Bayesian Interpretations of RKHS Embedding Methods

David Duvenaud

Cambridge University
Computational and Biological Learning Lab

December 8, 2012
Outline

- Optimally-weighted Herding is Bayesian Quadrature
  - Kernel Herding
  - Bayesian Quadrature
  - Unifying Results
  - Demos

- Frequentist Methods, Bayesian Takeaways
  - Kernel Herding
  - Mean Embeddings
  - Kernel Two-sample Test
  - Hilbert-Schmidt Independence Criterion
  - Determinantal Point Processes
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:

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

![Graph showing function $f(x)$, input density $p(x)$, and samples.](image)
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

\[\text{function } f(x)\]
\[\text{input density } p(x)\]
\[\times \text{ samples}\]
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 $O(1/N)$ convergence instead of $O(1/\sqrt{N})$ typical of random sampling, by spreading out samples.
Kernel Herding Objective

KH was found to minimize Maximum Mean Discrepancy:

\[ \text{MMD}_\mathcal{H}(p, q) = \sup_{f \in \mathcal{H}, \| f \|_{\mathcal{H}} = 1} \left| \int f(x)p(x)dx - \int f(x)q(x)dx \right| \]
Kernel Herding Objective

KH was found to minimize Maximum Mean Discrepancy:

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

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

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

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

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) - 2 \int k(x, x') p(x') dx' \right] \]
Kernel Herding in Action

New point is added at:

\[
x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right]
\]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in X} \left[ 2\int k(x, x') p(x') \, dx' \right. \]

\[ \left. - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[
x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' \right]
\[
- \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right]
\]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in X} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' \right. \]

\[ - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \]
Kernel Herding in Action

New point is added at:

\[
x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right]
\]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in X} \left[ 2 \int k(x, x') p(x') dx' \right. \]

\[ - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in X} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N + 1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding in Action

New point is added at:

\[ x_{N+1} = \arg\max_{x \in X} \left[ 2 \int k(x, x') p(x') dx' - \frac{1}{N+1} \sum_{m=1}^{N} k(x, x_m) \right] \]
Kernel Herding 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)

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


- 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)


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

\[
\begin{align*}
Z & \quad x \\
& \quad f(x) \\
& \quad p(x) \\
& \quad \text{GP mean} \\
& \quad \text{GP mean ± SD} \\
& \quad p(Z) \\
& \quad \times \quad \text{samples}
\end{align*}
\]
Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)


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

![Diagram showing Bayesian Quadrature](image)
Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)


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

![Graph showing Bayesian Quadrature](image)
Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)


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

\[
Z \propto p(x) f(x) \quad \text{GP mean} \quad \text{GP mean ± SD} \quad p(Z) \quad \text{samples}
\]
Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)


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

\[
\frac{p(Z)}{x} f(x) = \text{GP mean} \pm \text{SD}
\]

\[\text{samples}\]
Bayesian Quadrature (a.k.a. Bayesian Monte Carlo)


- 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 $f(x_s)$:

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

where

$$w_{BQ} = z^T K^{-1}$$

and

$$z_n = \int k(x, x_n) p(x) dx$$
Bayesian Quadrature Estimator

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

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

where

$$w_{\text{BQ}} = z^T K^{-1}$$

and

$$z_n = \int k(x, x_n) p(x) dx$$
How to select samples?

Natural to minimize the posterior variance of $Z$:

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

where $z_n = \int k(x, x_n) p(x) dx$.

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.
How to select samples?

- Natural to minimize the posterior variance of $Z$:

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

where $z_n = \int k(x, x_n) p(x) dx$
How to select samples?

- Natural to minimize the posterior variance of $Z$:

$$\nabla \mathbb{E}[Z|f(x_s)] = \int \int k(x, x') p(x)p(x') dx dx' - z^T K^{-1} z$$

where $z_n = \int k(x, x_n) p(x) dx$

- 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 $Z$:

$$\nabla [Z | f(x_s)] = \int\int k(x, x') p(x) p(x') dx dx' - z^T K^{-1} z$$

where $$z_n = \int k(x, x_n) p(x) dx$$

- 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 $Z$:

  \[
  \nabla [Z|f(x_s)] = \int \int k(x, x') p(x) p(x') dx dx' - z^T K^{-1} z \\
  \]

  where \( z_n = \int k(x, x_n) p(x) dx \)

- 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

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

Where

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

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

Where

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

**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:

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

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

\( \mathbb{V} [Z|f(x_s)] \) 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**

![Graph showing expected variance and MMD](image-url)
Rates of Convergence

What is rate of convergence of BQ?

Expected Variance / MMD

![Graph showing MMD vs number of samples for different methods: O(1/N), i.i.d. sampling, Herding with 1/N weights, and Herding with BQ weights.](image)
Rates of Convergence

What is rate of convergence of BQ?

Expected Variance / MMD

![Graph showing the convergence of MMD with different sampling methods and weights.](image-url)
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
Outline

- Optimally-weighted Herding is Bayesian Quadrature
  - Kernel Herding
  - Bayesian Quadrature
  - Unifying Results
  - Demos

- Frequentist Methods, Bayesian Takeaways
  - Kernel Herding
  - Mean Embeddings
  - Kernel Two-sample Test
  - Hilbert-Schmidt Independence Criterion
  - Determinantal Point Processes
GPs vs Log-GPs for Inference
GPs vs Log-GPs for Inference
GPs vs Log-GPs for Inference

\[ \ell(x) \]

\[ \log \ell(x) \]

- True Function
- Evaluations
- GP Posterior
- True Log-func
- Evaluations
GPs vs Log-GPs for Inference

The figure compares the true function (black line) and the GP posterior (red line) in two different plots. The top plot shows the function $\ell(x)$ and its evaluations (black crosses). The bottom plot shows the log of the function $\log \ell(x)$ and its evaluations (black crosses). The true log-function is also shown in blue. The evaluations are indicated by black crosses. The plots illustrate the performance of GPs and Log-GPs for inference.
GPs vs Log-GPs for Inference

- True Function
- Evaluations
- GP Posterior
- Log-GP Posterior

\[ \ell(x) \]

\[ \log \ell(x) \]
GPs vs Log-GPs for Inference

Takeaway: Herding assumptions are inappropriate for inference

- True Function
- Evaluations
- GP Posterior
- Log-GP Posterior

True Log-func
- Evaluations
- GP Posterior
Mean Embedding Interpretation

- [Muandet & Ghahramani, 2012] showed that if $f \sim \text{GP}$,
Mean Embedding Interpretation

- [Muandet & Ghahramani, 2012] showed that if \( f \sim \text{GP} \),

\[
\mu_{p(x)} = \int \phi(x)p(x)dx
= \int k(x, \cdot)p(x)dx
= \mathbb{E}_{f \sim \text{GP}} \left[ \int f(x)f(\cdot)p(x)dx \right]
= \mathbb{E}_{f \sim \text{GP}} \left[ f(\cdot) \int f(x)p(x)dx \right]
= \text{cov}_{f \sim \text{GP}} (f(\cdot), Z_p)
\]
Mean Embedding Interpretation

- [Muandet & Ghahramani, 2012] showed that if \( f \sim \text{GP} \),

\[
\mu_{p(x)} = \int \phi(x)p(x)dx
\]

\[
= \int k(x, \cdot)p(x)dx
\]

\[
= \mathbb{E}_{f \sim \text{GP}} \left[ \int f(x)f(\cdot)p(x)dx \right]
\]

\[
= \mathbb{E}_{f \sim \text{GP}} \left[ f(\cdot) \int f(x)p(x)dx \right]
\]

\[
= \text{cov}_{f \sim \text{GP}}(f(\cdot), Z_p)
\]

- \( \mu_{p(x)} \) is the covariance the function with its integral with respect to \( p(x) \).
Kernel two-sample test

- How to test whether two distributions $p(x)$ and $q(x)$ are the same?
Kernel two-sample test

- How to test whether two distributions $p(x)$ and $q(x)$ are the same?
- New test statistic: $\text{MMD}(p, q)$ [Gretton et. al, 2005]
Kernel two-sample test

- How to test whether two distributions $p(x)$ and $q(x)$ are the same?
- New test statistic: $\text{MMD}(p, q)$ [Gretton et. al, 2005]
- Equivalent Bayesian interpretation:

$$\text{MMD}^2_k(p, q) = \mathbb{V}_{f \sim \text{GP}_k} \left[ \int f(x)p(x)dx - \int f(x)q(x)dx \right]$$

$p$ and $q$ are similar if integrals of functions drawn from a GP prior have similar integrals.
Kernel two-sample test

• How to test whether two distributions $p(x)$ and $q(x)$ are the same?

• New test statistic: $\text{MMD}(p, q)$ [Gretton et. al, 2005]

• Equivalent Bayesian interpretation:

$$\text{MMD}^2_k(p, q) = \mathbb{V}_{f \sim \text{GP}_k} \left[ \int f(x)p(x)dx - \int f(x)q(x)dx \right]$$

$p$ and $q$ are similar if integrals of functions drawn from a GP prior have similar integrals.

Possible takeaways: Decision-theoretic choice of kernel, sampling-based methods for computing $\text{MMD}$
Hilbert-Schmidt Independence Criterion

- Given samples \( \{X, Y\} \sim p(x, y) \), how to test whether 
  \( p(x, y) = p(x)p(y) \)?
Hilbert-Schmidt Independence Criterion

- Given samples \( \{X, Y\} \sim p(x, y) \), how to test whether \( p(x, y) = p(x)p(y) \)?
- New test statistic based on infinite-dimensional Frobenius norm of cross-covariance matrix of features of \( x \) and \( y \):
  \[ \text{HSIC}(p(x, y), k_x, k_y) = ||C_{xy}||^2 \]
  \[ = E_{x, x', y, y'} \left[ k_x(x, x') k_y(y, y') \right] + E_{x, x'} \left[ k_x(x, x') \right] E_{y, y'} \left[ k_y(y, y') \right] - 2 E_{x, y} \left[ E_{x'} \left[ k_x(x, x') \right] E_{y'} \left[ k_y(y, y') \right] \right] \]

[Gretton et. al, 2005]
Hilbert-Schmidt Independence Criterion

- Given samples \( \{X, Y\} \sim p(x, y) \), how to test whether \( p(x, y) = p(x)p(y) \)?

- New test statistic based on infinite-dimensional Frobenius norm of cross-covariance matrix of features of \( x \) and \( y \): [Gretton et. al, 2005]

\[
\text{HSIC}(p(x, y), k_x, k_y) = \|C_{xy}\|_{HS}^2 \\
= \mathbb{E}_{x,x',y,y'} [k_x(x, x')k_y(y, y)] + \mathbb{E}_{x,x'} [k_x(x, x')] \mathbb{E}_{y,y'} [k_y(y, y')] \\
- 2\mathbb{E}_{x,y} [\mathbb{E}_{x'} [k_x(x, x')] \mathbb{E}_{y'} [k_y(y, y)]]
\]
Hilbert-Schmidt Independence Criterion

- Given samples \( \{X, Y\} \sim p(x, y) \), how to test whether \( p(x, y) = p(x)p(y) \)?
- New result: Assuming \( k(x, y, x', y') = k_x(x, x')k_y(y, y') \)

\[
\text{HSIC}(p(x, y), k_x, k_y) = \\
= \mathbb{V}_{f \sim \text{GP}_k} \left[ \int f(x, y)p(x, y)dx\,dy - \int f(x, y)p(x)p(y)dx\,dy \right]
\]
Determinantal Point Processes

- Probability of a set $P(\mathcal{X}) = |K(\mathcal{X}, \mathcal{X})|$
Determinantal Point Processes

- Probability of a set $P(\mathcal{X}) = |K(\mathcal{X}, \mathcal{X})|$
- Greedy MAP maximizes $P(\mathcal{X} \cup x_i)$

Related to sensor placement work by Andreas Krause

Thanks to Le Song and Roman Garnett
Determinantal Point Processes

- Probability of a set $P(\mathcal{X}) = |K(\mathcal{X}, \mathcal{X})|$
- Greedy MAP maximizes $P(\mathcal{X} \cup x_i)$

\[
P(\mathcal{X} \cup x_i)
= |K(\mathcal{X} \cup x_i, \mathcal{X} \cup x_i)|
= |K(\mathcal{X}, \mathcal{X})| [k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)]
\propto k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)
= \mathbb{V}_{f \sim \text{GP}_k} [f(x_i)|\mathcal{X}]
\]
Determinantal Point Processes

- Probability of a set $P(\mathcal{X}) = |K(\mathcal{X}, \mathcal{X})|$  

- Greedy MAP maximizes $P(\mathcal{X} \cup x_i)$

\[
P(\mathcal{X} \cup x_i) \\
= |K(\mathcal{X} \cup x_i, \mathcal{X} \cup x_i)| \\
= |K(\mathcal{X}, \mathcal{X})| [k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)] \\
\propto k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i) \\
= \nabla_{f \sim \text{GP}_k} [f(x_i) | \mathcal{X}]
\]

- New DPP Point added at location with highest marginal variance in GP posterior, conditioned on the other points

Thanks to Le Song and Roman Garnett.
Determinantal Point Processes

- Probability of a set $P(\mathcal{X}) = |K(\mathcal{X}, \mathcal{X})|$ 
- Greedy MAP maximizes $P(\mathcal{X} \cup x_i)$

$$P(\mathcal{X} \cup x_i) = |K(\mathcal{X} \cup x_i, \mathcal{X} \cup x_i)|$$

$$= |K(\mathcal{X}, \mathcal{X})| [k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)]$$

$$\propto k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)$$

$$= \nabla_{f \sim \text{GP}_k} [f(x_i)|\mathcal{X}]$$

- New DPP Point added at location with highest marginal variance in GP posterior, conditioned on the other points
- Related to sensor placement work by Andreas Krause

Thanks to Le Song and Roman Garnett
Determinantal Point Processes

- Probability of a set $P(\mathcal{X}) = |K(\mathcal{X}, \mathcal{X})|$
- Greedy MAP maximizes $P(\mathcal{X} \cup x_i)$

\[
P(\mathcal{X} \cup x_i)
= |K(\mathcal{X} \cup x_i, \mathcal{X} \cup x_i)|
= |K(\mathcal{X}, \mathcal{X})| [k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)]
\propto k(x_i, x_i) - k(x_i, \mathcal{X})K(\mathcal{X}, \mathcal{X})^{-1}k(\mathcal{X}, x_i)
= \nabla_{f \sim \text{GP}_k} [f(x_i)|\mathcal{X}]
\]

- New DPP Point added at location with highest marginal variance in GP posterior, conditioned on the other points
- Related to sensor placement work by Andreas Krause
- Thanks to Le Song and Roman Garnett
### Summary of Connections

<table>
<thead>
<tr>
<th>Frequentist Method</th>
<th>Bayesian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel herding</td>
<td>Bayesian Quadrature</td>
</tr>
<tr>
<td>Mean Embedding</td>
<td>Covariance with integral</td>
</tr>
<tr>
<td>Kernel Two-sample test</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>HSIC</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>DPP MAP</td>
<td>Marginal uncertainty in GP posterior</td>
</tr>
</tbody>
</table>
### Summary of Connections

<table>
<thead>
<tr>
<th>Frequentist Method</th>
<th>Bayesian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel herding</td>
<td>Bayesian Quadrature</td>
</tr>
<tr>
<td>Mean Embedding</td>
<td>Covariance with integral</td>
</tr>
<tr>
<td>Kernel Two-sample test</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>HSIC</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>DPP MAP</td>
<td>Marginal uncertainty in GP posterior</td>
</tr>
</tbody>
</table>

Some possible extensions:
## Summary of Connections

<table>
<thead>
<tr>
<th>Frequentist Method</th>
<th>Bayesian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel herding</td>
<td>Bayesian Quadrature</td>
</tr>
<tr>
<td>Mean Embedding</td>
<td>Covariance with integral</td>
</tr>
<tr>
<td>Kernel Two-sample test</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>HSIC</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>DPP MAP</td>
<td>Marginal uncertainty in GP posterior</td>
</tr>
</tbody>
</table>

Some possible extensions:

- Log-kernel herding for inference
## Summary of Connections

<table>
<thead>
<tr>
<th>Frequentist Method</th>
<th>Bayesian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel herding</td>
<td>Bayesian Quadrature</td>
</tr>
<tr>
<td>Mean Embedding</td>
<td>Covariance with integral</td>
</tr>
<tr>
<td>Kernel Two-sample test</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>HSIC</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>DPP MAP</td>
<td>Marginal uncertainty in GP posterior</td>
</tr>
</tbody>
</table>

Some possible extensions:

- Log-kernel herding for inference
- Interpretation of conditional mean embeddings
## Summary of Connections

<table>
<thead>
<tr>
<th>Frequentist Method</th>
<th>Bayesian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel herding</td>
<td>Bayesian Quadrature</td>
</tr>
<tr>
<td>Mean Embedding</td>
<td>Covariance with integral</td>
</tr>
<tr>
<td>Kernel Two-sample test</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>HSIC</td>
<td>Variance of difference of integrals</td>
</tr>
<tr>
<td>DPP MAP</td>
<td>Marginal uncertainty in GP posterior</td>
</tr>
</tbody>
</table>

Some possible extensions:

- Log-kernel herding for inference
- Interpretation of conditional mean embeddings
- Different low-rank approximations based on sparse GPs
<table>
<thead>
<tr>
<th>Frequentist Method</th>
<th>Bayesian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel Regression</td>
<td>GP Regression</td>
</tr>
<tr>
<td>Functional ANOVA, Hierchical Kernel Learning</td>
<td>Additive Gaussian Processes</td>
</tr>
<tr>
<td>Kernel Bayes’ Rule</td>
<td>Bayesian Quadrature for Ratios</td>
</tr>
<tr>
<td>RKHS Embeddings of Conditional Distributions</td>
<td>GP Dynamics Models</td>
</tr>
<tr>
<td>Kernel Message Passing</td>
<td>???</td>
</tr>
<tr>
<td>Frequentist Method</td>
<td>Bayesian Method</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Kernel Regression</td>
<td>GP Regression</td>
</tr>
<tr>
<td>Functional ANOVA, Hierchical Kernel Learning</td>
<td>Additive Gaussian Processes</td>
</tr>
<tr>
<td>Kernel Bayes’ Rule</td>
<td>Bayesian Quadrature for Ratios</td>
</tr>
<tr>
<td>RKHS Embeddings of Conditional Distributions</td>
<td>GP Dynamics Models</td>
</tr>
<tr>
<td>Kernel Message Passing</td>
<td>???</td>
</tr>
</tbody>
</table>

Thanks!