DOI 10.1007/s00500-014-1355-0

METHODOLOGIES AND APPLICATION

Effect of simple ensemble methods on protein secondary structure prediction

Hafida Bouziane · Belhadri Messabih · Abdallah Chouarfia

Published online: 19 August 2014

Abstract Ensemble methods for building improved classifier models have been an important topic in machine learning, pattern recognition and data mining areas, where they have shown great promise. They boast a robustness that has spearheaded their application in many practical classification problems, especially when there is a significant diversity among the ensemble members. Actually, they replace traditional machine learning techniques in many applications and special attention has been devoted to them as a mean to improve the prediction accuracy for problems of high complexity. Several combination rules have been investigated in this context. However, it is claimed that no rule is always better than others for designing an optimal decision. The present study evaluates the performance of two different ensemble methods for protein secondary structure prediction. We focus on weighted opinions pooling and the most common aggregation rules for decisions inference. The ensemble members are accurate protein secondary structure single model predictors namely, Multi-Class Support Vector Machines and Artificial Neural Networks. Experiments are carried out using cross-validation tests on RS126 and CB513 benchmark datasets. Our results clearly confirm that ensembles are more accurate than a single model and the experimental comparison of the investigated ensemble schemes demonstrates that the newly introduced rule called Exponential Opinion Pool competes well against state-of-the-art fixed rules, especially the sum rule which in some cases is able to achieve better performance.

Keywords Ensemble methods · Simple aggregation rules · Weighted opinions pooling · Protein secondary structure prediction

Abbreviations

ANN	Artificial Neural Networks
BLAST	Basic Local Alignment Search Tool
BLOSUM	BLOck SUbstitution Matrix
ExpOP	Exponential Opinion Pool
FNN	Feed-Forward Neural Network
IFS	Ideal fold selection
LinOP	Linear Opinion Pool
LogOP	Logarithm Opinion Pool
MLP	Multi-Layer Perceptron
M-SVM	Multi-Class Support Vector Machines
MV	Majority vote
PSI-BLAST	Position-Specific Iterative BLAST
PSSP	Protein secondary structure prediction
RBFNN	Radial Basis Function Neural Network
SVM	Support Vector Machines
WMax	Weighted Max
WMin	Weighted Min

1 Introduction

In recent years, there has been a growing interest in combining multiple decisions in machine learning, pattern recognition, and data mining areas. Known under a variety of names

Communicated by V. Loia.

H. Bouziane (✉) · B. Messabih · A. Chouarfia
Department of Computer Science, USTO-MB University,
BP 1505, El M'naouer, Oran, Algeria
e-mail: hafida_chouarfia@yahoo.fr; h_bouziane@univ-usto.dz

B. Messabih
e-mail: belhadri.messabih@univ-usto.dz

A. Chouarfia
e-mail: chouarfia@univ-usto.dz

in the literature such as ensemble of classifiers, expert combination, ensemble committee, classifier fusion, mixture of experts, and more (Kuncheva and Whitaker 2003). The combined classifiers are commonly referred to as base classifiers. This relatively recent learning paradigm where many classifiers are jointly used to solve a problem is still an important topic (Dietterich 2000). The main idea behind is that the generalization ability of an ensemble is often significantly better than each of its members separately (Dietterich 2000). Of course, the chosen members must be cooperative or in other words, complementary in final decision making. If they always agree, the gain in performance of the ensemble is negligible, if it is not null. Whereas if they disagree, errors made by one or some members can be corrected by the others. Although, this assumption appears unrealistic, the ultimate goal remains to improve the confidence of making right decision even if the improvement is minor. Therefore, for designing a typical ensemble, two issues are pertaining. First, selection of efficient ensemble members. Whereas the second issue consists in finding an appropriate rule that combines their outputs. For this purpose, many research studies concentrate on classifier ensembles. Up until now, there is no standard procedure to design an effective ensemble. However, it has been clearly established that the gain obtained by building an ensemble is mainly affected by the chosen ensemble members. The number that should be used for a specific application and the requirements that they should fulfill remain the two key issues to consider. It is well-known fact that the success of an ensemble relies primarily on the diversity of the individual models combined, which reinforces the assumption of uncorrelated errors (Kuncheva and Whitaker 2003; Kuncheva 2005). Many diversity/ambiguity measures have been developed for building effective ensembles, avoiding the problem of combining identical or very similar models (Didaci et al. 2013), and various ensemble schemes have been devised and studied. Yet, there is no definitive taxonomy of classifier ensembles and the most up to date emerged combination techniques provide a rich collection for tackling any kind of problems. One can in general distinguish two classifier combination models according to the type of base classifiers. Homogeneous classifiers based model (perturb & combine approach) which uses the same learning algorithm as the basis for individual classifiers on different distributions of the training set, obtained by resampling techniques (Breiman 1996; Schapire and Freund 2012), and heterogeneous classifiers based model which generates classifiers using different learning algorithms on the training set (Wolpert 1992; Xu et al. 2012). However, another category also exists which consists of hybrid systems that mixes different models (Baumgartner and Serpen 2012; Whalen and Pandey 2013). For more details on ensemble methods, an excellent review can be found in Dietterich (2002) and more recently in Sewell (2011). The most popular way of combining multiple classifiers consists in using jointly the opinions of all the available classifiers for the same input. The final decision is usually made using simple aggregation rules (non-trainable combiners) instead of using the outputs for learning at a meta-level (trainable combiners). This study concentrates on the first strategy. We focus only on the most commonly used decision rules for classifier combination. To the best of our knowledge, there is no best decision rule for all situations and a significant gain in performance is not always guaranteed. Yet, ensemble methods have been widely used to address bioinformatics problems such as gene regulatory networks inference (Zong et al. 2010; Jiao et al. 2013) and protein secondary structure prediction (PSSP) which have been among the growing trends in this field. In PSSP, the sequential nature of data requires specific ensembles. Common ensembles based on resampling or injecting randomness could not be applied since the order of amino acids in protein sequences is essential for coherent predictions and the prediction success relies on amino acids dependencies. Thus, special attention must be paid to simple aggregation rule-based ensembles to explore much more fully their generalization ability in PSSP. The present study was somewhat inspired by Kittler et al. (1998) experiments for identity recognition, where some simple aggregation rules were investigated and compared. The authors have shown that sum rule outperformed all the other rules. Here, it is of interest to see whether this rule gives quite good performance in PSSP. Another question of interest is to estimate the upper prediction limit for this type of ensemble methods. Accordingly, the paper has three contributions. The first is an investigation of the most well-known fixed aggregation rules on PSSP problem, to provide a comparison of their performance since they have rarely been compared to each other in this context, integrating as ensemble member classifiers that yield good performance as protein secondary structure predictors, namely Multi-Class Support Vector Machines (M-SVMs) and Artificial Neural Networks (ANNs). The second contribution is an analysis of a new rule to aggregate the individual ensemble member decisions named Exponential Opinion Pool. The proposed consensus scheme is assessed so as to estimate how much improvement in PSSP performance it can give rather than each studied ensemble method and each individual ensemble member. The third contribution is to establish an upper limit of the prediction accuracy (i.e., the best possible prediction accuracy) of the studied ensemble schemes for PSSP problem according to the integrated ensemble members. The experimental comparison of the designed ensemble schemes is performed using the most common PSSP evaluation metrics. To do so, the different ensemble methods have been implemented in ANSI C code and executed on linux environment.

The remainder of this paper is organized as follows. In the next section, we shortly review the combination rules that are investigated and compared in this study and explain how the outputs of the chosen ensemble members are combined. Next, Sect. 3 gives a brief introduction to the PSSP problem. Section 4 covers the secondary structure prediction tools and benchmark datasets used. Section 5 is devoted to the detailed description of the experimental results. Section 6 concludes and closes the paper with some directions for future research.

2 Methods

A classifier generally assigns a class for each example, but it may also provide confidence values estimating the probability of belonging to each one of the possible classes. So, the performance of the classifier is not only related to the score achieved, but also to the estimated class posterior probabilities which give insights on the prediction quality and enable eventual post-processing. Many actual studies focus on the dependency between the classification score and the estimated class posterior probabilities. In this study, the class posterior probabilities provided by each classifier are considered. Given a feature set $\mathcal{X}$, a classifier is a function f that maps an example $x = (x_1, \ldots, x_d) \in \mathcal{X}$ ($\mathcal{X} \subset \mathbb{R}^d$) to a class $c_k \in \mathcal{Y} = \{c_1, c_2, \ldots c_Q\}$, a set of Q class values.

$$f : \mathcal{X} \subset \mathbb{R}^d \longmapsto \mathcal{Y}, \quad \mathcal{Y} = \{c_1, c_2, \ldots, c_Q\} \tag{1}$$

Each class c_l is modeled by the probability density function $P(x_1, \ldots, x_d|c_l)$ and its a priori probability of occurrence $P(c_l)$. According to Bayesian theory, the example x should be assigned to class c_k with the highest value of posterior probability such that:

$$f(x) = c_k, \quad k = \underset{l=1,\ldots,Q}{\operatorname{argmax}} P(c_l|x_1, \ldots, x_d) \tag{2}$$

Then, by applying Bayes Theorem, we have:

$$P(c_k|x_1, \ldots, x_d) = \frac{P(x_1, \ldots, x_d|c_k)P(c_k)}{P(x_1, \ldots, x_d)} \tag{3}$$

The unconditional joint probability density is expressed in terms of the conditional feature distributions, so that:

$$P(x_1, \ldots, x_d) = \sum_{l=1}^{Q} P(x_1, \ldots, x_d|c_l)P(c_l) \tag{4}$$

Let us represent by $\mathcal{C}$ a set of M individual classifiers generated by applying the learning algorithms $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M$ to a single training data set $\mathcal{X}$. We assume that each classifier from $\mathcal{C}$ predicts the posterior probability for each class $c_l, l \in \{1, \ldots, Q\}$. For an example $x \in \mathcal{X}$, the Q component vector of the predicted class posterior probabilities by a classifier $\mathcal{A}_l$ is then:

$$P_{\mathcal{A}_l} = (P_{\mathcal{A}_l}(c_1|x), P_{\mathcal{A}_l}(c_2|x), \ldots, P_{\mathcal{A}_l}(c_Q|x)) \tag{5}$$

where $\{c_1, c_2, \ldots c_Q\}$ is a set of possible class values and $P_{\mathcal{A}_l}(c_l|x))(l \in \{1, \ldots, Q\})$ represents the class posterior probability estimating the probability that the example x belongs to the class c_l. Once the M predictions of the individual classifiers $\mathcal{A}_l$ of $\mathcal{C}$ for an example x are obtained, they are combined in some way to produce the final decision of the total ensemble $\mathcal{C}$.

2.1 Simple rule-based ensemble method

This section introduces the usual aggregation rules that are investigated in this study. All the rules described below deal with class posterior probabilities. The others simple rules for combining classifiers outputs, based on the generated class labels only, like the naive Bayes combiner and the (weighted) majority vote are not considered in this stage. However, the latest combiner is chosen for a second stage of classification. To describe each rule, let us define the so-called Decision Profile matrix introduced by Kuncheva et al. (2001). The decision profile matrix DP(x) for an example x consists of elements $d_{jk} \in [\![0, 1]\!]$ which represent the support given by the jth classifier to a class c_k. The rows of DP(x) represent the support given by individual classifiers to each of the classes, whereas the columns represent the support received by a particular class from all classifiers.

$$\mathrm{DP}(x) = \begin{bmatrix} d_{11}(x) & d_{12}(x) & \ldots & d_{1Q}(x) \\ . & . & . & . \\ d_{M1}(x) & d_{M2}(x) & \ldots & d_{MQ}(x) \end{bmatrix} \tag{6}$$

where M represents the number of classifiers and Q the number of classes. Each element d_{jk} of the matrix can be expressed in term of probabilities using Eqs. (3) and (4) as:

$$d_{jk} = \frac{P(x_1, \ldots, x_d|c_k)P(c_k)}{\sum_{l=1}^{Q} P(x_1, \ldots, x_d|c_l)P(c_l)} \tag{7}$$

The total support received by a class c_k which depends on the kth column of the decision profile DP(x) can be expressed as:

$$\begin{aligned} \mu_k(x) &= \mathcal{R}[d_{1k}(x), \ldots, d_{Mk}(x)] \\ &= \mathcal{R}[P_{\mathcal{A}_1}(c_k|x), , \ldots, P_{\mathcal{A}_M}(c_k|x)] \end{aligned} \tag{8}$$

where $\mathcal{R}$ is the combination rule, such as one of those listed below (i.e., Sum/Mean, Product, Max, and Min). Usually, each developed rule takes into account the class posterior probabilities estimated by each classifier in the ensemble.

2.1.1 Sum/Mean rule

The total support for a class c_k is obtained as the sum/average of the kth outputs of all the considered classifiers, as follows:

$$\mu_k(x) = \sum_{j=1}^{M} d_{jk}(x)$$

$$\mu_k(x) = \frac{1}{M}\sum_{j=1}^{M} d_{jk}(x) \tag{9}$$

The errors in the confidences are averaged out by the summation. In either case, the ensemble decision is taken as the class c_k for which the total support $\mu_k(x)$ is highest. By averaging the classifier predictions, the risk of selecting a very bad model is reduced.

2.1.2 Product rule

The product rule multiplies the kth outputs provided by the M classifiers. Thus, the final support received by a class c_k is expressed as follows:

$$\mu_k(x) = \frac{1}{M}\prod_{j=1}^{M} d_{jk}(x) \tag{10}$$

This rule is very sensitive to the low support (very small or close to 0) due to the nature of the multiplication by zero, which reduces the chance to the other classifiers to be selected. However, it will be good for independent classifiers but unfortunately, this situation is unrealistic in practice since classifiers are never really independent.

2.1.3 Max rule

Applied to the outputs produced by the set of M classifiers, this rule takes the maximum among the classifier outputs for each class. The final support received by a class c_k is:

$$\mu_k(x) = \underset{j=1,\ldots,M}{\operatorname{argmax}} \quad d_{jk}(x) \tag{11}$$

Unfortunately, this combination rule does not guarantee good performance for the simple reason that the final decision is sensitive to over-fitting. If some classifiers are more overtrained than others, by applying this rule their confidences may be considered.

2.1.4 Min rule

Similarly to the Max rule, the final support provided by the Min rule is given by:

$$\mu_k(x) = \underset{j=1,\ldots,M}{\operatorname{argmin}} \quad d_{jk}(x) \tag{12}$$

This rule is known to have a difficulty to improve the performance, especially when the base classifiers have no comparable success.

2.2 Weighted pooling

In the aforementioned rules, the combination strategy does not take into account the fact that some ensemble members may be more accurate than others, since their opinions are considered with the same importance (are given uniform weights) for deducing the final decision. It may be advantageous to consider their relative influence by assigning to each ensemble member a weight proportional to its performance. Thus, the total support received by a particular class c_k $(1 \leqslant k \leqslant Q)$ can be expressed as follows:

$$\mu_k(x) = F(w_j, d_{jk}(x)), \quad j = 1, \ldots, M \tag{13}$$

where the function F represents the pooling operator and w_j denotes the weight associated with the opinion of the jth ensemble member. The assigned weights obey $\sum_{j=1}^{M} w_j = 1$ and $w_j \geq 0$. There has been different suggestions to assign weights to the individual models. Generally, they can be fixed or dynamically determined. The most common way is that the weights are set proportional to the performance on the training set (Opitz and Shavlik 1996) according to the formula:

$$w_j = \frac{1 - E_j}{\sum_{l=1}^{M}(1 - E_l)} \tag{14}$$

where E_j is the individual classifier's error on the training set. In this study, a newly proposed consensus scheme that we named Exponential Opinion Pool (ExpOP) is investigated against the two common weighted ensembles, Linear Opinion Pool (LinOP) and Logarithmic Opinion Pool (LogOP). Furthermore, two additional pooling rules are analyzed namely, weighted Min (WMin) and weighted Max (WMax) besides Decision Templates and Dempster–Shafer aggregation rules.

2.2.1 Linear Opinion Pool

The Linear Opinion Pool (LinOP) or weighted algebraic average is the most common way for combining classifiers taking the linear mean of their weighted opinions. The total support received by a particular class c_k $(1 \leqslant k \leqslant Q)$ is thus as follows:

$$\mu_k(x) = \sum_{j=1}^{M} w_j d_{jk}(x) \tag{15}$$

2.2.2 Logarithmic Opinion Pool

The logarithmic Opinion Pool (LogOP) or weighted geometric average (Hansen 2000) is a weighted geometric mean of the individual opinions. The support received by a particular class c_k $(1 \leqslant k \leqslant Q)$ from all classifiers is given by:

$$\mu_k(x) = \frac{1}{Z} \exp\left(\sum_{j=1}^{M} w_j \log(d_{jk}(x))\right) \tag{16}$$

where Z is a normalization factor satisfying:

$$Z = \sum_{l=1}^{Q} \exp\left(\sum_{j=1}^{M} w_j \log(d_{jl}(x))\right) \tag{17}$$

2.2.3 Exponential Opinion Pool

The proposed pooling rule named Exponential Opinion Pool (ExpOP) combines the weighted opinions so as the aggregate opinion for a particular class c_k $(1 \leqslant k \leqslant Q)$ becomes:

$$\mu_k(x) = \frac{\exp\left(\sum_{j=1}^{M} w_j d_{jk}(x)\right)}{\sum_{j=1}^{M} \exp(w_j d_{jk}(x))} = \frac{\prod_{j=1}^{M} \exp(w_j d_{jk}(x))}{\sum_{j=1}^{M} \exp(w_j d_{jk}(x))} \tag{18}$$

It is fairly easy to see that $\frac{1}{Me} \leqslant \mu_k(x) \leqslant \frac{e}{M}$ and thus is inevitably in $[\![0, 1]\!]$ interval.

2.2.4 Weighted Max rule

Applied to the outputs produced by the set of M classifiers, this rule takes the maximum among the classifiers weighted outputs for each class. The final support received by a class c_k is thus:

$$\mu_k(x) = \underset{j=1,\ldots,M}{\operatorname{argmax}} \quad w_j d_{jk}(x) \tag{19}$$

This rule gives too much importance to the powerful individual classifiers which probably increases the ensemble performance.

2.2.5 Weighted Min rule

Similarly to the Weighted Max rule, the final support provided by the weighted Min rule is given by:

$$\mu_k(x) = \underset{j=1,\ldots,M}{\operatorname{argmin}} \quad w_j d_{jk}(x) \tag{20}$$

2.3 Decision Templates

Decision Templates combiner operates using Decision Profile (DP) concept (Kuncheva 2001). It consists in building the most typical DP for each class from training data based on the ensemble member outputs. A Decision Template for each class is thus the mean of such decision profile. So, for each class c_k $(1 \leqslant k \leqslant Q)$, the corresponding Decision Template is expressed as:

$$\mathrm{DT}_k = \frac{1}{N_k} \sum_{\{x \mid f(x)=c_k\}} \mathrm{DP}(x) \tag{21}$$

where N_k is the number of training examples belonging to the class c_k. To label a test example x, the combiner makes a final decision by comparing each $\mathrm{DP}(x)$ to the Q template matrices $(\mathrm{DT}_1, \ldots, \mathrm{DT}_Q)$ based on a similarity measure (Euclidean, Mahalanobis, Minkowski,...). The closest match given by the chosen similarity measure $\mathcal{S}$ is then used to label x. The support received by each class c_k from all classifiers is then given by:

$$\mu_k(x) = \mathcal{S}(\mathrm{DP}(x), \mathrm{DT}_k) \tag{22}$$

Using $\mathcal{S}$ as the squared Euclidian distance Eq. (22) will be equivalent to:

$$\mu_k(x) = 1 - \frac{1}{M \times Q} \sum_{j=1}^{M} \sum_{l=1}^{Q} [dt_k(j,l) - d_{jl}(x)]^2 \tag{23}$$

where $dt_k(j,l)$ is the (j,l)th entry in the Decision Template DT_k. Consequently, the example x is assigned to the class of highest support.

2.4 Dempster–Shafer theory of evidence

Dempster–Shafer (DS) theory was introduced as a mathematical way to combine measures of evidence from different sources Shafer (1976). Let us consider the multi-class case to briefly describe the method. Multiple classifier outputs are combined using the similarity between the Decision Profile $\mathrm{DP}(x)$ for an input x and the Q Decision Template (DT) matrices expressed as the class means of the classifier outputs for training data, given by Eq. (21). These Decision Templates are then matched to the Decision Profile to obtain the closeness between the DT and the output of each classifier for a test example x. For example using the Euclidean distance, the closeness value of the output of the jth classifier to a class c_k is expressed as follows:

$$\phi_k^j(x) = \frac{(1 + \|\mathrm{DT}_k^j - \mathrm{DP}^j(x)\|^2)^{-1}}{\sum_{l=1}^{Q} (1 + \|\mathrm{DT}_l^j - \mathrm{DP}^j(x)\|^2)^{-1}} \tag{24}$$

Once these closeness values are calculated, a belief degree for each classifier for each one of the classes, for each test example may be obtained as follows:

$$\mathrm{bel}_k^j(x) = \frac{\phi_k^j(x) \prod_{l \neq k} (1 - \phi_l^j(x))}{1 - \phi_k^j(x) \left[1 - \prod_{l \neq k} (1 - \phi_l^j(x))\right]} \tag{25}$$

The support received by a particular class c_k $(1 \leqslant k \leqslant Q)$ from all classifiers is then given by:

$$\mu_k(x) = K \prod_{j=1}^{M} \mathrm{bel}_k^j(x) \tag{26}$$

where K is a normalization factor that maintains the total support in $[\![0, 1]\!]$ interval.

2.5 Ensemble members description

Many studies have shown that Support Vector Machines (SVMs) and multi-class SVMs give higher prediction accuracy than Artificial Neural Networks (ANNs) in PSSP. However, Feed-Forward Neural Network (FNN) like Multi-Layer Perceptron (MLP) or Radial Basis Function Neural Network (RBFNN) might have nearly similar performance in this context in the case of well-chosen architecture and parameters. In this study, the ensemble members consist of six individual classifiers. A Multi-Layer Perceptron (MLP) trained using the backpropagation algorithm with a sigmoidal activation function for both the hidden and output layers, a RBFNN provided by QuickRBF[1] package (Ou et al. 2005) which uses an efficient least mean square error method to determine the weights associated with the links between the hidden layer and the output layer, and four M-SVMs. The latest solves directly the multi-class problem by extending the standard formulation of the SVM to multi-class case. The single optimisation problem is solved using standard quadratic programming (QP) optimisation techniques. The four M-SVMs are the one of Weston and Watkins (M-SVM$_{\mathrm{WW}}$) Weston and Watkins (1998), the one of Crammer and Singer (M-SVM$_{\mathrm{CS}}$) Crammer and Singer (2001), the one of Lee, Lin and Wahba (M-SVM$_{\mathrm{LLW}}$) Lee et al. (2004) and the one of Guermeur and Monfrini (M-SVM^2) Guermeur and Monfrini (2011). The four machines provided in a single package MSVMpack[2] Lauer and Guermeur (2011) share the same architecture but exhibit distinct properties. The soft margin constant C and the kernel parameters are generally empirically optimized by trials or using model selection strategies.

2.6 Ensemble member outputs post-processing

As it has been stated above, the study concentrates on the class posterior probabilities fusion. The standard SVMs as well as the M-SVMs do not provide such probabilities, but an uncalibrated distance measurement of an example to the separating hyperplane in the feature space. Thus, the outputs must be post-processed prior to being combined. Different post-processing can be applied to map SVMs and M-SVMs outputs into class posterior probabilities (Platt 2000). The quality of posterior probability estimates is subject to many recent studies (Zhang and Jordan 2006; Guermeur and Thomarat 2011; Wallace 2012). Here, we used the softmax function which is the most common mapping. The obtained outputs are normalized so as to ensure that the Q outputs are not nulls and always sum to one as follows:

$$d_{jk} = \frac{\exp(o_{jk})}{\sum_{l=1}^{Q} \exp(o_{jl})} \tag{27}$$

where o_{jk} is the kth output (corresponding to the class c_k) of the jth classifier.

3 Protein secondary structure prediction problem

It is well-known fact that the process by which a protein is folding in its three-dimensional (3D) shape gives relevant clues to its function. However, one of the most challenging problems in molecular biology remains precisely the prediction of this 3D structure referred to as tertiary structure. The amino acids sequence (primary or 1D structure) has a great importance for this aim since at its own dictates the required structure (Anfinsen's dogma) (Anfinsen 1973). Despite many decades of intensive research, all the attempts remain insufficient to solve this problem. To date, the experimental determination of this structure remains a difficult task. Although the experiments are accurate, they are still laborious, expensive, time-consuming, and sometimes unfeasible. For all these reasons, computational methods appear as good alternatives to address this problem. Presently, machine learning approaches become increasingly important in this context. Because of their usefulness in functional annotation of the ever growing number of newly discovered proteins, they are still topic of extensive research. A large number of approaches try to predict the protein's structural features such as solvent accessibility, contact maps, disulfide bonding state and secondary (2D) structure rather than the full tertiary structure, because of its computational complexity. The secondary structure represents the structural conformations conventionally grouped into three types of patterns: the two common regular patterns α-helices and β-sheets (Extended strands), originally predicted by Pauling et al. (1951) and the random coils which represent all the other patterns without apparent regularities. The mapping from 3D to 2D structures by projection onto strings of structural states represented by single letters (DSSP code, see Sect. 4.1) is a fundamental intermediate step toward the full 3D structure elucidation, taking into account only the amino acid sequence. The secondary structure prediction can be analyzed as a typical problem of pattern recognition, where the category (structural state) of a given amino acid named also residue in the sequence is predicted in one of the three common states: helix (H), sheet (E) or coil (C). Numerous methods using different algorithmic approaches have been proposed for this aim. The best results have been achieved using evolutionary information in the form of protein sequence profiles (position-dependent frequency vectors derived from multiple alignments) rather then using only amino acid sequences. The average tree-state per-residue score Q_3 [a prediction accuracy measure that gives the percentage of correctly predicted

[1] http://muse.csie.ntu.edu.tw/~yien/quickrbf/index.php.

[2] http://www.loria.fr/~lauer/MSVMpack.

secondary structures (Qian and Sejnowski 1988; Rost and Sander 1993)] has improved from 33 % (random guessing) reaching a limited value of 55 % for the first generation methods (single residue statistics-based prediction) and increased to over 60 % to attain a limit of 67 % for the second generation methods (segment statistics-based prediction). Thereafter, an improvement to over 70 % was achieved by the third generation methods based on the evolutionary information. However, the latest generation requires the existence of similar proteins (homologous sequences) with known structures. So, the difficulty of getting better results remains for the orphan proteins which do not exhibit significant similarity to any already categorized protein in the PDB[3] (Brookhaven Protein Data Bank of solved structures). Hence, better prediction methods for single sequences are highly required. Recently, there have been evolutionary information based-consensus approaches combining results from different predictors, achieving even higher accuracy. The claimed Q_3 score varies between 75 and 80 %, depending on the benchmark datasets used (Bouziane et al. 2011). Presumably, all existing machine learning algorithms have been applied to the PSSP problem and further improvements of few percentage points are still required. Undoubtedly, the scientific community is expecting a progress similar to that caused by the use of evolutionary information, which would be a good standpoint for a new generation of PSSP methods.

The aim of this paper in this context is not to announce a new Q_3 score but to quantify the usefulness of combining multiple opinions by investigating how the performance can be enhanced by the different ensemble methods on both single amino acid sequences and sequence profiles in the form of Position-Specific Scoring Matrix (PSSM) generated by Position-Specific Iterated Basic Local Alignment Search Tool (PSI-BLAST).[4]

4 Secondary structure prediction tools and benchmarks

4.1 Protein secondary structure definition

As mentioned above, the most common secondary structures are α helices and β sheets (extended strands). However, one precisely distinguishes a total of 8 secondary structure conformations: H(α-helix), I(π-helix), G(3_{10}-helix), E(β-strand), B(isolated β-bridge), T(hydrogen bonded turn), S(bend), and rest (apparently random conformations). The majority of PSSP methods deals with 3 conformational states (H, E, C), generated from the 8 states using assignment methods. Generally, defining the boundaries between helix, sheet and coil structures is arbitrary and standard assignments do not exist. As the assignment method influences the prediction accuracy (Cuff and Barton 1999), one generally tends to use an assignment scheme which leads to higher performance. Dictionary of secondary structure of proteins program (DSSP)[5] (Kabsch and Sander 1983) is the most adopted one. The principle consists in assigning secondary structures according to hydrogen bonding patterns in known structures. In this study, we concentrate exclusively on this method using the scheme that treats B(isolatedβ-bridge) as part of a β-sheet (E), which is explicitly: H, G to H; E, B to E; all other states to C. This assignment scheme increases the proportions of the three states which might be helpful for avoiding minority classes.

4.2 Preprocessing and data coding

Given a set of proteins, structural class prediction is usually performed in two steps: conversion of the set of sequences from strings of alphabetic characters in capital letters, corresponding to the twenty naturally occurring amino acids into a feature-based representation and then fit it into the predictor. The most common approach for protein sequences encoding uses the window concept, so as each queried residue is typically represented as a set of input features and its corresponding observed class label. So, one considers a sliding window of typically 11, 13 until 21 amino acids and predicts the class label of the central amino acid of the window, knowing only its neighbors. The observed class labels are secondary structure states according to the chosen assignment scheme. Some works have paid more attention to analyse the influence of the window size on the prediction result. Therefore, for this study, we opted for a window of length 13 which is common to cover relevant sequences. All the experiments are performed using sevenfold cross-validation on the used protein datasets. At each fold, one subset is used for test and the 6 other subsets constitute the learning set. Afterwards, the results are averaged. For each example (queried residue), the predicted secondary structure state is then compared with the observed secondary structure for performance assessment.

4.3 Training and testing datasets

Some protein non-redundant datasets have been developed to make objective evaluation and comparison of PSSP methods. Non-redundancy measures have been established to remove the internal homology. Here, we experiment with two datasets RS126 and CB513. The first has been proposed by Rost and Sander (1993). It contains 126 non-homologous protein chains with a total of 23,349 residues and an average protein sequence length of 185. In this dataset, the non-redundancy has been defined by Sander and Schneider

[3] http://www.rcsb.org/pdb/.

[4] http://www.ncbi.nlm.nih.gov/BLAST/.

[5] http://swift.cmbi.ru.nl/gv/dssp/.

(1991) as no two proteins in the dataset share more than 25 % sequence identity over a length of more than 80 residues, a measure which has been considered poor by Cuff and Barton (1999). So, a more sophisticated similarity measure using the so-called SD score has been proposed and gave birth to the most used independent dataset CB513 with a total of 84,119 residues. It contains 396 sequences, usually named CB396 and the 117 effectively non-redundant sequences from RS126 (9 sequences were removed according to the given definition of SD score) with an average sequence length of 179. The distribution of secondary structure types in both the two datasets is uneven and that is a common property in protein datasets. There is approximately 32 % α-helix, 23 % β-sheet, and 45 % coil in RS126 dataset and about 36.4 % α-helix, 22.9 % β-sheet and 41.5 % coil in CB513 dataset. The two datasets are available at the Barton Group website.

4.4 Architecture and parameter settings

When using each classifier for secondary structure prediction, settings for the different hyperparameters have a great importance for prediction performance. Here, the identified hyperparameters have been tuned by trials. The MLP has been experimented using a single hidden layer with 10 units and QuickRBF is used with 12,000 selected centers. When experimenting with single sequences, the M-SVMs were used with a dedicated kernel proposed in Guermeur et al. (2004). Whereas the Gaussian (rbf) kernel is used with PSSM profiles. The penalty parameter $C = 1.0$ and the kernel parameter $\gamma = \frac{1}{10 \times d}$, where d refers to the dimensionality of the input examples. The classifiers have been modified so as to produce class posterior probability estimates besides the class label. Euclidean distance is chosen as similarity measure for both Decision Templates and Dempster–Shafer combiners.

4.5 Evaluation metrics

For evaluating the performance of both individual classifiers and ensemble methods, we used the standard accuracy measures suggested in the literature for PSSP methods. The most popular measure is the Q_3 score. Complementary measures such as the Matthews correlation coefficients ($C_\mathrm{H}, C_\mathrm{E}, C_\mathrm{C}$) Matthews (1975) and the segment overlap SOV (Rost and Sander 1994; Zemla et al. 1999) are also calculated to evaluate the performance.

5 Results and discussion

In this section, four more popular simple aggregation rules are investigated namely, Sum, Product, Min and Max besides Dempster–Shafer and Decision Templates combiners. The resulting ensemble methods are compared to five weighted opinion pooling-based ensembles in terms of Q_3 score, Matthews correlation coefficients ($C_\mathrm{H}, C_\mathrm{E}, C_\mathrm{C}$) and SOV measure. The experiments are also aimed to compare the performance of individual classifiers and each ensemble method. The section is split into three parts: experiments using only the amino acid sequences, experiments using sequence profiles generated from multiple sequence alignments and experiments estimating the upper prediction limit of the designed ensemble methods, according to the integrated ensemble members. The results of each part are reported and discussed below.

5.1 Single sequences based experiments

For these experiments, each example (residue in the sequence) is represented by a set of 13 features (amino acids in the window) and a class label. Each amino acid has its own code varying from 0 to 21, corresponding to the 20 naturally occurring amino acids and the value 21 represents unknown amino acids, which are usually designed by ‘X’ or ‘?’ in the published databases. The value 0 represents an empty position in the window. The chosen individual classifiers and ensemble methods are applied to both RS126 and CB513 benchmark datasets. The individual classifiers prediction results are reported in Tables 1 and 2. The ensemble methods prediction results are tabulated in Tables 3 and 4. The Q_3 score

Table 1 Performance comparison of the six individual classifiers for RS126 dataset using single sequences

Classifiers	Accuracy measures									
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV$_\mathrm{H}$ (%)	SOV$_\mathrm{E}$ (%)	SOV$_\mathrm{C}$ (%)	SOV (%)
M-SVM$_\mathrm{CS}$	65	61.89	43.84	76.73	0.478	0.389	0.431	54.2	53.1	59.3	54.3
M-SVM$_\mathrm{LLW}$	64.89	63.49	35.27	79.30	0.471	0.367	0.435	56.8	47.5	59.6	54.4
M-SVM^2	65.07	62.99	43.12	76.47	0.475	0.388	0.437	55.3	53	59.1	54.7
M-SVM$_\mathrm{WW}$	65.09	60.59	42.31	78.50	0.473	0.386	0.436	55.3	53.6	61.1	56.4
MLP	62.25	57	39.47	76.16	0.405	0.343	0.408	51.4	49.9	58.8	52.8
RBFNN	64.60	59.43	42.07	78.34	0.461	0.380	0.431	54	53.4	60.9	55.4

Table 2 Performance comparison of the six individual classifiers for CB513 dataset using single sequences

Classifiers	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV_H (%)	SOV_E (%)	SOV_C (%)	SOV (%)
M-SVM$_{CS}$	65.35	66.25	44.97	75.41	0.504	0.406	0.446	58.5	51.2	60.9	57.7
M-SVM$_{LLW}$	65.26	69.99	38.35	75.68	0.501	0.394	0.448	60.9	46.6	61.6	57.9
M-SVM^2	65.61	67.30	43.08	76.17	0.508	0.408	0.450	60.4	50.1	62	58.9
M-SVM$_{WW}$	65.67	66.63	44.30	76.22	0.510	0.411	0.450	60.3	51.1	61.9	59
MLP	64.60	66.21	44.72	73.81	0.490	0.391	0.440	55.3	49.7	60.2	55.4
RBFNN	65.13	66.10	44.93	75.05	0.497	0.402	0.447	57.9	50.7	61.4	57.4

Table 3 Performance comparison of the six ensemble methods for RS126 dataset using single sequences

Aggregation rules	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV_H (%)	SOV_E (%)	SOV_C (%)	SOV (%)
Sum	65.20	61.55	40.78	78.77	0.473	0.387	0.438	56	52.4	60.9	55.9
Product	65.19	61.55	40.76	78.76	0.474	0.386	0.438	55.7	52.4	60.8	55.8
Min	64.86	60.56	39.97	79.08	0.463	0.381	0.436	55.1	52.2	60.7	55.4
Max	65.06	62.38	39.71	78.41	0.471	0.381	0.439	56.6	51.4	60.7	55.7
DS	65.14	64.45	51.06	72.01	0.482	0.401	0.447	56.5	59.1	57.9	55.9
DT	65.14	64.35	51.04	72.09	0.482	0.401	0.447	56.5	58.7	57.8	55.8
WMin	64.68	60.13	39.63	79.16	0.458	0.377	0.435	54	51.9	60.7	54.9
WMax	65.21	62.38	40.23	78.48	0.474	0.386	0.439	56.6	51.9	60.8	55.8
LinOP	65.19	61.59	40.78	78.73	0.473	0.387	0.438	56	52.4	60.8	55.9
LogOP	65.19	61.59	40.78	78.73	0.474	0.387	0.438	56	52.4	60.8	55.9
ExpOP	65.20	61.54	40.76	78.80	0.474	0.386	0.438	55.7	52.4	60.8	55.8

Table 4 Performance comparison of the six ensemble methods for CB513 dataset using single sequences

Aggregation rules	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV_H (%)	SOV_E (%)	SOV_C (%)	SOV (%)
Sum	65.89	67.94	43.28	76.22	0.514	0.412	0.454	61.6	50.3	62.2	59.4
Product	65.88	67.89	43.24	76.25	0.513	0.411	0.454	61.6	50.4	62.2	59.5
Min	65.72	67.64	42.61	76.40	0.510	0.406	0.453	60.5	49.8	61.8	58.7
Max	65.79	68.85	42.38	75.71	0.510	0.408	0.454	61.8	49.7	62.1	59.2
DS	65.72	67.36	51.53	71.92	0.516	0.416	0.455	61	56	61.1	60
DT	65.73	67.20	51.49	72.08	0.516	0.416	0.455	60.8	56	61.1	59.9
WMin	65.70	67.66	42.59	76.35	0.511	0.406	0.452	60.5	49.7	61.6	58.6
WMax	65.76	68.74	42.43	75.72	0.510	0.409	0.454	61.7	49.6	62	59.1
LinOP	65.89	67.93	43.27	76.21	0.513	0.412	0.454	61.6	50.3	62.2	59.4
LogOP	65.87	67.87	43.23	76.25	0.513	0.411	0.454	61.6	50.4	62.2	59.5
ExpOP	65.90	67.93	43.27	76.22	0.513	0.412	0.454	61.5	50.3	62.2	59.4

values in both Tables 1 and 2 reveal that the highest score is achieved for coil state, followed by helix and then sheet.

From Table 3, we can see for RS126 dataset that the Q_3 score ranges from 62.25 to 65.09 %, where the lowest value is given by the MLP. From these results, we can see that the four M-SVMs and the RBFNN achieve better performance. However, the RBFNN appears somewhat less competent than the four M-SVMs. In Table 2, the result for CB513 dataset agrees with the previous observation, the highest Q_3 score achieved is 65.61 %. Still M-SVM$_{WW}$ seems to be the best one among the three other machines. The results show well the high success of the M-SVMs in predicting secondary structure.

From Tables 3 and 4, we can see that the Q_3 score has increased over the best individual model by 0.21 % for RS126

Table 5 Performance comparison of the six individual classifiers for RS126 dataset using PSSM profiles

Classifiers	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV_H (%)	SOV_E (%)	SOV_C (%)	SOV (%)
M-SVM_{CS}	77.79	77.17	64.85	84.35	0.710	0.621	0.611	72.9	66.0	71.4	70.9
M-SVM_{LLW}	77.62	76.68	64.47	84.49	0.713	0.612	0.608	71.8	64.9	70.1	69.2
M-SVM^2	77.94	76.82	65.86	84.43	0.717	0.624	0.610	73.3	65.8	70	70.3
M-SVM_{WW}	78.11	77.36	65.80	84.46	0.724	0.624	0.610	73.0	65.7	70.2	70.1
MLP	74.42	73.25	61.24	81.47	0.662	0.554	0.562	64.0	60.9	63.9	62.7
RBFNN	77.05	75.92	62.49	84.72	0.709	0.601	0.595	71.0	62.0	68.1	67.1

dataset and 0.22 % for CB513 dataset, a small but statistically significant difference.

The results reported in Tables 3 and 4 clearly show the well-performing ensemble schemes and the very similar ones. More visible differences were noticed when showing the results of the individual models and the ensemble results. Compared to the best individual model, the prediction accuracies for the three classes have improved with ensemble methods, Matthews correlation coefficients (C_H, C_E, C_C) and the SOV score became better than those of the best individual model. However, in Table 3, we can see that for RS126 dataset, the Min rule-based ensemble performed worse than the best individual model and from Table 4, we can see that the performance improvement for CB513 is also far less than in the other ensemble methods. So, we face a case where there is no improvement of the ensemble over their members performance. Notice that the sheet state is always significantly underpredicted than the other states, even when ensemble methods are employed. This is because the length of the window has a difficulty to capture long-range interactions between amino acids. The guarantee for a high precision of β-sheets prediction still remains rather low even if a quite good window size is used. This is one of the major drawbacks of the windowing methods (Chen and Chaudhari 2006). Thus, better prediction of β-sheets remain the main challenge to face when using typical prediction methods. Another further interesting observation that can be drawn from Tables 3 and 4 is that besides the Min rule, the cases where no significant improvement over individual models was achieved are by DS and DT rule-based ensemble methods, this confirms that the power of the resulting ensemble may be significant only when combining models of comparable success. The major limitation of DS rule comes from the requirement that the decisions of the ensemble members must be independent. However, considering the so close results of the four M-SVMs, it is unlikely that they are independent as required by this fusion rule. Consequently, DS rule may not be the best suited to combine the outputs of such classifiers. It is worth noting that the two ANNs have been integrated, so as to prevent the ensemble members to perform equally in both cases. The assumption that the MLP and the RBFNN produce predictions different from those of the four M-SVMs come from the fact that they are based on different classification architectures and principles. Different architectures lead more likely to complementary classifiers (Tuliakov et al. 2008). All the experiments showed that DT and DS have similar performance. The Sum rule and the three weighted opinion pooling schemes LinOP, LogOP and ExpOP have also approximately the same performance. So, there is no clear preference of one weighted pooling scheme over the rest. However, it is obvious that ExpOP is good in all cases. In other hand, the Wmax combiner achieved better results than Max and Wmin which performed worse.

5.2 PSI-BLAST derived profiles based experiments

It has long been established that prediction using multiple sequence alignment of protein sequences with homologous proteins rather than single sequences is more effective. In this part of experiments, we used Position-Specific Scoring Matrix (PSSM) profiles generated by PSI-BLAST for RS126 and CB513 datasets, setting the parameter j (number of iterations) to 3, using an e value threshold of 0.001 with the nonredundant NCBI's nr[6] database as sequences database and BLOSUM62 matrix scores for each alignment position. The profile matrix elements obtained in the range ±7 are scaled to the required 0–1 range to fit into the predictor. So, prediction at a given position in the window depends on amino acid frequencies in the profile at the position and neighboring positions within a range defined by the window. Here, a window of length 13 is used, which implies that each input example has 260 (20 × 13) features, besides its observed class label. The individual classifiers prediction results are reported in Tables 5 and 6. Tables 7 and 8 show the ensemble methods prediction results. From Table 5, we can see for RS126 dataset that the Q_3 score ranges from 74.42 to 78.11 %, where the lowest value is given by the MLP. From these results, we can see that the four M-SVMs and the RBFNN achieve good

[6] ftp://ftp.ncbi.nih.gov/blast/db.

Table 6 Performance comparison of the six individual classifiers for CB513 dataset using PSSM profiles

Classifiers	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV$_\mathrm{H}$ (%)	SOV$_\mathrm{E}$ (%)	SOV$_\mathrm{C}$ (%)	SOV (%)
M-SVM$_\mathrm{CS}$	76.11	76.97	64.96	81.33	0.706	0.614	0.568	71.0	65.3	67.8	69.5
M-SVM$_\mathrm{LLW}$	75.51	78.37	60.66	81.07	0.688	0.599	0.567	72	63.3	68.4	69.8
M-SVM^2	75.80	77.64	62.83	81.17	0.695	0.607	0.568	71.8	64.7	68.4	70
M-SVM$_\mathrm{WW}$	76.08	76.95	64.48	81.51	0.704	0.614	0.568	71.4	65.1	67.9	69.7
MLP	72.97	74.15	62.09	77.77	0.645	0.560	0.532	62.8	62.5	64.7	63.3
RBFNN	76.04	77.39	62.46	82.15	0.700	0.611	0.572	70.7	64.8	69.1	70.1

Table 7 Performance comparison of the six ensemble methods for RS126 dataset using PSSM profiles

Aggregation rules	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV$_\mathrm{H}$ (%)	SOV$_\mathrm{E}$ (%)	SOV$_\mathrm{C}$ (%)	SOV (%)
Sum	78.13	77.23	65.28	84.83	0.720	0.624	0.615	73.8	65.9	71.2	71.2
Product	78.11	77.21	65.24	84.83	0.72	0.62	0.61	73.8	66	71.2	71.2
Min	77.29	75.77	64.58	84.35	0.705	0.610	0.604	71.5	65.6	0.1	69.7
Max	77.76	76.89	65.57	84.13	0.712	0.618	0.611	71.6	66	70.4	69.9
DS	78.17	77.20	69.26	83.09	0.723	0.627	0.615	73.4	68.6	70.2	71
DT	78.20	77	68.91	83.42	0.725	0.626	0.615	73.3	68.4	70.6	71.3
WMin	76.66	75.10	63.56	83.93	0.695	0.598	0.594	70.1	64.2	68.4	68
WMax	78.05	77.17	65.80	84.46	0.716	0.623	0.616	72.5	66.7	71.2	70.9
LinOP	78.14	77.18	65.34	84.86	0.720	0.624	0.615	73.7	65.9	71.1	71.1
LogOP	78.12	77.20	65.28	84.83	0.720	0.624	0.615	73.7	66	71.2	71.2
ExpOP	78.14	77.18	65.34	84.86	0.720	0.624	0.615	73.7	65.9	71.1	71.1

Table 8 Performance comparison of the six ensemble methods for CB513 dataset using PSSM profiles

Aggregation rules	Accuracy measures										
	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV$_\mathrm{H}$ (%)	SOV$_\mathrm{E}$ (%)	SOV$_\mathrm{C}$ (%)	SOV (%)
Sum	76.42	77.78	63.92	81.94	0.708	0.617	0.576	72.4	65.4	69	70.9
Product	76.41	77.76	63.85	81.96	0.708	0.617	0.576	72.2	65.3	69.1	70.8
Min	75.54	76.78	62.83	81.26	0.694	0.601	0.563	69.7	64.5	68.1	68.8
Max	76.11	77.72	64.21	81.11	0.701	0.613	0.572	71.4	65.2	68.6	69.9
DS	76.26	76.59	66.66	81.07	0.707	0.616	0.573	70.9	67.1	68.5	70.5
DT	76.27	76.66	66.48	81.14	0.708	0.616	0.573	71.3	67	68.7	70.7
WMin	75.26	76.62	62.74	80.80	0.690	0.596	0.559	69.1	64.2	67.5	68.1
WMax	76.22	77.78	64.33	81.26	0.703	0.615	0.573	71.8	65.3	68.8	70.2
LinOP	76.42	77.77	63.93	81.95	0.708	0.617	0.576	72.4	65.3	69	70.9
LogOP	76.42	77.75	63.89	81.99	0.708	0.617	0.576	72.2	65.4	69.1	70.9
ExpOP	76.44	77.77	63.94	81.96	0.708	0.618	0.576	72.4	65.4	69	70.9

performance. In Table 6, the results for CB513 show that the Q_3 score varies from 72.97 to 76.11 %, the lowest score is given by the MLP and the highest is given by the M-SVM$_\mathrm{CS}$. As previously mentioned, for almost all the ensemble methods, the accuracy is generally higher for the helical and coil states. The Q_3 score values in both Tables 5 and 6 show that M-SVM$_\mathrm{WW}$ performed well. However, the three other M-SVMs and the RBFNN achieve results nearly similar. The MLP is still the poorer predictor. The results reveal once again the superiority of M-SVMs in predicting secondary structure. This further highlights the power of kernel machines when dealing with difficult problems.

Table 9 Improved performance on both RS126 and CB513 datasets

Data set	Classifiers	Accuracy measures										
		Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	SOV_H (%)	SOV_E (%)	SOV_C (%)	SOV (%)
RS126												
Single sequences	IFS	65.37	62.62	44.33	76.81	0.476	0.400	0.441	56.4	54.1	59.7	55.8
	IFS + Filter	65.74	61.70	43.42	78.63	0.493	0.399	0.438	60.6	54.3	62.8	60.5
	MV	65.23	61.69	40.94	78.68	0.474	0.387	0.439	56.1	52.4	60.8	55.9
	MV + Filter	65.47	60.42	40.15	80.41	0.487	0.388	0.434	60.6	52.8	63.4	60
PSSM profiles	IFS	78.39	77.13	67.91	84.22	0.725	0.631	0.618	73.7	68	70.9	71.5
	IFS + Filter	78.50	76.42	67.23	85.25	0.734	0.631	0.616	76	68.5	71.7	73.7
	MV	78.13	77.18	65.68	84.67	0.720	0.624	0.615	73.8	66.2	71.1	71.2
	MV + Filter	78.30	76.53	64.95	85.82	0.731	0.624	0.613	75.8	67.4	71.8	73.4
CB513												
Single sequences	IFS	65.98	67.62	47.16	74.61	0.517	0.416	0.456	61.2	53	61.7	59.6
	IFS + Filter	66.15	66.78	46.66	75.97	0.528	0.418	0.451	65.6	54	62.4	63.1
	MV	65.88	67.95	43.46	76.08	0.513	0.412	0.454	61.7	50.5	62.2	59.5
	MV + Filter	65.96	66.99	42.76	77.41	0.522	0.413	0.448	65.8	51.2	62.3	62.4
PSSM profiles	IFS	76.44	77.77	63.94	81.99	0.708	0.618	0.577	72.4	65.4	69.1	70.9
	IFS + Filter	76.65	77.03	63.72	83.18	0.716	0.622	0.576	73.5	66.4	69.9	73.1
	MV	76.40	77.60	64.22	81.89	0.707	0.617	0.576	72.2	65.6	69.1	70.9
	MV + Filter	76.58	76.84	63.95	83.06	0.716	0.621	0.575	73.4	66.6	69.9	73

From the results in Tables 7 and 8, some comments can be drawn. Firstly, ensembles performance is better than the single model, except when using Min rule. The Matthews correlation coefficients (C_H, C_E, C_C) increased significantly and SOV measure improved from 60 to 70 %, achieving a gain of 10 %. On the other hand, the best results are obtained when using Sum rule and weighted opinion pooling. WMax is still better than the Max rule. The Sum rule results are better than those of the individual models. LinOP as well as the ExpOP achieved better performance. Even more, ExpOP achieved performance slightly better than Sum rule. The weighted opinion pooling ensembles are more accurate here. Their higher performance can be attributed to the predominance of the M-SVM$_{WW}$ since it has the highest performance. A comparison of the different metrics between the investigated ensemble methods confirms that ExpOP has an ensured potential to improve the performance and it generally outperforms LogOP and provides in some cases an accuracy which is at least comparable to that which would be obtained by LinOP and Sum rule. So, an ExpOP-based ensemble can obtain good results even when the ensemble members are not really independent.

5.3 Improving the consensus results

Experiments have been conducted to see whether the prediction performance could be improved using consensus opinion of the ensembles built previously. Majority Voting (MV) rule is used in the hope to compensate the errors of the designed ensembles. For each residue in the sequence, the predicted secondary structure state corresponds to the class that the majority of ensemble schemes agree on. In the other hand, to estimate the top limit of the prediction accuracy of the designed ensembles on each dataset according to the chosen ensemble members, we design a predictor based on the predictions of certain ensemble schemes. At each fold of the sevenfold cross-validation, the best predicted fold is selected among all the others for the same subset. The so-called "Ideal Fold Selection" (IFS) predictor thus assembles the 7 best predicted folds to form the predicted conformational states of the entire dataset. Furthermore, a main condition for obtaining coherent secondary structures is that the shortest length of consecutive states H must be 3 and 2 for consecutive states E. To eliminate unrealistic structures, the resulting predictions for each conformational state are then refined by applying the heuristic-based filter used in Bouziane et al. (2011). The results are thus reported in Table 9.

As it is shown, for RS126, the Q_3 score increased from 65.23 % (best ensemble) to 65.47 % (consensus by MV rule), from 78.13 to 78.60 % for RS126 PSSM, from 65.88 to 65.96 % for CB513 and from 76.40 to 76.58 % for CB513 PSSM. A slight difference between the scores achieved by the best ensemble and those of the consensus. So, the expected improvement by assigning to queried residues the predictions that the majority of ensembles agree on has not been achieved since it remains far away from the estimated upper

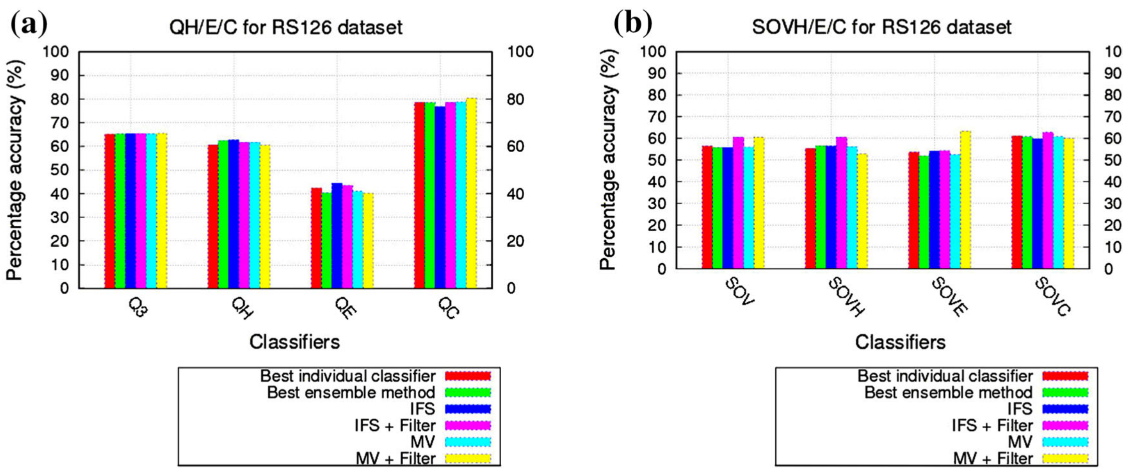

Fig. 1 The Q3 (**a**) and SOV (**b**) scores for the best individual classifier, the best ensemble method, IFS, IFS + Filter, MV and MV + Filter on RS126 dataset. QH/E/C and SOVH/E/C are, respectively, the predicted *Q* and SOV scores for each conformational state (helix, strand and coil)

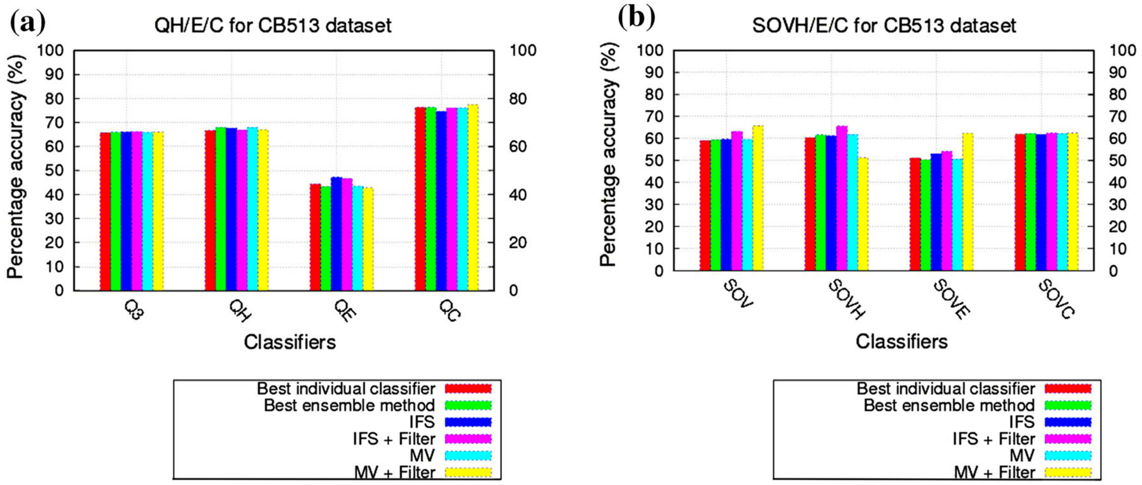

Fig. 2 Q3 (**a**) and SOV (**b**) scores for the best individual classifier, the best ensemble method, IFS, IFS + Filter, MV and MV + Filter on CB513 dataset. QH/E/C and SOVH/E/C are, respectively, the predicted *Q* and SOV scores for each conformational state (helix, strand and coil)

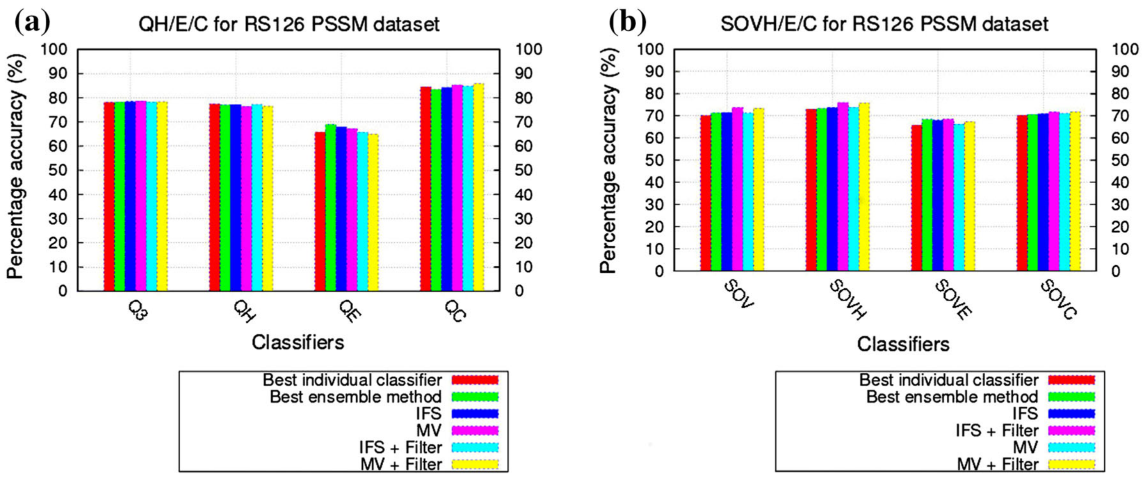

Fig. 3 The Q3 (**a**) and SOV (**b**) scores for the best individual classifier, the best ensemble method, IFS, IFS + Filter, MV and MV + Filter on RS126 PSSM dataset. QH/E/C and SOVH/E/C are, respectively, the predicted *Q* and SOV scores for each conformational state (helix, strand and coil)

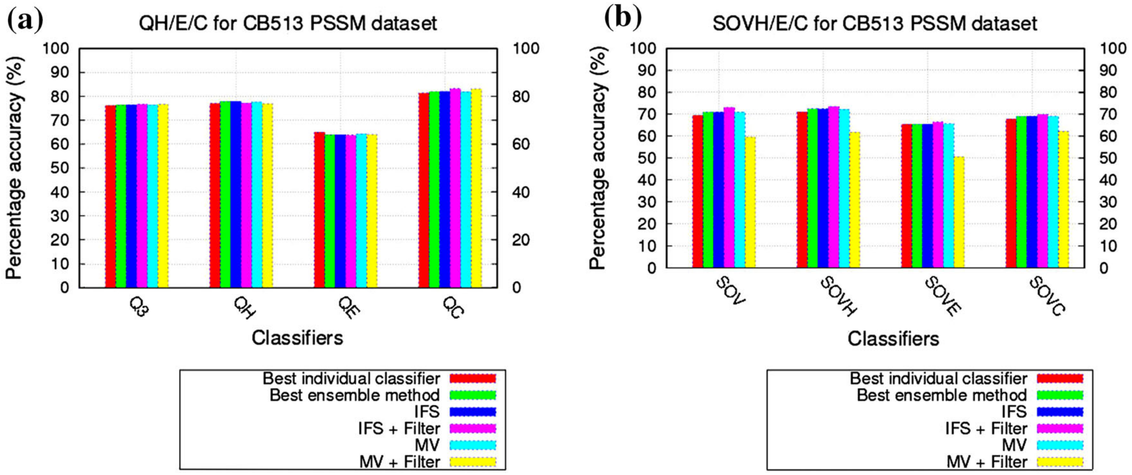

Fig. 4 The Q3 (**a**) and SOV (**b**) scores for the best individual classifier, the best ensemble method, IFS, IFS + Filter, MV and MV + Filter on CB513 PSSM dataset. QH/E/C and SOVH/E/C are, respectively, the predicted Q and SOV scores for each conformational state (helix, strand and coil)

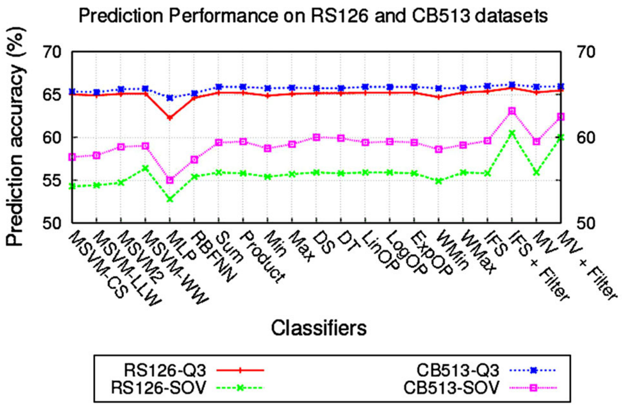

Fig. 5 Comparison of prediction accuracies (y axis) between the best individual classifier, the best ensemble method, IFS, IFS + Filter, MV and MV + Filter (x axis) on both RS126 and CB513 datasets

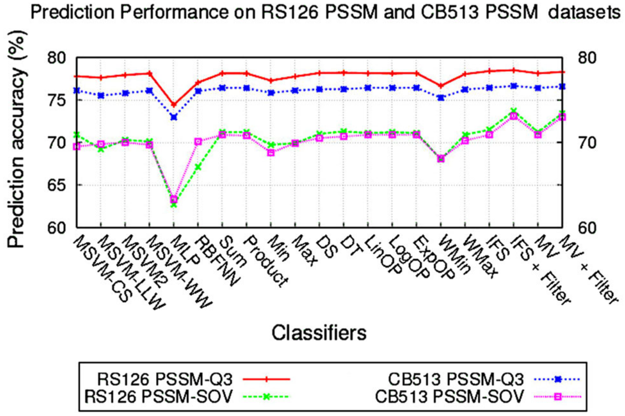

Fig. 6 Comparison of prediction accuracies (y axis) between the best individual classifier, the best ensemble method, IFS, IFS + Filter, MV and MV + Filter (x axis) on both RS126 and CB513 PSSM datasets

limit. However, satisfactory results have been obtained by filtering the predictions, the Q_3 score gained at least 0.18 % and the SOV gained 3 %. The variation of Q_3 and SOV scores is depicted in Figs. 1 and 3 for RS126 and Figs. 2 and 4 for CB513. From the figures illustrated, it is obvious that the improvement over the best individual model is more significant, especially after the filtering stage which increased the Q_3 and SOV values, see Figs. 5 and 6. The Q_3 score has increased by at least 0.50 percentage point which is really a significant difference. Because the improvement in Q_3 is greater than the previously achieved values, the Matthews correlation coefficients ($C_{\mathrm{H}}, C_{\mathrm{E}}, C_{\mathrm{C}}$) increased substantially. The achieved Q_3 scores 78.50 % for RS126 dataset and 76.65 % for CB513 represent quiet respectable levels of performance on the two datasets. It is worth noting that without any real effort for tuning each individual model parameters, the results are close to those achieved by the current prominent PSSP methods. It should also be noted that by improving the post-processing filter, the results might be better. Developing an efficient filter is still matter of ongoing research. The gain in performance is not very impressive but it is possible that by increasing the number of members in the ensemble and using large training datasets, the improvement would be substantial. An important point to note in this discussion is that the idea of exploring trainable aggregation rule-based ensemble methods to analyse their effect on PSSP, would provide useful information on their performance in this context.

6 Conclusion

Many studies have pointed out that simple aggregation rules in some cases can provide better performance than state-of-the-art combination techniques. In this paper, some fixed rules involving class posterior probabilities are experimentally explored and compared to each other for protein secondary structure prediction, using sevenfold cross-validation on RS126 and CB513 benchmark datasets. All the designed ensemble methods are evaluated using PSSP performance measures. Furthermore, a consensus of the subsequent ensembles has been realized using majority vote rule to compare with the estimated upper limit of the prediction accuracy obtained by selecting the best predicted fold in terms of Q_3 score at each fold of the sevenfold cross-validation. The resulting predictions have been analyzed and improved for better performance using an heuristic-based filter. The experiments demonstrate that the M-SVMs performance is significantly better than those of the feed-forward ANNs for PSSP, and ensemble methods perform better than the best individual model. All the results obtained confirm that ensembles are good alternatives for improving prediction performance, especially for difficult problems. An effective combination scheme is to simply sum/average the predictions. The new proposed weighted opinion pooling rule named exponential opinion pool is competitive with Linear Opinion Pool and slightly superior to Logarithmic Opinion Pool. However, it is sometimes as accurate as Sum rule at least in this particular case. The experimental results showed that the benefit is not as much in combining the different ensemble schemes, possibly this may be attributed to the fact that the predictions are highly correlated. In this study, we have not confer special attention to long-range interactions between amino acids. An important direction for future work is to integrate single models that deal with long-range interactions such as bidirectional recurrent neural networks. In other hand, the study brought up the idea of investigating trainable combiner-based ensemble methods for PSSP, to see how much improvement they give rather than fixed rules. Finally, as the focus of the present study is to evaluate the performance of ensemble methods in PSSP, the use of other larger benchmark datasets to further improve the prediction accuracy is underway. Another interesting direction is to explore cluster ensembles in PSSP which seems to be a "hot-topic" in the area of machine learning. We hope our work will provide helpful information and lead to some novel ideas that may be considered in ensemble methods design and will encourage further investigations.

References

Anfinsen C (1973) Principles that govern the folding of protein chains. Science 181:223

Baumgartner D, Serpen G (2012) Global-local hybrid ensemble classifier for KDD 2004 cup particle physics dataset. Int J Mach Learn Comput 2(3):231–234

Bouziane H, Messabih B, Chouarfia A (2011) Profiles and majority voting-based ensemble method for protein secondary structure prediction. Evolut Bioinform 7:171–189

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Chen J, Chaudhari N (2006) Bidirectional segmented-memory recurrent neural network for protein secondary structure prediction. Soft Comput 10:315–324

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

Cuff J, Barton G (1999) Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins Struct Funct Genet 34(4):508–519

Didaci L, Fumera G, Roli F (2013) Diversity in classifier ensembles: fertile concept or dead end? Lecture Notes in Computer Science, vol 7872, pp 37–48

Dietterich T (2000) Ensemble methods in machine learning. Lecture Notes in Computer Science, vol 1857, pp 1–15

Dietterich T (1997) Machine-learning research: four current directions. AI Mag 18(4):97–136

Dietterich T (2002) Ensemble learning. In: Arbib MA (ed) The handbook of brain theory and neural networks, 2nd edn. Bradford Books, The MIT Press, Cambridge

Guermeur Y, Lifchitz A, Vert R (2004) Kernel methods in computational biology. MIT Press, Cambridge

Guermeur Y, Monfrini E (2011) A quadratic loss multi-class SVM for which a radius-margin bound applies. Informatica 22(1):73–96

Guermeur Y, Thomarat F (2011) Estimating the class posterior probabilities in protein secondary structure prediction. In: 6th IAPR international conference on pattern recognition in bioinformatics, pp 260–271

Hansen J (2000) Combining predictors: meta machine learning methods and bias/variance & ambiguity decompositions. PhD thesis, BRICS, Department of Computer Science, University of Aarhus, pp 1–191

Jiao T, Zong G, Zheng W (2013) New stability conditions for GRNs with neutral delay. Soft Comput 17:703–712

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

Kittler J, Hatef M, Duin R, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20:226–239

Kuncheva L, Bezdek J, Guin R (2001) Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognit 34(2):299–314

Kuncheva L (2001) Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognit 34:299–314

Kuncheva L (2005) Combining pattern classifiers. Wiley Press, New York

Kuncheva L, Whitaker C (2003) Measures of diversity in classifier ensembles and their relationship with ensemble accuracy. Mach Learn 51:181–207

Lauer F, Guermeur Y (2011) MSVMpack: a multi-class support vector machine package. J Mach Learn Res 12:2269–2272. http://www.loria.fr/lauer/MSVMpack

Lee Y, Lin Y, Wahba G (2004) Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. J Am Stat Assoc 99(465):67–81

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

Opitz D, Shavlik J (1996) Generating accurate and diverse members of a neural network ensemble. In: Touretzky DS, Mozer MC, Hasselmo ME (eds) Advances in neural information processing systems, vol 8. The MIT Press, Cambridge, pp 535–541

Ou Y, Oyang Y, Chen C (2005) A novel radial basis function network classifier with centers set by hierarchical clustering. In: International joint conference on neural networks (IJCNN), vol 1, pp 1383–1388

Pauling L, Corey R, Branson H (1951) The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. Natl Acad Sci USA 37(4):205–211

Platt J (2000) Probabilities for SV machines. In: Smola A, Bartlett P, Schölkopf B, Schuurmans D (eds) Advances in large margin classifiers, chapter 5. The MIT Press, Cambridge, pp 61–73

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

Rost B, Sander C (1993) Prediction of protein secondary structure at better than 70 % accuracy. J Mol Biol 232(2):584–599

Rost B, Sander C (1993) Prediction of secondary structure at better than 70 % accuracy. J Mol Biol 232:584–599

Rost B, Sander C (1994) Combining evolutionnary information and neural networks to predict protein secondary structure prediction. Proteins 19:55–72

Sander C, Schneider R (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 9:56–68

Schapire R, Freund Y (2012) Boosting: foundations and algorithms. MIT Press, Cambridge

Sewell M (2011) Ensemble learning. Research Note, pp 1–12

Shafer G (1976) A mathematical theory of evidence. Princeton University Press, New Jersey

Tuliakov S, Jaejer S, Govindaraju V, Doermann D (2008) Review of classifier combination methods, vol 90. Machine learning in document analysis and recognition. Springer, Berlin

Wallace B (2012) Class probability estimates are unreliable for imbalanced data (and How to Fix Them). In: 13th IEEE international conference on data mining, pp 695–704

Weston J, Watkins C (1998) Multi-class support vector machines. Tech. Rep. CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science

Whalen S, Pandey G (2013) A comparative analysis of ensemble classifiers: case studies in genomics. In: 13th IEEE international conference on data mining

Wolpert D (1992) Stacked generalisation. Neural Netw 5:241–259

Xu L, Krzyÿzak A, Suen C (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans Syst 22(3):418–435

Zemla A, Venclovas Č, Fidelis K, Rost B (1999) A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. Proteins Struct Funct Genet 34:220–223. http://proteinmodel.org/AS2TS/SOV/sov.html

Zhang Z, Jordan M (2006) Bayesian multicategory support vector machines. In: UAI'06, pp 552–559

Zong G, Liu J, Zhang Y, Hou L (2010) Delay-range-dependent exponential stability criteria and decay estimation for switched hopfield neural networks of neutral type. Nonlinear Anal Hybrid Syst 4(3):583–592