Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Linear-Time Learning on Distributions with Approximate Kernel Embeddings

Danica J. Sutherland[*] and **Junier B. Oliva**[*] and **Barnabás Póczos** and **Jeff Schneider**
Carnegie Mellon University
{dsutherl,joliva,bapoczos,schneide}@cs.cmu.edu

Abstract

Many interesting machine learning problems are best posed by considering instances that are distributions, or sample sets drawn from distributions. Most previous work devoted to machine learning tasks with distributional inputs has done so through pairwise kernel evaluations between pdfs (or sample sets). While such an approach is fine for smaller datasets, the computation of an $N \times N$ Gram matrix is prohibitive in large datasets. Recent scalable estimators that work over pdfs have done so only with kernels that use Euclidean metrics, like the L_2 distance. However, there are a myriad of other useful metrics available, such as total variation, Hellinger distance, and the Jensen-Shannon divergence. This work develops the first random features for pdfs whose dot product approximates kernels using these non-Euclidean metrics. These random features allow estimators to scale to large datasets by working in a primal space, without computing large Gram matrices. We provide an analysis of the approximation error in using our proposed random features, and show empirically the quality of our approximation both in estimating a Gram matrix and in solving learning tasks in real-world and synthetic data.

Introduction

A great deal of effort in machine learning has been devoted to learning functions on vectors of fixed dimension. While complex static vector instances are useful in a myriad of applications, many machine learning problems in datasets with richer, more complex instances are more naturally posed by considering instances that are distributions, or sets drawn from distributions. For instance, political scientists can learn a function from community demographics to vote percentages to understand who supports a candidate (Flaxman, Wang, and Smola 2015). The mass of dark matter halos can be inferred from the velocity of galaxies in a cluster (Ntampaka et al. 2015). Expensive expectation propagation messages can be sped up by learning a "just-in-time" regression model (Jitkrittum et al. 2015). All of these applications are aided by working directly over sets drawn from the distribution of interest, rather than having to develop a per-problem ad-hoc set of summary statistics.

Distributions are inherently infinite-dimensional objects, since in general they require an infinite number of parameters for their exact representation. Hence, it is not immediate how to extend traditional finite vector machine learning techniques to distributional instances. However, recent work has provided various approaches for dealing with distributional data in a nonparametric fashion. For example, regression from distributional covariates to real or distributional responses is possible via kernel smoothing (Póczos et al. 2012a; Oliva, Póczos, and Schneider 2013), and many learning tasks can be solved with RKHS approaches (Muandet et al. 2012; Póczos et al. 2012b). A major shortcoming of both approaches is that they require computing N kernel evaluations per prediction, where N is the number of training instances in a dataset. Often, this implies that one must compute a $N \times N$ Gram matrix of pairwise kernel evaluations. Such approaches fail to scale to datasets where the number of instances N is very large. Another shortcoming of these approaches is that they are often based on Euclidean metrics, either working over a linear kernel, or one based on the L_2 distance over distributions. While such kernels are useful in certain applications, better performance can sometimes be obtained by considering non-Euclidean based kernels. To this end, Póczos et al. (2012b) use a kernel based on Rényi divergences; however, this kernel is not positive semi-definite (PSD), leading to even higher computational cost and other practical issues.

This work addresses these major shortcomings by developing an embedding of random features for distributions. The dot product of the random features for two distributions will approximate kernels based on various distances between densities (see Figure 1). With this technique, we can approximate kernels based on total variation, Hellinger, and Jensen-Shannon divergences, among others. Since one

Figure 1: We approximate kernels between densities p_i, p_j with random features of sample sets $\chi_i \overset{iid}{\sim} p_i, \chi_j \overset{iid}{\sim} p_j$.

[*] These two authors contributed equally.

may work over the primal space induced by the random features there is no need to compute a Gram matrix; thus, one will be able to use these kernels while still scaling to datasets with a large number of instances using primal-space techniques. We provide an approximation bound for the embeddings, and demonstrate the efficacy of the embeddings on both real-world and synthetic data. To the best of our knowledge, this work provides the first non-discretized embedding for non-L_2 kernels for probability density functions.

Related Work

The two main lines of relevant research are the development of kernels on probability distributions and explicit approximate embeddings for scalable kernel learning.

Learning on distributions In computer vision, the popular "bag of words" model (Leung and Malik 2001) represents a distribution by quantizing it onto codewords (usually by k-means on points from all sets), then compares those histograms with some kernel (often exponentiated χ^2).

Another approach estimates a distance between distributions, often the L_2 distance or Kullback-Leibler (KL) divergence, parametrically (Jaakkola and Haussler 1998; Moreno, Ho, and Vasconcelos 2003; Jebara, Kondor, and Howard 2004) or nonparametrically (Sricharan, Wei, and Hero 2013; Krishnamurthy et al. 2014). The distance can then be used in kernel smoothing (Póczos et al. 2012a; Oliva, Póczos, and Schneider 2013) or Mercer kernels (Moreno, Ho, and Vasconcelos 2003; Kondor and Jebara 2003; Jebara, Kondor, and Howard 2004; Póczos et al. 2012b).

These approaches can be powerful, but usually require computing an $N \times N$ matrix of kernel evaluations, which can be infeasible for large datasets. The use of divergences in Mercer kernels faces an additional challenge, which is that the estimated Gram matrix may not be PSD, due to estimation error or because some divergences in fact do not induce a PSD kernel. In general this must be remedied by altering the Gram matrix a "nearby" PSD one. Typical approaches involve eigendecomposing the Gram matrix, which usually costs $O(N^3)$ computation and also presents challenges for traditional inductive learning, where the test points are not known at training time (Chen et al. 2009).

One way to alleviate the scaling problem is the Nyström extension (Williams and Seeger 2001), in which some columns of the Gram matrix are used to estimate the remainder. In practice, one frequently must compute many columns, and methods to make the result PSD are known only for mildly-indefinite kernels (Belongie et al. 2002).

Another approach is to represent a distribution by its mean RKHS embedding under some kernel k. The RKHS inner product is known as the *mean map kernel* (MMK), and the distance the *maximum mean discrepancy* (MMD) (Gretton et al. 2009; Muandet et al. 2012; Szabó et al. 2015). When k is the RBF kernel, the MMK estimate is proportional to an L_2 inner product between Gaussian kernel density estimates.

Approximate embeddings Interest in approximate kernel embeddings was spurred by Rahimi and Recht (2007), whose random kitchen sink (RKS) embedding approximates shift-invariant kernels by sampling their Fourier transform.

A related line of work considers additive kernels of the form $K(x,y) = \sum_{j=1}^{\ell} \kappa(x_j, y_j)$, usually defined on $\mathbb{R}_{\geq 0}^{\ell}$ (e.g. histograms). Maji and Berg (2009) embed the intersection kernel $\sum_{j=1}^{\ell} \min(x_j, y_j)$ via step functions. Vedaldi and Zisserman (2010) allow any homogeneous κ, so that $\kappa(tx, ty) = t\,\kappa(x,y)$, providing embeddings for histogram kernels such as the additive χ^2 kernel and Jensen-Shannon divergence. Their embedding uses the same fundamental result of Fuglede (2005) as ours; we expand to the continuous rather than the discrete case. Vempati et al. (2010) later apply RKS embeddings for generalized RBF kernels (1).

For embedding kernels on spaces other than $\mathbb{R}^{\ell}$, the RKS embedding extends naturally to locally compact abelian groups (Li, Ionescu, and Sminchisescu 2010). Oliva et al. (2014) embedded an L_2 estimate between continuous densities via orthonormal basis functions. MMK also has a simple embedding when the base kernel k is embeddable (Flaxman, Wang, and Smola 2015; Jitkrittum et al. 2015; Lopez-Paz et al. 2015; Sutherland and Schneider 2015).

Embedding Information Theoretic Kernels

For a broad class of distributional distances d, including many common and useful information theoretic divergences, we consider generalized RBF kernels of the form

$$K(p,q) = \exp\left(-\tfrac{1}{2\sigma^2} d^2(p,q)\right), \tag{1}$$

for pdfs p, q: $[0,1]^{\ell} \to \mathbb{R}_{\geq 0}$. We construct features $z(A(\cdot))$ such that $K(p,q) \approx z(A(p))^{\mathsf{T}} z(A(q))$ as follows:

Embedding HDDs into L_2 We define a random function ψ such that $d(p,q) \approx \|\psi(p) - \psi(q)\|$, where $\psi(p)$ is a function from $[0,1]^{\ell}$ to $\mathbb{R}^{2M}$. Thus the metric space of densities with distance d is approximately embedded into the metric space of $2M$-dimensional L_2 functions.

Finite Embeddings of L_2 We use orthonormal basis functions to approximately embed smooth L_2 functions into finite vectors in $\mathbb{R}^{|V|}$. Combined with the previous step, we obtain features $A(p) \in \mathbb{R}^{2M|V|}$ such that d is approximated by Euclidean distances between the $A(\cdot)$ features.

Embedding RBF Kernels into $\mathbb{R}^D$ We use the RKS embedding $z(\cdot)$ so that inner products between $z(A(\cdot))$ features, in $\mathbb{R}^D$, approximate $K(p,q)$.

We can thus use the powerful kernel K without needing to compute an expensive $N \times N$ Gram matrix.

Homogeneous Density Distances (HDDs)

We consider kernels based on metrics which we term homogeneous density distances (HDDs):

$$d^2(p,q) = \int_{[0,1]^{\ell}} \kappa(p(x), q(x)) \,\mathrm{d}x, \tag{2}$$

where $\kappa(x,y) : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is a negative-type kernel, i.e. a squared Hilbertian metric, and $\kappa(tx,ty) = t\kappa(x,y)$ for all $t > 0$. Table 1 shows a few important instances. Note we assume the distributions are supported within $[0,1]^{\ell}$.

Name	$\kappa(p(x), q(x))$	$\mathrm{d}\mu(\lambda)$
JS	$\sum_{r\in\{p,q\}} \frac{1}{2} r(x) \log\left(\frac{2r(x)}{p(x)+q(x)}\right)$	$\frac{\mathrm{d}\lambda}{\cosh(\pi\lambda)(1+\lambda^2)}$
H^2	$\frac{1}{2}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2$	$\frac{1}{2}\,\delta(\lambda=0)\,\mathrm{d}\lambda$
TV	$\lvert p(x) - q(x)\rvert$	$\frac{2}{\pi}\frac{1}{1+4\lambda^2}\,\mathrm{d}\lambda$

Table 1: Squared HDDs. JS is Jensen-Shannon divergence; H is Hellinger distance; TV is total variation distance.

We then use these distances in a generalized RBF kernel (1). d is a Hilbertian metric (Fuglede 2005), so K is positive definite (Haasdonk and Bahlmann 2004). Note we use the $\sqrt{\mathrm{TV}}$ metric, even though TV is itself a metric.

Below we expound on the embeddings used to construct features $z(A(\cdot))$ such that $K(p,q) \approx z(A(p))^\mathsf{T} z(A(q))$.

Embedding HDDs into L_2

Fuglede (2005) shows that κ corresponds to a bounded measure $\mu(\lambda)$, as in Table 1, with

$$\kappa(x,y) = \int_{\mathbb{R}_{\geq 0}} |x^{\frac{1}{2}+\mathbf{i}\lambda} - y^{\frac{1}{2}+\mathbf{i}\lambda}|^2 \,\mathrm{d}\mu(\lambda). \tag{3}$$

Let $Z = \mu(R_{\geq 0})$ and $c_\lambda = (-\frac{1}{2} + \mathbf{i}\lambda)/(\frac{1}{2} + \mathbf{i}\lambda)$; then

$$\kappa(x,y) = \mathbb{E}_{\lambda \sim \frac{\mu}{Z}} |g_\lambda(x) - g_\lambda(y)|^2$$

where $g_\lambda(x) = \sqrt{Z} c_\lambda (x^{\frac{1}{2}+i\lambda} - 1)$.

We can approximate the expectation with an empirical mean. Let $\lambda_j \overset{iid}{\sim} \frac{\mu}{Z}$ for $j \in \{1, \dots, M\}$; then

$$\kappa(x,y) \approx \frac{1}{M} \sum_{j=1}^{M} |g_{\lambda_j}(x) - g_{\lambda_j}(y)|^2.$$

Hence, using $\Re, \Im$ to denote the real and imaginary parts, $d^2(p,q)$ is equal to:

$$\begin{aligned}
&\int_{[0,1]^\ell} \kappa(p(x), q(x))\,\mathrm{d}x \\
&= \int_{[0,1]^\ell} \mathbb{E}_{\lambda \sim \frac{\mu}{Z}} |g_\lambda(p(x)) - g_\lambda(q(x))|^2 \,\mathrm{d}x \\
&\approx \frac{1}{M} \sum_{j=1}^{M} \int_{[0,1]^\ell} \Big(\big(\Re(g_{\lambda_j}(p(x))) - \Re(g_{\lambda_j}(q(x)))\big)^2 \\
&\qquad + \big(\Im(g_{\lambda_j}(p(x))) - \Im(g_{\lambda_j}(q(x)))\big)^2 \Big) \mathrm{d}x \\
&= \|\psi(p) - \psi(q)\|^2,
\end{aligned} \tag{4}$$

where $[\psi(p)](x)$ is given by

$$\frac{1}{\sqrt{M}} \big(p^R_{\lambda_1}(x), \dots, p^R_{\lambda_M}(x), p^I_{\lambda_1}(x), \dots, p^I_{\lambda_M}(x) \big),$$

defining $p^R_{\lambda_j}(x) = \Re(g_{\lambda_j}(p(x)))$, $p^I_{\lambda_j}(x) = \Im(g_{\lambda_j}(p(x)))$. Hence, the HDD between densities p and q is approximately the L_2 distance from $\psi(p)$ to $\psi(q)$, where ψ maps a function $f : [0,1]^\ell \mapsto \mathbb{R}$ to a vector-valued function $\psi(f) : [0,1]^\ell \mapsto \mathbb{R}^{2M}$ of λ functions. M can typically be quite small, since the kernel it approximates is one-dimensional.

Finite Embeddings of L_2

If densities p and q are smooth, then the L_2 metric between the p_λ and q_λ functions may be well approximated using projections to basis functions. Suppose that $\{\varphi_i\}_{i\in\mathbb{Z}}$ is an orthonormal basis for $L_2([0,1])$; then we can construct an orthonormal basis for $L_2([0,1]^\ell)$ by the tensor product:

$$\{\varphi_\alpha\}_{\alpha\in\mathbb{Z}^\ell} \qquad \text{where} \quad \varphi_\alpha(x) = \prod_{i=1}^{\ell} \varphi_{\alpha_i}(x_i),\ x \in [0,1]^\ell,$$

$$\forall f \in L_2([0,1]^\ell),\ f(x) = \sum_{\alpha\in\mathbb{Z}^\ell} a_\alpha(f)\, \varphi_\alpha(x)$$

and $a_\alpha(f) = \langle \varphi_\alpha, f \rangle = \int_{[0,1]^\ell} \varphi_\alpha(t)\, f(t)\,\mathrm{d}t \in \mathbb{R}$. Let $V \subset \mathbb{Z}^\ell$ be an appropriately chosen finite set of indices. If $f, f' \in L_2([0,1]^\ell)$ are smooth and $\vec{a}(f) = (a_{\alpha_1}(f), \dots, a_{\alpha_{|V|}}(f))$, then $\|f - f'\|^2 \approx \|\vec{a}(f) - \vec{a}(f')\|^2$. Thus we can approximate d^2 as the squared distance between finite vectors:

$$\begin{aligned}
&d^2(p,q) \approx \|\psi(p) - \psi(q)\|^2 \\
&\approx \frac{1}{M} \sum_{j=1}^{M} \|\vec{a}(p^R_{\lambda_j}) - \vec{a}(q^R_{\lambda_j})\|^2 + \|\vec{a}(p^I_{\lambda_j}) - \vec{a}(q^I_{\lambda_j})\|^2 \\
&= \|A(p) - A(q)\|^2
\end{aligned} \tag{5}$$

where $A : L_2([0,1]^\ell) \to \mathbb{R}^{2M|V|}$ has $A(p)$ given by

$$\frac{1}{\sqrt{M}} \left(\vec{a}(p^R_{\lambda_1}), \dots, \vec{a}(p^R_{\lambda_M}), \vec{a}(p^I_{\lambda_1}), \dots, \vec{a}(p^I_{\lambda_M}) \right). \tag{6}$$

We will discuss how to estimate $\vec{a}(p^R_\lambda)$, $\vec{a}(p^I_\lambda)$ shortly.

Embedding RBF Kernels into $\mathbb{R}^D$

The A features approximate the HDD (2) in $\mathbb{R}^{2M|V|}$; thus applying the RKS embedding (Rahimi and Recht 2007) to the A features will approximate our generalized RBF kernel (1). The RKS embedding is[1] $z : \mathbb{R}^m \to \mathbb{R}^D$ such that for fixed $\{\omega_i\}_{i=1}^{D/2} \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2} I_m)$ and for each $x, y \in \mathbb{R}^m$:

$$\begin{aligned}
&z(x)^\mathsf{T} z(y) \approx \exp\left(-\tfrac{1}{2\sigma^2}\|x - y\|^2\right), \text{ where} \\
&z(x) = \sqrt{\tfrac{2}{D}} \left(\sin(\omega_1^\mathsf{T} x), \cos(\omega_1^\mathsf{T} x), \dots\right).
\end{aligned} \tag{7}$$

Thus we can approximate the HDD kernel (1) as:

$$\begin{aligned}
K(p,q) &= \exp\left(-\frac{1}{2\sigma^2} d^2(p,q)\right) \\
&\approx \exp\left(-\frac{1}{2\sigma^2}\|A(p) - A(q)\|^2\right) \\
&\approx z(A(p))^\mathsf{T} z(A(q)).
\end{aligned} \tag{8}$$

Finite Sample Estimates

Our final approximation for HDD kernels (8) depends on integrals of densities p and q. In practice, we are unlikely to directly observe an input density, but even given a pdf p, the

[1] There are two versions of the embedding in common use, but this one is preferred (Sutherland and Schneider 2015).

integrals that make up the elements of $A(p)$ are not readily computable. We thus first estimate the density as $\hat{p}$, e.g. with kernel density estimation (KDE), and estimate $A(p)$ as $A(\hat{p})$. Recall that the elements of $A(\hat{p})$ are:

$$a_\alpha(\hat{p}^S_{\lambda_j}) = \int_{[0,1]^\ell} \varphi_\alpha(t)\, \hat{p}^S_{\lambda_j}(t)\, \mathrm{d}t \tag{9}$$

where $j \in \{1, \ldots, M\}, S \in \{R, I\}, \alpha \in V$. In lower dimensions, we can approximate (9) with simple Monte Carlo numerical integration. Choosing $\{u_i\}_{i=1}^{n_e} \overset{iid}{\sim} \mathrm{Unif}([0,1]^\ell)$:

$$\hat{a}_\alpha(\hat{p}^S_{\lambda_j}) = \frac{1}{n_e} \sum_{i=1}^{n_e} \varphi_\alpha(u_i)\, \hat{p}^S_{\lambda_j}(u_i), \tag{10}$$

obtaining $\hat{A}(\hat{p})$. We note that in high dimensions, one may use any high-dimensional density estimation scheme (e.g. Lafferty, Liu, and Wasserman 2012) and estimate (9) with MCMC techniques (e.g. Hoffman and Gelman 2014).

Summary and Complexity

The algorithm for computing features $\{z(A(p_i))\}_{i=1}^N$ for a set of distributions $\{p_i\}_{i=1}^N$, given sample sets $\{\chi_i\}_{i=1}^N$ where $\chi_i = \{X_j^{(i)} \in [0,1]^\ell\}_{j=1}^{n_i} \overset{iid}{\sim} p_i$, is thus:

1. Draw M scalars $\lambda_j \overset{iid}{\sim} \frac{\mu}{Z}$ and $D/2$ vectors $\omega_r \overset{iid}{\sim} \mathcal{N}(0, \sigma^{-2} I_{2M|V|})$, in $O(M\,|V|\,D)$ time.
2. For each of the N input distributions i:
   (a) Compute a kernel density estimate from χ_i, $\hat{p}_i(u_j)$ for each u_j in (10), in $O(n_i n_e)$ time.
   (b) Compute $\hat{A}(\hat{p}_i)$ using a numerical integration estimate as in (10), in $O(M\,|V|\,n_e)$ time.
   (c) Get the RKS features, $z(\hat{A}(\hat{p}_i))$, in $O(M\,|V|\,D)$ time.

Supposing each $n_i \asymp n$, this process takes a total of $O\left(Nnn_e + NM\,|V|\,n_e + NM\,|V|\,D\right)$ time. Taking $|V|$ to be asymptotically $O(n)$, $n_e = O(D)$, and $M = O(1)$ for simplicity, this is $O(NnD)$ time, compared to about $O(N^2 n \log n + N^3)$ for the methods of Póczos et al. (2012b) and $O(N^2 n^2)$ for Muandet et al. (2012).

Theory

We bound $\Pr\left(\left|K(p,q) - z(\hat{A}(\hat{p}))^\mathsf{T} z(\hat{A}(\hat{q}))\right| \geq \varepsilon\right)$ for two fixed densities p and q by considering each source of error: kernel density estimation (ε_{KDE}); approximating $\mu(\lambda)$ with M samples (ε_λ); truncating the tails of the projection coefficients ($\varepsilon_{\text{tail}}$); Monte Carlo integration (ε_{int}); and the RKS embedding (ε_{RKS}).

We need some smoothness assumptions on p and q: that they are members of a periodic Hölder class $\Sigma_{\text{per}}(\beta, L_\beta)$, that they are bounded below by ρ_* and above by ρ^*, and that their kernel density estimates are in $\Sigma_{\text{per}}(\hat{\gamma}, \widehat{L})$ with probability at least $1 - \delta$. We use a suitable form of kernel density estimation, to obtain a uniform error bound with a rate based on the function C^{-1} (Giné and Guillou 2002). We use the Fourier basis and choose $V = \{\alpha \in \mathbb{Z}^\ell \mid \sum_{j=1}^\ell |\alpha_j|^{2s} \leq t\}$ for parameters $0 < s < \hat{\gamma}, t > 0$.

Then, for any $\varepsilon_{\text{RKS}} + \frac{1}{\sigma_k \sqrt{e}} \left(\varepsilon_{\text{KDE}} + \varepsilon_\lambda + \varepsilon_{\text{tail}} + \varepsilon_{\text{int}}\right) \leq \varepsilon$, the probability of the error exceeding ε is at most:

$$\begin{aligned} &2\exp\left(-D\varepsilon_{\text{RKS}}^2\right) + 2\exp\left(-M\varepsilon_\lambda^4/(8Z^2)\right) + \delta \\ &+ 2C^{-1}\left(\frac{\varepsilon_{\text{KDE}}^4 n^{2\beta/(2\beta+\ell)}}{4\log n}\right) + 2M\left(1 - \mu\big([0, u_{\text{tail}})\big)\right) \\ &+ 8M\,|V|\exp\left(-\tfrac{1}{2} n_e \left(\frac{\sqrt{1 + \varepsilon_{\text{int}}^2/(8\,|V|\,Z)} - 1}{\sqrt{\rho^*} + 1}\right)^2\right) \end{aligned}$$

where $u_{\text{tail}} = \sqrt{\max\left(0, \frac{\rho_* t}{8M\ell\widehat{L}^2} \frac{4^{\hat{\gamma}} - 4^s}{4^{\hat{\gamma}}} \varepsilon_{\text{tail}}^2 - \frac{1}{4}\right)}$.

The bound decreases when the function is smoother (larger β, $\hat{\gamma}$; smaller $\widehat{L}$) or lower-dimensional (ℓ), or when we observe more samples (n). Using more projection coefficients (higher t or smaller s, giving higher $|V|$) improves the approximation but makes numerical integration more difficult. Likewise, taking more samples from μ (higher M) improves that approximation, but increases the number of functions to be approximated and numerically integrated.

For the proof and further details, see the appendix.[2]

Numerical Experiments

Throughout these experiments we use $M = 5$, $|V| = 10^\ell$ (selected as rules of thumb; larger values did not improve performance), and use a validation set (10% of the training set) to choose bandwidths for KDE and the RBF kernel as well as model regularization parameters. Except in the scene classification experiments, the histogram methods used 10 bins per dimension; performance with other values was not better. The KL estimator used the fourth nearest neighbor.

We evaluate RBF kernels based on various distances. First, we try our JS, Hellinger, and TV embeddings. We compare to L_2 kernels as in Oliva et al. (2014): $\exp\left(-\frac{1}{2\sigma^2}\|p - q\|_2^2\right) \approx z(\vec{a}(\hat{p}))^\mathsf{T} z(\vec{a}(\hat{q}))$ (L2). We also try the MMD distance (Muandet et al. 2012) with approximate kernel embeddings: $\exp\left(-\frac{1}{2\sigma^2}\widehat{\mathrm{MMD}}(p,q)\right) \approx z\left(\bar{z}(\hat{p})\right)^\mathsf{T} z\left(\bar{z}(\hat{q})\right)$, where $\bar{z}$ is the mean embedding $\bar{z}(\hat{p}) = \frac{1}{n}\sum_{i=1}^n z(X_i)$ (MMD). We further compare to RKS with histogram JS embeddings (Vempati et al. 2010) (Hist JS); we also tried χ^2 embeddings, but their performance was quite similar. We finally try the full Gram matrix approach of Póczos et al. (2012b) with the KL estimator of Wang, Kulkarni, and Verdú (2009) in an RBF kernel (KL), as did Ntampaka et al. (2015).

Gram Matrix Estimation

We first illustrate that our embedding, using the parameter selections as above, can approximate the Jensen-Shanon kernel well. We compare three different approaches to estimating $K(p_i, p_j) = \exp(-\frac{1}{2\sigma^2}\,\mathrm{JS}(p_i, p_j))$. Each approach uses kernel density estimates $\hat{p}_i$. The estimates are compared on a dataset of $N = 50$ random GMM distributions $\{p_i\}_{i=1}^N$ and samples of size $n = 2\,500$: $\chi_i = \{X_j^{(i)} \in [0,1]^2\}_{j=1}^n \overset{iid}{\sim} p_i$. See the appendix for more details.

[2] cs.cmu.edu/~joliva/papers/joliva_aaai16_supp.pdf

The first approach approximates JS based on empirical estimates of entropies $\mathbb{E}\log\hat{p}_i$. The second approach estimates JS as the Euclidean distance of vectors of projection coefficients (5) : $\mathrm{JS}_{\mathrm{pc}}(p_i,p_j) = \|\hat{A}(\hat{p}_i) - \hat{A}(\hat{p}_j)\|^2$. For these first two approaches we compute the pairwise kernel evaluations in the Gram matix as $G_{ij}^{\mathrm{ent}} = \exp(-\frac{1}{2\sigma^2}\,\mathrm{JS}_{\mathrm{ent}}(p_i,p_j))$, and $G_{ij}^{\mathrm{pc}} = \exp(-\frac{1}{2\sigma^2}\,\mathrm{JS}_{\mathrm{pc}}(p_i,p_j))$ using their respective approximations for JS. Lastly, we directly estimate the JS kernel with dot products of our random features (8): $G_{ij}^{\mathrm{rks}} = z(\hat{A}(\hat{p}_i))^\mathsf{T} z(\hat{A}(\hat{p}_j))$, with $D = 7\,000$.

Figure 2 shows the N^2 true pairwise kernel values versus the aforementioned estimates. Quantitatively, the entropy method obtained a squared correlation to the true kernel value of $R^2_{\mathrm{ent}} = 0.981$; using the A features with an exact kernel yielded $R^2_{\mathrm{pc}} = 0.974$; adding RKS embeddings gave $R^2_{\mathrm{rks}} = 0.966$. Thus our method's estimates are nearly as good as direct estimation via entropies, while allowing us to work in primal space and avoid $N \times N$ Gram matrices.

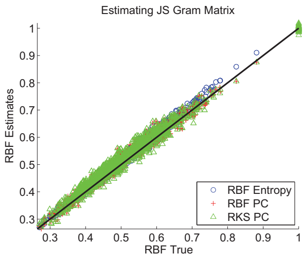


Figure 2: Estimating RBF with JS divergence.

Estimating the Number of Mixture Components

We will now illustrate the efficacy of HDD random features in a regression task; following Oliva et al. (2014), we estimate the number of components from a mixture of truncated Gaussians. We generate the distributions as follows: Draw the number of components Y_i for the ith distribution as $Y_i \sim \mathrm{Unif}\{1,\ldots,10\}$. For each component select a mean $\mu_k^{(i)} \sim \mathrm{Unif}[-5,5]^2$ and covariance $\Sigma_k^{(i)} = a_k^{(i)} A_k^{(i)} A_k^{(i)\mathsf{T}} + B_k^{(i)}$, where $a \sim \mathrm{Unif}[1,4]$, $A_k^{(i)}(u,v) \sim \mathrm{Unif}[-1,1]$, and $B_k^{(i)}$ is a diagonal 2×2 matrix with $B_k^{(i)}(u,u) \sim \mathrm{Unif}[0,1]$. Then weight each component equally in the mixture. Given a sample χ_i, we predict the number of components Y_i. An example distribution and sample are shown in Figure 3; predicting the number of components is difficult even for humans.

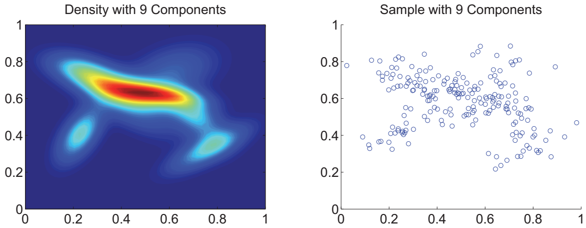


Figure 3: A GMM and 200 points drawn from it.

Figure 4 presents results for predicting with ridge regression the number of mixture components Y_i, given a varying number of sample sets χ_i, with $|\chi_i| \in \{200, 800\}$; we use $D = 5\,000$. The HDD-based kernels achieve substantially lower error than the L_2 and MMD kernels. They also outperform the histogram kernel, especially with $|\chi_i| = 200$, and the KL kernel. Note that fitting mixtures with EM and selecting a number of components using AIC (Akiake 1973) or BIC (Schwarz 1978) performed much worse than regression; only AIC with $|\chi_i| = 800$ outperformed the best constant predictor of 5.5. Linear versions of the L_2 and MMD kernels were also no better than the constant predictor.

The HDD embeddings were more computationally expensive than the other embeddings, but much less expensive than the KL kernel, which grows at least quadratically in the number of distributions. Note that the histogram embeddings used an optimized C implementation (Vedaldi and Fulkerson 2008), as did the KL kernel[3], while the HDD embeddings used a simple Matlab implementation.

Image Classification

As another example of the performance of our embeddings, we now attempt to classify images based on their distributions of pixel values. We took the "cat" and "dog" classes from the CIFAR-10 dataset (Krizhevsky and Hinton 2009), and represented each 32×32 image by a set of triples (x, y, v), where x and y are the position of each pixel in the image and v the pixel value after converting to grayscale. The horizontal reflection of the image was also included, so each sample set $\chi_i \subset \mathbb{R}^3$ had $|\chi_i| = 2\,048$. This is certainly not the best representation for these images; rather, we wish to show that given this simple representation, our HDD kernels perform well relative to the other options.

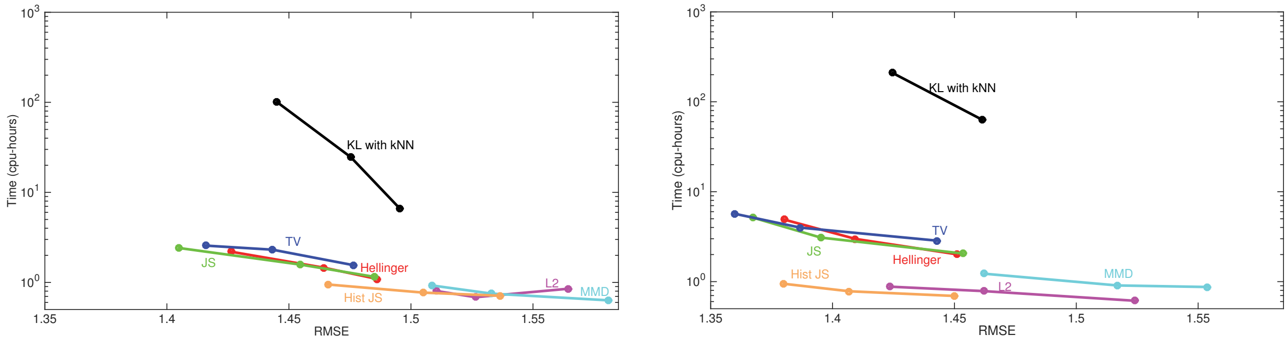
We used the same kernels as above in an SVM classifier from LIBLINEAR (Fan et al. 2008, for the embeddings) or LIBSVM (Chang and Lin 2011, for the KL kernel), with $D = 7\,000$. Figure 5 shows computation time and accuracy on the standard test set (of size 2K) with 2.5K, 5K, and 10K training images. Our JS and Hellinger embedding approximately match the histogram JS embedding in accuracy here, while our TV embedding beats histogram JS; all outperform L_2 and MMD. We could only run the KL kernel for the 2.5K training set size; its accuracy was comparable to the HDD and histogram embeddings, at far higher computational cost.

Scene Classification

Modern computer vision classification systems typically consist of a deep network with several convolutional and pooling layers to extract complex features of input images, followed by one or two fully-connected classification layers. The activations are of shape $n \times h \times w$, where n is the number of filters; each unit corresponds to an overlapping patch of the original image. We can thus treat the final pooled activations as a sample of size hw from an n-dimensional distribution, similarly to how Póczos et al. (2012b) and Muandet et al. (2012) used SIFT features from image patches. Wu, Gao, and Liu (2015) set accuracy records on several scene classification datasets with a particular ad-hoc method of extracting features from distributions (D3); we compare to our more principled alternatives.

We consider the Scene-15 dataset (Lazebnik, Schmid, and Ponce 2006), which contains $4\,485$ natural images in 15

[3] github.com/dougalsutherland/skl-groups/



(a) Samples of size 200.

(b) Samples of size 800.

Figure 4: Error and computation time for estimating the number of mixture components. The three points on each line correspond to training set sizes of 4K, 8K, and 16K; error is on the fixed test set of size 2K. Note the logarithmic scale on the time axis. The KL kernel for $|\chi_i| = 800$ with 16K training sets was too slow to run. AIC-based predictions achieved RMSEs of 2.7 (for 200 samples) and 2.3 (for 800); BIC errors were 3.8 and 2.7; a constant predictor of 5.5 had RMSE of 2.8.

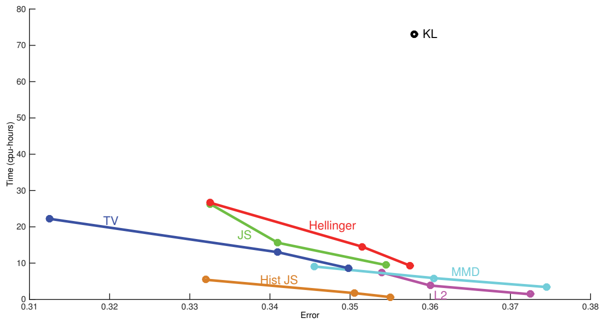


Figure 5: Misclassification rate and computation time for classifying CIFAR-10 cats versus dogs. The three points on each line show training set sizes of 2.5K, 5K, and 10K; error is on the fixed test set of size 2K. Note the linear scale for time. The KL kernel was too slow to run for 5K or 10K training points.

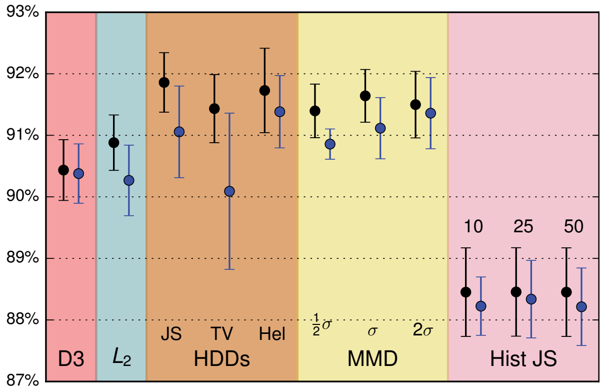


Figure 6: Mean and standard deviation of accuracies on the Scene-15 dataset in 10 random splits. The left, black lines use $\hat{A}(\cdot)$ features; the right, blue lines show $z(\hat{A}(\cdot))$ features. MMD methods vary bandwidth are relative to σ, the median of pairwise distances; histogram methods vary the number of bins.

location categories, and follow Wu, Gao, and Liu in extracting features from the last convolutional layer of the `imagenet-vgg-verydeep-16` model (Simonyan and Zisserman 2015). We replace that layer's rectified linear activations with sigmoid squashing to $[0, 1]$.[4] hw ranges from 400 to 1 000. There are 512 filter dimensions; we concatenate features $\hat{A}(\hat{p}_i)$ extracted from each independently.

We train on the standard for this dataset of 100 images from each class (1500 total) and test on the remainder; Figure 6 shows results. We did not include spatial information; still, we match the best prior published performance of 91.59 ± 0.48, trained on a large scene classification dataset (Zhou et al. 2014). Adding spatial information brought the D3 method to about 92% accuracy; their best hybrid method obtained 92.9%. With these features, however, our methods match or beat MMD and substantially outperform D3, L_2, and the histogram embeddings.

Discussion

This work presents the first nonlinear embedding of density functions for quickly computing HDD-based kernels, including kernels based on the popular total variation, Hellinger and Jensen-Shanon divergences. Nonparametric uses of kernels with these divergences previously necessitated the computation of a large $N \times N$ Gram matrix, prohibiting their use in large datasets. Our embeddings allow one to work in a primal space while using information theoretic kernels. We analyze the approximation error of our embeddings, and show their quality on several synthetic and real-world datasets.

Acknowledgements

This work was funded in part by NSF grant IIS1247658 and by DARPA grant FA87501220324. DJS is also supported by a Sandia Campus Executive Program fellowship.

[4] We used piecewise-linear weights before the sigmoid function such that 0 maps to 0.5, the 90th percentile of the positive observations maps to 0.9, and the 10th percentile of the negative observations to 0.1, for each filter.

References

Akiake, H. 1973. Information theory and an extension of the maximum likelihood principle. In *2nd Int. Symp. on Inf. Theory*.

Belongie, S.; Fowlkes, C.; Chung, F.; and Malik, J. 2002. Spectral partitioning with indefinite kernels using the Nyström extension. In *ECCV*.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2(3):1–27.

Chen, Y.; Garcia, E. K.; Gupta, M. R.; Rahimi, A.; and Cazzanti, L. 2009. Similarity-based classification: Concepts and algorithms. *JMLR* 10:747–776.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A library for large linear classification. *JMLR* 9:1871–1874.

Flaxman, S. R.; Wang, Y.-x.; and Smola, A. J. 2015. Who supported Obama in 2012? Ecological inference through distribution regression. In *KDD*, 289–298.

Fuglede, B. 2005. Spirals in Hilbert space: With an application in information theory. *Exposition. Math.* 23(1):23–45.

Giné, E., and Guillou, A. 2002. Rates of strong uniform consistency for multivariate kernel density estimators. *Ann. Inst. H. Poincaré Probab. Statist.* 38(6):907–921.

Gretton, A.; Fukumizu, K.; Harchaoui, Z.; and Sriperumbudur, B. K. 2009. A fast, consistent kernel two-sample test. In *NIPS*.

Haasdonk, B., and Bahlmann, C. 2004. Learning with distance substitution kernels. In *Pattern Recognition: 26th DAGM Symposium*, 220–227.

Hoffman, M. D., and Gelman, A. 2014. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *JMLR* 15(1):1593–1623.

Jaakkola, T., and Haussler, D. 1998. Exploiting generative models in discriminative classifiers. In *NIPS*.

Jebara, T.; Kondor, R.; and Howard, A. 2004. Probability product kernels. *JMLR* 5:819–844.

Jitkrittum, W.; Gretton, A.; Heess, N.; Eslami, S.; Lakshminarayanan, B.; Sejdinovic, D.; and Szabó, Z. 2015. Kernel-based just-in-time learning for passing expectation propagation messages. *UAI*.

Kondor, R., and Jebara, T. 2003. A kernel between sets of vectors. In *ICML*.

Krishnamurthy, A.; Kandasamy, K.; Poczos, B.; and Wasserman, L. 2014. Nonparametric estimation of Rényi divergence and friends. In *ICML*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *University of Toronto, Tech. Rep.*

Lafferty, J.; Liu, H.; and Wasserman, L. 2012. Sparse nonparametric graphical models. *Statistical Science* 27(4):519–537.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*.

Leung, T., and Malik, J. 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV* 43.

Li, F.; Ionescu, C.; and Sminchisescu, C. 2010. Random Fourier approximations for skewed multiplicative histogram kernels. In *Pattern Recognition: DAGM*, 262–271.

Lopez-Paz, D.; Muandet, K.; Schölkopf, B.; and Tolstikhin, I. 2015. Towards a learning theory of causation. *ICML*.

Maji, S., and Berg, A. C. 2009. Max-margin additive classifiers for detection. In *ICCV*.

Moreno, P. J.; Ho, P. P.; and Vasconcelos, N. 2003. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *NIPS*.

Muandet, K.; Fukumizu, K.; Dinuzzo, F.; and Schölkopf, B. 2012. Learning from distributions via support measure machines. In *NIPS*.

Ntampaka, M.; Trac, H.; Sutherland, D. J.; Battaglia, N.; Póczos, B.; and Schneider, J. 2015. A machine learning approach for dynamical mass measurements of galaxy clusters. *The Astrophysical Journal* 803(2):50.

Oliva, J. B.; Neiswanger, W.; Póczos, B.; Schneider, J.; and Xing, E. 2014. Fast distribution to real regression. In *AISTATS*.

Oliva, J. B.; Póczos, B.; and Schneider, J. 2013. Distribution to distribution regression. In *ICML*.

Póczos, B.; Rinaldo, A.; Singh, A.; and Wasserman, L. 2012a. Distribution-free distribution regression. *AISTATS*.

Póczos, B.; Xiong, L.; Sutherland, D. J.; and Schneider, J. 2012b. Nonparametric kernel estimators for image classification. In *CVPR*.

Rahimi, A., and Recht, B. 2007. Random features for large-scale kernel machines. In *NIPS*.

Schwarz, G. 1978. Estimating the dimension of a model. *Ann. Statist.* 6(2):461–464.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Sricharan, K.; Wei, D.; and Hero, III, A. O. 2013. Ensemble estimators for multivariate entropy estimation. *IEEE Trans. Inf. Theory* 59:4374–4388.

Sutherland, D. J., and Schneider, J. 2015. On the error of random Fourier features. In *UAI*.

Szabó, Z.; Gretton, A.; Póczos, B.; and Sriperumbudur, B. 2015. Two-stage sampled learning theory on distributions. *AISTATS*.

Vedaldi, A., and Fulkerson, B. 2008. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/.

Vedaldi, A., and Zisserman, A. 2010. Efficient additive kernels via explicit feature maps. In *CVPR*.

Vempati, S.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. V. 2010. Generalized RBF feature maps for efficient detection. In *British Machine Vision Conference*.

Wang, Q.; Kulkarni, S. R.; and Verdú, S. 2009. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Trans. Inf. Theory* 55(5):2392–2405.

Williams, C. K. I., and Seeger, M. 2001. Using the Nyström method to speed up kernel machines. In *NIPS*.

Wu, J.; Gao, B.-B.; and Liu, G. 2015. Visual recognition using directional distribution distance.

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using Places database. In *NIPS*.