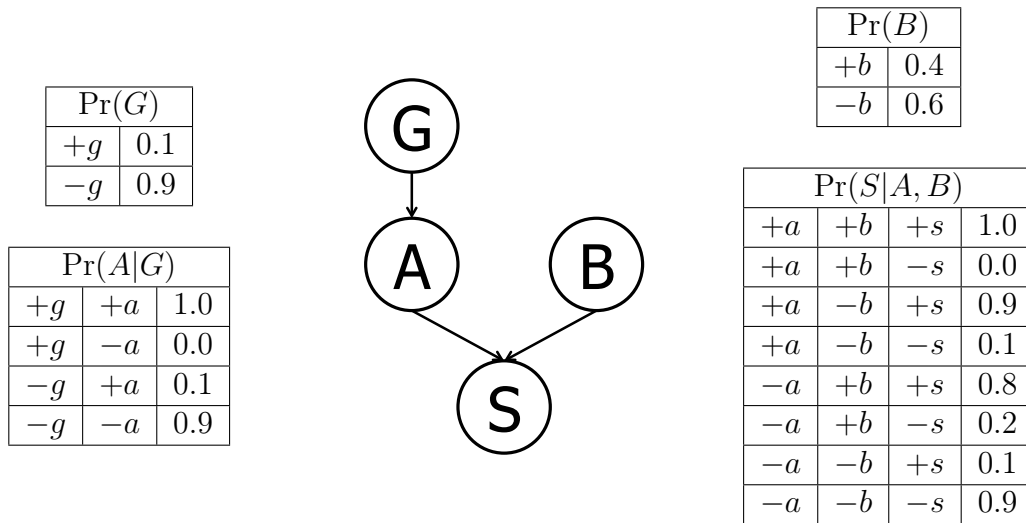


1 Inferência Exata

Suponha que o fato de um paciente apresentar um determinado sintoma (S) possa ser causado por duas doenças diferentes, A e B . Sabe-se que a variação do gene G desempenha um grande papel na manifestação da doença A . A rede bayesiana e tabelas de probabilidade condicionais correspondentes para esta situação são apresentadas abaixo.



1.1 Construção da rede bayesiana

Você deve inicialmente construir essa rede bayesiana usando a biblioteca pgmpy. O resultado desta parte do trabalho deve ser um objeto da classe `pgmpy.models.BayesianModel` que codifica adequada e completamente a rede bayesiana aqui descrita.

1.2 Inferência

Agora responda (por meio do método de inferência denominado Variable Elimination) as seguintes consultas.

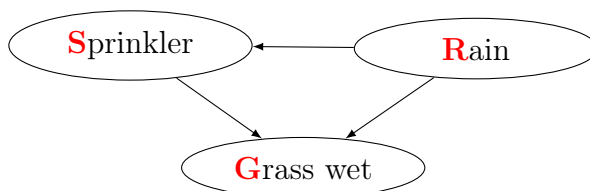
- i. Compute a seguinte entrada para a distribuição conjunta: $\Pr(+g, +a, +b, +s) =$
- ii. Qual é a probabilidade de que um paciente tem a doença A ? $\Pr(+a) =$

- iii. Qual é a probabilidade de que um paciente tem a doença A dado que el tem a doença B ? $\Pr(+a|+b) =$
- iv. Qual é a probabilidade de que um paciente tem a doença A dado que ele tem o sintoma S e a doença B ? $\Pr(+a|+s,+b) =$
- v. Qual é a probabilidade de um paciente carregar a variante do gene G dado que ele tem a doença A ? $\Pr(+g|+a) =$
- vi. Qual é a probabilidade de um paciente carregar a variante do gene G dado que ele tem a doença B ? $\Pr(+g|+b) =$

2 Inferência Aproximada

Considere a rede bayesiana fornecida abaixo.

S		
R	T	F
F	0.4	0.6
T	0.01	0.99



R	
T	F
0.2	0.8

G			
S	R	T	F
F	F	0.4	0.6
F	T	0.01	0.99
T	F	0.01	0.99
T	T	0.01	0.99

2.1 Construção da rede bayesiana

Você deve inicialmente construir essa rede bayesiana usando a biblioteca pgmpy. O resultado desta parte do trabalho deve ser um objeto da classe `pgmpy.models.BayesianModel` que codifica adequada e completamente a rede bayesiana aqui descrita.

2.2 Inferência

Nessa parte, você irá construir respostas para diversas consultas sobre a rede bayesiana previamente construída. Entretanto, você deverá produzir respostas para consultas por meio de métodos de inferência aproximativos (ou seja, o uso da classe `VariableElimination` não é permitido nesta parte do trabalho).

Implemente em Python os métodos de Amostragem a Priori (*Prior Sampling*) e Ponderação por Verossimilhança (*Likelihood Weighting*). Esses métodos devem ser implementados como funções cujas assinaturas devem ser as fornecidas a seguir:

```
1 def amostragem_a_priori(rede_bayesiana, q, obs, num_amostras):
2     # SEU CÓDIGO AQUI
3     return dict_r
```

```
1 def ponderacao_por_verossimilhanca(rede_bayesiana, q, obs, num_amostras):
2     # SEU CÓDIGO AQUI
3     return dict_r
```

Considere que a consulta que deve ser respondida é da forma a seguir:

$$\Pr(Q \mid E_0, E_2, \dots, E_n)$$

Na assinatura das funções acima, há diversos parâmetros cuja descrição é fornecida a seguir:

- **rede_bayesiana**: objeto da classe `pgmpy.models.BayesianModel` que representa a rede bayesiana.
- **q**: um `string` Python contendo o nome da variável não observada Q .
- **obs**: um `dict` Python contendo uma entrada para cada evidência E_i ($0 \leq i \leq m$). Em cada elemento (k, v) , a chave k deve ser o nome da variável, e v deve ser o valor observado para a variável. Se a consulta tiver uma lista vazia de variáveis observadas (evidências), então deve ser passado `None` como valor para esse parâmetro.
- **num_amostras**: um inteiro positivo correspondente à quantidade de amostras que devem ser geradas para responder à consulta.

Cada uma das funções acima deve retornar um dicionário Python (`dict`). Em cada elemento (k, v) desse dicionário, k é um valor possível da variável Q , e v é o valor de probabilidade computado de forma aproximada pela função.

A seguir, usando cada uma das funções implementadas, compute aproximações para as seguintes consultas: $\Pr(R)$, $\Pr(S)$, $\Pr(G)$ e $\Pr(G|S = \text{True})$ e $\Pr(G|S = \text{False})$. Para cada consulta, gere respostas com valores os seguintes valores para o parâmetro **num_amostras**: 5, 50, 500. Você nota alguma diferença nos resultados produzidos, em função do método usado, ou em função da quantidade de amostras usadas? Apresente uma discussão acerca disso em seu relatório.

Dica: A seção intitulada *Geração de realizações de variáveis* no tutorial fornecido sobre o `pgmpy` fornece um ponto de partida para você implementar o que se pede nessa parte do trabalho. Em particular, para gerar as amostras a partir da distribuição conjunta codificada na rede bayesiana, o método `get_values` da classe `TabularCPD` será útil.

O que deve ser entregue

Você deve produzir um notebook Jupyter que deve apresentar as implementações de cada parte desse trabalho. Nesse notebook, descreva **em detalhes** de que forma implementou cada uma das duas partes.

Você deve compactar todos os arquivos do trabalho em um único arquivo, que deve obrigatoriamente ser nomeado com o seguinte padrão: SEU_NOME_COMPLETO_T4.ipynb. Esse deve ser o arquivo a ser submetido até a data acordada.

Você deve também elaborar um vídeo de 10 a 15 minutos no qual você deve explicar os aspectos mais importantes de cada parte do seu trabalho. O link para acesso a esse vídeo deve estar contido na primeira célula (de texto) do notebook Jupyter.