Connections Between Fuzzy Inference Systems and Kernel Machines

Jorge Guevara
IBM Research
Sao Paulo, Brazil
jorgegd@br.ibm.com

Jerry M. Mendel
University of Southern California
Los Angeles, California
mendel@sipi.usc.edu

R. Hirata Jr.
University of Sao Paulo
Sao Paulo, Brazil
hirata@ime.usp.br

***Abstract*— This paper explores the connection between fuzzy inference systems and kernel methods. Particularly, we explore the connection between the fuzzy-basis-function expansion of non-singleton fuzzy systems and regularized kernel methods with positive-definite kernels. We show that positive-definite kernel on fuzzy sets, i.e., a real-valued function defined on the set of fuzzy sets, is the core concept enabling such a connection. Furthermore, we rewrite the fuzzy-basis-function (FBF) expansion of non-singleton fuzzy systems in terms of a kernel-basis-function expansion with positive-definite kernels on fuzzy sets. The consequence of doing this is that fuzzy systems can be trained by regularized kernel methods, and the advantages of doing this are: 1) fuzzy systems can have more resistance to the curse of dimensionality, 2) they can explicitly regularize the function being approximated, 3) they can satisfy the Representer Theorem of kernel methods, which bounds the number of learned rules to the number of training observations, and, 4) they can learn functions in the functional space induced by the kernel on fuzzy sets. Consequently, from this perspective fuzzy systems can be regarded as kernel methods. Different from previous works, this connection relates the FBF-expansion directly to kernel methods without dropping the denominator of the expansion; also, thanks to the use of kernels on fuzzy sets, we provide an algorithm that makes it possible to drop the restriction of having fuzzy sets that share some of their antecedent parameters, e.g., Gaussian membership functions with the same variance in rule antecedents as is usually done in related works.**

Keywords*— *fuzzy systems, kernel methods, kernel on fuzzy sets, non-singleton fuzzy systems, positive-definite kernel.

I. Introduction

Fuzzy systems and kernel methods are two important fields of scientific research that have greatly impacted not only science but several industrial applications. *Fuzzy systems* are learning machines based on fuzzy logic inference. Their learning process can be done using numerical data (data-driven) and/or using knowledge in the form of linguistic IF-THEN rules (knowledge-driven) [12]. On the other hand, the *kernel method* [18], [20], [21] is an approach for machine learning based on two parts: 1) a mapping defined by a *kernel function*, that embeds the data into a feature space and; 2) an algorithm defined on this feature space. Machine learning models based on the kernel method are known as *kernel machines* or *kernel methods*.

Kernel methods can use several types of kernel functions, however, positive-definite kernels are the most used in practice. Those kernels can embed different types of data like strings, graphs, numerical data, distributions, images, etc., into a feature space called Reproducing Kernel Hilbert Space (RKHS). Kernel algorithms implicitly perform computations in a RKHS by using the widely-known "kernel trick" [18], [20], [21].

In this paper, we provide the first step to research the connections between fuzzy systems and kernel methods, so as to discover how they both can benefit from each other. Specifically, we explain how to express a fuzzy system as a kernel basis-function expansion, using positive-definite kernels and non-singleton fuzzy systems. These two-way connections are important in terms of the adoption of the main properties and advantages of both methods. A direct consequence of this connection is a new view of a class of fuzzy systems in terms of kernel machines.

A core concept in our work is the *positive-definite kernel on fuzzy sets*, which is a real-valued symmetric function defined on the set of fuzzy sets [11], [36]. This kind of kernel is important because it provides a direct connection between fuzzy systems and kernel methods. Not all kernels on fuzzy sets provide this kind of connection; however, we show how a positive-definite kernel on fuzzy sets can be naturally derived from the input-antecedent interactions in non-singleton fuzzy systems.

We conjecture the following benefits for both areas:

A. Benefits for Fuzzy Systems Using Kernel Methods

1) Resistance to the Curse of Dimensionality [46], [47]: Kernel methods based on positive-definite kernels are non-parametric consistent estimators that show good asymptotic convergence properties, i.e., they converge to a true function as the number of observations goes to infinity. Their rate of convergence depends only on the number of observations if the true underlying function belongs to the kernel space. In the worst-case scenario, the rate of convergence will have an exponential dependence on the number of dimensions of the data. In general, kernel methods show some kind of resistance to the curse of dimensionality when the true underlying function is smooth and lies between Lipschitz-continuous functions and functions in the induced kernel function space [42], [46], [47]. This kind of resistance is governed by a regularization parameter which deals with overfitting. To the best of our knowledge, the curse of dimensionality is a disadvantage of fuzzy systems since having a good approximation of an arbitrary function may mean that the number of rules has to grow exponentially with the number of dimensions of the data.

2) Function Approximation with Explicit Regularization: Regularization plays an important role in the resistance to the curse of dimensionality and overfitting. A fundamental characteristic of regularized kernel methods with positive-definite kernels is the explicit regularization of the function that is being approximated; this is stated in terms of a kernel matrix estimated from the data and the kernel function, and provides a clear interpretation of the regularization of the model.

3) Training Fuzzy Systems with Regularized Kernel Methods: Connecting fuzzy systems and kernel methods can allow the training of fuzzy systems using regularized kernel algorithms, and the use of those fuzzy systems for solving machine learning tasks. For example, we conjecture that: a) kernel PCA may be used to train a fuzzy system for novelty-detection and image de-noising; b) kernelized Lasso Machine may define a fuzzy system able to perform variable selection; c) Support Vector Machine may define a fuzzy system operating as a maximal margin classifier; d) Support Vector Data Description may define a fuzzy system working as a minimum enclosing ball; etc.

4) Representer Theorem for Fuzzy Systems: The Representer Theorem of kernel methods [20] can provide a bound on the number of rules learned from data by a fuzzy system, limiting the number of those rules to be at most the number of observations.

5) Learning Fuzzy Systems in Reproducing Kernel Hilbert Spaces (RKHS): Fuzzy systems can learn or approximate functions within a RKHS, which is a proper Hilbert space of functions, with the property that the closeness of two functions, in the sense of a norm, means closeness on the image values of those functions. This provides fuzzy systems additional mathematical tools for learning on those spaces.

6) Design of Fuzzy Systems for Structured Data: Positive-definite kernels can be defined on structured data like graphs or strings. We conjecture that the connection between fuzzy systems and kernel methods can leverage further research on this topic.

B. Benefits for Kernel Methods using Fuzzy Systems

1) Priors on the Structure of the Model: We conjeture that kernel methods will be able to impose priors on the structure of the learned function by using prior-knowledge encoded in the form of linguistic IF-THEN rules. To the best of our knowledge, the fuzzy-basis-function expansion of a fuzzy system is the only basis function expansion that incorporates both linguistic and numerical information for function approximation.

2) Applying Kernel Machines on Fuzzy Data: We show in Section IV that fuzzy systems naturally induce some class of positive-definite kernels on fuzzy sets. We conjecture that this will allows kernel machines to deal with fuzzy sets as inputs via the kernel on fuzzy sets thereby leveraging machine learning applications in datasets containing point-wise uncertainty (ontic and epistemic uncertainty).

3) Explainability of Black-Box Kernel Methods: It is widely known that in some applications fuzzy systems are highly interpretable thanks to the use of linguistic IF-THEN rules. Kernel methods can benefit from this property of fuzzy systems.

4) Noise-Resistant Kernel Machines: Non-singleton fuzzy systems, which are explained in Section III, can provide a pre-filtering of the inputs to a kernel machine via a non-singleton fuzzification operation. Consequently, we conjecture that the resulting non-singleton kernels can be used for creating noise-resistant kernel machines.

Remark about the word kernel. The word *kernel* is widely used by the fuzzy set community to denote the crisp subset of a fuzzy set where all the elements are ones. In machine learning the word *kernel* is used to denote statistical models based on non-negative kernels, e.g., kernel smoothing (Nadaraya–Watson kernel regression) and Kernel density estimator. In this paper, we deal strictly with positive-definite kernels, consequently:

Kernel Machines $\stackrel{\text{def}}{=}$ *regularized machine learning models with positive-definite kernels*

Related work

Previous works apply support vector machines (SVMs) (which is a kernel machine) for learning singleton fuzzy systems [33] and [34] (and references therein). However, those works link SVMs directly to fuzzy systems by using a simplified version of the fuzzy-basis-function expansion of fuzzy systems. They ignore/remove the denominator of the fuzzy-basis-function expansion. The main consequence of doing this is that they transform a non-linear representation of a FBF to a linear one (linear in the basis functions). See for example a related discussion in the context of RBF-networks and fuzzy systems in [29]. In the present paper, we provide a connection not only between SVMs but also between kernel methods in general and non-singleton fuzzy systems, without dropping the denominator from their fuzzy-basis-function expansion.

On the other hand [35] and references therein introduce fuzzy numbers into SVMs algorithms. The main disadvantage of this approach is that one has to then modify the kernel algorithm. A more convenient way to do this in terms of kernels on fuzzy sets without modifying the kernel algorithm is described in [11], [36], [37].

II. Kernel Machines

Kernel machines are a class of algorithms used for statistical machine learning. They have two components: *a positive-definite kernel function* that maps the input data into a RKHS (a.k.a. kernel feature space), and a *kernel algorithm* defined in the RKHS that performs the analysis of the mapped data. The RKHS is a space that can be infinite-dimensional, however, thanks to the Reproducing property of positive-definite kernels, inner products in a RKSH are equivalent to kernel evaluations in the input space. Moreover, kernel methods are *modular*, i.e., it is always possible to change the kernel function without changing the algorithm. Popular examples of kernel machines are Support Vector Machines, Kernel PCA, Support Vector Data Description, etc., [14], [18], [20], [21].

A. Positive-definite Kernels and Reproducing Kernels

Positive-definite kernels play an important role in fields such as analysis [14], [28], approximation theory [15], engineering [16], geostatistics [17], probability theory and statistics [18], [19], among others. Although kernel functions can be complex-valued or operator-valued, the most common kernels in machine learning are real-valued functions of the form: $k: \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \rightarrow \mathcal{R}$, where $\boldsymbol{\mathcal{X}}$ is a non-empty set. Note that there is no topological restriction about the input space $\boldsymbol{\mathcal{X}}$, consequently $\boldsymbol{\mathcal{X}}$ can be any set, e.g., the set of real numbers, multi-dimensional vectors, probability distributions, fuzzy sets, strings, graphs, etc.

Herein, $\boldsymbol{\mathcal{X}}$ denotes a non-empty set, $\mathcal{H}$ a real RKHS of real-valued functions on $\boldsymbol{\mathcal{X}}$. Notation $k(\cdot, \boldsymbol{x})$ means the mapping $\boldsymbol{x}' \rightarrow k(\boldsymbol{x}', \boldsymbol{x})$ with fixed $\boldsymbol{x}$ where k is a function on $\boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}}$.

Definition 1 [Reproducing kernel] A real-valued function k on $\boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}}$ is a Reproducing kernel of a Hilbert space $\mathcal{H}$ if [18]:

1) $\forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}},\ k(\cdot, \boldsymbol{x}) \in \mathcal{H}$

2) $\forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}, \forall h \in \mathcal{H}, h(\boldsymbol{x}) = \langle h, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$,

Requirement (2) is called the *Reproducing Property*. Thus:

$$\forall (\boldsymbol{x}, \boldsymbol{x}') \in \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}},\ k(\boldsymbol{x}, \boldsymbol{x}') = \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}') \rangle_{\mathcal{H}} \quad (1)$$

Definition 2 [RKHS] A real Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ is a Hilbert Space of real-valued functions on $\boldsymbol{\mathcal{X}}$ with Reproducing kernel [18].

RKHS are interesting for statistical Machine Learning because: 1) if two functions are close in the sense of a norm then their function images are also close (i.e., a sequence converging in the norm also converges pointwise); 2) it is possible to implicitly perform inner products in $\mathcal{H}$ as $k(\boldsymbol{x}, \boldsymbol{x}') = \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}') \rangle_{\mathcal{H}}$; and 3) the evaluation of the estimated function h on a point $\boldsymbol{x}'$ is interpreted as $h(\boldsymbol{x}) = \langle h, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$, this leverage: a) the design of geometrical algorithms (e.g. Support Vector Machines, Minimum Enclosing Balls), b) the extension of widely-known methods (Kernel PCA, Kernel Ridge Regression) and 3) the design of regularizers that explicitly penalize the function that is being approximated.

The equivalence between Reproducing Kernels and positive-definite kernels is given as follows:

Theorem 1 [RKHS] Any Reproducing kernel , $k: \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \rightarrow \mathcal{R}$ is a symmetric positive-definite kernel, that is, it satisfies:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \quad (2)$$

for all positive integer numbers N, all $\alpha_i \alpha_j \in \mathcal{R}$ and, all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{\mathcal{X}}$. The converse is also true.

Examples of positive-definite kernels $k(\boldsymbol{x}, \boldsymbol{x}')$ are: the linear kernel $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}'$, the polynomial kernel $(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}' + 1)^{\gamma}, \gamma > 0$, the Gaussian kernel $\exp(\gamma||\boldsymbol{x} - \boldsymbol{x}'||^2)$, $\gamma > 0$, kernels on measures [38], [39], kernels on strings [40], kernels on graphs [41], kernels on fuzzy sets [11], among others [20], [21].

B. Kernel Algorithms and the Representer Theorem

Given a dataset:

$$\mathcal{D} \equiv \{(\boldsymbol{x}_1, y_1), \ldots (\boldsymbol{x}_N, y_N)\} \subset \boldsymbol{\mathcal{X}} \times \mathcal{Y}, \quad (3)$$

where $\boldsymbol{\mathcal{X}} \equiv \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_D$, and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iD})^T \in \boldsymbol{\mathcal{X}}$ for $i = 1, \ldots N$; kernel algorithms seek to solve the following optimization problem:

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}}\ \Omega(\lambda, ||h||_{\mathcal{H}}) + \frac{1}{N}\sum_{i=1}^{N} L(y_i, h(\boldsymbol{x}_i)), \quad (4)$$

where $\mathcal{H}$ is a RKHS, λ is a non-negative regularization parameter, Ω is a regularization function and L is a loss function.

1) Representer Theorem [20]*:* The Representer Theorem characterizes the solution of (4) in terms of the data sample $\mathcal{D}$, a symmetric positive-definite kernel k, an empirical loss L and a regularizer Ω. It states that any minimizer h^* of (4) has the form:

$$h^*(\boldsymbol{x}') = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}', \boldsymbol{x}_i) \quad (5)$$

where $\alpha_i \in \mathcal{R}$.

2) Regularization: The regularizer Ω, explicitely penalizes the function that is being approximated, e.g. $\Omega(\lambda, ||h||_{\mathcal{H}}) = \lambda||h||_2^2 = \lambda\langle h, h \rangle_{\mathcal{H}} \equiv \boldsymbol{\alpha}^T K \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^T$ and K is the kernel matrix, i.e., $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

III. FUZZY SETS AND SYSTEMS

Many fuzzy systems are universal approximators of functions [1], and have been used successfully in robotics, control, dynamic systems, machine learning, deep learning, intelligent systems [2], [3], [4], [5], [6], etc. Fuzzy systems approximate a function $h: \boldsymbol{\mathcal{X}} \rightarrow \mathcal{Y}$ by applying operations from fuzzy set theory and fuzzy logic. At the heart of fuzzy systems is the concept of fuzzy set, originally defined by Zadeh as a mapping $\boldsymbol{X}: \boldsymbol{\mathcal{X}} \rightarrow [0,1]$, where $\boldsymbol{X}(\boldsymbol{x})$ is interpreted as the membership degree of element $\boldsymbol{x}$ in the fuzzy set $\boldsymbol{X}$ and is also called membership function that plays an analogous role to the characteristic function in classic set theory [7]. Fuzzy sets are used in fuzzy logic to extend the true values of classic Boolean logic to a [0,1]-valued logic, and, according to Dubois and Prade [13], they have three semantics: 1) *similarity* between a value and a prototype, 2) *degree of uncertainty* about an ill-known value, and 3) *preference*. Generalizations of fuzzy sets include the $\mathcal{L}$-fuzzy set, which is a mapping $\boldsymbol{X}: \boldsymbol{\mathcal{X}} \rightarrow \mathcal{L}$, where $\mathcal{L}$ is a complete lattice [8]; type-2 fuzzy sets which are mappings of the form $\boldsymbol{X}: \boldsymbol{\mathcal{X}} \rightarrow \mathcal{F}(\boldsymbol{\mathcal{X}})$ where $\mathcal{F}(\boldsymbol{\mathcal{X}})$ is the set of all the fuzzy sets on $\boldsymbol{\mathcal{X}}$, etc. For an account of different kinds of fuzzy sets and their generalizations, see [10].

A. Mamdani Fuzzy Systems

In Mamdani fuzzy systems [25] the approximation of a function $h: \boldsymbol{\mathcal{X}} \rightarrow \mathcal{Y}$ is carried out by applying a sequence of fuzzy logic operations to the input $\boldsymbol{x}' \in \boldsymbol{\mathcal{X}}$, where $\boldsymbol{x}' = (x_1', \ldots, x_D')^{\mathrm{T}} \in \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_D \equiv \boldsymbol{\mathcal{X}}$. Given a fuzzy rule-base, the sequence of fuzzy logic operations involves the application of a fuzzification operator on the input $\boldsymbol{x}'$, a fuzzy-inference algorithm applied to the output of the fuzzification operator and the rule base, and a defuzzification operator that transforms the fuzzy-inference algorithm's output into the $\mathcal{Y}$ space.

The fuzzification operator, denoted by F, transforms the input $\boldsymbol{x}'$ into D fuzzy sets. We denote this transformation by $X'_d(\cdot\,|x'_d) = F(x'_d)$, $d = 1,\dots D$, where the notation $X'_d(\cdot\,|x'_d)$ means a mapping $x_d \rightarrow X'_d(x_d|x'_d)$ with fixed x'_d. Note that this is because $X'_d(\cdot\,|x'_d)$ is a fuzzy set constructed around x'_d. The next step is to propagate those X'_d, $d = 1,2,\dots D$, fuzzy sets through a rule-base of M rules whose elements are fuzzy-logic rules of the form $R_l \equiv \boldsymbol{A}_l \Rightarrow C_l$, ($l = 1,2,\dots,M$) where $\boldsymbol{A}_l$ represent the antecedent part of a fuzzy rule (which is composed by D fuzzy sets: $A_{l1}, A_{l2},\dots, A_{lD}$), symbol $\Rightarrow$ denotes a fuzzy implication operator, and $C_l : \mathcal{Y} \rightarrow [0,1]$ is a fuzzy set representing the consequent part of a fuzzy rule. In Mamdani fuzzy systems, the input fuzzy sets X'_d, $d = 1,2,\dots D$ are aggregated to construct the input fuzzy set $\boldsymbol{X}'(\cdot\,|\boldsymbol{x}') : \boldsymbol{\mathcal{X}} \rightarrow [0,1]$ with:

$$\boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') = T_{d=1}^{D} X'_d(x_d|x'_d), \tag{6}$$

where T (and $\star$ in what follows) denotes the T-norm operator. Moreover, rule R_l is the fuzzy relation $R_l : \boldsymbol{\mathcal{X}} \times \mathcal{Y} \rightarrow [0,1]$ with

$$R_l(\boldsymbol{x}, y) = T_{d=1}^{D} A_{ld}(x_d) \star C_l(y) \tag{7}$$

Also, the antecedent part $\boldsymbol{A}_l$ is a fuzzy set $\boldsymbol{A}_l : \boldsymbol{\mathcal{X}} \rightarrow [0,1]$ with:

$$\boldsymbol{A}_l(\boldsymbol{x}) = T_{d=1}^{D} A_{ld}(x_d) \tag{8}$$

In fact, this way of modeling the fuzzy implication operator is the principal characteristic of Mamdani fuzzy systems.

The propagation of the input $\boldsymbol{X}'(\cdot\,|\boldsymbol{x}')$ through the rule-base is done by a fuzzy-inference algorithm. A common approach is to use the sup-t-norm composition as the compositional rule of inference between the input $\boldsymbol{X}'(\cdot\,|\boldsymbol{x}')$ and a fuzzy relation R_l from the fuzzy rule base. We denote this operation by:

$$\boldsymbol{X}'(\cdot\,|\boldsymbol{x}')\ o\ R_l \equiv \sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star R_l(\boldsymbol{x}, y) \tag{9}$$

Observe that ([12] and [22]) the supremum in (9) is taken only for all $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$. It follows, by replacing the definition of the fuzzy relation R_l in (9), that:

$$\boldsymbol{X}'(\cdot\,|\boldsymbol{x}')\ o\ R_l \equiv C_l \star \sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x}) \tag{10}$$

Thus, the output of a Mamdani fuzzy system is given by:

$$h(\boldsymbol{x}') = D_{fuzz}\left(\left\{\left(\sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x}) \star C_l\right)\right\}_{l=1}^{M}\right) \tag{11}$$

In (11) D_{fuzz} is a defuzzification procedure that transforms the set in (11) into an element of $\mathcal{Y}$. We refer to [12] and [24] for definitions of defuzzifiers.

B. Fuzzy-Basis-Function Expansion of Mamdani Fuzzy Systems

There are different kinds of Mamdani fuzzy systems, depending on what is done with the outputs of the inference engine. Here we focus on processing these outputs only using a defuzzification operator that leads to a basis function expansion. An important result in fuzzy systems is that Mamdani fuzzy systems are universal approximators of functions that can be expressed as the following fuzzy-basis-function (FBF) expansion [12], [23], [43]:

$$h(\boldsymbol{x}') = \sum_{l=1}^{M} \alpha_l \phi_l(\boldsymbol{x}'), \tag{12}$$

where ϕ_l functions are called *fuzzy basis functions*, $\alpha_l \in \mathcal{R}$, and:

$$\phi_l(\boldsymbol{x}') = \frac{\sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x})}{\sum_{l=1}^{M} \sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x})} \tag{13}$$

One of the main advantages of the FBF expansion is that it is possible to decompose the right-hand of (12) into a sum of two terms, one associated with FBFs that are associated with rules that come from numerical data and the other associated with rules that come from linguistic information, i.e.,

$$h(\boldsymbol{x}') = h_N(\boldsymbol{x}') + h_L(\boldsymbol{x}') \tag{14}$$

This capability to include both numerical data and linguistic data for function approximation is a unique characteristic of this FBF expansion of fuzzy systems.

C. Singleton and Non-singleton Mamdani Fuzzy Systems

Popular instances of Mamdani fuzzy systems are the singleton and non-singleton Mamdani fuzzy systems. Note that singleton fuzzy systems are a special case of non-singleton fuzzy systems, i.e. $\sup_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x}) = \boldsymbol{A}_l(\boldsymbol{x}')$, (see [12] for a detailed derivation). For example, the FBF for singleton fuzzification and product T-norm are ($l = 1,2,\dots,M$):

$$\phi_l(\boldsymbol{x}') = \frac{\prod_{d=1}^{D} A_{ld}(x'_d)}{\sum_{l=1}^{M} \prod_{d=1}^{D} A_{ld}(x'_d)} \tag{15}$$

Similarly, FBF for non-singleton fuzzification and product T-norm are:

$$\phi_l(\boldsymbol{x}') = \frac{\prod_{d=1}^{D} \sup_{x_d\in X_d} X'_d(x_d|x'_d) * A_{ld}(x_d)}{\sum_{l=1}^{M} \prod_{d=1}^{D} \sup_{x_d\in X_d} X'_d(x_d|x'_d) * A_{ld}(x_d)} \tag{16}$$

An example of (16) when all the fuzzy sets are Gaussians and the product T-norm is [22], [12]:

$$\phi_l(\boldsymbol{x}') = \frac{\prod_{d=1}^{D} exp\left(-0.5\,\frac{(x'_d - m_{ld})^2}{\sigma_d^2 + \sigma_{ld}^{\ 2}}\right)}{\sum_{l=1}^{M} \prod_{d=1}^{D} exp\left(-0.5\,\frac{(x'_d - m_{ld})^2}{\sigma_d^2 + \sigma_{ld}^{\ 2}}\right)} \tag{17}$$

where x'_d, σ_d^2 are the parameters of the input Gaussian fuzzy set X'_d and m_{ld}, σ_{ld} are the parameters of the Gaussian fuzzy set A_{ld} of the antecedent part of the rule l. Non-singleton fuzzy systems are particularly interesting because they have the important property of prefiltering the inputs, leading to noise-resistance fuzzy systems.

IV. Kernel Basis Function Expansion of Non-singleton Mamdani Fuzzy System

A. Non-singleton Kernel on Fuzzy Sets

This section shows that the input-antecedent interaction of the rules of non-singleton Mamdani fuzzy systems,

$\sup_{x \in \mathcal{X}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x})$, define a positive-definite kernel on fuzzy sets.

Definition 3 [Nonsingleton Kernel on Fuzzy Sets]. Let $\boldsymbol{X}$ and $\boldsymbol{X}'$ be two fuzzy sets defined on $\mathcal{X}$. Then the nonsingleton kernel on fuzzy sets is a mapping $k: \mathcal{F}(\mathcal{X}) \times \mathcal{F}(\mathcal{X}) \to [0,1]$ defined by:

$$k(\boldsymbol{X}, \boldsymbol{X}') = \sup_{x \in \mathcal{X}} \boldsymbol{X}(\boldsymbol{x}) \star \boldsymbol{X}'(\boldsymbol{x}) \quad (18)$$

Such a kernel was defined in [11],[27] and some of its properties are:

1) Is a positive-definite kernel: As a consequence, it is a Reproducing kernel defining a RKHS $\mathcal{H}$. Thus $k(\boldsymbol{X}, \boldsymbol{X}')$ implicitly induces an inner product $\langle k(\cdot, \boldsymbol{X}), k(\cdot, \boldsymbol{X}')\rangle_{\mathcal{H}}$ in $\mathcal{H}$. Notice that $k(\cdot, \boldsymbol{X}): \mathcal{F}(\mathcal{X}) \to \mathcal{R}$ and $k(\cdot, \boldsymbol{X}'): \mathcal{F}(\mathcal{X}) \to \mathcal{R}$ are representative functions of fuzzy sets $\boldsymbol{X}, \boldsymbol{X}'$, respectively and that the implicit inner product of the kernel evaluation allows one to manipulate fuzzy sets in a geometrical way (the kernel trick). This has a substantial impact on the analysis of fuzzy data, because the space of fuzzy sets is not a vector space. Consequently, it is straigthforward to define a metric between fuzzy sets, induced by the kernel as:

$$\begin{aligned} d(\boldsymbol{X}, \boldsymbol{X}')^2 &= ||k(\cdot, \boldsymbol{X}) - k(\cdot, \boldsymbol{X}')||_H^2 \\ &= k(\boldsymbol{X}, \boldsymbol{X}) - 2\, k(\boldsymbol{X}, \boldsymbol{X}') + k(\boldsymbol{X}', \boldsymbol{X}'). \end{aligned} \quad (19)$$

The metric in (19) can be used to estimate distances between fuzzy sets and to define new positive-definite kernels as in [37], i.e., $k_d(\boldsymbol{X}, \boldsymbol{X}') = exp(-\gamma\, d(\boldsymbol{X}, \boldsymbol{X}')^2)$, $\gamma > 0$.

2) Is a type of intersection kernel on fuzzy sets: Because T-norm operators implement intersection on fuzzy sets, we can argue that the non-singleton kernel can be written more generally as:

$$k(\boldsymbol{X}, \boldsymbol{X}') = \sup_{x \in \mathcal{X}} \boldsymbol{X} \cap \boldsymbol{X}'(\boldsymbol{x}) \quad (20)$$

3) Is a non-negative kernel on fuzzy sets. That kernel is a [0,1]-valued kernel, consequently, is a non-negative kernel. We argue that this is important because those kernels can help to design non-parametric estimators on fuzzy datasets.

B. Kernel Matrix

In practice, kernel methods use positive-definite kernels to construct kernel matrices. To construct a kernel matrix from the non-singleton kernel on fuzzy sets we introduce the concept of fuzzy dataset, as follows: Given a dataset $\mathcal{D}$, as in (**3**), a *fuzzy dataset* is the set containing fuzzy sets modeling each observation $\boldsymbol{x}_i$ in $\mathcal{D}$. To do this, we propose to transform each $\boldsymbol{x}_i = (x_{i1}, \dots x_{iD})^T$ into the fuzzy sets: $X_{id}(\cdot\, |x_{id}) = F(x_{id})$, $d = 1, \dots D; i = 1, \dots, N$; where F is a non-singleton fuzzifier; and to use those fuzzy sets for constructing the following *fuzzy dataset*:

$$\mathcal{D}_{\mathcal{F}} = \{(\boldsymbol{X}_1(\cdot\, |\boldsymbol{x}_1), y_1), \dots (\boldsymbol{X}_N(\cdot\, |\boldsymbol{x}_N), y_N)\}, \quad (21)$$

where each $\boldsymbol{X}_i(.\, |\boldsymbol{x}_i), i = 1, \dots, N$ is a fuzzy set given by (6).

The kernel matrix, which is positive-semidefinite, for the non-singleton kernel is:

$$\begin{aligned} K_{ij} &\equiv k\left(\boldsymbol{X}_i(\cdot\, |\boldsymbol{x}_i), \boldsymbol{X}_j(\cdot\, |\boldsymbol{x}_j)\right) \\ &= \sup_{x \in \mathcal{X}} \boldsymbol{X}_i(\boldsymbol{x}|\boldsymbol{x}_i) \star \boldsymbol{X}_j(\boldsymbol{x}|\boldsymbol{x}_j) \end{aligned} \quad (22)$$

where $i, j = 1, \dots, N$.

For example, when measurements x_{id} are modeled by Gaussian fuzzy membership with parameters x_{id}, σ_{id}, $d = 1, \dots D; i = 1, \dots, N$; then the kernel matrix in (22) is:

$$\begin{aligned} K_{ij} &= k\left(\boldsymbol{X}_i(\cdot\, |\boldsymbol{x}_i), \boldsymbol{X}_j(\cdot\, |\boldsymbol{x}_j)\right) \\ &= exp\left(-0.5 \sum_{d=1}^{D} \frac{\left(x_{id} - x_{jd}\right)^2}{\sigma_{id}^2 + \sigma_{jd}^2}\right) \end{aligned} \quad (23)$$

Finally, kernels in kernel methods use parameters to dictate up to what extent two observations are similar. This parameter is usually tuned via cross-validation or Maximum Likelihood estimation. This leads to the following version of the non-singleton Gaussian kernel with kernel parameter $\gamma > 0$:

$$\begin{aligned} K_{ij} &= k\left(\boldsymbol{X}_i(\cdot\, |\boldsymbol{x}_i), \boldsymbol{X}_j(\cdot\, |\boldsymbol{x}_j)\right) \\ &= exp\left(-0.5\, \gamma \sum_{d=1}^{D} \frac{\left(x_{id} - x_{jd}\right)^2}{\sigma_{id}^2 + \sigma_{jd}^2}\right) \end{aligned} \quad (24)$$

C. Kernel Basis Function Expansion

In this section we express the FBF expansion for Mamdani fuzzy systems in terms of non-singleton kernels on fuzzy sets, and call this basis expansion the *kernel basis function (KBF) expansion* of Mamdani fuzzy systems. By definition of the non-singleton kernel on fuzzy sets (Definition 3) and the FBF of Mamdani fuzzy sets ((12) and (13)), we have:

Definition 4 [KBF expansion of Mamdani Fuzzy Systems]. Let k be a non-singleton kernel on fuzzy sets as in Definition 3. A *KBF expansion* for Mamdani fuzzy systems, is given by:

$$h(\boldsymbol{x}') = \frac{\sum_{l=1}^{M} \alpha_l k(\boldsymbol{X}'(\cdot\, |\boldsymbol{x}'), \boldsymbol{A}_l)}{\sum_{l=1}^{M} k(\boldsymbol{X}'(\cdot\, |\boldsymbol{x}'), \boldsymbol{A}_l)} \quad (25)$$

Note that the arguments for the non-singleton kernel on fuzzy sets given by (18) are two fuzzy sets: the input $\boldsymbol{X}'(\,.\,|\boldsymbol{x}')$ and the fuzzy set for the antecedent of rule l: $\boldsymbol{A}_l$. Thus, (18) can be rewritten as:

$$k(\boldsymbol{X}'(\cdot\, |\boldsymbol{x}'), \boldsymbol{A}_l) = \sup_{x \in \mathcal{X}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x}) \quad (26)$$

Using this fact, we can write (25) as.:

$$h(\boldsymbol{x}') = \frac{\sum_{l=1}^{M} \alpha_l \sup_{x \in \mathcal{X}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x})}{\sum_{l=1}^{M} \sup_{x \in \mathcal{X}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x})} \quad (27)$$

Equation (25) can be expressed as in (12) and (13) which connects the KBF and FBF expansion for a Mamdani fuzzy system

D. Connection to Nadaraya-Watson Kernel estimator

The Nadaraya-Watson Kernel estimator is a non-parametric statistical model that estimates a regression function from a dataset $\{(x_1, y_1), \dots (x_N, y_N)\} \subset \mathcal{R}^D \times \mathcal{R}$ as:

$$h_{NW}(\boldsymbol{x}') = \frac{\sum_{l=1}^{N} y_i k_h(\boldsymbol{x}' - \boldsymbol{x}_i)}{\sum_{l=1}^{N} k_h(\boldsymbol{x}' - \boldsymbol{x}_i)}, \tag{28}$$

where k_h is a non-negative kernel, i.e. a function such that $\forall x, x' \in \mathcal{R}^D;\ k_h(x, x') \geq 0$. Note that the FBF expansion of a singleton Mamdani fuzzy system is directly related to this statistical estimator, when singleton fuzzification is used $\sup_{x \in \mathcal{X}} \boldsymbol{X}'(\boldsymbol{x}|\boldsymbol{x}') \star \boldsymbol{A}_l(\boldsymbol{x}) = \boldsymbol{A}_l(\boldsymbol{x}')$, and, substituting this into (12) and (13) we obtain:

$$h_s(\boldsymbol{x}') = \frac{\sum_{l=1}^{M} \alpha_l \boldsymbol{A}_l(\boldsymbol{x}')}{\sum_{l=1}^{M} \boldsymbol{A}_l(\boldsymbol{x}')} \tag{29}$$

Defining $M = N, y = \alpha$ and $\boldsymbol{A}_l(\boldsymbol{x}') \equiv \boldsymbol{A}_l(\boldsymbol{x}'|\boldsymbol{x}_i) = k_h(\boldsymbol{x}' - \boldsymbol{x}_i)$, we obtain an equivalence between h_s and h_{NW} (reference [33] explores $\boldsymbol{A}_l(\boldsymbol{x}'|\boldsymbol{x}_i) = k_h(\boldsymbol{x}' - \boldsymbol{x}_i)$ in the context of support vector learning). However, it turns out that our formulation of KBF expansion for Mamdani fuzzy systems in Definition 4 is more general in the sense that it contains the Nadaraya-Watson Kernel estimator as a special case, because of the use of the kernel on fuzzy sets. We conjecture that a non-singleton kernel on fuzzy sets can extend this statistical estimator for working with fuzzy datasets. This can permit prefiltering the inputs due to the non-singleton fuzzification used. We leave details about this topic for further research

E. Connection to RBF Neural Networks

As discussed in [29], there is no direct connection of FBF expansions of fuzzy systems to non-normalized Radial-Basis-Function networks. i.e., models of the form $h_{RBF}(\boldsymbol{x}') = \sum_i \alpha_i \phi(||\boldsymbol{x}' - \boldsymbol{x}_i||)$, where ϕ are radial basis functions. However, there is a functional equivalence between normalized Radial-Basis-Function networks and a set of restricted fuzzy inference systems, as is explained in [30]. [31]. [32]. We hypothesize that, thanks to the definition of a non-singleton kernel on fuzzy sets given herein, it is possible to drop the third restriction in [30] that: *"The membership functions within each rule are chosen as Gaussian functions with the same variance"*, because the kernel on non-fuzzy sets can allow passing the variances related to individual Gaussian membership functions directly into the arguments of the non-singleton kernel, as is evident in (23).

V. Learning Fuzzy Systems with Kernel Machines

In this section, we explain how it is possible to learn the structure of non-singleton Mamdani fuzzy systems using kernel methods. Note that, we are not going to show any kind of "equivalence" between kernel methods and fuzzy inference systems. Instead, we use the KBF expansion for relating kernel methods and fuzzy systems.

Definition 5 [KBF expansion as kernel].
Let k be the non-singleton kernel on fuzzy sets as in Definition 3. Let $\boldsymbol{X}_i,\ i = 1,2 \dots N$ be fuzzy sets defined on $\boldsymbol{\mathcal{X}}$, and, set $\mathcal{L} \equiv \{X_i \mid i = 1,2 \dots N\}$. A kernel on fuzzy sets according to the *KBF expansion* for Mamdani fuzzy systems, is:

$$k_{\mathcal{L}}(\boldsymbol{X}, \boldsymbol{X}_i) = \frac{k(\boldsymbol{X}, \boldsymbol{X}_i)}{\sum_{i=1}^{N} k(\boldsymbol{X}, \boldsymbol{X}_i)}, \tag{30}$$

Observe that $k_{\mathcal{L}}$ is a real-valued kernel on fuzzy sets, i.e., it is defined on $\mathcal{F}(\boldsymbol{\mathcal{X}}) \times \mathcal{F}(\boldsymbol{\mathcal{X}})$. The subscript $\mathcal{L}$ in $k_{\mathcal{L}}$ denotes the explicit dependence of this kernel on the set $\mathcal{L}$. It is worth comparing this new definition with (12) and (13). Observe that if $N = M$, and $\boldsymbol{X}_i = \boldsymbol{A}_l$, then (25) can be expressed as:

$$h(x') = \sum_{l=1}^{M} \alpha_l k_{\mathcal{L}}(\boldsymbol{X}'(\cdot\,|\boldsymbol{x}'), \boldsymbol{A}_l), \tag{31}$$

Lemma 1 Kernel $k_{\mathcal{L}}$ is positive-definite.
Proof: A nonsingleton kernel on fuzzy sets is positive-definite and also is a non-negative kernel because it is a $[0,1]-$ valued kernel. Consequently $\gamma = 1 / \sum_{i=1}^{N} k(\boldsymbol{X}, \boldsymbol{X}_i) \geq 0$, and, by closure properties of positive-definite kernels, kernel $k_{\mathcal{L}}$ is also positive-definite.

The kernel matrix of kernel $k_{\mathcal{L}}$ is given by:

$$K_{\mathcal{L}_{ij}} = k_{\mathcal{L}}\left(\boldsymbol{X}_i(\cdot\,|\boldsymbol{x}_i), \boldsymbol{X}_j(\cdot\,|\boldsymbol{x}_j)\right) \tag{32}$$

This matrix is equivalent to the FBF matrix Φ constructed from (13), i.e., $\Phi_{ij} = \phi_j(\boldsymbol{x}_i) \equiv K_{\mathcal{L}_{ij}}$. This FBF matrix was presented in [12] and used for learning fuzzy systems with the least squares and SVD-QR methods.

Lemma 2 Kernel $k_{KBF}(\boldsymbol{X}_i, \boldsymbol{X}_j) = k_{\mathcal{L}}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) + k_{\mathcal{L}}(\boldsymbol{X}_j, \boldsymbol{X}_i)$ is a positive-definite symmetric kernel.

The kernel matrix of this kernel is $K_{KBF_{ij}} = k_{KBF}\left(\boldsymbol{X}_i(\cdot\,|\boldsymbol{x}_i), \boldsymbol{X}_j(\cdot\,|\boldsymbol{x}_j)\right) = K_{\mathcal{L}_{ij}} + K_{\mathcal{L}_{ji}} \equiv \Phi_{ij} + \Phi_{ji} = \phi_j(\boldsymbol{x}_i) + \phi_i(\boldsymbol{x}_j)$

Lemma 2 allows rewrite the FBF expansion of Mamdani fuzzy systems as:

$$h(x') = \sum_{l=1}^{M} \alpha_l k_{KBF}(\boldsymbol{X}'(\cdot\,|\boldsymbol{x}'), \boldsymbol{X}_l) \equiv \sum_{l=1}^{M} \alpha_l \left(k_{\mathcal{L}}(\boldsymbol{X}'(\cdot\,|\boldsymbol{x}'), \boldsymbol{X}_l) + k_{\mathcal{L}}(\boldsymbol{X}_l(\cdot\,|\boldsymbol{x}_l), \boldsymbol{X}')\right) \tag{33}$$

The positiveness of kernel k_{KBF}, Definitions 4 and 5, and Lemma 2, allow to relate Mamdani fuzzy inference systems and kernel methods as is done in the next Section.

A. Kernel Algorithms for Learning the Structure of Mamdani Fuzzy Systems.

We provide in this section a methodology for performing learning in Mamdani fuzzy systems using kernel machines. The learning algorithm is going to learn the rule-antecedents, where each observation on the dataset can potentially define a rule-antecedent. To explain this consider solving a regression task with the following data set:

$$\mathcal{D} = \{\{(\boldsymbol{x}_1, y_1), \dots (\boldsymbol{x}_N, y_N)\} \subset \mathcal{R}^D \times \mathcal{R}\} \qquad (34)$$

We seek to estimate a function h, such that $y = h(\boldsymbol{x})$ has high generalization and low prediction error. Consider also that we want to use a fuzzy inference system as a learning model such the model is trained with the regularized model for kernel methods in (4).

Algorithm 1, stated below, proposes a general methodology for learning the structure of non-singleton Mamdani fuzzy systems by using kernel methods. The input to the learning algorithm is given by a dataset $\mathcal{D}$ (as in (34)), a kernel on fuzzy sets k_{KBF} (as in Lemma 2), an appropriate loss function L, a regularization parameter λ, and a regularizer Ω.

In step 1, Algorithm 1 constructs a fuzzified version of the data that transforms elements within $\mathcal{D}$, (i.e., each point $\boldsymbol{x}_i \in \mathcal{R}^D, i = 1, \dots, N$) into the fuzzy sets $X_{i1}, X_{i2}, \dots, X_{iD}$. This leads to a *fuzzy dataset* $\mathcal{D}_{\mathcal{F}}$ in (21).

Algorithm 1. Learning the structure of a non-singleton Mamdani fuzzy system by using kernel methods.

Input: $\mathcal{D}, k_{KBF}, L, \lambda, \Omega$

Step-1: data fuzzification

$\mathcal{D}_{\mathcal{F}} \leftarrow$ *data fuzzification*($\mathcal{D}$)

Step-2: kernel method learning

$K_{KBF} \leftarrow$ *compute kernel matrix* $(k_{KBF}, \mathcal{D}_{\mathcal{F}})$

$h \in \mathcal{H},\ N, \alpha_i \leftarrow$ *learning*(K_{KBF}, L, λ, Ω)

Step-3: fuzzy system structure learning

$M \leftarrow N$,

$l \leftarrow i$,

$A_l \leftarrow X_l$, such $\alpha_i \geq 0$

Step-4: (optional)

$M, A_l \leftarrow$ *postprocessing*(M, A_l)

Output: Non-singleton Mamdani fuzzy system with M rules - antecedents A_l

In Step 2, Algorithm 1 estimates the KBF matrix K_{KBF} using the fuzzy dataset $\mathcal{D}_{\mathcal{F}}$ and kernel function k_{KBF}. The estimated KBF matrix is used with an appropriate loss function L (e.g., mean-square error in kernel-ridge-regression or SVM-loss), a regularization parameter λ, and a regularizer Ω for learning the following function:

$$h(\boldsymbol{x}') = \sum_{i=1}^{N} \alpha_i k_{KBF}(\boldsymbol{X}'(\,\cdot\,|\boldsymbol{x}'), \boldsymbol{X}_i(\,.\,|\boldsymbol{x}_i)) \qquad (35)$$

where $\boldsymbol{X}'(\,\cdot\,|\boldsymbol{x}')$ is the fuzzified version of $\boldsymbol{x}'$, as described earlier. Note that h is a function belonging to the RKHS $\mathcal{H}$ induced by kernel k_{KBF}.

In Step 3, rules are chosen such that $|\,\alpha_i| \geq 0$, i.e., fuzzy sets $\boldsymbol{X}_i(\cdot\,|\boldsymbol{x}_i)$, with $|\,\alpha_i| \geq 0$ become rule-antecedents. This bounds the number of rules M, such that $M \leq N$. Note that by the structure of the learning algorithm each element $\boldsymbol{X}_i(\cdot\,|\boldsymbol{x}_i)$, in a fuzzy dataset $\mathcal{D}_{\mathcal{F}}$ can potentially define the fuzzy sets of a rule-antecedent, (i.e. $X_{i1}, X_{i2}, \dots, X_{iD}$ can potentially become fuzzy sets $A_{l1}, A_{l2}, \dots, A_{lD}$) because if $|\,\alpha_i| \geq 0$ then $l \leftarrow i$ and the fuzzy set $\boldsymbol{X}_i(\cdot\,|\boldsymbol{x}_i)$ is transformed into the fuzzy set $\boldsymbol{A}_l$. Note that the parameters for those fuzzy sets are not necessarily fixed to a unique value.

Finally, Step 4 of Algorithm 1, denotes postprocessing (optional), e.g., rule-reduction, adjustment of antecedent's parameters, adding linguistic rules, etc. This step can also take advantage of the structure of the KBF matrix, for example, one can investigate the potential use of low-rank kernel matrix approximations as in [44] with the purpose of rule reduction. We leave this topic for further research.

VI. Conclusions and Future Research

This work presents the first step in studying the connection between fuzzy inference systems and kernel methods. Specifically, we analyzed the connection between the FBF expansion of non-singleton Mamdani fuzzy systems and regularized kernel methods. We show that the input-antecedent interaction of non-singleton Mamdani fuzzy systems induces a kernel on fuzzy sets that we called, the *non-singleton kernel on fuzzy sets*. Such a kernel allows writing the FBF-expansion of fuzzy systems in terms of that kernel. We called such an expansion a *kernel-basis-function (KBF) expansion*.

The KBF-allows one to relate kernel methods with positive-definite kernels and non-singleton Mamdani fuzzy systems. The consequences and benefits for doing this are that fuzzy systems can: 1) be trained by kernel algorithms; 2) have resistance to the curse of dimensionality; 3) satisfy the Representer Theorem of kernel methods; 4) learn functions in RKHS; and 5) explicitly regularize the function that is being approximated.

We also provided some benefits from fuzzy systems towards kernel methods. Additionally, we related kernel methods to fuzzy systems without simplifying the denominator of the original FBF-expansion, as is done in previous works. We also dropped a restriction of previous works of having fuzzy sets sharing some of their antecedent parameters (e.g., Gaussian membership functions with the same variance) thanks to the use of kernel on fuzzy sets that allow constructing a kernel matrix directly from fuzzy sets. Finally, we presented an algorithm showing the methodology of learning fuzzy systems via kernel methods.

Open research questions are 1) Explore connections for other kinds of fuzzy systems (TSK, type-2, etc.), 2) validate experimentally the presented connection in several Machine learning tasks (classification, regression, novelty detection, etc.); 3) explore how to deal with structured data following this connection; 4) propose practical non-singleton kernels on fuzzy sets; and 5) investigate how to exploit the kernel structure beyond the arguments presented in this paper, for example, kernel low-rank approximations for rule reduction, or explore other possible connections between fuzzy systems and: a) stochastic processes following the view of kernels as covariance functions; b) non-parametric statistical estimators based on non-negative kernels (Kernel smoothing, KDE); and c) kernel methods with non-positive kernels as in [45].

REFERENCES

[1] Kosko, B. "Fuzzy systems as universal approximators." *IEEE transactions on computers* 43.11 (1994): 1329-1333

[2] Yager, R R. and L. A. Zadeh, eds. "An introduction to fuzzy logic applications in intelligent systems," Vol. 165. *Springer Science & Business Media*, 2012.

[3] Ying, H., "Fuzzy control and modeling: analytical foundations and applications," *Wiley-IEEE Press*, 2000.

[4] Sun, Y. L,, and Meng Joo Er. "Hybrid fuzzy control of robotics systems." *IEEE Transactions on Fuzzy Systems* 12.6 (2004): 755-765.

[5] Chen, C. L. P., et al. "Fuzzy restricted Boltzmann machine for the enhancement of deep learning." *IEEE Transactions on Fuzzy Systems* 23.6 (2015): 2163-2173.

[6] Hüllermeier, E., "Does machine learning need fuzzy logic?." *Fuzzy Sets and Systems* 281 (2015): 292-299.

[7] Zadeh, L. A., "Fuzzy sets." *Information and control* 8.3 (1965): 338-353.

[8] Goguen, J. A.. "L-fuzzy sets." *Journal of mathematical analysis and applications* 18.1 (1967): 145-174.

[9] Karnik, N. N., Mendel, J. M., and Liang, Q., "Type-2 fuzzy logic systems." *IEEE transactions on Fuzzy Systems* 7.6 (1999): 643-658.

[10] Bustince H., et al. "A historical account of types of fuzzy sets and their relationships." *IEEE Transactions on Fuzzy Systems* 24.1 (2015): 179-194.

[11] Guevara, J., Hirata Jr., R., and Canu, S., "Positive definite kernel functions on fuzzy sets." *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2014.

[12] Mendel, J. M., *Uncertain rule-based fuzzy systems. Introduction and New Directions*. 2nd Edition Springer, Cham, Switzerland, 2017.

[13] Dubois, D, and Prade, H., "The three semantics of fuzzy sets." *Fuzzy sets and systems* 90.2 (1997): 141-150.

[14] van den Berg, C., Christensen, J. P. R, Ressel, P., "Harmonic analysis on semigroups: theory of positive definite and related functions". Vol. 100. *New York: Springer*, 1984.

[15] Buhmann, M. D., "Radial basis functions: theory and implementations. " Vol. 12. *Cambridge University Press*, 2003.

[16] Cheney, E. W., William Allan Light. "A course in approximation theory." Vol. 101. *American Mathematical Society*., 2009.

[17] Matheron, G., "Les variables régionalisées et leur estimation: une application de la théorie des fonctions aléatoires aux sciences de la nature. " *Masson et CIE*, 1965.

[18] Berlinet, A., Thomas-Agnan, C., "Reproducing kernel Hilbert spaces in probability and statistics. " *Springer Science & Business Media*, 2011.

[19] Wahba, G., "Spline models for observational data. " Vol. 59. *SIAM*, 1990

[20] Scholkopf, B., Smola, A. J., "Learning with kernels: support vector machines, regularization, optimization, and beyond. " *MIT press*, 2001.

[21] Shawe-Taylor, J., Nello, C., "Kernel methods for pattern analysis." *Cambridge University Press*, 2004.

[22] Mouzouris, G. C., Mendel, J. M., "Nonsingleton fuzzy logic systems: theory and application." *IEEE Transactions on Fuzzy Systems* 5.1 (1997): 56-71.

[23] Wang, L., Mendel, J. M., "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning." *IEEE transactions on Neural Networks,* 3.5 (1992): 807-814.

[24] Van Leekwijck, W., Kerre, E. E., "Defuzzification: criteria and classification." *Fuzzy sets and systems,* 108.2 (1999): 159-178.

[25] Mamdani, E. H., "Application of fuzzy logic to approximate reasoning using linguistic synthesis." *Proceedings of the sixth international symposium on Multiple-valued logic*. IEEE Computer Society Press, 1976.

[26] Takagi, T., Sugeno, M., "Fuzzy identification of systems and its applications to modeling and control." *IEEE Transactions on Systems, Man, and Cybernetics* 1 (1985): 116-132.

[27] Guevara, J., Hirata Jr., R., Canu, S., "Kernel functions in Takagi-Sugeno-Kang fuzzy system with nonsingleton fuzzy input." *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2013.

[28] Aronszajn, N., "Theory of reproducing kernels." *Transactions of the American Mathematical Society* 68.3 (1950): 337-404.

[29] Kim, H. M., Mendel, J. M., "Fuzzy basis functions: Comparisons with other basis functions." *IEEE Transactions on Fuzzy Systems* 3, 2 (1995): 158-168.

[30] Jang, J-S.R., Sun, C.T., "Functional equivalence between radial basis function networks and fuzzy inference systems." *IEEE Transactions on Neural Networks* 4, 1 (1993): 156-159.

[31] Hunt, K. J., Haas, R., Murray-Smith, J.. "Extending the functional equivalence between radial basis function networks and fuzzy inference systems." *IEEE Transactions on Neural Networks* 7, no. 3 (1996): 776-781.

[32] Andersen, H. C., Lotfi, A., Westphal, L. C., "Comments on "Functional equivalence between radial basis function networks and fuzzy inference systems' [with reply]." *IEEE Transactions on Neural Networks* 9, no. 6 (1998): 1529-1532.

[33] Chen, Y., Wang, J. Z., "Support vector learning for fuzzy rule-based classification systems." *IEEE Transactions on Fuzzy Systems* 11, no. 6 (2003): 716-728.

[34] Chiang, J., Hao, P-Y.. "Support vector learning mechanism for fuzzy rule-based modeling: a new approach." *IEEE Transactions on Fuzzy systems* 12, no. 1 (2004): 1-12.

[35] Hao, P-Y., Chiang, J., "Fuzzy regression analysis by support vector learning approach." *IEEE Transactions on Fuzzy Systems* 16, no. 2 (2008): 428-441.

[36] Guevara, J., Hirata Jr., R, Canu, S., "Cross product kernels for fuzzy set similarity." In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-6. IEEE, 2017.

[37] Guevara, J., Hirata Jr., R, Canu, S., "Fuzzy set similarity using a distance-based kernel on fuzzy sets." *HAL archives-ouverts* (2015).

[38] Cuturi, M., Fukumizu, K. and Vert, J.P., 2005. Semigroup kernels on measures. *Journal of Machine Learning Research*, *6*(Jul), pp.1169-1198.

[39] Jebara, T., Kondor, R. and Howard, A., 2004. Probability product kernels. *Journal of Machine Learning Research*, *5*(Jul), pp.819-844.

[40] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C., 2002. Text classification using string kernels. *Journal of Machine Learning Research*, *2*(Feb), pp.419-444.

[41] Gärtner, T., 2003. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, *5*(1), pp.49-58.

[42] Tsybakov, A.B., 2008. *Introduction to nonparametric estimation*. Springer Science & Business Media.

[43] Wang, L.X., 1992, March. Fuzzy systems are universal approximators. In *[1992 Proceedings] IEEE International Conference on Fuzzy Systems* (pp. 1163-1170). IEEE.

[44] Bach, F., 2013, June. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory* (pp. 185-209).

[45] Ong, C.S., Mary, X., Canu, S. and Smola, A.J., 2004, July. Learning with non-positive kernels. In *Proceedings of the twenty-first international conference on Machine learning* (p. 81).

[46] Zhang, T., 2005. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, *17*(9), pp.2077-2098.

[47] Vert, R. and Vert, J.P., 2006. Consistency and convergence rates of one-class SVMs and related algorithms. *Journal of Machine Learning Research*, *7*(May), pp.817-854.