Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE @ DIRECT®
Neurocomputing 56 (2004) 305 – 327

NEUROCOMPUTING
www.elsevier.com/locate/neucom

Combining protein secondary structure prediction models with ensemble methods of optimal complexity

Yann Guermeur[a,*], Gianluca Pollastri[b], André Elisseeff[c], Dominique Zelus[d], Hélène Paugam-Moisy[e], Pierre Baldi[b]

[a] *Campus Scientifique, BP 239, LORIA-Université Henri Poincaré, 54506 Vandoeuvre-lès-Nancy Cedex, France*
[b] *Department of Information and Computer Science, Institute for Genomics and Bioinformatics, University of California, Irvine, CA 92697-3425, USA*
[c] *MPI for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany*
[d] *Wiener Lab, CIBIO, Av Presidente Juan D. Perón 2991, 2000 Rosario, Argentina*
[e] *ISC, UMR CNRS 5015, Université Lumière Lyon 2, 67 Boulevard Pinel, 69675 Bron Cedex, France*

Received 19 May 2003; received in revised form 28 June 2003; accepted 6 October 2003

Abstract

Many sophisticated methods are currently available to perform protein secondary structure prediction. Since they are frequently based on different principles, and different knowledge sources, significant benefits can be expected from combining them. However, the choice of an appropriate combiner appears to be an issue in its own right. The first difficulty to overcome when combining prediction methods is overfitting. This is the reason why we investigate the implementation of Support Vector Machines to perform the task. A family of multi-class SVMs is introduced. Two of these machines are used to combine some of the current best protein secondary structure prediction methods. Their performance is consistently superior to the performance of the ensemble methods traditionally used in the field. They also outperform the decomposition approaches based on bi-class SVMs. Furthermore, initial experimental evidence suggests that their outputs could be processed by the biologist to perform higher-level treatments.

Keywords: Protein secondary structure prediction; Multi-class support vector machines (M-SVMs); Ensemble methods; Hierarchical sequence processing systems

[*] Corresponding author. Tel.: +33-03-83-59-30-18; fax: +33-03-83-41-30-79.

E-mail addresses: yann.guermeur@loria.fr (Y. Guermeur), gpollast@ics.uci.edu (G. Pollastri), andre.elisseeff@tuebingen.mpg.de (A. Elisseeff), dzelus@wiener-lab.com.ar (D. Zelus), hpaugam@isc.cnrs.fr (H. Paugam-Moisy), pfbaldi@ics.uci.edu (P. Baldi).

0925-2312/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2003.10.004

1. Introduction

Since the early 1960s, model combination has proved to be an efficient alternative to model selection for a wide range of statistical inference problems. During the last decade, theory in the field has made rapid strides, especially for discrimination. Classifier combination is thus currently endowed with a rich theoretical framework, which is very useful... as long as the problem at hand satisfies the underlying hypotheses. Unfortunately, in many real-life situations, the practitioner is faced with the worst configuration one can think of: pretrained experts with different types of outputs, highly correlated errors, few training data, etc. In this context, the problem to be solved primarily consists in finding a combiner of adequate complexity, so that, with high probability, the training error will constitute an estimate reliable enough of the generalization error, and the gain in prediction accuracy, small as it should be, will be "guaranteed". This is precisely the type of situations for which Support Vector Machines (SVMs) have been conceived.

The aim of the support vector method (see for instance [10,12,69,31]), is to maximize the generalization capabilities of kernel machines [1,50,66], by minimizing an upper bound on the *expected risk* (or generalization error), named the *guaranteed risk*, with respect to the values of the model parameters. This inductive principle, called Structural Risk Minimization (SRM) [68], is implemented by the SVMs developed to estimate indicator or real-valued functions. However, this is no longer the case with the extensions of the support vector method which were initially proposed for multi-class discrimination [72,69,11]. Indeed, they are not related, at least explicitly, to a guaranteed risk, which makes it impossible to characterize a satisfactory compromise between training performance and complexity. Building upon the uniform strong law of large numbers introduced in [18], and extended in [28,26], we specified in [27,25] a new family of multi-class SVMs (M-SVMs). Two of these SVMs are assessed as classifier combiners on an open problem in structural biology: protein secondary structure prediction.

Building new prediction methods by fusion of existing ones appears particularly relevant for protein secondary structure prediction. Two main reasons can be put forward to support this assertion. First, the numerous methods already available to predict the secondary structure (see [24,60,4,59] for surveys) are based on different principles. Second, they use, in addition to the amino acid sequences (or profiles of multiple alignments), data from different knowledge sources. Consequently, whenever secondary structure is to be predicted, several sets of conformational scores are available, which can be expected not to be utterly correlated. Indeed, most of the current best prediction systems implement conformational score combinations, which can take many forms [8,61,5,16,51]. However, the choice of a particular combiner is hardly ever justified, and the gain resulting from the combination is seldom significantly superior to the one resulting from a simple averaging. A first attempt to overcome these limitations was described in [29]. The aim of this work is to establish that noticeable benefits can spring from combining protein secondary structure prediction models with M-SVMs.

The organization of the paper is as follows. Section 2 is devoted to the study of the generalization capabilities of multi-class discriminant models, i.e. the theory on which our work is grounded. Section 3 outlines the pathway through which M-SVMs can be derived from these bounds, as a direct implementation of the SRM inductive principle. Two machines are considered more specifically. Their implementation for very large database processing is detailed in Section 4. They are first assessed in Section 5, to combine three standard secondary structure prediction methods: SOPMA, GOR and SIMPA. They are then used in Section 6 to combine the experts that constitute one of the current best methods, SSpro.

2. Guaranteed risk for multi-class discriminant models

The theoretical framework of this study is Vapnik's statistical learning theory [69]. We are concerned with the case of Q-category pattern recognition problems, when $Q \geqslant 3$. Let $\mathscr{X}$ be the space of description and $\mathscr{C}$ the set of categories. We make the standard assumption that there is a joint probability distribution, fixed but unknown, on $\mathscr{X} \times \mathscr{C}$. The goal is to find, in a given set of multivariate regression functions $\mathscr{H} = \{h\}$, a function with lowest error rate (the corresponding discrimination function must be as close as possible to Bayes' decision rule). The decision function associated with $h=[h_k]$ is obtained by assigning each pattern x to the category C_k in $\mathscr{C}$ satisfying: $h_k(x) = \max_l h_l(x)$. In the common case where the $h_k(x)$ are estimates of the class posterior probabilities (see [57] for a characterization of this situation), choosing this decision function simply amounts to implementing Bayes' estimated decision rule. Hereafter, $C(x_i)$ will denote indifferently the category of pattern x_i, or the index of this category. In that context, the main uniform convergence result we established is based on the following definitions.

Definition 1 (ε-cover or ε-net). Let (E, ρ) be a pseudo-metric space, and $B(v, r)$ the closed ball of centre v and radius r in E. Let H be a subset of E. An ε-cover of H is a subset $\bar{H}$ of E such that:

$$H \subset \bigcup_{v \in \bar{H}} B(v, \varepsilon).$$

Definition 2 (Covering numbers). Let (E, ρ) be a pseudo-metric space. If $H \subset E$ has an ε-cover of finite cardinality, then its *covering number* $\mathscr{N}(\varepsilon, H, \rho)$ is the smallest cardinality of its ε-covers. If there is no such finite cover, then the covering number in defined to be ∞.

Definition 3. Let $\mathscr{H}$ be a set of functions from $\mathscr{X}$ into $\mathbb{R}^Q$. For a set s of points in $\mathscr{X}$, define the pseudo-metric $d_{l_\infty, l_\infty(s)}$ on $\mathscr{H}$ as: $\forall (h, \bar{h}) \in \mathscr{H}^2, d_{l_\infty, l_\infty(s)}(h, \bar{h}) = \max_{x \in s} \max_k |h_k(x) - \bar{h}_k(x)|$.

Definition 4. Let $\mathscr{H}$ be a set of functions from $\mathscr{X}$ into $\mathbb{R}^Q$ and h a function of $\mathscr{H}$. Define $\Delta h = [\Delta h_k]$, $(1 \leqslant k \leqslant Q)$, as the function from $\mathscr{X}$ into $\mathbb{R}^Q$ satisfying

$$\Delta h_k(x) = \tfrac{1}{2} \left\{ h_k(x) - \max_{l \neq k} h_l(x) \right\}.$$

For $\gamma \in (0,1]$, let $\pi_\gamma : \mathbb{R} \to [-\gamma, \gamma]$ be the piecewise-linear squashing function defined as

$$\pi_\gamma(x) = \begin{cases} \gamma\, sign(x) & \text{if } |x| \geqslant \gamma, \\ x & \text{otherwise} \end{cases}$$

$\forall h \in \mathscr{H}$, $\Delta h^\gamma = [\Delta h_k^\gamma] = [\pi_\gamma \circ \Delta h_k]$, $(1 \leqslant k \leqslant Q)$. $\Delta\mathscr{H}^\gamma = \{\Delta h^\gamma : h \in \mathscr{H}\}$. For all N in $\mathbb{N}^*$ and all γ in $(0,1]$, let $\mathscr{N}_{\infty,\infty}(\gamma/2, \Delta\mathscr{H}^\gamma, N) = \max_{s_N \in \mathscr{X}^N} \mathscr{N}(\gamma/2, \Delta\mathscr{H}^\gamma, d_{l_\infty, l_\infty(s_N)})$. Extending a definition from Bartlett [6], we introduced the following definition:

Definition 5. The empirical risk with margin $\gamma \in (0,1]$ on a training set s_N of size N is

$$R^\gamma_{s_N}(h) = \frac{1}{N} \left|\{(x_i, C(x_i)) \in s_N : \Delta h_{C(x_i)}(x_i) < \gamma\}\right|$$

With these definitions at hand, extending Lemma 4 and Corollary 9 from [6], as well as the basic lemma of Theorem 4.1 in [69], we established in [26] (see also [28]) the following theorem:

Theorem 1. *Let $s_N = \{(x_i, C(x_i))\}$, $(1 \leqslant i \leqslant N)$, be a set of labeled examples, iid according to the joint distribution on $\mathscr{X} \times \mathscr{C}$ characterizing the problem of interest. With probability at least $1 - \delta$, for every value of γ in $(0,1]$, the risk $R(h)$ of a function h computed by a numerical Q-class discriminant model $\mathscr{H}$ trained on s_N is bounded above by*

$$R(h) \leqslant R^\gamma_{s_N}(h) + \sqrt{\frac{1}{2N}\left(\ln(2\mathscr{N}_{\infty,\infty}(\gamma/2, \Delta\mathscr{H}^\gamma, 2N)) + \ln\left(\frac{2}{\gamma\delta}\right)\right)} + \frac{1}{N}. \qquad (1)$$

Note that a similar result, based on a different pseudo-metric, can also be found in [18].

3. M-SVMs

Theorem 1 applies to any multi-class discriminant system obtained by combining a multivariate model with Bayes' estimated decision rule. In this section, we turn to the specific case of multi-class SVMs. The study of the standard (bi-class) SVMs is usually performed in two steps: first, the linear case, corresponding to the specification of the maximal margin hyperplane, then the non-linear one, by introduction of kernels satisfying Mercer's conditions. The reference on the subject is [69]. In the same way as a "linear" SVM shares the architecture of the perceptron, a linear M-SVM is a multivariate linear (or more precisely affine) regression model, i.e. a set of hyperplanes of cardinality equal to the number of classes.

3.1. Linear M-SVMs

We assume that the data live in $\mathbb{R}^d$. Let $\mathscr{H}$ be the family of functions h considered, with:

$$\forall x \in \mathscr{X},\ h(x) = Wx + b = \begin{bmatrix} w_1^{\mathrm{T}} \\ \vdots \\ w_k^{\mathrm{T}} \\ \vdots \\ w_Q^{\mathrm{T}} \end{bmatrix} x + \begin{bmatrix} b_1 \\ \vdots \\ b_k \\ \vdots \\ b_Q \end{bmatrix}.$$

The capacity measure appearing in (1) can be expressed in various ways as a function of the constraints on the couple (W, b). In a nutshell, there are two main strategies to bound such covering numbers. The first one consists in making use of so-called generalized Sauer's lemmas to relate them to extended notions of Vapnik–Chervonenkis (VC) dimensions, such as the *graph dimension* [49] or the *fat-shattering dimension* [41]. An illustration of this strategy can be found in [34,2]. Alternatively, one can bound their values directly, thanks to functional analysis results such as those sketched out in [73]. Both approaches have been extended to the case of multi-class discriminant models in [18,28,26]. From these different bounds, different *structures* on the family of functions $\mathscr{H}$ can be built, corresponding to different implementations of the SRM inductive principle. These implementations define as many M-SVMs. A detailed account of the process according to which training procedures can be derived directly from (1) is given in [25]. We restrict here to two models with appealing statistical properties, which are used in the forthcoming experiments. The structure of the first one (M-SVM1) is associated with the values of the sum $\sum_{1 \leqslant k < l \leqslant Q} \|w_k - w_l\|^2$. The training procedure thus consists in solving the following quadratic programming (QP) problem:

Problem 1.

$$\min_{h \in \mathscr{H}} \left\{ \frac{1}{2} \sum_{1 \leqslant k < l \leqslant Q} \|w_k - w_l\|^2 + C \sum_{i=1}^{N} \sum_{k=1}^{Q} \xi_{ik} \right\}$$

$$\text{s.t.} \begin{cases} (w_{C(x_i)} - w_k)^{\mathrm{T}} x_i + b_{C(x_i)} - b_k \geqslant 1 - \xi_{ik},\ (1 \leqslant i \leqslant N), \\ \quad (1 \leqslant k \neq C(x_i) \leqslant Q), \\ \xi_{ik} \geqslant 0, \quad (1 \leqslant i \leqslant N),\ (1 \leqslant k \neq C(x_i) \leqslant Q), \\ \sum_{k=1}^{Q} w_k = 0_d. \end{cases}$$

As usual, the non-negative slack variables ξ_{ik} have been introduced to take into account the fact that the data could be "non-multilinearly" separable. Their values characterize the empirical risk (the $\xi_{iC(x_i)}$ are dummy variables systematically equal to 0). Note that this first model has been proposed, independently and under different formulations, by several teams [72,69,11,27]. Another model with appealing properties (M-SVM2) results from specifying the structure as a function of $\max_{k<l} \|w_k - w_l\|^2$. M-SVM2 is the solution of the following QP problem:

Problem 2.

$$\min_{h \in \mathscr{H}} \left\{ \frac{1}{2} t^2 + C \sum_{i=1}^{N} \sum_{k=1}^{Q} \xi_{ik} \right\}$$

$$\text{s.t.} \begin{cases} \|w_k - w_l\|^2 \leqslant t^2, \quad (1 \leqslant k < l \leqslant Q), \\ \text{Constraints of Problem1.} \end{cases}$$

The choice between this model and the former one can for instance be based on the knowledge available regarding the domain in which the data lives.

3.2. Specification of the training procedure

For the sake of simplicity, and without loss of generality, in what follows, we focus on the case of M-SVM1. The theoretical and technical reasons which enforce to solve the QP problems associated with standard SVMs in dual form still apply here. Let α be the vector of dual variables. The objective function becomes

$$J(\alpha) = \frac{1}{2Q} \left\{ \begin{array}{l} \displaystyle\sum_{i \simeq j} \sum_{k=1}^{Q} \sum_{l=1}^{Q} \alpha_{ik}\alpha_{jl} x_i^{\mathrm{T}} x_j - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{Q} \alpha_{ik}\alpha_{jC(x_i)} x_i^{\mathrm{T}} x_j \\ \displaystyle + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{Q} \alpha_{ik}\alpha_{jk} x_i^{\mathrm{T}} x_j \end{array} \right\}$$

$$- \sum_{i=1}^{N} \sum_{k=1}^{Q} \alpha_{ik}$$

$$= \frac{1}{2} \alpha^{\mathrm{T}} H \alpha - 1_{(Q-1)N}^{\mathrm{T}} \alpha,$$

where $i \simeq j$ expresses the fact that x_i and x_j belong to the same category. The problem corresponding to M-SVM1 is:

Problem 3.

$$\min_{\alpha} \{J(\alpha)\}$$

$$\text{s.t.} \begin{cases} \displaystyle\sum_{x_i \in C_k} \sum_{l=1}^{Q} \alpha_{il} - \sum_{i=1}^{N} \alpha_{ik} = 0 & (1 \leqslant k \leqslant Q-1), \\ 0 \leqslant \alpha_{ik} \leqslant C & (1 \leqslant i \leqslant N),\ (1 \leqslant k \leqslant Q),\ k \neq C(x_i). \end{cases}$$

3.3. Non-linear M-SVMs

Non-linear M-SVMs can be constructed by using different kernel functions k satisfying Mercer's conditions [1], in the same way as Vapnik and co-workers did. In practice, this simply amounts to replacing in the objective function $J(\alpha)$ of Problem 3 the inner products $x_i^{\mathrm{T}} x_j$ with the convolutions of the inner products $k(x_i, x_j)$. The same transform applies to compute the outputs of the machine. For instance, the equation of the hyperplane separating categories C_k and C_l will be:

$$\frac{1}{Q}\left\{\sum_{x_i \in C_k} \sum_{m=1}^{Q} \alpha_{im} k(x_i, x) - \sum_{i=1}^{N} (\alpha_{ik} - \alpha_{il}) k(x_i, x) - \sum_{x_i \in C_l} \sum_{m=1}^{Q} \alpha_{im} k(x_i, x)\right\} + b_k - b_l = 0. \tag{2}$$

The fact that the dimension of the feature space could be infinite, for instance if the kernel is Gaussian, rises no theoretical difficulty. More precisely, bounding the covering numbers of interest, let it be directly [73,18,32], or through an extension of the VC dimension [7,33,26], can still be done in the infinite dimensional case.

4. Implementation of M-SVMs

The choice of an optimization method is an issue in its own right, since dealing with Q classes multiplies the number of (dual) variables by $(Q-1)$. Indeed, although an algorithm devised to train a simplified variant of M-SVM1 has proved efficient on databases made up of tens of thousands examples [36], to the best of our knowledge, the experiments reported in this paper constitute the first assessment of M-SVMs on a large real-world problem. The size of the databases described in Sections 5.1 and 6.2 is far too big for classical solvers to be directly applied. To overcome this limitation, we implemented the iterative method introduced in [17], and derived from the Frank–Wolfe algorithm [19]. The basic idea was to linearize the problem in order to reduce the requirements in terms of memory. The algorithm also includes a decomposition method. For the sake of simplicity, we present both parts separately in the following subsections.

4.1. Frank–Wolfe algorithm

The Frank–Wolfe algorithm applies to problems with linear constraints of the form:

$$\min_z f(z)$$

$$\text{s.t.} \begin{cases} Az = b, \\ z \geqslant 0. \end{cases}$$

It is an iterative method which generates, starting from a feasible solution $z^{(0)}$, a series of points $z^{(0)}, z^{(1)}, \ldots, z^{(k)}, \ldots$ where, for each k, $z^{(k+1)}$ is derived from $z^{(k)}$ as follows:

(1) Solve the linear programming (LP) problem $LP(z^{(k)})$ given by

$$\min_t \{\nabla f(z^{(k)})^{\mathrm{T}} t\}$$

$$\text{s.t.} \begin{cases} At = b, \\ t \geqslant 0. \end{cases}$$

(2) Let $t^{(k)}$ be a vertex of the polyhedron optimal solution of $LP(z^{(k)})$. Then $z^{(k+1)}$ is chosen so as to minimise f on the segment $[z^{(k)}, t^{(k)}]$.

Applying the algorithm to M-SVM1 is straightforward. We can for instance set $\alpha^{(0)} = 0_{(Q-1)N}$. The LP program to be solved is:

Problem 4.

$$\min_\gamma \{\nabla J(\alpha^{(k)})^{\mathrm{T}} \gamma\},$$

$$\text{s.t.} \begin{cases} \displaystyle\sum_{x_i \in C_k} \sum_{l=1}^{Q} \gamma_{il} - \sum_{i=1}^{N} \gamma_{ik} = 0 & (1 \leqslant k \leqslant Q-1) \\ 0 \leqslant \gamma_{ik} \leqslant C & (1 \leqslant i \leqslant N),\ (1 \leqslant k \leqslant Q),\ k \neq C(x_i) \end{cases}$$

with

$$\nabla J(\alpha^{(k)})^{\mathrm{T}} \gamma = {\alpha^{(k)}}^{\mathrm{T}} H \gamma - 1_{(Q-1)N}^{\mathrm{T}} \gamma.$$

Let $\theta^{(k)} \in [0,1]$ be the coefficient of the optimal convex combination between $\alpha^{(k)}$ and $\gamma^{(k)}$, i.e.

$$\theta^{(k)} = \operatorname*{Argmin}_{\theta \in [0,1]} J((1-\theta)\alpha^{(k)} + \theta\gamma^{(k)}).$$

After some algebra, it comes:

$$\theta^{(k)} = \min \left\{ \frac{\nabla J(\alpha^{(k)})^{\mathrm{T}} \delta^{(k)}}{\delta^{(k)\mathrm{T}} H \delta^{(k)}}, 1 \right\},$$

where $\delta^{(k)} = \alpha^{(k)} - \gamma^{(k)}$.

4.2. Decomposition method

The main difficulty faced when solving directly Problem 3, let it be with the Frank–Wolfe algorithm or another standard algorithm, springs from the handling of the Hessian matrix H. On the one hand, this matrix can be too large to be stored in memory, since it belongs to $M_{(Q-1)N}(\mathbb{R})$. On the other hand, computing its components can be time consuming, since it involves the computation of the components $k(x_i, x_j)$ of the Gram matrix. A natural way to overcome these difficulties consists in using a decomposition method. This approach was already implemented in the initial study on SVMs by Vapnik and co-workers [10]. The *chunking* method they used was introduced in [68], for the case of a linear model. The main decomposition techniques introduced afterwards (see [65,15] for a review), consist in solving the dual problem while the values of part of the variables are fixed. This general framework is detailed below for Problem 4.

To simplify notations, and without loss of generality, we make the hypothesis that the working set is made up of the dual variables α_B associated with the N_B first examples in the training set, the dual variables α_H associated with the $N_H = N - N_B$ last examples being fixed. The objective function can then be rewritten as follows:

$$J(\alpha) = \frac{1}{2} \begin{bmatrix} \alpha_B \\ \alpha_H \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} H_{BB} & H_{BH} \\ H_{HB} & H_{HH} \end{bmatrix} \begin{bmatrix} \alpha_B \\ \alpha_H \end{bmatrix} - 1_{(Q-1)N}^{\mathrm{T}} \begin{bmatrix} \alpha_B \\ \alpha_H \end{bmatrix}.$$

This functional can still be rewritten as follows:

$$J(\alpha) = \tfrac{1}{2}\, \alpha_B^{\mathrm{T}} H_{BB} \alpha_B - (1_{(Q-1)N_B}^{\mathrm{T}} - \alpha_H^{\mathrm{T}} H_{HB})\alpha_B + \tfrac{1}{2}\, \alpha_H^{\mathrm{T}} H_{HH} \alpha_H - 1_{(Q-1)N_H}^{\mathrm{T}} \alpha_H.$$

We have thus:

$$\nabla J(\alpha_B) = H_{BB}\alpha_B + H_{BH}\alpha_H - 1_{(Q-1)N_B} = [\, H_{BB} \quad H_{BH} \,]\alpha - 1_{(Q-1)N_B}.$$

This last formula highlights the fact that the expression of the gradient of the objective function with respect to the variables of the working set remains unchanged. As a consequence, the time required to compute the "partial" gradient (main part of step (1) of the Frank–Wolfe algorithm) is equal to N_B/N times the time required to compute the whole gradient. The gain in the time required to compute the optimal learning rate (step (2) of the algorithm) is even larger, since the new denominator to be computed is

$$\{\gamma_B^{(k)} - \alpha_B^{(k)}\}^{\mathrm{T}} H_{BB} \{\gamma_B^{(k)} - \alpha_B^{(k)}\}.$$

The number of terms in this quadratic form is proportional to N_B^2 instead of N^2.

The efficiency of a decomposition method is obviously dependent on the way the working set is selected. The interested reader will find in [65,15] detailed surveys of the most popular possibilities. Our software, available through the website of kernel machines,[1] implements several of them.

[1] http://www.kernel-machines.org/index.html

5. Combining SOPMA, GOR and SIMPA

5.1. Experimental protocol

A first assessment of the two M-SVMs was obtained as an extension of the second set of experiments described in [29]. For the sake of completeness, we briefly summarize here the corresponding protocol. It consists in combining the outputs of three of the most widely used secondary structure prediction methods: SOPMA [23], GOR IV [21] and SIMPA96 [46]. The resulting predictions are compared with those of majority voting, a weighted average, optimal with respect to the least-squares criterion, a Multi-Layer Perceptron (MLP) (see for instance [9]) and the Multivariate Linear Regression Combiner (MLRC) introduced in [29]. There are three main ways to perform multi-category discrimination with binary pattern recognition SVMs. Historically, the first of them was the *one*-*against*-*all* (pairwise) method, implemented for instance in [64,69]. Then came the *one*-*against*-*one* approach [20,42]. The most recent one is the DAGSVM of Platt and co-workers [53]. To make the assessment more relevant, we added to the list of aforementioned combiners of reference the *one*-*against*-*all* method and the DAGSVM. The MLR combiner requires the outputs of the experts to be class posterior probability estimates. In order to compare the combiners in a fair way, the outputs of the base classifiers are thus preliminary post-processed with the structure-to-structure filtering neural network described in [29]. The corresponding architecture is depicted in the lower part of Fig. 1.

To constitute the training and test sets, a release of the PDBSELECT database [35], containing 629 chains made up of 147,518 residues, G629, is used. Secondary structure assignment was performed with the DSSP program [40]. The reduction from 8 to 3 conformational states was derived according to the CASP method, i.e. $\mathrm{H} + \mathrm{G} \rightarrow \mathrm{H}$ (α-helix), $\mathrm{E} + \mathrm{B} \rightarrow \mathrm{E}$ (β-strand), and all the other states in C (aperiodic or coil). This assignment is known to be somewhat harder to predict than the other ones used in the literature (see for instance [16]). In order to obtain unbiased estimates of the prediction accuracy, a variant of *stacked generalization* [74] is applied, to train in sequence the filtering networks and the combiners. The database is divided into seven disjoint parts of roughly equal size. Based on this splitting, a 2-stage cross-validation procedure is implemented. Each subset is iteratively used as test set. The six remaining sets are then grouped by three, to constitute disjoint training sets for the filtering networks and the combiners. In this variant of stacked generalization, the leave-one-out cross-validation procedure is thus replaced with a 7-fold cross-validation. Prediction accuracy is assessed by means of four standard measures: the percentage of correctly predicted residues Q_3 for a three-state description of secondary structure (helix, extended and aperiodic), Pearson's/Matthews' correlation coefficient C [48], the segment overlap measure Sov [62,75] and the standard deviation in the secondary structure content σ. The Sov measure is implemented in its initial version (Sov'94) in order to make the new results readily comparable with the former ones. Figures characterizing the behavior of the individual methods, before and after filtering, have been gathered in Table 1.

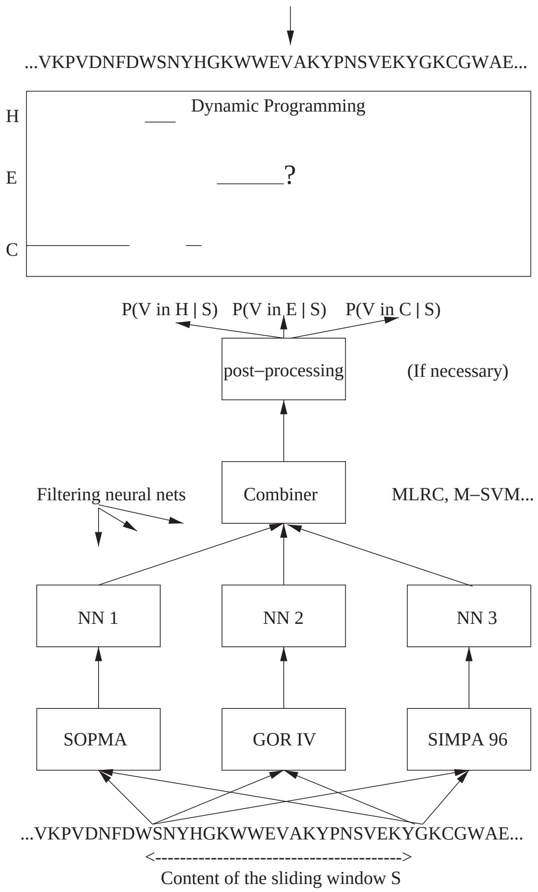

Fig. 1. Hierarchical architecture for protein secondary structure prediction.

Table 1
Initial relative prediction accuracy of individual experts (+f=after filtering) on the G629 set

	GOR IV	GOR IV+f	SOPMA	SOPMA+f	SIMPA	SIMPA+f
Q_3	64.1	66.5	68.4	69.7	69.2	69.4
C_α	0.47	0.51	0.55	0.58	0.56	0.57
C_β	0.39	0.43	0.48	0.49	0.49	0.49
C_c	0.44	0.46	0.49	0.50	0.49	0.49
$Sov'94$	0.66	0.68	0.72	0.71	0.71	0.70
σ_α	13.9	12.5	10.8	10.7	10.8	10.6
σ_β	11.5	11.6	10.3	11.1	11.2	10.7
σ_c	9.4	10.1	9.9	10.6	11.6	11.1

Table 2
Relative prediction accuracy of combiners on the G629 set

	Vote	Average	MLP	MLRC	$SVM_{\alpha+\beta+c}$	DAGSVM	M-SVM1	M-SVM2
Q_3	70.2	70.9	71.2	71.3	71.4	71.4	71.7	71.6
C_α	0.59	0.60	0.60	0.60	0.60	0.60	0.61	0.60
C_β	0.49	0.50	0.52	0.52	0.52	0.52	0.52	0.53
C_c	0.51	0.50	0.52	0.52	0.52	0.52	0.52	0.52
$Sov'94$	0.72	0.71	0.72	0.72	0.72	0.72	0.73	0.72
σ_α	10.5	10.0	10.1	10.3	10.6	10.7	10.6	10.6
σ_β	10.3	10.2	10.1	10.9	10.9	10.9	10.8	10.8
σ_c	10.1	10.3	10.5	11.4	11.3	11.3	11.2	11.1

5.2. Raw results of the combinations

Table 2 summarizes the relative performance of the different combiners. Figures given correspond to M-SVMs with radial basis kernels. Let d be the number of predictors (inputs). Here, $d=3(experts)\times 3(categories)=9$. We set $2\sigma^2=0.1d$ and $C=1.0$. This parameterization was selected since it appeared to be "satisfactory" for both models. However, additional testing performed with polynomial kernels suggests that the choice of the kernel could have significant incidence on the prediction accuracy (data not shown).

The comparison of the predictive success of native methods and combinations illustrates the usefulness of implementing ensemble methods. M-SVMs obtain the best results, the difference with MLRC being statistically significant with high confidence (>0.95).

5.3. Post-processing of the conformational scores

Promising as they may seem, these results are not sufficient to determine to what extent the conformational scores computed by the M-SVMs can be of interest for the biologist, for subsequent use as input to ab initio calculations or threading algorithms, or simply to provide a measure of reliability of the predictions [61,22,58]. In order to evaluate the quality of the combiners with respect to these criteria, their outputs were post-processed with a Dynamic Programming (DP) algorithm inspired by [56] (upper part of Fig. 1). Recent studies [47,44] have highlighted the fact that pattern recognition SVMs target directly at the Bayes rule without estimating the class posterior probabilities. As a consequence, the dot products they compute cannot be used in a straightforward manner to estimate observation probability distributions. In [52], Platt has proposed a solution to overcome this difficulty in the bi-class case. Here, we simply standardized the outputs by application of a softmax function. This could not be done for the DAGSVM, which was thus discarded. The underlying Inhomogeneous Hidden Markov Model (IHMM) is depicted in Fig. 2.

It has three states, one for each conformational state. The observations are the residues of the primary structure. The specificity of the algorithm lies in the state duration modeling. Instead of the standard stationary state transition probabilities, the

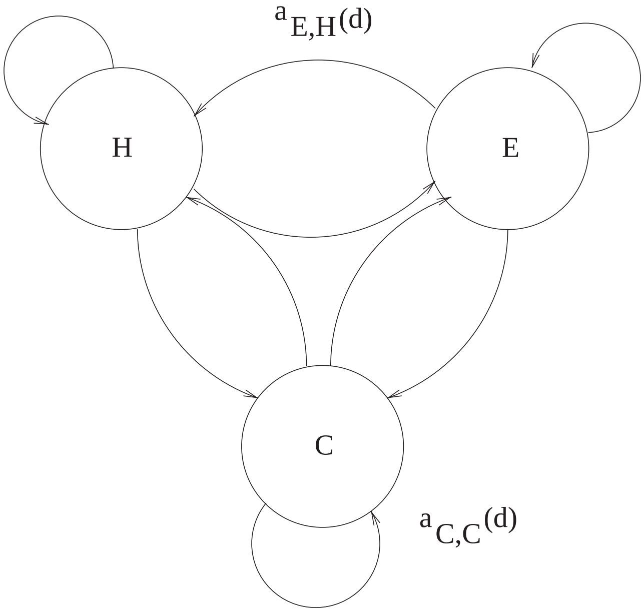

Fig. 2. Topology of the IHMM used to post-process the outputs of the combiners.

Table 3
Quality of the predictions when the outputs of the combiners have been post-processed by an inhomogeneous DP algorithm

	Average	MLP	MLRC	$SVM_{\alpha+\beta+c}$	M-SVM1	M-SVM2
Q_3	71.1	71.5	71.5	72.0	72.3	72.2
C_α	0.60	0.61	0.61	0.61	0.62	0.60
C_β	0.50	0.52	0.52	0.51	0.53	0.51
C_c	0.52	0.52	0.52	0.52	0.53	0.52
$Sov'94$	0.72	0.74	0.74	0.74	0.74	0.73
σ_α	10.6	11.8	10.8	10.9	10.7	10.7
σ_β	10.4	10.6	11.1	11.0	10.9	10.9
σ_c	10.6	10.8	11.6	11.8	11.8	11.7

terms $a_{ij}(d)$ are used, where parameter d represents the duration spent in the current (conformational) state i. These probabilities are estimated by the corresponding frequencies observed on the training set, whereas the observation pdf are derived from the outputs of the combiners, by means of Bayes' theorem.

As can be seen in Table 3, such a post-processing induces a global improvement of all the measures of prediction accuracy. Once more, the M-SVMs obtain the best results. The margin with respect to the standard combiners is larger, and statistically significant with confidence exceeding 0.99 for M-SVM1. This means that the level of the outputs of these machines carries some valuable information.

6. Combining BRNNs models from SSpro

SSpro is a protein secondary structure prediction method based on Bidirectional Recurrent Neural Networks (BRNNs) [5]. In its initial configuration, SSpro1, it was already one of the best methods published, with a recognition rate exceeding 76%. Recently, an improved version, SSpro2 [54], making use of new PSI-BLAST profiles [3,39], has become available online.[2] It achieved a sustained performance of about 78% correct prediction on different test sets. In order to assess the possibility for M-SVM combinations to improve consistently the test performance, even in the case of state-of-the-art experts, we implemented M-SVM1 to combine first the BRNNs incorporated in SSpro1, and then the BRNNs of SSpro2.

6.1. SSpro1

Experiments were performed with the 126 chains of soluble proteins used in [61] (RS126 set), with a total of 23,348 residues. This choice had the advantage to make our results readily comparable with those of most of the state-of-the-art prediction methods. It could be done since the sequences on which the 11 BRNNs had been trained have less than 25% identity with them. Secondary structure assignment was performed with the DSSP program, in the same way as for the G629 base (see Section 5.1 for details). The database was devided into four parts of equal size, in order to implement a simple 4-fold cross-validation procedure (no filtering was required here, since the outpouts of the BRNNs are already class posterior probability estimates). The recognition rates of the BRNNs on the 126 chains were ranging from 73.3% to 75.3%, with an average of 74.6%. Five types of combiners were assessed. A simple average with equal weights, corresponding to what was actually performed in SSpro1, a MLP, MLRC, M-SVM1 and a consensus prediction ($SVM_{\alpha+\beta+c}$) resulting from the one-against-all decomposition. Test performance is summarized in Table 4. The columns SVM_α, SVM_β and SVM_c contain the figures corresponding to the two-class SVMs devoted to the recognition of one single conformational state. The parameters of the training algorithms of the SVMs have been set as in Section 5.2. Here, $d = 11 \times 3 = 33$.

Q_α (resp. Q_β, Q_c) stands for the recognition rate when the problem of interest simply consists in determining whether or not the conformational state of a given residue is α-helix (resp. β-strand or random coil). The Sov measure implemented corresponds to the modified definition introduced in [75]. In contrast with the MLP, MLRC and the combination of two-class SVMs, the M-SVM succeeds once more in improving over the performance of the averaging. However, the gain is no longer high enough to be statistically significant. This obviously springs from the fact that its value is lower, but also from the fact that the database used is smaller. We took this latter aspect into consideration to derive the experimental protocol regarding SSpro2.

[2] http://www.igb.uci.edu/tools/scratch/

Table 4
Combination of the 11 BRNNs of SSpro1 with two-class and multi-class SVMs (performance is measured on the RS126 set)

	Average	MLP	MLRC	SVM_α	SVM_β	SVM_c	$SVM_{\alpha+\beta+c}$	M-SVM1
Q_3	76.62	76.25	76.65	–	—	—	76.23	76.73
Q_α	88.4	88.3	88.5	88.6	—	—	88.3	88.6
Q_β	86.3	85.8	86.3	—	86.2	—	85.9	86.4
Q_c	78.5	78.4	78.5	—	—	78.4	78.3	78.5
C_α	0.73	0.73	0.73	—	—	—	0.73	0.74
C_β	0.60	0.60	0.60	—	—	—	0.60	0.60
C_c	0.57	0.56	0.57	—	—	—	0.56	0.57
Sov	70.6	70.2	71.0	—	—	—	70.2	70.6
Sov_α	74.5	74.8	74.7	—	—	—	73.2	73.0
Sov_β	65.5	67.4	65.6	—	—	—	65.6	66.3
Sov_c	67.6	66.0	68.0	—	—	—	67.5	67.7

6.2. SSpro2

In order to assess the combination of the 11 BRNNs of SSpro2, we had first to generate a new database exhibiting no homology with the training set. To do so, we used the latest release of the PDB. We first excluded all NMR proteins, and then ordered the remaining sequences by decreasing quality (increasing value of the resolution). All structures with a resolution worse than three angstroms were discarded. We then run the standard all-against-all redundancy reduction with a 25% threshold for proteins of length greater than 80 amino acids, and larger for shorter chains (see [63] for details). The same redundancy reduction was also applied to discard the sequences exhibiting a too high similarity with sequences of the training set. At last, all sequences with non-standard amino acids were excluded. We ended up with a set of 1096 protein sequences (P1096), made up of 255,551 amino acids. Their PSI-BLAST profiles were compiled according to the protocol described in [54]. This way, SSpro2 could be run on them in optimal conditions, providing us with the outputs of the 11 BRNN models. Their prediction accuracy was ranging from 72.4% to 75.2%, with an average of 74.3%.

The combination of the outputs was performed according to two different protocols. The first one is the protocol used for SSpro1 (three categories, defined as in the CASP experiments, 33 predictors, each of them corresponding to one output of a BRNN, 4-fold cross-validation, same set of five ensemble methods, etc.). In the second set of experiments, new predictors were added. They represented the coding of the PSI-BLAST derived profiles for a window of size seven centered on the residue for which the prediction is made. Each location in the window is coded on 21 positions, one for each of the 20 amino acids plus one for non-standard cases. The total number of covariates was thus $d=33+7\times21=180$. The results of these two sets of experiments can be found respectively in Tables 5 and 6.

Two distinct conclusions can be derived from these statistics. Whereas no ensemble method performs significantly better that the simple averaging in the case when the sole BRNN outputs are combined, adding the profiles of alignment induces an increase

Table 5
Combination of the 11 BRNNs of SSpro2 with two-class and multi-class SVMs (performance is measured on the P1096 set)

	Average	MLP	MLRC	SVM_α	SVM_β	SVM_c	$SVM_{\alpha+\beta+c}$	M-SVM1
Q_3	76.94	76.91	77.11	—	—	—	77.01	77.09
Q_α	86.7	86.7	86.8	86.8	—	—	86.8	86.8
Q_β	87.7	87.6	87.7	—	87.8	—	87.3	87.8
Q_c	79.5	79.6	79.7	—	—	79.6	79.6	79.6
C_α	0.72	0.72	0.72	—	—	—	0.71	0.72
C_β	0.62	0.63	0.62	—	—	—	0.62	0.63
C_c	0.58	0.58	0.58	—	—	—	0.58	0.58
Sov	72.2	72.2	72.3	—	—	—	72.0	72.4
Sov_α	75.6	76.1	75.7	—	—	—	76.0	76.1
Sov_β	67.1	69.0	67.3	—	—	—	67.3	68.9
Sov_c	69.0	67.6	68.9	—	—	—	68.1	68.5

Table 6
Combination of the 11 BRNNs of SSpro2 and PSI-BLAST derived profiles with two-class and multi-class SVMs (performance is measured on the P1096 set)

	MLP	$SVM_{\alpha+\beta+c}$	M-SVM1
Q_3	77.02	77.06	77.26
Q_α	86.9	86.9	87.1
Q_β	87.6	87.3	87.8
Q_c	79.5	79.7	79.6
C_α	0.72	0.73	0.73
C_β	0.63	0.62	0.64
C_c	0.58	0.58	0.58
Sov	72.3	72.2	72.5
Sov_α	74.8	74.8	74.6
Sov_β	69.0	68.7	69.4
Sov_c	68.8	68.5	68.7

in the prediction accuracy. This increase is particularly noticeable for M-SVM1. The difference with the average (see the last column of Table 6 and the second column of Table 5) is significant with confidence exceeding 0.95. With the results at hand, it is difficult to characterize more precisely the nature of the improvement, which seems to affect primarily the β-sheets. We are currently performing additional testing, with larger sizes of the sliding window, so as to clarify this point.

7. Discussion

7.1. SVM versus standard ensemble methods

Given the number of methods currently available to predict the secondary structure of proteins, and the speed at which their performance improves, there is no denying that basic jury decisions will soon become inefficient to perform combinations in the field. In the past, significant advances have resulted from designing prediction methods based on simple MLPs [55,61,51]. However, quite surprisingly, these models do not appear appropriate to combine prediction systems. Indeed, the main problem faced when implementing them for this task is overfitting. This observation, which holds even for small sizes of the hidden layer, is confirmed by leading experts of the domain (B. Rost, personal communication). Although the generalization performance of MLRC is systematically superior to the generalization performance of a simple average, weighted or not, this combiner suffers from a drawback in the context of interest: it can only take class posterior probabilities as inputs. SVMs, on the contrary, do not suffer from overfitting, and can process virtually any kind of data. They should thus rise high expectations as ensemble methods. If the initial results reported here are only promising, many improvements can be considered. For instance, we are currently assessing the effect of presenting in input, in addition to the conformational scores provided by the experts and the content of a sliding window on PSI-BLAST profiles, additional physico-chemical data.

7.2. Bi-class versus multi-class SVMs

As for the choice between an architecture based on binary SVMs and a M-SVM, two strong arguments speak in favor of the latter. First, the (empirical) recognition rate of M-SVMs is higher. Second, M-SVMs use far fewer support vectors. In [37], the authors predicted the secondary structure directly from BLAST alignment profiles, using binary SVMs and different decomposition schemes, among which the one-against-all method and decision trees. According to their Table 6, this systematically resulted in a ratio of support vectors of about 50%. Such a ratio is hardly acceptable, if one keeps in mind that in Vapnik's theory, it corresponds to a very high bound on the expected risk (see for instance [67,69]). In our experiments, the M-SVMs had at most three times fewer support vectors (non-zero dual variables) than the decomposition schemes with which they were compared. This had the expected effect on the decoding time, which was far lower. Indeed, representer theorems establish that the output of a bi-class SVM is of the form: $h(x)=\sum_{i=1}^{N} \beta_i k(x_i,x)+b$ and that, similarly, any output $h_k(x)$ of any M-SVM is of the form: $h_k(x)=\sum_{i=1}^{N} \beta_{i,k} k(x_i,x)+b_k$ (at least, this is the case for all training criteria proposed so far). The lower the number of support vectors, the lower the number of terms in the sum, and, by way of consequence, the lower the time required to compute the outputs. Note that an additional advantage of the M-SVMs is that the value $k(x_i,x)$ needs to be computed only once for all the expansions $h_k(x)$, $(1 \leqslant k \leqslant Q)$. This more than compensates for the fact that the expression of $\beta_{i,k}$ (M-SVM) in terms of the dual variables is more complex than the expression of β_i (bi-class SVM).

To illustrate the gain in decoding time, the following experiment was implemented. M-SVM1 and the three bi-class SVMs, SVM_α, SVM_β and SVM_c (see Section 6.1), were trained directly on the primary sequences of the P1096 set. Precisely, the input was the coding of the content of a sliding window of size 13, centered on the residue to be assigned. This choice had the advantage to make it possible to compare the ratios of support vectors obtained with those reported by Hua and Sun. For the sake

Table 7
Time in minutes and percentage of support vectors required to process the P1096 set (training+test) with M-SVM1 and the one-against-all approach

	SVM_α	SVM_β	SVM_c	$SVM_{\alpha+\beta+c}$	M-SVM1
Training (mn)	567	532	605	1704	6030
Decoding (mn)	218	277	233	**728**	**157**
% SV	51.2	45.5	58.7	**51.8**	**16.9**

of simplicity, the training set was also used as test set. Although this procedure makes the recognition rates observed meaningless, this is utterly satisfactory when the concern is only on cpu time. For each machine, training was stopped when the ratio of the dual objective function on the estimate of the primal objective function exceeded 0.95. This could be done since the software (eval_SVM) actually computes an upper bound on the primal objective function (see the technical documentation for details). The interested reader will find alternative stopping criteria in [15]. The computer used is a DELL Precision 530 MT. It has two Xeon 2.8GHz/512k processors and 4GB of RDRAM memory. Note however that the released version of our programs, which should be parallelized in a near future, currently only use one processor at a time. A Gaussian kernel was used with $2\sigma^2 = 6d$ and the value of the soft margin parameter C was set to 10.0. Performance in terms of cpu time (training+test) as well as percentage of support vectors (SV) are reported in Table 7. The percentage of support vectors is simply the ratio of the number of positive dual variables over the total number of dual variables. This means for instance that for M-SVM1, one example can be considered twice in the numerator, if its dual variables associated with the two categories to which it does not belong are both positive.

8. Conclusion and future work

We have introduced a family of multi-class SVMs, the learning algorithms of which correspond to explicit implementations of the SRM inductive principle. This family includes the first M-SVM proposed in literature, thus endowing it with a theoretical grounding which was lacking so far. Two of these models have been used to combine protein secondary structure prediction methods. To the best of our knowledge, these combinations represent the first assessment of M-SVMs on a very large real-world problem. M-SVMs appear to give better performance than standard ensemble methods, or the implementation of decomposition schemes involving binary SVMs. Additional experiments are currently underway, to extend the comparison to the main multi-category SVMs which have been introduced lately [13,14,44,43]. Furthermore, experimental evidence suggests that the conformational scores produced could be processed by the biologist to perform higher-level treatments. This is all the more promising that no special effort was made to derive class posterior probabilities from them.

We are confident that noticeable benefits should be expected from generalizing the use of M-SVMs in the discriminant models performing tasks in biocomputing, such as gene expression measurements processing, translation initiation sites recognition, or the identification of ligand molecules binding by affinity on given target molecules. In order to meet the requirements of these implementations, we are currently investigating a development of central importance, the design of kernels taking into account the characteristics of the problem at hand [30]. This approach, which has already proved fruitful in bioinformatics [38,76,71,70,45], should lead us to specify the training procedure accordingly, i.e. to specify new machines, for which original uniform convergence results will have to be derived.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments, as well as Prof. G. Deléage, Dr. J. Garnier, Dr. C. Geourjon, Dr. J.-F. Gibrat, and Dr. J.-M. Levin for the availability of the software and predictions of the SOPMA, GOR IV, and SIMPA96 methods. Dr. C. Geourjon also provided us with the G629 data set. Thanks are also due to Dr. A. Zemla and Dr. C. Venclovas for the availability of the code of the Sov measure, and to D. Eveillard for carefully reading this manuscript. HPM, AE and YG gratefully acknowledge the support of the ESPRIT funded Working Group No. 27150 "Neural Networks and Computational Learning Theory". The work of PB and GP is supported by a Laurel Wilkening Faculty Innovation award and a Sun Microsystems award to PB at UCI.

References

[1] M. Aizerman, E. Braverman, L. Rozoner, Theoretical foundations of the potential function method in pattern recognition learning, Automat. Remote Control 25 (1964) 821–837.

[2] N. Alon, S. Ben-David, N. Cesa-Bianchi, D. Haussler, Scale-sensitive dimensions, uniform convergence, and learnability, J. ACM 44 (1997) 615–631.

[3] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D.J. Miller, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucl. Acids Res. 25 (17) (1997) 3389–3402.

[4] P. Baldi, S. Brunak, Bionformaics: The Machine Learning Approach, 2nd Edition, MIT Press, Cambridge, MA, 2001.

[5] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, G. Soda, Exploiting the past and the future in protein secondary structure prediction, Bioinformatics 15 (11) (1999) 937–946.

[6] P.L. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, IEEE Trans. Inf. Theory 44 (2) (1998) 525–536.

[7] P.L. Bartlett, J. Shawe-Taylor, Generalization performance of support Vector machines and other pattern classifiers, in: B. Schölkopf, C.J.C. Burges, A. Smola (Eds.), Advances in Kernel Methods, Support vector Learning, The MIT Press, Cambridge, 1999, pp. 43–54.

[8] V. Biou, J.-F. Gibrat, J.-M. Levin, B. Robson, J. Garnier, Secondary structure prediction: combination of three different methods, Prot. Eng. 2 (1988) 185–191.

[9] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.

[10] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: COLT'92, Pittsburgh, PA, 1992, pp. 144–152.

[11] E.J. Bredensteiner, K.P. Bennett, Multicategory classification by support vector machines, Comput. Optim. Appl. 12 (1/3) (1999) 53–79.

[12] C. Cortes, V.N. Vapnik, Support-vector networks, Mach. Learning 20 (1995) 273–297.

[13] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problem, in: Proceedings of the 13th Annual Conference on Computational Learning Theory (COLT), 2000, pp. 35–46.

[14] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learning Res. 2 (2001) 265–292.

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

[16] J.A. Cuff, G.J. Barton, Evaluation and improvement of multiple sequence methods for protein secondary structure prediction, Proteins 34 (1999) 508–519.

[17] A. Elisseeff, Etude de la complexité et contrôle de la capacité des systèmes d'apprentissage: SVM multi-classe, réseaux de régularisation et réseaux de neurones multicouches, Ph.D. Thesis, ENS Lyon, 2000 (in French).

[18] A. Elisseeff, Y. Guermeur, H. Paugam-Moisy, Margin error and generalization capabilities of multi-class discriminant models, Technical Report NC-TR-99-051-R, NeuroCOLT2, 1999 (revised in 2001).

[19] M. Frank, P. Wolfe, An algorithm for quadratic programming, Nav. Res. Logist, Quart. 3 (1956) 95–110.

[20] J. Friedman, Another approach to polychotomous classification, Technical Report, Department of Statistics, Stanford University, 1996.

[21] J. Garnier, J.-F. Gibrat, B. Robson, GOR method for predicting protein secondary structure from amino acid sequence, Methods Enzymol. 266 (1996) 540–553.

[22] C. Geourjon, G. Deléage, SOPM: a self-optimized method for protein secondary structure prediction, Protein Eng. 7 (2) (1994) 157–164.

[23] C. Geourjon, G. Deléage, SOPMA: significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments, CABIOS 11 (6) (1995) 681–684.

[24] Y. Guermeur, Combinaison de classifieurs statistiques, application à la prédiction de la structure secondaire des protéines, Ph.D. Thesis, Université Paris 6, 1997 (in French).

[25] Y. Guermeur, Combining discriminant models with new multi-class SVMs, Pattern Anal. Appl. 5 (2) (2002) 168–179.

[26] Y. Guermeur, A simple unifying theory of multi-class support vector machines, Technical Report RR-4669, INRIA, 2002.

[27] Y. Guermeur, A. Elisseeff, H. Paugam-Moisy, A new multi-class SVM based on a uniform convergence result, in: IJCNN'00, Vol. IV, Como, Italy, 2000, pp. 183–188.

[28] Y. Guermeur, A. Elisseeff, D. Zelus, Bounding the capacity measure of multi-class discriminant models, Technical Report NC-TR-2002-123-R, NeuroCOLT2, 2002 (revised).

[29] Y. Guermeur, C. Geourjon, P. Gallinari, G. Deléage, Improved performance in protein secondary structure prediction by inhomogeneous score combination, Bioinformatics 15 (5) (1999) 413–421.

[30] Y. Guermeur, A. Lifchitz, R. Vert, A kernel for protein secondary structure prediction, in: B. Schölkopf, K. Tsuda, J.-P. Vert (Eds.), Kernel Methods in Computational Biology, MIT Press, Cambridge, MA, 2003, to appear.

[31] Y. Guermeur, H. Paugam-Moisy, Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines, in: M. Sebban, G. Venturini (Eds.), Apprentissage Automatique, Hermès, Paris, France, 1999, pp. 109–138 (in French).

[32] Y. Guo, P.L. Bartlett, J. Shawe-Taylor, R.C. Williamson, Covering numbers for support vector machines, IEEE Trans. Inf. Theory 48 (1) (2002) 239–250.

[33] L. Gurvits, A note on a scale-sensitive dimension of linear bounded functionals in Banach spaces, Theoret. Comput. Sci. 261 (1) (2001) 81–90.

[34] D. Haussler, P.M. Long, A generalization of Sauer's lemma, J. Combin. Theory Ser. A 71 (1995) 219–240.

[35] U. Hobohm, C. Sander, Enlarged representative set of protein structures, Protein Sci. 3 (1994) 522–524.

[36] C.-W. Hsu, C.-J. Lin, A comparison of methods for multi-class support vector machines, IEEE Trans. Neural Networks 13 (2002) 415–425.

[37] S. Hua, Z. Sun, A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach, J. Mol. Biol. 308 (2001) 397–407.

[38] T. Jaakkola, M. Diekhans, D. Haussler, Using the Fisher kernel method to detect remote protein homologies. in: ISMB'99, Heidelberg, Germany, 1999, pp. 149–158.

[39] D.T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, J. Mol. Biol. 292 (1999) 195–202.

[40] W. Kabsch, C. Sander, Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers 22 (12) (1983) 2577–2637.

[41] M.J. Kearns, R.E. Schapire, Efficient distribution-free learning of probabilistic concepts, in: Proceedings of the 31st Annual Symposium on Foundations of Computer Science, Vol. 1, IEEE Computer Society Press, Silver Springer, MD, 1990, pp. 382–391.

[42] U. Kreßel, Pairwise classification and support vector machines, in: B. Schölkopf, C.J.C. Burges, A. Smola (Eds.), Advances in Kernel Methods, Support Vector Learning, The MIT Press, Cambridge, 1999.

[43] Y. Lee, Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, Technical Report 1063, Department of Statistics, University of Wisconsin, Madison, 2002.

[44] Y. Lee, Y. Lin, G. Wahba, Multicategory support vector machines, Technical Report 1043, Department of Statistics, University of Wisconsin, Madison, 2001.

[45] C. Leslie, E. Eskin J. Weston, W.S. Noble, Mismatch string kernels for SVM protein classification, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, Vol. 15, MIT Press, Cambridge, MA, 2003, pp. 1417–1424.

[46] J.-M. Levin, Exploring the limits of nearest neighbour secondary structure prediction, Protein Eng. 10 (7) (1997) 771–776.

[47] Y. Lin, Support vector machines and the Bayes rule in classification, Technical Report 1014, Department of Statistics, University of Wisconsin, Madison, 1999.

[48] B.W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochim. Biophys. Acta 405 (1975) 442–451.

[49] B.K. Natarajan, On learning sets and functions, Mach. Learning 4 (1989) 67–97.

[50] N.J. Nilsson, Learning Machines: Foundations of Trainable Pattern Classifying Systems, McGraw-Hill, New York, 1965.

[51] T.N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, S. Brunak, G.P. Gippert, O. Lund, Prediction of protein secondary structure at 80% accuracy, PROTEINS: Struct. Funct. Genet. 41 (1) (2000) 17–20.

[52] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 1999.

[53] J.C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin DAGs for multiclass classification, in: NIPS'12, Vancouver, B.C., Canada, 2000, pp. 547–553.

[54] G. Pollastri, D. Przybylski, B. Rost, P. Baldi, Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles, Proteins 47 (2) (2002) 228–235.

[55] N. Qian, T.J. Sejnowski, Predicting the secondary structure of globular proteins using neural network models, J. Mol. Biol. 202 (1988) 865–884.

[56] P. Ramesh, J.G. Wilpon, Modeling state durations in hidden Markov models for automatic speech recognition, in: ICASSP-92, Vol. I, San Francisco, CA, 1992, pp. 381–384.

[57] M.D. Richard, R.P. Lippmann, Neural network classifiers estimate bayesian a posteriori probabilities, Neural Comput. 3 (1991) 461–483.

[58] S. Riis, A. Krogh, Improving prediction of protein secondary structure using structured neural network and multiple sequence alignments, J. Comput. Biol. 3 (1996) 163–183.

[59] B. Rost, Review: protein secondary structure prediction continues to rise, J. Struct. Biol. 134 (2) (2001) 204–218.

[60] B. Rost, S. O'Donoghue, Sisyphus and prediction of protein structure, CABIOS 13 (4) (1997) 345–356.

[61] B. Rost, C. Sander, Prediction of protein secondary structure at better than 70% accuracy, J. Mol. Biol. 232 (1993) 584–599.
[62] B. Rost, C. Sander, R. Schneidder, Redefining the goals of protein secondary structure prediction, J. Mol. Biol. 235 (1994) 13–26.
[63] C. Sander, R. Schneider, Database of homology derived protein structures and the structural meaning of sequence alignment, Proteins 9 (1991) 56–68.
[64] B. Schölkopf, C. Burges, V. Vapnik, Extracting support data for a given task, in: KDD'95, Montreal, Canada, 1995, pp. 252–257.
[65] B. Schölkopf, J.C. Burges, A. Smola (Eds.), Advances in Kernel Methods, Support Vector Learning, The MIT Press, Cambridge, MA, 1999.
[66] B. Schölkopf, A.J. Smola, Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond, The MIT Press, Cambridge, MA, 2002.
[67] V. Vapnik, A. Chervonenkis, Theory of Pattern Recognition, Nauka, Moskow, 1974 (in Russian).
[68] V.N. Vapnik, Estimation of Dependences Based on Empirical Data, Springer, New York, 1982.
[69] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[70] J.-P. Vert, Extracting active metabolic pathways from gene expression data using kernel CCA, in: Statistical Learning, Theory and Applications, CNAM, Paris, France, 2002, pp. 89–92.
[71] J.-P. Vert, A tree kernel to analyze phylogenetic profiles, in: ISMB'02, Edmonton, Canada, 2002, pp. S276–S284.
[72] J. Weston, C. Watkins, Multi-class support vector machines, Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, 1998.
[73] R.C. Williamson, A.J. Smola, B. Schölkopf, Generalization performance of regularizations networks and support vector machines via entropy numbers of compact operators, IEEE Trans. Inf. Theory 47 (6) (2001) 2516–2532.
[74] D.H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.
[75] A. Zemla, Č. Venclovas, K. Fidelis, B. Rost, A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment, PROTEINS: Struct. Funct. Genet. 34 (2) (1999) 220–223.
[76] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, K.-R. Müller, Engineering support vector machine kernels that recognize translation initiation sites, Bioinformatics 16 (9) (2000) 799–807.

Yann Guermeur received a French "diplôme d'ingénieur" from the IIE in 1991. He then worked in the industry for 2 years. Four years later, he obtained a Ph.D. in computer science from the University Paris 6. He has worked at the ENS of Lyon and the University Paris 6. At present, he is assistant professor at the University Henri Poincaré–Nancy 1. He makes his research at the LORIA. His main research topics are statistical learning theory and its applications to Bioinformatics.

Gianluca Pollastri received his Master's Degree in Telecommunication Engineering from the University of Florence in 1999, with honours. For his thesis on a new family of adaptive connectionistic algorithms for sequences, he received a Best-MS-Thesis national award. He received his Ph.D. in Bioinformatics in 2003 from the University of California, Irvine, where he is working chiefly on protein structure prediction using machine learning algorithms. At UCI, he has designed a number of Internet servers, including one for protein secondary structure prediction currently ranked among the best in the world.

André Elisseeff studied Support Vector Machines during his thesis that he defended at the Ecole Normale Superieure de Lyon, France in 2000. He then joined a company named BIOwulf Technologies in New York, USA where he worked 2 years before coming back to Europe as a post-doc of the Max Planck Institut fur Biological Cybernetics of Tuebingen, Germany.

Dominique Zelus received a Ph.D. in Biology from the University Lille 2 in 1999. She made a post-doc at the Facultad de Ciencias Bioquimicas y Farmaceuticas of Rosario, in Argentina. She is currently Project Leader at Wiener Lab, CIBIO, in Rosario, where she is in charge of the ELISA system.

Hélène Paugam-Moisy obtained the French degree Agrégation de Mathématiques in 1987 and she received a Ph.D. in Computer Science in 1992 at University Lyon 1 and Ecole Normale Supérieure de Lyon. She is presently Professor at University Lyon 2. She manages a research team on Neural Networks and Cognitive Modelling at the CNRS Institute in Cognitive Science. Her research interest is on neural networks, learning theory, cognitive science and parallel computing.

Pierre Baldi is a Professor in the Department of Information and Computer Science and in the Department of Biological Chemistry at the University of California, Irvine where he is also the Director of the Institute for Genomics and Bioinformatics. He received a Ph.D. in Mathematics from the California Institute of Technology in 1986. Dr. Baldi is the author of over 100 scientific articles and three books. His research focuses on bioinformatics, machine learning, probabilistic modeling and statistical inference, artificial intelligence, and communication networks.