



**Ana Rita Assunção
Teixeira**

**Técnicas Não Lineares Baseadas em Componentes
Principais no Estudo de Séries Temporais**



**Ana Rita Assunção
Teixeira**

**Técnicas Não Lineares Baseadas em Componentes
Principais no Estudo de Séries Temporais**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Doutora Ana Maria Perfeito Tomé, Professora Associada do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

Aos meus pais e à Daisy.

o júri

Presidente

Prof. Doutor Francisco António Cardoso Vaz
Professor Catedrático da Universidade de Aveiro

Vogais

Prof. Doutor António Martins da Silva
Professor Catedrático do Instituto de Ciências Biomédicas Abel Salazar

Prof. Doutor André Teixeira Puga
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Doutora Ana Maria Perfeito Tomé
Professora Associada da Universidade de Aveiro

agradecimentos

A execução deste trabalho foi possível devido à colaboração de algumas pessoas e instituições a quem eu gostaria de agradecer especialmente,

À minha orientadora, Prof. Dra. Ana Maria Tomé, não só pelo apoio, interesse e incansável ajuda na execução deste trabalho, mas sobretudo pela sua amizade e paciência durante todo este período.

Aos meus pais, avó e tio Fernando, pela compreensão, dedicação e amor.

Ao Márcio, por todo o carinho, companheirismo e ajuda prestada ao longo deste percurso.

À Daisy, por todos os momentos.

Ao Prof. Dr. António Martins da Silva, pela inteira disponibilidade e espírito crítico na avaliação dos resultados obtidos.

Aos meus amigos do IEETA, em especial ao Samuel, à Liliana e ao Mário, pelo incentivo e apoio.

Aos Técnicos do Hospital Geral de Santo António, em especial à Técnica Mónica Quintas, pela amizade e ajuda na aquisição e selecção dos sinais EEG.

Ao Prof. Dr. Elmar Lang, pelas conversas e sugestões dadas na interpretação dos resultados.

Aos, meus colegas da Universidade de Regensburg do departamento de Biofísica, em especial ao Peter Gruber, ao Kurt Stadlthanner e ao Mathias Böhm, pelas sugestões dadas para a execução deste trabalho.

Aos meus amigos, em especial à Joana Campos, à Ana Helena e à Rita Simões.

E a todos, que de um modo geral contribuíram de forma directa ou indirecta para a realização deste trabalho.

palavras-chave

Análise de Componentes Principais (PCA), Kernel PCA, *Singular Spectrum Analysis* (SSA), Séries Temporais, Electroencefalograma (EEG)

resumo

Este trabalho teve como objectivo estudar técnicas não lineares para a eliminação de ruído em séries temporais. O estudo efectuado baseou-se nos algoritmos SSA e KPCA. É apresentado um novo algoritmo, designado por Local SSA, que representa uma extensão do SSA. O algoritmo KPCA é descrito numa abordagem diferente da apresentada na literatura.

Os algoritmos foram aplicados a sinais artificiais para estudar a influência dos parâmetros na performance dos mesmos.

Foi efectuado um estudo preliminar da aplicação destes algoritmos a sinais EEG para eliminação de artefactos, nomeadamente, do sinal EOG.

keywords

Principal Component Analysis (PCA), Kernel PCA, Singular Spectrum Analysis (SSA), Time Series, Electroencephalogram (EEG)

abstract

The main goal of this work was to study non linear techniques to remove noise in time series. The study was based on Singular Spectrum Analysis (SSA) and Kernel Principal Component Analysis (KPCA) algorithms.

A new algorithm is presented, named as Local SSA, which consists on extension of the SSA. KPCA algorithm is described in a different approach from the one presented in the literature.

The performance of the algorithms, with distinct parameters, was studied using artificial signals. A preliminary study was carried out, applying these algorithms to EEG signals in order to remove high amplitude artefacts like the interference of the EOG signal.

Índice

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Estrutura da dissertação | 2 |
| 2 | Redução de Ruído em Séries Temporais | 5 |
| 2.1 | Introdução | 5 |
| 2.2 | <i>Singular Spectrum Analysis</i> | 5 |
| 2.3 | Local SSA | 6 |
| 2.3.1 | <i>Embedding</i> | 7 |
| 2.3.2 | Agrupamento (<i>Clustering</i>) | 8 |
| 2.3.3 | Decomposições Locais | 8 |
| 2.3.4 | Agrupamento das Matrizes Elementares | 9 |
| 2.3.5 | Desagrupamento (<i>Unclustering</i>) | 9 |
| 2.3.6 | Diagonalização da matriz de trajectória | 9 |
| 2.4 | MSSA | 10 |
| 2.5 | Seleção dos parâmetros do algoritmo | 11 |
| 2.6 | Caracterização dos sinais de teste | 12 |
| 2.7 | Influência dos parâmetros com SNR elevado | 13 |
| 2.7.1 | Aplicação do critério MDL | 14 |
| 2.7.2 | Seleção de um número de direcções fixo | 16 |
| 2.8 | Influência dos parâmetros com SNR baixo | 17 |
| 2.8.1 | Aplicação do critério MDL | 18 |
| 2.8.2 | Seleção de um número de direcções fixo | 20 |
| 2.9 | Validação dos parâmetros | 21 |
| 2.10 | Conclusões | 23 |
| 3 | Kernel PCA | 25 |
| 3.1 | Introdução | 25 |
| 3.2 | Mapeamento dos Dados e Matriz de Produtos Internos | 25 |
| 3.3 | Funções de Kernel | 28 |
| 3.4 | Algoritmo Kernel PCA | 29 |
| 3.4.1 | Matriz de Kernel Centrada | 30 |
| 3.4.2 | Decomposição da matriz de Kernel | 31 |
| 3.4.3 | Reconstrução no espaço Característico | 33 |

| | | |
|----------|--|-----------|
| 3.5 | Reconstrução no espaço de entrada | 33 |
| 3.6 | Método iterativo | 34 |
| 3.7 | Método das distâncias | 36 |
| 3.7.1 | Distância no Espaço Característico | 38 |
| 3.7.2 | Distância no Espaço de Entrada | 40 |
| 3.7.3 | Relação das Distâncias | 41 |
| 3.8 | Resultados | 43 |
| 3.8.1 | Comparação dos algoritmos KPCA e SSA | 43 |
| 3.8.2 | Comparação gráfica dos algoritmos KPCA e Local SSA | 46 |
| 3.9 | Seleccção do Parâmetro RBF | 49 |
| 3.10 | Conclusões | 49 |
| 4 | Dimensão do Sub-espaço do Sinal | 53 |
| 4.1 | Formulação do Problema | 53 |
| 4.2 | Critérios | 54 |
| 4.2.1 | Variância Explicada | 54 |
| 4.2.2 | MDL e AIC | 54 |
| 4.3 | Resultados | 59 |
| 4.3.1 | Comparação global dos critérios MDL, AIC e VE | 59 |
| 4.3.2 | Aplicação dos critérios ao MSSA | 60 |
| 4.4 | Aplicação do critério MDL ao SSA | 61 |
| 4.5 | Conclusões | 62 |
| 5 | Aplicação dos algoritmos a sinais EEG | 69 |
| 5.1 | Introdução | 69 |
| 5.2 | O Electroencefalograma | 69 |
| 5.3 | Electrooculograma | 70 |
| 5.3.1 | Motivação | 70 |
| 5.4 | Método de Registo do sinal EEG | 72 |
| 5.5 | Aplicação dos algoritmos Local SSA ao sinal EEG | 73 |
| 5.5.1 | Parâmetros do algoritmo | 73 |
| 5.5.2 | Extracção dos movimentos oculares | 73 |
| 5.5.3 | Influência do número de <i>clusters</i> | 75 |
| 5.5.4 | Quantificação dos resultados | 77 |
| 5.6 | Aplicação do algoritmo Kernel PCA ao sinal EEG | 79 |
| 5.6.1 | Parâmetros do algoritmo | 79 |
| 5.6.2 | Extracção dos movimentos oculares | 80 |
| 5.7 | Aplicação do algoritmo Local SSA a outros artefactos | 81 |
| 5.8 | Conclusões | 83 |
| 6 | Conclusões e Trabalho Futuro | 87 |
| A | Decomposição em Valores Singulares - SVD | 95 |

| | |
|---|----|
| B Pré-Imagem - método das distâncias | 97 |
| C Parâmetros dos algoritmos SSA e Local SSA | 99 |

Lista de Figuras

| | | |
|------|---|----|
| 2.1 | Grupo de sinais artificiais utilizados: sinal sinusoidal, sinal <i>funny</i> e sinal dente-de-serra representado no domínio do tempo e no domínio da frequência. | 13 |
| 2.2 | Grupo de sinais artificiais da figura 2.1 com um SNR=20dB. | 14 |
| 2.3 | Grupo de sinais artificiais da figura 2.1 com um SNR=5dB. | 14 |
| 2.4 | Erro quadrático médio centrado entre o sinal dente-de-serra original e o sinal dente-de-serra reconstruído pelo algoritmo SSA, considerando SNR=20dB. Foram considerados 5 períodos diferentes ($P = 15, 25, 35, 45$ e 55) para o sinal dente-de-serra e uma janela de dimensão M , que varia entre $1 \leq M \leq 150$. . . | 15 |
| 2.5 | Sinais extraídos no domínio do tempo (SNR=20dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 16 |
| 2.6 | Sinais extraídos no domínio da frequência (SNR=20dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 16 |
| 2.7 | Sinais extraídos no domínio do tempo (SNR=20dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 17 |
| 2.8 | Sinais extraídos no domínio da frequência (SNR=20dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 17 |
| 2.9 | Sinais extraídos no domínio do tempo (SNR=20dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 18 |
| 2.10 | Sinais extraídos no domínio da frequência (SNR=20dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 18 |
| 2.11 | Sinais extraídos no domínio do tempo (SNR=5dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 19 |
| 2.12 | Sinais extraídos no domínio da frequência (SNR=5dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 19 |

| | | |
|------|---|----|
| 2.13 | Sinais extraídos no domínio do tempo (SNR=5dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 20 |
| 2.14 | Sinais extraídos no domínio da frequência (SNR=5dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 20 |
| 2.15 | Sinais extraídos no domínio do tempo (SNR=5dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 21 |
| 2.16 | Sinais extraídos no domínio da frequência (SNR=5dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$ | 21 |
| 2.17 | Função Densidade de Probabilidade do número de clusters que minimiza o valor da kurtosis do erro usando SSA e Local SSA. | 23 |
| 3.1 | Esquema adaptado de [1]. Conceptualmente, o KPCA efectua um mapeamento não linear aos dados $\phi(\mathbf{x})$ para projectar os vectores de entrada num espaço de maior dimensão \mathfrak{S} , (1). Posteriormente, é aplicado um PCA linear aos dados do espaço característico, obtendo um espaço de menor dimensão KPCA (2). Para reconstruir um vector no espaço de entrada, a representação KPCA é projectada no espaço característico (3) e é efectuado um mapeamento $\phi(\mathbf{x})$ inverso (4) | 30 |
| 3.2 | Representação esquemática do Método Iterativo (<i>Fixed-Point</i>) para encontrar a pré-imagem no algoritmo KPCA. | 34 |
| 3.3 | Representação esquemática do Método das Distâncias para encontrar a pré-imagem no algoritmo KPCA. | 38 |
| 3.4 | Representação do sinal sinusoidal e do sinal sinusoidal adicionado de ruído gaussiano (SNR=10dB) a duas dimensões. | 45 |
| 3.5 | Representação dos 30 primeiros valores próprios normalizados, da matriz de kernel, dos dados da figura 3.4 (b). | 45 |
| 3.6 | Reconstrução do sinal sinusoidal a duas dimensões adicionado com ruído (SNR=10dB) pelo método KPCA, considerando o algoritmo das distâncias no cálculo das pré-imagens. O número de direcções seleccionadas da matriz de kernel variou entre $k = 1$ até $k = 4$ | 46 |
| 3.7 | Comparação gráfica entre o algoritmo KPCA e o algoritmo SSA na reconstrução de uma sinusóide a duas dimensões com um SNR=20dB. São ilustradas quatro situações a duas dimensões:(a) reconstrução pelo algoritmo KPCA- método das distâncias, (b) agrupamento dos dados pelo algoritmo k -means, (c) aplicação do SVD a cada agrupamento - Local SVD e (d) diagonalização da matriz de trajectória - SSA. | 47 |

| | | |
|-----|---|----|
| 3.8 | Comparação gráfica entre o algoritmo KPCA e o algoritmo SSA na reconstrução do mapa de Hénon com um SNR=20dB. São ilustradas 4 situações a duas dimensões:(a) mapa de Hénon, (b)mapa de Hénon com ruído gaussiano (SNR=20dB), (c) reconstrução pelo algoritmo Local SSA, (d) reconstrução pelo algoritmo KPCA- método das distâncias. | 48 |
| 3.9 | Reconstrução de um sinal dente-de-serra pelo algoritmo KPCA, usando o método das distâncias e considerando dois níveis de ruído: (a) Sinal dente-de-serra adicionado com SNR=20dB, (b) Gráfico dos valores próprios da matriz do Kernel RBF, considerando $\sigma^2 = 10\sigma_s^2$, (c) Sinal dente-de-serra reconstruído, (d) Sinal dente-de-serra adicionado com SNR=0dB, (e) Gráfico dos valores próprios da matriz do Kernel RBF, considerando $\sigma^2 = 10\sigma_s^2$, (f) sinal dente-de-serra reconstruído, (g) Sinal dente-de-serra adicionado com SNR=0dB, (h) Gráfico dos valores próprios da matriz do Kernel RBF, considerando $\sigma^2 = 2\sigma_s^2$, (f) sinal dente-de-serra reconstruído. | 51 |
| 4.1 | Porcentagem normalizada do número de valores próprios seleccionados pelos critérios MDL, AIC e VE, considerando dois tipos de matrizes de misturas ($H = 3 \times 3$ e $H = 5 \times 3$), fazendo variar o número de atrasos ($M=1,\dots,11$) e o SNR=(none 0 5 10 15 20). | 63 |
| 4.2 | Simulação de 1000 iterações ($H = 3 \times 3$, $SNR = 20$, $M = 5$) e considerando $N = 50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração. | 64 |
| 4.3 | Simulação de 1000 iterações ($H = 5 \times 3$, $SNR = 20$, $M = 5$) e considerando $N=50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração. | 65 |
| 4.4 | Simulação de 1000 iterações ($H = 3 \times 3$, $SNR = 5$, $M = 5$) e considerando $N = 50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração. | 66 |

| | | |
|-----|---|----|
| 4.5 | Simulação de 1000 iterações ($H = 5X3$, $SNR = 5$, $M = 5$) e considerando $N = 50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração. | 67 |
| 4.6 | Os gráficos (a), (b), (c) representam o logaritmo dos valores próprios estimados da matriz de covariância normalizada $ln(\lambda_k)$, versus o número de valores próprios k seleccionados em cada um dos três <i>clusters</i> e os gráficos (d), (e), (f) representam o logaritmo dos valores próprios estimados da matriz de covariância normalizada $ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada um dos três <i>clusters</i> | 68 |
| 4.7 | Os gráficos (a), (b), (c),(d), (e) representam o logaritmo dos valores próprios estimados da matriz de covariância normalizada do sinal sinusoidal $ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada um dos cinco <i>clusters</i> , quando são considerados $M = 11$ atrasos. | 68 |
| 5.1 | Representação dos sinais EEG associados a cada uma das bandas de frequência fundamentais: bandas alfa, beta, teta e delta. | 70 |
| 5.2 | Segmento de dois canais de um sinal EEG, Fp1-Cz e O1-Cz. | 71 |
| 5.3 | Mapa esquemático da posição dos eléctrodos na aquisição do sinal EEG segundo o sistema internacional 10-20. | 72 |
| 5.4 | Canal frontal de um sinal EEG reconstruído pela aplicação do algoritmo Local SSA, considerando 6 <i>clusters</i> pelo critério MDL. No gráfico estão representados : canal frontal do sinal EEG original (1), sinal EOG extraído (2) e canal frontal do sinal EEG corrigido (3). | 74 |
| 5.5 | Canal frontal de um sinal EEG reconstruído pela aplicação do algoritmo Local SSA, considerando 6 <i>clusters</i> e seleccionando um valor próprio em cada <i>cluster</i> . No gráfico estão representados : canal frontal do sinal EEG original (1), sinal EOG extraído (2) e canal frontal do sinal EEG corrigido (3). | 74 |
| 5.6 | Função densidade espectral (dB/Hz) versus Frequência (Hz) do canal frontal do sinal EEG, do canal frontal corrigido pela aplicação do algoritmo SSA e do algoritmo Local SSA, considerando $q = 4$ e $q = 6$ <i>clusters</i> | 75 |
| 5.7 | Representação do logaritmo dos valores próprios $ln(\lambda_k)$ versus o número de valores próprios k de cada <i>cluster</i> quando é aplicado o algoritmo Local SSA ao canal frontal do sinal EEG. (- - -) A tracejado está representada a dimensão seleccionada pelo critério MDL. | 76 |
| 5.8 | Distribuição dos vectores coluna da matriz de trajectória do segmento de dados, considerando $q = 6$ <i>clusters</i> | 77 |

| | | |
|------|--|----|
| 5.9 | Função densidade de probabilidade do número de <i>clusters</i> ($pdf(q)$) versus o número de <i>clusters</i> (q) que minimiza o valor da kurtosis do sinal EEG corrigido pelo algoritmo Local SSA. O número de <i>clusters</i> variou entre $q = 1$ e $q = 7$ e para cada cluster foram efectuadas 1000 simulações. | 77 |
| 5.10 | Representação da energia instantânea do sinal ao longo do tempo para cada uma das bandas de frequência fundamental. Estão representados 4 sinais : \star - Canal frontal do sinal EEG Original ; \square - Canal frontal do sinal EEG corrigido pelo algoritmo Local SSA com $q = 6$ Cluster; \bullet - Canal frontal do sinal EEG corrigido pelo algoritmo Local SSA com $q = 4$ Cluster; ∇ - Canal frontal do sinal EEG corrigido pelo algoritmo SSA. | 78 |
| 5.11 | Aplicação do algoritmo KPCA para extracção dos movimentos oculares. Segmentação do sinal EEG em 4 segmentos de 3 segundos. Representação do canal frontal do sinal EEG original (1), sinal EOG extraído (2) e canal frontal do sinal EEG corrigido (3). | 81 |
| 5.12 | Espectograma do sinal EEG original, do sinal EOG extraído pelo algoritmo KPCA e do sinal EEG corrigido. | 82 |
| 5.13 | Espectograma do canal frontal do sinal EEG original, do sinal EOG extraído pelo algoritmo SSA, considerando 6 <i>clusters</i> e do canal frontal do sinal EEG corrigido. | 83 |
| 5.14 | Representação da energia instantânea ao longo do tempo para a banda de frequência delta: \star - Canal frontal do sinal EEG Original ; \bullet - Canal frontal do sinal EEG corrigido pelo algoritmo Local SSA com 6 <i>cluster</i> ; ...- Canal frontal do sinal EEG corrigido pelo algoritmo KPCA. | 84 |
| 5.15 | Segmento de 10 segundos de um sinal EEG que antecede uma crise de epilepsia. | 84 |
| 5.16 | Sinais extraídos de forma automática pela aplicação do algoritmo Local SSA aos canais Fp2, F4, T4, Fp1, F3, O1, F7 e T5. | 85 |
| 5.17 | Segmento de sinal EEG corrigido pela subtracção das componentes da figura 5.16 ao sinal EEG original, figura 5.15. | 85 |

Lista de Tabelas

| | | |
|-----|--|-----|
| 2.1 | Erro quadrático médio entre os sinais originais e reconstruídos apresentados nas figuras 2.11 e 2.13. | 22 |
| 2.2 | Erro quadrático médio entre os sinais originais e reconstruídos apresentados na figura 2.15. | 22 |
| 3.1 | Matriz de kernel | 27 |
| 4.1 | Erro médio absoluto entre a variância do ruído e a variância dos valores próprios desprezados pelo critério MDL, considerando uma matriz de mistura 5×3 . . . | 61 |
| 5.1 | Relação entre a dimensão do subespaço k e o número de amostras em cada <i>cluster</i> , da aplicação do algoritmo SSA e do algoritmo Local SSA ao canal frontal do sinal EEG. | 76 |
| 5.2 | Parâmetros do algoritmo KPCA usados em cada um dos segmentos do sinal EEG, considerando $M = 21$ atrasos: número de direcções seleccionadas na matriz de kernel, k , e o valor do parâmetro do kernel RBF. | 79 |
| 5.3 | Parâmetros do algoritmo KPCA usados em cada um dos segmentos do sinal EEG, considerando $M = 41$ atrasos: número de direcções seleccionadas na matriz de kernel, k , e o valor do parâmetro do kernel RBF. | 79 |
| 5.4 | Número de <i>clusters</i> utilizado pelo algoritmo Local SSA para cada canal processado do sinal da figura 5.15. | 84 |
| C.1 | Relação entre a dimensão do subespaço k e o número de amostras em cada <i>cluster</i> da aplicação dos algoritmos SSA e Local SSA ao sinal sinusoidal . . . | 99 |
| C.2 | Relação entre a dimensão do subespaço k e o número de amostras em cada <i>cluster</i> da aplicação dos algoritmos SSA e Local SSA ao sinal <i>funny</i> | 100 |
| C.3 | Relação entre a dimensão do subespaço k e o número de amostras em cada <i>cluster</i> da aplicação dos algoritmos SSA e Local SSA ao sinal dente-de-serra . . | 100 |

Lista de Algoritmos

| | | |
|----|---|----|
| 1 | Mapeamento dos dados | 7 |
| 2 | Clustering | 8 |
| 3 | Decomposições Locais | 10 |
| 4 | Diagonalização da Matriz de Trajectória | 11 |
| 5 | Matriz de Kernel Centrada | 31 |
| 6 | KPCA: Projecções nas direcções principais | 32 |
| 7 | Pré-imagem: Método Iterativo | 37 |
| 8 | Distância no Espaço Característico | 39 |
| 9 | Distância no Espaço Entrada | 42 |
| 10 | Cálculo da pré-imagem | 44 |
| 11 | Variância Explicada | 55 |
| 12 | MDL | 57 |
| 13 | AIC | 58 |

Glossário

AIC *Akaike Information Criterion.*

BIC *Bayesian Information Criterion.*

BSS *Blind Source Separation.*

dB Decibel.

ECG Electrocardiograma.

EEG Electroencefalograma.

EMG Electromiograma.

EOG Electrooculograma.

ICA *Independent Component Analysis.*

ITC *Information Theoretic Criteria.*

KPCA *Kernel Principal Component Analysis.*

MDL *Minimum Description Length.*

MSE *Mean Square Error.*

MSSA *Multiple Singular Spectrum Analysis.*

NMR *Nuclear Magnetic Resonance.*

PCA *Principal Component Analysis.*

PDF *Probability Density Function.*

PSD *Power Spectral Density.*

RBF *Radial Basis Function.*

SNR *Signal to Noise Ratio.*

SSA *Singular Spectrum Analysis.*

SVD *Singular Value Decomposition.*

VE Variância Explicada.

Capítulo 1

Introdução

As técnicas de separação em componentes independentes têm sido aplicadas a vários tipos de sinais, (telecomunicações, biomédicos, multimédia, etc) com dois objectivos diferentes: eliminação do ruído inerente ao processo de aquisição ou separação de sinais registados simultaneamente, mas com proveniência diferente.

Em qualquer um dos casos trata-se de extrair sinais de misturas (lineares ou não-lineares) sem o conhecimento a priori dos sinais originais (sinais fonte ou componentes independentes) e do processo de mistura. Na literatura são descritos diferentes métodos para resolver este problema, nomeadamente, métodos baseados na decomposição generalizada em valores/vectores próprios. Nesta família de algoritmos, com sinais misturados, são calculadas matrizes (de correlação, de covariância, de cumulante) que são simultaneamente diagonalizadas (AMUSE, EFOBI) [2] ou aproximadamente diagonalizadas (SOBI, JADE) [2]. Na diagonalização simultânea reconhece-se que a eficácia do método é bastante dependente do nível de ruído existente no processo de identificação do número de componentes. Consequentemente, a implementação destes métodos é efectuada em dois passos baseados na decomposição standard em vectores e valores próprios [3]. O primeiro passo, usualmente designado por "*whitening* ou *sphering*" dos dados, tem sido implementado utilizando estratégias diversas, com o objectivo de minimizar os efeitos do ruído. Na forma mais simples, o primeiro passo do algoritmo é semelhante a uma decomposição em componentes principais, *Principal Component Analysis* (PCA).

Neste trabalho foram estudadas técnicas não lineares para a eliminação de ruído a serem incorporadas futuramente, neste primeiro passo do algoritmo.

As técnicas não lineares consistem na projecção dos dados num espaço de dimensão elevada, utilizando matrizes de trajectória definidas pelo método *Singular Spectrum Analysis* (SSA) [4] ou utilizando funções de kernel, como é descrito pelos métodos *Kernel Principal Component Analysis* (KPCA) [5].

Foi introduzido um novo algoritmo designado por Local SSA, que consiste no algoritmo SSA adicionado de dois novos passos: *clustering* e *unclustering*. Este novo algoritmo tem como objectivo agrupar a matriz de trajectória por diferentes gamas de amplitude para a

decomposição em valores próprios e vectores próprios de forma local - Local SSA. Após as projecções dos dados num espaço de dimensão superior é necessário escolher as componentes relacionadas com o ruído, bem como as componentes relacionadas com os sinais. Como é verificado pelo método PCA, os valores próprios da matriz de correlação dos dados são um factor relevante na determinação das componentes associadas ao ruído. Para seleccionar os valores próprios da matriz de trajectória, foram introduzidos os seguintes critérios: *Minimum Description Length* (MDL), *Akaike Information Criterion* (AIC), Variância Explicada (VE) e técnicas de *clustering* [6, 7, 8].

O estudo destes algoritmos foi avaliado com sinais artificiais e a sua eficiência foi comparada com parâmetros de performance frequentemente utilizados, tais como o *Mean Square Error* (MSE), coeficientes de correlação, análise espectral e valor da kurtosis [9].

Os algoritmos foram também aplicados a sinais Electroencefalograma (EEG) com o objectivo de eliminar artefactos provenientes do campo electromagnético induzido pela corrente eléctrica e iluminação, dos movimentos oculares ou ainda provenientes dos eléctrodos ou de movimentos do paciente.

1.1 Estrutura da dissertação

A dissertação está dividida em sete capítulos e três apêndices:

- **Capítulo 1** - *Introdução*

O capítulo inicial apresenta o tema, os objectivos da dissertação e a estrutura da mesma.

- **Capítulo 2** - *Redução de ruído em séries temporais*

É apresentado o algoritmo SSA e uma nova versão deste, designada por Local SSA. São efectuadas algumas experiências para estudar a selecção dos parâmetros dos algoritmos, utilizando sinais artificiais e são apresentadas algumas conclusões.

- **Capítulo 3** - *Kernel PCA*

Neste capítulo é feita uma exposição do algoritmo KPCA, nomeadamente das funções de kernel e de dois métodos para a reconstrução dos dados no espaço de entrada: método das distâncias e método iterativo. Posteriormente, são apresentados alguns resultados do estudo dos parâmetros do algoritmo com sinais artificiais e efectuada uma comparação entre o algoritmo Local SSA e KPCA. Por fim, são apresentadas algumas conclusões.

- **Capítulo 4** - *Dimensão do subespaço do sinal*

São apresentados três critérios para a selecção do número de direcções associadas ao ruído do sinal no subespaço de maior dimensão: MDL, AIC e VE. Posteriormente, são apresentados os resultados das experiências efectuadas com sinais artificiais, para estudar a influência de alguns parâmetros na performance dos critérios e retiradas algumas conclusões.

- **Capítulo 5** - *Aplicações dos algoritmos a sinais EEG*

É efectuada uma introdução sobre os sinais EEG e Electrooculograma (EOG) e o seu método de registo. São apresentados os resultados da aplicação dos algoritmos SSA, Local SSA e KPCA a segmentos de sinais EEG na eliminação de artefactos e apresentadas algumas conclusões.

- **Capítulo 6** - *Conclusões e Trabalho futuro*

Neste capítulo são apresentadas as conclusões gerais do trabalho e algumas propostas de trabalho futuro.

- **Apêndice 1** - *Decomposição em Valores Singulares - Singular Value Decomposition (SVD)*

Neste apêndice é descrita a decomposição de uma matriz em valores e vectores singulares.

- **Apêndice 2** - *Pré-Imagem - método das distâncias*

Descrição de uma metodologia proposta em [10] para resolver o problema da pré-imagem no método das distâncias do algoritmo KPCA.

- **Apêndice 3** - *Resultados SSA*

Descrição dos parâmetros da aplicação dos algoritmos SSA e Local SSA aos sinais artificiais com um *Signal to Noise Ratio* (SNR)=5dB (número de direcções seleccionadas pelo critério MDL e número de amostras por *cluster*).

Capítulo 2

Redução de Ruído em Séries Temporais

2.1 Introdução

A redução de ruído em séries temporais multidimensionais é um problema que pode ser resolvido, aplicando o algoritmo SVD. No entanto, em sinais unidimensionais a utilização deste algoritmo não é viável, devido à dimensão do conjunto de dados.

Uma alternativa à resolução deste problema é mapear os dados num espaço de dimensão superior, através de atrasos sucessivos ao sinal. Sobre o conjunto de dados mapeados é aplicado o algoritmo SVD. Por fim, são calculadas as médias das diagonais da matriz dos dados reconstruídos para obter novamente o sinal unidimensional. Este processo designa-se por SSA. A SSA é uma técnica que se baseia no método SVD para decompor uma série temporal num somatório de várias componentes, num espaço de dimensão superior.

Neste capítulo são descritos detalhadamente todos os passos do método SSA. Posteriormente, é descrito um outro método baseado na SSA com a introdução de dois novos passos, designados por *clustering* e *unclustering*. Este passo tem como objectivo permitir a decomposição da série temporal de forma local, (Local SSA). Depois é apresentada uma extensão do método SSA para análise de séries multidimensionais, (*Multiple Singular Spectrum Analysis* (MSSA)).

Na secção 2.5 são apresentados alguns resultados obtidos com dados artificiais que ilustram a importância e influência de alguns parâmetros dos algoritmos SSA e Local SSA na redução de ruído em séries temporais. Por fim, são apresentadas algumas conclusões.

2.2 *Singular Spectrum Analysis*

A SSA é uma técnica nova de análise de séries temporais para extracção de informação, aplicada em vários tipos de séries, nomeadamente séries climáticas, [11], geográficas [12, 13] e meteorológicas [4], com o objectivo de decompor as séries temporais num somatório de várias

componentes. Estas componentes são habitualmente interpretadas como a tendência da série (quando a componente é muito lenta), oscilações (quando há variações de amplitudes) ou como o ruído.

A análise espectral singular é decomposta em dois passos principais: decomposição e reconstrução.

- **Decomposição**

1. **Mapeamento dos dados - *Embedding***

Transformação de uma sequência $x[n]$ unidimensional para uma sequência multidimensional, através de uma janela de dimensão M , onde $1 < M < N$. O novo conjunto de dados é um sinal multidimensional em que cada vector tem dimensão M , eq. 2.1.

A variável M corresponde ao número de atrasos efectuados ao sinal e é também designada por dimensão da janela.

$$\mathbf{x}[n] = [x[n], x[n-1], x[n-2], \dots, x[n-M+1]]^T \quad (2.1)$$

2. **Decomposição SVD e PCA no espaço de dimensão M .**

No espaço de dimensão M é calculada a matriz de covariância dos dados multidimensionais e a sua decomposição em valores próprios e vectores próprios.

- **Reconstrução**

1. **Agrupamento** A decomposição da matriz de covariância em valores e vectores próprios pode ser escrita como um somatório das submatrizes associadas a cada valor próprio maior que zero, originando uma única matriz de trajectória. Posteriormente, os dados são projectados nas direcções principais e reconstruídos.

2. **Cálculo do sinal unidimensional**

Este passo transforma a matriz de trajectória numa série temporal.

No algoritmo SSA foram introduzidos dois novos passos (*clustering* e *unclustering*) com o objectivo de agrupar os dados e projectá-los localmente nas suas principais direcções [14]. Este novo algoritmo designa-se por Local SSA. Na secção seguinte, estão descritos os passos deste algoritmo.

2.3 Local SSA

O Local SSA é um método que se baseia no tradicional SSA, adicionado de dois novos passos que correspondem ao agrupamento dos dados (*clustering*) e ao respectivo desagrupamento dos mesmos (*unclustering*). Este algoritmo é composto pelos seguintes passos:

1. Decomposição

- Embedding
- Agrupamento (*Clustering*)
- Decomposições Locais

2. Reconstrução

- Agrupamento das decomposições locais
- Desagrupamento (*Unclustering*)
- Diagonalização

Nas subsecções seguintes vão ser descritos detalhadamente todos os passos deste algoritmo.

2.3.1 *Embedding*

Considerando N amostras da sequência unidimensional $x[n] = [x[0], x[1], \dots, x[N-1]]$, a matriz de trajectória X , $M \times (N - M + 1)$ obtida para esta sequência é dada por

$$X = \begin{bmatrix} x[M-1] & x[M] & \cdots & x[N-1] \\ x[M-2] & x[M-1] & \cdots & x[N-2] \\ \vdots & \vdots & \ddots & \vdots \\ x[0] & x[1] & \cdots & x[N-M] \end{bmatrix} \quad (2.2)$$

onde M representa o número de atrasos a efectuar ao sinal. As colunas da matriz de trajectória constituem $K = N - M + 1$ vectores atrasados, tal como foi descrito pela equação 2.1. Assim, o *embedding* corresponde ao mapeamento dos dados originais num espaço de dimensão $M < N$.

Algoritmo 1 Mapeamento dos dados

| | |
|---------|---|
| entrada | N amostras de \mathbf{x} ; <i>Embedding</i> M $X = []$ <i>for</i> $ic = M + 1 : -1 : 1$ $X = [X; \mathbf{x}(ic : end - M - 1 + ic)]$ <i>end</i> |
| saída | Matriz de Trajectória: X |

2.3.2 Agrupamento (*Clustering*)

Este passo corresponde ao agrupamento das colunas da matriz de trajectória. Ao conjunto dos $(N - M + 1)$ vectores colunas da matriz de trajectória é aplicado o algoritmo de *clustering k-means* [15] que agrupa os vectores coluna em q conjuntos, formando q submatrizes de trajectória X^{c_i} , com $i = 1, \dots, q$ [9]. O número de colunas N_{c_i} de cada submatriz de trajectória verifica a condição

$$\sum_{i=1}^q N_{c_i} = N - M + 1 \quad (2.3)$$

Depois de se aplicar o algoritmo de *clustering* aos vectores coluna da matriz X , (eq. 2.2), os índices $k = 1, 2, \dots, N - M + 1$ das colunas da matriz de trajectória são divididos em q conjuntos: c_1, \dots, c_q . Cada submatriz de trajectória X^{c_i} é formada pelas colunas da matriz X , associadas ao índice do conjunto c_i .

Algoritmo 2 Clustering

| | |
|---------|---|
| entrada | Matriz Trajectória X , Número de <i>Clusters</i> q |
| | $[\text{clust } X^{c_i}] = \text{kmeans}^1(X, q), i = 1, \dots, q$ |
| saída | Submatrizes de X : $X^{c_i}, i = 1, \dots, q$ Vector dos índices das colunas de cada submatriz X^{c_i} : Clust |

2.3.3 Decomposições Locais

Para cada submatriz X^{c_i} é calculada a matriz de covariância e efectuada a decomposição em valores e vectores próprios, sendo extraídas as componentes associadas às k principais direcções. Consideremos \mathbf{m}_{c_i} o vector das médias de cada submatriz de trajectória

$$\mathbf{m}_{c_i} = \frac{1}{N_{c_i}} X^{c_i} \mathbf{j}_{N_{c_i}} \quad (2.4)$$

onde $\mathbf{j}_{N_{c_i}} = [1, 1, \dots, 1]^T$ com dimensão N_{c_i} .

A cada *cluster* é-lhe retirado a média:

$$Y = X^{c_i} - \mathbf{m}_{c_i} (\mathbf{j}_{N_{c_i}})^T \quad (2.5)$$

É calculada a matriz de covariância C para cada submatriz de trajectória X^{c_i} , $C = \langle YY^T \rangle$. Posteriormente, é efectuada uma decomposição em valores e vectores próprios de cada matriz

¹Função implementada na *Statistical Pattern Recognition Toolbox for matlab* [16]

de covariância C e calculadas as projecções normalizadas v_l nas principais direcções

$$\mathbf{v}_l = \lambda_l^{-1/2} Y^T \mathbf{u}_l, \quad (l = 1, \dots, d) \quad (2.6)$$

onde $(\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_M \geq 0)$ correspondem aos valores próprios e $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M)$ aos vectores próprios e $d = \max(l | \lambda_l > 0)$.

A decomposição SVD da matriz de trajectória pode ser escrita como

$$Y = Y_1 + Y_2 + \dots + Y_d \quad (2.7)$$

onde $Y_l = (\sqrt{\lambda_l} \mathbf{u}_l) \mathbf{v}_l^T = \mathbf{u}_l \mathbf{u}_l^T Y$ representam as matrizes elementares da reconstrução de cada projecção.

A selecção do número principal de direcções k no espaço de dimensão M pode ser efectuada, considerando diferentes critérios. Estes critérios encontram-se descritos no capítulo 4.

2.3.4 Agrupamento das Matrizes Elementares

Este passo consiste na soma das k matrizes elementares. A submatriz de trajectória reconstruída é dada por:

$$X_r^{c_i} = Y_1 + Y_2 + Y_3 + \dots + Y_k + \mathbf{m}_{c_i} (\mathbf{j}_{N_{c_i}})^T \quad (2.8)$$

É de notar que este procedimento é efectuado para cada uma das matrizes de covariância dos q clusters. Assim, as equações 2.5, 2.6 e 2.7 são aplicadas a cada uma das diferentes submatrizes de trajectória.

2.3.5 Desagrupamento (*Unclustering*)

Depois de reconstruir todas as submatrizes de trajectória nas k principais direcções, é necessário desagrupar os dados (*unclustering*).

Todas as q submatrizes de trajectória $X_r^{c_i}$ vão ser agrupadas numa única matriz X_r , de acordo com os índices do conjunto c_i que lhes corresponde. Assim, X_r vai representar a nova matriz multidimensional de dados de dimensão $M \times (N - M + 1)$.

2.3.6 Diagonalização da matriz de trajectória

A diagonalização corresponde à transformação de uma matriz multidimensional X_r , $M \times (N - M + 1)$, num vector $\hat{\mathbf{x}}_r$, $1 \times N$.

Esta transformação consiste na substituição de todos os elementos de cada diagonal da matriz multidimensional pela sua média, obtendo uma matriz que corresponde à matriz de trajectória. Este é o processo que garante a minimização da norma de Forbenius da diferença entre a matriz original e a matriz de trajectória, tendo em conta todas as soluções possíveis para obter uma

Algoritmo 3 Decomposições Locais

| | |
|---------|---|
| entrada | Matriz dos dados X do cluster c_i Número de direcções principais: k |
| | $[M \ c] = \text{size}(X)$ $Y = X - \text{mean}(X, 2)(\mathbf{1}_{N_c})^T$ onde $j_{N_c} = [1, 1, \dots, 1]^T$ de dimensão c $C = YY^T$ $[U, \Lambda] = \text{eig}(C)$ $Y_l = \mathbf{u}_l \mathbf{u}_l^T Y, \quad l = 1, \dots, M$ Escolha das k principais direcções $X_r = Y_1 + Y_2 + \dots + Y_k + \text{mean}(X, 2)(j_{N_c})^T$ |
| saída | novo conjunto de dados: X_r |

matriz com todas as diagonais iguais [4]. A sequência $\hat{\mathbf{x}}_r$ é constituída pelas médias de todas as diagonais da matriz X_r (ver algoritmo 4).

2.4 MSSA

A análise singular espectral multidimensional representa uma extensão da SSA. Considerando um sinal multidimensional

$$\mathbf{x}[n] = [x_1[n], x_2[n], \dots, x_L[n]]^T \quad (2.9)$$

constituído por L sinais (onde cada sinal é $\mathbf{x}_i[\mathbf{n}] = [x_i[0], x_i[1], \dots, x_i[N-1]]$, $i = 1 \dots L$), é calculada uma matriz de trajectória para cada um deles. A matriz de trajectória X ($LM \times (N - M + 1)$) obtida para a série multidimensional $\mathbf{x}[n]$ é obtida pela concatenação de L matrizes de trajectória :

$$X = [X_1, X_2, \dots, X_L] \quad (2.10)$$

ou seja,

Algoritmo 4 Diagonalização da Matriz de Trajectória

| | |
|---------|--|
| entrada | Matriz de dados X_r , $M \times (N - M + 1)$ |
| | $[M, C]=\text{size}(X)$ $y = []$ for $i=-M+1:C-1$ $y=[y \text{ mean}(\text{diag}(X),i)]$ end |
| saída | Vector \mathbf{y} , $1 \times N$ |

$$X = \begin{bmatrix} x_1[M-1] & x_1[M] & \cdots & x_1[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ x_1[0] & x_1[1] & \cdots & x_1[N-M] \\ x_2[M-1] & x_2[M] & \cdots & x_2[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ x_2[0] & x_2[1] & \cdots & x_2[N-M] \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_L[M-1] & x_L[M] & \cdots & x_L[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ x_L[0] & x_L[1] & \cdots & x_L[N-M] \end{bmatrix} \quad (2.11)$$

em que cada coluna da matriz X é constituída pelos vectores L vectores \mathbf{x}_i . O processamento da matriz X é efectuado tal como é descrito na secção 2.3, excepto a opção descrita em 2.3.6, que deverá ser efectuada por blocos.

São considerados L blocos e as diagonais de cada bloco são substituídas pelas suas médias, obtendo um sinal com dimensão igual à original, $L \times N$.

2.5 Selecção dos parâmetros do algoritmo

O algoritmo Local SSA descrito nas secções anteriores apresenta um conjunto de parâmetros que é necessário escolher: número de *clusters* q e número de atrasos M . Existe um terceiro

parâmetro que consiste no número de direcções k a seleccionar em cada *cluster*. Este parâmetro pode ser inferido pelo critério MDL ou de uma forma constante, seleccionando um número de direcções fixas em cada *cluster*, $k = 1$. O parâmetro k influencia também a performance do algoritmo e está relacionado com os outros parâmetros.

A escolha do número de atrasos para efectuar o *embedding* no algoritmo SSA depende da informação conhecida a priori da série (por exemplo, periodicidade ou frequências fundamentais). A maior dificuldade na escolha da dimensão da janela reside no facto da variação da dimensão da janela influenciar fortemente as capacidades de separação do algoritmo SSA [4]. Contudo, existem alguns princípios gerais assumidos por diversos autores para a selecção da dimensão M da janela, nomeadamente:

- Se a dimensão da janela for aproximadamente metade da dimensão do conjunto de dados ($M \approx \frac{N}{2}$), então é obtida uma decomposição detalhada e uma separação estável da série temporal [4].

Se a dimensão da janela for pequena, a separação das componentes não é tão evidente. Na maior parte das vezes, a escolha de janelas de dimensão elevada (quando $M, K \mapsto \infty$) permite obter melhores resultados.

- Se o sinal for um sinal periódico de banda larga, de período P , a dimensão da janela deve ser superior ao período do sinal, $M > P$ [4].
- Se pretendermos eliminar num sinal componentes associadas a uma determinada banda de frequência, a dimensão da janela escolhida é baseada no menor valor da banda de frequência de interesse, f_L ,

$$M \geq \frac{f_s}{f_L} \quad (2.12)$$

onde f_s representa a frequência do sinal e f_L a frequência mais baixa de interesse [17].

Nas secções seguintes, as experiências efectuadas tiveram como objectivo ilustrar a aplicação dos algoritmos SSA e Local SSA na redução de ruído branco gaussiano aditivo em séries temporais. Além disso, foram efectuadas várias experiências para estudar a influência dos parâmetros (número de *clusters* q , número de atrasos M e número de direcções k) na performance do algoritmo.

2.6 Caracterização dos sinais de teste

A sequência temporal utilizada nestas experiências é dada por:

$$x[n] = y[n] + \nu[n] \quad (2.13)$$

onde $y[n]$ representa um sinal periódico e $\nu[n]$ o ruído aleatório gaussiano adicionado.

Foi considerado um grupo de três sinais periódicos $x[n]$ com diferentes bandas de frequência

fundamentais: sinal de banda estreita (sinusóide), sinal de banda larga (*funny*) e um sinal dente-de-serra cuja energia mais alta se encontra concentrada na banda das baixas frequências (figura 2.1). A este grupo de sinais foi adicionado ruído gaussiano aleatório, tal que, o SNR é dado pela expressão 2.14

$$SNR = 10 \log \left\{ \frac{\sigma_x^2}{\sigma_v^2} \right\} (dB) \quad (2.14)$$

tomasse valores de 20dB e de 5dB. O parâmetro σ_x^2 representa a variância do sinal e o parâmetro σ_v^2 a variância do ruído.

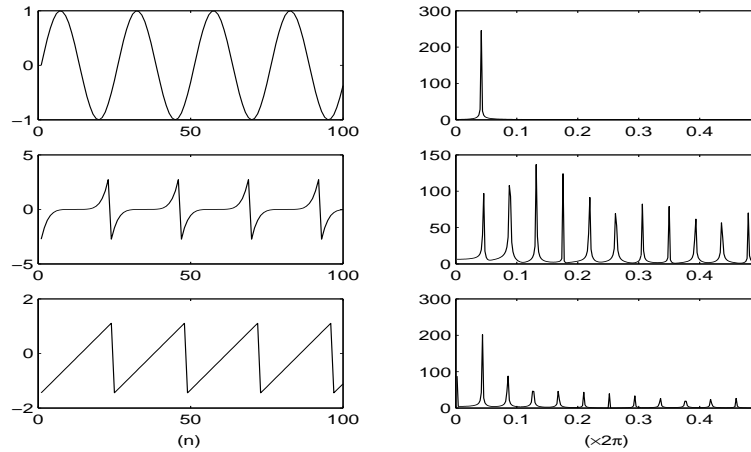


Figura 2.1: Grupo de sinais artificiais utilizados: sinal sinusoidal, sinal *funny* e sinal dente-de-serra representado no domínio do tempo e no domínio da frequência.

Para estudar a influência dos parâmetros do algoritmo, foram efectuadas várias experiências com os sinais da figura 2.1. Dos resultados obtidos, verificou-se que o SNR altera de forma significativa a relação entre os parâmetros. Assim sendo, os resultados apresentados estão divididos, tendo em conta a influência dos parâmetros com um SNR elevado (20dB) (figura 2.2) e um SNR baixo (5dB) (figura 2.3).

2.7 Influência dos parâmetros com SNR elevado

Nesta secção vão ser apresentados alguns resultados que ilustram a influência do número de *clusters*, do número de atrasos e do número de direcções a seleccionar, em sinais com $SNR = 20dB$, considerando o conjunto de dados da figura 2.1. A este conjunto de dados foi aplicado o algoritmo SSA e o algoritmo Local SSA, considerando $q = 3$ e $q = 5$ *clusters* e duas janelas de dimensão $M = 11$ e $M = 36$.

A escolha do número de direcções principais a seleccionar em cada *cluster* foi obtida por dois critérios diferentes:

- selecção do número de direcções de forma automática, pelo critério MDL

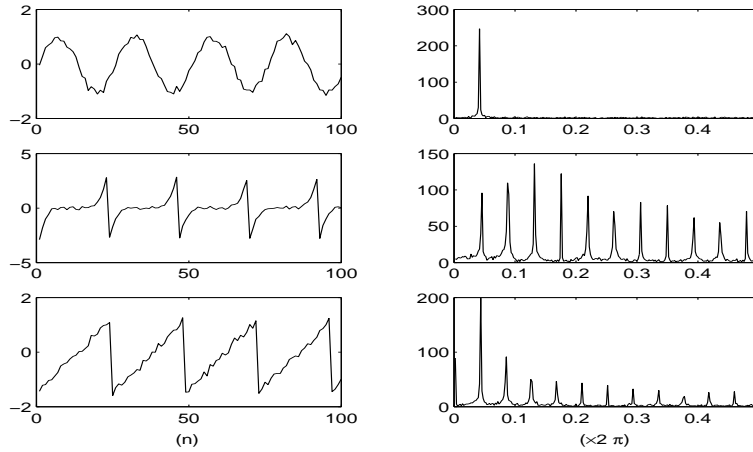


Figura 2.2: Grupo de sinais artificiais da figura 2.1 com um SNR=20dB.

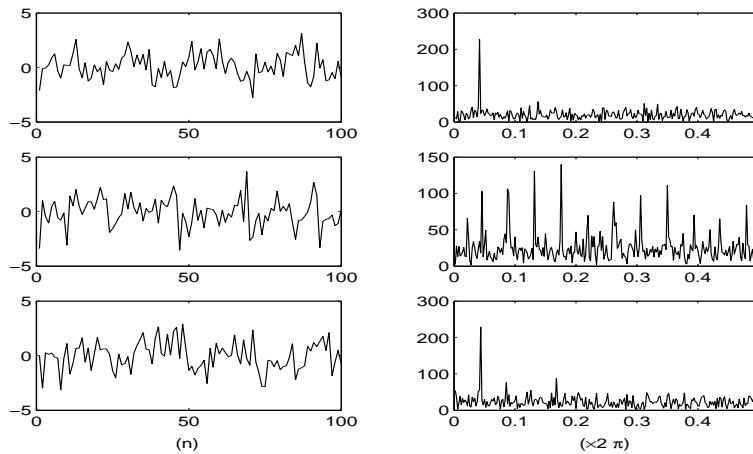


Figura 2.3: Grupo de sinais artificiais da figura 2.1 com um SNR=5dB.

- selecção de um número fixo de direcções ($k = 1$)

2.7.1 Aplicação do critério MDL

Para SNR elevados, a dimensão do *embedding* do algoritmo SSA está relacionada com a periodicidade do sinal [4].

Para ilustrar a influência do *embedding* na redução de ruído gaussiano em sinais periódicos, foi calculado o erro quadrático médio entre o sinal original e o sinal reconstruído pelo algoritmo SSA

$$MSE = \frac{1}{N} \sum_{n=1}^N |y[n] - \hat{y}[n]|^2 \quad (2.15)$$

onde N representa a dimensão do conjunto de dados, $y[n]$ representa o sinal original e o $\hat{y}[n]$ representa o sinal reconstruído.

Foi considerado o sinal dente-de-serra (figura 2.1) com cinco períodos diferentes:

$P = 15, 25, 35, 45$ e 55 e um SNR igual a $20dB$. A dimensão do *embedding* variou entre $M = 1$ e $M = 151$. Para cada M foi calculado o erro quadrático médio centrado (figura 2.4).

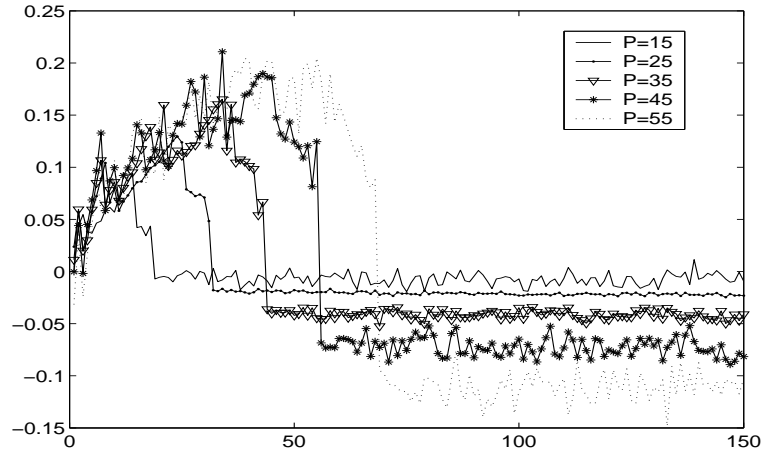


Figura 2.4: Erro quadrático médio centrado entre o sinal dente-de-serra original e o sinal dente-de-serra reconstruído pelo algoritmo SSA, considerando SNR=20dB. Foram considerados 5 períodos diferentes ($P = 15, 25, 35, 45$ e 55) para o sinal dente-de-serra e uma janela de dimensão M , que varia entre $1 \leq M \leq 150$

O gráfico mostra que o erro quadrático médio centrado decresce abruptamente quando a dimensão do *embedding* é superior ao período do sinal. Este resultado é válido para sinais de banda larga, sinal dente-de-serra e sinal *funny* (figuras 2.5, 2.6, 2.7e 2.8, coluna 1). Para sinais de banda estreita (sinal sinusoidal), o mesmo não se verifica. A escolha de um *embedding* inferior ao período do sinal permite obter um sinal no domínio do tempo e da frequência, próximo do sinal original (figuras 2.5 e 2.6).

Note-se que no caso da aplicação do algoritmo SSA ao sinal *funny* com $M = 11$ atrasos, o sinal reconstruído corresponde ao sinal de entrada, uma vez que o critério MDL selecciona todas as direcções do subespaço de dimensão superior, não havendo eliminação do ruído adicionado (figura 2.5).

Quando se aplica o algoritmo Local SSA a sinais de banda larga, perde-se a relação entre a dimensão do *embedding* e a periodicidade do sinal porém a dimensão do *embedding* necessária para a reconstrução do sinal pode ser inferior ao período do mesmo ($M = 11$) (figuras 2.5 e 2.6, coluna 2 e 3).

Se o número de atrasos for superior ao período do sinal ($M = 36$), todos os sinais são reconstruídos em toda a gama de frequência (figuras 2.7 e 2.8).

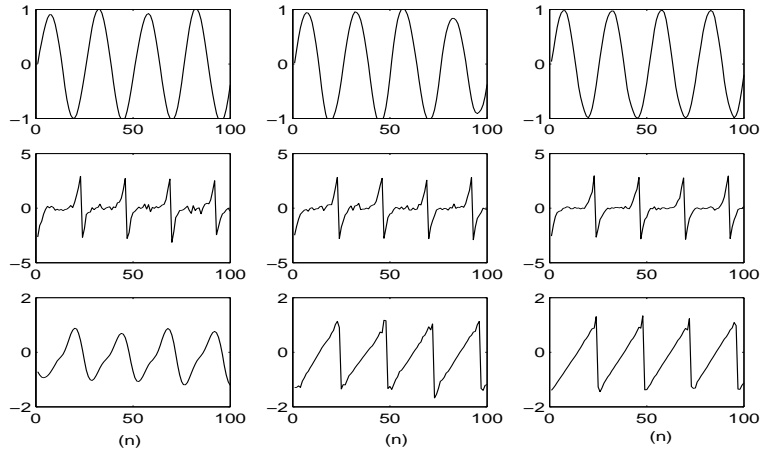


Figura 2.5: Sinais extraídos no domínio do tempo (SNR=20dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

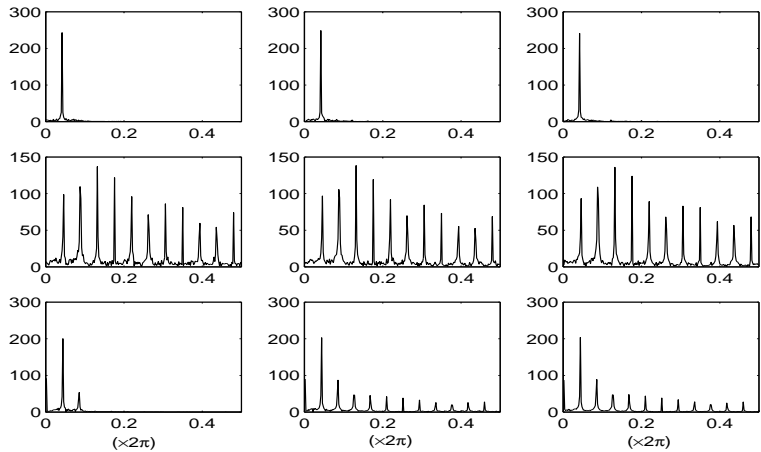


Figura 2.6: Sinais extraídos no domínio da frequência (SNR=20dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

2.7.2 Selecção de um número de direcções fixo

Em alternativa à aplicação do critério MDL, foi efectuada o mesmo estudo da secção anterior, seleccionando em cada *cluster* um número de direcções fixo, neste caso, $k = 1$.

Para esta situação, o aumento da dimensão do *embedding* não altera significativamente os resultados. Nos exemplos apresentados, foi escolhida uma janela de dimensão $M = 11$.

Na aplicação do algoritmo SSA a sinais de banda estreita, a selecção de uma única direcção é suficiente para reconstruir o sinal, como está ilustrado nas figuras 2.9 e 2.10, para o caso do sinal sinusoidal.

Se o sinal for de banda larga, o algoritmo SSA não reconstrói o sinal em todas as suas bandas de frequência (figura 2.9 e 2.10, coluna 1).

O aumento do número de *clusters* origina melhores resultados, pois há uma maior sensibilidade para a reconstrução do sinal em toda a gama de frequência (figuras 2.9 e 2.10, coluna 2 e 3).

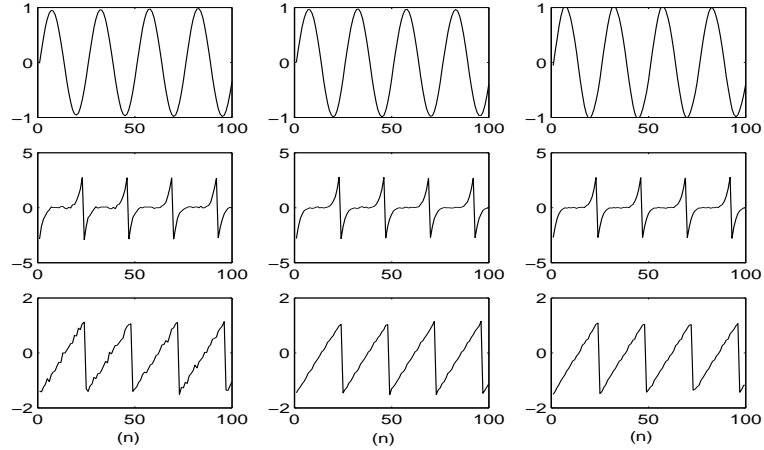


Figura 2.7: Sinais extraídos no domínio do tempo (SNR=20dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

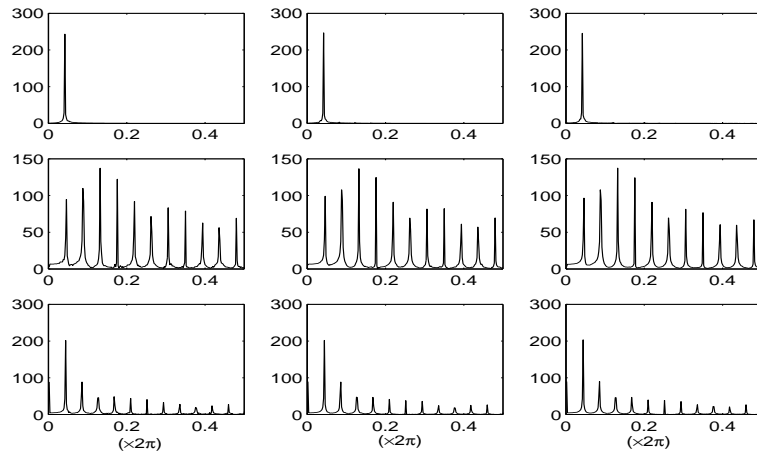


Figura 2.8: Sinais extraídos no domínio da frequência (SNR=20dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

2.8 Influência dos parâmetros com SNR baixo

Nesta secção vão ser apresentados alguns resultados que ilustram a influência do número de *clusters*, do número de atrasos e do número de direcções a seleccionar, em sinais com $SNR = 5dB$. Foi considerado o mesmo conjunto de dados da figura 2.1. A este conjunto de dados foi aplicado o algoritmo SSA e o algoritmo Local SSA, para $q = 3$ e $q = 5$ *clusters* e duas janelas de dimensão $M = 11$ e $M = 36$.

A escolha do número de direcções principais a seleccionar em cada *cluster* foi obtida por dois critérios diferentes:

- selecção do número de direcções de forma automática, pelo critério MDL
- selecção de um número fixo de direcções, $k = 1$

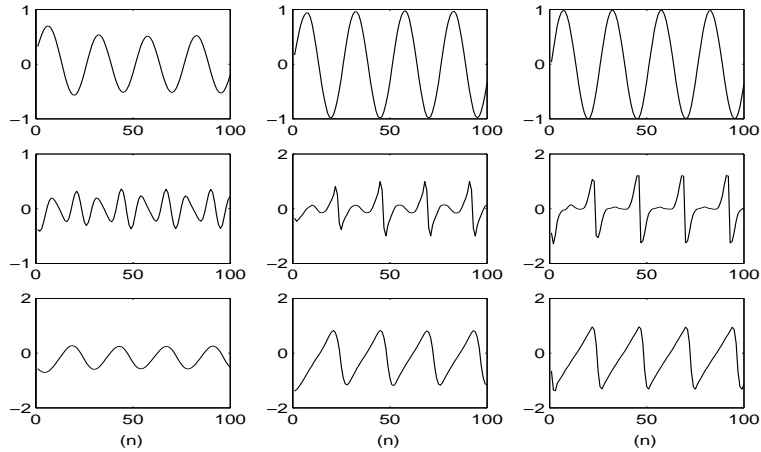


Figura 2.9: Sinais extraídos no domínio do tempo (SNR=20dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

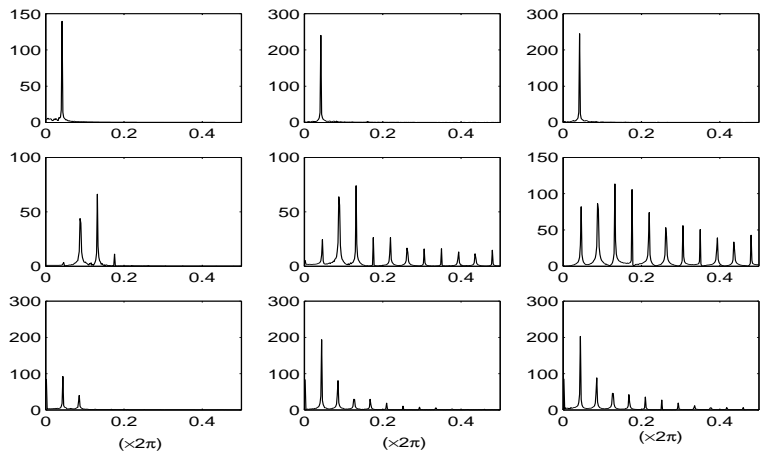


Figura 2.10: Sinais extraídos no domínio da frequência (SNR=20dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

2.8.1 Aplicação do critério MDL

Quando o SNR é baixo, ou seja, quando a amplitude do $\nu[n]$ é muito próxima da amplitude das ondas periódicas, não existe uma separação tão evidente das direcções associadas ao ruído do sinal.

Nesta situação, o algoritmo Local SSA, apresenta melhores resultados que o algoritmo SSA (figuras 2.11 e 2.13).

Com um SNR baixo, a dimensão do *embedding* deve ser superior ao período do sinal, evitando a sobreestimação da dimensão do subespaço por parte do critério MDL, nomeadamente para os sinais de banda larga.

No caso do algoritmo SSA, a relação entre a dimensão do *embedding* e o período do sinal não existe, porque o padrão de periodicidade do sinal é perdido com SNR baixo. Mesmo considerando um *embedding* elevado, o algoritmo SSA não consegue reconstruir os sinais de

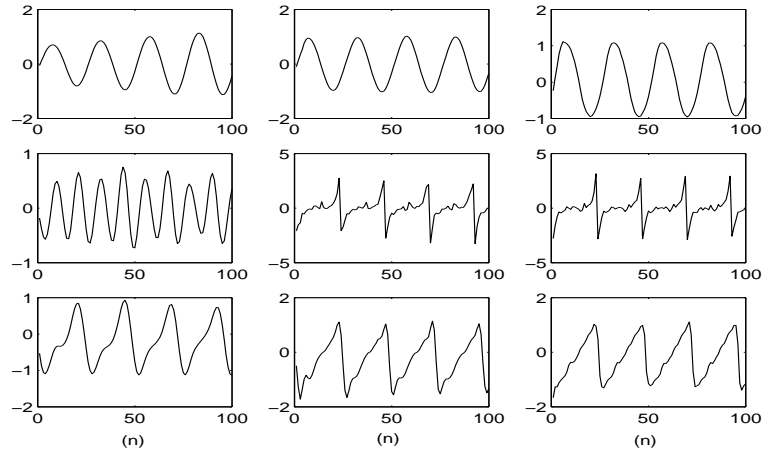


Figura 2.11: Sinais extraídos no domínio do tempo (SNR=5dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

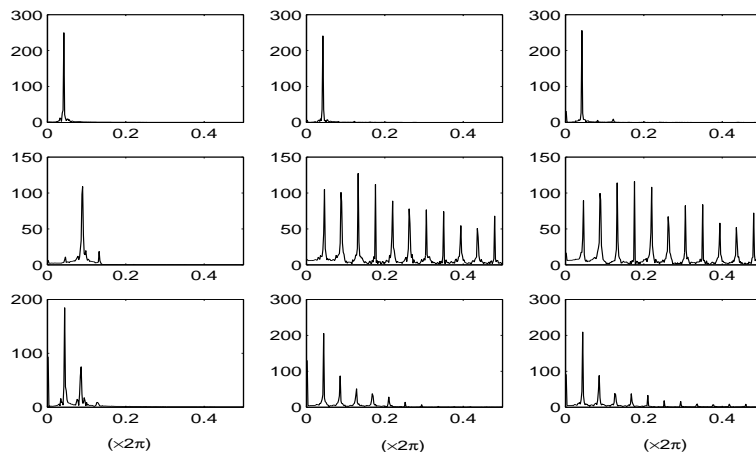


Figura 2.12: Sinais extraídos no domínio da frequência (SNR=5dB) com $M = 36$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

banda larga (figura 2.11 e 2.12, coluna 1). O mesmo não se verifica com os sinais de banda estreita pois *embedding* inferior ao período do sinal ($M = 11$) é suficiente para reconstruir o sinal (figuras 2.13 e 2.14).

No caso do algoritmo Local SSA, existe uma sobreestimação da dimensão do subespaço (figuras 2.13 e 2.14) para o caso do sinal sinusoidal e do sinal dente-de-serra se a dimensão do *embedding* for muito pequena ($M = 11$). Com uma dimensão do *embedding* elevada ($M = 36$) e à medida que o número de *clusters* aumenta, melhor se torna a performance do algoritmo, para os sinais de banda larga: sinal *funny* e sinal dente-de-serra. Observando a figura 2.12, verifica-se que o aumento do número de *clusters* permite reconstruir o sinal sem ruído em toda a gama de frequências.

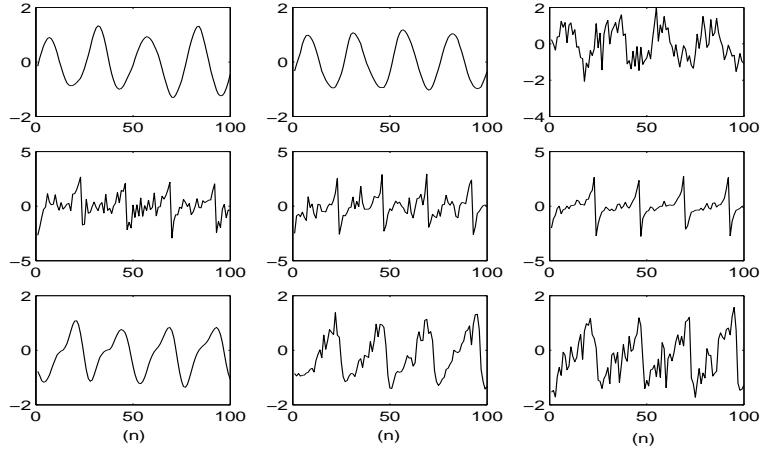


Figura 2.13: Sinais extraídos no domínio do tempo (SNR=5dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

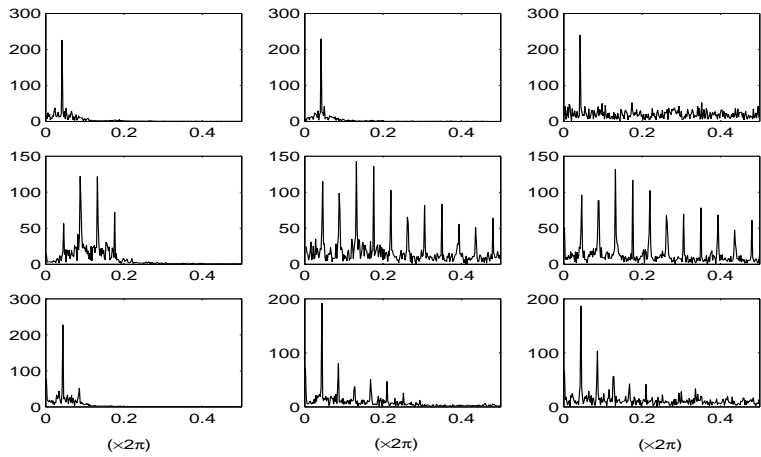


Figura 2.14: Sinais extraídos no domínio da frequência (SNR=5dB) com $M = 11$ atrasos e usando o critério MDL: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

2.8.2 Selecção de um número de direcções fixo

Em alternativa à aplicação do critério MDL, seleccionou-se uma única direcção em cada *cluster*, efectuando as mesmas experiências da secção anterior.

Para esta situação, o aumento da dimensão do *embedding* não altera significativamente os resultados. Nos exemplos apresentados, foi escolhida uma janela de dimensão $M = 11$.

Os algoritmos SSA e Local SSA apresentam resultados muito semelhantes aos apresentados com um SNR=20dB. No caso do SSA, a selecção de uma única direcção para sinais de banda larga não é suficiente para os reconstruir em toda a gama de frequências (figuras 2.15 e 2.16, coluna 2 e 3), enquanto para sinais de banda estreita uma direcção é suficiente (figuras 2.15 e 2.16, coluna 1).

No caso da aplicação do algoritmo Local SSA, o aumento do número de *clusters* permite obter melhores resultados nos sinais de banda larga, devido ao agrupamento do sinal por diferentes

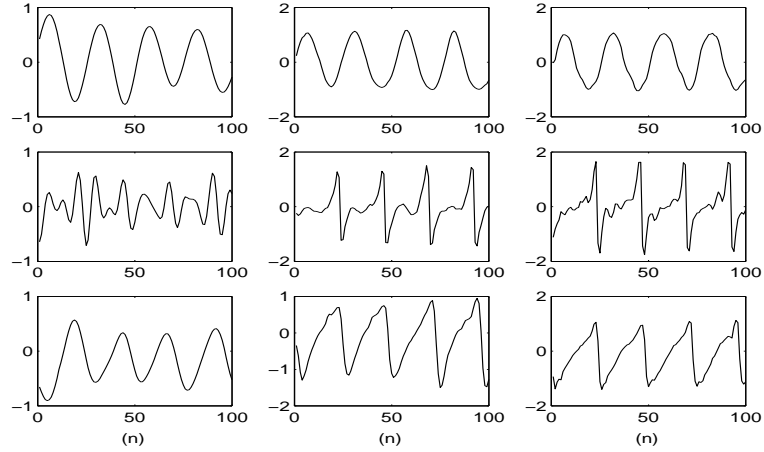


Figura 2.15: Sinais extraídos no domínio do tempo (SNR=5dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

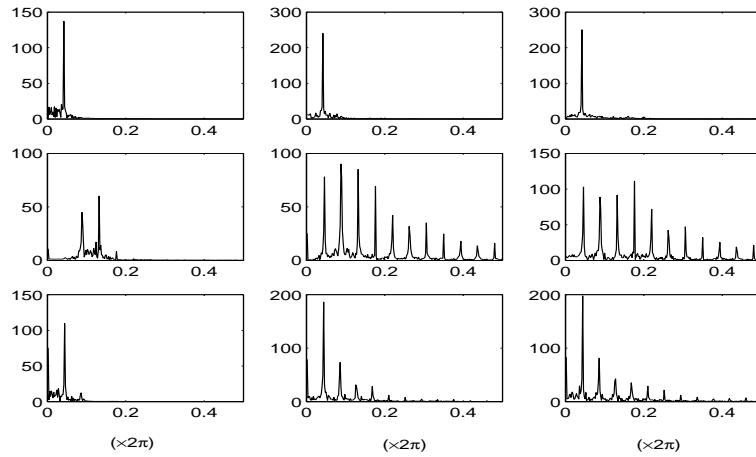


Figura 2.16: Sinais extraídos no domínio da frequência (SNR=5dB) com $M = 11$ atrasos e uma única direcção: coluna da esquerda $q = 1$ (SSA), centro $q = 3$ e direita $q = 5$.

clusters (figuras 2.15 e 2.16, coluna 2 e 3).

2.9 Validação dos parâmetros

A selecção do número de *clusters* é um parâmetro difícil de estimar. Por questões práticas, o número de vectores em cada *cluster* deve ser superior à dimensão do número de atrasos ($N_{c_i} \geq M$) o que pode constituir um limite superior do número de *clusters*.

Para a selecção do número óptimo de *clusters*, fez-se variar o número dos mesmos e foram efectuadas duas experiências :

- Cálculo do erro quadrático médio , MSE, entre o sinal original e o sinal reconstruído.
- Aplicação da Kurtosis ao erro obtido entre o sinal original e o sinal reconstruído ($\hat{v}[n] =$

$$x[n] - \hat{y}[n]), [18, 19].$$

As experiências efectuadas tiveram por base os resultados apresentados na secção 2.8.

| <i>Clusters</i> (q) | Sinusóide | | | Sinal <i>Funny</i> | | | Dente-de-Serra | | |
|-------------------------|-----------|--------|--------|--------------------|-------|-------|----------------|-------|-------|
| | 1 | 3 | 5 | 1 | 3 | 5 | 1 | 3 | 5 |
| $M = 11$ | 0.0405 | 0.0313 | 0.3169 | 0.31 | 0.091 | 0.045 | 0.177 | 0.076 | 0.206 |
| $M = 36$ | 0.012 | 0.006 | 0.009 | 0.758 | 0.041 | 0.031 | 0.219 | 0.052 | 0.025 |

Tabela 2.1: Erro quadrático médio entre os sinais originais e reconstruídos apresentados nas figuras 2.11 e 2.13.

A tabela 2.1 apresenta o MSE entre os sinais originais e os sinais reconstruídos quando é utilizado o critério MDL e um SNR=5dB. A tabela 2.2 apresenta o MSE entre os sinais originais e os sinais reconstruídos, quando é seleccionado o maior valor próprio em cada *cluster* e usado um SNR=5dB.

| <i>Clusters</i> (q) | Sinusóide | | | Sinal <i>Funny</i> | | | Dente-de-Serra | | |
|-------------------------|-----------|-------|------|--------------------|-------|-------|----------------|-------|-------|
| | 1 | 3 | 5 | 1 | 3 | 5 | 1 | 3 | 5 |
| $M = 11$ | 0.11 | 0.021 | 0.03 | 0.731 | 0.247 | 0.116 | 0.261 | 0.093 | 0.041 |
| $M = 36$ | 0.11 | 0.001 | 0.01 | 0.815 | 0.440 | 0.240 | 0.298 | 0.119 | 0.061 |

Tabela 2.2: Erro quadrático médio entre os sinais originais e reconstruídos apresentados na figura 2.15.

Pela análise da tabela 2.1 verifica-se que o número de atrasos e o número de *clusters* que minimiza o MSE é $M = 36$ atrasos e $q = 5$ *clusters* para os três sinais.

Quando é seleccionado o maior valor próprio em cada *cluster* (tabela 2.2) o menor MSE é encontrado para $M = 11$ atrasos e $q = 5$ *clusters*, quando os sinais são de banda larga e $M = 36$ atrasos e $q = 3$ *clusters* para o sinal sinusoidal.

Uma outra forma de validar os parâmetros deste algoritmo é calcular o valor da kurtosis do ruído estimado $\hat{v}[n]$ uma vez que o ruído tem uma distribuição gaussiana.

Os gráficos da figura 2.17 mostram o número de *clusters* que minimizam o valor da kurtosis, quando é aplicado o critério MDL. O número de *clusters* variou entre $q = 1, \dots, 6$, foram consideradas duas janelas de dimensão $M = 11$ (Primeira Linha) e $M = 36$ (Segunda Linha) e efectuadas 1000 simulações para cada caso.

Os gráficos da figura 2.17 representam a função densidade de probabilidade do número de *clusters* que minimiza o valor da kurtosis.

Observando a figura 2.17, verifica-se que o número de *clusters* que minimiza o valor da kurtosis é $q = 5$ em todos os sinais, considerando $M = 11$ e $M = 36$ atrasos, excepto para o sinal sinusoidal, quando são usados $M = 11$ atrasos. Neste caso, é seleccionado $q = 1$ *cluster*, devido à sobreestimação do critério MDL. Este resultado confirma o apresentado nos gráficos

das figuras 2.13 e 2.14.

Quando o número de direcções é fixo, o aumento do número de *clusters* não apresenta problemas relativamente à sobreestimação do subespaço. Nas experiências efectuadas nas secções anteriores, foram considerados apenas $q = 5$ *clusters*, para uma comparação com os resultados obtidos do critério MDL. No entanto, quando é seleccionada apenas uma direcção, um aumento do número de *clusters*, permite obter melhores resultados.

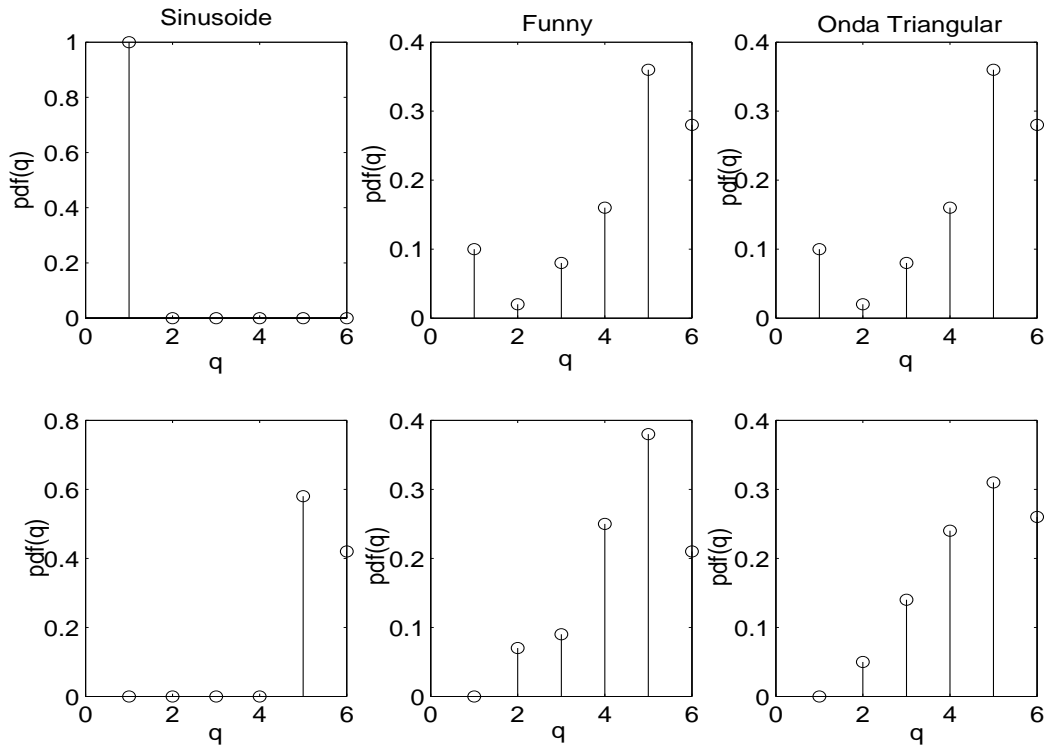


Figura 2.17: Função Densidade de Probabilidade do número de clusters que minimiza o valor da kurtosis do erro usando SSA e Local SSA.

2.10 Conclusões

Neste capítulo apresentou-se o algoritmo Local SSA, que se baseia no algoritmo SSA com a introdução de dois novos passos: *clustering* e *unclustering*. Foi efectuada uma descrição em pseudo-código do mesmo, (algoritmos 1 a 4).

Fez-se um estudo com sinais artificiais, comparando o desempenho dos algoritmo Local SSA e SSA na redução de ruído em séries temporais periódicas, fazendo variar o número de *clusters* e a dimensão do *embedding*. Verificou-se que para um SNR elevado, o algoritmo SSA apresenta um bom desempenho, se a dimensão do *embedding* for superior ao período do sinal. Para um SNR baixo, há perda da periodicidade do sinal, registando-se assim um pior desempenho do algoritmo. Uma alternativa para reduzir o ruído elevado em séries temporais é introduzir *clusters* no algoritmo SSA - Local SSA. Os *clusters* permitem agrupar o sinal por diferentes

amplitudes, havendo uma maior sensibilidade para a extração do ruído.

Pelos resultados apresentados, conclui-se que o algoritmo Local SSA é mais eficaz na redução de ruído em séries temporais do que o algoritmo SSA, nomeadamente em sinais de banda larga e com SNR baixo.

Capítulo 3

Kernel PCA

3.1 Introdução

O Kernel PCA é um método não linear de análise de componentes principais, recentemente introduzido em [20] e baseia-se na selecção de características de um determinado conjunto de dados, num espaço de dimensão elevada.

Os dados do espaço de entrada são mapeados através de uma função não linear ϕ , formando o espaço característico. Posteriormente, é aplicado um PCA linear para a extracção das características associadas às principais direcções. A ideia principal deste método consiste em aplicar a análise de componentes principais a um conjunto de dados não lineares, baseando-se apenas na informação dos dados do espaço de entrada.

Neste capítulo é descrita a forma de construir uma matriz de kernel e são também descritos os vários tipos de kernel. Posteriormente, é descrito o algoritmo designado por KPCA e dois métodos de reconstrução dos dados para o espaço de entrada.

São apresentados alguns resultados da aplicação do algoritmo KPCA a dois conjuntos de dados e efectuada uma comparação dos resultados com os resultados obtidos pelo algoritmo Local SSA . Por fim, são apresentados alguns resultados que evidenciam a importância da selecção do parâmetro do kernel *Radial Basis Function* (RBF) na extracção de ruído e apresentadas algumas conclusões.

3.2 Mapeamento dos Dados e Matriz de Produtos Internos

Seja $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ a matriz do conjunto de dados no espaço de entrada, onde $\mathbf{x}_i \in R^d$.

O mapeamento dos dados de entrada num espaço de dimensão superior corresponde a transformar os dados através de uma função não linear ϕ :

$$\Phi : \mathbf{x} \rightarrow \phi(\mathbf{x}) \in \mathfrak{S} \quad (3.1)$$

em que $\Phi(X) = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_N)]$ vai representar o novo conjunto de dados no espaço característico \mathfrak{S} [21].

Seja C a matriz de correlação (não normalizada) e K a matriz de produtos internos ou matriz de kernel dos dados no espaço característico:

$$C = \Phi(X)\Phi^T(X) \quad e \quad K = \Phi^T(X)\Phi(X) \quad (3.2)$$

Note-se que a matriz C pode ser a matriz de covariância, se os dados forem centrados ($\sum \phi(\mathbf{x}_i) = 0$). Por consequência da decomposição em valores singulares, os valores próprios e os vectores próprios da matriz de correlação C e da matriz de kernel K estão relacionados entre si [22], conforme descrito no apêndice A. Os valores próprios das duas matrizes são iguais e os vectores próprios da matriz de covariância podem ser calculados a partir dos vectores próprios da matriz de kernel:

$$\mathbf{u}_k = \Phi(X)\mathbf{v}_k\lambda_k^{-\frac{1}{2}} \quad (3.3)$$

onde \mathbf{u}_k representa o k -ésimo vector próprio da matriz de correlação, \mathbf{v}_k representa o k -ésimo vector próprio da matriz de kernel e $\lambda_k^{-\frac{1}{2}}$ representa o k -ésimo valor próprio. Esta relação permite calcular os vectores próprios da matriz C , conhecidos os vectores próprios e respectivos valores próprios da matriz K .

Exemplo 1. Consideremos um espaço de entrada $X \subseteq \mathbf{R}^2$ e um mapeamento no espaço característico dado por

$$\phi : \mathbf{x} = (x_1, x_2) \rightarrow \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \in \mathfrak{S} = \mathbf{R}^3. \quad (3.4)$$

O mapeamento do conjunto de dados de entrada para o conjunto característico corresponde a passar de um conjunto de dados bidimensional para um conjunto de dados tridimensional. O produto interno dos dados no espaço característico é dado por:

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle \left(x_1^2, x_2^2, \sqrt{2}x_1x_2 \right), \left(z_1^2, z_2^2, \sqrt{2}z_1z_2 \right) \right\rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 \end{aligned} \quad (3.5)$$

A relação linear encontrada no espaço característico, corresponde a uma relação quadrática no espaço de entrada.

Como ilustra o exemplo anterior, é possível calcular o produto interno entre dois pontos do conjunto de dados do espaço característico sem avaliar explicitamente as suas coordenadas, ou seja, tendo por base apenas as coordenadas do espaço de entrada.

Exemplo 2. Seja $X = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$ o conjunto de dados de entrada.

Considerando o mapeamento do exemplo 1, eq. 3.4, os dados no espaço característico são:

$$\Phi(X) = \begin{bmatrix} 1 & 4 \\ 1 & 1 \\ \sqrt{2} & 2\sqrt{2} \end{bmatrix}.$$

que se pode calcular usando a relação 3.5 A matriz de Kernel dos dados X é dada por:

$$K = \Phi^T(X)\Phi(X) = \begin{bmatrix} 4 & 9 \\ 9 & 25 \end{bmatrix}.$$

Calculando a matriz de correlação não normalizada dos dados no espaço característico temos

$$C = \Phi(X)\Phi^T(X) = \begin{bmatrix} 17 & 5 & 9\sqrt{2} \\ 5 & 2 & 3\sqrt{2} \\ 9\sqrt{2} & 2\sqrt{2} & 10 \end{bmatrix}.$$

Decompondo a matriz de kernel e a matriz de correlação dos dados no espaço característico, verifica-se que estas apresentam os mesmos valores próprios (0.67 e 28.32) e os seus vectores próprios estão relacionados, conforme a equação 3.3.

Verifica-se, assim, pelos exemplos anteriores que, através da matriz de kernel, é possível extrair informação acerca da matriz de correlação dos dados no espaço de maior dimensão, manipulando os dados do espaço de entrada. Calcular a matriz de kernel é suficiente para extrair informação da matriz de correlação no espaço de dimensão elevada.

Por definição, a função de kernel k entre $\mathbf{x}, \mathbf{z} \in X$ satisfaz a condição

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = \phi^T(\mathbf{x})\phi(\mathbf{z}) \quad (3.6)$$

onde ϕ representa uma função que permite calcular as entradas da matriz de kernel, ou seja, o mapeamento dos dados do espaço de entrada. A matriz de kernel pode ser descrita como uma matriz de Gram, da seguinte forma [22]:

| K | 1 | 2 | ... | N |
|-----|---------------------------------|---------------------------------|-----|---------------------------------|
| 1 | $k(\mathbf{x}_1, \mathbf{x}_1)$ | $k(\mathbf{x}_1, \mathbf{x}_2)$ | ... | $k(\mathbf{x}_1, \mathbf{x}_N)$ |
| 2 | $k(\mathbf{x}_2, \mathbf{x}_1)$ | $k(\mathbf{x}_2, \mathbf{x}_2)$ | ... | $k(\mathbf{x}_2, \mathbf{x}_N)$ |
| ... | ... | ... | ... | ... |
| N | $k(\mathbf{x}_N, \mathbf{x}_1)$ | $k(\mathbf{x}_N, \mathbf{x}_2)$ | ... | $k(\mathbf{x}_N, \mathbf{x}_N)$ |

Tabela 3.1: Matriz de kernel

Se considerarmos o conjunto de dados de entrada X , a matriz de Gram G é definida como uma matriz $N \times N$, onde as entradas da matriz são $G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Se for usada uma função de kernel para calcular o produto interno entre os vectores do espaço característico, através do mapeamento ϕ , as entradas da matriz de Gram são $G_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j)$. Na literatura, o cálculo do produto interno no espaço característico usando os dados de entrada é designado por “kernel trick”.

Em suma, os dados são projectados num espaço de maior dimensão, através da função de kernel. Por consequência da decomposição em valores singulares, é possível encontrar os vectores próprios da matriz de covariância do espaço característico associados às principais direcções, usando apenas a informação dos dados do espaço de entrada.

3.3 Funções de Kernel

Nesta secção são apresentadas algumas funções de kernel para o cálculo da matriz 3.2. São considerados dois tipos de funções de kernel:

- **Invertíveis**

Dependem apenas da distância entre os dados no espaço de entrada. São designados por kernel isotrópico.

1. Gaussiano

$$k(\mathbf{x}, \mathbf{z}) = \exp(\langle \mathbf{x}, \mathbf{z} \rangle) \quad (3.7)$$

2. RBF- Kernel Gaussiano normalizado

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= \frac{\exp(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\sigma^2})}{\sqrt{\exp(\frac{\|\mathbf{x}\|^2}{\sigma^2}) \exp(\frac{\|\mathbf{z}\|^2}{\sigma^2})}} \\ &= \exp\left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\sigma^2} - \frac{\langle \mathbf{x}, \mathbf{x} \rangle}{2\sigma^2} - \frac{\langle \mathbf{z}, \mathbf{z} \rangle}{2\sigma^2}\right) \\ &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \end{aligned} \quad (3.8)$$

onde $\sigma > 0$. No caso do kernel RBF, as imagens de todos os pontos têm norma 1 no espaço característico uma vez que $k(\mathbf{x}, \mathbf{x}) = \exp(0) = 1$.

- **Não invertíveis**

Dependem apenas do produto interno dos dados no espaço de entrada

1. Linear

$$k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle \quad (3.9)$$

2. Polinomial

$$k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + c)^d \quad (3.10)$$

onde d representa a ordem do polinómio e c uma constante.

3. Sigmóide

$$k(\mathbf{x}, \mathbf{z}) = \tanh(\vartheta \langle \mathbf{x}, \mathbf{z} \rangle + \delta) \quad (3.11)$$

onde $\vartheta > 0$ e $\delta < 0$ representam constantes.

3.4 Algoritmo Kernel PCA

O algoritmo KPCA pode ser interpretado como o cálculo das componentes principais não lineares de um conjunto de dados, no espaço de maior dimensão a partir da informação do espaço de entrada.

Este algoritmo pode ser executado tendo em conta os seguintes passos (fig.3.1):

1. Mapeamento dos dados num espaço de dimensão superior e cálculo da matriz de kernel
 - $\Phi(X) = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_N)]$ representa o espaço de dimensão superior ou espaço característico
 - $K = \Phi^T(X)\Phi(X)$ representa a matriz de Kernel
2. Aplicação do PCA no espaço característico e cálculo das principais direcções
 - Decomposição da matriz de kernel em valores próprios e vectores próprios ($K = V\Lambda V^T$)
 - Cálculo dos vectores próprios e valores próprios da matriz de correlação, tendo por base a eq. 3.3
 - Cálculo do vector das projecções β , no espaço de dimensão superior
3. Reconstrução dos dados no espaço característico, $\phi_{rec}(\mathbf{z})$
4. Reconstrução dos dados no espaço de entrada, cálculo de $\hat{\mathbf{z}}$. A reconstrução pode ser efectuada tendo por base dois métodos:
 - Método iterativo (Schoelkopf's fix-point algorithm) [23, 24]
 - Método das distâncias (Kwok & Tsang) [5].

Na prática, nunca se trabalha directamente no espaço característico. Como veremos mais à frente, é possível calcular as principais variáveis do processo com os dados no espaço de entrada, utilizando o “kernel trick”. O mapeamento dos dados é feito directamente pela função de kernel entre o espaço de entrada e o espaço KPCA, como está representado pela linha a tracejado (figura 3.1) e as pré-imagens dos dados $\phi(\mathbf{x})$ no espaço de entrada são efectuadas implicitamente a partir da matriz de kernel.

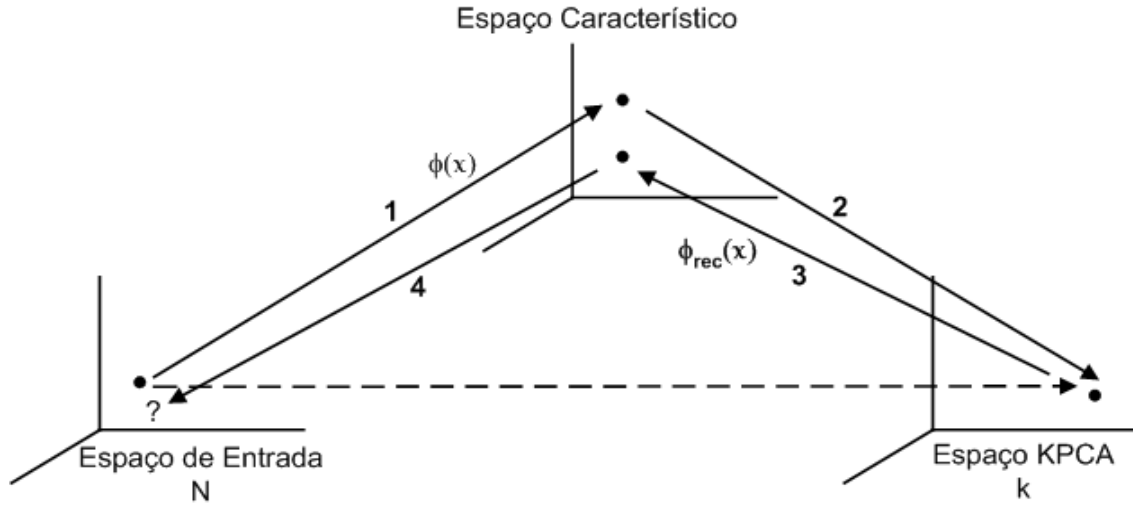


Figura 3.1: Esquema adaptado de [1]. Conceptualmente, o KPCA efectua um mapeamento não linear aos dados $\phi(\mathbf{x})$ para projectar os vectores de entrada num espaço de maior dimensão \mathfrak{S} , (1). Posteriormente, é aplicado um PCA linear aos dados do espaço característico, obtendo um espaço de menor dimensão KPCA (2). Para reconstruir um vector no espaço de entrada, a representação KPCA é projectada no espaço característico (3) e é efectuado um mapeamento $\phi(\mathbf{x})$ inverso (4)

3.4.1 Matriz de Kernel Centrada

Tal como é efectuado no caso do algoritmo PCA, o conjunto de dados no espaço característico (Φ) também é centrado,

$$\tilde{\Phi}(X) = \Phi(X) - \frac{1}{N}\Phi(X)\mathbf{j}_N\mathbf{j}_N^T = \Phi(X)(I - H) \quad (3.12)$$

onde $\tilde{\Phi}(X)$ representa o novo mapeamento dos dados no espaço característico, $\mathbf{j}_N = [1, 1, \dots, 1]^T$ com dimensão N e $H = \frac{\mathbf{j}_N\mathbf{j}_N^T}{N}$.

A nova matriz de kernel centrada (\tilde{K}), obtida através da manipulação do conjunto de dados do espaço de entrada é dada por

$$\begin{aligned} \tilde{K}(X, X) &= \langle \tilde{\Phi}(X), \tilde{\Phi}(X) \rangle \\ &= \langle \Phi(X)(I - H), \Phi(X)(I - H) \rangle \\ &= (\Phi(X)(I - H))^T \Phi(X)(I - H) \end{aligned} \quad (3.13)$$

Desenvolvendo a expressão anterior temos:

$$\begin{aligned}
\tilde{K}(X, X) &= \Phi^T(X)\Phi(X) - H\Phi^T(X)\Phi(X) - \Phi^T(X)\Phi(X)H + H\Phi^T(X)\Phi(X)H \\
&= K(X, X) - HK(X, X) - K(X, X)H + HK(X, X)H \\
&= K - HK - KH + HKH
\end{aligned}
\tag{3.14}$$

onde $K(X, X)$ é a matriz de Gram representada na tabela 3.2 e $\tilde{K}(X, X)$, a matriz de kernel centrada.

Algoritmo 5 Matriz de Kernel Centrada

| | |
|---------|---|
| Entrada | Conjunto de dados $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ |
| | $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \dots, N$ $H = \frac{\mathbf{j}\mathbf{j}^T}{N}$ onde $\mathbf{j} = [1, 1, \dots, 1]^T$ tem dimensão N $\tilde{K} = K - HK - KH + HKH$ |
| Saída | Matriz de Kernel Centrada \tilde{K} |

3.4.2 Decomposição da matriz de Kernel

A decomposição em valores próprios e vectores próprios da matriz \tilde{K} é dada por:

$$\tilde{K} = V\Lambda V^T \tag{3.15}$$

onde $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N]$ com $\mathbf{v}_i = [\nu_{1i}, \nu_{2i}, \dots, \nu_{Ni}]^T$ representa a matriz dos vectores próprios e $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ o conjunto dos valores próprios associados.

A matriz dos vectores ortogonais da matriz de covariância do espaço característico C é dada por

$$U = \tilde{\Phi}(X)A = \Phi(X)(I - H)A \tag{3.16}$$

em que $A = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_N]$, sendo $\boldsymbol{\alpha}_n = \lambda_n^{-\frac{1}{2}}\mathbf{v}_n$ obtido em termos do n -ésimo par (v_n, λ_n) , vector próprio-valor próprio, da matriz de kernel centrada, \tilde{K} .

Para extrair as componentes principais não lineares das imagens Φ dos pontos \mathbf{z} são consideradas apenas as direcções que correspondem aos l valores próprios mais significativos, onde $l \ll N$.

Calcula-se o vector de projecção l -dimensional de Φ no espaço U_l , ou seja, são escolhidas

apenas as l principais direcções que vão constituir a dimensão do espaço característico β ,

$$\begin{aligned}
 \beta &= U_l^T \Phi(\mathbf{z}) \\
 &= A_l^T (I - H)^T \Phi^T(X) \phi(\mathbf{z}) \\
 &= A_l^T (I - H)^T k(X, \mathbf{z})
 \end{aligned}
 \tag{3.17}$$

em que $\beta = [\beta_1, \dots, \beta_l]^T$ representa o conjunto das l componentes principais não lineares e $k(X, \mathbf{z}) = [k(\mathbf{x}_1, \mathbf{z}), \dots, k(\mathbf{x}_N, \mathbf{z})]^T$ uma coluna da matriz de kernel [5]. Se \mathbf{z} é um dos elementos do conjunto de treino X , então $k(X, \mathbf{z})$ representa a coluna j da matriz de kernel, ou seja, $k(X, \mathbf{x}_j) = K_{i,j}, i = 1, \dots, N$. Note-se que os vectores próprios U_l não são explicitamente calculados (eq. 3.16) para saber o valor das projecções nas direcções principais. Este é obtido, recorrendo apenas à informação da matriz de kernel (eq. 3.17).

O algoritmo KPCA não é mais do que um PCA no espaço de dimensão \mathfrak{S} . O vector de entrada é mapeado num espaço de maior dimensão onde vão ser aplicados métodos lineares.

Algoritmo 6 KPCA: Projecções nas direcções principais

| | |
|---------|---|
| entrada | matriz de Kernel centrada \tilde{K} matriz de Kernel K exemplar do conjunto de treino $\mathbf{z} = X(:, i), \forall i \in N$ |
| | $[V, \Lambda] = eig(\tilde{K})$ $\lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_l > \lambda_{l+1} > \dots > \lambda_N$ Seleccção dos l valores próprios mais significativos $A = [\alpha_1, \alpha_2, \dots, \alpha_l]$, onde $\alpha_l = \lambda_k^{-\frac{1}{2}} \mathbf{v}_l$ $\beta = A^T (I - H)^T k(X, \mathbf{z})$ |
| saída | Componentes principais não lineares $\beta = \{\beta_1, \dots, \beta_l\}$ Vectores próprios normalizados $A = \{\alpha_1, \dots, \alpha_l\}$ |

3.4.3 Reconstrução no espaço Característico

Para reconstruir a imagem $\phi(\mathbf{z})$ no subespaço de projecção nas l principais direcções do espaço \mathfrak{S} , é definido $\phi_{rec}(\mathbf{z})$, [5] dado por:

$$\begin{aligned}\phi_{rec}(\mathbf{z}) &= U\beta = UU^T\phi(\mathbf{z}) \\ &= \Phi(X)(I - H)A\beta \\ &= \Phi(X)\gamma\end{aligned}\tag{3.18}$$

onde $\gamma = (I - H)A\beta$ representa um vector $N \times 1$. Se o valor de l for suficientemente elevado para projectar os dados em todas as direcções associadas aos valores próprios diferentes de zero, temos que $\phi_{rec}(\mathbf{z}) = \phi(\mathbf{z})$. Se o valor de $l < N$, então são satisfeitas duas condições:

- o erro quadrático médio da reconstrução, $\sum \|\phi_{rec}(\mathbf{z}) - \phi(\mathbf{z})\|$ torna-se menor à medida que aumentamos o número de componentes
- a variância retida das projecções nas principais direcções ortogonais do subespaço \mathfrak{S} deve ser máxima

Na prática, a eq.3.18 nunca é explicitamente usada, no entanto é necessária para a manipulação algébrica dos dados quando se pretende efectuar o mapeamento inverso destes para o espaço de entrada.

3.5 Reconstrução no espaço de entrada

Tal como está referido em [24, 25], a reconstrução no espaço de entrada pode ter interesse nas seguintes aplicações:

- De-noising: Dado \mathbf{x} , efectua-se o mapeamento $\phi(\mathbf{x})$, selecciona-se as componentes associadas aos maiores valores próprios, obtém-se $\phi_{rec}(\mathbf{x})$ e calcula-se a pré-imagem $\hat{\mathbf{z}}$. Nesta aplicação, são apenas retidas as l principais direcções, sendo as componentes desprezadas associadas ao ruído do sinal. Assim, o vector \mathbf{x} sem ruído é representado por $\hat{\mathbf{z}}$.
- Compressão: Dado o conjunto dos vectores próprios normalizados A e o vector das projecções β de $\Phi(X)$ e desconhecendo X , calcula-se a pré-imagem como uma reconstrução aproximada de X .

Para encontrar a pré-imagem $\hat{\mathbf{z}}$ no espaço de entrada existem dois métodos a considerar:

- Método iterativo (fixed-point algorithm) [23, 24].

- Método das distâncias (Kwok & Tsang) [5].

Estes métodos têm como objectivo encontrar a pré-imagem de $\hat{\mathbf{z}}$, de tal forma que $\phi(\hat{\mathbf{z}}) = \phi_{rec}(\mathbf{z})$. No entanto, existem duas restrições:

1. Existência: $\hat{\mathbf{z}}$ pode não existir [20]
2. Unicidade: se $\hat{\mathbf{z}}$ existir, pode não ser único

3.6 Método iterativo

O objectivo da reconstrução iterativa é encontrar a pré-imagem $\hat{\mathbf{z}}$, correspondente a $\phi_{rec}(\mathbf{x})$ do espaço característico de forma iterativa.

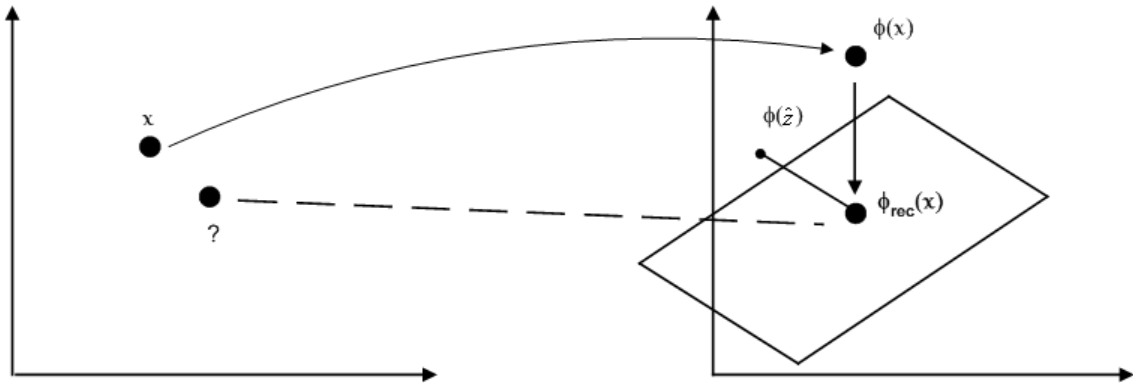


Figura 3.2: Representação esquemática do Método Iterativo (*Fixed-Point*) para encontrar a pré-imagem no algoritmo KPCA.

A pré-imagem $\hat{\mathbf{z}}$ é obtida, recorrendo a uma solução por aproximação que minimiza a distância quadrática entre $\phi(\hat{\mathbf{z}})$ e $\phi_{rec}(\mathbf{x})$, distância no espaço característico [23]:

$$\begin{aligned}
 \rho(\hat{\mathbf{z}}) &= \|\phi(\hat{\mathbf{z}}) - \phi_{rec}(\mathbf{x})\|^2 = \\
 &= \langle \phi(\hat{\mathbf{z}}) - \phi_{rec}(\mathbf{x}), \phi(\hat{\mathbf{z}}) - \phi_{rec}(\mathbf{x}) \rangle = \\
 &= \|\phi(\hat{\mathbf{z}})\|^2 - 2\phi_{rec}^T(\mathbf{x})\phi(\hat{\mathbf{z}}) + \|\phi_{rec}(\mathbf{x})\|^2 = \\
 &= k(\hat{\mathbf{z}}, \hat{\mathbf{z}}) - 2\phi_{rec}^T(\mathbf{x})\phi(\hat{\mathbf{z}}) + \Omega
 \end{aligned} \tag{3.19}$$

onde Ω representa os termos independentes de $\hat{\mathbf{z}}$.

Substituindo na equação anterior as equações 3.18 e 3.16, temos:

$$\begin{aligned}
\rho(\hat{\mathbf{z}}) &= k(\hat{\mathbf{z}}, \hat{\mathbf{z}}) - 2(\Phi(X)(I - H)A\beta)^T \phi(\hat{\mathbf{z}}) \\
&= k(\hat{\mathbf{z}}, \hat{\mathbf{z}}) - 2\beta^T A^T (I - H)^T k(X, \hat{\mathbf{z}}) \\
&= k(\hat{\mathbf{z}}, \hat{\mathbf{z}}) - 2\gamma^T k(X, \hat{\mathbf{z}})
\end{aligned} \tag{3.20}$$

Note-se que a resolução do problema é possível pela eq. 3.20, sem recorrer explicitamente aos dados mapeados no espaço característico, mas apenas recorrendo à informação da matriz de kernel.

Para um dado extremo, o gradiente da eq. (3.20) em relação a $\hat{\mathbf{z}}$ é dado por:

$$\frac{\partial \rho(\hat{\mathbf{z}})}{\partial \hat{\mathbf{z}}} = \frac{\partial k(\hat{\mathbf{z}}, \hat{\mathbf{z}})}{\partial \hat{\mathbf{z}}} - 2\gamma^T \frac{\partial k(\hat{\mathbf{z}}, X)}{\partial \hat{\mathbf{z}}} \tag{3.21}$$

Consideremos dois casos distintos:

1. Assumindo que $k(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{c})$, a equação 3.21 é dada por:

$$\begin{aligned}
\frac{\partial \rho(\hat{\mathbf{z}})}{\partial \hat{\mathbf{z}}} &= \gamma_1(\mathbf{x}_1 - \hat{\mathbf{z}})e^{(-\frac{\|\mathbf{x}_1-\hat{\mathbf{z}}\|^2}{c})} + \gamma_2(\mathbf{x}_2 - \hat{\mathbf{z}})e^{(-\frac{\|\mathbf{x}_2-\hat{\mathbf{z}}\|^2}{c})} + \dots + \gamma_N(\mathbf{x}_N - \hat{\mathbf{z}})e^{(-\frac{\|\mathbf{x}_N-\hat{\mathbf{z}}\|^2}{c})} \\
&= X(\gamma \diamond k(X, \hat{\mathbf{z}})) - \hat{\mathbf{z}}\gamma^T k(X, \hat{\mathbf{z}})
\end{aligned} \tag{3.22}$$

onde \diamond representa o produto de Hadamard ¹ [23, 21]. Manipulando a eq. 3.22 obtém-se o valor extremo:

$$\hat{\mathbf{z}} = \frac{X(\gamma^T \diamond k(X, \hat{\mathbf{z}}))}{\gamma^T k(X, \hat{\mathbf{z}})} \tag{3.23}$$

Considerando o algoritmo *Fixed-Point*, é possível encontrar iterativamente o valor de $\hat{\mathbf{z}}$, [23, 21] dado por:

$$\hat{\mathbf{z}}_{t+1} = \frac{X(\gamma^T \diamond k(X, \hat{\mathbf{z}}_t))}{\gamma^T k(X, \hat{\mathbf{z}}_t)} \tag{3.24}$$

2. Assumindo que $k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + c)^d$, a equação 3.21 é dada por [5]:

¹Produto elemento a elemento de vectores

$$\frac{\partial \rho(\hat{\mathbf{z}})}{\partial \hat{\mathbf{z}}} = d\hat{\mathbf{z}}(\hat{\mathbf{z}}^T \hat{\mathbf{z}} + c)^{d-1} - 2d(\gamma \diamond (X^T \hat{\mathbf{z}} + c)^{d-1}) \quad (3.25)$$

Manipulando a eq. 3.25 temos:

$$\begin{aligned} \hat{\mathbf{z}}_{t+1} &= X \left[\gamma \diamond \left(\frac{X^T \hat{\mathbf{z}}_t + c}{\hat{\mathbf{z}}_t^T \hat{\mathbf{z}}_t + c} \right)^{(d-1)} \right] \\ &= X \left[\gamma \diamond \left(\frac{k_1(X, \hat{\mathbf{z}}_t)}{k_1(\hat{\mathbf{z}}_t, \hat{\mathbf{z}}_t)} \right) \right] \end{aligned} \quad (3.26)$$

onde $k_1(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + c)^{d-1}$.

Este método acarreta algumas instabilidades numéricas, relacionadas com a convergência dos valores procurados, nomeadamente, a escolha de inicialização $\hat{\mathbf{z}}_0$ [23]. Se o valor de $\gamma^T k(X, \hat{\mathbf{z}}_t)$ for muito pequeno (eq. 3.23), o processo não converge e deve ser novamente inicializado com outro valor de $\hat{\mathbf{z}}_0$.

3.7 Método das distâncias

Recentemente, foi proposto um método em [5], para encontrar a pré-imagem de um ponto do espaço característico, designado por método das distâncias.

Este método baseia-se no facto de ser possível calcular as coordenadas de um novo ponto no espaço de entrada, conhecendo as distâncias desse ponto a um conjunto de pontos no espaço característico (figura 3.3). Para dois vectores \mathbf{x} e \mathbf{z} mapeados no espaço característico pela função ϕ , $\phi(\mathbf{x})$ e $\phi(\mathbf{z})$, calcula-se o quadrado da distância euclidiana entre eles

$$\begin{aligned} \tilde{d}^2(\phi(\mathbf{x}), \phi(\mathbf{z})) &= \langle \phi(\mathbf{x}) - \phi(\mathbf{z}), \phi(\mathbf{x}) - \phi(\mathbf{z}) \rangle \\ &= \|\phi(\mathbf{x})\|^2 + \|\phi(\mathbf{z})\|^2 - 2\phi^T(\mathbf{x})\phi(\mathbf{z}) \end{aligned} \quad (3.27)$$

Note-se que esta distância pode ser calculada, recorrendo à matriz de kernel e não a partir da informação dos dados mapeados .

Analogamente, calcula-se o quadrado da distância euclidiana dos respectivos pontos no espaço de entrada

$$\begin{aligned} d^2(\mathbf{x}, \mathbf{z}) &= \langle \mathbf{x} - \mathbf{z}, \mathbf{x} - \mathbf{z} \rangle \\ &= \|\mathbf{x}\|^2 + \|\mathbf{z}\|^2 - 2\mathbf{x}^T \mathbf{z} \end{aligned} \quad (3.28)$$

Algoritmo 7 Pré-imagem: Método Iterativo

| | |
|----------------|--|
| entrada | Tipo de Kernel: RBF ou Polinomial Projecções não lineares $\beta = \{\beta_1, \dots, \beta_l\}$ Vectores próprios normalizadas $A = \{\alpha_1, \dots, \alpha_l\}$ Valor do limiar |
| | Calcular o valor inicial z_0 $\gamma = (I - H)A\beta$ $G = \gamma \diamond k(X, \hat{z}_t)$ Caso Kernel RBF $\hat{z}_{t+1} = \frac{XG}{\sum G}$ Caso Kernel Polinomial $\hat{z}_{t+1} = X \left[\gamma \diamond \left(\frac{k_1(X, \hat{z}_t)}{k_1(\hat{z}_t, \hat{z}_t)} \right) \right]$ Parar quando $ z_{t+1} - \hat{z}_t ^2 < \text{limiar}$ |
| saída | Pré-imagem \hat{z} |

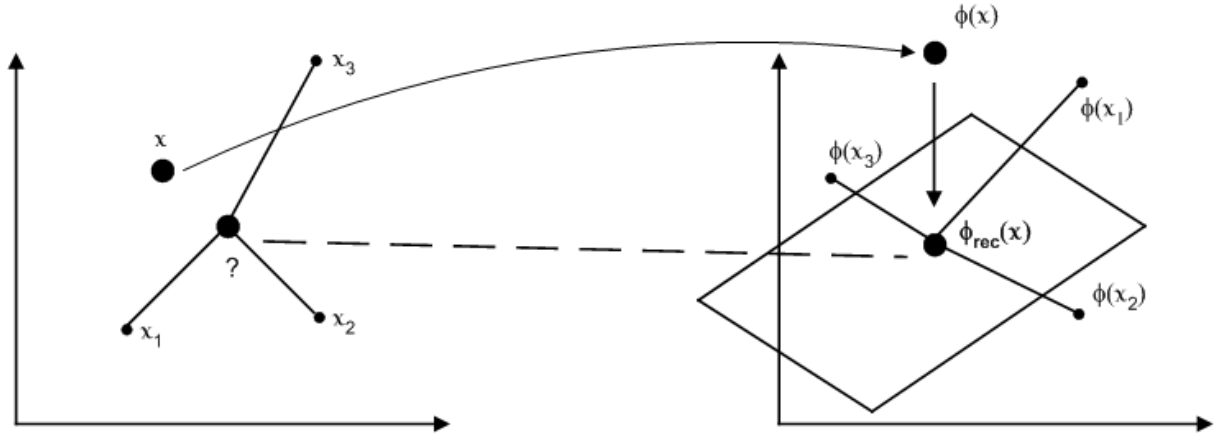


Figura 3.3: Representação esquemática do Método das Distâncias para encontrar a pré-imagem no algoritmo KPCA.

Através da relação existente entre as suas distâncias, é possível encontrar a pré-imagem \hat{z} , como está descrito nas secções seguintes. Este método apresenta algumas vantagens relativamente ao método iterativo: não apresenta instabilidade numérica de convergência, nem é necessário escolher nenhum parâmetro de inicialização.

3.7.1 Distância no Espaço Característico

Para calcular o vector de distâncias no espaço característico, são calculados os produtos internos de $\phi_{rec}(z)$ com todas as colunas de $\Phi(X)$.

$$\tilde{d}^2(\phi_{rec}(z), \Phi(X)) = \|\phi_{rec}(z)\|^2 \mathbf{j}^T + D - 2\phi_{rec}^T(z)\Phi(X) \quad (3.29)$$

onde D representa a diagonal da matriz de kernel, $D = \text{diag}(K(X, X))$.

$$D = [\phi^T(\mathbf{x}_1)\phi(\mathbf{x}_1), \phi^T(\mathbf{x}_2)\phi(\mathbf{x}_2), \dots, \phi^T(\mathbf{x}_N)\phi(\mathbf{x}_N)] \quad (3.30)$$

Pela eq.3.18

$$\begin{aligned}
\|\phi_{rec}(\mathbf{z})\|^2 &= (\Phi(X)\gamma)^T (\Phi(X)\gamma) \\
&= \gamma^T K(X, X)\gamma
\end{aligned}
\tag{3.31}$$

e

$$\begin{aligned}
\phi_{rec}^T(\mathbf{z})\Phi(X) &= \gamma^T \Phi^T(X)\Phi(X) \\
&= \gamma^T K(X, X)
\end{aligned}
\tag{3.32}$$

Assim, substituindo a eq.(3.31) e a eq.(3.32) na eq.(3.29), obtemos :

$$\tilde{\mathbf{d}}^2(\phi_{rec}(\mathbf{z}), \Phi(X)) = \gamma^T K(X, X)\gamma \mathbf{j}^T + D - 2\gamma^T K(X, X)
\tag{3.33}$$

Note-se que o cálculo da distância no espaço característico depende apenas da matriz de kernel e não dos dados mapeados.

Algoritmo 8 Distância no Espaço Característico

| | |
|----------------|---|
| entrada | matriz de Kernel K Projecções não lineares $\beta = \{\beta_1, \dots, \beta_l\}$ Vectores próprios normalizadas $A = \{\alpha_1, \dots, \alpha_l\}$ |
| saída | $\gamma = (I - H)A\beta$ $N = \gamma^T K \gamma$ $P = \gamma^T K$ $D = \text{diag}(K)$ $\tilde{\mathbf{d}} = N\mathbf{j}^T + D - 2P$ |

3.7.2 Distância no Espaço de Entrada

Dada a distância no espaço característico entre o ponto \mathbf{z} e os pontos da matriz X , é necessário encontrar a distância correspondente dos dados no espaço de entrada.

Para calcular a distância quadrada no espaço de entrada, (\mathbf{d}^2) entre o ponto \mathbf{z} e todas as pontos da matriz X consideram-se dois casos distintos:

1. Assumindo que $k(\mathbf{x}, \mathbf{z})$ é um kernel isotrópico, ou seja, depende da distância dos dados no espaço de entrada, existe uma relação entre a distância do espaço característico ($\tilde{\mathbf{d}}^2$) e a distância do espaço de entrada (\mathbf{d}^2).

A distância entre \mathbf{z} e todos os pontos da matriz X no espaço característico é dada por:

$$\begin{aligned}\tilde{\mathbf{d}}^2(\mathbf{z}, X) &= k(\mathbf{z}, \mathbf{z})\mathbf{j}^T + D - 2\phi^T(\mathbf{z})\Phi(X) \\ &= k(\mathbf{z}, \mathbf{z})\mathbf{j}^T + D - 2\exp\left(-\frac{\mathbf{d}^2}{2\sigma^2}\right)\end{aligned}\tag{3.34}$$

onde $\exp\left(-\frac{\mathbf{d}^2}{2\sigma^2}\right)$ representa uma função de kernel RBF, equação 3.8. Assim

$$\begin{aligned}\mathbf{d}^2 &= -2\sigma^2\ln\left(\frac{1}{2}(k(\mathbf{z}, \mathbf{z})\mathbf{j}^T + D - \tilde{\mathbf{d}}^2)\right) \\ &= -2\sigma^2\ln\left(\mathbf{j}^T - \frac{1}{2}\tilde{\mathbf{d}}^2\right)\end{aligned}\tag{3.35}$$

uma vez que $k(\mathbf{z}, \mathbf{z}) = 1$ e $D = \mathbf{j}^T$.

2. Assumindo que o kernel é obtido pelo produto interno, como é o caso do kernel linear, polinomial e sigmóide, existe uma relação entre o produto interno no espaço de entrada, $\mathbf{x}^T \mathbf{z}$ e o produto interno no espaço característico, $\phi(\mathbf{x})^T \phi(\mathbf{z})$.

A distância entre o ponto \mathbf{z} e todos os pontos da matriz X no espaço de entrada é dada por:

$$\mathbf{d}^2 = \tilde{D} - \mathbf{z}^T \mathbf{z} \mathbf{j}^T - 2\mathbf{z}^T X\tag{3.36}$$

onde \tilde{D} representa a diagonal da matriz $X^T X$, i.e. $\tilde{D} = [\mathbf{x}_1^T \mathbf{x}_1, \dots, \mathbf{x}_N^T \mathbf{x}_N]$.

Se a função de kernel usada for uma função invertível, é possível encontrar o valor de $\mathbf{z}^T \mathbf{z}$ e $\mathbf{z}^T X$ de uma forma implícita à custa da distância no espaço característico. Caso contrário, o problema não tem solução.

Por exemplo, se a função de kernel for uma função polinomial (equação 3.10), então

$$\mathbf{z}^T \mathbf{x} = k(\mathbf{z}, \mathbf{x})^{\frac{1}{p}} - c\tag{3.37}$$

considerando p um valor real ímpar, a solução é única.

Se a função de kernel for uma função sigmóide (equação 3.11), então

$$\mathbf{z}^T \mathbf{x} = \frac{\tanh^{-1}(k(\mathbf{z}, \mathbf{x})) - \delta}{\vartheta} \quad (3.38)$$

Substituindo as equações 3.38 e 3.37 para o caso do kernel polinomial e sigmóide, na equação 3.36 temos:

- Para calcular o valor $\mathbf{z}^T \mathbf{z}$, $k(\mathbf{z}, \mathbf{z}) = \phi_{rec}^T(\mathbf{z})\phi_{rec}(\mathbf{z})$ é o ponto calculado a partir da equação 3.31.
- Para calcular o vector $\mathbf{z}^T X = [\mathbf{z}^T \mathbf{x}_1, \mathbf{z}^T \mathbf{x}_2 \dots \mathbf{z}^T \mathbf{x}_N]^T$, $k(\mathbf{z}, X) = \phi_{rec}^T(\mathbf{z})\Phi(X)$ é o vector obtido pela equação 3.32, ou seja, as equações 3.37, 3.38 são calculadas para os N pontos do vector $\mathbf{z}^T X$.

Conclui-se assim que, para construir o vector de distâncias do espaço característico (\mathbf{d}^2) existem três formas: são usadas as equações (3.34), (3.35) para o caso dos kernels isotrópicos, as equações (3.36) (3.32) (3.31) e (3.37) para o caso de kernels polinomiais e as equações (3.36) (3.32) (3.31) e (3.38), para o caso de kernels sigmóides.

3.7.3 Relação das Distâncias

Como é referido na secção 3.5, uma das restrições deste método é a não existência do ponto que procuramos. Uma forma de contornar este problema é considerar as n distâncias entre o $\phi_{rec}(\mathbf{z})$ e os n vizinhos mais próximos de \mathbf{z} , $\Phi(X)$, no espaço de entrada:

$$\mathbf{d}^2 = [d_1^2, d_2^2, \dots, d_n^2]^T \quad (3.39)$$

Alternativamente, podem ser considerados os n vizinhos mais próximos de \mathbf{z} no espaço característico.

Seja $L = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ a matriz $d \times n$ dos n vizinhos mais próximos de cada ponto do espaço de entrada. Decompondo em valores singulares a matriz centrada, apêndice A, temos

$$L(I - H) = E\Sigma V^T = EW \quad (3.40)$$

$E = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_d]$ representa uma matriz $d \times d$ formada por colunas ortogonais, \mathbf{e}_i , e $W = \Sigma V^T = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]$ representa uma matriz $d \times n$ em que as colunas \mathbf{w}_i representam as projecções de x_i nas direcções \mathbf{e} e $H = \frac{\mathbf{j}\mathbf{j}^T}{n}$, com $\mathbf{j} = [11\dots 1]^T$ um vector de dimensão n .

A distância de cada ponto do espaço de entrada à origem, neste caso ao centroíde dos n vizinhos mais próximos, é igual a $\|\mathbf{w}_i\|^2$, sendo $\mathbf{d}_0^2 = [\|\mathbf{w}_1\|^2, \dots, \|\mathbf{w}_n\|^2]^T$.

Assume-se que a imagem $\hat{\mathbf{z}}$ está contida no espaço dos n vizinhos considerados. Geralmente,

Algoritmo 9 Distância no Espaço Entrada

| | |
|----------------|--|
| entrada | matriz de Kernel K , parâmetros da matriz de kernel $\sigma, p, c, \vartheta, \delta$ Vector das distancias no espaço característico, $\tilde{\mathbf{d}}$ Projecções não lineares $\beta = \{\beta_1, \dots, \beta_l\}$ Vectores próprios normalizadas $A = \{\alpha_1, \dots, \alpha_l\}$ |
|----------------|--|

Caso kernel Isotrópico

$$D = \text{diag}(K)$$

$$\mathbf{d} = -2\sigma^2 \log(j^T - \frac{1}{2}\tilde{\mathbf{d}})$$

Caso kernel de Produtos Internos

$$\gamma = (I - H)A\beta$$

$$N = \gamma^T K \gamma$$

$$\mathbf{P} = \gamma^T K \quad \text{onde } \mathbf{P} = [p(1)p(2)\dots p(N)]$$

$$D = \text{diag}(X^T X)$$

\Rightarrow kernel Polinomial

$$\mathbf{d} = D - (N^{\frac{1}{p}} \mathbf{j}^T - c \mathbf{j}^T) - 2([p(1)^{\frac{1}{p}} - cp(2)^{\frac{1}{p}} - c\dots p(N)^{\frac{1}{p}} - c]^T)$$

\Rightarrow kernel Sigmóide

$$\mathbf{d} = D - \left[\frac{\tanh^{-1}(N) - \delta}{\vartheta} \right] \mathbf{j}^T - 2 \left[\frac{\tanh^{-1}(p(1)) - \delta}{\vartheta} \frac{\tanh^{-1}(p(2)) - \delta}{\vartheta} \dots \frac{\tanh^{-1}(p(N)) - \delta}{\vartheta} \right]^T$$

| | |
|--------------|--|
| saída | Vector das distâncias no espaço entrada \mathbf{d} |
|--------------|--|

a pré-imagem não existe no espaço de entrada e, conseqüentemente, a solução que satisfaz estas condições não existe.

Em [10] está descrito um método para calcular o valor de \hat{z} (Anexo B). Consideremos \hat{w} a pré-imagem que procuramos no novo espaço de dimensão $d \times n$, com $n < N$ (espaço constituído pelos vizinhos mais próximos). O ponto \hat{w} deve satisfazer a equação 3.41:

$$-2W^T \hat{w} = (d^2 - d_0^2) - \frac{j j^T}{n} (d^2 - d_0^2) \quad (3.41)$$

Como, $W \frac{j j^T}{n} = 0$ devido aos dados estarem centrados, a pré-imagem pode ser obtida

$$\hat{w} = -\frac{1}{2} (W W^T)^{-1} W (d^2 - d_0^2) = -\frac{1}{2} \Sigma^{-1} V^T (d^2 - d_0^2) \quad (3.42)$$

O valor da pré-imagem deve ser expresso em termos do sistema de coordenadas definido pelos e'_j s, ou seja, no espaço de entrada. Transformando \hat{w} para o sistema de coordenadas originais, temos:

$$\hat{z} = E \hat{w} + L \frac{j}{n} \quad (3.43)$$

em que $L \frac{j}{n}$ representa a média dos pontos vizinhos e \hat{z} a pré-imagem de $\phi(\hat{z})$.

3.8 Resultados

Foram efectuadas três experiências com sinais artificiais a duas dimensões no domínio do tempo. A primeira experiência evidencia uma das vantagens da aplicação do algoritmo KPCA, relativamente ao algoritmo Local SSA, considerando um SNR=10dB. A segunda experiência compara graficamente a performance do algoritmo Local SSA e KPCA, considerando um sinal de banda larga e um sinal de banda estreita, adicionados com SNR=20dB. A terceira experiência mostra a influência do parâmetro do kernel RBF, (σ^2), na eliminação de ruído gaussiano aleatório no sinal sinusoidal e no sinal dente-de-serra, considerando dois níveis de ruído (SNR=20dB e SNR=0dB).

3.8.1 Comparação dos algoritmos KPCA e SSA

Foi considerado um sinal sinusoidal com $N = 500$ amostras adicionado de ruído gaussiano aleatório (SNR=10dB). O sinal foi transformado num sinal a duas dimensões ($M = 2$), (figura 3.4). Foi calculada a matriz de kernel RBF (eq. 3.8) do sinal representado na figura 3.4 (b), considerando $\sigma^2 = 0.55$ e foi efectuada uma decomposição em valores e vectores próprios (figura 3.5). Ao sinal da figura 3.4 (b) foi aplicado o algoritmo KPCA e foram calculadas as pré-imagens pelo método das distâncias. O número de direcções seleccionadas no espaço

Algoritmo 10 Cálculo da pré-imagem

| | |
|----------------|--|
| entrada | matriz dos dados X Vector das distâncias no espaço entrada \mathbf{d} Vector das distâncias no espaço característico $\tilde{\mathbf{d}}$ Número de vizinhos n |
| | <p>Kernel RBF</p> <p>$[dummy, inx] = sort(\tilde{\mathbf{d}});$</p> <p>$L = X(:, inx(1 : n))$ Matriz dos n vizinhos mais próximos</p> <p>Dois casos:</p> <ol style="list-style-type: none"> 1. Considerando a distância no espaço de entrada $\mathbf{d} = \mathbf{d}(inx(1 : n));$ 2. Considerando a distância no espaço característico $\tilde{\mathbf{d}} = \tilde{\mathbf{d}}(inx(1 : n))$ <p>$[E, \Sigma, V] = svd(L(I - H))$</p> <p>$W = \Sigma V^T$</p> <p>$\mathbf{d0} = \sum [W \diamond W]^T$</p> <ol style="list-style-type: none"> 1. $w = -\frac{1}{2}\Sigma^{-1}V^T(\mathbf{d} - \mathbf{d0})$ 2. $w = -\frac{1}{2}\Sigma^{-1}V^T(\tilde{\mathbf{d}} - \mathbf{d0})$ <p>$\hat{\mathbf{z}} = E\mathbf{w} + L\frac{\mathbf{1}}{n}$</p> |
| saída | Pré-imagens $\hat{\mathbf{z}}$ |

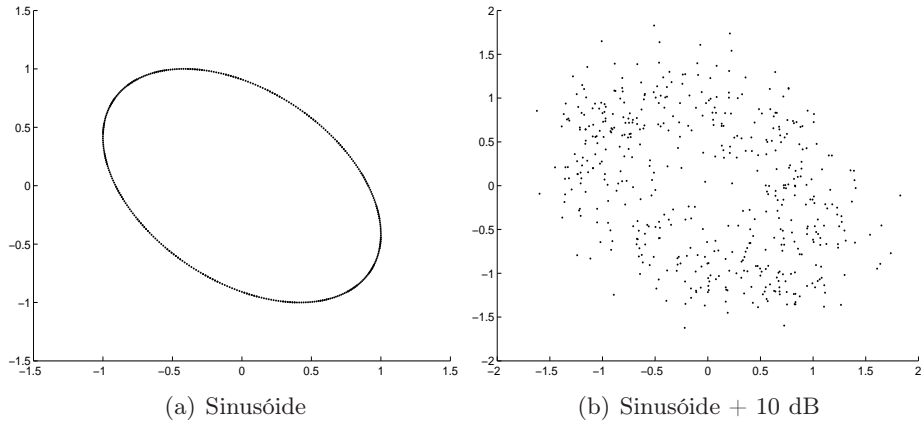


Figura 3.4: Representação do sinal sinusoidal e do sinal sinusoidal adicionado de ruído gaussiano (SNR=10dB) a duas dimensões.

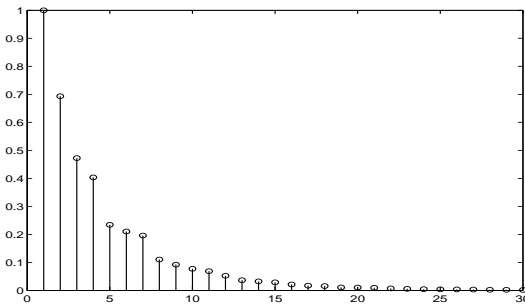


Figura 3.5: Representação dos 30 primeiros valores próprios normalizados, da matriz de kernel, dos dados da figura 3.4 (b).

característico do sinal foram:

- 1 direcção
- 2 direcções
- 3 direcções
- 4 direcções

A Figura 3.6 ilustra os resultados obtidos nos quatro casos mencionados. Verifica-se que à medida que o número de direcções seleccionadas aumenta, melhor se torna a reconstrução. Neste caso, o sinal reconstruído com quatro direcções é o que se aproxima mais do sinal original (figura 3.6 (d)).

No caso da aplicação do algoritmo SSA a este conjunto de dados, o número máximo de direcções a seleccionar seriam duas, dado que a matriz de covariância tem dimensão 2×2 (dois valores próprios). Para este caso, a aplicação do algoritmo SSA não permite reconstruir o sinal. Se fosse seleccionada uma única direcção, o sinal obtido seria uma recta e se fossem seleccionadas duas direcções, o sinal obtido seria o sinal original.

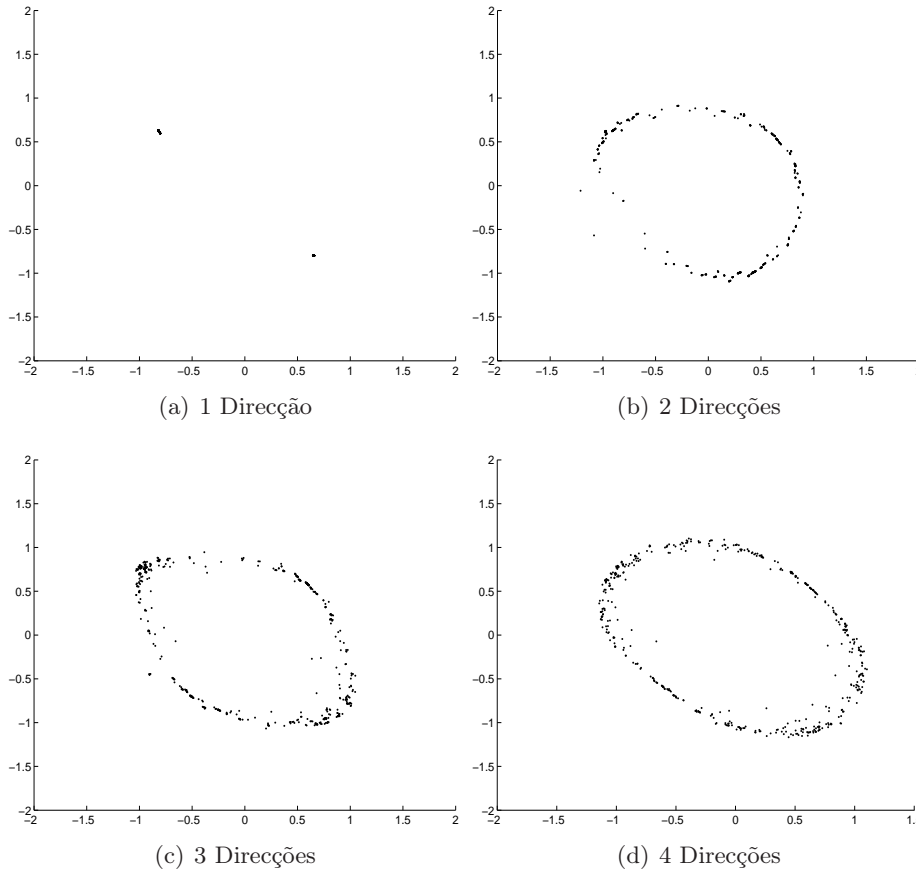


Figura 3.6: Reconstrução do sinal sinusoidal a duas dimensões adicionado com ruído (SNR=10dB) pelo método KPCA, considerando o algoritmo das distâncias no cálculo das pré-imagens. O número de direcções seleccionadas da matriz de kernel variou entre $k = 1$ até $k = 4$.

Para obtermos resultados semelhantes aos da figura 3.6(d) com o algoritmo SSA, seria necessário transformar o sinal unidimensional num sinal multidimensional de maior dimensão.

No caso da aplicação do algoritmo Local SSA, o número máximo de direcções a seleccionar em cada *cluster* é também depois, porém, a selecção de uma única direcção em cada *cluster* origina resultados muito semelhantes aos do KPCA com quatro direcções, como iremos ver na próxima secção.

3.8.2 Comparação gráfica dos algoritmos KPCA e Local SSA

Nesta secção são apresentados graficamente os resultados da aplicação do algoritmo KPCA e Local SSA a dois conjuntos de dados (sinal sinusoidal a 2D e mapa de Hénon)

Ao sinal sinusoidal representado na figura 3.4(a) com um SNR=20dB, foi aplicado o algoritmo KPCA (figura 3.7 (a)), considerando o kernel RBF com $\sigma^2 = 1$ e o método das distâncias para o cálculo das pré-imagens usando os dez vizinhos mais próximos no espaço característico. O número de direcções seleccionadas no espaço característico foi $k = 3$.

Para além desse algoritmo, foi aplicado o algoritmo Local SSA, usando dez *clusters* e selecci-

onando em cada *cluster* uma única direcção.

A figura 3.7 (b),(c),(d) ilustra a aplicação do algoritmo Local SSA considerando três fases

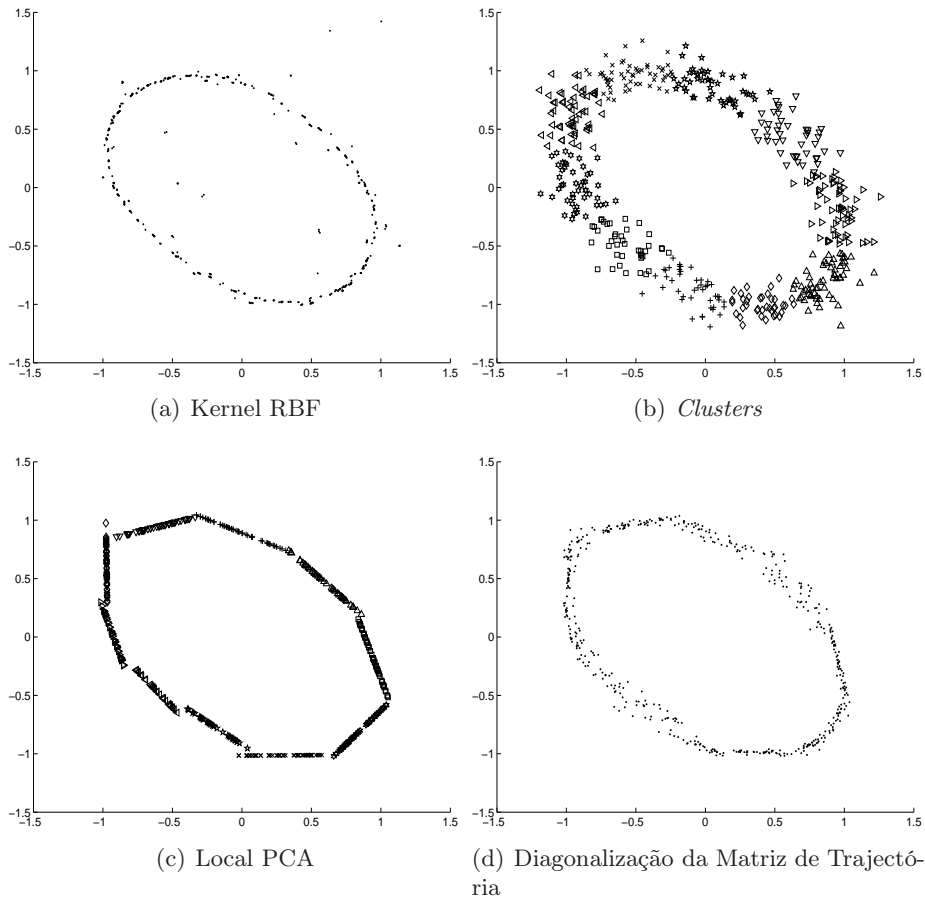


Figura 3.7: Comparação gráfica entre o algoritmo KPCA e o algoritmo SSA na reconstrução de uma sinusóide a duas dimensões com um SNR=20dB. São ilustradas quatro situações a duas dimensões:(a) reconstrução pelo algoritmo KPCA- método das distâncias, (b) agrupamento dos dados pelo algoritmo *k*-means, (c) aplicação do SVD a cada agrupamento - Local SVD e (d) diagonalização da matriz de trajectória - SSA.

do algoritmo:

- Distribuição do agrupamento dos dados - *cluster*, Figura 3.7 (b)
- Aplicação do algoritmo PCA a cada *cluster* - Local SSA, Figura 3.7 (c)
- Diagonalização da matriz de trajectória - SSA, Figura 3.7 (d)

Como ilustra a Figura 3.7, existe uma aproximação entre o algoritmo KPCA e o algoritmo Local SSA.

Foi considerada uma outra sequência dada pela equação não linear 3.44

$$y_{n+1} = 1 - 1.4y_n^2 + 0.3y_{n-1} \quad (3.44)$$

que apresenta um conteúdo de frequência espalhado por toda a gama. Esta sequência mapeada a duas dimensões corresponde ao mapa de Hénon (figura 3.8 (a)) [26].

Foram consideradas $N = 1000$ amostras acrescidas de um ruído aleatório gaussiano, cujo $\text{SNR}=20\text{dB}$ (figura 3.8 (b)).

Este sinal foi sujeito aos algoritmos KPCA e Local SSA, com o objectivo de eliminar o ruído adicionado para comparação da sua performance.

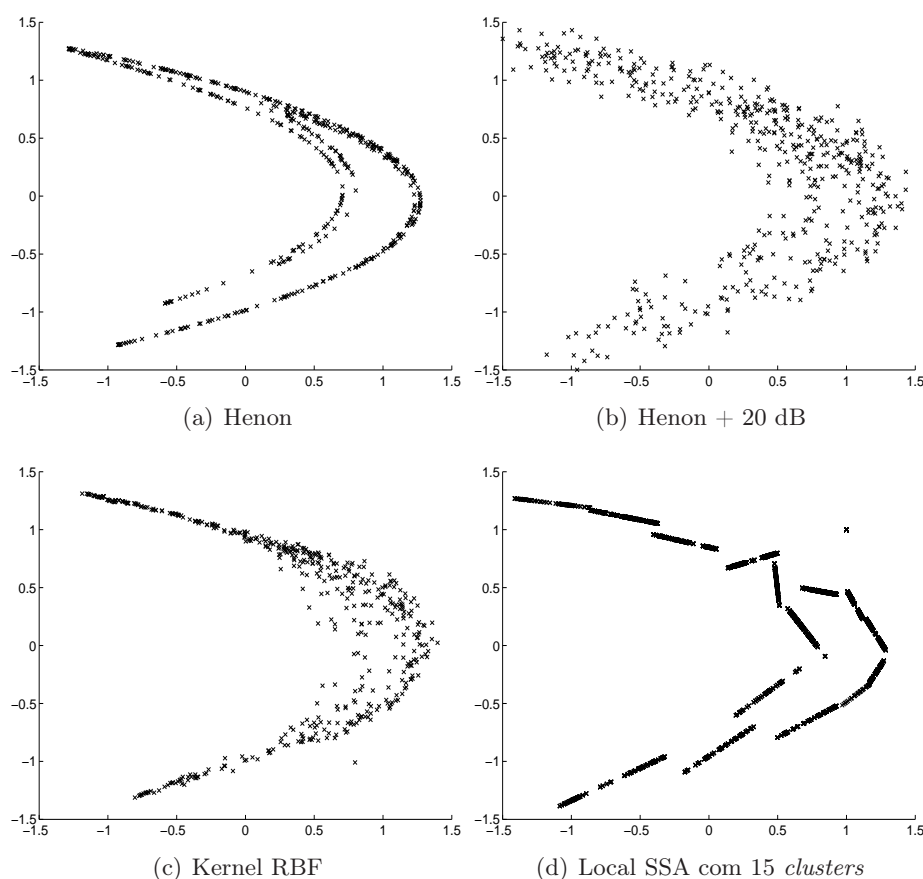


Figura 3.8: Comparação gráfica entre o algoritmo KPCA e o algoritmo SSA na reconstrução do mapa de Hénon com um $\text{SNR}=20\text{dB}$. São ilustradas 4 situações a duas dimensões:(a) mapa de Hénon, (b)mapa de Hénon com ruído gaussiano ($\text{SNR}=20\text{dB}$), (c) reconstrução pelo algoritmo Local SSA, (d) reconstrução pelo algoritmo KPCA- método das distâncias.

Pelas figuras 3.8 verifica-se que em ambos os casos existe eliminação do ruído. Na aplicação do algoritmo Local SSA, foram considerados $q = 15$ clusters e seleccionada uma única direcção em cada cluster, associada ao maior valor próprio. No caso do algoritmo KPCA, foi utilizado o kernel RBF, com $\sigma^2 = 1$ e aplicado o método das distâncias, seleccionando $k = 4$ direcções associadas aos quatro maiores valores próprios da matriz de kernel. Foram considerados dez vizinhos.

A eliminação do ruído é visível através da inspecção visual dos gráficos da figura 3.8 e através do MSE obtido entre a sequência diagonalizada obtida pelos algoritmos Local SSA e KPCA e a sequência original. O MSE é de 0.0096 para a sequência obtida pelo algoritmo KPCA e

0.0071 para a sequência obtida pelo algoritmo SSA. Os resultados apresentados para o sinal sinusoidal, sinal de banda estreita, bem como os resultados apresentados para a sequência de Hénon, sinal de banda larga, permitem concluir que os algoritmos KPCA e Local SSA apresentam resultados muito semelhantes. No caso do sinal sinusoidal, é possível reconstruir toda a trajetória do sinal, enquanto para o mapa de hénon é possível distinguir apenas duas das três trajetórias do sinal.

3.9 Seleção do Parâmetro RBF

Nas experiências efectuadas com o algoritmo KPCA foi utilizada a função de kernel RBF para o cálculo da matriz de kernel (eq. 3.8). Esta função está dependente de um parâmetro (σ^2) que é descrito na literatura como sendo a variância do sinal [22]. No entanto, resultados experimentais mostram que a escolha deste parâmetro não depende apenas da variância do sinal de entrada e que existe uma forte influência do parâmetro no funcionamento do algoritmo. Foi efectuada uma experiência com o sinal dente-de-serra com um período $P = 25$. Foi considerado um *embedding* $M = 30$ e dois níveis de ruído (SNR=20dB e SNR=0dB). Para cada um dos casos, foi aplicado o algoritmo KPCA, fazendo variar o parâmetro RBF, σ^2 , entre σ_s^2 e $10\sigma_s^2$, em que σ_s^2 representa a variância do sinal e calculou-se o MSE entre o sinal original e o sinal reconstruído. O número de direcções foi seleccionado por inspecção visual do decaimento dos valores próprios.

Pelas experiências efectuadas, verificou-se que o parâmetro RBF controla o decaimento dos valores próprios da matriz de kernel. Para um SNR elevado, o parâmetro que minimizou o MSE foi $\sigma^2 = 10\sigma_s^2$, originando um decaimento lento dos valores próprios (figura 3.9-1^a linha). No caso de um SNR baixo, se considerarmos o mesmo valor para o parâmetro RBF, verifica-se que há um decaimento lento, mas não o suficiente para eliminar o ruído do sinal (figura 3.9-2^a linha). Nesta situação, a diminuição do parâmetro ($\sigma^2 = 2\sigma_s^2$) origina um maior decaimento dos valores próprios, havendo uma maior sensibilidade da estrutura dos dados (figura 3.9-3^a linha).

Para os três casos apresentados foram escolhidas $k = 4$ direcções. O MSE para o primeiro caso foi MSE=0.0338, para o segundo foi MSE=0.41 e para o terceiro caso foi MSE=0.1623. Pelos resultados apresentados conclui-se que a extracção de informação de um sinal pelo algoritmo KPCA usando um kernel RBF está dependente do parâmetro RBF (σ^2). Este parâmetro influencia o decaimento dos valores próprios da matriz de kernel, podendo ser considerado um indicador da sensibilidade para a extracção de informação do sinal.

3.10 Conclusões

Neste capítulo foi apresentado o algoritmo KPCA, recorrendo a uma formulação alternativa da que é usualmente apresentada na literatura, [5]. Foi escolhida uma notação matricial para a

maioria das operações, simplificando a interpretação do algoritmo. É apresentada a descrição do pseudo-código de todos os passos do KPCA, para implementação do mesmo (algoritmos 4-10).

Foi escolhido um caso particular de kernel (kernel RBF) e usado o método das distâncias para o cálculo da pré-imagem. Estudou-se o desempenho do algoritmo, variando três parâmetros: σ^2 , dimensão do *embedding* e o número de direcções a seleccionar no espaço característico.

Os resultados obtidos foram comparados com o algoritmo Local SSA e concluiu-se que ambos apresentam uma performance muito semelhante. No entanto, o algoritmo KPCA é mais complexo na selecção dos parâmetros e necessita de um maior tempo de execução para reconstrução do sinal no espaço de entrada.

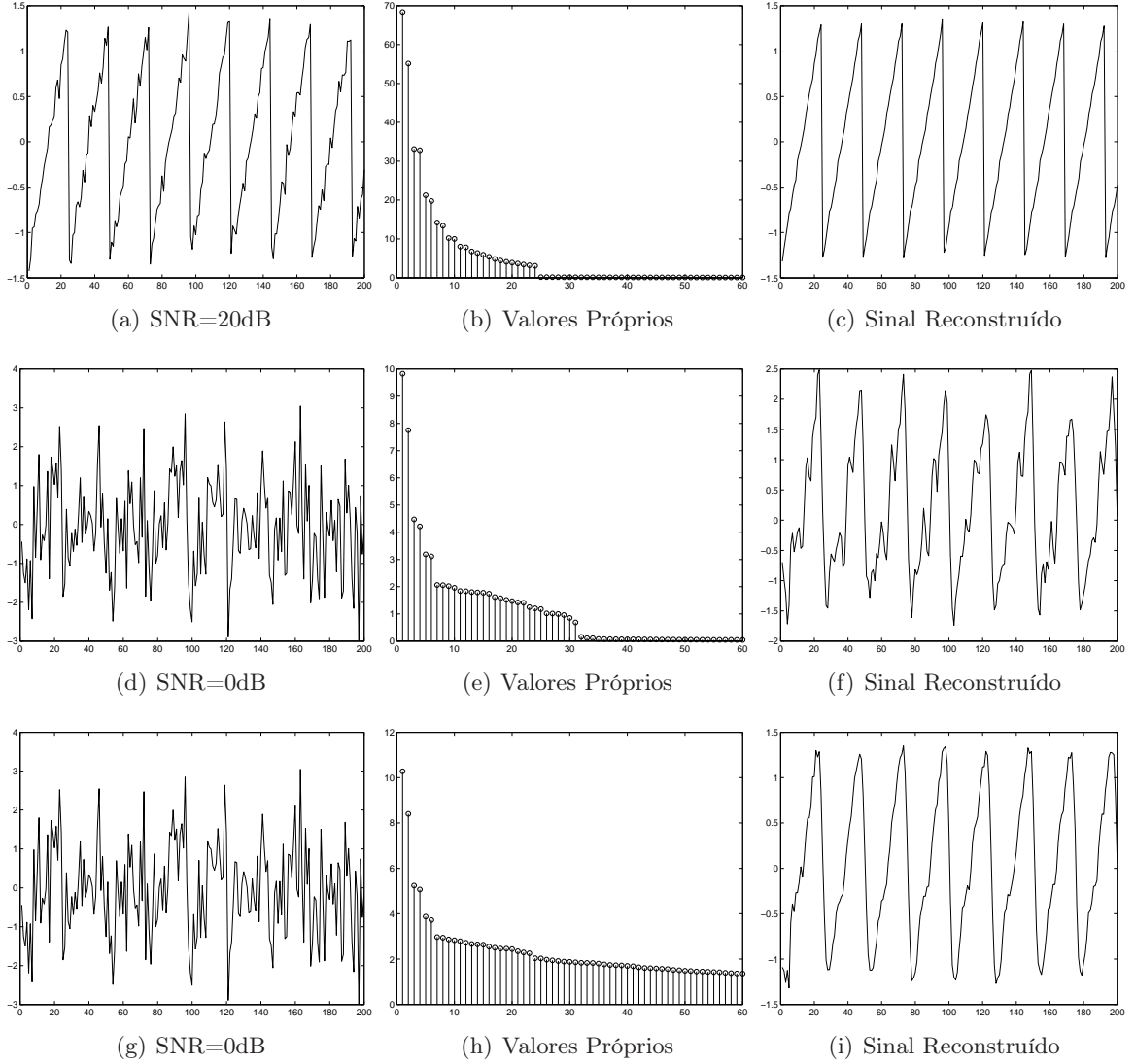


Figura 3.9: Reconstrução de um sinal dente-de-serra pelo algoritmo KPCA, usando o método das distâncias e considerando dois níveis de ruído: (a) Sinal dente-de-serra adicionado com SNR=20dB, (b) Gráfico dos valores próprios da matriz do Kernel RBF, considerando $\sigma^2 = 10\sigma_s^2$, (c) Sinal dente-de-serra reconstruído, (d) Sinal dente-de-serra adicionado com SNR=0dB, (e) Gráfico dos valores próprios da matriz do Kernel RBF, considerando $\sigma^2 = 10\sigma_s^2$, (f) sinal dente-de-serra reconstruído, (g) Sinal dente-de-serra adicionado com SNR=0dB, (h) Gráfico dos valores próprios da matriz do Kernel RBF, considerando $\sigma^2 = 2\sigma_s^2$, (i) sinal dente-de-serra reconstruído.

Capítulo 4

Dimensão do Sub-espço do Sinal

A estimação do número de valores próprios de uma matriz de covariância associados ao sinal é um problema da análise de séries temporais na eliminação de ruído e reconhecimento de padrões.

Existem alguns critérios para estimar os valores próprios associados ao ruído do sinal, nomeadamente, o critério MDL, o AIC, o *Bayesian Information Criterion* (BIC) e o critério da variância explicada, entre outros.

No presente capítulo vão ser apresentados os critérios MDL, AIC e o critério da variância explicada. Foram ainda efectuadas algumas simulações a um determinado modelo de mistura com o objectivo de comparar a performance dos critérios, alterando alguns parâmetros do modelo. Por fim, são apresentadas algumas conclusões.

4.1 Formulação do Problema

Consideremos uma matriz de covariância definida por:

$$C = \psi + \sigma^2 I \quad (4.1)$$

onde $\sigma^2 I$ representa a matriz de covariância do ruído e ψ representa uma matriz simétrica, semipositiva, q -dimensional com $q < p$, i.e. $(p - q)$ valores próprios são iguais a zero. Consequentemente, $(p - q)$ valores próprios da matriz C são iguais a σ^2 , i.e.

$$\lambda_{q+1} = \lambda_{q+2} = \dots = \lambda_p = \sigma^2 \quad (4.2)$$

Na prática, a igualdade da equação 4.2 nunca se verifica, o que torna difícil a estimação do número de valores próprios $(p - q)$ associados ao ruído do sinal, sendo necessário utilizar critérios baseados na análise residual.

4.2 Critérios

Existem critérios para calcular o número de valores próprios da matriz de covariância associados ao sinal para a redução do ruído no conjunto de dados de entrada.

Assume-se que os valores próprios da matriz de covariância do modelo da equação 4.1 se encontram por ordem decrescente:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \quad (4.3)$$

Nas secções seguintes são apresentados três critérios: o critério da VE, o critério MDL e o critério AIC.

4.2.1 Variância Explicada

O critério usualmente mencionado na literatura para redução da dimensão do conjunto de dados é baseado no “scree plot”, i.e., no gráfico da soma acumulada dos valores próprios normalizados versus o número de componentes. A partir desse gráfico, existem algumas considerações a ter em conta:

- eliminar os valores próprios que tenham um valor menor do que a média de todos os valores próprios;
- inspeccionar visualmente o gráfico dos valores próprios acumulados e verificar se há um decréscimo acentuado entre eles. Muitas vezes é referido que se não há um decréscimo acentuado entre os valores próprios, não é possível reduzir a dimensão do conjunto de dados;
- seleccionar o número de valores próprios da matriz de correlação do sinal, de forma a representarem uma determinada percentagem da variância do sinal original [22] (na prática $0.9 \leq VAR \leq 0.95$). O número de valores próprios é reduzido de p para k , sendo desprezados $(p - k)$ valores próprios, que se assume estarem associados ao ruído do modelo

$$VE = \frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_k + \dots + \lambda_p} \geq VAR \quad (4.4)$$

4.2.2 MDL e AIC

O critério AIC e o critério MDL foram originalmente propostos em [27] e [28] e popularizados na análise de processamento de sinal por [29].

Existem alguns estudos [30, 6, 7, 18] que apresentam o AIC e o MDL como critérios de selecção da estimação da dimensão do subespaço do sinal em diversas aplicações (selecção da ordem de modelos autoregressivos). Os critérios MDL e AIC são usados para estimar a ordem q ,

Algoritmo 11 Variância Explicada

| | |
|---------|--|
| entrada | $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ Valor de VAR |
| | $S = \sum \lambda_i, \quad i = 1, \dots, p$ $S_k = \frac{\sum \lambda_i}{S} \leq VAR \quad i = 1, \dots, k$ |
| saída | nova dimensão dos dados: k |

baseados na aplicação de *Information Theoric Criteria* (ITC), descrita da seguinte forma [29]:

$$\begin{aligned}
 AIC(k) &= -2\ln f(X/\hat{\theta}) + 2K \\
 MDL(k) &= -\ln f(X/\hat{\theta}) + \frac{1}{2}K\ln N
 \end{aligned}
 \tag{4.5}$$

onde $\hat{\theta}$ representa a estimação da máxima verosimilhança dos parâmetros do vector θ , K o número de parâmetros livres de θ e N representa a dimensão do conjunto de dados e $f(X/\hat{\theta})$ representa a função de probabilidade condicional parametrizada por $\hat{\theta}$.

A matriz de covariância dos dados é caracterizada por

$$C = \psi + \sigma^2 I \tag{4.6}$$

onde ψ representa uma matriz semipositiva definida de dimensão $k \in \{0, 1, \dots, p - 1\}$ e σ um escalar desconhecido.

Usando o teorema da representação espectral de álgebra linear, C pode ser expressa da seguinte forma:

$$C = \sum_{i=1}^k (\lambda_i - \sigma^2) v_i v_i^T + \sigma^2 I \tag{4.7}$$

onde $\lambda_1, \dots, \lambda_k$ e v_1, \dots, v_k representam os valores próprios e os vectores próprios respectivamente da matriz C . A determinação do subespaço k é baseada na estimação da máxima verosimilhança dos parâmetros do vector θ , [31]

$$\theta = (\lambda_1, \dots, \lambda_k, \sigma^2, v_1, \dots, v_k)$$

onde

$$\begin{aligned}
\hat{\lambda}_i &= \lambda_i, \quad i = 1, \dots, k \\
\sigma^2 &= \frac{1}{p-k} \sum_{i=k+1}^p \lambda_i \\
\hat{v}_i &= v_i, \quad i = 1, \dots, k
\end{aligned} \tag{4.8}$$

Esta estimação é usada para encontrar o valor de $L(\hat{\theta}) = \ln f(X/\hat{\theta})$

$$L(\hat{\theta}) = N(p-q) \ln \frac{\prod_{i=k+1}^p \lambda_i^{\frac{1}{p-k}}}{\frac{1}{p-k} \sum_{i=k+1}^p \lambda_i} \tag{4.9}$$

onde o argumento do logaritmo depende da razão entre a média geométrica e a média aritmética dos $(p-k)$ valores próprios.

O valor de K da equação 4.5 é obtido, considerando o número de parâmetros de θ e a complexidade da estimação. Se o conjunto de dados for real

$$K = k + 1 + pk - \frac{k^2}{2} - \frac{k}{2} = pk - \frac{k^2}{2} + \frac{k}{2} + 1 \tag{4.10}$$

Se o conjunto de dados for complexo, então a matriz C é complexa e cada coluna tem uma parte real e imaginária, logo o parâmetro K é

$$K = k + 1 + 2(pk - \frac{k^2}{2} - \frac{k}{2}) \cong k(2p - k) \tag{4.11}$$

Substituindo as equações 4.9 e 4.10 nas equações 4.5, obtemos a expressão geral para o critério AIC e MDL aplicado a dados reais:

$$AIC(k) = -2N(p-q) \ln \frac{\prod_{i=k+1}^p \lambda_i^{\frac{1}{p-k}}}{\frac{1}{p-k} \sum_{i=k+1}^p \lambda_i} + 2(pk - \frac{k^2}{2} + \frac{k}{2} + 1) \tag{4.12}$$

$$MDL(k) = -N(p-q) \ln \frac{\prod_{i=k+1}^p \lambda_i^{\frac{1}{p-k}}}{\frac{1}{p-k} \sum_{i=k+1}^p \lambda_i} + \frac{1}{2}(pk - \frac{k^2}{2} + \frac{k}{2} + 1) \ln(N) \tag{4.13}$$

O número de sinais k é determinado pelo valor de $k \in \{0, 1, \dots, p-1\}$, que minimiza o AIC e o MDL.

Assumindo que a dimensão do sinal é q , existem três observações a fazer [30]:

1. Se os valores próprios do sinal, $\lambda_1, \dots, \lambda_q$ são da mesma ordem de grandeza e se estão nitidamente separados dos valores próprios do ruído, $\lambda_{q+1}, \dots, \lambda_p$, então a ordem seleccionada é menor do que a ordem do modelo, $k < q$.
2. Se os valores próprios do ruído não se encontrarem estritamente agrupados, o ITC pode

Algoritmo 12 MDL

entrada $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$

Dimensão dos dados: N

$mdl = []$

for $i=0 : p-1$

$Geom = \prod \lambda_i^{\frac{1}{p-k}}$

$Arit = \frac{1}{p-k} \sum \lambda_i$

$K = pk - \frac{k^2}{2} + \frac{k}{2} + 1$

$mdl = [mdl; -N(p-k) \ln \frac{Geom}{Arit} + \frac{1}{2}K \ln(N)]$

end

$[D \ pos] = \min((mdl(:, 1)))$

if $(mdl(pos, 2) == 0)$

$k = p$

else

$k = mdl(pos, 2)$

end

saída nova dimensão dos dados: k

Algoritmo 13 AIC

| | |
|---------|--|
| entrada | $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ Dimensão dos dados: N |
| | $aic = []$ for $i=0 : p-1$ $Geom = \prod \lambda_i^{\frac{1}{p-k}}$ $Arit = \frac{1}{p-k} \sum \lambda_i$ $K = pk - \frac{k^2}{2} + \frac{k}{2} + 1$ $aic = [aic; -2N(p-q)\ln \frac{Geom}{Arit} + 2K]$ end $[D \ pos] = \min((aic(:, 1)))$ if $(aic(pos, 2) == 0)$ $k = p$ else $k = aic(pos, 2)$ end |
| saída | nova dimensão dos dados: k |

ignorar qualquer desfasamento entre os valores próprios do sinal e os valores próprios do ruído, estimando uma ordem superior à ordem do model ($k > q$).

3. Se os valores próprios do sinal e do ruído não se encontrarem bem separados e se os valores próprios do ruído forem da mesma ordem de grandeza, a ordem do modelo estimada é superior à ordem do modelo ($k > q$).

4.3 Resultados

A selecção da dimensão do sub-espaco para um conjunto de sinais L -dimensional é um problema que surgiu da implementação e estudo do algoritmo *dAmuse* [3].

Em [3] foi considerada a VE=95% como critério para a selecção das direcções principais do primeiro passo do algoritmo.

Para estudar a performance dos três critérios apresentados, foi considerado um conjunto de sinais artificiais (\mathbf{s}) (figura 2.1) misturados com uma matriz de mistura aleatória A e adicionados de ruído gaussiano ($\boldsymbol{\nu}$), i.e. $\mathbf{x}[n] = A\mathbf{s}[n] + \boldsymbol{\nu}[n]$.

Utilizou-se um segmento de N amostras e foram efectuados M atrasos aos sinais (equação 2.11) obtendo um espaco de entrada de dimensão LM .

A este conjunto de sinais foram aplicados os critérios MDL, AIC e VE fazendo variar alguns parâmetros (número de amostras, dimensão da matriz de mistura e SNR).

Foram efectuados três tipos de estudos com os sinais artificiais:

- comparação global dos critérios;
- aplicação dos critérios a MSSA;
- aplicação dos critérios ao Local SSA.

4.3.1 Comparação global dos critérios MDL, AIC e VE

As simulações efectuadas pretenderam estudar a performance global dos três critérios, na selecção da dimensão do subespaco, fazendo variar o SNR=0, 5, 10, 15, 20 dB e os atrasos introduzidos nos sinais de entrada $M = 1, \dots, 11$. Foram considerados os sinais da figura 2.1, misturados com dois tipos de matrizes aleatórias: uma matriz de mistura de dimensão 3×3 e uma matriz de mistura 5×3 .

Foram efectuadas mil iterações para cada uma das combinações possíveis entre os dois parâmetros de variação, calculando-se depois a percentagem de valores próprios seleccionados, i.e, o coeficiente entre o número de valores próprios seleccionados por cada um dos critérios e o número total de valores próprios da matriz de correlação dos dados, figura 4.1. Pela análise da figura 4.1, verifica-se que os critérios MDL e AIC se comportam com a mesma tendência. Tal como é também referido em [30], a ordem do modelo diminui à medida que o número de atrasos introduzidos nos sinais de entrada aumenta. O critério da variância explicada apresenta um comportamento inverso, ou seja, a ordem do modelo aumenta à medida que o

número de atrasos introduzidos nos sinais de entrada aumenta.

Se o número de misturas for maior do que o número de sinais de entrada (caso da matriz 5×3) a percentagem dos valores próprios seleccionados por cada critério é mais baixa, quando comparada com a percentagem obtida para um número de misturas igual ao número de sinais. Esta situação verifica-se porque há um aumento artificial da dimensão do conjunto de dados (introdução de mais dois valores próprios próximos de zero para cada atraso efectuado aos sinais).

Para além de terem sido feitas simulações com os sinais da figura 2.1, também foram efectuadas simulações para um outro grupo de sinais, de banda estreita, sendo os resultados análogos aos apresentados.

4.3.2 Aplicação dos critérios ao MSSA

Nesta secção vão ser apresentados alguns resultados que evidenciam a performance dos três critérios, tendo em conta três factores de variação:

- a dimensão do conjunto de dados, ($N = 50$ e $N = 5000$);
- a matriz de mistura aleatória, ($H = 3 \times 3$ e $H = 5 \times 3$);
- o coeficiente sinal ruído, ($SNR = 5$ e $SNR = 20$);

para um atraso fixo, $M = 5$ e efectuando mil iterações.

Nas figuras 4.2, 4.3, 4.4, 4.5 (a) e (c) estão representados, na escala logarítmica, a média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração. Nas figuras 4.2, 4.3, 4.4, 4.5 (b) e (d) está representada a estimação da *Probability Density Function* (PDF), do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração.

Os resultados apresentados nas figuras 4.2, 4.3, 4.4 e 4.5, mostram que a selecção da dimensão do subespaço depende da existência de um decaimento abrupto dos valores próprios que é influenciado pela dimensão do conjunto de dados, pela dispersão dos valores próprios do ruído e pela dimensão da matriz de mistura.

Pela análise das figuras 4.2, 4.3, 4.4 e 4.5 observa-se que:

1. quanto maior for o número de amostras dos sinais ($N = 5000$) melhor é o desempenho dos critérios MDL e AIC, sendo o critério MDL mais preciso na selecção da dimensão do conjunto de dados, figuras 4.2, 4.3, 4.4 e 4.5.
2. quando o SNR é elevado, a estimação da dimensão é mais precisa, já que o aumento da dimensão do conjunto de dados proporciona um desfasamento na grandeza dos valores próprios, permitindo agrupar aqueles que estão associados ao ruído (figura 4.2 e 4.3 (c)). Quando o SNR é baixo, não é possível separar de uma forma tão clara os valores

| N | SNR | | |
|------|--------|--------|--------|
| | 20 dB | 10 dB | 5 dB |
| 50 | 0.0070 | 0.0520 | 0.0958 |
| 5000 | 0.0007 | 0.0095 | 0.0230 |

Tabela 4.1: Erro médio absoluto entre a variância do ruído e a variância dos valores próprios desprezados pelo critério MDL, considerando uma matriz de mistura 5×3 .

próprios associados ao ruído do sinal, existindo uma pior performance dos critérios de selecção (figura 4.4 e 4.5).

- quando o número de misturas é maior do que o número de sinais, o critério MDL selecciona a dimensão do conjunto de dados com maior precisão (figura 4.2 e 4.3).

Pelos resultados apresentados verifica-se que o critério da variância explicada é apenas influenciado pelo SNR. Quanto maior for o SNR, mais baixa é a gama de valores seleccionados. À medida que o SNR aumenta, mais elevada se torna a dimensão seleccionada pelo critério VE. Comparando o desempenho do critério MDL com o critério AIC, verifica-se que o critério MDL é mais preciso na selecção da dimensão do subespaço enquanto o critério AIC, na maior parte das vezes, sobrestima essa mesma dimensão.

Foi efectuado um estudo comparando a variância do ruído estimado pelo critério MDL com a variância do ruído original, através do erro médio absoluto entre as duas variâncias (tabela 4.1). Foram considerados três níveis de ruído, SNR=20dB, 10dB e 5dB, duas dimensões diferentes $N = 50$ e $N = 5000$ e uma matriz de mistura 5×3 . A tabela 4.1 mostra a dependência entre o erro de estimação e a dimensão do conjunto de dados. Verifica-se que o erro médio absoluto é menor quando o número de amostras é elevado ($N = 5000$). Além disso, à medida que o SNR diminui, maior é o erro de estimação do ruído do sinal. Os resultados apresentados na tabela 4.1 confirmam a precisão de estimação por parte do critério MDL na selecção da dimensão do subespaço.

4.4 Aplicação do critério MDL ao SSA

No capítulo 2, foi aplicado o critério MDL em cada *cluster*, para a selecção do número de direcções associadas ao ruído do sinal, considerando SNR=5dB. Os resultados apresentados nas tabelas C.1, C.2 e C.3 do apêndice C contêm a informação do número de direcções seleccionadas, bem como a dimensão de cada *cluster*.

Analisando os gráficos das figuras 2.11 e 2.13, verifica-se que para $M = 11$ atrasos, o aumento do número de *clusters* para os sinais sinusoidal e dente-de-serra, piora os resultados. Para este caso, o critério MDL não encontra um decaimento brusco entre os valores próprios, sobrestimando a dimensão do subespaço.

Na figura 4.6, está representado o logaritmo dos valores próprios do sinal *funny* com $q = 3$ *clusters*, quando são utilizados $M = 11$ e $M = 36$ atrasos.

No terceiro *cluster*, para $M = 11$ (figura 4.6 (c)) o critério MDL seleccionou $k = 11$

direcções, devido ao decaimento lento dos valores próprios.

Este tipo de comportamento dos valores próprios da matriz de covariância normalizada faz com que os resultados apresentados na secção 2.8 não sejam os melhores. Existem outras situações idênticas à apresentada na figura 4.6. Uma dessas situações é o caso do sinal sinusoidal, quando são considerados $M = 11$ atrasos e $q = 5$ *clusters*. Podemos ver pela figura 4.7 que não existe um decaimento abrupto entre os valores próprios, logo o critério MDL selecciona todas as direcções, não havendo eliminação de ruído.

4.5 Conclusões

Neste capítulo estudou-se a performance dos critérios MDL, AIC e VE na selecção da dimensão do subespaço. Foram efectuadas várias experiências, com sinais artificiais variando a dimensão do conjunto de dados, a matriz de mistura e o SNR.

Para um número de amostras reduzido, os critérios apresentam uma má performance. Com o aumento da dimensão do conjunto de dados, há um desfasamento dos valores próprios, agrupando os valores próprios associados ao ruído e os valores próprios associados ao sinal. Nesta situação, o critério MDL é o que estima melhor a dimensão do subespaço.

Se considerarmos um SNR baixo, o decaimento dos valores próprios é muito lento, originando uma sobrestimação da dimensão do espaço por parte do critério MDL e AIC.

Comparando os resultados obtidos, conclui-se que o critério MDL estima de uma forma mais consistente a dimensão do subespaço, enquanto o critério AIC fornece dimensões mais elevadas, revelando a inconsistência estatística em sobrestimar a dimensão, nomeadamente quando o número de amostras é muito elevado, como é referido em [30].

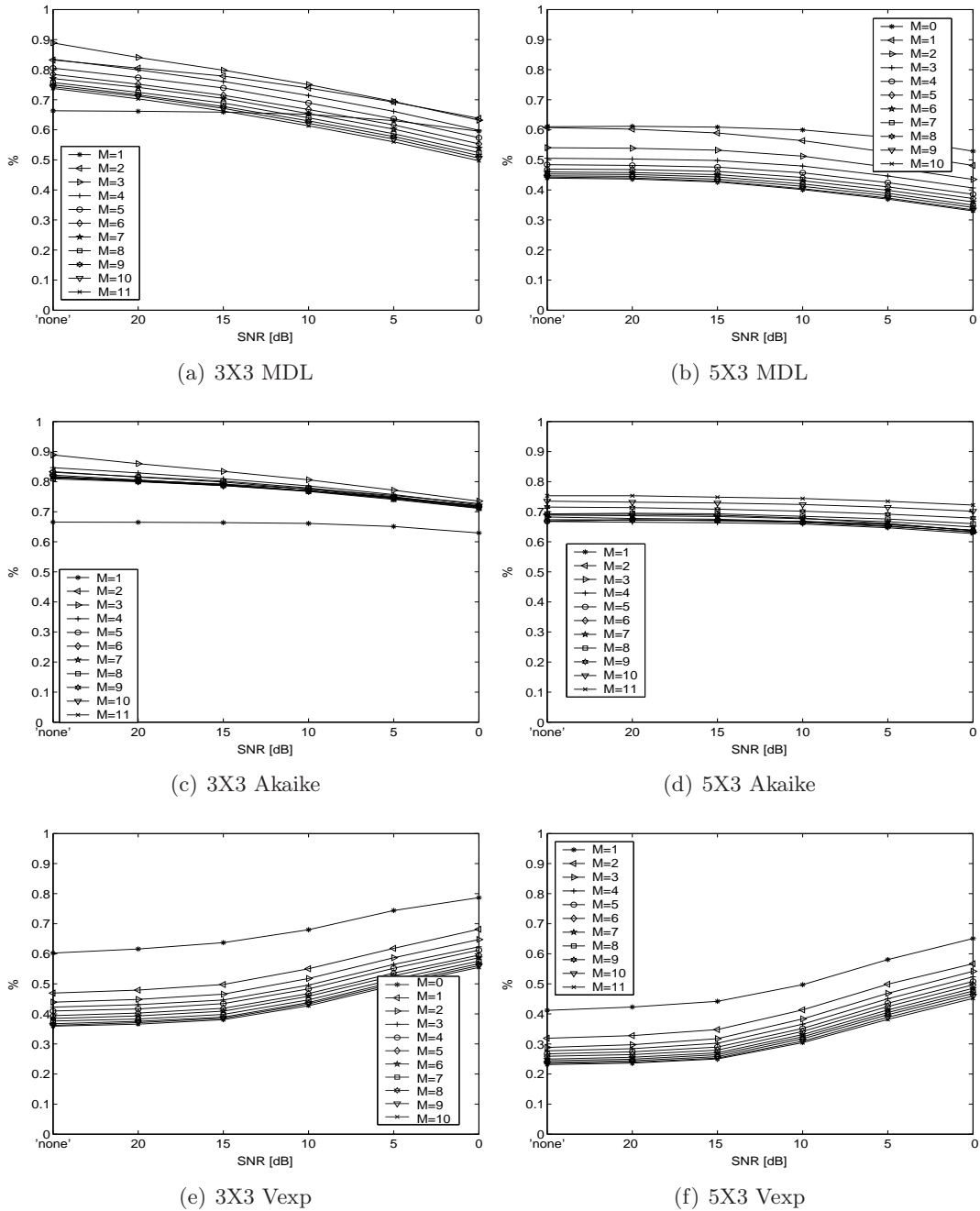


Figura 4.1: Percentagem normalizada do número de valores próprios seleccionados pelos critérios MDL, AIC e VE, considerando dois tipos de matrizes de misturas ($H = 3 \times 3$ e $H = 5 \times 3$), fazendo variar o número de atrasos ($M=1, \dots, 11$) e o SNR=(none 0 5 10 15 20).

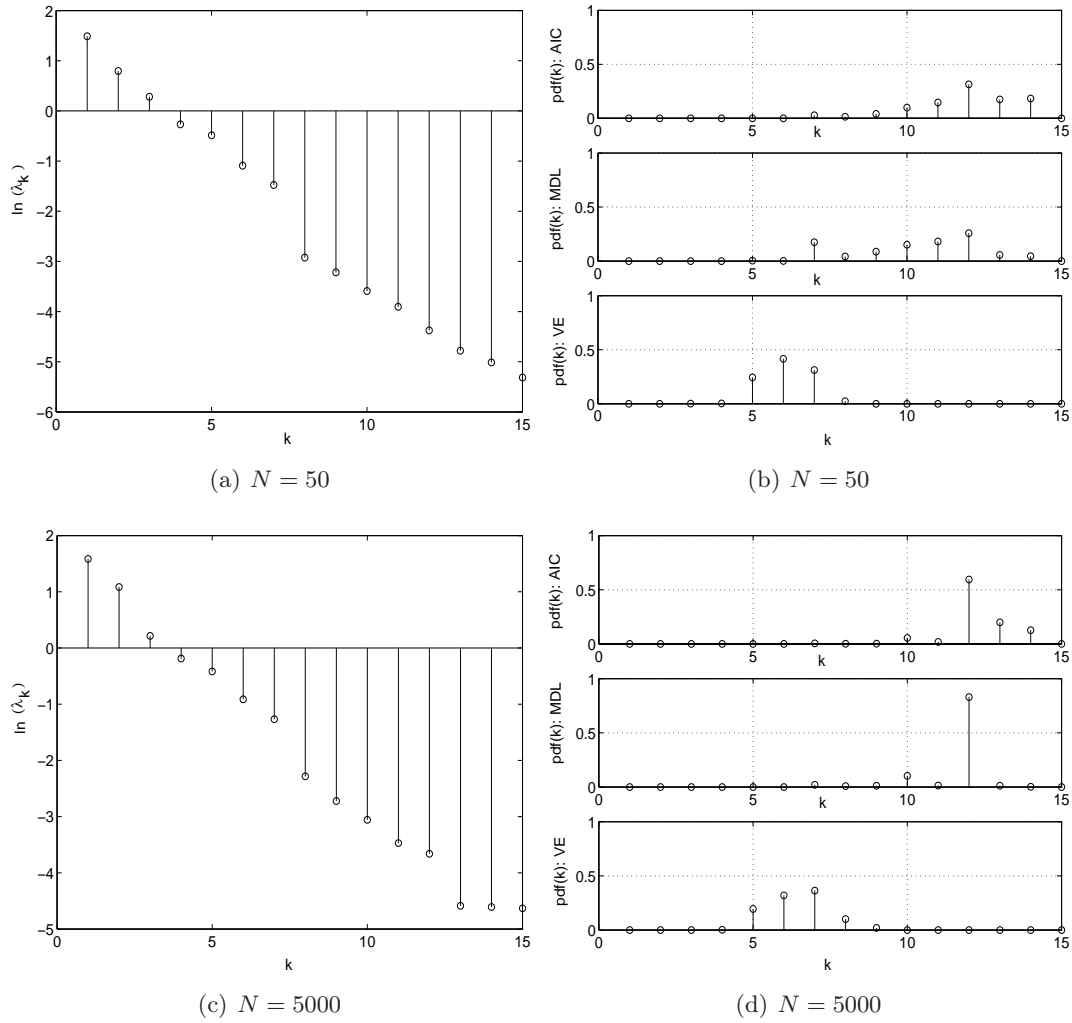


Figura 4.2: Simulação de 1000 iterações ($H = 3 \times 3$, $SNR = 20$, $M = 5$) e considerando $N = 50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração.

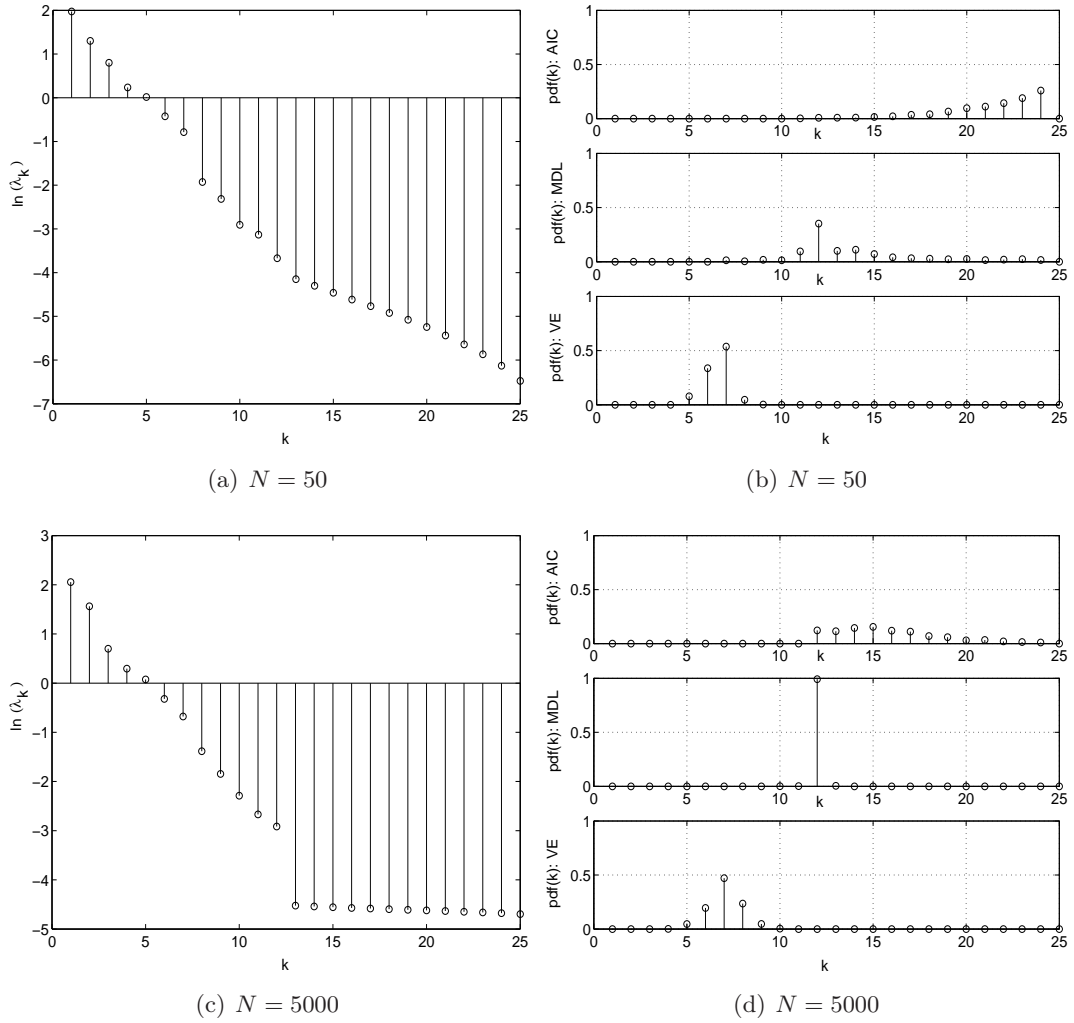


Figura 4.3: Simulação de 1000 iterações ($H = 5 \times 3$, $SNR = 20$, $M = 5$) e considerando $N=50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração.

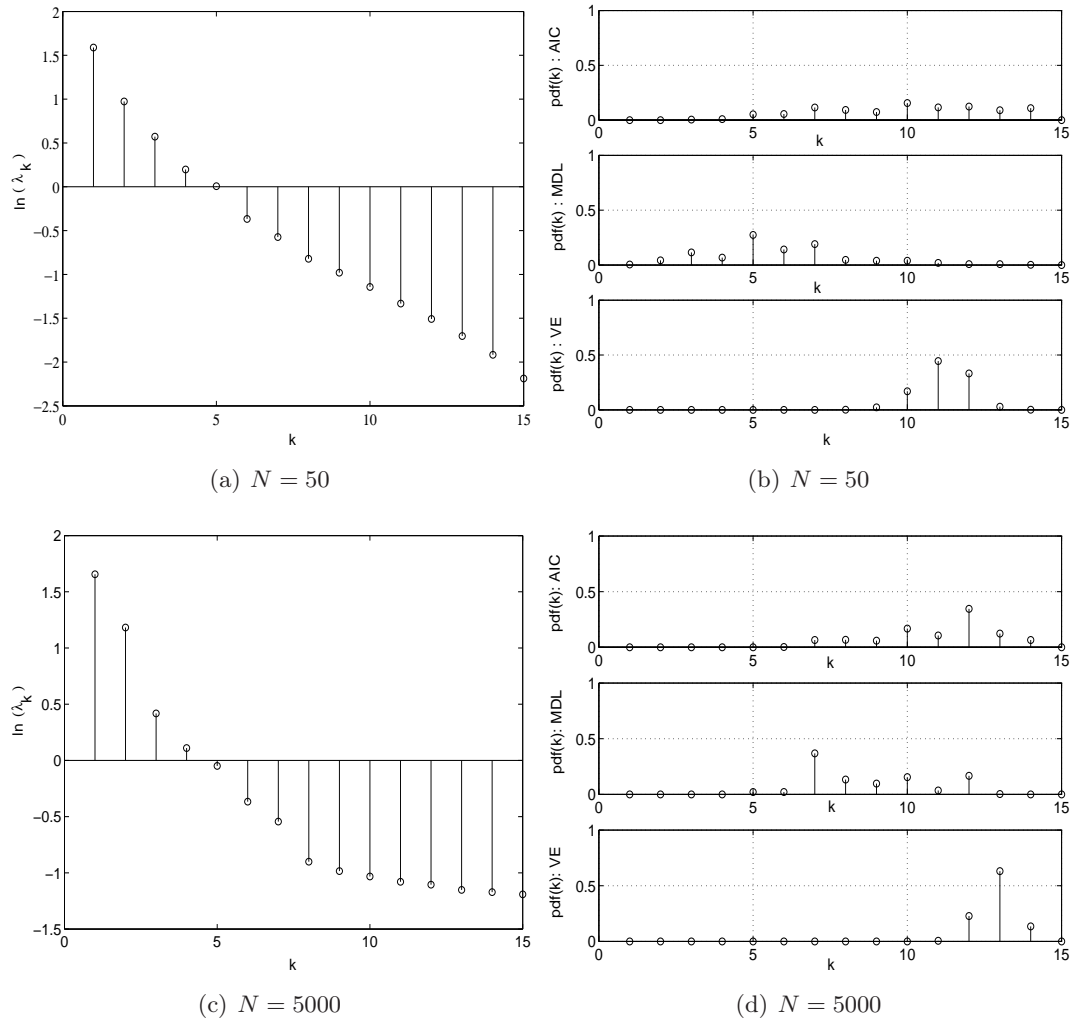


Figura 4.4: Simulação de 1000 iterações ($H = 3 \times 3$, $SNR = 5$, $M = 5$) e considerando $N = 50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração.

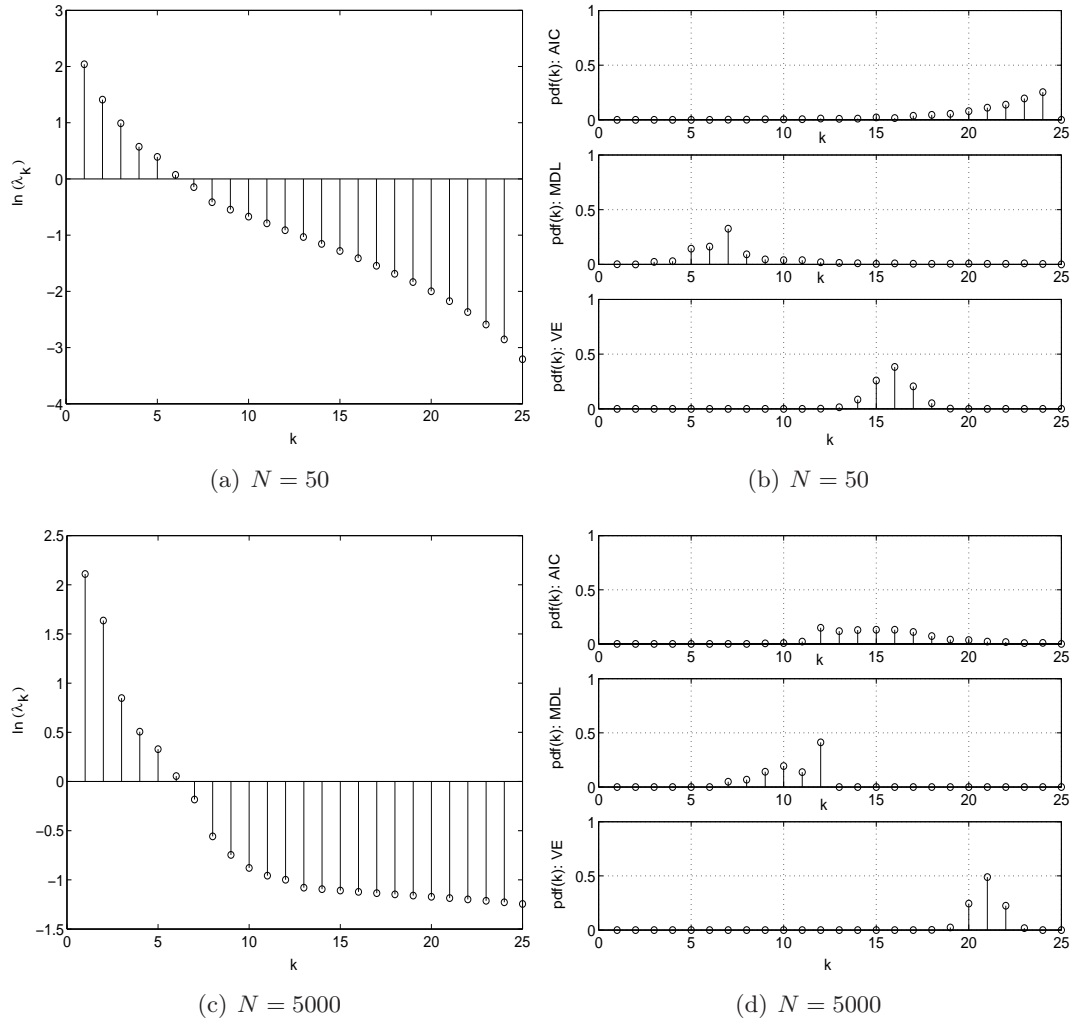


Figura 4.5: Simulação de 1000 iterações ($H = 5 \times 3$, $SNR = 5$, $M = 5$) e considerando $N = 50$ ((a) e (b)) e $N = 5000$ ((c) e (d)): (a) e (c) representam o logaritmo da média dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada iteração; (b) e (d) representam a estimação da função densidade de probabilidade, $pdf(k)$ do critério AIC, MDL e VE versus o número de valores próprios k seleccionados em cada iteração.

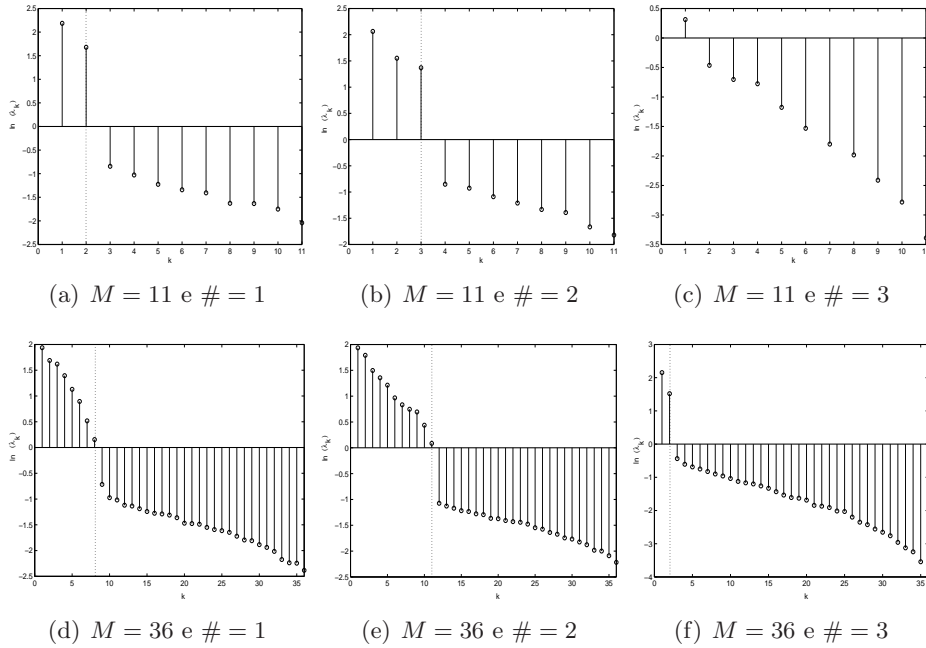


Figura 4.6: Os gráficos (a), (b), (c) representam o logaritmo dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$, versus o número de valores próprios k seleccionados em cada um dos três *clusters* e os gráficos (d), (e), (f) representam o logaritmo dos valores próprios estimados da matriz de covariância normalizada $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada um dos três *clusters*.

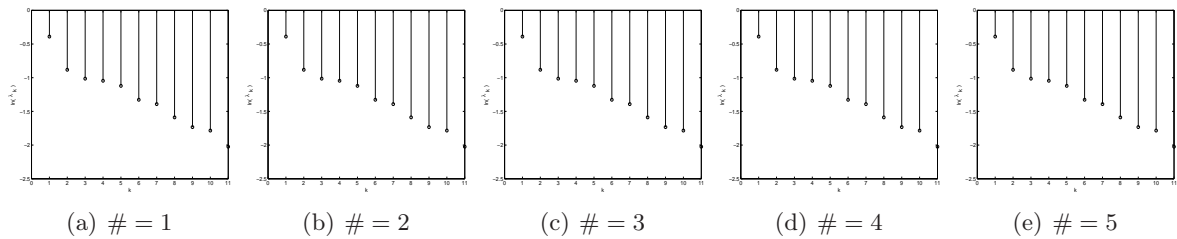


Figura 4.7: Os gráficos (a), (b), (c),(d), (e) representam o logaritmo dos valores próprios estimados da matriz de covariância normalizada do sinal sinusoidal $\ln(\lambda_k)$ versus o número de valores próprios k seleccionados em cada um dos cinco *clusters*, quando são considerados $M = 11$ atrasos.

Capítulo 5

Aplicação dos algoritmos a sinais EEG

5.1 Introdução

Neste capítulo são apresentados alguns resultados da aplicação dos algoritmos KPCA e Local SSA na eliminação de artefactos no sinal EEG. Essencialmente, os artefactos removidos foram os movimentos oculares, a actividade eléctrica e movimentos provocados pelo paciente ou pelos eléctrodos.

Inicialmente, descreve-se de forma sucinta as principais características do sinal EEG para o estudo em causa e o método de registo do mesmo. Posteriormente, são apresentados os resultados da aplicação dos algoritmos Local SSA e KPCA a um canal EEG para a eliminação dos movimentos oculares, sendo ilustrada uma aplicação do algoritmo Local SSA, de forma automática, a um grupo de canais EEG para a eliminação de outros artefactos. Por fim, são apresentadas algumas conclusões.

5.2 O Electroencefalograma

O EEG é um sinal eléctrico produzido pelo cérebro, cujas flutuações podem variar de amplitude, de duas a várias centenas de micro volts, quando registadas a partir de eléctrodos colocados no escalpe. Essas flutuações cobrem uma variação de frequências de aproximadamente 0.5 a 70 Hz, denominada banda de frequência. Esta banda é dividida em quatro subcategorias, identificadas pelas letras gregas: alfa, beta, teta e delta (figura 5.1).

- **Delta:** Tem uma amplitude entre 20 a 200 μV e uma gama de frequência entre 0.5 e 3 Hz;
- **Teta:** Tem uma amplitude entre 20 a 100 μV e uma gama frequência entre 4 e 7 Hz;
- **Alfa:** Esta onda pode dividir-se em Alfa Baixo de 8 a 10 Hz e em Alfa Alto, de 11 a 13 Hz. Ambas têm uma amplitude de 20 a 60 μV ;

- **Beta:** Caracteriza-se por uma onda irregular com uma gama de frequência de 14 a 30 Hz e com uma amplitude entre 2 a 20 μV . Em alguns casos esta onda divide-se em três grupos: Beta: 14 – 16Hz; Beta 1: 16 – 20Hz e Beta 2: 20 – 30Hz;

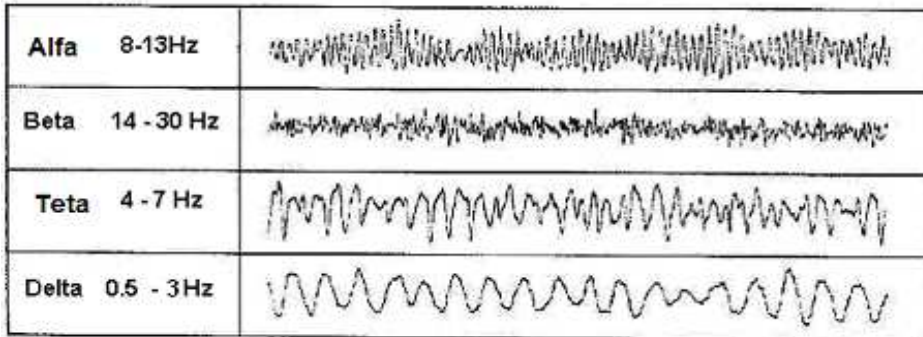


Figura 5.1: Representação dos sinais EEG associados a cada uma das bandas de frequência fundamentais: bandas alfa, beta, teta e delta.

5.3 Electrooculograma

Durante o registo do electroencefalograma existem outros sinais que podem ser obtidos simultaneamente para análise: o Electrocardiograma (ECG), o Electromiograma (EMG) e o EOG fazem parte desses sinais e permitem, por exemplo, caracterizar a vigília, o sono e seus estágios.

O EOG é registado através de eléctrodos colocados nos cantos externos de ambos os olhos, no plano horizontal. Cada um destes eléctrodos é referenciado a um eléctrodo colocado no lobo da orelha do lado correspondente.

O sinal EOG é um registo das diferenças de potencial provocadas pelos dipolos dos globos oculares durante os respectivos movimentos. Este sinal permite identificar com maior exactidão no sinal EEG as zonas associadas aos movimentos oculares. Na análise de sinais com crises epilépticas, a presença de movimentos oculares é habitual, nomeadamente antes da crise, o que provoca uma interferência no sinal, podendo atenuar a identificação de focos epilépticos. Nos registos dos sinais processados neste capítulo, o sinal EOG não é registado.

5.3.1 Motivação

Os movimentos oculares, a actividade muscular, bem como o ruído, são artefactos que se encontram presentes no sinal EEG e que dificultam a sua análise e interpretação. A figura 5.2 apresenta um segmento de dois canais EEG. Os movimentos oculares são visíveis nos canais frontais (Fp1-Cz) e nas derivações dos canais occipitais (O1-Cz).

A componente ocular representa a maior variação do sinal, como se pode ver pela figura 5.2. A eliminação dos movimentos oculares no sinal EEG pode ser efectuada com base no sinal

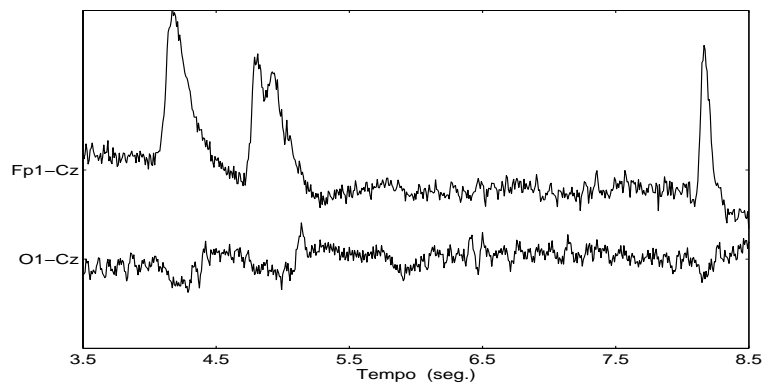


Figura 5.2: Segmento de dois canais de um sinal EEG, Fp1-Cz e O1-Cz.

EOG quando este é adquirido, por técnicas de rejeição e métodos de regressão, ou pode ser efectuada através de outros métodos que têm como objectivo corrigir o sinal EEG sem o conhecimento a priori do sinal EOG. O método de rejeição consiste na eliminação de segmentos de sinal EEG que tenham a presença de movimentos oculares. Esta técnica pode conduzir à perda de informação significativa. Por exemplo, no estudo de sinais epilépticos a rejeição de segmentos com movimentos oculares ou artefactos provocados pelos movimentos do doente constitui um problema para a interpretação do sinal, uma vez que a manifestação do foco epiléptico pode estar presente num dos segmentos rejeitados.

A rejeição pode ser obtida de forma automática, rejeitando segmentos acima de uma determinada amplitude. A detecção dos segmentos pode ser feita *on-line* ou *off-line*. A vantagem da detecção *on-line* consiste na possibilidade de prolongar o registo, conseguindo-se obter mais dados para análise. Em [32] são referidas estratégias para a minimização das perdas de sinal, associadas ao estudo de potenciais evocados.

Uma outra técnica tradicionalmente utilizada para a extracção dos movimentos oculares é a aproximação baseada em técnicas de regressão [33]. A análise de regressão é utilizada para estimar um sinal EOG associado ao sinal EEG. A correcção do sinal EEG é efectuada subtraindo a proporção de EOG estimada ao EEG original. No entanto, este tipo de correcções não remove apenas o sinal associado aos movimentos oculares, mas também actividade cerebral importante para análise e necessita do registo do sinal EOG.

Para além destas técnicas, existem outros estudos para a eliminação dos movimentos oculares no sinal EEG, independentes do sinal EOG, nomeadamente análise de componentes principais [34], *Independent Component Analysis* (ICA) [34], [35], [36], [37], [38] e técnicas de filtragem adaptativa [39].

Apesar de existirem algumas técnicas para a extracção dos artefactos no sinal EEG, não há nenhum método capaz de extrair os artefactos de forma automática. Por exemplo, no ICA a identificação das componentes associadas aos artefactos é efectuada por inspecção visual das mesmas.

Neste capítulo, vão ser apresentados resultados da eliminação de artefactos, aplicando os algoritmos discutidos nos capítulos 2.4 e 3.2, nomeadamente:

1. Eliminação dos movimentos oculares num canal frontal pelo algoritmo Local SSA;
2. Eliminação dos movimentos oculares num canal frontal pelo algoritmo KPCA, usando o método das distâncias na determinação da pré-imagem;
3. Eliminação de movimentos oculares e artefactos de um grupo de canais EEG num segmento de dez segundos, usando de forma automática o algoritmo Local SSA.

A componente ocular representa a maior variação do sinal, como se pode ver pela figura 5.2, ou seja, está associada às principais direcções do sinal. Assim sendo, a extracção dos movimentos oculares foi obtida, subtraindo ao sinal original a componente ocular extraída pelos algoritmos KPCA e Local SSA.

5.4 Método de Registo do sinal EEG

Os sinais recolhidos pertencem a uma base de dados de EEG's de rotina de doentes epilépticos, monitorizados digitalmente. Os sinais foram adquiridos com uma frequência de amostragem de 128 Hz e têm uma duração variável entre 1h a 4h. A gravação teve por base um conversor analógico/digital de 16 bits.

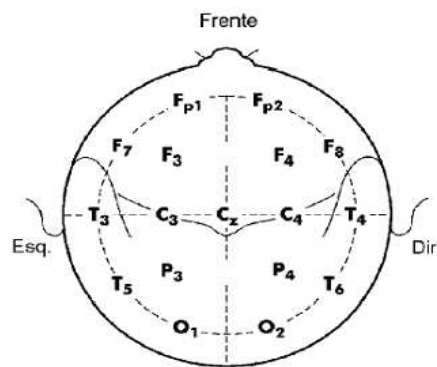


Figura 5.3: Mapa esquemático da posição dos eléctrodos na aquisição do sinal EEG segundo o sistema internacional 10-20.

Na aquisição do sinal foram usados dezanove eléctrodos colocados segundo o sistema internacional 10-20, como está representado na figura 5.3. O sinal EOG não foi registado. Para a visualização dos sinais EEG podem ser usadas diversas montagens. Neste caso, foi escolhida uma montagem monopolar com referência a um eléctrodo (Cz) localizado na zona central do cérebro.

5.5 Aplicação dos algoritmos Local SSA ao sinal EEG

Foi considerado um segmento com $N = 1536$ amostras (aproximadamente 12 segundos) do canal frontal ($Fp1 - Cz$) de um sinal EEG amostrado a 128 Hz .

Ao sinal unidimensional contaminado com movimentos oculares e ruído a 50Hz (figura 5.2) foi aplicado o algoritmo Local SSA, considerando três situações distintas:

- aplicação do algoritmo SSA e do critério MDL para selecção da dimensão;
- aplicação do algoritmo Local SSA e do critério MDL;
- aplicação do algoritmo Local SSA, com $k = 1$.

Nas secções seguintes são apresentados os parâmetros do algoritmo e os resultados da extracção dos movimentos oculares para as três situações descritas anteriormente. Posteriormente, segue-se uma secção que mostra a influência dos parâmetros do algoritmo nos resultados e uma outra, que apresenta a quantificação dos resultados obtidos.

Os gráficos apresentados com os sinais EEG foram obtidos, recorrendo à apresentação gráfica do software EEGLAB [40].

5.5.1 Parâmetros do algoritmo

O algoritmo Local SSA apresenta um conjunto de parâmetros que têm de ser previamente escolhidos pelo utilizador, nomeadamente a dimensão do *embedding*, M , o número de *clusters* q e o número de direcções principais a seleccionar em cada *cluster* k .

O número de direcções em cada *cluster* foi escolhido de forma automática por dois critérios:

- aplicação do critério MDL;
- selecção de uma única direcção em cada *cluster*, associada ao maior valor próprio;

O número de atrasos efectuados ao sinal EEG baseou-se na relação $M > \frac{f_s}{f_L}$, onde f_s representa a frequência de amostragem do sinal (128 Hz) e f_L a frequência associada ao sinal que queremos rejeitar, neste caso os movimentos oculares (3.5 Hz) [17]. Assim sendo, foram considerados $M = 41$ atrasos.

O número de atrasos efectuados ao sinal influenciam a escolha do número de *clusters*. Por questões práticas, o número de vectores em cada *cluster* deve ser superior ao *embedding* do sinal, sendo $q = 6$ *clusters* o valor adequado.

5.5.2 Extracção dos movimentos oculares

A componente ocular representa a maior variação do sinal, estando associada às principais direcções do mesmo. Assim sendo, a extracção dos movimentos oculares é obtida subtraindo

a componente ocular extraída pelo algoritmo Local SSA ao sinal original.

Nas figuras 5.4 e 5.5 estão representados os resultados obtidos aquando da aplicação do algoritmo Local SSA com seis *clusters* quando é aplicado o critério MDL para a selecção do número de direcções e quando é seleccionada uma única direcção em cada *cluster* respectivamente. O primeiro sinal representa o sinal EEG original, o segundo sinal representa o sinal EOG extraído e o último sinal, o sinal EEG corrigido.

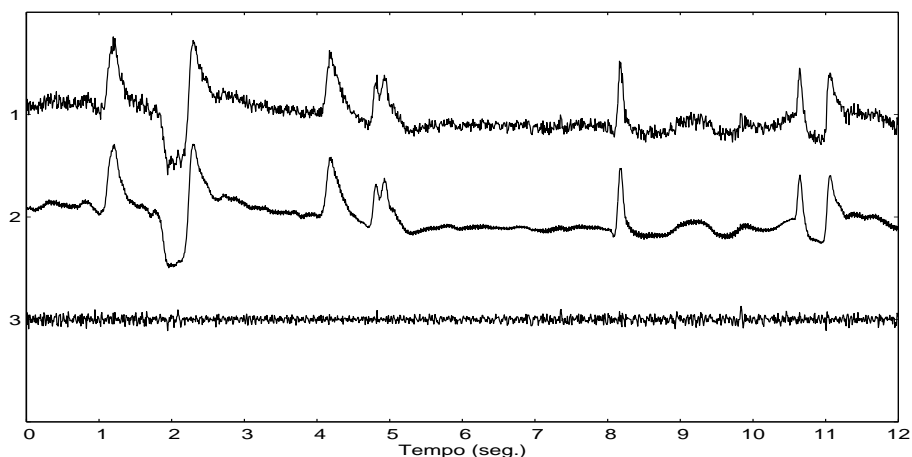


Figura 5.4: Canal frontal de um sinal EEG reconstruído pela aplicação do algoritmo Local SSA, considerando 6 *clusters* pelo critério MDL. No gráfico estão representados : canal frontal do sinal EEG original (1), sinal EOG extraído (2) e canal frontal do sinal EEG corrigido (3).

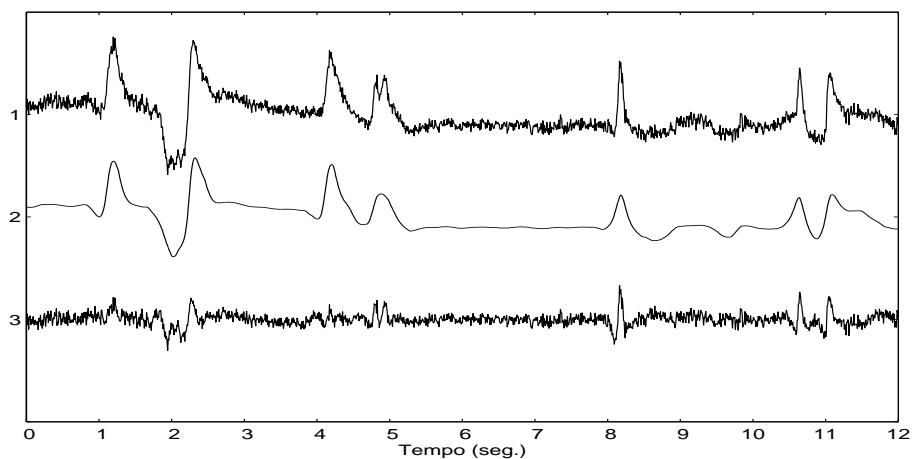


Figura 5.5: Canal frontal de um sinal EEG reconstruído pela aplicação do algoritmo Local SSA, considerando 6 *clusters* e seleccionando um valor próprio em cada *cluster*. No gráfico estão representados : canal frontal do sinal EEG original (1), sinal EOG extraído (2) e canal frontal do sinal EEG corrigido (3).

Analisando a figura 5.5, verifica-se que a selecção de um único valor próprio em cada *cluster* permite extrair a maioria das oscilações de grande amplitude (movimentos oculares) verificando-se, no entanto, que ainda estão presentes algumas oscilações no sinal corrigido.

Por sua vez, quando é usado o critério MDL, a componente associada aos movimentos oculares é eliminada por completo, como se pode ver pela figura 5.4. Além do algoritmo Local SSA, foi aplicado o algoritmo SSA ao mesmo segmento de dados, considerando o critério MDL. O MSE entre o sinal EOG extraído pelos algoritmos SSA e Local SSA, foi cerca de 6.67. Podemos ver pelo gráfico da função densidade espectral da figura 5.6 que a banda das baixas frequências ($< 10Hz$) aumenta com o número de *clusters* [41].

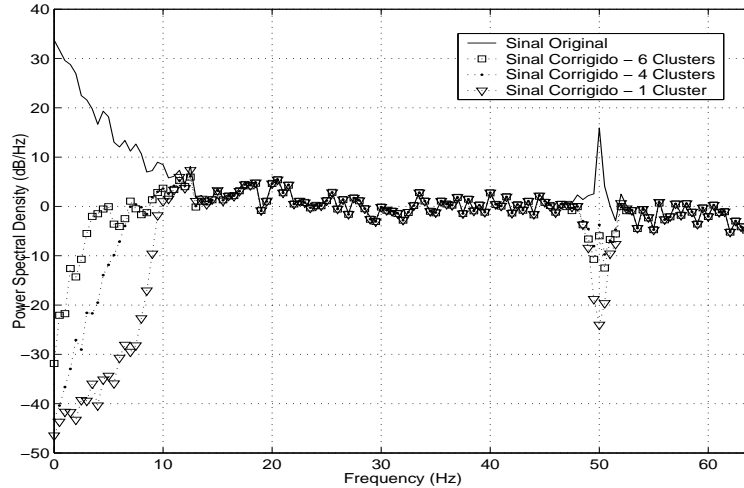


Figura 5.6: Função densidade espectral (dB/Hz) versus Frequência (Hz) do canal frontal do sinal EEG, do canal frontal corrigido pela aplicação do algoritmo SSA e do algoritmo Local SSA, considerando $q = 4$ e $q = 6$ *clusters*.

Existe uma diferença significativa nesta banda quando é aplicado o algoritmo Local SSA ou o algoritmo SSA. É de salientar que, para além da eliminação da banda das baixas frequências, existe a eliminação da frequência associada à corrente eléctrica (50 Hz) para qualquer um dos três casos apresentados: SSA, Local SSA com 4 *clusters* e Local SSA com seis *clusters*.

5.5.3 Influência do número de *clusters*

O número de *clusters* é um parâmetro que está limitado pela dimensão do *embedding* no conjunto de dados. Por razões práticas, o número de vectores em cada cluster deve ser superior à dimensão do *embedding*, neste caso $N_{c_i} > 41$ [42]. Na tabela 5.1 está apresentado o número de amostras e o número de direcções seleccionadas em cada *cluster* considerando o algoritmo SSA e o algoritmo Local SSA com $q = 4$ e $q = 6$ *clusters*.

Verifica-se que para $q = 6$ *clusters*, o número de vectores em cada cluster é sempre superior a 41. Se forem considerados mais do que seis *clusters*, o critério MDL tende a sobrestimar o subespaço de dimensão, uma vez que origina *clusters* com um número de vectores próximo de M .

Na figura 5.7 está representado o logaritmo dos valores próprios resultantes da aplicação do algoritmo Local SSA para $q = 6$ *clusters*. As linhas verticais correspondem à dimensão seleccionada pelo critério MDL. Podemos ver pelos gráficos que a estimação do MDL coincide

| | <i>Clusters</i> | Dimensão (k) | Size (N_{c_i}) |
|--------------------|-----------------|------------------|--------------------|
| sem Cluster | 1 | 9 | 1624 |
| 4 Clusters | 1 | 7 | 101 |
| | 2 | 7 | 956 |
| | 3 | 7 | 132 |
| | 4 | 8 | 435 |
| 6 Clusters | 1 | 6 | 128 |
| | 2 | 10 | 112 |
| | 3 | 5 | 754 |
| | 4 | 7 | 208 |
| | 5 | 8 | 324 |
| | 6 | 10 | 98 |

Tabela 5.1: Relação entre a dimensão do subespaço k e o número de amostras em cada *cluster*, da aplicação do algoritmo SSA e do algoritmo Local SSA ao canal frontal do sinal EEG.

com o agrupamento dos valores próprios relacionados com o sinal residual.

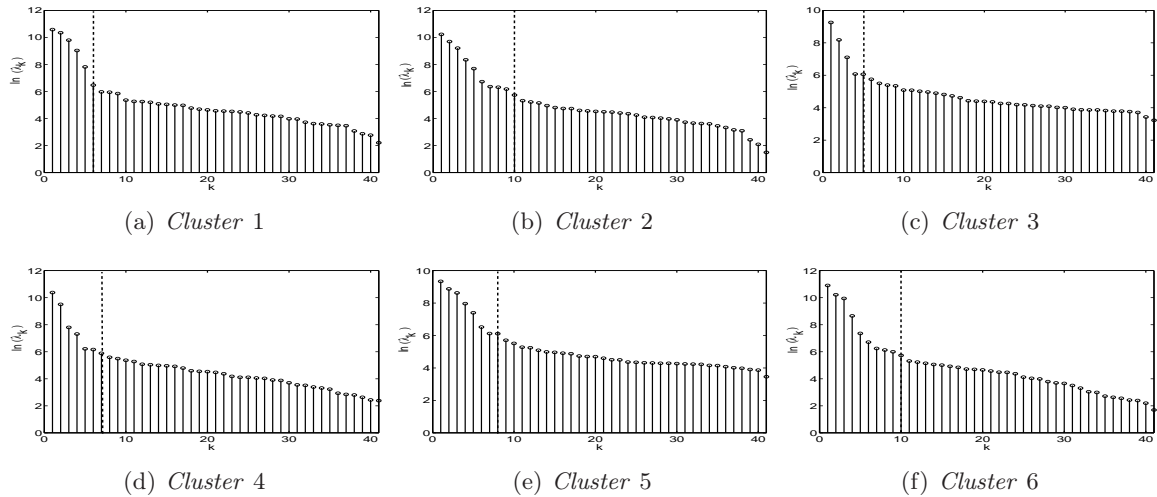


Figura 5.7: Representação do logaritmo dos valores próprios $\ln(\lambda_k)$ versus o número de valores próprios k de cada *cluster* quando é aplicado o algoritmo Local SSA ao canal frontal do sinal EEG. (- -) A tracejado está representada a dimensão seleccionada pelo critério MDL.

A distribuição dos vectores do *embedding* do sinal multidimensional em cada *cluster* é efectuada de forma não homogénea. A figura 5.8 representa a distribuição das amostras quando são considerados seis *clusters*. Podemos ver que o *cluster* número três é o de maior dimensão e está relacionado com as zonas onde há ausência de movimentos oculares.

Tendo em conta a aleatoriedade da selecção dos vectores-coluna da matriz de trajectória, foi efectuada uma experiência com o objectivo de encontrar o número de *clusters* óptimo para este segmento EEG. Foi aplicado o algoritmo Local SSA, fazendo variar o número de *clusters* entre $q = 1$ e $q = 7$ e foram efectuadas 1000 simulações. Para cada simulação, foi calculado o *cluster* que minimizou o valor kurtosis do sinal corrigido. A figura 5.9 representa a função

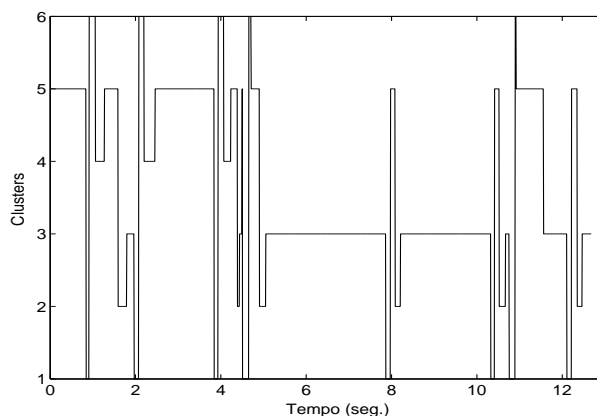


Figura 5.8: Distribuição dos vectores coluna da matriz de trajectória do segmento de dados, considerando $q = 6$ clusters.

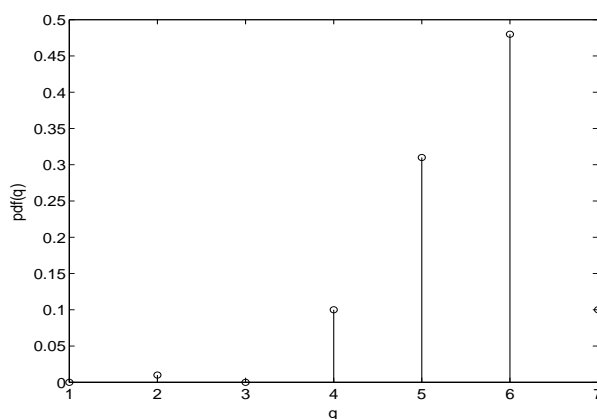


Figura 5.9: Função densidade de probabilidade do número de clusters ($pdf(q)$) versus o número de clusters (q) que minimiza o valor da kurtosis do sinal EEG corrigido pelo algoritmo Local SSA. O número de clusters variou entre $q = 1$ e $q = 7$ e para cada cluster foram efectuadas 1000 simulações.

densidade de probabilidade do número de clusters que minimiza o valor da kurtosis do sinal corrigido versus o número de clusters.

Analisando a figura 5.9 concluí-se que o número de clusters óptimo para este segmento é $q = 6$ com uma percentagem de 48%.

5.5.4 Quantificação dos resultados

Para avaliar os resultados obtidos da aplicação dos algoritmos SSA e Local SSA com $q = 4$ e $q = 6$ clusters, foi efectuado um estudo da energia instantânea do sinal no domínio da frequência, considerando três bandas de frequência fundamentais: teta (3.5-7.5Hz), alfa (7.5-13Hz) e beta (13-25Hz) [42].

Para cada uma das bandas indicadas, foi calculada a energia instantânea do sinal, utilizando uma janela de Hanning de 2s, que corresponde a uma resolução de $0.5Hz$, e com 50% de

sobreposição, figura 5.10.

Observando a figura 5.10, verifica-se que na banda teta a energia do sinal extraído tem valores inferiores aos do sinal original, havendo, assim, perda de informação. Nas restantes bandas, a

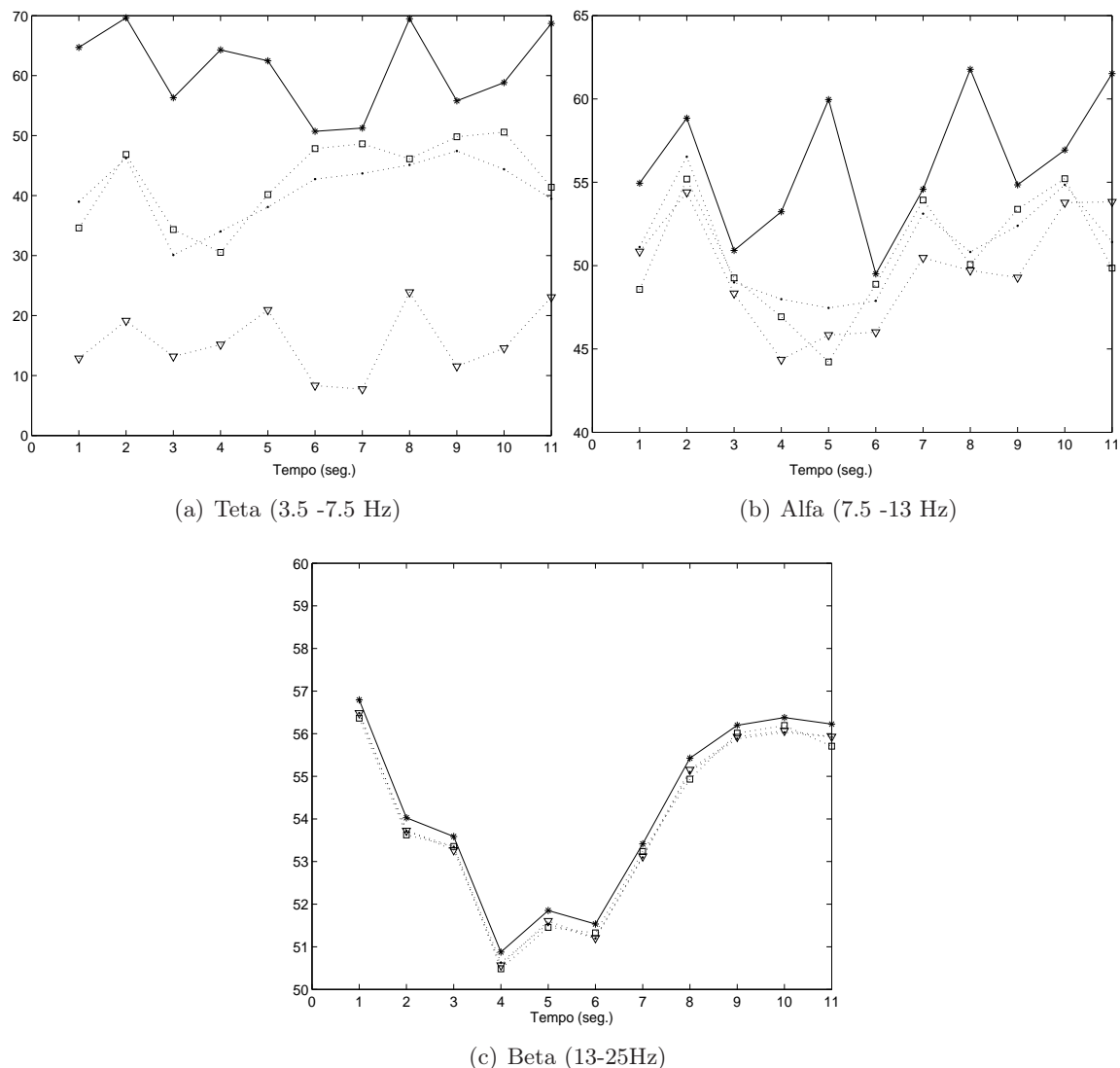


Figura 5.10: Representação da energia instantânea do sinal ao longo do tempo para cada uma das bandas de frequência fundamental. Estão representados 4 sinais : * - Canal frontal do sinal EEG Original ; □ - Canal frontal do sinal EEG corrigido pelo algoritmo Local SSA com $q = 6$ Cluster; ● - Canal frontal do sinal EEG corrigido pelo algoritmo Local SSA com $q = 4$ Cluster; ▽ - Canal frontal do sinal EEG corrigido pelo algoritmo SSA.

perda de informação relevante tende a ser menor, à medida que a frequência aumenta. Neste caso de estudo, verifica-se que na banda beta não existe perda de informação, apesar de esta ser dominante nos canais frontais.

Entre os 5.5 – 7.5 segundos, zona onde não estão presentes movimentos oculares, existe uma menor perda de informação entre o sinal EEG e o sinal extraído, nomeadamente nas bandas teta e alfa.

Comparando os resultados obtidos pela aplicação dos algoritmos SSA e Local SSA, verificou-se que para $q = 6$ *clusters* os resultados são mais satisfatórios, dado que a perda de informação nas bandas teta e alfa é menor, comparativamente com os outros dois casos.

5.6 Aplicação do algoritmo Kernel PCA ao sinal EEG

O algoritmo KPCA é utilizado habitualmente em imagem para eliminação de ruído, no reconhecimento de padrões [43, 24] e no reconhecimento de formas [1].

A aplicação do algoritmo a séries temporais ou a sinais biológicos é pouco usual. Em [44] é apresentado um trabalho da aplicação do algoritmo KPCA para a extracção de características em sinais EEG.

Nesta secção vão ser apresentados alguns resultados da aplicação do algoritmo KPCA ao sinal EEG.

5.6.1 Parâmetros do algoritmo

Foi utilizado o mesmo segmento de sinal EEG da secção anterior (canal frontal Fp1-Cz). Devido à complexidade do algoritmo KPCA, o sinal EEG foi dividido em quatro sub-sinais de três segundos ($N = 384$). Os parâmetros utilizados no algoritmo foram obtidos de forma experimental. Foram efectuados $M = 41$ e $M = 21$ atrasos ao sinal. A correlação do sinal EEG corrigido com $M = 41$ e $M = 21$ foi 0.99. A selecção do número de direcções da matriz

| Segmentos | Dimensão (k) | Parâmetro RBF |
|-------------------|------------------|--------------------------|
| Segmento 1 | 8 | $0.05 \times \sigma_s^2$ |
| Segmento 2 | 10 | $0.1 \times \sigma_s^2$ |
| Segmento 3 | 5 | $0.5 \times \sigma_s^2$ |
| Segmento 4 | 6 | $0.3 \times \sigma_s^2$ |

Tabela 5.2: Parâmetros do algoritmo KPCA usados em cada um dos segmentos do sinal EEG, considerando $M = 21$ atrasos: número de direcções seleccionadas na matriz de kernel, k , e o valor do parâmetro do kernel RBF.

| Segmentos | Dimensão (k) | Parâmetro RBF |
|-------------------|------------------|-------------------------|
| Segmento 1 | 8 | $0.5 \times \sigma_s^2$ |
| Segmento 2 | 8 | $0.5 \times \sigma_s^2$ |
| Segmento 3 | 8 | $0.5 \times \sigma_s^2$ |
| Segmento 4 | 8 | $0.5 \times \sigma_s^2$ |

Tabela 5.3: Parâmetros do algoritmo KPCA usados em cada um dos segmentos do sinal EEG, considerando $M = 41$ atrasos: número de direcções seleccionadas na matriz de kernel, k , e o valor do parâmetro do kernel RBF.

de kernel e do parâmetro RBF foi efectuada de forma experimental. Como está descrito na secção 3.9, o parâmetro RBF influencia o decaimento dos valores próprios. Nas experiências

efectuadas, o parâmetro RBF foi seleccionado de forma a obter um decaimento mais lento dos valores próprios. Nas tabelas 5.2 e 5.3 estão apresentados os parâmetros do algoritmo KPCA usados em cada um dos segmentos do sinal EEG, quando se consideram $M = 21$ e $M = 41$ atrasos: número de direcções (k) seleccionadas na matriz de kernel e o valor do parâmetro do kernel RBF, em função da variância dos segmentos do sinal de entrada, σ_s^2 . O número de vizinhos considerado em ambos os casos para o cálculo da pré-imagem pelo método das distâncias foi $n = 10$.

Se a dimensão do *embedding* for $M = 41$, a escolha dos parâmetros para os segmentos é efectuada de forma homogénea. Se a dimensão do *embedding* for inferior ($M = 21$) a selecção dos parâmetros é diferente para cada um dos segmentos, tabela 5.2.

5.6.2 Extracção dos movimentos oculares

Na figura 5.11 estão apresentados os resultados da aplicação do algoritmo KPCA aos quatro segmentos do sinal EEG quando são considerados $M = 21$ atrasos (tabela 5.2). Cada gráfico contém o sinal original, o sinal extraído e o sinal corrigido.

Verifica-se que foram eliminadas as oscilações de grande amplitude, não havendo presença de movimentos oculares nos resultados apresentados.

Para uma melhor interpretação dos resultados, foi efectuado o espectrograma do sinal original, do sinal extraído e do sinal corrigido para este resultado e para o resultado obtido na aplicação do algoritmo Local SSA com seis *clusters* representado na figura 5.4.

Os espectrogramas da figura 5.13 representam a evolução da frequência dos termos do sinal (eixo dos yy) em função do tempo (eixo dos xx). Por análise do espectrograma do sinal original, conclui-se que existem duas zonas de maior energia: zona das baixas frequências associada aos movimentos oculares ($< 10\text{Hz}$) e a zona das altas frequências associada à corrente eléctrica (50Hz). O sinal EOG extraído está fundamentalmente associado à banda das baixas frequências ($< 10\text{Hz}$), figuras 5.12 e 5.13.

Comparando os resultados obtidos pelos dois algoritmos, verifica-se que o algoritmo Local SSA para além de eliminar a zona das baixas frequências, permite eliminar a contaminação da corrente eléctrica a 50Hz . Quanto ao algoritmo KPCA, este elimina a zona associada às baixas frequências, mas de uma forma menos acentuada do que o algoritmo SSA.

Foi calculada a frequência instantânea ao longo do tempo para a banda delta ($0.5 - 3.5\text{Hz}$), para o sinal EEG original, para o sinal EEG corrigido pelo algoritmo KPCA e para o sinal EEG corrigido pelo algoritmo Local SSA, considerando $q = 6$ *clusters* (figura 5.14).

Verifica-se que entre os 3s e os 5s há uma diferença significativa dos resultados nesta banda, no entanto o sinal EOG extraído em ambos os métodos tem uma correlação de 0.994 e a correlação do sinal corrigido pelos dois métodos é de 0.81.

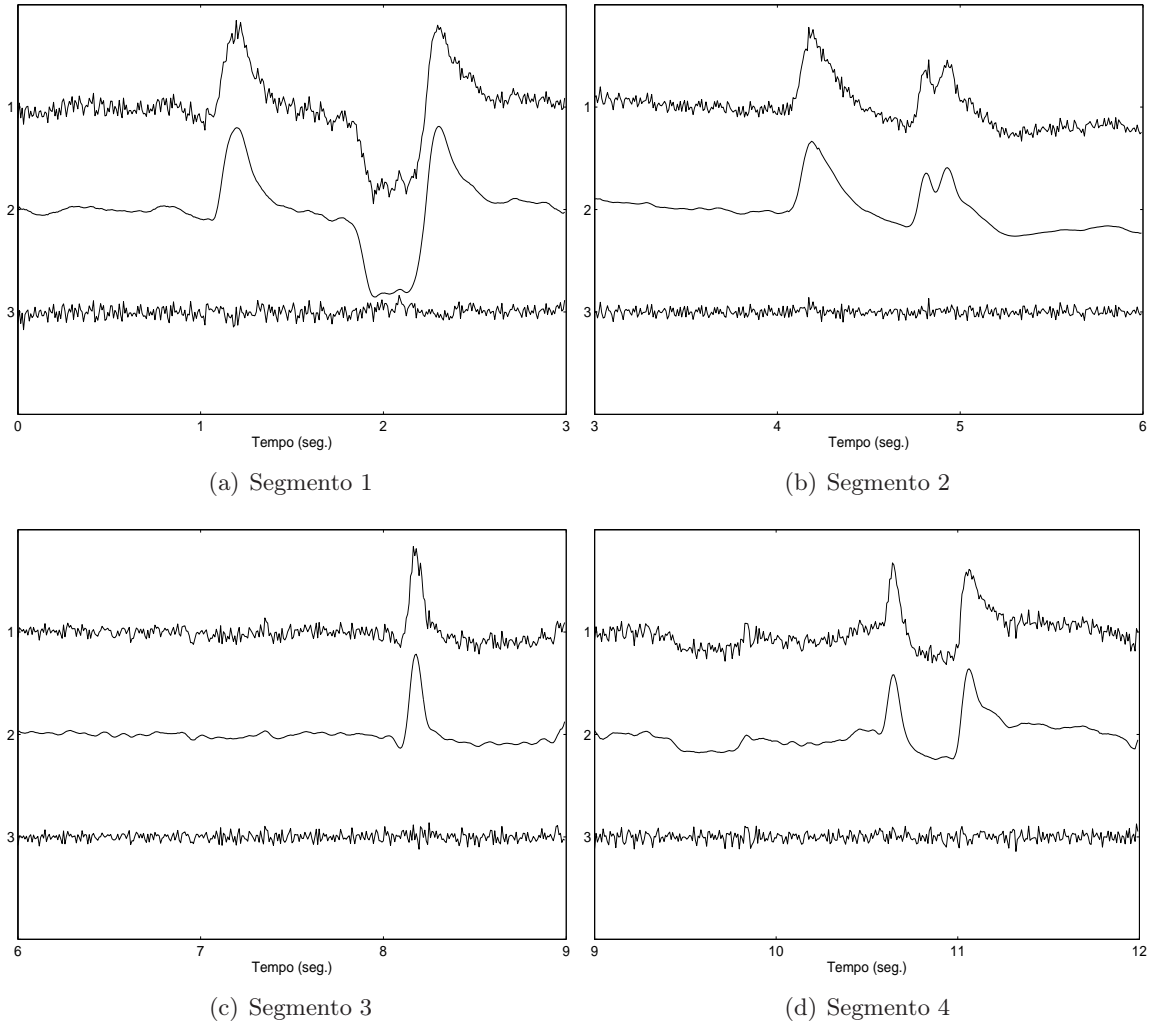


Figura 5.11: Aplicação do algoritmo KPCA para extracção dos movimentos oculares. Segmentação do sinal EEG em 4 segmentos de 3 segundos. Representação do canal frontal do sinal EEG original (1), sinal EOG extraído (2) e canal frontal do sinal EEG corrigido (3).

5.7 Aplicação do algoritmo Local SSA a outros artefactos

A eliminação automática de artefactos num sinal EEG é um problema que tem vindo a ser abordado por diversos autores, baseado na ICA [38, 33].

Nesta secção vai ser apresentado um resultado da aplicação automática do algoritmo Local SSA a um segmento EEG de dez segundos. Este segmento de sinal pertence ao mesmo indivíduo das experiências anteriores, porém neste sinal existem crises de epilepsia. Os 10 segundos de sinal representados na figura 5.15 antecedem uma das crises de epilepsia registadas. O sinal foi amostrado a 128Hz e visualizado através de uma montagem monopolar com referência a um eléctrodo comum (Cz). Foram considerados $M = 41$ atrasos e o número de *clusters* foi obtido de forma automática, considerando dois critérios de selecção: o número de vectores em cada *cluster* tem de ser superior ao *embedding* do sinal ($N_c > 41$) e a dimensão seleccionada pelo critério MDL menor do que metade do *embedding* ($k < \frac{M}{2}$, tabela 5.4). O número de

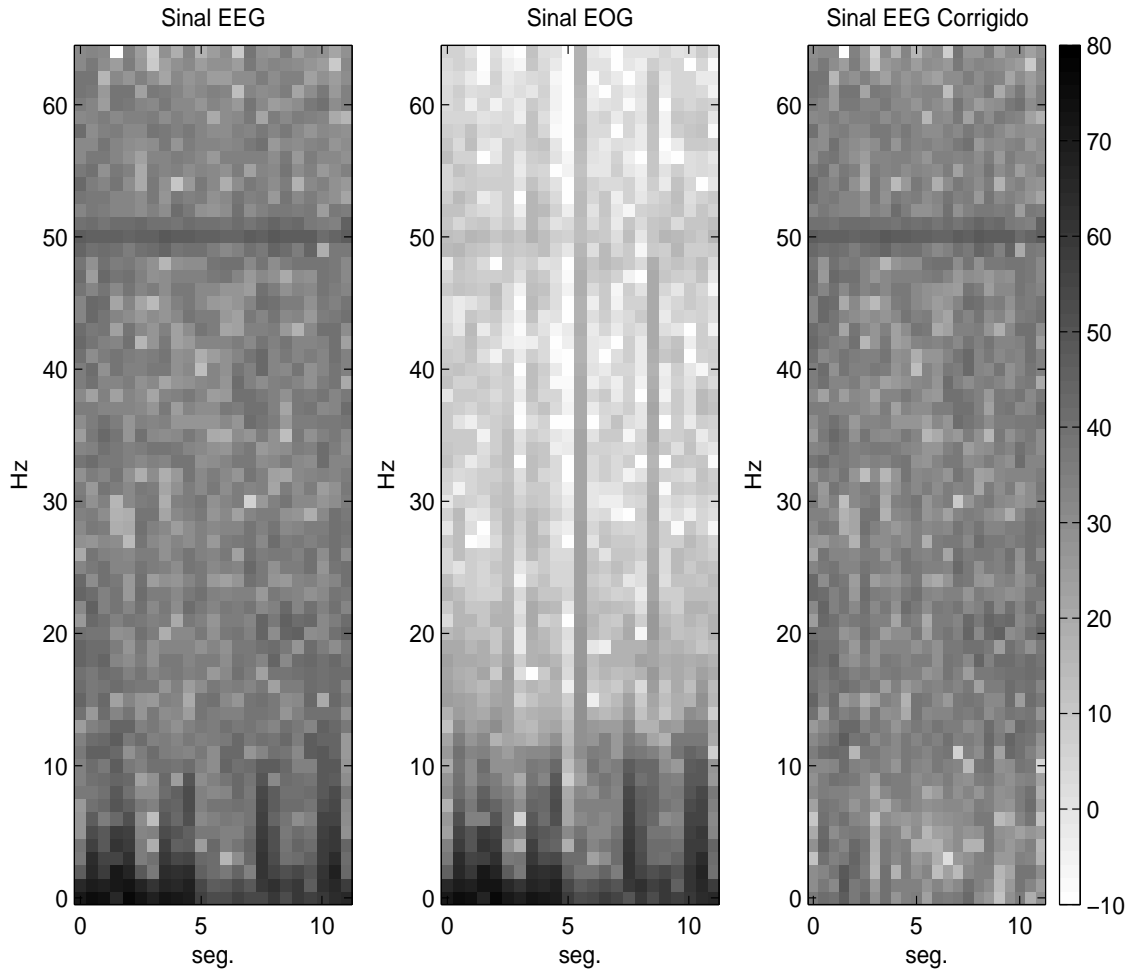


Figura 5.12: Espectrograma do sinal EEG original, do sinal EOG extraído pelo algoritmo KPCA e do sinal EEG corrigido.

clusters variou entre $q = 1$ e $q = 10$. Como podemos observar pela figura 5.15, nos canais Fp2, F4, Fp1, F3 estão presentes movimentos oculares e no canal F7 está presente um artefacto provocado pelo movimento do paciente seguido de um movimento ocular. Os canais T4, O1 e T5 também foram processados para a eliminação de possíveis oscilações associadas ao artefacto e aos movimentos oculares existentes. Cada canal foi processado de forma individual. Na figura 5.16 estão representados os sinais extraídos pela aplicação do algoritmo Local SSA. Verifica-se que os movimentos oculares foram removidos dos canais Fp2, F4, Fp1 e F3, bem como o artefacto e o movimento ocular, presentes no canal F7.

Nos restantes canais, T4 O1 e T5, foi eliminada a oscilação existente provocada pelos movimentos oculares.

Subtraindo os sinais extraídos aos sinais originais, obtemos o sinal EEG corrigido (figura 5.17). No sinal corrigido, as oscilações de elevada amplitude foram removidas, assim como o artefacto existente. A leitura e análise do sinal corrigido torna-se mais clara, sendo possível observar no canal F3 a existência de pontas que estavam atenuadas pela presença dos movimentos oculares.

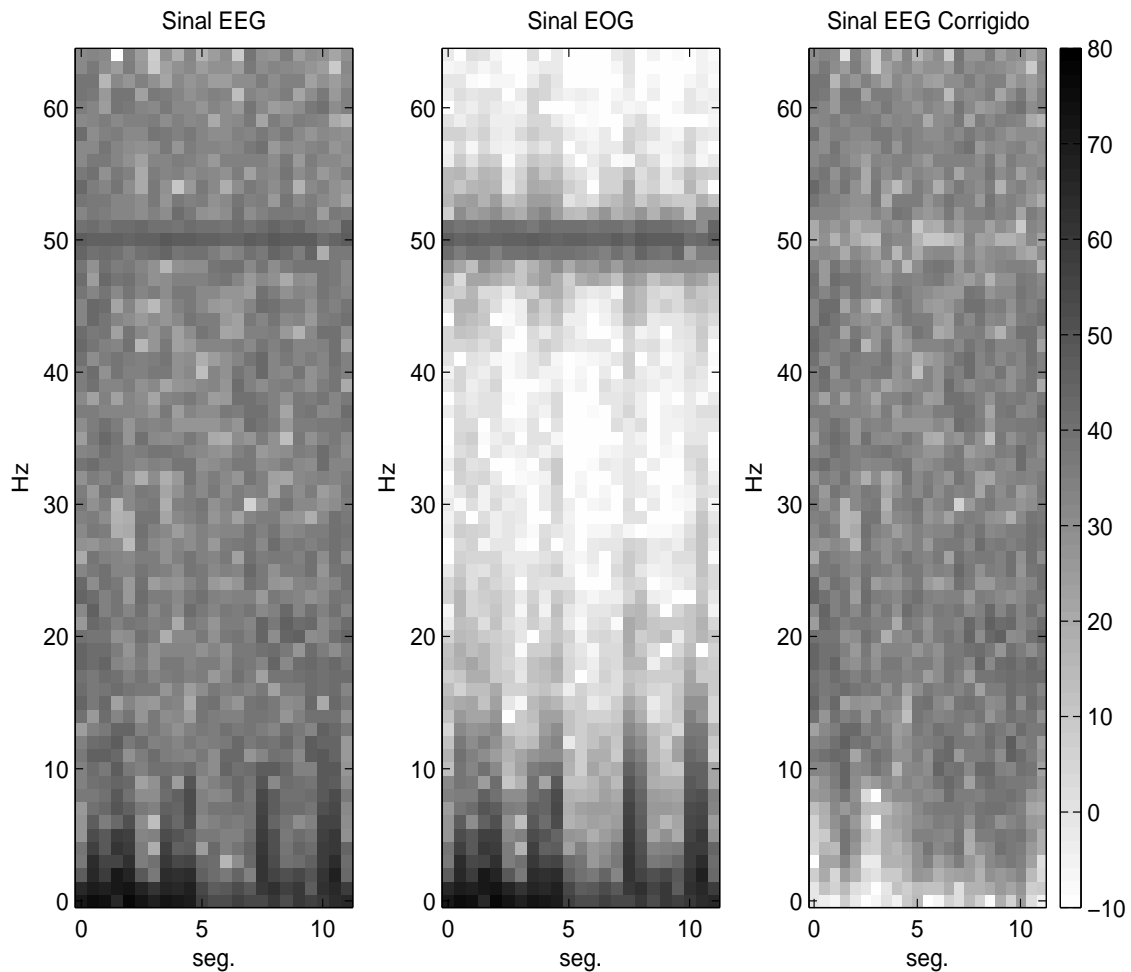


Figura 5.13: Espectrograma do canal frontal do sinal EEG original, do sinal EOG extraído pelo algoritmo SSA, considerando 6 *clusters* e do canal frontal do sinal EEG corrigido.

5.8 Conclusões

Neste capítulo foi efectuado um estudo preliminar da aplicação dos algoritmos KPCA e Local SSA ao sinal EEG para a extracção de artefactos, nomeadamente movimentos oculares, corrente eléctrica e movimentos originados pelo paciente ou pelos eléctrodos .

Dos resultados obtidos, verificou-se que ambos os algoritmos retiraram o sinal EOG de forma muito similar. Os sinais residuais (o sinal EEG reconstruído) aparecem menos distorcidos, na banda das baixas frequências, com a aplicação do algoritmo KPCA.

Comparando o modo de execução dos dois algoritmos, verifica-se que o KPCA apresenta uma maior complexidade de processamento do que o algoritmo Local SSA.

Conclui-se que o algoritmo Local SSA é mais fácil de configurar para extrair artefactos do sinal EEG, de forma automática.

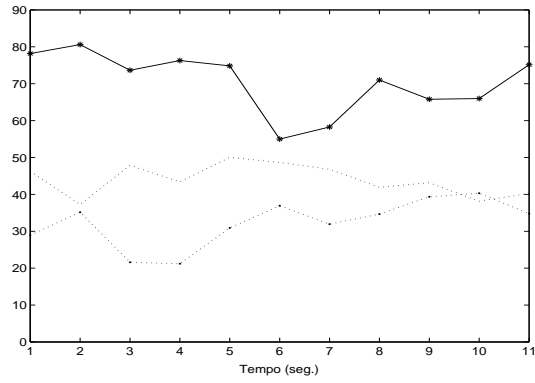


Figura 5.14: Representação da energia instantânea ao longo do tempo para a banda de frequência delta: ★ - Canal frontal do sinal EEG Original ; ● - Canal frontal do sinal EEG corrigido pelo algoritmo Local SSA com 6 *cluster*; ⋯- Canal frontal do sinal EEG corrigido pelo algoritmo KPCA.

| Canais | # Clusters |
|--------|------------|
| Fp2 | 6 |
| F4 | 5 |
| T4 | 5 |
| Fp1 | 4 |
| F3 | 5 |
| O1 | 3 |
| F7 | 4 |
| T5 | 5 |

Tabela 5.4: Número de *clusters* utilizado pelo algoritmo Local SSA para cada canal processado do sinal da figura 5.15.

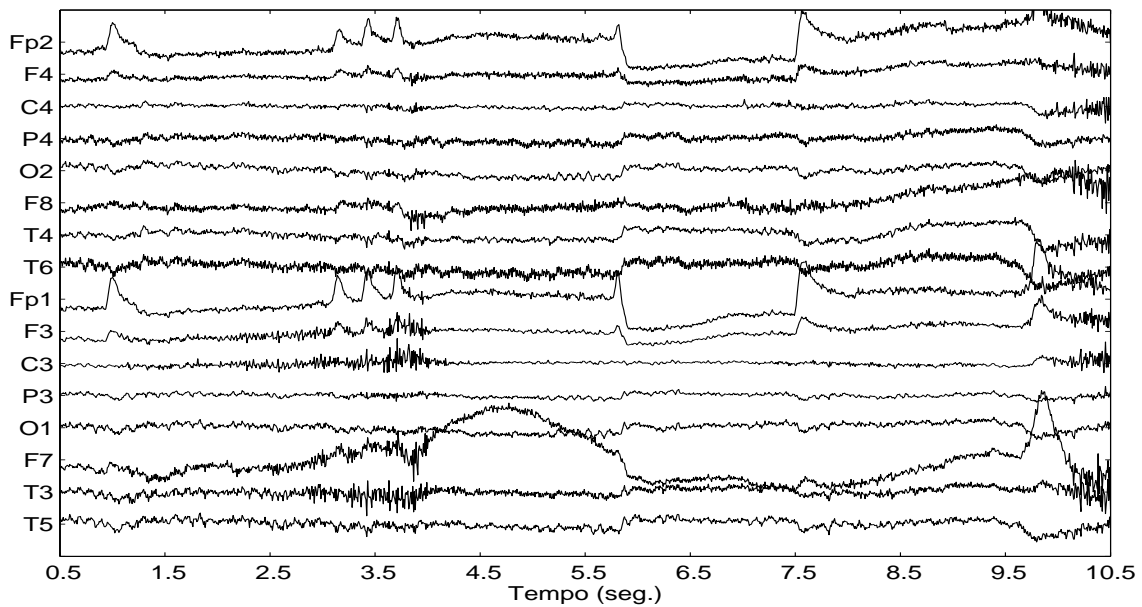


Figura 5.15: Segmento de 10 segundos de um sinal EEG que antecede uma crise de epilepsia.

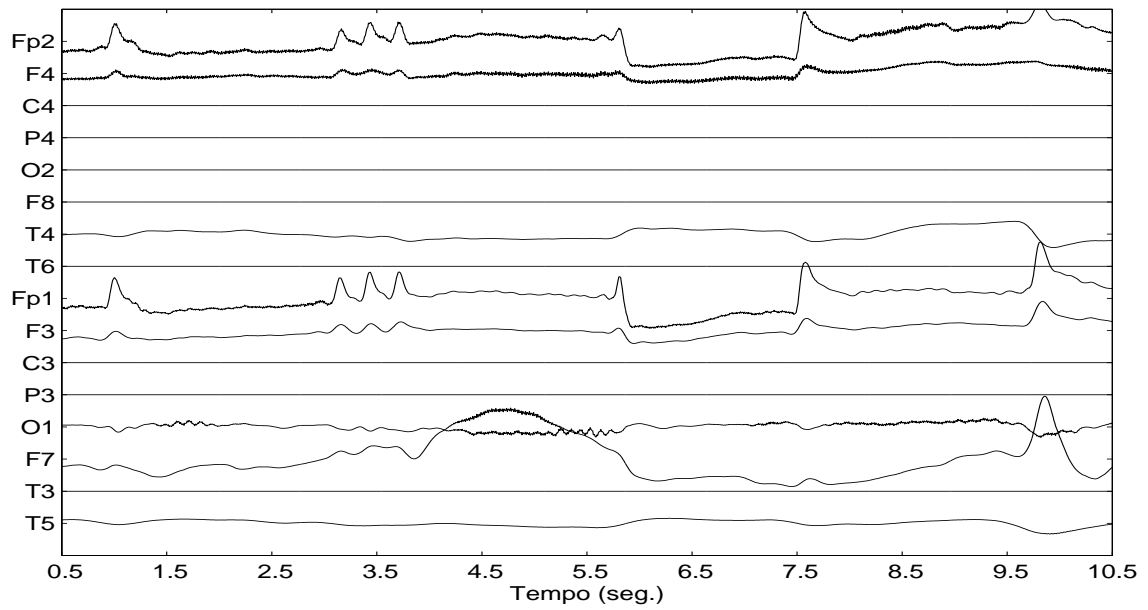


Figura 5.16: Sinais extraídos de forma automática pela aplicação do algoritmo Local SSA aos canais Fp2, F4, T4, Fp1, F3, O1, F7 e T5.

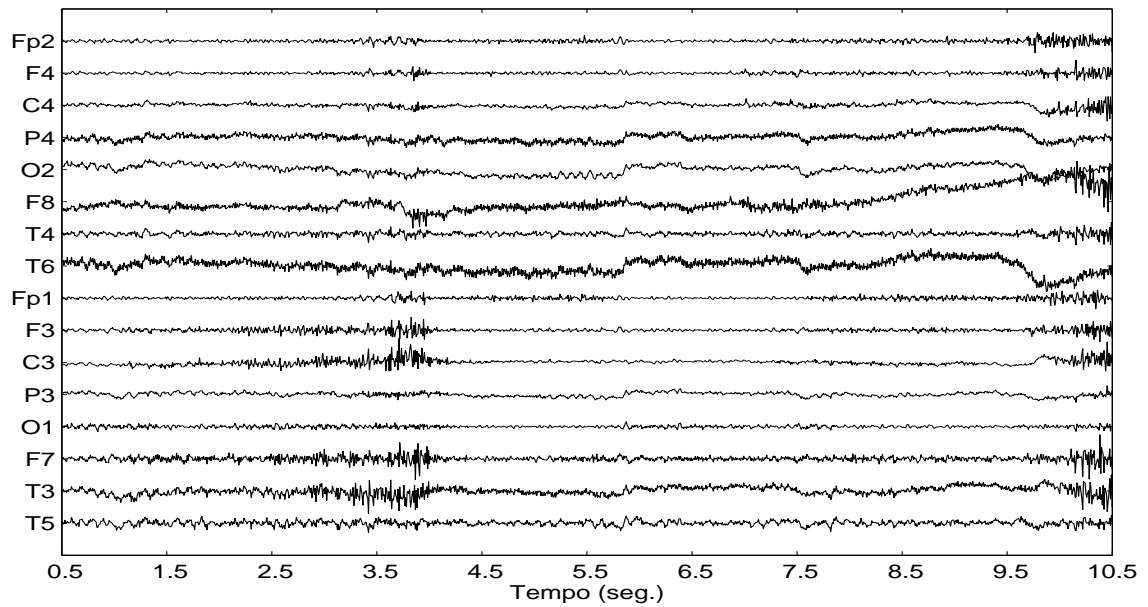


Figura 5.17: Segmento de sinal EEG corrigido pela subtração das componentes da figura 5.16 ao sinal EEG original, figura 5.15.

Capítulo 6

Conclusões e Trabalho Futuro

Neste trabalho foram estudadas técnicas não-lineares para a redução de ruído em séries temporais. As técnicas consistem na transformação dos dados para um espaço de dimensão superior, transformação essa que é obtida através de matrizes de trajectória definidas pelos algoritmos SSA e Local SSA ou através de funções de kernel obtidas pelo algoritmo KPCA.

Os algoritmos foram descritos, considerando na maior parte das operações uma notação matricial, simplificando a sua interpretação. Para cada um deles foi apresentada uma descrição em pseudo-código. Alguns dos algoritmos foram implementados em ambiente Matlab, enquanto para outros foram usadas as implementações disponibilizadas pela *toolbox* [16]. Para estudar a influência de alguns dos seus parâmetros, os algoritmos foram validados com sinais artificiais.

Pelos resultados obtidos, conclui-se que os algoritmos KPCA e Local SSA apresentam resultados muito semelhantes na eliminação de ruído. Em qualquer um dos algoritmos, é preciso escolher um conjunto de parâmetros de forma experimental.

No caso do algoritmo Local SSA, a selecção do número de direcções no espaço de dimensão superior foi obtida comparando vários critérios, tais como o critério MDL, o critério AIC e o critério VE. Dos resultados obtidos com sinais artificiais, concluiu-se que o critério MDL é aquele que estima com maior consistência a dimensão do subespaço. Para a selecção da dimensão do *embedding*, foi considerada a relação entre a frequência de amostragem do sinal original e a frequência do sinal que pretendemos eliminar [17]. No caso da selecção do número de *clusters* do algoritmo Local SSA, esta é possível através de um conjunto de regras obtidas por experimentação. O número de *clusters* foi obtido, tendo em conta que o número de elementos em cada *cluster* tem de ser superior à dimensão do *embedding* (M) e que o número de direcções seleccionadas pelo critério MDL tem de ser inferior a esse mesmo valor.

No caso do algoritmo KPCA, embora os estudos preliminares indiquem que deve haver um decaimento lento dos valores próprios da matriz de kernel, controlado pela escolha do parâmetro RBF, não se estabeleceu ainda um bom critério de escolha deste parâmetro, nem do número de direcções a seleccionar na reconstrução.

Os algoritmos Local SSA e KPCA foram também aplicados ao sinal EEG com o objectivo de eliminar artefactos. Verificou-se que os algoritmos Local SSA e KPCA apresentam resulta-

dos muito semelhantes quando aplicados a um canal frontal na eliminação do sinal EOG. O algoritmo Local SSA é o mais simples de utilizar na eliminação de artefactos, uma vez que a selecção dos parâmetros pode ser obtida de forma automática, eliminando para além do sinal EOG, outro tipo de artefactos, tais como a corrente eléctrica e os movimentos provocados pelos eléctrodos ou pelo paciente.

Verificou-se que a aplicação do algoritmo Local SSA, escolhendo apenas uma direcção ($k = 1$) em cada *cluster*, é uma versão simplificada deste, que pode ser utilizada quando se pretende ter uma versão aproximada do sinal a extrair. Este foi o caso da eliminação da componente da água num estudo de *Nuclear Magnetic Resonance* (NMR), em que a versão aproximada era a entrada do algoritmo de *simulated annealing* [45]. Este estudo, [46], teve por objectivo identificar quantas saídas do algoritmo *dAmuse* [3] contribuíram para a reconstrução de um sinal idêntico ao extraído pelo Local SSA.

Quanto à complexidade dos algoritmos, verificou-se que o algoritmo KPCA apresenta uma grande complexidade de processamento que está relacionada com a dimensão da matriz de kernel, (que por sua vez depende do número de amostras a processar) e com o algoritmo de reconstrução dos dados no espaço de entrada. Neste estudo, contornou-se o problema, segmentando o sinal e aplicando o método das distâncias. No entanto, há formas de calcular as direcções principais no espaço característico, seleccionando os dados relevantes no conjunto de treino - algoritmo *Greedy KPCA* [47].

Pelos resultados apresentados, podemos concluir que os algoritmos KPCA e Local SSA permitem eliminar ruído em séries temporais de um modo eficaz, tendo em atenção a selecção dos parâmetros dos algoritmos.

Trabalho Futuro

Os resultados obtidos neste trabalho deixam em aberto vários assuntos para estudo futuro, tais como:

- Selecção de parâmetros do algoritmo KPCA: parâmetro do kernel RBF, determinação do número de direcções a seleccionar na matriz de kernel e escolha do número de vizinhos a considerar no caso do método das distâncias.
- Redução da complexidade de processamento do algoritmo KPCA através da implementação e estudo do algoritmo *Greedy KPCA*, [47].
- Estudo comparativo dos algoritmos KPCA e Local SSA com outros métodos, para eliminação automática de artefactos no sinal EEG [38, 34], considerando diferentes casos clínicos [48, 49, 50].

- Aplicação dos algoritmos a um maior conjunto de sinais EEG, para validação dos resultados de uma forma mais consistente. Análise estatística dos resultados obtidos e quantificação dos mesmos, com base na análise espectral.
- Implementação do algoritmo Local SSA no software EEGLAB [40], para uma melhor visualização e interpretação dos resultados, por parte dos especialistas clínicos.

Bibliografia

- [1] S. Romdhani, S. Gong, and A. Psarrou, “A Multi-View Nonlinear Active Shape Model Using Kernel PCA,” in *In Proceedings of BMVC*, (Nottingham, UK), pp. 483–492, 1999.
- [2] A. Chichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. Wiley, 2002.
- [3] A. M. Tomé, A. R. Teixeira, E. W. Lang, K. Stadlthanner, A. P. Rocha, and R. Almeida, “dAmuse - A New Tool for Denoising and Blind Source Separation,” *Digital Signal Processing, Elsevier*, vol. 15, pp. 400–421, July 2005.
- [4] N. Golyandina, V. Nekrutkin, and A. Zhigljavsky, *Analysis of Time Series Structure: SSA and Related Techniques*. 2001.
- [5] J.T.Kwork and I.W.Tsang, “The Pre-Image Problem in Kernel Methods,” in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, (Washington DC), pp. 408–415, 2003.
- [6] P. Stoica and Y. Selén, “Model-Order Selection-A Review of Information Criterion Rules,” *IEEE Signal Processing Magazine*, vol. 21, pp. 36–46, July 2004.
- [7] Z. Leonowicz, J. Karvanen, T. Tanaka, and J. Rezmer, “Model Order Selection Criteria: Comparative Study and Applications,” in *Computational Problems of Electric Engineering (CPEE2004)*, (Zakopane, Poland), pp. 193–196, 2004.
- [8] M. Hansen and B. Yu, “Minimum Description Length Model Selection Criteria for Generalized Linear Models,” *Journal of the American Statistical Association*, vol. 96, pp. 746–774, 2001.
- [9] P. Gruber, F. J. Theis, K. Stadlthanner, E. W. Lang, A. M. Tomé, and A. R. Teixeira, “Denoising Using Local ICA and Kernel-PCA,” in *International Joint Conference on Neuronal Networks, IJCNNt2004* (IEEE, ed.), (Budapest), pp. 1959–1964, 2004.
- [10] J. Gower, “Adding a Point to Vector Diagrams in Multivariate Analysis,” *Biometrika*, vol. 55, no. 3, pp. 582–585, 1968.
- [11] M.R.Allen and A.W.Robertson, “Distinguishing Modulated Oscillations from Colored Noise in Multivariate Datasets,” *Climate Dynamics*, vol. 12, pp. 775–784, 1996.

- [12] M. Ghil, M. R. Allen, M.D.Dettinger, K. Ide, and e. al., “Advanced Spectral Methods for Climatic Time series,” *Reviews of geophysics*, vol. 40, no. 1, 2002.
- [13] D. H. Schoellhamer, “Singular Spectrum Analysis for Time Series with Missing Data,” *Geophysical Research Letters*, vol. 28, no. 19, pp. 3187–3190, 2001.
- [14] N. Kanbhatla and T. K. Leen, “Dimension Reduction by Local Principal Component Analysis,” *Neural Computation*, vol. 9, no. 7, pp. 1493–1517, 1997.
- [15] A. D.G., *Practical Statistics for Medical Research*. Chapman & Hall, 1999.
- [16] V. Franc and V. Hlaváč, “Statistical Pattern Recognition Toolbox,” <http://cmp.felk.cvut.cz/~xfrancv/stprtool/>, 2005.
- [17] C. J. James and D. Lowe, “Extracting Multisource Brain Activity from a Single Electromagnetic Channel,” *Artificial Intelligence in Medicine*, vol. 28, pp. 89–104, 2003.
- [18] P. Gruber, K. Stadlthanner, A. M. Tomé, A. R. Teixeira, F.J.Theis, C. G. Puntonet, and E. W. Lang, “Denoising Using Local ICA and a Generalized Eigendecomposition with Time-Delayed Signals,” in *ICAt2004*, vol. LNCS 3195, (Granada), pp. 993–1000, 2004.
- [19] P. Gruber, F. J. Theis, and E. W. Lang, “Automatic Denoising Using Local Independent Component Analysis,” in *4th International ICSC Symposium on Engineering of Intelligent Systems*, (Island of Madeira, Portugal), 2004.
- [20] B. Schölkopf, A. J. Smola, and K.-R. Müller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [21] B. Schölkopf, S. Mika, and e. al, “Input Space vs. Feature Space in Kernel-Based Methods,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.
- [22] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [23] S.Mika, B.Schölkopf, and e. al, “Kernel PCA and Denoising in Features Spaces,” in *Advances in Neural Information Processing Systems 11*, (San Mateo), pp. 536–542, Cambridge, MA: MIT Press, 1998.
- [24] B. Schölkopf, S. Mika, A. Smola, G. Ratsh, and K.R.Müller, “Kernel PCA Pattern Reconstruction via Approximate Pre-Images,” in *Proceedings of the 8th International Conference on Artificial Neural Networks*, (Berlin), pp. 147–152, Springer Verlag, 1998.
- [25] G. Bakir, J. Weston, and B. Schölkopf, “Learning to Find Pre-Images,” *Advances in Neural Information Processing Systems*, vol. 16, pp. 449–456, 2004.
- [26] A. Leontitsis, T. Bountis, and J. Pange, “Noise: A Short Simplified Review on Local Estimation and Reduction,” in *Non-linear Dynamics: Chaos and Complexity*, (Patras, Greece), 2002.

- [27] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Transactions Automatic Control*, vol. AC-19, pp. 716–723, December 1974.
- [28] G. Schwartz, "Estimating the Dimension of a Model," *Journal of Artificial Intelligence Research*, vol. 6, pp. 461–464, 1978.
- [29] M. Wax and T. Kailath, "Detection of Signals by Information Theoretic Criteria," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-33, pp. 387–392, April 1985.
- [30] A. P. Liavas and P. A. Regalia, "On the behavior of Information Theoretic Criteria for Model Order Selection," *IEEE Transactions on Signal Processing*, vol. 49, pp. 1689–1695, August 2001.
- [31] M. E. and C. M. Bishop, "Probabilistic Principal Component Analysis," 1999.
- [32] G. Gratton, "Dealing with Artifacts: The EOG Contamination of the Event-Related Brain Potential," *Behavior Research Methods, Instruments & Computers*, vol. 30, no. 1, pp. 44–53, 1998.
- [33] G. Garrick, R.E.Kass, A. Miller, J.F.Cohn, and N.A.Fox, "Automatic Correction of Ocular Artifacts in the EEG: a Comparison of Regression-based and Component-based Methods," *International Journal of Psychophysiology*, vol. 53, pp. 105–119, 2004.
- [34] T.-P. Jung, S. Makeig, and e. all, "Removing Electroencephalographic Artifacts by Blind Source Separation," *Psychophysiology*, vol. 37, no. 2, pp. 163–178, 2000.
- [35] A. Delorma and S. Makeig, "EEGLAB: An Open Source Toolbox for Analysis of Single-Trial EEG Dynamics Including Independent Component Analysis," *Journal of Neuroscience Methods*, vol. 134, pp. 9 – 21, 2004.
- [36] T.-P. Jung, S. Makeig, and e. al, "Removal of Eye Activity Artifacts from Visual Event-Related Potentials in Normal and Clinical Subjects.," *Clinical Neurophysiology*, vol. 111, no. 10, pp. 1745–1758, 2000.
- [37] J. Iriate, E. Urrestarazu, M. Valencia, M. Alegre, A. Malanda, C. Viteri, and J. Artieda, "Independent Component Analysis as a Tool to Eliminate Artifacts in EEG:A Quantitative Study," *Journal of Clinical Neurophysiology*, vol. 20, no. 4, pp. 249–257, 2003.
- [38] C. Joyce, I. Gorodnitsky, and M. Kutas, "Automatic Removal of Eye Movement and Blink Artifacts from EEG Data Using Blind Component Separation," *Psychophysiology*, vol. 41, pp. 313–325, 2003.
- [39] P.He, G.Wilson, and C.Russel, "Removal of Ocular Artifacts from Electro-encephalogram by Adaptive Filtering," *Medical & Biological Engineering & Computing*, vol. 42, no. 3, pp. 407–412, 2004.

- [40] S. Makeig, S. Debener, J. Onton, and A. Delorme, “EEGLAB,” <http://www.sccn.ucsd.edu/eeglab/>, 2005.
- [41] A. R. Teixeira, A. M. Tomé, E. W. Lang, P. Gruber, and A. d. Siva, “Removal of Ocular Artifacts from Electroencephalogram by Singular Spectrum Analysis,” in *2nd Int. Conf. Comput. Intelligence Medicine Healthcare (CIMED)*, (Costa da Caparica, Portugal), pp. 24–29, 2005.
- [42] A. R. Teixeira, A.M.Tomé, E.W.Lang, P.Gruber, and A. Silva, “On the Use of Clustering and Local Singular Spectrum Analysis to Remove Ocular Artifacts from Electroencephalograms,” in *International Joint Conference on Neuronal Networks, IJCNNt2005*, (Montreal, Canada), pp. 2514–2519, 2005.
- [43] P. Xi and T. Xu, “De-noising and Recovering Images Based on Kernel PCA Theory,” in *The 12-th International Conference in Central Europe on Computer Graphics Visualization and Computer Vision’2004* (W. 2004, ed.), (Plzen-Bory, Czech Republic), pp. 197–200, 2003.
- [44] R. Rosipal, M. Girolami, L.Trejo, and A.Cichocki, “Kernel PCA for Feature Extraction and De-noising in Non-Linear Regression,” *Neural Computing & Applications, Springer*, vol. 10, pp. 231–243, 2001.
- [45] M. Böhm, K.Stadlthnaner, E.W.Lang, A.M.Tomé, A.R.Teixeira, F.J.Theis, and C.G.Puntonet, “A Hybridization of Simulated Annealing and Local PCA for Automatic Component Assignment within ICA,” in *8th International Work-Conference on Artificial Neural Networks (IWANN’2005)*, (Barcelona), pp. 1075–1082, Springer, 2005.
- [46] M. Böhm, K.Stadlthnaner, P. Gruber, F.J.Theis, E.W.Lang, A.M.Tomé, A.R.Teixeira, W. Gronwald, and H. R. Kalbitzer, “On the Use of Simulated Annealing to Automatically Assign Independent Components,” *IEEE Transactions on Biomedical Engineering*, 2005 (accepted).
- [47] V. Franc and V. Hlaváč, “Greedy Algorithm for Training Set Reduction in the Kernel Methods,” in *Computer Analysis of Images and Patterns (CAIP03, ed.)*, pp. 426–433, 2003.
- [48] W. Zhou and J. Gotman, “Removing Eye-Movement Artifacts from the EEG during the Intracarotid Amobarbital Procedure,” *Epilepsia*, vol. 46, no. 3, pp. 409–414, 2005.
- [49] E. Urrestarazu, J. Iriarte, M. Alegre, M. Valencia, C. Viteri, and J. Artieda, “Independent Component Analysis Removing Artifacts in Ictal Recordings,” *Epilepsia*, vol. 45, no. 9, pp. 1071–1078, 2004.
- [50] L. Cao, K. Chua, W. K. Chong, H.P.Lee, and Q.M.Gu, “A Comparison of PCA, KPCA and ICA for Dimensionality Redution in Support Vector Machine,” *Neurocomputing Elsevier*, vol. 55, pp. 321–336, 2003.

Apêndice A

Decomposição em Valores Singulares - SVD

Uma matriz X ($m \times n$) decomposta em valores singulares pode ser escrita da forma

$$X = U\Sigma V', \quad (\text{A.1})$$

onde as colunas da matriz ortogonal U ($m \times m$) representam os vectores próprios de XX' , as colunas da matriz ortogonal V ($n \times n$) representam os vectores próprios de $X'X$ e a matriz Σ ($m \times n$) representa a matriz diagonal dos valores singulares.

Consideremos $C = X'X$ a matriz de covariância dos dados e $K = XX'$ a matriz de Kernel. Efectuando a decomposição em valores e vectores próprios de ambas as matrizes temos:

$$K = X'X \quad e \quad C = XX' \quad (\text{A.2})$$

Substituindo a eq.(A.1) nas eq. (A.2), temos:

$$K = X'X = V\Sigma U'U\Sigma V' = V\Sigma^2 V' \quad (\text{A.3})$$

e

$$C = XX' = U\Sigma V'V\Sigma U' = U\Sigma^2 U' \quad (\text{A.4})$$

onde as colunas ortogonais \mathbf{u} da matriz U correspondem aos vectores próprios da matriz de covariância, e as colunas ortogonais \mathbf{v} da matriz V correspondem aos vectores próprios da matriz de kernel e $V'V = I$ e $U'U = I$.

Pelas eq. (A.3 , A.4), concluí-se que a matriz de Kernel e a matriz de covariância de um conjunto de dados X , tem os mesmos valores próprios (Σ^2). Consequentemente $U\Sigma = XV$ e as k -colunas de U pode ser representada como uma combinação linear dos vectores da matriz de covariância:

$$\mathbf{u}_k = X\mathbf{v}_k\lambda_k^{-\frac{1}{2}} \quad (\text{A.5})$$

Apêndice B

Pré-Imagem - método das distâncias

Seja P_i , com $i = 1, \dots, n$ o conjunto dos n vizinhos mais próximos da pré-imagem que é procurada, P_{n+1} . Consideremos

$$P_{n+1} = (w_{n+1,1}, w_{n+1,2}, \dots, w_{n+1,d}) \quad (\text{B.1})$$

as coordenadas da pré-imagem d -dimensional que se pretende encontrar.

Segundo [10], para calcular a pré-imagem P_{n+1} é necessário considerar as n equações quadráticas das distâncias do ponto procurado aos n vizinhos mais próximos

$$d_{i,n+1}^2 = \sum_{k=1}^d (w_{k,n+1} - w_{k,i})^2 = d_{n+1}^2 + d_i^2 - 2 \sum_{k=1}^d w_{k,i} w_{k,n+1} \quad (\text{B.2})$$

Devido aos termos estarem centrados e assumindo que estas equações estão a ser calculadas à volta do i , o termo correspondente produto cruzado é nulo. Assim, considerando as n equações temos:

$$\sum_{i=1}^n d_{i,n+1}^2 = n d_{n+1}^2 + \sum_{k=1}^d d_k^2 \Leftrightarrow d_{n+1}^2 = \frac{1}{n} \sum_{i=1}^n d_i^2 - d_{i,n+1}^2 \quad (\text{B.3})$$

A partir da expressão anterior substituímos d_{n+1}^2 na equação B.2 e obtemos:

$$-2 \sum_{k=1}^d w_{k,i} w_{k,n+1} = d_i^2 - d_{i,n+1}^2 - \frac{1}{n} \sum_{i=1}^n (d_{i,n+1}^2 - d_i^2) \quad (\text{B.4})$$

Representando a expressão acima em notação matricial temos:

$$-2W^T \hat{\mathbf{w}} = (\mathbf{d}^2 - \mathbf{d}_0^2) - \frac{\mathbf{j}\mathbf{j}^T}{n} (\mathbf{d}^2 - \mathbf{d}_0^2) \quad (\text{B.5})$$

Como, $W \frac{\mathbf{j}\mathbf{j}^T}{n} = 0$ devido aos dados estarem centrados, a pré-imagem pode ser obtida

$$\hat{\mathbf{w}} = -\frac{1}{2}(WW^T)^{-1}W(\mathbf{d}^2 - \mathbf{d}_0^2) = -\frac{1}{2}\Sigma^{-1}V^T(\mathbf{d}^2 - \mathbf{d}_0^2) \quad (\text{B.6})$$

Apêndice C

Parâmetros dos algoritmos SSA e Local SSA

Nas tabelas C.1, C.2 e C.3 estão apresentados os parâmetros da aplicação dos algoritmos SSA e Local SSA ao sinal sinusoidal, sinal *funny* e ao sinal dente-de-serra apresentados nas secções 2.7.1 e 2.8.1. Foram considerados $q = 3$ e $q = 5$ *clusters* e duas janelas de dimensão $M = 11$ e $M = 36$ atrasos .

As tabelas ilustram a dimensão de cada *cluster* e o número de direcções seleccionadas em cada *cluster* pelo critério MDL.

| | | M=36 | | M=11 | |
|--------------------|----------|--------------|----------------|--------------|----------------|
| | Clusters | Dimensão (k) | Size (N_c) | Dimensão (k) | Size (N_c) |
| sem Cluster | 1 | 2 | 465 | 2 | 490 |
| 3 Clusters | 1 | 1 | 159 | 1 | 158 |
| | 2 | 1 | 152 | 1 | 164 |
| | 3 | 1 | 154 | 1 | 168 |
| 5 Clusters | 1 | 1 | 86 | 11 | 106 |
| | 2 | 1 | 98 | 11 | 93 |
| | 3 | 1 | 93 | 11 | 90 |
| | 4 | 1 | 98 | 11 | 105 |
| | 5 | 1 | 90 | 11 | 96 |

Tabela C.1: Relação entre a dimensão do subespaço k e o número de amostras em cada *cluster* da aplicação dos algoritmos SSA e Local SSA ao sinal sinusoidal

Pela análise das tabelas podemos concluir que:

- O sinal sinusoidal reconstruído com 5 *clusters* e $M = 11$, representa o sinal de entrada (sinal sinusoidal adicionado de ruído com SNR=5dB), uma vez que o critério MDL seleccionou em todos os *clusters* o número máximo de direcções, não havendo eliminação de ruído. O mesmo acontece para o sinal *funny* quando é aplicado o algoritmo SSA e $M = 11$ atrasos.

| | | M=36 | | M=11 | |
|--------------------|----------|--------------|----------------|--------------|----------------|
| | Clusters | Dimensão (k) | Size (N_c) | Dimensão (k) | Size (N_c) |
| sem Cluster | 1 | 2 | 465 | 11 | 490 |
| 3 Clusters | 1 | 8 | 174 | 2 | 64 |
| | 2 | 11 | 231 | 3 | 97 |
| | 3 | 2 | 60 | 11 | 239 |
| 5 Clusters | 1 | 2 | 64 | 11 | 149 |
| | 2 | 10 | 222 | 1 | 22 |
| | 3 | 2 | 59 | 2 | 171 |
| | 4 | 1 | 40 | 11 | 42 |
| | 5 | 3 | 80 | 3 | 106 |

Tabela C.2: Relação entre a dimensão do subespaço k e o número de amostras em cada *cluster* da aplicação dos algoritmos SSA e Local SSA ao sinal *funny*

| | | M=36 | | M=11 | |
|--------------------|----------|--------------|----------------|--------------|----------------|
| | Clusters | Dimensão (k) | Size (N_c) | Dimensão (k) | Size (N_c) |
| sem Cluster | 1 | 4 | 465 | 2 | 490 |
| 3 Clusters | 1 | 3 | 160 | 1 | 183 |
| | 2 | 3 | 140 | 1 | 124 |
| | 3 | 2 | 165 | 1 | 183 |
| 5 Clusters | 1 | 1 | 97 | 11 | 132 |
| | 2 | 2 | 91 | 11 | 175 |
| | 3 | 2 | 88 | 1 | 59 |
| | 4 | 3 | 80 | 1 | 59 |
| | 5 | 2 | 109 | 1 | 65 |

Tabela C.3: Relação entre a dimensão do subespaço k e o número de amostras em cada *cluster* da aplicação dos algoritmos SSA e Local SSA ao sinal dente-de-serra

- A diminuição da dimensão do *embedding* nos sinais origina origina a sobrestimação da dimensão do subespaço por parte do critério MDL (selecção $k = 11$ direcções).