

Relazione per l'esame di Intelligenza Artificiale

Lorenzo Giannella
5299652

1. Introduzione

Una rete Bayesiana è un modello probabilistico che permette di rappresentare relazioni e variabili stocastiche in un grafo aciclico diretto. Un esempio di utilizzo classico è l'applicazione di reti bayesiane per rappresentare le relazioni statistiche tra sintomi e malattie. Ad ogni nodo è associata una distribuzione di probabilità definita dentro un dominio. Le connessioni in una rete Bayesiana rappresentano la relazione di dipendenza tra le distribuzioni dei nodi connessi. Una CPT (conditional probability table) definisce dato un nodo il livello di confidenza che abbiamo nel verificarsi di un evento in relazione ai nodi genitori.

2. Scopo

L'elaborato riguarda l'inferenza su modelli probabilistici orientati. Lo scopo di questo elaborato è la realizzazione di un programma che, dopo aver costruito una rete Bayesiana, la analizza e ottiene le probabilità condizionali delle variabili data evidenza su un set di variabili.

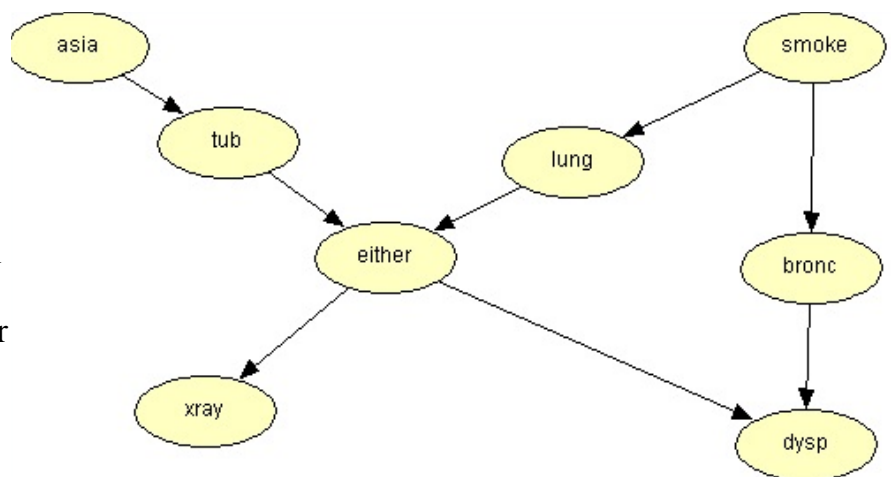
I risultati ottenuti sono poi confrontati con quelli risultanti dall'analisi della stessa rete da parte di Hugin Educational.

3. Strumenti di lavoro

Il linguaggio di programmazione usato è Python.

La realizzazione della rete è avvenuta grazie a una libreria di metodi utili per la creazione e analisi di una rete Bayesiana, fornita da BayesFusion LCC, il cui wrapper per python è chiamato pysmile.

La rete analizzata con Hugin Educational è asia.net ed è mostrata in figura.



4. Analisi del programma

La classe più importante definita da pysmile è Network. Gli oggetti contenuti in questa classe contengono i nodi e sono responsabili della loro creazione e distruzione. I nodi e gli archi sono sempre creati e distrutti invocando i metodi della classe Network. L'accesso ai nodi e agli archi è quindi sempre effettuato attraverso l'oggetto Network. Non ci sono classi che rappresentano i nodi o gli archi ma Network funziona da interfaccia.

Per creare il Network usiamo il costruttore di default: `net = pysmile.Network()`

I nodi sono creati grazie a `create_cpt_node(self, net, id, name, outcomes, x_pos, y_pos)`. La funzione crea un nodo CPT con uno specifico identificatore, nome, risultato e posizione nello schermo. I nodi CPT sono creati con due risultati chiamati stato0 e stato1. Per cambiare il numero dei risultati e rinominarli usiamo due loops, il primo rinomina i risultati di default e il secondo ne aggiunge di nuovi.

Per aggiungere o cancellare nodi usiamo i metodi:

```
net.add_node  
net.delete_node
```

Per iterare sui nodi nella rete usiamo:

```
net.get_first_node  
net.get_next_node
```

I metodi per ottenere un array di nodi, identificatori o numero di nodi:

```
net.get_all_nodes  
net.get_all_node_ids  
net.get_node_count
```

Per aggiungere gli archi che collegano i nodi usiamo `net.addArc` usando gli handles dei nodi o gli identifiers.

La creazione delle tabelle di probabilità condizionata:

```
tubDef = [  
    0.05, # P(tub=Y|asia=Y)  
    0.95, # P(tub=N|asia=Y)  
    0.01, # P(tub=Y|asia=N)  
    0.99 # P(tub=N|asia=N)  
]  
net.set_node_definition(t, tubDef);
```

`net.set_node_definition` specifica come il nodo interagisce con gli altri nella rete.

Consiste nella CPT (conditional probability table) e nella lista degli stati.

Il value del nodo contiene i valori (tipicamente la distribuzione marginale di probabilità) calcolati dall'algoritmo di inferenza.

L'Update delle probabilita' avviene attraverso l'helper method `print_all_posteriors` che le mostra su schermo. Per far partire l'inferenza e ottenere la probabilita', date le evidenze impostate nella rete, usiamo il metodo `net.update_beliefs`.

Il metodo `print_posteriors` controlla se il nodo ha evidenza chiamando `net.is_evidence`, in caso positivo il nome dell' evidenza viene mostrato.

`print_posteriors` itera su tutti gli stati e mostra la posterior probability di ognuno.

Utilizziamo più volte `change_evidence_and_update` per cambiare le evidenze, aggiornare il network e mostrare i posteriors.

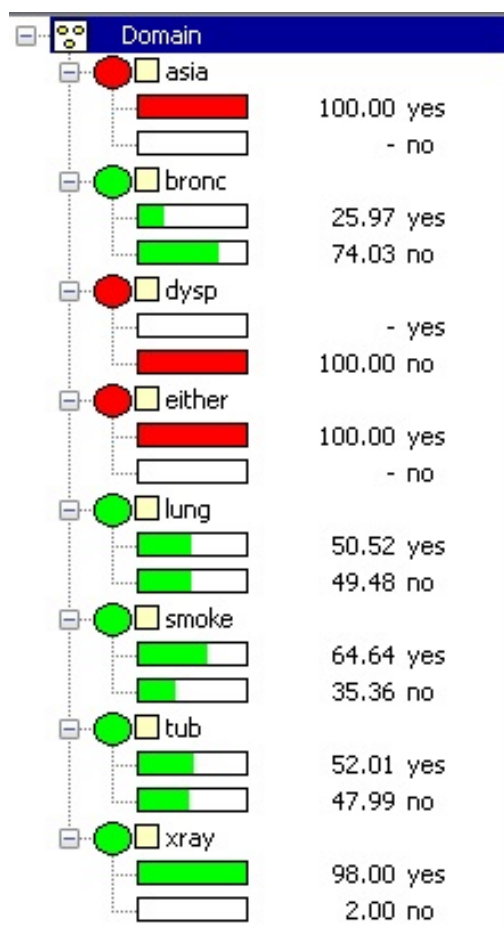
Il metodo `print_all_posteriors` mostra le posterior probabilities calcolate da `update_beliefs` per ogni nodo. Per iterare sui nodi sono usati `net.get_first_node` e `net.get_next_node`.

Le probabilita' dei nodi sono mostrate da `print_cpt_matrix`. Dall'array prendiamo le probabilita' con `get_node_definition` e le traduciamo in un array multidimensionale (grazie al metodo `index_to_coords`).

5. Risultati

L'output del programma è stato confrontato con l'analisi della rete eseguita da Hugin Educational evidenziando gli stessi risultati, come mostrato in figura.

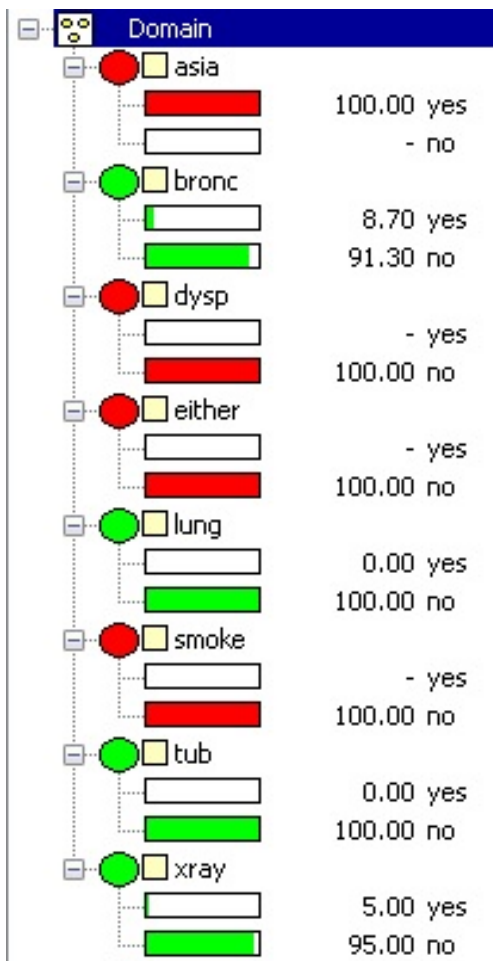
Le evidenze vengono applicate una alla volta, l'ultimo blocco rappresenta il risultato confrontato con Hugin.



```
Setting Either=Yes, Asia=Yes and Dyspnea=No
P(asia=Yes)=0.0157725057074
P(asia=No)=0.984227494293
P(tub=Yes)=0.160424507929
P(tub=No)=0.839575492071
either has evidence set <Yes>
P(xray=Yes)=0.98
P(xray=No)=0.02
P(lung=Yes)=0.848398840007
P(lung=No)=0.151601159993
P(smoke=Yes)=0.843462701302
P(smoke=No)=0.156537298698
P(bronc=Yes)=0.553038810391
P(bronc=No)=0.446961189609
P(dysp=Yes)=0.810607762078
P(dysp=No)=0.189392237922

asia has evidence set <Yes>
P(tub=Yes)=0.488997555012
P(tub=No)=0.511002444988
either has evidence set <Yes>
P(xray=Yes)=0.98
P(xray=No)=0.02
P(lung=Yes)=0.537897310513
P(lung=No)=0.462102689487
P(smoke=Yes)=0.709046454768
P(smoke=No)=0.290953545232
P(bronc=Yes)=0.51271393643
P(bronc=No)=0.48728606357
P(dysp=Yes)=0.802542787286
P(dysp=No)=0.197457212714

asia has evidence set <Yes>
P(tub=Yes)=0.520059435364
P(tub=No)=0.479940564636
either has evidence set <Yes>
P(xray=Yes)=0.98
P(xray=No)=0.02
P(lung=Yes)=0.505200594354
P(lung=No)=0.494799405646
P(smoke=Yes)=0.646359583952
P(smoke=No)=0.353640416048
P(bronc=Yes)=0.259658246657
P(bronc=No)=0.740341753343
dysp has evidence set <No>
```

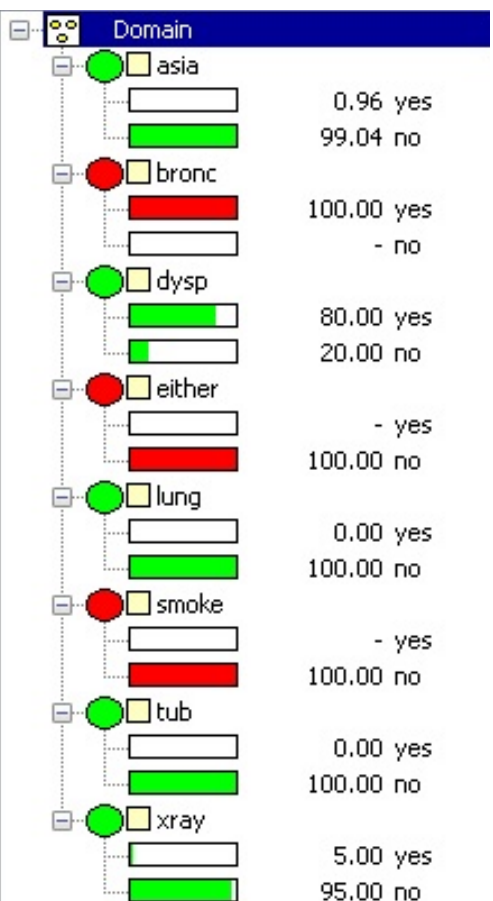


```

C:\Python27\python.exe
Setting Smoke=No, changing Either to No, keeping Asia=Yes and Dyspnea=No
asia has evidence set (Yes)
P(tub=Yes)=0.840336134454
P(tub=No)=0.159663865546
either has evidence set (Yes)
P(xray=Yes)=0.98
P(xray=No)=0.02
P(lung=Yes)=0.168067226891
P(lung=No)=0.831932773109
smoke has evidence set (No)
P(bronc=Yes)=0.125
P(bronc=No)=0.875
dysp has evidence set (No)

asia has evidence set (Yes)
P(tub=Yes)=0.0
P(tub=No)=1.0
either has evidence set (No)
P(xray=Yes)=0.05
P(xray=No)=0.95
P(lung=Yes)=0.0
P(lung=No)=1.0
smoke has evidence set (No)
P(bronc=Yes)=0.0869565217391
P(bronc=No)=0.913043478261
dysp has evidence set (No)

```



```

Removing evidence from Asia and Dyspnea, setting Bronchitis=Yes, keeping Either=
No and Smoke=No
P(asia=Yes)=0.00959983831851
P(asia=No)=0.990400161681
P(tub=Yes)=0.0
P(tub=No)=1.0
either has evidence set (No)
P(xray=Yes)=0.05
P(xray=No)=0.95
P(lung=Yes)=0.0
P(lung=No)=1.0
smoke has evidence set (No)
P(bronc=Yes)=0.0869565217391
P(bronc=No)=0.913043478261
dysp has evidence set (No)

P(asia=Yes)=0.00959983831851
P(asia=No)=0.990400161681
P(tub=Yes)=0.0
P(tub=No)=1.0
either has evidence set (No)
P(xray=Yes)=0.05
P(xray=No)=0.95
P(lung=Yes)=0.0
P(lung=No)=1.0
smoke has evidence set (No)
P(bronc=Yes)=0.3
P(bronc=No)=0.7
P(dysp=Yes)=0.31
P(dysp=No)=0.69

P(asia=Yes)=0.00959983831851
P(asia=No)=0.990400161681
P(tub=Yes)=0.0
P(tub=No)=1.0
either has evidence set (No)
P(xray=Yes)=0.05
P(xray=No)=0.95
P(lung=Yes)=0.0
P(lung=No)=1.0
smoke has evidence set (No)
bronc has evidence set (Yes)
P(dysp=Yes)=0.8
P(dysp=No)=0.2

```