Linear-time Learning on Distributions with Approximate Kernel Embeddings

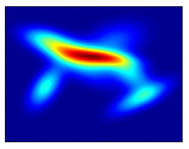
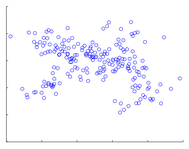
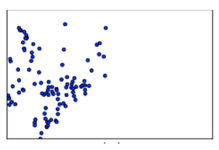
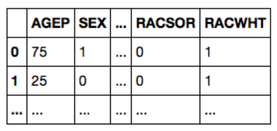
Dougal J. Sutherland*, Junier B. Oliva*, Barnabás Póczos, Jeff Schneider
*: equal contribution

Carnegie Mellon University

Auton Lab

Learning on Distributions

We want to do machine learning where the inputs are *distributions*, instead of vectors.

distribution	observed sample	label
		9 components
		"seaside city"
 Ntampaka et al. (ApJ 2015)		$\log M = 14.63$
 Flaxman et al. (KDD 2015)		county voted 54% for Obama

	AGEP	SEX	...	RACSOR	RACWHT
0	75	1	...	0	1
1	25	0	...	0	1
...	...	...	...	...	...

and more...

Distribution Kernels

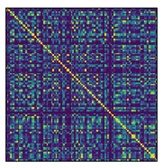
Preferred approach:

- Choose a distance on distributions d
- Make a kernel with $K(p, q) = \exp(-\gamma d^2(p, q))$

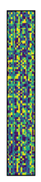
Distances to use:

- L_2 between densities
- Maximum mean discrepancy (MMD)
- Total variation (TV)
- Hellinger distance (H)
- Jensen-Shannon divergence (JS; based on KL)

Approximate Embeddings

Needs an $N \times N$ kernel matrix: 

If N is big, way too slow.

Instead, use an $N \times D$ feature matrix that approximates K: 

Then primal-space solvers run quickly.

$K($  ,  $) \approx z($  $)^\mathsf{T} z($  $)$

Our Contribution

An **approximate embedding** for **distributional kernels** based on total variation, Hellinger, and Jensen–Shannon distances.

Setup

Embeddings exist for L_2 and MMD.

We give one for a class including $\sqrt{\text{TV}}$, $\sqrt{\text{JS}}$, and H:

$$d^2(p, q) = \int_{\mathcal{X}} \kappa(p(x), q(x)) \, \mathrm{d}x$$

(κ needs to be homogeneous and negative-type.)

Name	$\kappa(p(x), q(x))$
JS	$\sum_{r \in \{p,q\}} \frac{1}{2} r(x) \log\left(\frac{2r(x)}{p(x)+q(x)}\right)$
H^2	$\frac{1}{2}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2$
TV	$\lvert p(x) - q(x) \rvert$

Method

$$K(p, q) = \exp\left(-\gamma d(p, q)^2\right)$$
$$\approx \exp\left(-\gamma \left\| A(p) - A(q) \right\|^2\right)$$

A is our embedding of d into $\mathbb{R}^m$:

1. Embed d into $L_2(\mathcal{X})^{2M}$ based on Fuglede (2005)
2. Embed $L_2(\mathcal{X})$ into $\mathbb{R}^n$ with projection coefficients

$$\approx z(A(p))^\mathsf{T} z(A(q))$$

3. Use random Fourier features z for the RBF kernel (Rahimi and Recht, 2007) to get output in $\mathbb{R}^D$

Finite Sample Estimates

- Use kernel density estimation (KDE) to get $\hat{p}$.
- Use simple Monte Carlo for integrals in A.

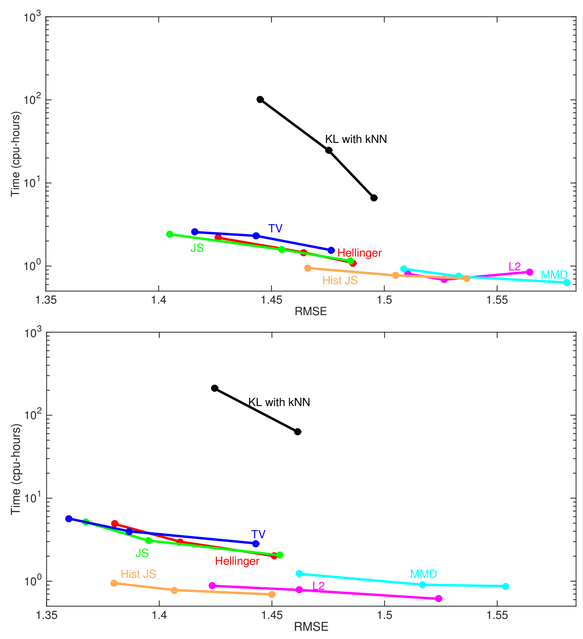
Theory

For fixed p and q, the paper bounds

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

Experiment: Gaussian Mixtures

Mixtures as above with 1 to 10 components.
Train on 4 000, 8 000, and 16 000 distributions.
Test on 2 000.



Details

1.
- First, embed κ into L_2: κ corresponds to a measure $\mu(\lambda)$ such that:

$$\kappa(x, y) = \mathbb{E}_{\lambda \sim \frac{\mu}{Z}} \left|g_\lambda(x) - g_\lambda(y)\right|^2 \approx \frac{1}{M} \sum_{j=1}^{M} \left|g_{\lambda_j}(x) - g_{\lambda_j}(y)\right|,$$

where $Z = \mu(\mathbb{R}_{\geq 0})$, $g_\lambda(x) = \sqrt{Z} \frac{-\frac{1}{2} + \mathrm{i}\lambda}{\frac{1}{2} + \mathrm{i}\lambda} \left(x^{\frac{1}{2} + \mathrm{i}\lambda} - 1\right)$ and $\{\lambda_j\}_{j=1}^M \overset{iid}{\sim} \frac{\mu}{Z}$.

- Then

$$d^2(p, q) = \int_{\mathcal{X}} \kappa(p(x), q(x)) \, \mathrm{d}x$$
$$\approx \frac{1}{M} \sum_{j=1}^{M} \|p^R_{\lambda_j} - q^R_{\lambda_j}\|^2_{L_2} + \|p^I_{\lambda_j} - q^I_{\lambda_j}\|^2_{L_2}$$

where $p^R_\lambda(x) = \Re(g_\lambda(p(x)))$, $p^I_\lambda(x) = \Im(g_\lambda(p(x)))$.

2.
Let $\{\varphi_\alpha(x)\}_{\alpha \in \mathbb{Z}^d}$ be an orthonormal basis for $L_2(\mathcal{X})$.

- For $f \in L_2(\mathcal{X})$, $f(x) = \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f) \varphi_\alpha(x)$ where $a_\alpha(f)$ are projection coefficients $\langle \varphi_\alpha, f \rangle = \int_{\mathcal{X}} \varphi_\alpha(x) f(x) \, \mathrm{d}x$.
- Then, making a smoothness assumption,

$$\langle f, g \rangle = \left\langle \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f) \varphi_\alpha, \sum_{\beta \in \mathbb{Z}^d} a_\beta(g) \varphi_\alpha \right\rangle = \sum_{\alpha \in \mathbb{Z}^d} a_\alpha(f) a_\alpha(g)$$
$$\approx \sum_{\alpha \in V} a_\alpha(f) a_\alpha(g) = \vec{a}(f)^\mathsf{T} \vec{a}(g).$$

- Hence,

$$d^2(p, q) \approx \|A(p) - A(q)\|^2$$
$$A : L_2(\mathcal{X}) \to \mathbb{R}^{2M|V|} \quad p \mapsto A(p) = \frac{1}{\sqrt{M}} \left(\vec{a}(p^R_{\lambda_1}), \vec{a}(p^I_{\lambda_1}), \dots\right).$$

3.
- Sample $\{\omega_i\}_{i=1}^{D/2} \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2} I_m)$. Define

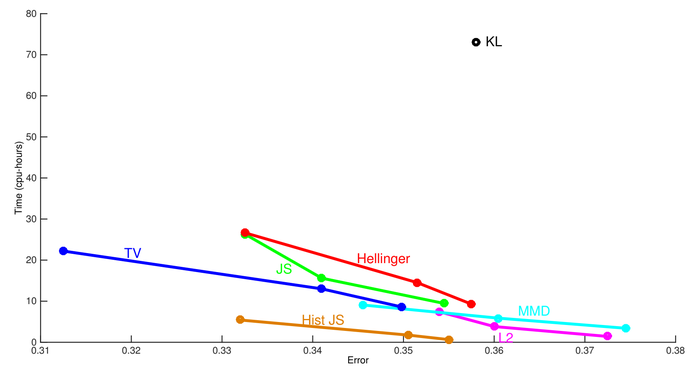
$$z : \mathbb{R}^m \to \mathbb{R}^D \quad x \mapsto z(x) = \sqrt{\frac{2}{D}} \left(\sin(\omega_1^\mathsf{T} x), \cos(\omega_1^\mathsf{T} x), \dots\right).$$

- Then $z(x)^\mathsf{T} z(y) \approx \exp\left(-\frac{1}{2\sigma^2} \|x - y\|^2\right)$, and so

$$K(p, q) \approx z(A(p))^\mathsf{T} z(A(q)).$$

Experiment: Image Classification

CIFAR-10 cats vs dogs, pixel features.
Train on 2 500, 5 000, and 10 000.
Test on 2 000.



Experiment: Scene Classification

SCENE-15, features from `imagenet-vgg-verydeep-16`.
Left (black) use $\hat{A}(\cdot)$; right (blue) use $z(\hat{A}(\cdot))$.