Kernel Adaptive Metropolis-Hastings

Arthur Gretton,*

*Gatsby Unit, CSML, University College London

NIPS, December 2015
Metropolis-Hastings MCMC

- Unnormalized target $\pi(x) \propto p(x)$
- Generate Markov chain with invariant distribution $p$
  - Initialize $x_0 \sim p_0$
  - At iteration $t \geq 0$, propose to move to state $x' \sim q(\cdot|x_t)$
  - Accept/Reject proposals based on ratio

$$x_{t+1} = \begin{cases} x', & \text{w.p. } \min \left\{ 1, \frac{\pi(x')q(x_t|x')}{\pi(x_t)q(x'|x_t)} \right\} \\ x_t, & \text{otherwise.} \end{cases}$$

- What proposal $q(\cdot|x_t)$?
Unnormalized target $\pi(x) \propto p(x)$

Generate Markov chain with invariant distribution $p$
- Initialize $x_0 \sim p_0$
- At iteration $t \geq 0$, propose to move to state $x' \sim q(\cdot|x_t)$
- Accept/Reject proposals based on ratio

$$x_{t+1} = \begin{cases} x', & \text{w.p. min} \left\{1, \frac{\pi(x')q(x_t|x')}{\pi(x_t)q(x'|x_t)}\right\}, \\ x_t, & \text{otherwise.} \end{cases}$$

What proposal $q(\cdot|x_t)$?
- Too narrow or broad: $\rightarrow$ slow convergence
- Does not conform to support of target $\rightarrow$ slow convergence
Adaptive MCMC

- **Adaptive Metropolis** *(Haario, Saksman & Tamminen, 2001):* Update proposal $q_t(\cdot | x_t) = \mathcal{N}(x_t, \nu^2 \hat{\Sigma}_t)$, using estimates of the target covariance.
Adaptive MCMC

- **Adaptive Metropolis** *(Haario, Saksman & Tamminen, 2001):* Update proposal \( q_t(\cdot | x_t) = \mathcal{N}(x_t, \nu^2 \hat{\Sigma}_t) \), using estimates of the target covariance.
Adaptive MCMC

- Adaptive Metropolis \((\text{Haario, Saksman & Tamminen, 2001})\):
  Update proposal \(q_t(\cdot | x_t) = \mathcal{N}(x_t, \nu^2 \hat{\Sigma}_t)\), using estimates of the target covariance

Locally miscalibrated for strongly non-linear targets: directions of large variance depend on the current location
Motivation: Intractable & Non-linear Targets

- **Previous solutions** for non-linear targets: Hamiltonian Monte Carlo (HMC) or Metropolis Adjusted Langevin Algorithms (MALA) (Roberts & Stramer, 2003; Girolami & Calderhead, 2011).
- Require **target gradients and second order information**
Motivation: Intractable & Non-linear Targets

- **Previous solutions** for non-linear targets: Hamiltonian Monte Carlo (HMC) or Metropolis Adjusted Langevin Algorithms (MALA) (Roberts & Stramer, 2003; Girolami & Calderhead, 2011).
- Require target gradients and second order information

**Our case:** not even target $\pi(\cdot)$ can be computed – **Pseudo-Marginal MCMC** (Beaumont, 2003; Andrieu & Roberts, 2009).
Bayesian Gaussian Process Classification

Example: when is target not computable?

- **GPC model**: latent process $f$, labels $y$, (with covariate matrix $X$), and hyperparameters $\theta$:

$$p(f, y, \theta) = p(\theta)p(f|\theta)p(y|f)$$

$$f|\theta \sim \mathcal{N}(0, K_\theta)$$ GP with covariance $K_\theta$
Bayesian Gaussian Process Classification

Example: when is target not computable?

- **GPC model**: latent process $f$, labels $y$, (with covariate matrix $X$), and hyperparameters $\theta$:

  $$p(f, y, \theta) = p(\theta)p(f|\theta)p(y|f)$$

  $$f|\theta \sim \mathcal{N}(0, \mathcal{K}_\theta) \text{ GP with covariance } \mathcal{K}_\theta$$

- **Automatic Relevance Determination (ARD) covariance**:

  $$(\mathcal{K}_\theta)_{ij} = \kappa(x_i, x_j'|\theta) = \exp \left( -\frac{1}{2} \sum_{s=1}^{d} \frac{(x_i,s - x_j',s)^2}{\exp(\theta_s)} \right)$$
Pseudo-Marginal MCMC

Example: when is target not computable?

- Gaussian process classification, latent process $f$

\[ p(\theta|\mathbf{y}) \propto p(\theta)p(\mathbf{y}|\theta) = p(\theta) \int p(f|\theta)p(\mathbf{y}|f, \theta)df =: \pi(\theta) \]

... but cannot integrate out $f$
Pseudo-Marginal MCMC

Example: when is target not computable?

- **Gaussian process classification**, latent process $f$

  \[ p(\theta | y) \propto p(\theta) p(y | \theta) = p(\theta) \int p(f | \theta) p(y | f, \theta) df =: \pi(\theta) \]

  ... but cannot integrate out $f$

- MH ratio:

  \[ \alpha(\theta, \theta') = \min \left\{ 1, \frac{p(\theta') p(y | \theta') q(\theta | \theta')}{p(\theta) p(y | \theta) q(\theta' | \theta)} \right\} \]
Pseudo-Marginal MCMC

Example: when is target not computable?

- **Gaussian process classification**, latent process $f$

\[
p(\theta | y) \propto p(\theta)p(y|\theta) = p(\theta) \int p(f|\theta)p(y|f, \theta)df =: \pi(\theta)
\]

... but cannot integrate out $f$

- MH ratio:

\[
\alpha(\theta, \theta') = \min \left\{ 1, \frac{p(\theta')p(y|\theta')q(\theta'|\theta)}{p(\theta)p(y|\theta)q(\theta'|\theta)} \right\}
\]

- **Filippone & Girolami, 2013** use Pseudo-Marginal MCMC: unbiased estimate of $p(y|\theta)$ via importance sampling:

\[
\hat{p}(\theta | y) \propto p(\theta)\hat{p}(y|\theta) \approx p(\theta)\frac{1}{n_{\text{imp}}} \sum_{i=1}^{n_{\text{imp}}} p(y|f^{(i)}) \frac{p(f^{(i)}|\theta)}{Q(f^{(i)})}
\]
Pseudo-Marginal MCMC

Example: when is target not computable?

- **Gaussian process classification**, latent process $f$

  \[
p(\theta | y) \propto p(\theta) p(y | \theta) = p(\theta) \int p(f | \theta) p(y | f, \theta) df =: \pi(\theta)
  \]

  ... but cannot integrate out $f$

- **Estimated** MH ratio:

  \[
  \alpha(\theta, \theta') = \min \left\{ 1, \frac{p(\theta') \hat{p}(y | \theta') q(\theta | \theta')}{p(\theta) \hat{p}(y | \theta) q(\theta' | \theta)} \right\}
  \]
Pseudo-Marginal MCMC

Example: when is target not computable?

- **Gaussian process classification**, latent process $f$

$$p(\theta|y) \propto p(\theta)p(y|\theta) = p(\theta) \int p(f|\theta)p(y|f, \theta)df =: \pi(\theta)$$

... but cannot integrate out $f$

- Estimated MH ratio:

$$\alpha(\theta, \theta') = \min \left\{ 1, \frac{p(\theta')\hat{p}(y|\theta')q(\theta'|\theta)}{p(\theta)\hat{p}(y|\theta)q(\theta'|\theta)} \right\}$$

- Replacing marginal likelihood $p(y|\theta)$ with **unbiased estimate** $\hat{p}(y|\theta)$ still results in **correct invariant distribution** [Beaumont, 2003; Andrieu & Roberts, 2009]
Intractable & Non-linear Target in GPC

- Sliced posterior over hyperparameters of a Gaussian Process classifier on UCI Glass dataset obtained using Pseudo-Marginal MCMC

Adaptive sampler that learns the shape of non-linear targets without gradient information?
Two strategies for adaptive sampling

Kameleon (Sejdinovic et al. 2014)
- Learns covariance in RKHS.
- Locally aligns to (non-linear) target covariance, gradient free.

Kernel Adaptive Hamiltonian Monte Carlo (Strathmann et al. 2015)
- Learns global estimate of gradient of log target density
The Kameleon

D. Sejdinovic, H. Strathmann, M. Lomeli, C. Andrieu, and A. Gretton, ICML 2014
Use feature space covariance

- Capture non-linearities using linear covariance $C_z$ in feature space $\mathcal{H}$
Use feature space covariance

- Capture non-linearities using linear covariance $C_z$ in feature space $\mathcal{H}$
Capture non-linearities using linear covariance $C_z$ in feature space $\mathcal{H}$

Find a nearby pre-image in $\mathcal{X}$
(gradient descent)
Use feature space covariance

- Capture non-linearities using linear covariance $C_z$ in feature space $\mathcal{H}$

Input space $\mathcal{X}$

Feature space $\mathcal{H}$

Feature space sample $f$

Find a nearby pre-image in $\mathcal{X}$ (gradient descent)
Proposal Construction Summary

1. Get a chain subsample $z = \{z_i\}_{i=1}^n$
2. Construct an RKHS sample $f \sim \mathcal{N}(\phi(x_t), \nu^2 C_z)$
3. Propose $x^*$ such that $\phi(x^*)$ is close to $f$ (with an additional exploration term $\xi \sim \mathcal{N}(0, \gamma^2 I_d)$).
Proposal Construction Summary

1. Get a chain subsample $z = \{z_i\}_{i=1}^n$
2. Construct an RKHS sample $f \sim \mathcal{N}(\phi(x_t), \nu^2 C_z)$
3. Propose $x^*$ such that $\phi(x^*)$ is close to $f$ (with an additional exploration term $\xi \sim \mathcal{N}(0, \gamma^2 I_d)$).

Integrate out RKHS samples $f$, gradient step, and $\xi$ to obtain marginal Gaussian proposal on the input space:

$$q_z(x^* | x_t) = \mathcal{N}(x_t, \gamma^2 I_d + \nu^2 M_{z,x_t} H M_{z,x_t}^\top)$$

$$M_{z,x_t} = 2 [\nabla_x k(x, z_1)|_{x=x_t}, \ldots, \nabla_x k(x, z_n)|_{x=x_t}],$$

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}.$$
Examples of Covariance Structure for Standard Kernels

Kameleon proposals capture local covariance structure

**Gaussian kernel:** \[ k(x, x') = \exp\left(-\frac{1}{2} \sigma^{-2} \|x - x'\|_2^2\right) \]

\[
[\text{cov}[q_{z(.|y)}]]_{ij} = \gamma^2 \delta_{ij} + \frac{4\nu^2}{\sigma^4} \sum_{a=1}^{n} [k(y, z_a)]^2 (z_{a,i} - y_i)(z_{a,j} - y_j) + O\left(\frac{1}{n}\right).
\]
Kernel Adaptive Hamiltonian Monte Carlo (KMC)

Heiko Strathmann, Dino Sejdinovic, Samuel Livingstone, Zoltan Szabo, and Arthur Gretton, NIPS 2015
Hamiltonian Monte Carlo

- HMC: distant moves, high acceptance probability.
- Potential energy \( U(q) = -\log \pi(q) \), auxiliary momentum \( p \sim \exp(-K(p)) \), simulate for \( t \in \mathbb{R} \) along Hamiltonian flow of \( H(p, q) = K(p) + U(q) \), using operator

\[
\frac{\partial K}{\partial p} \frac{\partial}{\partial q} - \frac{\partial U}{\partial q} \frac{\partial}{\partial p}
\]

- Numerical simulation (i.e. leapfrog) depends on gradient information.

What if gradient unavailable, e.g. in Bayesian GP classification?
Infinite dimensional exponential families

Proposal is RKHS exponential family model [Fukumizu, 2009; Sriperumbudur et al. 2014], but accept using true Hamiltonian (to correct for both model and leapfrog)

$$\text{const} \times \pi(x) \approx \exp(\langle f, k(x, \cdot) \rangle_\mathcal{H} - A(f))$$

- Sufficient statistics: feature map $k(\cdot, x) \in \mathcal{H}$, satisfies $f(x) = \langle f, k(x, \cdot) \rangle_\mathcal{H}$ for any $f \in \mathcal{H}$.
- Natural parameters: $f \in \mathcal{H}$.

The model is

- dense in continuous densities on compact domains (TV, KL, etc.),
- relatively robust to increasing dimensions, as opposed to e.g. KDE.

How to learn $f$ from samples without access to $A(f)$?
Score matching

- Estimation of unnormalised density models from samples [Sriperumbudur et al. 2014]
- Minimises Fisher divergence

\[ J(f) = \frac{1}{2} \int \pi(x) \| \nabla f(x) - \nabla \log \pi(x) \|_2^2 \, dx \]

- Possible without accessing \( \nabla \log \pi(x) \) and accessing \( \pi(x) \) only through samples \( x := \{x_i\}_{i=1}^t \)

\[ \hat{J}(f) = \hat{E}_x \left\{ \sum_{\ell=1}^d \left[ \frac{\partial^2 f(x)}{\partial x_\ell^2} + \frac{1}{2} \left( \frac{\partial f(x)}{\partial x_\ell} \right)^2 \right] \right\} \]

Expensive: full solution requires solving \((td + 1)\)-dimensional linear system.
Approximate solution: KMC finite

\[ f(x) = \theta^\top \phi_x \]

- **Random Fourier Features**
  \[ \phi_x^\top \phi_y \approx k(x, y) \]

- \( \theta \in \mathbb{R}^m \) can be computed from
  \[ \hat{\theta}_\lambda := (C + \lambda I)^{-1} b \]

  \[
  b := -\frac{1}{t} \sum_{i=1}^{t} \sum_{\ell=1}^{d} \hat{\phi}_{x_i}^{\ell} \\
  C := \frac{1}{t} \sum_{i=1}^{t} \sum_{\ell=1}^{d} \hat{\phi}_{x_i}^{\ell} \left( \phi_{x_i}^{\ell} \right)^\top
  \]

  where \( \hat{\phi}_{x}^{\ell} := \frac{\partial}{\partial x_{\ell}} \phi_{x} \) and \( \phi_{x}^{\ell} := \frac{\partial^2}{\partial x_{\ell}^2} \phi_{x} \).

- **On-line updates** cost \( O(dm^2) \).

Updates fast, uses *all* Markov chain history. Caveat: need to initialise correctly.

Gradient norm: Gaussian

KMC Finite
Approximate solution: KMC lite

\[ f(x) = \sum_{i=1}^{n} \alpha_i k(z_i, x) \]

- \( z \subseteq x \) sub-sample.
- \( \alpha \) from linear system

\[ \hat{\alpha}_\lambda = -\frac{\sigma}{2} (C + \lambda I)^{-1} b \]

where \( C \in \mathbb{R}^{n \times n} \), \( b \in \mathbb{R}^n \) depend on kernel matrix

- Cost \( O(n^3 + n^2 d) \) (or cheaper with low-rank approx., conjugate gradient).

Geometrically ergodic on log-concave targets (fast convergence).

Gradient norm: Gaussian

KMC Lite
Does kernel HMC work in high dimensions?

Challenging Gaussian target (**top**):
- Eigenvalues: $\lambda_i \sim \text{Exp}(1)$.
- Covariance: $\text{diag}(\lambda_1, \ldots, \lambda_d)$, randomly rotate.
- Use Rational Quadratic kernel to account for resulting highly ‘non-singular’ length-scales.
- KMC scales up to $d \approx 30$.

An easy, isotropic Gaussian target (**bottom**):
- More smoothness allows KMC to scale up to $d \approx 100$. 
Synthetic targets: Banana

**Banana**: $B(b, \nu)$: take $X \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \text{diag}(\nu, 1, \ldots, 1)$, and set $Y_2 = X_2 + b(X_1^2 - \nu)$, and $Y_i = X_i$ for $i \neq 2$. (Haario et al, 1999; 2001)
Synthetic targets: Banana

**Acc. rate**

**Minimum ESS**

*KMC behaves like HMC as number n of oracle samples increases.*
Gaussian Process Classification on UCI data

- Standard GPC model

\[ p(f, y, \theta) = p(\theta)p(f|\theta)p(y|f) \]

where \( p(f|\theta) \) is a GP and with a sigmoidal likelihood \( p(y|f) \).

- Goal: sample from \( p(\theta|y) \propto p(\theta)p(y|\theta) \).

- Unbiased estimate of \( \hat{p}(y|\theta) \) via importance sampling.

- No access to likelihood or gradient.
Gaussian Process Classification on UCI data

- Standard GPC model

\[ p(f, y, \theta) = p(\theta)p(f|\theta)p(y|f) \]

where \( p(f|\theta) \) is a GP and with a sigmoidal likelihood \( p(y|f) \).

- Goal: sample from

\[ p(\theta|y) \propto p(\theta)p(y|\theta). \]

- Unbiased estimate of \( \hat{p}(y|\theta) \) via importance sampling.

- No access to likelihood or gradient.

**Significant mixing improvements over state-of-the-art.**
Conclusions

- Simple, versatile, gradient-free adaptive MCMC samplers:
  - Kameleon:
    - Uses local covariance structure of the target distribution at the current chain state
  - Kernel HMC
    - Derivative of log density fit to samples, use this as proposal in HMC.
- Outperforms existing adaptive approaches on nonlinear target distributions
- Future work: For Kameleon, does feature space covariance track high density regions in original space? For kernel HMC, how does convergence rate degrade with increasing dimension?

- **Kameleon code**: https://github.com/karlnapf/kameleon-mcmc
- **Kernel HMC code**: https://github.com/karlnapf/kernel_hmc
Bayesian Gaussian Process Classification

- GPC model: latent process $f$, labels $y$, (with covariate matrix $X$), and hyperparameters $\theta$:

  $$p(f, y, \theta) = p(\theta)p(f|\theta)p(y|f)$$

  where $f|\theta \sim \mathcal{N}(0, K_\theta)$ is a realization of a GP with covariance $K_\theta$ (covariance between latent processes evaluated at $X$).
Bayesian Gaussian Process Classification

- **GPC model**: latent process $f$, labels $y$, (with covariate matrix $X$), and hyperparameters $\theta$:

  $$ p(f, y, \theta) = p(\theta) p(f|\theta) p(y|f) $$

  where $f|\theta \sim \mathcal{N}(0, K_\theta)$ is a realization of a GP with covariance $K_\theta$ (covariance between latent processes evaluated at $X$).

- $K_\theta$: exponentiated quadratic Automatic Relevance Determination (ARD) covariance:

  $$ (K_\theta)_{ij} = \kappa(x_i, x_j'|\theta) = \exp \left( -\frac{1}{2} \sum_{s=1}^{d} \frac{(x_{i,s} - x_{j,s}')^2}{\exp(\theta_s)} \right) $$
Bayesian Gaussian Process Classification (2)

- Fully Bayesian treatment: Interested in the posterior $p(\theta|y)$
Bayesian Gaussian Process Classification (2)

- Fully Bayesian treatment: Interested in the posterior $p(\theta|y)$
- Cannot use a Gibbs sampler on $p(\theta, f|y)$, which samples from $p(f|\theta, y)$ and $p(\theta|f, y)$ in turns, since $p(\theta|f, y)$ is extremely sharp
Bayesian Gaussian Process Classification (2)

- Fully Bayesian treatment: Interested in the posterior $p(\theta|y)$
- Cannot use a Gibbs sampler on $p(\theta, f|y)$, which samples from $p(f|\theta, y)$ and $p(\theta|f, y)$ in turns, since $p(\theta|f, y)$ is extremely sharp
- Filippone & Girolami, 2013 use Pseudo-Marginal MCMC to sample $p(\theta|y) = p(\theta) \int p(\theta, f|y)p(f|\theta)df$. 

Unbiased estimate of $\hat{p}(y|\theta)$ via importance sampling:

$\hat{p}(\theta|y) \propto p(\theta)p(y|\theta) \approx p(\theta)\frac{1}{n_{imp}}\sum_{i=1}^{n_{imp}} p(y|f(i))p(f(i)|\theta)Q(f(i))$
Bayesian Gaussian Process Classification (2)

- Fully Bayesian treatment: Interested in the posterior $p(\theta|y)$
- Cannot use a Gibbs sampler on $p(\theta, f|y)$, which samples from $p(f|\theta, y)$ and $p(\theta|f, y)$ in turns, since $p(\theta|f, y)$ is extremely sharp
- **Filippone & Girolami, 2013** use Pseudo-Marginal MCMC to sample $p(\theta|y) = p(\theta) \int p(\theta, f|y)p(f|\theta)df$.
- Unbiased estimate of $\hat{p}(y|\theta)$ via importance sampling:

  $$\hat{p}(\theta|y) \propto p(\theta)\hat{p}(y|\theta) \approx p(\theta) \frac{1}{n_{imp}} \sum_{i=1}^{n_{imp}} p(y|f(i)) \frac{p(f(i)|\theta)}{Q(f(i))}$$
Fully Bayesian treatment: Interested in the posterior $p(\theta|y)$

Cannot use a Gibbs sampler on $p(\theta, f|y)$, which samples from $p(f|\theta, y)$ and $p(\theta|f, y)$ in turns, since $p(\theta|f, y)$ is extremely sharp

Filippone & Girolami, 2013 use Pseudo-Marginal MCMC to sample $p(\theta|y) = p(\theta) \int p(\theta, f|y)p(f|\theta)df$.

Unbiased estimate of $\hat{p}(y|\theta)$ via importance sampling:

$\hat{p}(\theta|y) \propto p(\theta)p(\theta|y) \approx p(\theta)\frac{1}{n_{\text{imp}}} \sum_{i=1}^{n_{\text{imp}}} p(y|f^{(i)}) \frac{p(f^{(i)}|\theta)}{Q(f^{(i)})}$

No access to likelihood, gradient, or Hessian of the target.
RKHS and Kernel Embedding

- For any positive semidefinite function $k$, there is a unique RKHS $\mathcal{H}_k$. Can consider $x \mapsto k(\cdot, x)$ as a feature map.

- Definition (Kernel embedding)
  Let $k$ be a kernel on $X$, and $P$ a probability measure on $X$. The kernel embedding of $P$ into the RKHS $\mathcal{H}_k$ is $\mu_k(P) \in \mathcal{H}_k$ such that $E_{P} f(X) = \langle f, \mu_k(P) \rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$.

Alternatively, can be defined by the Bochner integral $\mu_k(P) = \int k(\cdot, x) dP(x)$ (expected canonical feature).

For many kernels $k$, including the Gaussian, Laplacian and inverse multi-quadratics, the kernel embedding $P \mapsto \mu_P$ is injective: characteristic (Sriperumbudur et al, 2010), captures all moments (similarly to the characteristic function).
RKHS and Kernel Embedding

- For any positive semidefinite function $k$, there is a unique RKHS $\mathcal{H}_k$. Can consider $x \mapsto k(\cdot, x)$ as a feature map.

**Definition (Kernel embedding)**

Let $k$ be a kernel on $\mathcal{X}$, and $P$ a probability measure on $\mathcal{X}$. The kernel embedding of $P$ into the RKHS $\mathcal{H}_k$ is $\mu_k(P) \in \mathcal{H}_k$ such that $\mathbb{E}_P f(X) = \langle f, \mu_k(P) \rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$.

- Alternatively, can be defined by the Bochner integral $\mu_k(P) = \int k(\cdot, x) \, dP(x)$ (expected canonical feature).

- For many kernels $k$, including the Gaussian, Laplacian and inverse multi-quadratics, the kernel embedding $P \mapsto \mu_P$ is injective: characteristic (Sriperumbudur et al, 2010), captures all moments (similarly to the characteristic function).
Covariance operator

**Definition**

The covariance operator of $P$ is $C_P : \mathcal{H}_k \rightarrow \mathcal{H}_k$ such that $\forall f, g \in \mathcal{H}_k$, 
$$\langle f, C_P g \rangle_{\mathcal{H}_k} = \text{Cov}_P [f(X)g(X)].$$
Covariance operator

**Definition**

The covariance operator of $P$ is $C_P : \mathcal{H}_k \rightarrow \mathcal{H}_k$ such that $\forall f, g \in \mathcal{H}_k$, $\langle f, C_P g \rangle_{\mathcal{H}_k} = \text{Cov}_P [f(X)g(X)]$.

- **Covariance operator**: $C_P : \mathcal{H}_k \rightarrow \mathcal{H}_k$ is given by
  \[
  C_P = \int k(\cdot, x) \otimes k(\cdot, x) \, dP(x) - \mu_P \otimes \mu_P \text{ (covariance of canonical features)}
  \]

- **Empirical versions of embedding and the covariance operator**:
  \[
  \begin{align*}
  \mu_z &= \frac{1}{n} \sum_{i=1}^{n} k(\cdot, z_i) \\
  C_z &= \frac{1}{n} \sum_{i=1}^{n} k(\cdot, z_i) \otimes k(\cdot, z_i) - \mu_z \otimes \mu_z
  \end{align*}
  \]

The empirical covariance captures **non-linear** features of the underlying distribution, e.g. Kernel PCA.
Kernel distance gradient

\[ g(x) = k(x, x) - 2k(x, y) - 2 \sum_{i=1}^{n} \beta_i [k(x, z_i) - \mu_z(x)] \]

\[ \nabla_x g(x)|_{x=y} = \nabla_x k(x, x)|_{x=y} - 2\nabla_x k(x, y)|_{x=y} - M_{z,y}H\beta \]

where \( M_{z,y} = 2 [\nabla_x k(x, z_1)|_{x=y}, \ldots, \nabla_x k(x, z_n)|_{x=y}] \) and \( H = I_n - \frac{1}{n}1_{n \times n} \)
Cost function $g$

g varies most along the high density regions of the target
Synthetic targets: Flower

**Flower**: $\mathcal{F}(r_0, A, \omega, \sigma)$, a $d$-dimensional target with:

$$
\mathcal{F}(x; r_0, A, \omega, \sigma) \propto \exp \left( - \frac{\sqrt{x_1^2 + x_2^2} - r_0 - A \cos (\omega \text{atan2} (x_2, x_1))}{2\sigma^2} \right)
\times \prod_{j=3}^{d} \mathcal{N}(x_j; 0, 1).
$$

Concentrates on $r_0$-circle with a periodic perturbation (with amplitude $A$ and frequency $\omega$) in the first two dimensions.
Synthetic targets: convergence statistics

8-dimensional $\mathcal{F}(10, 6, 6, 1)$ target;
iterations: 120000, burn-in: 60000