Gibbs sampling for Bayesian Networks

Modo de funcionamento

O presente projeto em Python implementa o algoritmo de Gibbs sampling para redes Bayesianas utilizando um sistema de comandos da seguinte forma:

```
python3.6 main.py -t TARGET -u UNIVERSE -b BIF_FILE [-r] [-c] [-i]
```

- -t: Define qual a variável sobre o qual queremos inferir uma probabilidade.
- -u: Define o "universo" conhecido.
- -b: Qual a rede bayesiana a carregar (redes enviadas com o projeto encontram-se explicadas na próxima secção)
- -r: Quantas vezes queremos inferir a probabilidade. Valor default: 1
- -c: Qual o fator que permite detetar convergência, isto é, o programa deteta convergência quando após 5000 iterações, a variação do valor da probabilidade é menor que -c. Valor default: 0.001
- -i: Número de iterações máximas. Valor default: 25000

Exemplos de comandos:

1. Pr(JohnCalls=True | Burglary=True, Earthquake=True)

```
python3.7 main.py -t JohnCalls=True -u Burglary=True Earthquake=True
Alarm=True -r 5 -b earthquake.bif
```

2. Pr(JohnCalls=True | Burglary=False, Earthquake=True)

```
python3.7 main.py -t JohnCalls=True -u Burglary=False Earthquake=True -r 5 -b
earthquake.bif
```

3. Pr(MaryCalls=True | Burglary=False, Earthquake=False)

```
python3.7 main.py -t MaryCalls=True -u Burglary=False Earthquake=False -r 5 -
c 0.001 -i 25000 -r 3 -b earthquake.bif
```

4. Pr(Cancer=positive | Pollution=high, Smoker=True)

```
python3.7 main.py -t Xray=positive -u Pollution=high Smoker=True -r 5 -c
0.001 -i 25000 -b cancer.bif
```

5. Pr(Cancer=True | Pollution=high, Xray=positive)

```
python3.7 main.py -t Cancer=True -u Pollution=high Xray=positive -r 10 -c
0.001 -i 25000 -b cancer.bif
```

6. Pr(T=car | A=young, S=F)

```
python3.7 main.py -t T=car -u A=young S=F -r 10 -c 0.001 -i 25000 -b
survey.bif
```

7. Pr(T=train | A=young)

```
python3.7 main.py -t T=train -u A=young -r 10 -c 0.001 -i 25000 -b survey.bif
```

8. Pr(Accident=None | RiskAversion=Cautious, Age=Adult)

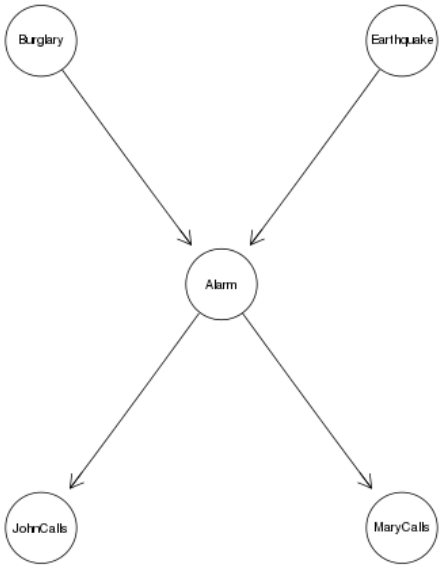
```
python3.7 main.py -t Accident=None -u RiskAversion=Cautious Age=Adult -r 10 -
c 0.001 -i 25000 -b insurance.bif
```

ATENÇÃO: Os nomes e valores das variáveis são *case sensitive*!!!

Redes bayesians testadas

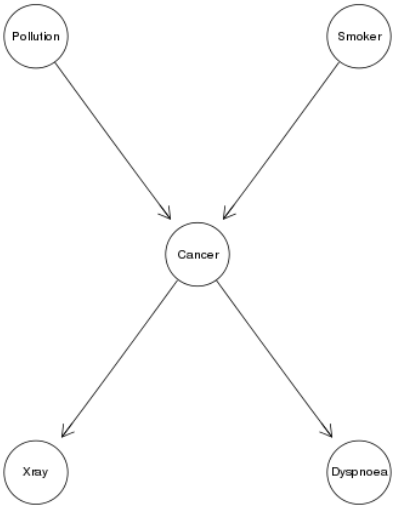
Durante o desenvolvimento foram utilizadas 4 redes bayesianas: earthquake, cancer, survey e asia.

Earthquake



Nesta rede, todos os campos têm valores True ou False.

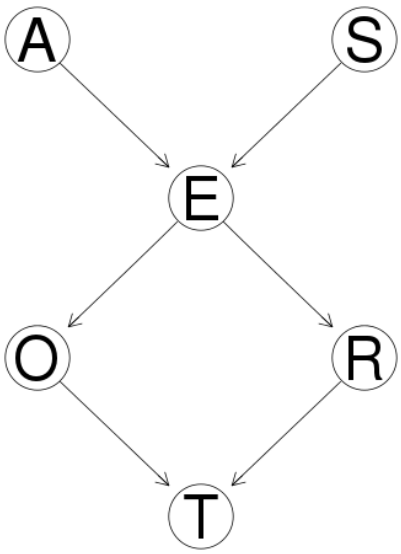
Cancer



Nesta rede, a dificuldade aumenta pois existem nós que não têm valores True ou False, como é o caso da variável Pollution (*high* ou *low)* e Xray (*positive* ou *negative*).

Dito isto, agora passou a ser necessário alguma forma de *encoding* de variáveis categóricas.

Survey



A -> “Idade”: (*young*, *adult* ou *old*)

S -> “Sexo”: (M ou F)

E -> “Educação”: (*high* ou *uni*)

O -> (desconhecido): (*emp* ou *self*)

R -> “Salário”: (*high* ou *low*)

T -> “Transporte”: (*train*, *car* ou *other*)

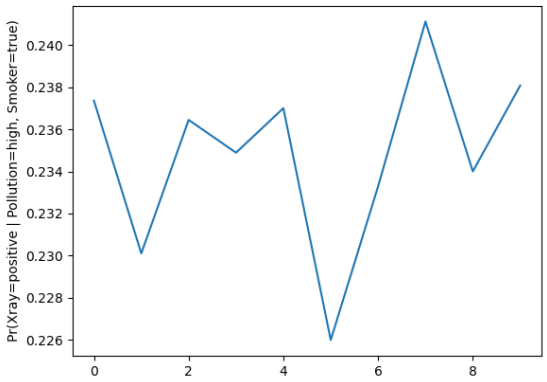
Agora, para além de termos vários valores categóricos diferentes na rede, as variáveis deixam de ser binárias, passado a poder assumir mais que dois valores.

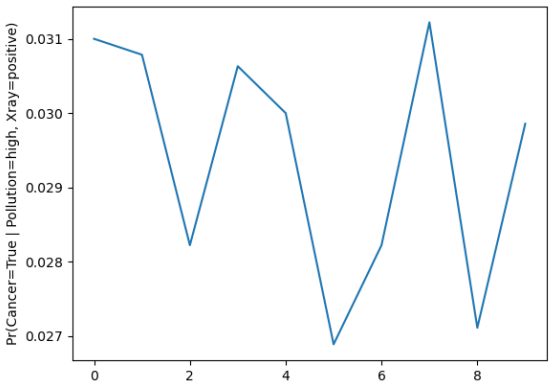
Insurance

Desta vez, observamos uma rede muito mais complexa com um elevado número de dependências.

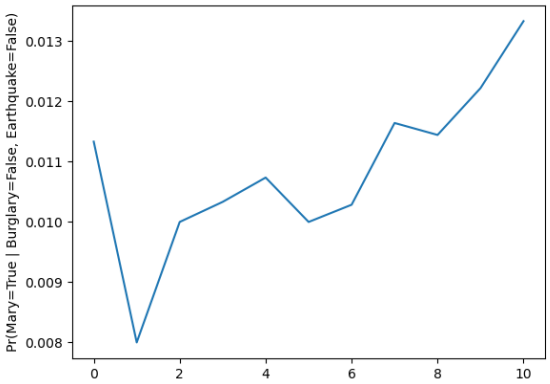
Alguns resultados obtidos

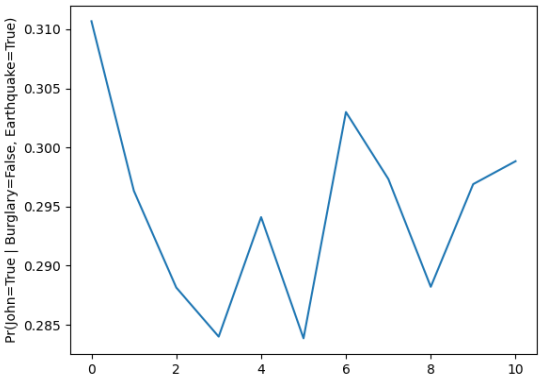
Cancer

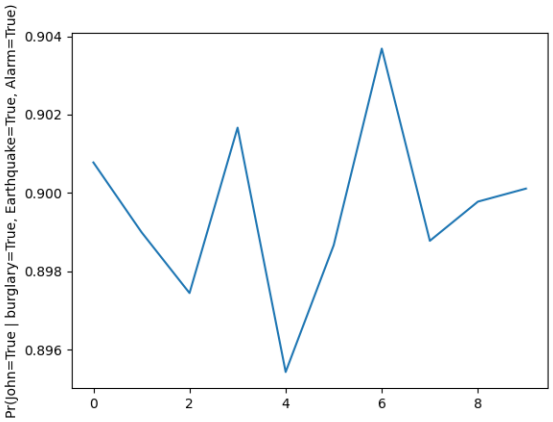




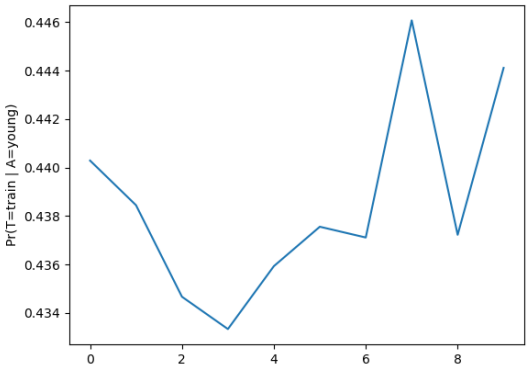
Earthquake

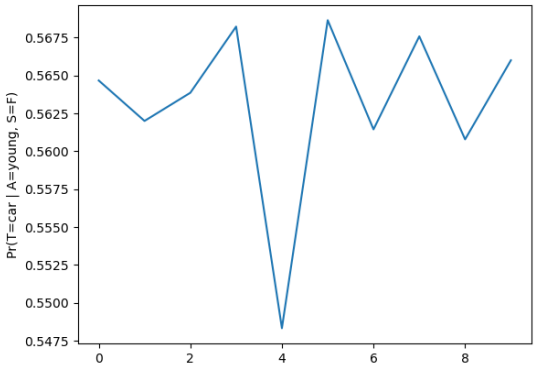




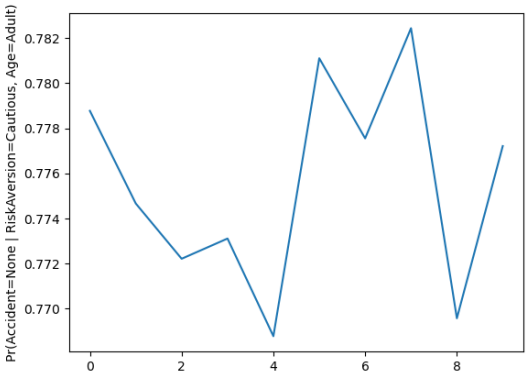


Survey





Insurance



Como se pode observar, em todos os casos a probabilidade obtida está sempre num intervalo de cerca de 1%, podendo assim ser assumido que o algoritmo está a convergir.