8
Bibliografia

[1] Speech recognition – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Speech_recognition&oldid=474161027

[2] Mel-frequency cepstrum – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=450556487

[3] B. Gajic e K. K. Paliwal, "Robust Parameters for Speech Recognition Based on Subband Spectral Centroid Histograms", Eurospeech 2001, v. 1, pp. 591–594, Setembro de 2001.

[4] C. Kim e R. M. Stern, "Feature extraction for robust speech recognition using a power-law nonlinearity and power-bias subtraction", INTERSPEECH-2009, pp. 28–31, Setembro de 2009.

[5] O. Frooq e S. Datta, "Wavelet-based denoising for robust feature extraction for speech recognition", IEE Electronics Letters, vol. 39, no. 1, pp. 163-165, 9 de Janeiro de 2003

[6] S. Haykin, "Neural Networks: A Comprehensive Foundation", 2nd ed., Prentice Hall, 1999.

[7] Hidden Markov model – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Hidden_Markov_model&oldid=473681336

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognitions," Proceedings of the IEEE, vol.77, no 2, pp. 257-286, Fevereiro de 1989.

[9] S. Cerqueira Uispo dos Santos e A. Alcaim, "Role of syllables and function-words in continuous Portuguese speech recognition", IEE Electronics Letters, vol. 36, no. 12, pp. 1083-1085, 8 de Junho de 2000

[10] N.J. Hunt, N. Lennig e P. Mermeletein, "Experiments in Syllable-Based Recognition of Continuous Speech", Proc. Int. Conf. Acoust. Speech Signal Process., pp. 880-883, 1980.

[11] X. Huang, A. Acero e H-W Hon, "Spoken Language Processing: A Guide to Theory, Algorithm and System Development", Prentice Hall, 2001.

[12] C. Martins, A. Teixeira e J. Neto, "Language Models in Automatic Speech Recognition", Revista do Detua, vol.4, no 2, Janeiro de 2004

[13] Bayes' theorem – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=473895564

[14] G. D. Forney, Jr., "The Viterbi Algorithm," Proc. IEEE, vol. 61, pp. 268-278, Março de 1973.

[15] Central limit theorem – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Central_limit_theorem&oldid=464547069

[16] Derivation of the Baum-Welch algorithm for HMMs Acesso em 2 de fevereiro de 2012. Disponível em: http://pandamatak.com/people/anand/771/html/node26.html

[17] J. Ming, "Robust Speech Recognition Using Probabilistic Union Models" IEEE Transactions on Speech and Audio Processing, vol. 10, no 6, pp. 403-414, Setembro de 2002.

[18] Pre-emphasis – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Pre-emphasis&oldid=435730491

[19] Window function – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Window_function&oldid=473404098#Hamming_window

[20] Discrete Fourier transform – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Discrete_Fourier_transform&oldid=474379311

[21] Min Xu et al., "HMM-Based Audio Keyword Generation", Advances in Multimedia Information Processing - PCM 2004: 5th Pacific Rim Conference on Multimedia. Springer, 2004.

[22] Mel scale – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Mel_scale&oldid=471852897

[23] B. Gajic, "Auditory Based Methods for Robust Speech Feature Extraction", Telektronikk, v. 2, pp. 45-58, 2003.

[24] C. Kim e R. M. Stern, "Feature extraction for robust speech recognition based on maximizing the sharpness of the power distribution and on power flooring", Proc. IEEE ICASSP, 2010.

[25] R. D. Patterson e J. Holdsworth, "A functional model of neural activity patterns and auditory images", Advances in Speech, Hearing and Language Processing, pp. 547-563, 1996.

[26] M. Slaney, "An efficient implementation of the Patterson-Holdsworth auditory filter bank", Tech. Rep. 35, Apple Computer Inc., 1993

[27] Discrete cosine transform – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Discrete_cosine_transform&oldid=472845638

[28] J.S. Mason e X. Zhang, "Velocity and acceleration features in speaker recognition", Proc. ICASSP, vol. 5, pp. 3673-3677, 1991.

[29] Stéphane Mallat: A Wavelet Tour of Signal Processing, 2nd edition, Hardcover, 1999.

[30] L. Chun-Lin, "A Tutorial of the Wavelet Transform". 23 de Fevereito de 2010

[31] Supervised learning – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Supervised_learning&oldid=46937

5320
[32] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks", Neural Networks, vol. 4 issue 2, 1991
[33] LDC Catalog. Acesso em 2 de fevereiro de 2012. Disponível em: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1
[34] Signal-to-noise ratio – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=Signal-to-noise_ratio&oldid=472976865
[35] NOISEX. Acesso em 2 de fevereiro de 2012. Disponível em: http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html
[36] HTK Speech Recognition Toolkit. Acesso em 2 de fevereiro de 2012. Disponível em: http://htk.eng.cam.ac.uk/
[37] S. Young, et al., "The HTK Book: HTK Tools and Reference Manuals", Entropic, 1999.
[38] MATLAB – Wikipedia, the free encyclopedia. Acesso em 2 de fevereiro de 2012. Disponível em: http://en.wikipedia.org/w/index.php?title=MATLAB&oldid=474313431
[39] continuous-speech-recognition-using-matlab – Continuous Speech Recognition using Matlab – Google Project Hosting. Acesso em 2 de fevereiro de 2012. Disponível em: http://code.google.com/p/continuous-speech-recognition-using-matlab/
[40] F. Dellaert, "The Expectation Maximization Algorithm", College of Computing, Georgia Institute of Technology Technical, Report number GIT-GVU-02-20, Fevereiro de 2002.

9
Apêndice: Demonstrações

Este apêndice contém demonstrações de algumas expressões apresentadas nos capítulos anteriores.

9.1
Probabilidade de Observação num HMM

Seja $O = \{O_1, O_2, \ldots, O_T\}$ a sequência de vetores extraídos do sinal de voz. O objetivo é calcular a probabilidade $P(O)$ em **cada um dos modelos de palavra**. Isso será feito em etapas.

Para começar, seja s_1 o primeiro estado ocorrido, s_2 o segundo, etc, até o T-ésimo estado s_T. Deve-se lembrar que a sequência de estados é aleatória e, por isso, não se sabe a priori qual estado ocorreu em cada instante. A probabilidade de ocorrer a sequência específica $s = \{s_1, s_2, \ldots, s_T\}$ é dada por

$$\begin{aligned} P(s) &= P(s_1, s_2, \ldots, s_T) \\ &= P(s_1)\frac{P(s_1, s_2)}{P(s_1)}\frac{P(s_1, s_2, s_3)}{P(s_1, s_2)}\cdots\frac{P(s_1, s_2, \ldots, s_T)}{P(s_1, s_2, \ldots, s_{T-1})} \end{aligned} \tag{47}$$

Usando a definição de probabilidade condicional e aplicando a propriedade (1), a expressão se reduz a

$$\begin{aligned} P(s) &= P(s_1)P(s_2|s_1)P(s_3|s_1, s_2)\ldots P(s_T|s_{T-1}, s_{T-2}, \ldots, s_1) \\ &= P(s_1)P(s_2|s_1)P(s_3|s_2)\ldots P(s_T|s_{T-1}) \\ &= \pi_{s_1}\prod_{t=2}^{T} a_{s_{t-1}s_t} \end{aligned} \tag{48}$$

Outro cálculo simples é a probabilidade de ocorrer uma determinada sequência de observações dado uma sequência transição de estados. Conhecido o caminho, sabe-se exatamente qual é a densidade de probabilidade em cada instante.

$$
\begin{aligned}
P(O|s) &= P(O_1, O_2, ..., O_T \,|\, s_1 s_2 ... s_T) \\
&= P(O_1|s_1) P(O_2|s_2) ... P(O_T|s_T) \\
&= b_{s_1}(O_1) b_{s_2}(O_2) .. b_{s_T}(O_T) \\
&= \prod_{t=1}^{T} b_{s_t}(O_t)
\end{aligned}
\tag{49}
$$

Finalmente, usando o teorema da probabilidade total, obtém-se a probabilidade de ocorrer uma determinada sequência de observações num modelo.

$$
\begin{aligned}
P(O) &= \sum_{\text{todo s}} P(O|s) P(s) \\
&= \sum_{\text{todo s}} \pi_{s_1} b_1(O_t) \prod_{t=2}^{T} a_{s_{t-1} s_t} b_{s_t}(O_t)
\end{aligned}
\tag{50}
$$

9.2
Algoritmo Forward-Backward

Seja $P(O_1, O_2, ... , O_t)$ a probabilidade das primeiras t observações, com $t \leq T$. Usando o teorema da probabilidade total, têm-se

$$
P(O_1 O_2 ... O_t) = \sum_{\text{para todo } s_t} P(O_1, O_2, ..., O_t \,|\, s_t) = \sum_{j=1}^{N} P(O_1, O_2, ..., O_t, s_t = j)
\tag{51}
$$

O objetivo passa a ser então encontrar $P(O_1, ... , O_t, s_t = j)$ em função da expressão do instante anterior $P(O_1, ... , O_{t-1}, s_{t-1} = j)$, para reaproveitar os cálculos iniciais e diminuir o custo computacional. Com a definição de probabilidade condicional, tem-se

$$
\begin{aligned}
&P(O_1, O_2, ..., O_t, s_t = j) = \\
&P(O_t \,|\, O_1, O_2, ..., O_{t-1}, s_t = j) P(O_1, O_2, ..., O_{t-1}, s_t = j)
\end{aligned}
\tag{52}
$$

O primeiro termo pode ser simplificado porque, conhecido o estado no instante t, a probabilidade de observar O_t não depende das observações passadas. O segundo termo é expandido pelo teorema da probabilidade total, para se acrescentar o estado anterior q_{t-1}. Após algumas manipulações matemáticas e propriedades do HMM, a expressão é reescrita enfim como função do instante anterior, resultando em

$$
\begin{aligned}
&P(O_1,\ldots,O_t,s_t=j)=P(O_t|s_t=j)\sum_{i=1}^{N}P(O_1,\ldots,O_{t-1},s_t=j,s_{t-1}=i)\\
&=b_j(O_t)\sum_{i=1}^{N}P(O_1,\ldots,O_{t-1}|s_t=j,s_{t-1}=i)P(s_t=j,s_{t-1}=i)\\
&=b_j(O_t)\sum_{i=1}^{N}P(O_1,\ldots,O_{t-1}|s_{t-1}=i)P(s_t=j,s_{t-1}=i)\\
&=b_j(O_t)\sum_{i=1}^{N}\frac{P(O_1,\ldots,O_{t-1},s_{t-1}=i)}{\cancel{P(s_{t-1}=i)}}P(s_t=j|s_{t-1}=i)\,\cancel{P(s_{t-1}=i)}\\
&=b_j(O_t)\sum_{i=1}^{N}P(O_1,\ldots,O_{t-1},s_{t-1}=i)a_{ij}
\end{aligned}
\tag{53}
$$

Para que a indução se feche, resta apenas definir a probabilidade no instante 1, o que é imediato pelas definições do Modelo de Markov Escondido.

$$
P(O_1,s_1=j)=P(O_1|s_1=j)P(s_1=j)=b_j(O_1)\pi_j \tag{54}
$$

Conectando (51), (53) e (54), o método completo é resumido em

$$\alpha_t(j)=P(O_1,O_2,\ldots,O_t,s_t=j)$$

$$
\begin{aligned}
&\text{1) Inicialização: } \alpha_1(j)=b_j(O_1)\pi_j\\
&\text{2) Indução: } \alpha_t(j)=b_j(O_t)\sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}\\
&\text{3) Finalização: } P(O)=P(O_1,O_2,\ldots,O_T)=\sum_{j=1}^{N}\alpha_T(j)
\end{aligned}
\tag{55}
$$

9.3
Probabilidade de Ocorrência de Palavras

Seja $w = \{w_1, w_2, \ldots, w_V\}$ uma sequência de V palavras. A probabilidade de observar essa sequência é dada por

$$
\begin{aligned}
P(w)&=P(w_1,w_2,\ldots,w_V)\\
&=P(w_1)\frac{P(w_1,w_2)}{P(w_1)}\frac{P(w_1,w_2,w_3)}{P(w_1,w_2)}\ldots\frac{P(w_1,w_2,\ldots,w_V)}{P(w_1,w_2,\ldots,w_{V-1})}\\
&=P(w_1)P(w_2|w_1)P(w_3|w_1,w_2)\ldots P(w_m|w_1,w_2,\ldots w_{V-1})\\
&=\prod_{i=1}^{V}P(w_i|w_1,\ldots w_{i-1})
\end{aligned}
\tag{56}
$$

É possível aproximar a probabilidade condicional acima, para diminuir o trabalho computacional. A observação da palavra w_i é muito mais influenciada pela ocorrência das palavras w_{i-1} e w_{i-2} (imediatamente anteriores) do que pela ocorrência das palavras iniciais w_1 e w_2. Logo, pode-se considerar apenas as G–1 palavras anteriores

$$\begin{aligned} P\left(w_i \middle| w_1,...w_{i-1}\right) &\approx P\left(w_i \middle| w_{i-(G-1)},w_{i-(G-2)},...w_{i-1}\right) \\ &\Downarrow \\ P(w) &= \prod_{i=1}^{V} P\left(w_i \middle| w_1,...w_{i-1}\right) \\ &\approx \prod_{i=1}^{V} P\left(w_i \middle| w_{i-(G-1)},w_{i-(G-2)},...w_{i-1}\right) \end{aligned} \tag{57}$$

Por fim, a probabilidade de observação de w_i, dadas as G–1 palavras passadas, pode ser estimada por uma contagem estatística em textos de treinamento

$$\begin{aligned} P\left(w_i \middle| w_{i-(G-1)},w_{i-(G-2)},...w_{i-1}\right) &= \frac{P\left(w_{i-(G-1)},w_{i-(G-2)},...w_i\right)}{P\left(w_{i-(G-1)},w_{i-(G-2)},...w_{i-1}\right)} \\ &\approx \frac{count\left(w_{i-(G-1)},w_{i-(G-2)},...w_i\right)}{count\left(w_{i-(G-1)},w_{i-(G-2)},...w_{i-1}\right)} \end{aligned} \tag{58}$$

Valores grandes de N trazem melhores resultados, mas o custo de processamento aumenta rapidamente. Por causa da limitação do software utilizado neste trabalho (vide seção 6.2), G é arbitrado para o valor 3, resultando num modelo de trigramas. Assim, as equações (57) e (58) se reduzem respectivamente a

$$P(w) \approx \prod_{i=1}^{V} P\left(w_i \middle| w_{i-2},w_{i-1}\right) \tag{59}$$

$$P\left(w_i \middle| w_{i-2},w_{i-1}\right) \approx \frac{count\left(w_{i-2},w_{i-1},w_i\right)}{count\left(w_{i-2},w_{i-1}\right)} \tag{60}$$

9.4 Algoritmo de Viterbi

Seja a variável $V_t(j)$ igual a

$$V_t(j) = \max_{s_1,s_2,\ldots,s_{t-1}} P(O_1,O_2,\ldots,O_t,s_1,s_2,\ldots,s_{t-1},s_t = j) \tag{61}$$

Com o mesmo recurso usado em (53), podemos expressar $V_t(j)$ em função dos termos anteriores $V_{t-1}(i)$ com

$$\begin{aligned}
&V_t(j) = \\
&= \max_{s_1,s_2,\ldots,s_{t-1}} P(O_1,O_2,\ldots,O_{t-1},s_1,s_2,\ldots,s_{t-2}|O_t,s_{t-1},s_t = j)P(O_t,s_t = j,s_{t-1}) \\
&= \max_{s_1,s_2,\ldots,s_{t-1}} P(O_1,O_2,\ldots,O_{t-1},s_1,s_2,\ldots,s_{t-2}|s_{t-1})P(O_t|s_t = j)P(s_t = j,s_{t-1}) \\
&= P(O_t|s_t = j) \max_{s_1,s_2,\ldots,s_{t-1}} \frac{P(O_1,O_2,\ldots,O_{t-1},s_1,s_2,\ldots,s_{t-2},s_{t-1})}{P(s_{t-1})} P(s_t = j,s_{t-1}) \\
&= b_j(O_t) \max_{s_1,s_2,\ldots,s_{t-1}=1\le i\le N} P(O_1,O_2,\ldots,O_{t-1},s_1,s_2,\ldots,s_{t-2},s_{t-1})P(s_t = j|s_{t-1}) \\
&= b_j(O_t) \max_{1\le i\le N} V_{t-1}(i)a_{ij}
\end{aligned} \tag{62}$$

A equação (62) traduz a ideia geral do algoritmo: considerar apenas o máximo dos caminhos passados. Para que a indução se feche, resta apenas o termo inicial, obtido facilmente com as propriedades do HMM.

$$V_1(j) = P(O_1,s_1 = j) = P(O_1|s_1 = j)P(s_1 = j) = b_j(O_1)\pi_j \tag{63}$$

A partir de cada $V_{t-1}(i)$, é possível acumular os nós anteriores de maior probabilidade numa variável $B_t(i)$, conforme se vê em

$$\begin{aligned}
&\text{1) Inicialização: } \begin{cases} V_1(j) = b_j(O_1)\pi_j \\ B_1(j) = 0 \end{cases} \\
&\text{2) Indução: } \begin{cases} V_t(j) = b_j(O_t)\max_{1\le i\le N} V_{t-1}(i)a_{ij} \\ B_t(j) = \arg\max_{1\le i\le N} V_{t-1}(i)a_{ij} \end{cases} \\
&\text{3) Finalização: } \widehat{s_T} = \arg\max_{1\le i\le N} B_T(i) \\
&\text{4) Backtracing: } \begin{cases} \widehat{s_t} = B_{t+1}(\widehat{s_{t+1}}), t = T-1,T-2,\ldots,1 \\ \hat{s} = \{\widehat{s_1},\widehat{s_2},\ldots,\widehat{s_T}\} \end{cases}
\end{aligned} \tag{64}$$

9.5 Algoritmo de Baum-Welch

Esta técnica utiliza o algoritmo *Expectation-Maximization* (EM) [40]. Ele é apropriado justamente para estimar parâmetros quando não se conhece uma das variáveis (neste caso, a sequência de estados *s*). Apesar do resultado não ser

necessariamente o máximo global, ele garante a convergência para um máximo local.

O procedimento é explicado na equação (65).

$$Q\left(\theta,\theta^{(n-1)}\right)=E_{s|O,\theta^{(n-1)}}\left[\log P\left(O,s|\theta\right)\right]=\sum_{\text{todo s}}P\left(s\left|O,\theta^{(n-1)}\right.\right)\log P\left(O,s|\theta\right)$$

1) Faça $n=0$ e escolha um $\theta^{(n)}$ inicial

2) Adicione 1 a n e faça $\theta^{(n)}=\arg\max_{\theta}Q\left(\theta,\theta^{(n-1)}\right)$

3) Repetir 2 até convergência (65)

Portanto, o conjunto de parâmetros θ é recalculado várias vezes (com base nos valores encontrados no passo anterior), até que os novos resultados não apresentem grande mudança (convergência). O índice (n) indica qual o número da reestimação.

Resta saber como calcular $Q(\theta,\theta^{(n-1)})$ e como maximizá-lo de forma eficiente. Utilizando (6), tem-se

$$\begin{aligned}\log P\left(O,s|\theta\right)&=\log\left(\pi_{s_1}\prod_{t=2}^{T}a_{s_{t-1}s_t}\prod_{t=1}^{T}b_{s_t}\left(O_t\right)\right)\\&=\log\pi_{s_1}+\sum_{t=2}^{T}\log a_{s_{t-1}s_t}+\sum_{t=1}^{T}\log b_{s_t}\left(O_t\right)\end{aligned}\tag{66}$$

onde $b_s(O_t)$ é dado por (13), com parâmetros c_{ms}, μ_{ms} e Λ_{ms}.

Com isso, $Q(\theta,\theta^{(n-1)})$ se desenvolve para

$$\begin{aligned}Q\left(\theta,\theta^{(n-1)}\right)&=\sum_{\text{todo s}}P\left(s\left|O,\theta^{(n-1)}\right.\right)\log P\left(O,s|\theta\right)\\&=\sum_{\text{todo s}}\frac{P\left(O,s\left|\theta^{(n-1)}\right.\right)}{P\left(O\left|\theta^{(n-1)}\right.\right)}\left(\log\pi_{s_1}+\sum_{t=2}^{T}\log a_{s_{t-1}s_t}+\sum_{t=1}^{T}\log b_{s_t}\left(O_t\right)\right)\end{aligned}\tag{67}$$

Aqui que se percebe a vantagem do algoritmo EM: graças ao logaritmo, o somatório de (67) pode ser separado em três partes distintas.

$$Q\left(\theta,\theta^{(n-1)}\right)=Q_\pi\left(\pi,\theta^{(n-1)}\right)+Q_a\left(a,\theta^{(n-1)}\right)+Q_b\left(b,\theta^{(n-1)}\right)$$

$$Q_\pi\left(\pi,\theta^{(n-1)}\right)=\sum_{\text{todo s}}\frac{P\left(O,s\middle|\theta^{(n-1)}\right)}{P\left(O\middle|\theta^{(n-1)}\right)}\log\pi_{s_t}$$

$$Q_a\left(a,\theta^{(n-1)}\right)=\sum_{\text{todo s}}\frac{P\left(O,s\middle|\theta^{(n-1)}\right)}{P\left(O\middle|\theta^{(n-1)}\right)}\sum_{t=2}^{T}\log a_{s_{t-1}s_t}$$

$$Q_b\left(b,\theta^{(n-1)}\right)=\sum_{\text{todo s}}\frac{P\left(O,s\middle|\theta^{(n-1)}\right)}{P\left(O\middle|\theta^{(n-1)}\right)}\sum_{t=1}^{T}\log b_{s_t}\left(O_t\right) \tag{68}$$

Como cada termo da soma de Q carrega termos diferentes de θ (π, a, b), a maximização total será a soma de cada maximização. Em outras palavras, Q_π, Q_a e Q_b podem ser maximizados separadamente, reduzindo bastante a complexidade do processo. A otimização de cada termo usa técnicas como o algoritmo forward-backward e multiplicadores de Lagrange, como demonstrado em [11 pp. 172-176], resultando em

$$\pi_i^{(n)}=\gamma_1(i)$$

$$a_{ij}^{(n)}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$c_{jk}^{(n)}=\frac{\sum_{t=1}^{T}\rho_t(j,k)}{\sum_{t=1}^{T}\sum_{m=1}^{T}\rho_t(j,m)}$$

$$\mu_{jk}^{(n)}=\frac{\sum_{t=1}^{T}\rho_t(j,k)O_t}{\sum_{t=1}^{T}\rho_t(j,k)}$$

$$\Lambda_{jk}^{(n)}=\frac{\sum_{t=1}^{T}\rho_t(j,k)\left(O_t-\mu_{jk}\right)\left(O_t-\mu_{jk}\right)^T}{\sum_{t=1}^{T}\rho_t(j,k)} \tag{69}$$

$$
\begin{aligned}
\alpha_1(j) &= b_j^{(n-1)}(O_1)\pi_j^{(n-1)} \\
\alpha_t(j) &= b_j^{(n-1)}(O_t)\sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}^{(n-1)} \\
\beta_T(i) &= 1 \\
\beta_t(i) &= \sum_{j=1}^{N} a_{ij}^{(n-1)}\, b_j^{(n-1)}(O_{t+1})\beta_{t+1}(i) \\
\gamma_t(i) &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)} \\
\xi_t(i,j) &= \frac{\alpha_t(i)a_{ij}^{(n-1)}\, b_j^{(n-1)}(O_{t+1})\beta_{t+1}(i)}{\sum_{k=1}^{N}\sum_{l=1}^{N}\alpha_t(k)a_{kl}^{(n-1)}\, b_l^{(n-1)}(O_{t+1})\beta_{t+1}(l)} \\
\rho_t(j,k) &= \gamma_t(j)\frac{c_{jk}^{(n-1)}\ \phi\left(O_t,\mu_{jk}^{(n-1)}\ ,\Lambda_{jk}^{(n-1)}\ \right)}{\sum_{m=1}^{M}c_{jk}^{(n-1)}\ \phi\left(O_t,\mu_{jm}^{(n-1)}\ ,\Lambda_{jm}^{(n-1)}\ \right)}
\end{aligned}
\tag{70}
$$

9.6
Criação dos Filtros do Método MFCC

Para criar os filtros triangulares da Figura 28, primeiro deve-se calcular os valores k_0, k_1, ..., k_{B+1}.

Como explicado em 3.4.1, a variação das frequências em mel acompanha a percepção humana das vibrações. Surge então a ideia de dividir eixo de frequências na escala mel, e não em Hz. Assim, são gerados os valores igualmente espaçados m_0, m_1, ..., m_{B+1}. Convertendo os números de volta para Hz, obtém-se $f_0, f_1, ..., f_{B+1}$, como mostra a Figura 26.

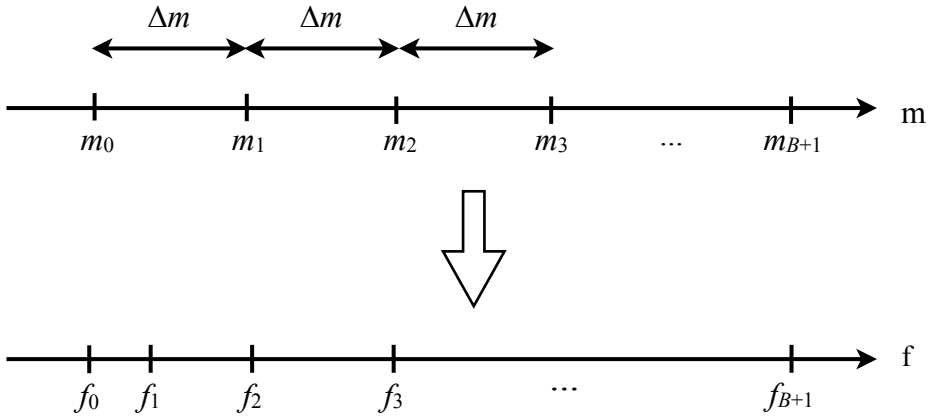

Figura 62: Divisão da escala mel em intervalos de mesmo comprimento e a conversão dessas fronteiras para a escala em Hz.

Para determinar matematicamente os segmentos, parte-se das frequências extremas f_0 e f_{B+1} do espectro. Esses valores são convertidos para mel usando (19) e usados para dividir o intervalo em B+1 partes iguais.

$$\begin{aligned} \Delta m &= \frac{m_{B+1} - m_0}{B+1} = \frac{M\left(f_{B+1}\right) - M\left(f_0\right)}{B+1} \\ m_b &= m_1 + b\Delta m = M\left(f_0\right) + b\Delta m \end{aligned} \tag{71}$$

Para obter os valores em Hz, basta aplica a função inversa.

$$f_b = M^{-1}\left(m_b\right) = M^{-1}\left(M\left(f_0\right) + b\frac{M\left(f_{B+1}\right) - M\left(f_0\right)}{B+1} \right) \tag{72}$$

Como o sinal é discreto, as frequências contínuas f_b precisam ser convertidas para frequências digitais k_b.

$$k_b = \frac{N}{F_S} f_b \tag{73}$$

onde N é o número de amostras da transformada discreta de Fourier do sinal e F_S é a sua taxa de amostragem.

Definidas as fronteiras, os filtros triangulares digitais são dados por

$$H_b\left(k\right) = \begin{cases} 0 & k < k_{b-1} \\ \dfrac{k - k_{b-1}}{k_b - k_{b-1}} & k_{b-1} \le k < k_b \\ \dfrac{k_{b+1} - k}{k_{b+1} - k_b} & k_b \le k < k_{b+1} \\ 0 & k \ge k_{b+1} \end{cases} \tag{74}$$