Andrea Augello
Department of Engineering, University of Palermo, Italy

Reti Bayesiane 1

Università
degli Studi
di Palermo

Una rete bayesiana è un modello grafico probabilistico che rappresenta un insieme di variabili stocastiche con le loro dipendenze condizionali attraverso l'uso di un grafo aciclico diretto (DAG)

- I nodi del grafo rappresentano le variabili stocastiche
- Gli archi rappresentano le relazioni di dipendenza statistica tra le variabili e le distribuzioni locali di probabilità dei nodi figlio rispetto ai valori dei nodi padre.

bnlearn

bnlearn

- `bnlearn` è un pacchetto Python per l'apprendimento di reti bayesiane
- Consente di apprendere sia i parametri che la struttura di una rete bayesiana
  - Imparare i parametri di una rete bayesiana significa apprendere le distribuzioni di probabilità condizionale dei nodi figlio rispetto ai valori dei nodi padre
  - Imparare la struttura di una rete bayesiana significa apprendere le relazioni di dipendenza statistica tra le variabili stocastiche

https://erdogant.github.io/bnlearn/pages/html/index.html

Esempio 0

Esempio di Naive Bayes

Consideriamo un esempio di classificazione con il modello Naive Bayes per determinare se si giocherà una partita di tennis.

- Supponiamo di avere due feature: `Pioggia` e `Vento`
- La variabile target è `Giocare`
- Vogliamo calcolare la probabilità di `Giocare` dato `Pioggia` e `Vento`

Esempio di Naive Bayes

Consideriamo un esempio di classificazione con il modello Naive Bayes per determinare se si giocherà una partita di tennis.

- Supponiamo di avere due feature: `Pioggia` e `Vento`
- La variabile target è `Giocare`
- Vogliamo calcolare la probabilità di `Giocare` dato `Pioggia` e `Vento`

Dati noti:

- Il 90% delle partite programmate vengono giocate regolarmente
- Se la partita è saltata, c'è una probabilità del 70% che quel giorno piovesse. Se invece la partita si è giocata, c'è una probabilità del 60% che quel giorno ci fosse il sole.
- Nel 15% delle partite saltate c'era vento, mentre solo nel 5% delle partite giocate c'era vento.

Definizione del modello

Definiamo il modello Naive Bayes in Python utilizzando la libreria `bnlearn`.

```
import bnlearn as bn

# Definiamo la struttura del grafo
edges = [('Giocare', 'Pioggia'), ('Giocare', 'Vento')]
DAG = bn.make_DAG(edges)
bn.plot(DAG)
```

Nota bene: nei modelli Naive Bayes, tutti gli archi partono dal nodo target e arrivano ai nodi delle feature.

Aggiungere le distribuzioni di probabilità

- Il 90% delle partite programmate vengono giocate regolarmente
- Se la partita è saltata, c'è una probabilità del 70% che quel giorno piovesse. Se invece la partita si è giocata, c'è una probabilità del 60% che quel giorno ci fosse il sole.
- Nel 15% delle partite saltate c'era vento, mentre solo nel 5% delle partite giocate c'era vento.

Ovvero:

- $P(Giocare = 1) = 0.9$
- $P(Pioggia = 1|Giocare = 0) = 0.7$
- $P(Pioggia = 0|Giocare = 1) = 0.6$
- $P(Vento = 1|Giocare = 0) = 0.15$
- $P(Vento = 1|Giocare = 1) = 0.05$

Aggiungere le distribuzioni di probabilità

$P(Giocare = 0)$	$P(Giocare = 1)$
	0.9

$P(Pioggia = 0 \mid Giocare)$	$P(Pioggia = 1 \mid Giocare)$	$Giocare$
	0.7	0
0.6		1

$P(Vento = 0 \mid Giocare)$	$P(Vento = 1 \mid Giocare)$	$Giocare$
	0.15	0
	0.05	1

Aggiungere le distribuzioni di probabilità

$P(Giocare = 0)$	$P(Giocare = 1)$
0.1	0.9

$P(Pioggia = 0 \mid Giocare)$	$P(Pioggia = 1 \mid Giocare)$	$Giocare$
0.3	0.7	0
0.6	0.4	1

$P(Vento = 0 \mid Giocare)$	$P(Vento = 1 \mid Giocare)$	$Giocare$
0.85	0.15	0
0.95	0.05	1

Aggiungere le distribuzioni di probabilità

Specifichiamo manualmente le tabelle delle distribuzioni di probabilità condizionale per ogni nodo figlio rispetto ai valori dei nodi padre.

```
from pgmpy.factors.discrete import TabularCPD

cpt_giocare = TabularCPD(variable='Giocare', variable_card=2,
    values=[[0.1], [0.9]])
cpt_pioggia = TabularCPD(variable='Pioggia', variable_card=2,
    evidence=['Giocare'], evidence_card=[2], values=[[0.3, 0.6],
    [0.7, 0.4]])
cpt_vento = TabularCPD(variable='Vento', variable_card=2,
    evidence=['Giocare'], evidence_card=[2], values=[[0.85 ,0.95],
    [0.15, 0.05]])

DAG = bn.make_DAG(edges, CPD=[cpt_giocare, cpt_pioggia, cpt_vento])
```

Inferenza sul modello

Per fare inferenza sul modello, dobbiamo specificare le evidenze e quali variabili vogliamo calcolare.

- Esempio: $P(Giocare|Pioggia = 1, Vento = 0)$

```
result = bn.inference.fit(DAG, variables=['Giocare'],
    evidence={'Pioggia': 1, 'Vento': 0})
print(result)
```

```
+----+-----------+----------+
|    |   Giocare |        p |
+====+===========+==========+
|  0 |         0 | 0.148194 |
+----+-----------+----------+
|  1 |         1 | 0.851806 |
+----+-----------+----------+
```

Risultati dell'inferenza

I risultati dell'inferenza mostrano la probabilità della variabile target dato le evidenze specificate.

- $P(Giocare = 0 | Pioggia = 1, Vento = 0) = 0.15$
- $P(Giocare = 1 | Pioggia = 1, Vento = 0) = 0.85$

Esempio 1

Costruire un grafo

Se si possiede già conoscenza a priori sulla struttura della rete bayesiana (o se stiamo usando un modello naive bayes), è possibile costruire il grafo a mano.

Esempio

Braccio robotico che tenta di sollevare un blocco:

Abbiamo un braccio robotico che tenta di muovere un blocco. Il robot non può sollevare il blocco se la batteria è scarica. Per conoscere lo stato della batteria, abbiamo a disposizione un indicatore non troppo affidabile. Alcuni blocchi sono troppo pesanti per essere sollevati e, in questi casi, è più difficile che il robot riesca a spostarli.

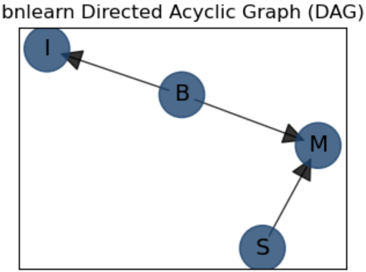
Variabili booleane:

- M = c'è movimento
- B = la batteria è carica
- S = il blocco è sollevabile
- I = l'indicatore è acceso

Costruire un grafo

Per prima cosa dobbiamo definire le relazioni uno-ad-uno tra le variabili. Definiamo gli archi:

- $S \rightarrow M$
- $B \rightarrow M$
- $B \rightarrow I$




```
import bnlearn as bn
from pgmpy.factors.discrete import
    TabularCPD

# Define the structure.

edges = [('S','M'), ('B','M'), ('B','I')]
DAG = bn.make_DAG(edges)
bn.plot(DAG)
```

Aggiungere le distribuzioni di probabilità

Al momento il grafo non contiene nessuna distribuzione condizionale di probabilità, quindi non può essere usato per fare inferenza.
Specifichiamo manualmente le tabelle delle distribuzioni di probabilità condizionale per ogni nodo figlio rispetto ai valori dei nodi padre.

$p(I\|\neg B)$	0.1
$p(I\|B)$	0.95

$p(M\|\neg B, \neg S)$	0
$p(M\|\neg B, S)$	0
$p(M\|B, \neg S)$	0.05
$p(M\|B, S)$	0.9

$p(S)$	0.95

$p(B)$	0.9

Aggiungere le distribuzioni di probabilità

Al momento il grafo non contiene nessuna distribuzione condizionale di probabilità, quindi non può essere usato per fare inferenza.
Specifichiamo manualmente le tabelle delle distribuzioni di probabilità condizionale per ogni nodo figlio rispetto ai valori dei nodi padre.

$p(I \mid \neg B)$	0.1
$p(I \mid B)$	0.95

$p(M \mid \neg B, \neg S)$	0
$p(M \mid \neg B, S)$	0
$p(M \mid B, \neg S)$	0.05
$p(M \mid B, S)$	0.9

$p(S)$	0.95

$p(B)$	0.9

```
cpt_i = TabularCPD(variable='I', variable_card=2,
                   values=[[0.9, 0.05],
                           [0.1, 0.95]],
                   evidence=['B'], evidence_card=[2])

cpt_m = TabularCPD(variable='M', variable_card=2,
                   values=[[1, 1, 0.95, 0.1],
                           [0, 0, 0.05, 0.9]],
                   evidence=['B', 'S'], evidence_card=[2, 2])

cpt_s = TabularCPD(variable='S', variable_card=2,
                   values=[[0.05],[0.95]])
cpt_b = TabularCPD(variable='B', variable_card=2,
                   values=[[0.1],[0.9]])
```

Inferenza sul grafo

Per fare inferenza sul grafo, dobbiamo specificare le evidenze e quali variabili vogliamo calcolare.

Esempi di inferenza:

$p(M\|I) = 0.85$	`bn.inference.fit(DAG, variables=['M'], evidence={'I':1})`
$p(M\|B) = 0.86$	`bn.inference.fit(DAG, variables=['M'], evidence={'B':1})`
$p(M\|S) = 0.81$	`bn.inference.fit(DAG, variables=['M'], evidence={'S':1})`
$p(M\|S, I) = 0.89$	`bn.inference.fit(DAG, variables=['M'], evidence={'S':1, 'I':1})`
$p(M\|S, \neg I, B) = 0.90$	`bn.inference.fit(DAG, variables=['M'], evidence={'S':1, 'I':0, 'B':1})`

Esempio 2

Esempio 2

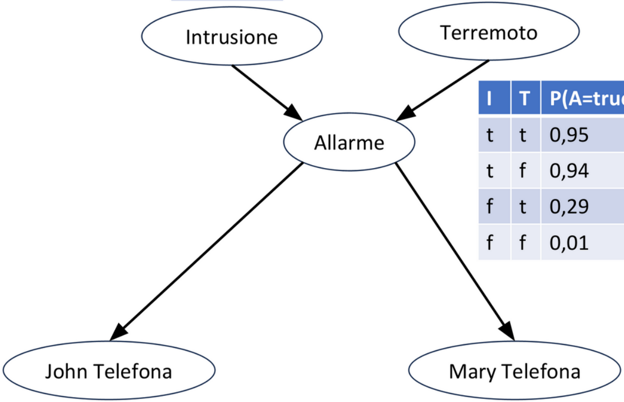
- Abbiamo installato un nuovo antifurto, che è abbastanza affidabile ma occasionalmente scatta anche per piccoli terremoti
- Abbiamo due vicini, John e Mary, che hanno promesso di telefonare se scatta l'allarme
- John chiama quasi sempre quando scatta l'allarme, ma a volte scambia lo squillo del telefono per l'allarme
- Mary non sempre sente l'allarme

Variabili: `['Intrusione', 'Terremoto', 'Allarme', 'John', 'Mary']`
Inferenza: `bn.inference.fit(DAG, variables=['Intrusione'], evidence={'John':1, 'Mary':0})`

Esempio 2

P(I=true)
0,001

P(T=true)
0,002



I	T	P(A=true | I, T)
t	t	0,95
t	f	0,94
f	t	0,29
f	f	0,01

A	P(J=true | A)
t	0,90
f	0,05

A	P(M=true | A)
t	0,70
f	0,01

Soluzione

```
edges = [('Intrusione', 'Allarme'), ('Terremoto', 'Allarme'), ('Allarme', 'John'),
     ('Allarme', 'Mary')]

cpt_intrusione = TabularCPD(variable='Intrusione', variable_card=2,
                            values=[[0.999], [0.001]])
cpt_terremoto  = TabularCPD(variable='Terremoto', variable_card=2,
                            values=[[0.998], [0.002]])

cpt_allarme    = TabularCPD(variable='Allarme', variable_card=2,
                            values=[[0.999, 0.71, 0.06, 0.05],
                                    [0.001, 0.29, 0.94, 0.95]],
                            evidence=['Intrusione', 'Terremoto'],
                            evidence_card=[2, 2])

cpt_john       = TabularCPD(variable='John', variable_card=2,
                            values=[[0.95, 0.1],
                                    [0.05, 0.9]],
                            evidence=['Allarme'], evidence_card=[2])

cpt_mary       = TabularCPD(variable='Mary', variable_card=2,
                            values=[[0.99, 0.3],
                                    [0.01, 0.7]],
                            evidence=['Allarme'], evidence_card=[2])

dag = bn.make_DAG(edges, CPD=[cpt_intrusione, cpt_terremoto, cpt_allarme, cpt_john,
     cpt_mary])
```

Esempio 3

Esempio 3

Si implementi in Python una rete bayesiana appropriata date le seguenti informazioni:

- La rete ha come obiettivo determinare la probabilità che vengano fatti degli acquisti nella sessione corrente su un sito di e-commerce.
- Se l'utente è collegato da un dispositivo mobile (35% degli utenti), la probabilità di acquisto è del 10%, indipendentemente da tutti gli altri fattori.
- Se l'utente è collegato da un dispositivo desktop, se ha già effettuato un acquisto in passato la probabilità di acquisto è del 50%, altrimenti è del 20%.
- Sappiamo inoltre che la probabilità che un utente abbia effettuato un acquisto in passato è influenzata dal fatto che l'utente sia stato il target di una <u>campagna pubblicitaria</u> e che l'utente sia <u>suggestionabile</u>:
  - Il 40% degli utenti suggestionabili che sono stati target di una campagna pubblicitaria ha effettuato un acquisto in passato.
  - Il 10% degli utenti che non sono stati target di una campagna pubblicitaria o che non sono suggestionabili ha effettuato un acquisto in passato.
  - La campagna ha raggiunto il 3% degli utenti.
  - Un'indagine di mercato ha rivelato che il 70% degli utenti è suggestionabile.

Modellazione del problema

- Abbiamo 5 variabili: `['Acquisto', 'Dispositivo', 'AcquistoPassato', 'Campagna', 'Suggestionabile']`

Modellazione del problema

- Abbiamo 5 variabili: [`'Acquisto'`, `'Dispositivo'`, `'AcquistoPassato'`, `'Campagna'`, `'Suggestionabile'`]
- `'Acquisto'` dipende da `'Dispositivo'` e `'AcquistoPassato'`

Modellazione del problema

- Abbiamo 5 variabili: [`'Acquisto'`, `'Dispositivo'`, `'AcquistoPassato'`, `'Campagna'`, `'Suggestionabile'`]
- `'Acquisto'` dipende da `'Dispositivo'` e `'AcquistoPassato'`
- Conosciamo la probabilità della variabile `'Dispositivo'`

Modellazione del problema

- Abbiamo 5 variabili: [`'Acquisto'`, `'Dispositivo'`, `'AcquistoPassato'`, `'Campagna'`, `'Suggestionabile'`]
- `'Acquisto'` dipende da `'Dispositivo'` e `'AcquistoPassato'`
- Conosciamo la probabilità della variabile `'Dispositivo'`
- `'AcquistoPassato'` invece dipende da `'Campagna'` e `'Suggestionabile'`

Modellazione del problema

- Abbiamo 5 variabili: `['Acquisto', 'Dispositivo', 'AcquistoPassato', 'Campagna', 'Suggestionabile']`
- `'Acquisto'` dipende da `'Dispositivo'` e `'AcquistoPassato'`
- Conosciamo la probabilità della variabile `'Dispositivo'`
- `'AcquistoPassato'` invece dipende da `'Campagna'` e `'Suggestionabile'`
- Per entrambe le variabili abbiamo informazioni sulla loro probabilità