Motivation: Hamiltonian Monte Carlo and Intractable Targets

- Goal: Efficient sampling from density \( f \) on \( \mathbb{R}^d \).
- HMC proposes distant moves with high acceptance probability.
- Given potential energy \( U(q) = \log f(q) \), sample auxiliary momentum \( p \) according to \( \exp(-K(p)) \) and simulate for \( t \in \mathbb{R} \) along Hamiltonian flow \( q^t \cdot (p, q) \rightarrow (p^t, q^t) \) of the joint log-density \( H(p, q) = K(p) + U(q) \) using the operator \( \frac{d}{dt} q^t |_{t=0} = -p \) of the local tangent space with the gradient of \( U \).
- Numerical simulation (i.e., leapfrog) depends on gradient information.
- Often unavailable, e.g., in Bayesian GP classification. More generally in Pseudo-Marginal MCMC [1] or Approximate Bayesian Computation [6].
- Right: Marginal hyper-parameters of a GP classifier.

We want a HMC sampler that automatically learns gradients.

So far. (Kernel) Adaptive Metropolis-Hastings

- Idea: use history of Markov chain to learn target structure.
- Adaptive Metropolis-Hastings [2]:
  - Learns global linear covariance.
  - Pro: Automatically learns proposal scaling, fast.
  - Con: Local steps, does not work well on non-linear targets.
- Kernel Adaptive Metropolis-Hastings [3]:
  - Learns covariance in RKHS.
  - Pro: Locally aligns to (non-linear) target covariance, gradient free.
  - Con: Local steps, random walk.

Can we combine ‘global’ and ‘non-linear’ — without gradients?

Hamiltonian Monte Carlo with kernel induced potential energy

- Learn gradient ‘surrogate’ model \( \nabla U \approx \nabla U = -\nabla \log f \) from Markov chain history \( \{x_i\}_{i=1}^\infty \).
- Replace \( \frac{d}{dt} q^t |_{t=0} = -p \) with \( \frac{d}{dt} \hat{q}^t |_{t=0} = -\hat{p} \) if \( \hat{q} \) can be simulated using the operator \( \frac{d}{dt} \hat{q}^t |_{t=0} = -\hat{p} \).
- Accept using true Hamiltonian (depends on \( U \) but not \( \nabla U \)) with probability \( \min \{1, \exp(-H(\hat{q}^t, \hat{p}^t) - H(q^t, p^t)) \} \).
- Corrects for both leapfrog error and surrogate induced Hamiltonian flow \( \nabla U \) asymptotically correct.
- Note: \( \exp(U(q)) \) can be replaced with unbiased estimator, e.g., Pseudo-Marginal MCMC.

Key quantity: average gradient error \( \frac{1}{n} \sum_{i=1}^n \| \nabla \log f(x_i) - \nabla \log f(x_i) \|_2 \).

Illustration of kernel induced Hamiltonian flow

- Standard HMC dynamics using \( \nabla U \) (plot shows gradient norm \( \| \nabla U \| \)).
- Dynamics on kernel surrogate \( \nabla \hat{U} \), fitted from samples.
- We need an expressive yet tractable model.

Infinite dimensional exponential families [6]

(Unnormalised) exponential family model in a RKHS:
\[
\text{const} \times \pi(x) = \exp \left\{ \langle f(x), k(x) \rangle - \mathcal{A}(f) \right\}
\]

- Sufficient statistics: feature map \( k(x) \in \mathcal{H} \), satisfies \( \mathcal{A}(f) = \langle f, k(x) \rangle \) for any \( f \in \mathcal{H} \).
- Natural parameters: \( \theta \) in \( \mathcal{H} \).
- Normalising constant \( \mathcal{A}(f) \) is intractable.

The model is:

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

How to learn \( f \) from samples without access to \( \mathcal{A}(f) \)?

Score matching [3]

- Allows estimation of unnormalised density models from samples.
- Minimises Fisher divergence (precisely the average gradient error):
\[
\hat{\mathcal{A}}(f) = \frac{1}{2} \int \left( \nabla \log f(x) - \nabla \log \hat{f}(x) \right) dx
\]

Possible without accessing \( \nabla \log \hat{f}(x) \) and accessing \( \nabla (f(x)) \) only through samples:
\[
\hat{\mathcal{A}}(f) = \frac{1}{2} \int \left( \nabla \log f(x) - \nabla \log \hat{f}(x) \right) dx
\]

Expensive: Closed form full solution requires solving \((d+1)\)-dimensional linear system.

Approximation I: KMC Lite

Assume that the model takes the form (Gaussian kernel \( k \) with bandwidth \( \sigma \))
\[
f_{\theta}(x) = \frac{1}{\sqrt{2\pi} \sigma} \exp \left( -\frac{(x-a)^2}{2\sigma^2} \right)
\]

- \( z \subseteq \mathbb{R} \) is a random sub-sample, \( \sigma \in \mathbb{R}^d \) are real valued parameters.
- Solution \( f_{\theta,z}(x) \) lies in smaller RKHS sub-space than original model, yet grows with \( n \).
- Can learn \( \theta \) from linear system
\[
\hat{\theta} = \text{C}^{-1} \text{d} + \text{bias}
\]

where \( C \in \mathbb{R}^{m \times l} \), \( b \in \mathbb{R}^l \) depend on kernel’s choice, and \( l > 0 \).

- Costs \( O(n^2 + n d) \). Can further reduce with low-rank approximations and conjugate gradient.

Approximation II: KMC Finite

Assume that the model takes the form
\[
f_{\theta}(x) = \exp \left( -\frac{(x-a)^2}{2\sigma^2} \right)
\]

- \( \sigma \in \mathbb{R}^d \) is approximate feature map such that \( \phi(\sigma) \approx \exp \left( -\frac{(x-a)^2}{2\sigma^2} \right) \), c.f. Random Fourier Features.
- \( \theta \in \mathbb{R}^d \) can be computed from
\[
\hat{\theta} = \text{C}^{-1} \text{d} + \text{bias}
\]

where \( C = \sum_{i=1}^l c_i \phi(\sigma_i) \in \mathbb{R}^{m \times n} \)

- \( c_i \in \mathbb{R}^m \) and \( \phi(\sigma_i) \in \mathbb{R}^m \), and \( l > 0 \).
- \( C \), \( b \) are running averages. On-line updates cost \( O(d^2) \).

Lite vs. Finite: geometric ergodicity & the tails

- KMC finite is geometrically ergodic on log-concave targets (fast convergence).
- KMC finite updates fast and uses all Markov chain history. Can’t: need to initialise correctly.
- Gradient norm of a Gaussian

KMC Lite

KMC Finite

Stability in growing dimensions

- KMC behaves like HMC as number of oracle samples increases.
- More smoothness allows KMC to scale up to \( d \approx 100 \).

Gaussian Process Classification on UCI data

- Standard GPC model
\[
p(\mathbf{y} | \mathbf{f}) = \prod_{i=1}^n p(\mathbf{y}_i | \mathbf{f}(\mathbf{x}_i))
\]

where \( p(\mathbf{f} | \mathbf{y}) \) is a GP and with a sigmoidal likelihood \( p(\mathbf{y} | \mathbf{f}) \).
- Goal: sample from \( p(\mathbf{f} | \mathbf{y}) \) with importance sampling.
- No access to likelihood or gradient.

Approximate Bayesian Computation on a Snow-Normal model

- Likelihood free MCMC for ABC via simulation from likelihood.
- Can fit (Gaussian) synthetic likelihoods.

This often induces bias, simple example:
\[
p(\mathbf{y} | \mathbf{f}) = 2\mathcal{N}(\mathbf{y} | \mathbf{f}(\mathbf{x}), \mathbf{I})
\]

with Gaussian CDF \( \Phi \) and skewness \( \alpha = 10^{-1} \).

- Compared to Hamiltonian ABC (gradients by stochastic finite differences):
  - No synthetic likelihood.
  - No stochastic gradients.

- No additional bias and reduced number of likelihood simulations.

References