

Relazione per l'esame di Intelligenza Artificiale

Alessio Chen

1 Introduzione

Nella seguente relazione viene descritto il funzionamento dell'elaborato che è stato sviluppato per l'esame. In particolare, l'elaborato inerente l'apprendimento della struttura di reti Bayesiane utilizzando l'approccio descritto in (**Heckerman 1997**). Inizialmente, viene prodotto un data set di dimensione fissa, composta da n righe e da un numero di colonne corrispondenti al numero di nodi presenti nella rete, considerando le probabilità condizionate che sono state prese dalla rete utilizzate come esempio. Il passo successivo è quello di effettuare l'apprendimento della struttura sul data set creato, attraverso l'uso dell'algoritmo K2. Infine viene fatto un confronto con la rete originale e quella appresa.

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. La struttura della rete è rappresentata nella **Figura 1**.

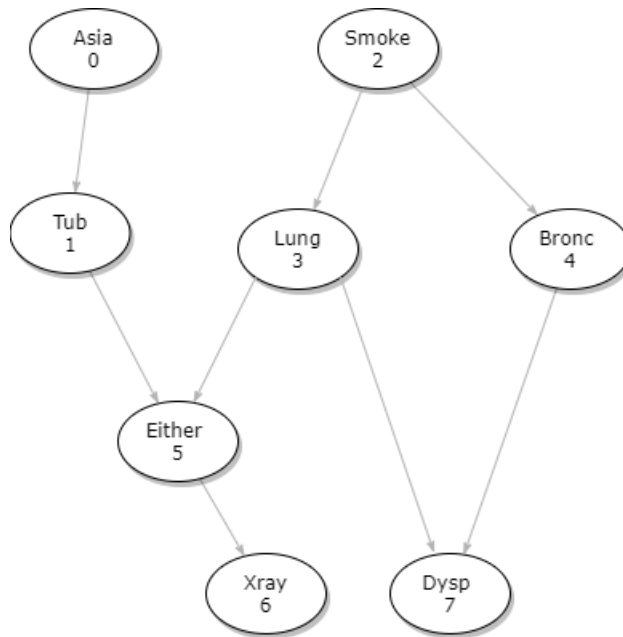


Figura 1: Grafo della rete "Asia"

3 Generazione dataset

La generazione del dataset viene divisa in due casistiche di nodi: i nodi che non possiedono uno o più nodi genitore e i nodi senza genitori.

I nodi senza genitori sono più facili da considerare per la generazione del dataset: si confronta semplicemente la probabilità del nodo con un numero generato in maniera random da 0 a 1, inserendo uno dei due estremi nel data set in base all'esito del confronto (0 o 1 se l'esito ha avuto successo o meno).

Nell'altro caso, bisogna considerare l'eventuale accadimento degli eventi rappresentati dai nodi genitore rispetto al nodo considerato, rendendo quindi necessario un ordinamento topologico dei nodi della rete. Successivamente viene nuovamente eseguito il confronto tra un numero random compreso tra 0 e 1 e la probabilità condizionata trovata del nodo in questione.

Il data set generato avrà n righe pari al numero di prove effettuate e come numero di colonne 8 pari al numero di nodi della rete.

4 Algoritmo K2

4.1 Assunzioni

L'algoritmo per poter essere applicato ha bisogno di alcune assunzioni:

- È necessario eseguire un ordinamento topologico dei nodi, per garantire la coerenza con le dipendenze condizionali della rete durante la generazione del data set.
- $\mathbf{X} = \{X_1, \dots, X_n\}$ costituisce l'insieme di tutte le variabili aleatorie che rappresentano gli eventi considerati nella rete Bayesiana con le relative probabilità, le quali assumono valori discreti.
- $\mathbf{D} = \{X_1 = x_1, \dots, X_n = x_n\}$ costituisce l'insieme del data set, ovvero l'insieme di tutte le prove i.i.d. effettuate nella verifica dell'avvenimento o meno di tutti gli eventi \mathbf{X} . L'insieme \mathbf{D} viene generato completo, cioè non deve avere data mancanti.

L'algoritmo K2 è un algoritmo score-based che effettua una ricerca greedy incrementale per l'apprendimento della struttura. Come input necessita di un data set ed un insieme di nodi ordinati. Inizialmente si parte con un grafo vuoto (tutti i nodi sono senza padre). Dopodiché dato l'ordinamento $\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$, l'insieme dei padri di ogni nodo x_i è determinato valutando se ogni variabile dell'insieme dei relativi predecessori, $\mathbf{pred}(\mathbf{X}_i) = \{\mathbf{X}_1, \dots, \mathbf{X}_{i-1}\}$ possa essere padre di X_i , ovvero quale $\mathbf{x}_j \in \mathbf{pred}(\mathbf{X}_i)$ massimizza la metrica K2 espressa dalla funzione di score espressa nel seguente modo:

$$f(i, \pi_i) = \sum_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \sum_{k=1}^{r_i} \alpha_{ijk}!$$

dove r_i è il numero di valori che può assumere X_i , q_i è il numero totale di possibili combinazioni dei nodi genitori di X_i . α_{ijk} è il numero di casi nel data set in cui $X_i = K$ quanto $Pa(X_i) = j$, infine N_{ij} è la frequenza con $Pa(X_i) = j$.

La fase di ricerca, per l'insieme dei padri di X_i termina dopo aver esaminato tutte le variabili in $\mathbf{pred}(\mathbf{X}_i)$ 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 dell'algoritmo utilizzato è il fatto di non poter "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.

5 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 (visità in profondità). 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.

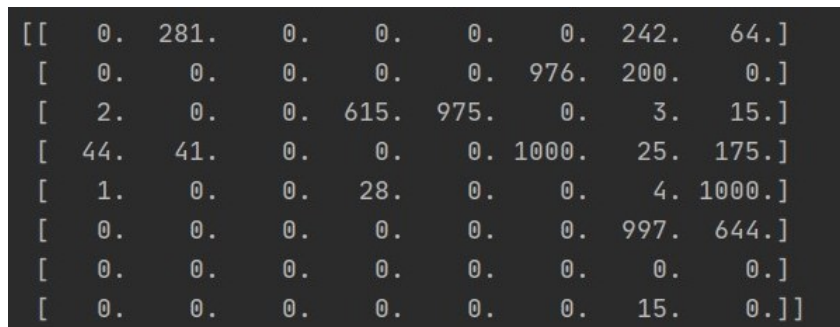
In ogni prova si genera un dataset (a partire dalla rete "ASIA") differente da 150 riga ciascuna e si applica l'algoritmo di apprendimento K2 su di esso, sommando i risultati via via ottenuti in una matrice, che ci rappresenterà con quale probabilità ogni arco può comparire nella rete.

6 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.  0.  0.]
 [ 0.  0.  0.  0.  0.  0. 15.  0.]]

```

Figura 3: Risultato ottenuto dai test.

7 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 e degli archi extra per via della natura greedy dell'algoritmo utilizzato, per poter avere un risultato migliore esistono delle euristiche migliori rispetto a quella usata nell'elaborato.