UNIVERSIDADE DE SÃO PAULO

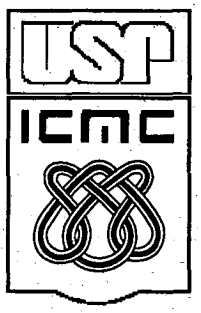
Instituto de Ciências Matemáticas e de Computação

Modelo de Misturas ICA Aperfeiçoado Para Classificação Não Supervisionada

Patrícia Rufino Oliveira
Roseli Aparecida Francelin Romero

Nº 82

NOTAS



São Carlos - SP

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2577

Modelo de Misturas ICA Aperfeiçoado Para Classificação Não Supervisionada

Patrícia Rufino Oliveira
Roseli Aparecida Francelin Romero

Nº 82

NOTAS

Série Computação

São Carlos – SP
Set./2004

Modelo de Misturas ICA Aperfeiçoado para Classificação Não Supervisionada

Patrícia Rufino Oliveira
Roseli Aparecida Francelin Romero

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {rufino, rafrance}@icmc.usp.br

Resumo

O Modelo de Misturas ICA foi originalmente proposto para realizar a classificação não supervisionada de dados modelados como uma mistura de classes descritas por combinações lineares de densidades independentes não Gaussianas. Uma vez que o algoritmo de aprendizagem original é baseado em uma técnica de otimização de gradiente, notou-se que o seu desempenho é afetado, entre outros fatores, por algumas limitações associadas a esse tipo de abordagem.

Neste trabalho, melhorias relativas a aspectos de implementação e modelagem são incorporadas ao Modelo de Misturas ICA, com o objetivo de atingir melhores resultados de classificação. Nesse sentido, algumas modificações nas regras de aprendizado e modelagem do sistema são propostas, baseando-se em uma abordagem para maximização da informação transferida em uma rede neural de entradas não lineares.

Além disso, o método de otimização de Levenberg-Marquardt, que utiliza a informação da segunda derivada da função objetivo, é incorporada ao algoritmo de aprendizagem para garantir e melhorar a convergência do modelo. Resultados comparativos experimentais obtidos pelo modelo aperfeiçoado e pelo original são apresentados para mostrar que as modificações propostas podem melhorar significativamente o desempenho da classificação, considerando dados simulados (gerados aleatoriamente) e o bem conhecido conjunto de dados Iris. Para avaliar o desempenho dos modelos em uma aplicação de processamento de imagens, resultados de segmentação, envolvendo imagens de diferentes domínios também são apresentados.

Palavras-Chave: Análise de Componentes Independentes, Métodos de Otimização, Classificação Não Supervisionada, Modelos de Misturas, Segmentação de Imagens.

Setembro 2004

Sumário

1 Introdução 1

2 Análise de Componentes Independentes 3
- 2.1 Motivação . . . 3
- 2.2 ICA em algumas aplicações reais . . . 5
- 2.3 Definição da Técnica ICA . . . 6
- 2.4 Suposições e Restrições da ICA . . . 7
- 2.5 Ambiguidades do modelo ICA . . . 8
- 2.6 Decorrelação, Branqueamento e Independência . . . 9

3 Abordagem ICA baseada na Teoria da Informação 11
- 3.1 Formulação do problema ICA utilizando conceitos de teoria de informação 12
- 3.2 Abordagem Infomax para ICA . . . 13
  - 3.2.1 Regras de Aprendizado ICA para uma Rede Neural com uma Entrada e uma Saída . . . 15
- 3.3 Regras de Aprendizado ICA para uma Rede Neural com N Entradas e N Saídas . . . 18
- 3.4 Equivalência entre Maximização de Informação e Estimação por Máxima Verossimilhança . . . 19

4 Modelo de Misturas ICA 21
- 4.1 Derivação do Algoritmo ICAMM . . . 23
  - 4.1.1 Estimação das Matrizes de Bases . . . 24
  - 4.1.2 Regras de Aprendizado do Modelo ICAMM . . . 28
  - 4.1.3 Estimação dos Vetores de Bias . . . 28

5 Modelo de Misturas ICA Aperfeiçoado (EICAMM) 29
- 5.1 Reformulação do Modelo de Classes . . . 29
- 5.2 Regra de Aprendizado para os Termos de *Bias* . . . 30
- 5.3 As matrizes de bases no EICAMM são ortogonais . . . 30
- 5.4 O modelo EICAMM utiliza informações de segunda derivada . . . 31
  - 5.4.1 Método de Newton . . . 31
  - 5.4.2 Método de Levenberg-Marquardt . . . 32

6 Resultados Experimentais 32

7 Conclusões 35

A Derivação das Regras de Aprendizado ICA 36
- A.1 Para uma Rede Neural com uma Entrada e uma Saída . . . 36
  - A.1.1 Função de transferência logística . . . 36
- A.2 Função de transferência tangente hiperbólica . . . 38
- A.3 Para uma Rede Neural com N Entradas e N Saídas . . . 39

Referências 44

Lista de Figuras

1 Exemplo de sinais originais. 4
2 Exemplos de misturas de sinais observadas. 4
3 Estimativas dos sinais de fontes originais 5
4 Exemplos de funções de densidade de probabilidade supergaussianas e subgaussianas 8
5 A função logística e sua derivada 15
6 Fluxo ótimo em neurônios sigmoidais 17
7 Exemplo de dados simulados com duas classes Laplacianas. 33
8 Exemplo de dados simulados com duas classes Laplacianas e uma classe uniforme. 34

Lista de Tabelas

1 Resultados de Classificação para Dados Simulados – ICAMM. 33
2 Resultados de classificação para dados simulados – EICAMM. 34
3 Resultados de classificação para o conjunto de dados Iris. 34

1 Introdução

Classificação e agrupamento de padrões são problemas encontrados com frequência em vários campos da ciência, como biologia, medicina, visão computacional e inteligência artificial (Jain, Duin, and Mao 2000). Um problema de classificação de padrões pode ser resolvido computacionalmente por algoritmos supervisionados ou não supervisionados (Watanabe 1985). Em uma classificação supervisionada, cada padrão do conjunto de dados é identificado como sendo um membro de uma classe predefinida. Já um algoritmo de classificação não supervisionado associa cada padrão a uma classe baseando-se somente em estatísticas, sem nenhum conhecimento sobre as classes de treinamento.

Uma abordagem para classificação não supervisionada é baseada em modelos de misturas (ver, por exemplo, (Duda and Hart 1973; Bishop 1994)), nos quais a distribuição dos dados é modelada como uma soma ponderada de densidades condicionais. Por exemplo, no caso de um modelo de misturas Gaussianas (também chamadas de normais), assume-se que os dados em cada classe têm uma distribuição Gaussiana multivariada. Entretanto, essa suposição implica que o modelo de misturas Gaussianas explora somente estatísticas de segunda ordem (médias e covariâncias) dos dados observados para estimar as densidades *a posteriori*.

Nos últimos anos, a técnica da Análise de Componentes Independentes (ICA)[1] tem sido bastante aplicada em várias áreas da ciência, devido ao fato de que esse método explora estatísticas de ordem mais altas em um conjunto de dados (Hyvärinen, Karhunen, and Oja 2001), (Comon 1994), (Bell and Sejnowski 1995). De fato, a técnica ICA é uma generalização da Análise de Componentes Principais (PCA)[2] (ver, por exemplo (Johnson and Wichern 1998), (Manly 1986) e (Oliveira 1997)), uma vez que o método ICA transforma linearmente as variáveis originais em componentes independentes, ao invés de somente não correlacionadas, como no caso da PCA. [3]

Variáveis aleatórias $y_1, y_2, \ldots, y_n$ são consideradas independentes se a informação referente ao valor de y_i não fornece nenhuma informação sobre os valores de y_j, para $i \neq j$. O conceito de independência pode ser definido por meio de densidades de probabilidade. Denotando por $p(y_1, y_2, \ldots, y_n)$ a função de densidade de probabilidade (fpd) conjunta de y_i e por $p_i(y_i)$ a fdp marginal de y_i, isto é, a fdp individual de y_i, não interessando os valores das demais variáveis aleatórias, pode-se dizer que as variáveis y_i são independentes se, e somente se, a fdp conjunta for fatorável da seguinte forma:

$$p(y_1, y_2, \ldots, y_n) = p_1(y_1) \cdot p_2(y_2) \cdot \ldots \cdot p_n(y_n). \tag{1}$$

O Modelo de Misturas ICA (ICAMM)[4] foi proposto por Lee e colaboradores em (Lee, Lewicki, and Sejnowski 2000), com o objetivo de superar uma limitação da técnica ICA,

[1] do original, em inglês, Independent Component Analysis.

[2] do original, em inglês, Principal Component Analysis.

[3] Recomenda-se que o leitor tenha algum conhecimento da técnica PCA para que possa melhor compreender o conteúdo deste trabalho.

[4] do original, em inglês, ICA Mixture Model.

que consiste na suposição de que as fontes geradoras dos dados são independentes. Em tal abordagem, essa suposição foi relaxada utilizando-se o conceito de modelo de misturas.

Cada classe no ICAMM é descrita por uma combinação linear de fontes independentes com densidades não Gaussianas. O algoritmo encontra as componentes independentes e a matriz de bases para cada classe utilizando o algoritmo de aprendizagem infomax estendido (Lee, Girolami, and Sejnowski 1999) e também calcula a probabilidade de pertinência de classe para cada padrão do conjunto de dados. As regras de aprendizado para o ICAMM foram derivadas utilizando o método de otimização do gradiente ascendente para maximizar a função de log-verossimilhança dos dados.

Apesar de algumas características promissoras do ICAMM terem sido reportadas em (Lee, Lewicki, and Sejnowski 2000), no presente trabalho, apesar das inúmeras tentativas, não foi obtido sucesso na reprodução dos resultados experimentais obtidos por Lee e colaboradores. Ao invés do bom desempenho de classificação do método, relatado no trabalho original, foram observados em experimentos com dados simulados e com o conjunto de dados de flores Iris, uma convergência muito lenta e resultados de classificação insatisfatórios. Em dois artigos encontrados na literatura (Ridder, Kittler, and Duin 2000), (Shah, Arora, Robila, and Varshney 2002), o ICAMM também foi aplicado sem grandes resultados. Em (Ridder, Kittler, and Duin 2000), o ICAMM teve um bom desempenho para dados 2-D simulados, porém a vantagem de utilizar o método para segmentar imagens não foi provada conclusivamente. Em (Shah, Arora, Robila, and Varshney 2002), algumas técnicas de extração de características foram consideradas como etapas de pré-processamento para reduzir a dimensionalidade dos dados, tentando, assim, aumentar a eficiência do ICAMM. Apesar das precisões médias de classificação obtidas por essa abordagem terem sido maiores do que aquelas obtidas pelo método k-médias (MacQueen 1967), os autores encontraram algumas limitações e suposições que comprometem o uso do ICAMM em classificação de dados de sensoreamento remoto.

Com o objetivo de melhorar o desempenho do ICAMM, o presente trabalho introduz o Modelo de Misturas ICA Aperfeiçoado (EICAMM)[5], que implementa algumas modificações no ICAMM originalmente proposto. Assim como no ICAMM original, algumas modificações também são baseadas na abordagem de maximização de informação para separação cega de fontes e deconvolução cega de fontes, proposta por Bell e Sejnowski em (Bell and Sejnowski 1995). Nesse trabalho, foi introduzido um novo algoritmo auto-organizante que maximiza a informação transferida em uma rede de unidades não lineares. As não linearidades na função de transferência são capazes de captar momentos de ordem mais altas (Hyvärinen, Karhunen, and Oja 2001) das distribuições de entrada e realizar redução de redundância entre as unidades na representação de saída.

Um dos problemas associados ao ICAMM está relacionado ao fato que o seu algoritmo de aprendizagem é baseado em uma técnica de otimização de gradiente. Portanto, foi observado que o desempenho do método é afetado, entre outros fatores, por algumas limitações conhecidas associadas a esse tipo de abordagem. A técnica do gradiente ascendente (ou descendente) tornou-se famosa na literatura como um método padrão para treinamento em redes neurais (Masters 1995). O uso difundido dessa técnica está relacionado principalmente a sua propriedade mais importante: pode ser provado matematicamente

[5]do original, em inglês, Enhanced ICA Mixture Model.

que esse algoritmo sempre irá convergir para um mínimo local da função objetivo, apesar de um grande número de iterações ser frequentemente necessário. Além disso, um outro problema importante a ser resolvido é que não existe garantia que o método não ficará preso a um mínimo local.

Com o objetivo de amenizar este problema, algumas melhorias foram adicionadas ao ICAMM original por meio da incorporação de algumas características de métodos de otimização não linear. Nesse sentido, o método de Levenberg-Marquardt (ver, por exemplo, (Masters 1995)) foi incorporado ao algoritmo de aprendizagem para garantir e melhorar a convergência do modelo.

Para mostrar a eficiência do modelo proposto, um estudo comparativo apresentado neste trabalho discute os resultados obtidos pelos modelos EICAMM e ICAMM. A partir dessa discussão, pode-se notar que as modificações propostas neste trabalho podem melhorar significativamente o desempenho de classificação do método, considerando-se conjuntos de dados simulados (gerados aleatoriamente) e o bem conhecido conjunto de dados de flores Iris.

Este trabalho está organizado como segue. Na Seção 2, conceitos básicos da técnica ICA são apresentados. O Modelo de Misturas ICA original é descrito na Seção 3. Na Seção 4, o Modelo de Misturas ICA Aperfeiçoado é apresentado. Os resultados experimentais deste trabalho são apresentados e discutidos na Seção 5. Finalmente, na Seção 6, conclusões e trabalhos futuros são apresentados.

2 Análise de Componentes Independentes

Em contraste a transformações baseadas em medidas de correlação, como a PCA, a técnica ICA não somente transforma os sinais em novos sinais não correlacionados entre si (utilizando estatísticas de segunda ordem), mas também reduz dependências estatísticas de mais altas ordens, na tentativa de tornar os sinais os mais independentes uns dos outros quanto possível.

2.1 Motivação

Uma motivação bastante utilizada para ilustrar o funcionamento do método ICA é considerar uma situação na qual existem vários sinais emitidos por fontes ou objetos físicos diferentes (Hyvärinen, Karhunen, and Oja 2001). Tais fontes poderiam ser, por exemplo, diferentes áreas do cérebro emitindo sinais elétricos; pessoas conversando em um ambiente e emitindo, portanto, sinais de fala; ou telefones móveis emitindo ondas de rádio. Deve-se assumir, ainda, que existem vários sensores ou receptores. Esses sensores estão em diferentes posições, de modo que cada sinal registrado é uma soma ponderada (isto é, uma combinação linear) dos diferentes sinais de fontes originais.

Como exemplo, pode-se imaginar a seguinte situação: dois interlocutores estão conversando em um ambiente no qual estão instalados dois microfones em diferentes localizações. Os microfones 1 e 2 fornecem sinais registrados em instantes de tempo e que poderiam ser

denotados por $x_1(t)$ e $x_2(t)$, respectivamente. Neste caso, x_1 e x_2 são as amplitudes do sinal e t o instante de tempo. Cada um desses sinais registrados é uma soma ponderada dos sinais de fala $s_1(t)$ e $s_2(t)$ emitidos pelos dois interlocutores. Dessa forma, os sinais $x_1(t)$ e $x_2(t)$ podem ser expressos pelas seguintes equações lineares:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \tag{2}$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t), \tag{3}$$

em que a_{11}, a_{12}, a_{21} e a_{22} são alguns parâmetros que dependem das distâncias entre os microfones (sensores) e os interlocutores (fontes). Assume-se que esses parâmetros são desconhecidos, uma vez que não é possível saber o valor de $a_{ij}(i, j = 1, 2)$, sem conhecer todas as propriedades do sistema físico, o que, em geral, é uma tarefa difícil. O objetivo é, nesse caso, estimar os dois sinais originais $s_1(t)$ e $s_2(t)$, usando somente os sinais registrados $x_1(t)$ e $x_2(t)$. Esse problema específico é referenciado na literatura como problema do *cocktail party*, ou, de forma mais geral, problema da separação cega de sinais de fontes (BSS) [6].

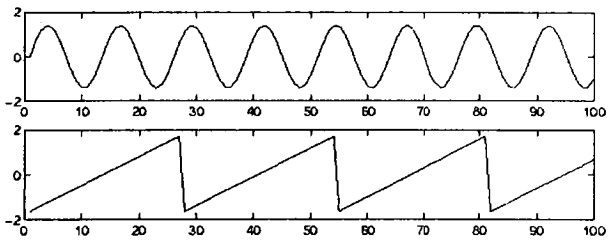


Figura 1: Exemplo de sinais originais.

Para ilustrar esse problema, pode-se considerar as formas de ondas das Figuras 1 e 2 (Hyvärinen, Karhunen, and Oja 2001), apesar destas não corresponderem a sinais realistas. Os sinais originais $s_1(t)$ e $s_2(t)$ poderiam se apresentar como na Figura 1 e os sinais misturados $x_1(t)$ e $x_2(t)$ poderiam ser como aqueles vistos na Figura 2. O problema resume-se, então, a recuperar os dados da Figura 1 usando somente os dados presentes na Figura 2.

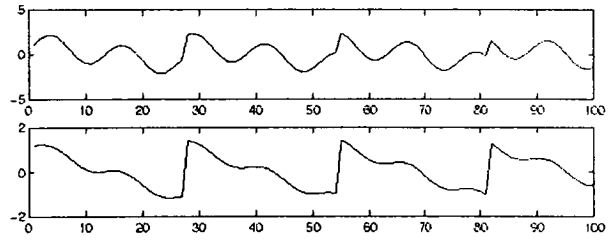


Figura 2: Exemplos de misturas de sinais observadas.

[6] do original, em inglês, Blind Source Separation.

Uma abordagem para solucionar esse problema seria utilizar alguma informação sobre as propriedades estatísticas dos sinais $s_i(t)$ para estimar os parâmetros a_{ij}. No caso da ICA, é suficiente assumir que $s_1(t)$ e $s_2(t)$, em qualquer instante t, são estatisticamente independentes. Essa suposição não é realista em muitos casos, e não é necessário que seja exatamente verdadeira na prática (Hyvärinen, Karhunen, and Oja 2001). A técnica ICA pode ser utilizada para estimar os parâmetros a_{ij}, baseando-se na informação de independência, o que permite separar os dois sinais de fontes originais $s_1(t)$ e $s_2(t)$ a partir de suas misturas $x_1(t)$ e $x_2(t)$. A Figura 3 (Hyvärinen, Karhunen, and Oja 2001) fornece os dois sinais estimados pelo método ICA. Como pode ser visto, estes sinais estão muito próximos dos sinais de fontes originais (a inversão dos sinais não é importante nesse caso).

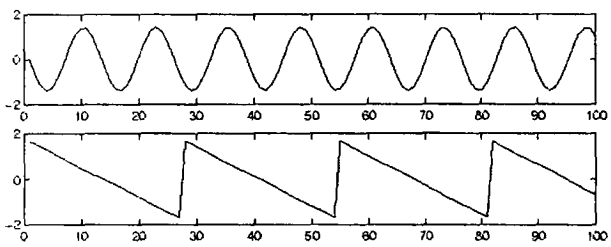


Figura 3: As estimativas dos sinais de fontes originais, calculadas usando somente os sinais observados na Figura 2.

2.2 ICA em algumas aplicações reais

Uma aplicação real importante da ICA diz respeito à análise de registros de atividades cerebrais, como por exemplo, registros dados por eletroencefalograma (EEG) e magnetoencefalograma (MEG). Esses dados consistem em registros de potenciais elétricos ou magnéticos medidos em diferentes localizações do escalpo [7] ou do cérebro. Neste caso, pode-se assumir que os dados são gerados pela mistura de componentes relativas a atividades cerebrais e musculares. Essa situação é bem similar ao problema do *cocktail party*: deseja-se encontrar as componentes originais da atividade cerebral, mas somente é possível observar as misturas de componentes. A ICA pode, então, revelar informações interessantes sobre a atividade cerebral, utilizando as componentes independentes encontradas pelo método. Dentre vários trabalhos relevantes envolvendo ICA e esse tipo de aplicação, pode-se citar (Makeig, Jung, Bell, Ghahramani, and Sejnowski 1997), (Vigário, Särelä, Jousmäki, Hämäläinen, and Oja 2000) e (Jung, Makeig, McKeown, Bell, Lee, and Sejnowski 2001).

Também nas ciências sociais, a ICA tem sido considerada uma ferramenta importante para a descoberta de fatores independentes, por exemplo, em problemas de ecomometria (Back and Weigend 1997), (Oja, Kiviluoto, and Mãlãroiu 2000).

Uma outra aplicação importante da ICA refere-se à utilização desta técnica como método para extração de características relevantes em um conjunto de dados. As carac-

[7] Pele que cobre o topo da cabeça humana

terísticas encontradas nesse tipo de aplicação da ICA podem ser usadas para representar imagens (Bell and Sejnowski 1997), (Hyvärinen, Oja, Hoyer, and Hurri 1998), (Olshausen and Field 1996), dados de áudio (Bell and Sejnowski 1996) e outros tipos de dados em tarefas como compressão e supressão de ruídos.

2.3 Definição da Técnica ICA

Para definir rigorosamente a técnica ICA (Comon 1994), (Jutten and Herault 1991), pode-se utilizar o modelo estatístico de variáveis latentes, que são variáveis que não podem ser diretamente observadas. Assume-se que são observadas n variáveis aleatórias $x_1, x_2, \ldots, x_n$ modeladas como combinações lineares de n variáveis aleatórias latentes $s_1, s_2, \ldots, s_n$:

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \ldots + a_{in}s_n, \qquad \text{para todo } i = 1, \ldots, n, \tag{4}$$

em que $a_{i,j}, i, j = 1, \ldots, n$ são alguns coeficientes reais. Por definição, as variáveis s_i são estatisticamente mutuamente independentes.

O modelo ICA é, portanto, um modelo generativo, isto é, que descreve como os dados são gerados por um processo de mistura das variáveis s_j, denominadas de componentes independentes. Os únicos termos observáveis no modelo ICA são as variáveis aleatórias x_i, uma vez que as variáveis s_j são latentes e os coeficientes a_{ij} são desconhecidos. Portanto, tanto as componentes independentes s_j quanto os coeficientes a_{ij} devem ser estimados utilizando somente os valores observados das variáveis aleatórias x_i.

O índice relativo ao instante de tempo t foi omitido na Equação(4), uma vez que, no modelo ICA, assume-se que cada mistura x_i, assim como cada componente independente s_j, são variáveis aleatórias, deixando de ser consideradas como sinais em um determinado instante de tempo.

Utilizando a notação vetorial, denota-se por $\mathbf{x}$ o vetor aleatório cujos elementos são as combinações lineares $x_1, x_2, \ldots, x_n$, e por $\mathbf{s}$ o vetor aleatório com os elementos $s_1, s_2, \ldots, s_n$ e por $\mathbf{A}$ a matriz com os coeficientes $a_{i,j}$. Nesse caso, convenciona-se que todos os vetores do modelo são vetores-coluna; portanto $\mathbf{x}^T$, o vetor transposto de $\mathbf{x}$, é um vetor-linha.

Dessa forma, na notação vetorial, o modelo ICA pode ser escrito como:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{5}$$

ou, de forma equivalente,

$$\mathbf{x} = \sum_{i=1}^{n} \mathbf{a}_i s_i \tag{6}$$

em que os vetores $\mathbf{a}_i$ são as colunas da matriz $\mathbf{A}$.

No decorrer deste trabalho, assume-se que o número de vetores observados é igual ao número de sinais de fontes, embora em várias aplicações seja mais realístico assumir que esses números sejam diferentes (ver, por exemplo, (Olshausen and Field 1997), (Lewicki and Sejnowski 2000)).

2.4 Suposições e Restrições da ICA

Para assegurar que o modelo ICA básico apresentado na Seção(2.3) possa ser estimado, algumas suposições e restrições, enumeradas a seguir, são necessárias.

1. Deve-se assumir que as componentes independentes s_j são estatisticamente independentes. Esse princípio é a base da ICA.

2. As componentes independentes devem apresentar distribuições não gaussianas.

   Os cumulantes de alta ordem são iguais a zero para distribuições gaussianas, porém informações de ordem mais alta são essenciais para a estimação do modelo ICA (ver seções 7.4 e 7.5 em (Hyvärinen, Karhunen, and Oja 2001)). Portanto, o procedimento para a técnica ICA fica praticamente impossível se as variáveis observadas tiverem distribuições gaussianas.

   Uma medida de não normalidade frequentemente utilizada na estimação ICA é a medida de kurtosis, dada pela seguinte equação:

   $$\text{kurt}(y) = E\{y^4\} - 3, \tag{7}$$

   em que y é uma variável aleatória. Uma variável gaussiana apresenta um valor de kurtosis igual a zero. Variáveis com kurtosis positivas possuem uma distribuição supergaussiana, como é o caso da distribuição de Laplace (ou dupla exponencial). Já um valor de kurtosis negativo implica em uma distribuição subgaussiana, como no caso da distribuição uniforme. Exemplos dessas distribuições podem ser vistos na Figura (4).

3. Para simplificar o modelo, pode-se assumir que a matriz de coeficientes $\mathbf{A}$ é quadrada.

   Em outras palavras, o número de componentes independentes é igual ao número de variáveis observadas. Embora, em alguns casos, essa suposição possa ser relaxada (ver, por exemplo, (Olshausen and Field 1997), (Lewicki and Sejnowski 2000)), é muito importante considerá-la, uma vez que a mesma simplifica bastante o modelo. Partindo dessa suposição, e também assumindo que $\mathbf{A}$ é inversível, depois de estimar a matriz $\mathbf{A}$, é possível computar a sua inversa, $\mathbf{W}$, e obter as componentes independentes por meio de:

   $$\mathbf{s} = \mathbf{W}\mathbf{x}. \tag{8}$$

   Dessa forma, também deve-se assumir que a matriz de coeficientes $\mathbf{A}$ é inversível.

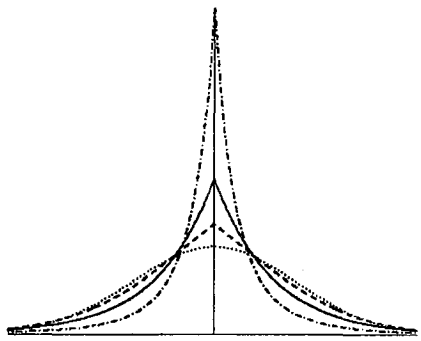


Figura 4: Exemplos de funções de densidade de probabilidade supergaussianas e subgaussianas. Linha sólida: densidade Laplaciana. Linha tracejada: densidade moderadamente supergaussiana. Linha pontilhada-tracejada: densidade fortemente supergaussiana. Linha pontilhada: densidade gaussiana.

4. Sem perda de generalidade, pode-se assumir que as variáveis observadas e as componentes independentes possuem média igual a zero. Essa suposição é feita pois simplifica bastante a teoria e os algoritmos derivados da técnica ICA.

   Se essa suposição não for verdadeira, pode-se proceder com uma etapa de pré-processamento para ICA, na qual as variáveis observadas são subtraídas da sua média amostral. Isso significa que as variáveis originais, denotadas aqui por $\mathbf{x}'$ podem ser pré-processadas utilizando-se a seguinte equação:

$$\mathbf{x} = \mathbf{x}' - E\{\mathbf{x}'\}, \tag{9}$$

   onde $E\{.\}$ denota o operador de esperança matemática. Dessa forma, as componentes independentes também terão média zero, uma vez que:

$$E\{\mathbf{s}\} = \mathbf{A}^{-1}E\{\mathbf{x}\}. \tag{10}$$

2.5 Ambiguidades do modelo ICA

Analisando o modelo ICA formulado na Equação (5), pode-se notar que as seguintes ambiguidades ou indeterminâncias precisam ser consideradas:

1. Não é possível determinar as variâncias das componentes independentes.

   A razão para essa afirmação reside no fato que, uma vez que $\mathbf{s}$ e $\mathbf{A}$ são desconhecidos, qualquer multiplicador escalar α_i aplicado a uma das fontes s_i pode ser cancelado dividindo-se a coluna correspondente $\mathbf{a}_i$ de $\mathbf{A}$ pelo mesmo escalar:

$$\mathbf{x} = \sum_i (\frac{1}{\alpha_i}\mathbf{a}_i)(s_i\alpha_i). \tag{11}$$

Procedendo dessa forma, as magnitudes das componentes independentes podem ser facilmente restauradas. Felizmente, essa ambiguidade é insignificante na maioria das aplicações.

2. Não é possível determinar a ordem das componentes independentes, tal como acontece na técnica PCA, na qual...

   Novamente devido ao fato de que **s** e **A** são desconhecidos, a ordem dos termos da somatória na Equação (6) pode mudar livremente, considerando qualquer uma das componentes independentes como sendo a primeira.

2.6 Decorrelação, Branqueamento e Independência

Dadas algumas variáveis aleatórias, é possível utilizar métodos lineares para transformá-las em variáveis não correlacionadas. Um processo utilizado para esse fim é denominado de branqueamento (*whitening*) e pode ser implementado utilizando-se a técnica PCA. Nessa seção, é explicado porque métodos de decorrelação de variáveis, como a PCA, não podem ser utilizados para encontrar as componentes independentes em um conjunto de dados.

Não correlação pode ser considerada como uma forma mais fraca de independência. Duas variáveis aleatórias y_1 e y_2 são ditas não correlacionadas se a covariância entre estas variáveis for igual a zero:

$$\text{cov}(y_1, y_2) = E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0. \tag{12}$$

Assumindo que as variáveis aleatórias y_1 e y_2 têm média zero, a covariância entre estas é igual à correlação $\text{corr}(y_1, y_2) = E\{y_1 y_2\}$. Dessa forma, duas variáveis são consideradas não correlacionadas se a correlação entre estas for igual a zero.

Se duas variáveis aleatórias forem independentes, estas também serão não correlacionadas. Isso acontece devido a uma propriedade importante de independência estatística que estabelece que se y_1 e y_2 são independentes, então, para quaisquer duas funções h_1 e h_2, tem-se que:

$$E\{h_1(y_1)h_2(y_2)\} = E\{h_1(y_1)\}E\{h_2(y_2)\}. \tag{13}$$

Dessa forma, tomando-se $h_1(y_1) = y_1$ e $h_2(y_2) = y_2$, pode-se notar que independência implica em correlação, uma vez que, para essas escolhas de h_1 e h_2, a condição apresentada na Equação(12) é satisfeita. Como exemplo, pode-se assumir que o par (y_1, y_2), com valores discretos para as variáveis aleatórias, segue uma distribuição de probabilidade na qual as observações $(0, 1)$, $(0, -1)$, $(1, 0)$ e $(-1, 0)$ ocorrem com probabilidades iguais a $1/4$. Nesse caso,

$$E\{y_1 y_2\} = 0 \quad \text{e} \quad E\{y_1\}E\{y_2\} = 0,$$

Portanto, observando a condição apresentada na Equação(12) pode-se concluir que y_1 e y_2 são não correlacionados. Por outro lado, não correlação não implica em independência, pois considerando $h_1(y_1) = y_1^2$ e $h_2(y_2) = y_2^2$, tem-se que:

$$E\{y_1^2 y_2^2\} = 0 \quad \text{e} \quad E\{y_1^2\}E\{y_2^2\} = 1/4,$$

o que viola a propriedade de independência, apresentada na Equação(13).

Uma propriedade um pouco mais forte que a não correlação é a propriedade de branqueamento. Um vetor aleatório com média zero, $\mathbf{y}$, é branqueado se seus componentes são não correlacionados e possuem variâncias iguais a um. Em outras palavras, a matriz de covariância (assim como a matriz de correlação) de $\mathbf{y}$ é igual à matriz identidade:

$$E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}. \tag{14}$$

Consequentemente, o processo de branqueamento corresponde a transformar linearmente o vetor de dados observados $\mathbf{x}$, multiplicando-o por alguma matriz $\mathbf{V}$:

$$\mathbf{z} = \mathbf{V}\mathbf{x}, \tag{15}$$

de modo a obter um novo vetor $\mathbf{z}$ que satisfaça a Equação (15).

Uma vez que a operação de branqueamento consiste essencialmente em uma operação de decorrelação seguida de uma mudança de escala, a técnica PCA pode ser usada para esse propósito.

Seja $\mathbf{E} = (\mathbf{e}_1 \ldots \mathbf{e}_n)$ a matriz cujas colunas são os autovetores com norma unitária da matriz de covariância $\mathbf{C_x} = E\{\mathbf{x}\mathbf{x}^T\}$. Esses autovetores podem ser computados a partir de uma amostra dos vetores $\mathbf{x}$ por meio de métodos estatísticos clássicos (ver, por exemplo, (Johnson and Wichern 1998)), ou utilizando algum algoritmo de aprendizagem para PCA (ver, por exemplo, (Diamantaras and Kung 1996) e (Oliveira 1997)).

Seja $\mathbf{D} = \text{diag}(d_1 \ldots d_n)$ a matriz diagonal com os autovalores de $\mathbf{C}$. Então uma transformação linear de branqueamento pode ser dada por:

$$\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^T, \tag{16}$$

uma vez que os elementos de $\mathbf{D}$ correspondem às variâncias das componentes originais $\mathbf{x}$ e os vetores de $\mathbf{E}$ fornecem as bases para o novo subespaço no qual as componentes são não correlacionadas.

Além do operador linear $\mathbf{V}$ da Equação(16), qualquer matriz $\mathbf{UV}$, com $\mathbf{U}$ sendo uma matriz ortogonal, também é uma matriz de branquemento. Uma instância importante de $\mathbf{UV}$ é a matriz $\mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$, que foi obtida multiplicando-se a matriz $\mathbf{V}$ da Equação(16), pela esquerda, pela matriz ortogonal $\mathbf{E}$. A matriz resultante dessa operação é a matriz inversa da raiz quadrada de $\mathbf{C_x}$, denotada por $\mathbf{C_x}^{-1/2}$, o que simplifica os cálculos envolvidos no processo de branqueamento.

3 Abordagem ICA baseada na Teoria da Informação

Nos trabalhos pioneiros sobre componentes independentes, o estudo da separação de sinais de fontes misturados e observados em um vetor de sinais de sensores era considerado um problema clássico de processamento de sinais de difícil solução. Num trabalho importante de separação cega de fontes (BSS), Herault e Jutten (Herault and Jutten 1986) introduziram um algoritmo adaptativo em uma arquitetura simples, com feedback, capaz de separar várias fontes independentes desconhecidas. Essa abordagem foi posteriormente pesquisada por Jutten e Herault (Jutten and Herault 1991), Karhunen e Joutsensalo (Karhunen and Joutsensalo 1994) e Cichocki et al. (Cichocki, Unbehauen, and Rummert 1994). Comon (Comon 1994) introduziu o conceito de Análise de Componentes Independentes e propôs funções de custo relacionadas à aproximação da minimização da informação mútua entre os sensores.

Em paralelo aos estudos envolvendo o problema de BBS, regras de aprendizado não supervisionado baseadas na teoria da informação foram propostas por Linsker (Linsker 1992). A meta era maximizar a informação mútua entre as entradas e saídas de uma rede neural. Essa abordagem está relacionada ao princípio da redução de redundância sugerido por Barlow (Barlow 1961) como uma estratégia para codificação em neurônios. De acordo com esse princípio, cada neurônio deveria codificar características que fossem tão estatisticamente independentes dos outros neurônios quanto possível, considerando um determinado conjunto de entradas. Roth e Baram (Roth and Baram 1996) e Bell e Sejnowski (Bell and Sejnowski 1995) derivaram, independentemente, regras de aprendizado por gradiente estocásticas para essa maximização e aplicaram-nas, respectivamente, à previsão e análise de séries temporais e à separação cega de fontes. Bell e Sejnowski (Bell and Sejnowski 1995) incorporaram o problema BBS a um arcabouço de teoria da informação e demonstraram a eficiência do modelo para a separação de fontes misturadas. Seus métodos são mais plausíveis a partir de uma perspectiva de processamento neural do que as funções de custo baseadas em cumulantes proposta por Comon (Comon 1994). Um método adaptativo para BBS, similar a esse trabalho de Comon, foi proposto por Cardoso e Laheld (Cardoso 1998).

Outros algoritmos para realizar ICA foram propostos a partir de diferentes pontos de vista. Abordagens de Estimação por Máxima Verossimilhança (MLE)[8] para ICA foram inicialmente propostos por Gaeta e Lacoume (Gaeta and Lacoume 1990) e elaborados por Pearlmutter e Parra (Pearlmutter and Parra 1996). Algoritmos de PCA não linear, que foram desenvolvidos por Karhunen e Joutsensalo (Karhunen and Joutsensalo 1994), Xu (Xu 1993) e Oja (Oja 1997), também podem ser vistos a partir do princípio de maximização da informação (também chamado de *infomax*), uma vez que estes algoritmos aproximadamente minimizam a informação mútua nas saídas da rede.

A seguir, será demonstrado como uma abordagem para ICA pode ser formulada em um arcabouço para o problema de separação de fontes, baseando-se em conceitos da teoria da informação.

[8]do original, em inglês, Maximum Likelihood Estimation.

3.1 Formulação do problema ICA utilizando conceitos de teoria de informação

Inicialmente, assume-se que existe um vetor M-dimensional com média zero $\mathbf{s}(t) = [s_1(t), \ldots, s_M(t)]^T$, cujos componentes são mutuamente independentes. O vetor $\mathbf{s}(t)$ corresponde aos M sinais de fontes independentes $s_i(t)$ com valores escalares. Dessa forma, pode-se escrever a fdp multivariada do vetor $\mathbf{s}(t)$ como o produto das distribuições marginais independentes:

$$p(\mathbf{s}(t)) = \prod_{i=1}^{M} p_i(s_i(t)). \tag{17}$$

Um vetor de dados $\mathbf{x}(t) = [x_1(t), \ldots, x_N(t)]^T$ é observado a cada instante de tempo t, tal que:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \tag{18}$$

onde $\mathbf{A}$ é uma matriz escalar $N \times M$. Como as componentes x_i dos vetores observados não são consideradas como sendo independentes, a fdp multivariada $p(\mathbf{x})$ não irá satisfazer a igualdade de produto na Equação (17). A informação mútua $I(\mathbf{x})$ do vetor observado é dada pela medida de divergência $D(.\|.)$ de Kullback-Leibler (KL) entre a densidade multivariada $p(\mathbf{x})$ e a densidade escrita na forma de produto:

$$I(\mathbf{x}) = D\left(p(\mathbf{x}) \middle\| \prod_{i=1}^{N} p_i(x_i)\right) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\prod_{i=1}^{N} p_i(x_i)} d\mathbf{x}. \tag{19}$$

A informação mútua é sempre positiva e é igual a zero somente quando as componentes x_i são independentes (Cover and Thomas 1991). Portanto, nessa abordagem, a informação mútua dos dados observados é utilizada como uma medida de dependência a ser minimizada.

A meta da ICA é encontrar uma transformação linear $\mathbf{W}$ dos sinais de sensores dependentes $\mathbf{x}$ que torne as saídas $\mathbf{u}$ tão independentes quanto possível:

$$\mathbf{u}_t = \mathbf{W}\mathbf{x}_t = \mathbf{W}\mathbf{A}\mathbf{s}_t, \tag{20}$$

onde $\mathbf{u}$ é uma estimativa das fontes originais. As fontes são recuperadas de forma exata quando $\mathbf{W}$ é a inversa de $\mathbf{A}$. Todavia, existem ambiguidades nesse problema (ver Seção (2.5)) e a matriz $\mathbf{A}^{-1}$ geralmente não pode ser determinada na sua forma exata. O que pode ser feito é estimar uma versão rescalonada e permutada dos M sinais de fontes, uma vez que não é possível conhecer a magnitude e a ordem desses sinais.

Para a estimação da matriz $\mathbf{W}$ pela abordagem infomax, além das suposições e restrições da ICA apresentadas na Seção(2.4), deve-se supor a ausência de ruídos de sensores,

ou no máximo a existência de sinais com poucos ruídos aditivos. Essa suposição é necessária para satisfazer a condição de maximização de informação, que estabelece que a informação mútua entre as saídas somente é minimizada no caso de pouco ruído (Linsker 1992), (Nadal and Parga 1994).

3.2 Abordagem Infomax para ICA

Considerando um processador neural com entradas $\mathbf{x}$ e saídas $\mathbf{y}$, Nadal e Parga (Nadal and Parga 1994) mostraram que a maximização da transferência de informação em uma rede neural não linear minimiza a informação mútua entre as saídas quando a otimização é aplicada aos pesos sinápticos $\mathbf{W}$ e à função de transferência não linear $g(\mathbf{u})$. Roth e Baram (Roth and Baram 1996) e Bell e Sejnowski (Bell and Sejnowski 1995) derivaram, independentemente, regras de aprendizado por gradiente para resolver esse problema de maximização, e aplicaram-nas, respectivamente, para previsão e análise de séries temporais e separação cega de fontes. Bell e Sejnowski (Bell and Sejnowski 1995) propuseram um algoritmo de aprendizado simples para uma rede neural *feedforward* que separa cegamente misturas lineares $\mathbf{x}$ de fontes independentes $\mathbf{s}$, utilizando o princípio de maximização de informação. Eles mostraram que maximizar a entropia conjunta $H(\mathbf{y})$ da saída de um processador neural pode aproximadamente minimizar a informação mútua entre as componentes de saída $y_i = g(u_i)$, em que $g(u_i)$ é uma não linearidade monotônica inversível, [9] sendo $\mathbf{u} = \mathbf{W}\mathbf{x}$.

A entropia conjunta das saídas de uma rede neural é dada por:

$$H(y_1, \ldots, y_N) = H(y_1) + \ldots + H(y_N) - I(y_1, \ldots, y_N), \tag{21}$$

onde $H(y_i)$ são as entropias marginais das saídas e $I(y_1, \ldots, y_N)$ é a informação mútua dessas saídas. Maximizar $H(y_1, \ldots, y_N)$ consiste, portanto, em maximizar as entropias marginais e minimizar a informação mútua $I(\mathbf{y})$.

Esse problema é, ainda, equivalente à maximizar a informação mútua $I(\mathbf{y}, \mathbf{x})$ que as saídas $\mathbf{y}$ de uma rede neural possuem sobre as suas entradas $\mathbf{x}$. Essa medida pode ser definida como:

$$I(\mathbf{y}, \mathbf{x}) = H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}), \tag{22}$$

onde $H(\mathbf{y}|\mathbf{x})$ é a entropia da saída que não tenha advindo da entradas $\mathbf{x}$. No caso de ausência de ruído (ou melhor, quando não se sabe o que é ruído e o que é sinal na entrada), o mapeamento entre $\mathbf{x}$ e $\mathbf{y}$ é determinístico e $H(\mathbf{y}|\mathbf{x})$ possui um valor muito pequeno, podendo, portanto, ser desprezado no processo de maximização.

As saídas $\mathbf{y}$ são variáveis aleatórias de amplitude limitada e, portanto, as entropias marginais $H(y_i)$ atingem seus valores máximos quando as distribuições de y_i forem uniformes. Maximizar a entropia conjunta também irá diminuir $I(y_1, \ldots, y_N)$, uma vez que

[9] a função $g(u_i)$ é monotônica inversível se esta possuir uma única função inversa $g^{-1}(u_i)$.

a informação mútua é sempre positiva. Para $I(y_1, \ldots, y_N) = 0$, a entropia conjunta é a soma das entropias marginais:

$$H(y_1, \ldots, y_N) = H(y_1) + \ldots + H(y_N). \tag{23}$$

Dessa forma, o valor máximo para $H(y_1, \ldots, y_N)$ é atingido quando a informação mútua entre as variáveis aleatórias $y_1, \ldots, y_N$ for igual a zero e suas distribuições marginais forem uniformes.

Nessa abordagem, os pesos sinápticos $\mathbf{W}$ são determinados maximizando-se a entropia conjunta em relação à $\mathbf{W}$. Nesse caso, a derivada da Equação (21) em relação à $\mathbf{W}$ pode ser escrita em termos da divergência KL entre a distribuição uniforme multivariada, denotada por $p_1(\mathbf{y})$, e a estimativa da distribuição multivariada $p(\mathbf{y})$:

$$\frac{\partial H(\mathbf{y})}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}}(-D(p_1(\mathbf{y})\|p(\mathbf{y}))). \tag{24}$$

No limite, quando a função de transferência $g(u_i)$ e $\mathbf{W}$ são otimizados, a entropia conjunta $H(\mathbf{y})$ é máxima, e $p(\mathbf{y}) = p_1(\mathbf{y})$, de modo que $I(\mathbf{y}) = 0$. Se $g(u_i)$ for um mapeamento inversível de u_i para y_i, a divergência KL na Equação(24) é igual à divergência KL entre as estimativas da distribuição de fontes $p(\mathbf{u})$ e da distribuição das fontes verdadeiras $p(\mathbf{s})$,

$$D(p_1(\mathbf{y})\|p(\mathbf{y})) = D(p(\mathbf{s})\|p(\mathbf{u})), \tag{25}$$

uma vez que a divergência KL é invariante para qualquer transformação inversível.

Se a informação mútua entre as saídas for igual à zero, ou seja, $I(y_1, \ldots, y_N) = 0$, a informação mútua antes de aplicar a não linearidade, $I(u_1, \ldots, u_N)$, também deve ser igual à zero. Isso ocorre porque a função não linear $g(u_i)$ não introduz quaisquer dependências. Um resultado fundamental encontrado na literatura estabelece que, quando $g(u_i)$ é monotonicamente crescente ou decrescente (isto é, possui uma única função inversa $g^{-1}(u_i)$), a relação entre u_i e y_i é dada por (Papoulis 1991):

$$p(y_i) = \frac{p(u_i)}{\left|\frac{\partial y_i}{\partial u_i}\right|} = \frac{p(u_i)}{\left|\frac{\partial g(u_i)}{\partial u_i}\right|}. \tag{26}$$

Se y_i apresentar uma distribuição uniforme, segue que:

$$p(u_i) = \left|\frac{\partial g(u_i)}{\partial u_i}\right|, \tag{27}$$

uma vez que, nesse caso, $p(y_i)$ atinge o seu valor máximo igual a 1.

Isso significa que u_i é uma variável independente com uma distribuição que possui aproximadamente a forma da derivada da não linearidade, g. No caso da função logística, (ver Figura(5a)), a fdp apropriada possui a forma apresentada na Figura(5b). Bell e

Sejnowski (Bell and Sejnowski 1995) realizaram a separação de misturas de vários sinais de fala e música utilizando a abordagem infomax com função de transferência logística, também chamada de função sigmóide.

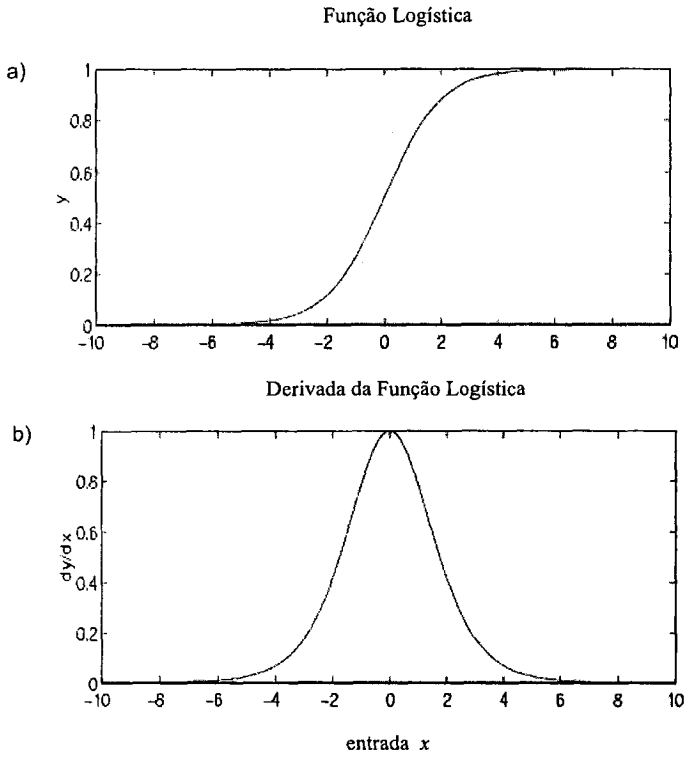


Figura 5: (a)A função logística ($y = 1/(1+\exp(-x))$) e (b) sua derivada ($\frac{\partial y}{\partial x} = y(1-y)$).

Uma arquitetura simples que pode realizar o mapeamento de **x** para **y** consiste em uma rede neural *feedforward* com uma única camada e uma função de ativação de saída não linear $y_i = g(u_i)$. A escolha dessa não linearidade é essencial para que a minimização da informação mútua realize a técnica ICA. Como proposto em (Bell and Sejnowski 1995) e apresentado a seguir, as regras de aprendizado para ICA podem ser derivadas por meio da maximização da entropia de saída $H(\mathbf{y})$ de um processador neural.

3.2.1 Regras de Aprendizado ICA para uma Rede Neural com uma Entrada e uma Saída

Quando uma única entrada x passa por uma função de transferência $g(x)$ para resultar em uma variável de saída y, a entropia da saída, $H(y)$ é maximizada, alinhando-se as partes de alta densidade da fdp de x com as partes de alta inclinação da função $g(x)$ (ver, como ilustração, a Figura(5)).

Adaptando a Equação(26) para esse caso, tem-se que:

$$p_y(y) = \frac{p_x(x)}{\left|\frac{\partial y}{\partial x}\right|}. \tag{28}$$

A entropia da saída é dada por:

$$H(y) = -E\{\ln p_y(y)\}. \tag{29}$$

Substituindo a Equação(28) na Equação(29), tem-se que:

$$H(y) = E\left\{\ln\left|\frac{\partial y}{\partial x}\right|\right\} - E\{\ln p_x(x)\}. \tag{30}$$

O segundo termo à direita da Equação(30) refere-se à entropia de x, podendo-se, portanto, considerar que o mesmo não é afetado pelas alterações no parâmetro w que determinam $g(x)$. Dessa forma, para maximizar a entropia de y, por meio de mudanças aplicadas à w, é necessário concentrar-se apenas na maximização do primeiro termo da Equação(30), que corresponde ao valor esperado do logaritmo do quanto a entrada x afeta a saída y. Isso pode ser realizado considerando-se um conjunto de treinamento de entradas x para aproximar a densidade $p_x(x)$, e derivando-se uma regra de aprendizado por subida de gradiente, da seguinte forma:

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}\left(\ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1}\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right). \tag{31}$$

No caso da função logística,

$$y = \frac{1}{1+e^{-u}}, \quad u = wx + w_0 \tag{32}$$

$$y' = \frac{\partial y}{\partial u} = y(1-y) \tag{33}$$

e as regras de aprendizado derivadas a partir da fórmula geral dada na Equação(31) são as seguintes:

$$\Delta w \propto \frac{1}{w} + x(1-2y) \tag{34}$$

$$\Delta w_0 \propto 1 - 2y. \tag{35}$$

As derivações completas para as regras nas Equações (34) e (35) podem ser encontradas no Apêndice A.

A Figura(6) ilustra o efeito das regras para atualização de w e w_0. Por exemplo, se a fdp da entrada, $p_x(x)$ for gaussiana, a regra para w_0 irá alinhar a parte mais inclinada da curva sigmóide com o pico de $p_x(x)$, casando a densidade da entrada com a curva da função y. Esse método para aproximar densidades foi inicialmente proposto em (Roth and Baram 1996). A regra para w irá, dessa forma, escalonar a inclinação da curva sigmóide

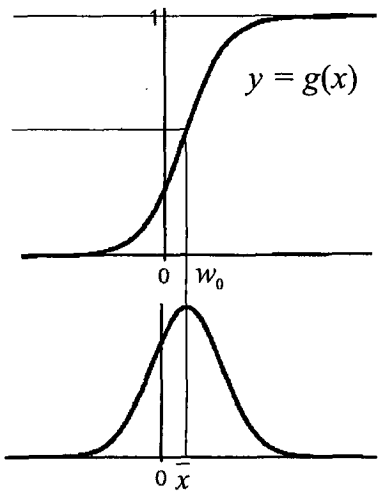


Figura 6: Uma entrada x com densidade $p_x(x)$, nesse caso uma densidade normal, passa por uma função não linear $g(x)$. Como resultado, a média da entrada, $\overline{x}$, deve coincidir com o termo w_0 e a variância da entrada, σ^2, deve coincidir com a inclinação da função $g(x)$.

para estimar a variância de $p_x(x)$. Por exemplo, uma fdp apresentando uma forma estreita iria resultar em uma sigmóide de alta inclinação.

Para a função tangente hiperbólica,

$$y = \tanh(u), \quad u = wx + w_0 \tag{36}$$

$$y' = \frac{\partial y}{\partial u} = 1 - y^2, \tag{37}$$

obtém-se as seguintes regras de aprendizado:

$$\Delta w \propto \frac{1}{w} - 2xy \tag{38}$$

$$\Delta w_0 \propto -2y. \tag{39}$$

As derivações completas para as regras nas Equações (38) e (39) podem ser encontradas no Apêndice A.

Pode-se observar que as regras para w, apresentadas nas Equações (34) e (38), possuem um termo anti-Hebbiano, $x(1 - 2y)$ (função logística) ou $-2xy$ (função tangente hiperbólica), e um termo anti-decaimento $1/w$. Esse termo tem o papel de evitar duas situações nas quais a saída y torna-se não informativa: (i) quando os valores de y ficam saturados em 0 ou 1 e (ii) quando w é tão pequeno que o valor de y permanece por volta de 0.5.

O efeito dessas regras é produzir uma distribuição de saída $f_y(y)$ que seja próxima

à distribuição uniforme, que é a distribuição com entropia máxima para uma variável limitada entre 0 e 1.

A seguir, será mostrado como essa abordagem pode ser adequada a uma rede neural artificial com entradas e saídas multidimensionais.

3.3 Regras de Aprendizado ICA para uma Rede Neural com N Entradas e N Saídas

Considere uma rede neural com um vetor de entrada $\mathbf{x} \in \mathbb{R}^N$, isto é, $\mathbf{x}$ é N-dimensional, uma matriz de pesos $\mathbf{W}$, um vetor de ruído $\mathbf{w_0}$ e um vetor de saídas $\mathbf{y} = g(\mathbf{Wx} + \mathbf{w_0}) \in \mathbb{R}^N$. De maneira análoga à Equação (28), é possível relacionar $p_{\mathbf{x}}(\mathbf{x})$ com $p_{\mathbf{y}}(\mathbf{y})$ pelo determinante da matriz Jacobiana $\mathbf{J}(\mathbf{x})$, da seguinte forma (Papoulis 1991):

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{|\mathbf{J}|}, \tag{40}$$

onde $|\mathbf{J}|$ é o valor absoluto do determinante da matriz Jacobiana, dada pela seguinte matriz de derivadas parciais:

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix} \tag{41}$$

A derivação das regras para a adaptação de $\mathbf{W}$ e $\mathbf{w_0}$, neste caso, procede como no caso para uma uma entrada e uma saída, visto anteriormente. No entanto, por estar considerando vetores no espaço $\mathbb{R}^N$, ao invés de maximizar $\ln|\partial y/\partial x|$, deve-se maximizar $\ln|\mathbf{J}|$. Essa quantidade representa o logaritmo do volume de espaço em $\mathbf{y}$ no qual os pontos em $\mathbf{x}$ são mapeados. Maximizando-se essa quantidade, tenta-se espalhar de forma uniforme, o conjunto de treinamento de pontos $\mathbf{x}$ em $\mathbf{y}$.

Pode-se considerar, primeiramente, o caso de uma rede neural composta por unidades com função de ativação sigmóide, $\mathbf{y} = g(\mathbf{u})$, $\mathbf{u} = \mathbf{Wx} + \mathbf{w_0}$, com g sendo a função logística $g(\mathbf{u}) = (1 + e^{-\mathbf{u}})^{-1}$. As regras de aprendizado derivadas para esse caso são similares às das Equações (34) e (35):

$$\Delta\mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} + (\mathbf{1} - 2\mathbf{y})\mathbf{x}^T \tag{42}$$

$$\Delta\mathbf{w_0} \propto \mathbf{1} - 2\mathbf{y}. \tag{43}$$

No entanto, agora $\mathbf{x}$, $\mathbf{y}$, $\mathbf{w_0}$ e $\mathbf{1}$ são vetores $\in \mathbb{R}^N$ ($\mathbf{1}$ é um vetor no qual todos os elementos são iguais a 1), $\mathbf{W}$ é uma matriz e o termo anti-Hebbiano tornou-se um produto interno entre dois vetores. O termo anti-decaimento foi generalizado para um termo anti-redundância: a inversa da transposta da matriz de pesos. Para um peso individual, w_{ij}, essa regra equivale à:

$$\Delta w_{ij} \propto \frac{\text{cof } w_{ij}}{\det \mathbf{W}} + x_j(1 - 2y_i), \tag{44}$$

onde cof w_{ij}, o cofator de w_{ij}, é obtido multiplicando-se $(-1)^{i+j}$ pelo determinante da matriz obtida removendo-se a i-ésima linha e j-ésima coluna da matriz $\mathbf{W}$.

A derivação completa para a regra em (42) pode ser encontrada no Apêndice A. Essa regra tem o mesmo efeito da regra para w apresentada na Equação(34). A exceção aqui é que, ao invés do ponto instável da dinâmica da rede ser $w = 0$, qualquer matriz $\mathbf{W}$ cujo determinante seja igual zero irá resultar em uma solução degenerada. Nesse caso, para que a rede encontre uma solução estável, as diferentes unidades y_i devem aprender a representar aspectos diferentes das entradas. Portanto, quando os vetores de pesos referentes a duas unidades de saída diferentes tornam-se muito parecidos, o determinante de $\mathbf{W}$ torna-se pequeno e a dinâmica natural do aprendizado faz com que esses dois vetores de pesos divirjam entre si. Na regra apresentada na Equação(44), esse efeito é atenuado pelo numerador, cof w_{ij}. Quando esse cofator torna-se pequeno, pode haver uma indicação de que existe uma degeneração na matriz de pesos do resto dessa camada (ou seja, aqueles pesos que não estão associados com a entrada x_j ou com a saída y_i), mostrando que, nesse caso, qualquer degeneração em $\mathbf{W}$ tem pouca relação com o peso específico w_{ij} que está sendo ajustado.

Para neurônios cujas funções de transferência sejam a função tangente hiperbólica, as seguintes regras de aprendizado são derivadas:

$$\Delta \mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} - 2\mathbf{y}\mathbf{x}^T \tag{45}$$

$$\Delta \mathbf{w}_0 \propto -2\mathbf{y}. \tag{46}$$

Generalizando as regras de aprendizado para $\mathbf{W}$, apresentadas nas Equações (42) e (45), para qualquer funçaõ de transferência ou ativação $g(\mathbf{u})$, chega-se à seguinte equação:

$$\Delta \mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} - \varphi(\mathbf{u})\mathbf{x}^T, \tag{47}$$

que é de fundamental importância na derivação do modelo ICAMM, que será apresentado na Seção (4). De fato, essa mesma regra pode ser derivada a partir de outros pontos de vista teóricos, tais como por meio das abordagens MLE (Pearlmutter and Parra 1996) e maximização da negentropia (Girolami and Fyfe 1997). Uma revisão dessas técnicas e as relações entre estas podem ser encontradas em (Lee, Girolami, Bell, and Sejnowski 1998).

3.4 Equivalência entre Maximização de Informação e Estimação por Máxima Verossimilhança

Como dito anteriormente, no modelo ICA, as observações $\mathbf{x}$ são assumidas como sendo geradas a partir de variáveis latentes $\mathbf{s}$ por meio de um mapeamento linear $\mathbf{A}$. No caso

da ausência de ruído, pode-se usar um estimador paramétrico da densidade $\hat{p}(\mathbf{x};\mathbf{a})$ para encontrar um vetor de parâmetros $\mathbf{a}$ que minimize a diferença entre o modelo generativo $\hat{p}(\mathbf{x};\mathbf{a})$ e a distribuição observada $p(\mathbf{x})$. Os vetores $\mathbf{a}$ são considerados os vetores de bases de $\mathbf{A}$, de modo que $\hat{p}(\mathbf{x};\mathbf{a})$ é uma estimativa da densidade $p(\mathbf{x})$ dos vetores observados. A diferença entre a estimativa da densidade e a densidade das obervações pode ser medida pela divergência KL:

$$D\big(p(\mathbf{x})\|\hat{p}(\mathbf{x};\mathbf{a})\big) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\hat{p}(\mathbf{x};\mathbf{a})} d\mathbf{x} = H(\mathbf{x}) - \int p(\mathbf{x}) \log \hat{p}(\mathbf{x};\mathbf{a}) d\mathbf{x}, \tag{48}$$

onde $p(\mathbf{x})$ é a fdp das observações $\mathbf{x}$ e $\hat{p}(\mathbf{x};\mathbf{a})$ é uma estimativa paramétrica da distribuição $p(\mathbf{x})$. A divergência $D(p(\mathbf{x})\|\hat{p}(\mathbf{x};\mathbf{a}))$ é zero somente quando a estimativa $\hat{p}(\mathbf{x};\mathbf{a})$ for igual à densidade das observações $p(\mathbf{x})$. Pearlmutter e Parra (Pearlmutter and Parra 1996) e Cardoso (Cardoso 1998) mostraram que as abordagens infomax e MLE são equivalentes para a ICA, como será descrito brevemente aqui.

A verossimilhança de que a amostra $\mathbf{X} = \{x_1, \ldots, x_T\}$ seja geradas com uma distribuição particular $\hat{p}(\mathbf{x};\mathbf{a})$ é:

$$\hat{p}(\mathbf{X};\mathbf{a}) = \prod_{i=1}^{T} \hat{p}(\mathbf{x}_i;\mathbf{a}). \tag{49}$$

Tomando-se o logaritmo da Equação(49) e dividindo a mesma pelo número de observações, resulta na log-verossimilhança normalizada, dada pela seguinte equação:

$$L(\mathbf{a}) = \frac{1}{T} \sum_{i=1}^{T} \log \hat{p}(\mathbf{x}_i;\mathbf{a}). \tag{50}$$

Uma vez que essa equação corresponde à média amostral de $\log \hat{p}(\mathbf{X};\mathbf{a})$, esta irá convergir, pela lei dos grandes números (ver, por exemplo, (DeGroot 1987)), para a sua esperança matemática:

$$L(\mathbf{a}) = \int p(\mathbf{x}) \log \hat{p}(\mathbf{x};\mathbf{a}) d\mathbf{x}. \tag{51}$$

Fazendo $\hat{p}(\mathbf{x};\mathbf{a}) = \frac{\hat{p}(\mathbf{x};\mathbf{a})}{p(\mathbf{x})} p(\mathbf{x})$, a equação anterior pode ser reescrita da seguinte forma:

$$\begin{aligned} L(\mathbf{a}) &= \int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} - \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\hat{p}(\mathbf{x};\mathbf{a})} d\mathbf{x} && (52) \\ &= H(\mathbf{x}) - D(p(\mathbf{x}), \hat{p}(\mathbf{x}\|\mathbf{a})). && (53) \end{aligned}$$

Uma vez que $H(\mathbf{x})$ não depende de $\mathbf{W}$, maximizar a log-verossimilhança corresponde a minimizar a divergência KL entre a densidade das observações $p(\mathbf{x})$ e a densidade estimada $\hat{p}(\mathbf{x};\mathbf{a})$,

$$\frac{\partial L(\mathbf{a})}{\partial \mathbf{W}} = -\frac{\partial}{\partial \mathbf{W}} D(p(\mathbf{x}) \| \hat{p}(\mathbf{x}, \mathbf{a})). \tag{54}$$

Dado que $\mathbf{A}$ é uma matriz inversível e a divergência KL é invariante sob uma transformação inversível, minimizar a divergência KL apresentada na Equação(54) equivale a minimizar a divergência entre a densidade das estimativas de fontes $p(\mathbf{u})$ e a densidade das fontes verdadeiras $p(\mathbf{s})$:

$$\frac{\partial L(\mathbf{a})}{\partial \mathbf{W}} = -\frac{\partial}{\partial \mathbf{W}} D(p(\mathbf{s}) \| \hat{p}(\mathbf{u})). \tag{55}$$

Dessa forma, as Equações (55) e (24) são equivalentes para o método ICA.

4 Modelo de Misturas ICA

Uma limitação da ICA é a suposição de que as fontes são independentes. Em (Lee, Lewicki, and Sejnowski 2000), é apresentada uma abordagem para relaxar essa suposição utilizando modelos de misturas. Em um modelo de misturas (ver, por exemplo, (Duda and Hart 1973)), os dados observados podem ser categorizados em várias classes mutuamente exclusivas. Quando os dados em cada classe são modelados com uma distribuição gaussiana (normal) multivariada, o modelo é chamado de modelo de misturas gaussianas. O conceito de modelo de misturas pode ser generalizado por meio da suposição de que os dados em cada classe são gerados por uma combinação linear de fontes não gaussianas independentes, como no caso da ICA. Tal modelo é chamado de Modelo de Misturas ICA (ICAMM). O algoritmo para a aprendizagem dos parâmetros desse modelo utiliza o método de otimização gradiente ascendente para maximizar a função de log-verossimilhança dos dados.

No decorrer desta seção, assume-se que os dados N-dimensionais $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_T$ são gerados independentemente por um modelo de misturas de densidades (Duda and Hart 1973). A verossimilhança dos dados é dada pela densidade conjunta:

$$p(\mathbf{X}|\Theta) = \prod_{t=1}^{T} p(\mathbf{x}_t|\Theta), \tag{56}$$

sendo a log-verossimilhança, portanto, dada pela seguinte equação:

$$L = \sum_{t=1}^{T} \log p(\mathbf{x}_t|\Theta). \tag{57}$$

A densidade das misturas é dada por:

$$p(\mathbf{x}_t|\Theta) = \sum_{t=1}^{K} p(\mathbf{x}_t|C_k, \theta_k)p(C_k), \tag{58}$$

onde $\Theta = (\theta_1, \ldots, \theta_K)$ são os parâmetros desconhecidos para cada $p(\mathbf{x}|C_k, \theta_k)$, chamadas de densidades das componentes da mistura. Nesse caso, C_k denota a classe k e assume-se que o número de classes, K, é conhecido de antemão. Assume-se, ainda que as densidades das componentes são não gaussianas e que os dados em cada classe são descritos por:

$$\mathbf{x}_t = \mathbf{A}_k\mathbf{s}_k + \mathbf{b}_k, \tag{59}$$

onde $\mathbf{A}_k$ é uma matriz escalar $N \times N$[10] e $\mathbf{b}_k$ é o vetor de ruído para a classe k. O vetor $\mathbf{s}_k$ é chamado de vetor de fontes para classe k.

A meta do ICAMM é classificar um conjunto de dados não rotulados e determinar os parâmetros para cada classe $(\mathbf{A}_k, \mathbf{b}_k)$ e as probabilidades condicionais $p(C_k|\mathbf{x}_t, \Theta)$ para todos os dados do conjunto.

O algoritmo iterativo que realiza a subida do gradiente da função de log-verossimilhança dos dados (ver Equação 57) possui os seguintes passos:

- Compute a log-verossimilhança dos dados para cada classe:

$$\log p(\mathbf{x}_t|C_k, \theta_k) = \log p(\mathbf{s}_k) - \log(|\det(\mathbf{A}_k)|), \tag{60}$$

  onde $\theta_k = \{\mathbf{A}_k, \mathbf{b}_k\}$. Nesse caso, $\mathbf{s}_k = \mathbf{A}_k^{-1}(\mathbf{x}_t - \mathbf{b}_k)$ é implicitamente modelado para a adaptação de $\mathbf{A}_k$;

- Compute a probabilidade para cada classe, dado o vetor de dados $\mathbf{x}_t$:

$$p(C_k|\mathbf{x}_t, \Theta) = \frac{p(\mathbf{x}_t|C_k, \theta_k)p(C_k)}{\sum_k p(\mathbf{x}_t|C_k\theta_k)p(C_k)}; \tag{61}$$

- Adapte as matrizes $\mathbf{A}_k$ e os vetores de bias $\mathbf{b}_k$ para cada classe, utilizando as seguintes regras:

$$\Delta\mathbf{A}_k \propto -p(C_k \mid \mathbf{x}_t, \Theta)\mathbf{A}_k[\mathbf{I} - \mathbf{K}\tanh(\mathbf{s}_k)\mathbf{s}_k^T - \mathbf{s}_k\mathbf{s}_k^T] \tag{62}$$

  e

$$\mathbf{b}_k = \frac{\sum_t \mathbf{x}_t p(C_k \mid \mathbf{x}_t, \Theta)}{\sum_t p(C_k \mid \mathbf{x}_t, \Theta)}, \tag{63}$$

  em que $t = 1, \ldots, T$ e $\mathbf{K}$ é uma matriz diagonal N-dimensional cujos elementos $k_{k,i}$ são adaptados da seguinte forma:

$$k_{k,i} = \text{sign}\big(E\{\text{sech}^2(s_{k,i,t})\}E\{s_{k,i,t}^2\} - E\{[\tanh(s_{k,i,t})]s_{k,i,t}\}\big). \tag{64}$$

[10]Neste caso, o número de fontes é igual ao número de sensores.

A distribuição é considerada supergaussiana quando $k_{k,i} = 1$ e subgaussiana quando $k_{k,i} = -1$ (Girolami 1998).

Como discutido na Seção (2), existem vários métodos para adaptar as matrizes $\mathbf{A}_k$ no modelo ICA. O algoritmo ICAMM é baseado na regra de aprendizado ICA infomax estendida (Lee, Girolami, and Sejnowski 1999), que possibilita separar fontes desconhecidas com distribuições sub e supergaussianas. Essa separação é atingida por meio de um tipo simples de regra de aprendizado que foi primeiramente derivada por Girolami (Girolami 1998). A regra de aprendizado em (Lee, Girolami, and Sejnowski 1999) utiliza a análise de estabilidade de (Cardoso 1998) para alternar entre regimes sub e super-Gaussianos.

Para a estimação da função log-verossimilhança da Equação (60), o termo $\log p(\mathbf{s}_k)$ é modelado como segue:

$$\log p(\mathbf{s}_{k,t}) \propto -\sum_{i=1}^{N}\left(k_{k,i}\log(\cosh s_{k,i,t}) - \frac{s_{k,i,t}^2}{2}\right). \tag{65}$$

4.1 Derivação do Algoritmo ICAMM

Para a derivação do algoritmo ICAMM, assume-se que $p(\mathbf{X}|\Theta)$, como dado na Equação (56) é uma função diferenciável de Θ. Lembrando que a log-verossimilhança L dos dados é dada por:

$$L = \sum_{t=1}^{T}\log p(\mathbf{x}_t|\Theta) \tag{66}$$

e, usando a Equação (58), o gradiente para os parâmetros de cada classe k é:

$$\begin{aligned}\nabla_{\theta_k} L &= \sum_{t=1}^{T}\frac{1}{p(\mathbf{x}_t|\Theta)}\nabla_{\theta_k} p(\mathbf{x}_t|\Theta) \\ &= \sum_{t=1}^{T}\frac{\nabla_{\theta_k}\left[\sum_{k=1}^{K} p(\mathbf{x}_t|C_k,\theta_k)p(C_k)\right]}{p(\mathbf{x}_t|\Theta)} \\ &= \sum_{t=1}^{T}\frac{\nabla_{\theta_k} p(\mathbf{x}_t|C_k,\theta_k)p(C_k)}{p(\mathbf{x}_t|\Theta)}.\end{aligned} \tag{67}$$

Usando a regra de Bayes (ver, por exemplo (Papoulis 1991)) a probabilidade de classe para um vetor de dados $\mathbf{x}_t$ é:

$$p(C_k|\mathbf{x}_t,\Theta) = \frac{p(\mathbf{x}_t|\theta_k,C_k)p(C_k)}{\sum_k p(\mathbf{x}_t|\theta_k,C_k)p(C_k)}. \tag{68}$$

Substituindo a Equação (68) na Equação (67), tem-se:

$$\begin{aligned}\nabla_{\theta_k} L &= \sum_{t=1}^{T} p(C_k|\mathbf{x}_t, \Theta) \frac{\nabla_{\theta_k} p(\mathbf{x}_t|\theta_k, C_k) p(C_k)}{p(\mathbf{x}_t|\theta_k, C_k) p(C_k)} \\ &= \sum_{t=1}^{T} p(C_k|\mathbf{x}_t, \Theta) \nabla_{\theta_k} \log p(\mathbf{x}_t|C_k, \theta_k).\end{aligned} \tag{69}$$

A função de log-verossimilhança na Equação (69) representa a log-verossimilhança para cada classe. No modelo ICAMM, a log-verossimilhança de classe é dada pela log-verossimilhança para o modelo ICA padrão:

$$\begin{aligned}\log p(\mathbf{x}_t|C_k, \theta_k) &= \log \frac{p(\mathbf{s}_k)}{|\det \mathbf{A}_k|} \\ &= \log p(\mathbf{A}_k^{-1}(\mathbf{x}_t - \mathbf{b}_k)) - \log |\det \mathbf{A}_k|.\end{aligned} \tag{70}$$

Em (Lee, Lewicki, and Sejnowski 2000), o método do gradiente ascendente é utilizado para estimar os parâmetros que maximizam a função de log-verossimilhança. Os parâmetros do gradiente para cada classe são o gradiente da matriz de bases $\mathbf{A}_k$ e o gradiente do vetor de ruído $\mathbf{b}_k$, $\nabla_{\theta_k} L = \{\nabla_{\mathbf{A}_k} L, \nabla_{\mathbf{b}_k} L\}$, os quais serão considerados, um por vez, a seguir.

4.1.1 Estimação das Matrizes de Bases

As matrizes de bases $\mathbf{A}_k$ para cada classe podem ser atualizadas utilizando a Equação (69).

$$\nabla_{\mathbf{A}_k} L = \sum_{t=1}^{T} p(C_k|\mathbf{x}_t, \Theta) \nabla_{\mathbf{A}_k} \log p(\mathbf{x}_t|C_k, \theta_k). \tag{71}$$

A atualização (adaptação) é realizada utilizando o método do gradiente ascendente do gradiente da densidade de componentes em relação às matrizes de base, resultando em:

$$\Delta \mathbf{A}_k \propto p(C_k|\mathbf{x}_t, \Theta) \nabla_{\mathbf{A}_k} \log p(\mathbf{x}_t|C_k, \theta_k). \tag{72}$$

Pode-se notar que na adaptação das matrizes de bases, o gradiente da densidade de componentes com respeito às matrizes de bases $\mathbf{A}_k$ é ponderado por $p(C_k|\mathbf{x}_t, \Theta)$.

A computação de $\nabla_{\mathbf{A}_k} \log p(\mathbf{x}_t|C_k, \theta_k)$ pode ser realizada por meio da técnica de ICA. A regra de aprendizagem para as matrizes de bases $\mathbf{A}_k$ pode ser derivada a partir de vários pontos de vista teóricos, tais como por meio das abordagens MLE (Pearlmutter and Parra

1996), infomax (Bell and Sejnowski 1995) e maximização da negentropia (Girolami and Fyfe 1997).

O algoritmo de aprendizado para o ICAMM é derivado em (Lee, Lewicki, and Sejnowski 2000) utilizando a Estimação por Máxima Verossimilhança (MLE)[11]. A abordagem MLE para separação cega de fontes foi primeiramente proposta por Gaeta e Lacoume (Gaeta and Lacoume 1990) e Pham e Garrat (Pham and Garrat 1997), sendo mais recentemente pesquisada por Peralmutter and Parra (Pearlmutter and Parra 1996) e Cardoso (Cardoso 1997). A função densidade de probabilidade das observações $\mathbf{x}$ pode ser expressa como (Papoulis 1991) (Amari and Cardoso 1997):

$$p(\mathbf{x}) = |\det \mathbf{W}| p(\mathbf{u}), \tag{73}$$

em que $p(\mathbf{u}) = \prod_{i=1}^{N} p_i(u_i)$ é a suposta distribuição de $p(\mathbf{s})$. A log-verossimilhança da Equação (73) é:

$$L(\mathbf{u}, \mathbf{W}) = \log |\det \mathbf{W}| + \sum_{i=1}^{N} \log p_i(u_i). \tag{74}$$

Como $\det \mathbf{W} = \sum_j w_{ij} \mathrm{cof}(w_{ij})$ para algum j, onde $\mathrm{cof}^T(\mathbf{W})$ denota a transposta de $\mathrm{cof}(\mathbf{W})$ e $\mathbf{W}^{-1} = \frac{1}{\det \mathbf{W}} \mathrm{cof}^T(\mathbf{W})$, tem-se que, para o primeiro termo da Equação (74):

$$\frac{\partial}{\partial \mathbf{W}} \log |\det \mathbf{W}| = \frac{1}{\det \mathbf{W}} \mathrm{cof}(\mathbf{W}) = (\mathbf{W}^T)^{-1}. \tag{75}$$

Por sua vez, o segundo termo da Equação (74) é derivado da seguinte forma:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}} \sum_{i=1}^{N} \log p_i(u_i) = & \frac{\partial}{\partial \mathbf{W}} \log p_1(u_2) + \ldots + \frac{\partial}{\partial \mathbf{W}} \log p_N(u_N) = \\ = & \left[\frac{\frac{\partial p_1(u_1)}{\partial u_1} \frac{\partial u_1}{\partial \mathbf{W}}}{p_1(u_1)} + \ldots + \frac{\frac{\partial p_N(u_N)}{\partial u_N} \frac{\partial u_N}{\partial \mathbf{W}}}{p_N(u_N)} \right] = \\ = & \left[\frac{\frac{\partial p_1(u_1)}{\partial u_1}}{p_1(u_1)}, \ldots, \frac{\frac{\partial p_N(u_N)}{\partial u_N}}{p_N(u_N)} \right]^T [x_1 x_2 \ldots x_N]. \end{aligned} \tag{76}$$

Denominando:

$$\varphi(\mathbf{u}) = -\frac{\frac{\partial p(\mathbf{u})}{\partial u}}{p(\mathbf{u})} = \left[-\frac{\frac{\partial p(u_1)}{\partial u_1}}{p(u_1)}, \ldots, -\frac{\frac{\partial p(u_N)}{\partial u_N}}{p(u_N)} \right]^T, \tag{77}$$

[11] do original, em inglês, Maximum-Likelihood Estimation.

maximizar a log-verossimilhança com respeito à $\mathbf{W}$, resulta na seguinte regra de aprendizado para $\mathbf{W}$:

$$\Delta \mathbf{W} \propto \left[(\mathbf{W}^T)^{-1} - \varphi(\mathbf{u})\mathbf{x}^T \right], \tag{78}$$

que é, por sua vez, idêntica à Equação(47), derivada por meio da abordagem infomax (Bell and Sejnowski 1995).

Uma maneira ainda mais eficiente para maximizar a log-verossimilhança é seguir o gradiente "natural" (Amari, Cichocki, and Yang 1996), obtido multiplicando-se o gradiente calculado no ponto $\mathbf{W}$, pela direita, pela matriz $\mathbf{W}^T\mathbf{W}$. O gradiente natural é utilizado neste caso pois simplifica a regra de aprendizado da Equação (78) e acelera a convergência do modelo (Amari 1998). A aplicação do termo de gradiente natural resulta, portanto, na seguinte regra de gradiente descendente para função de log-verossimilhança dos dados:

$$\Delta \mathbf{W} \propto \frac{\partial L(\mathbf{u}, \mathbf{W})}{\partial \mathbf{W}} \mathbf{W}^T\mathbf{W}. \tag{79}$$

Utilizando o fato de que $\mathbf{x}^T = \mathbf{u}^T(\mathbf{W}^T)^{-1}$ a Equação (78) pode ser reescrita da seguinte forma:

$$\Delta \mathbf{W} \propto \left[(\mathbf{W}^T)^{-1} - \varphi(\mathbf{u})\mathbf{u}^T(\mathbf{W}^T)^{-1} \right]. \tag{80}$$

Colocando o termo $(\mathbf{W}^T)^{-1}$ em evidência, tem-se que:

$$\Delta \mathbf{W} \propto \left[\mathbf{I} - \varphi(\mathbf{u})\mathbf{u}^T \right] (\mathbf{W}^T)^{-1}. \tag{81}$$

Por fim, multiplicando a Equação (81), pela direita, pelo termo $\mathbf{W}^T\mathbf{W}$ do gradiente natural, obtém-se:

$$\begin{aligned} \Delta \mathbf{W} &\propto \left[\mathbf{I} - \varphi(\mathbf{u})\mathbf{u}^T \right] (\mathbf{W}^T)^{-1}\mathbf{W}^T\mathbf{W} = \\ \Delta \mathbf{W} &\propto \left[\mathbf{I} - \varphi(\mathbf{u})\mathbf{u}^T \right] \mathbf{W}. \end{aligned} \tag{82}$$

A estimativa de densidade paramétrica $p_i(u_i)$ desempenha um papel essencial para o sucesso da regra de aprendizado da Equação (83). A convergência local é assegurada se $p_i(u_i)$ for uma estimativa da verdadeira densidade das fontes (Pham and Garrat 1997). Por exemplo, a função sigmóide usada no algoritmo de Bell e Sejnowski (Bell and Sejnowski 1995) é adequada para separar fontes supergaussianas, isto é, fdp's com caudas mais pesadas do que a distribuição gaussiana.

Uma maneira de generalizar a regra de aprendizado para fontes com distribuições subgaussianas ou supergaussianas é derivar regras de aprendizado separadas para componentes subgaussianas e supergaussianas. Para isso é necessário adequar, para cada caso, o termo não linear $\varphi(u)$, que é dependente da fdp das estimativas de fontes. No modelo infomax (Bell and Sejnowski 1995), apresentado na Seção (3.2), os termos não lineares $\varphi(u)$ foram derivados somente para distribuições de fontes supergaussianas. Dessa forma, a abordagem infomax original apresenta a limitação de não separar fontes que possuem densidades subgaussianas. O algortimo proposto em (Girolami 1998) apresenta derivações para termos não lineares $\varphi(u)$ que permitem realizar a separação estável de fontes supergaussianas e subgaussianas.

Uma densidade simétrica estritamente subgaussiana pode ser modelada utilizando-se uma forma simétrica do modelo de misturas de Pearson (Pearson 1894), como segue (Girolami 1998):

$$p(u) = \frac{1}{2}\big(N(\mu, \sigma^2) + N(-\mu, \sigma^2)\big), \tag{83}$$

em que $N(\mu, \sigma^2)$ é a densidade normal com média μ e variância σ^2. Fazendo $\mu = 1$ e $\sigma^2 = 1$, a Equação (77) fica reduzida a:

$$\varphi(u) = -\frac{\frac{\partial p(u)}{\partial u}}{p(u)} = u - \tanh(u). \tag{84}$$

Para fontes com densidades supergaussianas, pode-se adotar a seguinte densidade:

$$p(u) \propto N(u)\mathrm{sech}^2(u), \tag{85}$$

em que $N(u)$ é uma densidade normal com média zero com variância unitária. A não linearidade $\varphi(u)$ é, agora:

$$\varphi(u) = -\frac{\frac{\partial p(u)}{\partial u}}{p(u)} = u + \tanh(u). \tag{86}$$

As Equações (84) e (86) podem ser combinadas como:

$$\Delta\mathbf{W} \propto \left[\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T\right]\mathbf{W}$$

$$\begin{cases} k_i = 1 & \text{supergaussiana} \\ k_i = -1 & \text{subgaussiana,} \end{cases} \tag{87}$$

em que k_i são os elementos da matriz diagonal N-dimensional $\mathbf{K}$. Os elementos k_i's foram derivados a partir de uma análise formulada em (Cardoso 1998). A utilização dessa abordagem resultou na seguinte escolha para feita os k_i's, feita em (Lee, Girolami, and Sejnowski 1999):

$$k_i = \text{sign}\Big(E\{\text{sech}^2(s_{i,t})\}E\{s_{i,t}^2\} - E\{[\tanh(s_{i,t})]s_{i,t}\}\Big). \tag{88}$$

o que assegura a estabilidade da regra de aprendizado.

4.1.2 Regras de Aprendizado do Modelo ICAMM

Pode-se escrever as Equações (87) e (91) em termos das matrizes de bases $\mathbf{A}_k$ para cada classe:

$$\Delta\mathbf{A}_k \propto -p(C_k \mid \mathbf{x}_t, \Theta)\mathbf{A}_k[\mathbf{I} - \mathbf{K}\tanh(\mathbf{s}_k)\mathbf{s}_k^T - \mathbf{s}_k\mathbf{s}_k^T], \tag{89}$$

em que:

$$\mathbf{s}_k = \mathbf{A}_k^{-1}(\mathbf{x}_t - \mathbf{b}_k), \tag{90}$$

e

$$k_i = \text{sign}\Big(E\{\text{sech}^2(s_{i,t})\}E\{s_{i,t}^2\} - E\{[\tanh(s_{i,t})]s_{i,t}\}\Big). \tag{91}$$

A distribuição da fonte é supergaussiana quando $k_{k,i} = 1$ e subgaussiana quando $k_{k,i} = -1$. A adaptação do logaritmo da probabilidade *a priori* $\log p(\mathbf{s}_k)$ pode ser aproximada da seguinte forma:

$$\log p(\mathbf{s}_k) \propto -\sum_{i=1}^{N}\left(k_{k,i}\log(\cosh s_{k,i,t}) - \frac{s_{k,i,t}^2}{2}\right). \tag{92}$$

4.1.3 Estimação dos Vetores de Bias

A Equação (69) pode ser utilizada para adaptar os vetores de bias $\mathbf{b}_k$ para cada classe, da seguinte forma:

$$\nabla_{\mathbf{b}_k} L = \sum_{t=1}^{T} p(C_k|\mathbf{x}_t, \Theta)\nabla_{\mathbf{b}_k}\log p(\mathbf{x}_t|C_k, \theta_k). \tag{93}$$

A adaptação é realizada utilizando-se a subida do gradiente da densidade das componentes com respeito ao vetor de bias $\mathbf{b}_k$, resultando em:

$$\Delta\mathbf{b}_k = \propto p(C_k|\mathbf{x}_t, \Theta)\nabla_{\mathbf{b}_k}\log p(\mathbf{x}_t|C_k, \theta_k). \tag{94}$$

Utilizando a Equação (70) na Equação (94), é possível adaptar $\mathbf{b}_k$ como segue:

$$\Delta \mathbf{b}_k \propto p(C_k|\mathbf{x}_t, \Theta)\nabla_{\mathbf{b}_k}[\log p(\mathbf{A}_k^{-1}(\mathbf{x}_t - \mathbf{b}_k)) - \log|\det \mathbf{A}_k|]. \tag{95}$$

Entretanto, ao invés de utilizar a informação de gradiente, em (Lee, Lewicki, and Sejnowski 2000), foi utilizada uma fórmula aproximada para a adaptação do vetor de bias. Nessa aproximação, foi assumido que:

$$\begin{aligned} \nabla_{\mathbf{b}_k} L &= 0 \\ \sum_{t=1}^{T} p(C_k|\mathbf{x}_t, \Theta)\nabla_{\mathbf{b}_k} \log p(\mathbf{x}_t|C_k, \theta_k) &= 0 \end{aligned} \tag{96}$$

A substituição da Equação (70) na Equação (96) mostra que o gradiente do primeiro termo da Equação (70) tem que ser igual a zero. A partir dessa conclusão, segue que:

$$\nabla_{\mathbf{b}_k} \log p(\mathbf{A}_k^{-1}(\mathbf{x}_t - \mathbf{b}_k)) = 0. \tag{97}$$

Na derivação apresentada em (Lee, Lewicki, and Sejnowski 2000), assume-se ainda se que um grande volume de dados x_t estiver disponível e a fdp $p(\mathbf{s}_t)$ for simétrica e diferenciável, então $\log p(\mathbf{s}_t)$ também será simétrico e o vetor de bias pode ser, portanto, aproximado pela média ponderada das observações:

$$\mathbf{b}_k = \frac{\sum_t \mathbf{x}_t p(C_k \mid \mathbf{x}_t, \Theta)}{\sum_t p(C_k \mid \mathbf{x}_t, \Theta)}. \tag{98}$$

5 Modelo de Misturas ICA Aperfeiçoado (EICAMM)

Modelo de Misturas ICA Aperfeiçoado (EICAMM), proposto neste trabalho, foi derivado a partir de algumas modificações importantes realizadas no ICAMM, considerando aspectos de modelagem e implementação. Essas modificações são apresentadas nesta seção.

5.1 Reformulação do Modelo de Classes

Ao invés de considerar o termo de *bias* como sendo adicionado aos dados depois que estes foram gerados por um modelo ICA (ver Equação (59)), no EICAMM, os vetores de *bias* são considerados como sendo adicionados aos sinais das fontes geradoras. A partir dessa modificação, origina-se uma outra equação, diferente daquela correspondente no ICAMM, para descrever os dados em cada classe, dada por:

$$\mathbf{x}_t = \mathbf{A}_k(\mathbf{s}_k + \mathbf{b}_k). \tag{99}$$

A justificativa para essa modificação reside no fato que, na abordagem infomax, supõe-se a ausência de ruídos nos sensores $\mathbf{x}$, ou no máximo a existência de sinais com poucos ruídos aditivos. Portanto, com o objetivo de garantir a ausência de ruídos adicionados aos sensores, estes ruídos são considerados como adicionados às fontes $\mathbf{s}$, antes do processo de mistura de acordo com o modelo generativo ICA apresentado na Equação (4).

5.2 Regra de Aprendizado para os Termos de *Bias*

Com o objetivo de formular uma regra de aprendizado mais informativa para os termos de *bias* $\mathbf{b}_k$, no EICAMM, uma nova regra de aprendizado é derivada utilizando a abordagem proposta em (Bell and Sejnowski 1995) para maximizar a informação mútua que a saída Y de um processador de rede neural possui sobre a sua entrada X. De acordo com essa idéia, quando uma única entrada x passa através de uma função de transferência não linear $g(x)$, resulta numa variável de saída y, de modo que a informação mútua entre essas duas variáveis é maximizada. Por meio dessa transformação, as partes de alta densidade da fdp de x alinha-se com as partes de alta inclinação da função $g(x)$, como apresentado na Figura (5).

Dessa forma, as regras de aprendizado para pesos e *biases* em uma rede neural, de acordo com a abordagem em (Bell and Sejnowski 1995), são formuladas utilizando alguma função de transferência não linear. Na formulação do EICAMM, uma nova regra de aprendizado para os termos de *bias* $\mathbf{b}_k$ para cada classe é proposta, considerando-se, aqui, a função de transferência como sendo a tangente hiperbólica:

$$\Delta \mathbf{b}_k = -2 \tanh(\mathbf{s}_k). \tag{100}$$

As vantagens dessa nova regra para atualização dos vetores de *bias* em relação à regra correspondente no ICAMM são:

- A regra do EICAMM é adaptativa, ao contrário da regra correspondente no ICAMM, no sentido que esta leva em conta, na iteração corrente, os resultados obtidos nas iterações anteriores. Essa vantagem pode ser verificada comparando-se as Equações (98) e (100) para atualização dos termos de *bias* nos modelos ICAMM e EICAMM, respectivamente.
- Essa nova regra é formulada de acordo com a teoria de maximização de informação formalmente proposta em (Bell and Sejnowski 1995), o que leva ao relaxamento da suposição na que se baseia a regra para a adaptação dos termos de *bias* do ICAMM, de que um grande volume de dados de entrada esteja disponível.

5.3 As matrizes de bases no EICAMM são ortogonais

Na derivação do modelo ICAMM, considera-se que $\mathbf{W}_k = \mathbf{A}^{-1}$. Dessa forma, a Equação(87), derivada em (Lee, Lewicki, and Sejnowski 2000) para a adaptação das matrizes de bases pode ser reescrita como:

$$\Delta \mathbf{W} \propto [\mathbf{I} - \mathbf{K} \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]\mathbf{A}^{-1}. \quad (101)$$

Entretanto, para que a passagem na derivação feita em (Lee, Lewicki, and Sejnowski 2000) da Equação(87) para a Equação(89) possa ser realizada, deve-se considerar que as matrizes de bases estimadas pelo modelo sejam ortogonais.

Dessa forma, uma outra modificação no ICAMM foi incorporada ao modelo EICAMM por meio da suposição de que as matrizes $\mathbf{A}_k$ são ortogonais, de modo que $\mathbf{A}^{-1} = A^T$.

Portanto, a regra de adaptação para as matrizes de bases $\mathbf{A}_k$ no EICAMM é dada por:

$$\Delta \mathbf{A}_k \propto p(C_k \mid \mathbf{x}_t, \theta)\mathbf{A}_k(\mathbf{I} - \mathbf{K} \tanh(\mathbf{s}_k)\mathbf{s}_k^T - \mathbf{s}_k\mathbf{s}_k^T)^T, \quad (102)$$

onde $\mathbf{s}_k = \mathbf{A}_k^T\mathbf{x}_t - \mathbf{b}_k$, considerando o modelo para os dados de classe utilizado no EICAMM e formalizado na Equação(99). Aqui, nota-se que o operador de transposição é utilizado, ao invés da matriz inversa $\mathbf{A}_k^{-1}$ para a modelagem implícita dos vetores de fontes $\mathbf{s}_k$, o que implica em uma vantagem computacional do EICAMM, em relação ao ICAMM originalmente proposto.

Consequentemente, no modelo EICAMM, as matrizes de bases $\mathbf{A}_k$ são ortogonalizadas em cada iteração, utilizando a seguinte equação:

$$\mathbf{A}_k = \mathbf{A}_k(\mathbf{A}_k^T\mathbf{A}_k)^{1/2}. \quad (103)$$

5.4 O modelo EICAMM utiliza informações de segunda derivada

Quando a segunda derivada da função objetivo é relativamente fácil de calcular, pode-se incorporar essa informação para acelerar a convergência do algoritmo e garantir a aproximação do mínimo local da função (Masters 1995). Com base nessa motivação, uma modificação na regra de atualização para as matrizes de bases $\mathbf{A}_k$ foi proposta neste trabalho por meio da incorporação da segunda derivada da função de log-verossimilhança dos dados. A seguir, será apresentado como os métodos de Newton e Levenberg-Marquardt podem ser formalizados para serem utilizados no EICAMM.

5.4.1 Método de Newton

Para modelar o método de Newton para o EICAMM, as segundas derivadas da função de log-verossimilhança, dadas pela matriz Hessiana, devem ser incorporadas à regra de aprendizado da Equação (102). Consequentemente, a nova regra de atualização para as matrizes de bases $\mathbf{A}_k$ é dada por:

$$\Delta \mathbf{A}_k \propto p(C_k \mid \mathbf{x}_t, \boldsymbol{\theta})\mathbf{H}^{-1}\mathbf{A}_k(\mathbf{I} - \mathbf{K} \tanh(\mathbf{s}_k)\mathbf{s}_k^T - \mathbf{s}_k\mathbf{s}_k^T)^T, \quad (104)$$

tendo sido calculada, neste trabalho, por meio das seguintes derivações:

$$\Delta \mathbf{W} \propto [\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]\mathbf{W} \tag{105}$$

Calculando a segunda derivada em relação à $\mathbf{W}$ na Equação (105), obtém-se:

$$(\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T) + \frac{\partial(\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T)}{\partial \mathbf{W}}\mathbf{W} = \tag{106}$$

$$(\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T) + (-\mathbf{K}\tanh(\mathbf{u})\mathbf{s}^T\mathbf{A}^T - \mathbf{K}\mathrm{sech}^2(\mathbf{u})\mathbf{A}\mathbf{s}\mathbf{u}^T - \mathbf{u}\frac{\partial \mathbf{u}^T}{\partial \mathbf{W}} - \frac{\partial \mathbf{u}}{\partial \mathbf{W}}\mathbf{u}^T)\mathbf{W} = \tag{107}$$

$$(\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T) - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{K}\mathrm{sech}^2(\mathbf{u})\mathbf{u}\mathbf{u}^T - \mathbf{u}\mathbf{A}\mathbf{s}_t\mathbf{W} - \mathbf{A}\mathbf{s}_t\mathbf{u}^T\mathbf{W}. \tag{108}$$

Usando o resultado $\mathbf{u}_t = \mathbf{W}\mathbf{x}_t = \mathbf{W}\mathbf{A}\mathbf{s}_t$ e substituindo-o na Equação (X), obtém-se:

$$\mathbf{I} - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T - \mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \ - \mathbf{K}\mathrm{sech}^2(\mathbf{u})\mathbf{u}\mathbf{u}^T - 3\mathbf{u}\mathbf{u}^T =$$

$$\mathbf{I} - 2\mathbf{K}\tanh(\mathbf{u})\mathbf{u}^T - \mathbf{K}\mathrm{sech}^2(\mathbf{u})\mathbf{u}\mathbf{u}^T - 3\mathbf{u}\mathbf{u}^T. \tag{109}$$

5.4.2 Método de Levenberg-Marquardt

Por meio da incorporação do método de Levenberg-Marquardt à regra para atualização para as matrizes $\mathbf{A}_k$, uma outra modificação também foi proposta, no modelo EICAMM, para garantir que a matriz Hessiana $\mathbf{H}$ seja positiva definida[12], uma vez que essa é uma condição necessária para que $\mathbf{H}$ seja inversível (Bazaraa 1979). Essa modificação é formulada como:

$$\Delta \mathbf{A}_k \propto p(C_k \mid \mathbf{x}_t, \boldsymbol{\theta})(\mathbf{H} + \mu\mathbf{I})^{-1}\mathbf{A}_k(\mathbf{I} - \mathbf{K}\tanh(\mathbf{s}_k)\mathbf{s}_k^T - \mathbf{s}_k\mathbf{s}_k^T)^T, \tag{110}$$

onde μ é uma pequena constante no intervalo $(0, 1)$.

Nos experimentos deste trabalho, o modelo EICAMM utiliza o método de Levenberg-Marquardt em sua regra de aprendizado para as matrizes $\mathbf{A}_k$, como formalizado na Equação (110).

6 Resultados Experimentais

Para comparar o desempenho do EICAMM com aquele apresentado pelo ICAMM, ambas abordagens foram utilizadas para para classificar dados gerados aleatoriamente

[12]Uma matriz $\mathbf{A}$ é positiva definida se $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$, para qualquer vetor $\mathbf{x}$ diferente de zero.

e o bem conhecido conjunto de dados Iris. Primeiramente, foram geradas duas classes de dados 2D, 3D e 5D com distribuição de Laplace (supergaussiana) com o objetivo de avaliar os desempenhos dos métodos para o caso do número de classes $K = 2$ e dimensões $N = 2$ (ver Figura (7)), $N = 3$ e $N = 5$. Para testar o caso no qual $K = 3$, foi adicionada aos dados previamente gerados mais uma classe, desta vez com distribuição uniforme (subgaussiana), como pode ser observado na Figura (8). Nesses experimentos com dados simulados, foram gerados 1000 pontos de dados para cada classe, embora somente 100 tenham sido utilizados para gerar as Figuras (7) e (8).

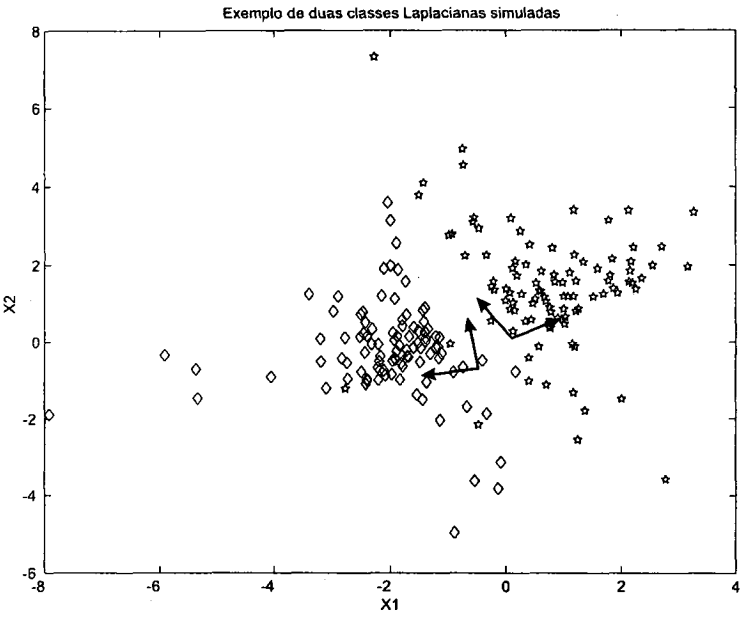


Figura 7: Exemplo de dados simulados com duas classes Laplacianas.

O conjunto de dados Iris (Duda and Hart 1973) contém três classes que representam tipos de flores, com 150 exemplos em cada uma, e quatro atributos numéricos. Uma das classes é linearmente separável das outras duas, as quais não são linearmente separáveis entre si. Os experimentos para esse conjunto de dados foram realizados para todas as três classes e para somente as duas classes que são linearmente separáveis.

Tabela 1: Resultados de Classificação para Dados Simulados - ICAMM.

Número de Dimensões	**Número de Classes**	**Precisão**
2	2	50.05%
2	3	75.00%
3	2	50.00%
3	3	75.00%
5	2	50.00%
5	3	75.00%

As Tabelas (1) e (2) apresentam os resultados de classificação para os dados simulados, em termos de precisão, para os modelos ICAMMM e EICAMM, respectivamente. Pode-se

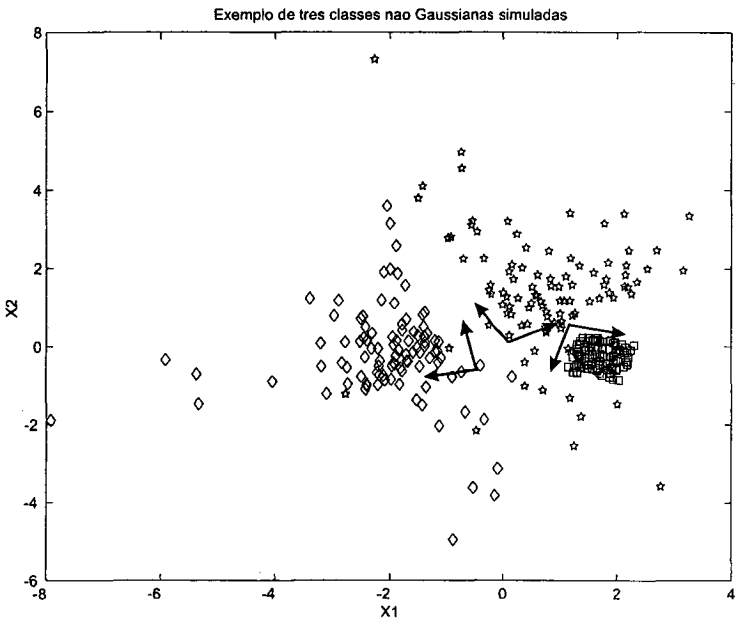


Figura 8: Exemplo de dados simulados com duas classes Laplacianas e uma classe uniforme.

Tabela 2: Resultados de classificação para dados simulados – EICAMM.

Número de Dimensões	**Número de Classes**	**Erro Aparente**
2	2	80.01%
2	3	85.40%
3	2	79.50%
3	3	71.13%
5	2	95.70%
5	3	61.37%

notar que o desempenho do EICAMM superou, de forma significativa, o desempenho do método ICAMM.

Os resultados de classificação, em termos de precisão, para o conjunto de dados de flores Iris e ambos os modelos são apresentados na Tabela (3). Novamente, o modelo EICAMM obteve melhores resultados do que aqueles atingidos pelo ICAMM.

Tabela 3: Resultados de classificação para o conjunto de dados Iris.

Número de Classes	**Modelo**	**Precisão**
2	ICAMM	54.00%
2	EICAMM	99.00%
3	ICAMM	25.00%
3	EICAMM	75.76%

7 Conclusões

Neste trabalho, foi apresentado o Modelo Aperfeiçoado de Misturas ICA (EICAMM), proposto pelos autores. O método incorpora algumas modificações ao modelo ICAMM, considerando determinados aspectos de modelagem e implementação. Para melhor compreensão do modelo proposto, também é realizada uma revisão de conceitos e abordagens relacionados à técnica ICA.

Com o objetivo de avaliar a eficiência do modelo proposto, um estudo comparativo apresentou e discutiu alguns resultados obtidos pelos modelos EICAMM e ICAMM. Notou-se que as modificações propostas podem melhorar significativamente o desempenho de classificação do modelo original, considerando experimentos com dados simulados e o conjunto de dados Iris.

Testes adicionais, considerando dados de imagens para segmentação, estão sendo realizados para avaliar o desempenho do EICAMM nesta aplicação de processamento de imagens.

A Derivação das Regras de Aprendizado ICA

A.1 Para uma Rede Neural com uma Entrada e uma Saída

A.1.1 Função de transferência logística

A função de transferência logística é dada pela seguinte equação não linear:

$$y = \frac{1}{1+e^{-u}}, \quad u = wx + w_0. \tag{111}$$

A derivada parcial de y em relação à u é calculada da seguinte forma:

$$\begin{aligned} y' = \frac{\partial y}{\partial u} &= \frac{1}{(1+e^{-u})^2}\, e^{-u} = \frac{1}{(1+e^{-u})}\left(\frac{e^{-u}}{1+e^{-u}}\right) = \\ &= \frac{1}{(1+e^{-u})}\left(1 - \frac{1}{(1+e^{-u})}\right) = y(1-y). \end{aligned} \tag{112}$$

Para a derivação da regra de aprendizado para w, a seguinte equação diferencial deve ser resolvida:

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}\left(\ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1} \frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right), \tag{113}$$

sendo que, neste caso, y é a função logística, apresentada na Equação (111).

Tem-se que:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x} = y(1-y)w = wy(1-y), \tag{114}$$

$$\frac{\partial y}{\partial w} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial w} = y(1-y)x = yx(1-y), \tag{115}$$

$$\begin{aligned} \tfrac{\partial}{\partial x}\left(y(1-y)\right) &= \frac{\partial y}{\partial x}(1-y) + y\left(\frac{-\partial y}{\partial x}\right) = \\ &= wy(1-y)(1-y) - ywy(1-y) \quad (\text{usando}(114)) = \\ &= wy(1-y)(1-y-y) = \\ &= wy(1-y)(1-2y). \end{aligned} \tag{116}$$

Logo:

$$\begin{aligned}
\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) &= \frac{\partial}{\partial x}\left(\frac{\partial y}{\partial w}\right) = \\
&= \frac{\partial}{\partial x}\Big(yx(1-y)\Big) \quad \text{(usando(115))} \quad = \\
&= y(1-y) + x\frac{\partial}{\partial x}\Big(y(1-y)\Big) \quad = \\
&= y(1-y) + xwy(1-y)(1-2y) \quad \text{(usando(116))} \quad = \\
&= y(1-y)(1+wx(1-2y)). \qquad (117)
\end{aligned}$$

Dividindo a Equação (117) pela Equação (114), obtém-se:

$$\Delta w = \frac{1}{w} + x(1-2y) \qquad (118)$$

Para a derivação da regra de aprendizado para w_0, a seguinte equação diferencial deve ser resolvida:

$$\Delta w_0 \propto \frac{\partial H}{\partial w_0} = \frac{\partial}{\partial w_0}\left(\ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1}\frac{\partial}{\partial w_0}\left(\frac{\partial y}{\partial x}\right), \qquad (119)$$

sendo que, neste caso, y é a função logística, apresentada na Equação (111).

Tem-se que:

$$\frac{\partial y}{\partial w_0} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial w_0} = y(1-y). \qquad (120)$$

Dessa forma,

$$\begin{aligned}
\frac{\partial}{\partial w_0}\left(\frac{\partial y}{\partial x}\right) &= \frac{\partial}{\partial x}\left(\frac{\partial y}{\partial w_0}\right) = \\
&= \frac{\partial}{\partial x}\Big(y(1-y)\Big) \quad \text{(usando(120))} \quad = \\
&= \frac{\partial y}{\partial x}(1-y) + y\left(\frac{-\partial y}{\partial x}\right) = \\
&= wy(1-y)(1-2y). \qquad (121)
\end{aligned}$$

Dividindo a Equação (121) pela Equação (114), obtém-se:

$$\Delta w_0 = 1 - 2y. \qquad (122)$$

A.2 Função de transferência tangente hiperbólica

A função de transferência tangente hiperbólica é dada pela seguinte equação não linear:

$$y = \tanh(u), \quad u = wx + w_0 \tag{123}$$

A derivada parcial de y em relação à u é calculada da seguinte forma:

$$y' = \frac{\partial y}{\partial u} = \operatorname{sech}^2(u) = 1 - \tanh^2(u), = 1 - y^2. \tag{124}$$

Para a derivação da regra de aprendizado para w, a seguinte equação diferencial deve ser resolvida:

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}\left(\ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1}\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right), \tag{125}$$

sendo que, neste caso, y é a função tangente hiperbólica, apresentada na Equação (123).

Tem-se que:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x} = (1 - y^2)w \quad \text{(usando(124))} \tag{126}$$

$$\frac{\partial y}{\partial w} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial w} = (1 - y^2)x \tag{127}$$

$$\begin{aligned}
\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) &= \frac{\partial}{\partial x}\left(\frac{\partial y}{\partial w}\right) = \\
&= \frac{\partial}{\partial x}\Big((1 - y^2)x\Big) \quad \text{(usando(127))} \quad = \\
&= (1 - y^2) + x\frac{\partial}{\partial x}\Big((1 - y^2)\Big) \quad \text{(usando a regra do produto para derivada)} \quad = \\
&= (1 - y^2) + x(-2y)\frac{\partial y}{\partial x} \quad = \\
&= (1 - y^2) - 2xy(1 - y^2)w = \\
&= (1 - y^2)(1 - 2xyw).
\end{aligned} \tag{128}$$

Dividindo a Equação (128) pela Equação (126), obtém-se:

$$\Delta w = \frac{1}{w} - 2xy. \tag{129}$$

Para a derivação da regra de aprendizado para w_0, a seguinte equação diferencial deve ser resolvida:

$$\Delta w_0 \propto \frac{\partial H}{\partial w_0} = \frac{\partial}{\partial w_0}\left(\ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1} \frac{\partial}{\partial w_0}\left(\frac{\partial y}{\partial x}\right), \tag{130}$$

sendo que, neste caso, y é a função tangente hiperbólica, apresentada na Equação (123).

Tem-se que:

$$\frac{\partial y}{\partial w_0} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial w_0} = (1 - y^2). \tag{131}$$

Dessa forma,

$$\begin{aligned}
\frac{\partial}{\partial w_0}\left(\frac{\partial y}{\partial x}\right) &= \frac{\partial}{\partial x}\left(\frac{\partial y}{\partial w_0}\right) = \\
&= \frac{\partial}{\partial x}\left(1 - y^2\right) \quad \text{(usando(131))} \quad = \\
&= -2y\frac{\partial y}{\partial x} = \\
&= -2y(1 - y^2)w.
\end{aligned} \tag{132}$$

Dividindo a Equação (132) pela Equação (126), obtém-se:

$$\Delta w_0 = -2y. \tag{133}$$

A.3 Para uma Rede Neural com N Entradas e N Saídas

Como $\mathbf{W}$ é uma matriz quadrada e g é uma função inversível, a fdp multivariada de $\mathbf{y}$ pode ser escrita como (Papoulis 1991):

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{|\mathbf{J}|}, \tag{134}$$

onde $|\mathbf{J}|$ é o valor absoluto da matriz Jacobiana de derivadas parciais (ver Equação (41)). Esse valor pode ser simplificado pelo produto entre o determinante da matriz de pesos e as derivadas, y_i', das saídas, y_i em relação às entradas da rede:

$$\mathbf{J} = (\det \mathbf{W}) \prod_{i=1}^{N} y_i'. \tag{135}$$

Por exemplo, no caso da função sigmóide logística:

$$y_i = \frac{1}{1+e^{-u_i}}, \tag{136}$$

$$y_i' = y_i(1-y_i). \tag{137}$$

A função a ser maximizada, no caso da abordagem infomax, refere-se à entropia conjunta da saída, dada por:

$$\begin{aligned} H(\mathbf{y}) = & -E[\ln p_{\mathbf{y}}(\mathbf{y})] = \\ = & E[\ln|\mathbf{J}|] - E[\ln p_{\mathbf{x}}(\mathbf{x})] \quad \text{(usando a relação em (134))} \end{aligned} \tag{138}$$

Aqui, os pesos devem ser ajustados para maximizar $H(\mathbf{y})$. Nesse caso, as mudanças aplicadas aos pesos em $\mathbf{W}$ só irão afetar o termo $E[\ln|\mathbf{J}|]$ na Equação (138). Portanto, substituindo a Equação (135) na Equação (138), resulta em:

$$\Delta\mathbf{W} \propto \frac{\partial H}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}} \ln|\mathbf{J}| = \frac{\partial}{\partial \mathbf{W}} \ln|\det \mathbf{W}| + \frac{\partial}{\partial \mathbf{W}} \ln \prod_i |y_i'|. \tag{139}$$

Uma vez que $\det \mathbf{W} = \sum_j w_{ij} \operatorname{cof} w_{ij}$, para qualquer linha i, tem-se, para um único peso w_{ij}:

$$\frac{\partial}{\partial w_{ij}} \ln|\det \mathbf{W}| = \frac{\operatorname{cof} w_{ij}}{\det \mathbf{W}}. \tag{140}$$

Para a derivada em relação à matriz completa $\mathbf{W}$, utiliza-se a definição de uma matriz inversa[13] e o fato que a matriz adjunta, adj $\mathbf{W}$, é a transposta da matriz de cofatores[14]. Isso resulta em:

$$\frac{\partial}{\partial \mathbf{W}} \ln|\det \mathbf{W}| = \frac{(\text{adj } \mathbf{W})^T}{\det \mathbf{W}} = \left[\mathbf{W}^T\right]^{-1}. \tag{141}$$

Para o segundo termo na Equação (139), nota-se que o produto, $\ln \prod_i |y_i'|$, divide-se em uma somatória de termos logarítmicos, sendo que somente um desses termos depende de um particular w_{ij}. O cálculo da derivada correspondente a esse termo depende da função não linear utilizada para a transferência de informação e procede de maneira análoga ao caso de uma rede neural com uma unidade.

[13] A inversa de uma matriz quadrada $\mathbf{A}$ é uma matriz denotada por $\mathbf{A^{-1}}$, com a propriedade que $\mathbf{A} * \mathbf{A}^{-1} = \mathbf{A}^{-1} * \mathbf{A} = \mathbf{I}$.

[14] A matriz adjunta de uma matriz quadrada $\mathbf{A}$ tem a propriedade que: $\mathbf{A} *$ adj $(\mathbf{A}) =$ adj $(\mathbf{A}) * \mathbf{A} =$ det$(\mathbf{A}) * \mathbf{I}$. Portanto, a inversa de $\mathbf{A}$ pode ser escrita como: $\mathbf{A}^{-1} = (1/\det(\mathbf{A})) *$ adj$(\mathbf{A})$.

Referências

Amari, S. and J.-F. Cardoso (1997). Blind source separation – semiparametric statistical approach. *IEEE Transactions on Signal Processing 45*(11), 2692–2700.

Amari, S., A. Cichocki, and H. Yang (1996). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems 8*, 757–763.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation 10*(2), 251–276.

Back, A. D. and A. S. Weigend (1997). A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural Systems 8*, 473–484.

Barlow, H. B. (1961). Possible principles underlying the transformation of sensory messages. In W. A. Rosenblith (Ed.), *Sensory Communication.* MIT Press.

Bazaraa, M. S. (1979). *Nonlinear Programming.* John Wiley & Sons.

Bell, A. J. and T. J. Sejnowski (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation 7*(6), 1129–1159.

Bell, A. J. and T. J. Sejnowski (1997). The 'independent components' of natural scenes are edge filters. *Vision Research 37*, 3327–3338.

Bell, A. and T. Sejnowski (1996). Learning higher-order structure of a natural sound. *Network: Computation in Neural Systems 7*, 261–266.

Bishop, C. M. (1994). Mixture density networks. Neural Computing Research Group Report NCRG/94/004, Aston University.

Cardoso, J.-F. (1997). Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters 4*(4), 112–114.

Cardoso, J.-F. (1998). Blind signal separation: Statistical principles. *Proceedings of IEEE 86*(10), 2009–2025.

Cichocki, A., R. Unbehauen, and E. Rummert (1994). Robust learning algorithm for blind separation of signals. *Eletronics Letters 30*(17), 1386–1387.

Comon, P. (1994). Independent component analysis—a new concept ? *Signal Process 36*, 287–314.

Cover, T. M. and J. A. Thomas (1991). *Elements of information theory.* New York: Wiley.

DeGroot, M. H. (1987). *Probability and statistics,* 2nd. Edition. Oxford: Addison-Wesley, Reading.

Diamantaras, K. I. and S. Kung (1996). *Principal Component Neural Networks: Theory and Applications.* New York.

Duda, R. and P. Hart (1973). *Pattern classification and scene analysis.* New York: Wiley.

Gaeta, M. and J. Lacoume (1990). Sources separation without *a priori* knowledge: The maximum likelihood solution. *Proceedings of Eusipco 90*, 621–624.

Girolami, M. and C. Fyfe (1997). Generalized independent component analysis through unsupervised learning with emergent Bussgang properties. *Proceedings of International Conference on Neural Networks 3*, 1788–1791.

Girolami, M. (1998). An alternative perspective on adaptive independent component analysis algorithms. *Neural Computation 10*(8), 2103–2114.

Herault, J. and C. Jutten (1986). Space or time adaptive signal processing by neural network models. *Neural Networks for Computing: AIP Conference Proceedings*, 206–211.

Hyvärinen, A., J. Karhunen, and E. Oja (2001). *Independent component analysis.* John Wiley & Sons.

Hyvärinen, A., E. Oja, P. Hoyer, and J. Hurri (1998). Image feature extraction by sparse coding and independent component analysis. *Proceedings of International Conference on Pattern Recognition (ICPR'98)*, 1268–1273.

Jain, A. K., R. P. W. Duin, and J. Mao (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(1), 4–37.

Johnson, R. A. and D. W. Wichern (1998). *Applied multivariate statistical analysis.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Jung, T.-P., S. Makeig, M. J. McKeown, A. Bell, T.-W. Lee, and T. J. Sejnowski (2001). Imaging brain dynamics using independent component analysis. *Proceedings of the IEEE 89*(7), 1107–1122.

Jutten, C. and J. Herault (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing 24*, 1–10.

Karhunen, J. and J. Joutsensalo (1994). Representation and separation of signals using nonlinear PCA type learning. *Neural Networks 7*, 113–127.

Lee, T.-W., M. Girolami, A. J. Bell, and T. J. Sejnowski (1998). A unifying information-theoretic framework for independent component analysis. *International Journal of Mathematical and Computer Modelling 39*, 1–21.

Lee, T., M. Girolami, and T. J. Sejnowski (1999). Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation 11*(2), 409–433.

Lee, T., M. S. Lewicki, and T. J. Sejnowski (2000). ICA mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind signal separation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(10), 1078–1089.

Lewicki, M. and T. J. Sejnowski (2000). Learning overcomplete representations. *Neural Computation 12*, 337–365.

Linsker, R. (1992). Local synaptic learning rules suffice to maximize mutual information in a linear network. *Neural Computation 4*, 691–702.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability 1*, 281–297.

Makeig, S., T.-P. Jung, A. J. Bell, D. Ghahramani, and T. Sejnowski (1997). Blind separation of auditory event-related brain responses into independent components. *Proceedings of National Academy of Sciences (USA) 94*, 10979–10984.

Manly, B. (1986). *Multivariate statistical methods: A primer.* London: Chapman and Hall.

Masters, T. (1995). *Advanced algorithms for neural networks.* New York: Wiley.

Nadal, J. P. and N. Parga (1994). Non-linear neurons in the low-noise limit: A factorial code maximizes information transfer. *Network 4*, 295–312.

Oja, E., K. Kiviluoto, and S. Mãlãroiu (2000). Independent component analysis for financial time series. *Proceedings of IEEE Symposium on Adaptive Systems for Signal Processing, Communications, and Control (AS-SPCC'00)*, 111–116.

Oja, E. (1997). The nonlinear PCA learning rule in independent component analysis. *Neurocomputing 17*, 25–45.

Oliveira, P. R. (1997). *Redes Neurais Artficiais para extração de Características.* ICMC-USP: Dissertação de Mestrado.

Olshausen, B. A. and D. J. Field (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature 381*, 607–609.

Olshausen, B. A. and D. J. Field (1997). Sparse coding with an overcomplete basis set: A estrategy employed by v1? *Vision Research 37*, 3311–3325.

Papoulis, A. (1991). *Probability, random variables and stochastic processes.* New York: McGraw-Hill.

Pearlmutter, B. and L. Parra (1996). A context-sensitive generalization of ICA. *Proceedings of ICONIP'96*, 151–157.

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society A 185*(71), 71–110.

Pham, D. T. and P. Garrat (1997). Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Transactions on Signal Processing 45*(7), 1712–1725.

Ridder, D. D., J. Kittler, and R. Duin (2000). Probabilistic PCA and ICA subspace mixture models for image segmentation. In M. Mirmehdi and B. Thomas (Eds.), *Proceedings of British Machine Vision Conference 2000 (BMVC 2000)*, Bristol, UK, pp. 112–121.

Roth, Z. and Y. Baram (1996). Multidimensional density shaping by sigmoids. *IEEE Transactions on Neural Networks 7*(5), 1291–1298.

Shah, C. A., M. K. Arora, S. A. Robila, and P. K. Varshney (2002). ICA mixture model based unsupervised classification of hyperspectral imagery. *Proceedings of 31st Applied Imagery Pattern Recognition Workshop,* Washington, EUA, 112–121.

Vigário, R., J. Särelä, V. Jousmäki, M. Hämäläinen, and E. Oja (2000). Independent component approach to the analysis of EEG and MEG recordings. *IEEE transactions on biomedical engineering 47*(5), 589–593.

Watanabe, S. (1985). *Pattern recognition: human and mechanical.* New York: Wiley.

Xu, L. (1993). Least MSE reconstruction: A principle for self-organizing nets,. *Neural Networks 6*, 627–648.

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

081/2004 HOTO, R.; ARENALES, M.; MACULAN, N. – The compartmentalized knapsack problem: a case study.

080/2004 KAIBARA, M.K.; FERREIRA, V.G.; NAVARRO, H.A. – Upwinding finite-difference schemes for convection dominated problems – Part I: theoretical results.

079/2004 PAIVA, D. M. B.; FREIRE, A. P.; FORTES, R. P. M. – Web engineering process – a case study from academic development

078/2004 SOUZA, R.; SILVA, C.; ARENALES, M.N. – Método do tipo dual simplex para problemas de otimização linear canalizados: teoria

077/2004 SILVA, A M.P.; NUNES, M.G.V. - Using multiword lists for lexically aligning brazilian portuguese and english texts..

076/2004 BÍSCARO, H.H; CASTELO FILHO, A.; NONATO, L.G. – A topological approach to curve reconstruction from scattered points.

075/2004 NONATO, L.G.; CASTELO FILHO, A.; CAMPOS, J.E.P.P.; BÍSCARO, H.; MINGHIM, R. – Topological tetrahedron characterization with applications in volumetric reconstruction.

074/2003 TOMÉ, M.F.; CUMINATO, J.A.; CASTELO, A.; FERREIRA, V.G.; McKEE, S. - Recent Developments in the MAC Technique.

073/2003 SOUZA, F.S.; MANGIAVACCHI, N.; NONATO, L.G.; CASTELO FILHO, A.; TOMÉ, M.F.; FERREIRA, V.G.; CUMINATO, J.A.; McKEE, S. – A front-tracking finite difference method to solve the 3D navier-stokes equations for multi-fluid flows with free surfaces.

072/2003 MANGIAVACCHI, N.; CASTELO FILHO, A.; TOMÉ, M.F.; CUMINATO, J.A.; OLIVEIRA, M.L.B.; McKEE, S. – A numerical technique for incluiding surface tension effects for axisymmetric and planar flows using the gensmac method.