2
Conceitos básicos

O objetivo deste Capítulo é abordar teoricamente os assuntos que formam a base para o desenvolvimento do modelo proposto e a descrição do modelo de Fritchman, que devido sua frequente aplicação em trabalhos encontrados na literatura da área, foi utilizado como modelo de referência para o ajuste dos dados segundo as estatísticas de interesse. Assim, este Capítulo está dividido em duas partes. A primeira trata dos modelos de Markov, incluindo os Modelos Escondidos de Markov (HMM) e o método de Baum-Welch para estimação de seus parâmetros. A segunda parte aborda as ferramentas de otimização empregadas para estimação dos parâmetros do modelo proposto, incluindo uma técnica de otimização clássica e o processo heurístico conhecido como PSO (Particle Swarm Optimization).

2.1.
Modelos de Markov

Estes modelos tem como elemento básico a Cadeia de Markov, que é um processo estocástico de tempo discreto em que a transição para um estado no instante t, depende somente do estado visitado no instante t-1.

Define-se como probabilidade de transição aquela que caracteriza a transição para o estado J_t (estado no instante t) dado que o estado anterior foi I_{t-1}.

Esta é a definição do que é denominado de cadeia de Markov de ordem um, pois é possível conceber a situação onde a transição para um estado no instante t depende dos k últimos estados visitados. Neste caso a cadeia de Markov é denominada de ordem k.

Neste trabalho, concentraremos nossa atenção nas cadeias de Markov homogêneas e estacionárias de primeira ordem caracterizada por um número finito de estados, onde o estado no instante t é denotado por s_t.

Assim uma cadeia de Markov pode ser caracterizada pela trinca $\{S, \pi, P\}$ onde:

$S = \{1,2,...,u\}$ – Conjunto dos estados do modelo

$\pi = [p_1\ p_2 \ldots p_u]$, $p_j = P(s_0 = j)$ – Vetor das probabilidades iniciais de estado

$P = [\ p_{ij}\ ]_{u \times u}$, $p_{ij} = P(s_t = j \mid s_{t-1} = i)$ – Matriz de probabilidades de transição

A Figura abaixo ilustra uma cadeia de Markov de três estados, sendo que entre eles, todas as transições são possíveis.

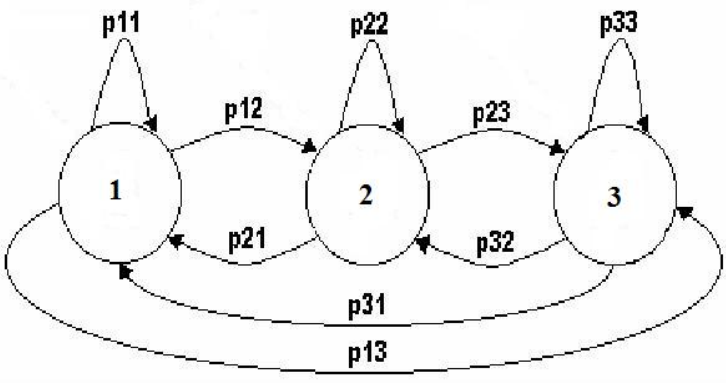

Figura 2.1 – Cadeia de Markov de 3 estados

2.2. Modelos Escondidos de Markov

Denomina-se de Modelo Escondido de Markov (HMM) a um modelo baseado em uma cadeia de Markov onde:

- Uma observação é produzida ao se acessar cada estado. Esta observação é de natureza estocástica e particular de cada estado.
- Os estados da cadeia de Markov subentendida não são observáveis (daí o nome "escondido").

Assim um Modelo Escondido de Markov é caracterizado por uma cadeia de Markov subjacente e por distribuições de probabilidade das observações produzidas em cada estado. Mais formalmente, um HMM pode ser descrito pelo seguinte conjunto de parâmetros:

$$\{ S, A, \pi, P, B \}$$

onde:

$S = \{1,2,...,u\}$ – Conjunto dos estados do modelo;

$A = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ – Conjunto dos símbolos de saída;

$\pi = [p_1\ p_2\ ...\ p_u]$, $p_j = P(s_1=j)$ – Vetor das probabilidades iniciais de estado;

$P = [\ p_{ij}\ ]_{u\ x\ u}$, $p_{ij} = P(s_t= j \mid s_{t-1}= i)$ – Matriz de probabilidades de transição;

$B = [P(O \mid j)]$ – Vetor com a distribuição de probabilidades das observações (O) por estado (j).

Em resumo, o HMM é um modelo composto de uma cadeia de Markov subjacente cujos estados não são observáveis, onde uma observação é emitida quando da permanência em qualquer um de seus estados.

Como veremos mais adiante, os modelos HMM tem se mostrado extremamente úteis em Telecomunicações, particularmente no que diz respeito à modelagem de canais de comunicações em processos de transmissão digital [16], [17]. Esta discussão é o objetivo da próxima seção.

2.3. HMM aplicado a canais sujeitos a erros em surtos

Em comunicações digitais, podemos supor, sem perda de generalidade, que as imperfeições existentes no canal de comunicações farão com que a sequência recebida seja diferente da sequência transmitida, o que caracteriza o que é chamado de *erro de transmissão*. Assim todo o efeito danoso do canal de comunicações pode ser igualmente representado por uma sequência binária onde por convenção, usaremos 1 e 0 para respectivamente representar a ocorrência e não ocorrência de erro num particular instante do processo de transmissão de dados.

Assim caracterização do comportamento de um canal de comunicações pode ser descrita por uma longa sequência binária, majoritariamente constituída por 0's, onde os 1's representam a ocorrência de erros no processo de transmissão.

Tem sido o objetivo de vários pesquisadores nesta área, a busca por modelos teóricos de natureza estocástica que representem probabilisticamente e descrevam adequadamente este fenômeno de ocorrência de erros provocados pelo canal de comunicações. A utilidade de tais modelos é indiscutível e dentre as suas principais vantagens podemos citar a capacidade de simular novos sistemas de comunicações em canais de comunicações cujos efeitos sobre os dados transmitidos podem ser avaliados sem a necessidade de práticas experimentais.

Com relação aos efeitos danosos causados pelos canais de comunicações com memória constata-se que os erros não podem ser considerados estatisticamente independentes. Observa-se em muitos casos que erros acontecem na forma que é comumente denominada de *surtos*, onde longos períodos de ausência de erros são seguidos por curtos períodos de alta incidência de erros.

Assim, a busca de modelos teóricos que sejam capazes de explicar estatisticamente este fenômeno de surtos, pode ser considerada extremamente importante. Observa-se que a literatura tem revelado que os modelos escondidos de Markov tem sido usados na explicação estatística de alguns fenômenos de ocorrência de erros em surtos [3, 4, 7, 16, 17,19, 20, 25, 26]. Porém estes não são adequados a muitos dos casos de geração de surtos, e ainda estão sujeitos a limitações impostas pelo impraticável aumento de complexidade matemática quando se busca maior precisão de ajustes por meio do aumento do número de estados.

Esta tese é basicamente motivada por essa constatação, e visa propor e discutir um modelo que apresente adequada capacidade para representar tal fenômeno através de seu comportamento estatístico, superando as dificuldades acima mencionadas, de um novo HMM com estrutura especial, direcionada a geração de sequências de erros.

A título de ilustração, serão apresentados a seguir, alguns modelos clássicos de HMM encontrados na literatura para a caracterização estatística de canais de comunicações.

2.4. Modelo de Gilbert-Elliot

O modelo denominado de Gilbert-Elliott, em referência a seus autores, é um HMM de dois estados. O diagrama de estados correspondente encontra-se na Figura 2.2.

Este modelo é constituído de um estado onde a ocorrência de erros possui baixa probabilidade (daí o estado ser chamado de “bom” ou “good” (G)) e por um segundo estado , onde a ocorrência de erros possui alta probabilidade (estado chamado de “ruim” ou “bad” (B)). A frequência com que se permanece em cada um destes estados é controlada pelas probabilidades de transição entre estados.

Assim, este modelo é caracterizado por cinco parâmetros e descrito por:

$\{ S, A, \pi, P, B_e \}$

onde:

$S = \{ 1, 2 \}$

$A = \{ 0, 1 \}$

$\pi = [p_1 \;\; 1\text{-}p_1]$

$$P = \begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix}$$

$B_e = [p_{e1} \;\; p_{e2}]$ (Vetor das probabilidades de erro por estado)

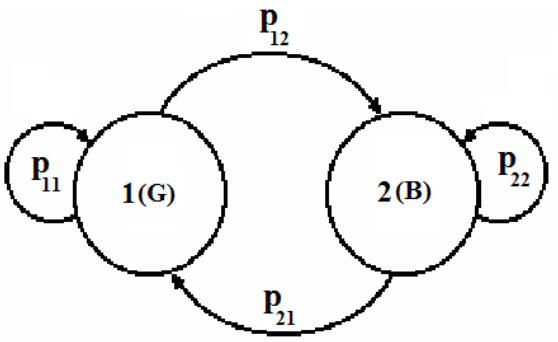

Figura 2.2 Modelo de Gilbert-Elliott

2.5. Modelo de Fritchman

Este modelo representa uma generalização do anterior e é um modelo consagrado e amplamente empregado na modelagem de erros em surtos. Por ser particularmente importante, será tomado como referência de comparação com o modelo proposto nesta tese.

Num modelo de Fritchman de N estados, estes são particionados em dois grupos, um com k estados de erro (ruins), que geram "1"s com probabilidade 1, e o outro com N-k estados livres de erros (bons), que geram "0"s com probabilidade 1.

Somente são permitidas auto-transições ou transições entre estes grupos[7]. Assim, a distribuição do comprimento dos *gaps* (intervalo entre surtos) é descrita por uma combinação linear de *N-k* distribuições exponenciais, enquanto a distribuição de comprimentos de surtos é descrita por uma combinação linear de *k* distribuições exponenciais [4]. Apesar de este ser modelo mais rico do que o Gilbert-Elliott, tem-se verificado que é inadequado para muitos casos de interesse [26, 32]. Este modelo possui um número maior de estados e conseqüentemente a estimação de seus parâmetros apresenta complexidade em geral elevada.

Uma particularização do modelo de Fritchman é o denominado de modelo Fritchman-SES (SES – *Single-Error-State*), selecionado em [9] para modelar o "HF SchEMe" (*Skywave Channel Error Model*), e corresponde ao caso em que k=1.

A Figura 2.3 ilustra um modelo de Fritchman-SES de N estados, onde o estado ruim é indicado por (*B*) e os *N-1* estados bons são respectivamente indicados por (G_1), (G_2) ... (G_{N-1}). Este modelo é descrito pela matriz de probabilidades de transição mostrada em (2.1), onde o estado ruim é o de índice *N*:

$$\mathbf{P} = \begin{bmatrix} p_{11} & 0 & 0 & \cdots & 0 & p_{1N} \\ 0 & p_{22} & 0 & \cdots & 0 & p_{2N} \\ 0 & 0 & p_{33} & \cdots & 0 & p_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p_{N-1,N-1} & 0 \\ p_{N1} & p_{N2} & p_{N3} & \cdots & p_{N,N-1} & p_{NN} \end{bmatrix} \tag{2.1}$$

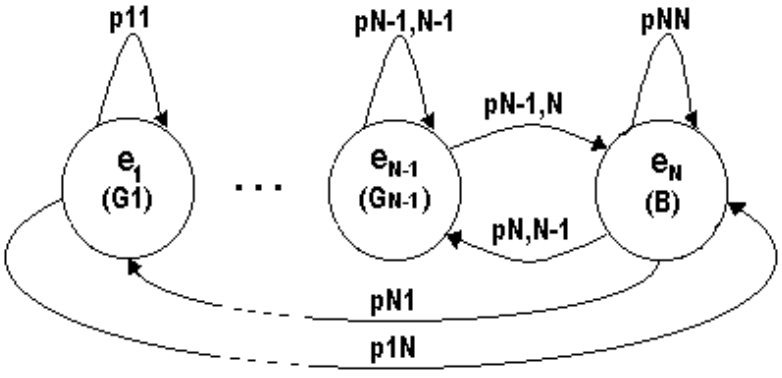

Figura 2.3 - Modelo de Fritchman-SES de *N* estados

Outros modelos de Fritchman específicos são também utilizadas em diversas aplicações. Estudos sobre modelagem de canais de satélites de baixa órbita (LEO) descritos em [4] empregam, além dos modelos de Fritchman-SES de três e quatro estados, o modelo de Fritchman (k=2, N=4). O primeiro modelo mencionado acima é utilizado em [4] para descrever a estatística de surtos de erros do sistema LEO-UHF para os ângulos de elevação de 23^o e 52^o; o segundo para os ângulos de elevação restantes, e o terceiro para canais sujeitos a interferências.

2.6. Estimação de parâmetros do HMM

A estimação de parâmetros dos HMM's a partir de uma sequência de dados observados dá origem a três problemas clássicos e de grande interesse:

i) o cálculo do estimador de máxima verossimilhança (ML) das observações condicionado aos parâmetros do HMM;

ii) a estimação da sequência de estados mais provável e

iii) a estimação ML de parâmetros. A seguir, será apresentada uma breve descrição de cada um desses três problemas [14].

2.6.1. Cálculo da probabilidade de geração da sequência observada

Com o objetivo de simplificar as notações, considere um HMM $\lambda=(P,B,\pi)$, onde, P representa sua matriz de probabilidades de transição; B a matriz das distribuições de probabilidades das emissões em cada estado e π o vetor das probabilidades iniciais de cada estado.

Seja a seqüência $O=\{ o_1, o_2, \ldots o_T \}$ uma sequência de observações. Deseja-se calcular a probabilidade das observações terem sido geradas pelo modelo λ:

$$V(O|\lambda); \tag{2.2}$$

Se este cálculo for feito de maneira direta, a sua complexidade o torna computacionalmente inviável, mesmo para seqüências de pequeno comprimento, pois envolve um número de operações da ordem N^T, que cresce exponencialmente com o tamanho da seqüência (T), sendo N o número de estados. Assim devemos

buscar um método de cálculo que reduza esta complexidade, de modo a tornar o problema computacionalmente tratável. Para tanto, faremos uso de duas variáveis auxiliares: $\alpha_t(i)$ – Variável Progressiva e $\beta_t(i)$ – Variável Regressiva.

Variável Progressiva

A Variável Progressiva $\alpha_t(i)$ é definida como a probabilidade de ocorrência da seqüência parcial de observações $O_1^t = \{o_1, o_2, \ldots o_t\}$, sendo o_t gerada pelo estado *i.* ou seja,

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, s_t = i \mid \lambda) \tag{2.3}$$

Analisando todas as possibilidades de transição para um estado *j*, podemos trivialmente, deduzir a seguinte relação recursiva:

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) p_{ij}, \qquad 1 \le j \le N, \quad 1 \le t \le T-1, \tag{2.4}$$

onde:

p_{ij} é o elemento da matriz de probabilidades de transição *P* correspondente à transição do estado *i* para o estado *j*. $b_j(o_{t+1})$ é a probabilidade de emitir a observação o_{t+1}, no instante t+1, dado que se esteja no estado *j*;

Os cálculos podem ser iniciados por:

$$\alpha_1(j) = \pi_j b_j(o_1), \qquad 1 \le j \le N\,. \tag{2.5}$$

onde:

π_j é a probabilidade inicial de se estar no estado *j*.

Através das relações acima se calcula esta variável para toda a sequência:

$$\alpha_T(i), \qquad 1 \le j \le N \tag{2.6}$$

Por fim, a probabilidade de geração da sequência observada é dada por

$$V(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.7}$$

Note-se que a complexidade deste cálculo é proporcional a N^2T, ou seja, tem crescimento linear com o tamanho da seqüência T.

Variável Regressiva

A Variável Regressiva $\beta_t(i)$ é definida como a probabilidade condicional de ocorrência da seqüência parcial de observações $O_{t+1}{}^T = \{o_{t+1}, o_{t+2}, \ldots, o_T\}$, dado o estado s_T é igual a i, ou seja,

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T \mid s_T = i, \lambda) \tag{2.8}$$

Como no caso de $\alpha_t(i)$, podemos facilmente chegar à seguinte relação recursiva para o cálculo de $\beta_t(i)$:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) p_{ij} b_j(o_{t+1}), \qquad 1 \le i \le N, \quad 1 \le t \le T-1, \tag{2.9}$$

onde:

$$\beta_T(i) = 1, \qquad 1 \le i \le N \tag{2.10}$$

Das relações acima temos:

$$\alpha_t(i)\beta_t(i) = P(O, s_t = i \mid \lambda), \qquad 1 \le i \le N, \quad 1 \le t \le T \tag{2.11}$$

Assim sendo, obtemos outro modo para o cálculo de $V(O|\lambda)$, como a seguir:

$$V(O \mid \lambda) = \sum_{i=1}^{N} P(O, s_t = i \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{2.12}$$

2.6.2. Determinação da sequência de estados mais provável

Dado um HMM $\lambda=(P,B,\pi)$ e uma sequência $O=\{ o_1, o_2, ... o_T \}$ de observações, deseja-se saber qual a seqüência de estados $S = \{s_1, s_2, ..., s_T\}$ com maior probabilidade de ter gerado as observações O. Este problema equivale à maximização da probabilidade $P(S,O|\lambda)$ e pode ser resolvido pelo *Algoritmo de Viterbi*, como se mostra a seguir:

Primeiramente definimos a variável $\delta_t(i)$, que indica a seqüência de estados mais provável de comprimento t, relativa às t primeiras observações, e que termina no estado i:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(s_1 s_2 \ldots s_{t-1}, s_t = i,\ o_1 o_2 \ldots o_t \mid \lambda) \tag{2.13}$$

Por indução pode-se mostrar que:

$$\delta_{t+1}(j) = \max_i [\delta_t(i) p_{ij}]\, b_j(o_{t+1}) \tag{2.14}$$

Assim encontramos a seqüência de estados $s_1 s_2, ..., s_t$, que satisfaz a equação (2.14) para cada t e j.

Para obter a seqüência de estados mais provável podemos seguir os seguintes passos:

a) Inicialização:

$$\delta_1(i) = \pi_i b_i(o_1) \tag{2.15}$$

$$\psi_1(i) = 0 \qquad 1 \le i \le N \tag{2.16}$$

b) Recursividade:

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) p_{ij}]\, b_j(o_t) \tag{2.17}$$

$$\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i) p_{ij}] \qquad 2 \le t \le T \quad e \quad 1 \le j \le N \tag{2.18}$$

c) Término:

$$P^* = \max_{1 \le i \le N}[\delta_T(i)] \tag{2.19}$$

$$q_T^* = \arg\max_{1 \le i \le N}[\delta_T(i)] \tag{2.20}$$

d) Seqüência de estados:

$$q_t^* = \psi_{t+1}(s_{t+1}^*) \qquad t=T-1,\ T-2,\ ...,\ 1 \tag{2.21}$$

2.6.3. Estimação ML de Parâmetros

Dada uma seqüência $O=\{ o_1, o_2, ... o_T \}$ de observações, deseja-se ajustar os parâmetros do modelo HMM $\lambda=(A,B,\pi)$, de modo a maximizar:

$$V_{tot} = P(O/\lambda) \tag{2.22}$$

Não existe nenhuma maneira conhecida de se resolver este problema analiticamente, ou seja, encontrar os parâmetros do HMM λ que maximize V_{tot} [14]. No entanto podemos encontrar parâmetros que maximizem localmente V_{tot}, através do uso do *Algoritmo de Baum-Welch,* o qual está descrito a seguir.

2.6.4.
Algoritmo de Baum-Welch

Primeiramente definimos a função auxiliar *Q*:

$$Q(\lambda', \lambda) = \sum_{q} P(O, S \mid \lambda') \log P(O, S \mid \lambda) \tag{2.23}$$

a qual deverá ser maximizada em λ a fim de atualizar λ' no sentido de aumentar *V(O|λ)*, pois mostra-se em [14] que:

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(O \mid \lambda) \geq P(O \mid \lambda') \tag{2.24}$$

Uma característica muito importante deste algoritmo é a garantia da convergência para um ponto de máximo local [14].

Definem-se ainda, mais duas variáveis auxiliares em adição às *variáveis, progressiva e regressiva,* definidas anteriormente. A primeira delas é dada pela probabilidade de se estar no estado *i* em *t* e no estado *j* em *t+1* dada a seqüência de observações, ou seja:

$$\xi_t(i, j) = P(s_t = i, s_{t+1} = j \mid O, \lambda) \tag{2.25}$$

ou ainda:

$$\xi_t(i, j) = \frac{P(s_t = i, s_{t+1} = j, O \mid \lambda)}{P(O \mid \lambda)} \tag{2.26}$$

Em função das variáveis progressiva e regressiva, temos:

$$\xi_t(i, j) = \frac{\alpha_t(i) p_{ij} \beta_{t+1}(j) b_j(o_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) p_{ij} \beta_{t+1}(j) b_j(o_{t+1})} \quad . \tag{2.27}$$

A segunda variável auxiliar é dada pela probabilidade de se estar no estado *i* em *t*, dada a seqüência de observações, ou seja:

$$\gamma_t(i) = P(s_t = i \mid O, \lambda) \tag{2.28}$$

Em função das variáveis progressiva e regressiva, podemos obter:

$$\gamma_t = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \right] \quad . \tag{2.29}$$

A relação entre $\gamma(i)$ e $\xi(i,j)$ é dada por:

$$\gamma_t = \sum_{j=1}^{N} \xi_t(i,j), \qquad 1 \le i \le N, \quad 1 \le t \le T \tag{2.30}$$

Os parâmetros de um HMM são atualizados a cada iteração, no sentido de maximizar a probabilidade $P(O|\lambda)$, supondo um modelo inicial $\lambda_1 = (P_1, B_1, \pi_1)$.

Inicialmente calculamos os *'α's* utilizando as equações (2.4) e (2.5), e os *'β's* utilizando as equações (2.9) e (2.10). Em seguida calculamos ξ e γ através das equações (2.27) e (2.29), respectivamente. Finalmente atualizamos os parâmetros do HMM seguindo as equações a seguir, conhecidas como *fórmulas de reestimação*:

$$\overline{\pi}_i = \gamma_1(i), \qquad 1 \le i \le N \tag{2.31}$$

$$\overline{p}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \qquad 1 \le i \le N, \quad 1 \le j \le N \tag{2.32}$$

$$\overline{b}_j(v_k) = \frac{\sum_{\substack{t=1 \\ O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \qquad 1 \le j \le N, \quad 1 \le k \le m \tag{2.33}$$

onde:

v_k pertence ao alfabeto de símbolos $\{v_1, v_2, ..., v_m\}$

Em função das variáveis progressiva e regressiva, temos:

$$\overline{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N}\alpha_T(j)} \tag{2.34}$$

$$\overline{p}_{ij} = \frac{\sum_{t=1}^{T}\alpha_{t-1}(i)p_{ij}b_j(o_t)\beta_t(j)}{\sum_{t=1}^{T}\alpha_{t-1}(i)\beta_{t-1}(i)} \tag{2.35}$$

$$\overline{b}_i(v_k) = \frac{\sum_{t=1}^{T}\alpha_t(i)\beta_t(i)\delta(o_t, v_k)}{\sum_{t=1}^{T}\alpha_t(i)\beta_t(i)} \tag{2.36}$$

sendo:

$$\delta(o_t, v_k) = \begin{cases} 1 & se \quad o_t = v_k \\ 0 & se \quad o_t \neq v_k \end{cases}$$

Problemas relativos à implementação do algoritmo

De posse das fórmulas de reestimação do algoritmo de Baum-Welch, verifica-se que ainda existem vários detalhes em nível de implementação, para que este possa ser executado em computador. Serão comentados em seguida os mais significativos para este trabalho, acompanhado da solução adotada.

Limite de ordem de grandeza inferior

As operações recursivas realizadas durante a estimação dos parâmetros de um HMM fazem com que estes tendam a zero com o número de iterações. Isto acontece devido ao fato de que as variáveis $\alpha(i)$ e $\beta(i)$ são produzidas por um cálculo recursivo e que cada iteração envolve produtos de probabilidades, como observado nas equações (2.4) e (2.9).

Assim, mesmo que a seqüência não seja muito grande, o limite de precisão de qualquer computador é rapidamente atingido ao se executar este algoritmo.

Para combater este problema utilizaremos uma técnica de ponderar estas variáveis, ou seja, multiplicá-las por um fator de escala 'c_t' dependente do tempo, mas que não dependa de 'i' [14], de forma que os valores destas variáveis sejam mantidos dentro dos limites de precisão do computador, quando o valor de t varia de 1 até T. Os cálculos deste procedimento estão descritos a seguir:

- para t=1, calcula-se $\alpha_1(i)$ como na equação (2.4);

- cálculo do fator c_1:

$$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_1(i)} \quad \text{e} \quad \hat{\alpha}_1(i) = c_1 \alpha_1(i) \tag{2.37}$$

onde, $\hat{\alpha}_1(i)$ representa o $\alpha_1(i)$ ponderado pelo fator c_1;

- cálculo recursivo de $\hat{\alpha}_1(i)$:

$$\hat{\hat{\alpha}}_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) p_{ji} b_i(o_t); \tag{2.38}$$

- após calculado, $\hat{\hat{\alpha}}_t(i)$ deve ser aplicado nas equações (2.39), para o cálculo dos novos valores de c_t e $\hat{\alpha}_t(i)$, que deverão ser novamente aplicados na equação (2.38), e assim sucessivamente, sendo:

$$\hat{\alpha}_t(i) = c_t \hat{\hat{\alpha}}_t(i) \quad \text{e} \quad c_t = \frac{1}{\sum_{i=1}^{N} \hat{\hat{\alpha}}_t(i)} \quad . \tag{2.39}$$

Como $\beta_t(i)$ possui a mesma ordem de grandeza de $\alpha_t(i)$, utilizaremos o mesmo fator de escala c_t para ponderar estas variáveis, que também serão mantidas em limites razoáveis para o cálculo computacional. Assim, temos:

$$\hat{\beta}_t(i) = c_t \beta_t(i) \ . \tag{2.40}$$

Sequência de observações de tamanho insuficiente

Outro problema relativo à implementação deste algoritmo é o efeito causado por uma pequena seqüência de observação.

No caso de $b_i(v_k)$, que representa a probabilidade de ocorrer a observação v_k no estado i, observa-se que o numerador da equação de reestimação (2.36) é uma soma cujos termos em que a observação o_t é diferente do símbolo v_k, são nulos.

Se esta probabilidade é pequena e a amostra da seqüência utilizada na reestimação for também pequena, pode não ocorrer nenhuma vez o evento o_t igual a v_k no estado i. Com isso, o valor de $b_i(k)$ reestimado e os seguintes serão nulos, levando os cálculos a valores irreais.

Possíveis soluções seriam aumentar o tamanho da seqüência observada, diminuir o número de estados, ou diminuir o número de símbolos do modelo utilizado, o que geralmente não pode ser feito. Uma solução prática seria estipular limiares mínimos para os parâmetros do modelo, como no exemplo a seguir [14]:

$$b_i(k) = \begin{cases} b_i(k), & b_i(k) \geq \delta_b; \\ \delta_b, & b_i(k) < \delta_b; \end{cases} \tag{2.41}$$

onde δ_b é o valor mínimo fixado.

Estimação inicial dos parâmetros

As equações de reestimação produzem valores dos parâmetros de um HMM, correspondentes à convergência para o máximo global ou para qualquer máximo local da função verossimilhança, se houver. Não existe nenhuma maneira simples e direta para resolver este problema.

2.7. Ferramentas de Otimização

Neste trabalho foram utilizadas duas ferramentas de otimização para a estimação dos parâmetros, a otimização clássica e *Particle Swarm Optimization*

(PSO). Ambas puderam ser empregadas, devido ao modelo proposto ter possibilitado a dedução de uma função de verossimilhança, matematicamente simples.

A otimização clássica, foi inicialmente empregada para a estimação ML dos parâmetros do modelo proposto. Observou-se a existência de diversos máximos locais na função de verossimilhança, levando à necessidade de uma apropriada inicialização dos parâmetros que aumentasse as chances de convergência para o máximo global da função de verossimilhança.

Para tanto se adotou o PSO, que apesar de ser um método que consome um tempo de processamento relativamente alto para ser empregado como método principal de otimização, se mostrou bastante eficiente na estimação de valores iniciais para a estimação ML dos parâmetros do modelo proposto. Um resumo da teoria de otimização clássica encontra-se no Apêndice B e os detalhes sobre o emprego deste método neste trabalho serão abordados no Capítulo 4.

2.7.1. Particle Swarm Optimization (PSO)

O PSO é um método de otimização global, que difere de outros métodos conhecidos como Algoritmos Evolucionários (AE) [30]. Como nos AE, um conjunto de possíveis soluções é utilizado para sondar o espaço de pesquisa adequado ao problema, porém nenhum tipo de operação baseada em rotinas de evolução é aplicado ao conjunto de soluções, para atualizá-lo.

No caso do PSO, fazendo-se uma analogia no espaço tridimensional, é como se um "enxame" de partículas se locomovessem neste espaço, em busca de um posicionamento adequado a cada uma, que gere a solução desejada. Cada elemento, aqui chamado *partícula*, do conjunto de soluções chamado *enxame*, atualiza sua trajetória considerando a sua melhor posição anterior e a melhor posição anterior alcançada por qualquer membro de sua *vizinhança* topológica [30].

Deste modo, sendo todo o *enxame* considerado como a *vizinhança*, ocorre o compartilhamento global de informações e as partículas se valem das descobertas e experiência de todas as outras, durante a busca Muitas variações da técnica de PSO tem sido propostas [30].

Para formalização de um algoritmo PSO b considerando o espaço de pesquisa como *D*-dimensional, a *i-ésima* partícula do enxame é representada pelo vetor *D*-dimensional $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, e a partícula correspondente ao maior valor da função que se deseja maximizar de todo o enxame indicada pelo índice *g*. A melhor posição conseguida e velocidade, correspondentes a partícula X_i são, respectivamente representadas e armazenadas pelos vetores:

$$P_i = (p_{i1}, p_{i2}, ..., p_{iD}) \text{ e } V_i = (v_{i1}, v_{i2}, ..., v_{iD}) \tag{2.42}$$

As partículas se relacionam segundo as seguintes equações:

$$V_i^{n+1} = wV_i^n + c_1 r_{i1}^n (P_i^n - X_i^n) + c_2 r_{i2}^n (P_g^n - X_i^n) \tag{2.43}$$

$$X_i^{n+1} = X_i^n + \chi V_i^{n+1}, \qquad i = 1, 2, ..., N \tag{2.44}$$

onde:

- os super-escritos indicam a iteração;
- χ é um fator de constrição para controlar a velocidade;
- *w* é d *peso inercial* e sua finalidade é descrita a seguir- c_1 e c_2 são constantes positivas chamadas de parâmetro cognitivo e social respectivamente, cujo emprego é descrito abaixo
- r_{i1}^n *e* r_{i2}^n são dois números uniformemente distribuídos no intervalo [0, 1], cuja finalidade é comentada abaixo.

A equação (2.43) é usada para calcular a nova velocidade da *i-ésima* partícula em cada iteração, sendo o primeiro termo desta equação, wV_i^n, representa a velocidade anterior da partícula multiplicadas pelo *peso inercial w.* O segundo termo $(P_i^n - X_i^n)$ representa a distância entre a melhor posição anterior e atual de cada partícula. O terceiro termo $(P_g^n - X_i^n)$, é a distância entre a melhor posição entre todas as partículas do enxame e a posição atual da *i-ésima* partícula.

Os parâmetros $c_1 r_{i1}^n$ *e* $c_2 r_{i2}^n$ introduzem aleatoriedade, a qual deixa a técnica menos previsível, porém mais flexível [30].

A equação (2.44) calcula a nova posição da *i-ésima* partícula, adicionando a sua nova velocidade com a posição atual.

O desempenho de cada partícula é medida de acordo com a função objetivo em questão.

O *peso inercial w* é considerado fundamental para a convergência do PSO, pois esse é empregado para controlar o impacto do histórico das velocidades sobre a velocidade atual. Assim, o parâmetro *w* regula a troca entre as habilidades de exploração global e local do *enxame*, sendo que grandes valores do *peso inercial* facilitam a exploração global, enquanto pequenos valores facilitam a exploração local, o que pode ser visto como um ajuste fino da pesquisa na região atual. Assim sendo, a escolha deste parâmetro resulta diretamente na taxa de convergência para o ponto de máximo desejado.

A população inicial de partículas e suas respectivas velocidades podem ser geradas randomicamente ou por um gerador de sequências de Sobol [30], que assegura que os vetores *D*-dimensionais serão uniformemente distribuídos no espaço de pesquisa. O conceito de recombinação está relacionado ao movimento estocástico de cada partícula em direção à sua melhor posição anterior, assim como em direção a melhor posição global de todo enxame ou à melhor posição de sua vizinhança, dependendo de qual das variações de PSO é utilizada. Além de tudo.

A técnica de PSO tem se mostrado eficaz para resolver problemas de otimização global, em ambientes continuamente variantes ou em treinamento de redes neurais e ainda em problemas de otimização multi-objetivo [30].

2.7.2. Técnica “Branch and Bound” (BB)

A técnica *Branch and Bound* é amplamente utilizada para a resolução de problemas de PSO. Nesta técnica, a região pertinente ao problema é particionada em várias sub-regiões, isso é chamado de *branch*. Ao longo destas sub-regiões, limites (*bounds*) inferiores e superiores para os valores da função podem ser

determinados. A técnica BB pode ser esquematizada em forma de algoritmo, como se segue [30]:

1. Inicialmente, define-se uma região $M_0 \supset S$ e seu particionamento em um número finito de subconjuntos M_i, i =1, 2, . . ., m, onde S é a região pertinente ao problema.

2. Para cada subconjunto M_i, determinam-se os limites, inferior e superior da função, $\beta(M_i)$ e $\alpha(M_i)$, respectivamente, satisfazendo a seguintes inequações:

$$\beta(M_i) \leq \inf\left(f(M_i \cap S\right) \leq \alpha(M_i) \tag{2.45}$$

onde f é a função objetivo em consideração

Define-se os limites globais, inferior e superior, pelas equações (2.46) e (2.47), respectivamente.

$$\beta = \min_{i=1,2,\cdots,m} \beta(M_i) \tag{2.46}$$

$$\alpha = \min_{i=1,2,\cdots,m} \alpha(M_i) \tag{2.47}$$

sendo:

$$\beta \leq \min f(S) \leq \alpha \tag{2.48}$$

3. Define-se o critério de parada como:

$$\alpha = \beta \quad \text{ou} \quad \alpha - \beta \leq \varepsilon, \quad \varepsilon > 0$$

4. Caso contrário, escolhem-se alguns dos subconjuntos M_i a fim de particioná-los e se obter um particionamento mais refinado de M_0. Determinam-se limites sobre os novos elementos da nova partição, e repete-se o procedimento.

Uma vantagem da técnica BB é que durante o processo de iteração, pode-se normalmente excluir subconjuntos de *S*, nos quais o mínimo de *f* não pode ser alcançado. Esta técnica tem sido aplicada com sucesso em problemas de programação inteira [30].