Redefine the variational problem.

- The variational family is mean-field Gaussian

\[ q(\zeta; \nu) = \prod_{k=1}^{K} \varphi(\zeta_k; \nu_k), \]

- The ELBO is

\[ \mathcal{L} = \mathbb{E}_{q(\zeta)} \left[ \log p(x, s(\zeta)) + \log | \det J_s(\zeta) | \right] + \mathbb{H}(q) \]
Automatic differentiation variational inference

3. Use the reparameterization gradient

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

Implementation

- Stan automates going from $\log p(x, z)$ to
  \[
  \log p(x, s(\zeta)) + \log |\det J_s(\zeta)|
  \]
  \[
  \nabla_\zeta (\log p(x, s(\zeta)) + \log |\det J_s(\zeta)|)
  \]

- 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
Black box variational inference

- BBVI with the score function estimator
- BBVI with the reparameterization gradient
- Probabilistic programming and autodifferentiation VI
- How to derive BBVI
Score gradient

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

\[ \text{score function} \quad \text{instantaneous ELBO} \]

Reparameterization gradient

\[ \nabla \mathcal{L} = \mathbb{E}_{s(\epsilon)} \left[ \nabla_z \left[ \log p(x, z) - \log q(z; \nu) \right] \right. \left. \nabla_{\nu} t(\epsilon, \nu) \right] \]

\[ \text{gradient of instantaneous ELBO} \quad \text{gradient of transformation} \]
A recipe for variational inference

\[ p(z, x) \]

Posit a model, a joint distribution of hidden and observed variables.
A recipe for variational inference

\[ q(z; \nu) \]

Choose the variational family, distributions of the hidden variables.
A recipe for variational inference

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

Write the ELBO, the objective function for finding a \( q(z; \nu) \) close to \( p(z \mid x) \).
A recipe for variational inference

\[ \mathcal{L}(\nu) = x \nu^2 + \log \nu \]  (example)

Integrate: The ELBO is a function of data and variational parameters.
A recipe for variational inference

\[ \nabla_{\nu} \mathcal{L}(\nu) = 2x \nu + 1/\nu \quad \text{(example)} \]

Take derivatives.
A recipe for variational inference

\[ \nu_{t+1} = \nu_t + \rho_t \nabla_{\nu} \mathcal{L} \]

Optimize.
A recipe for variational inference

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

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

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

\[
z \sim N(0, 1)
\]
\[
y_i \mid x_i, z \sim \text{Bernoulli}(\sigma(zx_i)),
\]

where \(\sigma(\cdot)\) is the logistic function, mapping reals to \((0, 1)\).
VI for Bayesian logistic regression

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

\[
\mathcal{L}(\mu, \sigma^2) = \mathbb{E}_q[\log p(z) + \log p(y | x, z) - \log q(z)]
\]
VI for Bayesian logistic regression

\[ \mathcal{L}(\mu, \sigma^2) = \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y | x, z)] \]
VI for Bayesian logistic regression

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

\[ = -\frac{1}{2} \mu^2 + \frac{1}{2} \log \sigma^2 + \mathbb{E}_q[\log p(y | x, z)] + C \]
VI for Bayesian logistic regression

\[ \mathcal{L}(\mu, \sigma^2) = \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y | x, z)] \]
\[ = -\frac{1}{2} (\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + \mathbb{E}_q[\log p(y | x, z)] + C \]
\[ = -\frac{1}{2} (\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \exp(xz))] \]
VI for Bayesian logistic regression

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

\[ = -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C \]

\[ = -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \exp(xz))] \]

\[ = -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))] \]
VI for Bayesian logistic regression

\[ \mathcal{L}(\mu, \sigma^2) = E_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \]
\[ = -\frac{1}{2} (\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + E_q[\log p(y \mid x, z)] + C \]
\[ = -\frac{1}{2} (\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + E_q[yxz - \log(1 + \exp(xz))] \]
\[ = -\frac{1}{2} (\mu^2 + \sigma^2) + \frac{1}{2} \log \sigma^2 + yx\mu - E_q[\log(1 + \exp(xz))] \]

We are stuck—we cannot analytically take the expectation.
Options?

- Derive a model-specific bound
  [Jordan and Jaakola 1996], [Braun and McAuliffe 2008], others

- Use other approximations (that require model-specific analysis)
  [Wang and Blei 2013], [Knowles and Minka 2011]

- But neither satisfies the *black box criteria*. 
The problem with the VI recipe

The integral is hard to take.
Solution: Swap integration and differentiation

\[ p(x, z) \quad \nabla \nu \quad \int (\cdots) q(z; \nu) \, dz \quad q(z; \nu) \]

Swap! Now we can use MC gradients and stochastic optimization.
The new recipe

- This is the key idea behind modern methods in variational inference
- It has enabled score gradients, reparameterization gradients, amortized inference, probabilistic programming, complex variational families, and alternative divergences.
- How do we reverse differentiation and integration?
Reversing the gradient and the expectation

- Denote the *instantaneous ELBO*

  \[ g(z, \nu) = \log p(x, z) - \log q(z; \nu). \]

- The ELBO is its expectation

  \[ \mathcal{L} = \mathbb{E}_q [g(z, \nu)] = \int q(z; \nu) g(z, \nu) dz \]

- We want to calculate \( \nabla_\nu \mathcal{L}. \)
Reversing the gradient and the expectation

Fact:

\[ \nabla_\nu q(z; \nu) = q(z; \nu) \nabla_\nu \log q(z; \nu). \]
Reversing the gradient and the expectation

Fact:

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

With this,

$$\nabla_{\nu} \mathcal{L} = \nabla_{\nu} \int q(z; \nu) g(z, \nu) dz$$
Reversing the gradient and the expectation

Fact:

\[ \nabla_{\nu}q(z; \nu) = q(z; \nu)\nabla_{\nu}\log q(z; \nu). \]

With this,

\[ \nabla_{\nu}L = \nabla_{\nu}\int q(z; \nu)g(z, \nu)dz \]
\[ = \int \nabla_{\nu}q(z; \nu)g(z, \nu) + q(z; \nu)\nabla_{\nu}g(z, \nu)dz \]
Reversing the gradient and the expectation

Fact:

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

With this,

$$\nabla_{\nu} \mathcal{L} = \nabla_{\nu} \int q(z; \nu) g(z, \nu) dz$$

$$= \int \nabla_{\nu} q(z; \nu) g(z, \nu) + q(z; \nu) \nabla_{\nu} g(z, \nu) dz$$

$$= \int q(z; \nu) \nabla_{\nu} \log q(z; \nu) g(z, \nu) + q(z; \nu) \nabla_{\nu} g(z, \nu) dz$$
Reversing the gradient and the expectation

Fact:

\[ \nabla_\nu q(z; \nu) = q(z; \nu) \nabla_\nu \log q(z; \nu). \]

With this,

\[
\nabla_\nu L = \nabla_\nu \int q(z; \nu) g(z, \nu) dz
\]

\[
= \int \nabla_\nu q(z; \nu) g(z, \nu) + q(z; \nu) \nabla_\nu g(z, \nu) dz
\]

\[
= \int q(z; \nu) \nabla_\nu \log q(z; \nu) g(z, \nu) + q(z; \nu) \nabla_\nu g(z, \nu) dz
\]

\[
= \mathbb{E}_{q(z; \nu)} [\nabla_\nu \log q(z; \nu) g(z, \nu) + \nabla_\nu g(z, \nu)]
\]
Reversing the gradient and the expectation

Fact:

$$\nabla_\nu q(z; \nu) = q(z; \nu) \nabla_\nu \log q(z; \nu).$$

With this,

$$\nabla_\nu \mathcal{L} = \nabla_\nu \int q(z; \nu) g(z, \nu) dz$$

$$= \int \nabla_\nu q(z; \nu) g(z, \nu) + q(z; \nu) \nabla_\nu g(z, \nu) dz$$

$$= \int q(z; \nu) \nabla_\nu \log q(z; \nu) g(z, \nu) + q(z; \nu) \nabla_\nu g(z, \nu) dz$$

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

We have written the gradient as an expectation.
Black box variational inference

- Derive the score gradient
- Derive the reparameterization gradient
The score gradient

- The black-box gradient is

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

- The black-box gradient is

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

- Simplify the second term

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

- The black-box gradient is

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

- Simplify the second term

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

- This gives the score gradient

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

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

\[
\epsilon \sim s(\epsilon) \\
z = t(\epsilon, \nu) \\
\rightarrow z \sim q(z; \nu)
\]

- For example,

\[
\epsilon \sim \text{Normal}(0, 1) \\
z = \epsilon \sigma + \mu \\
\rightarrow z \sim \text{Normal}(\mu, \sigma^2)
\]

- Also assume \( \log p(x, z) \) and \( \log q(z) \) are differentiable with respect to \( z \)
The reparameterization gradient

- The black box gradient is

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

\[\text{Note that } \nabla_{\nu} \log s(\epsilon) = 0. \text{ Now use the chain rule:} \]

\[
\nabla_{\nu} \mathcal{L} = \mathbb{E}_{s(\epsilon)}[\nabla_{\nu} g(t(\epsilon, \nu), \nu)] = \mathbb{E}_{s(\epsilon)}[\nabla_{z} \log p(x, z) - \log q(z; \nu)] \nabla_{\nu} t(\epsilon, \nu) - \nabla_{\nu} \log q(z; \nu)
\]

This is the reparameterization gradient.

\[\text{[Glasserman 1991; Fu 2006; Kingma + 2014; Rezende + 2014; Titsias + 2014]}\]
The reparameterization gradient

- The black box gradient is

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

- Rewrite using using \( z = t(\epsilon, \nu) \),

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

- The black box gradient is

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

- Rewrite using using \( z = t(\epsilon, \nu) \),

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

- Note that \( \nabla_{\nu} \log s(\epsilon) = 0 \). Now use the chain rule:

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

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

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

This is the reparameterization gradient.

[Glasserman 1991; Fu 2006; Kingma+ 2014; Rezende+ 2014; Titsias+ 2014]
Black box variational inference

- BBVI with the score gradient
- BBVI with the reparameterization gradient
- Probabilistic programming and autodifferentiation VI
- How to derive BBVI
Nonconjugate models

\[ p(\beta, z, x) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i | \beta) \]

- 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
all models

evaluable

cconditionally conjugate
differentiable
A Tour of Variational Inference (with one picture)
PROBABILISTIC MACHINE LEARNING

- ML methods that *connect domain knowledge to data*.
- Provides a computational methodology for scalable modeling
- 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 **General** and **scalable** 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
  - Probabilistic programming
- 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

$p(z \mid x)$

$KL(q(z; \nu^*) \mid\mid p(z \mid x))$

- Expectation propagation [Minka 2001]
- Belief propagation [Yedidia 2001]
- Operator variational inference [Ranganath+ 2016]
- $\chi$-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]
Should I be skeptical about variational inference?

- **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]
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?


