Introdução aos Modelos Escondidos de Markov (HMM)

Luiz Eduardo Soares de Oliveira, Marisa Emika Morita

Pontifícia Universidade Católica do Paraná – PUC-Pr.
PPGIA – Programa de Pós-Graduação em Informática Aplicada
Rua Imaculada Conceição, 1155 – 80215-901 – Curitiba, Pr. Brasil

{soares,marisa}@ppgia.pucpr.br

Resumo

Esse artigo apresenta uma introdução aos Modelos Escondidos de Markov (HMM). Uma apresentação formal de todos os elementos do HMM é mostrada, bem como os três problemas básicos do HMM que devem ser resolvidos para que o modelo possa ser utilizado em aplicações do mundo real, bem como a solução para os mesmos. Algumas limitações e vantagens também são apresentadas.

1. Introdução

Os processos no mundo real geralmente produzem sinais (seqüência de observações). Os sinais podem ser discretos (caracteres de um alfabeto finito, vetores quantizados de um alfabeto ou *codebook*) ou contínuo (exemplo de vozes, medidas de temperatura, música, etc). A fonte do sinal pode ser estacionária (propriedades estatísticas não variam com o tempo) ou não estacionária (propriedades estatísticas variam sobre o tempo). Além disso, os sinais podem ser puro (vem somente de uma fonte restrita) ou não puro (ruído, outras fontes de sinais) [13].

O objetivo nesse caso é caracterizar os sinais do mundo real em termos de modelos de sinais, pois:

- Um modelo de sinal pode prover a base para uma descrição teórica de um sistema de processamento de sinal, o qual pode ser usado para processar o sinal, bem como prover uma saída desejada;
- Os modelos de sinais são capazes de nos levar a aprender bastante sobre a fonte do sinal (possibilidade de simular a fonte);
- Os modelos de sinais trabalham extremamente bem na prática, e nos permite realizar importantes sistemas práticos. Ex: sistema de reconhecimento.

Os sinais podem ser modelados utilizando-se as classes deterministas ou estatísticas. Os modelos deterministas geralmente exploram algumas propriedades específicas do sinal. Tudo que é requerido é determinar (estimar) valores dos parâmetros do modelo do sinal (amplitude, freqüência ...). Os modelos estatísticos tentam caracterizar somente propriedades estatísticas dos sinais (processos de *Gauss*, *Poison*, *Markov*, HMM entre outros).

O principal interesse nesse artigo é descrever sobre os modelos estatísticos, especificamente sobre HMM (*Hidden Markov Model*).

HMM foi descrito pela primeira vez ao final dos anos 60 e início dos anos 70 [2] [3] [5]. A aplicação desses modelos em reconhecimento de palavras começou a ser utilizada em meados dos anos 70 [1].

Durante os últimos 15 anos, HMM tem sido largamente aplicado em várias áreas, incluindo reconhecimento de voz [11] [14], modelagem de linguagens [9], reconhecimento de palavras manuscritas [10] [17] [18], verificação on-line de assinatura [19], aprendizado de ações humanas [20], detecção de falhas em sistemas dinâmicos [16] e reconhecimento de *moving light displays* [7].

HMM é um processo duplamente estocástico, com um processo estocástico não visível, o qual não é observável (daí o nome de escondido), mas que pode ser observado através de outro processo estocástico que produz a seqüência de observações [13].

Os processos escondidos consistem de um conjunto de estados conectados por transições com probabilidades (autômato finito), enquanto que os processos observáveis (não escondidos) consistem de um conjunto de saídas ou observações, cada qual podem ser emitido por cada estado de acordo com alguma saída da função de densidade de probabilidade (fdp).

Dependendo de sua fdp, várias classes de HMM's podem ser distinguidas, a seguir:

- Discreta – observação discreta por natureza ou discretizada por vetor quantizado produzindo assim um alfabeto ou *codebook*.
- Contínuo – observação contínua, com sua fdp contínua usualmente aproximada para uma mistura de distribuição normal.
- Semi-contínuo – entre o discreto e o contínuo (híbrido).

HMM tem se tornado recentemente, a abordagem predominante para o reconhecimento da fala. Esses modelos estocásticos tem sido mostrados particularmente bem adaptados para caracterizar a variabilidade envolvida em sinais que variam no tempo. A maior vantagem do HMM situa-se na sua natureza probabilística, apropriada para sinais corrompidos por ruídos tal como a fala ou escrita, e na sua fundação teórica devido a existência de algoritmos poderosos para ajustar automaticamente os parâmetros do modelo através de procedimentos iterativos [18].

Na seção 2 serão apresentados os elementos do HMM, na seção 3 as suas principais arquiteturas, na 4 serão discutidos os três problemas básicos do HMM, na seção 5 o problema de reconhecimento, na seção 6 algumas limitações e vantagens do HMM.

2. Elementos de um HMM

Um HMM para as observações de símbolos discretos é caracterizado por:

- *N*, o número de estados no modelo. Os estados individuais são rotulados como {1, 2, ...*N*} e o estado no tempo *t* como q_t .
- *M*, o número de símbolos de observações distintos por estado, por exemplo, o tamanho do alfabeto discreto. Os símbolos individuais são denotados como $V = \{v_1, v_2, ..., v_M\}$.
- A distribuição de probabilidade da transição do estado $A = \{a_{ij}\}$ onde:

$$a_{ij} = P[q_{t+1} = j \mid q_t = i],\ 1 \le i, j \le N \qquad (1)$$

Para o caso especial onde qualquer estado pode alcançar qualquer outro estado em

uma simples etapa, tem-se $a_{ij} > 0$ para todo *i, j*. Para outros tipos de HMM, podem-se ter $a_{ij} = 0$ para um ou mais pares (i, j).

- A distribuição de probabilidade de símbolos de observações, $B = \{b_j(k)\}$ define a distribuição de símbolos no estado *j*, *j* = 1, 2, ..., *N*, onde:

$$b_j(k) = P[O_t = v_k \mid q_t = j],\ 1 \le i \le M \qquad (2)$$

- A distribuição do estado inicial $\pi = \{\pi_i\}$, onde:

$$\pi_i = P[q_1 = i],\ 1 \le i \le N \qquad (3)$$

Pode-se observar que uma completa especificação de um HMM requer especificação de dois parâmetros do modelo, *N* e *M*, especificação da observação de símbolos, e a especificação de três conjuntos de medidas de probabilidade *A*, *B* e π. Por conveniência será utilizada a notação compacta $\lambda = (A, B, \pi)$ para indicar o completo conjunto de parâmetros do modelo. Este conjunto de parâmetros, naturalmente, define a medida de probabilidade para *O*, por exemplo, $P(O \mid \lambda)$, o qual será discutido nas seções seguintes.

3. Principais arquiteturas de HMM

A estrutura do modelo e o número de estados escolhidos são fatores fundamentais para a determinação do HMM ótimo [17].

Em geral existem dois tipos de estruturas para os HMM, a seguir:

- Modelos sem restrições ou ergóticos;
- Modelos esquerda-direita;

Nos modelos ergóticos (Figura 1(a)) todas as transições possíveis entre os estados da cadeia são autorizados. Isto é possível se não restringir-se nenhum dos valores de a_{ij} em ser nulo.

Os modelos sequenciais e paralelos fazem parte dos modelos esquerda-direita. Para esses modelos, a matriz de transição entre estados é triangular superior. Os modelos sequenciais (Figura 1(b)) funcionam segundo uma evolução em série do modelo através de seus estados, mesmo que qualquer um desses estados possam ser saltados no curso do processo. Para os modelos paralelos (Figura 1(c)) muitos caminhos através da rede de *Markov* são permitidos, sabendo que cada um desses caminhos salte um ou vários estados do modelo.

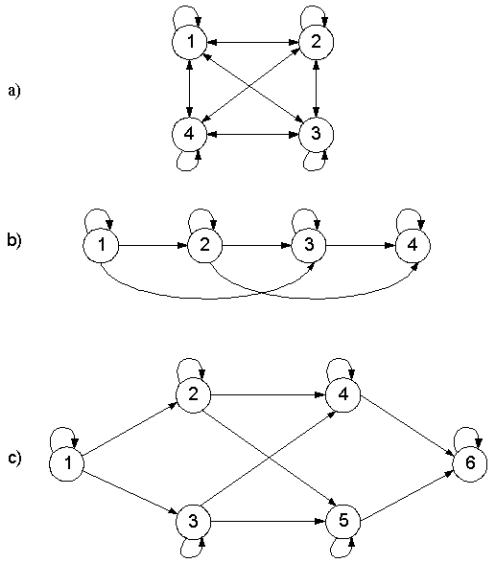


Figura 1: Estrutura dos modelos de *Markov*: (a) modelo sem restrições, (b) modelo sequencial, (c) modelo paralelo.

Cada uma das estruturas dos modelos apresentadas na Figura 1, podem ser generalizadas por incluir um número de estados arbitrário. Entretanto, o número de parâmetros a serem estimados em um modelo de *Markov* é da

ordem N^2 para a matriz *A*, mais *NM* para a matriz *B*. Assim, se *N* é muito grande, uma determinação coerente e precisa das matrizes A e B ótimas vem a ser muito difícil de realizar por uma base de aprendizagem de tamanho fixado.

Não existe meios teóricos para determinar de maneira precisa o número de estados necessários no modelo devido ao estados não estarem sempre fisicamente ligados aos fenômenos observáveis.

4. Os três problemas básicos do HMM

Existem três problemas básicos que devem ser resolvidos para que modelo possa ser utilizado em aplicações do mundo real [15]. Esses problemas são os seguintes:

Problema 1 (problema de avaliação): Dado a seqüência de observação $O = (o_1, o_2, ..., o_T)$, e o modelo $\lambda = (A, B, \pi)$, como calcular eficientemente $P(O \mid \lambda)$, a probabilidade da seqüência de observações, dado o modelo ?

Problema 2 (problema da busca da melhor seqüência de estados): Dado a seqüência de observações $O = (o_1, o_2, ..., o_T)$, e o modelo λ, como escolher uma seqüência de estados correspondente $Q = (q_1, q_2, ..., q_T)$?

Problema 3 (problema de treinamento): Como ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$ para maximizar $P(O \mid \lambda)$?

4.1. Solução do problema de avaliação

O problema 1 é o problema de avaliação, isto é, dado um modelo e uma seqüência de observações, como calcular a probabilidade que a seqüência observada seja produzida pelo modelo ? Este problema pode ser visto como um dado modelo corresponde a uma dada seqüência de observações. Isso é extremamente útil. Por exemplo, considerando o caso no qual deve-se tentar escolher um modelo entre vários, a solução desse problema permite escolher o modelo que melhor corresponde as observações.

A maneira mais direta de calcular a probabilidade de uma seqüência de observações $O = (o_1, o_2, ..., o_T)$, dado o modelo $\lambda = (A, B, \pi)$ é através da enumeração de todas as possíveis seqüências de estados de tamanho *T* (o número de observações), pela seguinte expressão:

$$P(O \mid \lambda) = \sum_{q_1, q_2, ..., q_T} \begin{matrix} \pi_{q_1} b_{q_2}(o_1) a_{q_1 q_2} \\ b_{q_2}(o_2) .. a_{q_{T-1} q_T} b_{q_T}(o_T) \end{matrix} \quad (4)$$

A expressão acima envolve uma ordem de $2T - N^T$ cálculos, para cada *t* = 1, 2, ..., *T*, existem *N* possíveis estados que podem ser alcançados, ou seja, existem N^T possíveis seqüências de estados, e para cada qual $2T$ cálculos são necessários (para ser preciso é necessário $(2T-1)N^T + N^T - 1$). Este cálculo é computacionalmente inviável mesmo para pequenos valores de *N* e *T*, por exemplo, para *N* = 5 (estados), *T* = 100 (observações) existem na ordem de 10^{72} cálculos. Esse problema pode ser resolvido de maneira mais eficiente utilizando-se dos procedimentos *forward-backward* [3] [4].

4.1.1. Procedimento *Forward- Backward*

A variável *forward* $\alpha_t(i)$ é a probabilidade da seqüência de observações

parciais $o_1, o_2, ..., o_t$ (até o tempo *t*) e estado *i* no tempo *t*, dado o λ, onde:

$$\alpha_t(i) = P(o_1, o_2, ..., o_t, q_t = i \mid \lambda) \quad (5)$$

$\alpha_t(i)$ pode ser resolvido recursivamente , utilizando as seguintes expressões:

1. Inicialização

$$\alpha_1(i) = \pi_i b_i(o_1), \ 1 \le i \le N \quad (6)$$

2. Indução

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \quad (7)$$

$1 \le t \le T-1$ e $1 \le j \le N$

3. Terminação

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \quad (8)$$

Para o mesmo exemplo citado acima, *N*=5 (estados), *T* = 100 (observações) são necessários 3.000 cálculos através do método *forward* , contra 10^{72} do cálculo direto [15].

De maneira similar , a variável $\beta_t(i)$ é definida como a probabilidade da seqüência de observações parciais de *t+1* para o final, dado estado *i* no tempo *t* e o modelo λ, onde:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T \mid q_t = i, \lambda) \quad (9)$$

$\beta_t(i)$ pode ser resolvido recursivamente , utilizando as seguintes expressões:

1. Inicialização

$$\beta_T(i) = 1, \ 1 \le i \le N \quad (10)$$

2. Indução

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad (11)$$

$t = T-1, T-2, \ldots, 1 \ 1 \le i \le N$

3. Terminação

$$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) \quad (12)$$

Somente uma das duas variáveis α ou β é necessária para o problema de avaliação. Entretanto, ambas foram introduzidas aqui, pois as mesmas são utilizadas no problema do treinamento (Problema 3).

4.2. Solução do problema da busca da melhor seqüência de estados

O problema 2 procura descobrir a parte escondida do modelo, ou seja, encontrar a seqüência de estados *correta*. Este problema geralmente é resolvido usando um procedimento próximo ao ótimo, o algoritmo de *Viterbi* [8] [12], que procura a melhor seqüência de estados $Q = (q_1, q_2, ..., q_T)$ para uma dada seqüência de observações $O = (o_1, o_2, ..., o_T)$.

4.2.1. Algoritmo de Viterbi

Para encontrar a melhor seqüência de estados, $Q = (q_1, q_2, \ldots, q_T)$, para uma dada seqüência de observações $O = (o_1, o_2, ..., o_T)$ define-se a quantidade:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, \ q_t = i, o_1 o_2 \ldots o_t \mid \lambda] \quad (13)$$

na qual, $\delta_t(i)$ é o melhor resultado (probabilidade mais alta) ao longo de um

caminho simples no tempo *t*, o qual leva em consideração as *t* primeiras observações e termina no estado *i*. Por indução tem-se:

$$\delta_{t+1}(j) = [\max_i \ \delta_t(i)a_{ij}]\, b_j(o_{t+1}) \qquad (14)$$

Para recuperar a seqüência de estados, é necessário manter os argumentos que maximizam a expressão anterior, para cada *t* e *i* através do *array* $\psi_t(j)$. O procedimento completo para encontrar a melhor seqüência de estados é a seguinte:

1. Inicialização

$$\delta_t(i) = \pi_i b_i(o_1),\ 1 \le i \le N \qquad (15a)$$

$$\psi_1(i) = 0 \qquad (15b)$$

2. Recursão

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i)a_{ij}]b_j(o_t), \qquad (16a)$$

$$2 \le t \le T \text{ e } 1 \le j \le N$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i)a_{ij}], \qquad (16b)$$

$$2 \le t \le T \text{ e } 1 \le j \le N$$

3. Terminação

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \qquad (17a)$$

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)] \qquad (17b)$$

4. Caminho (seqüência de estados) *backtracking*

$$q_t^* = \psi_{t+1}(q_{t+1}^*),\ t = T\text{-}1, T\text{-}2, ..., 1. \qquad (18)$$

Com exceção da etapa de *backtracking* , o algoritmo de *Viterbi* e o procedimento de *forward* tem basicamente a mesma implementação. A única diferença entre eles é que o somatório do procedimento *forward* é trocado pela maximização no algoritmo de *Viterbi*.

4.3. Solução do problema de treinamento

O terceiro e mais difícil, é determinar um método para ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$ para satisfazer um certo critério de otimização. A seqüência de observações utilizada para ajustar os parâmetros do modelo é chamada de seqüência de treinamento porque é utilizada para treinar o HMM. Não existe uma maneira conhecida de resolver analiticamente o conjunto de parâmetros do modelo que maximiza a probabilidade da seqüência de observações de uma maneira fechada. Entretanto, pode-se escolher $\lambda = (A, B, \pi)$ tal que sua probabilidade, $P(O \mid \lambda)$, é localmente maximizada usando um procedimento iterativo tal como o método de *Baum-Welch* (também conhecido como o método EM (*expectation-maximization*)), ou técnicas de gradiente [13]. Nesta seção será apresentado um procedimento iterativo, baseado primeiramente no trabalho clássico de *Baum*, para escolher a probabilidade máxima dos parâmetros do modelo.

Para descrever o procedimento de reestimação (atualização iterativa e melhorias) dos parâmetros do HMM, é primeiro definido $\xi_t(i, j)$, a probabilidade de estando no estado *i* no tempo *j*, e estado *j* no tempo *t+1*, dado o modelo e a seqüência de observações, onde:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j \mid O, \lambda) \qquad (19)$$

Através das definições das variáveis de *forward* e *backward* , pode-se escrever $\xi_t^u(i, j)$ na forma:

$$\xi_t^u(i, j) = \frac{\alpha_t^u(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}^u(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t^u(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}^u(j)} \quad (20)$$

onde *u* refere-se a imagem. $\gamma_t(i)$ pode ser definida como a probabilidade de estando no estado *i* no tempo *t*, dado toda a seqüência de observações e o modelo.

$$\gamma_t(i) = P(q_t = i \mid O, \lambda) \quad (21)$$

Da mesma forma $\overset{u}{\gamma}_t(i)$ pode ser escrita em função das variáveis *forward* e *backward* na forma:

$$\lambda_t^u(i) = \frac{\beta_t^u(i)\alpha_t^u(i)}{P^u(O \mid \lambda)} \quad (22)$$

Portanto, os parâmetros π , *A* e *B* do HMM de reestimação utilizando-se *U* imagens no treinamento são escritas na seguinte maneira:

$$\overline{\pi_i} = \frac{\sum_{u=1}^{U} \gamma_1^u(i)}{U} \quad (23)$$

$$\overline{a_{ij}} = \frac{\sum_{u=1}^{U}\sum_{t=1}^{T-1} \xi_t^u(i, j)}{\sum_{u=1}^{U}\sum_{t=1}^{T-1} \gamma_t^u(i)} \quad (24)$$

$$\overline{b_j}(k) = \frac{\sum_{u=1}^{U}\sum_{t=1}^{T} \gamma_t(j)\, \delta\left(O_t^u, v_k\right)}{\sum_{t=1}^{U}\sum_{t=1}^{T} \gamma_t^u(j)} \quad (25)$$

Se um modelo é definido como $\lambda = (A, B, \pi)$ e seu modelo reestimado é definido como $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ através das expressões acima, *Baum* provou que se o modelo inicial λ está no ponto crítico de função de probabilidade, neste caso $\overline{\lambda} = \lambda$ ou o modelo $\overline{\lambda}$ é mais promissor que o modelo λ no sentido $P(O \mid \overline{\lambda}) > P(O \mid \lambda)$, ou seja, foi encontrado um novo modelo $\overline{\lambda}$ na qual a seqüência de observações é mais provável para ter sido produzida.

Baseado nesse procedimento é utilizado iterativamente $\overline{\lambda}$ no lugar de λ e repetido o cálculo de reestimação, assim pode-se melhorar a probabilidade de *O* sendo observado do modelo até que algum ponto de limitação seja alcançado. O resultado final deste procedimento de reestimação é uma probabilidade máxima estimada do HMM. O algoritmo de *forward-backward* leva para o local máximo somente, e que em muitos problemas de interesse, a função de probabilidade é muito complexa e tem muitos locais máximos.

5. Problema de Reconhecimento

Dado uma seqüência de observações $O = (o_1, o_2, ..., o_T)$ extraída de uma palavra desconhecida, o problema consiste em encontrar a classe (palavra) que corresponde a melhor probabilidade de produzir a seqüência de observações $O = (o_1, o_2, ..., o_T)$. Para resolver esse problema, existem duas soluções: Modelo Discriminante e Modelo de Caminho Discriminante [6].

O primeiro, conhecido como Modelo Discriminante, consiste na construção de um modelo para cada classe ou palavra afim de escolher a classe do modelo que leva a melhor probabilidade de produzir a seqüência de observações $O = (o_1, o_2, ..., o_T)$. A probabilidade pode ser calculada através dos algoritmos de *Viterbi* ou *Forward.*

O segundo, conhecido como Modelo de Caminho Discriminante, consiste na construção de um único modelo no qual cada caminho corresponde a uma seqüência de letras. Então o algoritmo de *Viterbi* permite encontrar a seqüência de letras (palavra) que leva a melhor probabilidade de produzir a seqüência de observações $O = (o_1, o_2, ..., o_T)$.

6. Limitações e principais vantagens do HMM

Embora o uso da tecnologia de HMM tem contribuído grandemente para avanços recentes no reconhecimento da fala, há algumas limitações herdadas do tipo de modelo estatístico para a fala. A maior limitação é a hipótese que as observações sucessivas (*frames* da fala) são independentes, e entretanto a probabilidade da seqüência de observações $P(o_1, o_2, ..., o_T)$ pode ser escrita como o produto da probabilidade de observações individuais [13], por exemplo:

$$P(o_1, o_2, ..., o_T) = \prod_{i=1}^{T} P(O_i) \qquad (26)$$

Além disso, os processos sequenciais reais são não como os Markovianos. Esta limitação é de todos os processos e não somente de *Markov*.

Quanto as vantagens na utilização de HMM, são muitas, em relação as suas limitações, a seguir:

HMM apresenta uma rica representação, ou seja, as probabilidades de saída $(B = \{b_j(k)\})$ representam a variabilidade dos *frames* ou segmentos (distorções de caracteres na escrita) e as probabilidades de transição $(A = \{a_{ij}\})$ representam o relacionamento temporal entre os *frames* ou segmentos.

HMM possui uma base matemática sólida, devido a garantia de convergência para um ótimo ponto. A existência de um treinamento eficiente que automaticamente otimiza os parâmetros dos dados e a decodificação de técnicas que descrevem uma *string* de entrada em termos da melhor seqüência de estados.

HMM requer supervisão mínima, pois não necessita de segmentação preliminar em unidades básicas.

HMM permite a integração de vários níveis de conhecimento em um *framework* unificado, ou seja, toda fonte de conhecimento (lingüístico, sintático, semântico ...) participa simultaneamente em cada decisão.

7. Conclusão

Neste artigo foi apresentado uma introdução ao HMM. Primeiramente foram citadas alguns linhas de pesquisa onde o HMM tem sido largamente utilizado. HMM foi primeiramente aplicado no reconhecimento de voz, porém atualmente tem sido bastante utilizado no reconhecimento de palavras manuscritas.

Todos os elementos que compõem um HMM para a observação de símbolos discretos foram descritos. Foram mostradas as principais arquiteturas de HMM, e que a estrutura do modelo e o número de estados escolhidos são fatores fundamentais para a determinação de um HMM ótimo.

Foram abordadas os três problemas básicos do HMM que devem ser resolvidos para que o modelo possa ser utilizado em aplicações do mundo real, bem como a solução para os mesmos. Também mostrou-se que existem duas soluções para o problema do reconhecimento, conhecidas como Modelo Discriminante e Modelo de Caminho Discriminante.

Por ultimo, foram colocadas algumas limitações, bem como algumas vantagens no uso do HMM.

8. Referências Bibliográficas

[1] Baker J. K., "*Stochastic Modeling as a Means of Automatic Speech Recognition*", PhD. Dissertation, Carriegie-Mellon University, 1975.

[2] Baum L. E, Petrie T., "*Statistical Inference for Probabilistic Functions of Finite State Markov Chains*", Ann Math. Stat. 37, pp. 1554-1563, 1966.

[3] Baum L. E, Eagon J. A., "*An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology*", Bull. Amer. Math. Soc. 73, pp. 360-363, 1967.

[4] Baum L. E, Sell G. R., "*Grown Functions for Transformations on Manifolds*", Pac. J. Math., Vol. 27, n. 2, pp. 211-227, 1968.

[5] Baum L. E, "*An Inequality and Associated Maximisation Technique in Statistical Estimation for Probabilistic Functions of a Markov Process"*, Inequalities III, pp.1-8, 1972.

[6] Casey R. G. , Lecolinet E., "*A Survey of Methods and Strategies in Character Segmentation*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, n. 7, Jul. 1996, pp. 690-706.

[7] Fielding K. H., Ruck D. W. "*Recognition of Moving Light Displays using Hidden Markov Models*", Pattern Recognition, Vol. 28, n. 9, pp. 1415-1421, 1995.

[8] Forney G.D. "*The Viterbi Algorithm*", Procs of the IEEE, Vol. 61, n. 3 pp. 268-278, 1973.

[9] Jelinek F., Mercer R. L., Roukos S. "*Principles of Lexical Language Modeling for Speech Recognition*", Advances in Speech Signal Processing, Edited by Sadaoki Furui and M. Mohan Sondhi., pp. 651-699. 1992

[10] Kundu A., He Y., Bahl P. "*Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach*", Pattern Recogniton, Vol 22, n. 3, pp. 283-297, 1989.

[11] Lee K.F, Hon H.W., Hwang M.Y., Huang X. "*Speech Recognition Using Hidden Markov Models: A CMU Perspective*", Speech Communication 9, Elsevier Science Publishers B. V., North Holland, pp. 497-508, 1990.

[12] Lou H.L. "*Implementing the Viterbi Algorithm*", IEEE Signal Processing Magazine, pp. 42-52, 1995.

[13] Rabiner L. R., "*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*", Procedings of the IEEE, Vol. 77, n. 2, Feb. 1989.

[14] Rabiner L. R. "*High Performace Connected Digit Recognition Using Markov Models*", IEEE Transactions on Acoustic, Speech and Signal Processing, Vol. 37, n. 8, pp. 1214-1224, 1989.

[15] Rabiner L., Juang B-H. "*Fundamentals of Speech Recognition*", Prentice Hall Signal Processing Series, 1993.

[16] Smyth P. "*Hidden Markov Models for Fault Detection in Dynamic Systems*", Pattern Recognition, Vol. 27, n. 1, pp. 149-167, 1994.

[17] Yacoubi A., "*Modélisation Markovienne de L'écriture Manuscrite Application à la Reconnaissance des Adresses Postales"*, Thèse de doctorat, Université de Rennes 1, Sep. 1996.

[18] Yacoubi A., Sabourin R., Gilloux M., Suen C. Y, "*Off-line Handwritten Word Recognition using Hidden Markov Modesl*", Knowledge Techniques in Character Recognition, CRC Press LLC, to appear by April 1.999.

[19] Yang L., Widjaja B. K., Prasad R. "*Application of Hidden Markov Models for Signature Verification*", Pattern Recognition, Vol. 28, n. 2, pp. 161-171, 1995.

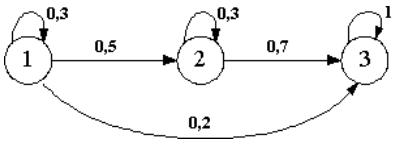
[20] Yang J., Xu Y., Chen S. "*Human Action Learnig via Hidden Markov Models*", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol. 27, n. 1, pp. 34-44, 1997.

1. Exemplo

Seja o modelo $\lambda = (A, B, \pi)$ compreendido por três estados 1, 2 e 3, e permitindo a observação de dois símbolos *a* e *b* e dado a seqüência "*aabb*". Determinar:

1. Cálculo da variável *forward* α.
2. Cálculo de $P(O \mid \lambda)$.
3. Cálculo da variável *backward* β.
4. Calcular a melhor seqüência de estados utilizando o algoritmo de Viterbi.

$$A = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0 & 0.3 & 0.7 \\ 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \text{ e } \pi = \begin{bmatrix} 0.6 \\ 0.4 \\ 0 \end{bmatrix}$$



1. Cálculo de α

Inicialização (Equação 6)

$\alpha_1(1) = 0.6 \times 1 = 0.6$, $\alpha_1(2) = 0.4 \times 0.5 = 0.2$, $\alpha_1(3) = 0 \times 0 = 0$

Indução (Equação 7)

$$\alpha_2(2) = \left[\sum_{i=1}^{3} \alpha_1(i) a_{i2}\right] b_2(O_2) = [\,0.6 \times 0.5 + 0.2 \times 0.3 + 0 \times 0\,] \times 0.5 = 0.18$$

$$\alpha_2(3) = \left[\sum_{i=1}^{3} \alpha_1(i) a_{i3}\right] b_3(O_2) = [\,0.6 \times 0.2 + 0.2 \times 0.7 + 0 \times 1\,] \times 0 = 0$$

$$\alpha_3(1) = \left[\sum_{i=1}^{3} \alpha_2(i) a_{i1}\right] b_1(O_3) = [\,0.18 \times 0.3 + 0.18 \times 0 + 0 \times 0\,] \times 0 = 0$$

$$\alpha_3(2) = \left[\sum_{i=1}^{3} \alpha_2(i) a_{i2}\right] b_2(O_3) = [\,0.18 \times 0.5 + 0.18 \times 0.3 + 0 \times 0\,] \times 0.5 = 0.072$$

$$\alpha_3(3) = \left[\sum_{i=1}^{3} \alpha_2(i) a_{i3}\right] b_3(O_3) = [\,0.18 \times 0.2 + 0.18 \times 0.7 + 0 \times 1\,] \times 1 = 0.162$$

$$\alpha_4(1) = \left[\sum_{i=1}^{3} \alpha_3(i) a_{i1}\right] b_1(O_4) = [\,0 \times 0.3 + 0.072 \times 0 + 0.162 \times 0\,] \times 0 = 0$$

$$\alpha_4(2) = \left[\sum_{i=1}^{3} \alpha_3(i) a_{i2} \right] b_2(O_4) = [\, 0 \times 0.5 + 0.072 \times 0.3 + 0.162 \times 0 \,] \times 0.5 = 0.0108$$

$$\alpha_4(3) = \left[\sum_{i=1}^{3} \alpha_3(i) a_{i3} \right] b_3(O_4) = [\, 0 \times 0.2 + 0.072 \times 0.7 + 0.162 \times 1 \,] \times 1 = 0.2124$$

2. Cálculo de $P(O \mid \lambda)$

Terminação (Equação 8)

$$P(O \mid \lambda) = \sum_{i=1}^{3} \alpha_4(i) = \alpha_4(1) + \alpha_4(2) + \alpha_4(3) = 0 + 0.0108 + 0.2124 = 0.2232$$

3. Cálculo de β

Inicialização (Equação 10)

$$\beta_4(1)=1,\ \beta_4(2)=1,\ \beta_4(3)=1$$

Indução (Equação 11)

$$\beta_3(1)=\sum_{j=1}^{3}a_{1j}b_j(O_4)\beta_4(j)=[0.3\times0\times1+0.5\times0.5\times1+0.2\times1\times1]=0.45$$

$$\beta_3(2)=\sum_{j=1}^{3}a_{2j}b_j(O_4)\beta_4(j)=[0\times0\times1+0.3\times0.5\times1+0.7\times1\times1]=0.85$$

$$\beta_3(3)=\sum_{j=1}^{3}a_{3j}b_j(O_4)\beta_4(j)=[0\times0\times1+0\times0.5\times1+1\times1\times1]=1$$

$$\beta_2(1)=\sum_{j=1}^{3}a_{1j}b_j(O_3)\beta_3(j)=[0.3\times0\times0.45+0.5\times0.5\times0.85+0.2\times1\times1]=0.4125$$

$$\beta_2(2)=\sum_{j=1}^{3}a_{2j}b_j(O_3)\beta_3(j)=[0\times0\times0.45+0.3\times0.5\times0.85+0.7\times1\times1]=0.8275$$

$$\beta_2(3)=\sum_{j=1}^{3}a_{3j}b_j(O_3)\beta_3(j)=[0\times0\times0.45+0\times0.5\times0.85+1\times1\times1]=1$$

$$\beta_1(1)=\sum_{j=1}^{3}a_{1j}b_j(O_2)\beta_2(j)=[0.3\times1\times0.4125+0.5\times0.5\times0.8275+0.2\times0\times1]=0.330625$$

$$\beta_1(2)=\sum_{j=1}^{3}a_{2j}b_j(O_2)\beta_2(j)=[0\times1\times0.4125+0.3\times0.5\times0.8275+0.7\times0\times1]=0.124125$$

$$\beta_1(3)=\sum_{j=1}^{3}a_{3j}b_j(O_2)\beta_2(j)=[0\times1\times0.4125+0\times0.5\times0.8275+1\times0\times1]=1$$

$$\beta_0(1)=\sum_{j=1}^{3}a_{1j}b_j(O_1)\beta_1(j)=[0.3\times1\times0.330625+0.5\times0.5\times0.124125+0.2\times0\times1]=0.13021875$$

$$\beta_0(2)=\sum_{j=1}^{3}a_{2j}b_j(O_1)\beta_1(j)=[0\times1\times0.330625+0.3\times0.5\times0.124125+0.7\times0\times1]=0.01861875$$

$$\beta_0(3)=\sum_{j=1}^{3}a_{3j}b_j(O_1)\beta_1(j)=[0\times1\times0.330625+0\times0.5\times0.124125+1\times0\times1]=0$$

Terminação (Equação 12)

$$P(O/\lambda)=\sum_{i=1}^{N}\pi_i b_i(o_1)\beta_1(i)=0.6\times1\times0.33025+0.4\times0.5\times0.124125+0\times0\times0=0.2232$$

Obs: Pode-se observar que o resultado da $P(O \mid \lambda)$ é o mesmo para as variáveis *forward-backward*, ou seja, para a resolução desse problema basta que somente uma variável seja calculada (*forward* ou *backward).*

4. Calcular a melhor seqüência de estados utilizando o algoritmo de Viterbi

Inicialização (Equações 15a e 15b)

$\delta_1(1) = 0.6 \times 1 = 0.6$, $\psi_1(1) = 0$

$\delta_1(2) = 0.4 \times 0.5 = 0.2$, $\psi_1(2) = 0$

$\delta_1(3) = 0 \times 0 = 0$, $\psi_1(3) = 0$

Recursão (Equações 16a e 16b)

$$\delta_2(1) = \max_{1 \le i \le 3}\left[\delta_1(i)a_{i1}\right] b_1(O_2) = \max\begin{bmatrix} \delta_1(1)a_{11} = 0.6 \times 0.3 \\ \delta_1(2)a_{21} = 0.2 \times 0 \\ \delta_1(3)a_{31} = 0 \times 0 \end{bmatrix} \times 1 = 0.18$$

$$\psi_2(1) = \arg\max_{1 \le i \le 3}\left[\delta_1(i)a_{i1}\right] = \arg\max\begin{bmatrix} \delta_1(1)a_{11} = 0.6 \times 0.3 \\ \delta_1(2)a_{21} = 0.2 \times 0 \\ \delta_1(3)a_{31} = 0 \times 0 \end{bmatrix} = 1$$

$$\delta_2(2) = \max_{1 \le i \le 3}\left[\delta_1(i)a_{i2}\right] b_2(O_2) = \max\begin{bmatrix} \delta_1(1)a_{12} = 0.6 \times 0.5 \\ \delta_1(2)a_{22} = 0.2 \times 0.3 \\ \delta_1(3)a_{32} = 0 \times 0 \end{bmatrix} \times 0.5 = 0.15$$

$$\psi_2(2) = \arg\max_{1 \le i \le 3}\left[\delta_1(i)a_{i2}\right] = \arg\max\begin{bmatrix} \delta_1(1)a_{12} = 0.6 \times 0.5 \\ \delta_1(2)a_{22} = 0.2 \times 0.3 \\ \delta_1(3)a_{32} = 0 \times 0 \end{bmatrix} = 1$$

$$\delta_2(3) = \max_{1 \le i \le 3}\left[\delta_1(i)a_{i3}\right] b_3(O_2) = \max\begin{bmatrix} \delta_1(1)a_{13} = 0.6 \times 0.2 \\ \delta_1(2)a_{23} = 0.2 \times 0.7 \\ \delta_1(3)a_{33} = 0 \times 1 \end{bmatrix} \times 0 = 0$$

$$\psi_2(3) = \arg\max_{1 \le i \le 3}\left[\delta_1(i)a_{i3}\right] = \arg\max\begin{bmatrix} \delta_1(1)a_{13} = 0.6 \times 0.2 \\ \delta_1(2)a_{23} = 0.2 \times 0.7 \\ \delta_1(3)a_{33} = 0 \times 1 \end{bmatrix} = 2$$

$$\delta_3(1) = \max_{1 \le i \le 3}\left[\delta_2(i)a_{i1}\right] b_1(O_3) = \max\begin{bmatrix} \delta_2(1)a_{11} = 0.18 \times 0.3 \\ \delta_2(2)a_{21} = 0.15 \times 0 \\ \delta_2(3)a_{31} = 0 \times 0 \end{bmatrix} \times 0 = 0$$

$$\psi_3(1) = \arg\max_{1 \le i \le 3}\left[\delta_2(i)a_{i1}\right] = \arg\max\begin{bmatrix} \delta_2(1)a_{11} = 0.18 \times 0.3 \\ \delta_2(2)a_{21} = 0.15 \times 0 \\ \delta_2(3)a_{31} = 0 \times 0 \end{bmatrix} = 1$$

$$\delta_3(2) = \max_{1 \le i \le 3}\left[\delta_2(i)a_{i2}\right] b_2(O_3) = \max\begin{bmatrix} \delta_2(1)a_{12} = 0.18 \times 0.5 \\ \delta_2(2)a_{22} = 0.15 \times 0.3 \\ \delta_2(3)a_{32} = 0 \times 0 \end{bmatrix} \times 0.5 = 0.045$$

$$\psi_3(2) = \arg\max_{1 \le i \le 3}\left[\delta_2(i)a_{i2}\right] = \arg\max\begin{bmatrix} \delta_2(1)a_{12} = 0.18 \times 0.5 \\ \delta_2(2)a_{22} = 0.15 \times 0.3 \\ \delta_2(3)a_{32} = 0 \times 0 \end{bmatrix} = 1$$

$$\delta_3(3) = \max_{1\le i\le 3}\left[\delta_2(i)a_{i3}\right]b_3(O_3) = \max\begin{bmatrix}\delta_2(1)a_{13} = 0.18\times 0.2\\ \delta_2(2)a_{23} = 0.15\times 0.7\\ \delta_2(3)a_{33} = 0\times 1\end{bmatrix}\times 1 = 0.105$$

$$\psi_3(2) = \arg\max_{1\le i\le 3}\left[\delta_2(i)a_{i3}\right] = \arg\max\begin{bmatrix}\delta_2(1)a_{13} = 0.18\times 0.2\\ \delta_2(2)a_{23} = 0.15\times 0.7\\ \delta_2(3)a_{33} = 0\times 1\end{bmatrix} = 1$$

$$\delta_4(1) = \max_{1\le i\le 3}\left[\delta_3(i)a_{i1}\right]b_1(O_4) = \max\begin{bmatrix}\delta_3(1)a_{11} = 0\times 0.3\\ \delta_3(2)a_{21} = 0.045\times 0\\ \delta_3(3)a_{31} = 0.105\times 0\end{bmatrix}\times 0 = 0$$

$$\psi_4(1) = \arg\max_{1\le i\le 3}\left[\delta_3(i)a_{i1}\right] = \arg\max\begin{bmatrix}\delta_3(1)a_{11} = 0\times 0.3\\ \delta_3(2)a_{21} = 0.045\times 0\\ \delta_3(3)a_{31} = 0.105\times 0\end{bmatrix} = 1$$

$$\delta_4(2) = \max_{1\le i\le 3}\left[\delta_3(i)a_{i2}\right]b_2(O_4) = \max\begin{bmatrix}\delta_3(1)a_{12} = 0\times 0.5\\ \delta_3(2)a_{22} = 0.045\times 0.3\\ \delta_3(3)a_{32} = 0.105\times 0\end{bmatrix}\times 0.5 = 0.0135$$

$$\psi_4(2) = \arg\max_{1\le i\le 3}\left[\delta_3(i)a_{i2}\right] = \arg\max\begin{bmatrix}\delta_3(1)a_{12} = 0\times 0.5\\ \delta_3(2)a_{22} = 0.045\times 0.3\\ \delta_3(3)a_{32} = 0.105\times 0\end{bmatrix} = 1$$

$$\delta_4(3) = \max_{1\le i\le 3}\left[\delta_3(i)a_{i3}\right]b_3(O_4) = \max\begin{bmatrix}\delta_3(1)a_{13} = 0\times 0.2\\ \delta_3(2)a_{23} = 0.045\times 0.7\\ \delta_3(3)a_{33} = 0.105\times 1\end{bmatrix}\times 1 = 0.105$$

$$\psi_4(3) = \arg\max_{1\le i\le 3}\left[\delta_3(i)a_{i3}\right] = \arg\max\begin{bmatrix}\delta_3(1)a_{13} = 0\times 0.2\\ \delta_3(2)a_{23} = 0.045\times 0.7\\ \delta_3(3)a_{33} = 0.105\times 1\end{bmatrix} = 3$$

Terminação (Equações 17a e 17b)

$$P^* = \max_{1\le i\le 3}\left[\delta_4(i)\right] = \begin{bmatrix}\delta_4(1) = 0\\ \delta_4(2) = 0.0135\\ \delta_4(3) = 0.105\end{bmatrix} = 0.105\,,\quad q_T^* = \arg\max\begin{bmatrix}\delta_4(1) = 0\\ \delta_4(2) = 0.0135\\ \delta_4(3) = 0.105\end{bmatrix} = 3$$

Backtracking (Equação 18)

$q_3^* = \psi_4(q_4^*) = 3$, $q_2^* = \psi_3(q_3^*) = 2$, $q_1^* = \psi_2(q_2^*) = 1$

Assim, a melhor seqüência de estados obtida é 1, 2, 3 e 3.