pattern recognition, combining classifiers,
protein secondary structure prediction

Tomasz WOŁOSZYŃSKI, Marek KURZYŃSKI

ON A NEW METHOD OF COMBINING CLASSIFIERS APPLIED TO THE PROTEIN SECONDARY STRUCTURE PREDICTION

We introduce common framework for fusion methods using dynamic weights in decision making process. Both weighted average combiners with dynamic weights and combiners which dynamically estimate local competence are considered. Few algorithms presented in the literature are shown in accordance with our model. In addition we propose two new methods for combining classifiers. The problem of protein secondary structure prediction was selected as a benchmark test. Experiments were carried out on previously prepared dataset of non-homologous proteins for fusion algorithms comparison. The results have proved that developed framework generalises dynamic weighting approaches and should be further investigated.

1. INTRODUCTION

Information fusion has been investigated with much attention in recent years. The idea of using ensemble of classifiers instead of single one proved to be useful, assuring higher classification accuracies in many pattern recognition problems. In general, combining methods may be divided into two groups: classifier fusion and classifier selection. The first one assumes that the final decision should be made using all classifiers outputs. The latter chooses single classifier with the highest local competence and relies only on its supports. In section 2 we present common framework for both weighted average combiners with dynamic weights (WAD) and combiners which estimate local competence dynamically (LCE) [11]. Described approaches make sense in problems where similarity between objects can be measured. Although continuous character of input features seems to be a good criterion for selecting a specific task it may be very interesting to examine performance of introduced fusion methods elsewhere. The protein secondary structure prediction, being one of the most important challenges in computational biology provides us with such testing data. Two main differences between classical pattern recognition problem and predicting three-dimensional conformation of a protein making the task more demanding are: variable length of input object and computation of distance between two proteins using evolutionary matrix. In section 3 we describe the protein dataset and discuss the results of benchmark tests performed on proposed combining algorithms and few other fusion methods for comparison. Conclusions for presented combiners are given afterwards.

2. THEORETICAL STUDY

2.1. WAD AND LCE COMBINERS

We are given the ensemble of *N* base classifiers, each of them producing a row vector with supports for *M* classes. All of the support vectors form a decision profile matrix *DP(x)* [11] for any input object *x*:

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \cdots & d_{1,M}(x) \\ \vdots & & \vdots \\ d_{N,1}(x) & \cdots & d_{N,M}(x) \end{bmatrix}, \tag{1}$$

where $d_{n,m}(x)$ denotes support of *n*-th classifier for *m*-th class for object *x*. Without loss of generality we can restrict $d_{n,m}(x)$ within the interval [0,1] and additionally $\sum_m d_{n,m}(x) = 1$. We assume that weights $w_{n,m}(x)$ ($n = 1,2,...,N,\ m = 1,2,...,M$) used both by WAD and LCE combiners in fusion procedure depend on the input object *x* and form a matrix $W(x)$. For a WAD combiner final support for class *m* is given by weighted sum of supports of base classifiers, viz.

$$\mu_m(x) = \sum_{n=1}^{N} w_{n,m}(x)\, d_{n,m}(x), \tag{2}$$

whereas in the case of LCE this support is equal to the support of base classifier with the greatest local (at point *x*) competence. As a competence measure we adopt the sum of classifier weights, which leads to the following final support of LCE combiner:

$$\mu_m(x) = d_{n,m}(x), \quad \text{where} \quad \sum_{j=1}^{M} w_{n,j}(x) = \max_k \sum_{j=1}^{M} w_{k,j}(x).. \tag{3}$$

The class which gets the highest final support is assigned to the input object.

2.2. FRAMEWORK FOR COMBINERS USING DYNAMIC WEIGHTS

Let us assume that the feature space is divided into *K* disjoint regions R_k. Suppose that E^* and E_k^* are fusion methods with best possible static matrix weights for whole feature space and for region *k* respectively. The following inequality holds:

$$\bigvee_{k=1,\ldots,K} P_c(E_k^* | R_k) \geq P_c(E^* | R_k), \tag{4}$$

where $P_c(E|R_k)$ denotes probability of correct classification for ensemble *E* under condition that object *x* lies in region R_k. It is clear that the feature space division provides us with better classification accuracy:

$$P_c(\,fusion) = \sum_k P_c(E_k^* | R_k) P(R_k) \geq P_c(E^*)\,. \tag{5}$$

If we split feature space into infinite number of regions so that each of them shrinks to a single point we get:

$$P_c(\,fusion) = \int_X P_c(E_x^* | x)\, f(x)\mathrm{dx}\,, \tag{6}$$

where the term under integral takes value from the set {0,1}. Therefore in order to maximize the probability (6) it is sufficient to maximize just this term for any given *x*:

$$\max_E P_c(\,fusion) \equiv \max_E P_c(E_x^* | x)\,. \tag{7}$$

This approach may be used only with objects x_l for which class memberships i_l are known. We denote such learning set by $S_L = \{x_l, i_l\}$ and its cardinality by *L*. For any other object *x* we suggest finding the weights matrix $W(x)$ by following equation:

$$W(x) = \sum_{l=1}^{L} g(x, x_l) W(x_l)\,, \tag{8}$$

where $g(x, x_l)$ is a function dependent on the distance $d(x, x_l)$ between objects *x* and x_l.

	Weights matrix W(x) $r = i_l$	**Distance dependent function** $g(x, x_l)$
CC1 (Distance-based *k*-nn)	$w_{n,r}(x_l) = d_{n,r}(x_l)$	$g(x, x_l) = \frac{1}{d(x, x_l)}$
CC2 (Potential functions)	$w_{n,r}(x_l) = \begin{cases} 1 & if \quad d_{n,r}(x_l) = \max_j d_{n,j}(x_l) \\ -1 & otherwise \end{cases}$	$g(x, x_l) = \frac{1}{1 + (d(x, x_l))^2}$
CC3	$w_{n,r}(x_l) = \begin{cases} d_{n,r}(x_l) & if \quad d_{n,r}(x_l) = \max_j d_{n,j}(x_l) \\ -\max_j d_{n,j}(x_l) & otherwise \end{cases}$	$g_1(x, x_l) = \frac{1}{d(x, x_l)}$
CC4	$w_{n,r}(x_l) = \begin{cases} d_{n,r}(x_l) & if \quad d_{n,r}(x_l) = \max_j d_{n,j}(x_l) \\ -\max_j d_{n,j}(x_l) & otherwise \end{cases}$	$g_2(x, x_l) = \frac{1}{1 + (d(x, x_l))^2}$
CC5	$w_{n,m}(x_l) = \begin{cases} d_{n,m}(x_l) & if \quad d_{n,r}(x_l) = \max_j d_{n,j}(x_l) \\ 0 & otherwise \end{cases}$	$g_1(x, x_l) = \frac{1}{d(x, x_l)}$
CC6	$w_{n,m}(x_l) = \begin{cases} d_{n,m}(x_l) & if \quad d_{n,r}(x_l) = \max_j d_{n,j}(x_l) \\ 0 & otherwise \end{cases}$	$g_2(x, x_l) = \frac{1}{1 + (d(x, x_l))^2}$

Table 1. Tested algorithms presented in accordance with proposed framework

The way of creating matrices $W(x_l)$ as well as defining function $g(x, x_l)$ are parameters of introduced algorithm. We have adapted two LCE methods proposed in the literature: distance-based *k-nn* [4] and potential functions [11] as well as two new ones using presented model. All of them were tested during experiments and are shown in Table 1.

3. APPLICATION TO PROTEIN SECONDARY STRUCTURE PREDICTION

3.1. INTRODUCTION TO PROTEIN PREDICTION

The problem of secondary structure prediction for a given protein is of great importance in the field of drug designing. Current measuring methods providing three-dimensional protein structures i.e. X-ray crystallography using diffraction images or NMR are based on expensive and long processes, therefore computational techniques are used to overcome these disadvantages. In general the prediction approach is very similar to classical pattern recognition model. We are given the input object *x* being the sequence of letters, so called primary structure. Each letter encodes one amino acid and takes one of 20 different values. The sequence length (number of residues in the chain) depends on the protein. The classification problem is defined as follows: for each amino acid in the given sequence predict their conformational state which can be either α-helix (encoded by letter H), β-strand (E) or other (C).

3.2. EXPERIMENTS

We have derived non-homologous protein dataset from PDBSELECT [7] with 25% similarity threshold. The total number of 583 proteins with 49322 residues were selected for the experiment. Only proteins with at most 150 amino acids in the sequence were taken into account. PDBFINDER2 [9] was used for finding DSSP [8] predictions (class memberships). We have reduced number of classes on DSSP output to previously described three (H, E, C). The ensemble of base classifiers is built of 3 different methods: GORIV [2] (based on information theory), HNN [5] (hierarchical neural network) and SOPMA [3] (based on multiple alignments).

	Q_H	Q_E	Q_C	Q_3	SOV_H	SOV_E	SOV_C	SOV_3
GORIV	59.26	60.78	65.83	60.43	67.26	67.37	71.62	72.02
HNN	68.09	56.45	74.48	66.65	73.85	60.66	79.23	75.77
SOPMA	72.85	64.35	68.04	68.73	78.23	71.12	76.27	79.07
MAX	65.34	59.87	72.32	65.53	72.15	65.73	78.50	76.10
MEAN	65.53	59.78	74.39	66.65	72.54	65.71	78.86	76.72
VOTE	68.65	60.41	75.15	68.60	75.06	65.84	80.70	78.36
ORACLE	73.06	68.10	77.95	82.30	86.63	80.75	91.24	89.40

Table 2. Prediction accuracies for base classifiers and selected combiners (in %)

The results were gathered using NPS@ server [1]. Each of classifiers gives three degrees of support for every amino acid in the chain. The outputs were processed using modified softmax method in order to estimate posterior probabilities. The distance between a pair of amino acids from different proteins was computed using BLOSUM30 [6] scoring matrix with window size of 11. Ten most similar amino acids were taken into account during computation process. The classical accuracy rate given by the quotient of properly classified amino acids to their total number is denoted by Q_3 for the whole class set or by Q_H, Q_E and Q_C for α-helix, β-strand and other respectively. Although this kind of measure is very common in pattern recognition problems it may be misleading when dealing with protein secondary structure prediction. Segment overlap rate SOV [12] was developed specially for this task and is much more competent.

	Q_H	Q_E	Q_C	Q_3	SOV_H	SOV_E	SOV_C	SOV_3
CC1 (Distance-based *k*-nn)	65.84	58.12	74.60	66.06	73.33	64.96	80.64	76.44
CC2 (Potential functions)	68.18	60.55	74.31	67.79	76.31	68.18	82.09	78.45
CC3	67.84	60.30	74.50	67.67	76.09	67.80	82.01	78.16
CC4	67.83	60.29	74.50	67.67	76.07	67.82	82.02	78.15
CC5	65.17	56.17	78.10	66.81	73.40	64.99	82.39	76.82
CC6	65.18	56.19	78.10	66.80	73.37	65.04	82.39	76.79

Table 3. Prediction accuracies for proposed classifiers and selected combiners (in %)

Prediction accuracies for base classifiers and simple combining methods such as max, mean and majority voting [10] for the whole dataset are presented in Table 2. First of all it should be stated that all base classifiers give quite distinct predictions. Still SOPMA method seems to be the best one among others. Combiners such as max and mean are always less accurate than the best single classifier but in overall they are superior to both GORIV and HNN. Majority voting combiner gets the highest score for predicting class C and is almost as good as SOPMA algorithm for total conformational state prediction. The results given for oracle classifier are very interesting and meaningful. It is on average at least 10 percentage points better than any of other methods. This proves that there is much space for improvement for combining algorithms. Testing methods described in section 2 was carried out with ten-fold cross validation. Results are shown in Table 3. The CC1 fusion approach is inferior to all other combiners. This fact may be caused by the way of computing weights matrix where no penalty policy was applied. Similar situation can be seen for CC5 and CC6 algorithms despite the best accuracies for class C. The latter two were tested using WAD final supports (1). The best three combiners are CC2, CC3 and CC4. Each of them is better than majority voting method in all classes, but surprisingly the overall scores are lower. Nonetheless they assure good performance and all were examined using LCE approach (2), which is worth mentioning.

4. CONLUSIONS

We have introduced a framework for combining classifiers based on dynamic weights. Two parameters in our model: distance dependent function and weights matrix allow us to modify the fusion process in many ways. The generalisation ability of developed algorithm was proven by adapting existing LCE combiners in accordance to our approach. Future investigation should be focused on selecting the most proper parameters for a given problem. Improvement of method proposed for computing the weight matrix for particular input object *x* would also be desirable. The accuracies gained during experiments on protein secondary structure prediction are satisfactory in comparison to other types of combiners. However it should be stated that the process of adapting protein dataset to the pattern recognition model could be done in different manners providing even better performance of introduced fusion methods.

BIBLIOGRAPHY

[1] COMBET C., BLANCHET C., GEOURJON C., DELEAGE G., NPS@: Network Protein Sequence Analysis, TIBS Vol. 25, No. 3, pp.147-150, 2000.
[2] GARNIER J., GIBRAT J-F., ROBSON B., GOR secondary structure prediction method version IV, Methods in Enzymology R.F. Doolittle Ed., Vol. 266, pp.540-553, 1996.
[3] GEOURJON C., DELEAGE G., SOPMA: Significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments, Comput Appl Biosci, pp.681-684, 1995.
[4] GIACINTO G., ROLI F., Design of effective neural network ensembles for image classification processes, Image Vision and Computing Journal, pp.699-707, 2001.
[5] GUERMEUR Y., Combinaison de classifieurs statistiques, Application a la prediction de structure secondaire des proteines, PhD Thesis.
[6] HENIKOFF S., HENIKOFF JG., Amino acid substitution matrices from protein blocks, PNAS USA, pp.10915-10919, 1992.
[7] HOBOHM U., SANDER C., Enlarged representative set of protein structures, Protein Science pp.522, 1994.
[8] KABSCH W., SANDER C., Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers, pp.2577-2637, 1983.
[9] KRIEGER E., HOOFT R., NABUURS S., VRIEND G., PDBFinderII - a database for protein structure analysis and prediction, 2004.
[10] KUNCHEVA L., A theoretical study on six classifier fusion strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 2, pp.281-286, 2002.
[11] KUNCHEVA L., Combining pattern classifiers: methods and algorithms, John Wiley & Sons, New Jersey, 2004.
[12] ZEMLA A., VENCLOVAS C., FIDELIS K., ROST B., PROTEINS: Structure, Function, and Genetics, pp.220-223, 1999.