Amortized Inference With Implicit Models

Qiang Liu
Dartmouth College

August 12, 2017
Approximate Inference (or Sampling)

**Problem:** Given a distribution $p$, draw sample $\{x_i\}_{i=1}^n \sim p$.

**Assumption:** $p$ is defined through an un-normalized density function:

\[ p(x) = \frac{\bar{p}(x)}{Z}, \quad Z = \int \bar{p}(x) dx. \]

- Widely appears in: Bayesian inference, learning latent variable models, graphical models, etc.
- Intractable to draw examples exactly.
- Approximation methods: Markov chain Monte Carlo, variational inference, etc.
- **This talk**: We need to sample lots of similar distributions.
- **Applications**: Reasoning with *lots of datasets, users, objects*: meta-learning, personalized prediction, streaming inference, etc.
- *Inference as inner loops of learning*: variational auto-encoders, learning un-normalized energy models, graphical models, etc.
- **Other**: *reinforcement learning*, probabilistic programming, etc.
Amortized Inference

Replace the expert-designed, hand-crafted inference methods (e.g., MCMC), with adaptively trained simulators (e.g., neural networks).
Problem Definition

- Given: A set of distributions $\mathcal{P} = \{p(z)\}$.
  A class of simulators $G_\eta(\xi; p)$.
  - $\eta$: parameter to be decided;
  - $\xi$: random seed from a fixed, but perhaps unknown distribution.

- Goal: Find optimal parameter $\eta$, such that the distribution of the output $z = G_\eta(\xi; p)$ is close to $p(z)$.

“Neural random number generator”
Problem Definition

- Given: A set of distributions $\mathcal{P} = \{p(z)\}$.
  A class of simulators $G_\eta(\xi; p)$.
- $\eta$: parameter to be decided;
  $\xi$: random seed from a fixed, but perhaps unknown distribution.

- Goal: Find optimal parameter $\eta$, such that the distribution of the output $z = G_\eta(\xi; p)$ is close to $p(z)$.

This talk: matching a **distribution**

GAN: matching an observed **dataset**
Variational Autoencoder

- Given observed \( \{x_{obs,i}\} \), learn latent variable model:

\[
p_\theta(x) = \int p_\theta(x, z)dz.
\]

- \( x \): observed variable;
- \( z \): missing variable;
- \( \theta \): model parameter.

- Maximum likelihood estimate of \( \theta \) by EM.

- **Difficulty**: Need to sample from the posterior distribution \( p_\theta(z|x_{obs,i}) \) at each iteration, for each \( x_{obs,i} \).

- **Amortized inference**: Construct an “encoder”: \( z = G_\eta(\xi, x) \), such that \( z \sim p_\theta(z|x) \) [Kingma, Welling 13].
Given observed \( \{ x_{\text{obs},i} \}_{i=1}^{n} \), want to learn energy-based model:

\[
p_{\theta}(x) = \frac{1}{Z} \exp(\psi_{\theta}(x)),
\]

\( \psi_{\theta}(x) \): a neural net.

\( Z_{\theta} \): normalization constant.

- Classical method: estimating \( \theta \) by maximum likelihood.

- **Difficulty**: \( \log Z_{\theta} \) is intractable; requires to sample from \( p_{\theta} \) at every iteration to approximate the gradient.

- **Amortized inference**: Amortizing the generation of the negative samples yields GAN-style algorithms [Kim & Bengio16, Liu+16, Zhai+16].
Meta-Learning for Speeding up Bayesian Inference

- Bayesian inference: given data $D$, and unknown random parameter $z$, sample posterior $p(z|D)$.

- Traditional MCMC: can be viewed as hand-crafted simulators $G_\eta$, with hyper-parameter $\eta$.

- **Amortized inference**: can be used to optimize the hyper-parameters of MCMC, adaptively improving the performance when processing lots of similar datasets.
Reinforcement Learning with Deep Energy-base Policies [Haarnoja+ 17]

- Maximum entropy policy: \( p_\theta(a|s) \propto \exp\left(\frac{1}{\alpha} Q(s,a)\right) \).

- Implementing the policy requires drawing samples from \( p_\theta(a|s) \) repeatedly, at each iteration.

- **Amortized Inference**: construct generator \( G_\eta(\xi) \) (an implementable policy) to sample from \( p_\theta(a|s) \).
The Variational Inference Approach

Let $q_\eta$ be the distribution of the output $z = G_\eta(\xi)$.

$$\min_\eta \left\{ KL(q_\eta \| p) \right\}$$
The Variational Inference Approach

Let \( q_\eta \) be the distribution of the output \( z = G_\eta(\xi) \).

\[
\min_{\eta} \left\{ KL(q_\eta \| p) \equiv \mathbb{E}_{z \sim q_\eta} [\log q_\eta(z)] - \mathbb{E}_{z \sim q_\eta} [\log p(z)] \right\}.
\]

**Difficulty:**

- We can estimate \( \mathbb{E}_{q_\eta} [\cdot] \) by drawing \( z \sim q_\eta \).
- But cannot calculate \( \log q_\eta(z) \) for given \( z \) except for simple cases:

\[
q_\eta(z) = \int_{\xi} \mathbb{I}[z = G_\eta(\xi)] p_0(\xi) d\xi.
\]
The Variational Inference Approach

Let $q_\eta$ be the distribution of the output $z = G_\eta(\xi)$.

$$\min_\eta \left\{ \text{KL}(q_\eta \| p) \equiv \mathbb{E}_{z \sim q_\eta} [\log q_\eta(z)] - \mathbb{E}_{z \sim q_\eta} [\log p(z)] \right\}.$$ 

entropy, difficult! expectation, easy

Existing approaches:

- Design expressive, yet tractable $q_\eta$: normalizing flow [Rezende, Mohamed 15; Kingma+ 16]; Gaussian process [Tran+ 15], etc.

- Use Entropy or Density ratio estimation: [Mescheder+17; Huszar 17; Shakir+ 17; Train+ 17; Li+ 17 etc].

- Use alternative discrepancy objective functions (Stein discrepancy). [Ranganath+ 16; Liu+ 16].
This talk: Lifting the optimization to Infinite Dimensions

\[
\min_{\eta} \text{KL}(q_{\eta} \parallel p) \quad \iff \quad \min_{q} \text{KL}(q \parallel p), \quad \text{s.t.} \quad q \in \mathcal{G},
\]

where \( \mathcal{G} = \{ q_{\eta} : \forall \eta \} \).

\[ T : \text{a (nonparametric) update that descends KL:} \]
\[ \text{use any MCMC transition} \]
\[ \text{KL}(T q_{\eta} \parallel p) \leq \text{KL}(T q_{\eta} \parallel p) \]

\[ \text{Proj}_{\mathcal{G}} : \text{a projection operator:} \]
\[ \text{use any GAN approach} \]
\[ \text{Proj}_{\mathcal{G}}(q) = \arg \min_{q'} \Delta(q' \parallel q) \]
This talk: Lifting the optimization to Infinite Dimensions

\[
\min_{\eta} \text{KL}(q_\eta \| p)
\]

\[\min_{q} \text{KL}(q \| p), \quad \text{s.t.} \quad q \in \mathcal{G}.
\]

where \( \mathcal{G} = \{q_\eta : \forall \eta\} \).

- Projected Fixed Point:

\[
q_{\eta_{t+1}} = \text{Proj}_G( T q_{\eta_t} ),
\]

- \( T \): a (nonparametric) update that descends KL:

\[
\text{KL}(T q \| p) \leq \text{KL}(T q \| p).
\]

- \( \text{Proj}_G \): a projection operator:

\[
\text{Proj}_G(q) = \arg \min_{q'} \Delta(q' \| q).
\]

\[\text{Amortized MCMC} \quad [\text{Li+17}]\]
This talk: Lifting the optimization to Infinite Dimensions

\[
\min_{\eta} \text{KL}(q_\eta \| p) \quad \iff \quad \min_q \text{KL}(q \| p), \quad \text{s.t.} \quad q \in \mathcal{G}.
\]

where \( \mathcal{G} = \{ q_\eta : \forall \eta \} \).

- **Projected Fixed Point:**
  \[
  q_{\eta_{t+1}} = \text{Proj}_\mathcal{G}(T q_{\eta_t}),
  \]

- **Amortized MCMC [Li+ 17]:**
  - \( T \): a (nonparametric) update that descends \( \text{KL} \): [use any MCMC transition]
    \[
    \text{KL}(Tq \| p) \leq \text{KL}(Tq \| p).
    \]
  - \( \text{Proj}_\mathcal{G} \): a projection operator: [use any GAN approach]
    \[
    \text{Proj}_\mathcal{G}(q) = \arg \min_{q'} \Delta(q' \| q).
    \]
Optimal Transform?

- Given \( q \): tractable to sample from
  - \( p \): intractable to sample from.

- Apply transform \( T(x) \) on \( x \sim q \).

- Find an optimal, yet computationally tractable transform \( T \), such that the distribution \( Tq \) of \( T(x) \) is as close to \( p \) as possible?
Consider deterministic maps $T$ of form

$$T(x) \leftarrow x + \epsilon \phi(x),$$

$\epsilon$: step-size. $\phi$: perturbation direction.
$Tq$: the distribution of $T(x)$ when $x \sim q$.

What is the best $\phi$ to make $Tq$ as close to $p$ as possible?

Idea: maximize the decrease of KL divergence:

$$\phi = \arg \max_{\phi \in \mathcal{F}} \left\{ \text{KL}(q \mid\mid p) - \text{KL}(Tq \mid\mid p) \right\}$$
Consider deterministic maps $T$ of form

$$T(x) \leftarrow x + \epsilon \phi(x),$$

$\epsilon$: step-size. $\phi$: perturbation direction.

$Tq$: the distribution of $T(x)$ when $x \sim q$.

**What is the best $\phi$ to make $Tq$ as close to $p$ as possible?**

Idea: maximize the decrease of KL divergence:

$$\phi = \arg \max_{\phi \in \mathcal{F}} \left\{ \text{KL}(q \mid \mid p) - \text{KL}(Tq \mid \mid p) \right\}$$

$$\approx \arg \max_{\phi \in \mathcal{F}} \left\{ - \frac{\partial}{\partial \epsilon} \text{KL}(Tq \mid \mid p) \bigg|_{\epsilon=0} \right\}, \quad \text{//when step size $\epsilon$ is small}$$
**Stein Variational Gradient** [Liu Wang, 2016]

Key: the objective is a *simple, linear functional* of $\phi$:

$$
- \frac{\partial}{\partial \epsilon} \text{KL}(\mathcal{T}q \mid \mid p)|_{\epsilon=0} = \mathbb{E}_{x \sim q}[\mathcal{T}_p \phi(x)].
$$

where $\mathcal{T}_p$ is a linear operator called **Stein operator** related to $p$:

$$
\mathcal{T}_p \phi(x) \overset{\text{def}}{=} \nabla_x \log p(x)^\top \phi(x) + \nabla_x^\top \phi(x).
$$
Stein Variational Gradient [Liu Wang, 2016]

Key: the objective is a simple, linear functional of $\phi$:

$$\frac{\partial}{\partial \epsilon} \text{KL}(Tq \parallel p)_{\epsilon=0} = \mathbb{E}_{x \sim q}[T_p \phi(x)].$$

where $T_p$ is a linear operator called Stein operator related to $p$:

$$T_p \phi(x) \overset{\text{def}}{=} \nabla_x \log p(x)^\top \phi(x) + \nabla_x^\top \phi(x).$$

Score function $\nabla_x \log p(x) = \frac{\nabla_x p(x)}{p(x)}$, independent of the normalization constant $Z$!
Stein Variational Gradient [Liu Wang, 2016]

Key: the objective is a simple, linear functional of $\phi$:

$$-\frac{\partial}{\partial \epsilon} \text{KL}(Tq || p)|_{\epsilon=0} = \mathbb{E}_{x \sim q}[\mathcal{T}_p \phi(x)].$$

where $\mathcal{T}_p$ is a linear operator called Stein operator related to $p$:

$$\mathcal{T}_p \phi(x) \overset{\text{def}}{=} \nabla_x \log p(x)^\top \phi(x) + \nabla_x^\top \phi(x).$$

- **Stein’s method**: theoretical techniques for proving probabilistic approximation bounds and limit theorems.

- A large body of theoretical work. Known to be “remarkably powerful”.

- Recently extended to practical machine learning [Liu+; Oates+; Mackey+; Chwialkowski+; Ranganath+].
Stein Discrepancy

The optimization is equivalent to

\[
\mathcal{D}(q \parallel p) \overset{\text{def}}{=} \max_{\phi \in \mathcal{F}} \left\{ -\frac{\partial}{\partial \epsilon} \text{KL}(q \parallel p) \bigg|_{\epsilon=0} \right\} \\
\overset{\text{def}}{=} \max_{\phi \in \mathcal{F}} \left\{ \mathbb{E}_q [T_p \phi] \right\}
\]

where \( \mathcal{D}(q \parallel p) \) is called Stein discrepancy: \( \mathcal{D}(q \parallel p) = 0 \) iff \( q = p \) if \( \mathcal{F} \) is “large” enough.
Geometric Interpretation

Stein gradient can be formally viewed as a functional gradient of $\text{KL}(q \| p)$ under a type of “Stein-induced” manifold $\mathcal{M}$ of distributions.

\[
T(x) = x + \epsilon \phi^*(x)
\]

\[
T q = q - \epsilon \nabla_q \text{KL}(q \| p).
\]

\[
\mathbb{D}(q \| p) = \|\nabla_q \text{KL}(q \| p)\|_{\mathcal{M}}.
\]

The minimum cost of transporting the mass of $q$ to $p$.

A new geometry structure on the space of distributions.
Kernel Stein Discrepancy [Liu et al. 16; Chwialkowski et al. 16]

- Take $\mathcal{F}$ to be the unit ball of any reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, with positive kernel $k(x, x')$:

$$\mathbb{D}(q \parallel p) \overset{\text{def}}{=} \max_{\phi \in \mathcal{H}} \left\{ \mathbb{E}_q[\nabla_p \phi] \quad \text{s.t.} \quad \|\phi\|_{\mathcal{H}} \leq 1 \right\}$$

- Closed-form solution:

$$\phi^*(\cdot) \propto \mathbb{E}_{x \sim q}[\nabla_p k(x, \cdot)]$$

$$= \mathbb{E}_{x \sim q}[\nabla_x \log p(x) k(x, \cdot) + \nabla k(x, \cdot)]$$

- Kernel Stein Discrepancy:

$$\mathbb{D}(q, p)^2 = \mathbb{E}_{x, x' \sim q}[\nabla_x^\top \nabla_{x'}^\top k(x, x')]$$

$\nabla^\top_p \nabla^\top_{p'}$: Stein operator w.r.t. variable $x$, $x'$. 

1. $\mathcal{F}$ to be the unit ball of any reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, with positive kernel $k(x, x')$:

$$\mathbb{D}(q \parallel p) \overset{\text{def}}{=} \max_{\phi \in \mathcal{H}} \left\{ \mathbb{E}_q[\nabla_p \phi] \quad \text{s.t.} \quad \|\phi\|_{\mathcal{H}} \leq 1 \right\}$$

- Closed-form solution:

$$\phi^*(\cdot) \propto \mathbb{E}_{x \sim q}[\nabla_p k(x, \cdot)]$$

$$= \mathbb{E}_{x \sim q}[\nabla_x \log p(x) k(x, \cdot) + \nabla k(x, \cdot)]$$

- Kernel Stein Discrepancy:

$$\mathbb{D}(q, p)^2 = \mathbb{E}_{x, x' \sim q}[\nabla_x^\top \nabla_{x'}^\top k(x, x')]$$

$\nabla^\top_p \nabla^\top_{p'}$: Stein operator w.r.t. variable $x$, $x'$. 

1. $\mathcal{F}$ to be the unit ball of any reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, with positive kernel $k(x, x')$:

$$\mathbb{D}(q \parallel p) \overset{\text{def}}{=} \max_{\phi \in \mathcal{H}} \left\{ \mathbb{E}_q[\nabla_p \phi] \quad \text{s.t.} \quad \|\phi\|_{\mathcal{H}} \leq 1 \right\}$$

- Closed-form solution:

$$\phi^*(\cdot) \propto \mathbb{E}_{x \sim q}[\nabla_p k(x, \cdot)]$$

$$= \mathbb{E}_{x \sim q}[\nabla_x \log p(x) k(x, \cdot) + \nabla k(x, \cdot)]$$

- Kernel Stein Discrepancy:

$$\mathbb{D}(q, p)^2 = \mathbb{E}_{x, x' \sim q}[\nabla_x^\top \nabla_{x'}^\top k(x, x')]$$

$\nabla^\top_p \nabla^\top_{p'}$: Stein operator w.r.t. variable $x$, $x'$.
Kernel Stein Discrepancy [Liu et al. 16; Chwialkowski et al. 16]

- Take $\mathcal{F}$ to be the unit ball of any reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, with positive kernel $k(x, x')$:

$$\mathbb{D}(q \parallel p) \overset{\text{def}}{=} \max_{\phi \in \mathcal{H}} \left\{ \mathbb{E}_q[\mathcal{T}_p \phi] \quad \text{s.t.} \quad ||\phi||_{\mathcal{H}} \leq 1 \right\}$$

- Closed-form solution:

$$\phi^*(\cdot) \propto \mathbb{E}_{x \sim q}[\mathcal{T}_p k(x, \cdot)] = \mathbb{E}_{x \sim q}[\nabla_x \log p(x)k(x, \cdot) + \nabla k(x, \cdot)]$$

- Kernel Stein Discrepancy:

$$\mathbb{D}(q, p)^2 = \mathbb{E}_{x, x' \sim q}[\mathcal{T}_p^x \mathcal{T}_p^{x'} k(x, x')]$$

$\mathcal{T}_p^x, \mathcal{T}_p^{x'}$: Stein operator w.r.t. variable $x, x'$. 
Two Basic Tools Derived From Stein

Both $\phi^*$ and $\mathbb{D}(q, p)^2$ can be estimated unbiasedly given $\{x_i\} \sim q$:

- **Stein gradient:** Improve the generator $q$ towards $p$:

  $$\phi^*(\cdot) \approx \frac{1}{n} \sum_{j=1}^{n} \left[ T_p k(x_j, \cdot) \right]$$

  * Applications: Stein variational gradient descent [Liu+ 16, 17].

- **Stein discrepancy:** Evaluate the generator $q$ w.r.t. $p$:

  $$\mathbb{D}(q, p)^2 \approx \frac{1}{n(n-1)} \sum_{i \neq j} T_p^x T_p^{x'} k(x_i, x_j).$$

  * Applications: Goodness of test fit [Liu+ 16, Chwialkowski+ 16].
Stein Variational Gradient Descent

Given sample \( \{ x_i \} \) (drawn from unknown \( q \)), the optimal variable transform:

\[
x_i \leftarrow x_i + \frac{1}{n} \sum_{j=1}^{n} \left[ \nabla_{x_j} \log p(x_j) k(x_j, x_i) + \nabla_{x_j} k(x_j, x_i) \right], \quad \forall i = 1, \ldots, n.
\]

Two terms:

- \( \nabla_{x} \log p(x) \): moves the particles \( \{ x_i \} \) towards high probability regions of \( p(x) \).
- \( \nabla_{x} k(x, x') \): enforces diversity in \( \{ x_i \} \) (otherwise all \( x_i \) collapse to modes of \( p(x) \)).
Stein Variational Gradient Descent

Given sample \( \{x_i\} \) (drawn from unknown \( q \)), the optimal variable transform:

\[
x_i \leftarrow x_i + \epsilon \frac{1}{n} \sum_{j=1}^{n} \left[ \nabla_{x_j} \log p(x_j) k(x_j, x_i) + \nabla_{x_j} k(x_j, x_i) \right], \quad \forall i = 1, \ldots, n.
\]

Two terms:

- \( \nabla_x \log p(x) \): moves the particles \( \{x_i\} \) towards high probability regions of \( p(x) \).
- \( \nabla_x k(x, x') \): enforces diversity in \( \{x_i\} \) (otherwise all \( x_i \) collapse to modes of \( p(x) \)).
Amortized Stein Variational Gradient Descent

Repeat:

- **Simulate** $x_i = G_{\eta_{old}}(\xi_i)$ from the current generator.
- **Improve** $\{x_i\}$ using Stein gradient: $x'_i = x_i + \epsilon \hat{\phi}(x_i)$.
- **Projection**: update $\eta$ to chase $\{x'_i\}$:

  $$
  \eta_{\text{new}} = \arg \min_{\eta} \sum_{i=1}^{n} \left\| x'_i - G_{\eta}(\xi_i) \right\|_2^2
  $$
Similar Ideas Used in Deep Reinforcement Learning

- Amortized SVGD:
  \[
  \eta_{\text{new}} = \arg \min_{\eta} \sum_{i=1}^{n} \left\| \hat{T}(x_i) - G_{\eta}(\xi_i) \right\|_2^2
  \]

- Deep Q-Learning:
  - Bellman operator \( Q^* = TQ^* \).
  \[
  \eta_{t+1} = \arg \min_{\eta} \mathbb{E}(\hat{T}Q_{\eta_t}(s, a) - Q_{\eta}(s, a))^2.
  \]

  - Convergence can not theoretically guaranteed (except linear cases).
  - Empirically works well.
Amortized Stein Variational Gradient Descent

Repeat:

- **Simulate** $x_i = G_{\eta_{\text{old}}} (\xi_i)$ from the current generator.

- **Improve** $\{x_i\}$ using Stein gradient: $x'_i = x_i + \epsilon \hat{\phi}(x_i)$.

- **Projection**: update $\eta$ to chase $\{x'_i\}$:

$$\eta_{\text{new}} = \arg\min_{\eta} \sum_{i=1}^{n} \|x'_i - G_{\eta}(\xi_i)\|_2^2$$

$$\approx \eta_t + \epsilon \sum_i \nabla_{\eta} G_{\eta_{\text{old}}}(\xi_i) \hat{\phi}(x_i)$$  // run a single gradient step

backpropagating Stein gradient to $\eta$
A general back-propagation rule:

\[ \eta_{\text{new}} \approx \eta + \epsilon \sum_i \nabla_\eta G_\eta(\xi_i) \hat{\phi}(x_i). \]

Different methods back-propagate different signals.

- Amortized Stein variational gradient descent:

  \[ \phi(x) = \mathbb{E}_{y \sim q}[\nabla \log p(y)k(y, x) + \nabla_y k(y, x)]. \]

- Typical variational inference with re-parameterization trick:

  \[ \phi(x) = \nabla \log p(x) - \nabla \log q_\eta(x). \]

  Problem: requires to calculate the intractable log density \( \log q_\eta(x) \).

- “Learning to optimize” for making \( x = G_\eta(\xi) \) the maximum of \( \log p \) (max_\eta \mathbb{E}_\xi[\log p(G_\eta(\xi))] )

  \[ \phi(x) = \nabla \log p(x). \]

  Problem: does not take entropy into account.
Amortized SVGD for Variational Auto-encoder

- Typical Gaussian encoder function:
  \[ G_\eta(\xi, x) = \mu(x; \eta) + \sigma(x; \eta)\xi, \quad \xi: \text{standard Gaussian.} \]

- We use a dropout encoder function:
  \[ G_\eta(\xi, x) = NN(x; \xi \odot \eta), \quad \xi: \text{0/1 Bernoulli.} \]

<table>
<thead>
<tr>
<th>Model</th>
<th>NLL/nats</th>
<th>ESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE-f</td>
<td>90.32</td>
<td>84.11</td>
</tr>
<tr>
<td>SteinVAE-f</td>
<td>88.85</td>
<td>83.49</td>
</tr>
<tr>
<td>VAE-CNN</td>
<td>84.68</td>
<td>85.50</td>
</tr>
<tr>
<td>SteinVAE-CNN</td>
<td>84.31</td>
<td>86.57</td>
</tr>
</tbody>
</table>

See also Pu+ 17 Stein Variational Autoencoder.
Hyper-parameter Optimization for MCMC

- Typical MCMC: can be viewed as simulators $G_\eta$:
  - Architecture hand-crafted by researchers, theoretically motivated.
  - (Hyper)-parameters (e.g., step sizes) $\eta$: often set heuristically, but can be adaptive trained by amortized inference.

- Example: Langevin dynamics:
  $$z^{\ell+1} \leftarrow z^\ell + \eta^\ell \odot \nabla_z \log p(z^\ell) + \sqrt{2\eta^\ell} \odot \xi^\ell.$$

- Can be viewed as a “deep resnet”
  - Parameter $\eta$: the step sizes.
  - Random inputs $\xi$: the Gaussian noise + the random initialization.
  - The architecture of $G_\eta$ depends on $p$ through $\nabla_z \log p(z)$. 
Optimizing Step Size for Langevin Dynamics

Goal: Use Langevin dynamics for Bayesian neural network. Optimize the step size using amortized SVGD.

Setting:
- Take 9 similar datasets (a1a to a9a) from UCI repository.
- Train the step size using amortized SVGD using one of the dataset (a9a).
- Test the performance of trained step size on the remaining 8 datasets.
Optimizing Step Size for Langevin Dynamics

- **Goal:** Use Langevin dynamics for Bayesian neural network. Optimize the step size using amortized SVGD.
- **Setting:**
  - Take 9 similar datasets (a1a to a9a) from UCI repository.
  - Train the step size using amortized SVGD using one of the dataset (a9a).
  - Test the performance of trained step size on the remaining 8 datasets.

![Graph showing the comparison of different methods](image)

- Amortized SVGD (10 steps)
- Best Power Decay (10^4 steps)
- Best Power Decay (10^3 steps)
- Best Power Decay (10^2 steps)
- Best Power Decay (10 steps)

Steps of Langevin updates:

(a) Bayesian logistic regression

(b) Bayesian neural networks
Toy Example: Gaussian-Bernoulli RBM

- Train Langevin dynamics to sample randomly generated Gaussian Bernoulli RBM. 100 dimensions, 10 hidden variables.
- Evaluate the MSE of estimating $\mathbb{E}_p[h(x)]$, for different test functions $h$.

![Graphs showing MSE for different sample sizes and functions](image)

- Sample Size ($n$)
  - (a) $\mathbb{E}(x)$
  - (b) $\mathbb{E}(x^2)$
  - (c) $\mathbb{E}(\cos(wx + b))$
Conclusion

- Amortization is a beautiful idea!
- Need efficient methods for amortized inference with implicit models.
- More theories and applications.
Conclusion

- Amortization is a beautiful idea!
- Need efficient methods for amortized inference with implicit models.
- More theories and applications.

Thank You

Powered by SVGD