Sparse Graph-based Representations of SSVEP Responses Under the Variational Bayesian Framework

Vangelis P. Oikonomou*,[1], Spiros Nikolopoulos[1] and Ioannis Kompatsiaris[1]

2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE) | 978-1-6654-4261-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBE52308.2021.9635427

***Abstract*— The recognition of Steady State Visual Evoked Potentials (SSVEP) constitutes a challenging problem in Brain Computer Interfaces (BCI), especially when the number of EEG sensors is limited. In this work, we propose a new sparse representation classification scheme that extends current schemes by exploiting the graph properties of relevant features. Based on this scheme each test signal is represented as a linear combination of train signals. Our expectation is that this constrained linear combination, exploiting the graph's structure of the training data, will lead to representations that are more robust. Moreover, in order to avoid overfitting and provide a model with good generalization abilities we adopt the bayesian framework and, in particular, the Variational Bayesian Framework since we use a specific prior distribution to exploit the graph structure of the data. The proposed algorithm has been evaluated on two SSVEP datasets achieving state-of- the-art performance against well known classification methods in SSVEP literature.**

***Index Terms*— Sparse Representations, Steady State Visual Evoked Potentials, Eigenbrains, Brain Computer Interfaces**

I. Introduction

A Brain Computer Interface (BCI) system allows the connection between the human brain and the computer, and hence it provides a communication channel for people with motor disabilities [1], [2] and an alternative communication medium for healthy individuals. In most BCI systems, the brain activity is measured by electroencephalograms (EEG) using simple and inexpensive equipment. EEG-based BCI systems which utilize Steady State Visual Evoked Potentials (SSVEPs) have attracted the attention of many researchers due to their lower training requirements for the end-user and higher information transfer rates [3], [4]. A SSVEP is the brain response, evoked in occipital and occipital - parietal areas of the brain, by a visual stimulus flashing at a fixed frequency [5]. SSVEP responses normally include the fundamental frequency of the visual stimulus as well as its harmonics. SSVEP BCI systems detect the different frequency components corresponding to the visual stimuli and translate them into commands. The detection of SSVEP responses is achieved by using an EEG-based pattern recognition algorithm.

* Corresponding author.

[1]V. P. Oikonomou, Spiros Nikolopoulos and I. Kompatsiaris are with the Information Technologies Institute, Centre for Research and Technology Hellas, CERTH-ITI, 6th km Charilaou-Thermi Road, 57001 Thermi-Thessaloniki, Greece. {viknmu, nikolopo, ikom}@iti.gr

This work was supported by the NeuroMkt project, co-financed by the European Regional Development Fund of the EU and Greek National Funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under RESEARCH CREATE INNOVATE (T2EDK-03661).

Due to the frequency characteristics of SSVEPs, methods based on Power Spectrum Density Analysis (PSDA) such as fast Fourier transform (FFT) were widely used for frequency detection [6]. Usually, the frequency with the maximal PSD value is detected as the target frequency. However, PSDA approaches are sensitive to noise, and, they require a relatively large time window [7], [8]. To overcome the above shortcomings of PSDA, several approaches based on spatial filtering have been proposed. In [8] the Minimum Energy Combination (MEC) method has been proposed while in [7] the Canonical Correlation Analysis (CCA) method was introduced. Both methods use reference templates using sinusoids waves and solve an optimization problem, based on multi-channel SSVEP data, in order to obtain the optimal spatial filters. Finally, extensions of CCA have been proposed in [9]–[11], where data-driven approaches for the construction of templates are used.

We can observe here that all aforementioned methods are using templates, either by using sinusoids waves or by constructing the templates using the EEG data during a training stage. However, methods that are template - free have been also proposed in the literature. More specifically, in [12] Support Vector Machines (SVMs) and the Linear Discriminant Analysis (LDA) have been used to detect SSVEPs. In [13], [14] Convolutional Neural Networks (CNN) based on time frequency analysis are used to discriminate between SSVEP responses. In addition, in [15] the use of Multivariate Linear Regression (MLR) was proposed to learn discriminative features for improving SSVEP classification, while, in [16] kernel - based extensions of MLR were proposed using SSVEP-related kernels as an integral part of the Sparse Bayesian Learning (SBL) framework.

Methods such as the *typical* PSDA, CCA and MEC do not use training data, but they use a (mathematical or theoretical) model (or a template) to describe SSVEP responses and examine how close to this SSVEP model are the current observations (or EEG data). It is clear that the above methods make strong assumptions about the underlying structure that governs SSVEP responses. On the contrary SSVEP recognition systems based on SVM [12], the MLR approach [15], the SBL approach [16] and on convolutional neural networks [13] need EEG data for the training phase of the overall pattern recognition system. These techniques do not make any assumption about the structure of SSVEP responses. In addition, these techniques are not severely affected by the number of EEG channels which make them suitable for cases where a limited number of EEG channels is available.

In the proposed work, we provide a new classification

algorithm for the recognition of SSVEP responses. More specifically, Sparse Graph-based Representations are used for the discrimination of SSVEP responses. By adopting the bayesian framework we are able to introduce the graph properties of the EEG data, through the prior distribution, into the inference/learning procedure. In addition to the above our algorithm makes use of the eigebrains [17]. The remainder of this paper is organized as follows. In Section II we describe the proposed algorithm. The application of the proposed methodology in two SSVEP datasets and the obtained experimental results are presented in Section III. Finally, the concluding remarks and directions for future work are provided in Section VI.

II. Methodology

A. EigenBrain

In a SSVEP experiment the subject is seated in front of the stimulator (most probably a computer screen), where visual stimuli are delivered. During the experiment raw EEG data are collected in order to calibrate the overall system. Also, after the segmentation of raw EEG data (using event triggers), we obtain a number of trials for each visual stimulus (or class). These EEG trials are used for the calibration of the BCI system (for example training the classifier).

Let us assume that the SSVEP dataset is a collection of EEG trials $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_m\}$. Each $\mathcal{S}_i, i = 1, \cdots, m$ is a matrix of $N_{ch} \times N_t$, where N_{ch} is the number of channels and N_t the number of samples. We can also vectorize the above matrices to obtain $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_m\}$, where each $\mathbf{s}_i, i = 1, \ldots, m$ is a vector with $N = N_{ch} \times N_t$ elements. Due to the nature of SSVEP data, usually, we have $N >> m$ since the number of trials is relative small, making the processing significantly difficult. In order to reduce the dimension of the data we utilize the PCA method, resulting into the transformed (or features') dataset $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_m\}$, where $\mathbf{f}_i, i = 1, \ldots, m$ is a vector with $q (< m < N)$ elements. The new feature vectors, $\mathbf{f}_i$, are defined by the following linear transformation $\mathbf{f}_i = \mathbf{B}^T \mathbf{s}_i$, $\mathbf{B} : N \times q$ is the eigenvector matrix of PCA. Due to its connections to the PCA and EigenFaces [18] these (eigen)vectors in matrix $\mathbf{B}$ are called eigenbrains [17]. It must be noted here that eigenbrains can be constructed under varied conditions and modalities [17]. In our study these eigenbrains are referred to the raw EEG data.

B. Basic SRC scheme

Let C be the number of SSVEP classes and p_c the number of training EEG trials of class c. The i-th trial from class c is represented by an eigenbrain (or a feature vector), $\mathbf{f}_i^c \in \Re^q, i = 1, \cdots, p_c$. Stacking all eigenbrains from the same class into a matrix we obtain a class specific model:

$$\mathbf{X}_c = \left[\mathbf{f}_1^c, \mathbf{f}_2^c, \cdots, \mathbf{f}_{p_c}^c\right] \in \Re^{q \times p_c} \tag{1}$$

A basic assumption of our approach is that a test eigenbrain $\mathbf{y} \in \Re^q$ of the same class will approximately lie in the linear subspace spanning from the training eigenbrains:

$$\mathbf{y} = \mathbf{X}_c \boldsymbol{\alpha}_c \tag{2}$$

where $\boldsymbol{\alpha}_c \in \Re^{p_c}$ is a coefficient vector describing the participation of each eigenbrain to the procedure. In the beginning, the label of the test eigenbrain $\mathbf{y}$ is unknown hence we can represented it as a linear combination of training samples from all classes:

$$\mathbf{y} = \mathbf{X}\mathbf{w} \tag{3}$$

where $\mathbf{X} = [\mathbf{X}_1 \ \mathbf{X}_2 \ \cdots \ \mathbf{X}_C] \in \Re^{q \times m}$ is a matrix containing all training eigenbrains from all classes, $m = \sum_{c=1}^{C} p_c$ is the number of training eigenbrains, and $\mathbf{w}$ is the coefficient vector whose entries are zero expect those of class c, $\mathbf{w} = [0, \cdots, 0, \boldsymbol{\alpha}_c^T, 0, \cdots, 0]^T \in \Re^m$.

In the case where $q < m$, the system of equations $\mathbf{y} = \mathbf{X}\mathbf{w}$ is underdetermined and to obtain a feasible solution we need to place some constraints. A natural approach is to choose constraints based on the ℓ_2-norm, however, this approach does not take into account the structure of our data where most of the coefficients are expected to be (or close to) zero. Hence seeking a sparse solution describes better the desired one. A sparse solution can be obtained by the following ℓ_1-minimization problem:

$$\hat{\mathbf{w}} = \arg\max \|\mathbf{w}\|_1 \text{ subject to } \mathbf{X}\mathbf{w} = \mathbf{y} \tag{4}$$

An alternative to ℓ_1-norm is the ℓ_0-norm, however, in the case of ℓ_0-norm the complexity of the problem is much higher [19], [20]. Also, under some circumstances the two solutions coincide [19]. Until now, we assumed that Eq. (3) holds exactly, however, in real cases the EEG trials are noisy, hence, a more accurate model must take into account this noise. Now, the model describing the relation between the test eigebrain and the training eigenbrains is given by:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e} \tag{5}$$

where $\mathbf{e} \in \Re^q$ is the noise term with bound energy $\|\mathbf{e}\|_2 \leq \epsilon$. Also, the ℓ_1-minimization problem is transformed to:

$$\hat{\mathbf{w}} = \arg\max \|\mathbf{w}\|_1 \text{ subject to } \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2 \leq \epsilon \tag{6}$$

which can be written as [19], [21]:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \{\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \rho\|\mathbf{w}\|_1\} \tag{7}$$

Now that we have seen how a test eigenbrain can be described as a linear combination of training eigenbrains, we will discuss how we could use this linear combination to provide a classification rule. In an ideal scenario, the solution $\hat{\mathbf{w}}$ should have non zero coefficients in indices that correspond to training eigenbrains that belong to the same class with the test eigenbrain. However, this is not the case since SSVEP responses are very noisy and non-stationary signals, thus non-zero coefficients could appear on the indices of other eigenbrains than the desired eigenbrain. In the literature different approaches have been proposed on how to deal with this [18], [22]. The approach based on residuals tends to be the most used. More specifically, if we let $\delta_c(\cdot) : \Re^m \to \Re^m$ to be a function that selects the coefficients associated with the class c, while zeroing all irrelevant coefficients, then we can calculate the residuals

for each class as: $r_c(\mathbf{y}) = \|\mathbf{y} - \mathbf{X}\delta_c(\hat{\mathbf{w}})\|_2$, $c = 1, \cdots, C$. The class for the given test eigenbrain is found by taking the minimum of the residuals $class(\mathbf{y}) = \arg\min_c\{r_c(\mathbf{y})\}$. The overall algorithm is described in Algorithm 1.

We can see that the algorithm contains two basic steps. The first step is related to the minimization problem, while the second step is related to the classification rule. A typical approach to solve the above ℓ_1-minimization problem is the Basis Pursuit (BP) algorithm [19]. However, in Compress Sensing (CS) literature we can found many other solvers, that presents better performance than BP [23] [24] [25] and they also take into account various other properties of the data such as group sparsity [23] [26] [25]. It is our intention in this work to explore other possibilities for the sparse graph-based representations of a given test trial.

Algorithm 1 Basic Sparse Representation Classification scheme [18], [22]

Require: Training samples, $\mathbf{X}$, and one test sample, $\mathbf{y}$

1. Solve the minimization problem:
$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}\{\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \rho\|\mathbf{w}\|_1\}$
2. Calculate the residuals:
$r_c(\mathbf{y}) = \|\mathbf{y} - \mathbf{X}\delta_c(\hat{\mathbf{w}})\|_2$, $c = 1, \cdots, C$

Ensure: $class(\mathbf{y}) = \arg\min_c\{r_c(\mathbf{y})\}$

C. Sparse Bayesian Model and Graph-based Prior

a) Graph-based Prior [27]: From a machine learning perspective, sparsity is a very helpful property since the processing is faster in a sparse representation where few coefficients reveal the information we are looking for. Hence, sparse priors help us to determine the model order in an automatic way and reduce its complexity. In addition to the sparseness, other structures of the data such as closeness or proximity, can be introduced into the bayesian framework by treating carefully the prior distribution. In our study we want to use information related to the proximity of the training eigenbrains. The proximity between training eigenbrains will be introduced to our model by using the notion of *Graph*.

More specifically, the coefficients $\mathbf{w}$ are treated as a random variable following:

$$p(\mathbf{w}|\mathbf{a};\boldsymbol{\lambda}) \propto \Big(\frac{|\mathbf{L}^T\boldsymbol{\Lambda}\mathbf{L}|}{2\pi}\Big)^{m/2} \exp\Big\{-\frac{1}{2}\mathbf{w}^T\mathbf{L}^T\boldsymbol{\Lambda}\mathbf{L}\mathbf{w}\Big\} \quad (8)$$

In our study, we assume that the matrix $\boldsymbol{\Lambda}$ is a diagonal matrix with elements $a_i^{-1}\lambda_i^{-1}, i = 1, \cdots, m$. Each parameter a_i, which controls the prior distribution of the parameters $\mathbf{w}$, follows a Gamma distribution, so the overall prior over all a_i is a product of Gamma distributions given by: $p(\mathbf{a}) = \prod_{i=1}^m Gamma(a_i; b_a, c_a)$. Furthermore, parameters λ_i are assumed known and deterministic quantities at this point. Graph's structure is incorporated into the matrix $\mathbf{L}$. A graph can be represented by various kinds of matrix representations such as the adjacency matrix, the incidence matrix, the Laplacian matrix etc. In our study we use the the Laplacian matrix. More specifically, the matrix $\mathbf{L}$ is the Laplacian matrix of the graph $\mathcal{G}$. The graph $\mathcal{G}$ is constructed as follows: each eigenbrain (or feature vector) $\mathbf{f}_i$ is a node of the graph, while, an edge is created between two nodes if they belong to the same SSVEP response (or class).

b) Prior for Noise: The overall precision (inverse variance) β of the noise follows a Gamma distribution: $p(\beta) = Gamma(\beta; b, c) = \frac{1}{\Gamma(c)}\frac{\beta^{(c-1)}}{b^c}\exp\Big\{-\frac{\beta}{b}\Big\}$, where b and c is the scale and the shape of the Gamma distribution, respectively. We use the Gamma distribution since this distribution is conjugate to the Gaussian distribution, and, it places the positivity restriction on the overall variance and the scaling parameters.

So, the overall prior over model parameters $\{\mathbf{w}, \mathbf{a}, \beta\}$ is given by: $p(\mathbf{w}, \mathbf{a}, \beta; \boldsymbol{\lambda}) = p(\mathbf{w}|\mathbf{a}; \boldsymbol{\lambda})\prod_{i=1}^m p(a_i)p(\beta)$.

c) Likelihood: The likelihood of the data is given by:

$$p(\mathbf{y}|\mathbf{w}, \beta; \boldsymbol{\lambda}) = \frac{\beta^{\frac{q}{2}}}{(2\pi)^{\frac{q}{2}}} \cdot \exp\Big\{-\frac{\beta}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})\Big\} \quad (9)$$

d) VB Update Equations: To apply the VB methodology [28] we need to define an approximate posterior based on one factorization over the parameters $\{\mathbf{w}, \mathbf{a}, \beta\}$. In our study we choose the following factorization: $q(\mathbf{w}, \mathbf{a}, \beta; \boldsymbol{\lambda}) = q(\mathbf{w}|\mathbf{a}; \boldsymbol{\lambda})\prod_{i=1}^m q(a_i)q(\beta)$. Applying the VB methodology, and taking into account the above factorization, the following posteriors are obtained:

$$q(\mathbf{w}) = \mathcal{N}(\hat{\mathbf{w}}, \mathbf{C}_{\mathbf{w}}), \quad (10)$$

$$q(\beta) = Gamma(\beta; b', c'), \quad (11)$$

$$q(\mathbf{a}) = \prod_{i=1}^m Gamma(a_i; b_{a_i}^{'}, c_{a_i}^{'}), \quad (12)$$

The moments of each distribution are calculated by applying iteratively the following equations until convergence:

$$\mathbf{C}_{\mathbf{w}}^{(k+1)} = (\hat{\beta}^{(k)}\mathbf{X}^T\mathbf{X} + \mathbf{L}^T\hat{\boldsymbol{\Lambda}}^{(k+1)}\mathbf{L})^{-1}, \quad (13)$$

$$\hat{\mathbf{w}}^{(k+1)} = (\hat{\beta}^{(k)}\mathbf{X}^T\mathbf{X} + \mathbf{L}^T\hat{\boldsymbol{\Lambda}}^{(k+1)}\mathbf{L})^{-1}\hat{\beta}\mathbf{X}^T\mathbf{y}, \quad (14)$$

$$\frac{1}{b_{a_i}^{(k+1)'}} = \frac{\lambda_i^{(k+1)}}{2}\Big(\hat{\mathbf{w}}^{(k+1)T}\mathbf{L}_i^T\mathbf{L}_i\hat{\mathbf{w}}^{(k+1)} + trace(\mathbf{L}_i\mathbf{L}_i^T\mathbf{C}_{\mathbf{x}}^{(k+1)})\Big) + \frac{1}{b_a}, \quad (15)$$

$$c_{a_i}^{(k+1)'} = \frac{1}{2} + c_a, \quad (16)$$

$$\hat{a}_i^{(k+1)} = b_{a_i}^{(k+1)'}c_{a_i}^{(k+1)'}, \quad (17)$$

$$\frac{1}{b_{\beta}^{(k+1)'}} = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w}^{(k+1)})^T(\mathbf{y} - \mathbf{X}\mathbf{w}^{(k+1)}) + tr(\mathbf{X}^T\mathbf{X}\mathbf{C}_{\mathbf{w}}^{(k+1)}) + \frac{1}{b}, \quad (18)$$

$$c_{\beta}^{(k+1)'} = \frac{N}{2} + c, \quad (19)$$

$$\hat{\beta}^{(k+1)} = b_{\beta}^{(k+1)'}c_{\beta}^{(k+1)'}, \quad (20)$$

In the above equations, $\mathbf{L}_i$ is the i-th column of the Laplacian matrix, describing the neighborhood of i-th eigenbrain (or EEG trial). Also, the matrix $\hat{\boldsymbol{\Lambda}}^{(k+1)}$ is a diagonal matrix with $\hat{a}_i^{(k)} \cdot \lambda_i^{(k+1)}$ in its main diagonal. For $\lambda_i^{(k+1)}$ we follow

the considerations of [29] and we set them to $\frac{1}{|\hat{x}_i^{(k)}|}$. With respect to other similar approaches [30], [31] we can observe the difference in Eqs. 14 and 15. More specifically, in our approach the parameter $b_{a_i}^{'}$ is weighted by the corresponding parameter λ_i.

Concluding this section, we provide a description of the proposed classification algorithm in Algorithm 2. In the first step we construct the Laplacian matrix of the graph. Then, we find the coefficients, and finally, we apply the classification rule. Our classification rule is based on the maximum value of coefficients. The position of the maximum value in the vector $\mathbf{w}$ determines the class - label. The rationality behind using the max-rule is that the coefficient with the maximum value shows us which training sample is more similar to the test sample. Additionally, the max-rule, in our preliminary results on SSVEP data, has presented better results than the classification rule (step 2) of Algorithm 1.

Algorithm 2 Sparse Graph-based Representation Classification scheme (SGRC)

Require: Training samples, $\mathbf{X}$, with its corresponding labels, ℓ, and one test sample, $\mathbf{y}$

1. Construct the Laplacian matrix $\mathbf{L}$ of the graph.
2. Find vector $\mathbf{w}$ by applying iterative procedure described by Eqs. (13) - (20)
3. Find position, p, of maximum value in vector $\mathbf{w}$

Ensure: $class(\mathbf{y}) = \ell(p)$

III. Experimental Study and Results

A. SSVEP datasets

In order to evaluate the performance of the proposed methodology, we have used two SSVEP datasets, *EPOC dataset* and the *Speller*, which are described below.

EPOC dataset: EEG signals, from 11 subjects executing a SSVEP-based experimental protocol, were acquired. The frequencies of visual stimulation were: 6.66Hz, 7.50Hz, 8.57Hz, 10.00Hz and 12.00Hz. The experiment was designed using the OpenViBE tool where the visual stimuli were projected on a 22 inches LCD monitor, with a refresh rate of 60 Hz and 1680x1080 screen resolution. EEG data were recorded with the Emotiv Epoc, using 14 wireless channels and a sampling rate of 128 Hz. In our analysis, we have used EEG data from channels O_1, and O_2. Each subject was asked to gaze at one of the visual stimuli indicated by the stimulus program in a random order, and complete 20 trials for each of the five targets. The EEG data have been band-pass filtered from 5Hz to 45Hz. More information about this dataset can be found in [12] and https://physionet.org/content/mssvepdb/1.0.0/.

a) Speller [32]: In this dataset, 40-target visual stimuli were presented on a 23.6-in LCD monitor. In this dataset participated thirty five healthy subjects (with normal or corrected-to-normal vision), where eight of them had experience with using a SSVEP-based BCI speller. EEG data were recorded with 64 channels according to an extended 10–20 system. In our study we have used only electrode O_z, lying in the center of the occipital areas. For each subject, the experiment consisted of 6 blocks. In each block, subjects were asked to gaze at one of the visual stimuli indicated by the stimulus program in a random order for 5s, and complete 40 trials corresponding to all 40 targets. Data epochs were extracted according to event triggers generated by the stimulus program. All data epochs were down-sampled to 250Hz. The EEG data have been band-pass filtered from 8Hz to 90Hz with an infinite impulse response (IIR) filter using the *filtfilt* function in MATLAB. As indicated in [32] a delay of 140ms in the visual system was considered.

The visual latency between the stimulus and the SSVEP response plays an important role in SSVEP detection. An accurate estimation of the visual delay ensures that the extracted data epochs only contain SSVEP responses to the stimulation. In the *Speller* dataset the visual latency is defined using the classification accuracy [32], while, in the *EPOC dataset* an alignment procedure is used to extract data epochs containing the SSVEP responses.

B. Results

In our analysis we compare the proposed algorithm (SGRC) to three well known algorithms, the CCA [7], the MLR [15], and the multiclass SVM (mSVM) with coding scheme *one_vs_one*. The first two approaches are widely used methods for the detection of SSVEP responses, while, the mSVM is a well - known classifier. For evaluating the performance of the examined algorithms we have used the Classification Accuracy, which constitutes the most straightforward performance evaluation metric, and it is defined as the ratio between the number of correctly classified trials to the total number of trials. Finally, the Leave-One-Block-Out (LOBO) cross validation technique was adopted. This cross-validation technique is more robust to the time-varying nature of the EEG signal.

Fig. 1 shows the averaged accuracies over all subjects for various time windows (TW) of trials on the *EPOC dataset*. We can observe that the SGRC method outperforms other approaches for all TW values in terms of accuracy. Fig. 2 shows the averaged accuracies over all subjects for various TW of trials on the *Speller* dataset. Here, we can observe that SGRC and MLR present very similar performance, while, both methods outperforms mSVM and CCA for large TW values ($\geq$1.5s). For TW $\leq$ 1 the mSVM presents better performance than the other methods. It is clear from the reported results on two different SSVEP datasets that the SGRC method presents the best behaviour since in both cases achieves the most competing performance.

Furthermore, it is worth to discuss more thoroughly the SGRC method with mSVM method since these two methods can be considered to belong in the same category of classifiers [18], [22], [33]. From the reported results we can observe that the SGRC method presents superior performance compared to the mSVM. More specifically our method presents better results than the mSVM for all TWs in the case of *EPOC* dataset, a SSVEP dataset that can be considered a difficult and noisy dataset due to the

EEG device that was used. In the case of *Speller* dataset, our method provides considerable better performance than mSVM in most of TWs. More specifically, in large TWs the difference in accuracy is around 10%. On the contrary, when the TW is small ($\leq$ 1s) then mSVM presents a slightly better performance than SGRC.

The SGRC algorithm does not have an explicit training step other than storing the training data (lazy learner), similar to Nearest Neighbor Classifier. In contradiction the MLR and the mSVM have a training step where the model parameters are learned, and then are fixed for the testing phase. Also, the CCA algorithm does not have a training step since the template is a mathematical model. This means that the SGRC method can be easily adapted to scenarios where the statistical properties of the class can be changed with the passage of time. This effect can be observed in problems which use EEG signals that are time-varying in nature. The only adaptation that is necessary for the SGRC is the collection of new samples. On the contrary, both MLR and mSVM must learn, from scratch, the new model parameters, and the CCA will need a new mathematical model (or template) to take into consideration the new statistical properties.

The prediction of SGRC is based on the comparison of the test eigenbrain with eigenbrains in the training set and for each test point provides a different local linear approximation. MLR and mSVM are creating a global model (approximate a target function) in order to provide prediction, while the CCA makes the prediction by comparing the test signal with a mathematical model. Due to the above observations, in the testing step, the computational cost of the SGRC algorithm is larger than the other algorithms since most of its computations are performed in this step. More precisely, when a test point arrives the computations on MLR and mSVM involve additions and multiplication, while the SGRC method needs to solve an optimization problem. Also, the CCA algorithm needs to solve a generalized eigenvalue problem. Due to this shortcoming of the SGRC method, incremental strategies [27] of SGRC is advisable for real-time SSVEP BCI systems.

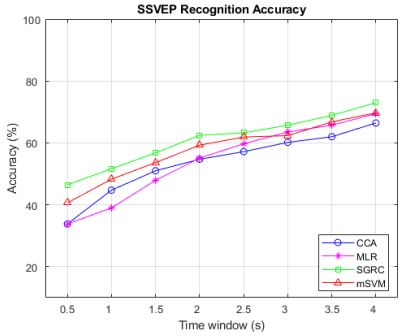

Fig. 1. EPOC dataset: Average accuracy over all subjects.

IV. Conclusions

In this paper, a novel classification algorithm is proposed to classify SSVEP responses, a multiclass classification problem. The algorithm exploits the sparse graph-based representations of a test signal with the help of training signals. Besides sparsity, the algorithm exploits the dictionary structure by adopting the concept of 'eigenbrain'. Eigenbrains consitute a mathematical basis for the brain function. Experiments in two SSVEP datasets, both publicly available, have shown the usefulness of the proposed algorithm. More specifically, the comparison of the proposed algorithm with the CCA, the MLR and the mSVM has shown that the proposed algorithm provides us with superior performance when we have a small number of EEG sensors. In the future, we intend to study how SRC schemes could be modified to develop incremental classification schemes. These schemes are very useful in SSVEP applications due to the time varying nature of EEG signal. Also, we intend to adopt other approaches to incorporate the graph's structure into the model. In addition, approaches using filter banks have been shown to be useful in the analysis of SSVEPs, hence, future extensions of our algorithm could be based on filter banks. Finally, more thorough studies with respect to the classification rule (step 3 of Algorithm 2), and its alternatives, of our algorithm will be performed.

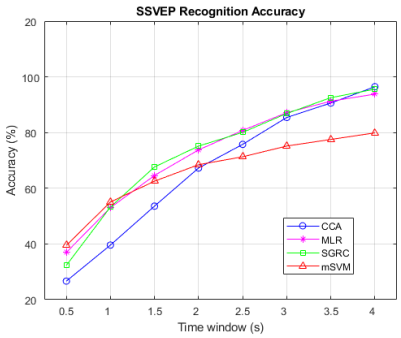

Fig. 2. Speller dataset: Average accuracy over all subjects.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767 – 791, 2002.

[2] "Multimedia authoring and management using your eyes and mind." https://www.mamem.eu, 2018.

[3] G. Bin, X. Gao, Y. Wang, B. Hong, and S. Gao, "VEP-based brain-computer interfaces: time, frequency, and code modulations ," *IEEE Computational Intelligence Magazine*, vol. 4, pp. 22–26, November 2009.

[4] R. Zerafa, T. Camilleri, O. Falzon, and K. P. Camilleri, "To train or not to train? a survey on training of feature extraction methods for SSVEP-based BCIs," *Journal of Neural Engineering*, vol. 15, p. 051001, jul 2018.

[5] S. Gao, Y. Wang, X. Gao, and B. Hong, "Visual and Auditory Brain Computer Interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1436–1447, May 2014.

[6] H. Wang, T. Li, and Z. Huang, "Remote control of an electrical car with SSVEP-Based BCI," in *2010 IEEE International Conference on Information Theory and Information Security*, pp. 837–840, 2010.

[7] Z. Lin, C. Zhang, W. Wu, and X. Gao, "Frequency Recognition Based on Canonical Correlation Analysis for SSVEP-Based BCIs," *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 2610–2614, Dec 2006.

[8] O. Friman, I. Volosyak, and A. Graser, "Multiple Channel Detection of Steady-State Visual Evoked Potentials for Brain-Computer Interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 742–750, April 2007.

[9] Y. Zhang, G. Zhou, J. Jin, M. Wang, X. Wang, and A. Cichocki, "L1-Regularized Multiway Canonical Correlation Analysis for SSVEP-Based BCI," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, pp. 887–896, Nov 2013.

[10] G. Bin, X. Gao, Y. Wang, Y. Li, B. Hong, and S. Gao, "A high-speed BCI based on code modulation VEP," *Journal of Neural Engineering*, vol. 8, no. 2, p. 025015, 2011.

[11] M. Nakanishi, Y. Wang, Y. Wang, and T. Jung, "A Comparison Study of Canonical Correlation Analysis Based Methods for Detecting Steady-State Visual Evoked Potentials," *PLoS ONE*, p. e0140703, October 2015.

[12] V. P. Oikonomou, G. Liaros, K. Georgiadis, E. Chatzilari, K. Adam, S. Nikolopoulos, and I. Kompatsiaris, "Comparative evaluation of state-of-the-art algorithms for SSVEP-based BCIs." arXiv:1602.00904, February 2016.

[13] H. Cecotti, "A time-frequency convolutional neural network for the offline classification of steady-state visual evoked potential responses," *Pattern Recognition Letters*, vol. 32, no. 8, pp. 1145 – 1153, 2011.

[14] N.-S. Kwak, K. Muller, and S.-W. S.-W.Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PLOS ONE*, vol. 12, pp. 1–20, 02 2017.

[15] H. Wang, Y. Zhang, N. R. Waytowich, D. J. Krusienski, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Discriminative feature extraction via multivariate linear regression for ssvep-based bci," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, pp. 532–541, May 2016.

[16] V. P. Oikonomou, S. Nikolopoulos, and I. Kompatsiaris, "A bayesian multiple kernel learning algorithm for ssvep bci detection," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 1990–2001, Sep. 2019.

[17] E. Maiorana, D. L. Rocca, and P. Campisi, "Eigenbrains and eigentensorbrains: Parsimonious bases for eeg biometrics," *Neurocomputing*, vol. 171, pp. 638–648, 2016.

[18] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, Feb 2009.

[19] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, "Introduction to compressed sensing," in *Compressed Sensing: Theory and Applications* (Y. C. Eldar and G. Kutyniok, eds.), pp. 1–64, Cambridge University Press, 2012.

[20] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, April 2006.

[21] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Boston: Birkhaüser, 2013.

[22] Y. Shin, S. Lee, J. Lee, and H.-N. Lee, "Sparse representation-based classification scheme for motor imagery-based brain–computer interface systems," *Journal of Neural Engineering*, vol. 9, p. 056002, Aug 2012.

[23] S. D. Babacan, S. Nakajima, and M. N. Do, "Bayesian group-sparse modeling and variational inference," *IEEE Transactions on Signal Processing*, vol. 62, pp. 2906–2921, June 2014.

[24] L. He, H. Chen, and L. Carin, "Tree-structured compressive sensing with variational bayesian analysis," *IEEE Signal Processing Letters*, vol. 17, pp. 233–236, March 2010.

[25] V. P. Oikonomou, S. Nikolopoulos, and I. Kompatsiaris, "A novel compressive sensing scheme under the variational bayesian framework," in *27th European Signal Processing Conference (EUSIPCO 2019)*, pp. 1–4, Sep. 2019.

[26] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, pp. 1982–2001, April 2010.

[27] B. Jiang, H. Chen, B. Yuan, and X. Yao, "Scalable graph-based semi-supervised learning through sparse bayesian model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2758–2771, 2017.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, October 2007.

[29] G. Deng, "Iterative learning algorithms for linear gaussian observation models," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2286–2297, Aug 2004.

[30] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Mach. Learn. Research*, vol. 1, pp. 211–244, 2001.

[31] C. M. Bishop and M. Tipping, "Variational relevance vector machines," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, (San Francisco, CA, USA), pp. 46–53, Morgan Kaufmann Publishers Inc., 2000.

[32] Y. Wang, X. Chen, X. Gao, and S. Gao, "A benchmark dataset for SSVEP-based brain - computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, pp. 1746–1752, Oct 2017.

[33] V. P. Oikonomou, S. Nikolopoulos, and I. Kompatsiaris, "Robust motor imagery classification using sparse representations and grouping structures," *IEEE Access*, vol. 8, pp. 98572–98583, 2020.