Advances in kernel exponential families

Arthur Gretton

Gatsby Computational Neuroscience Unit, University College London

NIPS, 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
Goal: learn high dimensional, complex densities

We want:
- Efficient computation and representation
- Statistical guarantees
The exponential family

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

$$p(x) = \exp \left( \langle \begin{array}{c} \eta \\ \text{natural parameter} \\ \end{array} , \begin{array}{c} T(x) \\ \text{sufficient statistic} \\ \end{array} \rangle - \begin{array}{c} A(\eta) \\ \text{log normaliser} \\ \end{array} \right) + q_0(x)$$

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?
Infinitely many features using kernels

Kernels: dot products of features

Feature map \( \varphi(x) \in \mathcal{H} \),

\[ \varphi(x) = [\ldots \varphi_i(x) \ldots] \in l_2 \]

For positive definite \( k \),

\[ k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}} \]
\[ = \langle k(x, \cdot), k(x', \cdot) \rangle_{\mathcal{H}} \]

Infinitely many features \( \varphi(x) \), dot product in closed form!
Infinitely many features using kernels

Kernels: dot products of features

Feature map $\varphi(x) \in \mathcal{H}$,

$$\varphi(x) = [\ldots \varphi_i(x) \ldots] \in \ell_2$$

For positive definite $k$,

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$$

$$= \langle k(x, \cdot), k(x', \cdot) \rangle_{\mathcal{H}}$$

Infinitely many features $\varphi(x)$, dot product in closed form!

Exponentiated quadratic kernel

$$k(x, x') = \exp \left(-\gamma \|x - x'\|^2\right)$$

Features: Gaussian Processes for Machine learning, Rasmussen and Williams, Ch. 4.
Functions of infinitely many features

Functions are linear combinations of features:

\[ f(x) = \langle f, \varphi(x) \rangle_{\mathcal{H}} = \sum_{\ell=1}^{\infty} f_\ell \varphi_\ell(x) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix}^T \begin{bmatrix} \varphi_1(x) \\ \varphi_2(x) \\ \varphi_3(x) \\ \vdots \end{bmatrix} \]
How to represent functions?

Function with exponentiated quadratic kernel:

\[ f(x) : = \sum_{i=1}^{m} \alpha_i k(x_i, x) \]
\[ = \sum_{i=1}^{m} \alpha_i \langle \varphi(x_i), \varphi(x) \rangle_{\mathcal{H}} \]
\[ = \left\langle \sum_{i=1}^{m} \alpha_i \varphi(x_i), \varphi(x) \right\rangle_{\mathcal{H}} \]
\[ = \sum_{\ell=1}^{\infty} f_{\ell} \varphi_{\ell}(x) \]
Function with exponentiated quadratic kernel:

\[ f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x) \]

\[ = \sum_{i=1}^{m} \alpha_i \langle \varphi(x_i), \varphi(x) \rangle_H \]

\[ = \left\langle \sum_{i=1}^{m} \alpha_i \varphi(x_i), \varphi(x) \right\rangle_H \]

\[ = \sum_{\ell=1}^{\infty} f_{\ell} \varphi_{\ell}(x) \]
How to represent functions?

Function with exponentiated quadratic kernel:

\[ f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x) \]

\[ = \sum_{i=1}^{m} \alpha_i \langle \varphi(x_i), \varphi(x) \rangle_{\mathcal{H}} \]

\[ = \left\langle \sum_{i=1}^{m} \alpha_i \varphi(x_i), \varphi(x) \right\rangle_{\mathcal{H}} \]

\[ = \sum_{\ell=1}^{\infty} f_{\ell} \varphi_{\ell}(x) \]

\[ f_{\ell} = \sum_{i=1}^{m} \alpha_i \varphi_{\ell}(x_i) \]
The kernel exponential family


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

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


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

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 \( k(x, y) = xy + x^2 y^2 \)
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:

\[
f_{ML} = \arg \max_{f \in \mathcal{F}} \sum_{i=1}^{n} \log p_f(X_i)
\]

\[
= \arg \max_{f \in \mathcal{F}} \sum_{i=1}^{n} f(X_i) - n \log \int e^{f(x)} q_0(x) \, dx.
\]

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 \(p_{f_{ML}} = \frac{dP_{ML}}{dx}\).

*Ill posed for infinite dimensional \(\varphi(x)!*)
Loss is \textbf{Fisher Score:}

$$D_F(p_0, p_f) := \frac{1}{2} \int p_0(x) \| \nabla_x \log p_0(x) - \nabla_x \log p_f(x) \|^2 \, dx$$
Score matching (general version)

Assuming $p_f$ to be differentiable (w.r.t. $x$) and
\[ \int p_0(x) \| \nabla_x \log p_f(x) \|^2 \, dx < \infty, \forall \theta \in \Theta \]

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

\[
\overset{(a)}{=} \int p_0(x) \sum_{i=1}^{d} \left( \frac{1}{2} \left( \frac{\partial \log p_f(x)}{\partial x_i} \right)^2 + \frac{\partial^2 \log p_f(x)}{\partial x_i^2} \right) \, dx
\]

\[+ \frac{1}{2} \int p_0(x) \left\| \frac{\partial \log p_0(x)}{\partial x} \right\|^2 \, dx \]

where partial integration is used in (a) under the condition that
\[ p_0(x) \frac{\partial \log p_f(x)}{\partial x_i} \to 0 \text{ as } x_i \to \pm \infty, \forall i = 1, \ldots, d \]
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 $\lambda ||f||_H^2$ 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:

\[ p_f(x) = e^{\langle f, \phi(x) \rangle_{\mathcal{H}} - A(f)} q_0(x) \]

Thus

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

Kernel trick for derivatives:

\[ \frac{\partial}{\partial x_i} f(X) = \left\langle f, \frac{\partial}{\partial x_i} \phi(X) \right\rangle_{\mathcal{H}} \]

Dot product between feature derivatives:

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

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

Need to solve a linear system

\[ \beta_n^* = -\frac{1}{\lambda} \left( G_{XX} + n\lambda I \right)^{-1} h_X \]

Very costly in high dimensions!
The Nystrom approximation
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:

\[
\beta_{n,m}^* = - \left( \frac{1}{n} B_{XY} B_{XY} + \lambda G_{YY} \right)^\dagger h_Y
\]

Solve in time \( O(n m^2 d^3) \), evaluate in time \( O(md) \).
- Sill cubic in \( d \), but similar results if we take a random dimension per datapoint.
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
  $$D_F(p_0, p_{fn}) = O_{p_0} \left( n^{-\min\left\{ \frac{2}{3}, \frac{\beta}{2(\beta+1)} \right\}} \right)$$

- 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
  $$D_F(p_0, p_{fn,m}) = O_p\left(n^{-\min\left\{\frac{2}{3}, \frac{\beta}{2(\beta+1)}\right\}}\right)$$
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
The kernel conditional exponential family

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

\[
P(X_1, ..., X_d) = \prod_i P(X_i \mid \underbrace{X_{\pi(i)}}_{\text{parents of } X_i})
\]

- Estimate each factor independently
Kernel conditional exponential family

General definition, kernel conditional exponential family

[Smola and Canu, 2006]

\[ p_f(y|x) = e^{\langle f, \psi(x,y) \rangle \mathcal{H} - A(f, x)} q_0(y) \quad A(f, x) = \log \int q_0(y) e^{\langle f, \psi(x,y) \rangle \mathcal{H}} dy \]

(joint feature map \( \psi(x, y) \))
Kernel conditional exponential family

Our kernel conditional exponential family:

\[
p_f(x) = e^{\langle f_x, \phi(y) \rangle_G} - A(f, x) q_0(y) \quad A(f, x) = \log \int q_0(y) e^{\langle f_x, \phi(y) \rangle_G}
\]

linear in the sufficient statistic \( \phi(y) \in \mathcal{G} \).
Kernel conditional exponential family

Our kernel conditional exponential family:

\[ p_f(x) = e^{\langle f_x, \phi(y) \rangle_G - A(f, x)} q_0(y) \quad A(f, x) = \log \int q_0(y) e^{\langle f_x, \phi(y) \rangle_G} \]

linear in the sufficient statistic \( \phi(y) \in G \).

What does this RKHS look like?

[Micchelli and Pontil, (2005)]

\[
\langle f_x, \phi(y) \rangle_G \\
= \langle \Gamma_x^* f, \phi(y) \rangle_G \\
= \langle f, \Gamma_x \phi(y) \rangle_H
\]
Kernel conditional exponential family

Our kernel conditional exponential family:

\[ p_f(x) = e^{\langle f_x, \phi(y) \rangle_G - A(f, x)} q_0(y) \]
\[ A(f, x) = \log \int q_0(y) e^{\langle f_x, \phi(y) \rangle_G} \]

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

What does this RKHS look like?

[Micchelli and Pontil, (2005)]

\[ \langle f_x, \phi(y) \rangle_G = \langle \Gamma_x^* f, \phi(y) \rangle_G = \langle f, \Gamma_x \phi(y) \rangle_\mathcal{H} \]

- \( \Gamma_x^* : \mathcal{H} \rightarrow \mathcal{G} \) is a linear operator
Kernel conditional exponential family

Our kernel conditional exponential family:

\[ p_f(x) = e^{\langle f_x, \phi(y) \rangle_G - A(f, x)} q_0(y) \quad A(f, x) = \log \int q_0(y) e^{\langle f_x, \phi(y) \rangle_G} \]

linear in the sufficient statistic \( \phi(y) \in G \).

What does this RKHS look like?

[Micchelli and Pontil, (2005)]

\[ \langle f_x, \phi(y) \rangle_G = \langle \Gamma^*_x f, \phi(y) \rangle_G = \langle f, \Gamma_x \phi(y) \rangle_H \]

- \( \Gamma_x : G \rightarrow H \) is a linear operator.
- The feature map \( \psi(x, y) := \Gamma_x \phi(y) \)
What is our loss function?

The obvious approach: minimise

\[ D_F [p_0(x)p_0(y|x)\|p_f(x)p_f(y|x)] \]

Problem: the expression still contains \( \int p_0(y|x)dy \).
What is our loss function?

The obvious approach: minimise

$$D_F [p_0(x)p_0(y|x)||p_f(x)p_f(y|x)]$$

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

Our loss function:

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

for some $\pi(x)$ that includes the support of $p(x)$. 
Finite sample estimate of the conditional density

Use the simplest operator-valued RKHS $\Gamma_x = I_G k(x, \cdot)$.

\[ \Gamma_x : \mathcal{G} \rightarrow \mathcal{H} \]
\[ \Gamma_x \phi(y) \mapsto \phi(y) k(x, \cdot) \]
Finite sample estimate of the conditional density

Use the simplest operator-valued RKHS $\Gamma_x = I_G k(x, \cdot)$.

$$\Gamma_x : G \rightarrow \mathcal{H}$$

$$\Gamma_x \phi(y) \mapsto \phi(y) k(x, \cdot)$$

Solution:

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

where

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

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

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

30/39
Expected conditional score: a failure case

- $P(Y|X = 1)$
- $P(Y|X = -1)$
- $P(Y) = \frac{1}{2}(P(Y|X = 1) + P(Y|X = -1))$
Expected conditional score: a failure case

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

$$\tilde{D}_F(p(y|x), p(y)) = 0$$
Expected conditional score: a failure case

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

\[
\tilde{D}_F(p(y|x), p(y)) = 0
\]

\[\text{target} \quad \text{model}\]
Expected conditional score: a failure case

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

\[
\tilde{D}_F(p(y|x), p(y)) = 0
\]
Expected conditional score: a failure case

Why does it fail? Recall

\[ \tilde{D}_F(p_0(y|x), p_f(y|x)) := \int \pi(x) D_F(p_0(y|x), p_f(y|x)) dx \]

Note that

\[ D_F(p(y|x = 1), p(y)) = \int p(y|1) \| \nabla_x \log p(y|1) - \nabla_x \log p(y) \|^2 dy \]

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.

<table>
<thead>
<tr>
<th></th>
<th>Parkinsons</th>
<th>Red Wine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td>Samples</td>
<td>5875</td>
<td>1599</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th></th>
<th>Parkinsons</th>
<th>Red wine</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCEF</td>
<td>$2.86 \pm 0.77$</td>
<td>$11.8 \pm 0.93$</td>
</tr>
<tr>
<td>LSCDE</td>
<td>$15.89 \pm 1.48$</td>
<td>$14.43 \pm 1.5$</td>
</tr>
<tr>
<td>NADE</td>
<td>$3.63 \pm 0.0$</td>
<td>$9.98 \pm 0.0$</td>
</tr>
</tbody>
</table>
Results: unconditional model

Red Wine

Parkinsons

Data
KEF

$X_7$

$X_6$

$X_{16}$

$X_{15}$
Results: conditional model

Red Wine

Data

KCEF

-6 -4 -2 0 2 4 x 6

Parkinsons

Data

KCEF

-6 -4 -2 0 2 4 x 6
Co-authors

From Gatsby:
- Michael Arbel
- Heiko Strathmann
- Dougal Sutherland

External collaborators:
- Kenji Fukumizu
- Bharath Sriperumbudur

Questions?
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

\[ D_F(p_0, p_f) \]
\[ = \frac{1}{2} \int_a^b p_0(x) \left( \frac{d \log p_0(x)}{dx} - \frac{d \log p_f(x)}{dx} \right)^2 dx \]
\[ = \frac{1}{2} \int_a^b p_0(x) \left( \frac{d \log p_0(x)}{dx} \right)^2 dx + \frac{1}{2} \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right)^2 dx \]
\[ - \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{d \log p_0(x)}{dx} \right) dx \]
Score matching: 1-D proof

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

Final term:
\[ \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{d \log p_0(x)}{dx} \right) \, dx \]
\[ = \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{1}{p_0(x)} \frac{dp_0(x)}{dx} \right) \, dx \]
\[ = \left[ \left( \frac{d \log p_f(x)}{dx} \right) p_0(x) \right]^b_a - \int_a^b p_0(x) \frac{d^2 \log p_f(x)}{dx^2} \, dx. \]
Score matching: 1-D proof

\[ D_F(p_0, p_f) \]

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

\[ = \frac{1}{2} \int_a^b p_0(x) \left( \frac{d \log p_0(x)}{dx} \right)^2 dx + \frac{1}{2} \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right)^2 dx \]

\[ - \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{d \log p_0(x)}{dx} \right) dx \]

Final term:

\[ \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{d \log p_0(x)}{dx} \right) dx \]

\[ = \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{1}{p_0(x)} \frac{dp_0(x)}{dx} \right) dx \]

\[ = \left[ \left( \frac{d \log p_f(x)}{dx} \right) p_0(x) \right]_a^b - \int_a^b p_0(x) \frac{d^2 \log p_f(x)}{dx^2} dx \]
Score matching: 1-D proof

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

Final term:

\[
\int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{d \log p_0(x)}{dx} \right) \, dx \\
= \int_a^b p_0(x) \left( \frac{d \log p_f(x)}{dx} \right) \left( \frac{1}{p_0(x)} \frac{dp_0(x)}{dx} \right) \, dx \\
= \left[ \left( \frac{d \log p_f(x)}{dx} \right) p_0(x) \right]_a^b - \int_a^b p_0(x) \frac{d^2 \log p_f(x)}{dx^2} \, dx.
\]