Advances in Approximate Inference

Yingzhen Li, Cheng Zhang
Microsoft Research Cambridge
What is the Number?

07? 09?
67? 69?
...

🤔 🐱
What is the Number?

09
What is the Diagnosis?

Injury?
Osteoarthritis?
Neuropathic pain?

... ...
What is the Diagnosis?

Neuropathic pain
(might have spine injury)
Uncertainty is Important
Bayesian ML / Probability Theory

Decision making under uncertainty

Image courtesy of Sebastian Nowozin
Re-use of the image for any other purpose is not allowed
Graphical Representation

\[ p(x, y, z) = p(y)p(z)p(x|y, z) \]
Graphical Representation

\[ p(x, y, z) = p(z) \prod_{n} p(y_n) p(y_n | y_n, z) \]
Graphical Representation

\[ p(x, y, z) = p(z) \prod_{n}^{N} p(x_n) p(y_n | x_n, z) \]

\[ p(x, y) = p(z) \prod_{n}^{N} p(y_n) p(x_n | y_n, z) \]
Discriminative Model vs Generative Model

\[ p(x, y, z) = p(z) \prod_{n} p(x_n) P(y_n|x_n, z) \]

\[ p(x, y) = p(z) \prod_{n} p(y_n) P(x_n|y_n, z) \]
Discriminative Model Example

- Bayesian Logistic Regression

<table>
<thead>
<tr>
<th>Name</th>
<th>A-level math score</th>
<th># parents in STEM</th>
<th>Study STEM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>89</td>
<td>0</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Bob</td>
<td>95</td>
<td>1</td>
<td>1 (Yes)</td>
</tr>
<tr>
<td>Ty</td>
<td>82</td>
<td>1</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Emma</td>
<td>98</td>
<td>2</td>
<td>1 (Yes)</td>
</tr>
<tr>
<td>Anna</td>
<td>92</td>
<td>0</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Mo</td>
<td>88</td>
<td>1</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Li</td>
<td>95</td>
<td>0</td>
<td>1 (Yes)</td>
</tr>
</tbody>
</table>
## Discriminative Model Example:

- **Bayesian Logistic Regression**

\[
p(y = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1 + w_2 x_2}}
\]

### Example Data

<table>
<thead>
<tr>
<th>Name</th>
<th>A-level math score</th>
<th># parents in STEM</th>
<th>Study STEM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>89</td>
<td>0</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Bob</td>
<td>95</td>
<td>1</td>
<td>1 (Yes)</td>
</tr>
<tr>
<td>Ty</td>
<td>82</td>
<td>1</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Emma</td>
<td>98</td>
<td>2</td>
<td>1 (Yes)</td>
</tr>
<tr>
<td>Anna</td>
<td>92</td>
<td>0</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Mo</td>
<td>88</td>
<td>1</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Li</td>
<td>95</td>
<td>0</td>
<td>1 (Yes)</td>
</tr>
</tbody>
</table>

The table above shows the names, A-level math scores, number of parents in STEM, and whether they studied STEM.
Discriminative Model Example

• Bayesian Logistic Regression

<table>
<thead>
<tr>
<th>Name</th>
<th>A-level math score</th>
<th># parents in STEM</th>
<th>Study STEM?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>89</td>
<td>0</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Bob</td>
<td>95</td>
<td>1</td>
<td>1 (Yes)</td>
</tr>
<tr>
<td>Ty</td>
<td>82</td>
<td>1</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Emma</td>
<td>98</td>
<td>2</td>
<td>1 (Yes)</td>
</tr>
<tr>
<td>Anna</td>
<td>92</td>
<td>0</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Mo</td>
<td>88</td>
<td>1</td>
<td>0 (No)</td>
</tr>
<tr>
<td>Li</td>
<td>95</td>
<td>0</td>
<td>1 (Yes)</td>
</tr>
</tbody>
</table>

\[ p(y = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1 + w_2 x_2}} \]
Discriminative Model Example

\[ p(y = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1 + w_2 x_2}} \]
Discriminative Model Example

Computation Graph

\[ p(y = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1 + w_2 x_2}} \]

\[ w^T \Phi(X) \]
Discriminative Model Example:

\[ p(y = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1 + w_2 x_2}} \]

Computation Graph
### Discriminative Model Example

**Computation Graph**

\[ p(y = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1 + w_2 x_2}} \]

**Gaussian Processes**

Radford Neal’s PhD thesis (1994)
Generative Model Example

• Topic Model
Generative Model Example:

• Latent Dirichlet Allocation

<table>
<thead>
<tr>
<th>ID</th>
<th>topic</th>
<th>neural</th>
<th>distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>16</td>
<td>1</td>
</tr>
</tbody>
</table>

Topic distribution per document:
e.g. 30% “topic model”, 40% “natural language processing”,
30% “interpretability”

Word distribution per topic:
e.g. under “topic model”: “Dirichlet” 2%, “topic” 4%,
“Categorical” 1.5%, .....
Generative Model Example

Topics
User embedding
Underlying health conditions

Documents
Movie rating
Symptoms
How to infer the unknowns?
The Central Computation for Inference

- Inference: infer the **unknowns**
  - Unobserved/latent variables in the model
  - Quantities depending on the latent variables in the model
The Central Computation for Inference

• Inference: infer the unknowns
  • Unobserved/latent variables in the model
  • Quantities depending on the latent variables in the model

\[ \int F(\theta) \pi(\theta) d\theta \]

(For discrete probability measures, integration becomes discrete sum.)

- Integrand function
- Probability measure
- Random variable (unobserved)
Bayesian Inference

\[ \pi(\theta) = p(\theta|\text{data}) \]

\[ P(\theta|\text{data}) = \frac{P(\theta)P(\text{data} | \theta)}{P(\text{data})} \]

- \( P(\theta) \): prior distribution
- \( P(\text{data} | \theta) \): likelihood of \( \theta \) given \( \text{data} \)
- \( P(\theta|\text{data}) \): posterior distribution of \( \theta \) given \( \text{data} \)
- \( P(\text{data}) \): marginal likelihood/model evidence
  \[ P(\text{data}) = \int P(\theta)P(\text{data} | \theta) \]

Image courtesy of Sebastian Nowozin
Re-use of the image for any other purpose is not allowed
Computation Challenge

• The central equation for inference:

\[
\int F(\theta) \pi(\theta) d\theta
\]

“What is the prediction distribution of the test output given a test input?”

\[ F(\theta) = p(y|x, \theta), \pi(\theta) = p(\theta | D), \]
\[ D = \text{observed datapoints} \]
Computation Challenge

• The central equation for inference:

\[ \int F(\theta) \pi(\theta) d\theta \]

“What is the mean of this distribution?”

\( F(\theta) = \theta, \pi(\theta) \) can be complicated and high dimensional
Computation Challenge

• The central equation for inference:

\[ \int F(\theta) \pi(\theta) d\theta \]

“What is the probability of generating this image?”

\[ F(\theta) = \delta(\text{NN}(\theta) = x_0), \pi(\theta) = N(0, I) \]

\[ \theta \sim \pi(\theta) \]

\[ x_0 \]
Computation Challenge

• The central equation for inference:

$$\int F(\theta) \pi(\theta) d\theta$$

“What is the weather forecast for tomorrow?”

Answering this in a Bayesian way:

$\theta$: forecasting simulator settings
$D$: historical weather record

$F(\theta) = Simulator(\theta), \pi(\theta) = p(\theta \mid D)$
Nature laughs at the difficulties of integration.

--Pierre-Simon Laplace

Integration in Bayesian Computation

\[ \int F(\theta) \pi(\theta) d\theta \]

The probability distribution \( \pi(\theta) \) is intractable

Approximate Inference
(in a strict sense)

This tutorial
Integration in Bayesian Computation

The integrand $F(\theta)$ is intractable

The probability distribution $\pi(\theta)$ is intractable

Implicit Models
Bayesian Optimisation, Probabilistic Numerics

Approximate Inference
(in a strict sense)

This tutorial
Integration in Bayesian Computation

The integrand $F(\theta)$ is intractable

The probability distribution $\pi(\theta)$ is intractable

Both $F(\theta)$ and $\pi(\theta)$ are intractable
Approximate Inference

• Central task: approximate $\pi(\theta)$

$q(\theta) \approx \pi(\theta)$

(Assumed $\int F(\theta)q(\theta)d\theta$ can be computed or approximated efficiently.)
Approximate Inference

• Central task: approximate $\pi(\theta)$

$$q(\theta) \approx \pi(\theta)$$

Approximate distribution design

Explicit distributions  Implicit distributions
Approximate Inference

- Central task: approximate $\pi(\theta)$

$$q(\theta) \approx \pi(\theta)$$

Approximate distribution design

Algorithm for fitting $q(\theta)$ to $\pi(\theta)$

Explicit distributions

Implicit distributions

Optimisation-based approaches

Sampling-based approaches

min $\text{Loss}(q(\theta), \pi(\theta))$
Tutorial Outline

Basics
- Probabilistic modelling
- Approximate inference
- Variational inference

Advances
- Scalable variational inference
- Monte Carlo techniques
- Amortized inference
- $q$ distribution design
- Optimization objective design

Applications
- Bayesian neural networks
- Partially observed VAEs
- Future challenges
Bayesian Inference

\[ P(\theta | D) = \frac{P(\theta)P(D | \theta)}{P(D)} \]

- \( P(\theta) \): prior
- \( P(D | \theta) \): likelihood
- \( P(\theta | D) \): posterior
- \( P(D) \): marginal
Variational Inference (VI)

The posterior

\[ p(\theta|D) = \frac{p(D | \theta)p(\theta)}{p(D)} \]

The variational distribution

\[ q_\phi(\theta) \]
Inference as Optimization

\[ p(\theta|D) \quad \text{Kullback-Leibler (KL) divergence} \quad q_\phi(\theta) \]
Kullback-Leibler Divergence

\[ KL[q(\theta)||p(\theta)] = -\int q(\theta) \log \frac{p(\theta)}{q(\theta)} \, d\theta = E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}] \]

• When \( p = q \), KL is 0
• Otherwise, KL > 0
• It measures how similar are these two distributions
Let’s Derive the Objective of VI

• Minimize $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = -E_{q(\theta)} \left[ \log \frac{p(\theta|D)}{q(\theta)} \right]$$
Let’s Derive the Objective of VI

• Minimize $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = -E_{q(\theta)} \left[ \log \frac{p(\theta|D)}{q(\theta)} \right]$$

$$= -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{p(D)q(\theta)} \right] = -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{q(\theta)} - \log p(D) \right]$$
Let’s Derive the Objective of VI

• Minimize $KL[q(\theta) \| p(\theta|D)]$

$$
KL[q(\theta) \| p(\theta|D)] = -E_{q(\theta)} \left[ \log \frac{p(\theta|D)}{q(\theta)} \right]
$$

$$
= -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{p(D)q(\theta)} \right] = -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{q(\theta)} - \log p(D) \right]
$$

$$
= \log p(D) - E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{q(\theta)} \right]
$$

Model Evidence
Let’s Derive the Objective of VI

Minimize \( KL[q(\theta)||p(\theta|D)] \)

\[
KL[q(\theta)||p(\theta|D)] = \log p(D) - E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]
\]

Maximize \( E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \)
Let’s Derive the Objective of VI

Minimize \( KL[q(\theta)||p(\theta|D)] \)

\[
KL[q(\theta)||p(\theta|D)] = \log p(D) - E_q(\theta) \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]
\]

Maximize \( L = E_q(\theta) \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \)

Evidence Lower Bound (ELBO)

Model Evidence

“Model Evidence = ELBO + KL”
Alternative Derivation

Let’s start with the model evidence

\[ \log p(D) \]
Alternative Derivation

\[ \log p(D) = \log \int p(\theta, D) \, d\theta \]
Alternative Derivation

\[
\begin{align*}
\log p(D) &= \log \int p(\theta, D) \, d\theta \\
&= \log \int \frac{p(\theta, D) q(\theta)}{q(\theta)} \, d\theta \\
&= \log E_{q(\theta)} \left[ \frac{p(\theta, D)}{q(\theta)} \right]
\end{align*}
\]
Jensen’s Inequality

If $f$ is a convex function, then

$$f(E[X]) \leq E[f(X)]$$

If $f$ is a concave function, then

$$f(E[X]) \geq E[f(X)]$$

Alternative Derivation

\[
\log p(D) = \log \int p(\theta, D) \, d\theta \\
= \log \int \frac{p(\theta, D) q(\theta)}{q(\theta)} \, d\theta \\
= \log E_{q(\theta)}\left[\frac{p(\theta, D)}{q(\theta)}\right]
\]

Log is a concave function, then
\[
f(E[X]) \geq E[f(X)]
\]

Jensen’s inequality
Alternative Derivation

Model Evidence
\[
\log p(D) = \log \int p(\theta, D) \, d\theta \\
= \log \int \frac{p(\theta, D) \, q(\theta)}{q(\theta)} \, d\theta \\
= \log E_{q(\theta)} \left[ \frac{p(\theta, D)}{q(\theta)} \right] \\
\geq E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]
\]

Evidence Lower Bound (ELBO)
Variational Inference (VI)

The posterior

\[ p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \]

The variational distribution

\[ q_\phi(\theta) \]

\[ L = E_{q_\phi(\theta)} \left[ \log \frac{p(D, \theta)}{q_\phi(\theta)} \right] = \log p(D) - KL[q_\phi(\theta)||p(\theta)] \]
Mean-field Variational Inference

• A type of choices of the variational distribution
• The name origins in the mean field theory of physics
• The variational distribution factorizes

\[
q_\phi(\theta) = \prod_{i=1}^{K} q_{\phi_i}(\theta_i)
\]

A Gaussian Example

\[ p(z) = N(z|\mu, \Lambda^{-1}) \]

\[ q(z) = q(z_1)q(z_2) \]

Mean-field Variational Inference

$$L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]$$

Fully Factorized Variational Distribution

$$q(\theta) = \prod_{i=1}^{K} q_{\phi_i}(\theta_i)$$

Mean-field Variational Inference

\[ L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \]

ELBO

\[ q(\theta) = \prod_{i=1}^{K} q_{\phi_i}(\theta_i) \]

Fully Factorized Variational Distribution

\[ L = \int q(\theta_j) E_{q(\theta_{-j})} \left[ \log p(\theta_j, D | \theta_{-j}) \right] d\theta_j - \int q(\theta_j) \log p(\theta_j) d\theta_j + c_j \]

Mean-field Variational Inference

\[
L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]
\]

ELBO

\[
q(\theta) = \prod_{i=1}^{K} q_{\phi_i}(\theta_i)
\]

Fully Factorized Variational Distribution

\[
L = \int q(\theta_j) E_{q(\theta_{-j})} \left[ \log p(\theta_j, D | \theta_{-j}) \right] d\theta_j - \int q(\theta_j) \log p(\theta_j) d\theta_j + c_j
\]

\[
q^*(\theta_j) \propto \exp \left( E_{q(\theta_{-j})} \left[ \log p(\theta_j, D | \theta_{-j}) \right] \right)
\]

Part II: Advances

- Scalable variational inference
- Monte Carlo methods
- Amortized inference
- Approximate distribution design
- Optimization objective design
Stochastic Variational Inference

$p(\theta, \xi, x) = p(\theta) \prod_{i=1}^{M} p(\xi_i | \theta) p(x_i | \xi_i, \theta)$
Stochastic Variational Inference

\[ L = E_q \left[ \log \frac{p(\theta, \xi, x)}{q(\theta, \xi)} \right] \]

\[ = E_q \left[ \log \frac{p(\theta) \prod_{i=1}^{M} p(\xi_i | \theta)p(x_i | \xi_i, \theta)}{q(\theta) \prod_{i=1}^{M} q(\xi_i)} \right] \]

\[ = E_q \left[ \log \frac{p(\theta)}{q(\theta)} + \sum_{i=1}^{M} E_q \left[ \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right] \right] \]

- \( O(M) \) time to compute in each update iteration
- \( M \) can be extremely large
- Even one iteration might not be affordable

\[ p(\theta, \xi, x) = p(\theta) \prod_{i=1}^{M} p(\xi_i | \theta)p(x_i | \xi_i, \theta) \]
Stochastic Variational Inference

\[ L = E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^{M} E_q \left[ \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right] \]

stochastic approximation with \( S \ll M \)

\[ \hat{L} = E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \frac{M}{S} \sum_{i=1}^{S} E_q \left[ \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right] \]

Computational complexity: \( O(M) \rightarrow O(S) \)

Hoffman et al. Stochastic Variational Inference. JMLR 2013.
How Stochastic Gradient Works

$\nabla F(x_i)$ gradient of each single data point $x_i$

$E_x[\nabla F(x)]$ batch gradient considering all data points
How Stochastic Gradient Works

\[ \nabla F(x_i) \] gradient of each single data point \( x_i \)

\[ E_x[\nabla F(x)] \] batch gradient considering all \( M = 10 \) data points

\[ \frac{M}{S} \sum_{s=1}^{S} \nabla F(x_s) \] mini-batch gradient/stochastic gradient estimated using \( S=3 \) data points
How Stochastic Gradient Works

\[ \nabla F(x_i) \text{ gradient of each single data point } x_i \]

\[ E_x[\nabla F(x)] \text{ batch gradient considering all } M = 10 \text{ data points} \]

\[ \frac{M}{S} \sum_{s=1}^{S} \nabla F(x_s) \text{ mini-batch gradient/stochastic gradient estimated using } S=3 \text{ data points} \]

Stochastic Variational Inference

\[
L = E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^{M} E_q \left[ \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right]
\]

\[
\nabla L = \nabla E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^{M} E_q \left[ \nabla \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right]
\]

Stochastic approximation with \( S \ll M \)

\[
\hat{L} = E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \frac{M}{S} \sum_{i=1}^{S} E_q \left[ \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right]
\]

\[
\nabla \hat{L} = \nabla E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \frac{M}{S} \sum_{i=1}^{S} E_q \left[ \nabla \log \frac{p(\xi_i | \theta)p(x_i | \xi_i)}{q(\xi_i)} \right]
\]

Stochastic Gradient

Hoffman et al. Stochastic Variational Inference. JMLR 2013.
Nature laughs at the difficulties of integration.

--Pierre-Simon Laplace

Monte Carlo Approximation

• To approximate: $E_{p(x)}[f(x)]$
• MC Approximation:
  1. Sample $x_1, x_2, \ldots, x_K \sim p(x)$
  2. Evaluate $f(x_i)$ for each sample
  3. Compute $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^{K} f(x_i)$

Unbiased Monte Carlo estimate
Log-derivative Trick

\[ \nabla_{\theta} \log p_{\theta}(x) \]
Log-derivative Trick

\[ \nabla_{\theta} \log p_{\theta}(x) = \frac{1}{p_{\theta}(x)} \nabla p_{\theta}(X) \]
Log-derivative Trick

$$\nabla_\theta \log p_\theta(x) = \frac{1}{p_\theta(x)} \nabla p_\theta(X)$$

$$\nabla_\theta p_\theta(x) = p_\theta(x) \nabla_\theta \log p_\theta(x)$$
REINFORCE Gradients

\[ \text{ELBO} \quad L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] \]

Gradient of the ELBO

\[ \nabla_\phi L = \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} \, d\theta \]

REINFORCE Gradients

**ELBO**  \( L = E_{q\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q\phi(\theta)} \right] \)

Gradient of the ELBO

\[
\nabla_\phi L = \nabla_\phi E_{q\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q\phi(\theta)} \right] = \int \nabla_\phi \left\{ q\phi(\theta) \log \frac{p(\theta, D)}{q\phi(\theta)} \right\} d\theta \\
= \int \nabla_\phi q\phi(\theta) \log \frac{p(\theta, D)}{q\phi(\theta)} d\theta + \int q\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q\phi(\theta)} d\theta
\]

**REINFORCE Gradients**

**ELBO**  
\[ L = E_{q_{\phi}(\theta)} \left[ \log \frac{p(\theta,D)}{q_{\phi}(\theta)} \right] \]

Gradient of the ELBO

\[ \nabla_{\phi} L = \nabla_{\phi} E_{q_{\phi}(\theta)} \left[ \log \frac{p(\theta,D)}{q_{\phi}(\theta)} \right] = \int \nabla_{\phi} \left\{ q_{\phi}(\theta) \log \frac{p(\theta,D)}{q_{\phi}(\theta)} \right\} d\theta \]

\[ = \int \nabla_{\phi} q_{\phi}(\theta) \log \frac{p(\theta,D)}{q_{\phi}(\theta)} d\theta + \int q_{\phi}(\theta) \nabla_{\phi} \log \frac{p(\theta,D)}{q_{\phi}(\theta)} d\theta \]

\[ = \int q_{\phi}(\theta) \nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta,D)}{q_{\phi}(\theta)} d\theta - \int \nabla_{\phi} q_{\phi}(\theta) d\theta \]

Log-derivative trick

\[ \nabla q_{\phi}(\theta) = q_{\phi}(\theta) \nabla_{\phi} \log q_{\phi}(\theta) \]


REINFORCE Gradients

ELBO \[ L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] \]

Gradient of the ELBO
\[ \nabla_\phi L = \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta \]
\[ = \int \nabla_\phi q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta \]
\[ = \int q_\phi(\theta) \nabla_\phi \log q_\phi(\theta) \left\{ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right\} d\theta - \int \nabla_\phi q_\phi(\theta) d\theta \]
\[ = \nabla_\phi \int q_\phi(\theta) d\theta = \nabla_\phi 1 = 0 \]

REINFORCE Gradients

\[ \text{ELBO} \quad L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] \]

Gradient of the ELBO

\[ \nabla_\phi L = \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta \]

\[ = \int \nabla_\phi q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta \]

\[ = \int q_\phi(\theta) \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta - \int \nabla_\phi q_\phi(\theta) d\theta \]

\[ = E_{q_\phi(\theta)} \left[ \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] \]

Score function

BBVI

**ELBO**

\[ L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \]

\[
\nabla_{\phi} L = E_{q(\theta)} \left[ \nabla_{\phi} \log q(\theta) \log \frac{p(\theta, D)}{q(\theta)} \right]
\]

- To approximate: \( E[f(x)] \)
- **MC Approximation:**
  1. Sample \( x_1, x_2, \ldots, x_K \sim p(x) \)
  2. Evaluate \( f(x_i) \) for each sample
  3. Compute \( E[f(x)] \approx \frac{1}{K} \sum_{i=1}^{K} f(x_i) \)

Ranganath et al. Black box variational inference. AISTATS 2014
BBVI

To approximate: \( E[f(x)] \)

MC Approximation:
1. Sample \( x_1, x_2, \ldots, x_K \sim p(x) \)
2. Evaluate \( f(x_i) \) for each sample
3. Compute \( E[f(x)] \approx \frac{1}{K} \sum_{i=1}^{K} f(x_i) \)

Gradient of the ELBO
\[
\nabla_{\phi} L = E_{q(\theta)} \left[ \nabla_{\phi} \log q(\theta) \log \frac{p(\theta, D)}{q(\theta)} \right]
\]

1. Sample \( \theta_1, \theta_2, \ldots, \theta_K \sim q(\theta) \)

Ranganath et al. Black box variational inference. AISTATS 2014
BBVI

Gradient of the ELBO

$$\nabla_{\phi} L = E_{q_{\phi}(\theta)} \left[ \nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta, D)}{q_{\phi}(\theta)} \right]$$

- To approximate: $E[f(x)]$
- MC Approximation:
  1. Sample $x_1, x_2, \ldots, x_K \sim p(x)$
  2. Evaluate $f(x_i)$ for each sample
  3. Compute $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^{K} f(x_i)$

Ranganath et al. Black box variational inference. AISTATS 2014
BBVI

Gradient of the ELBO
\[ \nabla_\phi L = E_{q_\phi(\theta)} \left[ \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] \]

1. Sample \( \theta_1, \theta_2, \ldots, \theta_K \sim q_\phi(\theta) \)
2. Evaluate \( \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \) for each sample
3. The approximated gradient is:
\[ \nabla_\phi \hat{L} = \frac{1}{K} \sum_{i=1}^K \nabla_\phi \log q_\phi(\theta_i) \log \frac{p(\theta_i, D)}{q_\phi(\theta_i)} \]

To approximate: \( E[f(x)] \)

MC Approximation:
1. Sample \( x_1, x_2, \ldots, x_K \sim p(x) \)
2. Evaluate \( f(x_i) \) for each sample
3. Compute \( E[f(x)] \approx \frac{1}{K} \sum_{i=1}^K f(x_i) \)

Ranganath et al. Black box variational inference. AISTATS 2014
Black-box Variational Inference (BBVI)

Go beyond conjugate exponential family
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon), \ \theta = g(\epsilon, \phi)$
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon), \theta = g(\epsilon, \phi)$

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon$$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014
Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon)$, $\theta = g(\epsilon, \phi)$

\[
\theta \sim N(\mu, \sigma^2) \\
\epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon
\]

ELBO \[ L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] \]
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon)$, $\theta = g(\epsilon, \phi)$

\[ \theta \sim N(\mu, \sigma^2) \]
\[ \epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon \]

\[ \text{ELBO} \quad L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] \]

\[ \text{Gradient} \quad \nabla_\phi L = \nabla_\phi E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] \]

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014
Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon)$, $\theta = g(\epsilon, \phi)$

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon$$

ELBO

$$L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta,D)}{q_\phi(\theta)} \right] = E_r(\epsilon) \left[ \log \frac{p(g(\epsilon,\phi),D)}{q_\phi(g(\epsilon,\phi))} \right]$$

Gradient

$$\nabla_\phi L = \nabla_\phi E_r(\epsilon) \left[ \log \frac{p(g(\epsilon,\phi),D)}{q_\phi(g(\epsilon,\phi))} \right] = E_r(\epsilon) \left[ \nabla_\phi \log \frac{p(g(\epsilon,\phi),D)}{q_\phi(g(\epsilon,\phi))} \right]$$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014
Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon), \ \theta = g(\epsilon, \phi)$

\[ \theta \sim N(\mu, \sigma^2) \]
\[ \epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon \]

ELBO
\[ L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] \]

Gradient
\[ \nabla_\phi L = \nabla_\phi E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_r(\epsilon) \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] \]
\[ \nabla_\phi \hat{L} = \frac{1}{K} \sum_{k=1}^{K} \nabla_\phi \log \frac{p(g(\epsilon_k, \phi), D)}{q_\phi(g(\epsilon_k, \phi))}, \epsilon_k \sim r(\epsilon) \]
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon)$, $\theta = g(\epsilon, \phi)$

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon$$

ELBO

$$L = E_{q_\theta(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

Gradient

$$\nabla_\phi L = \nabla_\phi E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_r(\epsilon) \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

$$\nabla_\phi \hat{L} = \frac{1}{K} \sum_{k=1}^{K} \nabla_\phi \log \frac{p(g(\epsilon_k, \phi), D)}{q_\phi(g(\epsilon_k, \phi))}, \epsilon_k \sim r(\epsilon)$$
Reparameterization Trick

Express $q_\phi(\theta)$ as $\epsilon \sim r(\epsilon)$, $\theta = g(\epsilon, \phi)$

$\theta \sim N(\mu, \sigma^2)$
$\epsilon \sim N(0, 1), \theta = \mu + \sigma \epsilon$

ELBO

$\mathcal{L} = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$

Gradient

$\nabla_\phi \mathcal{L} = \nabla_\phi E_r(\epsilon) \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_r(\epsilon) \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$

$\nabla_\phi \hat{\mathcal{L}} = \frac{1}{K} \sum_{k=1}^{K} \nabla_\phi \log \frac{p(g(\epsilon_k, \phi), D)}{q_\phi(g(\epsilon_k, \phi))}, \epsilon_k \sim r(\epsilon)$
Variance Reduction Techniques in MCVI

• When non-differentiable, falls back to REINFORCE gradient

High variance!
Variance Reduction Techniques in MCVI

• When non-differentiable, falls back to REINFORCE gradient

High variance!

• Solutions to high variance REINFORCE gradients:
  • Low variance unbiased estimators with control variates
  • Biased estimators to enable reparam. trick (potentially low variance)
Variance Reduction Techniques in MCVI

• Control variate method:
  • Assume we want to estimate with MC simulation

\[ E_{q(\theta)}[F(\theta)] \approx \frac{1}{K} \sum_{k}^{K} F(\theta_k), \quad \theta_k \sim q(\theta) \]
Variance Reduction Techniques in MCVI

- Control variate method:
  - Assume we want to estimate with MC simulation
    \[
    E_q(\theta)[F(\theta)] \approx \frac{1}{K} \sum_{k=1}^{K} F(\theta_k), \quad \theta_k \sim q(\theta)
    \]

- Control variate: define a control function \( G(\theta) \) satisfying:
  - \( V_q(\theta)[G(\theta)] < \infty \)
  - Known or fast computable \( E_q(\theta)[G(\theta)] \)
Variance Reduction Techniques in MCVI

• Control variate method:
  • Then define the new MC estimator

\[ E_{q(\theta)}[F(\theta)] \approx \frac{1}{K} \sum_{k=1}^{K} \hat{F}(\theta_k), \quad \theta_k \sim q(\theta), \]

\[ \hat{F}(\theta) = F(\theta) - G(\theta) + E_{q(\theta)}[G(\theta)] \]

\[ V_{q(\theta)}[\hat{F}(\theta)] = V_{q(\theta)}[F(\theta)] + V_{q(\theta)}[G(\theta)] - 2 \text{Cov}_{q(\theta)}[F(\theta), G(\theta)] \]

< 0 if \( F \) and \( G \) are strongly and positively correlated
Variance Reduction Techniques in MCVI

• Application to REINFORCE gradient:
  
  • \( F(\theta) = \log \frac{p(D,\theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta) \) 
  
  • Define \( G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta) \)
  
  \[ \Rightarrow f(\theta) \]
Variance Reduction Techniques in MCVI

• Application to REINFORCE gradient:
  
  • \( F(\theta) = \log \frac{p(D,\theta)}{q(\theta)} \nabla_{\phi} \log q_{\phi}(\theta) \)
  
  • \( \hat{F}(\theta) = f(\theta) - g(\theta)\nabla_{\phi} \log q_{\phi}(\theta) + E_{q_{\phi}(\theta)}[G(\theta)] \)

  • Define \( G(\theta) = g(\theta)\nabla_{\phi} \log q_{\phi}(\theta) \)


• Define \( \Delta(\theta) \)
Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:
  \[ F(\theta) = \log \frac{p(D,\theta)}{q(\phi(\theta))} \nabla_{\phi} \log q(\phi(\theta)) \]
  \[ := f(\theta) \]
  \[ \text{Variance reduced gradient: } \hat{F}(\theta) = (f(\theta) - g(\theta)) \nabla_{\phi} \log q(\phi(\theta)) + E_{q(\phi(\theta))}[G(\theta)] \]
  \[ := \Delta(\theta) \]

- “Baseline” approach:
  \[ g(\theta) = b \]
  \[ \Rightarrow E_{q(\theta)}[G(\theta)] = b E_{q(\theta)}[\nabla_{\phi} \log q(\theta)] = b \nabla_{\phi} \int q(\phi(\theta)) d\theta = b \nabla_{\phi} 1 = 0 \] (log-derivative trick)
  \[ \Rightarrow \hat{F}(\theta) = \Delta(\theta) \nabla_{\phi} \log q(\phi(\theta)) \]
Variance Reduction Techniques in MCVI

• Application to REINFORCE gradient:

\[ F(\theta) = \log \frac{p(D, \theta)}{q(\theta)} \nabla_{\phi} \log q(\theta) \]

\[ := f(\theta) \]

- Variance reduced gradient: \( \hat{F}(\theta) = (f(\theta) - g(\theta))\nabla_{\phi} \log q(\theta) + E_{q(\theta)}[G(\theta)] \)

\[ := \Delta(\theta) \]

- “Baseline” approach:

\[ g(\theta) = b \]

\[ \Rightarrow E_{q(\theta)}[G(\theta)] = bE_{q(\theta)}[\nabla_{\phi} \log q(\theta)] = 0 \]

\[ \Rightarrow \hat{F}(\theta) = \Delta(\theta)\nabla_{\phi} \log q(\theta) \]

- Define \( G(\theta) = g(\theta)\nabla_{\phi} \log q(\theta) \)

\[ \Delta(\theta) \]

\[ \theta_1 \]

\[ \theta_2 \]

b fitted by minimising either \( V_{q(\theta)}[\hat{F}(\theta)] \) or \( E_{q(\theta)}[\Delta(\theta)^2] \)

Ranganath et al. Black Box Variational Inference. AISTATS 2014
Mnih and Gregor. Neural Variational Inference and Learning in Belief Networks. ICML 2014
Variance Reduction Techniques in MCVI

- **Application to REINFORCE gradient:**
  - \( F(\theta) = \log \frac{p^{(D,\theta)}}{q_{\phi}(\theta)} \nabla_{\phi} \log q_{\phi}(\theta) \)
  - Define \( G(\theta) = g(\theta)\nabla_{\phi} \log q_{\phi}(\theta) \)
  - Variance reduced gradient: \( \hat{F}(\theta) = (f(\theta) - g(\theta))\nabla_{\phi} \log q_{\phi}(\theta) + E_{q_{\phi}(\theta)}[G(\theta)] \)
  - “Taylor expansion” approach (e.g. 1st order):
    \[ g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0) \]

Paisley et al. Variational Bayesian Inference with Stochastic Search. ICML 2012
Gu et al. MuProp: Unbiased Backpropagation for Stochastic Neural Networks. ICLR 2016
Variance Reduction Techniques in MCVI

• Application to REINFORCE gradient:

\[ F(\theta) = \log \frac{p^{(D,\theta)}}{q_\phi(\theta)} \nabla \phi \log q_\phi(\theta) \]

\[ := f(\theta) \]

\[ \begin{align*}
\text{Variance reduced gradient: } \hat{F}(\theta) &= (f(\theta) - g(\theta)) \nabla \phi \log q_\phi(\theta) + E_{q_\phi(\theta)}[G(\theta)] \\
&= \Delta(\theta)
\end{align*} \]

• “Taylor expansion” approach (e.g. 1st order):

\[ g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0) \quad \Rightarrow \quad E_{q(\theta)}[G(\theta)] = (f(\theta_0) - \nabla_{\theta_0} f(\theta_0)\theta_0) E_{q(\theta)}[\nabla \phi \log q(\theta)] \\
+ \nabla_{\theta_0} f(\theta_0) E_{q(\theta)}[\theta \nabla \phi \log q(\theta)] \]

\[ = \nabla_{\theta_0} f(\theta_0) \nabla \phi E_{q(\theta)}[\theta] \quad \text{(log-derivative trick)}\]

Paisley et al. Variational Bayesian Inference with Stochastic Search. ICML 2012
Gu et al. MuProp: Unbiased Backpropagation for Stochastic Neural Networks. ICLR 2016
Variance Reduction Techniques in MCVI

• Application to REINFORCE gradient:

\[ F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta) \]

\[ G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta) \]

\[ \Rightarrow E_{q_\phi(\theta)}[G(\theta)] = (f(\theta_0) - \nabla_{\theta_0} f(\theta_0) \theta_0) E_{q_\phi(\theta)}[\nabla_\phi \log q(\theta)] + \nabla_{\theta_0} f(\theta_0) E_{q_\phi(\theta)}[\nabla_\phi \log q(\theta)] \]

\[ \approx \nabla_{\theta_0} f(\theta_0) \nabla_\phi [E_{q_\phi(\theta)}[\nabla_\phi \log q(\theta)] \text{ (log-derivative trick)}] \]

\[ \Rightarrow \hat{F}(\theta) = \Delta(\theta) \nabla_\phi \log q_\phi(\theta) + \nabla_{\theta_0} f(\theta_0) \nabla_\phi E_{q_\phi(\theta)}[\theta] \]

\[ := \nabla_\phi f(\theta_0) \text{ if } \theta_0 = E_{q_\phi(\theta)}[\theta] \]

- Define \( G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta) \)

- “Taylor expansion” approach (e.g. 1\textsuperscript{st} order):

\[ g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0) \]

\[ \Rightarrow E_{q_\phi(\theta)}[G(\theta)] = (f(\theta_0) - \nabla_{\theta_0} f(\theta_0) \theta_0) E_{q_\phi(\theta)}[\nabla_\phi \log q(\theta)] \]

\[ \approx 0 \text{ (log-derivative trick)} \]
Variance Reduction Techniques in MCVI

• Gumbel-Softmax trick
  • Biased gradient estimator
  • Empirically found to have smaller variance

Categorical distribution:

\[ p(y = k) = \pi_k, \sum_k \pi_k = 1 \]
Variance Reduction Techniques in MCVI

- Gumbel-Softmax trick
  - Biased gradient estimator
  - Empirically found to have smaller variance

Gumbel trick to sample $y$:
$$y = \arg\max g_k + \log \pi_k,$$
$$g_k \sim \text{Gumbel}(0, 1)$$
Variance Reduction Techniques in MCVI

• Gumbel-Softmax trick
  • Biased gradient estimator
  • Empirically found to have smaller variance

Gumbel trick to sample $y$:
$$y = \arg \max [g_k + \log \pi_k],$$
$$g_k \sim \text{Gumbel}(0, 1)$$

Jang et al. Categorical Reparameterization with Gumbel-Softmax. ICLR 2017
Maddison et al. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. ICLR 2017
Variance Reduction Techniques in MCVI

- Gumbel-Softmax trick
  - Biased gradient estimator
  - Empirically found to have smaller variance

\[
[y_1, \ldots, y_K] = \text{softmax}\left(\frac{(g_1 + \log \pi_1)}{\tau}, \ldots, \frac{(g_K + \log \pi_K)}{\tau}\right)
\]

Concrete distribution/Gumbel-Softmax trick:
sample the “soft vector” (instead of one-hot encoding of \( y \))

Jang et al. Categorical Reparameterization with Gumbel-Softmax. ICLR 2017
Maddison et al. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. ICLR 2017
Variance Reduction Techniques in MCVI

- VIMCO
- Jackknife
- SUMO
- REBAR
- Rao-Blackwellization
- unbiased
- MuProp
- reparam. gradient
- REINFORCE

For an incomplete list of variance reduced gradient estimators, see [http://yingzhenli.net/home/en/?page_id=1262](http://yingzhenli.net/home/en/?page_id=1262)
Latent Variable Model

\[ i \in [1, M] \]

- \( Z_i \)
- \( x_i \)

Topics
- Documents

Health conditions
- Symptoms

Student skill
- Exam performance

House ‘type’
- House information

\[ x_i \in [1, M] \]
Deep Latent Variable Model

\[ i \in [1, M] \]

\[ \theta \]

\[ z_i \rightarrow x_i \]

Topics
Documents

Health conditions
Symptoms

Student skill
Exam performance

House ‘type’
House information

\[ \theta \]

\[ z \]

\[ x \]
Amortized Inference

\[ i \in [1, M] \]

\[ \phi_i \rightarrow Z_i \rightarrow \theta \]

\[ \phi \text{ parameter for variational distribution} \]

\[ \theta \text{ decoder parameter} \]
Amortized Inference

\[ i \in [1, M] \]

\( \phi_i \quad Z_i \quad \theta \)

\( x_i \)

\( \phi \) parameter for variational distribution

\( \theta \) decoder parameter
Amortized Inference

\[ L = \log p(x) - KL[q(z)||p(z|x)] \]

\[ L_{amortized} = \log p(x) - KL[q(z|x)||p(z|x)] \]
Variational Auto-Encoders (VAE)

\[ L = \log p(x) - KL[q(z)||p(z|x)] \]

\[ L_{amortized} = \log p(x) - KL[q(z|x)||p(z|x)] \]
Variational Auto-Encoders (VAE)

\[ L = \log p(x) - KL[q(z)||p(z|x)] \]

\[ L_{amortized} = \log p(x) - KL[q(z|x)||p(z|x)] \]

Rezende et al. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014.
Variational Auto-Encoders (VAE)

\[ \epsilon \sim N(0,1) \]

\[
L_{amortized} = \log p(x) - KL[q(z|x) \mid\mid p(z|x)] \\
= E_{z \sim q(z|x)} [\log p_\theta(x|z)] - KL[q(z|x) \mid\mid p(z|x)]
\]

Rezende et al. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014.
How to apply amortization to other inference methods?
Amortized Inference: Further Examples

- Amortized SMC

Find the optimal proposal distribution for each sequence \( \{x_{1:T}\} \) & each time step

Explicitly parameterise & optimise \( (x_{1:T}, z_{1:t}) \rightarrow \text{proposal dist. for } z_t \)

Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Amortized Inference: Further Examples

• Amortized MCMC

Li et al. Approximate Inference with Amortised MCMC. ICML 2017 Workshop on Implicit Models
Amortized Inference: Further Examples

- Amortized Monte Carlo integration

Goal: estimate with
importance sampling

\[ E_p(z|x)[F_\eta(z)] \]

Find the optimal proposal
distributions for each \((x, \eta)\) pair

Explicitly parameterise & optimise
\((x, \eta) \rightarrow \text{proposal dist. for } z\)
Amortized Inference: Limitations

- Amortised approximate posteriors in practice are sub-optimal

- The "refinement" idea:
  - Initialise $q(z|x) = N(z; \mu, \sigma^2)$ with the amortised solution $\mu \leftarrow \mu_\phi(x), \sigma \leftarrow \sigma_\phi(x)$
  - Then run $T$ more VI gradient steps to update $\mu, \sigma$

Cremer et al. Inference Suboptimality in Variational Autoencoders. ICML 2018
Marino et al. Iterative Amortized Inference. ICML 2018
Kim et al. Semi-Amortized Variational Autoencoders. ICML 2018
Part II: Advances

- Scalable variational inference
- Monte Carlo methods
- Amortized inference
- Approximate distribution design
- Optimization objective design
Designing $q$ Distributions

- Structured approximations
- Normalizing flows
- Auxiliary variables & mixture distributions
- Implicit approximate posteriors
Structured Approximations

• introduce dependencies between random variables for $q$:

$$q(z) = \prod_i q(z_i)$$

Exact posterior $p(z \mid x)$

$$z_i \perp z_j \mid x$$

Hidden Markov Model

Mean-field approximation
Structured Approximations

- introduce dependencies between random variables for $q$:

$$q(z) = \prod_i q(z_i)$$

Main design question: the grouping and conditional dependency structure
Structured Approximations

• Auto-regressive distributions (as a specific dependency structure)

Hidden Markov Model

Structured approximation

Auto-regressive approximation

Main design question: the ordering of the latent variables
Normalizing Flows

• Change-of-variable formula:
  • $x$ is a random variable with probability density function (PDF) $p_X(x)$
  • $y = f(x)$ is an invertible mapping
Normalizing Flows

• Change-of-variable formula:
  • $x$ is a random variable with probability density function (PDF) $p_X(x)$
  • $y = f(x)$ is an invertible mapping
  • The probability mass is preserved, and the PDF for $y = f(x)$ satisfies
    \[ p_Y(y)dy = p_X(x)dx \]
    prob. mass of region around $y$  prob. mass of region around $x$
Normalizing Flows

• Change-of-variable formula:
  • $x$ is a random variable with probability density function (PDF) $p_X(x)$
  • $y = f(x)$ is an invertible mapping
  • The probability mass is preserved, and the PDF for $y = f(x)$ satisfies
    \[ p_Y(y)dy = p_X(x)dx \]
    prob. mass of region around $y$          prob. mass of region around $x$

\[
\begin{align*}
p_Y(y) &= p_X(x)\left|\frac{dx}{dy}\right|
\end{align*}
\]

\[
\begin{align*}
p_X(x) &= p_Y(y)\left|\frac{dy}{dx}\right|
\end{align*}
\]
Normalizing Flows

• Variational inference with Normalizing flow
  • Assume $q_0(z_0) = N(z_0; 0, I)$
  • Define $z = f_\phi(z_0)$ where $f_\phi(\cdot)$ is an invertible mapping parameterized by $\phi$

$$q(z) = q_0(z_0)|\text{det} \left( \frac{dz}{dz_0} \right)|^{-1} \quad \text{with} \quad z_0 = f_\phi^{-1}(z)$$

(change of variable: $q(z)dz = q_0(z_0)dz_0$)

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015
Normalizing Flows

• Variational inference with Normalizing flow
  • Assume \( q_0(z_0) = N(z_0; 0, I) \)
  • Define \( z = f_\phi(z_0) \) where \( f_\phi(\cdot) \) is an invertible mapping parameterized by \( \phi \)

\[
q(z) = q_0(z_0) \det \left( \frac{dz}{dz_0} \right)^{-1} \quad \text{with} \quad z_0 = f_\phi^{-1}(z)
\]

• Fit \( q(z) \) to \( p(x \mid z) \) with VI:

\[
L(q(z)) = E_{q(z)}[\log p(x \mid z) + \log p(z) - \log q(z)]
\]

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015
Normalizing Flows

• Variational inference with Normalizing flow
  
  • Assume $q_0(z_0) = N(z_0; 0, I)$
  
  • Define $z = f_\phi(z_0)$ where $f_\phi(\cdot)$ is an invertible mapping parameterized by $\phi$

  \[
  q(z) = q_0(z_0) \left| \det \frac{dz}{dz_0} \right|^{-1} \quad \text{with} \quad z_0 = f_\phi^{-1}(z)
  \]

• Fit $q(z)$ to $p(x \mid z)$ with VI:

  \[
  L(q(z)) = E_{q(z)} \left[ \log p(x \mid z) + \log p(z) - \log q(z) \right] = E_{q(z)} \left[ \log p(x, z) - \log q_0(z_0 = f_\phi^{-1}(z)) \left| \det \frac{dz}{dz_0} \right|^{-1} \right]
  \]

by def. of $q(z)$

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015
Normalizing Flows

• Variational inference with Normalizing flow
  • Assume \( q_0(z_0) = N(z_0; 0, I) \)
  • Define \( z = f_\phi(z_0) \) where \( f_\phi(\cdot) \) is an invertible mapping parameterized by \( \phi \)

\[
q(z) = q_0(z_0) | \det \left( \frac{dz}{dz_0} \right) |^{-1} \quad \text{with} \quad z_0 = f_\phi^{-1}(z)
\]

• Fit \( q(z) \) to \( p(x \mid z) \) with VI:

\[
L(q(z)) = E_{q(z)} \left[ \log p(x \mid z) + \log p(z) - \log q(z) \right]
\]

\[
= E_{q(z)} \left[ \log p(x, z) - \log q_0(z_0 = f_\phi^{-1}(z)) | \det \left( \frac{dz}{dz_0} \right) |^{-1} \right]
\]

\[
= E_{q_0(z_0)} \left[ \log p(x, f_\phi(z_0)) - \log q_0(z_0) + \log | \det \left( \frac{df_\phi}{dz_0} \right) | \right]
\]

by def. of \( q(z) \)

reparam. trick:

\( z \sim q(z) \iff z_0 \sim q_0(z_0), z = f_\phi(z_0) \)
Normalizing Flows

- Variational inference with Normalizing flow
  - Assume \( q_0(z_0) = N(z_0; 0, I) \)
  - Define \( z = f_\phi(z_0) \) where \( f_\phi(\cdot) \) is an invertible mapping parameterized by \( \phi \)

\[
q(z) = q_0(z_0) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1}\quad \text{with } z_0 = f_\phi^{-1}(z)
\]

- Fit \( q(z) \) to \( p(x \mid z) \) with VI:

\[
L(q(z)) = E_{q(z)} \left[ \log p(x \mid z) + \log p(z) - \log q(z) \right]
= E_{q(z)} \left[ \log p(x, z) - \log q_0(z_0 = f_\phi^{-1}(z)) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \right]
= E_{q_0(z_0)} \left[ \log p(x, f_\phi(z_0)) - \log q_0(z_0) + \log \left| \det \left( \frac{df_\phi}{dz_0} \right) \right| \right]
\]

- Computing ELBO requires \( \log \left| \det \left( \frac{df_\phi}{dz_0} \right) \right| \)

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015
Normalizing Flows

- Variational inference with Normalizing flow
  - Idea: define $f_\phi$ such that $\log |\det \left( \frac{df_\phi}{dz_0} \right)|$ is easy to compute!
  - Chain simple invertible mappings together to make a flexible mapping

$$f_\phi = f_K \circ f_{K-1} \circ \cdots \circ f_1, f_k(\cdot) := f_{\phi_k}(\cdot), \phi = \{\phi_k\}_{k=1}^K$$

- For each simple mapping, hopefully the Jacobian log-determinant is easy to compute

$$\Rightarrow \log |\det \left( \frac{df_\phi}{dz_0} \right)| = \sum_{k=1}^K \log |\det \left( \frac{dz_k}{dz_{k-1}} \right)|$$

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015
Normalizing Flows

• Goal: construct $f_k$ to enable fast compute of $\log | \det \left( \frac{dz_k}{dz_{k-1}} \right) |$

• Example (RealNVP): $y := f_{\phi_k}(x)$ computed as follows
Normalizing Flows

• Goal: construct $f_k$ to enable fast compute of $\log |\det(\frac{dz_k}{dz_{k-1}})|$

• Example (RealNVP): $y := f_{\phi_k}(x)$ computed as follows

Dinh et al. Density Estimation using Real NVP. ICLR 2017
Normalizing Flows

• Goal: construct $f_k$ to enable fast compute of $\log | \det \left( \frac{dz_k}{dz_{k-1}} \right) |$

  • Example (RealNVP): $y := f_{\phi_k}(x)$ computed as follows

\[ x = [x_1, x_2] \]

split:

\[ x_1 \]

identity mapping: $y_1 = x_1$

\[ x_2 \]
Normalizing Flows

• Goal: construct $f_k$ to enable fast compute of $\log | \det \left( \frac{dz_k}{dz_{k-1}} \right) |$

• Example (RealNVP): $y := f_{\phi_k}(x)$ computed as follows

identity mapping: $y_1 = x_1$

split: $x = [x_1, x_2]$

affine transform: $y_2 = x_2 \odot \exp(s(x_1)) + t(x_1)$
Normalizing Flows

• Goal: construct $f_k$ to enable fast compute of $\log | \det \left( \frac{dz_k}{dz_{k-1}} \right) |$

• Example (RealNVP): $y := f_{\phi_k}(x)$ computed as follows

$$y_2 = x_2 \odot \exp(s(x_1)) + t(x_1)$$

identity mapping: $y_1 = x_1$

split: $x = [x_1, x_2]$

merge: $y = [y_1, y_2]$

affine transform: $y_2 = x_2 \odot \exp(s(x_1)) + t(x_1)$

Dinh et al. Density Estimation using Real NVP. ICLR 2017
Normalizing Flows

• Goal: construct $f_k$ to enable fast compute of $\log |\det \left( \frac{dz_k}{dz_{k-1}} \right) |

• Example (RealNVP): $y := f_{\phi_k}(x)$ computed as follows

$$
\begin{align*}
    x &= [x_1, x_2] \\
    y_1 &= x_1 \\
    y_2 &= x_2 \odot \exp(s(x_1)) + t(x_1) \\
    y &= [y_1, y_2]
\end{align*}
$$

Jacobian:
$$
\frac{df_{\phi_k}}{dx} = \begin{pmatrix} I & 0 \\ dy_2/dx_1 & \text{diag}(\exp(s(x_1))) \end{pmatrix}
$$

Log-determinant of Jacobian:
$$
\Rightarrow \log \left| \det \left( \frac{df_{\phi}}{dx} \right) \right| = \sum_i s(x_1)_i
$$
Auxiliary Variables & Mixture Distributions

• Construct $q(\theta)$ as a (hierarchical) mixture distribution

$$q(\theta) = \int q(\theta | a) q(a) \, da$$

• $a$ is the auxiliary variable used to enrich the approximate posterior
Auxiliary Variables & Mixture Distributions

• Construct $q(\theta)$ as a (hierarchical) mixture distribution

$$q(\theta) = \int q(\theta | a) q(a) \, da$$

• $a$ is the auxiliary variable used to enrich the approximate posterior

• Example: Mixture of Gaussians

$a \sim q(a) = \text{Categorical}(\pi_1, ..., \pi_K)$

$\theta \sim q(\theta | a) = N(\theta; m_a, \Sigma_a)$

Can be very flexible with many components!
Auxiliary Variables & Mixture Distributions

• Construct $q(\theta)$ as a (hierarchical) mixture distribution

$$
q(\theta) = \int q(\theta | a) \, q(a) \, da
$$

• $a$ is the auxiliary variable used to enrich the approximate posterior
• Now the variational lower-bound becomes intractable:

$$
L(\phi) = E_{q(\theta)}[\log p(D, \theta)] - E_{q(\theta)}[\log q(\theta)]
$$

Estimated by Monte Carlo:
$$
a_k \sim q(a), \theta_k \sim q(\theta | a_k)
$$

Intractable density
$$
q(\theta) = \int q(\theta | a)q(a) \, da
$$
Auxiliary Variables & Mixture Distributions

• Solution: introducing an auxiliary variational lower-bound $L(\phi, r)$ with an auxiliary distribution $r(a|\theta)$:

\[
L(\phi) = E_{q(\theta)}[\log p(D, \theta)] - E_{q(\theta)}[\log q(\theta)]
\]

\[
L(\phi, r) = E_{q(\theta, a)}[\log p(D|\theta)] - KL[q(\theta, a)||p(\theta)r(a|\theta)]
\]

\[
KL(q(\theta)||p(\theta|D))
\]

\[
E_{q(\theta)}[KL[q(a|\theta)||r(a|\theta)]]
\]

Agakov and Barber. An Auxiliary Variational Method. ICONIP 2004
Salimans et al. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. ICML 2015
Auxiliary Variables & Mixture Distributions

• Solution: introducing an auxiliary variational lower-bound $L(\phi, r)$ with an auxiliary distribution $r(a|\theta)$:

$$L(\phi) = E_{q(\theta)}[\log p(D, \theta)] - E_{q(\theta)}[\log q(\theta)]$$

$$L(\phi, r) = E_{q(\theta,a)}[\log p(D|\theta)] - KL[q(\theta,a)||p(\theta)r(a|\theta)]$$

• Optimize $r(a|\theta)$ to close the gap!
• $L(\phi, r)$ estimated by Monte Carlo: $a_k \sim q(a), \theta_k \sim q(\theta | a_k)$

Agakov and Barber. An Auxiliary Variational Method. ICONIP 2004
Salimans et al. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. ICML 2015
Auxiliary Variables & Mixture Distributions

• Hierarchical mixture distributions for $q(\theta, a)$
  • VI-MCMC hybrid: build $q(\theta)$ with a Markov Chain:
Auxiliary Variables & Mixture Distributions

- Hierarchical mixture distributions for $q(\theta, a)$
- VI-MCMC hybrid: build $q(\theta)$ with a Markov Chain:

$$\theta_0 \implies \theta_1 \implies \theta_2 \implies \cdots \implies \theta_T$$

learn the transition kernel with VI:

$$\theta := \theta^T, a = \{\theta^{0:T-1}\}$$

$$q(\theta^T) = \int q_0(\theta^0) \prod_{t=1}^{T} K_{\phi}(\theta^t | \theta^{t-1}) d\theta^{0:T-1}$$

Salimans et al. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. ICML 2015
Huang et al. Improving Explorability in Variational Inference with Annealed Variational Objectives. NeurIPS 2018
Implicit Approximate Posteriors

• Two quantities computed in (approximate) Bayesian inference:
  
  approximate Bayesian predictive
  
  \[ p(y^*|x^*, D) \approx E_{q(\theta)}[p(y^*|x^*, \theta)] \]

  approximate posterior moments
  
  \[ E_{q(\theta)}[F(\theta)] \]
Implicit Approximate Posteriors

- Two quantities computed in (approximate) Bayesian inference:

  \[
  p(y^*|x^*, D) \approx E_{q(\theta)}[p(y^*|x^*, \theta)] \\
  \approx \frac{1}{K} \sum_k p(y^*|x^*, \theta_k), \ \theta_k \sim q(\theta)
  \]

  \[
  E_{q(\theta)}[F(\theta)] \\
  \approx \frac{1}{K} \sum_k F(\theta_k), \ \theta_k \sim q(\theta)
  \]

  Computed with Monte Carlo estimates

Only require fast sampling from \( q \)!
(no need for analytic form of the \( q \) distribution)

Li and Liu. Wild Variational Inference. AABI 2016
Huszár. Variational Inference using Implicit Distributions. arXiv 2017
Implicit Approximate Postersiors

• Two quantities computed in (approximate) Bayesian inference:

approximate Bayesian predictive

\[ p(y^*|x^*, D) \approx E_{q(\theta)}[p(y^*|x^*, \theta)] \]

\[ \approx \frac{1}{K} \sum_{k=1}^{K} p(y^*|x^*, \theta_k), \ \theta_k \sim q(\theta) \]

approximate posterior moments

\[ E_{q(\theta)}[F(\theta)] \]

\[ \approx \frac{1}{K} \sum_{k=1}^{K} F(\theta_k), \ \theta_k \sim q(\theta) \]

Computed with Monte Carlo estimates

Only require fast sampling from \( q \)!
(no need for analytic form of the \( q \) distribution)

Li and Liu. Wild Variational Inference. AABI 2016
Huszár. Variational Inference using Implicit Distributions. arXiv 2017

implicit distributions
Implicit Approximate Posteriors

\[ L(\phi) = E_{q(\theta)}[\log p(D|\theta)] - E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}] \]

estimated by Monte Carlo
intractable
(q density unknown)

Mescheder et al. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. ICML 2017
Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018
Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018
Implicit Approximate Posteriors

\[ L(\phi) = E_{q(\theta)}[\log p(D|\theta)] - E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}] \]

estimated by Monte Carlo

Approximated by using a discriminator (AVB):

\[ \theta \sim p(\theta) \]

\[ \theta \sim q(\theta) \]

\[ \theta \text{ sampled from } p \text{ or } q? \]

Mescheder et al. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. ICML 2017
Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018
Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018
Implicit Approximate Posteriors

\[
L(\phi) = E_{q(\theta)}[\log p(D|\theta)] - E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}]
\]

estimated by Monte Carlo

Approximated by using a discriminator (AVB):

\[\theta \sim p(\theta)\]

\[\theta \sim q(\theta)\]

\[\theta\] sampled from \(p\) or \(q\)?

Mescheder et al. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. ICML 2017


Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018
Objective Functions

For fitting the approximate posterior
VI Ingredients

\[ L = E_{\theta \sim q_{\phi}} \left[ \log \frac{p(D, \theta)}{q_{\phi}(\theta)} \right] = \log p(D) - KL[q_{\phi} \| p] \]

- **p**: your model design
- **q**: your choice of variational distribution, e.g. mean field, flow based
- **KL**: defines the algorithm

\[ q \in Q \]

\[ q^*(\theta) \]

\[ p(\theta | D) \]
Does It Work?

VI underestimate the uncertainty

(Rényi) \( \alpha \)-Divergence

\[ \alpha > 0, \alpha \neq 1 \]

\[ D_\alpha[p||q] = \frac{1}{\alpha - 1} \log \int p(\theta)^\alpha q^{1-\alpha} d\theta \]

\[ \alpha = 1 \]

\[ D_1[p||q] = \lim_{\alpha \to 1} D_\alpha(p||q) = KL(p||q) \]
VI with $\alpha$-Divergence

\[
L = E_{\theta \sim q_\phi} \left[ \log \frac{p(D, \theta)}{q_\phi(\theta)} \right] = \log p(D) - KL[ q_\phi || p ]
\]

Variational Rényi bound:

\[
L_\alpha = \frac{1}{1 - \alpha} E_{\theta \sim q_\phi} \left[ \left( \log \frac{p(D, \theta)}{q_\phi(\theta)} \right)^{1-\alpha} \right] = \log p(D) - D_\alpha[ q_\phi || p ]
\]

\[
\lim_{\alpha \to 1} L_\alpha = L
\]

Dieng et al. Variational Inference via $\chi$-Upper Bound Minimization. NeurIPS 2017
Does It Work?

Dieng et al. Variational Inference via $\chi$-Upper Bound Minimization. NeurIPS 2017
Does It Work?

How to choose alpha?

Dieng et al. Variational Inference via $\chi$-Upper Bound Minimization. NeurIPS 2017
Does It Work?

How to choose alpha?

\[
L_\alpha = \frac{1}{1-\alpha} E_{\theta \sim q_\phi} \left[ \left( \log \frac{p(D, \theta)}{q_\phi(\theta)} \right)^{1-\alpha} \right]
\]

Does It Work?

How to choose alpha?

\[ L_\alpha = \frac{1}{1-\alpha} E_{\theta \sim q_\phi} \left[ \left( \log \frac{p(D, \theta)}{q_\phi(\theta)} \right)^{1-\alpha} \right] \]

Too small or too big alpha leads to extremely big variances

Dieng et al. Variational Inference via $\chi$-Upper Bound Minimization. NeurIPS 2017
Revisiting Perturbation Theory for VI

\[ V(x, z) \equiv \log q_\lambda(z) - \log p(x, z) \]

\[ \log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} \left[ e^{-\beta V(x, z)} \right] \right) \mid_{\beta=1} \]

Bamler*, Zhang* et al. Perturbative Black Box Variational Inference. NeurIPS 2017
Revisiting Perturbation Theory for VI

$$V(x, z) \equiv \log q_\lambda(z) - \log p(x, z)$$

$$\log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} \left[ e^{-\beta V(x, z)} \right] \right) \mid \beta = 1$$

Taylor expansion around $\beta = 1$:

$$\log p(x) \approx E_{q_\lambda}[-V] + \frac{1}{2} \left[ (V - E_{q_\lambda}[-V])^2 \right] - \frac{1}{3!} \left[ (V - E_{q_\lambda}[-V])^3 \right] + \frac{1}{4!} \left[ (V - E_{q_\lambda}[-V])^4 \right] - \ldots$$

Bamler*, Zhang* et al. Perturbative Black Box Variational Inference. NeurIPS 2017
Revisiting Perturbation Theory for VI

\[ V(x, z) \equiv \log q_\lambda(z) - \log p(x, z) \]

\[ \log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} \left[ e^{-\beta V(x, z)} \right] \right) \bigg|_{\beta=1} \]

Taylor expansion around \( \beta = 1 \) :

\[ \log p(x) \approx \left[ E_{q_\lambda} [-V] \right] + \frac{1}{2} \left[ (V - E_{q_\lambda} [-V])^2 \right] - \frac{1}{3!} \left[ (V - E_{q_\lambda} [-V])^3 \right] \]

\[ + \frac{1}{4!} \left[ (V - E_{q_\lambda} [-V])^4 \right] - \ldots \]

\[ E_{q_\lambda} [-V(x, z)] = E_{q_\lambda} [\log p(x, z) - \log q_\lambda(z)] \]

Bamler*, Zhang* et al. Perturbative Black Box Variational Inference. NeurIPS 2017
Revisiting Perturbation Theory for VI

\[ V(x, z) \equiv \log q_\lambda(z) - \log p(x, z) \]

\[ \log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} \left[ e^{-\beta V(x, z)} \right] \right) \mid_{\beta=1} \]

Taylor expansion around \( \beta = 1 \):

\[ \log p(x) \approx E_{q_\lambda}[-V] + \frac{1}{2} \left[ (V - E_{q_\lambda}[-V])^2 \right] - \frac{1}{3!} \left[ (V - E_{q_\lambda}[-V])^3 \right] \\
+ \frac{1}{4!} \left[ (V - E_{q_\lambda}[-V])^4 \right] - \ldots \]

\[ E_{q_\lambda}[-V(x, z)] = E_{q_\lambda}[\log p(x, z) - \log q_\lambda(z)] \]

Truncation at any odd number term provides a bound.

Bamler*, Zhang* et al. Perturbative Black Box Variational Inference. NeurIPS 2017
Behaviour of PBBVI

- Better uncertainty estimation than KLVI
- Better bias-variance trade-off comparing to $\alpha$-VI

Bamler*, Zhang* et al. Perturbative Black Box Variational Inference. NeurIPS 2017
Behaviour of PBBVI

- Better uncertainty estimation than KLVI
- Better bias-variance trade-off comparing to $\alpha$-VI

Where do we truncate? Is it flexible enough?

Bamler*, Zhang* et al. Perturbative Black Box Variational Inference. NeurIPS 2017
F-Divergence

\[ D_f[p||q_\phi] = E_{\theta \sim q_\phi} [f \left( \frac{p(\theta)}{q_\phi(\theta)} \right) - f(1)] \]
F-Divergence

\[ D_f[p||q_\phi] = E_{\theta \sim q_\phi}[f\left(\frac{p(\theta)}{q_\phi(\theta)}\right) - f(1)] \]

- \( f(t) = -\log t \) \( \implies KL(q||p) \)
- \( f(t) = t\log t \) \( \implies KL(p||q) \)
- \( f(t) = \frac{t^\alpha}{\alpha(\alpha - 1)} \) \( \implies D_\alpha(p||q) \)
Integral Probability Metric (IPM)

- Using a test function to describe difference:

\[ D[q(z), p(z|x)] = \sup_{f \in F} |E_{q(z)}[f(z)] - E_{p(z|x)}[f(z)]| \]

Figure adapted, source: Dougal Sutherland


Integral Probability Metric (IPM)

- Using a test function to describe difference:

\[ D[q(z), p(z|x)] = \sup_{f \in F} |E_{q(z)}[f(z)] - E_{p(z|x)}[f(z)]| \]

- Stein discrepancy: only requires \( z \sim q(z) \) and
\[ \nabla_z \log p(z|x) = \nabla_z \log p(z, x) \]

\[ S[q(z), p(z|x)] = \sup_{f \in F_q} |E_{q(z)}[\nabla_z \log p(z, x)^T f(z) + \nabla_z^T f(z)]| \]

Figure adapted, source: Dougal Sutherland

Looking Back

VI with IPM

VI with $\alpha$ divergence

VI with $f$ divergence

Perturbative VI

ELBO with KL

VI with Stein discrepancy
How to Choose the Inference Algorithm?

Choose divergence by meta-learning!

Zhang et al. Meta-Learning for Variational Inference. AABI 2019
Improved Monte Carlo Bounds

• Importance weighted auto-encoder (IWAE) bound:

\[
L_K(\phi) = E_{z_1, \ldots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)} \right]
\]

Importance sampling estimate of \(p(x)\)

Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019
Improved Monte Carlo Bounds

• Importance weighted auto-encoder (IWAE) bound:

\[ L_K(\phi) = E_{z_1, \ldots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)} \right] \]

Importance sampling estimate of \( p(x) \)

\[ \log p(x) \]

\[ L(\phi) \quad (K = 1) \]

Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019
Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

\[
L_K(\phi) = E_{z_1, \ldots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)} \right]
\]

Importance sampling estimate of \( p(x) \)

\[
L_K'(\phi) \quad (K = k' \geq k)
\]

\[
L_K(\phi) \quad (K = k > 1)
\]

\[
L(\phi) \quad (K = 1)
\]

increase \( K \)

Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019
Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

\[ L_K(\phi) = E_{z_1,\ldots,z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)} \right] \]

Importance sampling estimate of \( p(x) \)

- \( \log p(x) \) (\( K \to \infty \))
- \( L_{k'}(\phi) \) (\( K = k' \geq k \))
- \( L_k(\phi) \) (\( K = k > 1 \))
- \( L(\phi) \) (\( K = 1 \))

Increase \( K \)

Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019
**Improved Monte Carlo Bounds**

- Importance weighted auto-encoder (IWAE) bound:

\[
L_K(\phi) = E_{z_1, \ldots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)} \right]
\]

Importance sampling estimate of \(p(x)\)

\[
L_k'(\phi) \quad (K = k' \geq k)
\]

\[
L_k(\phi) \quad (K = k > 1)
\]

\[
L(\phi) \quad (K = 1)
\]

**Optimize** \(q(z) \rightarrow p(z|x)\) to close the gap

---

Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019
Improved Monte Carlo Bounds

• Constructing lower-bounds from an estimator $R$ of the marginal:

\[
E_{q(h)}[R(h, x)] = p(x) \Rightarrow E_{q(h)}[\log R(h, x)] \leq \log p(x)
\]

Jensen’s inequality
Improved Monte Carlo Bounds

• Constructing lower-bounds from an estimator \( R \) of the marginal:

\[
E_{q(h)}[R(h, x)] = p(x) \implies E_{q(h)}[\log R(h, x)] \leq \log p(x)
\]

  • Variational lower-bound: \( h = z \), \( R(z, x) = \frac{p(x, z)}{q(z)} \)
  • IWAE bound: \( h = (z_1, \ldots, z_K) \), \( R(h, x) = \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)} \)

Jensen’s inequality

Improved Monte Carlo Bounds

• Constructing lower-bounds from an estimator $R$ of the marginal:

$$E_{q(h)}[R(h, x)] = p(x) \quad \Rightarrow \quad E_{q(h)}[\log R(h, x)] \leq \log p(x)$$

- Variational lower-bound: $h = z$, $R(z, x) = \frac{p(x, z)}{q(z)}$
- IWAE bound: $h = (z_1, ..., z_K)$, $R(h, x) = \frac{1}{K} \sum_{k=1}^{K} \frac{p(x, z_k)}{q(z_k)}$

• Fit $q$ using existing Monte Carlo estimators of $p(x)$
  - Example: antithetic sampling with Gaussian $q(z)$:

$$R(z, x) = \frac{p(x, z) + p(x, T(z))}{2q(z)} \quad T(z) = \mu_q - (z - \mu_q)$$
Improved Monte Carlo Bounds

• Constructing lower-bounds from an estimator $R$ of the marginal:

$$E_{q(h)}[R(h, x)] = p(x) \Rightarrow E_{q(h)}[\log R(h, x)] \leq \log p(x)$$

  • Variational lower-bound: $h = z, R(z, x) = \frac{p(x,z)}{q(z)}$
  • IWAE bound: $h = (z_1, ..., z_K), R(h, x) = \frac{1}{K} \sum_{k=1}^{K} \frac{p(x,z_k)}{q(z_k)}$

• Fit $q$ using existing Monte Carlo estimators of $p(x)$
  • Example: antithetic sampling with Gaussian $q(z)$:

$$R(z, x) = \frac{p(x,z)+p(x,T(z))}{2q(z)}, \quad T(z) = \mu_q - (z - \mu_q)$$

Free-energy as an Objective

• Bethe free-energy & message passing:

\[ \mu_{z \rightarrow f}(z) \quad \mu_{f \rightarrow z}(z) \]

• Both \( q \) and the inference algorithm are defined by the factor graph

• Optimal \( q \) achieved at the fixed point of the Bethe free energy

Li and Turner. A Unifying Approximate Inference Framework from Variational Free Energy Relaxation. AABI 2016
Landscape of Advances

e.g. mean-field: $q(\theta) = \prod_i q(\theta_i)$

variational lower-bound:
$L(\phi) = E_{q(\theta)}[\log p(D|\theta)] - KL[q(\theta)\|p(\theta)]$
Landscape of Advances

mini-batch training:
\[ \log p(D|\theta) \approx \frac{N}{M} \log p(B|\theta), B \sim D^M \]
Landscape of Advances

Monte Carlo estimates require fast sampling of $\theta \sim q(\theta)$ to enable estimation of $\theta \sim q(\theta)$. The objective design estimates $\theta \sim q(\theta)$, and stochastic optimization relies on $\log p(D|\theta) \approx \frac{N}{M} \log p(B|\theta), B \sim D^M$. Scale-up facilitates the process.
Landscape of Advances

REINFORCE gradient:
\[ \nabla \phi E_{q_{\phi}(\theta)}[F(\theta)] = E_{q_{\phi}(\theta)}[F(\theta)\nabla \phi \log q_{\phi}(\theta)] \]

gradient estimators

\[
\nabla \phi E_{q_{\phi}(\theta)}[F(\theta)] = E_p(\epsilon) [\nabla \phi F(g_{\phi}(\epsilon))] 
\]

stochastic optimization

scale-up

Monte Carlo estimates

\[ \theta \sim q(\theta) \]

estimate

\[ \text{(another type of MC estimate)} \]

\[ \text{enable} \]

\[ \text{estimate} \]

\[ \text{scale-up} \]
Landscape of Advances

- Monte Carlo estimates
  - Gradient estimators
  - Stochastic optimization
    - Variance reduction
      - Control variates
        - Gumbel-softmax trick
    - Scale-up
  - Estimate
    - (Another type of variance reduction)
- Objective design
  - Estimate
  - (Another type of MC estimate)
- q design
  - Enable
  - Scale-up
Landscape of Advances

- **amortised inference**
  - $q(z) := q(z|x)$ (dist. params. of $q$ depend on $x$)
  - $z \sim q(z|x)$
  - enable
  - estimate

- **Monte Carlo estimates**
  - (for end-to-end training) utilise
  - estimate

- **gradient estimators**
  - utilise
  - estimate

- **objective design**
  - (another type of MC estimate)

- **stochastic optimization**
  - (another type of variance reduction)

- **reduce variance**

- **variance reduction**

$q(z) := q(z|x)$ (dist. params. of $q$ depend on $x$)
Landscape of Advances

Monte Carlo estimates

- amortised inference
- q design

- utilise
- enable

gradient estimators

- estimate

variance reduction

- scale-up

stochastic optimization

- (another type of variance reduction)

objective design

- estimate

α-divergences
f-divergences
IPMs
Free-energy

structured approx.
normalizing flow
hierarchical mixture
implicit distribution

reduce variance
Landscape of Advances

amortised inference → utilise → gradient estimators

utilise

Monte Carlo estimates → estimate → stochastic optimization

to variance reduction

reduce variance

(can another type of variance reduction)

objective design

scale-up

can be (de-)coupled (e.g. auxiliary lower-bound for mixture \( q \), Bethe free-energy for message passing)

\( q \) design

enable

scale-up
Monte Carlo estimates

- amortised inference
- $q$ design
- objective design
- stochastic optimization

- utilise
- estimate
- scale-up
- reduce variance
- variance reduction

- can be (de-)coupled
Part III: Applications

• Bayesian neural networks
• Generative models for decision making
• Future directions
Why Estimating Uncertainty in DL?

• Models are often over-parameterised
  • E.g. BERT, GPT-3 in NLP
  • E.g. ResNet-152 for vision tasks
Why Estimating Uncertainty in DL?

• Models are often over-parameterised
  • E.g. BERT, GPT-3 in NLP
  • E.g. ResNet-152 for vision tasks
Why Estimating Uncertainty in DL?

- Models are often over-parameterised
  - E.g. BERT, GPT-3 in NLP
  - E.g. ResNet-152 for vision tasks
- Multiple parameter settings can fit the same data
  - They might provide different predictions on test data
Why Estimating Uncertainty in DL?

- Critical tasks need uncertainty estimates to assist decision making
  - Inform end users when uncertain, for safe decision making

Healthcare AI
Why Estimating Uncertainty in DL?

- Critical tasks need uncertainty estimates to assist decision making
  - Inform end users when uncertain, for safe decision making

Healthcare AI

Autonomous driving
Bayesian Neural Network (BNN) 101

Classifying different types of animals:
- \( x \): input image; \( y \): output label
- Build a neural network with parameters \( \theta \):
  \[
p(y|x, \theta) = softmax(f_\theta(x))
\]
Bayesian Neural Network (BNN) 101

Classifying different types of animals:
- $x$: input image; $y$: output label
- Build a neural network with parameters $\theta$:
  \[ p(y|x, \theta) = softmax(f_\theta(x)) \]

A typical neural network (with non-linearity $g(\cdot)$):

\[
f_\theta(x) = W^L g(W^{L-1} g(... g(W^1 x + b^1) + b^{L-1}) + b^{L-1}) + b^L,
\]

\[
h^l = g(W^l h^{l-1} + b^l), h^1 = g(W^1 x + b^1).
\]

Neural network parameters: $\theta = \{W^l, b^l\}_{l=1}^L$
Bayesian Neural Network (BNN) 101

Classifying different types of animals:
• $x$: input image; $y$: output label
• Build a neural network with parameters $\theta$:
  
  \[ p(y|x, \theta) = \text{softmax}(f_\theta(x)) \]

Typical deep learning solution:
• Optimize $\theta$ to obtain a point estimates (MLE):

  \[ \theta^* = \arg \max \log p(D | \theta), \]
  \[ \log p(D | \theta) = \sum_{n=1}^{N} \log p(y_n | x_n, \theta), \]
  \[ D = \{(x_n, y_n)\}_{n=1}^{N} \]

• Prediction: using $p(y^* | x^*, \theta^*)$
Bayesian Neural Network (BNN) 101

Classifying different types of animals:
• $x$: input image; $y$: output label
• Build a neural network with parameters $\theta$:
  
  \[
  p(y | x, \theta) = \text{softmax}(f_\theta(x))
  \]

Bayesian solution:
• Put a prior $p(\theta)$ on network parameters $\theta$, e.g. Gaussian prior
  
  \[
  p(\theta) = N(\theta; 0, \sigma^2 I)
  \]
• Compute the posterior distribution $p(\theta | D)$:
  
  \[
  p(\theta | D) \propto p(D | \theta) p(\theta)
  \]
• Bayesian predictive inference:
  
  \[
  p(y^* | x^*, D) = E_{p(\theta | D)}[p(y^* | x^*, \theta)]
  \]
Bayesian Neural Network (BNN) 101

Classifying different types of animals:
• $x$: input image; $y$: output label
• Build a neural network with parameters $\theta$:
  $$p(y|x, \theta) = \text{softmax}(f_\theta(x))$$

Approximate (Bayesian) inference solution:
• Exact posterior intractable, use approximate posterior:
  $$q(\theta) \approx p(\theta | D)$$
• Approximate Bayesian predictive inference:
  $$p(y^* | x^*, D) \approx E_{q(\theta)}[p(y^* | x^*, \theta)]$$
• Monte Carlo approximation:
  $$p(y^* | x^*, D) \approx \frac{1}{K} \sum_{k=1}^{K} p(y^* | x^*, \theta_k), \quad \theta_k \sim q(\theta)$$
Bayesian Neural Network (BNN) 101

Prediction on in-distribution data: ensemble over networks, using weights sampled from $q(\theta)$
Bayesian Neural Network (BNN) 101

Prediction on OOD/noisy/adversarial data:
Disagreement (i.e. uncertainty) exists over networks sampled from $q(\theta)$
Bayesian Neural Network (BNN) 101

Prediction on OOD/noisy/adversarial data when \( q(\theta) \) is over-confident:
Return confidently wrong answers (close to point estimate)
Bayesian Neural Network (BNN) 101

Prediction on in-distribution data when $q(\theta)$ is under-confident:
Low accuracy in prediction tasks (less desirable)
Approximate Inference in BNNs

• Key steps of approximate inference in BNNs
  1. Construct the $q(\theta) \approx p(\theta \mid D)$ distribution
     • Simple distributions: e.g. Mean-field Gaussian
     • Structured approximations, e.g. low-rank Gaussians
     • Others (non-Gaussian)
Approximate Inference in BNNs

• Key steps of approximate inference in BNNs

  1. Construct the $q(\theta) \approx p(\theta \mid D)$ distribution
     • Simple distributions: e.g. Mean-field Gaussian
     • Structured approximations, e.g. low-rank Gaussians
     • Others (non-Gaussian)

  2. Fit the $q(\theta)$ distribution
     • E.g. with variational inference
Approximate Inference in BNNs

- Key steps of approximate inference in BNNs
  1. Construct the $q(\theta) \approx p(\theta | D)$ distribution
     - Simple distributions: e.g. Mean-field Gaussian
     - Structured approximations, e.g. low-rank Gaussians
     - Others (non-Gaussian)
  2. Fit the $q(\theta)$ distribution
     - E.g. with variational inference
  3. Compute prediction with Monte Carlo approximations
Approximate Inference in BNNs

• Step 1: construct the $q(\theta) \approx p(\theta \mid D)$ distribution
  • Example: Mean-field Gaussian distribution:

$$q(\theta) = \prod_{l=1}^{L} q(W^l) q(b^l)$$

$$q(W_i) = \prod_{ij} q(W_{ij}^l), \quad q(W_{ij}^l) = N(W_{ij}^l; M_{ij}^l, V_{ij}^l)$$

$$q(b^l) = \prod_i q(b_i^l), \quad q(b_i^l) = N(b_i^l; m_i^l, v_i^l)$$

• Variational parameters: $\phi = \{M_{ij}^l, \log V_{ij}^l, m_i^l, \log v_i^l\}_{l=1}^{L}$

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015
Approximate Inference in BNNs

• Step 2: fit the $q(\theta)$ distribution:
  • Variational inference: $\phi^* = \arg\max L(\phi)$
    \[
    L(\phi) = E_{q(\theta)}[\log p(D | \theta)] - KL[q(\theta) || p(\theta)]
    \]
Approximate Inference in BNNs

• Step 2: fit the $q(\theta)$ distribution:
  • Variational inference: $\phi^* = \arg\max L(\phi)$
    $$L(\phi) = \mathbb{E}_{q(\theta)}[\log p(D \mid \theta)] - KL[q(\theta) \mid \mid p(\theta)]$$
  • First scalable technique: **Stochastic optimization**
    • i.i.d. assumption of data: $\log p(D \mid \theta) = \sum_{n=1}^{N} \log p(y_n \mid x_n, \theta)$
    • Enable mini-batch training with $\{(x_m, y_m)\} \sim D^M$:
      $$L(\phi) \approx \frac{N}{M} \sum_{m=1}^{M} E_{q(\theta)}[\log p(y_m \mid x_m, \theta)] - KL[q(\theta) \mid \mid p(\theta)]$$

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015
Approximate Inference in BNNs

• Step 2: fit the $q(\theta)$ distribution:
  • Variational inference: $\phi^* = \text{argmax } L(\phi)\n  L(\phi) = \mathbb{E}_{q(\theta)}[\log p(D | \theta)] - KL[q(\theta) || p(\theta)]\n  • First scalable technique: \textbf{Stochastic optimization}\n    • i.i.d. assumption of data: $\log p(D | \theta) = \sum_{n=1}^{N} \log p(y_n | x_n, \theta)$\n    • Enable mini-batch training with $\{(x_m, y_m)\} \sim D^M$:
      \[
      L(\phi) \approx \frac{N}{M} \sum_{m=1}^{M} E_{q(\theta)}[\log p(y_m | x_m, \theta)] - KL[q(\theta) || p(\theta)]
      \]

  reweighting to ensure calibrated posterior concentration
Approximate Inference in BNNs

• Step 2: fit the $q(\theta)$ distribution:
  • 2nd scalable technique: Monte Carlo sampling
  • $E_{q(\theta)}[\log p(y \mid x, \theta)]$ intractable even with Gaussian $q(\theta)$
  • Solution: Monte Carlo estimate:

$$E_{q(\theta)}[\log p(y \mid x, \theta)] \approx \frac{1}{K} \sum_{k=1}^{K} \log p(y \mid x, \theta_k), \quad \theta_k \sim q(\theta)$$

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015
Approximate Inference in BNNs

• Step 2: fit the \( q(\theta) \) distribution:
  • 2nd scalable technique: Monte Carlo sampling
    • \( E_{q(\theta)}[\log p(y | x, \theta)] \) intractable even with Gaussian \( q(\theta) \)
    • Solution: Monte Carlo estimate:
      \[
      E_{q(\theta)}[\log p(y | x, \theta)] \approx \frac{1}{K} \sum_{k} \log p(y | x, \theta_k), \quad \theta_k \sim q(\theta)
      \]
  • Reparameterization trick to sample mean-field Gaussians:
    \( \theta_k \sim q(\theta) \iff \theta_k = m_\theta + \sigma_\theta \epsilon_k, \ \epsilon_k \sim N(0, I) \)

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015
Approximate Inference in BNNs

• Step 2: fit the $q(\theta)$ distribution:
  • 2nd scalable technique: Monte Carlo sampling
    • $E_{q(\theta)}[\log p(y \mid x, \theta)]$ intractable even with Gaussian $q(\theta)$
    • Solution: Monte Carlo estimate:
      \[
      E_{q(\theta)}[\log p(y \mid x, \theta)] \approx \frac{1}{K} \sum_{k=1}^{K} \log p(y \mid x, \theta_k), \quad \theta_k \sim q(\theta)
      \]
  • Reparameterization trick to sample mean-field Gaussians:
    \[
    \theta_k \sim q(\theta) \iff \theta_k = m_\theta + \sigma_\theta \epsilon_k, \quad \epsilon_k \sim N(0, I)
    \]
    \[
    \Rightarrow E_{q(\theta)}[\log p(y \mid x, \theta)] \approx \frac{1}{K} \sum_{k=1}^{K} \log p(y \mid x, m_\theta + \sigma_\theta \epsilon_k), \epsilon_k \sim N(0, I)
    \]

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015
Approximate Inference in BNNs

• Combining both steps:

\[
L(\phi) \approx \frac{N}{M} \sum_{m=1}^{M} \frac{1}{K} \sum_{k=1}^{K} \log p(y_m | x_m, \theta_k) - KL[q(\theta) \| p(\theta)], \theta_k \sim q(\theta)
\]

analytic between two Gaussians
(if not, can also be estimated with Monte Carlo)

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015
Approximate Inference in BNNs

• Step 3: compute prediction with Monte Carlo approximations:

\[ p(y^* \mid x^*, D) \approx \frac{1}{K} \sum_{k=1}^{K} p(y^* \mid x^*, \theta_k), \quad \theta_k \sim q(\theta) \]

Mean-field Gaussian case:
\[ \theta_k = m_{\theta} + \sigma_{\theta} \epsilon_k, \quad \epsilon_k \sim N(0, I) \]
Applications of BNNs: Image Segmentation

Kendall and Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? NeurIPS 2017
Applications of BNNs: Super Resolution

Applications of BNNs: Continual Learning

\[ L_{VCL}^{t}(q_t(\theta)) = E_{q_t(\theta)}[\log p(D_t \mid \theta)] - KL[q_t(\theta) \mid \mid q_{t-1}(\theta)] \]

Nguyen et al. Variational Continual Learning. ICLR 2018
Recent Progress in BNNs: Inference

\[
\text{SGD: } \quad \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t)
\]

\[
\text{SGLD: } \quad \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta} \epsilon, \quad \epsilon \sim N(0, I)
\]

Stochastic gradient MCMC

Li et al. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. AAAI 2016
Zhang et al. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. ICLR 2020
Recent Progress in BNNs: Inference

SGD: \[ \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) \]

SGLD: \[ \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta} \epsilon, \quad \epsilon \sim N(0,I) \]

Stochastic gradient MCMC

Monte Carlo dropout

Recent Progress in BNNs: Inference

**SGD:** \[ \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \bar{U}(\theta_t) \]

**SGLD:** \[ \begin{align*}
    \theta_{t+1} &= \theta_t - \eta \nabla_{\theta_t} \bar{U}(\theta_t) + \sqrt{2\eta} \epsilon, \\
    \epsilon &\sim N(0, I)
\end{align*} \]

Stochastic gradient MCMC

Monte Carlo dropout

Deterministic approximations

Hernandez-Lobato and Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. ICML 2015

Wu et al. Deterministic Variational Inference for Robust Bayesian Neural Networks. ICLR 2019
Recent Progress in BNNs: Inference

SGD: \[ \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) \]

SGLD: \[ \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta} \epsilon, \quad \epsilon \sim N(0, I) \]

Stochastic gradient MCMC

Deterministic approximations

Monte Carlo dropout

Function space approximate inference

Ma et al. Variational Implicit Processes. ICML 2019
Sun et al. Functional Variational Bayesian Neural Networks. ICLR 2019
Recent Progress in BNNs: Theory

Connections to GPs:
- BNN with very wide hidden layers
  ≈ Gaussian process
- Width limit convergence: in both prior (Neal’s result) and posterior

Matthews et al. Gaussian Process Behaviour in Wide Deep Neural Networks. ICLR 2018
Lee et al. Deep Neural Networks as Gaussian Processes. ICLR 2018
Hron et al. Exact posterior distributions of wide Bayesian neural networks. 2020
Recent Progress in BNNs: Theory

Connections to GPs:
• BNN with very wide hidden layers \approx Gaussian process
• Width limit convergence: in both prior (Neal’s result) and posterior

Approx. vs exact inference:
• Theoretical limitation of MFVI in shallow BNNs with ReLU activations
• Empirically deep BNNs with MVFI still fails in certain cases

Foong et al. On the Expressiveness of Approximate Inference in Bayesian Neural Networks. NeurIPS 2020
Farquhar et al. Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations. NeurIPS 2020
Dynamic Information Acquisition

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
Dynamic Information Acquisition

Neu: Neuropathic pain
Noc: Nociceptive pain
Pys: Psychiatric pain

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
Dynamic Information Acquisition

Step 1: Missing Value Prediction with Deep Generative Model

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
Dynamic Information Acquisition

- Step 1: Missing Value Prediction with Deep Generative Model
- Step 2: Active element-wise information acquisition

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
A Deep Generative Model

Variational Auto-encoder (VAE)

$$L_{amortized} = \log p(x) - KL (q(z|x) | p(z|x))$$
Challenges

- Utilizing Deep Learning for scalable inference
- Cannot handle partial observation

\[ i \in [1, M] \]

Encoder/ inference network

Decoder/ generator
VAE to Partial VAE

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
We aim to infer the missing values $X_U$ from the observed values $X_O$.

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
$L_{amortized} = \log p(x) - KL[q(z|x)||p(z|x)]$

$= E_{z \sim q(z|x)} [\log p_\theta(x|z)] - KL[q(z|x)||p(z)]$

$= E_{z \sim q(z|x)} [\log p_\theta(x|z)] - KL[q(z|x)||p(z)]$

$L_{amortized} = \log p(x_o) - KL[q(z|x_o)||p(z|x_o)]$

$= E_{z \sim q(z|x_o)} [\log p_\theta(x|z)] - KL[q(z|x_o)||p(z)]$

The ELBO still holds.
The challenge is how to design an inference net.

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
VAE to Partial VAE

Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR 2017
VAE to Partial VAE

\[ c(x_0) := g(\vec{h}(s_1), \vec{h}(s_2), ..., \vec{h}(s_O)) \]

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
Dynamic Information Acquisition

- Inflammation
- Lower Back Pain
- Have pats

**Step 1:** Missing Value Prediction with Deep Generative Model

**Step 2:** Active element-wise information acquisition

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
Actively Select the Next Variable

\[ i^* = \arg \max_{i \in U} R(i|\mathbf{x}_0) \]
Actively Select the Next Variable

\[ R(i, x_0) = E_{x_i \sim p(x_i|x_0)} [KL[p(x_\phi|x_i, x_0) || p(x_\phi|x_0)]] \]

\[ i^* = \arg \max_{i \in U} R(i|x_0) \]
Actively Select the Next Variable

\[ R(i, x_o) = E_{x_i \sim p(x_i | x_o)}[KL[p(x_\phi | x_i, x_o) \| p(x_\phi | x_o)]] \]

Efficient Approximation

\[ i^* = \arg \max_{i \in U} R(i | x_o) \]

Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019
Predicting a House’s Price

Kim (customer)

How far away is the supermarket?

Ty (broker)
Our solution

Baseline

**TARGET VARIABLE**
median value of owner-occupied homes in $1000's

**QUESTION 1**
index of accessibility to racial highways

1.0000

weighted distances to five Boston employment centres

36.55

**TARGET VARIABLE**
median value of owner-occupied homes in $1000's

**QUESTION 1**
weighted distances to five Boston employment centres

40.66
Our solution

Baseline

Model prediction based on the current information
### Our solution

**Baseline**

**Target Value / Ground Truth**

- **Target Variable**: Median value of owner-occupied homes in $1000's
- **Baseline**: Median value of owner-occupied homes in $1000's

#### Question 1

- **Index of accessibility to radial highways**: 18.00, Target: 5.00

- **Full-value property-tax rate per $10,000**: 1.00, Target: 3.00

- **Proportion of non-retail business acres per town**: 3.90, Target: 2.00

- **Pupil-teacher ratio by town**: 10.00, Target: 8.00

- **% Lower status of the population**: 10.00, Target: 20.00

- **Average number of rooms per dwelling**: 10.00, Target: 6.00

- **Per capita crime rate by town**: 20.00, Target: 30.00

- **Proportion of residential land zoned for lots over 25,000 sq ft**: 10.00, Target: 5.00

- **Charles river dummy variable (true if tract bounds river false otherwise)**: 0.00, Target: 1.00

- **Nitrous oxides concentration (parts per 10 million)**: 1.00, Target: 2.00

- **Proportion of owner-occupied units built prior to 1940**: 20.00, Target: 10.00

- **Weighted distances to five Boston employment centres**: 40.00, Target: 30.00

**Model prediction based on the current information**
List of questions that we could ask
Our solution

List of questions that we could ask

TARGET VARIABLE
median value of owner-occupied homes in $1000's

Baseline
TARGET VARIABLE
median value of owner-occupied homes in $1000's

QUESTION 1
weighted distances to five Boston employment centres

Information reward
- List of questions from survey (e.g. US veteran) for mental health monitoring
- List of technical questions from interview for recruiting
- List of medical tests for diagnosis
- ...
Does it work when there is few training data?
Partial Amortized Bayesian Deep Latent Gaussian Model (PA-BELGAM)

- Point estimate of global parameter $\theta$

$$p(x_o, z) = \prod_{i=1}^{N} \prod_{d \in \mathcal{O}_i} p(x_{i,d}|z_i)p(z_i)$$

- Stochastic variable $\theta$

$$p(x_o, \theta, z) = p(\theta) \prod_{i=1}^{N} \prod_{d \in \mathcal{O}_i} p(x_{i,d}|z_i, \theta)p(z_i)$$

Gong et al. Icebreaker: Element-wise efficient information acquisition with a Bayesian deep latent gaussian model. NeurIPS 2019
Partial Amortized Bayesian Deep Latent Gaussian Model (PA-BELGAM)

\[ q(\theta, z | x_o) \approx q(\theta | x_o)q_\phi(z | x_o) \]

Gong et al. Icebreaker: Element-wise efficient information acquisition with a Bayesian deep latent gaussian model. NeurIPS 2019
When Data are Heterogeneous

Ma et.al. VAEM: a Deep Generative Model for Heterogeneous Mixed Type Data. NeurIPS 2020
When Robustness is Needed

Causal Reasoning

Deep Casual Manipulation Augmented Model

Zhang et.al. A Causal View on Robustness of Neural Networks. NeurIPS 2020
The probability distribution $\pi(\theta)$ is intractable.

\[ \int F(\theta) \pi(\theta) d\theta \]
Summary

The probability distribution $\pi(\theta)$ is intractable.

Approximate Inference

$$\int F(\theta) \pi(\theta) d\theta$$
The probability distribution $\pi(\theta)$ is intractable.
Future Directions: Methodology

Better optimization

Zhang et al. Noisy natural gradient as variational inference. ICML 2018
Khan et al. Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam. ICML 2018
Future Directions: Methodology

Better optimization

- $q$ distribution design
- Objective design
- Amortised inference
- Scalable inference

... 

Combined approaches

- Rejection sampling
- Importance sampling
- SMC, MCMC, Quasi MC

... 

$q$ distribution design

Objective design

Amortised inference

Scalable inference

... 

Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Future Directions: Methodology

Better optimization

- $q$ distribution design
- Objective design
- Amortised inference
- Scalable inference

Combined approaches

- Rejection sampling
- Importance sampling
- SMC, MCMC, Quasi MC

Meta-learning inference algorithms

Gong et al. Meta-learning for Stochastic Gradient MCMC. ICLR 2019
Zhang et al. Meta-Learning for Variational Inference. AABI 2019
Future Directions: Error Analyses

Errors in inference

\[ D[q(\theta)||p(\theta|D)] = ? \]

\[ D[q(y^*|x^*)||p(y^*|x^*, D)] = ? \]

\[ = \int p(y^*|x^*, \theta) q(\theta) d\theta \]

Analysis needed for deep probabilistic models!
- Optimization error
- Approximation gap

Foong et al. On the Expressiveness of Approximate Inference in Bayesian Neural Networks. NeurIPS 2020
Future Directions: Error Analyses

Errors in inference

\[ D[q(\theta)||p(\theta|D)] = ? \]

\[ D[q(y^*|x^*)||p(y^*|x^*, D)] = ? \]

\[ = \int p(y^*|x^*, \theta) q(\theta) d\theta \]

Analysis needed for deep probabilistic models!
- Optimization error
- Approximation gap

Model misspecification

Future Directions: Error Analyses

Errors in inference

\[ D[q(\theta)||p(\theta|D)] = ? \]
\[ D[q(\gamma^*|x^*)||p(\gamma^*|x^*, D)] = ? \]
\[ = \int p(y^*|x^*, \theta) q(\theta) d\theta \]

Analysis needed for deep probabilistic models!
- Optimization error
- Approximation gap

Model misspecification

[Diagram showing causal graph and probabilistic graphical model with support of the prior distribution on possible \( \theta \)]

Separation of inference & modelling?

Future Directions: Applications

Uncertainty estimation

$X_i = ?$

$X_i = !$
Future Directions: Applications

Uncertainty estimation

Model selection & averaging

Future Directions: Applications

Uncertainty estimation

Causal reasoning

"what if the patient was treated with drug B?"

Model selection & averaging
Thank You!

Questions? Ask at:
liyzhen2@gmail.com (Yingzhen Li)
Cheng.Zhang@microsoft.com (Cheng Zhang)