IDENTIFICAÇÃO DE LOCUTOR: OTIMIZAÇÃO DO NÚMERO DE COMPONENTES GAUSSIANAS

Ricardo José da Rocha Cirigliano

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____

Prof. Fernando Gil Vianna Resende Junior, Ph.D.

_____

Prof. Abraham Alcaim, Ph.D.

_____

Prof[a]. Mariane Rembold Petraglia, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2007

CIRIGLIANO, RICARDO JOSE DA ROCHA
Identificação de Locutor: Otimização do Número de Componentes Gaussianas [Rio de Janeiro] 2007
XI, 61 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2007)
Dissertação - Universidade Federal do Rio de Janeiro, COPPE
1. Identificação de Locutor
I. COPPE/UFRJ II. Título (série)

Aos meus pais, Anina e Fernando
e minha irmã, Flávia.

Agradecimentos

Ao professor Fernando Gil pelo apoio, orientação e tempo dedicado ao projeto.

Aos meus pais e irmã por todo o apoio, incentivo e paciência.

À Sabrina, por todo apoio, carinho e companheirismo.

À toda comunidade científica, pelos vários e-mails trocados, esclarecimento de dúvidas e dicas extremamente úteis.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

IDENTIFICAÇÃO DE LOCUTOR: OTIMIZAÇÃO DO NÚMERO DE COMPONENTES GAUSSIANAS

Ricardo José da Rocha Cirigliano

Março/2007

Orientador: Fernando Gil Vianna Resende Junior

Programa: Engenharia Elétrica

Neste trabalho é apresentado um algoritmo que busca o número ótimo de componentes gaussianas utilizado para modelar cada locutor da base de dados em um sistema de identificação de locutor independente de texto. Este algoritmo proposto é utiliza a razão entre as distorções máxima e média (distâncias euclidianas entre os vetores mais distantes dos centróides de cada célula e distância média de todos os vetores para o centróide de cada célula) obtidas após o algoritmo de clusterização k-médias.

Foram realizados três conjuntos de testes: teste de ortogonalização, teste de otimização e teste comparativo com outros sistemas. Os resultados mostram que o uso do GMM ortogonal em comparação com GMM convencional, sem a utilização do algoritmo de otimização proposto, diminui em até 4,1% a taxa de erro, com uma redução de 75% do número de componentes gaussianas. A utilização do algoritmo de otimização diminuiu, para a base de dados utilizada, em até 1,8% a taxa de erro e em até 70% o esforço computacional em comparação com o melhor resultado obtido com o GMM ortogonal para um número fixo de componentes gaussianas para todos os locutores.

Os resultados comparativos com outros sistemas de identificação de locutores, GMM-UBM MAP, TTD-GMM e FP GMM, mostram que, para a base de dados utilizada, o algoritmo proposto apresentou a menor taxa de erro, 1,12%, e o segundo menor esforço computacional, cerca de 2% acima do menor valor obtido pelo algoritmo TTD-GMM.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SPEAKER IDENTIFICATION: OPTIMIZATION OF THE NUMBER OF GAUSSIAN COMPONENTS

Ricardo José da Rocha Cirigliano

March/2007

Advisor: Fernando Gil Vianna Resende Junior

Department: Electrical Engineering

In this paper we present an algorithm that searches the optimal number of Gaussian components used to model each speaker in a database in a text independent speaker recognition system. The proposed algorithm is based on the ratio between the maximum and the average distortion (Euclidian distances between the most distant vector and the centroid in each cell and the average distance between all vectors in a cell and its centroid) in each cell obtained through the k-means algorithm.

Three sets of tests were performed: ortogonalization test, optimization test and comparative test with other speaker recognition algorithms. The results show that the usage of orthogonal GMM instead of conventional GMM decreases the error rate up to 4.1%, also reducing the number of Gaussian components in 75%. The proposed optimization algorithm decreased, for the tested database, in 1.8% the error rate and in 70% the computational effort, when compared to the best result obtained for a fixed number of Gaussian components for speaker modeling.

In the comparative test with other speaker recognition algorithms, GMM-UBM MAP, TTD-GMM and FP GMM, the proposed algorithm presented the lowest total error rate, 1.12%, and the second lowest computational effort, around 2% higher than the lowest one, obtained by the TTD-GMM algorithm.

Sumário:

Índice de Figuras:

Índice de Tabelas:

Lista de Acrônimos:

EM	Expectation Maximization
FP GMM	Frame Pruning GMM
GMM	Gaussian Mixture Model
GMM UBM-MAP	GMM UBM – Maximum a Priori
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
ML	Maximum Likelihood
NIST	National Institute of Standards and Technology
QV	Quantizador Vetorial
RSR	Razão Sinal-Ruído
TTD-GMM	Training-Time-Dependent GMM
UBM	Universal Background Model

Lista de Variáveis:

D - ordem do vetor de características

M - número de células / número de componentes gaussianas / dimensão do modelo gaussiano

J - faixa de possíveis números de componentes

T - número de vetores de características em uma locução / número de blocos de uma locução para extração de características

N - tamanho da janela para análise de características

L_m - número de vetores de características na *m*-ésima célula

R - fator usado no algoritmo de otimização

$\mathbf{x}$ - vetor de características de dimensão D

$\mathbf{\Sigma}_m$ - matriz de covariância da *m*-ésima componente

$\mathbf{\Sigma}_{Dm}$ - matriz de covariância diagonalizada da *m*-ésima componente

$\boldsymbol{\mu}_m$ - vetor de médias da *m*-ésima componente

$\boldsymbol{\mu}_{Dm}$ - vetor de médias diagonalizado da *m*-ésima componente

w_m - coeficiente de ponderação da *m*-ésima componente

$\mathbf{\Omega}$ - matriz de diagonalização

λ - densidade gaussiana formada por w_m, $\boldsymbol{\mu}_m$ e $\mathbf{\Sigma}_m$

$b_m(\mathbf{x})$ - função de probabilidade da *m*-ésima componente

1 Introdução

1.1 Motivação e objetivos

O esforço computacional de um sistema de identificação de locutor está diretamente relacionado ao número de componentes gaussianas utilizado para modelar cada locutor presente na base de dados. Nesse sentido, uma solução para diminuir esse esforço computacional seria simplesmente utilizar menos componentes gaussianas. Contudo, essa abordagem gera um segundo problema: aumento da taxa de erro do sistema devido à má modelagem dos alguns locutores.

O objetivo deste trabalho é identificar o número ideal de componentes gaussianas que serão utilizadas para modelar cada usuário da base de dados de um sistema de identificação de locutor, levando-se em conta a taxa de erro do sistema e o esforço computacional. Para tal é proposto um algoritmo de otimização baseado nas distorções máxima e média de cada célula durante a etapa de treinamento do sistema.

Os objetivos desta dissertação podem ser resumidos em:

- Implementação do classificador GMM (*gaussian mixture model* - modelo de misturas gaussianas) convencional;
- Implementação do classificador GMM baseado em vetores de características ortogonalizados;
- Proposta e implementação do algoritmo de otimização do número de componentes gaussianas para cada locutor da base de dados;
- Implementação de algoritmos do estado da arte: GMM-UBM MAP, TTD-GMM e FP GMM;
- Análise comparativa dos resultados utilizando vetores de características ortogonalizados;

- Análise comparativa dos resultados utilizando número variável de componentes gaussianas por locutor.

1.2 Estado da Arte

Nos últimos anos as técnicas para identificação de locutor têm sido amplamente pesquisadas e melhoradas. Diversas abordagens foram desenvolvidas com o objetivo de melhorar o desempenho dos sistemas: novos parâmetros para a modelagem dos locutores, novas formas de modelagem dos locutores, novos classificadores, pré-processadores do sinal da fala, etc.

Anualmente o NIST (*National Institute of Standards and Technology*) promove um concurso de âmbito mundial a fim de avaliar as técnicas de identificação de locutor. Ele define as regras e fornece a base de dados para que cada competidor possa testar o seu sistema. Nos últimos anos, a técnica de identificação de locutor mais amplamente utilizada é o GMM [1] [2] . Outras técnicas utilizadas são o HMM (*hidden Markov models* – modelos escondidos de Markov) [2] , o AR-Vetorial [4] [5] [6] , redes neurais [7] [8] e classificadores polinomiais [9] . Comparações do GMM com outros métodos de modelagem podem ser vistas em [2] [7] [10] [11] . Também têm sido realizadas pesquisas sobre novas características para modelar o sinal de voz [12] [40] [41] e sobre pré-processamentos do sinal de voz [13] .

Outras linhas de pesquisa estão voltadas para a utilização de segmentos de voz, como fones, difones ou trifones para a identificação de locutor [7] [14] [15] [16] [17] [18] , aplicando técnicas desenvolvidas inicialmente para o reconhecimento de fala contínua.

A característica para modelagem do sinal de voz mais utilizada em identificação de locutor independente de texto nos últimos anos é o mel-cepstro. Estudos mostram que para esta finalidade, o mel-cepstro apresenta os melhores resultados [19] .

No estado da arte para o identificação de locutor independente de texto, o classificador mais utilizado é o GMM [8] [43] [44] . Novos estudos têm focado em dois pontos: otimização do

modelo do GMM [20] e melhorias no desempenho dos sistemas quando as características de treinamento e teste são diferentes [45] [46] [47] . Novos parâmetros de modelagem e novos classificadores também vêm sendo desenvolvidos, como mostrado em [9] [40] [41] .

1.3 Descrição do Trabalho

Este trabalho está dividido da seguinte forma: no Capítulo 2 são apresentados os principais métodos de classificação para identificação de locutor. No Capítulo 3 é detalhado o GMM. O método de ortogonalização é descrito no Capítulo 4. O algoritmo de otimização proposto é apresentado no Capítulo 5. O Capítulo 6 apresenta a base de dados utilizada nos testes. No Capítulo 7 são descritos os testes realizados. As conclusões são apresentadas no Capítulo 8.

2 Classificadores

2.1 HMM

Nos anos 70 e especialmente nos anos 80, os pesquisadores da área de voz começaram a voltar suas atenções para a modelagem estocástica da voz, com a intenção de endereçar problemas como a variabilidade, particularmente em sistema com grandes vocabulários. Os HMMs [1] passaram a ser o foco dos estudos em reconhecimento de voz, sendo utilizados em diversos sistemas comerciais que utilizam grandes vocabulários.

Os HMMs funcionam como máquinas de estados que são utilizadas para modelar ocorrências. Estas ocorrências podem ser palavras (utilizadas em sistemas com pequeno vocabulário) ou sub-unidades de palavras, como fones ou trifones (utilizadas em sistemas com grande vocabulário). Os HMMs são formados basicamente por estados, que modelam a informação que aquele HMM traz, e matrizes de transição, que modelam a probabilidade da passagem de um estado para outro. O modelo estatístico que o HMM utiliza pode variar, mas em geral é utilizada a modelagem gaussiana, fazendo com que cada estado seja composto por um conjunto de médias e variâncias.

Nos últimos anos, os HMMs vem sendo utilizados também para reconhecimento e verificação de locutor. Uma vez que os HMMs modelam bem informações temporais, a grande maioria dos sistemas que utilizam essa abordagem trabalham com sub-unidades de palavras [14] [15] [16] [17] [18] .

HMMs de várias formas são utilizados para a modelagem na identificação de locutor dependente e independente de texto. O HMM não modela somente classes acústicas desconhecidas, mas também a seqüência temporal entre essas classes. Embora a modelagem de estruturas temporais seja vantajosa para a tarefa de identificação de locutor dependente do

texto, no caso de independência do texto, esta modelagem não apresenta relevância. Por esse motivo, o HMM apresenta limitações no desempenho de tarefas independentes do texto.

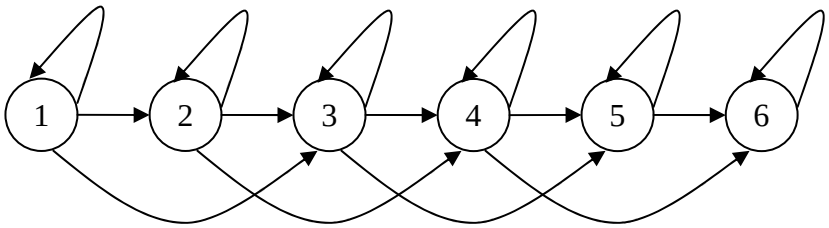

Figura 1 – Exemplo de HMM com 6 estados.

$$\begin{bmatrix} a(1|1) & a(1|2) & \cdots & a(1|S-1) & a(1|S) \\ \vdots & \vdots & a(i|j) & \vdots & \vdots \\ a(S|1) & a(S|2) & \cdots & a(S|S-1) & a(S|S) \end{bmatrix}$$

Figura 2 – Exemplo de matriz de transição de um HMM com S estados.

2.2 AR-Vetorial

O modelo autorregressivo vetorial (AR-vetorial) faz a modelagem da evolução espectral da voz, sendo uma generalização de um modelo muito comum na análise da voz, o modelo de predição linear (LPC). O AR-vetorial é estimado a partir de uma seqüência de vetores extraídos do sinal (geralmente vetores de características), enquanto que o LPC é estimado a partir de blocos contendo o sinal de voz.

Na identificação de locutor, o AR-vetorial é utilizado para medir a similaridade entre modelos de locutores previamente estimados. Seu uso é motivado por extrair de forma aproximada as características dinâmicas do locutor, ou seja, a forma como fala ao passar do tempo.

Apesar de apresentar bons resultados como apresentado em [23] , o objetivo deste trabalho é estudar métodos de otimização de número de componentes gaussianas em modelos estocásticos e, por isso, o AR-vetorial não foi utilizado em testes comparativos.

2.3 GMM

Existem duas abordagens para se apresentar a estrutura do GMM. Na primeira, o GMM é formado basicamente de dois sub-sistemas utilizados para o identificação de locutor, um classificador uni-modal gaussiano e um quantizador vetorial (QV). Assim, o GMM combina a robustez do modelo gaussiano, paramétrico, com a modelagem arbitrária de um modelo de QV, não-paramétrico. Assim como o QV, o GMM faz a separação espacial de duas classes acústicas. Porém, enquanto o QV separa as classes de acordo com as distâncias entre elas, o GMM utiliza as probabilidades geradas por conjuntos de funções de densidade de probabilidade (fdp) gaussianas previamente estimadas.

Na segunda abordagem o GMM pode ser entendido como um HMM de um único estado, tendo como observações misturas de fdp's gaussianas, onde cada uma destas misturas pode representar uma ou várias classes fonéticas para caracterizar o som produzido por uma pessoa. Na aproximação do QV, cada locutor é representado por um dicionário de amostras espectrais representando grupos de classes fonéticas. Esta técnica tem demonstrado bom desempenho na identificação de locutor com vocabulários pequenos, como dígitos, sendo limitada em modelar possíveis variações encontradas na verificação de locutor independente do texto. Tem sido mostrado que modelos estatísticos fornecem uma melhor modelagem acústica da voz.

O GMM, suprindo as deficiências dos métodos anteriores, vem sendo atualmente a ferramenta que apresenta uma das melhores respostas na tarefa de verificação de locutor independente do texto e sua utilização é amplamente justificada em termos físicos (modelagem de classes acústicas) e práticos (bons resultados).

3 GMM

A densidade de mistura gaussiana completa, chamada λ, é parametrizada por

$$\lambda = \{w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}, \ m = 1, \ldots, M \ , \tag{1}$$

onde $\boldsymbol{\mu}_m$ é o vetor de médias, $\boldsymbol{\Sigma}_m$ é a matriz de covariância, w_m é o coeficiente de ponderação de cada componente e M é o número de componentes gaussianas.

Uma mistura de densidades de probabilidades gaussianas é uma soma ponderada de M densidades, dada pela equação

$$p(\mathbf{x} \mid \lambda) = \sum_{m=1}^{M} w_m b_m(\mathbf{x}) \tag{2}$$

onde $\mathbf{x}$ é um vetor aleatório de ordem D e $b_m(\mathbf{x})$ são as densidades componentes. Cada densidade componente é uma função gaussiana de dimensão D da forma

$$b_m(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} \mid \boldsymbol{\Sigma}_m \mid^{1/2}} e^{\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{x}-\boldsymbol{\mu}_m)\right\}} \tag{3}$$

onde |.| indica determinante e $(\ )^T$ representa transposição. A ponderação das misturas satisfaz à condição $\sum_{m=1}^{M} w_m = 1$.

O GMM pode apresentar três tipos de matriz de covariância:

- Distribuída a cada componente gaussiana;
- Uma única matriz para todas as componentes gaussianas de um dado modelo;
- Uma única matriz de covariância para todas as componentes de todos os modelos.

Além disto, a matriz de covariância também pode ser completa ou diagonal [11] .

O GMM representa de forma geral a dependência das características espectrais da voz associadas ao locutor, em conjunto com a capacidade de modelar densidades de probabilidades desconhecidas [1] [22] , especificamente a distribuição dos vetores de características extraídos de uma locução.

3.1 Estimação dos parâmetros de máxima verossimilhança

Em um sistema de identificação de locutor, cada locutor é representado por um GMM com seu modelo λ. Existem vários métodos de estimação dos parâmetros do GMM [1] . Um método bem difundido e que apresenta bons resultados é a estimação da máxima verossimilhança (*Maximum Likelihood* – ML).

Para um conjunto de dados de treinamento, a estimação ML tenta encontrar os parâmetros do modelo que maximizem a verossimilhança do GMM. Para uma seqüência de vetores de características (supondo independência entre os mesmos), $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, a verossimilhança do GMM é dada por

$$p(\mathbf{X} \mid \lambda) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \lambda) . \tag{4}$$

Normalizando pelo número total de vetores T e usando o logaritmo, chega-se a

$$\log p(\mathbf{X} \mid \lambda) = \frac{1}{T} \prod_{t=1}^{T} \log p(\mathbf{x}_t \mid \lambda) , \tag{5}$$

onde $\mathbf{x}_t$ é o *t*-ésimo vetor de $\mathbf{X}$.

Esta expressão é uma função não-linear de parâmetros λ e cuja maximização direta não é facilmente implementada [1] . Entretanto, a estimação dos parâmetros obtidos pelo método ML poderá ser conseguida iterativamente, utilizando-se um caso especial do algoritmo de máxima expectativa (*Expectation Maximization* – EM) [1] [22] .

A idéia básica do algoritmo EM é a de iniciarmos com um modelo inicial λ para a estimação de um novo modelo $\overline{\lambda}$, tal que $p(\mathbf{X} \mid \overline{\lambda}) \geq p(\mathbf{X} \mid \lambda)$. O novo modelo torna-se, então, o modelo inicial para a próxima iteração, e o processo é repetido até que um limiar de convergência seja alcançado. Esta é a mesma idéia básica para a estimação dos parâmetros do HMM através do algoritmo de reestimação de Baum-Welch [3] .

Em cada iteração do EM, as seguintes fórmulas de reestimação são usadas para a modelagem da m-ésima gaussiana, as quais garantem um crescimento monotônico do modelo de verossimilhança:

$$\overline{w}_m = \frac{1}{T}\sum_{t=1}^{T} P(m \mid \mathbf{x}_t, \lambda), \tag{6}$$

$$\overline{\mu}_m = \frac{\sum_{t=1}^{T} P(m \mid \mathbf{x}_t, \lambda)\mathbf{x}_t}{\sum_{t=1}^{T} P(m \mid \mathbf{x}_t, \lambda)} \tag{7}$$

e

$$\overline{\sigma}_{mj}^2 = \frac{\sum_{t=1}^{T} P(m \mid \mathbf{x}_t, \lambda)x_{tj}^2}{\sum_{t=1}^{T} P(m \mid \mathbf{x}_t, \lambda)} - \overline{\mu}_{mj}^2 \tag{8}$$

onde $\overline{\sigma}_{mj}^2$, x_{tj} e $\overline{\mu}_{mj}$, $j = 0,\ldots,D$, $m = 1,\ldots,M$ referem-se aos elementos dos vetores $\overline{\sigma}_m^2$, $\mathbf{x}_t$ e $\overline{\boldsymbol{\mu}}_m$ respectivamente e $P(m \mid \mathbf{x}_t, \lambda)$ é a probabilidade a posteriori para uma classe acústica m dada por

$$P(m \mid \mathbf{x}_t, \lambda) = \frac{w_m b_m(\mathbf{x}_t)}{\sum_{k=1}^{M} p_k b_k(\mathbf{x}_t)}. \tag{9}$$

A inicialização dos parâmetros a priori para o algoritmo EM foi realizada, neste trabalho, através do algoritmo k-médias [48] .

3.2 Identificação de locutor utilizando GMM

A Figura 3 apresenta o sistema de identificação de locutor utilizando GMM.

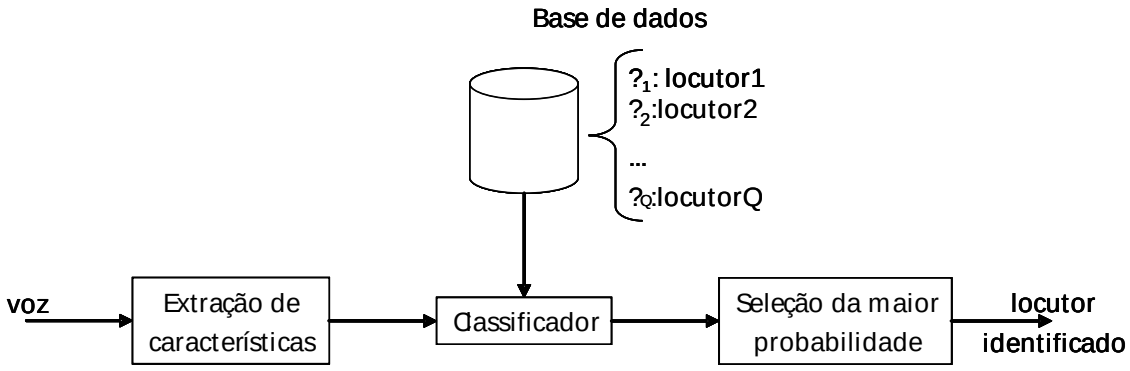

Figura 3 - Modelo do sistema de identificação de locutor

Dado um conjunto de Q locutores, a base de dados do sistema de identificação de locutor será composta pelos modelos λ_1, λ_2,..., λ_Q. O objetivo do sistema de identificação de locutor é encontrar qual modelo apresenta a maior probabilidade *a posteriori*, dada uma seqüência de vetores de características (extraídos do sinal de voz a ser identificado). Assim,

$$\hat{Q} = \arg\max_{1 \le n \le Q} P(\lambda_n \mid X) = \arg\max_{1 \le n \le Q} \frac{p(X \mid \lambda_n) P(\lambda_n)}{p(X)}. \qquad (10)$$

Assumindo que todos os modelos da base de dados têm probabilidade igual de ocorrerem, tem-se que $P(\lambda_n) = 1/Q$. Também se pode assumir que $p(X)$ é igual para todos os testes. Utilizando logaritmo e assumindo independência entre as observações, tem-se

$$\hat{Q} = \arg\max_{1 \le n \le Q} \sum_{t=1}^{T} \log p(x_t \mid \lambda_n). \qquad (11)$$

O modelo da base de dados que apresentar maior valor para $\hat{Q}$ é dito como o modelo identificado.

4 Ortogonalização de GMM

Normalmente, assume-se que a matriz de covariância $\mathbf{\Sigma}_m$ é diagonal. Segundo [29] , a matriz $\mathbf{\Sigma}_m$ pode ser diagonalizada se os vetores de características forem transformados para o espaço definido pelos seus autovetores. Em [29] é demonstrado que a matriz de covariância diagonalizada $\mathbf{\Sigma}_{Dm}$ e o correspondente vetor de médias $\boldsymbol{\mu}_{Dm}$ podem ser obtidos por

$$\mathbf{\Sigma}_{Dm} = \mathbf{\Omega}^T \mathbf{\Sigma}_m \mathbf{\Omega} \tag{12}$$

e

$$\boldsymbol{\mu}_{Dm} = \mathbf{\Omega}^T \boldsymbol{\mu}_m \tag{13}$$

onde $\mathbf{\Omega}$ é a matriz de transformação formada pelos autovetores da matriz de covariância $\mathbf{\Sigma}_m$.

A etapa de treinamento do GMM ortogonal é similar à etapa de treinamento do GMM convencional, exceto pelo fato de todos os vetores de características serem pré-multiplicados pela matriz de transformação $\mathbf{\Omega}^T$. Após a transformação, aplica-se o algoritmo k-médias para distribuir os vetores de características em M células que serão a base para o algoritmo EM.

Após a etapa de treinamento, a matriz de transformação é armazenada juntamente com o modelo treinado. Durante a etapa de teste, todos os vetores de características são pré-multiplicados pela matriz de transformação e em seguida é utilizado o mesmo procedimento de teste utilizado com o GMM convencional.

As Figuras 4 e 5 apresentam a saída do algoritmo k-médias utilizando vetores de características convencionais e ortogonalizados respectivamente. Foram utilizados vetores de duas dimensões para possibilitar a análise gráfica do resultado. Pode-se perceber que a ortogonalização concentra os parâmetros em células com menor dispersão.

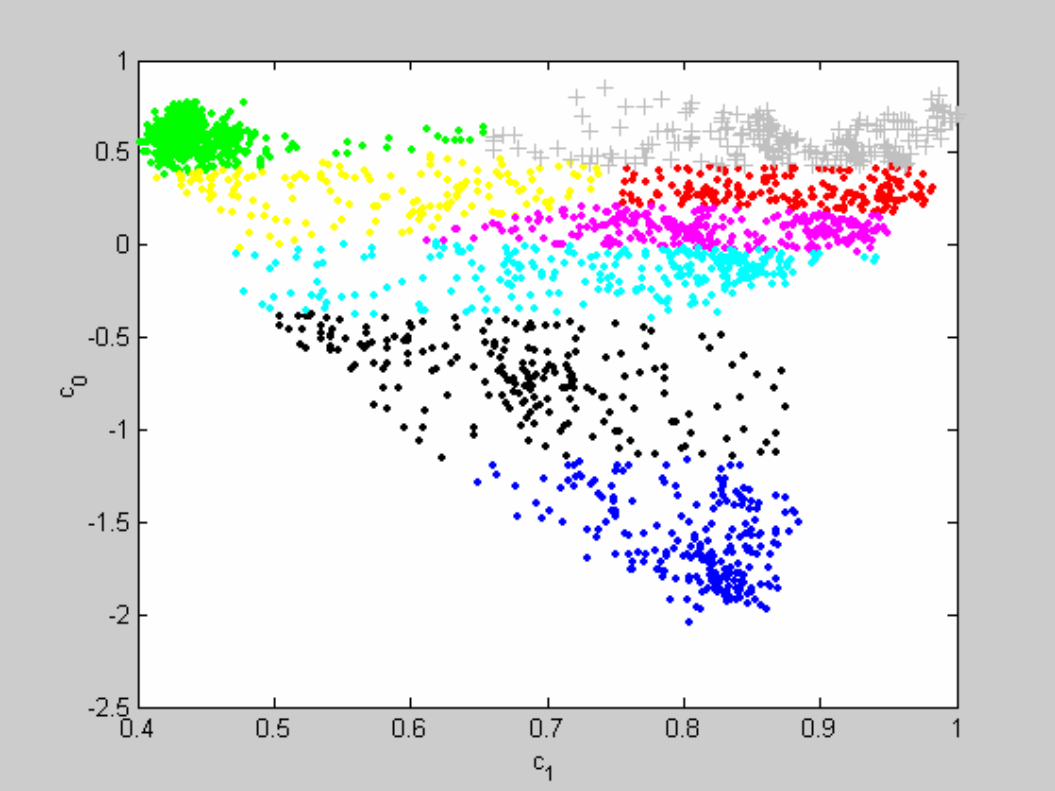

Figura 4- Resultado do algoritmo k-médias para parâmetros não ortogonalizados (8 misturas).

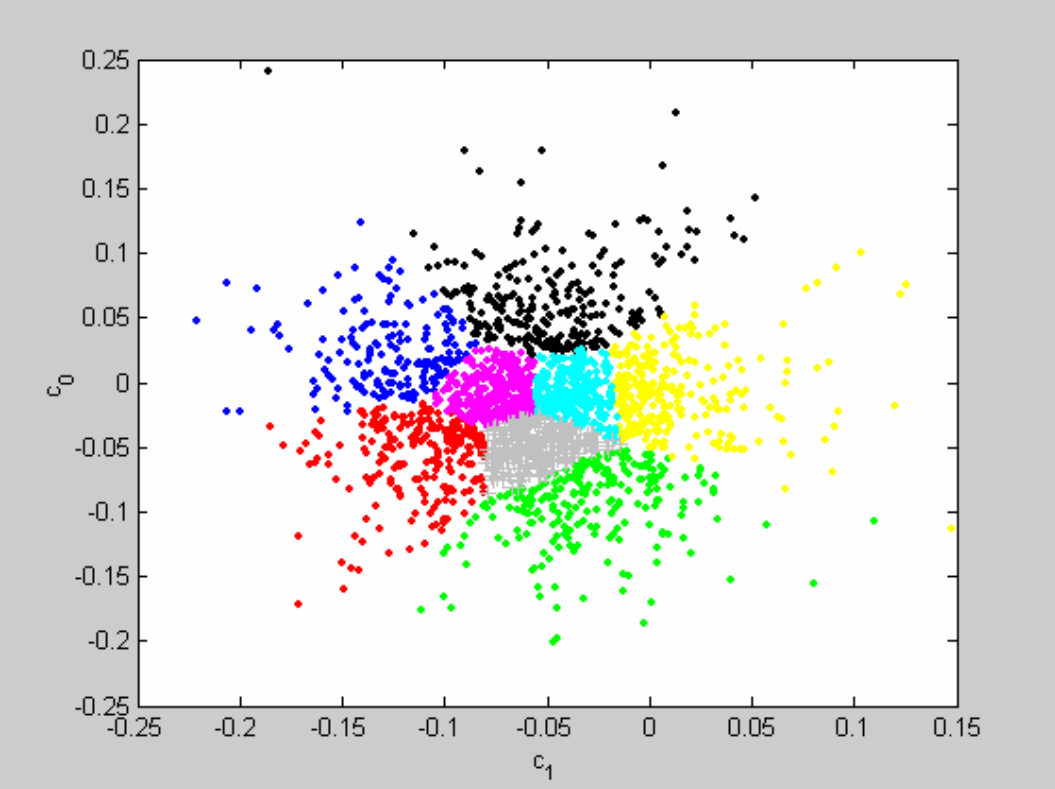

Figura 5 - Resultado do algoritmo k-médias para parâmetros ortogonalizados (8 misturas).

A Tabela 1 apresenta o esforço computacional necessário para a extração de parâmetros com e sem a utilização do algoritmo de ortogonalização. O esforço computacional foi calculado em termos do número de operações realizadas pelo algoritmo, normalizado pelo valor obtido para o tipo convencional de 25 parâmetros e multiplicado por 10 para tornar a visualização dos números mais fácil.

O algoritmo implementado está dividido em duas partes, uma pertencente ao *loop* de extração dos parâmetros e outra não pertencente a este *loop*. Para que o esforço computacional da parte não pertencente ao *loop* não tivesse um peso relevante no resultado final, os valores obtidos na tabela são referentes à extração de 100 vetores de parâmetros.

Tipo	2 parâmetros	10 parâmetros	15 parâmetros	25 parâmetros	50 parâmetros
Convencional	0,83	4,67	6,45	10,00	18,53
Ortogonal	0,89	4,88	6,95	10,83	19,85
Diferença	0,06	0,21	0,50	0,83	1,32

Tabela 1 - Esforço computacional para extração de parâmetros com e sem o uso da ortogonalização.

Pode-se observar que, conforme o esperado, ao aumentar o número de parâmetros extraídos, aumenta a diferença do esforço computacional utilizando o algoritmo de ortogonalização. Em termos proporcionais, para 25 parâmetros, o aumento no esforço computacional obtido representa cerca de 8,3% do esforço computacional original.

5 Algoritmo proposto

A distribuição dos vetores de características em um espaço de ordem D não é a mesma para todos os locutores. Em alguns casos, os vetores estão demasiadamente espalhados, necessitando de mais células para cobrir todo o espaço com a mesma precisão utilizada em locutores que tenham vetores menos espalhados. O uso de poucas componentes pode ocasionar que vetores com características diferentes fiquem na mesma célula, resultando em componentes gaussianas com alta variância.

Por outro lado, alguns locutores possuem vetores de características concentrados em alguns pontos, necessitando apenas de algumas células para cobrir todo o espaço. Se este tipo de locutor for modelado com muitas componentes, estas se tornarão similares e não trarão benefícios durante o processo de reconhecimento, apenas aumento do esforço computacional.

A fim de resolver este problema, a seguir é apresentado um algoritmo que utiliza a razão entre as distorções máxima e média em cada célula para encontrar o número ótimo de componentes para cada locutor.

A idéia básica deste algoritmo é que células com esta razão elevada possuem muitos vetores afastados do centróide, o que irá resultar em alta variância no modelo, e conseqüentemente, um locutor mal modelado. Por outro lado, ter esta razão igual a 1 significa que temos uma componente gaussiana para cada vetor de características, resultando em um modelo sem variância.

A fim de se evitar este problema, é definido para cada locutor uma faixa de possíveis números de componentes entre $4 \le j \le 24$. Então, para um possível número de componentes, é executado o algoritmo k-médias na base de dados de treinamento, e as distorções máxima e média para cada célula j é calculada através de

$$Max.Dist.(j) = \max\{d_{\max}(m)\}, m = 4,\ldots, j \qquad (14)$$

onde

$$d_{\max}(m) = \max\left\{\sqrt{\sum_{d=1}^{D}(x(m,l,d) - c_m(d))^2}\right\},\ l = 1,\ldots,L_m \tag{15}$$

e

$$Avg.Dist.(j) = \max\{d_{avg}(m)\},\ m = 4,\ldots,j \tag{16}$$

onde

$$d_{avg}(m) = \frac{1}{L_m}\sum_{l=1}^{L_m}\sqrt{\sum_{d=1}^{D}(x(m,l,d) - c_m(d))^2} \tag{17}$$

onde M é o número de componentes, L_m é o número de vetores na m-ésima célula, $x(m,l,d)$ é o d-ésimo parâmetro do l-ésimo vetor na m-ésima célula.

Estes dois valores são utilizados para calcular o fator $R(j)$, dado por

$$R(j) = \frac{Max.Dist.(j)}{Avg.Dist.(j)},\ j = 4,\ldots,24. \tag{18}$$

Os valores de $R(j)$ são comparados a um limiar pré-definido. O número de componentes que apresentar o valor $R(j)$ mais próximo do limiar *R*, definido durante a etapa de treinamento, é escolhido como o número de componentes para modelar aquele locutor. Este processo é repetido para cada locutor até que toda a base tenha sido treinada.

É importante notar que $R(j)$ nunca pode ser menor que 1, uma vez que a distorção máxima não pode ser menor que a distorção média.

A Tabela 2 apresenta os parâmetros utilizados no algoritmo de otimização, assim como os resultados parciais, obtidos através da Equação (18), para cada número de componentes. As etapas do algoritmo de otimização podem ser vistas entre a Figura 6 e a Figura 14. Cada célula esta representada por uma cor, e o quadrado dentro de cada célula é a posição de seu centróide.

Pode-se observar que o aumento do número de componentes reduz significativamente a distorção média. Isto ocorre, pois as células tornam-se menores, fazendo com que, na média, os vetores fiquem mais próximos dos centróides. A distorção máxima não sofre grande alteração ao aumentarmos o número de componentes, pois os vetores mais distantes permanecem distantes dos centróides.

Tabela 2 – Resultados parciais do algoritmo de otimização para um vetor de características com duas dimensões.

Componentes (*j*)	**Distorção Máxima**	**Distorção Média**	$R(j)$
12	0.035744	0.023623	1.513057
13	0.033238	0.024907	1.334484
14	0.031341	0.026712	1.155641
15	0.034887	0.024020	1.452414
16	0.035221	0.021034	1.674451
17	0.034792	0.020282	1.715399
18	0.035341	0.019405	1.821289
19	0.033875	0.018832	1.798831
20	0.035939	0.018989	1.892644

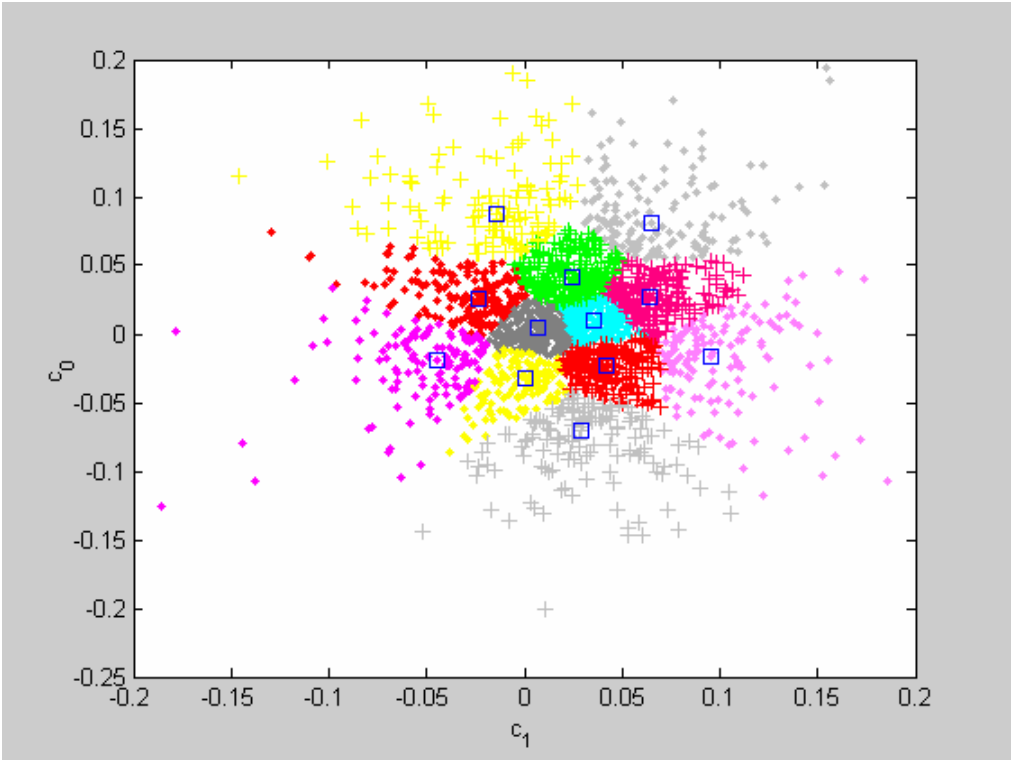

Figura 6 - Distribuição dos vetores de características utilizando 12 células.

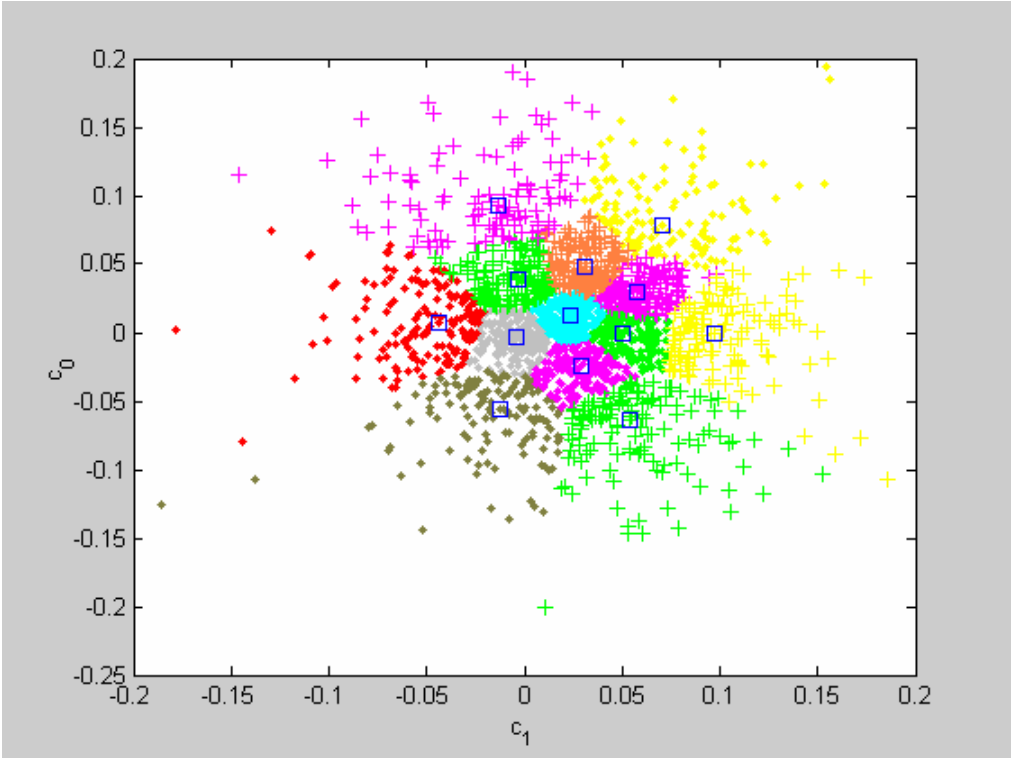

Figura 7 - Distribuição dos vetores de características utilizando 13 células.

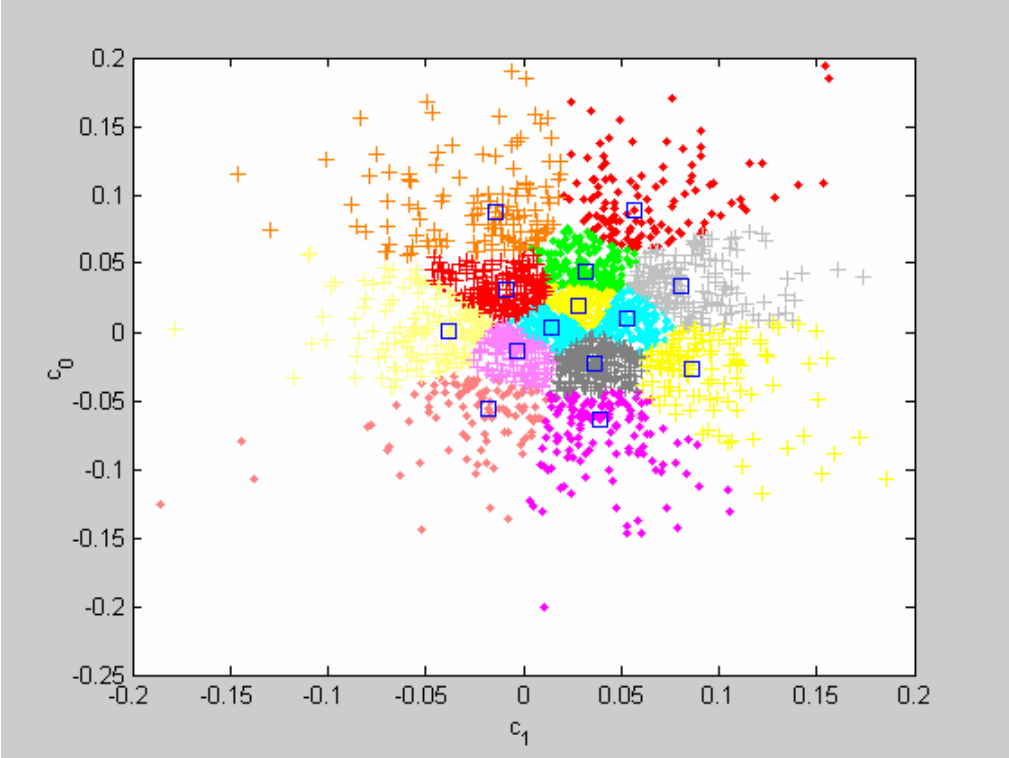

Figura 8 - Distribuição dos vetores de características utilizando 14 células.

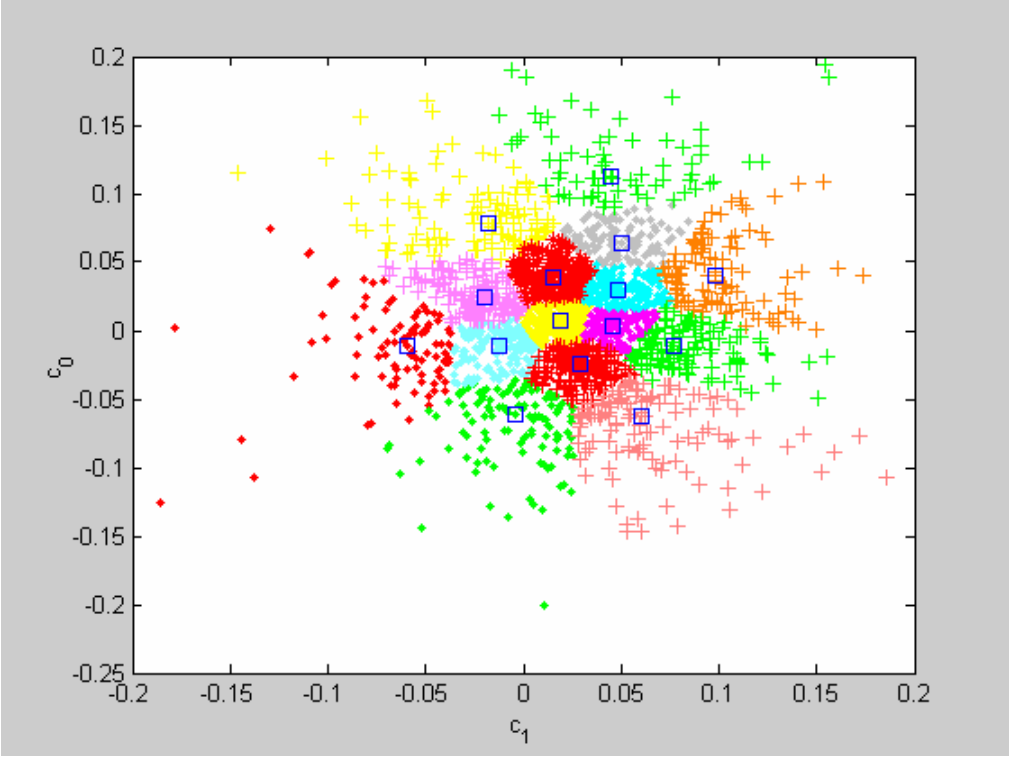

Figura 9 - Distribuição dos vetores de características utilizando 15 células.

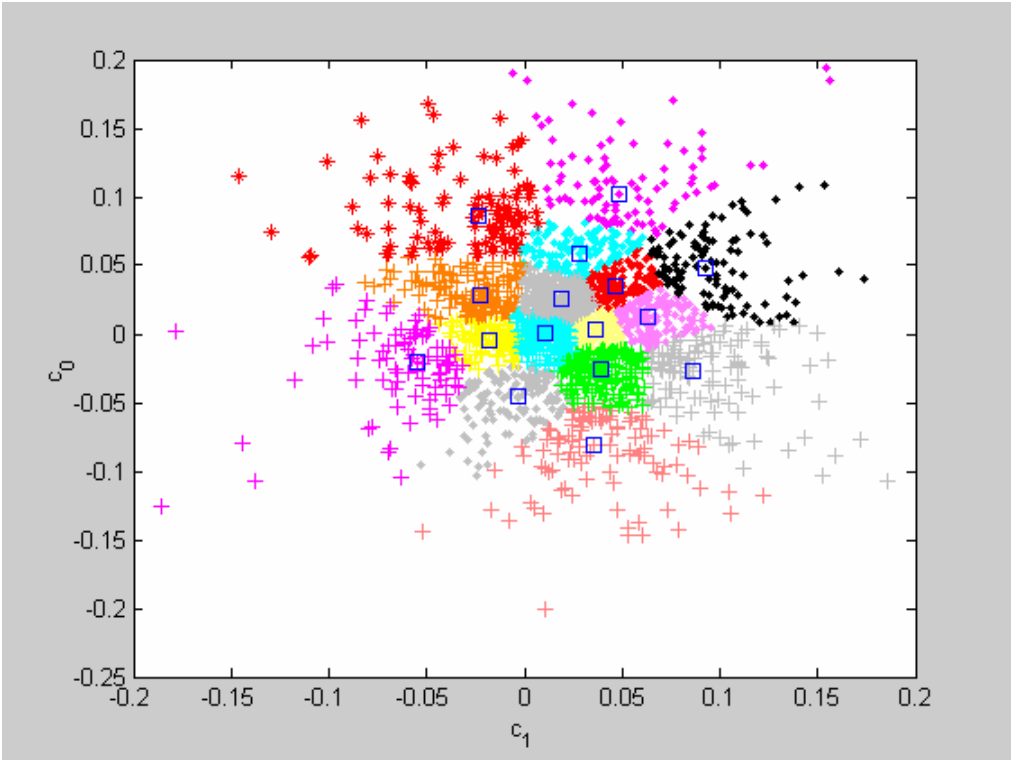

Figura 10 - Distribuição dos vetores de características utilizando 16 células.

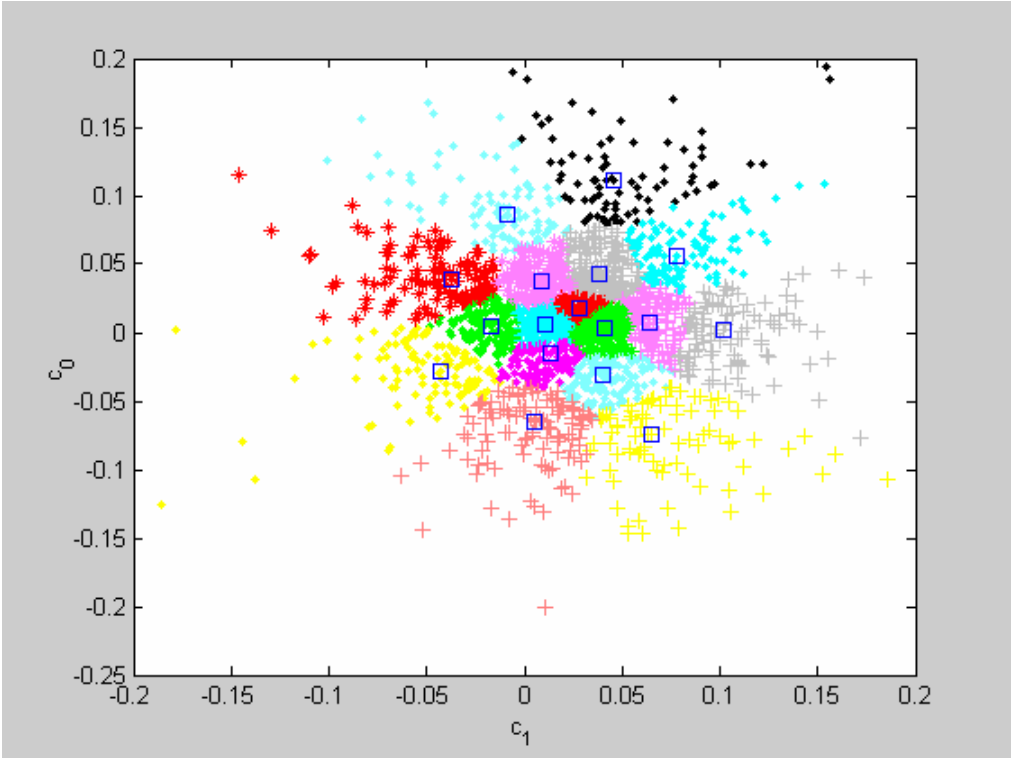

Figura 11 - Distribuição dos vetores de características utilizando 17 células.

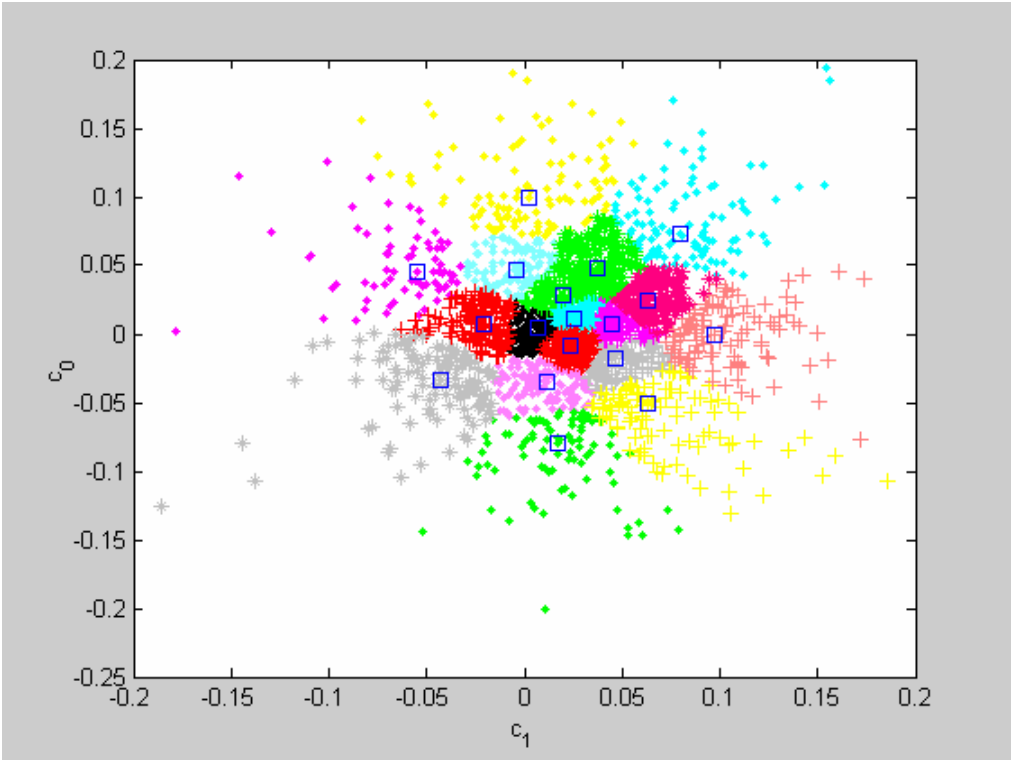

Figura 12 - Distribuição dos vetores de características utilizando 18 células.

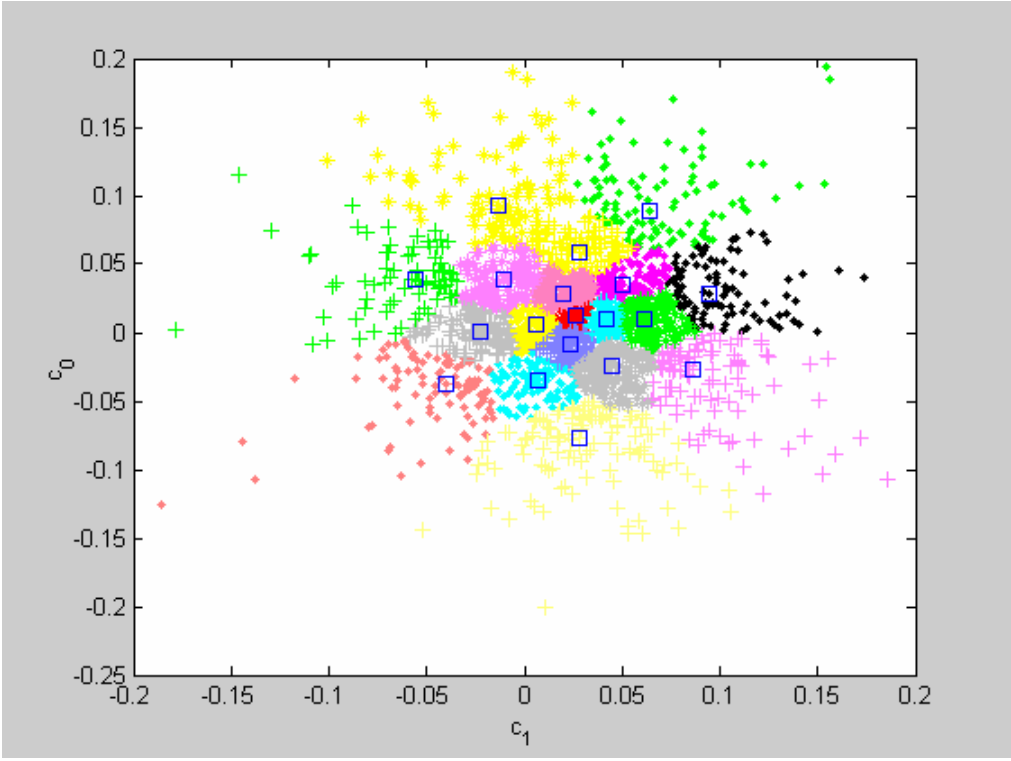

Figura 13 - Distribuição dos vetores de características utilizando 19 células.

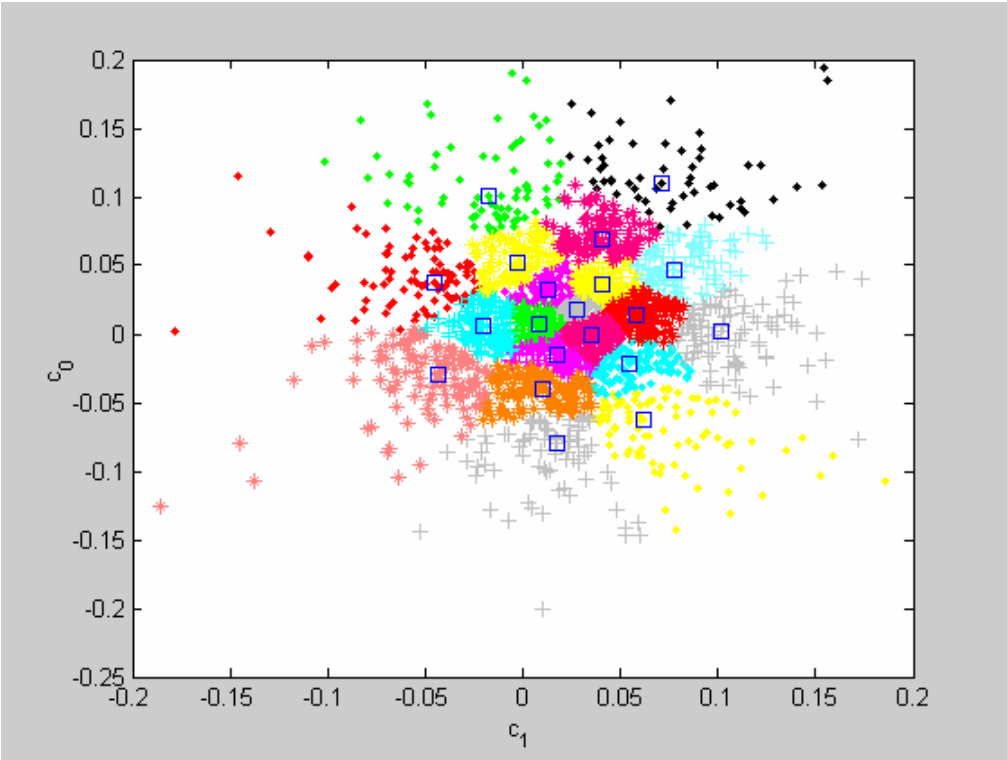

Figura 14 - Distribuição dos vetores de características utilizando 20 células.

6 Base de Dados

A base de dados utilizada nos experimentos é composta de 342 locutores [37] . Cada locutor gravou ao menos 25 segundos de voz, amostrada a 16 kHz. Ruído branco com RSR (razão sinal-ruído) de 25 dB foi adicionado a cada locução. Esta base de dados foi dividida em dois grupos: grupo de treinamento e grupo de teste. No grupo de treinamento, cada locutor possui dois segmentos de 10 segundos que serão utilizados em testes cruzados, enquanto no grupo de teste cada locutor possui cinco segmentos de um segundo cada. Os segmentos de treinamento e de teste são combinados em pares, formando 10 ambientes de treinamento/teste. Os resultados obtidos nos testes representam a média destes 10 ambientes.

O UBM (*universal background model*) utilizado nos testes é composto por 10 locutores (cinco masculinos e cinco femininos) não presentes na etapa de treinamento. Cada locução possui 10 segundos, totalizando 100 segundos para a modelagem do UBM.

Outros 13 locutores foram removidos do *corpus* de treinamento para serem utilizados nos testes como usuários externos à base de dados. Para cada um destes locutores foram utilizadas cinco locuções de um segundo nos testes.

Os locutores foram modelados utilizando-se 25 coeficientes mel-cepstrais generalizados, definidos no Apêndice 1, e seus respectivos delta e delta-delta, extraídos de blocos de 400 amostras (20 ms), espaçados de 80 amostras (4 ms).

7 Testes

Neste trabalho foram realizados os seguintes testes a fim de validar o algoritmo proposto:

- Teste de ortogonalização;
- Teste de otimização;
- Testes comparativos com outros sistemas.

Cada um destes testes é subdividido em seqüências de testes detalhadas nas próximas subseções. Nos testes realizados existem quatro resultados possíveis para o reconhecimento de cada locutor:

- Falsa Identificação: ocorre quando um usuário que não pertence à base de dados é reconhecido como um usuário da base de dados ou quando um usuário que pertence à base de dados é reconhecido como outro usuário que pertence à base de dados;
- Falsa Negação: ocorre quando um usuário que faz parte da base de dados é reconhecido como UBM, ou seja, como o usuário de escape;
- Correta Identificação: ocorre quando um usuário que pertence à base de dados é corretamente reconhecido;
- Correta Negação: ocorre quando um usuário que não pertence à base de dados é reconhecido como o usuário de escape.

Com base nestes quatro resultados possíveis, as taxas de erro e acerto são calculadas da seguinte maneira:

$$\%\,\text{Erro} = \frac{(\text{total de falsas identificações}) + (\text{total de falsas negações})}{\text{total de locuções testadas}} \tag{19}$$

e

$$\%\,\text{Acerto} = \frac{(\text{total de corretas identificações}) + (\text{total de corretas negações})}{\text{total de locuções testadas}}. \tag{20}$$

7.1 Teste de ortogonalização

Neste primeiro teste, são comparadas as taxas de acerto entre o GMM convencional e o GMM ortogonal a fim de se verificar se a ortogonalização apresenta ganhos ao sistema. Para cada experimento foram utilizados números fixos de componentes por usuário, ou seja, não está sendo utilizado o algoritmo de otimização proposto.

Foram realizados dois tipos de teste: o primeiro comparando as taxas de acerto do GMM convencional com as taxas de acerto do GMM ortogonal, e o segundo teste comparando as taxas de acerto do GMM ortogonal com e sem UBM.

7.1.1 Teste sem UBM

Neste teste não foi utilizado o UBM a fim de se verificar o impacto da ortogonalização sobre o reconhecimento apenas de elementos da base de dados. Com isso, não foram utilizados para os testes os 13 locutores extraídos da base de dados. Desta forma, os resultados Falsa Negação e Correta Negação são iguais à zero. Os resultados podem ser vistos na Tabela 3 e na Figura 15.

Tabela 3 - Taxa de acerto para GMM convencional, sem o uso do UBM

Número de misturas	GMM convencional	GMM ortogonal
4	64.0379%	93.0599%
8	84.2271%	96.5300%
16	89.5899%	97.1924%
32	90.8517%	96.7918%
64	93.0599%	96.5800%
128	90.8517%	96.5100%

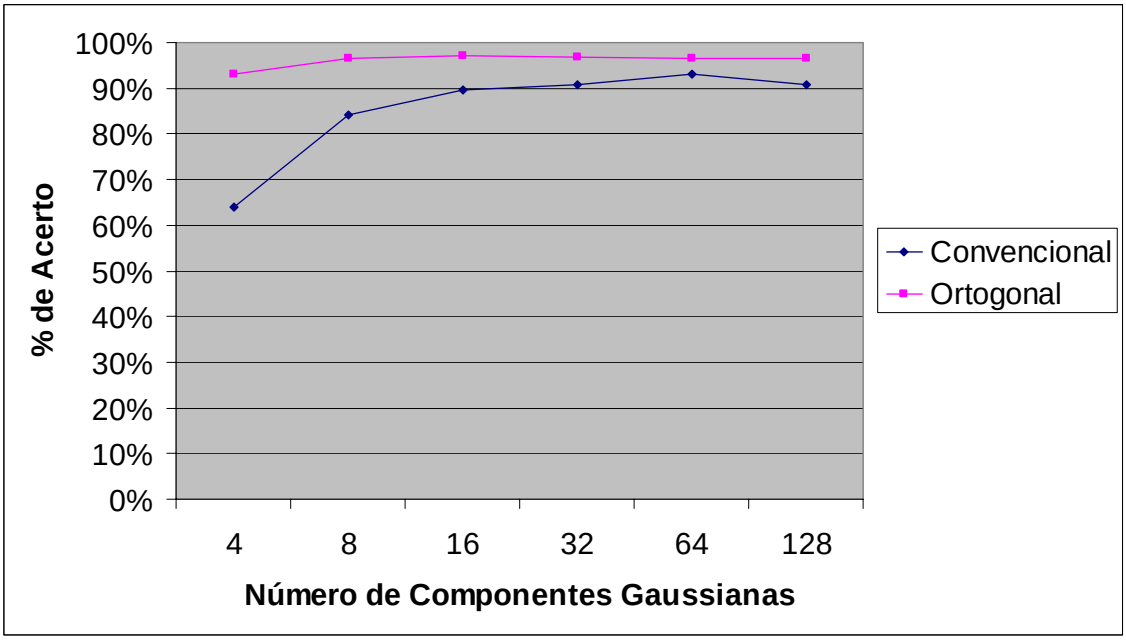

Figura 15 - Comparação das taxas de acerto entre GMM convencional e GMM ortogonal, sem o uso do UBM.

Os resultados obtidos demonstram que, para a base de dados utilizada, a introdução do algoritmo de ortogonalização trouxe um aumento significativo no percentual de acerto para a identificação de locutor sem o uso do UBM. Os melhores resultados foram 93,06% utilizando-se GMM convencional com 64 componentes e 97,19% utilizando-se GMM ortogonal com 16 componentes.

Além deste aumento significativo no percentual de acerto (cerca de 4.1%), pode-se perceber também que houve uma diminuição no número de misturas gaussianas do melhor resultado: 64 utilizando-se GMM convencional e 16 utilizando-se GMM ortogonal. Esse fato resulta em uma diminuição do esforço computacional, que será visto na Seção 7.2.

7.1.2 Teste com UBM

Neste teste foi utilizado o UBM a fim de verificar se a introdução deste causa perdas ao sistema. Foram utilizados diversos modelos de UBM, treinados com números fixos de

misturas gaussianas, variando de 64 até 512. A simbologia UBM-X significa UBM modelado com X componentes.

Tabela 4 – Comparação das taxas de acerto entre o GMM ortogonal com e sem UBM.

Número de misturas	Sem UBM	UBM-64	UBM-128	UBM-256	UBM-512
4	93.0599%	91.7981%	91.7971%	91.7511%	91.4826%
8	96.5300%	95.2681%	94.9527%	95.2981%	95.2281%
16	97.1924%	96.8991%	96.8991%	97.1045%	97.0223%
32	96.7918%	95.8991%	95.5836%	96.5300%	96.0114%
64	96.5800%	94.6372%	94.6323%	94.9527%	94.2145%
128	96.5100%	89.2744%	89.5899%	89.9854%	89.9211%

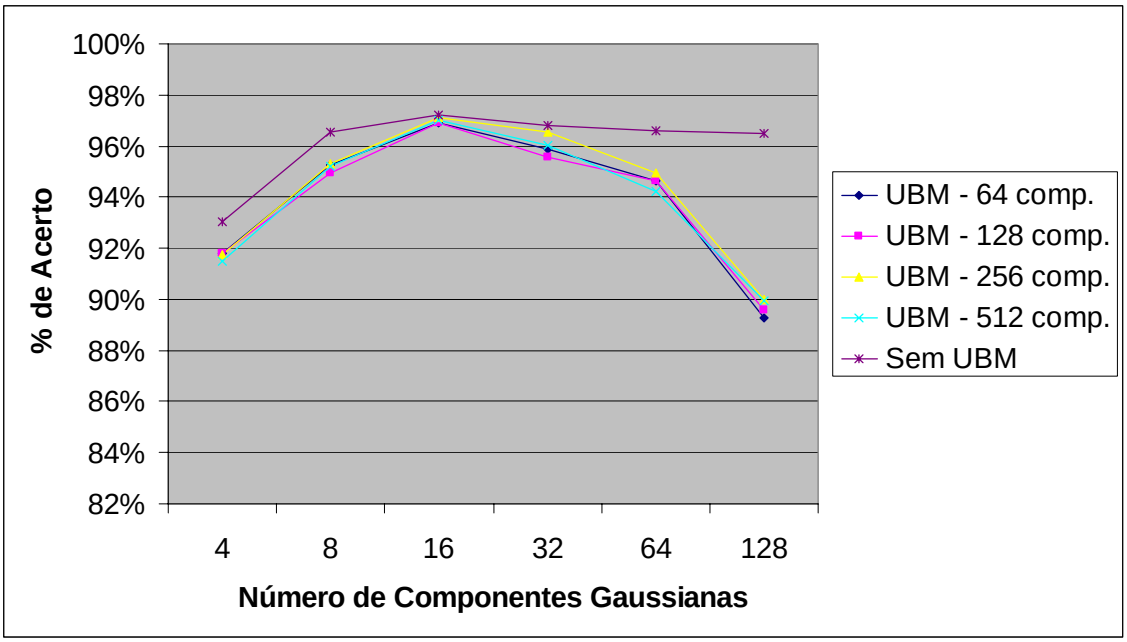

Figura 16 - Comparação das taxas de acerto entre GMM ortogonal com e sem o uso do UBM.

Os resultados deste teste mostram que a introdução do UBM causa uma redução na taxa de acerto do sistema, conforme esperado, uma vez que existe um modelo a mais na lista de possíveis identificações.

Pode-se observar que os melhores resultados utilizando-se UBM ocorreram para o GMM ortogonal com 16 misturas, mantendo-se aproximadamente o resultado obtido no teste anterior. Para este número de misturas, o melhor resultado foi obtido utilizando-se um UBM modelado com 256 misturas. Para este caso, a diminuição na taxa de acerto devido à inserção do UBM foi de aproximadamente 0,1%.

Também é possível observar que para todos os possíveis números de misturas gaussianas, a exceção do GMM ortogonal de 4 misturas, o UBM que apresentou melhores resultados foi o treinado com 256 misturas. A exceção ocorrida com o modelo treinado com 4 misturas pode ser explicada pelo fato do UBM ter sido significativamente melhor modelado que os usuários da base de dados, mesmo para a situação com o menor número de misturas (64).

7.2 Testes de otimização

Neste segundo conjunto de testes é avaliado o desempenho do algoritmo proposto sob dois aspectos: taxa de erro e esforço computacional. A taxa de erro foi calculada conforme mostrado na Equação (19). O esforço computacional foi calculado em termos do número de operações realizado pelo algoritmo. O resultado obtido foi normalizado pelo valor obtido no teste utilizando GMM ortogonal, número fixo de gaussianas (16) e sem a presença do UBM e em seguida multiplicado por 10 apenas para facilitar a visualização dos gráficos.

Foram criadas 96 Configurações de teste divididas em três conjuntos:

- Configurações com GMM convencional, sem UBM: base de dados treinada utilizando-se GMM convencional. Testes realizados sem a presença do UBM (Tabela 5).
- Configurações com GMM ortogonal, sem UBM: base de dados treinada utilizando-se GMM ortogonal. Testes realizados sem a presença do UBM (Tabela 6).
- Configurações com GMM ortogonal, com UBM: base de dados treinada utilizando-se GMM ortogonal. Testes realizados com a presença do UBM (Tabela 7).

As configurações C13 até C24 e C49 até C96 apresentam diversos valores na coluna **Número de misturas** das Tabelas 4, 5 e 6. Isto significa que cada locutor da base de dados pode utilizar uma destas quantidades de misturas gaussianas. O parâmetro R é o valor ao qual o resultado *R(j)* da Equação (18) é comparado.

Tabela 5 - Configurações utilizando GMM convencional, sem UBM. O termo N.A. significa "Não Aplicável".

Configuração	GMM	Número de misturas	Fator R	UBM
C1	Conv.	4	N.A.	N.A.
C2	Conv.	8	N.A.	N.A.
C3	Conv.	16	N.A.	N.A.
C4	Conv.	32	N.A.	N.A.
C5	Conv.	64	N.A.	N.A.
C6	Conv.	128	N.A.	N.A.

Tabela 6 - Configurações utilizando GMM ortogonal, sem UBM. O termo N.A. significa "Não Aplicável".

Configuração	GMM	Número de misturas	Fator R	UBM
C7	Orto.	4	N.A.	N.A.
C8	Orto.	8	N.A.	N.A.
C9	Orto.	16	N.A.	N.A.
C10	Orto.	32	N.A.	N.A.
C11	Orto.	64	N.A.	N.A.
C12	Orto.	128	N.A.	N.A.
C13	Orto.	4, 8, 16, 32, 64, 128	1	N.A.
C14	Orto.	4, 8, 16, 32, 64, 128	1.1	N.A.
C15	Orto.	4, 8, 16, 32, 64, 128	1.2	N.A.
C16	Orto.	4, 8, 16, 32, 64, 128	1.3	N.A.
C17	Orto.	4, 8, 16, 32, 64, 128	1.4	N.A.
C18	Orto.	4, 8, 16, 32, 64, 128	1.5	N.A.

C19	Orto.	12,13,14,15,16,17,18,19,20	1	N.A.
C20	Orto.	12,13,14,15,16,17,18,19,20	1.1	N.A.
C21	Orto.	12,13,14,15,16,17,18,19,20	1.2	N.A.
C22	Orto.	12,13,14,15,16,17,18,19,20	1.3	N.A.
C23	Orto.	12,13,14,15,16,17,18,19,20	1.4	N.A.
C24	Orto.	12,13,14,15,16,17,18,19,20	1.5	N.A.

Tabela 7 - Configurações utilizando GMM ortogonal, com UBM. O termo N.A. significa "Não Aplicável".

Configuração	GMM	Número de misturas	Fator R	UBM
C25	Orto.	4	N.A.	64
C26	Orto.	4	N.A.	128
C27	Orto.	4	N.A.	256
C28	Orto.	4	N.A.	512
C29	Orto.	8	N.A.	64
C30	Orto.	8	N.A.	128
C31	Orto.	8	N.A.	256
C32	Orto.	8	N.A.	512
C33	Orto.	16	N.A.	64
C34	Orto.	16	N.A.	128
C35	Orto.	16	N.A.	256
C36	Orto.	16	N.A.	512
C37	Orto.	32	N.A.	64
C38	Orto.	32	N.A.	128
C39	Orto.	32	N.A.	256
C40	Orto.	32	N.A.	512
C41	Orto.	64	N.A.	64
C42	Orto.	64	N.A.	128
C43	Orto.	64	N.A.	256
C44	Orto.	64	N.A.	512

C45	Orto.	128	N.A.	64
C46	Orto.	128	N.A.	128
C47	Orto.	128	N.A.	256
C48	Orto.	128	N.A.	512
C49	Orto.	4, 8, 16, 32, 64, 128	1	64
C50	Orto.	4, 8, 16, 32, 64, 128	1	128
C51	Orto.	4, 8, 16, 32, 64, 128	1	256
C52	Orto.	4, 8, 16, 32, 64, 128	1	512
C53	Orto.	4, 8, 16, 32, 64, 128	1.1	64
C54	Orto.	4, 8, 16, 32, 64, 128	1.1	128
C55	Orto.	4, 8, 16, 32, 64, 128	1.1	256
C56	Orto.	4, 8, 16, 32, 64, 128	1.1	512
C57	Orto.	4, 8, 16, 32, 64, 128	1.2	64
C58	Orto.	4, 8, 16, 32, 64, 128	1.2	128
C59	Orto.	4, 8, 16, 32, 64, 128	1.2	256
C60	Orto.	4, 8, 16, 32, 64, 128	1.2	512
C61	Orto.	4, 8, 16, 32, 64, 128	1.3	64
C62	Orto.	4, 8, 16, 32, 64, 128	1.3	128
C63	Orto.	4, 8, 16, 32, 64, 128	1.3	256
C64	Orto.	4, 8, 16, 32, 64, 128	1.3	512
C65	Orto.	4, 8, 16, 32, 64, 128	1.4	64
C66	Orto.	4, 8, 16, 32, 64, 128	1.4	128
C67	Orto.	4, 8, 16, 32, 64, 128	1.4	256
C68	Orto.	4, 8, 16, 32, 64, 128	1.4	512
C69	Orto.	4, 8, 16, 32, 64, 128	1.5	64
C70	Orto.	4, 8, 16, 32, 64, 128	1.5	128
C71	Orto.	4, 8, 16, 32, 64, 128	1.5	256
C72	Orto.	4, 8, 16, 32, 64, 128	1.5	512
C73	Orto.	12,13,14,15,16,17,18,19,20	1	64

C74	Orto.	12,13,14,15,16,17,18,19,20	1	128
C75	Orto.	12,13,14,15,16,17,18,19,20	1	256
C76	Orto.	12,13,14,15,16,17,18,19,20	1	512
C77	Orto.	12,13,14,15,16,17,18,19,20	1.1	64
C78	Orto.	12,13,14,15,16,17,18,19,20	1.1	128
C79	Orto.	12,13,14,15,16,17,18,19,20	1.1	256
C80	Orto.	12,13,14,15,16,17,18,19,20	1.1	512
C81	Orto.	12,13,14,15,16,17,18,19,20	1.2	64
C82	Orto.	12,13,14,15,16,17,18,19,20	1.2	128
C83	Orto.	12,13,14,15,16,17,18,19,20	1.2	256
C84	Orto.	12,13,14,15,16,17,18,19,20	1.2	512
C85	Orto.	12,13,14,15,16,17,18,19,20	1.3	64
C86	Orto.	12,13,14,15,16,17,18,19,20	1.3	128
C87	Orto.	12,13,14,15,16,17,18,19,20	1.3	256
C88	Orto.	12,13,14,15,16,17,18,19,20	1.3	512
C89	Orto.	12,13,14,15,16,17,18,19,20	1.4	64
C90	Orto.	12,13,14,15,16,17,18,19,20	1.4	128
C91	Orto.	12,13,14,15,16,17,18,19,20	1.4	256
C92	Orto.	12,13,14,15,16,17,18,19,20	1.4	512
C93	Orto.	12,13,14,15,16,17,18,19,20	1.5	64
C94	Orto.	12,13,14,15,16,17,18,19,20	1.5	128
C95	Orto.	12,13,14,15,16,17,18,19,20	1.5	256
C96	Orto.	12,13,14,15,16,17,18,19,20	1.5	512

As configurações utilizando o algoritmo de otimização apresentam dois conjuntos de valores possíveis de misturas para cada locutor: [4,8,16,32,64,128] e [12,13,14,15,16,17,18,19,20]. O segundo conjunto foi criado por ter sido verificado nos testes da Seção 7.1 que o GMM ortogonal apresenta melhores resultados em torno de 16 misturas.

É importante notar que utilizar o fator R igual a 1, na verdade, significa permitir que o algoritmo automaticamente selecione o menor valor de $R(j)$ na Equação (18).

Assim como nos testes realizados na Seção 7.1.1, os testes utilizando as configurações sem UBM (C1 até C24) não utilizaram os 13 locutores ausentes na base de dados e apresentam os resultados Falsa Negação e Correta Negação iguais a zero.

As tabelas a seguir apresentam a taxa de erro de falsa negação, taxa de erro de falsa aceitação, taxa de erro total e o esforço computacional obtidos para cada uma das configurações testadas.

Tabela 8 - Resultados para reconhecimento utilizando GMM convencional, sem UBM.

Configuração	Taxa de erro: falsa negação	Taxa de erro: falsa identificação	Taxa de erro total	Esforço computacional
C1	N.A.	35.9621%	35.9621%	1.36
C2	N.A.	15.7729%	15.7729%	4.17
C3	N.A.	10.4101%	10.4101%	7.92
C4	N.A.	9.1483%	9.1483%	15.92
C5	N.A.	6.9401%	6.9401%	31.75
C6	N.A.	9.1483%	9.1483%	62.32

Tabela 9 - Resultados para reconhecimento utilizando GMM ortogonal, sem UBM.

Configuração	Taxa de erro: falsa negação	Taxa de erro: falsa identificação	Taxa de erro total	Esforço computacional
C7	N.A.	6.9401%	6.9401%	2.13
C8	N.A.	3.4700%	3.4700%	5.09
C9	N.A.	2.8076%	2.8076%	10.00
C10	N.A.	3.2082%	3.2082%	18.38
C11	N.A.	3.4200%	3.4200%	35.75

C12	N.A.	3.4900%	3.4900%	66.31
C13	N.A.	2.7678%	2.7678%	3.29
C14	N.A.	4.8978%	4.8978%	4.06
C15	N.A.	5.1354%	5.1354%	5.06
C16	N.A.	7.0710%	7.0710%	6.74
C17	N.A.	6.3225%	6.3225%	8.35
C18	N.A.	8.1354%	8.1354%	9.70
C19	N.A.	0.9889%	0.9889%	5.06
C20	N.A.	3.0968%	3.0968%	5.19
C21	N.A.	3.7888%	3.7888%	5.22
C22	N.A.	3.2335%	3.2335%	5.45
C23	N.A.	3.1134%	3.1134%	5.19
C24	N.A.	3.7757%	3.7757%	5.26

Tabela 10 - Resultados para reconhecimento utilizando GMM ortogonal, com UBM.

Configuração	**Taxa de erro: falsa negação**	**Taxa de erro: falsa identificação**	**Taxa de erro total**	**Esforço computacional**
C25	4.0210%	4.1809%	8.2019%	2.08
C26	3.9633%	4.2396%	8.2029%	2.22
C27	3.9427%	4.3062%	8.2489%	2.24
C28	4.1760%	4.3414%	8.5174%	2.52
C29	2.0105%	2.7213%	4.7319%	5.03
C30	2.4926%	2.5547%	5.0473%	4.98
C31	2.0294%	2.6725%	4.7019%	5.18
C32	2.3457%	2.4261%	4.7719%	5.47
C33	1.3659%	1.7351%	3.1009%	10.14
C34	1.2225%	1.8784%	3.1009%	9.98
C35	1.0804%	1.8151%	2.8955%	10.00
C36	1.4270%	1.5507%	2.9777%	10.02

C37	1.8229%	2.2781%	4.1009%	18.24
C38	2.0360%	2.3804%	4.4164%	18.25
C39	1.4680%	2.0021%	3.4700%	18.35
C40	1.9331%	2.0555%	3.9886%	18.14
C41	2.4885%	2.8743%	5.3628%	35.83
C42	2.3756%	2.9921%	5.3677%	35.79
C43	2.0375%	3.0098%	5.0473%	35.84
C44	2.5404%	3.2451%	5.7855%	35.61
C45	4.9800%	5.7455%	10.7256%	66.49
C46	4.7461%	5.6640%	10.4101%	66.43
C47	4.5620%	5.4526%	10.0146%	66.52
C48	4.9700%	5.1089%	10.0789%	66.21
C49	2.1070%	2.3094%	4.4164%	3.21
C50	2.0323%	2.3841%	4.4164%	3.02
C51	1.6686%	2.4324%	4.1009%	3.11
C52	1.5073%	2.2781%	3.7855%	3.21
C53	2.3251%	2.7223%	5.0473%	4.13
C54	2.0280%	3.0194%	5.0473%	4.27
C55	2.1687%	2.8787%	5.0473%	4.27
C56	2.4923%	2.7641%	5.2565%	4.06
C57	2.3698%	3.2759%	5.6456%	5.05
C58	2.6929%	2.7426%	5.4355%	5.05
C59	2.4437%	2.6917%	5.1354%	5.12
C60	2.2536%	2.9921%	5.2456%	5.12
C61	3.4826%	4.0884%	7.5710%	6.73
C62	3.3773%	4.1937%	7.5710%	6.81
C63	3.3248%	4.2462%	7.5710%	6.84
C64	3.8430%	4.0434%	7.8864%	6.62
C65	3.3948%	3.5453%	6.9401%	8.38

C66	3.4220%	3.8335%	7.2555%	8.32
C67	3.2414%	4.0141%	7.2555%	8.31
C68	3.4352%	3.5049%	6.9401%	8.31
C69	4.4280%	5.2186%	9.6465%	9.77
C70	3.8825%	4.7931%	8.6756%	9.67
C71	3.5822%	4.5532%	8.1354%	9.76
C72	4.3436%	4.5333%	8.8769%	9.80
C73	1.3774%	1.4993%	2.8768%	4.98
C74	0.8637%	1.5819%	2.4457%	5.02
C75	0.2395%	0.8829%	1.1224%	5.12
C76	1.7892%	1.8866%	3.6758%	5.09
C77	1.9069%	2.1940%	4.1009%	5.01
C78	1.8090%	2.2920%	4.1009%	5.20
C79	1.3020%	2.1680%	3.4700%	5.09
C80	1.6421%	2.0147%	3.6568%	5.01
C81	1.7781%	2.3228%	4.1009%	5.29
C82	2.2381%	2.4937%	4.7319%	5.46
C83	1.8995%	2.5169%	4.4164%	5.42
C84	2.0470%	2.3694%	4.4164%	5.48
C85	1.8504%	2.2506%	4.1009%	5.30
C86	1.5973%	2.5036%	4.1009%	5.58
C87	1.8499%	1.9356%	3.7855%	5.49
C88	1.8977%	2.1158%	4.0135%	5.45
C89	1.6308%	2.4701%	4.1009%	5.58
C90	1.7510%	2.3156%	4.0665%	5.39
C91	1.4831%	2.3847%	3.8678%	5.39
C92	1.9353%	2.0882%	4.0235%	5.41
C93	1.6238%	2.4772%	4.1009%	5.90
C94	1.9553%	2.1457%	4.1009%	5.76

C95	1.5043%	2.5011%	4.0053%	5.71
C96	1.7810%	2.4255%	4.2065%	5.74

7.2.1 Análise dos resultados

Com base nos resultados apresentados nas tabelas, foram gerados os seguintes gráficos para facilitar a análise dos resultados:

- Figuras 17 e 18: comparação entre todas as configurações sem a utilização do UBM.
    - Eixo X: Esforço Computacional;
    - Eixo Y: Percentual de Erro Total.
- Figuras 19 e 20: comparação entre as configurações com GMM ortogonal utilizando UBM.
    - Eixo X: Esforço Computacional;
    - Eixo Y: Percentual de Erro Total.
- Figuras 20 e 21: comparação entre as configurações com GMM ortogonal utilizando UBM.
    - Eixo X: Percentual de Erro de Falsa Negação.
    - Eixo Y: Percentual de Erro de Falsa Identificação.

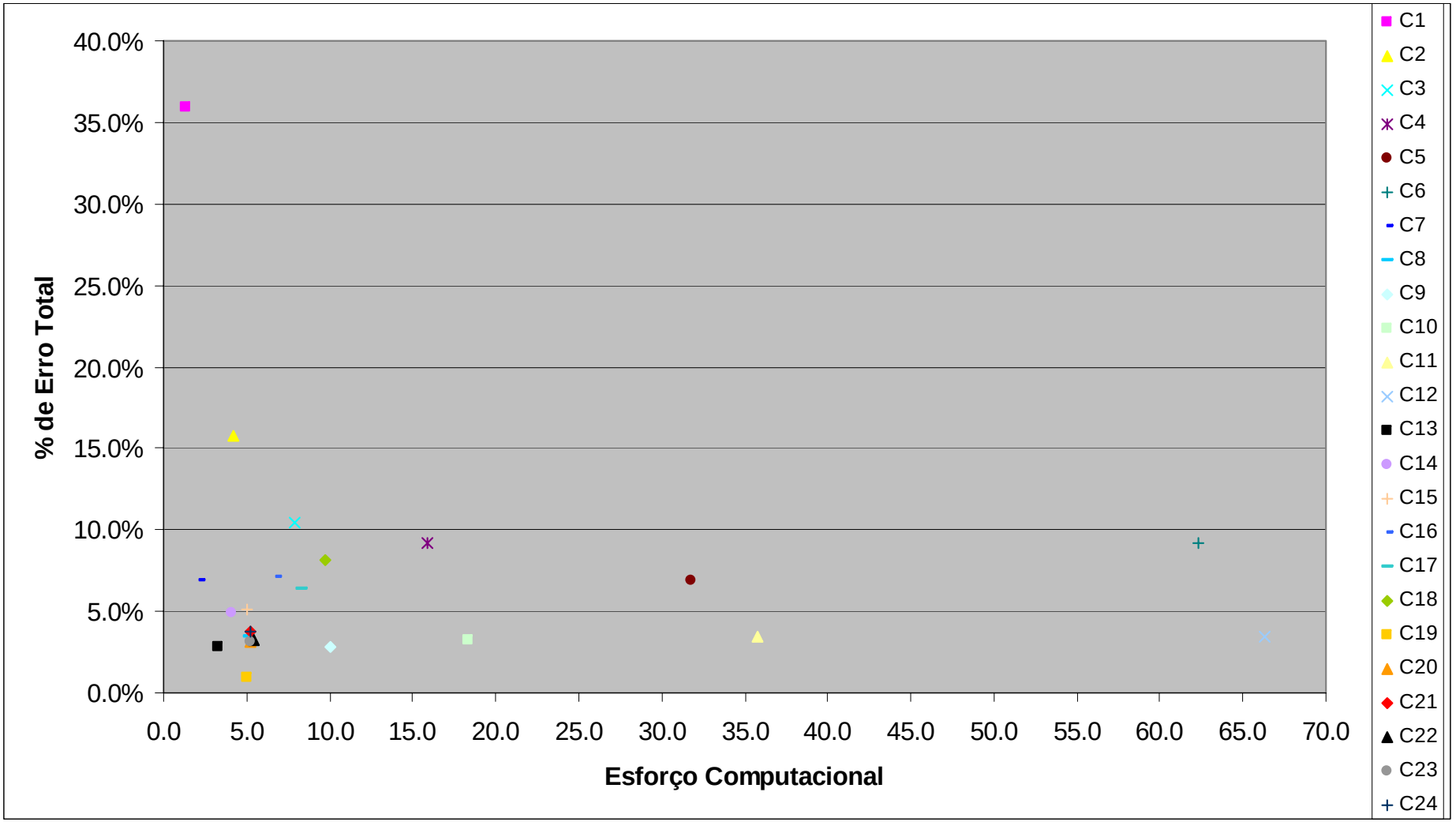

Figura 17 - Comparação dos resultados para reconhecimento sem a utilização do UBM.

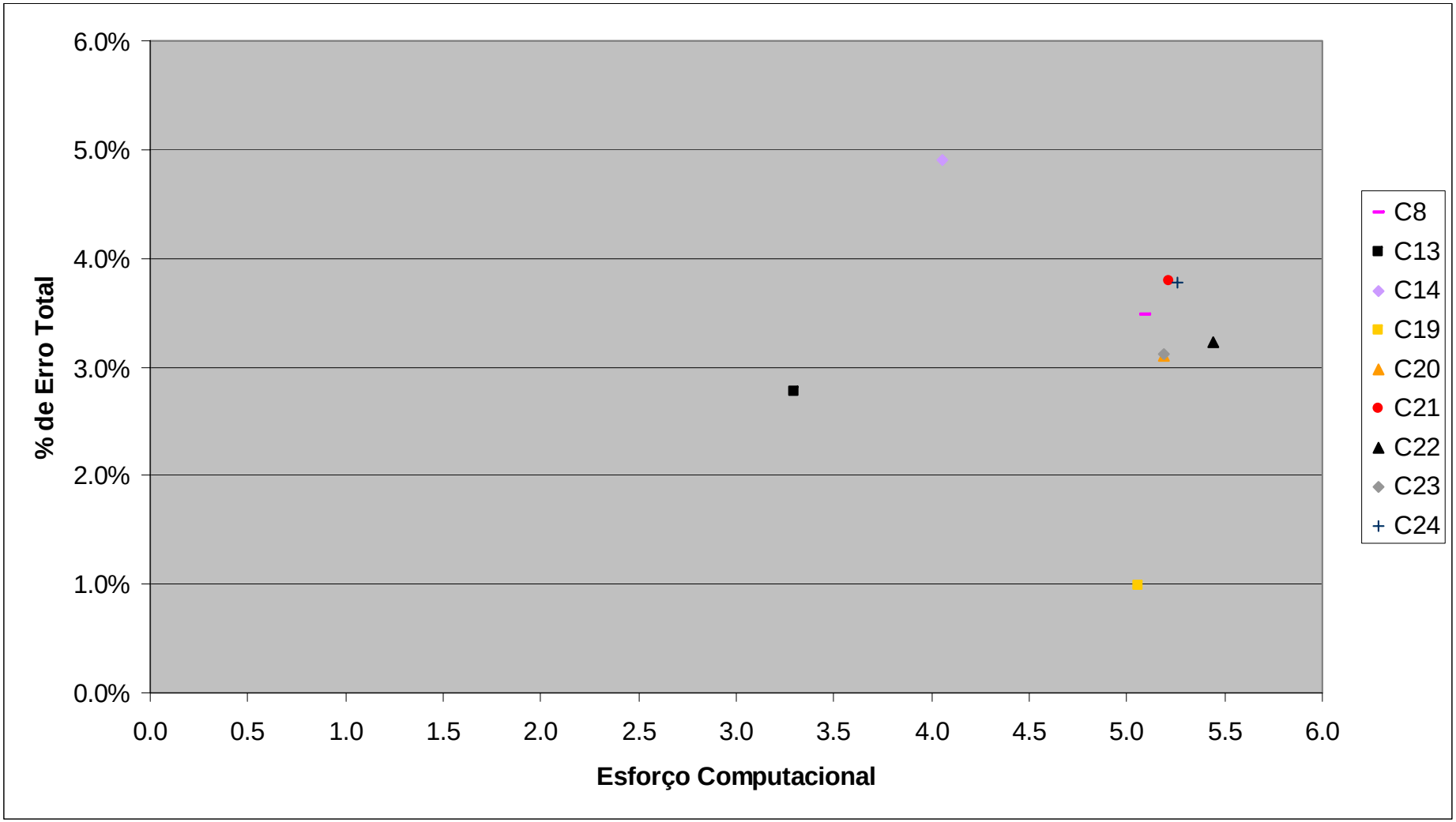

Figura 18 - Ampliação da Figura 17 próxima à origem.

Comparando-se o resultado das configurações C1 até C6 com o resultado das configurações C7 até C12, ou seja, configurações com número fixo de misturas e sem utilização de UBM, pode-se verificar que:

- Como verificado no teste anterior, a configuração com número igual de gaussianas para todos os locutores que apresenta a menor taxa de erro é a que utiliza GMM ortogonal com 16 misturas.
- Conforme esperado, os resultados obtidos mostram que a configuração C1 (GMM convencional, 4 misturas, sem UBM) apresenta o menor esforço computacional, porém a maior taxa de erro total uma vez que possui poucas misturas e assim não modela suficientemente bem as locuções da base de dados utilizada.
- A utilização do GMM ortogonal aumenta em até 56% o esforço computacional em comparação com o GMM convencional;
- Ao se comparar o esforço computacional das configurações que apresentam menor taxa de erro utilizando GMM convencional (C5) e utilizando GMM ortogonal (C9), pode-se verificar que apenas o uso da ortogonalização reduziu em cerca de 4,1% a taxa de erro e em cerca de 68,5% o esforço computacional;

O efeito da utilização do algoritmo de otimização pode ser observado nos resultados obtidos nas configurações C13 até C24:

- A menor taxa de erro foi obtida com a configuração C19, que utiliza GMM ortogonal e fator R=1. Este é um importante resultado, pois mostra que não é necessário pré-definir um valor para o parâmetro R: defini-lo como 1 é o mesmo que deixar o sistema automaticamente buscar o menor valor de $R(j)$ na Equação (18).
- Ao comparar os resultados dos dois conjuntos de valores possíveis de misturas para cada locutor: [4,8,16,32,64,128] e [12,13,14,15,16,17,18,19,20], pode-se perceber que para o mesmo valor do fator R, em todas as situações o conjunto [12,13,14,15,16,17,18,19,20]

obteve melhores resultados. Isso pode ser explicado pelo fato, já observado nos testes anteriores, de que os melhores resultados utilizando-se GMM ortogonal ocorrem para 16 misturas para a base de dados utilizada.

- Nenhuma configuração obteve esforço computacional menor que a configuração C1. Isto era esperado, uma vez que a utilização de mais de 4 misturas em ao menos uma locução da base de dados resulta em mais operações do que as utilizadas quando todos as locuções são treinadas com 4 misturas.

- De forma geral, o aumento do valor do parâmetro R resulta em um aumento no esforço computacional, principalmente para o primeiro conjunto de valores possíveis de misturas: [4,8,16,32,64,128]. Isto ocorre porque para se obter um valor mais alto para $R(j)$ na Equação 18 é necessário aumentar a razão entre a distorção máxima e a distorção média. Assim, ao se aumentar o número de células, aumenta-se automaticamente o esforço computacional. Esse aumento é menos sensível para o segundo conjunto de valores possíveis de misturas, pois os valores são muito próximos, indo de 12 até 20, enquanto no segundo conjunto eles vão de 4 até 128.

Baseado nos resultados obtidos utilizando-se esta base de dados, pode-se verificar que, se não há necessidade de utilizar UBM, a configuração que apresenta a melhor relação Taxa de Erro x Esforço Computacional é a C19 (GMM ortogonal, segundo conjunto de misturas, R = 1,0). Contudo, se é necessário ter o mínimo esforço computacional possível com uma taxa de erro razoável, a configuração C13 (GMM ortogonal, primeiro conjunto de misturas, R = 1,0) apresenta o melhor resultado: redução do esforço computacional em cerca de 35%, mas aumento da taxa de erro em cerca de 1,78% em relação à configuração C19.

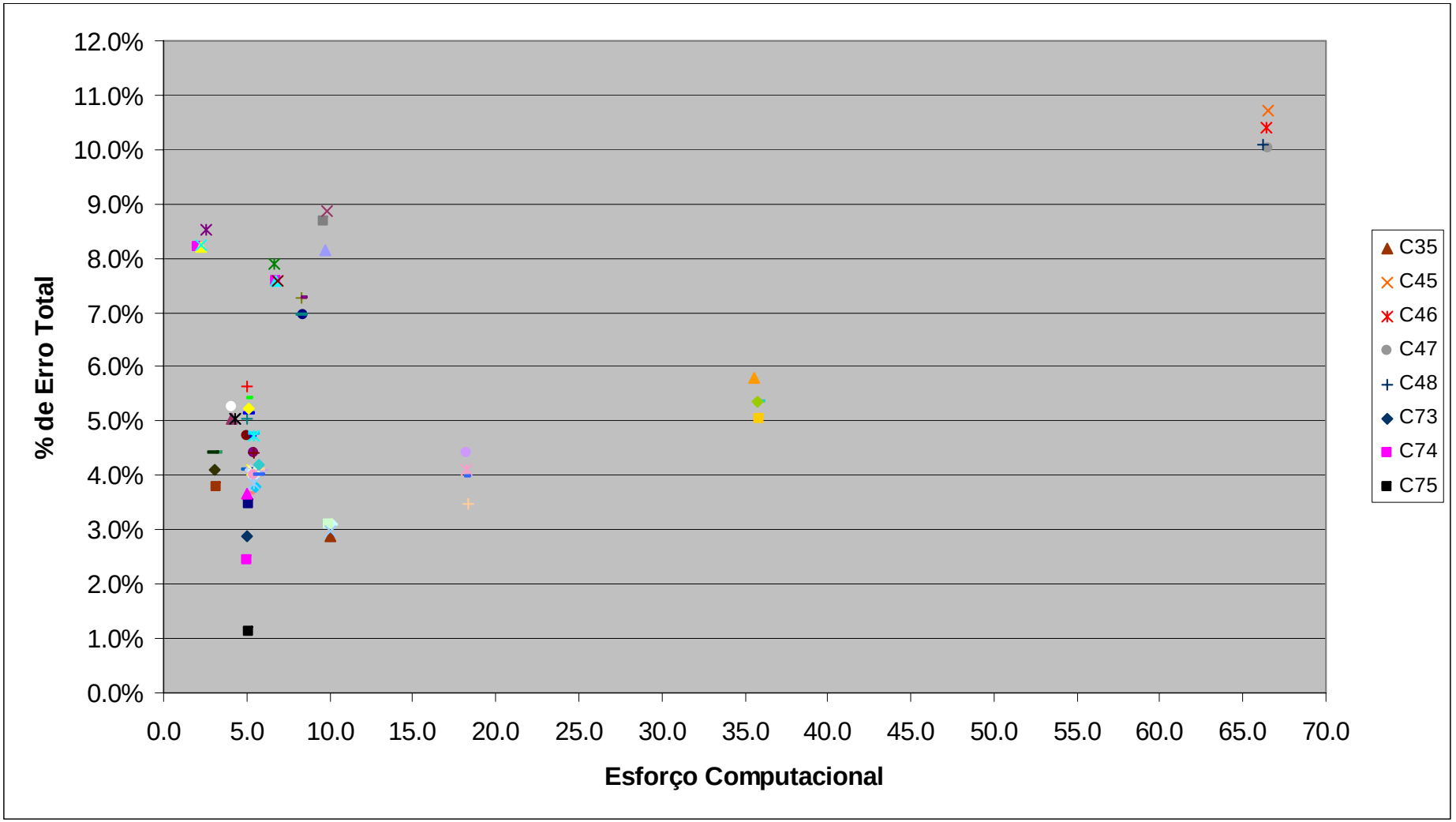

Figura 19 - Comparação entre todas as configurações que utilizam UBM: esforço computacional x taxa de erro total. A legenda contém apenas os valores que apresentaram os resultados mais extremos (canto superior direito e canto inferior esquerdo).

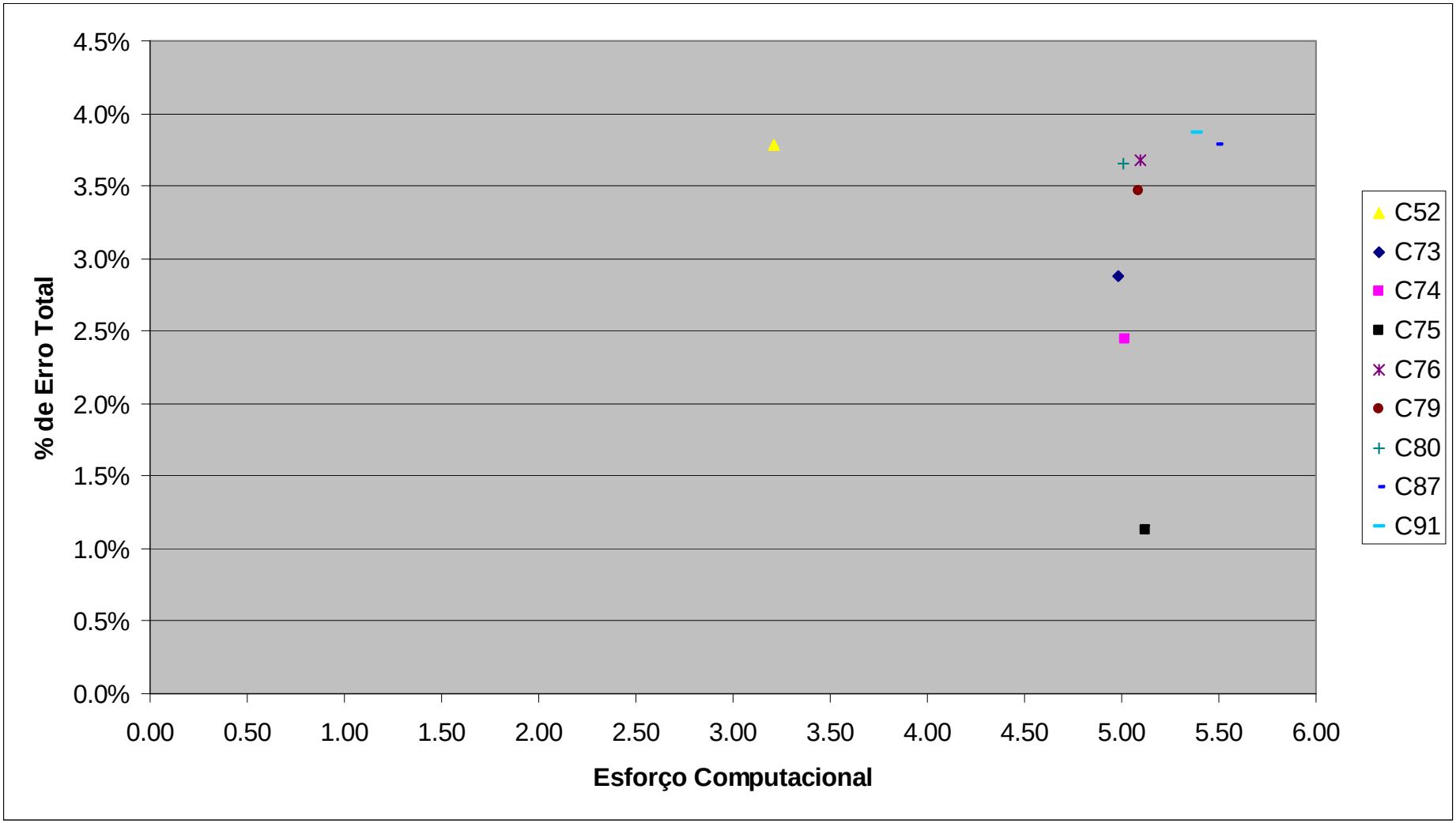

Figura 20 - Ampliação da Figura 19 próxima à origem.

Ao se analisar os resultados dos testes utilizando GMM ortogonal com o algoritmo de otimização e UBM, pode-se verificar que:

- Assim como nos testes sem UBM, as menores taxas de erro são obtidas utilizando-se o algoritmo de otimização e o segundo conjunto de valores possíveis para as misturas ([12,13,14,15,16,17,18,19,20]), reforçando a idéia de que o GMM ortogonal apresenta as menores taxas de erro em torno de 16 misturas.
- A configuração sem utilizar o algoritmo de otimização que apresentou a menor taxa de erro foi a C35 (2,90%). A utilização do UBM resultou em um aumento na taxa de erro em relação à configuração C9 de cerca de 0,01%.
- A configuração C75, utilizando GMM ortogonal, UBM com 256 misturas e fator R=1 apresentou a menor taxa de erro entre todas as configurações testadas. Esse resultado confirma o resultado obtido na Seção 7.1.2 que mostra que o número de misturas ideal para a base de dados utilizada nos testes é de 256. Assim como nos testes sem UBM, o fator R=1 é importante, pois mostra não ser necessário pré-definir este parâmetro antes de iniciar a etapa de treinamento do sistema.
- O esforço computacional das configurações que utilizam o primeiro conjunto de valores possíveis para as misturas é mais sensível ao aumento do fator R, conforme verificado nos testes sem a presença do UBM.
- A configuração utilizando UBM que apresentou o menor esforço computacional foi a C50, cerca de 70% a menos que o esforço computacional obtido na configuração C75. Contudo, a configuração C50 apresenta taxa de erro cerca de 3,3% maior que a configuração C75.

Com base nos resultados obtidos, pode-se verificar que, para a base de dados utilizada, o algoritmo de otimização reduziu a taxa de erro em cerca de 1,78%, comparando-se as configurações C35 (melhor resultado sem o algoritmo de otimização) e C75 (melhor resultado

com o algoritmo de otimização). Em relação ao esforço computacional, a utilização da configuração C75 reduziu o esforço computacional em cerca de 50% em comparação à configuração C35. Contudo, a configuração C52 apresentou o menor esforço computacional (3,21) com taxa de erro inferior a 4% para a base de dados utilizada.

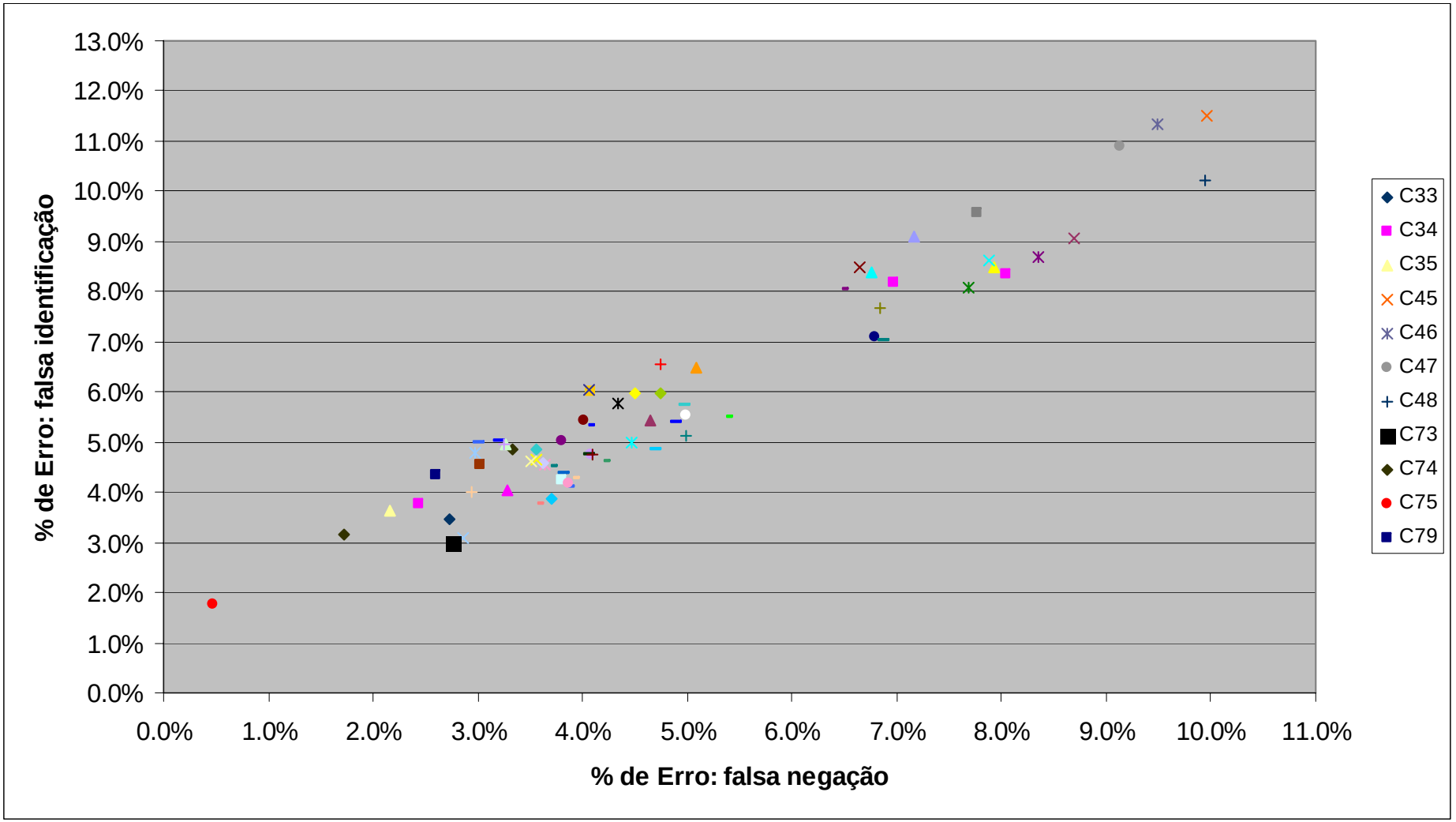

Figura 21 - Comparação entre todas as configurações que utilizam UBM: taxa de erro de falsa negação x taxa de erro de falsa identificação. A legenda contém apenas os valores que apresentaram os resultados mais extremos (canto superior direito e canto inferior esquerdo).

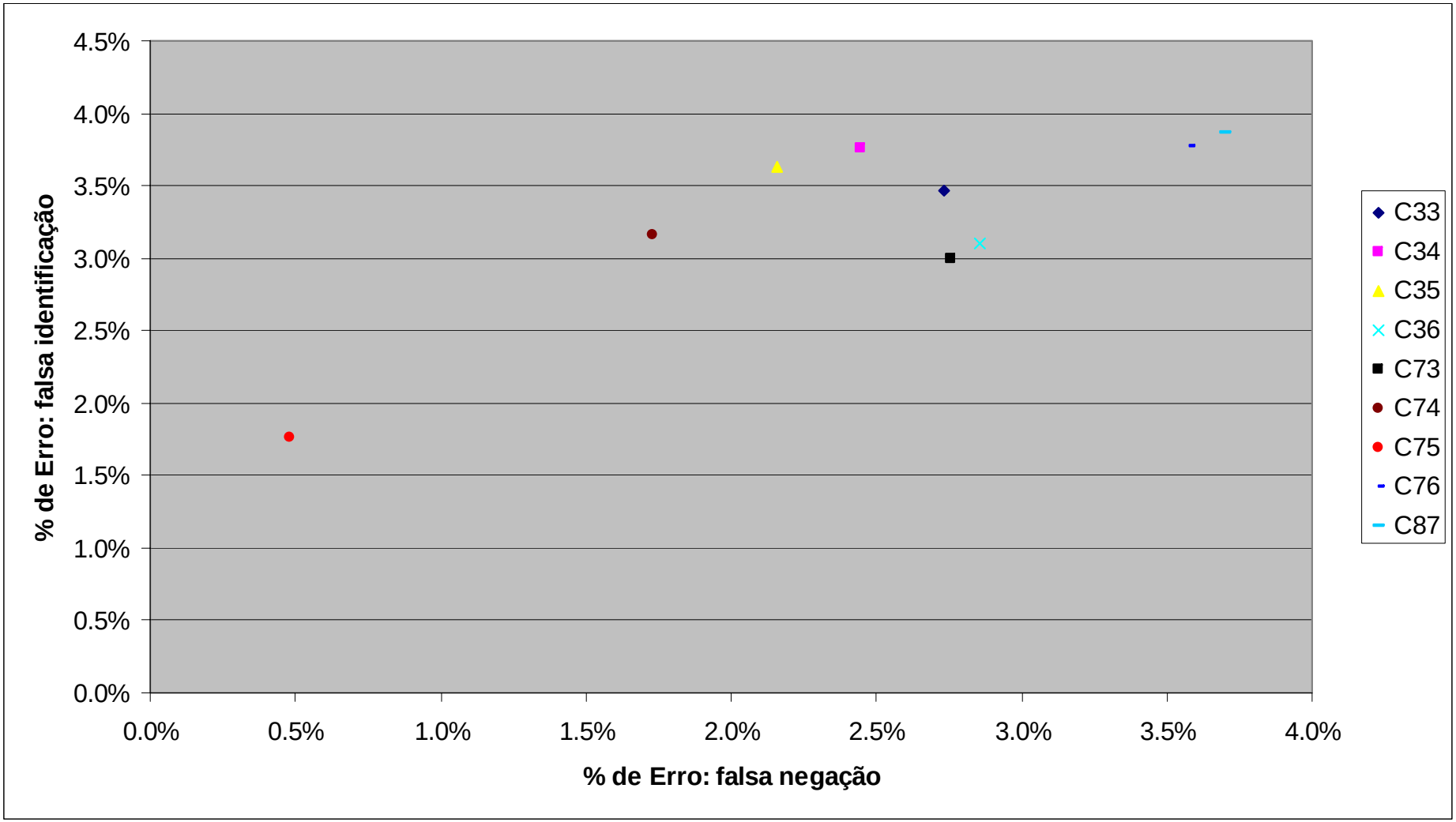

Figura 22 - Ampliação da Figura 21 próxima à origem.

Os resultados mostrados na Figura 21 e na Figura 22 tem como objetivo avaliar se o algoritmo de otimização acentua a ocorrência de erros de falsa negação ou de falsa identificação, ou se ambos são reduzidos. Os resultados mostram que:

- A configuração C75 apresentou os menores valores de erro de falsa identificação e falsa negação. Essa foi a mesma configuração que a apresentou o menor erro total, ou seja, o algoritmo de otimização proposto não acentuou a ocorrência de nenhum dos dois tipos de erros para que o outro fosse diminuído.
- As configurações C45, C46, C47 e C48 apresentaram o pior resultado. Essas quatro configurações utilizam um número fixo de misturas (128) para modelar os usuários e já haviam apresentado os piores resultados nos testes da Seção 7.1.2. A elevada taxa de erro destas configurações pode ser explicado pelo fato da base de dados de treino não ser muito longa, apenas 10 segundos. Assim, não há grande quantidade de dados para serem modelados e o uso de 128 misturas pode causar super modelagem das locuções.
- Nenhuma configuração, com ou sem a utilização do algoritmo de otimização, apresenta favorecimento à ocorrência de erro por falsa negação ou erro por falsa identificação.

7.3 Teste comparativo com outros sistemas

Neste teste é realizada uma comparação do sistema proposto com três sistemas de identificação de locutores: GMM-UBM MAP [24] , TTD GMM [38] , FP GMM [39] .

O GMM-UBM MAP (*maximum a priori*) é um método bem conhecido. Na configuração utilizada nos testes o UBM foi treinado com locuções masculinas e femininas formando um único conjunto, resultando em 2048 componentes gaussianas, conforme sugerido em [24]

O GMM-UBM MAP foi testado utilizando dados de locutores masculinos e femininos combinados para o treinamento de um UBM com 2048 amostras, como sugerido em [24] . O TTD GMM (*training-time-dependent* GMM) é um método de otimização do número de componentes gaussianas baseado na duração do sinal de treinamento. O FP GMM (*frame*

pruning com GMM) é um método de otimização dos trechos de voz das locuções que serão utilizados para o treinamento dos modelos. Foi utilizado o fator T=30, conforme sugerido em [39] .

Foi utilizada a configuração C75 do sistema proposto por ela apresentar a menor taxa de erro e o esforço computacional baixo.

A Tabela 11 apresenta os resultados obtidos para a taxa de erro e o esforço computacional.

Tabela 11 - Resultado do teste comparativo com outras técnicas de identificação de locutor.

Sistema	Taxa de erro: falsa negação	Taxa de erro: falsa identificação	Taxa de erro total	Esforço computacional
C75	0,2395%	0,8829%	1,1224%	5,12
GMM-UBM MAP	0,1622%	1,2171%	1,3793%	28,34
TTD GMM	1,2217%	1,2772%	2,4989%	5,04
FP GMM	1,3322%	2,3323%	3,6645%	10,00

Os resultados mostram o método proposto utilizando a configuração C75 apresenta a menor taxa de erro total do sistema. Contudo, o método GMM-UBM MAP apresenta a menor taxa de erro por falsa negação. Isto deve-se ao fato deste método modelar melhor o UBM do que o método proposto. O método TTD GMM apresentou o menor esforço computacional, mas obteve a taxa de erro cerca de 2% acima do método proposto.

Com base nos resultados, pode-se concluir que, para a base de dados utilizada, o método proposto apresenta a melhor relação entre taxa de erro total e esforço computacional. Caso seja necessário utilizar o menor esforço computacional possível, o método TTD GMM deve ser utilizado.

8 Conclusões

Neste trabalho é proposto um algoritmo de otimização do número de componentes gaussianas utilizado para modelar cada usuário da base de dados em um sistema de identificação de locutor baseado em GMM. O método proposto é independente para cada usuário. Desta forma, a inserção de um novo usuário no sistema não acarreta em novo treinamento de toda a base de dados.

São apresentados os resultados de três conjuntos de testes: teste de ortogonalização, teste de otimização e teste comparativo com outros sistemas. O primeiro tem como objetivo avaliar se a ortogonalização da matriz de covariância traz benefícios à identificação de locutor. O segundo conjunto de testes avalia se o algoritmo de otimização proposto reduz a taxa de erro total do sistema e o esforço computacional.

Os resultados do primeiro conjunto de testes mostram que a ortogonalização reduz em cerca de 4,1% a taxa de erro total do sistema quando não é utilizado UBM nos testes. Além disto, a menor taxa de erro total utilizando-se GMM ortogonal foi obtida com 16 misturas para cada usuário, enquanto para o GMM convencional foram necessárias 64 misturas: redução de 75% no número de misturas. Ainda neste primeiro conjunto de testes, foi verificado que a utilização de UBM em conjunto com o GMM ortogonal aumentou a taxa de erro em cerca de 0,1%. Esse resultado foi obtido utilizando-se UBM modelado com 256 misturas.

Os resultados do segundo conjunto de testes mostram que, para a situação sem utilização do UBM, o algoritmo de otimização diminui a taxa de erro em até 35% sem resultar em um aumento significativo na complexidade computacional (1,78%). Utilizando UBM, a menor taxa de erro, 1,12%, foi obtida modelando-se cada locutor com um número de misturas entre 12 e 20 e o UBM com 256 misturas e fator R=1. É importante ressaltar que R=1 significa não ser necessário pré-definir nenhum limiar para o sistema. O menor esforço computacional foi

obtido modelando-se cada locutor com um número de misturas igual a 8, 16, 32, 64 ou 128 e o UBM com 128 misturas e fator R=1. Essa configuração apresentou um esforço computacional cerca de 70% menor que a configuração com a menor taxa de erro, mas houve um aumento de 3,3% na taxa de erro.

Os resultados do terceiro teste mostram que o algoritmo proposto apresentou taxa de erro comparável a dos outros sistemas testados: 1,12% contra 1,37% do GMM-UBM MAP, 2,49% do TTD-GMM e 3,66% do FP GMM para a base de dados utilizada. O menor esforço computacional foi obtido pelo algoritmo TTD-GMM, sendo cerca de 2% menor do que o esforço computacional do algoritmo proposto.

De forma resumida:

- Este trabalho apresenta um algoritmo de otimização do número de componentes gaussianas utilizado para modelagem de cada locutor em um sistema de identificação de locutores;
- Foram realizados três conjuntos de testes para verificação do desempenho do algoritmo proposto;
- Foi verificado que o uso do GMM ortogonal reduz em cerca de 4,1% a taxa de erro em comparação ao GMM convencional;
- O algoritmo de otimização proposto reduziu em até 35% a taxa de erro em comparação com o melhor resultado obtido pelo GMM ortogonal sem otimização;
- A menor taxa de erro para a base de dados utilizada foi obtida utilizando-se o algoritmo de otimização: 1,12%;
- O algoritmo proposto obteve, para a base de dados utilizada, taxa de erro (1,12%) comparável a dos outros três métodos de identificação de locutor: GMM-UBM MAP (1,37%), TTD-GMM (2,49%) e FP GMM (3,66%).

9 Trabalhos futuros

- Validação do algoritmo proposto em situações adversas, como presença de ruído telefônico e perda de pacotes em uma rede IP;
- Validação do algoritmo proposto para bases de treinamento com duração inferior a 10 segundos, utilizado neste trabalho. Para aplicações comerciais é importante o teste com bases de treinamento curtas, de até 1 segundo, o equivalente à duração de aproximadamente uma palavra;
- Utilização conjunta do algoritmo proposto com algoritmos de separação de trechos com voz e sem voz nas locuções, a fim de utilizar-se somente informação de voz na modelagem dos locutores.

Referências

[1] REYNOLDS, D.A., "Robust text-independent speaker identification using Gaussian mixture speaker model", *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 1, pp72-83, Janeiro 1995.

[2] PRZYBOCKI, M., MARTIN, A., "NIST speaker recognition evaluation chronicles", 2004.

[3] RABINER, L., "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. of the IEEE*, Vol. 77, No 2, Fevereiro 1989.

[4] BIMBOT, F., MATHAN L., LIMA, A., CHOLLET, G., "Standard and target driven AR-vector models for speech analysis and speaker recognition", *Proc. of ICASSP*, Vol. 2, ppII-5-II-8, Março 1992.

[5] MONTACIE, C., FLOCH, J.L., "AR-vector models for free-text speaker recognition", *Proc. of ICSLP*, pp611-614, 1992.

[6] CHAGNOLLEU, I.M., DUROU, G., "Application of time-frequency principal component analysis to speaker verification", *Digital Signal Processing*, Vol. 10, pp226-236, 2000.

[7] FURUI, S., *Automatic speech and speaker recognition, advanced topics*, Boston: Kluwer Academic Publishers, 1996.

[8] MARTIN, A., PRZYBOCKI, M., "The NIST 1999 recognition evaluation – an overview", *Digital Signal Processing*, Vol. 10, pp1-18, 2000.

[9] CAMPBELL, W.M., ASSALEH, K.T., "Polynomial classifier techniques for speaker verification", *Proc. of ICASSP*, Vol. 1, pp321-324, Março, 1999.

[10] LIMA, C.B., "Sistemas de verificação de locutor independente do texto baseados em GMM e AR-Vetorial utilizando PCA", Dissertação de Mestrado, *Instituto Militar de Engenharia*, Rio de Janeiro, 2001.

[11] REYNOLDS, D.A., "A guassian mixture model modeling approach to text independent speaker identification", Tese de Doutorado, *Georgia Institute of Technology*, 1992.

[12] MALAYATH, N., "Data-driven methods for extracting features from speech", Tese de Doutorado, *Oregon Graduate Institute*, 2000.

[13] SOTOMAYOR, C.A., "Realce de voz aplicado à verificação automática de locutor", Tese de Mestrado, *Instituto Militar de Engenharia*, 2003.

[14] SARMA, S.V., "A segment-based speaker verification system using SUMMIT", Tese de Mestrado, *Massachussets Institute of Technology*, 1999.

[15] FREDOUILLE, C., BONASTRE, J.F., MERLIN, T., “AMIRAL: a block-segmental multirecognizer architecture for speaker recognition”, *Digital Signal Processing*, Vol. 10, pp172-197, 2000.

[16] DELACRETAZ, D.P., “Segmental approaches for automatic speaker verification”, *Digital Signal Processing*, Vol. 10, pp198-212, 2000.

[17] SAVIC, M. SORENSEN, J., “Phoneme based speaker verification”, *Trans. of IEEE*, ppII-165-II-168, 1992.

[18] AUCKENTHALER, R., PARRIS, E.S., CAREY, M.J., “Improving a GMM speaker verification system by phonetic weighting”, *Proc. of ICASSP*, Vol. 1, pp313-316, Março, 1999.

[19] REYNOLDS, D.A., “Experimental evaluation of features for robust speaker identification”, *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 4, pp639-643, Outubro 1994.

[20] VERGIN, R., SHAUGHNESSY, D., “A double gaussian mixture modeling approach to speaker recognition”, *Proc. of Eurospeech*, 2001.

[21] REYNOLDS, D.A., HECK, L.P., “Automatic speaker recognition-recent progress, current applications and future trends”, *AAAS 2000 Meeting Humans, Computers and Speech Symposium*, Fevereiro 2000.

[22] VUUREN, V.S., “Speaker verification in a time-feature space”, Tese de Doutorado, *Oregon Graduate Institute of Science and Technology*, 1999.

[23] LIMA, C.B., “Sistemas de verificação de locator independente do texto baseados em GMM e AR-Vetorial utilizando PCA”, Tese de Mestrado, *Instituto Militar de Engenharia*, 2001.

[24] REYNOLDS, D.A, QUATIERI, T.F., DUNN, R.B., “Speaker verification using adapted gaussian mixture models”, *Digital Signal Processing*, Vol. 10, pp19-41, 2000.

[25] ATAL, B.S., “Automatic recognition of speaker from their voices”, *Proc. of the IEEE*, Vol. 64, pp460-475, Abril 1976.

[26] DODDINGTON, G.R., “Speaker recognition – identifying people by their voices”, *Proc. of the IEEE*, Vol. 73, pp1641-1664, Novembro 1985.

[27] CAMPBELL, J.P., “Speaker recognition: a tutorial”, *Proc. of the IEEE*, Vol. 85, pp1437-1462, Setembro 1997.

[28] AGUIRRE, L.A., *Introdução à identificação de sistemas – técnicas lineares e não-lineares aplicadas a sistemas reais*, Editora UFMG, 2000.

[29] LIU, L., JIALONG, H., “On the use of orthogonal GMM in speaker recognition”, *Proc. of ICASSP*, Vol. 2, pp845-848, Março, 1999.

[30] SINGH, G., PANDA, A., BHATTACHARYYA, S., SRIKANTHAN, T., ”Vector quantization techniques for GMM based speaker verification”, *Proc. of ICASSP*, Vol. 2, ppII65-8, Abril, 2003.

[31] SIVAKUMARAN, P., ARIYAEEINIA, A.M., ”The use of sub-band cepstrum in speaker verification”, *Proc. of ICASSP*, Vol. 2, ppII1073-II1076, 2000.

[32] REYNOLDS, D.A., “Comparison of background normalization methods for text-independent speaker verification”, *Proc. of Eurospeech*, 1997.

[33] IMAI, S., SUMITA, K., FURUICHI, C., “Mel log spectrum approximation (MLSA) filter for speech synthesis”, *Trans. IECE*, Vol. JGG-A, pp122-129, 1983

[34] TOKUDA, K., KOBAYASHI, T., FUKADA, T., SAITO, H., IMAI, S., “Spectral estimation of speech based on mel-cepstral representation”, *Trans. IEICE*, vol. Ji4-A, pp1240-1248, 1991.

[35] GOHBERG, I., KOLTRACHT, I., *Efficient algorithm for Toeplitz plus Hankel matrices*, Integral Equations and Operator Theory, Vol. 12, pp136-142, 1989.

[36] WIDROW, B., STEARNS, S. D., *Adaptive Signal Processing*, Englewood Cliffs, N.J.: Prentice-Hall. 1985.

[37] G.M. University, “Speech accent archive”, http://accent.gmu.edu.

[38] TADJ, C., DUMOUCHEL, P., OUELLET, P., ”GMM based speech identification using training-time-dependet number of mixtures”,

[39] BESACIER, L., BONASTRE, J.F., “Frame pruning for speaker recognition”, *Proc. of ICASSP*, Vol. 2, pp765-768, Maio, 1998.

[40] SANT'ANA, R.; COELHO, R.F; Alcaim, A.. “Text-Independent Speaker Recognition Based on the Hurst Parameter and the Multidimensional Fractional Brownian Motion Model”, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 3, pp931-940, 2006.

[41] SANT'ANA, R.; COELHO, R.F.; Alcaim, A.. “On the Performance of Hurst-Vectors for Speaker Identification Systems”, *Lecture Notes in Computer Science*, Vol. 3686, pp514-521, 2005.

[42] TOKUDA, K., KOBAYASHI, T., MASUKO, T., IMAI, S., “Mel-generalized Cepstral Analysis – A Unified Approach to Speech Spectral Estimation”, *Proc. of ICSLP*, Vol. 3, pp1043-1046, Setembro. 1994.

[43] DRYGAJLO, A. EL-MALIKI, M., “Speaker verification in noisy environments with combined spectral subtraction and missing feature theory”, *Proc. of ICASSP*, Vol. 1, pp121-124, Maio 1998.

[44] FINE, S., NAVRATIL, J., GOPINATH, R.A., “A hybrid GMM/SVM approach to speaker identification”, *Proc. of ICASSP*, Vol. 1, pp417-420, 2001.

[45] MEDINA, C.A. ; APOLINÁRIO JR, J.A. ; ALCAIM, A. ; ALVES, R. G. .”Robust Speaker Verification in Colored Noise Environment”, *In: 37th Annual Asilomar Conference on Signals, Systems and Computers*, pp4, 2003..

[46] MEDINA, C.A. ; APOLINÁRIO JR, J.A. ; ALCAIM, A. .”Improving the Performance of Speaker Verification Systems in Noisy Environments”, *Proc. of the ICSES*, pp413-416, 2004.

[47] MURTHY, H.A., BEAUFAYS, F., HECK, L.P., WEINTRAUB, M., “Robust text-independent speaker identification over telephonechannels”, *IEEE Transactions on Speech and Audio Processing*,Vol. 7, Setembro 1999.

[48] RABINER, L. R., JUANG, B., *Fundamentals on Speech Recognition*, New Jersey, Prentice Hall, 1996.

Apêndice 1

Coeficientes Mel-cepstrais normalizados

O espectro $G(e^{j\omega})$ pode ser representado por coeficientes mel-cepstrais generalizados [42] $c(d)$ de ordem D através de

$$G(z) = \exp \sum_{d=0}^{D} c(d)\tilde{z}^{-d} \tag{21}$$

onde

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, |\alpha| < 1 \tag{22}$$

A característica de fase da função de transferência $\tilde{z}^{-1} = e^{-j\omega}$ é dada por

$$\tilde{\omega} = \tan^{-1} \frac{(1-\alpha^2)\sin(\omega)}{(1+\alpha^2)\cos(\omega) - 2\alpha} \tag{23}$$

Para obter uma estimativa não tendenciosa, é utilizado o critério a seguir [33] , que é minimizado em relação a $\{c(d)\}_{d=0}^{D}$:

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} \{e^{R(\omega)} - R(\omega) - 1\} d\omega \tag{24}$$

onde

$$R(\omega) = \log(I_N(\omega)) - \log\left|H(e^{j\omega})^2\right| \tag{25}$$

e $I_N(\omega)$ é o periodograma modificado de um processo $x(n)$ fracamente estacionário com uma janela no tempo de tamanho N. Para extrairmos o ganho K de $G(z)$, a equação (3) é reescrita como

$$G(z) = \exp \sum_{d=0}^{D} b(d)\Phi_d(z) = KR(z) \tag{26}$$

onde

$$K = e^{b(0)} \tag{27}$$

e

$$R(z) = \exp \sum_{d=1}^{D} b(d)\Phi(z) \tag{28}$$

sendo

$$c(d) = \begin{cases} b(d), & d = D \\ b(d) + \alpha b(d+1), & 0 \le d < D \end{cases} \tag{29}$$

e

$$\mathbf{\Phi}_d(z) = \begin{cases} 1, & d = 0 \\ \dfrac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}} \tilde{z}^{-(d-1)}, & d \ge 1 \end{cases} \tag{30}$$

Como $G(z)$ é um sistema de fase mínima, foi mostrado em [42] que a minimização de E com relação a $\{c(d)\}_{d=0}^{D}$ é equivalente à minimização de

$$\varepsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{I_N(\omega)}{\left|R\left(e^{j\omega}\right)\right|^2} d\omega \tag{31}$$

com relação a

$$\mathbf{b} = [b(1), b(2), ..., b(D)]^T \tag{32}$$

O ganho K que minimiza E é obtido fazendo-se $\partial E \setminus \partial K = 0$:

$$K = \sqrt{\varepsilon_{\min}} \tag{33}$$

onde $\varepsilon_{\min}$ é o valor minimizado de ε.

Uma vez que ε é convexo com relação a $\mathbf{b}$ [34] , o problema de minimização de (11) pode ser resolvido pelo método de Newton-Raphson. Para o *i*-ésimo resultado $\mathbf{b}^{(i)}$, resolvendo um conjunto de equações lineares

$$\mathbf{H}\Delta\mathbf{b}^{(i)} = -\nabla\varepsilon\Big|_{\mathbf{b}=\mathbf{b}^{(i)}} \tag{34}$$

são encontrados os valores

$$\Delta \mathbf{b}^{(i)} = \left[\Delta b^{(i)}(1), \Delta b^{(i)}(2), ..., \Delta b^{(i)}(D)\right]^T \tag{35}$$

onde $\mathbf{H}$ é a matriz Hessiana $\mathbf{H} = \partial^2 \varepsilon / \partial \mathbf{b} \partial \mathbf{b}^T$. Então, o próximo resultado é obtido por

$$\mathbf{b}^{(i+1)} = \mathbf{b}^{(i)} + \Delta \mathbf{b}^{(i)} \tag{36}$$

O gradiente $\nabla \varepsilon$ é dado por

$$\nabla \varepsilon = -2\tilde{\mathbf{r}} = -2\left[\tilde{r}(1), \tilde{r}(2), ..., \tilde{r}(D)\right]^T, \tag{37}$$

onde

$$\tilde{r}(d) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{I_N(\omega)}{\left|R\left(e^{j\omega}\right)\right|^2} \Phi_m^*\left(e^{j\omega}\right) d\omega \tag{38}$$

e a matriz Hessiana $\mathbf{H}$ é calculada através de

$$\mathbf{H} = 2\left\{h(i, j)\right\}_{i,j=1}^{D} \tag{39}$$

onde

$$h(i, j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{I_N(\omega)}{\left|R\left(e^{j\omega}\right)\right|^2} \left\{\Phi_i\left(e^{j\omega}\right) + \Phi_i^*\left(e^{j\omega}\right)\right\} \Phi_j^*\left(e^{j\omega}\right) d\omega \tag{40}$$

Como a matriz $\mathbf{H}$ é uma matriz simétrica Toeplitz e Hankel, (14) pode ser resolvido utilizando um algoritmo rápido recursivo [35] . Os coeficientes $h(i, j)$ e $\tilde{r}(d)$ podem ser calculados eficientemente através da FFT e de fórmulas de recursão [34] . Um valor inicial de $\mathbf{b}^{(0)}$ pode ser obtido pela FFT cepstral utilizando também uma fórmula de recursão [34] .

Substituindo $\mathbf{H}$ na equação (14) pela matriz identidade, pode-se derivar o método *steepest descent* do método de Newton-Raphson. Isto é, do i-ésimo resultado $\mathbf{b}^{(i)}$ o próximo valor $\mathbf{b}^{(i+1)}$ é dado por

$$\mathbf{b}^{(i+1)} = \mathbf{b}^{(i)} - \mu \Delta \varepsilon \big|_{\mathbf{b}=\mathbf{b}^{(i)}}, \tag{41}$$

onde μ é o passo de adaptação.

Assumindo que o tamanho N da janela de tempo é suficientemente grande, pode-se interpretar (11) como o valor esperado do quadrado de $e(n)$

$$\varepsilon = E\left[e^2(n)\right] \tag{42}$$

onde $e(n)$ é a saída do filtro inverso alimentado por $x(n)$. De acordo com o que foi assumido, o gradiente calculado na equação (17) pode ser reescrito como

$$\nabla\varepsilon = -2E\left[e(n)\mathbf{e}_{\Phi}^{(n)}\right] \tag{43}$$

onde

$$\mathbf{e}_{\Phi}^{(n)} = \left[e_1(n), e_2(n), ..., e_D(n)\right]^T \tag{44}$$

e $e_d(n)$ é a saída do filtro $\Phi_d(z)$.

Para se chegar a um algoritmo adaptativo, é introduzida uma estimativa instantânea similar ao algoritmo LMS [36]

$$\hat{\nabla}\varepsilon^{(n)} = -2e(n)\mathbf{e}_{\Phi}^{(n)} \tag{45}$$

De forma a suprimir a flutuação de $\mathbf{b}$, $\nabla\varepsilon$ é estimado utilizando-se uma janela exponencial da seguinte forma

$$\overline{\nabla}\varepsilon^{(n)} = -2(1-\tau)\sum_{i=-\infty}^{n}\tau^{n-i}e(i)\mathbf{e}_{\Phi}^{(i)} = \tau\overline{\nabla}\varepsilon^{(n-1)} - 2(1-\tau)e(n)\mathbf{e}_{\Phi}^{(n)},\ 0 \le \tau < 1. \tag{46}$$

Com esta estimativa do gradiente, é possível especificar um algoritmo adaptativo baseado no método *steepest descent*: os coeficientes do vetor $\mathbf{b}^{(n)}$ no instante n são atualizados por

$$\mathbf{b}^{(n+1)} = \mathbf{b}^{(n)} - \mu^{(n)}\overline{\nabla}\varepsilon^{(n)} \tag{47}$$

Quando o ganho do sinal $x(n)$ é variante no tempo, μ é normalizado por

$$\mu^{(n)} = \frac{a}{D\varepsilon^{(n)}},\ 0 < a < 1 \tag{48}$$

onde $\varepsilon^{(n)}$ é uma estimativa de ε no instante n, dada por

$$\varepsilon^{(n)} = (1-\lambda)\sum_{i=-\infty}^{n} \lambda^{n-i} e^2(i) = \lambda\varepsilon^{(n-1)} + (1-\lambda)e^2(n),\ 0 \le \lambda < 1. \quad (49)$$

Utilizando (13), pode-se estimar K no instante n a partir de $\varepsilon^{(n)}$.

Os coeficientes mel-cesptrais $\{c(d)\}_{d=0}^{D}$ podem ser obtidos de K e **b** usando-se (7) e (9).

Uma vez calculados os coeficientes mel-cepstrais, eles serão normalizados em energia através de

$$c_{norm}^{t}(0) = \frac{c^{t}(0)}{c_{\max}(0)},\ 1 \le t < T \quad (50)$$

onde t representa o t-ésimo bloco, T o total de blocos do sinal, c_{norm}^{t} é o vetor de coeficientes normalizados do t-ésimo bloco e $c_{\max}(0)$ é o maior valor de $c(0)$ dentre todos os blocos do sinal.

Após a normalização, a média de cada uma das seqüências de coeficientes ao longo dos blocos do sinal é modificada para zero através de

$$c_{norm}^{t}(d) = c^{t}(d) - c_{médio}(d),\ 1 \le \mathrm{d} < \mathrm{D} \quad (51)$$

onde

$$c_{médio}(d) = \frac{1}{T}\sum_{t=1}^{T} c^{t}(d),\ 1 \le d < D. \quad (52)$$

Apêndice 2

Lista de publicações:

R. da S. Maia, R. J. da R. Cirigliano, D. Rojtenberg, F.G.V. Resende Jr., “Comparative analysis among fixed codebook design techniques in CELP speech coders”, SAWCAS, 2001.

R. da S. Maia, R. J. da R. Cirigliano, D. Rojtenberg, F.G.V. Resende Jr., “CELP speech coding: a comparison in terms of quantization techniques for the synthesis filter parameters”, ITS, 2002.

R. da S. Maia, R. J. da R. Cirigliano, D. Rojtenberg, F.G.V. Resende Jr., “An HMM-based phonetic vocoder using mixed excitation”, ICOSYS, 2002.

F. Paiva, G.C.R. Abrahão, R.J.R. Cirigliano, R.S. Maia, F.G.V. Resende Jr., “Conversão de arquivos WAVE em MDI”, AES, 2003.

R. da S. Maia, R. J. da R. Cirigliano, D. Rojtenberg, F.G.V. Resende Jr., “Mixed-excited phonetic vocoder at 265 bps”, ICASSP, 2003.

R. J. da R. Cirigliano, D. Rojtenberg, R. da S. Maia, F.G.V. Resende Jr., “Codificação de voz CELP em tempo real a 4 kbps”, Semana de Eletrônica-UFRJ, 2003.

R. J. da R. Cirigliano, F.G.V. Resende Jr., “Sistema de reconhecimento de comandos utilizando linguagem natural”, Semana de eletrônica-UFRJ, 2004.

Ricardo J.R. Cirigliano, Clarisse Monteiro, Filipe Leandro de F. Barbosa, Fernando Gil Vianna Resende Jr., Letícia Rebollo Couto, João A. de Moraes, “Um cinjunto de 1000 frases balanceadas para o português brasileiro obtido utilizando a abordagem de algoritmos genéticos”, SBT, 2005.

Ricardo J.R. Cirigliano, Fernando Gil Vianna Resende Jr., “Codificador CELP a 4 kbps utilizando coeficientes mel-cepstrais generalizados”, SBT, 2005.

R.J.R. Cirigliano, F.G.V. Resende Jr., “Optimization of the number of Gaussian components for speaker identification”, ITS, 2006.