INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Scale-Invariance of Support Vector Machines based on the Triangular Kernel

Hichem Sahbi — François Fleuret

N° 4601

Octobre 2002

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4601--FR+ENG

Scale-Invariance of Support Vector Machines based on the Triangular Kernel

Hichem Sahbi, François Fleuret

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

Rapport de recherche n° 4601 — Octobre 2002 — 13 pages

Abstract: This report focuses on the scale-invariance and the good performances of Support Vector Machines based on the triangular kernel. After a mathematical analysis of the scale-invariance of learning with that kernel, we illustrate its behavior with a simple 2D classification problem and compare its performances to those of a Gaussian kernel on face detection and handwritten character recognition

Key-words: support vector machine, kernel methods, statistical learning, object recognition

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

Invariance au changement d'échelle des SVMs utilisant le noyau triangulaire

Résumé : Ce rapport propose des résultats concernant l'invariance aux changements d'échelles et les bonnes performances des *Support Vector Machines* basées sur le noyau triangulaire. Aprés une analyse mathématique de l'invariance aux changements d'échelles de l'apprentissage avec ce noyau, nous l'illustrons avec un problème simple de classification en 2D, et nous comparons ses performances à celles obtenues avec un noyau Gaussien sur des tâche de détection de visages et de reconnaissance de caractères.

Mots-clés : support vector machine, noyaux, apprentissage statistique, reconnaissance de formes

1 Introduction

For a decade now, Support Vector Machines [3] have proven to be generic and efficient algorithms for classification and regression. SVMs got their popularity both from a solid theoretical support [1, 8], and because they clearly untie the specification of the model from the training. The former corresponds to the choice of the underlying kernel, and the later can be done optimally with classical quadratic optimization methods.

We study in this paper SVMs based on the triangular kernel, and we provide experimental results showing their good performances. We will show that, although less popular than the Gaussian kernel, the triangular kernel makes the training process invariant to a scaling of the data.

In §2 we summarize the standard formalization of SVMs, and we present the triangular kernel. In §3 we show analytically how training a SVM using the triangular kernel is invariant to scaling. Then, we give in §4 results on a simple 2D problem and on real-world tasks to illustrate what this invariance means and to show how it improves the generalization performance.

2 Support Vector Machines

2.1 Standard Formalization

We will focus on SVMs for classification. Basically, SVM methods project data to classify in a space of large (or infinite) dimension, where a linear criterion is used. For any training set, one can choose an appropriate projection Ψ so that linear separability can be acheived. Computation is done without an explicit form of the projection, but only with the kernel corresponding to the scalar product between projections.

The model is thus specified by choosing the kernel k:

$$k(x,\ x') \quad = \quad < \Psi(x),\ \Psi(x') >$$

And let's denote f the function which sign is the predicted class:

$$f(x) \quad = \quad < \omega,\ \Psi(x) > \ + b$$

Let X be a random variable on R^N standing for the feature distribution of data to classify (in the original space), and Y on $\{-1,\ 1\}$ the real class of the data. We will denote $\mathcal{T} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ a training set generated i.i.d according to (X, Y). The computation of ω is acheived by minimizing $||\omega||$ under correct classification of the training set, i.e. $\forall i,\ y_i\ f(x_i) \geq 1$. This is equivalent to maximizing the margin between the training points and the separating hyper-plan (in the high-dimensional space). This ensures good generalization property on the real population. It can be proven [1] that ω is of the form $\sum_i \alpha_i y_i\ \Psi(x_i)$, where the α_i come from the following quadratic optimization problem:

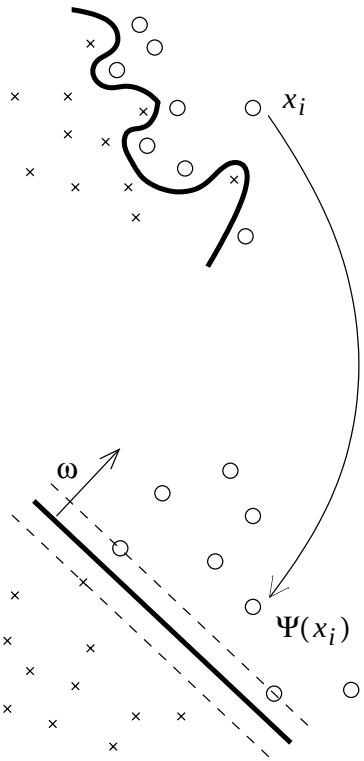

Figure 1: *Classification is performed by a SVM by projecting the original data in a high dimension space, where a linear criterion is used.*

$$\begin{array}{rcl} \text{Minimize} & & \\ L(\alpha) & = & \sum_i \alpha_i \; - \; \sum_{i,j} \alpha_i \alpha_j \; y_i y_j \; k(x_i, \; x_j) \\ \text{under} & & \\ \forall i, & \alpha_i \; \geq \; 0 & \text{and} \quad \sum_i \alpha_i y_i \; = \; 0 \end{array}$$

The value of b does not appear in the optimization, and it has to be computed, given the α_i:

$$b \quad = \quad -\frac{\max\limits_{y_i=-1} \sum\limits_j \alpha_j y_j \; k(x_i, x_j) + \min\limits_{y_i=1} \sum\limits_j \alpha_j y_j \; k(x_i, x_j)}{2}$$

Finally, using the expansion of ω, we can write the classification function as:

$$f(x) \quad = \quad \sum_i \alpha_i y_i \; k(x_i, \; x) \; + \; b$$

2.2 Triangular kernel

The classification power of SVMs comes directly from the complexity of the underlying kernel. Many kernels can be used, the most standard being the Gaussian $k(x, \; x') = \exp\left(-||x - x'||^2/\sigma^2\right)$. 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-dimension space is also almost linear and useless. On the contrary, when underestimated, the function lacks any regularization power and the decision boundary is jagged and irregular, highly sensitive to noisy training data (see figure 3, middle row). Several methods have been developed to estimate an optimal σ, so that the whole process would be invariant to scaling [4].

We focus here on a less standard kernel referred as the *triangular kernel*, which is basically an affine function of the Euclidean distance between the points in the original space, expressed as $k_{\mathrm{T}}(x, \; y) = (1 - ||x - y||/\sigma)^+$. The $()^+$ forces this mapping to be positive, and ensures this expression to be a kernel. We will make the assumption that we can chose σ so that all the data lives in a ball of radius $\frac{\sigma}{2}$, i.e. $P\left(||X|| \leq \sigma/2\right) = 1$

As we will see below, the process is independant of σ. Finally, the kernel can be defined without forcing its positiveness:

$$k_{\mathrm{T}}(x, \; y) \quad = \quad 1 - \frac{||x - y||}{\sigma}$$

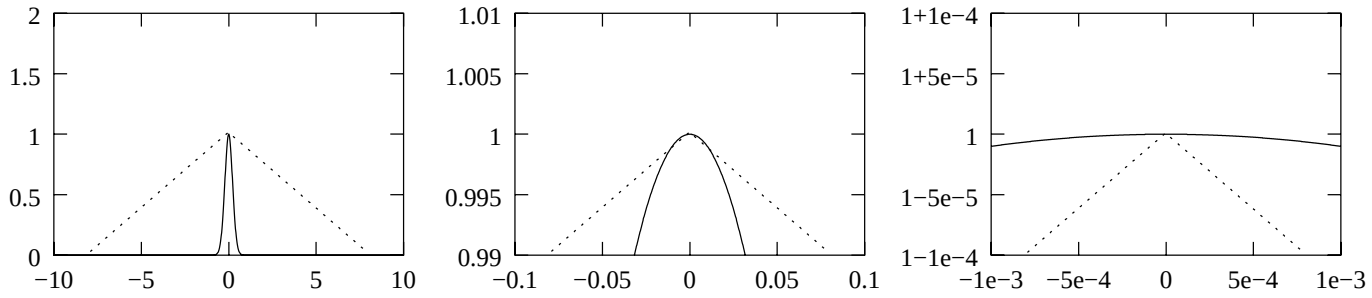

Figure 2: Gaussian kernel (continuous line) and triangular kernel (dashed line) at various scales (left to right, respectively $\times 10^0$, $\times 10^2$ and $\times 10^4$). Intuitively, whereas the triangular kernel is the same at all scales, the Gaussian kernel has different shapes, from a Dirac-like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

Note that it can be easily proven ([2], page 27) that this kernel has an infinite VC dimension, and is at least as powerful as the Gaussian kernel in term of separability.

3 Scale-invariance of the classifier

3.1 Scaling of the triangular kernel

An interesting property of the triangular kernel is his invariance "in shape" to scaling (see figure 2). Given a scaling factor $\gamma > 0$, such an invariance can be formally expressed as:

$$\begin{aligned} k_{\mathrm{T}}(\gamma\ x,\ \gamma\ y) &= 1 - \gamma\ \frac{||x-y||}{\sigma} \\ &= \gamma\ k_{\mathrm{T}}(x,\ y)\ +\ (1-\gamma) \end{aligned} \tag{1}$$

Thus, when the points are scaled by a certain factor γ, the value of the kernel scales by γ, plus a constant term that does not depend on the points but only on γ. As we will see, the equilibrium constraint $\sum_i \alpha_i y_i = 0$ makes the $1-\gamma$ to vanish in the decision function.

3.2 Invariance of the classifier

In the following, we consider a situation where we scale the data by a factor $\gamma > 0$. Let's denote $\mathcal{T}^\gamma = \{\gamma\ x_1,\ \ldots,\ \gamma\ x_n\}$ a training set for that population. We denote f^γ the classification function obtained by training the SVM on $\mathcal{T}^\gamma$ (thus, f^1 is the classifier built from the data at original scale). We will show the following equality:

$$\forall x, \qquad f^{\gamma}(\gamma\ x) = f^{1}(x)$$

Let α^{γ}, ω^{γ} and b^{γ} be the parameters of the classification function estimated on $\mathcal{T}^{\gamma}$. We have:

$$f^{\gamma}(x) \quad = \quad \sum_i \alpha_i^{\gamma}\ y_i\ k_{\mathrm{T}}(\gamma\ x_i,\ x)\ +\ b^{\gamma}$$

Thus, the α_i^{γ} come from the minimization system corresponding to $\mathcal{T}^{\gamma}$:

$$\begin{array}{rcl} \text{Minimize} & & \\ L^{\gamma}(\alpha^{\gamma}) & = & \displaystyle\sum_i \alpha_i^{\gamma} - \sum_{i,j} \alpha_i^{\gamma}\ \alpha_j^{\gamma}\ y_i y_j\ k_{\mathrm{T}}(\gamma\ x_i,\ \gamma\ x_j) \\ \text{under} & & \\ \forall i, & \quad \alpha_i^{\gamma}\ \geq\ 0 \quad \text{and} \quad \displaystyle\sum_i \alpha_i^{\gamma}\ y_i\ =\ 0 & \end{array}$$

It follows, from equality (1):

$$\begin{array}{rcl} L^{\gamma}(\alpha^{\gamma}) & = & \displaystyle\sum_i \alpha_i^{\gamma} - \sum_{i,j} \alpha_i^{\gamma}\ \alpha_j^{\gamma}\ y_i y_j\ (\gamma k_{\mathrm{T}}(x_i,\ x_j) + 1 - \gamma) \\ & = & \displaystyle\sum_i \alpha_i^{\gamma} - \gamma \sum_{i,j} \alpha_i^{\gamma}\ \alpha_j^{\gamma}\ y_i y_j\ k_{\mathrm{T}}(x_i,\ x_j) \\ & & \displaystyle - (1-\gamma)\ \left(\sum_i \alpha_i^{\gamma}\ y_i\right)\ \left(\sum_j \alpha_j^{\gamma}\ y_j\right) \\ & = & \displaystyle\sum_i \alpha_i^{\gamma} - \gamma \sum_{i,j} \alpha_i^{\gamma}\ \alpha_j^{\gamma}\ y_i y_j\ k_{\mathrm{T}}(x_i,\ x_j) \\ & = & \displaystyle\frac{1}{\gamma} \left(\sum_i \gamma \alpha_i^{\gamma} - \sum_{i,j} \gamma \alpha_i^{\gamma}\ \gamma \alpha_j^{\gamma}\ y_i y_j\ k_{\mathrm{T}}(x_i,\ x_j)\right) \\ & = & \displaystyle\frac{1}{\gamma}\ L^{1}(\gamma\ \alpha^{\gamma}) \end{array}$$

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

$$\begin{aligned}\sum_j \alpha_j^\gamma y_j k_{\mathrm{T}}(\gamma x,\ \gamma x_j) &= \sum_j \frac{1}{\gamma}\alpha_j^1 y_j (k_{\mathrm{T}}(x,\ x_j) + 1 - \gamma) \\ &= \sum_j \alpha_j^1 y_j k_{\mathrm{T}}(x,\ x_j) + \frac{1-\gamma}{\gamma}\sum_j \alpha_j^1 y_j \\ &= \sum_j \alpha_j^1 y_j k_{\mathrm{T}}(x,\ x_j)\end{aligned}$$

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

$$\begin{aligned}f^\gamma(\gamma x) &= \sum_i \alpha_i^\gamma\ y_i\ k_{\mathrm{T}}(\gamma\ x_i,\ \gamma\ x)\ +\ b^\gamma \\ &= \sum_i \frac{1}{\gamma}\alpha_i^1\ y_i\ (\gamma\ k_{\mathrm{T}}(x_i,\ x) + (1-\gamma))\ +\ b^1 \\ &= \sum_i \alpha_i^1\ y_i\ k_{\mathrm{T}}(x_i,\ x) + \frac{1-\gamma}{\gamma}\sum_i \alpha_i^1\ y_i\ +\ b^1 \\ &= \sum_i \alpha_i^1\ y_i\ k_{\mathrm{T}}(x_i,\ x)\ +\ b^1 \\ &= f^1(x)\end{aligned}$$

4 Experiments

4.1 Simple 2D classification problem

To illustrate the scale invariance of the triangular kernel, we have set up a simple classification task in two dimension. 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 its location in the square (see figure 3, 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 on figure 3.

As expected, the gaussian kernel either smoothes too much (figure 3, middle row, left), is accurate (figure 3, middle row, center) or overfits (figure 3, middle row, right), while the triangular kernel behaves similarly at all scales.

4.2 Face detection

The motivation for this study is to understand the good generalization performance of the triangular kernel in the context of face detection. We have developed in our lab a highly

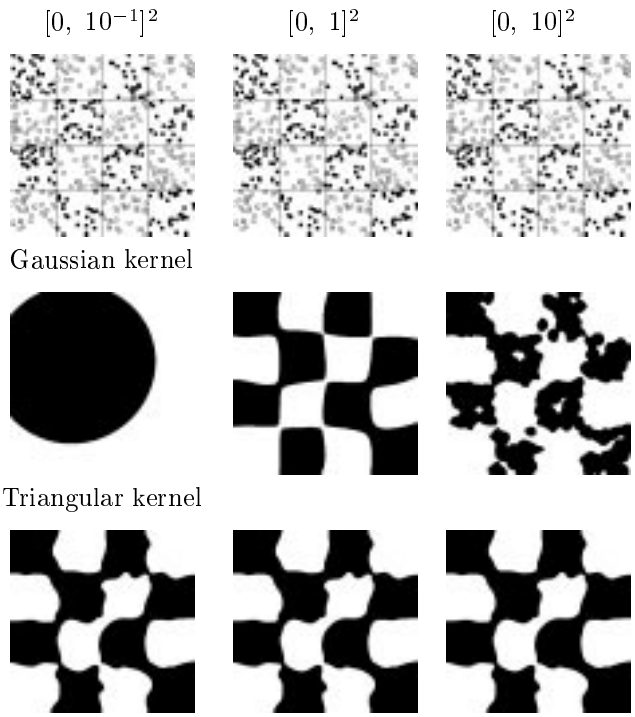

Figure 3: *A simple classification task in 2D. The upper row shows the training set scaled by three different factor. The figures are zoomed accordinf to the same factor for ease of representation. The middle row shows the results of the classifications with a gaussian kernel, and the lower row shows results with the triangular kernel.*

Figure 4: *Training samples two face populations. The upper row shows samples from a loosely constrained population while the lower row shows a sample from a highly constrained population.*

efficient detector based on a hierarchy of SVMs [7]. The main idea behind this approach is to decompose the space of face pictures by constraining more and more their poses (eye locations) in the image plan [5, 6].

We do not go into the details of the face-detection scheme, but we just focus on the generalization performances of individual classifiers dedicated to constrained populations. Figure 4 shows some examples from two of them. The first far less constrained than the second. Both are synthetically generated by doing affine bitmap transformations of the original pictures (ORL database of faces). 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 1 show the performance of both the triangular and the Gaussian kernel. While the Gaussian kernel relies heavily on the choice of σ, the triangular kernel achieves the same order of performances without tuning of any scale parameter.

4.3 Handwritten character recognition

This last experiment is a classical problem of handwritten digit recognition on the MNIST database. Pictures of this database have a resolution of 28×28 black and white pixels.

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 the $10,000$ remaining

Table 1: 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%

Table 2: Performance comparison between the triangular and the Gaussian kernel on handwritten digit classification.

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%

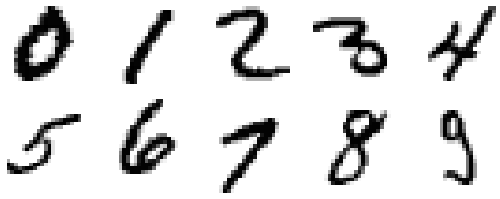

Figure 5: *Some handwritten digits from the MNIST database.*

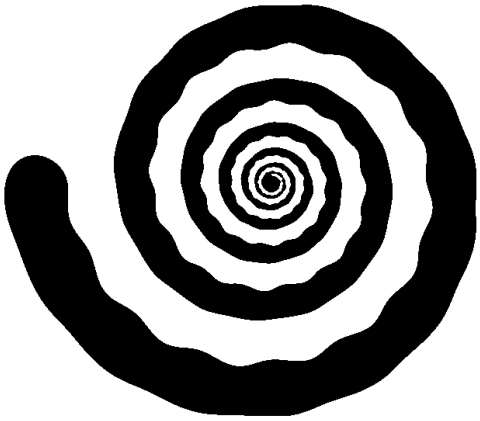

Figure 6: *The triangular kernel can separate two population, even if it requires various scales.*

images. We use a 64 Haar-wavelet coefficient representation of the pictures to gain local invariance to local 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:

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

Results are shown on table 2 for the Gaussian kernel at various σ and the triangular kernel.

5 Conclusion

We have shown in this article that classification with SVMs based on the triangular kernel is invariant to the scaling of the data. Therefore, using this kernel avoids the estimation of an optimal scaling parameter. Such an estimation is usually based on cross validation and is computationnaly intensive, since it requires to run several times the complete training process.

We believe the benefits to get from the triangular kernel to go beyond simple scale invariance as defined in this paper. For instance, experiments show that it ensures a correct classification of population mixing various scales (see the spiral example, for which the boundary at the center requires a scale factor far smaller than at the outer area, figure 6).

References

[1] B. E. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *in Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 5:144–152, 1992.

[2] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[3] N. Christiani and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[4] N. Cristianini, C. Campbell, and J. Shawe-Taylor. Dynamically adapting kernels in support vector machines. In *Proceedings of NIPS*, volume 11, 1998.

[5] F. Fleuret and D. Geman. Coarse-to-fine visual selection. *International Journal of Computer Vision*, 41(1/2):85–107, 2001.

[6] F. Fleuret and D. Geman. Fast face detection with precise pose estimation. In *Proceedings of ICPR2002*, volume 1, pages 235–238, 2002.

[7] S. Sahbi, D. Geman, and N. Boujemaa. Face detection using coarse-to-fine support vector classifiers. In *Proceedings of ICIP2002*, 2002.

[8] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1998.

Contents

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay 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 - 38330 Montbonnot-St-Martin (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