BNCI Evaluation Suite

In the following Documentation pages we describe the different modules implemented for the BNCI Evaluation Suite. This toolbox aims at the implementation of several functionalities around the Enobio sensor for the realization of BCI functionalities. The idea of the toolkit is to provide some functionalities in Matlab that can be combined for prototyping and evaluating BCI solutions. These can be implemented within the toolkit or together with further BCI prototyping MATLAB toolkits like BCILAB.

BCI systems attain to translate brain signals, which can include several physiological modalities, into control commands of a particular device, e.g. wheelchair, computer, mobile phone, robot arm. A generic block diagram of such systems can be observed in Figure 1. Taking such a structure as a reference, if a system makes use of a classification procedure as feature translator, the resulting system can be analysed from a pattern recognition perspective. This is the approach we follow in the BNCI Evaluation Suite. Therefore you can find functionalities for feature extraction and for classification in multi-class problems.

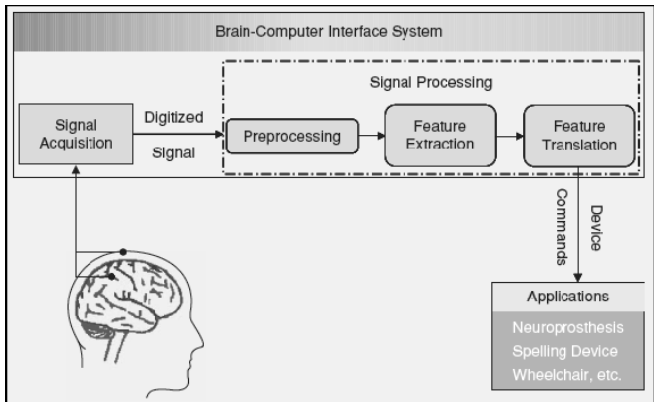

Figure 1. Block diagram of a generic BCI system based on a pattern recognition approach. Reproduced from [Kachenoura et al. 2008] with permission.

Different paradigms for the realization of BCI exist. Among them the more common used ones are so-called motor imagery, P300, and SSEVP. The implementation works described herein are focused on the motor imagery, where control commands are generated upon the classification of the corresponding EEG signal streams into diverse classes, and p300 paradigms, where control signals are generated after a particular Event Related Potential (ERP) placed at 300 ms after setting of stimuli.

In the here described works we have decided to use two different frameworks respectively for motor imagery and P300 data classification. The work by Gao Xiaorong and colleagues presented at the BCI Competition III and described here, has been implemented for the motor imagery paradigm. Some variations of the original work as described in [Cester and Soria, 2011] have been implemented.

As for the P300 code we provide the code presented in [Rakotomamonjy and Guige, 2008]. We offer the code as provided by the authors, together with a refinement we have undertaken in order to refine the function interfaces. Our version of the code is the current version of a work in progress.

Summary and structure of toolkit:

- featureExtraction: Directory including different feature extraction methodologies
- multiclass: Directory including different classification, data fusion, and performance

evaluation methodologies
- motorImageryDemos: Demo scripts use din the implementation of the motor imagery framework mentioned above
- p300: P300 framework as provided by Rakotomamonjy
- p300toolkit: Reproduction of the Rakotomamonjy framework as implemented in this toolkit

1 Bagging System for motor imagery BCI and Compared Feature Stages

Motor imagery is known to trigger event related desynchronization (ERD) of the mu (10-12 Hz) and beta (20-25 Hz) rhythm in related motor cortex areas. The mu rhythm is generated mainly in the post-Rolandic somatosensory area and the central beta rhythm in the pre-Rolandic motor area. The motor imagery classes can be distinguished because the corresponding EEG signals have been generated by the subject by imagining different types of movements, e.g. right hand, left foot, tongue.

We have used Xiaorong et al.'s system[1] for comparison of the feature extraction and selection stages. We detail the different parts of this system in the following paragraphs and mention the changes we have undertaken and tested. The pre-processing in this system includes the following stages:
- Cutting in epochs.
- Mean subtraction.

The feature extraction stage includes then three different steps:
- Laplacian computation.
- Filter the signals in an 8-30 Hz band.
- OVR, which is explained in detail in Sec. 4.1.2.2.1.

This feature extraction stage has been compared in AsTeRICS works with the extraction of wavelet coefficients, and two different feature selection approaches based on ANOVA and Gas. The code used for such a comparison has been integrated in the Evaluation Suite.

Lastly the classification stage transforms the selected features into a class membership function of the 4 imagined motor movements. This stage is realized through a multi-classifier methodology, which combines the classification of a support vector machine (SVM), a linear discriminant analysis (LDA), and a k-nearest neighbour (KNN) approaches. For this we have integrated following functions: classify (Statistics Toolbox), svmclassify (SVM and Kernel Methods Matlab Toolbox by S. Canu, et al., from Perception Systems et Information, INSA, Rouen, France)[2], and fuzzy_knn (Emre Akbas' implementation)[3].

The corresponding classifier fusion stage further processes the outputs of these classifiers. The classifier fusion stage is implemented again in a supervised manner. Therefore the best fusion operator among max, min, median, majority voting, sum, and product is sought for each time sample of the sequences in training data set. This operator sequence is further applied in the recall phase. Lastly a decimation stage is applied.

The described multi-classifier module is integrated in a bagging strategy, whereby it is trained with 10 different data subsets. The output of the corresponding decimation stages is then fused again in a further fusion stage, which we denote as bagging fusion, and then the membership label is

[1] We use the description in http://www.bbci.de/competition/iii/results/graz_IIIa/GaoXiaorong_desc.pdf because of the lack of scientific references on the system (to the best of our knowledge).
[2] http://asi.insa-rouen.fr/enseignants/_arakotom/toolbox/index.html
[3] http://www.mathworks.com/matlabcentral/fileexchange/13358-fuzzy-k-nn

generated. A sketch of feature translation stage is depicted in Figure 2.

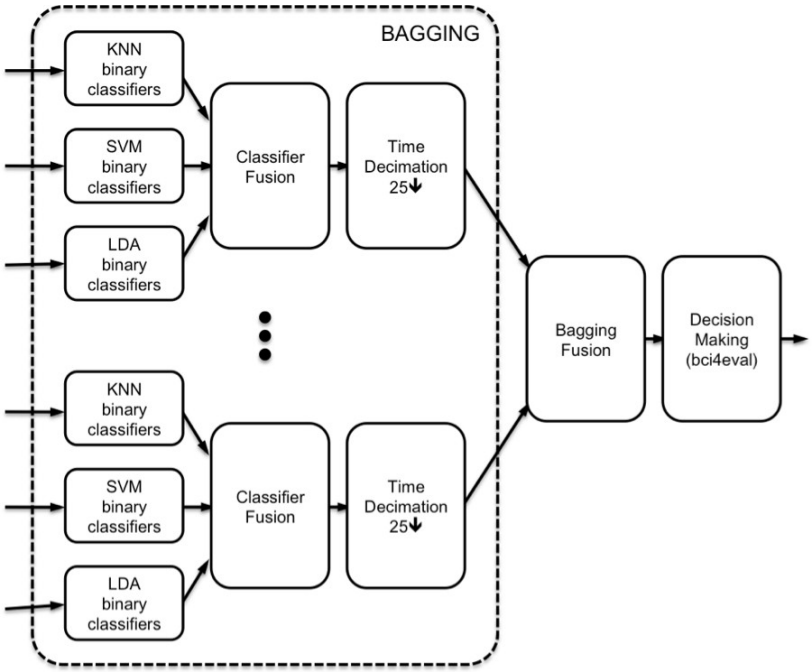

Figure 2. Block diagram of the classifier system proposed by Xiaorongs et al. in BCI competition III (no known paper available).

We have focused our work in the feature extraction stage. In this context we compare the performance of using the current OVR approach with this of a wavelet transformation based one. We further compare two different feature selection strategies based on the well-known ANOVA and on Genetic Algorithm methodologies. We have implemented all mentioned methjodologies, although the GA needs from the corresponding MATLAB toolkit, together with the classification system for its usage in the BNCI Evaluation Suite, making all available for BCI research. In this context it is worth mentioning the lack of known papers explaining the complete methodology, so we have based the implementation on the information delivered via the BCI competition III web site[4].

We describe in detail the implemented feature extraction stages in the following sections.

1.1 One Versus the Rest

Common Spatial Patterns (CSP) constitute a frequent approach in BCI technologies for the improvement of classification-based systems. The methodology attains the extraction of suitable features. CSP constitutes a supervised feature extraction procedure and it is generally used in binary classification problem. Moreover CSP is based on the projection of the original features into a new feature space. Hence CSP makes use of the ground truth in the training set, i.e. the actual class membership of each epoch in the training set is employed in order to find the projection matrices for the feature space associated to the output class. This intermediate projection attains increasing the capability of discriminating among the features corresponding to each class. This is achieved by minimizing the intra-class variability and maximizing the interclass one.

One Versus the Rest (OVR) is one of the existent generalizations of CSP for multi-class problems. Therefore OVR implements a projection of the feature space into several feature spaces, i.e. one per class, whereby the distances among features of different classes are maximized. We have implemented the OVR approach for its usage within the BNCI Evaluation Suite.

[4] http://www.bbci.de/competition/iii/results/graz_IIIa/GaoXiaorong_desc.pdf

OVR attains in this framework the computation of 4 different projection matrices, i.e. the number of motor imagery classes. As a result of its application, 4 different projected epochs are obtained from each one. This requires the assessment of those projection matrices.

1.2 Joint Time-Frequency through Wavelet Coefficients

Additionally we have implemented an approach for conducting a joint Time-Frequency analysis, which can be used for feature extraction in the analysis of motor imagery data. The implemented approach avoids the general physiological priors with respect to frequency bands and electrode positions that usually drive the feature selection. Therefore we do not look for particular bands and/or electrode locations. We use first an extensive time-frequency analysis instead. Hence single trial classification of motor imagery is performed based on time-frequency information derived from 60 channels raw EEG signal by means of wavelet analysis. This has been realized by applying a Morlet wavelet, which has been recommended for the analysis of EEG signals.

The pre-processing and joint time-frequency feature extraction process applied can be given as follows:

- Decimate the raw signal by a factor of 4, from 250 to 62.5 Hz.
- Trials with signals over +-100 uV are considered artifacts and rejected.
- Signal of all channels is referenced to the mean signal of all channels.
- Wavelet coefficients computation as described in the following paragraph.

Time-frequency information from the signals is extracted through a complex Morlet wavelet with bandwidth parameter F_b=1 and a wavelet centre frequency F_c=1.5. The module of the solution is selected as representative of the power of the signal for each frequency band at each electrode site. A total of 30 coefficients corresponding to 1-30 Hz bandwidth are obtained for each of the 60 channels. That is a total of 1800 coefficients for each time sample.

Since the number of the extracted wavelet coefficients is rather large, a feature selection stage should be better applied. This attains to prune the most relevant time-frequency information from all available data. The main goal is to fit the feature selection to each specific subject and session, and to avoid a priori discarding relevant information based on physiological priors. As already mentioned we compare the performance of selection through analysis of variance (ANOVA) and of Genetic Algorithms for the implementation of this stage. For the GA implementation we provide the implementation of the fitness functions, which can be used with the MATLAB GA Toolkit.

2 SVM ensemble for p300 detection

One further task in AsTeRICS was devoted to the development of p300 state-of-the-art for the classification of p300 data. We have focused the works in the spell methodology as published by [Rakotomamonjy and Guiges, 2008]. The goal of the works is to implement and test the presented methodology. Moreover we are trying to test whether the published schema is susceptible of being improved for the same application of spelling. At the current stage the integrated version in the toolkit is in progress. So we provide the original code as provided by the paper authors, together with a preliminary refined version, which we expect to be more modular than the original one.

3 Framework independent implemented functions

We described in the following section some of the implemented methodologies that can be used within the formerly described frameworks but in new ones to be implemented with the Evaluation

Suite.

3.1 Data processing

In this category we group functionalities for transforming input signals into intermnediate signals that can be further processed.

3.1.1 Morlet wavelet extraction

This function pre-processess and extracts the Wavelet coefficients for BCI motor imagery data. It combines these steps with the ANOVA feature selection. This function has been used in our motor imagery works.

The stages included in the function involve the following:

1. Baseline extraction and decimation
2. Reject epoch with NaN's
3. Reference to mean all channels
4. Reject epoch by thresholding
5. Save data in a file array for handling large array size
6. Compute time-frequency representation using wavelets (1-30 Hz -->30 coefs for each channel).
7. Calculates an Anova to find the most significant coefficients for class discrimination.

3.1.2 Data standardization

This functionality normalizes data in order to achieve zero mean and unit variance. This is an important step in order to avoid difference among features due to larger values.

3.2 Classification methodologies

We have implemented a multi-classifier functionality. It performs a classification using different classifiers available: K-nearest-neighbour, LDA, SVM and these of the BIOSIG toolbox. For the moment it is not possible to change parameters in BIOSIG classifiers. These functions can deliver a class label (option not enabled) or a posterior probability, which is the option the multi-classifier function uses. This is thought to allow a quick implementation of fusion schemes on the results. However for this we needed to know which is the output range of the BIOSIG classifiers. We observed two types [-infty,0], and [0,1], so that some normalization function has to be applied prior to its fusion. After some discussions with Alois Schloegl responsible for BIOSIG development[5] it is clear that each classifier has its own output range and that no normalization function can be recommended in a general form for all of them. Therefore we decided to output BIOSIG classifiers as they are (leaving normalization to be implemented outside this function). Therefore the only normalized outputs are those of the original LDA and fuzzy KNN, which are normalized to range [-1,1].

3.3 Data fusion operators

As analysed in the State of the Art (AsTeRICS deliverable D4.1) one further gap to be covered in the implementation of the Evaluation Suite was focused on the development of further operators for BCI data fusion.

3.3.1 Simple fusion operators

The most basic operators develop in mathematics are the sum and the product. These operators have

[5] Personal communication.

been used together with some other lightly evolved ones like the ordinal operators maximum, median and minimum and the majority voting operator in data fusion from an early stage of research. The majority voting one fuses class labels, i.e. decision level fusion. It can be shown that in this case this operator is equivalent to the sum rule over the class labels. We provide a majority vote fusion operator in the Evaluation Suite.

One implemented fusion stage is ta muti-operator scheme, which can be trained by means of a training data set. In this function the simple fusion operators are selected among the set of operators mentioned in the former paragraph at sample per sample basis, i.e. one operator is selected for each time sample. This has been included in the BNCI Evaluation Suite and used in the motor imagery works.

3.3.2 Weighted Sum

The weighted sum is an operator used in different application domains, e.g. descriptive statistics, neural networks. It presents the same structure as the mean but with the particularity that weights can be established on the values being operated

$$z = \frac{\sum_{i=1}^{n} w_i \cdot x_i}{\sum_{i=1}^{n} w_i} \quad (5),$$

where the sum of the weights is usually normalised to sum up to 1 (this ensures as well working in the unit hypercube). We provide the corresponding implementation in the Evaluation Suite.

3.3.3 Order Weighted Averaging (OWA)

A generalisation of the average, where the weighting is established upon ordinal data, was proposed by Yager in 1988 and denoted as Ordered Weighted Averaging (OWA). OWA operator presents the following expression:

$$z = \sum_{i=1}^{n} w_{(i)} x_{(i)} \quad (28),$$

where w(i) are the weights of the operator. The bracketed subindices state for a sorting operation that is applied on x_i before aggregating their values, e.g. $x_{(1)}$ state for the larger x_i, $x_{(n)}$ for the lowest one. As it can be observed the weights are hence applied on the sorting result. This results in a unique weighting set, but that is applied to different channels on each canonical region of the unit hypercube. As it can be observed the OWA generalises the average as well as the minimum, and the maximum operators.

3.3.4 Fuzzy Integral

The fuzzy integral was proposed Sugeno in 1974 as a means of fusing data simulating subjective evaluation undertaken by humans. The operator present some similarities to the OWA described in the former section, since the applied weighting depends on the particular canonical subspace of the input variables in the unit hypercube, i.e. on the result of a sorting operation. In contrast to the OWA, the weighting set in the fuzzy integral is not unique, but it changes in each canonical region.

The fuzzy integral uses as fusion operators a combination of T- and S-norms (see Sec. 3.3.2.5). These define different types of fuzzy integrals. The most known of them are the Sugeno and the Choquet fuzzy integrals.

We first recall the expressions in order to elucidate the operators. The Sugeno fuzzy integral can be expressed as:

$$z = \bigvee_{i=1}^{n} \mu(A_{(i)}) \wedge x_{(i)} \tag{29},$$

where $\vee$ states for the maximum operator, $\wedge$, for the minimum, and $\mu()$, for the coefficients of the so-called fuzzy measures, i.e. the weighting coefficients in the fuzzy integral. There are 2^{n-1} coefficients, one for each subset that can be established on the information sources to be fused. As formerly mentioned, the bracketed indices represent the result of a sorting operation. Hence only n coefficients of the fuzzy measure are selected for the aggregation. These coefficients correspond to the subsets: $A_{(1)}=\{x_{(1)}\}$, $A_{(2)}=\{x_{(1)},x_{(2)}\},...$, and $A_{(n)}=\{x_{(1)},x_{(2)},...,x_{(n)}\}$. Therefore the actual weight set for each aggregation depends on the canonical region defined by the input variables $x_1,...,x_n$.

In case of the Choquet fuzzy integral the maximum and the minimum are respectively substituted by the sum and the product:

$$z = \sum_{i=1}^{n} [\mu(A_{(i)}) - \mu(A_{(i-1)})] \cdot x_{(i)} \tag{30},$$

where $\mu(A_{(0)})=0$. As it can be proofed the Choquet fuzzy integral generalises the weighted sum and the OWA operators.

The implemented version of the Choquet integral is in vectorized form. This allows a very efficient computation time for several fuzzy measures.

3.4 Performance evaluation functions

Different performance functions have been implemented in the BNCI Evaluation Suite. In this context and beyond the usual indices we came to the problem that measuring occurrence of events in a temporal sequence turns out to be not as trivial as initially thought. This has been solved by implementing two functionalities as described in the following sections.

3.4.1 Confusion matrix and TPR/FPR computation

We can use ground truth labels for computing the performance of a system. In binary classification problems the most suitable is to use the so-called Receiver Operating Curves, which relates True Positive Rate (TPR) to the False Positive Rate (FPR) with respect to the value of a particular parameter, usually the detection threshold, in a binary classification. The TPR is computed as the ratio of the number of points correctly detected as positives with respect to the number of actual positives. The FPR is the ratio among the number of points correctly detected as positives with respect to the number of actual positives number of points wrongly detected as positives with respect to the number of actual negatives. These two quantities can be derived from the computation of so-called confusion matrices, which in binary classification problems are 4-element matrices relating predicted positives and negatives versus actual positives and negatives.

ROCs can be used for visual inspection. The closer to the point of TPR=1, FPR=0, the better. If a numerical comparison is needed the Area Under the Curve can be used. This is the numerical integral of the ROC with respect to the FPR. In an AUC comparison, the larger, the better.

We have implemented in the AsTeRICS works different functionalities for the computation of confusion matrices, and its translation into TPR and FPR indices. These can be easily transformed when recalled recursively into ROC. Moreover we give access in the evaluation suite to the Kappa

index computation as implemented in the BioSig toolkit. The Kappa index is a unique ratio relation TPR and FPR.

3.4.2 Precision-Recall Curves

ROC present good properties that make their appearance independent of the proportion of positive and negative examples within binary classification problems. Moreover they can be converted in convex curves what eases the comparison among alternatives solving a classification problem. However there is a problem when detecting events in temporal streaming data. In this case and for a reasonable sampling frequency the number of positives to be detected is extremely small with respect to the number of negative samples. This translates into the fact that the TPR becomes less significant because of the few number of positives. Moreover it can become unstable as well because the numerator and denominator of this ratio tend to 0. We have undertaken some research in this context and come up with the advice of using so-called precision-recall (PR) curves with this type of data.

PR curves have been introduced within web research for evaluating the performance of search engines. They can be related to ROCs. While precision represents the same ratio as the TPR, the FPR is substituted by the recall ratio. Recall performance is computed as the ration between the number of points correctly detected as positives and the number of all points detected as positives. We have implemented functions within the Evaluation Suite for the computation of these performance measures.

3.4.3 Performance with temporal tolerance

As part of AsTeRICS works we propose herein a new scheme for the performance evaluation in event driven interaction systems. The final goal is to compute the Precision-Recall curves of a user interaction session as described in the former section.

One problem that arises in this context is based on the fact that the ground truth to be used for performance evaluation is given as a temporal stream. This is somehow different to performance evaluation in regular pattern recognition systems. The ground truth in applications like the ones being developed with the AsTeRICS Runtime Environment (ARE) are given as a temporal sequence of 0s (no event should be detected) and 1s, where events of interest have to be detected. This last part of the ground truth is usually given by acquisition software in form of a delta in the corresponding temporal position. However we think it is suitable to let the processing system realize the detection with some temporal tolerance, i.e. if the event is detected after 200 ms after the label this should be considered as a valid detection if the tolerance value is larger than 200 ms. We propose to transform the delta sequence of the ground truth into a pulse sequence with initial points in the position of the deltas. Moreover the duration of the pulses correspond to the allowed temporal tolerance in the event detection. One problem that arises if we undertake the performance evaluation with this pulse sequence is there can be more than one true positive in each pulse if the decision threshold delivers several detection pulses within the tolerance pulse. In order to avoid this problem we have used mathematical morphology. Particularly we make use of the operation denoted as reconstruction by dilation. In this operation the ones of a binary signal grow but restricting its possible growing areas to the ones of a further signal denoted as mask. Therefore we reconstruct by dilation the detection stream, i.e. the signal resulting from the decision threshold, using the tolerance pulse ground truth stream as a mask. After this we derive the reconstructed stream in order to account just one positive for each continuous pulse of detections.

The methodologies described in the former paragraphs have been implemented and integrated in the BNCI Evaluation Suite. The functionality allows generating a PR curve by taking the tolerance

interval into account. This allows on its own to analyse the performance of the detection system for different tolerance values.