Published in final edited form as:

Proc IEEE Int Symp Biomed Imaging. 2018 April ; 2018: 752–756. doi:10.1109/ISBI.2018.8363682.

INCORPORATING TEMPORAL DEPENDENCY ON ERP BASED BCI

Aziz Koçanoğlu, Fernando Quivira, and Deniz Erdem

Cognitive Systems Laboratory, ECE Department, Northeastern University

Abstract

In brain computer interface (BCI) systems based on event related potentials (ERPs), a windowed electroencephalography (EEG) signal is taken into consideration for the assumed duration of the ERP potential. In BCI applications inter stimuli interval is shorter than the ERP duration. This causes temporal dependencies over observation potentials thus disallows taking the data into consideration independently. However, conventionally the data is assumed to be independent for decreasing complexity. In this paper we propose a graphical model which covers the temporal dependency into consideration by labeling each time sample. We also propose a formulation to exploit the time series structure of the EEG.

Keywords

BCI; ERP; Temporal Dependency; RNN; RSVP Keyboard™

1. INTRODUCTION

Brain computer interfaces (BCIs) are designed to help people with locked in syndrome communicate. Rapid serial visual presentation (RSVP) is a BCI technique in which symbols are shown once at a time on a fixed location at a rapid rate in a random order [1, 2]. RSVP Keyboard™ is a framework which relies on event related potentials (ERPs) for binary classification [3, 4]. It is considered that ERP lasts 500[ms]. Each sequence of rapid flashes has a fixed number of trials (symbol flashes). Between sequences, the time gap is more than 500[ms] which separates sequence data temporally. Bayesian update can be performed between sequences multiplicatively as the sequences are independent. Evidence likelihood in sequences in RSVP scheme is often calculated multiplying the trial scores. Likelihood estimate can be calculated multiplying the trial scores when trial data are statistically independent from each other. However in RSVP scheme inter stimuli interval between flashes is often shorter than 500[ms]. This results in overlaps between trials. Therefore due to temporal dependency trial data is not statistically independent. In this paper we investigate a formulation of the decision making procedure specified for RSVP Keyboard™ which can be generalized for other BCIs by keeping the recursive Bayesian update between sequences and taking temporal dependency into consideration.

To keep the descriptions clear and simple in the paper we summarize the notation used throughout the paper in Tab.1, where the lower notation denotes the order index for the variable's domain.

Assume that an intent decision is made after i sequences. Recursive Bayesian update with temporal independence assumption is defined as,

$$\begin{aligned}\hat{x} &= \arg \max_x p(x | \varepsilon^{S_{0:i}}, \Phi^{S_{0:i}}) \\ &= \arg \max_x p(x | \varepsilon^{S_{0:i-1}}, \Phi^{S_{0:i-1}}) \frac{p(\varepsilon_{0:N}^{S_i} | x, \Phi_{0:M}^{S_i})}{p(\varepsilon^{S_i} | \Phi^{S_i})}\end{aligned}\quad (1)$$

where S_i denotes i^{th} sequence. The time independence simplifies the conditional EEG distribution as,

$$p(\varepsilon^{S_i} | x, \Phi^{S_i}) \propto \prod_k p(\varepsilon_k^{S_i} | \varepsilon^{S_i}) \quad (2)$$

where k denotes the trial index in a particular sequence and ε_k denotes the label of the corresponding binary question. Inserting Eq.(2) into Eq.(1) yields the decision process.

In BCI frameworks binary classification over $\mathcal{L}$ in Eq.(2) is performed using support vector machines [5, 6, 7], quadratic and linear discriminators [8, 9] conventionally. These methods assume that for a RSVP paradigm, the data within 500[ms] interval after the symbol flash is related with the intent. In such methods a classification can be performed using the entire data in which the temporal structure of the data is lost and the trial data is assumed to be independent. In RSVP Keyboard™ the classification is performed using regularized discriminant analysis [10] with the assumption of independent trials. With the independence assumption Bayesian update can be performed multiplying scores of trials [11].

However, as we explained before, trial data overlap and the independence, even it eases the calculations, should not be assumed. Therefore it is not convenient to have multiplications denoted in Eq.(2). Temporal dependency can be dealt with by redefining the trial data and separating it to non-overlapping EEG blocks. Using this information different blocks for each EEG yields a multi class classification problem for each trial [12]. This formulation is promising yet complicated. Authors propose to train a set of classifiers for sets of EEG chunks in a one versus all manner. This division does not entirely cover dependency and still has gaps in sequence updates.

In this paper we propose the graphical model in Fig. 1. We label each time sample of EEG data which allows to enforce statistical independence among EEG given the label information. In previous work the independence among EEG data is assumed without statistically representing it with a model and/or is limited with dividing EEG into chunks. We also propose a model to exploit time series structure of the EEG data. In conventional

methods likelihood of the labels are computed feeding the entire chunk of EEG into a classifier which neglects the sequentiality in data.

2. METHOD

In RSVP Keyboard™, update in Eq.(1) is performed after a sequence of trials. The sequences are independent of each other. Let us denote the set of sequences within an epoch with $S = \{S_0, S_1, \dots, S_C\}$. Within each sequence let us denote the observed EEG and the displayed symbols of i^{th} sequence as $\varepsilon_i^{S_i}$ and $\Phi_i^{S_i}$ respectively. We denote the prior probability of each letter with $p(x)$, which can be also stated as the ‘contextual prior’. The estimate is obtained after the posterior probability exceeds a threshold. Let us restate Eq.the decision making,

$$\begin{aligned} \hat{x} &= \arg \max_x p(x | \varepsilon^{S_{0:C}}, \Phi^{S_{0:C}}) \quad (3) \\ \text{subject to: } &p(x | \varepsilon^{S_{0:C}}, \Phi^{S_{0:C}}) \geq \tau \end{aligned}$$

Here, C is the smallest possible integer. The optimization problem in Eq.(3) tells us that, the framework keeps presenting sequences unless posterior distribution of any of the symbol passes a threshold value τ . We can observe that the decision making procedure consists independent sequence updates as denoted in Eq.(1). In this equation we need sequence likelihoods to update the posterior. In next section we investigate the sequence update.

Posterior Update in sequence

For simplicity in this section we assume the sequence of interest is S_i . Also for simplicity we can assume the prior probability in a sequence S_i is the posterior $p\left(x \middle| \varepsilon^{S_{0:i-1}}, \Phi^{S_{0:i-1}}\right)$. This directly follows the independence of sequences. In the sequence update assuming $p\left(\varepsilon_i^{S_i} | \Phi_i^{S_i}\right)$ in Eq.(1) is a common factor we are only interested in $p\left(\varepsilon_i^{S_i} | x, \Phi_i^{S_i}\right)$ for the particular sequence,

$$p(x | \varepsilon^{S_{0:i}}, \Phi^{S_{0:i}}) \propto p(x | \varepsilon^{S_{0:i-1}}, \Phi^{S_{0:i-1}}) p\left(\varepsilon_i^{S_i} | x, \Phi_i^{S_i}\right) \quad (4)$$

here ε denotes the EEG samples obtained in the sequence and Φ denotes the flashed letters in order where Here M denotes the total number of the flashed symbols and N denotes the length of the discrete time sequence. We aim to come up with an update procedure for letters. Assume we have an alphabet X of size $|X|$. We want to update the posteriors $p(x | \varepsilon_{0:N}^{S_{0:i}}, \Phi_{0:M}^{S_{0:i}})$ of each letter $x \in X$.

We presented in the model Fig. 1 the latent variable ℓ . It carries the information of each EEG sample throughout the sequence. $\ell \in \{0, 1\}$ where 1 means the EEG sample has ERP component, 0 means the EEG sample does not belong to the target symbol flash.

We know from RSVP Keyboard™, due to the inter stimuli interval, the EEG samples are correlated, so do the labels. This piece wise dependency is modeled in Fig. 1–(a). We further define the time labelling in 1–(b). Using such representation allows us to represent the temporal targetness for each time sample. Also defining such a label vector will allow us to exploit the sequential structure of the data. Unfortunately, it can be observed that defined labels does not become independent of each other as the dependency is caused by the intent x . This problem can be tackled with using label definition. In order to expand this definition to each EEG sample, we marginalize the latent variable over $p(\ell_{0:N}^{S_i} | x, \Phi_{0:M}^{S_i})$ in Eq.(4). This leads us to,

$$\begin{aligned}
 p(x | \ell_{0:i}^{S_{0:i}}, \Phi_{0:i}^{S_{0:i}}) & \quad (6) \\
 & \propto p(x | \ell_{0:i-1}^{S_{0:i-1}}, \Phi_{0:i-1}^{S_{0:i-1}}) \sum_L p(\ell_{0:N}^{S_i} | \ell_{0:N}^{S_i}) p(\ell_{0:N}^{S_i} | x, \Phi_{0:M}^{S_i}) \\
 & \propto p(x | \ell_{0:i-1}^{S_{0:i-1}}, \Phi_{0:i-1}^{S_{0:i-1}}) \sum_L \frac{p(\ell_{0:N}^{S_i} | \ell_{0:N}^{S_i})}{p(\ell_{0:N}^{S_i})} p(\ell_{0:N}^{S_i} | x, \Phi_{0:M}^{S_i})
 \end{aligned}$$

Observe that in Eq.(6), the proportional term is obtained after applying Bayes' rule over the EEG likelihood. Also notice that L denotes set of all possible ℓ combinations. This constant factor is $p(\ell_{0:N}^{S_i})$. We can easily say, the factor is common for each letter, therefore we easily ignore the effects of this term by proportionally estimating the posterior probabilities. We also have the freedom of using a discriminative classifier for the inference. To update the posterior we require the definitions of the terms in Eq.(6).

We can easily observe $p(\ell_{0:N}^{S_i} | x, \Phi_{0:M}^{S_i}) \in \{0, 1\}$ which works as a selection operator. We can observe this relation from the definition of the latent variable,

$$\begin{aligned}
 p(\ell_{0:N}^{S_i} | x, \Phi_{0:M}^{S_i}) &= \prod_{k=0}^N p(\ell_k^{S_i} | x, \Phi_{0:M}^{S_i}) \quad (5) \\
 p(\ell_k^{S_i} | x, \Phi_{0:M}^{S_i}) &= \begin{cases} 1 & \text{if } k \in [t_m, t_m + fs/2] \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

where f_s denotes the sampling frequency of the system. Probability distribution of latent label variables $p(\ell_{0:N}^{S_i})$ depend on the symbol display policy. Remember we denote the displayed symbols Φ^{S_i} and the intended symbol x . We say that $p(\ell_{0:N}^{S_i} = 0^{N \times 1}) = p(x \notin \Phi_{0:M}^{S_i}) = (|X| - M)/M$. We also know the display policy randomly inserts the probable letters into the display list. Therefore each of the other possible labeling have the probability $1/|X|$.

Conditional probability distribution over latent variables $p(\ell_{0:N}^{S_i} | \epsilon_{0:N}^{S_i})$ can be rewritten using the model in Fig. 1,

$$p(\ell_{0:N}^{S_i} | \epsilon_{0:N}^{S_i}) = \prod_{m=0}^N p(\ell_m^{S_i} | \ell_{<m}^{S_i}, \epsilon_m^{S_i}) \quad (6)$$

where $< m$ denotes all possible indices less than m . This is possible as in a sequence S_i labels are not independent of each other as the influence interval of Φ^{S_i} are overlapping. This allows us to treat the label as the latent state. Finding an estimate for state conditional probabilities in Eq.(6) can be achieved using long short term memory (LSTM) based recurrent neural networks (RNNs) [13, 14, 15]. Observe that RNN's are also used to factoring graphical models [16] similar to our application. RNN as a function approximate can easily output state probabilities throughout the time series by exploiting the sequential structure. We can train the network by minimizing Kullback Liebler divergence. Once we get a tight approximate of the probability distribution we can insert everything into Eq.(6) to calculate the probability distribution over symbols.

3. EXPERIMENTS / RESULTS

Here in this section we question the performance of the proposed model by comparing it with the current framework in the RSVP Keyboard™. Brain signals are recorded in 16 channels separately using a 10–20 EEG measuring system with channels 'Fp1, Fp2, F3, F4, Fz, Fc1, Fc2, Cz, P1, P2, C1, C2, Cp3, Cp4, P5, P6'. To increase the signal quality, signals are drifted using a FIR linear-phase bandpass filter in [1.5, 42][Hz] with zero DC gain and a notch filter at 60[Hz]. This data is used for determining the user intent.

Instead of using 16 channel data, we focus 4 channels C1, C2, Cz and Fz. We observed that empirically, increasing number of channels, increases the number of parameters to be estimated dramatically, which results in bad performance in temporal training procedure also decreasing number of channels actually reduces the complexity of the system as well. We also observed that incorporating frontal channels Fp1, Fp2 makes the model confused and increases discrimination difficulty of ERP from eye gaze potentials. We also give details about the steps in the method of training the neural network, in which we used the Chainer environment [17].

We train the recurrent neural network with fully connected 4 LSTM layers with 35 nodes which is justified through cross validation. We minimized KL divergence using a conjugate gradient descent method. We demonstrate a cross validation of a particular architecture in Fig. 2. We can observe, training error converges to zero, however there are some validation sets that does not match with the rest of the data. Observe that, we plot only one training error as it has a similar pattern in other validations.

As we discussed in earlier sections, declaring a latent variable for each time stamp allows us to visually give a distribution over event related potential presence. We give the temporal plot in Fig. 3. We can observe from the figure, the temporal plot of probability distributions of latent variables follows the relationship with Eq.(6), also learning a temporal model allows us to locate the ERP response in the observed EEG. The second plot in Fig. 3 gives us an intuition, depending on the framework, the user might have performed weak or responded to another symbol which is also captured with this model.

We compared the method using calibration data from 8 different users. We trained an RNN with 10 fold cross validation on data and choose the model with minimum validation error. We calculated the AUC scores using the same procedure as conventional RSVP does and visualize the scores in Fig. 3. We can observe that classifying using sequence data increases the performance for badly classified trials. We observed also there are users, whose ERP can be distinguished from non-ERP using a quadratic boundary. Therefore we conclude by stating, taking the entire sequence into consideration increases the performance in poorly classified data sets.

4. CONCLUSION

In this paper, we proposed a graphical model for a sequence that incorporates temporal dependency. We observed proposed model leverages poorly classified users. This is promising as the users with AUC lower than 0.75 values could not manage to type accurately in online sessions. With proposed method we can increase the typing performance of individual users and help them communicate. We also denote that hyper parameters in conventional model are optimized explicitly for AUC (separability of classes), instead the proposed model is trained to detect ERP activity. We can state that proposed model performs similar/better using a comparison metric where the conventional model has the strong hand. As described before, we want to investigate the typing speed improvement with the method.

We observed from experiments sequential modeling degrades the performance for high AUC users with values around 0.9. This can be explained with the Gaussianity of feature vectors of particular users. We aim to get an ensemble method with both models to provide individual user's ability to type without decreasing other user's performance. In our future work we want to try different architectures on RNN to avoid such errors. We also want to incorporate the probabilistic nature of ERP appearance. As we observed through experiments, the users can react to non-target symbols or can not react to targets. Therefore we can include a confusion matrix which denotes the probability of response appearance.

Acknowledgments

Our work is supported by NSF (IIS-1149570, CNS-1544895), NIDLRR (90RE5017-02-01), and NIH (R01DC009834).

References

- [1]. Acqualagna Laura and Blankertz Benjamin, "Gaze-independent bci-spelling using rapid serial visual presentation (rsvp)," *Clinical Neurophysiology*, vol. 124, no. 5, pp. 901–908, 2013. [PubMed: 23466266]
- [2]. Acqualagna Laura, Treder Matthias Sebastian, Schreuder Martijn, and Blankertz Benjamin, "A novel brain-computer interface based on the rapid serial visual presentation paradigm," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. IEEE, 2010*, pp. 2686–2689.
- [3]. Orhan Umut, Erdogmus Deniz, Roark Brian, Oken Barry, and Melanie Fried-Oken, "Offline analysis of context contribution to erp-based typing bci performance," *Journal of neural engineering*, vol. 10, no. 6, pp. 066003, 2013. [PubMed: 24099944]
- [4]. Orhan Umut, Hild Kenneth E, Erdogmus Deniz, Roark Brian, Oken Barry, and Fried-Oken Melanie, "Rsvp keyboard: An eeg based typing interface," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012*, pp. 645–648.
- [5]. Rakotomamonjy Alain, Guigue Vincent, Mallet Gregory, and Alvarado Victor, "Ensemble of svms for improving brain computer interface p300 speller performances," *Artificial Neural Networks: Biological Inspirations–ICANN 2005*, pp. 45–50, 2005.
- [6]. Rakotomamonjy Alain and Guigue Vincent, "Bci competition iii: dataset ii-ensemble of svms for bci p300 speller," *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1147–1154, 2008. [PubMed: 18334407]
- [7]. Combaz Adrien, Manyakov Nikolay V, Chumerin Nikolay, Johan AK Suykens, and Van Hulle Marc M, "Feature extraction and classification of eeg signals for rapid p300 mind spelling," in *Machine Learning and Applications, 2009. ICMLA'09. International Conference on. IEEE, 2009*, pp. 386–391.
- [8]. Krusienski Dean J, Sellers Eric W, Cabestaing François, Bayoudh Sabri, McFarland Dennis J, Vaughan Theresa M, and Wolpaw Jonathan R, "A comparison of classification techniques for the p300 speller," *Journal of neural engineering*, vol. 3, no. 4, pp. 299, 2006. [PubMed: 17124334]
- [9]. Krusienski Dean J, Sellers Eric W, McFarland Dennis J, Vaughan Theresa M, and Wolpaw Jonathan R, "Toward enhanced p300 speller performance," *Journal of neuroscience methods*, vol. 167, no. 1, pp. 15–21, 2008. [PubMed: 17822777]
- [10]. Friedman Jerome H, "Regularized discriminant analysis," *Journal of the American statistical association*, vol. 84, no. 405, pp. 165–175, 1989.
- [11]. Moghadamfalahi Mohammad, Orhan Umut, Akcakaya Murat, Nezamfar Hooman, Fried-Oken Melanie, and Erdogmus Deniz, "Language-model assisted brain computer interface for typing: A comparison of matrix and rapid serial visual presentation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 5, pp. 910–920, 2015. [PubMed: 25775495]
- [12]. Orhan Umut, Delia Fernandez-Canellas Murat Akcakaya, Brooks Dana H, and Erdogmus Deniz, "Utilization of temporal trial dependency in erp based bcis," in *Signals, Systems and Computers, 2014 48th Asilomar Conference on. IEEE, 2014*, pp. 26–30.
- [13]. Funahashi Ken-ichi and Nakamura Yuichi, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural networks*, vol. 6, no. 6, pp. 801–806, 1993.
- [14]. Hüskén Michael and Stagge Peter, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, pp. 223–235, 2003.
- [15]. Williams Ronald J and Peng Jing, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.
- [16]. Johnson Matthew, David K Duvenaud Alex Wiltchko, Adams Ryan P, and Datta Sandeep R, "Composing graphical models with neural networks for structured representations and fast inference," in *Advances in Neural Information Processing Systems, 2016*, pp. 2946–2954.

- [17]. Tokui Seiya, Oono Kenta, Hido Shohei, and Clayton Justin, “Chainer: a next-generation open source framework for deep learning,” in Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

(a) Graphical model of a single Sequence

(b) Concatenated labels and EEG

Fig. 1:

(a) Graphical model for a single sequence $\mathcal{S}_i$. Model represents one recursive Bayesian update denoted in Eq.(1). Observe that a single letter flash $\left(\Phi_m^{S_i}\right)$ effects multiple label

chunks $\bar{\ell}_{c_1:c_2}^{S_i}$. (b) Definition of label and EEG chunks respectively. Observe that there lies a

dependency between labels if the flashed letters are not given. Therefore in (b) labels will have dependencies over each other.

Fig. 2:

Training error and validation error samples in a training. We can observe, training data is not explicitly labeled as it is extremely user dependent. In the data there exists data without response that damages the training procedure (Validation error does not decrease at all).

Fig. 3:

Correct detection and misclassification example. We can observe that the RNN is capable of giving a probability distribution over all latent variables. In the first example ERP is localized. In the second example we can observe that based on the training procedure the RNN captures some ERP shaped responses where the labeled are not considered to be ERP.

Fig. 4:

We calculate the AUC values for sequence posterior update, using posterior multiplicative scores as target or non-target scores. Conventional RSVP calculates scores for target and non-target trials.

Table 1:

Notation Legend

X : Symbol set with cardinality $|X|$

S_{ij} : all sequences from i^{th} to j^{th} where $j > i$

$\varepsilon_{0:k}^{S_i}$: EEG samples in i^{th} seq. between $[0, k]$

$\ell_{0:k}^{S_i}$: Latent variable in i^{th} seq. between $[0, k]$

$\Phi_{0:k}^{S_i}$: Flashed symbols in i^{th} seq. between $[0, k]$

$\varepsilon^i = \begin{bmatrix} \varepsilon_0^{S_i}, \varepsilon_1^{S_i}, \dots, \varepsilon_N^{S_i} \end{bmatrix} \in \mathbb{R}^{N \times 1}$

$\Phi^i = \begin{bmatrix} \Phi_0^{S_i}, \Phi_1^{S_i}, \dots, \Phi_M^{S_i} \end{bmatrix} \Phi_m \in X \forall m$