Hidden Markov Models

Rui Mendes

23 de Maio de 2015

Secção 1

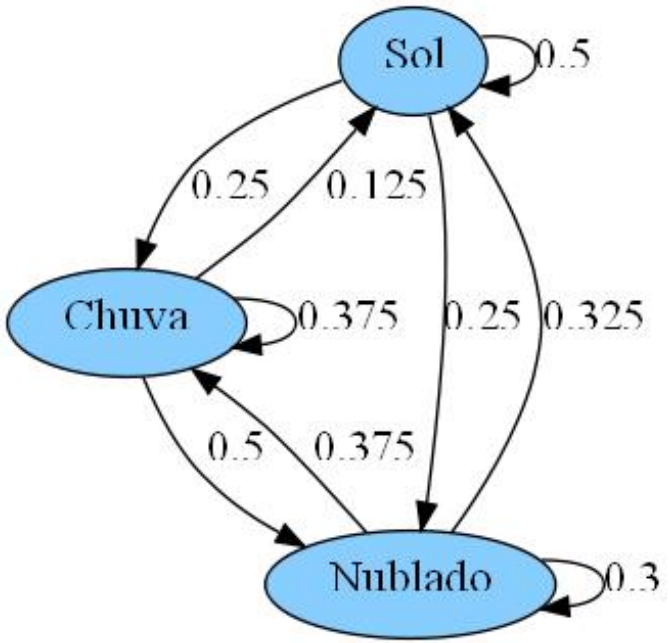
Modelos de Markov

- Possuem n estados, $s_1, s_2, \ldots, s_n$;
- Existem estados especiais: iniciais e finais;
- A cada instante t, o sistema encontra-se num dos estados $q_t \in \{s_1, s_2, \ldots s_n\}$;
- O próximo estado é escolhido de forma determinística seguindo uma transição desde o estado actual até ao próximo estado.

- Possuem n estados, $s_1, s_2, \ldots, s_n$;
- A simulação está dividida em instantes de tempo discretos;
- A cada instante t, o sistema encontra-se num dos estados $q_t \in \{s_1, s_2, \ldots s_n\}$;
- O próximo estado é escolhido de forma aleatória;
- Escolhe-se aleatóriamente uma transição a partir do estado actual;
- Cada transição contém como peso a probabilidade de escolher essa transição.

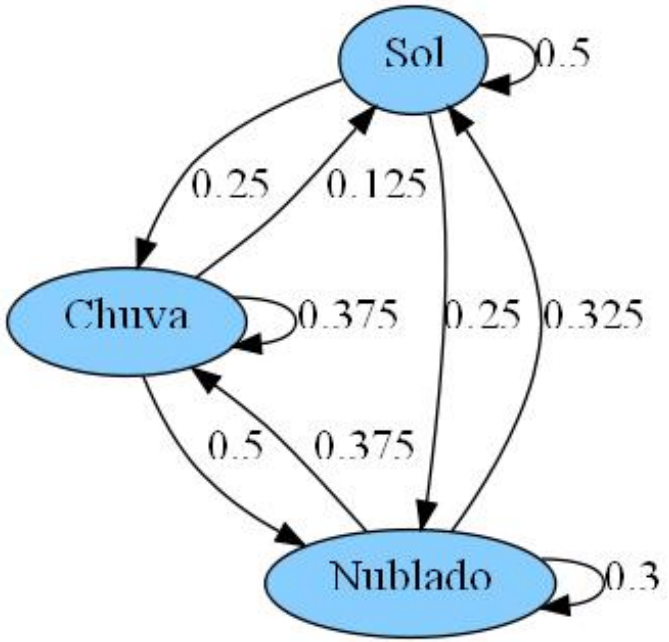
Modelos de Markov: Um Exemplo

- Pequeno modelo sobre o tempo;
- Três estados: tempo de chuva, sol e nublado;
- Cada vértice do grafo representa um dos estados;
- Os ramos representam a transição do estado q_t para q_{t+1};
- O peso do ramo é a probabilidade de escolher essa transição;



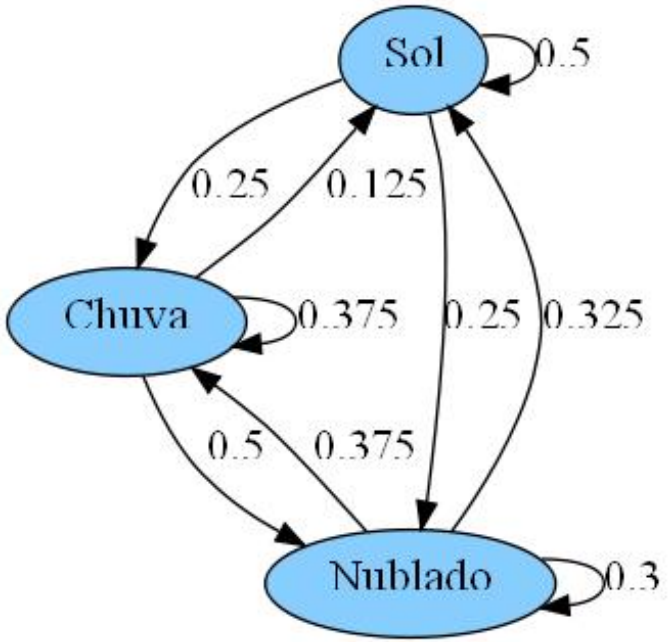
Modelos de Markov: Um Exemplo

- Pequeno modelo sobre o tempo;
- Três estados: tempo de chuva, sol e nublado;
- Cada vértice do grafo representa um dos estados;
- Os ramos representam a transição do estado q_t para q_{t+1};
- O peso do ramo é a probabilidade de escolher essa transição;
- Como determinamos q_1?



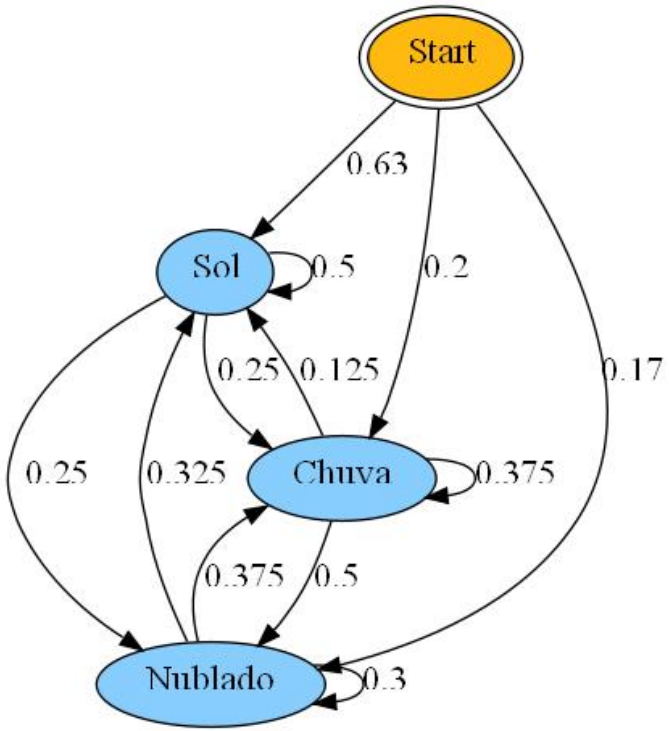
Modelos de Markov: Um Exemplo

- Pequeno modelo sobre o tempo;
- Três estados: tempo de chuva, sol e nublado;
- Cada vértice do grafo representa um dos estados;
- Os ramos representam a transição do estado q_t para q_{t+1};
- O peso do ramo é a probabilidade de escolher essa transição;
- Existe um estado inicial que determina a probabilidade de q_1.



Modelos de Markov: Um Exemplo

- Pequeno modelo sobre o tempo;
- Três estados: tempo de chuva, sol e nublado;
- Cada vértice do grafo representa um dos estados;
- Os ramos representam a transição do estado q_t para q_{t+1};
- O peso do ramo é a probabilidade de escolher essa transição;
- Existe um estado inicial que determina a probabilidade de q_1.

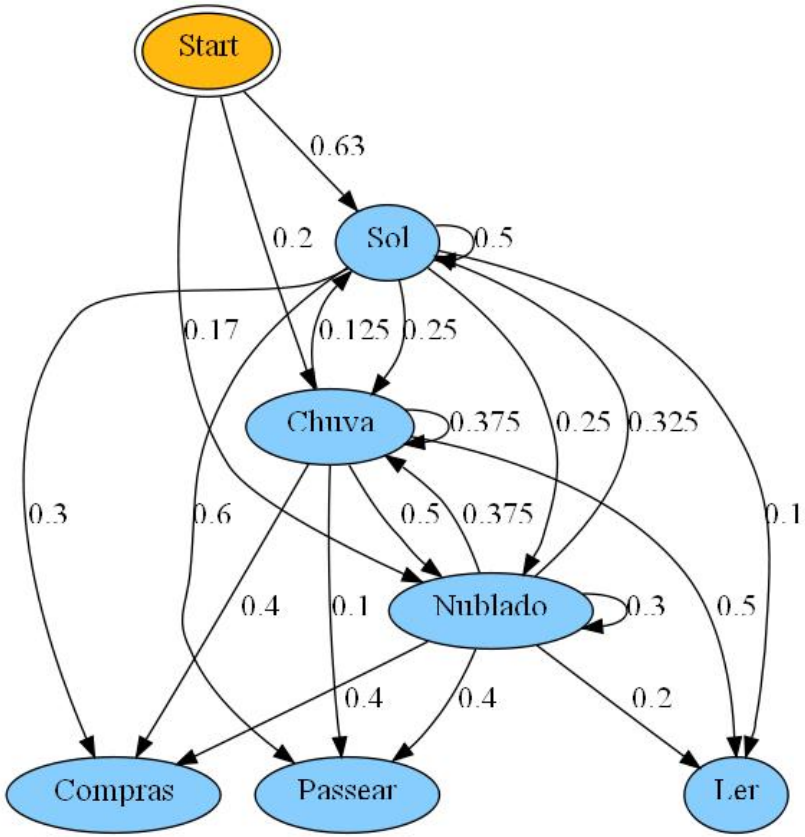


- q_{t+1} está condicionalmente dependente de q_t;
- q_{t+1} é condicionalmente independente de $\{q_{t-1}, q_{t-2}, \ldots, q_1, q_0\}$;
- $p(q_{t+1} = s_j | q_t = s_i, q_{t-1} = s_k, \ldots) = p(q_{t+1} = s_j | q_t = s_i)$;
- Poderiamos colocar os valores das probabilidades de transição numa matriz, i.e., $a_{ij} = p(q_{t+1} = s_j | q_t = s_i)$;
- E os valores das probabilidades dos estados iniciais num vector $\pi_i = p(q_1 = s_i)$.

Um amigo nosso gosta de fazer três coisas: ir às compras, ir passear ou ficar em casa a ler. Nós conhecêmo-lo bem e por isso sabemos que a probabilidade de ele fazer cada uma dessas coisas depende do tempo.
Como podemos modelar isso?

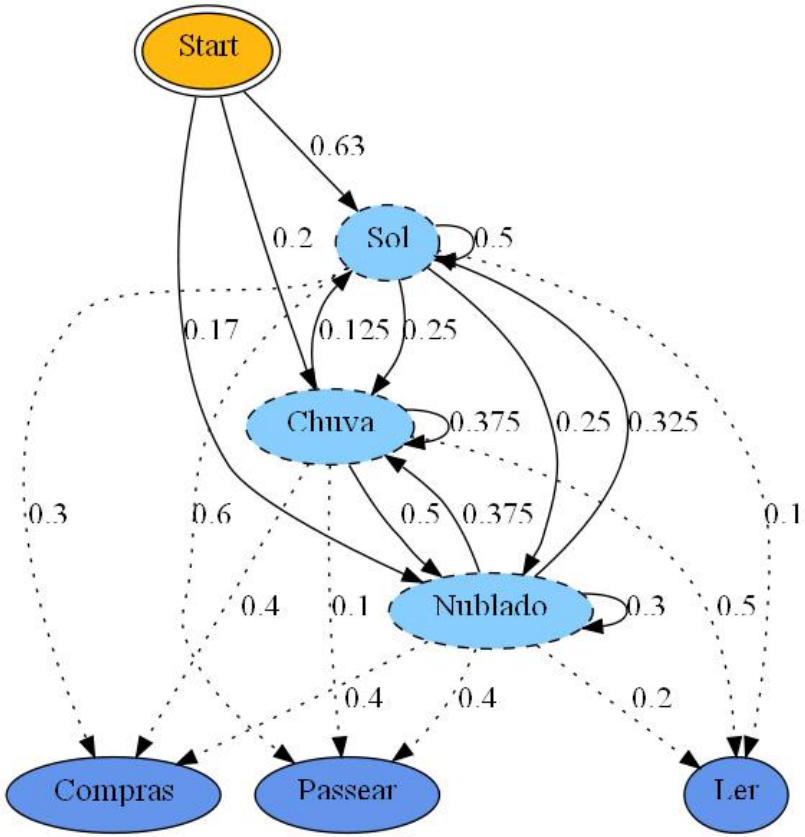
Um amigo nosso gosta de fazer três coisas: ir às compras, ir passear ou ficar em casa a ler. Nós conhecêmo-lo bem e por isso sabemos que a probabilidade de ele fazer cada uma dessas coisas depende do tempo.
Como podemos modelar isso?
Colocando estados para as acções compras, passear e ler.

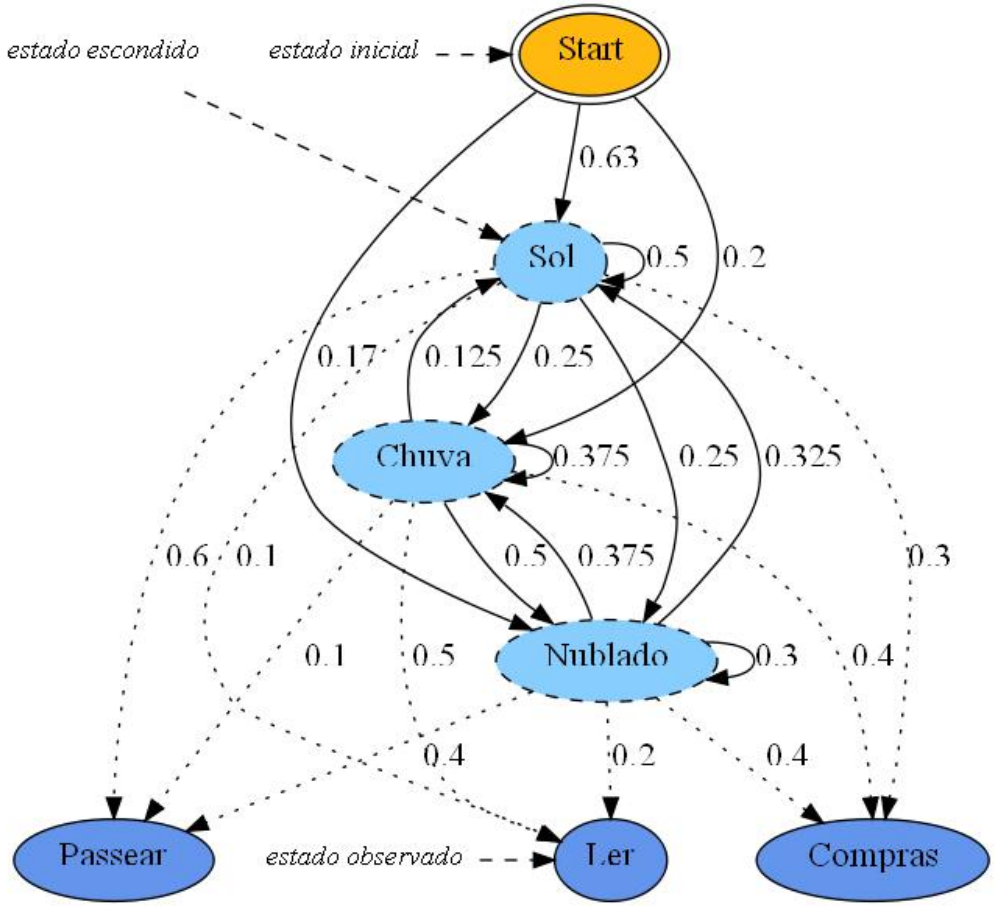
Situação


Start
0.63
0.2
Sol
0.5
0.17
0.125
0.25
Chuva
0.375
0.25
0.325
0.3
0.6
0.5
0.375
0.1
0.4
0.1
Nublado
0.3
0.5
0.4
0.4
0.2
Compras
Passear
Ler

Imaginemos que o nosso amigo tem um blog onde costuma dizer o que faz em cada dia. Contudo, ele nunca coloca informação no blog sobre o tempo. Neste caso, existem estados que nós conseguimos observar e estados que não conseguimos observar.

Situação


Start
0.63
0.2
Sol
0.5
0.17
0.125
0.25
Chuva
0.375
0.25
0.325
0.3
0.6
0.5
0.375
0.1
0.4
0.1
Nublado
0.3
0.5
0.4
0.4
0.2
Compras
Passear
Ler


estado escondido
estado inicial
Start
0.63
Sol
0.5
0.2
0.17
0.125
0.25
Chuva
0.375
0.25
0.325
0.6
0.1
0.5
0.375
0.3
0.1
0.5
Nublado
0.3
0.4
0.4
0.2
0.4
Passear
estado observado
Ler
Compras

Hidden Markov Model

- Tem n estados escondidos, $s_1, s_2, \ldots, s_n$;
- Tem m estados observados, $e_1, e_2, \ldots, e_m$;
- A simulação está dividida em instantes de tempo discretos;
- A cada instante t, o sistema encontra-se num dos estados escondidos $q_t \in \{s_1, s_2, \ldots s_n\}$ que emite um estado observado $o_t \in \{e_1, e_2, \ldots, e_m\}$;
- O estado inicial é determinado por $\pi_i = p(q_1 = s_i)$
- A escolha do próximo estado depende do estado actual segundo $a_{ij} = p(q_{t+1} = s_j | q_t = s_i)$;
- O estado observado o_t depende do estado actual q_t segundo $b_{ij} = p(o_t = e_j | q_t = s_i)$;

Um modelo caracteriza-se por

$$< s, e, \pi, a, b >$$

em que:

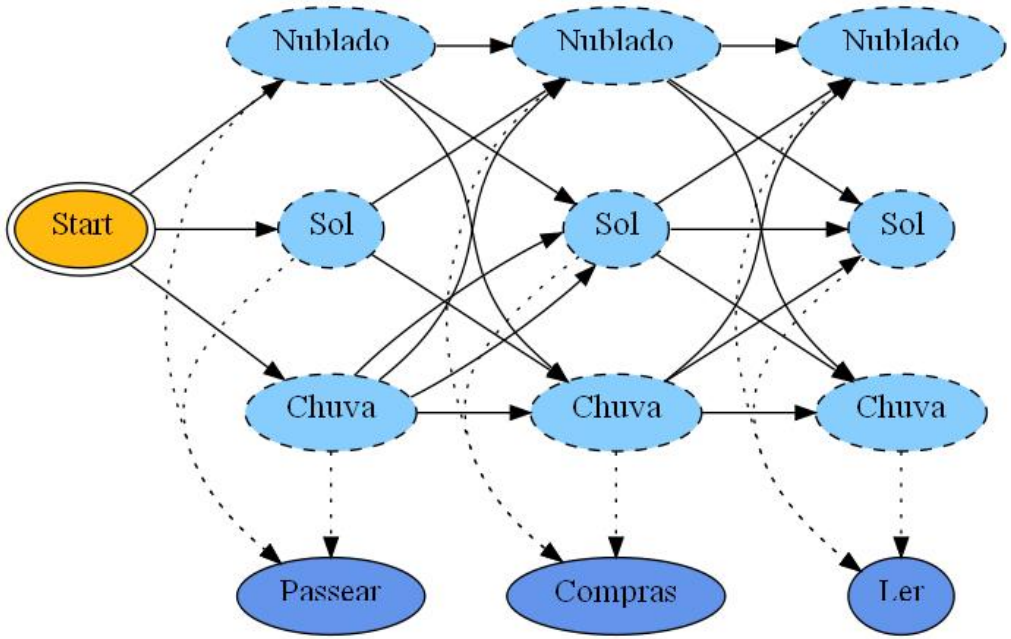
$$\begin{aligned} s &= \{s_1, \ldots, s_n\} \\ e &= \{e_1, \ldots, e_m\} \\ \pi_i &= p(q_1 = s_i) \\ a_{ij} &= p(q_{t+1} = s_j | q_t = s_i) \\ b_{ij} &= p(o_t = e_j | q_t = s_i) \end{aligned}$$

Três Questões

Existem três questões pertinentes que se colocam quando falamos de HMMs.

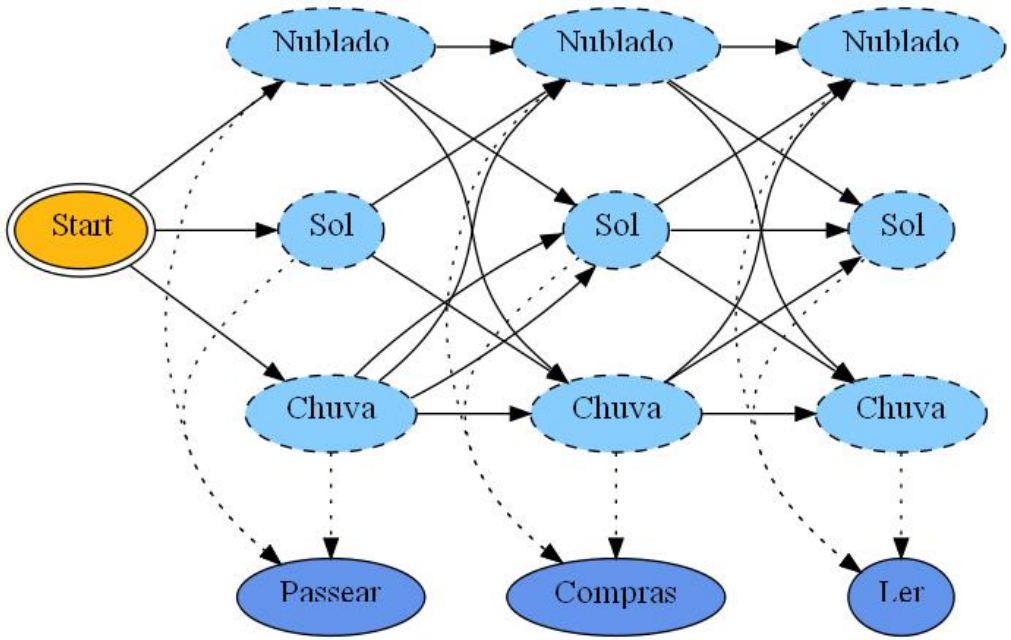
1. Qual é a probabilidade de um modelo gerar uma determinada sequência de observações?
2. Dado um modelo e uma sequência de observações, qual é a sequência mais provável de estados que as emitiu?
3. Dado um conjunto (ou conjuntos) de observações, qual é o modelo mais adequado?

Como calcular a probabilidade de ocorrer a sequência de observações <Passear, Compras, Ler>?



Como calcular a probabilidade de ocorrer a sequência de observações <Passear, Compras, Ler>?

$$\sum_{s_1,s_2,s_3 \in S} p(\text{Passear}|s_1)p(\text{Compras}|s_2)p(\text{Ler}|s_3)p(s1,s2,s3)$$



Como calcular a probabilidade de ocorrer a sequência de observações <Passear, Compras, Ler>?

$$\sum_{s_1,s_2,s_3 \in S} p(\text{Passear}|s_1)p(\text{Compras}|s_2)p(\text{Ler}|s_3)p(s1, s2, s3)$$

Quantas combinações são?

Probabilidade de ocorrer um conjunto de observações

Como calcular a probabilidade de ocorrer a sequência de observações <Passear, Compras, Ler>?

$$\sum_{s_1,s_2,s_3 \in S} p(\text{Passear}|s_1)p(\text{Compras}|s_2)p(\text{Ler}|s_3)p(s1,s2,s3)$$

Quantas combinações são? 3^3

Probabilidade de ocorrer um conjunto de observações

Como calcular a probabilidade de ocorrer a sequência de observações <Passear, Compras, Ler>?

$$\sum_{s_1,s_2,s_3 \in S} p(\text{Passear}|s_1)p(\text{Compras}|s_2)p(\text{Ler}|s_3)p(s1,s2,s3)$$

Quantas combinações são?

$$\text{estados}^{\text{observações}}$$

Será possível fazer isto melhor?

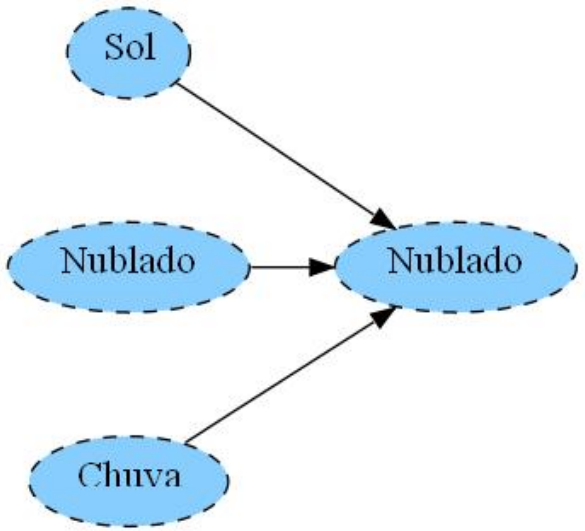
Será possível fazer isto melhor?
Utilizando programação dinâmica.

Será possível fazer isto melhor?
Utilizando programação dinâmica.
Como?

Probabilidade atingir um dado estado escondido

É possível calcular a probabilidade de chegar a um dado estado escondido num dado instante de tempo. Podemos definir essa probabilidade por indução:

$$
\begin{aligned}
p(q_1 = s_i) &= \pi_i \\
p(q_t = s_j) &= \sum_{s_i \in s} a_{ij} p(q_{t-1} = s_i)
\end{aligned}
$$



Probabilidade ocorrer uma dada observação

Para calcular a probabilidade de ocorrência de uma dada observação e_j se se estiver num dado estado escondido s_i só se necessita de multiplicar a probabilidade de atingir esse estado escondido s_i pela probabilidade de observar e_j estando em s_i.

$$p(o_t = e_j) \quad = \quad p(q_t = s_i) b_{ij}$$

Assim, poderemos definir esta probabilidade por indução baseada nas equações apresentadas no slide anterior.

$$\begin{aligned} p(q_1 = s_i | o_1 = e_j) &= \pi_i b_{ij} \\ p(q_t = s_j | o_t = e_k) &= b_{jk} \sum_{s_i \in s} a_{ij} p(q_{t-1} = s_i | o_{t-1}) \end{aligned}$$

Uma pergunta que se faz muitas vezes é qual seria o caminho mais provável (conjunto de estados escondidos mais provável) para uma dada sequência de observações. O procedimento é muito semelhante ao anterior mas estamos interessados simplesmente na transição mais provável. Assim, podemos definir esta relação por indução da seguinte forma:

$$\begin{aligned} p(q_1 = s_i | o_1 = e_j) &= \pi_i b_{ij} \\ p(q_t = s_j | o_t = e_k) &= b_{jk} \max_{s_i \in s} a_{ij} p(q_{t-1} = s_i | o_{t-1}) \end{aligned}$$

Repare que ao mesmo tempo se armazena o caminho percorrido (tal como no algoritmo de alinhamento de sequências) noutra matriz para se poder reconstruir o caminho.

Treinar um HMM (supervisionado)

Caso conheçamos tudo, os estados observados e os estados escondidos então usa-se o algoritmo de Maximum Likelihood Estimate.
Neste caso o que pretendemos é maximizar a probabilidade da ocorrência dos estados observados e escondidos.
O que se costuma fazer é começar pela frequência observada dos estados.

Treinar um HMM (supervisionado)

Caso 1

Escondido	Sol	Chuva	Chuva	Sol	Sol
Observado	Passear	Ler	Ler	Passear	Passear

Caso 2

Escondido	Chuva	Chuva	Chuva	Chuva	Sol
Observado	Ler	Ler	Ler	Ler	Passear

$$p(s_1 = Sol|s_0) = ?$$

$$p(o_i = Passear|s_i = Sol) = ?$$

$$p(s_{i+1} = Nublado|s_i = Sol) = ?$$

Treinar um HMM (supervisionado)

Caso 1

Escondido	Sol	Chuva	Chuva	Sol	Sol
Observado	Passear	Ler	Ler	Passear	Passear

Caso 2

Escondido	Chuva	Chuva	Chuva	Chuva	Sol
Observado	Ler	Ler	Ler	Ler	Passear

$$p(s_1 = Sol|s_0) = ?$$
$$p(o_i = Passear|s_i = Sol) = ?$$
$$p(s_{i+1} = Nublado|s_i = Sol) = ?$$

Treinar um HMM (supervisionado)

Caso 1

Escondido	Sol	Chuva	Chuva	Sol	Sol
Observado	Passear	Ler	Ler	Passear	Passear

Caso 2

Escondido	Chuva	Chuva	Chuva	Chuva	Sol
Observado	Ler	Ler	Ler	Ler	Passear

$$p(s_1 = Sol|s_0) = ?$$
$$p(o_i = Passear|s_i = Sol) = ?$$
$$p(s_{i+1} = Nublado|s_i = Sol) = ?$$

Treinar um HMM (supervisionado)

O que se deve fazer quando um estado não ocorre?

Usa-se uma pseudocontagem. Adicionamos mais uma contagem fantasma a cada estado.

Treinar um HMM (supervisionado)

O que se deve fazer quando um estado não ocorre?
Usa-se uma pseudocontagem. Adicionamos mais uma contagem fantasma a cada estado.

Treinar um HMM (supervisionado)

Caso 1

Escondido	Sol	Chuva	Chuva	Sol	Sol
Observado	Passear	Ler	Ler	Passear	Passear

Caso 2

Escondido	Chuva	Chuva	Chuva	Chuva	Sol
Observado	Ler	Ler	Ler	Ler	Passear

Sem pseudocontagem

$$p(s_1 = Sol|s_0) = \frac{1}{2}$$

$$p(s_1 = Chuva|s_0) = \frac{1}{2}$$

$$p(s_1 = Nublado|s_0) = \frac{0}{2}$$

Com pseudocontagem

$$p(s_1 = Sol|s_0) = \frac{1+1}{2+3} = \frac{2}{5}$$

$$p(s_1 = Chuva|s_0) = \frac{1+1}{2+3} = \frac{2}{5}$$

$$p(s_1 = Nublado|s_0) = \frac{0+1}{2+3} = \frac{1}{5}$$

Treinar um HMM (supervisionado)

Caso 1

Escondido	Sol	Chuva	Chuva	Sol	Sol
Observado	Passear	Ler	Ler	Passear	Passear

Caso 2

Escondido	Chuva	Chuva	Chuva	Chuva	Sol
Observado	Ler	Ler	Ler	Ler	Passear

Sem pseudocontagem

$$p(Sol|Chuva) = \frac{2}{6}$$

$$p(Chuva|Chuva) = \frac{4}{6}$$

$$p(Nublado|Chuva) = \frac{0}{6}$$

Com pseudocontagem

$$p(Sol|Chuva) = \frac{2+1}{6+3}$$

$$p(Chuva|Chuva) = \frac{4+1}{6+3}$$

$$p(Nublado|Chuva) = \frac{0+1}{6+3}$$

Treinar um HMM (não supervisionado)

Dada uma sequência de observações, qual é o modelo mais adequado? Um dos algoritmos mais conhecidos é o algoritmo de Baum-Welch.
O algoritmo de Baum-Welch é um generalized expectation-maximization (GEM) algorithm. Ele calcula estimativas de máxima semelhança e posteriormente estima os valores dos parâmetros (probabilidades de transição e emissão) de um HMM com base nas emissões que lhe foram dadas.
O algoritmo tem dois passos:

1. Calcular a forward probability e backward probability de cada estado do HMM;
2. Com base nisso, determinar a frequência do par transição/emissão e dividi-la pela frequência de toda a sequência. Este valor é o novo valor da transição.

Secção 2

POS Tagging

Aqueça	água	numa	panela
V	N	PRP	N

estados observados palavras

estados escondidos POS tags

Part of Speech Tagging

Aqueça	água	numa	panela
V	N	PRP	N

estados observados palavras

estados escondidos POS tags

Part of Speech Tagging

Aqueça	água	numa	panela
V	N	PRP	N

estados observados palavras

estados escondidos POS tags

Modelo baseado em bigramas

$$
\begin{aligned}
p(V\ N\ PRP\ N \quad | \quad & Aqueca\ agua\ numa\ panela) = \\
& p(V|\#) \cdot p(N|V) \\
& \cdot p(PRP|N) \cdot p(N|PRP) \\
& \cdot p(Aqueca|V) \cdot p(agua|N) \\
& \cdot p(numa|PRP) \cdot p(panela|N)
\end{aligned}
$$

Part of Speech Tagging

Aqueça	água	numa	panela
V	N	PRP	N

estados observados palavras

estados escondidos POS tags

Modelo baseado em trigramas

$$
\begin{aligned}
p(V\ N\ PRP\ N \quad | \quad & Aqueca\ agua\ numa\ panela) = \\
& p(V|\#\ \#) \cdot p(N|\#\ V) \\
& \cdot p(PRP|V\ N) \cdot p(N|N\ PRP) \\
& \cdot p(Aqueca|V) \cdot p(agua|N) \\
& \cdot p(numa|PRP) \cdot p(panela|N)
\end{aligned}
$$

- O que fazer quando nunca vimos uma dada sequência de tags?
- Temos que alisar a probabilidade das transições.
- O que fazer quando não conhecemos uma dada palavra?
- Temos que alisar a probabilidade das emissões.

Smoothing

- O que fazer quando nunca vimos uma dada sequência de tags?
- Temos que alisar a probabilidade das transições.
- O que fazer quando não conhecemos uma dada palavra?
- Temos que alisar a probabilidade das emissões.

Smoothing

- O que fazer quando nunca vimos uma dada sequência de tags?
- Temos que alisar a probabilidade das transições.
- O que fazer quando não conhecemos uma dada palavra?
- Temos que alisar a probabilidade das emissões.

Smoothing

- O que fazer quando nunca vimos uma dada sequência de tags?
- Temos que alisar a probabilidade das transições.
- O que fazer quando não conhecemos uma dada palavra?
- Temos que alisar a probabilidade das emissões.

Probabilidade de transições para bigramas

$$\begin{aligned} p(s_2|s_1) = &\quad \lambda_1 \cdot p(s_2) + \lambda_2 \cdot p(s_2|s_1) \\ \lambda_1 + \lambda_2 = &\quad 1 \end{aligned}$$

Probabilidade de transições para trigramas

$$\begin{aligned} p(s_3|s_1 s_2) = &\quad \lambda_1 \cdot p(s_3) + \lambda_2 \cdot p(s_3|s_2) + \lambda_3 \cdot p(s_3|s_1 s_2) \\ \lambda_1 + \lambda_2 + \lambda_3 = &\quad 1 \end{aligned}$$

Probabilidade de emissões

- Calcular a probabilidade de emissão de uma palavra desconhecida p_d dada uma tag t $P(p_d|t)$ parte o conjunto de treino em c_1 e c_2
- Calcula o vocabulário V de c_1
- Substitui todas as palavras $p \in c_2$ tais que $p \notin c_1$ pela palavra desconhecida p_d
- Calcula $P(p_d|t)$ usando esta transformação sobre c_2
- Reajustar as probabilidades para que somem 1

Outras Alternativas para Smoothing

Probabilidade de emissões

- Tags mais prováveis para palavras desconhecidas: NN, V
- Tags menos prováveis para palavras desconhecidas: DET, INTERJ
- Usar a distribuição de palavras só encontradas uma só vez

Probabilidade de emissões

- Usar informação sobre capitalização, prefixos, sufixos, etc
- Probabilidades de palavras desconhecidas são calculadas usando n-gramas baseados em letras usando as últimas m letras l_i de uma palavras de L letras:

$$p(p|l_{L-m+1}, \ldots, l_L)$$

- A ideia é que os sufixos de palavras desconhecidas dão uma boa indicação da POS de uma palavra
- Qual é o tamanho de m? Deve ser baseado no maior sufixo encontrado nos dados mas nunca maior do que 10
- Estas probabilidades também podem ser alisadas por interpolação

Algoritmo de Viterbi

Dados

- Sequência $x_1, \ldots, x_n$ a etiquetar
- $q(s|u, v)$ probabilidade de aparecer a tag s sabendo que as duas tags anteriores foram u e v
- $e(x|s)$ probabilidade de emitir a palavra x estando na tag s
- T conjunto de tags possíveis

Algoritmo de Viterbi

Inicialização

$\pi(0, *, *) = 1$ e $\pi(0, u, v) = 0$ se u ou v não forem $*$

Algoritmo

Para $k = 1 \ldots n$

Para $u, v \in T$

$$\pi(k, u, v) = \max_{w \in T} \pi(k-1, w, u) \times q(v|w, u) \times e(x_k|v)$$

$$t(k, u, v) = \arg\max_{w \in T} \pi(k-1, w, u) \times q(v|w, u) \times e(x_k|v)$$

$$(t_{n-1}, t_n) = \arg\max_{(u,v) \in T} \pi(n, u, v) \times q(STOP|u, v)$$

$$t_k = t(k+2, t_{k+1}, t_{k+2}) \;\; \forall k \in \{n-2, \ldots, 1\}$$

Devolver

$t_1, \ldots, t_n$

Secção 3

Profile Hidden Markov Models

Imagine que lhe dão um alinhamento múltiplo de sequências.

```
LEVK
LDIR
LEIK
LDVE
```

Como poderia definir um modelo baseado em HMMs para, dada uma sequência, determinar se esta pertence ou não à família?

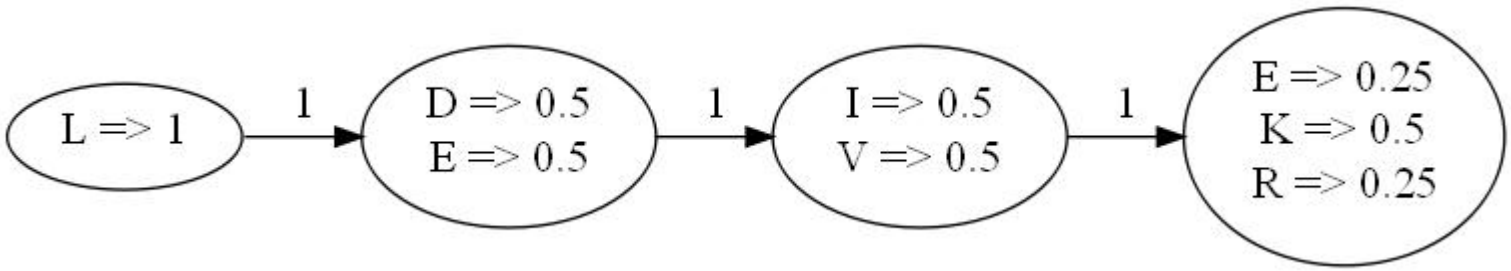
Imagine que lhe dão um alinhamento múltiplo de sequências.

```
LEVK
LDIR
LEIK
LDVE
```

Como poderia definir um modelo baseado em HMMs para, dada uma sequência, determinar se esta pertence ou não à família? Uma forma possível seria considerar um HMM em que cada estado escondido correspondesse a um caracter das sequências. Neste caso, teriamos 4 estados. As observações seriam os caracteres possíveis para cada estado.

Imagine que lhe dão um alinhamento múltiplo de sequências.

```
LEVK
LDIR
LEIK
LDVE
```



Calcular o score

Para calcular o score de uma sequência, é necessário multiplicar as probabilidades da sua ocorrência. Por exemplo, a probabilidade da sequência `LDVR` seria:

$$1 \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{16}$$

Como se pode ver, se a cadeia for grande, as probabilidades aproximam-se rapidamente de zero. Assim, é normal utilizar o logaritmo das probabilidades. Assim, a sequência em causa daria:

$$0 - 0.69315 - 0.69315 - 1.38629 = -2.77259$$

Calcular o score

No caso do modelo poder gerar cadeias de tamanho diferente (veja como mais à frente) costuma-se calcular a probabilidade da sequência relativa à probabilidade desta ocorrer por acaso:

$$\log \frac{P(S)}{\frac{1}{20}^L} = \log P(S) - L \log \frac{1}{20}$$

em que $P(S)$ é a probabilidade da sequência e L é o tamanho desta; usamos $\frac{1}{20}$ porque existem 20 aminoácidos.
A maneira mais fácil de implementar isto consiste em substituir cada probabilidade de emissão p por $\log p - \log \frac{1}{20}$ e somar as probabilidades ao invés de multiplicá-las.

Tratar de espaçamentos

```
VGA--HAGEY
V----NVDEV
VEA--DVAGH
VKG------D
VYS--TYETS
FNA--NIPKH
IAGADNGAGY
```

Como resolver o problema neste caso?

Repare que neste caso, temos mais dois tipos de situações:

1. Posições em que só algumas das sequências tem espaçamentos (e.g., a 2^a, 3^a posição e da 6^a à 9^a);
2. Posições em que a maior parte das sequências tem espaçamentos (e.g., 4^a e 5^a posições).

Tratar de espaçamentos

```
VGA--HAGEY
V----NVDEV
VEA--DVAGH
VKG------D
VYS--TYETS
FNA--NIPKH
IAGADNGAGY
```

Como resolver o problema neste caso?
Repare que neste caso, temos mais dois tipos de situações:

1. Posições em que só algumas das sequências tem espaçamentos (e.g., a 2^a, 3^a posição e da 6^a à 9^a);
2. Posições em que a maior parte das sequências tem espaçamentos (e.g., 4^a e 5^a posições).

Tratar de espaçamentos

```
VGA--HAGEY
V----NVDEV
VEA--DVAGH
VKG------D
VYS--TYETS
FNA--NIPKH
IAGADNGAGY
```

Como resolver o problema neste caso?
Repare que neste caso, temos mais dois tipos de situações:

1. Posições em que só algumas das sequências tem espaçamentos (e.g., a 2^a, 3^a posição e da 6^a à 9^a);
2. Posições em que a maior parte das sequências tem espaçamentos (e.g., 4^a e 5^a posições).

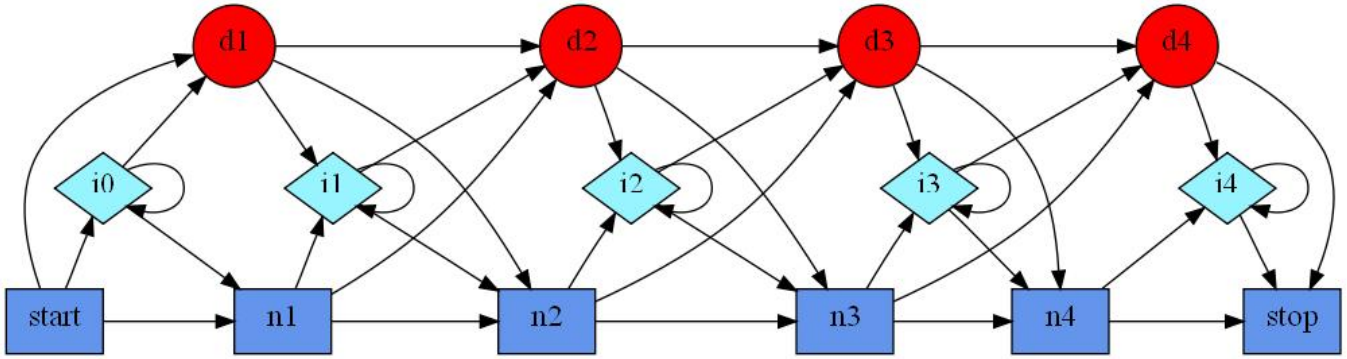
Tratar de espaçamentos

```
VGA--HAGEY
V----NVDEV
VEA--DVAGH
VKG------D
VYS--TYETS
FNA--NIPKH
IAGADNGAGY
```

Como resolver o problema neste caso?
Repare que neste caso, temos mais dois tipos de situações:

1. Posições em que só algumas das sequências tem espaçamentos (e.g., a 2^a, 3^a posição e da 6^a à 9^a);
2. Posições em que a maior parte das sequências tem espaçamentos (e.g., 4^a e 5^a posições).

Profile Hidden Markov Models

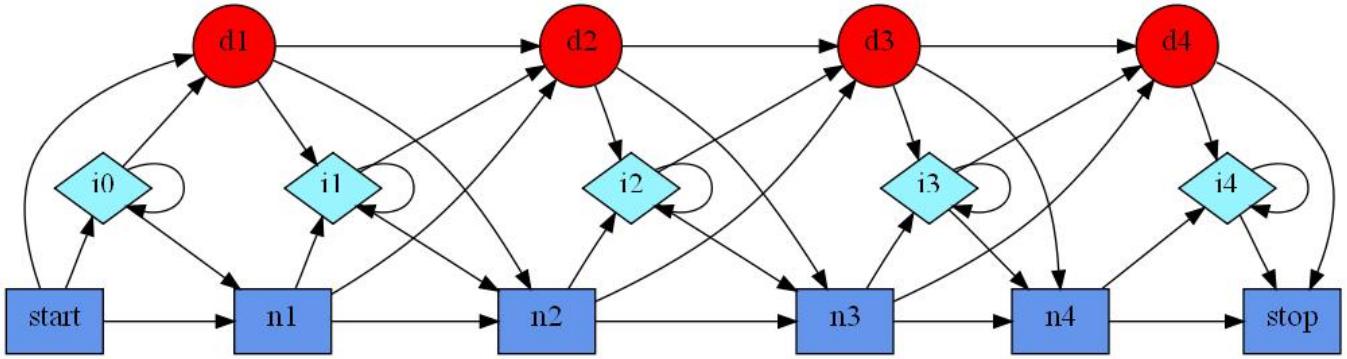


rectângulos Estados utilizados normalmente quando pelo menos 50% das sequências não são espaçamentos;

losangos Estados de inserção, utilizados nos casos em que a maioria das sequências tem espaçamentos;

círculos Estados de remoção, utilizados nos casos em que as sequências tenham espaçamentos; não emitem observações.

Profile Hidden Markov Models

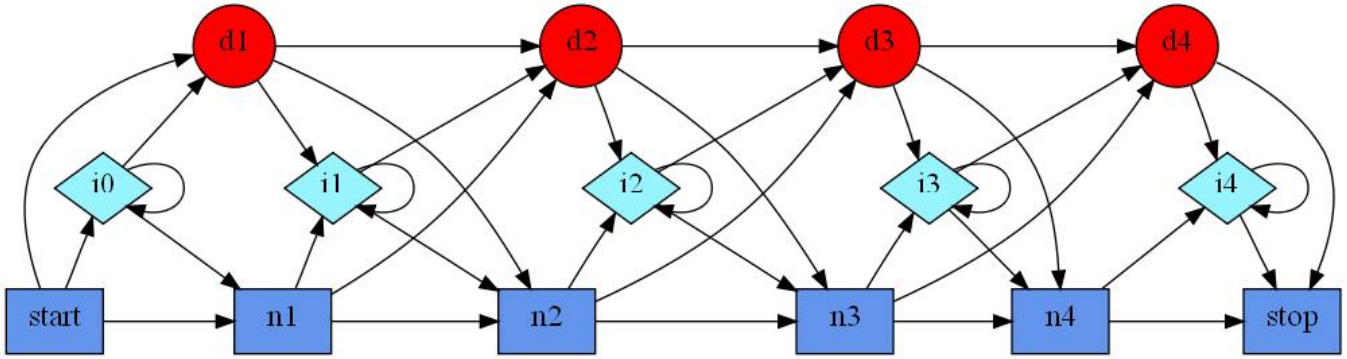


rectângulos Estados utilizados normalmente quando pelo menos 50% das sequências não são espaçamentos;

losangos Estados de inserção, utilizados nos casos em que a maioria das sequências tem espaçamentos;

círculos Estados de remoção, utilizados nos casos em que as sequências tenham espaçamentos; não emitem observações.

Profile Hidden Markov Models

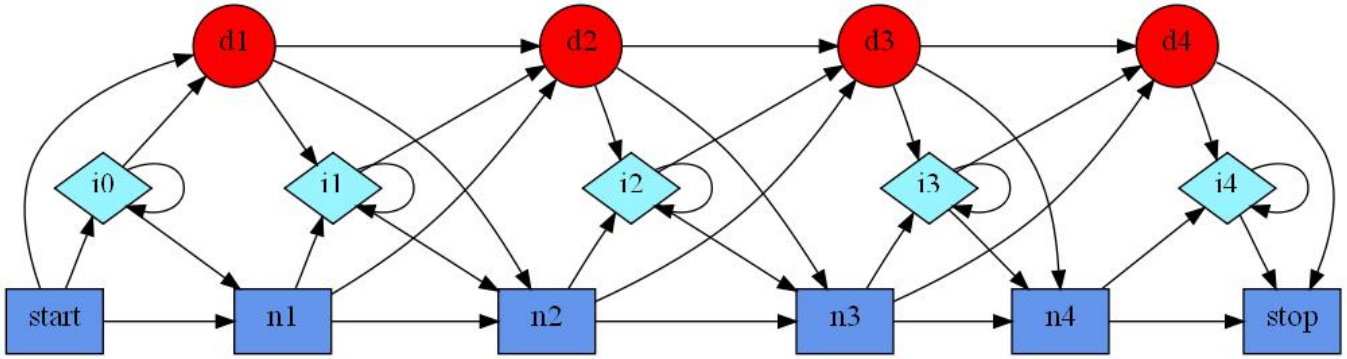


rectângulos Estados utilizados normalmente quando pelo menos 50% das sequências não são espaçamentos;

losangos Estados de inserção, utilizados nos casos em que a maioria das sequências tem espaçamentos;

círculos Estados de remoção, utilizados nos casos em que as sequências tenham espaçamentos; não emitem observações.

Profile Hidden Markov Models



rectângulos Estados utilizados normalmente quando pelo menos 50% das sequências não são espaçamentos;

losangos Estados de inserção, utilizados nos casos em que a maioria das sequências tem espaçamentos;

círculos Estados de remoção, utilizados nos casos em que as sequências tenham espaçamentos; não emitem observações.

Para prevenir o sobre-ajustamento faz sentido que se permita que um modelo aceite aminoácidos que não estão em nenhuma das sequências dadas. Para isso podemos utilizar modelos mais ou menos inteligentes. O mais simples consiste em assumir que todos os aminoácidos tem uma contagem mínima de 1 que é somada aos valores existentes,

Assim, se numa dada posição aparecer 5 vezes o V, e uma vez o F e o I, as probabilidades seriam:

$$\begin{aligned} \text{V} &\quad \frac{5+1}{7+20} = \frac{6}{27} \\ \text{F e I} &\quad \frac{1+1}{7+20} = \frac{2}{17} \\ \text{outros} &\quad \frac{0+1}{7+27} = \frac{1}{27} \end{aligned}$$

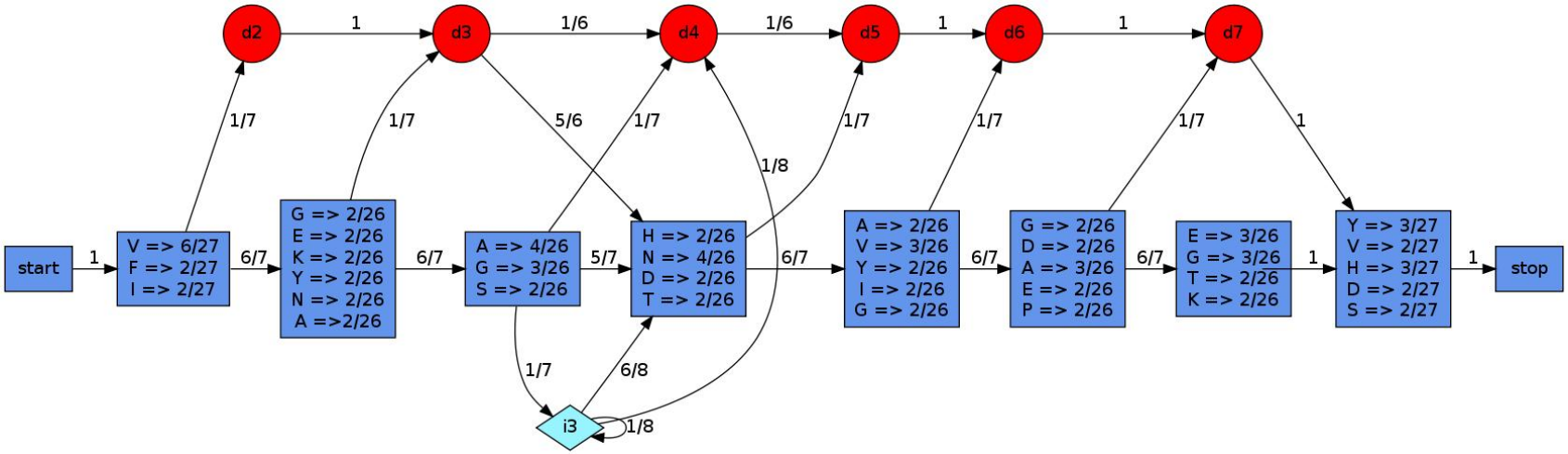
Podemos utilizar modelos mais inteligentes como utilizar frequências de aminoácidos proporcionais às frequências observadas ou então dependente do contexto. Por exemplo, se uma coluna contém uma predominância de aminoácidos hidrofóbicos, esperar-se-ia que a frequência para aminoácidos hidrofóbicos fosse superior à frequência de aminoácidos hidrofílicos.

Exemplo

```
VGA--HAGEY
V----NVDEV
VEA--DVAGH
VKG------D
VYS--TYETS
FNA--NIPKH
IAGADNGAGY
```

Exemplo

Esta solução é demasiado simplista no que diz respeito aos espaçamentos.



Vantagens dos Profile HMMs

- Após ter um profile HMM que representa uma família de proteínas podemos:
- Usando a sequência de estados mais provável podemos criar o alinhamento múltiplo entre as várias cadeias;
- Podemos usar o modelo para calcular a probabilidade de uma dada cadeia pertencer à família;
- Podemos também usar o modelo para gerar proteínas que possam pertencer à família.

Treino de Modelos

- O objectivo é criar um profile HMM para representar uma família de proteínas;
- Criamos um profile HMM em que o número de estados normais seja dado pelo tamanho médio das sequências;
- Usamos um algoritmo de optimização para ajustar as probabilidades das transições do modelo de forma a maximizar a probabilidade da ocorrência de todas as cadeias de proteínas;
- Podemos usar muitos algoritmos de optimização para fazê-lo:
  1. Algoritmos evolucionários com codificação real;
  2. Simulated Annealing;
  3. Differential Evolution;
  4. Particle Swarm Optimization.
- A função objectivo é a mesma para todos os algoritmos.