

A Performance Model of Selection Techniques for P300-Based Brain-Computer Interfaces

Jean-Baptiste Sauvan
INRIA/Supélec
jsauvan@irisa.fr

Anatole Lécuyer
INRIA
alecuyer@irisa.fr

Fabien Lotte
INSA/IRISA
flotte@irisa.fr

Géry Casiez
INRIA/University of Lille 1
gery.casiez@lifl.fr

ABSTRACT

In this paper, we propose a model to predict the performance of selection techniques using Brain-Computer Interfaces based on P300 signals. This model is based on Markov theory and can compute both the time required to select a target and the number of visual flashes needed. We illustrate how to use this model with three different interaction techniques to select a target. A first experimental evaluation with three healthy participants shows a good match between the model and the experimental data.

Author Keywords

Brain-Computer Interface, Interaction Technique, P300 Evoked Potential, Markov Chains

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Theory and methods

INTRODUCTION

A Brain-Computer Interface (BCI) is a communication system which enables a person to send commands to a computer only by means of voluntary variations of her brain activity [7]. The current BCI systems rely mostly on ElectroEncephaloGraphy (EEG) to measure brain electrical signals and identify brain activity. Among the various existing BCI paradigms, the “P300 evoked potential” consists in making use of a positive waveform appearing 300ms in the EEG signals after a rare stimulus expected by the user has occurred [4].

In typical applications based on P300, several buttons or objects are displayed on screen and flash randomly. The user is instructed to count, over a time period, the times that the object he wants to select is flashing. When a P300 is detected in the EEG, the system can identify the target as the object that was flashing 300ms earlier. Since the detection of the P300 in EEG signals is currently not perfect (i.e., the accuracy of the EEG classifier remains below 100%), the flashing-counting operation is repeated several times. Signals col-

lected through these repetitions are then averaged in order to increase the confidence in the P300 detection and as such in the target selection. However, even with this averaging process, it should be noted that the BCI system still makes mistakes and can sometimes select the wrong target. The P300 signal is often used in virtual keyboard applications (“P300 speller”), in which users select alphabet letters successively [4]. The letters are displayed on a uniform grid (or matrix) with each letter flashing successively (global technique), or the letters can flash by groups (N-chotomic technique) using a decision tree with successive layers of letters displayed until the final letter is selected. The P300 approach was also used in multimedia and virtual reality applications to select or activate virtual objects [2], and to move a wheelchair [6]. Studies on the design and evaluation of P300-based interactive techniques have already been conducted [3, 1], but they mostly focus on specific aspects of visual feedback.

In this paper, we introduce a novel approach that can model a wide range of techniques for BCI based on P300 signals. Our model is based on Markov chains that can predict both the time required to perform an action and the number of flashes needed. Such a model could be useful to anticipate the efficiency of different techniques before testing them experimentally. We illustrate how to use this model with three different interaction techniques aiming at selecting a target on a map. Finally, we compare our model with experimental data obtained with three healthy participants.

A MODEL FOR P300 INTERACTION TECHNIQUES

General Markov Model

An interaction technique using P300 can be represented by a graph comprising a finite set of states, and transitions between these different states. The time can also be considered as discrete: transitions between states can only occur at times known in advance. The time between two transitions is a function of the number of flashes, the duration of these flashes, and the duration of an action (if an action is performed when a state changes). Each transition is associated with a probability which is a function of the accuracy of the EEG classifier. Thus, this probability is related to the chance of correctly interpreting the user’s choice.

Consequently, we propose to model P300-based interaction processes by using static Markov chains [5], which can be seen as probabilistic graphs. The nature of Markov chains [5] implies that the modeled interaction technique should only depend on the current state and that the associated graph

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4 - 9, 2009, Boston, Massachusetts, USA.

Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00.

should not depend on time. Current P300-based interaction techniques satisfy these conditions, making Markov chains a suitable tool for modelling such techniques.

Computation of the Model

The Markov chains can be used to compute: (1) the time required to perform an action, and (2) the corresponding number of flashes needed. Note that in practice, a high number of flashes might induce visual fatigue since the user must concentrate without blinking as flashes occur.

We define \mathbf{P} as the transition matrix associated to the Markov chain. \mathbf{P} is a square matrix whose size is equal to the number of states and where each element P_{ij} is equal to the probability to move from state i to state j . In order to build this model, we assume we know the state the user wants to reach and define this state as an absorbing state (i.e., a state without outgoing transition). We denote by \mathbf{Q} the submatrix of \mathbf{P} which contains only the rows and columns of \mathbf{P} corresponding to transient states, i.e., states different from the absorbing state. We then define the matrix \mathbf{N} by using $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$ with \mathbf{I} being the identity matrix. The ij entry N_{ij} of \mathbf{N} is the expected number of times the chain is in state number j , given that it started in state i [5]. Thus, if we know the time t_j the user has to spend in order to leave state j , the expected time T_i needed to reach the absorbing state from state i is:

$$T_i = \sum_j N_{ij} t_j + T_c \quad (1)$$

where T_c is a constant time that does not depend on the graph structure and can model any additional time required (it can be equal to 0). Similarly, if we know the number of flashes m_j needed to leave state j , M_i is the total number of times a flash occurs before the user reaches final state.

$$M_i = \sum_j N_{ij} m_j \quad (2)$$

In order to compute T_i we should compute the value of each t_j . These values depend on the number of flashes needed to perform a selection, the flash duration, the time intervals between these flashes, and the duration of the action following the selection, if any. In order to perform a selection, a flash with a duration T_f occurs for each selectable target (or group of targets): i.e., for each one of the n_j states connected to the current state. There is a break with a duration of T_w between each one of these flashes. In order to perform the averaging process mentioned in the introduction section, this set of flashes is repeated n_s times, with a break of duration T_t between each set of flashes. When all these flashes have occurred, the selection is performed as well as the associated action, if any. This action has a duration of T_a . Consequently, the time needed to move from state j to another state is:

$$t_j = n_j \times T_f \times n_s + (n_j - 1) \times T_w \times n_s + T_t \times n_s + T_a \quad (3)$$

Substituting Equation 3 into Equation 1 enables us to compute the mean time T_i needed to move from state i to the

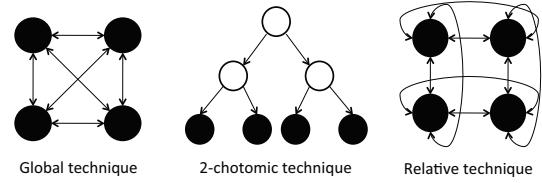


Figure 1. Illustration of the Markov graphs associated with three selection techniques for a map with four targets. Targets are displayed as black disks and circles represent groups of targets

final state using a given P300-based interaction technique. Similarly, we can also compute the number of flashes M_i by substituting into Equation 2:

$$m_j = n_j \times n_s \quad (4)$$

Building Markov Chains for Various Selection Techniques

So far, we have proposed a general model that can be used to compute the time and number of flashes needed to reach a given state for any P300-Markov chain. In this section, we illustrate how to use our model for three specific P300-based selection techniques that aim at selecting a target on a map. In this case, states represent targets or groups of targets as areas on a map. The three techniques proposed are the following:

- A *global technique*, where all the targets on the map flash alternatively in order to be directly selected. The graph associated with this technique is a totally connected graph (see Figure 1, left): from each state, we can go to any target.
- A *N-chotomic technique*, where, at each step, the targets are divided into N sub-regions. The user selects sub-regions of targets before selecting single targets. The graph associated with this technique has the shape of a well-balanced tree where each region is represented by an intermediate state. Such an intermediate state is the root of a subtree where leaves represent the targets contained in the sub-region (see Figure 1, center).
- A *relative technique*, where a current target is defined and only the closest targets around it are selectable and are alternatively flashing. After a selection, the current state becomes the selected one. As a result, the user reaches the final target by moving from one target to another. Here the right border of the map is connected to the left border, and the upper border is connected to the lower border of the map, so that each target is equivalent. Each state of this graph is connected to the four closest states of the graph and represents one of the targets where the user can move to (see Figure 1, right).

To compute the mean time needed to select any target on the map, we assume that the user plans to follow the shortest path between an initial state `initialState` and the final state `finalState` which corresponds to the target to be selected. The algorithm 1 describes how to build the Markov chain and compute performances for a given interaction technique. In this algorithm, p is the EEG classifier accuracy.

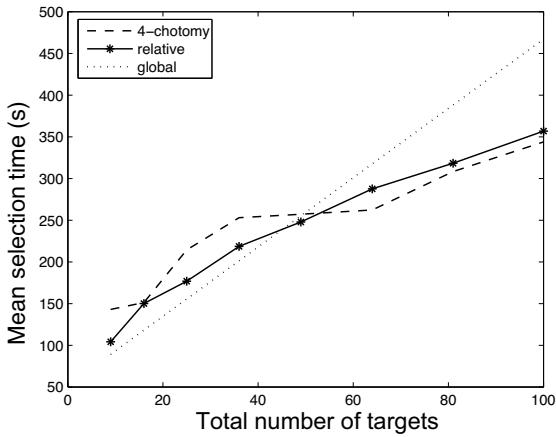


Figure 2. Time predicted by the model for the global, 4-chotomy, and relative techniques for a square map. Each point represents the mean time to select a target considering a classifier accuracy varying between 70% and 100%. The square map was chosen to simplify the simulation. The performance of relative technique is calculated as an average on all targets.

Algorithm 1: builds a Markov chain for a given P300-based interaction technique and computes its performance

Input: technique, initialState, finalState, p
Output: graph, meanTime, meanFlashNumber
graph = computeGraph(technique)
foreach state S_i in graph except finalState **do**
 shortestPath \leftarrow computeShortestPath(S_i , finalState)
 nbTransition \leftarrow getNbTransition(S_i)
 foreach transition t_i connected to S_i **do**
 if t_i in shortestPath **then**
 setProbability(t_i , p)
 end
 else
 setProbability(t_i , $(1-p)/(nbTransition-1)$)
 end
 end
end
meanTime \leftarrow computeMeanTime(initialState, finalState, graph)
meanFlashNumber \leftarrow computeMeanFlashNumber(initialState, finalState, graph)

- `computeGraph(technique)` defines the structure of the Markov chain for a given P300-based interaction technique. In other words, this function defines the states and possible transitions between these states, depending on the interaction technique used.
- `computeShortestPath(state, final state)` returns the list of transitions connected to state `state` which are on the shortest path from state `state` to state `final state`. To do so, a suitable graph theory algorithm could be used, such as Dijkstra's algorithm.
- `setProbability(transition, p)` associates the probability `p` to the transition `transition`.

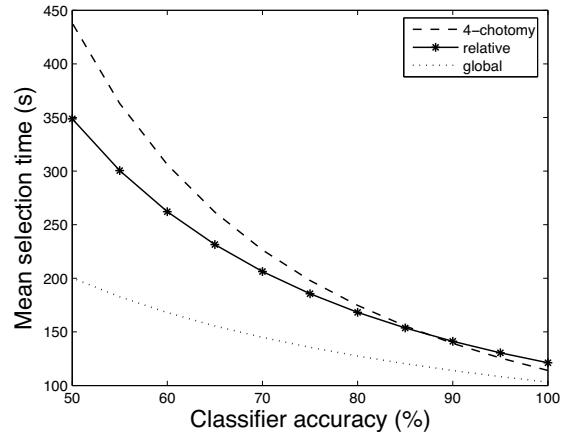


Figure 3. Time predicted by the model for the global, 4-chotomy, and relative techniques for a given square map (size 4x4) as function of BCI classifier accuracy. The performance of relative technique is calculated as an average on all targets.

- `getNbTransition(state)` returns the number of transitions connected to state `state`.
- `computeMeanTime(initialState, finalState, graph)` computes the mean time needed to go from state `initialState` to state `finalState` in graph `graph` using Equation 1.
- `computeMeanFlashNumber(initialState, finalState, graph)` computes the mean number of flashes needed to go from state `initialState` to state `finalState` in graph `graph` using Equation 2.

Examples of Predictions

With this algorithm, our model can be used to predict the efficiency of different selection techniques before testing them experimentally. As an example, Figure 2 displays the selection time as a function of the number of targets for a regular square map. The global technique is predicted to perform the best up to 50 targets. The 4-chotomy technique becomes faster for larger map sizes above 50 targets. The 4-chotomy is preferred to the 2-chotomy as our model predicts better performance. Figure 3 displays the influence of the classifier's accuracy on the selection time for the different techniques using a single 4x4 map size. The classifier accuracy can affect the difference in performance between the three techniques: low accuracy gives stronger differences across the techniques compared to high accuracy.

COMPARISON TO EXPERIMENTAL RESULTS

To test the validity of our model, we compared its predicted selection times and number of flashes with experimental data.

Design

Three people participated with a mean age of 25. The task was a two dimensional selection task, requiring participants to count successive occurrences of flashes on the desired target (or group of desired targets). There were 16 round targets (10 mm diameter) displayed on a square 4 x 4 grid on a 15

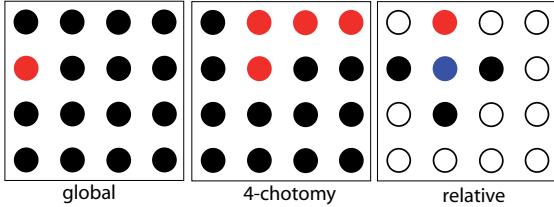


Figure 4. Display snapshots of the three techniques evaluated in the experiment. The flashing targets are in red while the current target is in blue for the relative technique.

inch LCD screen (Figure 4). Before starting a trial, the target to be selected (known by the system) was displayed in green for 3 seconds before turning black. Participants had to successfully select this target before moving to the next one. No confirmation step was used, and missed targets had to be repeated. We used a red color for flashes, and for the relative technique, the current target was highlighted in blue (Figure 4). Depending on the technique, a group of targets, or a single target, flashed 12 times ($n_s = 12$) at each step of the selection process. The 12 flashes are described with the following model parameters: $T_f = 0.1s$, $T_w = 0.2s$, $T_t = 3.0s$, $T_a = 6.0s$ and $T_c = 6.0s$. EEG signals were measured using 6 electrodes (Cz, P3, Pz, P4, O1, O2) and digitized at 512Hz. The independent variable was Technique (global, 4-chotomy, relative – with two paths: 1 and 3 steps–). The two dependent variables measured were the selection time and number of flashes. Each technique was repeated 8 times. The experiment lasted around 100 minutes including breaks. A Linear Discriminant Analysis (LDA) classifier was trained one time for each interaction technique and each participant. More precisely, before using a given interaction technique, the participants had to focus their attention successively on a set of flashing targets, in the same fashion as with the real interaction technique except that they were not provided with any feedback (i.e., the state did not change). These recorded EEG signals were used to train the LDA classifier, and the trained classifier could then be used during the experiment.

Results

The classifier accuracy ranged from 50% to 100% across all participants and techniques. We used that range to compute the average selection time and number of flashes predicted by the model. Taken together, our results suggest that the global technique is more efficient than the 4-chotomy in terms of selection time but at the cost of more flashes. Statistical analysis did not show significant differences between the experimental and model data. The differences between the experimental data and the model range from 0% to 22% for the selection time and the number of flashes (Figure 5). This preliminary result shows a good match between the model and the experimental results.

CONCLUSION

We have presented a model based on Markov chains that can predict the performance of interaction techniques for Brain-Computer Interfaces based on P300 signals. We have described how to use this model with three different tech-

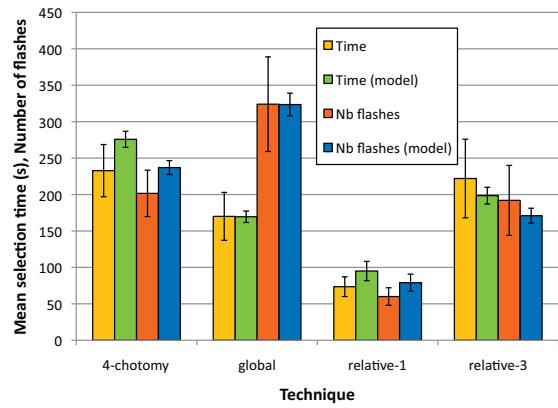


Figure 5. Comparison between experimental and model data for selection time and number of flashes for a square map (4x4). Error bars represent 95% CI. Relative-1 corresponds to close targets (1 step to reach the target), Relative-3 corresponds to farther targets (3 steps to reach the target).

niques for target selection. A preliminary experiment with three healthy participants showed that our experimental data is similar to the performance predicted by our theoretical model. Future work is necessary to fully evaluate our model with more participants, different sizes of maps and other selection techniques. This expanded testing should also take into account the handling of errors.

ACKNOWLEDGEMENTS

We thank the participants, and Daniel Vogel for proofreading and comments.

REFERENCES

1. B. Z. Allison and J. A. Pineda. Effects of SOA and flash pattern manipulations on ERPs, performance, and preference: Implications for a BCI system. *International Journal of Psychophysiology*, 59(2):127–140, 2006.
2. J. D. Bayliss. Use of the evoked potential P3 component for control in a virtual apartment. *IEEE Transactions on Rehabilitation Engineering*, 11(2):113–116, June 2003.
3. F. Bießmann. New methods for the P300 visual speller. *Lab Rotation Report, MPI Biological Cybernetics, Dept. Empirical Inference*, 2006.
4. E. Donchin, K. M. Spencer, and R. Wijesinghe. The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE Transaction on Rehabilitation Engineering*, 8(2):174–179, 2000.
5. C. M. Grinstead and J. L. Snell. *Introduction to probability*. American Mathematical Society, 1997.
6. B. Rebsamen, C. L. Teo, Q. Zeng, M. H. A. Jr., E. Burdet, C. Guan, H. Zhang, and C. Laugier. Controlling a wheelchair indoors using thought. *Intelligent Systems*, 22(2):18–24, April 2007.
7. J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Bain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.