

Um Estudo sobre Separação Cega de Fontes e Análise de Componentes Independentes

Marcelo Ramos Romano e Romis Attux

Resumo—Este trabalho apresenta um estudo de uma técnica de Separação Cega de Fontes chamada Análise de Componentes Independentes (ICA, do inglês *Independent Component Analysis*). Nesse artigo, serão apresentados alguns dos principais métodos de ICA: maximização da curtose, negentropia e verossimilhança e minimização da informação mútua. Esses métodos serão implementados e validados para sinais sintéticos no software MATLAB, sendo que também será feito um ensaio para sinais de áudio reais.

Palavras-Chave—Separação Cega de Fontes, Análise de Componentes Independentes, Processamento de Sinais.

Abstract—This work is a study of a technique of Blind Source Separation called Independent Component Analysis (ICA). Some of the most important ICA methods will be presented: maximization of kurtosis, negentropy and likelihood and minimization of mutual information. These methods will be implemented and validated for synthetic signals within the software MATLAB. A test with real audio signals will also be carried out.

Keywords—Blind Source Separation, Independent Component Analysis, Signal Processing.

I. INTRODUÇÃO

O problema de Separação Cega de Fontes pode ser descrito da seguinte forma: um vetor de sinais aleatórios, aqui denominamos fontes, $s[n]$, é aplicado à entrada de um sistema de múltiplas entradas e saídas (MIMO), gerando, como saída desse sistema, um novo vetor, dito de misturas, $x[n]$. Desejamos conhecer o vetor de fontes s , entretanto, só conhecemos o vetor de misturas x . Portanto, o problema de Separação Cega de Fontes consiste em estimar o vetor de fontes s a partir do vetor de misturas x , sem um conhecimento prévio do sistema.

Uma técnica que possibilita obter a separação das misturas é a Análise de Componentes Independentes (ICA, do inglês *Independent Component Analysis*). Para utilizarmos essa técnica, é necessário supor que todas as componentes do vetor s são independentes entre si. Além disso, é usual supor que as misturas são lineares e instantâneas, de tal forma que a relação entre os vetores s e x pode ser dada pela equação abaixo:

$$x = As \quad (1)$$

.Dessa forma, os métodos de ICA buscarão encontrar uma matriz inversa de mistura W que estime o vetor de fontes s . Nesse artigo serão apresentados os métodos de maximização da curtose, negentropia e verossimilhança. Também será apresentado um algoritmo evolutivo de minimização da informação mútua.

II. METODOLOGIA

A partir da equação (1) dada acima, pode-se observar que o vetor de fontes estimadas y é dado pela equação abaixo:

$$y = WVx = WVAs \quad (2)$$

Na equação acima, V é uma matriz de branqueamento. Essa matriz transforma o vetor x em um vetor branco z , facilitando os métodos de ICA, pois a matriz inversa de mistura, W , torna-se ortogonal.

A partir da equação (2) acima, podemos observar que, idealmente, cada componente do vetor y será uma combinação linear das componentes do vetor s . O que queremos é que apenas uma componente do vetor s seja capturada separadamente. Pelo Teorema Central do Limite, a soma de variáveis aleatórias independentes é sempre mais gaussiana do que elas isoladas [2]. Portanto, podemos recuperar as componentes independentes ao maximizar a não-gaussianidade do vetor y . Uma grandeza que indica o grau de gaussianidade de uma variável aleatória é a curtose, definida para sinais reais como:

$$\text{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2 \quad (3)$$

Se essa grandeza for zero, a variável considerada é usualmente gaussiana, se for menor ou maior que zero, ela será, respectivamente, subgaussiana ou supergaussiana.

Como estimaremos uma componente de cada vez, podemos usar medidas escalares de não-gaussianidade. Ao maximizar o módulo da curtose, maximizamos a não-gaussianidade da componente do vetor y considerada. Portanto, o algoritmo consistirá em encontrar cada linha w da matriz W , incrementando-a na direção do gradiente do módulo da equação (3) acima [3]:

$$\nabla \text{kurt} = 4\text{sign}(\text{kurt})[E\{z(w^T z)^3\} - 3w\|w\|^2] \quad (4)$$

$$\Delta w \propto \text{sign}(\text{kurt})[E\{z(w^T z)^3\} - 3w] \quad (5)$$

$$w = w + \Delta w \quad (6)$$

$$w = w/\|w\| \quad (7)$$

Outra grandeza que maximiza a não-gaussianidade é a negentropia, definida como a diferença entre a entropia de uma variável gaussiana e a entropia da variável aleatória considerada, tendo ambas a mesma variância. No entanto, será necessário estimar essa grandeza:

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \quad (8)$$

Na equação acima, $G(\cdot)$ é uma função arbitrária par e v é uma variável gaussiana de média nula e variância unitária.

Marcelo R. Romano, Departamento de Engenharia de Computação e Automação Industrial. Romis Attux, Departamento de Engenharia de Computação e Automação Industrial. Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas-SP, Brasil, E-mails: marcelorromano@gmail.com, attux@dca.fee.unicamp.br. Este trabalho de iniciação científica foi financiado pelo CNPq.

Tomamos $G(\cdot)$ como sendo $\log(\cosh(\cdot))$. Tomando o gradiente da equação acima, temos:

$$J(\mathbf{w}^T \mathbf{z}) \propto [E\{G(y)\} - E\{G(v)\}]E\{zg(\mathbf{w}^T \mathbf{z})\} \quad (9)$$

em que $g(\cdot)$ é a derivada da função $G(\cdot)$. Dessa forma, teríamos que incrementar \mathbf{w} na direção do gradiente, de modo similar ao algoritmo da curtose.

O algoritmo de máxima verossimilhança consiste em encontrar a matriz de mistura mais verossímil de ter gerado cada uma das amostras do vetor de misturas, maximizando a probabilidade condicional dada pela equação abaixo:

$$L(\mathbf{W}) = p(\mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[N] | \mathbf{W}) = \prod_{k=1}^N p(\mathbf{x}[k] | \mathbf{W}) \quad (10)$$

Com isso, constrói-se um algoritmo que incrementa a matriz \mathbf{W} na direção do gradiente da verossimilhança, que também será estimado, dado pela equação (11) abaixo. Comumente utiliza-se a função $g(\cdot)$ como sendo $\tanh(\cdot)$, havendo variações para variáveis supergaussianas ou subgaussianas.

$$\frac{1}{N} \nabla \log L = E\{g(\mathbf{W}\mathbf{x})\mathbf{x}^T\} + [\mathbf{W}^T]^{-1} \quad (11)$$

Por fim, desenvolveu-se um algoritmo evolutivo, que buscou minimizar a informação mútua entre as fontes estimadas: o CLONALG [1]. Esse algoritmo realiza uma busca estocástica e populacional, baseada na aplicação de (12) a um conjunto de clones de cada solução \mathbf{W} , com subsequente seleção determinística. Em (12), $N(0,1)$ representa uma matriz de valores gaussianos de média zero e variância unitária e f é o valor do fitness (custo) de cada indivíduo, sendo este um mapeamento inverso da informação mútua dada em (13).

Pode-se notar que, em (12), a mutação é menor conforme o fitness da célula for maior. O parâmetro β é arbitrário e sua finalidade é controlar a amplitude da mutação. Em (13), a função $H(\cdot)$ é a entropia de um vetor ou variável aleatória.

$$\mathbf{W}' = \mathbf{W} + \alpha N(0,1); \alpha = \beta^{-1} \exp(-f^*) \quad (12)$$

$$I(y_1, y_2, \dots, y_n) = \sum_{i=1}^n H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{W}| \quad (13)$$

III. RESULTADOS

Os algoritmos descritos na seção anterior foram implementados no software MATLAB. As figuras abaixo apresentam a simulação dos algoritmos baseados no gradiente, com passo unitário, mostrando a evolução do erro quadrático médio entre o sinal de fontes e o sinal estimado de fontes. Na Fig. 1, a figura da esquerda utilizou um vetor de fontes subgaussianas e a figura da direita, supergaussianas.

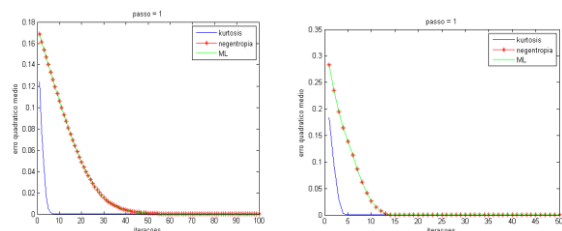


Figura 1 – Evolução do Erro Quadrático Médio

A Fig. 2 ilustra o algoritmo evolutivo de minimização da informação mútua, mostrando a evolução do fitness médio e do fitness máximo. O fitness médio oscila, pois, a cada 50 iterações, suprimimos algumas células e inserimos outras aleatórias no lugar das antigas. O erro quadrático médio final foi da ordem de 10^{-8} .

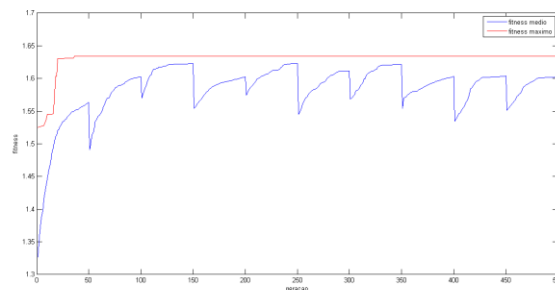


Figura 2 – Evolução do Fitness Máximo e do Fitness Médio

Com relação à simulação para sinais de áudio reais, a Fig. 3 (esquerda) ilustra os sinais fontes e a da direita os sinais de mistura. Os sinais fontes utilizados foram um sinal de voz e um sinal de palmas.

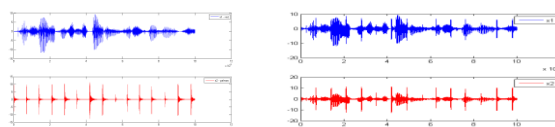


Figura 3 – Fontes e Misturas

Na Fig. 4, encontram-se as comparações entre os sinais estimados e os sinais fontes:

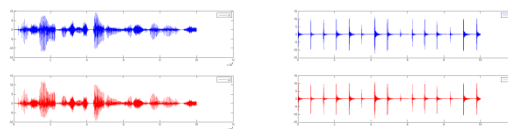


Figura 4 – Fontes e Sinais Estimados

IV. CONCLUSÕES

Diante dos resultados apresentados, podemos observar que as técnicas de ICA se mostraram eficientes, até mesmo para sinais reais. Com isso, podemos também concluir que as hipóteses apresentadas de independência e linearidade se mostraram razoáveis.

REFERÊNCIAS

[1] L. N. de Castro, J.I. Timmis, *Artificial immune systems: A new computational intelligence approach*, Springer-Verlag, 2002.
 [2] P. Comon, "Independent Component Analysis: a New Concept?", *Signal Processing*, Vol. 36, No. 3, pp. 287 – 314, 1994.
 [3] A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component Analysis*, Wiley, 2001.