



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E COMPUTAÇÃO

**Separação cega de fontes lineares e não lineares usando
Algoritmo Genético, Redes Neurais Artificiais RBF e
Negentropia de Rényi como medida de independência**

Nielsen Castelo Damasceno

Dissertação de Mestrado

**NATAL/RN - Brasil
2010**

Nielsen Castelo Damasceno

**Separação cega de fontes de fontes lineares e não lineares
usando Algoritmo Genético, Redes Neurais Artificiais RBF e
Negentropia de Rényi como medida de independência**

Dissertação submetida à Banca Examinadora do
Programa de Pós-Graduação e Engenharia
Elétrica e de Computação, da Universidade
Federal do Rio Grande do Norte, como parte
dos requisitos para obtenção do título de Mestre
em Ciências.

Orientador:

Prof. D.Sc. Allan de Medeiros Martins

Co-orientador:

Prof. D.Sc. Adrião Duarte Doria Neto

NATAL, RN - Brasil
2010

Nielsen Castelo Damasceno

**Separação cega de fontes lineares e não lineares usando
Algoritmo Genético, Redes Neurais Artificiais RBF e
Negentropia de Rényi como medida de independência**

Dissertação submetida ao corpo docente da Coordenação do Programa de Pós-graduação em Engenharia Elétrica e de Computação, da Universidade Federal do Rio Grande do Norte, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências.

Aprovado por:

Prof. Adrião Duarte Dória Neto, D.Sc. - UFRN

Prof. Allan de Medeiros Martins, D.Sc. - UFRN

Prof. Evandro Ottoni Teatini Salles, D.Sc. – UFSC

Prof. Jorge Dantas de Melo, D.Sc. - UFRN

AGRADECIMENTOS

Primeiramente agradecer a Deus por ter me dado toda sabedoria e amparo para chegar aonde estou.

Aos meus pais pelo amor incondicional, me permitindo chegar bem mais longe do que um dia acreditei. Aos meus irmãos, familiares e à família Dantas pelo apoio moral e espiritual. Muito Obrigado.

Um agradecimento especial à minha ex-noiva (início do mestrado), hoje minha querida esposa, visto que desde o primeiro instante me apoiou e caminhou ao meu lado, sua paciência ao ouvir-me explicar os detalhes da minha pesquisa e por permitir compartilhar meu entusiasmo por aquilo que faço.

Ao meu orientador D.Sc. Allan de Medeiros Martins e co-orientador D.Sc. Adrião Duarte Dória Neto, por fazer um aprendizado um bom trabalho. Obrigado pelas broncas (“no bom sentido”), pois sem elas não poderia chegar até aqui e principalmente pela paciência multiplicada por setenta vezes sete na doação sem limite e sem restrições, muito obrigado.

Aos professores da Faculdade Mater Christi, em especial ao professor Msc. Erick Augusto Pereira Caldas, pelas lições de ciências e orientações, ao professor D.Sc. Francisco das Chagas de Lima Júnior pelas sugestões nas equações matemáticas, à coordenação do curso de Sistemas de Informação da Mater Christi que fizeram deste objetivo uma realidade.

Ao professor D.Sc. José C. Príncipe, diretor do Laboratório Computacional Neuro Engenharia da Universidade da Flórida (CNEL), por compartilhar informações de suas equações. Sem sua contribuição, que eu admiro muito, esta dissertação não teria sido possível.

Finalmente aos meus amigos, colegas e professores da UnP, UFRN e Mater Christi que compartilharam todo o processo de desenvolvimento científico.

RESUMO

Os métodos convencionais para resolver o problema de separação cega de fontes não lineares em geral utilizam uma série de restrições à obtenção da solução, levando muitas vezes a uma não perfeita separação das fontes originais e alto custo computacional. Neste trabalho, propõe-se uma alternativa de medida de independência com base na teoria da informação e utilizam-se ferramentas da inteligência artificial para resolver problemas de separação cega de fontes lineares e posteriormente não lineares. No modelo linear aplica-se algoritmos genéticos e a *Negentropia* de Rényi como medida de independência para encontrar uma matriz de separação linear a partir de misturas lineares usando sinais de forma de ondas, áudios e imagens. Faz-se uma comparação com dois tipos de algoritmos de Análise de Componentes Independentes bastante difundidos na literatura. Posteriormente, utiliza-se a mesma medida de independência como função custo no algoritmo genético para recuperar sinais de fontes que foram misturadas por funções não lineares a partir de uma rede neural artificial do tipo base radial. Algoritmos genéticos são poderosas ferramentas de pesquisa global e, portanto, bem adaptados para utilização em problemas de separação cega de fontes. Os testes e as análises se dão através de simulações computacionais.

Palavras-chave: Análise de Componentes Independentes. *Negentropia* de Rényi. Algoritmos Genéticos. Redes Neurais.

ABSTRACT

Conventional methods to solve the problem of blind source separation nonlinear, in general, using series of restrictions to obtain the solution, often leading to an imperfect separation of the original sources and high computational cost. In this paper, we propose an alternative measure of independence based on information theory and uses the tools of artificial intelligence to solve problems of blind source separation linear and nonlinear later. In the linear model applies genetic algorithms and Rényi of negentropy as a measure of independence to find a separation matrix from linear mixtures of signals using linear form of waves, audio and images. A comparison with two types of algorithms for Independent Component Analysis widespread in the literature. Subsequently, we use the same measure of independence, as the cost function in the genetic algorithm to recover source signals were mixed by nonlinear functions from an artificial neural network of radial base type. Genetic algorithms are powerful tools for global search, and therefore well suited for use in problems of blind source separation. Tests and analysis are through computer simulations.

Keywords: Independent Component Analysis. *Negentropy Rényi*. Algorithms Genetics. Neural Network.

LISTA DE FIGURAS

Figura 1	O problema do <i>cocktail party</i>	20
Figura 2	Dois sinais do discurso original	21
Figura 3	Sinais dos discursos misturados	22
Figura 4	Sinais de fontes estimados a partir das misturas.....	23
Figura 5	Diagrama esquemático do problema de separação cega linear	23
Figura 6	Distribuição conjunta de duas variáveis aleatórias gaussianas.....	31
Figura 7	Gráfico superior é o histograma da VA x e inferior é da VA y	31
Figura 8	Distribuições <i>platokúrticas</i> , <i>kurtosis</i> de $s_1 = -1,1877$ e $s_2 = -1,2112$	31
Figura 9	Distribuição conjunta é uniforme em um quadrado	34
Figura 10	Distribuição das misturas dos componentes	34
Figura 11	Distribuição conjunta das estimativas usando PCA	34
Figura 12	Exemplo de funções de densidade de probabilidade para variáveis aleatórias normalmente distribuídas.	37
Figura 13	Distribuição normal bivariada	38
Figura 14	Gráfico de $H(p)$ em função de p	40
Figura 15	Gráfico de $H_2(p)$ e $H(p)$ em função de p	41
Figura 16	Esquema de estimação das componentes independentes.....	44
Figura 17	Mistura PNL e modelo de separação	50
Figura 18	Demonstração de estimação de fontes usando abordagem PNL.....	52
Figura 19	Distribuição conjunta dos sinais de fontes e das misturas PNL.....	53
Figura 20	Gráfico da função $f(x) \times \sin(10\pi x) + 1$	54
Figura 21	Pseudocódigo simples do Algoritmo Genético	56
Figura 22	Esquema básico de um Algoritmo Genético	56
Figura 23	Representação de 03 cromossomos	58
Figura 24	Representação da codificação real.....	58
Figura 25	Roleta viciada para quatro cromossomos	60
Figura 26	Processo de recombinação usando um ponto. a) Ponto de corte no cromossomo, b) resultado da recombinação.....	61
Figura 27	Processo de recombinação usando dois pontos. a) Ponto de corte no cromossomo, b) resultado da recombinação.....	62
Figura 28	Processo de mutação em um cromossomo	63

Figura 29	Redes RBF com 3 camadas	67
Figura 30	Exemplo simples do sistema misturador não linear	74
Figura 31	Imagem original com tamanho de 256x256 pixels.....	81
Figura 32	Imagem misturada com uma matriz aleatória de 3x3 não singular	82
Figura 33	Recuperação das imagens utilizando abordagem linear	82
Figura 34	Sinais de fontes originais com 50 amostras	83
Figura 35	Forma de onda misturada com uma matriz 3x3 aleatória.....	83
Figura 36	Forma de onda das fontes estimadas	83
Figura 37	Método AG com 100 amostras e população de 50 indivíduos	85
Figura 38	Método <i>P-ICA</i> com 100 amostras.....	85
Figura 39	Método AG com 1000 amostras e população de 50 indivíduos	86
Figura 40	Método <i>P-ICA</i> com 1000 amostras.....	86
Figura 41	Método AG utilizando apenas 20 amostras das 1000 amostras que representam os sinais	87
Figura 42	Método <i>P-ICA</i> utilizando todas as 1000 amostras	87
Figura 43	Recuperação de sinais de áudio usando <i>Negentropia</i> de <i>Renyi</i>	88
Figura 44	Recuperação dos sinais de áudio usando <i>P-ICA</i>	89
Figura 45	Diagrama esquemático do problema de separação cega não linear.....	89
Figura 46	Separação de fontes de uma onda quadrada e senoidal usando abordagem não linear.....	92
Figura 47	Separação de sinais com 200 amostras, na parte inferior da figura, foram sobrepostos os sinais de fontes e os estimados, em que os sinais azul são sinais de fontes e vermelho os estimados	93
Figura 48	Recuperação de imagens usando abordagem não linear para uma imagem com 128x128 pixels	94
Figura 49	Recuperação de imagens usando abordagem não linear para uma imagem com 256x256 pixels	95
Figura 50	Separação de sinais de áudio utilizando AG não linear.....	95
Figura 51	Ilustração de sinais de fontes com 334 amostras usando AG não linear	96
Figura 52	Sinais de fontes em cor azul e estimados e vermelho, sobrepostos.....	96
Figura 53	Modelo do sistema linear com adição de um ruído gaussiano	98
Figura 54	Separação de fontes lineares com adição de ruído gaussiano.....	98
Figura 55	Erro da soma dos mínimos quadrados para cada sinal versus a matriz de separação em cada geração	99

Figura 56	Separação de fontes não linear com ruído no sistema de separação.....	100
Figura 57	Sinais de fontes originais e estimadas sobrepostos com a adição de ruído no sistema de separação	100
Figura 58	Erro da soma dos mínimos quadrados para cada sinal, versus os sinais originais para cada geração com adição de ruído	101

LISTA DE TABELAS

Tabela 1	Conjunto de função de base radial.....	69
Tabela 2	Exemplos de função kernel.....	77
Tabela 3	Erro entre AG e abordagem <i>P-ICA</i> e <i>FastICA</i>	84
Tabela 4	Análise da <i>Negentropia</i> de Rényi dos sinais da Figura 51	90
Tabela 5	Análise da <i>Negentropia</i> de Rényi dos sinais da Figura 50	90
Tabela 6	Erro mínimo quadrado no AG não linear	97

LISTA DE ABREVIATURAS

ACI	Análise de Componente Independente
AFA	Análise Fatorial Adaptativo
AG	Algoritmo Genético
BSS	<i>Blind Source Separation</i> (Separação Cega de Fontes)
<i>cdf</i>	<i>cumulative distribution function</i> (Função de distribuição cumulativa)
CNEL	<i>Computacional NeuroEngineering Lab</i>
ECG	Eletrocardiograma
EEG	Eletroencefalograma
<i>fdp</i>	Função de distribuição de probabilidade
FECG	Eletrocardiograma Fetal
HOS	<i>Higher Order Statistics</i> (Estatística de Ordem Superior)
ICA	<i>Independent Component Analysis</i>
JADE	<i>Joint Approximate Diagonalization of Eigenmatrices</i>
KLT	Transformada de <i>Karhunen-Loève</i>
MECG	Eletrocardiograma Maternal
MEG	Magnetoencefalograma
NLICA	<i>Nonlinear Independent Component Analysis</i>
PCA	<i>Principal Component Analysis</i> (Análise de Componente Principal)
PNL	<i>Post-Nonlinear</i>
RBF	<i>Radial Basis Functions</i> (Função de Base Radial)
VA	Variável aleatória

LISTA DE SÍMBOLOS

\mathbf{A}	Matriz de Mistura
\mathbf{A}^{-1}	Inversa da matriz \mathbf{A}
$J(\cdot)$	Negentropia
\mathbf{W}, \mathbf{B}	Matriz de separação
\mathbf{C}	Matriz de centros iniciais
$\hat{\mathbf{C}}$	Matriz de centros estimados
$\hat{\mathbf{W}}$	Matriz de pesos estimados
\mathbf{x}	Vetor de mistura
\mathbf{x}_i	Sinal de uma mistura
x	Variável aleatória
x_g	Variável aleatória gaussiana
\mathbf{s}	Vetor de sinais de fontes
s_i	Sinal de uma de fonte independente
$F(\cdot)$	Sistema misturador
$G(\cdot)$	Sistema separador
L	Memorias atrasadas ou número de bits
y	Variável aleatória
\mathbf{y}	Vetor de estimativa das fontes
$p_x(x)$	Função de densidade de probabilidade da va aleatória x
$p_{x,y}(x, y)$	Função de densidade de probabilidade conjunta das va de x e y
$E[\cdot]$	Valor esperando
n, k	Instante de tempo ou número de amostras
i	Número de pixels ou amostras
\mathbf{yr}	Vetor de sinais de fontes com adição de ruído
P	Matriz de permutação

$\text{Inteiro}(\cdot)$	Função que retorna a parte inteira da expressão entre parêntese
\bar{x}	Valor médio
σ	Desvio padrão ou parâmetro de suavização do kernel
σ^2	Variância de uma va normal
α	Parâmetro da entropia
μ	Valor esperado de uma variável aleatória normal
K	Kurtosis
Σ	Matriz de covariância
R_{xx}	Matriz de autocorrelação
C_{xx}	Matriz de autocovariancia
R_{xy}	Matriz de correlação cruzada
C_{xy}	Matriz de covariância cruzada
I	Matriz identidade
Im	Imagem
$I(\cdot)$	Medida de Independência estatística
z	Vetor de mistura branqueado
E, U	Matriz ortogonal
D, Λ	Matriz diagonal
\hat{A}	Estimativa da matriz A
t	Trasposta de uma matriz ou vetor
$ \cdot $	Determinante de uma matriz
$H(\cdot)$	Entropia de uma variável aleatória
$\psi(\cdot)$	Função contrate
v	Vetor aleatório
V	Potencial da informação
k	Constante de normalização
$k(\cdot)$	Função kernel

$\ \cdot\ $	Norma euclidiana
$\mathbf{r}(\cdot)$	Vetor de ruído gaussiano
w	Vetor de pesos sinápticos
φ	Funções arbitrárias de base radial

SUMÁRIO

1	INTRODUÇÃO	17
2	ANÁLISE DE COMPONENTES INDEPENDENTES	20
2.1	BREVE HISTÓRICO	24
2.2	APLICAÇÕES ICA.....	25
2.2.1	Monitoramento de batimentos cardíacos	25
2.2.2	Cancelamento de ruído e interferência.....	25
2.2.3	Sistemas de comunicação digital	26
2.3	CONCEITOS BÁSICOS	26
2.3.1	Independência estatística	26
2.3.2	Conceito de momento	27
2.3.3	Primeiro Momento... ..	27
2.3.4	Momento central.....	28
2.3.5	Segundo Momento	28
2.3.6	Terceiro Momento	29
2.3.7	Quarto Momento	30
2.3.8	Correlação	32
2.3.9	Análise de Componentes Principais	33
2.3.10	Branqueamento.....	35
2.3.11	Distribuições gaussianas	36
2.3.12	Misturas gaussianas.....	38
2.3.13	Entropia.....	39
2.3.14	Entropia de <i>Shannon</i>	40
2.3.15	Entropia de <i>Rényi</i>	41
2.3.16	Negentropia	42
2.4	DEFINIÇÃO DE ICA	43
2.4.1	As ambiguidades ao modelo	45
2.5	SEPARAÇÃO DE MISTURAS LINEARES	45
2.5.1	Algoritmo ICA utilizando PCA	46
2.5.2	Separação de fontes usando Negentropia.....	47
2.6	SEPARAÇÃO DE MISTURAS NÃO LINEARES	48
2.6.1	Modelo de separação PNL	50

3	ALGORITMOS GENÉTICOS	54
3.1	ESQUEMA BÁSICO DE UM ALGORITMO GENÉTICO	55
3.2	REPRESENTAÇÃO CROMOSSÔMICA	56
3.3	INICIALIZAÇÃO DA POPULAÇÃO	59
3.4	AValiação	60
3.5	SELEÇÃO	60
3.6	REPRODUÇÃO	61
3.6.1	Recombinação	61
3.6.2	Mutação	63
3.7	OUTROS PARÂMETROS	63
4	REDES NEURAIS	65
4.1	O NEURÔNIO ARTIFICIAL	65
4.2	ARQUITETURA	66
4.3	RBF (RADIAL BASIS FUNCTIONS)	67
4.3.1	Arquitetura RBF	67
5	MÉTODO PROPOSTO	71
5.1	SISTEMA MISTURADOR OU SEPARADOR	72
5.2	ENTROPIA DE <i>RÉNYI</i> PARA UMA VARIÁVEL ALEATORIA GAUSSIANA	74
5.3	FUNÇÃO KERNEL	76
5.4	JANELAS DE <i>PARZEN</i> COM ENTROPIA QUADRÁTICA DE <i>RÉNYI</i>	77
6	RESULTADOS EXPERIMENTAIS	80
6.1	SEPARAÇÃO CEGA DE FONTES LINEARES	80
6.1.1	Comparação entre o método proposto e outras abordagens	84
6.2	SEPARAÇÃO CEGA DE FONTES NÃO LINEARES	89
6.2.1	Algoritmo Genético com mistura não linear	91
6.3	SEPARAÇÃO CEGA DE FONTES COM ADIÇÃO DE RUÍDO	97
7	CONSIDERAÇÕES FINAIS	102
	REFERÊNCIAS	104

1 INTRODUÇÃO

Nos últimos anos, as técnicas de processamentos de sinais sofreram consideradas evoluções. O objetivo era separar sinais de fontes de um ou mais sinais de interferência, sendo esta linha de pesquisa iniciada por *Winner* (WINNER, 1949). Os trabalhos de *Bode* e *Shannon* no final dos anos 40 e no início dos anos 50 (BOLDE, SHANNON, 1950) iniciaram um estudo de filtragem temporal (HAYKIN, 2001b). Recentemente, processamentos de sinais tornou-se um tema importante em pesquisas contemporâneas, em virtude da ampla gama de aplicações na engenharia. O surgimento de novas técnicas de separação de sinais podem ser aplicados nos casos em que as técnicas mais antigas não podem ser usadas, possibilitando resoluções de problemas nas áreas das ciências como, por exemplos, a separação de imagens, reconhecimento de falas e outras pesquisas mais atuais (BIRBAUMER, 2007).

A Análise de Componentes Independentes (ACI) recebeu atenção mais ampla em diversas áreas, tal como processamentos de sinais biomédicos (MÜLLER; VIGARIO, 2004). Portanto, Análise de Componentes Independentes é um popular algoritmo de processamentos de sinais e separação cega de fontes. Na verdade, um grande número de algoritmos de separação cega de fontes está disponível, mais especificamente algoritmos ACI, que utilizam Algoritmos Genéticos empregando como função custo a *Negentropia* (KAI; MINGLI, 2006). Podemos também destacar um estudo preliminar feito por *Yoshiota* et al. (YOSHIOTA; OMATU, 1998), que emprega Algoritmos Genéticos (AG) para separar imagens corrompidas de ruídos usando a divergência de *Kullback Leibler*. Em segundo momento, no trabalho de *Tan* (2000) utilizou-se AG para resolver um problema de separação cega de fontes não lineares. E finalmente, o AG foi bem aplicado pela primeira vez em filtragem eletrocardiograma (ECG) (PALANIAPPAN; GUPTA, 2006).

Neste trabalho, propomos a utilização da *Negentropia* de *Rényi* como função objetivo no Algoritmo Genético para solucionar um problema de BSS (do inglês: *Blind Source Separation* ou Separação Cega de Fontes) para o caso de misturas lineares e não lineares. Em particular, a utilização da informação mútua de *Shannon* pode ser descrito como a soma das entropias marginais de *Shannon* menos o conjunto das entropias, onde a dificuldade desta está nas estimativas das entropias marginais. Para estimar a entropia marginal, *Comon* e outros pesquisadores aproximaram a saída marginal da função de densidade de probabilidade (*f_{dp}* ou do inglês *probability density functions* - *pdf*)

utilizando expansões polinomiais (CHOI; CICHOCKI; AMARI, 2000), que naturalmente apresenta um erro no procedimento de estimação (COMON, 1994). Há, também, a abordagem paramétrica em BSS, onde o usuário (ou projetista) assume parâmetros específicos para o modelo de distribuição de fontes baseado no conhecimento prévio do problema (CHOI; CICHOCKI; AMARI, 2000). Notadamente, as contribuições de *Alfred Rényi* (HYVÄRINEN, 1999b) mostraram que as definições de *Shannon* para estimação da entropia e informação mútua eram, na realidade, casos particulares de uma família mais geral das definições. Estas definições generalizadas são chamadas de entropia de *Rényi* e informação mútua de *Rényi*, embora as definições de *Rényi* englobem as definições de *Shannon* como um caso especial para o parâmetro da família correspondente a 1. Por causa do amplo reconhecimento do trabalho de *Shannon* em virtude às suas implicações significativas nas teorias das comunicações, o trabalho de *Rényi* não foi muito bem apreciado como uma ferramenta útil por pesquisadores de engenharia e outras áreas.

Na década de 90, alguns interesses na entropia de *Rényi* em campos diferentes, incluindo reconhecimento de padrões (SAHOO; WILKINS; YEAGER, 1997) e criptografia (CACHIN, 1997), vêm se difundindo. Em filtragem adaptativa, *Principe* e seus colaboradores no CNEL (*Computational NeuroEngineering Lab, University of Florida*) deram os passos iniciais para quebrar o legado da entropia de *Shannon* (XU; PRINCIPE et al, 2002). Eles aplicaram, com sucesso, a entropia de *Rényi* e derivados do critério de otimização em problemas de separação cega de fontes, redução da dimensionalidade e extração de características.

Atualmente, há vários trabalhos utilizando entropia de *Rényi* como, por exemplo, o algoritmo proposto por *Xu* (2002). Este evita a expansão polinomial empregando Janelas de Parzen para estimar diretamente a entropia conjunta de *Rényi* (PARZEN, 1967).

Diante deste histórico, viu-se a necessidade de se fazer estudos de BSS, em que utilizaremos um cenário de mistura linear e não linear. Utiliza-se duas ferramentas bastante difundidas na inteligência artificial, os Algoritmos Genéticos e Redes Neurais Artificiais, especificamente as Redes de Base Radial (RBF).

Como função objetivo no Algoritmo Genético utiliza-se uma medida de independência, a *Negentropia de Rényi*.

No cenário linear utiliza-se uma matriz quadrada para gerar as misturas com os sinais de fontes. A partir desta mistura, o Algoritmo Genético busca uma matriz de separação para recuperar os sinais de fontes.

No cenário não linear, a mistura é gerada por uma rede RBF. Assim, o Algoritmo Genético tem o objetivo de encontrar outra rede RBF que separe os sinais de fontes a partir da mistura.

As análises e os resultados se dão através da implementação computacional de modelos matemáticos e numéricos para a solução do problema. Quando da utilização prática de algoritmos genéticos em problemas de separação cega de fontes, em cenários não lineares, torna-se um problema complexo e, muitas vezes, alto custo computacional. Têm-se vários trabalhos publicados usando entropia de *Rényi*, em que alguns mostram que se pode recuperar fontes com redução dos números de amostras, servindo de subsídios e incentivos para utilização de ferramentas da inteligência artificial e para soluções de problemas de separação cega de fontes.

Outra relevância para este estudo é o fato de estarmos trabalhando com um algoritmo genético, em cada indivíduo deste algoritmo é uma rede neural do tipo RBF, cujo objetivo é inverter o processo de mistura inicial, gerado por outra rede neural, também RBF, e assim recuperar as fontes originais.

Os algoritmos implementados foram utilizados para avaliar os efeitos causados no sistema proposto, abrangendo problemas de testes empíricos.

A estrutura desta dissertação está organizada da seguinte forma: na Seção 1 elabora-se um breve histórico sobre processamentos de sinais, ICA; na Seção 2 será exposto ICA no contexto de misturas lineares, bem como conceitos de estatística, distribuição de probabilidade, entropia de *Shannon*, entropia de *Rényi*, a *Negentropia* e também misturas não-lineares; na Seção 3, descreve-se o Algoritmo Genético proposto neste trabalho; na Seção 4 será abordado sobre redes neurais artificiais, especificamente RBF; na Seção 5 será descrito sobre o método proposto, a *Negentropia* de *Rényi*; na Seção 6 serão discutidos os resultados experimentais: aplicações de BSS para o caso linear e posteriormente o não-linear; finalmente, a Seção 7 será sobre as considerações finais.

2 ANÁLISE DE COMPONENTES INDEPENDENTES

Nesta seção, elaboram-se conceitos básicos relacionados à separação cega de fontes, apresentam-se também algumas aplicações e em seguida à formalização matemática.

Em linhas gerais, podemos dizer que a motivação do uso de Análise de Componentes Independentes (ACI) ou do inglês: *Independent Componente Analysis* (ICA) é o BSS (*Blind Source Separation* ou Separação Cega de Fontes) (HYVÄRINEN, OJA, 1999). Todavia, vamos utilizar ACI e ICA, doravante, para representar a mesma entidade. Sobretudo, um dos problemas típicos investigados pela técnica de separação cega de fontes é motivado por um problema chamado “*cocktail party*” ou separação de sinais de áudio.

Considere duas pessoas conversando em uma sala fechada utilizando sensores (microfones) para capturar suas vozes. Esta situação é representada na Figura 1. O problema consiste em separar os sinais captados pelos microfones sabendo que os sinais estão agora correlacionados. A particularidade da separação cega de fontes perante as outras técnicas de filtragens é que, nesse caso, não precisamos conhecer precisamente os sinais de fontes (HYVÄRINEN, 1999a).

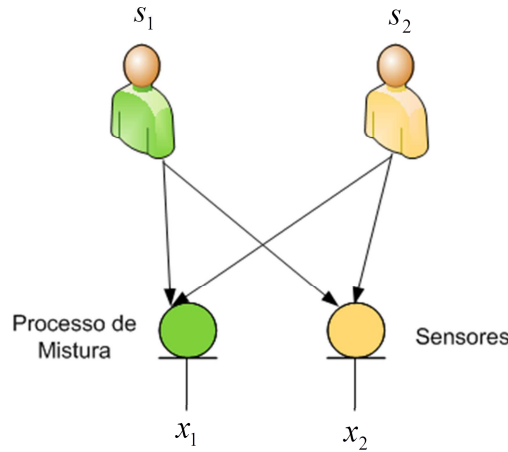


Figura 1: O problema do *cocktail-party*.

O problema do *cocktail-party* pode ser representado da seguinte forma: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ e } \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \text{ ou seja,}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad (2.1)$$

ou pode-se reescrever a Equação 2.1 utilizando a notação matricial, que tornar-se-á:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2.2)$$

Denota-se \mathbf{x} pelo vetor aleatório cujos elementos representam as misturas ou sensores, a matriz \mathbf{A} com elementos a_{ij} representam a atenuação ou amplificação sobre o vetor aleatório \mathbf{s} que representam o sinal de fontes s_1 e s_2 .

Por enquanto, deixemos de lado qualquer momento de atraso e outros fatores extras a partir de nosso modelo simplificado de mistura. Como ilustração, considere os sinais representados na Figura 2 e 3.

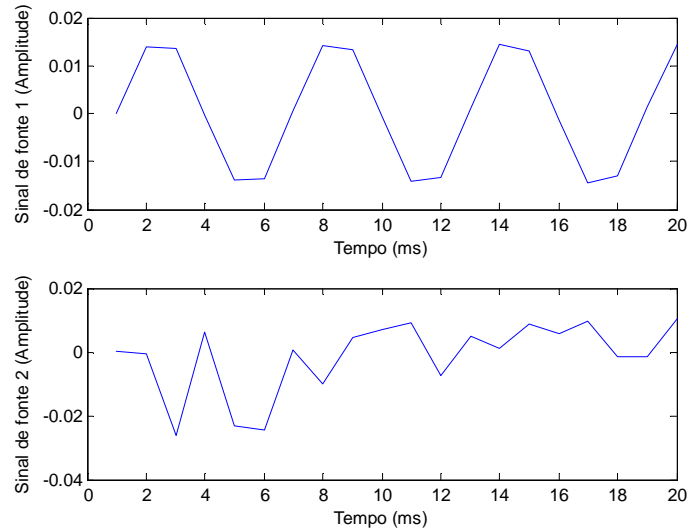


Figura 2: Dois sinais do discurso original.

Os sinais do discurso original é semelhante aos sinais representado na Figura 2 e as misturas poderiam se parecer com os sinais na Figura 3. Nos gráficos acima as coordenadas abscissas representam o número de amostra do sinal em cada período de tempo e a ordenada representa a amplitude do sinal. O problema consiste em recuperar os dados na Figura 2 utilizando apenas os dados da Figura 3.

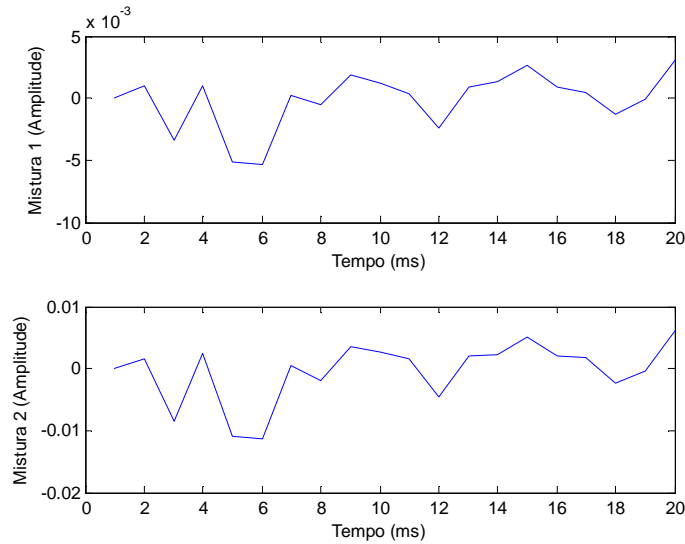


Figura 3: Sinais dos discursos misturados.

Na verdade, se soubéssemos os parâmetros a_{ij} poderíamos resolver o sistema na Equação 2.1 pelos métodos clássicos. O ponto, porém, é que não sabemos o valor de a_{ij} , de forma que o problema se torna consideravelmente difícil.

Uma abordagem para resolver este problema seria usar alguma informação estatística sobre as propriedades dos sinais $s_i(n)$ para estimar a todos. Na verdade, e talvez surpreendentemente, verifica-se que ela seja suficiente para presumir que $s_1(n)$ e $s_2(n)$ a cada instante de tempo n são estatisticamente independentes. A técnica recentemente desenvolvida conhecida como ICA pode ser usado para estimar os valores de a_{ij} baseado nas informações de sua independência, o que permite recuperar ou estimar o sinal original $s_1(n)$ e $s_2(n)$ a partir de suas misturas $x_1(n)$ e $x_2(n)$.

A Figura 4 representa os sinais de fontes estimados por ICA usando abordagem PCA (KUN; CHAN, 2006).

Na Figura 5 ilustra um diagrama esquemático do problema, onde um conjunto de sinais de fontes são submetidos à ação de um sistema misturador, cujas saídas correspondem a misturas de tais fontes. Devemos, de alguma forma, projetar um sistema separador que seja capaz de estimar as fontes, ou seja, inverter a ação do sistema misturador. Este problema é dito cego em razão da falta de informação que temos sobre as misturas e as fontes.

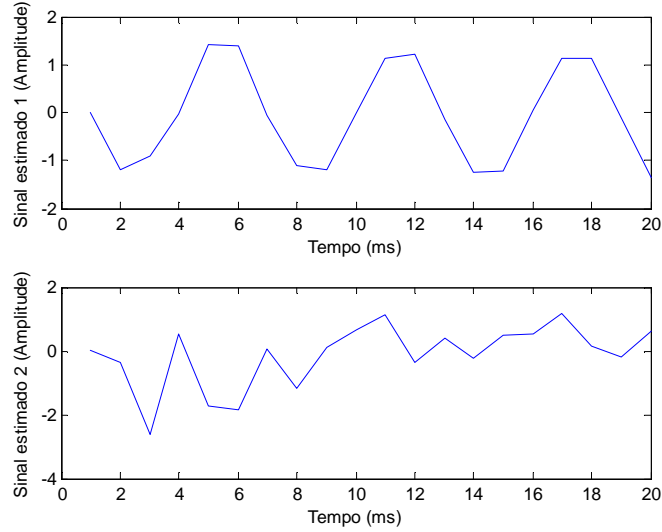


Figura 4: Sinais de fonte estimados a partir das misturas.

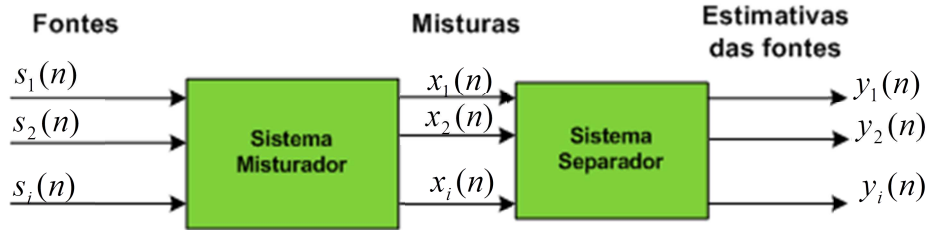


Figura 5: Diagrama esquemático do problema de separação cega linear.

Podemos também representar de forma simples o processo de misturas das fontes pela seguinte expressão:

$$x(n) = F(s(n), s(n-1), \dots, s(n-L), r(n)) \quad (2.3)$$

onde o $F(\cdot)$ corresponde a ação do sistema misturador, L é associado às memórias (amostras atrasadas) no sistema e o vetor r representa o ruído presente nas fontes. Um sistema misturador é dito linear se o mapeamento $F(\cdot)$ atende o princípio da superposição, caso contrário é dito não linear.

Nas situações em que o sistema misturador depende das amostras passadas ($L > 0$) é dito que o sistema misturador é convolutivo (com memória). Entretanto, há situações em que ($L = 0$) o sistema é chamado de instantâneo (COMON, 1994).

Em outras situações onde o número de sensores podem ser maiores que o número de sinais de fontes, tem-se o caso sobre determinado. Analogamente, temos o caso subdeterminado quando o número de sensores é menor que os sinais de fontes.

2.1 BREVE HISTÓRICO

ICA está intimamente ligado à separação cega de fontes, haja vista ser uma ferramenta tipicamente utilizada para resolver problemas de separação cega de fonte e teve seu início nos anos 80, com o trabalho de Héroult e Jutten (HÉRAULT; JUTTEN, 1985). O trabalho foi motivado pelo estudo de processamento de sinais neurofisiológico que consistia em um método simples de movimento muscular, ao passo que com apenas um único sinal codificava-se dois sinais: o deslocamento e a velocidade angular do movimento. Através deste modelo simplificado, abriram-se as margens para várias outras pesquisas e vertentes.

Outros pesquisadores que contribuíram com o avanço desse tema foram os franceses *Jean François Cardoso* (CARDOSO, 1998) e *Pierre Comon* (COMON, 1994). Eles contribuíram com a formalização matemática explorando dois temas (CARDOSO, 1998): teoria da informação e estatística de ordem superior (*Higher Order Statistics* - HOS).

O trabalho de *Jean Cardoso* consiste num estudo sobre o estimador de máxima verossimilhança em BSS (CARDOSO, 1998), que contribuiu para o desenvolvimento do BSS. Ele também desenvolveu um método de otimização conhecido como gradiente relativo (AMARI; YANG et. al., 1997).

Com o desenvolvimento do BSS/ICA, destacam-se os trabalhos do grupo composto pelos pesquisadores finlandeses: *Karhunen*, *Oja* e *Hyvärinen*. Eles interpretaram ICA como uma extensão não linear da técnica Análise de Componentes Principais (*Principal Component Analysis* – PCA). Destaca-se *Hyvärinen* com a contribuição do critério da maximização da não-gaussianidade e do algoritmo *FastICA*.

Além dessas contribuições expostas anteriormente há outro trabalho importante. Em 2006, *Kun Zhang* e *Lai-Wan Chan*, da Universidade de Hong Kong, desenvolveram uma nova abordagem de ICA usando PCA (KUN; CHAN, 2006).

Portanto, algoritmos baseados em ICA vêm fornecendo boas soluções e são aplicados em diversos campos. A seguir, são descritos de forma sucinta algumas destas aplicações.

2.2 APLICAÇÕES DE ICA

Um dos grandes desafios da engenharia biomédica é na avaliação das alterações fisiológicas que ocorrem em diversos órgãos internos do corpo humano. Existem problemas nas extrações das informações relevantes para diagnósticos, ou seja, sinais de fontes biomédicos são geralmente fracos, não estacionários, com ruídos e interferências (CICHOCKI; AMARI, 2002). A seguir têm-se algumas aplicações da ICA em problemas de separação cega de fontes.

2.2.1 Monitoramento de batimentos cardíacos

A ação mecânica dos músculos do coração é estimulada por sinais elétricos. Estes níveis de sinais podem ser medidos e visualizados como funções de tempo usando eletrocardiograma (ECG) (CICHOCKI; AMARI, 2002). Tal como para adultos também seria possível medir a atividade elétrica do coração de um feto. As características de um eletrocardiograma fetal (FECG) podem ser muito úteis para determinar se um feto está se desenvolvendo corretamente, por exemplo. Estas características incluem uma elevação da frequência cardíaca fetal que indica estresse, arritmias. Obter uma informação fiel do FECG é uma tarefa não trivial. Problemas podem acontecer em virtude de que o eletrocardiograma também contém informações dos batimentos cardíacos materno (MECG) (JAHN; AMARI et al., 1999). Além disso, o FECG irá ocasionalmente sobrepor sinais ao MECG e torná-lo normalmente difícil de detectar (CARDOSO, 1998). Também juntamente com o MECG ruídos extensivos nestes sensores interferem no FECG e podem mascarar completamente este. A separação destes sinais de fontes fetal e materno de uma mulher grávida pode ser modelado como um problema BSS (HAYKIN, 2001a).

2.2.2 Cancelamento de ruído e interferência

O sistema nervoso dos seres humanos e dos animais deve codificar e processar informações sensoriais. Dentro deste contexto, os sinais codificados (as imagens, sons) têm propriedades muito específicas. Uma das tarefas desafiadoras é a de saber como

fielmente detectar, localizar e melhorar os sinais cerebrais em que muitas vezes as fontes são corrompidas. ICA e Análise Fatorial Adaptativo (AFA) são abordagens promissoras para a eliminação de artefatos e ruídos provenientes dos EEG / MEG (HE; REED, 1996) (JAHN; AMARI et al., 1999).

2.2.3 Sistemas de comunicação digital

Considere um sistema onde se têm múltiplos sinais de uma propagação de comunicação sem fio, bem como um número de usuários difundidos em sinais modulados digitalmente para uma estação base em um ambiente de vários usuários. A transmissão destes sinais interage com diversos objetos físicos na região antes de chegar à antena ou estação de base. Cada sinal segue caminhos diferentes com atraso e atenuação. Este é um típico problema que é conhecido como “*multi-path fading*” (CARDOSO, 1998). Além disto, em algumas redes de celulares existem outras fontes adicionais de distorções. Estas interferências podem ser causadas por vários utilizadores que partilham a mesma frequência e tempo. Um problema desafiador é a separação e processamento de sinais cegos conjunta de espaço-tempo e equalização dos sinais transmitidos, isto é, para estimar a fonte de sinais e seus canais na presença de outros sinais e ruído (HAYKIN, 2001a).

2.3 CONCEITOS BÁSICOS

Esta seção é dedicada a apresentar alguns conceitos de independência estatística, momentos e cumulantes que são comumente utilizados na caracterização da distribuição.

2.3.1 Independência estatística

A separação cega de fontes consiste em recuperar os sinais originais a partir de uma mistura. Um princípio bastante utilizado para determinar ou inferir nas misturas

tem sido a independência estatística, ou seja, o valor de qualquer um dos componentes não fornece nenhuma informação sobre os valores dos outros componentes.

Normalmente, uma distribuição de probabilidade é caracterizada em termos de sua função de densidade ao invés de *cdf* (função de distribuição cumulativa ou do inglês: *cumulative distribution function*). Formalmente, a função de densidade de probabilidade é obtida derivando *cdf*. ICA está intimamente ligado à independência estatística. Matematicamente, a independência estatística é definida em termos de densidade de probabilidade (PAPOULIS, 1991), por exemplo. As variáveis aleatórias x e y são ditas independentes, se e somente se,

$$p_{x,y}(x,y) = p_x(x)p_y(y) \quad (2.4)$$

Em outras palavras, a densidade conjunta $p_{x,y}(x,y)$ de x e y devem fatorar no produto das suas densidades marginais $p_x(x)$ e $p_y(y)$. Equivalente à independência, esta pode ser definida pela substituição das funções de densidade de probabilidade na Equação 2.4 pelas respectivas funções de distribuição cumulativa, que também deve ser fatoráveis.

2.3.2 Conceito de Momento

Como estamos interessados em analisar a estrutura de vetores aleatórios, podemos fazer uma simples pergunta: O quão fortemente esses vetores dependem uns dos outros? Isto pode ser medido com uma aproximação ou utilizando as correlações (posteriormente faremos uma definição clássica deste conceito). Entretanto, qualquer variável aleatória (VA) pode ser caracterizada pela sua função de distribuição de probabilidade, que pode ser descrita por um conjunto de parâmetros designados por momentos.

O conceito de momento é muito importante para o caso em que não tenham nenhuma informação das misturas. Pode-se usar o conceito de momento para caracterizar as distribuições dos sinais estimados (HYVÄRINEN; OJA, 1999).

2.3.3 Primeiro Momento

O valor esperado é uma medida muito importante em estatística, já que é muito usado para descrever distribuições, avaliar o desempenho dos estimadores, bem como obter dados de testes e muitas outras aplicações.

O primeiro momento de uma fdp p_x corresponde ao valor médio \bar{x} do sinal x .

A média do valor \bar{x} é também conhecida como valor esperado ou esperança $E[x]$ da variável x (PAPOULIS, 1991). Para uma variável x com fdp p_x o primeiro momento é dado por:

$$E[x] = \int_{x=-\infty}^{\infty} p_x(x) x dx \quad (2.5)$$

2.3.4 Momento central

Precisamos encontrar um número que resuma as características das misturas. Estamos interessados em 2 tipos principais de medidas numéricas: as que caracterizem a localização do centro das misturas e as que caracterizem a dispersão dos dados. O momento central descreve o comportamento da distribuição em relação ao seu valor médio, sendo definido como:

$$E[(x - \bar{x})^2] \quad (2.6)$$

2.3.5 Segundo Momento

As medidas de tendência central não são as únicas medidas necessárias para caracterizar as misturas. Precisamos saber o quanto as misturas estão “espalhadas”. A seguir, tem-se o segundo momento $E[x^2]$ de uma fdp de x que é dado por:

$$E[x^2] = \int_{x=-\infty}^{+\infty} p_x(x) x^2 dx \quad (2.7)$$

Também pode ser demonstrado que:

$$E[x^2] = E[x^2] + E[(x - E[x])^2] \quad (2.8)$$

$$E[x^2] = \bar{x}^2 + E[(x - \bar{x})^2] \quad (2.9)$$

onde $E[(x - \bar{x})^2]$ é conhecido como a variância de x . O segundo momento central de x^2 é a variância de $(x - \bar{x})$. A raiz quadrada da variância é o desvio padrão, denotada por σ e é uma importante medida de variabilidade, conforme a expressão a seguir:

$$\sigma = \sqrt{E[(x - \bar{x})^2]} \quad (2.10)$$

De forma que podemos escrever a Equação 2.9 como:

$$E[x^2] = \bar{x}^2 + \sigma^2 \quad (2.11)$$

2.3.6 Terceiro Momento

A média e a variância são casos particulares que chamamos de “momentos” de uma distribuição de probabilidade. Os momentos de uma distribuição servem para caracterizar a distribuição de probabilidade, não apenas no que se refere a sua centralidade e dispersão, mas também com relação a outras características como assimetria ou simetria da densidade de probabilidade, por exemplos.

A seguir, o terceiro momento $E[x^3]$ de uma *fdp* de x é dado por:

$$E[x^3] = \int_{x=-\infty}^{+\infty} p_x(x)x^3 dx$$

O momento central de x^3 é frequentemente chamado de medida de assimetria ou *skewness*. As distribuições normal e uniforme são exemplos de distribuições simétricas. Mais genericamente, a *skewness* é definida como:

$$E[(x - \bar{x})^3] = \int_{-\infty}^{+\infty} p_x(x)(x - \bar{x})^3 dx \quad (2.12)$$

2.3.7 Quarto Momento

Para medir o grau de “achatamento” de uma distribuição de dados usa-se uma divisão do momento do quarto grau centrado na média pela variância ao quadrado.

Portanto, o quarto momento $E[x^4]$ de uma *fdp* de x é dado por:

$$E[x^4] = \int_{-\infty}^{+\infty} p_x(x)x^4 dx \quad (2.13)$$

Se x tem média zero, significa uma versão normalizada de $E[x^4]$ que é conhecido como *kurtosis* (COMON, 1994). A *kurtosis* é definida em termos de uma relação com o quarto e o segundo momento central dado por:

$$K = \frac{E[x^4]}{E[x^2]^2} - 3 \quad (2.14)$$

A Figura 6 ilustra uma distribuição conjunta de duas variáveis aleatórias gaussianas independentes, x e y . A *kurtosis* dessas variáveis aleatórias são: $K_x = 0,0672$ e $K_y = 0,0248$. Na Figura 6, verifica-se que a *kurtosis* para uma variável aleatória gaussiana tende a zero (HYVÄRINEN, 1999a). Na verdade, este resultado é muito importante, pois tem relação com o teorema central do limite que será descrito posteriormente. Desta forma, a *kurtosis* pode ser utilizada para verificar a semelhança entre determinada distribuição e uma distribuição normal.

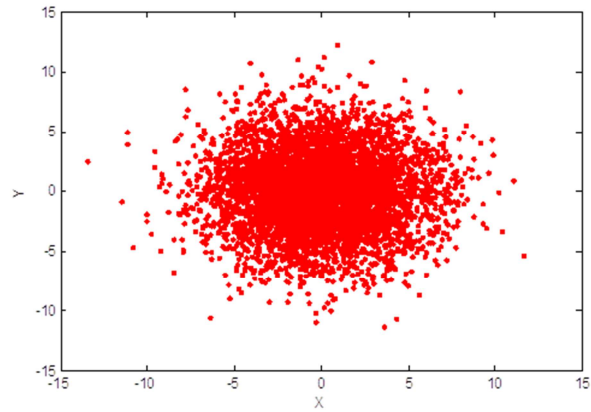


Figura 6: Distribuição conjunta de duas variáveis aleatórias gaussianas.

Distribuições cuja *kurtosis* é positiva são chamadas de supergaussianas ou *leptokúrticas* e quando a *kurtosis* é negativa são chamadas de subgaussianas ou *platokúrticas*. A Figura 08 ilustra o histograma de duas distribuições em que a *kurtosis* é negativa.

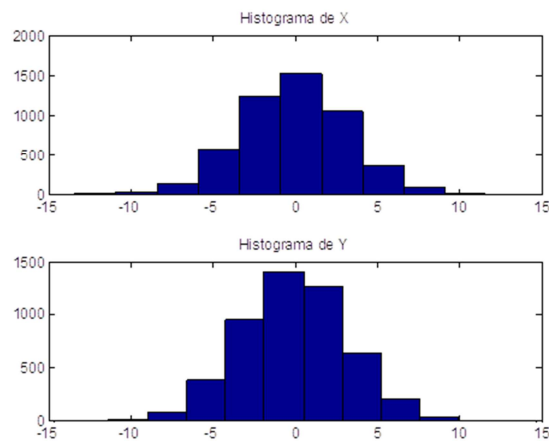


Figura 7: Gráfico superior é o histograma da VA x e inferior é da VA y .

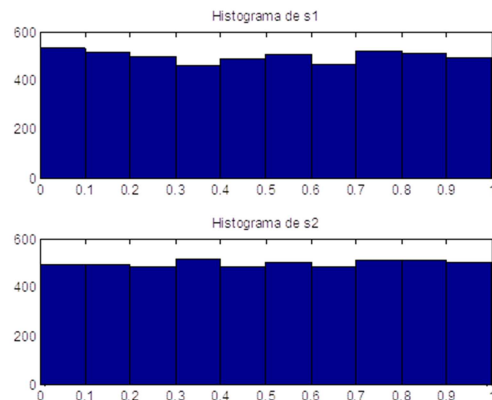


Figura 08: Distribuições *platokúrticas*, *kurtosis* de $s_1 = -1,1877$ e $s_2 = -1,2112$.

2.3.8 Correlação

A correlação é uma medida de relação linear entre duas variáveis aleatórias, ou seja, existem situações em que interessa estudar o comportamento no conjunto de duas variáveis que podem se comportar de formas distintas. Quando a função de densidade de probabilidade é desconhecida podemos usar o momento para caracterizar uma determinada distribuição. Neste contexto, podemos fazer às estimativas dos valores esperados dadas as únicas informações disponíveis, as misturas.

Outro conjunto de momento é chamado de correlação, que é dado pelo momento de segunda ordem. Podemos expressar a correlação de forma matricial, dada pela matriz de correlação, onde t representa a transposta da matriz ou vetor (PAPOULIS, 1991),

$$R_{xx}E\{xx^t\} \quad (2.15)$$

onde x representa um vetor de variáveis aleatórias.

Outra definição importante é a autocovariância, que é definido pela matriz de autocovariância. Portanto, o momento central correspondente à matriz de autocorrelação é denominado de autocovariância, que pode ser definido por:

$$C_{xx} = E\{(x - m_x)(x - m_x)^t\} \quad (2.16)$$

onde m_x é o vetor média. Caso este vetor for de zeros, a matriz de autocorrelação e autocovariância são as mesmas. Vamos passar este conceito para o caso em que temos uma densidade conjunta, onde obtemos:

$$R_{xy}E\{xy^t\} \quad (2.17)$$

onde x e y são duas VA. A expressão anterior da matriz de correlação cruzada e a matriz de covariância cruzada de x e y é representada por:

$$C_{xy} = E\{(x - m_x)(y - m_y)^t\} \quad (2.18)$$

2.3.9 Análise de Componentes Principais

PCA é uma técnica de análise de dados muito utilizado em aplicações de compressão de dados e extração de características (HYVÄRINEN, 2001).

Uma vez que se deseja obter componentes independentes y em particular as misturas são correlacionadas. Uma técnica eficiente para obter variáveis não correlacionadas através de uma transformação linear é a PCA, conhecida como transformada *de Karhunen-Loève* (KLT) (JAIN, 1989).

Anteriormente viu-se que ICA utiliza a independência estatística como medida de redundância nas misturas, bem como mostrou-se na seção anterior o significado da correlação entre duas VA, além do mais precisamos separar momentos para estudar independência. Em linhas gerais, a PCA utiliza a correlação para realizar essa tarefa.

No entanto, a correlação é considerada uma medida “fraca”. Todavia, pode-se verificar que este procedimento permite a determinação de uma transformação linear sobre a mistura. À luz deste paradigma, conclui-se que a PCA considera apenas estatística de segunda ordem, diferentemente de ICA, que considera estatística de ordem superior. Assim, utiliza-se a PCA como um pré-processamento ao ICA que chamamos de branqueamento (POBLADOR; MORENO et. al., 2004). Este método será detalhado na próxima seção.

Para ilustrar a tentativa de recuperar as fontes usando branqueamento, representamos duas fontes independentes, s_1 e s_2 , uniformemente distribuídas em um quadrado (Figura 9). Usou-se uma matriz quadrada 2×2 para gerar as misturas dadas pela Equação 2.2. O resultado desta mistura é apresentado na Figura 10 e por fim suas estimativas obtidas pelo processo de branqueamento são ilustradas na Figura 11.

Considerando o efeito do sistema misturador linear, verifica-se a dificuldade da recuperação das fontes (Figura 11) usando estatística de segunda ordem. Claramente, o método consegue recuperar as escalas das fontes, mas é incapaz de recuperar a rotação, pois existe uma indeterminação referente a uma matriz ortogonal cujo efeito é a rotação dos dados (HYVÄRINEN; OJA, 1999).

Intuitivamente, percebe-se que a utilização dessa ideia para um pré-processamento no algoritmo ICA dito anteriormente é obrigatório na utilização do branqueamento para a separação das fontes, visto que em distribuições gaussianas conhecemos apenas duas características, a média e variância, ou seja, este problema vai além da estatística de segunda ordem. Percebe-se com este resultado a ineficácia da

estatística de segunda ordem, no que tange a impossibilidade de recuperar fontes gaussianas e este resultado foi provado por *Comon* (1994) e será descrito na seção 2.4.

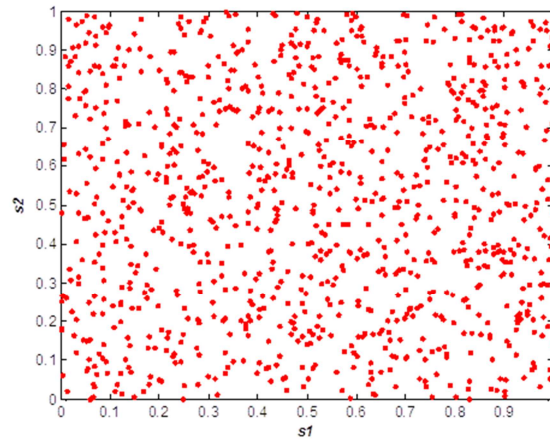


Figura 9: Distribuição conjunta é uniforme em um quadrado.

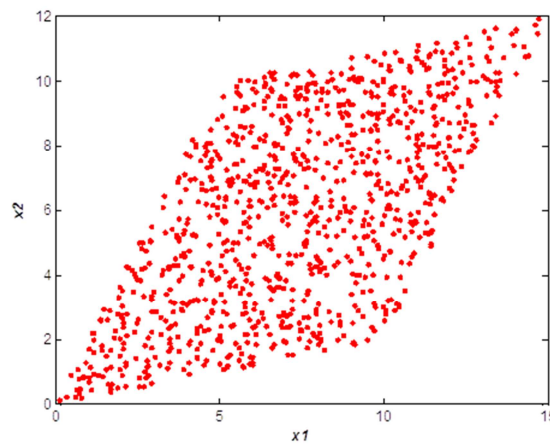


Figura 10: Distribuição das misturas dos componentes.

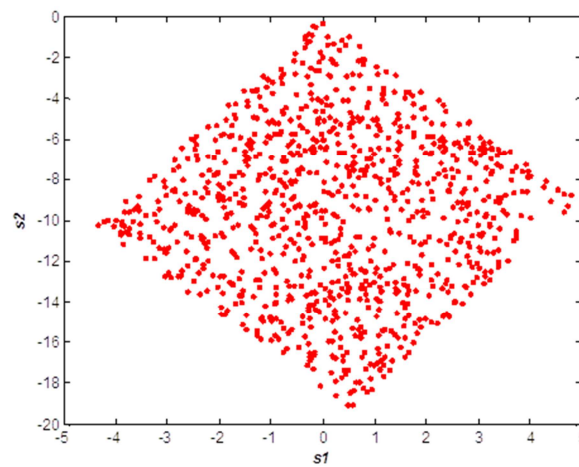


Figura 11: Distribuições conjuntas das estimativas usando PCA.

2.3.10 Branqueamento

A seção 2.3.9 ilustra a tentativa de recuperar duas fontes estatisticamente independentes a partir de uma mistura linear usando branqueamento. Isto nos mostra que devemos encontrar uma forma de rotacionar os dados na Figura 11 e consequentemente estimar as fontes.

A maioria dos algoritmos ICA encontrado na literatura utiliza o branqueamento para tornar as estimações das fontes mais simplificadas. A seguir, descreve-se este assunto, mas primeiramente vamos rever outro conceito de estatística.

Relembrando o conceito de independência, se duas VA são independentes então elas são descorrelacionadas, mas o inverso não é verdade (COMON, 1994). Se duas variáveis aleatórias são descorrelacionadas e têm variância igual a 1 então elas são chamadas de brancas (POBLADOR; MORENO et. al., 2004), ou melhor, a matriz de covariância é igual à identidade conforme a equação a seguir:

$$E\{zz^t\} = I \quad (2.21)$$

Podemos obter variáveis brancas a partir de uma transformação linear qual seja:

$$z = Vx \quad (2.22)$$

O branqueamento de x pode ser feito pela matriz V tal que,

$$V = ED^{-1/2}(E)^t, \quad (2.23)$$

onde, E é uma matriz ortogonal cujas colunas são os autovetores de $E\{xx^t\}$ e D é uma matriz diagonal com os autovalores correspondentes:

$$E\{xx^t\} = EDE^t \quad (2.24)$$

Assim, o branqueamento transforma a matriz A em uma nova matriz \hat{A} , de forma que agora temos:

$$z = VAs = \hat{A}s \quad (2.25)$$

Vamos agora fazer uma transformação ortogonal em \mathbf{z} fazendo:

$$\mathbf{y} = U\mathbf{z} \quad (2.26)$$

Em razão à ortogonalidade de U , \mathbf{y} também é branco como mostra a seguinte expressão:

$$E\{\mathbf{y}\mathbf{y}^t\} = E\{U\mathbf{z}\mathbf{z}^tU^t\} = UIU^t = I \quad (2.27)$$

Portanto, mostrou-se que o branqueamento é um pré-processamento para o ICA, pois ele não é suficiente para que se estimem as fontes independentes, e fornecendo apenas uma transformação ortogonal em \mathbf{s} . O que precisamos agora é de uma estratégia elaborada para rotacionar os dados das misturas.

2.3.11 Distribuições gaussianas

Nesta seção, apresenta-se uma revisão da distribuição de probabilidade bastante utilizada em estatística chamada distribuição normal ou gaussiana. A distribuição gaussiana aparece em muitos contextos diferentes e pode ser motivado a partir de uma variedade de perspectivas diferentes. Para uma VA real, por exemplo, a distribuição que maximiza a entropia é a gaussiana. O fato desta distribuição ser importante para nosso estudo está na relação com o teorema central do limite (PAPOULIS, 1991). O teorema central do limite garante que a soma de VA independentes e identicamente distribuídas é outra VA que tende para uma distribuição gaussiana.

Alguns algoritmos ICA utilizam esse conceito com o intuito de maximizar a não-gaussianidade das misturas. Portanto, estamos interessados em maximizar a não-gaussianidade das fontes.

Uma distribuição gaussiana pode ser expressa por:

$$f(x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x - \mu)^2}{2\sigma^2}\right\}, \quad (2.28)$$

onde $-\infty < x < \infty$, $-\infty < \mu < \infty$ e $\sigma^2 > 0$. A distribuição normal é determinada pelos seus parâmetros (μ e σ^2), que também são respectivamente o valor esperado e variância de uma VA normal. Para ilustrar a distribuição normal, a Figura 12 consiste na representação de três funções com distribuição normal. Observa-se que com o aumento da variância a altura da função densidade de probabilidade da média diminui.

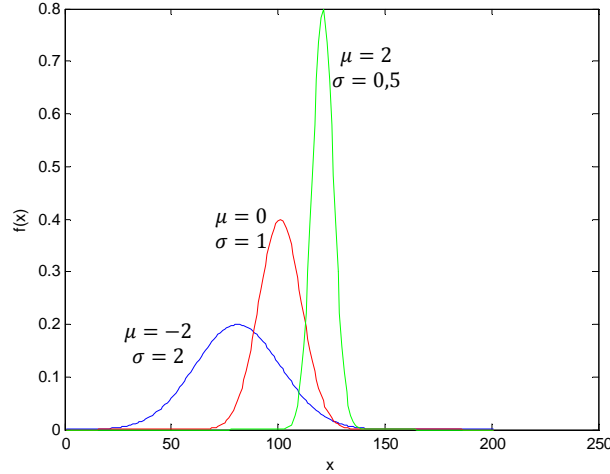


Figura 12: Exemplo de funções de densidade de probabilidade para variáveis aleatórias normalmente distribuídas.

Uma distribuição gaussiana pode ser multivariada e sua função de densidade de probabilidade para várias dimensões é dada por:

$$f(x, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)' \Sigma^{-1} (x - \mu) \right\} \quad (2.29)$$

onde x é um vetor coluna, μ é a média e Σ é matriz de covariância. A notação $|\cdot|$ denota a determinante de uma matriz e d é a dimensão dos dados. A Figura 13 mostra um gráfico com distribuição gaussiana bivariada centrada em zero e covariância igual à matriz de identidade (PAPOULIS, 1991).

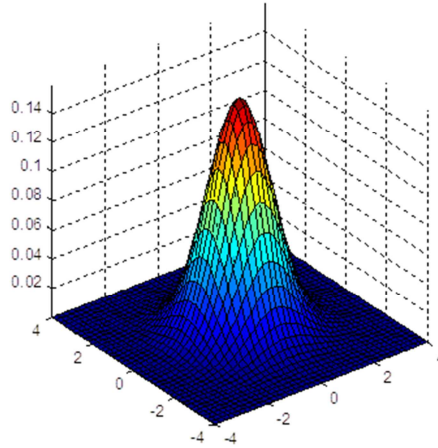


Figura 13: Distribuição normal bivariada.

2.3.12 Misturas gaussianas

Misturas gaussianas é uma ferramenta poderosa para análise e modelagem dos dados estatísticos. Contudo, a seleção do modelo, ou seja, a seleção do número de gaussianas na mistura para um conjunto de amostras ainda é uma tarefa difícil.

O modelo de mistura pode ser chamado de “semi-paramétrico”, onde este tipo de mistura foi amplamente utilizado em várias áreas da engenharia como, por exemplo, o modelo oculto de *Markov* para o reconhecimento da fala.

Embora termos diversos métodos na literatura, geralmente a forma mais tradicional de estimar os parâmetros de misturas gaussianas para uma VA x de várias dimensões é assumido como:

$$f(x) = \sum_{k=1}^N c_k G(x - \mu_k, \Sigma_k) \quad (2.30)$$

onde N é o número de fontes de misturas, c_k são os coeficientes das misturas não negativos e sua soma é igual a 1, ou seja, $\sum_{k=1}^N c_k = 1$. μ_k e Σ_k são a média e a matriz de covariância para cada fonte gaussiana, respectivamente. Podemos notar a semelhança da Equação 2.30 com o método de estimativa de *kernel* gaussiano, que será descrito posteriormente. Na verdade, o método de estimação de *kernel* gaussiano é o caso extremo do modelo de mistura gaussiana, onde todas as médias dos dados e todos os

coeficientes das misturas e também da matriz de covariância são iguais. Posteriormente, descreve-se sobre mapeamento de entrada e saída de uma rede neural RBF Gaussiana, em que esta é muito semelhante à técnica estatística Mistura de Modelos (*Mixture Models*).

2.3.13 Entropia

Por volta do ano de 1928, Harley fundamentou que quando um símbolo é escolhido a partir de um conjunto finito de possíveis símbolos S então o número de escolha pode ser considerado como uma medida de informação, a qual ele chamou de “quantidade de informações” (HARLEY, 1928). Segundo Harley, a quantidade de informações em um conjunto de n símbolos é proporcional ao número de diferentes escolhas e é dada por:

$$H_0 = \log_{10} S^n = n \log_{10} S \quad (2.31)$$

Notadamente, preferimos utilizar dois símbolos $[0,1]$ e o logaritmo usualmente utilizado é de base 2. Mas como estamos interessados na informação transmitida pela VA sobre um conjunto de mensagem, *Shannon* definiu a incerteza de um conjunto de variáveis aleatórias dada pela seguinte expressão:

$$H(x) = -\sum_i p_i \log_2(p_i) \quad (2.32)$$

onde os p_i é a probabilidade de obter x .

O conceito de entropia é bastante utilizado, pois há necessidade de extrair informações de VA desconhecidas. A entropia é uma medida da incerteza de uma VA (PAPOULIS, 1991). Para ilustrar este conceito, vamos supor uma moeda fictícia, onde a probabilidade de se obter cara é jogando uma moeda de forma aleatória, seja p , e a probabilidade de se obter coroa é de $1 - p$. Podemos modelar a entropia desta VA pela seguinte expressão matemática:

$$H = -p \log(p) - (1 - p) \log(1 - p) \quad (2.33)$$

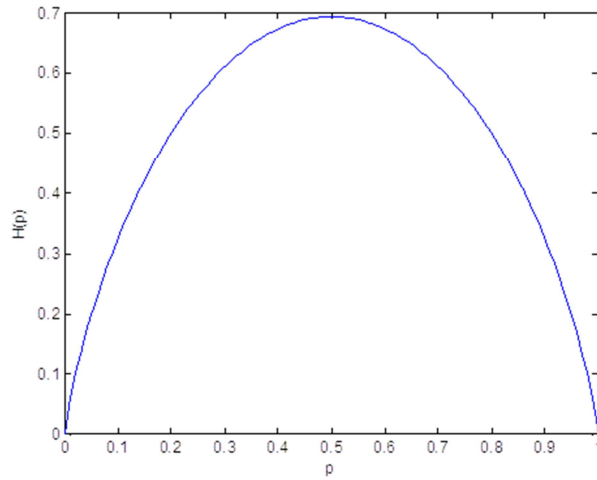


Figura 14: Gráfico de $H(p)$ em função de p .

A Figura 14 representa o gráfico de $H(p)$ em função de p . O máximo deste gráfico ocorre em $p = 0,5$ (probabilidade de obter cara ou coroa). O resultado nos diz que temos máxima entropia neste valor, seja $p = 0,3$ a probabilidade de obter o resultado igual a cara e $p = 0,7$ a probabilidade de obter coroa. Neste caso, têm-se mais informações sobre a moeda em relação à probabilidade $p = 0,5$ para cara e coroa.

Pressupõe-se agora que a probabilidade de obter cara seja $p = 0$ e coroa seja igual a $p = 1$. Neste limite, a entropia é zero e assim teremos máxima informação. Portanto, quanto menor for a probabilidade do acontecimento maior é a quantidade de informação, ou seja, máxima entropia. Este conceito de máxima entropia é bastante utilizado em estatística (MEDEIROS, 2005).

2.3.14 Entropia de Shannon

A ilustração do exemplo na seção anterior da moeda foi modelada pela entropia de *Shannon* para uma variável aleatória discreta x e é definida pela Equação 2.32. Esta definição pode ser generalizada para VA de valores contínuos, chamada de entropia diferencial. A entropia diferencial H com base na distribuição de probabilidade $p(x)$ de uma VA é definida por:

$$H(x) = - \int p(x) \log(p(x)) dx \quad (2.34)$$

2.3.15 Entropia de Rényi

As definições da entropia foram estendidas em inúmeras direções. Assim, a entropia de *Rényi* é parametrizada por uma constante α , onde a probabilidade p_i de ocorrência dos valores x_i da variável x é definida no caso discreto como:

$$H_{\alpha}(x) = \frac{1}{1-\alpha} \log \left(\sum_i p_i^{\alpha} \right) \quad (2.35)$$

Para o caso contínuo, basta substituir o somatório pela integral e p_i passa a ser a densidade de probabilidade $p(x)$. A Figura 15 ilustra o exemplo na seção 2.3.13 utilizando uma moeda fictícia. Entretanto, acrescentamos a entropia de *Rényi* em função da probabilidade de ocorrência de uma das faces da moeda. Para este exemplo utilizamos $\alpha = 2$.

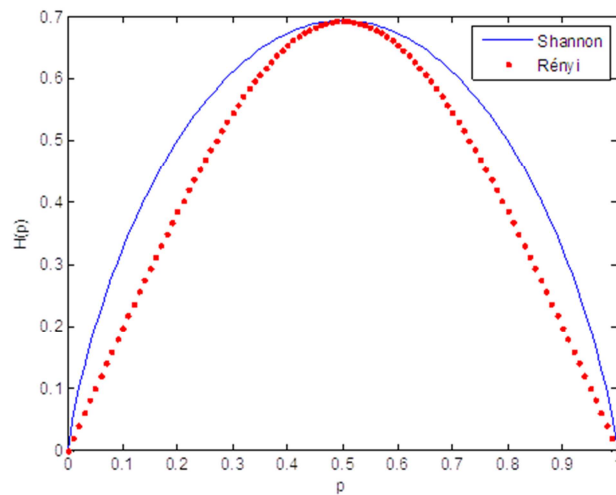


Figura 15: Gráfico $H_2(p)$ e $H(p)$ em função de p .

Para o caso de $\alpha = 1$, a entropia de *Rényi* torna-se igual à entropia de *Shannon* (ZYCZKOWSKI; RÉNYI, 2003).

No que tange a utilização da entropia *Shannon*, há os algoritmos de aprendizado competitivo para modelagem de misturas gaussianas. Este se mostrou como um bom método. Entretanto, nos trabalhos de *Jianwei* e *Jinwen* mostrou-se pelas experiências de simulações que usando entropia de *Rényi* converge muito mais rápido que os de *Shannon* (JIANWEI; JINWEN, 2008). Esse resultado é muito importante e serve de subsídios e incentivos para utilizar a entropia de *Rényi* na presente dissertação.

2.3.16 Negentropy

Visto que o objetivo do trabalho é a utilização do conceito de *Negentropy* como medida de independência nas misturas desconhecidas, esta seção é dedicada a este assunto.

A *Negentropy* é uma medida baseada na entropia diferencial (medida quantitativa), conhecida como medida de não-gaussianidade, ou seja, o quanto a distribuição de uma VA não se parece com uma gaussiana. Esta medida é bastante empregada em métodos que utilizam Análise de Componentes Principais (HYVÄRINEN, 2001).

A *Negentropy* é definida como (COMON, 1994):

$$J(x) = H(x_g) - H(x), \quad (2.36)$$

onde $H(x)$ é a entropia de x e $H(x_g)$ é uma variável aleatória gaussiana com mesma média e matriz de covariância da variável aleatória x . Para estimar o valor da *Negentropy*, há necessidade do conhecimento ou estimativa da *fdp*. Existem na literatura várias maneiras de se estimar a *Negentropy* de uma VA (HYVÄRINEN, 1999a). Podemos utilizar momento de alta ordem como *kurtosis* e *skewness* para obtermos uma estimativa. Um dos métodos para estimar $J(x)$ é a expansão em densidade polinomial. Ou podemos usar a expansão *Edgeworth* ou a expansão de *Gram-Charlier* (WALLACE, 1958). O método clássico para estimação da *Negentropy* é representado pela seguinte equação:

$$J(x) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} K(y)^2 \quad (2.37)$$

onde y tem média zero e variância unitária. Entretanto, este método usa *kurtosis* e sofre os mesmos problemas que a estimação por *kurtosis*, pois ela depende de uma boa aproximação das cumulantes.

Outra abordagem seria substituir o momento polinomial y^3 e y^4 por uma função G , propondo, desta forma, uma aproximação baseado em expectância E , conforme a equação a seguir (HYVÄRINEN; KARHUNEN; OJA, 2001):

$$J(x) \approx \sum_{i=1}^n k_i [E(G_i(x)) - E(G_i(x_g))]^2 \quad (2.38)$$

onde k são constantes positivas, x_g é uma variável aleatória gaussiana com média zero e variância unitária e G são funções não-quadráticas. Geralmente, escolhe-se uma função G de forma que ela não cresça muito rapidamente. A seguir, têm-se vários exemplos de função G .

$$G_1(y) = \frac{1}{a_1} \log(\cosh(a_1 y))$$

$$G_2(y) = \exp\left(-\frac{y^2}{2}\right)$$

$$G_3(y) = \frac{y^4}{4}$$

onde $1 \leq a_1 \leq 2$. Assim, podemos obter um estimador robusto e computacionalmente simples.

A *Negentropy* será sempre não negativa e uma tendência a zero se estiver uma distribuição gaussiana.

A maioria dos algoritmos de separação de sinais usando *Negentropy* requer que as misturas sejam branqueadas antes do processo de separação (discutido anteriormente). Nas maiorias das aplicações de algoritmos com *Negentropy* obtiveram-se bons resultados usando fontes supergaussianas e subgaussianas.

Conclui-se que a maximização da *Negentropy* é a mesma coisa que maximizar a não-gaussianidade das fontes. Todavia, a maximização das fontes seria buscar a minimização das entropias marginais das estimativas das fontes.

2.4 DEFINIÇÃO DE ICA

A contribuição de *Comon* (1994) foi responsável pela definição matemática da ICA. A ICA de um vetor aleatório $\mathbf{x} = [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_N]^T$ consiste na determinação de uma transformação linear

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \quad (2.39)$$

de maneira que os elementos das variáveis aleatórias $\mathbf{y} = [\mathbf{y}_1 \mathbf{y}_2 \cdots \mathbf{y}_N]^T$ minimizem uma função custo $\psi(\mathbf{y})$ chamada de função contraste, que expressa uma medida de

independência. A função contraste realiza um mapeamento no conjunto de fdp , que é associada aos elementos de \mathbf{y} e a um conjunto de números reais satisfazendo as seguintes propriedades:

- Um contraste deve ser invariante às permutações dos elementos de \mathbf{y} :

$$\psi(\mathbf{y}) = \psi(P\mathbf{y}),$$

para qualquer matriz de permutação P .

- Um contraste deve ser invariante à escala:

$$\psi(\mathbf{y}) = \psi(\Lambda\mathbf{y}),$$

para uma matriz diagonal Λ .

- Quando os elementos de \mathbf{y} forem independentes entre si, deve valer:

$$\psi(\mathbf{y}) \geq \psi(A\mathbf{y})^4,$$

para qualquer matriz inversível A .

A meta de separação linear da ICA é obter uma matriz de separação \mathbf{W} , tal que a Equação 2.39 seja uma boa estimativa dos sinais de fontes.

A Equação 2.39 é equivalente a Equação 2.2. Assim, o seguinte teorema provado por *Comon* (1994) descreve claramente a condição necessária para que um dado sistema misturador linear (a_{ij}) seja separável. O modelo representado pela Equação 2.2 é separável se, e somente se, a matriz \mathbf{A} possuir posto completo e no máximo umas das componentes independentes forem gaussianas (HYVÄRINEN, 2001). Isto faz sentido, pois se tentássemos estimar os dados da Figura 6 não teríamos resultados satisfatórios nas estimações das componentes, visto que as duas distribuições não possuem informação sobre as direções das colunas dessa matriz. A seguir, a Figura 16 ilustra o que é necessário para estimar as componentes independentes.

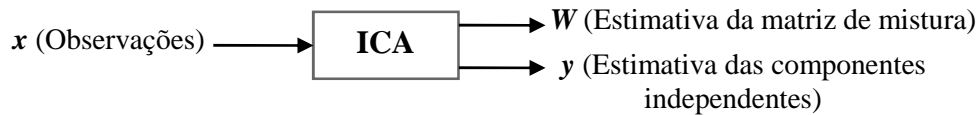


Figura 16: Esquema de estimativa das componentes independentes.

2.4.1 As ambiguidades ao modelo

Observou-se que o modelo de mistura é dito separável se tivermos $\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{\Lambda}\mathbf{P}\mathbf{s}$, onde $\mathbf{\Lambda}$ e \mathbf{P} correspondem à matriz diagonal e permutação, respectivamente. No entanto, há duas ambiguidades associadas a este procedimento, no sentido que não podemos recuperar a ordem das fontes nem seus ganhos. Neste aspecto, temos:

- Escalamento: As energias das componentes independentes não podem ser estimadas efetuando a multiplicação das componentes independentes por uma constante qualquer e simultaneamente efetuar a divisão da respectiva coluna da matriz \mathbf{A} pela mesma constante não altera a mistura \mathbf{x} ;
- Permutação: A ordem das componentes não pode ser estimada, dado que a alteração da ordem dos termos da Equação 2.39 não altera nenhum valor da mistura.

Os aspectos que foram discutidos até o presente momento mostram que é possível resolver o problema de separação com base na independência estatística. A seguir, descreveremos um algoritmo ICA que tem sido superior a outros métodos e de simples implementação computacional.

2.5 SEPARAÇÃO DE MISTURAS LINEARES

Nesta seção, apresentamos duas estratégias existentes em BSS linear na situação em que o sistema misturador é linear, instantâneo e o número de fontes igual ao número de sensores.

Em essência, o desenvolvimento de uma técnica de BSS é dado pela definição da estrutura de um sistema separador. Em seguida, estabeleceremos um critério que guie a busca por um bom conjunto de parâmetros do sistema separador.

2.5.1. Algoritmo ICA utilizando PCA

Um dos algoritmos que foi desenvolvido recentemente provou ser superior a algumas abordagens ICA (KUN, CHAN, 2006). Conhecido como *P-ICA*, esta abordagem basicamente resolve o problema de BSS linear aplicando PCA e posteriormente usa uma transformação para recuperar os sinais de fontes.

Considerando os dados observados representados por \mathbf{x} , PCA e ICA visam à transformação linear dado pela Equação 2.39. No entanto, elas exploram os dados de formas diferentes. O PCA visa encontrar uma transformação ortogonal em \mathbf{W} que dá resultados não correlacionados (vale lembrar que se mostrou anteriormente que PCA considera apenas estatística de segunda ordem). Porém, o PCA utiliza a distribuição conjunta gaussiana para ajustar os dados e encontrar uma transformação ortogonal que faz a distribuição conjunta gaussiana fatorável independente da verdadeira distribuição dos dados. Neste contexto, a ICA tenta encontrar uma transformação linear que faz a verdadeira distribuição conjunta dos dados transformados fatorável, de modo que as saídas são mutuamente independentes.

Grande parte dos algoritmos ICA requerem o branqueamento das misturas como descrito na seção 2.3.10, podemos citar como exemplo o *FastICA* (HYVÄRIEN, 1999c) e o *JADE* (CARDOSO, 1999). Discorremos na mesma seção que o processo de branqueamento pode ser feito a partir de PCA, bem como usar decomposição de autovalores e autovetores.

Em linhas gerais, Kun e Chan (2006) mostrou e provou que as componentes independentes têm diferentes *kurtosis*, no qual se tem o seguinte teorema: Dado \mathbf{s} , \mathbf{v} , e \mathbf{z} vetores aleatórios tal que $\mathbf{v} = \mathbf{W}\mathbf{s}$, onde \mathbf{W} é uma matriz ortogonal e $\mathbf{z} = \|\mathbf{v}\| \mathbf{v}$. Suponha que \mathbf{s} tem média zero e que as componentes independentes têm diferentes *kurtosis*. Então, a matriz ortogonal \mathbf{U} que é dado pela componente principal de \mathbf{z} (\mathbf{z} não é centralizado) realiza ICA em \mathbf{v} .

O método *P-ICA* se resume nos seguintes procedimentos que consiste em quatro etapas:

1. Branqueamento de \mathbf{x} usando a Equação 2.22;
2. Faça uma transformação $\mathbf{z} = \|\mathbf{v}\| \mathbf{v}$;
3. Encontre \mathbf{U} usando PCA em \mathbf{z} ;
4. Depois de encontrar a matriz ortogonal \mathbf{U} finalmente a matriz de separação pode ser estimada usando a expressão $\mathbf{W} = \mathbf{U}\mathbf{V}$.

2.5.2 Separação de fontes usando Negentropia

O algoritmo *FastICA* foi primeiramente publicado por Hyvärinen (1999b). O objetivo do algoritmo é encontrar uma matriz \mathbf{W} e ajustar as suas linhas denotadas por w_i^t , de modo que $\mathbf{y}_i = w_i^t \mathbf{x}$ resulte numa estimativa das fontes, lembrando que a maximização da *Negentropia* é baseada nos momentos polinomiais (HYVÄRINEN, 1999a). Utilizando a aproximação da *Negentropia* e considerando que os dados foram branqueados, a maximização da *Negentropia* se resume em encontrar uma matriz \mathbf{W} que é descrito pelo seguinte problema de otimização (HYVÄRINEN, 2000):

$$\hat{w}_i = \arg \max_{w_i} \left(E\{G(y_i)\} - E\{G(x_g)\} \right)^2$$

Fazendo uma restrição na etapa de adaptação, temos que restringir a potência de cada uma das estimativas assumindo que:

$$E\{y_i\} = E\{w_i^t \mathbf{x}\} = 1$$

O máximo da função \hat{w}_i é quando encontramos certo valor ótimo de $E\{G(y_i)\}$ em razão ao $E\{G(x_g)\}$ ser constante. Assim, considerando o primeiro termo da equação, o problema de maximizar e otimizar são equivalentes. Podemos mostrar que o problema de otimização é resolvido usando o método de *Lagrange*, quando a seguinte condição é satisfatória (HYVÄRINEN, 1999):

$$E\{\mathbf{x}G'(w_i^t \mathbf{x})\} + \beta w_i = 0,$$

onde β é uma constante.

Considerando que as misturas estejam branqueadas, aplica-se o método de *Newton* para a solução da expressão anterior e assim se obtêm a seguinte regra de atualização:

$$w_i \leftarrow E\{\mathbf{x}G(w_i^t \mathbf{x})\} - E\{G'(w_i^t \mathbf{x})\}w_i$$

$$w_i \leftarrow \frac{w_i}{\|w_i\|}$$

onde G é uma função não quadrática e G' é sua derivada. A expressão a seguir poderia representar G e sua derivada, respectivamente:

$$G(u) = \frac{1}{a_1} \log \cosh(a_1 u)$$

$$G'(u) = \tanh(a_1 u)$$

Para recuperar várias fontes a partir da regra de atualização (ou regra de ajuste), se faz necessário executá-la para vários vetores w_i . Frequentemente, se tem um problema em que o algoritmo sempre encontra a mesma fonte, ou seja, converge para o mesmo ótimo. Neste caso, o problema é contornado da seguinte maneira: Considere um problema de separação de três fontes distintas feita a extração da primeira fonte. A extração da segunda fonte é feita aplicando a regra de ajuste. Entretanto, a cada iteração se retira do vetor em processo de estimação a contribuição do vetor referente à primeira fonte, de modo que esses dois vetores sejam ortogonais. Podemos usar esta mesma estratégia para extrair a terceira fonte. Deve-se, em cada iteração, retirar a contribuição dos dois vetores estimados e assim por diante. Finalmente, a regra de aprendizagem do *FastICA* é descrito:

1. Centralizar e branquear as misturas;
2. Definir aleatoriamente valores iniciais para w_i (colunas de \mathbf{W}) e ortogonalizar \mathbf{W} de acordo com passo 5;
3. Para todo i faça $w_i \leftarrow E\{\mathbf{x}G(w_i^t \mathbf{x})\} - E\{G'(w_i^t \mathbf{x})\}w_i$;
4. Divida w_i por sua norma;
5. Ortogonalizar $\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^t)^{-1/2}\mathbf{W}$;
6. Caso não convirja, volta para o passo 3.

2.6 SEPARAÇÃO DE MISTURAS NÃO LINEARES

Ao longo desta dissertação, mostrou-se BSS no paradigma de misturas lineares, ou seja, uma solução pode ser obtida de acordo com o modelo ICA se encontrarmos uma matriz inversa $\mathbf{W} = \mathbf{A}^{-1}$, tal que $\mathbf{s} = \mathbf{W}\mathbf{x}$.

Podemos estender esse conceito para o caso não linear. Uma formulação geral para o caso da Análise de Componentes Independentes não lineares (*nonlinear independent component analysis – NLICA*) pode ser descrito da seguinte forma:

Considere os sinais de misturas \mathbf{x} sendo formados por um modelo de mistura instantâneo não linear dado por:

$$\mathbf{x} = F(\mathbf{s}) \quad (2.40)$$

onde $F(\cdot)$ é um mapeamento não linear em que o número de fontes é igual ao número de misturas. Naturalmente, no *NLBSS (Nonlinear Blind Source Separation)* a solução do problema de separação cega de fontes não linear requer o ajuste de um sistema separador que também seja não linear.

A proposta é estimar G (sistema separador), tal que:

$$\mathbf{y} = G(\mathbf{x}) \quad (2.41)$$

sabendo que as componentes \mathbf{y} são estatisticamente independentes. Portanto, se $G = F^{-1}$ as fontes são perfeitamente recuperadas e finalmente \mathbf{s} será igual a \mathbf{y} (JUTTEN; KARHUNEN, 2003).

Viu-se que para o caso linear há vários métodos para encontrar o sistema separador e um desses métodos é o *P-ICA* e o *FastICA*. Em especial no caso não linear, esta tarefa é agora um pouco mais difícil, pois nem sempre é possível encontrar um sistema separador que seja capaz de inverter a ação do sistema de mistura.

Uma característica do problema de *NLICA* é que as soluções são difíceis (JUTTEN; KARHUNEN, 2003). Se \mathbf{x} e \mathbf{y} são VA independentes é fácil observar que $F(\mathbf{x})$ e $G(\mathbf{y})$ onde $F(\cdot)$ e $G(\cdot)$ são funções diferenciáveis e também independentes. No *NLICA* há um número infinito de soluções para o mapeamento inverso G em uma determinada aplicação, de forma que temos que aplicar algumas restrições ao problema em questão.

Geralmente, o número de parâmetros a ser estimado em um modelo não linear aumenta quando comparado com o modelo linear. Em vista disso, os algoritmos *NLICA* apresentam maior complexidade computacional e de convergência.

Entre os algoritmos *NLICA* propostos na literatura, há várias classes de métodos que impõe restrições ao modelo de mistura, garantindo que a estimativa das

componentes independentes não lineares sejam iguais às fontes. Porém, muitas vezes para o caso não linear, independência estatística não é suficiente para garantir a independência das fontes, ao passo que o mapeamento não linear que preserva a independência é chamado de triviais (JUTTEN; KARHUNEN, 2003). Alguns exemplos de mapeamentos triviais podem ser encontrados em (JUTTEN; ZADEH; HOSSEINI, 2004).

2.6.1 Modelo de separação PNL

O modelo PNL (*Post-nonlinear*) (TALEB; JUTTEN, 1999) assume que os sinais observados são gerados por misturas lineares seguidas por componentes não lineares. Os sinais observados podem ser expressos como:

$$\mathbf{x}_i = f_i(\mathbf{A}\mathbf{s}) \quad (2.42)$$

onde o mapeamento completo não linear é $F(\cdot) = [f_1(\cdot), f_2(\cdot), \dots, f_n(\cdot)]^t$. Apesar deste modelo ser restrito ele pode ser aplicado em vários problemas práticos, especialmente quando os sinais de fontes se propagam através de um canal linear e estão presentes na não linearidade no conjunto de sensores.

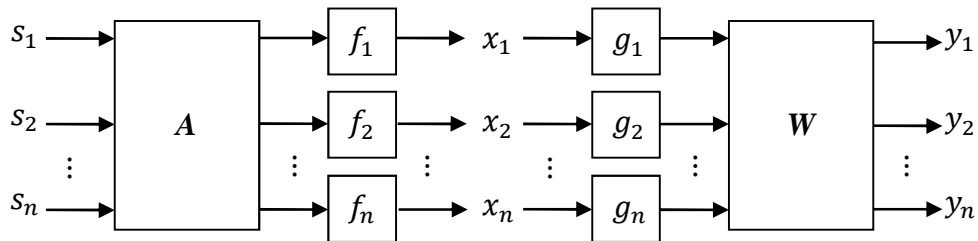


Figura 17: Mistura PNL e modelo de separação.

Como ilustrado na Figura 17, a recuperação dos sinais de fontes é feita por um modelo que compreende a inversa não linear e um estágio linear (Matriz W). A seção não linear (g_i) é geralmente estimada utilizando arquitetura de redes neurais como a *Multi-Layer-Perceptrons* (MLP) ou *Radial Base Function* (RBF) (TALEB; JUTTEN, 1999).

Diferentes medidas de independência estatísticas são utilizadas para o treinamento. A parte linear pode ser executada por qualquer algoritmo ICA linear. As fontes estimadas são calculadas através de:

$$\mathbf{y}_i(n) = \sum_{j=1}^k w_{ij} g_j(\mathbf{x}_j(n)) \quad (2.43)$$

O algoritmo que foi desenvolvido por *Taleb e Jutten* (1999) e é uma das primeiras tentativas propostas na literatura sobre o PNL. O algoritmo executa a estimação iterativa da estatística do sinal recuperado.

Em diferentes abordagens, a função inversa g_j é aproximada por uma função de transferência sigmoideal e muitas vezes pela falta do conhecimento da mistura se utiliza uma função de transferência mais flexível com base em polinômios de p -ésima ordem, tal como:

$$g_j(\mathbf{x}_j) = \sum_{l=1}^p g_{jl} \mathbf{x}_j^{2l-1} \quad (2.44)$$

Onde $g_j = [g_{j1}, \dots, g_{jp}]$ é o vetor de parâmetros a ser determinado. Desta forma, os sinais de fontes são calculados como:

$$\mathbf{y}_i = \sum_{j=1}^k w_{ij} \sum_{l=1}^p g_{jl} \mathbf{x}_j^{2l-1} \quad (2.45)$$

Como existem muitos parâmetros a serem ajustados no modelo inverso e o problema de otimização envolve funções não lineares algumas vezes os algoritmos vão para mínimos locais (JUTTEN; KARHUNEN, 2003).

Diferentes procedimentos foram propostos na literatura a fim de melhorar o desempenho de treinamento da rede neural do modelo de mistura PNL. *Tan* (2001) e *Kai* (2006) utilizaram algoritmo genético para realizar uma busca global para encontrar o melhor conjunto de parâmetros que maximizem uma medida de independência. Em (ROJAS; PUNTONET, 2004) foram testados e comparados diferentes algoritmos de

otimização global com o *Competitive Learning* e *Simulated Annealing* para encontrar uma boa solução de separação de fontes com mistura PNL.

No contexto de arquiteturas de redes neurais em (TAN; WANG, 2000) foram aplicados com sucesso ao problema de separação de fontes PNL. No trabalho de (SOLAZZI; PIAZZA; UNCINI, 2001) foi proposto um algoritmo de separação usando *adaptive spline neural networks*.

Para ilustrar o método PNL, a Figura 18 representa dois sinais de fontes e a Figura 19 mostra a distribuição conjunta das fontes e da mistura.

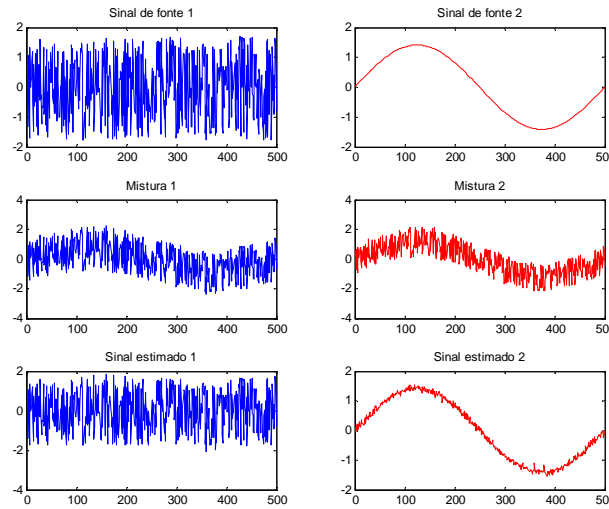


Figura 18: Demonstração de estimação de fontes usando abordagem PNL.

Neste exemplo, são utilizados dois sinais de fontes com 500 amostras. A função não linear é representada por uma função tangente hiperbólica dada por $f(x) = \tan(1,5x)$ e como medida de independência usou-se a minimização da informação mútua. O algoritmo para estimação da parte linear (Matriz W) foi o *P-ICA*.

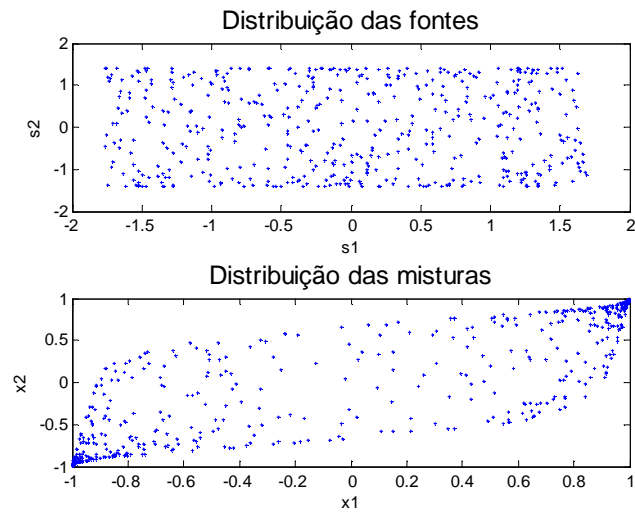


Figura 19: Distribuição conjunta dos sinais de fontes e das misturas PNL.

3 ALGORITMOS GENÉTICOS

A teoria dos Algoritmos Genéticos (AG) foi desenvolvida por *John Holland* (1975). Ela é um método dinâmico (heurísticos) de busca baseado no mecanismo de seleção e evolução natural (SANKAR, 2007). O objetivo principal desta técnica de inteligência artificial é encontrar o indivíduo ótimo de uma população. Um AG fornece estratégias de busca que podem ser usadas em problemas de otimização ou classificação (SANKAR; EIBEN, 2007), bem como usa modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas.

Um dos problemas encontrados na literatura é a otimização (ou maximização), que geralmente apresenta um espaço de busca e uma função objetivo.

Matematicamente, podemos descrever um problema básico de maximização da seguinte forma: encontrar o ponto máximo da função $f(x) = x \sin(10\pi x) + 1$ dentro de um intervalo de busca $-1 \leq x \leq 2$. Verifica-se na Figura 20 que o problema aparentemente é simples, entretanto existem vários máximos (máximos locais) que torna o problema agora não trivial. Alguns métodos como o método do gradiente tem dificuldade de localizar o ponto máximo dessa função. O AG é uma boa ferramenta para a solução deste problema, onde o máximo global se encontra no 15º pico da função, ou seja, o seu valor é 2,85.

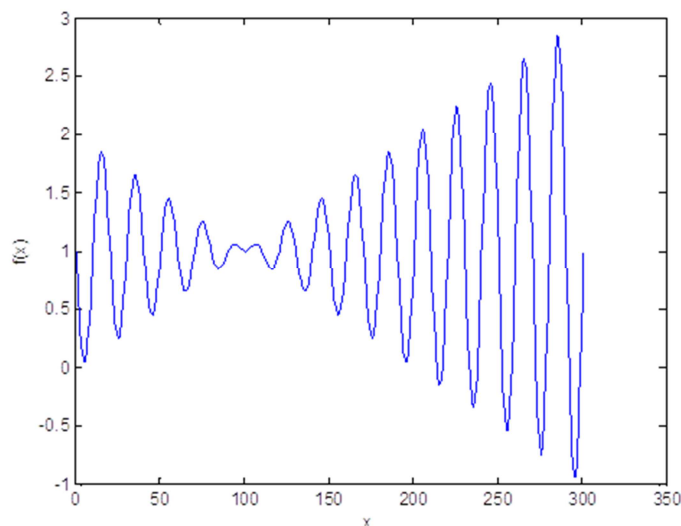


Figura 20: Gráfico da função $f(x) = x \sin(10\pi x) + 1$.

Para compreender melhor o AG, se faz necessário introduzir certos conceitos básicos do campo da biologia genética. Todo organismo é formado por uma ou mais

células e dentro de cada célula o organismo possui uma cópia completa do conjunto de um ou mais cromossomos (do grego *homologos* = igual, semelhante) que descrevem o organismo (conjunto chamado de genoma).

Um cromossomo consiste de genes (unidade básica da informação) que são blocos de sequências de DNA. Analogamente, podemos comparar o gene a determinada característica humana como, por exemplo, a cor dos olhos, que também possui 4 alelos. Portanto, neste exemplo chamamos de alelos as suas possíveis cores: preto, verde, azul ou castanho.

Um genótipo é um conjunto de genes que representam uma determinada característica como, por exemplos: cor dos olhos, cabelo, altura. O genótipo é hereditário, diferente do fenótipo (expressão das características físicas e mentais codificadas pelos genes), que é a manifestação do genótipo, ou seja, pode ser modificado pelo ambiente.

O processo do AG é o mecanismo de seleção e evolução natural, ou seja, temos que transmitir essas informações (seleções) para as gerações futuras. Esta transmissão é realizada através da reprodução existente na natureza que são: Assexuada e Sexuada. A sexuada exige a presença de dois organismos, na maioria das vezes de sexos opostos, para troca de material genético e a assexuada é tipicamente de organismos inferiores, como bactérias.

Neste momento já possuímos conceitos básicos de genética. Agora podemos entender melhor o funcionamento básico dos algoritmos genéticos através de um esquema básico.

3.1 ESQUEMA BÁSICO DE UM ALGORITMO GENÉTICO

O esquema básico apresentado nesta seção tem objetivo de introduzir uma visão global do funcionamento e principais etapas na implementação de um Algoritmo Genético utilizado neste trabalho.

A seguir, apresenta-se um resumo do algoritmo básico representado pelo pseudocódigo na Figura 21 que consiste em: inicialize uma população aleatoriamente, em seguida avalie a população com alguma função matemática, após esse procedimento utilize os operadores genéticos para selecionar os mais aptos para formar uma nova

população e finalmente avalie novamente a população até que algum critério de parada seja atendido.

```

t := 0;
inicializa_população P(0)
Enquanto não termina faça
    avalie_população P(t)
    P' := selecione_pais P(t)
    P' = recombinação_e_mutação P'
    avalie_população P'
    P(t + 1) = selecione_sobreviventes P(t), P'
    t := t + 1
Fim enquanto

```

Figura 21: Pseudocódigo simples do Algoritmo Genético.

A Figura 22 representa um fluxograma básico para implementação do algoritmo genético proposto neste trabalho. A seguir, é detalhado cada etapa do esquema básico do Algoritmo genético proposto nesta dissertação.

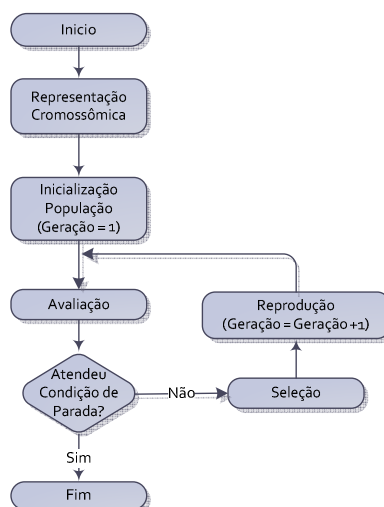


Figura 22: Esquema básico de um Algoritmo Genético.

3.2 REPRESENTAÇÃO CROMOSSÔMICA

A representação dos cromossomos é fundamental para traduzir de forma adequada a informação do nosso problema para o algoritmo computacional, a fim de obter maior qualidade dos resultados. Uma representação simples e bastante usada é a binária (AMARI; YANG et al., 1997).

Todavia, cada indivíduo é definido univocamente por um cromossomo. Assim, os termos indivíduo e cromossomo serão usados para expressar a mesma entidade. Um grupo de cromossomos é também conhecido como população (RANDY; WERNER, 2007). A seguir, é elaborado um exemplo simples de uma representação didática para que o leitor tenha um conceito básico da representação cromossomial.

Suponha-se que se deseja encontrar uma representação usando números binários (alfabeto binário). Para isso, utiliza-se os números decimais 1,00, 1,50 e 2,00 com uma precisão de 10^{-3} .

Para representar esses números temos que levar em consideração dois fatores: a faixa de operação de cada variável e a precisão desejada. Estes dois parâmetros definem, em conjunto, quantos bits por variável pode ser utilizado para representar nosso indivíduo que é dado pela seguinte expressão (LIDEN, 2006):

$$Comprimento = 1 + Inteiro \left\{ \frac{\log \left(1 + \frac{Fim - Inicio}{Precisão} \right)}{\log 2} \right\} \quad (3.1)$$

Substituído os valores na expressão temos:

$$Comprimento = 1 + Inteiro \left\{ \frac{\log \left(1 + \frac{2 - 1}{0,001} \right)}{\log 2} \right\}$$

$$Comprimento = 1 + Inteiro(9,6772)$$

$$Comprimento = 1 + 9 = 10 \text{ genes}$$

Na Equação 3.1 o comprimento é a quantidade de bits de cada indivíduo, Inteiro é uma função que retorna somente a parte inteira da expressão entre chaves, \log é o logaritmo neperiano. Na expressão anterior verifica-se que o comprimento de cada indivíduo são 10 genes, conforme representados na Figura 23. Neste exemplo, o indivíduo A corresponde ao menor valor (valor 1,0) e o indivíduo C corresponde ao maior valor (valor 2,0).

0	0	0	0	0	0	0	0	0	0	Indivíduo A
1	0	0	0	0	0	0	0	0	0	Indivíduo B
1	1	1	1	1	1	1	1	1	1	Indivíduo C

Figura 23: Representação de 03 cromossomos.

Outro tipo de codificação utilizada é a codificação real. A codificação binária possui algumas desvantagens quando aplicada no problema de alta dimensionalidade e precisão numérica (MICHALEWICZ, 1996).

Na codificação real, o cromossomo é representado semelhante à codificação binária, no qual cada posição do gene contém um valor real, conforme representada na Figura 24.

1,029	0,039	2,07	0,247	1,240	1,734	0,456
-------	-------	------	-------	-------	-------	-------

Figura 24: representação da codificação real.

Quando se utiliza valores binários é preciso codificá-los para valores reais dentro do seu espaço de busca. O cálculo para achar os valores decimais dos indivíduos é dado pela seguinte expressão:

$$x = \min(+(\max - \min) \frac{VDI}{VDS} \quad (3.2)$$

onde VDI é o valor decimal do individuo corrente (ou gene), VDS é o valor decimal correspondente ao individuo binário superior, \min é o valor decimal inferior e \max é o valor decimal superior da cadeia binária.

Exemplo: Considere um individuo com 3 genes e cada gene representa 3 bits. Temos um intervalo de busca de $[-10,10]$.

Representamos o indivíduo pela seguinte cadeia binária:

1 0 0 1 1 0 0 1 0

Gene 1

Vamos achar o valor de VDI pegando os primeiros genes (1 0 0) do individuo e convertemos para decimal.

$$VDI = 4.$$

Agora vamos achar o valor de VDS , resultando resulta na seguinte fórmula:

$$VDS = 2^L,$$

onde L é o número de bits. Assim:

$$VDS = 2^3 = 8$$

Fazemos o cálculo para o primeiro gene:

$$gene1 = -10 + (10 - (-10)) \frac{4}{8}$$

$$gene1 = 0$$

Gene 2

Pegamos o segundo gene (1 1 0) do individuo, temos:

$$VDI = 6$$

Já sabemos que o $VDS = 8$ então virá:

$$gene2 = -10 + (10 - (-10)) \frac{6}{8}$$

$$gene2 = 5$$

Gene 3

Pega-se o terceiro gene (0 1 0):

$$VDI = 2$$

$$gene3 = -10 + (10 - (-10)) \frac{2}{8}$$

$$gene3 = -5$$

Finalmente, o vetor [0 5 -5] representa os valores da cadeia binária convertida para números decimais.

3.3 INICIALIZAÇÃO DA POPULAÇÃO

O tamanho da população é extremamente sensível ao desempenho do algoritmo genético. Este parâmetro deve ser escolhido com bastante cuidado. Geralmente a primeira população deve ser inicializada de forma aleatória (LIDEN, 2006).

Para o nosso propósito, uma população é representada através de uma matriz com cada linha correspondente a um cromossomo, conforme a expressão a seguir.

$$população = \begin{bmatrix} crom_1 \\ crom_2 \\ \vdots \\ crom_N \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1M} \\ g_{21} & g_{22} & \cdots & g_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N1} & g_{N2} & \cdots & g_{NM} \end{bmatrix} \quad (3.3)$$

3.4 AVALIAÇÃO

Avaliação ou função de avaliação ou função custo é utilizada para determinar a qualidade de cada cromossomo ao problema em questão, ou seja, cada cromossomo de uma determinada população será avaliado de acordo com uma função matemática que traduz o seu comportamento (posteriormente mostra-se que essa função é representada por uma medida de independência).

3.5 SELEÇÃO

O método de seleção deve ser parecido com o processo biológico (seleção natural), em que apenas os membros mais saudáveis ou melhores da população são permitidos para sobreviver na próxima geração. Os indivíduos selecionados são chamados de pais. Existem três tipos de seleção: Determinísticas, Estocástica e Híbrida.

A Determinística atende a um critério desejado ou um critério previamente estabelecido. A estocástica quando os seus eventos possuem probabilidade de ocorrer em função do tempo. Híbrida quando parte do processo segue na forma determinística e outra parte estocástica.

Um dos processos de seleção mais utilizado é o método da roleta viciada, também conhecida do inglês como *roulette wheel*. Este processo é estocástico, probabilístico e variante de geração a geração e funciona similar a uma roleta de cassino, onde cada cromossomo recebe um pedaço proporcional a sua avaliação.

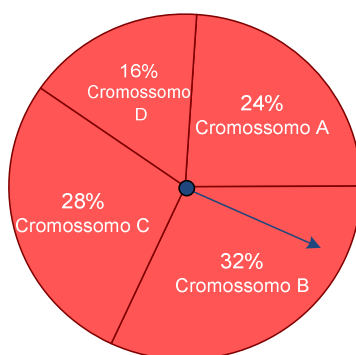


Figura 25: Roleta viciada para quatro cromossomos.

Na Figura 25 há uma representação didática de uma roleta viciada, sorteando quatro cromossomos para serem pais na próxima geração. Quanto maior a área maior é a probabilidade do correspondente cromossomo vir a ser escolhido como pai. Neste exemplo, o cromossomo B tem maior chance de ser escolhido, ao passo que o D tem a menor chance de escolhido.

3.6 REPRODUÇÃO

A reprodução é a fase onde são gerados os novos indivíduos com objetivo de completar a nova geração. As reproduções são implementadas através da ação dos operadores genéticos, onde os mais utilizados são: recombinação, mutação, inversão e rotação.

3.6.1 Recombinação

A recombinação, também conhecido do inglês como *crossover*, é um operador genético que envolve a participação de dois cromossomos, cujo objetivo principal consiste na troca de material genético entre os pais gerando dois filhos.

Uma maneira simples de implementar o *crossover* entre dois indivíduos é a escolha de um ponto de corte, que consiste numa posição entre dois genes de um cromossomo, ou seja, o ponto de corte é o ponto de separação entre cada um dos genes que compõe o material genético de cada pai (LIDEN, 2006).

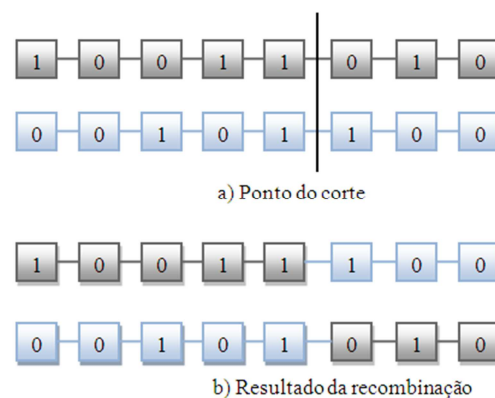


Figura 26: Processo de recombinação usando um ponto. a) Ponto de corte no cromossomo; b) resultado da recombinação.

O processo de recombinação usando um ponto é representado na Figura 26, onde este ponto constitui uma posição entre dois genes. Depois de sorteado o ponto de corte, separa-se os pais em duas partes: uma a esquerda e outra a direita do ponto de corte.

Existem outras maneiras de implementar a recombinação como, por exemplo dois pontos, uniforme. A seguir vamos descrever o processo de recombinação usando dois pontos.

O processo usando dois pontos de corte é similar a um ponto de corte, onde os dois pontos delimitarão a região de troca dos genes compreendidos entre estes pontos (recombinação interna). O processo de recombinação usando dois pontos pode ser observado na Figura 27. Ao processar a recombinação, geram-se dois filhos, onde apenas um destes sobreviverá.

Para escolher qual o filho que sobrevivera na próxima geração dar-se por critério de Sorteio (escolha feita pelo simples sorteio entre os dois candidatos) ou Elitista (a escolha é o candidato de maior adaptabilidade, caso seja empatado, usa-se novamente o critério de sorteio).

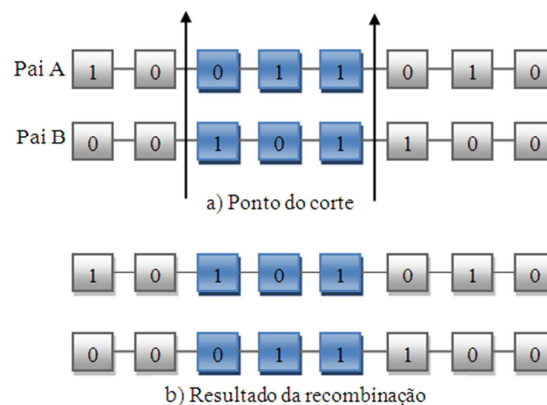


Figura 27: Processo de recombinação usando dois pontos. a) Ponto de corte no cromossomo, b) resultado da recombinação.

Da mesma forma que temos uma representação real para os cromossomos também temos uma representação real para o processo de *crossover*, onde todos os operadores de representação binária podem ser aplicados para representação real (OBITKO, 2010). A seguir, vamos discorrer sobre operador genético mutação.

3.6.2 Mutação

O operador de mutação é fundamental no algoritmo genético, pois garante a continuidade de diversidade genética na população. Este operador é monoparticipativo, o qual envolve a participação de apenas um indivíduo.

A técnica consiste na mudança da posição de um único gene do indivíduo pai, gerando um indivíduo filho na próxima geração. Na Figura 28 representa-se o operador de mutação em que é escolhido aleatoriamente um ponto que identificará a posição do gene que sofrerá a transformação.

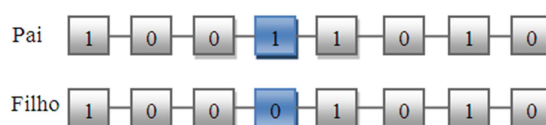


Figura 28: Processo de mutação em um cromossomo.

Na Figura 28 o gene correspondente na posição 4 da esquerda para direita sofreu uma mudança do alelo 1 para alelo 0. A taxa de mutação normalmente é pequena. Neste trabalho estamos utilizando um valor entre 0,5 e 0,01% da população. Caso o operador seja um valor alto demais, então o AG passará a ter um comportamento com um processo *random walk* e perderá algumas de suas propriedades (AMARI; YANG et al., 1997). Entretanto, se este mesmo valor for muito pequeno, a população poderá não ter a diversidade depois de certos números de gerações.

3.7 OUTROS PARÂMETROS

Há outro parâmetro na implementação do algoritmo genético, no qual vale destacar a adaptabilidade, o número de cópias esperadas e a adaptabilidade relativa.

A adaptabilidade indica quando um determinado indivíduo está adaptado aos aspectos modelados matematicamente pela função de avaliação. O número de cópias esperadas fornece a quantidade de cópias esperadas de um determinado cromossomo na próxima geração. A adaptabilidade relativa indica a adaptação de um determinado indivíduo em relação aos demais indivíduos da população. Por fim, outro método bastante utilizado é o elitismo. Este é um método que muitas vezes garante que os

melhores cromossomos não sejam perdido, pelo processo de recombinação e mutação, de uma geração para outra.

4 REDES NEURAIS

Um dos primeiros trabalhos utilizando redes neurais artificiais foi realizado por *Warren McCulloch* e *Walter Pitts* (HAYKIN, 2001a). Eles se basearam no conhecimento da fisiologia básica e da função dos neurônios no cérebro com a finalidade de simular o comportamento dos neurônios biológicos. Esses dois pesquisadores propuseram um modelo de neurônio artificial, no qual cada neurônio se caracteriza por estar “ligado” ou “desligado”. Eles mostraram que qualquer função computável podia ser calculada por uma rede de neurônios conectados e que a maioria dos conectivos lógicos (e, ou, não) podia ser implementada por estruturas de redes simples. *McCulloch* e *Pitts* também afirmaram que as redes poderiam ser capazes de aprender.

Donald Hebb (HAYKIN, 2001a) demonstrou uma regra de atualização simples para modificar as intensidades de conexão entre neurônios. Sua regra, agora chamada aprendizagem de *Hebb*, continua a ser modelo influente até hoje (RUSSELL; NORVIG, 2003).

Redes Neurais Artificiais (RNA) são sistemas inspirados nos neurônios biológicos e processamento paralelo do cérebro, com capacidade de adquirir e utilizar conhecimento experimental. Para entender o funcionamento das RNA torna-se necessário conhecer um pouco do neurônio biológico.

O neurônio biológico possui o corpo celular (soma) onde estão o núcleo e ramificações de dois tipos: os detritos e o axônio. A junção entre dois neurônios é chamada de sinapse (GUYTON, 1988).

4.1 O NEURÔNIO ARTIFICIAL

O neurônio artificial é composto de nós conectados por vínculos orientados. O corpo do neurônio faz a soma ponderada do produto dos pesos da entrada e uma transferência é aplicada sobre a função de ativação para gerar a saída. Os pesos são as intensificações da força sináptica e podem ser fixos ou treináveis implementando as ligações entre as unidades e a intensidade com que o sinal é transmitido de um neurônio para outro.

O modelo de *McCulloch-Pitts* (MCP) é considerado um marco inicial das pesquisas em Redes Neurais Artificiais. Este modelo ficou conhecido como neurônio de MCP ou neurônio com peso (“*weight-based*”) e é a base para grande parte das redes neurais atualmente (HAYKIN, 2001a).

O modelo foi introduzido em 1943, onde as entradas são combinadas por um somatório de pesos sinápticos. O neurônio de MCP deve possuir a capacidade de autoajuste de acordo com os pares de entrada e saída. Esta característica é chamada de adaptação, em que os pesos sinápticos variam em resposta a um estímulo externo, onde o conhecimento do ambiente disponível ao professor (“*teacher*”) é transferido para a rede neural (HAYKIN, 2001a).

4.2 ARQUITETURA

A forma como os neurônios artificiais são conectados entre si definem a topologia da RNA. Ela é dividida em duas arquiteturas predominantes que são: Redes Neurais Recorrentes e Redes Neurais Multicamadas.

- **Redes Neurais Recorrentes:** As redes neurais recorrentes possuem as seguintes características básicas: Apresentam realimentação de sinais (“*feedback*”) e não entradas e saídas bem definidas. Nesta topologia, os sinais de saída retornam às entradas, mas não para o próprio neurônio. Esta arquitetura é também denominada Rede de *Hopfield*. As Redes Neurais recorrentes também são chamadas de redes dinâmicas, em virtude desse comportamento.
- **Redes Neurais Multicamadas:** São neurônios dispostos em várias camadas, em que as principais características desta rede são: não possuem realimentação entre as suas camadas e os sinais se propagam das entradas para as saídas.

4.3 RBF (*RADIAL BASIS FUNCTIONS*)

Esta seção trata sobre redes neurais com múltiplas camadas que não são treinadas por retropropagação (*backpropagation*) e não têm unidades de processamento com função de ativação.

Um dos primeiros trabalhos utilizando esta abordagem foi introduzido por Medgassy (HAYKIN, 2001a), cujos resultados foram posteriormente utilizados para interpolações para estimação de densidade, aproximações de funções de multivariações suave (HAYKIN, 2001a) dentre outros. Esta abordagem de rede neural é inspirada na propriedade de alguns neurônios biológicos chamada de resposta localmente sintonizada (*locally tuned response*). Tais células nervosas respondem seletivamente a um intervalo finito do espaço de sinais de entrada.

4.3.1 Arquitetura da RBF

As redes RBF são redes de alimentação diretas (*feedforward*) consistindo tipicamente de três camadas: entrada, escondida e saída. A Figura 29 ilustra um exemplo de arquitetura das redes RBF.

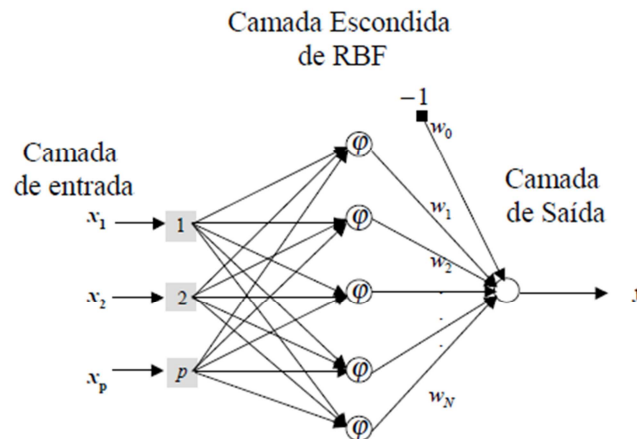


Figura 29: Redes RBF com 3 camadas.

A estrutura na Figura 29 mostra que a RBF realiza um mapeamento não linear do espaço de entrada para um novo espaço (camada escondida), da mesma forma realiza um mapeamento linear do espaço escondido para outro (camada de saída). A ativação

de uma unidade escondida é determinada por uma função não linear da distância entre o vetor de entrada e um vetor de referência.

O treinamento de uma RBF é tipicamente feita em dois estágios: determinam-se primeiramente as funções de base radial por meios de técnicas não supervisionadas usando para tal apenas os dados de entradas e a segunda camada (pesos) sendo determinadas por métodos lineares supervisionados.

No contexto de aproximações de funções, as redes RBF foram desenvolvidas para interpolação de dados em espaços multidimensionais. De acordo com Haykin (2001a), o problema de interpolação de dados pode ser formulado da seguinte forma:

Dado um conjunto de N pontos diferentes $\{\bar{x}_i \in \mathbb{R}^p \mid i=1,2,\dots,N\}$ em um conjunto correspondente com N números reais $\{d_i \in \mathbb{R}^l \mid i=1,2,\dots,N\}$ encontrar uma função $F : \mathbb{R}^p \rightarrow \mathbb{R}^l$ que satisfaça a condição de interpolação: $F(\bar{x}_i) = d_i, i=1,2,\dots,N$.

A superfície de interpolação $F(\cdot)$ deve passar em todos os pontos usados na aprendizagem. Assim, a função $F(\cdot)$ pode ser usada para mapear vetores \bar{x}_i que não pertence ao conjunto original de pontos desejados d_i . Uma possível solução para o mapeamento é escolher $F(\bar{x})$, tal que:

$$F(\bar{x}) = \sum_{i=1}^N w_i \varphi(\|\bar{x} - \bar{c}_i\|^2) \quad (4.1)$$

onde $\{\varphi(\|\bar{x} - \bar{c}_i\|^2) \mid i = 1,2,\dots,N\}$ é o conjunto de funções arbitrárias de base radial, $\|\cdot\|$ é a norma euclidiana, $\{\bar{x}_i \in \mathbb{R}^p, i = 1,2,\dots,N\}$ é o centro das funções de base radial. Em notação matricial, podemos escrever da seguinte forma:

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \quad (4.2)$$

onde $\varphi_{ij} = \varphi(\|\bar{x}_j - \bar{x}_i\|^2)$, $j, i = 1,2,\dots,N$, podemos fazer $\bar{d} = [d_1, d_2, \dots, d_N]$ e também $\bar{w} = [w_1, w_2, \dots, w_N]$. Assim, $\bar{\varphi} = \{\varphi_{ij} \mid j, i = 1,2,\dots,N\}$ é a matriz de interpolação. Reescrevendo com uma notação mais simples temos:

$$\bar{\varphi} \cdot \bar{w} = \bar{d} \rightarrow \bar{w} = \bar{\varphi}^{-1} \cdot \bar{d}, \quad (4.3)$$

considerando $\bar{\varphi}$ é não-singular.

A Tabela 1 mostra um conjunto de funções de base radial que são adequadas para interpolações (HAYKIN, 2001a).

Tabela 1: Conjunto de função de base radial.

<i>Nome</i>	<i>Função</i>
Lâmina spline fina	$\varphi(\xi) = \frac{\xi}{\sigma^2} \log\left(\frac{\xi}{\sigma}\right)$
Multi-quadrada	$\varphi(\xi) = \sqrt{\xi^2 + \sigma^2}$
Multi-quadrada inversa	$\varphi(\xi) = \frac{1}{\sqrt{\xi^2 + \sigma^2}}$
Gaussiana	$\varphi(\xi) = \exp\left(-\frac{\xi^2}{2\sigma^2}\right)$

O parâmetro σ ajusta o raio de influência de cada função. O valor de σ determina o quão rapidamente o valor da função de base radial cai a zero na medida em que \bar{x} se afasta do centro (\bar{c}_i).

A função do tipo gaussiana é comumente utilizada em muitas aplicações. Nela, a função do parâmetro σ corresponde ao desvio padrão da gaussiana. O σ nas redes RBF define a distância euclidiana média, que mede o espalhamento dos dados que são representados pela função de base radial em torno de seu centro (uma função radial típica é mostrada na Figura 13).

Em aplicações práticas, o valor do raio das funções de base radial afeta as propriedades numéricas do algoritmo de aprendizado. Entretanto, sua capacidade de aproximação não é afetada (HAYKIN, 2001a).

A maioria dos problemas utilizando RBF consiste em localizar os centros e outros parâmetros da função de base e ajustar os pesos em relação aos exemplos de treinamento.

Neste trabalho, elabora-se um projeto de um sistema misturador utilizando arquitetura de uma Rede Neural do tipo RBF. Posteriormente será mostrado que o aprendizado do sistema proposto se resume na estimação dos pesos e centros, ou seja,

equivalente a encontrar uma superfície em espaço multidimensional que melhor se adapte ao conjunto de exemplos de treinamento.

5 MÉTODO PROPOSTO

Nesta seção será descrito o método e as técnicas propostas no presente trabalho. Na seção 2, expomos o modelo geral da estratégia linear. Logo, vimos ao longo desta dissertação que se precisa estudar independência (doravante representamos independência como $I(\cdot)$) para separar sinais de fontes. Todavia, faz-se necessário separar momentos a fim de medir a independência de uma VA.

Para facilitar a tarefa de medir a independência estatística, mostra-se que é possível utilizar o branqueamento, o qual utiliza a correlação e que este tem suas limitações.

Observou-se também que podemos utilizar ferramentas da teoria da informação para maximizar a não-gaussianidade das fontes ou até mesmo maximizar a independência estatística.

Partindo do princípio da maximização da independência, podemos descrever o processo de separação das fontes lineares com os seguintes procedimentos:

1. Os sinais de fontes são misturados utilizando a Equação 2.2;
2. Deve-se especificar uma medida de independência de maneira que ela seja máxima quando aplicada à \mathbf{s} , ou seja, $I(\cdot)$ seja máxima;
3. Gerar uma matriz aleatória \mathbf{W} quadrada do tamanho de \mathbf{A} ;
4. Calcula-se a independência $I(\mathbf{W}\mathbf{x})$;
5. Se $\mathbf{W} = \mathbf{A}^{-1}$, então diz-se que o cálculo da independência tornar-se-á:

$$I(\mathbf{W}\mathbf{x}) = I(\mathbf{W}\mathbf{A}\mathbf{s})$$

$$I(\mathbf{W}\mathbf{x}) = I(\mathbf{A}^{-1}\mathbf{A}\mathbf{s})$$

$$I(\mathbf{W}\mathbf{x}) = I(\mathbf{s}) \quad (5.1)$$

6. Armazena-se a matriz (\mathbf{W}) que resultou na maior independência ou *Negentropia*.
7. Volta-se para o passo 3 fazendo esse processo iterativamente até que um critério de parada seja atingido.

Após esse processo, toma-se a matriz W que resultou na maior independência e assim se estimam as fontes independentes utilizando a Equação 2.39.

Na verdade, no passo 5 basta encontrar uma matriz de permutação conforme descrito na seção 2.4.

Visto o modelo linear, pretende-se agora estender a estratégia anterior ao conceito de não linearidade da seguinte forma:

1. Os sinais de fontes são misturados utilizando a Equação 2.40;
2. Especifica-se uma medida de independência $I(\cdot)$, de maneira que essa máxima seja aplicada à \mathbf{s} , ou seja, $I\{F(\mathbf{s})\}$ seja máxima;
3. Gera-se uma função G qualquer de dimensão de F ;
4. Calcula-se a independência $I(G(\mathbf{x}))$;
5. Se $G = F^{-1}$, então diz-se que: $I(G(\mathbf{x})) = I\{G(F(\mathbf{s}))\} = I(\mathbf{s})$, em que $I(\mathbf{s})$ é máxima;
6. Volta-se para o passo 3 gerando outro $G(\cdot)$ várias vezes e assim armazenar o $G(\cdot)$ que resultou na maior independência até que algum critério de parada seja atingido.

O passo 3 é segundo as regras dos Algoritmos Genéticos e a medida de independência $I(\cdot)$ é a *Negentropia* de Rényi e a função $F(\cdot)$ e $G(\cdot)$ é representada pela RBF. Assim, o $G(\cdot)$ que resultou na maior independência é a inversa de $F(\cdot)$.

5.1 SISTEMA MISTURADOR OU SEPARADOR

No decorrer da apresentação da estratégia linear mostrou-se que o sistema de mistura é representado por uma matriz quadrada. À luz desse paradigma, podemos agora definir o sistema misturador ou separador de natureza não linear.

Utiliza-se a estrutura de uma RBF para gerar um sistema misturador. Para representarmos nosso sistema de mistura não linear usa-se a seguinte estrutura dos pesos e centros: por simplicidade, supõe-se que queremos utilizar 2 centros e 2 pesos para

gerar uma mistura para dois sinais de fontes. Para cada centro e peso são precisos de dois pontos, que são representados da seguinte forma:

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

onde, os centros são representado por: $\mathbf{C}_1 = \begin{bmatrix} c_{11} \\ c_{21} \end{bmatrix}$ e $\mathbf{C}_2 = \begin{bmatrix} c_{12} \\ c_{22} \end{bmatrix}$. Os pesos são representados por $\mathbf{W}_1 = \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix}$ e $\mathbf{W}_2 = \begin{bmatrix} w_{12} \\ w_{22} \end{bmatrix}$, ou seja,

$$\mathbf{C} = [\mathbf{C}_1 \quad \mathbf{C}_2]$$

$$\mathbf{W} = [\mathbf{W}_1 \quad \mathbf{W}_2]$$

Com essa notação, são definidos os números de bits para representar nossos indivíduos no AG (apenas 2 bits para cada gene). Neste caso, obtêm-se 4 genes para representar os centros e mais 4 genes para representar os pesos, somando-se a um total de 8 genes.

Finalmente, uma população para essa simples representação poderá ser parecida com a expressão a seguir:

$$População = [c_{11} \quad c_{12} \quad c_{21} \quad c_{22} \quad w_{11} \quad w_{12} \quad w_{21} \quad w_{22}]$$

A partir deste momento os centros e os pesos iniciais são gerados aleatoriamente e são representados por \mathbf{C} e \mathbf{W} , respectivamente. Analogamente, os pesos e centros estimados (solução candidata da RBF) são representados respectivamente por $\widehat{\mathbf{W}}$ e $\widehat{\mathbf{C}}$. A função de base radial que estamos utilizando neste trabalho é gaussiana. A Figura 30 ilustra a estrutura da RBF para dois sinais de fontes com apenas 2 pesos e 2 centros.

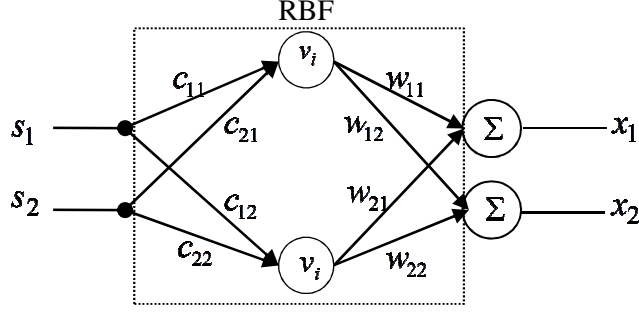


Figura 30: Exemplo simples do sistema misturador não linear.

A notação que representa v_i é equivalente a equação 4.1 e é ponderada pelos seus respectivos pesos. Finalmente a expressão que representa o sistema misturador ou separador de forma matricial é dada pela seguinte equação:

$$x = \widehat{W} \exp \left(- \frac{\|s - \widehat{C}\|^2}{2\sigma^2} \right) \quad (5.2)$$

Notadamente, a Figura 30 é semelhante à Figura 29, bastando apenas o algoritmo de aprendizado ajustar um parâmetro: os neurônios da camada oculta.

5.2 ENTROPIA DE RÉNYI PARA UMA VARIÁVEL ALEATÓRIA GAUSSIANA

Na Seção 2 descreveu-se alguns conceitos de *Negentropy*. Vimos que algumas delas utilizam aproximações por estatística de ordem superior e que é umas das categorias (separação cega de fontes) utilizadas em virtude de sua robustez.

A *Negentropy* significa o negativo da entropia, lembrando que maximizando a *Negentropy* é a mesma coisa que maximizando a não-gaussianidade das fontes. Outra abordagem para o uso da *Negentropy* é mostrado em (XU; PRINCIPE et al., 2002).

Neste momento, mediremos a entropia real dos dados e a entropia de uma VA gaussiana em conformidade com a definição da *Negentropy* dado pela Equação 2.36.

Primeiramente, descreveremos a entropia de Rényi para uma VA gaussiana que pode ser expressa por (MEDEIROS, 2005):

$$H_\alpha = \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} (k e^{-0,5(x-\mu)^t \Sigma^{-1}(x-\mu)})^\alpha dx \right) \quad (5.3)$$

onde $k = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}}$ é uma constante de normalização da distribuição.

Fazendo uma simples substituição, $y = x - \mu$ temos agora a seguinte expressão:

$$H_\alpha = \frac{1}{1-\alpha} \log \left(k^\alpha \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{\alpha}{2}(y)^t \Sigma^{-1}(y)} dy \right)$$

Agora vamos considerar que $\Sigma = \alpha \Sigma_q$, de forma que teremos:

$$H_\alpha = \frac{1}{1-\alpha} \log \left(k^\alpha \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{\alpha}{2}(y)^t (\alpha \Sigma_q)^{-1}(y)} dy \right)$$

$$H_\alpha = \frac{1}{1-\alpha} \log \left(k^\alpha \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}(y)^t \Sigma_q^{-1}(y)} dy \right)$$

Resolvendo a integral, obtemos:

$$H_\alpha = \frac{1}{1-\alpha} \log(k^\alpha \sqrt{(2\pi)^d |\Sigma_q|})$$

$$H_\alpha = \frac{1}{1-\alpha} \log(k^\alpha \sqrt{(2\pi)^d \alpha^{-d} |\Sigma|})$$

Substituído o valor de k na expressão anterior temos:

$$H_\alpha = \frac{1}{1-\alpha} \log \left(\left(\frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \right)^\alpha \sqrt{(2\pi)^d \alpha^{-d} |\Sigma|} \right)$$

$$H_\alpha = \frac{1}{1-\alpha} \log \left(\frac{\sqrt{(2\pi)^d \alpha^{-d} |\Sigma|}}{((2\pi)^d |\Sigma|)^{\frac{\alpha}{2}}} \right)$$

Simplificando tem-se:

$$H_\alpha = \frac{1}{1-\alpha} \log \left(\sqrt{(\alpha^{-d} ((2\pi)^d |\Sigma|)^{0,5(1-\alpha)})} \right)$$

$$H_\alpha = 0,5 \frac{1}{1-\alpha} [-d \log(\alpha) + (1-\alpha)d \log(2\pi) + (1-\alpha) \log(|\Sigma|)]$$

Finalmente virá:

$$H_\alpha = 0,5 \left[-d \frac{\log(\alpha)}{1-\alpha} + d \log(2\pi) + \log(|\Sigma|) \right] \quad (5.4)$$

Neste trabalho, utiliza-se a entropia de *Rényi* de ordem 2 (entropia quadrática) com parâmetro $\alpha = 2$ para o cálculo da entropia de variável aleatória gaussiana (x_g) de dimensão d , com média μ e matriz de covariância Σ , que pode ser expresso por:

$$H_2(x_g) = 0,5(d \log(4\pi) + \log|\Sigma|) \quad (5.5)$$

O cálculo da entropia de *Rényi* de ordem 2 é bem mais simples que o cálculo para entropia de *Shannon*. Esta simplificação acontece porque no cálculo da entropia de *Rényi* a integral resultante ocorre somente no quadrado do somatório das gaussianas e não no logaritmo do somatório, como é o caso da entropia de *Shannon* (MEDEIROS, 2005).

A seguir, estimamos a entropia de *Rényi* em vez das convencionais entropia de *Shannon* das amostras dos conjuntos de dados. Antes de comentar sobre os casos em que a *fdp* são estimados usando Janelas de *Parzen* (PARZEN, 1962) com o *kernel* gaussiano vamos rever um breve conceito sobre o método de janelas de *Parzen* e em seguida descrever a medida de independência utilizada no presente trabalho.

5.3 FUNÇÃO KERNEL

Um kernel é uma função k para todo $x, z \in X$ que satisfaz a seguinte expressão:

$$k(x, z) = \langle \Phi(x), \Phi(z) \rangle, \quad (5.6)$$

onde Φ é um mapeamento em X para um espaço F , ou seja, $\Phi: x \mapsto \Phi(x) \in F$.

A função kernel é projetada para calcular o produto interno de F usando apenas os dados de entrada (GUSTAVO; ROJO; MARTÍNEZ, 2007). Assim, sempre que encontrarmos o produto interno $\langle \Phi(x), \Phi(z) \rangle$ podemos substituir pela função kernel $k(x, z)$. A escolha de k implicitamente determina Φ e F . A grande vantagem de usar a função kernel como produto interno é que caso tenhamos uma função kernel k então não precisamos saber a forma explícita de Φ . A função forma a base da técnica utilizada que será apresentada e aplicada neste trabalho. A seguir, serão ilustrados vários tipos de funções kernel que são comumente utilizadas.

Tabela 2: Exemplos de função kernel.	
Kernel	$k(x, z)$
Polinômio de degrau d	$(\langle x, z \rangle + c)^d$
Função gaussiana Base Radial	$\exp\left\{-\frac{\ x - z\ ^2}{2\sigma^2}\right\}$
Laplaciano	$\exp\left\{-\frac{\ x - z\ }{\sigma}\right\}$
Sigmóide	$\tan(a\langle x, z \rangle + b)$

Na Tabela 2, $\sigma > 0$ é um parâmetro de escala, $a, b, c \geq 0$, e d é um inteiro. A norma euclidiana é $\|x\|^2 = x^t x$, onde t significa operador transposto. O kernel gaussiano RBF e o laplaciano são exemplos de transformações invariantes a translação (ou estacionário) e o *kernel* polinomial são exemplos de *kernel* não estacionário (GUSTAVO; ROJO; MARTÍNEZ, 2007). Pretende-se utilizar neste trabalho a função gaussiana de base radial. Notadamente, esta função é semelhante ao sistema misturador proposto na seção 5.2. Basta apenas acrescentar os pesos de uma RBF nesta expressão.

5.4 JANELAS DE PARZEN COM ENTROPIA QUADRÁTICA DE RÉNYI

O método de Janelas de Parzen é também chamado de estimativa de *kernel* (PARZEN, 1962). Vários métodos não paramétricos para estimar uma função de densidade apareceram na década de 60. Entre estes métodos o de janelas de Parzen é o mais popular. De acordo com o método, primeiramente temos que determinar a função kernel (DUBA; HART, 1973). Por simplicidade, considere uma simples função *kernel* Gaussiana como:

$$G(x, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{i=1}^N \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \quad (5.7)$$

onde σ irá controlar o tamanho do kernel, ou seja, a dispersão da função gaussiana centrada na variável x_i e x é uma VA, de forma que a função de densidade será (PRINCIPE, 2010):

$$f(x) = \frac{1}{N} \sum_{i=1}^N G(x - x_i, \sigma^2) \quad (5.8)$$

em que x_i é um conjunto de dado, para um sinal x (x pode ser um sinal de várias dimensões). Na verdade, cada ponto de dados será ocupado por uma função de kernel e toda a densidade é a média de todas as funções do kernel. No entanto, se escolhermos a entropia quadrática e observando o fato de que a integração do produto de duas funções gaussianas pode ser ainda avaliada por outra função gaussiana (VIOLA; SCHRAUDOLPH; SEJNOWSKI, 1995), então chegamos a um método simples utilizando a Equação 5.8 para estimar a fdp e em seguida o cálculo da entropia quadrática de Rényi como (XU; PRINCIPE et al., 2002):

$$\begin{aligned}
V &= -\log\left(\int_{-\infty}^{\infty} f_X(x)^2 dx\right) \\
&= -\log\left(\int_{-\infty}^{\infty} \left\{\frac{1}{N} \sum_{i=1}^N G(x - x_i, \sigma^2)\right\} \left\{\frac{1}{N} \sum_{j=1}^N G(x - x_j, \sigma^2)\right\} dx\right) \\
&= -\log\left(\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \int_{-\infty}^{\infty} G(x - x_i, \sigma^2) G(x - x_j, \sigma^2) dx\right) \\
V &= -\log\left(\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N G(x_i - x_j, 2\sigma^2)\right) \tag{5.9}
\end{aligned}$$

A expressão anterior é conhecida como potencial da informação denotado por V . Assim, a entropia é expressa em termos do potencial da energia. Portanto, a maximização da entropia passa a ser equivalente à minimização do potencial da informação (PRINCIPE; XU, 1999).

Finalmente, para o caso do método utilizando janelas de Parzen com *kernel* gaussiano (baseada em uma estimação não paramétrica da fdp do conjunto de dados) e juntamente com a entropia de Rényi, escreve-se a medida de independência que será utilizada no presente trabalho a *Negentropia* de Rényi como:

$$J(x) = \frac{1}{2} (d \log(4\pi) + \log(|\Sigma|)) - \log \sum_i \sum_j G(x_i - x_j, 2\sigma^2) \tag{5.10}$$

Na expressão anterior σ^2 é o tamanho do *kernel* (XU; PRINCIPE et al., 2002) da estimativa de Parzen da fdp que executa a soma de todos os N pontos no conjunto de dados. $G(x, \sigma^2)$ é um *kernel* gaussiano com tamanho σ , d é a dimensão dos dados e $|\cdot|$ é a determinante. A escolha do tamanho do *kernel* pode ser feita através de testes

empíricos (XU; PRINCIPE et al., 2002) ou utilizando uma aproximação dada por $1,6 \cdot n^{-1/5}$ (SILVERMAN, 1986), em que n é a quantidade de amostras. A covariância sigma é estimada por:

$$\Sigma \approx \frac{1}{N} \sum_i (s_i - \bar{s})^t (s_i - \bar{s}) \quad (5.11)$$

onde \bar{s} é a média dos dados. Portanto, o argumento do logaritmo na expressão do lado direito da Equação 5.10 é também chamado de potencial da informação em face da analogia com a energia potencial dada por um conjunto de partículas em um sistema físico. (PRINCIPE, 2010).

6 RESULTADOS EXPERIMENTAIS

Nesta seção, são apresentados alguns experimentos no paradigma de mistura linear usando a *Negentropia* de *Renyi* e posteriormente se segue com os experimentos utilizando o método não linear usando o sistema misturador proposto na seção 5.2. O aplicativo utilizado para simulações computacionais foi o Matlab.

6.1 SEPARAÇÃO CEGA DE FONTES LINEARES

Nestes experimentos, propõe-se uma alternativa de função custo com base na teoria da informação e utiliza-se o AG para maximizar a *Negentropia*, sobretudo visando a busca de uma matriz W que atuará como uma matriz de separação de forma que cada indivíduo no AG irá representar os elementos dessa matriz.

Para testar cada candidato para uma possível solução, desfazemos “a ação de mistura” dos sinais com cada matriz candidata usando a Equação 2.2. Com os valores das misturas “descorrelacionados” calculamos a *Negentropia* de *Renyi* usando a Equação 5.10, usando os procedimentos descritos na Seção 3.1.

Existem vários critérios de parada que podem ser utilizados no AG. Eles determinam quando deve parar de gerar novas populações e utilizar os melhores indivíduos como a solução do problema. Os critérios de parada podem ser simples como, por exemplo, chegar a um determinado número de geração ou pode ser mais elaborado. Em nosso caso, utilizamos o número máximo de gerações como critério de parada.

Para que os operadores genéticos representem adequadamente cada matriz candidata (indivíduos do AG), representamos cada matriz candidata por uma sequência de números binários fazendo uma grande cadeia de bits. Utilizou-se a Equação 3.1 para a escolha da quantidade de bits no problema. Em geral, maiores cadeia de bits aumentam a precisão numérica da solução. Assim, a cadeia é composta por cada elemento da matriz expresso em binário de 16 bits e através de simulações computacionais verifica-se que essa representação é suficiente para uma boa solução.

Os operadores genéticos utilizados neste experimento fazem basicamente 2 operações: Recombinação e Mutação, conforme descrito na Seção 3.

Para o cruzamento, a cadeia binária de cada indivíduo é dividida em dois em um ponto aleatório para efetuar a mistura. Em todas as simulações, o parâmetro da recombinação utilizado foi de 80% da população.

A mutação é a operação responsável por pequenas alterações que torna a pesquisa mais esparsa no domínio de busca. Definimos uma taxa de mutação, em geral um número pequeno, 1% da população, e para cada indivíduo é gerado um número aleatório para comparar com a taxa de mutação.

Essas medidas são aplicadas no loop principal do programa até que o critério de parada seja atingido. No final, o melhor individuo será utilizado como a solução do problema.

Testamos o novo método em várias situações diferentes, com sinais de misturas de imagem e formas de ondas. Em ambos os casos, escolhemos uma mistura aleatória (neste caso 3x3) e depois misturamos aos sinais. Para cada j -ésima imagem de Im_j , é escolhido cada i -ésimo pixel (na posição $\{k_i, l_i\}$) e montamos um vetor $s_j = [Im_j(k_1, l_1), \dots, Im_j(k_n, l_n)]$, em que $\{k_i, l_i\}$ varre todos os pixels da imagem). Para os sinais de forma de onda nada é feito. Em seguida, montamos um vetor de exemplos de dados $s(i)$.

Assim, utiliza-se o sinal de mistura como na Equação 2.2 e apresentamos os novos dados $x(i)$ (versão misturada de $s(i)$) ao AG para obter uma matriz de separação W , recuperando, finalmente, os sinais originais da mistura usando a Equação 2.39.

A seguir são ilustrados alguns resultados com três imagens e posteriormente com três formas de ondas e sinais de áudios. No exemplo a seguir visualiza-se uma imagem com um grande número de amostras.

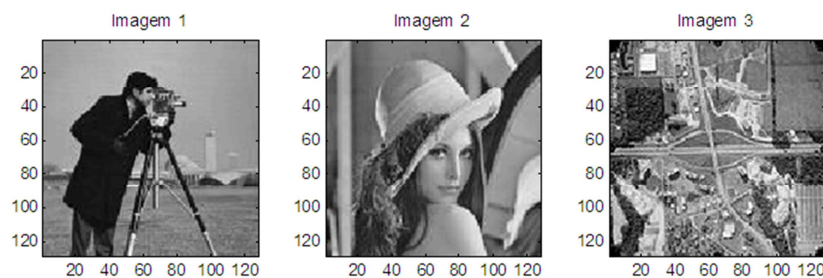


Figura 31: Imagem original com tamanho de 256x256 pixels.

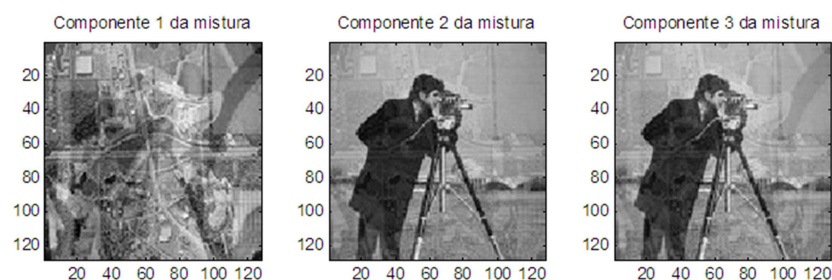


Figura 32: Imagens misturadas com uma matriz aleatória de 3x3 não-singular.

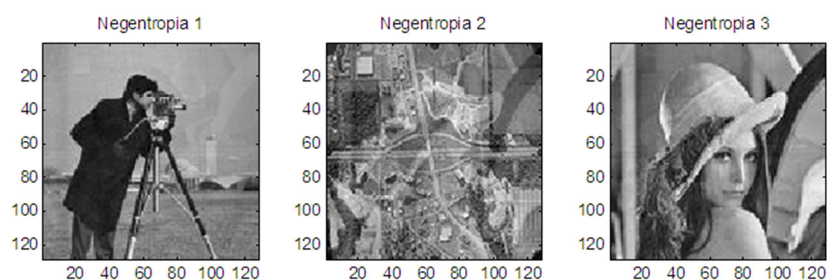


Figura 33: Recuperação das imagens utilizando abordagem linear.

A Figura 31 representa as imagens originais utilizadas neste experimento. Todas elas têm estatísticas diferentes, ou seja, histogramas diferentes. Na Figura 32 ver-se as imagens após a mistura. Claramente todas as três imagens têm alguma parte das imagens originais, mesmo com escalas diferentes (por exemplo, a imagem do mapa tem uma amplitude forte no primeiro componente, a imagem da câmera está fortemente presente nas duas anteriores e a imagem da Lena quase não aparece em todas as imagens). O resultado mostra que o algoritmo consegue separar as imagens mesmo em cenários mais complicados. Ver-se que aparecem pequenos artefatos os quais não comprometem os resultados (Figura 33).

No próximo exemplo, geramos um conjunto de formas de ondas que serão misturados por uma matriz aleatória. Entretanto, usamos apenas 50 amostras para produzir uma matriz de separação 3x3. Mesmo assim, o algoritmo foi capaz de produzir resultados bons. As Figuras 34, 35 e 36 mostram os sinais originais, as misturas e os sinais estimados, respectivamente.

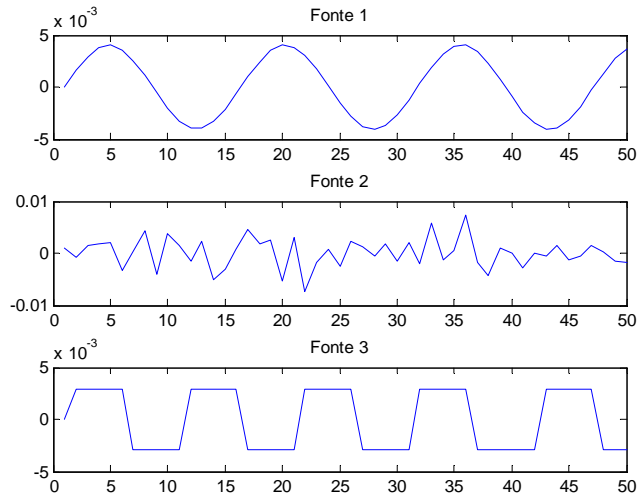


Figura 34: Sinais de fontes originais com 50 amostras.

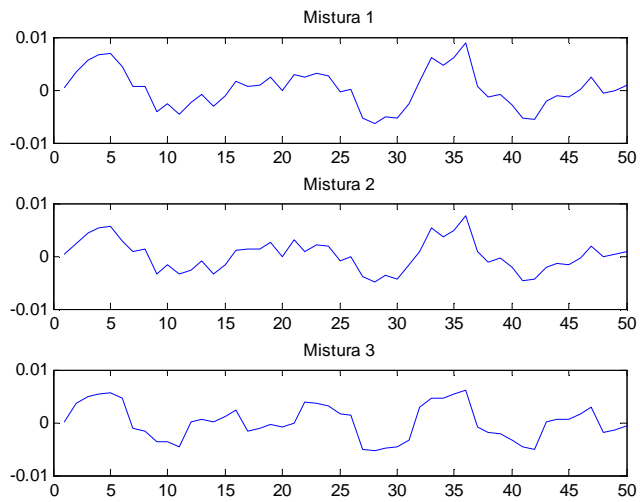


Figura 35: Forma de onda misturada com uma matriz 3x3 aleatória.

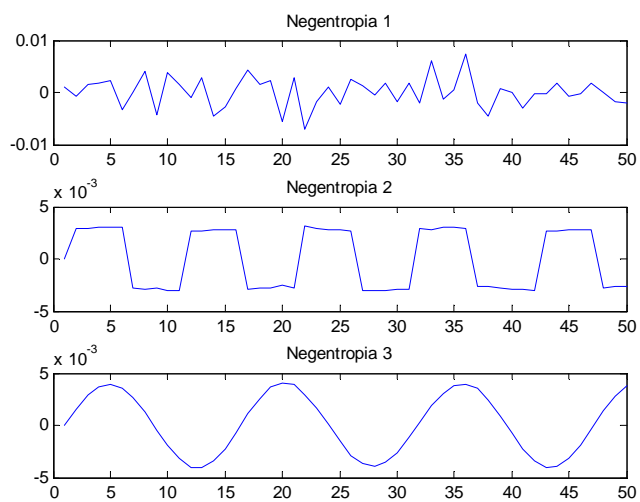


Figura 36: Forma de onda das fontes estimadas.

6.1.1 Comparação entre o método proposto e outras abordagens

Descreve-se nesta seção uma análise do AG com o algoritmo *P-ICA* proposto por (KUN, CHAN, 2006) e o *FastICA* (HYVÄRIEN, 1999b). A Tabela 3 representa o erro para vários tipos de tamanhos de populações e amostras. O erro é calculado como a soma dos mínimos quadrados para cada sinal *versus* a matriz de separação (DAMASCENO; MEDEIROS, 2010). A seguir, alguns experimentos e comentários relevantes.

Tabela 3 – Erro entre AG e abordagem *P-ICA* e *FastICA*.

Experimento	Amostras/Utilizados	População	Erro AG	Erro P-ICA	Erro FastICA
1	50/50	50	5.1818e-007	4.9440e-006	9.9365e-006
2	50/50	100	1.8315e-007	2.6616e-007	1.0328e-005
3	50/50	200	1.0785e-007	8.7361e-006	4.8179e-006
4	50/50	500	2.6961e-006	1.4256e-006	1.2347e-005
5	100/100	50	7.0970e-009	4.0143e-008	4.9859e-008
6	100/100	100	3.2528e-008	1.1351e-006	2.0106e-007
7	100/100	200	2.1759e-008	1.0717e-006	1.1719e-006
8	100/100	500	6.6869e-009	5.9524e-007	1.0514e-007
9	1000/100	50	9.0989e-013	6.8629e-010	2.8732e-010
10	1000/100	100	9.6552e-013	7.0855e-010	8.9876e-010
11	1000/67	200	1.2564e-011	6.4485e-010	4.1369e-010
12	1000/34	500	4.6911e-012	6.9532e-010	7.7314e-010
13	1000/20	500	2.6968e-011	1.5474e-011	9.0227e-010

Cada população na Tabela 03 foi gerada por misturas aleatórias e calculando-se os erros. As mesmas misturas utilizadas no AG foram utilizadas para o algoritmo *P-ICA* e *FastICA*. O *kernel* utilizado na *Negentropy* de Rényi foi de 0,05. A representação da recuperação das fontes pelo método AG e *P-ICA* são mostrados, conforme representado pela Figura 37.

Nesses experimentos empíricos observou-se que à medida que aumentamos o número da população o algoritmo converge com menos números de gerações como, por exemplo, no experimento 7 uma população de 200 indivíduos e com 100 amostras foi suficiente para o AG estimar as componentes independentes com 10 gerações em vez de 20 (Na maioria dos experimentos utilizamos 20 gerações como critério de parada).

No experimento 9 utilizamos 1000 amostras, das quais extraímos apenas 100 para gerar a matriz de separação e assim estimar as componentes independentes no AG, de forma que na abordagem *P-ICA* utilizamos as 1000 amostras, conforme resultados representados na Figura 39 e 40.

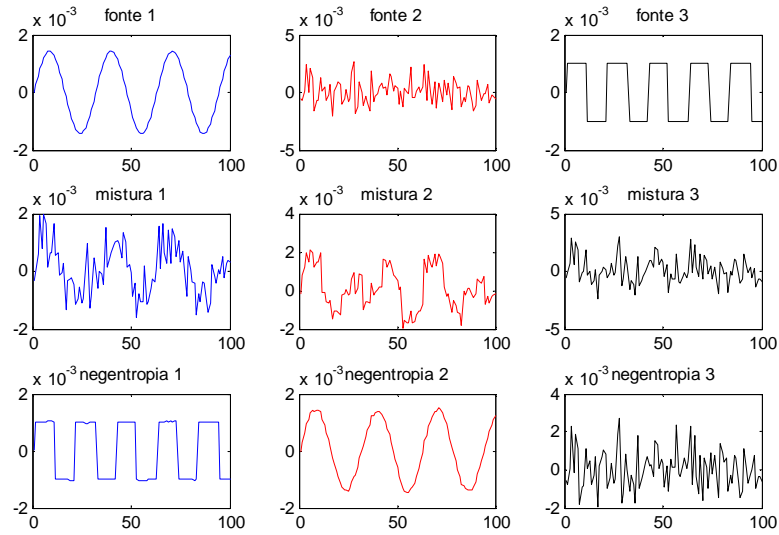


Figura 37: Método AG com 100 amostras e população de 50 indivíduos.

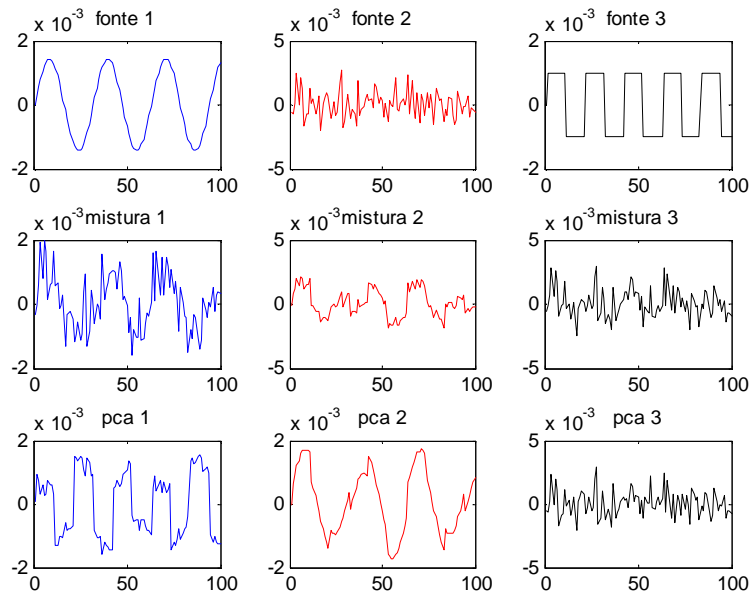


Figura 38: Método *P-ICA* com 100 amostras.

No experimento 11, das 1000 amostras na entrada do AG retiramos 67. Assim, o tempo de processamento do Algoritmo Genético obteve melhor desempenho. No entanto, utilizamos no *P-ICA* as 1000 amostras.

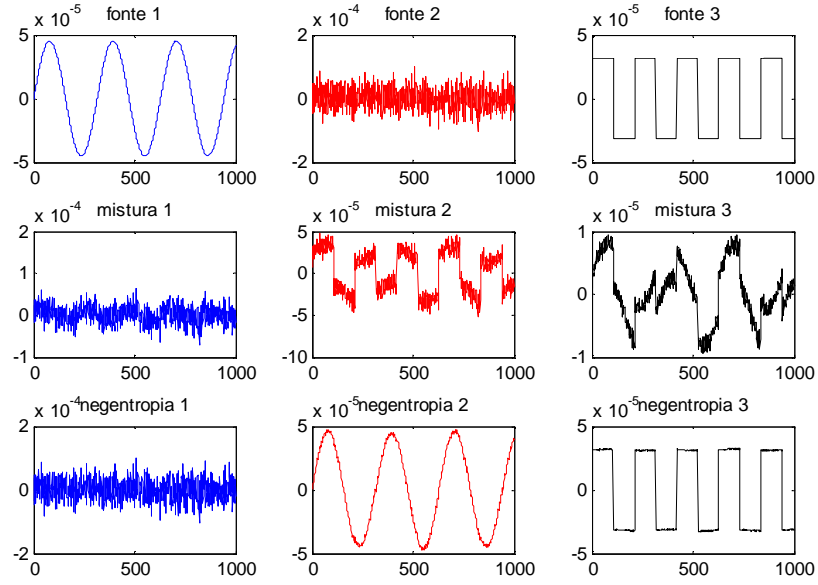


Figura 39: Método AG com 1000 amostras e população de 50 indivíduos.

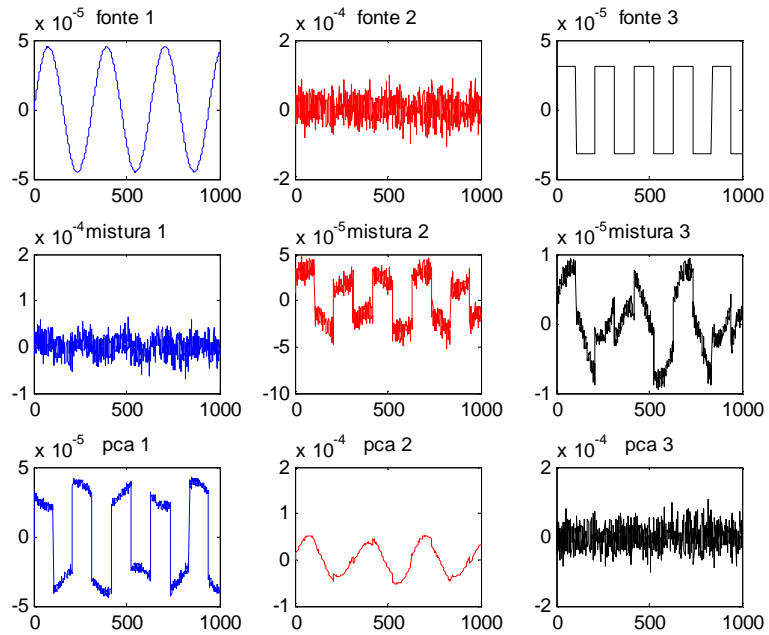


Figura 40: Método *P-ICA* com 1000 amostras.

No experimento 13, utilizamos o método de *Silverman* (SILVERMAN, 1986) para a escolha do *kernel* (parâmetro da *Negentropia*), que foi 0,2663 e 20 amostras para o aprendizado do AG. Entretanto, no *P-ICA* foram utilizadas 1000 amostras. Na Figura 41 e 42 são apresentados os resultados desses experimentos empíricos. Observe a dificuldade de ambos os métodos de estimar as componentes independentes, lembrando

que os sinais observados (misturas lineares) são utilizados em ambos os métodos.

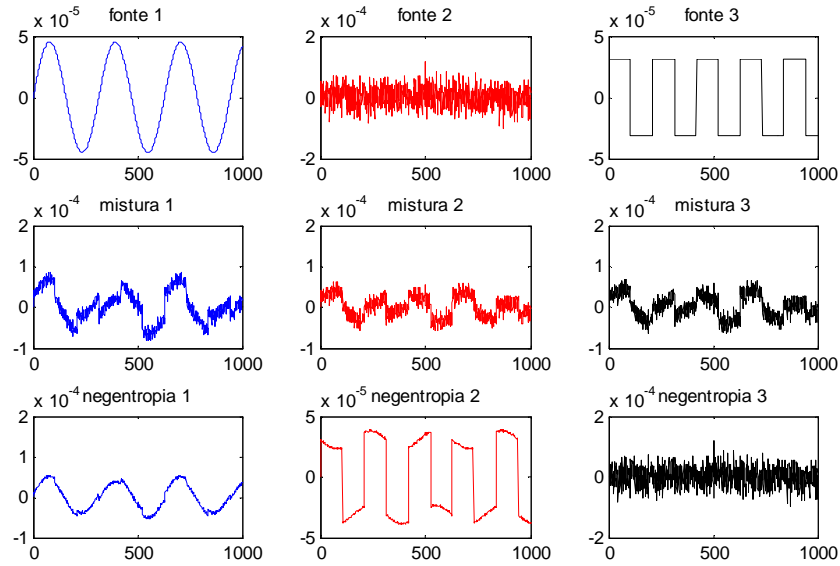


Figura 41: Método AG utilizando apenas 20 amostras das 1000 amostras que representam os sinais.

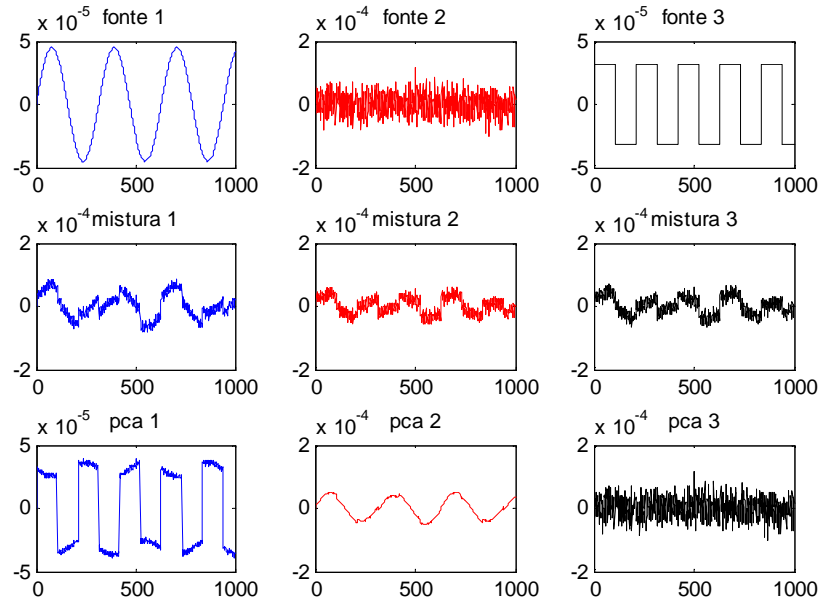


Figura 42: Método *P-ICA* utilizando todas as 1000 amostras.

Para ilustrar outra comparação entre os algoritmos proposto nesta seção utilizamos três sinais de voz com 23000 amostras. Os resultados foram obtidos com um *kernel* para a *Negentropia* de 0,1422 (usando *Silverman*). Para um melhor desempenho do algoritmo no AG, utilizamos das 23000 amostras dos sinais apenas 230

aleatoriamente. No entanto, no *P-ICA* utilizou-se todas as amostras, no qual o critério de parada no AG foi de 20 gerações e o tamanho da população utilizada foi de 50 indivíduos. O erro calculado como a soma dos mínimos quadrados foi $1,6468e^{-15}$ para o AG e $1,0585e^{-13}$ para o PCA. Estes resultados podem ser apreciados na Figura 43 e 44, respectivamente. Em algumas situações o *FastICA* teve dificuldade em estimar a 3ª componente independente. Em particular, um número elevado de amostras nos forçou a manipular seus parâmetros tais como a escolha da função *G* conforme descrito na Seção 2.5.2, bem como o máximo número de iterações.

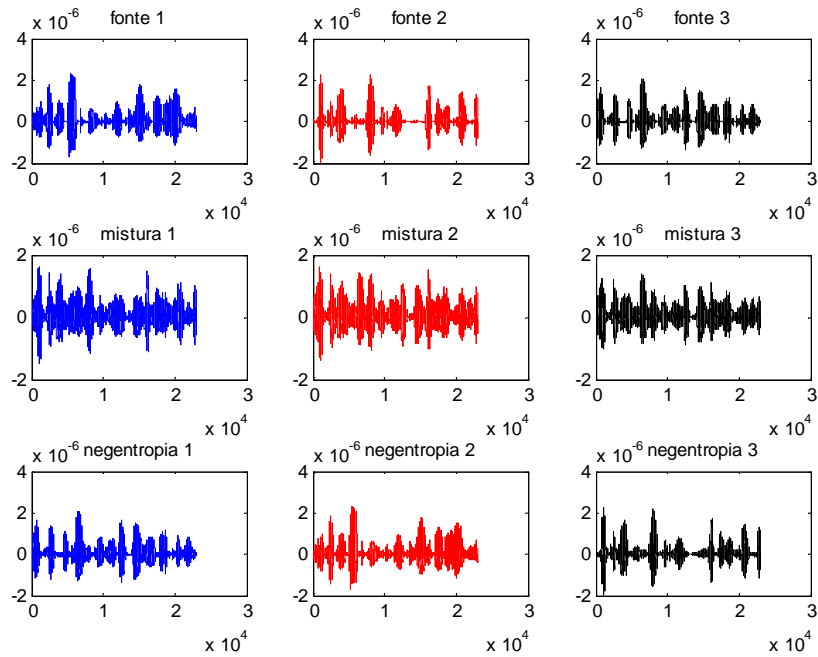


Figura 43: Recuperação de sinais de áudio usando a *Negentropia* de *Renyi*.

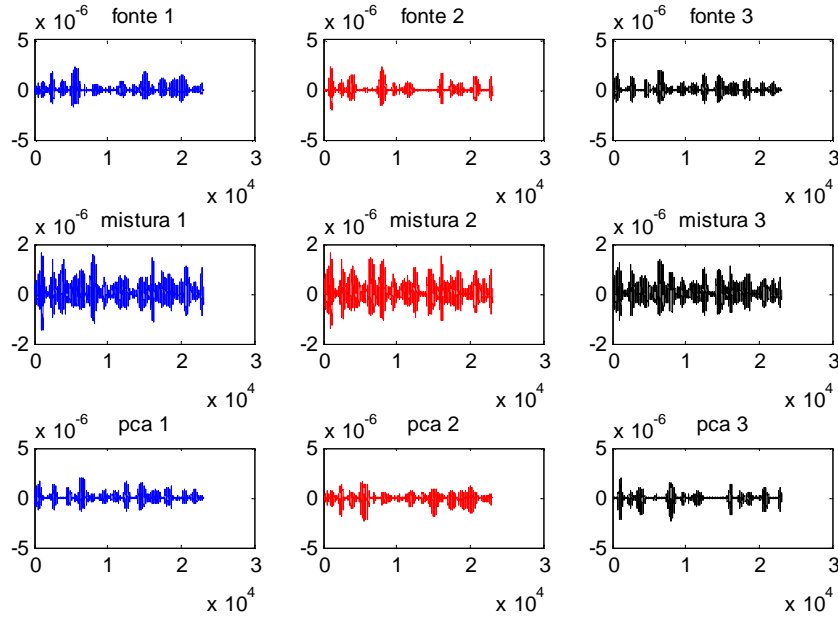


Figura 44: Recuperação de sinais de áudio usando *P-ICA*.

6.2 SEPARAÇÃO CEGA DE FONTES NÃO LINEARES

Na seção 2.6 expomos que devemos restringir o sistema misturador, pois há um número infinito de soluções.

Neste trabalho, utiliza-se uma rede RBF para gerar a mistura inicial, de forma que o objetivo é achar outra RBF que inverta o que a RBF inicial fez. Todavia, a população do AG são populações de RBF candidatas em vez de matrizes de misturas quadrada. A Figura 45 é semelhante à Figura 5 e ilustra o processo de mistura e separação usando esta abordagem.

Devemos projetar um sistema separador que seja capaz de inverter a ação do sistema misturador não linear, lembrando que este problema é também dito cego em razão a falta de informação sobre as misturas não lineares.

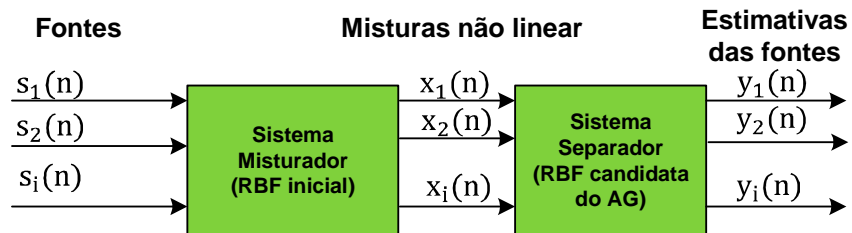


Figura 45: Diagrama esquemático do problema de separação cega não linear.

Após exposto o sistema misturador na seção 5.2, podemos partir para os testes empíricos. No entanto, antes de descrevermos os primeiros experimentos com a *Negentropia* de Rényi, pretende-se demonstrar, através de simulação computacional, se é possível separar as fontes usando este tipo de medida, mostrando que os sinais originais têm melhor medida (maior *Negentropia*) do que as misturas. Em especial, queremos saber se de alguma forma com o processo de busca por força bruta podemos encontrar uma *Negentropia* que seja máxima.

Nesse teste empírico, mediu-se o valor da *Negentropia* dos sinais de fontes, em seguida gerou-se várias redes RBF com inicialização aleatória para serem misturadas com os sinais de fontes, misturando estes sinais usando a Equação 5.2 e mediu-se a *Negentropia* (usando a Equação 5.10) de cada um destes, de forma que o valor da *Negentropia* dos sinais de fontes tem que ser maior que a *Negentropia* dos sinais misturados.

A Tabela 4 e 5 descrevem os testes realizados usando vários sinais de fontes. Para estes testes utilizamos 10 pesos e 10 centros. O parâmetro de suavização (*kernel* da *Negentropia* de Rényi) foi 0,05, a escolha do σ da função de base radial gaussiana (veja Tabela 1) foi 0,5.

Tabela 4 – Análise da *Negentropia* de Rényi dos sinais da Figura 51.

<i>Nº de RBF</i>	<i>Negentropia dos sinais de fontes</i>	<i>Máxima Negentropia da mistura</i>
1000	113,5771	113,5279
10000	113,5771	112,9077
50000	113,5771	112,8658

Tabela 5 – Análise da *Negentropia* de Rényi dos sinais na Figura 50.

<i>Nº de RBF</i>	<i>Negentropia dos sinais de fontes</i>	<i>Máxima Negentropia da mistura</i>
1000	203,4169	203,2584
10000	203,4169	203,1947
50000	203,4169	203,3334
100000	203,4169	203,2143

Pode-se constatar nestes experimentos, utilizando a ação de força bruta, o quão a máxima *Negentropia* é estimada. Isso nos motivou a pressupor que através desta busca (maximização da *Negentropia* de Rényi) encontra-se a máxima *Negentropia*, que resulta nos sinais de fontes, bastando apenas utilizar um algoritmo de busca global.

6.2.1 Algoritmo Genético com mistura não linear

Nesta seção, descreve-se o Algoritmo Genético usando *Negentropy* de Rényi para o caso de mistura não linear. Neste caso, a RBF é representada pelos parâmetros ajustados pelo AG para que a rede represente a inversa da função usada para misturar os sinais.

Toda informação sobre a mistura é dado pelas várias redes de base radial candidatas que são representadas pelas populações no AG. A seguir, são elencados no algoritmo AG os seguintes procedimentos:

1. Inicializar aleatoriamente W e C ;
2. Inicialize um σ para a RBF;
3. Use a Equação 5.2 para misturar os dados;
4. Gera N indivíduos na população;
5. Para cada indivíduo na população use a Equação 5.2 para estimar os sinais de fontes;
6. Use a Equação 6.10 para medir a *Negentropy* dos sinais estimados;
7. Selecione o mais apto na população;
8. Aplique o operador de recombinação e mutação na população;
9. Incremente o número de geração;
10. Enquanto o número máximo de geração não for atingido volte para o passo 5.

O critério de parada é semelhante à abordagem linear proposto na seção 6.1. Caso o número máximo de gerações seja atingido o algoritmo apresenta os resultados.

O passo 6 pode ser detalhado da seguinte forma: pega-se cada indivíduo da população que são pesos e centros e decodificamos (população é binária) para números reais; de posse dos possíveis pesos e centros usa-se a Equação 5.2 para estimar os sinais de fontes; mede-se a *Negentropy* desses sinais; em seguida armazena-se os valores dessa *Negentropy*; após atingir o critério de parada, pega-se o indivíduo que resultou na maior *Negentropy* na população; e utilizam-se os centros e os pesos dessa população para estimar as fontes independentes utilizando novamente a Equação 5.2. A seguir, são apresentados os primeiros resultados.

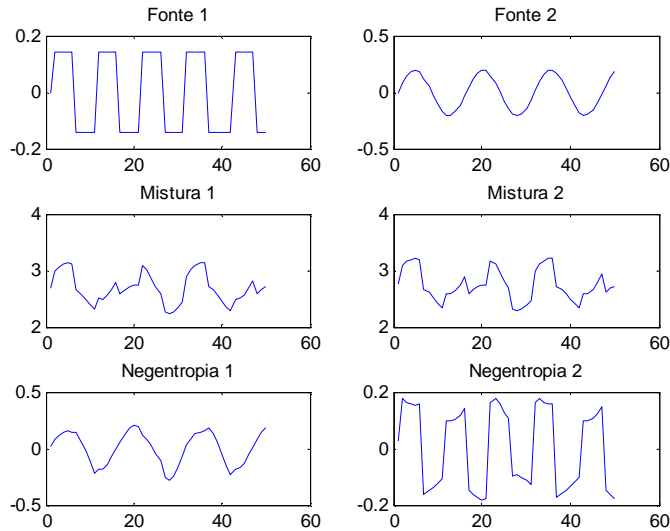


Figura 46: Separação de fontes de uma onda quadrada e senoidal usando abordagem não linear.

Nestes experimentos, os parâmetros do Algoritmo Genético foram previamente definidos e são basicamente os mesmos utilizados na abordagem linear, lembrando que a forma como são apresentados os exemplos no AG não linear é semelhante ao método linear exposto na seção 6.1.

Os resultados obtidos do algoritmo são mostrados na Figura 46. Utiliza-se 50 amostras para representar cada sinal de fonte, no qual o critério de parada utilizado foi o número máximo de gerações que foi 10000. Estes resultados foram obtidos com os seguintes parâmetros no algoritmo genético: 10 pesos e 10 centros para a RBF, *kernel* da *Negentropia* com 0,01, a quantidade de bits utilizados no AG foi 16 bits, o σ utilizado na função de base radial gaussiana foi 0,9 e a população foi de 200 indivíduos.

Mediu-se a *Negentropia* das fontes independentes, que foi 58,5271, e a máxima *Negnetropia* estimado pelo AG, que foi 55,7490. Isto nos leva a concluir que aumentando o número de gerações no AG ele tende a convergir.

É de fundamental importância descrever que para este resultado obteve-se um alto custo computacional, de forma que o AG passou mais de 3 horas para chegar ao número máximo de geração. A seguir, é mostrado outro resultado representado pela Figura 47 usando uma quantidade maior de amostras para os mesmos tipos de sinais utilizados no experimento anterior.

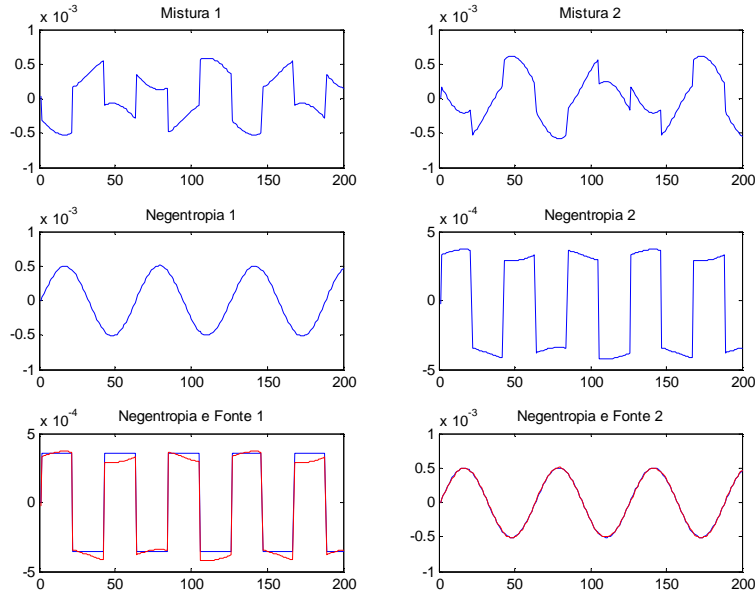


Figura 47: Separação de sinais para 200 amostras na parte inferior da figura foram sobrepostos os sinais as fontes e os sinais estimados, em que os sinais azul são os sinais de fontes e vermelho os estimados.

Vê-se claramente que os resultados foram satisfatórios, onde a recuperação da onda quadrada contém artefatos. Entretanto, não comprometendo o resultado, por outro lado a onda senoidal foi bem estimada.

Os resultados foram obtidos com 10 centros e 10 pesos na RBF, onde o σ da função de base radial gaussiana foi 0,5, o *kernel* para estimar a entropia quadrática de *Renyi* também foi 0,5. O número máximo de gerações, neste caso, foi de 30, na qual também utilizamos uma população de 100 indivíduos com 40 genes. O tempo gasto pelo AG até sua convergência foi de aproximadamente 15 minutos.

Nas simulações, utilizamos como parâmetros básicos no Algoritmo Genético 16 bits para população e 40 genes. No total de 10 centros e 10 pesos na RBF, uma taxa de mutação de 0,01% da população e recombinação de 80%. Com estes procedimentos, pretende-se otimizar o tempo de convergência das simulações.

Reiterando que o número do *kernel* da entropia e σ da função de base radial foram os mesmos (0,5), o tempo de convergência foi muito menor em relação aos experimentos anteriores.

A seguir, é apresentado outro resultado utilizando duas imagens com 128x128 pixels. Os exemplos foram apresentados no AG da mesma forma conforme descrito nos procedimentos utilizados na seção 6.1. Utilizou-se 16384 amostras, uma população de

200 indivíduos, a largura do *kernel* de na *Negentropia* foi 0,5, o σ da RBF utilizado foi 0,5. Sobretudo, a quantidade máxima de gerações atingidas foram de 3 gerações, de forma que houve um grande custo computacional chegando aproximadamente a 2 horas e 30 minutos para apresentar o seguinte resultado ilustrado na Figura 48.

Novamente, apresenta-se na figura 49 outro resultado utilizando duas imagens com 256x256 pixels. Para cada pixel da imagem monta-se um vetor com 65536 amostras. Todavia, utilizou-se os mesmos parâmetros do experimento anterior, obtendo-se um total de 4 gerações em aproximadamente 24 horas.

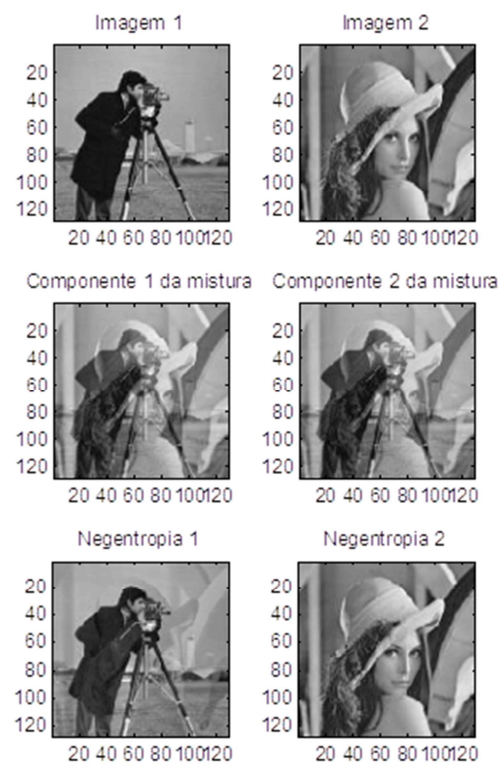


Figura 48: Recuperação de imagens usando abordagem não linear para uma imagem com 128x218 pixels.

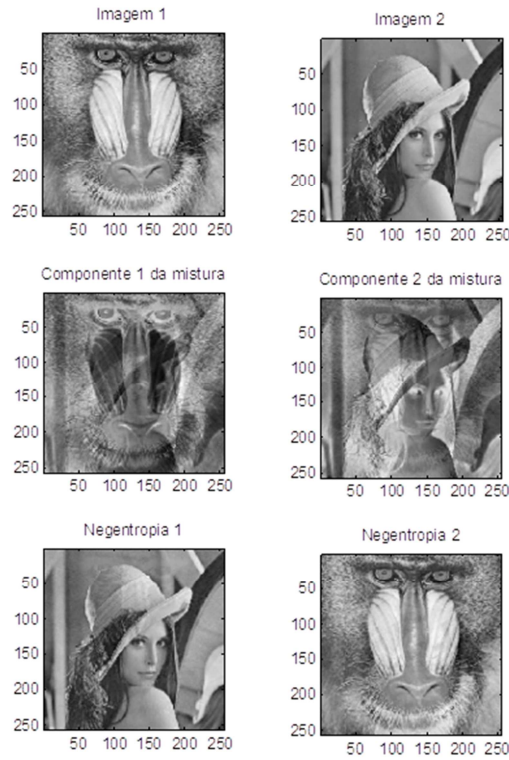


Figura 49: Recuperação de imagens usando abordagem não linear para uma imagem com 256x256 pixels.

No próximo experimento são utilizados dois sinais de áudio com 5000 amostras. Utilizou-se 30 centros e 30 pesos na RBF, uma população de 100 indivíduos, o parâmetro de suavização, o *kernel* da *Negentropia* foi 0,5 e o σ utilizado foi 0,5.

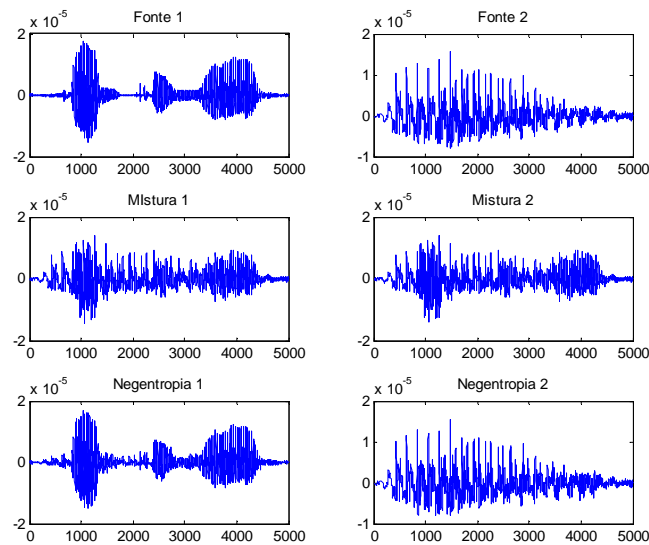


Figura 50: Separação de sinais de áudio utilizando AG não linear.

O Algoritmo convergiu com 5 gerações e consumiu aproximadamente 11 minutos para apresentar o resultado na Figura 50.

Finalmente, tem-se outro experimento em que são utilizados dois sinais de fontes com 334 amostras. Para esta simulação gerou-se novamente 10 centros e 10 pesos na RBF, bem como uma população de 100 indivíduos, o *kernel* da entropia foi 0,5 e o σ utilizado foi 5.

O programa utilizou aproximadamente 3 horas e 50 minutos para atingir 2000 gerações e apresentar os resultados expostos na Figura 51. Para termos uma visualização mais detalhada, na Figura 52 sobreposmos os sinais de fontes em azul e os estimados em vermelho.

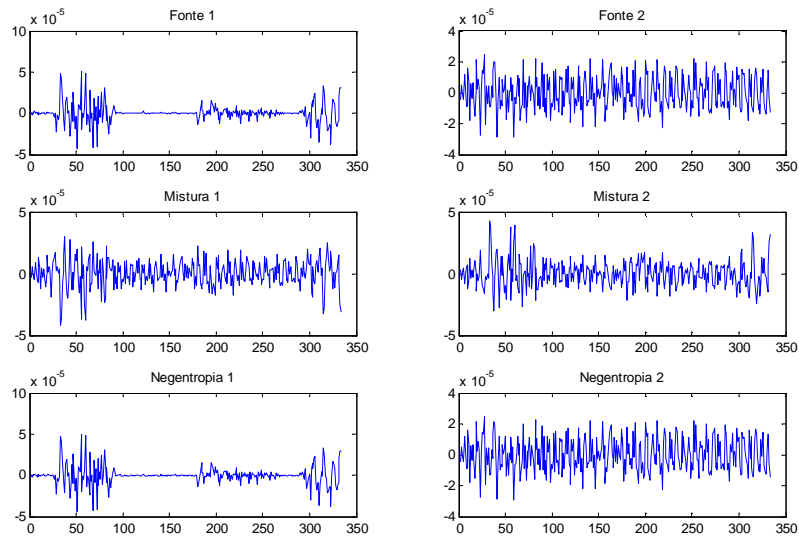


Figura 51: Ilustração de dois sinais de fontes com 334 amostras usando AG não linear.

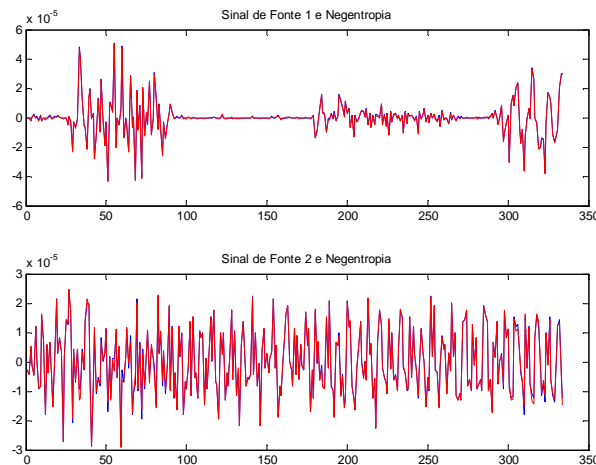


Figura 52: Sinais de fontes em cor azul e sinais estimados em cor vermelho, sobrepostos.

Para termos uma melhor visualização dos resultados de vários experimentos descritos nesta seção, a tabela 6 representa o erro para vários experimentos utilizando o modelo de mistura não linear, bem como alguns parâmetros do AG. O erro é calculado como a soma dos mínimos quadrados para cada sinal de fonte *versus* a função estimada.

Tabela 6 – Erro mínimo quadrado no AG não linear.

Experimento	Amostras	População	Nº de centros e pesos	Kernel Entropia	Sigma RBF	Erro AG
1	3500	50	10	0,5	0,5	1,7854e-005
2	80	100	10	0,5	1	3.0396e-006
3	3500	200	10	0,5	0,5	5,7138e-004
4	16384	200	10	0,5	0,5	3,7154e-017
5	200	100	10	0,5	0,5	1,3490e-007
6	334	100	10	0,5	5	8,2305e-012
7	65536	100	10	0,5	0,5	4.0027e-017
8	5000	100	30	0,5	1	6,9189e-012

6.3 SEPARAÇÃO CEGA DE FONTES COM ADIÇÃO DE RUÍDO

Nesta seção, será abordada a separação cega de fontes com adição de ruídos nos métodos linear e não linear propostos nesta dissertação.

Um ruído pode ser definido como um sinal que interfere no nosso sinal de comunicação. Ele pode estar presente em várias aplicações e dependendo da fonte do sinal podemos ter ruído acústico, eletrônico, eletromagnético (rádio).

Nesta abordagem de separação cega de fontes é mostrado o comportamento da ação do sistema separador quando aqueles sinais são submetidos ou contaminados por ruídos indesejáveis.

Na maioria das simulações utilizou-se um modelo generalizado de ruído gaussiano de média zero e variância unitária que são bastante utilizados na literatura (HAYKIN, 1996). Todavia, avalia-se que os algoritmos propostos nesta dissertação são capazes de estimar os sinais de fontes com a presença de ruído na mistura.

Há várias técnicas para reduzir o ruído tais como filtragem adaptativa, ICA, *Wavelets* e Análise de Componentes Principais, tendo o ICA tornado um método bem sucedido. Há também alguns trabalhos nesta área (SUN; CHAN, 2002) em que foi utilizado uma alternativa de redução de ruído de um Eletrocardiograma (ECG) empregando um algoritmo genético que maximiza a *kurtosis*.

Na primeira tentativa, foram utilizados três sinais de fontes. O experimento pode ser representado pelo seguinte diagrama da Figura 53 com adição de um ruído que é dado por $r(n)$:

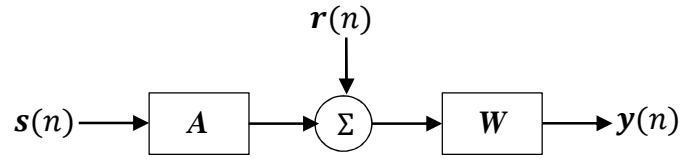


Figura 53: Modelo do sistema linear com adição de um ruído gaussiano.

Ao acrescentarmos um ruído gaussiano, em cada geração no Algoritmo Genético haverá um ruído aleatório na matriz de separação que tornará um pouco mais complicado a recuperação das fontes originais. Na Figura 54 são mostrados os resultados em que se utilizou os mesmos sinais de fontes expostos na Seção 6.1.

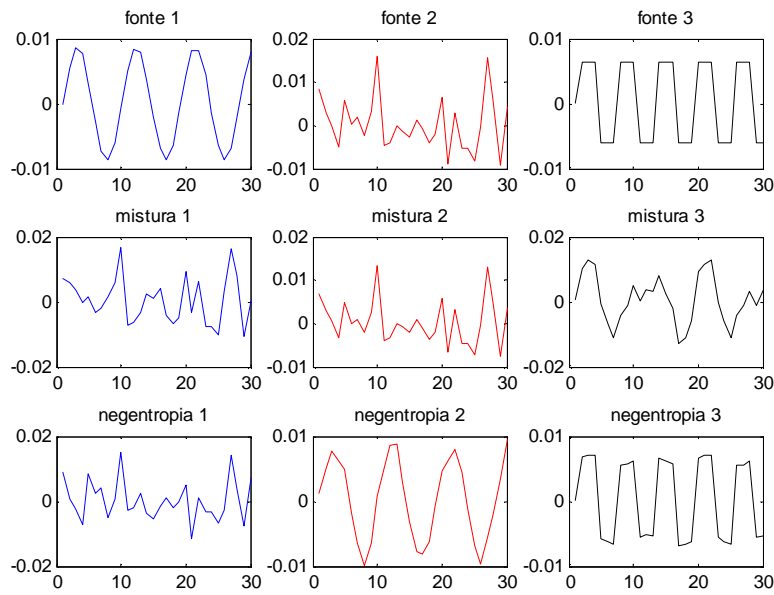


Figura 54: Separação de fontes linear com adição de ruído gaussiano.

Novamente o resultado foi satisfatório, levando em conta que a cada geração houve uma perturbação do sistema separador linear. Constata-se que somente a onda quadrada não foi bem estimada. Os resultados foram obtidos com os mesmos parâmetros do algoritmo genético exposto na seção 6.1. Utilizou-se apenas 30 amostras para gerar os sinais de fontes. O gráfico abaixo mostra o erro calculado com a soma dos

mínimos quadrados para cada sinal *versus* a matriz de separação em cada geração no AG.

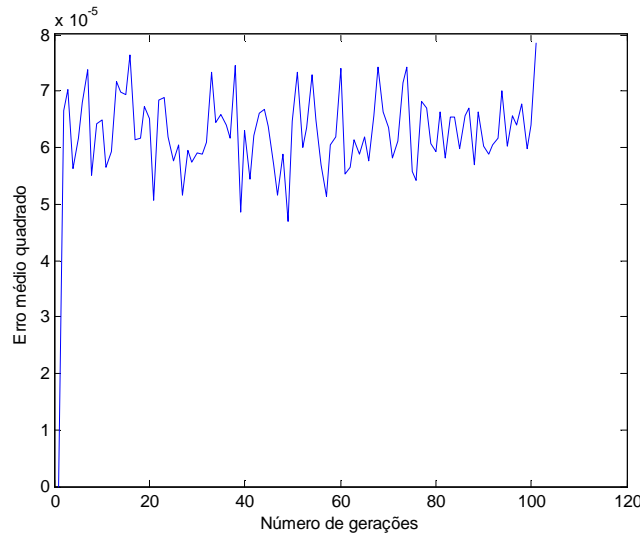


Figura 55: Erro da soma dos mínimos quadrados para cada sinal *versus* a matriz de separação a cada geração.

Na segunda tentativa de mostrar a capacidade da recuperação de fontes sobre a ação de um ruído gaussiano de média zero e variância unitária no sistema separador, utilizou-se outra abordagem para gerar o ruído gaussiano. Para todos os possíveis sinais estimados em cada geração no AG utilizou-se a seguinte expressão de forma matricial para gerar o ruído no sistema separador:

$$\mathbf{y} = \mathbf{x} \sigma \mathbf{r} \quad (6.1)$$

em que o σ é o mesmo utilizado na função da RBF, que neste caso foi 5, e \mathbf{r} é um ruído aleatório gaussiano. Assim, \mathbf{y} são funções candidatas com ruído.

Nesta simulação, utilizaram-se os mesmos parâmetros utilizados na Figura 51. A Figura 56 mostra o resultado do segundo experimento usando abordagem não linear com adição de ruído gaussiano.

A Figura 57 apresenta um gráfico dos sinais de fontes originais em azul e os sinais estimados em vermelho com 334 amostras que foram sobrepostos. Observe nesta mesma figura que os sinais estimados na parte superior da figura foram muito bem estimados. Os resultados foram obtidos com população de 200 indivíduos, 10 centros e 10 pesos na RBF, o σ utilizado na função de base radial foi 5, o *kernel* da entropia foi 0,5 e o algoritmo utilizou um tempo de aproximadamente 22 minutos até atingir seu critério de parada que foram 100 gerações.

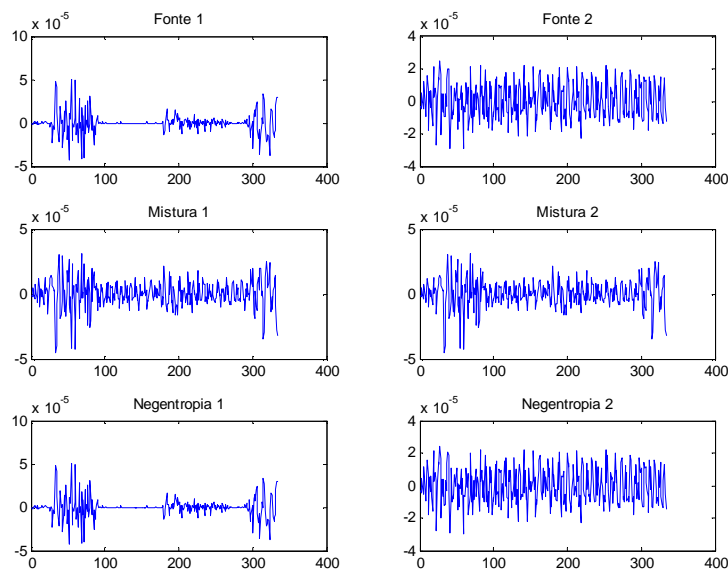


Figura 56: Separação de fontes não linear com ruído no sistema de separação.

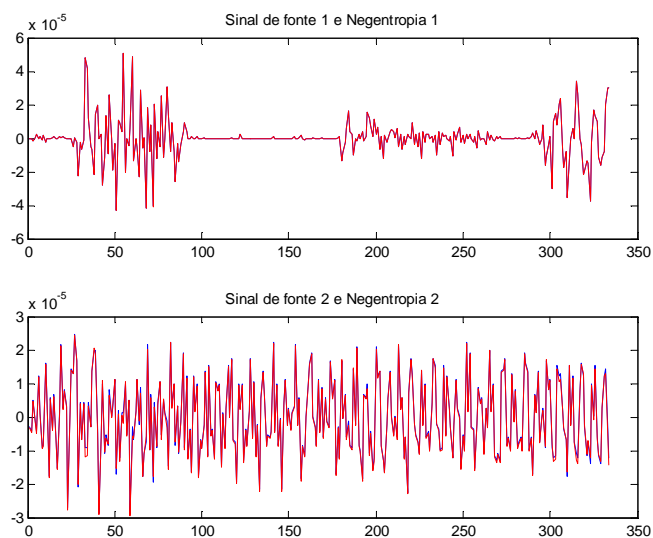


Figura 57: Sinais de fontes originais e estimados sobrepostos com adição de ruído no sistema de separação.

A Figura 58 ilustra o erro calculado com a soma dos mínimos quadrados para cada sinal *versus* os sinais de fontes estimados em cada geração.

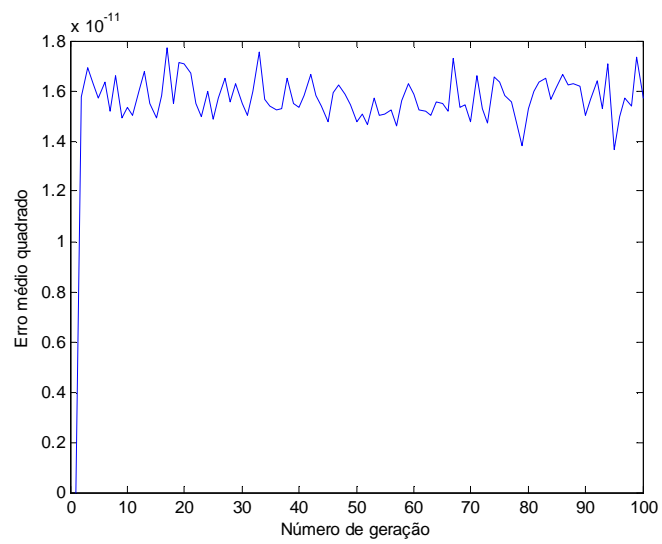


Figura 58: Erro da soma dos mínimos quadrados para cada sinal *versus* os sinais originais para cada geração com adição de ruído.

7 CONSIDERAÇÕES FINAIS

O cálculo da *Negentropia* envolve o cálculo da entropia dos dados. Esta, infelizmente, não é trivial e pode ser um problema. Existem várias maneiras de estimar a *Negentropia* de um sinal sem calcular a entropia diretamente, sendo a maioria deles baseados em estatísticas de alta ordem dos dados. Estes métodos não são bem adaptados numericamente em razão a questões relativas a grandes expoentes envolvidos no cálculo dos momentos.

Utilizou-se a entropia de *Rényi* das amostras do conjunto de dados em uma maneira não paramétrica, bem como a entropia de *Rényi* para uma VA gaussiana para estimar a *Negentropia* de *Rényi*.

Neste trabalho, propomos a aplicação de um AG para maximizar a *Negentropia* de *Rényi* das misturas usando uma mistura linear (matriz aleatória 3x3). Nos primeiros experimentos empíricos, utilizou-se o método linear para encontrar uma matriz de separação. O resultado foi satisfatório e fizemos uma comparação com duas técnicas bastante eficientes, de forma que concluímos que o algoritmo está bem adaptado para este tipo de problema, com devido ajustes nos seus parâmetros, tal como o parâmetro de *kernel* da *Negentropia* de 0,05 e também com um pequeno número de amostras.

Posteriormente, estendeu-se o conceito não linear no AG, sabendo que devemos restringir o sistema misturador. Neste caso, as misturas são geradas a partir de uma RBF. Os melhores rendimentos encontrados empiricamente foram para uma RBF com 10 centros e 10 pesos. Utilizou-se o parâmetro de suavização do *kernel* da entropia com um valor de 0,5, o σ da função de base radial foi sempre constante e o valor entre 0,5 a 5. Os melhores rendimentos, a priori, foram usando dois sinais de fontes que sejam supergaussianas ou subgaussianas. Como contribuição, há de ressaltar que houve um grande custo computacional, mesmo assim houve bons resultados.

Em ambos os métodos de separação cega (linear e não linear) geramos um ruído gaussiano com média zero e variância unitária para perturbar o sistema separador. Em ambos os métodos conseguiu-se estimar as fontes independentes. Outro fator relevante quanto a estimação das fontes não lineares é que a ordem das componentes independentes não pôde ser estimada em conformidade com ambiguidade descrito na Seção 2.

Para concluir, gostaríamos de expor algumas perspectivas para trabalhos futuros, ao passo que esta dissertação torne-se qualificação de doutorado, pretendendo-se

melhorar o desempenho do AG, fazer análise quanto à escolha do tamanho do *kernel* usando alguns métodos na literatura para escolha desse parâmetro, como por exemplo, o modelo dado por (SILVERNAN, 1986) ou buscar-se-á um novo modelo baseado nos próprios dados das misturas e assim fazer vários testes empíricos, a fim de encontrar uma boa estimativa das fontes aplicadas em cenários práticos e em tempo real, bem como experimentos preliminares em ambientes que o número de sensores é maior que sinais de fontes (caso sobre-determinados) ou subdeterminados.

E finalmente, levando em conta que o sistema separador não linear deve ser capaz de inverter a ação do sistema misturador, pretende-se demonstrar a prova da separabilidade do modelo proposto nesta dissertação.

REFERÊNCIAS

AMARI S., S.C. DOUGLAS, A. CICHOCKI, H.H. YANG. “*Novel on-line adaptive learning algorithms for blind deconvolution using the natural gradient approach*”. In Proc. 11th IFAC Symposium on System Identification, volume 3, pag. 1057–1062, Kitakyushu City, Japan, July 1997.

AMARI S., J.F. CARDOSO. “*Blind source separation , semi-parametric statistical approach*”, IEEE Trans. on Signal Processing, Dec. 1997.

ALAN J. I, “*Modern Multivariate Statistical Techniques, Regression, Classification, and Manifold Learning*”, Spring , USA, 2008.

BIRBAUMER N., L.G. COHEN, “*Brain-computer interfaces: Communication and restoration of movement in paralysis*”, J. Physiol., vol. 579, no. 3, pag. 621–636, 2007.

BODE H., C. SHANNON, “*A simplified derivation of linear least squares smoothing and prediction theory*,” Proc. IRE, Vol. 38, pag. 417-425, Apr. 1950.

CACHIN C., “*Smooth Entropy and Rényi Entropy*”, Lecture Notes in Computer Science, vol. 1233, (ed. Walter Fumy), Advances in Cryptology: Eurocrypt’97, Springer-Verlag, pag. 193-208, 1997.

CARDOSO J., “*Blind signal separation: statistical principles*”, Proc. IEEE 86 (10) 1998.

CARDOSO J., “*High-order contrasts for independent component analysis*”, Neural Computation, pag. 157-192, 1999.

CHOI S., A CICHOCKI, S. AMARI, “*Flexible independent component analysis*”, J. VLSI Signal Process, 2000.

COMON P., “*Independent component analysis, a new concept?*”, Signal Process, 1994.

CICHOCKI A., AMARI S., “*Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*”, John Wiley, New York, USA, 2002.

DAMASCENO N. C, MEDEIROS A. M., “*Blind Source Separation using genetic algorithms end negentropy as separation measure*”, XLII SBPO, Bento Gonsalves, RS, 2010.

DEMUTH, H.; BEALE, “*M. Neural Network Toolbox User’s Guide*”. The MathWorks, Inc: Massachusetts, USA, 1992.

DEVROYE L., L. GYORFI, “*Nonparametric Density Estimation*”, Wiley, New York, 1985.

DUDA R. O. P. E. HART, *“Pattern Classification and Scene Analysis”*, John Wiley & Sons, New York, 1973.

ERDOGMUS D., J.C. PRINCIPE *“Generalized information potential for training adaptive systems”*, IEEE Trans.Neural Networks, 2001.

FENG X., K. LOPARO, Y. FANG, *“Optimal State Estimation for Stochastic Systems: An Information Theoretic Approach”*, IEEE Transactions on Automatic Control, vol. 42, no. 6, pag. 771-785, 1997.

GAO P., L. KHOR, W. WOO, S. DLAY, *“Two-stage series-based neural network approach to nonlinear independent component analysis”*, Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pag. 4559-4562, Island of Kos, Greece, 2006.

GOLDBERG D., *“Genetic Algorithms in Search, Optimization and Machine Learning”*, Addison-Wesley 1989.

GUSTAVO C. V., J. L. A. ROJO, MARTÍNEZ R., *“Kernel Methods in Bioengineering, Signal and Image Processing”*, Idea Group Publishing, 2007.

GUYTON A. C *“Fisiologia Humana”*, 6ª ed. São Paulo: Guanabara Koogan S.A.1988.

HARTLEY R., *“Transmission of information”*, *Bell Syst. Tech. J.*, 7:535, 1928.

HAYKIN S., *“Adaptive Filter Theory”*, 3ª Ed., Prentice Hall, New Jersey, 1996.

HAYKIN S., *“Redes Neurais: Princípios e prática”*. 2ª ed. Porto Alegre: Bookman, 2001a.

HAYKIN S., *“Adaptive Filter Theory”*, 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2001b.

HE R., J. REED. *“A robust co-channel interference rejection technique for current mobile phone system”*. In Proc. IEEE VTC, pag. 1195–1199 vol.2, Atlanta, GA, 1996.

HÉRAULT, J., JUTTEN, C. & ANS, B. *“Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervise”*. In Actes du Xème Colloque GRETSI. Nice, France. 1985.

HILD K.E. II, D. ERDOGMUS, J.C. PRINCIPE, *“Blind source separation using Rényi’s mutual information”*, IEEE Signal Process, 2001.

HYVÄRINEN A., E.OJA. *“Independent component analysis: A tutorial”*. Technical report, 1999.

HYVÄRINEN A., *“Survey on independent component analysis”*, *Neural Computer Surveys* 2, 1999a.

HYVÄRINEN A., “*Fast and robust Fixed-point algorithms for independent component analysis*”, IEEE Trans. Neural Networks 10, 1999b.

HYVÄRINEN A., E. OJA. “*Independent Component Analysis: Algorithms and applications. Neural Networks*”, 2000.

HYVÄRINEN A., J. KARHUNEN, E. OJA, “*Independent Component Analysis*”. Wiley Interscience, 2001.

JAHN O., A. CICHOCKI, A. IOANNIDES, S. AMARI. “*Identification and elimination of artifacts from MEG signals using efficient independent components analysis*”, In Proc. of the 11th Int. Conference on Biomagnetism BIOMAG-98, pag. 224–227, Sendai, Japan, 1999.

JAIN A., “*Fundamentals of Digital Image Processing*”. Prentice Hall, New Jersey, 3 ed. 1989.

JOHN. HOLLAND, “*Adaptation in neural and Artificial systems*”, University of Michigan Press, 1975.

JIANWEI W., JINWEN. R. “*Entropy Penalized Learning Algorithm For Gaussian Mixture With Automated Model Selection*”, Ma2 ICSP2008 Proceedings, 2008.

JUTTEN C., M. B. ZADEH, S. HOSSEINI, “*Three easy ways for separating nonlinear mixtures*”, Signal Processing, pag 217-229, 2004.

JUTTEN C., J. KARHUNEN, “*Advances in nonlinear blind source separation*” Proceedings of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003), pag. 245-256, Nara, Japan, 2003.

KAI S., W. QI, and D. MINGLI, “*Approach to nonlinear blind source separation based on niche genetic algorithm*”, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), Jinan, China, 2006.

KUN Z, Lai-WAN CHAN, “*ICA by PCA Approach: Relating Higher-Order Statistics to Second-Order Moments*”, 2006.

LIDEN, R. “*Algoritmos genéticos*”, Rio de Janeiro: Brasport, 2006.

MEDEIROS, A. M., “*Contribuições aos processos de clustering com base em métricas não-euclidianas*”, Tese de doutorado, UFRN, Brasil, 2005.

MICHALEWICZ, Z. “*Genetics Algorithms + Data Structures = Evolution Programs*”, 3^a. ed. New York: Springer-Verlag Berlin Heidelberg, 1996.

MILLER G., D. HORN. “*Maximum entropy approach to probability density estimation*”, In Proceedings of Second International Conference on Knowledge- Based Intelligent Electronic Systems, 1998.

MÜLLER K.R., R. VIGARIO, F. MEINECKE, A. ZIEHE, “*Blind source separation techniques for decomposing event-related brain signal*”, Int. J. Bifurcation and Chaos, vol. 14, no. 2, pag. 773–791, 2004.

MULGREW B., “*Applying Radial Base Functions*”, IEEE Signal Processing Magazine, pag. 50-65. 1996.

OBITKO, M. “*Introduction to Genetic Algorithms*”. Disponível em: <<http://obitko.com/tutorials/genetic-algorithms>>, Março de 2010.

PAPOULIS, A. “*Probability, Random Variables, and Stochastic Processes*”, McGraw-Hill, 3rd edition, 1991.

PAJUNEN P., A. HYVÄRINEN, J. KARHUNEN. “*Nonlinear blind source separation by self-organizing maps*”. In Proc. Int. Conf. on Neural Information Processing, pag. 1207–1210, Hong Kong, 1996.

PANSALKAR, V.V.; SASTRY, P.S. “*Analysis of the Back-Propagation Algorithm with Momentum*”, IEEE Transactions on Neural Networks, Vol. 5, N°. 3, USA, pag. 505-506, 1994.

PALANIAPPAN R., C.N.GUPTA, “*Genetic Algorithm based independent component analysis to separate noise from Electrocardiogram signals*”, Proc. IEEE , 2006.

PARZEN E., “*On estimation of a probability function and mode*”, Ann. Math. Statist. , 1962.

PARZEN E., “*On estimation of a probability density function and mode, in: Time Series*”, Analysis Papers, Holden-Day, Inc., CA, 1967.

POBLADOR S., V., MONTE-MORENO, E. & SOLÉ-CASAL, J. “*ICA as a Preprocessing Technique for Classification*”, In Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation, ICA 2004, pag. 1165-1172, Granada, Espanha, 2004.

PRINCIPE J.C., D. XU, “*Information-theoretic learning using Rényi's quadratic entropy*”, 1st International Workshop on Independent Component Analysis and Signal Separation, Aussois, France, 1999.

PRINCIPE J.C., J.W. FISHER, D. XU, “*Information Theoretic Learning, in Unsupervised Adaptive Filtering*”, S. Haykin Editor, John Wiley & Sons, New York, 2000.

PRINCIPE J.C., D. XU, Q. ZHAO, J. FISHER, “*Learning from Examples with Information Theoretic Criteria*”, Journal of VLSI Signal Processing Systems, Special Issue on Neural Networks, pag. 61-77, 2000.

PRINCIPE J.C., “*Information Theoretic Learning, Renyi's Entropy and Kernel Perspectives*”, Springer New York, 2010.

RANDY L. H, DOUGLAS H. W. “*Genetic Algorithms in electromagnetic*”, IEEE Press, Wiley-Interscience publication, New Jersey, 2007.

RÉNYI A., “*Some Fundamental Questions of Information Theory*”. Selected Papers of Alfred Rényi, vol. 2, pp. 526-552, Akademia Kiado, Budapest, 1976.

RÉNYI A., “*Probability Theory*”, North-Holland, Amsterdam, 1970.

ROJAS F., M. RODRÍGUEZ-Á., I. R., MARTÍN C., “*Blind source separation in post-nonlinear mixtures using competitive learning, simulated annealing, and a genetic algorithm*”, IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, vol. 34, pag. 407-416, 2004.

RUSSELL, S.; NORVIG P. “*Inteligência Artificial*”, São Paulo: Campus, 2003.

RUBINSTEIN R.Y., “*Simulation and the Monte Carlo Method*”, Wiley, New York, 1981.

SANKAR K. Pal “*Eiben Classification and Learning Using Genetic Algorithms, applications in Bioinformatics*”, Spring, 2007.

SAHOO P., C. WILKINS, J. YEAGER, “*Threshold Selection Using Rényi's Entropy*” Pattern Recognition, vol. 30, pag. 71-84, 1997.

SILVERMAN, B. W. “*Density Estimation for statistics and data analysis*”, Chapman Hall, 1986.

SIVANANDAM S. N., S. N. DEEPA, “*Introduction to Genetic Algorithms*”, Spring Berlin Heidelberg New York, 2008.

SOLAZZI M., F. PIAZZA, A. UNCINI, “*Nonlinear blind source separation by spline neural networks*”, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 5, pag. 2781-2784, Salt Lake City, US, 2001.

SUN Y., K. L. CHAN e S. M. KRISHNAN, “*ECG signal conditioning by morphological filtering, Computer in Biology and Medicine*”, Vol 32, Pergamon, Elsevier, 2002.

TALEB A., C. JUTTEN, “*Source separation in post-nonlinear mixtures*” IEEE Trans on Signal Processing, no. 47, pag. 2807-2820, 1999.

TAN Y., JUNWANG, “*Nonlinear blind separation using an fbf network model*”, Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS00), vol. 3, pag. 634-637, Geneva, Switzerland, 2000.

TAN Y., J. WANG, “*Nonlinear blind source separation using higher order statistics and a genetic algorithm*”, IEEE Transactions on Evolutionary Computation, vol. 5, no. 6, pag. 600-612, 2001.

VIOLA P., N. SCHRAUDOLPH, T. SEJNOWSKI, “*Empirical Entropy Manipulation for Real-World Problems*”, Proceedings of Neural Information Processing System (NIPS 8) Conference, pag.851-857, Denver, Colorado, 1995.

XIANG-Yan Zeng, YEN-Wei Chen, ZENSHO Nakao and Gtsumi YAMASHITA, “*Signal separation by independent component analysis based on a genetic algorithm*”, IEEE. Department of Electrical & Electronics Engineering. Faculty of Engineering , University of the Ryukyus OOkinaawa, Japan, 2000.

XU D., J.C. PRINCIPE, J. FISHER, H. WU, “*A novel measure for independent component analysis (ICA)*”, to appear in IEEE Transactions, 2002.

YOSHIOKA M., S. OMATU, “*Signal separation method using genetic algorithms*”,in Proc. IEEE Int. Joint Conf. Neural Networks, vol.2, pp.909-912, 1998.

WALLACE, D. L., “*Asymptotic Approximations to Distributions, The Annals of Mathematical Statistics*”, Vol. 29, N° 03, pag. 635-654, Sep., 1958.

WIENER N., “*Extrapolation, Interpolation and Smoothing of Stationary Time Series, With Engineering Applications*”, Wiley, NY, New York, 1949.

ZSOLT, L. KOVÁCS, “*Redes Neurais Artificiais Fundamentos e Aplicações*”, 2ª ed. Editora Collegium Cognitio, 1996.

ZYCZKOWSKI K., “*Rényi extrapolation of shannon entropy*”, Open Syst. Inf. Dyn, 2003.