Simulação de Transições de Estado com Cadeias de Markov:

Equipe:

Henrico Gramacho Oliveira RA: 12722130475
Paulo Roberto de Souza Carneiro Filho RA: 1272215237
Lucca Dávila Bacelar RA: 1272218370

Código Fonte:

```
function simular_markov()
   # 1. Definição dos nomes dos estados
   estados = ["Estudando", "Relaxando", "Dormindo"]

   # 2. Matriz de transição P (3x3) — cada linha representa as probabilidades de ir para
outros estados
   P = [
      0.6  0.3  0.1;  # De Estudando
      0.4  0.4  0.2;  # De Relaxando
      0.2  0.3  0.5   # De Dormindo
   ]

   # 3. Vetor de estado inicial: 100% no estado "Estudando"
   estado_atual = transpose([1.0, 0.0, 0.0])

   # 4. Número de passos (iterações no tempo)
   n_passos = 10

   # 5. Mostrar o passo 0 (estado inicial)
   println("Passo 0:")
   for i in 1:length(estados)
      println("  ", estados[i], ": ", round(estado_atual[i], digits=4))
   end

   # 6. Loop principal: para cada passo de tempo, atualizar e imprimir as probabilidades
   for passo in 1:n_passos
      # Multiplicação do vetor de estado pela matriz de transição
      estado_atual = estado_atual * P

      # Mostrar as probabilidades após o passo
      println("Passo $passo:")
      for i in 1:length(estados)
         println("  ", estados[i], ": ", round(estado_atual[i], digits=4))
      end
   end
end
# 7. Chamada da função
simular_markov()
```

Explicações de aplicação:

A simulação da cadeia de Markov foi desenvolvida utilizando a linguagem Julia na plataforma online JDoodle, que permite executar códigos diretamente do navegador, sem a necessidade de instalação local.

1. Acessar o JDoodle
   - Entre no site: https://www.jdoodle.com/online-compiler-julia
2. Colar o código fonte
   - Copie todo o código-fonte da aplicação e cole no editor de código da página.
3. Executar
   - Clique no botão "Execute" no topo da tela.

   - O output aparecerá logo abaixo, mostrando a evolução das probabilidades dos estados a cada passo do tempo.

Situação problema e a solução aplicada:

O problema modelado neste projeto é a rotina diária de um estudante universitário, considerando três estados possíveis em que ele pode se encontrar: Estudando, Relaxando ou Dormindo. Esses estados representam atividades do dia-a-dia, permitindo a aplicação de uma cadeia de Markov para analisar as transições entre eles ao longo do tempo.

A cadeia de Markov é uma ferramenta matemática que modela processos "Estatísticos", nos quais o estado futuro depende do estado atual, uma propriedade conhecida como "memória curta". Para isso, foi definida uma matriz de transição com as probabilidades de mudança de um estado para outro:

- De "Estudando", há 60% de chance de continuar estudando, 30% de relaxar e 10% de dormir.

- De "Relaxando", há 40% de chance de voltar a estudar, 40% de continuar relaxando e 20% de dormir.

- De "Dormindo", há 20% de chance de estudar, 30% de relaxar e 50% de continuar dormindo.

A simulação foi feita em julia utilizando um vetor de estado inicial, representando que o estudante começa 100%/1.0 no estado "Estudando". Em seguida, o vetor é multiplicado repetidamente pela matriz de transição ao longo de 10 passos, mostrando como as probabilidades associadas a cada estado evoluem com o tempo.

Essa abordagem permite observar, por exemplo, a estabilização do comportamento ao longo do tempo e a tendência do estudante a passar mais tempo em certos estados, de acordo com as regras definidas.

A solução aplicada exemplifica bem a utilidade das cadeias de Markov para representar sistemas com transições entre estados, mesmo em situações simples do cotidiano, como a rotina de um aluno.

Caso de uso aplicado:

Consideramos a rotina diária de um estudante, que pode ser um dos seguintes estados:

- Estudando

- Relaxando

- Dormindo

Esses três estados formam a base da cadeia de Markov.

Matriz de Transição:

	Estudando	Relaxando	Dormindo
Estudando	0.6	0.3	0.1
Relaxando	0.4	0.4	0.2
Dormindo	0.2	0.3	0.5

Estado Inicial:

O vetor de estado inicial representa que, no tempo 0, o estudante está 100%/1.0 no estado "Estudando":

[1.0, 0.0, 0.0] (Estudando, Relaxando e Dormindo)

Cálculo manual dos primeiros passos:

Passo 1

Vetor anterior (passo 0):
[1.0, 0.0, 0.0] (100% no estado "Estudando”")
Estudando:
(1.0 x 0.6) + (0.0 x 0.4) + (0.0 x 0.2) =
0.6+0+0=06
Relaxando:
(1.0 x 0.3) + (0.0 x 04) + (0.0 x 0.3) =
03+0+0=03
Dormindo:
(1.0 x 0.1) + (0.0 x 0.2) + (0.0 x 0.5) =
01+0+0=01

Passo 2

Vetor anterior (passo 1):
[0.6, 0.3, 0.1]
Estudando:
(0.6 x 0.6) + (0.3 x 04) + (0.1 x 0.2) =
0.36 + 0.12 + 0.02 = 0.5
Relaxando:
(0.6 x 0.3) + (0.3 x 04) + (0.1 x 0.3) =
0.18 + 0.12 + 0.03 = 0.33
Dormindo:
(0.6 x 0.1) + (0.3 x 0.2) + (0.1 x 0.5) =
0.06 + 0.06 + 0.05 = 0.17

Passo 3

Vetor anterior (passo 3):
[0.5,0.33, 0.17] x P
Estudando:
0.5 x 0.6 + 0.33 x 0.4 + 0.17 x0.2=0.3+
0.132 + 0.034 = 0.466
Relaxando:
0.5 x 0.3 + 0.33 x 0.4 + 0.17 x 0.3 =0.15+
0.132 + 0.051 = 0.333
Dormindo:
0.5 x 0.1 + 0.33 x 0.2 + 0.17 x 0.5=0.05 +
0.066 + 0.085 = 0.201

Passo 4

Vetor anterior (passo 3):
[0.466, 0.333, 0.201] x P
Estudando:
0.466 x 0.6 + 0.333 x 0.4 + 0.201 x 0.2 = 0.453
Relaxando:
0.466 x 0.3 + 0.333 x 0.4 + 0.201 x 0.3 = 0.333
Dormindo:
0.466 x 0.1 + 0.333 x 0.2 + 0.201 x 0.5 = 0.213

Passo 5

Vetor anterior (passo 4):
[0.453, 0.333, 0.213]
Estudando:
(0.453 x 0.6) + (0.333 x 0.4) + (0.213 x 0.2) =
0.2718 + 0.1332 + 0.0426 = 0.4476
Relaxando:
(0.453 x 0.3) + (0.333 x 0.4) + (0.213 x 0.3) =
0.1359 + 0.1332 + 0.0639 = 0.333
Dormindo:
(0.453 x 0.1) + (0.333 x 0.2) + (0.213 x 0.5) =
0.0453 + 0.0666 + 0.1065 = 0.2184

Passo 6

Vetor anterior (passo 5):
[0.4479, 0.333, 0.2188]
Estudando:
(04479 x 0.6) + (0.333 x 0.4) + (0.2188 x 0.2) =
0.26874 + 0.1332 + 0.04376 = 0.4457
Relaxando:
(04479 x 0.3) + (0.333 x 0.4) + (0.2188 x 0.3) =
0.13437 + 0.1332 + 0.06564 = 0.3332
Dormindo:
(04479 x 0.1) + (0.333 x 0.2) + (0.2188 x 0.5) =
0.04479 + 0.0666 + 0.1094 = 0.2208

Passo 7

Vetor anterior (passo 6):
[0.4458, 0.333, 0.2209]
Estudando:
(0.4458 x 0.6) + (0.333 x 0.4) + (0.2209 x 0.2) =
0.26748 + 0.1332 + 0.04418 = 0.44486
Relaxando:
(0.4458 x 0.3) + (0.333 x 0.4) + (0.2209 x 0.3) =
0.13374 + 0.1332 + 0.06627 = 0.33321
Dormindo:
(0.4458 x 0.1) + (0.333 x 0.2) + (0.2209 x 0.5) =
0.04458 + 0.0666 + 0.11045 = 0.22163

Passo 8

Vetor anterior (passo 7):
[0.4450, 0.333, 0.2217]
Estudando:
(04450 x 0.6) + (0.333 x 0.4) + (0.2217 x 0.2) =
0.267 + 0.1332 + 0.04434 = 0.44454
Relaxando:
(04450 x 0.3) + (0.333 x 0.4) + (0.2217 x 0.3) =
0.1335 + 0.1332 + 0.06651 = 0.33321
Dormindo:
(0.4450 x 0.1) + (0.333 x 0.2) + (0.2217 x 0.5) =
0.0445 + 0.0666 + 0.11085 = 0.22195

Passo 9

Vetor anterior (passo 8):
[0.4447, 0.333, 0.2220]
Estudando:
(04447 x 0.6) + (0.333 x 0.4) + (0.2220 x 0.2) =
0.26682 + 0.1332 + 0.0444 = 0.44442
Relaxando:
(04447 x 0.3) + (0.333 x 0.4) + (0.2220 x 0.3) =
0.13341 + 0.1332 + 0.0666 = 0.33321
Dormindo:
(04447 x 0.1) + (0.333 x 0.2) + (0.2220 x 0.5) =
0.04447 + 0.0666 + 0.111 = 0.22207

Passo 10

Vetor anterior (passo 9):
[0.4445, 0.333, 0.2221]
Estudando:
(0.4445 x 0.6) + (0.333 x 0.4) + (0.2221 x 0.2) =
0.2667 + 0.1332 + 0.04442 = 0.44432
Relaxando:
(0.4445 x 0.3) + (0.333 x 0.4) + (0.2221 x 0.3) =
0.13335 + 0.1332 + 0.06663 = 0.33318
Dormindo:
(0.4445 x 0.1) + (0.333 x 0.2) + (0.2221 x 0.5) =
0.04445 + 0.0666 + 0.11105 = 0.2221

Conclusão:

Os cálculos manuais batem praticamente 100% com os valores exibidos no terminal da sua simulação em Julia, com pequenas diferenças apenas por arredondamentos de casas decimais.

Observações:

A partir do passo 6, as probabilidades começam a estabilizar, tendendo ao estado estacionário.

O sistema converge para aproximadamente: **44,45% Estudando; 33,33% Relaxando; 22,22% Dormindo;** Isso mostra que, ao longo do tempo, o comportamento do estudante se estabiliza nesses percentuais, independentemente do estado inicial.