



HAL
open science

Cross product kernels for fuzzy set similarity

Jorge Guevara, Roberto Hirata, Stéphane Canu

► **To cite this version:**

Jorge Guevara, Roberto Hirata, Stéphane Canu. Cross product kernels for fuzzy set similarity. 2017.
hal-01438607v2

HAL Id: hal-01438607

<https://hal.science/hal-01438607v2>

Preprint submitted on 25 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cross product kernels for fuzzy set similarity

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

Roberto Hirata Jr.
University of Sao Paulo
Sao Paulo, Brazil
Email: hirata@ime.usp.br

Stéphane Canu
Normandie Université
INSA de Rouen - LITIS
Email: scanu@insa-rouen.fr

Abstract—We present a new kernel on fuzzy sets: the *cross product kernel on fuzzy sets* which can be used to estimate similarity measures between fuzzy sets with a geometrical interpretation in terms of inner products. We show that this kernel is a particular case of the convolution kernel and it generalizes the widely-know kernel on sets towards the space of fuzzy sets. Moreover, we show that the cross product kernel on fuzzy sets performs an embedding of probability measures into a reproduction kernel Hilbert space. Finally, we experimentally show the applicability of this kernel on a supervised classification task on noisy datasets.

I. INTRODUCTION

Similarity measures between elements from a set are key components in several areas of research. For instance, in machine learning the concept of similarity measure is primal to find out regions within the data space with similar characteristics. That is useful in the definition of tasks like classification, clustering or density estimation. The same is true in computer vision and image processing where well-defined similarity measures play an important role in practical tasks like image segmentation, object tracking and recognition.

A similarity measure is a value, that is close to zero for objects with different properties. That is, two objects are *more similar* if that value is far away from zero. For instance, the inner product is a similarity measure defined on elements of a vector space. We emphasize the importance of the inner product as a similarity measure because it has a geometrical interpretation, and hence it enables a geometrical understating of algorithms that make use of it.

It is important to mention that most of the data collected from real word problems cannot be interpreted as an inner product space¹. For example, in social data analytics, data usually is given by collection of graphs; in bio-informatics, the biological information is encoded by strings; in natural language processing, procedures almost work with words and grammars; in artificial intelligence, the data could be defined by a set of logic predicates.

In the realm of fuzzy data analytics, the concept of similarity measure is fundamental [1]. There has been extensive research on defining similarity measures between fuzzy sets and applying those similarity measures on real tasks [2]. However, most of the existing similarity measures between fuzzy sets have a lack of geometrical interpretation because such similarity

measures are defined on the space of fuzzy sets which is not a vectorial space.

The main subject of this paper is to investigate similarity measures for fuzzy sets with geometrical interpretation on Hilbert spaces. In order to properly define similarity measures on fuzzy sets with such characteristics we use the concept of positive definite kernels.

A. Positive definite kernels

Positive kernels are functions acting as similarity measures between elements from a set. They provide a way to compute an inner product in Reproducing Kernel Hilbert Spaces (RKHS) \mathcal{H} for non-vectorial data. Formally, a kernel k is a real-valued function defined on $\mathcal{X} \times \mathcal{X}$ with \mathcal{X} being a non-empty set. A kernel k is positive definite if it satisfies:

$$\sum_{i=1}^N \sum_{j=1}^N c_i c_j k(x_i, x_j) \geq 0, \quad (1)$$

for all $N \in \mathbb{N}$, $\{c_1, \dots, c_N\} \subset \mathbb{R}$, $\{x_1, \dots, x_N\} \subset \mathcal{X}$. Positive definite kernels have the following properties

- 1) $\forall x \in \mathcal{X}$ there is a function $\phi_x = k(\cdot, x) \in \mathcal{H}$
- 2) $\forall f \in \mathcal{H}, \forall x \in \mathcal{X} \langle k(\cdot, x), f \rangle_{\mathcal{H}} = f(x)$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in \mathcal{H} .

Property 1) assures that for every element in the input space there always exist a function that represent the element in the RKHS. Property 2) is called *reproducing property* which implies an implicitly evaluation of inner products using kernels:

$$k(x, y) = \langle \phi_x, \phi_y \rangle_{\mathcal{H}}, \quad (2)$$

where functions $\phi_y = k(\cdot, y)$ and $\phi_x = k(\cdot, x)$, are the *representative functions* of elements $x, y \in \mathcal{X}$, respectively.

B. Contributions

We claim the following contributions:

- We define the cross product kernel on fuzzy sets as a new similarity measure for fuzzy sets using kernels on fuzzy sets with a geometrical interpretation by means of inner products.
- We show several properties of the proposed kernel, for instance, we show that it is a particular case of the convolution kernel, it generalizes the kernel on sets, it embeds probability distributions into a RKHS.
- We experimentally show the power of the proposed kernels through a supervised classification experiments on noisy attribute datasets.

¹An inner product space is a vectorial space equipped with an inner product

C. Previous work

Kernels on fuzzy sets is a new area of research on the realm of similarity measures between fuzzy sets. Some of those kernels are the intersection kernel on fuzzy sets which is defined in terms of the T-norm operator and it turns out to be positive definite [3]. The non-singleton Takagi-Sugeno-Kang fuzzy kernel, which can be casted as an intersection kernel. That kernel arises from the context of fuzzy systems and it has an interpretation as fuzzy equivalence relation [4]. The distance-based kernel on fuzzy sets is a kernel defined by putting a distance function between fuzzy sets into the kernel definition using the concept of distance substitution kernels [5]. Practical application on those kernels were given within the machine learning area, for instance the non-singleton Takagi-Sugeno-Kang was applied on the supervised classification of low quality data. Distance-based kernels on fuzzy sets and intersection kernel on fuzzy sets were used on a hypotheses testing task using heterogeneous data containing attributes given by linguistic values [4], [5].

II. THE CROSS PRODUCT KERNEL

Kernels on sets have been widely used in computer vision and machine learning applications as *similarity measures* between two sets [6], [7]. They are defined as follows: Let Ω be a non-empty set and $\mathcal{G}(\Omega)$ be the set of all non-empty countable finite subsets of Ω . The *cross product kernel* is a real-valued mapping $k : \mathcal{G}(\Omega) \times \mathcal{G}(\Omega) \rightarrow \mathbb{R}$ defined by [6]:

$$k_{set}(A, B) = \sum_{x \in A, y \in B} k(x, y), \quad (3)$$

where the k is a real-valued kernel on $\Omega \times \Omega$.

The kernel k_{set} defines a class of kernels on sets that depend on k . If k is positive definite then k_{set} defines a similarity measure for any two sets $A, B \in \mathcal{G}(\Omega)$ by:

$$k_{set}(A, B) = \langle \phi_A, \phi_B \rangle_{\mathcal{H}}, \quad (4)$$

where ϕ_A and ϕ_B are the *representer functions* in a RKHS \mathcal{H} for the sets A, B , respectively. For instance, if $(\Omega, \mathcal{A}, \mu)$ is a *finite measure space*, and $k : \Omega \times \Omega \rightarrow \mathbb{R}$ is a continuous function with finite integral², then k_{set} is given by³:

$$k_{set}(A, B) = \iint_{\substack{x \in A \\ y \in B}} k(x, y) d\mu(x) d\mu(y). \quad (5)$$

In the next section we define a kernel on fuzzy sets which can be used to define similarity measures between fuzzy sets.

III. THE CROSS PRODUCT KERNEL ON FUZZY SETS

In this section we define the cross product kernel on fuzzy sets. We show the case when that kernel is positive definite and consequently define a similarity measure for fuzzy sets in terms of the inner product operation. We use the following notation: Fuzzy sets on Ω are denoted by X, Y, Z . The

² k is a μ -integrable function, i.e., $k \in \mathcal{L}^1(\mu)$. (Definition 10.1, [8]).

³If $\iint_{\Omega} k(x, y) d\mu(x) d\mu(y) > 0$, $\forall \mu \in M_b(\Omega)$, where $M_b(\Omega)$ is the set of all finite signed Borel measures on Ω , then k is said to be *integrally strictly pd*.

membership function $\Omega \rightarrow [0, 1]$ of a fuzzy sets X is denoted by $X(\cdot)$, consequently $X(x)$, $x \in \Omega$ is the membership degree of x to the fuzzy set X . The support of a fuzzy set X , i.e. $\{x \in \Omega \mid X(x) > 0\}$, is denoted by $\text{supp}(X)$. Finally, the set of all fuzzy sets in Ω is denoted by $\mathcal{F}(\Omega)$.

Assuming that for all $X \in \mathcal{F}(\Omega)$ the set $\text{supp}(X)$ is a nonempty finite countable set, we present the cross product kernel on fuzzy sets in the following definition.

Definition 1 (Cross product kernel on fuzzy sets). Given two real-valued kernels k_1, k_2 defined on $\Omega \times \Omega$ and $[0, 1] \times [0, 1]$ respectively. The cross product kernel on fuzzy sets is a function $k_{\times} : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) \rightarrow \mathbb{R}$ given by:

$$k_{\times}(X, Y) = \sum_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} k_1 \otimes k_2((x, X(x)), (y, Y(y))), \quad (6)$$

where $X(x)$ and $Y(y)$ are the membership degrees for the elements $x, y \in \Omega$ to the fuzzy sets X, Y , and \otimes denotes the tensorial product: $k_1 \otimes k_2 : (\Omega \times [0, 1]) \times (\Omega \times [0, 1]) \rightarrow \mathbb{R}$ defined by:

$$k_1 \otimes k_2(x, X(x), y, Y(y)) = k_1(x, y) k_2(X(x), Y(y)). \quad (7)$$

The cross product kernel on fuzzy sets defines a class of kernels on fuzzy sets depending on the base kernels k_1, k_2 . The next section shows how k_{\times} can be used to define a similarity measure for two fuzzy sets by means of inner products in \mathcal{H} .

A. Positive Definite Kernels on fuzzy sets

In order to have an interpretation of the cross product kernel on fuzzy sets in terms of inner products in a RKHS \mathcal{H} we restrict the kernels k_1 and k_2 to be positive definite.

Lemma 1. *If k_1 and k_2 are real-valued positive definite kernels, then the cross product kernel on fuzzy sets is a real-valued positive definite kernel.*

Proof. The operator \otimes preserves positivity (Corollary 1.13, [9]) then $k_1 \otimes k_2 = k_1 k_2$ is a positive definite kernel on $(\Omega \times [0, 1]) \times (\Omega \times [0, 1])$. By closure properties of kernels, the sum of positive definite kernels is positive definite. Consequently, k_{\times} is positive definite. \square

Corollary 2. *A positive definite cross product kernel on fuzzy sets defines a similarity measure for two fuzzy sets $X, Y \in \mathcal{F}(\Omega)$ as follows:*

$$k_{\times}(X, Y) = \langle \phi_X, \phi_Y \rangle_{\mathcal{H}}, \quad (8)$$

Proof. The true of the above statement follows from the reproducing property of positive definite kernels. \square

Corollary 2 basically says that k_{\times} is a similarity measure for fuzzy sets with an implicit interpretation in terms of inner products in a RKHS. This result enables the use of geometrical algorithms, for instance kernel algorithms in machine learning, on sets of fuzzy sets. i.e., fuzzy data. Those algorithms can be used on elements $\phi_X \in \mathcal{H}$ by means of the kernel k_{\times} . Another consequence of Corollary 2 is that by means of $\langle X, Y \rangle_{\mathcal{F}(\Omega)} \stackrel{\text{def}}{=} k_{\times}(X, Y)$

$\langle \phi_X, \phi_Y \rangle_{\mathcal{H}}$, it is possible to induce an inner product on the fuzzy set space which induces a topology on the space of fuzzy sets. Moreover, it is even possible to use negative or indefinite kernels for k_1 and k_2 in k_{\times} , the geometrical representation for such cases is guaranteed in pseudoEuclidean and Krein spaces [9]–[11].

B. Examples

Example 1. Table I shows four instances of the cross product kernel on fuzzy sets when $k_2(X(x), Y(y)) = X(x)Y(y)$ and k_1 is a positive definite kernel.

$k_1(x, y)$	$k_{\times}(X, Y)$
linear	$\sum_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} xyX(x)Y(y)$
polynomial	$\sum_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} (\gamma\langle x, y \rangle + b)^{\beta} X(x)Y(y)$
exponential	$\sum_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} \exp(\gamma\langle x, y \rangle) X(x)Y(y)$
gaussian	$\sum_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} \exp(-\gamma\ x - y\ ^2) X(x)Y(y)$

TABLE I
EXAMPLES OF CROSS PRODUCT KERNELS ON FUZZY SETS.

Kernels from Table I weight each element of the set by their membership degree. That permits to incorporate into a machine learning classifier the uncertainty of observations by means of those kernels.

Example 2. Let $(\Omega, \mathcal{A}, \mu)$ be a *finite measure space*. Let k_1, k_2 be continuous functions with finite integral. The kernel

$$k_{\times}(X, Y) = \iint_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} k_1 \otimes k_2((x, X(x)), (y, Y(y))) d\mu(x) d\mu(y), \quad (9)$$

is a cross product kernel on fuzzy sets.

Example 3. Replacing the measure μ of the previous example with a probability measure \mathbb{P} results in the following cross product kernel on fuzzy sets:

$$k_{\times}(X, Y) = \iint_{\substack{x \in \text{supp}(X), \\ y \in \text{supp}(Y)}} k_1 \otimes k_2((x, X(x)), (y, Y(y))) d\mathbb{P}(x) d\mathbb{P}(y), \quad (10)$$

The previous example shows a way to included into the kernel two important concepts on uncertainty modeling: fuzzyness and randomness. Fuzzyness in the form of membership functions and randomness because, independently of the degree of membership of x to the fuzzy set X , the above formulation considers the values x being outcomes of a random variable with probability distribution \mathbb{P} .

C. Generalization towards a product space

A generalization of k_{\times} to deal with a D -tuple of fuzzy sets, i.e., $(X_1, \dots, X_D) \in \mathcal{F}(\Omega_1) \times \dots \times \mathcal{F}(\Omega_D)$ is implemented by the following kernel:

$$k_{\times}^{\pi}((X_1, \dots, X_D), (Y_1, \dots, Y_D)) = \prod_{d=1}^D k_{\times}^d(X_d, Y_d). \quad (11)$$

If all the kernels k_{\times}^d are positive definite then k_{\times}^{π} is positive definite by closure properties of kernels. Another generalization based on addition of positive definite kernels is also possible:

$$k_{\times}^{\Sigma}((X_1, \dots, X_D), (Y_1, \dots, Y_D)) = \sum_{d=1}^D \alpha_i k_{\times}^d(X_d, Y_d). \quad (12)$$

Kernel k_{\times}^{Σ} is positive definite if only if $\alpha_i \in \mathbb{R}^+$ and all the k_{\times}^d kernels are positive definite.

IV. PROPERTIES

We show that cross product kernels on fuzzy sets satisfy:

- k_{\times} is a convolution kernel;
- k_{\times} generalize the cross product kernel on sets;
- k_{\times} embeds probability distributions into a RKHS.

A. Kernel k_{\times} is a convolution kernel

Convolution kernels [6], [7] are kernels on sets, whose elements are discrete structures. We define them as follows.

Definition 2 (Convolution kernel). Let e and $\vec{e} = (e_1, e_2, \dots, e_L)$ be elements of the sets E and $E_1 \times \dots \times E_L$, respectively. Given the relation $R \subseteq (E_1 \times E_2 \times \dots \times E_L) \times E$, with characteristic function:

$$R : E_1 \times E_2 \times \dots \times E_L \times E \rightarrow \{\text{true}, \text{false}\} \\ (\vec{e}, e) \mapsto R(\vec{e}, e), \quad (13)$$

and given the decomposition $R^{-1}(e) = \{\vec{e} \mid R(\vec{e}, e) = \text{true}\}$. The convolution kernel is a real valued function on $E \times E$, satisfying $\forall e, e' \in E$:

$$k_{conv}(e, e') = \sum_{\vec{e} \in R^{-1}(e), \vec{e}' \in R^{-1}(e')} \prod_{l=1}^L k_l(e_l, e'_l), \quad (14)$$

where $k_l, 1 \leq l \leq L$ are positive definite kernels on $E_l \times E_l$.

Next, we prove that the cross product kernel on fuzzy sets is a convolution kernel. This can be achieved by properly define a relation among: 1) elements of fuzzy sets, 2) membership degrees, and 3) fuzzy sets. That idea is stated formally in the following proposition.

Proposition 3. *The kernel k_{\times} is a convolution kernel.*

Proof. Let R be a relation with characteristic function from $R : \Omega \times [0, 1] \times \mathcal{F}(\Omega)$ to $\{\text{true}, \text{false}\}$ such that $R(x, X(x), X)$ is *true* only if $x \in \text{supp}(X)$. Consider the decomposition: $R^{-1}(X) = \{(x, X(x)) \mid R(x, X(x), X) = \text{true}\}$. Replacing

those concepts into (14), we get the following convolution kernel on $\mathcal{F}(\Omega) \times \mathcal{F}(\Omega)$:

$$k_{conv}(X, Y) = \sum_{\vec{e} \in R^{-1}(X), \vec{e}' \in R^{-1}(Y)} \prod_{l=1}^L k_l(e_l, e'_l),$$

where $\vec{e} = (x, X(x))$ and $\vec{e}' = (y, Y(y))$. Now, if we restrict $L = 1$, and k_l be the kernel $k_1 \otimes k_2$ from Definition 1, we end up with

$$k_{conv}(X, Y) = \sum_{\substack{(x, X(x)) \in R^{-1}(X), \\ (y, Y(y)) \in R^{-1}(Y)}} k_1(x, y) k_2(X(x), Y(y)),$$

which is $k_{\times}(X, Y)$. \square

B. Kernel k_{\times} generalizes the cross product kernel on sets

A set $A \subset \Omega$ can be characterized by its characteristic function $\mathbf{1}_A : \Omega \rightarrow \{0, 1\}$, which is 1 if $x \in A$ and 0 otherwise. Hence, a set A could be viewed as a fuzzy set with membership function $\mathbf{1}_A$. From definition 1) and assuming that k_2 is the linear kernel we have:

$$\begin{aligned} k_{\times}(A, A') &= \sum_{\substack{x \in \text{supp}(A), \\ y \in \text{supp}(A')}} k_1(x, y) \mathbf{1}_A(x) \mathbf{1}_{A'}(y) \\ &= \sum_{x \in A, x' \in A'} k(x, x'), \end{aligned}$$

which is exactly the kernel on sets k_{set} from (4).

C. k_{\times} embeds probability distributions into a RKHS.

In order to show that, we denote by \mathcal{P} the set of all the probability measures on \mathbb{R}^D , probability measures by \mathbb{P} and random variables with probability distribution \mathbb{P} by \mathbb{X} . The embedding of probability measure $\mathbb{P} \in \mathcal{P}$ into a RKHS \mathcal{H} is the mapping from \mathcal{P} to \mathcal{H} defined by $\mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{\mathbb{P}}[k(\mathbb{X}, \cdot)]$, where \mathbb{E} denotes the expectation taken over \mathbb{X} and k is a positive definite kernel on $\mathbb{R}^D \times \mathbb{R}^D$ [12]. With this in mind, the kernel from Example 3 could be rewritten as

$$k_{\times}(X, Y) = \mathbb{E}_{\mathbb{P}} \left[k_1 \otimes k_2 \left((\mathbb{X}, X(\mathbb{X})), (\mathbb{Y}, Y(\mathbb{Y})) \right) \right] \quad (15)$$

Finally, if we define $k' = k_1 \otimes k_2$ and $\mathbb{X}' = (\mathbb{X}, X(\mathbb{X}))$ then the mapping $\mathbb{P} \mapsto \mu_{\mathbb{P}} = \mathbb{E}_{\mathbb{P}}[k'(\mathbb{X}', \cdot)]$ is a valid embedding of \mathbb{P} into a RKHS \mathcal{H} induced by kernel $k_1 \otimes k_2$.

V. EXPERIMENTS

The goal of this experiment is to test the efficacy of k_{\times} as similarity measure on a supervised classification task. We believe that a classification task provides a valid framework to test the goodness of similarity measures, for instance, a support vector machine (SVM) would have more generalization power if a good kernel is used to estimate the similarity measure between the observations within data.

Fuzzy datasets can be used to model observations with point-wise uncertainty introduced from several sources, for

instance noise measurements. With this in mind, we used the *PIMA and SONAR attribute noise data sets* from the *KEEL-dataset repository* [13] to perform the experiments.

We used MATLAB and R for the experiments, the code used in the experiments can be obtained from <https://github.com/jorjasso/crossProductKernel-noisyData>.

A. Datasets

Each either PIMA or SONAR attribute noise data set consist of twelve datasets which are noisy versions from the original either PIMA or SONAR datasets. Each one of those twelve datasets either from PIMA or SONAR are grouped in three categories, four sets of data per category. The categories are: noisy train - noisy test, noisy-train - clean test and clean train- noisy test (see Table II). The noisy algorithm injection is described in [14]. The four datasets within each category contain 5%, 10%, 15% e 20% noise levels. All the PIMA attribute noise data sets contain 768 observations belonging to one of the two classes. The proportion of observations within classes is 35%–65%, consequently they are unbalanced datasets. Each dataset contains 8 independent variables. All the SONAR attribute noise data sets contain 208 observations within two classes. The proportion of observations within classes is 47% – 53%. The datasets have 60 independent variables. Table II summarizes the PIMA and SONAR attribute noise data sets used in this experiment.

TABLE II
SUMMARY OF THE PIMA AND SONAR ATTRIBUTE NOISE DATASETS

Dataset	%Noise	Dataset	%Noise
pima-5an-nn	5%	sonar-5an-nn	5%
pima-10an-nn	10%	sonar-10an-nn	10%
pima-15an-nn	15%	sonar-15an-nn	15%
pima-20an-nn	20%	sonar-20an-nn	20%
pima-5an-nc	5%	sonar-5an-nc	5%
pima-10an-nc	10%	sonar-10an-nc	10%
pima-15an-nc	15%	sonar-15an-nc	15%
pima-20an-nc	20%	sonar-20an-nc	20%
pima-5an-cn	5%	sonar-5an-cn	5%
pima-10an-cn	10%	sonar-10an-cn	10%
pima-15an-cn	15%	sonar-15an-cn	15%
pima-20an-cn	20%	sonar-20an-cn	20%

B. Experimental methodology

We used the attribute noise data sets on a supervised classification task. In order to use kernel on fuzzy sets, we applied two fuzzification approaches on the original data. Such procedure estimate a membership degree for the values within the dataset. Further, to have a better estimate of the generalization error, we carried out nested cross-validation experiments using a SVM and those kernels. In the next sections we describe the two fuzzification approaches, the kernel setup and the nested cross-validation experiments.

C. Fuzzification

We constructed an augmented version of each noise data set by adding information of the membership degrees of the values

in the data. To that end, we modeled each independent variable per class with fuzzy sets using two fuzzification approaches :

a) *First fuzzification approach*: For each dataset in Table II, we constructed a fuzzy set for each independent variable per class, that procedure resulted in 16 fuzzy sets. i.e., 8 fuzzy sets per class. Each fuzzy set was constructed using a Gaussian membership function with parameters μ, σ . i.e., $\exp(-0.5(x - \mu)^2/\sigma^2)$. Those parameters were estimated in the following way: for each independent variable in a specific class, we estimated the first, second and third quartile of the distribution of values of that independent variable, we denote them as q_1, q_2, q_3 , respectively. Then we used the following setup: $\mu = q_2$, and $\sigma = |q_3 - q_1|/(2 * \sqrt{2 * \log 2})$. This setting correspond to the *full width at half maximum* used in signal processing. Finally, we augmented each dataset from Table II with the membership values of each value within the observation using those 16 fuzzy sets. This fuzzification approach is summarized in Algorithm 1. Noisy datasets are denoted by \mathcal{D} , number of observations by N and the number of independent variables by D . Each observation x_i is an element from \mathbb{R}^D and y_i is a value in $\{-1, 1\}$. We denote the output fuzzy dataset by \mathcal{MF} and the fuzzy sets by X_d .

Algorithm 1 First fuzzification approach

Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$
Output: $\mathcal{MF} = \{(x_i, X_1(x_i^1), \dots, X_D(x_i^D), y_i)\}_{i=1}^N$
for each class y_i **do**
 for $d = 1$ to D **do**
 $q_1, q_2, q_3 = \text{quantile}(x_{1 \leq i \leq N}^d, (0.25, 0.5, 0.75))$
 $\mu_d = q_2$
 $\sigma_d = |q_3 - q_1|/(2 * \sqrt{2 * \log 2})$
 $X_d(\cdot) = \exp(-0.5(\cdot - \mu_d)^2/\sigma_d^2)$
 end for
end for

b) *Second fuzzification approach*: The procedure is almost similar to the first fuzzification approach, however, the membership function of each fuzzy set is a scaled version of the empirical probability density function of the distribution of values of each independent variable per class. That is, for each independent variable we estimated the histogram of their values in a particular class. In order to create a valid membership function, we interpolated the scaled values from the histogram such that we end up with a $[0, 1]$ -valued function. Algorithm 2 describes this fuzzification approach.

D. Kernels

We used the linear kernel and the Gaussian kernel as baseline kernels, they were applied on datasets from Table II. We used the linear, exponential and Gaussian kernels for k_{\times} listed in Table I. We denoted those kernels by *fuzzy linear - I*, *fuzzy exp - I* and *fuzzy Gaussian - I*, respectively, if they were used on the augmented data from the first fuzzification approach. Otherwise, if they were used on the augmented data from the second fuzzification approach we denoted them by *fuzzy linear - II*, *fuzzy exp - II* and *fuzzy Gaussian - II*.

Algorithm 2 Second fuzzification approach

Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$
Output: $\mathcal{MF} = \{(x_i, X_1(x_i^1), \dots, X_D(x_i^D), y_i)\}_{i=1}^N$
for each class y_i **do**
 for $d = 1$ to D **do**
 $h = \text{histogram}(x_{1 \leq i \leq N}^d)$
 $h = h / \max(h)$
 $X_d(\cdot) = \text{linearInterpolation}(h)$
 end for
end for

E. Nested cross validation experiments

In order to have a good estimate of the generalization error, we performed a nested cross-validation experiment as suggested in [15] which uses an internal loop to perform model selection and an outer loop to access the model performance. We used the nested cross-validation procedure on each of the eight models given by a SVM with one of the eight kernels previously described. We used the Area Under the Curve (AUC) metric as performance metric.

We tested each classifier (a SVM with one of the eight kernels) using the five partitions available for each PIMA and SONAR attribute noise data set from the Keel repository. Each partition contains a pair of train and test subsets of the original data. For each of the five partitions, we performed model selection on the training subset, for this purpose, we estimated the hyper-parameters of the classifier with the smallest ten fold cross-validation error. The hyper-parameters were the kernel and the regularization parameter for the SVM. Those parameters were estimated from a grid of hyper-parameters. We trained a SVM with the kernel using that best hyper-parameters on all the training data for that partition, and further, we evaluated the performance of the classifier in the remaining test set using the AUC metric. Finally, we reported the mean of the AUC value across the test sets of the five partitions.

F. Results

Figure 1 presents from left to right the results for the clean train - noisy test, noisy train - noisy test, and noisy train - clean versions of the PIMA dataset, respectively. The x-axis shows the injected noise level and the y-axis shows the mean of the AUC across the five test partitions from the nested cross-validation procedure. We observed that a SVM with either a linear or a Gaussian kernel has worst performance than a SVM with the cross product kernels on fuzzy sets. Support vector machines with kernels on fuzzy sets are not only superior in terms of the AUC metric than the baseline case but they are more robust to the noise injection because the AUC metric remains somewhat constant. On the other hand, the SVM with the linear and Gaussian kernels suffer a decay in the AUC metric if the noise level is increased. Figures 2 from left to right the results for the clean train - noisy test, noisy train - noisy test, and noisy train - clean versions of the SONAR dataset, respectively. On all the cases we have that the cross

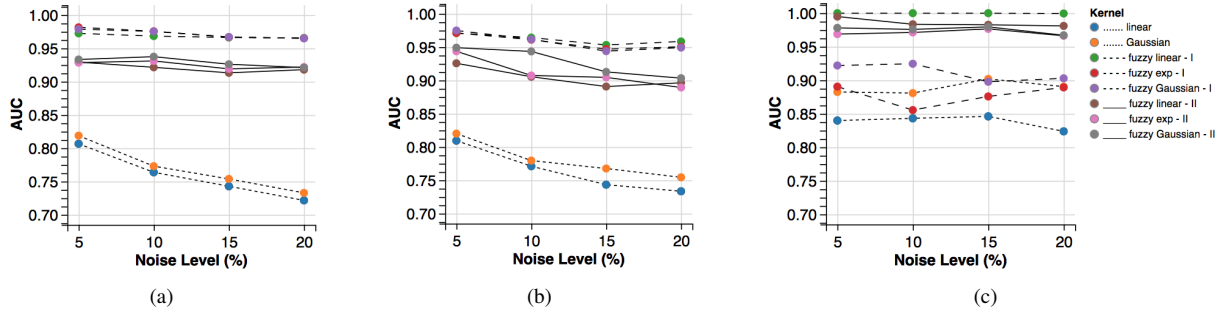


Fig. 1. Results for the PIMA attribute noise data set.

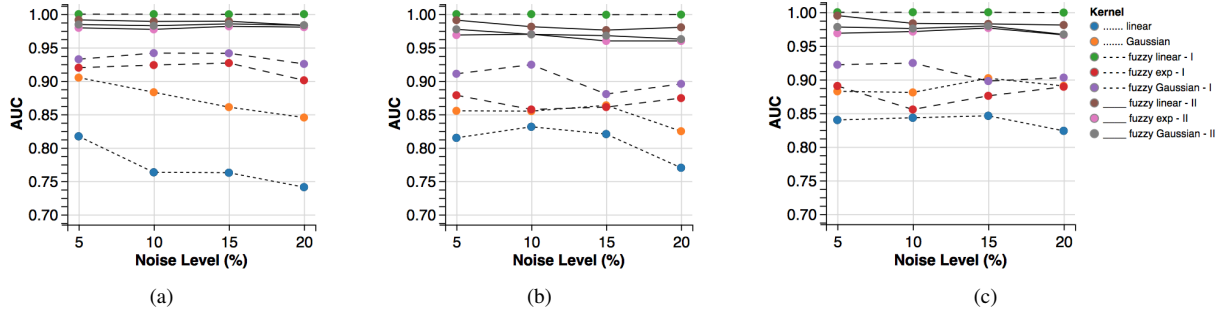


Fig. 2. Results for the SONAR attribute noise data set.

product kernel on fuzzy sets outperforms the baseline kernels. and it is more noise resistant.

VI. CONCLUSION

Cross product kernel on fuzzy gives a new approach to compute similarity measures between fuzzy sets. They allow a geometrical interpretation of similarity measures between fuzzy sets in RKHS's. Consequently, kernel algorithms like SVM's can embed the point-wise uncertainty of observations into the kernel definition via a kernel on fuzzy sets. Besides proving some properties of that kernel we empirically showed the practicality and power of those kernels through experiments on supervised classification on attribute noise datasets. Our findings suggest that those kernels not only outperform classical kernels but are more noise resistant.

ACKNOWLEDGMENT

The authors are thankful with FAPESP grant # 2011/50761-2, CNPq, CAPES, NAP eScience - PRP - USP for their financial support.

REFERENCES

- [1] R. Viertl, *Statistical Methods for Fuzzy Data*, ser. Wiley Series in Probability and Statistics. Wiley, 2011.
- [2] A. G. H. eorge A. Papakostas and V. G. Kaburlasos, Eds., *Handbook of Fuzzy Sets Comparison Handbook of Fuzzy Sets Comparison Theory, Algorithms and Applications Theory, Algorithms and Applications*. Gate to Computer Science and Research (GCSR), 2016.
- [3] J. Guevara, R. Hirata, and S. Canu, "Positive definite kernel functions on fuzzy sets," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, July 2014, pp. 439–446.
- [4] —, "Kernel functions in takagi-sugeno-kang fuzzy system with non-singleton fuzzy input," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, 2013, pp. 1–8.
- [5] —, "Fuzzy set similarity using a distance-based kernel on fuzzy sets," in *Handbook of Fuzzy Sets Comparison Theory, Algorithms and Applications*. Science Gate Publishing, 2016, vol. 6, pp. 103–120.
- [6] T. Gartner, *Kernels for structured data*. World Scientific, 2008, vol. 72.
- [7] D. Haussler, "Convolution kernels on discrete structures," Technical report, University of California at Santa Cruz, Tech. Rep., 1999.
- [8] R. L. Schilling, *Measures, integrals and martingales*. Cambridge University Press, 2005, vol. 13.
- [9] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups*. Berlin: Springer, 1984.
- [10] G. Loosli, S. Canu, and C. S. Ong, "Learning svm in krein spaces," *IEEE TPAMI*, vol. 38, no. 6, pp. 1204–1216, 2016.
- [11] B. Haasdonk, "Feature space interpretation of svms with indefinite kernels," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 4, pp. 482–492, 2005.
- [12] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *JMLR*, vol. 99, pp. 1517–1561, 2010.
- [13] J. Alcal-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. Garca, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.
- [14] X. Zhu, X. Wu, and Y. Yang, "Error detection and impact-sensitive instance ranking in noisy datasets," in *Proceedings of the 19th national conference on Artificial intelligence*. AAAI Press, 2004, pp. 378–383.
- [15] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2079–2107, 2010.