EP MAE0699 - Tópicos de probabilidades

Fabricio Kassardjian nusp:2234961
Robert nusp:xxxxx

15 de maio de 2019

Introdução

O trabalho se refere a estudar as curvas de distribuição para $T(v)$ e $C(v,w), onde **T** representa o caminho mais curto de retorno ao vértice v e **C** o caminho mais curto entre os vértices v e w. O modelo de Erdo-Rényi é utilizado para gerar os grafos aleátorios com n vértices e probabilidade p de ligação entre cada par de vértices.

Modelo 2

O modelo escolhido para o teste foi usando conexões não direcionadas e preguiçoso. O fato de ser preguiçoso implica que existem conexões para continuar no mesmo vértice. Além disso fica determinado que cada connexão só pode ser usada uma unica vez para cada caminho testado, assim evita-se que a distribuição **T** tenha apenas valores 1 e 2. Como as conexões são aleatórias podem existir vértices isolados e também como não pode ser utilizado a mesma conexão para voltar podem existir valores de **T** e **C** que podemos considerar inf.

Metodologia

Estimação das distribuições

Para o teste primeiro inicia-se uma matriz A representando com **TRUE** quando a ligação entre os vértices está presente e **FALSE** quando não há ligação. A linha i da matriz representa o vértice de saída e a coluna j representa o vértice de chegada. Como o modelo é preguiçoso pode existir **TRUE** na diagonal principal da matriz, e pelo fato das conexões não serem direcionadas a matriz é simétrica.

Exemplo de matriz de conexões para 5 vértices com probabilidade de conexão 0.4:

```
A = generateMatrix(5, 0.4)
print(A)
```

```
## 5 x 5 Matrix of class "lsyMatrix"
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] FALSE  TRUE  TRUE  TRUE  TRUE
## [2,]  TRUE FALSE FALSE FALSE FALSE
## [3,]  TRUE FALSE  TRUE FALSE FALSE
## [4,]  TRUE FALSE FALSE FALSE FALSE
## [5,]  TRUE FALSE FALSE FALSE  TRUE
```

Para cada mariz gerada é testado para cada vértice o menor caminho de volta, usando uma busca em profundidade dos caminhos possíveis da matriz e armazenado o vetor com a contagem de cada valor para **T** encontrado. O mesmo é feito para cada combinação de vértices possíveis para encontrar os valores de **C**. Os caminhos podem ter tamanhos até n e iremos considerar o valor $n + 1$ como sendo infinito.

Exemplo de valores de $T(v)$ para cada vértice de **A**

```
for(i in 1:5) {
  Ti = findPath(i,i,A,0,5+1)
```

```r
  cat(sprintf("T(%d) = %d\n",i, Ti))
}
```

```
## T(1) = 6
## T(2) = 6
## T(3) = 1
## T(4) = 6
## T(5) = 1
```

Será gerado para cada tamano de $n \in 6, 7, 8, 9, 10, 11, 12$ uma amostra de 20 matrizes e feita uma contagem para cada valor de **T** encontrado. A distribuição é estimada tirando a média da contagem por $n * 20$. Assim:

$$\hat{P}(T = k) = \frac{1}{n * 20} \sum_{i=1}^{n*20} \mathbb{1}_{(T=k)}$$

Para a distribuição de C é usado processo similar mas como temos as combinações entre os pares serão estimados $n * n$ valores para cada matriz, assim:

$$\hat{P}(C = k) = \frac{1}{n^2 * 20} \sum_{i=1}^{n^2*20} \mathbb{1}_{(C=k)}$$

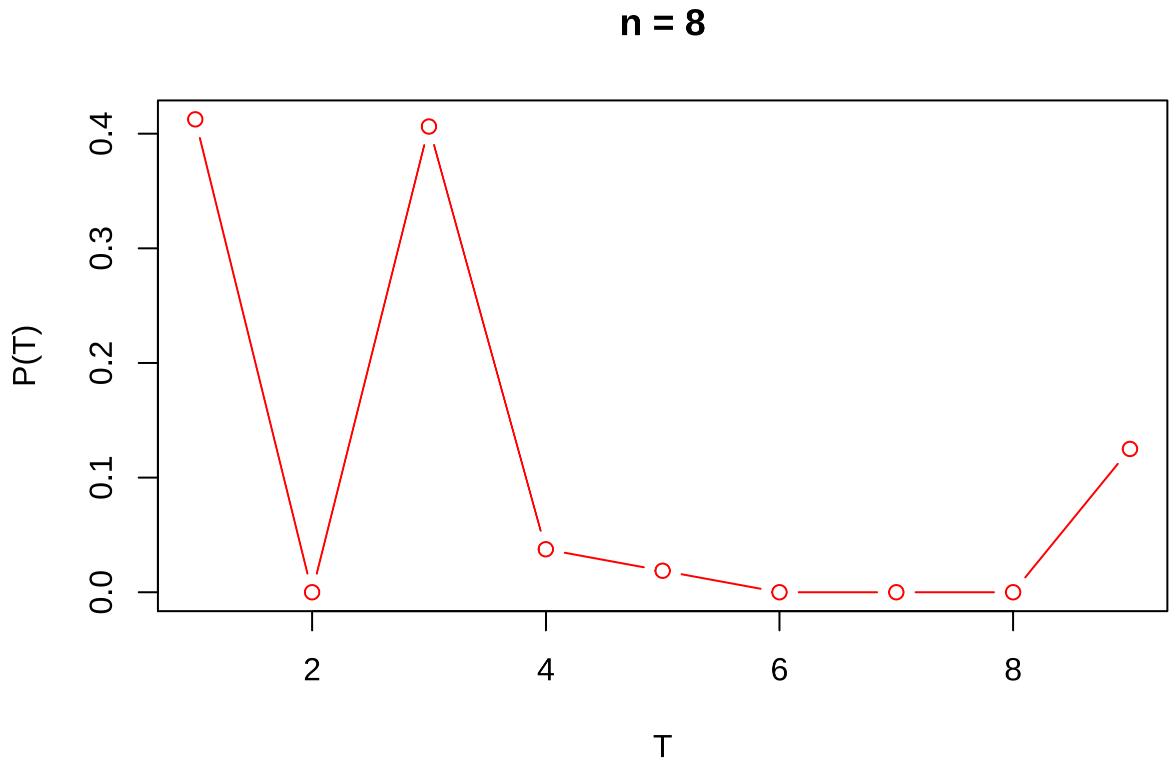
Teste de aderência

Simulação

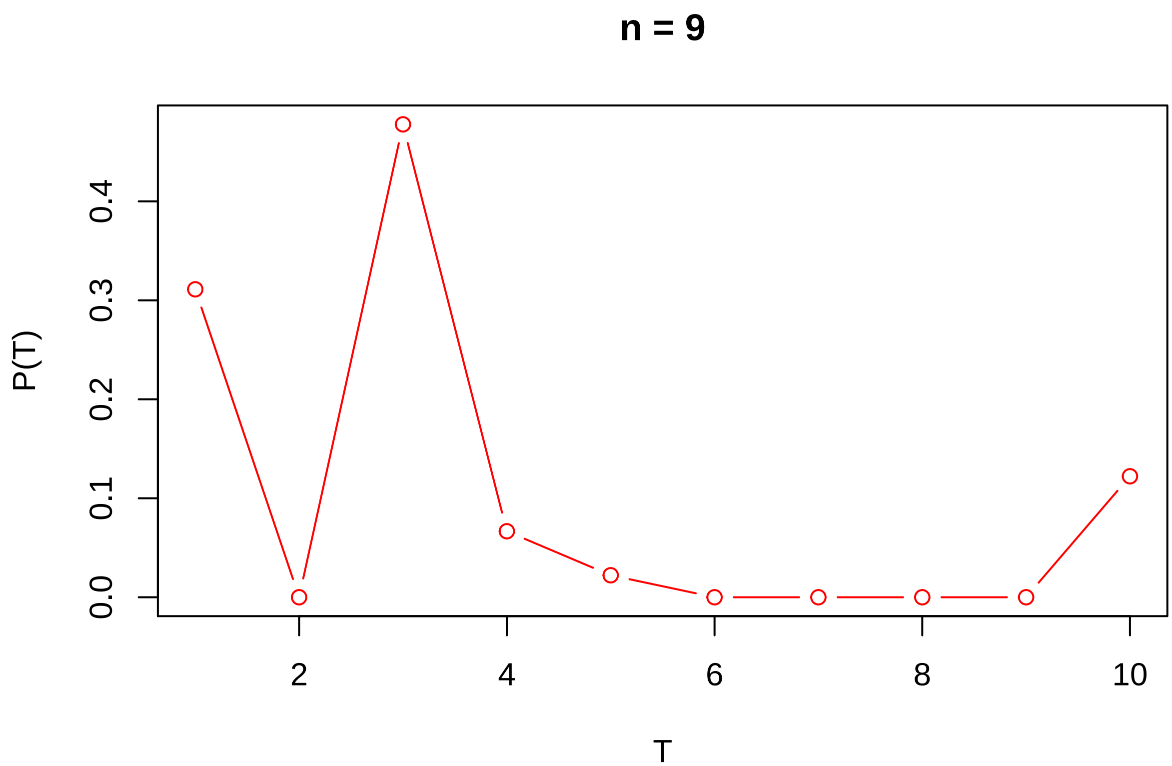
Distribuições para T

Valores estimados e gráficos da distribuição

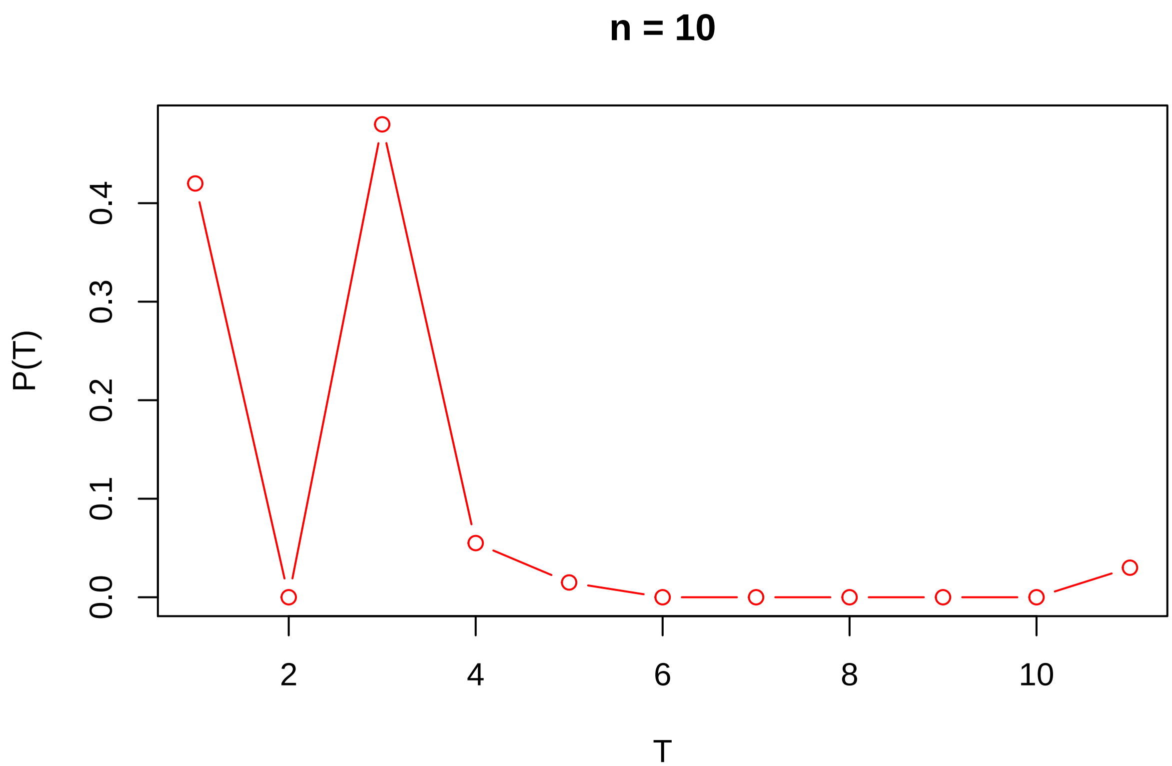
```
##
## n = 8
##                1      2      3      4      5      6      7      8    inf
## Distr. T 0.4125 0.0000 0.4062 0.0375 0.0187 0.0000 0.0000 0.0000 0.1250
```




```
##
## n = 9
##                 1      2      3      4      5      6      7      8      9
## Distr. T 0.3111 0.0000 0.4778 0.0667 0.0222 0.0000 0.0000 0.0000 0.0000
##               inf
## Distr. T 0.1222
```




```
##
## n = 10
##               1      2      3      4      5      6      7      8      9
## Distr. T 0.4200 0.0000 0.4800 0.0550 0.0150 0.0000 0.0000 0.0000 0.0000
##              10    inf
## Distr. T 0.0000 0.0300
```


n = 10
P(T)
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10
T