Probabilistic Inference and Learning with Stein’s Method

Lester Mackey

Microsoft Research New England

September 3, 2020

Collaborators: Jackson Gorham, Andrew Duncan, Sebastian Vollmer, Jonathan Huggins, Wilson Chen, Alessandro Barp, Francois-Xavier Briol, Mark Girolami, Chris Oates, Murat Erdogdu, Ohad Shamir, Marina Riabiz, Jon Cockayne, Pawel Swietach, Steven Niederer, and Anant Raj
Motivation: Large-scale Posterior Inference

Example: Logistic regression

1. Fixed feature vectors: \( v_l \in \mathbb{R}^d \) for each datapoint \( l = 1, \ldots, L \)
2. Binary class labels: \( Y_l \in \{0, 1\} \)
Example: Logistic regression

1. Fixed feature vectors: $v_l \in \mathbb{R}^d$ for each datapoint $l = 1, \ldots, L$
2. Binary class labels: $Y_l \in \{0, 1\}$, $\mathbb{P}(Y_l = 1 \mid v_l, \beta) = \frac{1}{1 + e^{-\langle \beta, v_l \rangle}}$
3. Unknown parameter vector: $\beta \in \mathbb{R}^d$
Example: Bayesian logistic regression

1. Fixed feature vectors: $v_l \in \mathbb{R}^d$ for each datapoint $l = 1, \ldots, L$
2. Binary class labels: $Y_l \in \{0, 1\}$, $P(Y_l = 1 \mid v_l, \beta) = \frac{1}{1 + e^{-\langle \beta, v_l \rangle}}$
3. Unknown parameter vector: $\beta \sim \mathcal{N}(0, I)$

- Generative model simple to express
Example: Bayesian logistic regression

1. Fixed feature vectors: $v_l \in \mathbb{R}^d$ for each datapoint $l = 1, \ldots, L$
2. Binary class labels: $Y_l \in \{0, 1\}$, $P(Y_l = 1 \mid v_l, \beta) = \frac{1}{1+e^{-\langle \beta, v_l \rangle}}$
3. Unknown parameter vector: $\beta \sim \mathcal{N}(0, I)$

- Generative model simple to express
- Posterior distribution over unknown parameters is complex
  - Normalization constant unknown, exact integration intractable
Motivation: Large-scale Posterior Inference

Example: Bayesian logistic regression

1. Fixed feature vectors: $v_l \in \mathbb{R}^d$ for each datapoint $l = 1, \ldots, L$
2. Binary class labels: $Y_l \in \{0, 1\}$, $\mathbb{P}(Y_l = 1 \mid v_l, \beta) = \frac{1}{1 + e^{-\langle \beta, v_l \rangle}}$
3. Unknown parameter vector: $\beta \sim \mathcal{N}(0, I)$

- Generative model simple to express
- Posterior distribution over unknown parameters is complex
  - Normalization constant unknown, exact integration intractable

Standard inferential approach: Use Markov chain Monte Carlo (MCMC) to (eventually) draw samples from the posterior distribution

- Benefit: Approximates intractable posterior expectations
  $\mathbb{E}_P[h(Z)] = \int_X p(x) h(x) dx$ with asymptotically exact sample estimates $\mathbb{E}_Q[h(X)] = \frac{1}{n} \sum_{i=1}^n h(x_i)$
Motivation: Large-scale Posterior Inference

Example: Bayesian logistic regression

1. Fixed feature vectors: $v_l \in \mathbb{R}^d$ for each datapoint $l = 1, \ldots, L$
2. Binary class labels: $Y_l \in \{0, 1\}$, $\mathbb{P}(Y_l = 1 \mid v_l, \beta) = \frac{1}{1 + e^{-\langle \beta, v_l \rangle}}$
3. Unknown parameter vector: $\beta \sim \mathcal{N}(0, I)$

- Generative model simple to express
- Posterior distribution over unknown parameters is complex
  - Normalization constant unknown, exact integration intractable

Standard inferential approach: Use Markov chain Monte Carlo (MCMC) to (eventually) draw samples from the posterior distribution

- **Benefit:** Approximates intractable posterior expectations
  \[ \mathbb{E}_P[h(Z)] = \int_X p(x) h(x) dx \] with asymptotically exact sample estimates
  \[ \mathbb{E}_Q[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i) \]

- **Problem:** Each new MCMC sample point $x_i$ requires iterating over entire observed dataset: prohibitive when dataset is large!
**Motivation: Large-scale Posterior Inference**

**Question:** How do we scale Markov chain Monte Carlo (MCMC) posterior inference to massive datasets?

- **MCMC Benefit:** Approximates intractable posterior expectations $\mathbb{E}_P[h(Z)] = \int_X p(x)h(x)dx$ with asymptotically exact sample estimates $\mathbb{E}_Q[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)$

- **Problem:** Each point $x_i$ requires iterating over entire dataset!

*Template solution:* Approximate MCMC with subset posteriors [Welling and Teh, 2011, Ahn, Korattikara, and Welling, 2012, Korattikara, Chen, and Welling, 2014] Approximate standard MCMC procedure in a manner that makes use of only a small subset of datapoints per sample Reduced computational overhead leads to faster sampling and reduced Monte Carlo variance Introduces asymptotic bias: target distribution is not stationary Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced
Motivation: Large-scale Posterior Inference

Question: How do we scale Markov chain Monte Carlo (MCMC) posterior inference to massive datasets?

- **MCMC Benefit:** Approximates intractable posterior expectations
  \[ \mathbb{E}_P[h(Z)] = \int_X p(x)h(x)dx \] with asymptotically exact sample estimates
  \[ \mathbb{E}_Q[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i) \]

- **Problem:** Each point \( x_i \) requires iterating over entire dataset!

**Template solution:** Approximate MCMC with subset posteriors


- Approximate standard MCMC procedure in a manner that makes use of only a small subset of datapoints per sample
- Reduced computational overhead leads to faster sampling and reduced Monte Carlo variance
- Introduces **asymptotic bias:** target distribution is not stationary
- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced
Motivation: Large-scale Posterior Inference

**Template solution:** Approximate MCMC with subset posteriors


- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced

Introduces new challenges
**Template solution:** Approximate MCMC with subset posteriors


- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced

**Introduces new challenges**

- How do we compare and evaluate samples from approximate MCMC procedures?
Template solution: Approximate MCMC with subset posteriors


- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced

Introduces new challenges

- How do we compare and evaluate samples from approximate MCMC procedures?
- How do we select samplers and their tuning parameters?
Motivation: Large-scale Posterior Inference

**Template solution:** Approximate MCMC with subset posteriors


- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced

**Introduces new challenges**

- How do we compare and evaluate samples from approximate MCMC procedures?
- How do we select samplers and their tuning parameters?
- How do we quantify the bias-variance trade-off explicitly?
**Template solution:** Approximate MCMC with subset posteriors


- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced

**Introduces new challenges**

- How do we compare and evaluate samples from approximate MCMC procedures?
- How do we select samplers and their tuning parameters?
- How do we quantify the bias-variance trade-off explicitly?

**Difficulty:** Standard evaluation criteria like effective sample size, trace plots, and variance diagnostics assume convergence to the target distribution and do not account for asymptotic bias
**Template solution:** Approximate MCMC with subset posteriors

- Hope that for fixed amount of sampling time, variance reduction will outweigh bias introduced

**Introduces new challenges**
- How do we compare and evaluate samples from approximate MCMC procedures?
- How do we select samplers and their tuning parameters?
- How do we quantify the bias-variance trade-off explicitly?

**Difficulty:** Standard evaluation criteria like effective sample size, trace plots, and variance diagnostics assume convergence to the target distribution and do not account for asymptotic bias

**This talk:** Introduce new quality measures suitable for comparing the quality of approximate MCMC samples
**Challenge:** Develop measure suitable for comparing the quality of *any* two samples approximating a common target distribution.

Given a continuous target distribution $P$ with support $X = \mathbb{R}^d$ and density $p$ known up to normalization, integration under $P$ is intractable. Sample points $x_1, \ldots, x_n \in X$ define a discrete distribution $Q_n$ with

$$E_{Q_n}[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)$$

used to approximate $E_P[h(Z)]$.

We make no assumption about the provenance of the $x_i$.

**Goal:**

- Detects when a sample sequence is converging to the target
- Detects when a sample sequence is not converging to the target
- Is computationally feasible
Quality Measures for Samples

**Challenge:** Develop measure suitable for comparing the quality of *any* two samples approximating a common target distribution.

**Given**
- **Continuous target distribution** $P$ with support $\mathcal{X} = \mathbb{R}^d$ and density $p$
  - $p$ known up to normalization, integration under $P$ is intractable
**Challenge:** Develop measure suitable for comparing the quality of any two samples approximating a common target distribution

**Given**

- **Continuous target distribution** $P$ with support $\mathcal{X} = \mathbb{R}^d$ and density $p$
  - $p$ known up to normalization, integration under $P$ is intractable
- **Sample points** $x_1, \ldots, x_n \in \mathcal{X}$
  - Define **discrete distribution** $Q_n$ with, for any function $h$, $\mathbb{E}_{Q_n}[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)$ used to approximate $\mathbb{E}_P[h(Z)]$
  - We make no assumption about the provenance of the $x_i$
Quality Measures for Samples

**Challenge:** Develop measure suitable for comparing the quality of any two samples approximating a common target distribution

**Given**

- **Continuous target distribution** $P$ with support $\mathcal{X} = \mathbb{R}^d$ and density $p$
  - $p$ known up to normalization, integration under $P$ is intractable
- **Sample points** $x_1, \ldots, x_n \in \mathcal{X}$
  - Define discrete distribution $Q_n$ with, for any function $h$,
    $$\mathbb{E}_{Q_n}[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)$$
    used to approximate $\mathbb{E}_P[h(Z)]$
  - We make no assumption about the provenance of the $x_i$

**Goal:** Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$
Quality Measures for Samples

**Challenge:** Develop measure suitable for comparing the quality of *any* two samples approximating a common target distribution

**Given**

- **Continuous target distribution** $P$ with support $\mathcal{X} = \mathbb{R}^d$ and density $p$
  - $p$ known up to normalization, integration under $P$ is intractable
- **Sample points** $x_1, \ldots, x_n \in \mathcal{X}$
  - Define *discrete distribution* $Q_n$ with, for any function $h$,
    \[
    \mathbb{E}_{Q_n}[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)
    \]
    used to approximate $\mathbb{E}_P[h(Z)]$
  - We make no assumption about the provenance of the $x_i$

**Goal:** Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$ in a manner that

1. Detects when a sample sequence is *converging* to the target
**Challenge:** Develop measure suitable for comparing the quality of *any* two samples approximating a common target distribution

**Given**

- **Continuous target distribution** $P$ with support $\mathcal{X} = \mathbb{R}^d$ and density $p$
  - $p$ known up to normalization, integration under $P$ is intractable
- **Sample points** $x_1, \ldots, x_n \in \mathcal{X}$
  - Define **discrete distribution** $Q_n$ with, for any function $h$,
    \[
    \mathbb{E}_{Q_n}[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)
    \]
    used to approximate $\mathbb{E}_P[h(Z)]$
  - We make no assumption about the provenance of the $x_i$

**Goal:** Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$ in a manner that

1. Detects when a sample sequence is converging to the target
2. Detects when a sample sequence is not converging to the target
Quality Measures for Samples

**Challenge:** Develop measure suitable for comparing the quality of any two samples approximating a common target distribution

**Given**
- **Continuous target distribution** $P$ with support $\mathcal{X} = \mathbb{R}^d$ and density $p$
  - $p$ known up to normalization, integration under $P$ is intractable
- **Sample points** $x_1, \ldots, x_n \in \mathcal{X}$
  - Define **discrete distribution** $Q_n$ with, for any function $h$,
    $$\mathbb{E}_{Q_n}[h(X)] = \frac{1}{n} \sum_{i=1}^{n} h(x_i)$$
    used to approximate $\mathbb{E}_P[h(Z)]$
  - We make no assumption about the provenance of the $x_i$

**Goal:** Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$ in a manner that
- I. Detects when a sample sequence is converging to the target
- II. Detects when a sample sequence is not converging to the target
- III. Is computationally feasible
Goal: Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$

Idea: Consider an integral probability metric (IPM) [Müller, 1997]

$$d_{\mathcal{H}}(Q_n, P) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{Q_n}[h(X)] - \mathbb{E}_P[h(Z)]|$$

- Measures maximum discrepancy between sample and target expectations over a class of real-valued test functions $\mathcal{H}$
- When $\mathcal{H}$ sufficiently large, convergence of $d_{\mathcal{H}}(Q_n, P)$ to zero implies $(Q_n)_{n \geq 1}$ converges weakly to $P$ (Requirement II)
Integral Probability Metrics

Goal: Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$

Idea: Consider an integral probability metric (IPM) [Müller, 1997]

$$d_{\mathcal{H}}(Q_n, P) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{Q_n}[h(X)] - \mathbb{E}_P[h(Z)]|$$

- Measures maximum discrepancy between sample and target expectations over a class of real-valued test functions $\mathcal{H}$
- When $\mathcal{H}$ sufficiently large, convergence of $d_{\mathcal{H}}(Q_n, P)$ to zero implies $(Q_n)_{n \geq 1}$ converges weakly to $P$ (Requirement II)

Problem: Integration under $P$ intractable!

$\Rightarrow$ Most IPMs cannot be computed in practice
Integral Probability Metrics

Goal: Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$

Idea: Consider an integral probability metric (IPM) [Müller, 1997]

$$d_{\mathcal{H}}(Q_n, P) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{Q_n}[h(X)] - \mathbb{E}_P[h(Z)]|$$

- Measures maximum discrepancy between sample and target expectations over a class of real-valued test functions $\mathcal{H}$
- When $\mathcal{H}$ sufficiently large, convergence of $d_{\mathcal{H}}(Q_n, P)$ to zero implies $(Q_n)_{n \geq 1}$ converges weakly to $P$ (Requirement II)

Problem: Integration under $P$ intractable!

$\Rightarrow$ Most IPMs cannot be computed in practice

Idea: Only consider functions with $\mathbb{E}_P[h(Z)]$ known a priori to be 0

- Then IPM computation only depends on $Q_n$!
Integral Probability Metrics

Goal: Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$

Idea: Consider an **integral probability metric (IPM)** [Müller, 1997]

$$d_H(Q_n, P) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{Q_n}[h(X)] - \mathbb{E}_P[h(Z)]|$$

- Measures maximum discrepancy between sample and target expectations over a class of real-valued test functions $\mathcal{H}$
- When $\mathcal{H}$ sufficiently large, convergence of $d_H(Q_n, P)$ to zero implies $(Q_n)_{n \geq 1}$ converges weakly to $P$ (**Requirement II**)

Problem: Integration under $P$ intractable!

$\Rightarrow$ Most IPMs cannot be computed in practice

Idea: Only consider functions with $\mathbb{E}_P[h(Z)]$ known *a priori* to be 0

- Then IPM computation only depends on $Q_n$!
- How do we select this class of test functions?
Integral Probability Metrics

**Goal:** Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$

**Idea:** Consider an **integral probability metric (IPM)** [Müller, 1997]

$$d_{\mathcal{H}}(Q_n, P) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{Q_n}[h(X)] - \mathbb{E}_P[h(Z)]|$$

- Measures maximum discrepancy between sample and target expectations over a class of real-valued test functions $\mathcal{H}$
- When $\mathcal{H}$ sufficiently large, convergence of $d_{\mathcal{H}}(Q_n, P)$ to zero implies $(Q_n)_{n \geq 1}$ converges weakly to $P$ (Requirement II)

**Problem:** Integration under $P$ intractable!

$\Rightarrow$ Most IPMs cannot be computed in practice

**Idea:** Only consider functions with $\mathbb{E}_P[h(Z)]$ known *a priori* to be 0

- Then IPM computation only depends on $Q_n$!
- How do we select this class of test functions?
- Will the resulting discrepancy measure track sample sequence convergence (Requirements I and II)?
**Integral Probability Metrics**

**Goal:** Quantify how well $\mathbb{E}_{Q_n}$ approximates $\mathbb{E}_P$.

**Idea:** Consider an **integral probability metric (IPM)** [Müller, 1997]

$$d_{\mathcal{H}}(Q_n, P) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{Q_n}[h(X)] - \mathbb{E}_P[h(Z)]|$$

- Measures maximum discrepancy between sample and target expectations over a class of real-valued test functions $\mathcal{H}$.
- When $\mathcal{H}$ sufficiently large, convergence of $d_{\mathcal{H}}(Q_n, P)$ to zero implies $(Q_n)_{n \geq 1}$ converges weakly to $P$ (**Requirement II**).

**Problem:** Integration under $P$ intractable!

$\implies$ Most IPMs cannot be computed in practice.

**Idea:** Only consider functions with $\mathbb{E}_P[h(Z)]$ known *a priori* to be 0.

- Then IPM computation only depends on $Q_n$!
- How do we select this class of test functions?
- Will the resulting discrepancy measure track sample sequence convergence (**Requirements I and II**)?
- How do we solve the resulting optimization problem in practice?
Stein’s Method

Stein’s method [1972] provides a recipe for controlling convergence:

1. **Identify operator** $\mathcal{T}$ and set $\mathcal{G}$ of functions $g : \mathcal{X} \rightarrow \mathbb{R}^d$ with

$$\mathbb{E}_P[(\mathcal{T}g)(Z)] = 0 \quad \text{for all} \quad g \in \mathcal{G}.$$
Stein’s method [1972] provides a recipe for controlling convergence:

1. **Identify operator** $\mathcal{T}$ **and set** $\mathcal{G}$ **of functions** $g : \mathcal{X} \to \mathbb{R}^d$ **with**

   $$\mathbb{E}_P[(\mathcal{T} g)(Z)] = 0 \quad \text{for all} \quad g \in \mathcal{G}.$$ 

   $\mathcal{T}$ and $\mathcal{G}$ together define the **Stein discrepancy** [Gorham and Mackey, 2015]

   $$\mathcal{S}(Q_n, \mathcal{T}, \mathcal{G}) \overset{\Delta}{=} \sup_{g \in \mathcal{G}} |\mathbb{E}_{Q_n}[(\mathcal{T} g)(X)]| = d_{\mathcal{T} \mathcal{G}}(Q_n, P),$$

   an IPM-type measure with no explicit integration under $P$. 

---

Inference and Learning with Stein’s Method  
September 3, 2020 7 / 33
Stein’s Method

Stein’s method [1972] provides a recipe for controlling convergence:

1. **Identify operator** $\mathcal{T}$ and set $\mathcal{G}$ of functions $g : \mathcal{X} \to \mathbb{R}^d$ with

   $$\mathbb{E}_P[(\mathcal{T}g)(Z)] = 0 \quad \text{for all} \quad g \in \mathcal{G}.$$ 

   $\mathcal{T}$ and $\mathcal{G}$ together define the **Stein discrepancy** [Gorham and Mackey, 2015]

   $$S(Q_n, \mathcal{T}, \mathcal{G}) \triangleq \sup_{g \in \mathcal{G}} |\mathbb{E}_{Q_n}[(\mathcal{T}g)(X)]| = d_{\mathcal{T} \mathcal{G}}(Q_n, P),$$

   an IPM-type measure with no explicit integration under $P$

2. **Lower bound** $S(Q_n, \mathcal{T}, \mathcal{G})$ by reference IPM $d_H(Q_n, P)$

   $\Rightarrow (Q_n)_{n \geq 1}$ converges to $P$ whenever $S(Q_n, \mathcal{T}, \mathcal{G}) \to 0$ (**Req. II**)

   - Performed once, in advance, for large classes of distributions
Stein’s method [1972] provides a recipe for controlling convergence:

1. **Identify operator** $\mathcal{T}$ and set $\mathcal{G}$ of functions $g : \mathcal{X} \to \mathbb{R}^d$ with 
   $$
   \mathbb{E}_P[(\mathcal{T}g)(Z)] = 0 \quad \text{for all} \quad g \in \mathcal{G}.
   $$
   $\mathcal{T}$ and $\mathcal{G}$ together define the **Stein discrepancy** [Gorham and Mackey, 2015] 
   $$
   S(Q_n, \mathcal{T}, \mathcal{G}) \triangleq \sup_{g \in \mathcal{G}} |\mathbb{E}_{Q_n}[(\mathcal{T}g)(X)]| = d_{\mathcal{T}\mathcal{G}}(Q_n, P),
   $$
   an IPM-type measure with no explicit integration under $P$.

2. **Lower bound** $S(Q_n, \mathcal{T}, \mathcal{G})$ by reference IPM $d_{\mathcal{H}}(Q_n, P)$
   $$
   \Rightarrow (Q_n)_{n \geq 1} \text{ converges to } P \text{ whenever } S(Q_n, \mathcal{T}, \mathcal{G}) \to 0 \ (\text{Req. II})
   $$
   - Performed once, in advance, for large classes of distributions.

3. **Upper bound** $S(Q_n, \mathcal{T}, \mathcal{G})$ by any means necessary to demonstrate convergence to 0 (Requirement I)

---

The inference and learning with Stein's Method presented here are based on the research conducted by Mackey (MSR) as of September 3, 2020.
Stein’s Method

Stein’s method [1972] provides a recipe for controlling convergence:

1. **Identify operator** $\mathcal{T}$ and set $\mathcal{G}$ of functions $g : \mathcal{X} \to \mathbb{R}^d$ with
   \[
   \mathbb{E}_P[(\mathcal{T}g)(Z)] = 0 \quad \text{for all} \quad g \in \mathcal{G}.
   \]
   $\mathcal{T}$ and $\mathcal{G}$ together define the **Stein discrepancy** [Gorham and Mackey, 2015]
   \[
   S(Q_n, \mathcal{T}, \mathcal{G}) \triangleq \sup_{g \in \mathcal{G}} |\mathbb{E}_{Q_n}[(\mathcal{T}g)(X)]| = d_{\mathcal{T}g}(Q_n, P),
   \]
   an IPM-type measure with no explicit integration under $P$.

2. **Lower bound** $S(Q_n, \mathcal{T}, \mathcal{G})$ by reference IPM $d_H(Q_n, P)$
   \[\Rightarrow (Q_n)_{n \geq 1} \text{ converges to } P \text{ whenever } S(Q_n, \mathcal{T}, \mathcal{G}) \to 0 \text{ (Req. II)}\]
   - Performed once, in advance, for large classes of distributions

3. **Upper bound** $S(Q_n, \mathcal{T}, \mathcal{G})$ by any means necessary to demonstrate convergence to 0 (Requirement I)

**Standard use:** As analytical tool to prove convergence

**Our goal:** Develop Stein discrepancy into practical quality measure
Goal: Identify operator $\mathcal{T}$ for which $\mathbb{E}_P[(\mathcal{T}g)(Z)] = 0$ for all $g \in \mathcal{G}$
Goal: Identify operator $\mathcal{T}$ for which $\mathbb{E}_P[(\mathcal{T}g)(Z)] = 0$ for all $g \in \mathcal{G}$


- Identify a Markov process $(Z_t)_{t \geq 0}$ with stationary distribution $P$
- Under mild conditions, its infinitesimal generator
  
  $$(\mathcal{A}u)(x) = \lim_{t \to 0} \frac{\mathbb{E}[u(Z_t) | Z_0 = x] - u(x)}{t}$$

  satisfies $\mathbb{E}_P[(\mathcal{A}u)(Z)] = 0$

Overdamped Langevin diffusion: $dZ_t = \frac{1}{2} \nabla \log p(Z_t) dt + dW_t$

- Generator: $(\mathcal{A}_P u)(x) = \frac{1}{2} \langle \nabla u(x), \nabla \log p(x) \rangle + \frac{1}{2} \langle \nabla, \nabla u(x) \rangle$
Goal: Identify operator $T$ for which $\mathbb{E}_P[(Tg)(Z)] = 0$ for all $g \in \mathcal{G}$


- Identify a Markov process $(Z_t)_{t \geq 0}$ with stationary distribution $P$
- Under mild conditions, its infinitesimal generator
  $$(\mathcal{A}u)(x) = \lim_{t \to 0} \left( \mathbb{E}[u(Z_t) \mid Z_0 = x] - u(x) \right)/t$$
  satisfies $\mathbb{E}_P[(\mathcal{A}u)(Z)] = 0$

Overdamped Langevin diffusion: $dZ_t = \frac{1}{2} \nabla \log p(Z_t) dt + dW_t$

- Generator: $(\mathcal{A}P u)(x) = \frac{1}{2} \langle \nabla u(x), \nabla \log p(x) \rangle + \frac{1}{2} \langle \nabla, \nabla u(x) \rangle$
- Stein operator: $(\mathcal{T}_P g)(x) \triangleq \langle g(x), \nabla \log p(x) \rangle + \langle \nabla, g(x) \rangle$
  [Gorham and Mackey, 2015, Oates, Girolami, and Chopin, 2016]
  - Depends on $P$ only through $\nabla \log p$; computable even if $p$
    cannot be normalized!
  - Multivariate generalization of density method operator
    $$(\mathcal{T} g)(x) = g(x) \frac{d}{dx} \log p(x) + g'(x)$$ [Stein, Diaconis, Holmes, and Reinert, 2004]
Identifying a Stein Operator $\mathcal{T}$

**Goal:** Identify operator $\mathcal{T}$ for which $\mathbb{E}_P[(\mathcal{T}g)(Z)] = 0$ for all $g \in \mathcal{G}$

**Approach:** Generator method of Barbour [1988, 1990], Götze [1991]

- Identify a Markov process $(Z_t)_{t \geq 0}$ with stationary distribution $P$
- Under mild conditions, its **infinitesimal generator**
  \[
  (\mathcal{A}u)(x) = \lim_{t \to 0} \frac{(\mathbb{E}[u(Z_t) | Z_0 = x] - u(x))}{t}
  \]
  satisfies $\mathbb{E}_P[(\mathcal{A}u)(Z)] = 0$

**Overdamped Langevin diffusion:**
\[
dZ_t = \frac{1}{2} \nabla \log p(Z_t) dt + dW_t
\]

- **Generator:** $(\mathcal{A}_P u)(x) = \frac{1}{2} \langle \nabla u(x), \nabla \log p(x) \rangle + \frac{1}{2} \langle \nabla, \nabla u(x) \rangle$

- **Stein operator:** $(\mathcal{T}_P g)(x) \triangleq \langle g(x), \nabla \log p(x) \rangle + \langle \nabla, g(x) \rangle$
  [Gorham and Mackey, 2015, Oates, Girolami, and Chopin, 2016]

  - Depends on $P$ only through $\nabla \log p$; computable even if $p$ cannot be normalized!
  - $\mathbb{E}_P[(\mathcal{T}_P g)(Z)] = 0$ for all $g : \mathcal{X} \to \mathbb{R}^d$ in classical Stein set

\[
\mathcal{G}_{\| \cdot \|} = \{ g : \sup_{x \neq y} \max \left( \| g(x) \|^{*}, \| \nabla g(x) \|^{*}, \frac{\| \nabla g(x) - \nabla g(y) \|^{*}}{\| x - y \|^{*}} \right) \leq 1 \}
\]
Goal: Show classical Stein discrepancy $S(Q_n, T_P, G_{\|\cdot\|}) \to 0$ if and only if $(Q_n)_{n\geq 1}$ converges to $P$

- In the univariate case ($d = 1$), known that for many targets $P$, $S(Q_n, T_P, G_{\|\cdot\|}) \to 0$ only if Wasserstein $d_{\mathcal{W}_{\|\cdot\|}}(Q_n, P) \to 0$


- Few multivariate targets have been analyzed (see [Reinert and Röllin, 2009, Chatterjee and Meckes, 2008, Meckes, 2009] for multivariate Gaussian)
Detecting Convergence and Non-convergence

Goal: Show classical Stein discrepancy $S(Q_n, T_P, G_{\|\cdot\|}) \to 0$ if and only if $(Q_n)_{n \geq 1}$ converges to $P$

- In the univariate case ($d = 1$), known that for many targets $P$, $S(Q_n, T_P, G_{\|\cdot\|}) \to 0$ only if Wasserstein $d_{W_{\|\cdot\|}}(Q_n, P) \to 0$


- Few multivariate targets have been analyzed (see [Reinert and Röllin, 2009, Chatterjee and Meckes, 2008, Meckes, 2009] for multivariate Gaussian)

New contribution [Gorham, Duncan, Vollmer, and Mackey, 2019]

**Theorem (Stein Discrepancy-Wasserstein Equivalence)**

If the Langevin diffusion couples at an integrable rate and $\nabla \log p$ is Lipschitz, then $S(Q_n, T_P, G_{\|\cdot\|}) \to 0 \iff d_{W_{\|\cdot\|}}(Q_n, P) \to 0$.

- Examples: strongly log concave $P$, Bayesian logistic regression or robust $t$ regression with Gaussian priors, Gaussian mixtures

- Conditions not necessary: template for bounding $S(Q_n, T_P, G_{\|\cdot\|})$
- For target $P = \mathcal{N}(0, 1)$, compare i.i.d. $\mathcal{N}(0, 1)$ sample sequence $Q_{1:n}$ to scaled Student’s t sequence $Q'_{1:n}$ with matching variance.
- Expect $S(Q_{1:n}, \mathcal{T}_P, \mathcal{G}_{\|\cdot\|}, Q, G_1) \rightarrow 0$ & $S(Q'_{1:n}, \mathcal{T}_P, \mathcal{G}_{\|\cdot\|}, Q, G_1) \not\rightarrow 0$
Middle: Recovered optimal functions $g$
Right: Associated test functions $h(x) \triangleq (\mathcal{T}_P g)(x)$ which best discriminate sample $Q$ from target $P$
Selecting Sampler Hyperparameters

**Target posterior density:** \( p(x) \propto \pi(x) \prod_{l=1}^{L} \pi(y_l \mid x) \)

**Stochastic Gradient Langevin Dynamics** [Welling and Teh, 2011]

\[ x_{k+1} \sim \mathcal{N}(x_k + \frac{\epsilon}{2} \nabla \log \pi(x_k) + \frac{L}{|B_k|} \sum_{l \in B_k} \nabla \log \pi(y_l \mid x_k)), \epsilon I) \]

- Random batch \( B_k \) of datapoints used to draw each sample point
  - Step size \( \epsilon \) too small \( \Rightarrow \) slow mixing
  - Step size \( \epsilon \) too large \( \Rightarrow \) sampling from very different distribution
  - Standard MCMC selection criteria like effective sample size (ESS) and asymptotic variance do not account for this bias

ESS maximized at \( \epsilon = 5 \times 10^{-2} \), Stein minimized at \( \epsilon = 5 \times 10^{-3} \)
Goal: Identify a more “user-friendly” Stein set $\mathcal{G}$ than the classical approach: Reproducing kernels $k: X \times X \to \mathbb{R}$.

A reproducing kernel $k$ is symmetric ($k(x, y) = k(y, x)$) and positive semidefinite ($\sum_{i, l} c_i c_l k(z_i, z_l) \geq 0$, $\forall z_i \in X, c_i \in \mathbb{R}$).

Gaussian: $k(x, y) = e^{-\frac{1}{2} \|x - y\|^2}$.

IMQ: $k(x, y) = \left(1 + \|x - y\|^2\right)^{-\frac{1}{2}}$.

Generates a reproducing kernel Hilbert space (RKHS) $K_k$.

Define the kernel Stein set $\mathcal{G}_k \equiv \{g = (g_1, \ldots, g_d) | \|v\|_* \leq 1 \text{ for } v_j \equiv \|g_j\|_{K_k}\}$.

Yields closed-form kernel Stein discrepancy (KSD) $S(Q_n, T_P, \mathcal{G}_k) = \|w\|$ for $w_j \equiv \sqrt{\sum_{i, i'} k_{0j}(x_i, x_{i'})}$.

Reduces to parallelizable pairwise evaluations of Stein kernels $k_{0j}(x, y) \equiv \frac{1}{p(x)p(y)} \nabla_x g_j \nabla_y g_j(p(x)k(x, y)p(y))$.

Mackey (MSR) Inference and Learning with Stein’s Method September 3, 2020 13 / 33
**Goal:** Identify a more “user-friendly” Stein set $\mathcal{G}$ than the classical

**Approach:** Reproducing kernels $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ [Oates, Girolami, and Chopin, 2016, Chwialkowski, Strathmann, and Gretton, 2016, Liu, Lee, and Jordan, 2016]

- A reproducing kernel $k$ is symmetric ($k(x, y) = k(y, x)$) and positive semidefinite ($\sum_{i,l} c_i c_l k(z_i, z_l) \geq 0, \forall z_i \in \mathcal{X}, c_i \in \mathbb{R}$)
  - Gaussian: $k(x, y) = e^{-\frac{1}{2} \|x-y\|^2}$, IMQ: $k(x, y) = \frac{1}{(1+\|x-y\|^2)^{1/2}}$
  - Generates a reproducing kernel Hilbert space (RKHS) $\mathcal{K}_k$
**Goal:** Identify a more “user-friendly” Stein set $\mathcal{G}$ than the classical

**Approach:** Reproducing kernels $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ [Oates, Girolami, and Chopin, 2016, Chwialkowski, Strathmann, and Gretton, 2016, Liu, Lee, and Jordan, 2016]

- A reproducing kernel $k$ is symmetric ($k(x, y) = k(y, x)$) and positive semidefinite ($\sum_{i,l} c_i c_l k(z_i, z_l) \geq 0, \forall z_i \in \mathcal{X}, c_i \in \mathbb{R}$)
- Gaussian: $k(x, y) = e^{-\frac{1}{2}\|x-y\|^2}$, IMQ: $k(x, y) = \frac{1}{(1+\|x-y\|^2)^{1/2}}$
- Generates a reproducing kernel Hilbert space (RKHS) $\mathcal{K}_k$
- Define the **kernel Stein set** [Gorham and Mackey, 2017]
  
  $$\mathcal{G}_k \triangleq \{ g = (g_1, \ldots, g_d) \mid \|v\|^* \leq 1 \text{ for } v_j \triangleq \|g_j\|_{\mathcal{K}_k} \}$$
**Goal:** Identify a more “user-friendly” Stein set $\mathcal{G}$ than the classical

**Approach:** Reproducing kernels $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ \cite{Oates2016, Chwialkowski2016, Liu2016}

- A reproducing kernel $k$ is **symmetric** ($k(x, y) = k(y, x)$) and **positive semidefinite** ($\sum_{i, l} c_i c_l k(z_i, z_l) \geq 0, \forall z_i \in \mathcal{X}, c_i \in \mathbb{R}$).

  - Gaussian: $k(x, y) = e^{-\frac{1}{2}\|x-y\|^2}$, **IMQ**: $k(x, y) = \frac{1}{(1+\|x-y\|^2)^{1/2}}$

Generates a reproducing kernel Hilbert space (RKHS) $\mathcal{K}_k$

- Define the **kernel Stein set** \cite{Gorham2017}

  \[ \mathcal{G}_k \triangleq \{ g = (g_1, \ldots, g_d) \mid \|v\|^* \leq 1 \text{ for } v_j \triangleq \|g_j\|_{\mathcal{K}_k} \} \]

Yields **closed-form kernel Stein discrepancy (KSD)**

\[ S(Q_n, T_P, \mathcal{G}_k) = \|w\| \text{ for } w_j \triangleq \sqrt{\sum_{i, i'=1}^n k_0^j(x_i, x_{i'})}. \]

- Reduces to **parallelizable** pairwise evaluations of **Stein kernels**

  \[ k_0^j(x, y) \triangleq \frac{1}{p(x)p(y)} \nabla x_j \nabla y_j (p(x)k(x, y)p(y)) \]
Goal: Show \((Q_n)_{n \geq 1}\) converges to \(P\) whenever \(S(Q_n, T_P, G_k) \to 0\)
Detecting Non-convergence

**Goal:** Show \((Q_n)_{n \geq 1}\) converges to \(P\) whenever \(S(Q_n, T_P, G_k) \to 0\)

**Theorem (Univariate KSD detects non-convergence [Gorham and Mackey, 2017])**

Suppose \(P \in \mathcal{P}\) and \(k(x, y) = \Phi(x - y)\) for \(\Phi \in C^2\) with a non-vanishing generalized Fourier transform. If \(d = 1\), then \((Q_n)_{n \geq 1}\) converges weakly to \(P\) whenever \(S(Q_n, T_P, G_k) \to 0\).

- \(\mathcal{P}\) is the set of targets \(P\) with Lipschitz \(\nabla \log p\) and distant strong log concavity \(\left(\frac{\langle \nabla \log(p(x)/p(y)), y-x \rangle}{\|x-y\|_2^2}\right) \geq k\) for \(\|x - y\|_2 \geq r\)
  - Includes Bayesian logistic and Student’s t regression with Gaussian priors, Gaussian mixtures with common covariance, ...

- Justifies use of KSD with popular Gaussian, Matérn, or inverse multiquadric kernels \(k\) **in the univariate case**
Detecting Non-convergence

**Goal:** Show \((Q_n)_{n \geq 1}\) converges to \(P\) whenever \(S(Q_n, T_P, G_k) \rightarrow 0\)

- In higher dimensions, KSDs based on common kernels fail to detect non-convergence, even for Gaussian targets \(P\)

\[k(x, y) = e^{-\frac{1}{2}x^2} \] and Matérn kernels fail for \(d \geq 3\)

\[k(x, y) = (1 + \|x - y\|^2)^{\beta} \] with \(\beta < -1\) fail for \(d > \frac{1}{2}(1 + \beta)\)

The violating sample sequences \((Q_n)_{n \geq 1}\) are simple to construct.

Problem: Kernels with light tails ignore excess mass in the tails.
Detecting Non-convergence

**Goal:** Show \((Q_n)_{n \geq 1}\) converges to \(P\) whenever \(S(Q_n, T_P, G_k) \to 0\)

- In higher dimensions, KSDs based on common kernels fail to detect non-convergence, even for Gaussian targets \(P\)

---

**Theorem (KSD fails with light kernel tails [Gorham and Mackey, 2017])**

Suppose \(d \geq 3\), \(P = N(0, I_d)\), and \(\alpha \triangleq (\frac{1}{2} - \frac{1}{d})^{-1}\). If \(k(x, y)\) and its derivatives decay at a \(o(\|x - y\|_2^{-\alpha})\) rate as \(\|x - y\|_2 \to \infty\), then \(S(Q_n, T_P, G_k) \to 0\) for some \((Q_n)_{n \geq 1}\) not converging to \(P\).

- Gaussian \((k(x, y) = e^{-\frac{1}{2}\|x-y\|_2^2})\) and Matérn kernels fail for \(d \geq 3\)
- Inverse multiquadric kernels \((k(x, y) = (1 + \|x - y\|_2^2)^{\beta})\) with \(\beta < -1\) fail for \(d > \frac{2\beta}{1+\beta}\)
- The violating sample sequences \((Q_n)_{n \geq 1}\) are simple to construct

**Problem:** Kernels with light tails ignore excess mass in the tails
Goal: Show \((Q_n)_{n\geq 1}\) converges to \(P\) whenever \(S(Q_n, T_P, G_k) \to 0\)

- Consider the inverse multiquadric (IMQ) kernel 
  \[ k(x, y) = (c^2 + \|x - y\|_2^2)^{\beta} \text{ for some } \beta < 0, c \in \mathbb{R}. \]

- IMQ KSD fails to detect non-convergence when \(\beta < -1\)
Detecting Non-convergence

Goal: Show \( (Q_n)_{n \geq 1} \) converges to \( P \) whenever \( S(Q_n, \mathcal{T}_P, G_k) \to 0 \)

- Consider the inverse multiquadric (IMQ) kernel
  \[
  k(x, y) = \left( c^2 + \|x - y\|_2^2 \right)^\beta
  \]
  for some \( \beta < 0, c \in \mathbb{R} \).
- IMQ KSD fails to detect non-convergence when \( \beta < -1 \)
- However, IMQ KSD detects non-convergence when \( \beta \in (-1, 0) \)

Theorem (IMQ KSD detects non-convergence [Gorham and Mackey, 2017])

Suppose \( P \in \mathcal{P} \) and \( k(x, y) = \left( c^2 + \|x - y\|_2^2 \right)^\beta \) for \( \beta \in (-1, 0) \). If \( S(Q_n, \mathcal{T}_P, G_k) \to 0 \), then \( (Q_n)_{n \geq 1} \) converges weakly to \( P \).
Goal: Show $S(Q_n, T_P, G_k) \rightarrow 0$ whenever $(Q_n)_{n \geq 1}$ converges to $P$. 
Detecting Convergence

**Goal:** Show $S(Q_n, T_P, G_k) \to 0$ whenever $(Q_n)_{n \geq 1}$ converges to $P$

**Proposition (KSD detects convergence [Gorham and Mackey, 2017])**

If $k \in C^{(2,2)}_b$ and $\nabla \log p$ Lipschitz and square integrable under $P$, then $S(Q_n, T_P, G_k) \to 0$ whenever the Wasserstein distance $d_{\mathcal{W}_\|\cdot\|_2}(Q_n, P) \to 0$.

- Covers Gaussian, Matérn, IMQ, and other common bounded kernels $k$
Stochastic Gradient Fisher Scoring (SGFS)
[Ahn, Korattikara, and Welling, 2012]

- Approximate MCMC procedure designed for scalability
  - Approximates Metropolis-adjusted Langevin algorithm but does not use Metropolis-Hastings correction
  - Target $P$ is not stationary distribution

- **Goal:** Choose between two variants
  - SGFS-f inverts a $d \times d$ matrix for each new sample point
  - SGFS-d inverts a diagonal matrix to reduce sampling time

- **MNIST handwritten digits** [Ahn, Korattikara, and Welling, 2012]
  - 10000 images, 51 features, binary label indicating whether image of a 7 or a 9

- Bayesian logistic regression posterior $P$
Selecting Samplers

**Left:** IMQ KSD quality comparison for SGFS Bayesian logistic regression (no surrogate ground truth used)

**Right:** SGFS sample points \((n = 5 \times 10^4)\) with bivariate marginal means and 95% confidence ellipses (blue) that align best and worst with surrogate ground truth sample (red)

Both suggest small speed-up of SGFS-d (0.0017s per sample vs. 0.0019s for SGFS-f) outweighed by loss in inferential accuracy
Goodness-of-fit testing

- Chwialkowski, Strathmann, and Gretton [2016] used the KSD $S(Q_n, T_P, G_k)$ to test whether a sample was drawn from a target distribution $P$ (see also Liu, Lee, and Jordan [2016])
- Test with default Gaussian kernel $k$ experienced considerable loss of power as the dimension $d$ increased

<table>
<thead>
<tr>
<th></th>
<th>$d=2$</th>
<th>$d=5$</th>
<th>$d=10$</th>
<th>$d=15$</th>
<th>$d=20$</th>
<th>$d=25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B&amp;H</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.91</td>
<td>0.57</td>
<td>0.26</td>
</tr>
<tr>
<td>Gaussian</td>
<td>1.0</td>
<td>1.0</td>
<td>0.88</td>
<td>0.29</td>
<td>0.12</td>
<td>0.02</td>
</tr>
<tr>
<td>IMQ</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Beyond Sample Quality Comparison

**Goodness-of-fit testing**

- Chwialkowski, Strathmann, and Gretton [2016] used the KSD $S(Q_n, T_P, G_k)$ to test whether a sample was drawn from a target distribution $P$ (see also Liu, Lee, and Jordan [2016])
- Test with default Gaussian kernel $k$ experienced considerable loss of power as the dimension $d$ increased
- We recreate their experiment with IMQ kernel ($\beta = -\frac{1}{2}, c = 1$)
  - For $n = 500$, generate sample $(x_i)_{i=1}^n$ with $x_i = z_i + u_i e_1$
    - $z_i \sim \mathcal{N}(0, I_d)$ and $u_i \sim \text{Unif}[0, 1]$. Target $P = \mathcal{N}(0, I_d)$.
  - Compare with standard normality test of Baringhaus and Henze [1988]

<table>
<thead>
<tr>
<th></th>
<th>d=2</th>
<th>d=5</th>
<th>d=10</th>
<th>d=15</th>
<th>d=20</th>
<th>d=25</th>
</tr>
</thead>
<tbody>
<tr>
<td>B&amp;H</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.91</td>
<td>0.57</td>
<td>0.26</td>
</tr>
<tr>
<td>Gaussian</td>
<td>1.0</td>
<td>1.0</td>
<td>0.88</td>
<td>0.29</td>
<td>0.12</td>
<td>0.02</td>
</tr>
<tr>
<td>IMQ</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Table:** Mean power of multivariate normality tests across 400 simulations
Improving sample quality

Given sample points \((x_i)_{i=1}^n\), can minimize KSD \(S(\tilde{Q}_n, T_P, G_k)\) over all weighted samples \(\tilde{Q}_n = \sum_{i=1}^n q_n(x_i) \delta_{x_i}\) for \(q_n\) a probability mass function.

Liu and Lee [2016] do this with Gaussian kernel \(k(x, y) = e^{-\frac{1}{h}||x-y||_2^2}\)
- Bandwidth \(h\) set to median of the squared Euclidean distance between pairs of sample points.

We recreate their experiment with the IMQ kernel \(k(x, y) = (1 + \frac{1}{h}||x - y||_2^2)^{-1/2}\)
Improving Sample Quality

- MSE averaged over 500 simulations ($\pm 2$ standard errors)
- Target $P = \mathcal{N}(0, I_d)$
- Starting sample $Q_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ for $x_i \overset{iid}{\sim} P$, $n = 100$. 

**Graph:**
- Y-axis: Average MSE, $\| E_P Z - E_{\tilde{Q}_n} X \|^2 / d$
- X-axis: Dimension, $d$
- Legend:
  - Red line: Initial $Q_n$
  - Green line: Gaussian KSD
  - Blue line: IMQ KSD

Sample
Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016]

- Uses KSD to repeatedly update locations of $n$ sample points:
  \[
  x_i \leftarrow x_i + \frac{\epsilon}{n} \sum_{l=1}^{n} \left( k(x_l, x_i) \nabla \log p(x_l) + \nabla_{x_l} k(x_l, x_i) \right)
  \]
- Approximates gradient step in KL divergence (but convergence is unclear)
- Simple to implement (but each update costs $n^2$ time)
Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016]
- Uses KSD to repeatedly update locations of \( n \) sample points:
  \[
  x_i \leftarrow x_i + \frac{\epsilon}{n} \sum_{l=1}^{n} (k(x_l, x_i) \nabla \log p(x_l) + \nabla x_l k(x_l, x_i))
  \]
- Approximates gradient step in KL divergence (but convergence is unclear)
- Simple to implement (but each update costs \( n^2 \) time)

Stein Points [Chen, Mackey, Gorham, Briol, and Oates, 2018]
- Greedily minimizes KSD by constructing \( Q_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i} \) with
  \[
  x_n \in \arg\min_x S\left( \frac{n-1}{n} Q_{n-1} + \frac{1}{n} \delta_x, \mathcal{T}_P, \mathcal{G}_k \right)
  \]
  \[
  = \arg\min_x \sum_{j=1}^{d} \frac{k_j(x,x)}{2} + \sum_{i=1}^{n-1} k_0^j(x_i, x)
  \]
- Can generate sample sequence from scratch or, under budget constraints, optimize existing locations by coordinate descent
- Sends KSD to zero at \( O\left(\sqrt{\log(n)/n}\right) \) rate
Generating High-quality Samples

Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016]
- Uses KSD to repeatedly update locations of $n$ sample points:
  $$x_i \leftarrow x_i + \frac{\epsilon}{n} \sum_{l=1}^{n} (k(x_l, x_i) \nabla \log p(x_l) + \nabla_{x_l} k(x_l, x_i))$$
- Approximates gradient step in KL divergence (but convergence is unclear)
- Simple to implement (but each update costs $n^2$ time)

Stein Points [Chen, Mackey, Gorham, Briol, and Oates, 2018]
- Greedily minimizes KSD by constructing $Q_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ with
  $$x_n \in \arg\min_x S\left(\frac{n-1}{n} Q_{n-1} + \frac{1}{n} \delta_x, \mathcal{T}_P, \mathcal{G}_k\right)$$
  $$= \arg\min_x \sum_{j=1}^{d} \frac{k^j_0(x,x)}{2} + \sum_{i=1}^{n-1} k^j_0(x_i, x)$$
- Can generate sample sequence from scratch or, under budget constraints, optimize existing locations by coordinate descent
- Sends KSD to zero at $O(\sqrt{\log(n)/n})$ rate

Stein Point MCMC [Chen, Barp, Briol, Gorham, Girolami, Mackey, and Oates, 2019]
- Suffices to optimize over iterates of a Markov chain
Stein Point MCMC [Chen, Barp, Briol, Gorham, Girolami, Mackey, and Oates, 2019]

- Greedily minimizes KSD by constructing $Q_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ with

$$x_n \in \arg\min_x S\left(\frac{n-1}{n}Q_{n-1} + \frac{1}{n}\delta_x, T_P, G_k\right)$$

$$= \arg\min_x \sum_{j=1}^{d} \frac{k^j_0(x,x)}{2} + \sum_{i=1}^{n-1} k^j_0(x_i, x)$$

- Suffices to optimize over iterates of a Markov chain
**Goodwin oscillator**: kinetic model of oscillatory enzymatic control

- Measure mRNA and protein product $(y_i)_{i=1}^{40}$ at times $(t_i)_{i=1}^{40}$
  
  \[ y_i = g(u(t_i)) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2 I) \]

- \[ \dot{u}(t) = f_\theta(t, u), \quad u(0) = u_0 \in \mathbb{R}^8 \]

- $P$ is posterior of $\log(\theta) \in \mathbb{R}^{10}$ given $y, t$ and $\log \Gamma(2, 1)$ priors

- Evaluating likelihood requires solving the ODE numerically
**IGARCH model** of financial time series with time-varying volatility

- Daily percentage returns $y = (y_t)_{t=1}^{2000}$ from S&P 500 modeled as
  
  \[ y_t = \sigma_t \epsilon_t, \quad \epsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1) \]

  \[ \sigma_t^2 = \theta_1 + \theta_2 y_{t-1}^2 + (1 - \theta_2)\sigma_{t-1}^2 \]

- $P$ is posterior of $\theta_1 > 0$, $\theta_2 \in (0, 1)$ given $y$ and uniform priors
Future Directions

Many opportunities for future development

1. Improving scalability while maintaining convergence control
   - Subsampling of likelihood terms in $\nabla \log p$
   - **Stochastic Stein discrepancies** [Gorham, Raj, and Mackey, 2020]: control convergence with probability 1
Future Directions

Many opportunities for future development

1. Improving scalability while maintaining convergence control
   - Subsampling of likelihood terms in $\nabla \log p$
     - **Stochastic Stein discrepancies** [Gorham, Raj, and Mackey, 2020]: control convergence with probability 1
   - Inexpensive approximations of kernel matrix
     - **Finite set Stein discrepancies** [Jitkrittum, Xu, Szabó, Fukumizu, and Gretton, 2017]: low-rank kernel, linear runtime (but convergence control unclear)
     - **Random feature Stein discrepancies** [Huggins and Mackey, 2018]: stochastic low-rank kernel, near-linear runtime + high probability convergence control when $(Q_n)_{n \geq 1}$ moments uniformly bounded

Exploring the impact of Stein operator choice

An infinite number of operators $T$ characterize $P$

How is discrepancy impacted? How do we select the best $T$?

Thm: If $\nabla \log p$ bounded and $k \in C^{(1,1)}$, then $S(Q_n, T_P, G_k) \to 0$ for some $(Q_n)_{n \geq 1}$ not converging to $P$

Diffusion Stein operators $(T_g)(x) = \frac{1}{p(x)} \langle \nabla, p(x) a(x) g(x) \rangle$ of Gorham, Duncan, Vollmer, and Mackey [2019] may be appropriate for heavy tails
Future Directions

Many opportunities for future development

1. Improving scalability while maintaining convergence control
   - Subsampling of likelihood terms in $\nabla \log p$
   - **Stochastic Stein discrepancies** [Gorham, Raj, and Mackey, 2020]: control convergence with probability 1
   - Inexpensive approximations of kernel matrix
     - **Finite set Stein discrepancies** [Jitkrittum, Xu, Szabó, Fukumizu, and Gretton, 2017]: low-rank kernel, linear runtime (but convergence control unclear)
     - **Random feature Stein discrepancies** [Huggins and Mackey, 2018]: stochastic low-rank kernel, near-linear runtime + high probability convergence control when $(Q_n)_{n \geq 1}$ moments uniformly bounded

2. Exploring the impact of Stein operator choice
   - An infinite number of operators $\mathcal{T}$ characterize $P$
   - How is discrepancy impacted? How do we select the best $\mathcal{T}$?
Many opportunities for future development

1. Improving scalability while maintaining convergence control
   - Subsampling of likelihood terms in $\nabla \log p$
   - **Stochastic Stein discrepancies** [Gorham, Raj, and Mackey, 2020]: control convergence with probability 1
   - Inexpensive approximations of kernel matrix
     - **Finite set Stein discrepancies** [Jitkrittum, Xu, Szabó, Fukumizu, and Gretton, 2017]: low-rank kernel, linear runtime (but convergence control unclear)
     - **Random feature Stein discrepancies** [Huggins and Mackey, 2018]: stochastic low-rank kernel, near-linear runtime + high probability convergence control when $(Q_n)_{n \geq 1}$ moments uniformly bounded

2. Exploring the impact of Stein operator choice
   - An infinite number of operators $T$ characterize $P$
   - How is discrepancy impacted? How do we select the best $T$?
   - **Thm:** If $\nabla \log p$ bounded and $k \in C_0^{(1,1)}$, then
     
     $$S(Q_n, T_P, G_k) \rightarrow 0$$
     
     for some $(Q_n)_{n \geq 1}$ not converging to $P$
Future Directions

Many opportunities for future development

1. Improving scalability while maintaining convergence control
   - Subsampling of likelihood terms in $\nabla \log p$
   - **Stochastic Stein discrepancies** [Gorham, Raj, and Mackey, 2020]: control convergence with probability 1
   - Inexpensive approximations of kernel matrix
     - **Finite set Stein discrepancies** [Jitkrittum, Xu, Szabó, Fukumizu, and Gretton, 2017]: low-rank kernel, linear runtime (but convergence control unclear)
     - **Random feature Stein discrepancies** [Huggins and Mackey, 2018]: stochastic low-rank kernel, near-linear runtime + high probability convergence control when $(Q_n)_{n \geq 1}$ moments uniformly bounded

2. Exploring the impact of Stein operator choice
   - An infinite number of operators $\mathcal{T}$ characterize $P$
   - How is discrepancy impacted? How do we select the best $\mathcal{T}$?
   - **Thm**: If $\nabla \log p$ bounded and $k \in C_0^{(1,1)}$, then
     $S(Q_n, \mathcal{T}_P, \mathcal{G}_k) \to 0$ for some $(Q_n)_{n \geq 1}$ not converging to $P$
   - **Diffusion Stein operators** $(\mathcal{T}g)(x) = \frac{1}{p(x)}\langle \nabla, p(x)a(x)g(x) \rangle$ of Gorham, Duncan, Vollmer, and Mackey [2019] may be appropriate for heavy tails
Future Directions

Many opportunities for future development

- Addressing other inferential tasks
- Training generative adversarial networks [Wang and Liu, 2016] and variational autoencoders [Pu, Gan, Henao, Li, Han, and Carin, 2017]

To minimize $f(x)$, choose $a(x) \preceq cI$ with $a(x) \nabla f(x) \text{Lipschitz and distant dissipative}$

$$\langle a(x) \nabla f(x) - a(y) \nabla f(y), x - y \rangle \geq k \quad \text{for} \quad \|x - y\|_2 \geq r$$

Approximate target sequence $p_n(x) \propto e^{-\gamma_n f(x)}$ using Markov chain $x_{n+1} \sim N(x_n - \epsilon_n^2 a(x_n) \nabla f(x_n) + \epsilon_n^2 \gamma_n \langle \nabla f, a(x_n) \rangle)$

Thm: $\min_{1 \leq i \leq n} E f(x_i) \rightarrow \min_x f(x)$ (with explicit error bounds) for appropriate $\epsilon_n$ and $\gamma_n$ when $\nabla f, \nabla a, \text{and } a \frac{1}{2}$ are Lipschitz.
Many opportunities for future development

- Addressing other inferential tasks
  - Training generative adversarial networks [Wang and Liu, 2016] and variational autoencoders [Pu, Gan, Henao, Li, Han, and Carin, 2017]
  - Non-convex optimization

Example (Optimization with Discretized Diffusions [Erdogdu, Mackey, and Shamir, 2018])

- To minimize $f(x)$, choose $a(x) \geq cI$ with $a(x) \nabla f(x)$ Lipschitz and distantly dissipative
  \[
  \left( \frac{\langle a(x) \nabla f(x) - a(y) \nabla f(y), x - y \rangle}{\|x - y\|_2^2} \right) \geq k \quad \text{for} \quad \|x - y\|_2 \geq r
  \]
Many opportunities for future development

- Addressing other inferential tasks
  - Training generative adversarial networks [Wang and Liu, 2016] and variational autoencoders [Pu, Gan, Henao, Li, Han, and Carin, 2017]
  - Non-convex optimization

Example (Optimization with Discretized Diffusions [Erdogdu, Mackey, and Shamir, 2018])

- To minimize $f(x)$, choose $a(x) \succeq cI$ with $a(x) \nabla f(x)$ Lipschitz and distantly dissipative
  \[
  \left( \frac{\langle a(x) \nabla f(x) - a(y) \nabla f(y), x - y \rangle}{\|x - y\|^2_2} \right) \geq k \text{ for } \|x - y\|_2 \geq r
  \]
- Approximate target sequence $p_n(x) \propto e^{-\gamma_n f(x)}$ using Markov chain
  \[
  x_{n+1} \sim \mathcal{N}(x_n - \frac{\epsilon_n}{2} a(x_n) \nabla f(x_n) + \frac{\epsilon_n}{2\gamma_n} \langle \nabla, a(x_n) \rangle, \frac{\epsilon_n}{\gamma_n} a(x_n))
  \]
Future Directions

Many opportunities for future development

- Addressing other inferential tasks
  - Training generative adversarial networks [Wang and Liu, 2016] and variational autoencoders [Pu, Gan, Henao, Li, Han, and Carin, 2017]
  - Non-convex optimization

Example (Optimization with Discretized Diffusions [Erdogdu, Mackey, and Shamir, 2018])

- To minimize $f(x)$, choose $a(x) \succ cI$ with $a(x)\nabla f(x)$ Lipschitz and distantly dissipative
  \[ \frac{\langle a(x)\nabla f(x) - a(y)\nabla f(y), x - y \rangle}{\|x - y\|^2_2} \geq k \text{ for } \|x - y\|_2 \geq r \]

- Approximate target sequence $p_n(x) \propto e^{-\gamma_n f(x)}$ using Markov chain
  \[ x_{n+1} \sim \mathcal{N}(x_n - \frac{\epsilon_n}{2} a(x_n) \nabla f(x_n) + \frac{\epsilon_n}{2\gamma_n} \langle \nabla, a(x_n) \rangle, \frac{\epsilon_n}{\gamma_n} a(x_n)) \]

- Thm: $\min_{1 \leq i \leq n} \mathbb{E} f(x_i) \to \min_x f(x)$ (with explicit error bounds) for appropriate $\epsilon_n$ and $\gamma_n$ when $\nabla f, \nabla a$, and $a^{1/2}$ are Lipschitz
Many opportunities for future development

- Addressing other inferential tasks
  - Training generative adversarial networks [Wang and Liu, 2016] and variational autoencoders [Pu, Gan, Henao, Li, Han, and Carin, 2017]
  - Non-convex optimization [Erdogdu, Mackey, and Shamir, 2018]

\[
\min_x f(x) = 5 \log(1 + \frac{1}{2} \|x\|_2^2), \quad a(x) = (1 + \frac{1}{2} \|x\|_2^2) I, \quad a(x) \nabla f(x) = 5x
\]
Future Directions

Many opportunities for future development

1. Improving scalability while maintaining convergence control
   - Subsampling of likelihood terms in $\nabla \log p$
     - **Stochastic Stein discrepancies** [Gorham, Raj, and Mackey, 2020]
     - Inexpensive approximations of kernel matrix
       - **Finite set Stein discrepancies** [Jitkrittum, Xu, Szabó, Fukumizu, and Gretton, 2017]
       - **Random feature Stein discrepancies** [Huggins and Mackey, 2018]

2. Exploring the impact of Stein operator choice
   - An infinite number of operators $\mathcal{T}$ characterize $P$
   - How is discrepancy impacted? How do we select the best $\mathcal{T}$?
   - **Diffusion Stein operators** $(\mathcal{T} g)(x) = \frac{1}{p(x)} \langle \nabla, p(x)m(x)g(x) \rangle$ of Gorham, Duncan, Vollmer, and Mackey [2019] may be appropriate for heavy tails

3. Addressing other inferential tasks
   - **Generative modeling** [Wang and Liu, 2016, Pu, Gan, Henao, Li, Han, and Carin, 2017]
   - **Non-convex optimization** [Erdogdu, Mackey, and Shamir, 2018]
   - **Parameter estimation** [Barp, Briol, Duncan, Girolami, and Mackey, 2019]
   - **MCMC thinning** [Riabiz, Chen, Cockayne, Swietach, Niederer, Mackey, and Oates, 2020]
References


### Comparing Discrepancies

**Left:** Samples drawn i.i.d. from either the bimodal Gaussian mixture target $p(x) \propto e^{-\frac{1}{2}(x+1.5)^2} + e^{-\frac{1}{2}(x-1.5)^2}$ or a single mixture component.

**Right:** Discrepancy computation time using $d$ cores in $d$ dimensions.
The Importance of Kernel Choice

- Target $P = \mathcal{N}(0, I_d)$
- Off-target $Q_n$ has all $\|x_i\|_2 \leq 2n^{1/d} \log n$, $\|x_i - x_j\|_2 \geq 2 \log n$
- Gaussian and Matérn KSDs driven to 0 by an off-target sequence that does not converge to $P$
- IMQ KSD $(\beta = -\frac{1}{2}, c = 1)$ does not have this deficiency
Selecting Sampler Hyperparameters

**Setup** [Welling and Teh, 2011]

- Consider the posterior distribution $P$ induced by $L$ datapoints $y_l$ drawn i.i.d. from a Gaussian mixture likelihood
  $$Y_l|X \overset{iid}{\sim} \frac{1}{2}N(X_1, 2) + \frac{1}{2}N(X_1 + X_2, 2)$$

  under Gaussian priors on the parameters $X \in \mathbb{R}^2$
  $$X_1 \sim N(0, 10) \perp \perp X_2 \sim N(0, 1)$$

- Draw $m = 100$ datapoints $y_l$ with parameters $(x_1, x_2) = (0, 1)$
- Induces posterior with second mode at $(x_1, x_2) = (1, -1)$

- For range of parameters $\epsilon$, run approximate slice sampling for 148000 datapoint likelihood evaluations and store resulting posterior sample $Q_n$

- Use minimum IMQ KSD ($\beta = -\frac{1}{2}, c = 1$) to select appropriate $\epsilon$
  - Compare with standard MCMC parameter selection criterion, effective sample size (ESS), a measure of Markov chain autocorrelation
  - Compute median of diagnostic over 50 random sequences
Selecting Samplers

Setup

- **MNIST handwritten digits** [Ahn, Korattikara, and Welling, 2012]
  - 10000 images, 51 features, binary label indicating whether image of a 7 or a 9
- Bayesian logistic regression posterior $P$
  - $L$ independent observations $(y_l, v_l) \in \{1, -1\} \times \mathbb{R}^d$ with
    \[
    P(Y_l = 1|v_l, X) = 1/(1 + \exp(-\langle v_l, X \rangle))
    \]
  - Flat improper prior on the parameters $X \in \mathbb{R}^d$
- Use IMQ KSD ($\beta = -\frac{1}{2}, c = 1$) to compare SGFS-f to SGFS-d
  - drawing $10^5$ sample points and discarding first half as burn-in
- For external support, compare bivariate marginal means and 95% confidence ellipses with surrogate ground truth Hamiltonian Monte chain with $10^5$ sample points [Ahn, Korattikara, and Welling, 2012]
The Importance of Tightness

Goal: Show $S(Q_n, T_P, G_k) \rightarrow 0$ only if $Q_n$ converges to $P$

- A sequence $(Q_n)_{n \geq 1}$ is **uniformly tight** if for every $\epsilon > 0$, there is a finite number $R(\epsilon)$ such that $\sup_n Q_n(\|X\|_2 > R(\epsilon)) \leq \epsilon$
  - Intuitively, no mass in the sequence escapes to infinity
The Importance of Tightness

**Goal:** Show $S(Q_n, T_P, G_k) \to 0$ only if $Q_n$ converges to $P$

- A sequence $(Q_n)_{n \geq 1}$ is **uniformly tight** if for every $\epsilon > 0$, there is a finite number $R(\epsilon)$ such that $\sup_n Q_n(\|X\|_2 > R(\epsilon)) \leq \epsilon$
  - Intuitively, no mass in the sequence escapes to infinity

---

**Theorem (KSD detects tight non-convergence [Gorham and Mackey, 2017])**

Suppose that $P \in \mathcal{P}$ and $k(x, y) = \Phi(x - y)$ for $\Phi \in C^2$ with a non-vanishing generalized Fourier transform. If $(Q_n)_{n \geq 1}$ is uniformly tight and $S(Q_n, T_P, G_k) \to 0$, then $(Q_n)_{n \geq 1}$ converges weakly to $P$.

- **Good news**, but, ideally, KSD would detect non-tight sequences automatically...