Using an Hebbian learning rule for multi-class SVM classifiers.

Thierry Viéville† and Sylvie Crahay BP 93, INRIA, Sophia, France

Abstract.

Regarding biological visual classification, recent series of experiments have enlighten the fact that data classification can be realized in the human visual cortex with latencies of about 100-150 ms, which, considering the visual pathways latencies, is only compatible with a very specific processing architecture, described by models from Thorpe et al.

Surprisingly enough, this experimental evidence is in coherence with algorithms derived from the statistical learning theory. More precisely, there is a double link: on one hand, the so-called Vapnik theory offers tools to evaluate and analyze the biological model performances and on the other hand, this model is an interesting front-end for algorithms derived from the Vapnik theory.

The present contribution develops this idea, introducing a model derived from the statistical learning theory and using the biological model of Thorpe et al. We experiment its performances using a restrained sign language recognition experiment.

This paper intends to be read by biologist as well as statistician, as a consequence basic material in both fields have been reviewed.

Keywords: Neuronal classifier, Supervised learning, Vapnik dimension, Biological model

† Thierry.Vieville@inria.fr, Tel: +33 6 13 28 64 59, Fax: +33 4 92 38 78 45
`http://www.inria.fr/Thierry.Vieville`

1. Introduction: biological classification is a fact

Biological visual classification[1] is a well-known and very common, but still intriguing fact. As illustrated in Fig. 1, an "object" is recognized in very extreme situations. More generally, the ability to group stimuli into such categories is a fundamental well-established cortical cognitive process (e.g (Freedman et al., 2002)).

Figure 1. The Dalmatian in this picture (image devised by R.C. James) suddenly pops out of the senseless black blobs and dots: a small portion of bottom-up active units quickly lights up the whole pattern of activity (van Tonder and Ejima, 2000), even if there is no explicit visual cues (edge, texture, etc..), thus no way to explicitly extract local features and combined for object detection (Wilson and Keil, 1999). This picture is well known because computer vision scientists confessed not being able to analyze it (Marr, 1982) and we are not able to trace down a sequence of steps (if any) leading to such an holistic percept (Wilson and Keil, 1999).

Recent series of experiments have enlighten this biological mechanism: data classification can be realized in the human visual cortex with latencies of about 150 ms (Thorpe et al., 1996) and even faster (Thorpe, 2002) which, considering the visual pathways latencies (Novak and Bullier, 1997), may only be compatible with a very specific processing architecture and mechanism (Thorpe and Fabre-Thorpe, 2001). Even "high level" visual data classification such as face recognition (Delorme and Thorpe, 2001) can be realized at such a very fast rate. The feed-forward propagation of information may be summarized in Fig. 2

[1] In the present work, data classification simply means being able to put a *unique label* on a given data input (e.g. "oh, there is a dog"). This differs from *categorization* (e.g. (Bajcsy and Solina, 1987)) where not only a *label* but a more complex "semantic structure" is extracted from a given data input.

It has been hypothesized that the underlying neuronal mechanism is based on a rank order coding scheme (Gautrais and Thorpe, 1998): the neuronal information is coded by the relative order in which these neurons fire. The connexionist "Delorme and Thorpe" classification model presented in (Thorpe et al., 2001) is a biologically plausible model of this mechanism. .

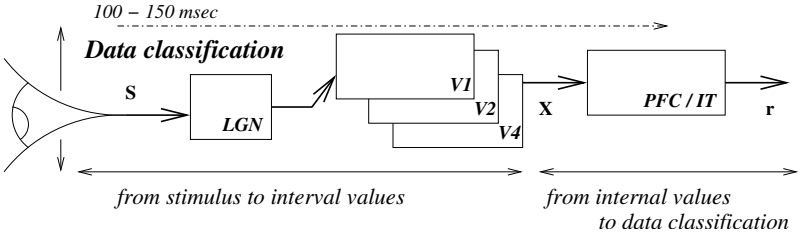

Figure 2. An abstract simplified view of the feed-forward propagation of information, see text for details.

The complex cellular characteristics of the parietal and ventral[2] division of the visual system make it especially suitable to provide various combinations of the data input. In the case of fast data classification, the magnocellular[3] pathway of the parietal/ventral division of the visual system is involved, as asserted by (Delorme et al., 2000) where it is demonstrated that rapid categorization of natural scenes is color blind. A step further, neurons found in the inferior temporal (IT) cortex respond to very complex stimulus features (e.g

[2] *About the "ventral" visual pathway.* Prior to the inferior temporal cortical area, is the so called parietal/ventral pathway (sometimes improperly called "parvocellular" pathway), neurons in the inter-blobs of V1 project to the pale stripes of V2. This pale stripes of V2 project to the inferior temporal cortex. Other feed-forward pathways include the V4 visual area, see (Bullier, 2001) for general review. This pathway is composed of feature detectors (simple, complex and hyper-complex cells) (e.g. (Hubel, 1994) for an introduction). Neurons in this pathway show a low sensitivity to contrast, high spatial resolution, and low temporal resolution or sustained responses to visual stimuli. See for instance (Durbin et al., 1989), Chap 2 for a discussion.

[3] *About (magno/parvo) cellular streams.* There are two classes of cells from the retina and LGN: magnocellular, and parvocellular. These two cell types are contained in different parts of the LGN, and they have different response properties: (i) magnocellular cell receptive fields are 2-3 times larger than parvocellular cell receptive, fields parvocellular have better acuity, resolution magnocellular have better sensitivity, magnocellular cells respond well to moving stimuli, whereas parvocellular cells do not parvocellular cells respond well to color stimuli, whereas magnocellular cells do not.
The magnocellular pathway (older than the parvocellular in phylogenetics) continues the processing of visual detail leading to the perception of shape in area V3 and movement in areas V5 and MST. It has less synaptic relays than the parvocellular pathway, but is faster.

(Burnod, 1993)), regardless of size or position on the retina. For instance, some neurons in this region respond selectively to faces of particular overall feature characteristics. Damage in this area induce disorders[4] of object recognition. There are many neuro-physiological evidences (e.g. (Durbin et al., 1989; Rolls and Treves, 1998)) about the fact that the visual temporal areas[5] function is related to data classification.

Surprisingly enough, this experimental evidence is in coherence with algorithms derived from the statistical learning theory, following the work of Vapnik. More precisely, there is a double link: on one hand, the statistical learning theory offers tools to evaluate and analyze such biological models and on the other hand, the Delorme and Thorpe model is an interesting front-end for algorithms derived from the statistical learning theory.

A step further implementations of statistical learning methods may be efficient biologically plausible models of cortical areas involved in object labelisation. Such an idea is for instance proposed in learning classification in the olfactory system of insects (Huerta at al, 2004) where it is shown that neurons that perform this linear classification are equivalent to hyperplanes tuned by local "Hebbian" learning.

The goal of this work is to develop this double link.

In the next section we introduce[6] the required material from statistical learning theory and consider the Guermeur multi-class SVM classifier, using an Hebbian learning rule to optimize the classifier. Applying this piece of theory, we experiment this mechanism and show that it may be viewed as an "optimized nearest-neighbor classifier". Thanks to these developments, we finally analyze the algorithmic and computational reasons that make the Delorme and Thorpe model interesting with respect to the statistical learning theory.

---

[4] Common examples of such disorders include visual agnosia, or the inability to identify objects in the visual world, and prosopagnosia, a subtype of visual agnosia that affects specifically the recognition of once familiar faces.

[5] *About IT.* The inferior temporal cortex is though to consist of three parts: The TEO (the occipital division of the intra-temporal cortex), the TE (the median division), and the STS (superior temporal sulcus).The TEO is used for making discriminations between 2-D patterns which differ in form, color, size, orientation, or brightness. The TE is used for recognition of 3-D objects. Both the TE and STS are thought to be used in facial recognition and in the recognition of familiar objects. The STS may be the place in which the feature maps of objects (which contain separate information about each primitive of an object, such as color, orientation, or form) become object files.

[6] *About footnotes* Since this paper presents material from both computer science and life science, we have introduced several footnotes reviewing basic facts for both sizes, providing the reader with a self-contained document.

2. Implementing a multi-class SVM classifier.

Data classification with supervised deterministic learning.

In computer science, a "data classifier" provides a "label" for a data corresponding to a set of "features" measured from inputs related to the observed object. See (Theodoridis and Koutroumbas, 1999) for a recent comprehensive and introductory treatise on the subject.

A *classifier* $c()$ is thus a function :

$$c : \mathcal{R}^n \to \{1..R\}$$

which associates to each *data* vector $\mathbf{x} \in \mathcal{R}^n$ (data is represented by an array of numerical values) a *category* $r \in \{1..R\}$ (the class or category is numbered from 1 to R), with $r = c(\mathbf{x})$.

Such a classifier is *trained* (i.e. calibrated) by a *calibration set* (also called training set) i.e. a set of M pairs $\{\cdots, (\mathbf{x}_i, r_i), \cdots\}$ if and only if $\forall i,\ r_i = c(\mathbf{x}_i)$. The calibration set contains *typical features* which are *exact* data. As such, the present paradigm corresponds to *supervised learning without training error*, called, say, *deterministic learning*. This differs[7] from usual paradigms used in statistical learning, where a training set is randomly sampled.

The fact we proposed to consider learning sets without training error is a simple technical "simplification" to lighten the derivations. Taking "mistakes" into account is a solved problem (Vapnik, 1995; Bartlett and Shawe-Taylor, 1999; Guermeur, 2002a).

Considering a set such *prototypes* (either the calibration set OR an improvement of it, derived in the sequel), a natural idea is to chose the category of data if and only if it is "closer" to one prototype of this category, than to prototypes of another category. For N prototypes,

[7] *Deterministic selection of calibration sample.* In usual statistical approach, it is assumed that the training samples are chosen by M independent draws from the same probability distribution as the future samples. This probability distribution is a model of the natural processes which give rise to the observed phenomenon. The training samples thus provide a "view" of the underlying model.
In our context, the calibration set is not "sampled" but "chosen" by an "expert". On one hand, this means that it is a "very lucky" set of draws, without mistake. On the other hand, this means that the expert must randomly choose a "representative" set of draws, i.e. so that the training sample distribution correspond to the future samples distribution.
This is not the unique strategy of such an expert: for instance a "discriminative" set of draws (in which examples close to the limit between two categories are chosen in order to help building this border, or in which "exceptional examples" are highlighted because not easily detected otherwise) would be an interesting alternative, but in contradiction with the underlying assumptions.

the classifier is thus formally[8] defined by:

$$c(\mathbf{x}) = \arg\max_{r_i, 1 \leq i \leq N} c_i(\mathbf{x}) \quad (1)$$

where $c_i()$ is the "proximity" to the ith sub-category related to the prototype $\mathbf{x}_i$ of index i.

In the particular case where categories are linearly separable we can simply write:

$$c_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} + b_i \quad (2)$$

for some $\mathbf{a}_i \in \mathcal{R}^n$ and $b_i \in \mathcal{R}$ and there is a duality between such linear proximities and a thresholded squared distance to prototypes, as reviewed in details in appendix A. Furthermore, if we consider all calibration data as prototypes, we obtain a classifier which outputs correct categories for the calibration set. Frontiers between categories, are piece-wise linear, as visible in Fig. 4.

Obviously, such trivial mechanism is far from being optimal. However, surprisingly enough, an optimal mechanism has a very similar architecture, as reviewed now.

Training capability and learning performances

The Vapnik learning theory (Vapnik, 1995) allows to formalize the idea that efficient models have a limited complexity. As such, it is a formalization and in fact an improvement of the well-known Occam's Razor principle[9].

Let us review this piece of theory following recent works in the field (Bartlett and Shawe-Taylor, 1999; Guermeur, 2002a; Guermeur, 2002b). For a given classifier c in a class $\mathcal{C}$ of classifiers, it relates:

- the *expected risk* $R(c)$ (i.e. the "average" probability for the classifier to provide a wrong answer) for a set of inputs, randomly chosen according to an unknown probability distribution

[8] *General categories definition and prototype proximity.* Let us explain why for a suitable set of function $c_i()$, (2) defines the data category in the general case. Let us consider that each sub-category related to the prototype of index i corresponds to a "region" of the data space, defining a partition of this space. Let us define the border of each region using, here, a general equation $c_i(\mathbf{x}) = 0$. This defines an hyper-surface which, according to the Jordan theorem, delimits what is inside (say when $c_i(\mathbf{x}) > 0$) and outside (when $c_i(\mathbf{x}) < 0$) this region. Here a category C is defined as the union of the sub-categories related to the prototype of index i belonging to C. In this general context, equation (1) precisely determines which $c_i(\mathbf{x}) > 0$, thus the region of a given data. There is thus no loss of generality in the present approach.

[9] *The Ockham razor.* William of Ockham (may be the most influential philosopher of the 14th century) stated: *one should not increase, beyond what is necessary, the number of entities required to explain anything* (see for instance `http://pespmc1.vub.ac.be/OCCAMRAZ.html` for details).

with

– the *empirical risk* $R^M_{emp}(c)$ (i.e. the "average" probability for the classifier to provide a wrong answer) for the calibration set of size M (Here, this quantity is zero, since we have chosen to consider a simple case without mistake in the calibration set and use this fact in the derivation).

More precisely, for a chosen probability δ, the *expected risk* can be bounded with a probability at least $1 - \delta$ as follows:

$$R(c) \leq \underbrace{R^M_{emp}(c) + \underbrace{\epsilon_\delta(M, \mathcal{C})}_{\text{bias}}}_{\text{"guaranteed risk"}} \tag{3}$$

where the *bias* (also called confidence bound) $\epsilon_\delta(M, \mathcal{C})$ is a function of the chosen probability δ, the calibration set size M and the *"complexity" of the class $\mathcal{C}$ of classifiers*, i.e. the set of classifiers used during the training phase.

For binary classifiers, an appropriate measure of complexity is the Vapnik-Chervonenkis (VC) dimension, which -in words[10]- is the (eventually unbounded) size of the largest set of points shattered without restriction by the classifier functions class (Vapnik, 1998). Another measure of complexity is the fat-shattering dimension,
e.g. (Bartlett and Shawe-Taylor, 1999), which may be viewed as the VC dimension obtained requiring that outputs are a fixed quantity above the correct classification threshold.

For multi-class classifiers (Guermeur, 2002a), the *covering number* $\mathcal{N}_{\gamma M\mathcal{C}}$ at a given scale γ is a measure of complexity, and the criterion used in the sequel (Guermeur, 2002b) is based on this concept. The author introduces a quantity, say the Guermeur dimension, and written Y_g here which is a monotonic function of the classifier complexity. This quantity is made explicit in (5).

Indeed, we expect the bias to decrease with the calibration set size M. The learning mechanism is *consistent* if and only if

$$\lim_{M\to\infty} \epsilon(M, \delta, \mathcal{C}) = 0$$

Better than that, if the classifier functions are bounded, at the convergence, the smallest/optimal value of the expected risk (Vapnik, 1995) is obtained. It appears that if the classifiers class $\mathcal{C}$ is too large (i.e. if the

[10] Intuitively, the highest this dimension, the highest the number of (eventually very exotic) data set such classifier class can discriminate; with a low VC dimension, this classifier class only accepts to discriminate a restrained (expected "reasonable" or "plausible") data set with the idea that classification is thus more robust.

complexity is too large), the process is *not* consistent: with a very large class of classifiers, we can classify everything, but what does everything does anything.

In the particular case of linear classifiers with N categories as defined in (1) and (2) it has been shown (Guermeur, 2002a) that in coherence with the previous piece of theory, a reasonable bound of the expected risk is an increasing function of a criterion which is minimal[11] if and only if

$$\frac{Y_g}{D^2} = \frac{N^2(N-1)}{2} \sum_{i=1}^{N} ||\mathbf{a}_i||^2 \tag{4}$$

is minimal. Here D is the radius of the smallest ball containing all data.

Using non-linear functions of the input.
In the general case where we want to consider *not linearly separable* categories, a natural idea (Vapnik, 1995; Shawe-Taylor et al., 1998; Guermeur, 2002b) is to choose a set of non-linear functions of the input but consider linear combinations of these non-linear functions. This allows to reuse, in a generalized case, the linear framework.

For instance, in the present implementation, we consider algebraic functions (i.e. polynomials) sufficient to define classifiers of relatively huge complexity since polynomials approximate any regular curve, e.g. (Benedetti and Risler, 1990). Polynomials are linear combinations of monomials and such non-linear classifiers appear as a linear classifier in the extended parameter space.

Choosing monomials of degree $1, 2 \cdots 3$ yields a sequence of classifiers classes $\mathcal{C}$ of increasing complexity. As the complexity increases the risk of over-fitting the data correspondingly increases. But if the chosen class is closer to the ground truth, this should also allow to consider a *smaller number N of prototypes* and thus decrease the complexity. Choosing the class for which the criterion bound is the tightest, thus allows to find the best balanced compromise, minimizing a kind of *structural risk*.

[11] *Derivation of the Y_g criterion.* More precisely, Theorem 1, 2 and 6 of (Guermeur, 2002a) establish that $\epsilon_\delta(M, \mathcal{C})$ is bounded by an increasing function of
$$Y'_g = D^2 \frac{N(N-1)}{2} \sum_{i<j}^{N} ||\mathbf{a}_i - \mathbf{a}_j||^2$$
while appendix A.3 of (Guermeur, 2002a) reviews that at the optimum
$$\sum_{i<j}^{N} ||\mathbf{a}_i - \mathbf{a}_j||^2 = N \sum_{i=1}^{N} ||\mathbf{a}_i||^2$$
eq. (4) being the combination of both.

USING AN HEBBIAN RULE TO MINIMIZE THE Y_g CRITERION.

Minimizing (4) for a fixed number N of prototypes on bounded parameters of limited precision as discussed in appendix B, eq. (15), for a centered thresholded nearest-neighbor classifier reviewed in appendix A, eq. (12), corresponds to the following optimization problem:

$$\min \sum_i^N ||\mathbf{a}_i||^2 \text{ with } \left\{ \begin{array}{l} \sum_i \mathbf{a}_i = \mathbf{0}, \sum_i b_i = 0 \\ \forall k \quad \max_{r_i=r_k, r_j \neq r_k} (\mathbf{a}_i - \mathbf{a}_j)^T \mathbf{x}_k - (b_i - b_j) > 1 \end{array} \right. \tag{5}$$

As being a convex quadratic criterion with linear constraints, it has a unique minimum and the local minimization of this criterion leads to the global minimum. In fact, the solution is *an affine combination of the calibration data, with a constant sum of weights* as derived in appendix C, eq. (16).

The Hebbian theory states that if a neuron $\mathbf{x}_l$ projects to neuron $\mathbf{a}_h$ and $\mathbf{x}_l$ and $\mathbf{a}_h$ are correlated (e.g. active simultaneously), the connection between $\mathbf{x}_l$ and $\mathbf{a}_h$ is increased (e.g. potentiated or reinforced).

Here, we implement this idea, considering a calibration data $\mathbf{x}_l, l = 1..M$ and a classifier parameter $(\mathbf{a}_h, b_h), h = 1..N$ with, for some increment (δ, ν), a rule of the form:

$$\mathbf{a}_h \leftarrow \mathbf{a}'_h = \mathbf{a}_h - \delta\, \mathbf{x}_l \text{ and } b_h \leftarrow b'_h = b_h - \nu \tag{6}$$

followed by:

$$\forall k, \mathbf{a}_k \leftarrow \mathbf{a}_k + \delta\, \mathbf{x}_l / N \text{ and } b_k \leftarrow b_k + \nu / N$$

in order to preserve $\sum_i \mathbf{a}_i = \mathbf{0}$ and $\sum_i b_i = 0$, as the reader can easily verify.

This is illustrated in Fig. 3.

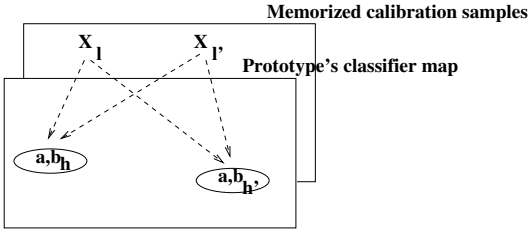

Figure 3. Implementing the criterion minimization using a Hebbian rule, see text for details.

Let us now demonstrate that the proposed rule, applied on all $\mathbf{x}_l$ and $\mathbf{a}_h$, allows to minimize the constrained criterion. This derivation will also provide us with a calculation of the increment (δ, ν).

Let us thus look for a (δ, ν) increment decreasing the constrained criterion (5).Writing $\delta = 2\,\kappa\,(\mathbf{x}_l^T\,\mathbf{a}_h)/||\mathbf{x}_l||^2$, and assuming $\mathbf{x}_l \neq \mathbf{0}$ (otherwise the Hebbian rule has no effect) and $\mathbf{x}_l^T\,\mathbf{a}_h \neq 0$ (otherwise our derivation is singular), the criterion decreases if and only if:

$$\begin{array}{ll} & ||\mathbf{a}'_h||^2 = \delta^2\,||\mathbf{x}_l||^2 - 2\,\delta\,(\mathbf{x}_l^T\,\mathbf{a}_h) + ||\mathbf{a}_h||^2 \leq ||\mathbf{a}_h||^2 \\ \Leftrightarrow & 4\,\kappa^2\,(\mathbf{x}_l^T\,\mathbf{a}_h)^2/||\mathbf{x}_l||^2 - 4\,\kappa(\mathbf{x}_l^T\,\mathbf{a}_h)^2/||\mathbf{x}_l||^2 \leq 0 \\ \Leftrightarrow & \kappa^2 - \kappa \leq 0 \\ \Leftrightarrow & 0 \leq \kappa \leq 1 \end{array}$$

the decrease being maximal for $\kappa = 1/2$, while the decrease for values $\kappa > 1/2$ corresponds to a decrease for a value $1 - \kappa$, below $1/2$.

We thus look for a couple (κ, ν), $0 \leq \kappa \leq 1/2$, with a maximal value of κ while the constraints:

$$\max_{r_i = r_k}[\mathbf{a}_i^T\,\mathbf{x}_k - b_i] - \max_{r_j \neq r_k}[\mathbf{a}_j^T\,\mathbf{x}_k - b_j] > 1$$

are verified.

These constraints can be rewritten in a more compact form:

$$\begin{array}{ll} \text{if } r_k = r_h & \max\,(u'_k, w_k - c_k\,\kappa + \nu) > v_k \\ \text{if } r_k \neq r_h & u_k > \max\,(v'_k, w_k - c_k\,\kappa + \nu) \end{array} \tag{7}$$

with the following notations:

$$\left\{ \begin{array}{lllll} u'_k & = & \max_{r_i = r_k, i \neq h}[\mathbf{a}_i^T\,\mathbf{x}_k - b_i] & v_k & = & \max_{r_j \neq r_k}[\mathbf{a}_j^T\,\mathbf{x}_k - b_j] + 1 \\ u_k & = & \max_{r_i = r_k}[\mathbf{a}_i^T\,\mathbf{x}_k - b_i] - 1 & v'_k & = & \max_{r_j \neq r_k, j \neq h}[\mathbf{a}_j^T\,\mathbf{x}_k - b_j] \\ w_k & = & \mathbf{a}_h^T\,\mathbf{x}_k - b_h & c_k & = & 2\,(\mathbf{x}_l^T\,\mathbf{a}_h)\,(\mathbf{x}_l^T\,\mathbf{x}_k)/||\mathbf{x}_l||^2 \end{array} \right. \tag{8}$$

A step further, since these constraints were already verified at the previous step, for the previous value of $(\mathbf{a}_h, b_h)$ i.e. for $\kappa = \nu = 0$ we also can write, from (7):

if $r_k = r_h$ then $\max(u'_k, w_k) > v_k$ and if $r_k \neq r_h$ then $u_k > \max(v'_k, w_k)$ As a consequence, (i) if $r_k = r_h$ and $u'_k > v_k$, the corresponding inequality is already verified and (ii) if $r_k \neq r_h$ we already have $u_k > v'_k$.

The inequalities (7) thus reduce to:

$$\begin{array}{lll} \text{if } r_k = r_h & \text{and } u'_k \leq v_k & w_k - c_k\,\kappa + \nu > v_k \\ \text{if } r_k \neq r_h & & u_k > w_k - c_k\,\kappa + \nu \end{array}$$

finally rewritten as:

$$\max_{r_k = r_h, u'_k \leq v_k}\,(v_k - w_k + c_k\,\kappa) < \nu < \min_{r_k \neq r_h}\,(u_k - w_k + c_k\,\kappa)$$

Summarizing: in order to find a solution to (6) decreasing (5), we have to look for a maximal value of κ compatible with the following inequalities (written using (8)):

$$\begin{array}{rcl} 0 \leq & \kappa & \leq 1/2 \\ \nu_{min}(\kappa) & < & \nu_{max}(\kappa) \end{array} \text{ with } \left\{ \begin{array}{lcl} \nu_{min}(\kappa) & = & \max_{r_k = r_h, u'_k \leq v_k}\,(v_k - w_k + c_k\,\kappa) \\ \nu_{max}(\kappa) & = & \min_{r_k \neq r_h}\,(u_k - w_k + c_k\,\kappa) \end{array} \right. \tag{9}$$

while $\kappa = 0$ is a known solution.

Let us finally note[12] that the same method could have been used to minimize the Guermeur criterion[11] Y'_g instead of Y_g.

Convergence of the mechanism

If there is an increment (δ, ν) for which the criterion decreases while preserving the constraints, the rule is repeated. Otherwise, this means that there is no linear combination of the form $\mathbf{a}_h += \sum_k \delta_{hk}\, \mathbf{x}_k$ which decreases this convex criterion. Because of convexity, there is thus no local linear variation, which improves this criterion. We thus are at a local minimum, this local minimum being the solution of the convex problem.

Furthermore, we may choose $\mathbf{a}_h$ and $\mathbf{x}_l$ either sequentially (as in our computer implementation, where we select the couple which induces a maximal local decrease of the criterion), in parallel or randomly, as soon as all couples are finally selected, the choice of a strategy being of no influence on the final result, but only the calculation duration.

Edition of the set of prototypes

Modifying $(\mathbf{a}_h, b_h)$ corresponds to a modification of the related prototype $\mathbf{x}_h = \mathbf{\Lambda}^{-1}\,(\bar{c}\,\mathbf{a}_h + \bar{\mathbf{a}})/2$ as made explicit in appendix A. We thus *optimize* the related set of prototypes of this nearest-neighbor classifier. This differs from the choice of support vectors in a SVM.

Furthermore, in (9), if $\forall k, r_k = r_h, u'_k > v_k$ there is no minimal bound for ν, thus no maximal bound for b_h in (6). As a consequence b_h may have huge values, large enough for the hth not to be used anymore in the comparison process. This simply means that this prototype is redundant and can thus be deleted. This mechanism thus automatically *edit the corresponding prototype list*.

However, when N varies, the criterion (5) is not necessarily convex as a function of N and only a local minimum is targeted.

---

[12] *Using Hebbian rules to minimize the Y'_g dimension.* If we consider the minimization of $\sum_{i<j}^{N} ||\mathbf{a}_i - \mathbf{a}_j||^2$ instead of $\sum_{i}^{N} ||\mathbf{a}_i||^2$ in (5) using the Hebbian rule defined in (6) we easily derive:

$$\arg\min_{\mathbf{a}_h} \sum_{i<j}^{N} ||\mathbf{a}_i - \mathbf{a}_j||^2 = \arg\min_{\mathbf{a}_h} \sum_{j\neq h}^{N} ||\mathbf{a}_h - \mathbf{a}_j||^2$$

so that if we replace $\mathbf{a}_h$ with $\mathbf{a}'_h = \mathbf{a}_h - \delta\,\mathbf{x}_l$ in order to obtain:

$$\sum_{j\neq h}^{N} ||\mathbf{a}_h - \mathbf{a}_j||^2 \geq \sum_{j\neq h}^{N} ||\mathbf{a}'_h - \mathbf{a}_j||^2$$
$$= \sum_{j\neq h}^{N} ||\mathbf{a}_h - \mathbf{a}_j||^2 - 2\,\delta \sum_{j\neq h}^{N} (\mathbf{a}_h - \mathbf{a}_j)^T \mathbf{x}_l + \delta^2 \sum_{j\neq h}^{N} ||\mathbf{x}_l||^2$$

writing $\delta = 2\,\kappa \sum_{j\neq h}^{N} (\mathbf{a}_h - \mathbf{a}_j)^T \mathbf{x}_l / \sum_{j\neq h}^{N} 1/||\mathbf{x}_l||^2$ the previous inequality reduces to $0 \leq \kappa \leq 1$ as for the derivation proposed for Y_g and the rest of the derivation is identical, as the reader can easily verify.

Implementation details
As far as, a computer implementation is to be derived, the linear programming problem in (9) can be solved, searching a maximal value $\kappa \in [0..1/2]$ such that $\nu_{min}(\kappa) < \nu_{max}(\kappa)$ using a dichotomy method and choosing:

$$\delta = 2\,\kappa\,(\mathbf{x}_l^T\,\mathbf{a}_h)/||\mathbf{x}_l||^2 \text{ and, say, } \nu = (\nu_{min}(\kappa) + \nu_{max}(\kappa))/2$$

Adaptivity of the mechanism
As soon as a calibration data $(\mathbf{x}_k, r_k)$ is added to the calibration set, without any lack of reactivity, the system is able to provide a first-approximation classification, inserting this data as a new prototype an applying the trivial nearest-neighbor classification mechanism, as reviewed in appendix A.

This not an optimal solution and the minimization of (5) takes place, but is an independent process, realized "when time is available".

As soon as a calibration data is deleted, the current solution is simply to be re-optimized, taking benefit of the fact that one constraint is removed.

With, these simple rules, the learning mechanism is entirely adaptive with respect to calibration data addition / deletion and also with respect to the computation time resources.

Biological plausibility
This mechanism corresponds to a so-called Hebbian-like learning rules as extensively discussed elsewhere
(Durbin et al., 1989; Rolls and Treves, 1998; Gisiger et al., 2000).

As far as biological plausibility is concerned, the previous derivation states that any a small $\kappa > 0$ compatible with (9) decreases the criterion. A biological system may thus simply use "epsilon-values", (say, $\kappa \simeq 10^{-3}$ since all quantities have been normalized with respect to unity), checking $\nu_{min}(\kappa) < \nu_{max}(\kappa)$ in (9).

The whole mechanism thus reduces to (1) linear operations (addition or scalar multiplication) which biological plausibility has been extensively discussed, e.g. (Bugmann, 1997), (2) a min/max operator also biologically plausible as reviewed in e.g. (Yu et al., 2003) and (3) "switches" (detecting redundant data, detecting if an increment is valid, etc..) which is related to so-called inhibition mechanisms, commonly observed in such biological neuronal layers, e.g. (Gisiger et al., 2000).

Considering the architecture of this mechanism we merge a simple nearest-neighbor classifier which may be implemented as standard neuronal network, e.g. (Theodoridis and Koutroumbas, 1999) with a Hebbian learning rule. This rule simply derives from a statistical learning criterion, contrary to e.g. (Soo-Young and Dong-Gyu, 1996) where

the theoretical justification of the generalization performances is only based on a heuristic.

Implementations of SVM like classifiers using fast and simple learning mechanisms has been introduced as "kernel adaptron" methods (Friess et al, 1998) but without a true Hebbian rule as here. Similarly (Krauth and Mezard, 1987) proposed the so called "monivar" learning algorithms in neural networks maximizing the geometrical margin and used in feed-forward layered networks (Mezard and Nadal, 1989) under the name of "tiling" algorithm.

3. Experimental results

Interactive 2D demonstration

Please refer to the on-line[13] software documentation for details about the software module. Examples of results, shown in Fig.5 and 6, illustrate the method behavior and allow to validate the implementation. The module can be experimented on Internet.

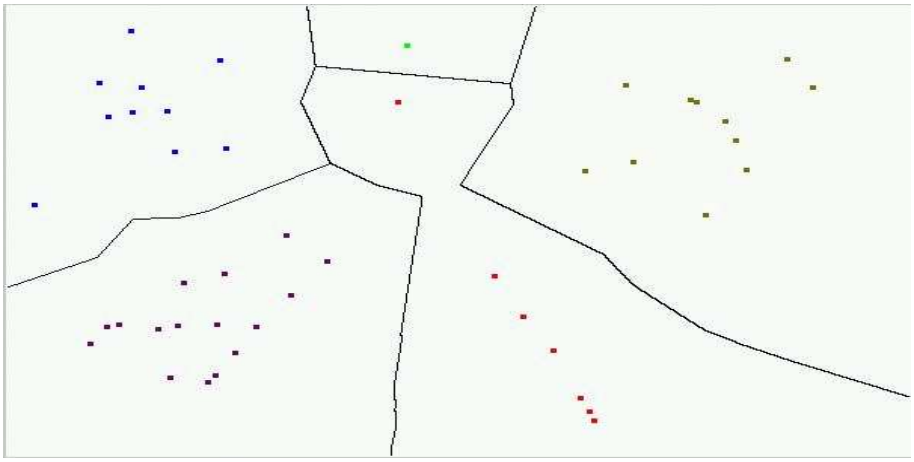

Figure 4. An example of raw nearest-neighbor classification, with 45 prototypes. A margin of 8 pixels is specified as an input of the optimization process. The Guermeur criterion as defined in (4) is huge $Y_g \simeq 10^4$.

In order to qualitatively compare this with a standard SVM method, we have also considered the 1-to-1 multi-class C-SVM method (Chang and Lin, 2001) (comparing each pairs of category using $N(N-1)/2$ standard SVM and choosing the category with the best result),

[13] In `http://www-sop.inria.fr/odyssee/imp` the `imp.math.Classifier` classes.

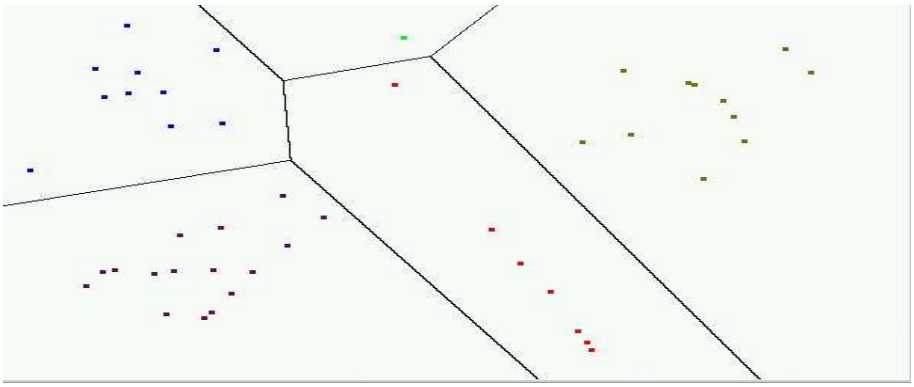

Figure 5. An example of result corresponding to the raw classification given in Fig. 4: the classifier is now optimized using only 1 prototype for each category and $Y_g = 35$.

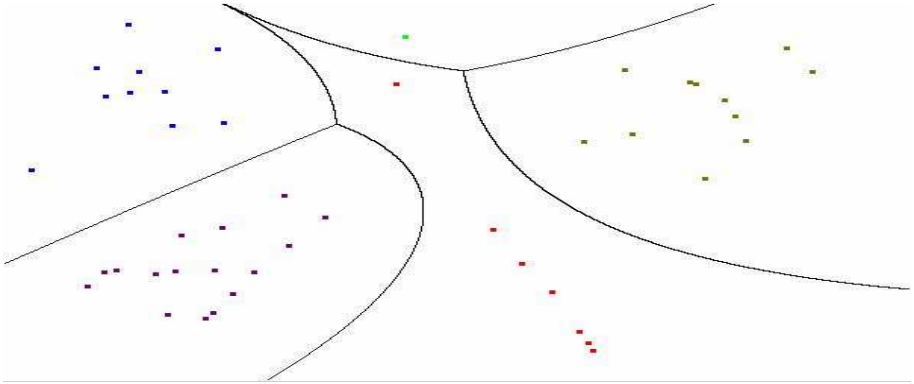

Figure 6. An example of result corresponding to the raw classification given in Fig. 4: the classifier is now optimized with a second order polynomial model, using only 1 prototype for each category and $Y_g = 28$. With this data set, second order polynomial model yields an optimal value of Y_g.

as implemented[14] by Chen and Lin (Chang and Lin, 2001). This is illustrated in Fig. 7.

[14] A free-ware library for support vector machines, thanks to (Chang and Lin, 2001), is available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`

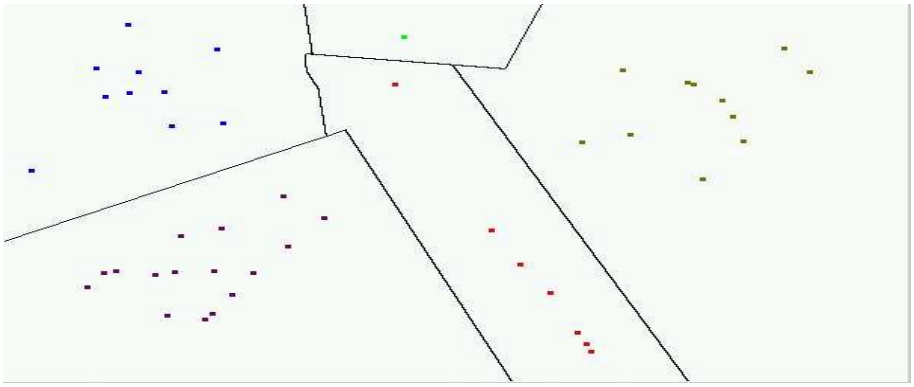

Figure 7. An example of result corresponding to the raw classification given in Fig. 4: here a standard 1-to-1 SVM method is used with 10 support vectors.

Experimenting on bench-mark data

In order to validate the method on a real data set, we have considered *Pima indians diabetes* as provided by B.D. Ripley[15]. The goal is to decide whether a subject has a diabetes or not[16].

We have selected this data set since performances of other methods are available for comparison, as reported by (Figueiredo and Jain, 2001). Percentage of test errors on the Pima data set, for a test set of 269 samples are:

SVM method	23.8 %
Sparse classifier	22.7%
Neuronal network	27.9 %
Other methods	24.2-25.3 %
This classifier	23.81 %

In our experimentation we have used only 200 among the 300 training samples provided because data were missing in the others.

Performances are thus similar to existing methods. This was not obvious because we are using here a learning data set with erroneous samples whereas our mechanism does not reject such errors. This is thus a confirmation that the method seems robust, even in such a case.

[15] Available at `http://www.stats.ox.ac.uk/pub/PRNN`

[16] See `http://www.stats.ox.ac.uk/pub/PRNN/README.html` for details.

Description of the experiment
We consider a tiny experiment related to the recognition of the Quebecian Sign Language Alphabet[17]. This can not be considered as a real experiment of sign-language recognition, whereas it is only used to evaluate the present method.

The static (one image) spelling of four subjects have been recorded using a standard video system with a resolution of 384×288, as follows:

Subject		data
Sy	Experimented	2 series of 9 letters (1 particularly good)
Ad	Beginner	2 series of 9 letters, 1 acceptable, 1 "bad" (used as counter-example)
Th	Beginner	3 series of 9 letters, 2 without shadow, 1 with hand shadow
Li	Beginner	1 series of 9 letters good quality

examples of such images being given in Fig. 8.

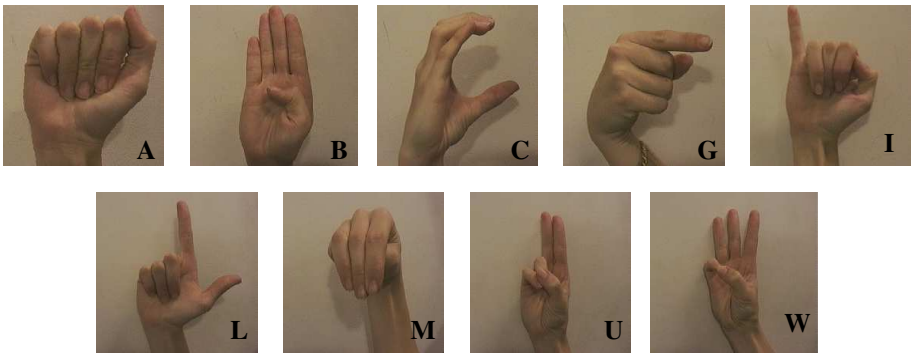

Figure 8. An example of the nine letters taken into account in this experiment, here sub-images containing containing the hand have been automatically cropped as discussed in the text.

The following experimental configurations have been chosen in order to evaluate the method with respect to combinations of data:

[17] See, e.g. `http://www.unites.uqam.ca/surdite`

Experiment label	learning set	test set
Experiment 1	Sy, 9 letters from all samples except 1 series	Sy, 9 letters from the last sample
Experiment 11	Th, 9 letters from all samples except 1 series	Th, 9 letters from the last sample
Experiment 2	Sy, all samples of 9 letters	Th, 3 series of 9 letters
Experiment 3	Th, series of 9 letters without shadow	Th, 2 series of 9 letters with shadow
Experiment 4	Sy/Li, all samples of 9 letters	Th, all samples of 9 letters
Experiment 41	Th/Li, all samples of 9 letters	Sy, all samples of 9 letters
Experiment 42	Sy/Th, all samples of 9 letters	Li, the sample of 9 letters

the learning and test sets intersection being always empty.

Parameter extraction
In order to extract relevant parameters, using standard image analysis methods, edges are detected using a fixed threshold. This simple paradigm is sufficient, with the lighting conditions. The first and second-order momenta (center of gravity, main orientation and lengths) are computed as schematized in Fig. 9. This allows to encapsulate the hand in an ellipse. This also allows to crop a rectangle in the image containing the hand, eliminating a part of the background influence.

The ellipse position (related to the hand position in space) and the ellipse length (related to the hand size and camera proximity) are not relevant to detect the hand sign, whereas the ellipse orientation and the ellipse length ratio (i.e. eccentricity) are. These two parameters are related to the hand relative position (whether it is open/close, tilted, etc..) and are thus used by the classifier with the second-order momenta. A step further, the histograms of the edge abscissa and ordinates are computed, these histogram being smoothed and normalized, as shown in Fig. 9. It appeared to provide a relevant set of parameters to discriminate different hand signs.

Performance evaluation
In order to compare the performances of the present method with a well established mechanism, we again use a standard 1-to-1 SVM method (Chang and Lin, 2001).

Obtained results are summarized in the following table:

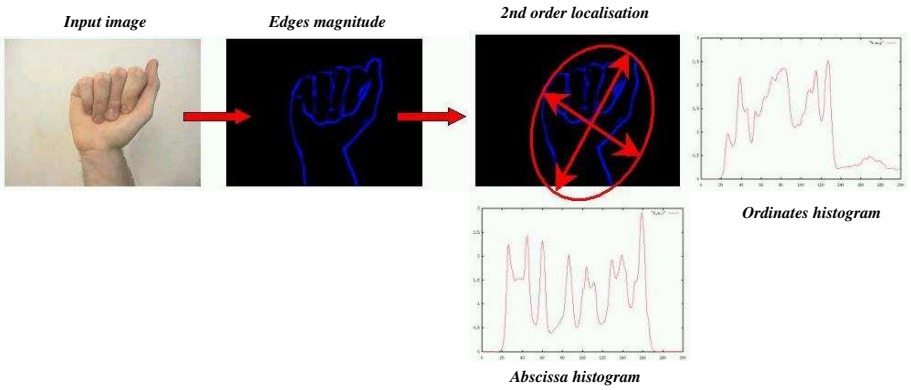

Figure 9. Extracting relevant parameters from raw images, see text for details.

Experiment label	Raw classifier	standard 1-to-1 SVM	The Yg optimal classifier
Experiment 1	11 %	11 %	11 %
Experiment 11	30 %	22 %	27 %
Experiment 2	37 %	33 %	28 %
Experiment 3	61 %	61 %	61 %
Experiment 4	56 %	41 %	62 %
Experiment 41	30 %	37 %	31 %
Experiment 42	0 %	11 %	0 %

where the percentage of errors have been reported. This clearly demonstrates, that up to the 1st order, both methods have similar performances. This was not entirely obvious since we have implemented the optimization using a biologically plausible minimization method. A step further, we clearly observed that the deterministic method have better performances when the calibration data set has no mistakes, while performances are degraded when it contains errors.

Training time for the standard SVM method (about 50 ms $\pm$ 20 ms) and our Yg optimal classifier (about 100 ms $\pm$ 40 ms) have the same order of magnitude, longer for the second method because of the Hebbian optimization method. Testing time is a bit faster for Yg optimized classifier (about 0.2 ms $\pm$ 0.1 ms) than 1-to-1 SVM methods (about 0.8 ms $\pm$ 0.2 ms), not surprising because the number of comparisons is higher in the latter case. The important fact here is that Hebbian-like optimization does not induce huge computation times, but only a little overhead.

4. Discussion

Evaluation of the Delorme and Thorpe model performances
Regarding fast data classification, the Delorme and Thorpe model (Thorpe et al., 2001) is based on the fact that very short observed latencies are only compatible with an information flow related to the *occurrence* of a neuronal signal than to e.g. the spike frequency. Furthermore, since the latency of a given neuron is a direct decreasing function of the neuron input value, only neurons with the *highest values* generate spikes fast enough to be taken into account. This has two consequences.

Quantification of the neuronal information: the quantitative value of the signal is directly related to the spike delay and is thus a *bounded value* with a *finite precision*. Let us consider that the temporal discrimination of a neuron, see e.g. (Carr, 1993) for an extensive study, is of about $\tau = 1\ ms$, so that two spikes arriving within the same millisecond are viewed as simultaneous. Since neural inputs received during a temporal window of $T = 10..20\ ms$ (Thorpe et al., 2001) is taken into account, the temporal resolution[18] is $\sigma = T/\tau$. Considering other coding scheme, such as phase coding
(Gutierrez-Galvez and Gutierrez-Osuna, 2003), a similar analysis of bounded values with a finite precision is derivable.

Sparseness of the neuronal information: among the rather huge number of input neurons (typically the dimension n of the neuronal "vector" has an order of magnitude of 10^5) only a rather small number ($\simeq 10^3$) is taken into account. Here we consider that this selection is made during the *learning phase*, for a given set of prototypes. This differs from thresholding the highest values, which is a non-linear process. As mentioned by one of the reviewer of this paper, a strict analysis should also consider the effects of allowing a choice of 10^3 non-zero parameters from among the 10^5 possibles, leading to an additional increase of the classification complexity.

In the computer implementation of the Delorme and Thorpe model (called *spike-net*, e.g. (Delorme et al., 1999)) a "nearest-neighbor" mech-

[18] *About rate-coding combinations.* If we consider the rate coding scheme (Thorpe et al., 2001), there seems to be $N!$ possible permutations for a given set of N data. However, in practice, at a given temporal resolution τ and during a time window T it is not possible to observe all these permutations. If at a given time t, an input i has been detected, all inputs in the $I_t = [t..t + \tau]$ interval are viewed as "synchronous with i". Such I_t interval is not fixed but triggered by the 1st occurring input. However, we are in a situation where N is very large, so that at the end of each I_t interval yet another spike very likely (almost) immediately occurs, starting a new $I_{t+\tau}$ interval and finally allowing to consider consecutive intervals of duration τ, like when building a temporal "histogram" of the spike occurrences. The number of possible combination is thus of $o(\sigma^N)$.

anism behavior is implemented with an underlying "semi-distance" based on such finite precision quantification and sparse representation. This corresponds to the present framework and we can take a look at its complexity, without detailing the algorithm. In the particular case where this classifier is used as a binary classifier we can easily quantify its complexity, using the VC dimension (Vapnik, 1998) which is bounded [19] by:

$$VC \leq \min\left(\left[\frac{D^2}{\rho^2}\right], n\right) + 1 \tag{10}$$

Here, with the assumptions of appendix B, from (14), $D^2/\rho^2 = \sigma^2/4 \simeq 10^2$ (this being an order of magnitude) leading to a small VC dimension, which does neither depend on the number of neuronal units nor on the complexity of the front-end mechanisms of extraction of data features. This is clearly not the case for standard neuronal networks used as classifiers, because considering for instance their VC dimension again, it is higher than the order of magnitude of the number of neurons. More precisely (Baum and Haussler, 1989), for an arbitrary feed-forward neuronal network with a binary activation function the V_c dimension is of $o(W\ \log(W))$ where W is the number of weights free parameters in the network, while (Koiran and Sontag, 1996) for a multi-layer feed-forward neuronal network with a sigmoid activation function, the V_c dimension is of $o(W^2)$.

Considering the biological model (Thorpe et al., 2001) we can conjecture that, similarly, the neuronal model has a bounded complexity, not because of the sparseness of the neuronal information but because of its quantification. A precise analysis is perspective of the present work.

Conclusion

The present approach allows to re-interpret basic nearest-neighbor classifiers, using the statistical learning theory, obtaining an *optimized version* of this basic mechanism. A key feature is that optimizing the statistical property of nearest-neighbor classifiers allows to automatically add/delete prototypes, edit them and remove redundant ones.

We also have made explicit and experimented that SVM like mechanisms can easily be implemented using Hebbian-like correction rules. Such optimization mechanism is not as fast as the standard method, but its biological plausibility is better, while final performances are similar.

[19] The smallest integer higher than $\left[D^2/\rho^2\right]$ is considered in this formula.

This point of view is in deep relation with fast visual recognition in the brain. It may, for instance, explain why biological classifiers have such surprising generalization performances.

More precisely, the Thorpe et al. model complexity is bounded and does not depends on the network size. It is likely bounded because of the quantification steps (in relation with the temporal resolution of neuronal encoding). This explains its very good performances.

References

Bajcsy, R. and F. Solina: 1987, 'Three Dimensional Object Representation Revisited'. In: *Proceedings of the 1st International Conference on Computer Vision*. London, England, IEEE Computer Society Press.

Bartlett, P. and J. Shawe-Taylor: 1999, 'Generalization Performance of Support Vector Machines and Other Pattern Classifiers'. In: B. Schölkopf, C. Burges, and A. Smola (eds.): *Advances in Kernel Methods, Support Vector Learning*. The MIT Press, Cambridge, Chapt. 4, pp. 43–54.

Baum, E. and D. Haussler: 1989, 'What size net gives valid generalization'. *Neural Computation* **1**, 151–160.

Benedetti, R. and J.-J. Risler: 1990, *Real algebraic and semi-algebraic sets*. Hermann, Paris.

Bugmann, G.: 1997, 'Biologically plausible neural computation'. *Biosystems* **40**, 11–19.

Bullier, J.: 2001, 'Integrated model of visual processing'. *Brain Res. Reviews* **36**, 96–107.

Burnod, Y.: 1993, *An adaptive neural network: the cerebral cortex*. Masson, Paris. 2nd edition.

Carr, C. E.: 1993, 'Processing of Temporal Information in the Brain'. *Annu. Rev. Neurosci.* **16**, 223–244.

Chang, C.-C. and C.-J. Lin: 2001, 'Training nu-Support Vector Classifiers: Theory and Algorithms'. *Neural Computation* **13**(9), 2119–214.

Cover, T. and P. Hart: 1967, 'Nearest Neighbor Pattern Classification'. *IEEE Trans.on Information Theory* **13**(1).

Delorme, A., J. Gautrais, R. VanRullen, and S. J. Thorpe: 1999, 'SpikeNET: A simulator for modeling large networks of integrate and fire neurons'. *Neurocomputing* **26**, 989–996.

Delorme, A., G. Richard, and M.Fabre-Thorpe: 2000, 'Rapid Categorisation of natural scenes is colour blind: A study in monkeys and humans'. *Vision Research* **40**(16), 2187–2200.

Delorme, A. and S. Thorpe: 2001, 'Face processing using one spike per neuron: resistance to image degradation.'. *Neural Networks* **14**, 795–804.

Duda, R. O., P. E. Hart, and D. G. Stork: 2000, *Pattern Classification, 2nd edition*. Wiley Interscience.

Durbin, R., C. Miall, and G. Mitchinson (eds.): 1989, *The computing neuron*. Addison-Wesley.

Figueiredo, M. A. T. and A. K. Jain: 2001, 'Bayesian Learning of Sparse Classifiers'. In: *Computer Vision and Pattern Recognition*.

Freedman, D. J., M. Riesenhuber, T. Poggio, and E. K. Miller: 2002, 'Categorical Representation of Visual Stimuli in the Primate Prefrontal Cortex'. *Science* **291**(5502), 312–316.

T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machine. In *Proc. 15th International Conference on Machine Learning*. Morgan Kaufman, 1998.

Gaspard, F. and T. Viéville: 2000, 'Non Linear Minimization and Visual Localization of a Plane'. In: *The 6th International Conference on Information Systems, Analysis and Synthesis*, Vol. VIII. pp. 366–371.

Gautrais, J. and S. Thorpe: 1998, 'Rate Coding vs Temporal Order Coding : a theorical approach'. *Biosystems* **48**, 57–65.

Gisiger, T., S. Dehaene, and J. P. Changeux: 2000, 'Computational models of association cortex'. *Curr. Opin. Neurobiol.* **10**, 250–259.

Guermeur, Y.: 2002a, 'Combining discriminant models with new multi-class SVMs'. *Pattern Analysis and Applications* **5**(2), 168–179.

Guermeur, Y.: 2002b, 'A simple unifying theory of Multi-Class Support Vector Machines'. Technical Report 4669, INRIA.

Gutierrez-Galvez, A. and R. Gutierrez-Osuna: 2003, 'Pattern completion through phase coding in population neurodynamics'. *Neural Networks* **16**, 649–656.

Hubel, D.: 1994, *L'oeil, le cerveau et la vision : les étapes cérébrales du traitement visuel*, L'univers des sciences. Pour la science.

R. Huerta, T. Nowotn, M. García-Sanchez, H. D. I. Abarbanel, and M. I. Rabinovich. Learning classification in the olfactory system of insects. in preparation, 2004.

Koiran, P. and E. Sontag: 1996, 'Neural Networks with quadratic VC dimension'. *Advances in Neural Information Processing System* **8**, 197–203.

W. Krauth and M. Mezard. Learning algorithms with optimal stability in neural networks. *J. Phis.*, 20:745–752, 1987.

Marr, D.: 1982, *Vision*. W.H. Freeman and Co.

M. Mezard and J. Nadal. Learning in feed forward layered networks: The tiling algorithm. *Journal of Physics*, 22:2191–2204, 1989.

Novak, L. and J. Bullier: 1997, *The Timing of Information Transfer in the Visual System*, Vol. 12 of *Cerebral Cortex*, Chapt. 5, pp. 205–241. Plenum Press, New York.

Rolls, E. T. and A. Treves: 1998, *Neural networks and brain function*. Oxford university press.

Shawe-Taylor, J., P. Bartlett, R. Williamson, and M. Anthony: 1998, 'Structural risk minimization over data-dependent hierarchies'. *IEEE Trans. on Information Theory* **44**(5).

Soo-Young, L. and J. Dong-Gyu: 1996, 'Merging Back-propagation and Hebbian Learning Rules for Robust Classifications'. *Neural Networks* **9**(7), 1213–1222.

Theodoridis, S. and K. Koutroumbas: 1999, *Pattern Recognition*. Academic Press.

Thorpe, S.: 2002, 'Ultra-Rapid Scene Categorization with a Wave of Spikes'. In: *Biologically Motivated Computer Vision*, Vol. 2525 of *Lecture Notes in Computer Science*. pp. 1–15, Springer-Verlag Heidelberg.

Thorpe, S., A. Delorme, and R. VanRullen: 2001, 'Spike based strategies for rapid processing.'. *Neural Networks* **14**, 715–726.

Thorpe, S. and M. Fabre-Thorpe: 2001, 'Seeking categories in the brain'. *Science* **291**, 260–263.

Thorpe, S., D. Fize, and C. Marlot: 1996, 'Speed of processing in the human visual system'. *Nature* **381**, 520–522.

van Tonder, G. J. and Y. Ejima: 2000, 'Bottom - up clues in target finding: Why a Dalmatian may be mistaken for an elephant'. *Perception* **29**(2), 149 –157.
Vapnik, V.: 1995, *The Nature of Statistical Learning Theory*. Springer-Verlag.
Vapnik, V.: 1998, *Statistical Learning Theory*. John Wiley.
Vieville, T.: 2000, 'Using markers to compensate displacements in MRI volume sequences.'. Technical Report 4054, INRIA.
Vieville, T., D. Lingrand, and F. Gaspard: 2001, 'Implementing a multi-model estimation method'. *The International Journal of Computer Vision* **44**(1).
Wilson, R. and F. Keil: 1999, *The MIT Encyclopedia of the Cognitive Sciences*. MIT Press, Cambridge, MA.
Yu, A. J., M. Giese, and T. Poggio: 2003, 'Biophysiologically plausible implementations of maximum operation'. *Neural Computation* **14**(12).

Acknowledgments: *Arnaud Delorme*, *Simon Thorpe* and *Yann Guermeur* are gratefully acknowledged for some powerful ideas in this work.

We are especially thankful for the deep help of the reviewers during the reviewing process.

Appendix

A. Thresholded nearest-neighbor (NN) classifiers.

Linear classifiers correspond to thresholded NN classifiers.
Within the generic framework proposed in (1), for any metric $||\mathbf{v}||_\Lambda^2 = \mathbf{v}^T\,\mathbf{\Lambda}\,\mathbf{v}$ of $\mathcal{R}^n$, defined by a positive definite symmetric matrix $\mathbf{\Lambda}$, let us consider for some thresholds θ_i, the following so called *centered thresholded linear squared proximity to a prototype*, i.e. :

$$\begin{aligned} c_i(\mathbf{x}) &= \left[-||\mathbf{x}-\mathbf{x}_i||_{\mathbf{\Lambda}}^2 + \theta_i + \left(||\mathbf{x}||_{\mathbf{\Lambda}}^2 - \bar{\mathbf{a}}^T\,\mathbf{x} + \bar{b}\right)\right]/\bar{c} \\ &= \mathbf{a}_i^T\,\mathbf{x} - b_i \text{ with } \left\{\begin{array}{rcl} \mathbf{a}_i &=& \left[2\,\mathbf{\Lambda}\,\mathbf{x}_i - \bar{\mathbf{a}}\right]/\bar{c} \\ b_i &=& \left[||\mathbf{x}_i||_{\mathbf{\Lambda}}^2 - \theta_i - \bar{b}\right]/\bar{c} \end{array}\right. \end{aligned} \tag{11}$$

choosing

$$\bar{\mathbf{a}} = 2\,\mathbf{\Lambda}\,\textstyle\sum_{j=1}^N \mathbf{x}_j/N,\ \bar{b} = \sum_{j=1}^N [||\mathbf{x}_j||_{\mathbf{\Lambda}}^2 - \theta_j]/N \text{ and } \bar{c} = 4\,n$$

while

$$\mathbf{x}_i = \mathbf{\Lambda}^{-1}\,(\bar{c}\,\mathbf{a}_i + \bar{\mathbf{a}})/2 \text{ and } \theta_i = ||\bar{c}\,\mathbf{a}_i + \bar{\mathbf{a}}||_{\mathbf{\Lambda}^{-1}}^2/4 - (\bar{c}\,b_i + \bar{b})$$

so that each proximity is parameterized by $\mathbf{a}_i$ and b_i. Here we:

1. consider the opposite of the squared distance $||\mathbf{x}-\mathbf{x}_i||_{\mathbf{\Lambda}}^2$, say the *proximity*, to the prototype $\mathbf{x}_i$ for the chosen metric,

2. *thresholded* by θ_i in order to :
   - control the relative influence of each prototype (the higher θ_i the higher the proximity to the ith prototype) and also to :
   - obtain a one to one correspondence between the linear function parameters $(\mathbf{a}_i, b_i)$ and the prototype data and threshold $(\mathbf{x}_i, \theta_i)$ up to an indetermination parameterized by $(\bar{\mathbf{a}}, \bar{b}, \bar{c})$,

3. add the same quantity $||\mathbf{x}||^2_{\mathbf{\Lambda}} - \bar{\mathbf{a}}^T \mathbf{x} + \bar{b}$ to each $c_i(\mathbf{x})$ and multiply by a common positive constant $\bar{c}$ so that:
   - the comparison in (1) is not modified, while:
   - adding $||\mathbf{x}||^2_{\mathbf{\Lambda}}$ allows to cancel quadratic terms in (11) and obtain a linear function and
   - indetermination[20] in the definition of $(\mathbf{a}_i, b_i)$ is canceled, verifying the constraints:
   $$\sum_{i=1}^{N} \mathbf{a}_i = \mathbf{0} \text{ and } \sum_{i=1}^{N} b_i = 0 \tag{12}$$
   so that $(\mathbf{a}_i, b_i)$ are *centered* while $\sum_{i=1}^{N} ||\mathbf{a}_i||^2$ is minimal[21] w.r.t. $\bar{\mathbf{a}}$.

As a consequence we obtain a linear classifier and frontiers F_{ij} between categories of index i and j are *piece-wise planar hyper-surfaces* of equation:

$$F_{ij} = \{\mathbf{x}, c_i(\mathbf{x}) - c_j(\mathbf{x}) = [\mathbf{a}_i - \mathbf{a}_j]^T \mathbf{x} + [b_i - b_j] = 0\}$$

as illustrated in Fig. 4. The distance from a prototype to the category frontier (geometrical margin) writes:

$$d(\mathbf{x}_i, F_{ij}) = 1/2\ [||\mathbf{x}_i - \mathbf{x}_j||_{\mathbf{\Lambda}} + [\theta_i - \theta_j]/||\mathbf{x}_i - \mathbf{x}_j||_{\mathbf{\Lambda}}]$$

as easily derived[22] since $\bar{c}\,||\mathbf{a}_i - \mathbf{a}_j||_{\mathbf{\Lambda}^{-1}} = 2\,||\mathbf{x}_i - \mathbf{x}_j||_{\mathbf{\Lambda}}$.

Properties and limitations of NN classifiers.
In the statistical interpretation of NN classifiers (e.g. (Theodoridis and Koutroumbas, 1999)), under "reasonable" assumptions, i.e. normal distribution of the data in each category with similar covariances, this corresponds to a Bayesian classifier, writing

$$\theta_i = 2\,log(p(r_i)) < 0 \text{ where } p(r_i) = e^{\theta_i/2} / \textstyle\sum_j e^{\theta_j/2}$$

[20] *Invariance in the arg-max equation*: In equation (1), the reader can easily verify that any strictly increasing transformation $t : \mathcal{R} \to \mathcal{R}$ of the proximities $c_i(\mathbf{x})$, i.e. $c_i(\mathbf{x}) \to t(c_i(\mathbf{x}))$ will not change the comparisons. On the reverse, if $t()$ is not a strictly increasing transformation comparisons may be modified for some $c_i(\mathbf{x})$.
The most general transformation is thus a composition with a strictly increasing function.
Now, if we want to preserve the linearity, i.e. that $c_i(\mathbf{x})$ being linear, $t(c_i(\mathbf{x}))$ is still linear, this transformation must be linear, the only solution being of the form $t(c_i(\mathbf{x})) = [c_i(\mathbf{x}) - (\bar{\mathbf{a}}^T \mathbf{x} - \bar{b})]/\bar{c}$ with $\bar{c} > 0$.
We also observe that this classifier with $N\,(n+1)$ components has $(N-1)\,(n+1) - 1$ independent parameter components, i.e. degrees of freedom, because $n+2$ parameters are constrained via the choice of $\bar{\mathbf{a}}$, $\bar{b}$ and $\bar{c}$.

[21] If we consider the criterion $\min_{\bar{\mathbf{a}}} ||\mathbf{a}'_i||^2$ with $\mathbf{a}'_i = \mathbf{a}_i - \bar{\mathbf{a}}$ the related normal equation is precisely $\sum_{i=1}^{N} \mathbf{a}'_i = \mathbf{0}$.

[22] *Distance to an hyper-plane.* The distance $d(\mathbf{x}, P)$ is the minimal value $||\mathbf{x} - \mathbf{z}||_{\mathbf{\Lambda}}$ for a point $\mathbf{z} \in P$, i.e. with $\mathbf{a}^T \mathbf{z} - b = 0$. Writing this as a criterion:

$$\min_{\mathbf{z}} \max_{\lambda}\ \tfrac{1}{2}||\mathbf{x} - \mathbf{z}||^2_{\mathbf{\Lambda}} + \lambda\,(\mathbf{a}^T \mathbf{z} - b)$$

the normal linear equations yield the formula: $d(\mathbf{x}, P) = |\mathbf{a}^T \mathbf{x} - b| / ||\mathbf{a}||_{\mathbf{\Lambda}^{-1}}$

is the a-priori probability for a given data to belong the ith sub-category.

Beside being conceptually extremely simple and also obvious to implement in practice, this well-known classifier (Duda et al., 2000) has reasonable performances. More precisely, the probability of error for the nearest neighbor rule (i.e. with $\theta_i = 0$ in (11)) given enough members in the training set is sufficiently close to the Bayes (optimal) probability of error. It has been shown (Cover and Hart, 1967) that as the size of the training set goes to infinity, the asymptotic nearest neighbor error is never two times worse than the Bayes (optimal) error. However, calibration or training set sizes never go to infinity! The real problem is to understand the performances of the classifier for a *limited* calibration set, as discussed in this paper.

Furthermore, the present approach does not provide, as it, any "modelization" of the calibration set. As a consequence, since no prediction/inference is possible with this method, the quality of the training is highly dependent upon the calibration set itself. It may not be very "accurate" with respect to data which are not calibration data, i.e. generalization performances are expected to be poor (Vapnik, 1995).

Another traditional criticism about NN classifiers pointed at large space requirement to store the entire calibration set and the seeming necessity to query the entire calibration set in order to make a single membership classification. There has been considerable interest in *editing* the training or calibration set in order to reduce its size (e.g.: proximity graphs, Delaunay triangulation) eliminating "redundant" data (see (Duda et al., 2000) for a review). Such editing mechanisms only delete redundant prototypes, whereas a much general mechanism is proposed here.

B. Considering bounded parameters of limited precision.

As noticed, e.g. in (Gaspard and Viéville, 2000), at the *specification level*, a "physical" parameter is always represented though a vector of bounded quantities, $x^i, x^i_{min} \leq x^i \leq x^i_{max}$ with a finite precision x^i_ϵ so that there is a finite range of significant values. This finite range size is $\sigma^i = \left[(x^i_{max} - x^i_{min})/x^i_\epsilon\right]$. This specification also applies, up to the 1st order, to non-linear combinations[23] of parameters.

[23] *The bound and 1st order precision of a monomial.* We also consider non-linear combinations of parameters, using rescaled monomial $m^\alpha = [\prod_{i=1}^n (x^i)^{\alpha_i}]/\bar{m}$ of degrees $\alpha = (\cdots, \alpha_i, \cdots)$ bounded by $|m^\alpha| \leq \sigma^\alpha$, with:

$$\sigma^\alpha = 1/\sum_{j=1}^n \alpha_j/(\sigma^j/2)^{\alpha_j} \text{ and } \bar{m} = \prod_{i=1}^n (\sigma^i/2)^{\alpha_i}/\sigma^\alpha$$

easily derived because on one hand, since $|x^i| \leq \sigma^i/2$ it is straightforward to derive

Using the transformation $x^i \to x^i/x^i_\epsilon - c^i$ with $c^i = (x^i_{max}+x^i_{min})/(2\,x^i_\epsilon)$ from now on and without any loss of generality we can consider $x^i_\epsilon = 1$ and that the quantity is bounded[24] by $|x^i| \leq \sigma^i/2$: quantities are now *centered and rescaled with respect to their precision.*

Such a precision is in practice very easy to estimate (e.g. 1 mm for a pupil ruler, 1 deg for a protractor, 1 pixel in an image, etc...) and so are bounds. These quantities are *not* precise numbers but orders of magnitude.

Following this track, two parameters x^i and x'^i can be considered *distinct* only if[25]:
$|x^i - x'^i| > 2$
Otherwise, we cannot decide whether (i) these values are the same or (ii) differ by a quantity too small to be measurable. In the latter case, we can not say that they are equal, but *indistinguishable.*

A step further, a vectorial *centered rescaled* parameters are *bounded* by[26]

$$||\mathbf{x}||_\infty \leq \max_i \sigma^i/2 \text{ and } ||\mathbf{x}|| \leq D = \sqrt{\sum_i (\sigma^i)^2}/2 \qquad (13)$$

e.g. if all sizes $\sigma^i = \sigma$ are equal $D = \sqrt{n}\,\sigma/2$.

Two vectorial parameters are indeed *distinguishable* if at least one component is distinguishable: $|x^i - x'^i| > 2$ for some i (i.e. $||\mathbf{x}-\mathbf{x}'||_\infty > 2$). But what happens if we combine quantities which are "almost distinguishable" ? The data space dimension being in practice very large, we interpret the precision uncertainty[27] as an additive Gaussian noise so that, if $\mathbf{x}$ and $\mathbf{x}'$ correspond to the same quantities, $||\mathbf{x}-\mathbf{x}'||^2$ follows a Ξ-square distribution which expected value is n, so that $||\mathbf{x}-\mathbf{x}'||/\sqrt{n}$

$|m^\alpha| \leq \sigma^\alpha = \prod_{i=1}^n (\sigma^i/2)^{\alpha_i}/\bar{m}$.
On the other hand, considering that precision is a 1st order quantity, since:

$$\partial m^\alpha \simeq 1/\bar{m} \sum_{j=1}^n \alpha_j \prod_{i=1, i\neq j}^n (x^i)^{\alpha_i} (x^j)^{\alpha_j - 1} \partial x^j$$

writing $x^j_\epsilon = |\partial x^j| = 1$ we obtain

$$m^\alpha_\epsilon \leq 1/\bar{m} \sum_{j=1}^n \alpha_j \prod_{i=1}^n (\sigma^i/2)^{\alpha_i}/(\sigma^j/2)^{\alpha_j} = 1$$

which is in fact the tightest bound not dependent on x^i.

[24] Here, $x^i_{min} \leq x^i \leq x^i_{max} \Leftrightarrow |x^i - (x^i_{max} + x^i_{min})/2| \leq (x^i_{max} - x^i_{min})/2$ yields $|x^i/x^i_\epsilon - c^i| \leq \sigma^i/2$ with our notations.

[25] Here, we have to double the value of the bound because each value may vary in a ± 1 range thus their difference may vary in twice this range.

[26] Here, we write $||\mathbf{x}||_\infty = \max_i |x^i|$ and $||\mathbf{x}||^2 = \sum_i (x^i)^2$, derivation of (13) being obvious.

[27] *Interpreting bounded precision as uncertainty:* we assume that if x^i an x'^i are two samples of the same quantity, $\varepsilon^i = x^i - x'^i$ is a normalized centered Gaussian variable of variance 1, so that $|x^i - x'^i| < 2$ with a probability $P > 0.95$. If $|x^i - x'^i| > 2$ we thus can consider that x^i an x'^i likely correspond to different quantities.
In the vectorial case $||\mathbf{x} - \mathbf{x}'||^2 = \sum_i (\varepsilon^i)^2$ follows a Ξ-square distribution with n degrees of freedom, thus of mean n and variance $2\,n$.

is expected to be 1. In coherence to the 1D case, we *distinguish* two quantities when the value is twice the expected value.

Summarizing, we propose to consider the *average quadratic* precision[28], specifying that two vectorial are *distinguishable* if and only if $||\mathbf{x} - \mathbf{x}'||/\sqrt{n} > 2$.

From these specifications we introduce a natural but important constraint for our paradigm: *data from the learning set must be distinguishable*, i.e. two learning data $\mathbf{x}_i$ and $\mathbf{x}_j$ must verify $||\mathbf{x}^i - \mathbf{x}^{'i}|| > 2\sqrt{n}$, the so-called geometrical margin e.g. (Vapnik, 1998) being $\rho = \sqrt{n}$. If two prototypes belonging to different categories are indistinguishable, the corresponding categories are indistinguishable and the problem ill-posed: this situation is to be rejected. A useful relation is

$$D/\rho = 1/2 \sqrt{\sum_i (\sigma^i)^2/n} \tag{14}$$

with $D/\rho = \sigma/2$ if all sizes $\sigma^i = \sigma$ are equal.

Furthermore, in (1), comparisons of the form $c_i(\mathbf{x}) > c_j(\mathbf{x})$ between two proximities are valid if and only if their difference is higher that the related precision. For thresholded nearest-neighbor proximities as defined in (11), $c_i(\mathbf{x}) = [-||\mathbf{x} - \mathbf{x_i}||^2 + \cdots]/\bar{c}$ so that, from what precedes, we must write

$$c_i(\mathbf{x}) > c_j(\mathbf{x}) + \gamma \text{ with } \gamma = [2\sqrt{n}]^2/\bar{c} = 1 \tag{15}$$

Over-simple, such an specification is very useful at both the theoretical and implementation[29] levels.

[28] *About the metric related to data precision.* Here quantities are *rescaled* before computing Euclidean distances, i.e. it writes

$$||\mathbf{x}^i - \mathbf{x}^{'i}|| = \sqrt{\sum_{i=1}^{n} \left[\frac{x^i - x^{'i}}{x^i_\epsilon}\right]^2}$$

This diagonal metric has an obvious statistical interpretation in terms of the inverse of a *covariance* or "quadratic information', e.g. (Vieville et al., 2001), interpreting the data precision as an uncertainty. The precision between two components may also be "coupled" (i.e. correlated), the metric not being diagonal anymore. It is however obvious to *diagonalize* any covariance matrix, considering linear combinations of these components and obtain decoupled components. There is thus no lack of generality with the present "diagonal" approach.

[29] *Physical parameter specification and local estimation.* It has been observed (e.g. (Vieville, 2000)) that there is a real gain to take this experimental specification into account: with such specification, "quasi-static" estimation methods, with step by step variations from an initial estimate towards the problem solution, are powerful strategies for local estimations (adaptations to limited range variations from a default value, interactive estimation where a user given initial estimate is to be refined, efficiency in tracking tasks ...), experimentally more efficient than standard usual

C. Deriving the form of the Yg minimum

In order to solve the minimization problem in (5), we can easily write the Lagrangian of this constrained criterion:

$$\mathcal{L} = \frac{1}{2}\sum_{i}^{N} ||\mathbf{a}_i||^2 + \sum_{i\,j\,k} \alpha_{ijk} \left[(\mathbf{a}_i - \mathbf{a}_j)^T \mathbf{x}_k - (b_i - b_j) - 1 \right] + \alpha \sum_i b_i + \beta^T \sum_i \mathbf{a}_i$$

with the related Kuhn-Tucker[30] conditions:

$$\alpha_{ijk} > 0 \Leftrightarrow \begin{cases} r_i = \arg\max_{r_i = r_k} \mathbf{a}_i^T \mathbf{x}_k + b_i \\ \text{and } r_j = \arg\max_{r_j \neq r_k} \mathbf{a}_j^T \mathbf{x}_k + b_j \\ \text{and } (\mathbf{a}_i - \mathbf{a}_j)^T \mathbf{x}_k - (b_i - b_j) = 1 \end{cases}$$

and the related normal equations:

$$\begin{cases} 0 = \frac{\partial \mathcal{L}}{\partial b_h} = \sum_{i\,k} \alpha_{ihk} - \sum_{j\,k} \alpha_{hjk} + \alpha = \sum_k \alpha_{hk} + \alpha \\ 0 = \frac{\partial \mathcal{L}}{\partial \mathbf{a}_h}^T = \mathbf{a}_h + \sum_{j\,k} \alpha_{hjk} \mathbf{x}_k - \sum_{i\,k} \alpha_{ihk} \mathbf{x}_k + \beta = \mathbf{a}_h - \sum_k \alpha_{hk} \mathbf{x}_k + \beta \end{cases}$$

with $\alpha_{hk} = \sum_i \alpha_{ihk} - \sum_j \alpha_{hjk}$.
The optimal solution of (5) thus writes:

$$\mathbf{a}_h = \sum_k \alpha_{hk} \mathbf{x}_k - \beta \text{ with } \sum_k \alpha_{hk} + \alpha = 0 \tag{16}$$

methods, because the stability of the estimation process is easy to control in this case. Furthermore, the estimation is stopped as soon as the required precision is obtained, whereas for standard methods, convergence to a non-negligible precision only is not so easy to obtain, so that overhead occurs. This mechanism is used in our implementation.

[30] *On Kuhn-Tucker conditions.* We consider for the purpose of this derivation, weak inequalities ($\geq$) instead of strict inequalities ($>$). The Kuhn-Tucker conditions state that the Lagrangian multiplier α_{ijk} (i) vanishes if and only if the inequality is strictly verified and (ii) is positive if the inequality is verified as an equality. In practice, this bound is numerically never attained.