Kernel methods and scale invariance using the triangular kernel

Hichem Sahbi, François Fleuret

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Kernel methods and scale invariance using the triangular kernel

Hichem Sahbi — François Fleuret

N° 5143

March 2004

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--5143--FR+ENG

INRIA
ROCQUENCOURT

Kernel methods and scale invariance using the triangular kernel

Hichem Sahbi*, François Fleuret*

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet IMEDIA

Rapport de recherche n° 5143 — March 2004 — 25 pages

Abstract: We focus in this paper on the scale invariance of kernel methods using a particular function referred to as the triangular kernel. The study in [1] reported scale invariance for support vector machines (SVM) and the current work is an extension for support vector regression (SVR) and kernel principal component analysis (KPCA). First, we review these kernel methods and we illustrate analytically the scale invariance of the training processes. Experiments are conducted in several cases showing the scale invariance and the good performance in real pattern recognition problems including shape description, face detection and recognition.

Key-words: Statistical learning, kernel methods, scale-invariance, triangular kernel, shape description, face detection and recognition.

* IMEDIA Research Group-INRIA

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Méthodes à noyaux et invariance par échelle à l'aide du noyau triangulaire

Résumé : Ce rapport contient une étude analytique et expérimentale de l'invariance par échelle de quelques méthodes à noyaux en utilisant un noyau dit triangulaire. L'étude menée dans [1] concerne l'invariance par échelle pour les machines à vecteurs de supports (SVM) et ce travail est une extension pour la régression (SVR) et l'analyse en composantes principales (KPCA). Au départ, on rappel ces méthodes à noyaux et on illustre l'invariance par échelle à l'aide du noyau triangulaire. Une étude expérimentale montre les bonnes performances de généralisation et l'invariance par échelle pour des données synthétiques et pour des problèmes pratiques en reconnaissance des formes tels que la description des formes, la détection et la reconnaissance des visages.

Mots-clés : Apprentissage statistique, méthodes à noyaux, invariance par échelle, noyau triangulaire, description des formes, détection et reconnaissance des visages.

Contents

1 Introduction

Our definition of invariance means that applying a transformation on training and test examples will leave the response of a trained function on these examples unchangeable. The issue of invariance in machine learning has been tackled by several authors in the kernel machine community for local invariance [2] and more general linear and non-linear transformations [3, 4].

In many pattern recognition problems, invariance to transformations is achieved by enlarging the size of the training set and by adding instances of training examples in different viewing conditions [5]. [4] introduced the method of VSV (virtual support vectors) which considers that a transformation applied to data far from the margin will leave them non-support vectors, so it is interesting to apply a transformation only on the support vectors in order to build a set of VSV. Hence, adding this set to the original support vectors and training a new classifier on this new enlarged set, makes it possible to achieve the expected invariance. This process of increasing the size of the training problem makes the Gram matrix growing quadratically with respect to the size of the training set, so solving the underlying quadratic programming problem may be intractable.

In the same context of support vector classification (SVM), [6, 7] introduced a method to enforce the trained machine to be invariant to the targeted transformation by adding an orthogonality term in the objective function of the support vector machine minimization problem. The later finds the normal of the separating hyper-plan which is orthogonal as much as possible to the direction of the transformation in order to guarantee a negligible change in the value of the classification function.

Kernels are excellents tools for incorporating invariance [6]. The well studied Gaussian kernel achieves translation, rotation invariance and it is proved to perform well in practice even-though its infinite VC-dimension [8, 9]. This kernel is not scale invariant and it requires a careful estimation of its underlying scale parameter for samples generated according to a given probability distribution. This task is usually achieved either by minimizing a predictive bound on the generalization error of the related classifier [10, 11, 12] or by cross-validation. The later consists in training several times a classifier with different parameters and estimating its generalization error using a validation set. Nevertheless, for large size training problems such as face detection [13, 14], this can quickly get out of hand.

This paper is an analytical and experimental study of scale invariance of some kernel methods using the triangular kernel and its application to different pattern recognition problems. One can state that it is sufficient to rescale an original training set with respect to the radius of a bounding ball enclosing the data in order to achieve scale invariance. For instance [3] proposed such an approach which achieves this scale invariance in the context of support vector classification. The author provides conditions that L_1 and L_2 soft-margin SVM provides the same solution for different kernels including radial basis functions (RBFs)

and Neural networks on a training set at different scales. We will show in §4.1.2 that such an approach is not suitable in one of our pattern recognition problems.

Beside translation and rotation invariance of the triangular kernel, we show the scale invariance of support vector machine, kernel principal component analysis (KPCA) and support vector regression (SVR) using this kernel. In §2, we review these kernel methods and we demonstrate the scale invariance in §3. Then, we show in §4 rigorous evaluation of the generalization performance on synthetic toy examples and on real pattern recognition problems. We follow this section with a discussion, we conclude and we provide extensions for future work in §6.

In the remainder of this paper, we use the following notations: $\mathcal{X} \subset \mathbb{R}^n$ denotes an *input space* and $\mathcal{Y} \subset \mathbb{R}$ is the set of all possible *labels* of the data in $\mathcal{X}$. These labels are discrete values $\{-1, +1\}$ for classification and real values for regression. Let X and Y be two random variables standing respectively for the training examples and their labels and $\mathcal{S} = \left\{(x^{(i)}, y^{(i)}), i = 1, ..., N\right\}$ be a training set generated i.i.d (independently and identically distributed) according to a particular and may be unknown probability distribution $P(X, Y)$. Other notations will be introduced as we go along through different sections of this paper.

2 Kernel methods

2.1 Training and generalization

According to Tikhonov regularization (see for instance [15]), the general statement of learning consists in fitting a training set $\mathcal{S}$ with a function $f : \mathcal{X} \to \mathcal{Y}$ in order to minimize both an empirical risk c and a quadratic regularizer $g(\|.\|)$:

$$c(\mathcal{S}, f) \; + \; \beta \; g(\|f\|) \tag{1}$$

for some fixed ($\beta > 0$). The regularizer g can be $g(\|f\|) = \|f\|^2$ the norm in the reproducing kernel Hilbert space. In the case of support vector classification c can be a squared loss function $c(\mathcal{S}, f) = \frac{1}{N} \sum_{i=1}^{N} \left[y^{(i)} - sign(f(x^{(i)}))\right]^2$ where $sign$ is equal to $+1$ if the $f(x) \geq 0$ and -1 otherwise. The solution f of the above problem can be written as [16]:

$$f_\alpha(.) = \sum_{i=1}^{N} \alpha_i \; k(., x^{(i)}) \tag{2}$$

this is for a particular vector of coefficients $\alpha = (\alpha_1, ..., \alpha_N)$ referred to as the *training parameters*. $k(x, x^{'})$ is a symmetric, continuous on $\mathcal{X} \times \mathcal{X}$ and positive definite function, i.e., it satisfies the Mercer conditions (see for instance [17]). This function is commonly called kernel. Many kernels can be used [18, 19], the most standard being the Gaussian [20]:

$$k(x, x') \;=\; exp\left(-\|x - x'\|^2/\sigma^2\right) \tag{3}$$

The parameter σ in this kernel is directly related to *scaling*. If it is overestimated, the exponential behaves almost linearly and it can be shown that the projection into the high-dimensional space is also almost linear and useless [10]. On the contrary, when underestimated, the function lacks any regularization power, it is jagged and irregular, highly sensitive to noisy training data (cf. figure 2). Several methods have been developed in order to estimate an optimal σ, so that the whole process would be invariant to scaling [10, 20].

2.2 Support vector classification, regression and kernel PCA

2.2.1 Support vector classification

Given a training set $\mathcal{S}$ where the class labels take binary values $y^{(i)} = \pm 1$, the basic training of SVMs [21] is to find a mapping $x \mapsto y = f_\alpha(x)$ and a vector of parameters α that balances the empirical risk and the generalization error. When training examples are linearly separable, SVM finds a separating hyper plan $(w^t, b^t) \in \mathbb{R}^n \times \mathbb{R}$ which maximizes the margin subject to the fact that training examples of different class labels lie in different sides of the classification function. In the case when the training set is not linearly separable, "slack variables" $\xi = (\xi_1, ..., \xi_N)$ are defined as the amount by which training examples in $\mathcal{S}$ violate the constraint on the separation of the data. Hence, for a fixed $C \geq 0$ and $k \in \mathbb{N}^+$ the general form of the underlying minimization problem is:

$$\begin{array}{ll} \text{Minimize} & \\ F(w, b, \xi) = & \frac{1}{2}\|w\|^2 \;+\; C \left(\sum_i \xi_i\right)^k \\ \text{under} & \\ & y^{(i)} \left(w^t \; x^{(i)} \;+\; b\right) \;-\; 1 \;+\; \xi_i \;\geq 0, \;\forall\; i \\ & \xi_i \;\geq\; 0 \end{array} \tag{4}$$

Using Lagrange theory [22, 23], it can be easily shown [21] that the dual form of this problem is:

$$\begin{array}{lll} \text{Maximize} & & \\ L(\alpha) & = & \sum_i \alpha_i \;-\; \frac{1}{2} \sum_i \sum_j \; \alpha_i \; \alpha_j \; y^{(i)} \; y^{(j)} \langle x^{(i)}, x^{(j)} \rangle \\ \text{under} & & \\ & & 0 \;\leq\; \alpha_i \;\leq\; C \\ & & \sum_i \; \alpha_i \; y^{(i)} = 0 \end{array} \tag{5}$$

Here $\langle x^{(i)}, x^{(j)} \rangle$ denotes an inner product. Non linearly separable training data can be implicitly mapped into a high dimensional *feature space* via a mapping function $\Phi(x)$. SVM training can be performed just by replacing the inner product in the above equation with a positive definite kernel $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$.

2.2.2 Support vector regression

Support vector regression consists in finding a function as flat as possible which has at most ϵ deviation from the y-labels of the training data. In the linear case, the regression function $f_\alpha(x)$ is given by:

$$f_\alpha(x) = \langle w, x \rangle \ + \ b, \quad w \in \mathbb{R}^n, \ b \in \mathbb{R} \tag{6}$$

The hyper-plane w is found by minimizing the Euclidean norm regularizer $\|w\|^2$ subject to the fact that the y-labels of the training data are in a tube of radius ϵ around $f_\alpha(x)$, resulting in the following constrained minimization problem:

$$\begin{array}{ll} \text{Minimize} & \\ F(w,b,\xi) = & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{under} & \left\{ \begin{array}{lcl} y^{(i)} - \langle w, x^{(i)} \rangle - b & \leq & \epsilon + \xi_i \\ \langle w, x^{(i)} \rangle + b - y^{(i)} & \leq & \epsilon + \xi_i^* \\ \xi_i, \ \xi_i^* & \geq & 0 \end{array} \right. \end{array} \tag{7}$$

Here $C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$ is a penalty term which measures the amount of which a training example $x^{(i)}$ is outside the tube in one side or another depending on which of the two slack variables ξ_i or ξ_i^* is not zero. It can be shown [24] that the dual form of this constrained minimization problem corresponds to the following quadratic programming system:

$$\begin{array}{ll} \text{Maximize} & \\ L(\alpha,\alpha^*) = & -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x^{(i)}, x^{(j)} \rangle - \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y^{(i)}(\alpha_i - \alpha_i^*) \\ \text{under} & \left\{ \begin{array}{l} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i, \ \alpha_i^* \leq C \end{array} \right. \end{array} \tag{8}$$

where $\alpha_i, \ \alpha_i^*, \quad i = 1, ..., N$, are the dual Lagrange variables. Now, the regression function can be written as :

$$f_\alpha(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\langle x^{(i)}, x \rangle \ + \ b \tag{9}$$

As in support vector classification, the non-linear case can be handled using a positive definite kernel. For both the quadratic programming problem and the regression function, the inner product $\langle x, x^{'} \rangle$ is replaced with a kernel function $k(x, x^{'})$.

2.2.3 Kernel PCA

PCA is an unsupervised statistical analysis which provides a set of orthogonal axes in the feature space where the projection of a training set using few of these axes, hopefully makes it possible to capture most of the statistical variance of the data. In practice, PCA can be used in image processing, feature extraction, reconstruction, classification, etc [25].

Assuming centered training examples, i.e., $\sum_{i=1}^{N} x^{(i)} = 0$. In the linear case, PCA finds the principal axes by diagonalizing the covariance matrix $M = \frac{1}{N} \sum_{j=1}^{N} x^{(j)} {x^{(j)}}^t$ where x^t stands for the transpose of x. The principal orthogonal axes $\{V_i,\ i = 1, ..., \min(n, N)\}$ can be found by solving the following eigenproblem:

$$M\ V_i = \lambda_i\ V_i \tag{10}$$

where V_i and λ_i are respectively the i^{th} eigenvector and its underlying eigenvalue. It can be shown (see for instance [25]) that the solution of the above eigenproblem lies in the span of the training data, i.e.:

$$\forall i = 1, ..., \min(n, N), \quad \exists \alpha_{i1}, ..., \alpha_{iN} \in \mathbb{R} \quad \text{s.t.} \quad V_i\ =\ \sum_{j=1}^{N} \alpha_{ij}\ x^{(j)} \tag{11}$$

For the non-linear case, we consider in a similar manner a mapping Φ of the data from the input space $\mathcal{X}$ into a high dimensional feature space such that $k(x, x^{'}) = \langle \Phi(x), \Phi(x^{'}) \rangle$. Assuming that data are centered in this feature space, (i.e., $\sum_{i=1}^{N} \Phi(x^{(i)}) = 0$), the covariance matrix M and the general form of the eigenvectors can be rewritten respectively as:

$$M = \frac{1}{N} \sum_{j=1}^{N} \Phi(x^{(j)}) \Phi(x^{(j)})^t \tag{12}$$

$$V_i\ =\ \sum_{j=1}^{N} \alpha_{ij}\ \Phi(x^{(j)}) \tag{13}$$

where $\alpha = (\alpha_{i1}, ..., \alpha_{iN})$ are found by solving the following eigenproblem [25]:

$$K\ \alpha\ =\ \lambda\ \alpha \tag{14}$$

here K is the Gram matrix of the centered training set in the feature space.

3 Scale invariance

In this section, we describe the general form of the triangular kernel and the scale invariance of different training processes including SVM and KPCA.

3.1 The triangular kernel

The global form of the unrectified triangular kernel is:

$$k_T(x, x') \;=\; -\|x - x'\|^p, \quad p \in \mathbb{R} \tag{15}$$

This defines a conditionally positive definite kernel [26]. This means that for any $x^{(1)}, ..., x^{(N)}$ and any $c_1, ..., c_N \in \mathbb{R}$ such that $\sum_i c_i = 0$, we have $\sum_{i,j} c_i c_j k_T(x^{(i)}, x^{(j)}) \geq 0$. Due to the equilibrium constraint (cf. equations (5), (8)), this ensures that k_T can be used for support vector classification and regression [27].

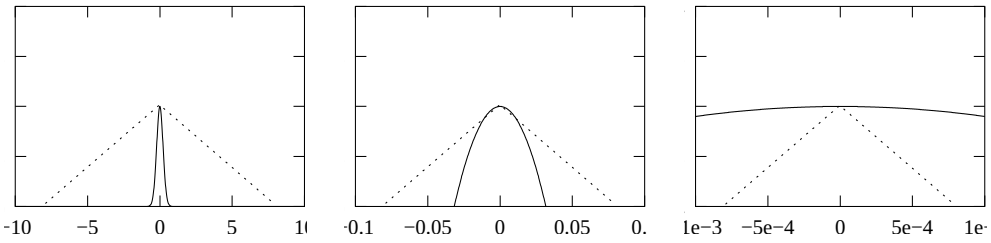

Figure 1: *Gaussian kernel (continuous line) and triangular kernel (dashed line) at various scales (left to right, respectively* $\times 10^0$, $\times 10^2$, $\times 10^4$*). Intuitively, whereas the triangular kernel has the same shape at all scales, the Gaussian kernel has different shapes, from a Dirac-like to a uniform weighting of the neighborhood.*

Since any Gram matrix built using this kernel is invertible for $0 < p < 2$ [28], it follows that this kernel has an infinite VC-dimension. Beside the affine invariance of this kernel (which is also achieved by the standard linear one), its discrimination power is high as the invertibility of the Gram matrix ensures that any training set can be approximated with a function with zero empirical error. This makes it possible, for instance in SVM, to separate any training set whatever its labeling.

Many generalization bounds [29] are proportional to the VC-dimension and this makes them pessimistic and useless for a class of functions with an infinite VC-dimension. Therefore these functions do not necessarily have bad generalization performances. In one hand, finite measures proportional to the cardinality of the training set can be used such as the *growth function* [30] which makes the generalization bounds tighter. On the other hand, many

experimental studies [20, 12] testify that a class of functions can perform well in practice even though trained using kernels with infinite VC-dimension (cf. §4.2).

3.2 Scaling of the triangular kernel

Even if the triangular kernel is not invariant to scaling, it still has an interesting weak property of invariance that we could describe as an invariance "in shape" (cf. figure 1, for $p = 1$). Given a scaling factor $\gamma > 0$, this weak invariance can be formally expressed as:

$$\begin{aligned} k_T(\gamma\ x, \gamma\ x') &= -\gamma^p\ \|x - x'\|^p \\ &= \gamma^p\ k_T(x, x') \end{aligned}$$

Thus, when the points are scaled by a certain factor γ, the value of the kernel scales by γ^p.

3.3 Invariance

In the following, we consider a situation where we scale the data by a factor $\gamma > 0$. Let's denote $\mathcal{S}^\gamma = \left\{\gamma\ x^{(1)}, ..., \gamma\ x^{(n)}\right\}$ a training set for that population. We denote f^γ, the function obtained after a training process on $\mathcal{S}^\gamma$ (thus, f^1 is the function built from the data at the original scale).

For both support vector classification and regression, our main interesting result to show is:

$$\forall x\ \in\ \mathbb{R}^n, \quad f^\gamma(\gamma\ x)\ =\ f^1(x) \tag{16}$$

while for kernel PCA we will show:

$$\forall\ k = 1, ..., N, \quad V_k^{(1)} = V_k^{(\gamma)} \tag{17}$$

here $\{V_1^{(1)}, ..., V_N^{(1)}\}$ and $\{V_1^{(\gamma)}, ..., V_N^{(\gamma)}\}$ denote the eigenvectors of respectively the original and the scaled training sets. We will show the validity of (16) only in the case of support vector classification and the proof can be derived in a similar way for regression.

Support vector classification: Let α^γ, ω^γ and b^γ be the parameters of the classification function estimated on $\mathcal{S}^\gamma$, we have:

$$f^\gamma(x)\ =\ \sum_i \alpha_i^\gamma\ y^{(i)}\ k_T(\gamma\ x^{(i)}, x)\ +\ b^\gamma \tag{18}$$

The α_i^γ come from the minimization problem related to $\mathcal{S}^\gamma$:

$$\begin{array}{lll} \text{Maximize} & & \\ L^\gamma(\alpha_i^\gamma) & = & \sum_i \alpha_i^\gamma - \frac{1}{2} \sum_{i,j} \alpha_i^\gamma \; \alpha_j^\gamma \; y^{(i)} \; y^{(j)} \; k_T(\gamma \; x^{(i)}, \; \gamma \; x^{(j)}) \\ \text{under} & & \alpha_i^\gamma \geq 0 \\ & & \sum_i \alpha_i^\gamma \; y^{(i)} = 0 \end{array} \tag{19}$$

It follows:

$$\begin{aligned} L^\gamma(\alpha_i^\gamma) &= \sum_i \alpha_i^\gamma - \frac{\gamma^p}{2} \sum_{i,j} \alpha_i^\gamma \; \alpha_j^\gamma \; y^{(i)} \; y^{(j)} \; k_T(x^{(i)}, x^{(j)}) \\ &= \frac{1}{\gamma^p} \left(\sum_i \gamma^p \; \alpha_i^\gamma - \frac{\gamma^p}{2} \sum_{i,j} \gamma^p \; \alpha_i^\gamma \; \gamma^p \; \alpha_j^\gamma \; y^{(i)} \; y^{(j)} \; k_T(x^{(i)}, x^{(j)}) \right) \\ &= \frac{1}{\gamma^p} \; L^1(\gamma^p \; \alpha^\gamma) \end{aligned} \tag{20}$$

this leads to: $\forall i, \; \alpha_i^\gamma = \frac{1}{\gamma^p} \alpha_i^1$ and to the following equality, $\forall \; x$:

$$\begin{aligned} \sum_j \alpha_j^\gamma \; y^{(j)} \; k_T(\gamma \; x, \gamma \; x^{(j)}) &= \sum_j \frac{1}{\gamma^p} \; \alpha_j^1 \; y^{(j)} \; \gamma^p \; k_T(x, x^{(j)}) \\ &= \sum_j \alpha_j^1 \; y^{(j)} \; k_T(x, x^{(j)}) \end{aligned} \tag{21}$$

Thus, we can easily show that $b^\gamma = b^1$, so we obtain our main result:

$$\begin{aligned} f^\gamma(\gamma \; x) &= \sum_i \alpha_i^\gamma \; y^{(i)} \; k_T(\gamma \; x^{(i)}, \gamma \; x) \; + \; b^\gamma \\ &= \sum_i \alpha_i^1 \; y^{(i)} \; k_T(x^{(i)}, x) \; + \; b^1 \\ &= f^1(x) \; \square \end{aligned} \tag{22}$$

Kernel PCA: The proof is straightforward, and comes from the fact that the Gram matrix K^γ of the scaled set can be written as:

$$K^\gamma = \gamma^p \; K^1 \tag{23}$$

Using (14) it follows that:

$$\begin{array}{rll} & K^\gamma \; \alpha^\gamma & = \; \gamma^p \; (K^1 \; \alpha^1) \\ \Rightarrow & \lambda^\gamma \; \alpha^\gamma & = \; \gamma^p \; \lambda^1 \; \alpha^1 \end{array} \tag{24}$$

which implies: $\lambda^\gamma = \gamma^p \; \lambda^1 \quad and \quad \alpha^\gamma = \alpha^1$, so from (13) $\forall \; k = 1, ..., N, \quad V_k^{(1)} = V_k^{(\gamma)} \; \square$.

4 Applications

4.1 Scale invariance

In this section, we will show the scale invariance property for KPCA and SVM on both synthetic and real pattern recognition problems.

4.1.1 The chess-board

To illustrate the scale invariance of SVMs and their generalization performance using the triangular kernel, we have set up a simple classification task in two dimensions. The original training population is a set of 512 points, uniformly distributed in the unit square. The class of each of those samples is a deterministic function of their location in the square, i.e., $\exists\ g\ :\ [0,1]^2 \longrightarrow \{-1,+1\}$ such that $Y = g(X)$. (see figure 2, upper row.)

From this sample, we have produced two others, one scaled down by a factor of 10, and the other scaled up by the same factor. We have built three SVMs based on a Gaussian kernel with $\sigma = 0.2$ on those three samples, and three SVMs based on the triangular kernel. Results are shown in figure 2. As expected the Gaussian kernel either smooths too much (middle row, left), is accurate (middle row, center) or overfits (middle row, right), while the triangular kernel behaves similarly at all scales.

4.1.2 Shape description

One interesting application in image retrieval is shape description. Usually objects in images can be represented by their contours and can be used for retrieval. Different signatures exist in the literature for shape retrieval among them the well studied edge orientation histogram, Radon transform, invariant moments, etc. [31]. For this application, the use of kernel PCA on the x and y coordinates of points belonging to a curve makes it possible to have an affine invariant description of its shape. Let $\mathcal{S}$ be a training set containing samples of 2D points from a curve, using KPCA transform and according to equation (24) any combination of translation and rotation will leave the eigenvalues of the KPCA transform unchangeable. Only scaling the data with a factor γ scales the eigenvalues by γ^p. Thus, these eigenvalues can be normalized with respect to their largest value so they can also be scale invariant and can be used as an efficient description of a curve. Notice that the largest eigenvalues provide us with the global shape (height, elongation, etc) of the curve while the smallest eigenvalues provide us with details (noise, fluctuations, etc.)

These eigenvalues have been successfully used to describe curves of different fish shapes. Indeed, we ran our KPCA on the SQUID[1] database [32] (see. figure 3) consisting of 1100 curves where the number of points ranges from 400 to 1600. Each curve is randomly sampled in order to extract 128 $2D$ training points which are used to synthesize 4 others curves with random orientations (in $[0^o, 360^o]$), scale factors (in $[0, 2]$) and locations (in ± 20 pixels). Thus, a total of 5500 curves are used for retrieval and each one is used to evaluate the

[1] www.ee.surrey.ac.uk/Research/VSSP/imagedb/squid.htm

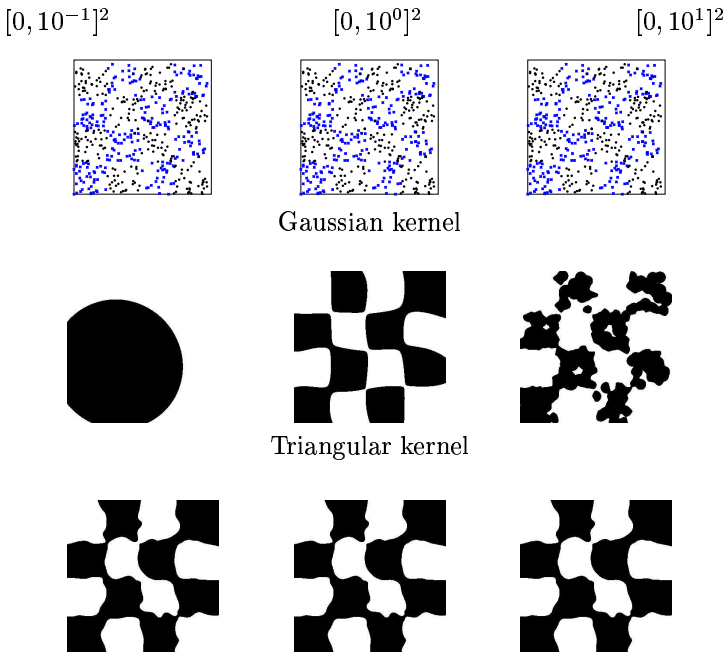

Figure 2: *A simple classification task in* $2D$*. The upper row shows the training set scaled by three different factors. The figures are zoomed according to the same factors for ease of visualization. The middle row shows the results of the classification with a Gaussian kernel* ($\sigma = 0.2$)*, and the lower row shows results with the triangular kernel.*

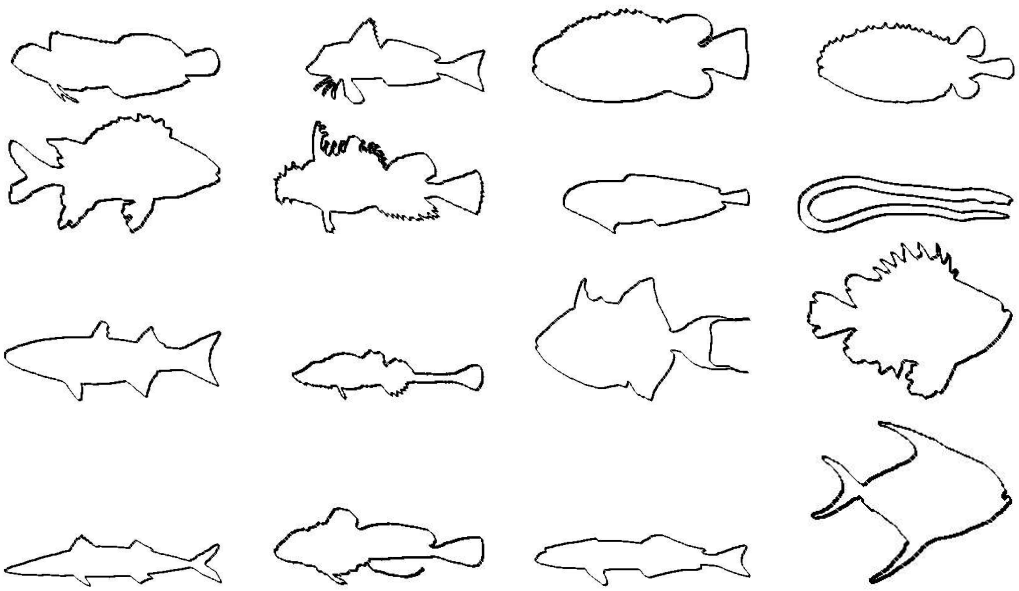

Figure 3: *Some fish contours randomly selected from the SQUID dataset.*

eigenvalues of the underlying Gram matrix. A curve is characterized by its 10 highest eigenvalues, so the remaining eigenvalues are skipped in order to avoid the noise effects. These 10 eigenvalues define our description space and the L_2 distance is used for retrieval. We can see in figure (4) the retrieval process and the robustness of the description to scaling, rotation and rotation effects. Indeed for each submitted shape, we find first the 4 most similar shapes which differ only by affine transformations, then we find the other similar cuves.

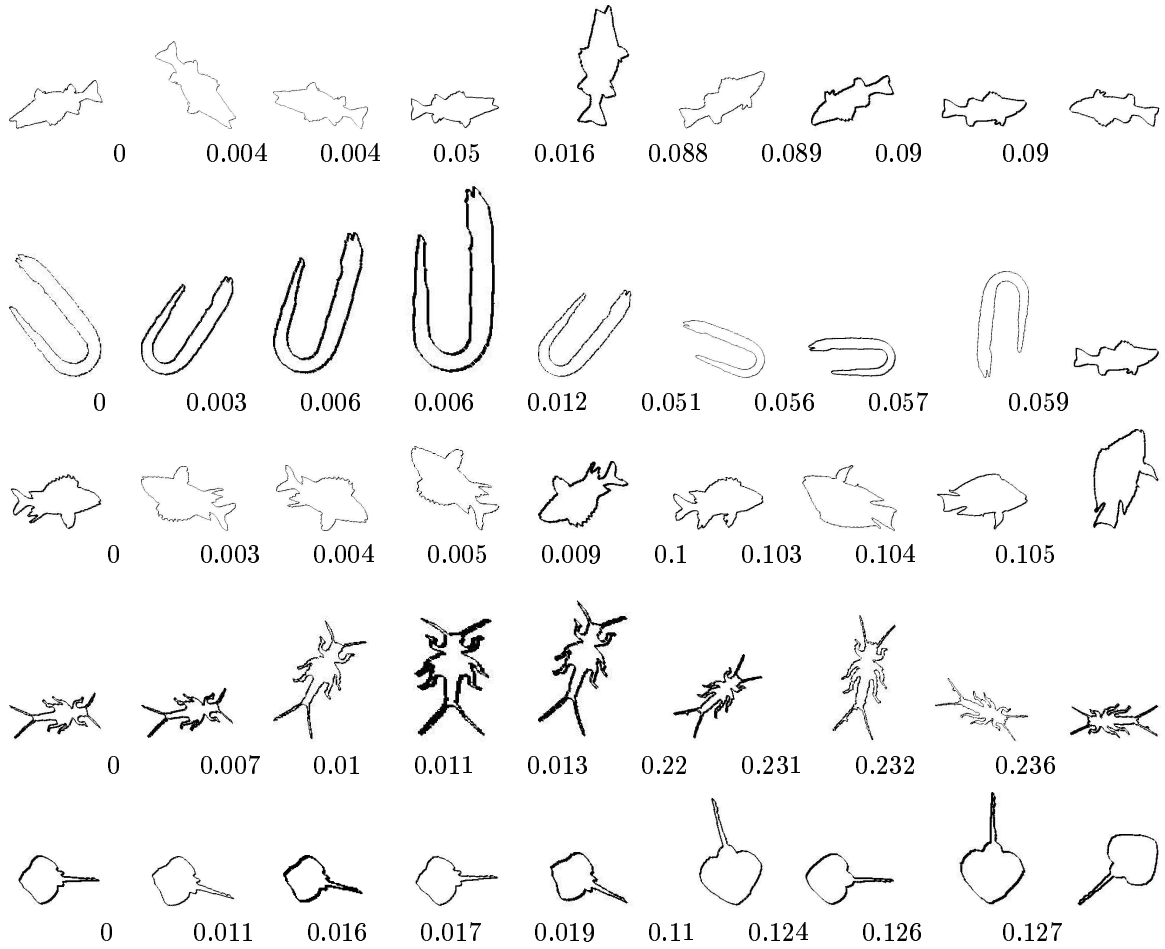

Figure 4: *Left images are query shapes while the others are some results sorted from left to right according to their dissimilarity.*

The statement provided in the introduction about the use of a normalization factor in order to achieve scale invariance is not suitable in this application as done in [3]. The

approach in [3] makes KPCA scale invariant when using other kernels such as the Gaussian, however each curve requires an appropriate selection of the best variance parameter σ using cross validation. This leads into different σ and feature spaces for different curves, so it may be meaningless to compare the underlying eigenvalues.

4.2 Generalization

In this section, we show the generalization performance of KPCA and SVM on synthetic and practical problems including face detection, recognition and handwritten character recognition. The triangular kernel is compared with respect to others such as the Gaussian and the linear kernels.

4.2.1 Alternating circles

Let's $C_i = (c_i, r_i)$ denotes a circle centered on c_i, with a radius r_i. In these experiments, we generate a set of positive and negative 2D points by respectively sampling N circles of radius $r_0, 10\ r_0, ..., 10^i\ r_0, ..., 10^{N-1}\ r_0$ and N circles of radius twice the radius of the previous ones i.e., $2\ r_0, 2 \times 10\ r_0, ..., 2 \times 10^i\ r_0, ..., 2 \times 10^{N-1}\ r_0$ where $N = 8$ in practice. As already expressed, the radiuses of circles belonging to the same class are multiple of 10 (cf. figure 5) and all the circles are centered on the same point. Training and test sets are randomly generated from these circles, each of the two sets contains 200 examples.

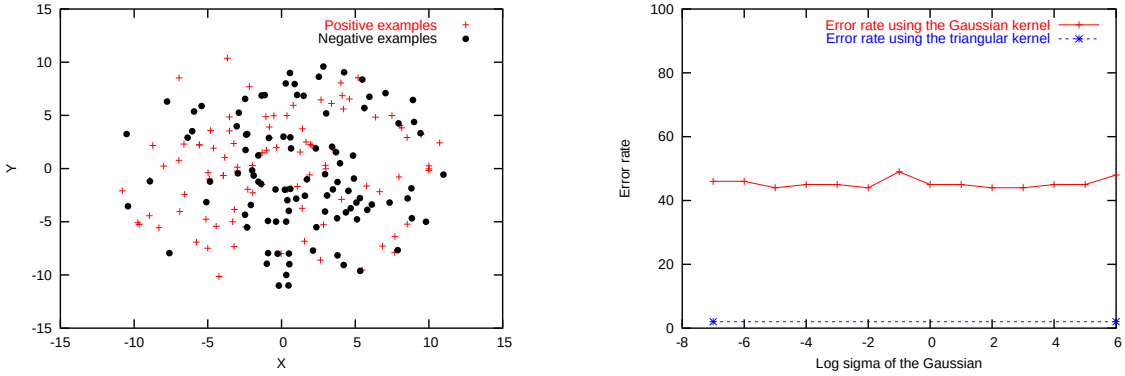

Figure 5: *(Left) Positive and negative examples (shown with two different colors) sampled randomly from the circles $C_0, ..., C_7$. In this picture, the radiuses of the sampled circles are considered in the logarithmic scale. (Right) Classification error using the triangular and the Gaussian kernels. This error is, of course, independent from the variance in the case of the triangular kernel.*

In order to show the performance of support vector classification on this particular task, we trained several SVMs using the triangular and the Gaussian kernels with various values

of σ. Performances are depicted in figure 5 according to the variance of the Gaussian (shown in a logarithmic scale). This diagram shows clearly the out-performances of the triangular kernel with respect to the Gaussian.

4.2.2 Handwritten character recognition

This experiment is a classical problem of handwritten digit recognition on the MNIST database [33]. This database contains 70.000 black and white digit pictures of size 28×28 pixels (see. figure 6). The features we use for that experiment are 64 low frequency Haar-wavelet coefficients, similar to the ones used for the face detection experiment (cf. §4.2.3). We train ten SVMs, $f^{(0)}, \ldots, f^{(9)}$, each one dedicated to one of the digits. The training for each of them is done on 60.000 examples and the testing is done on 10.000 other images. For each picture, we consider as features 64 simple Haar wavelet coefficients to gain local invariance to deformations. The final classifier F is based on a winner-take all rule: the result of the classification is the index of the SVM with the highest response.

$$F(x) = \arg\max_i \ f^{(i)}(x) \tag{25}$$

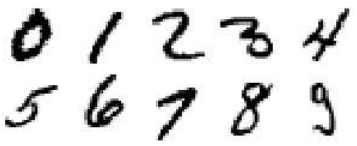

Figure 6: *Handwritten digits from the MNIST dataset.*

Results are shown on table (1) for the Gaussian kernel at various σ and for triangular kernel.

Table 1: *Performance comparison between the triangular and the Gaussian kernel on handwritten digit recognition.*

Kernel	Error rate
Triangular	3.93 %
Gaussian ($\sigma = 10^{-1}$)	35.87 %
Gaussian ($\sigma = 1$)	5.18 %
Gaussian ($\sigma = 10$)	6.89 %
Gaussian ($\sigma = 100$)	20.68 %

4.2.3 Face detection

The initial motivation for this study was to understand the good generalization performance of the triangular kernel in the context of face detection. [34] have developed a highly efficient detector based on a hierarchy of SVMs using the triangular kernel. Their approach consists in building several SVMs dedicated to population of face pictures more and more constrained in position in the image plan. We focus here on the generalization performances of individual classifiers dedicated to constrained populations of face pictures. Figure (7) shows some examples from two of them, the first less constrained than the second. Both are synthetically generated by doing affine bitmap transformations of the original pictures which are taken from the Olivetti database of faces [35]. Each picture is a 64×64 pixel in 256 gray levels and contains a face roughly centered. The distance between the eyes of each face ranges from 10 to 20 pixels. We use as a facial description a vector of 256 Haar wavelet coefficients. These simple Haar coefficients allow us to capture the main facial details at various orientations and resolutions and can be computed efficiently using the integral image [36].

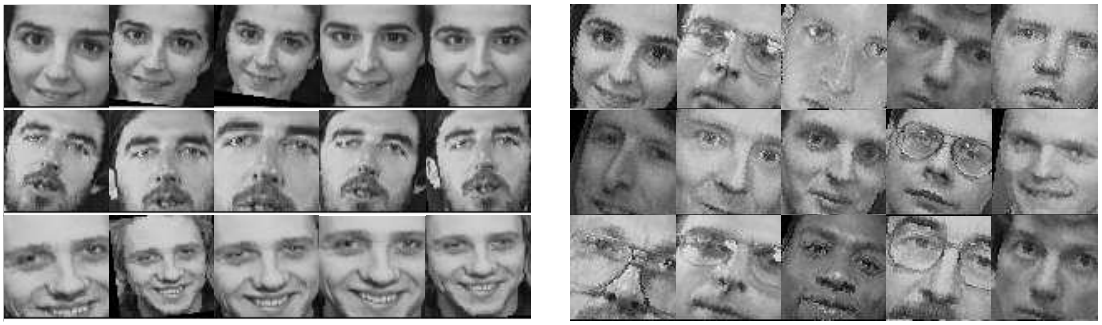

Figure 7: *Some face examples from, respectively, the least (left images) and the most (right images) constrained pose set in the hierarchy.*

The results given here correspond to SVMs trained with 400 face pictures and 600 background images. Error rates are estimated on 400 other face pictures, verifying the same pose constraints, and 600 other background pictures. As expected, the more the faces are constrained in pose, the easier is the classification, since tolerance to translation and rotation is no more expected from the SVM.

Results on table 2 show the performance of both the triangular and the Gaussian kernels. While the Gaussian kernel relies heavily on the choice of the scale σ, the triangular kernel achieves the same order of performances without tuning of any scale parameter.

Table 2: *Performance comparison between the triangular and the Gaussian kernel on the face vs. non-face classification problem.*

Kernel	Weak constraints	Hard constraints
Triangular	6.88 %	0.69 %
Gaussian ($\sigma = 10^3$)	7.36 %	1.56 %
Gaussian ($\sigma = 6.10^2$)	7.83 %	0.90 %
Gaussian ($\sigma = 10^2$)	21.14 %	37.73 %
Gaussian ($\sigma = 10$)	41.80 %	37.73 %

4.2.4 Face recognition

Experiments on face recognition have been conducted on the original 400 face images from the Olivetti database. These images are first processed using histogram equalization in order to compensate the lighting effects. Then, these raw values are resized to 64×64 pixels and used to compute the eigenvectors of the underlying Gram matrix. Each face image is projected using these eigenvectors, and only the coefficients related to the largest eigenvalues are retained.

Diagram (8, left) shows the precision of face recognition with respect to the number of dimensions used during projection. This precision is measured by the number of times a face query and its best match belong to the same person.

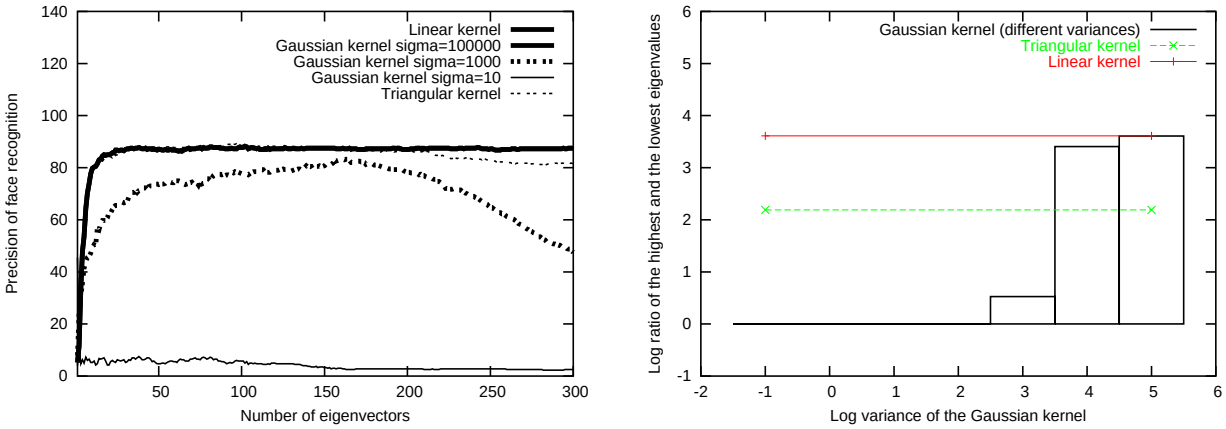

Figure 8: *(Left) Precision of face recognition with respect to the number of eigenvectors used during projections and for different kernels. (Right) Log the ratio of the largest and the smallest eigenvalues for different values of σ. The dotted line shows this ratio for the triangular kernel which is of course parameter free. A high value of this ratio leads to numerical instabilities.*

We can see that the performance of face recognition when using the Gaussian kernel ($\sigma = 100000$) is similar to the linear one. When the σ is very large, the ratio between the largest and the smallest eigenvalues is high[2] (cf. figure 8, right), so the ellipsoid englobing the data in the features space is extended in the dimensions corresponding to the highest eigenvalues and flatten in the other dimensions. On the contrary, when the variance of the Gaussian kernel is close to 0, the condition number is small and goes to 1, so the ellipsoid englobing the data will be homomorphic to a ball. The selection of the best variance can be interpreted as finding the shape of the ellipsoid such that the performances of discrimination are optimal. Notice that more dimensions retained make the coefficients of projection more sensitive to noise so the precision of face recognition falls (cf. figure 8, left).

5 Discussion

5.1 Soft margin

Soft margin SVM and SVR training consists in bounding the Lagrange coefficients α_i in order to control the influence of outliers on the learning process. Also, in many concrete situations, the range of values allowed for the coefficients is fixed by computer representations of the real numbers. With such bounding, the theoretical invariance to scale would not hold anymore.

Nevertheless, our main result shows that, with the triangular kernel k_T, the coefficients are proportional to the inverse of the scaling of the population. Such a linear increase is very reasonable and lead to values that could be handled without bounding in all our experiments.

5.2 Ideal invariant training set

Another interesting property appears when we consider an hypothetical infinite two dimensional spiral-shaped training set. Such an infinite set $\mathcal{S}$ could be built to be invariant to a certain mapping ρ, composition of a rotation and a scaling (this set would be an union of orbits of that mapping cf. figure 9). The training of an SVM with the triangular kernel would be also invariant under that transformation. So if we denote $f_{\mathcal{S}}$ (respectively $f_{\rho(\mathcal{S})}$) the classification function obtained by training on $\mathcal{S}$ (respectively on $\rho(\mathcal{S})$), as $\mathcal{S} = \rho(\mathcal{S})$ we would have:

$$\forall\, x, f_{\mathcal{S}}(x) = f_{\rho(\mathcal{S})}(\rho(x)) = f_{\mathcal{S}}(\rho(x)) \tag{26}$$

which means that the learned boundary itself would be invariant. That implies it would possess details at several scales at while. We do not have such an example in real, but we can still approximate that result by considering a finite spiral-shaped set. As it can be seen on figure 10, the boundary at the center has a finer scale far smaller than at the outer area.

[2] In the community of linear algebra, this ratio is usually referred to as the condition number

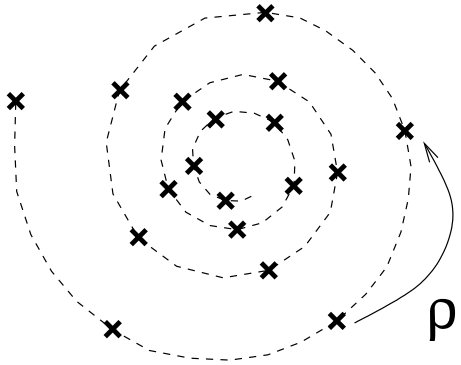

Figure 9: *The iterations of a mapping ρ, composition of a rotation and a scaling, generate an infinite set of point invariant under ρ.*

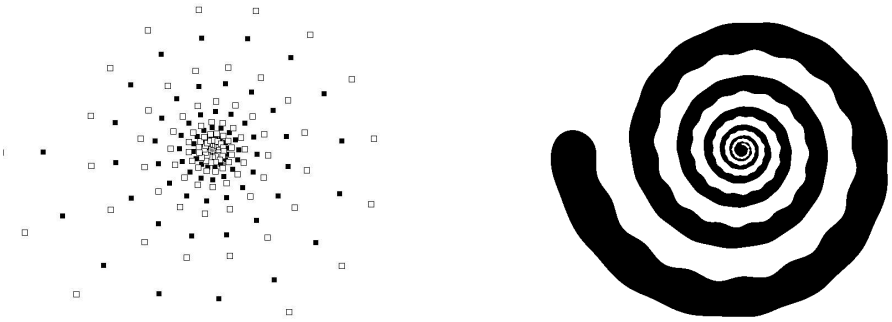

Figure 10: *The triangular kernel can separate two populations, even if it requires various scales. Training set is shown on left, and classification with the triangular kernel is shown on right.*

5.3 The condition number

The ratio of the largest to the smallest eigenvalues of the Gram matrix is referred to as the condition number. The base-b logarithm of this number is an estimate of how many base-b digits are lost in finding for instance the SVM Lagrange coefficients [37]. This number is known as the worst case loss of precision when solving a linear system $K\ \alpha = y$ where K is a Gram matrix of the training data in $\mathcal{S}$ and $y^t = (y^{(1)}...y^{(N)})$ is the underlying vector of labels. Some training parameters, such as the variance of the Gaussian, can control the condition number. If σ is too large, the condition number (denoted δ) will be very "high" (cf. figure 8) so the linear system becomes ill-conditioned. "High" means that $log(\delta)$ is bigger than the precision of the matrix entries. When this condition number is infinite the linear system is said to be singular.

According to our observations when using the triangular kernel, the condition number is of course scale invariant (cf. equation (24)) and also small compared to the condition number when using the Gaussian (when σ is large, cf. figure 8). As it is hard to find an appropriate σ when using the Gaussian kernel on a population mixing several scales, this may result into an overestimated σ on a particular subset of the training data (cf. the spiral example in §5.2). In this case, the rate of the increase of δ when using the Gaussian kernel will be higher than when using the triangular one (cf. figure 8).

6 Conclusion

We discussed in the paper the scale invariance of kernel methods using the triangular kernel and its application for pattern recognition problems. One of the main interesting points of this kernel is its good generalization performance, its high numerical stability in the sense of the condition number and of course its affine invariance.
Future issues will include the study of theoretical generalization performances of kernel methods using this kernel. More applications in computer vision may be found such as achieving photometric invariance under the hypothesis of the linear Lambertian model.

References

[1] H. Sahbi and F. Fleuret, "Scale-invariance of support vector machines based on the triangular kernel," *INRIA Research Report, N 4601*, 2002.

[2] C. Burges, "Geometry and invariance in kernel based methods," *In B. Schölkopf and C. J. C. Burges and A. J. Smola, editors, Advances in Kernel Methods — Support Vector Learning*, pp. 89–116, 1999.

[3] S. Abe, "On invariance of support vector machines," *Fourth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'03)*, 2003.

[4] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," *In Artificial Neural Networks — ICANN*, pp. 47–52, 1996.

[5] P. Niyogi, T. Poggio, and F. Girosi, "Incorporating prior information in machine learning by creating virtual examples," *IEEE proceedings on intelligent signal processing*, vol. 86, no. 11, pp. 2196–2209, 1998.

[6] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," *Proceedings of the conference on Advances in neural information processing systems*, pp. 640–646, 1998.

[7] O. Chapelle and B. Schölkopf, "Incorporating invariances in nonlinear support vector machines," *Availabe at: www-connex.lip6.fr/∼chapelle*, 2001.

[8] V.N. Vapnik and A.Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth, "Learnability and the vapnik-chervonenkis dimension," *Journal of the ACM*, vol. 36, no. 4, pp. 929–965, 1989.

[10] N. Cristianini, C. Campbell, and J. Shawe-Taylor, "Dynamically adapting kernels in support vector machines.," *In M.S. Kearns, S. A. Solla, and D. A. Cohn, editors, Advances in Neural Information Processing, 11. MIT Press*, 1998.

[11] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, 2002.

[12] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[13] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[14] B. Heisele, T. Serre, S. Mukherjee, and T. Poggio, "Feature reduction and hierarchy of classifiers for fast object detection in video images," *In: Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 18–24, 2001.

[15] A. N. Tikhonov and V. Y. Arsenin, "Solutions of ill-posed problems," *W. H. Winston, Washington, D.C.*, 1977.

[16] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *J. Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95, 1971.

[17] Vladimir N. Vapnik, "The nature of statistical learning theory," *Springer Verlag*, 1995.

[18] B. Scholkopf and A. Smola, "Learning with kernels," *MIT Press*, 2002.

[19] G.Wahba, "Support vector machines, reproducing kernel hilbert spaces and the randomized gacv," *Technical Report 984, University of Wisconsin, Madison*, 1997.

[20] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with gaussian kernels to radial basis function classiers.," *In IEEE Transcations on Signal Processing*, vol. 45, no. 11, pp. 2758–2765, 1997.

[21] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and SA. Solla, "structural risk minimization for character recognition," *Advanced in Neural Information Processing Systems*, vol. 4, pp. 471–479, 1992.

[22] J. J. More and S. J. Wright, "Optimization software guide," *Siam Publications*, 1993.

[23] R. Fletcher, *Practical Methods of Optimization*, vol. 1, John Wiley & sons, New York, 1980.

[24] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.*, 1998.

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

[26] C. Berg, "Harmonic analysis on semigroups: Theory of positive definite and related functions," *Springer Verlag*, 1984.

[27] Bernhard Scholkopf, "The kernel trick for distances," *in proceedings of NIPS*, pp. 301–307, 2000.

[28] C.A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Constr Approx*, vol. 2, no. 11, 1986.

[29] Vladimir N. Vapnik, "Statistical learning theory.," *A Wiley-Interscience Publication*, 1998.

[30] Martin Anthony, "Mathematical modeling of generalization," *In M. Marinaro, R. Tagliaferri (eds.), Neural Nets: 13th ItalianWorkshop on Neural Nets Springer LNCS 2486*, 2002.

[31] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," *IEEE transaction on pattern analysis and machine intelligence*, vol. 17, no. 6, pp. 545–561, 1995.

[32] F. Mokhtarian, S. Abbasi, and J. Kittler, "Robust and efficient shape indexing through curvature scale space," *In proceedings of British Machine Vision Conference*, pp. 53–62, 1996.

[33] ," *http://yann.lecun.com/exdb/mnist/.*

[34] H. Sahbi, D. Geman, and N. Boujemaa, "Face detection using coarse-to-fine support vector classifiers.," *In Proceedings of the IEEE International Conference on Image Processing.*, pp. 925–928, 2002.

[35] http://www.cam orl.co.uk/facedatabase.html, ," .

[36] P. Viola and M. Jones, "Robust real-time object detection. (to appear)," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.

[37] Z. Bai, J. Demmel, and A. McKenney, "On computing condition numbers for the nonsymmetric eigenproblem," *Source ACM Transactions on Mathematical Software (TOMS) archive*, vol. 19, no. 2, pp. 202–223, 1993.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399