Support Distribution Machines

Barnabás Póczos **Liang Xiong** **Dougal Sutherland** **Jeff Schneider**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA
USA, 15213

Abstract

Most machine learning algorithms, such as classification or regression, treat the individual data point as the object of interest. Here we consider extending machine learning algorithms to operate on *groups* of data points. We suggest treating a group of data points as a set of i.i.d. samples from an underlying feature distribution for the group. Our approach is to generalize kernel machines from vectorial inputs to i.i.d. sample sets of vectors. For this purpose, we use a nonparametric estimator that can consistently estimate the inner product and certain kernel functions of two distributions. The projection of the estimated Gram matrix to the cone of semi-definite matrices enables us to employ the kernel trick, and hence use kernel machines for classification, regression, anomaly detection, and low-dimensional embedding in the space of distributions. We present several numerical experiments both on real and simulated datasets to demonstrate the advantages of our new approach.

1 Introduction

Functional data analysis is a new and emerging field of statistics and machine learning. It extends the classical multivariate methods to the case when the data points are functions (Ramsay and Silverman, 2005). In many areas, including meteorology, economics, and bioinformatics (Muller et al., 2008), it is more natural to assume that the data consist of functions rather than finite-dimensional vectors. This setting is especially natural when we study time series and we wish to predict the evolution of a quantity using some other quantities measured along time (Kadri et al., 2010). Although this problem is important in many applications, the field is quite new and immature; we know very little about efficient algorithms.

In this paper we consider a version of the problem where the input functions are density functions, and we cannot even observe these functions directly. We only see some finite i.i.d. samples from them. If we can do machine learning in this scenario, then we have a way to do machine learning on groups of data points. We treat each data point in the group as a sample from the underlying feature distribution of the group. Using this setting, our goal is to generalize kernel machines from finite-dimensional vector spaces to the domain of finite sample sets, where each sample set represents a distribution.

We develop methods for *classification and regression of distributions*. In the classification problem our goal is to find a map from the space of distributions to the space of discrete objects, while in the regression problem (with scalar response) the goal is to find a map to the space of real numbers. For this purpose we extend the support vector machine algorithm (SVM) to the space of distributions. In our framework, some of the distributions in the training data will play the role of support vectors, hence we call this method a *support distribution machine* (SDM).

We also show how kernel machines on the space of distributions can be used to find *anomalous distributions*. It might happen that each measurement in a sample set looks normal, but the distribution of these values is different from those of other groups. Our goal is to detect these anomalous sample sets/distributions. The standard anomaly/novelty detection approach only focuses on finding individual points (Chandola et al., 2009). Our "group anomaly" detection task, however, is different; we want to find anomalous groups of points (i.e. anomalous distributions) in which each individual point can be normal.

Finally, we develop an algorithm for *linear distribution regression*. As opposed to regression where the

response is scalar, here we deal with a regression problem that has a distribution response: we are looking for a linear map from the space of distributions to the space of distributions. As an application we show how this method can be used to generalize *local linear embedding* (LLE) (Roweis and Saul, 2000) to the space of distributions. Here our goal is to embed the distributions (the sample sets of the training data) into a low-dimensional space preserving proximity, i.e. nearby distributions should be mapped into nearby points in a low-dimensional Euclidean space.

The paper is organized as follows. In the next section we review some related work. We formally introduce our problems and show how to define kernels on distributions in Section 3. Section 4 explains how to evaluate kernels in the space of distributions when the densities are unknown, and only a few i.i.d. samples are available to us from each distribution. Section 5 presents the results of numerical experiments that demonstrate the effectiveness of our proposed algorithm in several applications, including classification, regression, anomaly detection, and low-dimensional embedding. We show results on both simulated and real datasets. Finally, we conclude with a discussion of our work.

2 Related Work

Although the field of functional data analysis is improving quickly, there is still only very limited work available on this field. The traditional approach in functional regression represents the functions by an expansion in some basis, e.g. B-spline or Fourier basis, and the main emphasis is on the inference of the coefficients (Ramsay and Silverman, 2005). A more recent approach uses reproducing kernel Hilbert spaces (RKHS) (Berlinet and Thomas, 2004). For scalar responses, the first steps were made by Preda (2007). Inspired by this, Lian (2007) and Kadri et al. (2010) generalized the functional regression problem to functional responses as well. To predict infinite-dimensional function valued responses from functional attributes, they extended the concept of vector valued kernels in multi-task learning (Micchelli and Pontil, 2005) to operator valued kernels. Although these methods have been developed for functional regression, they can be used for classifications of functions as well (Kadri et al., 2011). We note that our studied problem is more difficult in the sense that we cannot even observe directly the inputs (densities of the distributions); only a few i.i.d. sample points are available to us. Luckily in several kernel functions we do not need to know these densities; their inner product is sufficient, and we will show in Section 4 how this can be estimated efficiently.

The first paper that used kernels on probabilities densities was the work of Jebara et al. (2004). Here the authors fit a parametric family (e.g. exponential family) to the densities, and using these fitted parameters they estimate the inner products between the distributions. However, in practice it is rare to know that the true densities belong to these parametric families. When this assumption does not hold, this method introduces some unavoidable bias in estimating the inner products between the densities. In contrast, our method is completely nonparametric and provides provably consistent kernel estimations for certain kernels. Furthermore, we avoid estimating the densities, which are nuisance parameters in our problem.

Póczos et al. (2011) used similar nonparametric estimators to solve certain machine learning problems in the space of distributions. However, that paper did not investigate inner product estimation or the relation to kernel machines. It studies only simple kNN based classifiers that apply divergence estimators. As such, the estimators cannot be used in Hilbert spaces, and hence cannot perform distribution regression or LLE of distributions.

SVMs for structured, complex output also have been investigated (Tsochantaridis et al., 2004). In our work, however, the input is also complex: sets of i.i.d. samples from distributions.

3 Formal Problem Setting

Here we formally define our problems and show how kernel methods can be generalized to distributions and sample sets. We investigate both supervised and unsupervised problems. We assume we have N inputs $X_1,\dots,X_N$, where the nth input $X_n = \{X_{n,1},\dots,X_{n,m_n}\}$ consists of m_n i.i.d. samples from density p_n, i.e. X_n is a set of sample points, and $X_{n,j} \sim p_n$, $j = 1,\dots m_n$. Let $\mathcal{X}$ denote the set of these sample sets, so that $X_n \in \mathcal{X}$.

3.1 Applications

Distribution classification In this supervised learning problem we have $\{(X_n, Y_n)\}_{n=1}^N$ (input, output) pairs. The output domain is discrete, i.e. $Y_n \in \mathcal{Y} \doteq \{y_1,\dots,y_d\}$. We are looking for a function $f : \mathcal{X} \to \mathcal{Y}$, such that for a new input and output pair $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ when the classification is perfect we have that $f(X) = Y$. We will use ideas from support vector machines to perform this classification. For simplicity, we only discuss the case when the class number $d = 2$. The ideas below can be extended to multiclass classification in the standard ways, e.g. (i) using one vs all classifiers, or (ii) training $d(d-1)/2$

classifiers between each class pairs and then applying all of them for test points. Finally we decide the class labels by voting.

Let $\mathcal{P}$ denote the set of density functions, $\mathcal{K}$ be a Hilbert-space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$, and $\phi : \mathcal{P} \to \mathcal{K}$ denote an operator that maps the density functions to the feature space $\mathcal{K}$. The dual form of the "soft margin SVM" is as follows (Schölkopf and Smola, 2002):

$$\hat{\alpha} = \arg\max_{\alpha \in \mathbb{R}^N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j G_{ij}, \qquad (1)$$

subject to $\sum_i \alpha_i y_i = 0$, $0 \le \alpha_i \le C$, where $C > 0$ is a parameter, $y_i \in \{-1, 1\}$ are the class labels, and $G \in \mathbb{R}^{N \times N}$ is the Gram matrix: $G_{ij} \doteq \langle \phi(p_i), \phi(p_j) \rangle_{\mathcal{K}} = K(p_i, p_j)$. Now, the predicted class label of a test density p is simply $f(p) = \text{sign}(\sum_{i=1}^{N} \hat{\alpha}_i y_i K(p_i, p) + b)$, where the bias term b can be obtained by averaging $b = y_j - \sum_i y_i \alpha_i G_{ij}$ over all points with $\alpha_j > 0$.

There are many tools available to solve the quadratic programming task in (1). All that remains is to estimate $\{K(p_i, p)\}_i$ and $\{K(p_i, p_j)\}_{i,j}$ based on the few i.i.d. samples available to us. We propose estimators for these kernels in Section 4.

Anomaly detection We can also use these ideas in a one-class SVM (Schölkopf et al., 2001) to find anomalous distributions.

Distribution regression with scalar response (DRSR) Learning real valued functionals of distributions (e.g. entropy, mutual information) is of great importance in statistics and machine learning. For certain functionals, however, this is very difficult, especially if our only information about the densities is a few i.i.d. samples. Assume that we can ask an expert or a reliable Monte Carlo method to tell us the values of the functionals for a few of those sample sets. Then we get new sample sets for which expert advice is unavailable, and we do not have the time to run Monte Carlo methods. Our goal is to estimate the functional values for these sample sets as well.

This is a regression task on sample sets with scalar response. Using the Gram matrix defined above in the support vector regression equations (Schölkopf and Smola, 2002), we can estimate the unknown function $f : \mathcal{X} \to \mathbb{R}$.

Distribution regression with distribution response (DRDR) A more general distribution regression problem arises when $\mathcal{Y} = \mathcal{X}$, that is the outputs are also distributions (or to be more precise, i.i.d. samples from distributions), and we are looking for an $f : \mathcal{X} \to \mathcal{X}$ regression function. Below we show how the coefficients of the linear regression can be calculated after transforming the distributions to the Hilbert space $\mathcal{K}$. The regression problem is given by the following quadratic problem:

$$\begin{aligned}\hat{\alpha} &= \arg\min_{\alpha \in \mathbb{R}^N} \|\phi(p) - \sum_{i=1}^{N} \alpha_i \phi(p_i)\|_{\mathcal{K}}^2 \\ &= K(p,p) - 2\sum_{i=1}^{N} \alpha_i K(p_i, p) + \alpha_i \alpha_j \sum_{i=1}^{N}\sum_{j=1}^{N} K(p_i, p_j).\end{aligned}$$

Local linear embedding of distributions (LLE) LLE (Roweis and Saul, 2000) performs nonlinear dimensionality reduction by computing a low-dimensional, neighborhood preserving embedding of high- (but finite-) dimensional data. As an application of DRDR, we show how to generalize LLE to the infinite-dimensional space of distributions and to i.i.d. sample sets of distributions. Our goal is to find a map $f : \mathcal{X} \to \mathbb{R}^d$ that preserves the local geometry of the distributions. To characterize the local geometry, we reconstruct each distribution from its κ neighbor distributions by DRDR.

As above, let $X_1, \ldots, X_N$ be our training set. The squared Euclidean distance between $\phi(p_i)$ and $\phi(p_j)$ is given by $\langle \phi(p_i) - \phi(p_j), \phi(p_i) - \phi(p_j) \rangle_{\mathcal{K}} = K(p_i, p_i) + K(p_j, p_j) - 2K(p_i, p_j)$. Let $\mathcal{N}_i$ denote the set of κ nearest neighbors of distribution p_i among $\{p_j\}_{j \neq i}$. The intrinsic local geometric properties of distributions $\{p_i\}_{i=1}^N$ are characterized by the reconstruction weights $\{W_{i,j}\}_{i,j=1}^N$ in the equation below.

$$\begin{aligned}\widehat{W} = \arg\min_{W \in \mathbb{R}^{N \times N}} &\sum_{i=1}^{N} \|\phi(p_i) - \sum_{j \in \mathcal{N}_i} W_{i,j} \phi(p_j)\|_{\mathcal{K}}^2 \qquad (2) \\ \text{s. t. } &\sum_{j \in \mathcal{N}_i} W_{i,j} = 1, \quad \text{and} \quad W_{i,j} = 0 \quad \text{if } j \notin \mathcal{N}_i.\end{aligned}$$

Note that the cost function in (2) can be rewritten:

$$\sum_{i=1}^{N} \left(G_{ii} - 2 \sum_{j \in \mathcal{N}_i} W_{i,j} G_{ij} + \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_i} W_{i,j} W_{k,j} G_{jk} \right).$$

Having calculated the weights $\{\widehat{W}_{i,j}\}$, we compute $Y_i = f(p_i) \in \mathbb{R}^d$ (i.e. the embedded distributions) as the d-dimensional vectors best reconstructed locally by these weights:

$$\widehat{Y} = \arg\min_{\{Y_i \in \mathbb{R}^d\}_{i=1}^N} \sum_{i=1}^{N} \|Y_i - \sum_{j \in \mathcal{N}_i} \widehat{W}_{i,j} Y_j\|^2$$

Finally, $\widehat{Y} = \{\widehat{Y}_1, \ldots, \widehat{Y}_N\}$, and $\widehat{Y}_i$ corresponds to the d-dimensional image of distribution p_i.

Note that many other nonlinear dimensionality reduction algorithms—including stochastic neighbor embedding (Hinton and Roweis, 2002), multidimensional

scaling (MDS) (Borg and Groenen, 2005), isomap (Tenenbaum et al., 2000), curvilinear component analysis (Sun et al., 2010)—use only Euclidean distances between the input points to perform dimensionality reduction. All of these methods can be generalized to distributions by simply replacing the finite-dimensional Euclidean metric with $\langle p-q, p-q\rangle_{\mathcal{K}}$.

3.2 Constructing Kernels

To use the methods above, we must estimate $K(p,q)$, a kernel value between distributions p and q using two finite i.i.d. sample sets from them. Many kernels, i.e. positive semi-definite functionals of p and q can be constructed from

$$D_{\alpha,\beta}(p\|q) = \int p^{\alpha}(x)\, q^{\beta}(x)\, p(x)\,\mathrm{d}x, \tag{3}$$

where $\alpha, \beta \in \mathbb{R}$. Linear ($K(p,q) = \int pq$), polynomial ($K(p,q) = (\int pq + c)^s$), and Gaussian kernels ($K(p,q) = \exp(-\frac{1}{2}\mu^2(p,q)/\sigma^2)$, where $\mu(p,q) = \int p^2 + q^2 - 2pq$) are examples of these. In the Gaussian kernel, one can also try to use other "distances", e.g. the Hellinger distance where $\mu^2(p,q) = 1 - \int p^{1/2}q^{1/2}$, the Rényi-$\alpha$ divergence where $\mu(p,q) = \frac{1}{\alpha-1}\log\int p^{\alpha}q^{1-\alpha}$, or the KL-divergence, which is its $\alpha \to 1$ limit case. These latter divergences are not symmetric, do not satisfy the triangle inequality, and do not lead to positive semi-definite Gram matrices. In Section 4.3 we will show how to fix this problem.

4 Nonparametric Kernel Estimation

To estimate the values of the kernels in Section 3.2, it is enough to estimate $D_{\alpha,\beta}(p\|q)$ terms for some α, β values. Using the tools that have been applied for Rényi entropy (Leonenko et al., 2008), Shannon entropy (Goria et al., 2005), KL divergence (Wang et al., 2009), and Rényi divergence estimation (Póczos and Schneider, 2011), we show how to estimate $D_{\alpha,\beta}(p\|q)$ in an efficient, nonparametric, and consistent way.

4.1 k-NN Based Density Estimators

k-NN density estimators operate using distances between the observations in a given sample and their kth nearest neighbors. Let $X_{1:n} \doteq (X_1, \ldots, X_n)$ be an i.i.d. sample from a distribution with density p, and similarly let $Y_{1:m} \doteq (Y_1, \ldots, Y_m)$ be an i.i.d. sample from a distribution having density q. Let $\rho_k(i)$ denote the Euclidean distance of the kth nearest neighbor of X_i in the sample $X_{1:n}$, and similarly let $\nu_k(i)$ denote the distance of the kth nearest neighbor of X_i in the sample $Y_{1:m}$. Let $\bar{c}$ denote the volume of a d-dimensional unit ball. Loftsgaarden and Quesenberry (1965) define the k-NN based density estimators of p and q at X_i as $\hat{p}_k(X_i) = k/((n-1)\bar{c}\rho_k^d(i))$, $\hat{q}_k(X_i) = k/(m\bar{c}\nu_k^d(i))$. Note that these estimators are consistent only when $k(n) \to \infty$. We will use these density estimators in our proposed divergence estimators; however, we will keep k fixed and will still be able to prove their consistency.

4.2 Kernel Estimation

In this section we introduce our estimator for $D_{\alpha,\beta}(p\|q)$. If we simply plugged $\hat{p}_k(X_i)$ and $\hat{q}_k(X_i)$ into (3), then we could estimate $D_{\alpha,\beta}(p\|q)$ with

$$\frac{1}{n}\sum_{i=1}^{n}\frac{k^{\alpha+\beta}}{\bar{c}^{\alpha+\beta}}(n-1)^{-\alpha}m^{-\beta}\rho_k^{-d\alpha}(i)\nu_k^{-d\beta}(i).$$

Nonetheless, this estimator is asymptotically biased for any fixed k. Using the same tools as in Póczos et al. (2011), one can prove that by introducing a multiplicative term the following estimator is L_2 consistent under certain conditions (see the Appendix for details):

$$\widehat{D}_{\alpha,\beta} = \frac{1}{n}\sum_{i=1}^{n}(n-1)^{-\alpha}m^{-\beta}\rho_k^{-d\alpha}(i)\nu_k^{-d\beta}(i)B_{k,\alpha,\beta}, \tag{4}$$

where $B_{k,\alpha,\beta} \doteq \bar{c}^{-\alpha-\beta}\frac{\Gamma(k)^2}{\Gamma(k-\alpha)\Gamma(k-\beta)}$. Notably, this multiplicative bias does not depend on p or q.

4.3 Projecting to the Cone of Positive Semi-definite Matrices

Under certain conditions $\widehat{D}_{\alpha,\beta}$ is a consistent estimator of $D_{\alpha,\beta}$, and thus by plugging these estimators into the formulae in Section 3.2 we get consistent estimators for those kernels. It means that the more sample points we have the better the quality of the kernel estimation is. However, this does not guarantee that the estimated Gram matrix is positive semi-definite. Therefore, we project this estimated Gram matrix back to the cone of positive semi-definite matrices using the alternating projection method presented in Higham (2002).

5 Numerical Experiments

5.1 SDM for Classification

In this section we demonstrate how SDMs can be used for image classification on a noisy version of the USPS dataset, on natural images, and on turbulence data.

5.1.1 USPS Dataset

The USPS dataset[1] (Hull, 1994) consists of 10 classes, and each data point is a 16×16 grayscale image. On

[1] `http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/datasets/zip.info`

this data, the human classification error rate is 2.5%. The best algorithms, e.g. the Tangent Distance algorithm (Simard et al., 1998) achieve this error rate, and SVM based algorithms perform slightly worse; they can achieve about 4% error (Mika et al., 2004).

We resized the grayscale images to 160×160, divided each pixel value by the sum of all pixel intensities, and considered each image as a 2d density function; We sampled 300 i.i.d. 2d coordinates from these densities, where the sampling probability of a 2d coordinate was proportional to the grayscale value of the point in the image. Finally we added Gaussian noise (zero mean, 0.1 variance) to these coordinates. Example noisy images are shown in Figure 1.

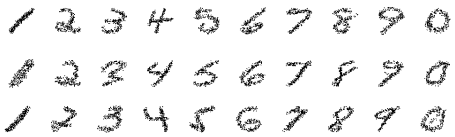


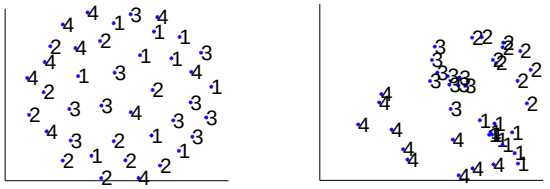
Figure 1: Noisy USPS dataset

This dataset is considerably more difficult than the original USPS dataset. In the original dataset, the raw images can be used as features for classification, and even the Euclidean distances between these images have high discriminative values. However, in the noisy USPS dataset the Euclidean distance between images becomes useless, and using the raw images as features performs poorly. We use 100-100 train/test splits and achieve only 67% accuracy using an SVM with a degree 3 polynomial kernel. However, using an SDM and estimating the same polynomial kernel nonparametrically with $k = 4$ nearest neighbors, we get 91% accuracy.

To demonstrate visually that our tools can keep more structure of the dataset than simply using the standard Euclidean metric between the raw images, we performed multidimensional scaling to 2d using both the Euclidean metric between the raw images (Fig. 2(a)) and using the estimation of the Euclidean distance between the distributions (Fig. 2(b)). We used 10 instances from letters {1,2,3,4}. We see that our method was able to preserve the class structure of the dataset; the letters form natural clusters. However, using raw images in MDS loses these clusters. Even in this unsupervised learning task, our method could keep the clusters of the letters. This explains why the classification task using distribution-based kernels was so much easier.

5.1.2 Natural Image Classification

One set of algorithms for classifying natural scene images is based on the so called "bag-of-words" (BoW)



(a) using raw images (b) using distributions

Figure 2: (a) MDS using Euclidean metric between the images. (b) MDS using the estimated Euclidean distance between the distributions of the 2d coordinates of the image pixels.

representation. With BoW, each image is considered as a collection of *visual words* (e.g. a coast image usually contains visual words of sky, sea and so on). The locations of and dependencies among those visual words are ignored. The collection of unique visual words is called the *visual vocabulary*.

Given a set of images, the visual words and vocabulary are often constructed in the following way: (1) Select local patches from each image. (2) Extract a feature vector for each patch. (3) Cluster/quantize all the patch features into V categories. By doing this, each category of patches would represent a visual concept such as "sky" or "window". These V categories form the "visual vocabulary", and each patch, represented by its category number, becomes a "visual word." (4) Within each image, count the number of different visual words and construct a histogram of size V. This is the BoW representation of the image.

SDM can be applied to image classification problems based on BoW. The key modification is that instead of discretizing the patch features as in step (3), we can use the features directly. SDM treats each image as a group of its patches' features, which are real-valued vectors, and then applies the proposed kernel functions to pairs of images to get the kernel matrix.

Dataset and feature extraction We use the image dataset published by Oliva and Torralba (2001) (OT). This dataset contains 8 outdoor scene categories: coast, mountain, forest, open country, street, inside city, tall buildings, and highways. There are 2688 images, each about 256×256 pixels.

The grayscale version of these images are used. We extract the dense SIFT features (Lowe (2004)) as in Bosch et al. (2008). For each image, we compute SIFT descriptors at points on a regular grid with a step size of 20. To achieve scale-invariance, at each point we compute three SIFT descriptors, each of which is 128-dimensional, with radii of $[6, 9, 12]$ pixels. After the feature extraction, each image is represented

by 409 vectors whose dimensionality is 128. Finally, PCA is applied to reduce their dimensionality to 19, preserving 70% of the total variance. We used the *VLFeat* package by Vedaldi and Fulkerson (2008) and its *PHOW* functionality to extract the SIFT features.

Performance evaluation We test the performance of SDM classification. In SDM, we used the Gaussian kernel $K(p,q) = \exp(-\mu^2(p,q)/(2\sigma^2))$, where $\mu(p,q)$ was either the KL-divergence (NP-KL) or the Hellinger distance (NP-Hel). We also tried the standard Euclidean distance, but found that it does not perform reliably in this high-dimensional situation.

To estimate the Hellinger distance based Gaussian kernel, we used the method presented in Section 3.2 and Section 4.2. To estimate the KL-divergence based Gaussian kernel, we applied the same approach, except that we approximated the KL divergence with Rényi-α divergence, where we used $\alpha = 0.99$. We applied $k = 3$ nearest neighbors in the estimator $\widehat{D}$ (Eq 4).

Note that it is not necessary to use non-parametric kernel estimators for the SDM. For comparison, we also implement parametric divergence estimators by doing density estimation based on single Gaussians (G-KL) and 5-component Gaussian mixture models (GMM-KL). For single Gaussians, the KL divergence can be obtained analytically, but for the mixture models we resort to the *Monte Carlo* method with 1000 samples. This strategy of kernel estimation is similar to what Jebara et al. (2004) proposed.

To apply SDM, we need to determine two parameters: the width of the kernel σ, and the cost parameter C for points that are in the margin. We select their values based on one run 5-fold cross-validation, individually for each divergence. The values of $\sigma = [5.63, 24.34, 7.16, 2.11], C = [2, 8, 8, 2]$ are used for G-KL, GMM-KL, NP-KL, NP-Hel respectively.

We evaluate the algorithms on 2-fold cross-validation runs on the whole dataset, following the set up used by Bosch et al. (2008). The performances of 16 random runs are reported in Figure 3. We also display the state-of-the-art accuracy from Bosch et al. (2008), who used SVM on grayscale BoW features (BoW), and BoW improved by PLSA (Hofmann (1999)) (BoW+PLSA). Most SDM methods achieved better results than these BoW methods, showing the advantage of continuous features over their discrete version. We can also see that the kernel based on the non-parametric KL divergence estimator performs better than those based on Gaussian/GMM density estimation, and achieves the best accuracy of 88.16%.

The OT dataset has been extensively used to evaluate scene classification algorithms. Therefore, it is possible to compare our performance and the previous results

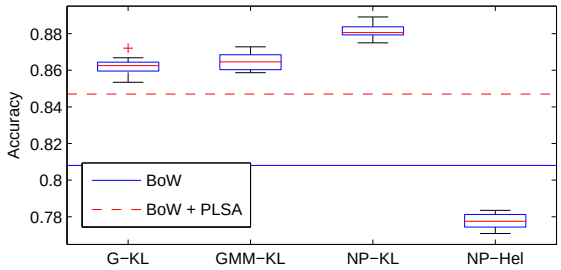


Figure 3: Image classification performances.

directly. In the original OT paper, Oliva and Torralba (2001) achieved an accuracy of 83.7% using a holistic representation called the *spatial envelope* of the images. Rasiwasia and Vasconcelos (2011) improves the BoW representation by introducing "contextual concepts" and achieved 85.60% accuracy. By using a spatial pyramid matching method that considers the spatial dependencies among visual words, Yin (2010) achieved 88.02% accuracy. Qin and Yung (2010a) proposed another way to encode the spatial dependencies of patches into the visual words and achieved 90.3% accuracy in 10-fold cross-validations. This result is further improved in Qin and Yung (2010b) to 91.57% where contextual visual words are combined with color information. For comparison, SDM based on the non-parametric KL divergence estimator achieves 89.95% average accuracy in 10-fold cross-validations.

We believe we have achieved the highest grayscale accuracy yet reported without using spatial dependencies. We expect further improvements by adding additional features to SDMs.

5.1.3 Turbulence Data

One of the challenges of doing science with modern large-scale simulations is identifying interesting phenomena in the results, finding them, and computing basic statistics about when and where they occurred.

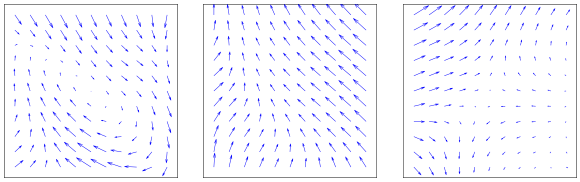
We performed an exploratory experiment on using SDM classifiers to assist in this process, using turbulence data from the JHU Turbulence Data Cluster[2] (Perlman et al., 2007) (TDC). TDC simulates fluid flow through time on a 3-dimensional grid, calculating 3-dimensional velocities and pressures of the fluid at each step. We used one time step of a contiguous $256 \times 256 \times 128$ sub-grid.

Our goal is to classify cross-sections of stationary vortices parallel to the xy plane. Each distribution is defined on the x and y components of each velocity in an 11×11 square, along with the squared magnitude of that point's distance from the center. The latter feature was included according to the intuition that

[2] `http://turbulence.pha.jhu.edu`

velocity in a vortex is related to its distance from the center of the vortex.

We trained an SDM on a manually labeled training set of 11 positives and 20 negatives, using a Gaussian kernel based on Hellinger distance. Some representative training examples are shown in Figure 4. Leave-one-out cross-validation on the training data yielded an accuracy of 97% (one mistake).



(a) Positive (b) Negative (c) Negative

Figure 4: Training examples for the vortex classifier.

This classifier was then used to evaluate groups along z-slices of the data, with a grid resolution of 2×2. One slice of the resulting probability estimates is shown in Figure 5(a); the arrows represent the mean velocity at each classification point. The high-probability region on the left is a canonical vortex, while the slightly-lower probability region in the upper-right deviates a little from the canonical form. Note that some other areas show somewhat complex velocity patterns but mostly have low probabilities.

5.2 Anomaly Detection with SDMs

As well as finding instances of known patterns, part of the process of exploring the results of a large-scale simulation is looking for unexpected phenomena.

To demonstrate anomaly detection with SDMs, we trained a one-class SDM on 100 distributions from the turbulence data, with centers chosen uniformly at random. Its evaluation on the same region as Figure 5(a) is shown in Figure 5(b). The two vortices are picked out, but the area with the highest score is a diamond-like velocity pattern, similar to Figure 4(c); these may well be less common in the dataset than are vortices.

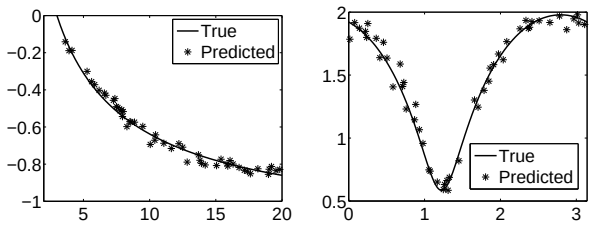
We believe that SDM classifiers and anomaly detectors serve as a proof of concept for a simulation exploration tool that would allow scientists to iteratively look for anomalous phenomena and label some of them. Classifiers could then find more instances of those phenomena and compute statistics about their occurrence, while anomaly detection would be iteratively refined to highlight only what is truly new.

5.3 DRSR Experiments

Below we show how distribution regression with scalar response (Section 3) can be used for learning real valued functionals of distributions from i.i.d. samples in a nonparametric way.

In the first experiment, we generated 150 sample sets from $Beta(a, 3)$ distributions where a was varied between $[3, 20]$ randomly. We had 100 sample sets for training and 50 for testing. Each sample set consisted of 500 $Beta(a, 3)$ distributed i.i.d. points. Our goal was to learn the skewness of $Beta(a, b)$ distributions. Figure 6(a) displays the predicted values for the 50 test sample sets. In this experiment we used a polynomial kernel with degree 3.

In the next experiment, we learn the entropy of Gaussian distributions. We randomly chose a 2×2 covariance matrix Σ, and then generated 150 sample sets from $\{\mathcal{N}(0, R(\alpha_i)\Sigma^{1/2})\}_{i=1}^{150}$. Where $R(\alpha_i)$ is a 2d rotation matrix with rotation angle $\alpha_i = i\pi/150$. From each $\mathcal{N}(0, R(\alpha_i)\Sigma^{1/2})$ distribution we sampled 500 2-dimensional i.i.d. points. Our goal was to learn the entropy of the first marginal distribution: $H = \frac{1}{2}\ln(2\pi e\sigma^2)$, where $\sigma^2 = M_{1,1}$ and $M = R(\alpha_i)\Sigma R^T(\alpha_i) \in \mathbb{R}^{2\times 2}$. Figure 6(b) shows the learned entropies of the 50 test sample sets.



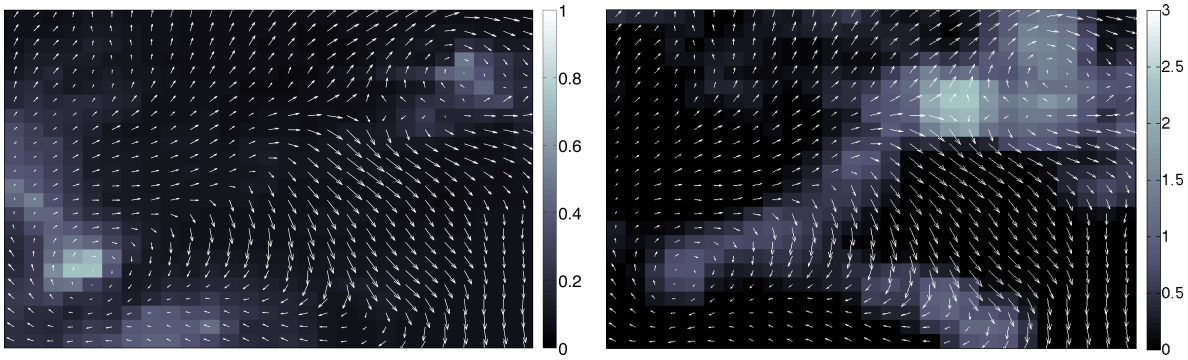
(a) Skewness of Beta (b) Entropy of Gaussian

Figure 6: (a) Learned skewness of $Beta(a, 3)$ distribution. Axis x: parameter a in $[3, 20]$. Axis y: skewness of $Beta(a, 3)$. (b) Learned entropy of a 1d marginal distribution of a rotated 2d Gaussian distribution. Axes x: rotation angle in $[0, \pi]$. Axis y: entropy.

5.4 Local Linear Embedding of Distributions

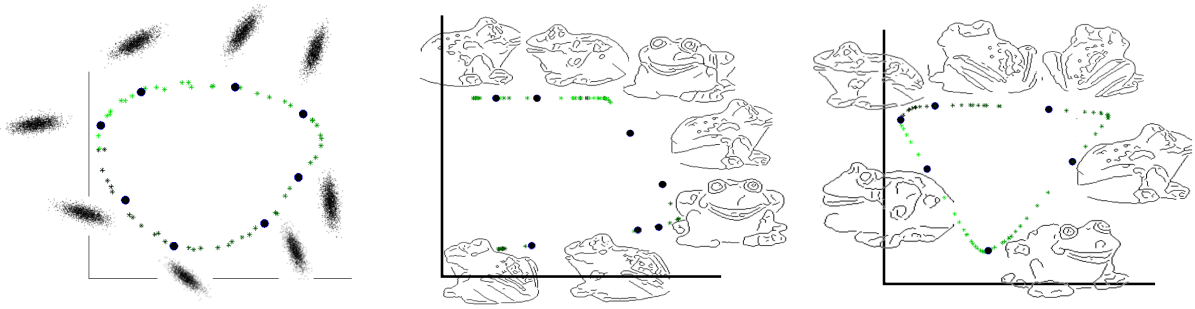
Here we show results on the local linear embedding extension to distributions. This algorithm uses the linear DRDR method as a subroutine.

We created a 2-dimensional zero mean Gaussian distribution with covariance matrix $\Sigma_{1,1} = 9$, $\Sigma_{1,2} = \Sigma_{2,1} = 0$, $\Sigma_{2,2} = 1$. Then we generated 2000 i.i.d. sample points from each of the rotated versions of this Gaussian distribution: $X_{i,n} \sim \mathcal{N}(0, R(\alpha_i)\Sigma^{1/2})$, $n = 1, \ldots, 2000$, $i = 1, \ldots, 63$. Here $R(\alpha_i)$ denotes the 2d rotation matrix with rotation angle $\alpha_i = (i-1)/20$. We ran the LLE algorithm (Section 3) and embedded these distribution into 2d. The results are shown in Figure 7(a). We can see that our method preserves



(a) Classification probabilities (b) Anomaly scores

Figure 5: Classification and anomaly scores along with velocities for one 54×48 slice of the turbulence data.



(a) Rotated 2d Gaussian distributions (b) using Euclidean image distances (c) results with kernel estimation

Figure 7: (a) LLE embedding of rotated 2d Gaussian distributions. (b) LLE embedding results using Euclidean distances between edge-detected images. The embedding failed to keep the local geometry of the original pictures. (c) LLE embedding results considering the edges as samples from unknown distributions. The embedding was successful; nearby objects are embedded to nearby places.

(a) Original image (b) Edge-detected

Figure 8: The original and the edge-detected object.

the local geometry of these sample sets. Distributions close to each other are mapped into nearby points.

We repeated this experiment on the edge-detected images of an object in the COIL dataset[3]. This dataset consists of 72 128x128 pictures of a rotated 3D object (Fig. 8(a)). We converted these images to grayscale and performed Canny edge detection on them (Fig. 8(b)). The number of detected edge points on these images was between 845 and 1158.

Our goal is to embed these edge-detected images into a 2d space preserving proximity. While this problem is easy using the original images, it is challenging when only the edge-detected images are available. If we simply use the Euclidean distances between these edge-detected images, the standard LLE algorithm fails (Fig. 7(b)). However, when we consider the edge-detected images as sample points from unknown 2d distributions, and use the LLE algorithm on distributions (Section 3), the embedding is successful. The embedded points preserve proximity and the local geometry of the original images (Fig. 7(c)).

6 Discussion and Conclusion

We have posed the problem of performing machine learning on groups of data points as one of machine learning on distributions where the data points are viewed as samples from an underlying distribution for the group. We proposed support distribution machines for learning on distributions and provided non-parametric methods for estimating the necessary kernel matrices. We showed that our methods work across a range of supervised and unsupervised tasks and in one case believe we have achieved the best classification accuracy yet reported.

[3] `http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php`

References

Berlinet, A. and Thomas, C. (2004). *Reproducing kernel Hilbert spaces in Probability and Statistics*. Kluwer Academic Publishers.

Borg, I. and Groenen, P. (2005). *Modern Multidimensional Scaling: theory and applications*. Springer-Verlag, New York.

Bosch, A., Zisserman, A., and Munoz, X. (2008). Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. PAMI*, 30(4).

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41-3.

Goria, M., Leonenko, N., Mergel, V., and Inverardi, N. (2005). A new class of random vector entropy estimators and its applications in testing statistical hypotheses. *Journal of Nonparametric Statistics*, 17:277–297.

Higham, N. J. (2002). Computing the Nearest Correlation Matrix a Problem From Finance. *IMA Journal of Nummerical Analysis*, pages 329–343.

Hinton, G. and Roweis, S. (2002). Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*.

Hull, J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 550–554.

Jebara, T., Kondor, R., Howard, A., Bennett, K., and Cesa-bianchi, N. (2004). Probability product kernels. *Journal of Machine Learning Research*, 5:819–844.

Kadri, H., Duflos, E., Preux, P., Canu, S., and Davy, M. (2010). Nonlinear functional regression: a functional rkhs approach. *Journal of Machine Learning Research - Proceedings Track*, pages 374–380.

Kadri, H., Rabaoui, A., Preux, P., Duflos, E., and Rakotomamonjy, A. (2011). Functional regularized least squares classication with operator-valued kernels. In *Proceedings of the 28th International Conference on Machine Learning*, pages 993–1000, New York, NY, USA. ACM.

Leonenko, N., Pronzato, L., and Savani, V. (2008). A class of Rényi information estimators for multidimensional densities. *Annals of Statistics*, 36(5):2153–2182.

Lian, H. (2007). Nonlinear functional models for functional responses in reproducing kernel hilbert spaces. *The Canadian Journal of Statistics*, 35(4):597–606.

Loftsgaarden, D. and Quesenberry, C. (1965). A nonparametric estimate of a multivariate density function. *Annals of Mathematical Statistics*, 36:1049–1051.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110.

Micchelli, C. and Pontil, M. (2005). On learning vector-valued functions. *Neural Comput.*, 17:177–204.

Mika, S., Schäfer, C., Laskov, P., Tax, D., and Müller, K.-R. (2004). *Handbook of Computational Statistics: Concepts and Methods*, chapter Support Vector Machines. Springer. J.E. Gentle, W. H"ardle, and Y. Mori.

Muller, H., Chiou, J., and Leng, X. (2008). Inferring gene expression dynamics via functional regression analysis. *BMC Bioinformatics*, 9(60).

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3).

Perlman, E., Burns, R., Li, Y., and Meneveau, C. (2007). Data exploration of turbulence simulations using a database cluster. In *Supercomputing SC*.

Póczos, B. and Schneider, J. (2011). On the estimation of α-divergences. In *14th International Conference on AI and Statistics, Ft. Lauderdale, FL, USA*.

Póczos, B., Xiong, L., and Schneider, J. (2011). Nonparametric divergence estimation with applications to machine learning on distributions. In *27th Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain*.

Preda, C. (2007). Regression models for functional data by reproducing kernel hilbert spaces methods. *Journal of Statistical Planning and Inference*, 137(3):829 – 840.

Qin, J. and Yung, N. H. (2010a). Scene categorization via contextual visual words. *Pattern Recognition*, 43.

Qin, J. and Yung, N. H. (2010b). Sift and color feature fusion using localized maximum-margin learning for scene classification. In *International Conference on Machine Vision*.

Ramsay, J. O. and Silverman, B. (2005). *Functional data analysis*. Springer, New York, 2nd edition.

Rasiwasia, N. and Vasconcelos, N. (2011). Holistic context models for visual recognition. *IEEE Trans. PAMI*.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471.

Schölkopf, B. and Smola, A. (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. The MIT Press.

Simard, P., LeCun, Y., Denker, J., and Victorri, B. (1998). Transformation invariance in pattern recognition - tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade*, LNCS 1524, pages 239–274. Springer.

Sun, J., Fyfe, C., and Crowe, M. (2010). Curvilinear component analysis and bregman divergences. In *European Symposium on Artificial Neural Networks*.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, New York, NY, USA. ACM.

van der Wart, A. (2007). *Asymptotic Statistics*. Cambridge University Press.

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

Wang, Q., Kulkarni, S. R., and Verdú, S. (2009). Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5).

Yin, H. (2010). Scene classification using spatial pyramid matching and hierarchical Dirichlet processes. Master's thesis, Rochester Institue of Technology.

APPENDIX—SUPPLEMENTARY MATERIAL

A Consistency of k-NN density estimators

The following theorems show the consistency of k-NN density estimators (Loftsgaarden and Quesenberry, 1965).

Theorem 1 (convergence in probability) *If $k(n)$ denotes the number of neighbors applied at sample size n, $\lim_{n\to\infty} k(n) = \infty$, and $\lim_{n\to\infty} n/k(n) = \infty$, then $\hat{p}_{k(n)}(x) \to_p p(x)$ for almost all x.*

Theorem 2 (convergence in sup norm) *If $\lim_{n\to\infty} k(n)/\log(n) = \infty$ and $\lim_{n\to\infty} n/k(n) = \infty$, then $\lim_{n\to\infty} \sup_x \left|\hat{p}_{k(n)}(x) - p(x)\right| = 0$ almost surely.*

B Consistency of $\widehat{D}_{\alpha,\beta}(X_{1:n}\|Y_{1:m})$

Our goal is to estimate

$$D_{\alpha,\beta}(p\|q) = \int_{\mathcal{M}} p^{\alpha}(x) q^{\beta}(x) p(x) \mathrm{d}px,$$

and our proposed estimator is

$$\widehat{D}_{\alpha,\beta}(X_{1:n}\|Y_{1:m}) = \frac{1}{n}\sum_{i=1}^{n} (n-1)^{-\alpha} m^{-\beta} \rho_k^{-d\alpha}(i) \nu_k^{-d\beta}(i) B_{k,\alpha,\beta},$$

where $B_{k,\alpha,\beta} \doteq \bar{c}^{-\alpha-\beta} \frac{\Gamma(k)^2}{\Gamma(k-\alpha)\Gamma(k-\beta)}$.

Let $\mathcal{B}(x,R)$ denote a closed ball around $x \in \mathbb{R}^d$ with radius R, and let $\mathcal{V}\big(\mathcal{B}(x,R)\big) = \bar{c}R^d$ be its volume, where $\bar{c}$ stands for the volume of a d-dimensional unit ball. The following theorems we will assume that almost all points of $\mathcal{M}$ are in its interior and that $\mathcal{M}$ has the following additional property:

$$\inf_{0<\delta<1} \inf_{x\in\mathcal{M}} \frac{\mathcal{V}\big(\mathcal{B}(x,\delta) \cap \mathcal{M}\big)}{\mathcal{V}\big(\mathcal{B}(x,\delta)\big)} \doteq r_{\mathcal{M}} > 0.$$

If $\mathcal{M}$ is a finite union of bounded convex sets, then this condition holds.

Theorem 3 (Asymptotic unbiasedness) *Let $-k < \alpha, \beta < k$. If $0 < \alpha < k$, then let p be bounded away from zero and uniformly continuous. If $-k < \alpha < 0$, then let p be bounded. Similarly, if $0 < \beta < k$, then let q be bounded away from zero and uniformly continuous. If $-k < \beta < 0$, then let q be bounded. Under these conditions we have that*

$$\lim_{n,m\to\infty} \mathbb{E}\left[\widehat{D}_{\alpha,\beta}(X_{1:n}\|Y_{1:m})\right] = D_{\alpha,\beta}(p\|q),$$

i.e., the estimator is asymptotically unbiased.

The following theorem provides conditions under which $\widehat{D}_{\alpha,\beta}$ is L_2 consistent. In the previous theorem we have stated conditions that lead to asymptotically unbiased divergence estimation. In the following theorem we will assume that the estimator is asymptotically unbiased for (α,β) as well as for $(2\alpha, 2\beta)$, and also assume that $D_{\alpha,\beta}(p\|q) < \infty, D_{2\alpha,2\beta}(p\|q) < \infty$.

Theorem 4 (L_2 consistency) *Let $k \geq 2$ and $-(k-1)/2 < \alpha, \beta < (k-1)/2$. If $0 < \alpha < (k-1)/2$, then let p be bounded away from zero and uniformly continuous. If $-(k-1)/2 < \alpha < 0$, then let p be bounded. Similarly, if $0 < \beta < (k-1)/2$, then let q be bounded away from zero and uniformly continuous. If $-(k-1)/2 < \beta < 0$, then let q be bounded. Under these conditions we have that*

$$\lim_{n,m\to\infty} \mathbb{E}\left[\left(\widehat{D}_{\alpha,\beta}(X_{1:n}\|Y_{1:m}) - D_{\alpha,\beta}(p\|q)\right)^2\right] = 0;$$

that is, the estimator is L_2 consistent.

B.1 Proof Outline for Theorems 3-4

We can repeat the argument of Póczos and Schneider (2011). Using the k-NN density estimator, we can estimate $1/p(x)$ by $N\bar{c}\rho_k^d(x)/k$. From the Lebesgue lemma, one can prove that the distribution of $N\bar{c}\rho_k^d(x)$ converges weakly to an Erlang distribution with mean $k/p(x)$, and variance $k/p^2(x)$ (Leonenko et al., 2008). In turn, if we divide $N\bar{c}\rho_k^d(x)$ by k, then asymptotically it has mean $1/p(x)$ and variance $1/(kp^2(x))$. This implies that indeed (in accordance with Theorems 1–2) k should diverge in order to get a consistent estimator, otherwise the variance will not disappear. On the other hand, k cannot grow too fast: if, say, $k = N$, then the estimator would be simply $\bar{c}\rho_k^d(x)$, which is a useless estimator since it is asymptotically zero whenever $x \in \mathrm{supp}(p)$.

Luckily, in our case we do not need to apply consistent density estimators. The trick is that (3) has a special form: $\int p(x)p^\alpha(x)q^\beta(x)\mathrm{d}x$. In (4) this is estimated by

$$\frac{1}{N}\sum_{i=1}^{N}\left(\hat{p}_k(X_i)\right)^\alpha\left(\hat{q}_k(X_i)\right)^\beta B_{k,\alpha,\beta}, \tag{5}$$

where $B_{k,\alpha,\beta}$ is a correction factor that ensures asymptotic unbiasedness. Using the Lebesgue lemma again, we can prove that the distributions of $\hat{p}_k(X_i)$ and $\hat{q}_k(X_i)$ converge weakly to the Erlang distribution with means $k/p(X_i)$, $k/q(X_i)$ and variances $k/p^2(X_i)$, $k/q^2(X_i)$, respectively (Leonenko et al., 2008). Furthermore, they are conditionally independent for a given X_i. Therefore, "in the limit" (5) is simply the empirical average of the products of the αth (and βth) powers of independent Erlang distributed variables. These moments can be calculated in closed form. For a fixed k, the k-NN density estimator is not consistent since its variance does not vanish. In our case, however, this variance will disappear thanks to the empirical average in (5) and the law of large numbers.

While the underlying ideas of this proof are simple, there are a couple of serious gaps in it. Most importantly, from the Lebesgue lemma we can guarantee only the weak convergence of $\hat{p}_k(X_i)$, $\hat{q}_k(X_i)$ to the Erlang distribution. From this weak convergence we cannot imply that the moments of the random variables converge too. To handle this issue, we will need stronger tools such as the concept of asymptotically uniformly integrable random variables (van der Wart, 2007), and we also need the uniform generalization of the Lebesgue lemma. As a result, we need to put some extra conditions on the densities p and q in Theorems 3–4. The details follow from the slight generalization of the derivations in Póczos and Schneider (2011).