Modelagem e Avaliação de Desempenho

Pós Graduação em Engenharia Elétrica - PPGEE

Prof. Carlos Marcelo Pedroso

Exemplos usados na apresentação foram obtidos de "Introduction to Probability, C.M.Grinstead e JL Snell"

Cadeias de Markov

- Em 1907, Andrew Markov iniciou um estudo sobre um modelo onde o resultado de um experimento depende do resultado de um experimento anterior;
- Este processo de modelagem é conhecido atualmente como Cadeias de Markov (*Markov Chain*).

Cadeias de Markov

- Uma cadeia de Markov pode ser descrita da seguinte forma
  - Considere um conjunto de estados $S=\{s_1, s_2, \ldots, s_n\}$
  - O processo inicia-se em um destes estados e se move sucessivamente de um estado para outro;
  - Cada troca é chamada de passo;
  - Se o estado corrente é s_i, então ela se move para o estado s_j com uma probabilidade denotada por p_{ij}, e esta probabilidade não depende dos estados anteriores da cadeia de Markov.
  - As probabilidades p_{ij} são chamadas de probabilidades de transição.

Matriz P

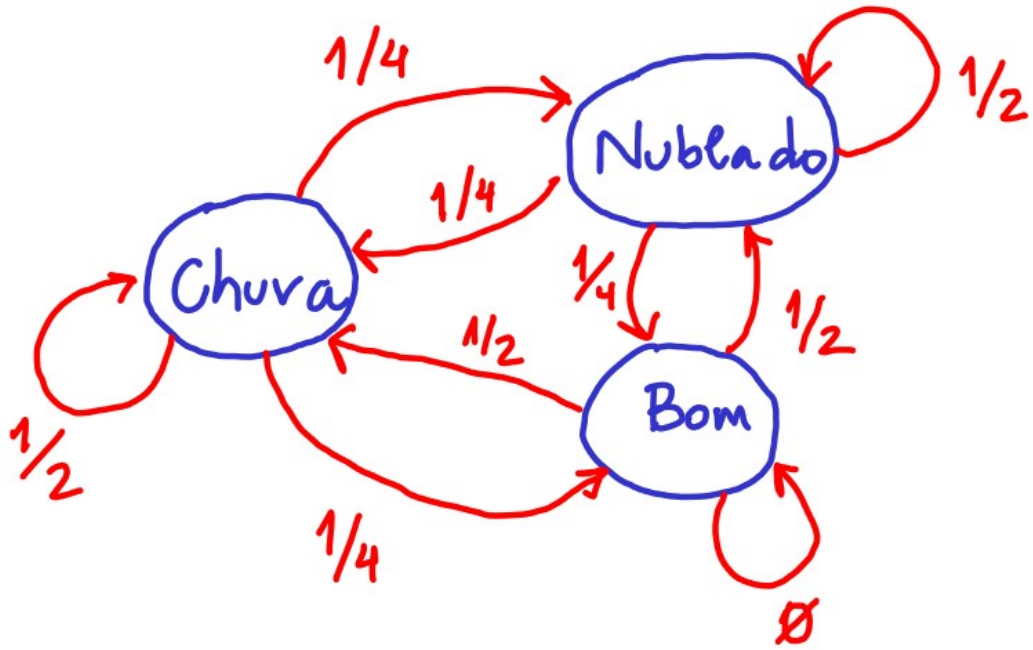
- P é chamada *matriz de transição* e possui algumas propriedades interessantes.
- As linhas representam a probabilidade de transição de um estado para outro.
- Para calcular a probabilidade da cadeia se encontrar no estado *j*, mas a *n* passos adiante, pode-se calcular P^n.

Exemplo

Suponha que Curitiba foi abençoada com muitas coisa, menos com bom tempo para astonomia. Curitiba nunca apresenta dois dias seguidos com bom tempo para observação astronômica. Se há um dia bom, existe uma chance igual de chuva ou nuvens no próximo dia. Se há chuva existe uma chance de 50% que no próximo dia haver chuva no dia seguinte, com igual probabilidade de haver tempo bom ou nublado no dia seguinte. Com o tempo nublado, o comportamento semelhante ao tempo com chuva.

Suponha a cadeia de Markov que representa a transição destes estados, onde C representa chuva, B representa tempo bom e N representa tempo nublado.

Exemplo



Maxima:

```
P:matrix([1/2,1/4,1/4],[1/2,0,1/2],[1/4,1/4,1/2]);
float(P);
```

Exemplo

- Para o caso do exemplo anterior (clima para astronomia em Curitiba), temos:

$$P^1 = \begin{array}{c} \\ C \\ B \\ N \end{array} \begin{array}{c} \begin{array}{ccc} C & B & N \end{array} \\ \begin{pmatrix} 0.500 & 0.250 & 0.250 \\ 0.500 & 0.000 & 0.500 \\ 0.250 & 0.250 & 0.500 \end{pmatrix} \end{array}$$

$$P^2 = \begin{array}{c} \\ C \\ B \\ N \end{array} \begin{array}{c} \begin{array}{ccc} C & B & N \end{array} \\ \begin{pmatrix} 0.438 & 0.188 & 0.375 \\ 0.375 & 0.250 & 0.375 \\ 0.375 & 0.250 & 0.438 \end{pmatrix} \end{array}$$

$$P^3 = \begin{array}{c} \\ C \\ B \\ N \end{array} \begin{array}{c} \begin{array}{ccc} C & B & N \end{array} \\ \begin{pmatrix} 0.406 & 0.203 & 0.391 \\ 0.406 & 0.188 & 0.406 \\ 0.391 & 0.203 & 0.406 \end{pmatrix} \end{array}$$

$$P^4 = \begin{array}{c} \\ C \\ B \\ N \end{array} \begin{array}{c} \begin{array}{ccc} C & B & N \end{array} \\ \begin{pmatrix} 0.402 & 0.199 & 0.398 \\ 0.398 & 0.203 & 0.398 \\ 0.398 & 0.199 & 0.402 \end{pmatrix} \end{array}$$

$$P^5 = \begin{array}{c} \\ C \\ B \\ N \end{array} \begin{array}{c} \begin{array}{ccc} C & B & N \end{array} \\ \begin{pmatrix} 0.400 & 0.200 & 0.399 \\ 0.400 & 0.399 & 0.400 \\ 0.399 & 0.200 & 0.400 \end{pmatrix} \end{array}$$

$$P^6 = \begin{array}{c} \\ C \\ B \\ N \end{array} \begin{array}{c} \begin{array}{ccc} C & B & N \end{array} \\ \begin{pmatrix} 0.400 & 0.200 & 0.400 \\ 0.400 & 0.200 & 0.400 \\ 0.400 & 0.200 & 0.400 \end{pmatrix} \end{array}$$

Maxima: float(P^^2); float(P^^3); ...

Matriz P

- Para se calcular a probabilidade de se encontrar no estado *j* dado um estado *i*, n passos adiante, pode-se calcular: $\mathbf{u}^{(n)} = \mathbf{u}\mathbf{P}^n$
- Exemplo: suponna que a probabilidade inicial para o clima para astronomia seja de (1/3, 1/3 e 1/3) e deseja-se fazer a previsão do tempo para 3 dias. Neste caso,

$$\mathbf{u}^{(3)} = \mathbf{u}\mathbf{P}^3 = \begin{pmatrix} 1/3, & 1/3, & 1/3 \end{pmatrix} \begin{pmatrix} .406 & .203 & .391 \\ .406 & .188 & .406 \\ .391 & .203 & .406 \end{pmatrix}$$

$$= \begin{pmatrix} .401, & .188, & .401 \end{pmatrix} .$$

Maxima: u:matrix([1/3,1/3,1/3]);
float(u.P^^3);

Exercício

Considere três grandes universidades americanas, Harvard, Darmouth e Yale. Suponha que os filhos de ex-alunos Harvard tem 80% de chance de estudar na mesma escola e os demais estudam em Yale. Suponha que 40% dos filhos de ex-alunos de Yale estudam também em Yale e os demais dividem-se igualmente entre Darmouth e e Harvard. Suponha que os filhos de ex-alunos de Darmouth tem 70% de chance de estudar em Darmouth, enquanto 20% entram em Harvard e 10% em Yale.

1) Encontre a matriz P.

2) Encontre a probabilidade de que um neto de um ex-aluno de Harvard estude em Darmouth.

3) Encontre a probabilidade de que um bisneto de um ex-aluno de Darmouth estude em Yale.

Suponha que os filhos de ex-alunos Harvard tem 80% de chance de estudar na mesma escola e os demais estudam em Yale. Suponha que 40% dos filhos de ex-alunos de Yale estudam também em Yale e os demais dividem-se igualmente entre Darmouth e e Harvard. Suponha que os filhos de ex-alunos de Darmouth tem 70% de chance de estudar em Darmouth, enquanto 20% entram em Harvard e 10% em Yale.

Cadeias de Markov Ergódicas

Uma cadeia de Markov ergódica é uma cadeia que possui as seguintes propriedades:

- **Irredutibilidade:**
  - Cada estado da cadeia pode ser alcançado a partir de qualquer outro estado, possivelmente em mais de um passo. Em outras palavras, existe um caminho (sequência de transições) de probabilidade positiva entre qualquer par de estados. Isso implica que todos os estados da cadeia estão conectados de alguma forma.
- **Recorrência Positiva:**
  - O tempo esperado para que a cadeia retorne ao mesmo estado (tempo de retorno) é finito para todos os estados. Isso significa que a cadeia não "demora" infinitamente para revisitar qualquer estado.
- **Aperiodicidade:**
  - Não existe um período fixo (número de passos) no qual o sistema revisita um estado. Em outras palavras, os estados não têm ciclos rígidos, o que garante que a cadeia não fique presa em padrões cíclicos previsíveis.

Cadeia de Markov Regular

- Uma cadeia de Markov regular é um caso específico de uma cadeia ergódica com a seguinte característica adicional:
  - Existe um número de passos *k* tal que, a partir de qualquer estado, a probabilidade de transição para qualquer outro estado é positiva em exatamente *k* passos. Em outras palavras, a matriz de transição elevada a alguma potência (P^*k*, onde *k* é um número inteiro) terá todas as entradas positivas.
- Em termos mais simples, uma cadeia regular é uma cadeia de Markov onde, após certo número de transições, todos os estados têm probabilidade positiva de transição para todos os outros estados, independentemente do estado inicial

Cadeias de Markov Regulares

- Seja P uma matriz de transição para uma cadeia de markov regular. Então, conforme *n* tende a infinito, as potências P^n se aproximam da matriz limite em que todas as linhas são iguais (vetor W). O vetor W é um vetor de probabilidade onde todos os componentes são positivos e sua soma é igual a 1.
  - Ver exemplo da terra do clima para astronomia em Curitiba.

Cadeias de Markov Ergódicas

- Suponha a matriz de transição de probabilidade dada por

$$\mathbf{P} = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}.$$

Maxima:
```
P:matrix([0,1],[1,0]);
float(P^^2);
float(P^^3);
```

A cadeia é ergódica

  - No entanto, não é regular (se o número de passo é ímpar não é possível atingir um dado estado)

Outro exemplo

$$\mathbf{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/4 & 0 & 3/4 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 3/4 & 0 & 1/4 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} .$$

Maxima:

```
P:matrix([0,1,0,0,0],[1/4,0,3/4,0,0],[0,1/2,0,1/2,0],[0,0,3/4,0,1/4],[0,0,0,1,0]);
 float(P^^2);
 float(P^^3);
```

- A cadeia é ergódica e regular?
- E a cadeia do exemplo do clima para astronomia em Curitiba?

Teorema do Limite

- Para uma cadeia de Markov Regular:
  - Convergência para a Distribuição Estacionária: Existe uma distribuição estacionária única $\mathbf{W} = [w_1, w_2, \ldots, w_n]$ tal que:

$$\mathbf{W}.\mathbf{P} = \mathbf{W}$$

  - Independência do Estado Inicial: Para qualquer distribuição $\boldsymbol{u}$ inicial a distribuição de probabilidade $\boldsymbol{u}.\mathbf{P}^t$ converge para $\mathbf{W}$ conforme t tende a infinito.
  - Estabilidade da Distribuição Limite: No longo prazo, a probabilidade de estar em qualquer estado $\boldsymbol{j}$ se aproxima de w_j, independentemente do estado inicial.
  - O sistema de equações resultante tem termo independente nulo (sistema homogêneo), que deve ser completado com $\sum_{i=1}^{n} W_i = 1$

Cadeias de Markov Regulares

- No exemplo do clima para astronomia em Curitiba:

$$\mathbf{P}^6 = \begin{array}{c} \\ \mathrm{R} \\ \mathrm{N} \\ \mathrm{S} \end{array} \begin{array}{c} \begin{array}{ccc} \mathrm{R} & \mathrm{N} & \mathrm{S} \end{array} \\ \begin{pmatrix} .4 & .2 & .4 \\ .4 & .2 & .4 \\ .4 & .2 & .4 \end{pmatrix} \end{array}$$

- De modo geral, $\mathbf{W} = \lim_{n \to \infty} \mathbf{P}^n$,

Cadeias de Markov Regulares

- Utilizando o resultado, é possível determinar o valor limite fazendo: $W.P=W$ e $\sum_{i=1}^{n} W_i = 1$

$$w_1 + w_2 + w_3 = 1$$

$$\begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} = \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} .$$

Cadeias de Markov Regulares

- Resolvendo:

$$
\begin{aligned}
w_1 + w_2 + w_3 &= 1 \,, \\
(1/2)w_1 + (1/2)w_2 + (1/4)w_3 &= w_1 \,, \\
(1/4)w_1 + (1/4)w_3 &= w_2 \,, \\
(1/4)w_1 + (1/2)w_2 + (1/2)w_3 &= w_3 \,.
\end{aligned}
$$

$$
\mathbf{w} = \begin{pmatrix} .4 & .2 & .4 \end{pmatrix}
$$

Maxima:

```
W:matrix([w1,w2,w3]);
P:matrix([1/2,1/4,1/4],[1/2,0,1/2],[1/4,1/4,1/2]);
W.P=W;
solve( [w3/4+w2/2+w1/2-w1,w3/4+w1/4-w2,w1+w2+w3-1],[w1,w2,w3] );
```

Exemplo 3

- O modelo de **Gilbert-Elliott** é um dos modelos mais famosos para caracterizar a perda de pacotes em redes.
- Este modelo usa dois estados: Bom (B) e Ruim (R). A transição do estado B para R tem probabilidade *p* e de R para B tem probabilidade *q*.
- A transição no diagrama de estados ocorre a cada novo pacote transmitido.
- A probabilidade de descarte de pacote no estado B é dado por (1-*k*) e no estado B por (1-*h*).
- Determine a probabilidade de descarte de pacotes.

Anote os desenvolvimentos apresentados na aula

Exemplo 4

- Um professor se alterna entre seu gabinete e sua casa. Se na hora da saída estiver chovendo e existir um guarda chuvas (no gabinete ou em casa), o professor sai com o guarda chuva e deixa no destino. Caso não haja guarda chuva disponível, o professor sai na chuva e se molha.
- Suponha que existe uma probabilidade *p* de chover e o professor possua 3 guarda chuvas. Determine:
  - Matriz de transição de probabilidade para o sistema e diagrama de transição de estados correspondente.
  - Probabilidade do professor se molhar.

Exemplo 5

- Considere um componente (ex. Lâmpada) que possui uma vida útil dada em unidades de tempo discreto (dias, por exemplo). Suponha que o componente fica ativo até falhar ou ser substituído no tempo N.
- A probabilidade de falha é constante dada por *p*.
- Determine a probabilidade de estado estacionário para este problema. Analise cuidadosamente a metodologia de solução porque este tipo de problema é bastante comum.
- Como esta probabilidade pode ser usada para estabelecer uma boa política de substituição (valor de N)?

Anote os desenvolvimentos apresentados na aula

Cadeias de Markov - Simulação

```
SSRNRNSSSSSSNRSNSSRNSRNSSSNSRRRNSSSNRRSSSSNRSSNSRRRRRRNSSS
SSRRRSNSNRRRRSRSRNSNSRRNRRNRSSNSRNRNSSRRSRNSSSNRSRRSSNRSNR
RNSSSSNSSNSRSRRNSSNSSRNSSRRNRRRSRNRRRNSSSNRNSRNSNRNRSSSRSS
NRSSSNSSSSSSNSSSNSNSRRNRNRRRRSRRRSSSSNRRSSSSSRSRRRNRRRSSSSR
RNRRRSRSSRRRRSSRNRRRRRRNSSRNRSSSNRNSNRRRRNRRRNRSNRRNSRRSNR
RRRSSSRNRRRNSNSSSSSRRRRSRNRSSRRRRSSSRRRNRNRRRSRSRNSNSSRRRR
RNSNRNSNRRNRRRRRRSSSNRSSRSNRSSSNSNRNSNSSSNRRSRRRNRRRRNRNRS
SSNSRSNRNRRSNRRNSRSSSRNSRRSSNSRRRNRRSNRRNSSSSSNRNSSSSSSSNR
NSRRRNSSRRRNSSSNRRSRNSSRRNRRNRSNRRRRRRRRRNSNRRRRRNSRRSSSSN
SNS
```

State	Times	Fraction
R	217	.413
N	109	.208
S	199	.379

Exercício

- Suponha que um experimento possui a matriz P como segue:

$$\mathbf{P} = \begin{pmatrix} .5 & .5 \\ p & 1-p \end{pmatrix}$$

  - O valor de *p* é desconhecido. No entanto, repetindo-se muitas vezes o experimento, 20% das vezes o sistema encontram-se no estado 1 e 80% no estado 2.
  - Encontre o valor *p*.

Cadeias de Markov Absorventes

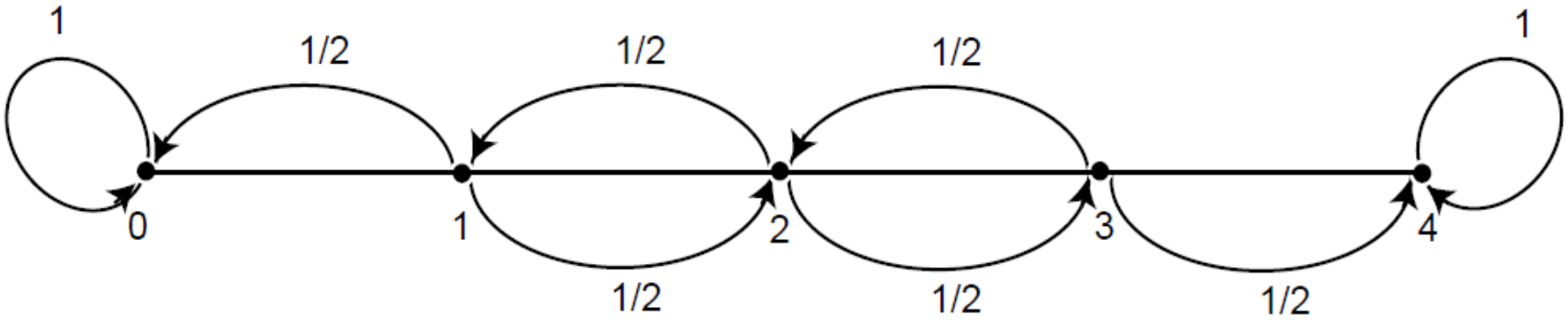
Considere uma cadeia de Markov onde existem estados onde não é possível realizar a transição para nenhum outro estado.

- Este estado é denominado estado absorvente.
- Um estado absorvente apresenta $p_{ij} = 1$.
- Esta é uma variação especial das cadeias de Markov.
- Em uma cadeia de Markov absorvente, o número de passos até atingir o estado absorvente é chamado transiente.

Cadeias de Markov Absorventes

Exemplo. Um bêbado caminha na rua. Cada número de 1 a 3 representa um quarteirão, enquanto o número 0 representa a casa dele e o número 4 representa o bar.

- Escreva a matriz P correspondente.



Cadeias de Markov Absorventes

$$\mathbf{P} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Cadeias de Markov Absorventes

- As questões que surgem são:
  - Qual a probabilidade de que o processo seja eventualmente absorvido?
  - Na média, quantos passos serão dados até que o processo seja absorvido?
  - Na média, quantas vezes um dado estado transiente será visitado até que o processo seja absorvido?
- As respostas a estas questões dependem do estado inicial e da matriz de transição.

Cadeias de Markov Absorventes

- Considere a matriz P com r estados absorventes (ABS) e t estados transientes (TR). A matriz P canônica é formada conforme abaixo:
  - I é uma matriz identidade r por r.
  - O é uma matriz 0 r por t.
  - R é uma matriz t por r.
  - Q é uma matriz t por t.

$$\mathbf{P} = \begin{array}{c} \\ \text{TR.} \\ \\ \text{ABS.} \end{array} \begin{array}{c} \begin{array}{cc} \text{TR.} & \text{ABS.} \end{array} \\ \left(\begin{array}{c|c} \mathbf{Q} & \mathbf{R} \\ \hline \mathbf{0} & \mathbf{I} \end{array} \right) \end{array}$$

Cadeias de Markov Absorventes

- Para uma cadeia de Markov absorvente, a matriz $N=(I-Q)^{-1}$ é chamada matriz fundamental para P.
  - Um elemento n_{ij} de N fornece o número esperado de vezes que o processo estará no estado transiente s_j caso o estado inicial seja o estado s_i

Cadeias de Markov Absorventes

- Exemplo:

$$\mathbf{Q} = \begin{pmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{pmatrix} ,$$

Maxima:

```
Q:matrix([0,1/2,0],[1/2,0,1/2],[0,1/2,0]);
I:matrix([1,0,0],[0,1,0],[0,0,1]);
N:invert(I-Q);
```

$$\mathbf{I} - \mathbf{Q} = \begin{pmatrix} 1 & -1/2 & 0 \\ -1/2 & 1 & -1/2 \\ 0 & -1/2 & 1 \end{pmatrix} .$$

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{pmatrix} \end{array}$$

- Iniciando-se no estado 2, o número médio de vezes em que o sistema permanece nos estados 1, 2 e 3 será, respectivamente, 1, 2 e 1.

Cadeias de Markov Absorventes

- Outro fator importante a se considerar é o número médio de passos para a absorção.
- Seja t número de passos até a absorção, dado que o estado inicial seja s_i e t ser o vetor coluna que armazena o número médio de passos para absorção a partir dos estados transientes e **c** é um vetor coluna com todos os elementos iguais a 1.
  - Então:

$$\mathbf{t} = \mathbf{Nc}$$

- Calcular para o exemplo anterior.

Resposta

$$\mathbf{N} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{pmatrix} \end{array} .$$

Maxima:

```
Q:matrix([0,1/2,0],[1/2,0,1/2],[0,1/2,0]);
I:matrix([1,0,0],[0,1,0],[0,0,1]);
N:invert(I-Q);
c:matrix([1],[1],[1]);
t:N.c;
```

$$\mathbf{t} = \mathbf{Nc} = \begin{pmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix} .$$

Cadeias de Markov Absorventes

- Um analista pode estar interessado também em calcular a probabilidade do sistema encerrar em um dos estados absorventes.
- Neste caso, se b_{ij} representar a probabilidade da cadeia ser absorvida por um estado s_j caso o estado inicial seja um estado transiente s_i, então a matriz B (t por r) será dada por:

$$\mathrm{B} = \mathrm{NR}$$

Exemplo 1

- Considerando o problema do bêbado:

$$\mathbf{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 4 \end{array} \\ \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \\ 0 & 1/2 \end{pmatrix} \end{array} .$$

Maxima:

```
Q:matrix([0,1/2,0],[1/2,0,1/2],[0,1/2,0]);
I:matrix([1,0,0],[0,1,0],[0,0,1]);
N:invert(I-Q);
c:matrix([1],[1],[1]);
t:N.c;
R:matrix([1/2,0],[0,0],[0,1/2]);
B:N.R;
```

$$\mathbf{B} = \mathbf{NR} = \begin{pmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{pmatrix} \cdot \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \\ 0 & 1/2 \end{pmatrix}$$

$$= \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 4 \end{array} \\ \begin{pmatrix} 3/4 & 1/4 \\ 1/2 & 1/2 \\ 1/4 & 3/4 \end{pmatrix} \end{array} .$$

Exemplo 2

- Considere o lançamento de uma moeda equilibrada. O resultado pode ser cara (A) ou coroa (O), com uma probabilidade de 1/2.
- Utilizando cadeias de Markov, determine o número médio de lançamentos para obter a sequência AOA.

Anote a solução apresentada na aula

Número médio de passos médio para primeira passagem e recorrência

- Duas medidas quantitativas de interesse para cadeias de Markov ergódicas são:
  - Número médio de passos para retornar a um determinado estado;
  - Número médio de passos para ir de um estado para outro.

Número médio de passos para primeira passagem

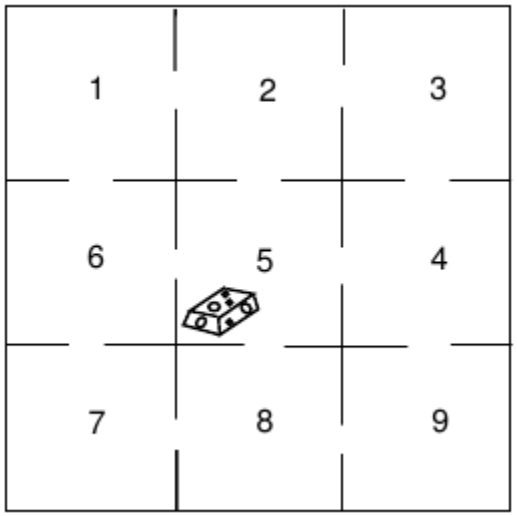
- Uma maneira de analisar o problema é o seguinte:
  - Suponha que a cadeia de Markov em estudo é ergódica (qualquer estado pode ser atingido a partir que qualquer estado inicial).
  - Para determinar o número médio de passos para atingir um determinado estado *i*, basta fazer este estado um estado absorvente.
  - Depois, apenas é necessário fazer o estudo com a teoria de cadeias de Markov absorventes.

Número médio de passos para primeira passagem

- Uma maneira de analisar o problema é o seguinte:
  - Suponha que a cadeia de Markov em estudo é ergódica (qualquer estado pode ser atingido a partir que qualquer estado inicial).
  - Para determinar o número médio de passos para atingir um determinado estado *i*, basta fazer este estado um estado absorvente.
  - Depois, apenas é necessário fazer o estudo com a teoria de cadeias de Markov absorventes.

Tempo médio para primeira passagem

- Exemplo: Labirinto



Número médio de passos para primeira passagem

- Exemplo: Labirinto

$$\mathbf{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \end{pmatrix} \end{array}$$

Como podemos determinar se o rato é mais esperto ?

Número médio de passos para primeira passagem

- Exemplo: Para calcular o tempo médio para atingir o estado 5, fazemos este estado absorvente:

$$\mathbf{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \\ \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 6 & 7 & 8 & 9 \end{array} \\ \left(\begin{array}{cccccccc|c} 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

Número médio de passos para primeira passagem

- Exemplo: Calculamos a matriz fundamental N

$$\mathbf{N} = \frac{1}{8}\begin{pmatrix} 14 & 9 & 4 & 3 & 9 & 4 & 3 & 2 \\ 6 & 14 & 6 & 4 & 4 & 2 & 2 & 2 \\ 4 & 9 & 14 & 9 & 3 & 2 & 3 & 4 \\ 2 & 4 & 6 & 14 & 2 & 2 & 4 & 6 \\ 6 & 4 & 2 & 2 & 14 & 6 & 4 & 2 \\ 4 & 3 & 2 & 3 & 9 & 14 & 9 & 4 \\ 2 & 2 & 2 & 4 & 4 & 6 & 14 & 6 \\ 2 & 3 & 4 & 9 & 3 & 4 & 9 & 14 \end{pmatrix}$$

- Iniciando-se no estado 1, o número médio de vezes em que o sistema permanece nos estados 1, 2, 3 , … etc., será, respectivamente, 14, 9, 4.

Número médio de passos para primeira passagem

- Exemplo: Labirinto

$$\mathbf{t} = \mathbf{Nc} \quad \Rightarrow \quad \mathbf{Nc} = \begin{pmatrix} 6 \\ 5 \\ 6 \\ 5 \\ 5 \\ 6 \\ 5 \\ 6 \end{pmatrix}$$

- Iniciando-se no estado 1, o sistema leva em média 6 passos para atingir o estado 5 (absorvente). Iniciando-se no estado 2, o sistema leva 5 passos para atingir o estado absorvente e assim por diante.

Número médio de passos para recorrência

- Qual será o número médio de passos em que um estado será visitado novamente?
  - Dado um estado s_i, qual será o número médio de passos que o sistema irá levar para se encontrar novamente no estado s_i no futuro?
  - Dado o vetor w, com a probabilidade limite, basta calcular $1/w_i$ e teremos o número médio de passos para visitar o estado

Número médio de passos para recorrência

- No exemplo do labirinto, pode ser calculado w.P=w (acrescentado somatório de w_i=1), obtendo-se:

$$\mathbf{w} = \left(\frac{1}{12} \quad \frac{1}{8} \quad \frac{1}{12} \quad \frac{1}{8} \quad \frac{1}{6} \quad \frac{1}{8} \quad \frac{1}{12} \quad \frac{1}{8} \quad \frac{1}{12} \right)$$

- De onde pode ser deduzido o vetor r (número médio de passos para recorrência):

$$\mathbf{r} = \left(12 \quad 8 \quad 12 \quad 8 \quad 6 \quad 8 \quad 12 \quad 8 \quad 12 \right)$$

Cadeias de Markov em T*empo Contínuo*

- *Ergodic Continous Time Markov Chain*
- A novidade é considerar a variável *tempo*.
- Neste caso, o tempo de permanência em cada transição é considerado como exponencialmente distribuído (esta é uma *exigência*, hipótese básica para validade deste raciocínio).
- Considere que o parâmetro que determina a taxa de transição do estado *i* para o próximo estado *j* seja dado por q_{ij}

Cadeias de Markov em T*empo Contínuo*

- Desta forma, podemos definir: $\Pi Q = 0, \sum_{i=1}^{s} \pi_i = 1$

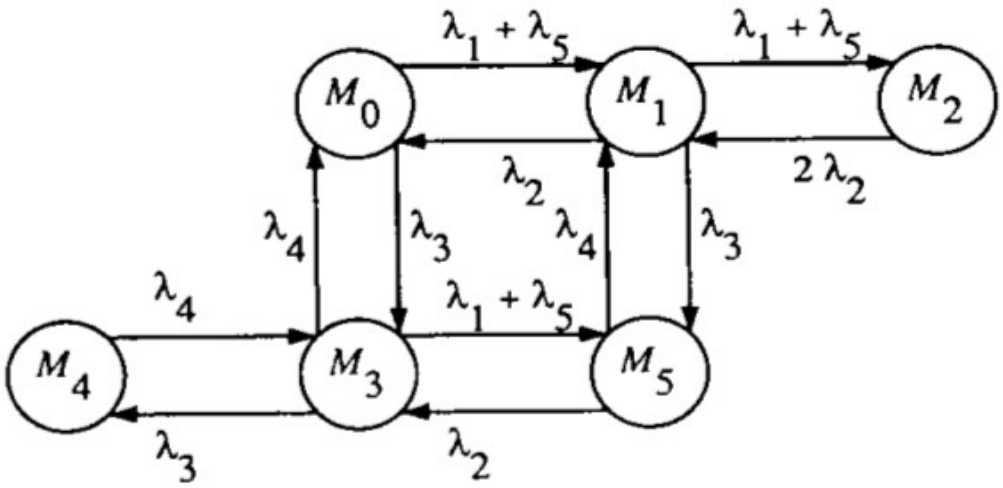
  - Onde Q é a matriz de transição de taxas
  - O vetor Π é o vetor de estado estacionário
  - Para o vetor Q, o elemento q_{ii} (diagonal principal) é obtido fazendo-se o complemento do somantório dos demais elementos da linha (ver exemplo em sala).

Cadeias de Markov em T*empo Contínuo*

- *Exemplo: Suponha dois servidores operando em cluster. Um servidor falha com uma taxa μ, exponencialmente distribuída (ou seja, o tempo médio entre falhas é dado por 1/μ). A taxa de reparo é dada por λ (ou seja, o tempo médio de reparo é dado por 1/λ). Suponha que as instalações de reparo podem trabalhar em dois servidores simultaneamente.*
  - *Deseja-se descobrir expressões para o estado estacionário.*
  - *Qual a probabilidade de falha total do sistema?*
  - *Ver solução apresentada em sala*

Cadeias de Markov em T*empo Contínuo*

- *Exercício:* Suponha um sistema com diagrama de transição de estados a seguir:



  - Suponha que as transições possuem distribuição exponencial e λ_i representa as taxas correspondentes. Calcule as probabilidades de estado estacionário.

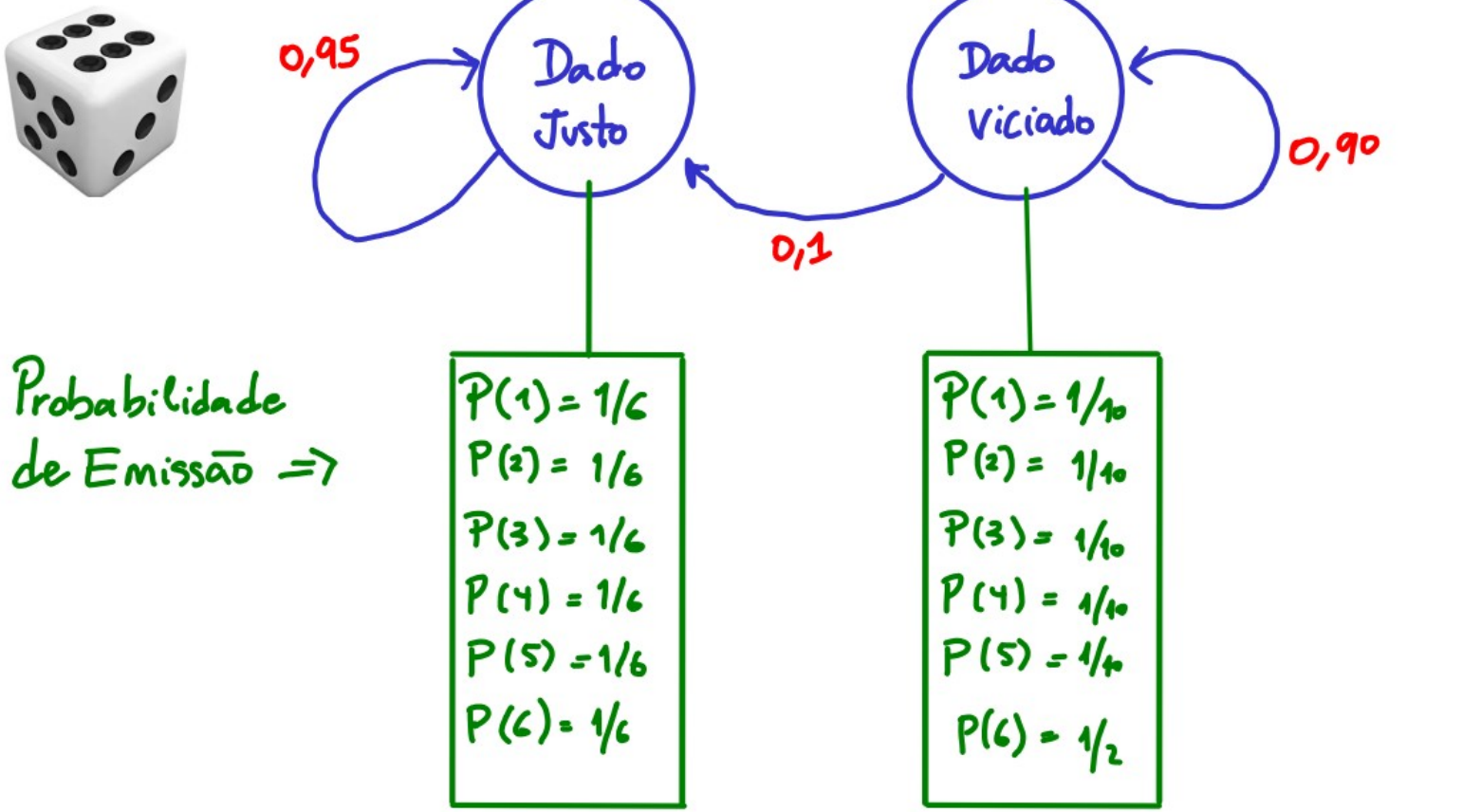
Modelos Ocultos de Markov

- *Hidden Markov Model (HMM)*
- É um sistema modelado por uma cadeia de Markov onde os estados não podem ser observados diretamente.
- Uma sequência de eventos observáveis podem ser utililizados para inferir o estado atual da cadeia de Markov.
- A HMM é definida por:
  - Conjunto de estados e probabilidades de transição.
  - Eventos observáveis.
  - Probabilidades de emissão.

Modelos Ocultos de Markov

- Exemplo: Unfair Cassino. Suponha um cassino que usa dois tipos de dado: um justo e outro viciado. O dado justo possui uma chance igual de obter valores de 1 a 6 (probabilidade de 1/6). O dado viciado possui uma probabilidade de ½ de obter o valor 6 e igual probabilidade de obter os demais valores.
- A única coisa que pode ser observada pelos jogadores é a sequência resultante do lançamento dos dados. O jogador está interessado em determinar em que momento está sendo usado o dado justo e o dado viciado.
- Ver exemplo em: *http://web.stanford.edu/class/stats366/hmmR2.html*

Exemplo: Unfair Cassino



Unfair Cassino

```
require(HMM)
 nSim = 2000
 States = c("Fair", "Unfair")
 Symbols = 1:6
 transProbs = matrix(c(0.99, 0.01, 0.02, 0.98), c(length(States),length(States)),
byrow = TRUE)

 emissionProbs = matrix(c(rep(1/6, 6), c(rep(0.1, 5), 0.5)),c(length(States),
length(Symbols)), byrow = TRUE)

 hmm = initHMM(States, Symbols, transProbs = transProbs, emissionProbs =
emissionProbs)
 sim = simHMM(hmm, nSim)
 vit = viterbi(hmm, sim$observation)


 f = forward(hmm, sim$observation)
 b = backward(hmm, sim$observation)
 i <- f[1, nSim]
 j <- f[2, nSim]
 probObservations = (i + log(1 + exp(j - i)))
 posterior = exp((f + b) - probObservations)
 x = list(hmm = hmm, sim = sim, vit = vit, posterior = posterior)
```

Exemplo Unfair Cassino

```
##Plotting simulated throws at top
 mn = "Fair and unfair die"
 xlb = "Throw nr."
 ylb = ""
plot(x$sim$observation, ylim = c(-7.5, 6), pch = 3, main =
mn,
    xlab = xlb, ylab = ylb, bty = "n", yaxt = "n")
axis(2, at = 1:6)
#######Simulated, which die was used
(truth)####################
 text(0, -1.2, adj = 0, cex = 0.8, col = "black", "True:
green = fair die")
 for (i in 1:nSim) {
    if (x$sim$states[i] == "Fair")
        rect(i, -1, i + 1, 0, col = "green", border = NA)
    else rect(i, -1, i + 1, 0, col = "red", border = NA)
   }
########Most probable path
(viterbi)######################
text(0, -3.2, adj = 0, cex = 0.8, col = "black", "Most
probable path")
for (i in 1:nSim) {
    if (x$vit[i] == "Fair")
        rect(i, -3, i + 1, -2, col = "green", border = NA)
    else rect(i, -3, i + 1, -2, col = "red", border = NA)
}
##################Differences:
text(0, -5.2, adj = 0, cex = 0.8, col = "black",
"Difference")
differing = !(x$sim$states == x$vit)
for (i in 1:nSim) {
    if (differing[i])
        rect(i, -5, i + 1, -4, col = rgb(0.3, 0.3, 0.3),
            border = NA)
    else rect(i, -5, i + 1, -4, col = rgb(0.9, 0.9, 0.9),
        border = NA)
       }
```

```
#################Posterior-
probability:#########################
 points(x$posterior[2, ] - 3, type = "l")
 ###############Difference with classification by
posterior-probability:############
 text(0, -7.2, adj = 0, cex = 0.8, col = "black",
"Difference by posterior-probability")
 differing = !(x$sim$states == x$vit)
 for (i in 1:nSim) {
    if (posterior[1, i] > 0.5) {
        if (x$sim$states[i] == "Fair")
            rect(i, -7, i + 1, -6, col = rgb(0.9, 0.9,
0.9),
              border = NA)
        else rect(i, -7, i + 1, -6, col = rgb(0.3, 0.3,
0.3),
            border = NA)
     }
     else {
         if (x$sim$states[i] == "Unfair")
             rect(i, -7, i + 1, -6, col = rgb(0.9, 0.9,
0.9),
               border = NA)
         else rect(i, -7, i + 1, -6, col = rgb(0.3, 0.3,
0.3),
            border = NA)
      }
    }
```

Modelos Ocultos de Markov

- Os principais problemas da HMM são:
  1) Determinar o modelo (número de estados, transições, eventos observáveis).
  2) Realizar o treinamento (encontrar as probabilidades de transição e as probabilidades de emissão).
  3) Decodificar o estado mais provável a partir de uma sequência de observação.

Modelos Ocultos de Markov

- O analista é fundamental para a primeira questão. O número de estados pode ser pesquisado através de algoritmos de clusterização como o *kmeans* ou com métodos gráficos como o *dendograma*.
- O segundo problema (treinamento ou aprendizado) é resolvido com a aplicação do algoritmo de Baum-Welch. Para aplicação deste algoritmo deve estar disponível um conjunto de dados para o treinamento.
- O terceiro problema, encontrar o estado mais provável do modelo oculto a partir de uma sequência observável, pode ser resolvido com o algoritmo de Viterbi.

Clusterização

- Clusterização (ou clustering) é uma técnica de aprendizado de máquina não supervisionado usada para agrupar objetos ou dados em subconjuntos chamados clusters, com base em características similares. Os elementos dentro de um cluster têm maior similaridade entre si e menor similaridade com os elementos de outros clusters.
- https://www.stat.berkeley.edu/~s133/Cluster2a.html

Tipos de Clusterização

- Particional:
  - Divide os dados em k k grupos, onde k k é especificado pelo usuário. Exemplo: K-means, K-medoids.
- Hierárquica:
  - Cria uma hierarquia de clusters em forma de árvore (dendrograma). Exemplo: Aglomerativa (bottom-up) e Divisiva (top-down).
- Baseada em Densidade:
  - Identifica clusters com base na densidade de pontos em uma região. Exemplo: DBSCAN, OPTICS.

Tipos de Clusterização

- Baseada em Modelos:
  - Assume que os dados seguem um modelo estatístico (e.g., distribuições gaussianas). Exemplo: Gaussian Mixture Models (GMMs).

- Baseada em Grafos: Usa conectividade ou relações em grafos para identificar clusters. Exemplo: Spectral Clustering.

- Ver exemplo: https://www.stat.berkeley.edu/~s133/Cluster2a.html

Modelos Ocultos de Markov

- O analista é fundamental para a primeira questão. O número de estados pode ser pesquisado através de algoritmos de clusterização como o *kmeans* ou com métodos gráficos como o *dendograma*.
- O segundo problema (treinamento ou aprendizado) é resolvido com a aplicação do algoritmo de Baum-Welch. Para aplicação deste algoritmo deve estar disponível um conjunto de dados para o treinamento.
- O terceiro problema, encontrar o estado mais provável do modelo oculto a partir de uma sequência observável, pode ser resolvido com o algoritmo de Viterbi.

Modelos Ocultos de Markov

- O analista é fundamental para a primeira questão. O número de estados pode ser pesquisado através de algoritmos de clusterização como o *kmeans* ou com métodos gráficos como o *dendograma*.
- O segundo problema (treinamento ou aprendizado) é resolvido com a aplicação do algoritmo de Baum-Welch. Para aplicação deste algoritmo deve estar disponível um conjunto de dados para o treinamento.
- O terceiro problema, encontrar o estado mais provável do modelo oculto a partir de uma sequência observável, pode ser resolvido com o algoritmo de Viterbi.

Algoritmo de Baum-Welch

- O algoritmo de Baum-Welch é uma técnica de aprendizado não supervisionado para ajustar os parâmetros de um Modelo Oculto de Markov (HMM). Ele é usado quando o modelo possui uma estrutura conhecida (número de estados e observações), mas os parâmetros de transição, emissão e probabilidade inicial precisam ser estimados a partir de dados observados.

Modelos Ocultos de Markov

- *Hidden Markov Model (HMM)*
- É um sistema modelado por uma cadeia de Markov onde os estados não podem ser observados diretamente.
- Uma sequência de eventos observáveis podem ser utililizados para inferir o estado atual da cadeia de Markov.
- A HMM é definida por:
  - Conjunto de estados e probabilidades de transição.
  - Eventos observáveis.
  - Probabilidades de emissão.

Algoritmo de Baum-Welch

- É um algoritimo de aprendizado não supervisionado
- O algoritmo é uma aplicação do método de máxima verossimiḷhança (Maximum Likelihood Estimation) baseado na expectativa-maximização (EM), e tem como objetivo maximizar a probabilidade da sequência observada, ajustando os parâmetros do modelo. Infere os estados ocultos com base nas observações e ajusta iterativamente os parâmetros.
- No exemplo do *unfair cassino*:

```
# Simulate a sequence of states and observations
set.seed(42)  # For reproducibility
simulated <- simHMM(hmm, 100)  # Generate a sequence of 100 rolls
observedSequence <- simulated$observation  # Use the observe rolls# Train the HMM using Baum-Welch
trainedHMM <- baumWelch(hmm, observedSequence, maxIterations = 50)

# View updated model parameters
trainedHMM

# Decode the most likely sequence of states using Viterbi algorithm
viterbiStates <- viterbi(trainedHMM, observedSequence)
viterbiStates
```

Exemplo de aprendizado supervisionado

```
library(HMM)

# 1. Definir parâmetros iniciais do HMM
states <- c("Fair", "Loaded")
observations <- as.character(1:6)

transition_initial <- matrix(c(
  0.9, 0.1,  # Fair -> Fair, Loaded
  0.2, 0.8   # Loaded -> Fair, Loaded
), nrow = 2, byrow = TRUE, dimnames = list(states, states))

emission_initial <- matrix(c(
  rep(1/6, 6),        # Fair die
  c(0.05, 0.05, 0.05, 0.05, 0.05, 0.75)  # Loaded die
), nrow = 2, byrow = TRUE, dimnames = list(states, observations))

initial_probs <- c(0.8, 0.2)  # Chance inicial de começar no estado "Fair"

# Criar o modelo inicial
hmm <- initHMM(states, observations, initial_probs, transition_initial, emission_initial)

# 2. Simular dados supervisionados
simulate_supervised <- function(hmm, num_rolls) {
  states <- hmm$States
  observations <- hmm$Symbols
  transition <- hmm$transProbs
  emission <- hmm$emissionProbs

  state_sequence <- character(num_rolls)
  observation_sequence <- character(num_rolls)
```

Exemplo de aprendizado supervisionado

```
  # Escolher o primeiro estado
  state_sequence[1] <- sample(states, 1, prob = hmm$startProbs)

  # Gerar observações
  for (t in 1:num_rolls) {
    current_state <- state_sequence[t]
    observation_sequence[t] <- sample(observations, 1, prob = emission[current_state, ])

    if (t < num_rolls) {
      state_sequence[t + 1] <- sample(states, 1, prob = transition[current_state, ])
    }
  }

  list(states = state_sequence, observations = observation_sequence)
}

set.seed(42)  # Para consistência
simulated_data <- simulate_supervised(hmm, num_rolls = 100)

# Visualizar dados simulados
cat("Estados supervisionados:\n", paste(simulated_data$states, collapse = " "), "\n")
cat("Observações supervisionadas:\n", paste(simulated_data$observations, collapse = " "), "\n")

# 3. Treinar supervisionadamente
train_supervised <- function(states, observations, state_sequence, observation_sequence) {
  # Contar frequências de transições e emissões
  transition_counts <- matrix(0, nrow = length(states), ncol = length(states),
                              dimnames = list(states, states))
  emission_counts <- matrix(0, nrow = length(states), ncol = length(observations),
                            dimnames = list(states, observations))
```

Exemplo de aprendizado supervisionado

```
  for (t in 1:(length(state_sequence) - 1)) {
    current_state <- state_sequence[t]
    next_state <- state_sequence[t + 1]
    transition_counts[current_state, next_state] <- transition_counts[current_state, next_state] + 1

    current_observation <- observation_sequence[t]
    emission_counts[current_state, current_observation] <- emission_counts[current_state, current_observation] + 1
  }

  # Incluir última observação no cálculo de emissões
  last_state <- state_sequence[length(state_sequence)]
  last_observation <- observation_sequence[length(observation_sequence)]
  emission_counts[last_state, last_observation] <- emission_counts[last_state, last_observation] + 1

  # Calcular probabilidades normalizadas
  transition_probs <- transition_counts / rowSums(transition_counts)
  emission_probs <- emission_counts / rowSums(emission_counts)

  list(transition_probs = transition_probs, emission_probs = emission_probs)
}

# Treinar o modelo com dados supervisionados
trained_params <- train_supervised(states, observations, simulated_data$states, simulated_data$observations)

# Mostrar resultados
cat("\nProbabilidades de Transição Treinadas:\n")
print(trained_params$transition_probs)

cat("\nProbabilidades de Emissão Treinadas:\n")
print(trained_params$emission_probs)
```

Exemplo de aprendizado supervisionado

```
  # Instalar pacote necessário
if (!requireNamespace("HMM", quietly = TRUE)) {
  install.packages("HMM")
}
library(HMM)

# 1. Definir parâmetros iniciais do HMM
states <- c("Fair", "Loaded")
observations <- as.character(1:6)

transition_initial <- matrix(c(
  0.9, 0.1,  # Fair -> Fair, Loaded
  0.2, 0.8   # Loaded -> Fair, Loaded
), nrow = 2, byrow = TRUE, dimnames = list(states, states))

emission_initial <- matrix(c(
  rep(1/6, 6),        # Fair die
  c(0.05, 0.05, 0.05, 0.05, 0.05, 0.75)  # Loaded die
), nrow = 2, byrow = TRUE, dimnames = list(states, observations))

initial_probs <- c(0.8, 0.2)  # Chance inicial de começar no estado "Fair"

# Criar o modelo inicial
hmm <- initHMM(states, observations, initial_probs, transition_initial, emission_initial)

# 2. Simular dados supervisionados
simulate_supervised <- function(hmm, num_rolls) {
  states <- hmm$States
  observations <- hmm$Symbols
  transition <- hmm$transProbs
  emission <- hmm$emissionProbs

  state_sequence <- character(num_rolls)
  observation_sequence <- character(num_rolls)

  # Escolher o primeiro estado
  state_sequence[1] <- sample(states, 1, prob = hmm$startProbs)

  # Gerar observações
  for (t in 1:num_rolls) {
    current_state <- state_sequence[t]
    observation_sequence[t] <- sample(observations, 1, prob = emission[current_state, ])

    if (t < num_rolls) {
      state_sequence[t + 1] <- sample(states, 1, prob = transition[current_state, ])
    }
  }

  list(states = state_sequence, observations = observation_sequence)
}

set.seed(42)  # Para consistência
simulated_data <- simulate_supervised(hmm, num_rolls = 100)

# Visualizar dados simulados
cat("Estados supervisionados:\n", paste(simulated_data$states, collapse = " "), "\n")
cat("Observações supervisionadas:\n", paste(simulated_data$observations, collapse = " "), "\n")

# 3. Treinar supervisionadamente
train_supervised <- function(states, observations, state_sequence, observation_sequence) {
  # Contar frequências de transições e emissões
  transition_counts <- matrix(0, nrow = length(states), ncol = length(states),
                              dimnames = list(states, states))
  emission_counts <- matrix(0, nrow = length(states), ncol = length(observations),
                            dimnames = list(states, observations))

  for (t in 1:(length(state_sequence) - 1)) {
    current_state <- state_sequence[t]
    next_state <- state_sequence[t + 1]
    transition_counts[current_state, next_state] <- transition_counts[current_state, next_state] + 1

    current_observation <- observation_sequence[t]
    emission_counts[current_state, current_observation] <- emission_counts[current_state, current_observation] + 1
  }

  # Incluir última observação no cálculo de emissões
  last_state <- state_sequence[length(state_sequence)]
  last_observation <- observation_sequence[length(observation_sequence)]
  emission_counts[last_state, last_observation] <- emission_counts[last_state, last_observation] + 1

  # Calcular probabilidades normalizadas
  transition_probs <- transition_counts / rowSums(transition_counts)
  emission_probs <- emission_counts / rowSums(emission_counts)

  list(transition_probs = transition_probs, emission_probs = emission_probs)
}

# Treinar o modelo com dados supervisionados
trained_params <- train_supervised(states, observations, simulated_data$states, simulated_data$observations)

# Mostrar resultados
cat("\nProbabilidades de Transição Treinadas:\n")
print(trained_params$transition_probs)

cat("\nProbabilidades de Emissão Treinadas:\n")
print(trained_params$emission_probs)
```

HMM: Algoritmo de Viterbi

- O algoritmo de Viterbi é um método de programação dinâmica usado para encontrar a sequência de estados mais provável em um Modelo Oculto de Markov (HMM), dado uma sequência de observações. Ele é amplamente utilizado em problemas onde os estados ocultos precisam ser inferidos com base em dados observados, como reconhecimento de fala, análise de sequências biológicas, e processamento de linguagem natural.
  - O algoritmo é eficiente, com complexidade $O(N^2T)$, onde N é o número de estados e T o comprimento da sequência observada.

HMM: Exemplo

- Exemplo de aplicação. Prever situações onde um estudante terá sucesso ou não em um curso de graduação.
  - Observações: notas, faltas, aprovações, horas matriculadas, horas aproveitadas, horas reprovadas, número de disciplinas
  - A primeira pergunta: qual o modelo oculto?

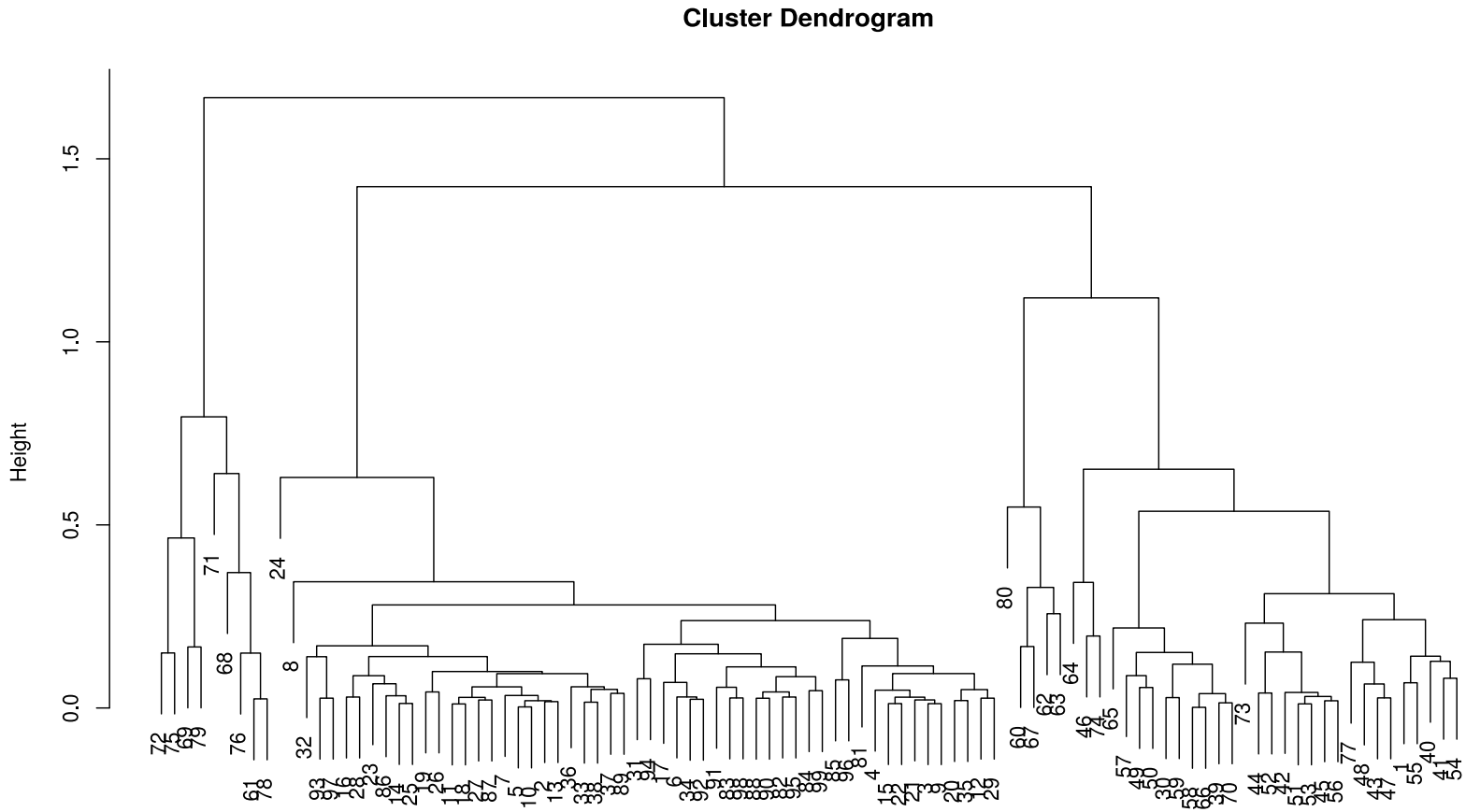
HMM: Exemplo previsão curso

- Para encontrar o número provável de estados do modelo oculto pode-se aplicar técnicas de clusterização.
- No exemplo: foram extraídas notas médias semestrais, frequências, número de disciplinas cursadas por semestre:

Aluno	Nota	Freq	Disc
1	0.8415	0.9186	0.6812
2	0.8810	0.9147	0.1014
3	0.8512	0.8795	0.0870
4	0.8737	0.8806	0.0725
5	0.8761	0.9047	0.0870
6	0.8713	0.9580	0.1159
7	0.8626	0.9267	0.1014
8	0.6959	0.9972	0.0435
9	0.8547	0.8683	0.0870
10	0.8761	0.9019	0.0870
..	...	...	...

Ver exemplo de uso do R durante a aula

Dendograma: HMM: Exemplo previsão curso



Ver exemplo de uso do R durante a aula

Modelo Sugerido

- Cadeia de Markov com 8 estados.
- Completamente conectado.
- Treinamento supervisionado.
- Interpretação do significado dos estados: verificar após o treinamento.
- Dois estados serão absorventes: sucesso ou fracasso.

Ver exemplo de uso do R durante a aula