MCMC Kameleon: Kernel Adaptive Metropolis-Hastings

Dino Sejdinovic*
joint work with: Heiko Strathmann*, Maria Lomeli Garcia*, Christophe Andrieu†, and Arthur Gretton*

*Gatsby Unit, CSML, University College London,
†School of Mathematics, University of Bristol

Kernel Methods for Big Data, Lille
31 March 2014
arXiv: 1307.5302

Overview

Outline

Metropolis-Hastings MCMC

- Access to unnormalized target $\pi(x) \propto P(x)$
- Generate a Markov chain with P as invariant distribution
  - 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 the MH acceptance ratio (preserves detailed balance)

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

MCMC proposals

- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?

MCMC proposals

- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?
  - Variance of the proposal is too small:
    small increments $\rightarrow$ slow convergence

MCMC proposals

- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?
    - Variance of the proposal is too small:
      small increments $\rightarrow$ slow convergence
    - Variance of the proposal is too large:
      too many rejections $\rightarrow$ slow convergence

MCMC proposals

- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?
  - Variance of the proposal is too small:
    small increments $\rightarrow$ slow convergence
  - Variance of the proposal is too large:
    too many rejections $\rightarrow$ slow convergence
- In high dimensions: very different scalings along different principal directions

MCMC proposals

- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?
  - Variance of the proposal is too small:
    small increments $\rightarrow$ slow convergence
  - Variance of the proposal is too large:
    too many rejections $\rightarrow$ slow convergence
- In high dimensions: very different scalings along different principal directions
- Gelman, Roberts & Gilks, 1996: in random walk Metropolis with proposals $\mathcal{N}(0, \Sigma)$ on a product target π (independent dimensions):
  - $\Sigma = \frac{2.38^2}{d}\Sigma_\pi$ is shown to be asymptotically optimal as $d \rightarrow \infty$
  - asymptotically optimal acceptance rate of 0.234

MCMC proposals

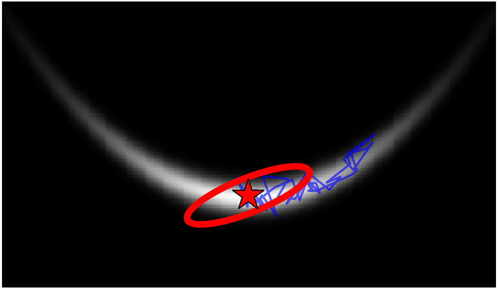
- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?
  - Variance of the proposal is too small:
    small increments $\rightarrow$ slow convergence
  - Variance of the proposal is too large:
    too many rejections $\rightarrow$ slow convergence
- In high dimensions: very different scalings along different principal directions
- **Gelman, Roberts & Gilks, 1996**: in random walk Metropolis with proposals $\mathcal{N}(0, \Sigma)$ on a product target π (independent dimensions):
  - $\Sigma = \frac{2.38^2}{d}\Sigma_\pi$ is shown to be asymptotically optimal as $d \rightarrow \infty$
  - asymptotically optimal acceptance rate of 0.234
- Σ_π unknown

MCMC proposals

- What proposal $q(\cdot|x_t)$ to use in Metropolis-Hastings algorithms?
  - Variance of the proposal is too small:
    small increments $\rightarrow$ slow convergence
  - Variance of the proposal is too large:
    too many rejections $\rightarrow$ slow convergence
- In high dimensions: very different scalings along different principal directions
- Gelman, Roberts & Gilks, 1996: in random walk Metropolis with proposals $\mathcal{N}(0,\Sigma)$ on a product target π (independent dimensions):
  - $\Sigma = \frac{2.38^2}{d}\Sigma_\pi$ is shown to be asymptotically optimal as $d \rightarrow \infty$
  - asymptotically optimal acceptance rate of 0.234
- Σ_π unknown
- Simple and often effective as rules of thumb, but based on assumptions not valid for complex targets

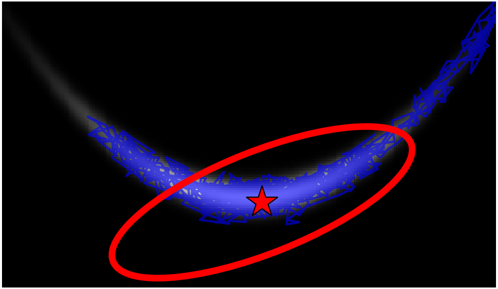
Adaptive MCMC

- **Adaptive MCMC** (Haario, Saksman & Tamminen, 2001): use history of Markov chain to learn covariance Σ_π of target π, i.e., scaling in principal directions



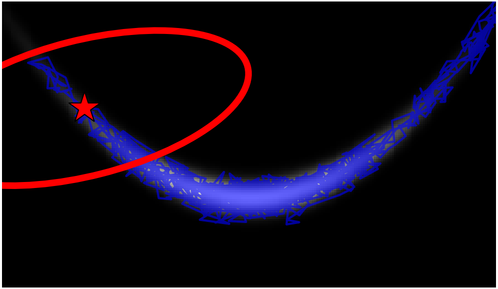
Adaptive MCMC

- **Adaptive MCMC** (Haario, Saksman & Tamminen, 2001): use history of Markov chain to learn covariance Σ_π of target π, i.e., scaling in principal directions



Adaptive MCMC

- **Adaptive MCMC** (Haario, Saksman & Tamminen, 2001): use history of Markov chain to learn covariance Σ_π of target π, i.e., scaling in principal directions
- May be locally miscalibrated for strongly non-linear targets: directions of large variance depend on the current location



Motivation: Intractable & Non-linear Targets

- Non-linear targets: Hamiltonian Monte Carlo (HMC) or Metropolis Adjusted Langevin Algorithms (MALA) (Roberts & Stramer, 2003; Girolami & Calderhead, 2011).

Motivation: Intractable & Non-linear Targets

- Non-linear targets: Hamiltonian Monte Carlo (HMC) or Metropolis Adjusted Langevin Algorithms (MALA) (Roberts & Stramer, 2003; Girolami & Calderhead, 2011).
- However, those depend on gradients of the target and second order information – often unavailable or expensive to compute.

Motivation: Intractable & Non-linear Targets

- Non-linear targets: Hamiltonian Monte Carlo (HMC) or Metropolis Adjusted Langevin Algorithms (MALA) (Roberts & Stramer, 2003; Girolami & Calderhead, 2011).
- However, those depend on gradients of the target and second order information – often unavailable or expensive to compute.
- Extreme case: not even target can be computed – **Pseudo-Marginal MCMC** (Beaumont, 2003; Andrieu & Roberts, 2009).

Outline

Pseudo-Marginal MCMC

- Missing data: parameters θ, latent process $\mathbf{f}$, observations $\mathbf{y}$ with

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

Pseudo-Marginal MCMC

- Missing data: parameters θ, latent process $\mathbf{f}$, observations $\mathbf{y}$ with

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

- Interested in posterior

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

Pseudo-Marginal MCMC

- Missing data: parameters θ, latent process $\mathbf{f}$, observations $\mathbf{y}$ with

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

- Interested in posterior

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

- Often impossible to integrate out the latent process $\mathbf{f}$, i.e., unable to compute marginal likelihood $p(\mathbf{y}|\theta)$

Pseudo-Marginal MCMC (2)

- Unable to compute correct Metropolis-Hasting acceptance probabilities:

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

Pseudo-Marginal MCMC (2)

- Unable to compute correct Metropolis-Hasting acceptance probabilities:

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

- However, we can often obtain an unbiased Monte Carlo estimate of $p(\mathbf{y}|\theta)$, e.g., by importance sampling

Pseudo-Marginal MCMC (2)

- Unable to compute correct Metropolis-Hasting acceptance probabilities:

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

- However, we can often obtain an unbiased Monte Carlo estimate of $p(\mathbf{y}|\theta)$, e.g., by importance sampling
- Remarkably, replacing the marginal likelihood with its unbiased estimate still results in the correct invariant distribution (Andrieu & Roberts, 2009)

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:

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

Bayesian Gaussian Process Classification (2)

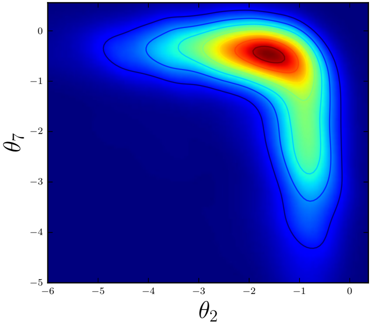
- Fully Bayesian treatment: Interested in the posterior $p(\theta|y)$
- Cannot use a Gibbs sampler on $p(\theta, \mathbf{f}|y)$, which samples from $p(\mathbf{f}|\theta, y)$ and $p(\theta|\mathbf{f}, y)$ in turns, since $p(\theta|\mathbf{f}, y)$ is extremely sharp
- **Filippone & Girolami, 2013** use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

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

- No access to likelihood, gradient, or Hessian of the target.

Intractable & Non-linear Target in GPC

- Sliced posterior over hyperparameters of a GP classifier (where target cannot be computed) on UCI Glass dataset (classification of window against non-window glass)

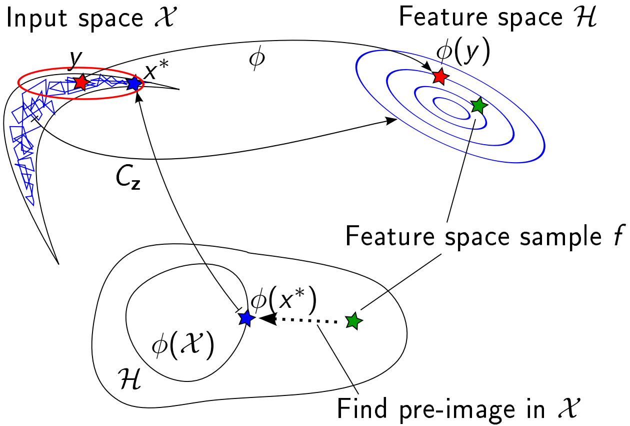


Adaptive sampler that learns the shape of non-linear targets without higher order information?

Outline

Use feature space covariance?

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



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

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:

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

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

Kernel Adaptive Metropolis Hastings: Construction

Target π on $\mathbb{R}^d$; Current chain state: y

Step 1: Obtain a subsample of the Markov chain history: $\mathbf{z} = \{z_i\}_{i=1}^n$, this induces empirical RKHS embedding and covariance:

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

Kernel Adaptive Metropolis Hastings: Construction (2)

Target π on $\mathbb{R}^d$; Current chain state: y, a subsample of the Markov chain history: $\mathbf{z} = \{z_i\}_{i=1}^n$

Step 2: Sample from the Gaussian Measure $\mathcal{N}(\mu_{\mathbf{z}}, \nu^2 C_{\mathbf{z}})$ on RKHS: it suffices to generate $\beta \sim \mathcal{N}(0, \frac{\nu^2}{n} I_n)$, then

$$f = k(\cdot, y) + \sum_{i=1}^{n} \beta_i [k(\cdot, z_i) - \mu_{\mathbf{z}}]$$

has the correct covariance structure.

Kernel Adaptive Metropolis Hastings: Construction (2)

$$
\begin{aligned}
&\mathbb{E}\left[(f-k(\cdot,y))\otimes(f-k(\cdot,y))\right]\\
&=\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j\left(k(\cdot,z_i)-\mu_{\mathbf{z}}\right)\otimes\left(k(\cdot,z_j)-\mu_{\mathbf{z}}\right)\right]\\
&=\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\left[\beta_i\beta_j\right]\left(k(\cdot,z_i)-\mu_{\mathbf{z}}\right)\otimes\left(k(\cdot,z_j)-\mu_{\mathbf{z}}\right)\\
&=\frac{\nu^2}{n}\sum_{i=1}^{n}\left(k(\cdot,z_i)-\mu_{\mathbf{z}}\right)\otimes\left(k(\cdot,z_i)-\mu_{\mathbf{z}}\right)\\
&=\nu^2 C_{\mathbf{z}}
\end{aligned}
$$

Kernel Adaptive Metropolis Hastings: Construction (3)

Target π on $\mathbb{R}^d$; Current chain state: y, a subsample of the Markov chain history: $\mathbf{z} = \{z_i\}_{i=1}^n$, RKHS sample $f \sim \mathcal{N}(\mu_\mathbf{z}, \nu^2 C_\mathbf{z})$

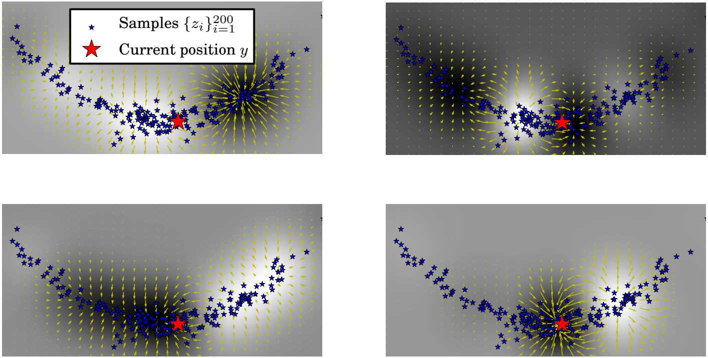
Step 3: Find a point x^* in input space $\mathcal{X}$ with feature $k(\cdot, x^*)$ close to f:

$$\operatorname{argmin}_{x\in\mathcal{X}} \|k(\cdot,x) - f\|_{\mathcal{H}}^2 =$$

$$\operatorname{argmin}_{x\in\mathcal{X}} \left\{ \underbrace{k(x,x) - 2k(x,y) - 2\sum_{i=1}^{n} \beta_i \left[k(x,z_i) - \mu_\mathbf{z}(x)\right]}_{=:g(x) \text{ where } g:\mathcal{X}\to\mathbb{R}} \right\}.$$

Take a single gradient step from y w.r.t g, and (optionally) add 'exploration term' $\xi \sim \mathcal{N}(0, \gamma^2 I_d)$.

Cost function g



g varies most along the high densoty regions of the target

Construction Summary

1. Get a chain subsample $\mathbf{z} = \{z_i\}_{i=1}^n$
2. Construct $f \sim \mathcal{N}(\mu_{\mathbf{z}}, \nu^2 C_{\mathbf{z}})$ – represented by $\beta \sim \mathcal{N}(0, \frac{\nu^2}{n} I_n)$
3. Find x^* close to f and add 'exploration term' $\xi \sim \mathcal{N}(0, \gamma^2 I_d)$.

Construction Summary

1. Get a chain subsample $\mathbf{z} = \{z_i\}_{i=1}^n$
2. Construct $f \sim \mathcal{N}(\mu_{\mathbf{z}}, \nu^2 C_{\mathbf{z}})$ – represented by $\beta \sim \mathcal{N}(0, \frac{\nu^2}{n} I_n)$
3. Find x^* close to f and add 'exploration term' $\xi \sim \mathcal{N}(0, \gamma^2 I_d)$.

This gives:

$$x^* | y, \beta, \xi = y - \eta \nabla_x g(x)|_{x=y} + \xi = y - M_{\mathbf{z},y} H \beta + \xi,$$

where $M_{\mathbf{z},y} = 2\eta \left[\nabla_x k(x, z_1)|_{x=y}, \ldots, \nabla_x k(x, z_n)|_{x=y}\right]$ is based on kernel gradients (readily available).

Construction Summary

1. Get a chain subsample $\mathbf{z} = \{z_i\}_{i=1}^n$
2. Construct $f \sim \mathcal{N}(\mu_{\mathbf{z}}, \nu^2 C_{\mathbf{z}})$ – represented by $\beta \sim \mathcal{N}(0, \frac{\nu^2}{n} I_n)$
3. Find x^* close to f and add 'exploration term' $\xi \sim \mathcal{N}(0, \gamma^2 I_d)$.

This gives:

$$x^*|y, \beta, \xi = y - \eta \nabla_x g(x)|_{x=y} + \xi = y - M_{\mathbf{z},y} H \beta + \xi,$$

where $M_{\mathbf{z},y} = 2\eta \left[\nabla_x k(x, z_1)|_{x=y}, \ldots, \nabla_x k(x, z_n)|_{x=y}\right]$ is based on kernel gradients (readily available).

We can integrate out RKHS samples and gradient step (i.e., β and ξ) and obtain a marginal Gaussian proposal on the input space:

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

MCMC Kameleon

Input: unnormalized target π; subsample size n; scaling parameters ν, γ, kernel k; update schedule $\{\delta_t\}$
At iteration $t+1$,

1. With probability δ_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

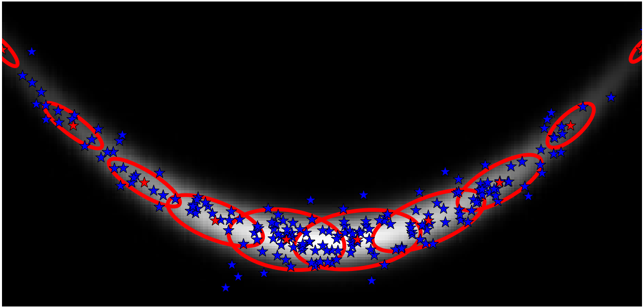
Input: unnormalized target π; subsample size n; scaling parameters ν, γ, kernel k; update schedule $\{\delta_t\}$
At iteration $t+1$,

1. With probability δ_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:

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

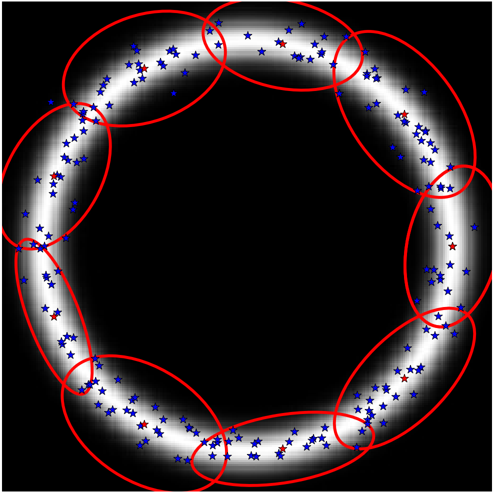
Convergence to target π preserved as long as $\delta_t \to 0$.

Locally aligned covariance

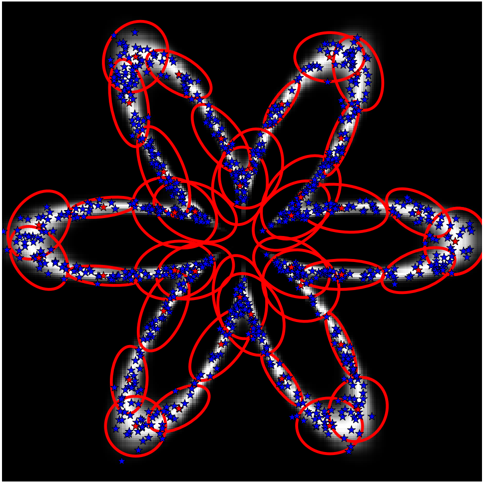


Kameleon proposals capture local covariance structure

Locally aligned covariance



Locally aligned covariance



Examples of Covariance Structure for Standard Kernels

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

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

which results in the classical Adaptive Metropolis of **Haario et al 1999;2001**.

Examples of Covariance Structure for Standard Kernels

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

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

  which results in the classical Adaptive Metropolis of Haario et al 1999;2001.

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

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

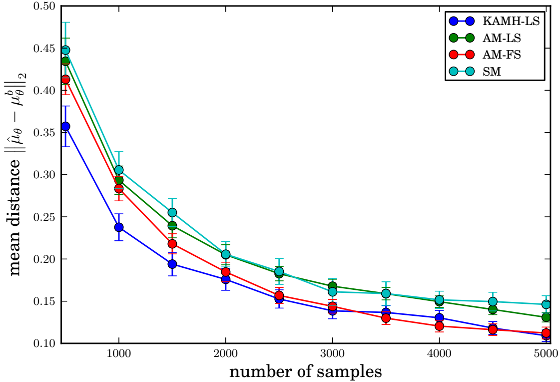
The influence of the previous points z_a on the covariance is weighted by their similarity $k(y, z_a)$ to the current location y.

Outline

Setup

- (**SM**) Standard Metropolis with the isotropic proposal $q(\cdot|y) = \mathcal{N}(y, \nu^2 I)$ and scaling $\nu = 2.38/\sqrt{d}$
- (**AM-FS**) Adaptive Metropolis with a learned covariance matrix and fixed global scaling $\nu = 2.38/\sqrt{d}$
- (**AM-LS**) Adaptive Metropolis with a learned covariance matrix and global scaling ν learned to bring the acceptance rate close to $\alpha^* = 0.234$
- (**KAMH-LS**) MCMC Kameleon with the global scaling ν learned to bring the acceptance rate close to $\alpha^* = 0.234$

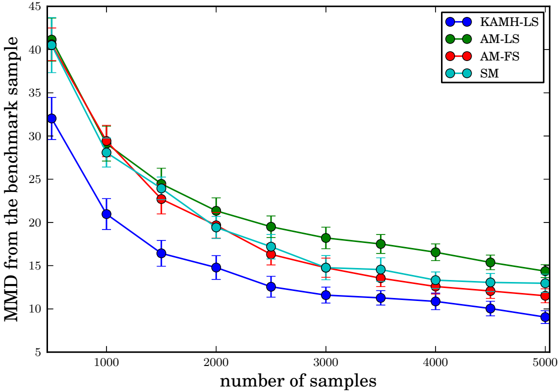
UCI Glass dataset



mean comparison

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

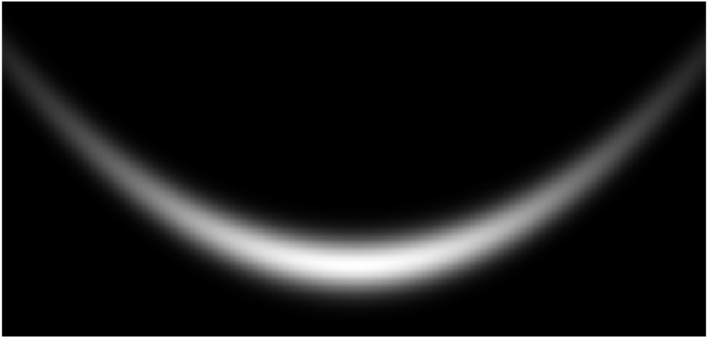


comparison in terms of all mixed moments up to order 3

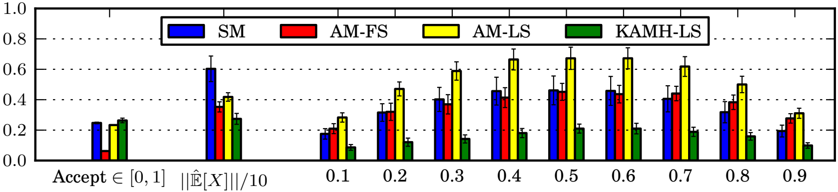
8-dimensional non-linear posterior $p(\theta|\mathbf{y})$: no ground truth, performance with respect to a long-run, heavily thinned benchmark sample.

Synthetic targets

Banana: $\mathcal{B}(b, v)$: 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)

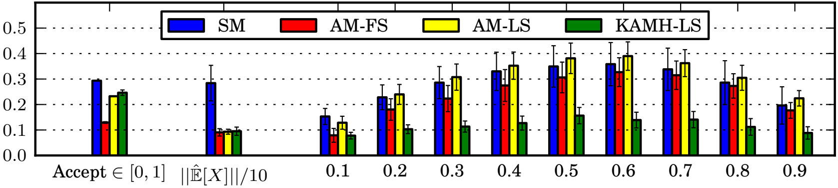


Synthetic targets: convergence statistics



Moderately twisted 8-dimensional $\mathcal{B}(0.03, 100)$ target; iterations: 40000, burn-in: 20000

Synthetic targets: convergence statistics

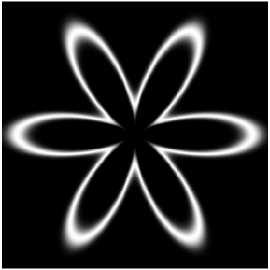


Strongly twisted 8-dimensional $\mathcal{B}(0.1, 100)$ target; iterations: 80000, burn-in: 40000

Synthetic targets

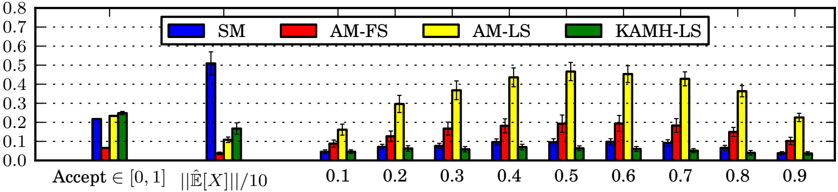
Flower: $\mathcal{F}(r_0, A, \omega, \sigma)$, a d-dimensional target with:

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



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

Conclusions

- A simple, versatile, gradient-free adaptive MCMC sampler

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

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

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

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

- **preprint: http://arxiv.org/abs/1307.5302**
- **code: https://github.com/karlnapf/kameleon-mcmc**