Assessing the Performance of a BCI: A Task-Oriented Approach

B. Dal Seno[1], L. Mainardi[2], M. Matteucci[1]

[1]Department of Electronics and Information, IIT-Unit, Politecnico di Milano, Italy
[2]Department of Bioengineering, IIT-Unit, Politecnico di Milano, Italy

bernardo.dalseno@polimi.it

Abstract

An accurate way to measure the performance of a brain-computer interface (BCI) is important to compare different analysis methods and different protocols. The decision of which BCI and what parameters to use should take into consideration the expected performance. Information transfer rate has been proposed as a benchmark, but existing information-based metrics measure the channel capacity of the BCI classifier, which may be much higher than what a BCI achieves in practice. Therefore, we introduce a novel task-oriented approach to the measuring of BCI performances, which takes into account how all the components of the BCI and the user interact. We apply it to the case of a P300 speller and show how the information transfer rate may be misleading. Moreover, we determine when the introduction of an automatic error-correction method is advantageous for a given user. This shows that our approach can be used to compare BCI variants.

1 Introduction

A brain-computer interface (BCI) [1] is an interface that does not entail muscle movements, but it bypasses any muscle or nerve mediation and connects a computer directly with the brain by picking up the brain activity signals. This simple definition hides two relevant issues: how fast it is possible to communicate with such an interface, and how often errors are made. These issues are obviously related since the communication speed highly depends on the error rate of the BCI.

An accurate way to measure a BCI performance is important to compare different analysis methods and different protocols. If many BCIs or many variants are available to a user, the decision of which one to use should take into consideration the expected performance for that user (other aspects, which are important, are not the focus of this work). To this aim, the measurement of the performance should be tailored to each specific BCI. For example, there are ongoing studies to introduce automatic correction systems in BCIs [2] based on error potentials (ErrPs), which are specific variations in the EEG induced by the subjective recognition of a committed mistake [3]. Such systems may have false positives, and hence introduce new errors; so, the evaluation of the opportunity of their introduction in a BCI requires the estimation of the potential improvement in the performance of the particular BCI. Intuitively, if the reliability of the ErrP detector is poor, the introduction of ErrP-based corrections will cause more damages than benefits.

In the literature, the performance of a BCI has been quantified by using different metrics, such as classification accuracy, information transfer rate, letters or words per minute, kappa statistic, and others. Among them, the *information transfer rate* (sometimes simply called *bit rate*) has been proposed as a benchmark for the evaluation of BCI performances because it does not depend on any particular protocol, it takes into account both the number of choices and time needed, and it could be applied also to continuous ranges of choices [4]. A formula for the information transfer rate is derived in [5] to compute the (mean) number of bits transferred per trial:

$$B = \log_2 N + p \log_2 p + (1-p) \log_2 \frac{1-p}{N-1} \,, \tag{1}$$

where N is the number of possible choices per trial, and p is the accuracy of the BCI, i.e., the probability that the BCI selects what the user intends. Equation (1) divided by the trial duration gives the mean number of bits transferred per time unit. This formula is derived from Shannon's theory [6], and it represents a measure of the *mutual information* between the user's choice and the BCI selection, under the assumption that all choices convey the same amount of information (i.e., they are chosen by the user with equal probability), p is the same for all the possible choices, and that all the wrong choices have the same probability in case of error. In other words, a BCI system is seen as a noisy channel, in which the selection of a wrong option is the noise.

According to Shannon's noisy channel coding theorem [6], it is possible to achieve an arbitrarily small error probability in a communication on a noisy channel as long as the information transfer rate does not go beyond a certain limit (the *channel capacity*). The channel capacity is given by the mutual information, and this seems to justify the use of mutual information in Equation (1). The only problem is that Shannon proved his famous theorem by transferring information embedded in ever increasing blocks of bits, and in telecommunication practice only very complex error correction schemes have permitted to get near Shannon's limit. For a BCI, where a human subject sits at one end of the noisy channel, it is not possible to implement such complex error correction schemes, and thus the limit given by the mutual information, as in Equation (1), represents a theoretical figure, unreachable by any real BCI whose error rate is significantly different from zero.

To better understand how far from practice can be the performance measure of Equation (1), let us focus on a simple example with a standard P300 speller [7], where the P300 is used to select letters in matrix of 36 symbols. Let us suppose that the speller speed is 4 letters per minute, and that a user achieves a performance of 45% in accuracy, which is low, but still far better than random-level accuracy (2.7%). By substituting $p = 45\%$ and $N = 36$ in Equation (1), we get $B = 1.36$ bits, and the information transfer rate for this user would be $4 \cdot B = 5.4$ bits/min; this is not very fast, but, still, communication should be possible. Now let us look at the practical use of such a BCI: the most natural way would be to move on to the next letter when the speller gets one right, and to "hit" backspace every time the speller is wrong. What is the real transfer rate for this speller? Since every letter is more likely to be wrong than not, and this happens to backspace as well, the expected time to spell a letter correctly is infinite, and thus the answer is, on average, exactly 0 bits/min (see also the derivation of Equation (7) in the next section). While it could be still possible to raise p above .5 by increasing the number of stimulations per letter and render the P300 speller of the example usable, the channel capacity measured by Equation (1) promises a performance far beyond what is attainable once the details of the BCI are taken into account. For this reason, we propose that the measure of the performance of a BCI, e.g., the information transfer rate, takes into account not only the behavior of the classifier contained in the BCI, but how all the components of the BCI and the user interact to perform the task the BCI is designed for.

There exists a generalization of formula (1) that makes use of the confusion matrix and allows each letter to have a different probability of occurrence and a different accuracy [8], but such formula has the same shortcoming of (1), i.e., it treats the BCI as a communication channel with no reference to the way the channel is actually used, and it has a huge number of parameters. In order to keep the exposition simpler, we have chosen to limit the discussion to the simplified formula. All our considerations can be easily extended to the general formula as well.

In the next section we show how the idea of a task-oriented approach to performance measurement can be applied to a P300 speller, and we compare it to Equation (1). In Section 3 we show how our approach can be used to evaluate the opportunity of introducing ErrP detection in the P300 speller. Some concluding remarks follow in Section 4.

2 Task-Related Performance Measurement

We give an example of a task-related performance measurement by deriving the performance of a P300 speller, based on the computation of the expected time t_{L} required to spell a letter correctly.

As explained above, we use the assumptions of Equation (1) to keep the exposition simpler;

moreover, we assume that the accuracy of the speller p is constant and the system has no memory, i.e., each trial is not influenced by the result of the previous one. If c is the time duration of every single trial, the expected time to correctly spell a letter is

$$t_{\mathrm{L}} = p \cdot c + (1-p) \cdot (c + t_{\mathrm{B}} + t_{\mathrm{L}}^{(1)}) = c + (1-p) \cdot (t_{\mathrm{B}} + t_{\mathrm{L}}^{(1)})\,, \tag{2}$$

where the term $p \cdot c$ is the contribution of the case where the letter is correctly spelled at the first attempt, while the second term represents the case where the letter is wrong, so a backspace must be entered (which takes t_{B} time) and the letter respelled ($t_{\mathrm{L}}^{(1)}$ time). As the system is stationary, we obviously have $t_{\mathrm{L}}^{(1)} = t_{\mathrm{L}}$. In addition, as the backspace should be treated as any other symbol, $t_{\mathrm{B}} = t_{\mathrm{L}}$ (this can be derived formally). We can rewrite Equation (2) in an iterative formulation

$$t_{\mathrm{L}} = c + 2 \cdot (1-p) \cdot t_{\mathrm{L}}\,, \tag{3}$$

which leads to

$$t_{\mathrm{L}} = \frac{c}{2p-1}\,. \tag{4}$$

This relationship is valid only when $2p - 1 > 0$, i.e., $p > 0.5$; when $p \leq 0.5$, it should be apparent from the expanding of (2) or (3) that the expected time to correctly spell a letter goes to infinite.

Using Equation (4), we can compute the information transfer rate for our P300 speller. This can be obtained as the ratio between the information contained in an ever growing number of symbols spelled and the time taken to spell them:

$$I_{\mathrm{R}} = \mathrm{E}\left[\lim_{K\to\infty} \frac{b \cdot K}{\sum_{i=1}^{K} c \cdot n_i}\right]\,, \tag{5}$$

where b is the information content (in bits) of one spelled symbol, and n_i is the number of trials needed to spell correctly the i-th symbol. Since

$$\lim_{K\to\infty} \frac{\sum_{i=1}^{K} c \cdot n_i}{K} = \mathrm{E}[c \cdot n] = t_{\mathrm{L}}\,, \tag{6}$$

and it holds $b = log_2(N-1)$ bits, Equation (5) can be rewritten as

$$I_{\mathrm{R}} = \frac{b}{t_{\mathrm{L}}} = \frac{(2p-1) \cdot \log_2(N-1)}{c}\,. \tag{7}$$

Only $N - 1$ symbols can appear in real words (the backspace cannot), and we are measuring the information contained in the spelled text, therefore the assumption of equal probability leads to the value of $log_2(N-1)$ bits.

This expression represents the expected performance of our speller, and we can compare it with the theoretical limit derived from Equation (1):

$$I_{\mathrm{T}} = \frac{B}{c} = \frac{\log_2 N + p\log_2 p + (1-p)\log_2 \frac{1-p}{N-1}}{c}\,. \tag{8}$$

Figure 1 compares the two measures of information transfer rates and shows how the effective performances can be far from the theory. In fact, while the (8) measures the *capacity of a channel*, i.e., the maximum performance obtainable by a noisy channel, the (7) measures the expected performance of the same channel when information is conveyed in a specific way; in our case, this is the natural way of using a P300 speller. As expected, the latter curve lies always below the theoretical limit, and it is equal to zero when the accuracy is too low. For high accuracy values, the two curves almost coincide.

It is worth noting that the graph may evidence regions in where the channel cannot work (when $p \leq 0.5$, in our case) and also areas where differences are very far from the theoretical limit.

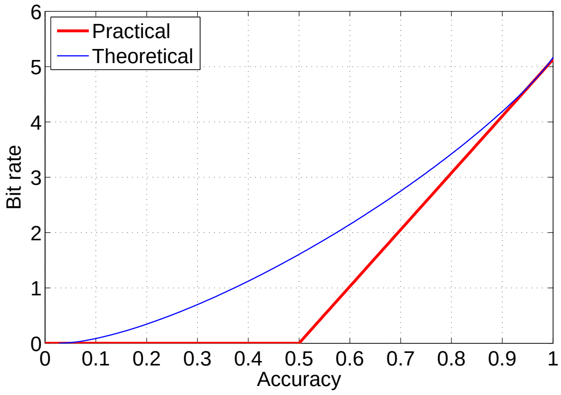


Figure 1: Comparison between theoretical and practical information transfer rate for a P300 speller with 36 symbols.

3 Performance Measurement with Error Detection

We now make use of the previous result to measure the improvement of the performance gained when an error-correction capability (based on ErrP detection) is added to our P300 speller. With the new feature added, the speller works as follows: it selects a letter by means of P300 detection and displays it on the screen; if an error is recognized, the latest letter is canceled, while if no error is detected, the latest letter is kept.

To derive the performance of this new system, we need to define the performance of the error-correction system. We characterized it with two parameters: 1. the recall for errors (r_E, the fraction of times that an actual error is recognized by the error classifier), and 2. the recall for correct trials (r_C, the fraction of times that a correctly spelled letter is recognized by the error classifier). We assume that r_E and r_C are constant and do not depend on the actual letter.

For each trial, four possible cases can happen, which are listed in Table 1. Each case occurs with a certain probability, and the expected time to correctly spell a letter obviously varies case by case. Both the probability and the expected time are reported in the table. The expected time to spell a letter correctly can be computed as in Equation (2) by summing the time required by each case weighted by their respective probabilities:

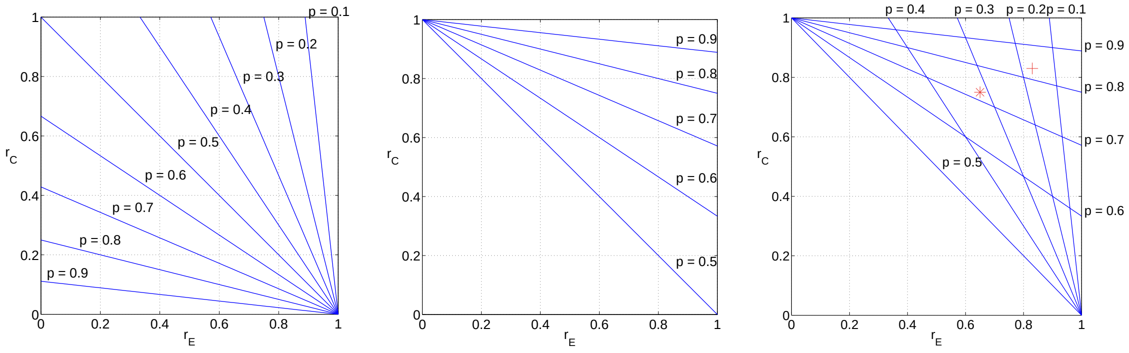
$$\begin{aligned} t_\mathrm{L} &= p_1 \cdot c + p_2 \cdot (c + t_\mathrm{B} + t_\mathrm{L}) + p_3 \cdot (c + t_\mathrm{L}) + p_4 \cdot (c + t_\mathrm{L}) \\ &= p \cdot r_\mathrm{C} \cdot c + (1-p) \cdot (1-r_\mathrm{E}) \cdot (c + t_\mathrm{B} + t_\mathrm{L}) + p \cdot (1-r_\mathrm{C}) \cdot (c + t_\mathrm{L}) + (1-p) \cdot r_\mathrm{E} \cdot (c + t_\mathrm{L}), \end{aligned} \tag{9}$$

where c is the constant duration of a trial, as before. Reasoning as in the previous section, we obtain

$$t_\mathrm{L} = t_\mathrm{B} = \frac{c}{p \cdot r_\mathrm{C} + (1-p) \cdot r_\mathrm{E} + p - 1}, \tag{10}$$

Event	Probability	Expected time
The P300 speller selects the correct letter, and the ErrP classifier correctly recognizes it.	$p_1 = p \cdot r_\mathrm{C}$	c
The P300 speller selects the wrong letter, and the ErrP classifier does not recognizes the error.	$p_2 = (1-p) \cdot (1-r_\mathrm{E})$	$c + t_\mathrm{B} + t_\mathrm{L}$
The P300 speller selects the correct letter, and the ErrP classifier wrongly detects an error.	$p_3 = p \cdot (1-r_\mathrm{C})$	$c + t_\mathrm{L}$
The P300 speller selects a wrong letter, and the ErrP classifier recognizes the error.	$p_4 = (1-p) \cdot r_\mathrm{E}$	$c + t_\mathrm{L}$

Table 1: Probabilities and expected times for the four possible outcomes of a trial



(a) (b) (c)

Figure 2: (a) Condition for the usability of a P300 speller with ErrP detection. (b) Comparison between two P300 speller with and without ErrP detection. (c) When ErrP detection improves the performance of a P300 speller.

where the result is valid only if the denominator is positive, i.e., when

$$r_{\mathrm{C}} > \frac{1-p}{p}\,(1-r_{\mathrm{E}})\,. \tag{11}$$

Figure 2.a shows the boundaries defined by Inequality (11) for different values of p; the inequality is satisfied for the points lying above the lines, and only in these cases the time for spelling a letter is finite (i.e., the P300 speller can be useful). It can be noticed that the constraint becomes tighter as p diminishes, with recall of errors becoming more and more important.

If we now compare Equation (4) with (10), we can evaluate when the error-detection system gives any improvement to the P300 speller. In order to have an improvement, the expected time, t_{L}, should be lower when error detection is used; this leads to the inequality

$$r_{\mathrm{C}} > \frac{p-1}{p}\,r_{\mathrm{E}} + 1\,. \tag{12}$$

Figure 2.b shows the boundaries defined by Inequality (12) for different values of p (for $p < 0.5$ the comparison has no sense); points above the lines represent values of r_{C} and r_{E} for which ErrP detection is advantageous. In this case, as p grows the area defined by Inequality (12) shrinks; this happens, because as p grows the performance of the P300 speller gets better and better, and it becomes harder and harder for the ErrP classifier to improve the speller performance.

Figure 2.c summarizes the first two graphs in Figure 2, and shows the values of r_{C} and r_{E} for which ErrP detection is advantageous for the whole range of p. As before, the part of the plane above the lines is the useful part; values below the lines are either useless or counterproductive. Figure 2.c can be used as a guide to decide to bias the ErrP classifier either toward correct or erroneous epochs, depending on the value of p.

Some practical examples may help to better understand the above ideas. Let say that for a particular user the P300 speller reaches 90% accuracy without error correction, and the error detection reaches $r_{\mathrm{C}} = r_{\mathrm{E}} = 83\%$. This situation corresponds to the cross in Figure 2.c, and the cross lies below the line $p = 90\%$. So, for this particular user the automatic error correction system is counterproductive. Another user's performance may be expressed by $p = 70\%$, $r_{\mathrm{E}} = 65\%$, $r_{\mathrm{C}} = 75\%$ (the asterisk in Figure 2.c); the asterisk lies above the line $p = 70\%$, and therefore the automatic error correction system should help this user.

4 Conclusions

We have shown that the measure of the information transfer rate of a BCI, intended as the channel capacity of the BCI classifier, can be highly misleading. We have proposed a task-oriented approach that takes into account how all the components of the BCI and the user interact, and we have applied it to a P300-based speller. We have compared the measure obtained with our approach to a formula for the information transfer rate that relies on a few simplifying assumptions; it is been possible to use a more general formula with less assumptions and reach the same conclusions, because the main shortcoming lies in the measuring of the channel capacity of the BCI classifier and not in the simplifying assumptions used.

The results for the P300 speller have been extended to derive a formula for a speller with an ErrP-based correction system added, and the formula has been used to assess the utility of the correction system. The use of a task-oriented model permits to make firm observations about the usefulness of ErrPs in a P300 speller, and to identify operating regions/settings in which a real improvement can be obtained. Using a model strictly related to the task performed by the BCI under study is fundamental to understand and quantify the real impact of variations of a BCI protocol, like the introduction of ErrP-based corrections. While we have applied the proposed approach to two specific cases, it is possible to use the same approach to study other kinds of BCIs and the impact of the modification of other parameters, and we think this should lead to a better comparison between different protocols.

Acknowledgments

This work has been partially supported by the Italian Institute of Technology (IIT), and by the grant "Brain-Computer Interfaces in Everyday Applications" from Politecnico di Milano and Regione Lombardia.

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*, 113:767–791, 2002.

[2] P. W. Ferrez and J. d. R. Millán. Error-related EEG potentials generated during simulated brain-computer interaction. *IEEE Transactions on Biomedical Engineering*, 55(3):923–929, 2008.

[3] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke. Effects of crossmodal divided attention on late ERP components. II. Error processing in choice reaction tasks. *Electroencephalography and Clinical Neurophysiology*, 78(6):447–455, 1991.

[4] A. Schlögl, C. Keinrath, R. Scherer, and G. Pfurtscheller. Information transfer of an EEG-based brain computer interface. In *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, pages 641–644, 2003.

[5] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, G. S. P. Hunter Peckham, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan. Brain-computer interface technology: A review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, 8(2):164–173, 2000.

[6] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* The University of Illinois Press, 1949.

[7] L. A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 6(70):510–523, 1988.

[8] A. Schlögl, J. Kronegg, J. E. Huggins, and S. G. Mason. *Towards Brain-Computer Interfacing*, chapter Evaluation Criteria for BCI Research, pages 327–342. MIT Press, 2007.