Kernel Adaptive Metropolis-Hastings

Dino Sejdinovic[⋆], Heiko Strathmann[⋆], Maria Lomeli Garcia[⋆], Christophe Andrieu[‡], and Arthur Gretton[⋆]

[⋆]Gatsby Unit, CSML, University College London,
[‡]School of Mathematics, University of Bristol

TU Berlin, 29 July 2014

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 $q(\cdot|x_t)$?

Metropolis-Hastings MCMC

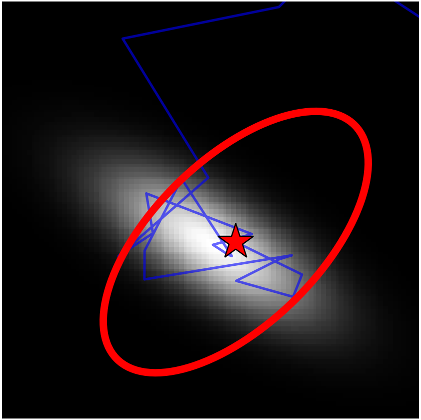
- Unnormalized target $\pi(x) \propto p(x)$
- Generate Markov chain with invariant distribution p
  - Initialize $x_0 \sim p_0$
  - At iteration $t \geq 0$, propose to move to state $x' \sim q(\cdot|x_t)$
  - Accept/Reject proposals based on ratio

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

- What proposal $q(\cdot|x_t)$?
  - Too narrow: small increments $\rightarrow$ slow convergence
  - Too broad: many rejections $\rightarrow$ slow convergence

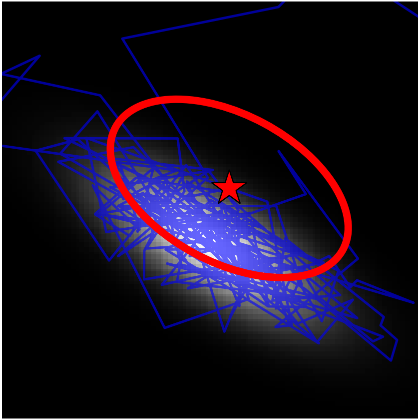
Adaptive MCMC

- **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



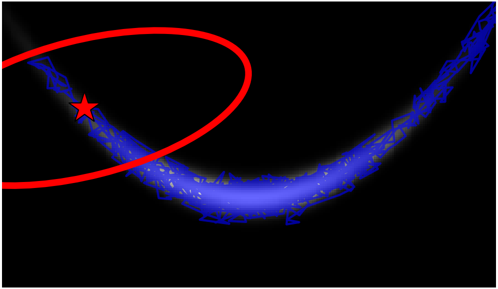
Adaptive MCMC

- **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** (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).

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).

Pseudo-Marginal MCMC

When is target not computable?

- Posterior inference, latent process $\mathbf{f}$

$$p(\theta|\mathbf{y}) \propto p(\theta)p(\mathbf{y}|\theta) = p(\theta)\int p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f},\theta)d\mathbf{f} =: \pi(\theta)$$

Pseudo-Marginal MCMC

When is target not computable?

- Posterior inference, latent process $\mathbf{f}$

$$p(\theta|\mathbf{y}) \propto p(\theta)p(\mathbf{y}|\theta) = p(\theta)\int p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f},\theta)d\mathbf{f} =: \pi(\theta)$$

- Cannot integrate out $\mathbf{f}$: e.g. Gaussian process classification, θ lengthscales of covariance. MH ratio:

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

Pseudo-Marginal MCMC

When is target not computable?

- Posterior inference, latent process **f**

$$p(\theta|\mathbf{y}) \propto p(\theta)p(\mathbf{y}|\theta) = p(\theta)\int p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f},\theta)d\mathbf{f} =: \pi(\theta)$$

- Cannot integrate out **f**: e.g. Gaussian process classification, θ lengthscales of covariance. MH ratio:

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

- Replace $p(\mathbf{y}|\theta)$ with Monte Carlo estimate $\hat{p}(\mathbf{y}|\theta)$

Pseudo-Marginal MCMC

When is target not computable?

- Posterior inference, latent process $\mathbf{f}$

$$p(\theta|\mathbf{y}) \propto p(\theta)p(\mathbf{y}|\theta) = p(\theta)\int p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f},\theta)d\mathbf{f} =: \pi(\theta)$$

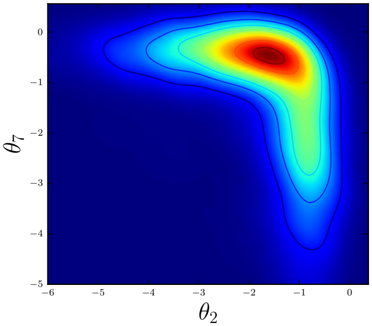
- Cannot integrate out $\mathbf{f}$: e.g. Gaussian process classification, θ lengthscales of covariance. MH ratio:

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

- Replace $p(\mathbf{y}|\theta)$ with Monte Carlo estimate $\hat{p}(\mathbf{y}|\theta)$
- Replacing marginal likelihood with *unbiased estimate* still results in correct invariant distribution (**Beaumont, 2003; Andrieu & Roberts, 2009**)

Intractable & Non-linear Target in GPC

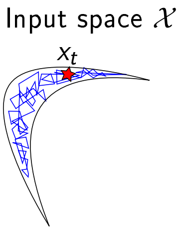
- Sliced posterior over hyperparameters of a Gaussian Process classifier on UCI Glass dataset obtained using Pseudo-Marginal MCMC



Adaptive sampler that learns the shape of non-linear targets without gradient information?

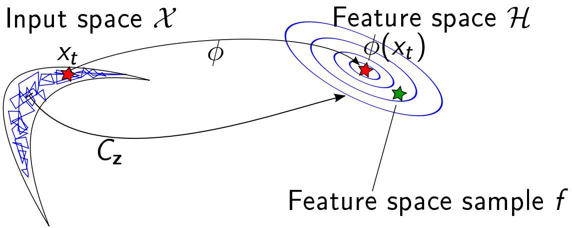
Use feature space covariance

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



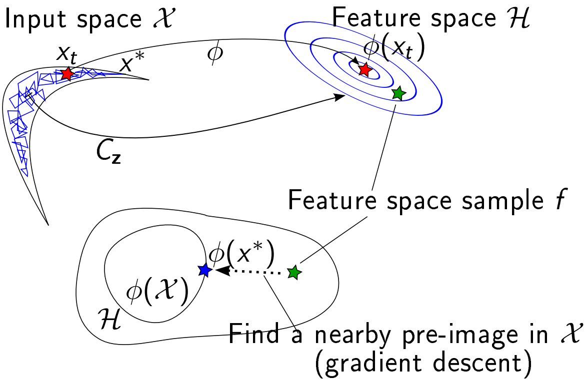
Use feature space covariance

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



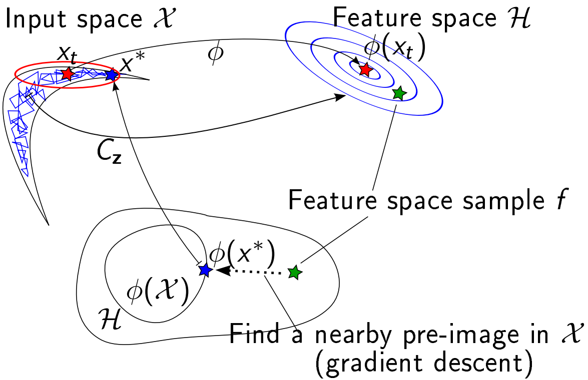
Use feature space covariance

- 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 $\mathcal{H}$



Proposal Construction Summary

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

Proposal Construction Summary

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

This gives:

$$x^* | x_t, f, \xi = x_t - \eta \nabla_x \left\| \phi\left(x\right) - f \right\|_{\mathcal{H}}^2 |_{x=x_t} + \xi$$

Proposal Construction Summary

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

This gives:

$$x^* | x_t, f, \xi = x_t - \eta \nabla_x \left\| \phi\left(x\right) - f \right\|_{\mathcal{H}}^2 |_{x=x_t} + \xi$$

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

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

$M_{\mathbf{z},x_t} = 2\left[\nabla_x k(x, z_1)|_{x=x_t}, \ldots, \nabla_x k(x, z_n)|_{x=x_t}\right]$,
$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$.

MCMC Kameleon

Input: unnormalized target π; subsample size n; scaling parameters ν, γ, kernel k;
update schedule $\{p_t\}_{t\geq 1}$ with $p_t \to 0$, $\sum_{t=1}^{\infty} p_t = \infty$

At iteration $t+1$,

1. With probability p_t, update a random subsample $\mathbf{z} = \{z_i\}_{i=1}^{n}$ of the chain history $\{x_i\}_{i=0}^{t-1}$,
2. Sample proposed point x^* from $q_{\mathbf{z}}(\cdot|x_t) = \mathcal{N}(x_t, \gamma^2 I_d + \nu^2 M_{\mathbf{z},x_t} H M_{\mathbf{z},x_t}^\top)$,
3. Accept/Reject with standard MH ratio:

$$x_{t+1} = \begin{cases} x^*, & \text{w.p. } \min\left\{1, \frac{\pi(x^*)q_{\mathbf{z}}(x_t|x^*)}{\pi(x_t)q_{\mathbf{z}}(x^*|x_t)}\right\}, \\ x_t, & \text{otherwise.} \end{cases}$$

MCMC Kameleon

Input: unnormalized target π; subsample size n; scaling parameters ν, γ, kernel k;
update schedule $\{p_t\}_{t\geq 1}$ with $p_t \to 0$, $\sum_{t=1}^{\infty} p_t = \infty$

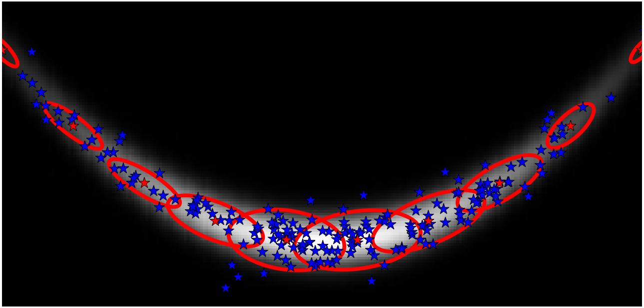
At iteration $t+1$,

1. With probability p_t, update a random subsample $\mathbf{z} = \{z_i\}_{i=1}^{n}$ of the chain history $\{x_i\}_{i=0}^{t-1}$,
2. Sample proposed point x^* from $q_{\mathbf{z}}(\cdot|x_t) = \mathcal{N}(x_t, \gamma^2 I_d + \nu^2 M_{\mathbf{z},x_t} H M_{\mathbf{z},x_t}^{\top})$,
3. Accept/Reject with standard MH ratio:

$$x_{t+1} = \begin{cases} x^*, & \text{w.p. } \min\left\{1, \frac{\pi(x^*)q_{\mathbf{z}}(x_t|x^*)}{\pi(x_t)q_{\mathbf{z}}(x^*|x_t)}\right\}, \\ x_t, & \text{otherwise.} \end{cases}$$

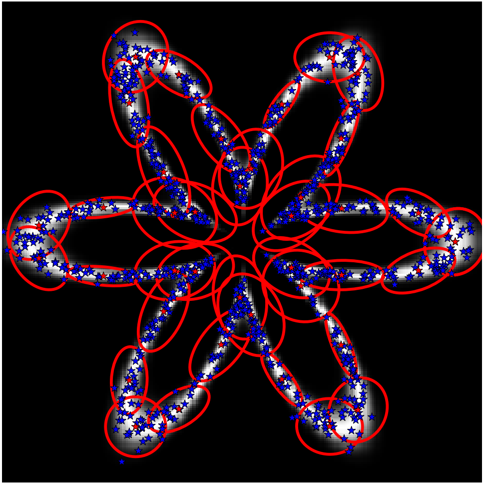
Convergence to target π preserved as long as $p_t \to 0$ (Roberts & Rosenthal, 2007).

Locally aligned covariance



Kameleon proposals capture local covariance structure

Locally aligned covariance



Examples of Covariance Structure for Standard Kernels

- **Linear kernel:** $k(x, x') = x^\top x'$

$$q_{\mathbf{z}}(\cdot|y) = \mathcal{N}(y, \gamma^2 I + 4\nu^2 \mathbf{Z}^\top H \mathbf{Z})$$

classical Adaptive Metropolis **Haario et al 1999;2001**.

Examples of Covariance Structure for Standard Kernels

- **Linear kernel:** $k(x, x') = x^\top x'$

$$q_{\mathbf{z}}(\cdot|y) = \mathcal{N}(y, \gamma^2 I + 4\nu^2 \mathbf{Z}^\top H \mathbf{Z})$$

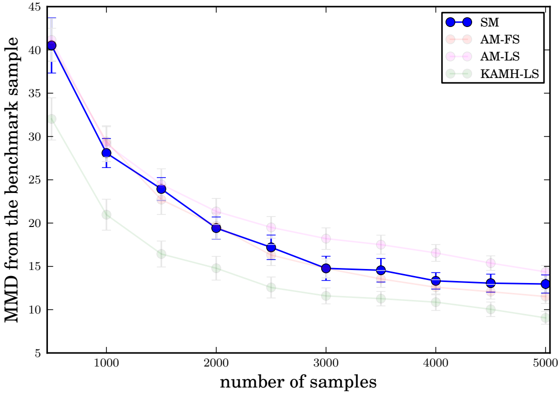
  classical Adaptive Metropolis Haario et al 1999;2001.

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

$$\begin{aligned}\left[\operatorname{cov}[q_{\mathbf{z}(\cdot|y)}]\right]_{ij} &= \gamma^2\delta_{ij} + \frac{4\nu^2}{\sigma^4}\sum_{a=1}^{n}[k(y, z_a)]^2(z_{a,i} - y_i)(z_{a,j} - y_j) \\ &+ \mathcal{O}\left(\frac{1}{n}\right).\end{aligned}$$

  Influence of previous points z_a on covariance is weighted by similarity $k(y, z_a)$ to current location y.

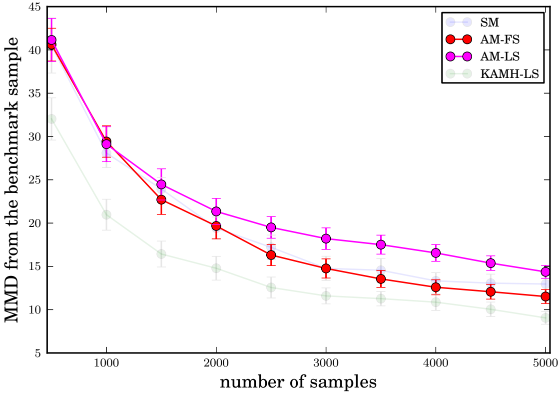
UCI Glass dataset



comparison in terms of all mixed moments up to order 3

8-dimensional non-linear posterior $p(\theta|\mathbf{y})$: no ground truth, performance with respect to a long-run, heavily thinned benchmark sample.

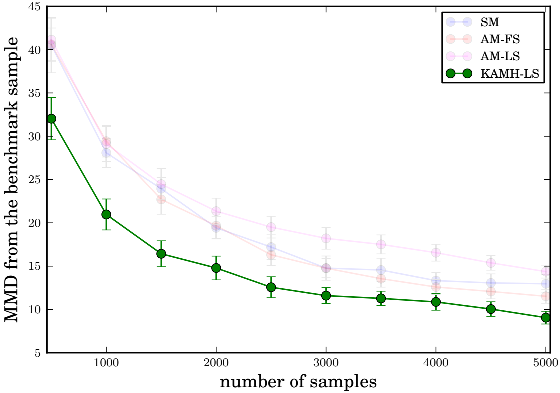
UCI Glass dataset



comparison in terms of all mixed moments up to order 3

8-dimensional non-linear posterior $p(\theta|\mathbf{y})$: no ground truth, performance with respect to a long-run, heavily thinned benchmark sample.

UCI Glass dataset

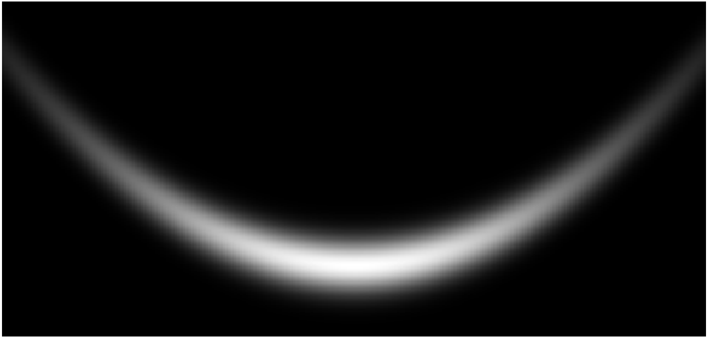


comparison in terms of all mixed moments up to order 3

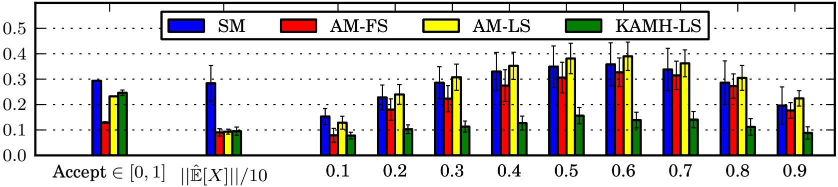
8-dimensional non-linear posterior $p(\theta|\mathbf{y})$: no ground truth, performance with respect to a long-run, heavily thinned benchmark sample.

Synthetic targets: Banana

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



Synthetic targets: convergence statistics



Strongly twisted 8-dimensional $\mathcal{B}(0.1, 100)$ target; iterations: 80000, burn-in: 40000

Conclusions

- A simple, versatile, gradient-free adaptive MCMC sampler
- Proposals automatically conform to the local covariance structure of the target distribution at the current chain state
- Outperforms existing approaches on nonlinear target distributions
- Future directions: tradeoff between the sub-sampling and convergence; samplers on non-Euclidean domains

- **code: https://github.com/karlnapf/kameleon-mcmc**

Bayesian Gaussian Process Classification

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

$$p(\mathbf{f}, \mathbf{y}, \theta) = p(\theta)p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f})$$

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

Bayesian Gaussian Process Classification

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

$$p(\mathbf{f}, \mathbf{y}, \theta) = p(\theta)p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f})$$

  where $\mathbf{f}|\theta \sim \mathcal{N}(0, \mathcal{K}_\theta)$ is a realization of a GP with covariance $\mathcal{K}_\theta$ (covariance between latent processes evaluated at X).
- $\mathcal{K}_\theta$: exponentiated quadratic Automatic Relevance Determination (ARD) covariance:

$$(\mathcal{K}_\theta)_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}'_j|\theta) = \exp\left(-\frac{1}{2}\sum_{s=1}^{d}\frac{(x_{i,s} - x'_{j,s})^2}{\exp(\theta_s)}\right)$$

Bayesian Gaussian Process Classification

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

$$p(\mathbf{f}, \mathbf{y}, \theta) = p(\theta)p(\mathbf{f}|\theta)p(\mathbf{y}|\mathbf{f})$$

  where $\mathbf{f}|\theta \sim \mathcal{N}(0, \mathcal{K}_\theta)$ is a realization of a GP with covariance $\mathcal{K}_\theta$ (covariance between latent processes evaluated at X).
- $\mathcal{K}_\theta$: exponentiated quadratic Automatic Relevance Determination (ARD) covariance:

$$(\mathcal{K}_\theta)_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}'_j|\theta) = \exp\left(-\frac{1}{2}\sum_{s=1}^{d}\frac{(x_{i,s} - x'_{j,s})^2}{\exp(\theta_s)}\right)$$

- $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i)$ is a product of sigmoidal functions:

$$p(y_i|f_i) = \frac{1}{1 - \exp(-y_i f_i)}, \qquad y_i \in \{-1, 1\}.$$

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 $p(\theta, \mathbf{f}|y)$, which samples from $p(\mathbf{f}|\theta, y)$ and $p(\theta|\mathbf{f}, y)$ in turns, since $p(\theta|\mathbf{f}, y)$ is extremely sharp

Bayesian Gaussian Process Classification (2)

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

Bayesian Gaussian Process Classification (2)

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

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

Bayesian Gaussian Process Classification (2)

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

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

- 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.

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 $C_P : \mathcal{H}_k \to \mathcal{H}_k$ such that $\forall f, g \in \mathcal{H}_k$, $\langle f, C_P g \rangle_{\mathcal{H}_k} = \text{Cov}_P\left[f(X)g(X)\right]$.

Covariance operator

Definition

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

- Covariance operator: $C_P : \mathcal{H}_k \to \mathcal{H}_k$ is given by $C_P = \int k(\cdot,x) \otimes k(\cdot,x)\, dP(x) - \mu_P \otimes \mu_P$ (covariance of canonical features)
- Empirical versions of embedding and the covariance operator:

$$\mu_{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} k(\cdot, z_i) \qquad C_{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} k(\cdot, z_i) \otimes k(\cdot, z_i) - \mu_{\mathbf{z}} \otimes \mu_{\mathbf{z}}$$

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

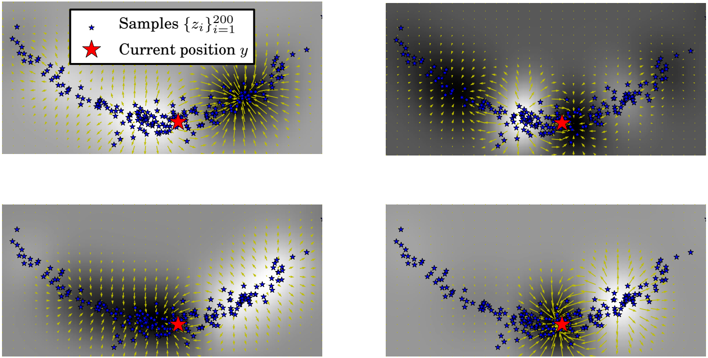
Kernel distance gradient

$$g(x) = k(x,x) - 2k(x,y) - 2\sum_{i=1}^{n} \beta_i \left[k(x,z_i) - \mu_{\mathbf{z}}(x)\right]$$

$$\nabla_x g(x)|_{x=y} = \underbrace{\nabla_x k(x,x)|_{x=y} - 2\nabla_x k(x,y)|_{x=y}}_{=0} - M_{\mathbf{z},y} H \beta$$

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

Cost function g

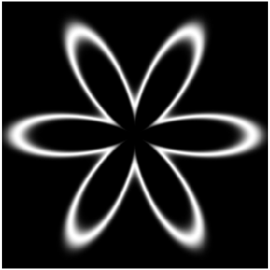


g varies most along the high density regions of the target

Synthetic targets: Flower

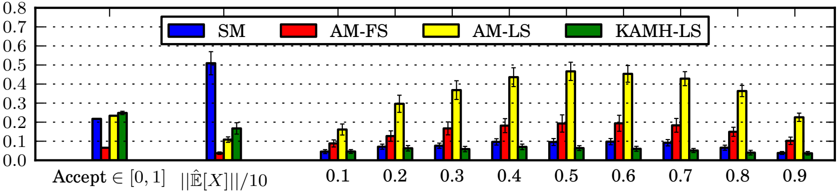
Flower: $\mathcal{F}(r_0, A, \omega, \sigma)$, a d-dimensional target with:

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



Concentrates on r_0-circle with a periodic perturbation (with amplitude A and frequency ω) in the first two dimensions.

Synthetic targets: convergence statistics



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