Vector Score Alpha Integration for Classifier Late Fusion

Gonzalo Safont, Addisson Salazar, Luis vergara

Institute of Telecommunications and Multimedia Applications,
Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

Abstract

Alpha integration is a family of integrators that encompasses many classic fusion operators (e.g., mean, product, minimum, maximum) as particular cases. This paper proposes vector score integration (VSI), a new alpha integration method for late fusion of multiple classifiers considering the joint effect of all the classes of the multi-class problem. Theoretical derivations to optimize the parameters of VSI for achieving the minimum probability of error are provided. VSI was applied to two classification tasks using electroencephalographic signals. The first task was the automatic stage classification of a neuropsychological test performed by epileptic subjects and the second one was the classification of sleep stages from apnea patients. Four single classifiers (linear and quadratic discriminant analysis, naive Bayes, and random forest) and three competitive fusion methods were estimated for comparison: mean, majority voting, and separated score integration (SSI). SSI is based on alpha integration, but unlike the proposed method, it considers the scores from each class in isolation, not accounting for possible dependencies among scores corresponding to different classes. VSI was able to optimally combine the results from all the single classifiers, in terms of accuracy and kappa coefficient, and outperformed the results of the other fusion methods in both applications.

1. Introduction

Recent advances in data acquisition and machine learning methods are paving the way for the optimal integration or fusion of complementary data modalities and/or classification methods in a wide variety of applications [1,2,3,4,5]. Data fusion is intended to exploit complementary properties of the results of several single modalities or classification methods, derived from different bases, in order to improve over their separate results. In addition, fusion can enable or enhance the approximation to more complex structured results (e.g., path trees and topological networks) [6,7]. This broad field of research has been named in different ways, such as: sensor data fusion; decision fusion; multimodal fusion; mixture of experts; and classifier combiners [8,9].

Particularly, the fusion of the scores of multiple classifiers is an interesting problem that has been increasingly studied as a suitable method for many complex problems, e.g., improving cardiovascular event prediction by combining genetic data and longitudinal health records [10] and classification of text documents [11]. The goal is to produce a new fused distribution in the score range for every class from the different score distributions, derived from different bases, of the single classifiers. The fusion of scores from multiple classifiers has been shown to improve classification performance, obtaining more stable and reliable results [8]. The fusion of scores is also known as late fusion, since it is made at the output of the classification process using the results before the decision of the single classifiers, i.e., the posterior probability (score) assigned to each class for the available testing records. Fusion can also be performed at the input of the classification process by combining the features estimated for classification (early fusion), or after the decision for each of the classifiers (late hard fusion).

There are many methods that have been proposed to perform late fusion, including functions whose parameters are learned to optimize a defined cost function under some criterion. This approach is followed in alpha integration, which was first proposed by Amari for integrating multiple stochastic models by minimizing their alpha divergence [12,13]. It has also been used to perform optimal integration of scores in binary classification (detection) problems [14]. Essentially, alpha integration is a family of integrators that encompasses many existing combinations as special cases of the alpha parameter. For instance, setting α=-1 would result in the average of the integrated measurements; α=1 would result in the product of the integrated measurements; and very high (low) values of α would result in the minimum (maximum) rule. The parameters of alpha integration can be learned by optimizing the least mean squared error (LMSE) or the minimum probability of error (MPE) criterion [14,15,16,17].

Recently, alpha integration was extended for integrating multi-class classifiers by considering the scores from each class in isolation in a method called separated score integration (SSI) [15]. Alpha integration was performed separately on the scores assigned to each class by all the classifiers. In this paper, we propose a new approach of alpha integration to late fusion of scores from multiple classifiers in a multi-class problem that we called vector score integration (VSI). In VSI, the joint effect of the scores of all classes of multiple classifiers is considered. Unlike SSI, the alpha parameters of VSI are applied jointly on all classes, and thus, the effect of the parameters is spread across all classes. We extend the minimum probability of error (MPE) criterion proposed in

[14] for binary classification to the multi-class classification problem. Thus, derivations to optimize the parameters of VSI for achieving MPE are provided.

The performance of VSI was tested on two applications using real electroencephalographic (EEG) signals. The first application consisted on classifying the samples of EEG signals from epileptic subjects while they were taking a neuropsychological visual memory test in three stages: stimulus display, retention interval, and subject response. The second application was the classification of EEG signals from apnea patients in three stages of sleep: wake, REM (rapid eye movement), and nREM (not REM). Four single classifiers were implemented: linear discriminant analysis (LDA), naive Bayes (NB), quadratic discriminant analysis (QDA), and random forests (RDF). Those methods were selected because of their performance and their widespread use in many applications. VSI was used to optimally combine the results from the single classifiers, improving classification performance. Besides, the results of VSI were compared with those of fusion using the mean, majority voting, and SSI.

The rest of this paper is organized as follows. Section 2 includes a review of the alpha integration method for binary classification, and Section 3 extends alpha integration to multi-class classification. Section 4 presents the results of the proposed method on two sets of real data. The paper is closed by the conclusions and future work.

2. Alpha integration for binary classification

Let us assume that a set of *D* detectors (binary classifiers) is available, each returning a different score *s*$_i$, *i=1…D*, for every input observation. We will assume that these scores are normalized between 0 and 1, with higher values denoting that the positive class is more likely than the negative class. Alpha integration is the weighted alpha mixture of these scores [12]:

$$s_\alpha([s_1 \ldots s_D]) = c\, h_\alpha^{-1}\left\{\sum\nolimits_{i=1}^{D} w_i\, h_\alpha(s_i)\right\} \tag{1}$$

$$h_\alpha(s_i) = \begin{cases} s_i^{(1-\alpha)/2} & ,\ \alpha \neq 1 \\ \log(s_i) & ,\ \alpha = 1 \end{cases} \tag{2}$$

where α and $\mathbf{w} = [w_1 \ldots w_D]^T$ are the parameters to be optimized, subject to $\sum\nolimits_{i=1}^{D} w_i = 1$, $w_i \geq 0$, and c is a normalization constant to ensure the result of alpha integration is a probability distribution. Alpha integration has been shown to be the optimal integration of the considered scores under alpha risk [13].

Most simple soft fusion functions can be obtained as particular selections of the parameters of alpha integration. For instance, α can be set to obtain the arithmetic mean $(\alpha = -1)$, the geometric mean $(\alpha = 1)$, and the harmonic mean $(\alpha = 3)$. Similarly, $\alpha = \infty$ $(-\infty)$ is equivalent to computing the minimum (maximum) of the scores. In general, however, the parameters of alpha integration are optimized to satisfy some criterion (e.g., the least mean square error [14,17]).

One such criterion is the minimization of the probability of error (MPE), which was introduced in [14]. Let us assume we have a set of couples $\{\mathbf{s}^j, y^j\}$, $j = 1 \ldots N$, where $\mathbf{s}^j = [s_1 \ldots s_D]^T$ is the vector of scores provided by the D detectors when y^j is the corresponding known binary decision ($y^j = 1$ for the positive class and $y^j = 0$ for the negative class). The minimization of the probability of error is equivalent to the maximization of the probability of obtaining correct decisions, P_c, through the whole set of couples $\{\mathbf{s}^j, y^j\}$, $j = 1 \ldots N$:

$$-\log P_c = -\sum\nolimits_{j=1}^{N} \left\{ y^j \log\left(s_\alpha\left(\mathbf{s}^j\right)\right) + \left(1 - y^j\right)\log\left(1 - s_\alpha\left(\mathbf{s}^j\right)\right) \right\} \tag{3}$$

The derivatives of (3) with respect to the parameters of alpha integration are

$$\frac{\partial(-\log P_c)}{\partial \alpha} = -\sum\nolimits_{j=1}^{N} \left(\frac{y^j}{s_\alpha\left(\mathbf{s}^j\right)} - \frac{1 - y^j}{1 - s_\alpha\left(\mathbf{s}^j\right)} \right) \frac{\partial s_\alpha\left(\mathbf{s}^j\right)}{\partial \alpha} \tag{4}$$

$$\frac{\partial(-\log P_c)}{\partial w_i} = -\sum\nolimits_{j=1}^{N} \left(\frac{y^j}{s_\alpha\left(\mathbf{s}^j\right)} - \frac{1 - y^j}{1 - s_\alpha\left(\mathbf{s}^j\right)} \right) \frac{\partial s_\alpha\left(\mathbf{s}^j\right)}{\partial w_i} \tag{5}$$

$$\frac{\partial s_\alpha\left(\mathbf{s}^j\right)}{\partial \alpha} = \frac{2 s_\alpha\left(\mathbf{s}^j\right)}{1 - \alpha} \left\{ \frac{\log\left(\sum\nolimits_{i=1}^{D} w_i\, h_\alpha\left(s_i^j\right)\right)}{1 - \alpha} + \frac{\sum\nolimits_{i=1}^{D} w_i \frac{\partial h_\alpha\left(s_i^j\right)}{\partial \alpha}}{\sum\nolimits_{i=1}^{D} w_i\, h_\alpha\left(s_i^j\right)} \right\} \tag{6}$$

$$\frac{\partial s_\alpha\left(\mathbf{s}^j\right)}{\partial w_i} = \begin{cases} \dfrac{2}{1 - \alpha} \dfrac{s_\alpha\left(\mathbf{s}^j\right) h_\alpha\left(s_i^j\right)}{\sum\nolimits_{l=1}^{D} w_l\, h_\alpha\left(s_l^j\right)} & ,\ \alpha \neq 1 \\ s_\alpha\left(\mathbf{s}^j\right) \log\left(s_i^j\right) & ,\ \alpha = 1 \end{cases} \tag{7}$$

And the derivative of the alpha representation h_α is $\frac{\partial h_\alpha\left(s_i^j\right)}{\partial \alpha} = -\frac{1}{2}\log\left(s_i^j\right)\left(s_i^j\right)^{(1-\alpha)/2}$. Equations (4) to (7) can be used to optimize the parameters of the model, for instance, using gradient descent.

3. Vector score integration (VSI)

In the following, we propose a vector score integration (VSI) method that generalizes alpha integration to multi-class classification $(K \geq 2)$ and accounts for cross dependencies among scores from different classes. We also present a learning algorithm to optimize the parameters of VSI with respect to the minimum probability of error criterion.

Let us assume we have a set of scores from D classifiers working on a classification problem with K classes. Each classifier with produce a vector of scores for each class, $\mathbf{s}_i = [s_{1i} \ldots s_{Ki}]^T$, $i = 1 \ldots D$, which are normalized to unit sum, $\sum\nolimits_{k=1}^{K} s_{ki} = 1$. All the scores are joined in matrix $\mathbf{S} = [\mathbf{s}_1 ... \mathbf{s}_D]$. The *m*th row of this matrix is denoted by $\mathbf{r}_m = [s_{m1} ... s_{mD}]$. The true class is denoted by a class identifier vector, $\mathbf{y} = [y_1 ... y_K]^T$, where

$$y_k = \begin{cases} 1 \text{ if the true class is } k \\ 0 \text{ otherwise} \end{cases} \qquad (8)$$

We can obtain a vector of integrated scores for each class, $\mathbf{s}_{\alpha_k} = [s_{\alpha_k 1} ... s_{\alpha_k K}]^T$, using alpha integration (1):

$$s_{\alpha_k m}\left(\mathbf{r}_m\right) = \begin{cases} \left(\sum_{i=1}^{D} w_{ki} \left(s_{mi}\right)^{(1-\alpha)/2}\right)^{2/(1-\alpha)} & , \alpha \neq 1 \\ \exp\left(\sum_{i=1}^{D} w_{ki} \log\left(s_{mi}\right)\right) & , \alpha = 1 \end{cases} , m = 1 ... K \qquad (9)$$

In VSI, we have *K* sets of alpha integration parameters (α_k and weights $\mathbf{w}_k = [w_{k1} ... w_{kD}]^T$) that are applied on the scores for each of the *K* classes, resulting in a $[K \times K]$ matrix. Once we have the vectors of integrated scores s_{α_k}, $k = 1 ... K$, classification is performed by choosing the vector that is closest to an ideal output $\mathbf{y}_{(k)}$ (1 in class *k* and 0 otherwise). In this work, we considered the Euclidean distance, thus arriving to

$$\hat{k} = \min_k \left\| \mathbf{y}_{(k)} - \mathbf{s}_{\alpha_k} \right\|^2 \qquad (10)$$

With $\mathbf{y}_{(k)}$ denoting a class identifier vector (8) whose true class is *k*. The fused scores provided by the method are those corresponding to the chosen class, $\mathbf{s}_{\alpha_k}$.

The differences between the proposed VSI and SSI [15] can be understood by comparing the differences between the alpha integration function for VSI, see equation (9), and that of SSI:

$$s_{\alpha_k}\left(\mathbf{r}_k\right) = \begin{cases} \left(\sum_{i=1}^{D} w_{ki} \left(s_{ki}\right)^{(1-\alpha)/2}\right)^{2/(1-\alpha)} & , \alpha \neq 1 \\ \exp\left(\sum_{i=1}^{D} w_{ki} \log\left(s_{ki}\right)\right) & , \alpha = 1 \end{cases} , k = 1 ... K \qquad (11)$$

Note the differences in the subindices of SSI (11) and VSI (9). Both methods have the same parameters, α_k and w_{ki}, $k = 1 ... K$, $i = 1 ... D$. However, in SSI, the alpha parameters of the kth class are applied on the scores for that same *k*th class, $k = 1 ... K$, resulting in *K* fused scores. In VSI, the alpha parameters of each class are applied on the scores for all classes, resulting in *K* fused scores per class, for a grand total of K^2 fused scores. In SSI, the scores provided by the classifiers are integrated separately for each class. This simplifies the optimization procedure, but it also means that possible dependencies between the scores assigned to different classes are not taken into account. Conversely, cross dependencies are considered in VSI since the alpha parameters are applied jointly on all classes.

The optimization of the alpha integration parameters of class *k* will be performed using the subset of the whole training set where the true class is *k*. We denote this subset by $\left\{\mathbf{S}_{(k)}^{j}, \mathbf{y}_{(k)}^{j}\right\}$, $j = 1 ... N_k$, where N_k is the number of training couples in the subset. As per the definition of $\mathbf{y}$ in (8), since all values in this subset belong to class *k*, $y_{(k)k}^{j} = 1$ and $y_{(k)m}^{j} = 0$, $m \neq k$.

Given these definitions, the MPE cost function for class *k* is:

$$\begin{aligned} -\log P_{ck} = -\sum_{j=1}^{N_k} \sum_{m=1}^{K} \Big\{ & y_{(k)m}^{j} \log\left(s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)\right) + \\ & \left(1 - y_{(k)m}^{j}\right) \log\left(1 - s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)\right) \Big\} \end{aligned} \qquad (12)$$

The derivatives of (12) with respect to the parameters of class *k* are:

$$\begin{aligned} \frac{\partial(-\log P_{ck})}{\partial \alpha_k} = -\sum_{j=1}^{N_k} \sum_{m=1}^{K} \Bigg(& \frac{y_{(k)m}^{j}}{s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)} - \\ & \frac{1 - y_{(k)m}^{j}}{1 - s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)} \Bigg) \frac{\partial s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)}{\partial \alpha_k} \end{aligned} \qquad (13)$$

$$\begin{aligned} \frac{\partial(-\log P_{ck})}{\partial w_{ki}} = -\sum_{j=1}^{N_k} \Bigg(& \frac{y_{(k)m}^{j}}{s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)} - \\ & \frac{1 - y_{(k)m}^{j}}{1 - s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)} \Bigg) \frac{\partial s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)}{\partial w_{ki}} \end{aligned} \qquad (14)$$

where

$$\begin{aligned} \frac{\partial s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)}{\partial \alpha_k} = \frac{2 s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)}{1 - \alpha_k} \Bigg\{ & \frac{\log\left(\sum_{i=1}^{D} w_{ki} h_{\alpha_k}\left(s_{mi(k)}^{j}\right)\right)}{1 - \alpha_k} + \\ & \frac{\sum_{i=1}^{D} w_i \frac{\partial h_{\alpha_k}\left(s_{mi(k)}^{j}\right)}{\partial \alpha_k}}{\sum_{i=1}^{D} w_i h_{\alpha_k}\left(s_{mi(k)}^{j}\right)} \Bigg\} \end{aligned} \qquad (15)$$

$$\frac{\partial h_{\alpha_k}\left(s_{mi(k)}^{j}\right)}{\partial \alpha_k} = -\frac{1}{2} \log\left(s_{mi(k)}^{j}\right) \left(s_{mi(k)}^{j}\right)^{(1-\alpha_k)/2} \qquad (16)$$

And the derivatives with respect to the weights w_{ki} are

$$\frac{\partial s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right)}{\partial w_{ki}} = \begin{cases} \dfrac{2}{1 - \alpha_k} \dfrac{s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right) h_{\alpha_k m}\left(s_{mi(k)}^{j}\right)}{\sum_{l=1}^{D} w_{kl} h_{\alpha_k m}\left(s_{ml(k)}^{j}\right)} & , \alpha \neq 1 \\ s_{\alpha_k m}\left(\mathbf{r}_{m(k)}^{j}\right) \log\left(s_{mi(k)}^{j}\right) & , \alpha = 1 \end{cases} \qquad (17)$$

Using these derivatives, we can estimate the parameters that optimize the MPE criterion, for instance, with a gradient descent algorithm.

4. Experiments on real data

4.1. Experiment on EEG data from epileptic subjects

The proposed multi-class alpha integration method was tested on a set of real EEG data of four epileptic patients undergoing a neuropsychological test. The tests were carried out in a clinical environment to evaluate the learning and short-term memory capabilities of the patients. The EEG signals was captured on 18 bipolar EEG channels set according to the 10-20 system (see Figure 1), sampled at 500 Hz and band-pass filtered between 0.5 and 30 Hz with the help of the Neurology and Neurophysiology Units at Hospital Universitari i Politècnic La Fe, Valencia (Spain).

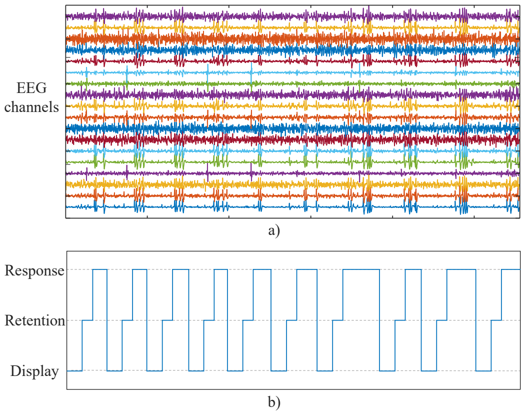


Figure 1. Example of the data captured for one of the subjects.

The implemented neuropsychological test was the Barcelona test (BT, [18]), a visual short-term memory task. During each trial of the BT, the subject is shown a probe item for 10 seconds, and after a 10-second retention interval, they attempt to recognize the probe item among a set of four similar items. The BT contains 10 trials that become progressively harder, and scoring is determined by the total number of correct responses. Each trial of the BT was divided in three stages: stimulus display (SD), retention interval (RI), and subject response (SR) corresponding to the 3 classes for classification. The problem was to assign one of those classes to every sample of the part of the EEG signals used for testing.

In order to perform classification, the following features were extracted from each EEG signal window (epoch) of 0.25 seconds: average, mean absolute value, centroid frequency, and power in the following frequency bands: delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-13 Hz) and beta (13-30 Hz). As was commented above, four single classifiers were implemented: LDA, QDA, NB, and RDF. The scores returned by the single classifiers were fused using majority vote (late hard fusion), mean, SSI, and the proposed VSI. First, the classification procedure split the epochs equally into three datasets: training, validation, and testing. In order to preserve the prior probabilities, the observations of each class were randomly distributed as evenly as possible across the three datasets. The single classifiers were trained using the training dataset, and both alpha integration methods were trained using the scores obtained by the single classifiers on the validation dataset. The parameters of alpha integration were optimized with respect to the MPE criterion using an interior point method (IPM, [19]) for constrained optimization of the cost function. The derivatives developed in Section 3 were used for VSI. Finally, all classifiers were compared by their performance on the testing dataset. The results for each method were obtained as the average of 100 iterations.

An example of the obtained classification for one of the subjects in shown in Figure 2. VSI returned the result closest to the ground truth, even better than the result of SSI. Conversely, classical fusion methods returned worse results than alpha integration and, in this case, than some single classifiers. For instance, the first trial (seconds 0 to 10 in Figure 2) was incorrectly classified by LDA, NB, QDA, and the fusions; RDF and SSI yielded a more accurate classification; and VSI achieved a result that was almost identical to the ground truth.

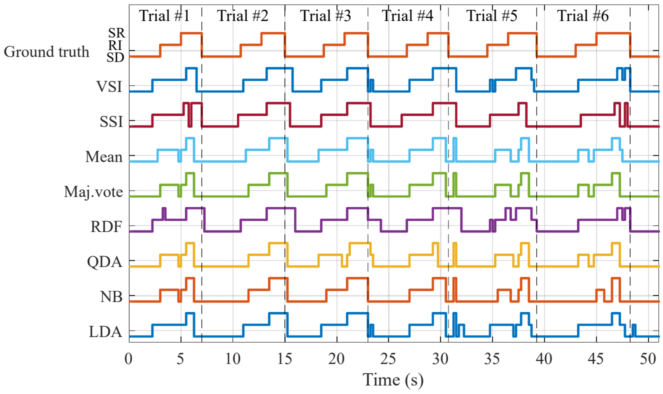


Figure 2. Classification returned by all methods for one of the subjects.

The average results of the experiment are shown in Figure 3. We considered two performance indices, the accuracy (Figure 3.a) and Cohen's kappa coefficient (Figure 3.b), the latter being more robust with respect to the different prior probabilities between classes. In accordance with the results of Figure 2, the best result was yielded by VSI for both indicators, and SSI yielded the second-best result. Two single classifiers, LDA and RDF, yielded better results than the considered classical fusion techniques. These results show that classical fusion techniques were unable to improve the results of the single classifiers, whereas the fusion returned by alpha integration was able to optimally combine all four single classifiers. Furthermore, the increased flexibility of the proposed VSI method yielded an even better combination than the less-flexible SSI. Numerically, VSI achieved an average 7.63% more accuracy and 10.69% more kappa than the best performing classical fusion (mean); 5.11% more accuracy and 8.36% more kappa than the best performing single classifier (RDF); and 1.47% more accuracy and 3.26% more kappa than SSI. Moreover, the results yielded by VSI were also more stable than those yielded by other methods, as seen by the smaller standard deviation bars in Figure 3. The standard deviation of VSI was the smallest, followed by those of SSI and two single classifiers, LDA and QDA. The standard deviation of VSI was 0.65% for accuracy and 0.97% for kappa; conversely, the standard deviation of the best performing classical fusion (mean) was 1.14% for accuracy and 1.68% for kappa; and the standard deviation of the best performing single classifier (RDF) was 1.03% for accuracy and 1.65% for kappa.

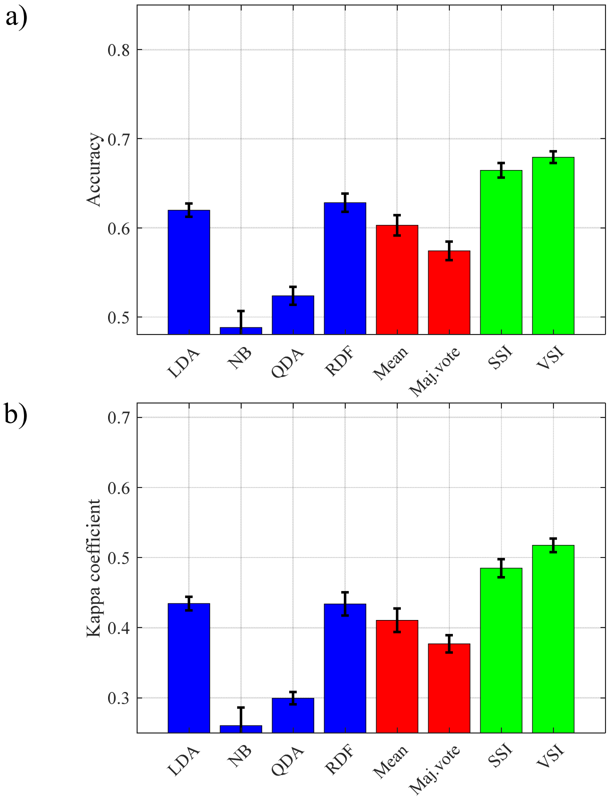


Figure 3. Average and standard deviation results of the experiment: a) accuracy; b) kappa coefficient.

4.2. Experiment on EEG data from patients with apnea

To further verify the performance of the proposed vector score integration method, a second experiment was performed on a publicly available dataset of real polysomnograms (PSG) from the St Vincent's University Hospital / University College Dublin Sleep Apnea Database in Physionet [20]. The database contains PSG from 25 adult subjects (21 male, 4 female) with suspected apnea, taken during a night of sleep. The PSG is a multimodal biomedical record that includes many kinds of physiological signals, but in this work, we considered the two available bipolar electroencephalographic channels: C3-A2 and C4-A1. The EEG signals were sampled at 128 Hz and band-pass filtered between 0.5 and 30 Hz. The data provided 30-second epoch split and each epoch was labeled by an expert into one of seven classes: wake, rapid eye movement (REM) sleep, sleep stages 1 through 4, artifacts, and indeterminate. For this experiment, three classes were considered: wake, REM sleep, and non-REM sleep (sleep stages 1 through 4). Samples belonging to artifacts and indeterminate classes were removed prior to the experiment. An example of the available signals is shown in Figure 4. The epochs corresponding to artifacts and indeterminate were eliminated.

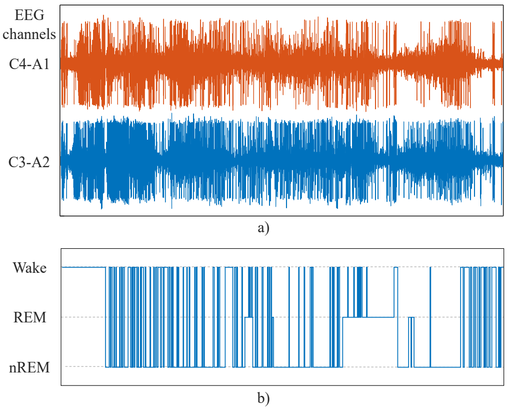


Figure 4. Example of the EEG data (a) and the classes (b) from one of the subjects.

In order to perform classification, the following features were extracted from each EEG signal in 30 second epochs: power in the delta (0-4 Hz), theta (5-7 Hz), alpha (8-12 Hz), sigma (13-15 Hz) and beta (16-30 Hz) frequency bands; and the activity, mobility and complexity of the signal [21]. These features are typically used in the literature on sleep staging [22]. We considered the same classification methods used for the experiment in Section 4.1. Each subject was classified independently from the rest. Similar to the previous experiment, the data were split into three datasets: training, validation and testing. In order to preserve the prior probabilities, the observations of each class were randomly distributed as evenly as possible across the three datasets. For some subjects, however, this meant that the single classifiers were trained using less observations than variables, which led to stability issues. For such subjects, the class containing insufficient observations was simply eliminated from the data. The considered single classifiers were trained on the training dataset, the proposed alpha integration methods were trained on the scores of the single classifiers on the validation dataset, and the performance of all methods was estimated on the testing dataset. The results for each subject were obtained as the average of 100 iterations.

An example of the classification obtained for one of the patients is shown in Figure 5. It can be seen that VSI yielded classes that more closely resembled the ones provided by the expert, particularly at the beginning and the end of the window shown in Figure 5 (00:28 to 02:11AM). The labels yielded by VSI tended to oscillate less than those provided by the other considered methods (including SSI alpha integration) resulting in less false alarms. For instance, VSI was the only method without false sleeping periods close to 00:35AM or false wake periods right before 2:00AM.

The average accuracy and kappa values for all 25 patients are shown in Figure 6, where the vertical lines displaying the standard error of the result. These values are similar to those in the literature for this dataset, e.g., [23,24]. Results are largely in accordance with those of the previous experiment (see Figure 3), albeit with an overall better performance, owing to the greater number of subjects. The best result was yielded by VSI, followed by SSI. Furthermore, the considered classical fusion techniques were consistently unable to outperform the best-performing single classifier (RDF). Numerically, VSI achieved an average 1.39% more accuracy and 2.12% more kappa than the best performing classical fusion (majority vote); 1.28% more accuracy and 2.13% more kappa than the best performing single classifier (RDF); and 0.14% more accuracy and 1.53% more kappa than SSI. Moreover, the results yielded by VSI were also more stable than those yielded by other methods, as seen by the smaller standard deviation bars in Figure 6. The standard deviation of VSI was the smallest, followed by those of SSI, majority vote, and RDF. The standard deviation of VSI was 0.61% for accuracy and 1.14% for kappa; conversely, the standard deviation of the best performing classical fusion (majority vote) was 0.66% for accuracy and 1.20% for kappa; and the standard deviation of the best performing single classifier (RDF) was 0.66% for accuracy and 1.22% for kappa.

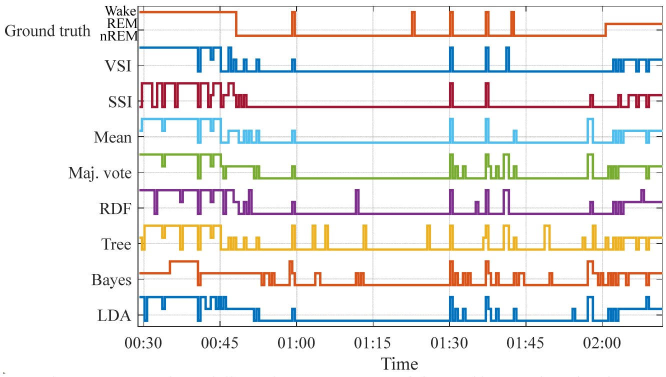
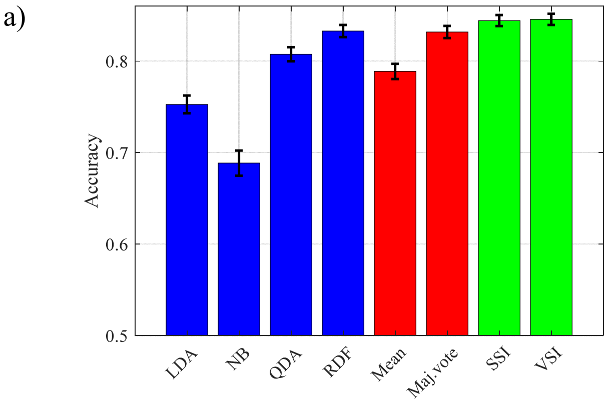


Figure 5. Classification returned by all methods for one of the subjects.



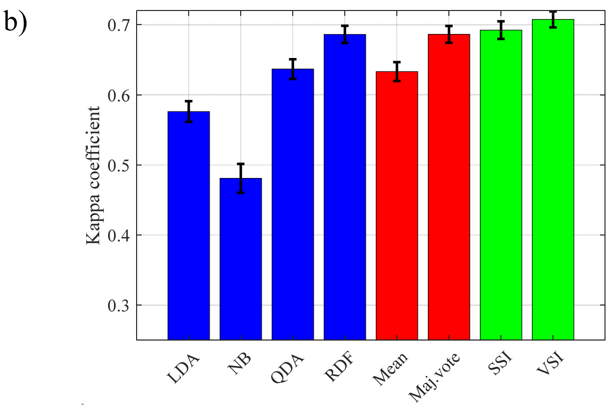


Figure 6. Average and standard deviation results of the experiment: a) accuracy; b) kappa coefficient.

5. Discussion and conclusions

The results show the proposed method VSI overcome the performance of all the other methods. The most competitive method was SSI, which is also based on alpha integration. The differences between VSI and SSI were 1.47% and 3.26% (Figure 3, neuropsychological test staging) and 0.14% and 1.53% (Figure 6, sleep apnea detection) for precision and kappa, respectively. Cohen's Kappa coefficient has shown to be more sensitive to differences in classification, given that it considers a priori probabilities of the classes. Both applications had unbalanced a priori probabilities for three-class classification, i.e., (41.22%, 22.34%, 36.43%) and (22.67%, 14.52%, 62.81%) for neuropsychological test staging and sleep apnea detection, respectively. This unbalance increased the difficulty of classification; it could be alleviated by augmentation of the sample size using replicates or surrogate samples estimated from the original data [25,26].

In addition, cross dependencies between a posteriori probabilities provided by the single classifiers for the different classes can affect cost function optimization for fusion-based methods, and thus affect the performance difference between them. From a practical standpoint, relatively small differences in classification might be important when diagnosing the patient's condition (e.g., memory and learning capabilities and sleep disorder degree) and therefore the clinical treatment to be followed.

Notice that in both experiments we have considered the fusion of 4 different classifiers, namely, LDA, NB, QDA and RDF. This covers a reasonable number of representative classifiers, although more classifiers could be added to the fuser. Predicting the optimum number D of classifiers in a particular experiment is an interesting but very complex problem. Assuming we had knowledge of the separate performance of every classifier, we would also need to define some type of multivariate statistical dependence model of the whole set of classifiers (much the same as it is done in [27] for the hard fusion of dependent detectors). Given that dependence model, we should consider the nonlinear soft fusion implicit in alpha integration in an effort to predict the fuser performance dependence on D. This is hardly approachable. Moreover, the defined dependence model is to be estimated from training data so, ultimately, a more practical approach is to successively incorporate a new classifier and testing if the performance improves or not. In our experiments, we have verified that VSI fusing 4 classifiers improves not only with respect to every single classifier (as shown in Figures 3 and 6), but also with respect to VSI fusing only two or three classifiers. Notice that, unlike mean and majority voting, VSI implements optimum fusion. Thus, for example, if the new classifier is poor, the method learns to give it little relevance. One expected result of this optimization is that incorporating a new classifier will never worsen the performance. Furthermore, the amount of possible improvement would depend on the statistical dependence with the rest of classifiers.

Finally, the conclusions of this work are as follows. The performance of VSI has been tested on two sets of real biomedical data. The first set consisted of electroencephalographic data from four epileptic subjects that were performing a neuropsychological visual memory test. The data were classified into the three stages of the test (display, retention, and response). Four single classifiers were considered: linear discriminant analysis, naive Bayes, quadratic discriminant analysis, and random forests. The single classifiers were combined using two classical fusion techniques (majority vote and score mean), separated score integration (SSI), and the proposed VSI method. The second set of biomedical data consisted of a public database of polysomnographic data from subject with suspected apnea. These data were classified into three sleep stages (wake, REM sleep, and non-REM sleep) using the same methods as for the first set. The results showed that both problems were difficult and classical fusion techniques were unable to improve the results over those of the best single classifier. Conversely, VSI was able to combine the scores from all classifiers and return an improved combined score that resulted in better accuracy and kappa coefficient in both experiments. These results demonstrate the capability of the proposed method to exploit the scores to improve performance in cases where dependencies are complex.

Notice that, ultimately, we face a problem of computing a posterior probability (score) from a multidimensional random variable: a matrix formed by the $K \times D$ scores to be fused. A Bayesian approach to the problem implies modelling the class conditional probabilities of the multidimensional random variable. In SSI, this is faced by assuming that the multidimensional probability density (MPD) conditioned to the kth class only depends on the kth row elements of the matrix (scores provided by every single classifier). Hence, the scores are separately integrated for every class. In VSI, however, the class conditioned MPD is assumed to depend on all the elements of the matrix by columnwise integration of the vector scores provided by every single classifier. This may explain the improved results of VSI. Certainly, columnwise integration is not the most general option to fuse all the scores; a more general alpha integration could apply an individual coefficient to every element of the matrix, but this would dramatically increase the number of parameters to be estimated. Thus, VSI is a good compromise between general modelling and computational burden.

Acknowledgments

This work was supported by the Spanish Administration and European Union under grant TEC2017-84743-P.

References

1. T. Baltrusaitis, C. Ahuja, and L.P. Morency, "Multimodal machine learning: a survey and taxonomy," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 423-443, 2019.
2. A. Jouirou, A. Baâzaoui, and W. Barhoumi, "Multi-view information fusion in mammograms: a comprehensive overview," Information Fusion, vol. 52, pp. 308-321, 2019.
3. M. Danelljan, G. Bhat, S. Gladh, F.S. Khan, M. Felsberg, "Deep motion and appearance cues for visual tracking," Pattern Recognition Letters, vol. 124, pp. 74-81, 2019.
4. F. Ringeval, F. Eyben, E. Kroupi, A. Yuce, J.-P. Thiran, T. Ebrahimi, D. Lalanne, B. Schuller, "Prediction of asynchronous dimensional emotion ratings from audiovisual and physiological data," Pattern Recognition Letters, vol. 66, pp. 22-30, 2015.
5. M. Sturari, D. Liciotti, R. Pierdicca, E. Frontoni, A. Mancini, M. Contigiani, P. Zingaretti, "Robust and affordable retail customer profiling by vision and radio beacon sensor fusion," Pattern Recognition Letters, vol. 81, pp. 30-40, 2016.
6. S. Thoma, A. Thalhammer, A. Harth, and R. Studer, "FusE: entity-centric data fusion on linked data," ACM Transactions on the Web, vol. 13, no. 2, pp. 8:1-36, 2019.
7. J. Park, D. Ahn, and J. Lee, "Development of data fusion method based on topological relationships using indoorGML core module," Journal of Sensors, vol. 2018, no. Article no. 4094235, pp. 1-15, 2018.
8. M. Mohandes, M. Deriche, and S.O. Aliyu, "Classifiers combination techniques: a comprehensive review," IEEE Access, vol. 6, pp. 19626-19639, 2018.
9. O. Sagi and L. Rokach, "Ensemble learning: a survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, pp. 1-18, 2018.

10. J. Zhao and et al., "Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction," Scientific Reports, vol. 9, no. Article no. 717, pp. 1-10, 2019.
11. S.N.Bharath-Bhushan and A. Danti, "Classification of text documents based on score level fusion approach," Pattern Recognition Letters, vol. 94, pp. 118-126, 2017.
12. S. Amari, "Integration of stochastic models by minimizing α-divergence," Neural Computation, vol. 19, no. 10, pp. 2796-2780, 2007.
13. S. Amari, Information Geometry and its Applications. Tokyo (Japan): Springer, 2016.
14. A. Soriano, L. Vergara, B. Ahmed, and A. Salazar, "Fusion of scores in a detection context based on alpha integration," Neural Computation, vol. 27, no. 9, pp. 1983-2010, 2015.
15. G. Safont, A. Salazar, and L. Vergara, "Multiclass alpha integration of scores from multiple classifiers," Neural Computation, vol. 31, no. 4, pp. 806-825, 2019.
16. H. Choi, S. Choi, A. Katake, and Y. Choe, "Learning α-integration with partially-labeled data," in Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), Dallas, TX (USA), 2010, pp. 2058-2061.
17. H. Choi, S. Choi, and Y. Choe, "Parameter learning for alpha integration," Neural Computation, vol. 25, no. 6, pp. 1585-1604, 2013.
18. M. Quintana et al., "Spanish multicenter normative studies (Neuronorma project): norms for the abbreviated Barcelona Test," Archives of Clinical Neuropsychology, vol. 26, no. 2, pp. 144-157, 2010.
19. J. Nocedal and S.J. Wright, Numerical Optimization. New York, NY (USA): Springer-Verlag, 2006.
20. C. Heneghan. (2011) Physionet. [Online]. https://www.physionet.org/pn3/ucddb/
21. J.M. Hjorth, "The physical significance of time domain descriptors in EEG analysis," Electroencephalography and Clinical Neurophysiology, vol. 34, no. 3, pp. 321-325, 1973.
22. S. Motamedi-Fakhr, M. Moshrefi-Torbati, M. Hill, C.M. Hill, and P.R. White, "Signal processing techniques applied to human sleep EEG signals – a review," Biomedical Signal Processing and Control, vol. 10, pp. 21-33, 2014.
23. B. Xie and H. Minn, "Real-time Sleep Apnea Detection by Classifier Combination," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 3, pp. 469-477, 2012.
24. S. Wang, G. Hua, G. Hao, and C. Xie, "A Cycle Deep Belief Network Model for Multivariate Time Series Classification," Mathematical Problems in Engineering, vol. 2017, pp. 1-7, 2017.
25. A. Salazar, G. Safont, and L. Vergara, "Semi-supervised learning for imbalanced classification of credit card transaction," in Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN 2018, Article no. 8489755, 2018, pp. 4976-4982.
26. A. Salazar, G. Safont, and L. Vergara, "Surrogate techniques for testing fraud detection algorithms in credit card operations," in Proceedings of the 48th IEEE International Carnahan Conference on Security Technology ICCST 2014, Article no. 6986987, 2014, pp. 124-129.
27. L. Vergara, A. Soriano, G. Safont, and A. Salazar, "On the fusion of non-independent detectors," Digital Signal Processing, vol. 50, pp. 24-33, 2016.