Laboratório VISGRAF
Instituto de Matemática Pura e Aplicada

Hidden Markov Models

Anderson Mayrink da Cunha
Luiz Velho (orientador)

Technical Report TR-02-02 Relatório Técnico

January - 2002 - Janeiro

Hidden Markov Models

Autor: Anderson Mayrink da Cunha
Orientador: Luiz Velho

Sumário

Capítulo 1

Probabilidade e Cadeias de Markov

1.1 Probabilidade

- Um espaço de probabilidade é um trio $(\Omega, \mathcal{A}, P)$ onde:

1. Ω é o espaço amostral, que é um conjunto não vazio de todos os resultados possíveis de um experimento. A todo resultado possível corresponde a um e somente um ponto $\omega \in \Omega$.
2. $\mathcal{A}$ é uma σ-álgebra de subconjuntos de Ω, isto é, $\mathcal{A}$ é uma família de subconjuntos de Ω a cujos elementos (eventos) é possível associar um valor real de medida de probabilidade.
3. A medida de probabilidade P é uma função que associa elementos de $\mathcal{A}$ (subconjuntos de Ω) a um número real. A probabilidade de um evento A, que denotamos por $P[A]$, corresponde à frequência relativa em uma situação experimental e satisfaz aos seguintes axiomas:

   a. $\forall A \subset \Omega, 0 \leq P[A] \leq 1$
   b. $P[\Omega] = 1$
   c. Se dois eventos A e B são mutuamente exclusivos (ie. $A \cap B = AB = \emptyset$), temos que $P[A \cup B] = P[A] + P[B]$

Probabilidade Condicional

A probabilidade condicional de um evento A dado que o evento B ocorreu é definida por:

$$P[A \mid B] = \frac{P[AB]}{P[B]}, \text{ se } P[B] \neq 0$$

A probabilidade condicional possui uma interpretação intuitiva em termos de frequências relativas:

$$P[A \mid B] = \frac{P[AB]}{P[B]} = \frac{\lim \frac{1}{n}(\text{n}^0 \text{ de ocorrências de } A \cap B)}{\lim \frac{1}{n}(\text{n}^0 \text{ de ocorrências de } B)} = \lim \frac{\text{n}^0 \text{ de ocorrências de } A \cap B}{\text{n}^0 \text{ de ocorrências de } B}$$

Teorema da Probabilidade Total

Seja A_i uma partição do espaço amostral Ω (ie: $\cup A_i = \Omega$ e $A_i A_j = \emptyset$). O teorema da probabilidade total diz que

$$P[B] = \sum_i P[A_i B] = \sum_i P[B \mid A_i] P[A_i]$$

Teorema de Bayes

Como $P[A_iB] = P[A_i \mid B]P[B] = P[B \mid A_i]P[A_i]$ temos o teorema de Bayes:

$$P[A_i \mid B] = \frac{P[B \mid A_i]P[A_i]}{P[B]}$$

Se os A_i são mutuamente exclusivos temos pelo teorema da probabilidade total que:

$$P[B] = \sum_j P[B \mid A_j]P[A_j]$$

Essa fórmula é útil quando conhecemos $P[A_i]$ e $P[B \mid A_i]$ mas não conhecemos diretamente $P[B]$.

1.2 Processos Estocásticos

Variável Aleatória

Uma variável aleatória (ou randômica) pode ser vista como o nome de um experimento aleatório. Seu valor é o resultado desse experimento. Seja um ponto amostral $\omega \in \Omega$, denotamos a variável aleatória por $X(\omega)$.

Esperança Matemática

Seja X uma variável aleatória discreta com função de probabilidade $p(x_i)$. A esperança de X é dada por:

$$E[X] = \sum_x xp(x) = \sum_i x_i P(X = x_i)$$

A esperança matemática de X também é chamada de média de X ou valor esperado de X. $E[X]$ é uma média ponderada, onde os pesos são as probabilidades $p(x)$.

Definição de Processos Estocásticos

Seja $(\Omega, \mathcal{A}, P)$ um espaço de probabilidade e $\omega \in \Omega$ um ponto amostral. Um processo estocástico (ou randômico) é uma família de variáveis aleatórias $X(t, \omega)$, onde as variáveis aleatórias são indexadas por um parâmetro de tempo t. Um processo estocástico é uma função $X(t, \omega)$ (denotamos $X(t)$ por simplicidade) cujos valores são variáveis aleatórias.

1.2.1 Classificação de Processos Estocásticos

A classificação de processos estocásticos depende de três quantidades:

- O espaço de estado, que é o conjunto de valores (ou estados) possíveis de $X(t)$. Se o espaço de estado é finito ou contável, temos um processo de estado discreto ou cadeia. Caso contrário temos processo de estado contínuo.
- O parâmetro de tempo. Se os tempos permitidos de troca de estado são finitos ou enumeráveis, temos um processo de parâmetro (de tempo) discreto. Caso contrário temos um processo de parâmetro contínuo.
- A dependência estatística entre as variáveis aleatórias $X(t)$ para diferentes valores do parâmetro t. Nos interessa um tipo de dependência descrita por Markov que será descrita a seguir.

1.3 Classificação de Padrões

O objetivo básico deste trabalho é o reconhecimento de escrita (inicialmente dígitos de 0 a 9). Dados um modelo para cada dígito (esse modelo é um HMM, como veremos no próximo capítulo) e uma observação x (conjunto de pontos conectados no plano). A observação x é melhor representada por qual dígito? Qual dígito minimiza a chance de erro dessa escolha? Há dois métodos básicos de classificação, como vemos a seguir.

1.3.1 Maximum likelihood

Classificamos a observação x como:

$$\omega^* = \arg\max_i \quad p(\mathsf{x}|\omega_i), \quad \text{onde } \omega_i \text{ é o modelo para o dígito } i.$$

Este é o classificador de máxima probabilidade para ω^*. Se conhecemos a priori as probabilidades $p(\omega_i)$ devemos alterar o método de classificação.

1.3.2 Bayes

Se conhecemos $p(\mathsf{x}|\omega_i)$ e $p(\omega_i)$ podemos classificar a observação x como:

$$\omega^* = \arg\max_i \quad p(\omega_i|\mathsf{x})$$

Pelo teorema de Bayes: $p(\omega_i|\mathsf{x}) = \frac{p(\mathsf{x}|\omega_i).p(\omega_i)}{p(\mathsf{x})}$

Como $p(\mathsf{x})$ é um fator comum de escala, temos que:

$$\omega^* = \arg\max_i \quad p(\mathsf{x}|\omega_i).p(\omega_i)$$

A equação acima é conhecida como o classificador ótimo de Bayes. Note que se os $p(\omega_i)$ são iguais, o classificador de Bayes se reduz ao classificador de máxima probabilidade.

1.4 Processos de Markov

Em 1907, Markov definiu e investigou propriedades que são hoje conhecidas como processos de Markov. A principal característica dos processos de Markov é que o modo que toda a história passada afeta o futuro está completamente resumida no valor atual do processo. Nos interessa principalmente cadeias de Markov em tempo discreto, que definimos na próxima seção.

1.4.1 Cadeias de Markov em Tempo Discreto

Considere um sistema cuja evolução seja descrita por um processo estocástico $\{X(n) = X_n, n = 1, 2, \dots\}$, consistindo de uma família de variáveis aleatórias. O valor s_n assumido pela variável aleatória X_n é chamado de estado do sistema no tempo discreto n. O conjunto de todos os valores que as variáveis aleatórias podem assumir é chamado de espaço de estados do sistema. Se a estrutura do processo estocástico é tal que a distribuição de probabilidade condicional de X_n depende somente do valor de X_{n-1} e é independente de todos os valores anteriores, dizemos que o processo é uma cadeia de Markov. Mais precisamente:

Definição 1 Uma sequência de variáveis aleatórias $X_1, X_2, \dots$ é uma cadeia de Markov em tempo discreto se para todo tempo n ($n = 1, 2, \dots$) e para todo o espaço de estados do sistema temos que:

$$P[X_n = s_n \mid X_1 = s_1, X_2 = s_2, \dots, X_{n-1} = s_{n-1}] = P[X_n = s_n \mid X_{n-1} = s_{n-1}]$$

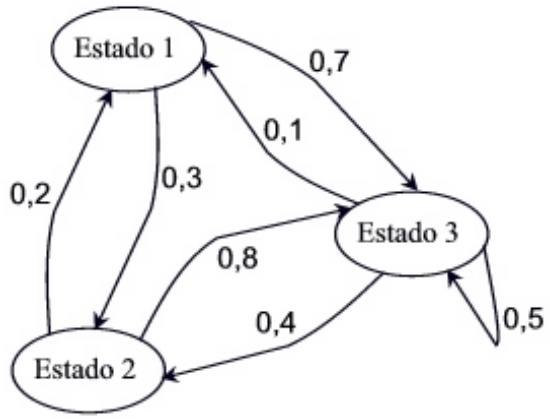

Figura 1.1: Cadeia de Markov Finita

Podemos pensar na cadeia de Markov como um modelo gerador consistindo de estados ligados entre si por transições possíveis. Em cada unidade de tempo n um estado particular é visitado e o modelo coloca na saída o símbolo associado àquele estado.

Se a probabilidade condicional $P[X_n = s_j \mid X_{n-1} = s_i]$ for independente do tempo n, chamamos a cadeia de Markov de homogênea. Todas as cadeias de Markov a partir daqui são homogêneas. Nesse caso temos que $P_{ij} = P[X_n = s_j \mid X_{n-1} = s_i]$ é independente de n e então podemos organizar os P_{ij} numa matriz de transição de probabilidades.

Uma cadeia de Markov fica totalmente determinada pela matriz $A = \{a_{ij} = P[X_n = s_j \mid X_{n-1} = s_i]\}$ e pelo vetor π de probabilidades iniciais.

Exemplo 1

Seja uma partícula que a cada unidade de tempo está em um entre três locais distintos (estados 1, 2 e 3). O novo estado da partícula depende somente do estado atual de acordo com a matriz de probabilidades

$$A = \begin{bmatrix} 0 & 0,3 & 0,7 \\ 0,2 & 0 & 0,8 \\ 0,1 & 0,4 & 0,5 \end{bmatrix}$$

Todos os elementos dessa matriz estão entre 0 e 1 e a soma dos elementos de cada linha é 1.

Esta é uma cadeia de Markov homogênea. A matriz de transição de probabilidades também pode ser expressa no grafo da figura 1.1.

Estimativa dos parâmetros de uma cadeia de Markov

Dado uma sequência de observação $X = \{X_1, X_2, \ldots, X_T\}$

A estimativa de maximum likelihood para os parâmetros da cadeia de Markov $\lambda = \{A, \pi\}$ é dada por:

$$a_{ij} = P[X_n = s_j \mid X_{n-1} = s_i] = \frac{\text{n}^0 \text{ de transições de } s_i \text{ para } s_j}{\text{n}^0 \text{ de vezes no estado } s_i}$$
$$\pi_i = 1, \text{ se } X_1 = s_i \text{ e } \pi_i = 0, \text{ caso contrário.}$$

1.4.2 Cadeias de Markov em Tempo Contínuo

Apesar de não serem o foco nesse trabalho, cadeias de Markov em tempo contínuo são muito importantes em teoria de filas, particularmente em redes de computadores e por isso vamos brevemente comentá-las.

Definição 2 Um processo aleatório X_t é uma cadeia de Markov em tempo contínuo se para qualquer sequência $t_1, t_2, \ldots, t_n, t_{n+1}$ com $t_1 < t_2 < \ldots < t_n < t_{n+1}$ e para todo o espaço de estados do sistema temos que:

$$P[X_{t_{n+1}} = s_{n+1} \mid X_{t_1} = s_1, X_{t_2} = s_2, \ldots, X_{t_n} = s_n] = P[X_{t_{n+1}} = s_{n+1} \mid X_{t_n} = s_n]$$

A interpretação de uma cadeia de Markov em tempo contínuo é próxima à de tempo discreto, com a diferença que a transição de estados pode ocorrer a qualquer tempo, e não somente em tempos inteiros.

Exemplo 2: Processos de nascimento e morte

Um processo nascimento e morte é um caso particular de processos de Markov em que somente transições de s_n para os estados vizinhos s_{n-1}, s_n, s_{n+1} são permitidas. A transição de s_n para s_{n+1} é chamada de nascimento e a transição de s_n para s_{n-1} é chamada de morte. Seja λ a taxa de nascimento e μ a taxa de morte. Assumimos que λ e μ são independentes do tempo e do estado. O gerador infinitesimal desse processo é da forma:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \ldots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \ldots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

O conjunto de equações diferenciais que nos fornecem a dinâmica do nosso sistema é:

$$\begin{cases} \frac{dP_k(t)}{dt} = -(\lambda+\mu)P_k(t) + \lambda P_{k-1}(t) + \mu P_{k+1}(t) & k \geq 1 \\ \frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t) & k = 0 \end{cases}$$

Este processo também é conhecido como sistema de filas M/M/1. O número médio de clientes desse sistema em equilíbrio é: $\overline{N} = \frac{\lambda/\mu}{1-\lambda/\mu}$.

Se $\mu = 0$ temos um processo de nascimento puro e $P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$.

Capítulo 2

Hidden Markov Models

Um modelo oculto de Markov (Hidden Markov Model - HMM) é um modelo de Markov (cadeia de Markov) onde os estados do modelo não são conhecidos, mas apenas o sinal emitido em cada unidade de tempo t. O sinal O emitido em dada estado é gerado pela função densidade de distribuição $b_i(O)$. Consideramos aqui somente HMMs com número de estados e de unidades de tempo (tamanho da observação) e tamanho do alfabeto (sinais distintos possíveis emitidos em cada estado) finitos.

Um exemplo de um HMM é uma partícula que para cada $t = 1, 2, \ldots, T$ muda para um local (ou estado) entre os N possíveis. Em cada um dos estados, a partícula emite um sinal (de um alfabeto de tamanho M). Temos que o estado futuro da partícula depende somente do estado atual, e não dos estados anteriores ou do tempo t. Não conhecemos os estados da partícula, mas somente os sinais emitidos por ela.

Para modelar as características naturais de um sinal de voz ou de escrita à mão, a abordagem tradicional na área de reconhecimento de voz (ou escrita) tem sido utilizar HMMs.

Abaixo temos algumas notações:

I N é o número de estados do modelo, ou locais onde a partícula pode ir. Para simplificar a notação denominamos o conjunto de estados $\mathcal{S} = \{1, 2, \ldots, N\}$.

I M é o número total de símbolos distintos, o tamanho do alfabeto de sinais que a partícula emite. $V = \{v_1, v_2, \ldots, v_M\}$ é o alfabeto.

I T é a quantidade de unidades de tempo (tamanho da observação).

I $Q = \{q_1, q_2, \ldots, q_T\}$ onde q_t é o estado do modelo no tempo t.

I $O = \{O_1, O_2, \ldots, O_T\}$ onde O_t é símbolo observado no tempo t.

I $\pi = \{\pi_i\}, i = 1, \ldots, N$. onde $\pi_i = P[q_1 = i]$ é a probabilidade de i ser o estado inicial do experimento.

I $A = \{a_{ij}\}$ é uma matriz $N \times N$, onde $a_{ij} = P[q_{t+1} = j \mid q_t = i]$ é a probabilidade da partícula ir do estado i para o estado j. Os a_{ij}'s são independentes do tempo t.

I $B = \{b_i(k)\}$ é uma matriz $N \times M$,onde $b_i(k)$ é a probabilidade do símbolo v_k ser observado no estado i.

- $\lambda = (A, B, \pi)$ é a notação compacta para um HMM.

2.1 Problemas de HMM

Os três principais problemas de HMM na maioria das aplicações serão discutidos nas subseções seguintes.

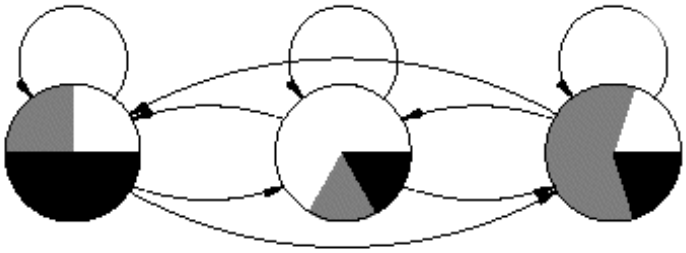

Figura 2.1: Exemplo de um HMM com 3 estados e 3 símbolos.

2.1.1 Problema 1:

Dado um modelo $\lambda = (A, B, \pi)$ como calcular $P[O \mid \lambda]$, a probabilidade da ocorrência da observação $O_1, O_2, \ldots, O_T$?, isto é, $P[O \mid \lambda] = ?$

O modo mais imediato de se calcular $P[O \mid \lambda]$ é achar $P[O \mid Q, \lambda]$ para um estado fixado $Q = q_1, q_2, \ldots, q_T$ e somar as probabilidades sobre todos os estados possíveis, isto é:

$$P[O \mid \lambda] = \sum_Q P[O, Q \mid \lambda] = \sum_Q P[O \mid Q, \lambda] P[Q, \lambda]$$

onde $P[O \mid Q, \lambda] = b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T)$ e $P[Q, \lambda] = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$

Para cada estado temos $2T - 1$ multiplicações para o cálculo de $P[O \mid Q, \lambda] P[Q, \lambda]$ e temos N^T estados. Daí temos na ordem de $2TN^T$ multiplicações para o cálculo de $P[O \mid \lambda]$. Isso é inviável computacionalmente pois, por exemplo, um modelo com 10 estados e 100 instantes de tempo (observações), isto é, $N = 10, T = 100$, temos ordem de 10^{102} multiplicações.

Para executar esse cálculo mais rapidamente usamos o procedimento abaixo.

Algoritmo Forward

A variável forward $\alpha_t(i)$ é a probabilidade da observação da sequência parcial $O_1, O_2, \ldots, O_t$ e que no tempo t tenhamos o estado i.

$$\alpha_t(i) = P[O_1, O_2, \ldots, O_t, q_t = i \mid \lambda]$$

$\alpha_t(i)$ pode ser calculada recusirvamente da seguinte forma:

$$\begin{cases} \alpha_1(i) = \pi_i b_i(O_1) & i = 1, \ldots, N \\ \alpha_{t+1}(j) = b_j(O_{t+1}) \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) & t = 1, 2, \ldots, T-1 \quad , \quad j = 1, \ldots, N \end{cases}$$

Temos que

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

Esse método de cálculo de $P[O \mid \lambda]$ envolve ordem de $N^2 T$ multiplicações.

A dedução da fórmula $\alpha_{t+1}(j) = b_j(O_{t+1})\left(\sum_{i=1}^{N}\alpha_t(i)\,a_{ij}\right)$ deve ser feita com o devido cuidado pois $P[AB] = P[A]P[B]$ só se A e B são independentes.

$\alpha_{t+1}(j) = P[O_1,\ldots,O_t,O_{t+1},q_{t+1}=j \mid \lambda] =$

$\sum_{i=1}^{N} P[O_1,\ldots,O_t,O_{t+1},q_{t+1}=j \mid q_t=i,\lambda]P[q_t=i \mid \lambda] =$

$\sum_{i=1}^{N} P[O_1,O_2,\ldots,O_t \mid q_t=i,\lambda]P[q_t=i \mid \lambda]P[O_{t+1},q_{t+1}=j \mid q_t=i,\lambda] =$

$\sum_{i=1}^{N} P[O_1,\ldots,O_t,q_t=i \mid \lambda]P[O_{t+1} \mid q_{t+1}=j,q_t=i,\lambda]P[q_{t+1}=j \mid q_t=i,\lambda] = b_j(O_{t+1})\left(\sum_{i=1}^{N}\alpha_t(i)\,a_{ij}\right)$

O cálculo da variável forward com o algoritmo descrito acima envolve problema de precisão numérica (underflow) pois, por exemplo, se temos um alfabeto de tamanho 100 e 100 observações $(T=100)$, α_{t+1} é da ordem $T=100$ vezes menor que α_t. Daí temos que $P[O \mid \lambda]$ é da ordem de 10^{-200}.

Para resolver este problema temos que escalar esse algoritmo. Esse escalamento consiste basicamente em cada passo do algoritmo criar uma variável auxiliar que indica o valor de $\sum_i \alpha_t(i)$.

Algoritmo Forward Escalado

1. Inicialização:

$\widetilde{\alpha}_1(i) = \pi_i b_i(O_1), i = 1\cdots N$

$c_1 = 1/\left(\sum_{i=1}^{N}\widetilde{\alpha}_1(i)\right)$

$\widehat{\alpha}_1(i) = c_1\widetilde{\alpha}_1(i), i = 1\cdots N$ (escalado)

2. Indução

$\widetilde{\alpha}_{t+1}(j) = b_j(O_{t+1})\left(\sum_{i=1}^{N}\widehat{\alpha}_t(i)\,a_{ij}\right), t = 1,\ldots,T-1 \quad , \quad j = 1,\ldots,N$

$c_{t+1} = 1/\left(\sum_{i=1}^{N}\widetilde{\alpha}_{t+1}(i)\right), t = 1,\ldots,T-1$

$\widehat{\alpha}_{t+1}(j) = c_{t+1}\widetilde{\alpha}_{t+1}(i), t = 1,\ldots,T-1 \quad , \quad j = 1,\ldots,N$ (escalado)

3. Finalização

$P[O \mid \lambda] = 1/\left(\prod_{t=1}^{T} c_t\right)$

isso é válido pois $\widehat{\alpha}_t(i) = \left(\prod_{\tau=1}^{t} c_\tau\right)\alpha_t(i)$ e $\sum_{i=1}^{N}\widehat{\alpha}_t(i) = 1$ logo $P[O \mid \lambda] = \sum_{i=1}^{N}\alpha_T(i) = \sum_{i=1}^{N}\widehat{\alpha}_T(i)\,/\left(\prod_{t=1}^{T} c_t\right) = \frac{1}{\prod_{t=1}^{T} c_t}\sum_{i=1}^{N}\widehat{\alpha}_T(i) = 1/\left(\prod_{t=1}^{T} c_t\right)$

Como $P[O \mid \lambda]$ pode ser muito pequeno, calculamos o seu logaritmo:

$$\log P[O \mid \lambda] = -\sum_{t=1}^{T}\log c_t$$

Algoritmo Backward

Podemos resolver esse problema de maneira quase análoga com a variável backward $\beta_t(i)$ definida como:

$$\beta_t(i) = P[O_{t+1}, O_{t+2}, \ldots, O_T \mid q_t = i, \lambda]$$

O cálculo de $\beta_t(i)$ também pode ser feito recursivamente do seguinte modo:

$$\begin{cases} \beta_T(i) = 1 & i = 1, \ldots, N \\ \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad t = T-1, T-2, \ldots, 1 \quad , & i = 1, \ldots, N \end{cases}$$

Temos que $P[O \mid \lambda] = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i)$

Esse cálculo também envolve ordem de N^2T multiplicações.

Vamos apresentar agora a versão escalada do algoritmo backward. As variáveis c_t são as calculadas no algoritmo forward escalado.

Algoritmo Backward Escalado

1. Inicialização:

$\widetilde{\beta}_T(i) = 1, i = 1 \cdots N$
$\widehat{\beta}_T(i) = c_T \widetilde{\beta}_T(i), i = 1 \cdots N$ (escalado)

2. Indução

$$\widetilde{\beta}_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \widehat{\beta}_{t+1}(i), \quad t = T-1, \ldots, 1 \quad , \quad i = 1, \ldots, N$$
$$\widehat{\beta}_t(i) = c_t \widetilde{\beta}_t(i), \quad t = T-1, \ldots, 1 \quad , \quad i = 1, \ldots, N \text{ (escalado)}$$

Temos que $\widehat{\beta}_t(i) = \left(\prod_{\tau=t}^{T} c_\tau \right) \beta_t(i)$

2.1.2 Problema 2:

Dado um modelo $\lambda = (A, B, \pi)$ e a sequência de observação $O_1, O_2, \ldots, O_T$. Qual a sequência de estados $Q = (q_1, q_2, \ldots, q_T)$ para que $P[Q \mid O, \lambda]$ seja maximizada?, isto é, $\arg\max_Q P[Q \mid O, \lambda] = ?$

Temos que $P[Q \mid O, \lambda] = \frac{P[Q, O|\lambda]}{P[O|\lambda]}$

Como $P[O|\lambda]$ é constante em relação à seqüência de estados Q, basta calcular:

$$Q^* = \arg\max_Q P[Q, O \mid \lambda]$$

Para executar o cálculo acima de modo eficiente usamos o algoritmo de Viterbi (essencialmente programação dinâmica), que descreveremos brevemente.

Seja a variável $\delta_t(i)$ definida por:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P[q_1, q_2, \ldots, q_t = i, O_1, \ldots, O_t \mid \lambda]$$

Podemos calcular $\delta_{t+1}(i)$ a partir da seguinte relação indutiva:

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

com $\delta_1(i) = P[q_1 = i, O_1|\lambda] = \pi_i b_i(O_1)$. Note que $P^* = \max_Q P[Q, O \mid \lambda] = \max_i \delta_T(i)$

Temos ainda que traçar o caminho de volta para achar a seqüência de estados

$q_T^* = \arg\max_i \delta_T(i)$ e $q_t^* = \arg\max_i [\delta_t(i) a_{iq_{t+1}^*}]$

Esse algoritmo é particularmente útil em reconhecimento de voz pois os estados ocultos podem ser interpretados como fonemas. Como no caso do algoritmo forward, é necessário algumas modificações para o cálculo escalado, o que evita underflow em computadores de precisão finita.

2.1.3 Problema 3:

Quais os parâmetros do modelo $\lambda = (A, B, \pi)$ para que $P[O \mid \lambda]$ seja maximizado?, isto é, $\arg\max_\lambda P[O \mid \lambda] = ?$

Antes do caso específico de HMMs (com o algoritmo de Baum-Welch), vamos analisar o problema de treinamento de um modo mais geral com o algoritmo EM.

2.2 Treinamento

2.2.1 Algoritmo EM

Problema: Seja X um dado qualquer e uma família de modelos parametrizada por λ. Qual é λ para que $P(X|\lambda)$ seja maximizada

Solução: Algoritmo EM (Dempster et al., 1977)

Passo E (Expectation): Calcular a esperança matemática das variáveis ocultas, dado λ.

Passo M (Maximization): Calcular λ^* para que $P(X|\lambda^*)$ seja maximizado, assumindo os valores das variáveis ocultas no passo E. (e iterar com o novo valor de λ^*)

É provado que $P(X|\lambda^*) \geq P(X|\lambda)$ e que o algoritmo EM sempre converge para um máximo local.

Daremos dois exemplos de aplicação do EM: quantização vetorial e o algoritmo de Baum-Welch para HMMs.

2.2.2 Quantização Vetorial

Problema: Seja X um conjunto de N pontos de R^d. Achar L classes tal que a distorção (soma das distâncias de cada ponto x_j ao centróide de sua classe) seja minimizada

Para aplicar o algoritmo EM devemos procurar dados incompletos do problema.

Podemos ver o dado $X = \{x_j\}$ como incompleto. O dado completo é:

$$z_j = \{x_j, y_{1j}, y_{2j}, \ldots, y_{Lj}\}, j = 1 \cdots N$$

onde y_{ij} indica se x_j pertence à classe ω_i. Isto é, $y_{ij} = 1$ se x_j pertence à classe ω_i e $y_{ij} = 0$ caso contrário.

Nota-se que a distorção é inversamente proporcional a $P(X|\lambda)$. Se maximizamos $P(X|\lambda)$, a distorção é minimizada.

Solução: Algoritmo k-means.

Passo E: Dado as classes ω_i (seus centróides z_i e a função distância), calcular as (esperanças das) variáveis y_{ij}. Basta classificar os pontos x_j:

$$y_{ij} = 1 \Leftrightarrow x_j \in \omega_i \Leftrightarrow dist(x_j, z_i) \leq dist(x_j, z_k), \forall k.$$

Passo M: Calcular as novas classes ω_i (seus centróides z_i) que minimizam a distorção. Basta calcular o novo codebook $\{z_i\}$.

$$z_i = \frac{1}{n_i} \left(\sum_{x_j \in \omega_i} x_j \right)$$

onde n_i é o número de vetores x_j na classe ω_i.

Esse algoritmo (e suas variantes, por exemplo, algoritmo LBG) é o mais usado (para propósitos gerais) para quantização vetorial.

2.2.3 Algoritmo de Baum-Welch

Vamos agora resolver o problema 3 de HMM:

Quais os parâmetros de um HMM $\lambda = (A, B, \pi)$ para que $P[O \mid \lambda]$ seja maximizado?, isto é, $\arg\max_{\lambda} P[O \mid \lambda] = ?$

Esse é um problema de otimização contínua de $N^2 + NM + M$ variáveis. Uma solução para esse problema é o algoritmo de Baum-Welch. Vamos apresentá-lo no contexto do algoritmo EM.

Se já conhecemos os valores dos estados ocultos, o cálculo dos parâmetros do modelo para que $P[O|\lambda]$ seja máximo é simples (análogo para cadeias de Markov):

$$a_{ij} = \frac{\text{número de transições do estado } i \text{ para o estado } j}{\text{número de vezes no estado } i}$$

$$b_j(k) = \frac{\text{número de vezes no estado } j \text{ com o símbolo } v_k}{\text{número de vezes no estado } j}$$

$$\pi_i = \text{número de vezes no estado } i \text{ no tempo } t = 1$$

Como não conhecemos os estados e os valores de transições de estados indicados acima, usamos o algoritmo EM.

Passo E: Calcular as esperanças das variáveis ocultas, nesse caso, número de vezes nos estados, transição dos estados e outras variáveis indicadas acima.

Passo M: Das variáveis ocultas do passo E, calcular novos parâmetros do modelo para que $P[O|\lambda]$ seja máximo (basta calcular a_{ij}, $b_j(k)$ e π_i nas fórmulas acima). Iterar com esses novos parâmetros.

No passo E temos que calcular as esperanças das seguintes variáveis ocultas: número de transições do estado i para o estado j, número de vezes no estado i, número de vezes no estado i com o símbolo v_k e número de vezes no estado i no tempo $t = 1$. Para esse cálculo vamos definir duas variáveis auxiliares:

$\gamma_t(i) =$ probabilidade de termos o estado i no tempo t.

$\zeta_t(i, j) =$ probabilidade de termos o estado i no tempo t e o estado j no tempo $t + 1$.

$$\gamma_t(i) = P[q_t = i|O, \lambda]$$
$$\zeta_t(i,j) = P[q_t = i, q_{t+1} = j \mid O, \lambda]$$

Note que:

$$\sum_{t=1}^{T-1}\gamma_t(i) = \text{número esperado de vezes no estado } i$$
$$\sum_{t=1}^{T-1}\zeta_t(i,j) = \text{número esperado de transições do estado } i \text{ para o estado } j$$

A partir dessas variáveis podemos apresentar as fórmulas de Baum-Welch:

$$\begin{cases} \widehat{\pi}_i = \gamma_1(i) \\ \widehat{a}_{ij} = \left(\sum_{t=1}^{T-1}\zeta_t(i,j)\right) / \left(\sum_{t=1}^{T-1}\gamma_t(i)\right) \\ \widehat{b}_j(k) = \left(\sum_{\substack{t=1 \\ O_t=k}}^{T}\gamma_t(j)\right) / \left(\sum_{t=1}^{T}\gamma_t(i)\right) \end{cases}$$

Vamos agora calcular $\gamma_t(i)$ e $\zeta_t(i,j)$ a partir dos parâmetros do HMM e das variáveis α e β:

$$\gamma_t(i) = P[q_t = i \mid O, \lambda] = \frac{P[q_t = i, O \mid \lambda]}{P[O \mid \lambda]} = \frac{\alpha_t(i)\beta_t(i)}{P[O \mid \lambda]}$$

pois $P[q_t = i, O \mid \lambda] = P[O \mid q_t = i, \lambda]P[q_t = i \mid \lambda] =$
$P[O_1, O_2, \cdots, O_t, O_{t+1}, \cdots, O_T \mid q_t = i, \lambda]P[q_t = i \mid \lambda] =$
$P[O_1, O_2, \cdots, O_t \mid q_t = i, \lambda]P[q_t = i \mid \lambda]P[O_{t+1}, \cdots, O_T \mid q_t = i, \lambda] =$
$P[O_1, O_2, \cdots, O_t, q_t = i \mid \lambda]P[O_{t+1}, \cdots, O_T \mid q_t = i, \lambda] = \alpha_t(i)\beta_t(i)$
pois $O_1, \cdots, O_t$ é independente de $O_{t+1}, \cdots, O_T$.

$$\zeta_t(i,j) = P[q_t = i, q_{t+1} = j|O, \lambda] = \frac{P[q_t = i, q_{t+1} = j, O|\lambda]}{P[O \mid \lambda]} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P[O \mid \lambda]}$$

pois $P[q_t = i, q_{t+1} = j, O|\lambda] = P[q_{t+1} = j, O|q_t = i, \lambda]P[q_t = i \mid \lambda] =$
$P[O_1, O_2, \cdots, O_t, O_{t+1}, \cdots, O_T, q_{t+1} = j|q_t = i, \lambda]P[q_t = i \mid \lambda] =$
$P[O_1, O_2, \cdots, O_t \mid q_t = i, \lambda]P[q_t = i \mid \lambda]P[O_{t+1}, \cdots, O_T, q_{t+1} = j \mid q_t = i, \lambda] =$
$\alpha_t(i)P[O_{t+2}, \cdots, O_T \mid O_{t+1}, q_{t+1} = j, q_t = i, \lambda]P[O_{t+1} \mid q_{t+1} = j, q_t = i, \lambda]P[q_{t+1} = j \mid q_t = i, \lambda] =$
$\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)$.

pois $P[ABC|D] = P[A|BCD]P[B|CD]P[C|D]$, temos que $O_{t+2}, \cdots, O_T$ é independente de O_{t+1} e de q_t e ainda que O_{t+1}é independente de q_t.

Daí temos as fórmulas de Baum-Welch:

$$\begin{cases} \widehat{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{P[O|\lambda]} \\ \widehat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1}\alpha_t(i)\beta_t(i)} \\ \widehat{b}_j(k) = \frac{\sum_{t=1}^{T}\alpha_t(j)\beta_t(j)}{\sum_{t=1}^{T}\alpha_t(j)\beta_t(j)} \end{cases}$$
$$_{O_t=k}$$

De $\widehat{\pi}_i$, $\widehat{a}_{ij}$ e $\widehat{b}_j$ temos o novo modelo $\hat{\lambda}$.

Como o algoritmo de Baum-Welch é um caso particular do EM, temos que $P[O \mid \hat{\lambda}] \geq P[O \mid \lambda]$ e que a sequência de modelos $\hat{\lambda}_i$ obtidos neste método converge para λ^* tal que $P[O \mid \lambda^*]$ é um máximo local.

O algoritmo de Baum-Welch converge localmente (mas não globalmente) para um máximo. No entanto, Baum-Welch é rápido e geralmente o máximo local é tipicamente um bom máximo local.

Baum-Welch escalado

A partir das relações $\hat{\alpha}_t(i) = \left(\prod_{\tau=1}^{t} c_\tau\right)\alpha_t(i)$ e $\widehat{\beta}_t(i) = \left(\prod_{\tau=t}^{T} c_\tau\right)\beta_t(i)$ e $P[O \mid \lambda] = 1/\left(\prod_{t=1}^{T} c_t\right)$ podemos escalar as fórmulas de Baum-Welch escaladas (para evitar problemas de underflow). Vamos inicialmente calcular $\gamma_t(i)$ e $\zeta_t(i,j)$ em função de $\hat{\alpha}_t(i)$, $\widehat{\beta}_t(i)$ e c_t.

$$\begin{cases} \gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P[O|\lambda]} = \frac{\left(\widehat{\alpha}_t(i)/\prod_{\tau=1}^{t} c_\tau\right)\left(\widehat{\beta}_t(i)/\prod_{\tau=t}^{T} c_\tau\right)}{P[O|\lambda]} = \frac{P[O|\lambda]}{P[O|\lambda]}\frac{\widehat{\alpha}_t(i)\widehat{\beta}_t(i)}{c_t} = \frac{\widehat{\alpha}_t(i)\widehat{\beta}_t(i)}{c_t} \\ \zeta_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P[O|\lambda]} = \frac{\left(\widehat{\alpha}_t(i)/\prod_{\tau=1}^{t} c_\tau\right)\left(\widehat{\beta}_{t+1}(i)/\prod_{\tau=t+1}^{T} c_\tau\right)a_{ij}b_j(O_{t+1})}{P[O|\lambda]} = \widehat{\alpha}_t(i)\,\widehat{\beta}_{t+1}(i)\,a_{ij}b_j(O_{t+1}) \end{cases}$$

Logo os novos parâmetros do HMM são:

$$\begin{cases} \widehat{\pi}_i = \gamma_1(i) = \frac{\hat{\alpha}_1(i)\hat{\beta}_1(i)}{c_1} \\ \widehat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\zeta_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} = \frac{\sum_{t=1}^{T-1}\widehat{\alpha}_t(i)\widehat{\beta}_{t+1}(i)a_{ij}b_j(O_{t+1})}{\sum_{t=1}^{T-1}\frac{\hat{\alpha}_t(i)\hat{\beta}_t(i)}{c_t}} \\ \widehat{b}_j(k) = \frac{\sum_{t=1}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(i)} = \frac{\sum_{t=1}^{T}\frac{\hat{\alpha}_t(j)\hat{\beta}_t(j)}{c_t}}{\sum_{t=1}^{T}\frac{\hat{\alpha}_t(j)\hat{\beta}_t(j)}{c_t}} \end{cases}$$

Baum-Welch escalado e múltiplas observações

Em muitas aplicações, como reconhecimento de escrita e voz, ao invés de termos uma única longa seqüência de observação O, temos várias seqüências $\left\{O^{(m)}\right\}, m = 1 \ldots M$. Nesse caso temos que:

$P[O \mid \lambda] = \prod_{m=1}^{M} P[O^{(m)} \mid \lambda]$

Como as fórmulas de Baum-Welch calculam a freqüência de determinados eventos, basta somar os valores (no numerador e no denominador) de cada observação $O^{(m)}$.

$$\begin{cases} \widehat{\pi}_i = \sum_{m=1}^{M}\gamma_1^m(i)/\sum_{m=1}^{M}1 = \frac{1}{M}\sum_{m=1}^{M}\frac{\hat{\alpha}_1^m(i)\hat{\beta}_1^m(i)}{c_1^m} \\ \widehat{a}_{ij} = \frac{\sum_{m=1}^{M}\left(\sum_{t=1}^{T_m-1}\zeta_t^m(i,j)\right)}{\sum_{m=1}^{M}\left(\sum_{t=1}^{T_m-1}\gamma_t^m(i)\right)} = \frac{\sum_{m=1}^{M}\left(\sum_{t=1}^{T_m-1}\widehat{\alpha}_t^m(i)\widehat{\beta}_{t+1}^m(i)a_{ij}b_j(O_{t+1}^m)\right)}{\sum_{m=1}^{M}\left(\sum_{t=1}^{T_m-1}\frac{\hat{\alpha}_t^m(i)\hat{\beta}_t^m(i)}{c_t^m}\right)} \\ \widehat{b}_j(k) = \frac{\sum_{m=1}^{M}\left(\sum_{\substack{t=1 \\ O_t^m=k}}^{T_m}\gamma_t^m(i)\right)}{\sum_{m=1}^{M}\left(\sum_{t=1}^{T_m}\gamma_t^m(i)\right)} = \frac{\sum_{m=1}^{M}\left(\sum_{\substack{t=1 \\ O_t^m=k}}^{T_m}\frac{\hat{\alpha}_t^m(i)\hat{\beta}_t^m(i)}{c_t^m}\right)}{\sum_{m=1}^{M}\left(\sum_{t=1}^{T_m}\frac{\hat{\alpha}_t^m(i)\hat{\beta}_t^m(i)}{c_t^m}\right)} \end{cases}$$

Referências Bibliográficas

[1] Leonard Kleinrock, Queueing Systems, John Wyley & Sons, 1975.

[2] Barry R. James, Probabilidade: um curso em nível intermediário. Projeto Euclides, IMPA, 1981.

[3] R. Dugad and U. B. Desai, "A Tutorial on Hidden Markov Models. Tech. Report: SPANN-96.1, 1996. http://vision.ai.uiuc.edu/dugad/

[4] Simon Haykin, Redes Neurais, Bookman, Porto Alegre, 2001.

[5] Claude E. Shannon and Warren Weaver , A Teoria Matemática da Comunicação, Difel, Rio de Janeiro, 1975.

[6] M. C. Nechyba, Spring 2000 Lecture Notes - EEL6935: Machine Learning in Robotics II, http://www.mil.ufl.edu/~nechyba/readings_s00.html