

Escola de Engenharia de São Carlos  
Universidade de São Paulo

Relatório Final de Iniciação Científica

# Classificação de error-related potentials utilizando algoritmos baseados em aprendizagem de máquina

Bolsista: Marco Aurélio Martins Mijam  
Orientador: Ivan Nunes da Silva

São Carlos  
8 de Agosto de 2018

# 1 Proposta

Animais humanos e não-humanos possuem a habilidade de aprender e se adaptar às mais diversas situações, essa habilidade é em grande parte devido à capacidade de se identificar ações errôneas. Diversos estudos mostram que esses eventos evocam respostas neurais distintas que podem ser observadas utilizando técnicas de neuroimagem, como a eletroencefalografia (EEG) por exemplo. Foi demonstrado que as assinaturas eletrofisiológicas desses eventos de identificação de erros, chamadas ErrPs (*error-related potentials*), podem ser decodificadas com sucesso para serem utilizadas em uma interface cérebro-computador (ICC) (1).

O objetivo desse trabalho é pesquisar a aplicabilidade dos ErrPs como sinal de realimentação de um sistema que deve ser controlado por um operador de maneira arbitrária a ele. O sistema utilizado será um software gráfico que mostrará na tela um pequeno quadrado que tem a capacidade de se mover tanto no eixo x quanto no eixo y. Um algoritmo de controle enviará o comando para o quadrado se mover baseado em quatro teclas do teclado que o operador irá utilizar. Antes do início do experimento o operador deve definir mentalmente qual a direção que ele quer que o quadrado se mova para cada uma das teclas que está disponível para ele pressionar, em seguida ele deve tentar mover o quadrado utilizando as teclas. Como o algoritmo de controle no início do experimento não vai estar calibrado, o quadrado terá uma chance pequena de se mover na mesma direção que o operador pretendia, caso o quadrado se mover na direção errada, um sinal de ErrP será detectado e enviado para o algoritmo de controle, que terá seus parâmetros ajustados para evitar que o sinal de ErrP seja gerado novamente. Espera-se deste modo que o algoritmo de controle aprenda depois de algum tempo, a função das teclas que o operador definiu, sem nenhuma intervenção direta, apenas com a análise dos ErrPs.

## 2 Pesquisa inicial

Foi feita uma revisão bibliográfica de artigos relacionados aos temas principais da pesquisa, como classificação de sinais, EEG e ErrP, assim como está sendo estudada a documentação da ferramenta MNE (2) que será utilizada para o processamento e visualização dos dados de EEG.

### 2.1 *Error-related potential*

Foram descobertos diversos tipos de ErrPs, tais como quando alguém tem que fazer alguma atividade em que tem que escolher entre diversas alternativas pressionado para se decidir rápido, e então percebe que fez a escolha errada (ErrPs de resposta); quando alguém recebe um feedback indicando que cometeu um erro (ErrP de feedback ou de reforço); quando alguém percebe um erro cometido por outra pessoa (ErrP de observação), ou quando alguém envia um comando para uma máquina, e a máquina executa outra ação (ErrP de interação) (3). Para este trabalho, visto as características do problema que envolve enviar um comando para o computador e checar quando o comando não é obedecido, o tipo de ErrP que será focado é o ErrP de interação, que é o que mais se adequa ao caso.

O ErrP de interação, de acordo com os experimentos em (4), apresenta um primeiro pico positivo de tensão 200 ms após o feedback, um pico negativo em 250 ms, um positivo novamente em 320 ms e então um último pico negativo cerca de 450 ms após o feedback. Embora experimentos feitos em pessoas diferentes tenham mostrado certa variância na amplitude dos sinais, o atraso temporal dos picos positivos

e negativos do sinal com relação ao feedback se mostrou similar em todas as pessoas.

## 2.2 Classificação de sinais

A classificação dos ErrPs a partir dos sinais de EEG mostra-se mais precisa quando feita com os dados no domínio da frequência, em vez de no domínio do tempo (1), e como a informação temporal é importante para este tipo de análise de sinal, foi escolhido utilizar a transformada de wavelet no sinal, para se obter as informações de frequência e tempo dele, que serão utilizadas na classificação do ErrP.

Alguns artigos foram escolhidos a fim de se estudar diversas técnicas de classificação, vendo suas vantagens e desvantagens, assim como sua aplicabilidade para o problema proposto neste trabalho.

Existem diversas categorias de algoritmos de classificação, como classificadores lineares, redes neurais artificiais, *nearest neighbour* e combinações de classificadores.

### 2.2.1 Classificadores lineares

Classificadores lineares são algoritmos discriminantes que utilizam funções lineares para distinguir as classes.

Existem dois tipos principais de classificadores lineares utilizados para classificação em ICC: *linear discriminant analysis* (LDA) e *support vector machine* (SVM). Outro classificador linear que vale a pena destacar é a *regularized Fisher's LDA* (RFLDA) que apesar de não ser tão utilizado em ICC quanto o LDA e o SVM, possui uma capacidade maior de generalização quando comparado ao LDA, proporcionando melhores resultados para ICC (5).

### 2.2.2 Redes neurais artificiais

Junto com os classificadores lineares, as redes neurais artificiais (RNA) são as categorias de classificadores mais utilizadas em ICC (5). Redes neurais artificiais são compostas de neurônios artificiais interligados, que nos possibilitam produzir bordas de decisão não lineares. A RNA mais utilizada em ICC é o perceptron multicamadas (PMC) que é composto de várias camadas de neurônios, sendo uma camada de entrada, possivelmente uma ou mais camadas escondidas e uma camada de saída.

Redes neurais artificiais são aproximadores universais, pois quando compostas de número suficiente de neurônios e de camadas, elas podem aproximar qualquer função contínua, assim como podem classificar qualquer número de classes (5). Devido ao fato das RNA serem aproximadores universais, isso faz com que elas sejam sensíveis a sobre treinamento, especialmente para dados ruidosos e não estacionários como EEG, portanto é necessário uma seleção cuidadosa da arquitetura e regularização.

### 2.2.3 *k*-Nearest neighbour

O *k*-nearest neighbour (*k*-NN), é um classificador não paramétrico, não linear, e relativamente simples. Ele classifica uma nova amostra com base na medida da "distância" entre ela e das amostras do conjunto de treinamento.

Algoritmos de classificação do tipo *k*-NN não são muito populares em aplicações envolvendo ICC, devido ao fato de ser conhecido que esse tipo de algoritmo perde sua eficiência em sistemas com muitas dimensões. No entanto, quanto utilizado em sistemas de ICC com vetores de características de baixa dimensão o *k*-NN pode se mostrar eficiente (5).

### 2.2.4 Combinações de classificadores

Uma forma de se realizar classificação é com a utilização de diversos classificadores em conjunto, agregados em diferentes formas, elas são as seguintes:

- *Boosting*

Nesta forma os classificadores são agregados em cascata, onde cada classificador se foca nos erros cometidos pelo anterior (5).

- *Voting*

Ao utilizar o voting, diversos classificadores são utilizados para classificar um vetor de parâmetros de entrada. A classe final vai ser a que a maioria dos classificadores encontrou.

Voting é a maneira mais popular de se combinar classificadores em pesquisa envolvendo ICC devido a sua simplicidade e eficiência principalmente (5).

- *Stacking*

Esta forma de se agrupar os classificadores consiste em classificar um vetor de parâmetros de entrada com diversos classificadores, esses classificadores são chamados de classificadores de nível-0, então as saídas desses classificadores são fornecidas como entrada para um meta-classificador (ou classificador de nível-1) que decide a classe final a ser escolhida (5).

## 3 Descrição do dataset utilizado

Foi utilizado o dataset (6) neste artigo, onde dados de EEG foram gravados de 10 pessoas diferentes enquanto jogavam um jogo de computador em que erros no controle do jogo aconteciam propositalmente a cada intervalo de tempo.

Os participantes do teste jogaram um jogo em que eles podiam controlar o movimento de um círculo vermelho na tela através de um controle. O objetivo do jogo é desviar o maior tempo possível de faixas azuis que se movem de cima para baixo na tela, como pode-se ver na figura 1.

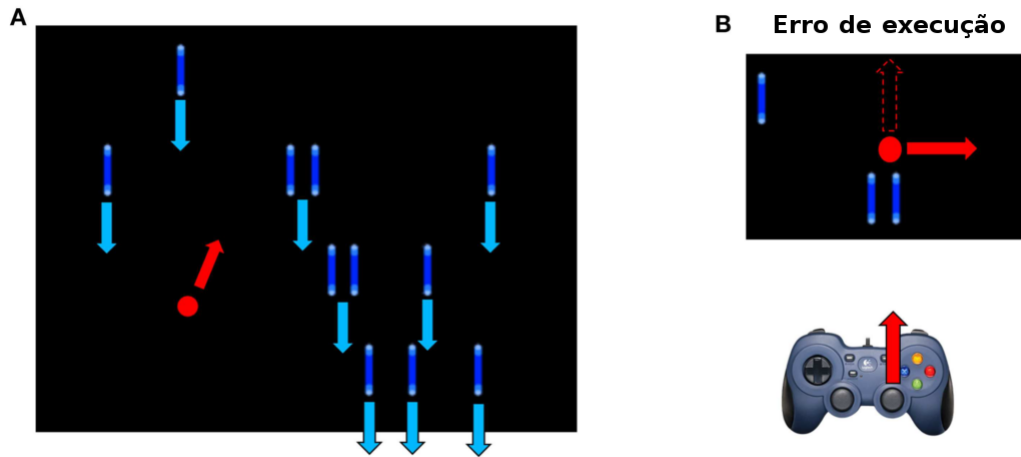


Figura 1: (A) Interface gráfica do jogo onde o usuário controla o círculo vermelho e tem que desviar das linhas azuis. (B) Erro de execução provocado em um momento aleatório pelo jogo, fazendo com que o círculo se mova em uma direção diferente da que o usuário deseja.

Para se estudar o ErrP, a cada intervalo de 5 a 8 segundos, definido aleatoriamente, um círculo se move em uma direção aleatória quando o participante envia um comando de movimentação para ele, por um período de 2 segundos, sendo que a posição é escolhida aleatoriamente entre  $45^\circ$ ,  $90^\circ$  e  $180^\circ$  para ambos os lados proporcionando assim um feedback errôneo ao participante (7).

## 4 Resultados

A partir do dataset obtido com pessoas realizando a tarefa descrita no capítulo 3, foram separados trechos de dados no intervalo de 1 segundo após ser enviando um comando para o jogo no período em que ele está programado para enviar uma resposta aleatória a um comando de direção. Apenas os eletrodos Cz e FCz do sistema 10-20 foram utilizados, os outros foram ignorados pois o ErrP é caracterizado por uma distribuição fronto-central na linha média do crânio (4).

O sinais passaram por uma filtro passa-faixa de 1-10 Hz para remoção de artefatos e ruídos.

Além dos trechos de dados contendo sinais de ErrP, foram retirados trechos em outros momentos aleatórios, chamados neste documento de sinais normais, que foram utilizados no treinamento para os classificadores conseguirem separar um ErrP de um sinal genérico qualquer.

Após a separação dos sinais, foi gerado um novo sinal composto pelo sinal do eletrodo FCz e Cz, um em sequência do outro, para o mesmo ErrP, alguns exemplos dos quais podem ser vistos na figura 2. O mesmo processo foi feito para os sinais normais, na figura 3 encontra-se alguns exemplos desses sinais.

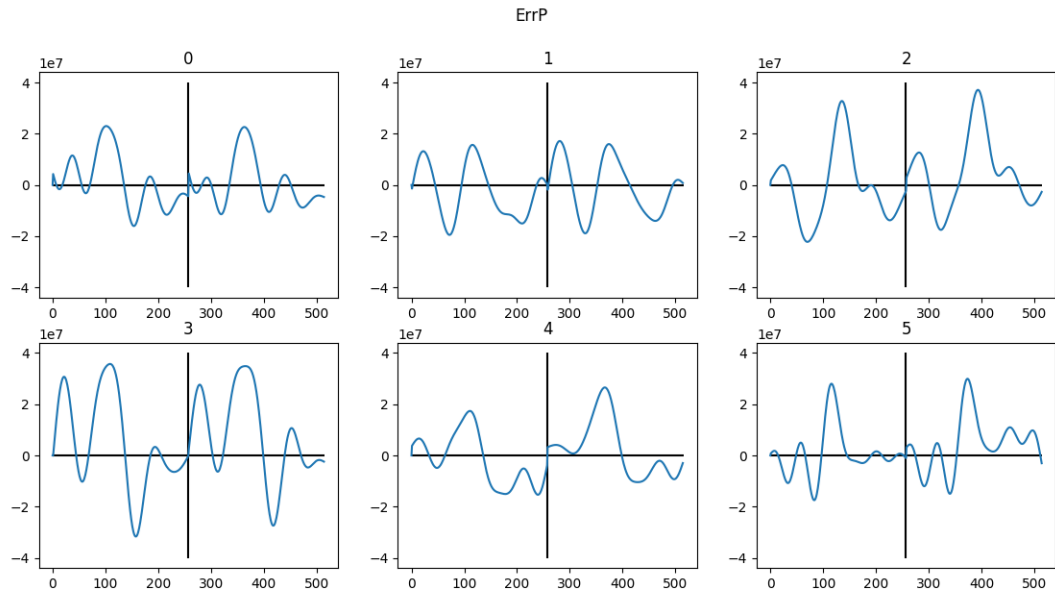


Figura 2: Alguns dos sinais ErrP que foram amostrados a partir do dataset utilizado.

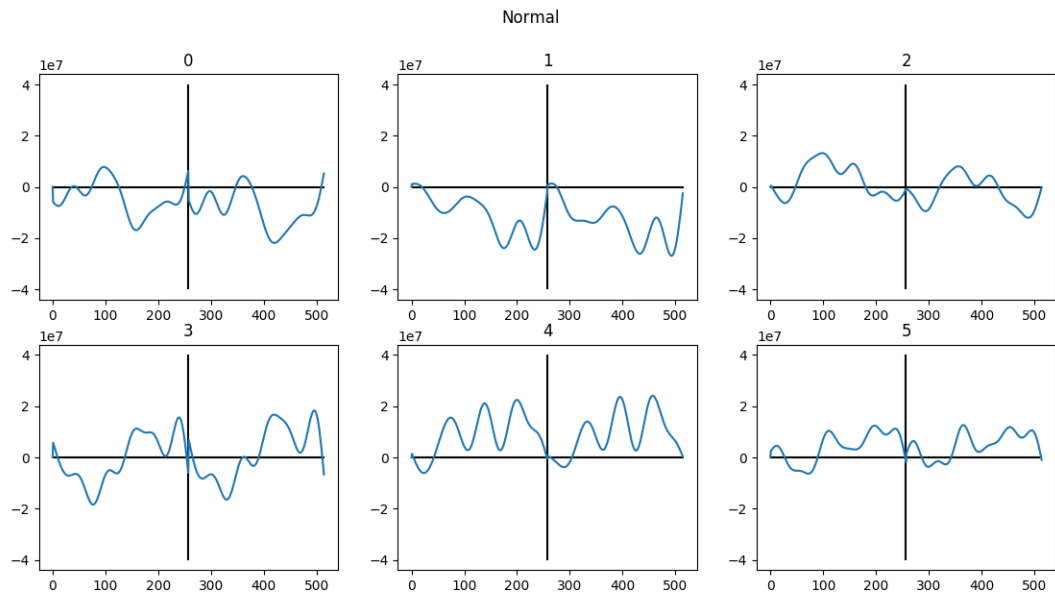


Figura 3: Alguns dos sinais normais que foram amostrados a partir do dataset utilizado.

A partir desses sinais, foram criados 5 databases para serem utilizados no treinamento dos classificadores:

- Haar final

Este database foi criado a partir da aplicação da transformada de Haar nos dados, e utilizando apenas a última camada da transformada.

- Haar complete

Este database foi criado a partir da aplicação da transformada de Haar nos dados, e utilizando todas as 8 camadas da transformada.

- Dominio do tempo

Este database foi criado a partir dos sinais puros, sem modificação alguma.

- RFFT

Este database foi criado a partir do calculo da Transformada de Fourier dos dados utilizando o algoritmo FFT, e então, por o resultado ser um sinal simétrico, foi desprezado metade dele para evitar redundâncias.

- FFT

Este database é exatagem igual o RFFT, porém sem desprezar os dados redundantes.

Além desses databases diferentes, foram utilizadas 80 diferentes topologias de redes neurais para classificação, dentre elas foi tentado topologias triangulares (mais neurônios nas primeiras camadas, e menos a cada camada em direção à camada de saída) e quadradas (todas as camadas com o mesmo número de neurônios), nesses tipos de topologias foi utilizado redes neurais com uma camada interna, duas e com nenhuma camada internada. Com relação ao número de neurônios, eles foram variados arbitrariamente, desde um total de 30 neurônios por rede neural para até 8000 neurônios no total por rede neural, fornecendo assim uma grande variedade de topologias diferentes para comparação. Em todas as topologias, a função de ativação utilizado foi a tangente hiperbólica.

Cada topologia de rede neural foi treinada utilizando cada database, e para cada combinação de database e topologia diferente, foi realizado 10 treinamentos variando aleatoriamente os dados que foram utilizados para treinamento e para teste da rede neural, e então feito a média dos resultados dos 10 treinamentos.

Os resultados foram avaliados através de um indicador que é caracterizado pela soma da taxa de acerto na classificação da rede neural após o treinamento com o quanto aumentou a taxa de acerto na classificação após o treinamento em relação ao início do treinamento.

Os melhores resultados obtidos, de acordo com o indicador utilizado, estão exibidos na tabela 1.

Melhoria	Acerto Inicial	Acerto Final	Indicador	Topologia	Dataset
0,313	0,510	0,823	1,137	200; 200; 200; 1	Domínio do tempo
0,290	0,543	0,833	1,123	175; 175; 1	Haar final
0,303	0,510	0,813	1,117	200; 100; 50; 1	Haar final
0,293	0,517	0,810	1,103	3000; 1	Haar completa
0,293	0,513	0,807	1,100	250; 125; 1	Domínio do tempo
0,313	0,470	0,783	1,097	600; 1	Haar final
0,277	0,540	0,817	1,093	4000; 1	Haar final
0,313	0,463	0,777	1,090	70; 35; 1	Haar final
0,300	0,487	0,787	1,087	100; 100; 100; 1	Haar final
0,293	0,500	0,793	1,087	150; 50; 20; 1	Haar final
0,270	0,540	0,810	1,080	250; 125; 1	Haar final
0,303	0,473	0,777	1,080	150; 75; 37; 1	Haar final
0,273	0,533	0,807	1,080	6000; 1	Haar final
0,290	0,497	0,787	1,077	100; 50; 25; 1	Domínio do tempo
0,290	0,490	0,780	1,070	100; 25; 1	Haar final
0,273	0,513	0,787	1,060	3000; 1	Domínio do tempo
0,290	0,480	0,770	1,060	50; 20; 1	Domínio do tempo
0,277	0,503	0,780	1,057	300; 100; 30; 1	Domínio do tempo
0,277	0,503	0,780	1,057	800; 1	Domínio do tempo
0,300	0,457	0,757	1,057	100; 100; 1	Haar final
0,287	0,483	0,770	1,057	300; 100; 1	Haar final
0,273	0,510	0,783	1,057	150; 1	Haar final
0,293	0,470	0,763	1,057	175; 175; 1	Domínio do tempo
0,287	0,480	0,767	1,053	2000; 1	Haar final
0,300	0,453	0,753	1,053	50; 1	Haar final
0,283	0,487	0,770	1,053	80; 1	Haar final
0,260	0,530	0,790	1,050	600; 200; 80; 1	Haar final

Tabela 1: Melhores resultados obtidos onde o Acerto Inicial indica a taxa de acerto do classificador antes de ser treinado; o Acerto Final indica a taxa de acerto do classificador após ser treinado; a Melhoria é calculada pela subtração do Acerto Final pelo Acerto Inicial; a Topologia tem a informação de quantas camadas foram utilizadas (quantidade de números na célula) e quantos neurônios havia em cada camada, sendo que a última camada sempre possui um neurônio pois é a camada de saída da rede neural.

Um resultado perceptível é que os treinamentos realizados utilizando os banco de dados que passaram pela FFT e RFFT tiveram os piores resultados, e nenhum deles constou entre os 28 melhores que estão na tabela 1.

## 5 Conclusão

Com a utilização dos ErrPs, os computadores podem aprender a tomar decisões que muitas vezes, devido à subjetividade de sua natureza ou nível de abstração, são muito difíceis de serem definidas ou



programadas corretamente na máquina.

Através dos diferentes testes realizados, obtive-se taxas de acerto acima de 80%, que é um valor que coincide com os melhores resultados obtidos nos artigos que foram estudados. Embora mais testes necessitem ser realizados e com mais pessoas diferentes para comprovar os resultados, este é um bom primeiro passo.

Não foi possível encontrar um padrão com relação à melhor topologia para este tipo de classificação, sendo que entre os melhores resultados elas foram bem variadas. Foi observado que a transformada de Haar final é uma boa transformada para ser aplicada neste tipo de problema, e de maneira inesperada, o sinal no domínio do tempo obteve um ótimo resultado também. Mais testes necessitam ser feitos para melhor compreender esse resultado e validá-lo de maneira mais efetiva.

## Referências

- 1 CHAVARRIAGA, R.; SOBOLEWSKI, A.; MILLÁN, J. d. R. *Errare machinale est: The use of error-related potentials in brain-machine interfaces*. 2014.
- 2 LARSON, E. et al. *mne-tools/mne-python: v0.15*. 2017. Disponível em: <<https://doi.org/10.5281/zenodo.1043893>>.
- 3 ITURRATE, I.; MONTESANO, L.; MÍNGUEZ, J. Robot reinforcement learning using EEG-based reward signals. In: *Proceedings - IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2010. ISBN 9781424450381. ISSN 10504729.
- 4 FERREZ, P. W.; Del R. Millán, J. Error-related EEG potentials generated during simulated brain-computer interaction. *IEEE Transactions on Biomedical Engineering*, v. 55, n. 3, p. 923–929, 2008. ISSN 00189294.
- 5 LOTTE, F. et al. *A review of classification algorithms for EEG-based brain-computer interfaces*. 2007.
- 6 SPÜLER, C. N. M. *Error-related potentials during continuous feedback: using EEG to detect errors of different type and severity*. *Frontiers in Human Neuroscience*. <[https://www-ti.informatik.uni-tuebingen.de/~spueler/eeg\\_data/](https://www-ti.informatik.uni-tuebingen.de/~spueler/eeg_data/)>.
- 7 SPÄLER, M.; NIETHAMMER, C. Error-related potentials during continuous feedback: using EEG to detect errors of different type and severity. *Frontiers in Human Neuroscience*, v. 9, 2015. ISSN 1662-5161.