VARIATIONAL INFERENCE: FOUNDATIONS AND INNOVATIONS

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

with Rajesh Ranganath (Princeton) and Shakir Mohamed (DeepMind)
Communities discovered in a 3.7M node network of U.S. Patents

[Gopalan and Blei, PNAS 2013]
### Topics found in 1.8M articles from the New York Times

[Hoffman, Blei, Wang, Paisley, JMLR 2013]
Scenes, concepts and control.

[Eslami+ 2016, Lake+ 2015]
Population analysis of 2 billion genetic measurements

[Gopalan, Hao, Blei, Storey, Nature Genetics (in press)]
Neuroscience analysis of 220 million fMRI measurements

[Manning+ PLOS ONE 2014]
Compression and content generation.

[Van den Oord+ 2016, Gregor+ 2016]
Analysis of 1.7M taxi trajectories, in Stan

[Kucukelbir+ 2016]
Customized data analysis is important to many fields.

Pipeline separates assumptions, computation, application

Eases collaborative solutions to statistics problems
Posterior inference is the key algorithmic problem.

Answers the question: What does this model say about this data?

Our goal: General and scalable approaches to posterior inference
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]
PART I

Main ideas and historical context
Probabilistic machine learning

- 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 turns inference into optimization.
- 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]
History

- Variational inference adapts ideas from statistical physics to probabilistic inference. Arguably, it began in the late eighties with Peterson and Anderson (1987), who used mean-field methods to fit a neural network.

- This idea was picked up by Jordan’s lab in the early 1990s—Tommi Jaakkola, Lawrence Saul, Zoubin Gharamani—who generalized it to many probabilistic models. (A review paper is Jordan et al., 1999.)

- In parallel, Hinton and Van Camp (1993) also developed mean-field for neural networks. Neal and Hinton (1993) connected this idea to the EM algorithm, which lead to further variational methods for mixtures of experts (Waterhouse et al., 1996), HMMs (MacKay, 1997), and neural networks (Barber and Bishop, 1998).
There is now a flurry of new work on variational inference, making it scalable, easier to derive, faster, more accurate, and applying it to more complicated models and applications.

Modern VI touches many important areas: probabilistic programming, reinforcement learning, neural networks, convex optimization, Bayesian statistics, and myriad applications.

Our goal is to teach you the basics, explain some of the newer ideas, and to suggest open areas of new research.
Variational inference

Part I: Main ideas and historical context

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

Part II: Mean-field VI and stochastic VI

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

Part III: Stochastic gradients of the ELBO

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

Part IV: Summary
Variational inference

$\mathcal{L} = \frac{1}{j} \sum_{j=1}^{J} \text{KL}(q(z; \nu^*) \| p(z \mid x))$

VI approximates difficult quantities from complex models.

With **stochastic optimization** we can

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

Mean-field variational inference
and stochastic variational inference
Mean-field variational inference casts Bayesian computation as optimization. Stochastic variational inference scales to massive data.
Motivation: Topic Modeling

Topic models use posterior inference to discover the hidden thematic structure in a large collection of documents.
Example: Latent Dirichlet Allocation (LDA)

**Seeking Life’s Bare (Genetic) Necessities**

*COLD SPRING HARBOR, NEW YORK—* How many genes does an organism need to survive? Last week at the genome meeting here,* eight 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
Example: Latent Dirichlet Allocation (LDA)

- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics
Example: Latent Dirichlet Allocation (LDA)

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

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

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

- Encodes assumptions about data with a factorization of the joint
- Connects assumptions to algorithms for computing with data
- Defines the posterior (through the joint)
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.
<table>
<thead>
<tr>
<th></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>1</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>2</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>3</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>4</td>
<td>Giants</td>
<td>1</td>
<td>House</td>
<td>Bush</td>
<td>Political Party</td>
<td>Clinton</td>
<td>Campaign</td>
<td>Republican</td>
<td>House</td>
<td>Party</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>Housing</td>
<td>House</td>
<td>Buildings</td>
<td>Development</td>
<td>Space</td>
<td>Percent</td>
<td>Real</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>Iraqi</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
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 \textit{i}th data point \( x_i \) only depends on \( z_i \) and \( \beta \).

Compute \( p(\beta, z \mid x) \).
Conditionally conjugate models

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,

\[
p(z_i | \beta, x_i) = h(z_i) \exp\{\eta_\ell(\beta, x_i)^\top z_i - a(\eta_\ell(\beta, x_i))\}
\]

\[
p(\beta | z, x) = h(\beta) \exp\{\eta_g(z, x)^\top \beta - a(\eta_g(z, x))\}.
\]
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 variable.

- The global parameter comes from conjugacy \[ \text{[Bernardo and Smith, 1994]} \]

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

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

Global variables

Local variables

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

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

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

- 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.
Mean-field variational inference

- 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,

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

$$q(\beta; \lambda) = h(\beta) \exp\{\lambda^\top \beta - a(\lambda)\}.$$
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].
  - Caveat: The ELBO is not convex.
Mean-field variational inference for LDA

- 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

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
Mean-field variational inference for LDA
Classical variational inference

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

Initialize $\lambda$ randomly.

repeat
  for each data point $i$ do
    Set local parameter $\phi_i \leftarrow \mathbb{E}_\lambda [\eta_\ell(\beta, x_i)]$.
  end

Set global parameter

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

until the ELBO has converged
Conditionally conjugate models

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

- 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)
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

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 $\times$’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.

28
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 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 \( x_1, x_2, \ldots \) 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]

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

- Construct a noisy natural gradient,

\[
\hat{\nabla}_\lambda \mathcal{L}(\lambda) = \alpha + n E_{\phi_j}[t(Z_j, x_j)] - \lambda. \]

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

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

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

**repeat**

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

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

Set intermediate global parameter

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

Set global parameter

$$\lambda = (1 - \rho_t) \lambda + \rho_t \hat{\lambda}.$$**until** forever
Stochastic variational inference

Subsample data

Infer local structure

Update global structure

MASSIVE DATA

GLOBAL HIDDEN STRUCTURE

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 $\sqrt{\cdot}$'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. 28
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

<table>
<thead>
<tr>
<th>Documents analyzed</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
<th>12288</th>
<th>16384</th>
<th>32768</th>
<th>49152</th>
<th>65536</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top eight words</td>
<td>systems</td>
<td>road</td>
<td>made</td>
<td>service</td>
<td>service</td>
<td>systems</td>
<td>companies</td>
<td>services</td>
</tr>
<tr>
<td></td>
<td>road</td>
<td>made</td>
<td>service</td>
<td>announced</td>
<td>national</td>
<td>west</td>
<td>language</td>
<td>road</td>
</tr>
</tbody>
</table>

[Hoffman et al., 2010]
### Figure 5

Topics found in a corpus of 1.8 million articles from the New York Times. Modified from Hofmann et al. (2013).

#### Topics

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game</td>
<td>Life</td>
<td>Film</td>
<td>Book</td>
<td>Wine</td>
</tr>
<tr>
<td>Season</td>
<td>Know</td>
<td>Movie</td>
<td>Life</td>
<td>Street</td>
</tr>
<tr>
<td>Team</td>
<td>School</td>
<td>Show</td>
<td>Books</td>
<td>Hotel</td>
</tr>
<tr>
<td>Coach</td>
<td>Street</td>
<td>Life</td>
<td>Novel</td>
<td>House</td>
</tr>
<tr>
<td>Play</td>
<td>Man</td>
<td>Television</td>
<td>Story</td>
<td>Room</td>
</tr>
<tr>
<td>Points</td>
<td>Family</td>
<td>Films</td>
<td>Man</td>
<td>Night</td>
</tr>
<tr>
<td>Games</td>
<td>Says</td>
<td>Director</td>
<td>House</td>
<td>Place</td>
</tr>
<tr>
<td>Giants</td>
<td>House</td>
<td>Man</td>
<td>War</td>
<td>Restaurant</td>
</tr>
<tr>
<td>Second</td>
<td>Children</td>
<td>Story</td>
<td>War</td>
<td>Park</td>
</tr>
<tr>
<td>Players</td>
<td>Night</td>
<td>Says</td>
<td>Children</td>
<td>Garden</td>
</tr>
</tbody>
</table>

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

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

### Figure 4

$c$ illustrates the graphical model. This model is closely related to a linear factor model, except that each cell's distribution is determined by hidden variables that depend on the cell's row and column. The overlapping plates show how the observations at the $n$th row share its embedding $w_n$ but use different variables $γ_m$ for each column. Similarly, the observations in the $m$th column share its embedding $γ_m$ but use different variables $w_n$ for each row. Casting matrix factorization...
Communities discovered in a 3.7M node network of U.S. Patents

[Gopalan and Blei, PNAS 2013]
SVI scales many models

- Bayesian mixture models
- Time series models (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models (LDA and some variants)
Mean-field variational inference casts Bayesian computation as optimization. Stochastic variational inference scales to massive data.
PART III

Black box variational inference
Monte Carlo gradients enable *black box variational inference*, algorithms that efficiently perform Bayesian computation in any model.
Approximate inference can be difficult to derive.

Especially true for models that are not conditionally conjugate

E.g., discrete choice models, Bayesian generalized linear models, ...
Black box variational inference

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

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

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

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

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

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

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

[Ranganath+ 2015]
Deep exponential families

All distributions are in canonical exponential family form

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

\[
\eta(\cdot) = g(z_{n,\ell+1}^{\top} w_{\ell,k}).
\]

(Note: Inner product is not strictly necessary.)
Deep exponential families

Possibilities for the hidden layers

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

Nonlinearities arise naturally. There are two sources.

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

E.g., when \( z \) are Bernoulli then \( \nabla a \) is the sigmoid function.
Example: Text data

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

$$x_{n,i} \sim \text{Poisson}(g(w_{0,i}^T z_{n,1}))$$
Bernoulli DEF, aka sigmoid belief network [Neal, 1990]

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

\[
\mathbb{E}[z_{n,\ell}, k] = \sigma(z_{n,\ell+1}^\top w_{\ell+1, k})
\]

- Variable \( z_{n,\ell, k} \) represents whether component \((\ell, k)\) is “on.”
Poisson DEF

- Poisson layers; identity link; Gaussian prior on the weights
- Expectation is the exponentiated dot product
  \[
  \mathbb{E}[z_{n,\ell,k}] = \exp(z_{n,\ell+1}^\top w_{\ell+1,k})
  \]
- Variable \( z_{n,\ell,k} \) represents the “count” of component \((\ell, k)\).
Poisson DEF (with log link)

- Poisson layers; log link $g(\cdot) = \log(\cdot)$.
- Expectation is dot product
  
  $$\mathbb{E}[z_{\ell,k}] = z_{n,\ell+1}^\top w_{\ell+1,k}$$

- Here we place a Gamma prior on $w$. 
Sparse Gamma DEF

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

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

- Sufficient statistics are \( z \) and \( \log z \)
- When \( \alpha < 1 \) the mass concentrates around zero.
Sparse Gamma DEF

- Use sparse gamma distributions for the latent variables.

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

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

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

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

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

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

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

Goal: Try out many DEFs on a data set
  - Explore distributions, link functions, number of layers

Solution: \textit{black box variational inference} (BBVI)

Let’s derive BBVI in general; we will get back to DEFs later.
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)

Calculate the resulting expectation, i.e., take the integral.
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
Simplest example: Bayesian logistic regression

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

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

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

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

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

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

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

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

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

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

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

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

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

- But neither satisfies our criteria for *generic inference*. 
The problem with the VI recipe

\[ p(x, z) \quad \Rightarrow \quad \int (\cdots) q(z; \nu) \, dz \quad \Rightarrow \quad \nabla_{\nu} \quad \Rightarrow \quad q(z; \nu) \]

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

\[ p(x, z) \xrightarrow{\nabla_{\nu}} \int (\cdots) q(z; \nu) \, dz \]

Now we can use Monte Carlo gradients and stochastic optimization.
The new recipe

- This is the key idea behind modern methods in variational inference

- It has enabled score gradients, reparameterization gradients, amortized inference, probabilistic programming, complex variational families, and alternative divergences.

- Next: The general mathematics and some specific examples
Reversing the gradient and the expectation

- Define the “instantaneous ELBO”
  \[ g(z, \nu) = \log p(x, z) - \log q(z; \nu). \]

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

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

Recall the fact

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

Recall the 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

Recall the 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

Recall the 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

Recall the fact

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

With this,

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

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

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

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

Recall the fact

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

With this,

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

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

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

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

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

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

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

- Recall

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

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

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

- Sometimes called the likelihood ratio or REINFORCE gradient

[Glynn 1990; Williams, 1992; Wingate+ 2013; Ranganath+ 2014; Mnih+ 2014]
Noisy unbiased gradients

- We construct noisy unbiased gradients with Monte Carlo,

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

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

- To compute a noisy gradient of the ELBO, we need to
  - 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 analysis needed.
Algorithm 1: Basic Black Box Variational Inference

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

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

while not converged do

Take $S$ samples from the variational distribution

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

Update the variational parameters

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

end
Black box criteria

<table>
<thead>
<tr>
<th>Variable</th>
<th>Log density $\log q(z; \lambda)$</th>
<th>Score $\nabla_\lambda \log q(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson($\lambda$)</td>
<td>$-\lambda + z \log \lambda - \log z!$</td>
<td>$-1 + \frac{z}{\lambda}$</td>
</tr>
<tr>
<td>Bernoulli($\sigma(\lambda)$)</td>
<td>$z \lambda - \log(1 + \exp(\lambda))$</td>
<td>$z - \sigma(\lambda)$</td>
</tr>
<tr>
<td>Gamma($\lambda$, 1)</td>
<td>$-\log \Gamma(\lambda) + (\lambda - 1) \log z - z$</td>
<td>$-\Psi(\lambda) + \log(z)$</td>
</tr>
<tr>
<td>Exponential($\frac{1}{\lambda}$)</td>
<td>$-\log \lambda - \frac{z}{\lambda}$</td>
<td>$-\frac{1.0}{\lambda} + \frac{z}{\lambda^2}$</td>
</tr>
<tr>
<td>Normal($\lambda$, 1)</td>
<td>$-\frac{1}{2} \log(2\pi) - \frac{1}{2}(z - \lambda)^2$</td>
<td>$z - \lambda$</td>
</tr>
</tbody>
</table>
Black box variational inference

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

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

- We must control the variance of the gradient
  - Rao-Blackwellization, control variates, importance sampling, ...

- Adaptive learning rates [Duchi+ 2011; Tieleman and Hinton 2012]

- Stochastic variational inference, for handling massive data
Nonconjugate models

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

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

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

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

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

[Ranganath+ 2015]
Empirical study

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

<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]
More assumptions?

The black box criteria are to

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

Can we make additional assumptions that are not too restrictive?
Roadmap

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

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

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

- 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

- Recall

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

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

- Rewrite using \( z = t(\varepsilon, \nu) \),
  \[ \nabla_\nu L = \mathbb{E}_{s(\varepsilon)}[\nabla_\nu \log s(\varepsilon)g(t(\varepsilon, \nu), \nu) + \nabla_\nu g(t(\varepsilon, \nu), \nu)] \]
The reparameterization gradient

- Recall

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

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

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

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

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

This is also known as the reparameterization gradient.

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

Black box variational inference (Kucukelbir et al., 2016) takes a different approach (Ranganath et al., 2014). The gradient estimator uses the gradient of the variational approximation and avoids using the gradient of the model. For example, the following estimator:

\[
q(\theta | x; T) \propto \exp(-\frac{1}{2} \theta^T J_T \theta - 1 \log q(\theta | x; T) - \frac{1}{2} \theta^T \theta)
\]

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 re-calculations of the gradient across an increasing number of samples. 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 \(N(y | \tanh(x)^T \theta, 1)\) 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. (2014). In both cases, the gradient estimator 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 \(\mathbb{R}^+\). Figure 9 shows three configurations of the Gamma, ranging from Gamma \(\Gamma(1, 2)\), which places most of its mass close to \(\theta = 0\), to Gamma \(\Gamma(10, 10)\), which is centered at \(\theta = 1\). Consider two transformations \(T_1\) and \(T_2\):

\[
T_1 \theta = \log \theta
\]

and

\[
T_2 \theta = \log \exp \theta - 1/\theta.
\]

[Kucukelbir+ 2016]
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 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
- Requires variational approximation to have form $z = t(\epsilon, \nu)$
- Better behaved variance
Roadmap

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

- Write a model down as a program; “compile” it to an inference procedure.
- We deployed reparameterization gradients in Stan; 10,000 modelers can use variational inference. [Kucukelbir+ 2016]
Supervised pPCA with ARD

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

Data: 1.7 million taxi rides from Porto, Portugal

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

What would take us weeks → a single day.
ADVI details

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

The mapping is in the joint,

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

Stan provides a library for transforming probabilistic programs.
2. **Redefine the variational problem** with a Gaussian variational distribution.

The variational family is

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

The evidence lower bound is

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

We can use this variational problem **across transformable models**.
3. Use the reparameterization trick, where \( \eta \sim \mathcal{N}(0, I) \),

\[
\mathcal{L} = \mathbb{E}\left[ \log p\left( x, T^{-1}(S_{\mu, \omega}^{-1}(\eta)) \right) + \log \left| \det J_{T^{-1}} \left( S_{\mu, \omega}^{-1}(\eta) \right) \right| \right] + \sum_{k=1}^{K} \omega_k.
\]

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

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

(a) Linear Regression with ARD

(b) Hierarchical Logistic Regression

(a) Gamma Poisson Predictive Likelihood

(b) Dirichlet Exponential Predictive Likelihood

[Kucukelbir+ 2016]
Roadmap

- Score function estimator and basic black box variational inference
- Reparameterization gradient
- Autodifferentiation VI and probabilistic programming
Monte Carlo gradients enable *black box variational inference*, algorithms that efficiently perform Bayesian computation in any model.
Nonconjugate models

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

- 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
PART IV

Summary
The probabilistic pipeline

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

- **Posterior inference** is the key algorithmic problem.
- Answers the question: What does this model say about this data?
- Our goal: **General** and **scalable** approaches to posterior inference
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 turns inference into optimization.
- 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, BR, and others.)
Variational inference

With **stochastic optimization** we can

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

Part I: Main ideas and historical context

Part II: Mean-field VI and stochastic VI
Hoffman+, *Stochastic Variational Inference*, 2013

Part III: Stochastic gradients of the ELBO
Ranganath+, *Black Box Variational Inference*, 2014
Rezende+, *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*, 2014
Kucukelbir+ *Automatic Differentiation Variational Inference*, 2016

Part IV: Summary
Conditionally conjugate models

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

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

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

- 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
Edward: Probabilistic modeling, inference, and criticism

github.com/blei-lab/edward

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

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

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

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