# Sampling with Stein Discrepancies

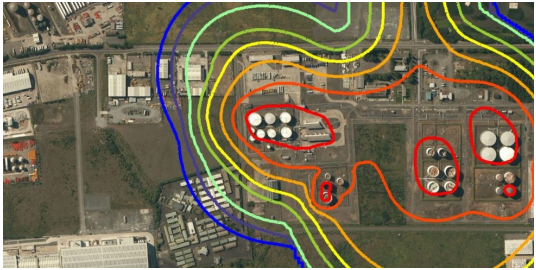
Chris. J. Oates

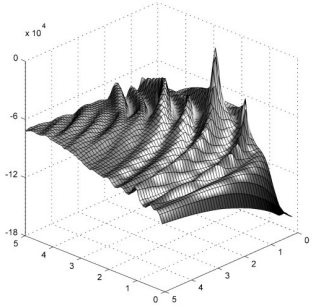
July 2022 @ MCQMC

Newcastle University

# Motivation: Bayesian Inference and Sampling

The result of integrating expert knowledge with experimental measurement is a *posterior* distribution that is implicitly defined:



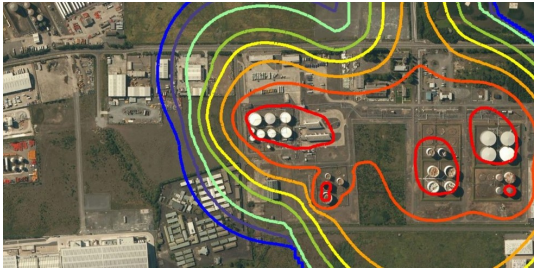


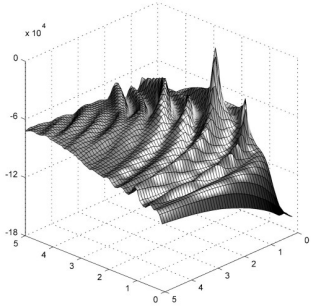

The (lack of) power of computational methodology for Bayesian inference forms a bottleck, imposing a limit of the sophistication of statistical models, or types of information, than can be integrated into a scientific investigation.

This tutorial concerns a new mathematical tool, *Stein discrepancy*, which has the potential to increase the range and scope of statistical analyses that can be performed.

# Motivation: Bayesian Inference and Sampling

The result of integrating expert knowledge with experimental measurement is a *posterior* distribution that is implicitly defined:



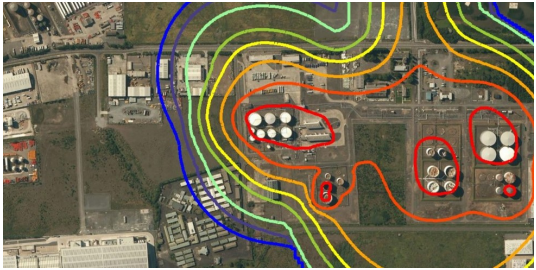


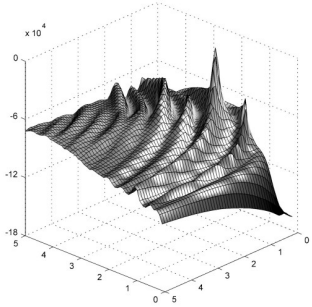

The (lack of) power of computational methodology for Bayesian inference forms a bottleck, imposing a limit of the sophistication of statistical models, or types of information, than can be integrated into a scientific investigation.

This tutorial concerns a new mathematical tool, *Stein discrepancy*, which has the potential to increase the range and scope of statistical analyses that can be performed.

# Motivation: Bayesian Inference and Sampling

The result of integrating expert knowledge with experimental measurement is a *posterior* distribution that is implicitly defined:






The (lack of) power of computational methodology for Bayesian inference forms a bottleck, imposing a limit of the sophistication of statistical models, or types of information, than can be integrated into a scientific investigation.

This tutorial concerns a new mathematical tool, *Stein discrepancy*, which has the potential to increase the range and scope of statistical analyses that can be performed.

# Outline of the Tutorial

# Measuring Sample Quality with Kernels

# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.

Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(x_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]

The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

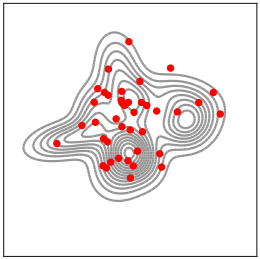
# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.

Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]



The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

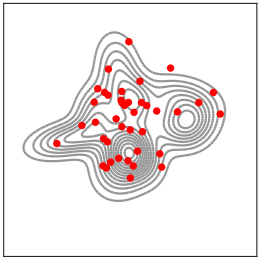
# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.

Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]



The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

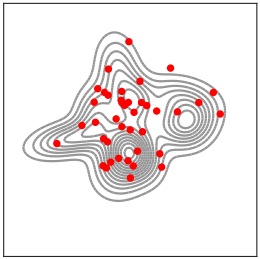
# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.

Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]



The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

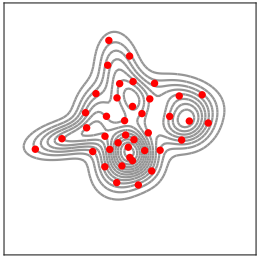
# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.

Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]



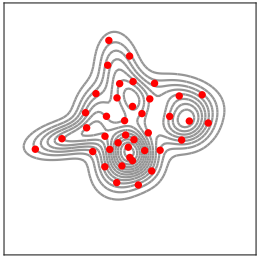
The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.



Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]

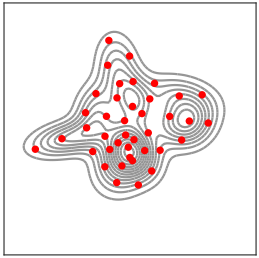
The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.



Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]

The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

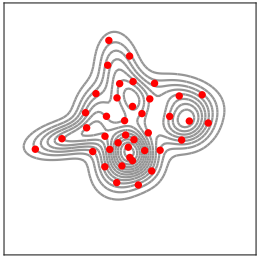
# Sampling

Consider general probability distributions $P$ on general domains $\mathcal{X}$.

Interested in *sampling*, which in this tutorial means approximation of the form

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i)$$

for some weights $w_i \in \mathbb{R}$ and some states $x_i \in \mathcal{X}$.
[Motivation: Forward UQ.]



The quality of the approximation should be close to optimal (in some sense to be specified) for the number $n$ of states used. [Note: This is not independent sampling from $P$.]

Markov chain Monte Carlo widely used but "typically" requires $n \approx 10^3$ or $10^4$ (to run diagnostics).

Quasi Monte Carlo is highly effective for specific $P$. [Motivation: General, implicitly defined $P$.]

# Measuring the Quality of a Sample

One of the most basic operations that one could hope to perform with a probability distribution is to compute expectations; i.e. to compute integrals of the form

$$\int f \mathrm{d}P, \qquad \text{or} \qquad \int f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

if the probability distribution $P$ admits a probability density function (PDF) $p(\boldsymbol{x})$.

For implicitly defined $P$, such integrals do not possess a closed form and numerical integration (also called *cubature*) will be required.

Using a sample, we arrive at a natural approximation

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \int f \mathrm{d}P \approx \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

to the integral. This suggests designing a sample for which the *cubature error*

$$\int f \mathrm{d}P - \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

is small. [Problem: Small cubature error could happen by "chance".]

# Measuring the Quality of a Sample

One of the most basic operations that one could hope to perform with a probability distribution is to compute expectations; i.e. to compute integrals of the form

$$\int f \mathrm{d}P, \qquad \text{or} \qquad \int f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

if the probability distribution $P$ admits a PDF $p(\boldsymbol{x})$.

For implicitly defined $P$, such integrals do not possess a closed form and numerical integration (also called *cubature*) will be required.

Using a sample, we arrive at a natural approximation

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \int f \mathrm{d}P \approx \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

to the integral. This suggests designing a sample for which the *cubature error*

$$\int f \mathrm{d}P - \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

is small. [Problem: Small cubature error could happen by "chance".]

# Measuring the Quality of a Sample

One of the most basic operations that one could hope to perform with a probability distribution is to compute expectations; i.e. to compute integrals of the form

$$\int f \mathrm{d}P, \qquad \text{or} \qquad \int f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

if the probability distribution $P$ admits a PDF $p(\boldsymbol{x})$.

For implicitly defined $P$, such integrals do not possess a closed form and numerical integration (also called *cubature*) will be required.

Using a sample, we arrive at a natural approximation

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \int f \mathrm{d}P \approx \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

to the integral. This suggests designing a sample for which the *cubature error*

$$\int f \mathrm{d}P - \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

is small. [Problem: Small cubature error could happen by "chance".]

# Measuring the Quality of a Sample

One of the most basic operations that one could hope to perform with a probability distribution is to compute expectations; i.e. to compute integrals of the form

$$\int f \mathrm{d}P, \qquad \text{or} \qquad \int f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

if the probability distribution $P$ admits a PDF $p(\boldsymbol{x})$.

For implicitly defined $P$, such integrals do not possess a closed form and numerical integration (also called *cubature*) will be required.

Using a sample, we arrive at a natural approximation

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \int f \mathrm{d}P \approx \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

to the integral. This suggests designing a sample for which the *cubature error*

$$\int f \mathrm{d}P - \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

is small. [Problem: Small cubature error could happen by "chance".]

# Measuring the Quality of a Sample

One of the most basic operations that one could hope to perform with a probability distribution is to compute expectations; i.e. to compute integrals of the form

$$\int f \mathrm{d}P, \qquad \text{or} \qquad \int f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

if the probability distribution $P$ admits a PDF $p(\boldsymbol{x})$.

For implicitly defined $P$, such integrals do not possess a closed form and numerical integration (also called *cubature*) will be required.

Using a sample, we arrive at a natural approximation

$$P \approx \sum_{i=1}^{n} w_i \delta(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \int f \mathrm{d}P \approx \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

to the integral. This suggests designing a sample for which the *cubature error*

$$\int f \mathrm{d}P - \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i)$$

is small. [Problem: Small cubature error could happen by "chance".]

# Cubature Error Representer

The basic idea is as follows: we consider the set $\mathcal{S}(k)$ of all functions of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i),$$

where $k$ is to be specified, the $\boldsymbol{y}_i$ are fixed states, and $n \in \mathbb{N}$. The function $k$ determines the regularity of the elements in $\mathcal{S}(k)$:

Figure: The left panel represents functions built from $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|)$, which is non-differentiable, while the right panel corresponds to functions built from $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2)$, which is smooth.

# Cubature Error Representer

The basic idea is as follows: we consider the set $\mathcal{S}(k)$ of all functions of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i),$$

where $k$ is to be specified, the $\boldsymbol{y}_i$ are fixed states, and $n \in \mathbb{N}$. The function $k$ determines the regularity of the elements in $\mathcal{S}(k)$:

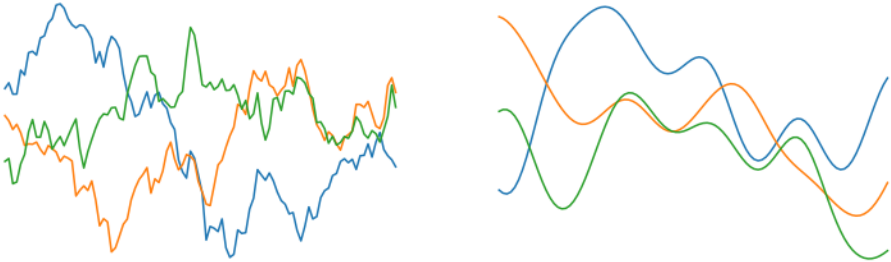


Figure: The left panel represents functions built from $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|)$, which is non-differentiable, while the right panel corresponds to functions built from $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2)$, which is smooth.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x}) \\ &= \Bigg\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)}\Bigg\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \int f(x)\mathrm{d}P(x) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, x_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, x)\rangle_{\mathcal{S}(k)}\mathrm{d}P(x) \\ &= \left\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, x_i) - \int k(\cdot, x)\mathrm{d}P(x)}_{=e(\cdot)} \right\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x}) \\ &= \Bigg\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)}\Bigg\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x})$$

$$= \left\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)} \right\rangle_{\mathcal{S}(k)},$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x}) \\ &= \Bigg\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)}\Bigg\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x}) \\ &= \left\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)}\right\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x}) \\ &= \Bigg\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)}\Bigg\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Cubature Error Representer

$\mathcal{S}(k)$ is a vector space (over the reals) of functions when (pointwise) addition and scalar multiplication are defined. In addition to that, we will want to make use of an inner product

$$\langle f, g\rangle_{\mathcal{S}(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_i c_j k(\boldsymbol{y}_i, \boldsymbol{z}_j), \quad f(\boldsymbol{x}) = \sum_{i=1}^{m} b_i k(\boldsymbol{x}, \boldsymbol{y}_i), \quad g(\boldsymbol{x}) = \sum_{j=1}^{n} c_j k(\boldsymbol{x}, \boldsymbol{z}_j),$$

for which we must require that $k$ is *symmetric* (i.e. $\langle f, g\rangle_{\mathcal{S}(k)} = \langle g, f\rangle_{\mathcal{S}(k)}$) and *positive definite* (i.e. $\langle f, f\rangle_{\mathcal{S}(k)} > 0$ for all $f \neq 0$). This inner product is useful because it satisfies a *reproducing property*, meaning that

$$\langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)} = f(\boldsymbol{x}),$$

and suggesting the formal manipulation

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \int f(\boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) &= \frac{1}{n}\sum_{i=1}^{n}\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{S}(k)} - \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{S}(k)}\mathrm{d}P(\boldsymbol{x}) \\ &= \Bigg\langle f, \underbrace{\frac{1}{n}\sum_{i=1}^{n} k(\cdot, \boldsymbol{x}_i) - \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})}_{=e(\cdot)}\Bigg\rangle_{\mathcal{S}(k)},\end{aligned}$$

where the *cubature error representer* $e(\cdot)$ is indepenent of $f$. Idea: Seek sample with $\|e\|_{\mathcal{S}(k)}$ small.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

**Definition 1 (Reproducing kernel Hilbert space)**

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *reproducing kernel Hilbert space (RKHS)* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, x) \in \mathcal{H}(k)$ for all $x \in \mathcal{X}$
2. $\langle f, k(\cdot, x) \rangle_{\mathcal{H}(k)} = f(x)$ for all $x \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

## Definition 1 (Reproducing kernel Hilbert space)

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *RKHS* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, x) \in \mathcal{H}(k)$ for all $x \in \mathcal{X}$
2. $\langle f, k(\cdot, x) \rangle_{\mathcal{H}(k)} = f(x)$ for all $x \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

## Definition 1 (Reproducing kernel Hilbert space)

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *RKHS* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, x) \in \mathcal{H}(k)$ for all $x \in \mathcal{X}$
2. $\langle f, k(\cdot, x) \rangle_{\mathcal{H}(k)} = f(x)$ for all $x \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

## Definition 1 (Reproducing kernel Hilbert space)

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *RKHS* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, \boldsymbol{x}) \in \mathcal{H}(k)$ for all $\boldsymbol{x} \in \mathcal{X}$
2. $\langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}(k)} = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

### Definition 1 (Reproducing kernel Hilbert space)

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *RKHS* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, \boldsymbol{x}) \in \mathcal{H}(k)$ for all $\boldsymbol{x} \in \mathcal{X}$
2. $\langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}(k)} = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

### Definition 1 (Reproducing kernel Hilbert space)

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *RKHS* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, \boldsymbol{x}) \in \mathcal{H}(k)$ for all $\boldsymbol{x} \in \mathcal{X}$
2. $\langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}(k)} = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

The main mathematical tool that we will exploit is that of a *reproducing kernel*:

### Definition 1 (Reproducing kernel Hilbert space)

Let $\mathcal{X}$ be a set and consider a symmetric and positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then a *RKHS* with *reproducing kernel* (or simply *kernel*) $k$ is an inner product space $\mathcal{H}(k)$ of functions $f : \mathcal{X} \to \mathbb{R}$, such that

1. $k(\cdot, \boldsymbol{x}) \in \mathcal{H}(k)$ for all $\boldsymbol{x} \in \mathcal{X}$
2. $\langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}(k)} = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ and all $f \in \mathcal{H}(k)$.

- Given a symmetric positive definite function $k$, it can be shown that there exists a unique RKHS $\mathcal{H}(k)$.
- Conversely, each RKHS admits a unique reproducing kernel, and that kernel is symmetric and positive definite.

# Reproducing Kernel Hilbert Spaces

In general it is difficult to characterise the inner product induced by a reproducing kernel, and hence the elements of the RKHS. However, there are a number of important cases where this can be carried out:

## Example 2

The linear span of a finite collection of functions $e_1(x), \ldots, e_p(x)$ can be endowed with the structure of an RKHS with reproducing kernel

$$k(x, y) = \sum_{i=1}^{p} e_i(x) e_i(y).$$

The induced inner product is

$$\langle f, g \rangle_{\mathcal{H}(k)} = b_1 c_1 + \cdots + b_p c_p, \qquad f(x) = \sum_{i=1}^{n} b_i e_i(x), \qquad g(x) = \sum_{i=1}^{n} c_i e_i(x).$$

## Example 3

The kernel

$$k(x, y) = \prod_{i=1}^{d} \left(1 + \min(1 - x_i, 1 - y_i)\right), \qquad x, y \in [0, 1]^d$$

reproduces a Hilbert space with inner product

$$\langle f, g \rangle_{\mathcal{H}(k)} = \sum_{\mathrm{u} \subseteq \{1, \ldots, d\}} \int_{[0,1]^d} \frac{\partial^{|\mathrm{u}|} f}{\partial x_{\mathrm{u}}}(x_{\mathrm{u}}, \mathbf{1}) \frac{\partial^{|\mathrm{u}|} g}{\partial x_{\mathrm{u}}}(x_{\mathrm{u}}, \mathbf{1}) \mathrm{d}x_{\mathrm{u}}$$

that features in the analysis of classical quasi Monte Carlo.

# Reproducing Kernel Hilbert Spaces

In general it is difficult to characterise the inner product induced by a reproducing kernel, and hence the elements of the RKHS. However, there are a number of important cases where this can be carried out:

## Example 2

The linear span of a finite collection of functions $e_1(\boldsymbol{x}), \ldots, e_p(\boldsymbol{x})$ can be endowed with the structure of an RKHS with reproducing kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{p} e_i(\boldsymbol{x}) e_i(\boldsymbol{y}).$$

The induced inner product is

$$\langle f, g \rangle_{\mathcal{H}(k)} = b_1 c_1 + \cdots + b_p c_p, \qquad f(\boldsymbol{x}) = \sum_{i=1}^{n} b_i e_i(\boldsymbol{x}), \qquad g(\boldsymbol{x}) = \sum_{i=1}^{n} c_i e_i(\boldsymbol{x}).$$

## Example 3

The kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{d} \left(1 + \min(1 - x_i, 1 - y_i)\right), \qquad \boldsymbol{x}, \boldsymbol{y} \in [0, 1]^d$$

reproduces a Hilbert space with inner product

$$\langle f, g \rangle_{\mathcal{H}(k)} = \sum_{\mathrm{u} \subseteq \{1, \ldots, d\}} \int_{[0,1]^d} \frac{\partial^{|\mathrm{u}|} f}{\partial \boldsymbol{x}_{\mathrm{u}}}(\boldsymbol{x}_{\mathrm{u}}, \mathbf{1}) \frac{\partial^{|\mathrm{u}|} g}{\partial \boldsymbol{x}_{\mathrm{u}}}(\boldsymbol{x}_{\mathrm{u}}, \mathbf{1}) \mathrm{d}\boldsymbol{x}_{\mathrm{u}}$$

that features in the analysis of classical quasi Monte Carlo.

# Reproducing Kernel Hilbert Spaces

In general it is difficult to characterise the inner product induced by a reproducing kernel, and hence the elements of the RKHS. However, there are a number of important cases where this can be carried out:

## Example 2

The linear span of a finite collection of functions $e_1(\boldsymbol{x}), \ldots, e_p(\boldsymbol{x})$ can be endowed with the structure of an RKHS with reproducing kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{p} e_i(\boldsymbol{x}) e_i(\boldsymbol{y}).$$

The induced inner product is

$$\langle f, g \rangle_{\mathcal{H}(k)} = b_1 c_1 + \cdots + b_p c_p, \qquad f(\boldsymbol{x}) = \sum_{i=1}^{n} b_i e_i(\boldsymbol{x}), \qquad g(\boldsymbol{x}) = \sum_{i=1}^{n} c_i e_i(\boldsymbol{x}).$$

## Example 3

The kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{d} \left(1 + \min(1 - x_i, 1 - y_i)\right), \qquad \boldsymbol{x}, \boldsymbol{y} \in [0, 1]^d$$

reproduces a Hilbert space with inner product

$$\langle f, g \rangle_{\mathcal{H}(k)} = \sum_{\mathfrak{u} \subseteq \{1, \ldots, d\}} \int_{[0,1]^d} \frac{\partial^{|\mathfrak{u}|} f}{\partial \boldsymbol{x}_{\mathfrak{u}}}(\boldsymbol{x}_{\mathfrak{u}}, \mathbf{1}) \frac{\partial^{|\mathfrak{u}|} g}{\partial \boldsymbol{x}_{\mathfrak{u}}}(\boldsymbol{x}_{\mathfrak{u}}, \mathbf{1}) \mathrm{d}\boldsymbol{x}_{\mathfrak{u}}$$

that features in the analysis of classical quasi Monte Carlo.

# Reproducing Kernel Hilbert Spaces

In general it is difficult to characterise the inner product induced by a reproducing kernel, and hence the elements of the RKHS. However, there are a number of important cases where this can be carried out:

## Example 2

The linear span of a finite collection of functions $e_1(\boldsymbol{x}), \ldots, e_p(\boldsymbol{x})$ can be endowed with the structure of an RKHS with reproducing kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{p} e_i(\boldsymbol{x}) e_i(\boldsymbol{y}).$$

The induced inner product is

$$\langle f, g \rangle_{\mathcal{H}(k)} = b_1 c_1 + \cdots + b_p c_p, \qquad f(\boldsymbol{x}) = \sum_{i=1}^{n} b_i e_i(\boldsymbol{x}), \qquad g(\boldsymbol{x}) = \sum_{i=1}^{n} c_i e_i(\boldsymbol{x}).$$

## Example 3

The kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{d} \left(1 + \min(1 - x_i, 1 - y_i)\right), \qquad \boldsymbol{x}, \boldsymbol{y} \in [0, 1]^d$$

reproduces a Hilbert space with inner product

$$\langle f, g \rangle_{\mathcal{H}(k)} = \sum_{\mathrm{u} \subseteq \{1, \ldots, d\}} \int_{[0,1]^d} \frac{\partial^{|\mathrm{u}|} f}{\partial \boldsymbol{x}_{\mathrm{u}}}(\boldsymbol{x}_{\mathrm{u}}, \mathbf{1}) \frac{\partial^{|\mathrm{u}|} g}{\partial \boldsymbol{x}_{\mathrm{u}}}(\boldsymbol{x}_{\mathrm{u}}, \mathbf{1}) \mathrm{d}\boldsymbol{x}_{\mathrm{u}}$$

that features in the analysis of classical quasi Monte Carlo.

# Kernel Mean Embedding

Idea: Work with $\mathcal{H}(k)$ instead of $\mathcal{S}(k)$.

**Definition 4 (Kernel mean embedding)**

For a kernel $k$ and a probability distribution $P$, we call $\mu_P = \int k(\cdot, x) \mathrm{d}P(x)$ the *kernel mean embedding* of $P$ in $\mathcal{H}(k)$, whenever it is well-defined.

**Example 5**

For the kernel

$$k(x, y) = \sum_{i=1}^{p} e_i(x) e_i(y),$$

with $e_i \in L^1(P)$, we have kernel mean embedding

$$\mu_P(x) = \sum_{i=1}^{p} \Big( \underbrace{\int e_i(y) \, \mathrm{d}P(y)}_{<\infty} \Big) e_i(x) \in \mathcal{H}(k).$$

In general, when is the kernel mean embedding well-defined?

# Kernel Mean Embedding

Idea: Work with $\mathcal{H}(k)$ instead of $\mathcal{S}(k)$.

## Definition 4 (Kernel mean embedding)

For a kernel $k$ and a probability distribution $P$, we call $\mu_P = \int k(\cdot, x)\mathrm{d}P(x)$ the *kernel mean embedding* of $P$ in $\mathcal{H}(k)$, whenever it is well-defined.

## Example 5

For the kernel

$$k(x, y) = \sum_{i=1}^{p} e_i(x)e_i(y),$$

with $e_i \in L^1(P)$, we have kernel mean embedding

$$\mu_P(x) = \sum_{i=1}^{p} \Big( \underbrace{\int e_i(y)\,\mathrm{d}P(y)}_{<\infty} \Big) e_i(x) \in \mathcal{H}(k).$$

In general, when is the kernel mean embedding well-defined?

# Kernel Mean Embedding

Idea: Work with $\mathcal{H}(k)$ instead of $\mathcal{S}(k)$.

### Definition 4 (Kernel mean embedding)

For a kernel $k$ and a probability distribution $P$, we call $\mu_P = \int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$ the *kernel mean embedding* of $P$ in $\mathcal{H}(k)$, whenever it is well-defined.

### Example 5

For the kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{p} e_i(\boldsymbol{x}) e_i(\boldsymbol{y}),$$

with $e_i \in L^1(P)$, we have kernel mean embedding

$$\mu_P(\boldsymbol{x}) = \sum_{i=1}^{p} \Big( \underbrace{\int e_i(\boldsymbol{y}) \, \mathrm{d}P(\boldsymbol{y})}_{<\infty} \Big) e_i(\boldsymbol{x}) \in \mathcal{H}(k).$$

In general, when is the kernel mean embedding well-defined?

# Kernel Mean Embedding

Idea: Work with $\mathcal{H}(k)$ instead of $\mathcal{S}(k)$.

## Definition 4 (Kernel mean embedding)

For a kernel $k$ and a probability distribution $P$, we call $\mu_P = \int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$ the *kernel mean embedding* of $P$ in $\mathcal{H}(k)$, whenever it is well-defined.

## Example 5

For the kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{p} e_i(\boldsymbol{x}) e_i(\boldsymbol{y}),$$

with $e_i \in L^1(P)$, we have kernel mean embedding

$$\mu_P(\boldsymbol{x}) = \sum_{i=1}^{p} \Big( \underbrace{\int e_i(\boldsymbol{y}) \, \mathrm{d}P(\boldsymbol{y})}_{<\infty} \Big) e_i(\boldsymbol{x}) \in \mathcal{H}(k).$$

In general, when is the kernel mean embedding well-defined?

# Kernel Mean Embedding

**Lemma 6**
*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**
Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot,x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot,x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

**Lemma 6**
*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**
Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot,x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot,x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

**Lemma 6**

*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**

Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot, x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot, x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

**Lemma 6**
*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**
Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot, x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot, x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

**Lemma 6**
*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**
Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot, x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot, x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

**Lemma 6**
*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**
Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot, x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot, x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

**Lemma 6**
*If* $\int \sqrt{k(x,x)}\mathrm{d}P(x) < \infty$ *then* $\mu_P(x) \in \mathcal{H}(k)$.

**Proof.**
Consider the linear operator $Lf = \int f(x)\mathrm{d}P(x)$ acting on $f \in \mathcal{H}(k)$. Claim that $L$ is a *bounded linear operator* from $\mathcal{H}(k)$ to $\mathbb{R}$. Indeed,

$$
\begin{aligned}
|Lf| = \left|\int f(x)\mathrm{d}P(x)\right| &\leq \int |f(x)|\mathrm{d}P(x) && (1)\\
&= \int |\langle f, k(\cdot, x)\rangle_{\mathcal{H}(k)}|\mathrm{d}P(x) && (2)\\
&\leq \int \|f\|_{\mathcal{H}(k)}\|k(\cdot, x)\|_{\mathcal{H}(k)}|\mathrm{d}P(x) && (3)\\
&= \int \sqrt{k(x,x)}\mathrm{d}P(x)\|f\|_{\mathcal{H}(k)} && (4)
\end{aligned}
$$

where (1) is Jensen's inequality, (2) is the reproducing property, (3) is Cauchy–Schwarz, and (4) is the reproducing property again.

# Kernel Mean Embedding

## Proof of Lemma 6, continued.

Thus, from the Riesz representation theorem, there exists $h \in \mathcal{H}(k)$ such that

$$Lf = \langle f, h \rangle_{\mathcal{H}(k)}.$$

Taking $f(x) = k(y, x)$ and using the reproducing property leads to

$$\begin{aligned} \int k(y, x) \mathrm{d}P(x) &= Lf \\ &= \langle f, h \rangle_{\mathcal{H}(k)} \\ &= h(y), \end{aligned}$$

so that $h(\cdot) = \int k(\cdot, x) \mathrm{d}P(x)$, and so $\mu_p = h \in \mathcal{H}(k)$, as was claimed.

## Standing Assumption 1

*For all reproducing kernels $k$ and probability distributions $P$ considered in the sequel, we assume that* $\int \sqrt{k(x, x)} \mathrm{d}P(x) < \infty$.

### Proof of Lemma 6, continued.

Thus, from the Riesz representation theorem, there exists $h \in \mathcal{H}(k)$ such that

$$Lf = \langle f, h \rangle_{\mathcal{H}(k)}.$$

Taking $f(x) = k(y, x)$ and using the reproducing property leads to

$$\begin{aligned} \int k(y, x) \mathrm{d}P(x) = Lf \\ &= \langle f, h \rangle_{\mathcal{H}(k)} \\ &= h(y), \end{aligned}$$

so that $h(\cdot) = \int k(\cdot, x) \mathrm{d}P(x)$, and so $\mu_p = h \in \mathcal{H}(k)$, as was claimed.

### Standing Assumption 1

*For all reproducing kernels $k$ and probability distributions $P$ considered in the sequel, we assume that $\int \sqrt{k(x, x)} \mathrm{d}P(x) < \infty$.*

# Kernel Mean Embedding

## Proof of Lemma 6, continued.

Thus, from the Riesz representation theorem, there exists $h \in \mathcal{H}(k)$ such that

$$Lf = \langle f, h \rangle_{\mathcal{H}(k)}.$$

Taking $f(x) = k(y, x)$ and using the reproducing property leads to

$$\begin{aligned} \int k(y, x) \mathrm{d}P(x) &= Lf \\ &= \langle f, h \rangle_{\mathcal{H}(k)} \\ &= h(y), \end{aligned}$$

so that $h(\cdot) = \int k(\cdot, x) \mathrm{d}P(x)$, and so $\mu_p = h \in \mathcal{H}(k)$, as was claimed.

## Standing Assumption 1

*For all reproducing kernels $k$ and probability distributions $P$ considered in the sequel, we assume that* $\int \sqrt{k(x, x)} \mathrm{d}P(x) < \infty$.

### Proof of Lemma 6, continued.

Thus, from the Riesz representation theorem, there exists $h \in \mathcal{H}(k)$ such that

$$Lf = \langle f, h \rangle_{\mathcal{H}(k)}.$$

Taking $f(x) = k(y, x)$ and using the reproducing property leads to

$$\begin{aligned} \int k(y, x) \mathrm{d}P(x) &= Lf \\ &= \langle f, h \rangle_{\mathcal{H}(k)} \\ &= h(y), \end{aligned}$$

so that $h(\cdot) = \int k(\cdot, x) \mathrm{d}P(x)$, and so $\mu_p = h \in \mathcal{H}(k)$, as was claimed.

### Standing Assumption 1

*For all reproducing kernels $k$ and probability distributions $P$ considered in the sequel, we assume that $\int \sqrt{k(x, x)} \mathrm{d}P(x) < \infty$.*

### Proof of Lemma 6, continued.

Thus, from the Riesz representation theorem, there exists $h \in \mathcal{H}(k)$ such that

$$Lf = \langle f, h \rangle_{\mathcal{H}(k)}.$$

Taking $f(x) = k(y, x)$ and using the reproducing property leads to

$$\begin{aligned} \int k(y, x) \mathrm{d}P(x) &= Lf \\ &= \langle f, h \rangle_{\mathcal{H}(k)} \\ &= h(y), \end{aligned}$$

so that $h(\cdot) = \int k(\cdot, x) \mathrm{d}P(x)$, and so $\mu_p = h \in \mathcal{H}(k)$, as was claimed.

### Standing Assumption 1

*For all reproducing kernels $k$ and probability distributions $P$ considered in the sequel, we assume that $\int \sqrt{k(x, x)} \mathrm{d}P(x) < \infty$.*

### Proof of Lemma 6, continued.

Thus, from the Riesz representation theorem, there exists $h \in \mathcal{H}(k)$ such that

$$Lf = \langle f, h \rangle_{\mathcal{H}(k)}.$$

Taking $f(x) = k(y, x)$ and using the reproducing property leads to

$$\begin{aligned} \int k(y, x) \mathrm{d}P(x) &= Lf \\ &= \langle f, h \rangle_{\mathcal{H}(k)} \\ &= h(y), \end{aligned}$$

so that $h(\cdot) = \int k(\cdot, x) \mathrm{d}P(x)$, and so $\mu_p = h \in \mathcal{H}(k)$, as was claimed.

### Standing Assumption 1

*For all reproducing kernels $k$ and probability distributions $P$ considered in the sequel, we assume that* $\int \sqrt{k(x, x)} \mathrm{d}P(x) < \infty$.

# Cubature Error Representer in RKHS

In this more general setting, the cubature error representer is the difference $e = \mu_{Q_n} - \mu_P$ of two kernel mean embeddings, where $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ is the discrete distribution on which the cubature rule is based. i.e.

$$\int f \mathrm{d}P - \sum_{i=1}^n w_i f(\boldsymbol{x}_i) = \langle f, \mu_P - \mu_{Q_n} \rangle_{\mathcal{H}(k)}, \quad \mu_P = \int k(\cdot, \boldsymbol{x}) \, \mathrm{d}P(\boldsymbol{x}), \quad \mu_{Q_n} = \sum_{i=1}^n w_i k(\cdot, \boldsymbol{x}_i).$$

There are several different ways to systematically assess the performance of a cubature rule, but here we focus on a *worst case* assessment:

**Definition 7 (Maximum mean discrepancy)**

The *maximum mean discrepancy (MMD)* between two distributions $P$ and $Q$ is

$$D_k(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}(k)} = \sup_{\|f\|_{\mathcal{H}(k)} \leq 1} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|,$$

also called the *worst case cubature error* in the unit ball of $\mathcal{H}(k)$.

# Cubature Error Representer in RKHS

In this more general setting, the cubature error representer is the difference $e = \mu_{Q_n} - \mu_P$ of two kernel mean embeddings, where $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ is the discrete distribution on which the cubature rule is based. i.e.

$$\int f \mathrm{d}P - \sum_{i=1}^n w_i f(\boldsymbol{x}_i) = \langle f, \mu_P - \mu_{Q_n} \rangle_{\mathcal{H}(k)}, \quad \mu_P = \int k(\cdot, \boldsymbol{x}) \, \mathrm{d}P(\boldsymbol{x}), \quad \mu_{Q_n} = \sum_{i=1}^n w_i k(\cdot, \boldsymbol{x}_i).$$

There are several different ways to systematically assess the performance of a cubature rule, but here we focus on a *worst case* assessment:

Definition 7 (Maximum mean discrepancy)

The *MMD* between two distributions $P$ and $Q$ is

$$D_k(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}(k)} = \sup_{\|f\|_{\mathcal{H}(k)} \leq 1} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|,$$

also called the *worst case cubature error* in the unit ball of $\mathcal{H}(k)$.

# Cubature Error Representer in RKHS

In this more general setting, the cubature error representer is the difference $e = \mu_{Q_n} - \mu_P$ of two kernel mean embeddings, where $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ is the discrete distribution on which the cubature rule is based. i.e.

$$\int f \mathrm{d}P - \sum_{i=1}^n w_i f(\boldsymbol{x}_i) = \langle f, \mu_P - \mu_{Q_n} \rangle_{\mathcal{H}(k)}, \quad \mu_P = \int k(\cdot, \boldsymbol{x}) \, \mathrm{d}P(\boldsymbol{x}), \quad \mu_{Q_n} = \sum_{i=1}^n w_i k(\cdot, \boldsymbol{x}_i).$$

There are several different ways to systematically assess the performance of a cubature rule, but here we focus on a *worst case* assessment:

## Definition 7 (Maximum mean discrepancy)

The *MMD* between two distributions $P$ and $Q$ is

$$D_k(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}(k)} = \sup_{\|f\|_{\mathcal{H}(k)} \leq 1} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|,$$

also called the *worst case cubature error* in the unit ball of $\mathcal{H}(k)$.

# Cubature Error Representer in RKHS

In this more general setting, the cubature error representer is the difference $e = \mu_{Q_n} - \mu_P$ of two kernel mean embeddings, where $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ is the discrete distribution on which the cubature rule is based. i.e.

$$\int f \mathrm{d}P - \sum_{i=1}^n w_i f(\boldsymbol{x}_i) = \langle f, \mu_P - \mu_{Q_n} \rangle_{\mathcal{H}(k)}, \quad \mu_P = \int k(\cdot, \boldsymbol{x}) \, \mathrm{d}P(\boldsymbol{x}), \quad \mu_{Q_n} = \sum_{i=1}^n w_i k(\cdot, \boldsymbol{x}_i).$$

There are several different ways to systematically assess the performance of a cubature rule, but here we focus on a *worst case* assessment:

## Definition 7 (Maximum mean discrepancy)

The *MMD* between two distributions $P$ and $Q$ is

$$D_k(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}(k)} = \sup_{\|f\|_{\mathcal{H}(k)} \leq 1} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|,$$

also called the *worst case cubature error* in the unit ball of $\mathcal{H}(k)$.

# Maximum Mean Discrepancy

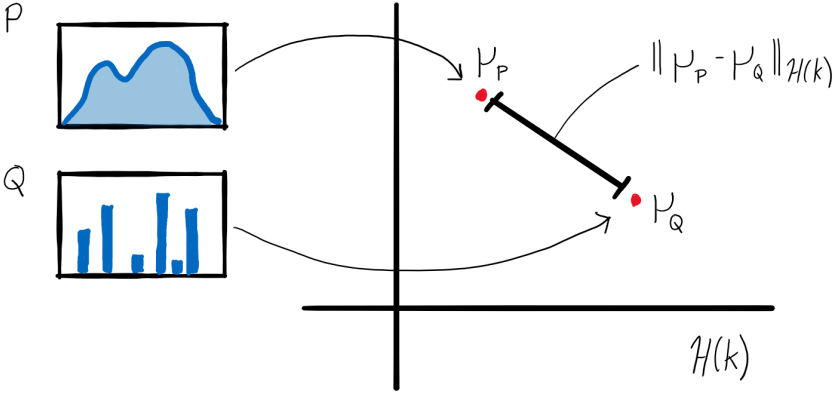



Figure: *Kernel mean embedding:* Two probability distributions $P$ and $Q$ are mapped to their respective elements $\mu_P$ and $\mu_Q$ in the RKHS $\mathcal{H}(k)$. The distance (in $\mathcal{H}(k)$) between these kernel mean embeddings $\mu_P$ and $\mu_Q$ is called the MMD between $P$ and $Q$.

# Maximum Mean Discrepancy

If $D_k(P, Q_n) = 0$, the cubature rule based on $Q_n$ will be exact for all integrands $f \in \mathcal{H}(k)$. Does this mean that $Q_n$ and $P$ are identical?

**Definition 8 (Characteristic kernel)**

A kernel $k$ is said to be *characteristic* if $D_k(P, Q) = 0$ implies $P = Q$.

**Example 9 (Polynomial kernel is not characteristic)**

From Example 2, the kernel $k(x, y) = \sum_{i=0}^{p} x^i y^i$ reproduces an RKHS whose elements are the polynomials of degree at most $p$ on the domain $\mathcal{X} = \mathbb{R}$. Thus $D_k(P, Q) = 0$ if and only if the moments $\int x^i \mathrm{d}P(x)$ and $\int x^i \mathrm{d}Q(x)$ are identical for $i = 1, \ldots, p$. In particular, $k$ is *not* a characteristic kernel.

**Example 10**

The Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2)$ is a characteristic kernel on $\mathcal{X} = \mathbb{R}^d$.

# Maximum Mean Discrepancy

If $D_k(P, Q_n) = 0$, the cubature rule based on $Q_n$ will be exact for all integrands $f \in \mathcal{H}(k)$. Does this mean that $Q_n$ and $P$ are identical?

## Definition 8 (Characteristic kernel)

A kernel $k$ is said to be *characteristic* if $D_k(P, Q) = 0$ implies $P = Q$.

## Example 9 (Polynomial kernel is not characteristic)

From Example 2, the kernel $k(x, y) = \sum_{i=0}^{p} x^i y^i$ reproduces an RKHS whose elements are the polynomials of degree at most $p$ on the domain $\mathcal{X} = \mathbb{R}$. Thus $D_k(P, Q) = 0$ if and only if the moments $\int x^i \mathrm{d}P(x)$ and $\int x^i \mathrm{d}Q(x)$ are identical for $i = 1, \ldots, p$. In particular, $k$ is *not* a characteristic kernel.

## Example 10

The Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2)$ is a characteristic kernel on $\mathcal{X} = \mathbb{R}^d$.

# Maximum Mean Discrepancy

If $D_k(P, Q_n) = 0$, the cubature rule based on $Q_n$ will be exact for all integrands $f \in \mathcal{H}(k)$. Does this mean that $Q_n$ and $P$ are identical?

### Definition 8 (Characteristic kernel)

A kernel $k$ is said to be *characteristic* if $D_k(P, Q) = 0$ implies $P = Q$.

### Example 9 (Polynomial kernel is not characteristic)

From Example 2, the kernel $k(x, y) = \sum_{i=0}^{p} x^i y^i$ reproduces an RKHS whose elements are the polynomials of degree at most $p$ on the domain $\mathcal{X} = \mathbb{R}$. Thus $D_k(P, Q) = 0$ if and only if the moments $\int x^i \mathrm{d}P(x)$ and $\int x^i \mathrm{d}Q(x)$ are identical for $i = 1, \ldots, p$. In particular, $k$ is *not* a characteristic kernel.

### Example 10

The Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2)$ is a characteristic kernel on $\mathcal{X} = \mathbb{R}^d$.

# Maximum Mean Discrepancy

If $D_k(P, Q_n) = 0$, the cubature rule based on $Q_n$ will be exact for all integrands $f \in \mathcal{H}(k)$. Does this mean that $Q_n$ and $P$ are identical?

### Definition 8 (Characteristic kernel)

A kernel $k$ is said to be *characteristic* if $D_k(P, Q) = 0$ implies $P = Q$.

### Example 9 (Polynomial kernel is not characteristic)

From Example 2, the kernel $k(x, y) = \sum_{i=0}^{p} x^i y^i$ reproduces an RKHS whose elements are the polynomials of degree at most $p$ on the domain $\mathcal{X} = \mathbb{R}$. Thus $D_k(P, Q) = 0$ if and only if the moments $\int x^i \mathrm{d}P(x)$ and $\int x^i \mathrm{d}Q(x)$ are identical for $i = 1, \ldots, p$. In particular, $k$ is *not* a characteristic kernel.

### Example 10

The Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2)$ is a characteristic kernel on $\mathcal{X} = \mathbb{R}^d$.

# Maximum Mean Discrepancy

**Notation:** Let $Q_n \Rightarrow P$ denote that the sequence $(Q_n)_{n=1}^{\infty}$ converges *weakly* (or *in distribution*) to $P$ (i.e. $\int f \mathrm{d}Q_n \to \int f \mathrm{d}P$ for all functions $f$ which are continuous and bounded).

### Definition 11 (Weak convergence control)

A kernel $k$ is said to have *weak convergence control* if $D_k(P, Q_n) \to 0$ implies that $Q_n \Rightarrow P$.

### Remark 1

*For a compact Hausdorff space $\mathcal{X}$, a bounded characteristic kernel $k$ has weak convergence control. This need not hold when the domain $\mathcal{X}$ is non-compact.*

Convergence control justifies attempting to minimise MMD for the purposes of quantisation and more general approximation, as we will attempt in the sequel.

### Example 12

The Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2)$ controls weak convergence of probability distributions on $\mathcal{X} = [0, 1]^d$. It can also be shown that the Gaussian kernel controls weak convergence on $\mathcal{X} = \mathbb{R}^d$; this can be deduced from e.g. Theorem 7 of Simon-Gabriel et al. [2020] and the general results in Sriperumbudur et al. [2011].

# Maximum Mean Discrepancy

**Notation:** Let $Q_n \Rightarrow P$ denote that the sequence $(Q_n)_{n=1}^{\infty}$ converges *weakly* (or *in distribution*) to $P$ (i.e. $\int f \mathrm{d}Q_n \to \int f \mathrm{d}P$ for all functions $f$ which are continuous and bounded).

## Definition 11 (Weak convergence control)

A kernel $k$ is said to have *weak convergence control* if $D_k(P, Q_n) \to 0$ implies that $Q_n \Rightarrow P$.

## Remark 1

*For a compact Hausdorff space $\mathcal{X}$, a bounded characteristic kernel $k$ has weak convergence control. This need not hold when the domain $\mathcal{X}$ is non-compact.*

Convergence control justifies attempting to minimise MMD for the purposes of quantisation and more general approximation, as we will attempt in the sequel.

## Example 12

The Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2)$ controls weak convergence of probability distributions on $\mathcal{X} = [0, 1]^d$. It can also be shown that the Gaussian kernel controls weak convergence on $\mathcal{X} = \mathbb{R}^d$; this can be deduced from e.g. Theorem 7 of Simon-Gabriel et al. [2020] and the general results in Sriperumbudur et al. [2011].

# Maximum Mean Discrepancy

**Notation:** Let $Q_n \Rightarrow P$ denote that the sequence $(Q_n)_{n=1}^{\infty}$ converges *weakly* (or *in distribution*) to $P$ (i.e. $\int f \mathrm{d}Q_n \to \int f \mathrm{d}P$ for all functions $f$ which are continuous and bounded).

## Definition 11 (Weak convergence control)

A kernel $k$ is said to have *weak convergence control* if $D_k(P, Q_n) \to 0$ implies that $Q_n \Rightarrow P$.

## Remark 1

*For a compact Hausdorff space $\mathcal{X}$, a bounded characteristic kernel $k$ has weak convergence control. This need not hold when the domain $\mathcal{X}$ is non-compact.*

Convergence control justifies attempting to minimise MMD for the purposes of quantisation and more general approximation, as we will attempt in the sequel.

## Example 12

The Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2)$ controls weak convergence of probability distributions on $\mathcal{X} = [0, 1]^d$. It can also be shown that the Gaussian kernel controls weak convergence on $\mathcal{X} = \mathbb{R}^d$; this can be deduced from e.g. Theorem 7 of Simon-Gabriel et al. [2020] and the general results in Sriperumbudur et al. [2011].

# Maximum Mean Discrepancy

**Notation:** Let $Q_n \Rightarrow P$ denote that the sequence $(Q_n)_{n=1}^{\infty}$ converges *weakly* (or *in distribution*) to $P$ (i.e. $\int f \mathrm{d}Q_n \to \int f \mathrm{d}P$ for all functions $f$ which are continuous and bounded).

## Definition 11 (Weak convergence control)

A kernel $k$ is said to have *weak convergence control* if $D_k(P, Q_n) \to 0$ implies that $Q_n \Rightarrow P$.

## Remark 1

*For a compact Hausdorff space $\mathcal{X}$, a bounded characteristic kernel $k$ has weak convergence control. This need not hold when the domain $\mathcal{X}$ is non-compact.*

Convergence control justifies attempting to minimise MMD for the purposes of quantisation and more general approximation, as we will attempt in the sequel.

## Example 12

The Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2)$ controls weak convergence of probability distributions on $\mathcal{X} = [0, 1]^d$. It can also be shown that the Gaussian kernel controls weak convergence on $\mathcal{X} = \mathbb{R}^d$; this can be deduced from e.g. Theorem 7 of Simon-Gabriel et al. [2020] and the general results in Sriperumbudur et al. [2011].

# Maximum Mean Discrepancy

**Notation:** Let $Q_n \Rightarrow P$ denote that the sequence $(Q_n)_{n=1}^{\infty}$ converges *weakly* (or *in distribution*) to $P$ (i.e. $\int f \mathrm{d}Q_n \to \int f \mathrm{d}P$ for all functions $f$ which are continuous and bounded).

## Definition 11 (Weak convergence control)

A kernel $k$ is said to have *weak convergence control* if $D_k(P, Q_n) \to 0$ implies that $Q_n \Rightarrow P$.

## Remark 1

*For a compact Hausdorff space $\mathcal{X}$, a bounded characteristic kernel $k$ has weak convergence control. This need not hold when the domain $\mathcal{X}$ is non-compact.*

Convergence control justifies attempting to minimise MMD for the purposes of quantisation and more general approximation, as we will attempt in the sequel.

## Example 12

The Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2)$ controls weak convergence of probability distributions on $\mathcal{X} = [0, 1]^d$. It can also be shown that the Gaussian kernel controls weak convergence on $\mathcal{X} = \mathbb{R}^d$; this can be deduced from e.g. Theorem 7 of Simon-Gabriel et al. [2020] and the general results in Sriperumbudur et al. [2011].

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x) \mathrm{d}P(x), \int k(\cdot, y) \mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x) \mathrm{d}Q(y) = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y) - 2 \iint k(x, y) \mathrm{d}P(x) \mathrm{d}Q(y) + \iint k(x, y) \mathrm{d}Q(x) \mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Maximum Mean Discrepancy

MMD is a convenient measure of sample quality because it can be computed:

$$\begin{aligned} D_k(P,Q)^2 &= \|\mu_P - \mu_Q\|^2_{\mathcal{H}(k)} \\ &= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{H}(k)} \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{H}(k)} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{H}(k)}. \end{aligned}$$

Considering for example the term $\langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)}$, we have

$$\begin{aligned} \langle \mu_P, \mu_Q \rangle_{\mathcal{H}(k)} &= \left\langle \int k(\cdot, x)\mathrm{d}P(x), \int k(\cdot, y)\mathrm{d}Q(y) \right\rangle_{\mathcal{H}(k)} \\ &= \iint \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}(k)} \mathrm{d}P(x)\mathrm{d}Q(y) = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y). \end{aligned}$$

Here we have used the reproducing property, as well as using Lemma 6 to justify the exchanges of integral and inner product. Proceeding similarly with all three terms results in the expression

$$D_k(P,Q)^2 = \iint k(x,y)\mathrm{d}P(x)\mathrm{d}P(y) - 2\iint k(x,y)\mathrm{d}P(x)\mathrm{d}Q(y) + \iint k(x,y)\mathrm{d}Q(x)\mathrm{d}Q(y).$$

How to exploit MMD for sampling?

# Sampling with Kernels

# Quantisation with Monte Carlo

The goal of *quantisation* is to find $Q$ of the form $Q_n = \frac{1}{n}\sum_{i=1}^{n} \delta(\boldsymbol{x}_i)$ such that $Q_n \approx P$.

As a simple baseline method for quantisation we consider Monte Carlo:

Figure: *Monte Carlo:* Independent samples (blue circles) from a "horseshoe" distribution $P$ (grey contours).

# Quantisation with Monte Carlo

The goal of *quantisation* is to find $Q$ of the form $Q_n = \frac{1}{n}\sum_{i=1}^{n} \delta(\boldsymbol{x}_i)$ such that $Q_n \approx P$.

As a simple baseline method for quantisation we consider Monte Carlo:

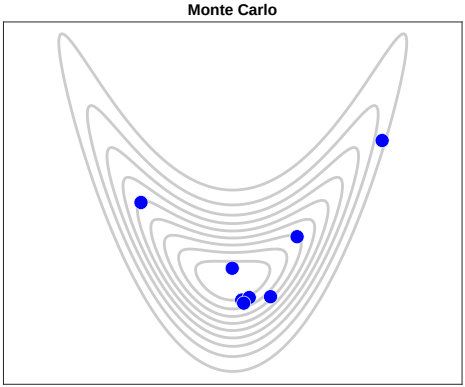



Figure: *Monte Carlo:* Independent samples (blue circles) from a "horseshoe" distribution $P$ (grey contours).

# Quantisation with Monte Carlo

## Proposition 1 (MMD of Monte Carlo)

*Let $x_1, \ldots, x_n \sim P$ be independent. Assume that $C := \int k(x, x) \mathrm{d}P(x) < \infty$. Then*

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C}{n}.$$

Proof.

From the closed form of MMD, with $Q = Q_n = \frac{1}{n}\sum_{i=1}^n \delta(x_i)$, we obtain that

$$D_k(P, Q_n)^2 = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y) - \frac{2}{n}\sum_{i=1}^n \int k(x, x_i) \mathrm{d}P(x) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j)$$

Taking expectations of both sides gives

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] = \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j)\right] - \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y)$$

# Quantisation with Monte Carlo

## Proposition 1 (MMD of Monte Carlo)

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim P$ be independent. Assume that $C := \int k(\boldsymbol{x}, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) < \infty$. Then*

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C}{n}.$$

## Proof.

From the closed form of MMD, with $Q = Q_n = \frac{1}{n}\sum_{i=1}^n \delta(\boldsymbol{x}_i)$, we obtain that

$$D_k(P, Q_n)^2 = \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}) - \frac{2}{n}\sum_{i=1}^n \int k(\boldsymbol{x}, \boldsymbol{x}_i) \mathrm{d}P(\boldsymbol{x}) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Taking expectations of both sides gives

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] = \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right] - \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y})$$

# Quantisation with Monte Carlo

## Proposition 1 (MMD of Monte Carlo)

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim P$ be independent. Assume that $C := \int k(\boldsymbol{x}, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) < \infty$. Then*

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C}{n}.$$

## Proof.

From the closed form of MMD, with $Q = Q_n = \frac{1}{n}\sum_{i=1}^n \delta(\boldsymbol{x}_i)$, we obtain that

$$D_k(P, Q_n)^2 = \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}) - \frac{2}{n}\sum_{i=1}^n \int k(\boldsymbol{x}, \boldsymbol{x}_i) \mathrm{d}P(\boldsymbol{x}) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Taking expectations of both sides gives

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] = \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right] - \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y})$$

# Quantisation with Monte Carlo

## Proposition 1 (MMD of Monte Carlo)

*Let $x_1, \ldots, x_n \sim P$ be independent. Assume that $C := \int k(x, x) \mathrm{d}P(x) < \infty$. Then*

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C}{n}.$$

## Proof.

From the closed form of MMD, with $Q = Q_n = \frac{1}{n}\sum_{i=1}^n \delta(x_i)$, we obtain that

$$D_k(P, Q_n)^2 = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y) - \frac{2}{n}\sum_{i=1}^n \int k(x, x_i) \mathrm{d}P(x) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j)$$

Taking expectations of both sides gives

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] = \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^n k(x_i, x_i) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j \neq i} k(x_i, x_j)\right] - \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y)$$

# Quantisation with Monte Carlo

## Proposition 1 (MMD of Monte Carlo)

*Let $x_1, \ldots, x_n \sim P$ be independent. Assume that $C := \int k(x, x) \mathrm{d}P(x) < \infty$. Then*

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C}{n}.$$

## Proof.

From the closed form of MMD, with $Q = Q_n = \frac{1}{n}\sum_{i=1}^n \delta(x_i)$, we obtain that

$$D_k(P, Q_n)^2 = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y) - \frac{2}{n}\sum_{i=1}^n \int k(x, x_i) \mathrm{d}P(x) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j)$$

Taking expectations of both sides gives

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] = \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^n k(x_i, x_i)\right] - \underbrace{\frac{1}{n}\iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y)}_{\geq 0}$$

# Quantisation with Monte Carlo

## Proposition 1 (MMD of Monte Carlo)

*Let $x_1, \ldots, x_n \sim P$ be independent. Assume that $C := \int k(x, x) \mathrm{d}P(x) < \infty$. Then*

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C}{n}.$$

## Proof.

From the closed form of MMD, with $Q = Q_n = \frac{1}{n}\sum_{i=1}^n \delta(x_i)$, we obtain that

$$D_k(P, Q_n)^2 = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y) - \frac{2}{n}\sum_{i=1}^n \int k(x, x_i) \mathrm{d}P(x) + \frac{1}{n^2}\sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j)$$

Taking expectations of both sides gives

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{1}{n}\int k(x, x) \mathrm{d}P(x)$$

as claimed.

# Optimal Quantisation

Monte Carlo sampling provides a consistent but potentially far from optimal quantisation of $P$.

Note that the convergence *rate* in Proposition 1 does not depend on the kernel $k$, which highlights the inefficiency of the Monte Carlo method. [But the rate is dimension-independent.]

The goal of *optimal* quantisation is to approximate $P$ as well as possible, with a fixed number of states $x_1, \ldots, x_n$.

In what follows we are going to introduce some algorithms that are **often inferior to QMC**, but have the advantage of being **applicable to general Bayesian inference tasks** with minor modification, as we will see in the sequel.

Algorithm 1 (Optimise Everything)

*Approximate a solution to*

$$\min_{(x_1,\ldots,x_n)\in\mathcal{X}\times\cdots\times\mathcal{X}} D_k\left(P, \frac{1}{n}\sum_{i=1}^{n}\delta(x_i)\right)$$

*using a numerical global optimisation method.*

# Optimal Quantisation

Monte Carlo sampling provides a consistent but potentially far from optimal quantisation of $P$.

Note that the convergence *rate* in Proposition 1 does not depend on the kernel $k$, which highlights the inefficiency of the Monte Carlo method. [But the rate is dimension-independent.]

The goal of *optimal* quantisation is to approximate $P$ as well as possible, with a fixed number of states $x_1, \ldots, x_n$.

In what follows we are going to introduce some algorithms that are **often inferior to QMC**, but have the advantage of being **applicable to general Bayesian inference tasks** with minor modification, as we will see in the sequel.

Algorithm 1 (Optimise Everything)

*Approximate a solution to*

$$\min_{(x_1,\ldots,x_n)\in\mathcal{X}\times\cdots\times\mathcal{X}} D_k\left(P, \frac{1}{n}\sum_{i=1}^{n}\delta(x_i)\right)$$

*using a numerical global optimisation method.*

# Optimal Quantisation

Monte Carlo sampling provides a consistent but potentially far from optimal quantisation of $P$.

Note that the convergence *rate* in Proposition 1 does not depend on the kernel $k$, which highlights the inefficiency of the Monte Carlo method. [But the rate is dimension-independent.]

The goal of *optimal* quantisation is to approximate $P$ as well as possible, with a fixed number of states $x_1, \ldots, x_n$.

In what follows we are going to introduce some algorithms that are **often inferior to QMC**, but have the advantage of being **applicable to general Bayesian inference tasks** with minor modification, as we will see in the sequel.

**Algorithm 1 (Optimise Everything)**

*Approximate a solution to*

$$\min_{(x_1,\ldots,x_n)\in\mathcal{X}\times\cdots\times\mathcal{X}} D_k\left(P, \frac{1}{n}\sum_{i=1}^{n}\delta(x_i)\right)$$

*using a numerical global optimisation method.*

# Optimal Quantisation

Monte Carlo sampling provides a consistent but potentially far from optimal quantisation of $P$.

Note that the convergence *rate* in Proposition 1 does not depend on the kernel $k$, which highlights the inefficiency of the Monte Carlo method. [But the rate is dimension-independent.]

The goal of *optimal* quantisation is to approximate $P$ as well as possible, with a fixed number of states $x_1, \ldots, x_n$.

In what follows we are going to introduce some algorithms that are **often inferior to QMC**, but have the advantage of being **applicable to general Bayesian inference tasks** with minor modification, as we will see in the sequel.

**Algorithm 1 (Optimise Everything)**

*Approximate a solution to*

$$\min_{(x_1,\ldots,x_n)\in\mathcal{X}\times\cdots\times\mathcal{X}} D_k\left(P, \frac{1}{n}\sum_{i=1}^{n}\delta(x_i)\right)$$

*using a numerical global optimisation method.*

# Optimal Quantisation

Monte Carlo sampling provides a consistent but potentially far from optimal quantisation of $P$.

Note that the convergence *rate* in Proposition 1 does not depend on the kernel $k$, which highlights the inefficiency of the Monte Carlo method. [But the rate is dimension-independent.]

The goal of *optimal* quantisation is to approximate $P$ as well as possible, with a fixed number of states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

In what follows we are going to introduce some algorithms that are **often inferior to QMC**, but have the advantage of being **applicable to general Bayesian inference tasks** with minor modification, as we will see in the sequel.

Algorithm 1 (Optimise Everything)

*Approximate a solution to*

$$\min_{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\in\mathcal{X}\times\cdots\times\mathcal{X}} D_k\left(P, \frac{1}{n}\sum_{i=1}^{n}\delta(\boldsymbol{x}_i)\right)$$

*using a numerical global optimisation method.*

# Optimal Quantisation

Monte Carlo sampling provides a consistent but potentially far from optimal quantisation of $P$.

Note that the convergence *rate* in Proposition 1 does not depend on the kernel $k$, which highlights the inefficiency of the Monte Carlo method. [But the rate is dimension-independent.]

The goal of *optimal* quantisation is to approximate $P$ as well as possible, with a fixed number of states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

In what follows we are going to introduce some algorithms that are **often inferior to QMC**, but have the advantage of being **applicable to general Bayesian inference tasks** with minor modification, as we will see in the sequel.

## Algorithm 1 (Optimise Everything)

*Approximate a solution to*

$$\min_{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\in\mathcal{X}\times\cdots\times\mathcal{X}} D_k\left(P, \frac{1}{n}\sum_{i=1}^{n}\delta(\boldsymbol{x}_i)\right)$$

*using a numerical global optimisation method.*

# Optimal Quantisation

Figure: *Optimal quantisation:* Global (continuous) minimisation of MMD to select states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. Example due to Gräf et al. [2012].

This example is the *stippling* procedure discussed in Gräf et al. [2012], implemented using a bandlimited kernel and the nonlinear CG method. See also *minimum energy designs* and MMD *gradient flow* [Arbel et al., 2019]. Requires a good initialisation to work well.

Also requires selecting $n$ in advance - a *non-extensible* sequence of approximations to $P$.

# Optimal Quantisation

Figure: *Optimal quantisation:* Global (continuous) minimisation of MMD to select states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. Example due to Gräf et al. [2012].

This example is the *stippling* procedure discussed in Gräf et al. [2012], implemented using a bandlimited kernel and the nonlinear CG method. See also *minimum energy designs* and MMD *gradient flow* [Arbel et al., 2019]. Requires a good initialisation to work well.

Also requires selecting $n$ in advance - a *non-extensible* sequence of approximations to $P$.

# Optimal Quantisation

Figure: *Optimal quantisation:* Global (continuous) minimisation of MMD to select states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. Example due to Gräf et al. [2012].

This example is the *stippling* procedure discussed in Gräf et al. [2012], implemented using a bandlimited kernel and the nonlinear CG method. See also *minimum energy designs* and MMD *gradient flow* [Arbel et al., 2019]. Requires a good initialisation to work well.

Also requires selecting $n$ in advance - a *non-extensible* sequence of approximations to $P$.

# Optimal Quantisation

To arrive at an extensible approximation, consider a greedy sequential algorithm:

**Algorithm 2 (Continuous Greedy Optimisation)**

*Approximate a solution to*

$$x_n \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \; D_k \left( P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1} \delta(x_i) \right), \qquad n = 1, 2, \ldots$$

*using a numerical global optimisation method.*

# Optimal Quantisation

To arrive at an extensible approximation, consider a greedy sequential algorithm:

## Algorithm 2 (Continuous Greedy Optimisation)

*Approximate a solution to*

$$\mathbf{x}_n \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \; D_k \left( P, \frac{1}{n}\delta(\mathbf{x}) + \frac{1}{n}\sum_{i=1}^{n-1} \delta(\mathbf{x}_i) \right), \qquad n = 1, 2, \ldots$$

*using a numerical global optimisation method.*

# Optimal Quantisation

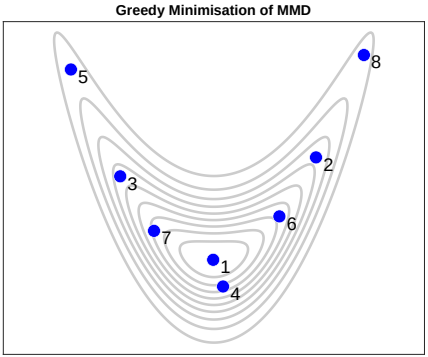



Figure: *Optimal quantisation:* Sequential (greedy) minimisation of MMD to select $n = 8$ states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. The numbers indicate the order in which the states $\boldsymbol{x}_i$ were selected.

However, the computation required to select the location $x_n$ becomes increasingly difficult as $n$ is increased. A computationally simpler algorithm is proposed next.

# Optimal Quantisation

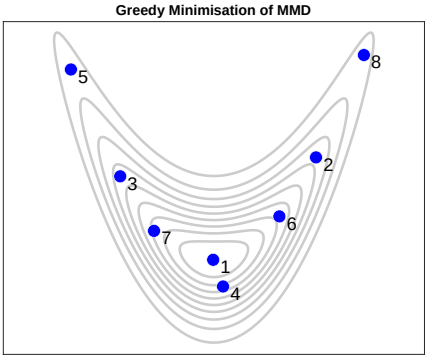



Figure: *Optimal quantisation:* Sequential (greedy) minimisation of MMD to select $n = 8$ states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. The numbers indicate the order in which the states $\boldsymbol{x}_i$ were selected.

However, the computation required to select the location $\boldsymbol{x}_n$ becomes increasingly difficult as $n$ is increased. A computationally simpler algorithm is proposed next.

# Optimal Quantisation

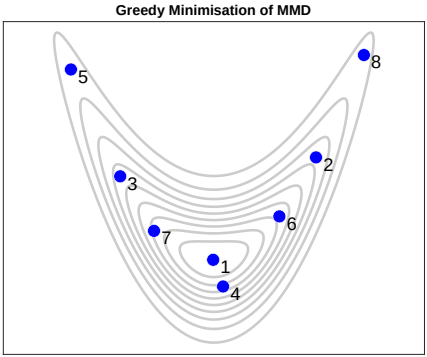



Figure: *Optimal quantisation:* Sequential (greedy) minimisation of MMD to select $n = 8$ states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. The numbers indicate the order in which the states $\boldsymbol{x}_i$ were selected.

However, the computation required to select the location $\boldsymbol{x}_n$ becomes increasingly difficult as $n$ is increased. A computationally simpler algorithm is proposed next.

# Optimal Quantisation

## Algorithm 3 (Discrete Greedy Optimisation)

*Let $(y_i)_{i\in\mathbb{N}}$ be a sample path from a Markov chain that is $P$-invariant. Then set*

$$x_n \in \operatorname*{arg\,min}_{x\in\{y_1,\ldots,y_N\}} D_k\left(P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1}\delta(x_i)\right), \qquad n = 1, 2, \ldots.$$

- Includes the case where the $y_i$ are independent samples from $P$. [Pre-empts application to the motivating Bayesian context.]
- For a sample path of length $N \gg 1$, the resulting sequence approximates that of continuous greedy optimisation (Algorithm 2).
- Under assumptions, that include sufficiently rapid mixing of the Markov chain,

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C_1}{N} + \frac{C_2(1+\log(N))(1+\log(n))}{n}.$$

  See Theorem 4 in Teymur et al. [2021].
- Same rate as Monte Carlo up to log terms; recall Proposition 1. [Possibly a theoretical gap here.]

# Optimal Quantisation

## Algorithm 3 (Discrete Greedy Optimisation)

*Let $(y_i)_{i\in\mathbb{N}}$ be a sample path from a Markov chain that is $P$-invariant. Then set*

$$x_n \in \operatorname*{arg\,min}_{x\in\{y_1,\ldots,y_N\}} D_k\left(P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1}\delta(x_i)\right), \qquad n = 1, 2, \ldots.$$

- Includes the case where the $y_i$ are independent samples from $P$. [Pre-empts application to the motivating Bayesian context.]
- For a sample path of length $N \gg 1$, the resulting sequence approximates that of continuous greedy optimisation (Algorithm 2).
- Under assumptions, that include sufficiently rapid mixing of the Markov chain,

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C_1}{N} + \frac{C_2(1+\log(N))(1+\log(n))}{n}.$$

  See Theorem 4 in Teymur et al. [2021].
- Same rate as Monte Carlo up to log terms; recall Proposition 1. [Possibly a theoretical gap here.]

# Optimal Quantisation

## Algorithm 3 (Discrete Greedy Optimisation)

*Let $(y_i)_{i\in\mathbb{N}}$ be a sample path from a Markov chain that is $P$-invariant. Then set*

$$x_n \in \operatorname*{arg\,min}_{x\in\{y_1,\ldots,y_N\}} D_k\left(P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1}\delta(x_i)\right), \qquad n = 1, 2, \ldots.$$

- Includes the case where the $y_i$ are independent samples from $P$. [Pre-empts application to the motivating Bayesian context.]
- For a sample path of length $N \gg 1$, the resulting sequence approximates that of continuous greedy optimisation (Algorithm 2).
- Under assumptions, that include sufficiently rapid mixing of the Markov chain,

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C_1}{N} + \frac{C_2(1+\log(N))(1+\log(n))}{n}.$$

  See Theorem 4 in Teymur et al. [2021].
- Same rate as Monte Carlo up to log terms; recall Proposition 1. [Possibly a theoretical gap here.]

# Optimal Quantisation

## Algorithm 3 (Discrete Greedy Optimisation)

*Let $(y_i)_{i\in\mathbb{N}}$ be a sample path from a Markov chain that is $P$-invariant. Then set*

$$x_n \in \operatorname*{arg\,min}_{x\in\{y_1,\ldots,y_N\}} D_k\left(P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1}\delta(x_i)\right), \qquad n = 1, 2, \ldots.$$

- Includes the case where the $y_i$ are independent samples from $P$. [Pre-empts application to the motivating Bayesian context.]
- For a sample path of length $N \gg 1$, the resulting sequence approximates that of continuous greedy optimisation (Algorithm 2).
- Under assumptions, that include sufficiently rapid mixing of the Markov chain,

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C_1}{N} + \frac{C_2(1+\log(N))(1+\log(n))}{n}.$$

  See Theorem 4 in Teymur et al. [2021].
- Same rate as Monte Carlo up to log terms; recall Proposition 1. [Possibly a theoretical gap here.]

# Optimal Quantisation

## Algorithm 3 (Discrete Greedy Optimisation)

*Let $(y_i)_{i\in\mathbb{N}}$ be a sample path from a Markov chain that is $P$-invariant. Then set*

$$x_n \in \operatorname*{arg\,min}_{x\in\{y_1,\ldots,y_N\}} D_k\left(P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1}\delta(x_i)\right), \qquad n = 1, 2, \ldots.$$

- Includes the case where the $y_i$ are independent samples from $P$. [Pre-empts application to the motivating Bayesian context.]
- For a sample path of length $N \gg 1$, the resulting sequence approximates that of continuous greedy optimisation (Algorithm 2).
- Under assumptions, that include sufficiently rapid mixing of the Markov chain,

$$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C_1}{N} + \frac{C_2(1+\log(N))(1+\log(n))}{n}.$$

  See Theorem 4 in Teymur et al. [2021].
- Same rate as Monte Carlo up to log terms; recall Proposition 1. [Possibly a theoretical gap here.]

# Optimal Quantisation

## Algorithm 3 (Discrete Greedy Optimisation)

*Let $(y_i)_{i\in\mathbb{N}}$ be a sample path from a Markov chain that is $P$-invariant. Then set*

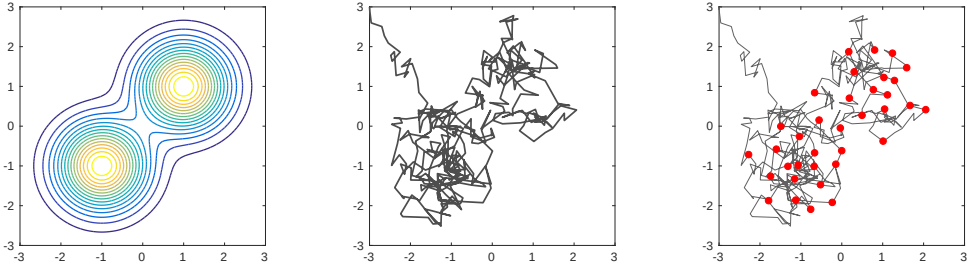
$$x_n \in \operatorname*{arg\,min}_{x\in\{y_1,\ldots,y_N\}} D_k\left(P, \frac{1}{n}\delta(x) + \frac{1}{n}\sum_{i=1}^{n-1}\delta(x_i)\right), \qquad n = 1, 2, \ldots.$$

- Includes the case where the $y_i$ are independent samples from $P$. [Pre-empts application to the motivating Bayesian context.]
- For a sample path of length $N \gg 1$, the resulting sequence approximates that of continuous greedy optimisation (Algorithm 2).
- Under assumptions, that include sufficiently rapid mixing of the Markov chain,

  $$\mathbb{E}\left[D_k(P, Q_n)^2\right] \leq \frac{C_1}{N} + \frac{C_2(1+\log(N))(1+\log(n))}{n}.$$

  See Theorem 4 in Teymur et al. [2021].
- Same rate as Monte Carlo up to log terms; recall Proposition 1. [Possibly a theoretical gap here.]

# Optimal Quantisation

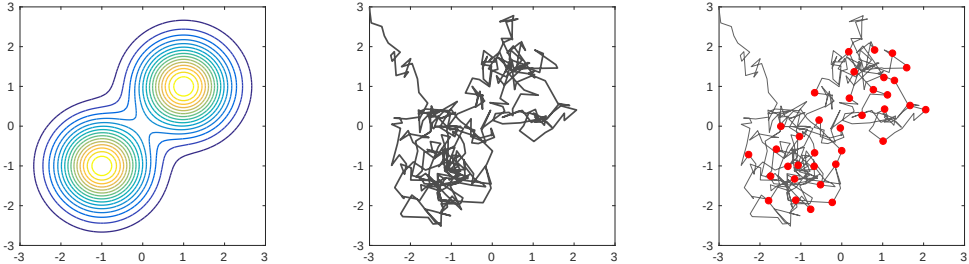



$P$ | MCMC output $(y_i)_{i=1}^N$ | Representative Subset $(x_i)_{i=1}^n$

Figure: *Optimal quantisation:* Discrete greedy optimisation of MMD to select $n = 35$ states $\{x_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$.

Q: What can be done to improve the disappointing theoretical convergence rate?

# Optimal Quantisation




$P$ | MCMC output $(y_i)_{i=1}^N$ | Representative Subset $(x_i)_{i=1}^n$

Figure: *Optimal quantisation:* Discrete greedy optimisation of MMD to select $n = 35$ states $\{x_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$.

Q: What can be done to improve the disappointing theoretical convergence rate?

# Optimal Approximation

Let's return to the case of *weighted* point sets; i.e. approximations of the form $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ for some weights $w_1, \ldots, w_n \in \mathbb{R}$.

**Lemma 13**

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\boldsymbol{w} \in \mathbb{R}^n}{\arg\min}\, D_k\left(P, \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)\right)$$

*are the solution of the linear system*

$$K\boldsymbol{w} = \boldsymbol{z} \tag{5}$$

*where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $z_i = \mu_P(\boldsymbol{x}_i)$.*

**Proof.**

The MMD between $P$ and $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ can be expressed as

$$D_k(P, Q_n)^2 = \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}) - 2 \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}Q(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}Q(\boldsymbol{x}) \mathrm{d}Q(\boldsymbol{y})$$

# Optimal Approximation

Let's return to the case of *weighted* point sets; i.e. approximations of the form $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ for some weights $w_1, \ldots, w_n \in \mathbb{R}$.

**Lemma 13**

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\boldsymbol{w} \in \mathbb{R}^n}{\arg\min}\, D_k \left( P, \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)$$

*are the solution of the linear system*

$$\boldsymbol{K}\boldsymbol{w} = \boldsymbol{z} \tag{5}$$

*where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $z_i = \mu_P(\boldsymbol{x}_i)$.*

Proof.

The MMD between $P$ and $Q_n = \sum_{i=1}^n w_i \delta(x_i)$ can be expressed as

$$D_k(P, Q_n)^2 = \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y) - 2 \iint k(x, y) \mathrm{d}P(x) \mathrm{d}Q(y) + \iint k(x, y) \mathrm{d}Q(x) \mathrm{d}Q(y)$$

# Optimal Approximation

Let's return to the case of *weighted* point sets; i.e. approximations of the form $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ for some weights $w_1, \ldots, w_n \in \mathbb{R}$.

## Lemma 13

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\boldsymbol{w} \in \mathbb{R}^n}{\arg\min}\, D_k \left( P, \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)$$

*are the solution of the linear system*

$$\boldsymbol{K}\boldsymbol{w} = \boldsymbol{z} \tag{5}$$

*where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $z_i = \mu_P(\boldsymbol{x}_i)$.*

## Proof.

The MMD between $P$ and $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ can be expressed as

$$D_k(P, Q_n)^2 = \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}) - 2 \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}Q(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}Q(\boldsymbol{x}) \mathrm{d}Q(\boldsymbol{y})$$

# Optimal Approximation

Let's return to the case of *weighted* point sets; i.e. approximations of the form $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ for some weights $w_1, \ldots, w_n \in \mathbb{R}$.

**Lemma 13**

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\boldsymbol{w} \in \mathbb{R}^n}{\arg\min}\, D_k\left(P, \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)\right)$$

*are the solution of the linear system*

$$\boldsymbol{K}\boldsymbol{w} = \boldsymbol{z} \tag{5}$$

*where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $z_i = \mu_P(\boldsymbol{x}_i)$.*

**Proof.**

The MMD between $P$ and $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ can be expressed as

$$D_k(P, Q_n)^2 = C - 2\boldsymbol{z}^\top \boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{K}\boldsymbol{w}$$

where $C = \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y})$ is independent of $\boldsymbol{w}$. This is a non-degenerate quadratic form in $\boldsymbol{w}$ (since $\boldsymbol{K}$ is a positive definite matrix), from which the result is easily verified.

# Optimal Approximation

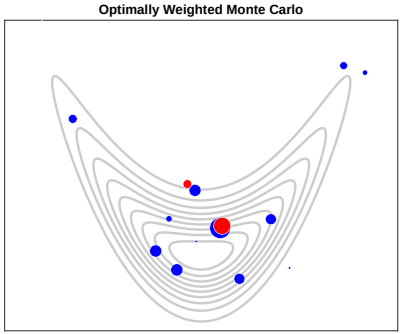



Figure: *Optimally weighted Monte Carlo samples:* The weights $w_1, \ldots, w_n$ are obtained by minimising MMD in the manner of Lemma 13. Blue indicates states $\boldsymbol{x}_i$ with positive weights $w_i > 0$, while red indicates negative weights $w_i < 0$. The size of the circles is proportional to $|w_i|$.

The linear system in (5), defining the optimal weights, can be numerically ill-conditioned and the general computational overhead is $O(n^3)$. However, several "tricks" are available, such as Karvonen and Särkkä [2018], Karvonen et al. [2019].

# Optimal Approximation

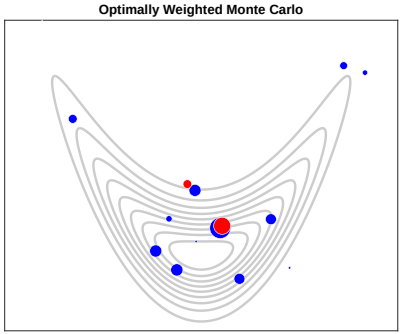
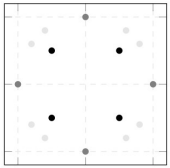



Figure: *Optimally weighted Monte Carlo samples:* The weights $w_1, \ldots, w_n$ are obtained by minimising MMD in the manner of Lemma 13. Blue indicates states $\boldsymbol{x}_i$ with positive weights $w_i > 0$, while red indicates negative weights $w_i < 0$. The size of the circles is proportional to $|w_i|$.

The linear system in (5), defining the optimal weights, can be numerically ill-conditioned and the general computational overhead is $O(n^3)$. However, several "tricks" are available, such as Karvonen and Särkkä [2018], Karvonen et al. [2019].



# Optimal Approximation

Do optimal weights improve convergence rates? It depends on the kernel!

**Notation:** The *multi-index* notation

$$\partial^{\alpha} f : \boldsymbol{x} \mapsto \frac{\partial^{\alpha} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\alpha}} = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(\boldsymbol{x}), \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^d$$

will be used, and we let $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$.

Definition 14

For $s > d/2$ and (sufficiently regular) $\mathcal{X} \subset \mathbb{R}^d$, the (order $s$) Sobolev space $H^s(\mathcal{X})$ is defined to be the set of functions $f : \mathcal{X} \to \mathbb{R}$ whose mixed partial derivatives $\partial^{\alpha} f$, $|\alpha| \leq s$, exist in $L^2(\mathcal{X})$. This becomes a Hilbert space with inner product

$$\langle f, g \rangle_{H^s(\mathcal{X})} = \sum_{|\alpha| \leq s} \int \frac{\partial^{\alpha} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\alpha}} \frac{\partial^{\alpha} g(\boldsymbol{x})}{\partial \boldsymbol{x}^{\alpha}} \mathrm{d}\boldsymbol{x}.$$

A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *Sobolev kernel* if there exists $0 < c_1 < c_2 < \infty$ such that, for all $f \in \mathcal{H}(k)$, we have $c_1 \|f\|_{H^s(\mathcal{X})} \leq \|f\|_{\mathcal{H}(k)} \leq c_2 \|f\|_{H^s(\mathcal{X})}$.

# Optimal Approximation

Do optimal weights improve convergence rates? It depends on the kernel!

**Notation:** The *multi-index* notation

$$\partial^{\alpha} f : \boldsymbol{x} \mapsto \frac{\partial^{\alpha} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\alpha}} = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(\boldsymbol{x}), \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^d$$

will be used, and we let $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$.

Definition 14

For $s > d/2$ and (sufficiently regular) $\mathcal{X} \subset \mathbb{R}^d$, the (order $s$) Sobolev space $H^s(\mathcal{X})$ is defined to be the set of functions $f : \mathcal{X} \to \mathbb{R}$ whose mixed partial derivatives $\partial^{\alpha} f$, $|\alpha| \leq s$, exist in $L^2(\mathcal{X})$. This becomes a Hilbert space with inner product

$$\langle f, g \rangle_{H^s(\mathcal{X})} = \sum_{|\alpha| \leq s} \int \frac{\partial^{\alpha} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\alpha}} \frac{\partial^{\alpha} g(\boldsymbol{x})}{\partial \boldsymbol{x}^{\alpha}} \mathrm{d}\boldsymbol{x}.$$

A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *Sobolev kernel* if there exists $0 < c_1 < c_2 < \infty$ such that, for all $f \in \mathcal{H}(k)$, we have $c_1 \|f\|_{H^s(\mathcal{X})} \leq \|f\|_{\mathcal{H}(k)} \leq c_2 \|f\|_{H^s(\mathcal{X})}$.

# Optimal Approximation

Do optimal weights improve convergence rates? It depends on the kernel!

**Notation:** The *multi-index* notation

$$\partial^{\boldsymbol{\alpha}} f : \boldsymbol{x} \mapsto \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(\boldsymbol{x}), \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^d$$

will be used, and we let $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$.

Definition 14

For $s > d/2$ and (sufficiently regular) $\mathcal{X} \subset \mathbb{R}^d$, the (order $s$) Sobolev space $H^s(\mathcal{X})$ is defined to be the set of functions $f : \mathcal{X} \to \mathbb{R}$ whose mixed partial derivatives $\partial^{\alpha} f$, $|\alpha| \leq s$, exist in $L^2(\mathcal{X})$. This becomes a Hilbert space with inner product

$$\langle f, g \rangle_{H^s(\mathcal{X})} = \sum_{|\alpha| \leq s} \int \frac{\partial^{\alpha} f(x)}{\partial x^{\alpha}} \frac{\partial^{\alpha} g(x)}{\partial x^{\alpha}} \mathrm{d}x.$$

A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *Sobolev kernel* if there exists $0 < c_1 < c_2 < \infty$ such that, for all $f \in \mathcal{H}(k)$, we have $c_1 \|f\|_{H^s(\mathcal{X})} \leq \|f\|_{\mathcal{H}(k)} \leq c_2 \|f\|_{H^s(\mathcal{X})}$.

# Optimal Approximation

Do optimal weights improve convergence rates? It depends on the kernel!

**Notation:** The *multi-index* notation

$$\partial^{\boldsymbol{\alpha}} f : \boldsymbol{x} \mapsto \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(\boldsymbol{x}), \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^d$$

will be used, and we let $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$.

## Definition 14

For $s > d/2$ and (sufficiently regular) $\mathcal{X} \subset \mathbb{R}^d$, the (order $s$) Sobolev space $H^s(\mathcal{X})$ is defined to be the set of functions $f : \mathcal{X} \to \mathbb{R}$ whose mixed partial derivatives $\partial^{\boldsymbol{\alpha}} f$, $|\boldsymbol{\alpha}| \leq s$, exist in $L^2(\mathcal{X})$. This becomes a Hilbert space with inner product

$$\langle f, g \rangle_{H^s(\mathcal{X})} = \sum_{|\boldsymbol{\alpha}| \leq s} \int \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} \frac{\partial^{\boldsymbol{\alpha}} g(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} \mathrm{d}\boldsymbol{x}.$$

A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *Sobolev kernel* if there exists $0 < c_1 < c_2 < \infty$ such that, for all $f \in \mathcal{H}(k)$, we have $c_1 \|f\|_{H^s(\mathcal{X})} \leq \|f\|_{\mathcal{H}(k)} \leq c_2 \|f\|_{H^s(\mathcal{X})}$.

# Optimal Approximation

Do optimal weights improve convergence rates? It depends on the kernel!

**Notation:** The *multi-index* notation

$$\partial^{\boldsymbol{\alpha}} f : \boldsymbol{x} \mapsto \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(\boldsymbol{x}), \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^d$$

will be used, and we let $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$.

## Definition 14

For $s > d/2$ and (sufficiently regular) $\mathcal{X} \subset \mathbb{R}^d$, the (order $s$) Sobolev space $H^s(\mathcal{X})$ is defined to be the set of functions $f : \mathcal{X} \to \mathbb{R}$ whose mixed partial derivatives $\partial^{\boldsymbol{\alpha}} f$, $|\boldsymbol{\alpha}| \leq s$, exist in $L^2(\mathcal{X})$. This becomes a Hilbert space with inner product

$$\langle f, g \rangle_{H^s(\mathcal{X})} = \sum_{|\boldsymbol{\alpha}| \leq s} \int \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} \frac{\partial^{\boldsymbol{\alpha}} g(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} \mathrm{d}\boldsymbol{x}.$$

A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *Sobolev kernel* if there exists $0 < c_1 < c_2 < \infty$ such that, for all $f \in \mathcal{H}(k)$, we have $c_1 \|f\|_{H^s(\mathcal{X})} \leq \|f\|_{\mathcal{H}(k)} \leq c_2 \|f\|_{H^s(\mathcal{X})}$.

# Optimal Approximation

Do optimal weights improve convergence rates? It depends on the kernel!

**Notation:** The *multi-index* notation

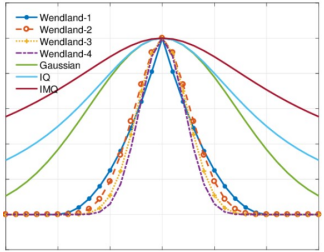
$$\partial^{\boldsymbol{\alpha}} f : \boldsymbol{x} \mapsto \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(\boldsymbol{x}), \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^d$$

will be used, and we let $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$.

## Example 14

Let $z_+^m$ denote $\max(0, z)^m$ in shorthand. Then examples of Sobolev kernels on (sufficiently regular) $\mathcal{X} \subseteq \mathbb{R}$ include the following, due to Wendland [1998]:

| $k(x, y) \quad (r = \lvert x - y \rvert,\ x, y \in \mathbb{R})$ | order |
|---|---|
| $(1 - r)_+$ | $s = 1$ |
| $(1 - r)_+^3 (3r + 1)$ | $s = 2$ |
| $(1 - r)_+^5 (8r^2 + 5r + 1)$ | $s = 3$ |




These kernels are convenient for numerical reasons, due to their compact support, which renders $K$ a *sparse* matrix.

# Optimal Approximation

Optimal weights can accelerate convergence:

**Theorem 15 (e.g. Ehler et al. [2019])**

*Let $x_1, \ldots, x_n \sim P$ be independent and let $w = w(x_1, \ldots, x_n)$ denote optimal weights in the sense of Lemma 13. Let $k(x, y)$ be an (order $s$) Sobolev kernel. Then, under regularity conditions on the domain $\mathcal{X}$, which is of dimension $d$, and the distribution $P$, there exists a constant $0 < C < \infty$ such that*

$$\mathbb{E}\left[D_k\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)\right] \leq C\left(\frac{\log(n)}{n}\right)^{s/d}.$$

For $s = d/2$ we recover the same rate as Proposition 1 for un-weighted Monte Carlo (up to log factors), while for $s > d/2$ we obtain faster convergence in MMD.

So surely it is a good idea to employ optimal weights? Not necessarily - the gain in rate has to out-weight the higher computational cost of $O(n^3)$. Superior error-per-cost only when $s > 3d/2$.

Numerical ill-conditioning at large $n$ also requires careful treatment. The use of points from optimal quantisation can be helpful.

Thus inferior to quasi-Monte Carlo in general, **but applicable in the Bayesian context!**

# Optimal Approximation

Optimal weights can accelerate convergence:

**Theorem 15 (e.g. Ehler et al. [2019])**

*Let $x_1, \ldots, x_n \sim P$ be independent and let $\boldsymbol{w} = \boldsymbol{w}(x_1, \ldots, x_n)$ denote optimal weights in the sense of Lemma 13. Let $k(x, y)$ be an (order $s$) Sobolev kernel. Then, under regularity conditions on the domain $\mathcal{X}$, which is of dimension $d$, and the distribution $P$, there exists a constant $0 < C < \infty$ such that*

$$\mathbb{E}\left[D_k\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)\right] \leq C\left(\frac{\log(n)}{n}\right)^{s/d}.$$

For $s = d/2$ we recover the same rate as Proposition 1 for un-weighted Monte Carlo (up to log factors), while for $s > d/2$ we obtain faster convergence in MMD.

So surely it is a good idea to employ optimal weights? Not necessarily - the gain in rate has to out-weight the higher computational cost of $O(n^3)$. Superior error-per-cost only when $s > 3d/2$.

Numerical ill-conditioning at large $n$ also requires careful treatment. The use of points from optimal quantisation can be helpful.

Thus inferior to quasi-Monte Carlo in general, **but applicable in the Bayesian context!**

# Optimal Approximation

Optimal weights can accelerate convergence:

**Theorem 15 (e.g. Ehler et al. [2019])**

*Let $x_1, \ldots, x_n \sim P$ be independent and let $\boldsymbol{w} = \boldsymbol{w}(x_1, \ldots, x_n)$ denote optimal weights in the sense of Lemma 13. Let $k(x, y)$ be an (order $s$) Sobolev kernel. Then, under regularity conditions on the domain $\mathcal{X}$, which is of dimension $d$, and the distribution $P$, there exists a constant $0 < C < \infty$ such that*

$$\mathbb{E}\left[D_k\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)\right] \leq C\left(\frac{\log(n)}{n}\right)^{s/d}.$$

For $s = d/2$ we recover the same rate as Proposition 1 for un-weighted Monte Carlo (up to log factors), while for $s > d/2$ we obtain faster convergence in MMD.

So surely it is a good idea to employ optimal weights? Not necessarily - the gain in rate has to out-weight the higher computational cost of $O(n^3)$. Superior error-per-cost only when $s > 3d/2$.

Numerical ill-conditioning at large $n$ also requires careful treatment. The use of points from optimal quantisation can be helpful.

Thus inferior to quasi-Monte Carlo in general, **but applicable in the Bayesian context!**

# Optimal Approximation

Optimal weights can accelerate convergence:

**Theorem 15 (e.g. Ehler et al. [2019])**

*Let $x_1, \ldots, x_n \sim P$ be independent and let $\boldsymbol{w} = \boldsymbol{w}(x_1, \ldots, x_n)$ denote optimal weights in the sense of Lemma 13. Let $k(x, y)$ be an (order $s$) Sobolev kernel. Then, under regularity conditions on the domain $\mathcal{X}$, which is of dimension $d$, and the distribution $P$, there exists a constant $0 < C < \infty$ such that*

$$\mathbb{E}\left[D_k\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)\right] \leq C\left(\frac{\log(n)}{n}\right)^{s/d}.$$

For $s = d/2$ we recover the same rate as Proposition 1 for un-weighted Monte Carlo (up to log factors), while for $s > d/2$ we obtain faster convergence in MMD.

So surely it is a good idea to employ optimal weights? Not necessarily - the gain in rate has to out-weight the higher computational cost of $O(n^3)$. Superior error-per-cost only when $s > 3d/2$.

Numerical ill-conditioning at large $n$ also requires careful treatment. The use of points from optimal quantisation can be helpful.

Thus inferior to quasi-Monte Carlo in general, **but applicable in the Bayesian context!**

# Optimal Approximation

Optimal weights can accelerate convergence:

**Theorem 15 (e.g. Ehler et al. [2019])**

*Let $x_1, \ldots, x_n \sim P$ be independent and let $\boldsymbol{w} = \boldsymbol{w}(x_1, \ldots, x_n)$ denote optimal weights in the sense of Lemma 13. Let $k(x, y)$ be an (order $s$) Sobolev kernel. Then, under regularity conditions on the domain $\mathcal{X}$, which is of dimension $d$, and the distribution $P$, there exists a constant $0 < C < \infty$ such that*

$$\mathbb{E}\left[D_k\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)\right] \leq C\left(\frac{\log(n)}{n}\right)^{s/d}.$$

For $s = d/2$ we recover the same rate as Proposition 1 for un-weighted Monte Carlo (up to log factors), while for $s > d/2$ we obtain faster convergence in MMD.

So surely it is a good idea to employ optimal weights? Not necessarily - the gain in rate has to out-weight the higher computational cost of $O(n^3)$. Superior error-per-cost only when $s > 3d/2$.

Numerical ill-conditioning at large $n$ also requires careful treatment. The use of points from optimal quantisation can be helpful.

Thus inferior to quasi-Monte Carlo in general, **but applicable in the Bayesian context!**

# Optimal Approximation

Optimal weights can accelerate convergence:

**Theorem 15 (e.g. Ehler et al. [2019])**

*Let $x_1, \ldots, x_n \sim P$ be independent and let $\boldsymbol{w} = \boldsymbol{w}(x_1, \ldots, x_n)$ denote optimal weights in the sense of Lemma 13. Let $k(x, y)$ be an (order $s$) Sobolev kernel. Then, under regularity conditions on the domain $\mathcal{X}$, which is of dimension $d$, and the distribution $P$, there exists a constant $0 < C < \infty$ such that*

$$\mathbb{E}\left[D_k\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)\right] \leq C\left(\frac{\log(n)}{n}\right)^{s/d}.$$

For $s = d/2$ we recover the same rate as Proposition 1 for un-weighted Monte Carlo (up to log factors), while for $s > d/2$ we obtain faster convergence in MMD.

So surely it is a good idea to employ optimal weights? Not necessarily - the gain in rate has to out-weight the higher computational cost of $O(n^3)$. Superior error-per-cost only when $s > 3d/2$.

Numerical ill-conditioning at large $n$ also requires careful treatment. The use of points from optimal quantisation can be helpful.

Thus inferior to quasi-Monte Carlo in general, **but applicable in the Bayesian context!**

# Stein Discrepancy

# Recap: Bayesian Inference and Sampling

Recall that we aim to perform optimal quantisation of a distribution $P$ that admits a PDF $p(\boldsymbol{x})$ on $\boldsymbol{x} \in \mathbb{R}^d$, such that

$$p(\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{Z},$$

where $\tilde{p}$ can be exactly evaluated but $Z$, and hence $p(\boldsymbol{x})$, cannot easily be evaluated or even approximated.

This setting is typical in applications of Bayesian inference, where we have

$$p(x) = \frac{\pi(x)\mathcal{L}(x)}{Z}$$

where $\pi(x)$ is a *prior* PDF, $\mathcal{L}(x)$ is a likelihood, and the implicitly defined normalisation constant $Z$ is the *marginal likelihood*.

Several methods have been developed in the statistics, applied probability, physics and machine learning literatures to approximate distributions $P$ with these characteristics, including *Markov chain Monte Carlo (MCMC)*, *sequential Monte Carlo (SMC)*, and *variational inference*. These techniques do not typically attempt *optimal* quantisation, since even the basic quantisation task can be difficult.

# Recap: Bayesian Inference and Sampling

Recall that we aim to perform optimal quantisation of a distribution $P$ that admits a PDF $p(\boldsymbol{x})$ on $\boldsymbol{x} \in \mathbb{R}^d$, such that

$$p(\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{Z},$$

where $\tilde{p}$ can be exactly evaluated but $Z$, and hence $p(\boldsymbol{x})$, cannot easily be evaluated or even approximated.

This setting is typical in applications of Bayesian inference, where we have

$$p(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})}{Z}$$

where $\pi(\boldsymbol{x})$ is a *prior* PDF, $\mathcal{L}(\boldsymbol{x})$ is a likelihood, and the implicitly defined normalisation constant $Z$ is the *marginal likelihood*.

Several methods have been developed in the statistics, applied probability, physics and machine learning literatures to approximate distributions $P$ with these characteristics, including *MCMC*, *SMC*, and *variational inference*. These techniques do not typically attempt *optimal* quantisation, since even the basic quantisation task can be difficult.

# Recap: Bayesian Inference and Sampling

Recall that we aim to perform optimal quantisation of a distribution $P$ that admits a PDF $p(\boldsymbol{x})$ on $\boldsymbol{x} \in \mathbb{R}^d$, such that

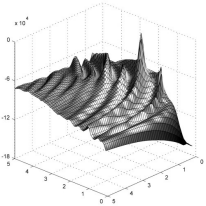
$$p(\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{Z},$$

where $\tilde{p}$ can be exactly evaluated but $Z$, and hence $p(\boldsymbol{x})$, cannot easily be evaluated or even approximated.

The integral

$$Z = \int \pi(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

is often extremely challenging to evaluate due to localised regions in which $\mathcal{L}$ takes very large values.




Several methods have been developed in the statistics, applied probability, physics and machine learning literatures to approximate distributions $P$ with these characteristics, including *MCMC*, *SMC*, and *variational inference*. These techniques do not typically attempt *optimal* quantisation, since even the basic quantisation task can be difficult.

# Recap: Bayesian Inference and Sampling

Recall that we aim to perform optimal quantisation of a distribution $P$ that admits a PDF $p(\boldsymbol{x})$ on $\boldsymbol{x} \in \mathbb{R}^d$, such that

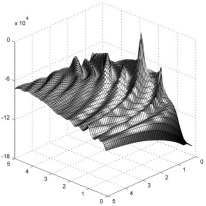
$$p(\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{Z},$$

where $\tilde{p}$ can be exactly evaluated but $Z$, and hence $p(\boldsymbol{x})$, cannot easily be evaluated or even approximated.

The integral

$$Z = \int \pi(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

is often extremely challenging to evaluate due to localised regions in which $\mathcal{L}$ takes very large values.




Several methods have been developed in the statistics, applied probability, physics and machine learning literatures to approximate distributions $P$ with these characteristics, including *MCMC*, *SMC*, and *variational inference*. These techniques do not typically attempt *optimal* quantisation, since even the basic quantisation task can be difficult.

# Stein Discrepancy

## How can our algorithms be applied?

The apparent difficulty is that we cannot compute integrals with respect to $P$, such as $\int k(\cdot, x)\mathrm{d}P(x)$, which are required for computation of MMD. A hint at a possible solution is provided by the following result:

**Lemma 16**

*Suppose $k_P : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel with $\int k_P(\cdot, x)\mathrm{d}P = 0$ for all $x \in \mathcal{X}$. Then*

$$D_{k_P}(Q) = D_{k_P}(P, Q) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d}Q \right|.$$

**Proof.**

For all $f \in \mathcal{H}(k_P)$ it holds that $\int f \mathrm{d}P = 0$, whence the result. Indeed, from the reproducing property, and using Lemma 6 with Standing Assumption 1 to interchange integral with inner product, $\int f \mathrm{d}P = \int \langle f, k(\cdot, x)\rangle_{\mathcal{H}(k_P)} \mathrm{d}P(x) = \left\langle f, \int k(\cdot, x)\mathrm{d}P(x)\right\rangle_{\mathcal{H}(k_P)} = \langle f, 0\rangle_{\mathcal{H}(k_P)} = 0.$

The important point here is that $D_{k_P}(Q)$ does not require integrals with respect to $P$ to be computed. A kernel $k_P$ with $\int k_P(\cdot, x)\mathrm{d}P = 0$ will be called a *Stein kernel* (for $P$).

# Stein Discrepancy

## How can our algorithms be applied?

The apparent difficulty is that we cannot compute integrals with respect to $P$, such as $\int k(\cdot, x)\mathrm{d}P(x)$, which are required for computation of MMD. A hint at a possible solution is provided by the following result:

**Lemma 16**

*Suppose $k_P : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel with $\int k_P(\cdot, x)\mathrm{d}P = 0$ for all $x \in \mathcal{X}$. Then*

$$D_{k_P}(Q) = D_{k_P}(P, Q) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d}Q \right|.$$

**Proof.**

For all $f \in \mathcal{H}(k_P)$ it holds that $\int f \mathrm{d}P = 0$, whence the result. Indeed, from the reproducing property, and using Lemma 6 with Standing Assumption 1 to interchange integral with inner product, $\int f \mathrm{d}P = \int \langle f, k(\cdot, x)\rangle_{\mathcal{H}(k_P)} \mathrm{d}P(x) = \left\langle f, \int k(\cdot, x)\mathrm{d}P(x)\right\rangle_{\mathcal{H}(k_P)} = \langle f, 0\rangle_{\mathcal{H}(k_P)} = 0.$

The important point here is that $D_{k_P}(Q)$ does not require integrals with respect to $P$ to be computed. A kernel $k_P$ with $\int k_P(\cdot, x)\mathrm{d}P = 0$ will be called a *Stein kernel* (for $P$).

# Stein Discrepancy

## How can our algorithms be applied?

The apparent difficulty is that we cannot compute integrals with respect to $P$, such as $\int k(\cdot, x)\mathrm{d}P(x)$, which are required for computation of MMD. A hint at a possible solution is provided by the following result:

**Lemma 16**

*Suppose $k_P : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel with $\int k_P(\cdot, x)\mathrm{d}P = 0$ for all $x \in \mathcal{X}$. Then*

$$D_{k_P}(Q) = D_{k_P}(P, Q) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d}Q \right|.$$

**Proof.**

For all $f \in \mathcal{H}(k_P)$ it holds that $\int f \mathrm{d}P = 0$, whence the result. Indeed, from the reproducing property, and using Lemma 6 with Standing Assumption 1 to interchange integral with inner product, $\int f \mathrm{d}P = \int \langle f, k(\cdot, x)\rangle_{\mathcal{H}(k_P)} \mathrm{d}P(x) = \left\langle f, \int k(\cdot, x)\mathrm{d}P(x)\right\rangle_{\mathcal{H}(k_P)} = \langle f, 0\rangle_{\mathcal{H}(k_P)} = 0.$

The important point here is that $D_{k_P}(Q)$ does not require integrals with respect to $P$ to be computed. A kernel $k_P$ with $\int k_P(\cdot, x)\mathrm{d}P = 0$ will be called a *Stein kernel* (for $P$).

# Stein Discrepancy

## How can our algorithms be applied?

The apparent difficulty is that we cannot compute integrals with respect to $P$, such as $\int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})$, which are required for computation of MMD. A hint at a possible solution is provided by the following result:

### Lemma 16

*Suppose $k_P : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel with $\int k_P(\cdot, \boldsymbol{x})\mathrm{d}P = 0$ for all $\boldsymbol{x} \in \mathcal{X}$. Then*

$$D_{k_P}(Q) = D_{k_P}(P, Q) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d}Q \right|.$$

Proof.

For all $f \in \mathcal{H}(k_P)$ it holds that $\int f \mathrm{d}P = 0$, whence the result. Indeed, from the reproducing property, and using Lemma 6 with Standing Assumption 1 to interchange integral with inner product, $\int f \mathrm{d}P = \int \langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}(k_P)} \mathrm{d}P(\boldsymbol{x}) = \left\langle f, \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x}) \right\rangle_{\mathcal{H}(k_P)} = \langle f, 0 \rangle_{\mathcal{H}(k_P)} = 0.$

The important point here is that $D_{k_P}(Q)$ does not require integrals with respect to $P$ to be computed. A kernel $k_P$ with $\int k_P(\cdot, \boldsymbol{x})\mathrm{d}P = 0$ will be called a *Stein kernel* (for $P$).

# Stein Discrepancy

## How can our algorithms be applied?

The apparent difficulty is that we cannot compute integrals with respect to $P$, such as $\int k(\cdot, x)\mathrm{d}P(x)$, which are required for computation of MMD. A hint at a possible solution is provided by the following result:

### Lemma 16

*Suppose $k_P : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel with $\int k_P(\cdot, x)\mathrm{d}P = 0$ for all $x \in \mathcal{X}$. Then*

$$D_{k_P}(Q) = D_{k_P}(P, Q) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d}Q \right|.$$

### Proof.

For all $f \in \mathcal{H}(k_P)$ it holds that $\int f \mathrm{d}P = 0$, whence the result. Indeed, from the reproducing property, and using Lemma 6 with Standing Assumption 1 to interchange integral with inner product, $\int f \mathrm{d}P = \int \langle f, k(\cdot, x)\rangle_{\mathcal{H}(k_P)} \mathrm{d}P(x) = \left\langle f, \int k(\cdot, x)\mathrm{d}P(x)\right\rangle_{\mathcal{H}(k_P)} = \langle f, 0\rangle_{\mathcal{H}(k_P)} = 0.$

The important point here is that $D_{k_P}(Q)$ does not require integrals with respect to $P$ to be computed. A kernel $k_P$ with $\int k_P(\cdot, x)\mathrm{d}P = 0$ will be called a *Stein kernel* (for $P$).

# Stein Discrepancy

## How can our algorithms be applied?

The apparent difficulty is that we cannot compute integrals with respect to $P$, such as $\int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})$, which are required for computation of MMD. A hint at a possible solution is provided by the following result:

### Lemma 16

*Suppose $k_P : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel with $\int k_P(\cdot, \boldsymbol{x})\mathrm{d}P = 0$ for all $\boldsymbol{x} \in \mathcal{X}$. Then*

$$D_{k_P}(Q) = D_{k_P}(P, Q) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d}Q \right|.$$

### Proof.

For all $f \in \mathcal{H}(k_P)$ it holds that $\int f \mathrm{d}P = 0$, whence the result. Indeed, from the reproducing property, and using Lemma 6 with Standing Assumption 1 to interchange integral with inner product, $\int f \mathrm{d}P = \int \langle f, k(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}(k_P)} \mathrm{d}P(\boldsymbol{x}) = \left\langle f, \int k(\cdot, \boldsymbol{x})\mathrm{d}P(\boldsymbol{x})\right\rangle_{\mathcal{H}(k_P)} = \langle f, 0\rangle_{\mathcal{H}(k_P)} = 0.$

The important point here is that $D_{k_P}(Q)$ does not require integrals with respect to $P$ to be computed. A kernel $k_P$ with $\int k_P(\cdot, \boldsymbol{x})\mathrm{d}P = 0$ will be called a *Stein kernel* (for $P$).

# Stein Discrepancy

As an example, consider a bounded linear operator

$$(\mathcal{A}_P g)(\boldsymbol{x}) = g(\boldsymbol{x}) - \int g \mathrm{d}P$$

acting on elements of an RKHS $\mathcal{H}(k)$.

If we apply $\mathcal{A}_P$ to both arguments of the kernel $k$, we obtain a Stein kernel

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) &= \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \\ &= k(\boldsymbol{x}, \boldsymbol{y}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}). \end{aligned} \tag{6}$$

Indeed, $\int k_P(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) = \int \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} \int \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} 0 = 0$, where interchange of $\mathcal{A}_P^{\boldsymbol{y}}$ and the integral is justified by noting that $\mathcal{A}_P^{\boldsymbol{y}}$ is a bounded linear operator and following similar reasoning to Lemma 6.

Unfortunately, the Stein kernel in (6) is not useful because it still involves the problematic integral $\int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$. A more useful construction of a Stein kernel is needed.

# Stein Discrepancy

As an example, consider a bounded linear operator

$$(\mathcal{A}_P g)(x) = g(x) - \int g \mathrm{d}P$$

acting on elements of an RKHS $\mathcal{H}(k)$.

If we apply $\mathcal{A}_P$ to both arguments of the kernel $k$, we obtain a Stein kernel

$$\begin{aligned} k_P(x, y) &= \mathcal{A}_P^y \mathcal{A}_P^x k(x, y) \\ &= k(x, y) - \int k(x, y) \mathrm{d}P(x) - \int k(x, y) \mathrm{d}P(y) + \iint k(x, y) \mathrm{d}P(x) \mathrm{d}P(y). \end{aligned} \tag{6}$$

Indeed, $\int k_P(\cdot, x) \mathrm{d}P(x) = \int \mathcal{A}_P^y \mathcal{A}_P^x k(x, y) \mathrm{d}P(x) = \mathcal{A}_P^y \int \mathcal{A}_P^x k(x, y) \mathrm{d}P(x) = \mathcal{A}_P^y 0 = 0$, where interchange of $\mathcal{A}_P^y$ and the integral is justified by noting that $\mathcal{A}_P^y$ is a bounded linear operator and following similar reasoning to Lemma 6.

Unfortunately, the Stein kernel in (6) is not useful because it still involves the problematic integral $\int k(\cdot, x) \mathrm{d}P(x)$. A more useful construction of a Stein kernel is needed.

# Stein Discrepancy

As an example, consider a bounded linear operator

$$(\mathcal{A}_P g)(\boldsymbol{x}) = g(\boldsymbol{x}) - \int g \mathrm{d}P$$

acting on elements of an RKHS $\mathcal{H}(k)$.

If we apply $\mathcal{A}_P$ to both arguments of the kernel $k$, we obtain a Stein kernel

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) &= \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \\ &= k(\boldsymbol{x}, \boldsymbol{y}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}). \end{aligned} \tag{6}$$

Indeed, $\int k_P(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) = \int \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} \int \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} 0 = 0$, where interchange of $\mathcal{A}_P^{\boldsymbol{y}}$ and the integral is justified by noting that $\mathcal{A}_P^{\boldsymbol{y}}$ is a bounded linear operator and following similar reasoning to Lemma 6.

Unfortunately, the Stein kernel in (6) is not useful because it still involves the problematic integral $\int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$. A more useful construction of a Stein kernel is needed.

# Stein Discrepancy

As an example, consider a bounded linear operator

$$(\mathcal{A}_P g)(\boldsymbol{x}) = g(\boldsymbol{x}) - \int g \mathrm{d}P$$

acting on elements of an RKHS $\mathcal{H}(k)$.

If we apply $\mathcal{A}_P$ to both arguments of the kernel $k$, we obtain a Stein kernel

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) &= \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \\ &= k(\boldsymbol{x}, \boldsymbol{y}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}). \end{aligned} \tag{6}$$

Indeed, $\int k_P(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) = \int \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} \int \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} 0 = 0$, where interchange of $\mathcal{A}_P^{\boldsymbol{y}}$ and the integral is justified by noting that $\mathcal{A}_P^{\boldsymbol{y}}$ is a bounded linear operator and following similar reasoning to Lemma 6.

Unfortunately, the Stein kernel in (6) is not useful because it still involves the problematic integral $\int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$. A more useful construction of a Stein kernel is needed.

# Stein Discrepancy

As an example, consider a bounded linear operator

$$(\mathcal{A}_P g)(\boldsymbol{x}) = g(\boldsymbol{x}) - \int g \mathrm{d}P$$

acting on elements of an RKHS $\mathcal{H}(k)$.

If we apply $\mathcal{A}_P$ to both arguments of the kernel $k$, we obtain a Stein kernel

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) &= \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \\ &= k(\boldsymbol{x}, \boldsymbol{y}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}). \end{aligned} \tag{6}$$

Indeed, $\int k_P(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) = \int \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} \int \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} 0 = 0$, where interchange of $\mathcal{A}_P^{\boldsymbol{y}}$ and the integral is justified by noting that $\mathcal{A}_P^{\boldsymbol{y}}$ is a bounded linear operator and following similar reasoning to Lemma 6.

Unfortunately, the Stein kernel in (6) is not useful because it still involves the problematic integral $\int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$. A more useful construction of a Stein kernel is needed.

# Stein Discrepancy

As an example, consider a bounded linear operator

$$(\mathcal{A}_P g)(\boldsymbol{x}) = g(\boldsymbol{x}) - \int g \mathrm{d}P$$

acting on elements of an RKHS $\mathcal{H}(k)$.

If we apply $\mathcal{A}_P$ to both arguments of the kernel $k$, we obtain a Stein kernel

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) &= \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \\ &= k(\boldsymbol{x}, \boldsymbol{y}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) - \int k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) + \iint k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) \mathrm{d}P(\boldsymbol{y}). \end{aligned} \tag{6}$$

Indeed, $\int k_P(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x}) = \int \mathcal{A}_P^{\boldsymbol{y}} \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} \int \mathcal{A}_P^{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{x}) = \mathcal{A}_P^{\boldsymbol{y}} 0 = 0$, where interchange of $\mathcal{A}_P^{\boldsymbol{y}}$ and the integral is justified by noting that $\mathcal{A}_P^{\boldsymbol{y}}$ is a bounded linear operator and following similar reasoning to Lemma 6.

Unfortunately, the Stein kernel in (6) is not useful because it still involves the problematic integral $\int k(\cdot, \boldsymbol{x}) \mathrm{d}P(\boldsymbol{x})$. A more useful construction of a Stein kernel is needed.

# Stein Operators

**Notation:** Let $\nabla f = (\partial_{x_1} f, \ldots, \partial_{x_d} f)^\top$ for differentiable functions $f : \mathbb{R}^d \to \mathbb{R}$.

### Standing Assumption 2

*The distribution $P$ admits a positive and differentiable PDF such that $x \mapsto (\nabla \log p)(x)$ is Lipschitz.*

### Definition 17 (Canonical Stein operator)

For a distribution $P$ admitting a positive and differentiable density $p$ on $\mathbb{R}^d$, we define the *canonical Stein operator*

$$(\mathcal{A}_P g)(x) = (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log p)(x)$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

The canonical Stein operator was introduced (for Gaussian $P$) in Stein [1972]. Importantly, observe that

$$(\nabla \log p)(x) = \frac{(\nabla p)(x)}{p(x)} = \frac{\frac{1}{Z}(\nabla \tilde{p})(x)}{\frac{1}{Z}\tilde{p}(x)} = \frac{(\nabla \tilde{p})(x)}{\tilde{p}(x)} = (\nabla \log \tilde{p})(x),$$

which can be computed without knowledge of $p$ or $Z$, provided $\tilde{p}$ and $\nabla \tilde{p}$ can be evaluated.

# Stein Operators

**Notation:** Let $\nabla f = (\partial_{x_1} f, \ldots, \partial_{x_d} f)^\top$ for differentiable functions $f : \mathbb{R}^d \to \mathbb{R}$.

## Standing Assumption 2

*The distribution $P$ admits a positive and differentiable PDF such that $x \mapsto (\nabla \log p)(x)$ is Lipschitz.*

## Definition 17 (Canonical Stein operator)

For a distribution $P$ admitting a positive and differentiable density $p$ on $\mathbb{R}^d$, we define the *canonical Stein operator*

$$(\mathcal{A}_P g)(x) = (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log p)(x)$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

The canonical Stein operator was introduced (for Gaussian $P$) in Stein [1972]. Importantly, observe that

$$(\nabla \log p)(x) = \frac{(\nabla p)(x)}{p(x)} = \frac{\frac{1}{Z}(\nabla \tilde{p})(x)}{\frac{1}{Z}\tilde{p}(x)} = \frac{(\nabla \tilde{p})(x)}{\tilde{p}(x)} = (\nabla \log \tilde{p})(x),$$

which can be computed without knowledge of $p$ or $Z$, provided $\tilde{p}$ and $\nabla \tilde{p}$ can be evaluated.

# Stein Operators

**Notation:** Let $\nabla f = (\partial_{x_1} f, \ldots, \partial_{x_d} f)^\top$ for differentiable functions $f : \mathbb{R}^d \to \mathbb{R}$.

## Standing Assumption 2

*The distribution $P$ admits a positive and differentiable PDF such that $\boldsymbol{x} \mapsto (\nabla \log p)(\boldsymbol{x})$ is Lipschitz.*

## Definition 17 (Canonical Stein operator)

For a distribution $P$ admitting a positive and differentiable density $p$ on $\mathbb{R}^d$, we define the *canonical Stein operator*

$$(\mathcal{A}_P g)(\boldsymbol{x}) = (\nabla \cdot g)(\boldsymbol{x}) + g(\boldsymbol{x}) \cdot (\nabla \log p)(\boldsymbol{x})$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $\boldsymbol{x} \in \mathbb{R}^d$.

The canonical Stein operator was introduced (for Gaussian $P$) in Stein [1972]. Importantly, observe that

$$(\nabla \log p)(x) = \frac{(\nabla p)(x)}{p(x)} = \frac{\frac{1}{Z}(\nabla \tilde{p})(x)}{\frac{1}{Z}\tilde{p}(x)} = \frac{(\nabla \tilde{p})(x)}{\tilde{p}(x)} = (\nabla \log \tilde{p})(x),$$

which can be computed without knowledge of $p$ or $Z$, provided $\tilde{p}$ and $\nabla \tilde{p}$ can be evaluated.

# Stein Operators

**Notation:** Let $\nabla f = (\partial_{x_1} f, \ldots, \partial_{x_d} f)^\top$ for differentiable functions $f : \mathbb{R}^d \to \mathbb{R}$.

## Standing Assumption 2

*The distribution $P$ admits a positive and differentiable PDF such that $\boldsymbol{x} \mapsto (\nabla \log p)(\boldsymbol{x})$ is Lipschitz.*

## Definition 17 (Canonical Stein operator)

For a distribution $P$ admitting a positive and differentiable density $p$ on $\mathbb{R}^d$, we define the *canonical Stein operator*

$$(\mathcal{A}_P g)(\boldsymbol{x}) = (\nabla \cdot g)(\boldsymbol{x}) + g(\boldsymbol{x}) \cdot (\nabla \log p)(\boldsymbol{x})$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $\boldsymbol{x} \in \mathbb{R}^d$.

The canonical Stein operator was introduced (for Gaussian $P$) in Stein [1972]. Importantly, observe that

$$(\nabla \log p)(\boldsymbol{x}) = \frac{(\nabla p)(\boldsymbol{x})}{p(\boldsymbol{x})} = \frac{\frac{1}{Z}(\nabla \tilde{p})(\boldsymbol{x})}{\frac{1}{Z}\tilde{p}(\boldsymbol{x})} = \frac{(\nabla \tilde{p})(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x})} = (\nabla \log \tilde{p})(\boldsymbol{x}),$$

which can be computed without knowledge of $p$ or $Z$, provided $\tilde{p}$ and $\nabla \tilde{p}$ can be evaluated.

# Stein Operators

Now we apply the Stein operator $\mathcal{A}_P$ to a standard kernel $k$, to obtain the following Stein kernel:

**Lemma 18**

*Suppose the kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \partial^{(\boldsymbol{\alpha},\boldsymbol{\beta})} k(\boldsymbol{x}, \boldsymbol{y})$ continuous and uniformly bounded for all $|\boldsymbol{\alpha}|, |\boldsymbol{\beta}| \leq 1$. Suppose $\int \|\nabla \log p(\boldsymbol{x})\| \mathrm{d}P(\boldsymbol{x}) < \infty$ and that $\sup_{\|\boldsymbol{x}\| \geq r} r^{d-1} p(\boldsymbol{x}) \to 0$ as $r \to \infty$. Then*

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) = \nabla_x \cdot \nabla_y k(\boldsymbol{x}, \boldsymbol{y}) + \nabla_x \log p(\boldsymbol{x}) \cdot \nabla_y k(\boldsymbol{x}, \boldsymbol{y}) + \nabla_y \log p(\boldsymbol{y}) \cdot \nabla_x k(\boldsymbol{x}, \boldsymbol{y}) \\ + (\nabla_x \log p(\boldsymbol{x})) \cdot (\nabla_y \log p(\boldsymbol{y})) k(\boldsymbol{x}, \boldsymbol{y}) \end{aligned}$$

*is a kernel with $\int k_P(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$.*

**Proof.**

First notice that

$$k_P(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{A}_P^y \begin{bmatrix} \vdots \\ \nabla_{x_i} k(\boldsymbol{x}, \boldsymbol{y}) + k(\boldsymbol{x}, \boldsymbol{y}) \nabla_{x_i} \log p(\boldsymbol{x}) \\ \vdots \end{bmatrix} = \mathcal{A}_P^y g(\boldsymbol{y})$$

where, under our assumptions, (a) $\boldsymbol{y} \mapsto g(\boldsymbol{y})$ is bounded, and (b) $\boldsymbol{y} \mapsto \nabla_y \cdot g(\boldsymbol{y})$ is integrable with respect to $P$. Thus it suffices to show that $\int \mathcal{A}_P g \mathrm{d}P = 0$ for *all* vector fields $g$ for which (a) and (b) hold.

# Stein Operators

Now we apply the Stein operator $\mathcal{A}_P$ to a standard kernel $k$, to obtain the following Stein kernel:

## Lemma 18

*Suppose the kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has $(x, y) \mapsto \partial^{(\alpha,\beta)} k(x, y)$ continuous and uniformly bounded for all $|\alpha|, |\beta| \leq 1$. Suppose $\int \|\nabla \log p(x)\| \mathrm{d}P(x) < \infty$ and that $\sup_{\|x\| \geq r} r^{d-1} p(x) \to 0$ as $r \to \infty$. Then*

$$\begin{aligned} k_P(x, y) = \nabla_x \cdot \nabla_y k(x, y) + \nabla_x \log p(x) \cdot \nabla_y k(x, y) + \nabla_y \log p(y) \cdot \nabla_x k(x, y) \\ + (\nabla_x \log p(x)) \cdot (\nabla_y \log p(y)) k(x, y) \end{aligned}$$

*is a kernel with $\int k_P(x, y) \mathrm{d}P(y) = 0$ for all $x \in \mathbb{R}^d$.*

## Proof.

First notice that

$$k_P(x, y) = \mathcal{A}_P^y \begin{bmatrix} \vdots \\ \nabla_{x_i} k(x, y) + k(x, y) \nabla_{x_i} \log p(x) \\ \vdots \end{bmatrix} = \mathcal{A}_P^y g(y)$$

where, under our assumptions, (a) $y \mapsto g(y)$ is bounded, and (b) $y \mapsto \nabla_y \cdot g(y)$ is integrable with respect to $P$. Thus it suffices to show that $\int \mathcal{A}_P g \mathrm{d}P = 0$ for *all* vector fields $g$ for which (a) and (b) hold.

# Stein Operators

Now we apply the Stein operator $\mathcal{A}_P$ to a standard kernel $k$, to obtain the following Stein kernel:

## Lemma 18

*Suppose the kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \partial^{(\boldsymbol{\alpha},\boldsymbol{\beta})} k(\boldsymbol{x}, \boldsymbol{y})$ continuous and uniformly bounded for all $|\boldsymbol{\alpha}|, |\boldsymbol{\beta}| \leq 1$. Suppose $\int \|\nabla \log p(\boldsymbol{x})\| \mathrm{d}P(\boldsymbol{x}) < \infty$ and that $\sup_{\|\boldsymbol{x}\| \geq r} r^{d-1} p(\boldsymbol{x}) \to 0$ as $r \to \infty$. Then*

$$\begin{aligned} k_P(\boldsymbol{x}, \boldsymbol{y}) = \nabla_{\boldsymbol{x}} \cdot \nabla_{\boldsymbol{y}} k(\boldsymbol{x}, \boldsymbol{y}) + \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) \cdot \nabla_{\boldsymbol{y}} k(\boldsymbol{x}, \boldsymbol{y}) + \nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) \cdot \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{y}) \\ + (\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})) \cdot (\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y})) k(\boldsymbol{x}, \boldsymbol{y}) \end{aligned}$$

*is a kernel with $\int k_P(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}P(\boldsymbol{y}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$.*

## Proof.

Let $g$ be such a vector field, and let $B_r = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq r\}$ and $S_r = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| = r\}$. The main idea is to apply the divergence theorem (i.e. integrate by parts):

$$\begin{aligned} \int \mathcal{A}_P g \mathrm{d}P &= \int (\nabla \cdot g) + g \cdot (\nabla \log p) \mathrm{d}P = \int (\nabla \cdot (pg))(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \lim_{r \to \infty} \int_{B_r} (\nabla \cdot (pg))(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \lim_{r \to \infty} \oint_{S_r} p(\boldsymbol{x})(g(\boldsymbol{x}) \cdot n(\boldsymbol{x})) \mathrm{d}\boldsymbol{x} \end{aligned}$$

where $n(\boldsymbol{x})$ is the outward unit normal to $S_r$ at $\boldsymbol{x}$. (The regularity assumptions ensure that the integrals $\int (\nabla \cdot g) \mathrm{d}P$ and $\int g \cdot (\nabla \log p) \mathrm{d}P$ exist.)

# Stein Operators

Now we apply the Stein operator $\mathcal{A}_P$ to a standard kernel $k$, to obtain the following Stein kernel:

## Lemma 18

*Suppose the kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has $(x, y) \mapsto \partial^{(\alpha,\beta)} k(x, y)$ continuous and uniformly bounded for all $|\alpha|, |\beta| \leq 1$. Suppose $\int \|\nabla \log p(x)\| \mathrm{d}P(x) < \infty$ and that $\sup_{\|x\| \geq r} r^{d-1} p(x) \to 0$ as $r \to \infty$. Then*

$$k_P(x, y) = \nabla_x \cdot \nabla_y k(x, y) + \nabla_x \log p(x) \cdot \nabla_y k(x, y) + \nabla_y \log p(y) \cdot \nabla_x k(x, y) + (\nabla_x \log p(x)) \cdot (\nabla_y \log p(y)) k(x, y)$$

*is a kernel with $\int k_P(x, y) \mathrm{d}P(y) = 0$ for all $x \in \mathbb{R}^d$.*

## Proof.

Now

$$\begin{aligned} \oint_{S_r} p(x)(g(x) \cdot n(x)) \mathrm{d}x &\leq \|g\|_\infty \sup_{\|x\| \geq r} p(x) \oint_{S_r} \mathrm{d}x \\ &= \|g\|_\infty \sup_{\|x\| \geq r} p(x) \frac{2\pi^{d/2}}{\Gamma(d/2)} r^{d-1} \\ &\to 0 \text{ as } r \to \infty, \end{aligned}$$

where we have used the formula for the surface area of the radius $r$ sphere in $\mathbb{R}^d$.

# Kernel Stein Discrepancy

For $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ and $k_P$ defined in Lemma 18, we obtain the kernel Stein discrepancy (KSD):

$$D_{k_P}(Q_n) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d} Q_n \right| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n w_i w_j k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

As with MMD, we can establish properties analogous to characteristicness and convergence control for KSD. Here we focus on convergence control:

**Theorem 19 (Gorham and Mackey [2017])**

*Let $P$ be distantly dissipative [see Gorham et al., 2019]. Consider the kernel*

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\sigma^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{-\beta}$$

*for some fixed $\sigma > 0$ and a fixed exponent $\beta \in (0, 1)$. Then $D_{k_P}(Q_n) \to 0$ implies $Q_n \Rightarrow P$.*

However, unlike MMD, the construction of KSD suffers from *blindness to mixing proportions*; figure taken from Wenliang and Kanagawa [2020]:

# Kernel Stein Discrepancy

For $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ and $k_P$ defined in Lemma 18, we obtain the KSD:

$$D_{k_P}(Q_n) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d} Q_n \right| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n w_i w_j k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

As with MMD, we can establish properties analogous to characteristicness and convergence control for KSD. Here we focus on convergence control:

**Theorem 19 (Gorham and Mackey [2017])**

*Let $P$ be distantly dissipative [see Gorham et al., 2019]. Consider the kernel*

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\sigma^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{-\beta}$$

*for some fixed $\sigma > 0$ and a fixed exponent $\beta \in (0, 1)$. Then $D_{k_P}(Q_n) \to 0$ implies $Q_n \Rightarrow P$.*

However, unlike MMD, the construction of KSD suffers from *blindness to mixing proportions*; figure taken from Wenliang and Kanagawa [2020]:

# Kernel Stein Discrepancy

For $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ and $k_P$ defined in Lemma 18, we obtain the KSD:

$$D_{k_P}(Q_n) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d} Q_n \right| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n w_i w_j k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

As with MMD, we can establish properties analogous to characteristicness and convergence control for KSD. Here we focus on convergence control:

**Theorem 19 (Gorham and Mackey [2017])**

*Let $P$ be distantly dissipative [see Gorham et al., 2019]. Consider the kernel*

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\sigma^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{-\beta}$$

*for some fixed $\sigma > 0$ and a fixed exponent $\beta \in (0, 1)$. Then $D_{k_P}(Q_n) \to 0$ implies $Q_n \Rightarrow P$.*

However, unlike MMD, the construction of KSD suffers from *blindness to mixing proportions*; figure taken from Wenliang and Kanagawa [2020]:

# Kernel Stein Discrepancy

For $Q_n = \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i)$ and $k_P$ defined in Lemma 18, we obtain the KSD:

$$D_{k_P}(Q_n) = \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} \left| \int f \mathrm{d} Q_n \right| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n w_i w_j k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

As with MMD, we can establish properties analogous to characteristicness and convergence control for KSD. Here we focus on convergence control:

## Theorem 19 (Gorham and Mackey [2017])

*Let $P$ be distantly dissipative [see Gorham et al., 2019]. Consider the kernel*

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\sigma^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2)^{-\beta}$$

*for some fixed $\sigma > 0$ and a fixed exponent $\beta \in (0, 1)$. Then $D_{k_P}(Q_n) \to 0$ implies $Q_n \Rightarrow P$.*

However, unlike MMD, the construction of KSD suffers from *blindness to mixing proportions*; figure taken from Wenliang and Kanagawa [2020]:

# Optimal Quantisation

**The algorithms previously described for MMD can immediately be applied to KSD!**

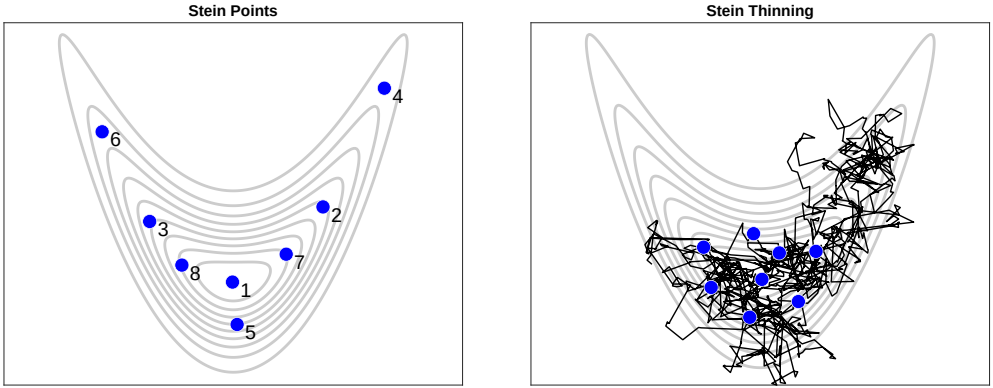



Figure: *Stein points* (left) *and Stein thinning* (right)*:* Stein points are generated by sequential (greedy) minimisation of KSD to select $n = 8$ states $\{\boldsymbol{x}_i\}_{i=1}^n$ in an approximation $Q_n$ to $P$. The numbers indicate the order in which the states $\boldsymbol{x}_i$ were selected. Stein thinning restricts the continuous inner-loop optimisation problem in Stein points to a discrete search over a MCMC sample path (black).

# Optimal Approximation

Optimal weights can again be computed, with a slight twist:

**Lemma 20**

*Let $x_1, \ldots, x_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\substack{\boldsymbol{w} \in \mathbb{R}^n \\ \mathbf{1}^\top \boldsymbol{w} = 1}}{\arg\min} D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)$$

*are $\boldsymbol{w} = (K_P^{-1}\mathbf{1})/(\mathbf{1}^\top K_P^{-1}\mathbf{1})$, where $[K_P]_{ij} = k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)$. [Note the constraint $w_1 + \cdots + w_n = 1$.]*

**Proof.**

From the explicit form of MMD we have

$$D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)^2 = \boldsymbol{w}^\top K_P \boldsymbol{w},$$

so the optimisation problem is

$$\arg\min \boldsymbol{w}^\top K_P \boldsymbol{w} \qquad \text{s.t.} \qquad \mathbf{1}^\top \boldsymbol{w} = 1.$$

This can be solved using the method of Lagrange multipliers to obtain the stated result.

# Optimal Approximation

Optimal weights can again be computed, with a slight twist:

**Lemma 20**

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\substack{\boldsymbol{w} \in \mathbb{R}^n \\ \mathbf{1}^\top \boldsymbol{w} = 1}}{\arg\min} D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)$$

*are $\boldsymbol{w} = (K_P^{-1}\mathbf{1})/(\mathbf{1}^\top K_P^{-1}\mathbf{1})$, where $[K_P]_{ij} = k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)$. [Note the constraint $w_1 + \cdots + w_n = 1$.]*

**Proof.**

From the explicit form of MMD we have

$$D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)^2 = \boldsymbol{w}^\top K_P \boldsymbol{w},$$

so the optimisation problem is

$$\arg\min \boldsymbol{w}^\top K_P \boldsymbol{w} \qquad \text{s.t.} \qquad \mathbf{1}^\top \boldsymbol{w} = 1.$$

This can be solved using the method of Lagrange multipliers to obtain the stated result.

# Optimal Approximation

Optimal weights can again be computed, with a slight twist:

**Lemma 20**

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\substack{\boldsymbol{w} \in \mathbb{R}^n \\ \mathbf{1}^\top \boldsymbol{w} = 1}}{\arg\min} D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)$$

*are $\boldsymbol{w} = (K_P^{-1}\mathbf{1})/(\mathbf{1}^\top K_P^{-1}\mathbf{1})$, where $[K_P]_{ij} = k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)$. [Note the constraint $w_1 + \cdots + w_n = 1$.]*

**Proof.**

From the explicit form of MMD we have

$$D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)^2 = \boldsymbol{w}^\top K_P \boldsymbol{w},$$

so the optimisation problem is

$$\arg\min \boldsymbol{w}^\top K_P \boldsymbol{w} \qquad \text{s.t.} \qquad \mathbf{1}^\top \boldsymbol{w} = 1.$$

This can be solved using the method of Lagrange multipliers to obtain the stated result.

# Optimal Approximation

Optimal weights can again be computed, with a slight twist:

**Lemma 20**

*Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ be distinct. The optimal weights*

$$\underset{\substack{\boldsymbol{w} \in \mathbb{R}^n \\ \mathbf{1}^\top \boldsymbol{w} = 1}}{\arg\min} D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)$$

*are $\boldsymbol{w} = (\boldsymbol{K}_P^{-1}\mathbf{1})/(\mathbf{1}^\top \boldsymbol{K}_P^{-1}\mathbf{1})$, where $[K_P]_{ij} = k_P(\boldsymbol{x}_i, \boldsymbol{x}_j)$. [Note the constraint $w_1 + \cdots + w_n = 1$.]*

**Proof.**

From the explicit form of MMD we have

$$D_{k_P} \left( \sum_{i=1}^n w_i \delta(\boldsymbol{x}_i) \right)^2 = \boldsymbol{w}^\top \boldsymbol{K}_P \boldsymbol{w},$$

so the optimisation problem is

$$\arg\min \boldsymbol{w}^\top \boldsymbol{K}_P \boldsymbol{w} \qquad \text{s.t.} \qquad \mathbf{1}^\top \boldsymbol{w} = 1.$$

This can be solved using the method of Lagrange multipliers to obtain the stated result.

# Optimal Approximation

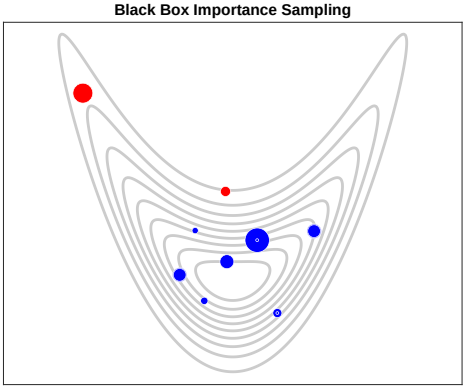



Figure: *Black box importance sampling* [Liu and Lee, 2017, Hodgkinson et al., 2020]: In black box importance sampling the weights $w_1, \ldots, w_n$ are obtained by minimising KSD in the manner of Lemma 20. Blue indicates states $\boldsymbol{x}_i$ with positive weights $w_i > 0$, while red indicates negative weights $w_i < 0$. The size of the circles is proportional to $|w_i|$.

Complexity $= O(n^3)$ and symmetry exploits fail, but $P_{\mathrm{ST}} \to P_{\mathrm{BBIS}}$ as $m \to \infty$ for $n$ fixed.

# Optimal Approximation

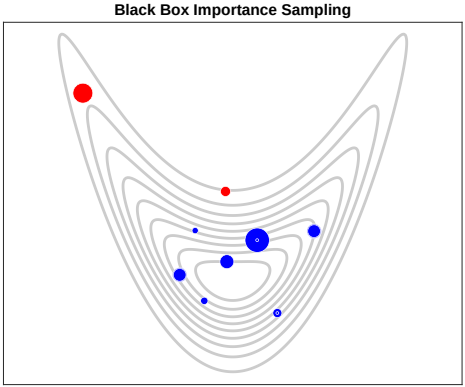
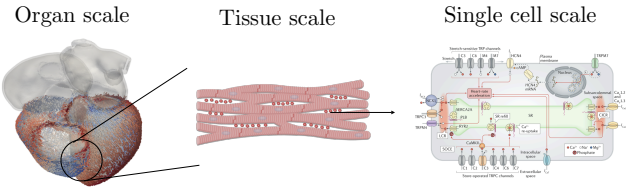



Figure: *Black box importance sampling* [Liu and Lee, 2017, Hodgkinson et al., 2020]: In black box importance sampling the weights $w_1, \ldots, w_n$ are obtained by minimising KSD in the manner of Lemma 20. Blue indicates states $\boldsymbol{x}_i$ with positive weights $w_i > 0$, while red indicates negative weights $w_i < 0$. The size of the circles is proportional to $|w_i|$.

Complexity $= O(n^3)$ and symmetry exploits fail, but $P_{\mathrm{ST}} \to P_{\mathrm{BBIS}}$ as $m \to \infty$ for $n$ fixed.

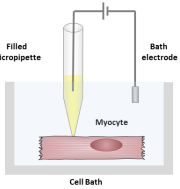
# Case Study: Cardiac Digital Twins

# Cardiac Models

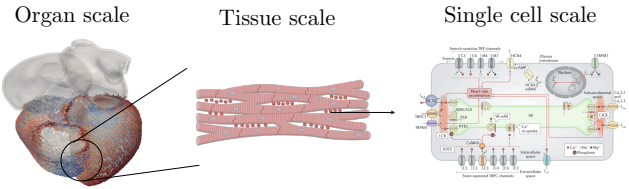



- Cardiac function determined by integrated action of myocytes.
- Calcium is intracellular end-point signal driving contraction.
- **Question**: is there cell-to-cell variability?
  - "Extreme" cells might be related to pathologies (arrythmias - ventricular fibrillation)?
  - Role for ageing in the extent of heterogeneity?
  - ...

- 25 myocytes, from different rat hearts, fed in a dish.
- **Patch clamp experiment**: micropipette controls cell membrane potential, and adds a drug to
  - block transients of ions that are not calcium;
  - stimulate activity of calcium handling proteins (facilitate system identification).

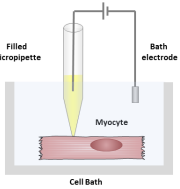


# Cardiac Models

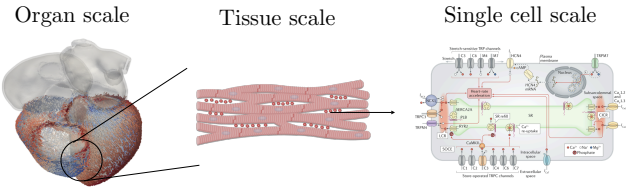



- Cardiac function determined by integrated action of myocytes.
- Calcium is intracellular end-point signal driving contraction.
- **Question**: is there cell-to-cell variability?
  - "Extreme" cells might be related to pathologies (arrythmias - ventricular fibrillation)?
  - Role for ageing in the extent of heterogeneity?
  - ...

- 25 myocytes, from different rat hearts, fed in a dish.
- **Patch clamp experiment**: micropipette controls cell membrane potential, and adds a drug to
  - block transients of ions that are not calcium;
  - stimulate activity of calcium handling proteins (facilitate system identification).

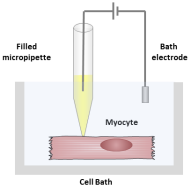


# Cardiac Models




- Cardiac function determined by integrated action of myocytes.
- Calcium is intracellular end-point signal driving contraction.
- **Question**: is there cell-to-cell variability?
  - "Extreme" cells might be related to pathologies (arrythmias - ventricular fibrillation)?
  - Role for ageing in the extent of heterogeneity?
  - ...

- 25 myocytes, from different rat hearts, fed in a dish.
- **Patch clamp experiment**: micropipette controls cell membrane potential, and adds a drug to
  - block transients of ions that are not calcium;
  - stimulate activity of calcium handling proteins (facilitate system identification).



# Statistical Model

- The experiment is modelled using mass action kinetics, in the form of an ordinary differential equation (ODE) with unknown parameters denoted $\theta \in \Theta$.
- The generic form of the ODE is

$$u(0) = u_0, \qquad \frac{\mathrm{d}u_1}{\mathrm{d}t} = f_1(t, u; \theta),$$
$$\vdots$$
$$\frac{\mathrm{d}u_p}{\mathrm{d}t} = f_p(t, u; \theta),$$

  and the solution at time $t$ is denoted $u(t; \theta)$. Here $p = 6$ and $d := \dim(\Theta) = 38$.
- The data are related to the ODE via a measurement error model

$$\pi(y|\theta) := \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - u_1(t_i; \theta))^2}{2\sigma^2}\right),$$

  where $\sigma$ is the resolution of the measurement equipment. (Less trivial measurement models may be needed, including temporal components in the measurement error.)

  (We assume $u_0$ and $\sigma$ are known in this talk.)

# Statistical Model

- The experiment is modelled using mass action kinetics, in the form of an ordinary differential equation (ODE) with unknown parameters denoted $\theta \in \Theta$.
- The generic form of the ODE is

$$u(0) = u_0, \qquad \begin{aligned} \frac{\mathrm{d}u_1}{\mathrm{d}t} &= f_1(t, u; \theta), \\ &\vdots \\ \frac{\mathrm{d}u_p}{\mathrm{d}t} &= f_p(t, u; \theta), \end{aligned}$$

  and the solution at time $t$ is denoted $u(t; \theta)$. Here $p = 6$ and $d := \dim(\Theta) = 38$.
- The data are related to the ODE via a measurement error model

$$\pi(y|\theta) := \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - u_1(t_i; \theta))^2}{2\sigma^2}\right),$$

  where $\sigma$ is the resolution of the measurement equipment. (Less trivial measurement models may be needed, including temporal components in the measurement error.)

  (We assume $u_0$ and $\sigma$ are known in this talk.)

# Statistical Model

- The experiment is modelled using mass action kinetics, in the form of an ordinary differential equation (ODE) with unknown parameters denoted $\theta \in \Theta$.
- The generic form of the ODE is

$$u(0) = u_0, \qquad \begin{aligned} \frac{\mathrm{d}u_1}{\mathrm{d}t} &= f_1(t, u; \theta), \\ &\vdots \\ \frac{\mathrm{d}u_p}{\mathrm{d}t} &= f_p(t, u; \theta), \end{aligned}$$

  and the solution at time $t$ is denoted $u(t; \theta)$. Here $p = 6$ and $d := \dim(\Theta) = 38$.
- The data are related to the ODE via a measurement error model

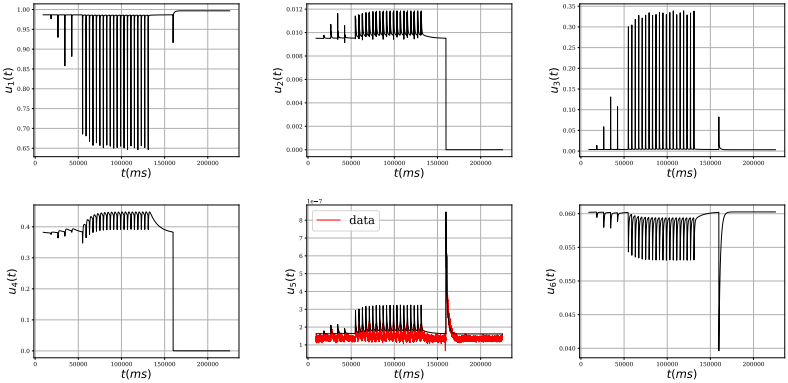
$$\pi(y|\theta) := \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - u_1(t_i; \theta))^2}{2\sigma^2}\right),$$

  where $\sigma$ is the resolution of the measurement equipment. (Less trivial measurement models may be needed, including temporal components in the measurement error.)

  (We assume $u_0$ and $\sigma$ are known in this talk.)

# Inverse Problem

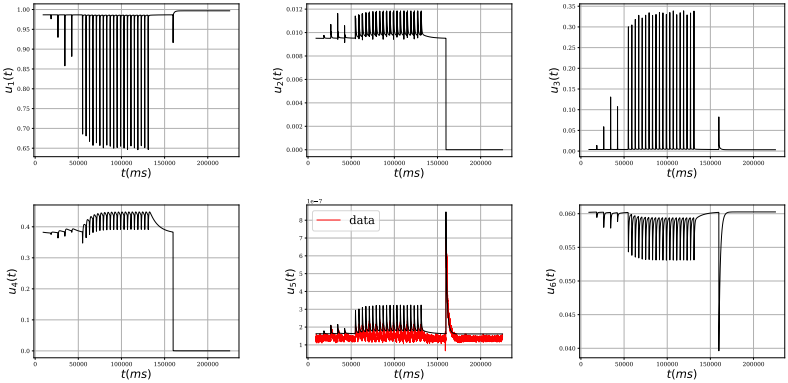
- Is cell-to-cell heterogeneity observed at the level of the parameters $\theta$? (The figure corresponds to slightly perturbed literature values for $\theta$).
- To answer this we need to solve an inverse problem for $\theta$ in the statistical model.




- Challenge: ODE solver failure.

# Inverse Problem

- Is cell-to-cell heterogeneity observed at the level of the parameters $\theta$? (The figure corresponds to slightly perturbed literature values for $\theta$).
- To answer this we need to solve an inverse problem for $\theta$ in the statistical model.




- Challenge: ODE solver failure.

# De-Biasing of Markov Chain Monte Carlo

MCMC samples are biased by ODE solver failure, but Stein Thinning does not require MCMC to be $P$-invariant - as long as the relevant part of the parameter space is explored:

**Theorem 21 ([Riabiz et al., 2022])**

*Let $(\theta_i)_{i\in\mathbb{N}}$ be a $Q$-invariant, time-homogeneous, reversible Markov chain, such that $P$ is absolutely continuous with respect to $Q$ and*

- *$(\theta_i)_{i\in\mathbb{N}}$ is $V$-uniformly ergodic with $V(\theta) \geq \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta)\sqrt{k_P(\theta,\theta)}$*
- $\sup_{i\in\mathbb{N}} \mathbb{E}[\frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)\sqrt{k_P(\theta_i,\theta_i)}V(\theta_i)] < \infty$
- $\exists\gamma > 0$ *s.t.* $b := \sup_{i\in\mathbb{N}} \mathbb{E}[e^{\gamma \max(1, \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)^2)k_P(\theta_i,\theta_i)}] < \infty$.

*Then the output of Stein Thinning satisfies*

$$\frac{1}{m}\sum_{i\in S}\delta(\theta_i) \Rightarrow P$$

*almost surely as $n, m \to \infty$ with $m \leq n$ and $\log(n) = O(m^{\beta/2})$ for some $\beta < 1$.*

# De-Biasing of Markov Chain Monte Carlo

MCMC samples are biased by ODE solver failure, but Stein Thinning does not require MCMC to be $P$-invariant - as long as the relevant part of the parameter space is explored:

### Theorem 21 ([Riabiz et al., 2022])

*Let $(\theta_i)_{i\in\mathbb{N}}$ be a $Q$-invariant, time-homogeneous, reversible Markov chain, such that $P$ is absolutely continuous with respect to $Q$ and*

- $(\theta_i)_{i\in\mathbb{N}}$ *is $V$-uniformly ergodic with* $V(\theta) \geq \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta)\sqrt{k_P(\theta,\theta)}$
- $\sup_{i\in\mathbb{N}} \mathbb{E}[\frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)\sqrt{k_P(\theta_i,\theta_i)}V(\theta_i)] < \infty$
- $\exists\gamma > 0$ *s.t.* $b := \sup_{i\in\mathbb{N}} \mathbb{E}[e^{\gamma \max(1,\frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)^2)k_P(\theta_i,\theta_i)}] < \infty$.

*Then the output of Stein Thinning satisfies*

$$\frac{1}{m}\sum_{i\in S}\delta(\theta_i) \Rightarrow P$$

*almost surely as $n, m \to \infty$ with $m \leq n$ and $\log(n) = O(m^{\beta/2})$ for some $\beta < 1$.*

# De-Biasing of Markov Chain Monte Carlo

MCMC samples are biased by ODE solver failure, but Stein Thinning does not require MCMC to be $P$-invariant - as long as the relevant part of the parameter space is explored:

### Theorem 21 ([Riabiz et al., 2022])

*Let $(\theta_i)_{i\in\mathbb{N}}$ be a $Q$-invariant, time-homogeneous, reversible Markov chain, such that $P$ is absolutely continuous with respect to $Q$ and*

- $(\theta_i)_{i\in\mathbb{N}}$ *is $V$-uniformly ergodic with* $V(\theta) \geq \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta)\sqrt{k_P(\theta,\theta)}$
- $\sup_{i\in\mathbb{N}} \mathbb{E}[\frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)\sqrt{k_P(\theta_i,\theta_i)}V(\theta_i)] < \infty$
- $\exists\gamma > 0$ *s.t.* $b := \sup_{i\in\mathbb{N}} \mathbb{E}[e^{\gamma \max(1, \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)^2)k_P(\theta_i,\theta_i)}] < \infty$.

*Then the output of Stein Thinning satisfies*

$$\frac{1}{m}\sum_{i\in S}\delta(\theta_i) \Rightarrow P$$

*almost surely as $n, m \to \infty$ with $m \leq n$ and $\log(n) = O(m^{\beta/2})$ for some $\beta < 1$.*

# De-Biasing of Markov Chain Monte Carlo

MCMC samples are biased by ODE solver failure, but Stein Thinning does not require MCMC to be $P$-invariant - as long as the relevant part of the parameter space is explored:

## Theorem 21 ([Riabiz et al., 2022])

*Let $(\theta_i)_{i\in\mathbb{N}}$ be a $Q$-invariant, time-homogeneous, reversible Markov chain, such that $P$ is absolutely continuous with respect to $Q$ and*

- $(\theta_i)_{i\in\mathbb{N}}$ *is $V$-uniformly ergodic with* $V(\theta) \geq \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta)\sqrt{k_P(\theta,\theta)}$
- $\sup_{i\in\mathbb{N}} \mathbb{E}[\frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)\sqrt{k_P(\theta_i,\theta_i)}V(\theta_i)] < \infty$
- $\exists\gamma > 0$ *s.t.* $b := \sup_{i\in\mathbb{N}} \mathbb{E}[e^{\gamma \max(1, \frac{\mathrm{d}P}{\mathrm{d}Q}(\theta_i)^2)k_P(\theta_i,\theta_i)}] < \infty$.

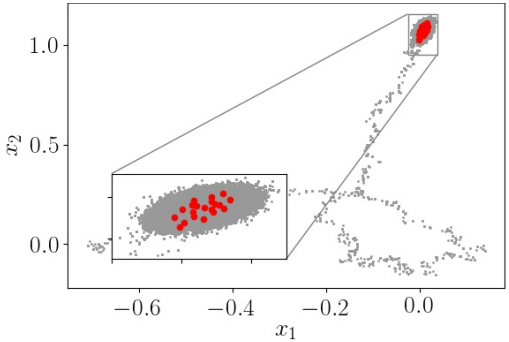
*Then the output of Stein Thinning satisfies*

$$\frac{1}{m}\sum_{i\in S}\delta(\theta_i) \Rightarrow P$$

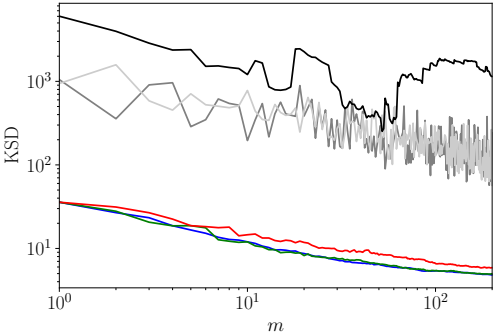
*almost surely as $n, m \to \infty$ with $m \leq n$ and $\log(n) = O(m^{\beta/2})$ for some $\beta < 1$.*

# Illustrative Application to Differential Equation Constrained Inverse Problems

Goodwin oscillator; $d = 4$ parameters to be estimated. (Red dots are Stein Thinning, while gray dots are MCMC.)




Cardiac model; $d = 38$ parameters to be esitmated. (Blue, red, and green are Stein Thinning, while black are MCMC.)



# Future Directions, Open Questions and Challenges

# Scalable Stein Thinning

Greedy selection may be sub-optimal. Also, the cost of selecting $m$ points from $n$ using Stein Thinning is high, at $O(m^2 n)$.

- A **non-myopic** algorithm selects $s$ points simultaneously.
- A **mini-batch** algorithm searches over a subset of $b \ll n$ candidates at each step.

Full details in Teymur et al. [2021].

# Scalable Stein Thinning

Greedy selection may be sub-optimal. Also, the cost of selecting $m$ points from $n$ using Stein Thinning is high, at $O(m^2 n)$.

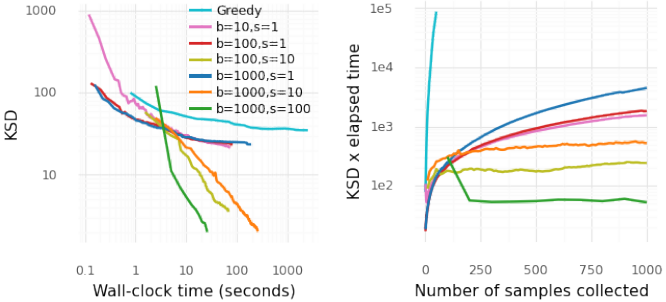
- A **non-myopic** algorithm selects $s$ points simultaneously.
- A **mini-batch** algorithm searches over a subset of $b \ll n$ candidates at each step.

Full details in Teymur et al. [2021].

# Scalable Stein Thinning

Greedy selection may be sub-optimal. Also, the cost of selecting $m$ points from $n$ using Stein Thinning is high, at $O(m^2 n)$.

- A **non-myopic** algorithm selects $s$ points simultaneously.
- A **mini-batch** algorithm searches over a subset of $b \ll n$ candidates at each step.




Full details in Teymur et al. [2021].

# Gradient-Free Kernel Stein Discrepancy

Gradients can be difficult to stably compute, e.g. for ODEs and PDEs.

**Definition 22 (Gradent-free Stein operator)**

For distributions $P$ and $Q$ admitting sufficiently regular densities $p$ and $q$ on $\mathbb{R}^d$, we define the *gradient-free Stein operator*

$$(\mathcal{A}_{P,Q}\, g)(x) = \frac{q(x)}{p(x)} \left[ (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log q)(x) \right]$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

- The associated Stein discrepancy can be computed up to proportionality using $\tilde{p}(x) = Z \times p(x)$.
- The differentiability and distant dissipativity requirements now apply to $q$ (a degree of freedom) instead of $p$ (the posterior target).
- Sufficient conditions have been established on $q$ for convergence detection and control.

# Gradient-Free Kernel Stein Discrepancy

Gradients can be difficult to stably compute, e.g. for ODEs and PDEs.

## Definition 22 (Gradent-free Stein operator)

For distributions $P$ and $Q$ admitting sufficiently regular densities $p$ and $q$ on $\mathbb{R}^d$, we define the *gradient-free Stein operator*

$$(\mathcal{A}_{P,Q}\, g)(x) = \frac{q(x)}{p(x)} \left[ (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log q)(x) \right]$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

- The associated Stein discrepancy can be computed up to proportionality using $\tilde{p}(x) = Z \times p(x)$.
- The differentiability and distant dissipativity requirements now apply to $q$ (a degree of freedom) instead of $p$ (the posterior target).
- Sufficient conditions have been established on $q$ for convergence detection and control.

# Gradient-Free Kernel Stein Discrepancy

Gradients can be difficult to stably compute, e.g. for ODEs and PDEs.

## Definition 22 (Gradent-free Stein operator)

For distributions $P$ and $Q$ admitting sufficiently regular densities $p$ and $q$ on $\mathbb{R}^d$, we define the *gradient-free Stein operator*

$$(\mathcal{A}_{P,Q}\, g)(x) = \frac{q(x)}{p(x)} \left[ (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log q)(x) \right]$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

- The associated Stein discrepancy can be computed up to proportionality using $\tilde{p}(x) = Z \times p(x)$.
- The differentiability and distant dissipativity requirements now apply to $q$ (a degree of freedom) instead of $p$ (the posterior target).
- Sufficient conditions have been established on $q$ for convergence detection and control.

# Gradient-Free Kernel Stein Discrepancy

Gradients can be difficult to stably compute, e.g. for ODEs and PDEs.

## Definition 22 (Gradent-free Stein operator)

For distributions $P$ and $Q$ admitting sufficiently regular densities $p$ and $q$ on $\mathbb{R}^d$, we define the *gradient-free Stein operator*

$$(\mathcal{A}_{P,Q}\, g)(x) = \frac{q(x)}{p(x)} \left[ (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log q)(x) \right]$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

- The associated Stein discrepancy can be computed up to proportionality using $\tilde{p}(x) = Z \times p(x)$.
- The differentiability and distant dissipativity requirements now apply to $q$ (a degree of freedom) instead of $p$ (the posterior target).
- Sufficient conditions have been established on $q$ for convergence detection and control.

# Gradient-Free Kernel Stein Discrepancy

Gradients can be difficult to stably compute, e.g. for ODEs and PDEs.

## Definition 22 (Gradent-free Stein operator)

For distributions $P$ and $Q$ admitting sufficiently regular densities $p$ and $q$ on $\mathbb{R}^d$, we define the *gradient-free Stein operator*

$$(\mathcal{A}_{P,Q}\, g)(x) = \frac{q(x)}{p(x)} \left[ (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log q)(x) \right]$$

acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

- The associated Stein discrepancy can be computed up to proportionality using $\tilde{p}(x) = Z \times p(x)$.
- The differentiability and distant dissipativity requirements now apply to $q$ (a degree of freedom) instead of $p$ (the posterior target).
- Sufficient conditions have been established on $q$ for convergence detection and control.

# Gradient-Free Kernel Stein Discrepancy

Gradients can be difficult to stably compute, e.g. for ODEs and PDEs.

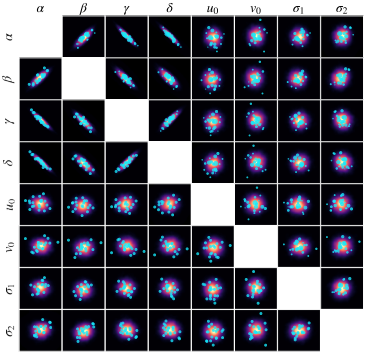
## Definition 22 (Gradent-free Stein operator)

For distributions $P$ and $Q$ admitting sufficiently regular densities $p$ and $q$ on $\mathbb{R}^d$, we define the *gradient-free Stein operator*

$$(\mathcal{A}_{P,Q}\, g)(x) = \frac{q(x)}{p(x)} \left[ (\nabla \cdot g)(x) + g(x) \cdot (\nabla \log q)(x) \right]$$

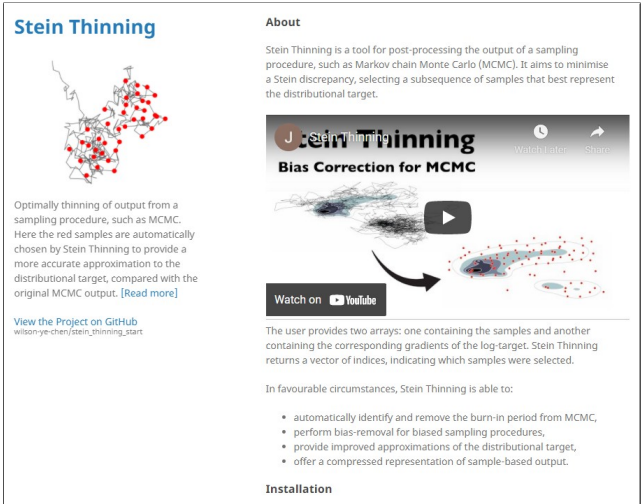
acting on differentiable vector field $g : \mathbb{R}^d \to \mathbb{R}^d$, where $x \in \mathbb{R}^d$.

- The associated Stein discrepancy can be computed up to proportionality using $\tilde{p}(x) = Z \times p(x)$.
- The differentiability and distant dissipativity requirements now apply to $q$ (a degree of freedom) instead of $p$ (the posterior target).
- Sufficient conditions have been established on $q$ for convergence detection and control.



# Software Development

To try out these algorithms (in Python, Matlab and R), visit `Stein-Thinning.org`



# Broader Context: Optimisation over $\mathcal{P}(\Theta)$

Going beyond optimisation in $\Theta$, we can consider optimisation in $\mathcal{P}(\Theta)$:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Variational Inference:** Ranganath et al. [2016], Hu et al. [2018], Fisher et al. [2021], ...

$$\min_{Q \in \mathcal{Q}} D_{\mathcal{H}(k_P),P}(Q), \qquad \text{(e.g.) } \mathcal{Q} = \{T_\# Q_0 : T \text{ a neural network}\}$$

  Avoids the requirement in VI that $T$ be a diffeomorphism (i.e. no need for normalising flows!).

- **Gradient Flow:** Korba et al. [2021]

$$\frac{\partial Q_t}{\partial t} + \mathrm{div}(Q_t v_{Q_t}) = 0, \qquad v_{Q_t} = -\nabla_{W_2} \mathcal{F}(Q_t), \qquad \mathcal{F}(Q) = \frac{1}{2} D_{\mathcal{H}(k_P),P}(Q)^2$$

  *not the same as SVGD [see Liu, 2017].

# Broader Context: Optimisation over $\mathcal{P}(\Theta)$

Going beyond optimisation in $\Theta$, we can consider optimisation in $\mathcal{P}(\Theta)$:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Variational Inference:** Ranganath et al. [2016], Hu et al. [2018], Fisher et al. [2021], ...

$$\min_{Q \in \mathcal{Q}} D_{\mathcal{H}(k_P),P}(Q), \qquad \text{(e.g.) } \mathcal{Q} = \{T_\# Q_0 : T \text{ a neural network}\}$$

  Avoids the requirement in VI that $T$ be a diffeomorphism (i.e. no need for normalising flows!).

- **Gradient Flow:** Korba et al. [2021]

$$\frac{\partial Q_t}{\partial t} + \operatorname{div}(Q_t v_{Q_t}) = 0, \qquad v_{Q_t} = -\nabla_{W_2} \mathcal{F}(Q_t), \qquad \mathcal{F}(Q) = \frac{1}{2} D_{\mathcal{H}(k_P),P}(Q)^2$$

  *not the same as SVGD [see Liu, 2017].

# Broader Context: Optimisation over $\mathcal{P}(\Theta)$

Going beyond optimisation in $\Theta$, we can consider optimisation in $\mathcal{P}(\Theta)$:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{\|f\|_{\mathcal{H}(k_P)} \leq 1} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Variational Inference:** Ranganath et al. [2016], Hu et al. [2018], Fisher et al. [2021], ...

$$\min_{Q \in \mathcal{Q}} D_{\mathcal{H}(k_P),P}(Q), \qquad \text{(e.g.) } \mathcal{Q} = \{T_\# Q_0 : T \text{ a neural network}\}$$

  Avoids the requirement in VI that $T$ be a diffeomorphism (i.e. no need for normalising flows!).

- **Gradient Flow:** Korba et al. [2021]

$$\frac{\partial Q_t}{\partial t} + \mathrm{div}(Q_t v_{Q_t}) = 0, \qquad v_{Q_t} = -\nabla_{W_2} \mathcal{F}(Q_t), \qquad \mathcal{F}(Q) = \frac{1}{2} D_{\mathcal{H}(k_P),P}(Q)^2$$

  *not the same as SVGD [see Liu, 2017].

# Broader Context: Sampling with Stein Discrepancies

For any Stein characterisation $(\mathcal{A}, \mathcal{F})$ we can consider an associated Stein discrepancy [Gorham and Mackey, 2015]:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{f \in \mathcal{F}} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Beyond Euclidean State Spaces:** Riemannian manifolds [Barp et al., 2022, Le et al., 2020], discrete spaces [Xu and Reinert, 2021], ...
- **Beyond Kernel Stein Sets:** bounded Lipschitz [Gorham and Mackey, 2015], neural network [Grathwohl et al., 2020], ...
- **Beyond the Canonical Stein Operator:** diffusion Stein operators [Gorham et al., 2019], ...
- **Scalable Stein Discrepancies:** random features [Huggins and Mackey, 2018], data sub-sampling [Gorham et al., 2020], slicing [Gong et al., 2020], ...

**The interaction between the sampling algorithms we have seen and the choice of Stein Discrepancy is not well-understood.**

# Broader Context: Sampling with Stein Discrepancies

For any Stein characterisation $(\mathcal{A}, \mathcal{F})$ we can consider an associated Stein discrepancy [Gorham and Mackey, 2015]:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{f \in \mathcal{F}} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Beyond Euclidean State Spaces:** Riemannian manifolds [Barp et al., 2022, Le et al., 2020], discrete spaces [Xu and Reinert, 2021], ...
- **Beyond Kernel Stein Sets:** bounded Lipschitz [Gorham and Mackey, 2015], neural network [Grathwohl et al., 2020], ...
- **Beyond the Canonical Stein Operator:** diffusion Stein operators [Gorham et al., 2019], ...
- **Scalable Stein Discrepancies:** random features [Huggins and Mackey, 2018], data sub-sampling [Gorham et al., 2020], slicing [Gong et al., 2020], ...

**The interaction between the sampling algorithms we have seen and the choice of Stein Discrepancy is not well-understood.**

# Broader Context: Sampling with Stein Discrepancies

For any Stein characterisation $(\mathcal{A}, \mathcal{F})$ we can consider an associated Stein discrepancy [Gorham and Mackey, 2015]:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{f \in \mathcal{F}} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Beyond Euclidean State Spaces:** Riemannian manifolds [Barp et al., 2022, Le et al., 2020], discrete spaces [Xu and Reinert, 2021], ...
- **Beyond Kernel Stein Sets:** bounded Lipschitz [Gorham and Mackey, 2015], neural network [Grathwohl et al., 2020], ...
- **Beyond the Canonical Stein Operator:** diffusion Stein operators [Gorham et al., 2019], ...
- **Scalable Stein Discrepancies:** random features [Huggins and Mackey, 2018], data sub-sampling [Gorham et al., 2020], slicing [Gong et al., 2020], ...

**The interaction between the sampling algorithms we have seen and the choice of Stein Discrepancy is not well-understood.**

# Broader Context: Sampling with Stein Discrepancies

For any Stein characterisation $(\mathcal{A}, \mathcal{F})$ we can consider an associated Stein discrepancy [Gorham and Mackey, 2015]:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{f \in \mathcal{F}} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Beyond Euclidean State Spaces:** Riemannian manifolds [Barp et al., 2022, Le et al., 2020], discrete spaces [Xu and Reinert, 2021], ...
- **Beyond Kernel Stein Sets:** bounded Lipschitz [Gorham and Mackey, 2015], neural network [Grathwohl et al., 2020], ...
- **Beyond the Canonical Stein Operator:** diffusion Stein operators [Gorham et al., 2019], ...
- **Scalable Stein Discrepancies:** random features [Huggins and Mackey, 2018], data sub-sampling [Gorham et al., 2020], slicing [Gong et al., 2020], ...

**The interaction between the sampling algorithms we have seen and the choice of Stein Discrepancy is not well-understood.**

# Broader Context: Sampling with Stein Discrepancies

For any Stein characterisation $(\mathcal{A}, \mathcal{F})$ we can consider an associated Stein discrepancy [Gorham and Mackey, 2015]:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{f \in \mathcal{F}} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Beyond Euclidean State Spaces:** Riemannian manifolds [Barp et al., 2022, Le et al., 2020], discrete spaces [Xu and Reinert, 2021], ...
- **Beyond Kernel Stein Sets:** bounded Lipschitz [Gorham and Mackey, 2015], neural network [Grathwohl et al., 2020], ...
- **Beyond the Canonical Stein Operator:** diffusion Stein operators [Gorham et al., 2019], ...
- **Scalable Stein Discrepancies:** random features [Huggins and Mackey, 2018], data sub-sampling [Gorham et al., 2020], slicing [Gong et al., 2020], ...

**The interaction between the sampling algorithms we have seen and the choice of Stein Discrepancy is not well-understood.**

# Broader Context: Sampling with Stein Discrepancies

For any Stein characterisation $(\mathcal{A}, \mathcal{F})$ we can consider an associated Stein discrepancy [Gorham and Mackey, 2015]:

$$D_{\mathcal{H}(k_P),P}(Q) := \sup_{f \in \mathcal{F}} |\mathbb{E}_{\vartheta \sim Q}[f(\vartheta)]|$$

- **Beyond Euclidean State Spaces:** Riemannian manifolds [Barp et al., 2022, Le et al., 2020], discrete spaces [Xu and Reinert, 2021], ...
- **Beyond Kernel Stein Sets:** bounded Lipschitz [Gorham and Mackey, 2015], neural network [Grathwohl et al., 2020], ...
- **Beyond the Canonical Stein Operator:** diffusion Stein operators [Gorham et al., 2019], ...
- **Scalable Stein Discrepancies:** random features [Huggins and Mackey, 2018], data sub-sampling [Gorham et al., 2020], slicing [Gong et al., 2020], ...

**The interaction between the sampling algorithms we have seen and the choice of Stein Discrepancy is not well-understood.**

# Broader Context: Alternatives to Direct Minimisation of Stein Discrepancy

- **Stein Variational Gradient Descent:** Liu and Wang [2016], Liu [2017], Liu and Zhu [2018], Detommaso et al. [2018], ...
- **MCMC with Stein Control Variates:** Assaraf and Caffarel [1999], Mira et al. [2013], CJO et al. [2017], Belomestny et al. [2017], South et al. [2022], ...

$$\text{Given a QoI } f, \text{ seek } (u, c) \text{ such that } c + \frac{\nabla \cdot (p \nabla u)}{p} = f. \qquad \text{Then } c = \mathbb{E}_{\vartheta \sim P}[f(\vartheta)].$$

In practice, an approximate solution $u$ gives rise to a control variate $v = \nabla \cdot (p \nabla u)/p$ for use in MCMC.

For more, please see the recent review paper of Anastasiou et al. [2022].

# Broader Context: Alternatives to Direct Minimisation of Stein Discrepancy

- **Stein Variational Gradient Descent:** Liu and Wang [2016], Liu [2017], Liu and Zhu [2018], Detommaso et al. [2018], ...
- **MCMC with Stein Control Variates:** Assaraf and Caffarel [1999], Mira et al. [2013], CJO et al. [2017], Belomestny et al. [2017], South et al. [2022], ...

$$\text{Given a QoI } f, \text{ seek } (u, c) \text{ such that } c + \frac{\nabla \cdot (p \nabla u)}{p} = f. \qquad \text{Then } c = \mathbb{E}_{\vartheta \sim P}[f(\vartheta)].$$

In practice, an approximate solution $u$ gives rise to a control variate $v = \nabla \cdot (p \nabla u)/p$ for use in MCMC.

For more, please see the recent review paper of Anastasiou et al. [2022].

# Broader Context: Alternatives to Direct Minimisation of Stein Discrepancy

- **Stein Variational Gradient Descent:** Liu and Wang [2016], Liu [2017], Liu and Zhu [2018], Detommaso et al. [2018], ...
- **MCMC with Stein Control Variates:** Assaraf and Caffarel [1999], Mira et al. [2013], CJO et al. [2017], Belomestny et al. [2017], South et al. [2022], ...

$$\text{Given a QoI } f, \text{ seek } (u, c) \text{ such that } c + \frac{\nabla \cdot (p \nabla u)}{p} = f. \qquad \text{Then } c = \mathbb{E}_{\vartheta \sim P}[f(\vartheta)].$$

In practice, an approximate solution $u$ gives rise to a control variate $v = \nabla \cdot (p \nabla u)/p$ for use in MCMC.

For more, please see the recent review paper of Anastasiou et al. [2022].

## Thank you for your attention!

### Collaborators:

Alessandro Barp, François-Xavier Briol, Lawrence Carin, Wilson Chen, Nicolas Chopin, Jon Cockayne, Chris Drovandi, Andrew Duncan, Martin Ehler, Matthew Fisher, Mark Girolami, Jackson Gorham, Manuel Graef, Matt Graham, Toni Karvonen, Jeremias Knoblauch, Lester Mackey, Takuo Matsubara, Antonietta Mira, Chris Nemeth, Steve Niederer, Tui Nolan, Emilio Porcu, Dennis Prangle, Marina Riabiz, Simo Särkkä, Shijing Shi, Leah South, Pawel Swietach, Onur Teymur

### References:


A. Anastasiou et al. Stein's method meets statistics: A review of some recent developments. *Statistical Science*, 2022. To appear.

M. Arbel, A. Korba, A. Salim, and A. Gretton. Maximum mean discrepancy gradient flow. *NeurIPS*, 2019.

R. Assaraf and M. Caffarel. Zero-variance principle for Monte Carlo algorithms. *Physical Review Letters*, 83(23):4682, 1999.

A. Barp, CJO, E. Porcu, and M. Girolami. A Riemann–Stein kernel method. *Bernoulli*, 2022. To appear.

D. Belomestny, L. Iosipoi, and N. Zhivotovskiy. Variance reduction via empirical variance minimization: convergence and complexity. *arXiv:1712.04667*, 2017.

CJO, M. Girolami, and N. Chopin. Control functionals for Monte Carlo integration. *JRSSB*, 79(3):695–718, 2017.

G. Detommaso, T. Cui, Y. Marzouk, R. Scheichl, and A. Spantini. A Stein variational Newton method. In *NeurIPS*, 2018.

M. Ehler, M. Gräf, and C. J. Oates. Optimal Monte Carlo integration on closed manifolds. *Statistics and Computing*, 29(6): 1203–1214, 2019.

M. A. Fisher, T. Nolan, M. M. Graham, D. Prangle, and CJO. Measure transport with kernel Stein discrepancy. *AISTATS*, 2021.

W. Gong, Y. Li, and J. M. Hernández-Lobato. Sliced kernelized Stein discrepancy. In *ICLR*, 2020.

J. Gorham and L. Mackey. Measuring sample quality with Stein's method. In *NeurIPS*, 2015.

J. Gorham and L. Mackey. Measuring sample quality with kernels. In *ICML*, 2017.

J. Gorham, A. B. Duncan, S. J. Vollmer, and L. Mackey. Measuring sample quality with diffusions. *AoAP*, 29(5):2884–2928, 2019.

J. Gorham, A. Raj, and L. Mackey. Stochastic Stein discrepancies. In *NeurIPS*, 2020.

M. Gräf, D. Potts, and G. Steidl. Quadrature errors, discrepancies, and their relations to halftoning on the torus and the sphere. *SIAM Journal on Scientific Computing*, 34(5):A2760–A2791, 2012.

W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, and R. Zemel. Learning the Stein discrepancy for training and evaluating energy-based models without sampling. In *ICML*, 2020.

L. Hodgkinson, R. Salomone, and F. Roosta. The reproducing Stein kernel approach for post-hoc corrected sampling. *arXiv:2001.09266*, 2020.

T. Hu, Z. Chen, H. Sun, J. Bai, M. Ye, and G. Cheng. Stein neural sampler. *arXiv preprint arXiv:1810.03545*, 2018.

J. Huggins and L. Mackey. Random feature Stein discrepancies. In *NeurIPS*, 2018.

T. Karvonen and S. Särkkä. Fully symmetric kernel quadrature. *SIAM Journal on Scientific Computing*, 40(2):A697–A720, 2018.

T. Karvonen, S. Särkkä, and CJO. Symmetry exploits for Bayesian cubature methods. *Statistics and Computing*, 29(6): 1231–1248, 2019.

A. Korba, P.-C. Aubin-Frankowski, S. Majewski, and P. Ablin. Kernel Stein discrepancy descent. In *ICML*, 2021.

H. Le, A. Lewis, K. Bharath, and C. Fallaize. A diffusion approach to Stein's method on Riemannian manifolds. *arXiv:2003.11497*, 2020.

C. Liu and J. Zhu. Riemannian Stein variational gradient descent for Bayesian inference. In *AAAI Conference on AI*, number 1, 2018.

Q. Liu. Stein Variational Gradient Descent as Gradient Flow. In *NeurIPS*, 2017.

Q. Liu and J. D. Lee. Black-box importance sampling. In *AISTATS*, 2017.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *NeurIPS*, 2016.

A. Mira, R. Solgi, and D. Imparato. Zero variance Markov chain Monte Carlo for Bayesian estimators. *Statistics and Computing*, 23(5):653–662, 2013.

R. Ranganath, D. Tran, J. Altosaar, and D. Blei. Operator variational inference. In *NeurIPS*, volume 29, 2016.

M. Riabiz, W. Chen, J. Cockayne, P. Swietach, S. A. Niederer, L. Mackey, and CJO. Optimal thinning of MCMC output. *JRSSB*, 2022.

C.-J. Simon-Gabriel, A. Barp, and L. Mackey. Metrizing weak convergence with maximum mean discrepancies. *arXiv:2006.09268*, 2020.

L. F. South, T. Karvonen, C. Nemeth, M. Girolami, and CJO. Semi-exact control functionals from Sard's method. *Biometrika*, 2022.

B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.

C. Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, pages 583–602. University of California Press, 1972.

O. Teymur, J. Gorham, M. Riabiz, and CJO. Optimal quantisation of probability measures using maximum mean discrepancy. In *AISTATS*, 2021.

H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *Journal of Approximation Theory*, 93(2):258–272, 1998.

L. K. Wenliang and H. Kanagawa. Blindness of score-based methods to isolated components and mixing proportions. *arXiv preprint arXiv:2008.10087*, 2020.

W. Xu and G. Reinert. A Stein goodness-of-test for exponential random graph models. In *AISTATS*, 2021.