Multi-Kernel Probability Distribution Regressions

Pingping Zhu Hongchuan Wei Wenjie Lu Silvia Ferrari
Department of Mechanical Engineering
and Materials Science
Duke University
Durham, NC 27708
Email: {pingping.zhu, hongchuan.wei, wenjie.lu, silvia.ferrari }@duke.edu

***Abstract*—This paper presents a multi-layer reproducing kernel Hilbert space (RKHS) approach for probability distribution to real and probability distribution to function regressions. The approach maps the distributions into RKHS by distribution embeddings and, then, constructs a multi-layer RKHS within which the multi-kernel distribution regression can be implemented using an existing kernel regression algorithm, such as kernel recursive least squares (KRLS). The numerical simulations on synthetic data obtained via Gaussian mixtures show that the proposed approach outperforms existing probability distribution (DR) regression algorithms by achieving smaller mean squared errors (MSEs) and requiring less training samples.**

I. Introduction

Classical regression analysis is concerned with learning a vector function or mapping from a real-valued input vector to a real-valued output vector from data. Examples of linear and nonlinear regression algorithms that have been proposed over the past decades include recursive least square (RLS) [1], least mean square (LMS) [1], artificial neural networks (ANNs) [2], Gaussian process regression (GPR) [3] and kernel adaptive filtering (KAF) [4]. Classical regression algorithms are often restricted by the dimensionality of the input and output domains and, thus, a great deal of attention has been recently devoted to functional regression in which a mapping from finite dimensional spaces to an infinite dimensional domain is to be learned from data [5]. In particular, distribution regression (DR) algorithms have been proposed for learning probability density functions from data, such as distribution to real regression (DRR) [6] and distribution to distribution regression (DDR) [7], in which input covariates are arbitrary distributions and output responses are real values and distributions, respectively.

The DRR algorithm proposed in [6] utilizes two kernels and, thus, is referred to as Kernel-Kernel DRR (KKDRR). One kernel is utilized in the kernel density estimator (KDE) [8] in order to estimate the probability density function (PDF) of the input distribution from samples, and the second kernel is used to measure the divergence between the testing and training PDFs. A Kernel-Kernel estimator is then applied to approximate the real-valued output based on these divergences. The KKDRR algorithm can be implemented easily, however, it suffers from two limitations. Firstly, KKDRR computes only the divergences between testing distributions and training distributions, but does not consider mutual similarities between all testing distributions. Secondly, because the calculations of the divergences involve numerical integrals, the accuracy depends on the number of grid sample points.

This paper presents a new multi-kernel approach for DRR that utilizes multi-layer RKHS mappings to solve two classes of functional regression problems. The first problem is to learn a mapping from probability distributions to real numbers, and the second problem is to learn a mapping from probability distributions to functions. RKHS methods, such as kernel recursive least square (KRLS) [9], kernel least mean square (KLMS) [10], quantized kernel recursive least square (QKRLS) [11] and quantized kernel least square (QKLMS) [12], have proven very effective at solving nonlinear regression problems because they are nonparametric and, thus, can adjust the model dimensionality to the data and greatly improve learning capabilities. In the new functional regression approach presented in this paper, probability distributions are first mapped into an RKHS, using the concept of distribution embeddings [13]–[15]. Then, the distribution embeddings in RKHS are used as inputs for an extended kernelized regression algorithm (KRLS) to implement the multi-kernel DRR (MKDRR).

The extended kernel used in this paper was first introduced by Christmann and Steinwart in [16] and was later used in [17] to develop support vector machines (SVMs) for distributional inputs. In this paper, the extended kernel is used to develop and implement the MKDRR and to develop a new framework for distribution regression and distribution to function regression (DFR), referred to as multi-kernel DFR (MKDFR). To our best knowledge, the MKDFR is the first algorithm developed for DFR and, unlike KKDRR, the multi-kernel distribution regression approach presented in this paper is not limited by the number of grid points, because the divergences are calculated directly from the samples, and the mutual divergences between all training distributions are considered to approximate the outputs. The DDR and DFR problems are formulated in Section II, and the multi-kernel distribution regressions approach is presented in Section III. The KRLS-based DRR and DFR algorithms described in Section IV are implemented and evaluated using the numerical simulations in Section V, which show that the proposed MKDRR algorithm outperforms existing KKDRR algorithms by achieving lower MSEs and requiring smaller numbers of training samples.

II. Formulation of Probability Distribution Regression Problems

Let $\mathcal{I}$ be a family of distributions that are compact with respect to the Lebesgue measure. We consider the DRR problem of inferring a mapping $\mathcal{F}: \mathcal{I} \mapsto \mathbb{R}^{n_z}$ from training sets $(P_1, \mathbf{z}_1), \ldots, (P_T, \mathbf{z}_T)$, such that,

$$\mathbf{z}_k = \mathcal{F}(P_k) + \boldsymbol{\epsilon}_k, \quad k = 1, \ldots, T \tag{1}$$

978-1-4799-1959-8/15/$31.00

where, $P_k \in \mathcal{I}$ is a probability distribution, $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is the corresponding output response, and $\boldsymbol{\epsilon}_k$ is a zero mean Gaussian noise variable. However, in real-world applications, the probability distribution of interest, P_k, cannot be always represented in closed analytic form. Thus, it is assumed that P_k is to be approximated from a data set $\mathcal{X}_k = \{\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,s}, \ldots, \mathbf{x}_{k,N_k}\}$, where $\mathbf{x}_{k,s} \in \mathbb{R}^{n_x}$ are *independent and identically distributed* (*i.i.d.*) samples drawn from P_k, and N_k is the number of samples in the data set $\mathcal{X}_k$. Then, the DRR problem is to learn the mapping $\mathcal{F}$ from the data set of observations $\mathcal{D}_{\mathcal{F}} = \{(\mathcal{X}_1, \mathbf{z}_1), \ldots, (\mathcal{X}_T, \mathbf{z}_T)\}$, and to predict any output $\mathbf{z} \in \mathbb{R}^{n_z}$, from a corresponding data set $\mathcal{X}$ drawn from a distribution P, where $(\mathcal{X}, \mathbf{z})$ are not necessarily in $\mathcal{D}_{\mathcal{F}}$.

Also, consider the DFR problem of learning a mapping $\mathcal{G} : \mathcal{I} \mapsto \mathbb{F}$ from the training pairs $(P_1, f_1), \ldots, (P_T, f_T)$, such that,

$$f_k = \mathcal{G}(P_k) + \boldsymbol{\varepsilon}_k \tag{2}$$

and where the functions $f_k \in \mathbb{F}$, $k = 1, \ldots, T$, are defined in a function space $\mathbb{F}$, and $\boldsymbol{\varepsilon}_k$ is a zero mean Gaussian process. One can find that the mapping $\mathcal{G}$ is an operator and, as in the DRR problem, a sample set $\mathcal{X}_k$ can be observed for each P_k. Then, the DFR problem is to learn this mapping $\mathcal{G}$ and predict a new function f from a new data set $\mathcal{X}$ drawn from a new distribution P.

For each output function f_k, the pairs in the set $\mathcal{Y}_k = \{(\mathbf{y}_{k,1}, \mathbf{z}_{k,1}), \ldots, (\mathbf{y}_{k,M}, \mathbf{z}_{k,M})\}$ can be observed, where $\mathbf{y}_{k,m} \in \mathbb{R}^{n_y}$ and $\mathbf{z}_{k,m} \in \mathbb{R}^{n_z}$ are the input and the corresponding output for function f_k, respectively, such that

$$\mathbf{z}_{k,m} = f_k(\mathbf{y}_{k,m}), \quad \forall k, m \tag{3}$$

It follows that the observed data for DFR are $\mathcal{D}_{\mathcal{G}} = \{(\mathcal{X}_1, \mathcal{Y}_1), \ldots, (\mathcal{X}_T, \mathcal{Y}_T)\}$, and substituting (2) into (3), the (3) can be written as

$$\mathbf{z}_{k,m} = \mathcal{G}(P_k)(\mathbf{y}_{k,m}) + \boldsymbol{\varepsilon}_k(\mathbf{y}_{k,m}) \tag{4}$$

If a new mapping, $\mathcal{G}'$, is defined such that,

$$\mathcal{G}'(P_k, \mathbf{y}_{k,m}) = \mathcal{G}(P_k)(\mathbf{y}_{k,m}), \tag{5}$$

then (4) can be written as,

$$\mathbf{z}_{k,m} = \mathcal{G}'(P_k, \mathbf{y}_{k,m}) + \boldsymbol{\epsilon}'_k \tag{6}$$

where the evaluation of $\boldsymbol{\varepsilon}(\mathbf{y}_{k,m})$ is denoted by $\boldsymbol{\epsilon}'_k$, which is a zero mean Gaussian noise. Comparing (6) to (1), it can be seen that for both mappings $\mathcal{F}$ and $\mathcal{G}'$ the outputs are real values. From (6), if a probability distribution P_k is given, the approximation of $\mathbf{z}_{k,m}$ can be obtained from any input $\mathbf{y}_{k,m}$, specifying a mapping from $\mathbf{y}$ to $\mathbf{z}$. Therefore, the mapping $\mathcal{G}'$ can be learned in lieu of $\mathcal{G}$ and, thus, in the remainder of the paper, the notations $\mathcal{G}$ and $\mathcal{G}'$ are interchangeable.

III. Multi-Kernel Distribution Regressions Methodology

The approach for mapping probability distributions into an RKHS and for constructing a multi-layer RKHS (ML-RKHS) is described in the following subsections and schematized in Fig. 1. The implementation of multi-kernel DRR (MKDRR) and multi-kernel DRF (MKDRF) through the ML-RKHS approach are presented in Section IV.

A. Distribution Embeddings

Given a random variable (R.V.) $X \in \mathbb{R}^{n_x}$ associated with a distribution P_X and a corresponding Probability Density Function (PDF) p_X, an embedding $\boldsymbol{\mu}_X$ in RKHS can be defined as,

$$\boldsymbol{\mu}_X := \mathbf{E}_X[k_X(X, \cdot)] = \int p_X(\mathbf{x}) k_X(\mathbf{x}, \cdot) d(\mathbf{x}) \tag{7}$$

where $\mathbf{E}_X[\cdot]$ indicates the expectation operator, $k_X(\cdot, \cdot)$ is a kernel defined on $\mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ associated with RKHS $\mathcal{H}_X$ [13]–[15]. It can be shown that the distribution embedding $\boldsymbol{\mu}_X$ is also in the RKHS $\mathcal{H}_X$, provided $\mathbf{E}_X[k_X(X, X)] < \infty$. Its empirical estimate is,

$$\hat{\boldsymbol{\mu}}_X = \frac{1}{N} \sum_{n=1}^{N} k_X(\mathbf{x}_n, \cdot) \tag{8}$$

where $\mathcal{D}_X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a training set that is assumed to have been drawn *i.i.d* from P_X. According to [14], [15], it is guaranteed that by a characteristic kernel, the mapping from distribution P_X to the distribution embedding $\boldsymbol{\mu}_X \in \mathcal{H}_X$ is injective. A famous characteristic kernel is the Gaussian kernel, which is used in this paper to specify the kernel function $k_X(\cdot, \cdot)$.

B. Kernel Design and Multi-Layer RKHS

The distribution embedding $\boldsymbol{\mu}_X \in \mathcal{H}_X$ can represent the corresponding distribution distinguishably [13]–[15] Therefore, the regression based on the distributions can be derived using the distribution embeddings. Similarly to the existing kernel regression algorithms [3], [9]–[12], we can define new kernels on $\mathcal{H}_X$, map the distribution embeddings to the new RKHS associated with the defined kernels, and develop the multi-kernel regression algorithms in these new RKHS.

In order to implement the DRR, we first map the distribution data sets $\mathcal{X}_i$ and $\mathcal{X}_j$, $i, j = 1, \ldots, T$, into $\mathcal{H}_X$ as distribution embeddings $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ by the Gaussian kernel $k_X(\cdot, \cdot)$ with the kernel size $\sigma_{\mathbf{x}}$. Then, we can have the extended kernel $\mathcal{K}_{\mathcal{F}}(\cdot, \cdot)$ on $\mathcal{H}_X \times \mathcal{H}_X$, which was introduced in [16],

$$\mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \exp\left[-\frac{D(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)}{2\sigma_{\boldsymbol{\mu}}^2}\right] \tag{9}$$

where $\sigma_{\boldsymbol{\mu}}$ is the kernel size and $D(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$ is the squared distance between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ in $\mathcal{H}_X$, which can be expressed by

$$\begin{aligned} D(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) &= \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2_{\mathcal{H}_X} \\ &= \|\boldsymbol{\mu}_i\|^2_{\mathcal{H}_X} + \|\boldsymbol{\mu}_j\|^2_{\mathcal{H}_X} - 2\left\langle \boldsymbol{\mu}_i, \boldsymbol{\mu}_j \right\rangle_{\mathcal{H}_X} \end{aligned} \tag{10}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}_X}$ denotes the inner product in RKHS $\mathcal{H}_X$. The distribution embeddings $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ can be approximated from samples $\mathcal{X}_i = \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,N_i}\}$ and $\mathcal{X}_j = \{\mathbf{x}_{j,1}, \ldots, \mathbf{x}_{j,N_j}\}$. According to (8), the inner product of the two distribution embeddings can be approximated by

$$\begin{aligned} \left\langle \boldsymbol{\mu}_i, \boldsymbol{\mu}_j \right\rangle_{\mathcal{H}_X} &\approx \left\langle \frac{1}{N_i} \sum_{m=1}^{N_i} k_X(\mathbf{x}_{i,m}, \cdot), \frac{1}{N_j} \sum_{n=1}^{N_j} k_X(\mathbf{x}_{j,n}, \cdot) \right\rangle_{\mathcal{H}_X} \\ &= \frac{1}{N_i N_j} \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} k_X(\mathbf{x}_{i,m}, \mathbf{x}_{j,n}) \end{aligned} \tag{11}$$

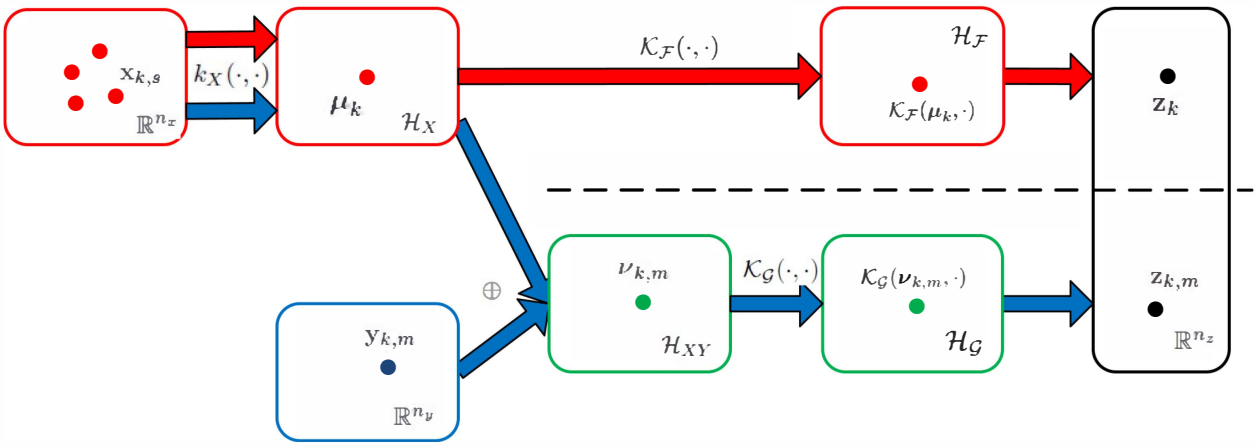

Fig. 1. The frameworks of the multi-layer RKHS to implement MKDRR (red arrows) and MKDFR (blue arrows), where red points denote the distribution representations, blue point denotes the input point and the green points denote the combination points, and the black points denote the desired points

Similarly, $\|\boldsymbol{\mu}_i\|^2_{\mathcal{H}_X}$ and $\|\boldsymbol{\mu}_j\|^2_{\mathcal{H}_X}$ can also be approximated using the data sets $\mathcal{X}_i$ and $\mathcal{X}_j$, respectively.

From (9), it can be seen that $\mathcal{K}_{\mathcal{F}}$ is a Gaussian-like kernel. The RKHS associated with the kernel $\mathcal{K}_{\mathcal{F}}$ is denoted by $\mathcal{H}_{\mathcal{F}}$, as shown in Fig. 1. It can be easily shown that the proposed functional $\mathcal{K}_{\mathcal{F}}$ is symmetrical and positive definite, which means $\mathcal{K}_{\mathcal{F}}$ is a positive definite kernel. In order to implement the MKDFR by (6), a new kernel is designed. First, $\mathcal{X}_i$ is mapped to $\mathcal{H}_X$ as the distribution embedding $\boldsymbol{\mu}_i$ by the Gaussian kernel $k_X(\cdot,\cdot)$, which is specified by the kernel size $\sigma_{\mathbf{x}}$. Then, the distribution embeddings $\boldsymbol{\mu}_i$ and $\mathbf{y}_{i,m}$ are combined into a new term $\boldsymbol{\nu}_{i,m} = \left[\boldsymbol{\mu}_i^T, \mathbf{y}_{i,m}^T\right]^T \in \mathcal{H}_X \oplus \mathbb{R}^{n_y}$, where $\oplus$ denotes the direct product operator. In short, we denote $\mathcal{H}_X \oplus \mathbb{R}^{n_y}$ by $\mathcal{H}_{XY}$. Note that the space $\mathcal{H}_{XY}$ is not an RKHS. Finally, by treating this new combination term $\boldsymbol{\nu}_{i,m}$ as an input, a new kernel $\mathcal{K}_{\mathcal{G}}(\cdot,\cdot)$ can be defined on $\mathcal{H}_{XY} \times \mathcal{H}_{XY}$ and associated with the RKHS $\mathcal{H}_{\mathcal{G}}$, such that,

$$\mathcal{K}_{\mathcal{G}}(\boldsymbol{\nu}_{i,m}, \boldsymbol{\nu}_{j,n}) = \exp\left[-\frac{(\boldsymbol{\nu}_{i,m} - \boldsymbol{\nu}_{j,n})^T \boldsymbol{\Sigma}_{XY}^{-1} (\boldsymbol{\nu}_{i,m} - \boldsymbol{\nu}_{j,n})}{2}\right] \tag{12}$$

with covariance,

$$\boldsymbol{\Sigma}_{XY} = \begin{bmatrix} \sigma_{\boldsymbol{\mu}}^2 \mathbf{I}_{\mathcal{H}_X} & \mathbf{0} \\ \mathbf{0} & \sigma_{\mathbf{y}}^2 \mathbf{I}_q \end{bmatrix}, \tag{13}$$

and where $\sigma_{\boldsymbol{\mu}}, \sigma_{\mathbf{y}} \in \mathbb{R}$, $\mathbf{I}_{\mathcal{H}_X} \in \mathcal{H}_X \times \mathcal{H}_X$, and $\mathbf{I}_{n_y} \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}$ are an identity operator and an identity matrix, respectively. The kernel $\mathcal{K}_{\mathcal{G}}(\cdot,\cdot)$ is a multivariate-normal-like kernel, which is also a positive definite kernel and can be extended as follows,

$$\begin{aligned}\mathcal{K}_{\mathcal{G}}(\boldsymbol{\nu}_{i,m}, \boldsymbol{\nu}_{j,n}) &= \exp\left[-\frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2_{\mathcal{H}_X}}{2\sigma_{\boldsymbol{\mu}}^2} - \frac{\|\mathbf{y}_{i,m} - \mathbf{y}_{j,n}\|^2}{2\sigma_{\mathbf{y}}^2}\right] \\ &= \mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) k_Y(\mathbf{y}_{i,m}, \mathbf{y}_{j,n})\end{aligned} \tag{14}$$

where,

$$k_Y(\mathbf{y}_{i,m}, \mathbf{y}_{j,n}) = \exp\left[-\frac{\|\mathbf{y}_{i,m} - \mathbf{y}_{j,n}\|^2}{2\sigma_{\mathbf{y}}^2}\right] \tag{15}$$

is a kernel defined on $\mathbb{R}^{n_y}$ associated with RKHS $\mathcal{H}_Y$.

With the proposed new kernels $\mathcal{K}_{\mathcal{F}}(\cdot,\cdot)$ and $\mathcal{K}_{\mathcal{G}}(\cdot,\cdot)$, the distribution P_i and $\boldsymbol{\nu}_{i,m}$ can be mapped into RKHS $\mathcal{H}_{\mathcal{F}}$ and $\mathcal{H}_{\mathcal{G}}$, respectively. By this approach, existing kernel regression algorithms, such as KAF and GPR, can then be implemented to solve the DRR and DFR problems formulated in Section II.

IV. Implementation of Multi-Kernel Distribution Regressions based on KRLS

As an example, the KRLS algorithm is utilized to implement the MKDRR (Section IV-A) and the MKDFR (Section IV-B) methods, based on the ML-RKHS approach presented in Section III.

A. Multi-Kernel Distribution to Real Regression

Because of the injective mapping of distribution embeddings and the kernel trick of the kernel $\mathcal{K}_{\mathcal{F}}$, we have

$$\mathcal{F}(P_k) = \langle \boldsymbol{\omega}_{\mathcal{F}}, \mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_k, \cdot)\rangle_{\mathcal{H}_{\mathcal{F}}} \tag{16}$$

where the feature weight $\boldsymbol{\omega}_{\mathcal{F}} \in \mathcal{H}_{\mathcal{F}}$ represents the mapping $\mathcal{F}$. Like the standard KRLS algorithm, we minimize the following cost function to learn the DRR defined in (1) from data sets $\mathcal{D}_{\mathcal{F}}$,

$$\begin{aligned}J_{DRR} = \min_{\boldsymbol{\omega}_{\mathcal{F}}} \Bigg[&\sum_{k=1}^{T} \left\| \mathbf{z}_k - \langle \boldsymbol{\omega}_{\mathcal{F}}, \mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_k, \cdot)\rangle_{\mathcal{H}_{\mathcal{F}}} \right\|^2 \\ &+ \lambda \|\boldsymbol{\omega}_{\mathcal{F}}\|^2_{\mathcal{H}_{\mathcal{F}}} \Bigg]\end{aligned} \tag{17}$$

where λ is a regularization factor. To this end, we obtain the same form of cost function with the standard KRLS. By introducing the feature matrices $\boldsymbol{\Phi}_k = [\mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_1, \cdot), \ldots, \mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_k, \cdot)]$ and desired matrices $\mathbf{Z}_k = [\mathbf{z}_1, \ldots, \mathbf{z}_k]^T$, we can approximate the feature weight $\boldsymbol{\omega}_{\mathcal{F}}$ at the kth iteration by

$$\boldsymbol{\omega}_{\mathcal{F}} = \boldsymbol{\Phi}_k \left[\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k + \lambda \mathbf{I}_k\right]^{-1} \mathbf{Z}_k = \boldsymbol{\Phi}_k \mathbf{Q}_{\boldsymbol{\mu}}(k) \mathbf{Z}_k \tag{18}$$

where $\mathbf{I}_k$ is a $k \times k$ identity matrix. Here, the inverse matrix $\mathbf{Q}_{\boldsymbol{\mu}}(k) = \left[\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k + \lambda \mathbf{I}_k\right]^{-1}$ can be calculated recursively like the KRLS algorithm with a computational complexity of $O(k^2)$ at the kth iteration.

Once the feature weight $\boldsymbol{\omega}_{\mathcal{F}}$ is approximated, we can calculate the predicted output $\mathbf{z}$ from the new input distribution P by

(16). The flowchart of the multi-kernel DRR based on KRLS (MKDRR-KRLS) algorithm presented in this subsection is shown by the red arrows of Fig. 1.

B. Multi-Kernel Distribution to Function Regression

Because of the injective mapping of distribution embeddings and the kernel trick of the kernel $\mathcal{K}_{\mathcal{F}}$, we also have that $\mathcal{F}_{\mathcal{G}}(P_k, \mathbf{y}_{k,m})$ can be expressed in RKHS $\mathcal{H}_{\mathcal{G}}$ as follows,

$$\mathcal{F}_{\mathcal{G}}(P_k, \mathbf{y}_{k,m}) = \langle \boldsymbol{\omega}_{\mathcal{G}}, \mathcal{K}_{\mathcal{G}}(\boldsymbol{\nu}_{k,m}, \cdot)\rangle_{\mathcal{H}_{\mathcal{G}}}, \tag{19}$$

where the feature weight $\boldsymbol{\omega}_{\mathcal{G}} \in \mathcal{H}_{\mathcal{G}}$ represents the mapping $\mathcal{G}$. Similarly to the MKDRR-KRLS method presented in Section IV-A, the following cost function is minimized to learn the DFR defined in (6) from the data set $\mathcal{D}_{\mathcal{G}}$

$$J_{DFR} = \min_{\boldsymbol{\omega}_{\mathcal{G}}} \left[\sum_{k=1}^{T} \sum_{m=1}^{M} \left\| \mathbf{z}_{k,m} - \mathcal{F}_{\mathcal{G}}(P_k, \mathbf{y}_{k,m}) \right\|^2 + \lambda \left\| \boldsymbol{\omega}_{\mathcal{G}} \right\|^2_{\mathcal{H}_{\mathcal{G}}} \right]. \tag{20}$$

By introducing input feature matrices,

$$\boldsymbol{\Psi}_k = [\mathcal{K}_{\mathcal{G}}(\boldsymbol{\nu}_{k,1}, \cdot) \ \ldots \ \mathcal{K}_{\mathcal{G}}(\boldsymbol{\nu}_{k,M}, \cdot)]$$

and

$$\boldsymbol{\Upsilon}_k = [\boldsymbol{\Psi}_1 \ \ldots \ \boldsymbol{\Psi}_k]$$

and the desired matrices,

$$\mathbf{U}_k = [\mathbf{z}_{k,1} \ \ldots \ \mathbf{z}_{k,M}]^T$$

and

$$\mathbf{V}_k = \left[\mathbf{U}_k^T \ \ldots \ \mathbf{U}_k^T\right]^T$$

at the kth iteration, the feature weight can be approximated by,

$$\boldsymbol{\omega}_{\mathcal{G}} \approx \boldsymbol{\Upsilon}_k \left[\mathbf{K}_k + \lambda \mathbf{I}_{(kM)}\right]^{-1} \mathbf{V}_k = \boldsymbol{\Upsilon}_k \mathbf{Q}(k) \mathbf{V}_k \tag{21}$$

where $\mathbf{I}_{(kM)}$ is a $kM \times kM$ identity matrix, and $\mathbf{K}_k = \boldsymbol{\Upsilon}_k^T \boldsymbol{\Upsilon}_k$ is the Gram matrix containing all data available at the kth step.

From (21), the given distribution, and the input $\mathbf{y}$, the corresponding embedding can be obtained and the corresponding output, $\mathbf{z}$, calculated from (19), which implements the DFR. The flowchart of the multi-kernel DFR based on KRLS (MKDFR-KRLS) algorithm is shown by the blue arrow in Fig. 1. Similarly, the inverse matrix $\mathbf{Q}(k) = \left[\mathbf{K}(k) + \lambda \mathbf{I}_{(kM)}\right]^{-1}$ can also be calculated recursively, with computational complexity $O(k^2M^3)$ at every kth iteration of the algorithm.

Considering that the inputs $\{\boldsymbol{\nu}_{k,m}\}_{m=1}^{M}$ are obtained simultaneously at the kth step, we can develop a new KRLS algorithm based on matrix-block calculation in order to approximate the DFR online. Because the Gram matrix can be expressed by,

$$\mathbf{K}(k) = \left[\boldsymbol{\Psi}_i^T \boldsymbol{\Psi}_j\right]_{i,j=1:k}. \tag{22}$$

if we assume that $\mathbf{y}_{i,m} = \mathbf{y}_{j,m}$ for all i and j, and introduce the Gram matrices,

$$\begin{aligned} \mathbf{K}_Y &= [k_Y(\mathbf{y}_{i,m}, \mathbf{y}_{j,n})]_{m,n=1:M} \\ \mathbf{K}_{\boldsymbol{\mu}}(k) &= \left[\mathcal{K}_{\mathcal{F}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)\right]_{i,j=1:k} \end{aligned}$$

then $\mathbf{K}(k) = \mathbf{K}_{\boldsymbol{\mu}}(k) \otimes \mathbf{K}_Y$, where $\otimes$ denotes the Kronecker product. Because of this assumption, the matrix $\mathbf{K}_Y$ is the same at all iterations and since $\mathbf{K}^{-1}(k) = \mathbf{K}_{\boldsymbol{\mu}}^{-1}(k) \otimes \mathbf{K}_Y^{-1}$, the following approximation holds,

$$\mathbf{Q}(k) = \left[\mathbf{K}(k) + \lambda \mathbf{I}_{(kM)}\right]^{-1} \approx \mathbf{Q}_{\boldsymbol{\mu}}(k) \otimes \mathbf{Q}_Y \tag{23}$$

and matrices $\mathbf{Q}_{\boldsymbol{\mu}}(k) = \left(\mathbf{K}_{\boldsymbol{\mu}}(k) + \lambda \mathbf{I}_k\right)^{-1}$ and $\mathbf{Q}_Y = \left(\mathbf{K}_Y + \lambda \mathbf{I}_M\right)^{-1}$, can both be calculated recursively, with computational complexity $O(k^2)$ and $O(M^2)$, respectively at every kth iteration of the algorithm. Furthermore, under the assumption $\mathbf{y}_{i,m} = \mathbf{y}_{j,m}$, $\forall i, j$, the inverse matrix $\mathbf{Q}(k)$ can be decomposed to reduce the computational complexity from $O(k^2M^3)$ to $O(k^2)$ at every kth step.

V. Numerical Experiments

The MKDRR-KRLS and MKDFR-KRLS algorithms presented in this paper are demonstrated here by learning the control law and probability distribution for a network of autonomous agents, based on their observed positions. As shown in [18]–[20], the control law for a network of distributed agents can be obtained as a function of the agent distribution using an approach known as distributed optimal control. Consider a network of N agents are distributed randomly according to an initial distribution with support $\mathcal{W} \subset \mathbb{R}^2$, and let their position in $\mathcal{W}$ be described by a point in the inertial $\xi\eta$-frame. The agents are controlled such that they must reach a known target distribution denoted by P_0. The multi-kernel functional regression approach presented in this paper is utilized to learn the control mapping from the agent distribution to the divergence between the agent distribution and the goal distribution P_0. Subsequently, the approach is to learn the mapping from the agent distribution to the its cumulated density function (CDF) and partial derivatives with respect to ξ and η. Thus, the first problem is that of learning a mapping from a probability distributions to real, and the second problem is that of learning a mapping from a probability distribution to a function. The proposed MKDRR-KRLS and MKDFR-KRLS algorithms will be applied to learn these mappings.

Agent distributions are generated by means of 2D Mixture Gaussian distributions with two equivalently weighted components, denoted by P_k, $k = 1, 2, \ldots$. The means $[\mu_{\xi,i}, \mu_{\eta,i}]$, $i = 1, 2$, and covariance matrices $\boldsymbol{\Sigma}_i = \text{diag}([\sigma_{\xi,i}, \sigma_{\eta,i}])$, $i = 1, 2$ are selected randomly, where $\text{diag}(\cdot)$ denotes an operator that places a vector on the diagonal of a zero matrix of proper dimensions. The parameters $\mu_{\xi,i}$ and $\mu_{\eta,i}$, $i = 1, 2$, are generated from a uniform distribution on $[-1, 1]$ (km), and the parameters $\sigma_{\xi,i}$ and $\sigma_{\eta,i}$, $i = 1, 2$, are all generated from a uniform distribution on $[1, 1.2]$ (km). The goal agent location distribution is specified by setting these parameters by their expectations. For each mapping regression problems, we generate $N_{train} = \{100, 300, 500, 1000, 2000\}$ training data sets, $N_{valid} = 25$ validation data sets, and $N_{test} = 50$ test data sets. The goal distribution P_0 and some testing distributions P_k, $1 \leq k \leq 50$, generated by this setting are plotted in Fig. 2.

A. Experiment of Multi-Kernel Distribution to Real Regression based on KRLS

In this experiment, the Cauchy-Schwarz divergence is selected as the measure of divergence between the agent distribu-

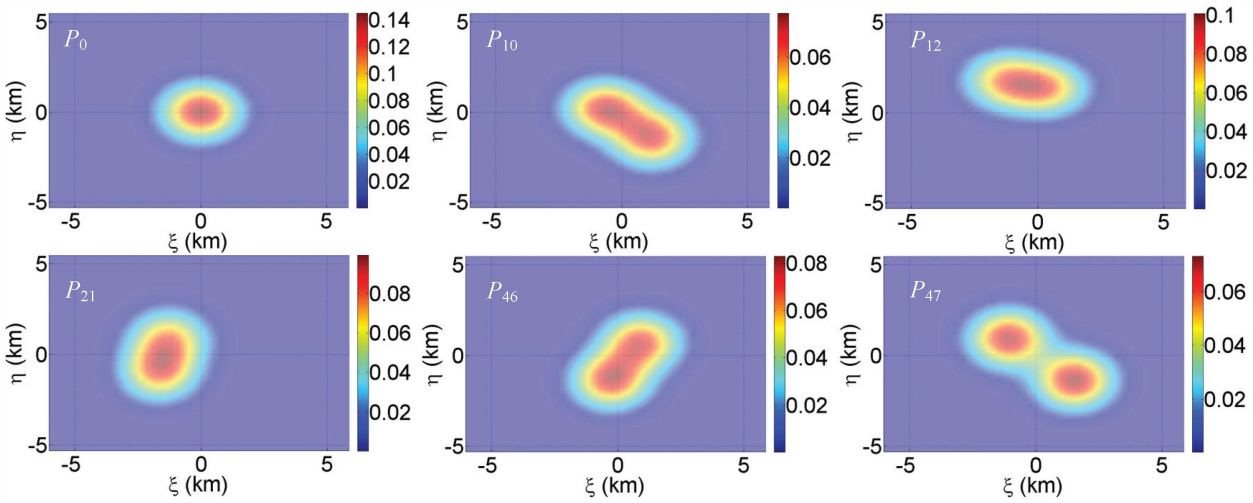

Fig. 2. Examples of goal distribution and testing distributions generated by Gaussian mixture in DR problem.

tion and the goal distribution because, for Gaussian mixtures, it can be represented in closed form [21]. The Cauchy-Schwarz divergence between two probability distributions P_k and P_0 is defined as,

$$D_{CS}(P_k||P_0) = -\log \frac{\iint p_k(\xi,\eta)p_0(\xi,\eta)d\xi d\eta}{\sqrt{\iint p_k^2(\xi,\eta)d\xi d\eta \iint p_0^2(\xi,\eta)d\xi d\eta}}, \tag{24}$$

where $p_k(\xi,\eta)$ and $p_0(\xi,\eta)$ are PDFs associated with distributions P_k and P_0. Given a new agent distribution P, the approximate Cauchy-Schwarz divergence, denoted by $\hat{D}_{CS}(P||P_0)$, can be calculated from the agent positions. Then, the MSE between the actual Cauchy-Schwarz divergence and the approximate divergence can be calculated to evaluate the regression performance.

In this experiment, for each agent distribution, a sample set $\mathcal{X}_k$ composed of $N_{sample} = 100$ samples of agent positions is generated. The regression performance of the MKDRR-KRLS algorithm is compared to that of the KKDRR algorithm proposed in [6]. For the KKDRR algorithm, 50×50 sample points on a uniformly distributed grid are used to estimate the divergence between two PDFs. The performance comparison results are plotted in Fig. 3. It can be seen that the MKDRR-KRLS algorithm outperforms the KKDRR algorithm because it requires a smaller number of training sets to learn the DR mapping with similar accuracy. Alternatively, when the same number of training sets, N_{train}, is used, MKDRR-KRLS displays a smaller MSE than the KKDRR algorithm.

B. Experiment of Multi-Kernel Distribution to Function Regression based on KRLS

The MKDFR-KRLS algorithm is evaluated by inferring the mapping from distributions to functions, using a 2D distribution $P_k(\xi,\eta)$ with PDF $p_k(\xi,\eta)$, and three different output functions, including the cumulative distribution function (CDF) $F_k(\xi,\eta) = \int_{-\infty}^{\xi}\int_{-\infty}^{\eta} p_k(\tilde{\xi},\tilde{\eta})d\tilde{\xi}d\tilde{\eta}$, and the gradient functions $g_\xi(\xi,\eta) = \frac{\partial}{\partial\xi}p_k(\xi,\eta)$ and $g_\eta(\xi,\eta) = \frac{\partial}{\partial\eta}p_k(\xi,\eta)$. In this experiment, $N_{train} = 500$ training sets are used. For each distribution, a sample set $\mathcal{X}_k$ with $N_{sample} = 500$ samples is

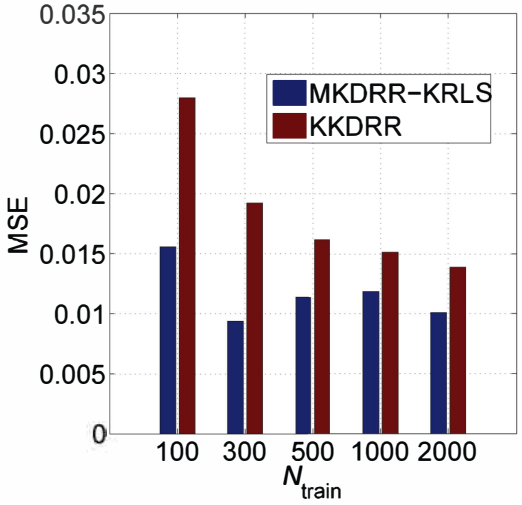

Fig. 3. Performance comparison between MKDRR-KRLS and KKDRR algorithms

generated. For each corresponding PDF p_k, $M = 50\times 50$ input samples $\mathbf{y}_{k,m} = [\xi_{k,m}, \eta_{k,m}]^T$, $m = 1,\ldots,M$, are generated on a uniformly distributed grid between the minimum and maximum values in the sample sets $\mathcal{X}_k$ in the 2D space, and the corresponding outputs $\mathbf{z}_{k,m}$ for three different output functions are calculated.

The DFR estimator is learned by applying the MKDFR-KRLS algorithm to the training data sets. The normalized mean square error (NMSE) between the actual function output $\mathbf{z}_{k,m}$ in the validation/testing data sets and the corresponding approximate output $\hat{\mathbf{z}}_{k,m}$ for each validation/testing distribution, defined as,

$$\text{NMSE} = \frac{\sum_{m=1}^{M}\|\mathbf{z}_{k,m} - \hat{\mathbf{z}}_{k,m}\|^2}{\sum_{m=1}^{M}\|\mathbf{z}_{k,m}\|^2} \tag{25}$$

is used to find the best parameter values. As an example, for the representative (testing) distribution $P_{46}(\xi,\eta)$ presented in Fig. 2 the regression performance of the DFR-KRLS algorithm is

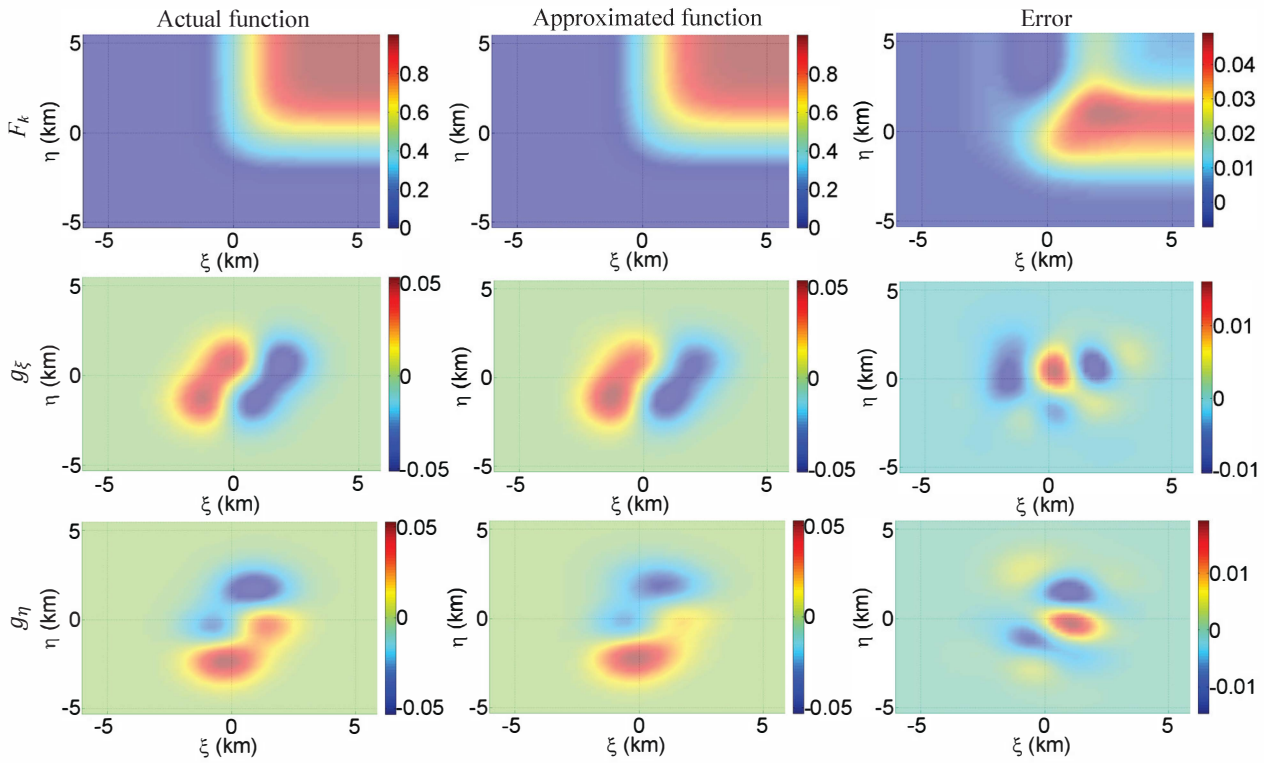

Fig. 4. Regression performance of MKDFR-KRLS, where figures in the first column present the output true functions, including F_k, g_ξ and g_η; the figures in the second column present the approximated output functions, including $\hat{F}_k$, $\hat{g}_\xi$ and $\hat{g}_\eta$; the figures in the last column present the differences between output functions and approximated output functions.

plotted in Fig. 4. A summary of results for all N_{train} training data sets are shown in Table I, in the form of (NMSE) "mean $\pm$ standard deviation".

TABLE I. REGRESSION PERFORMANCE RESULTS

Output functions	NMSE
CDF $F(\xi,\eta)$	0.0030 ± 0.0024
Gradient $g_\xi(\xi,\eta)$	0.0990 ± 0.0616
Gradient $g_\eta(\xi,\eta)$	0.0974 ± 0.0623

VI. CONCLUSION

This paper presents a new methodology for probability distribution regression using a multi-layer RKHS approach. The approach is demonstrated for the problem of distribution to real regression and distribution to function regression. To our knowledge, the proposed MKDFR is the first approach to deal with distribution to function regression. The distribution regressions can be implemented based on existing kernel regression algorithms in the multi-layer RKHS. KRLS is used as an example to demonstrate the MKDRR-KRLS and MKDFR-KRLS algorithms. These two proposed algorithms are demonstrated on synthetic data obtained by simulating a network of agents controlled by a distributed optimal control approach to reach a target distribution in a two-dimensional space. The MKDRR-KRLS and MKDFR-KRLS algorithms are successfully implemented to learn the agents control law from observations of their positions, as well as to learn functions of the agents distribution, such as the cumulative distribution function and the distribution gradients. These results show that the MKDRR-KRLS algorithm outperforms the recently proposed KKDRR algorithm and successfully performs all functional regression tasks.

ACKNOWLEDGMENT

This work was supported by NSF grant ECCS 1408022 and NFS grant DGE 1068871.

REFERENCES

[1] A. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[2] S. Haykin, *Neural Networks and Learning Machines (3rd Edition)*,. Prentice Hall, 2008.

[3] C. K. I. Williams and C. E. Rasmussen, "Gaussian Processes for Regression," *In Advances in Neural Information Processing Systerms*, Jan. 1996.

[4] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. John: Wiley, 2010.

[5] F. Ferraty and P. Vieu, *Nonparametric Functional Data Analysis: Theory and Practice*. Spriger, 2006.

[6] B. Pócaos, A. Rinaldo, A. Singh, and L. Wasserman, "Distribution-Free Distribution Regression," *AISTATS*, 2013.

[7] J. B. Oliva, B. Pócaos, and J. Schneider, "Distribution to distribution regression," *In International Conference on Machine Learning (ICML)*, pp. 1049–1057, 2013.

[8] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. Spriger, 2010.

[9] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[10] W. Liu, P. Pokharel, and J. C. Príncipe, "The Kernel Least Mean Square Algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, Feb. 2009.

[11] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized Kernel Least Mean Square Algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, Jan. 2012.

[12] ——, "Quantized Kernel Recursive Least Squares Algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 9, pp. 1484–1491, Sep. 2013.

[13] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert Space Embeddings of Conditional Distributions with Applications to Dynamical Systems," *In International Conference on Machine Learning (ICML)*, pp. 961–968, 2009.

[14] P. Zhu, B. Chen, and J. C. Príncipe, "Learning Nonlinear Generative Models of Time Series with a Kalman Filter in RKHS," *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 141–155, Jan. 2014.

[15] P. Zhu, *Kalman Filtering in Reproducing Kernel Hilbert Spaces*. Gainesivlle, FL, USA: PhD Thesis, Univeristy of Florida, 2013.

[16] A. Christmann and I. Steinwart, "Universal Kernels on Non-Standard Input Spaces ," *Advance in Neural Information Processing Systems*, vol. 23, pp. 406–414, 2010.

[17] K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf, "Learning from Distributions via Support Measure Machines," *Advance in Neural Information Processing Systems*, vol. 25, pp. 10–18, 2012.

[18] G. Foderaro and S. Ferrari, "Necessary conditions for optimality for a distributed optimal control problem," *Proc. of the IEEE Conference on Decision and Control (CDC)*, Dec. 2010.

[19] K. Rudd, G. Foderaro, and S. Ferrari, "A generalized reduced gradient method for the optimal control of multiscale dynamical systems," *Proc. of the IEEE Conference on Decision and Control (CDC)*, Dec. 2013.

[20] G. Foderaro, S. Ferrari, and T. A. Wettergren, "Distributed optimal control for multi-agent trajectory optimization," *Automatica*, vol. 50, pp. 149–154, Dec. 2014.

[21] K. Kampa, E. Hasanbelliu, and J. C. Príncipe, "Closed-form cauchy-schwarz pdf divergence for mixture of gaussians," *International Joint Conference on Neural Networks (IJCNN)*, pp. 2578–2585, 2011.