CENTRO UNIVERSITÁRIO DA FEI

RAFFAELLO CLASER

ESTUDO E APLICAÇÃO DA TÉCNICA MATCHING PURSUIT NA CLASSIFICAÇÃO ENTRE SINAIS DE VOZ E SILÊNCIO

São Bernardo do Campo

2013

RAFFAELLO CLASER

ESTUDO E APLICAÇÃO DA TÉCNICA MATCHING PURSUIT NA CLASSIFICAÇÃO ENTRE SINAIS DE VOZ E SILÊNCIO

Dissertação apresentada ao Centro Universitário da FEI, para a obtenção de Título de Mestre em Engenharia Elétrica, na Área de Inteligência Artificial aplicada a Automação.

Orientador: Prof. Dr. Ivandro Sanches

São Bernardo do Campo

2013

Claser, Raffaello.
Estudo e aplicação da técnica Matching Pursuit na classificação entre sinais de voz e silêncio / Raffaello Claser. São Bernardo do Campo, 2012.
49 f. : il.

Dissertação (Mestrado) - Centro Universitário da FEI.
Orientador: Profº. Dr Ivandro Sanches

1. Matching Pursuit. 2. Dicionários redundantes. 3. Detecção de voz. I. Sanches, Ivandro, orient. II. Título.

CDU 62-52

Centro Universitário da **FEI**

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA JULGADORA

PGE-10

Programa de Mestrado de Engenharia Elétrica

Aluno: Raffaello Claser **Matrícula:** 111115-2

Título do Trabalho: Estudo e aplicação da técnica matching pursuit na classificação entre sinais de voz e silêncio.

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Ivandro Sanches

Data da realização da defesa: 21/12/2012

A Banca Julgadora abaixo-assinada atribuiu ao aluno o seguinte:

APROVADO ☒ REPROVADO ☐

São Bernardo do Campo, 21 / 12 /2012

MEMBROS DA BANCA JULGADORA

Prof. Dr. Ivandro Sanches

Ass.: _____

Prof. Dr. José Carlos de Souza Júnior

Ass.: _____

Prof. Dr. Vítor Heloiz Nascimento

Ass.: _____

VERSÃO FINAL DA DISSERTAÇÃO

ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

Dedico esta dissertação aos meus exemplos de vida, João Carlos Claser (in memorium), Giuseppina Gattuso Claser e Carla Claser que, sempre, me estimularam a dar este grande passo. Estas três pessoas com muita sabedoria, discernimento, bom senso e dedicação estiveram ao meu lado me encorajando nas horas difíceis e me aplaudindo nos momentos de glória.

AGRADECIMENTOS

A Deus, que se mostrou criador, que foi criativo. Seu fôlego de vida em mim me foi sustento e me deu coragem para questionar realidades e propor sempre um novo mundo de possibilidades.

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Minha querida irmã, apesar da enorme distância, você sempre esteve preocupada com o meu sucesso e por isso sempre me encorajou e nunca me deixou desistir.

A todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.

RESUMO

As transformadas de Fourier e Wavelet são as representações/transformações mais comumentes utilizadas para se referir a um dado sinal, por serem rápidas e fáceis de se manipular. Porém, em casos em que a representação é construída a partir da seleção de elementos de conjuntos redundantes chamados de "dicionários", o uso de técnicas alternativas que permite uma maior "esparsidade" (dispersão) se faz necessário. Dessa forma, o objetivo deste trabalho visa buscar dicionários adequados de forma a solucionar o problema de se classificar trechos de sinal entre voz e silêncio utilizando dicionários redundantes e representação esparsa de sinais. Para esse fim, constrói-se um dicionário redundante de funções básicas (átomos) e analisa-se o sinal de voz via Matching Pursuit. Dessa análise, fase de treinamento, obtém-se a distribuição de probabilidade discreta a priori de ocorrência do conjunto de átomos para cada classe de interesse, permitindo a discriminação a posteriori entre as classes. Surpreendentemente, a técnica mencionada anteriormente não se baseia na variação de níveis de energia ao longo do sinal, mas nas características fundamentais que determinam a essência de cada uma dessas duas classes de sinais, nominalmente voz e silêncio. Entretanto, devido a ineficiência apresentada pela mesma, precisou-se utilizar os pesos dos átomos, os quais contém informação de energia, de forma a melhorar e refinar a classificação desempenhada pelo algoritmo.

Palavras-chave: *Matching Pursuit*, Dicionários Redundantes, Histogramas, Detecção de Voz.

ABSTRACT

The Fourier and Wavelet transforms are the representations/transformations more commonly used to refer to a given sign, because they are fast and easily to manipulate. However, in cases where the representation is builded by the selection of a set of redundant elements called "dictionaries", the use of alternative techniques which allows a greater "sparcity" (dispertion) it's necessary. Thus, the objective of this study aims to seek appropriate dictionaries in order to solve the problem of classifying passages of signal between voice and silence using redundantes dictionaries and sparse signals representation. To this end, construct a redundant dictionary of basic functions (atoms) and analyzes the speech signal via Matching Pursuit. From this analysis, the training phase, one obtains the discrete probability distribution a priori of occurrence of a collection of atoms for each class of interest, subsequently allowing discrimination between classes. Surprisingly, the aforementioned technique does not rely on the variation of energy level throughout the signal, but the fundamental characteristics which determine the essence of each of these two classes of signals, namely voice and silence. However, due to inefficiency presented by itself, had to be used weights of atoms, which contains energy information, in order to improve and refine the classification performed by the algorithm.

Keywords: *Matching Pursuit*, Redundant Dictionaries, Histograms, Voice Detection.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 REVISÃO BIBLIOGRÁFICA

1.1 Introdução

Recentemente, tem-se intensificado não só o estudo de técnicas capazes de obter representações mais compactas de sinais de forma a reduzir a largura de banda necessária em canais de transmissão de informação (DYMARSKI1; MOREAU; RICHARD, 2011), mas também, o estudo de técnicas que possibilitem classificar trechos de sinal entre as classes de voz e silêncio (KLING; ROADS, 2004). Essas técnicas têm obtido sucesso no processamento de sinais de áudio para fins de transcrição musical (NEEDELL; VERSHYNIN, 2010; STURM; CHRISTENSEN, 2010) e codificação (MALLAT; ZHANG, 1993; VERA-CANDEAS et al., 2004). A codificação de sinais de voz envolve diversas etapas tais como a análise, quantização, codificação e síntese. Para que o codificador apresente bom desempenho, é fundamental utilizar um método de análise capaz de aproximar o sinal com o menor número possível de coeficientes.

Normalmente, essa aproximação é realizada através de decomposições que aproximam os sinais que compõem um dado espaço, usando uma combinação linear de formas de onda pré-definidas (ou átomos) provenientes de um dicionário dito redundante por conter mais elementos que os necessários para gerar o espaço. Assim, pode-se escolher um subconjunto dos átomos de um dicionário redundante e associar-lhes diferentes pesos a fim de representar um dado sinal. Esse paradigma de representação de sinais é também conhecido por decomposição atômica (MALLAT; ZHANG, 1993). O método mais conhecido dentre os diversos métodos que realizam decomposições atômicas é o *Matching Pursuit* (*MP*) (MALLAT; ZHANG, 1993; JAGGI et al.,). Este método, decompõe um determinado sinal de entrada em uma soma ponderada de formas de onda, conhecidas como átomos, provindas de um dicionário redundante previamente definido.

Uma representação compacta pressupõe alto grau de similaridade entre os elementos do dicionário e os fenômenos presentes no sinal. O uso de dicionários redundantes permite alcançar maior nível de compacidade na representação que o uso de bases ortonormais reais, tais como as wavelets discretas, visto que um sinal pode apresentar estruturas complexas que não são bem representadas utilizando-se somente uma classe de formas de onda. Por sua vez, o dicionário redundante pode ser formado por uma união de bases ortogonais, por transformadas redundantes

(wavelet packet e frames de Gabor) ou por formas de ondas parametrizadas (MALLAT; ZHANG, 1993; UMAPATHY et al., 2005; KLING; ROADS, 2004; JURAFSKY; MARTIN, 2008).

A Figura 1.1 ilustra o esquema de codificação de sinais utilizando decomposições atômicas. O codificador analisa o sinal de forma a encontrar uma boa representação para este, com base em um dicionário $\mathcal{D}$. Nesse dicionário são modelados os fenômenos existentes nos sinais que se deseja aproximar. Em seguida, os coeficientes e os índices dos átomos da representação do sinal são transmitidos/armazenados. O decodificador, com base no mesmo dicionário, sintetiza/reconstrói o sinal a partir dos átomos e pesos provindos do codificador.

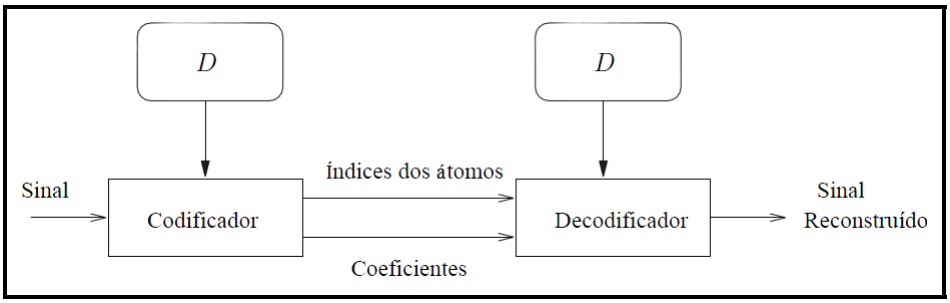


Figura 1.1 – Codificação de sinais usando dicionários redundantes/representações atômicas.

Sabendo-se de que forma os sinais de uma classe são produzidos, podemos utilizar representações atômicas para tentar modelar o sistema físico que gera o sinal. Uma vez que seja obtido um bom modelo e o mesmo seja codificado eficientemente, poderá se obter altas taxas de compressão, reduzindo assim a largura de banda necessária para o processo de transmissão do sinal de voz em questão.

1.2 Estrutura da Dissertação

Com base na introdução dada anteriormente, as seções restantes deste capítulo, destinam-se a discutir algumas das principais técnicas de representação esparsa utilizadas para decomposição atômica. Com base nisso, no capítulo 2, serão apresentados alguns dos dicionários que foram utilizados nas simulações para decomposição de sinais de voz, sendo um deles eleito o melhor ao término deste capítulo. Nos capítulo 3 e 4, serão apresentadas novas técnicas que permitem a classificação de sinais entre voz e silêncio, sem o uso da energia (no caso do capítulo 3) e com o uso, de forma indireta, (no caso do capítulo 4). E por fim, no capítulo 5, será

apresentada a conclusão do trabalho, no qual serão avaliadas as vantagens e desvantagens destas novas técnicas de classificação.

1.3 Representação Atômica

Numa representação atômica, um sinal x é descrito como uma combinação linear de formas de onda selecionadas a partir de um conjunto pré-definido conhecido como dicionário $\mathcal{D}$. Os sinais g_{γ_n} (chamados de átomos ou elementos) que compõem $\mathcal{D}$, são as formas de onda que podem ser utilizadas no processo de decomposição. Algoritmos que obtêm representações atômicas de sinais, escolhem um subconjunto de M elementos g_{γ_n} do dicionário $\mathcal{D}$ e, através de uma soma ponderada destes elementos (utilizando α_n), executa a aproximação de um dado sinal x através da expressão (1.1),

$$x \approx \hat{x} = \sum_{n=1}^{M} \alpha_n g_{\gamma_n}, \qquad g_{\gamma_n} \in \mathcal{D}. \tag{1.1}$$

A decomposição de sinais sobre uma família de átomos que são bem localizados tanto em tempo quanto em frequência, apresenta diferentes propriedades dependendo do átomo escolhido (VERA-CANDEAS et al., 2004). Para extrair informações de sinais complexos, é necessário adaptar a decomposição tempo - frequência com as estruturas particulares de cada sinal. Dessa forma, um conjunto de átomos pode ser gerado aplicando-se um fator de escala (que permite variar a amplitude do átomo), um fator de translação (que permite deslocar o átomo em tempo) e um fator de modulação (que permite variar a largura do átomo) em uma determinada função $g(t)$. Para esta função, deverá ser admitido que a mesma seja real, continuamente diferenciável e que possua norma igual a 1 ($\|g(t)\| = 1$). Para qualquer fator de escala $s > 0$, frequência de modulação ξ e translação u, se denotará $\gamma = (s, u, \xi)$ como sendo a tupla de parâmetros dos átomos. Sendo assim, baseado nos parâmetros de γ e na equação (1.2), pode-se definir a família de átomos que estará composta no dicionário,

$$g_\gamma(t) = \frac{1}{\sqrt{s}}\, g\left(\frac{t-u}{s}\right) e^{i\xi t}. \tag{1.2}$$

Quanto ao parâmetro s, neste trabalho o mesmo será discretizado obedecendo a expressão descrita a seguir,

$$s = 2^j, \qquad j \in \{0, 1, \ldots, \log_2(N)\}, \tag{1.3}$$

nesta expressão, o parâmetro N, equivale ao comprimento do sinal de voz em questão. No caso, uma vez que sinais de voz são processados em janelas de tempo, em virtude do excessivo número de amostras, o valor de N deverá ser igual ao tamanho da janela usada no processamento.

Uma vez que $g(t)$ seja par, $g_\gamma(t)$ estará centralizada na abscissa u, sendo portanto a maior parte da energia concentrada nas vizinhanças de u e seu tamanho proporcional ao fator de escala s. Na Figura 1.2, é apresentado um exemplo de átomo obtido quando utiliza-se uma função de Gabor (cosseno janelado com uma gaussiana). Neste exemplo, utilizou-se um cosseno de frequência 500 Hz (frequência adotada arbitrariamente).

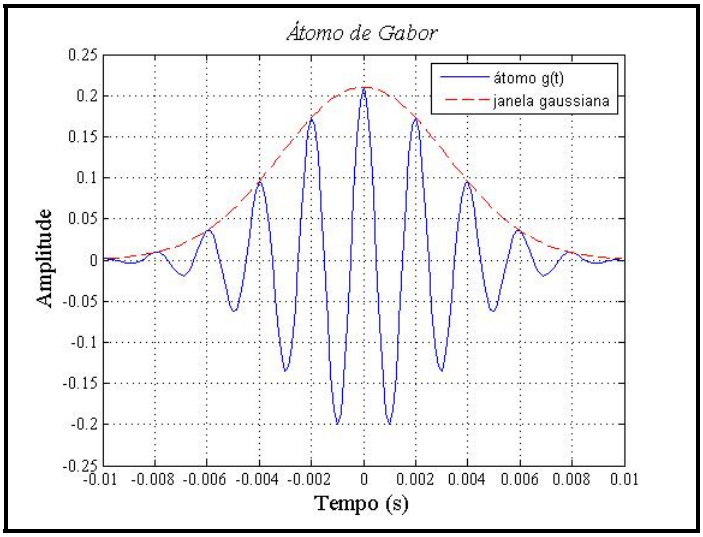


Figura 1.2 – Exemplo de átomo utilizando a função de Gabor.

Para se representar eficientemente qualquer função $f(t)$, deve-se selecionar um conjunto de átomos $(g_{\gamma_n}(t))_{n\in\mathbb{N}}$ onde $\mathbb{N}$ equivale ao conjunto de números naturais e com $\gamma_n = (s_n, u_n, \xi_n)$, tal que $f(t)$ possa ser escrita obedecendo a expressão seguinte,

$$f(t) = \sum_{n=0}^{+\infty} \alpha_n g_{\gamma_n}(t). \tag{1.4}$$

Dependendo dos átomos $g_{\gamma_n}(t)$ que são escolhidos, os coeficientes de expansão α_n fornecem informação explícita sobre algumas das propriedades do sinal a ser sintetizado $f(t)$,

como por exemplo, o tipo do sinal mais correlacionado com ele. Dessa forma, uma vez que uma margem de erro seja previamente definida e que os átomos mais correlacionados sejam utilizados no processo de síntese, a quantidade de informação necessária para reconstrução do sinal de entrada em questão será reduzida.

Um dos maiores problemas quando se faz decomposições adaptativas tempo-frequência, é encontrar um procedimento que expanda uma função em um conjunto de formas de onda, selecionadas apropriadamente, entre um amplo e redundante dicionário (PLUMBLEY et al., 2010; LIU; WU, 2004). A característica de redundância está associada ao fato de que, para um dado sinal $f(t)$, está associado a ele mais de um conjunto de átomos do dicionário, ou seja, é necessário mais de um átomo para expressar a mesma informação. Sendo assim, descreve-se a seguir, um algoritmo conhecido como *Matching Pursuit*, o qual possibilita a decomposição segundo os moldes destacados anteriormente..

Considerando uma função $f(t) \in \mathbf{H}$, onde $\mathbf{H}$ corresponde ao espaço de Hilbert, deseja-se computar a expansão linear de $f(t)$ sobre um conjunto de vetores de átomos selecionados a partir de $\mathcal{D}$, de tal forma que sejam escolhidos os vetores que mais se correlacionem com o sinal de entrada. Isto pode ser feito através de sucessivas aproximações de f com projeções ortogonais nos elementos de $\mathcal{D}$. Considerando o primeiro átomo do dicionário ($g_{\gamma_0} \in \mathcal{D}$), o vetor f pode ser decomposto de acordo com a expressão,

$$f = \langle f, g_{\gamma_0} \rangle g_{\gamma_0} + R_f, \tag{1.5}$$

nesta expressão, a notação $\langle\ ,\ \rangle$ equivale ao produto escalar entre os sinais f e g, enquanto que o parâmetro R_f corresponde ao vetor residual após aproximar f na direção de g_{γ_0}. Ou seja, o resíduo equivale a diferença entre o vetor sinal de entrada e o vetor que foi projetado em um dos átomos, no caso g_{γ_0}.

Sendo assim, uma vez que g_{γ_0} é ortogonal a R_f, pode-se escrever a expressão (1.5) de acordo com a expressão (1.6), ou seja:

$$\|f\|^2 = \|\langle f, g_{\gamma_0} \rangle\|^2 + \|R_f\|^2. \tag{1.6}$$

Para minimizar $\|R_f\|$, precisa-se escolher $g_{\gamma_0} \in \mathcal{D}$ tal que $\|\langle f, g_{\gamma_0}\rangle\|$ seja máxima. Isso significa que, na situação em que o resíduo é 0, o produto escalar obtido deve ser igual a $\|f\|^2$, uma vez que o sinal de entrada é igual a um dos átomos presentes em $\mathcal{D}$. No entanto, na maioria dos casos, só é possível encontrar um vetor g_{γ_0} que tende a se aproximar do sinal de entrada e que não é necessariamente igual, conforme demonstrado na expressão,

$$\|\langle f, g_{\gamma_0}\rangle\| \geq \alpha \ \max \ \|\langle f, g_{\gamma}\rangle\|, \tag{1.7}$$

nesta expressão, o fator α corresponde a um fator de otimalidade e situa-se na faixa entre $0 < \alpha < 1$, sendo 0 na situação em que não há nenhum átomo no dicionário que possua alguma correlação com o sinal de entrada e, 1 no caso hipotético de haver algum átomo que seja idêntico ao sinal de entrada. Nas demais situações, pode assumir qualquer valor, desde que situado na faixa estabelecida.

Com base nisso, pode-se afirmar que o objetivo do algoritmo *MP* é reduzir o resíduo baseando-se nos átomos que estejam mais correlacionados com o sinal de entrada, até que um determinado limiar (definido previamente pelo projetista) ou condição, seja alcançada. Por conseguinte, com base na expressão (1.5), podemos definir a expressão (1.8) utilizada na determinação dos resíduos para cada iteração,

$$R_f^{n+1} = R_f^n - \langle R_f^n, g_{\gamma_n}\rangle g_{\gamma_n}, \tag{1.8}$$

nesta expressão, assume-se para o resíduo inicial (R_f^0) o sinal de entrada ($f(t)$) e, para cada iteração, calcula-se o resíduo de ordem $n + 1$.

Com base nos M resíduos gerados, onde $0 \leq n \leq M - 1$, pode-se afirmar que o sinal sintetizado será composto da soma dos M átomos selecionados (multiplicados por seus respectivos pesos obtidos via produto escalar) com uma pequena parcela de erro, uma vez que foi estipulado um limiar de convergência para o algoritmo. Em outras palavras,

$$f = \sum_{n=0}^{M-1} \langle R_f^n, g_{\gamma_n}\rangle g_{\gamma_n} + erro, \tag{1.9}$$

dessa forma, o vetor f de entrada foi decomposto em uma soma de elementos de um dicionário, que foram escolhidos de forma a minimizar o M-ésimo resíduo, até que uma parcela de erro (previamente determinada) seja alcançada.

Com base no que foi dito anteriormente, define-se a seguir o pseudo-código do algoritmo *MP*. Neste algoritmo, a variável de entrada x corresponde ao sinal de voz que se deseja decompor; o parâmetro R_f^n corresponde ao erro residual definido entre a variável x e os átomos dos dicionários multiplicados por seus respectivos pesos; o parâmetro α corresponde aos pesos que foram obtidos via produto escalar entre os átomos do dicionário (g_{γ_m}) e o resíduo R_f^n; e por fim, o parâmetro $\#\mathcal{D}$ corresponde ao número de átomos do dicionário.

Inicialização das variáveis:

1: $n \leftarrow 0$
2: $R_f^0 \leftarrow x$
3: $erro \leftarrow definido\, pelo\, projetista$

Rotina principal:

1: **Enquanto** $\|R_f^n\| \geq$ *erro* **faça**
2: **Para** $m = 0$ to $\#\mathcal{D} - 1$ **faça**
3: $\alpha(m) \leftarrow \langle\, g_{\gamma_m}, R_f^n \,\rangle$
4: $k \leftarrow indice\, em\, que\, ocorreu\, max\, do\, vetor\, |\alpha|$
5: $R_f^{n+1} \leftarrow R_f^n - \alpha(k) g_{\gamma_k}$
6: $n \leftarrow n + 1$

Algoritmo 1.1 – Matching Pursuit (x)

1.4 Dicionários

Um dicionário completo permite representar qualquer sinal com um erro de aproximação arbitrariamente pequeno. Entretanto, o emprego de um dicionário completo não garante a obtenção de uma representação compacta, uma vez que a função $g(t)$ selecionada para elaboração do dicionário pode não ser a mais adequada para representar o sinal que se deseja analisar via *MP* (BREEN, 2009).

A probabilidade de encontrar um átomo no dicionário possuidor de grande semelhança com o sinal a ser decomposto, aumenta com $\#\mathcal{D}$. Logo, a utilização de dicionários com $\#\mathcal{D}$ grande é desejável em algumas aplicações, possibilitando que o mesmo contenha átomos semelhantes a todas as componentes potenciais dos sinais a serem decompostos.

É desejável que os átomos utilizados na representação de um dado sinal sejam escolhidos de acordo com o mesmo, ou seja, adaptativamente. Por isso, algoritmos que obtêm representações por M termos são chamados de algoritmos de decomposição adaptativa de sinais (CHEN; SAUNDERS, 2001). O uso de um dicionário redundante é um pré-requisito para obter representações adaptativas, pois assim pode-se escolher uma dentre diferentes combinações lineares dos elementos do dicionário para representar um dado sinal. O algoritmo empregado na decomposição influencia a representação por M termos obtida, pois diferentes critérios podem ser usados para selecionar os seus átomos.

Com relação ao tamanho do dicionário, dependendo do número de átomos que seja estipulado, pode-se haver um impacto principalmente nos seguintes aspectos:

a) Na complexidade computacional: em geral, o custo computacional dos algoritmos que obtêm representações atômicas está relacionado a $\#\mathcal{D}$, de forma tal que quanto maior for $\#\mathcal{D}$, maior será, em geral, o custo computacional dos algoritmos;

b) Na memória computacional: dependendo do tamanho de $\#\mathcal{D}$ e do formato com que as amplitudes dos átomos estão sendo armazenadas (por exemplo double), pode requerer uma quantidade excessiva de memória.

1.5 Análise de Complexidade Computacional

Encontrar uma decomposição esparsa linear ou uma aproximação esparsa de um dado sinal é um problema fundamental em muitos domínios tais como compressão e detecção sonora (MALLAT; ZHANG, 1993). Muitos algoritmos foram propostos de forma a obter uma boa aproximação em tempo polinomial. No entanto, desenvolver o projeto de um algoritmo que combine desempenho com erro de aproximação pequeno, ainda é muito desafiador.

Com base nisso, algumas técnicas de representação esparsa foram desenvolvidas ao longo do tempo. Dentre elas podemos destacar: *Matching Pursuit* (MALLAT; ZHANG, 1993), *Orthogonal Matching Pursuit* (REBOLLO-NEIRA; LOWE, 2002) e *Gradient Pursuit* (BLUMENSATH; DAVIES, 2009). A principal diferença existente entre esses algoritmos está na forma em que o resíduo é atualizado a cada iteração. A seguir é apresentado as regras de atualização do resíduo R_f^{n+1} para cada rotina.

1. *MP*: $R_f^{n+1} = R_f^n - \langle R_f^n, g_{\gamma_n} \rangle g_{\gamma_n}$;

2. *OMP*: $R_f^{n+1} = R_f^n - \Phi_n (\Phi_n^T \Phi_n)^{-1} \Phi_n^T R_f^n$;

3. *GP*: $R_f^{n+1} = R_f^n - \frac{\|\Phi_n^T R_f^n\|_2^2}{\|\Phi_n \Phi_n^T R_f^n\|_2^2} \Phi_n \Phi_n^T R_f^n$;

Nas expressões de atualização anteriores, o parâmetro Φ_n corresponde ao conjunto de átomos que possui maior produto escalar com o resíduo ($g_{\gamma_n} = argmax \| \langle R_f^n, g_\gamma \rangle \|$), ou seja, a cada iteração n, o valor de Φ_n deverá ser atualizado de acordo com $\Phi_{n+1} = \Phi_n \cup g_{\gamma_n}$.

O algoritmo *MP* é o mais rápido das rotinas de atualização descritas anteriormente (motivo pelo qual foi utilizado neste trabalho), pois apenas o coeficiente do melhor átomo (de maior correlação) é utilizado e não o conjunto deles, como é o caso das outras duas técnicas. No caso, tanto a técnica *OMP* quanto a *GP* utiliza todos os coeficientes dos átomos selecionados, permitindo que um menor erro de aproximação seja alcançado ao término das iterações, mas com um custo computacional mais elevado.

2 DICIONÁRIOS REDUNDANTES

2.1 Elaboração dos Dicionários

2.1.1 Dicionário senoidal/cossenoidal

Uma vez selecionado o algoritmo de representação esparsa utilizado para análise dos sinais de voz, deve-se determinar a função que será utilizada como base para elaboração do dicionário $\mathcal{D}$. Conforme mencionado anteriormente, adotou-se a técnica *Matching Pursuit* (*MP*) como base para determinação dos coeficientes, pois, apesar dela ser considerada a pioneira de todas as técnicas desenvolvidas ao longo do tempo, ela ainda é considerada a mais rápida em virtude da forma com que o resíduo é atualizado ao longo das iterações.

O primeiro dicionário adotado para análise foi elaborado a partir de um conjunto de funções senoidais e cossenoidais, uma vez que estas são amplamente utilizadas em outros tipos de decomposição, como por exemplo, na Transformada Rápida de Fourier (FFT[1]). Quanto aos parâmetros utilizados para análise, destacam-se: frequência de amostragem igual a 8 kHz ($f_s = 8$ kHz) e janela com tamanho de 160 amostras ($M = 160$), equivalente a 20 ms. Além disso, uma vez que o volume de dados dos sinais de voz utilizados para análise possuem um número elevado de amostras (da ordem de 24k amostras para um sinal de 3 s), estes precisam ser analisados em janelas de tempo, podendo ser com ou sem sobreposição (sendo a segunda utilizada como referência neste trabalho).

Para elaboração do dicionário $\mathcal{D}$, construiu-se uma família de átomos senoidais e cossenoidais (também conhecidos como harmônicas), cuja frequência é um múltiplo inteiro de uma frequência fundamental (f_0). O parâmetro f_0 pode ser obtido conforme demonstrado a seguir,

$$f_0 = \frac{f_s}{M}. \tag{2.1}$$

Uma vez que no domínio da frequência, a frequência máxima é igual a $\frac{f_s}{2}$, subdividiu-se o espectro de acordo com o intervalo $0 \leq n \leq \left\lfloor \frac{f_s}{2f_0} \right\rfloor$, onde $n \in \mathbb{N}$ e equivale a n-ésima

[1] Abreviatura para *Fast Fourier Transform.*

harmônica. No algoritmo 2.1, é apresentado o pseudo-código de elaboração de $\mathcal{D}$ de acordo com as harmônicas mencionadas anteriormente.

Inicialização das variáveis:

1: $f_0 \leftarrow \frac{f_s}{M}$
2: $t \leftarrow \{0, \frac{1}{f_s}, \frac{2}{f_s}, \ldots, \frac{M-1}{f_s}\}$
3: $\mathcal{D} \leftarrow \{\}$

Rotina principal:

1: **Para** $n = 0$ to $\left\lfloor \frac{f_s}{2f_0} \right\rfloor$ **faça**
2: $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \frac{\sin 2\pi n f_0 t}{\|\sin 2\pi n f_0 t\|}$
3: $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \frac{\cos 2\pi n f_0 t}{\|\cos 2\pi n f_0 t\|}$

Algoritmo 2.1 – Dicionário harmônicas (f_s, M)

2.1.2 Dicionário de Gabor

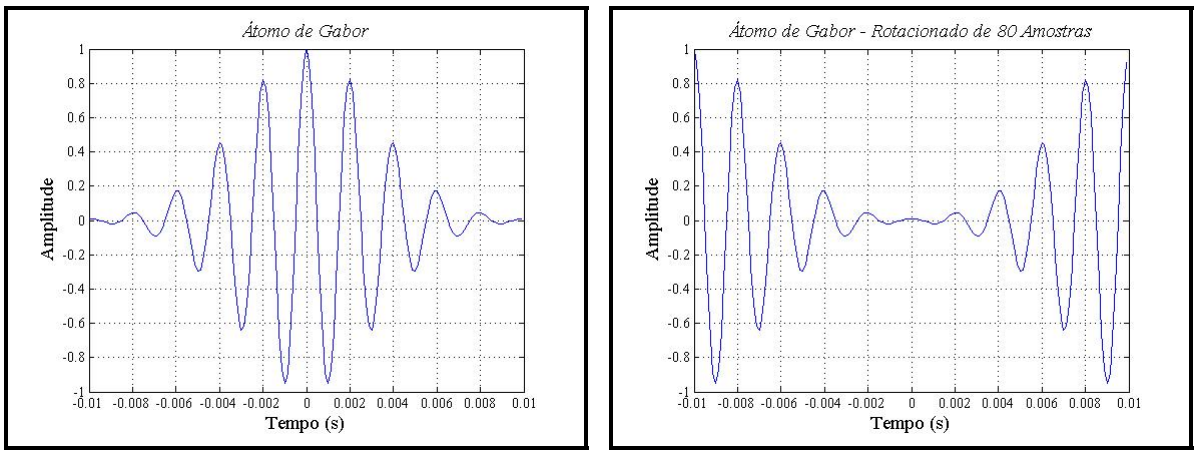
Este dicionário utiliza como função base, um sinal cossenoidal janelado por uma gaussiana. Para a construção do mesmo, utiliza-se o mesmo conjunto de frequências mútiplas de f_0 mencionadas anteriormente, contudo, após ter sido realizado o janelamento, rotaciona-se ou translada-se cada átomo do dicionário no passo de 1 amostra até M-1 amostras, onde M é o tamanho da janela. Na expressão (2.2) é apresentada a função de Gabor utilizada na elaboração de $\mathcal{D}$,

$$g(t)_{Gabor} = 2^{\frac{1}{4}} e^{-\pi\left(\frac{t}{s}\right)^2} \cos 2\pi f t. \tag{2.2}$$

Nela, o parâmetro s é utilizado para variar a largura da gaussiana utilizada no janelamento e o fator de modulação ξ, referente a tupla de parâmetros γ mencionado em (1.2), equivale a frequência angular $(2\pi f)$ da cossenóide utilizada nesta expressão.

É importante ressaltar que no processo de elaboração do dicionário de Gabor, adotou-se o processo de rotação dos átomos e não de translado. Na Figura 2.1a é apresentado um exemplo do átomo de Gabor e na Figura 2.1b o seu equivalente rotacionado de 80 amostras.

Com base na expressão (2.2), destaca-se no algoritmo 2.2, o pseudo-código de elaboração do dicionário de Gabor utilizando a expressão destacada em (2.2).



(a) (b)

Figura 2.1 – Átomo de Gabor: (a) sinal original; (b) equivalente rotacionado de 80 amostras.

Inicialização das variáveis:

$f_0 \leftarrow \frac{f_s}{M}$
$t \leftarrow \{0, \frac{1}{f_s}, \frac{2}{f_s}, \ldots, \frac{M-1}{f_s}\}$
$\mathcal{D} \leftarrow \{\}$

Rotina principal:

Para $n = 0$ to $\left\lfloor \frac{f_s}{2f_0} \right\rfloor$ **faça**
  Para $j = 0$ to $\lfloor \log_2(M) \rfloor$ **faça**
    $s \leftarrow 2^j$
    $g_{\gamma_n} \leftarrow 2^{\frac{1}{4}} e^{-\pi\left(\frac{t}{s}\right)^2} \cos 2\pi n f_0 t$
    $g_{\gamma_n} \leftarrow \frac{g_{\gamma_n}}{\|g_{\gamma_n}\|}$
    $\mathcal{D} \leftarrow \mathcal{D} \cup g_{\gamma_n}$

Rotina de deslocamento temporal:

$aux \leftarrow 1$
$gabor_{rotacionado} \leftarrow \mathcal{D}$
Enquanto $aux \neq M - 1$ **faça**
  $gabor_{rotacionado} \leftarrow rotacionar(gabor_{rotacionado}, 1\ amostra)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup gabor_{rotacionado}$
  $aux \leftarrow aux + 1$

Algoritmo 2.2 – Dicionário de Gabor (f_s, M)

2.1.3 Dicionário DCT

Para elaboração deste dicionário, utilizou-se como átomos a mesma família de cossenóides que seria utilizada em uma Transformada Discreta de Cosseno (DCT). Estes átomos cossenoidais foram criados de acordo com a expressão:

$$g_{DCT} = \cos\left(\frac{\pi n(2k+1)}{2M}\right), \quad 0 \leq n \leq M-1. \tag{2.3}$$

Nesta equação, o parâmetro M equivale ao tamanho da janela e k ao vetor de amostras, definido no intervalo $-M/2 \leq k \leq (M/2)-1$. Sendo assim, uma vez que para cada valor de n define-se um átomo g_{DCT}, constrói-se $\mathcal{D}$ a partir de todos os M átomos existentes em (2.3). Da mesma forma como foi feito para o dicionário de Gabor, todos os átomos de $\mathcal{D}$ devem ser deslocados circularmente de 1 amostra até M-1 amostras. No algoritmo 2.3, é apresentado o pseudo-código de elaboração do dicionário DCT.

Inicialização das variáveis:

$k \leftarrow \{0, 1, 2, \ldots, M-1\}$
$\mathcal{D} \leftarrow \{\}$

Rotina principal:

Para $n = 0$ to $M-1$ **faça**
  $g_{\gamma_n} \leftarrow \cos\left(\frac{\pi n(2k+1)}{2M}\right)$
  $g_{\gamma_n} \leftarrow \frac{g_{\gamma_n}}{\|g_{\gamma_n}\|}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup g_{\gamma_n}$

Rotina de deslocamento temporal:

$aux \leftarrow 1$
$DCT_{rotacionado} \leftarrow \mathcal{D}$
Enquanto $aux \neq M-1$ **faça**
  $DCT_{rotacionado} \leftarrow rotacionar(DCT_{rotacionado}, 1\ amostra)$
  $\mathcal{D} = \mathcal{D} \cup DCT_{rotacionado}$
  $aux \leftarrow aux + 1$

Algoritmo 2.3 – Dicionário DCT (M)

2.1.4 Análise de Desempenho

Para comparar o desempenho, em termos do número de coeficientes necessários para representação entre os três dicionários abordados anteriormente, utilizou-se um sinal de voz amostrado em 8 kHz e com um tamanho de 24k amostras (equivalente a 3 s de voz). Como critério de parada para o algoritmo *MP*, adotou-se um erro equivalente a 0,1% ou -30 dB da energia da janela em análise (a obtenção deste valor será discutido na seção seguinte). Na tabela 2.1, destaca-se o número total de coeficientes necessários para representação do sinal de voz utilizando cada um dos dicionários abordados anteriormente.

Dicionário	Dimensão do Dicionário	Número de Coeficientes	Tempo de Processamento (s)
Senoidal/Cossenoidal	160x162	17548	1.5
DCT	160x25600	14306	300
Gabor	160x102400	8246	120

Tabela 2.1 – Análise de desempenho para erro de -30 dB.

Conforme pode-se perceber pela tabela 2.1, apesar do dicionário de Gabor apresentar a maior dimensão entre todos os dicionários, possibilitou o uso do menor número possível de coeficientes para decompor os sinais de voz em análise. Por outro lado, em relação ao tempo de processamento, nota-se que o dicionário de Gabor apresentou um desempenho intermediário em virtude do maior número de átomos presente no mesmo, fazendo com que este seja um dos poucos pontos negativos deste dicionário.

Dessa forma, uma vez que este trabalho possui como premissa utilizar dicionários que reduzam a largura de banda necessária para sintetizar um determinado sinal de voz via *MP*, se utilizará para todas as análises, deste tópico em diante, o dicionário de Gabor.

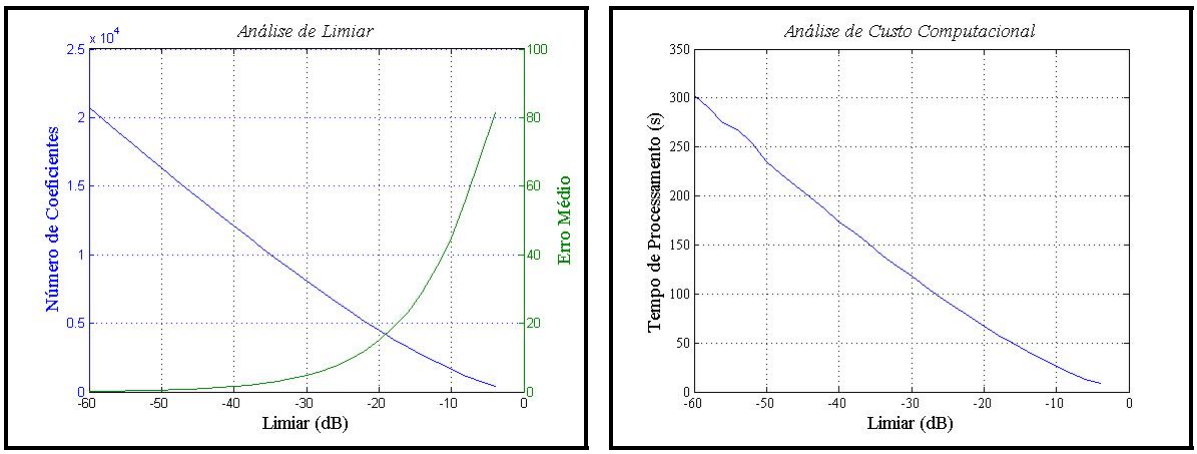
2.2 Análise de parâmetros para simulação

Dependendo do limiar que seja adotado, pode-se obter um erro médio tão pequeno quanto desejado, mas às custas de um tempo de processamento elevado.

Na Figura 2.2a é apresentado não só a variação do número de coeficientes necessários para representação de um sinal de voz, mas também, o erro médio associado a ele, a partir da variação do valor de limiar em dB. Na Figura 2.2b, é apresentado o tempo de processamento requerido no processo de decomposição para os diferentes valores de limiar.

Conforme pode ser observado na Figura 2.2a, à medida que o valor de limiar torna-se mais rigoroso (menor), o número de átomos necessários aumenta exponencialmente, bem como o tempo de processamento requerido. Por outro lado, o erro médio decresce exponencialmente, convergindo para zero para um $limiar = -\infty$. Para o cálculo do erro médio, adotou-se a expressão (2.4) como referência para obtenção do mesmo, sendo ela:

$$erro\_medio = \sum_{i=1}^{i=N} \|x(i) - x'(i)\|. \tag{2.4}$$



(a) (b)

Figura 2.2 – Análise de Limiar: (a) Número de coeficientes e Erro Médio vs Limiar; (b) Tempo de Processamento Requerido.

Nesta expressão, o parâmetro N corresponde ao tamanho do sinal de voz, $x(i)$ equivale à amostra do sinal de voz original e $x'(i)$ equivale à amostra do sinal de voz sintetizado via *MP*.

No entanto, não há uma forma de se determinar o melhor valor de limiar baseando-se nas variáveis discutidas anteriormente, uma vez que não é possível estabelecer uma relação Custo-Benefício ("*Trade-off*") que permita utilizar o menor número de átomos, no menor tempo possível e com menor erro médio. Sendo assim, convencionou-se que o valor de -30 dB atenderia as necessidades de simulação. Audivelmente, há pouca diferença entre os sinais de voz (original e sintetizado), com exceção em palavras que possuam algum tipo de fricativo[2], nas quais se faz necessário o uso de átomos de alta frequência. Conforme mencionado anteriormente, dependendo do trecho de voz em análise (sonoro ou fricativo), se faz necessário um uso menor ou maior de átomos no processo de síntese.

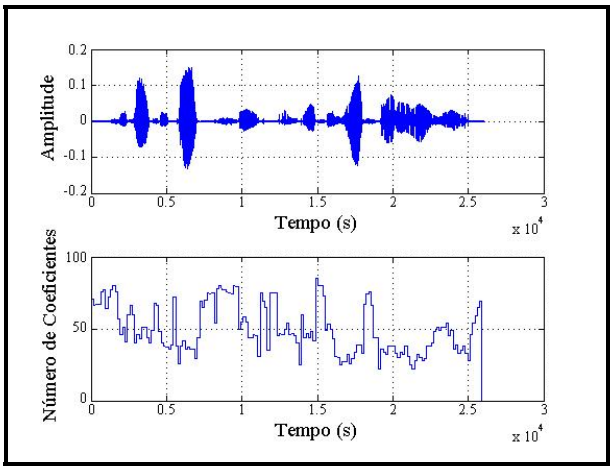


Figura 2.3 – Evolução do número de coeficientes necessários para síntese do sinal de voz.

[2] Sinal de voz similar a um ruído.

Na Figura 2.3, é apresentado um exemplo de sinal de voz e a sua respectiva evolução no número de átomos conforme são analisadas as diversas janelas temporais. Nesta figura, pode-se perceber que em trechos sonoros, o número de átomos necessários no processo de síntese é bem menor (aproximadamente metade) do que em trechos de alta frequência (fricativos) e em trechos de silêncio.

Com relação a energia, dependendo do tipo de sinal de voz em análise, poderá haver uma maior ou menor concentração de energia. Na Figura 2.4a, é apresentado o espectograma de um sinal de voz de aproximadamente 3 s, na Figura 2.4b o seu equivalente sintetizado através de um dicionário de Gabor com limiar de -25 dB e na Figura 2.4c com limiar de -30 dB.

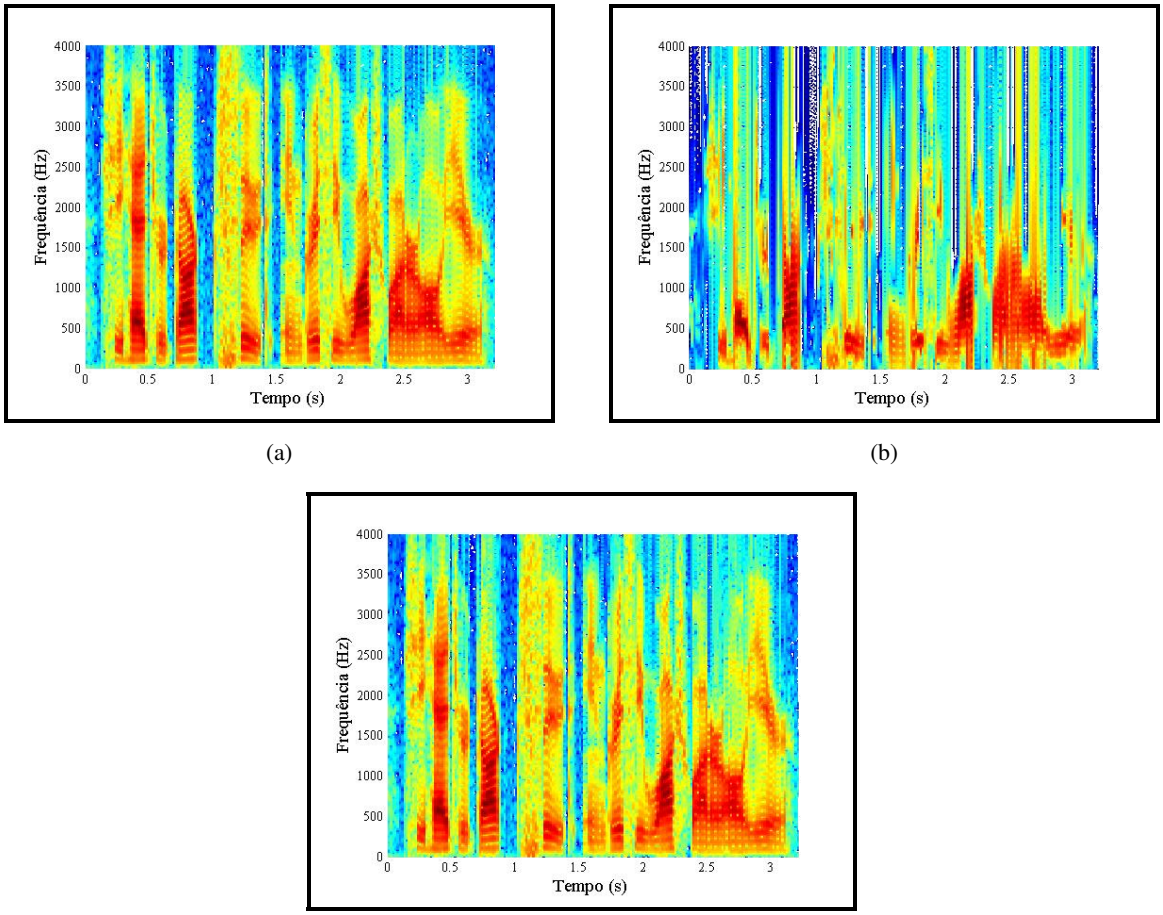


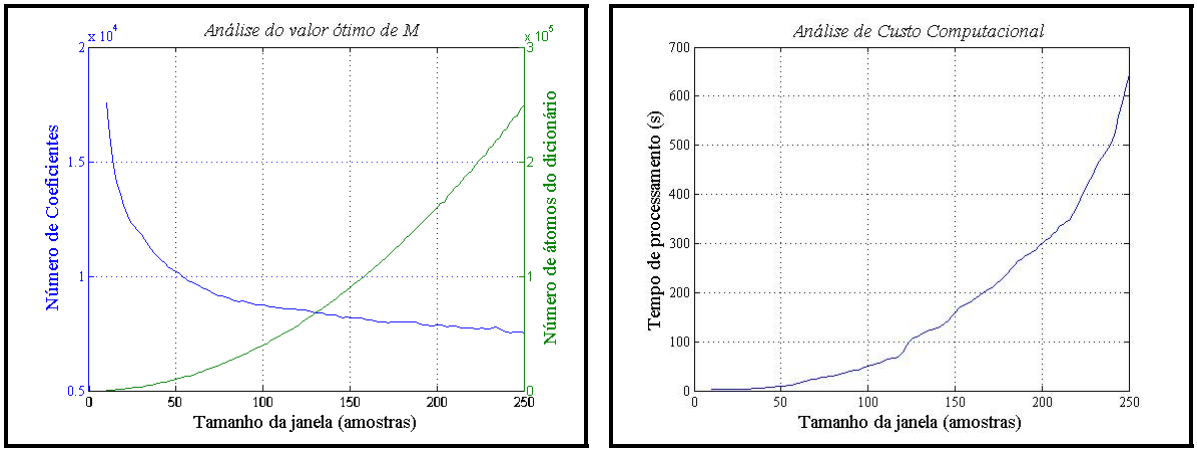
Figura 2.4 – Análise via Espectrograma: (a) Sinal de voz original; (b) Sinal de voz sintetizado com limiar = -25dB; (c) Sinal de voz sintetizado com limiar = -30dB.

Analisando a Figura 2.4, percebe-se que para um limiar de -25 dB, há uma degradação não só na concentração de energia envolvida nos trechos de sonoros (aproximadamente em 0.5s, 1.8s e 2.2s) mas também nas componentes de alta frequência (aproximadamente em 0.2s

e 1.2s). No entanto, no espectrograma da Figura 2.4c, estes problemas não ocorrem de forma tão evidente, comprovando assim, que o uso de -30 dB preserva as componentes fundamentais do sinal original.

Para valores de limiar abaixo de -30 dB, ocorre apenas um refinamento nas componentes espectrais do sinal de voz, tornando-as mais próximas das componentes do sinal original. Contudo, não há necessidade deste nível de detalhamento, uma vez que o mesmo acaba sendo imperceptível ao ouvido humano.

Com relação ao tamanho da janela adotada, na Figura 2.5a é apresentado um gráfico que relaciona o número de coeficientes com o tamanho da janela, sendo esta definida no intervalo entre 10 e 250 amostras. Na Figura 2.5b, é apresentado a evolução do tempo de processamento ao longo deste intervalo.



(a) (b)

Figura 2.5 – Dimensão da janela: (a) Número de coeficientes e Dimensão do dicionário vs Dimensão da janela; (b) Tempo de Processamento Requerido.

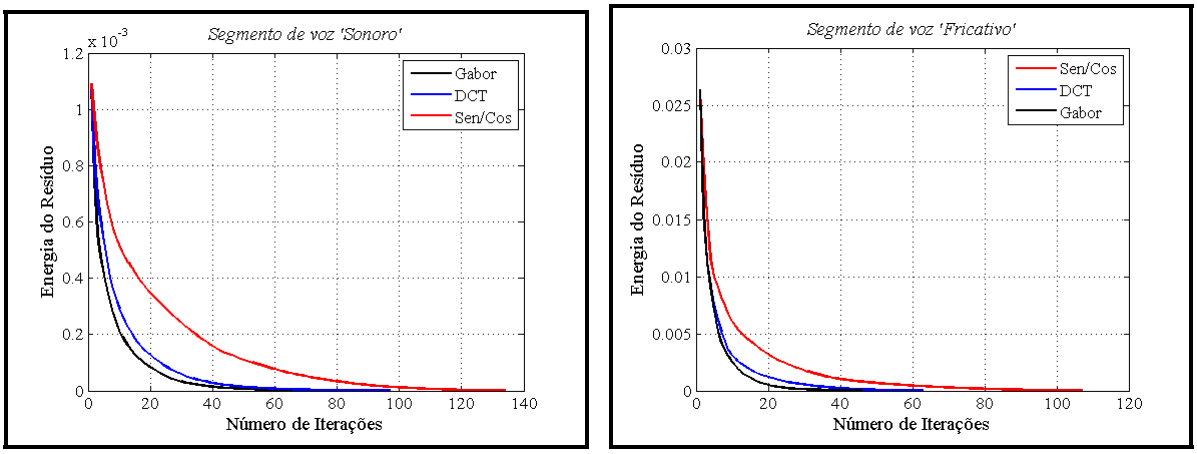
Conforme pode-se perceber na Figura 2.5, à medida que se aumenta o tamanho da janela, o número de coeficientes diminui bem como o tempo de processamento aumenta. Contudo, para janelas de tamanho maior ou igual a 160, o número de coeficientes torna-se praticamente constante, apesar do tamanho do dicionário e do tempo de processamento continuarem aumentando

Sendo assim, uma vez que a partir de 160 amostras o ganho em número de coeficientes é muito menor do que o custo envolvido (tanto para alocação de dicionário quanto para tempo de processamento), convencionou-se em todas as simulações um tamanho de janela igual a 160.

2.3 Análise de Convergência do Resíduo

A convergência do resíduo no algortimo *MP* pode ocorrer em um número maior ou menor de iterações, dependendo do tipo de dicionário utilizado.

Tendo em vista isso, na Figura 2.6a é apresentada a convergência do resíduo para um trecho sonoro (trecho de ressonância onde possivelmente tenha sido pronunciada uma vogal), e, na Figura 2.6b, a convergência do resíduo para um segmento onde tenha sido pronunciado um fricativo.



(a) (b)

Figura 2.6 – Análise de convergência do resíduo: (a) Trecho de voz com vogal; (b) Trecho de voz com fricativo.

Pela figura anterior, percebe-se que a convergência ocorre em um número menor de iterações para o dicionário de Gabor do que para os demais dicionários. Justificando assim, o número reduzido de coeficientes.

3 DETECÇÃO DE VOZ VIA HISTOGRAMAS

3.1 Elaboração de Histogramas

Uma vez que sinais de voz possuem um número elevado de amostras, se faz necessário segmentá-los de forma que as rotinas de processamento sejam feitas para cada trecho e não para o sinal inteiro. Uma vez feito isso, para cada janela de amostras (de forma simplificada para um sinal com alta relação sinal-ruído) podemos classificar cada segmento como sendo voz ou silêncio comparando a energia do trecho em análise com um limiar pré-determinado. Sendo assim, o trecho será caracterizado como voz uma vez que a energia seja superior ao limiar, e silêncio no caso contrário.

Com base nisso, a seguir será discutida uma técnica de detecção que se baseia no uso de representação esparsa e histogramas construídos de forma a caracterizar as diferenças entre essas duas classes de sinais. Esta por sua vez, é fundamentada na probabilidade de ocorrência de um determinado átomo quando o sinal em análise é caracterizado como voz ou como silêncio.

Baseando-se nos algoritmos *MP* e de elaboração do dicionário de Gabor, elaborou-se uma rotina de processamento cuja saída são dois histogramas, um para sinal de voz e outro para silêncio. Esta rotina se utiliza de uma base de dados de treinamento de sinais de voz, onde cada trecho é analisado e caracterizado como voz ou silêncio através do método da energia (utilizado como referência para treinamento do algoritmo de classificação).

Uma vez caracterizado o trecho, realiza-se a rotina de decomposição em átomos do *MP* e em paralelo um processo de contagem do número de ocorrências de cada átomo do dicionário em cada trecho de voz. No algoritmo 3.1, é apresentado o pseudo-código de elaboração dos histogramas.

Em outras palavras, se um dado átomo de índice n ocorreu 10 vezes (no total), por exemplo, ao longo de todos os trechos que foram caracterizados como voz, será contabilizado no histograma de voz o valor 10 para o átomo de índice n (um dicionário típico pode conter 64k átomos). Contudo, uma vez que o mesmo átomo pode aparecer tanto em trechos de voz quanto em silêncio, deve-se contabilizar também o número de ocorrências deste átomo em trechos de

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$erro \leftarrow definido\ pelo\ projetista$
$histo\_voz \leftarrow zeros(1, tamanho(x))$
$histo\_sil \leftarrow zeros(1, tamanho(x))$
$limiar\_energia \leftarrow definido\ pelo\ projetista$

Rotina principal:

Se $\|R_f^n\| > limiar\_energia$ **Então**
  Enquanto $\|R_f^n\| \geq$ *erro* **faça**
    Para $m = 0$ to $\#D - 1$ **faça**
      $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
    $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
    $R_f^{n+1} \leftarrow R_f^n - a(k)g_{\gamma_k}$
    $histo\_voz(k) \leftarrow histo\_voz(k) + 1$
    $n \leftarrow n + 1$
Senão
  Enquanto $\|R_f^n\| \geq$ *erro* **faça**
    Para $m = 0$ to $\#D - 1$ **faça**
      $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
    $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
    $R_f^{n+1} \leftarrow R_f^n - a(k)g_{\gamma_k}$
    $histo\_sil(k) \leftarrow histo\_sil(k) + 1$
    $n \leftarrow n + 1$

Algoritmo 3.1 – Histogramas (x ∈ base de dados de treinamento)

silêncio. No entanto, este novo valor deverá ser posicionado no histograma de silêncio e não no de voz.

3.2 Detecção de Voz via Votação

Uma vez obtidos ambos os histogramas, precisa-se normalizá-los de forma que se obtenha uma densidade de probabilidade cuja soma dos valores seja igual a 1. Para isso, basta dividirmos cada amostra do histograma de voz pela soma de todas amostras deste histograma. Em seguida, realiza-se a mesma operação para o histograma de silêncio.

A partir disso, pode-se executar a rotina de detecção de voz via votação. Esta rotina por sua vez, utiliza-se de um sinal de voz de uma base de dados de teste, e o analisa segundo a probabilidade dos átomos utilizados pertencerem a classe voz ou silêncio. Ao final do processo, é estabelecido um processo de contagem do número de átomos pertencentes a classe voz e silêncio que foram necessários para sintetizar o sinal de voz em questão. Caso tenha sido

necessário um número maior de átomos da classe voz do que da classe silêncio, o sinal em questão será classificado como voz. Na situação contrária, será classificado como silêncio. Em outras palavras, se um dado átomo de índice n possui uma probabilidade de 0,15 no histograma de voz e de 0,12 no histograma de silêncio, o contador de átomos de voz será acrescido de 1 unidade. Na situação contrária de probabilidade, acrescenta-se 1 no contador de átomos de silêncio. Ao final do processo, compara-se ambos os contadores e classifica-se o trecho como sendo voz ou silêncio, dependendo do contador que detenha o maior valor. No algoritmo 3.2, é apresentado o pseudo-código de detecção de voz via votação.

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$erro \leftarrow definido\ pelo\ projetista$
$cont\_voz \leftarrow 0$
$cont\_sil \leftarrow 0$
$histo\_voz \leftarrow histo\_voz / \sum(histo\_voz)$
$histo\_sil \leftarrow histo\_sil / \sum(histo\_sil)$

Rotina principal:

Enquanto $\|R_f^n\| \geq$ *erro* **faça**
  Para $m = 0$ to $\#D - 1$ **faça**
    $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
  $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
  $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
  Se $histo\_voz(k) > histo\_sil(k)$ **Então**
    $cont\_voz \leftarrow cont\_voz + 1$
  Senão
    $cont\_sil \leftarrow cont\_sil + 1$
  $n \leftarrow n + 1$
Se $cont\_voz > cont\_sil$ **Então**
  $audio \leftarrow voz$
Senão
  $audio \leftarrow silencio$

Algoritmo 3.2 – Detecção via votação (x ∈ base de dados de teste)

3.3 Detecção de Voz via Perplexidade

Além do método de detecção via votação, pode-se utilizar o conceito de perplexidade (JURAFSKY; MARTIN, 2008) para classificação entre voz e silêncio. Este método baseia-se no inverso do produto das probabilidades de ocorrência de um determinado átomo para classificar o

trecho como sendo voz ou silêncio. Na equação (3.1) é apresentada a expressão de perplexidade utilizada no processo de classificação, sendo ela:

$$PP = \prod_{n=1}^{N} \frac{1}{\sqrt[N]{P_n}}. \tag{3.1}$$

Nesta expressão, o parâmetro PP corresponde a perplexidade e P_n corresponde a probabilidade de ocorrência de um determinado átomo em uma dada classe, extraída dos histogramas construídos previamente.

Uma vez obtida a perplexidade para os átomos de voz e para os átomos de silêncio, compara-se ambos os valores e seleciona-se aquele que possuir menor valor para classificação, isto é, se a perplexidade referente a voz for menor do que a de silêncio, o trecho será classificado como voz. Na situação contrária, o trecho será classificado como silêncio. No algoritmo 3.3 é apresentado o pseudo-código de detecção via perplexidade.

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$n\_atomos \leftarrow definido\ pelo\ projetista$
$PP\_voz \leftarrow 1$
$PP\_sil \leftarrow 1$
$histo\_voz \leftarrow histo\_voz / \sum(histo\_voz)$
$histo\_sil \leftarrow histo\_sil / \sum(histo\_sil)$

Rotina principal:

Enquanto $n < n\_atomos$ **faça**
  Para $m = 0$ to $\#D - 1$ **faça**
    $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
  $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
  $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
  $PP\_voz \leftarrow PP\_voz * hist\_voz(k)$
  $PP\_sil \leftarrow PP\_sil * hist\_sil(k)$
  $n \leftarrow n + 1$
Se $(PP\_voz)^{-1/n\_atomos} < (PP\_sil)^{-1/n\_atomos}$ **Então**
  $audio \leftarrow voz$
Senão
  $audio \leftarrow silencio$

Algoritmo 3.3 – Detecção via perplexidade (x ∈ base de dados de teste)

Neste algoritmo, os valores de perplexidade são armazenados nas variáveis PP_voz e PP_sil e comparados no final da rotina principal. Vale ressaltar que a classificação utilizada não é baseada somente em qual átomo possui um valor maior nos histogramas de voz ou si-

lêncio, como é o caso do algoritmo 3.2, mas utiliza todas as probabilidades dos histogramas envolvidos nos átomos selecionados (voz e silêncio) para obtenção das respectivas perplexidades e comparação das mesmas.

3.4 Análise e Comparação dos Métodos de Detecção

Para a realização das simulações, utilizou-se o *software* MATLAB em uma plataforma *Windows 7* de 64 bits. Nestas simulações, utilizou-se uma base de dados de treinamento equivalente a 2 horas de sinais de voz (amostrados em 8 kHz) para elaboração dos histogramas.

Em todo processamento, padronizou-se a frequência de amostragem (f_s) em 8 kHz, tamanho do segmento de voz, para análise, em 20 ms ou 160 amostras e erro de -35 dB. Com relação ao dicionário, utilizou-se como base a função Gabor e um tamanho equivalente a 64k átomos. Na Figura 3.1a e 3.1b, são apresentados os histogramas de voz e de silêncio (sem normalização), que foram obtidos utilizando o algoritmo 3.1 abordado anteriormente.

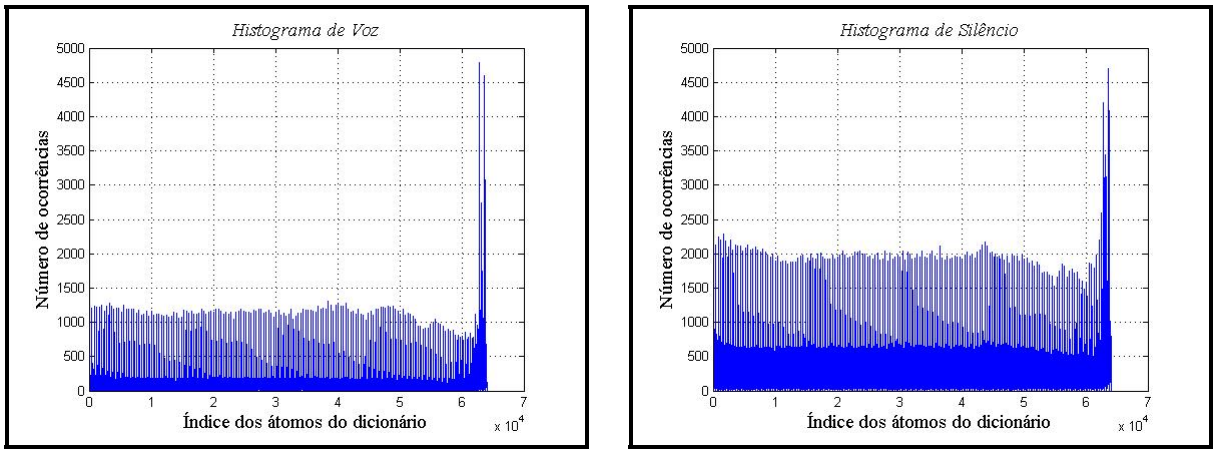


Figura 3.1 – (a) Histograma de Voz; (b) Histograma de Silêncio.

Com estes dois histogramas e com uma base de dados de teste com aproximadamente 15 min de sinais de voz (não se utilizou uma base de dados maior, devido ao excessivo tempo de processamento do MATLAB), executou-se a rotina de processamento abordada no algoritmo 3.2 (a Figura 3.2 apresenta o resultado da classificação para um dos sinais de voz utilizados). No entanto, apesar dos sinais utilizados tanto na base de dados de treinamento quanto na de teste apresentarem uma alta relação sinal-ruído, alguns erros de detecção ocorreram ao longo do sinal de voz (principalmente em trechos de silêncio). Estes erros podem ser observados na Figura 3.2.

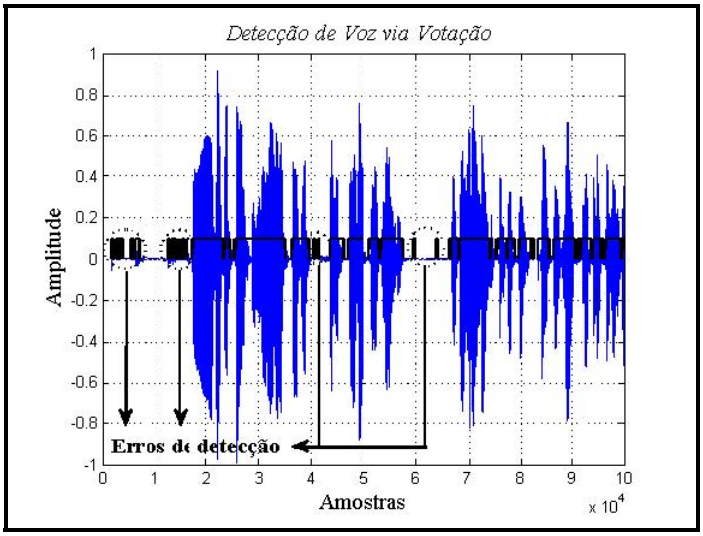


Figura 3.2 – Detecção de voz via votação.

Estes erros de detecção ocorreram em virtude do erro definido previamente (-35 dB) não corresponder ao limiar ótimo de operação. Dessa forma, comparando-se a classificação via votação com a classificação pelo limiar de energia (sendo a última usada como referência), obtêm-se a taxa de acerto de detecção para o sinal de voz em questão. Contudo, uma vez que para diferentes valores de erro obtêm-se diferentes taxas de acerto, decidiu-se determinar qual é o ponto ótimo de operação que maximiza esta taxa. Na Figura 3.3, é apresentada a curva que define o comportamento de variação entre a taxa de acerto e o erro definido.

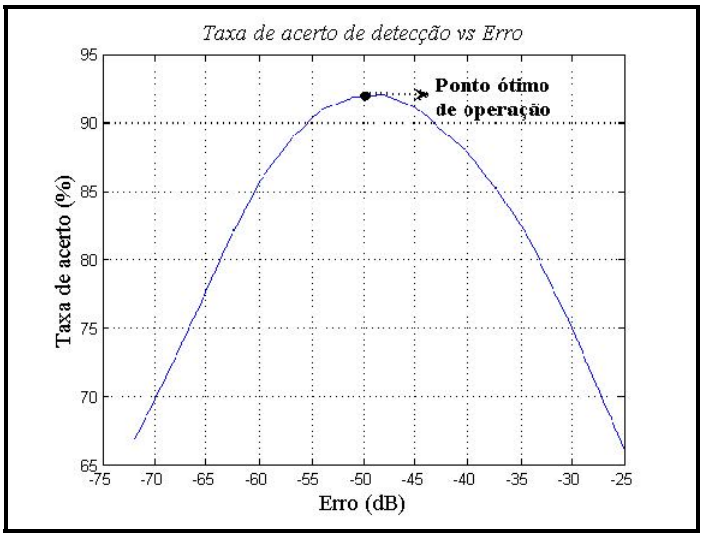


Figura 3.3 – Ponto ótimo de operação para o método de detecção via votação.

Percebe-se portanto, que o ponto ótimo de operação está definido para um erro de -50 dB, o qual resulta em uma taxa de acerto de aproximadamente 93%. Por outro lado, uma

vez que ambos os parâmetros estão fortemente correlacionados com o tipo de dicionário que está sendo utilizado, observa-se que, quanto maior for o nível de redundância e completude do dicionário, maior será a taxa de acerto e por conseguinte menor será a necessidade de se utilizar limiares de conversão mais rigorosos/intensos (quanto mais rigoroso o limiar de conversão, maior será a quantidade de átomos necessários para sintetizar um sinal mais próximo do sinal original).

É interessante observar que para valores de erro abaixo de -50 dB, a taxa de acerto começa a decair. Este fato indica, que o dicionário utilizado (limitado em 64k por motivos de processamento e memória) não é completo o suficiente para valores de erro abaixo de -50 dB, resultando em uso de átomos de baixa correlação (e por conseguinte baixa probabilidade) como forma de aumentar a precisão da síntese do algoritmo *MP*, e melhorar o processo de classificação entre voz e silêncio. Contudo, o uso destes átomos ao longo das iterações, resulta em um aumento na taxa de erro de detecção conforme o valor de erro é reduzido.

Uma vez que a quantidade de átomos aumenta conforme o limiar de conversão se torna mais rigoroso, optou-se por utilizar o conceito da perplexidade no processo de classificação entre voz e silêncio. Para isso, limitou-se o número de átomos por janela e observou-se a taxa de acerto obtida. Na Figura 3.4 é apresentada a curva obtida para valores entre 10 e 80 átomos por janela.

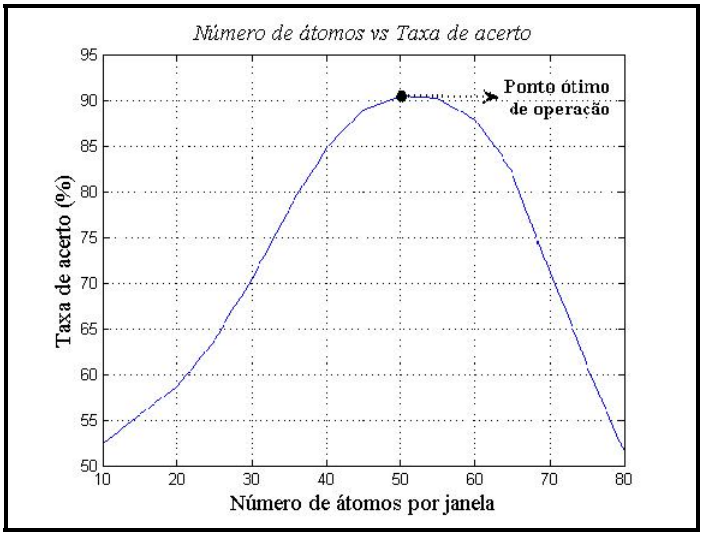


Figura 3.4 – Ponto ótimo de operação para o método de detecção via perplexidade.

Nesta figura, observa-se que o ponto ótimo de operação é obtido com 50 átomos por janela, resultando em uma taxa de acerto de aproximadamente 91%. Esta taxa é inferior à obtida anteriormente (93%), por outro lado, a quantidade de átomos utilizada é 47% menor. Enquanto que para um limiar de -50 dB foram necessários aproximadamente 59k átomos, pelo método da perplexidade foram necessários 31k átomos, o que permitiu uma redução na quantidade de átomos necessários no processo de classificação entre voz e silêncio.

4 ANÁLISE COM SINAIS RUIDOSOS

4.1 Introdução

Nos capítulos anteriores, avaliou-se o desempenho de duas técnicas de classificação de sinais de voz, denominadas votação e perplexidade. Em ambas as situações, foram utilizados para análise sinais de voz de alta relação sinal-ruído (SNR[1]), permitindo assim verificar o comportamento destes dois algoritmos sob condições ideais de operação.

Contudo, uma vez que em situações reais de operação há presença de ruído, adicionou-se aos sinais de voz pertencentes a base de dados de teste ruídos de característica específica, como por exemplo, branco ou pink. Dessa forma, as seções seguintes destinam-se não só para avaliação destes dois métodos de acordo com esta nova base de dados, mas também para discutir a influência dos pesos dos átomos no processo de classificação.

4.2 Modelamento com Pesos

Os métodos de classificação abordados no capítulo 3, utilizam como base de detecção, as probabilidades provindas dos histogramas, tanto de voz quanto de silêncio, que foram obtidos de acordo com o número de ocorrências de cada átomo do dicionário $\mathcal{D}$. Com isso, nenhuma informação referente a energia é considerada no processo de classificação. Contudo, é possível refinar ambos os algoritmos se for acrescentada informação relativa aos pesos atribuídos a cada átomo utilizado ao longo das iterações. Dessa forma, os parágrafos seguintes, destinam-se a discussão do modelamento matemático desenvolvido para associar a cada peso uma probabilidade equivalente, e com isso, como tratar esta informação de forma conjunta com a probabilidade dos histogramas e resultar em uma detecção mais precisa.

Ao longo do processo de decomposição do algoritmo *Matching Pursuit*, o mesmo átomo pode ser utilizado mais de uma vez em virtude das diferentes janelas que são processadas a cada iteração. Com isso, da mesma forma que o sinal de cada trecho pode pertencer as classes voz ou silêncio, as amplitudes associadas a cada átomo também podem ser discriminadas analogamente. De forma a possibilitar esta diferenciação, desenvolveu-se um algoritmo de trei-

[1] Abreviatura para *Signal to Noise Rate*

namento que permite distinguir os diferentes pesos nas duas classes em questão, mas usando como referência o algoritmo de detecção via energia. No caso, de forma similar ao método via histogramas, se um dado sinal é caracterizado como voz ou silêncio pelo método da energia, armazena-se todos os pesos dos átomos utilizados nas iterações nas variáveis denominadas a_voz e a_sil, respectivamente.

Uma vez obtido os pesos (a_voz e a_sil) de cada classe, define-se uma média e uma variância para cada classe e átomo de índice k pertencente a $\mathcal{D}$. Contudo, uma vez que para cada índice há um vetor de pesos associado a ele, define-se um índice z capaz de identificar cada elemento existente neste vetor. As equações (4.1) e (4.2) apresentam as expressões de obtenção da $media(k)$ e da $variancia(k)$ (respectivamente), dado que tenha se utilizado m vezes o mesmo átomo de índice k. No entanto, de forma a generalizar e simplificar estas 2 equações, considerou-se as variáveis a_voz e a_sil como sendo simplesmente $a(k, z)$, evitando assim a necessidade de se desmembrar estas equações para cada classe de sinal.

$$media(k) = \frac{1}{m}\sum_{z=1}^{m} a(k,z), \quad 1 \leq k \leq \#\mathcal{D} \tag{4.1}$$

$$variancia(k) = \frac{1}{m}\sum_{z=1}^{m}(a(k,z))^2 - \left(\frac{1}{m}\sum_{z=1}^{m} a(k,z)\right)^2, \quad 1 \leq k \leq \#\mathcal{D} \tag{4.2}$$

De posse dos valores de média e variância para cada classe, aproximou-se cada átomo de $\mathcal{D}$ como sendo uma distribuição normal, possibilitando assim extrair um valor de probabilidade para cada peso $a(k)$ utilizado na fase de teste. A equação (4.3) define a expressão de obtenção da densidade de probabilidade de um dado peso, atribuída a variável $prob\_peso(k)$, que deve ser obtida tanto para o conjunto de valores relacionados a média e variância dos sinais classificados como voz, como também, para os sinais caracterizados como silêncio. Nesta equação, a variável z foi omitida em virtude dos pesos utilizados não serem os mesmos aos obtidos na fase de treinamento, mas sim, aos novos pesos relacionados aos sinais pertencentes à base de dados de teste. Além disso, novamente por questões de simplicidade, esta equação não está vinculada a nenhuma classe de sinal de voz.

$$prob\_peso(k) = \frac{1}{\sqrt{2\pi variancia(k)}}\, e^{-\frac{1}{2}\frac{(a(k)-media(k))^2}{variancia(k)}}, \quad 1 \leq k \leq \#\mathcal{D} \tag{4.3}$$

Com isso, apresenta-se a seguir, no algoritmo 4.1, o pseudo-código de treinamento para obtenção dos pesos pertencentes as classes voz e silêncio.

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$erro \leftarrow definido\ pelo\ projetista$
$a\_voz \leftarrow \{\}$
$a\_sil \leftarrow \{\}$
$limiar\_energia \leftarrow definido\ pelo\ projetista$

Rotina principal:

Se $\|R_f^n\| > limiar\_energia$ **Então**
  Enquanto $\|R_f^n\| \geq$ *erro* **faça**
    Para $m = 0$ to $\#D - 1$ **faça**
      $a(m) \leftarrow \langle\, g_{\gamma_m}, R_f^n \,\rangle$
    $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
    $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
    $a\_voz(k) \leftarrow a\_voz(k) \cup a(k)$
    $n \leftarrow n + 1$
Senão
  Enquanto $\|R_f^n\| \leq$ *erro* **faça**
    Para $m = 0$ to $\#D - 1$ **faça**
      $a(m) \leftarrow \langle\, g_{\gamma_m}, R_f^n \,\rangle$
    $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
    $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
    $a\_sil(k) \leftarrow a\_sil(k) \cup a(k)$
    $n \leftarrow n + 1$

Algoritmo 4.1 – Pesos (x $\in$ base de dados de treinamento)

Uma vez obtida as probabilidades de acordo com (4.3), alterou-se os algoritmos de votação e perplexidade apresentados no capítulo 3 de forma a não utilizar as probabilidades provindas dos histogramas, mas sim, da distribuição gaussiana obtida segundo equação (4.3). O algoritmo 4.2 apresenta o pseudo-código de classificação, via votação, o qual utiliza somente os pesos dos átomos como principal fonte de informação no processo de detecção. Vale ressaltar que atribuiu-se uma probabilidade mínima, aos átomos de $\mathcal{D}$, que apresentaram variância igual a zero, satisfazendo a situação em que a probabilidade de um átomo que não foi ativado no treinamento retorne um valor mínimo de probabilidade na fase de teste.

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$erro \leftarrow definido\ pelo\ projetista$
$cont\_voz \leftarrow 0$
$cont\_sil \leftarrow 0$

Rotina principal:

Enquanto $\|R_f^n\| \geq$ *erro* **faça**
  Para $m = 0$ to $\#D - 1$ **faça**
    $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
  $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
  $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
  $prob\_peso\_voz = \frac{1}{\sqrt{2\pi variancia\_voz(k)}}\ e^{-\frac{1}{2}\frac{(a(k)-media\_voz(k))^2}{variancia\_voz(k)}}$
  $prob\_peso\_sil = \frac{1}{\sqrt{2\pi variancia\_sil(k)}}\ e^{-\frac{1}{2}\frac{(a(k)-media\_sil(k))^2}{variancia\_sil(k)}}$
  Se $prob\_peso\_voz > prob\_peso\_sil$ **Então**
    $cont\_voz \leftarrow cont\_voz + 1$
  Senão
    $cont\_sil \leftarrow cont\_sil + 1$
  $n \leftarrow n + 1$
Se $cont\_voz > cont\_sil$ **Então**
  $audio \leftarrow voz$
Senão
  $audio \leftarrow silencio$

Algoritmo 4.2 – Votação via Pesos (x ∈ base de dados de teste)

No algoritmo 4.3, é apresentado o pseudo-código de classificação via perplexidade utilizando somente os pesos como principal fonte de informação no processo de detecção. Nesse algoritmo, uma vez obtido o peso do átomo de maior correlação, calcula-se a probabilidade deste peso ser classificado como voz ou silêncio de acordo com a equação (4.3). Para cada iteração realizada na rotina principal, armazena-se um acúmulo de produtório das probabilidades classificadas tanto como voz quanto silêncio, nas variáveis PP_voz e PP_sil, respectivamente. Ao final, a classificação será efetuada de acordo com a variável que possuir o menor valor, pois, por definição, quanto menor o valor de perplexidade maior a probabilidade de ocorrência.

Os algoritmos de detecção, via pesos, apresentados anteriormente podem ser utilizados de forma conjunta com os algoritmos de detecção, via histogramas, abordados no capítulo 3. Dessa forma, o desempenho da classificação torna-se superior ao obtido executando-se cada método isoladamente.

Inicialização das variáveis:

1: $n \leftarrow 0$
2: $R_f^n \leftarrow x$
3: $n\_atomos \leftarrow definido\ pelo\ projetista$
4: $PP\_voz \leftarrow 1$
5: $PP\_sil \leftarrow 1$

Rotina principal:

1: **Enquanto** $n < n\_atomos$ **faça**
2: **Para** $m = 0$ to $\#D - 1$ **faça**
3: $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
4: $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
5: $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
6: $prob\_peso\_voz = \frac{1}{\sqrt{2\pi variancia\_voz(k)}}\ e^{-\frac{1}{2}\frac{(a(k)-media\_voz(k))^2}{variancia\_voz(k)}}$
7: $prob\_peso\_sil = \frac{1}{\sqrt{2\pi variancia\_sil(k)}}\ e^{-\frac{1}{2}\frac{(a(k)-media\_sil(k))^2}{variancia\_sil(k)}}$
8: $PP\_voz \leftarrow PP\_voz * prob\_peso\_voz$
9: $PP\_sil \leftarrow PP\_sil * prob\_peso\_sil$
10: $n \leftarrow n + 1$
11: **Se** $(PP\_voz)^{-1/n\_atomos} < (PP\_sil)^{-1/n\_atomos}$ **Então**
12: $audio \leftarrow voz$
13: **Senão**
14: $audio \leftarrow silencio$

Algoritmo 4.3 – Perplexidade via Pesos (x ∈ base de dados de teste)

No algoritmo 4.4 é apresentado o pseudo-código de classificação via votação que utiliza, de forma conjunta, as informações de probabilidade provindas dos histogramas e dos pesos.

Conforme apresentado no algoritmo 4.4, uma vez que as informações de probabilidades obtidas através dos pesos e dos histogramas de ocorrências podem ser adotadas como sendo independentes, utilizou-se o produto de ambas às probabilidades para classificar o sinal de teste como sendo voz ou silêncio. De posse desta metodologia, implementou-se no método via perplexidade, o mesmo conceito de eventos independentes no processo de detecção. Contudo, visto que a perplexidade se utiliza do acúmulo do produtório das probabilidades como base de classificação, se fez necessário alterar para uma soma logarítmica de forma a evitar valores pequenos ao longo do processamento, melhorando assim a precisão das comparações das probabilidades entre voz e silêncio.

No algoritmo 4.5 é apresentado o pseudo-código de classificação conjunta via perplexidade.

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$erro \leftarrow definido\ pelo\ projetista$
$cont\_voz \leftarrow 0$
$cont\_sil \leftarrow 0$

Rotina principal:

Enquanto $\|R_f^n\| \geq$ *erro* **faça**
  Para $m = 0$ to $\#D - 1$ **faça**
    $a(m) \leftarrow \langle\, g_{\gamma_m}, R_f^n \,\rangle$
  $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
  $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
  $prob\_peso\_voz = \frac{1}{\sqrt{2\pi variancia\_voz(k)}}\, e^{-\frac{1}{2}\frac{(a(k)-media\_voz(k))^2}{variancia\_voz(k)}}$
  $prob\_peso\_sil = \frac{1}{\sqrt{2\pi variancia\_sil(k)}}\, e^{-\frac{1}{2}\frac{(a(k)-media\_sil(k))^2}{variancia\_sil(k)}}$
  Se $prob\_peso\_voz * hist\_voz(k) \geq prob\_peso\_sil * hist\_sil(k)$ **Então**
    $cont\_voz \leftarrow cont\_voz + 1$
  Senão
    $cont\_sil \leftarrow cont\_sil + 1$
  $n \leftarrow n + 1$
Se $cont\_voz \geq cont\_sil$ **Então**
  $audio \leftarrow voz$
Senão
  $audio \leftarrow silencio$

Algoritmo 4.4 – Votação Conjunta (x ∈ base de dados de teste)

Nesse algoritmo, não limitou-se o número de iterações realizadas na rotina principal, a um número de átomos específicos conforme algoritmo 4.3, pois, o resultado da classificação para um erro pré-determinado, apresentou uma taxa de acerto superior ao que seria obtido se fosse utilizado um número fixo de átomos. Entretanto, uma vez que a quantidade de átomos utilizados, para classificação, torna-se superior ao que seria obtido via 4.3, o tempo de processamento do algoritmo aumenta proporcionalmente, inviabilizando esta técnica se comparada ao método via votação.

De posse desses algoritmos, na seção seguinte serão apresentados os resultados de classificação para sinais de voz que estejam sobre interferência de ruídos do tipo branco ou pink (obtidos segundo referência NOISEX-92... ()), sendo as amplitudes dos mesmos ajustadas de acordo com os valores de SNR previamente definidos.

Inicialização das variáveis:

$n \leftarrow 0$
$R_f^n \leftarrow x$
$erro \leftarrow definido\ pelo\ projetista$
$PP\_peso\_voz \leftarrow 0$
$PP\_peso\_sil \leftarrow 0$
$PP\_histo\_voz \leftarrow 0$
$PP\_histo\_sil \leftarrow 0$

Rotina principal:

Enquanto $\|R_f^n\| \geq$ *erro* **faça**
  Para $m = 0$ to $\#D - 1$ **faça**
    $a(m) \leftarrow \langle\ g_{\gamma_m}, R_f^n\ \rangle$
  $k \leftarrow indice\ em\ que\ ocorreu\ max\ do\ vetor\ \|a\|$
  $R_f^{n+1} \leftarrow R_f^n - a(k) g_{\gamma_k}$
  $prob\_peso\_voz = \frac{1}{\sqrt{2\pi variancia\_voz(k)}}\ e^{-\frac{1}{2}\frac{(a(k)-media\_voz(k))^2}{variancia\_voz(k)}}$
  $prob\_peso\_sil = \frac{1}{\sqrt{2\pi variancia\_sil(k)}}\ e^{-\frac{1}{2}\frac{(a(k)-media\_sil(k))^2}{variancia\_sil(k)}}$
  $PP\_peso\_voz \leftarrow PP\_peso\_voz + \log_{10} prob\_peso\_voz(k)$
  $PP\_peso\_sil \leftarrow PP\_peso\_sil + \log_{10} prob\_peso\_sil(k)$
  $PP\_histo\_voz \leftarrow PP\_histo\_voz + \log_{10} hist\_voz(k)$
  $PP\_histo\_sil \leftarrow PP\_histo\_sil + \log_{10} hist\_sil(k)$
  $n \leftarrow n + 1$
Se $((PP\_peso\_voz) * (PP\_histo\_voz))^{-1/n} \leq ((PP\_peso\_sil) * (PP\_histo\_sil))^{-1/n}$ **Então**
  $audio \leftarrow voz$
Senão
  $audio \leftarrow silencio$

Algoritmo 4.5 – Perplexidade Conjunta (x ∈ base de dados de teste)

4.3 Resultados

Conforme apresentado na seção anterior, os algoritmos de classificação foram executados segundo os seguintes parâmetros de simulação: f_s = 8 kHz (frequência de amostragem), $erro$ = -50 dB (erro de síntese) e M = 160 (tamanho da janela). Em um primeiro momento, observou-se o comportamento dos algoritmos de classificação adicionando-se ruído branco (NOISEX-92) aos sinais de voz pertencentes à base de dados de teste. Uma vez definidos os valores de SNR a ser testados, obteve-se, via equação (4.4), um fator multiplicativo (β) para cada elemento deste vetor. Este fator, ajusta a amplitude do ruído para que ao ser adicionado o sinal de voz, a SNR estabelecida seja a mesma que foi definida previamente. Nesta equação, o parâmetro E_S equivale a energia do sinal de voz e E_N equivale a energia original do ruído.

$$\beta = \sqrt{10^{-\frac{SNR}{10}} * \left(\frac{E_S}{E_N}\right)} \tag{4.4}$$

Na Figura 4.1 são apresentados os resultados de detecção, para uma variação de SNR de 1 a 35 dB, quando comparados ao método de classificação via energia.

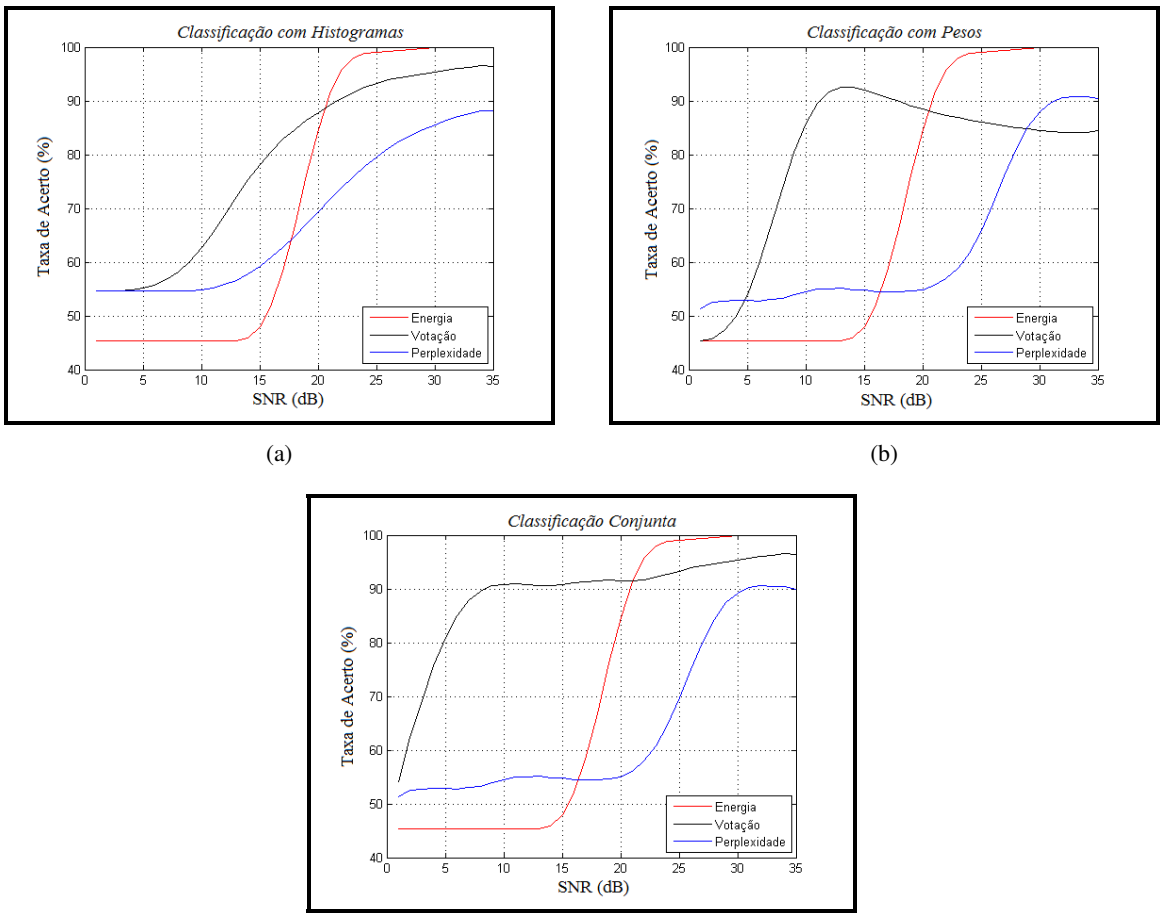


Figura 4.1 – Resultados de classificação para votação e perplexidade, com ruído branco, utilizando-se: (a) Histograma; (b) Pesos; (c) Histograma e Pesos.

Nesta figura, percebe-se que ao se utilizar o método de votação de forma conjunta com histogramas e pesos, a taxa de acerto permanece acima de 90 % para valores de SNR acima de 8 dB, fato que não acontece quando utiliza-se as técnicas de histogramas e pesos isoladamente. Infelizmente, o método de perplexidade não se mostrou robusto a baixos valores de SNR, assumindo uma taxa de acerto superior a 90 % apenas para valores de SNR acima de 30 dB.

Um fato curioso reside no método via pesos, conforme pode ser observado na Figura 4.1(b), pois a taxa de acerto começou a decair para valores de SNR acima de 13 dB. Por consta-

tação, conclui-se que os pesos dos átomos do dicionário, utilizados ao longo das iterações para sinais de baixa SNR, apresentaram melhor correlação com os pesos obtidos da base de dados de treinamento. Em outras palavras, no método via pesos, o sinal de voz com ruído pôde ser melhor classificado devido aos tipos de átomos que foram construídos em $\mathcal{D}$. O fato de técnicas distintas apresentarem desempenhos distintos para o mesmo dicionário, pode ser contornado se for utilizado um dicionário de tamanho mais elevado, contudo, o tempo de processamento aumentará proporcionalmente.

Um outro resultado de classificação pode ser observado na Figura 4.2. Nesta situação, utilizou-se um ruído do tipo pink (NOISEX-92), mas os resultados quanto ao desempenho foram similares ao obtido anteriormente.

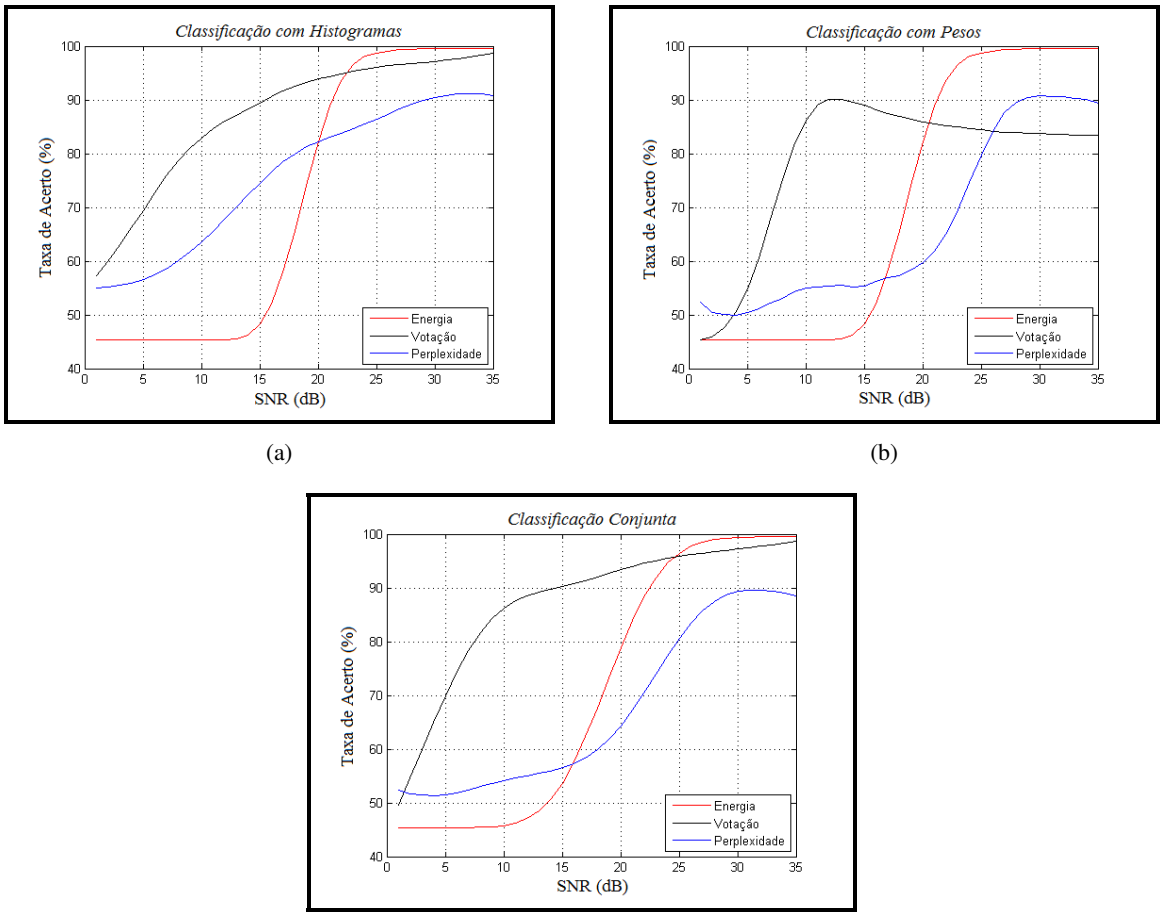


Figura 4.2 – Resultados de classificação para votação e perplexidade, com ruído pink, utilizando-se: (a) Histograma; (b) Pesos; (c) Histograma e Pesos.

5 CONCLUSÃO

O presente trabalho propôs duas técnicas novas de classificação de sinais de voz, definidas como votação e perplexidade, as quais se utilizam do método de decomposição atômica, conhecido como *Matching Pursuit*, como núcleo principal de processamento. Dentre os diferentes dicionários analisados, obteve-se melhor desempenho através dos átomos definidos pelo modelo de Gabor, o qual se utiliza de sinais cossenoidais janelados por gaussianas como base para construção do dicionário $\mathcal{D}$.

Entretanto, conforme abordado nos capítulos 3 e 4, apesar da técnica perplexidade ter se mostrado mais vantajosa (em termos de número de átomos no processo de classificação) do que a técnica votação, para sinais de alta SNR, infelizmente ela não apresentou o mesmo desempenho nas situações em que há influência de ruídos externos. De certa forma, houve uma tentativa de refinar ambas as técnicas utilizando as propabilidades provindas dos pesos e dos histogramas de ocorrência de forma individual ou conjunta, contudo, apenas no método via votação este refinamento mostrou-se vantajoso para sinais de alta ou baixa SNR. Contudo, embora o método via votação tenha apresentado algumas vantagens, em relação ao método via energia, no que diz respeito ao desempenho com sinais ruidosos, a complexidade computacional do algoritmo de classificação votação é superior ao algoritmo de classificação via energia. No caso, para um sinal de voz de tamanho N e dicionário de tamanho $\#\mathcal{D}$, define-se:

Método de Classificação	Complexidade
Energia	O(N)
Votação	O(N$\#\mathcal{D}^n$)

Tabela 5.1 – Análise de complexidade computacional.

Conforme destacado na tabela 5.1, a complexidade do algoritmo via votação é fortemente dependente do $\#\mathcal{D}^n$, onde n refere-se ao quanto o dicionário utilizado está adequado para descrever o sinal de entrada, sendo 1 no melhor caso e, maior do que 1 nas situações em que $\mathcal{D}$ não possui átomos fortemente correlacionados com o sinal de entrada. Com isso conclui-se que, conforme o número de átomos do dicionário aumenta, maior é o tempo de processamento requerido no processo de classificação via votação do que via energia.

De posse de todo o material que foi abordado nesta dissertação, é cabível fazer a seguinte pergunta: “Qual a vantagem dos métodos de classificação apresentados ?", infelizmente, em uma análise preliminar, não há vantagem em virtude da relação carga computacional exigida versus taxa de acerto de classificação. Contudo, existem dois fatores que influenciaram diretamente nos resultados e na carga computacional exigida, sendo eles: tamanho do dicionário e plataforma de simulação. Quanto ao primeiro, caso fosse utilizado um dicionário de tamanho mais elevado, por exemplo 300k átomos, maior seria a probabilidade de ocorrência de átomos que tivessem maior correlação com o sinal de voz em questão, e por conseguinte, menor seria o número de átomos necessários para síntese e posterior classificação. Com relação ao segundo fator, utilizou-se a plataforma Matlab em virtude do ambiente de simulação proporcionado por este *software*, pois, é possível analisar e validar os dados de uma forma mais fácil do que outros linguagens de programação, como por exemplo, C. Entretanto, o tempo de processamento exigido coloca-se como principal fator negativo e limitante para utilizar esta ferramenta como ambiente de processamento de sinais de voz via decomposição atômica. Dessa forma, apesar da dificuldade de se utilizar linguagens compiladas, como por exemplo C, esta seria um dos melhores recursos para execução das rotinas de classificação em tempo hábil de simulação.

REFERÊNCIAS

BLUMENSATH, T.; DAVIES, M. E. Stagewise weak gradient pursuits. **IEEE Transactions on Signal Processing**, v. 57, n. 11, p. 4333–4346, 2009.

BREEN, P. **Algorithms for Sparse Approximation**. [S.l.: s.n.], 2009.

CHEN, D. L. D. S. S.; SAUNDERS, M. Atomic decomposition by basis pursuit. **SIAM Review**, v. 43, n. 1, p. 129–159, 2001.

DYMARSKI1, P.; MOREAU, N.; RICHARD, G. Greedy sparse decompositions: a comparative study. **EURASIP Journal on Advances in Signal Processing**, 2011.

JAGGI, S. et al. High resolution pursuit for feature extraction. Technical report, MIT.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**. [S.l.]: Pearson Prentice Hall, 2008.

KLING, G.; ROADS, C. Audio analysis, visualization, and transformation with matching pursuits. 2004.

LIU, Q. W. Q.; WU, L. Size of the dictionary in matching pursuit algorithm. **IEEE Trans. Sig. Proc.**, v. 52, n. 12, p. 3403—-33408, 2004.

MALLAT, S.; ZHANG, Z. Matching pursuit with time-frequency dictionaries. **IEEE Trans. Sig. Proc.**, v. 41, p. 3397–3415, 1993.

NEEDELL, D.; VERSHYNIN, R. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. **IEEE J. Sel. Topics Signal Process**, v. 4(2), p. 310–316, 2010.

NOISEX-92 (**white noise**; **pink noise**). Disponível em: $< http : //spib.rice.edu/spib/select\_noise.html >$. Acesso em: 18 novembro. 2012, 15:23:00.

PLUMBLEY, M. et al. Sparse representations in audio and music: From coding to source separa-tion. **IEEE Trans. Sig. Proc.**, v. 98, n. 6, p. 995–1005, 2010.

REBOLLO-NEIRA, L.; LOWE, D. Optimized orthogo-nal matching pursuit approach. **IEEE Trans. Sig. Proc.**, v. 9, n. 4, p. 137–140, 2002.

STURM, B.; CHRISTENSEN, M. Cyclic matching pursuit with multiscale time-frequency dictionaries. 2010.

UMAPATHY, K. et al. Discrimination of pathological voices using a time-frequency approach. **IEEE Trans. Biomedical Eng.**, v. 52, n. 3, p. 421–430, 2005.

VERA-CANDEAS, P. et al. New matching pursuit based sinusoidal modelling method for audio coding. **IEEE Proc. Vis. Image Signal Process.**, v. 151, n. 1, p. 21–28, 2004.