UNIVERSIDADE PRESBITERIANA MACKENZIE

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Classificação de Sinais de Eletroencefalograma usando Máquinas de Vetores Suporte

Sandro Luiz das Chagas

Orientador: Prof. Dr. Clodoaldo Aparecido Moraes Lima

São Paulo
2009

Livros Grátis

http://www.livrosgratis.com.br

Milhares de livros grátis para download.

UNIVERSIDADE PRESBITERIANA MACKENZIE

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Classificação de Sinais de Eletroencefalograma usando Máquinas de Vetores Suporte

Sandro Luiz das Chagas

Orientador: Prof. Dr. Clodoaldo Aparecido Moraes Lima

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Presbiteriana Mackenzie, como parte das exigências para obtenção do Titulo de Mestre em Engenharia Elétrica.

São Paulo
2009

C433c Chagas, Sandro Luiz das.
Classificação de sinais de eletroencefalograma usando máquinas de vetores suporte. / Sandro Luiz das Chagas. - 2009.
98 f. : il. ; 30 cm.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Presbiteriana Mackenzie, São Paulo, 2009.
Bibliografia: f. 81-85.

1. Máquina de vetor de suporte (SVM). 2. Eletroencefalograma (EEG). 3. Séries temporais. 4. Extração de características. 5. *Wavelet*. I. Título.

CDD 621.3

UNIVERSIDADE PRESBITERIANA MACKENZIE

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

SANDRO LUIZ DAS CHAGAS

Classificação de Sinais de Eletroencefalograma usando Máquinas de Vetores Suporte

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Presbiteriana Mackenzie, como parte das exigências para obtenção do Titulo de Mestre em Engenharia Elétrica.

Aprovado em

BANCA EXAMINADORA

Prof. Dr. Clodoaldo Aparecido de Moraes Lima
Universidade Presbiteriana Mackenzie

Prof. Dr. Márcio Eisencraft
Universidade Presbiteriana Mackenzie

Profa. Dra. Ana Carolina Lorena
Universidade Federal do ABC

São Paulo
2009

Dedicatória

Aos **meus pais** e **irmãos**, pelo apoio e
pelo carinho sempre demonstrados;
à **minha família**, fonte constante de
incentivo; em especial ao **meu filho Matheus**,
pela paciência e pela compreensão.

Agradecimentos

A **Deus**, por me dar força e perseverança nesta jornada.

Ao **Professor Doutor Clodoaldo Aparecido de Moraes Lima**, pela paciência, cordialidade e incentivo; por acreditar na minha capacidade e propor desafios. Muito obrigado pelo tempo dedicado a mim e por todos os seus ensinamentos que me mostraram que para ser mestre é preciso sempre aprender.

Aos **colegas da Faculdade de Engenharia Elétrica**, juntos superamos momentos difíceis nesta caminhada.

Ao **corpo docente da Faculdade de Engenharia Elé**trica, que atendeu prontamente às diversas dúvidas e compartilhou conhecimentos valiosos.

Ao **MackPesquisa** pelo auxílio oferecido.

À **Coordenação de Aperfeiçoamento de Pessoal de Nível Superior** (CAPES), pelo suporte financeiro, e de forma geral por apoiar a pesquisa no Brasil.

A **todos os amigos** que me apoiaram na realização deste trabalho.

RESUMO

O eletroencefalograma (EEG) é um exame médico largamente utilizado no estudo da função cerebral e de distúrbios neurológicos. O EEG é uma série temporal que contém os registros de atividade elétrica do cérebro. Um grande volume de dados é gerado pelos sistemas de monitoração de EEG, o que faz com que a análise visual completa destes dados se torne inviável na prática. Com isso, surge uma grande demanda por métodos computacionais capazes de extrair, de forma automática, informação útil para a realização de diagnósticos. Para atender essa demanda, é necessária uma forma de extrair de um sinal de EEG as características relevantes para um diagnóstico e também uma forma de classificar o EEG em função destas características. O cálculo de estatísticas sobre coeficientes wavelet vem sendo empregado com sucesso na extração de características de diversos tipos de séries temporais, inclusive EEG. As máquinas de vetores de suporte (SVM – do inglês *Support Vector Machines*) constituem uma técnica de aprendizado de máquina que possui alta capacidade de generalização e têm sido empregadas com sucesso em problemas de classificação por diversos pesquisadores. Nessa dissertação é feita uma análise do impacto da utilização de vetores de características baseados em coeficientes wavelet na classificação de EEG utilizando diferentes implementações de SVM.

Palavras-chave: *máquina de vetor de suporte (SVM), eletroencefalograma (EEG), séries temporais, extração de características, wavelet.*

ABSTRACT

Electroencephalogram (EEG) is a clinical method widely used to study brain function and neurological disorders. The EEG is a temporal data series which records the electrical activity of the brain. The EEG monitoring systems create a huge amount of data; with this fact a visual analysis of the EEG is not feasible. Because of this, there is a strong demand for computational methods able to analyze automatically the EEG records and extract useful information to support the diagnostics. Herewith, it is necessary to design a tool to extract the relevant features within the EEG record and to classify the EEG based on these features. Calculation of statistics over wavelet coefficients are being used successfully to extract features from many kinds of temporal data series, including EEG signals. Support Vector Machines (SVM) are machine learning techniques with high generalization ability, and they have been successfully used in classification problems by several researches. This dissertation makes an analysis of the influence of feature vectors based on wavelet coefficients in the classification of EEG signal using different implementations of SVMs.

Keywords: *support vector machine (SVM), eletroencefalogram (EEG), temporal data series, feature extraction, wavelet.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SÍMBOLOS

$\mathbf{x}(t)$	Sinal de entrada.
$X(f)$	Transformada continua de Fourier do sinal x.
$X[f]$	Transformada discreta de Fourier do sinal x.
N	Número total de pontos do sinal de entrada.
r_{xy}	Função de correlação dos sinais **x** e **y**.
r_{xx}	Função de autocorrelação dos sinal **x**
$W(a,b)$	Transformada wavelet continua.
ψ	Função wavelet mãe (*mother wavelet*)
$d(k,l)$	Amostragem de W(a, b) em um ponto discreto *k* e *l*.
$\phi(\cdot)$	Função de escala da transformada wavelet discreta.
$H_0(\omega)$	Filtro de análise passa-baixa.
$H_1(\omega)$	Filtro de análise passa-alta.
$G_0(\omega)$	Filtro de síntese passa-baixa.
$G_1(\omega)$	Filtro de síntese passa-alta.
$\downarrow$	Diminuição da amostragem (*downsampling*).
$\uparrow$	Aumento da amostragem (*upsamplig*).
A	Coeficientes wavelet de aproximação.
D	Coeficientes wavelet de detalhe.
W	Vetor de pesos do classificador.
X	Vetor com os dados de entrada a serem classificados.
Y	Vetor de saída contém o resultado das classificações.
$(\mathbf{w} \bullet \mathbf{x})$	Denota um produto interno entre os vetores **x** e **w**.
B	Termo de polarização ou bias.
$\rho(\mathbf{w}, b)$	Margem de separação dos hiperplanos.
$\|\mathbf{w}\|$	Denota norma do vetor **w**.
$\Phi(\mathbf{w})$	Hiperplano de separação ótimo.
C	Parâmetro que controla a complexidade do modelo e do erro de treinamento.

NC	Número de classes no problema.
$K(x_i, x_j)$	Função de kernel.
ξ_i	Função associada ao erro.
α_i	Multiplicadores de Lagrange i = 1, ..., N.
$(\cdot)_+$	Significa que apenas os valores positivos são considerados.
p(z,k)	Função de aproximação/amortização.
$p(\mathbf{y} \mid \mathbf{w}, \sigma_\varepsilon^2)$	Função de verossimilhança dos dados observados.
$p(y' \mid \mathbf{y})$	Função de predição do y'.
f(x;w)	Saída do classificador.

SUMÁRIO

CAPÍTULO 1 –INTRODUÇÃO

1.1 POSICIONAMENTO E MOTIVAÇÃO DA PESQUISA

Uma série temporal é um conjunto de observações $x(t)$, onde cada uma destas ocorre em um tempo específico t (BROCKWELL; DAVIS, 2002). Uma série temporal é dita ser discreta quando o conjunto de tempos $T_{0,}$ no qual ocorrem as medições, é um conjunto discreto. Séries temporais contínuas são obtidas quando as observações são gravadas continuamente em um intervalo de tempo (BROCKWELL; DAVIS, 2002).

Dentre as áreas de estudo sobre séries temporais, encontram-se: análise e modelagem da série, que visa descrever a série, verificar suas características e possivelmente relacioná-la com outra série; a predição de valores futuros da série temporal, que consiste basicamente em tentar prever o futuro com base em acontecimentos passados. Análise e predição de séries temporais constituem uma área de pesquisa multidisciplinar, sendo de extrema relevância para outras áreas do conhecimento, tais como Geofísica, Astrofísica e Meteorologia (ciências da natureza), Demografia (ciências sociais), Neurofisiologia (ciências médicas), Finanças (ciências econômicas) e Engenharias (ciências tecnológicas).

Dentre os diversos tipos de séries temporais existentes, o estudo de técnicas computacionais para análise de EEG vem ganhando bastante espaço. A classificação destes sinais é de grande interesse para a Medicina e também para a área de Inteligência Artificial. Sistemas de suporte ao diagnóstico de EEG e outros sistemas de suporte à decisão vêm sendo amplamente pesquisados (SUBASI, 2007; ÜBEYLI, 2008; REVETT *et al.*, 2006).

Sistemas de suporte ao diagnóstico têm se tornado uma ferramenta essencial na tecnologia médica, cujo um dos principais componentes é uma máquina com capacidade de aprender certas características das doenças importantes para a obtenção do diagnóstico. Estes sistemas podem ser utilizados para diagnosticar futuros pacientes com estado de doença incerto.

O sinal de EEG é uma série temporal bastante variável (LEHNERTZ, 1999), e dependente do estado em que o paciente se encontra, como por exemplo: se o paciente está com os olhos abertos ou fechados, se o paciente está dormindo ou acordado, etc. Apesar dos rápidos avanços nos exames com neuro-imagem, o EEG continua tendo papel importante na

análise de anomalias no cérebro, tais como: epilepsia (SUBASI, 2007) e esquizofrenia (SABETI *et al.*, 2007). Com o objetivo de extrair informações a partir do registro das atividades elétricas do cérebro, várias técnicas computacionais foram desenvolvidas (LEHNERTZ, 1999).

Desde os estudos iniciais de processamentos de sinais de EEG, as representações mais comumente utilizadas são baseadas na transformada de Fourier (REVETT *et al.*, 2006). A transformada de Fourier é útil na caracterização do sinal de EEG, entretanto sua implementação clássica, a transformada rápida de Fourier (*Fast Fourier Transform* – FFT), apresenta grande sensibilidade a ruídos (REVETT *et al.*, 2006). A principal desvantagem da transformada de Fourier é a falta de informação da evolução temporal da série (SEMMLOW, 2004). A transformada de Fourier de curto tempo é uma das primeiras tentativas de suprir a falta de informação temporal. Basicamente consiste em aplicar a transformada de Fourier tradicional em vários segmentos, isto é, janelas da série a ser analisada. O problema desta técnica é definir o tamanho e o número de segmentos a serem analisados. A transformada wavelet foi introduzida em meados dos anos 80 como um método mais apropriado para a realização da análise no tempo e na freqüência (REVETT *et al.*, 2006). Desde então está técnica vem ganhando espaço em diversas aplicações.

A extração eficiente das características relevantes para detecção de uma doença, a partir dos resultados de exames clínicos como o EEG, é um componente importante dos sistemas de suporte ao diagnóstico. Baseado nessas características que o sistema fará seu diagnóstico. Na extração de características, vários métodos diferentes podem ser utilizados, assim como diversas características podem ser extraídas a partir do mesmo dado bruto. Para a grande maioria dos dados, as características extraídas por algum método podem, independentemente, representar os dados originais, mas nenhuma delas possui informações suficientes para aplicações práticas. Além disso, parece não haver uma forma simples de medir a relevância das características para uma tarefa de classificação. Para este tipo de classificação, diversos tipos de características, freqüentemente, necessitam ser usadas concomitantemente a fim de encontrar um desempenho eficaz.

O módulo de classificação é o estágio final no diagnóstico ou detecção automática. Este examina o vetor de características de entrada e baseado nas características apreendidas na fase de treinamento produz uma hipótese sugestiva. Diferentes técnicas de aprendizado de máquina podem ser aplicadas à classificação de sinais de EEG dentre elas, redes neurais é uma das mais utilizadas (SUBASI, 2007; ÜBEYLI; GÜLER, 2005; REVETT *et al.*, 2006; OKANDAN; KARA, 2007). Rede neural artificial é uma das técnicas de aprendizado de máquina largamente utilizada em problemas de classificação, sendo empregada em diversos domínios

práticos e ainda em muitas pesquisas. Nos últimos anos, outra técnica vem ganhando espaço na resolução de problemas de classificação devido ao seu alto desempenho na resolução destes problemas. Está técnica é conhecida como Máquina de Vetor de Suporte (*Support Vector Machine* – SVM). Ela vem sendo empregada com sucesso na classificação de diversos tipos de dados, inclusive de sinais de EEG (ÜBEYLI; GÜLER, 2007; CRISTIANINI; SHAWE-TAYLOR, 2000; OKANDAN; KARA, 2007; GUNN, 1998; VAPNIK, 1995). SVM é uma técnica de classificação que utiliza funções de kernel para mapear os dados em um espaço de dimensão mais alta, onde os dados são separados linearmente por um hiper-plano.

Nesta dissertação, é investigada a influência da técnica de extração de características utilizando coeficientes wavelet no desempenho de classificadores baseados em SVM e suas variantes para o problema de classificação de sinais de eletroencefalograma.

1.2 OBJETIVOS

O objetivo deste trabalho é analisar a influência da extração de características utilizando coeficientes wavelet para classificação de subconjuntos de um conjunto de sinais de EEG, utilizando como classificadores as Máquinas de Vetores de Suporte e algumas de suas variantes. Além disso, será analisada também a influência do tipo de função de kernel e seu parâmetro no desempenho final dos classificadores.

Os experimentos não têm como foco identificar alguma doença específica, elas visam medir a capacidade dos classificadores SVM e dos coeficientes wavelet em distinguir entre elementos com propriedades diferentes, i.e, pertencentes a subconjuntos distintos. Estes subconjuntos, por sua vez, podem estar associados a doenças ou estados específicos do paciente.

1.3 METODOLOGIA

Inicialmente, este estudo se concentra na familiarização e extração de características para sinais de EEG utilizando estatísticas sobre coeficientes wavelet. Uma descrição detalhada da extração de características utilizando wavelet é apresentada. Posteriormente, é realizado um estudo aprofundado sobre os modelos de classificação de sinais, com foco nos modelos baseados em Máquinas de Vetores Suporte. Por fim, é efetuada uma análise

comparativa da influência dos vetores de características, resultantes do cálculo de estatísticas sobre os coeficientes wavelet, no desempenho de classificadores SVM. Todas as implementações foram realizadas utilizando o software MATLAB.

1.4 PRINCIPAIS CONTRIBUIÇÕES DESTA DISSERTAÇÃO

As principais contribuições advindas desta dissertação são as seguintes:

i. Apresentar um estudo da arte sobre transformada wavelet, destacando suas vantagens e desvantagens como uma técnica de extração de características;
ii. Implementar os classificadores baseados nas Máquinas de Vetores Suporte e suas variações;
iii. Analisar a influência das variações das SVM na classificação de sinais de EEG;
iv. Analisar a influência do tipo de kernel e do seu parâmetro associado no desempenho final do classificador;
v. Analisar a influência do tipo de wavelet no desempenho final do classificador.

1.5 DESCRIÇÃO DOS CAPÍTULOS

Os capítulos a seguir descrevem em detalhes todos os resultados obtidos na busca do atendimento dos objetivos descritos anteriormente. Sendo assim, iremos descrever a seguir a forma de organização em termos dos objetivos propostos. Todos os capítulos, com exceção do Capítulo 1, contêm um resumo introdutório.

Tabela 1.1 – Descrição dos capítulos.

Capítulo 1	Apresenta as principais motivações e os objetivos deste trabalho.
Capítulo 2	Realiza uma descrição das principais técnicas de extração de características empregadas na análise de sinais de EEG. O foco principal deste capítulo é descrever a transformada de wavelet.
Capítulo 3	Realiza uma descrição das Máquinas de Vetores Suporte e suas variantes.
Capítulo 4	Apresenta os resultados experimentais obtidos.
Capítulo 5	Apresenta conclusões e trabalhos futuros.

CAPÍTULO 2 - EXTRAÇÃO DE CARACTERÍSTICAS

RESUMO: Este capítulo apresenta uma visão geral dos conceitos básicos que são de grande importância para a compreensão das técnicas de extração de características, como a teoria de wavelet. Na primeira parte são apresentados os conceitos básicos relacionados com a Transformada de Fourier. O objetivo deste é explicar por que o emprego da transformada de Fourier, que é a transformada mais aplicada em processamento de sinais, não é uma técnica adequada para o problema a ser tratado nesta dissertação. Posteriormente, será apresentada a Transformada de Fourier de Curto Tempo (do inglês *Short Time Fourier Transform* - STFT), que surgiu com o objetivo de tentar corrigir as deficiências da Transformada de Fourier nas representações tempo-freqüência de sinais não estacionários. A compreensão total da STFT é de suma de importância, já que a Transformada Wavelet foi desenvolvida como uma alternativa a STFT para resolver alguns problemas presentes na mesma. Para a compreensão de todos os conceitos que são mostrados ao longo deste capítulo se faz emprego de vários exemplos que podem ser reproduzidos facilmente fazendo uso dos programas no MATLAB™ que se indicam em cada caso.

2.1 INTRODUÇÃO

Muitos dos fenômenos físicos podem ser descritos mediante um sinal no domínio do tempo; isto é, uma das variáveis é tempo (variável independente) e outra a amplitude (variável dependente). Quando é realizado o gráfico deste sinal se obtém uma função tempo-amplitude, contudo a informação que se pode obter diretamente desta representação nem sempre é mais apropriada, posto que a informação que caracteriza o sinal, em muitos casos, pode ser observada mais claramente no domínio da freqüência, isto é, mediante um espectro de freqüência que mostre as freqüências existentes no sinal.

2.2 TRANSFORMADA DE FOURIER

Séries de dados temporais podem ser representadas de diferentes formas, dependendo do interesse em visualizar uma ou outra característica. Dentre essas várias formas de representação, a representação no domínio da freqüência é uma das mais conhecidas. A principal vantagem da representação em freqüência em relação à representação no tempo, é que a primeira permite uma clara visualização das periodicidades do sinal (SEMMLOW, 2004). Estas técnicas consistem em determinar as componentes de freqüência de um sinal e são conhecidas como análise espectral. Pode-se utilizar como analogia para ilustrar este conceito, que a análise espectral equivale a um "prisma matemático", que decompõe uma forma de onda nas freqüências que a compõem, assim como um prisma decompõe a luz nas cores de suas componentes primárias (SEMMLOW, 2004).

A análise espectral possui um longo histórico de aplicações em diversos problemas, e várias técnicas foram desenvolvidas para realizá-la, onde merece ser citada a transformada de Fourier. A transformada de Fourier produz uma representação de um sinal reescrevendo-o a partir da combinação de funções exponenciais complexas (POLLOCK, 1999) – i.e. pertencentes ao conjunto dos números complexos. Ou, de forma equivalente, por funções trigonométricas, já que a fórmula de Euler estabelece:

$$e^{j\theta} = \cos\theta + j\sin\theta \qquad 2.1$$

A transformada de Fourier para um sinal contínuo $x(t)$ é dada pela equação a seguir (LATHI; DING, 2009):

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi f t}\, dt \qquad 2.2$$

A transformada inversa de Fourier é dada pela equação 2.3.

$$x(t) = \int_{-\infty}^{+\infty} X(f)\, e^{j2\pi f t}\, df \qquad 2.3$$

Muitas vezes a expressão que descreve um sinal que se deseja analisar não está disponível, o que inviabiliza a utilização de um método analítico como o descrito acima. Nestes casos, utiliza-se a transformada discreta de Fourier, que é determinada com um método numérico. Para sinais discretos, ou para sinais contínuos amostrados, utiliza-se a transformada de Fourier discreta, que é dada pela equação 2.4:

$$X[n] = \sum_{n=0}^{N-1} x[n]\, e^{\left(\frac{-j2\pi mn}{N}\right)} \qquad 2.4$$

onde N é o número total de pontos do sinal discreto, m é um inteiro tal que $m = 0, ..., N - 1$ (SEMMLOW, 2004). A transforma inversa de Fourier discreta pode ser calculada pela equação 2.5.

$$x[m] = \frac{1}{N}\sum_{n=0}^{N-1} X[n]\, e^{\left(\frac{j 2\pi m n}{N}\right)} \qquad 2.5$$

Por volta dos anos 60 um eficiente conjunto de técnicas computacionais para calcular a transforma de Fourier discreta se tornou disponível e teve grande impacto em muitas áreas como ciências aplicadas e engenharia, onde a falta de métodos computacionais era um obstáculo para a sua utilização (POLLOCK, 1999). Este conjunto de técnicas ficou conhecido como transformada rápida de Fourier (do inglês *Fast Fourier Transform* – FFT). Ele é largamente empregado no processamento de sinais, e foi desenvolvido com o objetivo de tornar mais rápido o cálculo da transformada discreta de Fourier. Implementações deste método estão disponíveis em vários softwares como MATLAB^{TM} e Mathematica^{TM}.

A transformada rápida de Fourier se baseia na decomposição da transformada discreta, de uma determinada seqüência finita, em cálculos sucessivos de transformadas discretas menores. A forma de implementar este processamento é que diferencia os vários algoritmos de FFT. Dentre os vários algoritmos para cálculo da FFT pode ser citada a dizimação temporal (do inglês *decimation-in-time*). Para efeito de ilustração deste tipo de algoritmo, considere uma seqüência $x[n]$ com N pontos (onde por conveniência na ilustração foi assumido que N é um inteiro potência de 2). Como a seqüência tem um número par de elementos, pode-se calcular a transformada de Fourier $X[k]$ pela separação de $x[n]$ em duas seqüências de $N/2$ elementos cada uma, sendo uma formada pelos pontos pares e a outra pelos pontos ímpares.

$$X[k] = \sum_{n=0}^{N-1} x[n]\, E_N^{nk}, \qquad k = 0, 1, ..., N-1 \qquad 2.6$$

onde $E_N^{nk} = e^{-j(2\pi/N)kn}$.

Separando nas duas seqüências (pares e ímpares):

$$X[k] = \sum_{\text{pares}} x[n]\, E_N^{nk} + \sum_{\text{ímpares}} x[n]\, E_N^{nk} \qquad 2.7$$

Fazendo uma substituição de variáveis, $n = 2r$ para a seqüência dos pontos pares e $n = 2r + 1$ para a seqüência dos pontos ímpares:

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]\, E_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1]\, E_N^{(2r+1)k} = \sum x[2r]\,(E_N^2)^{rk} + \sum x[2r+1]\,(E_N^2)^{rk} \qquad 2.8$$

Como $E_N^2 = e^{-2j(2\pi/N)} = e^{-j2\pi/(N/2)} = E_{N/2}$, podemos reescrever a equação 2.8 como:

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r]E_{N/2}^{rk} + E_N^k \sum_{r=0}^{(N/2)-1} x[2r+1]E_{N/2}^{rk} = G[k] + E_N^k H[k], \qquad k = 0, 1, ..., N-1 \quad 2.9$$

Cada um dos somatórios da equação 2.9 é uma transformada discreta de Fourier de uma seqüência de *N*/2 pontos: sendo o primeiro, G[*k*], a transformada discreta de Fourier dos pontos pares da seqüência original; e o segundo somatório, H[*k*], a transformada discreta de Fourier dos pontos ímpares da seqüência original. Apesar de *k* variar de 0 até *N*-1, cada um dos somatórios só precisa ser calculado para *k* entre 0 e (N/2) -1, já que G[*k*] e H[*k*] são periódicos em *k* com período *N*/2.

Após o cálculo das duas transformadas discretas, G[*k*] e H[*k*], elas são combinadas de acordo com a equação 2.9 para produzir X[*k*]. Essa combinação é esquematizada na Ilustração 2.1, supondo uma seqüência inicial de tamanho igual a 8 (*n* = 8).

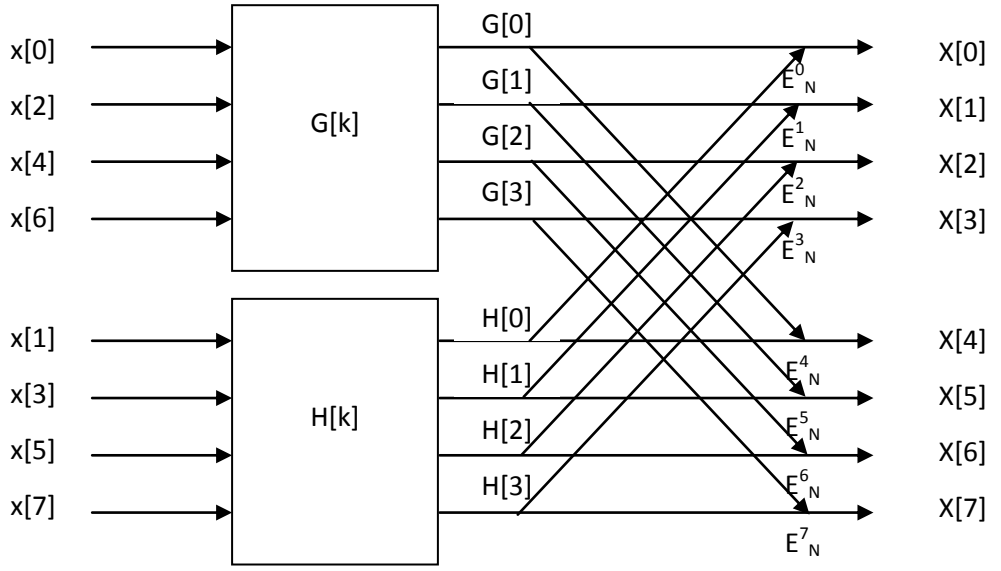


Ilustração 2.1 – Decomposição de uma transformada de Fourier discreta de N pontos em duas transformadas de Fourier discretas de N/2 pontos.

Na Ilustração 2.1, *X*[0] é obtido pela multiplicação entre *H*[0] e E_N^0 e adicionando o produto a G[0]. X[1] é obtido pela multiplicação entre H[1] e E_N^1 e adição do produto a G[1]. Para calcular X[4], a equação 2.9 diz que devemos multiplicar H[4] e E_N^4 e adicionar o produto a G[4]. Entretanto G[k] e H[k] são periódicos em k com período igual a 4, H[4] = H[0] e G[4] = G[0]. Assim, podemos obter X[4] a partir da multiplicação entre H[0] e E_N^4 e somar o resultado a G[0]. A mesma idéia vale para X[5], X[6] e X[7].

Para efeito de ilustração, o resultado do cálculo da transformada de Fourier discreta (utilizando FFT do MATLAB) de um sinal discreto é mostrado na Ilustração 2.2. Considere um sinal contínuo x(t) = cos(πt/5) com freqüência igual a 0.1 Hz. Suponha que este sinal seja amostrado a 1 Hz, ou seja, período de amostragem igual a 1 s. Logo o sinal discreto correspondente pode ser descrito como x[n] = cos(πn/5). O gráfico com o sinal contínuo, sinal discreto e a transformada de Fourier é apresentado a seguir.

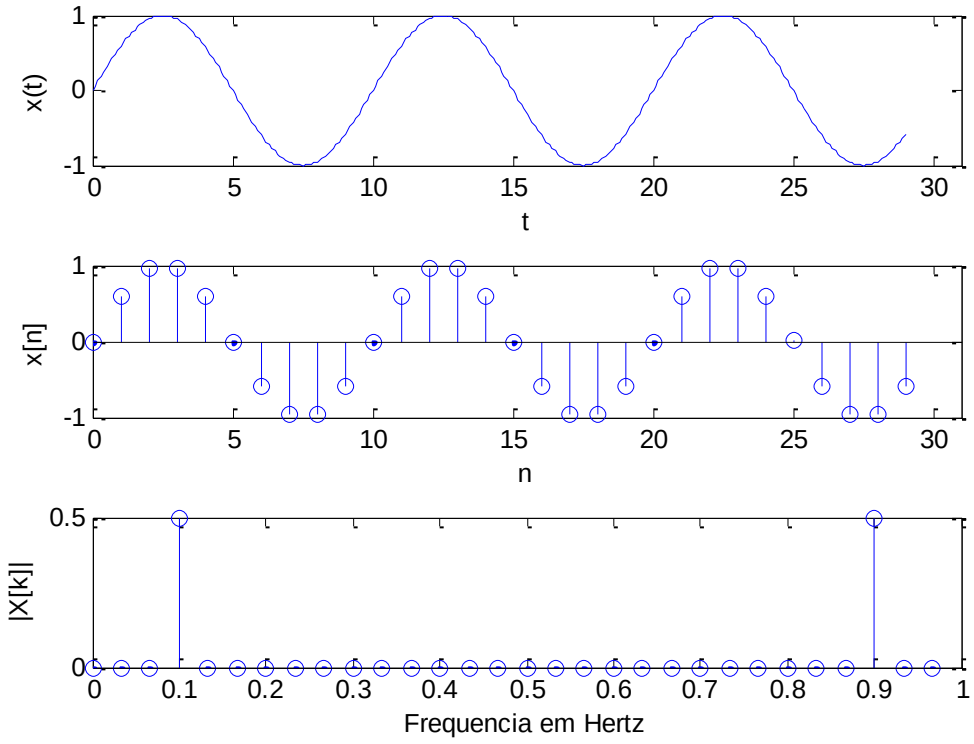


Ilustração 2.2 – Ilustração da transformada de Fourier do sinal x(t).

Observe na Ilustração 2.2 que |X[k]| é simétrico, portanto a segunda metade do gráfico é redundante. A transformada de Fourier apresenta a informação em freqüência do sinal, porém não indica o instante de tempo no qual esta aparece; esta informação não é necessária quando o sinal é estacionário; todavia é de crucial importância sinais não estacionários.

Portanto, a transformada de Fourier é uma ferramenta poderosa para o processamento de sinais e oferece uma solução adequada e completa para sinais estacionários, i.e., sinais que quando divididos em intervalos de tempo, os vários segmentos resultantes apresentam essencialmente as mesmas propriedades estatísticas (SEMMLOW, 2004). Entretanto, vários sinais (especialmente os de origem biológica), mudam suas propriedades substancialmente com o passar do tempo. Por exemplo, os sinais de EEG mudam consideravelmente

dependendo de vários estados internos do paciente, como por exemplo: se o paciente está em meditação ou repouso, se está dormindo ou acordado, se está com os olhos abertos ou fechados. Além disso, muitas vezes o objetivo é analisar essas mudanças (SEMMLOW, 2004). A fim de suprir esta deficiência foi proposta a transforma de Fourier de curto tempo.

2.3 TRANSFORMADA DE FOURIER DE CURTO TEMPO

Conforme mencionado na seção anterior, a transformada de Fourier provê uma boa descrição das freqüências de um sinal, mas não do seu comportamento temporal. Uma das primeiras abordagens para suprir essa carência da transformada de Fourier (e oferecer capacidade de análise no tempo e na freqüência) foi subdividir um sinal em segmentos e calcular a transformada de Fourier em cada um deles. Essa abordagem é conhecida como transformada de Fourier de curto tempo (do inglês *Shot-Term Fourier Transform* – STFT). A transformada de Fourier de curto tempo utiliza uma função para segmentar o sinal original (fazer janelas), e então aplica a transformada de Fourier em cada um dos segmentos gerados. Este método tem sido utilizado com sucesso em diversas aplicações biomédicas (SEMMLOW, 2004).

A transformada de Fourier de curto tempo contínua é dada pela equação 2.10.

$$X(f,\tau) = \int_{-\infty}^{\infty} x(t)\, w(t-\tau)\, e^{-j2\pi f t}\, dt \qquad 2.10$$

onde $w(t-\tau)$ é a função que segmenta o sinal $x(t)$ e τ é a variável que faz com que a janela se desloque sobre o sinal $x(t)$. E a transformada inversa é dada pela equação 2.11 a seguir:

$$x(t) = \frac{1}{\|w(t-\tau)\|} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} X(f,\tau)\, w(t-\tau)\, e^{j2\pi f t}\, df\, d\tau \qquad 2.11$$

onde $\|.\|$ é a norma do vetor.

Existem dois principais problemas com a transformada de Fourier de curto tempo (SEMMLOW, 2004), a saber:

i. determinar o tamanho ótimo de cada segmento de modo que este contenha diferentes características pode não ser possível;
ii. tempo versus freqüência: um segmento muito curto para aumentar a resolução no tempo irá causar perda na resolução na freqüência e vice-versa.

2.4 TRANSFORMADA WAVELET

A transformada wavelet é um método de descrever as propriedades de um sinal que muda durante o tempo. Neste método, o sinal não é dividido em segmentos de tempo, mas em segmentos chamados de escala. A transformada wavelet permite analisar os dados em diferentes escalas, ou resoluções, que permitem ver características mais ou menos detalhadas de um sinal. Por exemplo, considere um sinal no domínio de 0 a 1. Este sinal pode ser dividido em duas funções degrau que vão de 0 a 1/2 e de 1/2 a 1. Este sinal pode ser dividido novamente, mas agora utilizando quatro funções degrau de 0 a 1/4, de 1/4 a 1/2, de 1/2 a 3/4 e de 3/4 a 1, e assim sucessivamente. Cada um desses conjuntos representa o sinal original em uma escala ou resolução.

A análise wavelet consiste em adotar uma função base (wavelet base, ou o termo em inglês *mother wavelet*) e representar o sinal original como uma combinação linear de dilatações, contrações e translações (deslocamentos) sobre a função base.

A transformada wavelet contínua é dada pela equação 2.12.

$$W(a,b) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{|a|}} \psi^* \left(\frac{t-b}{a} \right) dt \qquad 2.12$$

onde ψ é uma função wavelet base, a variável *b* desloca a função no tempo, e a variável *a* atua na variação da escala de tempo da função wavelet. Se *a* for maior que 1, a função wavelet sofre uma dilatação no eixo do tempo; e se $0 < a < 1$, então a função wavelet sofre uma contração no eixo do tempo. Valores negativos da variável *a* simplesmente "gira" a função em relação ao eixo do tempo. A função ψ deve obedecer às seguintes condições (SEMMLOW, 2004):

- A integral da função deve ser zero, ou seja, a área total sob a curva deve ser nula.
- A função deve ter energia finita.

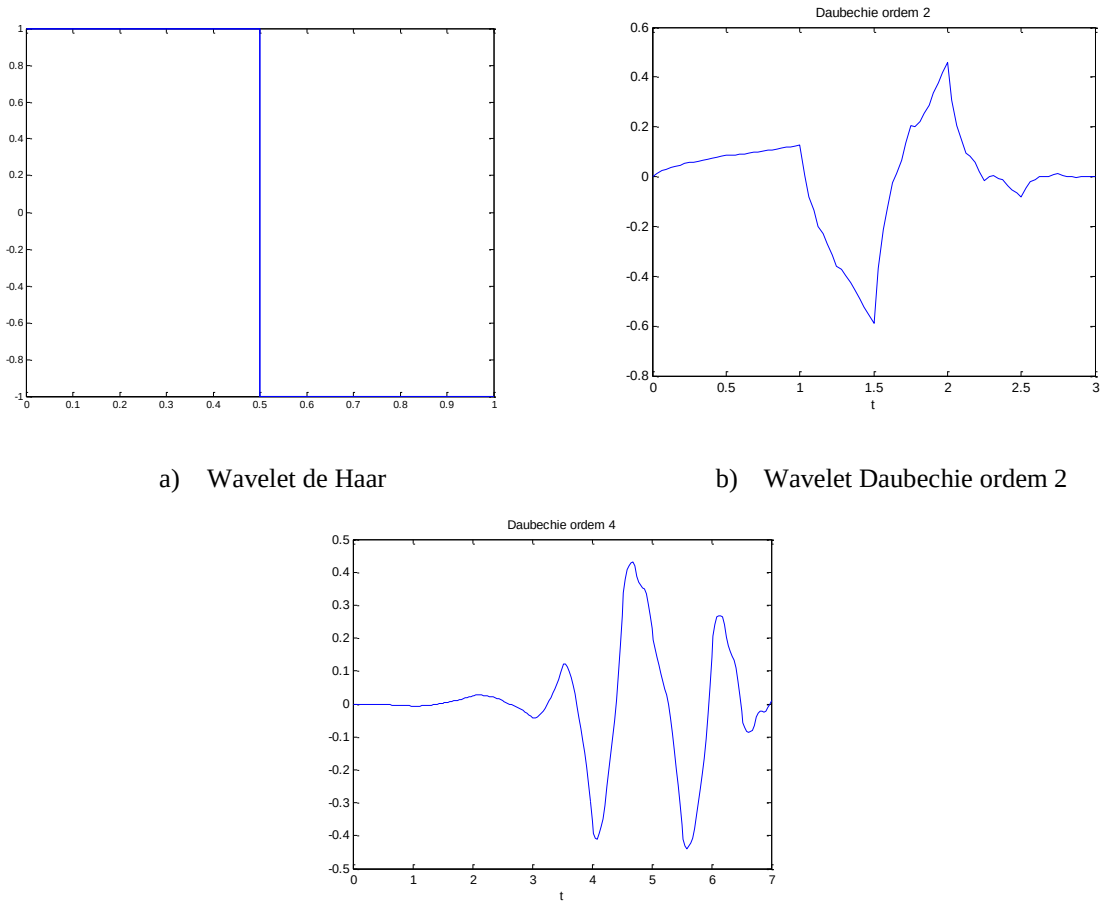
O símbolo * na equação 2.12 indica operação de conjugado complexo, e o fator de normalização $1/\sqrt{|a|}$ garante que a energia é a mesma para todos os valores de *a*, e também para todos os valores de *b* (já que deslocamentos não afetam a energia do sinal). Com $a = 1$ e $b = 0$, a wavelet está na sua forma natural (wavelet base), isto é $\psi_{1,0}(t) \equiv \psi(t)$. As variações de *a* e *b* provocam as alterações que geram as funções membro da família da função wavelet base (SEMMLOW, 2004).

A Ilustração 2.3 apresenta os gráficos das funções wavelet utilizadas nessa dissertação, os itens desta ilustração mostram:

a) Wavelet de Haar;

b) Wavelet Daubechie ordem 2;

c) Wavelet Daubechie de ordem 4.



a) Wavelet de Haar

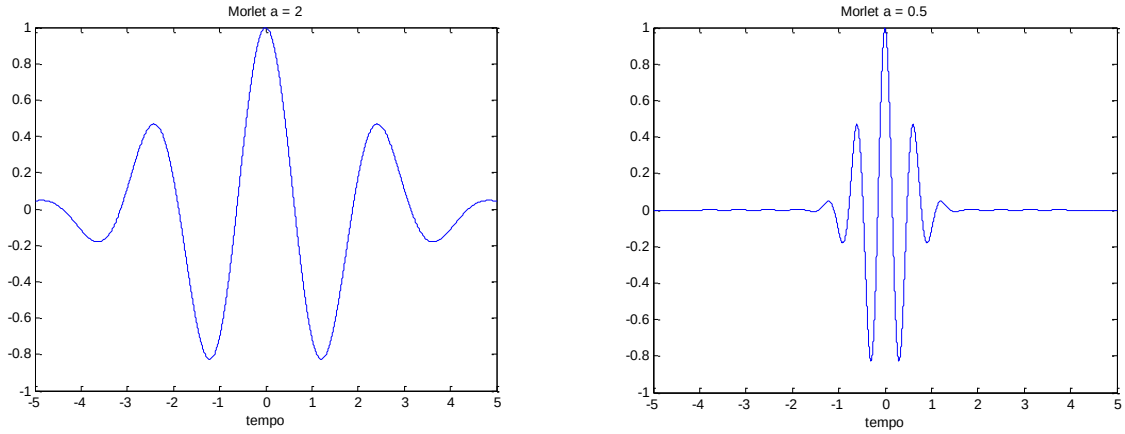
b) Wavelet Daubechie ordem 2

c) Wavelet Daubechie ordem 4

Ilustração 2.3 – Diferentes tipos de wavelet.

A Ilustração 2.4 mostra um wavelet base e as funções geradas por suas contrações e dilatações. A wavelet ilustrada é conhecida como Morlet e é definida pela equação 2.13:

$$\psi(t) = e^{-t^2} \cos\left(\pi\sqrt{\frac{2}{\ln 2}}\, t\right) \qquad 2.13$$



a) Dilatação b) Contração

Ilustração 2.4 – Dilatação e contração da wavelet Morlet.

Os coeficientes wavelet, W(*a*, *b*) (veja equação 2.12), descrevem a relação entre o sinal e a wavelet em várias translações e escalas, isto é, a similaridade entre o sinal e a wavelet em uma dada combinação de escala e posição, *a* e *b*. Os coeficientes fornecem as amplitudes de uma série de wavelets sobre um intervalo de escalas e translações, que serão necessários para reconstruir o sinal original (SEMMLOW, 2004). Se a função wavelet é escolhida de forma apropriada, então é possível reconstruir o sinal original a partir dos coeficientes de wavelet.

Uma vez que a transformada contínua wavelet decompõe o sinal em coeficientes de duas variáveis (*a* e *b*), é necessária uma soma (integral) dupla para reconstruir o sinal original. A inversa da transformada wavelet contínua é dada pela equação 2.14.

$$x(t) = \frac{1}{Ca} \int_{a=-\infty}^{\infty} \int_{b=-\infty}^{\infty} W(a,b)\, \psi_{a,b}(t)\, da\, db \qquad 2.14$$

onde:

$$Ca = \int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{|\omega|} d\omega \qquad 2.15$$

e $0 < Ca < \infty$ é a condição de admissibilidade para a reconstrução do sinal original usando a equação 2.14 (SEMMLOW, 2004).

2.4.1 TRANSFORMADA DE WAVELET DISCRETA

A transformada wavelet contínua apresenta um alto grau de redundância, isto é, ela gera mais coeficientes do que o necessário para especificar unicamente o sinal original. Essa redundância é muito custosa em aplicações que precisam reconstruir o sinal original (SEMMLOW, 2004). A transformada wavelet discreta trata essa redundância restringindo as variações (em translação e escala) a potências de 2. Ainda assim, para eliminar a redundância, a wavelet deve ser adequadamente escolhida de forma a conduzir a uma família ortogonal (i.e. base ortogonal).

A transformada wavelet discreta é comumente introduzida em termos da sua transformada inversa, dada pela equação 2.16 (SEMMLOW, 2004).

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} d(k,l)\, 2^{-k/2}\, \psi(2^{-k} t - l) \qquad 2.16$$

onde k se relaciona com a como $a = 2^k$; b se relaciona com l como $b = 2^k l$; e $d(k,l)$ é uma amostragem de $W(a, b)$ em um ponto discreto k e l.

Na transformada wavelet discreta, é introduzido um conceito de função de escala, uma função que facilita a computação da transformada. Para implementar a transformada discreta eficientemente, a resolução mais fina é calculada primeiro. Depois, as resoluções mais grossas são calculadas, mas não sobre o sinal original e sim sobre as versões das resoluções mais finas calculadas antes. Essas versões são obtidas com a utilização da função de escala. A função de escala é dada pela equação 2.17 a seguir:

$$\phi(t) = \sum_{n=-\infty}^{\infty} \sqrt{2} c(n)\, \phi(2t - n) \qquad 2.17$$

onde $c(n)$ é uma série de escalares que define uma função de escala específica. A própria função wavelet pode ser definida a partir da função de escala:

$$\psi(t) = \sum_{n=-\infty}^{\infty} \sqrt{2} d(n)\, \phi(2t - n) \qquad 2.18$$

onde $d(n)$ é uma série de escalares relacionados ao sinal original $x(t)$ (equação 2.16) e que define a wavelet discreta em termos da função de escala (SEMMLOW, 2004).

2.4.2 CODIFICAÇÃO EM SUB-BANDA

Apesar da transformada wavelet discreta poder ser implementada usando as equações anteriores, ela é comumente implementada usando uma combinação de filtros. O uso de um grupo de filtros para dividir um sinal em vários componentes espectrais é denominado codificação em sub-banda.

Por exemplo, uma implementação simples da transformada wavelet discreta utiliza apenas dois filtros $H_0(\omega)$ e $H_1(\omega)$. As características destes filtros devem ser cuidadosamente escolhidas com $H_0(\omega)$ sendo um filtro passa-baixa e $H_1(\omega)$ um filtro passa-alta. Neste caso, o sinal original é dividido em dois componentes y_{pb} e y_{pa}. A Ilustração 2.5 mostra a utilização destes dois filtros.

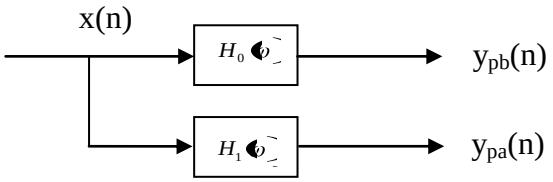


Ilustração 2.5 – Filtros passa-baixa e passa-alta aplicados ao mesmo sinal x(n).

O filtro passa-alta é análogo à aplicação do wavelet ao sinal original, e o filtro passa-baixa é análogo à aplicação da função de escala. Na teoria, se os filtros utilizados possuem filtros inversos então é possível construir filtros complementares que permitirão reconstruir o sinal original a partir de y_{pb} ou y_{pa}. Entretanto, freqüentemente o sinal original pode ser reconstruído mesmo se os filtros não são inversíveis, mas nesse caso serão necessários ambos componentes y_{pb} e y_{pa} (SEMMLOW, 2004). A reconstrução do sinal original é mostrada na Ilustração 2.6, onde a soma de um segundo par de filtros $G_0(\omega)$ e $G_1(\omega)$ é utilizada para gerar uma aproximação do sinal original (SEMMLOW, 2004).

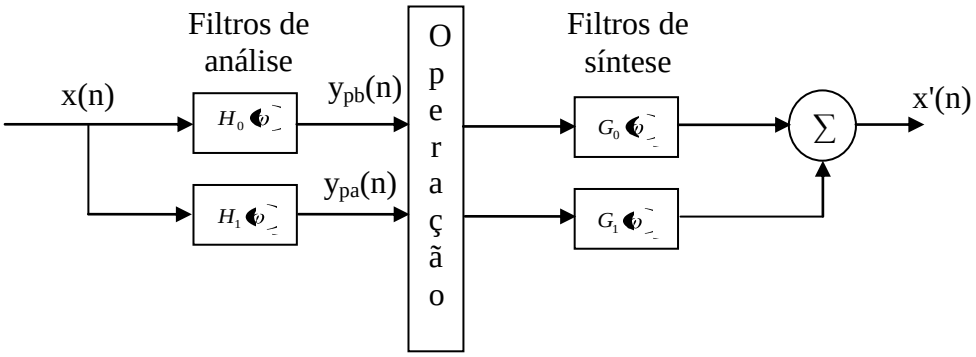


Ilustração 2.6 – Uma aplicação típica de wavelet: o sinal x(n) é decomposto por filtros de análise, as componentes obtidas y_{pb} e y_{pa} são processados por uma operação. Depois o sinal é reconstruído pelos filtros de síntese gerando uma aproximação x'(n).

Em algumas aplicações da transformada wavelet, uma operação é aplicada aos componentes de cada sub-banda (e.g. y_{pb} e y_{pa}), como mostrado na Ilustração 2.6. Já em outras aplicações, como por exemplo, a extração de características realizada nesta dissertação, a reconstrução do sinal original não é necessária.

A abordagem de processamento utilizando wavelet esquematizada na Ilustração 2.6 apresenta um problema, ela requer a geração e a manipulação de duas vezes o número de pontos do sinal original (considerando que a decomposição envolve apenas dois filtros). Esse problema piora quando mais filtros são envolvidos na decomposição. Se os filtros utilizados na decomposição são adequadamente escolhidos, então é possível reduzir o comprimento dos componentes gerados (e.g. y_{pb} e y_{pa}) pela metade e ainda ser capaz de reconstruir o sinal original. Para isso é necessário eliminar, por exemplo, todos os pontos ímpares; esta operação é conhecida como sub amostragem (*donwsampling*) e é ilustrada esquematicamente por $\downarrow 2$. Assim, a versão sub amostrada de um sinal y[n] incluirá apenas os pontos y(2), y(4), y(6), ...
Se a sub amostragem for utilizada, então é necessário um método para recuperar os dados perdidos (os pontos com índices ímpares) para reconstruir o sinal original. Neste caso utiliza-se um aumento da amostragem (*upsampling*), ilustrada por $\uparrow 2$, que irá preencher os dados faltantes com zeros. A Ilustração 2.7 mostra a utilização da sub amostragem para reduzir o número de pontos a serem manipulados.

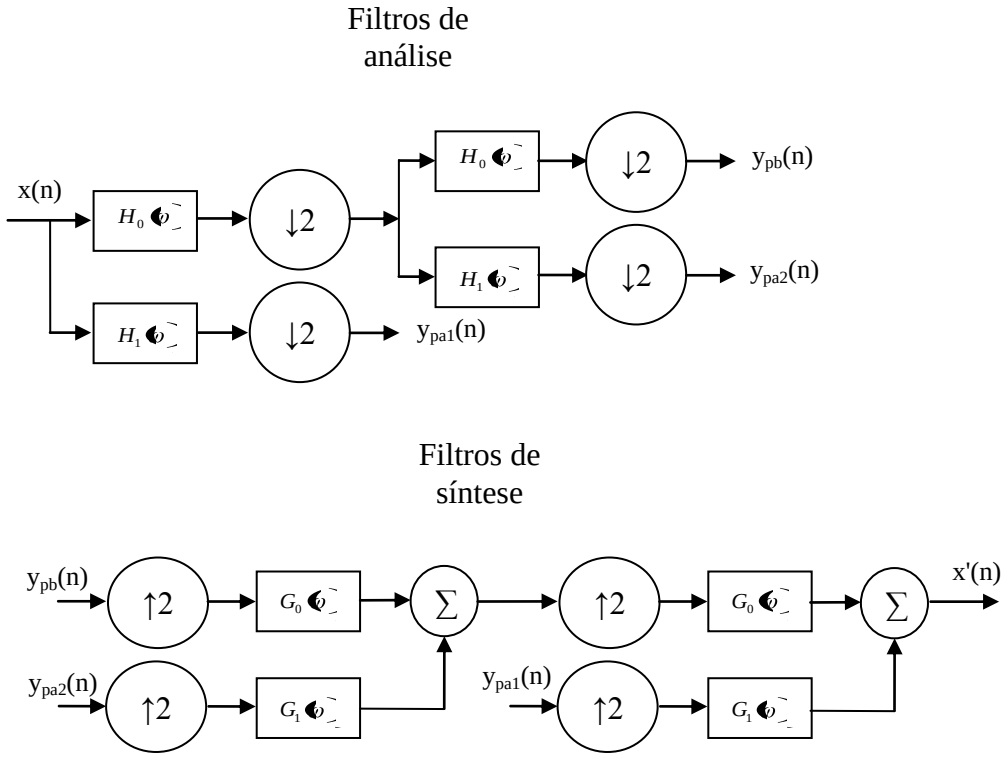


Ilustração 2.7 – A utilização da diminuição e do aumento da amostragem na decomposição e reconstrução de um sinal utilizando wavelet.

A implementação da transformada wavelet discreta como um banco de filtros demanda a escolha de filtros apropriados que atendam a certos critérios; um dos principais é a capacidade de recuperar o sinal após passar pelo processo de análise (decomposição). Após a realização da decomposição e de sub amostragem (como mostrado nas ilustrações 2.5 e 2.6), alguns sinais podem ficar sujeitos a problemas de sobreposição espectral (fenômeno conhecido como *aliasing*). A equação 2.19 fornece um critério para evitar a ocorrência da sobreposição:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 0 \qquad 2.19$$

onde z representa a transformada Z, $H_0(z)$ é a função de transferência do filtro de análise passa-baixa, $H_1(z)$ é a função de transferência do filtro de análise passa-alta, $G_0(z)$ é a função de transferência do filtro de síntese passa-baixa, e $G_1(z)$ é a função de transferência do filtro de síntese passa-alta (SEMMLOW, 2004). Para que seja possível recuperar um sinal, a partir de suas componentes decompostas (e.g. y_{pb} e y_{pa}), a equação 2.20 deve ser satisfeita:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2z^{-N} \qquad 2.20$$

onde as funções de transferência são as mesmas da equação 2.19, e N é o número de coeficientes do filtro, ou seja, a ordem do filtro (SEMMLOW, 2004).

Para que as saídas geradas pelos filtros passa-alta e passa-baixa sejam ortogonais entre si, a equação 2.21 deve ser satisfeita:

$$H_1(z) = -z^{-N} H_0(-z^{-1}) \qquad 2.21$$

Os filtros de síntese se relacionam com os filtros de análise pelas equações 2.22 e 2.23 dadas a seguir (SEMMLOW, 2004):

$$G_0(z) = H_1(z) = z^{-N} H_0(-z^{-1}) \qquad 2.22$$

$$G_1(z) = -H_0(z) = z^{-N} H_1(-z^{-1}) \qquad 2.23$$

O leitor interessado em no projeto de filtros e na transformada Z pode obter mais informações em OPPENHEIN; SCHAFER (2009).

O cálculo dos coeficientes wavelet pode ser implementado computacionalmente como um produto matricial ou com um esquema algoritmo como *lifting scheme* (CALDERBANK *et al.*, 1997).

Pela sua simplicidade foi utilizado o wavelet de Haar para ilustrar o cálculo dos coeficientes wavelet. O wavelet de Haar é uma função degrau e é definido da seguinte forma:

$$\psi(t) = \begin{cases} 1, \text{se } 0 \le t < 1/2 \\ -1, \text{se } 1/2 < t < 1 \\ 0, \text{se } t < 0 \text{ ou } t > 1 \end{cases} \qquad 2.24$$

Utilizando-se o *lifting scheme*, para o wavelet de Haar, os coeficientes de aproximação A_1 e detalhe D_1 (equivalentes aos coeficientes y_{pb} e y_{pa}) são dados pelas equações 2.25 e 2.26 a seguir:

$$A_1 = \frac{x[2j-1] + x[2j]}{\sqrt{2}} \qquad 2.25$$

$$D_1 = \frac{x[2j-1] - x[2j]}{\sqrt{2}} \qquad 2.26$$

onde x[i] é o sinal (ou função) a ser processado pela transformada wavelet, A designa aproximação e D designa detalhe e o índice j varia no intervalo entre 1 e N/2 (onde *N* é o número de pontos do sinal x).

Em uma implementação baseada em um produto matricial, os coeficientes wavelet, para o wavelet de Haar, são o resultado da multiplicação das matrizes na equação 2.27.

$$\begin{bmatrix} \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \begin{bmatrix} x[\\ x[\\ x[\\ x[\\ x[\\ x[\\ x[\\ x[\end{bmatrix} \quad 2.27$$

Para um exemplo prático, considere que o sinal a ser processado pela transformada wavelet é dado pela abaixo:

$$x = [0; -3; 2; 1; 0; 1; 2] \quad 2.28$$

Utilizando-se o *lifting scheme*, para o wavelet de Haar, os coeficientes A e D do sinal x descrito na equação 2.28 são obtidos da seguinte forma:

$$A = \left\{\frac{1+0}{\sqrt{2}}; \frac{-3+2}{\sqrt{2}}; \frac{1+0}{\sqrt{2}}; \frac{1+2}{\sqrt{2}}\right\} = [0,7071; -0,7071; 0,7071; 2,1213] \quad 2.29$$

$$D = \left\{\frac{1-0}{\sqrt{2}}; \frac{-3-2}{\sqrt{2}}; \frac{1-0}{\sqrt{2}}; \frac{1-2}{\sqrt{2}}\right\} = [0,7071; -3,5355; 0,7071; -0,7071] \quad 2.30$$

Considerando essa fórmula de cálculo, o sinal pode ser reconstruído combinando os coeficientes resultantes da seguinte forma:

$$x' = \left\{\frac{A[+ D[}{\sqrt{2}}, \frac{A[- D[}{\sqrt{2}}, \frac{A[+ D[}{\sqrt{2}}, \frac{A[- D[}{\sqrt{2}}, \frac{A[+ D[}{\sqrt{2}}, \frac{A[- D[}{\sqrt{2}}, \frac{A[+ D[}{\sqrt{2}}, \frac{A[- D[}{\sqrt{2}}\right\} \quad 2.31$$

onde x’ é uma reconstrução do sinal original x.

Para tornar o exemplo mais interessante, considere que antes da reconstrução do sinal acontece uma operação como na Ilustração 2.6. Suponha que essa tal operação seja considerar desprezível qualquer coeficiente que tenha valor em módulo menor do que 0,9; desta forma temos os seguintes coeficientes:

$$A' = [0; 0; 2,1213] \quad 2.32$$

$$D' = [-3,5355; 0; 0] \quad 2.33$$

onde A’ e D’ são resultantes da aplicação da operação aos coeficientes A e D respectivamente.

Agora, para reconstruir o sinal é necessário substituir A e D na equação 2.31 por A’ e D’ respectivamente. Assim o sinal reconstruído será:

$$x' = [0;\ -2{,}5;\ 2{,}5;\ 0;\ 0;\ 1{,}5;\ 1{,}5] \qquad 2.34$$

As ilustrações 2.8 e 2.9 mostram respectivamente o sinal original e o sinal reconstruído resultante da operação x’.

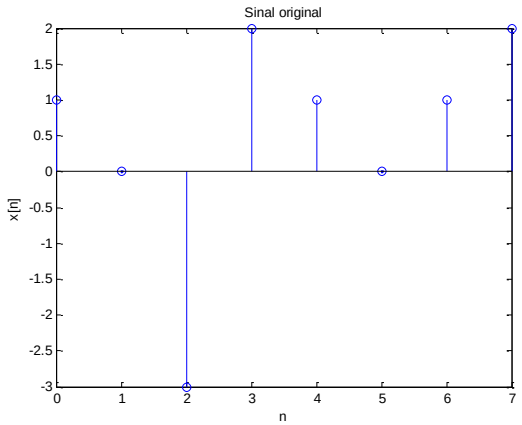


Ilustração 2.8 – Sinal original versus o sinal reconstruído.

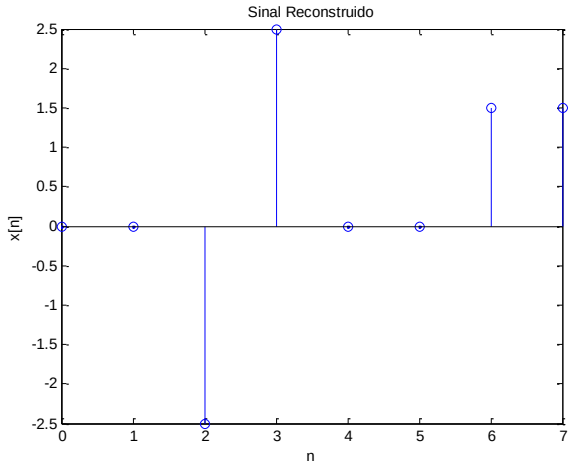


Ilustração 2.9 – Sinal original versus o sinal reconstruído desprezando alguns coeficientes.

Apenas para citar, considerando o cálculo dos coeficientes wavelet utilizando um produto matricial, para realizar a reconstrução de um sinal é necessário recorrer à inversa da matriz utilizada na decomposição.

A escolha do tipo de wavelet e do número de decomposições é de extrema importância para o sucesso de uma análise de sinais usando a transformada discreta wavelet. O número de decomposições é escolhido com base nas componentes do sinal com freqüência dominante (REVETT *et al.*, 2006). Normalmente, são realizados vários testes com diferentes tipos de wavelet, como por exemplo: Daubechie, Morlet, Chapéu Mexicano (*Mexican hat*) e Haar. O

wavelet escolhido será aquele que apresentar o resultado mais adequado para o problema em questão. O wavelet Daubechie é bastante utilizado no processamento de sinais de EEG por ser bem apropriado para analisar mudanças neste tipo de sinal (REVETT *et al.*, 2006; SUBASI, 2007; ÜBEYLI; GÜLER, 2005; SABETI *et al.*, 2007; ÜBEYLI, 2008).

CAPÍTULO 3 - MÁQUINAS DE VETORES SUPORTE E SUAS VARIAÇÕES

RESUMO: As máquinas de vetores-suporte (SVM) representam um procedimento de aprendizado não-paramétrico baseado na teoria de aprendizado estatístico (VAPNIK, 1995), capazes de manipular eficientemente a flexibilidade dos modelos de classificação e regressão em espaços de elevada dimensão. A abordagem SVM é uma implementação do princípio de indução denominado minimização do risco estrutural. Este princípio é baseado no fato de que a razão do erro de aprendizagem sobre o conjunto de dados de teste (isto é, a razão do erro de generalização) é limitada pela soma da razão do erro quadrático no conjunto de treinamento e um termo que depende da dimensão de Vapnik-Chervonenkis (VC), a qual é uma medida da capacidade de expressão de uma família de funções. O objetivo é construir um conjunto de hiperplanos tendo como estratégia a variação da dimensão VC, de modo que o risco empírico e a dimensão VC sejam minimizados ao mesmo tempo. Conseqüentemente, a SVM pode produzir um bom desempenho de generalização para problemas de modelagem não-paramétrica, partindo apenas do conjunto de dados e sem incorporar nenhum conhecimento prévio acerca do domínio do problema. Nas seções seguintes será descrito formalmente as SVM e suas variantes.

3.1 INTRODUÇÃO

Máquina de vetor de suporte (SVM – do inglês *Support Vector Machine*) é uma técnica de aprendizado de máquina largamente empregada em problemas de classificação e regressão (GUNN, 1998). A abordagem SVM é baseada na minimização do risco estrutural, na qual o erro de generalização é delimitado pelo somatório do erro de treinamento e uma parcela que depende da dimensão VC – Vapnik-Chervonenkis (VAPNIK, 1995). A partir da minimização deste somatório, uma grande capacidade de generalização pode ser obtida. Outra característica do aprendizado utilizando SVM é que os subproblemas de otimização são inerentemente convexos e não possuem mínimos locais, isto é, a solução produz um mínimo global. Este fato vem como resultado da aplicação das condições de Mercer na caracterização

dos kernels e também da formulação quadrática do problema (CRISTIANINI; SHAWE-TAYLOR, 2000).

3.2 CONCEITOS BÁSICOS

Nesta seção, serão apresentados alguns conceitos básicos importantes para facilitar a compreensão das Máquinas de Vetores de Suporte em problemas de classificação. Estes conceitos estão relacionados principalmente com o hiperplano de separação, margem de erros na separação e com as funções kernel.

3.2.1 O HIPERPLANO DE SEPARAÇÃO

Considere um problema de classificação envolvendo duas classes linearmente separáveis, e com *N* amostras no conjunto de treinamento $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ com $\mathbf{x} \in \Re^n$ e y $\in \{-1, +1\}$. Um hiperplano de separação para este problema é dado pela equação a seguir

$$(\mathbf{w} \bullet \mathbf{x}) + b = 0 \qquad 3.1$$

onde $(\mathbf{w} \bullet \mathbf{x})$ denota o produto interno entre os vetores **x** e **w**; $\mathbf{w} \in \Re^n$ e $b \in \Re$. O lado caracterizado pela equação 3.2 representa uma classe, e o lado caracterizado pela equação 3.3 representa a outra classe.

$$(\mathbf{w} \bullet \mathbf{x}) + b > 0 \qquad 3.2$$
$$(\mathbf{w} \bullet \mathbf{x}) + b < 0 \qquad 3.3$$

A Ilustração 3.1 mostra um exemplo de um hiperplano de separação.

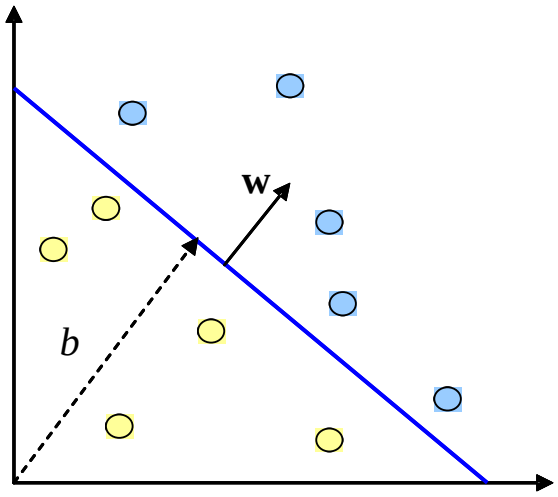


Ilustração 3.1 – Hiperplano de separação em um espaço bidimensional.

3.2.2 O HIPERPLANO DE SEPARAÇÃO ÓTIMO

Um conjunto de vetores é dito ser separado otimamente pelo hiperplano se ele é separado sem erro e se a distância do hiperplano à amostra mais próxima de cada classe é máxima. A distância de uma amostra mais próxima de uma classe à amostra mais próxima da outra classe é denominada de margem de separação $\rho(\mathbf{w},b)$. Logo, um hiperplano ótimo é aquele no qual a margem de separação é máxima. O hiperplano de separação ótimo também é conhecido como hiperplano de margem máxima. Os vetores **x** da amostra de treinamento devem obedecer à seguinte equação:

$$|(\mathbf{w}\bullet\mathbf{x})+b|\geq\frac{\rho(\mathbf{w},b)}{2} \qquad 3.4$$

A distância d(**w**,*b*;**x**$_i$), de um vetor $\mathbf{x}_i\in\Re^n$ ao hiperplano pode ser expressa da seguinte forma:

$$\mathrm{d}(\mathbf{w},\mathrm{b};\mathbf{x}_i)=\frac{|(\mathbf{w}\bullet\mathbf{x}_i)+\mathrm{b}|}{\|\mathbf{w}\|} \qquad 3.5$$

A margem de separação pode ser tão grande quanto se queira bastando ajustar os valores da norma **w** e do parâmetro *b*, assim é necessário impor limites sobre $\rho(\mathbf{w},b)$. Assumindo margem de separação $\rho(\mathbf{w},b)=2$, o problema de maximizar a margem se transforma em um problema equivalente de minimizar a norma de **w**, conduzindo a um hiperplano canônico onde **w** e *b* devem satisfazer a seguinte restrição:

$$\min_{w}|(\mathbf{w}\bullet\mathbf{x})+\mathrm{b}|=1 \qquad 3.6$$

Das equações anteriores, a norma de **w** pode ser expressa como

$$\|\mathbf{w}\| = \frac{2}{\rho(\mathbf{w}, b)} \qquad 3.7$$

Sendo assim, minimizando-se a norma de **w** é equivalente maximizar a margem de separação $\rho(\mathbf{w}, b)$. Como $y \in \{-1, +1\}$, um hiperplano de separação na forma canônica deve satisfazer à seguinte desigualdade junto às amostras de treinamento:

$$y_i \left[(\mathbf{w} \bullet \mathbf{x}_i) + b\right] \geq 1 \qquad 3.8$$

onde *i* varia de 1 até *N*. Assim, o hiperplano de separação ótimo pode ser obtido pela minimização da equação 3.9 sujeito à restrição dada pela equação 3.8:

$$\Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \qquad 3.9$$

Note que *b* não é utilizado no cálculo da maximização da distância. Mas o valor de *b* é importante para localizar corretamente o hiperplano, pois uma alteração no valor desta variável fará com que o hiperplano esteja mais próximo de uma classe que de outra, assim alguns pontos poderão ser classificados incorretamente.

Podem existir vários hiperplanos capazes de separar os pontos de classes distintas em um espaço de alta dimensão, entretanto apenas um dos hiperplanos será ótimo, ou seja, terá sua distância maximizada dos exemplos de cada uma das classes (vide Ilustração **3.2**).

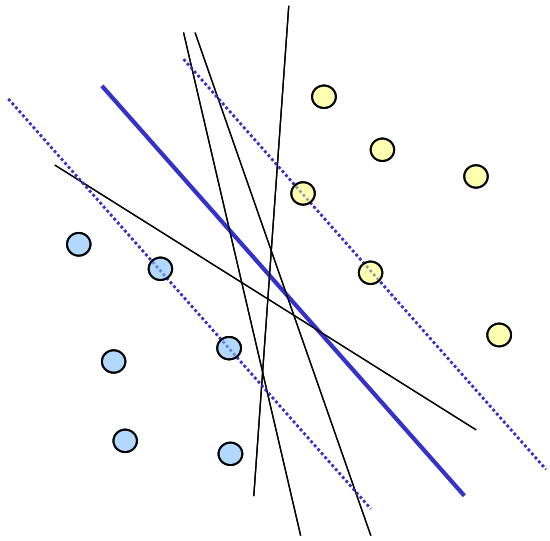


Ilustração 3.2 – Hiperplano de separação ótimo (linha cheia em azul) e vetores de suporte (sobre as linhas azuis tracejadas)

Maximizar a distância entre o hiperplano e os vetores referentes aos exemplos de treinamento de cada classe é importante, pois dá à SVM maior capacidade de generalização,

isto é, classificar corretamente exemplos distintos daqueles utilizados no treinamento, que estejam dentro da margem de separação.

Note na Ilustração **3.2**, que o hiperplano ótimo foi construído usando como referência dois outros hiperplanos (linhas tracejadas), e que esses hiperplanos auxiliares passam por alguns pontos (vetores). Estes pontos representam os chamados *vetores-suporte*.

3.2.3 HIPERPLANO DE MARGEM SUAVE

Existem várias situações onde não é possível separar totalmente os pontos pertencentes às diferentes classes. Nestes casos, é proposta uma modificação no SVM para utilizar uma margem suave de forma a tratar esses casos (VAPNIK, 1995).

A margem suave permite que alguns pontos caiam dentro da margem de separação, podendo inclusive cair no outro lado do hiperplano de separação, produzindo um erro de classificação. Como não se deseja que o SVM aceite um número elevado deste tipo de pontos, a utilização da margem suave introduz um parâmetro para controlar o número de pontos que serão permitidos ultrapassarem o hiperplano de separação.

Quando é esperado que o hiperplano não possa fazer a separação completa dos dados, um método baseado na introdução de uma função-custo (penalidade), associada com o erro da classificação, é introduzido. Uma função de penalidade normalmente adotada (CORTES; VAPNIK, 1995) é a dada pela equação 3.10 a seguir:

$$F_\sigma(\vec{\xi}) = \sum_{i=1}^{N} \xi_i^{\sigma}, \sigma > 0 \qquad 3.10$$

onde cada $\xi_i \geq 0$, $i = 1,...,N$, indica o erro de classificação associado a cada amostra de treinamento e σ caracteriza o tipo de função de perda utilizada, por exemplo $\sigma = 2$, representa função de perda norma 2.

Agora a equação a ser maximizada deve considerar um termo adicional vinculado ao erro de classificação, assim à equação 3.9 será acrescida de um valor de penalidade como o dado pela equação 3.10 (VAPNIK, 1995):

$$\Phi(\vec{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C.\sum_{i=1}^{N} \xi_i \qquad 3.11$$

A adição deste valor pode ser vista como uma tolerância à ocorrência de erros na classificação. Essa tolerância a erros é controlada pelo parâmetro C, que é um valor a ser

arbitrado pelo usuário. Assim, um dos pontos fracos desta abordagem é que não há uma forma ótima para determinação do parâmetro C. Entretanto, é razoável argumentar que o valor de C deve variar com o nível de ruído existente no conjunto de treinamento (CHERKASSKY; MA, 2004).

3.3 FUNÇÕES KERNEL

A função de kernel é uma técnica matemática que permite ao SVM separar linearmente as amostras pertencentes a diferentes classes, sem que estas sejam linearmente separáveis em sua dimensão original. A função de kernel faz o mapeamento dos dados de um espaço de baixa dimensão para um espaço de alta dimensão (hiperespaço). No espaço de alta dimensão, os pontos pertencentes a diferentes classes podem ser linearmente separáveis pelo hiperplano. Esta é a chave do funcionamento do SVM, utilizar uma função de kernel capaz de mapear os dados a partir do seu espaço original em um hiperespaço onde seja possível separar linearmente os pontos pertencentes a diferentes classes. Entretanto, a escolha da função de kernel é um parâmetro a ser ajustado na implementação de um SVM; e na ausência de uma técnica para fazer tal escolha, freqüentemente recorre-se ao método de "tentativa e erro" para determinar qual função de kernel apresenta os melhores resultados em um dado problema. Nesta dissertação, será analisado o impacto da escolha da função kernel no desempenho da classificação de sinais de EEG.

3.4 VARIAÇÕES DE IMPLEMENTAÇÕES DE SVM

A seguir, são apresentadas algumas variações de implementação de SVM. Considere um conjunto de treinamento $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ com a entrada $\boldsymbol{x}_\mathrm{i}$ pertencente a $\Re^\mathrm{n}$ e y_i pertencente a {+1,-1}. A SVM primeiro faz um mapeamento ϕ: $\Re^\mathrm{n} \rightarrow \Re^\mathrm{m}$. Onde *m* é usualmente muito maior que *n*, de modo que o vetor de entrada seja mapeado em um espaço de alta dimensão. Quando os dados são linearmente separáveis, a SVM constrói um hiper-plano em $\Re^\mathrm{n}$ $\mathbf{w}^T\phi(\mathbf{x})+b$ em que a fronteira entre os exemplos positivos e negativos é maximizada. Pode ser mostrado que **w**, para este hiper-plano ótimo, pode ser definido como a combinação linear $\phi(\mathbf{x}_i)$, que é $\mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i \phi(\mathbf{x}_i)$.

A implementação da SVM tradicional implica em resolver um problema de programação quadrática que demanda alto custo computacional. Com o objetivo de reduzir essa complexidade computacional foram propostas algumas alterações na formulação original, dentre as quais podem ser citadas: Least Square SVM (Suykens; Vandewalle, 1999) e Proximal SVM (Fung; Mangasarian, 2001), que conduzem a classificadores rápidos e simples, que demandam a solução de um sistema linear de equações. A Smooth SVM (Lee; Mangasarian, 2001) utiliza métodos de suavização, comumente aplicados em problemas matemáticos, para reformular a SVM tradicional como um problema sem restrições. Lagrangian SVM (Mangasarian; Musicant, 2001) que possui um algoritmo simples baseado em uma formulação Lagrangiana implícita.

3.4.1 SVM TRADICIONAL

A implementação tradicional da SVM consiste em encontrar o hiper-plano de separação ótimo determinado pelo vetor **w**, que minimize a equação 3.11 sujeito às restrições a seguir:

$$y_i[\mathbf{w}^T \varphi(\mathbf{x}_i) + b] > 1 - \xi_i \qquad 3.12$$

onde *i* varia de 1 até *N*, e $\varphi(\cdot)$ mapeia o vetor **x** em um espaço de hiper-dimensão. Uma alternativa para resolver este problema de otimização é utilizar multiplicadores de Lagrange para gerar uma formulação dual do problema. De acordo com CRISTIANINI; SHAWE-TAYLOR (2000), muitas vezes é mais fácil resolver essa forma dual do problema que manipular inequações de restrição como a inequação 3.12.

Então, o problema de programação quadrática na equação 3.11 pode ser escrito de maneira dual como:

$$\max_{\alpha} \; J(\alpha) = \max_{\alpha} \; \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \qquad 3.13$$

sujeito a $\sum_{i=1}^{N} \alpha_i y_i = 0$ e $0 \le \alpha_i \le C$, onde α_i são os multiplicadores de Lagrange e *i* = 1, ..., N. Para obter $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ na equação 3.13 não é preciso calcular $\phi(\mathbf{x}_i)$ e $\phi(\mathbf{x}_j)$ explicitamente. Ao invés disso, para algum ϕ, pode ser escolhido um kernel *K*(.,.) tal que $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Por exemplo, o kernel para um classificador com função base radial

(*radial basis function* – RBF) com variância σ^2 é $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2\sigma^2}\right)$. Então, a equação 3.13 se torna:

$$\max_{\alpha} J(\alpha) = \max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad 3.14$$

Para as amostras do conjunto de treinamento ao longo da fronteira de decisão, os correspondentes $\alpha_i's$ são maiores que zero, de acordo com o Teorema Kuhn-Tucker (FLETCHER, 1987). Essas amostras são chamadas de vetores de suporte. O número de vetores de suporte é normalmente muito menor do que *N*. Um vetor de teste $\mathbf{x} \in \Re^n$ é então atribuído a uma dada classe com respeito à expressão:

$$f(\mathbf{x}) = sign(w^T \phi(\mathbf{x}) + b) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \qquad 3.15$$

3.4.2 LEAST SQUARE SVM (LS-SVM)

Em SUYKENS; VANDEWALLE (1999), um classificador SVM do tipo mínimos quadrados (*Least Squares* – LS-SVM), foi introduzido. Este emprega uma função custo de mínimos quadrados, com uma igualdade em vez de desigualdade nas restrições. Portanto, os parâmetros **w** e *b* do classificador podem ser obtidos resolvendo o seguinte problema de otimização.

$$\min_{w,b,e} J(\mathbf{w}, b, e) = \tfrac{1}{2}(\mathbf{w}^T\mathbf{w}) + \tfrac{C}{2}\sum_{t=1}^{N}(\xi_i)^2 \qquad 3.16$$

sujeito a $y_i[\mathbf{w}^T\varphi(\mathbf{x}_i) + b] = 1 - \xi_i$, e i = 1, ..., N. Após algumas manipulações algébricas, incluindo a eliminação de **w**, é obtido o seguinte sistema linear KKT (Karush-Kuhn-Tucker).

$$\begin{bmatrix} 0 & -\mathbf{Y}^T \\ \mathbf{Y} & \Omega + C^{-1}I \end{bmatrix}\begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{1} \end{bmatrix} \qquad 3.17$$

com $\mathbf{Y} = [y_1; \cdots; y_N]$, $\vec{1} = [1; \cdots; 1]$, $\boldsymbol{\alpha} = [\alpha_1; \cdots \alpha_N]$, $\Omega_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ e para i, j = 1, ..., N.

Logo os coeficientes de Lagrange e o parâmetro *b* podem ser encontrados solucionando o sistema de equações linear descrito pela equação 3.17.

3.4.3 SMOOTH SVM (SSVM)

Smooth SVM (SSVM) é o resultado de duas mudanças simples na SVM tradicional: (i) a margem (distância) entre os planos separadores paralelos é maximizada considerando ambos **w** e b; (ii) o erro na margem suave foi minimizado usando norma-2 ao invés da norma-1 convencional. Então a função objetivo a ser minimizada pode ser escrita como:

$$\min_{w,b,e} J(\mathbf{w},b,e) = \tfrac{1}{2}(\mathbf{w}^T\mathbf{w}+b^2) + \tfrac{1}{2}C\sum_{i=1}^{N}\xi_i^2 \qquad 3.18$$

onde *C* é um valor definido a priori. Esta equação está sujeita às restrições $y_i[\mathbf{w}^T\varphi(\mathbf{x}_i)+b] > 1-\xi_i$, *i* = 1, ..., *N*. A solução para estas restrições é dada pela equação 3.19 a seguir:

$$\xi_i = (1-y_i[\mathbf{w}^T\varphi(\mathbf{x}_i)+b])_+ \qquad 3.19$$

em que $(\cdot)_+$ significa que apenas os valores positivos são considerados, i.e., os valores tais que $y_i[\mathbf{w}^T\varphi(\mathbf{x}_i)+b] < 1$. Assim, substituindo a equação 3.19 na equação 3.18 resulta no problema de otimização sem restrições a seguir (LEE; MANGASARIAN, 2001):

$$\min_{w,b,e} J(\mathbf{w},b,e) = \tfrac{1}{2}(\mathbf{w}^T\mathbf{w}+b^2) + \tfrac{C}{2}\left\|(1-y_i[\mathbf{w}^T\varphi(\mathbf{x}_i)+b])_+\right\| \qquad 3.20$$

Entretanto, a equação 3.20 não é duplamente diferenciável, fato este que impossibilita o uso do método de Newton em sua solução (XIONG *et al.*, 2006). As técnicas de suavização (*smoothing*) são aplicadas substituindo-se $(\cdot)_+$ por uma aproximação *p*(z,k) dada, por exemplo, pela equação 3.21.

$$p(z,k) = z + \frac{1}{k}\log(1+e^{-kz}) \qquad 3.21$$

onde $z = (1-y_i[\mathbf{w}^T\varphi(\mathbf{x}_i)+b])_+$, *k* > 0 é chamado de parâmetro de suavização. Fazendo esta substituição obtemos a equação a seguir:

$$\min_{w,b,e} J(\mathbf{w},b,e) = \tfrac{1}{2}(\mathbf{w}^T\mathbf{w}+b^2) + \tfrac{C}{2}\left\|p(1-y_i[\mathbf{w}^T\varphi(\mathbf{x}_i)+b],k)\right\|^2 \qquad 3.22$$

De acordo com (XIONG *et al.*, 2006), a equação 3.22 é duas vezes diferenciável, o que torna possível resolver esta equação utilizando um algoritmo programação quadrática. A solução para a equação 3.20 é obtida resolvendo-se a equação 3.22 para um *k* que tenda ao infinito.

Como mostrado em (XIONG *et al.*, 2006), é possível substituir o $(\cdot)_+$ por outras funções de aproximação, como as equações 3.23 e 3.24, que resultam respectivamente nas

implementações 1PSSVM (*1st-order polynomial smooth SVM*) e 2PSSVM (*2nd-order polynomial smooth SVM*).

$$p(z,k) = \frac{k}{4}z^2 + \frac{1}{2}z + \frac{1}{4k} \qquad 3.23$$

$$p(z,k) = \frac{1}{16k}(z+1)^2(z-3) \qquad 3.24$$

3.4.4 LAGRANGIAN SVM (LSVM)

A SVM Lagrangiana (*Lagrangian* SVM – LSVM) foi proposta baseada em uma formulação Lagrangiana implícita do dual de um SVM tradicional. Isso conduz à otimização de uma função convexa diferenciável sem restrições, em um espaço de dimensão igual ao número de pontos a serem classificados (MANGASARIAN; MUSICANT, 2001).

A função objetivo é a mesma da SSVM (equação 3.18), ou seja, na LSVM a margem de separação é maximizada em função de **w** e da localização em relação à origem *b*. Além disso, a margem suave (tolerância a erros) foi otimizada usando norma-2 ao invés da norma-1 utilizada na SVM tradicional. O problema de programação quadrática resultante pode ser escrito como:

$$\max_{\alpha} \; J(\alpha) = \max_{\alpha} \; \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j H(\mathbf{x}_i, \mathbf{x}_j) \qquad 3.25$$

sujeito a $\alpha_i \geq 0$ para $i = 1, ...,N$ e $H(\mathbf{x}_i, \mathbf{x}_j) = diag\left\{\frac{1}{C}\right\} + K(\mathbf{x}_i, \mathbf{x}_j)$.

Colocando na forma matricial, teremos:

$$\max_{\alpha \geq 0} J(\alpha) = \max_{\alpha} \frac{1}{2}\boldsymbol{\alpha}^T H\boldsymbol{\alpha} + \vec{1}^T > 0 \qquad 3.26$$

com $\vec{1} = [1, ..., 1]^T$.

De acordo com MANGASARIAN; MUSICANT (2001), a LSVM requer a inversão no início de uma matriz da ordem da dimensionalidade do espaço de entrada original mais um (*n* + 1).

3.4.5 PROXIMAL SVM (PSVM)

A idéia chave nessa abordagem é muito simples e similar à da LS-SVM. A mudança fundamental aqui é a substituição da equação 3.16 por:

$$\min_{w,b,e} \; J(\mathbf{w},b,e) = \tfrac{1}{2}(\mathbf{w}^T\mathbf{w} + b^2) + \tfrac{1}{2}C\sum_{i=1}^{N} \xi_i^2 \qquad 3.27$$

(onde C é um valor dado) sujeito às restrições $y_i[\mathbf{w}^T\varphi(\mathbf{x}_i) + b] = 1 - \xi_i$, i = 1, ..., N.

Após algumas manipulações algébricas, incluindo a eliminação do **w**, é obtido um sistema linear KKT (Karush-Kuhn-Tucker), similar àquele associado com a formulação da LS-SVM (FUNG; MANGASARIAN, 2001).

Geometricamente esta mudança pode ser interpretada da seguinte forma: os dois planos determinados por $\mathbf{w} \bullet \mathbf{x} = b \pm 1$, podem ser vistos não mais como fronteiras, mas sim como planos de aproximação, ao redor dos quais os pontos de uma classe se distribuem. E a distância entre estes planos é maximizada. Um ponto será rotulado como +1 ou -1 dependendo se está mais próximo de $\mathbf{w} \bullet \mathbf{x} = b + 1$ ou de $\mathbf{w} \bullet \mathbf{x} = b - 1$, respectivamente.

3.4.6 MÁQUINA DE VETORES RELEVANTES

Máquinas de Vetores Relevantes (do inglês *Relevance Vector Machine* – RVM) são um método de aprendizado estatístico e também um modelo probabilístico baseado em métodos de kernel. RVM tem uma forma funcional idêntica a da SVM, entretanto difere desta por fornecer uma classificação probabilística.

Apesar de serem empregadas com sucesso na resolução de diversos problemas, as SVMs apresentam algumas desvantagens práticas. Tipicamente o número de vetores de suporte necessários nas SVMs cresce linearmente com o tamanho do conjunto de treinamentos; na implementação clássica das SVMs, a as predições e as classificações não são probabilísticas; é necessário determinar o parâmetro C (para controlar os erros), o que é geralmente feito com a utilização de métodos que demandam bastante poder computacional, como por exemplo, o processo de validação cruzada. Além disso, as funções de kernel a serem utilizadas nas SVMs devem satisfazer às condições de Mercer. As RMVs foram propostas como um tratamento bayesiano alternativo à SVM, e não sofrem das desvantagens

citadas anteriormente para as SVMs (TIPPING, 2001). Estudos indicam que a função de decisão gerada por um RVM pode depender de menos dados de entrada que um SVM tradicional (LIMA *et al.*, 2009).

Se o erro entre a saída do classificador dada por $f(\mathbf{x}_i;\mathbf{w})$ e a saída desejada y possui uma distribuição normal com média zero e variância σ_ε^2, e os pares de entrada $\{\mathbf{x}_i, y_i\}_{i=1}^N$ são assumidos como gerados independentemente, então a verossimilhança dos dados observados pode ser escrita como:

$$p(\mathbf{y} \mid \mathbf{w}, \sigma_\varepsilon^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} \exp\left\{-\frac{1}{2\sigma_\varepsilon^2} \left\| y_i - f(\mathbf{x}_i;\mathbf{w}) \right\|^2\right\} \tag{3.28}$$

em que

$$f(\mathbf{x};\mathbf{w}) = \sum_{i=0}^{N} w_i K(\mathbf{x}, \mathbf{x}_i) \tag{3.29}$$

onde $K(\mathbf{x}, \mathbf{x}_i)$ é uma função de kernel.

A RVM utiliza a probabilidade a priori sobre os pesos do modelo regido por um conjunto de hiper-parâmetros. Cada um desses hiper-parâmetros está associado com cada peso, e os valores mais prováveis são estimados de forma iterativa a partir dos dados de treinamento (TIPPING, 2000). Em uma perspectiva Bayesiana, os parâmetros do modelo **w** e σ_ε^2 podem ser estimados atribuindo-se primeiro uma distribuição a priori aos parâmetros e então estimando-se a distribuição posterior usando a verossimilhança dos dados observados. TIPPING (2000, 2001) propôs para cada parâmetro uma distribuição a priori da forma:

$$p(w_j \mid \alpha_j) = \frac{\sqrt{\alpha_j}}{\sqrt{2\pi}} \exp\left(-\frac{\alpha_j w_j^2}{2}\right) = N(0, \alpha_j^{-2}) \tag{3.30}$$

onde $j = 1, ..., N$ e $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)^T$ é o vetor de hiper-parâmetros da RVM, que devem ser estimados iterativamente dos dados.

Com a distribuição a priori, utiliza-se a regra de Bayes para determinar a distribuição posterior dos parâmetros do modelo:

$$p(\mathbf{w}, \alpha, \sigma_\varepsilon^2 \mid \mathbf{y}) = p(\mathbf{w} \mid \mathbf{y}, \alpha, \sigma_\varepsilon^2)\, p(\alpha, \sigma_\varepsilon^2 \mid \mathbf{y}) \tag{3.31}$$

Assim, dado um novo exemplo, x, as predições para o correspondente rótulo y em termos de:

$$p(f(\mathbf{x};\mathbf{w}) \mid \mathbf{y}) = \int p(f(\mathbf{x};\mathbf{w}) \mid \mathbf{w}, \boldsymbol{\alpha}, \sigma^2)\, p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2 \mid \mathbf{y})\, dw\, d\alpha\, d\sigma^2 \tag{3.32}$$

Entretanto, uma expressão analítica para a distribuição posterior dos parâmetros do modelo ainda não está disponível. Para resolver este problema é necessário adotar uma aproximação efetiva. A distribuição posterior dos parâmetros pode ser decomposta em dois componentes de acordo a equação 3.33 (TIPPING, 2001).

$$p(\mathbf{w},\alpha,\sigma_\varepsilon^2 \mid \mathbf{y}) = p(\mathbf{w} \mid \mathbf{y},\alpha,\sigma_\varepsilon^2)\, p(\alpha,\sigma_\varepsilon^2 \mid \mathbf{y}) \qquad 3.33$$

O primeiro termo do lado direito da equação 3.32 é a probabilidade posterior dos pesos w dados σ_ε^2 e α. O segundo termo do lado direito da equação 3.33 é a probabilidade posterior de α e σ_ε^2. O cálculo destas probabilidades é bem detalhado em TIPPING (2000, 2001).

Uma vez que os pesos foram obtidos, os hiper-parâmetros α_i são atualizados da seguinte forma $\alpha_i = \frac{\lambda_i}{w_i^2}$, onde w_i^2 é o i-ésimo peso médio, e λi é definido como $\lambda_i = 1 - \Sigma_{ii}$; Σ_{ii} é o *i*-ésimo elemento da diagonal da matriz de covariância e pode ser interpretado como a medida de quão bem estimado está cada parâmetro w_i. A otimização dos hiper-parâmetros continua até que um limiar pré-determinado seja atingido, ou até que um certo número de iterações seja realizado.

3.5 PROBLEMAS DE CLASSIFICAÇÃO COM MÚLTIPLAS CLASSES

Problemas de classificação são largamente estudados e aplicados em diferentes áreas de conhecimento. Existem diversas técnicas de aprendizado de máquina utilizadas na resolução de problemas de classificação, tais como as redes neurais artificiais, algoritmos de árvore de decisão, e máquinas de vetores de suporte. Uma tarefa muito comum em problemas de classificação é classificar um conjunto de dados que inclui mais que duas classes. Problemas de classificação deste tipo são chamados de "problemas de classificação com múltiplas classes".

SVMs foram inicialmente concebidas para realizar classificação binária (CORTES; VAPNIK, 1995), ou seja, classificar um elemento como pertence a uma classe "A" ou a uma classe "B". Várias técnicas vêm sendo pesquisadas com o objetivo de utilizar SVM em problemas de classificação que envolvam mais que duas classes (classificação de múltiplas classes). Atualmente, prevalecem duas técnicas; uma delas propõe utilizar vários

classificadores binários combinados, a outra propõe considerar todos os dados diretamente na formulação do problema.

A formulação que considera todos os dados diretamente para fazer a classificação de uma só vez (utilizando apenas uma SVM) conduz a um problema de otimização em altas dimensões, pois há um crescimento das variáveis proporcional ao número de classes envolvidas no problema. Em geral, o custo computacional é maior para solucionar um problema com várias classes do que para solucionar vários problemas de classificação binária. Os métodos a seguir são utilizados para resolver o problema de classificação de múltiplas classes como várias classificações binárias.

3.5.1 UM CONTRA TODOS (ONE VERSUS ALL)

O método um contra todos é simples e efetivo para a solução de problemas com múltiplas classes. Ele foi provavelmente a primeira implementação bem sucedida para o uso de SVM em problemas de classificação com várias classes. Este método constrói *NC* modelos SVM, onde *NC* é o número de classes. O *n*-ésimo classificador SVM é treinado com todos os exemplos da *n*-ésima classe com rótulos positivos, e todos os exemplos das outras classes com rótulos negativos.

Um novo exemplo **x** será apresentado para cada um dos *nc* classificadores. Na solução ideal, apenas um dos *NC* classificadores SVMs retornará um resultado positivo para a classificação do exemplo **x**, e os demais retornarão resultados negativos. Entretanto, na prática, várias das SVMs podem retornar valor positivo para a classificação de **x**, indicando que **x** poderia pertencer a várias classes. Esta classificação ambígua é conhecida como problema do falso positivo. Uma das razões para o problema do falso positivo é que a fronteira de decisão entre a classe verdadeira e as outras classes complementares combinadas não pode ser definida claramente. Uma das formas de resolver este problema é analisar o valor da saída das *NC* SVMs que produziram saída positiva. A classe corresponde à amostra **x** será aquela no qual a SVM produziu maior valor positivo. Esta estratégia é conhecida como *Win Max* (corresponde à saída da classe com maior valor), e foi a utilizada nesta dissertação.

Para resolver o problema do falso positivo, DING; DUBCHAK (2001) propuseram adicionar um segundo passo ao método original, onde é realizada uma classificação

discriminante sobre os pares de classificadores entre todas as classes com predição positiva. Este método é conhecido como “Um Contra Todos Único”.

Para um exemplo **x**, o método Um contra Todos pode apontar q classificações positivas, isto e, **x** pode pertencer a q classes, com q menor ou igual a NC. Se o número de classes é NC, então um passo adicional envolve o treinamento de classificadores para todos os $NC(NC - 1)/2$ pares de classes. Os $q(q - 1)/2$ classificadores que envolvem as q classes são então, aos pares, aplicados ao exemplo consultado **x**, e produzem uma predição positiva para uma classe particular. Todos os votos dos $q(q - 1) / 2$ classificadores são marcados e a classe que recebeu mais votos representa a predição final; eliminando assim os falsos positivos.

Na eliminação dos falsos positivos, a fronteira de decisão é desenhada entre duas classes de treinamento, e não entre uma classe (rotulada como positiva) e as outras $NC - 1$ classes (rotuladas como negativas). O passo de eliminação de falsos positivos é basicamente uma técnica de redução de ruído. Um problema com esta técnica é quando ocorre empate na votação, nesse caso é necessário algum procedimento de desempate.

A Ilustração 3.3 mostra um exemplo de separação de três classes A, B e C utilizando-se classificadores treinados utilizando o método Um Contra Todos. O primeiro classificador é treinado com os rótulos dos elementos da classe A iguais a 1 e os rótulos dos elementos das classes B e C iguais a -1; o segundo classificador recebe os rótulos dos elementos da classe B iguais a 1 e os rótulos dos elementos das classe A e C iguais a -1; o terceiro classificador recebe os rótulos da classe C iguais a 1 e os rótulos dos exemplos das classes A e B iguais a -1.

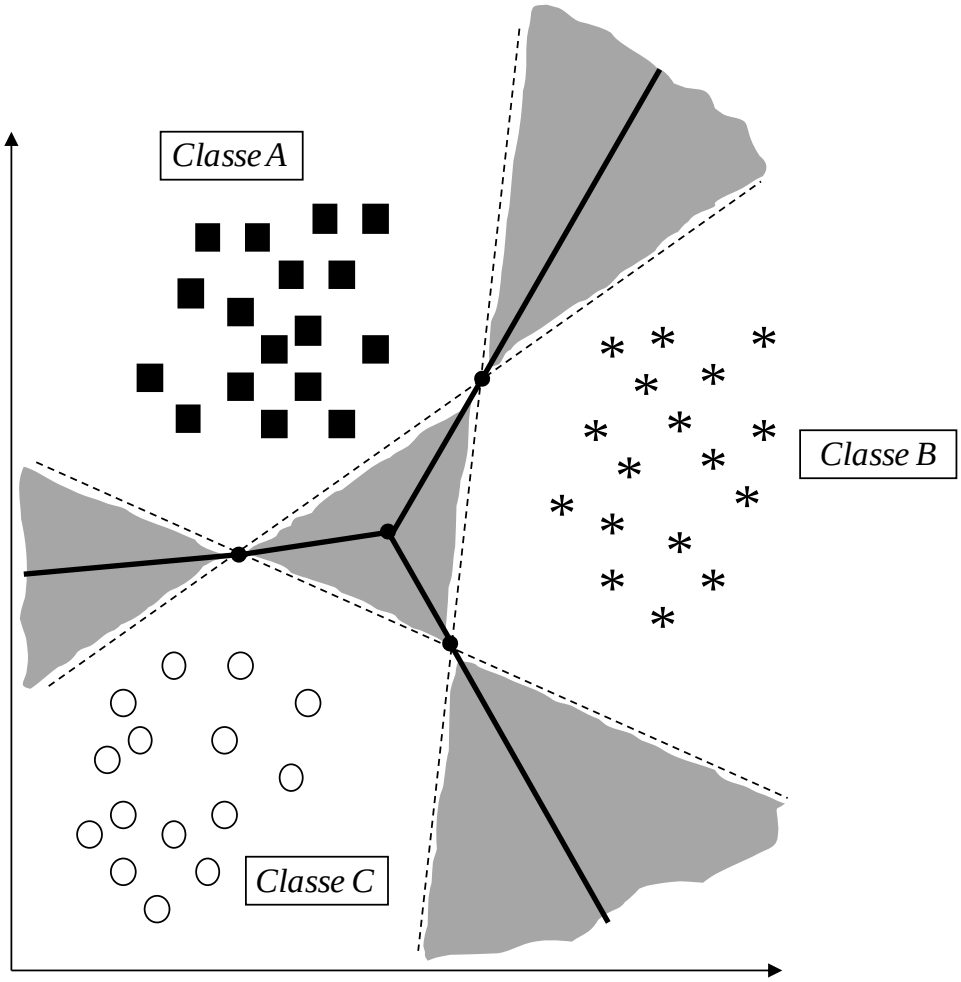


Ilustração 3.3 – Fronteira de decisão no Um Contra Todos.

3.5.2 UM CONTRA UM (ONE VERSUS ONE)

Este método foi introduzido por KNERR (1990), e os primeiros a utilizar esta estratégia para SVM foram FRIEDMAN (1996) e KREBEL (1999). Ele consiste em construir $NC(NC-1)/2$ classificadores, e treinar cada um destes sobre os dados de duas classes. Existem diversos procedimentos para combinar esses NC $(NC-1)/2$ classificadores a fim de atribuir um exemplo **x**, não pertencente aos dados de treinamento, a uma única classe. Um destes procedimentos sugere utilizar a estratégia da votação: se um classificador treinado sobre as classes A e B, rotula **x** como pertence à classe A, a classe recebe um voto; caso contrário, a classe B recebe um voto. No final deste processo a classe que receber mais votos será a

vencedora (voto majoritário), e **x** será rotulado como pertencente a esta classe. Nos casos em que ocorrer um empate no número de votos, uma técnica de desempate deverá ser utilizada.

A Ilustração 3.4 mostra um exemplo em que três classes A, B e C são separadas por três classificadores utilizando a estratégia Um Contra Um. Um dos classificadores é treinado com os exemplos da classe A com rótulos iguais a 1 e os exemplos da classe B com rótulos iguais a -1; outro classificador é treinado com os exemplos da classe A com rótulos iguais a -1 e os exemplos da classe C com rótulos iguais a 1; e o terceiro classificador treinado com os exemplos da classe B com rótulos iguais a 1 e os exemplos da classe C com rótulos iguais a -1.

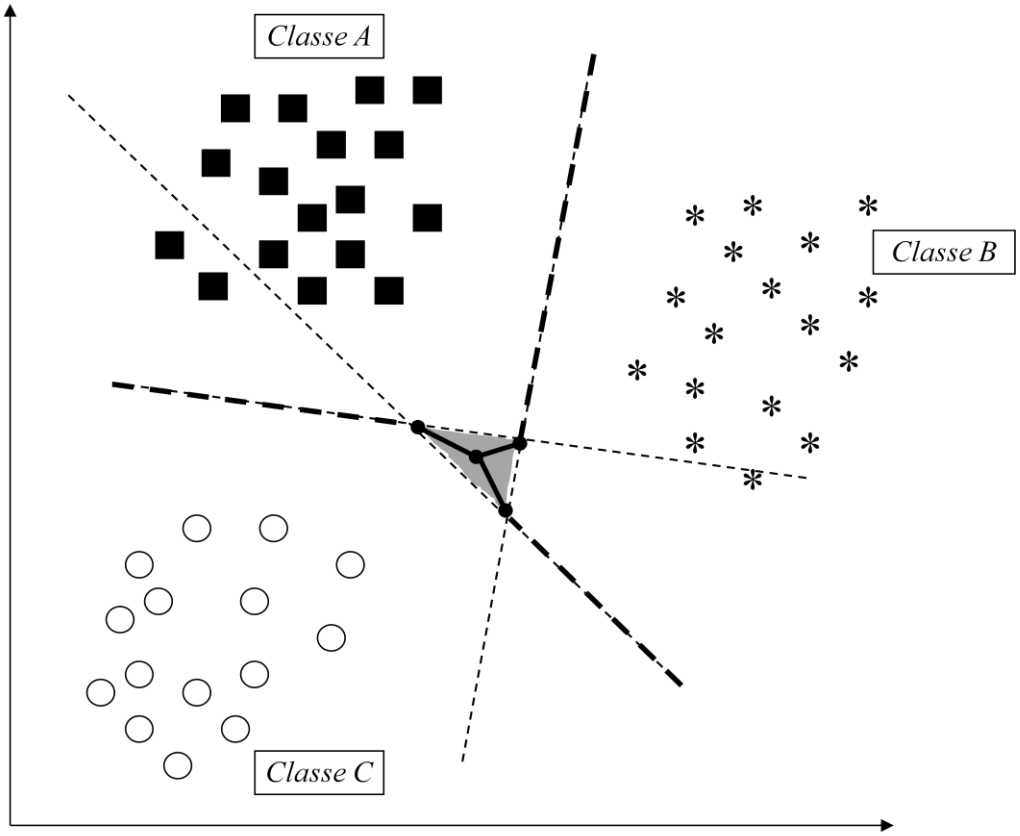


Ilustração 3.4 – Fronteira de decisão no Um Contra Um.

3.5.3 CODIFICAÇÃO POR CÓDIGOS DE CORREÇÃO DE ERROS (ERROR-CORRECTING OUTPUT CODE – ECOC)

ECOC é uma estratégia utilizada para resolver problemas de classificação com múltiplas classes utilizando classificadores binários. Esta estratégia tem sua origem associada à área de informação e engenharia de comunicação, e é comumente utilizada na solução de problemas de classificação com várias classes (DIETTERICH; BAKIRI, 1995). Este método converte um problema de classificação que envolve *NC* classes em *l* problemas de classificação binários.

Para isto, ele considera a utilização de uma matriz de códigos R com *NC* linhas e *l* colunas. Nesta matriz, as linhas representam as *NC* classes envolvidas no problema, e as colunas os *l* classificadores binários utilizados na solução do problema. Considere também que cada elemento R_{ij} da matriz R possa assumir um dos seguintes valores: 1 ou -1 (WIDJAJA *et al.*, 2003). Cada coluna da matriz R, R_j, contém as classificações feitas pelo classificador j para cada uma das *NC* classes. Assim, cada linha da matriz R, R_i, representa um código (*codeword*) associado a cada classe i. Este código é formado pelos resultados das classificações feitas por cada um dos classificadores para uma dada classe.

Dado um exemplo **x**, este é avaliado pelos *l* classificadores, e recebe dessa forma um código. Este código é então comparado com cada linha da matriz, ou seja, com o código de cada uma das *NC* classes; o exemplo receberá o rótulo da classe que tiver o código mais "similar" ao seu (ALLWEIN *et al.*, 2000). Esta similaridade é determinada por uma função. Nesta dissertação foi utilizada uma função que calcula a distância de Hamming.

A geração do código que representa cada classe, e que é também calculado para cada exemplo a ser classificado, é um dos maiores desafios para a utilização desta técnica na solução de problemas de classificação com várias classes (WIDJAJA *et al.*, 2003). Existem diferentes estratégias de codificação; os valores armazenados em R_{ij} podem ser, por exemplo, -1, 0 e 1. Diferentes estratégias de codificação levam a diferentes métodos para determinar a similaridade ou proximidade entre os códigos da matriz e os códigos avaliados para cada exemplo. Além disso, a forma de determinar os classificadores binários também pode afetar o desempenho deste método (ALLWEIN *et al.*, 2000).

A Ilustração 3.5 mostra um caso em que um exemplo **x** é rotulado como sendo da classe C, pois o código associado à classe C (-1 1 1) é o mesmo atribuído pelos três classificadores ao exemplo **x**.

	C1	C2	C3
A	1	-1	-1
B	-1	1	-1
C	-1	1	1

x	-1	1	1

Ilustração 3.5 – Matrizes de codificação de três classes A, B e C e três classificadores C1, C2 e C3.

3.5.4 CODIFICAÇÃO DE SAÍDA MÍNIMA (MINIMUM OUTPUT CODING – MOC)

A estratégia de utilizar SVM com MOC faz uso de saídas adicionais para codificar múltiplas classes. Um exemplo **x** é mapeado para um vetor $\mathbf{y}_n$, que é a combinação das n saídas geradas pelos classificadores SVM. Para efeito de exemplo, considere um problema hipotético que envolva a classificação de quatro classes distintas A, B, C e D. Elas poderiam ser mapeadas para os seguintes vetores {1, -1, -1, -1}, {-1, 1, -1, -1}, {-1, -1, 1, -1} e {-1, -1, -1, 1} respectivamente. Neste caso, os resultados das classificações de quatro SVMs foram combinados de maneira a gerar esses vetores de saída. Essa estratégia é muito próxima à clássica técnica de codificação de saídas de redes neurais (SUYKENS; VANDEWALLE, 1999).

As *NC* classes podem ser codificadas utilizando $\log_2 NC$ saídas de SVMs, ou *l* saídas de SVMs podem codificar 2^l classes. Quando esse tipo de codificação utiliza o menor número de saídas *l* possível, ele é chamado de codificação de saída mínima (VAN GESTEL *et al.*, 2004).

Considerando o exemplo anterior novamente, onde temos $NC = 4$ classes (A, B, C e D), para que a saída codificada seja mínima devemos utilizar $l = \log_2 NC = \log_2 4 = 2$ SVMs

para gerar as saídas. Assim teríamos os classificadores SVM C1 e C2 treinados da seguinte forma:

- C1 – atribui -1 para as classes A e B, e atribui 1 para as classes C e D.
- C2 – atribui -1 para as classes A e C, e atribui 1 para as classes B e D.

As saídas resultantes das classificações realizadas por estes classificadores são combinadas da seguinte forma:

- Classe A, se C1 = -1 e C2 = -1.
- Classe B, se C1 = -1 e C2 = 1.
- Classe C, se C1 = 1 e C2 = -1.
- Classe D, se C1 = 1 e C2 = 1.

Neste caso, os vetores de saída correspondentes às classes A, B, C e D seriam respectivamente: {-1, -1}, {-1, 1}, {1, -1} e {1, 1}.

O MOC reduz o número de classificadores binários utilizados. Porém gera uma codificação mais sujeita a cometer erros, pois se uma das saídas for trocada o resultado será totalmente diferente.

CAPÍTULO 4 - RESULTADOS EXPERIMENTAIS

RESUMO: Neste capítulo são descritos os experimentos realizados para a classificação de sinais de EEG utilizando classificadores baseados em Máquinas de Vetores de Suporte e suas variantes. Foram realizados cinco experimentos com objetivos distintos. Os resultados obtidos foram tabelados e analisados em termos da taxa de erro, matriz de confusão e curvas ROC.

4.1 INTRODUÇÃO

Para a realização dos experimentos, foram realizadas simulações utilizando um conjunto de séries temporais disponível em uma base de dados de domínio público descrita em ELGER *et al.*(2001). Em todos os experimentos realizados, o parâmetro C foi fixado em 100, os kernels empregados foram RBF e ERBF e seus parâmetros foram variados no intervalo 2^i $i \in$ [-10:15]. Para cada valor do parâmetro do kernel foi treinada uma SVM com validação cruzada (10 *folds*).

4.2 DESCRIÇÃO DOS DADOS

O conjunto de dados utilizados nesta dissertação consiste de sinais de EEG agrupados em cinco subconjuntos nomeados de A até E, cada um contendo 100 segmentos de amostras de sinais de EEG. Estes segmentos foram selecionados a partir de gravações de EEG após uma inspeção visual por especialistas. Os conjuntos A e B consistem de segmentos obtidos de gravações de EEG a partir de eletrodos superficiais realizadas em cinco voluntários saudáveis, usando um esquema de colocação de eletrodos padronizado chamado Sistema Internacional 10-20 de colocação de eletrodos (do inglês *The International 10-20 System*).

Este sistema de colocação de eletrodos foi proposto por JASPER (1958). Ele é baseado na relação entre a localização de um eletrodo e a área encoberta do córtex cerebral. Cada posição possui uma letra para identificar o lobo e um número ou outra letra para identificar a

localização no hemisfério. A Ilustração 4.1 mostra o esquema de colocação de eletrodos do Sistema Internacional 10-20.

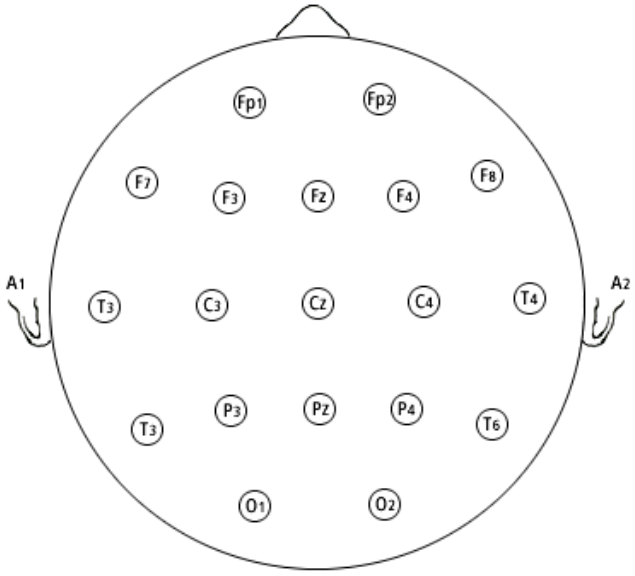


Ilustração 4.1 – O sistema internacional 10-20 de colocação de eletrodos.

As letras utilizadas na ilustração denotam:

- "F" – lobo frontal
- "T" – lobo temporal
- "C" – central
- "P" – lobo parietal
- "O" – lobo occipital

A letra "C" é usada para identificar a região central. Os números pares (2, 4, 6, 8) referem-se ao hemisfério direito e os números ímpares (1, 3, 5, 7) referem-se ao hemisfério esquerdo. A letra "z" refere-se a um eletrodo colocado na linha média (entre os hemisférios). Quanto menor o número, mais próximo ele se localiza da linha média. "Fp" corresponde à fronte polar.

Os números 10 e 20 do nome do sistema referem-se a 10% e 20% na distância de colocação dos eletrodos. As medidas no sentido ântero-posterior do escalpo são baseadas na distância entre o násio e o ínio (Ilustração 4.2 A), passando pelo vértex na linha média. Cinco pontos são distribuídos nesta linha: Fp, F, C, P e O. O ponto Fp está situado a 10% da distância entre o násio e o ínio, logo acima do násio. Entre os demais pontos, a distância é de 20% da distância total ente o násio e o ínio. As medidas laterais são distribuídas de acordo

com a Ilustração 4.2 B. E a distribuição sobre as regiões frontal, temporal e occipital é mostrada na Ilustração 4.2 C (FLEURY, 2007).

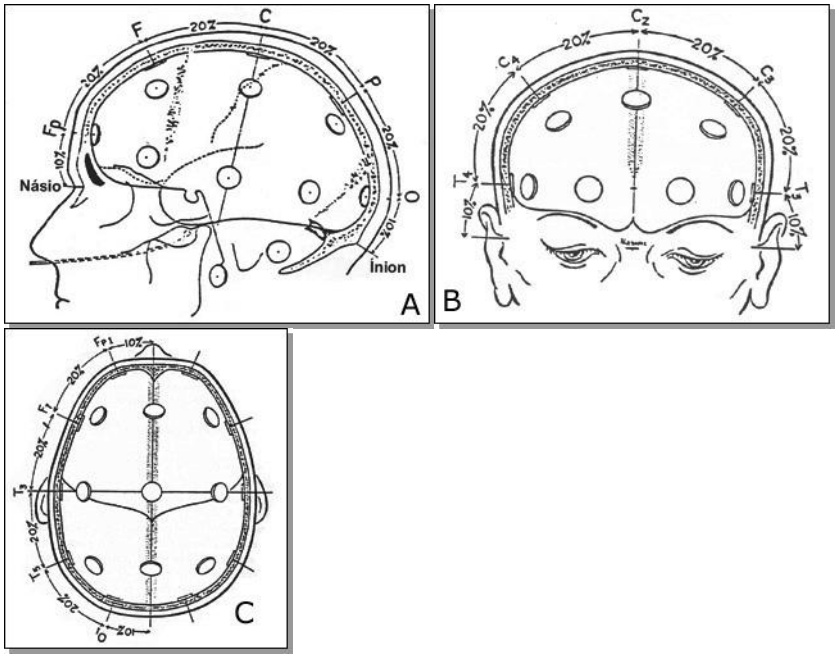


Ilustração 4.2 – Método de medida para o encontro das posições dos eletrodos mo sistema 10-20.

4.3 METODOLOGIA UTILIZADA NA OBTENÇÃO DOS DADOS

Para a realização dos exames de EEG, os voluntários estavam relaxados e com os olhos abertos (em A) e com olhos fechados (em B) respectivamente. Os conjuntos C, D e E originaram-se de arquivos de EEG de diagnóstico pré-cirúrgico. EEGs de cinco pacientes foram selecionados, todos aqueles que conseguiram o controle dos episódios de epilepsia após a remoção de uma das formações do hipocampo, a qual foi corretamente diagnosticada como zona epileptogênica. Segmentos no conjunto D foram gravados a partir de dentro da zona epileptogênica, e aquelas no conjunto C da formação hipocampal do hemisfério oposto do cérebro. Enquanto os conjuntos C e D continham apenas atividades medidas durante intervalos livres de anomalia, o conjunto E só continha registros com anomalias. Todos os sinais EEG foram registrados com o mesmo sistema de amplificador com 128 canais, utilizando uma referência comum. Os dados foram digitalizados a 173,61 amostras por segundo usando resolução de 12 bits. A configuração do filtro passa banda foi: 0.53-40 Hz (12dB/oct) (SUBASI, 2007; REVETT *et al.*, 2006).

A Ilustração 4.3 mostra exemplos de elementos destes conjuntos (um elemento de cada um dos conjuntos A, B, C, D e E foi escolhido de forma aleatória para essa ilustração). Para maior detalhamento da obtenção dos dados de EEG consulte ELGER *et al.* (2001).

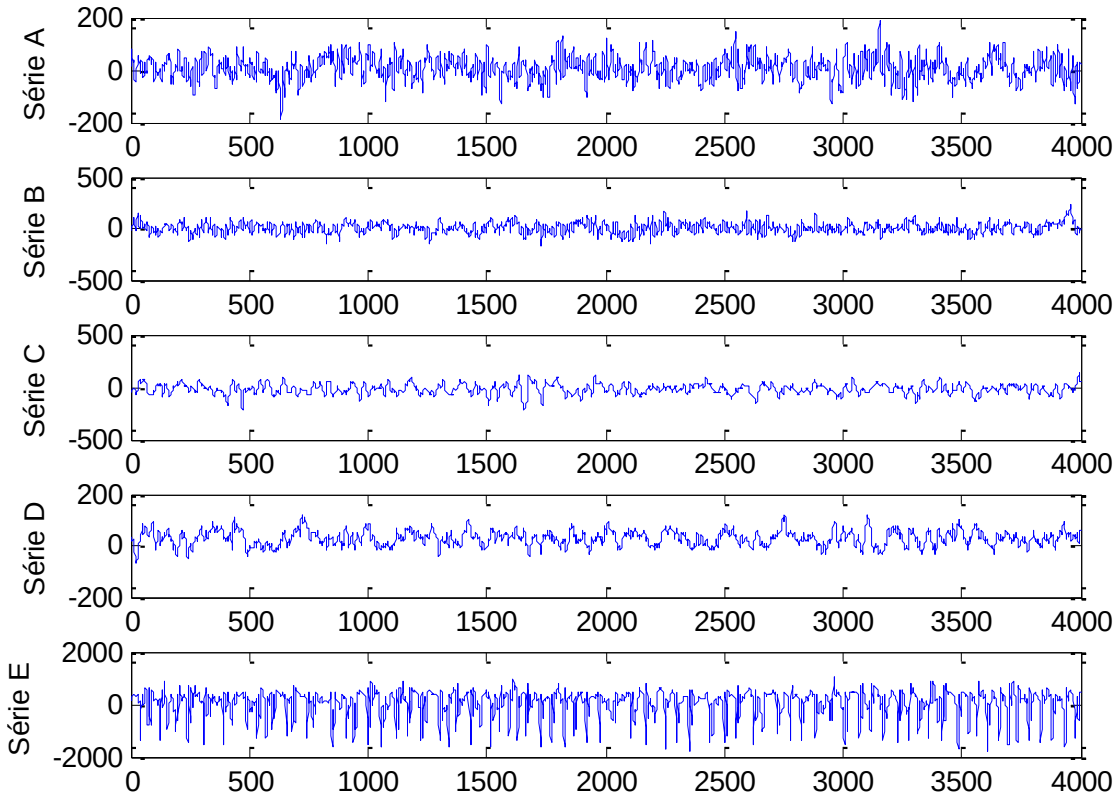


Ilustração 4.3 – Exemplos de elementos dos conjuntos de EEG A, B, C, D e E.

Sobre esse conjunto de dados, foram calculados os coeficientes wavelet que é uma das técnicas mais utilizadas no que diz respeito ao processamento de sinais de EEG. Com a finalidade de reduzir a dimensionalidade dos dados a serem analisados, é comum calcular medidas estatísticas em cima dos coeficientes wavelets resultantes (SUBASI, 2007; ÜBEYLI; GÜLER, 2005; ÜBEYLI, 2008; REVETT *et al.*, 2006). Essas medidas geram novos vetores de características que representam de forma alternativa a informação codificada nos coeficientes wavelet (e no sinal original por conseqüência).

O sinal de EEG foi decomposto utilizando a transformada wavelet, como descrito no capítulo 2 em cinco níveis de detalhe D1, D2, D3, D4 e D5 e uma aproximação A5. Quando não mencionado explicitamente, o tipo de wavelet utilizado foi a Daubechie de ordem quatro, pois de acordo com SUBASI (2007) sua característica de suavização a torna mais apropriada para detectar mudanças em sinais de EEG.

A partir destes dados, os vetores de características listados a seguir foram gerados e então utilizados como entrada nos experimentos de classificação que serão descritos neste capítulo:

- Média dos coeficientes wavelet (MeanW)
- Desvio Padrão dos coeficientes wavelet (StdW)
- Máximo dos coeficientes wavelet (MaxW)
- Mínimo dos coeficientes wavelet (MinW)
- Média das potências dos coeficientes wavelets em cada sub-banda (MSqW)
- Média dos valores absolutos dos coeficientes wavelets em cada sub-banda (MAbsW)
- Taxa entre as médias absolutas de cada sub-banda adjacente (MSbW).
- Todas as estatísticas anteriores (AllW)

Para a geração destes vetores com estatísticas sobre os coeficientes wavelet, foram implementadas funções para realizar os cálculos a seguir:

1. Média: para cada um dos níveis de detalhe e para a aproximação (D1, D2, D3, D4, D5 e A5) foi calculada a média. Isso resultou em um vetor de característica composto por seis médias: [mediaD1, mediaD2, mediaD3, mediaD4, mediaD5 e mediaA5].
2. Desvio-padrão: de maneira equivalente à média foi obtido um vetor de características correspondente ao desvio padrão em cada nível de decomposição.
3. Máximo e Mínimo: análogo ao calculo da média e do desvio padrão.
4. Médias das potências: para cada um dos detalhes D1, D2, D3, D4, D5 e para A5 foi calculada a soma dos quadrados dos coeficientes (e.g soma($D(i)^2$)) e o resultado foi dividido pelo número de elementos em cada um dos níveis.
5. Taxa entre as médias absoluta de cada sub-banda adjacente: este cálculo foi feito da seguinte forma: valor absoluto da média em D(i) dividido pelo valor absoluto da média em D(i+1), onde i varia de 1 até 5.

Além das características mencionadas anteriormente, considerou-se ainda um vetor formado pelos dados originais, isto é, o sinal não processado, que compreende um vetor de 4097 posições para cada segmento. Esta característica será importante para avaliarmos o desempenho da classificação utilizando SVM nos dados originais e também nos dados codificados.

4.4 CRITÉRIOS DE AVALIAÇÃO DO CLASSIFICADOR

A seguir são descritos os principais critérios de avaliação de um classificador adotado nesta dissertação.

4.4.1 MATRIZ DE CONFUSÃO

Uma maneira natural de apresentar as estatísticas para a avaliação de um modelo de classificação é por meio de uma tabulação cruzada entre a classe prevista pelo modelo e a classe real dos exemplos. Esta tabulação é conhecida como matriz de confusão (também chamada de tabela de contingência).

Considerando problemas de classificação binária, ou seja, problemas de classificação com apenas duas classes (geralmente rotuladas como "+" e "-"), a matriz de confusão é utilizada para análise de previsões corretas e errôneas, conforme mostra a Tabela 4.1.

Tabela 4.1 – Matriz de confusão

		Classe prevista pelo método	
		C+	**C-**
Classe real	**C+**	Soma de classificações Verdadeiras Positivas (TP)	Soma de classificações Falsas Positivas (FN)
	C-	Soma de classificações Falsas Positivas (FP)	Soma de classificações Verdadeiras Negativas (TN)

- **Verdadeiro Positivo (TP)**: o exemplo pertence à classe C+ e o classificador prediz que o exemplo pertence à classe C+;
- **Verdadeiro Negativo (TN)**: o exemplo pertence à classe C- e o classificador prediz que o exemplo pertence à classe C-;
- **Falso Positivo (FP):** o exemplo pertence à classe C- e o classificador prediz que o exemplo pertence à classe C+;
- **Falso Negativo (FN)**: o exemplo pertence à classe C+ e o classificador prediz que o exemplo pertence à classe C-;

Baseada na matriz de confusão pode-se definir as seguintes medidas de avaliação: precisão ou acuracidade (Acc), erro (Err), sensibilidade (Sens) e especificidade (Spec). Seguem abaixo as respectivas equações:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad 4.1$$

$$Err = 1 - Acc \quad 4.2$$

$$Sens = \frac{TP}{TP + FN} \quad 4.3$$

$$Spec = \frac{TN}{TP + FP} \quad 4.4$$

4.4.2 ANÁLISE ROC

Análise ROC (do inglês *Receiver Operating Characteristic*) é um método gráfico para avaliação, organização e seleção de sistemas de diagnóstico e/ou predição. Para realizar estas análises, gráficos ROC podem mostrar o limiar entre taxas de verdadeiros positivos e falsos negativos dos classificadores. Gráficos ROC foram originalmente utilizados em detecção de sinais, para se avaliar a qualidade de transmissão de um sinal em um canal com ruído (EGAN, 1975).

SPACKMAN (1989) foi um dos primeiros a empregar gráficos ROC em aprendizado de máquina. Ele que demonstrou a utilidade das curvas ROC na avaliação e comparação entre algoritmos. A análise ROC também tem sido utilizada para visualizar e analisar o comportamento de diagnóstico (SWETS, 1988), principalmente na medicina. Atualmente há um grande crescimento no uso de gráficos ROC na comunidade de aprendizado de máquina, em parte pela técnica na análise de classificadores.

Os gráficos ROC são bidimensionais, onde no eixo Y plota-se o valor de TP e no eixo X o valor de FP. A Ilustração 4.4 mostra um gráfico ROC simples, somente para classificadores discretos. Classificadores discretos são aqueles que geram como saída somente uma classe. Estes classificadores fornecem um par (FP, TP), correspondendo a um ponto no espaço ROC.

Muitos pontos são importantes no espaço ROC, veja a Ilustração 4.4. O ponto inferior esquerdo (0,0) representa uma estratégia que nunca gera uma classificação positiva, como um classificador que não comete erros de falso positivo, mas também não classifica nenhum

verdadeiro positivo. A estratégia oposta, de incondicionalmente gerar verdadeiros positivos é representada pelo ponto superior direito (1,1). O ponto (0,1) representa uma classificação perfeita. Este é o caso do classificador D. De maneira geral, um ponto, no espaço ROC, é melhor que outro se ele está mais a noroeste (TP é maior e FP é menor). Classificadores no lado esquerdo do gráfico ROC (perto do eixo Y) são ditos conservadores, pois fazem classificações positivas somente com uma evidência forte, portanto comentem poucos erros do tipo falso positivo. Classificadores no lado direito são ditos liberais, pois fazem classificações positivas com pouca evidência, mas comentem muitos erros de falso positivo. Na Ilustração 4.4, A é mais conservador que B. Qualquer classificador que esteja abaixo da linha diagonal que vai de do ponto (0,0) ao ponto (1,1) é dito pior que um classificador aleatório. Por isso, o triângulo inferior de um gráfico ROC está geralmente vazio. Pode-se ressaltar que se um classificador produz pontos abaixo da diagonal, pode-se negá-lo para produzir pontos acima desta. Na Ilustração 4.4, o ponto B é igual a E negado.

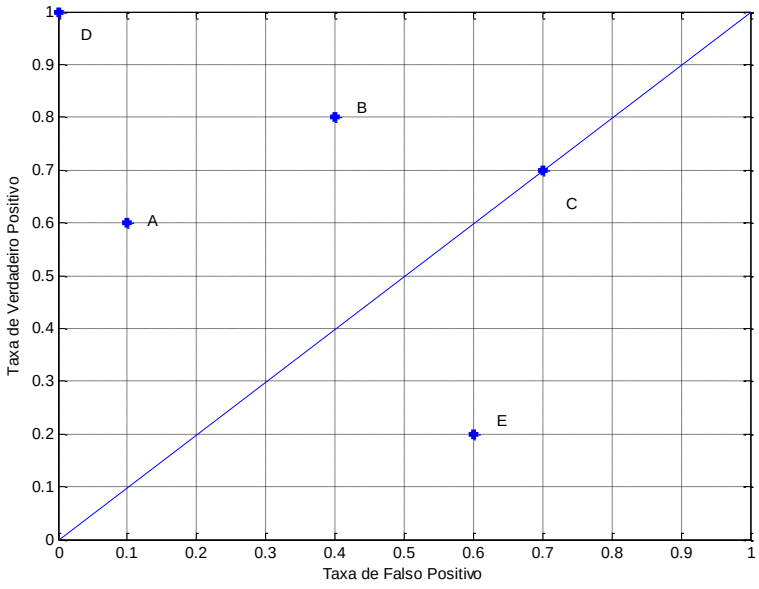


Ilustração 4.4 – Gráfico ROC mostrando 5 classificadores discretos. A é dito um classificador “conservador”, B é o inverso de E, D é um classificador perfeito e C é dito aleatório.

Os gráficos de ROC desta dissertação foram gerados da seguinte forma: suponha que as saídas de um classificador sejam os valores na primeira coluna da Tabela 4.2. Considere também que os rótulos corretos dos exemplos sejam dados pela coluna y. A estratégia utilizada foi variar o limiar de decisão do classificador, adotando cada uma das saídas como um limiar (representados nas colunas L(-0,7) até L(>1,0). Assim, a coluna referente a cada limiar é preenchida com o rótulo atribuído pelo classificador utilizando o limiar em questão.

Tabela 4.2 – Variação do limiar do classificador para gráfico ROC.

f(x;w)	**y**	**L (-0,7)**	**L(-0,6)**	**L(0,8)**	**L(0,9)**	**L(1,0)**	**L(>1,0)**
-0,7	-1	1	-1	-1	-1	-1	-1
-0,6	-1	1	1	-1	-1	-1	-1
0,8	1	1	1	1	-1	-1	-1
0,9	1	1	1	1	1	-1	-1
1,0	1	1	1	1	1	1	-1

Com base nisso foram calculados os valores percentuais de verdadeiros positivos e falsos positivos:

- Limiar -0,7: TP = 1 e FP =1, gerando o ponto (1; 1).
- Limiar -0,6: TP = 1 e FP = 0,5 gerando o ponto (0,5; 1).
- Limiar 0,8: TP = 1 e FP = 0 gerando o ponto (0; 1).
- Limiar 0,9: TP = 0,66 e FP = 0 gerando o ponto (0; 0,66).
- Limiar 1,0: TP = 0,33 e FP = 0 gerando o ponto (0; 0,33).

A partir destes pontos, pode ser traçada uma curva para o gráfico ROC do classificador em questão.

4.4.3 ÁREA ABAIXO DA CURVA ROC

Uma curva ROC é uma demonstração bidimensional do desempenho de um classificador. Para comparar classificadores é preciso reduzir a curva ROC a um valor escalar. Um método comum para realiza esta redução é calcular a área abaixo da curva ROC (AUC) Como a AUC é uma porção da área do quadrado unitário, seus valores vão de 0.0 à 1.0. Entretanto, como classificadores piores que os aleatórios não são encontrados no espaço ROC, não existem classificadores com AUC menor que 0.5 (0.5 é a área de um classificador aleatório).

4.5 RESULTADOS COMPUTACIONAIS

Antes de descrever os experimentos realizados é necessário analisar alguns tópicos importantes, tais como normalização e o critério de particionamento dos dados a ser adotado. Muitas vezes os dados apresentam escalas de medidas distintas, podendo possuir menor ou maior variância. Os atributos com menor variância podem indicar que uma pequena variação

deste pode ser mais relevante que outro com maior variância. Assim, para que se possa avaliar adequadamente a relevância de cada um, é interessante que todos tenham variância pequena. Os dados normalizados não perdem o significado da informação após a normalização, são apenas convertidos em uma nova escala de valores. A normalização basicamente faz uma projeção dos dados não somente para ajustar as regiões de contorno, como também para que cada atributo contribua de forma igualitária na avaliação de classificação. No entanto, dependendo do tipo de dado, a normalização pode distorcer as informações contidas na série. O experimento # 1 ilustra esta situação, neste a energia ou potência do sinal é utilizado como critério de classificação. Quando inserimos uma normalização nas séries, o desempenho de classificação sofre uma leve mudança. Isto pode ser justificado pelo fato de que o critério de decisão, no caso energia ou potência, são dependentes da amplitude da série. Qualquer modificação nesta irá influenciar no valor da energia ou potência do sinal. Investigações quanto à viabilidade de normalização das séries podem ser efetuadas com fins de comparação do comportamento das series, servindo mais como ferramenta de análise. As séries podem ser normalizadas segundo a amplitude ou segundo a distribuição (HAYKIN, 2004). Neste trabalho, adotou-se a normalização por distribuição, onde cada elemento da série ou atributo foi normalizado de forma a possuir média zero e desvio padrão unitário.

Outro fator importante diz respeito ao critério de particionamento do conjunto de dados a ser adotado. Geralmente, o conjunto de dados é dividido em três subconjuntos, a saber, treinamento, validação e teste. O subconjunto de treinamento é utilizado no treinamento do classificador, isto é, no ajuste de seus parâmetros, já o subconjunto de validação na seleção do melhor classificador ajustado e o subconjunto de teste serve para medir o desempenho do classificador escolhido. No entanto, em muitos problemas de classificação, o numero de pontos de dados disponíveis é muito pequeno (por exemplo, abaixo de 100 amostras). Caso adotássemos este esquema de divisão dos dados, as amostras podem não ser representativas, produzindo um classificador com desempenho polarizado. Para evitar este problema, uma estratégia seria dividir o conjunto de dados em k conjunto de mesma cardinalidade. Cada subconjunto é usado como teste e o restante como treinamento. Além disso, estes subconjuntos podem ser estratificados, ou seja, em cada conjunto as classes são representadas com aproximadamente a mesma proporção tanto no teste como no treinamento. A taxa de erro global é calculada pela média das taxas de erro calculadas em cada etapa. Esta estratégia é conhecida como validação cruzada k-fold, e com k normalmente variando entre 5 a 10. Neste trabalho, será adotada validação cruzada estratificada 10-*fold*.

A seguir descrevemos os experimentos realizados. Inicialmente, investigaremos o comportamento de modelos bastante simples baseados na energia e potência do sinal. Posteriormente, mostraremos os resultados envolvendo Máquinas de Vetores Suporte empregando coeficientes extraídos via transformada wavelet.

4.5.1 EXPERIMENTO # 1

O objetivo deste experimento é verificar se podemos classificar as séries A, B, C, D, E apenas através da potência ou energia para cada uma das 100 séries em cada um dos subconjuntos. Inicialmente, nenhum pré-processamento será realizado. Posteriormente, investigaremos o efeito da normalização no desempenho de classificação. A Ilustração 4.5 e 4.6 apresentam o gráfico resultante da potência e energia, respectivamente, para cada subconjunto. Já a Ilustração 4.7 e Ilustração 4.8 apresentam o gráfico da potência e energia, respectivamente, para cada subconjunto normalizado com média zero e desvio padrão unitário.

Inicialmente vamos supor que o problema a ser tratado tenha apenas duas classes {+1, - 1}. Baseado no conjunto de treinamento é calculado a energia ou potência média da série para cada rótulo. Assumindo que a energia ou potência tenha uma distribuição normal, de acordo com LATHI, DING (2009), o limiar de separação ótimo (L) é dado por:

$$L = \frac{\sigma_{-1}(z)\mu_{+1}(z) + \sigma_{+1}(z)\mu_{-1}(z)}{\sigma_{+1}(z) + \sigma_{-1}(z)} \qquad 5.5$$

onde z representa a energia ou potência das séries utilizada no treinamento, σ_i(z), $\mu_i(z)$ corresponde a média e o desvio padrão respectivamente para a classe i.

Logo o classificador binário pode ser definido como:

$$f(\mathrm{z}') = \begin{cases} 1, & \text{se} \quad \mathrm{z}' > L \\ -1, & \text{se} \quad \mathrm{z}' < L \end{cases} \qquad 5.6$$

onde z*'* representa a energia ou potência da série a ser classificada e L o limiar de separação definido na equação 5.5. O classificador binário definido na equação 5.6 pode ser estendido para classificação com múltiplas classes usando as abordagens descritas no capítulo 3.

	Potência	Energia
A versus B	0,240 ± 0,0305	0,240 ± 0,0305
A versus C	0,325 ± 0,0490	0,325 ± 0,0490
A versus D	0,365 ± 0,0428	0,365 ± 0.0428
A versus E	0,060 ± 0,0124	0,060 ± 0.0124

Tabela 4.3 – Taxa de erro obtida empregando energia ou potência para classificação binária sem normalização.

O resultado apresentado na Tabela 4.3 corresponde à media e ± um desvio padrão da taxa de erro alcançado pelo classificador nos 10 *folds*. Analisando a Tabela 4.3, podemos observar que o emprego da energia ou potência como característica para o classificador linear, definido em 5.6, produziu bons resultados somente quando as classes a serem separadas eram A e E. Para as outras combinações de classes, o desempenho foi bastante ruim, por exemplo, para a separação das classes A e D a taxa de erro foi de 0,365 e o desvio padrão 0,0428. Vale ressaltar também, que como o calculo da energia e da potencia difere apenas por um fator (número de pontos das classes) era de se esperar que tivéssemos o mesmo desempenho para ambos, conforme verificado.

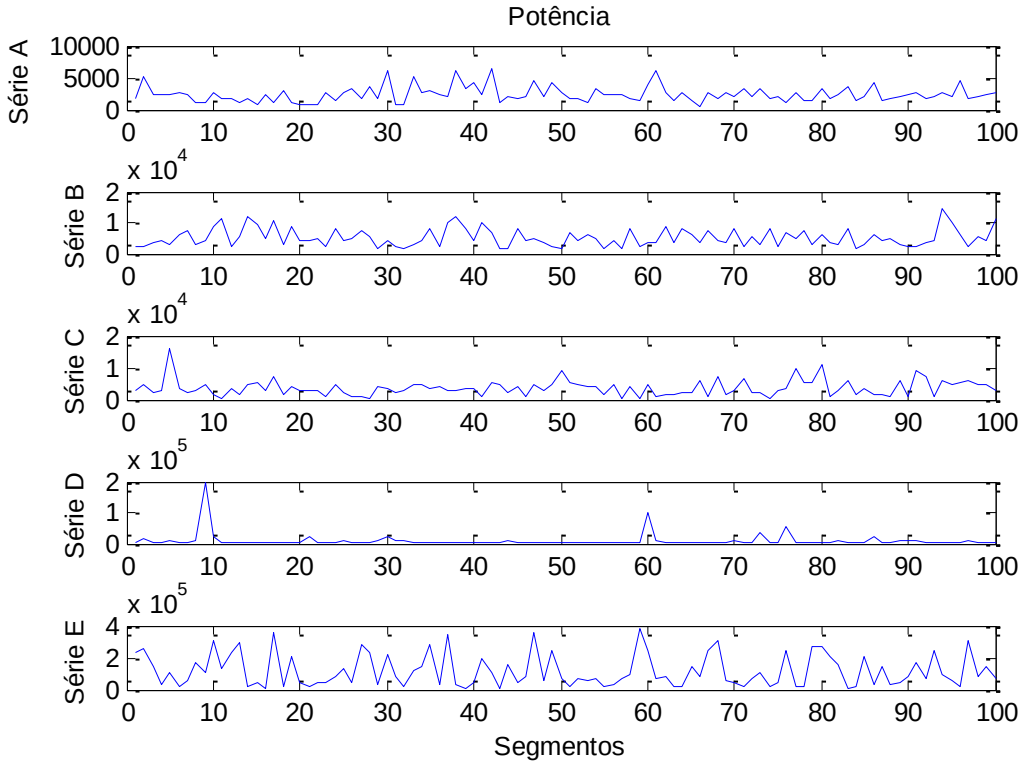


Ilustração 4.5 – Gráfico com a potência dos segmentos A, B, C, D, E não normalizados.

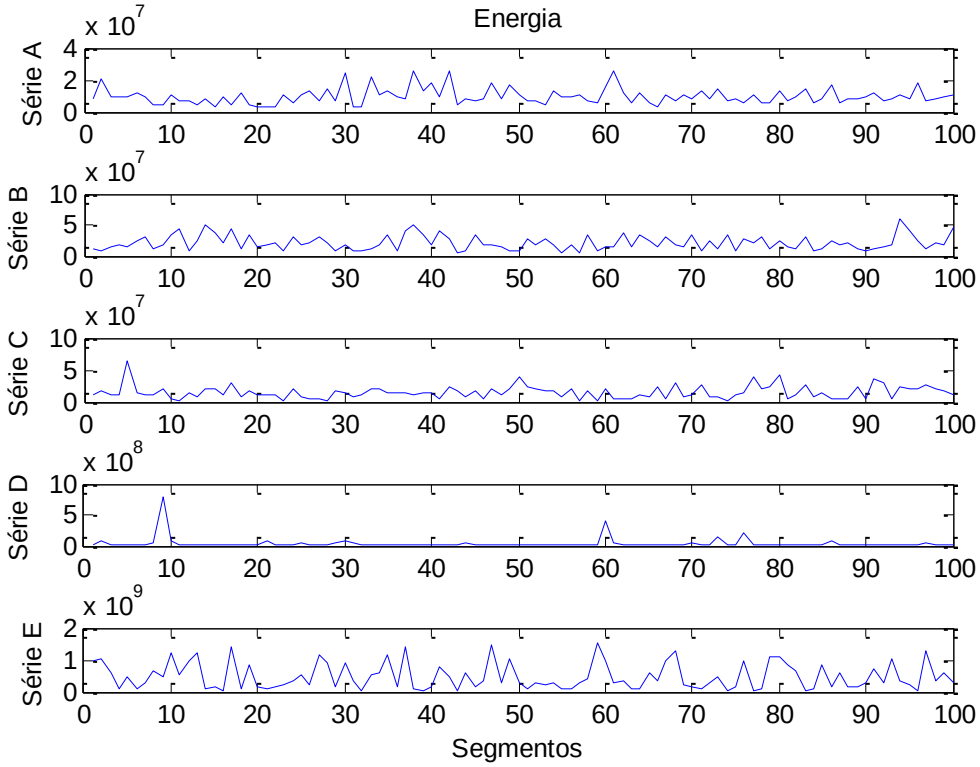


Ilustração 4.6 – Gráfico com a energia dos segmentos A, B, C, D, E não normalizados.

Conforme já mencionado, podemos estender o classificador binário, definido na equação 4.6, para tratar com problemas com múltiplas classes. Para isto teremos que aplicar os métodos de combinação de classificador binário abordados no capítulo 3. A Tabela 4.4 apresenta os resultados obtidos.

	Potência	Energia
Um contra um	**0,552 ± 0,0221**	**0,552 ± 0,0221**
Um contra todos	0,634 ± 0,0200	0,634 ± 0,0200
MOC	0,642 ± 0,0239	0,642 ± 0,0239
ECOC	0,668 ± 0,0350	0,668 ± 0,0350

Tabela 4.4 – Taxa de erro obtida empregando energia ou potência para classificação com multiplas classes (A, B, C, D, E), sem normalização

Analisando a Tabela 4.4, pode-se observar que o desempenho foi bastante ruim, com uma leve melhora quando é empregada a estratégia um contra um. Este resultado ruim já era esperado, pois analisando a Ilustração 4.7 e 4.8 nota-se que a potência ou energia são bastante distintas para as 100 séries da classe A e E, mas não podemos dizer o mesmo quando todas as classes são levadas em consideração.

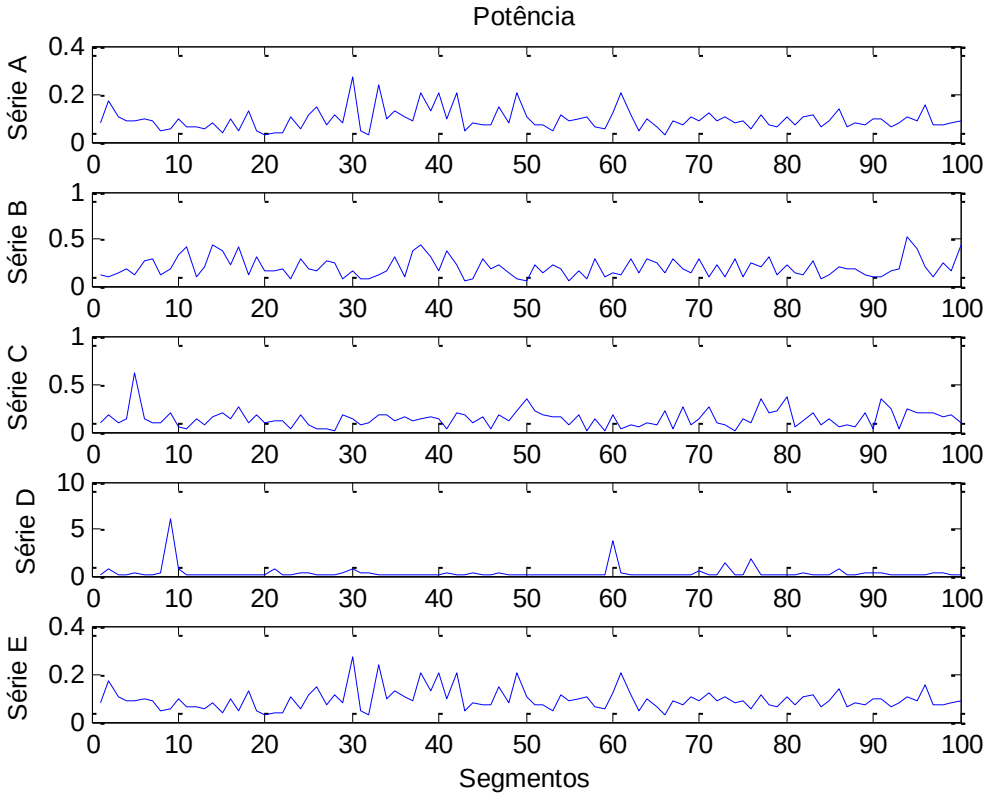


Ilustração 4.7 – Gráfico com a potência dos segmentos A, B, C, D, E normalizados.

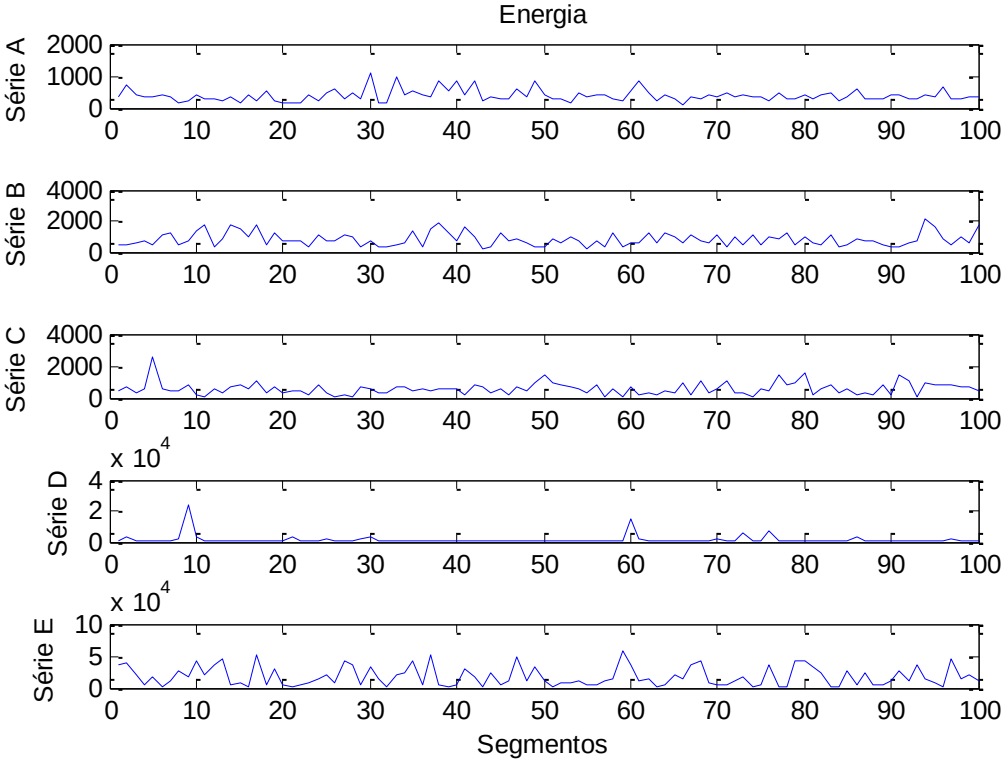


Ilustração 4.8 – Gráfico com a energia dos segmentos A, B, C, D e E normalizados.

A Tabela 4.5 apresenta o resultado de classificação empregando energia ou potência como definido na equação 4.6, quando os segmentos são normalizados. Comparando a Tabela 4.3 e 4.5, podemos observar uma pequena mudança no desempenho de classificação. Na classificação A versus B e A versus E houve uma melhora no desempenho. Já na classificação A versus C e A versus D houve uma piora no desempenho. Conforme já mencionado, está alteração deve-se ao fato que a energia e potência dependem da amplitude do sinal e qualquer alteração produzirá uma mudança no limiar de decisão.

	Potência	Energia
A versus B	0,230 ± 0,0270	0,230 ± 0,0270
A versus C	0,345 ± 0,0524	0,345 ± 0,0524
A versus D	0,360 ± 0,0413	0,360 ± 0,0413
A versus E	0,055 ± 0,0089	0,055 ± 0,0089

Tabela 4.5 – Taxa de erro obtida empregando energia ou potência para classificação binária.com normalização.

A Tabela 4.6 apresenta o resultado obtido, quando os segmentos são normalizados, para classificação com múltiplas classes, considerando as estratégias de combinação de classificador binário mencionadas no capítulo 3. Comparando a Tabela 4.4 e 4.6, pode-se observar que a estratégia um contra um produziu melhor resultado quando comparado com as outras. Em termos de desempenho, somente a estratégia um contra todos sofreu uma leve melhora no desempenho quando os dados foram normalizados. No entanto, o resultado de forma geral ainda é bastante ruim. Nos próximos experimentos, mostraremos que o emprego Máquinas de Vetores Suporte junto com transformada wavelet produziu resultados mais promissores.

	Potência	Energia
Um contra um	**0.564 ± 0.0208**	**0.564 ± 0.0208**
Um contra todos	0.630 ± 0.0200	0.630 ± 0.0200
MOC	0.634 ± 0.0208	0.634 ± 0.0208
ECOC	0.676 ± 0.0342	0.676 ± 0.0342

Tabela 4.6 – Taxa de erro obtida empregando energia ou potência para classificação com multiplas classes.(A, B, C, D, E) sem normalização.

4.5.2 EXPERIMENTO # 2

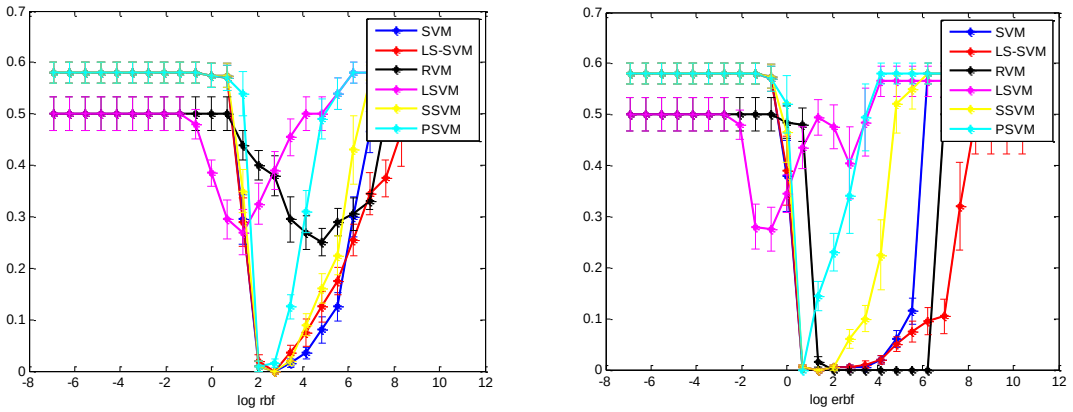
O objetivo deste experimento foi analisar o desempenho das diferentes implementações de SVM na tarefa de classificação dos sinais de EEG do grupo A e E e verificar quais destas implementações apresentam os melhores resultados. Na Tabela 4.7 são exibidos os melhores resultados das classificações realizadas pelas variantes das Máquinas de Vetor de Suporte versus as diversas estatísticas extraídas dos vetores de características baseados em wavelet. O resultado apresentado corresponde à média e ± um desvio da taxa de erro alcançado pelo classificador nos 10 *folds*.

Tabela 4.7 – Taxa de erro das variantes de SVM versus vetores de características extraidos via wavelet.

Tipo de kernel	Vetor de característica	Variantes de SVM					
		SVM	LS-SVM	RVM	LSVM	S-SVM	P-SVM
Rbf	Série EEG não processada	**0,000 ± 0,000**	**0,000 ± 0,000**	0,250±0,0269	0,270±0,0448	**0,000 ± 0,000**	0,010 ± 0,100
	Média	0,100± 0,0197	0,095±0,0229	0,100±0,0211	0,195±0,0311	0,100±0,0289	0,135±0,0269
	Desvio padrão	**0,000 ±0,0000**	**0,000±0,0000**	**0,000±0,0000**	**0,000±0,0000**	**0,000±0,0000**	0,005±0,0050
	Máximos	**0,000±0,0000**	0,0050±0,005	0,0050±0,005	0,005±0,0050	0,0050±0,005	0,0100±0,006
	Mínimos	**0,000±0,0000**	**0,000±0,0000**	**0,000±0,0000**	0,005±0,0050	**0,000±0,0000**	**0,000±0,0000**
	Médias das potências	0,0050±0,005	**0,000±0,000**	**0,000±0,000**	0,020±0,011	0,005±0,005	0,025±0,008
	Média dos valores absolutos	**0,000±0,0000**	**0,000±0,0000**	0,005±0,0050	0,005±0,0050	0,010±0,0100	0,010±0,0066
	Taxa entre sub-banda	0,385±0,0366	0,385±0,0259	0,405±0,0411	0,43±0,0343	0,425±0,0214	0,43±0,0226
	Todas as estatísticas	0,005±0,005	0,01±0,00667	**0,000 ± 0,000**	0,03 ±0,00816	0,01 ±0,0066	0,06 ±0,0145
Erbf	Série EEG não processada	**0,000 ±0,0000**	**0,000 ±0,0000**	**0,000 ±0,0000**	0,275 ±0,0430	**0,000 ±0,0000**	**0,000 ±0,0000**
	Média	0,105 ±0,0273	0,100 ±0,0289	0,115 ±0,0248	0,21 ±0,0314	0,115±0,0259	0,150 ±0,0298
	Desvio padrão	**0,000 ±0,0000**	**0,000 ±0,0000**	0,005 ±0,005	**0,000 ±0,0000**	**0,000 ±0,0000**	0,010±0,0066
	Máximos	**0,000 ±0,0000**	**0,000 ±0,0000**	**0,000 ±0,0000**	0,005 ±0,005	0,005±0,0050	0,010±0,0066
	Mínimos	**0,000 ±0,0000**	**0,000 ±0,0000**	**0,000 ±0,0000**	0,005 ±0,005	0,005±0,0050	0,010±0,0066
	Médias das potências	**0,000 ±0,0000**	**0,000 ±0,0000**	0,005 ±0,005	**0,000 ±0,0000**	0,005 ±0,0050	0,020 ±0,0111
	Média dos valores absolutos	0,005 ±0,0050	0,005 ±0,0050	**0,000 ±0,0000**	0,005 ±0,005	0,010±0,0066	0,010 ±0,0066
	Taxa entre sub-banda	0,415±0,0373	0,405±0,0383	0,385 ±0,0317	0,43 ±0,0238	0,465±0,0428	0,475 ±0,0344
	Todas as estatísticas	0,010±0,0066	0,005 ±0,0050	0,01 ±0,00667	0,025±0,0083	0,020±0,0081	0,075 ±0,0201

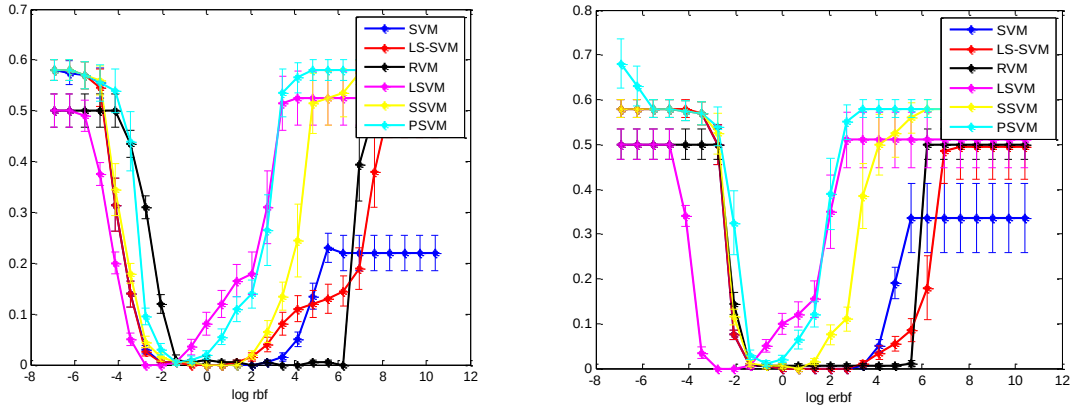
Observando-se os resultados obtidos, pode-se concluir que a SVM, a LS-SVM e a RVM apresentaram os melhores resultados. Também, pode-se concluir que os vetores de características associados à série EEG sem processamento, ao desvio padrão, ao máximo e ao

mínimo dos coeficientes wavelet produziram resultados com baixa taxa de erro para uma faixa de valores do parâmetro do kernel. As ilustrações a seguir mostram mais claramente o desempenho das variantes de SVM quando utilizando os vetores de características baseados nos coeficientes wavelet. Elas também ilustram o impacto da variação do parâmetro do kernel sobre as diversas implementações de SVM baseada na taxa de erro de classificação. As Ilustração 4.9, Ilustração **4.10**, Ilustração **4.11** e Ilustração **4.12** mostram o desempenho dos classificadores utilizando a série EEG original (sem processamento via transformada wavelet), vetor de características baseado no desvio padrão, no máximo e no mínimo dos coeficientes wavelet como entrada respectivamente.



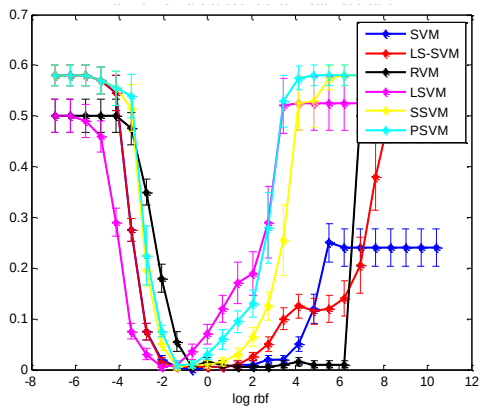
a) Kernel RBF b) Kernel ERBF

Ilustração 4.9 – Taxa de erro na classificação utilizando EEG original com kernel RBF e ERBF.

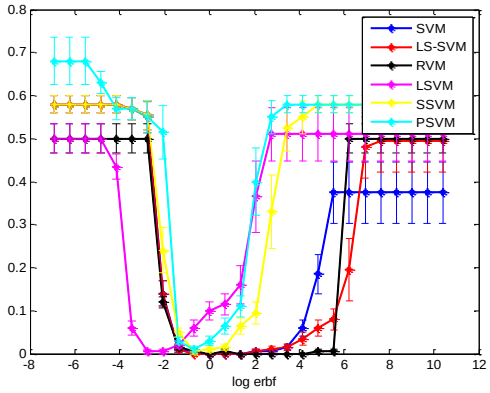


a) Kernel RBF b) Kernel ERBF

Ilustração 4.10 – Taxa de erro na classificação utilizando o desvio padrão dos coeficientes wavelet com kernel RBF e ERBF.

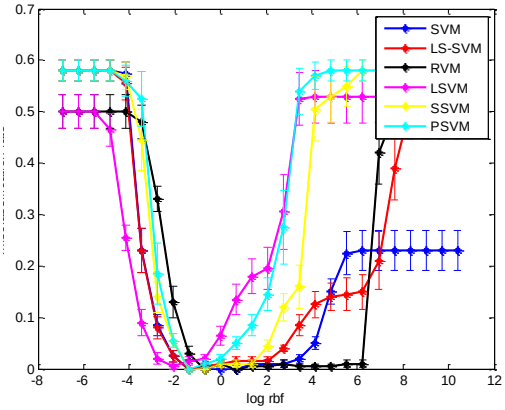


a) Kernel RBF

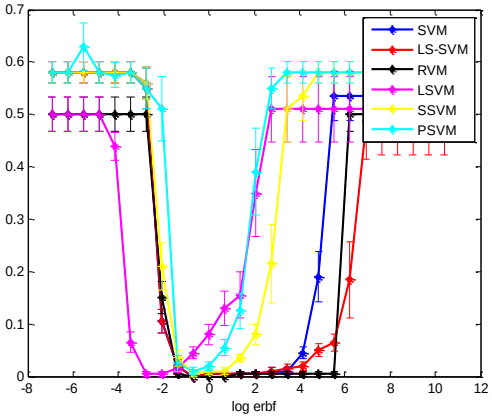


b) Kernel ERBF

Ilustração 4.11 – Taxa de erro na classificação utilizando o máximo dos coeficientes wavelet com kernel RBF e ERBF.



a) Kernel RBF

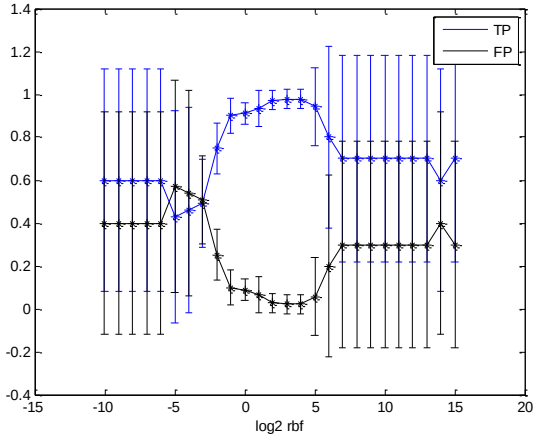


b) Kernel ERBF

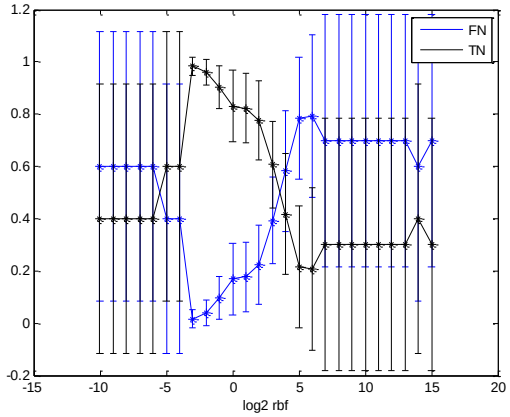
Ilustração 4.12 – Taxa de erro na classificação utilizando o mínimo dos coeficientes wavelet com kernel RBF e ERBF.

As Ilustração 4.13a e Ilustração 4.13b apresentam as taxas de verdadeiro/falso positivo e verdadeiro/falso negativo para cada um dos valores do parâmetro do kernel, quando empregando como vetor de características a média dos coeficientes wavelet. Comparando estas ilustrações com o melhor resultado apresentado na Tabela 4.7, observa-se que o valor do parâmetro do kernel que produz menor taxa de erro não coincide com o pico da taxa de verdadeiro negativo. Isto mostra que os valores dos parâmetros do kernel apresentados na Tabela 4.7 podem sofrer pequenas alterações se for adotado outro critério de desempenho. Já na Ilustração 4.14a apresentam-se as curvas ROC para o valor do parâmetro do kernel igual a 0.5, que corresponde à menor taxa de erro. Observe que, como foi utilizada validação cruzada (10-*folds*), foram gerados 10 modelos diferentes. A Ilustração 4.14a apresenta estas curvas, sendo que algumas estão sobrepostas e a área média sobre a curva ROC é de 0.93. Na Ilustração 4.13c, Ilustração 4.13d e na Ilustração 4.14b são apresentados os mesmos gráficos

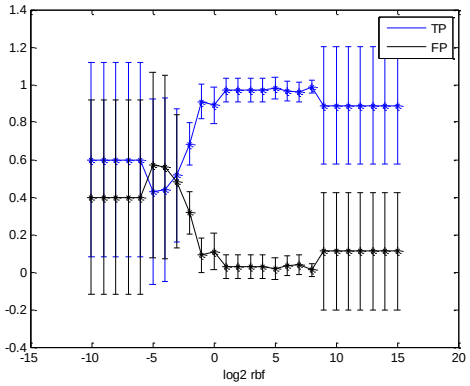
para vetor de características obtido via desvio padrão dos coeficientes wavelet. Observe que a área sobre a curva ROC é de 0.98, maior quando comparada ao vetor de características usando média. Isto confirma os resultados apresentados na Tabela 4.7, na qual as características extraídas via o desvio padrão produziram melhor resultado.



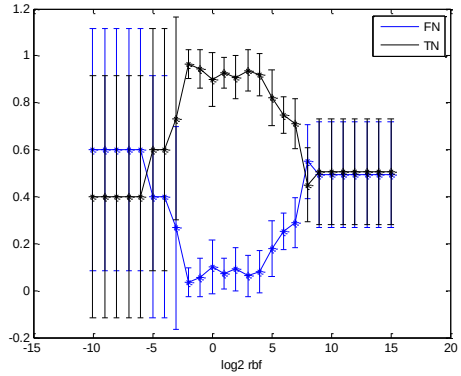
a) Verdadeiro/Falso Positivo realizado pela SVM usando média dos coef. wavelet.



b) Verdadeiro/Falso Negativo realizado pela SVM usando média dos coef. wavelet.



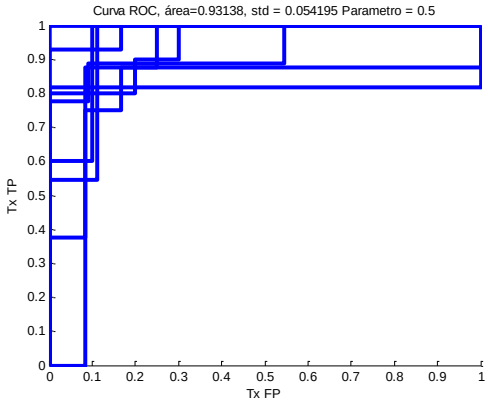
c) Verdadeiro/Falso Positivo realizado pela SVM usando desvio padrão dos coef. wavelet.



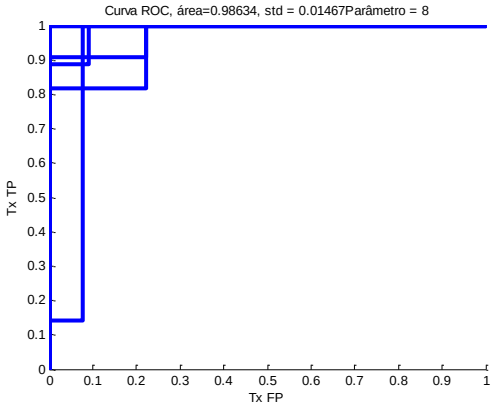
d) Verdadeiro/Falso Negativo realizado pela SVM usando desvio padrão dos coef. wavelet.

Ilustração 4.13 – Classificação utilizando o máximo dos coeficientes wavelet com kernel RBF e ERBF.

Considerando o erro médio dos classificadores gerados por cada uma das variantes de SVM (SVM, LS-SVM, SSVM, LSVM, PSVM e RVM), como mostrado na Tabela 4.7. Os classificadores SVM, LS-SVM e RVM mostraram um melhor desempenho comparado aos demais, i.e, eles apresentaram, em maior número, os melhores resultados (média de todas as classificações). A RVM mostrou-se menos sensível às variações do parâmetro do kernel, exibindo uma baixa taxa de erros para uma faixa maior de parâmetros.



a) Curva ROC para SVM com kernel RBF e parâmetro 0.5 usando média dos coef. wavelet.



b) Curva ROC para SVM com kernel RBF e parâmetro 0.5 usando desvio padrão dos coef. wavelet.

Ilustração 4.14 – Curva ROC utilizando o para o kernel RBF.

Já em relação à função de kernel, o maior impacto foi notado na classificação da série não processada. A Ilustração 4.9 mostra que os classificadores que utilizaram a função de kernel ERBF apresentaram bons resultados para uma faixa maior de valores do parâmetro do que os classificadores com kernel RBF.

Em relação aos vetores de características, a série original (sem o processamento utilizando via transformada wavelet) apresentou bons resultados na maioria dos classificadores. Isto pode ser justificado pelo fato que a série original contém todas as informações, bastando apenas que o classificador consiga extraí-las adequadamente. Entretanto foi possível obter resultados equivalentes utilizando o vetor de características baseado no desvio padrão dos coeficientes wavelet. Este último consiste de uma representação mais compacta dos dados, contendo características que permitiram realizar uma classificação com bons resultados. Além disso, com o emprego de estatísticas sobre a transformada wavelet foi possível obter bons resultados para uma faixa maior de valores do parâmetro do kernel

4.5.3 EXPERIMENTO # 3

O objetivo deste experimento foi analisar o impacto da utilização de diferentes tipos de wavelet base no desempenho de classificação, utilizando SVMs e suas variações. Foram utilizados os seguintes tipos de wavelet: Haar, Daubechie de ordem 2 (Db2) e Daubechie de ordem 4 (Db4). Foi dada ênfase aos wavelets da família Daubechie, pois de acordo com

SUBASI (2007), a característica de suavidade deste tipo de wavelet a torna mais apropriada para detectar mudanças em sinais de EEG.

Foi calculada a transformada wavelet discreta sobre o sinal de EEG e, a partir dos coeficientes gerados para cada tipo de wavelet, foram produzidos os vetores de características descritos no início deste capítulo, o que resultou em um conjunto de vinte e quatro vetores de características, sendo oito referentes ao wavelet Haar, oito referentes ao wavelet Db2 e oito referentes ao Db4. Para realizar a classificação destes vetores de características, foram empregados três tipos de classificadores: a SVM tradicional, a LS-SVM e a RVM. Estas variantes de SVM foram selecionadas com base nos resultados apresentados por estas no experimento #2.

As tabelas a seguir mostram a taxa de erro dos melhores resultados produzidas pelas três variações de SVM, na classificação do sinal EEG (classes A e E) para as wavelet do tipo Haar, Db2 e Db4 respectivamente. Nas colunas "parâmetro do kernel" são armazenados os valores do parâmetro do kernel utilizados pelos classificadores que obtiveram as menores taxa de erro. Quando existe mais de um destes valores, para o mesmo caso de classificação, significa que ambos produziram o mesmo resultado.

Nas Tabela 4.8, Tabela 4.9 e Tabela 4.10 são exibidas as taxas de erro na classificação dos vetores de características geradas a partir do wavelet de Haar, Db2 e Db4, respectivamente.

Tabela 4.8 – Taxa de erro na classificação dos vetores de características da wavelet Haar.

Tipo de kernel	Vetores de características	SVM		LS-SVM		RVM	
		Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro
Rbf	Média	2	0,090±0,0233	0,5	0,090±0,0180	0,015625	0,105±0,0252
	Desvio padrão	0,25; 0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,25; 0,5; 1; **2**; **4**	**0,000±0,0000**	0,0078125	**0,000±0,0000**
	Máximos	0,25; 0,5; 1; 8	**0,000±0,0000**	0,25; 0,5; 1; 8	**0,000±0,0000**	0,015625	0,010±0,0067
	Mínimos	0,25; 0,5; 1; **2**; **4**	**0,000±0,0000**	0,25; 0,5	**0,000±0,0000**	0,0078125	0,005±0,0050
	Médias das potências	0,0625; 0,125; 0,25; 0,5; 1; **2**	**0,000±0,0000**	0,0625; 0,125; 0,25; 0,5	**0,000±0,0000**	0,0019531	**0,000±0,0000**
	Média dos valores absolutos	0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,25; 0,5; 1; **2**	**0,000±0,0000**	0,0078125	0,005±0,0050
	Taxa entre sub-banda	1	0,380±0,0318	0,03125	0,375±0,0201	256; 512; 1024; 2048; 4096; 8192; 16384; 32768	0,500±0,0333
	Todas as estatísticas	8; 16	**0,000±0,0000**	8	0,005±0,0050	0,03125	0,070±0,0111
Erbf	Média	**4**	0,075±0,0239	2	0,070±0,0213	0,015625	0,170±0,0153
	Desvio padrão	0,5; 1; **2**; **4**; 8; 16; 32	**0,000±0,0000**	0,5; 1; **2**; **4**; 8; 16	**0,000±0,0000**	0,0078125	0,020±0,0082
	Máximos	0,5; 1; **2**; **4**; 8; 16	**0,000±0,0000**	0,5; 1; **2**; **4**; 8; 16	**0,000±0,0000**	0,03125	0,110±0,0414
	Mínimos	0,5; 1; **2**; **4**	**0,000±0,0000**	0,5; 1; **2**; 4	**0,000±0,0000**	0,0078125	0,100±0,0211
	Médias das potências	0,25; 0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,25; 0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,0039063	0,010±0,0100
	Média dos valores absolutos	0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,5; 1; **2**	**0,000±0,0000**	0,0078125	0,025±0,0112
	Taxa entre sub-banda	**4**	0,360±0,0277	**4**	0,365±0,0259	0.00097656; 0.0019531; 128; 256; 512; 1024; 2048; 4096; 8192; 16384; 32768	0,500±0,0333
	Todas as estatísticas	**4**; 8; 16	**0,000±0,0000**	**4**; 8; 16	0,005±0,0050	0,03125	0,085±0,0131

Tabela 4.9 – Taxa de erro na classificação dos vetores de características da wavelet Db2.

Tipo de kernel	Vetores de características	SVM		LS-SVM		RVM	
		Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro
Rbf	Média	**2**	0,180±0,0133	0,5	0,085±0,0224	0,015625	0,075±0,0227
	Desvio padrão	0,5; 1; **2**; 8; 16	**0,000±0,0000**	0,5; 1; **2**	**0,000±0,0000**	0,0078125	0,005±0,0050
	Máximos	0,25; 0,5; 1	**0,000±0,0000**	0,25; 0,5; 1; **2**; **4**	**0,000±0,0000**	0,015625	0,005±0,0050
	Mínimos	0,25	**0,000±0,0000**	0,25	0,005±0,0050	0,0078125	0,025±0,0112
	Médias das potências	0,25; 0,5; 1; **2**; **4**; 16	**0,000±0,0000**	0,25; 0,5; 1	**0,000±0,0000**	0,0019531	0,005±0,0050
	Média dos valores absolutos	1; **2**	**0,000±0,0000**	1; **2**	**0,000±0,0000**	0,015625	0,020±0,0111
	Taxa entre sub-banda	**4**	0,580±0,0260	**2**	0,570±0,0291	0,0019531	0,495±0,0263
	Todas as estatísticas	32	0,005±0,0050	8	**0,000±0,0000**	0,03125	0,045±0,0189
Erbf	Média	1; **2**; 4	0,090±0,0233	1; **2**	0,010±0,0233	0,015625	0,195±0,0229
	Desvio padrão	0,5; 1; **2**; **4**; 8; 16	**0,000±0,0000**	0,5; 1; **2**; **4**; 8; 16	**0,000±0,0000**	0,0078125	0,025±0,0083
	Máximos	0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,03125	0,070±0,0213
	Mínimos	0,5; 1; **2**; **4**; 16	0,005±0,0050	0,5; 1; **2**; **4**; 8	0,005±0,0050	0,03125	0,080±0,0170
	Médias das potências	0,5; 1; **2**; **4**; 8	**0,000±0,0000**	0,25; 0,5; 1	**0,000±0,0000**	0,0039063	0,015±0,0107
	Média dos valores absolutos	1; **2**; **4**; 8	**0,000±0,0000**	1; **2**	**0,000±0,0000**	0,0078125	0,005±0,0090
	Taxa entre sub-banda	8	0,570±0,0250	8	0,410±0,0277	0,0039063	0,490±0,0296
	Todas as estatísticas	**4**; 8	0,005±0,0050	**4**	0,005±0,0050	0,03125	0,055±0,0174

Tabela 4.10 – Taxa de erro na classificação dos vetores de características da wavelet Db4.

Tipo de kernel	Vetores de características	SVM		LS-SVM		RVM	
		Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro
Rbf	Média	0,5	0,100±0,0197	0,5	0,095±0,0229	0,015625	0,070±0,0238
	Desvio padrão	1; **2**; **4** ; 8	**0,000±0,0000**	0,5; 1; **2**	**0,000±0,0000**	0,0078125; 0,015625	0,010±0,0067
	Máximos	0,5	**0,000±0,0000**	0,25; 0,5; 1; **2**	0,005±0,005	0,015625	0,010±0,0067
	Mínimos	0,25; 0,5; 1; **2**	**0,000±0,0000**	0,25	**0,000±0,0000**	0,015625	0,010±0,0067
	Médias das potências	0,125; 0,25; 0,5; 1; **2**; **4**	0,005±0,005	0,5; 1	**0,000±0,0000**	0,0019531	0,020±0,0111
	Média dos valores absolutos	1; **2**; **4**	**0,000±0,0000**	1; **2**	**0,000±0,0000**	0,015625	0,020±0,0111
	Taxa entre sub-banda	0,125	0,395±0,0366	0,5	0,395±0,0259	512; 1024; 2048; 4096; 8192; 16384; 32768	0,500±0,0333
	Todas as estatísticas	32	0,005±0,0050	**2**; 8	0,010±0,0067	0,03125	0,035±0,0076
Erbf	Média	0,5	0,110±0,0256	0,5	0,100±0,0289	0,015625	0,200±0,0316
	Desvio padrão	1; **2**; **4**; 8; 16	**0,000±0,0000**	1; **2**; **4**; 8; 16	**0,000±0,0000**	0,0078125	0,045±0,0117
	Máximos	1; **2**; **4**	**0,000±0,0000**	0,5; 1; **2**; **4**	**0,000±0,0000**	0,015625	0,070±0,0153
	Mínimos	0,5	**0,000±0,0000**	0,5	**0,000±0,0000**	0,015625	0,060±0,0125
	Médias das potências	**4**; 8	**0,000±0,0000**	**2**; **4**	**0,000±0,0000**	0,0078125	0,035±0,0107
	Média dos valores absolutos	**2**; **4**; 8; 16; 32	0,005±0,0050	**2**; **4**; 8; 16	0,005±0,0050	0,0078125	0,120±0,0213
	Taxa entre sub-banda	**4**	0,415±0,0373	8	0,405±0,0383	0,00097656; 0,0019531; 64; 128; 256; 512; 1024; 2048; 4096; 8192; 16384; 32768	0,500±0,0333
	Todas as estatísticas	**4**; 8	0,010±0,0067	**4**	0,005±0,0050	0,03125	0,030±0,0082

Com base nestes resultados é possível notar que o fator que mais influencia na taxa de erro é o tipo de estatística utilizada para gerar os vetores de características e não a wavelet base empregada. Os vetores baseados no desvio padrão dos coeficientes wavelet permitiram aos classificadores analisados obter os melhores resultados. Outros vetores também mostraram bons resultados, como os vetores baseados nos máximos, mínimos e potência

média. Pode-se notar também que a RVM possui uma taxa de erro maior do que a SVM e a LS-SVM considerando um mesmo vetor de características. Dentre os classificadores analisados neste experimento, a implementação clássica da SVM apresentou os melhores resultados, seguida pela LS-SVM.

A partir dos resultados mostrados neste experimento, nota-se que a utilização de kernel RBF ou ERBF não causa grandes mudanças no desempenho dos classificadores aqui testados. Foi possível observar também que alguns valores do parâmetro das funções de kernel RBF e ERBF, como por exemplo, os valores 2 e 4, aparecem freqüentemente associados às menores taxas de erro. Este valor pode ser um bom ponto de partida para ajustes mais finos do parâmetro kernel empregado no classificador a ser utilizado.

Uma verificação adicional foi realizada com o objetivo de verificar o desempenho do classificador SVM, o qual apresentou os melhores neste experimento, na tarefa de distinguir outras combinações de classes, a saber, A versus B, C versus D, A e B versus C e D. As tabelas a seguir apresentam os resultados. Nas Tabela 4.11, Tabela **4.12** e Tabela **4.13** são mostrados os resultados (taxa de erro) obtidos pela SVM na classificação de vetores de características gerados a partir do wavelet Haar, db2, db4 respectivamente.

Tabela 4.11 – Taxa de erro na classificação com SVM dos vetores de características do wavelet Haar.

Tipo de kernel	Vetores de características	A vs. B		C vs. D		AB vs CD	
		Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro
Rbf	Média	**4**	0,365±0,0280	32	0,455±0,0345	**4**	0,285±0,0308
	Desvio padrão	**4**	**0,060±0,0194**	0,5	**0,240±0,0287**	**2**	**0,014±0,0042**
	Máximos	16	0,115±0,0248	16	0,345±0,0263	**4**	0,080±0,0186
	Mínimos	**4**	0,115±0,0183	**4**	0,320±0,0429	**4**	0,072±0,0120
	Médias das potências	1	0,070±0,0186	0,125	0,305±0,0369	1	**0,012±0.0042**
	Média dos valores absolutos	1	**0,050±0,0167**	0,5	**0,240±0,0420**	**2**	**0,017±0,0053**
	Taxa entre sub-banda	**2**	0,410±0,0297	0,5	0,430±0,0374	0,5	0,322±0,0192
	Todas as estatísticas	**4**	0,100±0,0258	**8**	0,300±0,0333	16	0,037±0,0077
Erbf	Média	0,5	0,345±0,0229	16	0,500±0,0333	**4**	0,290±0,0251
	Desvio padrão	8	**0,055±0,0189**	2	**0,225±0,0250**	**4**	**0,015±0,0056**
	Máximos	16	0,145±0,0263	**8**	0,355±0,0398	8	0,074±0,0177
	Mínimos	8	0,115±0,0224	**8**	0,330±0,0318	1	0,060±0,0100
	Médias das potências	**4**	0,070±0,0200	0,5	0,335±0,0395	1; 2	**0,017±0,0099**
	Média dos valores absolutos	**4**	**0,050±0,0149**	2	**0,250±0,0279**	2; **4**	**0,012±0,0042**
	Taxa entre sub-banda	8	0,410±0,0379	0,25	0,415±0,0373	8	0,300±0,0171
	Todas as estatísticas	16	0,095±0,0217	**8**	0,280±0,0318	2; **4**	0,032±0,0084

Tabela 4.12 – Taxa de erro na classificação com SVM dos vetores de características do wavelet Db2.

Tipo de kernel	Vetores de características	A vs. B		C vs. D		AB vs CD	
		Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro
Rbf	Média	**4**	0,345±0,0229	32	0,455±0,0293	**2**	0,290±0,0292
	Desvio padrão	**2**; **4**; 8	**0,050±0,1972**	0,5	**0,225±0,0227**	**2**	**0,007±0,0038**
	Máximos	**4**; 8; 16	0,155±0,0273	16	0,415±0,0365	**4**	0,075±0,0110
	Mínimos	16	0,160±0,0256	8	0,305±0,0293	**4**	0,082±0,0106
	Médias das potências	**2**	**0,050±0,0183**	0,25	0,270±0,0403	**2**	0,025±0,0075
	Média dos valores absolutos	**2**	**0,050±0,0197**	0,5	0,255±0,0321	**4**	**0,017±0,0065**
	Taxa entre sub-banda	0,25	0,370±0,0291	0,5	0,435±0,0334	0,25	0,417±0,0264
	Todas as estatísticas	32	0,080±0,0170	8	0,265±0,0289	8	**0,030±0,0082**
Erbf	Média	8	0,375±0,0214	16	0,505±0,0311	**2**	0,262±0,0277
	Desvio padrão	**2**; **4**	**0,045±0,0203**	**2**	**0,220±0,0281**	0,5; 1; **2**; **4**	**0,010±0,0049**
	Máximos	8	0,155±0,0293	0,5	0,380±0,0374	1	0,070±0,0012
	Mínimos	8	0,145±0,0302	16	0,285±0,0317	**4**	0,064±0,0106
	Médias das potências	1; **2**; **4**	**0,055±0,0138**	0,5	0,230±0,0327	**2**	**0,004±0,0038**
	Média dos valores absolutos	1	**0,045±0,0138**	**2**	**0,190±0,0323**	1; **2**; **4**	**0,010±0,0055**
	Taxa entre sub-banda	**2**	0,390±0,0306	**4**	0,435±0,0342	**4**	0,402±0,0206
	Todas as estatísticas	**4**; 8	0,085±0,0183	**4**	0,250±0,0279	**2**; 8	**0,022±0,0069**

Tabela 4.13 – Taxa de erro na classificação com SVM dos vetores de características do wavelet Db4.

Tipo de kernel	Vetores de características	A vs. B		C vs. D		AB vs CD	
		Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro	Parâmetro do kernel	Taxa de erro
Rbf	Média	**8**	0,370±0,0351	**2**	0,350±0,0337	**4**	0,310±0,0287
	Desvio padrão	**8**	**0,045±0,0138**	0,5	**0,245±0,0345**	2	**0,010±0,0055**
	Máximos	16	0,150±0,0269	0,25	0,355±0,0241	**4**	0,070±0,0133
	Mínimos	**4**	0,140±0,0125	**2**	0,310±0,0314	**4**	0,057±0,0065
	Médias das potências	**2**	**0,050±0,0129**	0,25	0,255±0,0450	**2**	0,025±0,0053
	Média dos valores absolutos	**8**	**0,055±0,0138**	1	**0,220±0,0359**	1	**0,007±0,0038**
	Taxa entre sub-banda	**2**	0,405±0,0320	0,03125	0,450±0,0316	0,5	0,362±0,0218
	Todas as estatísticas	32	0,080±0,0200	**2**	**0,275±0,0214**	16; 32	**0,027±0,0058**
Erbf	Média	0,5	0,370±0,0396	**2**	0,475±0,0271	1	0,265±0,0261
	Desvio padrão	**2; 4**	**0,055±0,0157**	**2**	**0,250±0,0387**	0,5; 1; 2; **4**	**0,012±0,0056**
	Máximos	16	0,145±0,0189	0,25	0,375±0,0423	8	0,062±0,0107
	Mínimos	1	0,170±0,0170	**4**	0,270±0,0238	1	0,052±0,0120
	Médias das potências	**2; 4**	**0,055±0,0189**	**2**	0,215±0,0350	8	0,010±0,0041
	Média dos valores absolutos	**2; 4**	**0,050±0,0149**	**4**	**0,200±0,0269**	8	**0,007±0,0038**
	Taxa entre sub-banda	16	0,405±0,0369	**2**	0,450±0,0236	**4**	0,402±0,0206
	Todas as estatísticas	**4**; 8	0,090±0,0233	**2**	**0,245±0,0376**	**2**; 8	**0,022±0,0069**

Com base nestes últimos resultados pode-se notar também que, na maioria dos casos, o melhor desempenho foi obtido com os vetores de características gerados com base no desvio padrão do coeficiente wavelet. Vale notar que o problema de separar os elementos das classes C e D mostrou-se mais difícil que os elementos das outras classes.

Também não ocorreu vantagem significativa entre os resultados obtidos com uma ou outra função de kernel. Os valores 2 e 4 do parâmetro do kernel foram aqueles que aparecerem mais associados aos melhores resultados obtidos com o classificador testado.

4.5.4 EXPERIMENTO # 4

O objetivo deste experimento é comparar o desempenho da abordagem clássica da SVM versus a RVM. Para isto foram utilizados os seguintes tipos de função de kernel: polinomial, RBF e ERBF. O parâmetro C foi fixado em 100, para o kernel RBF e para o ERBF, o valor de σ foi assumido um dos seguintes valores: 0.5, 1, 2, 4; e para o kernel polinomial os valores adotados foram: 1, 2, 3 e 4. Também foi utilizada validação cruzada 10-*fold*.

Na Tabela 4.14 são apresentados os resultados dos experimentos utilizando SVM e RVM para classificar vetores de características baseados em estatísticas sobre os coeficientes wavelet. Esta contém a taxa de erro obtida na classificação utilizando SVM e RVM, e o número de vetores suporte/relevantes empregados na classificação.

Analisando os resultados, observa-se que a RVM mostrou resultados promissores, produzindo um grande número de classificações corretas quando comparada à implementação clássica de SVM (CHAGAS *et al.*, 2008). No entanto, podemos ver que na maior parte dos casos a taxa de erro foi muito próxima nas duas implementações, mas a SVM apresenta uma pequena vantagem neste aspecto. Entretanto, na maioria dos casos, a RVM precisa de menos vetores de relevância (RV – *relevance vector*) quando comparado com o número de vetores de suporte (SV – *support vectors*) usados no SVM, com isso o tempo de classificação na fase de teste é reduzido para a RVM.

Tabela 4.14 – Taxa de erro de classificações usando SVM e RVM

Vetores de características	Tipo de função de kernel	Parâmetros do Kernel		SVM		RVM	
		σ	D	Taxa de erro	SV	Taxa de erro	RV
Sinal EEG não processado	ERBF	0.5	-	0,575 ±0,0226	180 ± 0,0000	0,500 ±0,0333	103,2 ±0,442
		1	-	0,395 ±0,0699	180 ± 0,0000	0,485 ± 0,032	96,3 ± 0,667
		2	-	0,005 ± 0,005	151,2 ± 0,533	0,480 ± 0,032	88,3 ± 0,667
		4	-	**0,000 ± 0,000**	66,6 ± 0,686	0,150 ± 0,010	57,5 ± 5,671
	Poly	-	1	0,290 ± 0,029	127,8 ± 1,331	0,250 ± 0,030	5,4 ± 0,266
		-	2	0,330 ± 0,041	130,3 ± 2,027	0,260 ± 0,030	8,2 ± 0,1333
		-	3	0,395 ± 0,040	105,0 ± 2,081	0,365 ± 0,079	4,1 ± 0,100
		-	4	0,340 ± 0,040	19,5 ± 0,909	0,500 ± 0,033	8,0 ± 0,258
	RBF	0.5	-	0,580 ± 0,020	180 ± 0,000	0,500 ± 0,033	180 ± 0,000
		1	-	0,575 ± 0,022	180 ± 0,000	0,500 ± 0,033	180 ± 0,000
		2	-	0,575 ± 0,022	180 ± 0,000	0,500 ± 0,033	23,7 ±0,5587
		4	-	0,195 ± 0,047	180 ± 0,0000	0,440 ± 0,028	100,5±10,896
Média dos coeficientes	ERBF	0.5	-	0,105 ± 0,027	154,8 ± 0,879	0,115 ± 0,030	35,2 ± 4,739
		1	-	0,115 ± 0,030	101,5 ± 0,957	0,115 ± 0,024	12,1 ± 0,276
		2	-	0,115 ± 0,030	98,7 ± 0,943	0,127 ± 0,027	18,6 ± 4,131
		4	-	0,115 ± 0,030	97,1 ± 0,7218	0,130 ± 0,024	79 ± 9,317
	Poly	-	1	0,455 ± 0,028	168,9 ± 1,858	0,560 ± 0,026	1,9 ± 0,2768
		-	2	0,120 ± 0,022	30,3 ± 0,6839	0,110 ± 0,022	10,5 ± 0,3726
		-	3	0,235 ± 0,025	51,2 ± 1,041	0,270 ± 0,060	11,3 ± 0,8171
		-	4	0,220 ± 0,022	47,1 ± 1,2949	0,375 ± 0,070	9,9 ± 0,9481
	RBF	0.5	-	0,100 ±0,0197	120,1 ± 1,187	0,250 ± 0,042	52,6 ± 1,3515
		1	-	0,130 ± 0,024	44,8 ± 0,8	0,150 ± 0,034	28,1 ± 2,6476
		2	-	0,120 ± 0,029	45,5 ± 1,013	0,125 ± 0,025	13,3 ± 1,333
		4	-	0,130 ± 0,026	65,1 ± 1,058	0,100 ± 0,021	94,8 ± 8,407
Desvio padrão dos coeficientes	ERBF	0.5	-	0,005 ± 0,005	87,6 ± 0,777	0,005 ± 0,005	96,7 ± 6,200
		1	-	**0,000 ± 0,000**	33 ± 0,4216	0,005 ± 0,005	17,9 ± 6,559
		2	-	**0,000 ± 0,000**	18,4 ± 0,4988	0,005 ± 0,005	71,1 ± 12,469
		4	-	**0,000 ± 0,000**	16,5 ± 0,4533	0,005 ± 0,005	133,8 ± 9,434
	Poly	-	1	0,005 ± 0,005	4,2 ± 0,13333	0,010 ± 0,006	1,9 ± 0,1000
		-	2	0,005 ± 0,005	6,7 ± 0,335	0,005 ± 0,005	2,2 ± 0,1333
		-	3	**0,000 ± 0,000**	3,6 ± 0,339	0,000 ± 0,000	2 ± 0,00
		-	4	0,010 ± 0,006	6,5 ± 0,268	0,005 ± 0,005	3,6 ± 0,266
	RBF	0.5	-	0,005 ± 0,005	63,5 ± 0,792	0,005 ± 0,005	67,1 ± 6,650
		1	-	**0,000 ± 0,000**	33,6 ± 0,6863	0,010 ± 0,006	32,5 ± 8,944
		2	-	**0,000 ± 0,000**	5,5 ± 0,7031	0,005 ± 0,005	65,2 ± 14,304
		4	-	**0,000 ± 0,000**	5,4±0,2211	0,005 ± 0,005	96 ± 11,174
Máximo dos coeficientes	ERBF	0.5	-	0,005 ± 0,005	90,9 ± 0,5859	0,005 ± 0,005	106,3 ± 4,96
		1	-	**0,000 ± 0,000**	28,6 ± 0,653	0,000 ± 0,000	26,1 ± 5,516
		2	-	**0,000 ± 0,000**	13,4 ± 0,476	0,005 ± 0,005	97,9 ± 6,685
		4	-	**0,000 ± 0,000**	12,3 ± 0,3666	0,000 ± 0,000	131,2 ± 4,239
	Poly	-	1	0,010 ± 0,006	5,4 ± 0,221	0,010 ± 0,006	1,9 ± 0,100
		-	2	0,015 ± 0,010	8,4 ± 0,400	0,040 ± 0,034	2,8 ± 0,200
		-	3	0,005 ± 0,005	6,3 ±0,213	0,005 ± 0,005	2 ± 0,000
		-	4	0,020 ± 0,008	10,8 ± 0,388	0,135 ± 0,078	21,6 ± 17,60
	RBF	0.5	-	**0,000 ± 0,000**	70,5 ± 0,542	0,005 ± 0,005	61,3 ± 10,127
		1	-	0,005 ± 0,005	9,3 ±2,221	0,015 ± 0,007	62,2 ± 8,7519
		2	-	0,005 ± 0,005	5,4 ±0,520	0,010 ± 0,006	40,6 ± 11,641
		4	-	0,010 ± 0,006	6,3 ±0,152	0,005 ± 0,005	118,2 ± 2,831

4.5.5 EXPERIMENTO # 5

O objetivo deste experimento foi verificar o comportamento das SVMs na classificação das múltiplas classes do conjunto de EEG descrito no início deste capítulo (A, B, C, D e E). Foram utilizados os métodos um contra um, um contra todos, MOC e ECOC para separar os elementos de cada classe. Estes métodos já foram descritos no capítulo 3 e são bastante populares em problemas de classificação com múltiplas classes. Como nos experimentos anteriores, foi utilizada validação cruzada 10-*fold*. Novamente, o parâmetro C foi fixado em 100 e o parâmetro do kernel foi variado no intervalo 2^i $i \in$ [-10:15]. Na Tabela 4.15 é apresentado a melhor taxa de acerto na classificação obtida com respectivo valor do parâmetro do kernel. Novamente, o resultado apresentado corresponde à média e ± um desvio da taxa de erro de classificação alcançado.

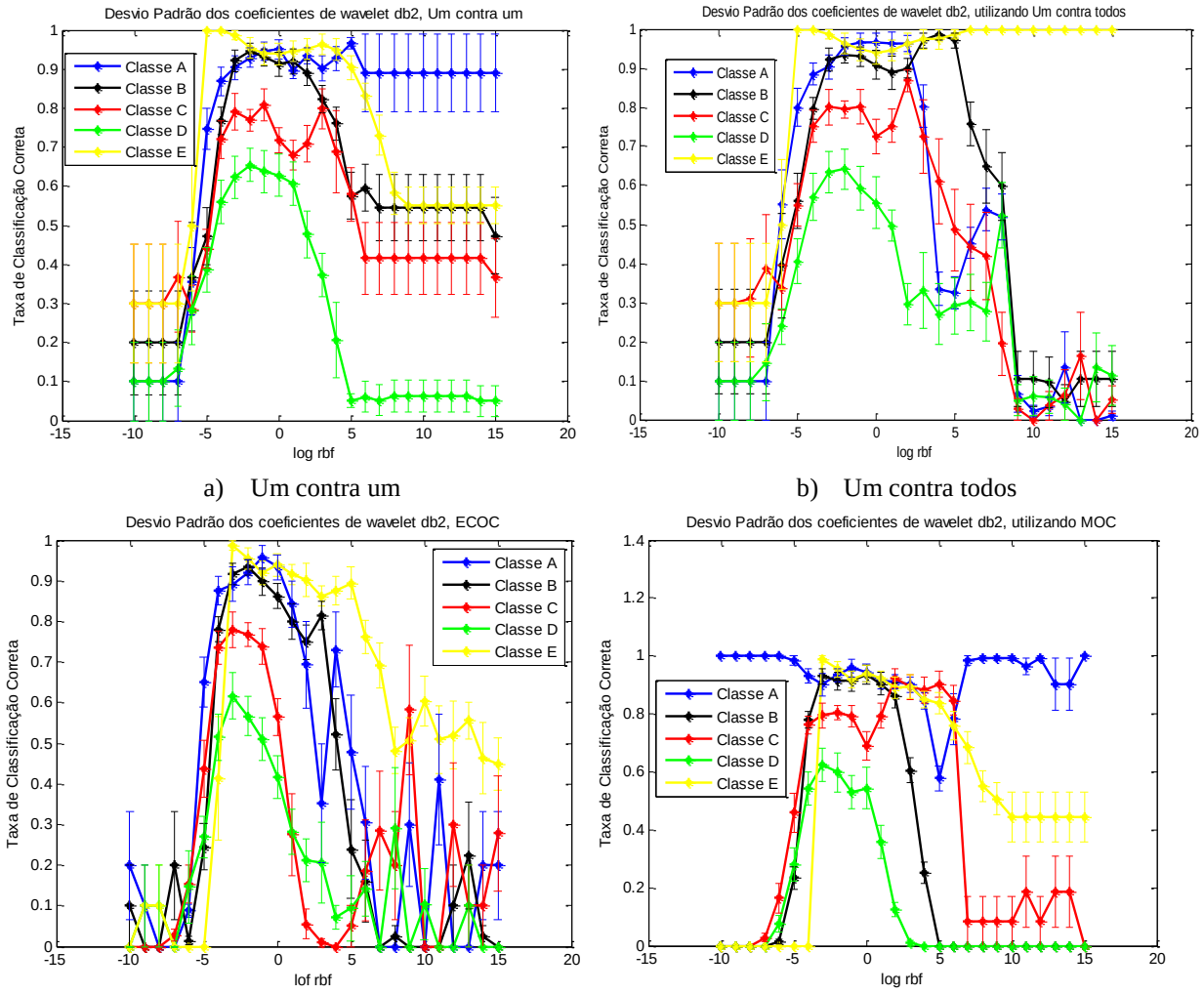
Tabela 4.15 – Resultados obtidos para SVMs usando o kernel RBF

Est.	Tipo Wave-let	Um Contra Um		Um Contra Todos		MOC		ECOC	
		σ^2	Taxa de erro na classificação	σ^2	Taxa de erro na classificação	σ^2	Taxa de erro na classificação		Taxa de erro na classificação
Média	Haar	2	0,540±0,0208	1	0,554 ±0,0238	0,5	0,646 ± 0,0438	1	0,546 ± 0,0169
	Db2	0,5	0,510±0,0193	1	0,522 ±0,0081	1	0,534 ± 0,0115	0,5	0,536 ± 0,0151
	Db4	1	0,533±0,0133	2	0,515 ±0,0358	1	0,595 ± 0,0468	1	0,465 ± 0,0353
Desvio	Haar	1	0,162±0,0145	0,125	0,156 ±0,0136	0,25	0,196 ± 0,0240	0,25	0,164 ± 0,0136
	Db2	0,5	0,150±0,0108	0,25	0,144±0,0145	0,125	0,150±0,01238	0,125	0,160 ± 0,0133
	Db4	2	0,150±0,0166	0,25	**0,142 ± 0,0141**	1	0,152 ± 0,0201	0,25	0,152 ± 0,0121

Os resultados ilustrados na Tabela 4.15 demonstram nitidamente a influência do tipo de vetor de características no desempenho do classificador. Pode-se observar que o desvio padrão gerou um vetor de características importante para a separação entre as classes, melhorando significativamente o desempenho do classificador. Este resultado já havia sido constato em problemas de classificação binária como os ilustrados nos experimentos descritos anteriormente.

As figuras a seguir ilustram o desempenho dos classificadores para cada estratégia de combinação, isto é, um contra um, um contra todos, MOC, ECOC. A Ilustração 4.15 mostra como a taxa de classificação correta, para cada classe, produzido pelos classificadores, utilizando o vetor de características baseado no desvio padrão dos coeficientes, varia em função do valor do parâmetro da função de kernel. Observe na Ilustração 4.15 que há uma faixa de valores do parâmetro do kernel que produz resultados bastante interessantes. No entanto, não existe nenhum valor do parâmetro do kernel, dentro da faixa empregada, que

produz 100 % de classificação correta para as classes D e C. As classes A e E são aquelas mais fáceis de serem classificadas corretamente. Dentre as abordagens empregadas, um contra um e um contra todos foram aquelas que produziram melhores resultados em termos de erro de classificação.



a) Um contra um

b) Um contra todos

c) ECOC

d) MOC

Ilustração 4.15 – Taxa de classificação correta dos classificadores para múltiplas classes.

Na Ilustração 4.16 é apresentada a taxa de classificação incorreta para cada estratégia abordada dos classificadores em função do valor do parâmetro do kernel. Nota-se que existe um valor de parâmetro de kernel na qual a taxa de classificação incorreta, entre os métodos de combinação, é muito similar.

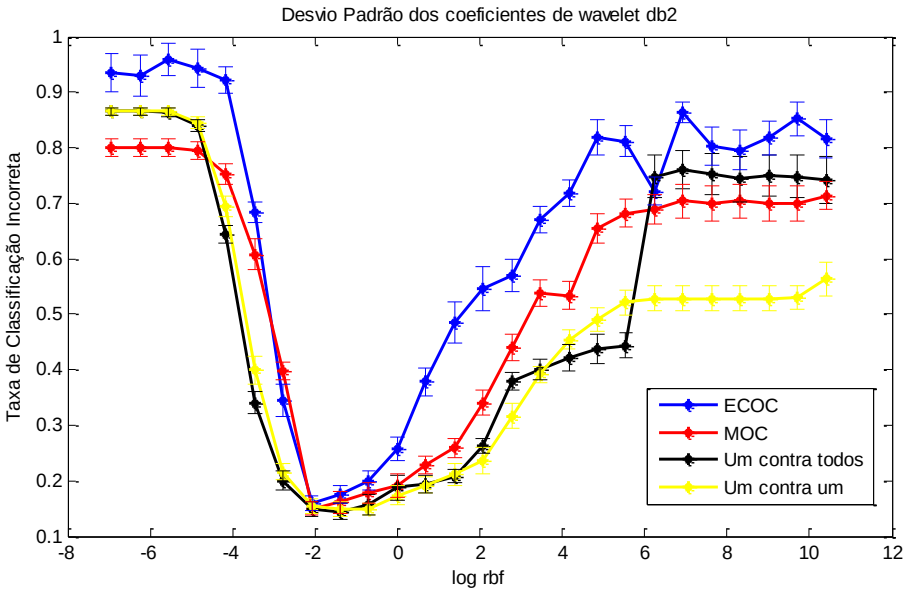


Ilustração 4.16 – Taxa de classificação incorreta dos classificadores para múltiplas classes.

Nas Tabela 4.16,Tabela 4.17, Tabela 4.18 e Tabela 4.19 é apresentado a matriz de confusão para alguns dos resultados ilustrados na Tabela 4.15. Especificamente, estas matrizes correspondem ao melhor resultado de classificação alcançada pela SVM, usando kernel RBF e desvio padrão dos coeficientes de wavelet Db2. Analisando os elementos da diagonal, que correspondem à classificação correta de cada classe, podemos observar que a taxa de classificação correta para as classes A, B, C atingem mais de 90%. Já para as classes C e D o desempenho não passa de 80%.

Tabela 4.16 – Matriz de confusão para SVMs com kernel RBF, $\sigma = 0.125$, desvio padrão sobre os coeficientes de wavelet Db2 e estratégia ECOC para classificação com múltiplas classes

	Classe Prevista				
Classe Real	Classe A	Classe B	Classe C	Classe D	Classe D
Classe A	0,891 ± 0,0428	0,043 ± 0,0228	0,000 ± 0,0100	0,010 ± 0,0100	0,009 ± 0,0091
Classe B	0,056 ± 0,0270	0,916 ± 0,0267	0,000 ± 0,0000	0,000 ± 0,0000	0,000 ± 0,0000
Classe C	0,018 ± 0,0182	0,000 ± 0,0000	0,779 ± 0,0490	0,145 ± 0,0490	0,017 ± 0,0116
Classe D	0,009 ± 0,0091	0,000 ± 0,0000	0,245 ± 0,0581	0,616 ± 0,0581	0,100 ± 0,0310
Classe E	0,000 ± 0,0000	0,000 ± 0,0000	0,000 ± 0,0000	0,012 ± 0,0125	0,987 ± 0,0125

Tabela 4.17 – Matriz de confusão para SVMs com kernel RBF, σ = 0.125, desvio padrão sobre os coeficientes de wavelet Db2 e estratégia MOC para classificação com múltiplas classes

	Classe Prevista				
Classe Real	Classe A	Classe B	Classe C	Classe D	Classe D
Classe A	0,901 ± 0,0389	0,062 ± 0,0294	0,016 ± 0,0167	0,010 ± 0,0100	0,009 ± 0,0091
Classe B	0,069 ± 0,0270	0,930 ± 0,0270	0,000 ± 0,0000	0,000 ± 0,0000	0,000 ± 0,0000
Classe C	0,018 ± 0,0182	0,000 ± 0,0000	0,795 ± 0,0428	0,154 ± 0,0487	0,031 ± 0,0169
Classe D	0,009 ± 0,0091	0,000 ± 0,0000	0,266 ± 0,0418	0,624 ± 0,0569	0,100 ± 0,0310
Classe E	0,000 ± 0,0000	0,000 ± 0,0000	0,000 ± 0,0000	0,012 ± 0,0125	0,987 ± 0,0125

Tabela 4.18 – Matriz de confusão para SVMs com kernel RBF, σ = 0.25, desvio padrão sobre os coeficientes de wavelet Db2 e estratégia um contra todos para classificação com múltiplas classes

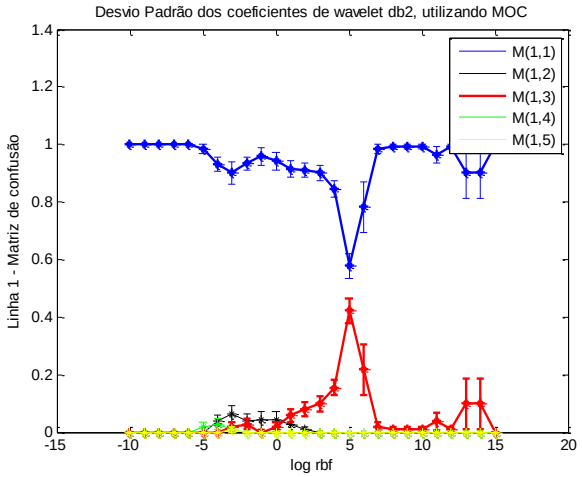
	Classe Prevista				
Classe Real	Classe A	Classe B	Classe C	Classe D	Classe D
Classe A	0,956 ± 0,0228	0,043 ± 0,0228	0,000 ± 0,0000	0,000 ± 0,0000	0,000 ± 0,0000
Classe B	0,045 ± 0,0158	0,933 ± 0,0195	0,000 ± 0,0000	0,021 ± 0,0152	0,000 ± 0,0000
Classe C	0,009 ± 0,0091	0,023 ± 0,0161	0,794 ± 0,0228	0,155 ± 0,0323	0,017 ± 0,0116
Classe D	0,009 ± 0,0091	0,022 ± 0,0151	0,258 ± 0,0440	0,641 ± 0,0520	0,068 ± 0,0247
Classe E	0,000 ± 0,0000	0,012 ± 0,0125	0,000 ± 0,0000	0,025 ± 0,0167	0,965 ± 0,0267

Tabela 4.19 – Matriz de confusão para SVMs com kernel RBF, σ = 0.5, desvio padrão sobre os coeficientes de wavelet Db2 e estratégia um contra um para classificação com múltiplas classes

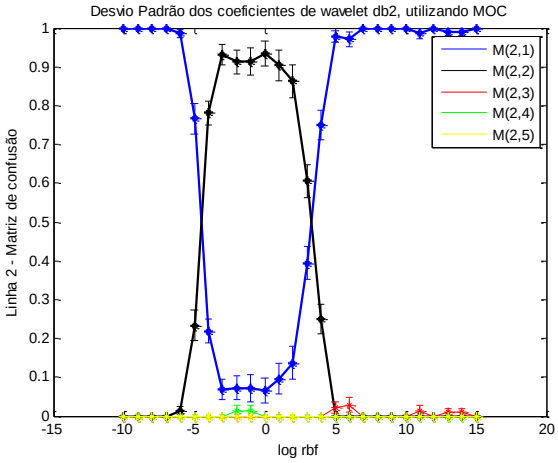
	Classe Prevista				
Classe Real	Classe A	Classe B	Classe C	Classe D	Classe D
Classe A	0,946 ± 0.0229	0,042 ± 0,0225	0,011 ± 0,0111	0,000 ± 0,0000	0,000 ± 0,0000
Classe B	0,055 ± 0,0217	0,930 ± 0,0224	0,000 ± 0,0000	0,014 ± 0,0143	0,000 ± 0,0000
Classe C	0,018 ± 0,0182	0,000 ± 0,0000	0,807 ± 0,0406	0,164 ± 0,0395	0,010 ± 0,0100
Classe D	0,000 ± 0,0000	0,000 ± 0,0000	0,295 ± 0,0477	0,638 ± 0,0539	0,065 ± 0,0194
Classe E	0,000 ± 0,0000	0,012 ± 0,0125	0,007 ± 0,0077	0,041 ± 0,0174	0,938 ±0,0281

A Ilustração 4.17 a seguir mostra o efeito da variação do parâmetro do kernel na matriz de confusão, onde M(*i* ,*j*) significa a taxa na qual a classe *i* (A, B, C, D, E) é rotulada para classe *j* pelo classificador, por exemplo M(1, 1) significa que a classe correta é A e o classificador atribuiu o rótulo para a classe A; M(1, 2) significa que a classe correta é A e o

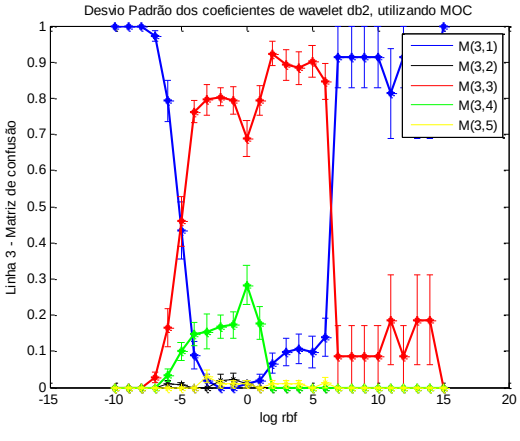
classificador atribuiu rótulo para classe B; M(1, 3) a classe correta é A e o classificador atribuiu rótulo para classe C; M(1, 4) a classe correta é A e o classificador atribuiu o rótulo para classe D; e M(1, 5) é o caso em que a classe correta é A e o classificador atribuiu o rótulo para classe E. A Ilustração 4.17a mostra que existe um valor do parâmetro do kernel que faz com o classificador classifique várias amostras pertencentes à classe A como classe D. Já na Ilustração 4.17b, pode-se observar que existe uma região na qual o classificador produz bom desempenho. Conforme já mencionado, analisando a Ilustração 4.17d e 4.17e nota-se que não há nenhum valor do parâmetro kernel que produz 100% de classificação correta dos elementos da classe C e D.



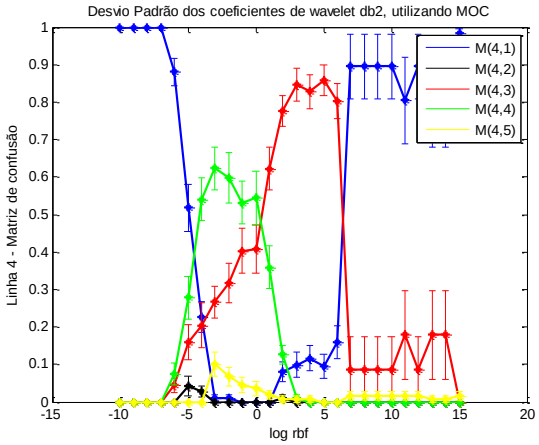
a) Classe A



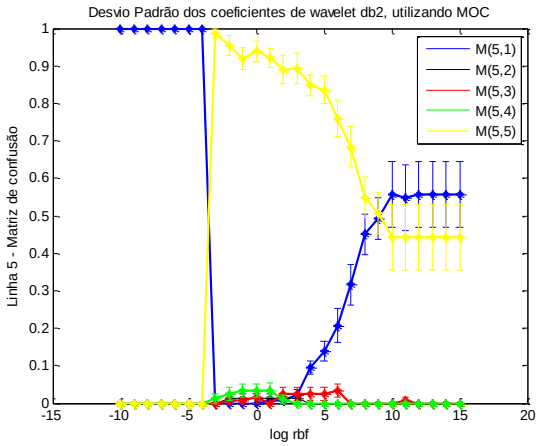
b) Classe B



c) Classe C



d) Classe D



e) Classe E

Ilustração 4.17 – Efeito da variação do parâmetro do kernel na classificação de cada classe.

De forma geral, os métodos utilizados para combinar os classificadores binários, apresentaram neste experimento, um desempenho equivalente entre si, com a estratégia um contra todos e um contra um apresentando resultados um pouco superiores. No entanto, novos experimentos são necessários de forma a exaurir todas as possibilidades.

CAPÍTULO 5 - CONCLUSÕES E TRABALHOS FUTUROS

5.1 CONCLUSÕES E PRÓXIMOS TRABALHOS

Os resultados obtidos a partir dos experimentos descritos no capítulo 4 permitem concluir que as técnicas baseadas em máquinas de vetores de suporte são bastante úteis na classificação dos sinais de EEG do conjunto de dados analisado. As SVMs mostraram uma boa capacidade de distinguir entre os elementos dos subconjuntos de sinais de EEG. Nesta dissertação, diferentes implementações de máquinas de vetores de suporte foram utilizadas para classificar um conjunto de sinais de EEG, utilizando diferentes vetores de características como entrada. Os classificadores, de forma geral, apresentaram bom desempenho tanto para a classificação binária quanto para múltiplas classes. A implementação tradicional da SVM mostrou os melhores resultados em relação aos demais. A RVM mostrou, na maioria dos experimentos, uma região maior onde os resultados são bons (ou seja, um maior número de parâmetros para o kernel associados aos bons resultados) e produziu na maioria dos casos uma solução com um número de vetores relevantes menores quando comparado com o numero de vetores suporte para SVM tradicional.

Quanto ao tipo de entrada utilizada, a série original (sem o processamento via wavelet e cálculo de estatísticas) permitiu que os classificadores atingissem bons resultados. Mas os experimentos também mostraram que os vetores resultantes do processo de extração de características, em especial o desvio padrão dos coeficientes wavelet, permitiram aos classificadores obter resultados equivalentes aos obtidos com a série original utilizando uma representação muito mais compacta da informação. O tipo de wavelet utilizado no processamento da série original não mostrou um grande impacto, sendo os resultados obtidos com a utilização de cada um delas equivalentes.

Além das classificações binárias (onde os vetores de características citados se destacaram), o vetor baseado no desvio padrão também se mostrou promissor na classificação envolvendo mais que duas classes. As classificações envolvendo múltiplas classes, precisam ser analisadas mais profundamente a fim de obter mais informações sobre o seu

comportamento. Os estudo preliminares apresentados nesta dissertação, permitem notar que as técnicas um contra um, um contra todos, ECOC e MOC apresentam valores competitivos entre si.

Dentre as diferentes implementações testadas, a implementação da SVM tradicional apresentou os melhores resultados, seguida da LS-SVM. A RVM, como comentado anteriormente, apresentou bons resultados e uma menor sensibilidade à variação do parâmetro do kernel, i.e., existem mais valores do parâmetro do kernel que podem conduzir a RVM às melhores classificações. Este comportamento foi observado na maior parte dos experimentos, independente do vetor de características utilizado.

O tipo de função de kernel teve pouco impacto no desempenho dos classificadores. Entretanto, teve um impacto significativo nos parâmetros da função de kernel associados com boas classificações quando foi utilizada como entrada a série EEG sem processamento (sem o cálculo de coeficientes wavelet e estatísticas); neste caso foi possível notar que o kernel ERBF fez com que mais valores do parâmetro do kernel estivem associados com boas classificações (taxas de erro mais baixas).

A utilização da série sem processamento, gerou bons resultados com as diferentes implementações de SVM. Vale lembrar que a série original é um vetor que contém todas as características disponíveis do conjunto de dados analisado e que neste caso, possui dimensão igual 4096. Entretanto os vetores de características possuem papel importante na classificação, os experimentos com a utilização dos vetores de características (em especial o desvio padrão dos coeficientes wavelet) mostraram que é possível obter classificações com taxas de erro comparáveis àquelas obtidas com a série sem processamento; porém com uma diferença considerável, utilizando vetores de entrada com dimensão igual a 6. Estes vetores de características conseguiram gerar outra representação para o sinal de EEG, bem mais compacta que do que a representação original, contendo características que permitiram realizar boas classificações. Esta diminuição da dimensão do vetor de entrada afeta também o custo computacional necessário para realizar a classificação.

Com base nas observações feitas sobre os experimentos realizados, o classificador obtido com a utilização da implementação tradicional da SVM, com o vetor de características baseado no desvio padrão dos coeficientes wavelet, e com o parâmetro do kernel com valores como 2 ou 4, é um bom ponto de partida para classificação de elementos do conjunto de EEGs analisado nesta dissertação e potencialmente para outros conjuntos de EEGs.

A capacidade de distinguir entre subconjuntos de sinais de EEG apresentada pelas técnicas de classificação baseadas em máquinas de vetor de suporte e na extração de

características baseada no cálculo de estatísticas sobre os coeficientes wavelet do sinal original, pode ser útil na análise de EEG associados a uma patologia específica, como por exemplo, casos de epilepsia ou de esquizofrenia.

Como trabalho futuro pretende-se aplicar a abordagem proposta a outros conjuntos de dados de sinais de EEG e a outros tipos de séries temporais, como por exemplo, séries de sinais de eletrocardiograma (ECG). Também pretende-se investigar outras formas de extração de características (tais como lyapunov e dimensão fractal) e comparar com os resultados obtidos via wavelet em termos de performance de classificação. Além disso, vislumbra-se a possibilidade de empregar técnicas de seleção de características, como por exemplo, as baseadas em algoritmo genético, para encontrar quais características são mais adequadas para determinado classificador.

REFERENCIAS BIBLIOGRÁFICAS

ALLWEIN, E. L.; SCHAPIRE, R. E.; SINGER, Y.. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1, p. 113-141, 2000.

BROCKWELL, P. J.; DAVIS, R. A.. Introduction to Time Series and Forecasting - 2nd ed., Springer-Verlag New York, 2002.

CALDERBANK, R. C.; DAUBECHIES, I.; SWELDENS, W.; YEO, B., Losless Image Compression using Integer to Integer Wavelet Transforms, In: International Conference on Image Processing (ICIP) Image Processing (ICIP), v. 1, p. 596-599, 1997.

CHAGAS, S. L.; EISENCRAFT, M. ; LIMA, C. A. M. . Relevance Vector Machine Applied to EEG Signals. In: XXVI Simpósio Brasileiro de Telecomunicações, 2008, Rio de Janeiro. Anais do XXVI Simpósio Brasileiro de Telecomunicações, 2008. p. 1-6.

CHERKASSKY, V.; MA, Y. Practical selection of SVM parameters and noise estimation for SVM regression. Neurocomputing vol. 17, no 1, 113-126, 2004.

CORTES, C.; VAPNIK, V.. Support-vector Networks. Machine Learning, vol. 20, pp. 273-207, 1995.

CRISTIANINI, N., SHAWE-TAYLOR, J.. An Introduction to Support Vector Machines, Cambridge University Press, 2000.

DIETTERICH, T. G.; BAKIRI, G.. Solving Multiclass Learning Problems via Error-Correcting Output Codes. Journal of Artificial Intelligence Research 2, p. 263-286, 1995.

DING, C. H. Q.; DUBCHAK, I.. Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics, 17, 349-358, 2001.

EGAN, J. P.. Signal Detection Theory and ROC Analysis, Series in Cognitition and Perception. New York: Academic Press, 1975.

ELGER, C. E.; RIEKE, C.; MORMANN, F.; LEHNERTZ, K.; DAVID, P.; ANDRZEJAK, R. G.. Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E, vol. 64, n. 6, Article ID 061907, 8 pages, 2001.

FLETCHER, R.. Practical Methods of Optimization, Second Edition, Chichester: John Wiley & Sons, Inc, 1987.

FLEURY. Manual de Neurodiagnósticos: Eletrencefalografia. São Paulo, 2007. Disponível em:<http://www.fleury.com.br/Medicos/SaudeEmDia/ManualNeuro/pages/Eletrencefalografia.aspx>. Acesso em: 30 mai. 2009.

FRIEDMAN, J. H.. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, Stanford, CA, 1996.

FUNG, G.; MANGASARIAN, O. L.. Proximal Support Vector Machine Classifiers. Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 77-86, 2001.

GUNN, S.. Support Vector Machine for Classification and Regression. Image Speech & Intelligent Systems Group, Technical Report ISIS-1-98, University of Southampton, Nov. 1998.

JASPER, H. H. The ten-twenty electrode system of the International Federation. Electroencephalogr Clin Neurophysiol 1958; 10: 371-3.

KNERR, S.; PERSONNAZ, L.; DREYFUS, G.. Single-layer learning revisited: A stepwise procedure for building and training a neural network. In: SOULIE, F. F., HERAULT, J.. Neurocomputing: Algorithms, Architectures and Applications, p. 41-50. Berlin: Springer-Verlag, 1990.

KREßEL, U. H.-G.. Pairwise classification and support vector machines. In: SCHÖLKOPF, B., BURGES, C., SMOLA, A. (editores). Advances in Kernel Methods: Support Vector Learning, chapter 15, p. 255-268. Cambridge: MIT Press, 1999.

LATHI, B. P.; DING, Z. . Modern Digital and Analog Communication Systems - 4rd Ed.. Oxford University Press, Inc., 2009.

LEE, Y.J.; MANGASARIAN, O. L.. SSVM: A Smooth Support Vector Machine. Computational Optimization and Applications, p. 5-22, 2001.

LEHNERTZ, K.. Non-linear time series analysis of intracranial EEG recordings in patients with epilepsy-an overview. International Journal of Psychophysiology, vol. 34, n. 1, p. 45-52, 1999.

LIMA, C. A. M.; COELHO, A. L. V.; CHAGAS, S. L.. Automatic EEG signal classification for epilepsy diagnosis with Relevance Vector Machines. Expert Systems with Applications, 2009.

MANGASARIAN, O. L.; MUSICANT, D. R.. Lagragian Support Vector Machines. Journal of Machine Learning Research 1, p. 161-177, 2001.

OKANDAN, M.; KARA, S.. Atrial fibrillation classification with artificial neural networks. Pattern Recognition 40, p. 2967 - 2973, 2007.

OPPENHEIN, A. V., SCHAFER, R. W.. Discrete-Time Signal Processing. Prentice Hall, Inc., 3rd Edition, 2009.

POLLOCK, D.S.G.. A Hanndbook of Time-Series Analysis, Signal Processing and Dynamics. Academic Press, 1999.

REVETT, K.; JAHANKHANI, P.; KODOGIANNIS, V.. EEG Signal Classification Using Wavelet Feature Extraction and Neural Networks. IEEE John Vincent Atanasoff 2006 International Symposium on Volume, p. 120 - 124, 2006.

SABETI, M.; BOOSTANI, R.; KATEBI, S. D.; PRICE, G. W.. Selection of relevant features for EEG signal classification of schizophrenic patients. Biomedical Signal Processing and Control 2, p. 122 - 134, 2007.

SEMMLOW, J. L.. Biosignal and Biomedical Image Processing: MATLAB - Based Applications. Marcel Dekker, Inc., 2004.

SPACKMAN, K. A.. Signal detection theory: Valuable tools for evaluating inductive learning., Proc. Sixth International Workshop on Machine Learning, p. 160-163, 1989.

SUBASI, A.. EEG signal classification using wavelet feature extraction and a mixture of expert model. Expert Systems with Applications 32, p. 1084-1093, 2007.

SUYKENS, J.A.K.; VANDEWALLE, J.. Multiclass Least Squares Support Vector Machines. International Joint Conference on Neural Networks IJCNN'99, Washington DC, 1999.

SWETS, J.. Measuring the accuracy of diagnostic systems., Science, 240, p. 1285-1293, 1988.

TIPPING, M. E.. The Relevance Vector Machine. Advances in Neural Information Processing Systems 12, Solla, S. A.; Leen, T. K.; Muller, K.-R. (editores). Cambridge: MIT Press, p. 652-658, 2000.

TIPPING, M. E.. Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research 1, p. 211-244, 2001.

ÜBEYLI, E. D.. Wavelet/mixture of experts network structure for EEG signals classification. Expert Systems with Applications 34, p. 1954-1962, 2008.

ÜBEYLI, E. D.; GÜLER, I.. A modified mixture of experts network structure for ECG beats classification with diverse features. Engineering Applications of Artificial Intelligence 18, p. 845-856, 2005.

ÜBEYLI, E. D.; GÜLER, I.. Multiclass Support Vector Machines for EEG-Signals Classification. IEEE Transactions on Information Technology in Biomedicine, vol. 11, p. 117 - 126, 2007.

VAN GESTEL, T.; SUYKENS, J. A. K.; BAESENS, B.; VIAENE, S.; VANTHIENEN, J.; DEDENE, G.; DE MOOR, B.; VANDEWALLE, J.. Benchmarking Least Squares Support Vector Machine Classifiers. Machine Learning, v.54 n.1, p. 5-32, 2004.

VAPNIK, V. N.. The Nature of Statistical Learning Theory, Springer Verlag, 1995.

WIDJAJA, I.; LEOW, W.K.; FANG-CHENG Wu. Identifying painters from color profiles of skin patches in painting images, Image Processing - International Conference on Volume 1, p. I - 845-848, 2003.

XIONG, J.; HU, T.; LI, G.; PENG, H.. A Comparative Study of Three Smooth SVM Classifiers. The Sixth World Congress on Intelligent Control and Automation, 2006 (WCICA 2006), vol. 2, p. 5962-5966, 2006.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo