

Relazione per l'esame di Intelligenza Artificiale

Alessio Chen

1 Introduzione

In questo elaborato si è fatto uso dell'algoritmo K2 per l'apprendimento di una struttura bayesiana attraverso il linguaggio python.

Una rete Bayesiana è un modello grafico di probabilità in cui i nodi rappresentano variabili e gli archi le dipendenze casuali fra le variabili. Una rete bayesiana è dunque un modello grafico che codifica la distribuzione congiunta di probabilità di un insieme di variabili aleatorie $X = X_1, \dots, X_n$. Questa consiste:

1. un grafico diretto aciclico G , in cui ogni nodo è associato ad un'unica variabile aleatoria X_i e ogni arco rappresenta la dipendenza condizionale fra i nodi che unisce;
2. un insieme P di distribuzione locali di probabilità, ciascuna associata a una variabile aleatoria X_i e condizionata dalle variabili corrispondenti ai nodi sorgenti degli archi entranti nel nodo a cui è associata X_i .

La mancanza di un arco tra due nodi riflette la loro indipendenza condizionale. Al contrario, la presenza di un arco dal nodo X_i al nodo X_j può essere interpretato come il fatto che X_i sia causa diretta di X_j . Dato un grafo G e le distribuzioni locali di probabilità di ciascun nodo $P(X_i|P_{a_i})$, dove P_{a_i} rappresenta l'insieme di nodi padri di X_i , la probabilità di distribuzione congiunta $P(X)$ si ottiene da :

$$P(X) = \prod_{i=1}^n P(X_i|P_{a_i})$$

è evidente come la coppia (G,P) codifichi in modo univoco $P(X)$

2 Apprendimento di una rete Bayesiana

Il processo di apprendimento di reti bayesiane comprende:

- **apprendimento della struttura:** apprendere la struttura della rete ovvero le relazioni fra le variabili;
- **apprendimento dei parametri:** apprendere i parametri , cioè apprendimento delle probabilità condizionate.

2.1 Apprendimento della struttura

Dato un dataset l'obiettivo dell'apprendimento della struttura è trovare la struttura che lo descriva al meglio. In altre parole l'intento è quello di esplicitare dalle osservazioni su un insieme di variabili (dominio), "cosa è connesso a cosa", cioè individuare le relazioni fra le entità del dominio. I metodi per l'apprendimento della struttura possono essere divisi in due filoni principali:

- **Search-score :** si usano funzioni score (punteggi) che permettono di confrontare l'adeguatezza tra le possibili strutture di una rete;
- **Constraint-based:** in questo caso si utilizzano delle misure per indagare l'esistenza di eventuali indipendenze condizionali tra le variabili aleatorie.

2.2 La rete ASIA

La rete utilizzata presa in esame in questo progetto è una semplice rete di Lauritzen e Spiegelhalter (1988) sulle malattie polmonari (tubercolosi, cancro ai polmoni o bronchite) e visite in Asia. La sua struttura è sostanzialmente basata su un DAG, ovvero un grado aciclico diretto: possiede 8 nodi e 8 archi e ogni nodo rappresenta una variabile casuale, in pratica un evento, con relativa probabilità che questo si verifichi o meno. Ogni arco descrive una sorta di condizione di dipendenza: un nodo figlio possiede la possibilità che l'evento si verifichi in base all'accadere o meno degli eventi rappresentati dai nodi padri, invece i nodi non connessi rappresentano gli eventi che sono condizionalmente indipendenti dagli altri.

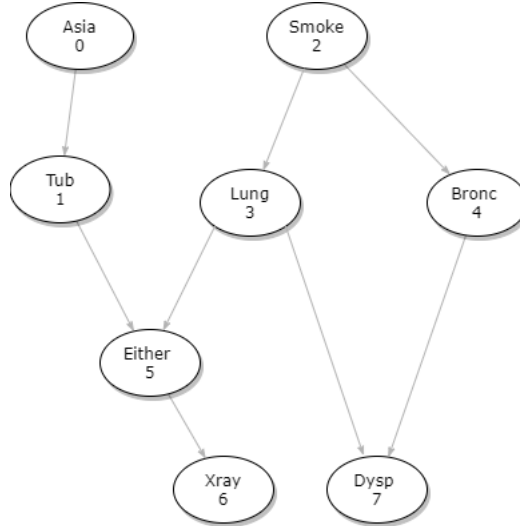


Figura 1: Grafo della rete “Asia”

2.3 Algoritmo K2

L'algoritmo K2 è un algoritmo score-based che effettua una ricerca greedy incrementale per l'apprendimento della struttura. Dato un dataset, inizialmente si parte con un grafo vuoto (tutti i nodi sono senza padre). Dopodiché dato l'ordinamento $\langle x_1, \dots, x_n \rangle$, l'insieme dei padri di ogni nodo x_i è determinato valutando se ogni variabile dell'insieme dei relativi predecessori, $pred(X_i) = \{X_1, \dots, X_{i-1}\}$ possa essere padre di X_i , ovvero quale $x_j \in pred(X_i)$ massimizza la metrica K2 espressa dalla funzione di score. La fase di ricerca, per l'insieme dei padri di X_i termina dopo aver esaminato tutte le variabili in $pred(X_i)$ o se viene raggiunto una soglia sul massimo numero di padri (indicata a priori). La natura greedy dell'algoritmo è rappresentata dal considerare come possibile padre ogni nodo dell'insieme $\{X_1, \dots, X_{n-1}\}$. L'esaminare un nodo alla volta, anche come probabile padre, e l'iterare questa procedura per tutti i nodi del dominio rivela la natura incrementale della fase di ricerca.

Uno svantaggio di tale approccio è il fatto che non è possibile “tornare indietro” dopo l'aggiunta di un arco.

Un altro svantaggio di questo algoritmo è la necessità di designare un corretto ordinamento dei nodi il che in assenza di informazioni a priori, non è semplice: l'ordine scelto influenza sia il risultato che la qualità della rete finale.

3 Procedimento

Inizialmente si prende in input una rete già nota, in questo caso una rete chiamata “ASIA” (Figura 1). Ogni nodo è codificato con il nome, lista dei padri, lista delle probabilità condizionate, il suo dominio (lista di valori che può assumere) e altri attributi che servono per l'algoritmo DFS. Mentre la rete è rappresentata attraverso una classe Python contenente la lista dei nodi e una matrice di adiacenza.

Una volta preso in input la rete si passa alla creazione del dataset, che viene creato sulla base della rete presa in input, il tutto viene salvato in una classe Python che oltre a contenere il dataset contiene anche

un ordine dei nodi che viene fornito dall'algoritmo di ricerca topologico DFS, questo serve per garantire l'ipotesi che si conosca a priori un ordine dei nodi per poter applicare poi l'algoritmo K2.

A questo punto si passa alla fase di apprendimento, la quale consiste in 1000 prove ripetute.

In ogni prova si genera un dataset (a partire dalla rete "ASIA") differente da 150 riga ciascuna e si applica l'algoritmo di apprendimento K2 su di esso, sommando i risultati via via ottenuti in una matrice, che ci rappresenterà con quale probabilità ogni arco può comparire nella rete.

4 Risultati

Di seguito sono riportate la matrice di adiacenza della rete originale (Figura 2) e la matrice risultato prodotto dai test (Figura 3)

	0	1	2	3	4	5	6	7
0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0
2	0	0	0	1	1	0	0	0
3	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Figura 2: Matrice di adiacenza della rete "ASIA" di cui si vuole fare l'apprendimento.

```
[
  [ 0. 281. 0. 0. 0. 0. 242. 64. ]
  [ 0. 0. 0. 0. 0. 976. 200. 0. ]
  [ 2. 0. 0. 615. 975. 0. 3. 15. ]
  [ 44. 41. 0. 0. 0. 1000. 25. 175. ]
  [ 1. 0. 0. 28. 0. 0. 4. 1000. ]
  [ 0. 0. 0. 0. 0. 0. 997. 644. ]
  [ 0. 0. 0. 0. 0. 0. 0. 0. ]
  [ 0. 0. 0. 0. 0. 0. 15. 0. ]]
```

Figura 3: Risultato ottenuto dai test.

5 Conclusioni

In conclusione, vediamo che la rete appresa dal dataset si avvicina molto alla rete originale, abbiamo solo un arco che viene appreso con una probabilità piccola per via della natura greedy dell'algoritmo utilizzato.