

Tarefa 2: Ler Redes Bayesianas, Escrever Programas Lógicos

Estado da Tarefa. Importação de Redes Bayesianas - OK; Construção de Programa Lógico a Partir de uma RB - Em Curso.

Importar uma Rede Bayesiana

Passos:

- Implementar
- Testar e Documentar
- Usar

Função `summary_dag(filename)` no módulo `bninput` . **Deve ser testada e documentada.**

Construir um Programa Lógico dada uma Rede Bayesiana

Passos:

- Implementar
- Testar e Documentar
- Usar

2023-07-20

O ficheiro `tarefa2.py` está **quase** adequado para esta tarefa. Em particular, tem código para converter a descrição de uma bn em *algo que se assemelha a um programa lógico*. No entanto:

Criar funções. À semelhança do que fez no `bninput` , deve **colocar o código "essencial" em funções**. Isto é, o essencial de

```

if __name__ == "__main__":
    summary = summary_dag("asia2.bif")
    model = summary["bnmodel"]
    probabilities = get_yes_probabilities(model)
    for node, yes_prob in probabilities.items():
        parents = model.get_parents(node)
        s = ""
        if len(parents) == 0:
...

```

deve ir para uma função. A minha sugestão é que o argumento dessa função seja um `model` que poderá resultar de, por exemplo, `summary_dag(...)`.

Adaptar a notação dos programas lógicos.

A sintaxe para os programas lógicos é a seguinte:

```

f.                /* Facto Determinista */
h :- b1, ..., bN. /* Regra Determinista */
p::f.            /* Facto Probabilístico */
p::h :- b1, ..., bN. /* Regra Probabilística */

```

em que `p` é uma probabilidade (um `float` entre 0 e 1); `f` é um "facto" (por exemplo, `asia`) e `h :- b1, ..., bN` é uma "regra" em que `h` é a "cabeça" ("*head*") e o "corpo" ("*body*") tem "literais" (factos ou negações de factos) `b1, ..., bN`. O símbolo `,` denota a *conjunção* (`()`), `-` a negação (`()`) e `:-` (em vez de `<-`, e lê-se "*if*" ou "*se*") denota `()`.

Além disso, em relação ao que o seu programa produz, cada regra e cada facto termina em `.`. Portanto, **falta acertar a sintaxe com a dos programas lógicos.**

Sintaxe, parte 2

Há, ainda, um aspeto adicional: Os programas que processam os programas lógicos não suportam (mais ou menos, em geral, por enquanto) factos e regras probabilísticas. Isso significa que a sintaxe

```

p::f.                /* Facto Probabilístico */
p::h :- b1, ..., bN. /* Regra Probabilística */

```

está "errada" para esses programas. O que podemos fazer, por enquanto, é escrever

```
%* p::f. *%  
f ; -f.  
%* p::h. *%  
h ; -h :- b1, ..., bN.
```

Por exemplo,

```
%* 0.01::asia. *%  
asia ; -asia.
```

em vez de

```
0.01::asia.
```

Nestes exemplos a sintaxe dos programas lógicos está acrescentada com " ; " para denotar a disjunção () e " %* ... *% " para blocos de comentários. Isto é,

```
%* 0.01::asia. *%  
asia ; -asia.
```

diz que temos um **facto disjuntivo**, `asia ; -asia` que indica que ou "acontece" `asia` ou "acontece" não `asia`. O comentário `%* 0.01::asia. *%` serve para "transportar" a informação sobre as probabilidades. Esta informação será tratada posteriormente, talvez na tarefa 4 ou na 5.