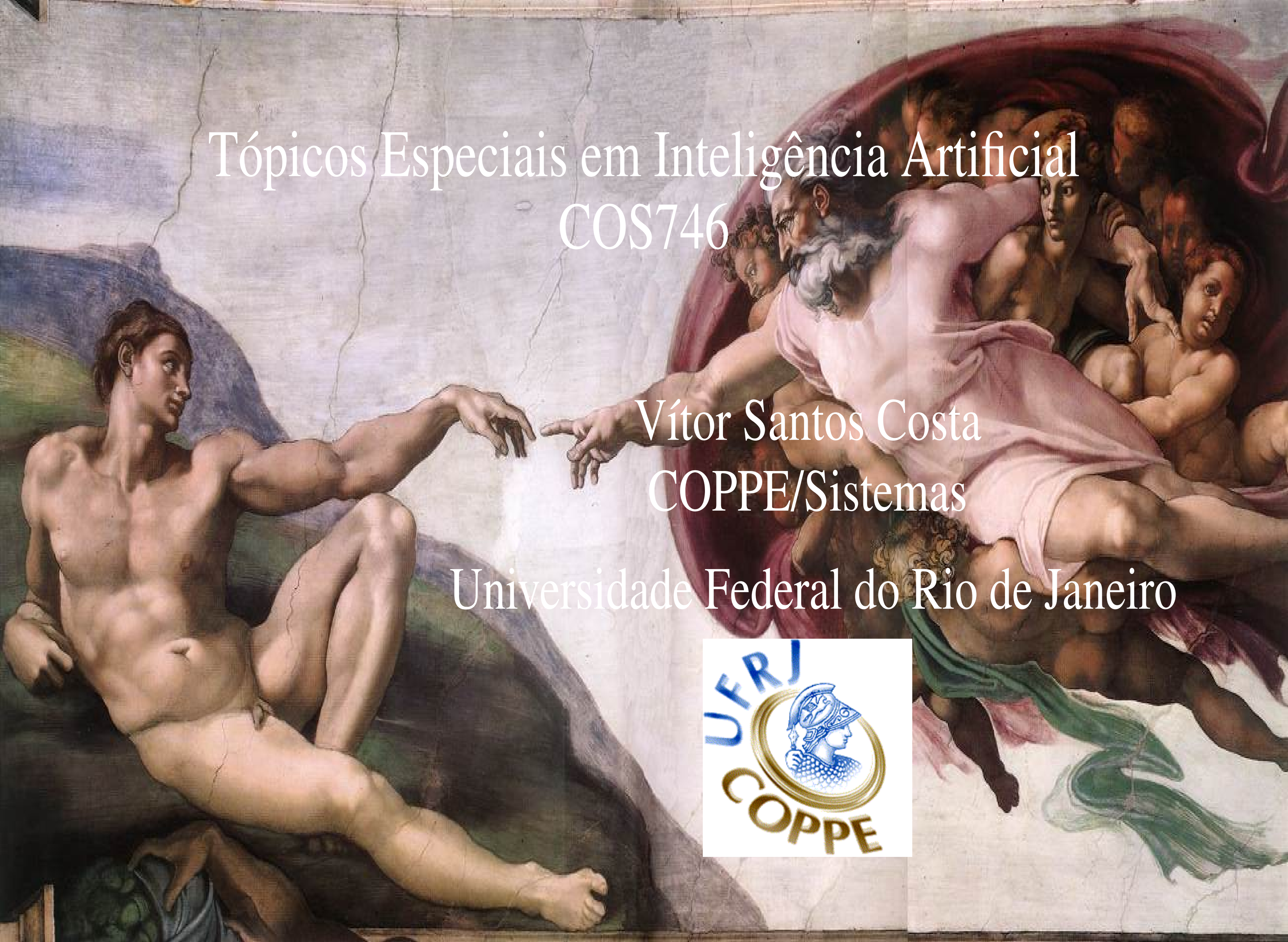


Tópicos Especiais em Inteligência Artificial
COS746

Vítor Santos Costa
COPPE/Sistemas

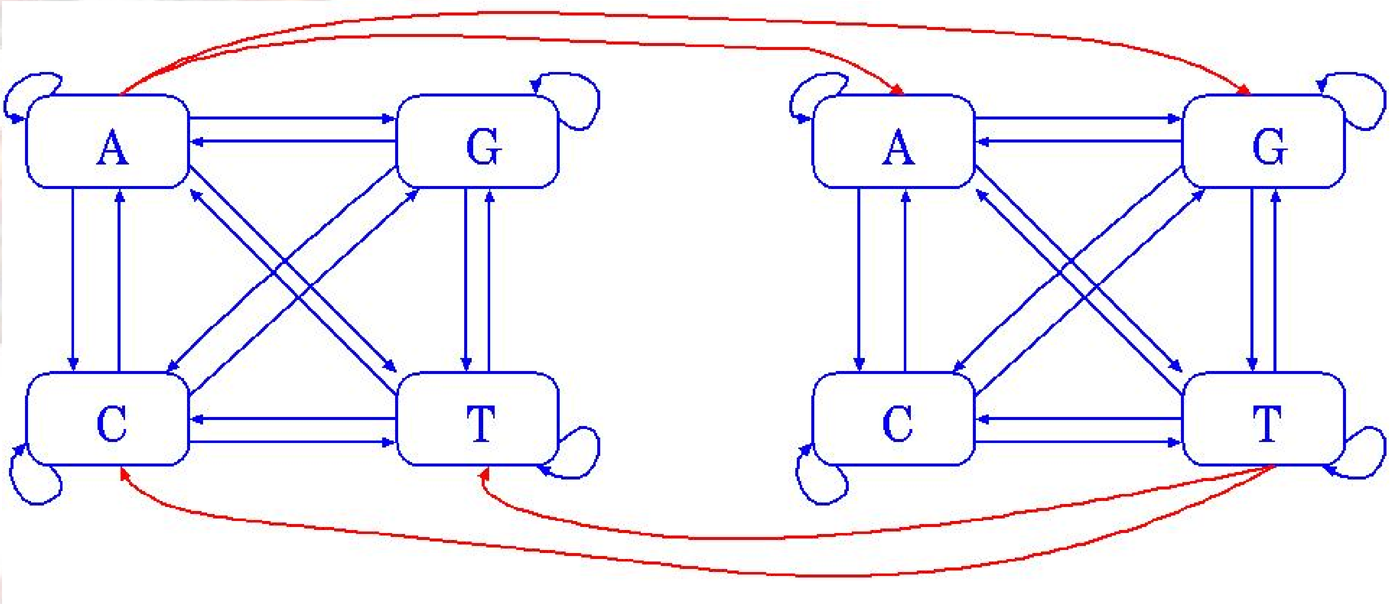
Universidade Federal do Rio de Janeiro

Agradecimento

- Copiado dos slides de Mark Craven/C. David Page para BMI/CS 576, UW-Madison

HMMs

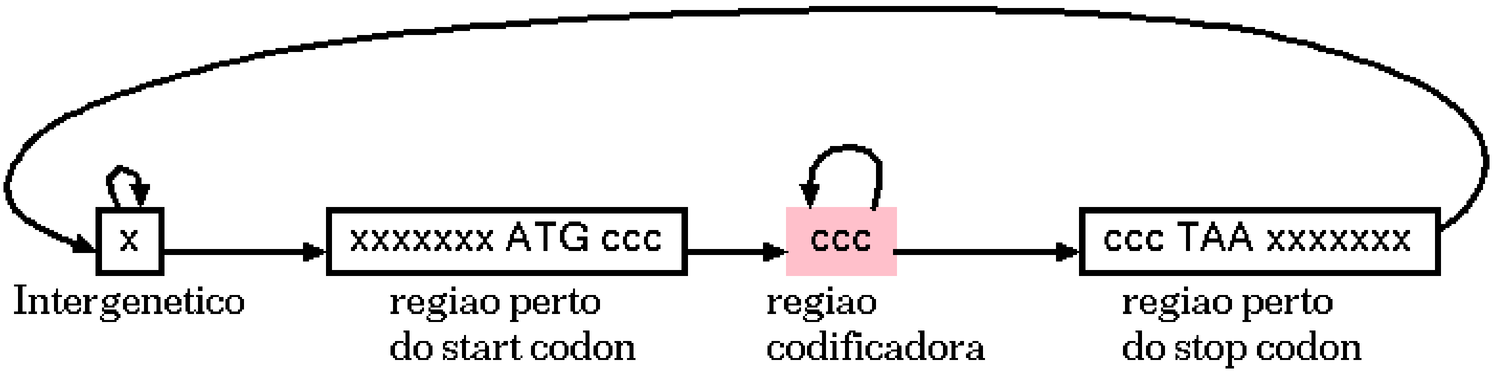
- Dado um T na nossa sequência de leitura, quem o emitiu?

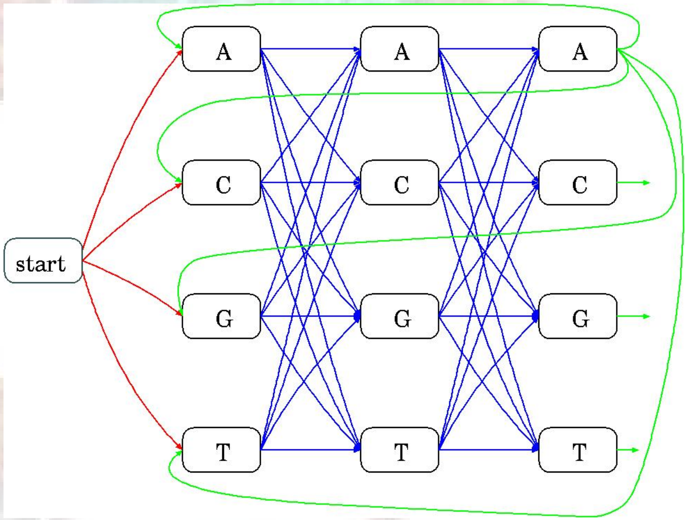


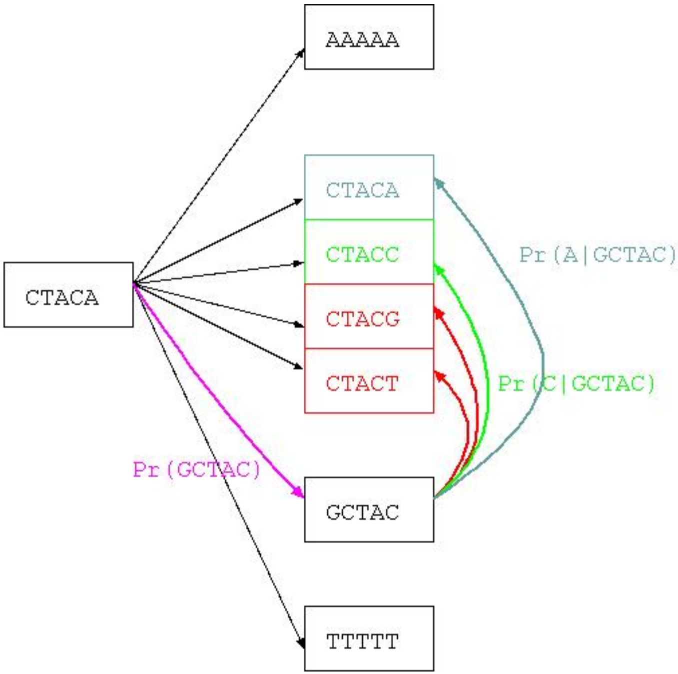
Estado Escondido

- Vamos distinguir entre as partes observadas de um problema e as parte escondidas.
- Nos modelos que consideramos antes, é claro que estado é responsável por que parte da sequência
- no modelo anterior, há múltiplos estados que podem causar cada parte de uma sequência observada:
  - ⋆ é a parte escondida do problema

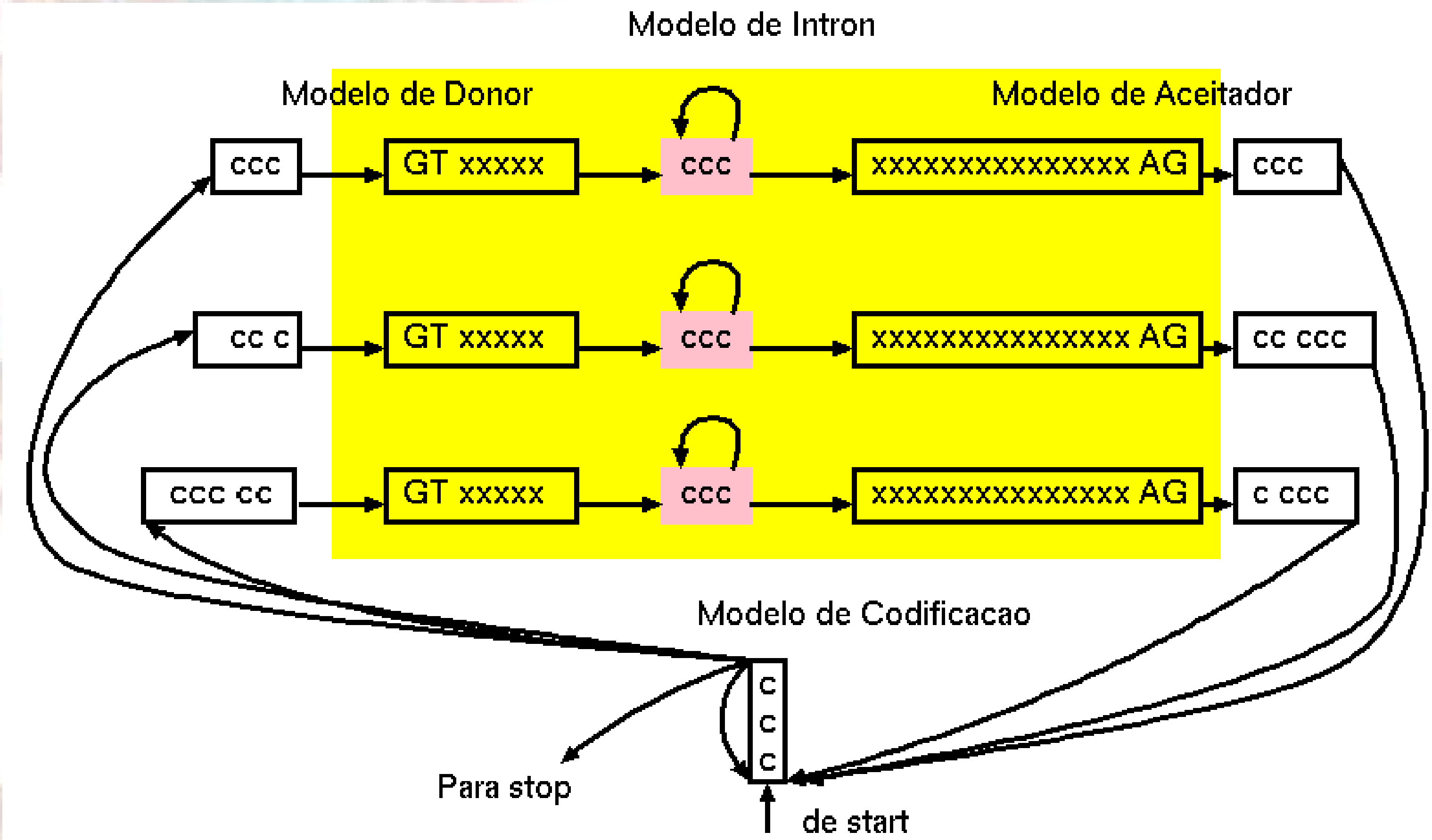
HMM Simples para Encontrar Genes







HMM Simples para Eucariotes



Parâmetros de um HMM

- Num modelo de cadeias de Markov, temos probabilidades de transição:

$$a_{kl} = Pr(\pi_i = l | \pi_{i-1} = k)$$

  - ⋆ probabilidade de uma transição do estado k para l
  - ⋆ π representa um caminho (sequência de estados) através do modelo.

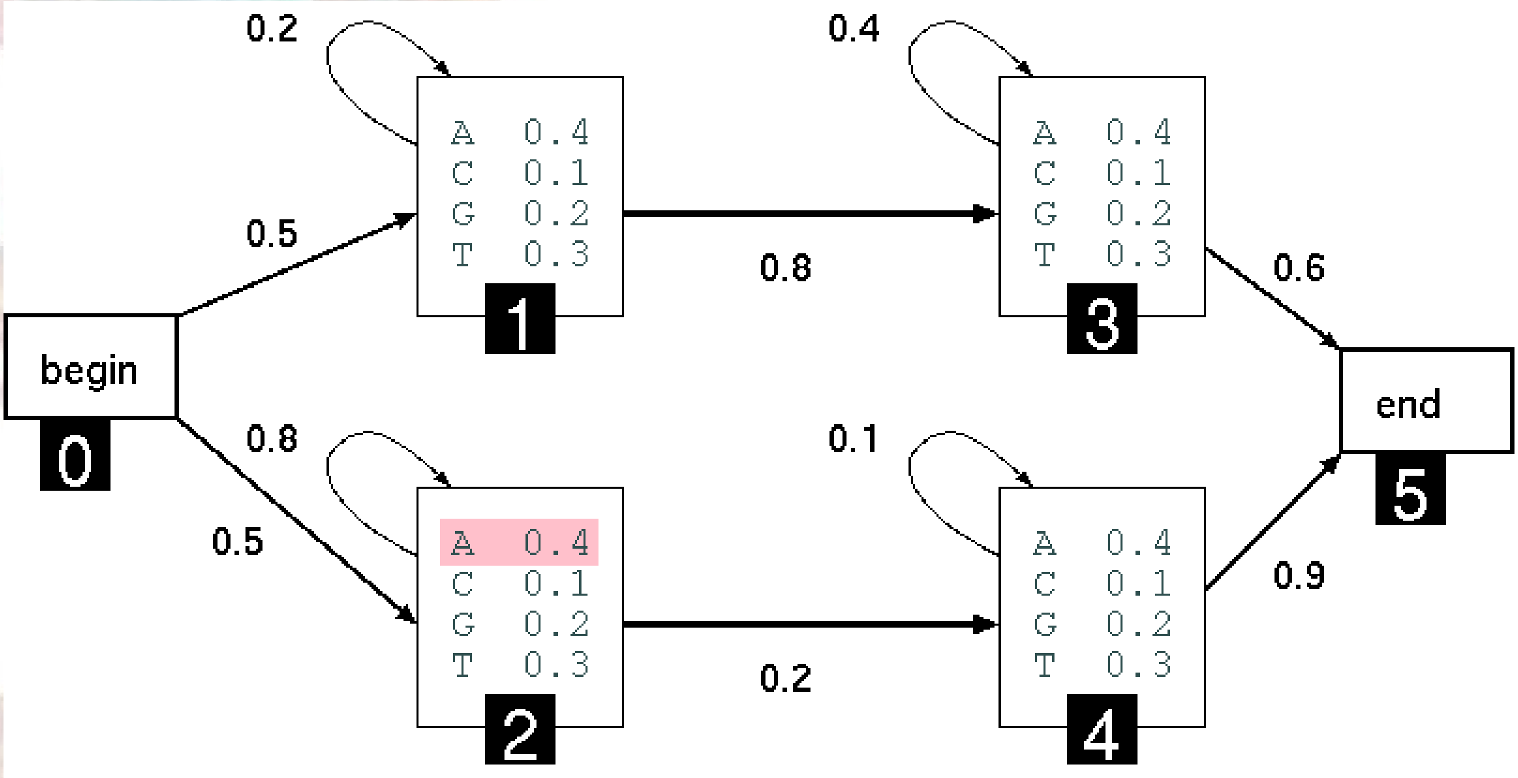
- Como separamos estados e caracteres, podemos também ter probabilidades de emissão:

$$e_k(b) = Pr(x_i = b | \pi_i = k)$$

  probabilidade de emitir o caracter $\mathtt{b}$ no estado k.

Um HMM Simples

- a_{13} probabilidade de transição do estado 1 para o estado 3
- $e_2(A)$ probabilidade de emitir o caracter A no estado 2



Três Questões Importantes

- Qual é a probabilidade de uma sequência dada?
  - ⋆ O algoritmo Forward
- Qual é o caminho mais provável para gerar uma sequência dada?
  - ⋆ O algoritmo de Viterbi
- Como podemos aprender os parâmetros de um HMM dada um conjunto de sequências?
  - ⋆ O algoritmo Forward-Backward (Baum-Welch)

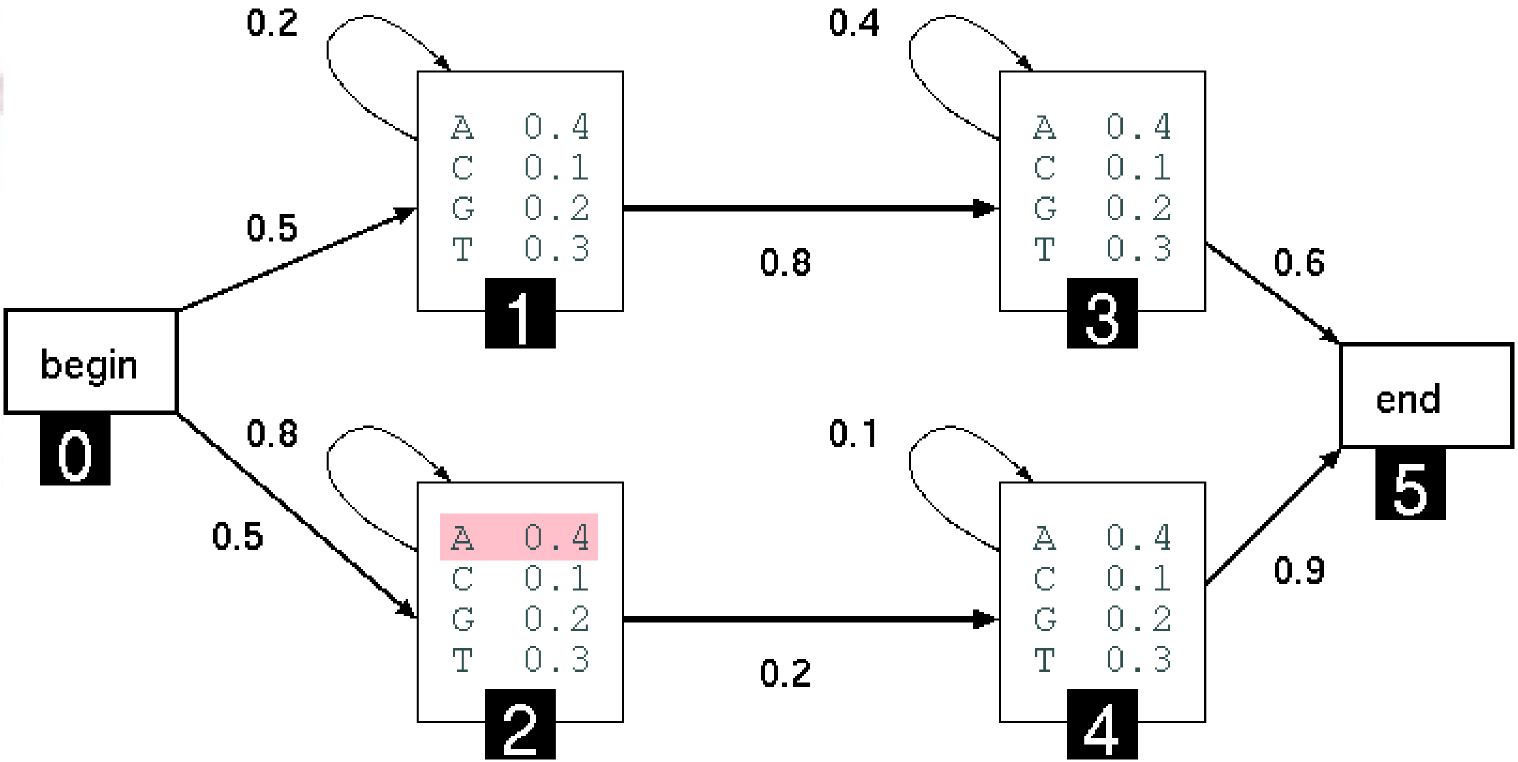
Qual é a Probabilidade de uma dada sequência

- A probabilidade de que o caminho $\pi_0 \ldots \pi_N$ seja tomado e a sequência $x_1 \ldots x_L$ seja gerada:

$$Pr(x_1 \ldots x_L, \pi_0 \ldots \pi_N) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

(assumimos que begin e end são os únicos estados silenciosos no caminho)

Qual é a Probabilidade de uma dada sequência



$$
\begin{aligned}
Pr(AAC, \pi) &= a_{01} \times e_1(\mathtt{A}) \times a_{11} \times e_1(\mathtt{A}) \times a_{13} \times e_3(\mathtt{C}) \times a_{35} \\
&= 0.5 \times 0.4 \times 0.2 \times 0.4 \times 0.8 \times 0.3 \times 0.6
\end{aligned}
$$

Qual é a Probabilidade de uma dada sequência

- a probabilidade sobre *todos* os caminhos é:

$$Pr(x_1 \ldots x_L) = \sum_{pi} Pr(x_1 \ldots x_l, \pi_0 \ldots \pi_N)$$

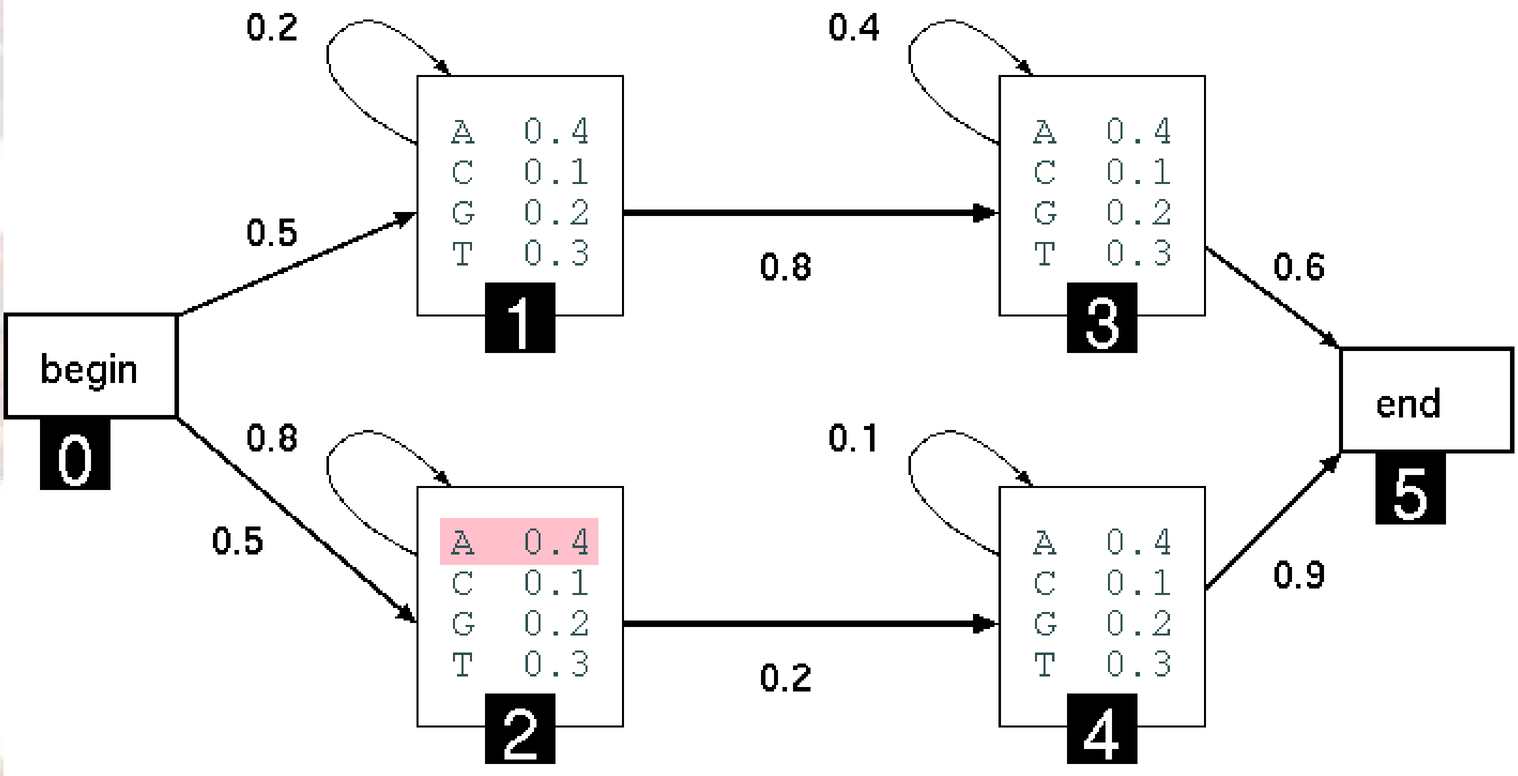
- mas o número de caminhos pode ser exponencial no tamanho da sequência
- o algoritmo Forward permite computar isto eficientemente

O Algoritmo Forward

- defina $f_k(i)$ como sendo a probabilidade de estar no estado k tendo observado os primeiros caracteres i de x
- queremos computar $f_N(L)$, a probabilidade de estar no estado final tendo observado todo x
- podemos usar recursão

O Algoritmo Forward

- por causa da propriedade de Markov, não temos que enumerar cada caminho explicitamente: podemos usar programação dinâmica



- eg, computar $f_4(i)$ com $f_2(i-1)$, $f_4(i-1)$

O Algoritmo Forward

- inicialização:
  - ⋆ $f(0) = 1$ probabilidade que estejamos no estado começo e tenhamos observado 0 caracteres da sequência
  - ⋆ $f_k(0) = 0$, para k que não são estados silenciosos.

O Algoritmo Forward

- Recursão para emitir estados ($i = 1 \ldots L$)

$$f_l(i) = e_l(i) \sum_k f_k(i-1) a_{kl}$$

- recursão para estados silenciosos:

$$f_l(i) = \sum_k f_k(i) a_{kl}$$

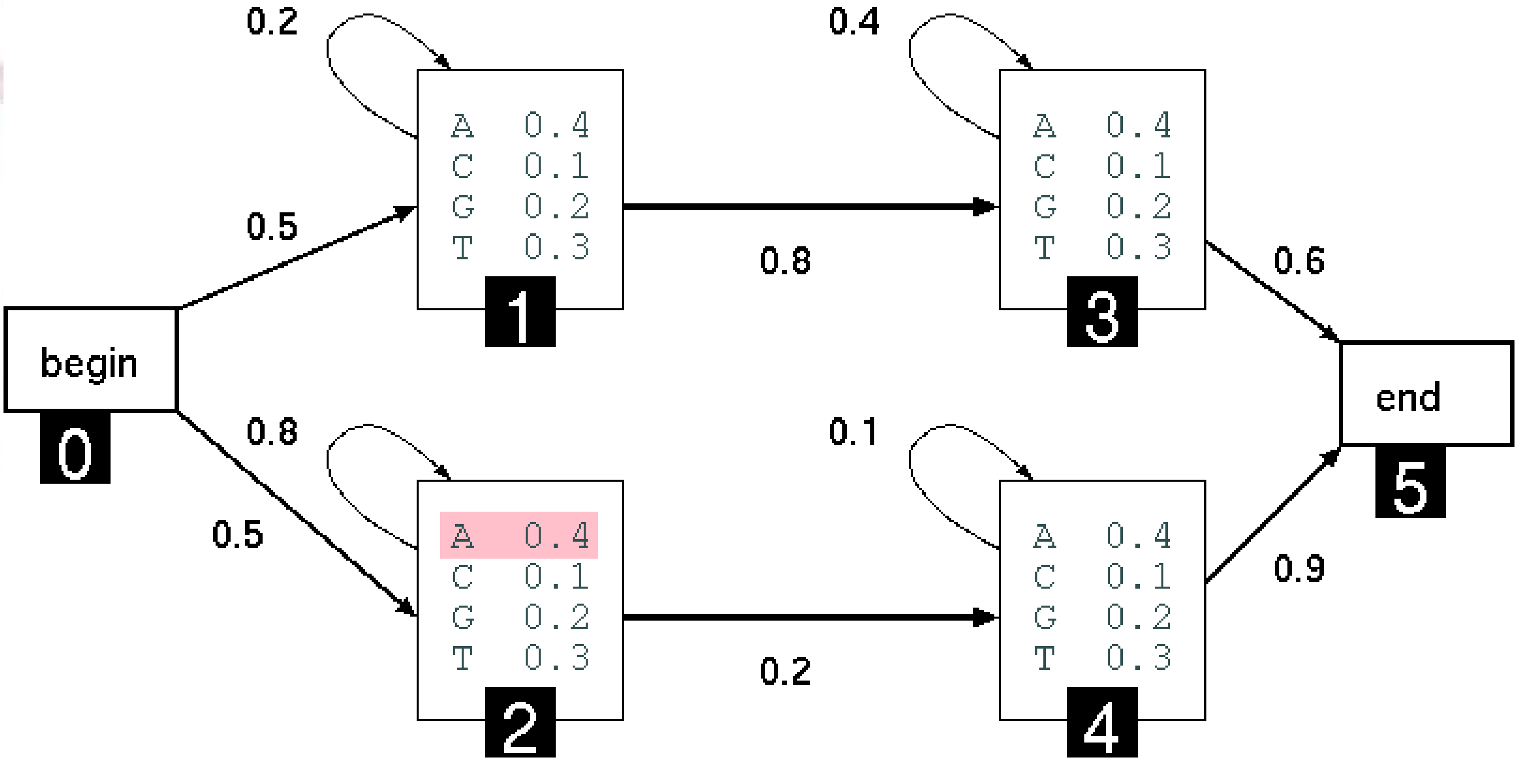
O Algoritmo Forward

- Terminação

$$Pr(x) = Pr(x_1 \ldots x_L) = f_N(L) = \sum_k f_k(L) a_{kN}$$

- probabilidade que estejamos no estado final e tenhamos observado toda a sequência

O Algoritmo Forward: Exemplo



- dada a sequência $x = \texttt{TAGA}$

O Algoritmo Forward: Exemplo

- dada a sequência $x = \texttt{TAGA}$
- inicialização

  $f_0(0) = 1 \qquad f_1(0) \quad \bullet \quad f_5(0) = 0$

- computando outros valores:

$$
\begin{aligned}
f_1(1) = \quad & e_1(T) \times (f_0(0) \times a_{01} + f_1(0)a_{11}) = \\
& 0.3 \times (1 \times 0.5 + 0 \times 0.2) = 0.15 \\
f_2(1) = \quad & 0.4 \times (1 \times 0.5 + 0 \times 0.8) \\
f_1(2) = \quad & e_1(\texttt{A}) \times f_0(l) \times a_{01} + f(l)a_{11}) = \\
& 0.4 \times (0 \times 0.5 + 0.15 \times 0.2)
\end{aligned}
$$

$$\bullet\bullet\bullet$$

$$Pr(\texttt{TAGA}) = \; f_5(4) = (f_3(4) \times a_{35} + f_4(4)a_{45})$$

O Algoritmo Forward: Notas

- em alguns casos o algoritmo pode ficar mais eficiente tomando em conta o número mínimo de passos que são precisos para atingir um estado
- para este HMM não precisamos de inicializar ou computar os valores:

$$f_3(0),\ f_4(0)$$
$$f_5(0),\ f_5(1)$$

Três Questões Importantes

- Qual é a probabilidade de uma sequência dada?
- Qual é o caminho mais provável para gerar uma sequência dada?
- Como podemos aprender os parâmetros de um HMM dada um conjunto de sequências?

Encontrando o Caminho Mais Provável: O Algoritmo de Viterbi

- defina $v_k(i)$ como sendo a probabilidade do *caminho mais provável* responsável pelos primeiros i caracteres de x e terminado num estado$k$
- queremos computar $v_N(L)$, a probabilidade do caminho mais provável responsável por toda a sequência e terminando no estado final
- podemos usar recursão
- podemos usar PD para encontrar $v_N(L)$ eficientemente

Encontrando o Caminho Mais Provável: O Algoritmo de Viterbi

- inicialização:
  - $v_0(0) = 1$
  - $v_k(0) = 0$, para k que não são estados silenciosos

O Algoritmo de Viterbi

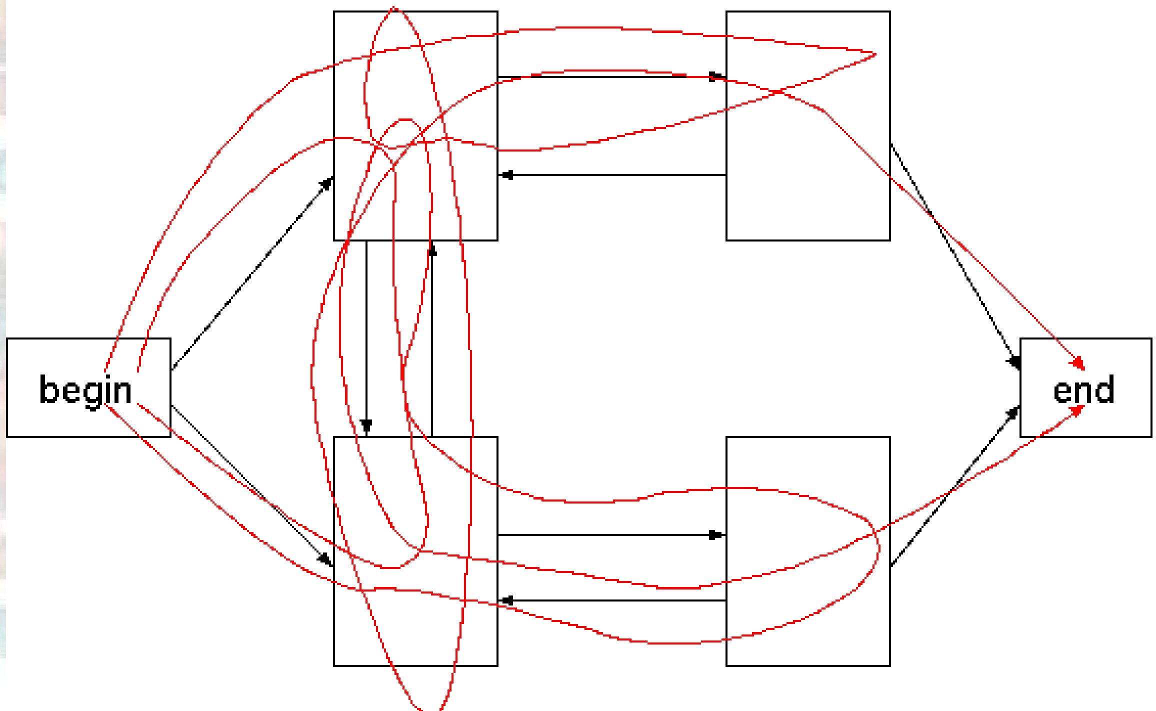
- recursão para estados que emitem ($i = 1 \ldots L$):
  - ⋆ $v_l(i) = e_l(x_i) max_k [v_k(i-1) a_{kl}]$
  - ⋆ $ptr_l(i) = argmax_k [v_k(i-1) a_{kl}]$ mantém o caminho mais provável
- recursão para estados silenciosos:
  - ⋆ $v_l(i) = max_k [v_k(i) a_{kl}]$
  - ⋆ $ptr_l(i) = argmax_k [v_k(i) a_{kl}]$ mantém o caminho mais provável

O Algoritmo de Viterbi

- terminação:
  - $Pr(x, \pi) = max_k(v_k(L)a_{kN})$
  - $\pi_L = argmax_k(v_k(L)a_{kN})$
- traceback: seguir ponteiros de volta começando em π_L

Algoritmos de Forward & Viterbi

- Os algoritmos de Forward/Viterbi consideram todos os caminhos possíveis para uma sequência
  - ⋆ Forward para encontrar probabilidade de uma sequência
  - ⋆ Viterbi para encontrar caminho mais provável
- considere uma sequência de tamanho 4:



Três Questões Importantes

- Qual é a probabilidade de uma sequência dada?
- Qual é o caminho mais provável para gerar uma sequência dada?
- Como podemos aprender os parâmetros de um HMM dada um conjunto de sequências?

A Tarefa de Aprendizagem

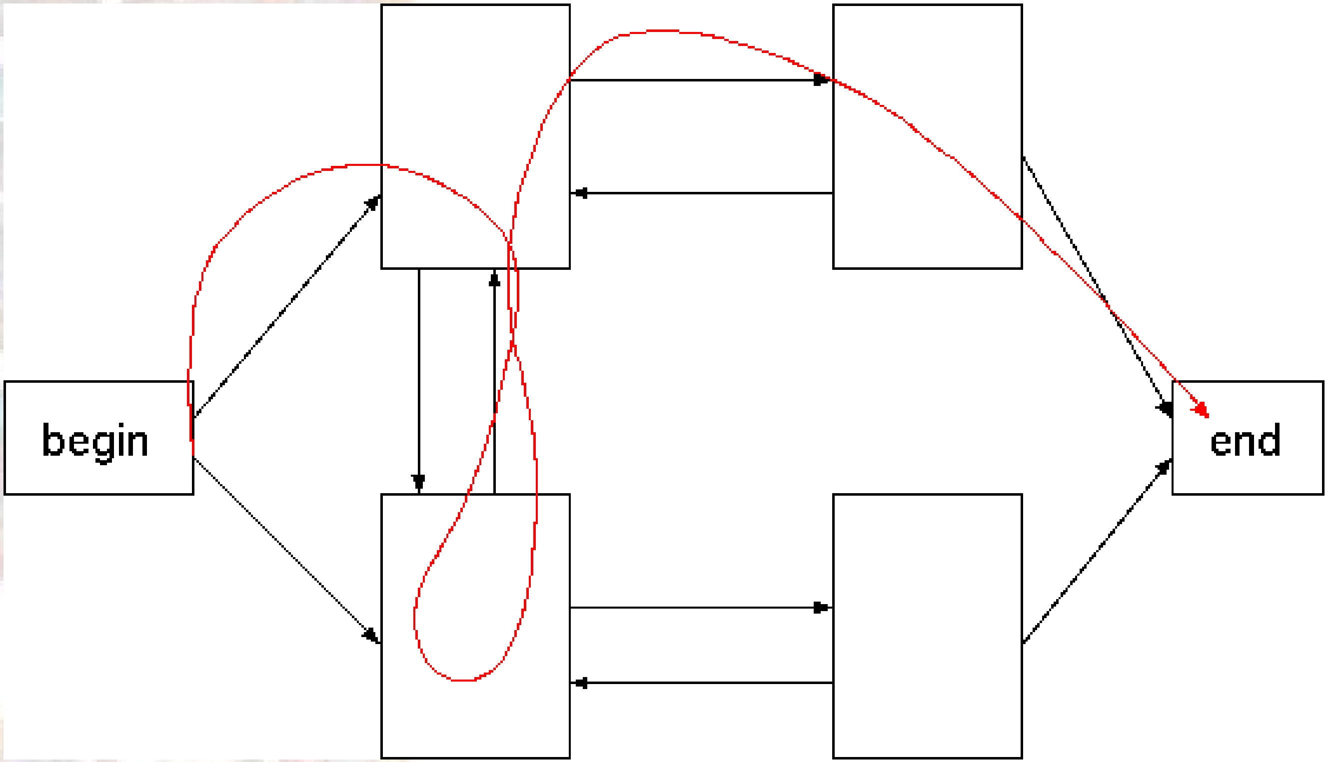
- dados:
  - ⋆ um modelo
  - ⋆ um conjunto de sequências (o conjunto de treino)
- faça:
  - ⋆ encontre os parâmetros mais prováveis para explicar a sequência de treino
- o objectivo é encontrar um modelo que *generalize* bem para sequências que nunca vimos antes.

Parâmetros de Aprendizagem

- Se sabemos o caminho para atingir o estado de cada sequência de treino, aprender os parâmetros do modelo é simples:
  - ⋆ não há estados escondidos no treino
  - ⋆ contar quantas vezes cada parâmetro é usado
  - ⋆ normalize/amaciar para obter probabilidades
  - ⋆ processar exactamente como um modelo de cadeias de Markov
- se não sabemos o caminho para cada sequência de treino, como determinar as contagens?
  - ⋆ ideia base: estimar a contagem considerando todos os caminhos pesados pela sua probabilidade

Aprender sem Estado Escondido

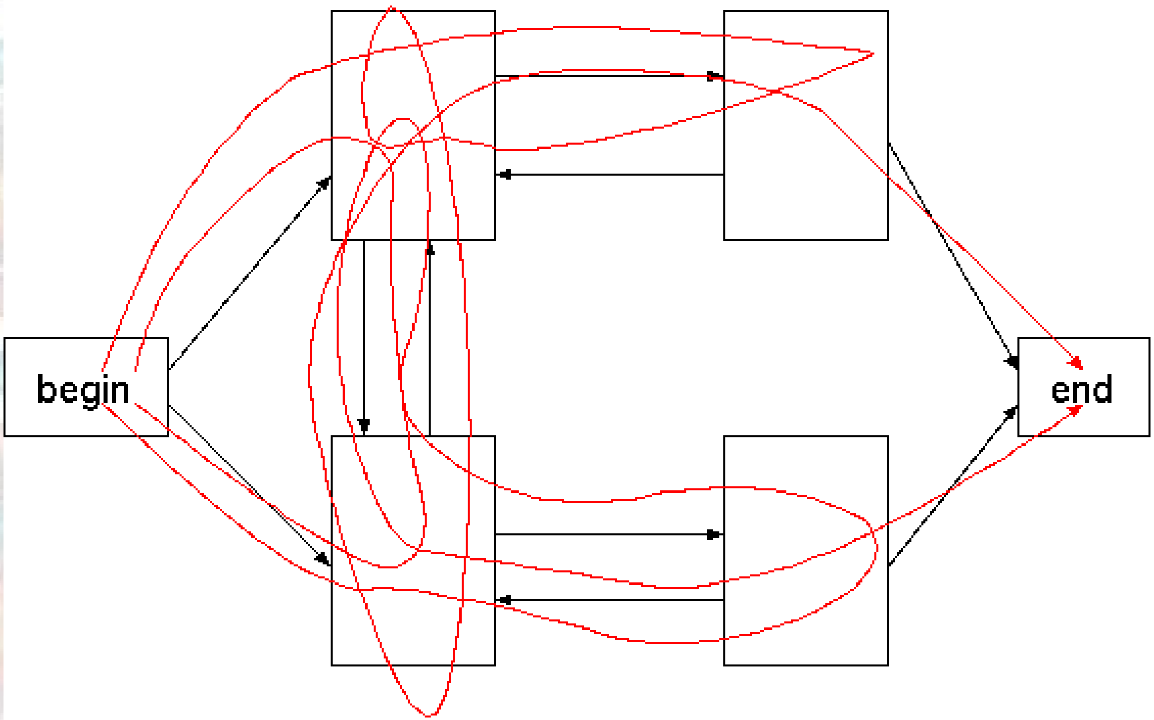
- aprender é fácil se sabemos o caminho correcto para cada sequência do conjunto de treino



- estimar parâmetros contando o número de vezes que o parâmetro é usado no conjunto de treino

Aprender com Estado Escondido

- se não sabemos o caminho correcto para cada sequência no conjunto de treino, considerar todos os caminhos possíveis para a sequência



- estimar parâmetros usando um procedimento que conta o número *esperado* de vezes que cada parâmetro é usado através do conjunto de treino

Aprender Parâmetros:
o Algoritmo de Baum-Welch

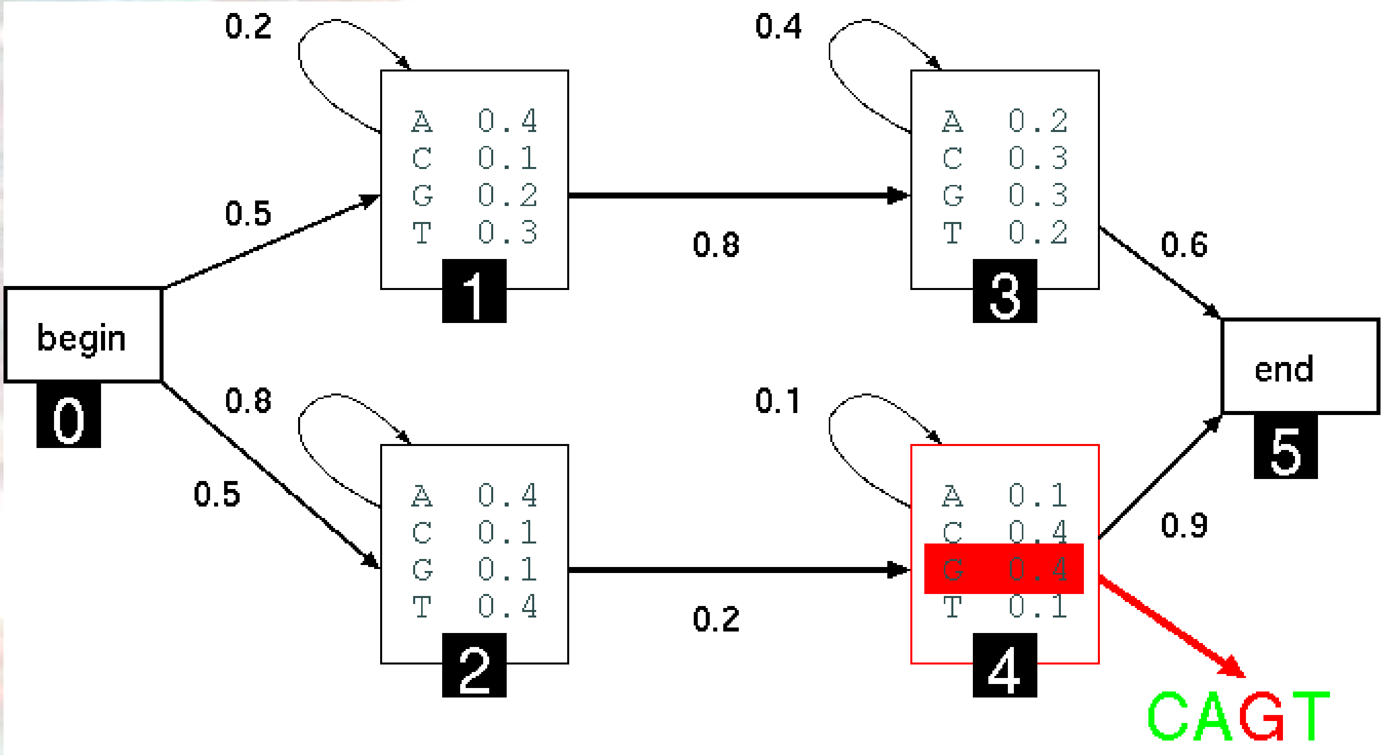
- aka o algoritmo Forward-Backward
- um algoritmo de *Maximização de Expectativas (EM)*:
  - ⋆ EM é uma família de algoritmos para aprender modelos probabilísticos em problemas que envolvem estado escondido
- neste contexto, o estado escondido é o caminho que melhor explica cada sequência de treino

Aprender Parâmetros:
o Algoritmo de Baum-Welch

- ideia do algoritmo
  - ⋆ inicializar parâmetros do modelo
  - ⋆ iterar até convergir
    - ∗ calcular o número *esperado* de vezes que cada transição ou emissão é usada
    - ∗ ajustar os parâmetros para *maximizar* a verosimilhança desses valores esperados

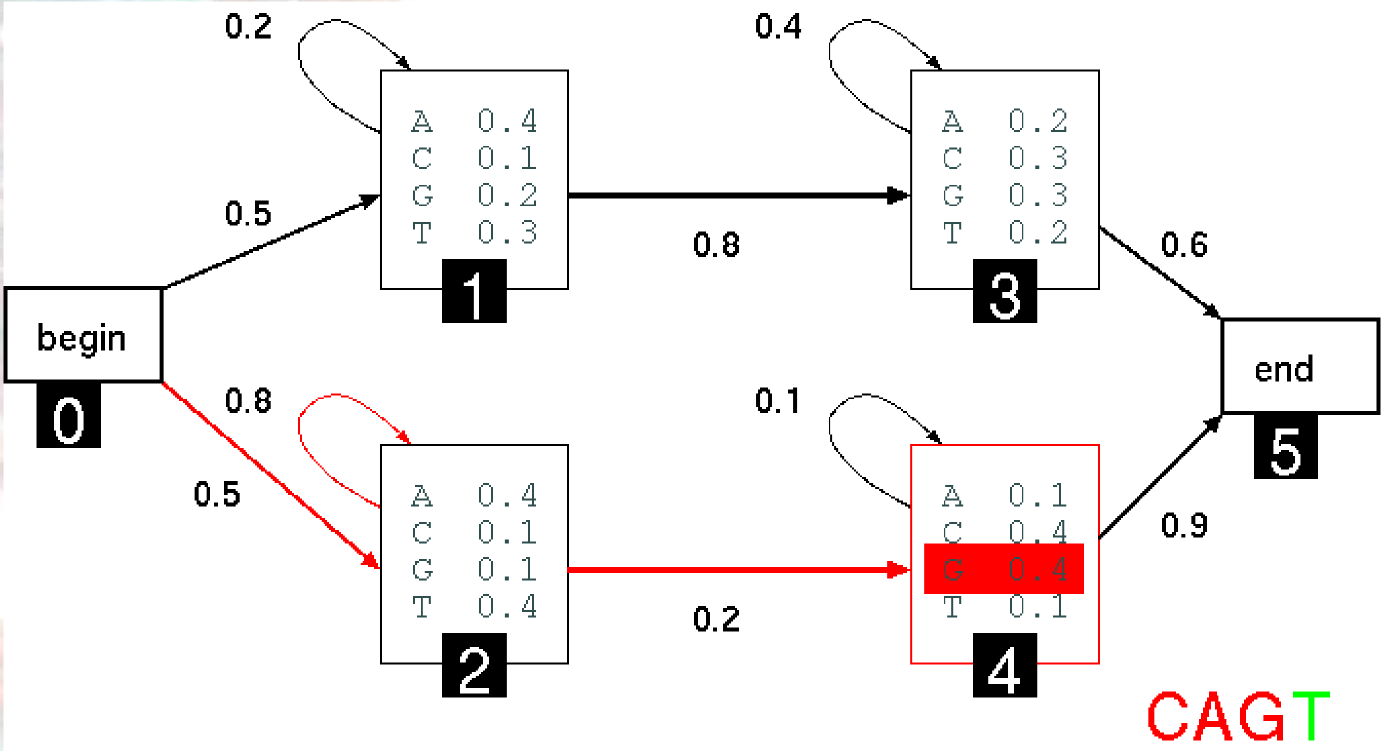
O Passo de Expectativa

- queremos saber a probabilidade de produzir uma sequência x com o símbolo i sendo produzido pelo estado k (para qualquer x, i e k)



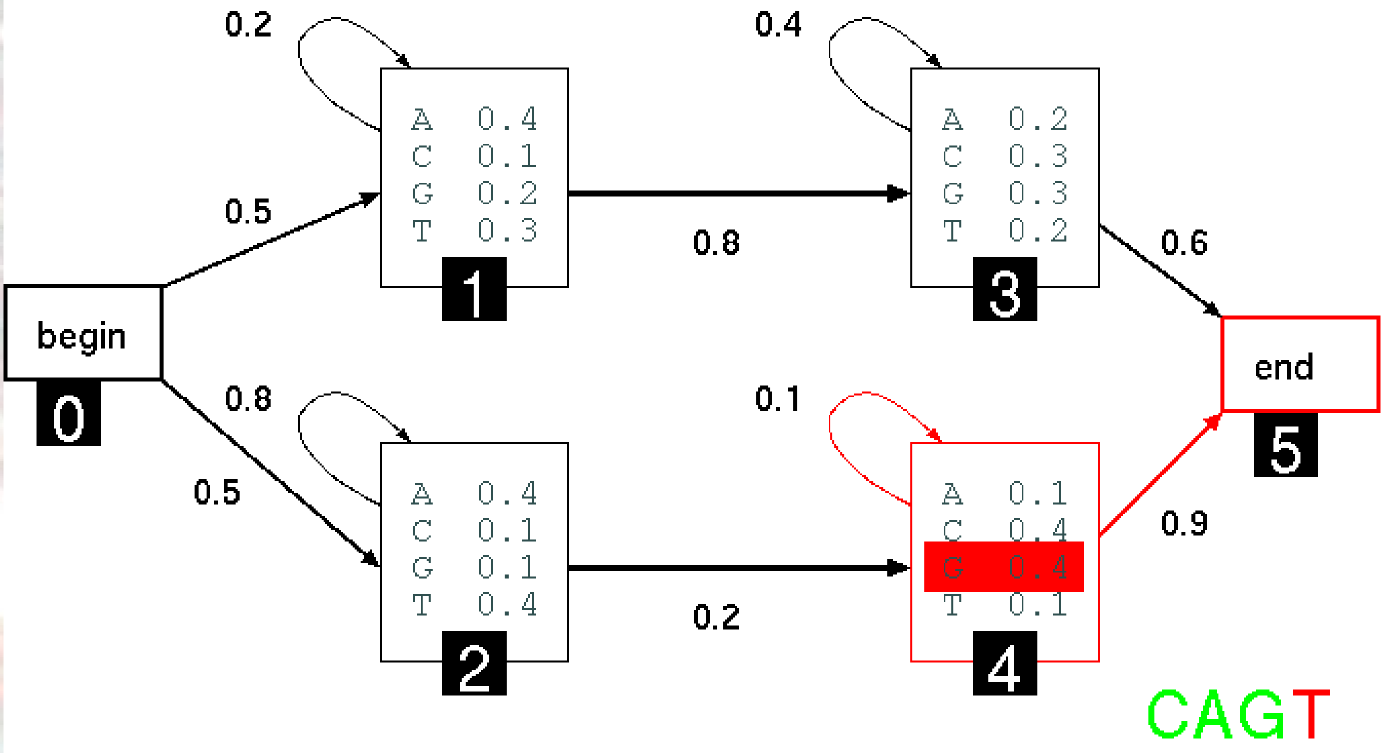
O Passo de Expectativa

- o algoritmo forward dá-nos $f_k(i)$, a probabilidade de estar no estado k tendo observado os primeiro i caracteres de x



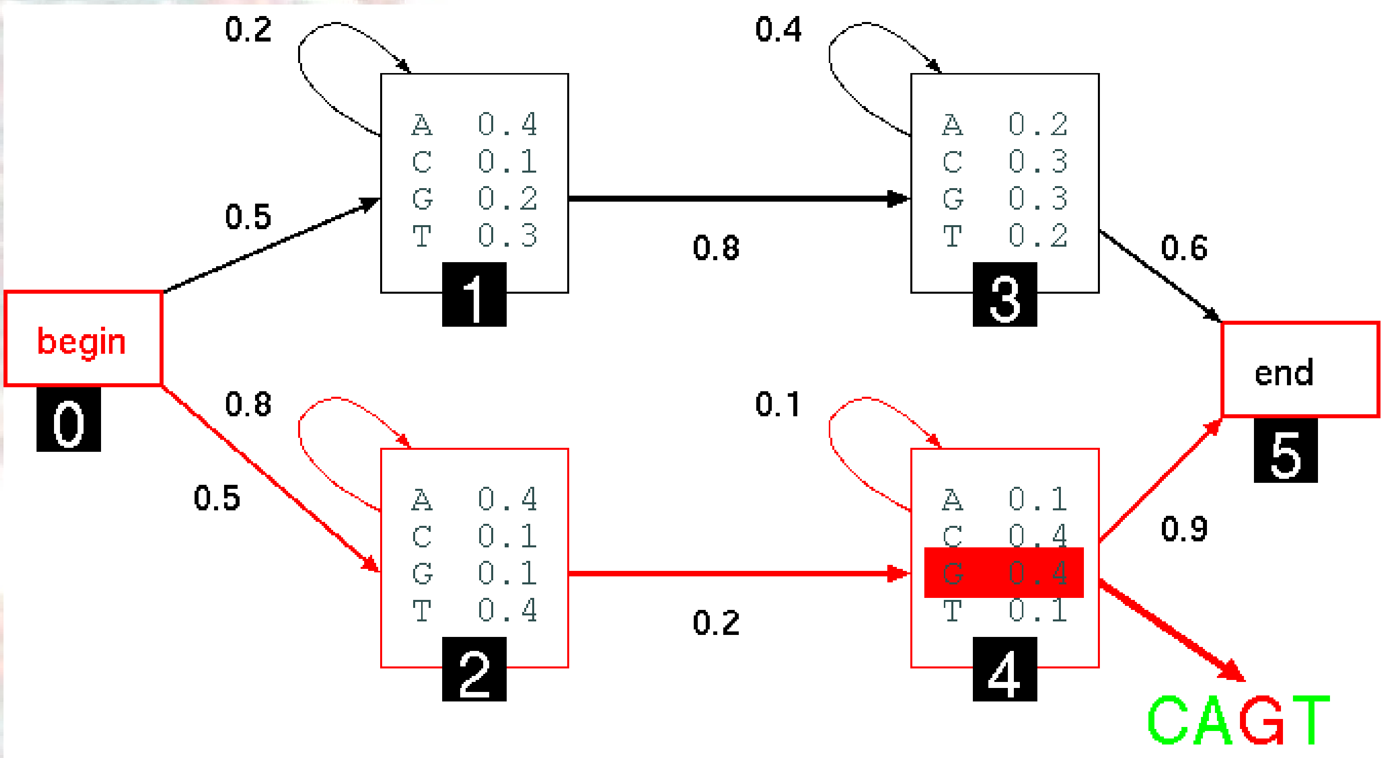
O Passo de Expectativa

- o algoritmo backward dá-nos $b_k(i)$, a probabilidade de observar o resto de x, dado que estamos no estado k depois de i caracteres



O Passo de Expectativa

- Juntando os passos forward e backward, podemos computar a probabilidade de produzir a sequência x com o símbolo na posição i sendo produzido pelo estado q



O Passo de Expectativa

- Primeiro, precisamos de saber a probabilidade do símbolo na posição i ser produzido pelo estado k, dada a sequência x

$$Pr(\pi_i = k|x)$$

- dado isto podemos computar os nossos totais esperados para transições de estados e emissões de caracteres

O Passo de Expectativa

- a probabilidade de produzir x com o símbolo na posição i sendo produzido pelo estado k é:

$$\begin{aligned} Pr(\pi_i = k, x) = \; & Pr(x_1 \ldots x_i, \pi_i = k) \times \\ & Pr(x_{i+1} \ldots x_L | \pi_i = k) \end{aligned}$$

- o primeiro termo é $f_k(i)$, computado pelo algoritmo forward
- o segundo termo é $b_k(i)$, computado pelo algoritmo backward

O Algoritmo Backward

- inicialização

$$b_k(L) = a_{kN}$$

O Algoritmo Backward

- Caso Recursivo:

$$b_k(i) = \sum_l \begin{cases} a_{kl} b_l(i), & \text{se } l \text{ fôr estado silencioso} \\ a_{kl} e_l(x_{i+1}) b_l(i+1), & \text{senão} \end{cases}$$

O Algoritmo Backward

- terminação:

$$Pr(x) = Pr(x_1 \ldots x_L) \sum_l \begin{cases} a_{0l} b_l(0), & \text{se } l \text{ fôr estado silenci} \\ a_{0l} e_l(x_1) b_l(1), & \text{senão} \end{cases}$$

O Passo de Expectativa

- Agora podemos calcular a probabilidade de que o símbolo i tenha sido produzido pelo estado k, dado uma sequência x:

$$
\begin{aligned}
Pr(\pi_i|k,x) &= \frac{Pr(\pi_i|k)}{Pr(x)} \\
&= \frac{f_k(i)b_k(i)}{Pr(x)} \\
&= \frac{f_k(i)b_k(i)}{f_N(L)}
\end{aligned}
$$

O Passo de Expectativa

- A partir daqui podemos calcular o número esperado de vezes que a letra c é emitida pelo estado k.
- De notar que adicionamos o índice j para referir a uma sequência específica no conjunto de treino.

$$n_{k,c} = \sum_{x^j} [\frac{1}{Pr(x^j)} \sum_{\{i|x^j=c\}} f_k^j(i) b_k^j(i)]$$

- ⋆ Soma sobre todas as sequências x^j no conjunto de treino
- ⋆ Soma sobre todas as posições onde c aparece em x^j

O Passo de Expectativa

- e podemos calcular o número esperado de vezes que a transição de k para l é usada:

$$n_{k \to l} = \sum_{x^j} \frac{\sum_i f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)}{Pr(x^j)}$$

- ou se l é um estado silencioso:

$$n_{k \to l} = \sum_{x^j} \frac{\sum_i f_k^j(i) a_{kl} b_l^j(i)}{Pr(x^j)}$$

O Passo de Maximização

- Seja $n_{k,c}$ o número esperado de emissões de c a partir do estado k para o conjunto de treino.
- Estime novos parâmetros de emissão por:

$$e_k(c) = \frac{n_{k,c}}{\sum_{c'} n_{k,c'}}$$

- Exactamente como no caso simples
- Mas habitualmente fazemos algum *amaciamento*, (ie, adicionar pseudo-contagens).

O Passo de Maximização

- Deixe $n_{k\to l}$ ser o número esperado de transições desde o estado k para o estado l para o conjunto de treino
- estime novos parâmetros de transição como:

$$a_{kl} = \frac{n_{k\to l}}{\sum_m n_{k\to m}}$$

O Algoritmo de Baum-Welch

- inicializar os parâmetros do HMM
- itere até convergir:
  - ⋆ inicializar $n_{k,c}.n_{k \to l}$ com pseudo-contagens
  - ⋆ *Passo-E*: para cada sequência de treino $j = 1 \ldots n$
    - $*$ calcule $f_k(i)$ para a sequência j
    - $*$ calcule $b_k(i)$ para a sequência j
    - $*$ adicione a contribuição da sequência j a $n_{k,c}$,$n_{k \to l}$
  - ⋆ *Passo-M*: actualize os parâmetros do HMM usando $n_{k,c}$,$n_{k \to l}$