KERNEL METHODS AND THEIR APPLICATIONS TO SIGNAL PROCESSING

Olivier Bousquet

Biological Cybernetics
Max Planck Institute
Spemannstr. 38
D-72076 Tübingen, Germany

Fernando Pérez-Cruz

Signal Processing and Communications
Universidad Carlos III de Madrid
Avda. Universidad 30
28911 Leganés (Madrid), Spain

ABSTRACT

Recently introduced in Machine Learning, the notion of kernels has drawn a lot of interest as it allows to obtain non-linear algorithms from linear ones in a simple and elegant manner. This, in conjunction with the introduction of new linear classification methods such as the Support Vector Machines has produced significant progress. The successes of such algorithms is now spreading as they are applied to more and more domains. Many Signal Processing problems, by their non-linear and high-dimensional nature may benefit from such techniques. We give an overview of kernel methods and their recent applications.

1. INTRODUCTION

Kernel-based algorithms have been recently developed in the Machine Learning community, where they were first introduced in the Support Vector Machine (SVM) algorithm [1]. There is now an extensive literature on SVM [2, 3] and the family of kernel-based algorithms [4]. The attractiveness of such algorithms stems from their elegant treatment of non-linear problems and their efficiency in high-dimensional problems.

They have allowed considerable progress in Machine Learning and they are now being successfully applied to many problems.

It is clear that many problems arising in Signal Processing are of statistical nature and require automatic data analysis methods. Moreover there are lots of non-linearities so that linear methods are not always applicable. Finally, the data is not always in vectorial form but is often sequential. All these reasons make kernel methods particularly suited for signal processing applications.

Another aspect is the amount of available data and the dimensionality. One needs methods that can use little data and avoid the curse of dimensionality. This is what Vapnik's approach aims at [2] and explains why using kernel methods and Vapnik's ideas may allow to efficiently handle data from signal processing problems.

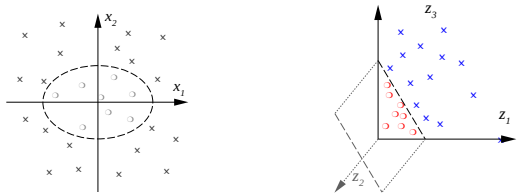
2. KERNEL METHODS

Many algorithms for data analysis are based on the assumption that the data can be represented as vectors in a finite dimensional vector space. These algorithms, such as linear discrimination, principal component analysis, or least squares regression, make extensive use of the linear structure. Roughly speaking, kernels allow to naturally derive non-linear versions of them.

2.1. The Kernel Trick

The general idea is the following. Given a linear algorithm (i.e. an algorithm which works in a vector space), one first maps the data living in a space $\mathcal{X}$ to a vector space $\mathcal{H}$ (the *feature space*) via a non-linear map $\Phi : \mathcal{X} \to \mathcal{H}$, and then runs the algorithm on the vector representation $\Phi(\mathbf{x})$ of the data. In other words, one performs non-linear analysis of the data using a linear method.

One of the purposes of the map Φ is to translate non-linear structures of the data into linear ones in $\mathcal{H}$. As an example, consider the following discrimination problem (see Figure 1) where the goal is to separate two sets of points. In the input space, the problem is non-linear, but after applying the transformation Φ which maps each vector to the three monomials of degree 2 formed by its coordinates, the separation boundary becomes linear. The gain of introducing the



(a) Input space $\mathcal{X}$ (b) Feature space $\mathcal{H}$

Fig. 1. Effect of the map $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

map Φ is not obvious yet. Indeed, we have just transformed

the data and we hope that in the new representation, linear structures will emerge. However, we may need to add a lot of dimensions to really make this happen and it may be hard to 'guess' the right Φ.

Here is where the so-called kernel comes into the game. We shall first restrict ourselves to algorithms that work in vector spaces endowed with an inner product. In this case, Φ has to map the input space to a Hilbert space.

If in the execution of the algorithms, only inner products between data vectors are considered, i.e. the data appears only in expressions like $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$, we can make use of the fact that for certain specific maps Φ this inner product can be computed directly from $\mathbf{x}$ and $\mathbf{x}'$ without explicitly computing $\Phi(\mathbf{x})$ and $\Phi(\mathbf{x}')$. This computational trick is termed the 'kernel trick'. More precisely, a kernel is a symmetric function of two variables that satisfies the following condition: for all $n \in \mathbf{N}$, and all $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, the kernel matrix, i.e. the matrix whose elements are $k(\mathbf{x}_i, \mathbf{x}_j)$ is positive semi-definite. The main property of functions satisfying this condition is that they implicitly define a mapping Φ from $\mathcal{X}$ to a Hilbert space $\mathcal{H}$ such that $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ and can thus be used in algorithms using inner products, introducing non-linearities via a straightforward modification.

2.2. Kernel Design

Because they correspond to inner products in some space, kernels can be considered as measures of similarity between two data points. Many different types of kernels are known [4] and among them, the most widely used are those operating on finite-dimensional vectors. For example the function $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$ trivially defines a kernel called the *linear kernel*. Another example is the polynomial kernel of degree d, defined as $k(\mathbf{x}, \mathbf{x}') = (a + \langle \mathbf{x}, \mathbf{x}' \rangle)^d$. Also the so-called Gaussian kernel with width σ, defined

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2) \,,$$

is very useful in practice. However, kernels are not limited to vector spaces and can be defined on graphs, on sequences, on groups, etc. (see [4]). This is a key feature of kernels: they allow to work in a simple (linear) way on data of various types (discrete or not, fixed or variable length).

3. KERNEL BASED ALGORITHMS

We will present two kinds of learning algorithms. The first ones, supervised learning algorithms, take as an input a set of *labeled examples*, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i$ are in some *input space* $\mathcal{X}$ and y_i are typically in $\mathbf{R}$, and they produce as an output, a function $f : \mathcal{X} \to \mathbf{R}$ which (hopefully) is able to predict the label y of new examples $\mathbf{x}$. The second ones, work on unlabeled examples ($\mathbf{x}_i$ only) and try to describe the structure of the data (e.g. its distribution).

3.1. Support Vector Machines

The SVM algorithm is used to perform binary classification ($y_i \in \{-1, 1\}$). The idea is to construct, in the feature space $\mathcal{H}$, a linear decision function from the hyperplane with maximum margin, i.e. which is at maximum distance from all the data points and classifies them correctly (see Figure 2). This corresponds to looking for a normal vector $\mathbf{w}$ and a parameter b corresponding to the hyperplane whose equation is $\mathbf{w} \cdot \Phi(\mathbf{x}) + b = 0$. The maximum margin hyperplane is obtained by minimizing $\|\mathbf{w}\|^2$ under constraints of correct classification of the data, i.e. $y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1$. It can be shown that the solution is obtained by solving the following dual problem (see e.g. [3] for details)

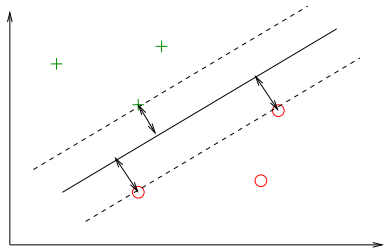


Fig. 2. Maximal margin hyperplane

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \,,$$

under constraints $\alpha_i \geq 0$. Using the extra constraint $\alpha_i \leq C$ yields the *soft-margin* SVM which allows some of the training data to be misclassified (for a more robust solution). Interestingly enough, this algorithm has an interpretation as a regularized optimization problem

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} \max(1 - f(\mathbf{x}_i) y_i, 0) + \lambda \|f\|^2 \,, \qquad (1)$$

where $\mathcal{H}$ is the space of functions generated by the kernel.

Other linear discrimination algorithms can be performed implicitly in feature space, such as Fisher discriminant analysis [5]. Also, other cost functions can be used in (1), such as the squared error $(f(\mathbf{x}_i) - y_i)^2$, yielding the kernel ridge regression algorithm which can be used to estimate a real-valued function.

3.2. Unsupervised Learning Algorithms

Principal Component Analysis (PCA) looks for directions of largest variance in the data. It turns out that PCA can be implicitly performed in feature space yielding kernel-PCA [6]. Figure 3 shows a toy example where PCA is performed both in the linear input space and in the feature space produced by the polynomial kernel. The result of kernel PCA

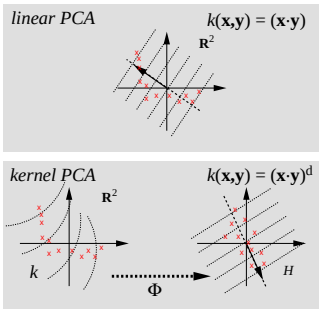


Fig. 3. Linear vs. kernel PCA

is to extract non-linear components which describe the non-ellipsoidal shape of the data cloud.

Another type of unsupervised learning algorithm is the so-called *one-class* SVM whose goal is to estimate the support of the data distribution. It works by applying a variant of the SVM algorithm with all y_i set to 1. The result is a non-linear boundary that encloses the data (or a prescribed fraction of the data). An important application is the problem of outliers or novelty detection [7].

Finally other types of algorithms that are often used in signal processing have been recently extended to the non-linear case via kernels. These include Vector Quantization [8], Independent Component Analysis [9, 10] and Canonical Correlation Analysis [11].

4. SIGNAL PROCESSING APPLICATIONS

Kernel methods have been applied to several signal processing and communications problems. Some of them are direct application of the standard SVM algorithm for detection or estimation and others incorporate prior knowledge into the learning process, either using virtual training samples or by constructing a relevant kernel for the given problem. The applications include speech and audio processing (speech recognition [12], speaker identification [13], extraction of audio features [14], audio signal segmentation [15]), image processing (face detection and recognition [16], image coding [8]) and communications (channel equalization [17, 18], non-stationary channel models [19], multi-user detection [20], signal classification [21]). This list is not exhaustive but shows the diversity of problems that can be treated by the techniques presented in previous sections.

4.1. Examples

Speech recognition has been usually solved applying Hidden Markov Models (HMM) trained using maximum likelihood estimates. The limitation of the approach is that the probability density function of the models is unknown and making assumptions about it can be difficult. It has been proposed [12] to incorporate the SVM into the HMM decision process [12], to be able to process the voice as it is generated but at the same time make the decisions according to the maximum margin principle that will ensure best separability under any density model.

Channel equalization and estimation is one of the key issues in digital communication because it involves linear and non-linear distortions and in many situations the training sequence need to be short in order not to reduce the payload bits. Indeed, communication channels introduce inter-symbol interference, i.e. each transmitted symbol is spread between some contiguous received symbols. Therefore a linear transversal filter, that contains several consecutive symbols, is used to estimate the incoming symbol. Least squares regression is frequently used to compute the weights of these filters. SVMs has been used with great success over non-linear channels directly using linear and non-linear kernels with very short training sequences [17] and some modifications using hidden Markov models have been also proposed [18]. One of the challenging issues in equalization is that the channel does not need to be stationary, so that the decision function has to change over time, without necessarily receiving a new training sequence. The SVM has been initially proposed for i.i.d. training sample, and has subsequently been modified for time varying communication channels by incorporating prior information about the time evolution of the channel in the cost function and margin [19].

An important feature of audio signals is that they carry information over time, which means that their amplitude at a given moment is less meaningful than the variations of this amplitude in time. It is thus of crucial importance, when applying data analysis techniques to signals, to represent them in an appropriate way. An approach based on kernels defined on time frequency representations (TFR) has been proposed [21] and promising results for signal classification and segmentation [15] were demonstrated.

4.2. Discussion and Perspectives

A large number of kernel methods applications to Signal Processing involve the use of the standard SVM algorithm with a Gaussian kernel applied to a vector representation of the data.

This surely yields a flexible and efficient method for classification. Indeed, the algorithm is simple, can be efficiently implemented and has few parameters (unlike e.g. neural networks). Moreover, the maximum margin principle allows to reduce the effective dimensionality of the problem, making generalization possible even with limited data. Finally, the fact that this method directly addresses the classification problem makes it really efficient in high dimensions compared to density estimation based approaches. This last remark is at the heart of Vapnik's philosophy and

has been fully demonstrated in applications such as speaker recognition [22] where previous approaches were based on probabilistic speaker models trained from the data. The guideline is thus: apply classification algorithms whenever possible.

So the easy things (proper reformulation of the problems and use of standard techniques) have been done, but now the focus should be on the kernels themselves which can really make a difference if correctly designed. Promising directions include the development of kernels for sequences or incorporating invariances [23] with respect to information-preserving signal transformations.

5. REFERENCES

[1] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Ann. Workshop on Computational Learning Theory*, D. Haussler, Ed. 1992, pp. 144–152, ACM Press.

[2] V. Vapnik, *Statistical Learning Theory*, J. Wiley, 1998.

[3] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[4] B. Schölkopf and A. Smola, *Learning with kernels*, MIT Press, 2002.

[5] S. Mika, A. J. Smola, and B. Schölkopf, "An improved training algorithm for kernel Fisher discriminants," In Jaakkola and Richardson [24], pp. 98–104.

[6] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., pp. 327–352. MIT Press, 1999.

[7] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Adv. in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. 2000, pp. 582–588, MIT Press.

[8] M. Tipping and B. Schölkopf, "A kernel approach for vector quantization with guaranteed distortion bounds," In Jaakkola and Richardson [24], pp. 129–134.

[9] F. Bach and M. Jordan, "Kernel independent component analysis," *J. of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[10] A. Gretton, R. Herbrich, and A. Smola, "Independence and kernel correlation," 2002, Preprint.

[11] M. Kuss and T. Graepel, "The geometry of kernel canonical correlation analysis," 2002, Preprint.

[12] N. Smith and M. Gales, "Speech recognition using SVMs," In *NIPS 14* [25].

[13] V. Wan and S. Renals, "Evaluation of kernel methods for speaker verification and identification," in *Proc. ICASSP 2002*, Orlando, FL, 2002.

[14] C.J.C. Burges, J.C. Platt, and S. Jana, "Extracting noise-robust features from audio data," in *Proc. ICASSP 2002*, Orlando, FL, 2002.

[15] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation," in *Proc. ICASSP 2002*, Orlando, FL, 2002.

[16] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 130–136.

[17] F. Pérez-Cruz, P. Alarcón-Diana, A. Navia-Vázquez, and A. Artés-Rodríguez, "SVC-based equalizer for burst TDMA transmissions," *Signal Processing*, vol. 81, no. 8, pp. 1681–1693, 2001.

[18] S. Chakrabartty and G. Cauwenberghs, "Sequence estimation qnd channel equalization using forward decoding kernel machines," in *Proc. ICASSP 2002*, Orlando, FL, 2002.

[19] F. Pérez-Cruz and A. Artés-Rodríguez, "Adaptive SVC for nonlinear channel equalization," in *Proc. of EUSIPCO'02*, Toulouse, France, 2002.

[20] S. Chen, A. K. Samingan, and L. Hanzo, "Support vector machine multiuser receiver for DS-CDMA signal in multipath channels," *IEEE Trans. Neural Networks*, vol. 12, no. 3, pp. 604–611, 2001.

[21] M. Davy, A. Gretton, A. Doucet, and P.J. Rayner, "Optimised support vector machines for nonstationary signal classification," Tech. Rep. 428, Cambridge University Eng. Dept., 2002.

[22] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. ICASSP 2002*, Orlando, FL, 2002.

[23] O. Chapelle and B. Scholkopf, "Incorporating invariances in non-linear SVMs," In *NIPS 14* [25].

[24] T. Jaakkola and T. Richardson, Eds., *Artificial Intelligence and Statistics*, Morgan Kaufmann, 2001.

[25] "Advances in neural information processing systems 14," MIT Press, 2001.