A kernel extension to handle missing data

Guillermo Nebot-Troyano and Lluís A. Belanche-Muñoz

Abstract An extension for univariate kernels that deals with missing values is proposed. These extended kernels are shown to be valid Mercer kernels and can adapt to many types of variables, such as categorical or continuous. The proposed kernels are tested against standard RBF kernels in a variety of benchmark problems showing different amounts of missing values and variable types. Our experimental results are very satisfactory, because they usually yield slight to much better improvements over those achieved with standard methods.

1 Introduction

In the last few years *kernel methods* have become a very popular topic of research. One of the most relevant problems in kernel-based learning machines, in terms of practical applications, is the *choice* of an appropriate kernel. This kernel should be a measure that adequately captures meaningful relations in the data. A proper kernel choice should result in more adequate learning machines, less likely to overfit and thus showing a better generalization ability.

Real-world data come from many different sources, described by mixtures of numeric and qualitative variables. These variables may require completely different treatments and are traditionally handled by *preparing* the data using a number of *coding methods*. These codings may entail an unknown change in input distribution or an increase in dimension, increasing the likelihood of overfitting and also the training or optimization time. Moreover, and most importantly, sometimes the data

Guillermo Nebot-Troyano
Faculty of Computer Science, Polytechnical University of Catalonia, Barcelona, Spain
e-mail: willynt@msn.com

Lluís A. Belanche-Muñoz (corresponding author)
Faculty of Computer Science, Polytechnical University of Catalonia, Barcelona, Spain
e-mail: belanche@lsi.upc.edu

sets exhibit *missing values* by diverse causes. These missing values are always a serious problem because they require a preprocessing (either a coding or an imputation) of the dataset in order to be able to use a classical kernel.

In this work we present a method for dealing with missing values that rigorously extends *any* kernel to one that copes with missing information and without the need of any coding or imputation mechanism. The method can make use of distributional or probabilistic assumptions about the variables. In the often encountered situation that this knowledge is not available, we advocate for the use of *sample* statistics (very much like in Naïve Bayes methods), in the form of density estimation or frequentist probabilities; contrary to other methods, no parametric knowledge is required. In addition, the proposed kernels can accept mixed data types, a common situation in real-world data. We present successful experimental results against standard RBF kernels in a variety of benchmark problems showing different amounts of missing values and different variable types.

2 Preliminaries

The Support Vector Machine (SVM) was developed by Vapnik and his coworkers, initially for classification problems and has won great popularity as a tool for the identification of nonlinear systems [16]. A nice introduction to SVMs and kernel machines is [5]. A key idea in kernel machines is that of the *kernel*, but the SVM formulation does not include criteria to select a kernel function. A standard result for identifying such functions can be derived from Mercer's result [10]:

Theorem 1. *A continuous and symmetric function* $K : \mathscr{H} \times \mathscr{H} \to \mathbb{R}$ *is a kernel if it satisfies the condition:*

$$\int_{\mathscr{H} \times \mathscr{H}} K(x,y) g(x) g(y) dx dy \geq 0$$

for any function g such that $\int_{\mathscr{H}} (g(x))^2 dx < \infty$

If the function K gives rise to a positive integral operator, its evaluation can be expressed as an absolutely and uniformly convergent series (finite or infinite), almost everywhere [10]. Except for specific cases, it may not be easy to check whether this condition is satisfied. For this reason we show another, equivalent, definition:

Theorem 2. *The function* $K : \mathscr{H} \times \mathscr{H} \to \mathbb{R}$ *is a kernel if and only if for any finite subset* $\{x_1, x_2, \ldots, x_n\} \in \mathscr{H}$ *the associated kernel matrix* $K_{n \times n} = (k_{ij})$*, where* $k_{ij} = K(x_i, x_j)$ *is a symmetric positive semidefinite (PSD) matrix.*

This condition is in general easier to check than Mercer's condition. Among the most widely used and well-known kernels we find the Polynomial kernel $K(u,v) = (< u,v > +\gamma)^d$ with $\gamma \geq 0 \in \mathbb{R}$ and $d \in \mathbb{N}$ parameters (where $<,>$ denotes scalar product) and the Gaussian kernel, one of a number of kernels known as Radial Basis

Function (RBF) kernels, $K(u,v) = exp(-\frac{||u-v||^2}{2\sigma^2})$, with $\sigma \in \mathbb{R}$ a parameter. This one is by far the most popular choice of kernel in SVMs; it also includes the polynomial kernel as a limiting case.

Kernel functions can be conceptually regarded as similarity functions [14], although not all kernels fulfill all the properties for a similarity (e.g. boundedness). The work of Gower in general similarity measures [7] shows some partial coefficients of similarity for three different types of features: Dichotomous (Binary), Qualitative (Categoric) and Quantitative (Continuous and Discrete) features, that are shown to produce PSD matrices; these functions can hence be seen as kernels. For any two observations $x_i, x_j \in \mathscr{H}$ to be compared on the basis of a feature k a *score* s_{ijk} is built: first δ_{ijk} is defined as 0 when the comparison of x_i, x_j cannot be performed on the basis of feature k for some reason (e.g., by the presence of missing values); δ_{ijk} is 1 when such comparison is meaningful. The coefficient of similarity between x_i, x_j is defined as the average score over all the partial comparisons:

$$S_{ij} = \frac{\sum_{k=1}^{n} s_{ijk}\delta_{ijk}}{\sum_{k=1}^{n} \delta_{ijk}}. \tag{1}$$

The scores s_{ijk} are defined as follows [7]:

i) *For Dichotomous (binary) features*: The presence of the feature is denoted by $+$ and its absence by $-$; negative matches (i.e., absence - absence) are not considered meaningful. When there are no missing values for feature k,

	Values			
observation x_i	$+$	$+$	$-$	$-$
observation x_j	$+$	$-$	$+$	$-$
s_{ijk}	1	0	0	0
δ_{ijk}	1	1	1	0

ii) *For Qualitative features*: Let $\mathbb{I}_{\{\cdot\}} = 1$ when the argument is true and 0 otherwise; then $s_{ijk} = \mathbb{I}_{\{x_{ik}=x_{jk}\}}$.

iii)*For Quantitative features*, $s_{ijk} = 1 - \frac{|x_{ik}-x_{jk}|}{R_k}$, where R_k is the *range* of feature k (the difference between the maximum and minimum attainable values).

Gower proves that, *if there are no missing values*, the matrix $S = (S_{ij})$ is PSD. This property may be lost when there are. An example will suffice: let $\mathscr{X}$ denote a missing value and consider three observations with four quantitative features in $[1,5]$ ($R_k = 4$), $x_1 = (1,2,3,1), x_2 = (1,3,3,\mathscr{X})$ and $x_3 = (1,3,3,5)$. In this case,

$$S = \begin{pmatrix} 1 & \frac{11}{12} & \frac{11}{16} \\ \frac{11}{12} & 1 & 1 \\ \frac{15}{16} & 1 & 1 \end{pmatrix}, \qquad \det(S) = -\frac{121}{2304} < 0$$

and therefore S is not PSD; but if we replace $\mathscr{X}$ by *any* precise value in $[1,5]$, then the matrix S is certainly PSD.

3 Main results

Missing information is an old issue in statistical analysis [9]. Missing values are very common in Medicine and Engineering, where many variables come from on-line sensors or device measurements, or are simply too costly to be measured at the same rate as other variables. In this section we present an approach that allows the *extension* of any kernel to one that is defined even in the presence of missing values. Moreover, the value returned by the kernels in this situation can be explained in meaningful terms. There are two basic ways of dealing with missing data, by *completing* the data description in a (hopefully) optimal way, or by *extending* the methods to work with incomplete descriptions. Our way to create kernels with missing values follows the latter idea and offers some important advantages:

1. Any kernel K can be extended to adapt to a dataset with missing values;
2. No preprocessing of the missing values is needed; we create kernels by calculating directly the values of $K(x,\mathscr{X})$ and $K(\mathscr{X},\mathscr{X})$ where $\mathscr{X}$ represents a missing value –behaving as an *incomparable* element w.r.t. any ordering relation– without the need to estimate the value of $\mathscr{X}$;
3. There is no need of removing information because of the missing values; i.e., no information is lost;
4. Missing values are allowed both in training and *test* examples (which is quite difficult with traditional imputation methods).

Lemma 1. *Let $\mathscr{H}$ any set, $x_1,x_2,\dots,x_n \in \mathscr{H}$ and let $f:\mathscr{H}\times\mathscr{H}\to\mathbb{R}$ a symmetrical function. Let $A\in\mathscr{M}_{n\times n}$ a PSD matrix where $A=[a_{ij}]$ with $a_{ij}=f(x_i,x_j)$. Let σ be any permutation of $x_1,\dots,x_n$, i.e., $\sigma(x_1,\dots,x_n)=(x_{\sigma(1)},\dots,x_{\sigma(n)})$; then the matrix $A^\sigma=[a^\sigma_{ij}]$ with $a^\sigma_{ij}=f(x_{\sigma(i)},x_{\sigma(j)})$ is PSD.*

Proof. Let A and A^σ be the matrices of the lemma and let σ any permutation of $x_1,\dots,x_n$, that is, $\sigma(x_1,\dots,x_n)=(x_{\sigma(1)},\dots,x_{\sigma(n)})$. In order for A^σ to be PSD, we must prove that $\forall z\in\mathbb{R}^n\ z^TA^\sigma z\geq 0$, provided $\forall y\in\mathbb{R}^n\ y^TAy\geq 0$.

Then $0\leq y^TAy=\sigma(y^T)\sigma(A)\sigma(y)=\sigma(y^T)A^\sigma\sigma(y)$, where $\sigma(y)=(y_{\sigma(1)},\dots,y_{\sigma(n)})$ and $\sigma(A)=[\sigma(a_{ij})]$, with $\sigma(a_{ij})=f(x_{\sigma(i)},x_{\sigma(j)})=a^\sigma_{ij}$; i.e., $\sigma(A)=A^\sigma$. Now we know that $\forall y\in\mathbb{R}^n$, $\sigma(y^T)A^\sigma\sigma(y)\geq 0$, that is the same that $\forall z\in\mathbb{R}^n\ z^TAz\geq 0$, because σ is a permutation function. □

This result is important and useful because if we prove that one matrix, that depends on a symmetrical function, is PSD for an arrangement of the dataset, then the matrix is PSD for any rearrangement (reordering of the observations) of it.

Theorem 3. *Let K be a kernel in a set $\mathscr{H}$ (e.g. a similarity function) and P a probability density function in $\mathscr{H}$. Then the function*

$$\hat{K}(x,y)=\begin{cases} K(x,y), & \text{if } x,y\neq\mathscr{X}\ ;\\ \int_{\mathscr{H}}P(y)K(x,y)dy, & \text{if } x\neq\mathscr{X} \text{ and } y=\mathscr{X};\\ \int_{\mathscr{H}}P(x)K(x,y)dx, & \text{if } x=\mathscr{X} \text{ and } y\neq\mathscr{X};\\ \int_{\mathscr{H}}P(x)\int_{\mathscr{H}}P(y)K(x,y)dydx, & \text{if } x=y=\mathscr{X} \end{cases}$$

is a kernel in $\mathscr{H} \cup \{\mathscr{X}\}$.

Proof. Developed in the Appendix, for clarity. □

Theorem 4. *Let K be a kernel in $\mathscr{H}$ (e.g. a similarity function) and P a probability mass function in $\mathscr{H}$. Then the function*

$$\hat{K}(x,y) = \begin{cases} K(x,y), & \text{if } x,y \neq \mathscr{X}\ ; \\ \sum_{y\in\mathscr{H}} P(y)K(x,y), & \text{if } x \neq \mathscr{X} \text{ and } y = \mathscr{X}; \\ \sum_{x\in\mathscr{H}} P(x)K(x,y), & \text{if } x = \mathscr{X} \text{ and } y \neq \mathscr{X}; \\ \sum_{x\in\mathscr{H}} P(x) \sum_{y\in\mathscr{H}} P(y)K(x,y), & \text{if } x = y = \mathscr{X} \end{cases}$$

is a kernel in $\mathscr{H} \cup \{\mathscr{X}\}$.

Proof. It is analogous to that of Theorem 3, changing the integrals by summations, since the summation has also the linearity property. □

3.1 Motivation of the extension

Given a two-place symmetric function $K : \mathscr{H} \times \mathscr{H} \to \mathbb{R}$, we aim to find that function κ that is the minimizer of

$$E[\kappa] = \int_{\mathscr{H}} \frac{1}{2} \int_{\mathscr{H}} (\kappa(z) - K(z,x))^2 p(z,x)\, dx\, dz$$

whose solution is $\kappa(z) = \int_{\mathscr{H}} K(z,x)p(x)\ dx$, by making use that, in the present situation, $p(z,x) = p(z)p(x)$. Therefore we define the kernel extension $\hat{K}(z,\mathscr{X}) = \kappa(z)$. The value of the kernel when *both* values are missing can be explained as follows. Focusing on one of the missing values, it certainly has to be one of the possible values, with some probability. Fixing it to, say, z, then the kernel has to be $K(z,\mathscr{X})$ by the previous result. The overall expression is therefore the *expectation* of $K(z,\mathscr{X})$ seen as a function of z.

3.2 Nonparametric Kernel Density estimation

If the densities or mass probability functions $f(x)$ are not known they can be estimated using the data set by applying non-parametric methods for estimation. One of these methods is the Parzen windows technique [11] or more generally *kernel density estimation* (KDE). A challenging task in the general case, in the univariate case the KDE approach is to consider $x_1,\ldots,x_n$ an i.i.d. sample of an absolutely continuous random variable X with unknown density $f(x)$, and define the empirical distribution function as $F_n(x) = n^{-1}\sum_{i=1}^{n} \mathbb{I}_{\{x_i \leq x\}}$, which is an estimator of the true (cumulative) distribution function $F(x)$ of X. Knowing that the density $f(x)$ is the derivative of the distribution function F we express $\hat{f}_h(x) = (2h)^{-1}[F_n(x+h) - F_n(x-h)]$, for a small $h > 0$. This is equivalent to the proportion of points in the interval $(x-h, x+h)$ divided by h. It is common that the amount of smoothing depends on the number of data points; then we have:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_n}\varphi\left(\frac{x-x_i}{h_n}\right) \tag{2}$$

A particular choice for the weight function (also called Parzen window or *uniform kernel*) is $\varphi(z) = \frac{1}{2}\mathbb{I}_{\{|z|\leq 1\}}$. Generally, φ and h must satisfy certain conditions of regularity, such that φ is bounded and absolutely integrable in $\mathbb{R}$ and integrates to 1 and $\lim\limits_{n\to\infty} h_n = 0$. Usually, $\varphi(z) \geq 0$ and $\varphi(z) = \varphi(-z)$. Among the most widely used kernels we also find the Gaussian or the Epanechnikov kernels [6]. If the bandwidth h is very small then the estimation of the density function degenerates to a collection of n spikes centered at the data points. If h is too big then the estimation is oversmoothed and tends to the uniform distribution. A typical choice is $h = h_0 n^{-1/2}$, where h_0 is a free parameter to be determined. This estimation is consistent and asymptotically normal [13]. In this work we use the bandwidth selection method using pilot estimation of derivatives, described in [15].

3.3 Extended kernel using uniform KDE

We illustrate the previous ideas by coupling the extended version of the kernels developed in section 2 with KDE. Let $H \in \mathbb{R}$ be any bounded subset and denote $b = \sup\limits_{x,y\in H} |x-y|$ and $a = \inf\limits_{x,y\in H} |x-y|$. According to Theorem 3, for any finite subset $\{x_1, x_2, \ldots, x_n\} \in H$,

$$\hat{K}_1(x_i, x_j) = \begin{cases} 1 - \frac{|x_i - x_j|}{b-a}, & \text{if } x_i, x_j \neq \mathscr{X}\ ; \\ g_1(x_i), & \text{if } x_i \neq \mathscr{X} \text{ and } x_j = \mathscr{X}; \\ g_1(x_j), & \text{if } x_i = \mathscr{X} \text{ and } x_j \neq \mathscr{X}; \\ G_1, & \text{if } x_i = x_j = \mathscr{X} \text{ and } i \neq j; \\ 1 & \text{if } x_i = x_j = \mathscr{X} \text{ and } i = j \end{cases}$$

is a valid PSD kernel, where

$$g_1(z) = \int_{-\infty}^{\infty} \hat{f}(x)\left(1 - \frac{|x-z|}{b-a}\right)dx = \int_{-\infty}^{\infty} \frac{1}{nh}\sum_{i=1}^{n}\varphi\left(\frac{x-x_i}{h}\right)\left(1 - \frac{|x-z|}{b-a}\right)dx$$

$$= \frac{1}{nh}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\varphi\left(\frac{x-x_i}{h}\right)\left(1 - \frac{|x-z|}{b-a}\right)dx = \frac{1}{nh}\sum_{i=1}^{n}\frac{1}{2}\int_{x_i-h}^{x_i+h}\left(1 - \frac{|x-z|}{b-a}\right)dx$$

$$= \frac{1}{2nh}\sum_{i=1}^{n}\alpha_i(z), \qquad \text{with } \alpha_i(z) = \begin{cases} \frac{2h(b-z+x_i-a)}{b-a}, & \text{if } z > x_i + h\ ; \\ \frac{2h(b-a)-(x_i-z)^2-h^2}{b-a}, & \text{if } x_i - h \le z \le x_i + h; \\ \frac{2h(b-x_i+z-a)}{b-a}, & \text{if } z < x_i - h \end{cases}$$

$$\text{and } G_1 = \int_{-\infty}^{\infty} \hat{f}(z) g_1(z)dz = \frac{1}{2nh}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\frac{1}{nh}\sum_{j=1}^{n}\varphi\left(\frac{z-x_j}{h}\right)\alpha_i(z)dz =$$

$$= \left(\frac{1}{2nh}\right)^2 \sum_{i=1}^{n}\sum_{j=1}^{n}\int_{x_j-h}^{x_j+h}\alpha_i(z)dz = \left(\frac{1}{2nh}\right)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}$$

with

$$\beta_{ij} = \begin{cases} \frac{4h^2(b-x_j+x_i-a)}{b-a}, & \text{if } x_i + h < x_j - h\ ; \\ \frac{12(b-a)h^2-(x_i-x_j)^3-2h(4h^2+3(x_i-x_j)^2)}{3(b-a)}, & \text{if } x_j - h \le x_i + h < x_j + h; \\ \frac{4h^2(3(b-a)-2h)}{3(b-a)}, & \text{if } x_j = x_i; \\ \frac{12(b-a)h^2+(x_i-x_j)^3-2h(4h^2+3(x_i-x_j)^2)}{3(b-a)}, & \text{if } x_j - h < x_i - h \le x_j + h; \\ \frac{4h^2(b-x_i+x_j-a)}{b-a}, & \text{if } x_j + h < x_i - h \end{cases}$$

3.4 Extended kernel for categoric features

Consider now a categoric feature that takes values in the finite set $\mathscr{V} = \{v_1, ..., v_l\}$. An extended kernel can be built around Gower's result for qualitative features (section 2). The probability mass function f for this type of feature can be estimated in the usual way from the data set by the frequency of every modality among the values that are non-missing for this feature. Then, for all $v_i, v_j \in \mathscr{V}$,

$$K_2(v_i, v_j) = \begin{cases} \mathbb{I}_{\{v_i = v_j\}}, & \text{if } v_i, v_j \ne \mathscr{X}\ ; \\ g_2(v_i), & \text{if } v_i \ne \mathscr{X} \text{ and } v_j = \mathscr{X}; \\ g_2(v_j), & \text{if } v_i = \mathscr{X} \text{ and } v_j \ne \mathscr{X}; \\ G_2, & \text{if } v_i = v_j = \mathscr{X} \text{ and } i \ne j; \\ 1 & \text{if } v_i = v_j = \mathscr{X} \text{ and } i = j \end{cases}$$

where $g_2(z) = \sum_{i=1}^{l} f(v_i)\mathbb{I}_{\{v_i = z\}} = f(z)$ and $G_2 = \sum_{i=1}^{l} f(v_i)^2$, is a PSD kernel in $\mathscr{V} \cup \{\mathscr{X}\}$.

3.5 Extended Heterogeneous Kernel

We show now how to create a full kernel in $\mathscr{H} = \mathscr{H}_1 \times ... \times \mathscr{H}_t$ from a collection of extended *partial* kernels K_i defined in the sets $\{\mathscr{H}_i\}_{i=1\div t}$.

Theorem 5. *If $\{K_i\}_{i=1\div t}$ are kernels defined in the sets $\mathscr{H}_i$, the function:*

$$\mathscr{K}(x,y) = \frac{1}{t}\sum_{i=1}^{t} K_i(x_i,y_i) \tag{3}$$

is a kernel in the product space $\mathscr{H}$.

Proof. The sum of $t > 0$ PSD matrices is a PSD matrix; take any real $r > 0$ and a PSD matrix A, then rA is again PSD (in the present case, $r = 1/t$). □

We will refer to (3) as an *Extended Heterogeneous Kernel* or EHK.

3.6 Adding flexibility to an EHK

Typically kernels have parameters that allow them to have a greater *flexibility*. In order to add this flexibility to an existing EHK, a non-linear *activation* function is needed, that depends on one parameter. Moreover, this activation function must preserve the PSD property.

Proposition 1. *Let K a Kernel in $\mathscr{H}$ and consider the function*

$$f_{act}(x) = \left(\frac{1}{1-\alpha x}\right)^{\frac{1}{\alpha}}$$

for any $\alpha \in (0,1)$. Then $f_{act}(K(x,y))$ is a kernel in $\mathscr{H}$.

Proof. Immediate using properties described in [4, 8]. □

We will refer to $f_{act}(K(x,y))$ as an EHK with parameter α or EHK$_\alpha$.

4 Experimental work

Experimental work is now presented in different benchmarking data sets: a specially designed synthetic data set, several problems from the UCI repository [2] and a couple of our own. We perform a comparative study between SVMs using two variants of RBF kernels (see below) and SVMs using the two EHK kernels[1].

4.1 Synthetic data

Our first problem has been created artificially for illustrative purposes. It consists of 11 features generated from known distributions, as indicated in Table 1.

[1] We used the R language for statistical computing [1] extended with the *kernlab* package.

Table 1 Probability distributions[a] and their parameters for the artificially generated problem.

Feature	1	2	3	4	5	6	7	8	9	10
Distrib.	Gau	Poi	Gmt	Unf	Unf	Exp	Gau	Gau	Bin	Ber
Params.	μ,σ^2	λ	p	a,b	a,b	λ	μ,σ^2	μ,σ^2	n,p	p
Value	$\mu=3$	$\lambda=3$	$p=0.6$	$a=-3$	$a=100$	$\lambda=4$	$\mu=0$	$\mu=0.5$	$n=20$	$p=0.28$
	$\sigma^2=0.5$			$b=10$	$b=200$		$\sigma^2=1$	$\sigma^2=2$	$p=1/3$	

[a] Gau=Gaussian, Poi=Poisson, Gmt=Geometric, Unf=Uniform, Exp=Exponential, Bin=Binomial, Ber=Bernoulli.

The eleventh feature is categoric with four equally-probable modalities (say A,B,C and D). The rules that set the class feature are as follows. Let v a vector instance of the data set and v_i stand for the value of its i-th feature; then

- **if** $v_1 > 2 \wedge v_2 \geq 1 \wedge v_3 < 4 \wedge v_4 > -2.4 \wedge v_5 \geq 103 \wedge v_6 \leq 1 \wedge v_7 \geq -1.9 \wedge v_8 < 4 \wedge v_9 \geq 4 \wedge v_{10} = 0 \wedge (v_{11} = \text{“}B\text{”} \vee v_{11} = \text{“}C\text{”})$ **then** the class is 1;
- **if** $v_1 < 3.8 \wedge v_2 \leq 6 \wedge v_3 \leq 2 \wedge v_4 \leq 9.4 \wedge v_5 < 196 \wedge v_6 > 0.01 \wedge v_7 \leq 2 \wedge v_8 \geq -3 \wedge v_9 \leq 8 \wedge (v_{11} = \text{“}A\text{”} \vee v_{11} = \text{“}D\text{”})$ **then** the class is 1;
- **otherwise** the class is -1.

We created random samples 500 instances each, and then introduced 5%, 10%, . . ., 85% of missing values, in steps of 5%. The aim is to ascertain how the methods can cope with the existence of an ever larger percentage of missing values. We use two methods to code missing values with the RBF kernel:

RBF1 missing values are imputed by mean or mode, depending on the feature being continuous or categoric.
RBF2 missing values are imputed by a zero and a new feature column is added with zeros; in the position of missing values, the zeros are replaced by ones.

In both methods, we code categorical attributes using a unary representation, a standard practice [12]. In Fig. 1 we see the results for the different methods. Each point is the mean of 50 different data sets. In each one, the methods were evaluated using 10 times of 10-fold cross-validation. EHK1 and EHKF1 represent the EHK and EHK$_\alpha$ kernels with the true density (or mass) function; EHK2 and EHKF2 represent the same kernels obtained using uniform KDE and frequentist probabilities.

We can see that the EHKF1 is the best method as could be expected, but EHKF2 is also quite good. For this reason, from here on, all the densities for numeric features are estimated using the kernel developed in section 3.3. Note also the drastic degradation of the RBF2 from 0% to 5%, probably due to the increment in input dimension (which only happens at this step). Also, at very high percentages (80% and more), all methods tend to perform as the baseline performance.

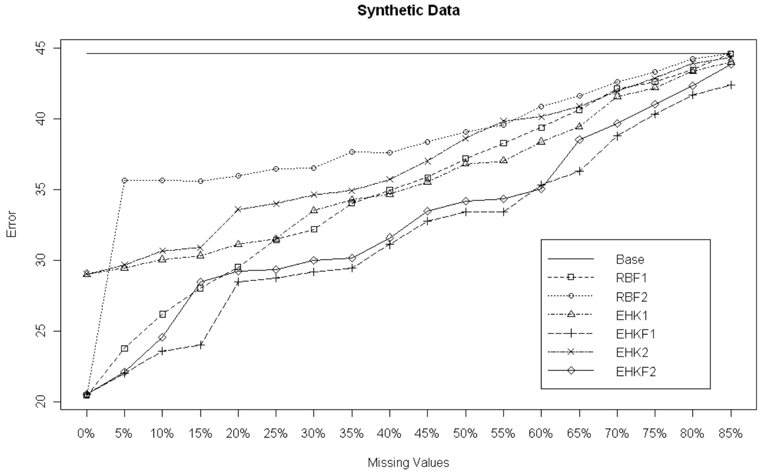

Fig. 1 Evolution of mean error (in %) in the synthetic problem for increasing percentages of missing values. The top horizontal line indicates the baseline performance using the majority class.

4.2 Real-world data sets

A description of the selected problems follows:

1. The CreditCRX data set (from UCI) has 690 instances and 15 features of which 9 are categoric and 6 are numeric. It contains a 0.65% of missing values.
2. The HorseColic data set (from UCI) has 366 instances and 22 features of which 12 are categoric, 7 are continuous and 3 are discrete. It contains a 23.75% of missing values. Note the original data set has 27 features; we have removed those numbered 3, 25, 26, 27 and 28 because they are declared as not relevant to the task. Further, two class features can be used: feature 23 (three possible cases: ’lived’, ’died’ and ’euthanized’) and 24 (the horse had surgical lesion or not).
3. The FecalSource data set has been donated by the Microbiology Department at the University of Barcelona. There are 144 instances with 10 dichotomous features, that are molecular tests signaling the presence of certain molecules in animal fecal samples. This dataset contains a 19.95% of missing values. The class feature has four possible cases: ’human’, ’bovine’, ’poultry’ and ’porcine’.
4. The Servo data set (from UCI) has 167 instances described by 4 many-valued categoric features. This data set does not contain missing values.
5. The Waste Water Treatment Plant (WWTP) data set has been donated by the Chemical Engineering Department at the University of Girona [3]. There are 279 instances and 91 continuous features that represent lagged information of plant process output. This dataset contains a 32.83% of missing values.

In Table 2 we can see the results obtained with the different methods. These are the results of parameter optimization (C,σ for the two RBFs, C for EHK or C,α

for EHK$_\alpha$) using again the mean of 10 times of 10-fold cross-validation. The ε parameter was also optimized in the regression tasks (SERVO and WWTP).

Table 2 Detailed results. In case of classification tasks these are the error rates in % and the 'Base' results correspond to 100% minus the majority class; in regression tasks these are *normalized root mean square errors* (NRMSE) and the 'Base' results correspond to the best constant model[a].

Problem/Method	Base	RBF$_1$	RBF$_2$	EHK	EHK$_\alpha$
CREDITCRX	44.49	13.80	14.09	12.81	12.54
HORSECOLIC-23	38.53	29.23	29.90	29.14	27.54
HORSECOLIC-24	36.96	16.50	18.89	15.95	15.47
FECALSOURCE	65.54	31.37	29.32	25.21	23.87
SERVO	1.000	0.406	0.406[b]	0.541	0.321
WWTP	1.000	0.456	0.531	0.396	0.395

[a] This corresponds to a NRMSE of 1.
[b] In the SERVO problem there are no missing values, thus both RBF methods coincide.

The two RBF methods do not seem to yield significant differences in performance. Given that the parameters have been fully optimized in both cases, this may indicate a lower bound in performance that cannot be surpassed with such direct ways of missing value treatment. On the other hand, the two EHK kernels behave comparably well, delivering better mean results, sometimes substantially, as in the FECALSOURCE problem. This problem is notoriously difficult, having four classes with less than 150 observations in total. It also seems that, as expected, the more flexible kernel EHK$_\alpha$ is able to achieve general better results.

5 Conclusions

This paper has presented a rigorous extension for univariate kernels that is able to deal with missing values. We would like to emphasize that we have advocated for the use of *partial* (or univariate) kernels for every descriptive variable and the building of a final kernel as the composition or *aggregation* of these partial kernels, an idea that can be traced back to Vapnik [17]. From the obtained results it can be concluded that the derived kernels have yielded satisfactory results. In the first place, our extended kernels behave very well when using the true densities, which provides empirical support for the theoretically developed ideas. Second, the extended kernels using non-parametric density estimation behave reasonably well and markedly better than standard kernels. This can be specially realized in the experiments with synthetic data. This of course is no proof that they are always a better choice, but adds strong support to the motivations of the work and to the solutions envisaged.

A recognized drawback of the work is the computational time, which we expect to improve in the future, by making more extensive use of incremental computations. A clear avenue for future research is the extension of the method for other data types for the features; for example, bit strings, fuzzy features, ordinal features, etc, could be accommodated with ease. We also envisage the extension of other kernels for complex data types already present in the literature (e.g., for trees or text).

Acknowledgements. Authors wish to thank the Spanish CICyT Project CGL2007-65980-C02-01/HID and the anonymous reviewers for their helpful suggestions.

Appendix

Proof for Theorem 3. If $M = [m_{ij}]$ is a $m \times n$ matrix whose elements are continuous functions in an interval, then the integral of M is again a $m \times n$ matrix whose elements are the integrals of the elements of M, that is to say:

$$\int_a^b M = [\int_a^b m_{ij}] \quad \text{where } a,b \in \mathbb{R}.$$

Suppose we have a finite sample $x_1,...,x_n \in \mathscr{H}$ of which k are non-missing values and $n-k$ are missing values. We order the sample so that the non-missing values go first and then come the missing ones, i.e., consider a permutation $\sigma(x_1,...,x_n) = (x_{m_1}, x_{m_2} ..., x_{m_k}, x_{m_{k+1}}, x_{m_{k+2}},..., x_{m_n})$, with $x_{m_1},\ldots,x_{m_k} \neq \mathscr{X}$ and $x_{m_{k+1}},\ldots,x_{m_n} = \mathscr{X}$. Then define $\mathscr{K} = [k_{ij}]$ with $k_{ij} = \hat{K}(x_i,x_j)$, $A = [a_{ij}]$ with $a_{ij} = K(x_{m_i},x_{m_j})$ and $A' = [a'_{ij}]$ with $a'_{ij} = \hat{K}(x_{m_i},x_{m_j})$. Hence,

$$A' = \left(\begin{array}{ccc|ccc} K(x_{m_1},x_{m_1}) & \ldots & K(x_{m_1},x_{m_k}) & \hat{K}(x_{m_1},x_{m_{k+1}}) & \ldots & \hat{K}(x_{m_1},x_{m_n}) \\ \vdots & \ddots & \vdots & \vdots & \ldots & \vdots \\ K(x_{m_k},x_{m_1}) & \ldots & K(x_{m_k},x_{m_k}) & \hat{K}(x_{m_k},x_{m_{k+1}}) & \ldots & \hat{K}(x_{m_k},x_{m_n}) \\ \hline \hat{K}(x_{m_{k+1}},x_{m_1}) & \ldots & \hat{K}(x_{m_{k+1}},x_{m_k}) & \hat{K}(x_{m_{k+1}},x_{m_{k+1}}) & \ldots & \hat{K}(x_{m_{k+1}},x_{m_n}) \\ \vdots & \ldots & \vdots & \vdots & \ddots & \vdots \\ \hat{K}(x_{m_n},x_{m_1}) & \ldots & \hat{K}(x_{m_n},x_{m_k}) & \hat{K}(x_{m_n},x_{m_{k+1}}) & \ldots & \hat{K}(x_{m_n},x_{m_n}) \end{array}\right)$$

$$= \left(\begin{array}{c|c} A'_1 & A'_2 \\ \hline A'_3 & A'_4 \end{array}\right) \qquad \text{where} \qquad A'_1 = \begin{pmatrix} K(x_{m_1},x_{m_1}) & \ldots & K(x_{m_1},x_{m_k}) \\ \vdots & \ddots & \vdots \\ K(x_{m_k},x_{m_1}) & \ldots & K(x_{m_k},x_{m_k}) \end{pmatrix},$$

$$A'_2 = \begin{pmatrix} \int_{\mathscr{H}} P(x_{m_{k+1}})K(x_{m_1},x_{m_{k+1}})dx_{m_{k+1}} & \ldots & \int_{\mathscr{H}} P(x_{m_n})K(x_{m_1},x_{m_n})dx_{m_n} \\ \vdots & \ldots & \vdots \\ \int_{\mathscr{H}} P(x_{m_{k+1}})K(x_{m_k},x_{m_{k+1}})dx_{m_{k+1}} & \ldots & \int_{\mathscr{H}} P(x_{m_n})K(x_{m_k},x_{m_n})dx_{m_n} \end{pmatrix},$$

$A'_3 = (A'_2)^T$ and

$$A_4' = \begin{pmatrix} \int_{\mathscr{H}} P(x_{m_{k+1}}) \int_{\mathscr{H}} P(x_{m_{k+1}}) K(x_{m_{k+1}}, x_{m_{k+1}}) dx_{m_{k+1}} dx_{m_{k+1}} & \dots & \int_{\mathscr{H}} P(x_{m_n}) \int_{\mathscr{H}} P(x_{m_{k+1}}) K(x_{m_{k+1}}, x_{m_n}) dx_{m_{k+1}} dx_{m_n} \\ \vdots & \ddots & \vdots \\ \int_{\mathscr{H}} P(x_{m_{k+1}}) \int_{\mathscr{H}} P(x_{m_n}) K(x_{m_n}, x_{m_{k+1}}) dx_{m_n} dx_{m_{k+1}} & \dots & \int_{\mathscr{H}} P(x_{m_n}) \int_{\mathscr{H}} P(x_{m_n}) K(x_{m_n}, x_{m_n}) dx_{m_n} dx_{m_n} \end{pmatrix}$$

An equivalent definition is $A' = \int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+1}})\; A\; dx_{m_{k+1}} \dots dx_{m_n}$, i.e.,

$$a'_{ij} = \int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+1}})\, a_{ij}\, dx_{m_{k+1}} \dots dx_{m_n} =$$

$$= \int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+1}}) K(x_{m_i}, x_{m_j}) dx_{m_{k+1}} \dots dx_{m_n}$$

because, if:

i) $x_{m_i}, x_{m_j} \neq \mathscr{X}$, then

$$a'_{ij} = \int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+1}}) K(x_{m_i}, x_{m_j}) dx_{m_{k+1}} \dots dx_{m_n} =$$

$$= K(x_{m_i}, x_{m_j}) \left(\int_{\mathscr{H}} P(x_{m_{k+1}}) dx_{m_{k+1}} \right) \dots \left(\int_{\mathscr{H}} P(x_{m_n}) dx_{m_n} \right) = K(x_{m_i}, x_{m_j})$$

ii) $x_{m_i} \neq \mathscr{X}$ and $x_{m_j} = \mathscr{X}$ where $j = k+1, ..., n$, then

$$a'_{ij} = \int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+1}}) K(x_{m_i}, x_{m_j}) dx_{m_{k+1}} \dots dx_{m_n} =$$

$$\left(\int_{\mathscr{H}} P(x_{m_{k+1}}) dx_{m_{k+1}} \right) \dots \left(\int_{\mathscr{H}} P(x_{m_j}) K(x_{m_i}, x_{m_j}) dx_{m_j} \right) \dots \left(\int_{\mathscr{H}} P(x_{m_n}) dx_{m_n} \right)$$

$$= \int_{\mathscr{H}} P(x_{m_j}) K(x_{m_i}, x_{m_j}) dx_{m_j}$$

iii) $x_{m_i} = x_{m_j} = \mathscr{X}$ where $i, j = k+1, ...n$, then

$$a'_{ij} = \int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+1}}) K(x_{m_i}, x_{m_j}) dx_{m_{k+1}} \dots dx_{m_n} =$$

$$\left(\int_{\mathscr{H}} P(x_{m_{k+1}}) dx_{m_{k+1}} \right) \dots \left(\int_{\mathscr{H}} P(x_{m_i}) \int_{\mathscr{H}} P(x_{m_j}) K(x_{m_i}, x_{m_j}) dx_{m_j} dx_{m_i} \right) \dots \left(\int_{\mathscr{H}} P(x_{m_n}) dx_{m_n} \right)$$

$$= \int_{\mathscr{H}} P(x_{m_i}) \int_{\mathscr{H}} P(x_{m_j}) K(x_{m_i}, x_{m_j}) dx_{m_j} dx_{m_i}$$

Now we are going to prove that A' is PSD. Using the last expression for A':

$$y^T A' y = y^T \left(\int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+2}}) \int_{\mathscr{H}} P(x_{m_{k+1}})\; A\; dx_{m_{k+1}} \dots dx_{m_n} \right) y$$

which, by the linearity of the integral, is equal to

$$\int_{\mathscr{H}} P(x_{m_n}) \dots \int_{\mathscr{H}} P(x_{m_{k+2}}) \int_{\mathscr{H}} P(x_{m_{k+1}}) \left(y^T A y \right) dx_{m_{k+1}} \dots dx_{m_n} \tag{4}$$

We know that $y^TAy \geq 0$ for all $y \in \mathbb{R}^n$, because K is a Kernel. The product of non-negative functions is non-negative and the definite integral of a non-negative function is non-negative. Therefore we have that $P(x_{m_{k+1}})y^TAy$ is a non-negative function because $P(x) \in [0,1]$ $\forall x \in \mathbb{R}$ and y^TAy is a non-negative function. Then

$$\int_{\mathscr{H}} P(x_{m_n})(y^TAy)dx_{m_n} \geq 0$$

In general we will have that $\int_{\mathscr{H}} P(x_{m_{k+1}})(y^TAy)dx_{m_{k+1}}$ is a non-negative function and $P(x_{m_{k+2}}) \geq 0$. Therefore,

$$\int_{\mathscr{H}} \left(P(x_{m_{k+2}}) \int_{\mathscr{H}} P(x_{m_{k+1}})y^TAydx_{m_{k+1}} \right) dx_{m_{k+2}} \geq 0$$

Iterating this argument we conclude that (4) is a non-negative function for all $y \in \mathbb{R}^n$ and consequently A' is PSD. By Lemma 1 $\mathscr{K}$ is PSD, and so K is a Kernel. □

References

1. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2008)
2. Asuncion, A. and Newman, D.J. (2007). UCI Machine Learning Repository, `http://www.ics.uci.edu/%7emlearn/MLRepository.html`. Irvine, CA: University of California, School of Information and Computer Science.
3. Belanche, LL., Valdés, J.J., Comas, J., Roda, I. and Poch, M. (1999) Towards a Model of Input-Output Behavior of Wastewater Treatment Plants using Soft Computing Techniques. *Environmental Modeling & Software*, 14: 409-419. Elsevier, 1999.
4. Berg, C., Christensen, J.P.R. and Ressel, P. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions.* Springer-Verlag, 1984.
5. Burges, J.C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2 (1998).
6. Duda, R. and Hart, P. *Pattern Classification and Scene Analysis*. Wiley (1973).
7. Gower, J.C. A general coefficient of similarity and some of its properties. *Biometrics*, 22, pp. 882-907 (1971).
8. Horn, R. and Johnson, C.R. *Matrix analysis.* Cambridge University Press, 1991.
9. Little, R.J.A. and Rubin, D.B. *Statistical analysis with missing data*. John Wiley, 1987.
10. Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A 209: 415-446.
11. Parzen, E. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3), pp. 1065-1076 (1962).
12. Prechelt, L. PROBEN1: A Set of Benchmarks and Benchmarking Rules for Neural Network Training Algorithms. Report 21/94. Fakultät für Informatik, Univ. Karlsruhe, 1994.
13. Rosenblatt, M. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3), pp. 832-837 (1956).
14. Schölkopf, B. *Learning with kernels*. John Wiley, 2001.
15. Sheather, S. J. and Jones, M.C. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series* B, 53, 683690, 1991.
16. Vapnik, V. *The nature of Statical Learning Theory*. Springer-Verlag, New York, 1995.
17. Vapnik, V. The support vector method of function estimation. *Neural networks and machine learning.* C. Bishop (Ed.), NATO ASI Series F. Springer, 1998.