

## Chapter 1

# HIGH-SIGNIFICANCE AVERAGES OF EVENT-RELATED POTENTIAL VIA GENETIC PROGRAMMING

Luca Citi<sup>1</sup>, Riccardo Poli<sup>1</sup>, and Caterina Cini<sup>1</sup>

<sup>1</sup>*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, CO4 3SQ, UK*

**Abstract** In this paper we use register-based genetic programming with memory-with-memory to discover probabilistic membership functions that are used to divide up data-sets of event-related potentials recorded via EEG in psycho-physiological experiments based on the corresponding response times. The objective is to evolve membership functions which lead to maximising the statistical significance with which true brain waves can be reconstructed when averaging the trials in each bin. Results show that GP can significantly improve the fidelity with which ERP components can be recovered.

**Keywords:** Event-related potentials, Averaging, Register-based GP, Memory-with-Memory

## 1. Introduction

The electrical activity of the brain is typically recorded from the scalp using Electroencephalography (EEG). This is used in electrophysiology, in psychology, as well as in Brain-Computer Interface (BCI) research. Particularly important for these purposes are Event-Related Potentials (ERPs). ERPs are relatively well defined shape-wise variations to the ongoing EEG elicited by a stimulus and/or temporally linked to it (Luck, 2005). ERPs include early exogenous responses, due to the primary processing of the stimulus, as well as later endogenous responses, which are a reflection of higher cognitive processing induced by the stimulus (Donchin and Coles, 1988).

While the study of *single-trial* ERPs has been considered of great importance since the early days of ERP analysis, in practice the presence of noise and artifacts has forced researchers to make use of *averaging* as part of their standard investigation methodology (Donchin and Lindsley, 1968). Even today, despite

enormous advances in acquisition devices and signal-processing equipment and techniques, ERP averaging is still ubiquitous (Handy, 2004; Luck, 2005).

ERP averaging is also a key element in many BCIs. BCIs measure specific signals of brain activity intentionally and unintentionally induced by the participant and translate them into device control signals (see, for example, (Farwell and Donchin, 1988; Wolpaw et al., 1991; Pfurtscheller et al., 1993; Birbaumer et al., 1999; Wolpaw et al., 2000; Furdea et al., 2009)). Averaging is frequently used to increase accuracy in BCIs where the objective is to determine which of the stimuli sequentially presented to a user is attended. This is achieved via the classification of the ERP components elicited by the stimuli. This form of BCI — which effectively started off with the seminal work of (Farwell and Donchin, 1988) who showed that it was possible to spell words through the detection of P300 waves — is now one of the most promising areas of the discipline (e.g., see (Bostanov, 2004; Rakotomamonjy and Guigue, 2008; Citi et al., 2008)).

Averaging has empirically been shown to improve the accuracy in ERP-based BCIs. However, the larger the number of trials that need to be averaged, the longer it takes for the system to produce a decision. So, only a limited number of trials can be averaged before a decision has to be taken. A limitation on the number of trials one can average is also present in psychophysiological studies based on ERPs: the larger the number of trials that are accumulated in an average, the longer an experiment will last, potentially leading to participants fatiguing, to increases in noise due to variations in electrode impedances, etc. So, both in psychophysiological studies and in BCIs it would be advantageous to make the absolute best use of all the information available in each trial. However, as we will discuss in Section 2, standard averaging techniques do not achieve this.

In recent work (Poli et al., 2009) we proposed, tested and theoretically analysed an extremely simple technique which can be used in forced-choice experiments. In such experiments response times are measured via a button press or a mouse click. Our technique consists of binning trials based on response times and then averaging. This can significantly alleviate the problems of other averaging methods, particularly when response times are relatively long. In particular, results indicated that the method produces clearer representations of ERP components than standard averaging, revealing finer details of components and helping in the evaluation of the true amplitude and latency of ERP waves.

The technique relies on dividing an ERP dataset into bins. The size and position of these bins is extremely important in determining the fidelity with which bin averages represent true brain waves. In (Poli et al., 2009) we simply used standard (mutually exclusive) bins. That is, each bin covered a particular range of response times, the ranges associated to different bins did not overlap and no gaps were allowed between the bins. As we will explain in Section 3, this implies that, in bin averages, true ERP components are distorted via the

convolution with a kernel whose frequency response is itself a convolution between the frequency response of the original latency distribution  $\ell(t)$  and the Fourier transform of a *rectangular window* (a *sinc* function).

While provably this has the effect of improving the resolution with which ERPs can be recovered via averages, it is clear that the convolution with *sinc* will produce distortions due to the Gibbs phenomenon. Also, the width and position of the bins we used in (Poli et al., 2009) was determined heuristically. We chose bins as follows: one gathered the lower 30% of the response time distribution, one the middle 30% and one the longer 30%.<sup>1</sup> However, it is clear that neither the choice of crisp mutually exclusive membership functions for bins (leading to convolution with *sinc*) nor the position and width of the bins is optimal.

So, although our binning method is a marked improvement over traditional techniques, it still does not make the best use of the information available in an ERP dataset. It is arguable, for example, that doing binning using gradual membership functions would provide even better ERP reconstruction fidelity. Similarly, setting the size of the bins on the basis of the noise in the data and the particular shape of the response time distribution would be beneficial to make best use of the available trials. Finding bin membership functions which satisfy these criteria, however, is difficult. It is also difficult to specify what notion of optimality one should use. In this paper we solve both problems.

The paper is organised as follows. After the reviews of previous work provided in Sections 2 and 3, we define what an optimal set of binning functions is (Section 4). As we will see this involves the use of statistical tests on the data belonging to different bins. Then (Section 5), we apply Genetic Programming (Poli et al., 2008) to the task of identifying optimal membership functions for bins in such a way as to get the best possible reconstruction of real ERP components from bin averages. The results of this process, as described in Section 6, provide significant improvements over the original technique. We give some conclusions and indications of future work in Section 7.

## 2. Averaging Techniques for ERPs

There are essentially three classes of methods that are commonly used to resolve ERP components via averaging. *Stimulus-locked averaging* requires extracting epochs of fixed duration from the EEG signal starting at the stimulus presentation and averaging the corresponding ERPs (Lindsley, 1968). An important problem with this form of averaging is that any ERP components whose latency is not phase-locked with the presentation of the stimuli may be

<sup>1</sup>Since extremely long response times are typically the sign of the participant being distracted or having had some other problem with providing a response, the 10% of the trials with the longest response times were discarded.

significantly distorted as a result of averaging (Spencer, 2004; Luck, 2005). This is because the average,  $a(t)$ , of randomly shifted versions of a waveform,  $w(t)$ , is the convolution between the original waveform and the latency distribution,  $\ell(t)$ , for that waveform, i.e.,  $a(t) = w(t) \star \ell(t) = \int w(t - \tau)\ell(\tau) d\tau$ , e.g., see (Zhang, 1998). This typically means that a stimulus-locked average can only show a smoothed (low-pass filtered) version of each variable-latency component.

The problem is particularly severe when the task a subject needs to perform after the presentation of the stimuli is relatively difficult since the variability in the latencies of endogenous ERP components and in response times increase with the complexity of the task (Luck, 2005; Polich and Comerchero, 2003). In these cases, multiple endogenous variable-latency components may appear as a single large blurred component in the average; a synthetic example is shown in Figure 1-1 (left).<sup>2</sup> This makes it very difficult to infer true brain area activity for any response occurring after the early exogenous potentials typically elicited by (and synchronised with) a stimulus.

In experiments in which the task requires participants to provide a specific behavioural response (e.g., in the form of a button press or a spoken response), *response-locked averaging* can be used as an alternative to stimulus-locked averaging to help resolve variable-latency ERP components that are synchronised with the response; see, for example, (Luck and Hillyard, 1990; Keus et al., 2005; Spencer, 2004; Töllner et al., 2008). In this case, however, the early responses associated and phase-locked with the stimulus will end up being blurred and hard to distinguish, since they are represented in the average by the convolution of their true waveform with the response-time distribution; see (Zhang, 1998). A synthetic example illustrating this problem is shown in Figure 1-1 (right).

Thus, inferring whether a component in an average represents a true effect or it is due to averaging biases can then be very difficult. Note that averaging more data does not help increase the fidelity of the reconstructed signals because there is a *systematic error* in the averaging process.

A third alternative to resolve variable-latency waves is to attempt to identify such components in each trial and estimate their latency. Then, shifting trials on the basis of estimated latencies and averaging may bring out the desired component from its noise background. However, most methods of this kind require prior knowledge of what type of component to expect and at what times. What if this knowledge is not available? Without this information automated detection algorithms have very little hope of finding the latency of the waves of interest. Also, latency detection algorithms assume that the component of

<sup>2</sup>Real EEG signals are extremely noisy. So, synthetic data illustrate the problem more clearly.

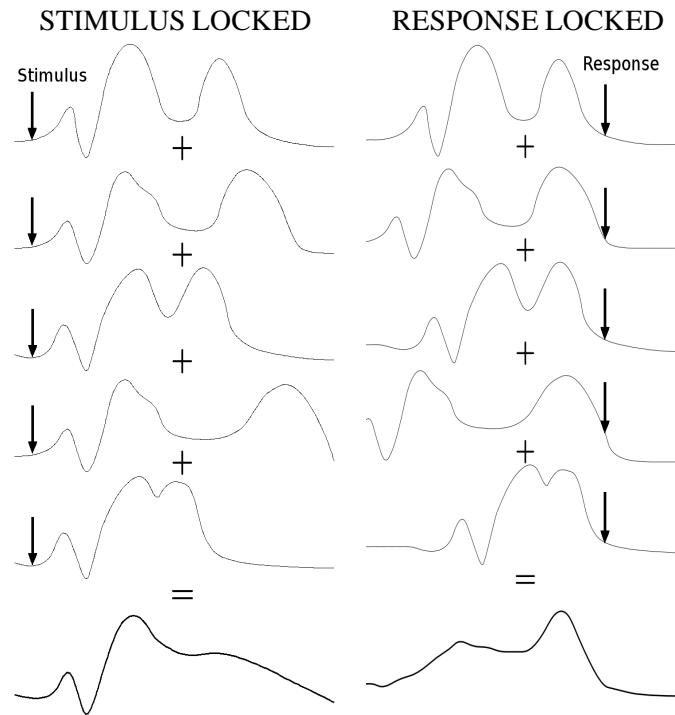


Figure 1-1. Example of distortions produced by averaging: the five sample ERPs at the top present two positive and one negative deflections each, which are phase-locked with a stimulus, as well as one positive component, which is of variable latency. Averaging them (plots at the bottom) preserves the exogenous components when trials are stimulus-locked (left). This, however, turns the variable-latency component into an inconspicuous plateau which could easily be misinterpreted as a continuation of the preceding positive wave. A response-locked average (right), on the other hand, preserves the variable-latency endogenous component but smears out the details of early potentials turning them into a single, wide positive deflection.

interest is present in every trial and we just need to find its latency in the trial. What if an ERP component is not always elicited by the stimuli? The presence of a component might be, for example, condition-dependent, or dependent on whether or not a participant attended a stimulus, whether a participant was rested or tired, whether there was habituation to the stimuli, etc. (Bonala et al., 2008; Wagner et al., 2000). If a component was absent frequently, running a latency-measuring algorithm on trials where the component did not occur would inundate the averaging process with bias and noise. And, unfortunately, thresholds or even more sophisticated algorithms for the detection of the *presence* of the component, which in principle could be used to properly handle trials that do not contain it, produce large numbers of mis-classification errors. So,

the composition of detection errors with latency-estimation errors may render component-locked averaging very unreliable in many situations.

Note also that all methods that realign trials based on component latencies can potentially suffer from a *clear-centre/blurred-surround problem*. That is, after shifting trials based on the latency of a particular ERP component, all instances of that component will be synchronised, thereby effectively becoming fixed-latency elements. However, stimulus-locked components will now become variable-latency components. Also, all (other) components that are phase-locked with some other event (e.g., the response), but not with the component of interest, will remain variable-latency. Not surprisingly, then, they will appear blurred and distorted in a component-locked average.

It is clear that the standard averaging techniques reviewed above are not entirely satisfactory and that a more precise and direct way of identifying variable-latency components as well as measuring their latency and amplitude is needed. In the following section we describe the binning technique we developed in (Poli et al., 2009), which significantly improves on previous methods.

### 3. Averaging Response-time Binned ERPs

In (Poli et al., 2009) we proposed an extremely simple technique — binning trials based on their recorded response time and then applying averaging to the bins. This has the potential of solving the problems with the three main ways of performing averages (stimulus-locked, component-locked and response-locked) discussed above, effectively reconciling the three methods. In particular, response-time binning allows one to significantly improve the resolution with which variable-latency waves can be recovered via averaging, even if they are distant from the stimulus-presentation and response times. The reason for this is simple to understand from a qualitative point of view.

The idea is that if one selects out of a dataset all those epochs where a participant was presented with qualitatively identical stimuli and gave the same response within approximately the same amount of time, it is reasonable to assume that similar internal processes will have taken place. So, within those trials, ERP components that would normally have a widely variable latency might be expected to, instead, present a much narrower latency distribution, i.e., they should occur at approximately the same time in the selected subset of trials. Thus, if we bin epochs on the basis of stimuli, responses and response times, we would then find that, for the epochs within a bin, the stimulus, the response, as well as fixed- and variable-latency components are much more synchronised than if one did not divide the dataset. Averaging such epochs should, therefore, allow the rejection of noise while at the same time reducing the undesirable distortions and blurring (the systematic errors) associated with averaging. Response-time binning and averaging should thus result in

the production of clearer, less biased descriptions of the activity which really takes place in the brain in response to the stimuli *without the need for prior knowledge* of the phenomena taking place and ERP components present in the EEG recordings.

In (Poli et al., 2009) we assessed the binning technique both empirically and theoretically. For empirical validation we modified and used an experiment originally designed by (Esterman et al., 2004) in which the task was relatively difficult, since target detection is based on specific combinations of multiple features (i.e., requiring feature binding), and where response times varied from around 400ms to over 2 seconds. We evaluated the empirical results in a number of ways, including: (a) a comparison between stimulus-locked and response-locked averages which showed how these are essentially identical under response-time binning; (b) an analysis of differences between bin means, medians and quartiles of the amplitude distributions and an analysis of statistical significance of amplitude differences using Kolmogorov-Smirnov tests which showed that bins indeed captured statistically different ERP components; and (c) an analysis of the signal-to-noise ratio (SNR) with and without binning which showed that the (expected) drop in SNR due to the smaller dataset cardinality associated with bins is largely compensated by a corresponding increase due to the reduction in systematic errors.

From the theoretical point of view, we provided a comprehensive analysis of the resolution of averages with and without binning, which showed that there are resolution benefits in applying response-time binning even when there is still a substantial variability in the latency of variable-latency components after response-time binning. We summarise this analysis below since this is the starting point for our fitness function, as we will show in Section 4.

Let us assume that there are three additive components in the ERPs recorded in a forced-choice experiment — a stimulus-locked component,  $s(t)$ , a response-locked component,  $r(t)$ , and a variable-latency component,  $v(t)$ . Let  $R$  be a stochastic variable representing the response time in a trial and let  $\rho(t)$  be its density function. Similarly, let  $L$  be a stochastic variable representing the latency of the component  $v(t)$  and let  $\ell(t)$  be the corresponding density function. Let us further assume that response time and latency do not affect the shape of these components. Under these assumptions we obtain the following equation for the stimulus-locked average  $a_s(t)$ :

$$a_s(t) = s(t) + v(t) \star \ell(t) + r(t) \star \rho(t) \quad (1.1)$$

where  $\star$  is the convolution operation.

Let us consider the most general conditions possible. Let  $L$  and  $R$  be described by an unspecified joint density function  $p(l, r)$ . So, the latency and

response-time distributions are marginals of this joint distribution:

$$\ell(l) = \int p(l, r) dr \quad \text{and} \quad \rho(r) = \int p(l, r) dl.$$

Note that by the definition of conditional density function, we have that

$$p(l, r) = p(r|l)\ell(l) \quad \text{and} \quad p(l, r) = p(l|r)\rho(r)$$

where  $p(r|l)$  is the pdf of  $R$  when  $L = l$  and  $p(l|r)$  is the pdf of  $L$  when  $R = r$ .

In (Poli et al., 2009) we showed that if one considers a classical “rectangular” bin collecting the subset of the trials having response times in the interval  $[\chi_1, \chi_2)$ , i.e., such that  $\chi_1 \leq R < \chi_2$ , the joint distribution of  $L$  and  $R$  transforms into

$$p^{[\chi_1, \chi_2)}(l, r) = \frac{\delta(\chi_1 \leq r < \chi_2)p(l, r)}{\int_{\chi_1}^{\chi_2} \rho(r) dr},$$

where the  $\delta(x)$  returns 1 if  $x$  is true and 0 otherwise. So, it has the function of zeroing the distribution outside the strip  $[\chi_1, \chi_2)$ . The denominator normalises the result so that  $p^{[\chi_1, \chi_2)}(l, r)$  integrates to 1.

We also showed that taking the marginal of this distribution w.r.t.  $l$  gives us the response time distribution for response-time bin  $[\chi_1, \chi_2)$ :

$$\rho^{[\chi_1, \chi_2)}(r) = \frac{\delta(\chi_1 \leq r < \chi_2) \rho(r)}{\int_{\chi_1}^{\chi_2} \rho(r) dr}.$$

The marginal of the distribution  $p^{[\chi_1, \chi_2)}(l, r)$  w.r.t.  $r$ , which gives us the latency distribution for the trials in response-time bin  $[\chi_1, \chi_2)$ , is:

$$\ell^{[\chi_1, \chi_2)}(l) = \frac{\Pr\{\chi_1 \leq R < \chi_2 | l\} \ell(l)}{\int_{\chi_1}^{\chi_2} \rho(r) dr}.$$

These two marginals are important because we can express the stimulus-locked bin average as follows:

$$a_s^{[\chi_1, \chi_2)}(t) = s(t) + v(t) \star \ell^{[\chi_1, \chi_2)}(t) + r(t) \star \rho^{[\chi_1, \chi_2)}(t).$$

The marginals determine in what ways and to what extent  $v(t)$  and  $r(t)$  appear distorted and blurred in the average. So, in order to understand why  $a_s^{[\chi_1, \chi_2)}(t)$  provides a better representation of  $r(t)$  and  $\ell(t)$  than  $a_s(t)$ , we need to analyse the differences between the distribution  $\rho^{[\chi_1, \chi_2)}(t)$  and  $\rho(t)$  and between the distribution  $\ell^{[\chi_1, \chi_2)}(t)$  and  $\ell(t)$ . We will concentrate on the former pair since the arguments for the latter are almost symmetric.

The key difference between  $\rho^{[\chi_1, \chi_2)}(t)$  and  $\rho(t)$  is that, apart from a scaling factor,  $\rho^{[\chi_1, \chi_2)}(t)$  is the product of  $\rho(t)$  and a rectangular windowing function,  $\delta(\chi_1 \leq t < \chi_2)$ . In the frequency domain, therefore, the spectrum of

$\rho^{[\chi_1, \chi_2]}(t)$ , which we denote with  $\mathcal{R}^{[\chi_1, \chi_2]}(f)$ , is the convolution between the spectrum of  $\rho(t)$ , denoted as  $\mathcal{R}(f)$ , and the spectrum of a translated rectangle,  $\Delta(f)$ . This is a scaled and rotated (in the complex plane) version of the *sinc* function (i.e., it behaves like  $\sin(f)/f$ ). The function  $|\Delta(f)|$  has a large central lobe whose width is inversely proportional to the bin width  $\chi_2 - \chi_1$ . Thus, when convolved with  $\mathcal{R}(f)$ ,  $\Delta(f)$  behaves as a low pass filter. As a result,  $\mathcal{R}^{[\chi_1, \chi_2]}(f) = \mathcal{R}(f) \star \Delta(f)$  is a smoothed and enlarged version of  $\mathcal{R}(f)$ . In other words, while  $\rho^{[\chi_1, \chi_2]}(t)$  is still a low-pass filter, it has a higher cut-off frequency than  $\rho(t)$ . So,  $a_s^{[\chi_1, \chi_2]}(t)$  provides a less blurred representation of  $r(t)$  than  $a_s(t)$ .

We will modify this analysis in the next section for the purpose of defining a suitable fitness measure the optimisation of which would lead to maximising the statistical significance with which ERP components can be reconstructed via binning and averaging.

#### 4. Binning Optimality and Fitness Function

As described in the previous section, in (Poli et al., 2009) we used the function  $\delta(\chi_1 \leq R < \chi_2)$  to bin trials. To get the best out of the binning technique, here we will replace this function with a *probabilistic membership function* which gives the probability that a trial characterised by a response time  $R$  would be accepted in a particular bin  $b$ . Let us denote this probabilistic membership function as

$$\mathcal{P}_b(r) = \Pr\{\text{accept trial in bin } b \mid \text{trial response time } R = r\}.$$

Naturally, when  $\mathcal{P}_b(r) = \delta(\chi_1 \leq R < \chi_2)$ , then  $b$  is a traditional (crisp, rectangular) bin.

Let us denote with a binary stochastic variable  $A$  the event {accept trial for averaging in bin  $b$ }. Let  $p(a, l, r)$  be the joint distribution of the events  $R = r$ ,  $L = l$  and  $A = a$ . This can be decomposed as follows

$$p(a, l, r) = p(a|l, r)p(l, r).$$

Since  $A$  does not depend on  $L$  but only on  $R$  (we base the decision to accept trials in a bin only on their associated response time), we have that  $p(A = \text{true}|l, r) = \mathcal{P}_b(r)$  and  $p(A = \text{false}|l, r) = 1 - \mathcal{P}_b(r)$ .

Focusing our attention on the subset of the trials falling within bin  $b$ , we obtain the following joint distribution of  $L$  and  $R$

$$p^b(l, r) = p(l, r \mid A = \text{true}) = \frac{p(A = \text{true}, l, r)}{p(A = \text{true})} = \frac{p(A = \text{true} \mid l, r)p(l, r)}{p(A = \text{true})}$$

Hence

$$p^b(l, r) = \frac{\mathcal{P}_b(r)p(l, r)}{\int \int \mathcal{P}_b(r)p(l, r) dr dl} = \frac{\mathcal{P}_b(r)p(l, r)}{\int \mathcal{P}_b(r)\rho(r) dr}.$$

So,

$$\rho^b(r) = \int p^b(l, r) dl = \frac{\mathcal{P}_b(r) \int p(l, r) dl}{\int \mathcal{P}_b(r) \rho(r) dr} = \frac{\mathcal{P}_b(r) \rho(r)}{\int \mathcal{P}_b(r) \rho(r) dr}.$$

Also,

$$\ell^b(l) = \int p^b(l, r) dr = \frac{\int \mathcal{P}_b(r) p(l, r) dr}{\int \mathcal{P}_b(r) \rho(r) dr} = \frac{\ell(l) \int \mathcal{P}_b(r) p(r|l) dr}{\int \mathcal{P}_b(r) \rho(r) dr}.$$

Again these two marginals are important because we can express the stimulus-locked bin average as follows:

$$a_s^b(t) = s(t) + v(t) \star \ell^b(t) + r(t) \star \rho^b(t).$$

From the equations above, one can clearly understand how different definitions of the probabilistic membership function  $\mathcal{P}_b(r)$  can lead to radically different results in terms of the resolution of true ERP components in bin averages.

Naturally, one will generally use multiple probabilistic response-time bins for the purpose of analysing ERP trials. For each, a membership function  $\mathcal{P}_b(r)$  must be defined. Our objective is to use GP to discover these membership functions in such a way as to maximise the information extracted from the raw data. To do so, we need to define an appropriate fitness function.

While we form bins based on response times, each data element in a bin actually represents a fragment of EEG signal recorded at some electrode site. The question we need to ask is: what do we mean by extracting maximal information from these data? Naturally, alternative definitions are possible. Here we want to focus on the *getting ERP averages which are maximally significantly different*.

An ERP bin average,  $a_s^b(t)$ , is effectively a vector, each element of which is the signal amplitude recorded at a particular time after stimulus presentation averaged over all the trials in a bin. Because we use probabilistic membership functions for the bins, the composition of a bin is in fact a stochastic variable. Let us denote the stochastic variable representing bin  $b$  with  $\mathcal{B}_b$ . The probability distribution of  $\mathcal{B}_b$  is determined by the membership function  $\mathcal{P}_b(r)$  and by the response time distribution  $\rho(r)$ . An instantiation of  $\mathcal{B}_b$ ,  $\beta_b$ , is effectively an array with as many rows as there are trials in bin  $b$  and as many columns as there are time steps in each epoch. An element in  $\beta_b$  represents the voltage amplitude recorded in a particular trial and in a particular time step in that trial at the chosen electrode. Let  $\beta_b(t)$  represent the set of the amplitudes recorded at time  $t$  in the trials in bin  $b$ .

Let us consider two bins,  $b_1$  and  $b_2$ . If  $\beta_{b_1}$  is an instantiation of  $\mathcal{B}_{b_1}$  and  $\beta_{b_2}$  is an instantiation of  $\mathcal{B}_{b_2}$ , one could check whether the signal amplitude distributions recorded in bins  $b_1$  and  $b_2$  at a particular time step  $t$  are statistically different by applying the Kolmogorov-Smirnov test for distributions to the datasets  $\beta_{b_1}(t)$  and  $\beta_{b_2}(t)$ . The test would return a  $p$  value, which we will call

$p_{b_1, b_2}(t)$ . The smaller  $p_{b_1, b_2}(t)$ , the better the statistical separation between the signal amplitude distributions in bins  $b_1$  and  $b_2$  at time step  $t$ . Naturally to get an indication of how statistically different the ERPs in different bins are one would then need to somehow integrate the  $p_{b_1, b_2}(t)$  values obtained at different  $t$ 's and for different pairs of bins.

Since we are interested in obtaining bins (via the optimisation of their membership functions  $\mathcal{P}_b(r)$ ) which contain maximally mutually statistically different trials, we require that the sum of the  $p$  values returned by the Kolmogorov-Smirnov test when comparing the signal amplitudes in each pair of bins over the time steps in an epoch be as small as possible. So, we want to maximise the following *fitness function*:

$$f = \sum_{b_1 \neq b_2} \sum_t (1 - E[p_{b_1, b_2}(t)]), \quad (1.2)$$

where the expectation operator  $E[\cdot]$  is required because  $p_{b_1, b_2}(t)$  is a stochastic variable in that we can only apply the Kolmogorov-Smirnov test to amplitude measurements obtained from *instantiations* of the stochastic variables  $\mathcal{B}_{b_1}$  and  $\mathcal{B}_{b_2}$ . For this reason, the use of Equation (1.2) as a fitness function would require repeatedly assigning trials to bins based on their membership functions, assessing the mutual statistical independence of the trials, and averaging the results. However, this repeated sampling is a very expensive operation (see Section 5). Therefore, we adopted a noisy fitness function, where the expectation operator is omitted. In other words, we only sample the stochastic variables  $\mathcal{B}_{b_1}$  and  $\mathcal{B}_{b_2}$  once per fitness evaluation. Fitness, however, gets re-evaluated periodically, as described in the next section. So, general and robust solutions to the problem are favoured by evolution.

## 5. GP System and Settings

We did our experiments using a linear register-based GP system. The system uses a steady-state update schedule.

The primitive set used in our experiments is shown in Table 1-1. The instructions refer to four registers: the input register `ri` which is loaded with the response time,  $r$ , of a trial before a program is evaluated, the two general-purpose registers `r0` and `r1` that can be used for numerical calculations, and the register `rs` which can be used as a swap area. `r0`, `r1` and `rs` are initialised to 0. The output of the program is read from `r0` at the end of its execution. In the addition and multiplication instructions we used the memory-with-memory technique proposed in (McPhee and Poli, 2008) with a memory coefficient of 0.5. So, for example the instruction `r0 <- r0 + ri` is actually implemented as `r0 = 0.5 * r0 + 0.5 * ( r0 + ri )` while `r1 <- r0 * r1` is implemented as `r1 = 0.5 * r1 + 0.5 * ( r0 * r1 )`.

Table 1-1. Primitive set used in our experiments.

NOP	r0 <- -1	r1 <- r0 + r1
r0 <- 0	r1 <- 1	r0 <- r0 * r1
r1 <- 0	r0 <- -r0	r1 <- r0 * r1
r0 <- 0.5	r1 <- -r1	r0 <- r0 * r0
r1 <- -0.5	r0 <- r0 + ri	r1 <- r1 * r1
r0 <- -0.1	r1 <- r1 + ri	rs <-> r0
r1 <- 0.1	r0 <- r0 + r1	rs <-> r1

As in (Poli et al., 2009), in our tests we consider three bins. So, we need to evolve three membership functions, which we will call  $\mathcal{P}_1(r)$ ,  $\mathcal{P}_2(r)$  and  $\mathcal{P}_3(r)$ . To help GP in this difficult task we constrained the family of functions from which the membership functions could be drawn. So, instead of evolving the three functions  $\mathcal{P}_1(r)$ ,  $\mathcal{P}_2(r)$  and  $\mathcal{P}_3(r)$ , we decomposed each function into three components and we asked GP to evolve the components used in the formulation of each  $\mathcal{P}_i(r)$ . So, each GP individual was actually made up of nine programs. All nine must be run to decide with which probability an element of an ERP dataset should belong to each response-time bin.

More specifically, our membership functions had the following form:

$$P_i(x) = \left( \text{pcos} \left( \frac{r - c(r)}{w(r)} \right) \right)^{|e(r)|}$$

where  $c(r) = c_i + p_{ic}(r)$ ,  $w(r) = w_i + p_{iw}(r)$ ,  $e(r) = e_i + p_{ie}(r)$  and  $\text{pcos}(x) = \cos\left(\frac{\pi}{2}x\right)$  if  $|x| < 1$ , and 0 otherwise. Here  $p_{1c}(r)$ ,  $p_{2c}(r)$ ,  $p_{3c}(r)$ ,  $p_{1w}(r)$ ,  $p_{2w}(r)$ ,  $p_{3w}(r)$ ,  $p_{1e}(r)$ ,  $p_{2e}(r)$ , and  $p_{3e}(r)$  are the nine programs forming a particular individual. The terms  $c_1$ ,  $c_2$ ,  $c_3$ ,  $w_1$ ,  $w_2$ ,  $w_3$ ,  $e_1$ ,  $e_2$  and  $e_3$  are constants which we defined so as to give meaningful bins even if  $p_{ic}(r) = p_{iw}(r) = p_{ie}(r) = 0$  for all  $i$  and  $r$ . Since we initialised the programs in the population with a high proportion of NOP operations, this ensured that even individuals in the first generation could obtain reasonable fitness levels. More specifically,  $c_1$ ,  $c_2$  and  $c_3$  were set to be the medians of the three bins chosen using the heuristic method described in (Poli et al., 2009) (where each bin gathered 30% of the response-time distribution), while  $w_1$ ,  $w_2$  and  $w_3$  were set to twice the standard deviation of the data in such bins. Standard deviations were estimated using the robust estimator provided by 1.4826 times the median absolute deviation from the median (or MAD for short) (Wilcox, 2005). Finally, the constants  $e_1$ ,  $e_2$  and  $e_3$  were all set to 0.5. This value is half-way between 0, which would give an perfectly rectangular bin, and 1, which gives bins a perfectly sinusoidal shape.

The system initialised the population as follows. All nine programs in an individual had identical length (50 instructions). The length was fixed, but

through the use of NOP instructions, the active code was effectively of variable size. The nine programs were concatenated, so effectively an individual was an array of 450 instructions. Programs were initially all made up only of NOP instructions, but they were immediately mutated with point mutation with a mutation rate of 8% so that on average approximately 4 instructions in each of the 9 programs were non-NOP. When an instruction was mutated, the instruction was replaced with a random instruction from the whole primitive set. These choices of parameters were based on some preliminary tests.

The system used tournament selection with tournament size 10. At each iteration, the system randomly decided whether to perform reevaluation of the fitness of an individual (keep in mind that our fitness function is noisy) or to create a new individual. It reevaluated fitness with probability 0.1 and performed crossover with a probability of 0.9. When fitness reevaluation was chosen, the new fitness value was blended with the old one using the formula:  $f = 0.8f_{old} + 0.2f_{new}$ . This effectively low-pass filters the fitness values using a simple IIR filter, thereby eventually leading to fitness values to stabilise around the expected value for each program. When crossover was performed, two parent individuals were selected, and 9-point crossover was performed. The 9 points were not constrained to fall within the 9 programs that form an individual. Crossover returned one offspring after each application. The offspring was mutated using point mutation with a mutation rate of 4% (so, on average each program was hit by two mutations) and then was evaluated. The offspring was then inserted in the population, replacing an individual which was selected using a negative tournament (with tournament size 10). Given the heavy computational nature of the task we used populations of size 1,000 and 5,000 and we performed 50 generations in each run. To see what kind of results could be obtained with smaller runs, we also performed runs with a population size of 50 run for 20 generations (for a total of 1,000 fitness evaluations).

The data used for our experiments were obtained as follows. We modified an experiment originally designed by (Esterman et al., 2004). In the experiment a composite stimulus is presented at a randomly chosen location (out of four possible locations) on a display for a very short time (between 50 and 150ms depending on conditions). The task of the subject is to identify whether the stimulus represented a target or a non-target stimulus. To correctly perform the task participants needed to identify and conjoin multiple features of the stimulus and then they needed to click a button to signal their response. While the participant performed the task they were connected to electroencephalographic equipment so that the waves generated during the task in different areas of the brain could be recorded. We used a BioSemi ActiveTwo system with 64 pre-amplified electrodes plus additional electrodes on the earlobes, the external canthi and infra-orbital positions. Signals were acquired at 2048 samples per second, were then bandpass-filtered between 0.15 and 40 Hz and, finally, were

down-sampled to 512 samples per second. We tested six students from the University of Essex, all with normal or corrected-to-normal vision. Each experiment lasted about one hour, and took about one further hour for preparation and for practice.

Trials were classified according to whether the target was present or absent and according to whether the response was ‘Correct’ or ‘Incorrect’. This resulted in four conditions: true positives (target present, correct response), true negatives (target absent, correct response), false positives (target absent, incorrect response) and false negatives (target present, incorrect response). For the tests reported in this paper we focused on the largest class, the True Negatives, which included a total of 2967 trials. We used epochs of approximately 1200ms (614 samples). That is, each trial contained a vector of 614 signal amplitude samples for each electrode. Each trial had an associated response/reaction time which represents the time lapsed between the presentation of the stimulus and the response provided by the user in the form of a mouse click. Following (Poli et al., 2009), the 10% of the trials with the longest response times were discarded. This left 2670 trials. In order to evaluate the fitness of an individual in the population, we needed to run the nine programs included in the individual on each of the trials in the dataset, i.e., the GP interpreter was invoked over 24,000 times before the fitness function could start executing.

With the fitness function defined in Section 4, the objective of evolution is to identify three membership functions which allow one to divide up this dataset into bins based on response times in such a way as to maximise the mutual statistical significance of differences in the bins’ amplitude averages. Note that evolution can choose to evolve functions that discard certain ranges of response times if this is advantageous.

With 3 bins (i.e., 3 bin-vs-bin comparisons), 64 electrodes and 614 samples per epoch evaluating our fitness function would require running 117,888 Kolmogorov-Smirnov tests per fitness evaluation. Since such tests are rather computationally expensive, we decided to scale down the problem by concentrating on one particular electrode (‘Pz’) and by further sub-sampling the amplitude data by a factor of 16. So, after performing the binning of the dataset, we needed to run the Kolmogorov-Smirnov test  $3 \times 38 = 114$  times per fitness evaluation.

## 6. Results

We show the response-time distribution recorded in our experiments for the True Negatives in Figure 1-2 (note that amplitudes have been normalised so that the curves are density functions; abscissas are in seconds). The boundaries of the 30%-quantile fixed-size bins produced with the method described in (Poli et al., 2009) are shown as vertical lines in Figure 1-2. The medians

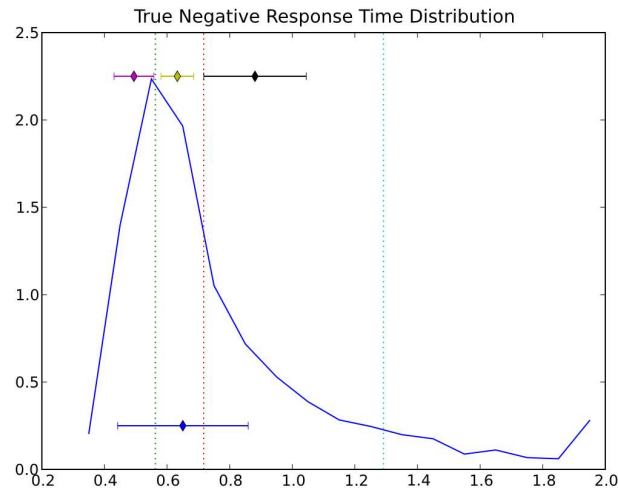


Figure 1-2. Response time distributions for true negative trials recorded in our experiments. Response times of 2000ms or longer have been grouped in the rightmost bin. The vertical lines within each plot represent the boundaries of the bins produced by the standard binning method described in (Poli et al., 2009). In each plot medians and standard deviations are also shown both for the bins (top) and for the overall distribution (bottom).

and standard deviations, estimated using MAD, for the whole distribution and for the bins are also shown in Figure 1-2. As indicated above, the objective of GP is to probabilistically divide up this distribution into bins using appropriate membership functions in such a way to maximise the statistical significance of bin averages.

The fitness value for the standard membership functions (rectangular bins) is approximately 0.8297, which corresponds to a mean Kolmogorov-Smirnov  $p$ -value of 0.1703. This implies that only for a fraction of the time steps in an epoch differences between bin averages are statistically significant at the standard confidence levels of 0.10 and 0.05. We want GP to improve on this.

We performed 50 runs with populations of size 50 and 1,000, and 10 runs with populations of size 5,000 on 182-core Linux cluster with Xeon CPUs. We report the mean, standard deviation, min and max of best fitnesses as well as the quartiles of the fitness distribution recorded in our experiments in Table 1-2. As one can see in all conditions, the method is very reliable, all standard deviations being very small. Even with the smallest population GP improved over the standard binning technique *in all runs*. This is particularly remarkable given that such runs required only approximately 2 minutes of CPU time each. Naturally, only runs with 1,000 and 5,000 individuals consistently achieved best

fitnesses close to or exceeding 0.9, which corresponds to average  $p$  values of 0.1 or less. This is a very significant improvement over the  $p$  value associated with rectangular bins. Now, for a large proportion of the time in an epoch differences between bin averages are statistically significant. CPU time was approximately 4 hours for runs of 1,000 individuals and approximately one day for runs of 5,000 individuals. Note that these long training times are not a problem in the domain of ERP analysis, since setting up an experiment, trialling it, then collecting the data with independent subjects, preparing the data for averaging and finally interpreting them after averaging require many weeks of work.

In order to achieve this high level of performance and reliability in the ERP binning problem, GP has discovered how to partition the data based on response times in such a way as to optimally balance two needs: (a) the need to include as many trials as possible in each bin so as to reduce noise in both variable-latency and fixed-latency ERP components, and (b) the need to make the bins as narrow as possible so as to reduce the systematic errors associated with averaging variable-latency components.

Table 1-2. Mean, standard deviation, min and max of best and quartiles of the fitness distribution recorded in out experiments.

Population size 50, 20 generations					
Statistic	Best	Qrtl 1	Qrtl 2	Qrtl 3	Qrtl 4
Mean	0.87750	0.86354	0.86020	0.85613	0.17514
StdDev	0.008868	0.006952	0.007123	0.008249	0.272409
Max	0.900335	0.877868	0.876486	0.872651	0.753881
Min	0.855929	0.845703	0.842577	0.837546	0.000000
Population size 1,000, 50 generations					
Statistic	Best	Qrtl 1	Qrtl 2	Qrtl 3	Qrtl 4
Mean	0.89862	0.88161	0.88056	0.87910	0.00000
StdDev	0.00396	0.00293	0.00288	0.00307	0.00000
Max	0.91348	0.89096	0.88979	0.88922	0.00000
Min	0.89197	0.87720	0.87526	0.87346	0.00000
Population size 5,000, 50 generations					
Statistic	Best	Qrtl 1	Qrtl 2	Qrtl 3	Qrtl 4
Mean	0.90431	0.88301	0.88214	0.88091	0.00000
StdDev	0.0060682	0.0039270	0.0038899	0.0040199	0.00000
Max	0.91763	0.89148	0.89053	0.88947	0.00000
Min	0.89914	0.88039	0.87956	0.87794	0.00000

As an example, we plot the best evolved bin membership functions in the 50 runs with a population of 1,000 individuals in Figure 1-3. These correspond to the following equations:

$$\mathcal{P}_1(r) = \left( \text{pcos} \left( \frac{r - 0.394}{0.127} \right) \right)^{0.375} \quad (1.3)$$

$$\mathcal{P}_2(r) = \left( \text{pcos} \left( \frac{r - 0.633}{0.129 - 0.5r} \right) \right)^{0.4+0.5r} \quad (1.4)$$

$$\mathcal{P}_3(r) = \left( \text{pcos} \left( \frac{r - 1.381}{0.327} \right) \right)^{0.688} \quad (1.5)$$

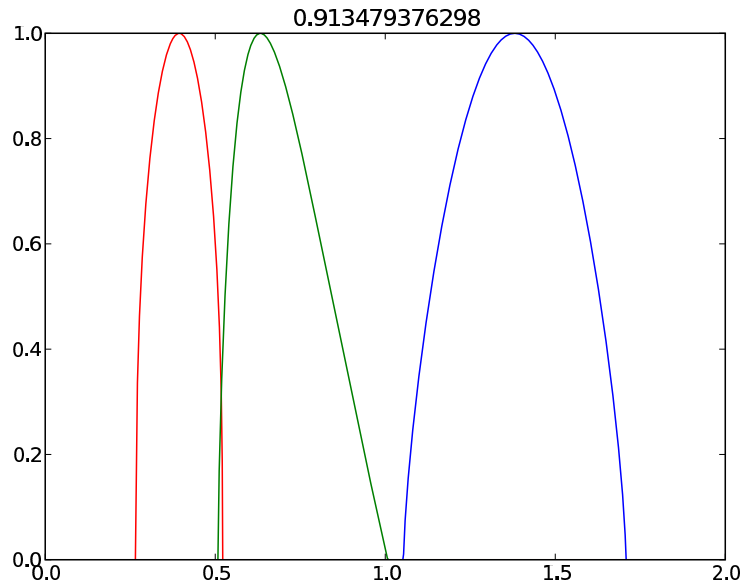


Figure 1-3. Best membership functions evolved in 50 runs with a population of 1,000 individuals.

These were obtained by analysing and then symbolically simplifying the nine programs making up the best individual evolved in such runs. The listing of the nine programs is shown in Table 1-3.

Table 1-3. The nine programs forming the best evolved solution to the ERP binning problem (NOP instructions have been edited out).

Bin 1			Bin 2			Bin 3		
Centre	Width	Exponent	Centre	Width	Exponent	Centre	Width	Exponent
r1 <- 0	rs <-> r1	r1 <- r1 + ri	r1 <- r1 + ri	r1 <- -.5	r1 <- -.5	r0 <- -1	r1 <- 1	r1 <- -r1
r1 <- -.5	r0 <- r0 + ri	r0 <- r0 + r1	rs <-> r0	r0 <- -r0	r1 <- r1 + ri	r0 <- r0 + r1	r1 <- r0 + r1	r0 <- -r0
r1 <- .1	rs <-> r1	r0 <- .5	r0 <- r0 * r1	r1 <- 0	r0 <- r0 + ri	r1 <- .1	r1 <- -.5	r1 <- r1 * r1
r0 <- .5	r0 <- r0 + ri	rs <-> r0	rs <-> r0	r0 <- -r0	r1 <- -r1	r0 <- 0	r1 <- r1 + ri	r1 <- 0
r0 <- r0 + ri	r1 <- .1	r1 <- 0	r1 <- .1	r0 <- r0 + r1	rs <-> r1	r0 <- r0 + r1	r0 <- r0 * r1	r0 <- .5
r0 <- -1	r0 <- r0 * r0	rs <-> r0	r1 <- 1	r0 <- -r0	r0 <- r0 * r0	r0 <- 0	r1 <- r0 + r1	r0 <- r0 * r1
r1 <- r0 + r1	r1 <- r0 + r1	r1 <- r1 + ri	r0 <- r0 + r1	r1 <- r1 + ri	r1 <- 0	r1 <- -r1	r1 <- r0 + r1	r1 <- r0 + r1
r0 <- r0 + r1	r1 <- -.5	r0 <- -r0	r1 <- 0	r1 <- -.5	r0 <- -.1	r0 <- -r0	r0 <- r0 * r0	r1 <- 1
r0 <- r0 * r1	r0 <- .5	rs <-> r1	r0 <- r0 * r0	r0 <- r0 + r1	r1 <- .1	r1 <- r0 * r1	r1 <- 1	r1 <- r1 * r1
r1 <- r1 * r1	r0 <- 0	r1 <- r1 + ri	r1 <- -r1	r1 <- r0 + r1	r0 <- .5	r1 <- r1 + ri	r0 <- 0	r0 <- r0 * r1
r0 <- r0 + ri	r1 <- r1 + ri	r0 <- r0 + ri	rs <-> r1	r0 <- r0 * r1	rs <-> r1	rs <-> r0	r0 <- -1	r1 <- r1 + ri
r0 <- -.1	r1 <- -r1	r0 <- -r0	r0 <- -r0	r1 <- -.5	r0 <- r0 * r0	r0 <- r0 + ri	r0 <- r0 * r0	r0 <- -r0
r1 <- 0	r1 <- r1 * r1	r0 <- r0 + r1	r1 <- 1	r0 <- .5	r1 <- 1	r0 <- 0	r0 <- .5	r1 <- r0 + r1
r1 <- r1 * r1	r1 <- 0	r0 <- r0 * r1	r1 <- 0	r0 <- -.1	r0 <- -r0	r0 <- -.1	r0 <- r0 + ri	rs <-> r1
r0 <- r0 * r0	r1 <- -r1	r0 <- r0 + r1	r0 <- r0 + r1	rs <-> r0	rs <-> r0	r0 <- -r0	r0 <- r0 * r0	r0 <- -1
r1 <- -r1	r0 <- r0 * r1	r1 <- r1 * r1	r1 <- -r1	r1 <- -.5	r0 <- r0 * r0	r1 <- -.5	r1 <- r1 + ri	r1 <- -.5
r0 <- r0 + ri	r0 <- -.1	r1 <- r1 * r1	rs <-> r1	r0 <- -.1	r0 <- -1	r0 <- .5	r1 <- -r1	r0 <- r0 + r1
rs <-> r1	rs <-> r1	r1 <- r0 + r1	r0 <- -r0	r0 <- r0 * r1	r1 <- r0 + r1	r1 <- r0 + r1	r0 <- -.1	r0 <- .5
r0 <- r0 * r0	r0 <- 0	rs <-> r0	r1 <- 1	r1 <- r0 * r1	r0 <- -.1	r0 <- r0 + ri	r0 <- -.1	r0 <- r0 * r0
r0 <- r0 * r1	r1 <- .1	r0 <- r0 * r0	r0 <- r0 + r1	r0 <- r0 + ri	r1 <- -r1	r0 <- r0 + r1	r1 <- r1 * r1	r0 <- -1
r1 <- -.5	r1 <- 1	r0 <- .5	r1 <- 1	r1 <- 0	r1 <- 0	r0 <- -1	r0 <- r0 + ri	r0 <- .5
r1 <- .1	r1 <- r1 + ri	r0 <- -r0	rs <-> r1	r1 <- r0 * r1	r0 <- -.1	r1 <- 1	r0 <- 0	r1 <- r1 * r1
r1 <- -r1	r1 <- .1	r1 <- r1 + ri	r1 <- r1 + ri	r1 <- r1 + ri	r0 <- -.1	r1 <- .1	r1 <- 1	r1 <- -r1
r0 <- r0 * r0	r1 <- 1	r0 <- r0 * r0	r0 <- r0 * r1	r1 <- r1 + ri	r0 <- .5	rs <-> r0	r1 <- 1	r1 <- r0 + r1
r1 <- 0	r0 <- r0 * r0	r1 <- -r1	r1 <- .1	r0 <- -r0	r1 <- 0	r0 <- -r0	r0 <- r0 + r1	r0 <- r0 * r0
r0 <- r0 + ri	r1 <- r1 + ri	r1 <- 1	r0 <- 0	r1 <- .1	r0 <- .1	r0 <- 0	r0 <- 0	r0 <- -r0
r0 <- 0	r1 <- -r1			r1 <- r0 * r1	r0 <- r0 + r1	r0 <- r0 + ri	r0 <- -r0	r1 <- r0 + r1
r1 <- .1	r0 <- r0 * r0			r1 <- r1 * r1	r0 <- -1	r0 <- r0 + r1	r1 <- .1	rs <-> r1
r0 <- -r0	r0 <- r0 * r0			r1 <- 0	r0 <- r0 * r0	r0 <- .5		rs <-> r0
r0 <- r0 * r1	r0 <- 0			r1 <- -1	r1 <- r0 * r1	r1 <- r1 * r1		rs <-> r1
r0 <- -r0	r1 <- 1			r0 <- 0	r0 <- 0	r1 <- r0 + r1		
r0 <- -.1	r0 <- -r0			r1 <- .5	r1 <- r1 + ri	r1 <- r1 * r1		
r1 <- -r1				r0 <- -.1	r0 <- -.1			
				r0 <- r0 + ri	r0 <- r0 + ri			
				r1 <- 0	r1 <- 0			
				r1 <- -.5	r1 <- -.5			
				r1 <- -r1	r1 <- -r1			

The ERP averages produced by this solution are shown in Figure 1-4. For reference we show the averages obtained with traditional rectangular bins in Figure 1-5. As one can see the ERP averages for the middle bins are almost identical to the full average in both cases. This is because both the reference bin and the GP-evolved bin capture the median response time and surrounding samples, which are representative of the central tendency of the whole distribution. However, when comparing the ERP averages for bins 1 and 3 with the corresponding reference averages, we see that the membership functions evolved by GP are more selective in their choice of trials. This produces bigger (and hence more statistically significant) variations between groups. Particularly interesting is the case of bin 3, which, with the standard binning method, is adjacent to bin 2 and is very broad. This led to averaging ERP components having an excessively wide distribution of latencies, leading to an ERP average where late endogenous components, which are typically associated with the preparation of the response, are hardly discernible. Instead, GP has produced a much narrower bin 3 and a large gap between bins 2 and 3. As one can see from Figure 1-4, this yields a much clearer representation of such late potentials.

## 7. Conclusions

In this paper we used a multi-program form of register-based GP to discover probabilistic membership functions for the purpose of binning and then averaging ERP trials based on response times. The objective was to evolve membership functions which could significantly improve the mutual statistical significance of bin averages thereby capturing more accurately true brain waves than when using simple rectangular bins.

Our results are very encouraging. GP can consistently evolve membership functions that almost double the statistical significance of the ERP bin averages with respect to the standard binning method.

In future work we will test the generality of evolved solution, by applying the bins found by GP to newly acquired (unseen) data. We also intend to make use of our new bin averaging technique in BCIs. Indeed, the work presented in this paper originated from the need to understand in exactly what ways stimulus features and task complexity, as well as cognitive errors, modulate ERP components in BCI (Cinel et al., 2004; Citi et al., 2004; Citi et al., 2008). Our long term objective is to formally link quantitative psychological models of feature binding and perceptual errors (Humphreys et al., 2000; Cinel et al., 2002; Cinel and Humphreys, 2006) with the presence of specific ERP components and the modulation of their latency and amplitude. This knowledge could then be used to design a new generation of BCIs where the behaviour and features of human cognitive systems are best exploited.

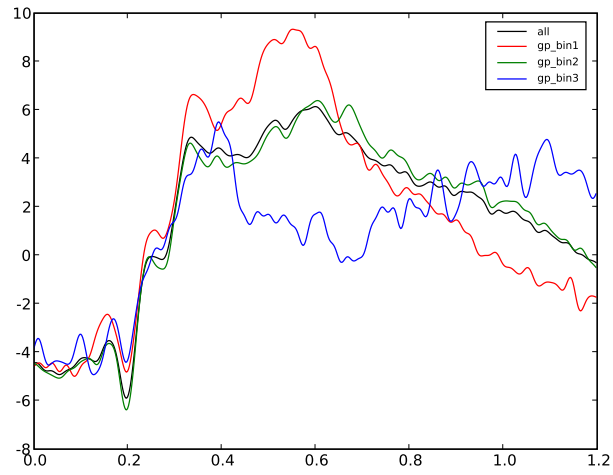


Figure 1-4. Averages obtained with the GP-evolved bin membership functions in Equations 1.3–1.5 and shown in Figure 1-3.

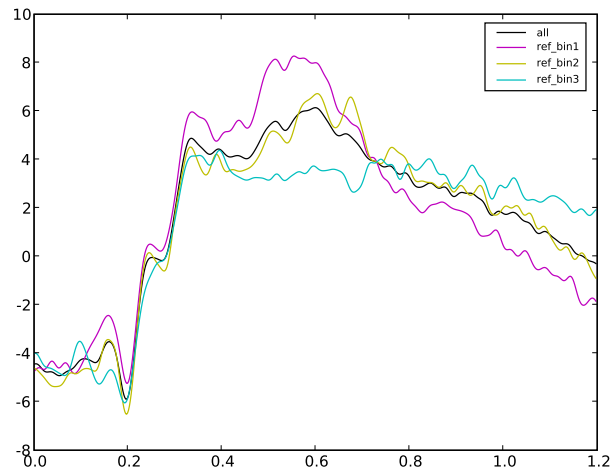


Figure 1-5. Averages obtained with traditional rectangular bins.

## Acknowledgements

We would like to thank the Engineering and Physical Sciences Research Council (grant EP/F033818/1) and by the Experimental Psychological Society (UK) (grant “Binding Across the Senses”) for financial support and Francisco Sepulveda for helpful comments and suggestions.

## References

- 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.
- Bonala, Bharat, Boutros, Nashaat N, and Jansen, Ben H (2008). Target probability affects the likelihood that a P300 will be generated in response to a target stimulus, but not its amplitude. *Psychophysiology*, 45(1):93–99.
- Bostanov, Vladimir (2004). BCI competition 2003–data sets Ib and IIB: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. *IEEE transactions on bio-medical engineering*, 51(6):1057–1061.
- Cinèl, Caterina and Humphreys, Glyn W (2006). On the relations between implicit and explicit spatial binding: evidence from Balint’s syndrome. *Cognitive, affective & behavioral neuroscience*, 6(2):127–140.
- Cinèl, Caterina, Humphreys, Glyn W, and Poli, Riccardo (2002). Cross-modal illusory conjunctions between vision and touch. *Journal of experimental psychology. Human perception and performance*, 28(5):1243–1266.
- Cinèl, Caterina, Poli, Riccardo, and Citi, Luca (2004). Possible sources of perceptual errors in P300-based speller paradigm. *Biomedizinische Technik*, 49:39–40. Proceedings of 2nd International BCI workshop and Training Course.
- Citi, L., Poli, R., Cinèl, C., and Sepulveda, F. (2008). P300-based BCI mouse with genetically-optimized analogue control. *IEEE transactions on neural systems and rehabilitation engineering*, 16(1):51–61.
- Citi, Luca, Poli, Riccardo, and Sepulveda, Francisco (2004). An evolutionary approach to feature selection and classification in P300-based BCI. *Biomedizinische Technik*, 49:41–42. Proceedings of 2nd International BCI workshop and Training Course.
- Donchin, E and Coles, M G H (1988). Is the P300 a manifestation of context updating? *Behavioral and Brain Sciences*, 11:355–372.
- Donchin, Emanuel and Lindsley, Donald B., editors (1968). *Average Evoked Potentials: Methods, Results, and Evaluations*, number NASA SP-191, San Francisco, California. NASA, NASA.

- Esterman, Michael, Prinzmetal, William, and Robertson, Lynn (2004). Categorization influences illusory conjunctions. *Psychonomic bulletin & review*, 11(4):681–686.
- Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical neurophysiology*, 70(6):510–523.
- Furdea, A., Halder, S., Krusienski, D. J., Bross, D., Nijboer, F., Birbaumer, N., and KÄ¼bler, A. (2009). An auditory oddball (P300) spelling system for brain-computer interfaces. *Psychophysiology*.
- Handy, Todd C., editor (2004). *Event-related potentials. A Method Handbook*. MIT Press.
- Humphreys, G. W., Cinel, C., Wolfe, J., Olson, A., and Klempe, N. (2000). Fractionating the binding process: neuropsychological evidence distinguishing binding of form from binding of surface features. *Vision research*, 40(10-12):1569–1596.
- Keus, Inge Martine, Jenks, Kathleen Marie, and Schwarz, Wolf (2005). Psychophysiological evidence that the SNARC effect has its functional locus in a response selection stage. *Brain research. Cognitive brain research*, 24(1):48–56.
- Lindsley, Donald B. (1968). Average evoked potentials – achievements, failures and prospects. In Donchin, Emanuel and Lindsley, Donald B., editors, *Average Evoked Potentials: Methods, Results, and Evaluations*, chapter 1. NASA.
- Luck, S. J. and Hillyard, S. A. (1990). Electrophysiological evidence for parallel and serial processing during visual search. *Perception & psychophysics*, 48(6):603–617.
- Luck, Stephen J. (2005). *An introduction to the event-related potential technique*. MIT Press, Cambridge, Massachusetts.
- McPhee, Nicholas F. and Poli, Riccardo (2008). Memory with memory: Soft assignment in genetic programming. In Keijzer, Maarten, Antoniol, Giuliano, Congdon, Clare Bates, Deb, Kalyanmoy, Doerr, Benjamin, Hansen, Nikolaus, Holmes, John H., Hornby, Gregory S., Howard, Daniel, Kennedy, James, Kumar, Sanjeev, Lobo, Fernando G., Miller, Julian Francis, Moore, Jason, Neumann, Frank, Pelikan, Martin, Pollack, Jordan, Sastry, Kumara, Stanley, Kenneth, Stoica, Adrian, Talbi, El-Ghazali, and Wegener, Ingo, editors, *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1235–1242, Atlanta, GA, USA. ACM.
- Pfurtscheller, Gert, Flotzinger, Doris, and Kalcher, Joachim (1993). Brain-computer interface: a new communication device for handicapped persons. *Journal of Microcomputer Applications*, 16(3):293–299.

- Poli, Riccardo, Cinel, Caterina, Citi, Luca, and Sepulveda, Francisco (2009). Reaction-time binning: a simple method for increasing the resolving power of erp averages. *Submitted*.
- Poli, Riccardo, Langdon, William B., and McPhee, Nicholas Freitag (2008). *A field guide to genetic programming*. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>. (With contributions by J. R. Koza).
- Polich, John and Comerchero, Marco D (2003). P3a from visual stimuli: typicality, task, and topography. *Brain topography*, 15(3):141–152.
- Rakotomamonjy, Alain and Guigue, Vincent (2008). BCI competition III: dataset II- ensemble of SVMs for BCI P300 speller. *IEEE transactions on bio-medical engineering*, 55(3):1147–1154.
- Spencer, Kevin M (2004). Averaging, detection and classification of single-trial erps. In Handy, Todd C., editor, *Event-related potentials. A Method Handbook*, chapter 10. MIT Press.
- Töllner, Thomas, Gramann, Klaus, Müller, Hermann J, Kiss, Monika, and Eimer, Martin (2008). Electrophysiological markers of visual dimension changes and response changes. *Journal of experimental psychology. Human perception and performance*, 34(3):531–542.
- Wagner, P., Roschke, J., Grozinger, M., and Mann, K. (2000). A replication study on P300 single trial analysis in schizophrenia: confirmation of a reduced number of 'true positive' P300 waves. *Journal of psychiatric research*, 34(3):255–259.
- Wilcox, Rand R. (2005). *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, second edition.
- Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., and Vaughan, T. M. (2000). Brain-computer interface technology: a review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, 8(2):164–173.
- Wolpaw, J. R., McFarland, D. J., Neat, G. W., and Forneris, C. A. (1991). An EEG-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology*, 78(3):252–259.
- Zhang, J. (1998). Decomposing stimulus and response component waveforms in ERP. *Journal of neuroscience methods*, 80(1):49–63.