Combining Statistical Models for Protein Secondary Structure Prediction

Yann Guermeur, Patrick Gallinari

LAFORIA IBP-UA CNRS 1095
Université Pierre et Marie Curie
Tour 46-00, Boîte 169
4, Place Jussieu - 75252 Paris cedex 05 France
guermeur,gallinari@laforia.ibp.fr

Abstract: We investigate the problem of combining experts to predict the secondary structure of globular proteins. We first present two different statistical models for this task. We then analyse an efficient linear combination technique, this sheds light on unexplained phenomena frequently encountered in practice for ensemble methods.

1 Introduction

One current challenge of molecular biology is to determine the three-dimensional (3D) structure of proteins. Although it is fully determined by the sequence (primary structure), this problem is still unsolved. The usual way to proceed is to predict first an intermediate one-dimensional description called the secondary structure. This classification task consists in assigning a conformation, α-helix, β-strand or coil, to each residue of a sequence. This problem presents most of the difficulties found in the processing of sequences: long-term dependencies exploitation, noise processing and constraint satisfaction. This paper focuses on the combination of methods for secondary structure prediction. We first present in §2 an original recurrent neural network (NN) architecture. In §3, we describe a statistical model which exploits second order correlations present in the sequence. We then propose in §4 an analysis of linear combination methods which leads to a fast algorithm for combining experts.

2 Neural Architecture

Most of the neural systems applied so far to the prediction of the secondary structure are of the Qian and Sejnowski type [1]. They include two levels: a first network takes as input a segment of adjacent residues and outputs the secondary structure of the central residue in the segment. A second network smoothes these predictions: it takes as input a series of outputs from the first level to make the final decision.

2.1 A Sequence-to-Structure Network

Residues are usually encoded by binary vectors of 20 components, each of which corresponds to an amino acid. The input is then a "window" over the sequence including the central residue plus a left and right context of some neighbours (typically 3 to 9 to each side). The dimension of the input space is thus 20 times the window size. We use here for a more compact encoding a weighted sum of consecutive vectors. Let u_i denote the ith residue of the window encoded in a binary form, and $x \in \mathbb{R}^{20}$:

$$x = \sum_{i=1}^{|W|} w_i u_i \tag{1}$$

$|W|$ (the window size) residues are thus mapped into one vector x using only $|W|$ adjustable parameters w. The structure of our network is depicted in figure 1. The

input is a segment of 17 residues. A first layer computes weighted sums as defined in (1). It has been found empirically that using three distinct sums over the window was more efficient. Recurrent connections with fixed weights from the output to the input allow to implement an iterative scheme in order to consider a context using both past and "future" outputs for predicting y_9 (see notations in figure 1). Correlations between consecutive outputs are then taken into account. This conformational context is stored in a specific data structure and copied into the y' units. An iteration or epoch will consist in the presentation of the set of sequences to the network. For training and testing, the components of the context associated with each residue are initialized to the vector of *a priori* probabilities of the classes. When y_i has been computed, y_i' is replaced by its value, and the context is updated, it is then entirely modified at the end of the epoch.
Back-propagation has been used for training. Learning ends when the network produces a balanced prediction on the validation set, i.e. when the predicted proportions for the three classes on this set are equal to the *a priori* probabilities of the classes.

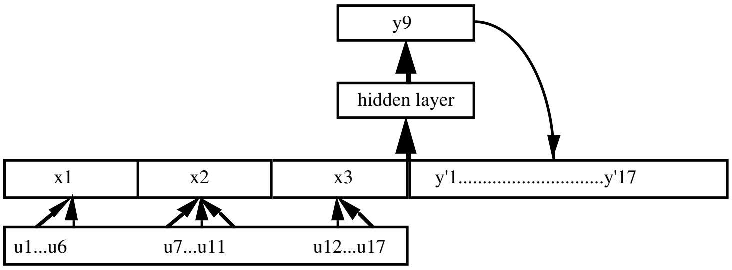

Fig. 1. Network architecture. Thick arrows designate full connections, thin arrows represent one-to-one connections. u denotes the residues, x_1, x_2, x_3 $(x_i \in \mathbb{R}^{20})$ *compute respectively a weighted sum for residues 1..6, 7..11, 12..17,* y' $(y_i' \in \mathbb{R}^3)$ *are copies of past outputs of the system corresponding to residues 1 to 17.* y_9 *is the prediction for* u_9.

2.2 A Structure-to-Structure Network

The outputs of the above system are then smoothed via a second classifier, a simple network with extra input cells encoding the charge and hydrophobicity of the residues as defined in [2]. Distinct sets of hidden units are devoted to the two types of inputs.

2.3 Experiments and Results

Database [1] was chosen since numerous methods have been assessed on it. Training and test sets are denoted respectively by setS and setT. A validation set was developed by selecting in another base [3] new sequences with less than 30% pairwise similarity with those of setT. 107 chains were thus gathered (setC). The assignment was done according to DSSP [4]. The following results describe the average behaviour over five runs of the architecture when trained with setS, setC being used for validation, and vice versa. For many biological reasons, the recognition rate cannot account alone for the quality of a prediction. The Matthews' correlation coefficients [5] denoted here C_α, C_β and C_{coil}, are also widely used since they give a measure of the prediction accuracy per structure. The performances compare favourably to those given by Sejnowski (prediction accuracy 64.3%, C_α=0.41, C_β=0.31, C_{Coil}=0.41).

Training set	recognition %	C_α	C_β	C_{coil}
setS	64.8	0.43	0.35	0.43
setC	66.0	0.46	0.37	0.43

Table1. Average test performances, for different training sets.

3 A Statistical Module

Simple NNs do not easily exploit higher order features present in the sequence (cf. [1]). The Bahadur expansion allows to do this explicitely [6]. A first attempt to use this technique for secondary structure prediction has been performed in [7], where the authors propose an extension of the original principle that only applies to dichotomies. We have derived a different extension in order to exploit co-occurences of residues in the various positions of the window.

Let x_{ij} be equal to 1 if residue i in the window is amino acid j and 0 otherwise. The content of the window is represented by the matrix $x=[x_{ij}]$, $(1 \le i \le |W|)$, $(1 \le j \le 20)$ ($|W|$ is the window size). The conditional probability of structure S_k given the context is $p(S_k|x) \propto p(x|S_k)p(S_k)$. The second order Bahadur expansion estimates $p(x|S_k)$:

$$p(x|S_k) \approx \hat{p}(x|S_k) = p_1(x|S_k)\left(1 + \sum_{20*i+j<20*l+m} r_{ijlmk} z_{ijk} z_{lmk}\right) \qquad (2)$$

with: $\alpha_{ijk} = E(x_{ij}|S_k)$, $z_{ijk} = (x_{ij} - \alpha_{ijk}) / \sqrt{\alpha_{ijk}(1-\alpha_{ijk})}$, $r_{ijlmk} = E(z_{ijk} z_{lmk})$,

$p_1(x|S_k) = \prod_i \prod_j \alpha_{ijk}^{x_{ij}} (1-\alpha_{ijk})^{1-x_{ij}}$.

z is the normalization of x with 0 mean and unit variance, r is a second order statistic, it measures the co-ocurrence of acids j and m respectively in positions i and l for structure S_k. p_1 denotes the joint distribution of the x_{ij} when they are supposed independant. $\hat{p}$ is thus a second order approximation of p in terms of pairwise correlations. An additional approximation, $\tilde{p}(x|S_k) = \sup(\hat{p}(x|S_k), 0)$, guarantees the non-negativity of the estimated values. Set setS + setC was used for training. The performances are significantly superior to those reported in [7].

Training set	recognition %	C_α	C_β	C_{Coil}
setS+setC	66.4	0.45	0.36	0.44

Table 2. Performances of the statistical model on the test set.

4 Ensemble Methods

4.1 Constrained Optimal Linear Combinations

Better classification performances may be obtained by using several classifiers and combining their outputs. The so called "ensemble methods" have been the subject of extensive studies in the field of NNs over these last years. Linear combinations, NNs, majority votes are among the techniques which have been used for this. We have found the former to be superior, when using adequate constraints. We propose below an analysis which sheds light on a frequently observed behaviour of this technique. It allows to implement an efficient algorithm for finding optimal combinations.

Our analysis is asymptotical. We make the assumption that our experts are unbiased. Let f_j $(1 \le j \le P)$ denote the classification function computed by expert j

$(f_j(x_i) \in \mathbb{R}^3)$ and $Q(v) = \frac{1}{2N} \sum_{i=1}^{N} \left\| y_i - \sum_{j=1}^{P} v_j f_j(x_i) \right\|^2$ the mean squared error (MSE) of the combination.

The linear combination of the experts will be optimal for MSE criterion and unbiased if it is the solution to the following constrained problem:

$$\begin{cases} \textit{Minimize } Q(v) \\ \text{subject to constraint } 1_P^T v = 1 \end{cases} \qquad \text{(Pb. 1)}$$

where 1_P is a column vector of P ones.

In this case, the solution is:

$$v^* = (1_P^T C^{-1} 1_P)^{-1} C^{-1} 1_P, \qquad Q(v^*) = \tfrac{1}{2}(1_P^T C^{-1} 1_P)^{-1},$$

where C is the estimated covariance matrix of the errors of the experts.

It has been frequently observed experimentally, e.g. [8] that using an additional constraint of non-negativity on the $v_i s$ led to significantly lower prediction errors. Unfortunately, the optimal convex combination can no more be expressed analytically, since it must be computed by constrained iterative methods, such as Uzawa's algorithm. *We suggest here a simple technique for computing a solution, and show its optimality in a simplified case*. Our algorithm is as follows:

Iterate
Compute the optimal solution for (Pb. 1),
remove the experts with a negative coefficient,
Until the solution is convex.

Although our method is not optimal in the general case, we observed that it produces in practice the right solution. We propose below an analysis of this phenomenon. For this purpose, an extra hypothesis is made:

The Off- diagonal correlation coefficients are all equal to a constant $\rho > 0$ (Hyp. 1)

This simplifying hypothesis is coherent with the observed behaviour of the experts (see also [7]), since their errors are almost uniformly correlated. The result below can then be shown:

Theorem: under hypothesis (Hyp. 1), the algorithm converges to the solution of the following problem $\begin{cases} \textit{Minimize } Q(v) \\ \text{subject to constraints } 1_P^T v = 1 \text{ and } v_i \geq 0 \ (1 \leq i \leq P) \end{cases}$ (Pb. 2)

Proof: see appendix.

4.2 Experiments and Discussion

Linear combinations (LC) have been compared to a NN which takes as input the predictions made for 17 consecutive residues. The experts are four NNs (trained respectively on setS and setC, by presenting the sequences from the N-terminus to the C-terminus and conversely), and the statistical module of §3. The training procedure of the combiners is derived from stacked generalization [9]. Learning sets are split into four disjoint parts of roughly equal size. One is removed in each set and experts are trained. Their outputs on the unseen training data are stored. This procedure is iterated four times, and afterwards the resulting predictions are used to train the combiners. Finally, the experts are trained again on their entire learning sets. Table 3 gives the comparative performances of the ensemble methods.

Combiner	recognition %	C_α	C_β	C_{Coil}
NN	66.8	0.48	0.38	0.45
optimal LC	67.3	0.49	0.37	0.46
convex LC	67.8	0.50	0.37	0.46

Table 3. Relative average performances of ensemble methods.

The variations in the behaviours of the experts between the two training phases explain the poor performances of the NN. Its larger parameterization is useless since the task it learns is significantly biased. Using complex combiners within the framework of stacked generalization is only appropriate when one strictly follows the computer demanding original procedure, or assumes a stable behaviour of the experts.

5 Conclusion

This study highlights some properties of linear and non-linear cooperation techniques applied to a difficult real-world problem, and illustrates their impact on the performances. It offers new insights into the choice of an ensemble method. A natural extension of this work is its transposition for the processing of multiple alignments. The best system developed so far for protein secondary structure prediction [10] takes as input the profile of amino acid substitutions, in place of simple sequences. We are currently working on the problem of weighting the predictions made independently for aligned sequences.

References

[1] Qian, N. and Sejnowski, T.J. (1988). Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *J. Mol. Biol.*, **202**, 865-884.

[2] Eisenberg, D., Wilcox, W. and Eshita, S. (1987). Hydrophobic moments as tools for analysis of protein sequences and structures. In *Proteins: structure and function*. Edited by James J. L'Italien, Plenum Press, 1987, 425-436.

[3] Colloc'h, N., Etchebest, C., Thoreau, E., Henrissat, B. and Mornon, J.P. (1993). Comparison of three algorithms for the assignment of secondary structure in proteins: the advantages of a consensus assignment. *Protein Engineering*, **vol. 6**, 377-382.

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

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

[6] Bahadur, R.R. (1961). A Representation of the Joint Distribution of Responses to *n* Dichotomous Items. In *Studies in Item Analysis and Prediction,* chapt. 9, 158-169, Stanford University Press.

[7] Zhang, X., Mesirov, J.P. and Waltz, D.L. (1992). Hybrid System for Protein Secondary Structure Prediction. *J. Mol. Biol.*, **225**, 1049-1063.

[8] Breiman, L. (1992). Stacked Regressions. Technical Report N° 367, August 1992, Department of statistics, University of California, Berkeley.

[9] Wolpert, D.H. (1992). Stacked Generalization. *Neural Networks*, **vol. 5**, 241-259.

[10] Rost, B. and Sander, C. (1994). Combining Evolutionary Information and Neural Networks to Predict Protein Secondary Structure. *Proteins*, **19**, 55-72.

Appendix: Proof of the theorem in section 4.1

Proving the theorem is equivalent to showing that each negative coefficient in the solution of (Pb. 1) corresponds to a null coefficient in the solution of (Pb. 2).

The computation of $v_{i,P}^*$, component of the solution of problem (Pb. 1), gives:

$$v_{i,P}^* = K_P \prod_{j=1,j\neq i}^{j=P} \sigma_j \left\{ \left(1+(P-2)\rho\right) \prod_{k=1,k\neq i}^{k=P} \sigma_k - \rho \left(\sum_{l=1,l\neq i}^{l=P} \prod_{m=1,m\neq l}^{m=P} \sigma_m \right) \right\} \quad (3)$$

where σ_i is the standard deviation of the error of expert *i*, and K_P is a positive constant. Let v_P^{**} be the solution of (Pb. 2). We assume that experts are sorted with increasing σ_i. The demonstration goes in three steps:

1) If $v_P^* \notin \mathbb{R}_+^P$, $\exists T \in \{2..P\} / v_{i,P}^* \geq 0 \ (1 \leq i \leq T-1)$, $v_{j,P}^* < 0 \ (T \leq j \leq P)$

This result is established by noticing that the sign of $v_{i,P}^*$ is the sign of the expression between brackets in (3), which decreases for increasing indexes.

2) The strict convexity of the cost function implies that $v_{P,P}^{**} = 0$

• Let $\theta = \min\limits_{i\in\{T..P\}} v_{i,P}^{**} / (v_{i,P}^{**} - v_{i,P}^*)$. If $\forall i \in \{T..P\}, v_{i,P}^{**} > 0$, then $\theta \in]0;1[$, and $w_P = \theta v_P^* + (1-\theta) v_P^{**}$ satisfies the constraints of (Pb. 2). Using the strict convexity of Q, we get $Q(w_P) < \theta Q(v_P^*) + (1-\theta) Q(v_P^{**}) < Q(v_P^{**})$. This is in contradiction with the definition of v_P^{**}, so $\exists i \in \{T..P\} / v_{i,P}^{**} = 0$. If *T*=*P*, the result is established.

• $2(Q(v_P^{**}) - Q(v_P^*)) = (C(v_P^{**} - v_P^*), v_P^{**}) = (C(v_P^{**} - v_P^*), v_P^{**} - v_P^*)$, i.e. v_P^{**} is the projection of v_P^* on the convex *U* defined by the constraints of (Pb. 2). According to the projection theorem, $\forall v_P \in U, (C(v_P^{**} - v_P^*), v_P - v_P^{**}) \geq 0$, whence can be derived $\forall v_P \in U, (C v_P^{**}, v_P - v_P^{**}) \geq 0$.

If we assume $v_{P,P}^{**} > 0$, let $Z \in \{T..P-1\} / v_{Z,P}^{**} = 0$, and let $w_P \in U / w_{i,P} = v_{i,P}^{**}$, $(1 \leq i \leq P-1, i \neq Z)$, $w_{Z,P} = v_{P,P}^{**}$, $w_{P,P} = 0$.

$$(C v_P^{**}, w_P - v_P^{**}) = \rho(\sigma_Z - \sigma_P) v_{P,P}^{**} \sum_{i\neq Z, i\neq P} v_{i,P}^{**} \sigma_i + (\rho\sigma_Z - \sigma_P) v_{P,P}^{**2} \sigma_P < 0.$$

This proves by contradiction that $v_{P,P}^{**} = 0$.

3) From (3) it can be derived: $\forall i \in \{T..P-1\}$, $v_{i,P-1}^* < 0$

$\left(1+(P-2)\rho\right) \prod\limits_{k\neq i} \sigma_k - \rho \sum\limits_{l\neq i} \prod\limits_{m\neq l} \sigma_m < 0$ and $\rho\left(\prod\limits_{m\neq P} \sigma_m - \prod\limits_{k\neq i} \sigma_k\right) < 0$.

Thus, $\sigma_P \left\{ (1+(P-3)\rho) \prod\limits_{k\neq i, k\neq P} \sigma_k - \rho \sum\limits_{l\neq i, l\neq P} \prod\limits_{m\neq l, m\neq P} \sigma_m \right\} < 0$ and $v_{i,P-1}^* < 0$.

(This implies that the same demonstration applies again to prove the nullity of all coefficients $v_{i,P}^{**}$ $(T \leq i \leq P-1)$.)