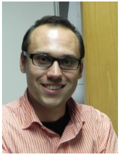
Kernel methods for adaptive Monte Carlo

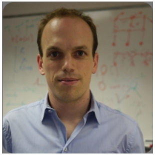
Heiko Strathmann

Gatsby Unit, UCL London

Greek stochastics θ, 10th July 2016

Joint work





Metropolis Hastings transition kernel

Target $\pi(\theta) \propto p(\theta|\mathcal{D})$

- At iteration $j+1$, state $\theta_{(j)}$
- Propose $\theta' \sim q\left(\theta|\theta_{(j)}\right)$
- Accept $\theta_{(j+1)} \leftarrow \theta'$ with probability

$$\min\left(\frac{\pi(\theta')}{\pi(\theta_{(j)})} \times \frac{q(\theta_{(j)}|\theta')}{q(\theta'|\theta_{(j)})}, 1\right)$$

- Reject $\theta_{(j+1)} \leftarrow \theta_{(j)}$ otherwise.

How to choose q when faced with intractable targets?

Intractable target – running example

Gaussian process classification model on $\{(x_i, y_i)\}_{i=1}^n$

- latent process response $\mathbf{f} \in \mathbb{R}^n$ where $\mathbf{f}_i = f(x_i)$
- labels $\mathcal{D} = \mathbf{y} \in \{-1, 1\}^n$
- hyper-parameters θ

Joint distribution

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

- $\mathbf{f}|\theta \sim \mathcal{N}(\mathbf{0}, \mathcal{K}_\theta)$ with covariance matrix $\mathcal{K}_\theta$
- $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n p(y_i|f_i)$ is a product of sigmoidal functions

Intractable target – running example

- Interested in posterior parameters

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

  c.f. Filippone & Girolami (2014), Murray & Adams (2011)

- Unbiased estimate via importance sampling:

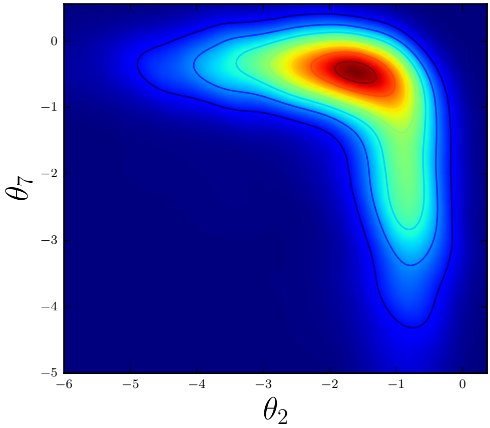
$$p(\mathbf{y}|\theta) \approx \frac{1}{n_{\mathrm{imp}}}\sum_{i=1}^{n_{\mathrm{imp}}}\frac{p(\mathbf{y}|\mathbf{f}^{(i)})p(\mathbf{f}^{(i)}|\theta)}{Q(\mathbf{f}^{(i)})}$$

  with $\mathbf{f}^{(i)} \sim Q(\mathbf{f})$, which is obtained via e.g. EP

- Instance of pseudo-marginal MCMC
  [Beaumont, 2003], [Andrieu & Roberts, 2009], ...
  [Lyne et. al 2015]

No access to likelihood, gradient, or Hessian of $p(\theta|y)$

Intractable target – running example



Induces nonlinear posterior on standard classification tasks

Learning covariance

- [Haario et al., 1999] learn covariance on the fly
- Given Markov chain at state $\theta_{(t)}$, then for $\lambda_t \in (0, 1)$, set

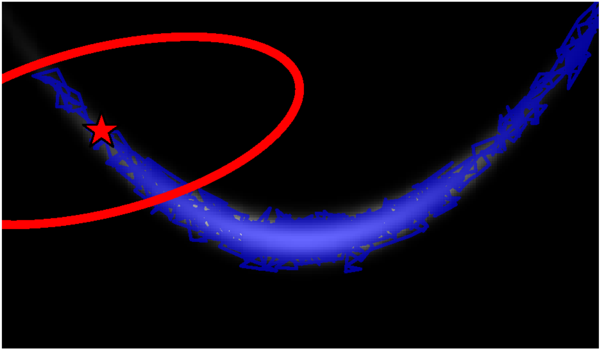
  $$\Sigma_t = (1 - \lambda_t)\Sigma_{t-1} + \lambda_t \left(\theta_{(t)}\theta_{(t)}^\top\right)$$

  and use proposal

  $$q(\cdot|\theta_{(t)}) = \mathcal{N}(\cdot|\theta_{(t)}, \Sigma_t)$$

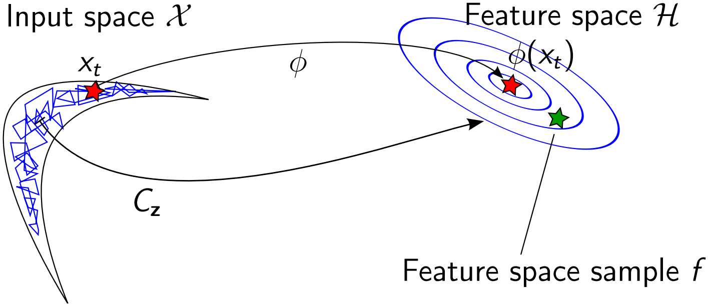
- Careful when q depends on $\{\theta_{(i)}\}_{i \le t}$
- Can choose λ_t s.t. $\Sigma_t \to \mathrm{Cov}(\pi)$ as $t \to \infty$ under some assumptions on π [Andrieu, 2008]

Adaptive Metropolis [Haario et al., 1999]

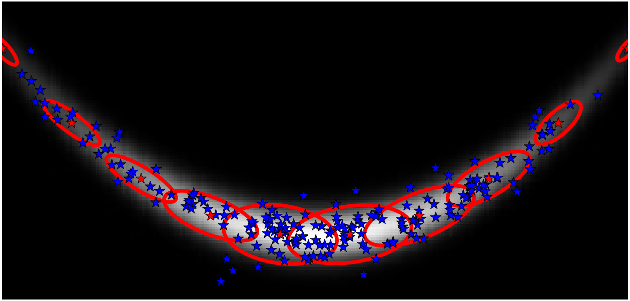


Improves mixing but is
locally miscalibrated for strongly nonlinear targets

Learning kernel covariance [Sejdinovic et al., 2012]



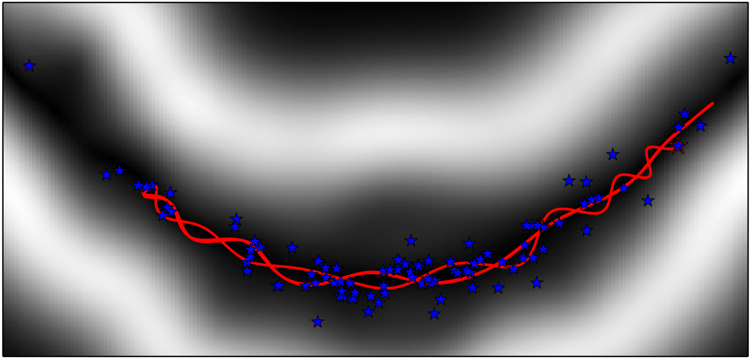
The Kameleon [Sejdinovic et al., 2012]



Learned kernel covariance allows to

- propose locally aligned moves
- improved mixing on nonlinear targets
- without the need for gradients

This talk: learning gradients



Gradients allow to

- propose distant moves with
- high acceptance probability
- in high dimensions

⇒significant mixing improvements

Hamiltonian dynamics 101

- Potential energy $U(q) = -\log \pi(q)$
- Momentum $p \sim \exp(-K(p))$, $K(p) = -\frac{1}{2}p^\top p$
- Hamiltonian

$$H(p, q) := K(p) + U(q)$$

- H-Flow is map

$$\phi_t^H : (p, q) \mapsto (p^*, q^*)$$

  s.t. $H(p^*, q^*) = H(p, q) \; \forall t$
- Acceptance probability along flow is 1.
- Generated by operator:

$$\frac{\partial K}{\partial p}\frac{\partial}{\partial q} - \frac{\partial U}{\partial q}\frac{\partial}{\partial p}$$

Exponential families in kernel spaces

- Need a surrogate density model to model gradient
- Kameleon used Gaussian in RKHS $\mathcal{H}$
- Here: exponential family [Sriperumbudur at al., 2014]

$$\pi(\theta) \approx \exp\left(\underbrace{\langle f, k(\theta, \cdot)\rangle_{\mathcal{H}}}_{=f(\theta)} - A(f)\right)$$

- For certain k, dense in probability densities (KL, TV, ...)
- Crux: fitting – normalising constant $A(f)$ is intractable

$$A(f) = \log \int \exp(f(\theta)) d\theta$$

- Maximum likelihood ill-posed, c.f. [Fukumizu, 2006]

Score matching [Hyvärien, 2005]

- Instead of ML, minimise Fisher divergence

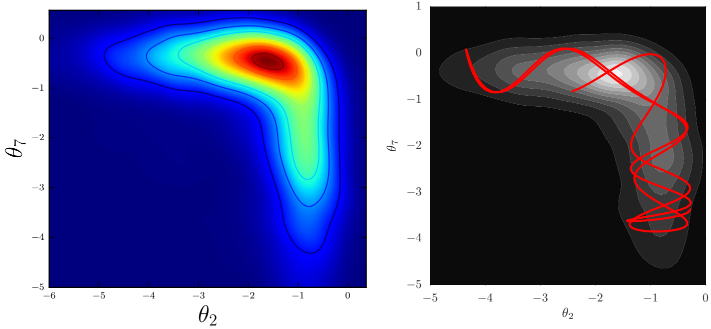
$$\operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{2} \int \pi(\theta) \left\| \nabla_\theta f(\theta) - \nabla_\theta \log \pi(\theta) \right\|_2^2 \, d\theta$$

- Intuition: match gradients in high density regions
- Remarkable: can rewrite and estimate from $\{\theta_i\}_{i=1}^n \sim \pi$

$$\operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sum_{\ell=1}^{d} \left[\frac{\partial^2 f(\theta)}{\partial \theta_\ell^2} + \frac{1}{2} \left(\frac{\partial f(\theta)}{\partial \theta_\ell} \right)^2 \right]$$

- Can be minimised in closed form. Reduces to regression.
- In practice much more robust than KDE.

Hamiltonian moves without gradients



Kernel induced Hamiltonian flow:

$$\frac{\partial K}{\partial p}\frac{\partial}{\partial q} - \frac{\partial f}{\partial q}\frac{\partial}{\partial p}$$

Kernel HMC [Strathmann et al., 2015]

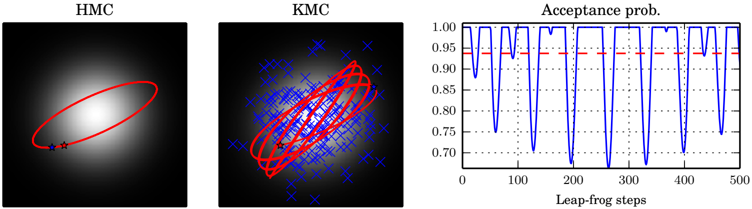
Start as random walk, transition to HMC.

Every iteration:

- Learn/update gradient model using past trajectory
- Use surrogate gradient to simulate Hamiltonian dynamics
- Correction for simulation error and gradient error:
  MH accept/reject step using estimator for π
- Stop adapting eventually

$\Rightarrow$Asymptotically correct, given a certain setup.

Computational considerations



- Bad fit ⇒ low acceptance rate ⇒ inefficient. But...

- Gradient model expensive to fit to Markov chain $\{\theta_i\}_{i=1}^t$:
  - $\mathcal{O}(t^3 d^3)$ time
  - $\mathcal{O}(t^2 d^2)$ memory
- Markov chain trajectory length t grows
- Aim is 'high' dimension d

One approximation: KMC Lite

$$f(\theta) = \sum_{i=1}^{n} \alpha_i k(z_i, \theta)$$

- $\{z_i\}_{i=1}^{n} \subseteq \{\theta_i\}_{i=1}^{t}$ sub-sample
- $\alpha \in \mathbb{R}^n$ from

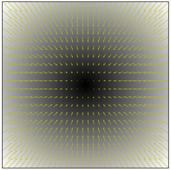
$$\hat{\alpha}_\lambda = -\frac{\sigma}{2}(C + \lambda I)^{-1} b$$

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

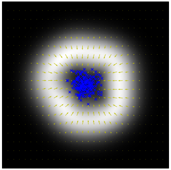
- Cost $\mathcal{O}(n^3 + n^2 d)$ (modulo low-rank, CG).

Geometrically ergodic on log-concave targets.

Gradient norm: Gaussian



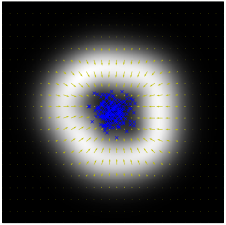
KMC Lite



Geometric ergodicity intuition

MCMC chain visits 'interesting' parts

- geometrically fast
- in particular when initialised in tails
- means: same guarantees as RWM



Proof idea

- In KMC lite, we have for $\|q\| \to \infty$

$$f(q) = \sum_{i=1}^{n} \alpha_i k(z_i, q) = \exp(-\|z_i - q\|) \to 0$$

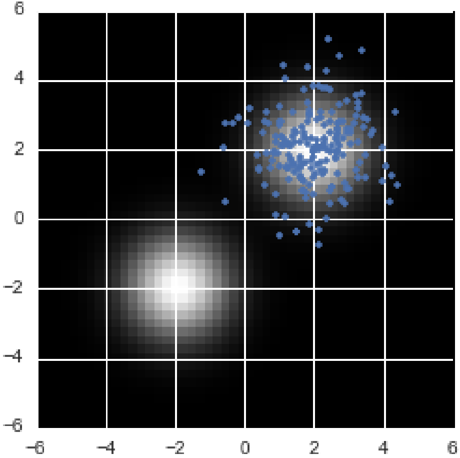
- Recall kernel H-flow is generated as

$$\frac{\partial K}{\partial p}\frac{\partial}{\partial q} - \frac{\partial f}{\partial q}\frac{\partial}{\partial p}$$

- KMC lite falls back to random walk, which is geo. erg.

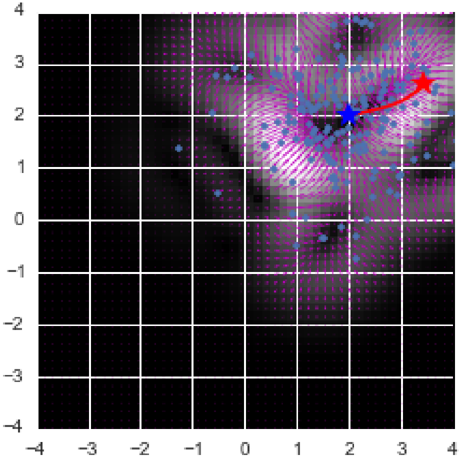
Why do we care?

Early adaptation stopping is potentially harmful...
But we need to for asymptotic correctness!



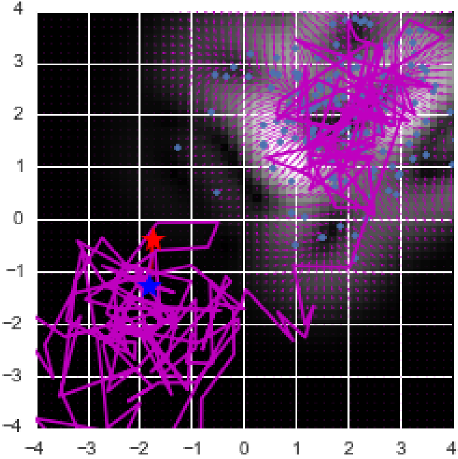
Why do we care?

Imagine we stopped adaptation early...
with a bad fit.



Why do we care?

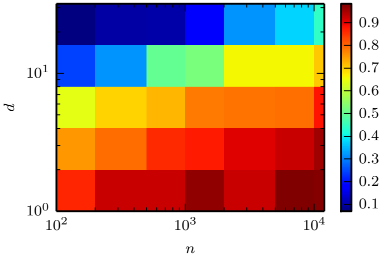
KMC lite falls back to random walk in 'the dark'



Acceptance rate in high dimensions

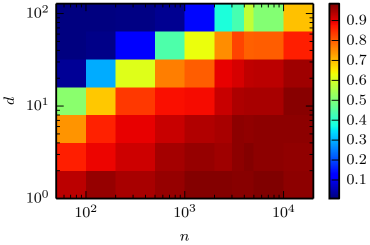
Challenging Gaussian (**top**):

- Eigenvalues: $\lambda_i \sim \texttt{Exp}(1)$.
- Covariance: $\mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, randomly rotate.
- 'Non-singular' length-scales
- KMC scales up to $d \approx 30$.



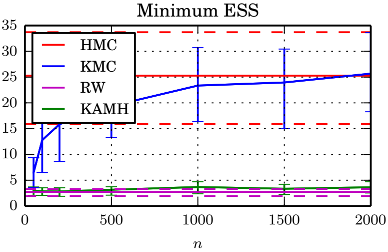
Isotropic Gaussian (**bottom**):

- More smooth
- KMC scales up to $d \approx 100$.

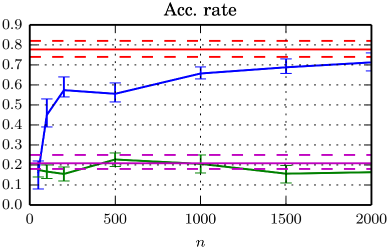


KMC asymptotically behaves as HMC

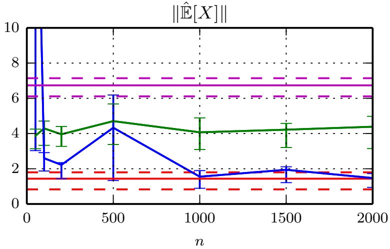
8-dimensional strongly nonlinear snythetic banana



Minimum ESS

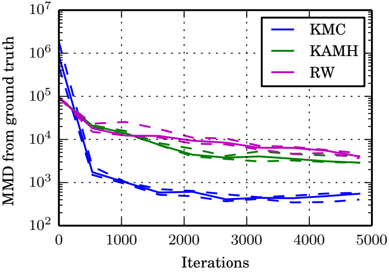


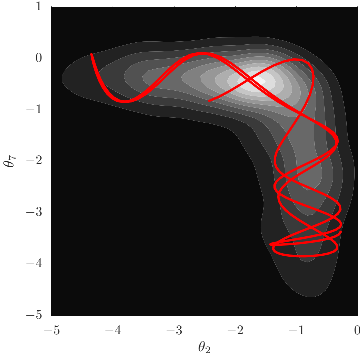
Acc. rate



$\|\hat{\mathbb{E}}[X]\|$

KMC improves mixing





Kernel sequential Monte Carlo

[Schuster & Strathmann et al., 2016]
Nonlinear versions of

- Adaptive Sequential Monte Carlo [Fearnhead et al., 2010]
- Feature space covariance

Gradient free versions of

- Gradient Importance Sampling [Schuster et al., 2015]
- Hamiltonian Importance Sampling [Naesseth et al., 2016]

Context:

- Intractable likelihoods, nested importance sampling
- IS^2/SMC^2 [Tran et al., 2013; Chopin et al., 2013]

Discussion

Kernel models as density emulators for Monte Carlo

- Covariance [Sejdinovic et al., 2012]
- Gradients [Strathmann et al., 2015]
- Leads to mixing improvements in practice
- Useful for intractable targets

The crucial trade-offs:

- Parameter selection
- Adaptation
- Computational costs
- Growing dimensions

Thank you

Questions?