Kernel exponential families

Arthur Gretton

Gatsby Computational Neuroscience Unit,
University College London

Advanced topics in ML, 2017

Outline

Motivating application:

- Fast estimation of complex multivariate densities

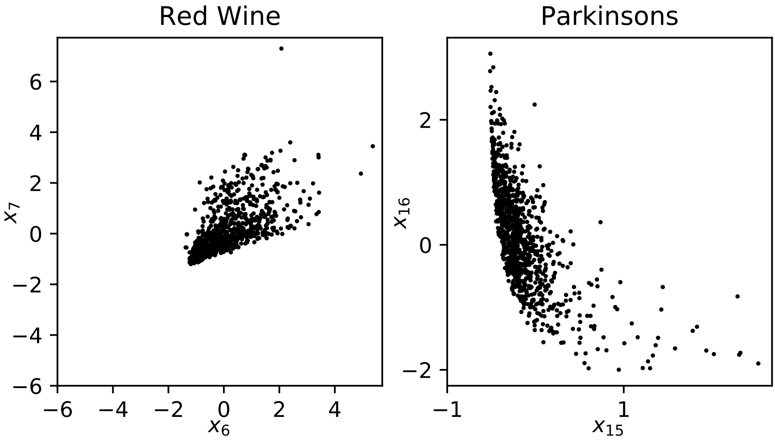
The infinite exponential family:

- Multivariate Gaussian $\rightarrow$ Gaussian process
- Finite mixture model $\rightarrow$ Dirichlet process mixture model
- Finite exponential family $\rightarrow$???

In this talk:

- Guaranteed speed improvements by Nystrom
- Conditional models
- Adaptive Hamiltonian Markov chain Monte Carlo

Goal 1 learn high dimensional, complex densities



We want:

- Efficient computation and representation
- Statistical guarantees

The (infinite) exponential family

[Sriperumbudur, Fukumizu, G., Hyvarinen, Kumar (2017)]

The exponential family

The exponential family in in $\mathbb{R}^d$

$$p(x) = \exp\left(\left\langle \underbrace{\eta}_{\substack{\text{natural}\\ \text{parameter}}}, \underbrace{T(x)}_{\substack{\text{sufficient}\\ \text{startistic}}} \right\rangle - \underbrace{A(\eta)}_{\substack{\text{log}\\ \text{normaliser}}} \right) \underbrace{q_0(x)}_{\substack{\text{base}\\ \text{measure}}}$$

Examples:

- Gaussian density: $T(x) = \begin{bmatrix} x & x^2 \end{bmatrix}$
- Gamma density: $T(x) = \begin{bmatrix} \ln x & x \end{bmatrix}$

Can we extend this to infinite dimensions?

The kernel exponential family

Kernel exponential families [Canu and Smola (2006), Fukumizu (2009)] and their GP counterparts [Adams, Murray, MacKay (2009), Rasmussen(2003)]

$$\mathcal{P} = \left\{ p_f(x) = e^{\langle f, \varphi(x) \rangle_{\mathcal{H}} - A(f)} q_0(x), \;\; x \in \Omega, f \in \mathcal{F} \right\}$$

where

$$\mathcal{F} = \left\{ f \in \mathcal{H} \; : \; A(f) = \log \int e^{f(x)} q_0(x) \, dx < \infty \right\}$$

The kernel exponential family

Kernel exponential families [Canu and Smola (2006), Fukumizu (2009)] and their GP counterparts [Adams, Murray, MacKay (2009), Rasmussen(2003)]

$$\mathcal{P} = \left\{ p_f(x) = e^{\langle f, \varphi(x) \rangle_{\mathcal{H}} - A(f)} q_0(x), \; x \in \Omega, f \in \mathcal{F} \right\}$$

where

$$\mathcal{F} = \left\{ f \in \mathcal{H} \; : \; A(f) = \log \int e^{f(x)} q_0(x)\, dx < \infty \right\}$$

Finite dimensional RKHS: one-to-one correspondence between finite dimensional exponential family and RKHS.

- Example: Gaussian kernel, $T(x) = \begin{bmatrix} x & x^2 \end{bmatrix} = \varphi(x)$ and $k(x, y) = xy + x^2 y^2$

Fitting an infinite dimensional exponential family

Given random samples, $X_1, \ldots, X_n$ drawn i.i.d. from an unknown density, $p_0 := p_{f_0} \in \mathcal{P}$, estimate p_0

How not to do it: maximum likelihood

Maximum likelihood:

$$
\begin{aligned}
f_{ML} &= \arg\max_{f \in \mathcal{F}} \sum_{i=1}^{n} \log p_f(X_i) \\
&= \arg\max_{f \in \mathcal{F}} \sum_{i=1}^{n} f(X_i) - n \log \int e^{f(x)} q_0(x)\, dx.
\end{aligned}
$$

Solving the above yields that f_{ML} satisfies

$$
\frac{1}{n} \sum_{i=1}^{n} \varphi(x_i) = \int \varphi(x) p_{f_{ML}}(x)\, dx
$$

where $p_{f_{ML}} = \frac{d\mathbb{P}_{\mathrm{ML}}}{dx}$.

Can this be solved?

How not to do it: maximum likelihood

- Finite dimensional case: Normal distribution $\mathcal{N}(\mu, \sigma)$

$$k(\cdot, x) = \begin{bmatrix} x & x^2 \end{bmatrix}^\top$$

- Max. likelihood equations give

$$\frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix} x_i & x_i^2 \end{bmatrix}^\top = \int \begin{bmatrix} x & x^2 \end{bmatrix}^\top p_{f_{ML}}(x)\, dx = \begin{bmatrix} \mu_{ML} & (\sigma_{ML}^2 + \mu_{ML}^2) \end{bmatrix}^\top$$

- System of likelihood equations: solvable.

How not to do it: maximum likelihood

- Finite dimensional case: Normal distribution $\mathcal{N}(\mu, \sigma)$

$$k(\cdot, x) = \begin{bmatrix} x & x^2 \end{bmatrix}^\top$$

- Max. likelihood equations give

$$\frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix} x_i & x_i^2 \end{bmatrix}^\top = \int \begin{bmatrix} x & x^2 \end{bmatrix}^\top p_{f_{ML}}(x)\, dx = \begin{bmatrix} \mu_{ML} & (\sigma_{ML}^2 + \mu_{ML}^2) \end{bmatrix}^\top$$

- System of likelihood equations: solvable.
- Infinite dimensional case, characteristic kernel: ill-posed! [Fukumizu (2009)]

Score matching

Journal of Machine Learning Research 6 (2005) 695–708 Submitted 11/04; Revised 3/05; Published 4/05

Estimation of Non-Normalized Statistical Models by Score Matching

Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Helsinki Institute for Information Technology (BRU)
Department of Computer Science
FIN-00014 University of Helsinki, Finland

Editor: Peter Dayan

Loss is Fisher Score:

$$D_F(p_0, p_f) := \frac{1}{2} \int p_0(x) \left\| \nabla_x \log p_0(x) - \nabla_x \log p_f(x) \right\|^2 \, dx$$

Score matching: 1-D proof

$$D_F(p_0, p_f)$$
$$= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx} - \frac{d\log p_f(x)}{dx}\right)^2 dx$$

Score matching: 1-D proof

$$
\begin{aligned}
& D_F(p_0, p_f) \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx} - \frac{d\log p_f(x)}{dx}\right)^2 dx \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx}\right)^2 dx + \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)^2 dx \\
&\qquad - \int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx
\end{aligned}
$$

Score matching: 1-D proof

$$
\begin{aligned}
&D_F(p_0, p_f) \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx} - \frac{d\log p_f(x)}{dx}\right)^2 dx \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx}\right)^2 dx + \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)^2 dx \\
&\qquad - \int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx
\end{aligned}
$$

Final term:

$$
\begin{aligned}
&\int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx \\
&= \int_a^b \cancel{p_0(x)}\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{1}{\cancel{p_0(x)}}\frac{dp_0(x)}{dx}\right) dx \\
&= \left[\left(\frac{d\log p_f(x)}{dx}\right)p_0(x)\right]_a^b - \int_a^b p_0(x)\frac{d^2\log p_f(x)}{dx^2}.
\end{aligned}
$$

Score matching: 1-D proof

$$
\begin{aligned}
&D_F(p_0, p_f) \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx} - \frac{d\log p_f(x)}{dx}\right)^2 dx \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx}\right)^2 dx + \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)^2 dx \\
&\qquad - \int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx
\end{aligned}
$$

Final term:

$$
\begin{aligned}
&\int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx \\
&= \int_a^b \cancel{p_0(x)}\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{1}{\cancel{p_0(x)}}\frac{dp_0(x)}{dx}\right) dx \\
&= \left[\left(\frac{d\log p_f(x)}{dx}\right) p_0(x)\right]_a^b - \int_a^b p_0(x)\frac{d^2\log p_f(x)}{dx^2}.
\end{aligned}
$$

Score matching: 1-D proof

$$
\begin{aligned}
& D_F(p_0, p_f) \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx} - \frac{d\log p_f(x)}{dx}\right)^2 dx \\
&= \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_0(x)}{dx}\right)^2 dx + \frac{1}{2}\int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)^2 dx \\
&\quad - \int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx
\end{aligned}
$$

Final term:

$$
\begin{aligned}
& \int_a^b p_0(x)\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{d\log p_0(x)}{dx}\right) dx \\
&= \int_a^b \cancel{p_0(x)}\left(\frac{d\log p_f(x)}{dx}\right)\left(\frac{1}{\cancel{p_0(x)}}\frac{dp_0(x)}{dx}\right) dx \\
&= \left[\left(\frac{d\log p_f(x)}{dx}\right) p_0(x)\right]_a^b - \int_a^b p_0(x)\frac{d^2\log p_f(x)}{dx^2}.
\end{aligned}
$$

Empirical score matching

p_n represents n i.i.d. samples from P_0

$$D_F(p_n, p_f) := \frac{1}{n} \sum_{a=1}^{n} \sum_{i=1}^{d} \left(\frac{1}{2} \left(\frac{\partial \log p_f(X_a)}{\partial x_i} \right)^2 + \frac{\partial^2 \log p_f(X_a)}{\partial x_i^2} \right) + C$$

Since $D_F(p_n, p_f)$ is independent of $A(f)$,

$$f_n^* = \arg\min_{f \in \mathcal{F}} D_F(p_n, p_f)$$

should be easily computable, unlike the MLE.

Empirical score matching

p_n represents n i.i.d. samples from P_0

$$D_F(p_n, p_f) := \frac{1}{n} \sum_{a=1}^{n} \sum_{i=1}^{d} \left(\frac{1}{2} \left(\frac{\partial \log p_f(X_a)}{\partial x_i} \right)^2 + \frac{\partial^2 \log p_f(X_a)}{\partial x_i^2} \right) + C$$

Since $D_F(p_n, p_f)$ is independent of $A(f)$,

$$f_n^* = \arg\min_{f \in \mathcal{F}} D_F(p_n, p_f)$$

should be easily computable, unlike the MLE.

Add extra term $\lambda \|f\|_{\mathcal{H}}^2$ to regularize.

A kernel solution

Infinite exponential family:

$$p_f(x) = e^{\langle f, \varphi(x) \rangle_{\mathcal{H}} - A(f)} q_0(x)$$

Thus

$$\frac{\partial}{\partial x} \log p_f(x) = \frac{\partial}{\partial x} \langle f, \varphi(x) \rangle_{\mathcal{H}} + \frac{\partial}{\partial x} \log q_0(x).$$

A kernel solution

Infinite exponential family:

$$p_f(x) = e^{\langle f, \varphi(x) \rangle_{\mathcal{H}} - A(f)} q_0(x)$$

Thus

$$\frac{\partial}{\partial x} \log p_f(x) = \frac{\partial}{\partial x} \langle f, \varphi(x) \rangle_{\mathcal{H}} + \frac{\partial}{\partial x} \log q_0(x).$$

Kernel trick for derivatives:

$$\frac{\partial}{\partial x_i} f(X) = \left\langle f, \frac{\partial}{\partial x_i} \varphi(X) \right\rangle_{\mathcal{H}}$$

Dot product between feature derivatives:

$$\left\langle \frac{\partial}{\partial x_i} \varphi(X), \frac{\partial}{\partial x_j} \varphi(X') \right\rangle_{\mathcal{H}} = \frac{\partial^2}{\partial x_i \partial x_{d+j}} k(X, X')$$

A kernel solution

Infinite exponential family:

$$p_f(x) = e^{\langle f, \varphi(x)\rangle_{\mathcal{H}} - A(f)} q_0(x)$$

Thus

$$\frac{\partial}{\partial x} \log p_f(x) = \frac{\partial}{\partial x} \langle f, \varphi(x)\rangle_{\mathcal{H}} + \frac{\partial}{\partial x} \log q_0(x).$$

Kernel trick for derivatives:

$$\frac{\partial}{\partial x_i} f(X) = \left\langle f, \frac{\partial}{\partial x_i} \varphi(X) \right\rangle_{\mathcal{H}}$$

Dot product between feature derivatives:

$$\left\langle \frac{\partial}{\partial x_i} \varphi(X), \frac{\partial}{\partial x_j} \varphi(X') \right\rangle_{\mathcal{H}} = \frac{\partial^2}{\partial x_i \partial x_{d+j}} k(X, X')$$

By representer theorem:

$$f_n^* = \alpha \hat{\xi} + \sum_{\ell=1}^{n} \sum_{j=1}^{d} \beta_{\ell j} \frac{\partial \varphi(X_\ell)}{\partial x_j}$$

An RKHS solution

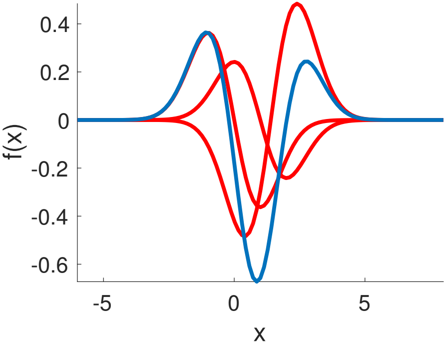
The RKHS solution

$$f_n^* = \alpha\hat{\xi} + \sum_{\ell=1}^{n}\sum_{j=1}^{d}\beta_{\ell j}\frac{\partial\varphi(X_\ell)}{\partial x_j}$$

Need to solve a linear system

$$\beta_n^* = -\frac{1}{\lambda}\left(\underbrace{G_{XX}}_{nd\times nd} + n\lambda I\right)^{-1} h_X$$

Very costly in high dimensions!



The Nystrom approximation

[Sutherland, Strathmann, Arbel, G. (2018)

Nystrom approach for efficient solution

- Find best estimator $f^*_{n,m}$ in $\mathcal{H}_Y := \text{span}\{\partial_i k(y_a, \cdot)\}_{a\in[m], i\in[d]}$, where $y_a \in \{x_i\}_{i=1}^n$ chosen at random.
- Nystrom solution:

$$\beta^*_{n,m} = -\left(\frac{1}{n} B_{XY}^\top \underbrace{B_{XY}}_{md\times nd} + \lambda \underbrace{G_{YY}}_{md\times md}\right)^\dagger h_Y$$

Solve in time $\mathcal{O}(nm^2 d^3)$, evaluate in time $\mathcal{O}(md)$.

- Sill cubic in d, but similar results if we take a random dimension per datapoint.

Consistency: original solution

Define C as the covariance between feature derivatives. Then from

[Sriperumbudur et al. JMLR (2017)]

- **Rates of convergence:** Suppose
  - $f_0 \in \mathcal{R}(C^{\beta})$ for some $\beta > 0$.
  - $\lambda = n^{-\max\left\{\frac{1}{3}, \frac{1}{2(\beta+1)}\right\}}$ as $n \to \infty$.

Then

$$D_F(p_0, p_{f_n}) = O_{p_0}\left(n^{-\min\left\{\frac{2}{3}, \frac{\beta}{2(\beta+1)}\right\}}\right)$$

Consistency: original solution

Define C as the covariance between feature derivatives. Then from

[Sriperumbudur et al. JMLR (2017)]

- Rates of convergence: Suppose
  - $f_0 \in \mathcal{R}(C^\beta)$ for some $\beta > 0$.
  - $\lambda = n^{-\max\left\{\frac{1}{3}, \frac{1}{2(\beta+1)}\right\}}$ as $n \to \infty$.

Then

$$D_F(p_0, p_{f_n}) = O_{p_0}\left(n^{-\min\left\{\frac{2}{3}, \frac{\beta}{2(\beta+1)}\right\}}\right)$$

- Convergence in other metrics: KL, Hellinger, $L_r, 1 < r < \infty$.

Consistency: Nystrom solution

Define C as the covariance between feature derivatives.

- Suppose
  - $f_0 \in \mathcal{R}(C^{\beta})$ for some $\beta > 0$.
  - Number of subsampled points $m = \Omega(n^{\theta} \log n)$ for $\theta = (\min(2\beta, 1) + 2)^{-1} \in \left[\frac{1}{3}, \frac{1}{2}\right]$
  - $\lambda = n^{-\max\left\{\frac{1}{3}, \frac{1}{2(\beta+1)}\right\}}$ as $n \to \infty$.
- Then

$$D_F(p_0, p_{f_{n,m}}) = O_{p_0}\left(n^{-\min\left\{\frac{2}{3}, \frac{\beta}{2(\beta+1)}\right\}}\right)$$

Consistency: Nystrom solution

Define C as the covariance between feature derivatives.

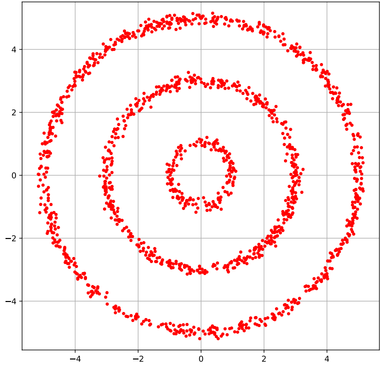
- Suppose
  - $f_0 \in \mathcal{R}(C^\beta)$ for some $\beta > 0$.
  - Number of subsampled points $m = \Omega(n^\theta \log n)$ for $\theta = (\min(2\beta, 1) + 2)^{-1} \in \left[\frac{1}{3}, \frac{1}{2}\right]$
  - $\lambda = n^{-\max\left\{\frac{1}{3}, \frac{1}{2(\beta+1)}\right\}}$ as $n \to \infty$.
- Then

$$D_F(p_0, p_{f_{n,m}}) = O_{p_0}\left(n^{-\min\left\{\frac{2}{3}, \frac{\beta}{2(\beta+1)}\right\}}\right)$$

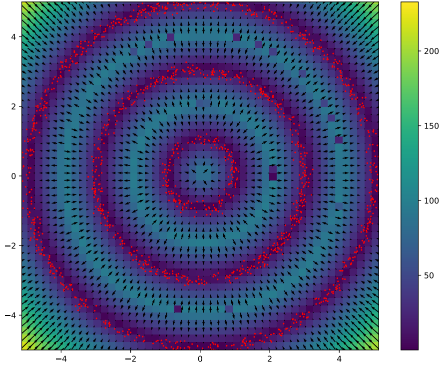
- Convergence in other metrics: KL, Hellinger, $L_r, 1 < r < \infty$. Same rate but saturates sooner.
  - Full KL original saturates at $O_{p_0}\left(n^{-\frac{1}{2}}\right)$
  - Nystrom saturates at $O_{p_0}\left(n^{-\frac{1}{3}}\right)$

Experimental results: ring

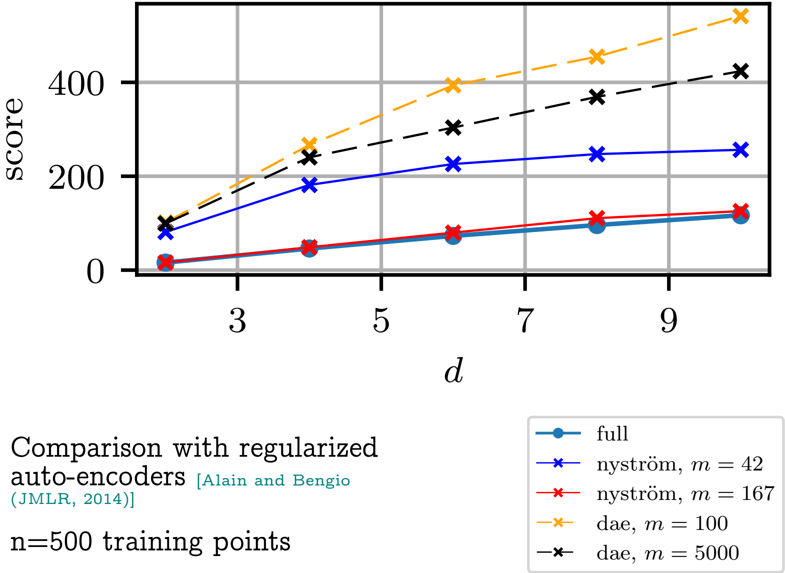
Sample:



Score:



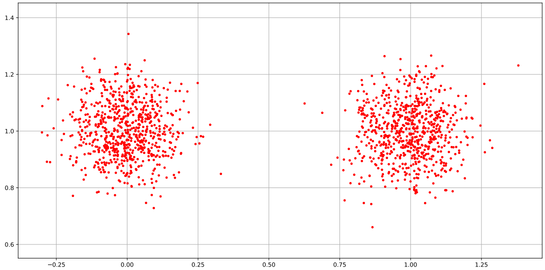
Experimental results: comparison with autoencoder



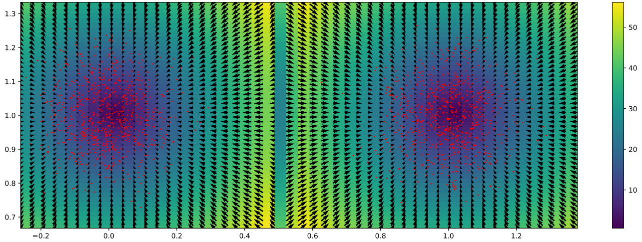
- Comparison with regularized auto-encoders [Alain and Bengio (JMLR, 2014)]
- n=500 training points

Experimental results: grid of Gaussians

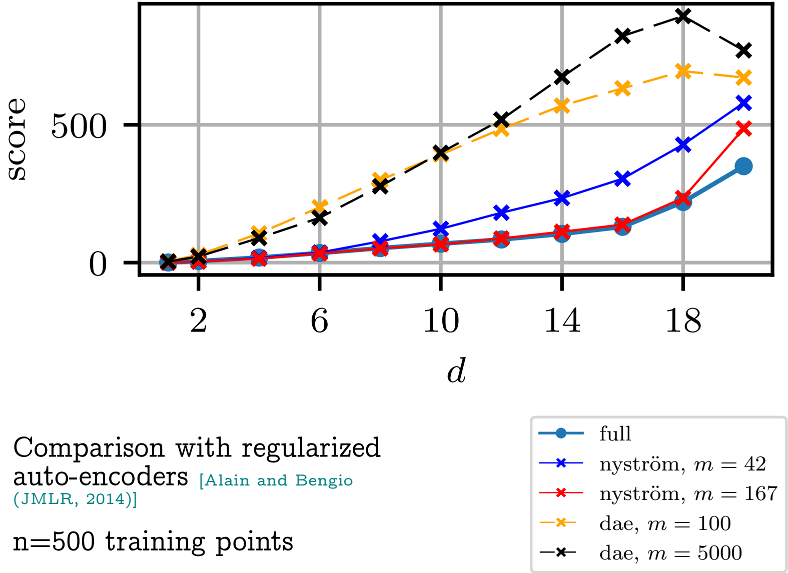
Sample:



Score:



Experimental results: comparison with autoencoder



- Comparison with regularized auto-encoders [Alain and Bengio (JMLR, 2014)]
- n=500 training points

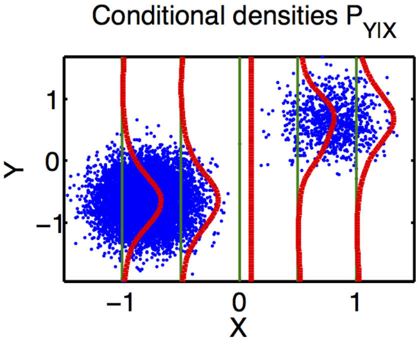
The kernel conditional exponential family [Arbel & G. (2018)]

The kernel conditional exponential family

- Can we take advantage of the graphical structure of $(X_1, ..., X_d)$?
- Start from a general factorization of P

$$P(X_1, ..., X_d) = \prod_i P(X_i | \underbrace{X_{\pi(i)}}_{\substack{\text{parents} \\ \text{of } X_i}})$$

- Estimate each factor independently



Kernel conditional exponential family

General definition, kernel conditional exponential family

[Smola and Canu, 2006]

$$p_f(y|x) = e^{\langle f, \psi(x,y)\rangle_{\mathcal{H}} - A(f,x)} q_0(y) \qquad A(f,x) = \log \int q_o(y) e^{\langle f, \psi(x,y)\rangle_{\mathcal{H}}} dy$$

(joint feature map $\psi(x, y)$)

Kernel conditional exponential family

Our kernel conditional exponential family:

$$p_f(x) = e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}} - A(f,x)} q_0(y) \qquad A(f,x) = \log \int q_o(y) e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}}}$$

linear in the sufficient statistic $\phi(y) \in \mathcal{G}$.

Kernel conditional exponential family

Our kernel conditional exponential family:

$$p_f(x) = e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}} - A(f,x)} q_0(y) \qquad A(f,x) = \log \int q_o(y) e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}}}$$

linear in the sufficient statistic $\phi(y) \in \mathcal{G}$.

What does this RKHS look like?

[Micchelli and Pontil, (2005)]

$$\begin{aligned}
&\langle f_x, \phi(y) \rangle_{\mathcal{G}} \\
&= \langle \Gamma_x^* f, \phi(y) \rangle_{\mathcal{G}} \\
&= \langle f, \Gamma_x \phi(y) \rangle_{\mathcal{H}}
\end{aligned}$$

Kernel conditional exponential family

Our kernel conditional exponential family:

$$p_f(x) = e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}} - A(f,x)} q_0(y) \qquad A(f,x) = \log \int q_o(y) e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}}}$$

linear in the sufficient statistic $\phi(y) \in \mathcal{G}$.

What does this RKHS look like?

[Micchelli and Pontil, (2005)]

$$\begin{aligned} &\langle f_x, \phi(y) \rangle_{\mathcal{G}} \\ &= \langle \Gamma_x^* f, \phi(y) \rangle_{\mathcal{G}} \\ &= \langle f, \Gamma_x \phi(y) \rangle_{\mathcal{H}} \end{aligned}$$

- $\Gamma_x^* : \mathcal{H} \to \mathcal{G}$ is a linear operator

Kernel conditional exponential family

Our kernel conditional exponential family:

$$p_f(x) = e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}} - A(f,x)} q_0(y) \qquad A(f,x) = \log \int q_o(y) e^{\langle f_x, \phi(y) \rangle_{\mathcal{G}}}$$

linear in the sufficient statistic $\phi(y) \in \mathcal{G}$.

What does this RKHS look like?

[Micchelli and Pontil, (2005)]

$$\begin{aligned} &\langle f_x, \phi(y) \rangle_{\mathcal{G}} \\ &= \langle \Gamma_x^* f, \phi(y) \rangle_{\mathcal{G}} \\ &= \langle f, \Gamma_x \phi(y) \rangle_{\mathcal{H}} \end{aligned}$$

- $\Gamma_x : \mathcal{G} \to \mathcal{H}$ is a linear operator.
- The feature map $\psi(x, y) := \Gamma_x \phi(y)$

What is our loss function?

The obvious approach: minimise

$$D_F\left[p_0(x)p_0(y|x) \| p_f(x)p_f(y|x)\right]$$

Problem: the expression still contains $\int p_0(y|x)\,dy$.

What is our loss function?

The obvious approach: minimise

$$D_F\left[p_0(x)p_0(y|x)||p_f(x)p_f(y|x)\right]$$

Problem: the expression still contains $\int p_0(y|x)dy$.

Our loss function:

$$\widetilde{D}_F(p_0, p_f) := \int D_F(p_0(y|x)||p_f(y|x))\pi(x)dx$$

for some $\pi(x)$ that includes the support of $p(x)$.

Finite sample estimate of the conditional density

Use the simplest operator-valued RKHS $\Gamma_x = I_{\mathcal{G}} k(x, \cdot)$.

$$\begin{aligned} \Gamma_x &: \mathcal{G} \to \mathcal{H} \\ \Gamma_x \phi(y) &\mapsto \phi(y) k(x, \cdot) \end{aligned}$$

Finite sample estimate of the conditional density

Use the simplest operator-valued RKHS $\Gamma_x = I_{\mathcal{G}} k(x, \cdot)$.

$$\begin{aligned} \Gamma_x \quad &: \quad \mathcal{G} \to \mathcal{H} \\ \Gamma_x \phi(y) \quad &\mapsto \quad \phi(y) k(x, \cdot) \end{aligned}$$

Solution:

$$f_n^*(y|x) = \sum_{b=1}^{n} \sum_{i=1}^{d} \beta_{(b,i)} k(X_b, x) \partial_i \mathfrak{K}(Y_b, y) + \alpha \hat{\xi}$$

where

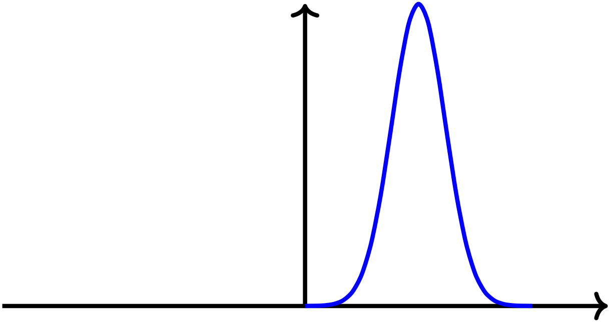
$$\beta_n^* = -\frac{1}{\lambda} (G + n\lambda I)^{-1} h$$

$$(G)_{(a,i),(b,j)} = k(X_a, X_b) \partial_i \partial_{j+d} \mathfrak{K}(Y_a, Y_b),$$

and $\langle \phi(y), \phi(y') \rangle_{\mathcal{G}} = \mathfrak{K}(y, y')$.

Expected conditional score: a failure case

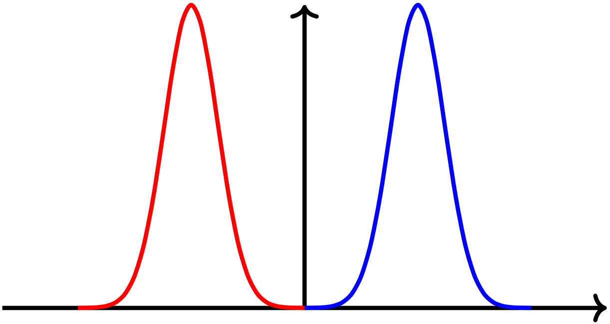
- $P(Y|X=1)$
- $P(Y|X=-1)$
- $P(Y) = \frac{1}{2}(P(Y|X=1) + P(Y|X=-1))$



$$\widetilde{D}_F(\underbrace{p(y|x)}_{\text{target}}, \underbrace{p(y)}_{\text{model}}) = 0$$

Expected conditional score: a failure case

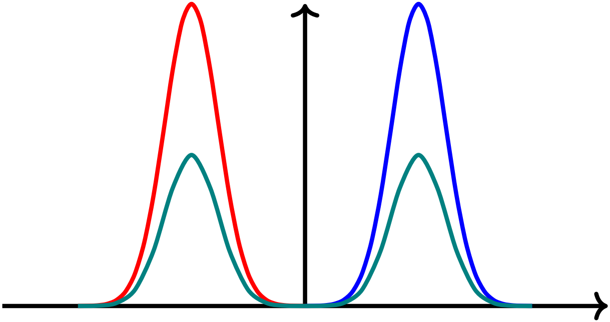
- $P(Y|X=1)$
- $P(Y|X=-1)$
- $P(Y) = \frac{1}{2}(P(Y|X=1) + P(Y|X=-1))$



$$\widetilde{D}_F(\underbrace{p(y|x)}_{\text{target}}, \underbrace{p(y)}_{\text{model}}) = 0$$

Expected conditional score: a failure case

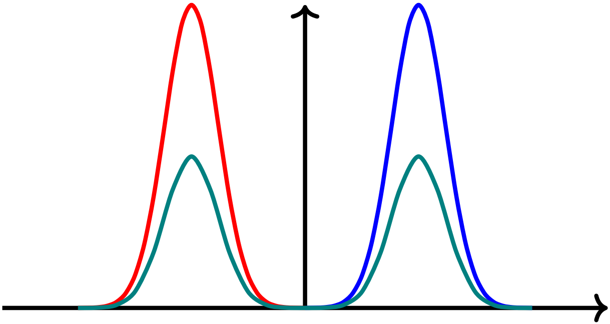
- $P(Y|X=1)$
- $P(Y|X=-1)$
- $P(Y) = \frac{1}{2}(P(Y|X=1) + P(Y|X=-1))$



$$\widetilde{D}_F(\underbrace{p(y|x)}_{\text{target}}, \underbrace{p(y)}_{\text{model}}) = 0$$

Expected conditional score: a failure case

- $P(Y|X=1)$
- $P(Y|X=-1)$
- $P(Y) = \frac{1}{2}(P(Y|X=1) + P(Y|X=-1))$



$$\widetilde{D}_F(\underbrace{p(y|x)}_{\text{target}}, \underbrace{p(y)}_{\text{model}}) = 0$$

Expected conditional score: a failure case

Why does it fail? Recall

$$\widetilde{D}_F(p_0(y|x), p_f(y|x)) := \int \pi(x) D_F(p_0(y|x), p_f(y|x)) dx$$

Note that

$$D_F(\underbrace{p(y|x=1)}_{\text{target}}, \underbrace{p(y)}_{\text{model}}) = \int p(y|1) \left\| \nabla_x \log p(y|1) - \nabla_x \log p(y) \right\|^2 \, dy$$

Model $p(y)$ puts mass where target conditional $p(y|1)$ has no support.

- Care needed when this failure mode approached!

Unconditional vs conditional model in practice

- Red Wine: Physiochemical measurements on wine samples.
- Parkinsons: Biomedical voice measurements from patients with early stage Parkinson's disease.

	Parkinsons	Red Wine
Dimension	15	11
Samples	5875	1599

Unconditional vs conditional model in practice

- Red Wine: Physiochemical measurements on wine samples.
- Parkinsons: Biomedical voice measurements from patients with early stage Parkinson's disease.

Comparison with

- LSCDE model: with consistency guarantees [Sugiyama et al., (2010)]
- RNADE model: mixture models with deep features of parents, no guarantees [Uria et al. (2016)]

Unconditional vs conditional model in practice

- Red Wine: Physiochemical measurements on wine samples.
- Parkinsons: Biomedical voice measurements from patients with early stage Parkinson's disease.

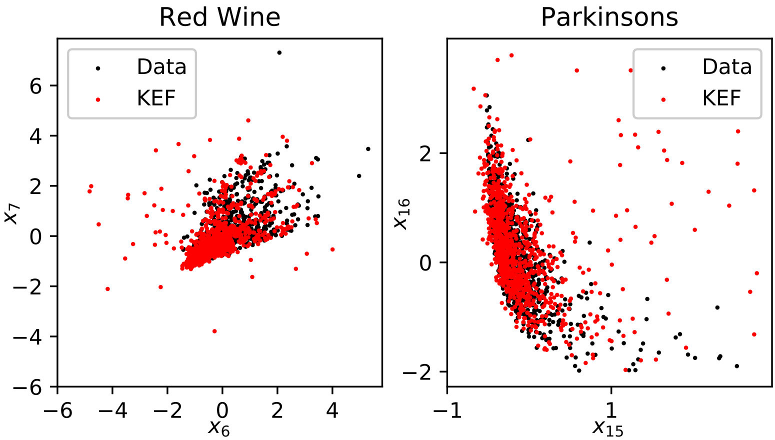
Comparison with

- LSCDE model: with consistency guarantees [Sugiyama et al., (2010)]
- RNADE model: mixture models with deep features of parents, no guarantees [Uria et al. (2016)]

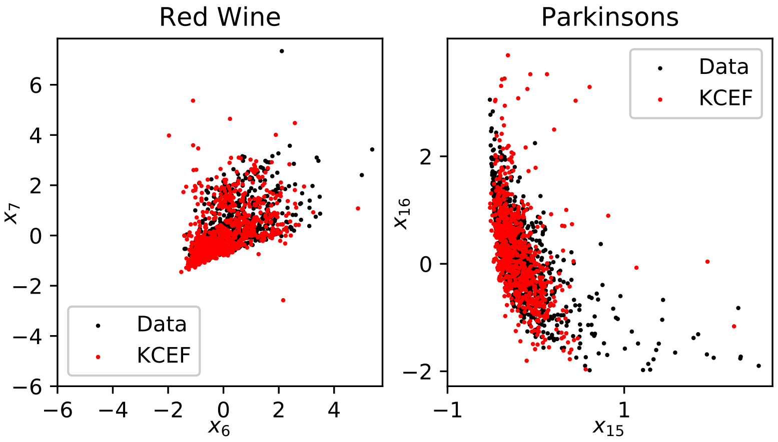
Negative log likelihoods (smaller is better, average over 5 test/train splits)

	Parkinsons	Red wine
KCEF	$\mathbf{2.86 \pm 0.77}$	11.8 ± 0.93
LSCDE	15.89 ± 1.48	14.43 ± 1.5
NADE	3.63 ± 0.0	$\mathbf{9.98 \pm 0.0}$

Results: unconditional model



Results: conditional model



Adaptive Hamiltonian Monte Carlo

Markov chain Monte Carlo

- We have a density of the form

$$p(x) = \frac{\pi(x)}{Z} \qquad Z = \int \pi(x)\,dx$$

Z often impractical to compute

- Goal: to compute expectations of functions,

$$\mathbb{E}_p[f(x)] = \int f(x)p(x)\,dx$$

- Given samples $\{x_i\}_{i=1}^n$ with distribution $p(x)$,

$$\widehat{\mathbb{E}}_p[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

How to generate these samples?

Markov chain Monte Carlo

- 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

Markov chain Monte Carlo

- 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

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

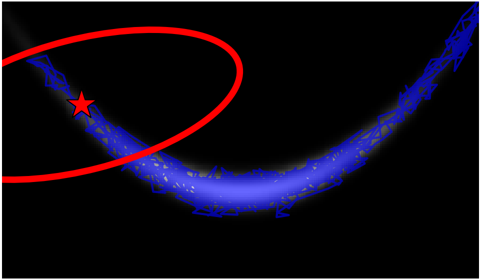
- What proposal $q(\cdot|x_t)$?
  - Too narrow or broad: $\rightarrow$ slow convergence
  - Does not conform to support of target $\rightarrow$ slow convergence

Basic adaptive Metropolis-Hastings

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.

Basic adaptive Metropolis-Hastings

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.



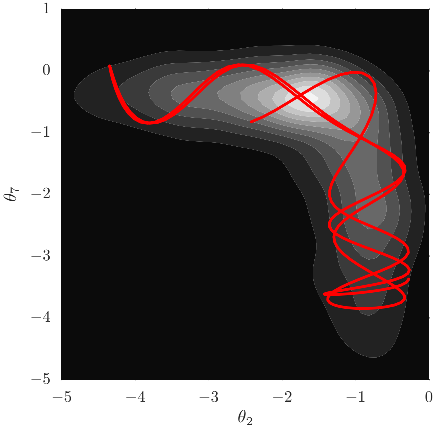
Locally miscalibrated for strongly non-linear targets: directions of large variance depend on the current location

Hamiltonian Monte Carlo

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

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

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



Bayesian Gaussian process classification

Our case: target $\pi(\cdot)$ and log gradient not computable - **Pseudo-Marginal MCMC**

When is target not computable?

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

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

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

- Automatic Relevance Determination (ARD) covariance:

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

- $p(\mathrm{y}|\mathrm{f}) = \prod_{i=1}^{n} p(y_i|f(x_i))$ where

$$p(y_i|f(x_i)) = (1 - \exp(-y_i f(x_i)))^{-1}, \qquad y_i \in \{-1, 1\}.$$

Bayesian Gaussian process classification

Our case: target $\pi(\cdot)$ and log gradient not computable - **Pseudo-Marginal MCMC**

When is target not computable?

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

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

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

- Automatic Relevance Determination (ARD) covariance:

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

- $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f(x_i))$ where

$$p(y_i|f(x_i)) = (1-\exp(-y_i f(x_i)))^{-1}, \qquad y_i \in \{-1,1\}.$$

Bayesian Gaussian process classification

Our case: target $\pi(\cdot)$ and log gradient not computable - **Pseudo-Marginal MCMC**

When is target not computable?

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

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

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

- Automatic Relevance Determination (ARD) covariance:

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

- $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f(x_i))$ where

$$p(y_i|f(x_i)) = (1 - \exp(-y_i f(x_i)))^{-1}, \qquad y_i \in \{-1, 1\}.$$

Bayesian Gaussian process classification

Our case: target $\pi(\cdot)$ and log gradient not computable - **Pseudo-Marginal MCMC**

When is target not computable?

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

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

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

- Automatic Relevance Determination (ARD) covariance:

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

- $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f(x_i))$ where

$$p(y_i|f(x_i)) = (1 - \exp(-y_i f(x_i)))^{-1}, \qquad y_i \in \{-1, 1\}.$$

Bayesian Gaussian process classification

Example: when is target not computable?

- Gaussian process classification, latent process $\mathbf{f}$

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

... but cannot integrate out $\mathbf{f}$

- 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: unbiased estimate of $p(\mathbf{y}|\theta)$ via importance sampling: [Filippone & Girolami, (2013)]

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

Bayesian Gaussian process classification

Example: when is target not computable?

- Gaussian process classification, latent process $\mathbf{f}$

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

  ... but cannot integrate out $\mathbf{f}$

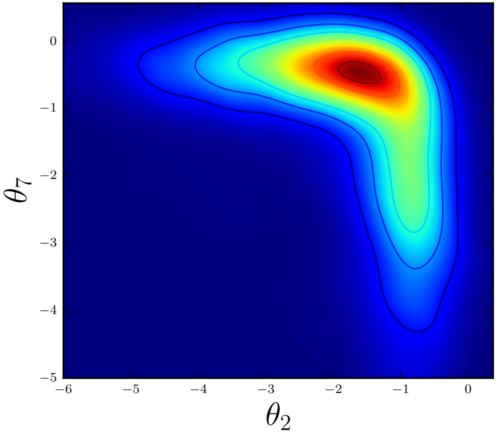
- Estimated MH ratio:

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

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

Basic adaptive Metropolis-Hastings

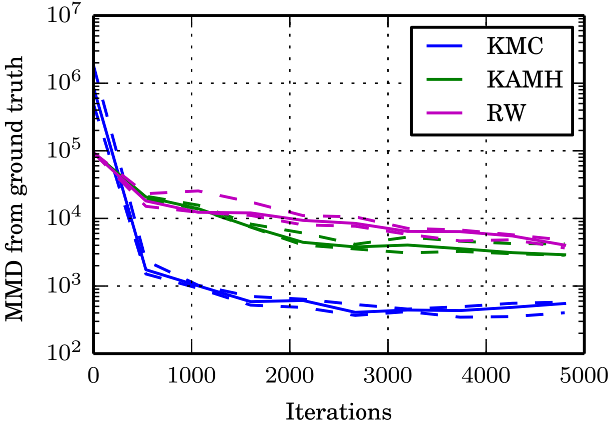
Sliced posterior over hyperparameters of a Gaussian Process classifier on UCI Glass dataset obtained using Pseudo-Marginal MCMC.



Can you learn an HMC sampler?

Basic adaptive Metropolis-Hastings

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



Significant improvements over random walk

Co-authors

From Gatsby:

- Michael Arbel
- Aapo Hyvarinen
- Heiko Strathmann
- Dougal Sutherland

External collaborators:

- Kenji Fukumizu
- Revant Kumar
- Bharath Sriperumbudur

Questions?