

Apprendimento di reti bayesiane da database di esempi

Enrico Papalini

papalini@biancaneve.ing.unifi.it

Michele Piccini

mpiccini@biancaneve.ing.unifi.it

corso di laurea in Ingegneria Informatica
Università di Firenze
a.a. 1996/97

Sommario

Le reti di Bayes sono modelli grafici di probabilità in cui i nodi rappresentano variabili aleatorie e gli archi le dipendenze causali fra variabili. Questa relazione descrive come apprendere la struttura grafica e la distribuzione di probabilità commesse alla rete a partire da un database di realizzazioni delle variabili. Viene presentata una tecnica basata su due passi: nel primo passo viene appresa la struttura selezionando il modello grafico che massimizza una appropriata funzione di costo, nel secondo vengono apprese le tabelle di probabilità associate a ciascun nodo tramite un approccio Bayesiano. Sono illustrati i fondamenti teorici della metodologia e i risultati ottenuti applicandola a database provenienti sia da processi simulati che reali.

1 Apprendimento bayesiano

Supponiamo di avere un insieme di variabili aleatorie $\mathbf{X} = \{X_1, \dots, X_n\}$ ed un insieme di loro realizzazioni $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, detto database di esempi. Supponiamo che sia ignoto il modello \mathbf{m} che descrive tale insieme di variabili e sia ignota la distribuzione di probabilità congiunta di questo insieme. Denotiamo questa distribuzione con $p(\mathbf{X} | \theta, \mathbf{m})$, dove θ rappresenta i parametri del modello \mathbf{m} . Il problema da risolvere è quello di stimare la distribuzione, noto il database D .

La distribuzione congiunta di \mathbf{X} può essere modellata con una rete bayesiana. Questa consiste in un grafo diretto aciclico S (detto *struttura*) 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. Inoltre una rete di Bayes contiene un insieme P di *distribuzioni locali di probabilità*, ciascuna associata ad una variabile aleatoria X_i e condizionata dalle variabili corrispondenti ai nodi sorgenti degli archi entranti nel nodo X_i .

La presenza di un arco dal nodo X_i al nodo X_j può essere interpretata come il fatto che X_i sia causa diretta di X_j . Possiamo pensare una rete di Bayes come un modello per la distribuzione delle variabili aleatorie in \mathbf{X} , chiamando \mathbf{m}_S il modello relativo ad una struttura S e definendo i parametri ad essa associati come $\theta = \{\theta_1, \dots, \theta_n\}$. Il simbolo θ_i rappresenta il vettore di parametri relativo alla distribuzione locale di probabilità riferita a X_i , indicata con $p(X_i | \mathbf{Pa}_i)$, ove \mathbf{Pa}_i è l'insieme dei nodi padri del nodo i -esimo.

La coppia (S, P) codifica in modo univoco $p(\mathbf{X})$, dato che la distribuzione di probabilità congiunta su \mathbf{X} è fattorizzabile come

$$p(\mathbf{X} | \theta, \mathbf{m}_S) = \prod_{i=1}^n p(X_i | \mathbf{Pa}_i, \theta_i, \mathbf{m}_S) \quad (1)$$

L'approccio bayesiano per risolvere il problema di stima proposto prevede di definire una variabile aleatoria discreta \mathbf{M} i cui stati sono i possibili \mathbf{m}_S , che codifica l'incertezza sulla struttura del modello tramite la distribuzione di probabilità a priori $p(\mathbf{M} = \mathbf{m}_S)$. Inoltre, per ogni modello \mathbf{m}_S viene definita una variabile aleatoria continua \mathbf{T} che codifica i possibili valori che i suoi parametri possono assumere, con un'incertezza a priori data dalla densità di probabilità $p(\mathbf{T} = \theta | \mathbf{m}_S)$.

Dato un database di esempi D , il teorema di Bayes permette di calcolare le distribuzioni a posteriori per le due variabili aleatorie, $p(\mathbf{m}_S | D)$ e $p(\theta | \mathbf{m}_S, D)$. Dopo aver scelto una distribuzione a priori per \mathbf{X} , condizionata al modello ed ai suoi parametri $p(\mathbf{x} | \theta, \mathbf{m}_S)$, la stima della distribuzione a posteriori si trova calcolando il valore atteso del prior rispetto $p(\mathbf{m}_S | D)$ e $p(\theta | D, \mathbf{m}_S)$:

$$p(\mathbf{x} | D) = \sum_{\mathbf{m}_S} p(\mathbf{m}_S | D) \int p(\mathbf{x} | \theta, \mathbf{m}_S) p(\theta | D, \mathbf{m}_S) d\theta \quad (2)$$

Purtroppo questo approccio bayesiano puro non può essere applicato nel caso dell'apprendimento delle reti bayesiane, dato che il numero dei possibili modelli rende il calcolo della sommatoria intrattabile. Si aggira il problema facendo l'ipotesi che la distribuzione $p(\mathbf{m}_S | D)$ sia localizzata attorno ad un particolare modello $\hat{\mathbf{m}}_S$. In questo caso, una volta selezionato $\hat{\mathbf{m}}_S$, la stima della distribuzione a posteriori di \mathbf{X} si riduce a:

$$p(\mathbf{x} | D, \hat{\mathbf{m}}_S) = \int p(\mathbf{x} | \theta, \hat{\mathbf{m}}_S) p(\theta | D, \hat{\mathbf{m}}_S) d\theta \quad (3)$$

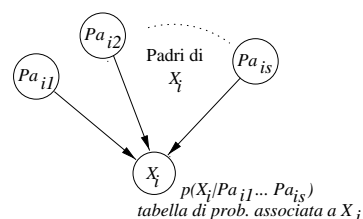


Figura 1: Schema della relazione fra un nodo ed i padri in una generica rete bayesiana.

Per selezionare $\hat{\mathbf{m}}_S$ si introduce una funzione che applicata ad un modello restituisca un “punteggio” che sia tanto più alto quanto il modello è più prossimo a $\hat{\mathbf{m}}_S$. È naturale utilizzare una funzione SCORE(\mathbf{m}_S) derivata da $p(\mathbf{m}_S | D)$, solitamente il suo logaritmo, che per il teorema di Bayes può essere messo in relazione con le distribuzioni a priori del modello e dei dati:

$$\begin{aligned} \text{SCORE}(\mathbf{m}_S) &= \log p(\mathbf{m}_S | D) \\ &= \log p(\mathbf{m}_S) + \log p(D | \mathbf{m}_S) - \log p(D) \\ &\simeq \log p(D | \mathbf{m}_S) \end{aligned} \quad (4)$$

L'approssimazione compiuta deriva dal fatto che $\log p(D)$ è una costante e che il prior sul modello $\log p(\mathbf{m}_S)$ può anch'esso essere supposto costante se si fa l'ipotesi che ogni modello sia equiprobabile (completa ignoranza a priori sul modello).

Per poter massimizzare SCORE(\mathbf{m}_S) debbono essere calcolate le distribuzioni $p(D | \mathbf{m}_S)$. Per poter stimare la distribuzione di \mathbf{X} è necessario il computo di $p(\mathbf{x} | \mathbf{D}, \mathbf{m}_S)$. Il loro calcolo può esser compiuto in forma chiusa applicando le seguenti ipotesi sulla rete bayesiana e sul database di esempi D .

- hp1.** Ogni variabile X_i è *discreta* (può assumere gli stati x_i^k , con $k = 1, \dots, r_i$) e la sua distribuzione locale di probabilità è una collezione di *distribuzioni multinomiali* $p(x_i^k | \mathbf{pa}_i^j, \theta_i, \mathbf{m}_S) = \theta_{ijk} > 0$, una per ogni stato \mathbf{pa}_i^j delle variabili padri ($j = 1, \dots, q_i$ con $q_i = \prod_{X_s \in \mathbf{Pa}_i} r_s$), tali che $\sum_{k=1}^{r_i} \theta_{ijk} = 1 \forall i, j$. Definiamo anche due vettori di parametri $\theta_{ij} = \{\theta_{ijk}\}_{k=1}^{r_i}$ e $\theta_i = \{\theta_{ij}\}_{j=1}^{q_i}$ per semplificare la notazione.
- hp2.** I parametri θ_{ij} sono *mutuamente indipendenti*. Ciò comporta, come illustrato in [1], che il problema diviene separabile nel senso espresso dall'equazione $p(\theta_S | \mathbf{m}_S) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\theta_{ij} | \mathbf{m}_S)$.
- hp3.** Ogni insieme di parametri θ_{ij} ha come distribuzione la coniugata della distribuzione della variabile X_i corrispondente. In questo caso è la *distribuzione di Dirchlet*, $p(\theta_{ij} | \mathbf{m}_S) \propto \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1}$ ove gli α_{ijk} sono degli iperparametri della distribuzione, tali che $\alpha_{ijk} > 0 \forall i, j, k$ e che $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.
- hp4.** D è *completo*, quindi non ci sono osservazioni mancanti.
- hp5.** Il campione D deve essere estratto da un fenomeno il cui modello è una struttura S di una rete di Bayes.

Sotto queste ipotesi in [2] sono riportati i seguenti risultati:

$$p(D | \mathbf{m}_S) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (5)$$

$$p(x_i^k | \mathbf{pa}_i^j, \mathbf{D}, \mathbf{m}_S) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \quad (6)$$

ove N_{ijk} è il numero delle volte che nel database D si ha $X_i = x_i^k$ e $\mathbf{Pa}_i = \mathbf{pa}_i^j$, mentre $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Dall'equazione 5 è immediato ricavare che SCORE(\mathbf{m}_S) può essere calcolato come

$$\begin{aligned} \text{SCORE}(\mathbf{m}_S) &= \sum_{i=1}^n \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \\ &+ \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned} \quad (7)$$

Per poter avere una formula computabile bisogna assegnare dei valori agli α_{ijk} . Questi iperparametri codificano la conoscenza a priori che l'utente ha sui parametri delle probabilità associate alla rete. Dato che abbiamo supposto una completa ignoranza, è logico porre $\alpha_{ijk} = \frac{\alpha}{q_i r_i}$ che deriva da $\alpha_i = \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{\alpha}{q_i r_i} = \alpha$, interpretabile come l'equiprobabilità di ogni istanza dello spazio delle probabilità congiunte su \mathbf{X}_i e \mathbf{Pa}_i . Resta così da assegnare un unico “iper”iperparametro α che in letteratura è chiamato *dimensione di un campione equivalente*.

Abbiamo tutti gli elementi per presentare uno schema della procedura di apprendimento di una rete bayesiana da un database D di esempi, che risolve il problema di stima proposto.

Procedura di stima della distribuzione di probabilità congiunta di \mathbf{X} .

- passo 1:** trovare, tramite un opportuno algoritmo di ricerca, la struttura $\hat{\mathbf{m}}_S$ che massimizza SCORE(\mathbf{m}_S), dopo aver assegnato un opportuno α .
- passo 2:** calcolare la distribuzione a posteriori $p(\mathbf{X} | \mathbf{Pa}_i, D, \hat{\mathbf{m}}_S)$ per ciascuno stato di ogni variabile X_i , in modo da ottenere le stime per i valori contenuti nelle tabelle di probabilità associate a ciascun nodo.

Per avere uno schema di apprendimento completo, manca solo la definizione di una metodologia di ricerca nello spazio delle strutture delle reti bayesiane associate all'insieme di variabili aleatorie \mathbf{X} . Seguendo i risultati esposti in [3] è stata scelta una procedura di ricerca “hill-climbing” per cercare di massimizzare la funzione SCORE(\mathbf{m}_S).

Scelta una struttura S è possibile valutare il guadagno di SCORE che si ha per ogni possibile variazione elementare degli archi, in modo da mantenere l'aciclicità del grafo. Queste variazioni sono l'aggiunta di un arco fra due nodi mutuamente indipendenti, la cancellazione di un arco fra due nodi dipendenti, il cambiamento di verso di un arco fra due nodi.

Sfruttando il fatto che la funzione di costo descritta dall'equazione 7 può essere scomposta nella somma di n addendi, ciascuno associato ad un nodo X_i ed ai suoi padri \mathbf{Pa}_i , la variazione di un solo arco della struttura S influirà al più su due addendi, relativi ai nodi sorgente e pozzo dell'arco variato. In particolare ciò accade soltanto se un arco della struttura viene invertito. Negli altri casi è sufficiente calcolare la variazione dell'addendo relativo al nodo pozzo del nuovo arco.

Dopo aver calcolato tutte le variazioni elementari possibili si effettua, se esiste, quella che porterebbe un guadagno positivo maggiore. Il nuovo SCORE viene aggiornato e si reitera il procedimento. La ricerca termina nel caso in cui nessuna modifica faccia aumentare lo SCORE oppure se viene raggiunto il limite massimo del numero di iterazioni possibili. Lo schema della procedura di ricerca appena esposta si trova in figura 2.

Questo tipo di approccio necessita di un grafo di partenza. Candidati per questo possono essere il grafo privo di archi, che codifica la completa ignoranza sulle relazioni che intercorrono fra le variabili, un grafo aciclico costruito inserendo archi in modo casuale oppure una rete che rappresenti la conoscenza a priori posseduta sul dominio del problema.

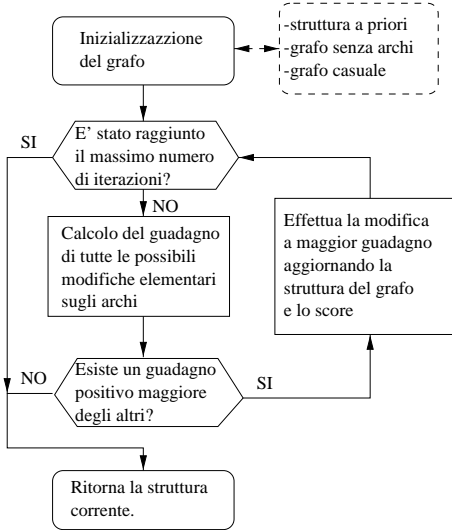


Figura 2: Schema della ricerca Hill-Climbing in uno spazio di grafi associati a reti bayesiane.

2 Risultati da reti simulate

Quanto esposto nella precedente sezione è stato implementato utilizzando il linguaggio C++ in ambiente Linux. Una prima analisi è stata compiuta su reti bayesiane simulate. Viene di seguito illustrata la procedura seguita per l'apprendimento di queste reti.

Procedura di apprendimento di una rete simulata.

- passo 1:** Generazione di una rete random, detta *gold network*, avente il numero dei nodi e il massimo numero di stati per nodo prefissati e con una specifica probabilità per l'inserimento degli archi fra i nodi.
- passo 2:** Campionamento della *gold network* in modo da ottenere un database di esempi D con un numero di campioni prestabilito.
- passo 3:** Perturbazione della *gold network* per ottenere una *start network* da cui cominciare la ricerca. Il metodo di perturbazione utilizza una certa probabilità per modificare gli archi esistenti ed un'altra per aggiungere nuovi archi, con il vincolo di mantenere il grafo aciclico.
- passo 4:** Apprendimento strutturale, a partire da D e dalla *start network*, utilizzando l'algoritmo di ricerca esposto nella sezione precedente.
- passo 5:** Apprendimento delle tabelle di probabilità associate a ciascun nodo tramite una procedura che implementa l'equazione 6.

Per completare l'analisi è stato necessario studiare un modo per misurare la "bontà" sia della struttura selezionata che delle tabelle apprese, rispetto a quelle della *gold network* di partenza. Per fare ciò ci siamo basati sui suggerimenti riportati in [3], in cui vengono utilizzati due tipi di misure. La prima è la *differenza strutturale* fra due reti, che rappresenta il grado con cui la struttura appresa ha catturato le relazioni causali fra variabili. Definita la differenza simmetrica δ_i fra i padri di X_i in due differenti reti P e Q come

$$\delta_i = |(\mathbf{Pa}_i^Q \cup \mathbf{Pa}_i^P) \setminus (\mathbf{Pa}_i^Q \cap \mathbf{Pa}_i^P)| \quad (8)$$

la differenza strutturale δ si calcola sommando tutte le δ_i , $\delta = \sum_{i=1}^n \delta_i$. Si osservi come la differenza strutturale sia la

misura del numero degli archi in cui le reti P e Q differiscono, contando due volte gli archi che sono stati invertiti nel passaggio da P a Q .

L'altra misura è la *cross-entropia* fra due reti, che denota quanto bene la rete bayesiana appresa predirà il prossimo campione del database D . Dette $p(\mathbf{X} | \mathbf{m}_P)$ e $p(\mathbf{X} | \mathbf{m}_Q)$ le distribuzioni congiunte di probabilità codificate dalle reti P e Q , la cross-entropia $H(P, Q)$ è data da

$$H(P, Q) = \sum_{\mathbf{X}} p(\mathbf{X} | \mathbf{m}_P) \log \frac{p(\mathbf{X} | \mathbf{m}_P)}{p(\mathbf{X} | \mathbf{m}_Q)} \quad (9)$$

Un primo esempio di risultato ottenuto si può osservare in figura 3, dove non sono state riportate le tabelle di probabilità associate ai nodi, in quanto troppo voluminose. Ad esempio, la tabella associata al nodo tre ha dimensioni 5×240 in quanto X_3 può assumere valori in 5 stati differenti ed i nodi padri \mathbf{Pa}_3 hanno in totale 240 stati. Si osservi come la rete appresa differisca dalla *gold network* di soli tre archi, tutti e tre cancellati rispetto all'originale. Questi archi fanno passare il numero degli stati dei padri di X_3 da 240 a 6, semplificando notevolmente la tabella delle probabilità condizionate associata. Una spiegazione di questo comportamento è data dal fatto che gli algoritmi proposti tendono a selezionare reti semplici rispetto ad un determinato database D . Se vi aggiungiamo il fatto che in questo caso D non ha una dimensione elevata (soltanto 4000 campioni), appare evidente come l'algoritmo abbia semplificato la rete per l'assenza di sufficienti informazioni per cogliere la complessità del terzo nodo. Si osservi come il resto della rete sia stato correttamente riconosciuto.

Un altro esempio, può essere osservato nell'output originale, allegato alla relazione e modificato per mettere in risalto gli archi aggiunti (A), invertiti (R) o cancellati (D). La *gold network* ha 20 nodi e variabili binarie: è stata campionata ottenendo un database D di 2000 campioni. Perturbata con una probabilità di $\frac{1}{5}$, ha generato la *start network*, annotata nell'output per mettere in luce le differenze rispetto alla rete originale. Il listato riporta la descrizione dei 39 passi dell'algoritmo di ricerca ($\alpha = 64$), che mostrano il valore attuale di $\text{SCORE}(\mathbf{m}_S)$ ed il guadagno ottenuto compiendo la variazione indicata fra parentesi. Di fianco è stata annotata la bontà della mossa (errata E, correzione di errore precedente C) e quale sarebbe stata la mossa esatta. L'output si conclude con la lista degli archi della rete appresa. Si osservi come anche in questo caso la differenza strutturale scenda da 35 a 9 con l'apprendimento e la bassa cross-entropia fra la *gold network* e la *learned network*. Questo dato fa sospettare un possibile "overfitting" dei dati da parte dell'algoritmo di apprendimento delle tabelle di probabilità. Interessante anche notare come l'algoritmo di apprendimento della struttura della rete rimedi solo una volta ad una scelta sbagliata compiuta in precedenza. A causa del suo carattere di ricerca locale, le scelte errate fatte ai primi passi fanno quasi certamente cadere in un massimo locale differente da quello globale, con poche probabilità che gli errori possano essere corretti. Comunque, anche in questo esempio la rete appresa è molto più vicina alla rete originale rispetto alla *start network*, dimostrando come la tecnica implementata affini realmente la conoscenza iniziale grazie al contributo delle osservazioni contenute nel database D .

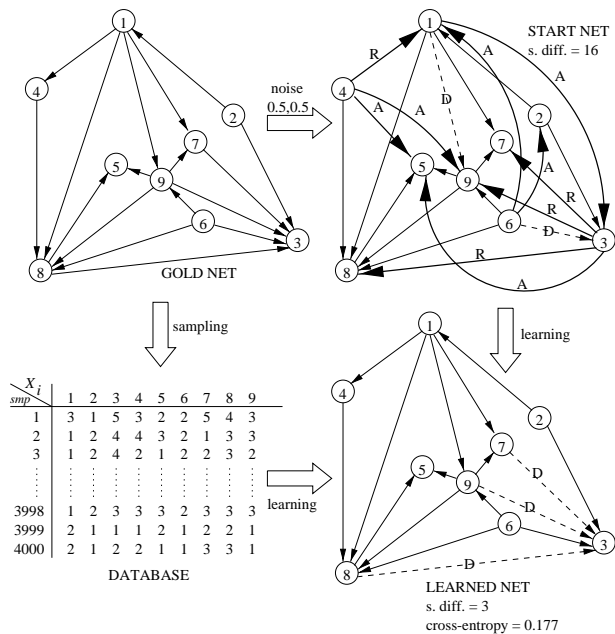


Figura 3: Risultati di un test su una rete “gold” generata a caso con 9 variabili e al più cinque stati per variabile. In un primo momento sono stati estratti 4000 campioni da questa rete, che poi è stata perturbata con probabilità $\frac{1}{2}$ ottenendo una rete “start”. Le lettere associate agli archi indicano se un arco è stato aggiunto(A), rimosso(D) o invertito(R). A partire dalla “start” utilizzando i dati contenuti nel database è stata eseguita la procedura di apprendimento a due passi (struttura + tabelle), con $\alpha = 100$, che ha dato come risultato la rete “learned”. Si osservi come la differenza strutturale dalla “gold” sia scesa da 16 a 3 con l’apprendimento, mentre le tabelle di probabilità della “learned” hanno una bassa cross-entropia, 0.176679, con quelle della “gold”.

Per concludere l’analisi dell’algoritmo su reti simulate presentiamo l’apprendimento di una rete più semplice delle precedenti, illustrata in figura 4. In questo caso l’algoritmo ha appreso la rete senza commettere errori: il motivo è il corretto dimensionamento del numero di esempi rispetto alle dimensioni del modello e la corretta scelta del parametro α .

3 Apprendimento da dati reali

Per applicare le tecniche espote ad un processo proveniente dal mondo reale, abbiamo utilizzato il database riportato a pag.44 in [4]. Si tratta di uno studio svolto dai professori Sewell e Shah riguardo le intenzioni di una popolazione scolastica di 10416 studenti dell’ultimo anno di superiori di proseguire gli studi, frequentando l’università. I dati sono stati raccolti alla *Wisconsin High School* nel 1968 e riguardano il sesso dello studente, il suo stato socioeconomico, il suo quoziente di intelligenza, l’incoraggiamento ricevuto dai genitori a proseguire gli studi e l’effettiva decisione di farlo o meno. La seguente tabella illustra in dettaglio le variabili ed i possibili stati che possono assumere, nonché i codici numerici che il prototipo assegna loro, utili per comprensione dei risultati ottenuti allegati in formato elettronico.

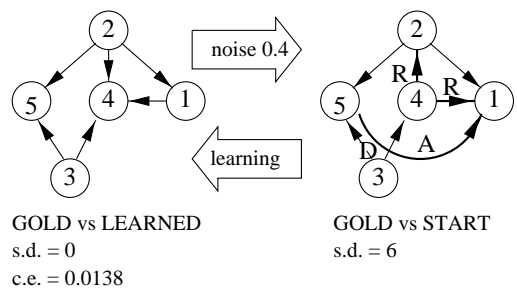


Figura 4: Risultati di un test su una rete “gold” generata a caso con 5 variabili e al più quattro stati per variabile. Sono stati estratti 5000 campioni da questa rete, poi è stata perturbata con probabilità $\frac{2}{5}$ ottenendo una rete “start”. La procedura di apprendimento a due passi, con $\alpha = 8$, ha prodotto la rete “learned” che è identica nella struttura alla “gold” e anche nelle tabelle di probabilità (cross-entropia pari a 0.0138233). I passi compiuti dall’algoritmo di ricerca strutturale sono stati, partendo da $SCORE(\mathbf{m}_{start}) = -26777.3$, add: $3 \rightarrow 5 (+310.8)$, rev: $2 \rightarrow 4 (+142.8)$, rev: $1 \rightarrow 4 (+179.5)$ e del: $5 \rightarrow 1 (+24.0)$, ottenendo uno $SCORE(\mathbf{m}_{learned}) = -26120.2$.

X_i sigla	X_1 SEX	X_2 SES	X_3 IQ	X_4 PE	X_5 CP
signif.	Sex	SocioEcon. Status	Intell. Quotient	Parental Encourag.	College Plans
1	male	low	low	low	yes
2	female	lower mid.	lower mid.	high	no
3		upper mid.	upper mid.		
4		high	high		

Abbiamo tentato un primo apprendimento partendo da una rete senza archi fra i nodi, e tenendo il parametro α ad un basso valore, dato che la *start network* fornita era del tutto non informativa. I risultati ottenuti sono riportati in forma grafica nella figura 5A. Come si può osservare, gli archi appresi codificano molte relazioni causali che ci saremmo potuti aspettare, come ad esempio il fatto che il fattore socioeconomico influenzi i genitori a dare o no ai figli stimoli per andare all’università. A nostro avviso, il solo arco che connette CP a IQ sembra illogico, in quanto supporrebbe il fatto che la scelta di andare all’università influisca sul quoziente di intelligenza dello studente.

Per tentare di evitare questo inconveniente abbiamo ripetuto l’apprendimento partendo da una rete in cui ogni nodo è collegato a CP con un arco. Questo equivale a supporre una conoscenza a priori sul fatto che tutti gli attributi possono influenzare in qualche modo la scelta universitaria. Avendo introdotto una *start network* che ritenevamo molto importante, l’apprendimento è stato eseguito con un parametro α elevato. I risultati ottenuti sono osservabili nella figura 5B. Il grafo appreso mostra che il problema riscontrato è stato eliminato (adesso c’è un arco da IQ a CP, che codifica il fatto che il quoziente di intelligenza può influenzare la scelta di proseguire gli studi). Però, la rete è logicamente debole rispetto ad un altro arco, quello che collega SES a IQ. In questo caso infatti sembrerebbe il quoziente di intelligenza poter essere influenzato dalla condizione socioeconomico dell’individuo.

Abbiamo tentato di percorrere una terza strada. Siamo ritornati ad una *start network* priva di archi, ma abbiamo imposto dei vincoli sulla struttura. In questo caso si è supposto che il nodo CP non potesse essere padre di altri nodi. I risultati ottenuti dall’apprendimento in queste condizioni sono riportati in figura 5C, partendo da un parametro α basso. La rete appresa è la stessa del caso precedente.

Da questi risultati si può trarre la conclusione che l’algoritmo proposto sia un buon approccio per trattare dati in una

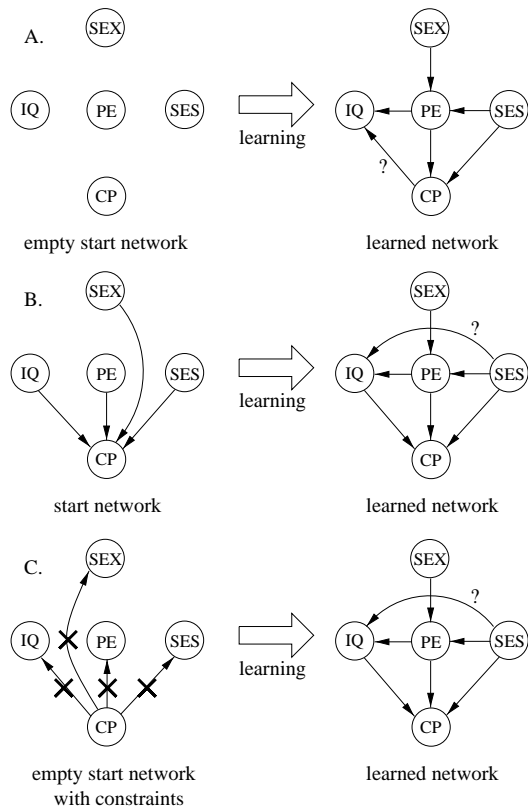


Figura 5: Risultati dell'apprendimento dal database di Sewell & Shah. **A.** $\alpha = 8$ e *start network* vuota. **B.** $\alpha = 256$ e *start network* con un arco entrante in CP da ogni nodo. **C.** $\alpha = 8$ e *start network* vuota, ma con il vincolo che CP non può essere padre di alcun nodo. Gli archi con un “?” sono, a nostro avviso, illogici.

prima fase, in modo da ottenere una rete di Bayes che illustri buona parte delle relazioni fra le variabili, commettendo sì qualche errore, ma fornendo già una buona visione di molte relazioni che intercorrono fra le variabili. In una seconda fase, partendo dai risultati ottenuti, è possibile utilizzare qualche metodo più sofisticato per tentare di raffinare la “bontà” causale della rete. Un esempio di questa tecnica può essere trovato in [4], in cui l'impiego di una metodologia che utilizza variabili nascoste permette di introdurre una nuova variabile che sembra rendere conto della “qualità” dei genitori, separando il quoziente di intelligenza dallo status economico.

4 Osservazioni conclusive

Abbiamo presentato una tecnica per apprendere reti bayesiane su un insieme di variabili aleatorie a partire da un database di loro realizzazioni. Il metodo esposto può essere utilizzato per comprendere le relazioni di causalità che intercorrono fra le variabili e per attuare una prima analisi dei dati ad esse relativi.

Sono stati messi in luce alcuni difetti, che abbiamo imputato sia alla non applicabilità delle ipotesi a molti casi di interesse reale che alla difficoltà di una corretta assegnazione dei parametri dell'algoritmo, primo fra tutti il coefficiente α . Inoltre è stato mostrato che nel caso in cui le ipotesi siano soddisfatte, i parametri correttamente dimensionati e si stia

utilizzando un numero sufficiente di campioni, l'algoritmo apprende senza errori la rete.

In ultima analisi, l'apprendimento basato su selezione del modello e stima bayesiana dei parametri è un buon metodo per affrontare il problema di trovare relazioni fra dati. I risultati ottenuti possono essere raffinati impiegando tecniche di apprendimento più potenti, quali sono quelle che suppongono la presenza di variabili nascoste.

Questo lavoro è stato realizzato per sostenere l'esame di “intelligenza artificiale” tenuto dal prof. G.Soda con la collaborazione del dott. P.Frasconi. Sono state utilizzate le strutture didattiche del laboratorio dell'informazione “ex-forno”, all'interno della facoltà di ingegneria dell'università di Firenze.

Riferimenti bibliografici

- [1] Spiegelhalter & Lauritzen, *Sequential updating of conditional probabilities on directed graphical structures*, Networks, 20:579-605, 1990
- [2] Cooper & Herskovits, *A Bayesian method for the induction of probabilistic networks from data*, Machine Learning, 9:309-347, 1992
- [3] Heckerman, Geiger & Chickering, *Learning Bayesian networks: The combination of knowledge and statistical data*, Machine Learning, 20:197-243, 1995
- [4] Heckerman, *A tutorial on learning with Bayesian Networks*, Microsoft Technical Report, TR-95-06, revised Nov. 1996