

Cadeias de Markov

?!

- O que esteve na origem do grande sucesso inicial da Google ?
- O que tem em comum esse sucesso com a capacidade de interagir por voz com computadores, robôs e smartphones ?
 - No reconhecimento de fala ?
 - Na síntese de fala ?

Exemplo 1

- Suponhamos que em cada dia que têm aulas de MPEI acordam e decidem se vêm ou não à aula.
- Se vieram à aula anterior, a probabilidade de virem é 70%;
- se faltaram à anterior, essa probabilidade é 80%
- Algumas questões:
 - Se vieram à aula esta segunda, qual a probabilidade de virem na aula de **SEGUNDA** da próxima semana ?
 - Assumindo que o semestre tem duração infinita (que horror!), qual a percentagem aproximada de aulas a que estariam presentes ?

Exemplo 2

- Dividir a turma em 3 grupos A, B e C no início do semestre
- No final de cada aula:
- $\frac{1}{3}$ do grupo A vai para o B e outro $\frac{1}{3}$ do grupo A vai para o grupo C
- $\frac{1}{4}$ do grupo B vai para A e $\frac{1}{4}$ de B vai para C
- $\frac{1}{2}$ do grupo C vai para o grupo B
- Como ficarão os grupos ao fim de n aulas ?

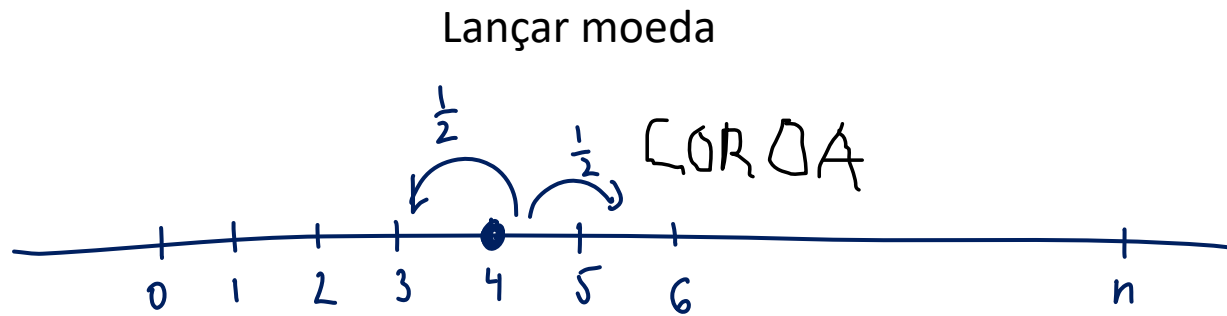
Exemplo 3 – “Pub Crawl”

- Bares junto a uma conhecida Universidade:



Outro exemplo

- Passeio aleatório (random walk)



Cara Coroa Coroa ... Cara ...

Muitas áreas de aplicação

- Muitas vezes estamos interessados na transição de algo entre certos estados.
- Exemplos:
 - Movimento de pessoas entre regiões
 - Estado do tempo
 - Movimento entre as posições num jogo de Monopólio
 - Pontuação ao longo de um jogo
 - Estado de Filas de atendimento

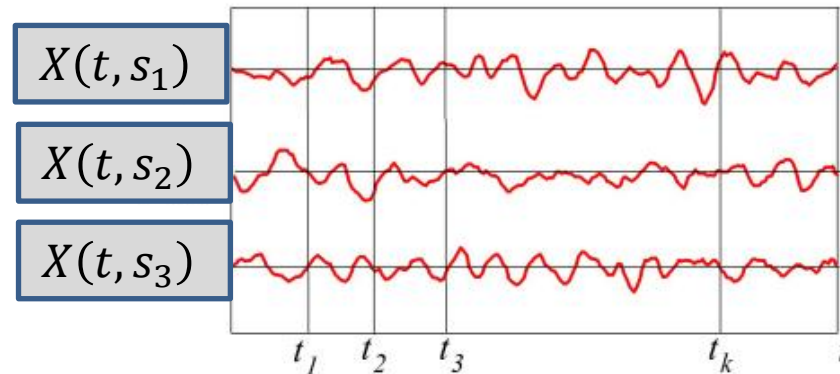
Princípios básicos

Processos estocásticos

- Estendem o conceito de variável aleatória
- Lidam com a dinâmica da teoria de probabilidades
- Uma v.a. X mapeia um acontecimento $s \in \Omega$ num número $X(s)$
- O processo mapeia o evento para números diferentes em tempos diferentes
 - O que implica que em lugar de termos um número $X(s)$ temos $X(t, s)$
 - Sendo $t \in T$ geralmente um conjunto de tempos

Processos estocásticos

- 3 realizações de um processo estocástico

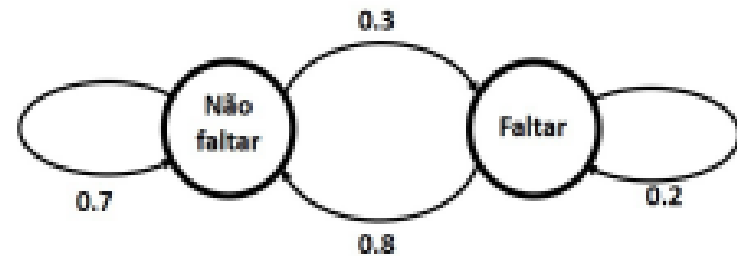
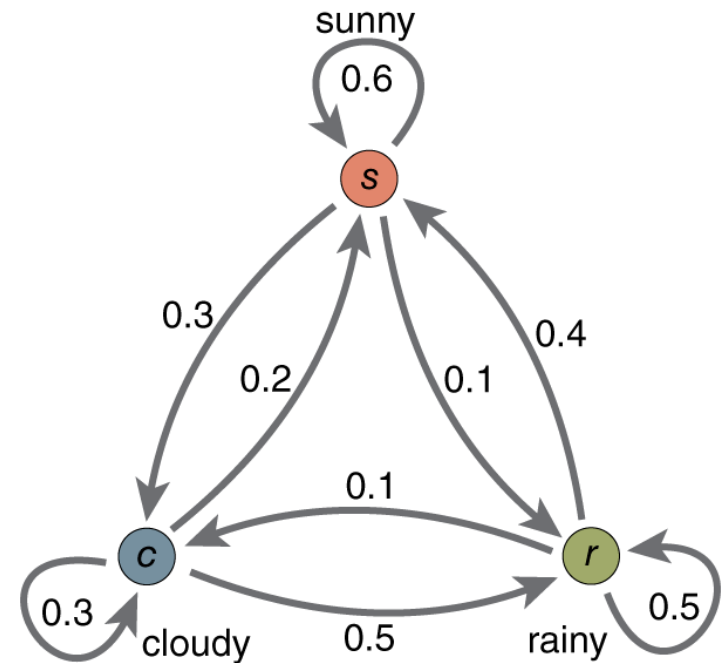
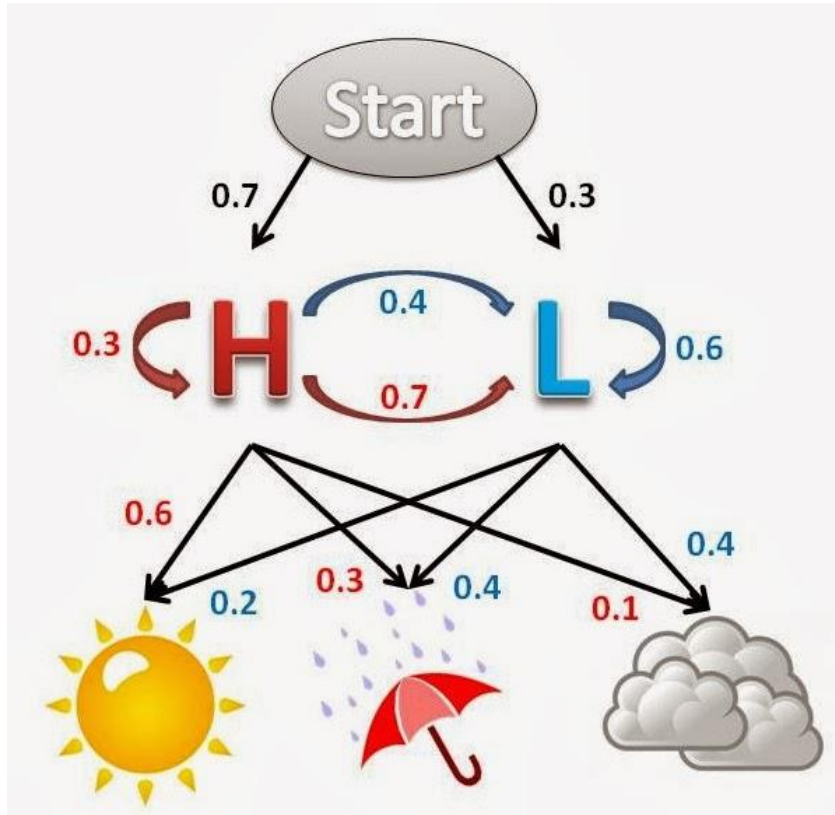


- Se fixarmos s , $X(t)$ é uma função real do tempo
- $X(t, s)$ pode então ser vista como uma **coleção de funções no tempo**
- Se fixarmos t temos uma função $X(s)$ que depende apenas de s , ou seja uma variável aleatória
- Um nome alternativo é processos aleatórios

Classificação de processos estocásticos

- Podem ser classificados segundo t e os valores que pode assumir (estados do processo)
- Quanto ao tempo :
 - Tempo contínuo: Se tempo é um intervalo contínuo
 - Tempo discreto: Se o tempo é um conjunto contável
 - Também chamada sequência aleatória e representada por $X[n]$
- Quanto ao conjunto de estados (E):
 - Contínuo
 - Discreto

Estados



Definição

- Um **processo de Markov** é um **processo estocástico** em que a probabilidade de o sistema **estar num estado** específico num determinado período de observação **depende apenas do seu estado** no período de observação imediatamente **precedente**
 - O futuro apenas depende do presente e não do passado

Tipos de processos de Markov

- Discretas/contínuas

		Espaço de estados	
		Discreto	Contínuo
Tempo	Discreto	Cadeia de Markov tempo discreto	Processo de Markov em tempo discreto
	Contínuo	Cadeia de Markov tempo contínuo	Processo de Markov em tempo contínuo

- Focaremos a nossa atenção em **cadeias de Markov de tempo discreto**

Cadeias de Markov discretas

- X_n : estado após n transições
 - Pertence a um conjunto finito,
 - Em geral $\{1, 2, \dots, m\}$
 - X_0 é dado ou aleatório

Questões comuns relativas a cadeias de Markov

- Qual a probabilidade de transição entre dois estados em n observações ?
- Existe algum equilíbrio ?
- Existe uma estabilidade a longo prazo ?

Propriedade de Markov

- Probabilidade de transição do estado i para o estado j :
- $p_{ji} = P(X_{n+1} = j | X_n = i, X_{n-1} = x_{n-1}, \dots, X_0 = x_0)$
 $= P(X_{n+1} = j | X_n = i)$
- Quando estas probabilidades p_{ji} não dependem de n a cadeia diz-se homogénea
 - Focaremos a nossa atenção neste tipo de cadeias de Markov

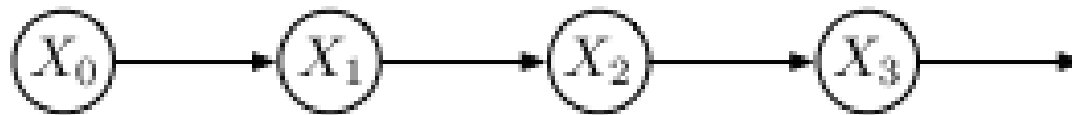
Propriedade de Markov

- $P(X_0 = x_0, X_1 = x_1, X_2 = x_2 \dots) = ?$

$$= P(X_0 = x_0) P(X_1 = x_1 | X_0 = x_0) P(X_2 = x_2 | X_0 = x_0, X_1 = x_1) \dots$$

$$= P(X_0 = x_0) P(X_1 = x_1 | X_0 = x_0) P(\textcolor{blue}{X}_2 = \textcolor{blue}{x}_2 | \textcolor{blue}{X}_1 = \textcolor{blue}{x}_1) \dots$$

- O processo “não tem memória”



Especificação de uma cadeia

- Identificar os **estados** possíveis
- Identificar as **transições** possíveis
- Identificar as **probabilidades de transição**

Aplicando ao exemplo 1 – faltar ou não faltar à aula

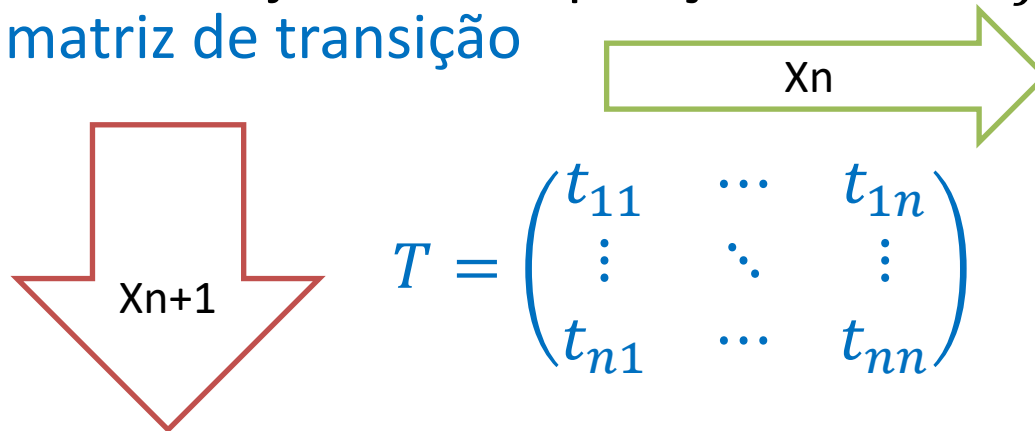
- Estados ?
- Transições ?
- Probabilidades de transição ?

Aplicando ao exemplo 1 – faltar ou não faltar à aula

- Estados ?
 - 2: {faltar, não faltar}
- Probabilidades de transição ?
 - Faltar-> não faltar : 0,8
 - Não faltar -> faltar : 0,3
 - Faltar -> faltar : 0,2
 - Não faltar -> não faltar: 0,7
- Transições ?
 - Faltar-> não faltar
 - Não faltar -> faltar
 - Faltar -> faltar
 - Não faltar -> não faltar

Matriz de transição

- É usual representar as probabilidades de transição através de uma matriz, **chamada de matriz de transição**
- Tendo o sistema n estados possíveis, para cada par i, j fazemos t_{ji} igual à probabilidade de mudar **do estado i para o estado j** .
- A matriz T cujo valor na posição linha = j , coluna = i é t_{ji} é a **matriz de transição**


$$T = \begin{pmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nn} \end{pmatrix}$$

- Nota: Alguns autores adoptam t_{ij} como a probabilidade de mudar do estado i para o estado j

Matriz T do Exemplo 1

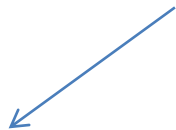
- $T = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix}$
- Considerando estado 1 “não faltar”, temos
- $T = \begin{matrix} \text{não faltar} & \rightarrow & (0,7 & 0,8) \\ \text{faltar} & \rightarrow & (0,3 & 0,2) \end{matrix}$

Matriz T do Exemplo 2

$$T = \begin{array}{c|ccc} & A & B & C \\ \hline A & ? & ? & ? \\ B & ? & ? & ? \\ C & ? & ? & ? \end{array}$$

Matriz T do Exemplo 2

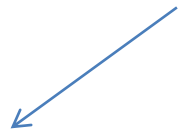
$$T = \begin{array}{c|ccc} & A & B & C \\ \hline A & 1/3 & & \\ B & 1/3 & & \\ C & 1/3 & & \end{array}$$



Futuro Estado

Matriz T do Exemplo 2

$$T = \begin{array}{c|ccc} & A & B & C \\ \hline A & 1/3 & 1/4 & 0 \\ B & 1/3 & 1/2 & 1/2 \\ C & 1/3 & 1/4 & 1/2 \end{array}$$



Futuro Estado

Matriz T é estocástica

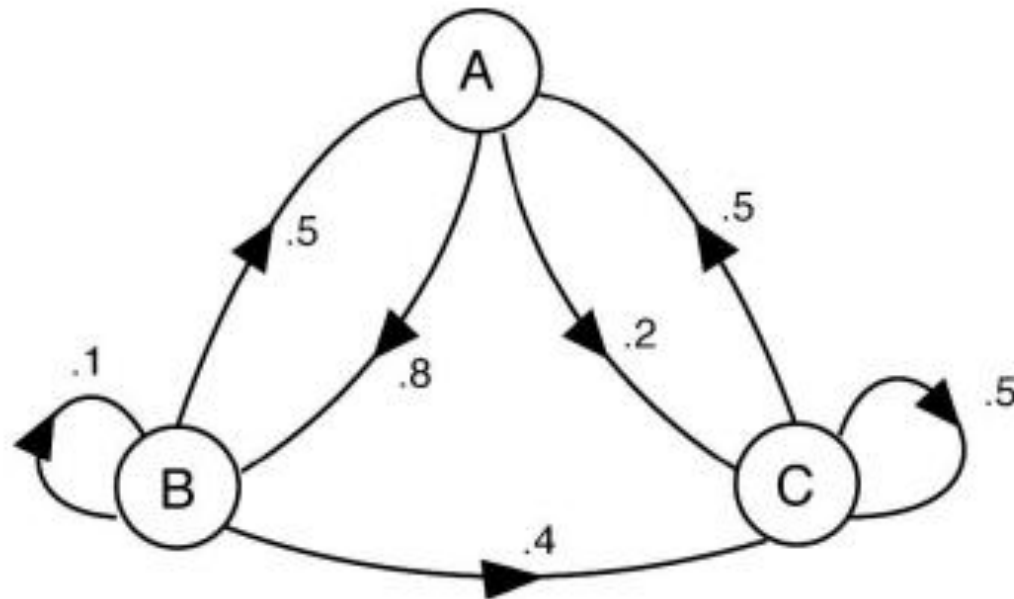
- A matriz de transição reflecte propriedades importantes das probabilidades:
 - Todas as entradas são não-negativas
 - Os valores em **cada COLUNA** somados dão sempre resultado 1
- Devido a estas propriedades a matriz é denominada de **matriz estocástica**

Representação gráfica da cadeia

- Apropriada e possível para número de estados pequeno
- **Nós**: representam todos os **estados**
- **Setas**: para todas as **transições permitidas** (one-step)
 - Ou seja, seta entre i e j apenas de $p_{ji} > 0$

Representação gráfica da cadeia

- Exemplo:



Simulação / Visualização dinâmica

- Estão disponíveis online formas de visualizar as transições entre estados ao longo do tempo ...
- Um desses exemplos é **Markov Chains - A visual explanation by [Victor Powell](#)**

- <http://setosa.io/blog/2014/07/26/markov-chains/index.html>

Que inclui:

- <http://setosa.io/markov/index.html#%7B%22tm%22%3A%5B%5B0.5%2C0.5%5D%2C%5B0.5%2C0.5%5D%5D%7D>

- Para usar precisamos apenas de introduzir a matriz T
 - Que define o número de estados, quais as transições possíveis e as probabilidades associadas a essas transições

Simulando os nossos exemplos

- Exemplo 1:
 - Matriz:
 $\begin{bmatrix} 0.7 & 0.3 \\ 0.8 & 0.2 \end{bmatrix}$
 - <http://setosa.io/markov/index.html#%7B%22tm%22%3A%5B%5B0.7%2C0.3%5D%2C%5B0.8%2C0.2%5D%5D%7D>
- Exemplo 2:
 - Matriz:
 $\begin{bmatrix} 0.33 & 0.33 & 0.34 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.5 & 0.5 \end{bmatrix}$
- Outro exemplo
 - $\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0.2 & 0.3 & 0.3 & 0.2 \end{bmatrix}$
 - O que vamos ver ?
 - Acesso directo:
<http://setosa.io/markov/index.html#%7B%22tm%22%3A%5B%5B0%2C1%2C0%2C0%5D%2C%5B0%2C0%2C1%2C0%5D%2C%5B0%2C0%2C0%2C1%5D%2C%5B0.2%2C0.3%2C0.3%2C0.2%5D%5D%7D>

Estado da cadeia num determinado instante

- O **estado** de uma cadeia de Markov com n estados no tempo (time step) k é dado pelo **vector estado**

$$\mathbf{x}^{(k)} = \begin{pmatrix} p_1^{(k)} \\ p_2^{(k)} \\ \vdots \\ p_n^{(k)} \end{pmatrix}$$

- Onde $p_j^{(k)}$ é a probabilidade de o sistema estar no estado j no instante de tempo k

Vector estado/probabilidade

- Considerando o exemplo 1:
- Suponhamos que após 10 aulas a probabilidade de faltar e não faltar são iguais
- Então o vector representativo do estado (state vector) seria:

$$\mathbf{x}^{(10)} = \begin{pmatrix} 0,5 \\ 0,5 \end{pmatrix}$$

- Este vector também se designa por **vector de probabilidade**
 - Todos elementos não-negativos
 - Soma dos elementos igual a um

Exemplo 2

- Supondo que começávamos com 20 estudantes no grupo A e 10 estudantes nos outros dois grupos, o vector relativo ao estado inicial seria

- $\mathbf{x}^{(0)} = \begin{pmatrix} 0,5 \\ 0,25 \\ 0,25 \end{pmatrix}$

Vector estado após uma transição

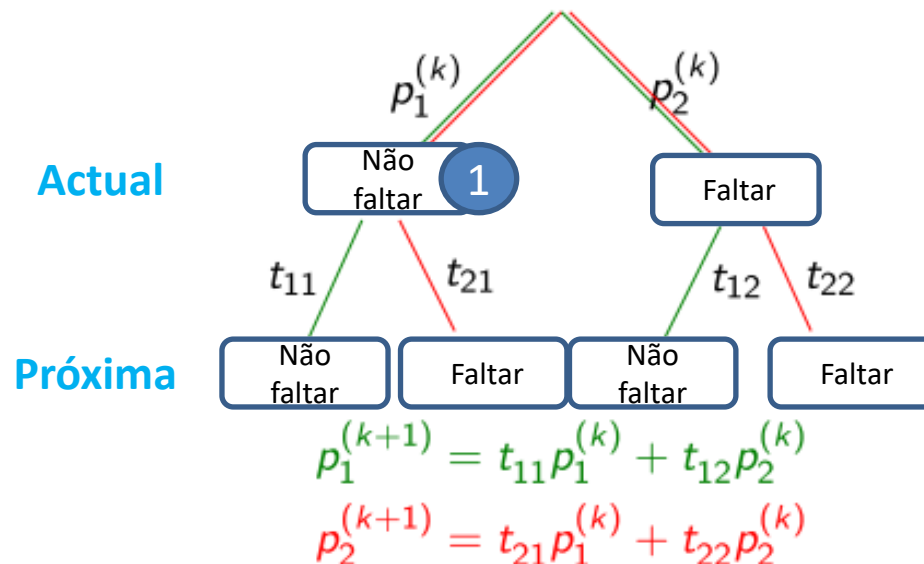
- Como obter $\mathbf{x}^{(k+1)}$?
- O vector de estado $\mathbf{x}^{(k+1)}$ no período de observação $k + 1$ pode ser determinado a partir do vector $\mathbf{x}^{(k)}$ através de:

$$\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)}$$

- Que resulta da probabilidade condicional:
- $P(\text{estado } j \text{ em } t = k + 1)$
- $= \sum_{i=1}^n P(\text{transição do estado } i \text{ para o } j)P(\text{estado } i \text{ em } t = k)$

Exemplo de aplicação – Exemplo 1

- De que forma depende a probabilidade de ir à aula seguinte da probabilidade de estar na aula actual ?



Estado após múltiplas transições

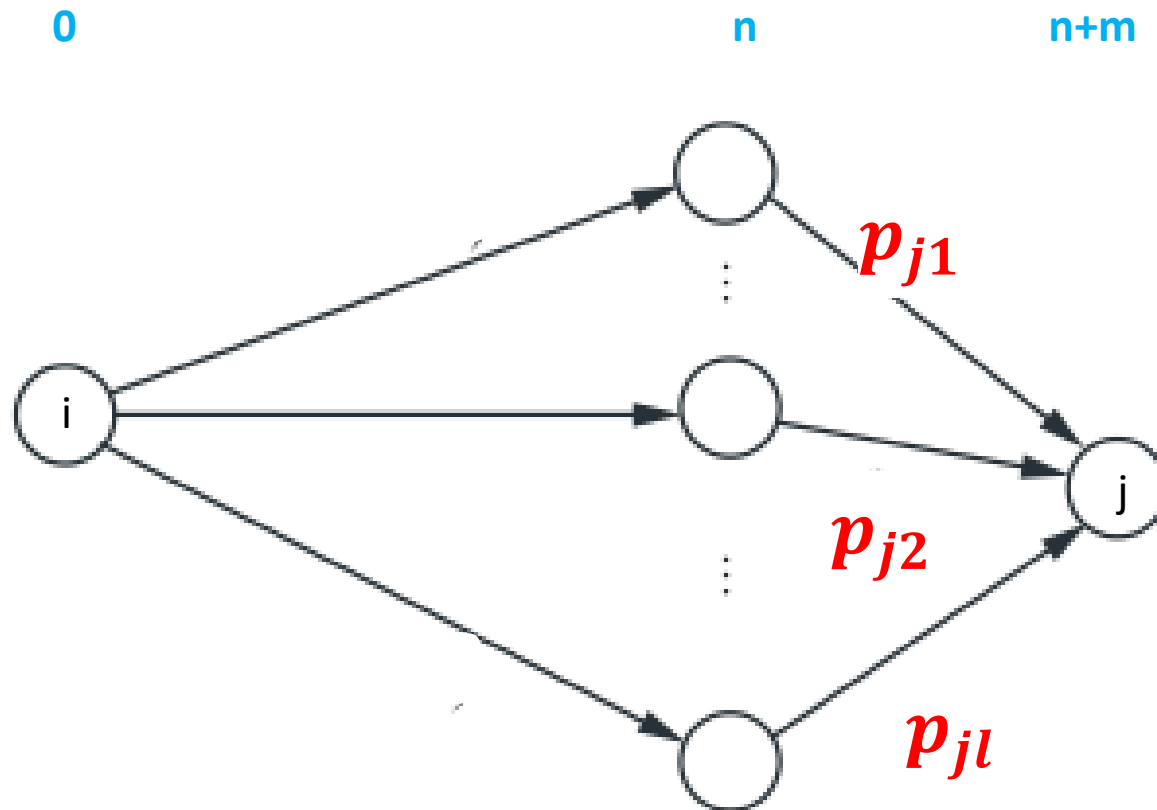
- Ataquemos agora problemas do “tipo”:
 - Qual a **probabilidade de transição entre dois estados em n observações/transições ?**
- Exemplo 1:
 - Qual a **probabilidade dos que estiveram na aula de uma segunda virem à aula na segunda seguinte**
 - Assumindo as probabilidades do nosso exemplo !
 - Tendo em conta que temos aulas segunda e quinta (TP2) ou segunda e terça (TP1).

Equações de Chapman-Kolmogorov

- Definindo a transição em n passos p_{ji}^n como a probabilidade de **um processo no estado i se encontrar no estado j após n transições** adicionais. Ou seja:
- $p_{ji}^n = P(X_{n+k} = j | X_k = i), \quad n \geq 0, i, j \geq 0$
- Obviamente $p_{ji}^1 = p_{ji}$
- As **equações de Chapman-Kolmogorov** permitem calcular estas probabilidades

$$p_{ji}^{n+m} = \sum_k p_{ki}^n p_{jk}^m \quad \forall n, m \geq 0, \forall i, j$$

Interpretação



Interpretação

- É fácil de compreender se tivermos em conta que $p_{ki}^n p_{jk}^m$ representa a probabilidade de:
 - Começando em i o processo ir para o estado j em $n + m$ transições..
 - Através de um caminho que o leva ao estado k na transição n
- Logo, somando para todos os estados intermédios k obtém-se a probabilidade de estar no estado j ao fim de $n + m$ transições

“Demonstração” Eqs. Chapman-Kolmogorov

- $p_{ji}^{n+m} = P(X_{n+m} = j | X_0 = i)$
- $= \sum_k P(X_{n+m} = j, X_n = k | X_0 = i)$
- $= \sum_k P(X_{n+m} = j | X_n = k) P(X_n = k | X_0 = i)$
- $\sum_k p_{jk}^m p_{ki}^n$

Em termos de matrizes

- Se usarmos $\mathbf{T}^{(n)}$ para representar a matriz com as probabilidades de n transições, a equação anterior transforma-se em:

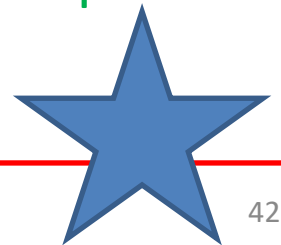
$$\mathbf{T}^{(n+m)} = \mathbf{T}^{(n)} \cdot \mathbf{T}^{(m)}$$

Em que o “.” significa multiplicação de matrizes

- Desta equação obtém-se facilmente:

$$\mathbf{T}^{(2)} = \mathbf{T}^{(1+1)} = \mathbf{T} \cdot \mathbf{T} = \mathbf{T}^2$$

- E por indução $\mathbf{T}^{(n)} = \mathbf{T}^{(n-1+1)} = \mathbf{T}^{n-1} \cdot \mathbf{T} = \mathbf{T}^n$
 - Ou seja, a matriz de transição relativa a n transições pode ser obtida multiplicando \mathbf{T} por si própria n vezes



Aplicação ao Exemplo 1

- Voltando a uma questão colocada no início da aula ...
- *Se vieram à aula esta segunda, qual a probabilidade de virem na aula de **SEGUNDA** da próxima semana ?*
- Solução:
- Temos $\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, significando “não faltar”
- Pretendemos $\mathbf{x}^{(2)}$, 0= hoje

...

- $\mathbf{x}^{(2)} = T\mathbf{x}^{(1)} = T(T\mathbf{x}^{(0)}) = T^2\mathbf{x}^{(0)}$

$$= \begin{pmatrix} 0.7 & 0.8 \\ 0.3 & 0.2 \end{pmatrix}^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.7 & 0.8 \\ 0.3 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} = \begin{pmatrix} 0.73 \\ 0.27 \end{pmatrix}$$

- Ou seja probabilidade igual a 0.73 de virem na próxima Segunda

Terminologia

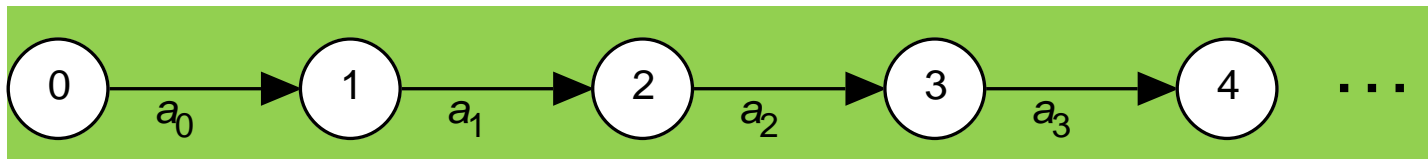
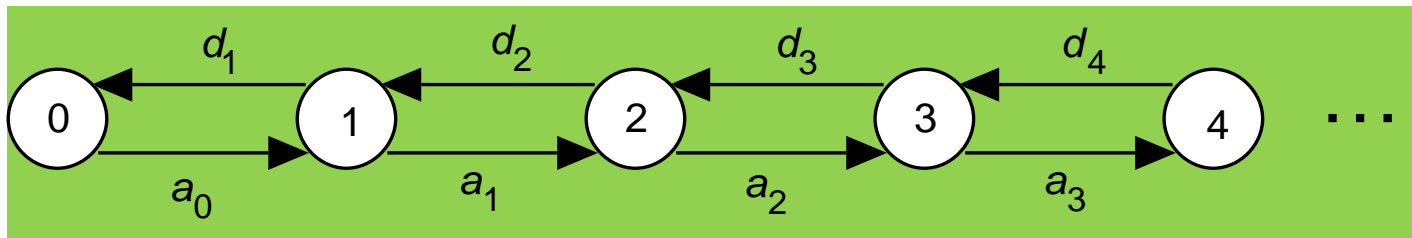
Tipos de estados

Tipos de matrizes de transição

...

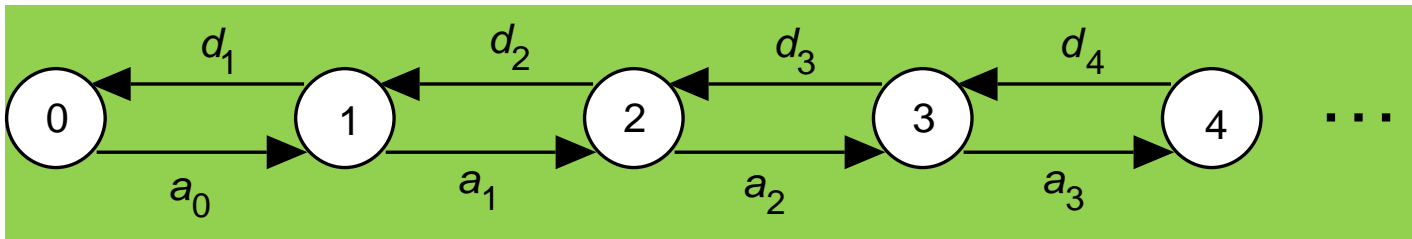
Acessibilidade de um estado

- Possibilidade de ir do estado i para o estado j (existe caminho na cadeia de i para j).



Estados comunicantes

- Dois estados comunicam se ambos são acessíveis a partir do outro.

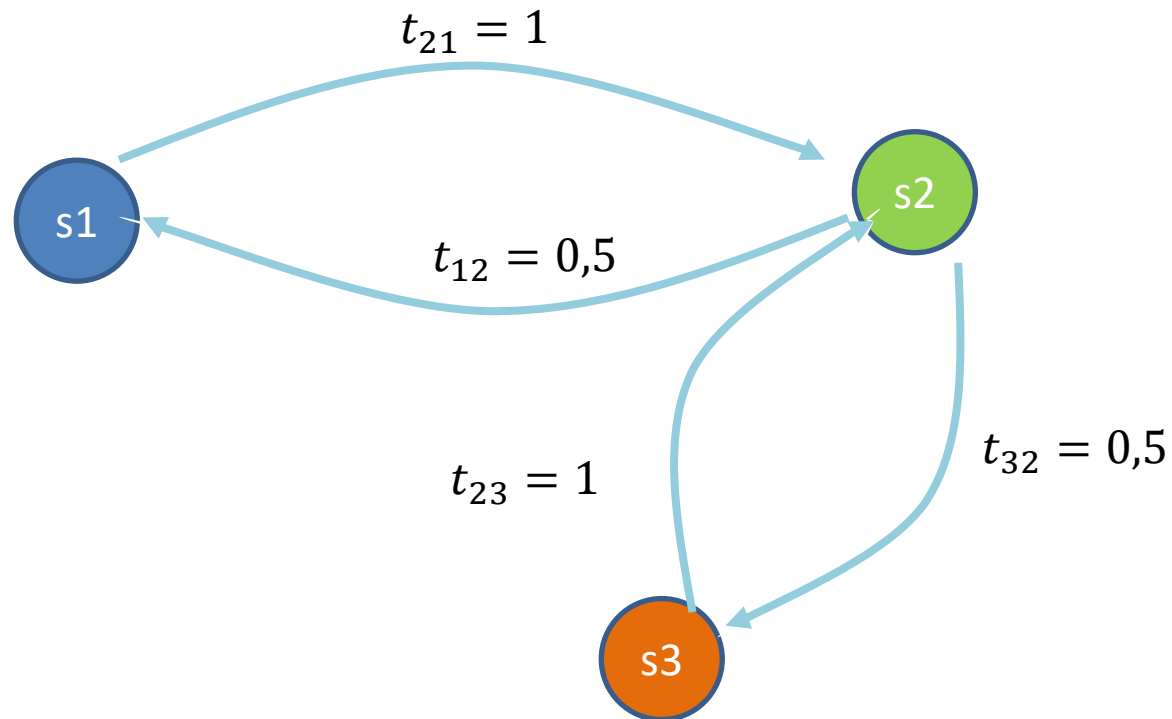


- Um sistema é não redutível (irreducible) se todos os estados comunicam
- Classe:** conjunto de estados que comunicam entre si

Estado recorrente

- Um estado s_i é um **estado recorrente** se o sistema puder sempre voltar a ele (depois de sair dele).
- De uma forma mais formal: s_i é um **estado recorrente** se, para todos os estados s_j , a existência de um inteiro r_j tal que $p_{ji}^{(r_j)} > 0$ implica que existe um inteiro r_i tal que $p_{ij}^{(r_i)} > 0$
- Um estado não recorrente é transiente

Estados recorrentes ?



- Os 3 estados são recorrentes

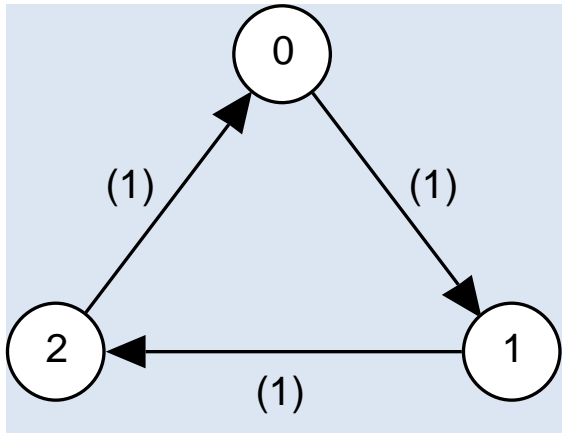
Estado transiente

- Um estado é transiente se **existe um outro estado qualquer para o qual o** processo de Markov **pode transitar, mas do qual o processo não pode retornar**
- Ou seja, se existe um estado s_j e um inteiro l tal que $p_{ji}^{(l)} \neq 0$ e $p_{ij}^{(r)} = 0$ para $r = 0, 1, 2, \dots$
- A probabilidade destes estados tende para zero quando n tende para infinito
 - Pois apenas são visitados um número finito de vezes

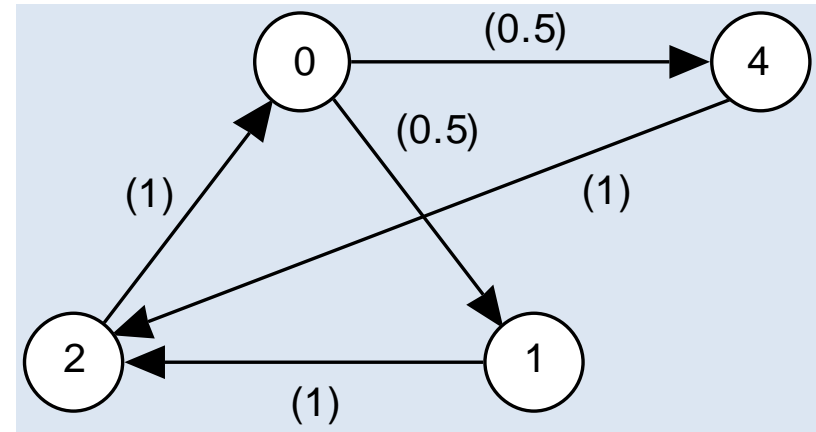
Estado periódico

- Um estado é **periódico** se apenas se pode regressar a ele após um número fixo de transições superior a 1 (ou múltiplos desse número).
- Formalizando:
 - Um estado recorrente s_i diz-se **periódico** se existe um inteiro $c > 0$ tal que $p_{ii}^{(r)}$ é igual a zero para todos os valores de r excepto $r = c, 2c, 3c, \dots$

Estado periódico



Todos estados visitados em múltiplos de 3 iterações

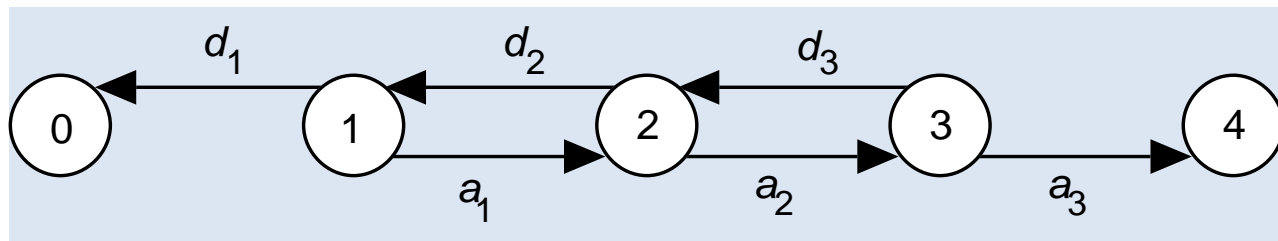


Todos estados visitados em múltiplos de 3 iterações

- Um estado não periódico é **aperiódico**
 - Como era de esperar!

Estado absorvente

- Um estado **absorvente** é um **estado do qual não é possível sair** (ou seja transitar para outro estado)
- Uma cadeia é absorvente se tiver pelo menos um estado absorvente

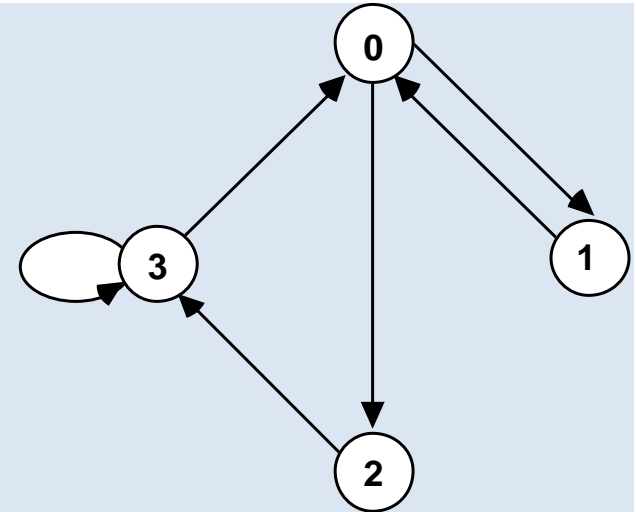


- Os estados 0 e 4 são absorventes

Aplicação dos conceitos

- Exemplo:

State	0	1	2	3
0	0	X	X	0
1	X	0	0	0
2	0	0	0	X
3	X	0	0	X



- Todos os pares de estados comunicam, formando uma única classe recorrente
 - Os estados são aperiódicos
- Em consequência o processo é aperiódico e irreduzível

Assuntos principais dados anteriormente

- Noção de processo estocástico
- Cadeias de Markov
- Propriedade de Markov
- Matriz de transição T
- Representação gráfica
- $\mathbf{T}^{(n)}$

Demos

- **Wolfram:**
 - **Finite-State, Discrete-Time Markov Chains**
 - <http://demonstrations.wolfram.com/ATwoStateDiscreteTimeMarkovChain/>
 - <http://demonstrations.wolfram.com/FiniteStateDiscreteTimeMarkovChains/>

O que acontece ao fim de muitas
transições ?

Potências de T quando $n \rightarrow \infty$

- Exemplo 2 (3 grupos de alunos):
- Vejamos o comportamento de T^n ao aumentar n ...

$$T = \begin{pmatrix} 0.333333 & 0.25 & 0. \\ 0.333333 & 0.5 & 0.5 \\ 0.333333 & 0.25 & 0.5 \end{pmatrix}$$

$$T^2 = \begin{pmatrix} 0.194444 & 0.208333 & 0.125 \\ 0.444444 & 0.458333 & 0.5 \\ 0.361111 & 0.333333 & 0.375 \end{pmatrix}$$

$$T^3 = \begin{pmatrix} 0.175926 & 0.184028 & 0.166667 \\ 0.467593 & 0.465278 & 0.479167 \\ 0.356481 & 0.350694 & 0.354167 \end{pmatrix}$$

$$T^4 = \begin{pmatrix} 0.17554 & 0.177662 & 0.175347 \\ 0.470679 & 0.469329 & 0.472222 \\ 0.353781 & 0.353009 & 0.352431 \end{pmatrix}$$

$$T^5 = \begin{pmatrix} 0.176183 & 0.176553 & 0.176505 \\ 0.470743 & 0.47039 & 0.470775 \\ 0.353074 & 0.353057 & 0.35272 \end{pmatrix}$$

$$T^6 = \begin{pmatrix} 0.176414 & 0.176448 & 0.176529 \\ 0.470636 & 0.470575 & 0.470583 \\ 0.35295 & 0.352977 & 0.352889 \end{pmatrix}$$

Continuando... (em Matlab)

% n =10

Tn =

0.1765 0.1765 0.1765

0.4706 0.4706 0.4706

0.3529 0.3529 0.3529

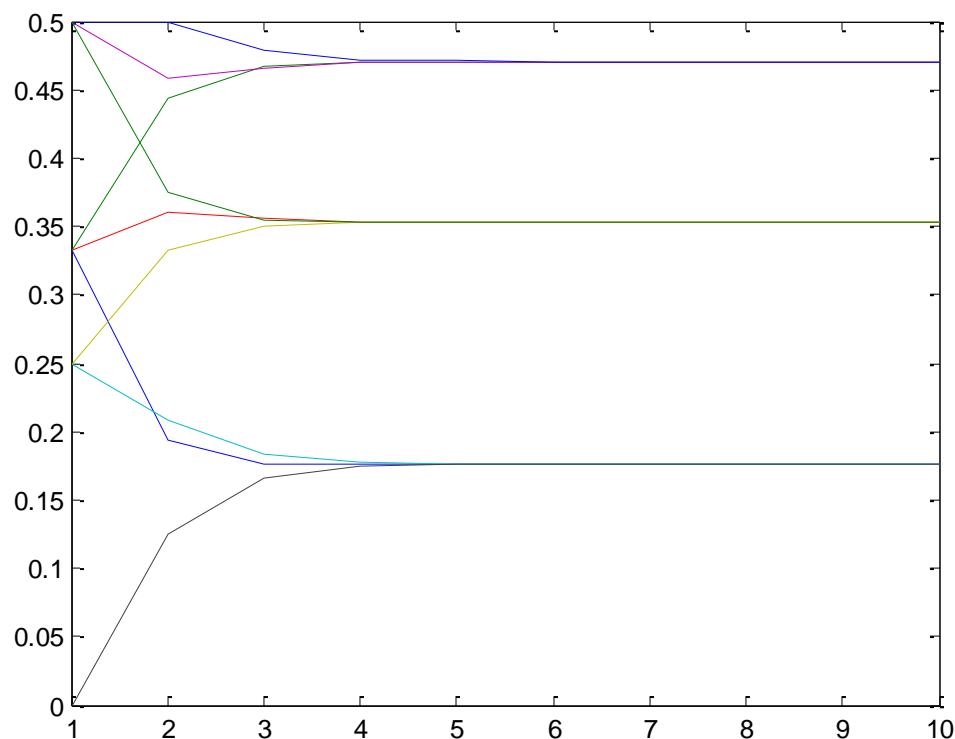
% n=100

Tn =

0.1765 0.1765 0.1765

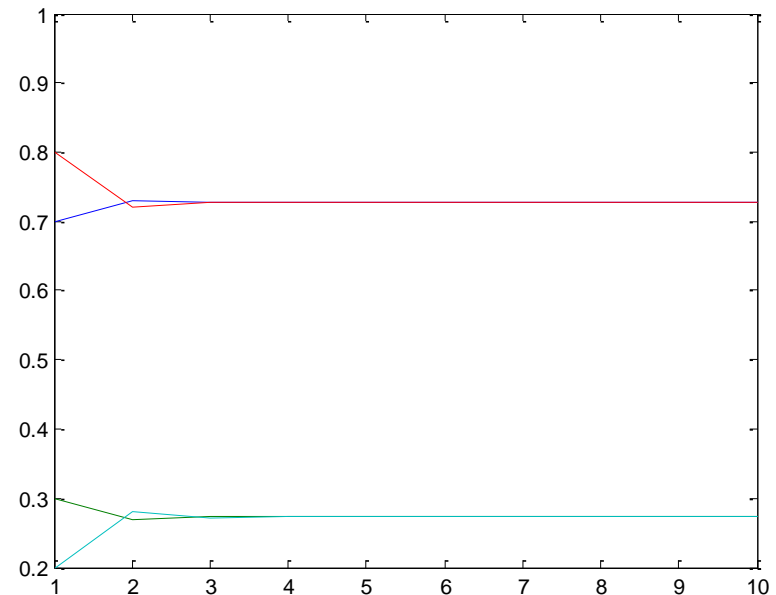
0.4706 0.4706 0.4706

0.3529 0.3529 0.3529



Exemplo 1 (faltar/não faltar)

```
T=[0.7 0.8  
    0.3 0.2]  
Tn= T;  
pij= Tn(:);  
for n=2:10  
    Tn= T*Tn;  
    pij=[ pij Tn(:)];  
    plot(pij')  
    drawnow  
end  
Tn
```



Tn =

0.7273	0.7273
0.2727	0.2727

Questões ?

- Converge ?
- Para quê ?

Equilíbrio

- As cadeias dos nossos exemplos atingem um equilíbrio.
- Quando isso acontece a probabilidade de qualquer estado torna-se constante independentemente do passo (step) e das condições iniciais
- Para analisar essa situação é necessário considerar um certo tipo de cadeias de Markov...

Matriz/ Processo regular

- A matriz de transição (ou o processo de Markov correspondente) é **regular** se alguma potência da matriz tem todos os valores não-nulos.
 - Existe uma **probabilidade de mudar de qualquer estado para qualquer estado**
- Qualquer matriz de transição sem elementos nulos é uma matriz regular.
- No entanto, uma matriz contendo elementos nulos pode ser regular.
 - Por exemplo: $T = \begin{bmatrix} 0 & 0 & 0.5 \\ 1 & 0 & 0.5 \\ 0 & 1 & 0 \end{bmatrix}$

...

- No caso de matrizes com elementos nulos, pode verificar-se se é regular substituindo os elementos não-nulos por “X” e calculando potências sucessivas
- No nosso exemplo:

$$\bullet \quad T = \begin{bmatrix} 0 & X & 0 \\ 0 & 0 & X \\ X & X & 0 \end{bmatrix}, T^2 = \begin{bmatrix} 0 & 0 & X \\ X & X & 0 \\ 0 & X & X \end{bmatrix} \dots, T^8 = \begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix}$$

Cadeia **ergódica**

- Uma cadeia de Markov diz-se ergódica se é possível efectuar transições de qualquer estado para qualquer outro estado
- Em consequência, uma **cadeia regular é também ergódica**

Cadeia ergódica

- No entanto, **nem todas as cadeias ergódicas são regulares**
 - Exemplo: se de um determinado estado se pode transitar para alguns estados apenas num número par de transições e para outros num número ímpar de transições, então todas as potências da matriz de transição terão elementos nulos

$$T = \begin{bmatrix} 0 & 0.5 & 0 \\ 1 & 0 & 1 \\ 0 & 0.5 & 0 \end{bmatrix}$$

...

- Potências pares

```
>> T^30
```

```
ans =
```

```
0.5000    0    0.5000
    0 1.0000    0
0.5000    0    0.5000
```

```
>> T^100
```

```
ans =
```

```
0.5000    0    0.5000
    0 1.0000    0
0.5000    0    0.5000
```

- Potências ímpares

```
>> T^5
```

```
ans =
```

```
    0 0.5000    0
1.0000    0 1.0000
    0 0.5000    0
```

```
>> T^51
```

```
ans =
```

```
    0 0.5000    0
1.0000    0 1.0000
    0 0.5000    0
```

$$\lim_{n \rightarrow \infty} T^n$$

- Se T é a matriz de transição de um processo de Markov **regular** então:
- $\lim_{n \rightarrow \infty} T^n$ é a matriz:

$$A = \begin{bmatrix} u_1 & u_1 & \cdots & u_1 \\ u_2 & u_2 & \cdots & u_2 \\ \cdots & \cdots & \cdots & \cdots \\ u_N & u_N & \cdots & u_N \end{bmatrix}$$

Com todas as colunas idênticas

- Cada coluna \mathbf{u} é um vector probabilidade em que todas as componentes são positivas

Vector estado estacionário (steady-state vector)

- Sendo T uma matriz de transição regular e \mathbf{u} o resultado de $\lim_{n \rightarrow \infty} T^n$ (slide anterior), demonstra-se que:
 - a) Para qualquer vector de probabilidade \mathbf{x} , $T^n \mathbf{x} \rightarrow \mathbf{u}$ quando $n \rightarrow \infty$
Sendo \mathbf{u} o vector estado estacionário (**steady-state vector**)
 - b) \mathbf{u} é o único vector de probabilidade que satisfaz a equação matricial $\mathbf{T}\mathbf{u} = \mathbf{u}$
 - c) $\lim_{n \rightarrow \infty} \left(\frac{\eta(i,n)}{n} \right) = \mathbf{u}(i)$, em que $\eta(i,n)$ é o número de visitas ao estado i em n passos (transições)

Cálculo do vector estado estacionário

- Sabemos que o vector correspondente ao estado estacionário é único.
- Usamos a equação que ele satisfaz para o calcular: $\mathbf{T}\mathbf{u} = \mathbf{u}$
- Ou, na forma matricial, $(\mathbf{T} - \mathbf{I})\mathbf{u} = 0$

Exemplo 1 (aulas)

- $\mathbf{T}\mathbf{u} = \mathbf{u}$

- $\begin{bmatrix} 0,7 & 0,8 \\ 0,3 & 0,2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

- $$\begin{cases} \frac{7}{10}u_1 + \frac{8}{10}u_2 = u_1 \\ \frac{3}{10}u_1 + \frac{2}{10}u_2 = u_2 \end{cases} = \begin{cases} \frac{-3}{10}u_1 + \frac{8}{10}u_2 = 0 \\ \frac{3}{10}u_1 + \frac{-8}{10}u_2 = 0 \end{cases}$$
$$\begin{matrix} u_1 + u_2 = 1 \end{matrix}$$

- ...

Em Matlab

Uma possível solução:

% matriz de transição

$T = [7 \ 8; 3 \ 2]/10$

% $(T-I)u$ aumentado com u_1+u_2

$M = [T - \text{eye}(2);$
 $\text{ones}(1,2)]$

%

$x = [0 \ 0 \ 1]'$

% resolver para obter u

$u = M \backslash x$

Resultado:

0.7273

0.2727

Ou seja aprox. 72 % de
probabilidade de não
faltarem



Exemplo1.m

Outra possibilidade

% matriz de transição

T=[7 8; 3 2]/10;

% resolver equação $Au = b$

aux= (T-eye(length(T)));

% usas as primeiras linhas de T + equação $x_1+x_2= 1$

A= [aux(1:end-1,:); 1 1];

b= [0 1]';

u= inv(A)*b

...

- Pode também ser resolvido usando uma matriz aumentada e a função **rref()**

`%Matlab`

`C= [M x]`

`rref(C)`

$$\left(\begin{array}{cc|c} -3/10 & 8/10 & 0 \\ 3/10 & -8/10 & 0 \\ 1 & 1 & 1 \end{array} \right) \rightsquigarrow \left(\begin{array}{cc|c} 1 & 0 & 8/11 \\ 0 & 1 & 3/11 \\ 0 & 0 & 0 \end{array} \right)$$

- Mais informação:
https://www.math.ucdavis.edu/~daddel/Math22al_S02/LABS/LAB2/lab2_w01/node9.html

Exemplo 2

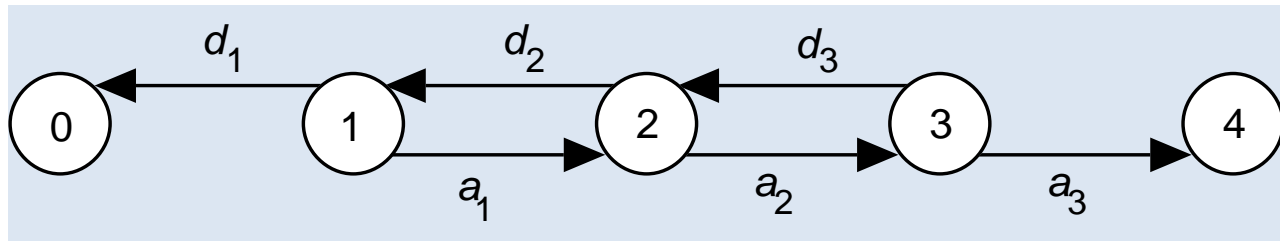
- Aplicando a última técnica ao nosso exemplo 2 (grupos) teremos

$$\left(\begin{array}{ccc|c} -2/3 & 1/4 & 0 & 0 \\ 1/3 & -1/2 & 1/2 & 0 \\ 1/3 & 1/4 & -1/2 & 0 \\ 1 & 1 & 1 & 1 \end{array} \right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 3/17 \\ 0 & 1 & 0 & 8/17 \\ 0 & 0 & 1 & 6/17 \\ 0 & 0 & 0 & 0 \end{array} \right)$$

Cadeias com estados absorventes

Estados absorventes

- Um **estado absorvente** é um estado do qual não é possível sair (ou seja transitar para outro estado)



- Os estados 0 e 4 são absorventes

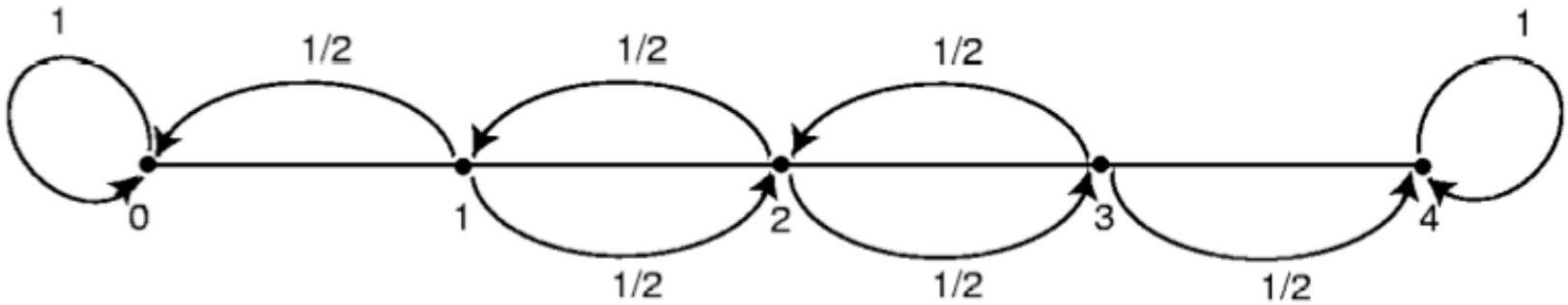
Cadeias absorventes

- Uma **cadeia** é **absorvente** se:

(1) tiver pelo menos um estado absorvente

(2) é possível ir de cada um dos estados não absorventes para pelo menos um dos estados absorventes num número finito de passos.

Exemplo simples



Demo

- Absorbing Markov Chain
 - <http://demonstrations.wolfram.com/AbsorbingMarkovChain/>

Forma canónica da matriz de transição

Forma canónica

- Se numa matriz de transição **agruparmos todos os estados absorventes** obtemos a denominada forma canónica (standard form)
- O mais usual é colocar **primeiro os não absorventes** e depois os absorventes.
- **A forma canónica** é muito útil para determinar as matrizes em situações limite de cadeias de Markov absorventes
 - Como veremos...

Forma canónica

- Rearranjar os estados da matriz T por forma a que os **estados transientes** apareçam **primeiro**

Matriz $t \times t$

$$\begin{array}{c} \text{TR.} \\ \text{ABS.} \end{array} \begin{pmatrix} \text{TR.} & \text{ABS.} \\ \hline Q & 0 \\ \hline R & I \end{pmatrix}$$

t : # estados transientes
 a : # estados absorventes

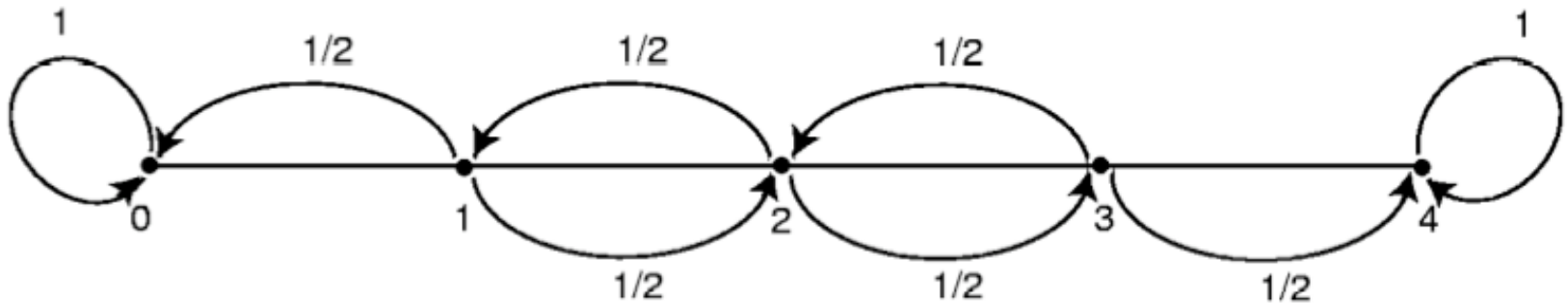
$a \times a$ matriz
identidade

Aplicação a um exemplo

- Homem a caminhar para casa de um bar
 - 4 quarteirões entre o bar e a casa
 - 5 estados no total
- Estados absorventes:
 - Esquina 4 – Casa
 - Esquina 0 – Bar
- No limite de cada quarteirão existe igual probabilidade de seguir em frente ou retroceder

Diagrama e matriz de transição

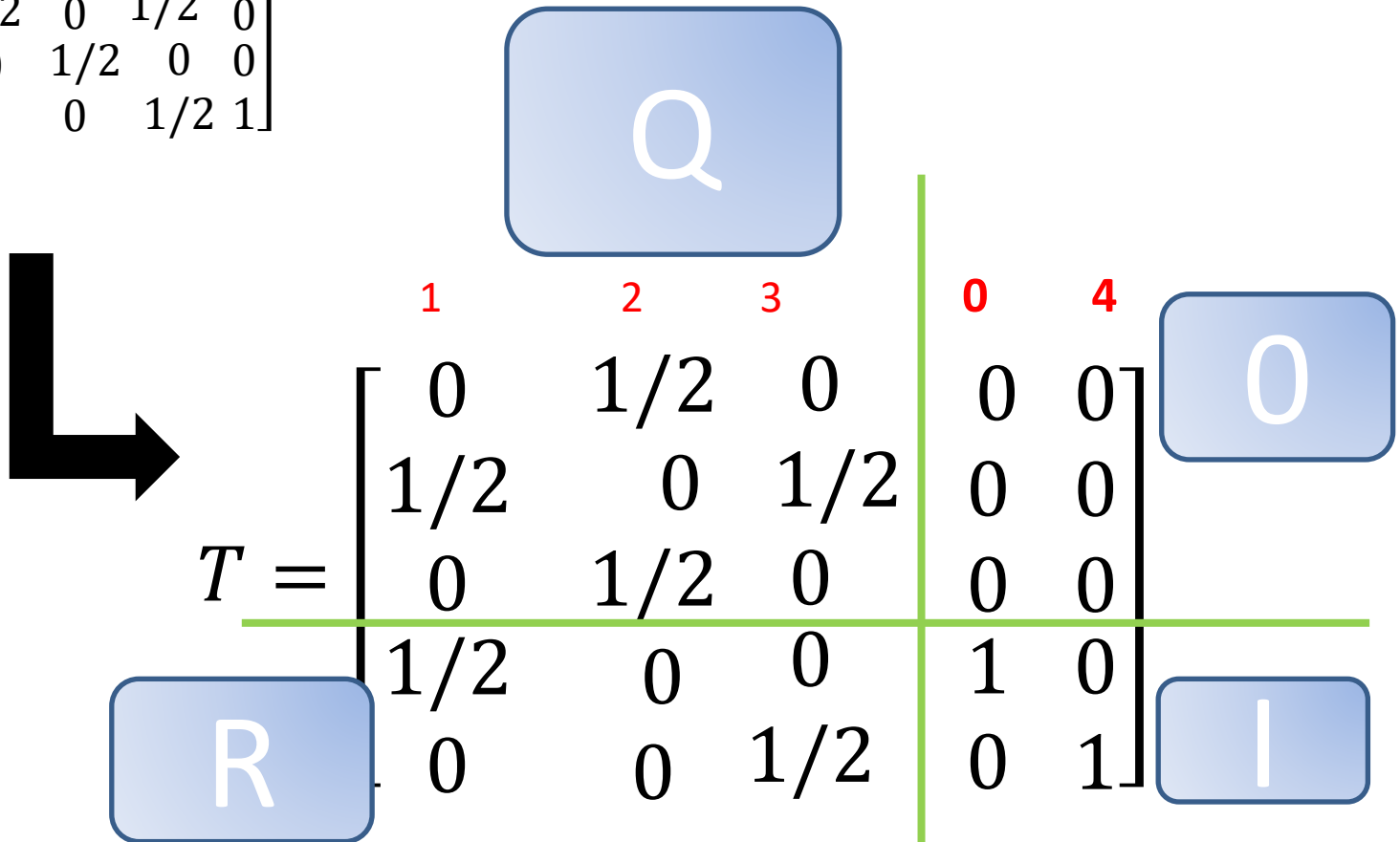
- Diagrama de transição



- $$T = \begin{bmatrix} 1 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1 \end{bmatrix}$$

Forma canónica

$$\bullet \quad T = \begin{bmatrix} 1 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1 \end{bmatrix}$$



Diferentes notações

- *Na nossa notação a estrutura é:*

$$\begin{array}{cc} Q & 0 \\ R & I \end{array}$$

- *Na notação alternativa:*

$$\begin{array}{cc} Q & R \\ 0 & I \end{array}$$

Q

- A sub-matriz Q descreve probabilidades de transição de estados não-absorventes para estados não-absorventes

Situação limite

Situação limite

- Situações limite de cadeias de Markov absorventes ?
- Como é óbvio a cadeia irá acabar por ficar indefinidamente num dos estados absorventes !
- Mas mesmo assim existem questões relevantes:
- Qual o estado absovervente mais provável quando temos vários ?
- Dado um estado inicial, qual o número esperado de transições até ocorrer absorção ?
- Dado um estado inicial, qual a probabilidade de ser absorvido por estado absorvente em particular ?

Potências de T

- Multiplicando repetidamente a matriz de transição na sua forma canónica vê-se que:
- $T^n = \begin{bmatrix} Q^n & 0 \\ X & I \end{bmatrix}$
- A expressão exacta de X não tem interesse, mas Q e Q^n são importantes

$$Q^n$$

- A matriz Q^n representa a probabilidade de permanecer em estados não-absorventes após n passos
 - Q^n tende para zero quando n aumenta
 - $Q^n \rightarrow 0$ quando $n \rightarrow \infty$

Matriz fundamental

- Multiplicando verifica-se que
- $(I - Q)(I + Q + Q^2 + \dots + Q^n) = I - Q^{n+1}$
- Fazendo $n \rightarrow \infty$ temos
- $(I - Q)(I + Q + Q^2 + \dots) = I$
- porque $Q^n \rightarrow 0$

- Isto mostra que
- $(I - Q)^{-1} = I + Q + Q^2 + Q^3 + \dots$

Matriz Fundamental

$$F = (I - Q)^{-1}$$

é a **matriz fundamental** do percurso aleatório

Interpretação de F

- Sejam $X_k(ji)$ as variáveis aleatórias definidas por:
- $$X_k(ji) = \begin{cases} 1, & \text{se estiver em } j \text{ após } k \text{ passos,} \\ & \text{partindo de } i \\ 0, & \text{caso contrário} \end{cases}$$
- A soma $X_0(ji) + X_1(ji) + \dots + X_n(ji)$ representa o número de visitas ao estado j , partindo do estado i , ao fim de n passos.
- O seu valor médio é dado por

$$E[X_0(ji) + X_1(ji) + \dots + X_n(ji)] = \sum_{k=0}^n E[X_k(ji)]$$

- Lembrar média de soma de variáveis !

Interpretação de F (continuação)

- Mas $E[X_k(ji)] = 1 \times p + 0 \times (1 - p)$ como em qualquer variável de Bernoulli
- E p designa a probabilidade de atingir o estado j após k passos, partindo de i
 - Ou seja exactamente o valor da coluna i e linha j de Q^k .
- Logo:

$$E[X_0(ji) + X_1(ji) + \cdots + X_n(ji)] = \sum_{k=0}^n Q^k(ji)$$

Interpretação de F (continuação)

- Os elementos de $I + Q + Q^2 + Q^3 + \dots + Q^n$ exprimem portanto o **número médio de visitas ao estado j partindo do estado i em n passos**
- Logo, **a matriz fundamental F** – que é o limite dessa quantidade quando $n \rightarrow \infty$ - **representa o número médio de visitas a cada estado antes da absorção**
- F_{ji} dá-nos o valor esperado para o número de vezes que um processo se encontra no estado s_j se começou no estado s_i
 - Antes de ser absorvido

Aplicando ao nosso exemplo

- $Q = \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{bmatrix}$

- $I - Q = \begin{bmatrix} 1 & -1/2 & 0 \\ -1/2 & 1 & -1/2 \\ 0 & -1/2 & 1 \end{bmatrix}$

- $F = (I - Q)^{-1} = \begin{bmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{bmatrix}$



Exemplo2.m

Tempo médio até à absorção

- O **tempo médio até à absorção** será a **soma do número médio de visitas a todos os estados transientes até à absorção**
- Ou seja a **soma da coluna i de F**

$$t_i = \sum_j F_{ji}$$

- Na forma matricial pode obter-se o vector t usando

$$t = F' \mathbf{1}$$

- Em que :
 - $\mathbf{1}$ é uma vector coluna com uns

Tempo médio até absorção

- A soma da coluna i de F representa:
 - O valor esperado do **número de vezes que a cadeia passa por um qualquer estado transiente partindo do estado inicial** i antes da absorção
 - Valor esperado do tempo necessário até absorção partindo do estado i
- O vetor t contém os tempos médios até à absorção partindo dos vários estados transientes

Aplicando ao Exemplo: Tempo até absorção

- $t = F' \cdot 1$

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

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



Exemplo2.m

Probabilidades de absorção

- As **probabilidades de absorção** b_{ji} no estado s_j se se iniciar no estado s_i podem ser obtidas através de:

$$B = R F$$

- Em que B é uma matriz $a \times t$ com entradas b_{ji}

Origem da expressão

- $B_{ji} = \sum_n \sum_k r_{jk} q^{(n)}_{ki}$
 - De i para k (transientes) e de k para j (absorvente)
 - Lembra-se de Chapman-Kolmogorov ?
- Trocando somatório:
- $B_{ji} = \sum_k \sum_n r_{jk} q^{(n)}_{ki}$
- Usando definição da matriz fundamental:
- $B_{ji} = \sum_k r_{jk} F_{ki}$
- De onde se obtém
- $B_{ji} = (R F)_{ji}$

Aplicação ao nosso exemplo

- Relembremos que temos:

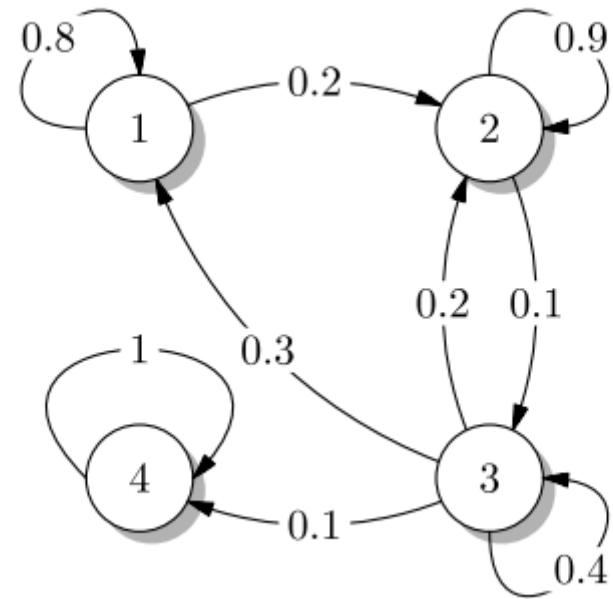
$$T = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 0 & 1 \end{bmatrix} \text{ e } F = \begin{bmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{bmatrix}$$

- E portanto $R = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 0 & 1/2 \end{bmatrix}$

- Multiplicando R e F obtemos $B = \begin{matrix} & \overset{1}{0} & \overset{2}{4} & \overset{3}{4} \end{matrix} \begin{bmatrix} 3/4 & 1/2 & 1/4 \\ 1/4 & 1/2 & 3/4 \end{bmatrix}$

Aplicação a páginas web...

- Consideremos o conjunto de páginas web da figura:
- Qual o **número médio de visitas** às páginas 1,2 e 3 quando se parte da página 1?
- Quais os **tempos médios até absorção** ?



Número médio de visitas às páginas 1,2 e 3 quando se parte da página 1?

- São dados directamente pela matriz F
- Em Matlab ...

% OBTENHA T na forma canónica

% Obter Q

submatriz de 3x3

% calcular F



Exemplo3.m

Matlab

```
estados=[1 2 3 4];
```

```
% matriz T
```

```
Tcan=zeros(4);
```

```
Tcan(1,1)=0.8; Tcan(2,1)=0.2;
```

```
Tcan(2,2)=0.9; Tcan(3,2)=0.1;
```

```
Tcan(1,3)=0.3; Tcan(2,3)=0.2; Tcan(3,3)=0.4; Tcan(4,3)=0.1;
```

```
Tcan(4,4)=1;
```

```
%% Q
```

```
Q=Tcan(1:3,1:3)
```

```
%% F
```

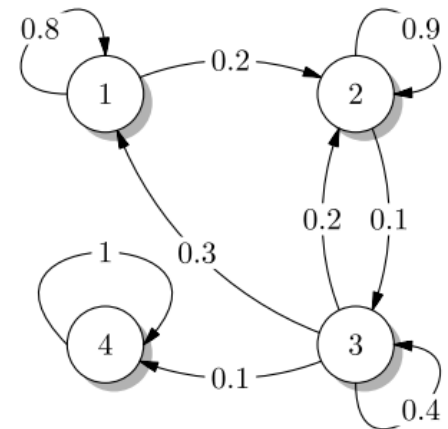
```
aux= eye(size(Q)) - Q
```

```
F=inv(aux)
```

F =		
20.0000	15.0000	15.0000
60.0000	60.0000	50.0000
10.0000	10.0000	10.0000

Resposta à questão

- Os valores que nos interessam são os da coluna 1 (correspondentes a começar na página 1)
- Quando se parte da página 1, o número médio de visitas aos estados 1, 2 e 3 antes de ocorrer absorção será 20, 60 e 10, respectivamente
 - A página 2 receberá mais visitas
 - A página 3, com ligação directa ao estado absorvente terá muito menos visitas



Tempos médios até absorção ?

- Basta obter o vector t correspondente à soma das colunas de F

%Em Matlab...

```
t=F' * ones(3,1) % ou sum(F)
```

t =

90.0000

85.0000

75.0000



Exemplo3.m

Matriz B ?

- Neste exemplo não faz sentido pedir B pois só temos um estado absorvente
- Mas se fizermos $B = R F$ obtemos um vector de 1x3 só com uns
 - Confirmando o esperado

PageRank

Serve também de revisão dos conceitos essenciais das aulas anteriores sobre Cadeias de Markov

Motivação

E um pouco de História ..

Como descobrir informação na web?

- Primeiras tentativas:

- Listas criadas por humanos

- Directórios da Web**

- Yahoo, DMOZ, LookSmart

- Segunda geração: **Web Search**

- **Information Retrieval :**

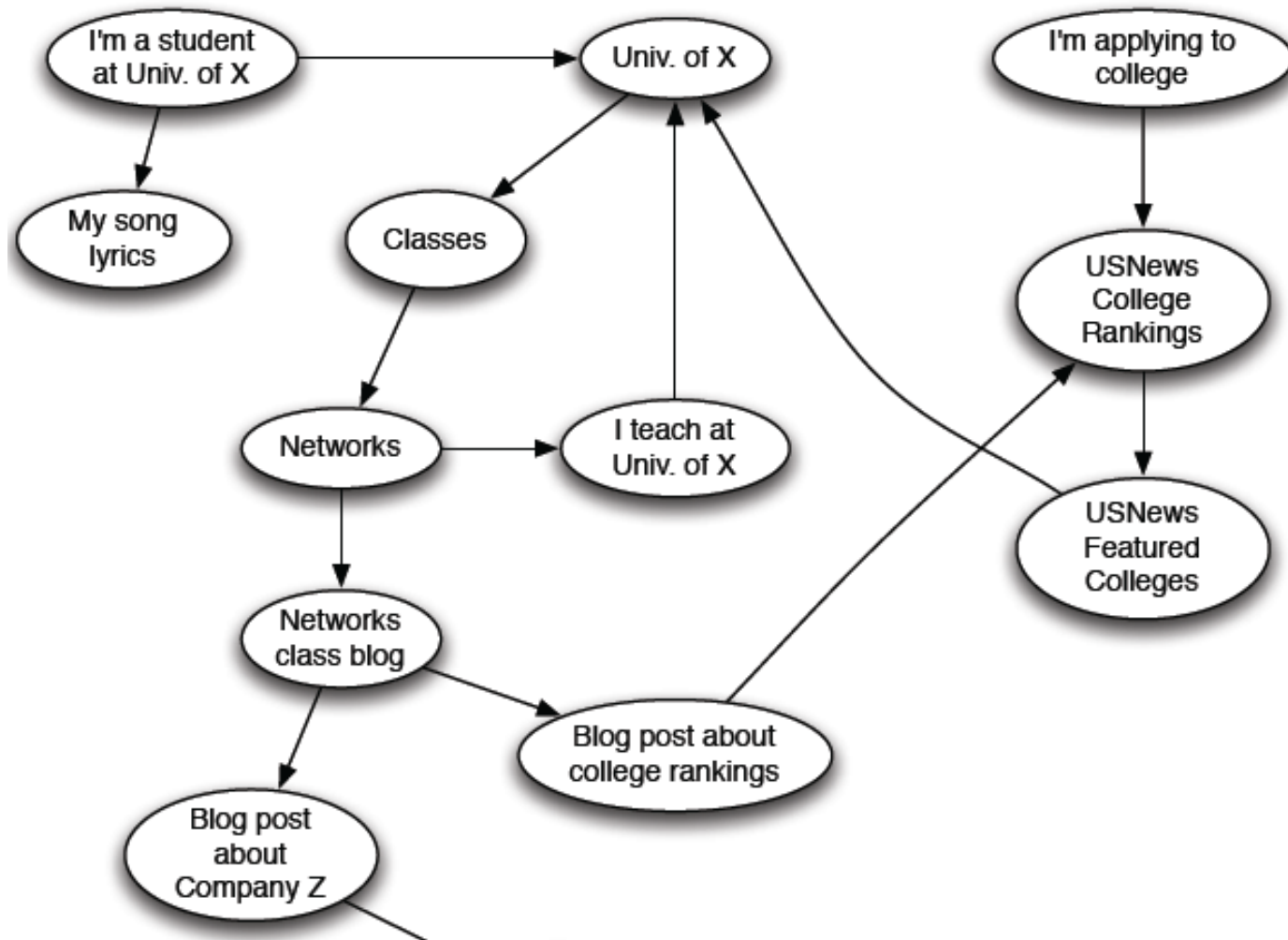
- Procura de documentos relevantes num conjunto pequeno e de confiança

- Artigos de jornais, Patentes, etc.

- **MAS:** A Web é gigantesca, cheia de documentos sobre os quais não temos garantias de ser de confiança, cheia de SPAM, etc.

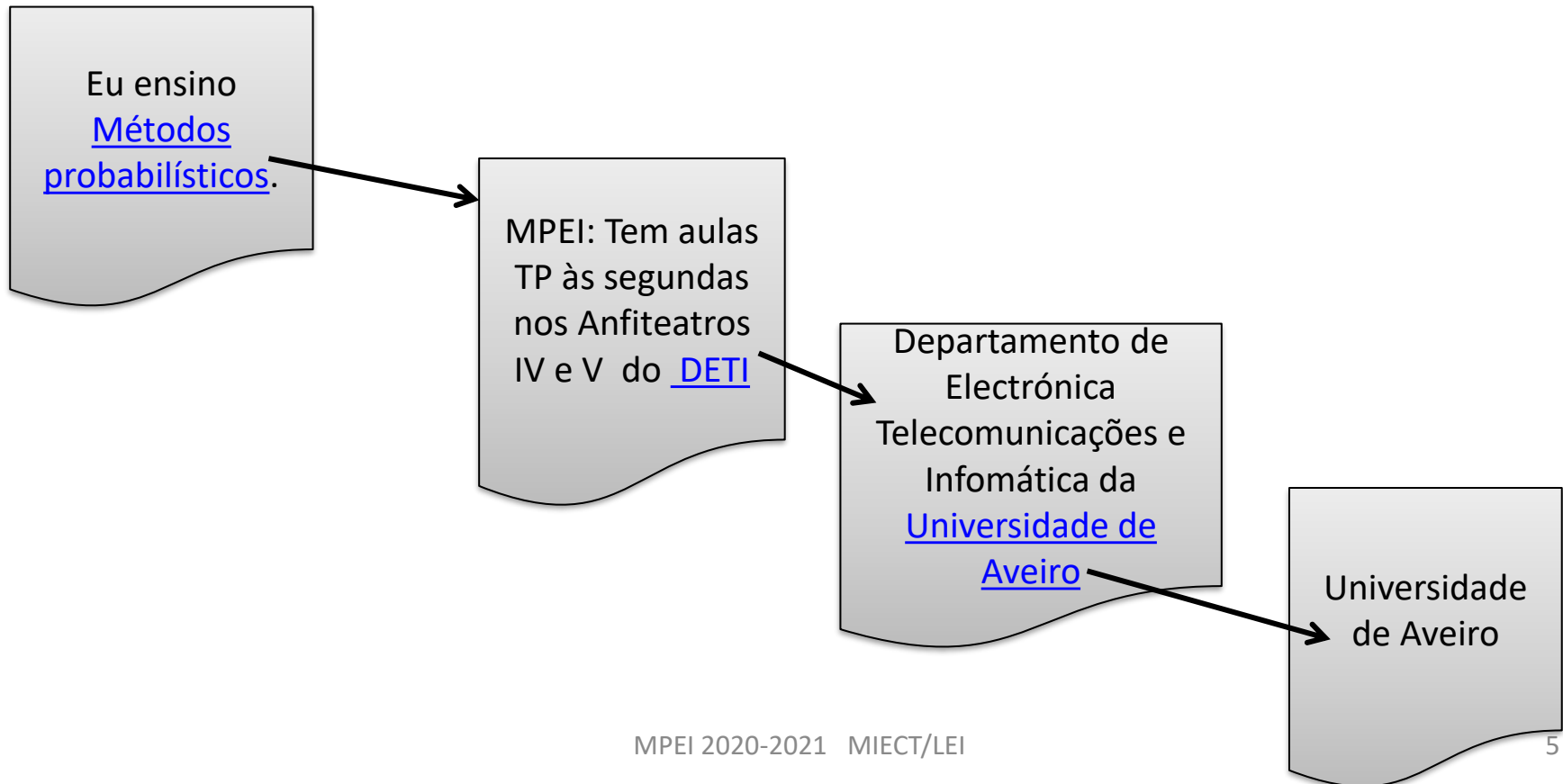


A Web como um grafo

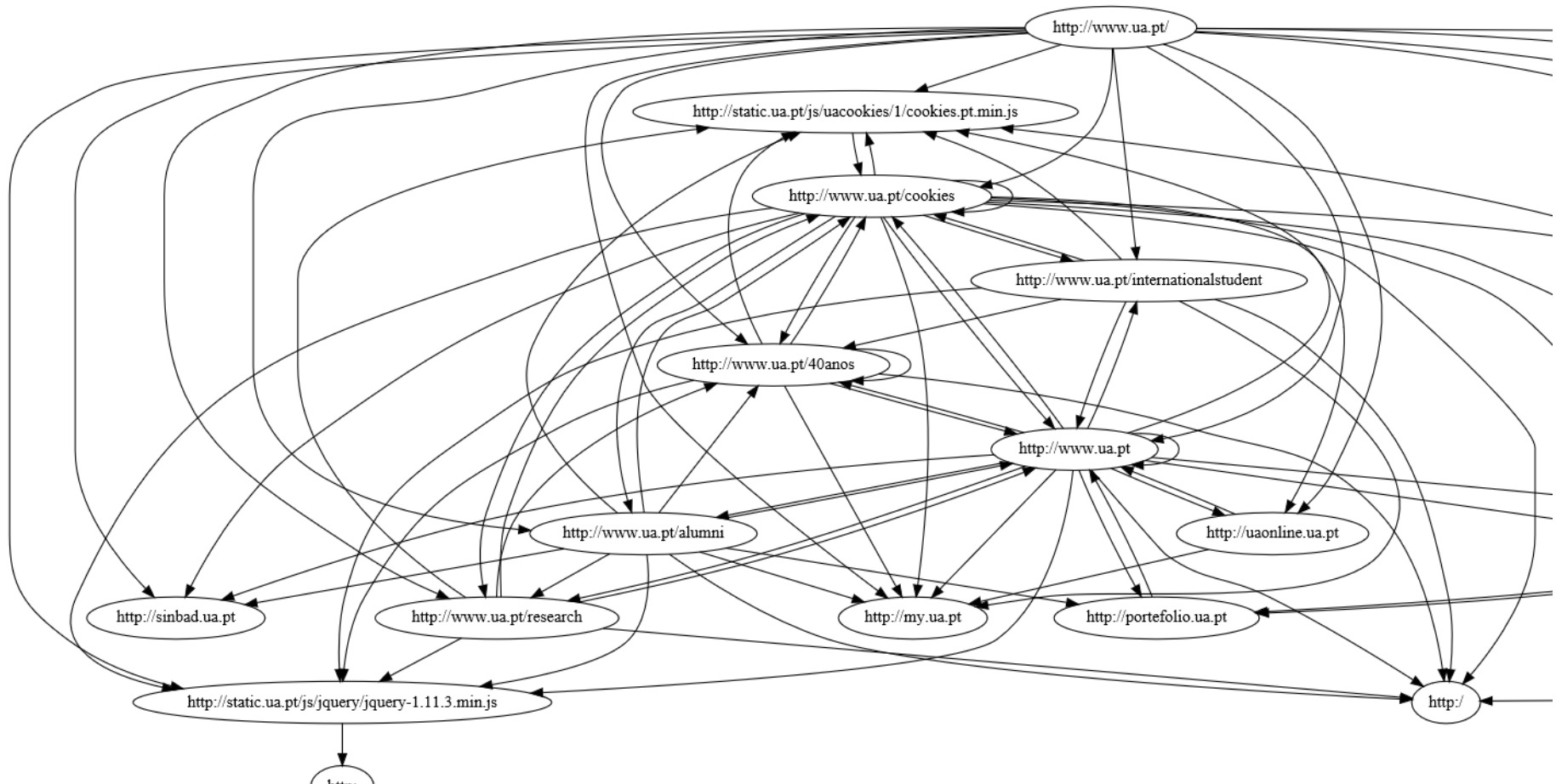


A Web como um Grafo

- Nós /nodos/vértices : Páginas Web
- Ligações/arcos: Hyperlinks



Uma pequena parte das páginas da UA



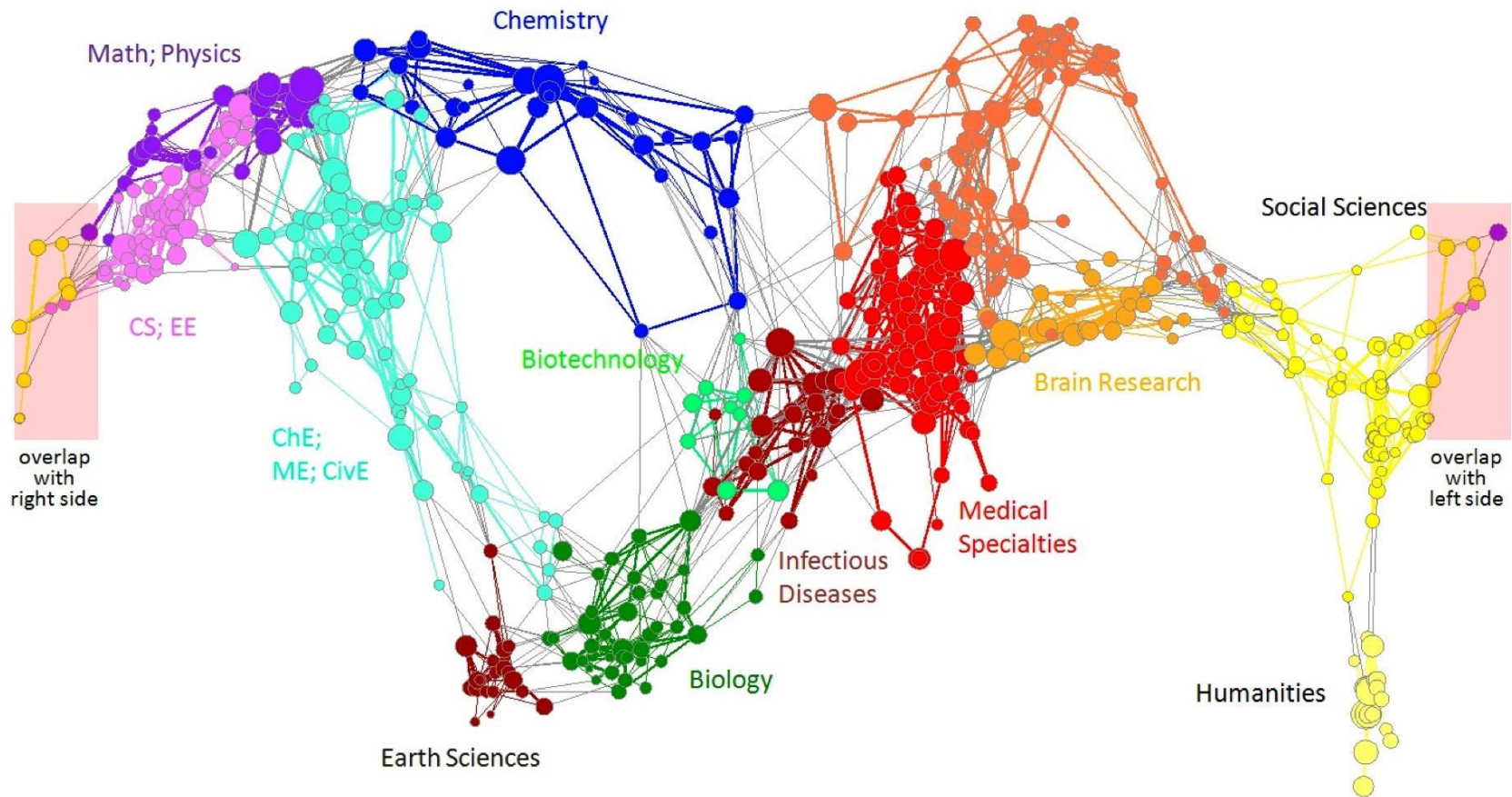
Outros grafos na web: Social Networks



Facebook social graph

4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]

Outros grafos na web: Redes de informação



Citation networks and Maps of science
[Börner et al., 2012]

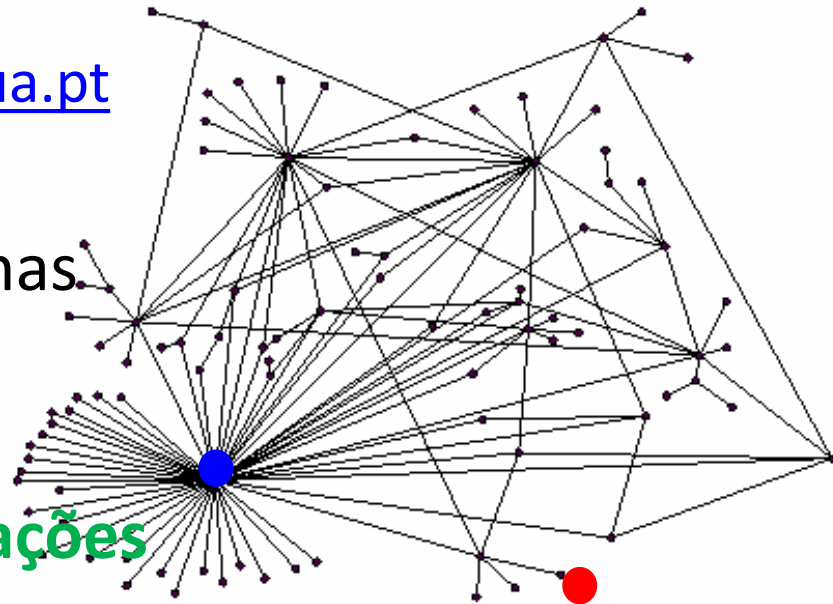
Estrutura do grafo

- As páginas da web Não são todas igualmente “importantes”

www.joe-schmoe.com vs. www.ua.pt

- Existe um grande diversidade nas ligações

- Ideia: Usar a estrutura das ligações para saber quais as páginas “importantes”



Os primeiros motores de procura

- Baseavam-se em percorrer (crawl) a web e **listar os termos** (palavras ou outras sequências de caracteres excluindo os espaços) de cada página, num **índice invertido**



SAPO

- Um índice invertido/inversion (**inverted index**) é uma estrutura de dados que torna simples, **dado um termo, descobrir** (apontar para) **todas as páginas em que o termo ocorre**.
 - Assunto da área de Information Retrieval

Ataques de spam

- Rapidamente estes primeiros [motores de procura](#) foram atacados.



SAPO



- Sendo **sensíveis às palavras nas páginas**, facilmente os detentores de páginas com menos escrúpulos podiam inflacionar a importância das suas páginas:
 - Adicionando muitas cópias de uma ou várias palavras ao conteúdo da página e tornando essa parte invisível quando mostrada num browser
 - Usando o motor de procura para saber a página mais importante segundo o seu algoritmo e copiando o seu conteúdo para as suas páginas (mantendo esta parte invisível no browser)

Contribuição da Google

- Não calcular a relevância das páginas apenas com base nos termos que contém mas usar também informação sobre as ligações a essa página
 - Desenvolvendo e patenteando o Pagerank
 - Que começou como um projecto de investigação

Ideia base

Random surfers / Passeios aleatórios

- Simular onde **passeios aleatórios** (de surfistas aleatórios / random surfers) pelas páginas, começando numa página aleatória, tendem a passar mais se se **escolherem aleatoriamente os links de saída de uma página** (em que se encontram)
 - E permitindo que o processo se repita muitas vezes
- As **páginas visitadas por muitos passeios (ou surfers) serão “mais importantes”** do que páginas raramente visitadas.
- O Google dá preferência a páginas mais importante ao decidir quais as páginas a mostrar primeiro em resposta a um *query*
 - Mas obviamente as páginas têm de conter os termos ...

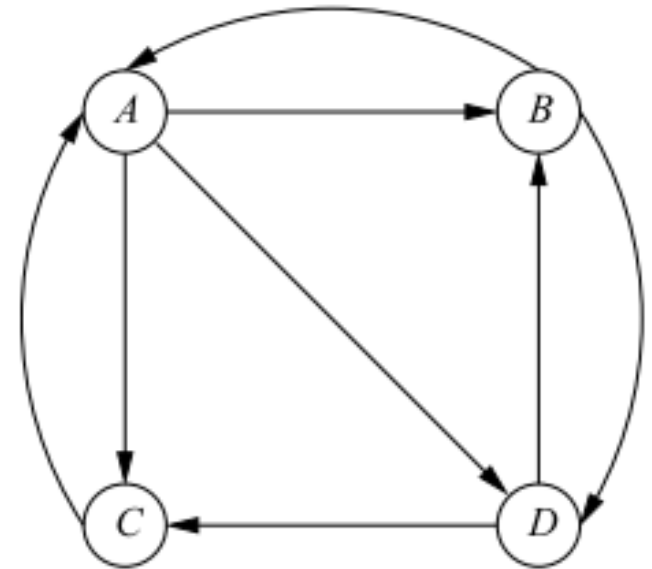
Versão base 'ideal'

- Consideremos a web como um **grafo orientado** (*directed*),
- em que as **páginas são os nodos** (ou vértices)
- e existe um arco (ou **ligação**) da página P_1 para a página P_2 se existe um ou mais links de P_1 para P_2

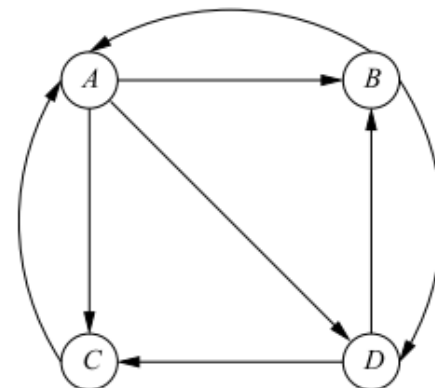
- Exemplo:

- Muito pequena rede: apenas 4 páginas

- A página A tem links para as outras 3
- A página B tem ligações apenas para a A e a D;
- A página C tem apenas um link, para a A
- E a página D tem links apenas para B e C



...



- Suponhamos que o surfista aleatório começa na página A:
- Existem links para B, C e D, logo o surfista estará de seguida numa dessas 3 páginas, com probabilidade $1/3$ [1 a dividir pelos links de saída]
 - E probabilidade zero de estar em A
- O surfista aleatório B terá, no próximo passo, probabilidade $1/2$ de estar em A, $1/2$ de estar em D e 0 de estar em C

Questão

- Estes passeios permitem mesmo aproximar a noção intuitiva de “importância” de uma página ?
- Possíveis Justificações:
 - Os utilizadores da web “votam com os seus pés”. Tendem a colocar links para páginas que consideram boas ou com informação útil
 - E não ligam a páginas de má qualidade ou inúteis
 - O comportamento dos *random surfers* indica quais as páginas que utilizadores da web visitarão com maior probabilidade.
 - Os utilizadores visitam mais páginas úteis do que não úteis.
- Independentemente das justificações anteriores, este método provou na prática que é capaz de atribuir uma medida de “importância” que permite um bom desempenho em procuras na web.
- **Veremos de seguida como funciona ...**

Definição de pagerank

- O PageRank é uma função/algoritmo que atribui um número real a cada página da Web
 - (ou a porção dela que foi processada e as ligações obtidas)
 - Designamos esse número por pagerank
 - Quanto maior é o valor mais “importante” é a página.
- Baseia-se na ideia dos random surfers
- Não existe propriamente um algoritmo fixo, havendo variações do algoritmo
 - Que podem dar valores diferentes de pagerank

pagerank

- O pagerank (r) de uma página P_j é, por **definição**:

$$r(P_j) = \sum_i \frac{r(P_i)}{d_i}$$

- sendo:
 - i o índice das **páginas que apontam para P_j**
 - d_i o número de páginas para as quais P_i aponta
ou seja, número de links de saída

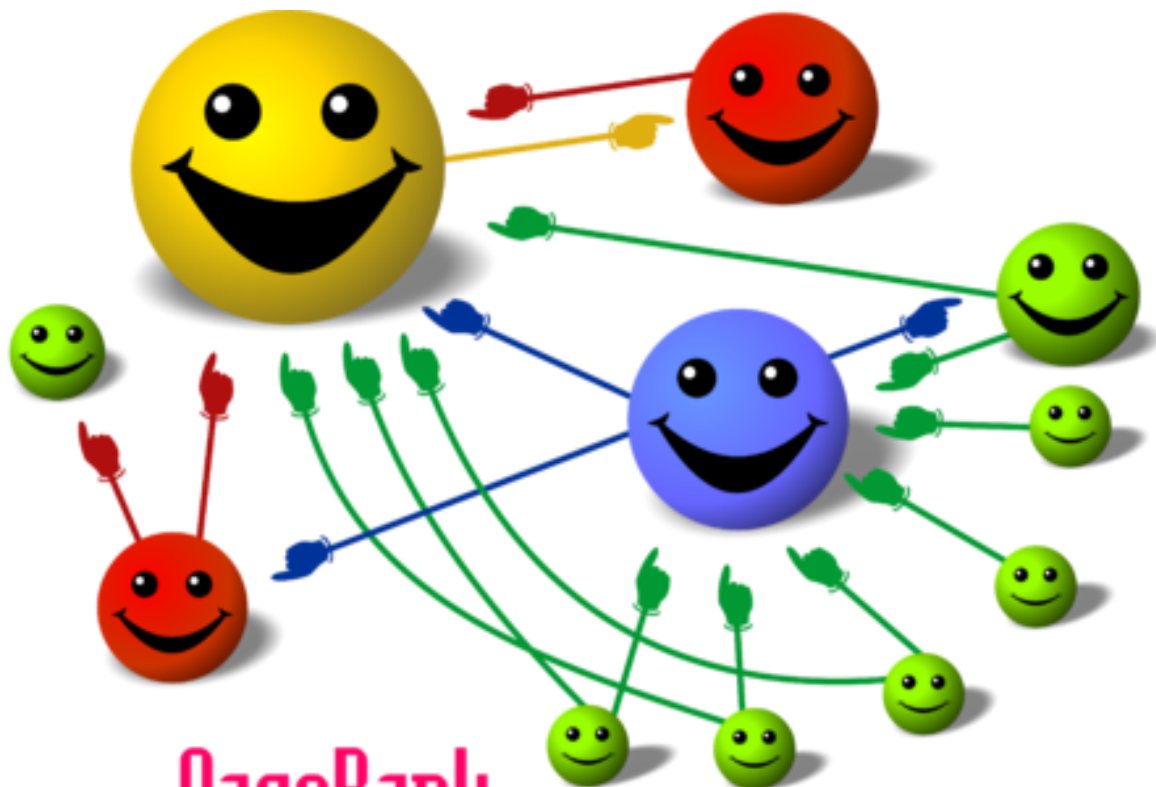
Calculo do pagerank

- O pagerank de uma página **depende do pagerank das páginas que têm links para ela**
 - identificadas pelo índice i

- O que sugere um cálculo iterativo

$$r_{k+1}(P_j) = \sum_i \frac{r_k(P_i)}{d_i}$$

- A **condição inicial** é $r_0(P_i) = 1/n$, com n igual ao número de páginas



PageRank

- Uma simplificação do sistema do PageRank,
- Cada bola representa uma página e o tamanho de cada uma a sua importância (PageRank).
- Quanto maior a bola, mais valor tem seu voto:
- Repare que a bola superior vermelha é grande mesmo recebendo só um voto, pois o voto que ela recebe, da bola maior amarela, tem mais valor

DE: <https://pt.wikipedia.org/wiki/PageRank>.

Forma matricial

- Definindo a **matriz de hyperlinks** H como

- $$H_{ji} = \begin{cases} \frac{1}{d_i} & , \text{se existir link de } i \text{ para } j \\ 0 & , \text{caso contrário} \end{cases}$$

- Teremos $r^{(k+1)} = H r^{(k)}$

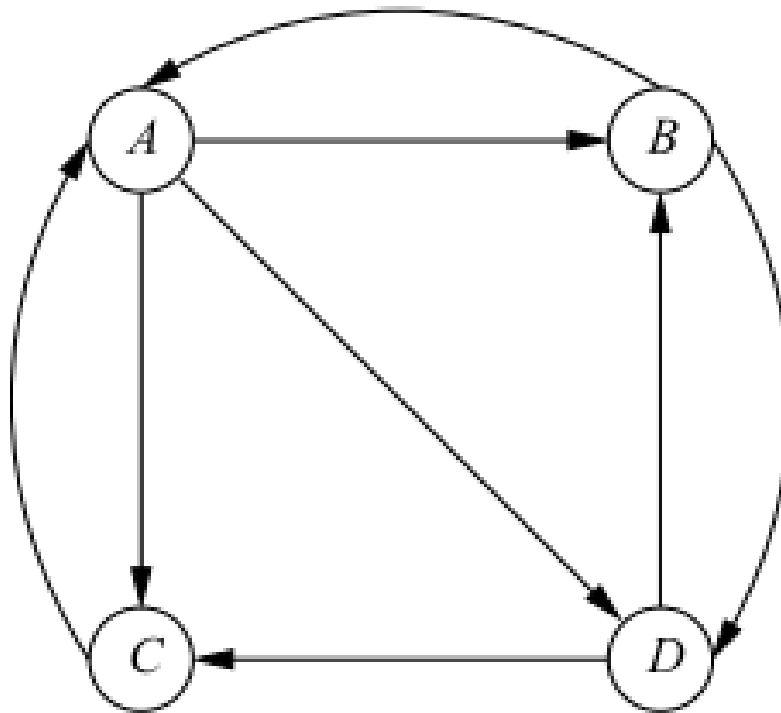
– Sendo $r^{(k)}$ o vector com *pageranks* na iteração k

Forma matricial

- A **matriz H** pode ser interpretada como **contendo as probabilidades de transição entre páginas** (os estados).
- Em consequência pode aplicar-se o que aprendemos nas aulas anteriores e calcular probabilidades após múltiplas transições, estudar o comportamento quando o número de transições (iterações) tende para infinito, etc .

Matriz para o nosso exemplo ?

- Qual será então a matriz H para a nossa mini web ?



Solução

$$\begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

- Acertaram ?

Limite

- Sabemos que a distribuição dos pageranks atingirá um **estado estacionário**, em que $r = Hr$
 - Pelo menos em certas condições:
 - O grafo ser fortemente ligado, sendo possível ir de qualquer página para qualquer página
 - Não existirem becos sem saída (*dead ends*): páginas que não têm links de saída
- O limite é atingido quando multiplicando os pageranks por H mais uma vez a distribuição de pageranks não se altera

Aplicando ao nosso exemplo

- Aplicando $r^{(k+1)} = H r^{(k)}$ sucessivamente
– e iniciando com $1/n$

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix}, \begin{bmatrix} 11/32 \\ 7/32 \\ 7/32 \\ 7/32 \end{bmatrix}, \dots, \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

Comentário

- Esta diferença em probabilidade é pequena
- Mas na web real, com biliões de páginas the grande variedade de importância, a verdadeira probabilidade de uma página como www.amazon.com é ordens de magnitude superior à probabilidade de outras páginas, como uma página pessoal

Questões

- É mesmo assim tão simples ?
- Converge sempre?
- Converge para o que queremos?
- Os resultados são razoáveis?

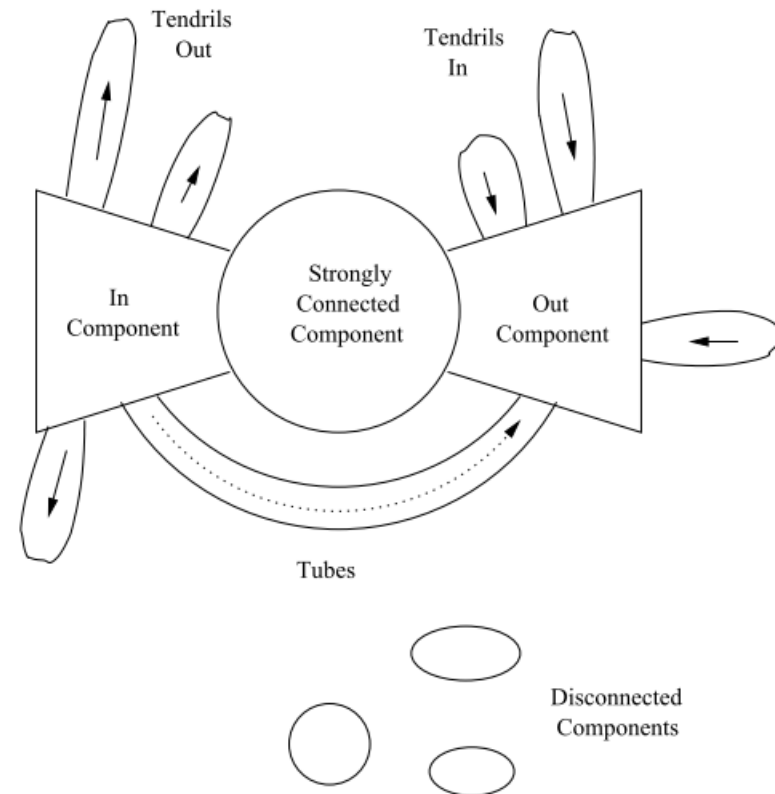
A realidade é sempre mais
complicada ...

Estrutura da web

- Será a **web tão fortemente ligada** como o nosso exemplo ?
- Seria bom que fosse...
- Mas, na prática, não é
 - Pelo menos não na sua totalidade

Estrutura da web

- Um estudo antigo da web revelou a estrutura à direita
- Existe uma parte fortemente ligada (o SCC)
- Mas também muitas páginas com:
 - Ligações ao SCC mas às quais não é possível chegar a partir do SCC (in-component)
 - Ligações a partir do SCC mas sem forma de chegar ao SCC (out-component)
 - Ligações do in-component mas incapazes de aceder a esse componente
 - Etc ..



Isto traz problemas

Temos **dois tipos de problemas** que têm de ser resolvidos

1. Becos sem saída (*dead ends*)

2. Grupos de páginas que têm links de saída mas apenas para esse grupo, impedindo a ida para outras páginas

— Estas estruturas são chamadas de *spider traps*

- Porquê ?

A spider is a program run by a [search engine](#) to build a summary of a website's content (*content index*). Spiders create a text-based summary of content and an address ([URL](#)) for each webpage.

Problemas (continuação)

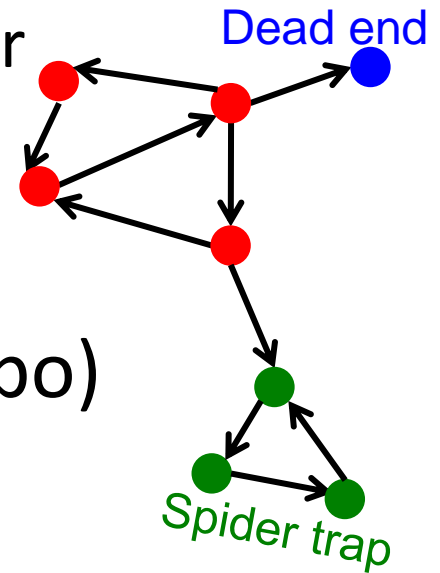
1. **Dead ends** (sem links de saída)

- Passeio aleatório não tem para onde ir

2. **Spider traps:**

(todos os links de saída para o grupo)

- O passeio aleatório fica “preso” na armadilha (trap)



- Qual o efeito nos pageranks ?

Efeito de *dead ends* e *spider traps* no pagerank

E como resolver.

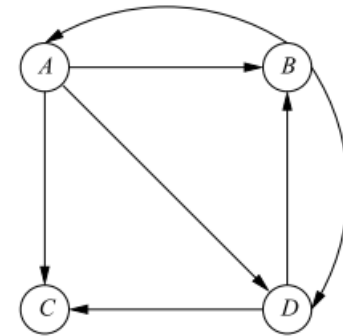
Efeitos dos *Dead ends*

- Neste caso as colunas correspondentes ficam com zeros e a sua soma é zero
- Em consequência a matriz de transição deixa de ser estocástica
- O que implica ?

Dead Ends – Exemplo

- Removendo a ligação de C para A, C passa a ser um *dead end*.

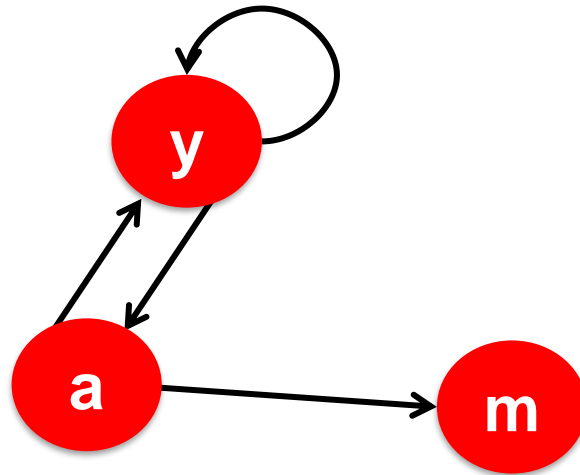
$$H = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$



- Multiplicando várias vezes H pelo estado inicial temos:

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 3/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 5/48 \\ 7/48 \\ 7/48 \\ 7/48 \end{bmatrix}, \begin{bmatrix} 21/288 \\ 31/288 \\ 31/288 \\ 31/288 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Dead Ends – Exemplo 2



	y	a	m
y	$\frac{1}{2}$	$\frac{1}{2}$	0
a	$\frac{1}{2}$	0	0
m	0	$\frac{1}{2}$	0

- Exemplo:**

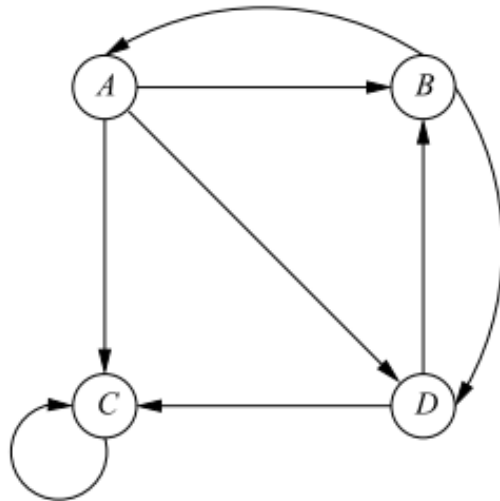
$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{matrix}$$

Iteração 0, 1, 2, ...

O PageRank “desaparece” pois a matriz não é estocástica.

Spider Traps – Exemplo 1

- Consideremos a seguinte rede e matriz

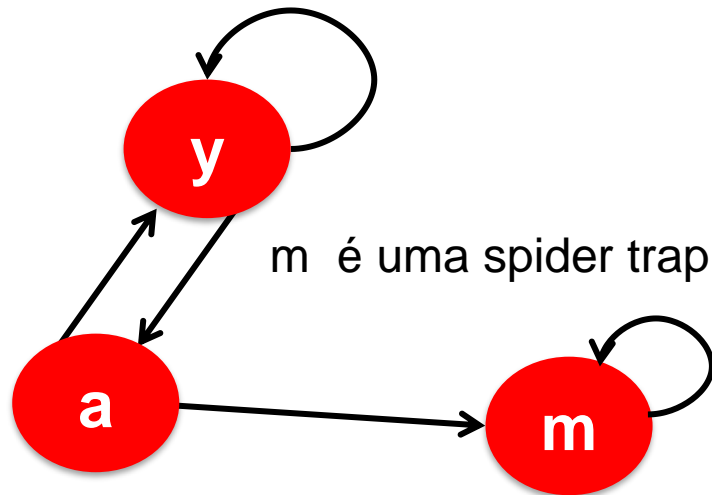


$$\begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

- Teremos para o vector estado

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 3/24 \\ 5/24 \\ 11/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 5/48 \\ 7/48 \\ 29/48 \\ 7/48 \end{bmatrix}, \begin{bmatrix} 21/288 \\ 31/288 \\ 205/288 \\ 31/288 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Spider Traps – Exemplo 2



	y	a	m
y	$\frac{1}{2}$	$\frac{1}{2}$	0
a	$\frac{1}{2}$	0	0
m	0	$\frac{1}{2}$	1

• Exemplo:

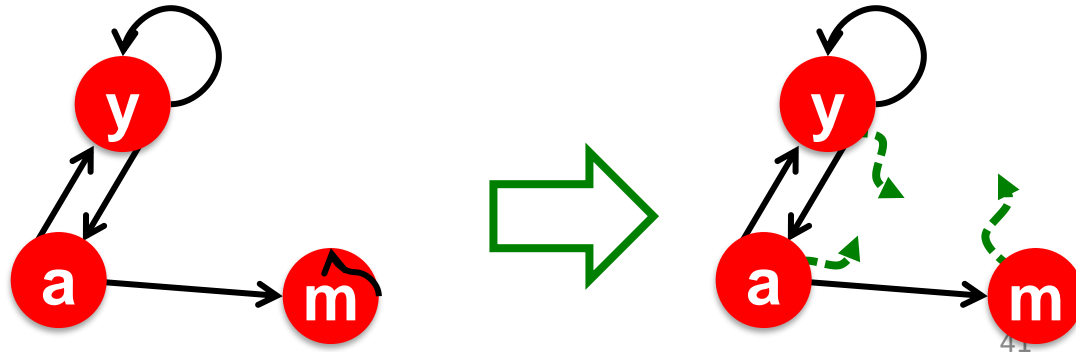
$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{pmatrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{pmatrix}$$

Iteração 0, 1, 2, ...

O PageRank é todo “apanhado” pelo nó m.

Solução para *spider traps*

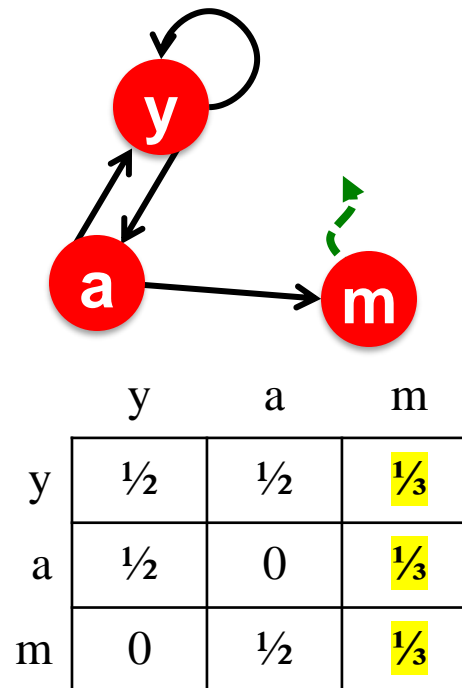
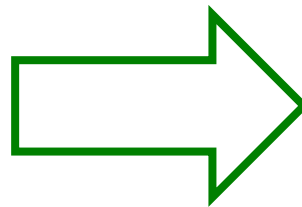
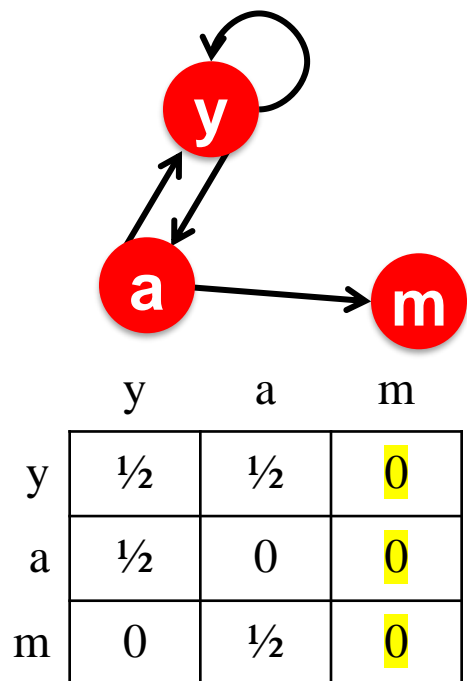
- Em cada passo, **o surfista aleatório tem duas opções**
 - Com probabilidade β :
 - **Seguir um link** aleatoriamente
 - Com probabilidade $1-\beta$,
 - **Saltar aleatoriamente para uma página qualquer**



- **O surfista teletransporta-se para fora da *spider trap* ao fim de alguns passos**
- Valores usuais para β : intervalo 0.8 to 0.9

Solução para *dead ends*

- Teletransportar (teleport) sempre
- **Implica ajustar a matriz** por forma a:
 - seguir um link com probabilidade $1/n$



Os teletransportes resolvem os dead-ends ?

- **SIM**
- **Deixa de haver dead-ends**
 - Existe sempre para onde ir
 - Matriz volta a ser estocástica

Solução: *Random Teleports* (teletransportes aleatórios)

- A solução da Google resolve a possibilidade de haver spider traps
- Em cada passo, o surfista aleatório tem duas opções
 - Com probabilidade β :
 - Seguir um link aleatoriamente
 - Com probabilidade $1-\beta$:
 - Saltar aleatoriamente para uma página qualquer
- PageRank equation [Brin-Page, 98]

$$r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N}$$

A Matriz da Google

- **Matriz da Google:**

$$A = \beta H + (1 - \beta) \left[\frac{1}{N} \right]_{N \times N}$$

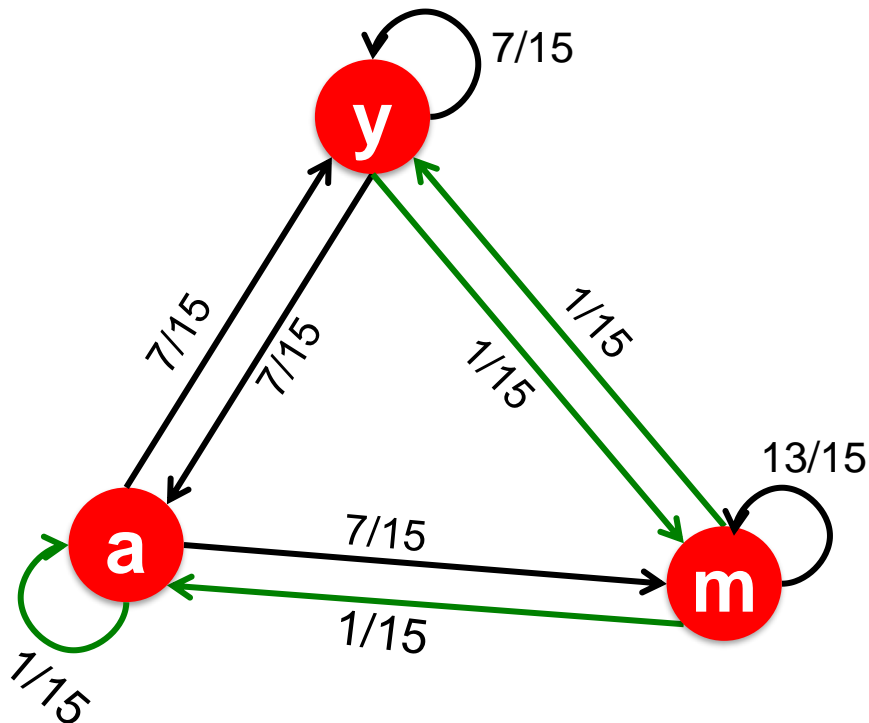
$[1/N]_{N \times N}$...matriz N por N com todas entradas iguais a $1/N$

- **Temos um problema recursivo:** $\mathbf{r} = A \cdot \mathbf{r}$
- A que se pode aplicar o método das potências (Power)

- β ?

– Na prática $\beta = 0.8, 0.9$ (5 passos em média para saltar)

Exemplo: Random Teleports ($\beta = 0.8$)



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

y	7/15	7/15	1/15
a	7/15	1/15	1/15
m	1/15	7/15	13/15

A

y		1/3	0.33	0.24	0.26		7/33
a	=	1/3	0.20	0.20	0.18	...	5/33
m		1/3	0.46	0.52	0.56		21/33

PageRank in Matlab

Ver

<http://www.mathworks.com/moler/exm/chapters/pagerank.pdf> para mais
informação

Aplicação a uma “pequena rede”

- Consideremos apenas um número reduzido de páginas
 - exemplo: $N=20$ páginas acedíveis a partir de <http://www.ua.pt>
- Principais passos:
 - Obter informação das páginas e suas ligações
 - Com base nos links obter M (ou H)
 - Criar a matriz A aplicando o método da Google para evitar dead ends e spider traps
 - Aplicar power method
 - Apresentar resultados

Obter informação das páginas e suas ligações

- Uma forma simples, em Matlab, é usando a função **surfer()**
 - **disponibilizada** em <http://www.mathworks.com/moler/exm/exmfilelist.html>
 - Copyright 2013 Cleve Moler & The MathWorks, Inc.
- Fazendo: `[U,L]=surfer('http://www.ua.pt',20);`
- Temos em U as URLs
- E em L as ligações

Resultados exemplificativos

```
>> U{1:6}
```

```
ans =
```

```
'http://www.ua.pt'
```

```
'http://static.ua.pt/js/jquery/jquery-1.11.3.min.js'
```

```
'http://static.ua.pt/js/uacookies/1/cookies.pt.min.js'
```

```
'http:/'
```

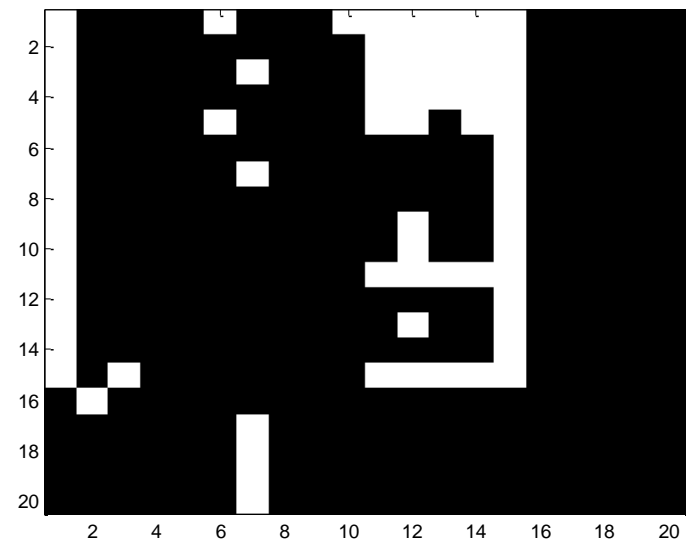
```
'http://my.ua.pt'
```

```
'http://uaonline.ua.pt'
```

L:

```
imagesc(L);
```

```
colormap(gray);
```



Obter H e A

```
H=full(L);
```

```
c=sum(full(L)); % número de ligações (d)
```

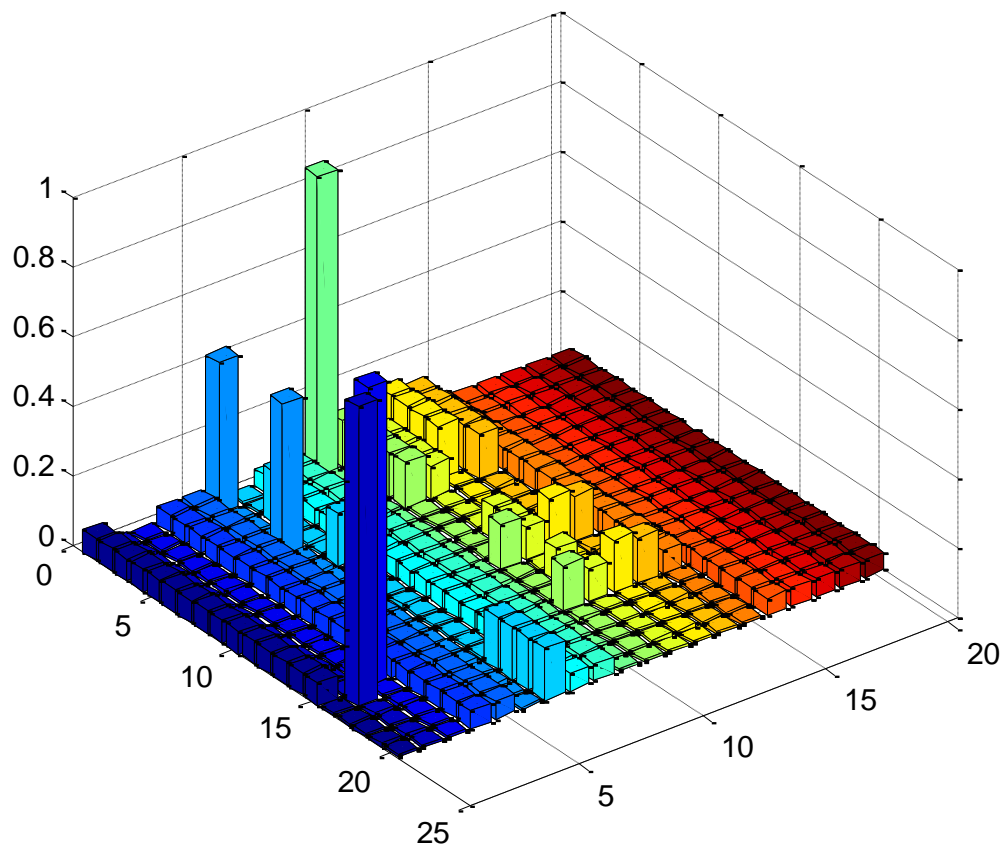
```
H=H./repmat(c,N,1)
```

```
p=0.85
```

```
A=p*H+(1-p)*ones(N)/N % matriz da Google
```

```
A(isnan(A))=1/N % resolver dead ends
```

A



Aplicar “power method”

```
x0=ones(N,1)/N;
```

```
% -----
```

```
iter=1;
```

```
x=x0;
```

```
epsilon=1e-3;
```

```
while 1
```

```
    fprintf(1,'iteração %d\n',iter);
```

```
    xold=x;
```

```
    x=A*x;
```

```
    if max(abs(x-xold))<epsilon break ; end
```

```
    iter=iter+1;
```

```
end
```

```
x
```

Apresentar resultados

```
[xs idx]=sort(x,'descend');  
  
for p=1:N  
    fprintf(1,'PageRank=%.3f : %s \n',  
            x(idx(p)), U{idx(p)});  
end
```

Resultados finais

PageRank=0.108 : <http://www.ua.pt/cookies> ,
PageRank=0.104 : <http://www.ua.pt> ,
PageRank=0.071 : <http://> ,
PageRank=0.065 : <http://my.ua.pt> ,
PageRank=0.062 : <http://static.ua.pt/js/uacookies/1/cookies.pt.min.js> ,
PageRank=0.056 : <http://static.ua.pt/js/jquery/jquery-1.11.3.min.js> ,
PageRank=0.056 : <http://> ,
PageRank=0.056 : <http://www.ua.pt/40anos> ,
PageRank=0.042 : <http://elearning.ua.pt> ,
PageRank=0.039 : <http://sinbad.ua.pt> ,
PageRank=0.039 : <http://portefolio.ua.pt> ,

- Estes resultados podem ser confirmados usando
- $r = \text{pagerank}(U, L)$
 - A diferença entre o vector x obtido e r tem de ser zero

Para casa

- Obter `surfer()` e `pagerank()` de Cleve Moler
 - <https://www.mathworks.com/matlabcentral/fileexchange/4822-using-numerical-computing-with-matlab-in-the-classroom/content/surfer.m>
- Fazer com que o Matlab consiga encontrar essas funções
 - Adicionando o directório em que as colocar à path ou colocando-as no seu directório de trabalho
- Fazer os exemplos em <http://www.mathworks.com/moler/exm/chapters/pagerank.pdf>
- Experimentar o código dos slides anteriores
- Experimentar com outras “mini redes”

E na realidade ?

Apenas 2 ou 3 slides para terem uma ideia, pois começa a sair do âmbito de MPEI

Calcular para toda a web ...

- O passo mais importante é a multiplicação

$$r^{k+1} = A \cdot r^k$$

- Fácil se desse para ter tudo em memória: A , r^{k+1} , r^k
- Se $N = 10^9$ páginas
 - 1 bilhão na América, 1000 milhões na Europa
- E considerarmos 4 bytes por entrada
- Temos:
 - 2×10^9 posições para os 2 vectores (aprox. 8GB)
 - Uma matriz A com N^2 elementos
 - Ou seja 10^{18}

Partes da solução ...

- Aproveitar o facto da matriz ser esparsa
- Codificá-la com base apenas nas entradas não-nulas

origem	grau	Nós destino
0	3	1, 5, 7
1	5	17, 64, 113, 117, 245
2	2	13, 23

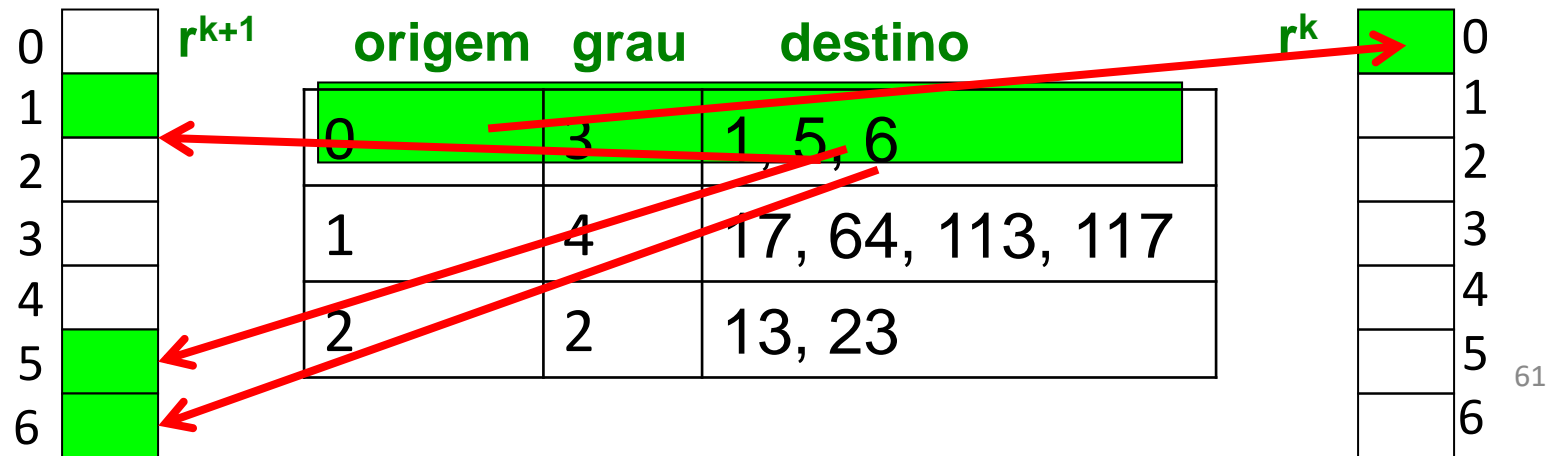
- Espaço proporcional aproximadamente ao número de links
- Por exemplo: 10N, or $4 \cdot 10^9 \cong 40\text{GB}$
- **Contínua a “não caber” em memória mas cabe em disco**

Algoritmo básico: Passo de actualização

- Assumindo que r^{k+1} cabe em memória
 - r^k e a matriz no disco
- 1 passo da iteração do método das potências :

Inicializar todas as entradas de r^{k+1} com $(1-\beta) / N$
Para cada página i (com grau d_i):
Ler para memória: $i, d_i, dest_1, \dots, dest_{d_i}, r^k(i)$
Para $j = 1 \dots d_i$
 $r^{k+1}(dest_j) += \beta r^k(i) / d_i$

Exemplo



61

Inicializar todas as entradas de r^{k+1} com $(1-\beta) / N$

Para cada página ORIGEM i (com grau d_i):

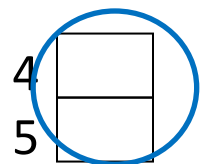
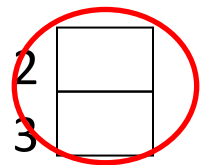
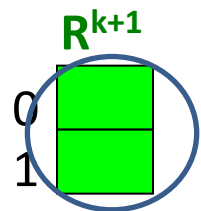
Ler para memória: $i, d_i, \text{dest}_1, \dots, \text{dest}_{d_i}, r^k(i)$

Para $j = 1 \dots d_i$

$$r^{k+1}(\text{dest}_j) += \beta r^k(i) / d_i$$

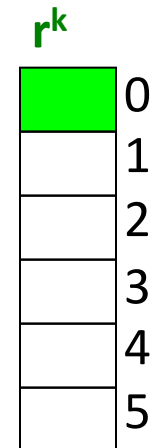
E se nem r^{k+1} cabe em memória ?

- Dividir r^{k+1} em k blocos que caibam em memória
- Processar a Matriz e r^k uma vez para cada bloco



origem	grau	destino
0	4	0, 1, 3, 5
1	2	0, 5
2	2	3, 4

H



Alguns problemas do Page Rank

- **Mede a importância genérica**
 - Não tem em conta “autoridades” num tópico específico
- **Solução:** Topic-Specific PageRank
- **Usa uma medida única de importância**
- **Solução:** Hubs-and-Authorities
- **Susceptível a spam de links**
 - Por exemplo “spam farms”: topografias artificiais de links criadas para aumentar o pagerank
- **Solução:** TrustRank

Para saber mais

- Capítulo *Link Analysis* do livro **Mining of Massive Datasets**
 - Disponível em <http://infolab.stanford.edu/~ullman/mmds/book.pdf>
- Capítulo *PageRank* do livro **Experiments with MATLAB** de C. Moler
 - e respectivo software
 - <http://www.mathworks.com/moler/exm/chapters.html>
- Notas do Prof. Paulo Jorge Ferreira “MPEI - pagerank”
 - Disponíveis no elearning
- Artigos dos autores do PageRank e fundadores da Google
 - É uma questão de usar o Google 😊

Nesta apresentação foram usados e/ou adaptados slides da seguinte apresentação:

Analysis of Large Graphs: Link Analysis, PageRank

Mining of Massive Datasets

Jure Leskovec, Anand Rajaraman, Jeff Ullman Stanford
University

<http://www.mmds.org>



Note to other teachers and users of these slides: We would be delighted if you found this our material useful in giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. If you make use of a significant portion of these slides in your own lecture, please include this message, or a link to our web site: <http://www.mmds.org>

Sets in Matlab

A small detour to learn some usefull things in Matlab

Sets of words and Matlab

- We need to know how to have sets of words in Matlab
 - Sets of different sizes
 - Words in each set with different sizes

Storing More Than Numbers

- MATLAB matrices store numeric results
- What about words, names, strings?
- What about arrays of arrays?
- What about Sets ?
- MATLAB provides more containers to store data
 - Character arrays
 - Cell arrays
 - Structures

Character Arrays

■ Examples:

```
» C = 'Hello';           %C is a 1x5 character array.  
» D = 'Hello there';    %D is a 1x11 character array.  
» A = 43;               %A is a 1x1 double array.  
» T = 'How about this character string?'
```

```
» size(T)  
ans =
```

```
1      32
```

How are Characters Stored?

- Character arrays are similar to vectors, except:
 - Each cell contains a single digit

- **Example**

```
» u = double(T)    % double is a dedicated function.  
» char(u)          % performs the opposite function.
```

- **Exercise**

```
» a = double('a')  
» char(a)
```

- **Questions:** What is the numerical value of 'a' and what does it mean?

Manipulating Strings

- Strings can be manipulated like arrays.

- **Examples**

```
» u = T(16:24)
» u = T(24:-1:16)
» u = T(16:24) '
» v = 'I can''t find the manual!' % Note quote in
    string
» u = 'If a woodchuck could chuck wood, ' ;
» v = 'how much wood could a woodchuck chuck? ' ;
» w = [u, ' ', v] % string concatenation in Matlab
» disp(w) % works just like for arrays
```

Cell Arrays

- Cell arrays are containers for “collections” of data of any type stored in a common container.
- Cell arrays are like a wall of PO boxes, with each PO box containing its own type of information.
- When mail is sent to a PO box the PO box number is given. Similarly each cell in a cell array is indexed.
- Cell arrays are created using cell indexing in the same way that data in a table or an array is created and referenced
- **The difference is the use of curly braces { }.**

Matrix of matrices

- Cell arrays are matrix of matrices

- Example:

```
x=[1:5]; y = floor(2.*randn(1,5));  
z = [100:-20:20];  
M = [x; y; z]
```

```
c = {M M+M; M(:,1) M(3,:) }
```

```
c =  
2×2 cell array
```

```
{3×5 double} {3×5 double}  
{3×1 double} {1×5 double}
```

Cell array example

- create same way as arrays but use (curly) braces

```
>> a = { i 5:-1:2 'carrots'; magic(2) 77 NaN }
```

```
a =
```

```
 [0 + 1.0000i] [1x4 double] 'carrots'  
 [2x2 double] [    77] [   NaN]
```


Create **empty cell array**

Using `cell()` function:

```
a = cell( rows, columns)
```

```
a = cell( 3, 6 )
```

```
a =
```

```
    []    []    []    []    []    []  
    []    []    []    []    []    []  
    []    []    []    []    []    []
```

```
whos a
```

Name	Size	Bytes	Class
a	3x6	72	cell

Cell Array Access

- Cell arrays look a lot like arrays but they cannot generally be manipulated the same way.
- Cell arrays should be considered more as data “containers” and must be manipulated accordingly.
 - *Cell arrays cannot be used in arithmetic computations like arrays can, e.g., $+ - * / ^$*

Addressing Cell Arrays

- $A(i,j) = \{x\}$

this is called CELL INDEXING

- $A\{i,j\} = x$

this is called CONTENT ADDRESSING

- either can be used, but be careful...

Examples

```
first = 'Hello';  
second = {'hello', 'world', 'from', 'me'};  
  
third(1,1) = {'happy'};    % Cell indexing  
third{2,1} = 'birthday';  % Content addressing  
third{3,1} = 40;
```

■ What will we obtain from ?

```
>> third  
>> third(1,1), third{1,1}  
>> third(2,1), third{2,1}  
>> third(3,1), third{3,1}
```

Cell Arrays of Strings

- All rows in a string array MUST have the same number of columns ... this is a problem for representing our sets of words
 - An many other problems
- Solution?
- **Cell arrays**

Exercise

```
C = {'How'; 'about'; 'this for a'; 'cell array of strings?'}
```

```
size(C)
```

```
C(2:3)
```

```
C([4,3,2,1])
```

```
[a,b,c,d] = deal(C{:})
```

Examples

```
» C = cell(2,3) % Defines C to be a cell array
» C(1,1) = {'This does work'} % ( ) refer to PO Box

» C{2,3} = 'This works too' % { } refers to
    contents
```

Try:

```
» A = cell(1,3) % Note 1 x 3

» A = {'My' , 'name', 'is' , 'Burdell'} % Note 1 x 4

» A = {'My'; 'name'; 'is' ; 'Burdell'}
```

Get more info:

```
» help lists
```

Set Operations

- Matlab provides several functions for set operations

<u>intersect</u>	Set intersection of two arrays
<u>ismember</u>	Array elements that are members of set array
<u>setdiff</u>	Set difference of two arrays
<u>setxor</u>	Set exclusive OR of two arrays
<u>union</u>	Set union of two arrays
<u>unique</u>	Unique values in array
<u>ismembertol</u>	Members of set within tolerance
<u>uniquetol</u>	Unique values within tolerance
<u>join</u>	Combine two tables or timetables by rows using key variables
<u>innerjoin</u>	Inner join between two tables or timetables
<u>outerjoin</u>	Outer join between two tables or timetables

join

Example

```
A={'a' 'e' 'i' 'o' 'u'}
```

```
B={'a','b','c','d','e'}
```

```
C=intersect(A,B) % o que dará ?
```

```
D=union(A,B)
```

```
ismember(A(1),C)
```

```
ismember(A,D) % o que dará ?
```

```
ans =  
    1    1    1    1    1
```

Some Useful functions

- » `iscellstr(A)` % logical test for a cell array of strings
- » `ischar(A)` % logical test for a string array
- » **`celldisp(B)`** % recursively displays cell array, i.e., if content a cell array, also displays its content
- » `cellstr(B)`

Use `help` to get information on each of these functions ...

Some Useful functions

- » **cellplot(B)** % displays in figure window drawing of 1D or 2D cell array
- » **cell2mat(B)** % convert a cell array of numbers to a numerical array
- » **num2cell(A)** % convert an array of numbers to a cell array
- » **cellfun(A)** % applies a specified function to the content of every element of a cell array

Structures

- Numeric, character and cell arrays all reference the individual elements by number
- Structures reference individual elements within each row (called “fields”) by name.
- To access these fields, the dot “.” notation is used.
- Assignment is as follows:
`structurename.fieldname =
datatype;`

Creating a Structure...

- Let's create a simple structure:

```
person.firstname = 'António';  
person.lastname = 'Teixeira';  
person.address1 = 'DETI/IEETA,  
University of Aveiro';  
person.city = 'Aveiro';  
person.zip = '3810-193 AVEIRO';
```

■ ■ ■

person =

firstname: 'António'

lastname: 'Teixeira'

address1: [1x32 char]

city: 'Aveiro'

zip: '3810-193 AVEIRO'

More on Structures...

- A structure can have a field that is a structure itself.
- A structure array is that which contains more than one record for each field name.
- As the structure array is expanded (more records are created), all unassigned fields are filled with an empty matrix.
- All structures have the same number of fields and elements in each field.

Example

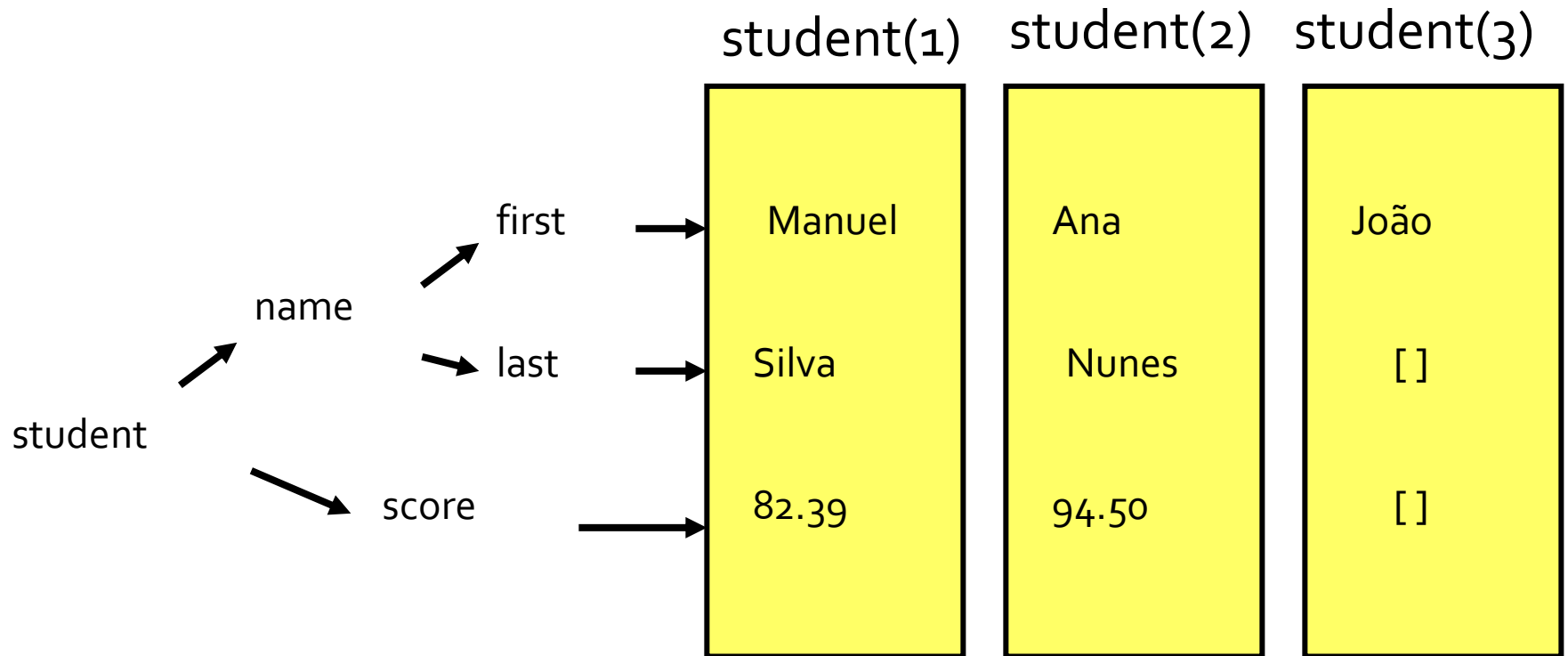
```
student(1).name.first = 'Manuel';  
student(1).name.last = 'Silva';  
Student(1).score = 82.39;
```

```
student(2).name.first = 'Ana';  
student(2).name.last = 'Nunes';  
student(2).score = 94.50;
```

```
student(3).name.first = 'João';
```

...

Example (cont.)



Sources used

- PPT on “Strings, Cell Arrays and Structures” of AE6382-9 Design Computing course, Georgia Tech, 2006
- PPT “Matlab Cell Arrays” by Greg Reese, Miami University, 2011
- Chapters 7 and 8 of Duane Hanselman and Bruce Littlefield (2003), “Matlab 6 Curso Completo”, Prentice Hall