

Aplicação de técnicas de aprendizagem automática para classificação de emoções humanas com sinais de EEG

Nelson Filipe Capela , Petia Georgieva

Resumo – Este artigo apresenta um estudo referente a várias técnicas de Aprendizagem Automática (MachineLearning) e a sua aplicação para discriminar dois tipos de emoções humanas - valência emocional positiva e negativa, baseado em dados de eletroencefalografia (EEG). Numa primeira fase foi efetuada uma abordagem introdutória das técnicas mais típicas de Aprendizagem Automática, nomeadamente k-NearestNeighbor (K-NN), NaiveBayes, Support Vector Machine (SVM), Artificial Neural Networks (ANN), DecisionTree. Na segunda fase, as mesmas técnicas foram aplicadas para a classificação das emoções humanas com sinais de EEG.

Palavras-chave — aprendizagem automática, classificação, EEG

I. INTRODUÇÃO

A Aprendizagem Automática (MachineLearning) é uma das áreas da Inteligência Artificial cujo objectivo é o desenvolvimento de técnicas que permita aos computadores aprender com a experiência, ou seja, criar programas capazes de induzir conhecimentos a partir de informação não estruturada em forma de exemplos (dados). A Aprendizagem Automática (AA) e a Exploração de Dados (Data Mining) são áreas muito próximas no entanto enquanto a Exploração de Dados (ED) foca a extracção de conhecimento compreensível (padrões estruturais que capturam de uma maneira explícita o conhecimento), AA entra mais nos métodos computacionais que fazem possível que computadores aprendam e melhorem o seu desempenho com a experiência. ED trata de grandes volumes de informação (normalmente bases de dados do mundo real) enquanto AA tipicamente utiliza conjunto de dados mais pequenos para testar e avaliar o desempenho dos métodos e algoritmos computacionais propostos.

O objetivo deste artigo é aplicar algumas das técnicas mais poderosas de AA para explorar dados de eletroencefalografia (EEG) e discriminar dois tipos de emoções humanas - positivas e negativas.

Este artigo está organizado da seguinte maneira: Secção II apresenta as cinco principais técnicas de Aprendizagem Automática. Secção III descreve a natureza do sinal de EEG, o processo de aquisição dos dados e as métricas consideradas para a análise da performance das várias técnicas de AA. Seguidamente, a Secção IV apresenta os resultados da classificação e faz uma comparação das técnicas de AA. Por fim, Secção V apresenta a conclusão de todo este trabalho.

II. TÉCNICAS DE APRENDIZAGEM AUTOMÁTICA

Esta secção apresenta uma visão global das várias técnicas de classificação utilizadas ao longo deste trabalho.

A. K-NearestNeighbor

O classificador *K-NearestNeighbor* (*K-NN*) é um método de classificação de objectos, que tem por base *K* exemplos de treino mais próximos do objecto a classificar. Neste algoritmo um objecto é classificado por uma maioria de votos dos seus vizinhos, com o objecto a ser atribuído à classe mais comum entre os seus *K* vizinhos mais próximos. *K* é, por norma, um número inteiro positivo ímpar, de forma a evitar empates e, por consequente, indefinições na classificação do objecto. Para as situações em que *K*=1, o objecto a classificar é simplesmente atribuído à classe do seu vizinho mais próximo.

De forma a representar o processo de classificação, vamos admitir que *D* representa um conjunto de objetos já conhecidos e classificados, ou seja, os dados de treino, e que cada objeto é representado por (x_1, x_2, \dots, x_n) , os quais podem ser vistos como pontos espaciais de dimensão *n*. Caso se pretenda classificar um novo objeto, $Z = (z_1, z_2, \dots, z_n)$, torna-se necessário determinar a distância espacial entre este novo objeto (*Z*) e os restantes *k*- objetos pertencentes ao conjunto *D*. Para a classificação final do objeto *Z* é tido em consideração qual das classes existentes, referentes aos *k*- objetos com menor distância, surgem com mais frequência. Este método pode ser visto como um processo de votação entre *k* entidades, em que a decisão final é tomada mediante a maioria absoluta [3]. As medidas da distância entre os objetos depende do tipo dos atributos dos objetos (nominais ou numéricos). Caso se tratem de atributos nominais pode-se utilizar, por exemplo, o *ModifiedValueDifferenceMetric* (MVDM). Este permite determinar a distância entre dois atributos (V_1 e V_2) através da expressão (1).

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right| \quad (1)$$

Onde n_1, n_2 é o numero total de ocorrências de cada atributo, n_{1i}, n_{2i} é o numero de ocorrências em cada classe i .

Caso se tratem de atributos numéricos, normalmente é usada a distância euclidiana, expressão (2).

$$d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}, \quad (2)$$

onde p e q representam os objetos em questão, x e y as respectivas coordenadas, considerando dimensão 2.

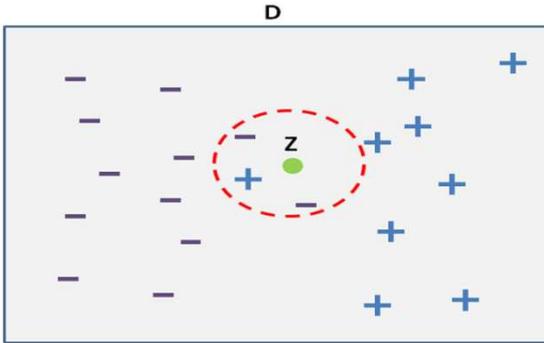


Figura 1: Exemplo da classificação via K-NN

Através da Figura 1 é apresentado um exemplo de classificação utilizando o método K-NN. Deste modo, uma vez que se está a admitir $k = 3$, ou seja, que vão ser tidos em consideração os 3 objetos mais próximos, o novo objeto Z vai ser classificado como “-”, uma vez que é este que se encontra em maioria.

B. NaiveBayes

O classificador NaiveBayes é uma técnica probabilística baseada no teorema de Bayes, expressão (3), para calcular a probabilidade a posterior da classe C_i , dado o objeto (o exemplo) \vec{x} ,

$$P(C_i|\vec{x}) = \frac{P(\vec{x}|C_i)P(C_i)}{P(\vec{x})} \quad (3)$$

onde $P(C_i)$ representa a probabilidade inicial (prior) de objetos da classe C_i , $P(\vec{x}|C_i)$ é a probabilidade condicionada do \vec{x} dada a classe C_i

$$P(\vec{x}|C_i) = \prod P(x_j|C_i) \quad (4)$$

$P(\vec{x})$ é a probabilidade total do objecto \vec{x}

$$P(\vec{x}) = \sum_{i=1}^M P(\vec{x}|C_i)P(C_i) \quad (5)$$

Após a determinação das probabilidades a posterior das todas as classes, a decisão da classificação é tomada segundo a expressão (6).

$$P(\vec{x}|C_i)P(C_i) > P(\vec{x}|C_j)P(C_j) \forall i \neq j \quad (6)$$

A probabilidade prior $P(C_i)$ é calculada como a percentagem dos exemplos da classe C_i no conjunto total de exemplos de treino.

Para exemplos com atributos nominais x_j a probabilidade condicionada $P(x_j|C_i)$ é obtida através da contagem de ocorrências do atributo x_j na classe C_i . Por outro lado, para exemplos com atributos numéricos a probabilidade condicionada é dada pela expressão (7).

$$P(x_j|C_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (7)$$

Onde σ representa a variância e μ a média dos valores do atributo x_j no conjunto de treino.

Em determinadas situações, pode acontecer que no conjunto de exemplos de treino, um determinado atributo x_j não seja observado numa classe C_i , fazendo com que $P(x_j|C_i) = 0$ e daí a probabilidade a posterior da mesma classe $P(C_i|\vec{x}) = 0$. Para evitar esta situação é utilizada a correção de Laplace. Tipicamente, para atributos nominais, $P(x_j|C_i)$ é determinada pela expressão (8), onde N_{ij} é o número de vezes que o atributo x_j surge na classe C_i e N_c é o número de atributos na classe C_i .

$$P(x_j|C_i) = \frac{N_{ij}}{N_c} \quad (8)$$

Aplicando a correção de Laplace, a probabilidade condicionada passa a ser dada pela expressão (9).

$$P(x_j|C_i) = \frac{N_{ij} + k}{N_c + k|x_j|} \quad (9)$$

Onde k é o fator de suavização (tipicamente 1) e $|x_j|$ o número de possíveis valores que o atributo x_j pode tomar [4].

C. Support Vector Machine

Support Vector Machine (SVM) é uma técnica de classificação que tem como objetivo definir um hiperplano que permita separar os dados de forma a identificar as classes. Tendo então um conjunto de dados, tal como representado pela Figura 2, esta técnica tenta determinar o plano que permite maximizar a margem entre as classes existentes, para que de seguida possa classificar novos objetos com uma maior precisão.

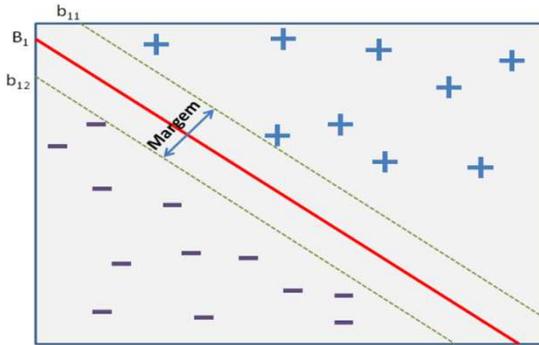


Figura 2: SVM – Exemplo de problema linear

As retas b_{11} , B_1 e b_{12} são matematicamente representadas pelas expressões (10), (11) e (12)

$$\vec{w} \cdot \vec{x} + b = +1 \tag{10}$$

$$\vec{w} \cdot \vec{x} + b = 0 \tag{11}$$

$$\vec{w} \cdot \vec{x} + b = -1 \tag{12}$$

A respectiva margem é determinada pela expressão (13).

$$Margem = \frac{2}{\|\vec{w}\|^2} \tag{13}$$

A técnica de classificação SVM tenta maximizar a expressão (13) tendo em conta as seguintes restrições [5].

$$f(\vec{x}_i) = \begin{cases} 1, & \text{se } \vec{w} \cdot \vec{x} + b \geq 1 \\ -1, & \text{se } \vec{w} \cdot \vec{x} + b \leq -1 \end{cases} \tag{14}$$

Para dados não linearmente separáveis (um exemplo representado na Figura 3) é introduzido o conceito de *kernel induced features space* e daí *kernel SVM*, uma técnica que permite transformar os dados num espaço de maior dimensão onde estes são já linearmente separáveis.

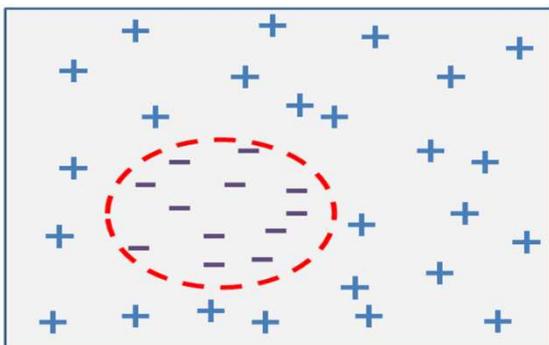


Figura 3: SVM – Exemplo de problema não linear

D. Redes Neurais Artificiais

A Rede Neuronal Artificial (*Artificial Neural Network-ANN*) é um modelo matemático inspirado da estrutura e as funcionalidades das Redes Neurais Biológicas (RNB). As Redes Neurais Artificiais (RNA) são constituídas por camadas de funções matemáticas, chamadas neurónios artificiais, que se aproximam em termos de função aos neurónios biológicos. Na Figura 4 é apresentada a arquitetura geral de uma RNA. Uma RNA é normalmente composta por três camadas, a camada de entrada, a camada intermédia e a camada de saída.

As RNA são uma aproximação grosseira quando comparadas com as RNB, em termos de complexidade. No entanto, as RNA são uma ferramenta muito utilizada nas áreas de classificação e reconhecimento de padrões. As suas propriedades de adaptação, filtragem do ruído, aprendizagem e processamento em paralelo fazem das RNA uma das ferramentas com mais aplicações (classificação e regressão) das técnicas de aprendizagem automática. A principal desvantagem dos RNA é o facto de necessitar de muito tempo de processamento para redes grandes [6].

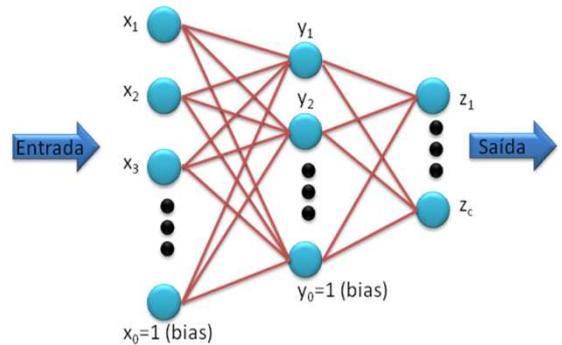


Figura 4:Arquitetura geral de RNA

A forma supervisionada para uma RNA “aprender” de classificar objetos é através do erro entre a resposta dada pela rede z_k (a classe que a RNA atribuiu) e a resposta esperada t_k (a class verdadeira de um determinado objeto do conjunto do treino)

$$E = \frac{1}{2} (z_k - t_k)^2 \tag{15}$$

Este erro é utilizado para ajustar os parâmetros (os pesos de ligação entre as camadas e os neurónios) de RNA. Entre vários algoritmos de minimização do erro, o algoritmo de *backpropagation* é um dos mais usados.

E. Árvores de decisão (AD)

Árvore de Decisão (*DecisionTree*) é técnica de classificação baseada na divisão de um problema complexo em vários subproblemas, repetindo este processo de forma recursiva através da geração de uma árvore (Figura 5).

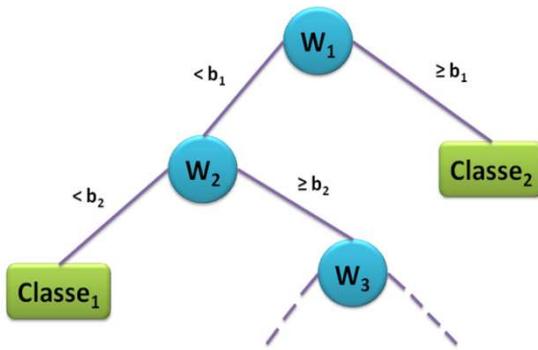


Figura 5: Árvores de Decisão (AD)– Exemplo de construção

Na Figura 5, é apresentado um esquema típico da AD. As variáveis W_i são os nós de decisão. A cada um destes nós está associado um atributo. As variáveis b_i representam os limites (threshold) dos atributos que dividem a decisão em dois caminhos da árvore. Estes podem ser nominais ou numéricos. As variáveis $classe_i$, permitem efetuar a classificação do objeto em análise.

Uma das principais fases para a construção da AD passa pela determinação da melhor forma para efetuar a divisão dos vários ramos e pela escolha do atributo mais indicado para se utilizar como raiz, ou seja, o atributo pelo qual se inicia o processo da criação da AD (W_1 , para o caso da Figura 5). As métricas mais típicas para otimização de uma AD são *GiniIndex* (expressão (16)), *Entropia* (expressão (17)) e *ERRO* (expressão (18)):

$$GINI(t) = 1 - \sum_j [p(j|t)]^2 \quad (16)$$

$$Entropia(t) = - \sum_j p(j|t) \log p(j|t) \quad (17)$$

$$ERRO(t) = 1 - \max_j P(j|t), \quad (18)$$

onde $p(j|t)$ representa a probabilidade da classe j no nó t .

Como critério de paragem do processo de criação de novos ramos, normalmente é considerada a situação em que todos os exemplos pertencem à mesma classe ou é definido um limite máximo para o número de ramos [7].

III. CASO DE ESTUDO- CLASSIFICAÇÃO DE SINAIS DE EEG

Nesta sessão é dedicada ao sinal de EEG, a forma de aquisição e pré-processamento dos dados, e seguidamente são apresentadas as medidas de performance dos classificadores.

Para melhor se compreender a classificação dos sinais EEG, é necessário conhecer as características do sinal a estudar. A actividade eléctrica do cérebro é medida através de eléctrodos (pequenas placas metálicas), que estão colocadas sobre a pele do crânio dos indivíduos, em posições específicas. Um dos sistemas mais usados de colocação de eléctrodos no crânio, é o sistema internacional 10-20. O sistema é baseado numa relação entre a posição dos eléctrodos no crânio e na área do

cérebro sobre o qual estão localizados. Os lobos cerebrais estão caracterizados com letras, como por exemplo: o “F” significa Frontal; o “P”, Parietal; o “T”, Temporal; e “O” Occipital. Os números pares correspondem à parte direita do hemisfério e os ímpares à esquerda, a letra “Z” é referente à linha central entre os dois hemisférios (esquerdo e direito). Quanto menor for o número, menor a distancia em relação à linha central. Os números ou as letras seguintes representam, juntamente com a primeira letra a posição específica no hemisfério cortical. O sinal EEG é proveniente da diferença de potencial eléctrico entre os diferentes eléctrodos, amplificado e filtrado, antes de ser registado digitalmente. Estas ondas cerebrais geralmente têm uma amplitude entre 0 e 200mV e são classificadas segundo a sua frequência (varia entre 0.5 e 50Hz), localização, amplitude e simetria em relação à linha média que divide os dois hemisférios (esquerdo e direito).

A. Processo de aquisição de EEG sinais

Foram adquiridos sinais de EEG de 26 voluntários durante a visualização de um conjunto de 24 imagens com um elevado nível de excitação (>6) e com níveis de valência positiva (7.29 ± 0.65) e negativa (1.47 ± 0.24), Figura 6. As imagens foram seleccionadas do *InternationalAffectivePictureSystem*.

Cada uma das imagens foi apresentada 3 vezes de uma forma pseudoaleatória com uma duração de 3500 ms por cada experiencia. Durante os primeiros 750 ms é apresentado uma cruz fixa (que serve para concentração do sujeito), seguidamente, durante 500 ms, é apresentada uma das imagens e por fim, é apresentado um ecrã preto durante 2250 ms (que serve para inicialização), tal como se pode observar na Figura 7.



Figura 6: Aquisição de sinais EEG

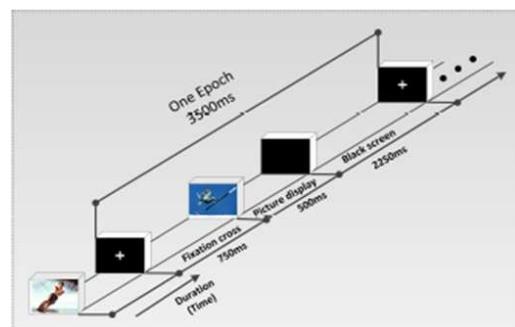


Figura 7: Fases contidas numa época de aquisição de dados

Amostras (de 1000Hz) de 21 canais de EEG posicionados de acordo com o sistema 10-20 (Figura) e 2 canais de Electro- Óculo-Grama(EOG) foram guardadas (Figura 8) e pré-processados (filtrados, correção da movimentação dos olhos, compensação inicial, divisão em épocas). Para cada um dos sinais obtidos foram determinadas as suas médias e utilizado um filtro *Zero-Phase*(Figura 8).

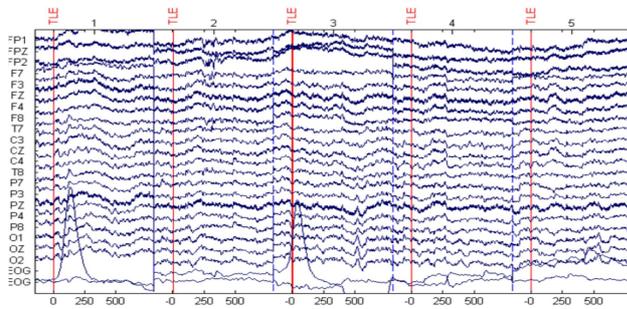
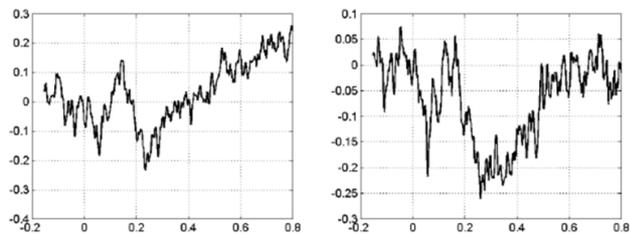


Figura 8: Canais EEG e EOG considerados no estudo



a) Estimulo (imagem) positivo b) Estimulo (imagem) negativo

Figura 9: Média dos sinais do canal FP1

Seguidamente, foram determinados os máximos e os mínimos das médias obtidas, e tendo como base o primeiro mínimo, foram definidos os atributos amplitude e latência com base nos mínimos e máximos consecutivos. Para isto, foram implementados três esquemas com a utilização de três filtros diferentes. Estes permitem apresentar à sua saída n máximos e n mínimos. Sempre que este padrão não é encontrado, o vetor do atributo em questão é preenchido com zeros. Foram utilizados os seguintes filtros: i) filtro *Butterworth* de quarta ordem com passa-banda em [0.5 - 15] Hz (Figura 10). Este permite obter um conjunto de 12 atributos correspondentes à latência e à amplitude dos primeiros três máximos e três mínimos após o instante 0s; ii) filtro *Butterworth* de quarta ordem com passa-banda em [0.5 - 4] Hz (Figura 11), o qual corresponde à banda de atividade cerebral Delta. Este permite obter oito atributos correspondentes à latência e à amplitude dos primeiros dois máximos e dois mínimos; iii) filtro *Butterworth* de quarta ordem com passa-banda em [4 - 8] Hz (Figura 12), o qual corresponde à banda de atividade cerebral Teta. Este permite obter um conjunto de 12 atributos correspondentes à latência e à amplitude dos primeiros três máximos e três mínimos após o instante 0s. Através

destes filtros, são obtidos três bases de dados distintas: FiltroAB, [0.5 - 15] Hz, FiltroAB Delta, [0.5 - 4] Hz e FiltroAB Teta [4 - 8] Hz.

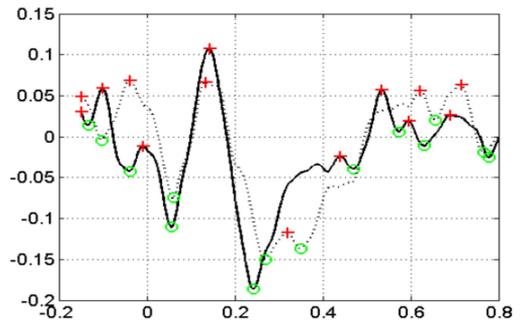


Figura 10 Filtro AB – Média e atributos de um canal de EEG. Estimulo positivo (bold line) , estimulo negativo (dot)

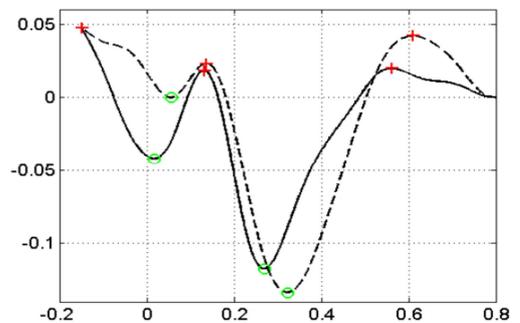


Figura 11 Filtro AB Delta – Média e atributos de um canal de EEG. Estimulo positivo (bold line) , estimulo negativo (dot)

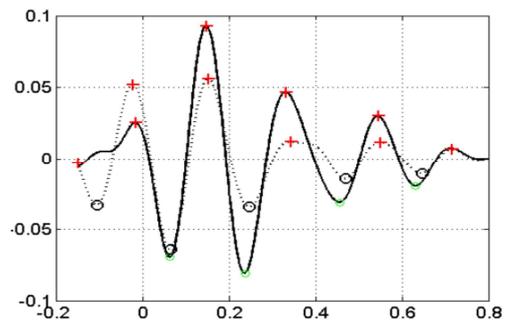


Figura 12 Filtro AB Teta – Média e atributos de um canal de EEG. Estimulo positivo (bold line) , estimulo negativo (dot)

B. Medidas de performance dos classificadores

A análise de performance das várias técnicas de classificação foi efetuada através da matriz de confusão (*confusionmatrix*). A estrutura base de uma matriz de confusão é apresentada na Tabela 1.

	Verdade Positiva	Verdade Negativa
Previsão Positiva	VP	FP
Previsão Negativa	FN	VN

Tabela 1: Matriz de Confusão (*confusionmatrix*)

Em que VP representa o número de previsões positivas que estão corretas, FP representa o número de previsões positivas que estão incorretas, FN é o número de previsões negativas que estão incorretas e VN é o número de previsões negativas que estão corretas. A partir da matriz de confusão, foram determinadas as seguintes medidas de performance:

$$accuracy = \frac{VP + VN}{VP + FP + VN + FN} \quad (19)$$

accuracy permite avaliar a percentagem de todas as previsões (positivas e negativas) que estão corretas.

$$precision = \frac{VP}{VP + FP} \quad (20)$$

precision determina a percentagem de previsões positivas que estão corretas.

$$specificity = \frac{VN}{VN + FP} \quad (21)$$

specificity determina a percentagem de ocorrências negativas que foram previstas corretamente.

$$recall = \frac{VP}{VP + FN} \quad (22)$$

recall fornece a percentagem de ocorrências positivas que foram previstas corretamente [8].

Enquanto *accuracy* é a medida de performance mais típica para dados balanceados, ou seja, dados com o mesmo número de casos de todas as classes, as restantes medidas (*precision*, *specificity*, *recall*) são mais adequadas para dados não balanceados. Embora os dados considerados neste estudo sejam balanceados (as amostras com valência positiva e valência negativa são

iguais), para obter uma análise completa foram calculadas as quatro medidas.

IV. PROCESSO DE CLASSIFICAÇÃO EM RAPIDMINER

Nesta secção as técnicas de aprendizagem automática introduzidas na secção II são usadas para a classificação de emoções humanas com sinais de EEG. A implementação dos modelos foi através da ferramenta RapidMiner [8].

A. Tratamento dos dados de EEG em RapidMiner

O primeiro passo de tratamento dos dados é a sua importação para RapidMiner (RM). Entre varias opções que o RM fornece, foi usada a opção *Import CSV File* como a mais adequada para ficheiros de texto com dados separados por caracteres. Durante a fase de importação, é necessário definir quais são as colunas de atributos normais de classificação (*regular*), a(s) coluna(s) das classes (*label*) e o tipo dos valores (reais, inteiros, booleanos, etc.). Após este processo é possível observar meta informação relativamente os dados importados, como por exemplo, os valores médios e a variância de cada um dos atributos, tal como se pode observar na Figura 13. No menu *Results-Data view* é possível visualizar varias formas de distribuição dos dados em função dos atributos e cores diferentes correspondentes as classes (Figura 14).

Role	Name	Type	Statistics	Range
label	class	integer	avg = 0.500 +/- 0.500	[0.000 ; 1.000]
regular	amp1	real	avg = -0.104 +/- 0.057	[-0.342 ; 0.012]
regular	amp2	real	avg = 0.109 +/- 0.060	[-0.014 ; 0.370]
regular	amp3	real	avg = -0.082 +/- 0.049	[-0.315 ; 0.016]
regular	amp4	real	avg = 0.050 +/- 0.036	[-0.027 ; 0.254]
regular	amp5	real	avg = -0.036 +/- 0.026	[-0.192 ; 0.014]
regular	amp6	real	avg = 0.031 +/- 0.020	[-0.025 ; 0.137]
regular	latency1	real	avg = 0.084 +/- 0.026	[0.001 ; 0.225]
regular	latency2	real	avg = 0.170 +/- 0.031	[0.047 ; 0.327]
regular	latency3	real	avg = 0.255 +/- 0.039	[0.112 ; 0.406]
regular	latency4	real	avg = 0.346 +/- 0.054	[0.182 ; 0.514]
regular	latency5	real	avg = 0.443 +/- 0.072	[0.241 ; 0.627]
regular	latency6	real	avg = 0.514 +/- 0.120	[0.000 ; 0.691]

Figura 13: Exemplo de apresentação dos dados após a sua importação em RapidMiner

Para tarefas de classificação a coluna de classe (o atributo *label*) deve conter valores do tipo nominal. Para isto, aplica se o operador de transformação de dados *Numerical_toPolynomial* (Figura 15). Por exemplo se as classes são inicialmente introduzidas como valores do tipo *integer* (1, 2, 3, etc.), devem ser transformadas em valores nominais (primeira, segunda, terceira, etc.). Por fim os dados importados e transformados conforme a tarefa pretendida são armazenados na base de dados de RM através do operador *Store*. O resultado da conversão e o formato final dos dados quadrados em RM é visualizado na Figura 16.

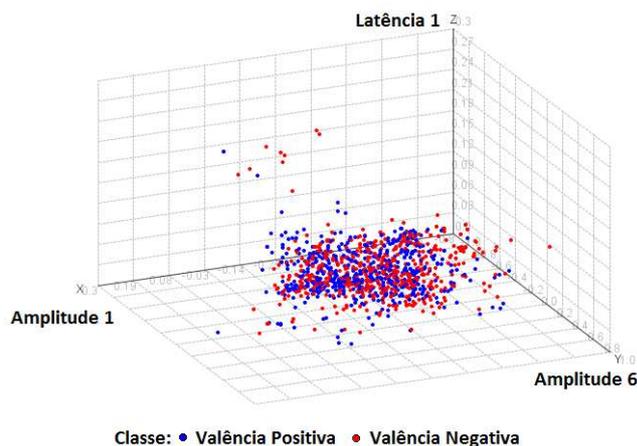


Figura 14: Exemplo de distribuição dos dados em função de 3 atributos (ampl. 1, ampl. 6, lat. 1)

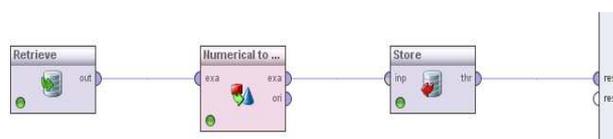


Figura 15: Conversão do atributo *class* de numérico para polinomial

Role	Name	Type	Statistics	Range
label	class	nominal	mode = 1 (546), least = 1 (546)	1 (546), 0 (546)
regular	amp1	real	avg = -0.104 +/- 0.057	[-0.342; 0.012]
regular	amp2	real	avg = 0.109 +/- 0.060	[-0.014; 0.370]
regular	amp3	real	avg = -0.082 +/- 0.049	[-0.315; 0.016]
regular	amp4	real	avg = 0.050 +/- 0.036	[-0.027; 0.254]
regular	amp5	real	avg = -0.036 +/- 0.026	[-0.192; 0.014]
regular	amp6	real	avg = 0.031 +/- 0.020	[-0.025; 0.137]
regular	latency1	real	avg = 0.084 +/- 0.026	[0.001; 0.225]
regular	latency2	real	avg = 0.170 +/- 0.031	[0.047; 0.327]
regular	latency3	real	avg = 0.255 +/- 0.039	[0.112; 0.406]
regular	latency4	real	avg = 0.346 +/- 0.054	[0.182; 0.514]
regular	latency5	real	avg = 0.443 +/- 0.072	[0.241; 0.627]
regular	latency6	real	avg = 0.514 +/- 0.120	[0.000; 0.691]

Figura 16: Exemplo de formato final dos dados guardados em RapidMiner

B. Classificação

Figura 17 ilustra o modelo tradicional de classificação. Neste caso concreto como técnica de aprendizagem automática é implementada uma rede neuronal artificial (RNA) através do operador *Neural_Net*. O operador *X-Validation* divide o processo de classificação e os dados em duas partes. Primeiro é executada a fase de treino da RNA (janela *training*) e depois a fase de teste (janela *testing*) onde a RNA treinada é aplicada (operador *ApplyModel*) aos dados não utilizados para o treino. A performance do classificador em termos de matriz de confusão e as quatro medidas (*accuracy*, *precision*, *specificity*, *recall*) são analisadas através do operador *Performance*. O operador *Optimize_Parameters* é opcional e serve para otimizar parâmetros do classificador previamente seleccionados. Por exemplo para o caso do classificador RNA podem se escolher os principais parâmetros do algoritmo de ajuste dos pesos (*learning rate* e *momentum*). Neste trabalho, alterando o bloco do classificador, foram otimizados e testados cinco classificadores: *K-NN*, *NaiveBayes*, *SVM*, *Neural Networks*, *DecisionTree*. Na Tabela 2 estão apresentados

os parâmetros dos classificadores após optimização com o operador *Optimize_Parameters*.

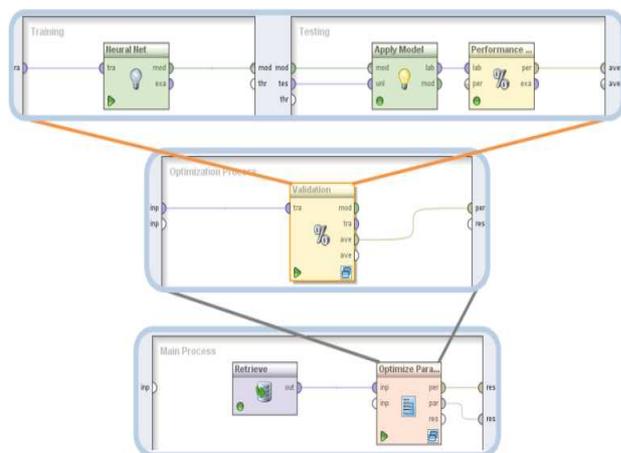


Figura 17: Processo de classificação

Parametros	K-NN		Naive Bayes		SVM	
	k	Weighted Vote	Laplace Correction	Kernel Type	C	Epsilon
FiltroAB	1	false	true	dot	NaN	NaN
FiltroAB Delta	1	true	false	dot	NaN	NaN
Filtro AB Teta	1	true	false	dot	NaN	NaN
Parametros	Neural Networks			Decision Tree		
	Momentum	Learning Rate	Minimal Size for Split	Minimal Leaf Size	Maximal Depth	Confidence
FiltroAB	0,2	0,2	4	1	23	0,3
FiltroAB Delta	0,143	0,143	5	1	23	0,3
Filtro AB Teta	0,2	0,1	5	1	23	0,3

Tabela 2: Parâmetros optimizados através do operador *Optimize_Parameters*

Os resultados da análise da performance dos cinco classificadores estão resumidos na Figura 18 (para base de dados Filtro AB), na Figura 19 (para base de dados Filtro AB Delta), na Figura 20 (para base de dados Filtro AB Teta). Tal como se pode observar, a técnica de classificação com melhor performance, independentemente da base de dados considerada, é a *K-NN*. Com uma boa performance temos também as técnicas *Neural Networks* e *DecisionTree*, sendo que estas apresentam valores de performance muito próximos. No entanto, estas apresentaram tempo de processamento muito superior ao apresentado pela técnica *K-NN*.

A classificação das emoções humanas com sinais de EEG é conseguida com maior sucesso usando a base de dados FiltroAB que correspondem a maior gama de frequência [0.5 - 15] Hz. Esta constatação pode-se justificar através das Figuras 10, 11 e 12, onde podemos observar que para o FiltroAB existe uma maior discrepância entre os atributos das duas classes (as amplitudes e as latências da Valência Positiva e Negativa)

, ou seja, tornasse mais fácil de identificar as diferentes classes neste base de dados.

técnicas que levam a melhor classificação das emoções humanas com sinais de EEG são *DecisionTree* e *Neural Networks* aplicados a base de dados FiltroABTeta.

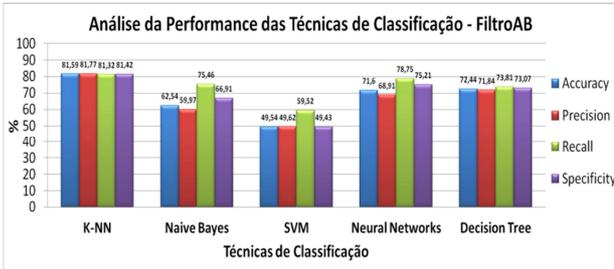


Figura 18: Filtro AB – Performance dos 5 classificadores

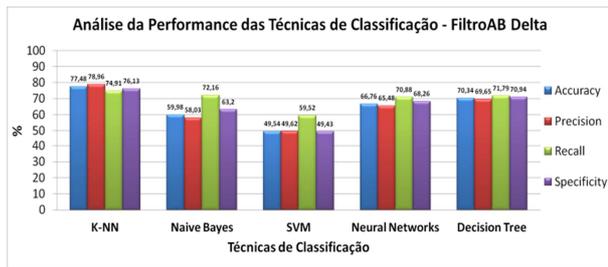


Figura 19: Filtro AB Delta – Performance dos 5 classificadores

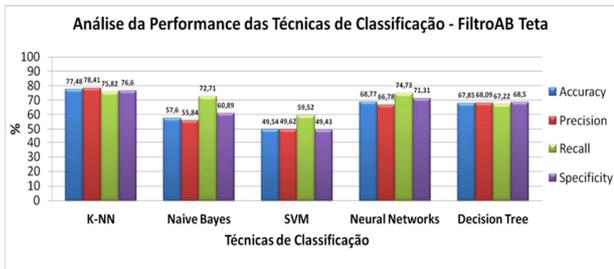


Figura 20: Filtro AB Teta – Performance dos 5 classificadores

Os resultados acima apresentados dizem respeito aos sinais adquiridos de 21 canais de EEG. Como o tempo de processamento desta base de dados é elevado, resolveu se efectuar uma análise idêntica à anterior, mas considerando apenas alguns canais de EEG. Como os estímulos da actividade cerebral são imagens, decidiu-se escolher como canais mais apropriados os canais correspondentes ao córtex visual, nomeadamente o canal P3 e o canal P4. Os parâmetros optimizados dos cinco classificadores relativamente a estas bases de dados são apresentados na Tabela 3. Os respectivos resultados da análise da performance dos cinco classificadores são visualizados na Figura 21 (base de dados Filtro AB), Figura 22 (base de dados Filtro AB Delta), Figura 23 (base de dados Filtro AB Teta). Neste caso, de base de dados reduzida, as

Parametros	K-NN		Naive Bayes		SVM	
	k	Weighted Vote	Laplace Correction	Kernel Type	C	Epsilon
FiltroAB	1	false	false	dot	NaN	NaN
FiltroAB Delta	1	true	true	dot	NaN	NaN
Filtro AB Teta	1	true	true	dot	NaN	NaN

Parametros	Neural Networks			Decision Tree		
	Momentum	Learning Rate	Minimal Size for Split	Minimal Leaf Size	Maximal Depth	Confidence
FiltroAB	0,1	0,9	30	1	80	0,2
FiltroAB Delta	0,4	0,7	1	1	40	0,25
Filtro AB Teta	0,5	0,9	1	1	35	0,2

Tabela 3: Parâmetros optimizados – canal P3 e P4

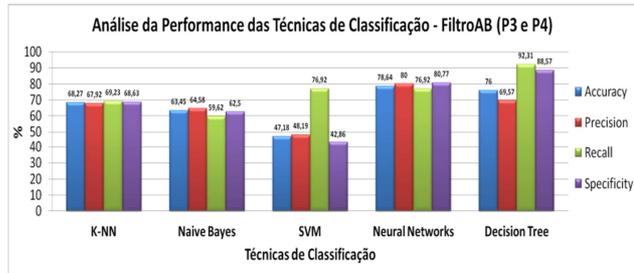


Figura 21: Filtro AB – Performance dos 5 classificadores (canal P3 e P4)

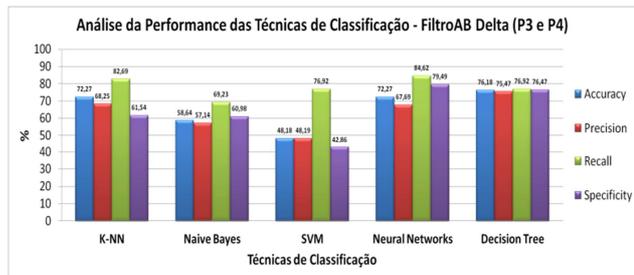


Figura 22: Filtro AB Delta – Performance dos 5 classificadores (canal P3 e P4)

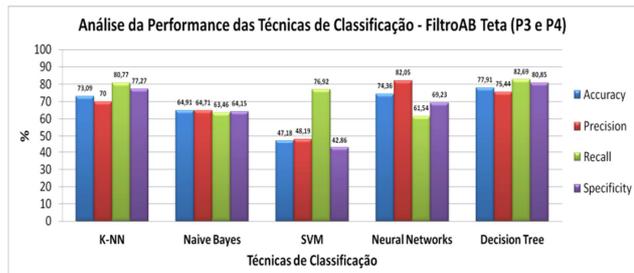


Figura 23: Filtro AB Teta – Performance dos 5 classificadores (canal P3 e P4)

V. CONCLUSÃO

Através deste artigo é apresentado um estudo referente a várias técnicas de aprendizagem automática (AA). Numa primeira fase foi efetuada uma abordagem introdutória das técnicas mais típicas de AA, nomeadamente *K-NN*, *Naive Bayes*, *SVM*, *Neural Networks*, *Decision Tree*. Na segunda fase, as mesmas técnicas foram aplicadas para a classificação das emoções humanas com sinais de EEG. O objetivo deste estudo é permitir analisar a performance e a robustez de cada uma das técnicas através da utilização de diferentes bases de dados de EEG. Os modelos de classificação foram implementados no ambiente programático RapidMiner. As técnicas de classificação que apresentam uma melhor performance são *K-NN*, *Neural Network* e *Decision Tree*. No entanto, o *K-NN* destaca-se perante as restantes em termos de tempo de procedimento e complexidade da solução. No que diz respeito à robustez relativamente varias bases de dados, as técnicas *Neural Network* e *Decision Tree* apresentam melhores resultados.

REFERENCIAS

- [1] L. M. Hirshfield et al., "Combining Electroencephalograph and Functional Near Infrared Spectroscopy to Explore Users' Mental Workload", *Neuroergonomics and Operational Neuroscience*, 5th International Conference, San Diego, CA, USA, 19-24 de Julho, 2009.
- [2] A. Savran et al., "Emotion Detection in the Loop from Brain Signals and Facial Images", *eINTERFACE'06*, 17 de Julho – 11 de Agosto, Dubrovnik, Croatia. Kantardzic, *Data Mining: Concepts, Models, Methods and Algorithms*. John Wiley & Sons
- [3] J. Gama, "Aprendizagem Bayesiana - Introdução". Site: http://www.liaad.up.pt/~jgama/Aulas_ECD/bayes.pdf, consultado em 4 Junho 2012.
- [4] N. Cristianini, John Shawe-Taylor, "An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods", Cambridge University Press, 23/03/2000.
- [5] P. Georgieva "ML Lecture 2: Artificial Neural Networks (ANN)". Slides da aula Unidade Curricular 3. 2012.
- [6] P. Georgieva, "Lecture 3: Alternative Classification Techniques". Slides da aula Unidade Curricular 3. 2012.
- [7] Wil M. P. van der Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes". Springer
- [8] RapidMiner 5[®]. Site: <http://rapid-i.com>