An Introduction to Kernel Methods

Gustavo Camps-Valls
Universitat de València, Spain

Manel Martínez-Ramón
Universidad Carlos III de Madrid, Spain

José Luis Rojo-Álvarez
Universidad Rey Juan Carlos, Spain

INTRODUCTION

Machine learning has experienced a great advance in the eighties and nineties due to the active research in artificial neural networks and adaptive systems. These tools have demonstrated good results in many real applications, since neither *a priori* knowledge about the distribution of the available data nor the relationships among the independent variables should be necessarily assumed. Overfitting due to reduced training data sets is controlled by means of a regularized functional which minimizes the complexity of the machine. Working with high dimensional input spaces is no longer a problem thanks to the use of kernel methods. Such methods also provide us with new ways to interpret the classification or estimation results. Kernel methods are emerging and innovative techniques that are based on first mapping the data from the original input feature space to a kernel feature space of higher dimensionality, and then solving a linear problem in that space. These methods allow us to geometrically design (and interpret) learning algorithms in the kernel space (which is nonlinearly related to the input space), thus combining statistics and geometry in an effective way. This theoretical elegance is also matched by their practical performance.

Although kernels methods have been considered from a long time ago in pattern recognition from a theoretical point of view (see, e.g., Capon, 1965), a number of powerful kernel-based learning methods emerged in the last decade. Significant examples are Support Vector Machines (SVMs) (Vapnik, 1998), Kernel Fisher Discriminant (KFD), (Mika, Ratsch, Weston, Scholkopf, & Mullers, 1999) Analysis, Kernel Principal Component Analysis (PCA) (Schölkopf, Smola and Müller, 1996), Kernel Independent Component Analysis Kernel (ICA) (Bach and Jordan, 2002), Mutual Information (Gretton, Herbrich, Smola, Bousquet, Schölkopf, 2005), Kernel ARMA (Martínez-Ramón, Rojo-Álvarez, Camps-Valls, Muñoz-Marí, Navia-Vázquez, Soria-Olivas, & Figueiras-Vidal, 2006), Partial Least Squares (PLS) (Momma & Bennet, 2003), Ridge Regression (RR) (Saunders, Gammerman, & Vovk, 1998), Kernel K-means (KK-means) (Camastra, & Verri, 2005), Spectral Clustering (SC) (Szymkowiak-Have, Girolami & Larsen, 2006), Canonical Correlation Analysis (CCA) (Lai & Fyfe, 2000), Novelty Detection (ND) (Schölkopf, Williamson, Smola, & Shawe-Taylor, 1999) and a particular form of regularized AdaBoost (Reg-AB), also known as Arc-GV (Rätsch, 2001). Successful applications of kernel-based algorithms have been reported in various fields such as medicine, bioengineering, communications, data mining, audio and image processing or computational biology and bioinformatics.

In many cases, kernel methods demonstrated results superior to their competitors, and also revealed some additional advantages, both theoretical and practical. For instance, kernel methods (i) efficiently handle large input spaces, (ii) deal with noisy samples in a robust way, and (iii) allow embedding user knowledge about the problem into the method formulation easily. The interest of these methods is twofold. On the one hand, the machine-learning community has found in the kernel concept a powerful framework to develop efficient nonlinear *learning* methods, and thus solving efficiently complex problems (e.g. pattern recognition, function approximation, clustering, source independence, and density estimation). On the other hand, these methods can be easily used and tuned in many research areas, e.g. biology, signal and image processing, communica-

tions, etc, which has also captured the attention of many researchers and practitioners in safety-related areas.

BACKGROUND

Kernel Methods offer a very general framework for machine learning applications (classification, clustering regression, density estimation and visualization) over many types of data (time series, images, strings, objects, etc). The main idea of kernel methods is to embed the data set $S \subseteq X$ into a higher (possibly infinite) dimensional Hilbert space $\mathcal{H}$. The mapping of the data S into the Hilbert Space $\mathcal{H}$ is done through a nonlinear transformation $x \mapsto f(x)$. Thus, there will be a nonlinear relationship between the input data x and its image in $\mathcal{H}$. Then, one can use linear algorithms to detect relations in the embedded data that will be viewed as nonlinear from the point of view of the input data.

This is a key point of the field: using linear algorithms provides many advantages since a well-established theory and efficient methods are available. The mapping is denoted here by $\phi: X \rightarrow \mathcal{H}$, where the Hilbert space $\mathcal{H}$ is commonly known also as *feature space*. Linear algorithms will benefit from this mapping because of the higher dimensionality of the Hilbert space. The computational burden would dramatically increase if one needed to deal with high dimensionality vectors, but there is a useful trick (the *kernel trick*) that allows us to use kernel methods. As a matter of fact, one can express almost any linear algorithm as a function of dot products among vectors. Then, one does not need to work with the vectors once the dot products have been computed. The kernel trick consists of computing the dot products of the data into the Hilbert space $\mathcal{H}$ as a function of the data in the input space. Such a function is called a Mercer's kernel. If it is available, one can implement a linear algorithm into a higher (possibly infinite) Hilbert Space $\mathcal{H}$ without needing to explicitly deal with vectors in these space, but just their dot products. Figure 1 illustrates several kernel methods in the feature spaces. In Figure 1(a), the classical SVM is shown, which basically solves the (linear) optimal separating hyperplane in a high dimensional feature spaces. Figure 1(b) shows the same procedure for the KFD, and Figure 1(c) shows how a novelty detection (known as one-class SVM) can be developed in feature spaces.

The above procedures are done under the framework of the Theorem of Mercer (Aizerman, Braverman & Rozonoér, 1964). A Hilbert space $\mathcal{H}$ is said to be a Reproducing Kernel Hilbert Space (RKHS) with a Reproducing Kernel Inner Product K (often called RKIP or more commonly, Kernel) if the members of $\mathcal{H}$ are functions on a given interval T and if kernel K is defined on the product $T \times T$ having the properties (Aronszajn, 1950):

- for every $t \in T$, $K(\cdot,t) \in \mathcal{H}$, with value at $s \in T$ equal to $K(s,t)$.
- There is a reproducing kernel inner product defined as $(g, K(\cdot,t))_K = g(t)$ for every g in $\mathcal{H}$.

The Mercer's theorem states that there exist a function $\varphi: \mathbb{R}^n \rightarrow \mathcal{H}$ and a dot product $K(s, t) = \langle \varphi(s), \varphi(t) \rangle$ if and only if for any function $g(t)$ for which $\int g(t)dt < \infty$ the inequality $\int K(s, t)g(s)g(t)dsdt \geq 0$ is satisfied.

This condition is not always easy to prove for any function. The first kernels to be proven to fit the Mercer theorem were the polynomial kernel and the Gaussian kernel.

It is worth noting here that mapping ϕ does not require to be explicitly known to solve the problem. In fact, kernel methods work by computing the *similarity* among training samples (the so-called *kernel matrix*) by implicitly measuring distances in the feature space through the pair-wise inner products $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ between mapped samples $\mathbf{x}, \mathbf{z} \in X$. The matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ (where $\mathbf{x}_i, \mathbf{x}_j$ are data points) is called the *kernel matrix* and contains all necessary information to perform many (linear) classical algorithms in the embedding space. As we said before, a linear algorithm can be transformed into its non-linear version with the so-called *kernel trick.*

The interested reader can find more information about all these methods in (Vapnik, 1998; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). Among all good properties revised before, at present the most active area of research is the design of kernels for specific domains, such as string sequences in bioinformatics, image data, text documents, etc. The website www.kernel-machines.org provides free software, datasets, and constantly updated pointers to relevant literature.