Inferenza con Junction Trees nei Belief Networks

Francesco Baiocchi

February 2022

1 Introduzione

In questa relazione vengono presentati i risultati ottenuti dall'utilizzo di un modulo software di Inferenza con Junction Trees nei Belief Networks su alcune reti di prova.
Il software presentato è raggiungibile al link `https://github.com/francescobaio/Elaborato_AI` ed è accompagnato dal file README.Inference.py per una descrizione dettagliata.
Il codice è testato su due reti presenti al link `https://github.com/ncullen93/pyBN/tree/master/data`, di cui è fornito il software per la grafica.
Sono inoltre presentate immagini raffiguranti le reti e i rispettivi Junction Tree calcolati manualmente.
Il nome dei nodi è per semplicità limitato alla sola iniziale.

2 Rete EarthQuake

La prima rete utilizzata è al link `https://github.com/ncullen93/pyBN/blob/master/data/earthquake.bn`.
Questa rete rappresenta la situazione in cui si vuole modellare contesto dove sono presenti 5 variabili che rappresentano eventi della vita reale come l'avvenimento di un terremoto,il suono di un allarme, la chiamata di un vicino o magari l'esecuzione di una rapina. Le variabili sono booleane e prendono gli interi che vanno da 1 a 5. B = 1, E = 2 ,A = 3 , J = 4, M = 5

2.1 Confronto Risultati Ottenuti

Il modulo software `https://github.com/francescobaio/Elaborato_AI/blob/main/Inference.py` presenta due possibilità per calcolare le probabilità condizionali $\mathbf{P}(\mathbf{Q}|\mathbf{e})$, con $\mathbf{Q} \in \mathbf{U}$ e $\mathbf{e} \subset \mathbf{U}$. Attraverso la classe Belief Network fornisce la possibilità di creare la tabella della probabilità congiunta della rete con il metodo *joint_probability()*.
E' possibile utilizzarlo solo per reti molto piccole infatti il numero di righe che

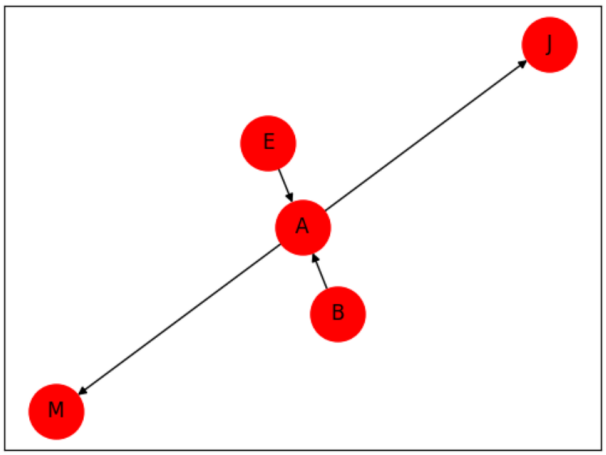

Figure 1: Belief Network del file earthquake

Con il modulo software graphics.py è possibile graficare la Belief Network ottenendo questo risultato,dove i nodi hanno come nome le iniziali dei veri nomi. Si può cosi anche graficare il Junction Tree associato utilizzando il metodo **junction_tree(G)** della libreria **"networkx"**

questa contiene è dato da tutte le possibili interpretazioni delle variabili.(Crescita Esponenziale). A questo punto per calcolare un la probabilità di una variabile data l'evidenza è necessario utilizzare il metodo *calculate_jp()*.

Attraverso il run del file `https://github.com/francescobaio/Elaborato_AI/blob/main/main.py` è possibile visualizzare i risultati relativi ai due metodi utilizzati. Qui viene presentata la tabella della probabilità congiunta della rete EarthQuake,si può verificare nell'ultima riga che questa somma a uno. Nel codice viene fatta evidenza del fatto che J sia 1,ovvero che John,il mio vicino mi ha chiamato e che M sia 0 ovvero che Mary ,l'altro vicino di casa non mi ha chiamato. Si calcola la probabilità che ,data questa evidenza, ci sia stata una rapina.

Utilizzando la definizione di probabilità condizionale,ovvero marginalizzando e poi normalizzando la full joint probability table si ottengono i risultati di Figura 3. Utilizzando il metodo del Junction Tree implementato è necessario dopo aver dato evidenza sulle CPTc dei cluster,scegliere una radice dell'albero.Scegliamo arbitrariamente "ABE" ,a quel punto l'informazione introdotta viene passata attraverso i metodo di *collect _evidence* e di *distribute _evidence*.Successivamente si trova un cluster che conteneva la variabile "B" e si marginalizza trovando la probabilità di B. Risultati in figura 4. Rinormalizzando questi risultati il risultato trovato è pressocchè simile,seppur diverso 89% rispetto al 95%.

```
[[1 2 3 4 5 0]
 [0 0 0 0 0 0.9115606269]
 [0 0 0 0 1 0.009207683099999999]
 [0 0 0 1 0 0.0479768751]
 [0 0 0 1 1 0.00048461489999999995]
 [0 0 1 0 0 2.9106e-05]
 [0 0 1 0 1 6.7914e-05]
 [0 0 1 1 0 0.000261954]
 [0 0 1 1 1 0.000611226]
 [0 1 0 0 0 0.013221548999999999]
 [0 1 0 0 1 0.000133551]
 [0 1 0 1 0 0.0006958710000000001]
 [0 1 0 1 1 7.029000000000002e-06]
 [0 1 1 0 0 0.00017225999999999998]
 [0 1 1 0 1 0.00040193999999999994]
 [0 1 1 1 0 0.00155034]
 [0 1 1 1 1 0.00361746]
 [1 0 0 0 0 0.0005530139999999999]
 [1 0 0 0 1 5.5859999999999995e-06]
 [1 0 0 1 0 2.9106e-05]
 [1 0 0 1 1 2.94e-07]
 [1 0 1 0 0 0.00027636]
 [1 0 1 0 1 0.00064484]
 [1 0 1 1 0 0.0024872399999999995]
 [1 0 1 1 1 0.005803559999999999]
 [1 1 0 0 0 9.405e-06]
 [1 1 0 0 1 9.5e-08]
 [1 1 0 1 0 4.95e-07]
 [1 1 0 1 1 5.000000000000001e-09]
 [1 1 1 0 0 5.7000000000000005e-06]

 [1 1 1 0 1 1.33e-05]
 [1 1 1 1 0 5.13e-05]
 [1 1 1 1 1 0.0001197]]
0.9999999999999997
```

Figure 2: Probabilità congiunta della Belief Network

```
[[1 4 5 0]
 [0 0 0 0]
 [0 0 1 0]
 [0 1 0 0.9515930817577307]
 [0 1 1 0]
 [1 0 0 0]
 [1 0 1 0]
 [1 1 0 0.04840691824226916]
 [1 1 1 0]]
[[1 0]
 [0 0.9515930817577307]
 [1 0.04840691824226916]]
```

Figure 3: Risultati con la definzione di probabilità condizionale

```
[[3 4 0]
 [0 0 0]
 [0 1 0.05]
 [1 0 0]
 [1 1 0.9]]
[[3 5 0]
 [0 0 0.99]
 [0 1 0]
 [1 0 0.3]
 [1 1 0]]
[[3 1 2 0]
 [0 0 0 0.9692297999999999]
 [0 0 1 0.014058000000000001]
 [0 1 0 0.000588]
 [0 1 1 1e-05]
 [1 0 0 0.0009702]
 [1 0 1 0.005742]
 [1 1 0 0.009212]
 [1 1 1 0.00019]]
[[1.         0.        ]
 [0.         0.05520537]
 [1.         0.0084917 ]]
```

Figure 4: Risultati con l'algoritmo del Junction Tree