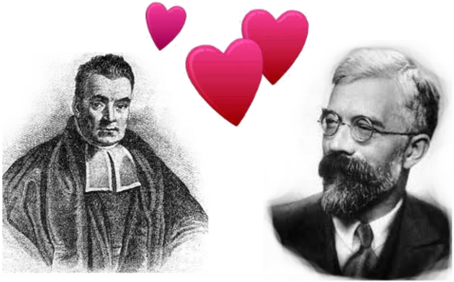
Deterministic Sampling Methods



David Duvenaud

Cambridge University
Computational and Biological Learning Lab

August 7, 2013

Outline

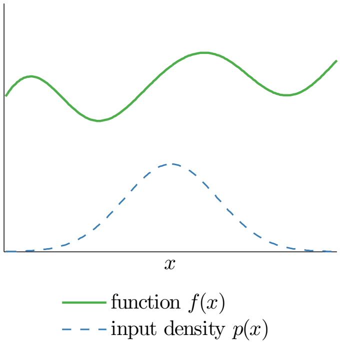
- Kernel Herding
- Bayesian Quadrature
- Unifying Results
- Demos

The Quadrature Problem

- We want to estimate an integral

$$Z = \int f(x)p(x)dx$$

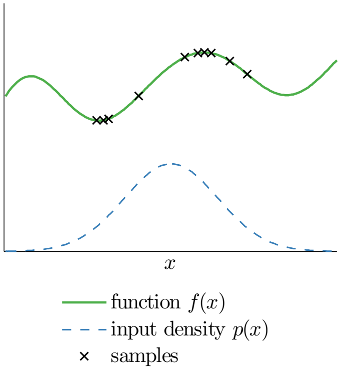
- Most computational problems in Bayesian inference correspond to integrals:
  - Expectations
  - Marginal distributions
  - Integrating out nuisance parameters
  - Normalization constants



Sampling Methods

- Monte Carlo methods:
  Sample from $p(x)$, take empirical mean:

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

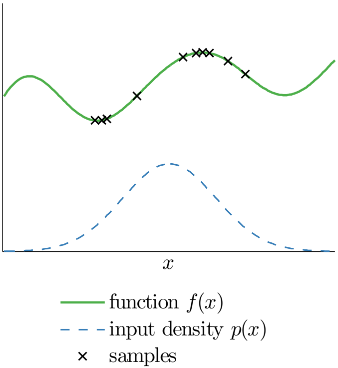


Sampling Methods

- Monte Carlo methods:
  Sample from $p(x)$, take empirical mean:

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

- Possibly sub-optimal for two reasons:

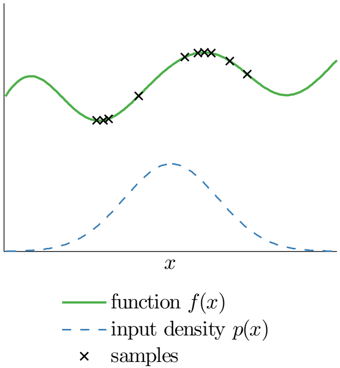


Sampling Methods

- Monte Carlo methods:
  Sample from $p(x)$, take empirical mean:

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

- Possibly sub-optimal for two reasons:
  - Random bunching up

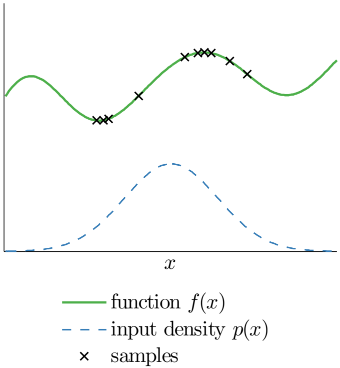


Sampling Methods

- Monte Carlo methods:
  Sample from $p(x)$, take empirical mean:

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

- Possibly sub-optimal for two reasons:
  - Random bunching up
  - Often, nearby function values will be similar

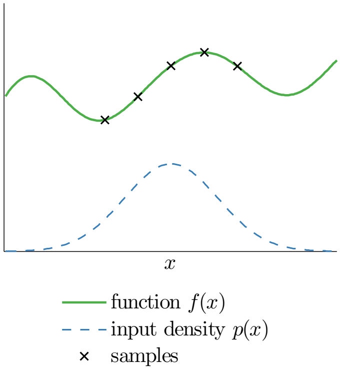


Sampling Methods

- Monte Carlo methods:
  Sample from $p(x)$, take empirical mean:

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

- Possibly sub-optimal for two reasons:
  - Random bunching up
  - Often, nearby function values will be similar
- Quasi-Monte Carlo methods spread out samples to achieve faster convergence.



Kernel Herding [Welling et. al., 2009, Chen et. al., 2010]

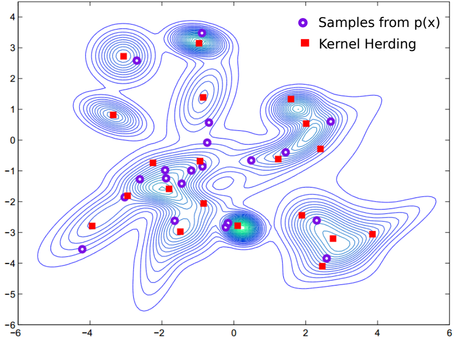
- A sequential procedure for choosing sample locations, depending on previous locations.

Kernel Herding [Welling et. al., 2009, Chen et. al., 2010]

- A sequential procedure for choosing sample locations, depending on previous locations.
- Keeps estimate rule $\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$

Kernel Herding [Welling et. al., 2009, Chen et. al., 2010]

- A sequential procedure for choosing sample locations, depending on previous locations.
- Keeps estimate rule $\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$
- Almost $\mathcal{O}(1/N)$ convergence instead of $\mathcal{O}(1/\sqrt{N})$ typical of random sampling, by spreading out samples.



Kernel Herding Objective

KH was found to minimize Maximum Mean Discrepancy:

$$\mathrm{MMD}_{\mathcal{H}}(p,q) = \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}}=1}} \left| \int f(x)p(x)dx - \int f(x)q(x)dx \right|$$

Kernel Herding Objective

KH was found to minimize Maximum Mean Discrepancy:

$$\mathrm{MMD}_{\mathcal{H}}\left(p, q\right) = \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} = 1}} \left| \int f(x)p(x)dx - \int f(x)q(x)dx \right|$$

In KH, $p(x)$ is true distribution, and $q(x)$ is a set of point masses at sample locations $\{x_1, \ldots, x_N\}$:

$$\epsilon_{KH}\left(\{x_1, \ldots, x_N\}\right) = \mathrm{MMD}_{\mathcal{H}}\left(p, \underbrace{\frac{1}{N}\sum_{n=1}^{N} \delta_{x_n}}_{q(x)} \right)$$

Kernel Herding

- Assuming function is in a Reproducing Kernel Hilbert Space defined by $k(\cdot, \cdot)$, MMD has closed form.

Kernel Herding

- Assuming function is in a Reproducing Kernel Hilbert Space defined by $k(\cdot, \cdot)$, MMD has closed form.
- When sequentially minimizing MMD, new point is added at:

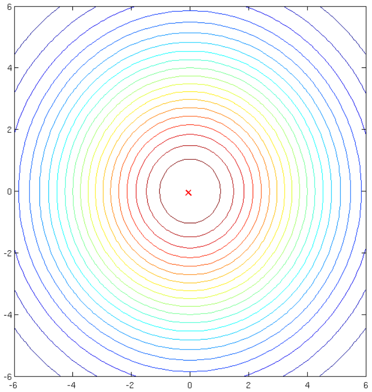
$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$

Kernel Herding in Action

New point is added at:

$$x_{N+1} =$$

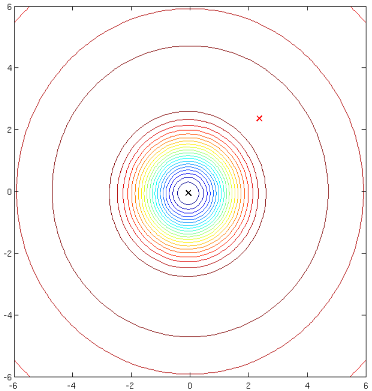
$$\underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

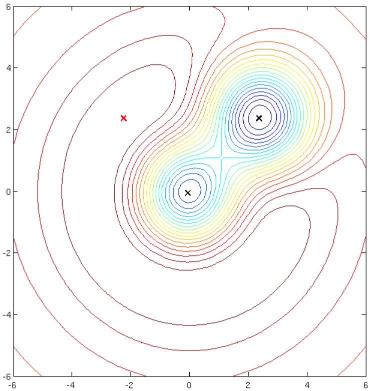
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

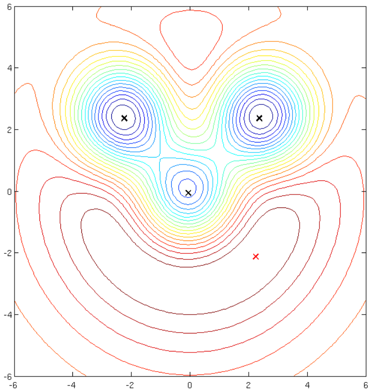
$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

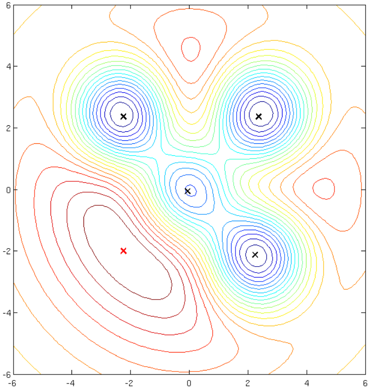
$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

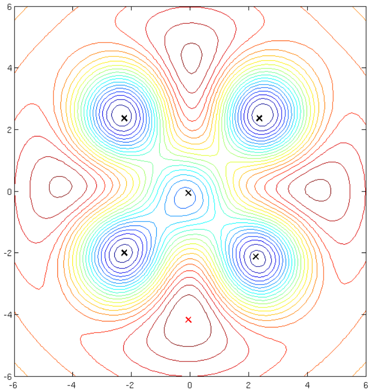
$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

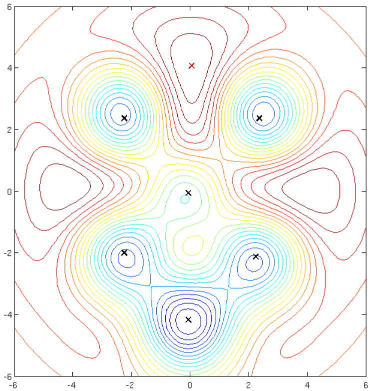
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$

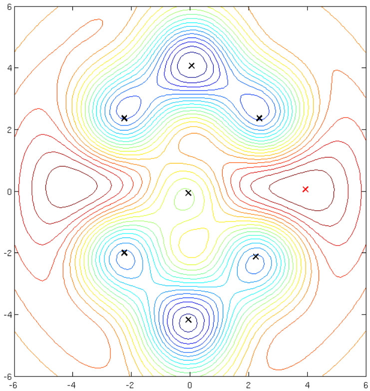


Kernel Herding in Action

New point is added at:

$$x_{N+1} =$$

$$\underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$

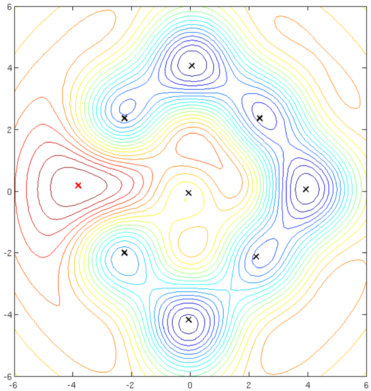


Kernel Herding in Action

New point is added at:

$$x_{N+1} =$$

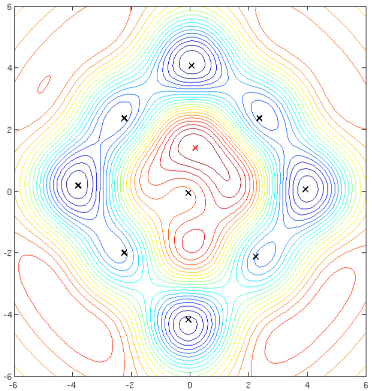
$$\underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

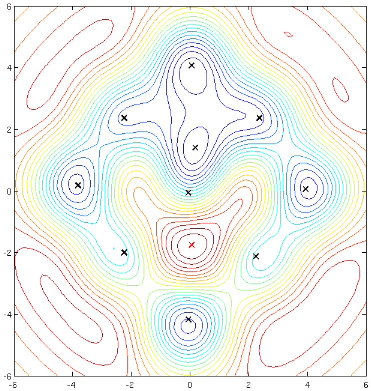
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

$$x_{N+1} =$$
$$\underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$

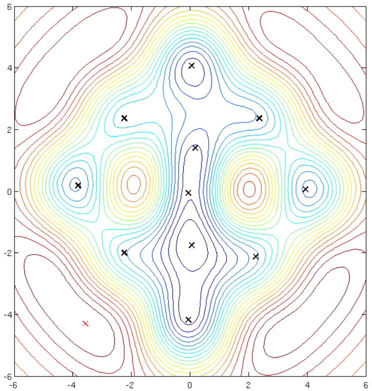


Kernel Herding in Action

New point is added at:

$$x_{N+1} =$$

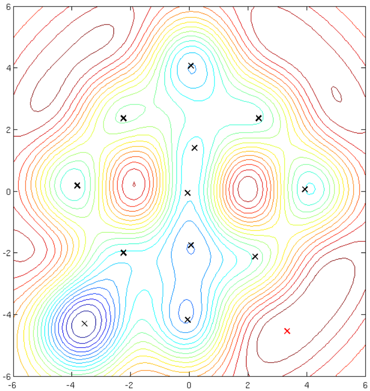
$$\operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

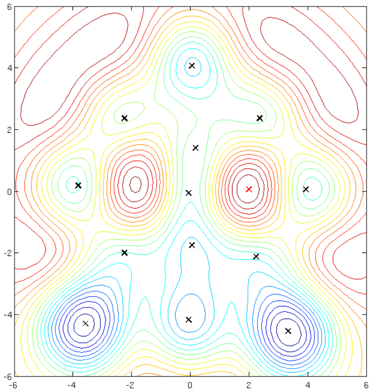
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

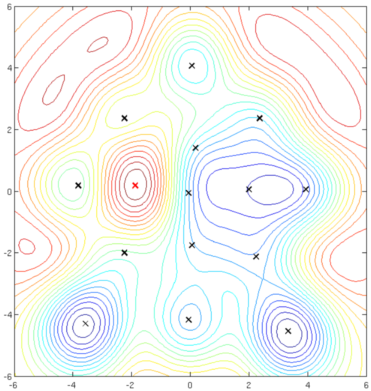
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

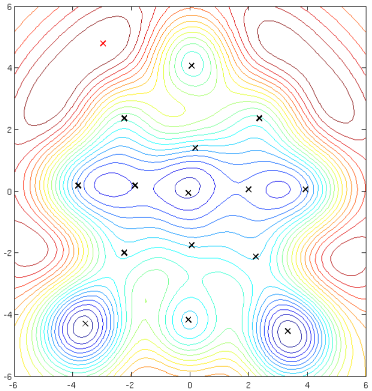
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

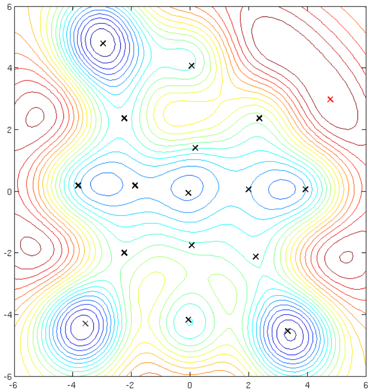
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

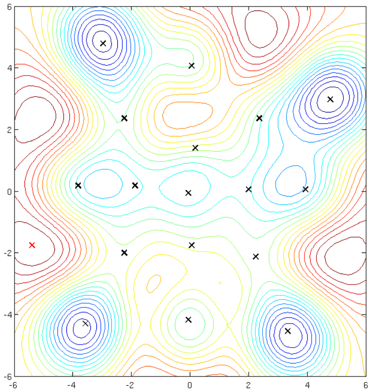
$$x_{N+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

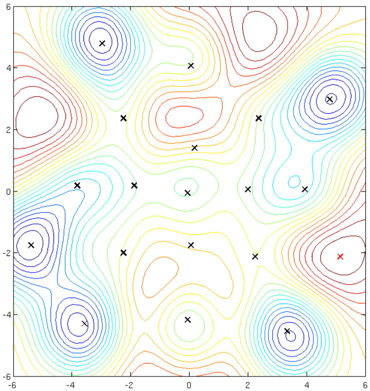
$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

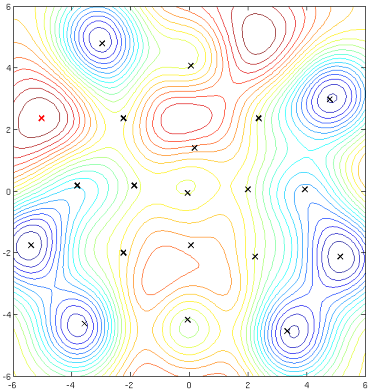
$$x_{N+1} =$$
$$\operatorname*{argmax}_{x \in \mathcal{X}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding in Action

New point is added at:

$$x_{N+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} \left[2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]$$



Kernel Herding Summary

- A sequential sampling method which minimizes a worst-case divergence, given that $f(x)$ belongs to a given RKHS.
- Like Monte Carlo, weights all samples $f(x_s)$ equally when estimating Z:

$$\hat{Z} = \sum_{i=1}^{N} \frac{1}{N} f(x_i)$$

Kernel Herding Summary

- A sequential sampling method which minimizes a worst-case divergence, given that $f(x)$ belongs to a given RKHS.
- Like Monte Carlo, weights all samples $f(x_s)$ equally when estimating Z:

$$\hat{Z} = \sum_{i=1}^{N} \frac{1}{N} f(x_i)$$

- What if we allowed different weights?

Kernel Herding Summary

- A sequential sampling method which minimizes a worst-case divergence, given that $f(x)$ belongs to a given RKHS.
- Like Monte Carlo, weights all samples $f(x_s)$ equally when estimating Z:

$$\hat{Z} = \sum_{i=1}^{N} \frac{1}{N} f(x_i)$$

- What if we allowed different weights?
- [Bach et. al. 2012] looked at weighted herding strategies, showed improvement in convergence rates.

Kernel Herding Summary

- A sequential sampling method which minimizes a worst-case divergence, given that $f(x)$ belongs to a given RKHS.
- Like Monte Carlo, weights all samples $f(x_s)$ equally when estimating Z:

$$\hat{Z} = \sum_{i=1}^{N} \frac{1}{N} f(x_i)$$

- What if we allowed different weights?
- [Bach et. al. 2012] looked at weighted herding strategies, showed improvement in convergence rates.

Can we reason about the optimal weighting strategy?

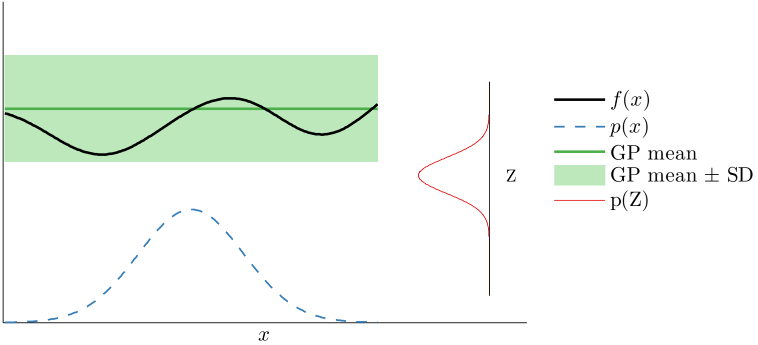
Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

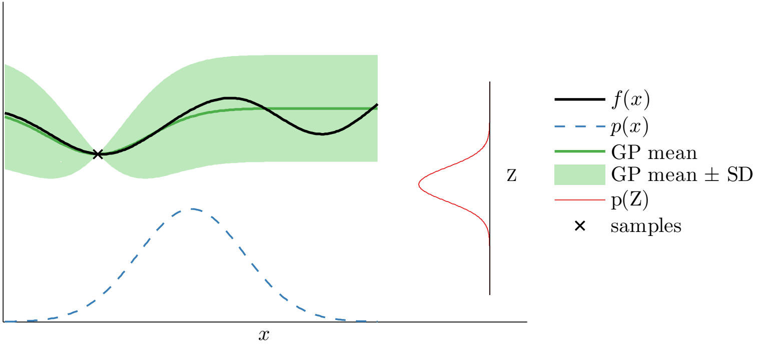
- Places a GP prior on f, defined by $k(\cdot, \cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

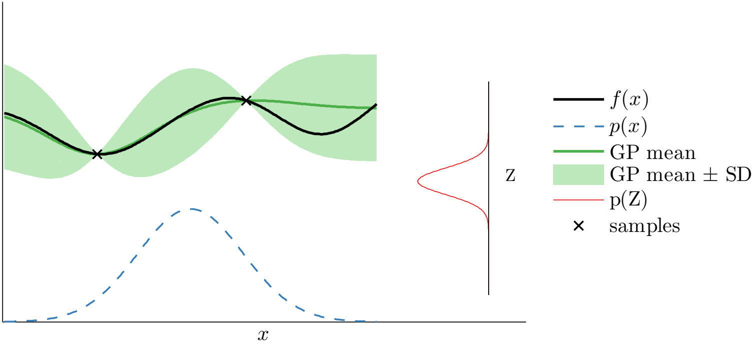
- Places a GP prior on f, defined by $k(\cdot, \cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

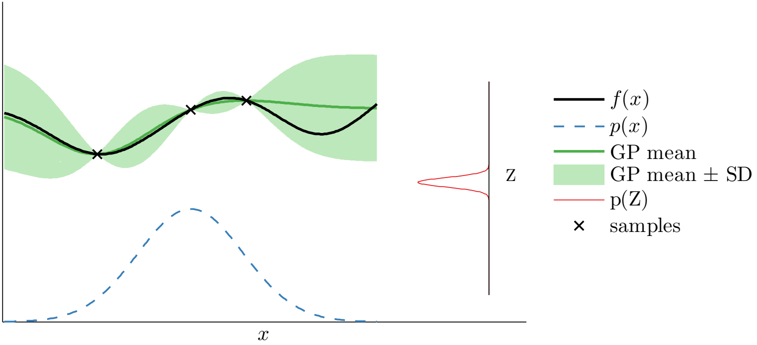
- Places a GP prior on f, defined by $k(\cdot, \cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

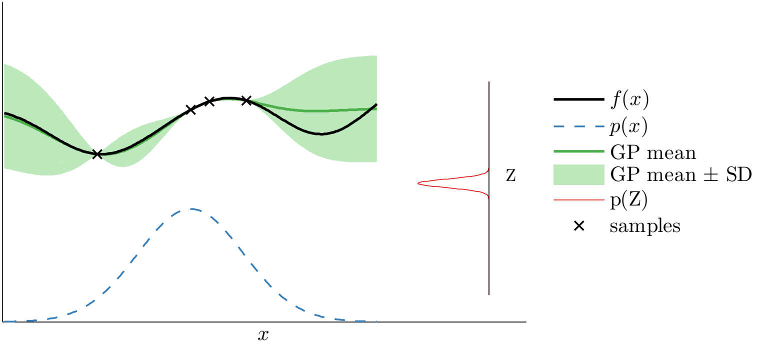
- Places a GP prior on f, defined by $k(\cdot, \cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

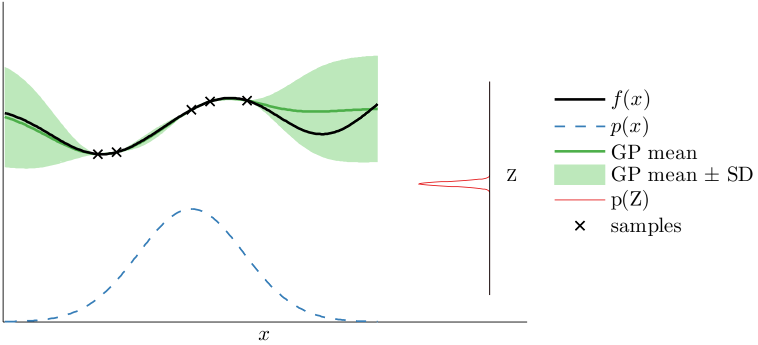
- Places a GP prior on f, defined by $k(\cdot, \cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

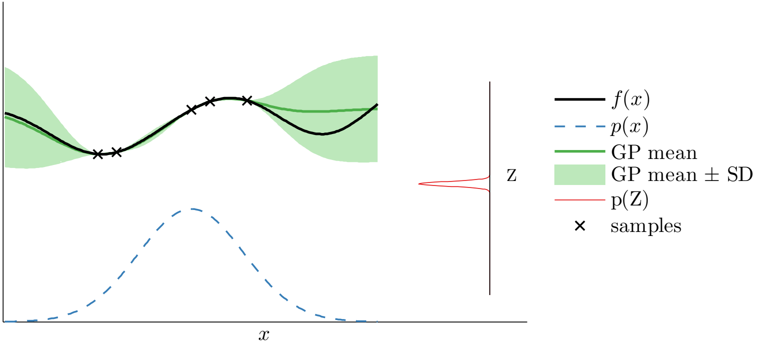
- Places a GP prior on f, defined by $k(\cdot,\cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)

[O'Hagan 1987, Diaconis 1988, Rasmussen & Ghahramani 2003]

- Places a GP prior on f, defined by $k(\cdot, \cdot)$ and a mean function.
- Posterior over f implies posterior over Z.



- Can choose samples however we want.

Bayesian Quadrature Estimator

Posterior over Z has mean linear in $f(x_s)$:

$$\mathbb{E}_{\text{GP}}\left[Z|f(x_s)\right] = \sum_{i=1}^{N} w_{BQ}^{(i)} f(x_i)$$

where

$$w_{BQ} = z^T K^{-1} \qquad \text{and} \qquad z_n = \int k(x, x_n) p(x) dx$$

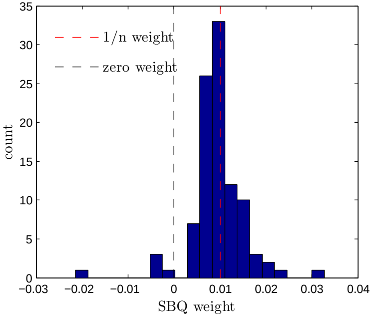
Bayesian Quadrature Estimator

Posterior over Z has mean linear in $f(x_s)$:

$$\mathbb{E}_{\mathrm{GP}}\left[Z|f(x_s)\right] = \sum_{i=1}^{N} w_{BQ}^{(i)} f(x_i)$$

where

$$w_{BQ} = z^T K^{-1} \qquad \text{and} \qquad z_n = \int k(x, x_n) p(x) dx$$



How to select samples?

How to select samples?

- Natural to minimize the posterior variance of Z:

$$\mathbb{V}\left[Z|f(x_s)\right] = \int\int k(x, x')p(x)p(x')dxdx' - z^T K^{-1} z$$

$$\text{where} \qquad z_n = \int k(x, x_n)p(x)dx$$

How to select samples?

- Natural to minimize the posterior variance of Z:

$$\mathbb{V}\left[Z|f(x_s)\right] = \iint k(x, x')p(x)p(x')dxdx' - z^T K^{-1} z$$
$$\text{where} \qquad z_n = \int k(x, x_n)p(x)dx$$

- Favours samples in regions where $p(x)$ is high, but where covariance with other sample locations is low. Similar flavour to herding objective.

How to select samples?

- Natural to minimize the posterior variance of Z:

$$\mathbb{V}\left[Z|f(x_s)\right] = \int\int k(x,x')p(x)p(x')dxdx' - z^T K^{-1} z$$

$$\text{where} \qquad z_n = \int k(x,x_n)p(x)dx$$

- Favours samples in regions where $p(x)$ is high, but where covariance with other sample locations is low. Similar flavour to herding objective.
- Does not depend on function values

How to select samples?

- Natural to minimize the posterior variance of Z:

$$\mathbb{V}\left[Z|f(x_s)\right] = \int\int k(x,x')p(x)p(x')dxdx' - z^T K^{-1} z$$
$$\text{where} \qquad z_n = \int k(x,x_n)p(x)dx$$

- Favours samples in regions where $p(x)$ is high, but where covariance with other sample locations is low. Similar flavour to herding objective.
- Does not depend on function values
- Can choose samples sequentially: Sequential Bayesian Quadrature.

Relating Objectives

KH and BQ have completely different motivations:

- KH minimizes a worst-case bound
- BQ minimizes a posterior variance

Is there any correspondence?

Relating Objectives

KH and BQ have completely different motivations:

- KH minimizes a worst-case bound
- BQ minimizes a posterior variance

Is there any correspondence?

First Main Result

$$\mathbb{V}\left[Z|f(x_s)\right] = \mathrm{MMD}^2(p, q_{\mathrm{BQ}})$$

Where

$$q_{\mathrm{BQ}}(x) = \sum_{n=1}^{N} w_{\mathrm{BQ}}^{(n)} \delta_{x_n}(x)$$

Relating Objectives

KH and BQ have completely different motivations:

- KH minimizes a worst-case bound
- BQ minimizes a posterior variance

Is there any correspondence?

First Main Result

$$\mathbb{V}\left[Z|f(x_s)\right] = \mathrm{MMD}^2(p, q_{\mathrm{BQ}})$$

Where

$$q_{\mathrm{BQ}}(x) = \sum_{n=1}^{N} w_{\mathrm{BQ}}^{(n)} \delta_{x_n}(x)$$

BQ is minimizing KH objective

Performance

- KH and BQ are minimizing the same objective, but BQ has freedom to choose weights.

Performance

- KH and BQ are minimizing the same objective, but BQ has freedom to choose weights.
- How does this affect performance?

Performance

- KH and BQ are minimizing the same objective, but BQ has freedom to choose weights.
- How does this affect performance?

Second Main Result

BQ estimator is the optimal weighting strategy:

$$\mathbb{V}\left[Z|f(x_s)\right] = \inf_{w\in\mathbb{R}^N} \sup_{\substack{f\in\mathcal{H} \\ \|f\|_{\mathcal{H}}\mathcal{H}=1}} \left| \int f(x)p(x)dx - \sum_{n=1}^{N} w_n f(x_n) \right|^2$$

Performance

- KH and BQ are minimizing the same objective, but BQ has freedom to choose weights.
- How does this affect performance?

Second Main Result

BQ estimator is the optimal weighting strategy:

$$\mathbb{V}\left[Z|f(x_s)\right] = \inf_{w \in \mathbb{R}^N} \sup_{\substack{f \in \mathcal{H} \\ \|f\|_{\mathcal{H}} \mathcal{H} = 1}} \left| \int f(x)p(x)dx - \sum_{n=1}^{N} w_n f(x_n) \right|^2$$

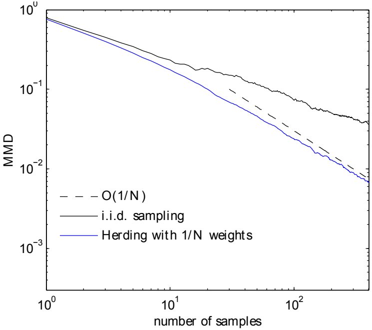
$\mathbb{V}\left[Z|f(x_s)\right]$ has two interpretations:

- Bayesian: posterior variance of Z under a GP prior.
- Frequentist: tight bound on estimation error of Z.

Rates of Convergence

What is rate of convergence of BQ?

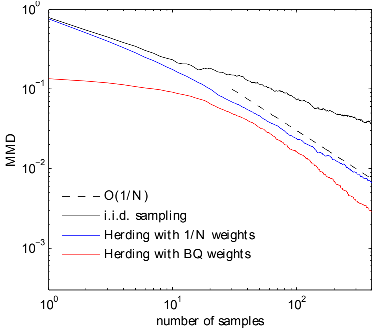
Expected Variance / MMD



Rates of Convergence

What is rate of convergence of BQ?

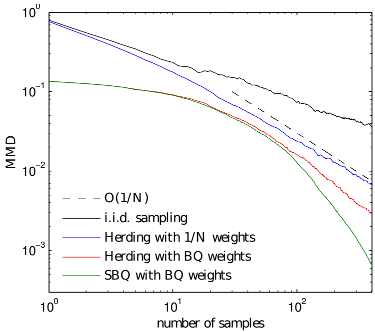
Expected Variance / MMD



Rates of Convergence

What is rate of convergence of BQ?

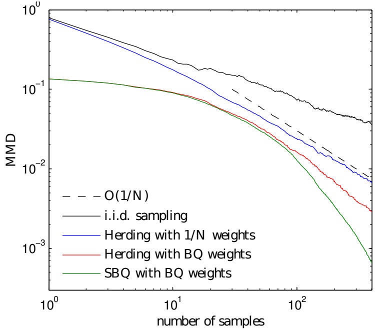
Expected Variance / MMD



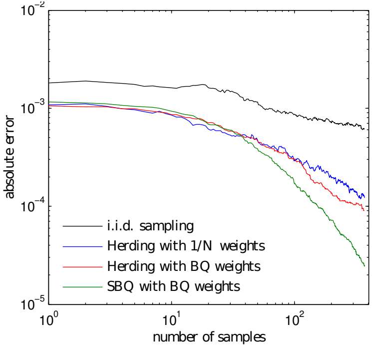
Rates of Convergence

What is rate of convergence of BQ?

Expected Variance / MMD



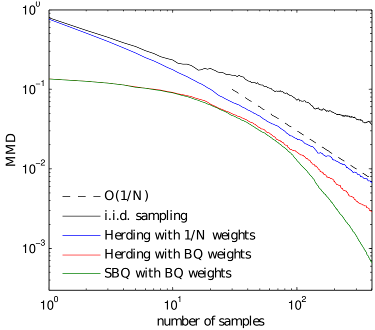
Empirical Rates in RKHS



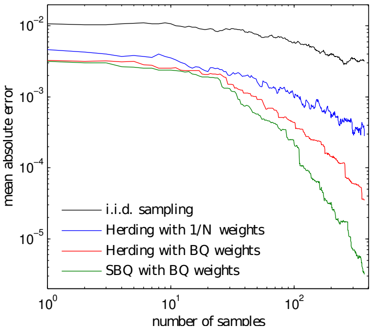
Rates of Convergence

What is rate of convergence of BQ?

Expected Variance / MMD



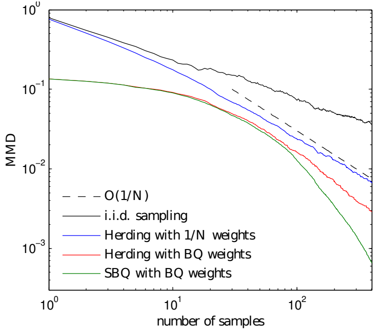
Empirical Rates out of RKHS



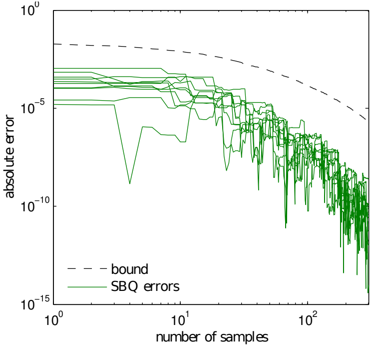
Rates of Convergence

What is rate of convergence of BQ?

Expected Variance / MMD



Bound on Bayesian Error



Summary

- Posterior variance of Z under GP prior is equivalent to Maximum Mean Discrepancy.

Summary

- Posterior variance of Z under GP prior is equivalent to Maximum Mean Discrepancy.
- RKHS assumption gives a tight, closed-form upper bound on Bayesian error.

Summary

- Posterior variance of Z under GP prior is equivalent to Maximum Mean Discrepancy.
- RKHS assumption gives a tight, closed-form upper bound on Bayesian error.
- BQ has very fast, but unknown convergence rate.

Summary

- Posterior variance of Z under GP prior is equivalent to Maximum Mean Discrepancy.
- RKHS assumption gives a tight, closed-form upper bound on Bayesian error.
- BQ has very fast, but unknown convergence rate.
- The optimal weighted herding strategy is Bayesian quadrature.

Summary

- Posterior variance of Z under GP prior is equivalent to Maximum Mean Discrepancy.
- RKHS assumption gives a tight, closed-form upper bound on Bayesian error.
- BQ has very fast, but unknown convergence rate.
- The optimal weighted herding strategy is Bayesian quadrature.
- Joint work with Ferenc Huzsar