

Kawan Oliveira Carneiro- 12724119606

Vinícius Paula Ribeiro- 12723131464

Arthur Cristo e Silva- 1272323659

Maria Clara Maluf Caldas De Souza-1272325024

Simulação de Cadeia de Markov em Julia- Controle de Praga Em Lavoura

Sumário:

1.0 Situação Problema

1.1 Explicação da Situação Problema

2.0 Solução

2.1 Explicação da solução

3.0 Código-Fonte

3.1. Prompt do Código

4.0 Conclusão

1.0 Situação Problema

Um agricultor dono de grandes fazendas no interior da Bahia, iniciou o plantio de milho para vender durante o período junino. Entretanto, ele notou o grande crescimento repentino de uma praga nas suas áreas de cultivo, causando perdas consideráveis na safra.

1.1 Explicação da Situação Problema

Esse agricultor, mapeou a lavoura em pequenos setores e observou a seguinte padrão na proliferação das pragas:

- Um setor que ainda não foi infestado tem uma chance de continuar sem pragas e uma chance de ser infestados após o decorrer de n dias.
- Um setor já infestado possui a chance de ficar ainda mais infestado ou de voltar ao estado natural após o uso de pesticida pós o decorrer de n dias.

2.0 Solução

Com o intuito de obter uma análise mais precisa do quantitativo do crescimento das pragas, o agricultor contratou uma equipe que desenvolveria um programa para simular a taxa de crescimento das pragas.

A equipe de desenvolvimento analisou a situação proposta e percebeu que a melhor alternativa seria criar uma simulação do crescimento das pragas usando uma Cadeia De Markov, já que o estado atual das pragas não dependia do estado de crescimento anterior, apenas dos fatores externos.

2.1 Explicação da Solução

A partir das observações feitas pelo agricultor, a equipe de desenvolvimento criou o seguinte modelo de algoritmo.

1. O agricultor pode definir quantos níveis de infestação achar necessário, sejam eles:

- 1.1. Sem infestação

- 1.2. Pouca Infestação
- 1.3. Média Infestação
- 1.4. Grave Infestação.

2. Após a definição dos estados, é definida uma matriz de transição entre os estados após 1 dia.

Ex:

$$SI = [0.4 \ 0.3 \ 0.2 \ 0.1]$$

$$PI = [0.2 \ 0.4 \ 0.3 \ 0.1]$$

$$MI = [0.1 \ 0.2 \ 0.5 \ 0.2]$$

$$GI = [0.1 \ 0.1 \ 0.3 \ 0.5]$$

Essa matriz de transição é utilizada para fazer o cálculo percentual para descobrir as chances de o estado inicial evoluir para um próximo estado depois de um período de n dias.

3. Depois de definir os estados percorridos e a matriz de transição entre os estados, será necessário definir a quantidades de dias que a matriz deve ser percorrida.

3.0 Código fonte

```
1 function simular_cadeia_markov()
2     println("Simulação de Cadeia de Markov - Controle de Pragas em Lavoura")
3
4     # Entrada do número de estados
5     print("Digite o número de estados (n): ")
6     n = parse{Int, readline()}
7
8     # Entrada da matriz de transição
9     matriz = zeros{Float64, n, n}
10    println("Digite os valores da matriz de transição (linha por linha):")
11    for i in 1:n
12        while true
13            print("Digite os valores da linha $i separados por espaço (devem somar 1): ")
14            entrada = split(chomp(readline()))
15            if length(entrada) != n
16                println("Erro: você deve digitar exatamente $n valores.")
17                continue
18            end
19
20            linha = parse{Float64, entrada}
21            if abs(sum(linha) - 1.0) > 1e-6
22                println("Erro: os valores da linha devem somar 1.")
23                continue
24            end
25
26            matriz[i, :] = linha
27            break
28        end
29    end
30
31    # Entrada do estado inicial
32    print("Digite o estado inicial (de 1 até $n): ")
33    estado_inicial = parse{Int, readline()}
34
35    # Vetor de estado inicial (1 na posição do estado inicial, 0 nas outras)
36    vetor_estado = zeros{Float64, n}
37    vetor_estado[estado_inicial] = 1.0
38
39    # Número de dias
40    print("Digite o número de dias para simular: ")
41    dias = parse{Int, readline()}
42
43    # Simulação
44    println("\nIniciando simulação...\n")
45    println("Dia 0: ", round.(vetor_estado; digits=4))
46
47    for dia in 1:dias
48        vetor_estado = vetor_estado' * matriz # Transposta para multiplicação correta
49        vetor_estado = vec(vetor_estado)      # Converte para vetor 1D novamente
50        println("Dia $dia: ", round.(vetor_estado; digits=4))
51    end
52
53    # Resultado final
54    println("\nApós $dias dias, a distribuição de probabilidade dos estados é:")
55    for i in 1:n
56        println("Estado $i: ", round(vetor_estado[i] * 100; digits=2), "%")
57    end
58 end
59
60 # Chamada da função principal
61 simular_cadeia_markov()
```

3.1 Prompt do Código

```

Activating new project at `C:\Users\kawan\.julia\environments\v1.11`
Simulação de Cadeia de Markov - Controle de Pragas em Lavoura
Digite o número de estados (n): 2
Digite os valores da matriz de transição (linha por linha):
Digite os valores da linha 1 separados por espaço (devem somar 1): 0.85 0.15
Digite os valores da linha 2 separados por espaço (devem somar 1): 0.9 0.1
Digite o estado inicial (de 1 até 2): 1
Digite o número de dias para simular: 5

Iniciando simulação...

Dia 0: [1.0, 0.0]
Dia 1: [0.85, 0.15]
Dia 2: [0.8575, 0.1425]
Dia 3: [0.8571, 0.1429]
Dia 4: [0.8571, 0.1429]
Dia 5: [0.8571, 0.1429]

Após 5 dias, a distribuição de probabilidade dos estados é:
Estado 1: 85.71%
Estado 2: 14.29%
* Terminal will be reused by tasks, press any key to close it.

```

4.0 Conclusões

Após o agricultor utilizar do software por um período de 30 dias a simulação de Cadeias de Markov aplicada ao controle de pragas em lavouras demonstrou-se uma ferramenta eficiente para prever e analisar a dinâmica de propagação de infestações agrícolas.

Utilizando a linguagem Julia, foi possível desenvolver um modelo computacional capaz de representar diferentes estados da lavoura: "sem praga", "infestação leve", "infestação moderada" e "infestação grave", e calcular as probabilidades de transição entre esses estados ao longo do tempo.

Os resultados da simulação permitiram identificar padrões relevantes no comportamento das pragas e, com isso, fornecer subsídios para a tomada de decisões estratégicas no manejo e controle agrícola. A implementação em Julia mostrou-se vantajosa pela sua alta performance numérica e sintaxe clara, contribuindo para o desenvolvimento ágil e eficiente do modelo.

Conclui-se que o uso de Cadeias de Markov em simulações agrícolas oferece uma abordagem promissora para antecipar cenários e reduzir os impactos causados por pragas.