AGRUPAMENTO HMM (*HMM Clustering*)

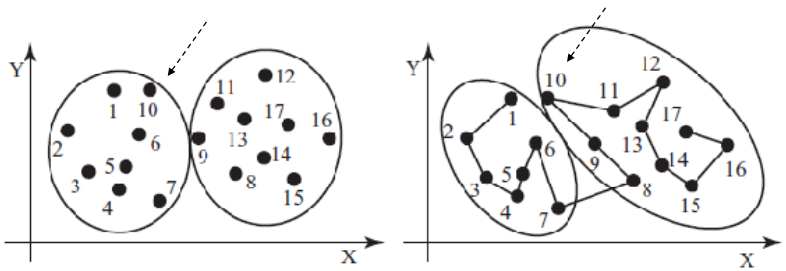
Haroldo Neves
Doutorando – Genética e Melhoramento Animal – FCAV/Unesp

Visão geral

- Motivação
  - Aplicações e Contextualização
- *Hidden Markov Models* (HMM)
  - Representação
  - Inferência usando HMM
- Agrupamento baseado em HMM

Motivação

- Agrupamento de seqüências



(a) A conventional clustering (b) A sequential clustering

Fonte: Xu & Wunsch (2009)

Aplicações

- Seqüências de DNA
- Processamento de voz
- Mineração de textos e web
- Análise de mercado de ações
- Perfil de consumidores
- Diagnóstico clínico
- Sensores robóticos

Como agrupar seqüências?

- Uso de medidas de proximidade
- Extração de atributos
- Modelos estatísticos
  - **Modelos Ocultos de Markov (HMM)**
  - Mistura de Cadeias de Markov
  - Mistura de Modelos Polinomiais
  - Mistura de Modelos ARMA

Hidden Markov Model (HMM)

- Talvez o mais empregado dentre os modelos para agrupamento de sequências (Xu & Wunsch, 2009)
- Começou a se popularizar em aplicações para reconhecimento de voz (Rabiner, 1989)

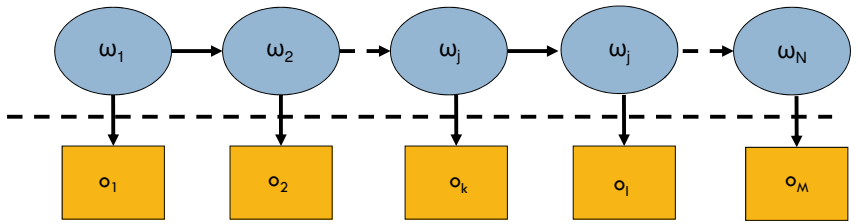
Cadeias de Markov

$$P(s_{n+1} \mid s_1, s_2, \ldots, s_n) = P(s_{n+1} \mid s_n)$$

Assume-se que a probabilidade de um dado estado no tempo *t* depende apenas do estado no tempo imediatamente anterior

Hidden Markov Models (HMM)

- Uma sequência de **estados não observáveis (*hidden*)**, em que cada estado é relacionado a outro processo estocástico que emite **símbolos observáveis.**



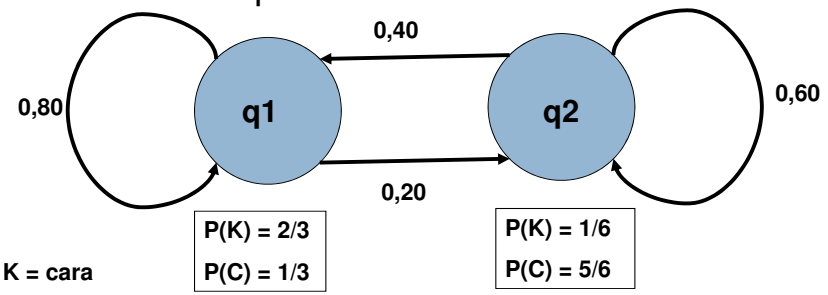
Definição formal de um HMM

- $\Omega = \{\omega_1, \omega_2, ..., \omega_N\}$ (N estados não-observáveis)
- $\mathbf{O} = \{o_1, o_2, ..., o_M\}$ (M símbolos observáveis)
- Três distribuições de probabilidade:
  - transição entre estados, $T=\{a_{ij}\}$:
    $a_{ij} = P(\omega_j(t+1)|\omega_i(t)), 1 \leq i,j \leq N; \quad \sum a_{ij} = 1$
  - emissão de símbolos, $E=\{b_{il}\}$:
    $b_{il} = P(o_l(t)|\omega_i(t)); \quad \sum b_{il} = 1$
  - estados iniciais, $\pi = \{\pi_i\}$:
    $\pi_i = P(\omega_i(1)), 1 \leq i \leq N$

$\lambda = \{T, E, \pi\}$

Um pequeno exemplo sobre HMM*

- Sejam q1 e q2 duas moedas "viciadas", lançadas uma por vez, de acordo com o modelo abaixo
- Uma pessoa observa o lançamento e desconhece as moedas que estão sendo usadas:



K = cara

C = coroa

*Adaptado de Singh(1999)

Um pequeno exemplo

- Por simplicidade, vamos assumir como probabilidades no lançamento inicial:

$p(q_1) = 1$ e $P(q_2) = 0$

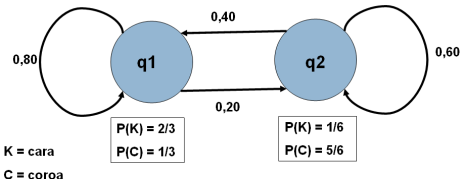
Parametrizando o exemplo como HMM

$$T_{NxN} = \begin{bmatrix} 0{,}80 & 0{,}20 \\ 0{,}40 & 0{,}60 \end{bmatrix}$$

$$E_{NxM} = \begin{bmatrix} 2/3 & 1/3 \\ 1/6 & 5/6 \end{bmatrix}$$

$$\pi' = [1 \quad 0]$$

$$\lambda_1 = \{T, E, \pi\}$$



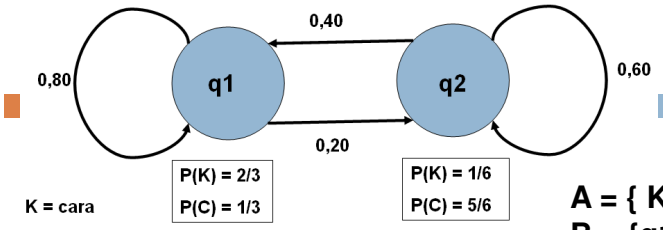
Um pequeno exemplo

- Após 5 lançamentos, obteve-se como resultado:

  { K, K, C, C, C } (evento A)

- Dado o modelo, pode ser de interesse a probabilidade de que uma dada seqüência de moedas tenha gerado o evento A, por exemplo:

  {q1,q1,q1,q2,q2} (evento B)



A = { K, K, C, C, C }
B = {q1,q1,q1,q2,q2}

P(A ∩ B) = P(A|B)* P(B)

$$P(A|B) = p(K|q1)*p(K|q1)*p(C|q1)*p(C|q2)*p(C|q2)$$
$$P(A|B) = (2/3)*(2/3)*(1/3)*(5/6)*(5/6) \approx \mathbf{0{,}1029}$$
$$P(B) = P(q1)*P(q1|q1)*P(q1|q1)*P(q2|q1)*P(q2|q2) =$$
$$P(B) = 1*0{,}8*0{,}8*0{,}2*0{,}6 = \mathbf{0{,}0768}$$

P(A ∩ B) ≈ 7,9*10^{-3}

***λ_1 é apenas um dentre os possíveis HMM, P(A ∩ B| λ_1) ≈ 7,9*10^{-3}**

Inferência em HMM (Xu & Wunsch, 2009)

- **Avaliação da verossimilhança,** $P(\mathbf{O}|\lambda)$: probabilidade de que uma seqüência **O** tenha sido produzida pelo HMM λ
- **Interpretação de estados** , $P(\Omega|\ \mathbf{O}, \lambda)$:

  qual a seqüência de estados mais provável , dado **O** e o λ
- **Treinamento de modelos**

  estimar parâmetros ótimos (λ) usando um conjunto de seqüências de treinamento

Avaliação da verossimilhança

- Avaliar a probalidade de uma sequência **o** de símbolos observados, para um dado modelo, $P(\mathbf{o}|\lambda_i)$.
- Como os estados não são conhecidos

$$P(\mathbf{o}|\lambda_i) = \sum_f P(\mathbf{o}, \omega_f|\ \lambda_i)$$
$$= \sum_f P(\mathbf{o}|\ \omega_f, \lambda_i)\ P\ (\omega_f|\ \lambda_i)$$

- Sendo L, o tamanho da sequência, há N^L sequências possíveis de estados
- Enumeração exaustiva é inviável na prática
- Solução: Programação dinâmica

Algoritmos **Forward** ou **Backward**, $O(N^2L)$ (maiores detalhes em Xu & Wunsch, 2009)

Interpretação de estados

- Podemos ter interesse em identificar a sequência ω^* com maior probabilidade de ter gerado **o**

$$\omega^* = \arg_{\omega}\max\ (P(\boldsymbol{\omega} \mid \mathbf{o}, \boldsymbol{\lambda}))$$

- Solução eficiente por meio de programação dinâmica

Algoritmo de Viterbi (Viterbi, 1967)

Treinamento de Modelos

- Não há solução análitica para identificar o modelo (λ') de máxima verossimilhança

$$\lambda' = \arg\max_{\lambda} P(O \mid \lambda)$$

- Atualização recursiva dos parâmetros de modo similar ao visto anteriormente no caso de EM
- **Algoritmo Baum-Welch** (Rabiner, 1989):

estima-se parâmetros (**T, E, π**) usando apenas sequências observadas (treinamento)

Agrupamento baseado em HMM

- Idéia básica:

Dentre um conjunto de **L** sequências observadas, elas podem ter diferentes probabilidades de terem sido geradas por um HMM , i.e. P(**O** | **λ**)

- Podemos agrupar sequências observadas em diferentes HMMs
- Não é um problema trivial:

As sequências podem ter diferentes tamanhos e não há uma métrica natural para a comparação das mesmas.

Agrupamento baseado em HMM

- Proposta de Smyth (1997): mistura de K modelos HMM
- Pode-se representar a densidade de probalidade de tal mistura por:

$$p(o \mid \theta) = \sum_{i=1}^{K} p(\mathbf{o} \mid C_i, \theta_i) P(C_i)$$

$$\sum_{i=1}^{K} P(Ci) = 1$$

C_i refere-se ao i-ésimo HMM, com parâmetros θ_i

- A interpretação da fórmula é similar àquela do contexto de mistura de gaussianas (visto no caso de EM), a diferença básica é que **o** refere-se a uma sequência.

Agrupamento baseado em HMM

- Se o objetivo é definir um modelo geral para as L sequências, K diferentes HMM poderiam ser reunidos num só HMM (**HMM composto,** cujos parâmetros são representados por θ) (Smyth, 1997)

Num HMM composto, para K=2, a matriz de transição T poderia ser representada por:

$$T = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix}$$

T1 e T2 são as matrizes de probabilidades de transição de cada HMM

Agrupamento baseado em HMM

- Versão resumida do algoritmo (Smyth,1997):

1. Modelar **cada** sequência o_i, $1 \leq i \leq L$, com um HMM composto por N estados e com parâmetros λ_j .
2. Calcular $\log(P(o_i|\lambda_j))$, $1 \leq j \leq L$
3. Agrupar as sequências em K clusters, usando uma medida de distância baseada em $\log(P(o_i|\lambda_j))$.
4. Modelar cada cluster obtido em (3) com um HMM e inicializar um HMM-composto
5. **Treinar o HMM-composto com o algoritmo Baum-Welch (até convergência)**

Agrupamento baseado em HMM

- Em (3) Smyth (1997) utilizou como medida de proximidade:

$$D(\lambda_i, \lambda_j) = [\, P(o_i|\lambda_j) + P(o_j|\lambda_i) \,] \,/\, 2$$

em conjunto com agrupamento hierárquico (*complete linkage*).

- No algoritmo apresentado, pressupôs-se K e N conhecidos *a priori*
- Uma extensão natural deste algoritmo, seria estimar K e N na etapa de treinamento. (Smyth, 1997)

Agrupamento baseado em HMM

- A discussão ao longo do seminário baseou-se em probabilidades discretas de emissão de símbolos (E)
- O caso de observações com distribuição contínua segue basicamente os mesmos princípios vistos anteriormente, exceto que E pode ser modelada como uma mistura de distribuições (e.g. mistura de gaussianas)

Obrigado

Referências

- Xu, R., Wunsch, D., Clustering, IEEE Press, 2009.
- Rabiner , L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE , 77 : 257 – 286 .
- Viterbi , A. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transactions on Information Theory, IT - 13 :260 – 269.

Referências

- Singh, M. Profile Hidden Markov Models. Course Notes. COS 597c: Topics in Computational Molecular Biology, Princeton, 1999.
- Smyth , P. (1997). Clustering sequences with hidden Markov models . In Advances in Neural Information Processing 9 , M. Mozer , M. Jordan and T. Petsche , Eds., Cambridge, MA : MIT Press , pp. 648 – 654

Material complementar*

- Os slides a seguir contêm uma versão resumida dos algoritmos citados ao longo da apresentação, utilizados em problemas de inferência em HMM:
  - Avaliação de Verossimilhança:
    Algoritmo Forward e Algoritmo Backward
  - Interpretação de Estados:
    Algoritmo de Viterbi
  - Treinamento de Modelos:
    Algoritmo Baum-Welch

*No livro de Xu & Wunsch (2009), pode-se encontrar material mais detalhado sobre tais algoritmos (cap. 7), bem como outras referências.

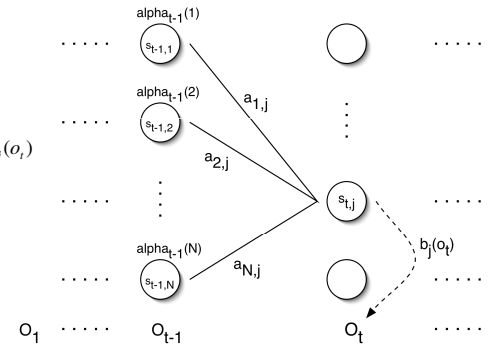
Algoritmos Forward e Backward

- Podem ser usados para avaliar a probalidade de que uma sequência **o** de símbolos observados tenha sido produzida por um dado HMM com parâmetros **λ**, i.e. P(**o** | **λ**).
- Possibilitam calcular P(**o** | **λ**) com complexidade O(N²L), sendo N o número de possíveis estados e L o tamanho da sequência.

Algoritmo Forward

$$\alpha_t(i) = P(o_1..o_t, q_t = s_i \mid \lambda)$$

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



Notação:

a_{ij} representa a probabilidade de transição do estado i para j.

$b_j(ot)$ é a probabilidade de que um estado j gere uma observação o, no tempo t

Algoritmo Forward

- Inicialização: $\alpha_1(i) = \pi_i b_i(o_1) \quad 1 \le i \le N$
- Indução:

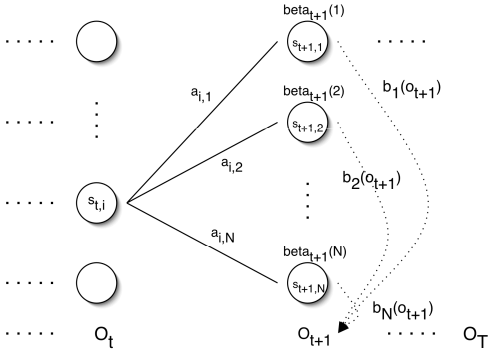
$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij}\right] b_j(o_t) \quad 2 \le t \le T, 1 \le j \le N$$

- Terminação:

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

Algoritmo Backward

$$\beta_t(i) = P(o_{t+1}...o_T \mid q_t = s_i, \lambda)$$



$$\beta_t(i) = \left[\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)\right]$$

Algoritmo Backward

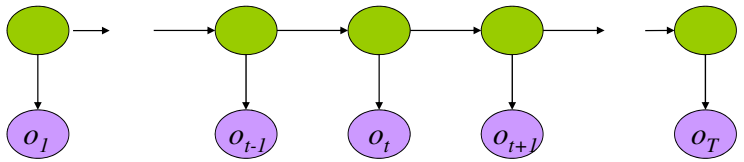
- Inicialização: $\beta_T(i) = 1, \quad 1 \le i \le N$
- Indução:

$$\beta_t(i) = \left[\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \right] \quad t = T-1 \ldots 1, 1 \le i \le N$$

- Terminação:

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

Algoritmo de Viterbi*

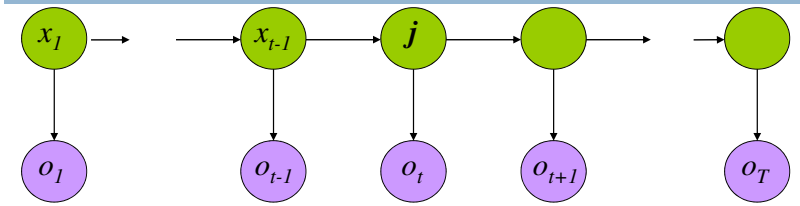


- Utilizado a fim de encontrar a sequência de estados X que melhor se ajusta às observações.
- Ou seja: $\arg\max_X P(X \mid O)$

Notação:
a_{ij} representa a probabilidade de transição do estado i para j.
$b_j o_t$ é a probabilidade de que um estado j gere uma observação o, no tempo t

*Adaptado de Blei (1999). O material original está disponível em :
*www-nlp.stanford.edu/fsnlp/hmm-chap/**blei**-hmm-ch9.ppt*

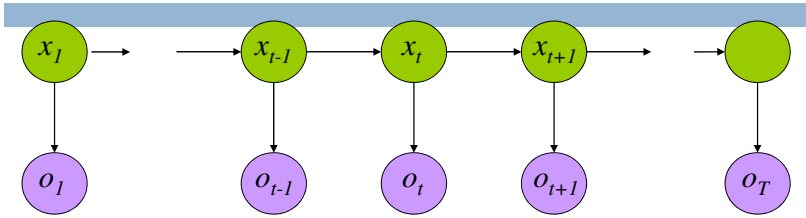
Algoritmo de Viterbi



$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

A sequência de estados X que maximiza a probabilidade das observações até o tempo t-1, dado que x_t= j, e observando o_t no tempo t

Algoritmo de Viterbi



$$\delta_j(t) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

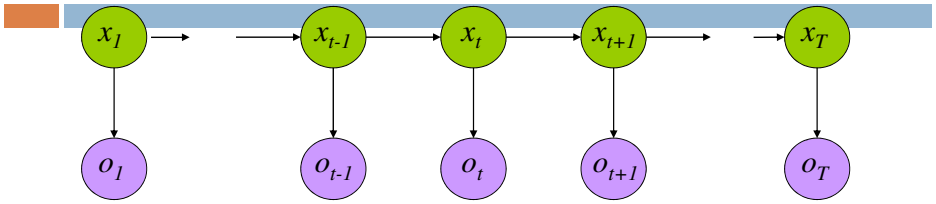
$$\delta_j(1) = \pi_j b_j o(1), \quad 1 \le j \le N$$
$$\psi_j(1) = 0, \quad 1 \le j \le N$$

Inicialização

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{j o_{t+1}}$$
$$\psi_j(t+1) = \arg\max_i \delta_i(t) a_{ij} b_{j o_{t+1}}$$

Calcular recursivamente, de t=2,…T

Algoritmo de Viterbi



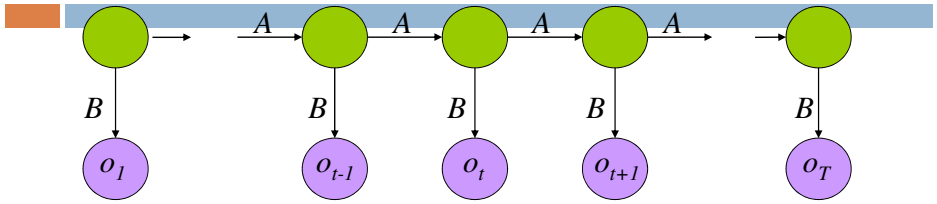
$$\hat{X}_T = \arg\max_i \delta_i(T)$$

$$\hat{X}_t = \psi_{\hat{X}_{t+1}}(t+1)$$

$$P(\hat{X}) = \arg\max_i \delta_i(T)$$

Obter a sequência mais provável, para t = T-1 até 1 .

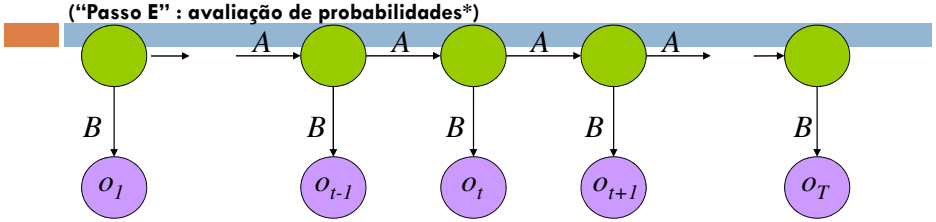
Estimação de Parâmetros



- Dada uma sequência observada, como encontar o modelo mais verossímil?
- Dados **λ** e a sequência **o**, atualizar T, E e π de modo que melhor se ajustem às observações.
- Algoritmo de Baum-Welch

Algoritmo de Baum-Welch

("Passo E" : avaliação de probabilidades*)



$$p_t(i,j) = P(x_{i(t)}, x_{j(t+1)} \mid o, \lambda)$$

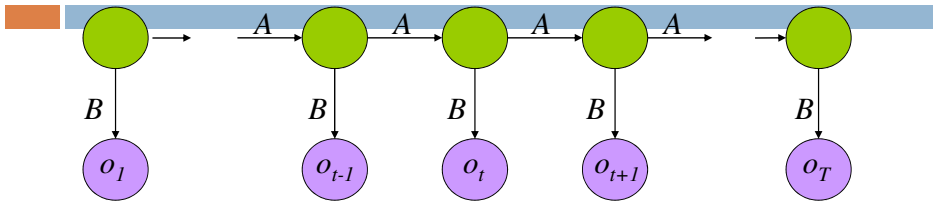
$$p_t(i,j) = \frac{\alpha_i(t)a_{ij}b_{jo_{t+1}}\beta_j(t+1)}{\sum_{m=1\ldots N}\alpha_m(t)\beta_m(t)}$$

$$\gamma_i(t) = \sum_{j=1\ldots N} p_t(i,j)$$

*Requer atualização das variáveis α_i (algoritmo forward) e β_j (algoritmo backward)

Algoritmo de Baum-Welch

("Passo M" : atualização de parâmetros)



$$\hat{\pi}_i = \gamma_i(1)$$

Probabilidades dos estados iniciais

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T}\gamma_i(t)}$$

Probabilidades de transição entre estados

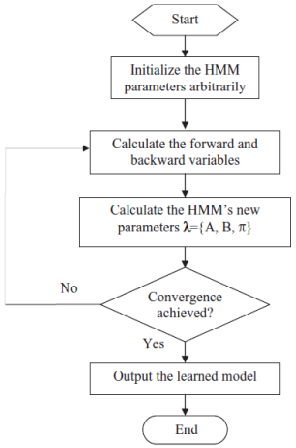
$$\hat{b}_{ik} = \frac{\sum_{\{t:o_t=k\}}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_i(t)}$$

Probabilidades de emissão de símbolos

Algoritmo de Baum-Welch

- Alterar entre os passos “E” e “M” até convergência

Algoritmo Baum-Welch: visão geral*



*Retirado de Xu & Wunsch (2009)