Scalable, Flexible and Active Learning on Distributions

Dougal J. Sutherland

Thesis Committee:
Jeff Schneider (Chair)
Nina Balcan
Barnabás Póczos
Arthur Gretton (UCL)
Learning on Distributions

We want to learn a distribution classification function....
Learning on Distributions

...based on sample sets.
Learning on Distributions

distribution

observed sample

label

9 components

“seaside city”

and more...

Mass $7 \times 10^{14} \ M_\odot$

no Cs137 present

county voted
54% for Obama
Contributions

• **Learning on distributions** with nonparametric kernels

• **Scalable** approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• **Flexible** distribution kernels
  – Deep mean maps in computer vision
  – MMD kernel learning for testing

• **Active** pointillistic pattern search
Kernel Methods

\[ \phi : \mathbb{R}^d \rightarrow \mathcal{H} \]

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

\[ f(x) = w^T x + b \]

Linear models...

\[ f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}} + b \]

\[ = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b \]

...in Hilbert space.
Kernel Methods

Can use a kernel on any domain.

\[ \phi : \mathcal{X} \rightarrow \mathcal{H} \]

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

\[ f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}} + b \]

\[ = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b \]

Linear models...in Hilbert space.
Kernels on Distributions

We’ll use a kernel on distributions based on a distance $\rho$:

$$K(\mathbb{P}, \mathbb{Q}) = \exp \left( - \frac{1}{2\sigma^2} \rho^2(\mathbb{P}, \mathbb{Q}) \right)$$

The popular Gaussian RBF kernel has this form, with $\rho$ the Euclidean distance between vectors.

A valid kernel as long as $\rho$ is Hilbertian.
Distances on Distributions

$p(x)$

$q(x)$
Distances on Distributions

$$TV(\mathbb{P}, \mathbb{Q}) = \int \frac{1}{2} |p(x) - q(x)| \, dx$$
Distances on Distributions

Total Variation

\[ TV(p, q) = \int \frac{1}{2} |p(x) - q(x)| \, dx \]

\[ L_2(p, q) = \int (p(x) - q(x))^2 \, dx \]

Hellinger

\[ H^2(p, q) = \int \frac{1}{2} \left( \sqrt{p(x)} - \sqrt{q(x)} \right)^2 \, dx \]

Kullback-Leibler

\[ KL(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} \, dx \]

Rényi-\(\alpha\)

\[ R_{\alpha}(p \parallel q) = \frac{1}{\alpha - 1} \int p(x)^\alpha q(x)^{1-\alpha} \, dx \]

Jensen-Shannon

\[ JS(p, q) = \frac{1}{2} KL \left( p \parallel \frac{p + q}{2} \right) + \frac{1}{2} KL \left( q \parallel \frac{p + q}{2} \right) \]

Maximum Mean Discrepancy

\[ MMD_k(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{H}_k} \mathbb{E}_{X \sim \mathbb{P}} f(X) - \mathbb{E}_{Y \sim \mathbb{Q}} f(Y) \]
Estimators of Distributional Distances

• Fit a parametric model and compute distances.
  – Some distances have closed form for some models.
  – Model introduces approximation error

\[
L_2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')) = \frac{1}{|4\pi \Sigma|^{\frac{1}{2}}} + \frac{1}{|4\pi \Sigma'|^{\frac{1}{2}}} - 2 \frac{\exp\left(-\frac{1}{2}(\mu - \mu')^T(\Sigma + \Sigma')^{-1}(\mu - \mu')\right)}{|2\pi(\Sigma + \Sigma')|^{\frac{1}{2}}}
\]

Jebara et al. 2004; Moreno et al. 2004
Estimators of Distributional Distances

• Fit a nonparametric model and compute distances.
  • Histograms
  • Kernel density estimation
  • $k$-nearest neighbor density estimation
  • Basis function projections
  • Empirical distribution (for MMD)
Algorithm: Learning on Distributions

Given sample sets $X_i \sim \mathbb{P}_i$, a distance $\rho$, and a b.w. $\sigma$:

1. Estimate $\hat{\rho}(X_i, X_j)$ for all $i, j$ nonparametrically.

2. Assemble into a kernel matrix

$$K_{ij} = \exp \left( -\frac{1}{2\sigma^2} \hat{\rho}(X_i, X_j) \right).$$

3. Run an SVM / GP / ridge regression / ... with $K$. 
Application: Galaxy Cluster Mass

Galaxy clusters are fundamental in the study of the universe. Their properties can tell us a lot about cosmology.

But they’re mostly dark matter; measuring their mass is hard.

Coma cluster, NASA
Application: Galaxy Cluster Mass

Fritz Zwicky (1933): Under reasonable assumptions, the velocity dispersion has power-law relationship to total mass.

Not so great...
- Assumptions violated
- Which galaxies are in cluster?
Application: Galaxy Cluster Mass

Alternative approach: consider each cluster as a *distribution* of galaxy features, and regress from these distributions to total mass.

Cluster Mass: Known Membership

- power law
- kNN-KL with $|v|$ only
Cluster Mass: With Interlopers

- Power law
- kNN-KL with |v| only
Contributions

• **Learning on distributions** with nonparametric kernels

• **Scalable** approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• **Flexible** distribution kernels
  – Deep mean maps in computer vision
  – MMD kernel learning for testing

• **Active** pointillistic pattern search
Scalability

These methods need an $n \times n$ Gram matrix:

$$K = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \ldots & K(x_1, x_n) \\ K(x_2, x_1) & K(x_2, x_2) & \ldots & K(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_n, x_1) & K(x_n, x_2) & \ldots & K(x_n, x_n) \end{bmatrix}$$

The matrix is $n^2$; operations can be as slow as $O(n^3)$.

Linear-kernel models usually scale like $O(n)$. 
Approximate Embeddings

Traditional kernel methods:

\[ \{ x_i \} \quad \{ \varphi(x_i) \} \subset \mathcal{H} \]
\[ \langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}} = k(x_i, y_j) \]
\[ f(x) = \langle w, \varphi(x) \rangle + b = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b \]

Approximate kernel embeddings:

\[ \{ x_i \} \quad \{ z(x_i) \} \subset \mathbb{R}^D \]
\[ z(x_i)^T z(x_j) \]
\[ f(x) = w^T z(x) + b \]

\[ K(\cdot, \cdot) \approx z(\cdot)^T z(\cdot) \]
Random Fourier Features

Rahimi and Recht (2007) developed random Fourier features, a.k.a. “random kitchen sinks”:

\[
    k(x, y) = \mathcal{K}\left(\Delta\right) = \exp\left(-\frac{1}{2\sigma^2}\|\Delta\|^2\right)
\]

\[
    = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}} \quad \text{(dim } \mathcal{H} = \infty) \\
    \approx z(x)^T z(y) \quad \text{ (} z : \mathbb{R}^d \to \mathbb{R}^D \text{)}
\]
Random Fourier Features

Rahimi and Recht (2007) developed random Fourier features, a.k.a. “random kitchen sinks”:

\[
k(x, y) = k(\Delta) = \exp\left(-\frac{1}{2\sigma^2}\|\Delta\|^2\right) \approx z(x)^T z(y)
\]

\[
\Omega(\omega) := \int k(\Delta) \exp(-i\omega^T\Delta) \, d\Delta \propto \exp\left(-\frac{\sigma^2}{2}\|\Delta\|^2\right)
\]

Bochner’s theorem: \( \Omega \) is a (scaled) probability distribution.

\[
\omega_i \sim \Omega
\]

\[
\tilde{z}(x) = \sqrt{\frac{2}{D}} \left[ \sin(\omega_1^T x) \cos(\omega_1^T x) \ldots \sin(\omega_D^T /2 x) \cos(\omega_D^T /2 x) \right]^T
\]

\[
\tilde{z}(x) = \sqrt{\frac{2}{D}} \left[ \cos(\omega_1^T x + b_1) \ldots \cos(\omega_D^T x + b_D) \right]^T b_i \sim \text{Unif}(0, 2\pi)
\]
Random Fourier Features

\[ \tilde{z}(x) = \sqrt{\frac{2}{D}} \begin{bmatrix} \sin(\omega_1^T x) & \cos(\omega_1^T x) & \ldots & \sin(\omega_{D/2}^T x) & \cos(\omega_{D/2}^T x) \end{bmatrix}^T \]

\[ \tilde{z}(x) = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos(\omega_1^T x + b_1) & \ldots & \cos(\omega_D^T x + b_D) \end{bmatrix}^T b_i \sim \text{Unif}(0, 2\pi) \]

The L_\infty error is also tighter, in bounds and empirically.
Contributions

• Learning on distributions with nonparametric kernels

• **Scalable** approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• **Flexible** distribution kernels
  – Deep mean maps in computer vision
  – MMD kernel learning for testing

• **Active** pointillistic pattern search
Mean embeddings of distributions

The *mean embedding* of a distribution in an RKHS:

\[ \mu_P = \mathbb{E}_{x \sim P} [\varphi(x)] \]

Remember \( \langle \varphi(x), \varphi(y) \rangle = k(x, y) \), so we can think of \( \varphi(x) \) as \( k(x, \cdot) \).
Mean embeddings of distributions

The *mean embedding* of a distribution in an RKHS:

\[ \mu_P = \mathbb{E}_{x \sim P}[\varphi(x)] \]

Remember \( \langle \varphi(x), \varphi(y) \rangle = k(x, y) \), so we can think of \( \varphi(x) \) as \( k(x, \cdot) \).
Mean embeddings of distributions

The *mean embedding* of a distribution in an RKHS:

\[ \mu_\mathbb{P} = \mathbb{E}_{x \sim \mathbb{P}}[\varphi(x)] \]

Remember \( \langle \varphi(x), \varphi(y) \rangle = k(x, y) \), so we can think of \( \varphi(x) \) as \( k(x, \cdot) \).
Mean embeddings of distributions

The *mean embedding* of a distribution in an RKHS:

\[
\mu_P = \mathbb{E}_{x \sim P} [\varphi(x)]
\]

Remember \( \langle \varphi(x), \varphi(y) \rangle = k(x, y) \), so we can think of \( \varphi(x) \) as \( k(x, \cdot) \).
Mean embeddings of distributions

The *mean embedding* of a distribution in an RKHS:

$$\mu_P = \mathbb{E}_{x \sim P}[\varphi(x)]$$

Remember $\langle \varphi(x), \varphi(y) \rangle = k(x, y)$, so we can think of $\varphi(x)$ as $k(\cdot, x)$. 
Maximum Mean Discrepancy (MMD)

The MMD is the distance between mean embeddings:

\[
\mu_P = \mathbb{E}_{X \sim P}[\varphi(X)]
\]

\[
\text{MMD}^2(P, Q) = \|\mu_P - \mu_Q\|_\mathcal{H}^2
\]

\[
= \langle \mu_P, \mu_P \rangle + \langle \mu_Q, \mu_Q \rangle - 2\langle \mu_P, \mu_Q \rangle
\]

\[
\langle \mu_P, \mu_Q \rangle = \langle \mathbb{E}_{X \sim P}[\varphi(X)], \mathbb{E}_{Y \sim Q}[\varphi(Y)] \rangle
\]

\[
= \mathbb{E}_{X \sim P} \mathbb{E}_{Y \sim Q} [\langle \varphi(X), \varphi(Y) \rangle]
\]

\[
= \mathbb{E}_{X \sim P} [k(X, Y)]
\]

Estimator: \[ \langle \hat{\mu}_P, \hat{\mu}_Q \rangle = \frac{1}{mn} \sum_{ij} k(X_i, Y_j) \]
Embedding MMD

\[ \widehat{\text{MMD}}(X, Y) = \| \hat{\mu}_P - \hat{\mu}_Q \|_H \]

\[ \langle \hat{\mu}_P, \hat{\mu}_Q \rangle = \frac{1}{mn} \sum_{ij} k(X_i, Y_j) \]

\approx \frac{1}{mn} \sum_{ij} z(X_i)^T z(Y_j)

= \left[ \frac{1}{m} \sum_{i} z(X_i) \right]^T \left[ \frac{1}{n} \sum_{j} z(Y_j) \right] = \left( \frac{1}{m} \mathbb{1} \right)^T \left( \frac{1}{n} \mathbb{1} \right)

= \| \bar{z}(X) \| \| \bar{z}(Y) \|

\widehat{\text{MMD}}_z(X, Y) = \| \bar{z}(X) - \bar{z}(Y) \|

\exp \left( -\frac{1}{2\sigma^2} \widehat{\text{MMD}}_z(X, Y) \right) \approx z(\bar{z}(X))^T z(\bar{z}(Y)) \]
Embedding $L_2$

Oliva, Neiswanger, Póczos, Schneider, Xing (AISTATS 2014) gave an embedding for

$$K(\hat{p}, \hat{q}) = \exp \left( -\frac{1}{2\sigma^2} \|\hat{p} - \hat{q}\|_{L_2}^2 \right)$$

based on projection coefficients onto an orthonormal basis for $L_2$. 

Embedding HDDs

\[ \rho^2(p, q) = \int_X \kappa(p(x), q(x)) \, dx \]

Homogeneous Density Distances

\( \rho^2 \) can be:

- Jensen-Shannon
- Total Variation
- Squared Hellinger
Algorithm: HDD Embedding

\[ \rho^2(p, q) = \int_X \kappa(p(x), q(x)) \, dx \quad \text{Homogeneous Density Distances} \]

1. Approximately embed \( \rho \) into \( L_2 \)
    - by embedding \( \kappa \) into \( L_2 \) (Fuglede 2005).

\[ \rho(p, q) \approx \| \psi(p) - \psi(q) \|_{L_2^M} \]

\( \rho \) on distributions \( \approx L_2 \) distance on random \( \psi \) functions

2. Approximately embed \( L_2 \) into \( \mathbb{R}^m \).

\[ \| \psi(p) - \psi(q) \|_{L_2^M} \approx \| A(p) - A(q) \|_{\mathbb{R}^{2M} | V} \]

\( L_2 \) distance on random \( \psi \) functions \( \approx \) Euclidean distance on \( A \) vectors

3. Use random Fourier features to embed \( K \) on \( \mathbb{R}^m \) into \( \mathbb{R}^D \).

\[ \exp \left( - \frac{1}{2\sigma^2} \rho^2(p, q) \right) \approx z(A(p))^T z(A(q)) \]

RBF kernel with \( \rho \approx \) random Fourier features of \( A \) vectors
HDD estimator convergence

The embedding is an estimator for the kernel.

For fixed $p$ and $q$, we have a finite-sample bound on

$$\Pr \left( \left| K(p, q) - z(\hat{A}(\hat{p}))^T z(\hat{A}(\hat{q})) \right| \geq \varepsilon \right)$$

which behaves as expected:

- Lower for smoother, lower-dimensional densities
- Lower for more samples
- More projection coefficients / samples from $\mu$:
  - better approximation, harder integration
Application: Gaussian Mixtures

Train with 4K, 8K, 16K distributions; test on 2K. 200 iid points each.

AIC: 2.7
BIC: 3.8
Mean: 2.8
Application: Nuclear Threat Detection

Small sensors and cluttered environments make lots of challenges for traditional detection algorithms.
Application: Nuclear Threat Detection
Application: Scene Classification

Scene-15 dataset (4,485 images)
Application: Scene Classification

Features from a deep convnet (16 layers):

Local features here for overlapping patches of input image
Application: Scene Classification

Using $A(p)$ features
Contributions

• Learning on distributions with nonparametric kernels

• Scalable approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• Flexible distribution kernels
  – Deep mean maps in computer vision
  – MMD kernel learning for testing

• Active pointillistic pattern search
Deep distribution kernel learning

We can put distribution embeddings in a deep network:

\[
x_1, x_2, \ldots, x_m
\]

\[
f(x_1), f(x_2), \ldots, f(x_m)
\]

\[
z
\]

\[
\hat{y}
\]

1. Convolve with frequencies
2. Sine/cosines
3. Global pooling

Fixed frequencies
Learn scale
Learn frequencies

Initial results for scene classification: small but consistent improvement.
Contributions

• **Learning on distributions** with nonparametric kernels

• **Scalable** approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• **Flexible** distribution kernels
  – Deep mean maps in computer vision
  – MMD kernel learning for testing

• **Active** pointillistic pattern search
Two-sample testing

Observed samples: \( X \sim P \quad Y \sim Q \)

Hypothesis test on unobserved distributions: \( P \overset{?}{=} Q \)

Applications:

- Neuroscience: do these areas of the brain behave differently under different conditions?
- Schema alignment: do these two database columns mean the same thing?
- Many more...
Two-sample testing

Test with MMD:

\[ X \sim \mathbb{P} = \mathcal{N}(0, 1) \]
\[ Y \sim \mathbb{Q} = \text{Laplace}\left(0, \frac{1}{\sqrt{2}}\right) \]

Test: reject if \( m \hat{\text{MMD}}^2(\mathbb{P}, \mathbb{Q}) > c_\alpha \).
Permutation testing

When $\mathbb{P} = \mathbb{Q}$, MMD is asymptotically gross, so hard to find a threshold $c_\alpha$ that way.

Use a permutation test:

$$X = \{x_1, x_2, \ldots, x_m\} \quad Y = \{y_1, y_2, \ldots, y_m\}$$

Split randomly to estimate MMD when $\mathbb{P} = \mathbb{Q}$:

$$\hat{m}\text{MMD}^2(X^{(1)}, Y^{(1)}) \quad \hat{m}\text{MMD}^2(X^{(2)}, Y^{(2)})$$

$\hat{c}_\alpha$: $(1-\alpha)$th quantile of $m\text{MMD}^2(X^{(i)}, Y^{(i)})$
The kernel matters!

*Witness function* $f$ helps compare samples:

\[
\text{MMD}(\mathcal{P}, \mathcal{Q}) = \mathbb{E}_{X \sim \mathcal{P}} f(X) - \mathbb{E}_{Y \sim \mathcal{Q}} f(Y)
\]

\[
f(x) = \mathbb{E}_{X \sim \mathcal{P}} k(x, X) - \mathbb{E}_{Y \sim \mathcal{Q}} k(x, Y)
\]

\[
\sigma = 0.75; p = 0.04
\]
The kernel matters!

*Witness function* $f$ helps compare samples:

$$\text{MMD}(\mathcal{P}, \mathcal{Q}) = \mathbb{E}_{X \sim \mathcal{P}} f(X) - \mathbb{E}_{Y \sim \mathcal{Q}} f(Y)$$

$$f(x) = \mathbb{E}_{X \sim \mathcal{P}} k(x, X) - \mathbb{E}_{Y \sim \mathcal{Q}} k(x, Y)$$

- $\sigma = 0.75; p = 0.04$
- $\sigma = 2; p = 0.43$
The kernel matters!

*Witness function* \( f \) helps compare samples:

\[
\text{MMD}(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{X \sim \mathbb{P}} f(X) - \mathbb{E}_{Y \sim \mathbb{Q}} f(Y)
\]

\[
f(x) = \mathbb{E}_{X \sim \mathbb{P}} k(x, X) - \mathbb{E}_{Y \sim \mathbb{Q}} k(x, Y)
\]

\[
\sigma = 0.75; p = 0.04
\]

\[
\sigma = 2; p = 0.43
\]

\[
\sigma = 0.1; p = 0.16
\]
Choosing a kernel

So we need a way to pick a kernel to do the test.

Before:

$\begin{align*}
X & \quad Y \\
\text{MMD with fixed } k
\end{align*}$
Choosing a kernel

So we need a way to pick a kernel to do the test.

Split data:

Choose a kernel $k$

Chosen $k$ in MMD test

How to pick $k$? Typically: maximize MMD.

But we want the (asymptotically) most powerful test.
Asymptotic power of MMD

When $\mathbb{P} \neq \mathbb{Q}$, MMD is asymptotically normal:

$$\frac{\hat{\text{MMD}}^2 - \text{MMD}^2}{\sqrt{V_m}} \xrightarrow{D} \mathcal{N}(0, 1) \quad V_m = \text{Var}_{X \sim P^m, Y \sim Q^m} \left[ \hat{\text{MMD}}^2(X, Y) \right]$$

and we can analyze the power:

$$\Pr_{H_1} \left( m\hat{\text{MMD}}^2 > \hat{c}_\alpha \right)$$
MMD $t$-statistic

$$\Pr_{H_1} \left( m\hat{\text{MMD}}^2 > \hat{c}_\alpha \right) \to 1 - \Phi \left( \frac{c_\alpha}{m\sqrt{V_m}} - \frac{\text{MMD}^2}{\sqrt{V_m}} \right)$$

So we can maximize the power by maximizing

$$\tau_U = \frac{\text{MMD}^2}{\sqrt{V_m}} - \frac{c_\alpha}{m\sqrt{V_m}}$$

$$\hat{\tau}_U = \frac{\text{MMD}^2}{\sqrt{\hat{V}_m}} - \frac{\hat{c}_\alpha}{m\sqrt{\hat{V}_m}}$$

But $V_m$ is $O(1/m)$, so the first term dominates for large $m$, and we should be able to get away with maximizing

$$t_U = \frac{\text{MMD}^2}{\sqrt{V_m}}$$

$$\hat{t}_U = \frac{\text{MMD}^2}{\sqrt{\hat{V}_m}}$$
**t-statistic estimator**

\[
\hat{\tau}_U = \frac{\text{MMD}^2}{\sqrt{\hat{V}_m}} - \frac{\hat{c}_\alpha}{m\sqrt{\hat{V}_m}}
\]

\[
\text{MMD}^2 := \frac{1}{\binom{m}{2}} \sum_{i \neq j} k(X_i, X_j) + k(Y_i, Y_j) - k(X_i, Y_j) - k(X_j, Y_i)
\]

\(\hat{c}_\alpha\) is from a permutation test, so the average of a bunch of MMD estimates
\[ \hat{V}_m = \frac{4(m-2)}{m(m-1)} \hat{\zeta}_1 + \frac{2}{m(m-1)} \hat{\zeta}_2 \]

\[ \hat{\zeta}_1 = \frac{1}{m(m-1)(m-2)} \left( 1^T \tilde{K}_{XX} \tilde{K}_{XX} 1 - \| \tilde{K}_{XX} \|_F^2 \right) - \left( \frac{1}{m(m-1)} 1^T \tilde{K}_{XX} 1 \right)^2 \]

\[ - \frac{2}{m^2(m-1)} 1^T \tilde{K}_{XX} K_{XY} 1 + \frac{2}{m^3(m-1)} 1^T \tilde{K}_{XX} 1 1^T K_{XY} 1 \]

\[ + \frac{1}{m(m-1)(m-2)} \left( 1^T \tilde{K}_{YY} \tilde{K}_{YY} 1 - \| \tilde{K}_{YY} \|_F^2 \right) - \left( \frac{1}{m(m-1)} 1^T \tilde{K}_{YY} 1 \right)^2 \]

\[ - \frac{2}{m^2(m-1)} 1^T \tilde{K}_{YY} K_{XY}^T 1 + \frac{2}{m^3(m-1)} 1^T \tilde{K}_{YY} 1 1^T K_{XY} 1 \]

\[ + \frac{1}{m^2(m-1)} \left( 1^T K_{XY}^T K_{XY} 1 - \| K_{XY} \|_F^2 \right) - 2 \left( \frac{1}{m^2} 1^T K_{XY} 1 \right)^2 \]

\[ + \frac{1}{m^2(m-1)} \left( 1^T K_{XY} K_{XY}^T 1 - \| K_{XY} \|_F^2 \right) \]

\[ \hat{\zeta}_2 = \frac{1}{m(m-1)} \| \tilde{K}_{XX} + \tilde{K}_{YY} - \tilde{K}_{XY} - \tilde{K}_{XY}^T \|_F^2 \]
$t$-statistic estimator

Can even get gradients of $t_U$ and (with some more effort) $\tau_U$, to help maximize it.

(automatic differentiation is your friend)
Kernel choice on Blobs

Blobs dataset:

Mixture of $\mathcal{N}(\mu_{ij}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$ vs Mixture of $\mathcal{N}(\mu_{ij}, \begin{bmatrix} 1 & \frac{\varepsilon - 1}{\varepsilon + 1} \\ \frac{\varepsilon - 1}{\varepsilon + 1} & 1 \end{bmatrix})$

When $\varepsilon=1$, $P = Q$; this picture has $\varepsilon=6$. 
Kernel choice on Blobs

- best choice
- MMD
- $\hat{t}_U$
- $\hat{\tau}_U$

Rejection rate

$m = 500$
Deep Kernels

Map through layers of a deep network:

\[
x_1 \rightarrow f(x_1) \\
x_2 \rightarrow f(x_2) \\
\vdots \\
x_m \rightarrow f(x_m)
\]

\[
y_1 \rightarrow f(y_1) \\
y_2 \rightarrow f(y_2) \\
\vdots \\
y_m \rightarrow f(y_m)
\]

MMD with Gaussian RBF

\[\hat{t}_U \text{ or } \hat{\tau}_U\]
Generative adversarial networks:

- **Generator** comes up with samples; trained to trick the adversary.
- **Adversary** tries to distinguish between generator sample and true data; trained to beat the generator.

But adversary is really just a two-sample test.

Kernel learning helps prevent the generator’s tricks.

<table>
<thead>
<tr>
<th>Generated</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Generator** vs **Adversary**
Contributions

• **Learning on distributions** with nonparametric kernels

• **Scalable** approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• **Flexible** distribution kernels
  – Deep mean maps in computer vision
  – MMD kernel learning for testing

• **Active** pointillistic pattern search
Active Pointillistic Pattern Search

Search for region patterns with point observations.

Ma*, Sutherland*, Garnett, Schneider, AISTATS 2015. (*: equal contribution)
Active Pointillistic Pattern Search

Ma*, Sutherland*, Garnett, Schneider, AISTATS 2015. (*: equal contribution)
Active Pointillistic Pattern Search

Ma*, Sutherland*, Garnett, Schneider, AISTATS 2015. (*: equal contribution)
Active Pointillistic Pattern Search

![Graph showing recall for matching regions against the number of data points collected.](image)

Ma*, Sutherland*, Garnett, Schneider, AISTATS 2015. (*: equal contribution)
Take-Home Messages

• Think about how you model your data.
• Distributions and sets can work pretty well.
  – Cosmology, nuclear threat detection, scene classification, parametric statistical inference, polling, autonomous sensing...
• Random embeddings can help with scalability...
  – if you use random Fourier features, use the right one
• ...and with flexibility
  – Plug the \textsc{MMD} embedding into deep learning and go crazy
Things Still to Do

• Deep kernel learning
  – Different parameterizations of kernels

• More applications!
  – Word and document embeddings
  – Kernel-learning two-sample test as adversary in a GAN

• Active learning on distributions
Thanks!

Rahul De

Ben Johnson
Contributions

• **Learning on distributions** with nonparametric kernels

• **Scalable** approximate kernel embeddings
  – Random Fourier features analysis
  – New embeddings for distribution kernels

• **Flexible** distribution kernels
  – Deep mean maps in computer vision
  – **MMD** kernel learning for testing

• **Active** pointillistic pattern search