

A Comparison of Classification Algorithms for Brain Computer Interface in Drug Craving Treatment

M. Mazzoleni*, F. Previdi*

* *Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione. Università degli Studi di Bergamo, Via Galvani 2, 24044 Dalmine (BG) (e-mail: mirko.mazzoleni@unibg.it).*

Abstract: In this paper, the use of Brain Computer Interfaces (BCIs) is proposed as a means to recover patients from craving diseases, with the aim of a clinical protocol. In order to understanding the BCI messages, a classification algorithm based on logistic regression has been developed. The choice was dictated by a comparison with other known classification techniques of different reasoning type, highlighting the pros and cons of them. Finally, a result regarding the brain areas which are more involved during the activity is reported.

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Brain Computer Interface, Neural activity Machine Learning, Pattern Recognition

1. INTRODUCTION

Brain Computer Interfaces (BCIs) are, as stated in their name, systems devoted to establish a dialogue between these two powerful processing units: the biological and the (only) logical one. They are composed by sensors (the headset) and algorithms that perform pattern identification and classification. According to the brain activities that have to be detected, two main approaches exist to control a BCI (Hoffmann et al. (2007)). In the first approach, subjects are exposed to a sequence of stimuli, while focusing on a particular one of them. When this target is recognized, an Event Related Potential (ERP), which represents an electrophysiological response to a specific stimulus, can be detected. Appearing 300 ms after a surprising or task-relevant event, and observed in EEG signals, the P300 is an example of such brain patterns, see Sutton et al. (1965). The BCI system therefore has to identify when the target stimulus has happened. The second approach consists of imagine a specific mental task, for example a hand movement or other actions. These activities are reflected by oscillatory brain waves in EEG data, such as the delta (0.5 – 4 Hz), theta (4 – 8 Hz), alpha and mu (8 – 13 Hz), beta (13 – 30 Hz), and gamma (31 – 40 Hz) rhythms. So, the BCI system must correlate the mental tasks with the features of the brain waves. Machine Learning algorithms are strongly used in BCI applications. In the view of a BCI system, a machine learning algorithm is expected to distinguish between mental statuses. This is accomplished by training a classification algorithm with data measured from the BCI electrodes. Since the subject knows what he/she was thinking, this information can be used to drive the algorithm training. This is known as supervised learning; otherwise the tuning is done in an unsupervised way.

The use of BCIs for medical purposes is an exciting challenge and already employed technology (Birbaumer et al. (1999)). As a *first contribution* of this paper, a therapy

for drug craving dependency using BCIs is introduced and its protocol developed. As a *second contribution*, a comparison of various types of learning algorithms has been made, discussing their strengths and weaknesses, in view of the application to addiction therapy. Algorithms such as Logistic Regression, Support Vector Machines, Naive Bayes, K-Nearest Neighbours and Decision Trees have been compared in terms of accuracy, computational time, easiness of results explanation (interpretability) and possibility to deal with the rate of false positive/negative decisions (adaptivity). These classifiers have been chosen as representative of different categories: linear, non-linear, probabilistic, instance-based and tree models. Finally, a *L1-regularized Logistic Regression* using features from frequency domain EEG data has been chosen. As a *third contribution*, a map of the brain areas and rhythms most involved in the training phase is presented for further interpretations.

The remainder of the paper is organized as follows. In Section 2, the use of BCI for the therapy of drug addiction diseases is explained. In Section 3, the steps involving the data acquisition setup and processing techniques adopted are described. Section 4 shows a comparison between different classifiers, with indications about the choices made, and a graphical visualization of the most used brain areas during the training stage is given. Section 5 is devoted to concluding remarks and future developments.

2. DRUG CRAVING THERAPY

The term *craving* refers to the impulsive desire for a psychoactive substance, food or any other rewarding thing: this supports the “addictive” behaviour and the compulsion aimed to avail oneself of the object of desire. Some authors have highlighted strong differences between the meaning that patients give to the term, and the interpretation given by the clininal staff, see Sitharthan et al. (1992). In this work, craving is considered as a strong desire

or an intense longing, as suggested by Kozlowski and Wilkinson (1987). Two models have been proposed to explain the mechanism with which the craving would contribute to cause a relapse. The first one suggests that this need shares common characteristics with the Obsessive-Compulsive Disorder (OCD), Modell et al. (1992). The second model tries to explain the craving as being induced by conditioning phenomena with positive and negative reinforcement mechanisms, Littleton (1995).

This work proposes the use of the BCI as an instrument to recover patients from craving diseases. The therapy consists of two consecutive moments: during the first activity, depicted in Fig. 1, the subject performs first a training of the device, and then uses the BCI in an active way, by trying to push away pictures which have a negative link with his/her disease. The training operation is done at the beginning of every cure session, in order to make the algorithm more reactive to the patient current mental state. It was experimentally observed, on healthy subjects, that the time of response and accuracy was higher on a freshly trained BCI, with respect to an algorithm trained even a few days before the test. This condition implies that the training session can't be beared for too much time. The training is done on the basis of two mental tasks: a *neutral task* and a *push task*. In the neutral task, the subject does not have to concentrate on a particular activity, while in the push task he/she is required to think about pushing away an object. The choice of focusing on this two tasks is due to the fact that most BCIs systems require the user to train as first action neutral thoughts. Furthermore, concentrating on only one of the two tasks makes the training phase lighter and faster, which can be beneficial for patients. Then, other actions could be trained. The ability to train correctly the BCI for two classes, i.e. the neutral one and another, is, based on the experiments carried out with the BCI equipped software, not time consuming (requiring several minutes). Just by only adding a third action, the training time grows and this is unacceptable in terms of a treatment of people who could be mentally unstable. By considering the neutral activity as a proper mental task, this problem has been overcome. During the treatment, via a software which can interact with the BCI, the patient undergoes to a series of different images, loaded by the software. This pictures could represent "bad" scenes, like those familiar with its craving disease. If such a picture appears, the subject has to actively try to repel it, thinking the push task: the picture then will be visually pushed away from the screen. If he/she fails to do this, or he/she is not concentrated enough, then a neutral task would be detected by the BCI system. In this case, the pictures will come toward the screen, forcing the patient to push it away. In case of pictures which represent "good" scenes (these are, as the bad scenes, subject dependent) he/she does not have to think the push task, but let the system detect the neutral activity.

The second moment of the therapy is related to the assessment of the cure. This could be done via the Stroop Test (see Jensen and Rohwer (1966) for a review of the topic). The test consists of displaying words, with one of two different font colors, for a defined time. The user has to select the color of the words prompted. The time

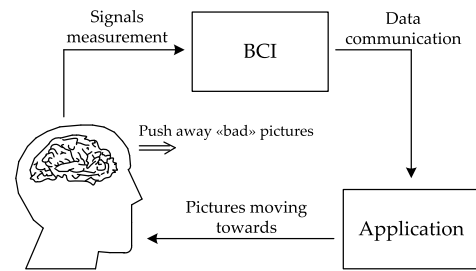


Fig. 1. Use of BCI for craving treatment

elapsed between the words appearance on screen and the response given is stored. When a word which meaning refers to something related to the craving disease, the "elaboration time" to give a response is afflicted by the word meaning, because it could be something which is of interest for the subject. By measuring these reaction times before and after the therapy, a degree of "drop of interest" for the craving-related words could be assessed. A software program which manages the first and the second moments of the therapy has been developed along with the machine learning algorithms explained in the next sections.

3. MENTAL TASK CLASSIFICATION

This section describes the methodological steps performed in order to define a classification algorithm for the Brain Computer Interface, which is capable of correctly identify between two mental tasks: a neutral task, and a thought of pushing away an object.

3.1 Experimental Setup

Data were collected from 14 EEG channels located on an Emotiv EPOC BCI headset, sampled at 128 Hz. The device, from its specification document, has an output band of [0.2 – 45 Hz] and measurements were already filtered via a notch filter at 50 Hz and 60 Hz to further reduce interferences. Neutral activity measurements have consisted in 33 consecutive recording trials, leading to a total of 118 016 samples for each channel, which corresponds to about 15 minutes of neutral recording. The necessity to perform consecutive and split trials has emerged in order to maintain the requested concentration and avoid artifacts as eye-blinks or head movements. During this stage, the thoughts were of not particular attention. The push activity measurements have consisted in 62 recording trials, taking to a total of 123 136 samples, which amounts to 16 minutes of push task recording. The number of trial of this second activity is greater due to the less mean duration of each measurement, because of the difficulty to concentrate on the task. During this stage, the thoughts were related to push away a wooden box with both hands. The total number of data collected amounted to 31 minutes. Manual inspection of the recording has shown that, despite the precautions taken, some artifacts were present in the data. These nuisances have been therefore removed prior to any other processing task. This operation has led to 11 minutes of neutral recording and 15 minutes of push task recording. The quite unbalance between the two classes has been taken in consideration in the choice of the classifier. After the acquisition phase, the data were filtered with a 2-order Butterworth high-pass filter,

centered at 0.6 Hz, in order to completely remove the DC component.

3.2 Feature Extraction

The next step is the feature extraction phase. For a detailed list of methods used in feature extraction for BCIs, see McFarland et al. (2006). Features have been extracted by means of sliding windows, with window length of 1 s and window overlapping time of 0.5 s. These times have been chosen thinking about a compromise between the number of data available for each window (the higher the better), and the response time of the classification algorithm (the shorter the better). Then, for every window of each channel, the spectral power in the frequency bands [0.5 – 3.9 Hz], [4 – 7.9 Hz], [8 – 12.9 Hz], [13 – 30.9 Hz] and [31 – 43 Hz] has been computed. These bands correspond to the brain rhythms cited in Sect. 1. By doing this, a total of 1312 neutral activity and 1794 push activity segments were obtained. By extracting 5 frequency features for 14 channels for each segment, this leads to a 3106×70 data matrix.

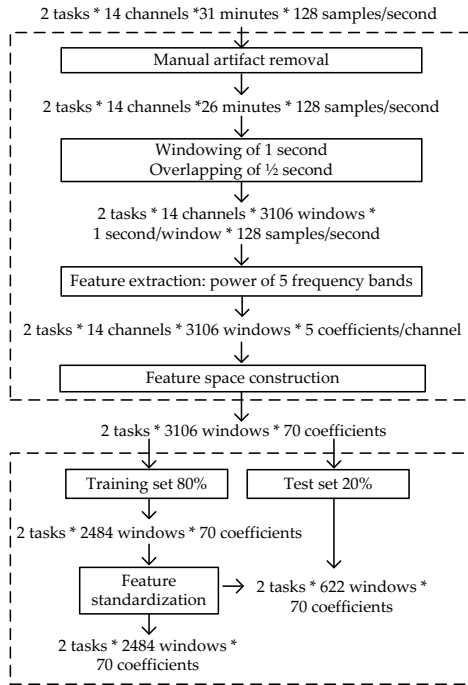


Fig. 2. Sequence diagram for the BCI classification algorithm design

The data have been then divided in training and test set, with respective proportions 80% and 20%. This has led to a number of training points equal to 2484, and 622 test points. The last step of data pre-processing has regarded the standardization of the features to zero mean and unit variance. The mean and variance were computed *only* on training data, and then the transformation was applied to the training data *and* the test data. The whole process is depicted in Fig. 2 to better clarify the methodological steps.

4. CLASSIFIERS COMPARISON

In this section, several classifiers for the BCI system development have been compared. Supported by the results, the

chosen algorithm for this task is a logistic regression with a Lasso penalizing term ($L1$ -regularization). The analysis has shown that using different types of classifiers could shed new lights and reveal new insights: it's the case of Decision Trees, which can be used to indirectly assess the importance of the features. In this view, it was possible to link the most relevant descriptors with the brain areas used during the training phase.

4.1 Logistic Regression

Logistic regression, Hastie et al. (2009), is one of the most known methods in machine learning and represents, despite its name, a binary linear classification algorithm. Considering the typical form of a linear model:

$$s = \sum_{i=1}^d w_i x_i \quad (1)$$

with $s \in \mathfrak{R}$ the model output and $\mathbf{w}, \mathbf{x} \in \mathfrak{R}^d$ representing the model coefficients and regressors in the d -dimensional space, the hypothesis $h(\mathbf{x}) = \theta(s)$ is learned, where $\theta(\cdot)$ represents the logistic function, such that:

$$\theta(s) = \frac{e^s}{1 + e^s} \quad (2)$$

The output of a logistic regression is therefore bounded between $[0 - 1]$, and represents a genuine probability that a point \mathbf{x} will belong to one of the two classes. If this probability is greater than 0.5 (the decision threshold), the point will be classified as belonging to class 1, otherwise to class 2. The algorithm configuration chosen is a logistic regression with a $L1$ -penalty term and regularization coefficient $C = 0.3$. Matching the model complexity depending on the number of training examples is crucial for good generalization performances (Abu-Mostafa et al. (2012)). Being that the training vectors belong to a 70-dimensional space, the VC -dimension (d_{VC}) of a logistic regression classifier will be $d_{VC} = 71$. Considering the rule of thumb of having a number of training data at least $10 \cdot d_{VC}$ in order to ensure good generalization, it turns out that this condition is met, having 2484 training points. The choice of using a $L1$ -regularization is based on the will to further reduce the d_{VC} of the classifier. This type of regularization is in fact known to produce sparse output with respect to a $L2$ -regularization, which tends to produce small weights. This type of regularization was also tried, but it has not given better results. The degree of regularization is controlled by the parameter C . This coefficient has been chosen via a 10-fold cross-validation: the choice is motivated accounting for the balance between the number of training and validation data which will be used by the routine; furthermore, this type of operation makes no assumptions about the data. The cost function used during the cross-validation was the $F1$ -score. This, as opposed to *accuracy*, better represents the case of unbalanced dataset, which is slightly this case. The final model, trained in 0.20 s, consists of a 65 non-null coefficients, with a training performance of 86.95% and a test performance of 86.01%. The learning curves of the algorithm can be observed in Fig. 3. They have been obtained by using incremental training set sizes, from 10% to 100% of the total training set data, with steps of 10%. Then, for each training set size, a 10-fold cross-validation

has been carried out training the previous selected $L1$ -classifier with $C = 0.3$. The mean cross-validation error represents the graph points, while the standard deviation is depicted by the dashed areas. As it can be noticed, the classifier is not suffering from high variance since the two curves converge quite fast as the number of training data grow up. Furthermore, the bias term is small, leading to an expected error of about 14%.

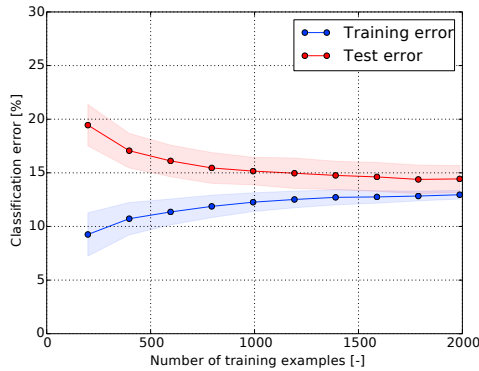


Fig. 3. Logistic regression learning curves

By looking at the confusion matrix of this classifier, reported in Table 1, it is possible to observe that the number of misclassifications is more unbalanced in the neutral class, with respect to the push class. The advantage of the logistic regression is that it directly provides a probabilistic output: by modifying the decision threshold for the classification, is possible to deal with this problem. The new threshold has been found via a 10-fold cross-validation to be 0.41. The new confusion matrix is shown in Table 2: the number of misclassified points has been reduced, the training and testing accuracy improved to 87.00% and 86.66%.

Table 1. Logistic regression confusion matrix

		Predicted class	
		Neutral	Push
True class	Neutral	203	46
	Push	41	332

Table 2. Logistic regression with tuned threshold confusion matrix

		Predicted class	
		Neutral	Push
True class	Neutral	219	30
	Push	53	320

4.2 Support Vector Machines

Support Vector Machines (SVMs) belongs to the class of binary maximum margin classifiers. The aim is to find a linear classifier which separates the data, but not just any: the chosen boundary is the one that leaves more “space” between the nearest data points of the two classes. This has implication on the VC -dimension, because the effect is that of shrinking the hypothesis space, Abu-Mostafa et al.

(2012). The optimal hyperplane is then found through an optimization problem of the form:

$$\begin{aligned} & \text{Minimize} && \frac{1}{2} \mathbf{w} \mathbf{w}^T + C \sum_{n=1}^N \xi_n \\ & \text{subject to} && y_n(\mathbf{w} \mathbf{x}_n + b) \geq 1 - \xi_n \quad n = 1, \dots, N \\ & && \xi_n \geq 0 \quad n = 1, \dots, N \end{aligned} \quad (3)$$

with $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ the hyperplane weight vector and bias, and $\xi \in \mathbb{R}^N$ the slack variables introduced to cope with non-linear separable datasets, given N the number of training points. The coefficient $C \in \mathbb{R}$ has a role similar to the regularization parameter of the logistic regression. But it’s only when equipped with a powerful mathematical tool called *kernel* that SVM can unleash all their power: it’s indeed possible to exploit the expressiveness of non-linear transformation, paying for such a big dimensionality only in terms of the number of *support vectors*, that is, training points for which the corresponding Lagrange multiplier is not null, and “support” the decision boundary. The trained classifier adopted has been a Radial Basis Function (RBF) kernel, with RBF parameter $\gamma = 0.01$ and $C = 1$, chosen via 10-fold cross-validation. The training time was 1.07s when probability estimates have not been enabled: in this case, the training time grows up to 5.19s. SVMs does not directly output a probability as logistic regression, but an additional procedure, known as Platt’s scaling, has to be carried out. The accuracy of the non-linear model is 92.15% on training data and 87.78% on test data, resulting a bit better than logistic regression. The number of support vectors was 70. Having to solve a quadratic optimization problem, the SVM algorithm can take longer times than logistic regression. The learning curve, reported in Fig. 4, shows a very lighth variance problem: more training data could help to obtain an even better generalization error. Despite their power, SVMs does not improve much the logistic regression model, requiring also more training time.

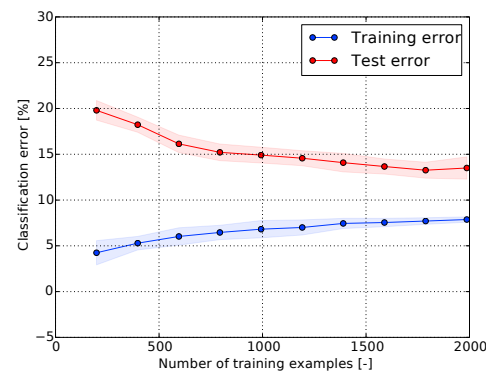


Fig. 4. Support Vector Machines learning curves

4.3 Naive Bayes

The Naive Bayes algorithm is a probabilistic classifier which derives from the optimal Bayesian one. Let ω_1 and ω_2 be the two classes in which the patterns belong, and $P(\omega_1)$ and $P(\omega_2)$ the a-priori class probabilities. Being $p(\mathbf{x}|\omega_m)$, $m = 1, 2$, the likelihood of ω_m with respect to \mathbf{x} , it’s possible to use the Bayes rule to obtain:

$$p(\omega_m|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_m)p(\omega_m)}{p(\mathbf{x})} \quad m = 1, 2 \quad (4)$$

The classification rule will be then:

$$\begin{aligned} \text{if } p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x}) \quad \mathbf{x} \in \omega_1 \\ \text{if } p(\omega_1|\mathbf{x}) < p(\omega_2|\mathbf{x}) \quad \mathbf{x} \in \omega_2 \end{aligned} \quad (5)$$

where $p(\mathbf{x})$ can be neglected because it is the same in each class. If the likelihoods are not known, they have to be estimated. This can be a problem in very high data dimensionality like BCI datasets. To simplify this computation, the Naive Bayes classifier assumes that the features are independent, so that:

$$p(\mathbf{x}|\omega_m) = \prod_{i=1}^d p(x_i|\omega_m)p(\omega_m) \quad m = 1, 2 \quad (6)$$

The chosen model is a Gaussian-likelihood Naive Bayes, because its ease of training and Gaussian distribution has been thought to better represent the BCI features, with respect to Bernoulli or Multinomial likelihoods. The classifier, trained in 0.15 s, has given a training classification score of 73.87% and a testing accuracy of 72.02%. The learning curve depicted in Fig. 5 shows that the Naive Bayes is a good algorithm because it does not suffer of variance problems, even if it is a bit too simple; an advantage is that it does not need parameters to be tuned, but the underline distribution assumptions may not hold always.

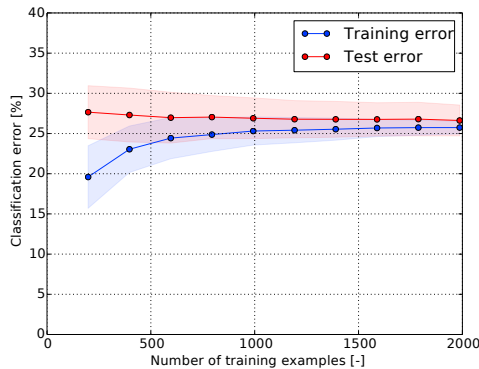


Fig. 5. Naive Bayes learning curves

4.4 *k*-Nearest Neighbour

The *k*-Nearest Neighbour (*k*-NN) is a non-parametric instance-based algorithm which classifies a point by first considering the *k* nearest vectors to that point. An advantage of this rule is that it is simple and intuitive, easy to implement and there is no “training phase”. The main disadvantage is the computational overhead. It can be shown that the VC-dimension of the classifier is infinite (Duda et al. (1973)). Since there are 2 classes, it is convenient to choose an odd number of neighbours to avoid indecisions on which class to classify the points. Therefore, via a 10-fold cross validation, a 17-NN algorithm has been chosen, with euclidean distance. The computation of distances has been done with the “Ball Tree” algorithm which has a computational time of $O(d \log N)$, leading to a query time on the test set of 0.61 s. The accuracies of the 17-NN were 86.79% and 84.56% on training and test sets. The learning curves in Fig. 6 shows that more points could improve the results, but this would lead to a higher response time.

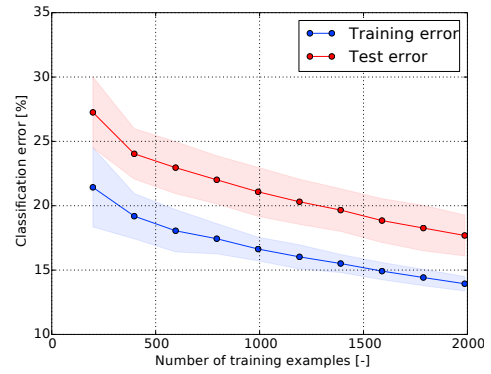


Fig. 6. 17-Nearest Neighbour learning curves

4.5 Decision Tree

Decision Trees divide the feature space into regions via a greedy approach, known as recursive binary splitting, where each region is a leaf of the tree. For classification trees, the predicted class, for each observation, is given by the most commonly occurring class of training observations that belong to the same terminal node. In order to define the binary split, usually the Gini index is used:

$$G = \sum_{m=1}^M \hat{p}_{rm}(1 - \hat{p}_{rm}) \quad (7)$$

where \hat{p}_{rm} represents the proportion of training observations in the *r*-th region that are from the *m*-th class, with *M* the number of classes. A small value of *G* indicates that a node contains predominantly observations from a single class. Decision trees are highly interpretable: indeed they can be easily visualized and explained. However, they tend to overfit: to overcome this, the tree has been trained by limiting the maximum tree depth to 5 levels, imposing a minimum of 6 samples per leaf and at least 2 samples required for splitting a node. These parameters have been chosen by a 10-fold cross-validation. The training score obtained with the decision tree was 81.73%, while the test score 74.11%. The training time was 0.38 s. The training curve in Fig. 7 shows that the

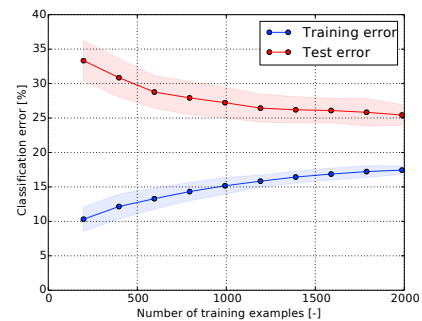


Fig. 7. Decision Tree learning curves

classifier has not the expressiveness of the logistic regression or SVM, and more examples are needed to improve it. Decision trees however produce an automatic feature selection: the more often a feature is used in the split points the more important that feature is: the tree has selected 17 out of 70 features to be informative. By considering that a feature number corresponds to a specific frequency band power detected by a specific sensor, and representing

it as *Frequency Band Power_{Channel Number}*, the features extracted in this way are, in decreasing ranking order: $[\vartheta_{13}, \delta_{11}, \vartheta_{12}, \alpha_{13}, \beta_{10}, \beta_4, \alpha_{10}, \gamma_{13}, \alpha_7, \gamma_{10}, \beta_1, \gamma_{12}, \vartheta_{11}, \delta_{10}, \beta_{13}, \gamma_4, \alpha_8]$. It's then possible to construct a map of the brain areas which links the most important features with the sensor from which they have been extracted. The map

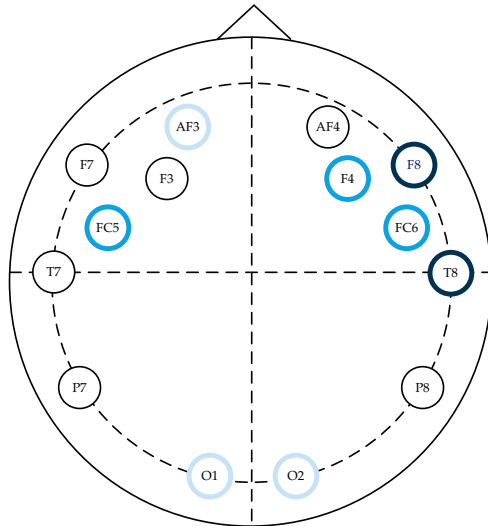


Fig. 8. Channels involved in discerning neutral vs. push activities. The channels which corresponds to a major number of representative features are depicted with a darker color

in Fig. 8 shows that the frontal area of the brain is more involved during the training process, due to the presence of the premotory and motory cortex, which can be related to the “pushing” of and object with the arms, thought during the training. The visual activities are captured lightly by the occipital sensors O1-O2. Furthermore, the most important features are related to the α , β , and γ rhythms (each of them has 4 representants in the 17 features selected by the tree), which are linked to the sensorimotor activity, see Bashashati et al. (2007). The δ and ϑ waves have respectively 2 and 3 exponents.

5. CONCLUSIONS AND FUTURE DEVELOPMENTS

In this paper, the use of Brain Computer Interface has been introduced in the field of rehabilitation from craving diseases. By actively using the instrument to repel illness-related pictures, the patient can push away even its craving desire. In order to recognize when the subject is trying to do this, a machine learning algorithm is mandatory. Several types of classifiers, each one with a different “way of thinking”, have been compared offline. The result is shown in Table 3. The chosen classifier has been a $L1$ -regularized logistic regression, for its accuracy, training speed, and adaptivity to deal with false positives and true negatives. This last property means that the algorithm can be easily tweaked, for example by varying the decision threshold. Interpretability means that the algorithm decisions and results can be easily explained to a potential customer. A lighter training time is beneficial for clinical purposes, so the patients are not stressed by this routine. By using the feature selection capability of the decision tree classifier, a map of the brain areas involved and which differentiates the two mental tasks is given. Future

Table 3. Classifier comparison summary

Classifier	Accuracy	Training time	Interpretability	Adaptivity
Logistic Regression	✓	✓		✓
Support Vector Machines	✓			
Naive Bayes		✓		✓
K-Nearest Neighbours	✓		✓	
Decision Tree		✓	✓	

developments consists of training the classifier for more than two classes and deploying it in an online environment, while collecting data from more than one subject will help to derive more sound conclusions.

REFERENCES

- Abu-Mostafa, Y.S., Magdon-Ismael, M., and Lin, H.T. (2012). *Learning from data*. AMLBook.
- Bashashati, A., Fatourech, M., Ward, R.K., and Birch, G.E. (2007). A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural engineering*, 4(2), R32.
- Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., and Flor, H. (1999). A spelling device for the paralysed. *Nature*, 398(6725), 297–298.
- Duda, R.O., Hart, P.E., et al. (1973). *Pattern classification and scene analysis*, volume 3. Wiley New York.
- Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.
- Hoffmann, U., Vesin, J.M., and Ebrahimi, T. (2007). Recent advances in brain-computer interfaces. In *IEEE International Workshop on Multimedia Signal Processing (MMSP07)*, LTS-CONF-2007-063.
- Jensen, A.R. and Rohwer, W.D. (1966). The stroop color-word test: A review. *Acta psychologica*, 25, 36–93.
- Kozlowski, L.T. and Wilkinson, D.A. (1987). Use and misuse of the concept of craving by alcohol, tobacco, and drug researchers. *British journal of addiction*, 82(1), 31–36.
- Littleton, J. (1995). Acamprosate in alcohol dependence: how does it work? *Addiction*, 90(9), 1179–1188.
- McFarland, D.J., Anderson, C.W., Muller, K., Schlogl, A., and Krusienski, D.J. (2006). Bci meeting 2005-workshop on bci signal processing: feature extraction and translation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2), 135.
- Modell, J.G., Glaser, F.B., Cyr, L., and Mountz, J.M. (1992). Obsessive and compulsive characteristics of craving for alcohol in alcohol abuse and dependence. *Alcoholism: Clinical and Experimental Research*, 16(2), 272–274.
- Sitharthan, T., McGRATH, D., Sitharthan, G., and Saunders, J.B. (1992). Meaning of craving in research on addiction. *Psychological reports*, 71(3), 823–826.
- Sutton, S., Braren, M., Zubin, J., and John, E. (1965). Evoked-potential correlates of stimulus uncertainty. *Science*, 150(3700), 1187–1188.