

UM ALGORITMO DE OTIMIZAÇÃO PARA MAXIMIZAR A NEGENTROPIA DE RÉNYI APLICADO AO PROBLEMA BLIND SOURCE SEPARATION NÃO LINEAR

Adrião D. Doria Neto

Universidade Federal do Rio Grande do Norte
Caixa Postal 1524, Lagoa Nova, Brasil
adriao@dca.ufrn.br

Allan de M. Martins

Universidade Federal do Rio Grande do Norte
Caixa Postal 1524, Lagoa Nova, Brasil
allan@dee.ufrn.br

Nielsen Castelo Damasceno

Universidade Federal do Rio Grande do Norte
Caixa Postal 1524, Lagoa Nova, Brasil
nielsen.tekla@gmail.com

RESUMO

Neste trabalho propomos um algoritmo para resolver o problema de separação cega de fontes não linear. Propomos uma função de custo baseado na teoria da informação. O algoritmo genético é utilizado para maximiza essa função. Foram realizadas simulações computacionais utilizando imagens e sinais em formas de ondas. E finalmente, são apresentados comparações com um método proposto na literatura.

PALAVRAS CHAVE. Rényi, RBF, Negentropia, não linear.

ABSTRACT

In this paper, we propose an algorithm to solve the problem of nonlinear blind source separation. We propose a cost function based on information theory. The genetic algorithm is used to maximize this function. Computer simulations were performed using images and signals in waveforms. Finally, comparisons are made with a method proposed in the literature.

KEYWORDS. Rényi, RBF, Negentropy, nonlinear.

1. Introdução

Os métodos convencionais para resolver problemas 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. Um dos típicos problemas investigado é a separação cega de fontes, motivada pelo problema conhecido como *cocktail party* para separação de sinais de áudio (Hyvärinen; Oja, 2000). Uma importante abordagem é quando as misturas são lineares. Neste caso, $\mathbf{s}_1(i), \mathbf{s}_2(i), \dots, \mathbf{s}_n(i)$ são misturados por uma matriz \mathbf{A} , gerando m misturas dado por $\mathbf{x}_1(i), \mathbf{x}_2(i), \dots, \mathbf{x}_m(i)$. Neste artigo, investigamos o caso $m = n$, de modo que cada saída $\mathbf{x}_j(i)$ é uma combinação linear de todas as entradas de $\mathbf{s}_j(i)$, dado por $\mathbf{s}_j(i) = \sum_k \mathbf{A}_{j,k} \mathbf{s}_k(i)$.

A Análise de Componentes Independente (ICA) é uma técnica de processamento de sinais muito utilizada para resolver problemas de Separação cega de fontes (BSS, acrônimo do inglês: *Blind Source Separation*). Seu objetivo é estimar os sinais de fontes originais usando apenas as misturas (Hyvärinen; Oja, 1999). Uma solução para este problema pode ser obtida se encontrarmos a inversa da matriz da mistura, ou seja,

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \quad (1)$$

onde \mathbf{y} seria igual a \mathbf{s} . Um princípio geral para estimar a matriz \mathbf{W} pode ser encontrado ao considerar que os sinais de fontes originais são estatisticamente independentes, uma vez que existem vários métodos matemáticos para encontrar os coeficientes da matriz de separação. Como exemplo, cita-se a maximização da não gaussianidade (Comon, 1994).

Podemos, ainda, estender o conceito de ICA linear para o caso não linear. Uma formulação geral para o caso de Análise de Componentes Independentes não lineares (*nonlinear independent component analysis* – NLICA) pode ser descrito da seguinte forma (Taleb; Jutten, 1999):

$$\mathbf{x} = \mathbf{F}(\mathbf{s}), \quad (2)$$

onde \mathbf{F} 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 encontrar \mathbf{G} (sistema separador), tal que:

$$\mathbf{y} = \mathbf{G}(\mathbf{x}) \quad (3)$$

Portanto, se $\mathbf{G} = \mathbf{F}^{-1}$ as fontes são perfeitamente recuperadas e finalmente \mathbf{s} será igual a \mathbf{y} .

Este artigo propõe um algoritmo genético que maximiza uma função custo, aplicado com sucesso a diversos problemas práticos de pesquisa operacional e otimização (Man et al. 1996). Para isso, propomos um algoritmo genético que utiliza um modelo de mistura não linear, utilizando Redes Neurais Artificiais (RNA) que são aplicados aos sinais de fonte. A partir da mistura, o algoritmo genético e a Negentropia de Rényi encontrará outra RNA que estima os sinais de fontes. As análises e os resultados constituíram de dados através de simulações computacionais.

2. Técnica Utilizada

2.1 Algoritmo Genético

A teoria dos Algoritmos Genéticos (AG) foi desenvolvida por Holland (1975). Ela é um método dinâmico (heurísticos) de busca baseado no mecanismo de seleção e evolução natural (Sankar, 2007), cujo objetivo principal é encontrar o indivíduo ótimo de uma população.

Basicamente, o algoritmo consiste em: inicializar uma população aleatoriamente; avaliar a população utilizando uma função matemática (Negentropia Rényi); utilizar os operadores genéticos (recombinação e mutação) para selecionar os mais capazes para formar uma nova população e; avaliar novamente a população até que algum critério de parada seja atendido.

No caso desta aplicação, os cromossomos são representados por uma cadeia binária que representa os valores dos parâmetros. O método de seleção deve ser semelhante ao processo de seleção natural, em que apenas os membros mais saudáveis da população são permitidos para sobreviver à próxima geração. Um processo de seleção utilizado é o método da roleta viciada, que funciona semelhante a uma roleta de cassino, em que cada cromossomo recebe uma parte proporcional à sua avaliação (Sankar, 2007).

2.2 Radial Basis Networks

As redes RBF são redes de alimentação diretas, consistindo tipicamente de três camadas: entrada, escondida e saída. Elas realizam 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 o vetor de referência.

No contexto de aproximações de funções, as redes RBF foram desenvolvidas para interpolação de dados em espaços multidimensionais. Uma típica função de base radial apropriado para interpolação é dado pela expressão (Haykin, 2001):

$$\varphi(\xi) \exp\left(-\frac{\xi^2}{2\sigma^2}\right) \quad (4)$$

A função do tipo gaussiana (equação 4) é comumente utilizada em muitas aplicações. Nela, a função do parâmetro σ corresponde ao desvio padrão da gaussiana, que ajusta o raio de influência de cada função. Em aplicações práticas, o raio das funções de base radial afeta as propriedades numéricas do algoritmo de aprendizagem. No entanto, a sua capacidade de aproximação não é afetado (Haykin, 2001).

Neste trabalho, elabora-se um projeto de um sistema misturador utilizando arquitetura de uma RNA do tipo RBF. Posteriormente, mostraremos que o aprendizado do sistema proposto resume-se 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.

2.3 Negentropia de Rényi

Nos problemas de BSS temos que escolher uma medida de independência ou função custo para ajustar os parâmetros do sistema separador. A Negentropia é uma medida baseada na entropia diferencial (medida quantitativa), conhecida como medida de não gaussianidade, ou seja, o quanto a distribuição de uma variável aleatória não se parece com uma gaussiana (Choi; Cichocki; Amari, 2000). Essa medida é empregada em métodos que utilizam Análise de Componentes Principais (Hyvrinen; Oja, 2000). A Negentropia para uma variável aleatória é definida como (Comon, 1994):

$$J(\mathbf{x}) \cong H(\mathbf{x}_g) - H(\mathbf{x}), \quad (5)$$

onde $H(\mathbf{x})$ é a entropia de \mathbf{x} e $H(\mathbf{x}_g)$ é uma variável aleatória gaussiana com mesma média e matriz de covariância da variável aleatória \mathbf{x} . O cálculo da Negentropia envolve o cálculo da entropia dos dados. Este cálculo não é trivial e pode ser uma fonte de problemas. No entanto, existem várias maneiras de se estimar a Negentropia de um sinal sem calcular a entropia dos dados diretamente, pois a maioria desses cálculos é baseada em estatística de ordem dos dados. Todavia, esses métodos não são adequados devido a problemas numéricos sobre os grandes expoentes envolvidos no cálculo dos momentos. Portanto, precisamos de uma expressão matemática para medir a entropia de dados reais e a entropia de uma variável aleatória gaussiana em conformidade com a definição da Negentropia dada pela equação (5).

Neste trabalho, usamos a entropia de Rényi (Rényi, 1970) de ordem 2,0 (entropia quadrática) com $\alpha = 2,0$ para o cálculo da entropia de uma variável aleatória gaussiana \mathbf{x} de dimensão d , com média μ e matriz de covariância Σ , que pode ser expresso por (Damasceno; Medeiros, 2011):

$$H(\mathbf{x}_g) = \frac{1}{2} (d \log(4\pi) + \log|\Sigma|) \quad (6)$$

O cálculo da entropia de Rényi de ordem 2,0 é bem mais simples que o cálculo para entropia de Shannon (Dória et al., 2011). Essa 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.

O método de Janelas de Parzen é também chamado de estimativa de kernel (Parzen, 1962). Principe *et al.* desenvolveram uma nova técnica para estimar a entropia Rényi (em vez de Shannon) a partir das amostras do conjunto de dados de um modo não paramétrico (Principe, 2010), como mostrado na equação (7). A técnica baseia-se numa estimativa não paramétrica da FDP (Função de distribuição de probabilidade) do conjunto de dados, utilizando, neste caso, janelas de Parzen.

$$H(\mathbf{x}) = -\log\left(\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N G(x_i - x_j, 2\sigma^2)\right) \quad (7)$$

Assim, a função objetivo do AG será a medida de independência dado pela Negentropia de Rényi, que pode ser estimada utilizando a equação (6) e (7) de acordo com a definição dada pela equação (5).

Na equação (7), σ^2 é o tamanho do kernel e 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 (Principe, 2010) ou utilizando alguma heurística como, por exemplo, a regra de Silverman (Silverman, 1986). A covariância sigma é estimada por:

$$\Sigma \approx \frac{1}{N} \sum_i (\mathbf{s}_i - \bar{\mathbf{s}})^t (\mathbf{s}_i - \bar{\mathbf{s}}), \quad (8)$$

onde $\bar{\mathbf{s}}$ é a média dos dados.

3. Sistema de mistura

3.1 Modelo baseado na RBF

Uma rede RBF é utilizada para gerar a mistura inicial, cujo objetivo do AG é encontrar outra RBF para reverter o que o RBF original fez. Logo, a população do AG são populações de RBF's em vez de matrizes de misturas, como é o caso do modelo linear. A Figura 1 ilustra o processo de mistura e separação utilizando esta abordagem. Neste caso, é necessário desenvolver um sistema de separação que seja capaz de reverter o processo do sistema não linear, utilizando-se a estrutura de uma RBF como um sistema de mistura. Para representar o modelo, usamos a seguinte expressão:

$$\mathbf{x} = \mathbf{W} \exp\left(-\frac{\|\mathbf{s}-\mathbf{c}\|^2}{2\sigma^2}\right) \quad (9)$$

O modelo de mistura pode ser representado por outras expressões, bem como a equação (10), (11) e (12). Posteriormente, descrevemos várias simulações em que o algoritmo converge utilizando diferente modelo de mistura.

$$\mathbf{x} = \mathbf{W} \frac{\|\mathbf{s}-\mathbf{c}\|}{\sigma^2} \log\left(\frac{\|\mathbf{s}-\mathbf{c}\|}{\sigma}\right) \quad (10)$$

$$\mathbf{x} = \mathbf{W} \frac{1}{\sqrt{\|\mathbf{s}-\mathbf{c}\|^2 + \sigma^2}} \quad (11)$$

$$\mathbf{x} = \mathbf{W}\sqrt{\|\mathbf{s} - \mathbf{C}\|^2 + \sigma^2} \quad (12)$$

Os sinais de fontes são submetidos à RBF, em que os centros e os pesos iniciais são gerados aleatoriamente pelo AG e é representado por \mathbf{C} e \mathbf{W} , respectivamente. Finalmente, uma descrição geral da estratégia não linear é descrita como se segue:

1. Os sinais de fonte são misturados usando a equação (9).
2. É especificado uma medida de independência (equação 5) de modo que seja máxima aplicada à \mathbf{s} , ou seja, onde $J\{\mathbf{F}(\mathbf{s})\}$ seja máxima.
3. Gera-se uma função \mathbf{G} qualquer de dimensão de \mathbf{F} .
4. Calcula-se a independência usando a equação (5).
5. Verifica se $\mathbf{G} = \mathbf{F}^{-1}$. Caso verdadeiro, então pode-se dizer que $J(\mathbf{G}(\mathbf{x})) = J\{\mathbf{G}(\mathbf{F}(\mathbf{s}))\} = J(\mathbf{s})$, em que $J(\mathbf{s})$ é máxima.
6. Volta para o passo 3, gerando outro \mathbf{G} até atingir um critério de parada. Assim, armazene o \mathbf{G} que resultou na maior independência.

O 3º passo segue as regras de Algoritmos Genéticos, e a medida de independência é a Negentropia de Rényi. As funções \mathbf{F} e \mathbf{G} são representadas pela RBF. Portanto, o \mathbf{G} que resultou em maior independência é o inverso da \mathbf{F} . Em implementações reais, as restrições impostas às funções \mathbf{F} e \mathbf{G} foram gerados aleatoriamente, utilizando uma função que gera aleatoriamente valores entre [-1, 1].

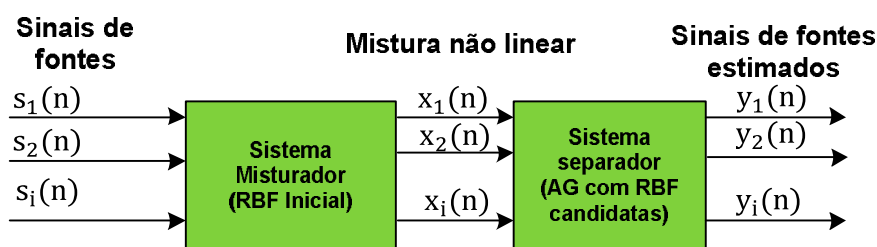


Figura 1. Um diagrama esquemático do problema da separação cega de fontes não linear.

3.2 Modelo PNL

O modelo PNL (*Post-nonlinear*) (Taleb; Jutten, 1999) assume que os sinais observados são gerados por misturas lineares seguidas por componentes não lineares, ou seja, $\mathbf{x} = \mathbf{F}(\mathbf{A}\mathbf{s})$. 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 a não linearidade no conjunto de sensores.

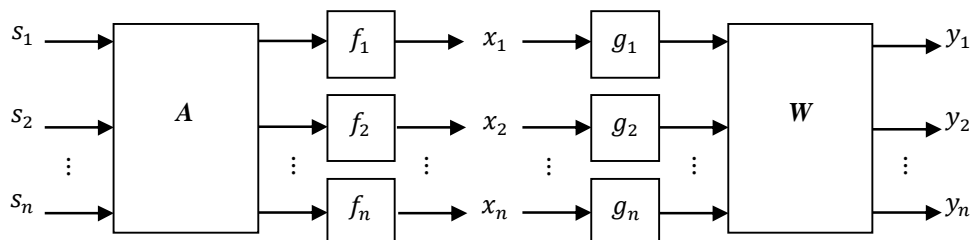


Figura 2. Diagrama esquemático do modelo de mistura PNL.

Como ilustrado na Figura 2, a recuperação dos sinais de fontes é feita por um modelo que compreende a inversa não linear e um estágio linear (matriz \mathbf{W}). A seção não linear (g_n) é geralmente estimada utilizando arquitetura de redes neurais como a *Multi-Layer-Perceptrons* (MLP) ou *Radial Base Function* (Taleb; Jutten, 1999).

Diferentes medidas de independência estatísticas são utilizadas para o treinamento. A parte linear pode ser executada por qualquer algoritmo ICA linear.

Vários procedimentos foram propostos na literatura a fim de melhorar o desempenho de treinamento da rede neural do modelo de mistura PNL. Tan et al. (2001) e Kai et al. (2006) utilizaram algoritmo genético para realizar uma busca global para encontrar o melhor conjunto de parâmetros que maximizem uma medida de independência.

Para ilustrar o modelo PNL, a Figura 3 representa dois sinais de fontes. Neste exemplo, utilizou-se 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 (Jutten; Karhunen, 2003). O algoritmo para estimação da parte linear (Matriz W) foi o P-ICA (Zhang; Wan, 2006).

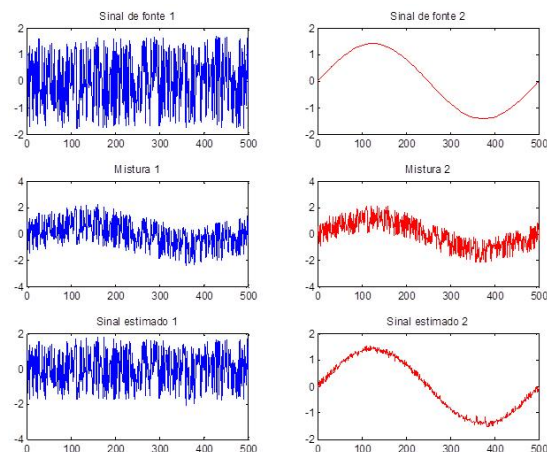


Figura 3. Demonstração de estimação de fontes usando abordagem PNL.

4. Método

Para testar cada candidato para uma possível solução, desfazemos “a ação de mistura” dos sinais com cada RBF candidata usando a equação (9) ou qualquer outra equação descrita na seção 3. Com os valores das misturas “descorrelacionados” calculamos a Negentropia de Rényi.

O critério de parada utilizado neste artigo é simples: utilizamos o número máximo de gerações. Para que os operadores genéticos representem adequadamente cada peso e centro (indivíduos do AG) representamos cada RBF candidata por uma sequência de números binários, fazendo uma grande cadeia de bits. A cadeia é composta por cada elemento da RBF expresso em binário de 16 bits. Os operadores genéticos utilizados neste experimento fazem basicamente duas operações: recombinação e mutação. Para o cruzamento, a cadeia binária de cada indivíduo é dividida em dois, em um ponto aleatório para efetuar a mistura. 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.

Definimos, também, uma taxa de recombinação de 85% da população. Essas medidas são aplicadas ao loop principal do programa até que o número máximo de gerações seja atingido. No final, o melhor indivíduo será utilizado como a solução do problema. Depois de atingir o critério de parada, pega-se o indivíduo com maior Negentropia na população e usa os pesos e os centros desse indivíduo para estimar as fontes independentes utilizando a equação (9).

5. Resultados

Testamos o método em muitas situações diferentes, tais como sinais de misturas de imagens e formas de ondas. Nos experimentos em que se utilizou imagem, para cada j -ésima imagem de Im_j escolhe-se um 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 formas de onda foram utilizados as seguintes funções para representar os sinais de fonte: $s(i) =$

$[sen(2i), sing(\sen(3i))]^t$. Esses mesmos sinais foram utilizados em (Damasceno N.C.; Medeiros A. M., 2010).

Na figura 4 temos os primeiros resultados usando uma onda quadrada e sinais senoidais como fontes, percebendo-se claramente que os resultados foram satisfatórios. Nestes foram obtidos com 10 centros e 10 pesos na RBF, o σ da função gaussiana foi 0,5 e a estimativa do kernel para a Negentropia de Rényi também foi 0,5. O número máximo de gerações foi de 30, onde fora utilizado uma população com 100 indivíduos. O tempo gasto pelo AG até sua convergência foi de aproximadamente 15 minutos e o modelo de mistura utilizado foi a equação (9).

Outro resultado foi apresentado utilizando duas imagens com 128x128 pixels. Elas foram convertidas para tons de cinza e apresentadas ao modelo de mistura utilizando a equação (10), com uma população de 200 indivíduos, e a largura do kernel da Negentropia foi de 0,5. O σ utilizado foi de 0,5. O número máximo de gerações obtidos no processo foi três, de modo que houve alto custo computacional, chegando aproximadamente duas horas e 30 minutos para apresentar os resultados na Figura 5.

Finalmente, a Figura 6 mostra dois sinais de forma de onda: dente de serra e seno. A Figura 6(a) mostra a estimativa de fontes, usando a equação (11) como o modelo mistura. Os mesmos sinais de fontes foram apresentados ao modelo de mistura ilustrado na Figura 6(b) e 6(c). Estes resultados foram utilizados com 10 pesos e 10 centros e o algoritmo convergiu com 10 gerações utilizando uma população de 100 indivíduos.

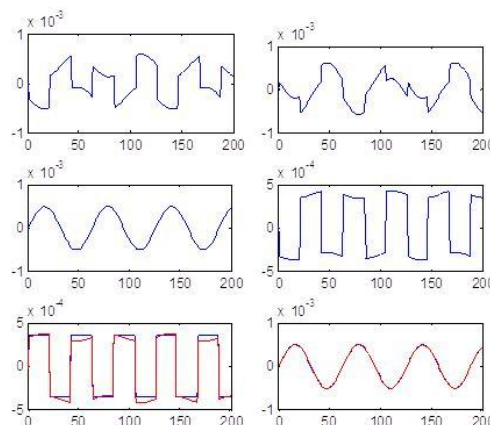


Figura 4. Separação de fontes. No topo: mostra as misturas dos sinais. No meio: os sinais estimados. Parte inferior: mostra os sinais estimados (vermelho) e os sinais originais (azul).

Para termos uma comparação entre o modelo de mistura proposto e o PNL. São apresentados na tabela 1 vários experimentos com diferentes números de amostras nos sinais de fontes. A comparação é dada pelo erro que é calculado como a soma dos mínimos quadrados para cada sinal de fonte versus a função estimada. Nestes experimentos utilizou-se os sinais de fontes que foram apresentado na Figura 4.

Tabela 1: Erro entre o algoritmo proposto (AG) e o proposto por (Taleb; Jutten, 1999).

Amostras	Erro PNL	Erro AG
50	4.4322e-010	4.0587e-006
100	5.79203e-012	6.0034e-009
200	1.30267e-015	8,7138e-012



Figura 5. Recuperação de imagens usando abordagem não linear para uma imagem com 128x128 pixels. Topo: fontes; Meio: as misturas; Inferior: imagem recuperada.

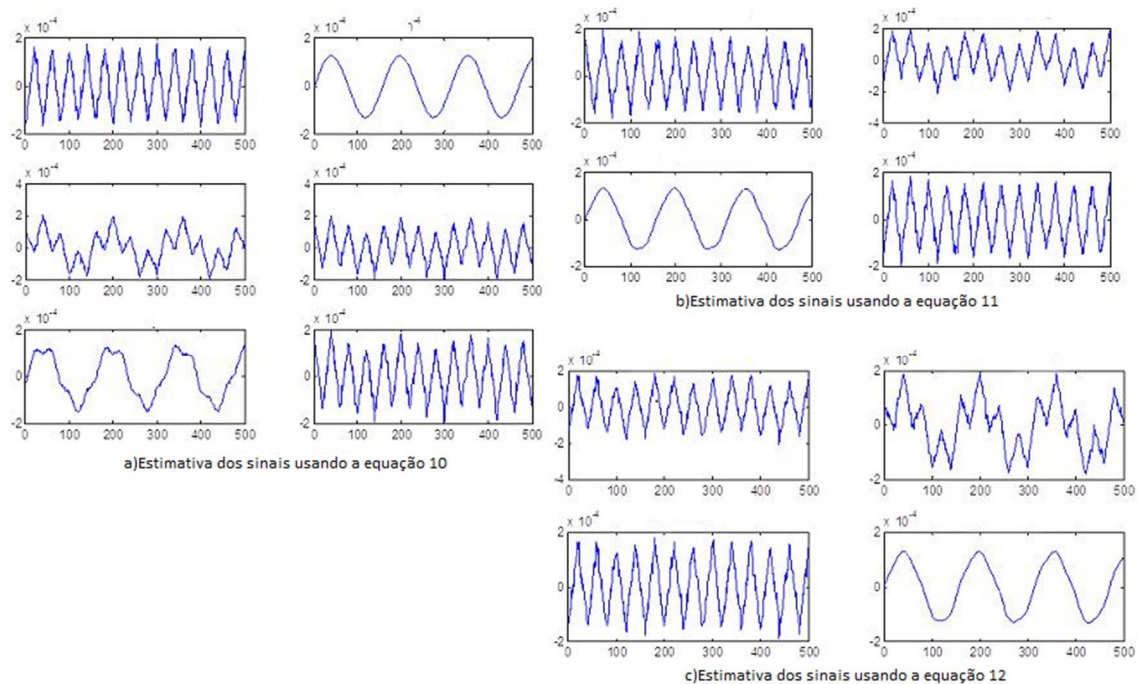


Figura 6. Problema utilizando três tipos de modelo de mistura e os mesmos sinais de fonte. A figura (a) mostra a forma de onda misturada usando a equação 10, Topo: mostra os sinais de fonte. Meio: mostra as misturas; Inferior: mostra os sinais estimados. Na figura (b), a forma de onda usando a equação 11, Topo: as misturas, Inferior: sinais estimados. A figura (c) mostra o experimento utilizando a equação 12, Na parte superior: as misturas; Na inferior: os sinais estimados.

6. Conclusão

A função de custo usando a Negentropia Rényi mostrou-se uma boa medida de independência, dada à qualidade dos resultados apresentados. Os melhores resultados foram encontrados empiricamente com uma RBF com 10 centros e 10 pesos. Usamos o parâmetro de kernel de suavização da entropia com 0,5. O σ da função de base radial variou entre 0,5 a 5. Embora nenhuma análise de desempenho envolvendo o tempo de execução foi discutida no artigo, o algoritmo genético está totalmente sintonizado para maximizar a Negentropia de Rényi e os resultados comparativos foram satisfatórios. Outra contribuição do artigo é a medida de independência, que por sua vez pode ser utilizado para medir clusters. Para concluir, gostaríamos de apresentar algumas perspectivas para trabalhos futuros como, por exemplo, uma análise com outros métodos existentes na literatura. Fazer a análise para a escolha do tamanho da kernel ou obter um novo modelo com base nos próprios dados a partir de misturas; Paralelizar o algoritmo para otimizar o custo computacional. Finalmente, tendo em conta que o sistema separador não linear deve ser capaz de inverter a ação do sistema de mistura, pretende-se demonstrar a prova da separabilidade do modelo proposto neste artigo.

Referências

- Choi S. e Cichocki A. e Amari S.** (2000), Flexible independent component analysis, J. VLSI Signal Process.
- Comon P.** (1994), Independent component analysis, a new concept?, Signal Process.
- Damasceno N. C. e Medeiros A. M.** (2011), Redes de Base Radiais e Algoritmos Genéticos aplicados a Separação Cega de fontes não lineares usando Negentropia de Rényi como medida de independência, JPC 2011, Pucallpa Peru.
- Damasceno N. C. e Medeiros A. M.** (2010), Blind Source Separation using genetic algorithms and negentropy as separation measure, XLII SBPO, Bento Gonsalves-RS 2010.
- Dória N. Adrião D. e Medeiros A. M. e Damasceno N. C.** (2011), Análise Comparativa entre o FastICA por maximização da Negentropia e AG-ICA usando Negentropia de Rényi, STIN 2011. Brasil, Rio grande do Sul.
- Haykin, S.** (2001), Redes Neurais: Princípios e pratica, 2rd ed, Porto Alegre: Bookman, Brasil.
- Holland J.** (1975), Adaptation in neural and Artificial systems, University of Michigan Press.
- Hyvärinen A. e Oja E.** (1999), Independent component analysis: A tutorial, Technical report.
- Hyvärinen A. e Oja E.** (2000), Independent Component Analysis: Algorithms and applications. Neural Networks.
- Jutten c., J. Karhunen,** (2003) 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.
- Kai S., W. Qi, D. Mingli,** (2006) 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.
- K. Zhang e Lai-Wan C.** (2006) ICA by PCA Approach: Relating Higher-Order Statistics to Second-Order Moments, ICA 2006, LNCS 3889, pag. 311–318.
- Man, K.F. e Tang, K.S. e Kwong, S.** (1996). Genetic algorithms: concepts and applications. IEEE Transactions on Industrial Electronics, v. 43, pag. 519–534.
- Parzen E.** (1962), On estimation of a probability function and mode, Ann. Math. Statist.
- Principe J.C.** (2010), Information Theoretic Learning, Rényi's Entropy and Kernel Perspectives, Springer New York.
- Rényi A.** (1970), Probability Theory, North-Holland, Amsterdam.
- Sankar K. Pal E.** (2007), Classification and Learning Using Genetic Algorithms, applications in Bioinformatics, Spring.

Silverman, B. W. (1986), Density Estimation for statistics and data analysis, Chapman Hall.

Taleb A. e Jutten C. (1999), Source separation in post-nonlinear mixtures, IEEE Tran. Signal Processing, no. 47, pag. 2807-2820.

Tan Y., J. Wang (2001), Nonlinear blind source separation using higher order statistics and a genetic algorithm, IEEE Transactions on Evolutionary Computation, vol. 5, no. 6, pag. 600-612.