Uma Proposta de Melhoria do Algoritmo Guloso de Estimação de Mistura de Gaussianas

Andre Paim Lemos[1], **Antonio Pádua Braga**[1]

[1]Universidade Federal de Minas Gerais - UFMG
Departamento de Engenharia Eletrônica
Av. Antônio Carlos, 6627 - Belo Horizonte - MG - Brasil

{andrepl,apbraga}@cpdee.ufmg.br

Abstract. *Este trabalho propõe modificações no critério de parada do algoritmo guloso de estimação de Misturas de Gaussianas, com o objetivo de melhorar sua acurácia na busca pelo número de componentes ótimo. Neste trabalho o critério de parada desse algoritmo é modificado para utilizar um teste de normalidade multivariado amostral, de forma que o algoritmo para quando todas as componentes da mistura passem nesse teste. O algoritmo modificado é comparado com o algoritmo original, que utiliza critérios de parcimônia como critério de parada. Resultados de simulações numéricas sugerem a melhoria na acurácia quando o criterio de parada proposto neste trabalho é utilizado.*

Resumo. *This work proposes modifications on the stop criterion of the greedy algorithm for Gaussian Mixtures, in order to increase the accuracy in the search for the optimum number of mixture components. In this work, the stop criterion is modified in order to use a sampling multivariate normality test. The algorithm stops when all mixture components pass on the proposed test. The modified algorithm is compared with the original one, that uses parsimony criterion as stop criterion. Numerical simulation results suggest the accuracy improvement when the stop criterion proposed in this work is used.*

1. Introdução

Modelos de Mistura de Gaussianas são usualmente estimados por meio de métodos iterativos de Maximização da Expectativa (*Expectation Maximization, EM*) [Dempster et al. 1977]. Embora esta abordagem para a maximização da função de verossimilhança seja eficiente em muitas situações, são comuns os problemas de convergência para ótimos locais, além de haver a necessidade de se prover, a priori, o número de componentes da mistura [Ververidis and Kotropoulos 2008].

Diversas abordagens podem ser encontradas na literatura para tratar as deficiências desse algoritmo. Em geral, as técnicas mais comuns tratam o problema através de buscas heurísticas pelo número ótimo de componentes [Ververidis and Kotropoulos 2008] ou através de metodologias de inicialização dos parâmetros do algoritmo visando a evitar a convergência para ótimos locais [Figueiredo et al. 2000].

Em [Verbeek et al. 2003] um algoritmo guloso de estimação de Misturas de Gaussianas é proposto. Esse algoritmo utiliza uma busca incremental pelo número de

componentes da mistura, iniciando de apenas uma, e inserindo componentes no modelo através da divisão das componentes já existentes. Como a componente inicial é a única, esta pode ser estimada a partir de todos os dados via estimador de máxima verossimilhança, resolvendo o problema de inicialização dos parâmetros. Além disso, a busca incremental evita que seja necessário o conhecimento do número de componentes a priori, lidando assim com as duas deficiências do algoritmo EM. Porém, o critério de parada desse algoritmo é baseado em critérios de parcimônia de modelos, como AIC [Ververidis and Kotropoulos 2005] e MDL [Biernacki et al. 1999], o que, segundo [Li and Ma 2008], em muitos casos, não garantem a busca pelo resultado ótimo, relativo ao número de componentes final da mistura.

Assim, esse trabalho propõe uma modificação no critério desse algoritmo, para melhorar sua acurácia na busca pelo número de componentes ótimo do modelo. Para isso, o critério de parada desse algoritmo é modificado para utilizar um teste de normalidade multivariado, baseado na *Distância de Mahalanobis*, proposto por [Ververidis and Kotropoulos 2008], de forma que o algoritmo só acrescenta componentes no modelo, ou seja, uma nova iteração é iniciada, se pelo menos alguma das componentes presentes rejeitar a hipótese nula do teste, dado um nível de significância.

Esse trabalho está dividido da seguinte maneira: a seção 2 descreve o modelo de Mistura de Gaussianas, em seguida, a seção 3 descreve o Algoritmo EM e apresenta suas principais deficiências. A seção 4 descreve algumas das abordagens descritas na literatura para tratar as deficiências desse algoritmo. Em seguida, a seção 5 descreve a metodologia proposta nesse trabalho baseada na modificação do algoritmo guloso, proposto em [Verbeek et al. 2003]. A seção 6 descreve experimentos avaliando a modificação proposta. Finalmente, a seção 7 apresenta as conclusões finais.

2. Mistura de Gaussianas

A função de densidade de probabilidade (*probability density function, pdf*) Gaussiana é uma função parametrizada pelo vetor médio μ e a matriz de covariância Σ em um espaço paramétrico de dimensão D, descrita pela seguinte equação:

$$N(x;\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \quad (1)$$

Os parâmetros desse modelo podem ser estimados a partir de um conjunto de amostras, utilizando-se o estimador de máxima verossimilhança [Duda et al. 2001]. Para isso, dado um conjunto de amostras de tamanho N, $x = x_1, \cdots, x_N$, define-se a função de verossimilhança como:

$$L(x;\Theta) = \prod_{n=1}^{N} p(x_n;\Theta) \quad (2)$$

esta função descreve a verossimilhança das amostras em função dos parâmetros. Assim, para encontrar os parâmetros que melhor descrevem o conjunto de amostras, deve-se encontrar o valor máximo dessa função:

$$\hat{\Theta} = \arg\max_{\Theta} L(x;\Theta) \tag{3}$$

Geralmente, essa função não é maximizada diretamente, mas sim seu logaritmo:

$$\ell(x;\Theta) = lnL(x;\Theta) = \sum_{n=1}^{N} lnp(x_n;\Theta) \tag{4}$$

definida como o logaritmo da verossimilhança, sendo geralmente mais fácil de ser manipulada.

Assim, para encontrar os parâmetros da pdf Gaussiana, $\Theta = [\mu \ \Sigma]$, deriva-se o logaritmo da verossimilhança e iguala-se o resultado a zero. Realizada essa manipulação, encontram-se os estimadores de máxima verossimilhança para o vetor médio e a matriz de covariância.

Modelos de mistura de gaussianas (*Gaussian Mixture Models, GMM*) são modelos formados por uma mistura de várias distribuições Gaussianas que podem ser utilizados para modelar dados multimodais [Duda et al. 2001]. A pdf desse modelo é definida como:

$$p(x;\Theta) = \sum_{c=1}^{C} \alpha_c N(x;\mu_c,\Sigma_c) \tag{5}$$

sendo α_c o peso de cada componente c do modelo, de forma que $0 < \alpha_c < 1$ e $\sum_{c=1}^{C} \alpha_c = 1$.

O conjunto de parâmetros:

$$\Theta = [\alpha_1, \mu_1, \Sigma_1, \cdots, \alpha_C, \mu_C, \Sigma_C] \tag{6}$$

definem o modelo.

Os parâmetros desse modelo não podem ser estimados a partir de um conjunto de amostras, de forma análoga a estimação dos parâmetros da pdf Gaussiana, uma vez que não se tem a informação de qual componente gerou cada uma das amostras. Caso essa informação estivesse disponível, a estimação dos parâmetros poderia ser feita de forma trivial e independente para cada componente. Porém, como essa informação não é disponível, os parâmetros são estimados via o Algoritmo de Maximização da Expectativa.

3. Algoritmo EM para Mistura de Gaussianas

O Algoritmo de Maximização da Expectativa (EM) é um algoritmo iterativo utilizado para estimar parâmetros de um modelo, baseado na função de verossimilhança, quando uma solução analítica é infactível ou quando conjunto de amostras possui dados incompletos.

Para isso, esse algoritmo assume que o conjunto de amostras utilizado na estimação dos parâmetros é formado por algumas características conhecidas e outras ocultas. Denota-se por x as características conhecidas, e y as ocultas. O algoritmo é formado então por dois passos. O primeiro passo, (*E-Step*), resolve a seguinte equação:

$$Q(\Theta;\Theta^i) = E_y[\ell(x,y;\Theta)|x;\Theta^i] \tag{7}$$

em que Θ^i é a estimativa da iteração anterior dos parâmetros e Θ é a nova estimativa. Este passo calcula o valor esperado da verossimilhança dos dados, incluindo os dados ocultos marginalizados em função da estimativa corrente dos parâmetros, Θ^i.

O segundo passo do algoritmo, (*M-Step*), maximiza $Q(\Theta,\Theta^i)$ com respeito ao parâmetro Θ, gerando uma nova estimativa para os parâmetros:

$$\Theta^{i+1} \leftarrow \arg\max_{\Theta} Q(\Theta;\Theta^i) \tag{8}$$

Esses passos se repetem até que uma condição seja atingida, como por exemplo:

$$Q(\Theta^{i+1};\Theta^i) - Q(\Theta^i;\Theta^{i-1}) \leq T \tag{9}$$

sendo T um valor definido como parâmetro do algoritmo.

O Algoritmo EM começa a partir de uma estimativa inicial Θ^0 para os parâmetros e garante que o logaritmo da função de verossimilhança aumenta a cada iteração, até que ocorra a convergência [Dempster et al. 1977].

Esse algoritmo pode ser utilizado para estimar os parâmetros de uma Mistura de Gaussianas, assumindo que o conjunto de amostras possui dados ocultos. Esses dados ocultos são definidos como o conhecimento de qual componente do modelo gerou cada amostra. Assim, define-se um vetor de variáveis ocultas $y = \{y_i\}_{i=1}^N$, sendo que para cada amostra x_n, existe uma variável y_n, tal que $y_n \in 1,\cdots,C$, e $y_n = c$, se a amostra x_n foi gerada pela componente c [Bilmes 1998]. O logaritmo da função de verossimilhança completa (dados ocultos e conhecidos) é dada por:

$$\ell(x,y;\Theta) = \sum_{n=1}^{N} ln(\alpha_{y_n} p(x_n|\Theta_{y_n})) \tag{10}$$

O passo *E-Step* deve calcular o valor esperado do logaritmo da verossimilhança condicionado aos dados conhecidos, x, e da estimativa corrente dos parâmetros Θ^i. Como o valor do vetor y não é conhecido, assume-se que y é um vetor aleatório, e a expressão de cada um dos seus componentes pode ser estimada utilizando-se a regra de *Bayes*:

$$p(y_n = k|x_n,\Theta^i) = \frac{\alpha_k^i p(x_n|\Theta_k^i)}{\sum_{c=1}^{C} \alpha_c p(x_n|\Theta_c)}) \tag{11}$$

E a expressão final de y é definida como:

$$p(y|x,\Theta^i) = \prod_{n=1}^{N} p(y_n|x_n,\Theta^i) \tag{12}$$

Assim, a expressão do valor esperado do logaritmo da verossimilhança é dado por [Bilmes 1998]:

$$Q(\Theta;\Theta^i) = \sum_{y\in\Upsilon} \ell(x,y;\Theta)p(y|x,\Theta^i) \tag{13}$$

sendo Υ os possíveis valores que y pode assumir. Essa expressão pode ser maximizada, derivando-se e igualando a zero, gerando assim as estimativas dos parâmetros correspondentes ao passo *M-Step* do Algoritmo EM.

Assim, os passos do Algoritmo EM para a mistura de gaussianas são definidos como:

E-Step: A probabilidade de que cada amostra foi gerada por cada componente é calculada pela equação (11) e uma nova estimativa da verossimilhança é gerada ($Q(\Theta;\Theta^i)$).

M-Step: Os parâmetros do modelo são atualizados, dada uma nova estimativa da verossimilhança, pelos seguintes estimadores:

$$\alpha_c^{i+1} = \frac{1}{N}\sum_{n=1}^{N} p(y_n = c|x_n,\Theta^i) \tag{14}$$

$$\mu_c^{i+1} = \frac{\sum_{n=1}^{N} x_n p(y_n = c|x_n,\Theta^i)}{\sum_{n=1}^{N} p(y_n = c|x_n,\Theta^i)} \tag{15}$$

$$\Sigma_c^{i+1} = \frac{\sum_{n=1}^{N} p(y_n = c|x_n,\Theta^i)(x_n-\mu_c^{i+1})(x_n-\mu_c^{i+1})^T}{\sum_{n=1}^{N} p(y_n = c|x_n,\Theta^i)} \tag{16}$$

O Algoritmo EM descrito possui as seguintes deficiências:

- O número de componentes C da mistura, deve ser definida a priori.
- O valor inicial dos parâmetros, Θ^0, afeta no resultado final, uma vez que o algoritmo pode atingir um máximo local.

4. Estado da Arte

Diversas técnicas podem ser encontradas na literatura para lidar com os problemas do Algoritmo EM, quando utilizado para estimação dos parâmetros de uma Mistura de Gaussianas. Em geral, essas técnicas podem ser divididas em três níveis, baseadas na parte do algoritmo EM em que são aplicadas [Ververidis and Kotropoulos 2008]. O terceiro nível corresponde a técnicas que utilizam heurísticas para estimar o número de componentes da mistura. Já o segundo nível, corresponde a técnicas que realizam uma modificação dos passos do algoritmo para evitar ótimos locais. Finalmente, o primeiro nível corresponde a técnicas que abordam o problema de inicialização dos parâmetros de cada componente.

O número de componentes da mistura pode ser estimado por critérios de parcimônia ou por operações de divisão e agregação (*split and merge*), aplicadas às componentes do modelo.

Métodos baseados em critérios de parcimônia, otimizam uma função que relaciona o logaritmo da verossimilhança com o número de parâmetros do modelo, a fim

de evitar o sobre ajuste (*overfitting*). Esses métodos, em geral podem ser divididos em: métodos baseados em buscas incrementais ou buscas decrementais, ou seja, incrementais iniciam a busca a partir de um modelo formado por apenas uma componente e adicionam componentes até que um critério seja atingido, enquanto métodos decrementais, iniciam a busca com um número alto de componentes e removem componentes até atingir o critério. O critério de parcimônia utilizado pode ser: o Critério de Informação de Akaike (AIC) [Ververidis and Kotropoulos 2005], o Tamanho Mínimo de Descrição (MDL) [Biernacki et al. 1999], entre outros.

Métodos baseados em operações de divisão e agregação de componentes utilizam diferentes critérios para verificar se cada componente do modelo deve ser dividida ou se duas componentes do modelo devem ser agregadas. Métodos de divisão, geralmente, são baseados em informações de *kurtosis* multivariável, pois um valor de *kurtosis* baixo ou alto é uma indicação que o componente não se ajusta adequadamente aos dados e deve ser dividida [Ververidis and Kotropoulos 2008]. Já métodos de agregação de componentes são baseados no produto interno entre os pesos de duas componentes [Ververidis and Kotropoulos 2008].

O método denominado *Deterministic Annealing EM* (DAEM) [Ueda and Nakano 1998] faz parte do segundo nível, ou seja, métodos que realizam alterações nos passos do Algoritmo EM. Esse método modifica o passo *E-Step* adicionando um parâmetro $1/\beta \in [1, \infty)$, denominado temperatura:

$$p(y_n = k|x_n, \Theta^i) = \frac{(\alpha_k^i p(x_n|\Theta_k^i))^\beta}{(\sum_{c=1}^{C} \alpha_c p(x_n|\Theta_c))^\beta} \tag{17}$$

A medida que $1/\beta$ aumenta, $p(y_n = k|x_n, \Theta^i) \to 1/C$, ou seja, uma determinada amostra tende a pertencer a todas as componentes com a mesma probabilidade. Assim, as componentes tornam-se similares e a chance de se escapar de um mínimo local é alta. Geralmente, inicia-se $\beta = 0.9$ e roda o algoritmo até sua convergência, em seguida o valor de β é acrescido de 0.05 e o algoritmo é executado novamente, esses passos são repetidos até que $\beta = 1$.

Finalmente, existem diversos métodos que tratam o problema de inicialização dos parâmetros das componentes. Esses podem ser estimados via algoritmos de agrupamento, como o k-médias [Ueda and Nakano 1998], através de técnicas de reamostragem, como bootstrap [Mclachlan 1987], ou aleatoriamente [Figueiredo et al. 2000], em que todas as componentes são inicializadas com pesos iguais a $1/C$, centros iguais a amostras escolhidas aleatoriamente e matrizes de variância iguais a $\sigma^2 I$, sendo I uma matriz identidade de dimensão $D \times D$ e σ^2 é proporcional à covariância amostral de todas as amostras [Figueiredo et al. 2000].

5. Metodologia Proposta

A metodologia proposta nesse trabalho baseia-se em uma modificação do algoritmo gulososo, ou ganancioso, proposto por [Verbeek et al. 2003] para aprendizado de Misturas de Gaussianas. Esse algoritmo utiliza uma busca incremental, realizando operações de divisão de componentes, para estimar o número de componentes do modelo e seus respectivos parâmetros, e com isso resolve o problema de inicialização dos componentes do

modelo, assim como evita a necessidade do conhecimento a priori do número de componentes.

O algoritmo proposto por [Verbeek et al. 2003] pode ser sumarizado pelos seguintes passos:

1. Calcula-se a mistura ótima formada por um componente f_1, setando $k = 1$.
2. Procura-se por uma nova componente $p(x_n; \Theta^*)$ e seu respectivo peso α^* que maximizem:

$$\{\Theta^*, \alpha^*\} = \arg\max_{\Theta,\alpha} \sum_{n=1}^{N} ln\left[(1-\alpha) f_k(x_n) + \alpha p(x_n; \Theta)\right] \tag{18}$$

3. Defini-se $f_{k+1} = (1-\alpha^*) f_k(x) + \alpha^* p(x_n; \Theta^*)$ e $k = k + 1$.
4. Atualiza-se f_k utilizando o algoritmo EM.
5. Se o critério de parada é atingido, sai, senão vai para passo 2.

A mistura ótima, formada por apenas um componente f_1 é estimada via estimador de máxima verossimilhança, utilizando todo o conjunto amostral (passo 1).

A procura pelo novo componente ótimo (passo 2) é realizada através de uma técnica de divisão das componentes já existentes no modelo. Inicialmente, o conjunto de amostras é particionado em k partições disjuntas definidas como $A_c = \{x_n \in x : c = \arg\max_j \{p(y_n = c|x_n; \Theta_c)\}\}$, ou seja, as amostras são relacionadas à componente com maior a posteriori relacionada, utilizando-se a regra de Bayes. Em seguida, para cada partição A_c $(c = 1, \cdots, k)$, m componentes candidatas são construídas. Para isso, duas amostras são escolhidas aleatoriamente na partição, denominadas x_l e x_r, e a partição é reparticionada em dois subconjuntos disjuntos A_{cl} e A_{cr}, sendo que A_{cl} representa as amostras da partição mais próximas de x_l e A_{cr} mais próximas de x_r. Esses subconjuntos são utilizados como componentes candidatas, com média e covariância estimados a partir dos dados e peso igual a $\alpha_c/2$, ou seja, metade do valor do peso da componente original. Essa operação é repetida $m/2$ vezes até que m componentes candidatas sejam construídas. Uma vez geradas as candidatas, para cada uma, é executado o Algoritmo EM parcial, ou seja, executa-se o Algoritmo EM fixando os parâmetros de f_k e otimizando apenas os parâmetros da componente. Assim, a componente candidata que apresentar o maior valor de verossimilhança final, após a execução do Algoritmo EM parcial, é inserida no modelo.

Uma vez que uma nova componente é inserida no modelo, atualiza-se o modelo f_k utilizando o algoritmo EM (passo 3).

O critério de parada (passo 4) é definido como um número máximo, pré-definido, de componentes ou baseado em um critério de parcimônia de modelos, tais como AIC ou MDL. Segundo [Li and Ma 2008], esses critérios tradicionais geralmente resultam em um número errado de componentes. Assim, esse trabalho propõe uma modificação no critério de parada deste algoritmo para utilizar um teste de normalidade multivariado, proposto por [Ververidis and Kotropoulos 2008], de forma que novas componentes são adicionadas no modelo somente se alguma das componentes do presentes rejeitarem a hipótese nula do teste, dado um nível de significância.

5.1. Teste de Normalidade Multivariado

O teste de normalidade multivariado proposto por [Ververidis and Kotropoulos 2008] é baseado na *Distância de Mahalanobis*. Assim, para estabelecer uma hipótese de que um

conjunto de amostras, x, é distribuído de acordo com uma pdf Gaussiana multivariada, inicialmente calcula-se essa distância para todas as amostras:

$$r_n = (x_n - \hat{x})^T S^{-1} (x_n - \hat{x}) \tag{19}$$

onde $\hat{x}$ é a média amostral e S a covariância amostral do conjunto de amostras.

Em seguida, constrói-se a função de distribuição cumulativa (*Cumulative Distribution Function, cdf*) amostral dessa distância baseando-se nas amostras r_n. A cdf amostral da distância, definida como $\hat{F}(r_n)$, é estimada via função de massa, isto é, ordenando os valores de $\{r_n\}_{n=1}^{N}$ em ordem crescente e definindo $\hat{F}(r_n) = n/N$. Define-se também a cdf teórica da distância $F(r_n)$, dado a média $\hat{x}$ e a matriz de covariância S amostrais, assumindo que a distância tem distribuição Beta [Ververidis and Kotropoulos 2008]. Caso N_{r_n} seja definido como o número de amostras dentro de uma elipse com valores equiprováveis de r_n, então N_{r_n} é descrito como uma variável aleatória com distribuição Binomial com parâmetros N e $F(r_n)$:

$$P(N_{r_n} = k) = \left(\begin{array}{c} N \\ k \end{array} \right) F(r_n)^k (1 - F(r_n))^{N-k} \tag{20}$$

dado que $F(r_n)$ é também a probabilidade de se encontrar uma amostra dentro da elipse com *Distância de Mahalanobis* igual a r_n.

Deve-se então definir um intervalo de confiança, denominado $[k^l_{n,\lambda} k^h_{n,\lambda}]$, para N_{r_n}, dado um nível de significância λ, de forma que esse intervalo deve satisfazer:

$$\sum_{k=k^h_{n,\lambda}}^{N} \left(\begin{array}{c} N \\ k \end{array} \right) F(r_n)^k (1 - F(r_n))^{N-k} = \frac{\lambda}{2}$$

$$\sum_{k=0}^{k^l_{n,\lambda}} \left(\begin{array}{c} N \\ k \end{array} \right) F(r_n)^k (1 - F(r_n))^{N-k} = \frac{\lambda}{2}$$

sendo que $\lambda \in \{0.10, 0.05, 0.01\}$.

Caso N seja suficientemente grande e $F(r_n)$ assuma valores próximos de 0 ou 1, ou seja:

$$NF(r_n)(1 - F(r_n)) \gg 1 \tag{21}$$

segundo o Teorema de *Demoivre-Laplace* a distribuição Binomial pode ser aproximada por uma distribuição Gaussiana com média $NF(r_n)$ e variância $NF(r_n)(1 - F(r_n))$. Um valor típico para assumir essa aproximação seria $NF(r_n)(1 - F(r_n)) > 25$ [Papoulis 1984]. Assim, o intervalo é calculado analiticamente como:

$$k^l_{i,\lambda} = \left[NF(r_n) - z_{1-\lambda} \sqrt{2NF(r_n)(1 - F(r_n))} \right]$$

$$k_{i,\lambda}^{h} = \left[NF(r_n) + z_{1-\lambda}\sqrt{2NF(r_n)(1-F(r_n))} \right] \tag{22}$$

em que o operador [] é definido como o inteiro mais próximo do número entre colchetes.

Caso a condição (21) seja violada, o intervalo pode ser estimado numericamente:

$$\begin{aligned} k_{i,\lambda}^{l} &= \arg\min_{k_1} \left| \sum_{k=0}^{k_1} \left(\begin{array}{c} N \\ k \end{array} \right) F(r_n)^k (1-F(r_n))^{N-k} - \frac{\lambda}{2} \right| \\ k_{i,\lambda}^{h} &= \arg\min_{k_2} \left| \sum_{k=k_2}^{N} \left(\begin{array}{c} N \\ k \end{array} \right) F(r_n)^k (1-F(r_n))^{N-k} - \frac{\lambda}{2} \right| \end{aligned} \tag{23}$$

sendo que k_1 e k_2 podem assumir valores no intervalo $[0, 1, \cdots, N]$.

A hipótese nula do teste deve ser validada caso:

$$\begin{aligned} N_{r_n} \in (k_{n,\lambda}^{l},\ k_{n,\lambda}^{h}) \Rightarrow \frac{N_{r_n}}{N} &\in \left(\frac{k_{n,\lambda}^{l}}{N}\ , \frac{k_{n,\lambda}^{h}}{N} \right) \\ \Rightarrow \hat{F}(r_n) &\in \left(\frac{k_{n,\lambda}^{l}}{N}\ , \frac{k_{n,\lambda}^{h}}{N} \right) \end{aligned}$$

$\forall\ n = 1, 2, \cdots, N$.

Assim a hipótese nula H_0 do teste, de que os dados são gerados por uma distribuição Gaussiana multivariada é aceita, dado um nível de significância λ, se:

$$\hat{F}(r_n) \in \left(\frac{k_{n,\lambda}^{l}}{N}\ , \frac{k_{n,\lambda}^{h}}{N} \right) \tag{24}$$

para pelo menos $(1-\lambda)N$ das N amostras.

A figura 1 ilustra os valores teóricos ($F(r_n)$), amostrais ($\hat{F}(r_n)$) e o intervalo de confiança analítico da cdf dos valores de r_n ($N = 200$ e $\lambda = 0.01$) para amostras geradas por um modelo de Mistura de Gaussianas. A figura 1a ilustra o caso de um modelo contendo apenas uma componente, enquanto a figura 1b ilustra um modelo contendo duas componentes com parâmetros distintos. Analisando a figura 1a percebe-se que os valores de $\hat{F}(r_n)$ estão dentro do intervalo de confiança, indicando a normalidade dos dados. Já analisando a figura 1b percebe-se que o número de valores de $\hat{F}(r_n)$ fora dos intervalos de confiança é maior que o limiar, definido como $(1-\lambda)N) = 2$, de forma que a hipótese nula é rejeitada, dado o nível de significância $\lambda = 0.01$.

Em [Ververidis and Kotropoulos 2008] sugere-se os seguintes valores para o nível de significância, dado o tamanho amostral:

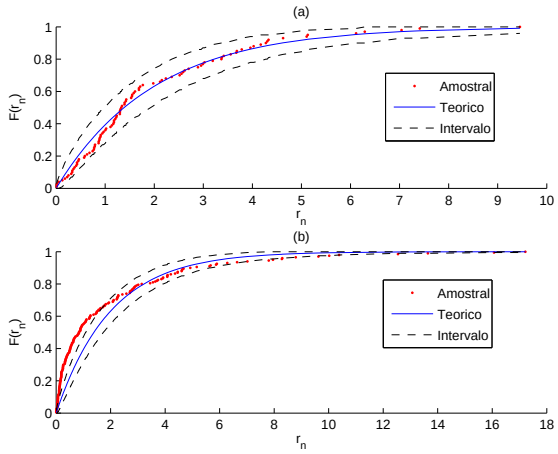

Figura 1. Critério de normalidade multivariado aplicado a Misturas de Gaussianas, contendo uma componente (a) e duas componentes com parâmetros distintos (b)

$$\lambda = \begin{cases} 0.99, & \text{se } N \geq 100 \\ 0.95, & \text{se } 20 \leq N < 100 \\ 0.90, & \text{se } 10 \leq N < 20 \end{cases} \tag{25}$$

Assim, para utilizar esse teste como critério de parada do algoritmo proposto por [Verbeek et al. 2003], inicialmente as amostras são particionadas em partições disjuntas utilizando a metodologia proposta no passo 2 do algoritmo guloso e o teste de normalidade multivariado é aplicado a cada uma das partições. Caso a hipótese nula seja validada para todas as partições, o critério de parada do algoritmo é atingido.

O teste de normalidade proposto possui ordem de complexidade temporal $O(NlogN)$, uma vez que a distância de Mahalanobis deve ser computada para cada amostra e o vetor resultante deve ser ordenado para estimar a cdf amostral.

6. Experimentos

Experimentos foram realizados para verificar se o critério de parada proposto neste trabalho resulta em um ganho na acurácia da seleção do número ótimo de componentes do modelo. Para isso, a metodologia proposta é comparada com critérios de parada baseados em critérios de parcimônia.

Foram utilizados os critérios de parcimônia AIC e MDL:

$$AIC = -\ell(x; \Theta^*) + 2\upsilon \tag{26}$$

$$MDL = -\ell(x; \Theta^*) + \frac{\upsilon}{2} ln(N) \tag{27}$$

sendo $-\ell(x; \Theta^*)$ o valor máximo do logaritmo da verossimilhança e υ é o número de parâmetros livres do modelo, dado por [Ververidis and Kotropoulos 2005]:

$$v = k\left(1 + D + 2D + \frac{D}{2}(1 + D)\right) \tag{28}$$

sendo k o número de componentes e D a dimensão das amostras.

Para os critérios de parcimônia, utilizou-se o critério de parada proposto por [Ververidis and Kotropoulos 2005], de forma que o algoritmo só executa uma nova iteração, caso o valor do critério de parcimônia da iteração corrente seja menor que o valor da iteração anterior, ou seja, para o AIC, o algoritmo finaliza caso:

$$AIC^k - AIC^{k-1} > 0 \tag{29}$$

Foram realizados 200 experimentos, sendo que em cada um gerou-se uma mistura contendo de 2 a 10 Gaussianas de dimensão $D = 2$ e parâmetros distintos e o número de pontos em cada mistura foi ajustado para 15 vezes o número de componentes da mistura. Para cada experimento, o algoritmo foi executado com os três critérios de parada: proposto, AIC e MDL. Para todas as três variações, o número de componentes candidatas por componente do modelo foi ajustado para $m = 6$ e o número máximo de componentes foi definido como 20, de forma que, caso o critério de parada não seja atingido e o modelo já possua 20 componentes, o algoritmo finaliza.

A tabela 1 ilustra os resultados obtidos em relação à acurácia de cada um dos critérios de parada. Nessa tabela são apresentados o número de experimentos em que o número de correto de componentes foi estimado, e a diferença média (média $\pm$ desvio padrão) entre o número correto e o número estimado, para cada um dos critérios de parada.

Tabela 1. Resultados Obtidos nos Experimentos

Método	Experimentos Corretos	Diferença Média
Proposto	55	0.07 ± 1.95
AIC	21	2.88 ± 3.01
MDL	20	2.87 ± 3.02

Analisando a tabela 1, percebe-se que o critério de parada proposto neste trabalho melhora a acurácia do algoritmo, em relação ao número ótimo de componentes, quando comparado com critérios baseados em critérios de parcimônia. O critério proposto é capaz de estimar o número de componentes corretamente cerca de 2.5 vezes mais que os métodos baseados no AIC e MDL. Além disso, a diferença média entre o número de componentes real e estimado para o algoritmo proposto é significativamente menor que a diferença média calculada para os outros métodos.

7. Conclusões

Este trabalho propôs modificações no algoritmo guloso de estimação de Misturas de Gaussianas proposto por [Verbeek et al. 2003], para melhorar sua acurácia, com relação ao número ótimo de componentes do modelo. Para isso, foi utilizado um teste de normalidade multivariável, proposto por [Ververidis and Kotropoulos 2008], para testar as componentes resultantes do modelo no final de cada iteração, de forma que, caso a hipótese

nula do teste seja validada para todas as componentes, o critério de parada do algoritmo é atingido. Os experimentos realizados sugerem uma melhoria na acurácia do algoritmo, quando o critério de parada proposto nesse trabalho é utilizado.

Como sugestões de trabalhos futuros, sugere-se comparar a metodologia proposta com outras metodologias de estimação do número ótimo de componentes. Além disso, o teste de normalidade descrito na seção 5.1 pode ser utilizado também para reduzir o custo computacional do algoritmo guloso, de forma a gerar componentes candidatas apenas a partir de componentes do modelo que rejeitem a hipótese nula do teste.

Agradecimentos

Esse trabalho contou com o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

Referências

Biernacki, C., Celeux, G., and Govaert, G. (1999). An improvement of the nec criterion for assessing the number of clusters in a mixture model. *Non-Linear Anal.*, 20(3):267–272.

Bilmes, J. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern classification*. John Wiley & Sons, Inc., New York, NY, USA.

Figueiredo, M. A. T., Figueiredo, M. A. T., and Jain, A. K. (2000). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:381–396.

Li, L. and Ma, J. (2008). A BYY scale-incremental EM algorithm for Gaussian mixture learning. *Applied Mathematics and Computation*, 205(2, Sp. Iss. SI):832–840.

Mclachlan, G. J. (1987). On bootstrapping the likelihood ratio test stastistic for the number of components in a normal mixture. *Applied Statistics*, 36(3):318–324.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes*. Mc-Graw Hill.

Ueda, N. and Nakano, R. (1998). Deterministic annealing em algorithm. *Neural Netw.*, 11(2):271–282.

Verbeek, J. J., Vlassis, N., and Kröse, B. (2003). Efficient greedy learning of gaussian mixture models. *Neural Comput.*, 15(2):469–485.

Ververidis, D. and Kotropoulos, C. (2005). Emotional speech classification using gaussian mixture models and the sequential floating forward selection algorithm. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1500–1503.

Ververidis, D. and Kotropoulos, C. (2008). Gaussian mixture modeling by exploiting the mahalanobis distance. *Signal Processing, IEEE Transactions on*, 56(7):2797–2811.