Università degli Studi di Salerno

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Elettronica

Tesi di Laurea

Reti Bayesiane: un approccio multi – esperto allo Structural Learning

Relatori
Ch.mo Prof. Mario Vento
Ch.mo Prof. Massimo De Santo

Co-relatore
Ing. Francesco Colace

Candidato
Sabatino Liguori
Matr. 66/01766

ANNO ACCADEMICO 2001-2002

INTRODUZIONE……..*4*

1 LE RETI BAYESIANE *7*

1.1 Introduzione **7**

1.2 Perché le reti Bayesiane? **8**

1.3 Bayesian Networks **10**
1.3.1 L'approccio Bayesiano alla probabilità 10
1.3.2 I modelli grafici e cenni alla teoria dei grafi 12
1.3.3 Definizione di Rete Bayesiana 14
1.3.4 Un esempio 15

1.4 L'Inferenza nelle reti Bayesiane **17**
1.4.1 L'Inferenza statistica 17
1.4.2 I diversi approcci all'inferenza statistica 18
1.4.2.1 Classico 18
1.4.2.2 Bayesiano 18
1.4.2.3 Teoria delle Decisioni 18
1.4.3 Il processo di inferenza e le reti Bayesiane 19
1.4.3.1 Un esempio di inferenza 22
1.4.4 Algoritmi per l'Inferenza nelle BN 26

1.5 Learning Bayesian Networks **28**

1.6 Stato dell'arte **30**
1.6.1 Causal network (CN) 30
1.6.2 Dynamic Bayesian Networks 32
1.6.3 Introduzione ai Diagrammi di influenza 33
1.6.4 Object-Oriented Networks 34
1.6.5 Function Bayesian Networks 35
1.6.6 Causal discovery 35
1.6.7 Models of Cognition 36
1.6.7.1 I sistemi esperti 36
1.6.7.1.1 I sistemi rule-based 38
1.6.7.2 L'incertezza nei sistemi esperti 39
1.6.7.3 I sistemi normative expert 40

1.7 Applicazioni delle reti Bayesiane **42**
1.7.1 Il processo di retrieval 43
1.7.1.1 Information Retrieval e BN 44
1.7.2 Data Mining 46
1.7.2.1 Data Mining e Reti di Bayes 46

2 L'APPRENDIMENTO DI RETI BAYESIANE *48*

2.1 L'Approccio Bayesiano **52**
2.1.1 Known Structure 52
2.1.2 Unknown Structure 59

2.2 Le Assunzioni **65**
2.2.1 La distribuzione a priori 71
2.2.1.1 Priori per i parametri 72
2.2.1.2 Priori per la struttura 73
2.2.2 Missing Values e Hidden Variables (Cenni) 74
2.2.2.1 L'Approssimazione Gaussiana 76

2.3 Parameter Learning **78**
2.3.1 L'Algoritmo EM 79

2.4 Structural Learning **82**

3 STRUCTURAL LEARNING 83

3.1 Introduzione 83

3.2 Lo Structural Learning 84
3.2.1 L' algoritmo di Structural Learning 87
3.2.1.1 I metodi di ricerca 89
3.2.1.1.1 Local Search 92
3.2.1.1.2 Hill climbing 92
3.2.1.1.3 Simulated annealing 93
3.2.1.2 Scoring Function 93
3.2.1.3 I test di indipendenza condizionata 94
3.2.1.3.1 Il test χ^2 come test di indipendenza 95
3.2.1.3.2 Mutua Informazione 97

3.3 Metodologie di valutazione 99

3.4 Gli algoritmi di Structural Learning 100
3.4.1 Gli algoritmi bayesiani 100
3.4.1.1 L'Algoritmo Bayesiano 100
3.4.1.2 L'Algoritmo K2 104
3.4.1.2.1 Modifiche al K2: l'algoritmo CB 109
3.4.1.3 MDL - based algorithm: l'algoritmo K3 110
3.4.2 I metodi Constraint Based 115
3.4.2.1 Premessa: L'indipendenza nei grafi 115
3.4.2.2 L'Algoritmo PC 125
3.4.2.3 L'Algoritmo TPDA 129
3.4.2.3.1 Learning Bayesian Network e Information Theory 130
3.4.2.3.2 Il Three-Phase Dependence Analysis Algorithm 131
3.4.2.3.3 Le tre fasi dell'algoritmo 132
3.4.2.3.3.1 Subroutine EdgeNeeded* 137
3.4.2.3.3.2 Subroutine EdgeNeeded_H (Heuristic) 138
3.4.2.3.3.3 Subroutine EdgeNeeded (Guaranteed) 141
3.4.2.3.3.4 Orienting Edges 142

3.5 Cenni ai metodi che gestiscono missing values 143

4 LA NOSTRA PROPOSTA:L'APPROCCIO MULTI-ESPERTO 146

4.1 Aspetti teorici del Multiple Experts (ME) 148
Regole di combinazione 149
4.1.1.1 Strategie basate sul voto 150
4.1.1.2 Strategie basate sul punteggio 150
4.1.1.3 L'approccio Multi-Esperto applicato alle Bayesian Network 151

4.2 Il tool realizzato: *BayExpert* 152
4.2.1 BayExpert: Acquisizione e memorizzazione di una Bayesian network 153
4.2.1.1 Il formato BIF 154
4.2.2 BayExpert: Structural Learning 155
4.2.3 BayExpert: Inferenza 158
4.2.4 BayExpert: Visualizzazione di una BN 158

4.3 La sperimentazione 161
4.3.1 Le modalità della sperimentazione 161
4.3.2 La descrizione dei database 163
4.3.2.1 La rete ALARM 163
4.3.2.2 La rete ASIA 164
4.3.2.3 La rete ANGINA 165
4.3.2.4 La rete PREGNANCY 166
4.3.2.5 La rete LED 167
4.3.2.6 La rete SPRINKLER 168
4.3.2.7 La rete COLLEGE 169

4.3.2.8 L'ontologia del corso Fondamenti di Informatica (CFI)..... 170

4.4 I risultati 174

CONCLUSIONI 203

APPENDICE 207

BIBLIOGRAFIA 233

INTRODUZIONE

"E' intelligenza artificiale quel settore dell'informatica che cerca di riprodurre nei computer quel tipo di comportamenti che, quando sono assunti dagli esseri umani, vengono generalmente considerati frutto della loro intelligenza"
Marvin Minsky

«Nam et ipsa scientia potestas est»
(*Il sapere è potere*)
Francis Bacon

Le due citazioni con cui si apre questo lavoro riassumono gli aspetti più importanti esaminati nella tesi. Marvin Masky, il cui asserto risale alla tesi di dottorato discussa nel '56, è considerato uno dei fondatori del campo disciplinare dell'Artificial Intelligence (A.I.) che è una delle aree più affascinanti dell'informatica sia per i risultati già raggiunti ma soprattutto per le invitanti prospettive future. Numerosi sono i campi di applicazione: dalla teoria dei giochi, alla percezione visiva, alla vita artificiale (ovvero sistemi che tendano a riprodurre situazioni reali). Sintetizzando, l'Intelligenza Artificiale ha sviluppato al proprio interno diversi filoni (paradigmi) che evidenziano l'esistenza di punti di vista differenti sulle possibilità di attuare il suo ambizioso programma:

- *approccio classico cognitivista*: rappresenta l'approccio top/down (dall'alto verso il basso) e presuppone che la metodologia base dell'A.I. consista nello studiare i processi mentali umani, formalizzarli e poi riprodurli nel computer con opportuni linguaggi (sistemi esperti).
- *approccio "emergente"*: l'idea principale che influenza questo approccio, detto anche bottom-up (dal basso verso l'alto), presuppone che il modo più efficace per pervenire all'A.I. sia quello di simulare sistemi in cui l'intelligenza possa emergere spontaneamente in seguito all'interazione del sistema riprodotto artificialmente con un ambiente naturale o simulato (ad esempio reti neurali, reti Bayesiane, algoritmi genetici).

L'assunto del Bacon esprime pienamente l'importanza della "conoscenza": ai nostri giorni tutto ciò che contribuisce ad arricchire il proprio sapere acquista un valore rilevante. Soprattutto in ambito economico, un know-how, al passo coi tempi ed in continua evoluzione, è un valore aggiunto per un'azienda, permettendole di essere concorrenziale sul mercato.

Con il progredire della tecnologia cambia anche il modo di apprendere: il mondo del Web è un esempio di come oggi sia più facile semplice reperire delle informazioni, ma la vera frontiera è il Knowledge Discovery, ovvero l'insieme di strategie che consentono di estrapolare informazioni utili da osservazioni (campioni) su un ambiente o un caso reale. L'Intelligenza artificiale è allo studio di metodi per automatizzare i processi di apprendimento (la percezione visiva, ad esempio). Un approccio recente al Knowledge Discovery è il Data Mining, che U.Fayyad, G.Piatetsky-Shapiro, P.Smyth, R.Uthurusamy (in "Advances in knowledge discovery and data mining") definiscono come "*non trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"*. Questa definizione evidenzia l'aspetto inferenziale del processo e la capacità di far emergere la "conoscenza" dai dati *senza richiedere la formulazione di specifiche ipotesi a priori*, come accade, invece, per l'Information Retrieval. Il data mining offre di un approccio esplorativo e non, come nel data retrieval, verificativo, evidenziando relazioni che non solo erano nascoste e sconosciute, ma che spesso non si era nemmeno mai ipotizzato potessero esistere.

Sia il data mining che l'AI sono campi di ricerca presso l'Università degli Studi di Salerno, con particolare riguardo all'elaborazione delle immagini e a sistemi di tutoring intelligenti per gli ambienti di e-learning. Infatti ognuno dei paradigmi dell'AI, classico o emergente, influenza in misura diversa i modi con cui rappresentare i processi di apprendimento, le teorie conseguenti e in alcuni casi alcuni aspetti delle pratiche didattiche, generando una vera e propria *Tecnologia Educativa*. La tecnologia dell'e-learning, in espansione negli ultimi anni, attrae sia aziende che università in quanto con il web e le nuove tecnologie (trasmissione satellitare, per citarne una) si dispone di strumenti che consentono una formazione flessibile, rapida e più idonea alle esigenze di un utente (un indubbio vantaggio è la possibilità di fruire dei contenuti formativi a proprio piacimento, dove e quando si vuole).

Il presente lavoro è concentrato sulle Bayesian Network (reti Bayesiane), una delle tecnologie emergenti di Intelligenza Artificiale. Con questi strutture, sinergia di una rappresentazione grafica e probabilistica, è possibile modellare ed inferire su problemi secondo un'analisi bottom-up, top-down ma anche seguendo schemi direzionali diversi. In particolare, sono allo studio algoritmi di data mining per le BN, in grado di ricostruire una rete Bayesiana da osservazioni campionarie su una certa realtà: tale compito non è semplice ma i risultati ottenuti sono soddisfacenti.

Gli obiettivi della tesi sono:

- caratterizzare le reti Bayesiane (**Capitolo 1**);
- analizzare l'apprendimento automatizzato di reti Bayesiane - learning from data - secondo l'ottica Bayesiana (**Capitolo 2**), in particolare dello structural learning (**Capitolo 3**);
- visto che in letteratura non è stato proposto nessun algoritmo che ricostruisca perfettamente una Bayesian Network si propone un approccio diverso, utilizzato negli ultimi tempi in diverse aree: il multi - esperto (ME) (**Capitolo 4**);
- realizzare un tool in Java per sperimentare l'approccio (ME) (**Capitolo 4**).

Nell'appendice, infine, sono presenti alcuni richiami sulla teoria delle probabilità, cenni sul test del chi quadro, ed approfondimenti sull'apprendimento delle reti Bayesiane quali un elenco dettagliato dei software disponibili per la tecnologia Bayesiana.

1 LE RETI BAYESIANE

1.1 INTRODUZIONE

L'esperto di genetica Sewall Wright, nel 1921, fu il primo a concepire una rappresentazione grafica di modelli probabilistici per l'esemplificazione di un problema.

Alla fine degli anni '70, l'interesse per la Scienza della Cognizione e l'Intelligenza Artificiale ha favorito lo sviluppo di modelli per rappresentare relazioni di tipo probabilistico fra un insieme di variabili in condizioni di incertezza: le Reti Bayesiane (BN - Bayesian Networks o Belief Networks, Belief Bayesian Networks BBN).

La disponibilità di una rigorosa base probabilistica e della teoria dei grafi, l'immediatezza dell'espressione grafica e la possibilità di realizzare un processo di inferenza di tipo direzionale[1] ha condotto, infatti, ad una rapida affermazione delle reti Bayesiane per la codifica della conoscenza, in condizioni di incertezza, nei sistemi esperti, integrando o sostituendo i sistemi rule - based (così definiti perché costruiti su un insieme di regole).

Le Belief Network sono *una diretta rappresentazione di un dominio*[2] *e non del processo di ragionamento*: i legami rappresentati nel modello grafico esprimono le reali connessioni fra le variabili e non il flusso del processo di ragionamento (come avviene invece nei sistemi rule-based).

In ogni caso le BN consentono la comprensione di un problema complesso, anche a persone non esperte, proprio grazie all'esplicazione dei legami fra le variabili. [BUN96] [JEN96] [PEA200]

[1] L'evidenza, in un processo di inferenza, è definita di tipo top – down (semantica, ovvero la causa A implica l'effetto B) e bottom – up (di percezione, ovvero quale è la causa che ha provocato l'effetto B?).

[2] Un dominio indica un insieme di variabili (aleatorie) con cui modellare un problema in condizioni di incertezza.

1.2 PERCHÉ LE RETI BAYESIANE?

Il crescente interesse verso queste strutture ha coinvolto diverse aree di ricerca: genetica, scienze sociali, statistica (in particolare i problemi con molte variabili), teoria della decisione, intelligenza artificiale (per modellare sistemi intelligenti di tipo probabilistico).

L'attenzione verso le Bayesian network è comprensibile perché:

- Le reti Bayesiane possono gestire velocemente insiemi *incompleti* di dati. Per esempio, consideriamo un problema di regressione dove due delle variabili di ingresso sono fortemente correlate. Se tutti gli ingressi fossero osservati, questa correlazione non rappresenterebbe un problema per le tecniche di apprendimento standard. Quando uno degli ingressi non è osservato, la maggior parte dei modelli, invece, produce una predizione non accurata mentre le reti Bayesiane offrono un modo naturale per l'inferenza anche in assenza di dati completi[3].
- Le reti Bayesiane permettono di apprendere le relazioni causali. L'apprendimento delle relazioni causali è importante perché aumenta il grado di comprensione del dominio di un problema e permette di fare predizioni in merito a interventi futuri. Ad esempio, un analista di mercato deve decidere se una campagna pubblicitaria influenza l'incremento della vendita di un prodotto, ovvero bisogna determinare se la pubblicità rappresenta una causa dell'incremento delle vendite. *L'uso di reti Bayesiane permette di risolvere problemi simili anche quando non è disponibile nessun esperimento a riguardo*.
- Le reti Bayesiane, in congiunzione con le tecniche statistiche di tipo Bayesiano, facilitano l'associazione fra la rappresentazione del dominio, la conoscenza a priori ed i dati. La conoscenza a priori del dominio è importante, specialmente quando i dati sono scarsi o costosi. Nelle reti Bayesiane, la semantica causale (rappresentazione grafica delle relazioni)

[3] Ad esempio, la mancanza dei dati impedisce il processo di inferenza in un modello di regressione lineare mentre una rete Bayesiana potrebbe essere fornita da un esperto. Diverso è il discorso dell'apprendimento, learning, che verrà illustrato in seguito, in cui i missing values complicano l'analisi delle BN.

e la possibilità di apprendere le probabilità condizionate rendono la codifica della *priori knowledge* particolarmente chiara.

- I metodi Bayesiani, in congiunzione con le reti Bayesiane, offrono un approccio efficiente per evitare l'overfitting[4] dei dati.

La ricerca, negli ultimi anni, è concentrata sullo sviluppo di metodi per apprendere - *learning* - reti Bayesiane dai dati. Le tecniche sviluppate sono nuove ed ancora in fase di studio ma i notevoli risultati ottenuti, in particolare per applicazioni legate all'analisi dei dati, costituiscono un forte stimolo per i ricercatori. [HEC94]

[4]Il termine overfitting indica, in tale ambito, un'eccessiva dipendenza del modello probabilistico, o di inferenza, dai dati in base ai quali è stato costruito.

1.3 BAYESIAN NETWORKS

Le reti Bayesiane *(Bayesian Network*, B*elief Networks* o *Bayesian Belief Networks – BN o BBN*) sono un valido strumento per rappresentare un dominio in condizioni di incertezza[5]. In particolare, si parla di *Causal Probabilistic Networks – CPN*, quando le reti Bayesiane sono impiegate per evidenziare relazioni di tipo causale all'interno del dominio.

Le Belief Network necessariamente non implicano un impiego dei metodi Bayesiani; piuttosto l'aggettivo "Bayesiano" è legato all'utilizzo della regola di Bayes per l'inferenza probabilistica[6]. Le probabilità codificate dalla rete Bayesiana possono infatti essere "Bayesiane" se derivate dalla conoscenza a priori (fornite da un esperto, ad esempio), "fisiche" quando l'apprendimento delle probabilità avviene esclusivamente da un database di esempi (interpretazione frequentista).

1.3.1 L'APPROCCIO BAYESIANO ALLA PROBABILITÀ

Per comprendere le reti Bayesiane e le tecniche di apprendimento ad esse associate, è importante delineare l'approccio Bayesiano alla probabilità e alla statistica.

La probabilità Bayesiana[7] di un evento[8] x è espressa dal livello di fiducia che una persona associa all'evento; quindi mentre la probabilità classica è una proprietà fisica del mondo, basata sull'interpretazione frequentista, quella Bayesiana è una proprietà della persona che assegna la probabilità all'evento. Per chiarire, consegnando una moneta a qualcuno e chiedendogli di assegnare una probabilità all'evento <<la moneta mostrerà 'testa' al prossimo lancio>>, questi, verosimilmente, risponderà ½. Se, invece, si convincesse la persona che la moneta è sbilanciata in favore di 'testa', egli assegnerebbe una probabilità più alta

[5] L'incertezza è imputabile alla comprensione imperfetta o alla conoscenza incompleta del dominio, alla casualità nel meccanismo che ne governa il comportamento ovvero ad una combinazione di questi fattori.
[6] L'inferenza probabilistica, in breve, determina quale sia la probabilità di un evento.
[7] In onore del Reverendo Thomas Bayes, uno scienziato della metà del '700 che diede significativi contributi alla teoria dell'inferenza probabilistica.
[8] Alcune nozioni (evento, ad esempio) e assiomi della teoria della probabilità sono riportati, in breve, nell'Appendice.

all'evento in base allo stato di conoscenza, ξ, acquisito. Per evidenziare l'approccio Bayesiano alla probabilità, anziché indicare la probabilità dell'evento x semplicemente come $p(x)$, la si indica con $p(x|\xi)$.

Un'importante differenza fra probabilità fisica e Bayesiana è che, per questa ultima, non si ha bisogno di tentativi ripetuti. Un esempio è fornito da domande del tipo: "che probabilità ha la Roma di vincere il campionato?" Lo statistico classico dovrebbe rimanere in silenzio, mentre il Bayesiano potrebbe assegnare una probabilità che rispecchi il proprio grado di conoscenza (ad esempio se la Roma ha giocatori migliori rispetto alle altre squadre e agli anni precedenti).

Una critica comune alla definizione Bayesiana della probabilità è l'*arbitrarietà*: perché il grado di fiducia dovrebbe rispettare le regole della probabilità? Con quali valori la probabilità potrebbe essere stimata? O meglio, ha senso assegnare una probabilità di uno (zero) ad un evento che (non) occorrerà e quale probabilità assegnare ai livelli di fiducia che non sono né l'evento certo né l'evento impossibile? Queste argomentazioni sono state oggetto di studio: molti ricercatori, sostenitori dell'approccio Bayesiano[9], hanno ricavato e dimostrato differenti proprietà che conducono, comunque, alle regole della probabilità.

Il *processo di stima del livello di fiducia* con cui esprimere la probabilità secondo l'approccio Bayesiano è noto come *probability assessment*: una tecnica molto semplice è la seguente. Si consideri una ruota con solo due regioni (ombra e non ombra), come quella illustrata in Figura 1. Assumendo che tutte le caratteristiche della ruota siano simmetriche (eccetto che per la zona in ombra), si conclude che la ruota ha uguale probabilità di trovarsi in qualsiasi posizione. Da questo giudizio e dalla regola della somma della probabilità, segue che la possibilità che la ruota si fermi nella regione "ombra" è uguale alla percentuale dell'area della ruota che è in ombra (0.3 per la ruota in figura). Questo approccio fornisce un riferimento per la misura delle probabilità relative ad altri eventi, associando, ad esempio, la zona in ombra al risultato che si prospetta essere il meno probabile.

[9] In tale proposito è opportuno osservare che l'approccio probabilistico ad un problema è un argomento ampiamente discusso in letteratura: in sintesi, nessun approccio rappresenta il "modello migliore" in assoluto ma può, invece, rappresentare la soluzione più adatta per un particolare problema.

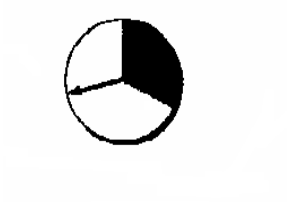


Figura 1 – Probability assessment

Un problema del probability assessment è la *precisione*: può una persona realmente indicare che la probabilità per un evento è 0.601 e non 0.599? In molti casi, no. D'altronde le probabilità spesso sono usate per prendere decisioni, quindi si usano tecniche di *analisi di sensitività* per stabilire il grado di precisione necessario. Un altro problema con il probability assessment è l'*accuratezza*, in quanto il modo con cui si pone una domanda può condurre ad assessment che non riflettono il reale livello di fiducia di una persona. [HEC94][HEC95]
Le probabilità quantificano l'incertezza associata al dominio codificato da una rete Bayesiana; l'aspetto qualitativo, il modello grafico, agevola l'interpretazione dei legami causali fra le variabili. Seguono, quindi, alcuni cenni sulla teoria dei grafi.

1.3.2 I MODELLI GRAFICI E CENNI ALLA TEORIA DEI GRAFI

I modelli grafici forniscono un'interfaccia intuitiva per modellare insiemi di variabili in condizioni di incertezza. Il fondamento dell'idea del modello grafico è la nozione di modularità: un sistema complesso è sviluppato unendo parti più semplici. La rappresentazione grafica costituisce la *sintassi* di un modello grafico mentre la *semantica* è espressa dalla teoria delle probabilità.
Un modello grafico è costituito da $K = \{1, 2, \ldots, k\}$ variabili (o nodi o vertici) con un insieme E di dipendenze (o collegamenti o *link* o archi) fra le variabili e un insieme P di funzioni distribuzione di probabilità per ogni variabile. Se due vertici (X_i, X_j) in un collegamento sono ordinati, allora si ha un *arco orientato*, in cui si esplicita la direzione ($X_i \rightarrow X_j$ o $X_i \leftarrow X_j$). Un *grafo orientato* è caratterizzato dall'avere tutti gli archi orientati. I grafi possono essere orientati, non orientati o parzialmente orientati.

Nei grafi non orientati la nozione di indipendenza è semplice: due nodi A e B sono condizionalmente indipendenti dati un terzo insieme C, se tutti i percorsi fra i nodi in A ed in B sono separati dal nodo C; nei grafi orientati la definizione di indipendenza deve considerare anche la direzione degli archi. I modelli orientati, presentano svariati vantaggi, quello più importante è l'opportunità di considerare un arco da A verso B come la codifica dell'asserto "A causa B".

Una *catena* è una serie di nodi in cui ogni nodo è collegato al precedente (non ha importanza la direzione del link); un *path* è una catena in cui l'arco ha sempre la stessa direzione, ad esempio $X_1 \rightarrow \ldots \rightarrow X_i \rightarrow \ldots \rightarrow X_k$. Un *ciclo* è un percorso (path) che inizia e termina nello stesso nodo. Un *grafo aciclico orientato*, **DAG (Direct Acyclic Graph)**, è un grafico orientato che non ha cicli. Una *relazione padre/figlio* si presenta quando vi è un collegamento del tipo $X_1 \rightarrow X_2$ (da X_1 a X_2) dove X_1 è padre di X_2 o, viceversa, X_2 è figlio di X_1. Una *relazione antenato/discendente* è un'estensione della relazione padre-figlio. Un grafo è detto *completo* se ogni nodo è connesso a tutti gli altri senza esplicitare alcuna direzione; si definisce *clique* (gruppo di oggetti) un sottoinsieme di nodi completo e che se ampliato, ad esempio con l'aggiunta di un nodo, perde la proprietà di completezza. [STE00]

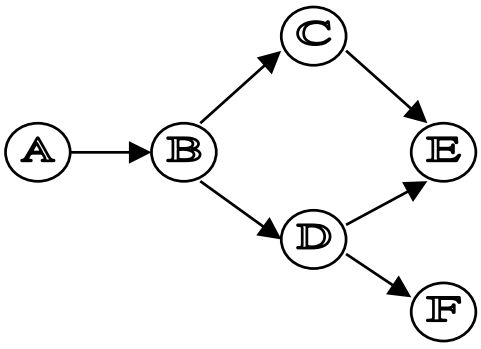


Figura 2 - A è padre di B, B è il padre di C e D. A è antenato di B,C,D,E,F.

Un possibile clique è A-B: difatti se aggiungessimo un nodo il grafo non sarebbe completo

1.3.3 DEFINIZIONE DI RETE BAYESIANA

Una *rete Bayesiana* è un grafico aciclico orientato (*directed acyclic graph – DAG* - ovvero tutti i percorsi sono orientati e non ci sono cicli) costituito dalla coppia (**S**,**P**):

1. **S** (***rappresentazione qualitativa***) - una struttura di rete che codifica
   - con dei nodi, le variabili casuali discrete (con un numero finito di stati) o continue (ad esempio con distribuzione gaussiana) del dominio $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$;
   - con degli archi orientati fra i nodi, l'insieme di asserzioni di indipendenza condizionata relative ad **X**;
2. **P** (***rappresentazione quantitativa***) - un insieme di distribuzioni di probabilità locali associate ad ogni variabile.

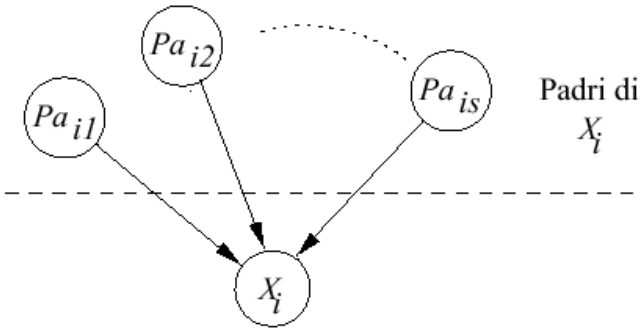


Figura 3 Relazione padre-figlio in una rete Bayesiana

La distribuzione di probabilità congiunta del dominio $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ assume quindi la seguente espressione:

$$p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i \mid \text{padri}(X_i))$$

I rami che collegano i nodi rappresentano le dipendenze causali fra le variabili. La semantica probabilistica espressa da queste strutture consente di quantificare queste dipendenze con la CPT di una variabile, dati i suoi padri. Nel caso di variabile discreta, ogni cella della CPT di un nodo esprime la probabilità

condizionata per lo stato di una variabile assegnata una *configurazione*[10] dei padri: *il numero di celle in una CPT per un nodo discreto uguaglia il prodotto fra il numero di valori (stati) assunti dalla variabile e il prodotto del numero degli stati dei padri*[11]. Se un nodo non ha padre (nessun collegamento punta ad esso), il nodo conterrà una tabella di probabilità marginale.

Nel caso di variabili continue si definisce la funzione di probabilità condizionata (*CPF)*, in genere di tipo gaussiano, e si associano al nodo parametri quali media e varianza che identificano la CPF. [HEC94][JEN96][BUN96][KRA98][STE00]

1.3.4 UN ESEMPIO

"Un giorno il signor Fletcher scopre che il suo albero di mele migliore perde le foglie e vuole capire perché ciò sta accadendo. Egli sa che se l'albero è secco (a causa della siccità) non c'è nessun mistero da svelare – è normale la perdita di foglie durante un periodo di siccità. D'altro canto l'albero potrebbe essere malato".

La situazione è modellata dalla BN in figura che consiste di tre nodi: **Sick** (malato), **Dry** (secco), and **Loses** (perdite), i quali possono tutti presentarsi in uno di due stati: **Sick** può essere "sick" o "not" - **Dry** può essere "dry" o "not" - e **Loses** può essere "yes" o "no". Il nodo **Sick** ci dice che l'albero è malato essendo nello stato "sick", altrimenti sarà nello stato "not". I nodi **Dry** e **Loses**, rispettivamente, ci dicono se l'albero è secco o sta perdendo le foglie.

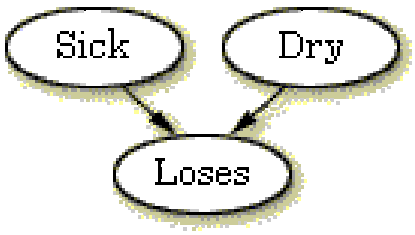


Figura 4 - Rete Bayesiana che rappresenta il dominio del problema

[10] A sua volta un nodo padre assume più valori (stati) e quindi una configurazione rappresenta un insieme dei possibili assegnazioni dei padri.

[11] Ad esempio, date tre variabili binarie (con due stati) A, B, C con A e B padri di C, le possibili configurazioni dei padri sono 4 mentre la CPT di C ha 8 celle.

Bisogna fare attenzione quando si modellano le dipendenze causali in una BN: talvolta non è abbastanza ovvio in quale direzione un collegamento debba puntare. Nel nostro esempio asseriamo che c'è una dipendenza causale da **Sick** a **Loses** perché quando un albero è secco potrebbe perdere le foglie. Ma non potremmo dire che quando l'albero perde le foglie, potrebbe essere secco e quindi disegnare il collegamento in senso opposto? No, in quanto è l'aridità che causa la perdita delle foglie e non viceversa. La figura precedente illustra la *rappresentazione qualitativa* della BN. Per poter definire una rete Bayesiana dobbiamo specificare anche la *rappresentazione quantitativa*, ovvero l'insieme delle CPT dei nodi. [HUG]

Sick="sick"	**Sick**="not"
0.1	0.9

Dry="dry"	**Dry**="not"
0.1	0.9

	Dry="dry"		**Dry**="not"	
	Sick="sick"	**Sick**="not"	**Sick**="sick"	**Sick**="not"
Loses="yes"	0.95	0.85	0.90	0.02
Loses="no"	0.05	0.15	0.10	0.98

Si osservi che tutte le tre tabelle mostrano la probabilità di un nodo di essere in uno specifico stato in dipendenza di una configurazione degli stati dei nodi padri; **Sick** e **Dry** non hanno nodi padri.

1.4 L'INFERENZA NELLE RETI BAYESIANE

1.4.1 L'INFERENZA STATISTICA

L'inferenza, in filosofia, indica un procedimento logico che permette di trarre una conclusione da determinate premesse. Allo stesso modo lo scopo dell'inferenza statistica è di estrarre un modello che consenta di trarre delle opportune conclusioni sull'andamento di una popolazione statistica (insieme di unità, individui, oggetti o altri enti in cui si manifesta il fenomeno che si studia). La conoscenza delle caratteristiche di un'intera popolazione spesso non è consentita sia per motivi di tempo che economici, quindi l'efficacia del processo di inferenza è nell'estendere a tutta la popolazione i risultati ottenuti su un campione limitato.
Un fenomeno viene rappresentato da un modello teorico o dalle caratteristiche di una popolazione; a partire dal modello la teoria della probabilità consente di prevedere il comportamento potenziale dei dati (problema diretto o deduttivo). Tutto ciò è la premessa necessaria per poter percorrere il cammino inverso: dai dati di un determinato campione si perviene al modello teorico (problema inverso o induttivo). L'inferenza rappresenta l'insieme dei metodi con cui si risolve il problema inverso.

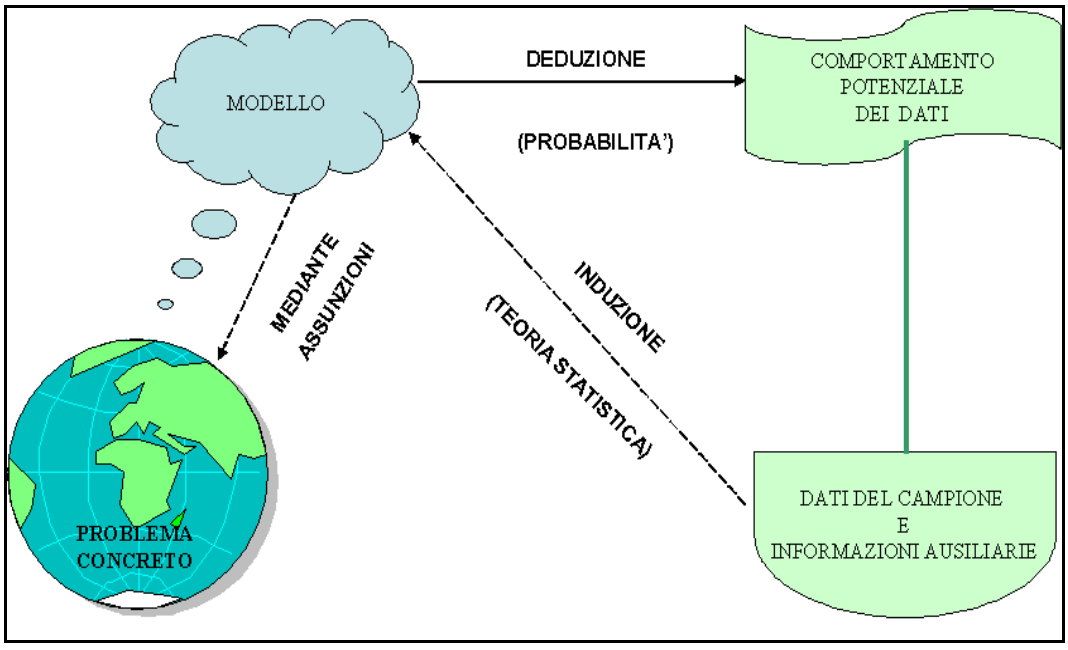


Figura 5 Problema diretto e problema inverso

1.4.2 I DIVERSI APPROCCI ALL'INFERENZA STATISTICA

I vari approcci all'inferenza statistica differiscono sia nella concezione stessa di inferenza che nelle singole tecniche di risoluzione dei problemi. Le diversità sono in parte legate ad interpretazioni differenti del concetto di probabilità e, in parte, sono dovute al tipo di obiettivi dell'inferenza statistica. Schematizzando, si può dire che tre sono gli approcci fondamentali.

1.4.2.1 CLASSICO

L'approccio classico, legato ai nomi di R. A. Fisher, J. Neyman, E.S. Pearson ed altri, include le tecniche, della stima puntuale e per intervallo, della verifica delle ipotesi, sviluppate a partire dalle distribuzioni campionarie delle statistiche. La concezione della probabilità è quella frequentista.. Il termine "classico" si giustifica con il fatto che i principi ed i metodi che lo caratterizzano sono ampliamente applicati e hanno preceduto (in termini, soprattutto, di sviluppo formale) dal punto di vista cronologico gli altri approcci.

1.4.2.2 BAYESIANO

L'inferenza Bayesiana, come quella classica, si articola nelle procedure di stima dei parametri e di verifica delle ipotesi, ma, a differenza di questa, utilizza nel processo di induzione, assieme ai dati del campione, le informazioni *a priori*. Le informazioni a priori vengono modificate dai dati del campione attraverso l'uso del teorema di Bayes, ovvero utilizzando lo stato di conoscenza. Questa impostazione non può basarsi unicamente su una visione frequentista delle probabilità; è inevitabile una interpretazione soggettivista.

1.4.2.3 TEORIA DELLE DECISIONI

La teoria delle decisioni è stata avviata dal lavoro di Wald (1950). L'obiettivo di questa impostazione è quello di stabilire delle regole di azione in situazioni di incertezza: le *regole di decisione*. In sintesi, il nucleo fondamentale consiste nella valutazione delle conseguenze di decisioni alternative espresse in termini di perdita o funzioni di perdita. La bontà di ogni regola di decisione, che si basa sui dati del campione e su informazioni a priori, viene misurata mediante la perdita

attesa o rischio. Alla teoria delle decisioni non è legata una particolare concezione della probabilità, tuttavia usufruendo anche di informazioni a priori, si fa riferimento a quella soggettivista. [CIC]

1.4.3 IL PROCESSO DI INFERENZA E LE RETI BAYESIANE

La semantica probabilistica delle reti Bayesiane e l'utilizzo del teorema di Bayes rendono agevole, a livello intuitivo, l'inferenza. Nota la rete (struttura, quindi i legami fra le variabili, e CPT per ogni nodo) possiamo introdurre l'evidenza in alcuni dei nodi ed osservare le variazioni, in termini probabilistici, della rete.

L'inferenza in una rete Bayesiana è il processo mediante il quale valutare la probabilità di ogni stato di un nodo quando le informazioni (evidenza) su altre variabili siano note (INFERENZA PROBABILISTICA).

Ricordando l'esempio dell'albero visto prima, supponiamo che l'albero stia perdendo le foglie, possiamo allora introdurre l'evidenza per il nodo/variabile **Loses** considerando che **Loses** = "yes". Quindi, applicando in modo opportuno il teorema di Bayes, possiamo stimare la probabilità che l'albero sia malato dalla probabilità **Sick** = "sick" e la probabilità dell'albero di essere secco come probabilità di **Dry** = "dry".

L'elaborazione delle probabilità di altre variabili (ma non tutte) data l'evidenza è nota come *Belief Updating*. Un'informazione più completa consiste nell'aggiornare, come fatto nell'esempio precedente, le probabilità degli stati di tutte le variabili casuali (della rete) data l'evidenza: questo processo è noto, invece, come *Belief Revision* e fornisce la configurazione di stati più probabile.

L'inferenza eseguita con i metodi Bayesiani elabora i dati in modalità:

- *batch*: completamente, come fanno anche i metodi classici. Quindi tutti i campioni costituiscono l'evidenza per un unico processo di inferenza.
- *sequentially*: considera un'osservazione alla volta. Ogni campione acquisito costituisce l'evidenza di un processo di inferenza.

In ogni caso il risultato finale sarà lo stesso. La natura incrementale (sequentially) è un ulteriore vantaggio dei metodi Bayesiani; l'acquisizione di nuovi dati non richiede difatti che, per il processo di inferenza, vadano rielaborati quelli

considerati in precedenza. Inoltre permette una maggiore flessibilità nel poter effettuare un controllo durante l'elaborazione.
Il processo associato all'inferenza è semplice quando sono disponibili tutte le evidenze sulle variabili antenate di un nodo (sono note cioè le cause ovvero la probabilità a priori). Quando l'evidenza è disponibile sui discendenti (effetto o probabilità a posteriori) delle variabili di interesse, bisogna ragionare nella direzione opposta a quella espressa dai rami della rete, e la metodologia da usare è meno intuitiva (il fondamento è comunque il teorema di Bayes).

Per organizzare un problema di inferenza bisogna:

1. identificare in modo corretto gli obiettivi del modello (ad esempio, se deve essere usato per una predizione o una classificazione);
2. identificare quante più osservazioni possibili rilevanti per il problema;
3. determinare quale sottoinsieme di queste osservazioni bisogna modellare;
4. organizzare le osservazioni in variabili aventi stati mutuamente esclusivi e collettivamente esaustivi.

Nel caso della rete Bayesiana, fissato un dominio **X** bisogna determinare la struttura della rete. In tale proposito, una fase fondamentale è il corretto ordinamento delle variabili $\mathbf{X} = \{X_1, X_2, \ldots, X_i, \ldots, X_n\}$, in modo da pervenire ad una relazione del tipo $p(X_i|X_1,\ldots,X_{i-1}) = p(X_i|\text{Padri}(Xi))$[12], $\forall$ i =1,…,n. Nello specifico, consideriamo la *chain-rule*:

$$p(X_1,...,X_n \mid \xi) = \prod_{i=1}^{n} p(X_i \mid X_1,...,X_{i-1},\xi) \qquad (1.1)$$

Per ogni X_i, esisterà un sottoinsieme $\pi_i \subseteq \{X_1,...,X_n\}$ tale che X_i e $\{X_1,...,X_{i-1}\} \setminus \pi_i$ risultano condizionatamente indipendenti dato π_i, quindi:

$$p(X_1,...,X_{i-1}) = p(X_i \mid \pi_i) \qquad (1.2)$$

[12] Notazioni equivalenti saranno $p(X_i|\text{Padre}_i)$, $p(X_i|\mathbf{Pa}_i)$, $p(X_i|\pi_i)$.

Combinando le equazioni precedenti, otteniamo:

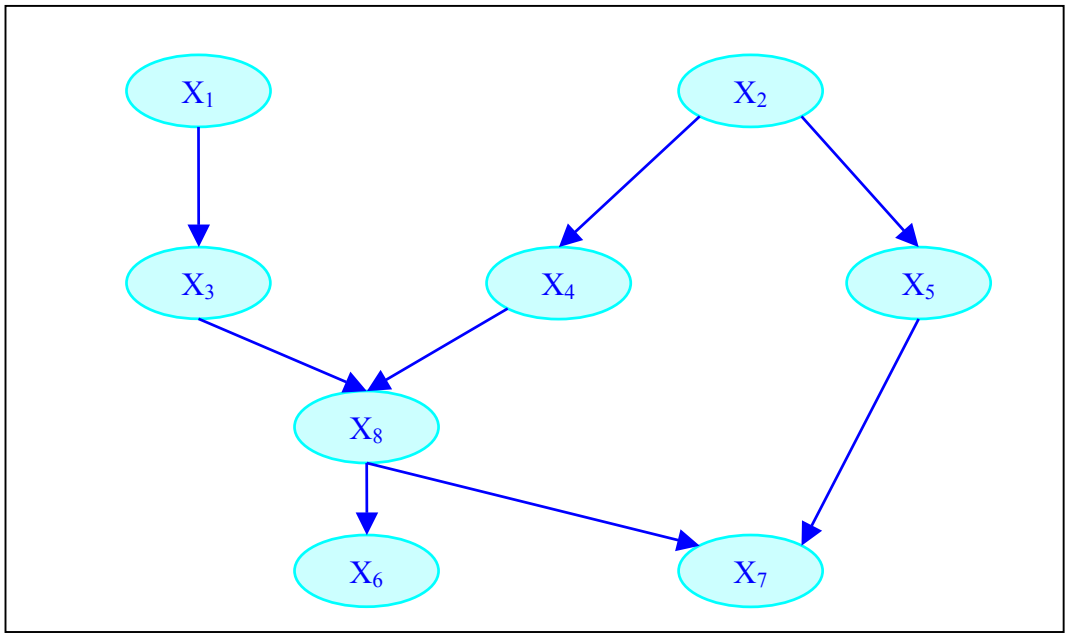
$$p(\boldsymbol{X} \mid \xi) = \prod_{i=1}^{n} p(X_i \mid \pi_i, \xi) \qquad (1.3)$$

Confrontando la (1) e la (3), osserviamo che l'insieme di variabili (π_i ,…, π_n) corrisponde ai padri della rete Bayesiana (Pa_1,…,Pa_n), che di volta in volta specificano completamente gli archi della struttura della rete. Di conseguenza, è evidente l'importanza di ordinare le variabili in modo opportuno. Infatti, se l'ordinamento delle variabili non è accurato, la rete ottenuta potrebbe fornire troppe dipendenze condizionate fra le variabili rendendo meno chiara la rappresentazione grafica. Tuttavia, considerando che

- spesso è facile riconosce *relazioni causali* fra variabili,
- le *relazioni causali* tipicamente corrispondono ad asserzioni di indipendenza condizionale,

la scelta di un corretto ordinamento non diventa così aleatoria.

Il passo finale per la costruzione della rete Bayesiana, ovvero del modello da usare per l'inferenza, è rappresentato dalla stima delle distribuzioni di probabilità locali $p(X_i|Padre_i)$ $\forall$ i =1,…,n. In tale proposito è opportuno osservare l'indubbio vantaggio delle reti Bayesiane nel semplificare il numero di probabilità da computare. Supponiamo, infatti, di avere un generico dominio $\mathbf{X} = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$ di variabili binarie: per stimare la distribuzione della probabilità congiunta p(**X**) bisognerebbe elicitare $2^7 - 1 = 127$ parametri (il "-1" segue dalla regola della somma delle probabilità) . Consideriamo, poi, la rete Bayesiana nella figura seguente che esemplifica il dominio; ipotizziamo, inoltre, l'aggiunta di una variabile X_8 (variabile nascosta - hidden variable - perché non presente nel dominio di partenza **X**) che renda più immediata la comprensione di alcune relazioni. Ebbene, nonostante l'aggiunta di questa ultima, per elicitare p(**X**), in base alla (1), sono necessarie soltanto 18 stime. [HEC94][JEN96][BUN96][KRA98][STE00]



$$p(\mathbf{X}) = p(X_1)\, p(X_2)\, p(X_3 \mid X_1)\, p(X_4 \mid X_2)\, p(X_5 \mid X_2)\, p(X_8 \mid X_3, X_4)\, p(X_6 \mid X_8)\, p(X_7 \mid X_8, X_5)$$

parametro	numero di probabilità da stimare
$p(X_1)$	1
$p(X_2)$	1
$p(X_3 \mid X_1)$	2
$p(X_4 \mid X_2)$	2
$p(X_5 \mid X_2)$	2
$p(X_6 \mid X_8)$	2
$p(X_8 \mid X_3, X_4)$	4
$p(X_7 \mid X_8, X_5)$	4

1.4.3.1 UN ESEMPIO DI INFERENZA

Per dare un taglio pratico a quanto esposto finora, si consideri il seguente esempio (tratto da "A Tutorial on Learning with Bayesian Networks" di D. Heckerman [HEC95]) che modella un problema di frode con carta di credito. Le variabili del dominio sono:

Fraud (f): indica che la persona commette una frode con carta di credito;
Jewerly (j): indica l'acquisto fraudolento di gioielli nelle ultime 24 ore;
Gas (g): indica l'acquisto fraudolento di carburante nelle ultime 24 ore;

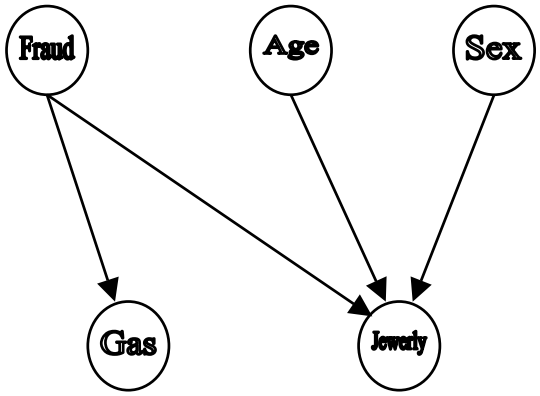
Age (a): determina l'età del possessore della carta di credito;

Sex (s): determina il sesso del possessore della carta di credito.

Un ordinamento ottimale, in base alla considerazioni fatte prima, è f,a,s,g,j che rispecchia le seguenti relazioni di indipendenza condizionata:

$$p(a|f) = p(a)$$
$$p(s|f,a) = p(s)$$
$$p(g|f,s,a) = p(g|f)$$
$$p(j|f,a,sg,) = p(j|f,a,s)$$

La rete Bayesiana associata a tale problema è mostrata di seguito.



Dalla rete Bayesiana si può osservare, per esempio, che se c'è un caso di frode da carta di credito, allora l'acquisto di gioielli o gas ne è condizionato; inoltre la possibilità che un gioiello sia acquistato è influenzata dall'età e dal sesso dell'acquirente.

La rappresentazione quantitativa è associata alla tabella seguente tratta dall'articolo su citato.

Probability			Conditions		
			Fraud	Age	Sex
Fraud = Yes	Fraud = No				
0.00001	0.99999		–	–	–
Age < 30	Age = 30-50	Age > 50			
0.25	0.40	0.35	–	–	–
Sex = Male	Sex = Female				
0.5	0.5		–	–	–
Gas = Yes	Gas = No				
0.2	0.8		Yes	–	–
0.01	0.99		No	–	–
Jewelry = Yes	Jewelry = No				
0.05	0.95		yes	*	*
0.0001	0.9999		no	<30	male
0.0004	0.9996		no	30-50	male
0.0002	0.9998		no	>50	male
0.0005	0.9995		no	<30	female
0.002	0.998		no	30-50	female
0.001	0.999		no	>50	female

Variable	*Fraud*	*Jewelry*	*Gas*	*Sex*	*Age*
Value	*?*	*Yes*	*No*	*Male*	*< 30*

Dalla tabella si può osservare, invece, che se Fraud è nello stato "Yes" allora c'è una probabilità pari a 0.2 che Gas sarà "Yes" ma se Fraud è "No" allora ci sarà una probabilità minore, 0.01, che Gas sia "Yes". In parole: se una persona sta usando in modo fraudolento una carta di credito è 20 volte più probabile che compri del gas rispetto a chi usa legittimamente la carta di credito.

Il compito dell'inferenza è determinare l'aggiornamento (a posteriori) della distribuzione di probabilità per una o più variabili del dominio basandosi sui valori noti dalle osservazioni (evidenza).

In riferimento all'esempio introdotto sopra, come si evince dal dato riportato in tabella, si è rilevato (evidenza) che un giovane maschio sta usando una carta per comprare dei gioielli ma non del gas (Sex = Male, Age = <30, Jewerly = Yes, Gas = No): possiamo inferire, data l'evidenza, per determinare se l'acquisto sia fraudolento e con quale probabilità.

In altre parole dobbiamo valutare la probabilità P(f|j,g,s,a). Dal teorema di Bayes:

$$P(f \mid j,g,s.a) = \frac{P(j,g,s,a,f)}{P(j,g,s,a)}$$

Poiché gli stati di f (indicati con f') sono mutuamente esclusivi ed esaustivi, possiamo trasformare il denominatore nel modo seguente

$$P(f \mid j,g,s.a) = \frac{P(j,g,s,a,f)}{\sum_{f'} P(j,g,s,a,f')}$$

Usando la regola della catena possiamo scomporre in fattori numeratore e denominatore

$$P(f \mid j,g,s.a) = \frac{P(j \mid g,s,a,f) * P(g \mid s,a,f) * P(s \mid a,f) * P(a \mid f) * P(f)}{\sum_{f'} P(j \mid g,s,a,f') * P(g \mid s,a,f') * P(s \mid a,f') * P(a \mid f') * P(f')}$$

Poiché alcune variabili sono fra loro *indipendenti* (se X_0 non è il figlio di X_n $P(X_0|X_1,\ldots,X_n,\ldots,X_k) = P(X_0|X_1,\ldots,X_{n-1},X_{n+1},\ldots,X_k)$), effettuando le opportune semplificazioni risulta

$$P(f \mid j,g,s.a) = \frac{P(j \mid s,a,f) * P(g \mid f) * P(s) * P(a) * P(f)}{\sum_{f'} P(j \mid s,a,f') * P(g \mid f') * P(s) * P(a) * P(f')}$$

che può essere valutata leggendo dalla tabella i valori relativi alle probabilità e all'evidenza

$$P(f = Yes \mid j = Yes, g = No, s = Male, a = <30) =$$
$$\frac{P(j = Yes \mid s = Male, a = <30, f = Yes) * P(g = No \mid f = Yes) * P(f = Yes)}{\sum_{f' \in (yes,no)} P(j = Yes \mid s = Male, a = <30, f') * P(g = No \mid f') * P(f')}$$

Il risultato è $P(f = Yes \mid j = Yes, g = No, s = Male, a =< 30)$=0.00402: quindi mentre la probabilità a priori era di 0.00001 la probabilità a posteriori, dopo il processo di inferenza, è 0.00402: l'evento *f = yes* risulta 400 volte più probabile.

1.4.4 ALGORITMI PER L'INFERENZA NELLE BN

Molti ricercatori hanno sviluppato algoritmi per l'inferenza probabilistica in reti Bayesiane con variabili discrete.

Per esempio, Howard e Matheson prima (1981), Olmsted, Shachter in seguito, hanno sviluppato un algoritmo che ribalta gli archi nella struttura della rete fino a che la risposta alla richiesta ("query") di una data probabilità non possa essere letta direttamente dal grafo. In questo algoritmo, ogni ribaltamento di un arco corrisponde all'applicazione del teorema di Bayes.

Pearl (1982) ha sviluppato uno schema di scambio di messaggi che aggiorna le distribuzioni di probabilità per ogni nodo in una rete Bayesiana in risposta alle osservazioni di una o più variabili (J. Pearl "Reverend Bayes on inference engines: A distributed hierarchical approach" e J.H. Kim e J. Pearl "A computational model for combined causal and diagnostic reasoning in inference systems"). Questo approccio inizialmente concepito per reti tree-structured è stato poi esteso a BN generiche da Lauritzen e Spiegelhalter (1988) attraverso il metodo *join tree propagation* (S.L. Lauritzen e D.J. Spiegelhalter "Local computations with probabilities on graphical structures and their application to expert systems").

Altri studiosi quali Jensen [JEN96] e poi Dawid hanno contribuito a perfezionare questo algoritmo, più noto come *junction tree* (fra i più usati negli applicativi software) che, per semplificare il processo di inferenza, trasforma la rete Bayesiana in un albero i cui nodi corrispondono ad un sottoinsieme di variabili del dominio.

Un altro metodo molto utilizzato per l'inferenza è il *bucket elimination* dovuto a R. Detcher, in cui si attua una procedura di eliminazione delle variabili.

E' facile comprendere che il processo di inferenza applicato alle reti Bayesiane, con l'obiettivo di soddisfare tutte le possibili "query", è complesso: in tale proposito si ricorre anche a tecniche approssimate basate su simulazioni di Monte Carlo (J. Pearl "Evidential reasoning using stochastic simulation of causal models") che forniscono miglioramenti graduali dei risultati all'aumentare del numero di campioni disponibili. Riferimenti più dettagliati sull'inferenza e le BN sono presenti nelle opere di Heckerman [HEC94] e in "A survey of algorithms for real – time Bayesian Network inference" di H.Guo e W. Hsu [GUO02].

1.5 LEARNING BAYESIAN NETWORKS

Le reti Bayesiane, nell'ambito dell'intelligenza artificiale e dei sistemi esperti, sono un valido strumento per una rappresentazione compatta di un dominio concepito in virtù della conoscenza degli esperti. L'acquisizione dei dati, per gli ingegneri della conoscenza, rappresenta un problema, specie se condotta manualmente, perché soggetta ad errori, imprecisa e dispendiosa in termini di tempo.

La semantica probabilistica e statistica delle reti Bayesiane ha costituito la base per lo sviluppo di metodi per l'apprendimento delle reti Bayesiane direttamente dai dati. Il processo di learning, piuttosto che dal parere degli esperti, è realizzato da sistemi intelligenti che "imparano" analizzando semplicemente il database di campioni delle variabili di un dominio, estrapolando le informazioni necessarie (è un primo passo verso il processo di data mining, una delle tecniche all'avanguardia di Knowledge Discovery).

Gli algoritmi per espletare un processo di inferenza su reti Bayesiane sono stati ampiamente diffusi, studiati e continuamente migliorati. Invece, negli ultimi anni è aumentato l'interesse, quindi la ricerca e lo sviluppo, per automatizzare, migliorare (meno errori) e rendere più veloce il processo di learning di reti Bayesiane. L'apprendimento di reti probabilistiche comprende:

- **learning structure** (o **structural learning**): apprendere la struttura della rete ovvero le relazioni fra le variabili;
- **learning parameters**: apprendere i parametri[13]; apprendimento delle probabilità condizionate. In particolare si presentano due possibili situazioni:
    1. **known structure**: la struttura della rete è nota, per esempio è fornita da un esperto.
    2. **unknown structure**: bisogna apprendere prima la struttura della rete e quindi i parametri.

[13] Per parametro si intende una costante che caratterizza la funzione di probabilità o di densità di una variabile casuale, ad esempio λ nella distribuzione di Poisson. Poiché le reti Bayesiane studiate sono caratterizzate da variabili discrete, l'accezione di parametro indica, semplicemente, la probabilità (eventualmente condizionata) ovvero $\theta = p\ (X = x)$.

Inoltre il learning risulta più complesso in presenza:

- variabili nascoste (***hidden variable***), ovvero variabili che non sono esplicitate fra quelle del dominio e che se evidenziate, spesso, ne semplificano lo studio;
- valori dispersi o non rilevati (***missing value***); in tale proposito bisogna effettuare una stima di questi valori prima di procedere con l'apprendimento.

[HEC94][HEC94][BUN96][KRA98]

Anche per i casi ora menzionati, le metodologie statistiche e la teoria della probabilità sono alla base di algoritmi, alcuni cronologicamente recenti, per risolvere il problema dell'apprendimento. Il learning costituisce l'aspetto principale di questo lavoro di tesi per cui sarà approfondito nei capitoli successivi.

1.6 STATO DELL'ARTE

Nei paragrafi seguenti saranno evidenziate alcuni varianti, presenti in letteratura, delle reti Bayesiane che rendono la tecnologia Bayesiana uno strumento sempre più flessibile ed adatto per il *problem solving*.

1.6.1 CAUSAL NETWORK (CN)

Le reti causali modellano un dominio come un insieme di meccanismi stabili che possono essere riconfigurati da interventi che mutano localmente il modello (J. Pearl "Causation, action, and counterfactuals"). Per semplificare, consideriamo il seguente esempio. "Una persona abita in una villetta circondata da un giardino e con un viale pavimentato. Affacciandosi alla finestra osserva che il viale è bagnato, quindi percepisce il pericolo di scivolare, a seconda che la causa sia l'annaffiatore in azione o anche una pioggia leggera". La figura seguente illustra questo semplice esempio descrivendo le relazioni di tipo causale fra la "stagione" (X_1 - season), "il tipo di pioggia" (X_2 – rain), "annaffiatore attivo?" (X3 - sprinkler), "pavimento bagnato" (X4 - wet) e "pavimento scivoloso" (X5 - slippery). L'assenza di un legame diretto fra X_1 e X_5, per esempio, esprime la mancanza di influenza *diretta* fra la stagione e un pavimento scivoloso, difatti la causa della variabile slippery è wet.

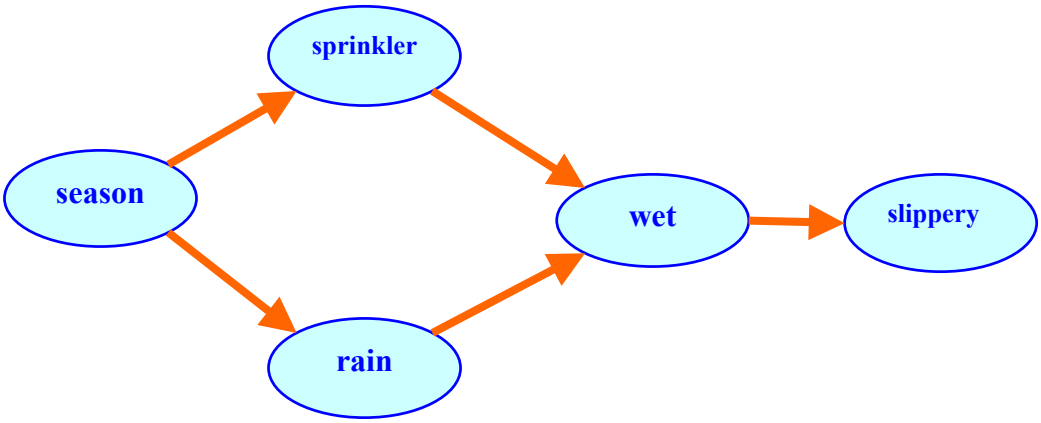


Figura 6 - Rete Sprinkler

La distribuzione di probabilità congiunta è

$$p(x_1,x_2, x_3,x_4,x_5) = p(X_1 = x1, X_2 = x_2, X_3 = x_3, X_4 = x_4, X_5 = x_5) =$$
$$p(x_1)\, p(x_2|x_1)\, p(x_3|x_1)\, p(x_4|x_2,x_3)\, p(x_5|x_4)$$

La maggior parte dei modelli probabilistici, incluse le BN, descrive una distribuzione su degli eventi osservabili ma non codifica l'effetto che è successivo ad un intervento/azione sul dominio. Per esempio, cosa avverrà dopo avere azionato l'annaffiatore? Quale effetto avrà questa azione sulla variabile season, o sulla connessione fra wet e slippery?
Una *rete causale* è una BN con la proprietà implicita che i padri di ogni nodo sono le cause dirette della presenza di un variabile.
Il risultato di un intervento quindi diventa esplicito: se "X_3 = on" allora scompare il legame con X_1 (l'azionamento dello sprinkler elude ogni dipendenza dalla stagione dell'anno: infatti in estate è più probabile che l'annaffiatore sia acceso), per cui:

$$p(x_1,x_2,x_4,x_5) = p(x_1)\, p(x_2|x_1)\, p(x_4|x_2,X_3 = on)p(x_5|x_4)$$

La sottile differenza (fra BN e CN) sta fra "l'agire" e "l'osservare": se avessimo semplicemente osservato "X3 = on" *ma non agito*, l'espressione precedente andrebbe modificata aggiungendo il fattore $p(X_3 = on \mid x_1)$; un'arbitraria azione sul dominio, invece, implica l'eliminazione di un legame.
Le *causal network* sono più propriamente definite come BN nelle quali il corretto modello probabilistico è reso rimuovendo il legame dai padri dopo l'introduzione dell'evidenza in un nodo.
La nozione di causalità nelle CN consente di esprimere anche semplici intuizioni: Fire →Smoke è una rete causale mentre non lo è Smoke → Fire seppure entrambe consentano, comunque, di elicitare la stessa distribuzione congiunta. [PEA00]

1.6.2 DYNAMIC BAYESIAN NETWORKS

Le BN, come quelle descritte finora, sono conosciute come *Static Bayesian Networks* (*SBN*). Le SBN sono concepite per rappresentare la situazione corrente e non per modellare esplicitamente sequenze temporali: il passato è ignorato e il futuro non è predetto. Per esempio, nella figura seguente, ci sono due malattie (**D1** e **D2**) che possono causare sintomi differenti (**S1** e **S2**). Usando le informazioni disponibili per i sintomi è possibile predire la probabilità di ogni malattia.

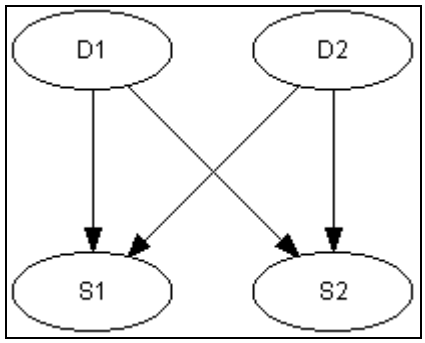


Figura 7 - Static Bayesian Network

In molti problemi è quasi inconcepibile rappresentare i dati e ragionare senza usare una dimensione temporale, poiché le situazioni evolvono nel tempo. La SBN, come quella rappresentata sopra in figura, non può essere usata e quindi la rete deve essere estesa per esprimere anche le informazioni temporali. Il modo più semplice di estendere una SBN è di includere istanze multiple che rappresentino la SBN in time slice differenti, e collegarle insieme: i modelli grafici risultanti sono noti come Dynamic Bayesian Networks (DBN). Un insieme di variabili $\mathbf{X}_t$ indica lo stato del dominio nell'istante *t* ed un insieme di variabili "sensori " $\mathbf{E}_t$ indica le osservazioni disponibili all'istante *t*. Il modello del sensore è codificato dalla distribuzione di probabilità condizionata per le variabili osservabili $p(\mathbf{E}_t|\mathbf{X}_t)$. Il modello di transizione $p(\mathbf{X}_{t+1}|\mathbf{X}_t)$ relaziona invece lo stato in *t* a quello in *t+1* (Thomas Dean e Keiji Kanazawa "A model for reasoning about persistence and causation").

Per esempio, la rete di Figura 8 è ottenuta collegando istanze multiple del modello di Figura 7. Usando questa nuova rete, è possibile predire l'evolversi delle malattie. [HUG][PEA00]

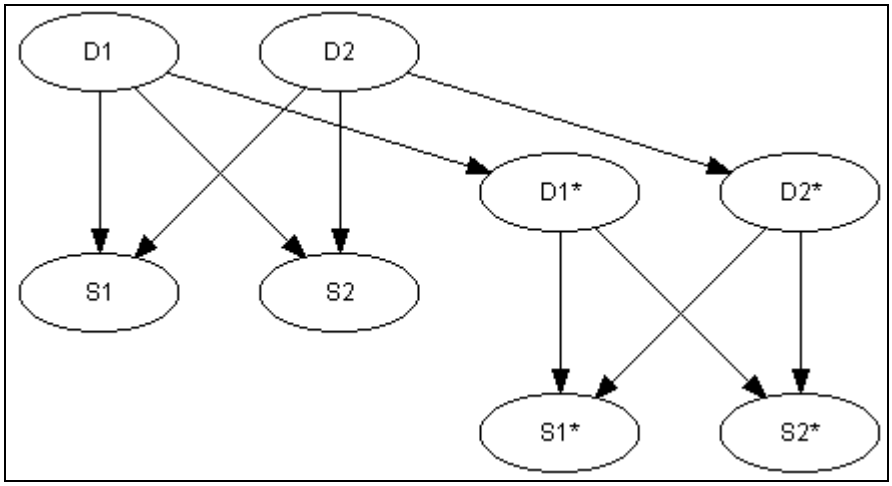


Figura 8 – Dynamic Bayesian Network

1.6.3 INTRODUZIONE AI DIAGRAMMI DI INFLUENZA

Un diagramma di influenza (*Influence Diagram* - **ID**) è una rete Bayesiana migliorata per modellare delle alternative in un processo di decisione (*decision making*).

In alcuni casi è possibile costruire un modello per il decison making con una BN pura, in altri è più indicato l'uso di un diagramma di influenza: la BN viene estesa con i nodi di *utility* e *decision* (*chance node* sono, invece, i nodi delle reti viste finora). I collegamenti che coinvolgono i nodi di decisione indicano una *precedenza temporale*, un collegamento da una variabile casuale ad una variabile di decisione indica che il valore della casuale risulta noto quando si prende una decisione a riguardo; un collegamento da una variabile di decisone ad un'altra indica l'ordine cronologico del decision making. La rete deve essere aciclica, e deve esistere un percorso diretto che contiene tutti i nodi di decisione. Un nodo utility ha una funzione di utilità che, per ogni configurazione degli stati dei suoi nodi padri, associa una funzione opportuna (i nodi di utility non hanno figli).

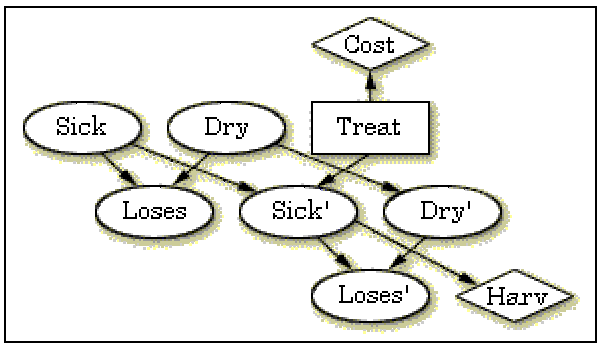


Figura 9 – Esempio di Influence Diagram

Nella figura precedente è rappresentato un semplice ID che ripropone un esempio visto in precedenza. In particolare, il signor Fletcher vuole decidere se effettuare un trattamento per la cura degli alberi dopo la raccolta. Dovendo modellare un'evoluzione temporale ("dopo la raccolta") si usa una DBN. Il decision node "Treat" (rettangolo) è associato al nodo Sick' perché solo in caso di malattia della pianta ha senso intervenire. L'entità dell'intervento, in termini di costo, viene associata al nodo utilitiy "Cost" (rombo). [HUG]

1.6.4 OBJECT-ORIENTED NETWORKS

Un *Object-Oriented Network* è una rete (ad esempio una rete Bayesiana o un diagramma di influenza) che, in aggiunta ai soliti nodi, contiene *instance nodes* (nodi di istanza) che *rappresentano l'istanza di un'altra rete*. Naturalmente, un nodo di istanza può contenere, a sua volta, nodi di istanza, per cui una rete orientata agli oggetti può essere vista come una descrizione gerarchica del dominio di un problema. Alcuni aspetti, riportati di seguito, evidenziano l'utilità di costruire una rete con nodi di istanza:

1. La costruzione del modello spesso implica *ripetuti cambiamenti nel livello di astrazione*, seguendo un approccio di top-down, bottom-up o una combinazione di entrambi nel processo di ragionamento.
2. I modelli dei sistemi spesso contengono *strutture ripetitive*.

3. Descrivere un modello in modo gerarchico garantisce *meno disordine* nel grafo per cui si dispone di uno strumento agevole per comunicare idee fra gli ingegneri della conoscenza e gli utenti. [HUG]

1.6.5 FUNCTION BAYESIAN NETWORKS

Le reti Bayesiane e le CN supportano sia il ragionamento data l'evidenza che la determinazione di un effetto data un'azione. Esaminiamo però la seguente query, riferita all'esempio dell'annaffiatore, per un eventuale processo di inferenza: "determinare la probabilità che il pavimento *non* sarebbe stato scivoloso se avessimo avuto lo sprinkler off, data l'evidenza che lo sprinkler sia on e che il pavimento sia scivoloso". E' impensabile fornire una risposta con le sole informazioni della rete e della distribuzione congiunta: per elaborare informazioni *counterfactual* ("in contrapposizione all'evidenza") è necessario un'ulteriore rifinitura in una BN. L'elaborazione di probabilità counterfactual richiede che le reti siano rappresentate anche con una forma funzionale, ovvero ogni probabilità condizionata $p(x_i|pa_i)$ è sostituita da una relazione $x_i = f_i(pa_i,\varepsilon_i)$, dove ε_i è un errore stocastico non osservato. Noti la funzione f_i ed ε_i, tutte le asserzioni counterfactual possono essere espresse da un'unica probabilità, usando la propagazione dell'evidenza in una struttura chiamata "twin network". Quando è disponibile solo una parziale conoscenza della forma funzionale f_i si impongono dei limiti sulle probabilità associate alle asserzioni counterfactual. (A. Balke e J. Pearl "Counterfactuals and policy analysis in structural models" - J. Pearl "Causality: models, reasoning, and inference") [PEA00]

1.6.6 CAUSAL DISCOVERY

Negli ultimi anni una delle prospettive più interessanti delle BN è la possibilità di usarle per scoprire strutture causali dai dati grezzi (semplici osservazioni relative ad un dominio), un compito considerato finora impossibile senza un'opportuna elaborazione precedente sui dati.

Ad esempio, se chiedessimo ad una persona di fornire una possibile struttura che evidenzi l'insieme di asserzioni fra i tre eventi A,B,C seguenti

A e B sono dipendenti - B e C sono dipendenti - A e C sono indipendenti

(per esempio A e C potrebbero essere il risultato del lancio di due monete e B un campanello che suona se entrambe le monete mostrano testa) molto probabilmente la risposta sarebbe "A e C come cause indipendenti e B come loro effetto comune"

$$A \rightarrow B \leftarrow C$$

Le stesse asserzioni però potrebbero essere espresse da uno scenario, matematicamente fattibile ma non naturale, in cui B è la causa e A e C sono gli effetti. La validazione di una struttura individuata è legata ad una stima precisa delle probabilità dai dati; una variazione delle probabilità (e quindi della robustezza delle relazioni apprese) implicherebbe anche una modifica dei legami presenti nel modello grafico. L'idea nella causal discovery è proprio di confrontare più pattern in modo da prediligere la struttura più compatibile e stabile rispetto ai dati. In sintesi, questo è il compito dello Structural Learning a cui si è accennato in precedenza. [PEA00]

1.6.7 MODELS OF COGNITION

Nel campo dell'Intelligenza Artificiale le reti Bayesiane sono state principalmente impiegate per la creazione di Intelligent Reasoning Systems, in particolare di sistemi esperti. Le BN offrono un modo naturale per la codifica dell'incertezza in un expert system.

1.6.7.1 I SISTEMI ESPERTI

Un esperto, in generale, è una persona che mette a disposizione la sua conoscenza su un particolare dominio per indicare quale sia l'azione più opportuna da intraprendere (*decision making*): si pensi ad un medico, ad un operatore finanziario, un consulente informatico.

Il lavoro di un esperto può essere così sintetizzato:

1. Stabilire lo stato del dominio (ad esempio, un medico usa la cartella clinica ed annota i sintomi del paziente per una diagnosi).
2. In base allo stato si decide un'azione (il medico decide la cura per il paziente).
3. Per qualsiasi azione l'esperto ha delle aspettative che possono verificarsi o meno. I risultati dell'azione, in ogni caso, arricchiscono la sua conoscenza, per interventi successivi (nell'esempio del medico, si apprende l'efficacia di una cura per una malattia).

Figura 10 – Il processo decisionale di un esperto

Lo sviluppo tecnologico e l'informatica hanno reso possibile la creazione di *sistemi esperti*, costruiti alla fine degli anni '60, così chiamati proprio perché concepiti come un modello computerizzato dell'esperto.

1.6.7.1.1 I sistemi rule-based

Una regola è un'espressione della forma

if *A* **then** *B*

dove *A* è un'asserzione/condizione e *B* può essere o un'azione o, a sua volta, un'altra asserzione.

Per esempio, le seguenti tre condizioni potrebbero essere una parte di un insieme più ampio di regole per la risoluzione di problemi relativi ad un sistema di pompe:

1) **If** (rottura nella pompa) **then** (la pressione è bassa)
2) **If** (rottura nella pompa) **then** (controllare il livello dell'olio)
3) **If** (perdita di potenza) **then** (rottura nella pompa).

Un *sistema rule-based* è un sistema esperto costituito da:

- una *base di conoscenza* (***knowledge base***), ovvero un insieme di regole che riflettono le relazioni essenziali presenti nel dominio ovvero *i modi di ragionare sul dominio*;
- un *sistema di inferenza (**inference system***): quando sono disponibili delle informazioni specifiche sul domino (osservazioni), si usano le regole e le osservazioni per pervenire a conclusioni sullo stato del dominio e per determinare appropriate azioni. Questo processo è noto come *inferenza*.

L'inferenza agisce come una 'reazione a catena': difatti, nell'esempio precedente, se ci viene detto che c'è una perdita di potenza, la regola 3) afferma che vi è anche una rottura nella pompa mentre la regola 1) ci dice che la pressione è bassa. La regola 2) invece raccomanda di controllare il livello dell'olio. Inoltre le regole possono anche essere usate nella direzione opposta: se la pressione è bassa, allora la regola 2) ipotizza una perdita di olio.

1.6.7.2 L'INCERTEZZA NEI SISTEMI ESPERTI

Le relazioni espresse dalle regole non sono assolutamente certe; le *informazioni* ottenute sono spesso soggette ad incertezza (le osservazioni possono essere incerte, le informazioni incomplete o non deterministiche: si pensi ad un paziente che afferma genericamente di "accusare un dolore alla testa") e le *relazioni* nel dominio non sono univoche (spesso gli stessi sintomi sono dovuti a malattie diverse). Il sistema di inferenza è ampliato con nuove asserzioni che consentano un ragionamento coerente in condizioni di incertezza: si associa una misura dell'incertezza alle regole del sistema, una regola fornirà una funzione (*f(x)*) che descrive quanto un cambiamento nell'evento (A) cambierà la certezza della conclusione (B). Nella forma più semplice la regola sarà così modificata

If *A* (con certezza *x*) **then** *B* (con certezza *f(x)*)

Ci sono molti schemi per trattare l'incertezza nei sistemi rule-based. I più comuni sono la *logica fuzzy, certainty factors,* e *Dempster-Shafer belief functions*. Il fattore comune a questi schemi è che la certezza è trattata *localmente*, cioè il trattamento è connesso direttamente alle regole in entrata e all'incertezza dei loro elementi.
Si immagini, ad esempio, che in aggiunta alla regola precedente si abbia la regola

If *C* (con certezza *x*) **then** *B* (con certezza *g(x)*)

con un legame tra *A* e *C* non esplicitato. Si vuole rispondere alla domanda

"Se *A* si verifica con certezza *a* e *C* con certezza *c*,
allora quale è la certezza di *B"*?

In tale proposito, un fattore comune alle differenti algebre per combinare le incertezze è che, in molti casi, si perviene a risultati e conclusioni non corretti. Questo perché l'incertezza non è un fenomeno locale, ma è fortemente dipendente anche da una visione complessiva: si intuisce, quindi, come l'uso delle reti

Bayesiane (figura seguente) possa essere più adeguato per rispondere alla domanda precedente.

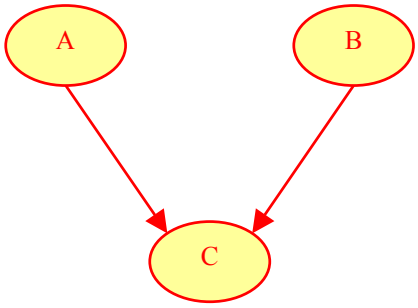


Figura 11 - if A then C & if B then C

1.6.7.3 I SISTEMI NORMATIVE EXPERT

I sistemi *normative expert* sono un'alternativa ai sistemi rule-based. Entrambi si occupano di decision making o argomenti simili, ma i principi alla base dei normative expert differiscono dai rule-based per i seguenti motivi:

- modellano il dominio (invece di modellare il processo di ragionamento condotto dall'esperto);
- usano la teoria classica della probabilità e la teoria della decisione (invece di effettuare calcoli da applicare alle regole in condizioni di incertezza);
- aiutano l'esperto nel decision making (invece di sostituire l'esperto).

Storicamente, l'uso della teoria classica della probabilità nei sistemi esperti fu tentato negli anni '60 da Gorry & Barnett ma la tecnologia non era pronta per elaborare notevoli quantità di calcoli: si pensi all'esempio visto in precedenza dove per solo 7 variabili bisogna elicitare 2^7 - 1 parametri! Negli anni '80 grazie ai progressi della tecnologia e agli studi sui modelli grafici di Judea Pearl, le reti Bayesiane furono, e lo sono tuttora, impiegate come "normative cognitive models" per il ragionamento in condizioni di incertezza. In particolare le BN permettono di espletare in modo naturale una forma di ragionamento, l' *explaining away* (to explain something away: giustificare, dare una spiegazione), difficile da implementare nei sistemi rule based e nelle reti neurali. Ad esempio, ricordando la rete sprinkler, se l'annaffiatore è on, allora molto "probabilmente" il pavimento sarà bagnato (*predizione*), invece, se qualcuno scivola sul pavimento si

ha un'evidenza (*adduzione*). D'altronde, se il pavimento fosse bagnato potremmo addurre o che lo sprinkler è on o che sta piovendo; l'evidenza "sprinkler on" riduce però la probabilità della variabile pioggia (*explaining away*).

Infine, nell'ambito dell'Intelligenza Artificiale, quando le reti Bayesiane sono considerate come potenziali modelli dell'attuale *human cognition*, bisognerebbe porsi due domande fondamentali:

1. Un'architettura somigliante a quella delle reti Bayesiane esiste in qualche parte del cervello umano? Allo stato attuale non esistono informazioni di modelli neurali plausibili per le BN, ma non è da escluderne l'esistenza.
2. Come potrebbero le BN gestire tipi di ragionamenti circa le classi di individui, relazioni, proprietà che costituiscono il pensiero umano? Una possibile risposta è il legame fra learning e background di conoscenza. Per esempio, la rete sprinkler è costruita per aiutare a capire se il pavimento è scivoloso e decidere come comportarsi. Il background di conoscenza potrebbe includere, in questo esempio, modelli di pavimenti, sprinkler, pioggia: tutte queste possibili informazioni andrebbero poi organizzate per costruire una specifica struttura di rete.

Sono stati fatti alcuni progressi nell'ambito di AI,BN & human cognition ma un vero modello cognitivo non è stato ancora proposto. (J. Y. Halpern "An analysis of first-order logics of probability" – D. Koller e A. Pfeffer "Probabilistic frame-based systems") [JEN96] [PEA00]

1.7 APPLICAZIONI DELLE RETI BAYESIANE

Di seguito sono elencate alcune delle applicazioni delle reti Bayesiane [LAM02]. Inoltre, presso i siti web di alcuni promulgatori della tecnologia Bayesiana quali Hugin Expert (www.hugin.com), Agena (www.agena.co.uk) e Norsys (produttore del tool Netica - www.norsys.com), sono consultabili anche dei *case studies*.

Image Understanding - Un sistema sviluppato al US Naval Research Laboratory impiega le reti Bayesiane per classificare un'imbarcazione mediante le rilevazioni effettuata da un sensore.
Forecastring (Previsione) - Il sistema ARCO1 è progettato per le previsioni sul mercato del petrolio. In [EZA96] le reti Bayesiane determinano le variazioni di customer account e transaction all'interno di una rete di telecomunicazioni attraverso il software, appositamente concepito, APRI (Advanced Pattern Recognition and Identification).
Intelligent Decision Making - Il sistema VISTA, sviluppato al NASA Mission Control Center, interpreta dati relativi alla telemetria e determina le operazioni utili per il sistema di propulsione di uno shuttle [JEN96].
Process monitoring - Il sistema esperto della General Eletric GEMS controlla l'apparecchiatura dei sistemi di potenza.
Diagnostics - In [PRZ00] si espone, suggerendo anche uno schema di analisi del problema, l'applicazione delle BN per la diagnosi del corretto funzionamento di diesel locomotives, satellite communications systems, satellite testing equipment. Nell'ambito medico il sistema PATHFINDER è progettato per la diagnosi di patologie da linfonodi di circa 60 malattie: è stato di recente integrato un sistema commerciale, INTELLIPATH, che è usato da numerose cliniche ed ospedali (negli Stati Uniti) [JEN96].
Altre applicazioni - Software maintenance [JEN96], comprensione del linguaggio naturale [BHA93].

Nel seguito, si presta un particolare attenzione all'utilizzo delle BN nei campi dell'Information Retrieval e del Data Mining.

1.7.1 IL PROCESSO DI RETRIEVAL

Ai nostri giorni, in cui Internet e il World Wide Web sono sempre più rilevanti nella nostra vita, lo scenario dell'Information Retrieval (IR - sviluppato all'inizio degli anni '40) muta velocemente. Nonostante questi cambiamenti e l'influenza delle nuove tecnologie, il fondamento dell'IR rimane sempre lo stesso: *organizzare e fornire le informazioni nel modo migliore a chi ne ha bisogno*.

Per *Information Retrieval* (IR) si intende l'insieme di tecniche che consentono un accesso mirato ed efficiente a raccolte quantitativamente rilevanti di oggetti contenenti principalmente testo (ad esempio, il recupero, in una biblioteca, di tutti i libri inerenti l'argomento "*Information Retrieval*"). Ogni oggetto è identificato (dopo una fase di *indexing*) da un insieme di "descrittori" che permettono di riferirlo. Attualmente la IR riguarda tecniche applicabili in modo algoritmico dai computer; quello che però rende un sistema un *Retrieval System* è la capacità di descrivere e tentare di soddisfare un ***fabbisogno informativo***, cioè un interesse specifico, dell'utente. Le descrizioni del fabbisogno informativo (*query*) sono espresse tipicamente in linguaggio naturale, ma sono implementate in altre forme, come l'uso di espressioni booleane (modello booleano) o di campioni di documenti che descrivano l'oggetto (clustering, modello probabilistico).

Il compito della IR è quello di estrarre da una raccolta di documenti la più grande quantità di informazioni su un dato argomento, quindi, il processo di *retrieval* è valutato in base a due fattori:

- *recall*: viene calcolato sugli oggetti recuperati rilevanti rispetto al fabbisogno informativo e ne misura la percentuale rispetto al totale contenuto nell'intera collezione;
- *precision* (precisione): viene calcolato sugli oggetti recuperati rilevanti rispetto al fabbisogno informativo e ne misura la percentuale rispetto al totale degli oggetti forniti al termine del retrieval step.

Un buon modello di *retrieval* sarà allora quello in grado di massimizzare *recall* e *precision***,** quindi di minimizzare rispettivamente il *silenzio* (assenza di informazione) e il *rumore* (deterioramento dell'informazione) . [JAC] [TUR]

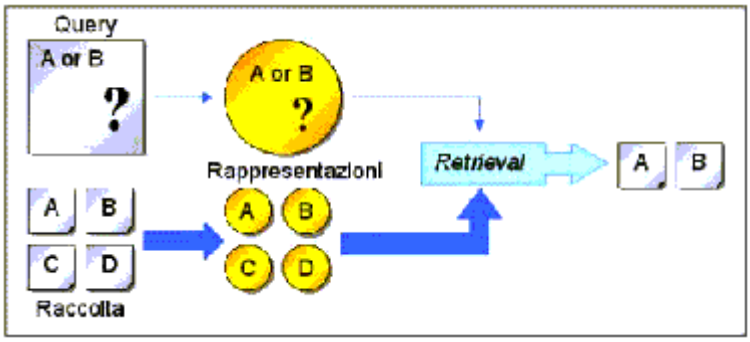


Figura 12 - Il processo di retrieval

1.7.1.1 INFORMATION RETRIEVAL E BN

Una rete Bayesiana può essere usata per recuperare documenti che riguardano una particolare informazione da un enorme insieme di dati memorizzati in formato elettronico (quindi anche nel Web). [HUE00]

Una tipica applicazione di un processo IR realizzata con le BN è riportata in [SHY00]: le informazioni sono spesso memorizzate in un database distribuito (Distributed Database – DDB) il che complica le operazioni per il recupero dei dati. Un IRS (Information Retrieval System) deve essere capace di aggiornare e recuperare le informazioni dal DDB in modo efficiente. Le reti Bayesiane rappresentano la struttura ideale per concepire un sistema di query intelligente che stabilisca quale sia l'ordine di visita (dei database) più indicato per il recupero dei dati. In [ANT02] è presentato, invece, il sistema ABN (Annotated Bayesian Netwotk) nel quale l'integrazione di etichette (informazioni testuali) e BN permette la concezione di un sistema di query intelligente per il retrieval di informazioni da database medici.

Il *web contestuale* e lo *user profiling*, infine, sono dei validi esempi di applicazione delle BN nel campo dell'IR. In particolare, per il web contestuale bisogna usare reti Bayesiane multiple. [BUT02]

Per illustrare l'utilità delle BN riguardo allo user profiling, possiamo menzionare, un semplice esempio. Immaginiamo che un utente riceva numerose e-mail quotidianamente e che voglia leggere soltanto quelle che ritiene più rilevanti: è possibile modellare il dominio delle e-mail ricevute in base alle preferenze dell'utente (un mittente preferito, un argomento relativo al suo lavoro, ecc.).

Per semplicità assumiamo che ogni messaggio sia rappresentato con un insieme di termini $\{A_1,A_2,\ldots,A_3\}$. Si supponga che ci sia un software di *auto-indicizzazzione* che assegni i valori $\{A_1 = a_1,\ A_2 = a_2,\ldots,\ A_n = a_n\}$ ad ogni nuovo messaggio arrivato, in pratica si estraggono delle informazioni dal messaggio in base al contenuto, al mittente, ecc. Dato un campione di e-mail che l'utente ha segnato come rilevanti e non rilevanti, possiamo utilizzare un algoritmo di apprendimento per rappresentare la rete Bayesiana. Quando arriva un nuovo messaggio lo si classifica secondo la seguente probabilità

$$P(\text{Relevance} = \text{relevant} \mid A_1 = a_1,\ldots, A_n = a_n)$$

Siano $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, $\mathbf{e}_4$, $\mathbf{e}_5$, $\mathbf{e}_6$ le rappresentazioni vettoriali di sei nuovi messaggi con probabilità condizionate

$$p(\text{Relevance} = \text{relevant} \mid \mathbf{e}_1) = 0.5$$
$$p(\text{Relevance} = \text{relevant} \mid \mathbf{e}_2) = 0.1$$
$$p(\text{Relevance} = \text{relevant} \mid \mathbf{e}_3) = 0.7$$
$$p(\text{Relevance} = \text{relevant} \mid \mathbf{e}_4) = 0.0$$
$$p(\text{Relevance} = \text{relevant} \mid \mathbf{e}_5) = 0.9$$
$$p(\text{Relevance} = \text{relevant} \mid \mathbf{e}_6) = 0.3$$

L'ordine di rilevanza quindi sarà: $\mathbf{e}_5$, $\mathbf{e}_3$, $\mathbf{e}_1$, $\mathbf{e}_6$, $\mathbf{e}_2$, $\mathbf{e}_4$. Seguendo questa classificazione, assumiamo che l'utente legga $\mathbf{e}_5$, $\mathbf{e}_3$, $\mathbf{e}_1$ e li classifichi poi come rilevante, non rilevante e rilevante. In base a questa preferenze espresse dall'utente, si provvede poi a raffinare la rete (*update*).

L'esempio precedente di user profiling si presta anche ad applicazioni di IR nell'ambito di piattaforme per l'e-learning sia per la gestione del profilo utente che per la ricerca degli argomenti più adatti alle sue esigenze (difficoltà, grado di interazione, ecc.).

Inoltre, nell'ambito dell'e-learning, significativo è il sito www.b-course.cs.helsinki.fi che illustra le finalità del sito B-course: un servizio web-based che permette all'utente di apprendere (quindi l'aspetto educativo) i fondamenti delle reti Bayesiane ed utilizzare il software proposto per l'analisi dei dati. [MIL01]

1.7.2 DATA MINING

Col nome ***data mining***[14] (**DM**) si intende l'applicazione di una o più tecniche che consentano l'esplorazione di grandi quantità di dati, archiviati in formato elettronico (database, data warehouse o altri tipi di repository), con l'obiettivo di estrapolare le informazioni più significative e di renderle disponibili e direttamente utilizzabili nell'ambito del decision making (processi decisionali). L'estrazione di conoscenza (informazioni significative) avviene tramite individuazione delle associazioni, o "patterns", o sequenze ripetute, o regolarità, o anomalie nascoste nei dati (in questo contesto un "pattern" indica una struttura, un modello, o, in generale, una rappresentazione sintetica).

Il termine *data mining* è utilizzato spesso come sinonimo di ***Knowledge Discovery in Databases*** **(KDD)**, anche se è più preciso identificare con data mining una particolare fase del *knowledge discovery*, ovvero del processo di estrazione della conoscenza, U.Fayyad, G.Piatetsky-Shapiro, P.Smyth, R.Uthurusamy (in "Advances in knowledge discovery and data mining" [FAY96]) definiscono il DM nel seguente modo: "*non trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"*. Questa definizione consente di mettere in luce l'aspetto inferenziale del processo, nonché le caratteristiche dei "patterns" in termini di validità, novità (non già noti), potenziale utilità e comprensibilità. Il *data mining* risponde a domande più generiche rispetto agli strumenti di *data retrieval* che consentono, invece, di avere risposte precise a qualsiasi domanda specifica. L'approccio del DM consente di far emergere dai dati le associazioni esistenti ***senza richiedere la formulazione di ipotesi a priori***. Sarà l'algoritmo a mettere in evidenza le altre caratteristiche (relazioni) che si presentano ripetutamente nei dati. Si tratta quindi di un approccio esplorativo e non, come nel *data retrieval*, verificativo. [HAN99]

1.7.2.1 DATA MINING E RETI DI BAYES

Il processo di data mining si differenzia da quello dell'IR per la maggiore flessibilità nel recupero di informazioni, per il maggior numero di informazioni

[14] *to mine - scavare, estrarre* - il nome sottolinea l'analogia fra la ricerca di informazione nei dati e la ricerca di un filone d'oro in una miniera.

che riesce a rilevare (si pensi alle informazioni di tipo nascosto), nello scoprire relazioni e strutture valide, nuove e utili nei dati. Le reti Bayesiane anche in questo campo sono apprezzate per la loro versatilità e rappresentano la nuova frontiera del DM: difatti sono molto recenti gli articoli, le testimonianze o esperienze relativi alla combinazione di data mining e reti Bayesiane.

Un esempio è il processo di data mining all'interno dei *log file* per facilitare il *web mining* nel World Wide Web: in tale proposito in [KAR01] si illustra come costruire, da un approccio distribuito, una BN che identifichi le relazioni fra i "web-log" (siti maggiormente ricercati) in base alle informazioni provenienti dai diversi nodi di una sottorete in Internet.

Il campo medico è una delle aree all'avanguardia nell'utilizzo delle reti bayesiane per il DM: in [WON99] è presentato l'utilizzo su database ospedalieri.

Una delle tecniche di data mining è la **classificazione**: i classificatori sono anche gli esempi più semplici di reti Bayesiane. Queste reti consistono solo di un nodo padre e più nodi figli. La classificazione definisce a quale classe (nodo figlio) un oggetto appartiene rispetto ad una classe padre(H in figura).

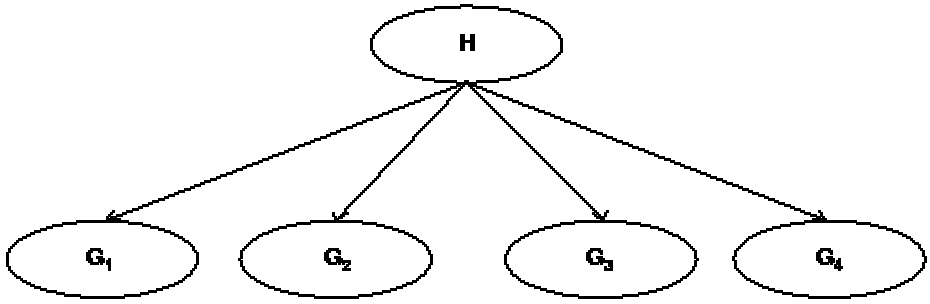


Figura 13 - Bayesian Classifier

Un esempio, in tale proposito, è la Naive Bayesian Network dalla quale prende il nome il metodo Naive Bayesian Classification usato per la classificazione di un insieme di dati; un software esistente che usa la classificazione è AutoClass.

In [BEN01] è riportato un esempio di utilizzo delle BN per la classificazione realizzata dal BNC (BN Classifier) che è un prototipo per il logical labeling (costruire la struttura logica di un documento partendo dal suo layout).

2 L'APPRENDIMENTO DI RETI BAYESIANE

L'ingegneria manuale delle Belief Net (o più in generale dei modelli probabilistici), in particolare nel configurare domini di notevoli dimensioni, è complessa, time consuming e fonte di imprecisioni. Negli ultimi anni, quindi, è maturato l'interesse della comunità scientifica nell'apprendere le reti bayesiane dai dati, ovvero da osservazioni campionarie attribuibili ad una distribuzione di probabilità implicita nel modello.

L'intervento dell'esperto, che talvolta risulta costoso e non accurato (non sempre la rete costruita dall'esperto risulta il modello più appropriato), presenta il beneficio della *life experience*. L'approccio del *learning from data*, tipico del machine learning[15], invece ha lo svantaggio di essere molto legato ai dati in quanto è proprio grazie ai database di osservazioni che la "macchina crea un background di esperienza" (il che implica la non disponibilità di un modello fintantoché non siano reperibili dei dati, mentre l'esperto è in grado di operare anche in assenza di tali informazioni). L'optimum sarebbe fornire all'esperto dei sussidi per il learning automatizzato che offrano l'opportunità di intervenire anche durante la fase di apprendimento.

La modalità di acquisizione più diffusa è il *batch learning* che consiste nel fornire un training set (database di campioni) a priori da impiegare per il learning. Spesso però un dominio descrive una realtà non statica ma in evoluzione (sia temporale che dal punto di vista delle caratteristiche del dominio); il learning automatizzato rende, inoltre, più agevole l'adeguamento del modello bayesiano a possibili cambiamenti del dominio. Con le nuove osservazioni è infatti possibile avviare un altro processo di apprendimento per così rappresentare le variazioni avvenute. In tale proposito, una modalità innovativa è l'*on line learning*, ovvero l'*adaptive learning*, che provvede ad aggiornare il modello in base ai nuovi campioni ed alle informazioni della rete pre-esistente.

Il learning, in particolare delle reti bayesiane, è riassumibile in:

[15] Insieme delle tecniche e metodi con cui una macchina è in grado di prendere decisioni in base all'esperienza acquisita dai casi precedenti o da training set.

- **learning structure** (o **structural learning**): apprendere la struttura della rete ovvero le relazioni fra le variabili;
- **learning parameters**: letteralmente apprendimento dei parametri. Conviene soffermarsi su questo concetto che sarà ripreso più volte in seguito. Per parametro si intende una costante che caratterizza la funzione di probabilità o di densità di una variabile casuale; ad esempio λ nella distribuzione di Poisson $p(X = x) = p(x) = \frac{\lambda^x}{x!} e^{-\lambda}$ (x intero), mentre per la gaussiana il parametro è un vettore rappresentato da media e varianza $\{\mu,\sigma\}$. In generale, con le reti bayesiane, si modella un dominio $\mathbf{X} = \{X_1,\ldots,X_n\}$ in cui ogni X_i è una variabile casuale discreta. $\mathbf{\Theta}$ è quindi il vettore di parametri associato a $\mathbf{X}$ che consente di definire la *likelihood* $p(\mathbf{X}|\mathbf{\Theta})$. In particolare, in statistica, si definisce la *log-likelihood* $L(\mathbf{X}|\mathbf{\Theta})=log p(\mathbf{X}|\mathbf{\Theta})$ che, nell'ipotesi di X_i iid (indipendenti ed identicamente distribuite), fattorizza in $L(\mathbf{X} \mid \mathbf{\Theta}) = \sum_{i=1}^{n} \log p(X_i \mid \mathbf{\Theta})$ [CIC].

  In sintesi, nelle reti bayesiane, l'accezione di parametro indica semplicemente la probabilità di un evento (condizionato) ovvero θ = *parametro* = $p\,(X = x|\,\theta)$[16]. Per chiarire si consideri di lanciare 6 volte una moneta e sia X_i una variabile che assume valore 1 se all'i-esimo lancio si ha testa, 0 altrimenti. Assumiamo che la probabilità che esca testa sia *p*: a sua volta *p* è anche il parametro della nostra variabile. In formule, essendo $\mathbf{X} = \{ X_1,X_2,X_3,X_4,X_5,X_6 \}$ e, ad esempio, $\mathbf{x} = \{1,0,0,0,1,0\}$ segue

$$\mathbf{\Theta} = p$$

$$L(\mathbf{X} = \mathbf{x} \mid \mathbf{\Theta}) = \sum_{i=1}^{n} \log(p(X_i = x_i \mid p)) = 2\log p + 4\log(1-p)$$

[16] La notazione vuole evidenziare il seguente aspetto: in generale è difficile conoscere la vera probabilità θ di un evento, anzi nelle learning from data viene stimata secondo l'interpretazione frequentista, quindi il condizionamento.

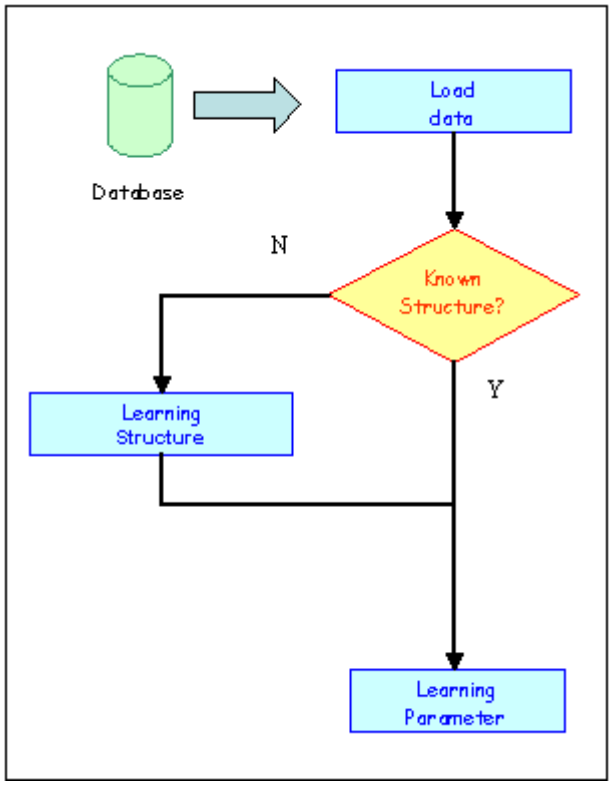


Figura 14 - Scenario dell'apprendimento

Lo scenario dell'apprendimento diventa variegato nelle combinazioni possibili a seconda che

a) il database di realizzazioni per il "learning from data":
   - sia *completo;*
   - presenti *missing value* (dati mancanti) o *hidden variable* (variabile non presente nel database ma che semplifica lo studio del dominio);

b) le informazioni disponibili sulla struttura del modello siano o meno disponibili:
   - *known structure***,** la struttura della rete è nota, per esempio, perché è fornita da un esperto;
   - *unknown structure*: bisogna apprendere la struttura della rete e dopo è possibile apprendere i parametri.

[HEC95] [BUN96] [STE00]

A seconda che i dati siano completi o incompleti e che la struttura sia nota, come riportato in Tabella 1, si utilizza l'algoritmo più appropriato. Le funzionalità di tali algoritmi saranno chiarite nel corso del capitolo e del successivo.

	Unkonown Structure	**Known Structure**
Complete database	Structural & Parameter Learning (K2, PC, MDL-based, TPDA)	Learning Parameter
Missing value or Hidden variable	PC, Structural EM	EM[17] Algorithm (Learning Parameter)

Tabella 1 - Algoritmi per il learning scenario

Il capitolo è impostato nel modo seguente: inizia con degli esempi con cui comprendere, nell'ottica di una metodologia bayesiana, lo scenario ora illustrato; dopodiché si procede con l'esposizione più rigorosa del Bayesian approach, alla base della maggior parte dei metodi di apprendimento. Quindi, imponendo alcune assunzioni semplificative, sono riportati i risultati utilizzati in numerosi algoritmi in letteratura.

[17] EM è un acronimo per Expectation Maximization. Gli altri acronimi saranno chiariti nel corso di questo capitolo e di quello dedicato allo Structural Learning.

2.1 L'APPROCCIO BAYESIANO

2.1.1 KNOWN STRUCTURE

Quando la struttura è nota, il problema è apprendere i parametri dal database. Un esempio - tratto da "A Tutorial on Learning with Bayesian Network" di D. Heckerman [HEC95] - renderà l'esposizione meno astratta.

Consideriamo una comune puntina da disegno - con una testa piatta e arrotondata. Se lanciamo in aria questa puntina, essa cadrà o sulla testa o sulla punta. Supponiamo di effettuare N+1 lanci, accertandoci che le proprietà fisiche della puntina e le condizioni sotto cui viene ripetuto l'esperimento permangano stabili nel tempo.

Date le prime N osservazioni, che costituiscono un database di eventi D, vogliamo determinare la probabilità dell'evento (*inferenza probabilistica*)[18] <<al lancio N+1-esimo (X_{N+1}) la puntina cade sulla testa>>.

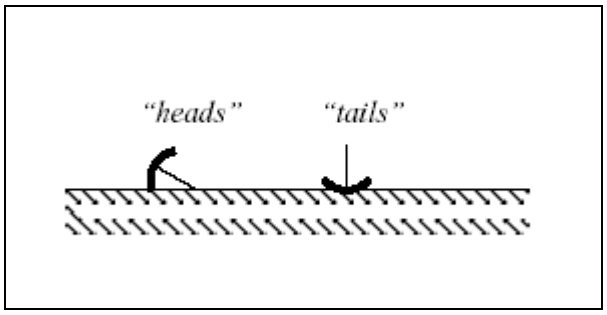


Figura 15 - I risultati dell'evento "lancio della puntina da disegno"

Per l'analisi bayesiana del problema conviene introdurre alcune notazioni. Denotiamo una variabile aleatoria con la lettera maiuscola (X,Y) e lo stato, o valore, della corrispondente variabile con la stessa lettera in minuscolo (x,y).

[18] Viene illustrato il processo di apprendimento di un parametro, ovvero di una probabilità: in effetti tale procedimento potrebbe rientrare anche fra i metodi di inferenza probabilistica, ma in tale ambito si preferisce parlare di learning parameter proprio per evidenziare l'assenza delle probabilità nel modello.

Denotiamo un insieme di variabili con (**X,Y**) mentre in minuscolo indichiamo un insieme di valori assunti dalle variabili: diremo che l'insieme della variabile **X** = $\{X_1,\ldots,X_n\}$è nello stato **x** = $\{x_1,\ldots,x_n\}$. Useremo $p(X=x|\xi)$ o $p(x|\xi)$ per denotare la probabilità che X = x condizionata (approccio bayesiano) dal livello di fiducia o background di conoscenza ξ[19] di cui si dispone. Useremo $p(x|\xi)$ per indicare anche la distribuzione di probabilità congiunta per X.

Nell'analisi classica di questo problema, si afferma che c'è una probabilità fisica che esca testa, la quale è sconosciuta: si **stima** questa probabilità dalle N ripetizioni dell'evento (interpretazione frequentista) e si usa questa stima per valutare, con il teorema di Bayes, $p(X_{N+1}$ = "testa"). Nell'approccio bayesiano, la probabilità che esca testa è associata anche alla conoscenza che si ha sull'evento.

Ritornando al problema, definiamo la variabile Θ il cui possibile valore , θ - *parametro*, rappresenta la vera probabilità (sconosciuta) dell'evento. L'incertezza su θ, data la conoscenza a priori o evidenza ξ (ad esempio un difetto della puntina che potrebbe condizionare il risultato dell'evento), è espressa da $p(\theta|\xi)$. In aggiunta, useremo X_l per indicare la variabile associata al lancio *l*-esimo, con *l* = 1,…,N+1, e D = $\{X_1 = x_1,\ldots,X_N = x_N\}$ per indicare l'insieme delle N osservazioni. In termini bayesiani, quindi, il problema della puntina da disegno si concretizza nel valutare:

$$p(X_{N+1} = x_{N+1}|D,\xi)$$

In tale proposito, la regola di Bayes fornisce la distribuzione di probabilità per Θ dato D e il background di conoscenza ξ:

$$p(X_i = heads \mid \theta,\xi) = \theta \qquad (2.1)$$

$$p(\theta \mid D,\xi) = \frac{p(\theta \mid \xi)p(D \mid \theta,\xi)}{p(D \mid \xi)} \qquad (2.2)$$

[19] L'aggiunta dello stato di conoscenza ξ è fondamentale perché rafforza la nozione bayesiana, soggettiva, della probabilità.

Le distribuzioni di probabilità p(θ|ξ) e p(θ|D,ξ) sono rispettivamente definite come ***a priori*** e ***a posteriori*** per Θ; il termine p(D|θ,ξ) è identificato come ***likelihood***. Il numeratore è invece esprimibile, per la legge del prodotto, nel modo seguente

$$p(D \mid \xi) = \int p(D \mid \theta, \xi) p(\theta \mid \xi) d\theta \qquad (2.3)$$

Sia gli statistici bayesiani che quelli classici concordano sull'espressione del likelihood p(D|θ,ξ) che è, relativamente a questo esempio, la funzione di probabilità per il campionamento binomiale. In particolare, dato il valore di Θ, le osservazioni in D sono mutamente indipendenti, e la probabilità di testa (punta) su qualsiasi osservazione è θ(1-θ). Di conseguenza l'equazione (2) diventa

$$p(\theta \mid D, \xi) = \frac{p(\theta \mid \xi)\theta^h (1-\theta)^t}{p(D \mid \xi)} \qquad (2.4)$$

dove *h* e *t* sono il numero di volte in cui, rispettivamente, il risultato dell'esperimento è testa o coda. Le quantità *h* e *t* vengono dette statistiche sufficienti[20] per il campionamento binomiale, poiché forniscono una rappresentazione dei dati che è sufficiente per valutare la distribuzione a posteriori da quella a priori. Infine, bisogna computare la media su tutti i possibili valori di Θ per determinare la probabilità che il lancio N+1-esimo fornisca testa:

$$p(X_{N+1} = testa \mid D, \xi) = \int p(X_{N+1} = testa \mid \theta, \xi) p(\theta \mid D, \xi) d\theta = \int \theta p(\theta \mid D, \xi) d\theta \equiv E_{p(\theta \mid D, \xi)}(\theta) \qquad (2.5)$$

ovvero *il valore atteso di θ rispetto alla distribuzione p(θ|D,ξ)* .

[20] Definiamo statistica **T**(**X**) una qualsiasi funzione reale o vettoriale su **X**. Se T($\mathbf{X}_1$)=T($\mathbf{X}_2$) per due campioni $\mathbf{X}_1$ e $\mathbf{X}_2$, con $\mathbf{X}_1 \neq \mathbf{X}_2$, allora **T** riassume i dati rappresentandoli con lo stesso valore. **T** è detta statistica *sufficiente* se ci sono delle funzioni g(**T**(**X**),**Θ**) e h(**X**) tali che p(**X**| **Θ**)=g(**T**(**X**), **Θ**)h(**X**) Dove tipicamente g(**T**(**X**),**Θ**) = p(**T**(**X**)|**Θ**) e h(**X**)=p(**X**|**T**(**X**)): il punto essenziale è che nel massimizzare ad esempio p(**X**| **Θ**) rispetto a **Θ** basta massimizzare g(**T**(**X**),**Θ**) poiché la statistica sufficiente riassume i dati. Nell'esempio della moneta, se n è la dimensione del campione (numero di lanci) e N_h il numero di volte in cui esce testa allora **T** =(Nh,n) è una statistica sufficiente poiché $p(\mathbf{X} \mid \mathbf{\Theta}) = p^{N_h}(1-p)^{n-N_h}$.

Per completare questo esempio, illustriamo un metodo per stimare la distribuzione a priori p(θ|ξ). Difatti, una volta valutata la distribuzione di probabilità a priori per θ, p(θ|ξ), risultano noti:

- la (2.3) in quanto p(D|θ,ξ) ha distribuzione binomiale;
- la (2.4) essendo il denominatore espresso dalla (2.3); e quindi, dalla relazione (2.5) è possibile valutare la distribuzione a posteriori per qualsiasi database D.

L'approccio più comune, di solito adottato per semplicità, è assumere la distribuzione p(θ|ξ) come Beta distribuzione, ne segue che

$$p(\theta \mid \xi) = Beta(\theta \mid \alpha_h, \alpha_t) \equiv \frac{\Gamma(\alpha)\theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1}}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \qquad (2.6)$$

dove α_h e α_t >0 (chiamati iperparametri per distinguerli dal parametro θ) sono i parametri della distribuzione Beta, con $\alpha=\alpha_h + \alpha_t$, e Γ(x) è la funzione Gamma di Eulero[21].

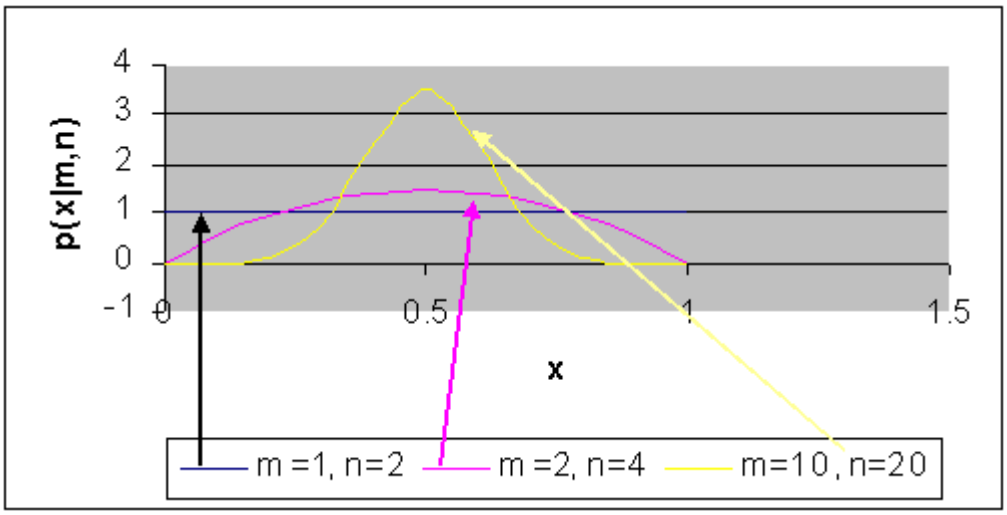


Figura 16- Beta(m,n) = [Γ(m+n)/(Γ(m) Γ(n))]$\theta^{m-1}(1-\theta)^{n-1}$ con θ in [0,1] m,n>0.
m,n sono definiti iperparametri per distinguerli dal parametro θ.
E(θ) =m/(m+n), Varianza = (mn)/((m+n)^2 (m+n+1)).
Se m=n=1 si ha una distribuzione a priori uniforme in [0,1].
Se m<n i valori di θ maggiori sono più probabili.

[21] Γ(x+1)=xΓ(x) (se x intero) e Γ(0)= Γ(1) = 1.

Dall'equazione (2.4), viene ricavata la distribuzione a priori che, a sua volta, è una distribuzione Beta

$$p(\theta \mid D,\xi) = \frac{\Gamma(\alpha+N)\theta^{\alpha_h+h-1}(1-\theta)^{\alpha+t_t-1}}{\Gamma(\alpha_h+h)\Gamma(\alpha_t+t)} = Beta(\theta \mid \alpha_h+h,\alpha_t+t) \qquad (2.7)$$

La media e la varianza della distribuzione Beta hanno, rispettivamente, le espressioni seguenti

$$\int \theta\, Beta\ (\theta \mid \alpha_h, \alpha_t)\, d\theta = \frac{\alpha_h}{\alpha}$$

$$\sigma^2 = \int (\theta - \mu)^2\ Beta\ (\theta \mid \alpha_h, \alpha_t) \cdot d\theta = \frac{\alpha_h \cdot \alpha_t}{\alpha^2 \cdot (1+\alpha)}$$

Per tale ragione α in letteratura è noto come *equivalent sample size*: infatti è assimilabile alla dimensione di un campione equivalente usato per stimare la media. Quindi, data una Beta a priori, dalla (2.5) si ricava la probabilità che esca testa al N+1-esimo tentativo:

$$p(X_{N+1} = heads \mid D,\xi) = \frac{\alpha_h + h}{\alpha + N} \qquad (2.8)$$

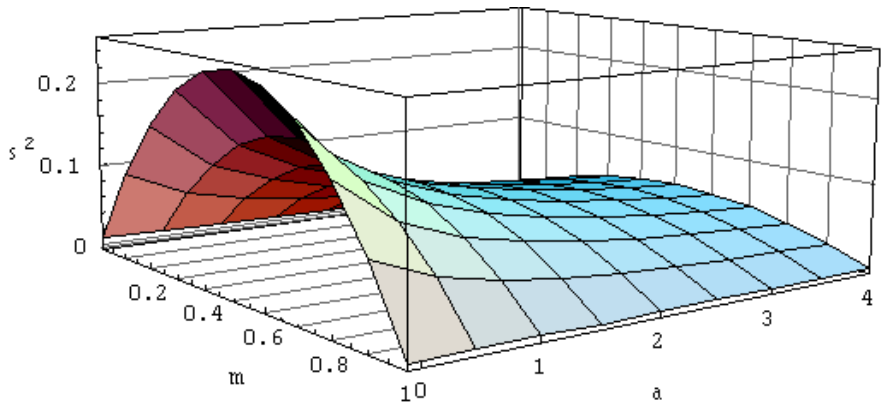


Figura 17 - Andamento della varianza s^2 *della distribuzione Beta in funzione della media* m *e della dimensione del campione* a

La distribuzione a priori $p(\theta|\xi)$ può essere *stimata* in molti modi, fra i quali:

- *metodo imagined future data*: poiché possiamo elicitare la probabilità di avere testa al primo tentativo (o in generale, dopo k lanci, la probabilità per il k+1 tentativo) possiamo usare queste stime per computare gli iperparametri. Ad esempio, Dall'equazione (2.8) si ha

$$p(X_1 = heads \mid \xi) = \frac{\alpha_h}{\alpha_h + \alpha_t}$$

$$p(X_2 = heads \mid X_1, \xi) = \frac{\alpha_h + 1}{\alpha_h + \alpha_t + 1}$$

  Date queste probabilità, possiamo risolvere le equazioni precedenti per valutare α_h e α_t.
- *metodo dei campioni equivalenti*: un altro metodo è basato sull'equazione (2.6). Questa equazione ci dice che, se partiamo con Beta(0,0) a priori, che codifica uno stato di informazione minima, e osserviamo α_h volte "testa" e α_t volte "punta" (conoscenza sul numero di lanci effettuati), allora la nostra distribuzione a posteriori sarà Beta(α_h ,α_t).

Sebbene la Beta a priori sia conveniente, non è accurata per alcuni problemi. Per esempio, supponiamo che la puntina sia stata acquistata in un negozio di articoli per la magia (in pratica può essere truccata). In questo caso, una conoscenza a priori più dettagliata può essere espressa da una combinazioni di distribuzioni Beta - per esempio

$$p(\theta|\xi) = 0.4\text{Beta}(20,1) + 0.4\text{Beta}(1,20) + 0.2\text{Beta}(2,2) \qquad (2.9)$$

dove 0.4 è la probabilità che la puntina sia più pensante sulla testa (coda).
In effetti, con questo ragionamento abbiamo introdotto una ***variabile nascosta***[22] o inosservata H, i cui stati corrispondono alle tre possibilità: (a) la puntina è polarizzata verso testa, (b) la puntina è polarizzata verso coda, (c) la puntina è

[22] Nei casi dove la Beta a priori è inadeguata è opportuno introdurre variabili, addizionali, nascoste.

normale; e abbiamo considerato che la variabile θ, condizionata da ogni stato, sia rappresentata da una distribuzione Beta.

Finora, abbiamo considerato soltanto le osservazioni ricavate da una distribuzione binomiale. In generale, le osservazioni possono essere estratte da una qualsiasi distribuzione:

$$P(X|\boldsymbol{\theta},\xi)=f(x,\boldsymbol{\theta})$$

dove $f(x,\boldsymbol{\theta})$ è la funzione di probabilità con parametri[23], in quantità finita, $\boldsymbol{\theta} = \{\theta_1 ,\ldots, \theta_n)$. Ad esempio, X può essere una variabile continua con andamento di tipo gaussiano

$$p(x|\theta,\xi)=(2\pi v)^{-1/2}e^{-(x-\mu)2/2v}$$

dove $\boldsymbol{\theta}$ = *parametri* $=\{\mu,v\}$. Senza considerare la forma della funzione, possiamo acquisire informazioni sui parametri avendo a disposizione i database di osservazioni e usando un approccio di tipo bayesiano: difatti, come visto nel caso binomiale, definiamo delle variabili aleatorie corrispondenti ai parametri sconosciuti e usiamo la regola di Bayes per associare le informazioni ricavate da D:

$$p(\boldsymbol{\theta}|D,\xi) = p(D,\boldsymbol{\theta}|\xi)\ p(\boldsymbol{\theta}|\xi)/\ p(D|\xi)$$

Per una classe di distribuzione conosciute come *famiglia esponenziale* (binomiale, multinomiale, normale, Gamma, Poisson,), i calcoli ora esposti possono essere eseguiti in forma chiusa.

[23] In questo caso è opportuno considerare il concetto di parametro nella sua forma più generale, ovvero una costante che caratterizza la funzione di probabilità o di densità di una variabile casuale, ad esempio λ nella distribuzione di Poisson.

2.1.2 UNKNOWN STRUCTURE

Quando la struttura è ignota, si ha la necessità di identificare il modello che meglio rappresenti il dominio (*model selection)*.
L'idea ottimale è di selezionare, con una metrica[24] opportuna, da un insieme di possibili modelli, la struttura che meglio si adatti ai campioni presenti in un database D.
Procediamo, come fatto prima, con un esempio per analizzare le difficoltà di questo scenario. Il database D dato in tabella è relativo a tre variabili aleatorie binarie A,B,C che assumono stati true (T) e false (F). In questo esempio, per semplicità[25], tutti i valori sono noti.

Campioni	**A**	**B**	**C**
1	T	F	T
2	T	T	T
3	F	T	T
4	F	T	T

Tabella 2 - Database di campioni

Generiamo prima uno spazio di ipotesi su tutte le possibili strutture della rete, alcune delle quali sono mostrate nella figura seguente

[24] **Metrica, funzione di costo, funzione obiettivo, scoring function**: sono tutti sinonimi che saranno usati nell'ambito di questo lavoro di tesi per identificare una funzione il cui valore rappresenti il grado di adattamento, di bontà, di un modello rispetto ai dati.
[25] Una variabile potrebbe assumere anche lo stato "?" per indicare che ha un valore sconosciuto o non rilevato (*missing value*).La presenza di missing value complica l'apprendimento perché bisogna effettuare un'analisi statistica sui dati che permetta di sostituire i valori mancanti.

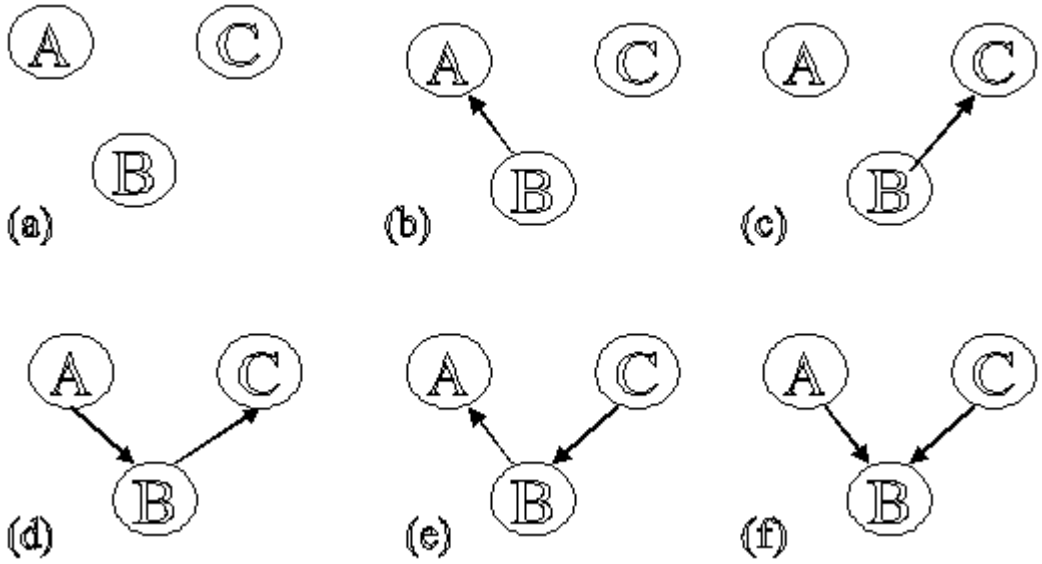


Figura 18 - Alcuni possibili DAG per tre variabili

La struttura (a), S_a, rappresenta le tre variabili fra loro indipendenti. Per questa struttura è necessario avere p(A), p(B), p(C). Invece per la struttura (c), S_c, sono necessarie le tabelle di probabilità p(A), p(B), $\theta_c = p(C|B)$.

Una rete di k variabili *binarie* necessita fra i k (completa indipendenza) e 2^k-1 (tutti i nodi connessi fra loro) valori reali, per ogni nodo, per specificare le tabelle di probabilità condizionate. Nell'esempio in questione si ha bisogno al massimo di 2^3-1=7 parametri.

Per quanto concerne il numero di differenti strutture G(n) su n nodi, nel contributo di R.D. Robinson "Counting unlabeled acyclic digraphs" (in Proceedings of the fifth Austrialian Conference on Combinatorial Mathematics, pages 28-43, 1976) è riportata la formula ricorsiva

$$G(0) = 1$$

$$G(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} G(n-i) \qquad (2.10)$$

dalla quale per n = 3 G(n) = 25, n = 4 G(n) = 543 e per n = 10 si ottengono $4.2 \cdot 10^{18}$ differenti strutture! Però molte di queste sono fra loro equivalenti in quanto rappresentano le stesse condizioni di indipendenza. In tale proposito, si

considerino le tre reti (d),(e),(f) per le quali la distribuzione di probabilità congiunta è:

(d) - p(A,B,C) = p(A) p(B|A) p(C|B)
(e) - p(A,B,C) = p(C) p(B|C) p(A|B)
(f) - p(A,B,C) = p(A) p(C) p(B|A,C)

Le reti (d) ed (e) hanno la stessa decomposizione e quindi le stesse proprietà di indipendenza mentre la (f) è differente: le strutture S_d e S_e rappresentano modelli di probabilità equivalenti[26].

Fissata una struttura S_m e i suoi parametri $\boldsymbol{\theta}_m$, per determinare il modello, fra quelli disponibili, che si adatta meglio ai campioni, definiamo come metrica la *verosimiglianza del campione (sample likelihood)*:

$$p(D \mid S_m, \boldsymbol{\theta}_m) = \prod_i p(caso_i \mid S_m, \boldsymbol{\theta}_m) \qquad (2.11)$$

Questa formulazione si basa sull'assunto che ogni caso sia indipendente dagli altri dato il "true model"[27] ("vero" modello) S_m e che i parametri in $\boldsymbol{\theta}_m$ siano indipendenti ed identicamente distribuiti.

Dalla (2.11), se la struttura della rete non è nota si considera prima un ipotesi sul modello dopodiché dai dati è possibile valutare $\boldsymbol{\theta}_m$ così da avere tutte le informazioni necessarie per elicitare $p(D|S_m, \boldsymbol{\theta}_m)$: in particolare, andrebbe scelta la coppia $(S_m, \boldsymbol{\theta}_m)$ che massimizza questa probabilità.

Per esempio, se la struttura S_d fosse candidata ad essere il true model, la probabilità del campione per la prima osservazione ($caso_1$) del database sarebbe:

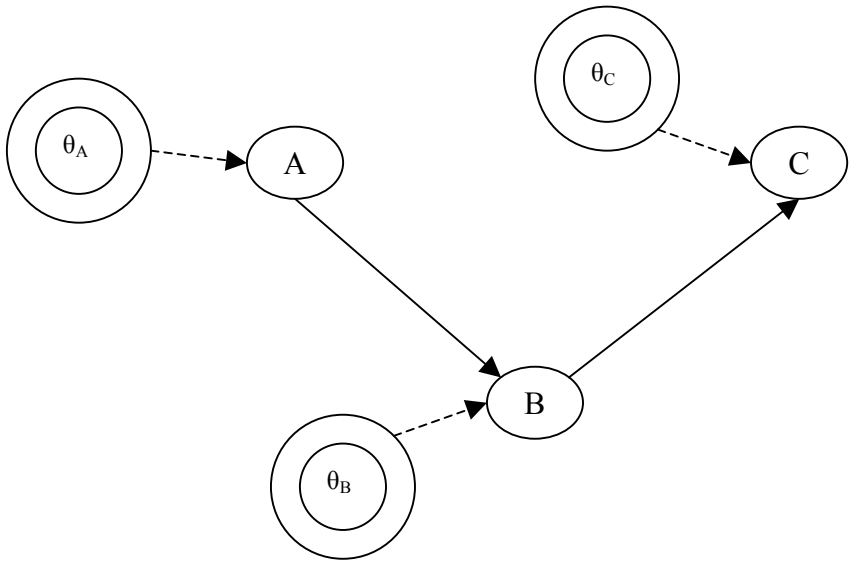
$$p(caso_1 \mid S_d, \boldsymbol{\theta}_d) = p(A = T \mid \boldsymbol{\theta}_d)\, p(B = F \mid A = T, \boldsymbol{\theta}_d)\, p(C = T \mid B = F, \boldsymbol{\theta}_d) \quad (2.12)$$

Analogamente si procede per i casi 2, 3 e 4 presenti nella tabella riportata sopra. Per ogni struttura quindi è possibile valutare, con la (2.11), $p(D| S_m , \boldsymbol{\theta}_m)$.Una

[26] Un esame dettagliato delle equivalenze e della sua utilità nel learning è presentato in "Learning Equivalence Classes of Bayesian Network structures" di D.M. Chickering in Journal of Machine Learning Research 2 (2002) 445-498.

[27] Il modello "vero" è quello fornito da un esperto o di cui è nota la struttura.

delle metodologie più semplici per scegliere, dato un modello S_m,il parametro $\boldsymbol{\theta}_m$ migliore consiste nel massimizzare la ***funzione di verosimiglianza***[28] $p(D| S_m , \boldsymbol{\theta}_m)$: criterio *Maximum Likelihood* (ML) rispetto al parametro (vettore di parametri) $\boldsymbol{\theta}_m$. In effetti, i parametri possono essere interpretati come i nodi di una rete "ampliata" come si evince dalla figura seguente per la struttura S_d.



Riconsideriamo proprio l'esempio dell'ipotesi su S_d. In pratica i tre termini al secondo membro della (2.12) sono ricavati dalle corrispondenti entrate nella tabella di probabilità. In particolare, i parametri in $\boldsymbol{\theta}_d$ possono essere scomposti per ogni nodo $\boldsymbol{\theta}_d=\{ \boldsymbol{\theta}_{d,A} ,\boldsymbol{\theta}_{d,B} ,\boldsymbol{\theta}_{d,C} \}$, così la sample likelihood diventa:

$$p(sample \mid S_d, \boldsymbol{\theta}_d) = \prod_i p(A_i \mid \boldsymbol{\theta}_{d,A}) p(B_i \mid A_i, \boldsymbol{\theta}_{d,B}) p(C_i, \boldsymbol{\theta}_{d,C}) \qquad (2.13)$$

[28] Si chiama funzione di verosimiglianza la densità congiunta (probabilità congiunta nel caso discreto) di $(X_1, X_2, \ldots, X_n)$ nel punto $(x_1, x_2, \ldots, x_n)$ considerata come funzione di un parametro θ. La funzione di verosimiglianza viene generalmente identificata con $L(\theta; x_1, x_2, \ldots, x_n)$ che evidenzia θ come parametro indipendente e la notazione L ricorda la terminologia inglese (Likelihhod). Nel nostro caso $L(\theta;\mathbf{x}) = P(\mathbf{X} = \mathbf{x} \mid \theta)$.

Poiché le variabili A,B,C sono binarie, la relazione precedente corrisponde a un prodotto di variabili binomiali. Ad esempio,

$$p(sample_A \mid \boldsymbol{\theta}_{d,A}) = \boldsymbol{\theta}_{d,A}^{p_A}(1-\boldsymbol{\theta}_{d,A})^{n_A} \qquad (2.14)$$

dove p_A e n_A forniscono l'occorrenza di (A = T) e (A = F) nella tabella dei campioni. Poiché nella struttura S_d il nodo A è padre ed assume solo gli stati True e False, fra loro mutuamente esclusivi, segue che è lecito scrivere $\boldsymbol{\theta}_{d,A} = \theta_{d,A} = p(A = True)$[29].

Il criterio Maximum Likelihood, nel caso binomiale, è soddisfatto proprio dalla frequenza osservata; quindi il valore che massimizza la (2.12) è:

$$\hat{\theta}_{d,A} = \frac{p_A}{n_A + p_A}$$

In modo analogo si procede per gli altri termini $\boldsymbol{\theta}_{d,B}$,$\boldsymbol{\theta}_{d,C}$ in modo da ricavare il parametro $\boldsymbol{\theta}_d = \{ \boldsymbol{\theta}_{d,A}, \boldsymbol{\theta}_{d,B}, \boldsymbol{\theta}_{d,C} \}$.

Un altro criterio, più generale del precedente, è il *Maximum A Posteriori (MAP)* in cui si massimizza la probabilità a posteriori. Ad esempio, per la variabile A e il parametro θ_A, dal teorema di Bayes la probabilità a posteriori è così espressa

$$p(\theta_A \mid D) = \frac{p(D \mid \theta_A)p(\theta_A)}{p(D)}$$

dove il numeratore contiene la probabilità del campione e quella a priori, mentre il denominatore è:

$$p(D) = \int p(D \mid \theta_A)p(\theta_A)d\theta_A$$

[29] P(A = False) si ricava per differenza: 1 - p(A = True).

Con alcune considerazioni analoghe a quelle viste per il criterio ML si procede a massimizzare la probabilità a posteriori.

La situazione esaminata è semplice avendo solo tre stati e nessun dato mancante. Ciononostante, si comprende come le operazioni siano numerose: quindi, come è indicato anche in letteratura, il problema dell'apprendimento Bayesiano presenta *complessità NP* (Non Polinomiale).

Si dispone di una formulazione matematica agevole del learning quando si esaminano particolari distribuzioni statistiche appartenenti alla famiglia delle funzioni esponenziali; uno dei pregi, ad esempio, è l'opportunità di ricavare relazioni esprimibili in forma chiusa: questa è soltanto una delle possibili assunzioni; altre sono illustrate nel paragrafo successivo. [BUN96][KRA98]

2.2 LE ASSUNZIONI

Per comprendere meglio le asserzioni riportate in questo paragrafo, di seguito sono presentate alcune notazioni che, sebbene appesantiscano il formalismo matematico, rendono allo sesso tempo più compatti gli enunciati ed i risultati ottenuti.

Assumiamo che il dominio sia costituito dalle variabili $\mathbf{X} = \{X_1,\ldots, X_n\}$ e che si disponga di un database di campioni $D = \{\mathbf{x}_1,\ldots, \mathbf{x}_N\}$, ove $\mathbf{x}_i$ ("caso" o osservazione o realizzazione) rappresenta l'i-esimo campione di tutte le osservazioni[30] delle variabili in **X**. D quindi rappresenta un campione casuale attribuibile ad una qualche distribuzione di probabilità per **X** che possa essere codificata, qualitativamente, da un modello grafico causale[31] con struttura **m**.

L'incertezza sulla struttura ipotizzata e sui i parametri del modello è espressa in termini probabilistici. In particolare si definisce una variabile discreta **M** i cui stati **m** corrispondono ai possibili modelli attribuibili al dominio **X**, mentre l'incertezza su **M** è espressa dalla distribuzione di probabilità p(**m**). Inoltre, per ogni struttura **m**, si definisce un vettore $\mathbf{\Theta}_m$, i cui valori $\boldsymbol{\theta}_m$ corrispondono ai possibili veri parametri. L'incertezza circa $\mathbf{\Theta}_m$ è rappresentata dalla densità di probabilità $p(\boldsymbol{\theta}_m|\mathbf{m})$.

Dato un campione casuale D, possiamo elaborare la distribuzione a posteriori, per ogni **m** e $\boldsymbol{\theta}_m$,usando la regola di Bayes

$$p(\mathbf{m} \mid D) = \frac{p(\mathbf{m})p(D \mid \mathbf{m})}{\sum_{m'} p(\mathbf{m}')p(D \mid \mathbf{m}')} \qquad (2.15)$$

Struttura

[30] Ovvero $\mathbf{x}_i = \{x_{i1},\ldots,x_{in}\}$. In effetti D è una matrice N x n.

[31] Causale: il rapporto causa/effetto è sinonimo di dipendenza condizionata (condizione di Markov).

$$p(\boldsymbol{\theta}_m \mid D, \mathbf{m}) = \frac{p(\boldsymbol{\theta}_m \mid \mathbf{m}) p(D \mid \boldsymbol{\theta}_m, \mathbf{m})}{p(D \mid \mathbf{m})} \qquad (2.16)$$

Parametri

dove

$$p(D \mid \mathbf{m}) = \int p(D \mid \boldsymbol{\theta}_m, \mathbf{m}) p(\boldsymbol{\theta}_m \mid \mathbf{m}) d\boldsymbol{\theta}_m \qquad (2.17)$$

è la ***marginal likelihood*** *o funzione di verosimiglianza marginale o evidenza*.
L'approccio bayesiano può essere così generalizzato: data un'ipotesi di interesse *h* relativa ad un modello **m**, determinare la probabilità che *h* sia vera, usufruendo delle informazioni nel database D. Effettuando una media - *model averaging* - su tutti i possibili modelli e parametri si ottiene:

$$p(h \mid D) = \sum_m p(\mathbf{m}|D) p(h|D, \mathbf{m}) \qquad (2.18)$$

Model averaging

$$p(h \mid D, \mathbf{m}) = \int p(h \mid \boldsymbol{\theta}_m, \mathbf{m}) p(\boldsymbol{\theta}_m|D, \mathbf{m}) d\boldsymbol{\theta}_m \qquad (2.19)$$

L'ipotesi *h* può rappresentare una possibile struttura del modello, una possibile distribuzione dei parametri o più semplicemente un evento. Ad esempio, se h = $\mathbf{X}_{N+1} = \mathbf{x}_{N+1}$ (si ricordi l'esempio della puntina da disegno), allora

$$p(\mathbf{x}_{N+1} \mid D, \mathbf{m}) = \sum_{\mathbf{m}} p(\mathbf{m} \mid D) \int p(\mathbf{x}_{N+1} \mid \boldsymbol{\theta}_m, \mathbf{m}) p(\boldsymbol{\theta}_m|D, \mathbf{m}) d\boldsymbol{\theta}_m \qquad (2.20)$$

dove $p(\mathbf{x}_{N+1}|\boldsymbol{\theta}_m, \mathbf{m})$ è la funzione di verosimiglianza per l'ipotesi $h = \boldsymbol{X}_{N+1} = \boldsymbol{x}_{N+1}$. (Un altro possibile esempio potrebbe essere l'ipotesi "X causa Y").

Le assunzioni[32] seguenti permetteranno di ottenere i risultati precedenti in forma chiusa.

[32] Una descrizione rigorosa e completa delle assunzioni, con dimostrazione, è presente in "Learning Bayesian Networks: the combination of knowledge and statistical data" di D. Heckerman, D. Geiger, D.M. Chickering.[HEC94]

ASSUNZIONE 1 - MULTINOMIAL SAMPLE: distribuzione multiniomiale dei campioni. Ogni variabile $X_i \in \mathbf{X}$ è discreta, avente r_i possibili valori $x_i^1,...,x_i^{r_i}$ (quindi D è dato da una matrice N,r_i) ed ogni funzione di distribuzione locale (per ogni nodo) è una collezione di distribuzioni multinomiali condizionate da $\mathbf{pa}_i^j \in \mathbf{Pa}_i$, ove $\mathbf{Pa}_i = \{\mathbf{pa}_i^1,\ldots, \mathbf{pa}_i^{q_i}\}$ ($q_i = \prod_{X_i \in Pa_i} r_i$) è l'insieme delle configurazioni dei padri di X_i. In simboli

$$p(x_i^k \mid \mathbf{pa}_i^j, \boldsymbol{\theta}_i, \mathbf{m}) = \theta_{ijk} > 0$$

dove, a sua volta, gli elementi del vettore $\boldsymbol{\theta}_i = ((\theta_{ijk})$, con $k = 2,\ldots, r_i$ e $j = 1,\ldots,q_i$, rappresentano i parametri (per $k = 1$ il parametro è ricavato per differenza). Per convenienza, definiamo il vettore (per tutti gli i e j)

$$\boldsymbol{\theta}_{ij} = \theta_{ij2},...,\theta_{ijr_i}$$

per cui $\boldsymbol{\theta}_m = (\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_n)$; ricordando l'esempio della puntina segue $\mathbf{p}(D \mid \boldsymbol{\theta}_m, \mathbf{m}) = \boldsymbol{\theta}_m$. Inoltre dalla relazione sulla distribuzione della probabilità congiunta per una BN (illustrata nel primo capitolo) si ha la seguente espressione per $p(\mathbf{X}|\boldsymbol{\theta}_m,\mathbf{m}) = p(\mathbf{x}|\boldsymbol{\theta}_m,\mathbf{m}) = p(X_1 = x_1, \ldots, X_n = x_n| \boldsymbol{\theta}_m,\mathbf{m})$:

$$p(\mathbf{x} \mid \boldsymbol{\theta}_m, \mathbf{m}) = \prod_{i=1}^{n} p(x_i \mid \mathbf{Pa}_i, \boldsymbol{\theta}_i, \mathbf{m})$$

$\boldsymbol{\theta}_i$ denota in sintesi l'insieme dei parametri associati alla funzione di verosimiglianza locale.

Assunzione 2 - Complete Data: D è completo, nel database non sono presenti dati mancanti (missing value).

Assunzione 3 - Parameter Indipendence: I parametri devono essere mutuamente indipendenti; pertanto questa assunzione, introdotta da Spiegelhater e Lauritzen, viene detta *indipendenza dei parametri*. Per esempio, date le funzioni di verosimiglianza multinomiali, l'indipendenza di $\boldsymbol{\theta}_{ij}$ implica

$$p(\boldsymbol{\theta}_m|\mathbf{m}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|\mathbf{m})$$

Assunzione 4 – Dirichlet: un approccio comune è scegliere una distribuzione a priori per i parametri *coniugata*. Una distribuzione a priori è coniugata quando, combinata con l'evidenza (ovvero la conoscenza a priori), la distribuzione a posteriori presenta la stessa forma funzionale della priori. La coniugazione restringe la scelta delle funzione ad una classe limitata di distribuzioni: la famiglia esponenziale, a cui appartiene la distribuzione di *Dirichlet*.

$$Dir(\boldsymbol{\theta} \mid \alpha_1,...,\alpha_r) \equiv \frac{\Gamma(\alpha)}{\prod_{k=1}^{r}\Gamma(\alpha_k)}\prod_{k=1}^{r}\boldsymbol{\theta}_k^{\alpha_k-1}$$

Distribuzione di Dirichlet per il paramertro θ con iperparametri $\alpha_1,\ldots,\alpha_r$ e $\alpha = \alpha_1 +\ldots+ \alpha_r$, $\alpha_k>0$ e $\theta_1 +\ldots+ \theta_r=1$

Un modo alternativo (invece della coniugata) per stimare la distribuzione a priori è basato sul concetto di **Maximum Entropy** (massima entropia): si richiede all'esperto di specificare alcune statistiche della distribuzione a priori quali media o varianza e si elabora, in base a D, la distribuzione a priori avente la massima entropia, ovvero caratteristiche simili, rispetto a quella fornita dall'esperto.

Illustriamo ora le conclusioni che derivano da tali asserzioni.

L'assunzione multinomiale permette di rappresentare, in simboli, la distribuzione di probabilità congiunta per il dominio $\mathbf{X} = \{X_1,\ldots, X_n\}$, rappresentato da una possibile struttura di rete $\mathbf{m}$ con parametri $\boldsymbol{\theta}_m$, come segue

$$p(\mathbf{X} \mid \boldsymbol{\theta}_m, \mathbf{m}) = \prod_{i=1}^{n} p(X_i \mid \mathbf{Pa}_i, \boldsymbol{\theta}_i, \mathbf{m})$$

Inoltre i parametri restano indipendenti anche dato il database D

$$p(\boldsymbol{\theta}_m|D,\mathbf{m}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|D,\mathbf{m})$$

il che consente di aggiornare ogni vettore dei parametri $\boldsymbol{\theta}_{ij}$ in modo indipendente.

In base all'assunzione di distribuzione a priori coniugata, assumendo che ogni vettore $\boldsymbol{\theta}_{ij}$ abbia una distribuzione a priori di Dirichlet ($\mathrm{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1},...,\alpha_{ijr_i})$), la distribuzione a posteriori, dato D, per i parametri è a sua volta di Dirichlet

$$p(\boldsymbol{\theta}_{ij}|D,\mathbf{m}) = \mathrm{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1}+N_{ij1},...,\alpha_{ijr_i}+N_{ijr_i})$$

dove N_{ijk} è il numero di occorrenze in D dei casi in cui $X_i = x_i^k$ e $\mathbf{Pa}_i = \mathbf{pa}_i^j$. In particolare le occorrenze N_{ijk} rappresentano delle statistiche sufficienti per il modello **m**. Infine[33], si ottiene la marginal likelihood (Cooper e Herskovits (1992) sono stati i primi a derivare questa equazione) con cui è possibile esprimere in forma chiusa l'espressione (2.17):

$$\begin{aligned} p(D|\mathbf{m}) &= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \\ \alpha_{ij} &= \sum_{k=1}^{r_i}\alpha_{ijk} \\ N_{ij} &= \sum_{k=1}^{r_i}N_{ijk} \end{aligned} \qquad (2.21)$$

Ricordando la metodologia illustrata nell'esempio della puntina da disegno, nota $p(D|\mathbf{m}) = \int p(D|\boldsymbol{\theta}_m,\mathbf{m})p(\boldsymbol{\theta}_m|\mathbf{m})d\boldsymbol{\theta}_m = \int \boldsymbol{\theta}_{ij} Dir(\boldsymbol{\theta}_{ij}|\alpha_{ij1}+N_{ij1},...,\alpha_{ijr_i}+N_{ijr})d\boldsymbol{\theta}_{ij}$ dalla (2.21), per il teorema di Bayes sono valutabili anche

[33] Una deduzione rigorosa e più precisa di queste relazioni è presente in "Learning Bayesian Networks: the combination of knowledge and statistical data" di D. Heckerman, D. Geiger, D.M. Chickering e in "A Tutorial on learning with Bayesian Networks" di D. Heckerman.

$$p(\mathbf{m} \mid D) = \frac{p(\mathbf{m})p(D \mid \mathbf{m})}{\sum_{m'} p(\mathbf{m}')p(D \mid \mathbf{m}')} \text{ (struttura)}, \quad p(\boldsymbol{\theta}_m \mid D, \mathbf{m}) = \frac{p(\boldsymbol{\theta}_m \mid \mathbf{m})p(D \mid \boldsymbol{\theta}_m, \mathbf{m})}{p(D \mid \mathbf{m})}$$

(parametri), nell'ipotesi che siano conosciute p(**m**) e p($\boldsymbol{\theta}_m$|**m**) (in merito si considerino i due paragrafi successivi).

Rendiamo concreto il discorso esposto finora: consideriamo come ipotesi *h* "l'evento che $\mathbf{X}_{N+1} = \mathbf{x}_{N+1}$". Valutiamo, quindi, p($\mathbf{x}_{N+1}$|D,**m**), dove $\mathbf{x}_{N+1}$ è il caso che si presenta dopo l'N-esimo in D. Da (2.17) e (2.20) si ha

$$p(\mathbf{x}_{N+1}|D,\mathbf{m}) = E_{p(\boldsymbol{\theta}_m|D,\,\mathbf{m})}\left(\prod_{i=1}^{n} \theta_{ijk}\right) = \int\left(\prod_{i=1}^{n} \theta_{ijk}\right) p(\boldsymbol{\theta}_m \mid D, \mathbf{m}) d\boldsymbol{\theta}_m =$$

data l'indipendenza dei parametri dato D

$$= \prod_{i=1}^{n} \int \theta_{ijk}\, p(\boldsymbol{\theta}_{ij}|D,\mathbf{m})\, d\boldsymbol{\theta}_{ij}$$

poiché ogni integrale rappresenta il valore atteso della distribuzione di Dirichlet, segue

$$= \prod_{i=1}^{n} \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}.$$

Infine per ottenere p($\mathbf{x}_{N+1}$|D) bisognerebbe effettuare una media di p($\mathbf{x}_{N+1}$|D,**m**) rispetto a tutti i possibili modelli **m**. Questo ultimo step rappresenta uno dei limiti dell'approccio bayesiano e, in particolare, del *model averaging*, sia in termini computazionali che di fattibilità poiché non è possibile, nella maggiore parte dei casi, valutare tutti i modelli per un dominio, specie se questa valutazione implichi, a monte, un processo di apprendimento della struttura.
Con lo stesso procedimento, prima, abbiamo contemplato sia l'ipotesi *h* = "quale è il parametro $\boldsymbol{\theta}_m$ associato a D e alla struttura **m**?" (apprendimento delle probabilità - p($\boldsymbol{\theta}_m$| D,**m**)), sia *h* = "quale è il modello **m** associato a D?" (apprendimento della struttura - p(**m**|D)). [HEC94][HEC95][KRA98]

2.2.1 LA DISTRIBUZIONE A PRIORI

Le relazioni di partenza per il learning, secondo l'approccio bayesiano sono:

$$p(\boldsymbol{m} \mid D) = \frac{p(\boldsymbol{m})p(D \mid \boldsymbol{m})}{\sum_{m'} p(\boldsymbol{m}')p(D \mid \boldsymbol{m}')}$$

$$p(\boldsymbol{\theta}_m \mid D, \boldsymbol{m}) = \frac{p(\boldsymbol{\theta}_m \mid \boldsymbol{m})p(D \mid \boldsymbol{\theta}_m, \boldsymbol{m})}{p(D \mid \boldsymbol{m})}$$

Per valutare la probabilità a posteriori relativa alla struttura di un modello o ai parametri, si deve prima stimare la priori p(**m**) o p($\boldsymbol{\theta}_m$|**m**). Invece, per quanto concerne l'espressione al denominatore p(D) e la sua decomposizione, se i modelli ammissibili **m**' sono numerosi, diventa intrattabile. Ciononostante, con alcune assunzioni, che sono da aggiungere alle precedenti, è possibile ricavare la distribuzione a priori per la struttura ed i parametri in modo agevole.

Di seguito si riportano le indicazioni presenti nelle opere di Heckerman; in proposito, premessa la validità delle assunzioni di distribuzione multinomiale e indipendenza dei parametri, esponiamo due nuovi asserti utili in seguito:

1. **Markov equivalence.** Due strutture di un modello **X** sono *Markov equivalenti* se rappresentano lo stesso insieme di asserzioni di indipendenza condizionata per **X**. Ad esempio, dato **X** = {X, Y, Z}, le strutture X→Y→Z, X←Y→Z e X←Y←Z rappresentano tutte l'indipendenza fra X e Z dato Y. Di conseguenza queste strutture sono Markov equivalenti. Un altro esempio di equivalenza di Markov è l'insieme di grafi completi su **X** (un modello completo ha tutti i nodi collegati fra loro, in pratica non ci sono indipendenze): se **X** contiene *n* variabili, ci sono *n*! possibili strutture complete, una per ogni possibile ordinamento, che sono Markov equivalenti rispetto alla distribuzione congiunta p(**X**). In generale, due modelli sono Markov equivalenti, se e

solo se hanno, ignorando la direzione degli archi, la stessa struttura ed anche le stesse v^{34}-structure.

2. **Equivalenza delle distribuzioni.** Il concetto di equivalenza delle distribuzioni è relazionato all'equivalenza di Markov. Assumiamo che tutti i modelli causali per **X** abbiano funzioni di verosimiglianza locali in un insieme (o famiglia) *F* (non rappresenta una restrizione perché tale insieme può essere molto ampio) sul quale stabilire una relazione di equivalenza. Due strutture $\mathbf{m}_1$ e $\mathbf{m}_2$ sono equivalenti per la distribuzione rispetto a *F* se esse rappresentano la stessa distribuzione di probabilità congiunta per **X**, cioè se per ogni $\boldsymbol{\theta}_{m1}$ esiste $\boldsymbol{\theta}_{m2}$ tale per cui $p(\mathbf{X}|\ \boldsymbol{\theta}_{m1}, \mathbf{m}_1) = p(\mathbf{X}|\ \boldsymbol{\theta}_{m2}, \mathbf{m}_2)$ e viceversa. L'equivalenza delle distribuzioni rispetto a *F* implica l'equivalenza di Markov ma non è vero il viceversa[35]. L'asserto esposto è importante perché se due modelli $\mathbf{m}_1$ e $\mathbf{m}_2$ sono *distribuzione - equivalenti rispetto a F* allora è ragionevole considerare che non possano essere distinti in base alle informazioni estratte dai dati, ovvero $p(D|\mathbf{m}_1) = p(D|\mathbf{m}_2)$ per qualsiasi D (proprietà di ***likelihood equivalence***).

2.2.1.1 PRIORI PER I PARAMETRI

In "A Bayesian Approach to Causal Discovery" di Heckerman, Meek, Cooper [MEE97], è indicato che le assunzioni di *indipendenza dei parametri* e *likelihood equivalence* implicano che, per un grafo completo $\mathbf{m}_c$, i parametri (ovvero likelihood locali) presentano una distribuzione di Dirichlet con il seguente vincolo sugli iperparametri:

$$\alpha_{ijk} = \alpha p(x_i^k, \boldsymbol{pa}_i^j \mid \boldsymbol{m}_c) \qquad (2.22)$$

Vincolo sugli iperparametri di Dirichlet in base alle assunzioni di indipendenza dei parametri e likelihood equivalence

[34] v – structure: consiste di archi che convergono nello stesso nodo come X→Y←Z.
[35] Alcune precisazioni a riguardo sono presenti nelle opere di Heckerman.

dove α è la dimensione equivalente del campione (definibile anche dall'esperto), e $p(x_i^k, \boldsymbol{pa}_i^j \mid \boldsymbol{m}_c)$ è ricavata dalla distribuzione di probabilità congiunta p(**X**|$\mathbf{m}_c$).

Ovviamente i modelli con cui si esplicitano le relazioni di indipendenza condizionata di un dominio non sono dei grafi completi. Per determinare la priori per i parametri in caso di modelli aventi strutture non completamente connesse incomplete si considera l'assunzione seguente: - **Modularità dei parametri** (*parameter modularity*) - *se X_i ha gli stessi padri in $\boldsymbol{m}_1$ e $\boldsymbol{m}_2$ allora $p(\boldsymbol{\theta}_{ij}|\boldsymbol{m}_1) = p(\boldsymbol{\theta}_{ij}|\boldsymbol{m}_2)$ per $j = 1,\ldots, qi$*[36]. Tale proprietà è detta di modularità dei parametri in quanto le distribuzioni per i parametri $\boldsymbol{\theta}_{ij}$ dipendono soltanto dalla struttura del modello che rappresenta X_i localmente, ovvero X_i ed i suoi padri.

Date le assunzioni di *parameter modularity* e *parameter indipendence*, è semplice costruire una priori per i parametri da un arbitrario modello, dato il priori sulla struttura completa. In particolare, grazie all'indipendenza dei parametri è possibile costruire la priori per i parametri di ogni nodo separatamente: se il nodo X_i ha padri $\mathbf{Pa}_i$ in una struttura, è sicuramente possibile identificare un grafo completo (fra gli $n!$) in cui X_i abbia anche quel particolare insieme di padri. In conclusione, date le stime su α e p(**X**|$\mathbf{m}_c$) e la (2.22), si possono ricavare gli iperparametri per la Dirichlet - priori $Dir(\boldsymbol{\theta}_{ij}| \alpha_{ij1},\ldots,\alpha_{ijr_i})$.

2.2.1.2 PRIORI PER LA STRUTTURA

Il modo più semplice per la stima della distribuzione a priori per la struttura del modello è assumere che ogni struttura sia egualmente probabile. Naturalmente, questa assunzione non è accurata ed usata soltanto per convenienza.

Un criterio più efficiente è l'intervento dell'esperto[37] affinché indichi la presenza/mancanza di alcuni collegamenti fra le variabili del dominio ed imponendo, quindi, una distribuzione a priori uniforme per gli archi su cui non si ha informazione.

[36] Numero delle configurazioni dei padri di X_i.

[37] Questo criterio è ad esempio disponibile nel tool BNC - Bayesian Power Constructor – Vedere l'elenco dei software nel Capitolo 1.

Anche in tale ambito, come riportato nelle opere di Heckerman, i ricercatori hanno ricavato alcune assunzioni semplificative.
Buntine, ad esempio, descrive un insieme di asserti che conduce ad un approccio più efficiente.
La prima assunzione è che le variabili possano essere ordinate. La seconda enuncia, in breve, che la presenza o assenza di archi sono eventi mutuamente indipendenti.
Un approccio alternativo consiste nell'utilizzare un modello, di partenza, a priori. L'idea di base è penalizzare la probabilità a priori di qualsiasi struttura a seconda di quanto si discosti dal modello concepito in partenza.
[HEC94][HEC95][MEE97]

2.2.2 MISSING VALUES E HIDDEN VARIABLES (CENNI)

Prima di accennare alle metodologie, è importante osservare che l'assenza nei dati di una variabile (*hidden variable - variabile nascosta*) o del valore ad essa associato (*missing value – valore mancante*) spesso può dipendere dallo stato della variabile stessa. Per comprendere tale affermazione è opportuno riportare un breve esempio.
Un'osservazione mancante nello studio dell'efficacia di una medicina può essere non rilevato a causa del peggioramento di un paziente che non rappresenterebbe, quindi, un teste attendibile per la ricerca. Il problema si presenta allorché questo peggioramento è un effetto collaterale della medicina!
Per ipotesi, quindi, l'assenza delle informazioni relative ad una variabile è indipendente dallo stato.
Supponiamo di osservare un singolo caso (record del databse) incompleto. Siano $\mathbf{Y} \subset \mathbf{X}$ e $\mathbf{Z} = \mathbf{X} \backslash \mathbf{Y}$ gli insiemi, rispettivamente, delle variabili osservate (**Y)** e non osservate (**Z**). Con l'assunzione di *indipendenza dei parametri* e di *likelihood multinomiale*, Spiegelhalter e Lauritzen hanno ricavato la distribuzione a posteriori di $\boldsymbol{\theta}_{ij}$ per la struttura **m**:

$$p(\boldsymbol{\theta}_{ij} \mid \boldsymbol{y},\boldsymbol{m}) = \sum_{z} p(\boldsymbol{z} \mid \boldsymbol{y},\boldsymbol{m})\, p(\boldsymbol{\theta}_{ij} \mid \boldsymbol{y},\boldsymbol{z},\boldsymbol{m}) =$$

$$= (1 - p(\boldsymbol{pa}_i^j \mid \boldsymbol{y},\boldsymbol{m}))\left\{p(\boldsymbol{\theta}_{ij} \mid \boldsymbol{m})\right\} + \sum_{k=1}^{r_i} p(x_i^k, \boldsymbol{pa}_i^j \mid \boldsymbol{y},\boldsymbol{m})\left\{p(\boldsymbol{\theta}_{ij} \mid x_i^k, \boldsymbol{pa}_i^j, \boldsymbol{m}\right\}$$

y variabili osservate, z variabili non osservate

Ogni termine fra parentesi graffe è una distribuzione di Dirichlet. Quindi, premesso che sia X_i che $\mathbf{Pa}_i$ siano osservati, la distribuzione a posteriori di $\boldsymbol{\theta}_{ij}$ sarà una combinazione lineare di distribuzioni di Dirichlet. Per un secondo caso incompleto, alcune o tutte delle componenti di Dirichlet saranno , a loro volta, una combinazione. Continuando ad osservare casi incompleti, ovvero ogni valore di **Z**, la distribuzione a posteriori per $\boldsymbol{\theta}_{ij}$ conterrà un numero esponenziale di componenti di Dirichlet, rendendo il calcolo intrattabile. [HEC95]

Per essere più pratici, supponiamo di avere due variabili binarie X_1 e X_2 con stati $\{x_1,\underline{x}_1\},\{x_2,\underline{x}_2\}$. Per X_2 si osserva lo stato x_2 mentre gli stati di X_1 sono sconosciuti: sono possibili allora due possibili configurazioni per il database $\{x_1,x_2\}$ o $\{\underline{x}_1,x_2\}$. Il valore del parametro relativo a X_2, condizionato dalla presenza di missing value, $\theta_{2|1}$ sarà:

$$p(\theta_{2|1}|x_2) = p(\theta_{2|1}|x_1,x_2)p(x_1|x_2) + p(\theta_{2|1}|\underline{x}_1,x_2)\, p(\underline{x}_1|x_2)$$

Poiché in questo esempio stiamo considerando variabili binarie, $p(\theta_{2|1}|x_1,x_2)$ e $p(\theta_{2|1}|\underline{x}_1,x_2)$ sono Beta distribuzioni mentre la posteriori, $p(\theta_{2|1}|x_2)$, è una loro combinazione con coefficienti $p(x_1|x_2)$, $p(\underline{x}_1|x_2)$. [KRA98]

Nell'eventualità di casi incompleti, per usufruire dei metodi sopra esposti in assenza di missing values e hidden variable, si dovrebbero esplorare tutte le possibili alternative per "sostituire" i valori mancanti, ad esempio, con valori validi. Ne consegue il ricorso a tecniche di approssimazione, quali, ad esempio, i metodi basati sul campionamento Monte Carlo,come il Gibbs-Sampling[38], o sull'approssimazione gaussiana, adatta, rispetto al precedente, quando la dimensione del campione è notevole.

[38] Dettagli sul Gibbs Sampling sono presenti in Operations for Learning with Graphical Models di W.L. Buntine [BUN94], nei contributi di Heckerman [HEC94] [HEC95] e di P.J.Krause [KRA98].

2.2.2.1 L'APPROSSIMAZIONE GAUSSIANA

L'idea alla base di questa approssimazione è che, per una grande quantità di dati la relazione

$$p(\boldsymbol{\theta}_{ms}|D,\mathbf{m}) \sim p(D|\boldsymbol{\theta}_{ms}, \mathbf{m})\cdot p(\boldsymbol{\theta}_{ms}| \mathbf{m})$$

possa essere approssimata da una distribuzione di tipo gaussiano. In particolare, poniamo

$$g(\boldsymbol{\theta}_m) \equiv \log(p(D \mid \boldsymbol{\theta}_m, \boldsymbol{m}) \cdot p(\boldsymbol{\theta}_m \mid \boldsymbol{m})) \qquad (2.23)$$

e sia $\tilde{\boldsymbol{\theta}}_m$ la configurazione di $\boldsymbol{\theta}_m$ che massimizza $g(\boldsymbol{\theta}_m)$: essa soddisfa il criterio **MAP** per $\boldsymbol{\theta}_m$ (Maximum A Posteriori) in quanto massimizza $p(\boldsymbol{\theta}_m|D, \mathbf{m})$. Quindi espandendo $g(\boldsymbol{\theta}_m)$ rispetto a $\tilde{\boldsymbol{\theta}}_m$ si ottiene

$$g(\boldsymbol{\theta}_m) \sim g(\tilde{\boldsymbol{\theta}}_m) - \frac{1}{2} (\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_m) \mathbf{A} (\boldsymbol{\theta}_s - \tilde{\boldsymbol{\theta}}_m))^t$$

dove **A** è l'Hessiano[39], preso con segno meno, di $g(\boldsymbol{\theta}_s)$ valutato in $\tilde{\boldsymbol{\theta}}_m$. Elevando alla potenza di *e* ed usando la (2.23) si ottiene

$$p(\boldsymbol{\theta}_m \mid D, \boldsymbol{m}) \propto p(D \mid \boldsymbol{\theta}_m, \boldsymbol{m}) p(\boldsymbol{\theta}_m \mid \boldsymbol{m}) \approx p(D \mid \tilde{\boldsymbol{\theta}}_m, \boldsymbol{m}) p(\tilde{\boldsymbol{\theta}}_m \mid \boldsymbol{m}) \exp\left\{-\frac{1}{2}\left(\boldsymbol{\theta}_m - \tilde{\boldsymbol{\theta}}_m\right) A \left(\boldsymbol{\theta}_m - \tilde{\boldsymbol{\theta}}_m\right)^t\right\}$$

che rappresenta un'approssimazione gaussiana per $p(\boldsymbol{\theta}_m|D,\mathbf{m})$.

[39] Matrice delle derivate del secondo ordine per una funzione scalare a valori vettoriali.

Usando l'approssimazione gaussiana si può elicitare la marginal likelihood sostituendo l'equazione precedente in $p(D \mid \mathbf{m}) = \int p(\boldsymbol{\theta}_m \mid \mathbf{m}) p(D \mid \boldsymbol{\theta}_m, \mathbf{m}) d\boldsymbol{\theta}_m$, integrando[40] ed estraendo il logaritmo del risultato si ha

$$\log p(D \mid \boldsymbol{m}) \approx \log p(D \mid \tilde{\boldsymbol{\theta}}_m, \boldsymbol{m}) + \log p(\tilde{\boldsymbol{\theta}}_m \mid \boldsymbol{m}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|A|$$

Dove d è la dimensione di $g(\boldsymbol{\theta}_m)$. Per un modello causale con distribuzione multinomiale tale dimensione è fornita da $\prod_{i=1}^{n} q_i (r_i - 1)$.

Un'approssimazione molto efficiente, per i calcoli, ma meno accurata la si ottine considerando solo quei termini dell'equazione precedente che aumentano con N (dimensione del campione)[41]. Per N elevato si approssima $\boldsymbol{\theta}_m$ con il **Maximum Likelihood** ("$\boldsymbol{\theta}_m$ circonflesso" nella formula seguente) ottenendo il cosiddetto Bayesian Information Criterion (dovuto a Schwarz - 1978):

$$\log p(D \mid \boldsymbol{m}) \approx \log p(D \mid \hat{\boldsymbol{\theta}}_m, \boldsymbol{m}) - \frac{d}{2}\log N$$

Bayesian Information Criterion (BIC)

Il BIC non dipende dalla distribuzione a priori. L'approssimazione è abbastanza intuitiva: un termine misura il grado di adattamento del modello ai dati (primo addendo), l'altro termine penalizza la complessità del modello (secondo addendo)[42].

[40] La tecnica di approssimazione usata nell'integrazione è il metodo di Laplace, quindi spesso l'approssimazione della marginal likelihhod è indicata, in letteratura, come approssimazione di Laplace.
[41] log ($p(D|\boldsymbol{\theta}_m,\mathbf{m})$) aumenta linearmente con N, log|A| incrementa come dlogN.
[42] Il BIC si avvicina alla metrica ispirata dal principio MDL illustrata nel capitolo relativo allo Structural Learning.

2.3 PARAMETER LEARNING

L'assunzione di database completo rende particolarmente chiara la stima dei parametri.
Infatti, dalla statistica, lo stimatore della probabilità soddisfa il criterio Maximum Likelihood e fornisce, quindi, una stima accurata (N_{ijk} è il numero di occorrenze in D dei casi in cui $X_i = x_i^k$ e $\mathbf{Pa}_i = \mathbf{pa}_i^j$.)

$$p(x_i^k \mid \mathbf{pa}_i^j, \boldsymbol{\theta}_i, \mathbf{m}) = \theta_{ijk} = \frac{N_{ijk}}{\sum_k N_{ijk}}$$

dall'approccio bayesiano, per l'assunzione di Dirichlet, segue invece

$$p(x_i^k \mid \mathbf{pa}_i^j, \boldsymbol{\theta}_i, \mathbf{m}) = \theta_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$$

In questo paragrafo si presenta uno degli algoritmi più diffusi per l'apprendimento dei parametri: l'algoritmo **EM**. A differenza dell'approccio bayesiano, l'EM è rilevante perché gestisce il learning anche in presenza di missing values.

L'algoritmo di **E**xpactation-**M**aximization è interpretabile come una versione deterministica del Gibbs sampling (citato in precedenza) per la stima, con il criterio MAP o ML, dei parametri del modello.

Nella fattispecie, l'EM è considerata una versione "deterministica" (nel senso che non si ha nessun campionamento) del Gibbs sampling perché, a differenza di questo ultimo, determina i valori mancanti stimando la media o la moda dei valori conosciuti, ottenendo così delle statistiche sufficienti. [HEC95][STE00]

2.3.1 L'ALGORITMO EM

Nelle analisi di domini reali, spesso, i dati disponibili per l'apprendimento sono incompleti poiché può essere difficile o anche impossibile osservare alcune variabili. E' perciò importante che un algoritmo di apprendimento sappia fare un uso efficiente dei dati osservati. In presenza di missing value il problema dell'apprendimento diventa molto più difficile; l'elicitare i parametri è un punto importante sia a causa della difficoltà di effettuare accurate stime numeriche della probabilità, sia perché l'apprendimento dei parametri è talvolta parte integrante di applicativi dedicati all'apprendimento della struttura (nel capitolo sullo Structural Learning si accennerà difatti ad allo Structural EM).

L'utilizzo dell'algoritmo di **E**xpectation - **M**aximization (EM) per il parameter learning è dovuto a A. Dempster (1977).

Assegnata una rete Bayesiana con struttura **m** ed un database di osservazioni campionarie, l'approccio dell'EM, per l'apprendimento dei parametri in presenza di missing values, segue il criterio MAP o ML. Si inizia considerando una configurazione di parametri $\boldsymbol{\theta}_m$ iniziale (casuale, o fornita dall'esperto o indicando tutti valori delle variabili come equiprobabili); poi, si elaborano le statistiche sufficienti (*Expectation step*) in modo da sostituire i missing values ed ottenere un data set completo. Nel caso di variabile discreta si ha

$$E_{p(x|D,\theta_s,S)}(N_{ijk}) = \sum_{i=1}^{N} p(x_i^k, pa_i^j \mid y_l, \theta_s, S) \qquad (2.24)$$

dove $\mathbf{y}_l$ è il possibile l-esimo caso incompleto in D e N il numero di record in D. In alternativa, è possibile usare un algoritmo di inferenza bayesiana per valutare i termini mancanti. Le statistiche sufficienti diventano quindi le statistiche per creare un database completo D_c.

Assumendo di massimizzare (*Maximization step*) secondo il criterio ML, si determina la configurazione $\boldsymbol{\theta}_m$ che massimizza $p(D_c| \boldsymbol{\theta}_m, \mathbf{m})$. Nel caso di variabili multinomiali discrete segue

$$\theta_{ijk} = \frac{E_{p(x|D,\theta_s,S)}(N_{ijk})}{\sum_{k=1}^{r_i} E_{p(x|D,\theta_s,S)}(N_{ijk})}$$

mentre seguendo l'approccio MAP (considerando ancora valida l'assunzione di Dirichlet distribution)

$$\theta_{ijk} = \frac{\alpha_{ijk} + E_{p(x|D,\theta_s,S)}(N_{ijk})}{\sum_{k=1}^{r_i} (\alpha_{ijk} + E_{p(x|D,\theta_s,S)}(N_{ijk}))}$$

Il problema dell'apprendimento, sia strutturale che dei parametri, rappresenta un problema di ottimizzazione a più variabili. In tale proposito è opportuno ricordare che le funzioni a più variabili sono dotate di massimi locali (local maxima - nell'intorno di un intervallo) e massimo assoluto (il valore massimo della funzione nel campo di esistenza). Gli algoritmi di learning, quindi, presentano l'inconveniente di determinare un massimo locale e non globale, determinando un possibile buon modello ma non il migliore. [HEC95]

Riassumendo, l'approccio EM, considerata una rete bayesiana S, descritta da un vettore $\overline{\theta}$ di parametri, e dato un insieme D di osservazioni, risolve il problema del learning parameter apprendendo un nuovo vettore di parametri $\tilde{\theta}$ per S da D. Due fattori influenzano la scelta di $\tilde{\theta}$: il grado di adattamento a D e il non allontanarsi troppo dal modello preesistente $\overline{\theta}$. Per rispettare questi vincoli si introduce una funzione F da ottimizzare composta dal logaritmo della (2.24) (log - likelihood) e dalla distanza fra $\overline{\theta}$ e $\tilde{\theta}$. La forma esatta della funzione F dipende dai pesi (i coefficienti) che forniamo al log - likelihood e alla distanza ed anche dalla scelta di come valutare ($\overline{\theta}$ - $\tilde{\theta}$). Possibili misure per la distanza sono, ad esempio, *relative entropy* (o Kullback-Leibler \ KL-divergence), *chi - quadro* (che è un'approssimazione lineare della precedente). Il processo di apprendimento dell'EM è iterativo: ad ogni step, si migliora la computazione di $\tilde{\theta}$ a partire da $\overline{\theta}$ (risultato dello step precedente) e da D fintantoché non si raggiunge un criterio di convergenza o un massimo numero di iterazioni. [KOL97][COZ01]

Bauer, Koller e Singer in "Update rules parameter estimation in Bayesian Networks" [KOL97] illustrano l'algoritmo, da loro proposto, EM(η)[43] che è una generalizzazione dell'algoritmo EM standard discusso in questo paragrafo; anzi sono riscontrati dei miglioramenti nei tempi di convergenza verso l'effettiva distribuzione delle probabilità scegliendo un valore $\eta = 1.8$. Tuttavia lo standard EM è più robusto nei confronti del problema del local maxima. La procedura dell'EM(η) si presta anche per l'on-line learning dove la scelta di η può essere adattata al numero di campioni acquisiti. In particolare una versione dedicata all'on-line learning, chiamata Voting EM, è illustrata in "Online learning of Bayesian Network parameters" di Cohen, Bronstein e Cozman. [COZ01]

[43] In Appendice.

2.4 STRUCTURAL LEARNING

Il problema dello Structural Learning (SL) affronta l'apprendimento della struttura di un modello grafico, nello specifico di una rete bayesiana, da un database di esempi. L'apprendimento della struttura, ovvero delle dipendenze causali del modello grafico di probabilità, è spesso il primo passo del ragionamento in condizioni di incertezza: difatti in molte applicazioni si parla di C*ausal Discovery* per sottolineare l'estrapolazione dei legami fra le variabili di un dominio.

Heckerman, Geiger e Chickering in "Learning Bayesian Networks: The combination of knowledge and statistical data" [HEC94], hanno mostrato che tale problema è NP, anche senza considerare variabili latenti (missing) o nascoste. In particolare Chickering ha affrontato in modo rigoroso il problema nell'opera "Learning is NP-complete". [CHI96]

Apprendere la struttura di una rete dai dati è spesso definito come problema di *selezione del modello* (*model selection problem*) nel senso che ad un dominio corrispondono modelli differenti e soltanto uno deve essere selezionato in base ai dati. Di recente, gli studiosi parlano di *modello di incertezza* (*model uncertainity*) perché si è constatato che la selezione di un singolo modello "migliore" non è realizzabile mentre è preferibile prendere in considerazione un sottoinsieme di "ragionevoli" grafi quantificando l'incertezza ad essi correlata. In tale proposito, è importante ricordare la **condizione di equivalenza di Markov**: "*due grafi sono Markov equivalenti se implicano lo stesso insieme di indipendenze condizionate*", per cui bisognerebbe soprattutto selezionare modelli fra loro non Markov - equivalenti. Una esposizione più rigorosa sullo structural learning e alcuni algoritmi studiati saranno approfonditi nel prossimo capitolo.

3 STRUCTURAL LEARNING

3.1 INTRODUZIONE

Il formalismo delle reti probabilistiche si presta ad una rappresentazione immediata e comprensibile della conoscenza, di uno o più esperti, su un dominio in condizioni di incertezza.

Il processo di apprendimento (learning), in particolare lo Structural, implica continue revisioni (testing) da parte degli esperti su un modello concepito in una fase iniziale: un apprendimento dai dati (*learning from data*) consentirebbe sia di agevolare la fase iniziale, per determinare un grafo di partenza, sia di supportare il ciclo di testing.

Per apprezzare tale opportunità si consideri che lo stesso Beinlich, per giungere alla rappresentazione di un dominio che consentisse di modellare potenziali problemi di anestesia in sala operatoria, la rete ALARM, ha impiegato circa 10 ore per costruire un modello di 37 variabili (ogni nodo ha dai due ai quattro stati) legate da 46 archi e 20 ore per assegnare le probabilità condizionate; un algoritmo presenta, invece, un run-time dell'ordine delle decine di minuti (a seconda della complessità della rete e del numero di campioni).

Sebbene in un primo momento il *learning from data* sia stato concepito proprio per affiancare il lavoro degli esperti, nell'ambito dell'Intelligenza Artificiale sono stati sviluppati *learning algorithm* con l'intento di sostituire la "human experience". Tuttavia, non è stato ancora implementato un metodo in grado di costruire un modello grafico così come concepito dalla "expert's knowledge".

L'attenzione verso gli algoritmi di apprendimento, in particolare modo dello Structural Learning (apprendimento della struttura), nasce dai risultati lusinghieri foniti da questo approccio non solo nel ricostruire un grafo ma soprattutto nel rivelare spesso legami, o pattern, spesso ignorati dagli stessi esperti (data mining).

3.2 LO STRUCTURAL LEARNING

L'intento dello Structural Learning è esplicitare, dalle osservazioni su un insieme di variabili (dominio), "cosa è conneso a cosa", cioè individuare le relazioni fra le entità del dominio e, in secondo luogo, specificarne, se possibile, un vincolo di causalità. Varie sono le soluzioni ideate per perseguire questo obiettivo: in una prima fase il learning approach è classificabile in *non* bayesiano (dependence analysis) e bayesiano (search & score). L'approccio non bayesiano esegue dei test (di indipendenza) statistici sui database di osservazioni campionarie, attribuibili ad una distribuzione di probabilità implicita nel modello[44], per inferire l'esistenza di relazioni di dipendenza fra le variabili del dominio. L'approccio bayesiano, come esposto nel capitolo precedente, codifica l'incertezza sulla struttura di un dominio $\mathbf{X} = \{X_1,\ldots,X_n\}$ introducendo una variabile aleatoria, **M**, i cui stati sono proprio le possibili strutture (*structure hypothesis*) associate ad **X**. Dopodiché si sceglie il modello **m** che massimizza la probabilità a posteriori p(**m**|D), dove D è il database di campioni. Per essere più precisi, nell'ambito dell'approccio bayesiano, va delineata un'ulteriore distinzione fra l'approccio puramente bayesiano e l'*optimization*, in cui il miglior modello non massimizza la probabilità a posteriori ma un'opportuna *misura di qualità (scoring function, metrica, funzione di costo)* dell'adattamento del modello **m** ai dati in D.

In sintesi i differenti metodi di apprendimento della struttura dai dati sono:

- ⇒ **approccio bayesiano:** si sceglie il modello con la più alta probabilità a posteriori utilizzando una Bayesian scoring metric: BD (Bayesian Dirichlet) (o la sua estensione, con opportune ipotesi, "BDe" come illustrato da Heckerman, Geiger, e Chickering in "Learning Bayesian Network: the combination of Knowledge and statistical data"[HEC94]).
- ⇒ **optimization (scoring function):** la base è sempre l'approccio bayesiano. La ricerca della struttura nello spazio dei possibili modelli è basata però sulla massimizzazione di metriche che misurino "quanto" ogni possibile

[44] Cioè i campioni sono stati generati in riferimento ad una certa struttura di rete e secondo una data legge di distribuzione delle probabilità, associata alla rete stessa, che quantifica l'incertezza sul dominio.

struttura si adatti ai dati. Un'importante caratteristica di queste metriche è la *proprietà di scomposizione*, ovvero

$$Score(G, D) = \sum_i Score(X_i \mid \mathbf{Pa}(X_i), N_{X_i, \mathbf{Pa}(X_i)}) \qquad (3.1)$$

dove N indicha le statistiche di X_i e $\mathbf{Pa}(X_i)$ (il numero di istanze in D delle possibili coppie $(X_i, \mathbf{Pa}(X_i))$. La decomposizione ottenuta permette di computare facilmente la (3.1), per ogni possibile variazione locale del grafo (inversione, rimozione o aggiunta di un arco), rispetto ad un modello di riferimento senza necessariamente ricalcolare tutti i termini della sommatoria ma valutando solo quelli interessati dalla modifica.

Il problema dell'apprendimento è una ricerca della struttura che meglio si adatti ai dati. In genere, si inizia con un grafo privo di legami e si aggiungono dei link, testando con la metrica se la nuova struttura sia migliore di una precedente (in alternativa si potrebbe partire da una rete completamente connessa e rimuovere gli archi). In caso affermativo, si mantiene il nuovo arco aggiunto e si cerca di aggiungerne un altro, continuando fino a che nessuna struttura "nuova" è migliore di una precedente. La maggior parte di questi algoritmi usano un metodo di ricerca euristico; l'uso dell'euristica è giustificato in quanto l'apprendimento di reti bayesiane è un problema di complessità Non Polinomiale (NP).

⇒ **constraint satisfaction (testing delle relazioni di indipendenza):** si cerca di stimare le proprietà di indipendenza condizionata fra le variabili applicando test di ipotesi statistica (ad esempio test del X^2) sui campioni presenti nel database. La rete appresa deve soddisfare tutti i vincoli implicati dalla dipendenze condizionate empiriche riscontrate nei dati; il risultato degli algoritmi *constraint-based* è un PDAG[45] (Partially Directed Acyclic Graph) chiamato anche pattern o grafico essenziale (essential graph)

[45] Un PDAG è un chain graph ovvero un grafo senza cicli con percorsi sia diretti che non orientati.

L'approccio constraint satisfaction è efficiente, in termini di tempo, ma soggetto agli errori del test, quindi l'approccio optimization, anche se più lento, è spesso preferito per lo Structural Learning from data[46].

L'approccio bayesiano, in cui rientra l'optimization, in effetti presenta alcuni vantaggi su quello non bayesiano ma entrambi gli approcci mostrano pregi e difetti. L'approccio constraint-based è più efficiente dell'optimazion per le *sparse network* (le reti che non sono densamente connesse) ma richiede un numero esponenziale di verifiche di Conditional Independence (CI) e spesso sono test di ordine (ovvero il numero di variabili condizionanti - *condition set*) elevato. Studiosi quali Cooper e Herskovits hanno evidenziato come i test CI, in presenza di un condition set di elevate dimensioni, necessitano di enormi database di campioni per essere attendibili [COO92]. Il risultato, inoltre, è vincolato al livello di fiducia (o significance level) che rappresenta il valore di soglia designato per l'interpretazione del test. Invece nell'approccio bayesiano è fornito naturalmente un criterio di stopping per la ricerca, poiché la fase di searching termina quando viene individuato il modello con la massima probabilità a posteriori (o il massimo valore della scoring function); d'altro canto, i metodi search & score non è detto che trovino la rete migliore a causa della natura euristica ed analizzano un numero superiore di modelli rispetto ai test CI risultando, quindi, meno efficienti rispetto alla dimensione temporale. [COO92] [HEC94] [BUN96] [SIN94] [FRI98] [KRA98] [SUZ99] [CHE97]

In realtà le due metodologie, a volte, sono opportunamente integrate: ad esempio, l'approccio non bayesiano è usato per fornire o una rete di partenza (in "A hybrid anytime algorithm for the construction of causal models from sparse data" di D. Dash e M.J. Druzdzel [DAS99]) o un possibile ordinamento delle variabili come input per un algoritmo bayesiano (in "Construction of bayesian network structures from data: a brief survey and efficient algorithm" di M.Singh e M.Valtorta [SIN94]).

[46]E' un'affermazione ricorrente in letteratura, come nell'articolo di Wong, Lam, Leung e Cheng in "Applying Evolutionary algorithms to discover knowledge from medical databases" [WON99].

3.2.1 L' ALGORITMO DI STRUCTURAL LEARNING

In generale, non considerando vincoli restrittivi sulla ricerca del grafo e prescindendo dal campo specifico di applicazione, i più diffusi algoritmi di apprendimento strutturale, attualmente, generano reti che differiscono da quelle originali di riferimento, anche con database contenenti qualche migliaio di casi, perché:

- evidenziano legami inesatti; archi in più o in meno rispetto al grafo originale;
- invertono la direzione degli archi.

Il problema dello structural learning, quindi, è tutt'altro che risolto[47]: difatti, se tali risultati possono essere considerati soddisfacenti in alcuni settori, non sono accettabili in altri, in cui, ad esempio, non si possono imporre i vincoli necessari dettati dall'euristica per la mancanza di sufficienti informazioni a priori sul dominio. Un algoritmo di Structral Learning, in generale, comprende i seguenti step:

- Collezionare i dati
- Determinare le variabili dai dati acquisiti
- Determinare un grafo di partenza
- Scegliere il metodo di di ricerca (bayesian approach) o il test statistico (constraint satisfaction approach)

[47] A conferma di ciò, è da notare che tra i molti software in commercio per le BN (in Appendice), pochi implementano lo Structural Learining.

Approccio bayesiano

1. Aggiungere, eliminare o invertire un arco X_i - X_j;
2. Se la modifica al passo 1) migliora la scoring function renederla effettiva;
3. Ripetere i passi 1- 2) finchè non è soddisfatto il criterio di stopping locale: ad esempio, una ricerca *candidate - based* termina quando l'insieme dei nodi "candidati a padre" di X_i non cambia - $C_i^n = C_i^{n-1}$;
4. criterio di stopping globale (fine dell'algoritmo): *score - based*, non si ha variazione nello score, $Score(B_n) = ScoreB(_{n-1}))$.

Approccio constraint satisfaction

1. Fissare un Significance Level (soglia) per il test statistico;
2. Scegliere una coppia di nodi X, Y;
3. Eseguire i test di indipendenza fra X e Y fissato un insieme di nodi C (esclusi X e Y);
4. Se il test di indipendenza fallisce inserire un legame fra X e Y;
5. Terminare (locale) se sono stati considerati tutti i possibili C per la coppia (X,Y);
6. Terminare (globale) se sono state analizzate tutte le possibili coppie, altrimenti step 2).
7. Orientare i legami rilevati.

3.2.1.1 I METODI DI RICERCA

In letteratura, si evince che sono tre le principali strategie di ricerca applicate allo structural learning di modelli grafici: *simple heuristic search*, *evolutionary computation* e *Markov Chain Monte Carlo method* (MCMC). Tutti hanno lo stesso obiettivo: il *learning from data* di modelli grafici. Con la simple heuristic search o l'evolutionary computation (questo ultimo è così denominato poiché usa la tecnica di programmazione Evolutionary Programming, simile agli algoritmi genetici), di solito, si sceglie la struttura migliore fra quelle che massimizzano un certo criterio (di adattamento ai dati - scoring function; con potere predittivo - la probabilità a posteriori); con il metodo MCMC, invece, si cerca di fornire un quadro accurato dello spazio di ricerca in modo da disporre dell'incertezza relativa ad ogni modello ottenuto con metodi di campionamento Monte Carlo.
Il cuore dell'algoritmo di optimization e dell'approccio bayesiano, ed anche di alcuni algoritmi ibridi, è proprio la fase di ricerca che può essere condotta in modo esaustivo, **greedy search**, o secondo un **approccio euristico**.
L'algoritmo esaustivo esplora tutte le combinazioni che è possibile instaurare fra i nodi della rete; l'evidente svantaggio è nel notevole numero di modelli da analizzare all'aumentare delle variabili del dominio.
In un approccio euristico, invece, si considerano eventuali informazioni a priori in quanto la ricerca nello spazio delle possibili strutture potrebbe indurre ad esaminare modelli estremamente irragionevoli. Ad esempio, se è nota la presenza di un arco fra due variabili allora è opportuno considerare solo quei grafi in cui quel legame è esplicitato, restringendo, in questo modo, l'ambito dello screening.
In entrambi i casi la natura del processo di ricerca è *incrementale*, cioè ad ogni iterazione la struttura di partenza è modificata per osservare come varia la scoring function e l'eventuale modifica

- ☑ aggiunta
- ☑ rimozione
- ☑ inversione

di un arco fra due variabili è resa effettiva se la metrica esibisce un valore maggiore. In particolare, bisogna evitare che queste operazioni violino la

condizione di aciclicità che caratterizza le reti bayesiane; un apposito algoritmo deve verificare che l'operazione non implichi la creazione di un ciclo[48].

I metodi di ricerca comprendono una fase di *inizializzazione* ed una fase di *ricerca* vera e propria: ad esempio nell'algoritmo K2 (illustrato in dettaglio in una delle sezioni successive) l'inizializzazione consiste nel fornire un ordinamento delle variabili dai padri ai figli.

La fase di inizializzazione è importante per almeno due aspetti. Primo, come l'esperto ha bisogno di continue modifiche per raffinare una rete di partenza, così la scelta di un modello opportuno da cui iniziare la fase di searching migliora l'efficienza dell'algoritmo, in termini di tempo, e rende più probabile la convergenza verso il "true model". Possibili scelte per un grafo iniziale sono una rete con legami generati in modo casuale, una "priori network" in cui è codificata la expert's knowledge, o, come spesso accade, la struttura senza legami. Il metodo più intuitivo, inoltre, è assumere, ad ogni iterazione, come modello di riferimento quello appreso durante l'elaborazione precedente. Secondo, al crescere del numero di variabili, il confronto di tutti i possibili modelli non è realizzabile, presenta complessità NP; i metodi euristici o le intuizioni dell'esperto permettono di limitare lo spazio di ricerca. Per chiarire,vediamo cosa cambia, se all'inizio l'esperto attribuisce un ordinamento alle variabili. Sia β: U $\rightarrow$ {1,…,n} un ordinamento topologico su **X** per cui, date due variabili X e Y, X viene prima di Y, $X <_\beta Y$, se $\beta(X) < \beta(Y)$; come hanno indicato Cooper e Herskovits, la cardinalità |G| dello spazio G dei possibili modelli non è più esponenziale: il numero delle possibili strutture di rete è $2^{\frac{n(n-1)}{2}}$ poiché l'insieme dei possibili padri per un singolo nodo X_i è 2^{j-1}, con j = 1,…, *n* (l'esponente è (j-1) perché j≠i), ovvero $|G| = \prod_{j=1}^{n} 2^{j-1}$. In proposito, si consideri questo semplice esempio. Date tre variabili A,B,C, il numero di possibili strutture è G(3) = 25[49]; con l'ordinamento $A <_\beta B <_\beta C$, devono essere escluse dalla ricerca tutte le strutture che non

[48] In "Topological Sorting of Large Networks" di A.B.Kahn - Communications of the ACM, vol.5, 558-562, 1962 - è illustrato un algoritmo per determinare l'ordinamento topologico dei nodi di un grafo. Molto semplicemente l'acicilicità è indicata dalla presenza di un nodo più volte in questo ordinamento: A→B→C ammette come ordinamento A,B,C mentre per A→B→C→A si ha A,B,C,A quindi è presente un ciclo rispetto ad A.

[49] Si ricordi l'espressione ricorsiva citata nel capitolo precedente.

soddisfano questo vincolo, restringendo il campo di ricerca da 25 ad 8 possibili modelli.

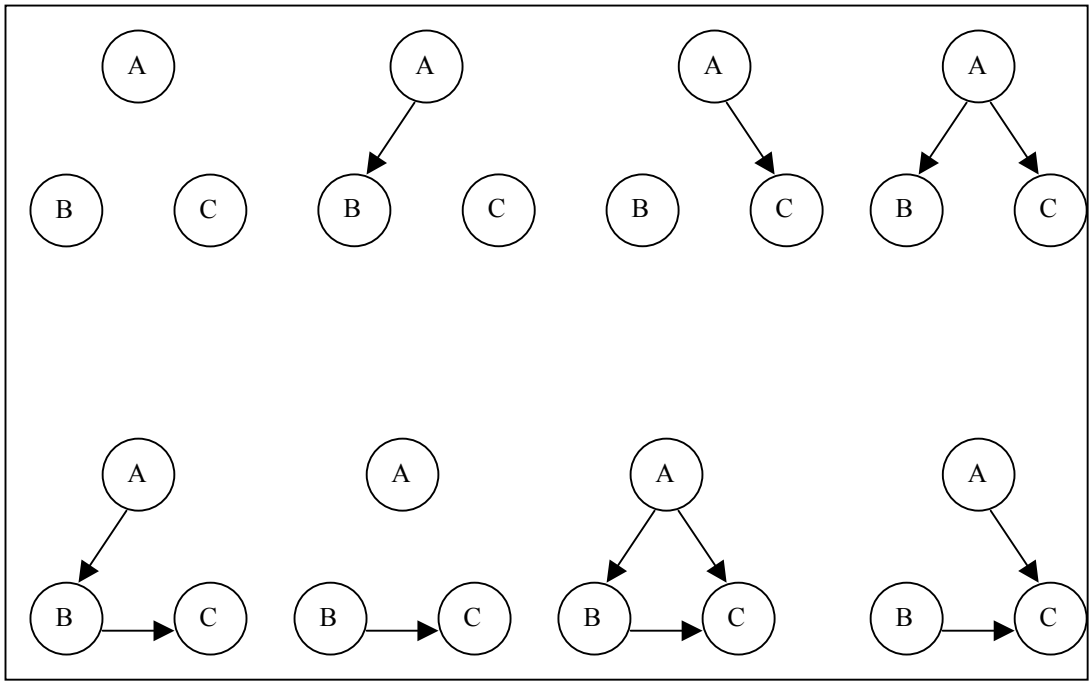


Figura 19 – Le 8 strutture che rispettano il vincolo espresso dall'ordinamento A $<_\beta$ B$<_\beta$ C

Infine è necessario evidenziare il problema del "local maxima". La metrica valutata durante il processo di ricerca è, dal punto di vista matematico, una funzione di più variabili, quindi ammetterà più massimi locali. L'ideale è trovare il modello che determina il massimo assoluto per la scoring function (nell'approccio dependence based non si presenta questo problema). Nella maggior parte dei casi, invece, l'algoritmo restituisce il modello relativo ad un massimo locale: difatti il problema della ricerca del massimo è un caso di ottimizzazione di una funzione con più variabili, ovvero complessità NP. Ad esempio, un possibile metodo per ovviare all'individuazione di un local maxima è di modificare la struttura quando il processo di ricerca evidenzia che le correzioni apportate interessano soltanto una particolare variabile del dominio.

Nel seguito sono presentati brevemente alcuni searching method. Come premessa, specifichiamo alcune notazioni: *E* rappresenta l'insieme di modifiche effettuabili sul grafo e $\Delta(e)$ la variazione della scoring metric causata dalla modifica *e*. Per la

proprietà di scomposizione delle scoring function, se rispetto ad X_i è aggiunto o rimosso un arco, bisogna valutare soltanto la variazione Score(X_i|$\mathbf{Pa}_i$) (varia l'insieme dei padri solo per X_i) per determinare $\Delta(e)$. Allo stesso modo, se una arco tra X_i e X_j è invertito, bisogna valutare sia Score(X_i|$\mathbf{Pa}_i$) che Score(X_j|$\mathbf{Pa}_j$). [COO92] [BOU94] [HEC94] [HEC95] [KRA98]

3.2.1.1.1 Local Search

E' il più semplice degli algoritmi. Scelto un grafo iniziale, si valuta $\Delta(e)$ per tutte le modifiche *e* contenute nell'insieme *E* rendendo la modifica effettiva se implica una $\Delta(e)$ massima e positiva. La procedura termina quando non sono presenti modifiche *E* che implichino un valore positivo per $\Delta(e)$.

Possibili candidati per il grafo di partenza sono la struttura senza connessioni fra le variabili, una random structure o una priori network (una struttura suggerita da un esperto, ad esempio).

Un potenziale problema di questo algoritmo è che si può concentrare nella ricerca di un massimo locale della scoring function; i metodi seguenti evitano questo inconveniente.

3.2.1.1.2 Hill climbing

Si applica una ricerca locale finché non si raggiunge un local maximum. Quindi si perturba in modo casuale la struttura corrente della rete e si ripete il processo per un numero accettabile di iterazioni. Ad ogni stadio è possibile anche memorizzare le *l* migliori strutture (per il model averaging, ad esempio).

Algoritmo hill climbing
Scegli G in qualche modo[50]
While not converged
For each G' in nbd(G)[51]
Compute score(G') = P(G')P(D|G')
G = arg max $_{G'}$ Score (G')*
If Score(G) > Score(G) Then G = G**
Else convereged = true;

[50] I test di indipendenza sono un valido strumento per fornire una rete iniziale.
[51] nbd(G) = neighbourhood of the current state = insieme di DAG che differiscono di 1 arco da G ottenuto aggiungendo, eliminando o cambiando direzione ad un collegamento fra due variabili, rispettando il vincolo dell'aciclicità.

3.2.1.1.3 Simulated annealing

Il Simulated Annealing è un metodo stocastico di ottimizzazione usato per trovare la massima distribuzione di probabilità delle scoring function relative ad uno spazio di ricerca di tipo combinatorio ed in presenza di local maxima.

L'evoluzione del processo di ricerca è controllata da un parametro, definito *temperature*, che viene poi diminuito affinché la ricerca si stabilizzi e converga al massimo globale.

Il parametro temperature è opportunamente inizializzato ad un valore T_0. Quindi si sceglie una possibile modifica *e* a caso e si valuta l'espressione $p = e^{\Delta(e)/T_0}$. Se p>1 allora si apporta la modifica alla struttura altrimenti la si effettua con probabilità p. Si ripete questo processo di selezione e valutazione o α volte o finché si realizzano β cambiamenti alla struttura di rete. Se non si realizza nessuna modifica durante le α iterazioni, si termina la ricerca. Altrimenti si abbassa il valore della temperature moltiplicando T_0 per un fattore $0<\gamma<1$, e si continua il processo. In questo caso, la ricerca termina quando T_0 è stato diminuito più di δ volte. Anche con questo metodo è possibile ottenere le *l* strutture migliori.

L'algoritmo è controllato dai cinque parametri T_0,α,β,γ,δ. Una possibile configurazione iniziale è considerare come grafo la struttura senza connessioni e scegliere T_0 abbastanza grande affinché sia valutata ogni possibile variazione in *E*. In alternativa si può iniziare con un basso valore di T_0 ed usare una random network o priori network.

3.2.1.2 SCORING FUNCTION

Nei metodi *score – based* è definita una funzione che misura il grado di adattamento della struttura ai dati per un possibile modello **m**. In particolare, un algoritmo di structural learning score – based implica:

- la definizione dello spazio di ricerca e quindi anche la specifica di operatori con cui modificare una struttura;
- la scelta di un metodo di ricerca;
- la scelta in una *scoring function*.

Differenti criteri di scoring sono stati applicati in letteraura: Bayesian scoring, entropy-based, minimum message length. Le principali metriche sono:

- ***BD (Bayesian Dirichlet score),*** l'approccio bayesiano puro; e sue approssimazioni quali
  - ***BIC*** (Bayesian Information Criterion):
  - ***AIC*** (A - o Akaike - Information Criterion)
  - ***Log likelihood***, ovvero il logaritmo della probabilità a posteriori espressa dalla BD;
- ***Minimum Description Length (MDL)***, si sfruttano i concetti della teoria dell'informazione. Spesso questa metrica viene usata in combinazione con algoritmi di Evolutionary Programming.
- ***Cross Entropy***: derivata dalla teoria dell'informazione. Usa una opportuna relazione per il confronto delle probabilità.

[COO92] [BOU94] [HEC94] [BUN96] [KRA98]

3.2.1.3 I TEST DI INDIPENDENZA CONDIZIONATA

Se nel bayesian approach è la ricerca il cuore dell'algoritmo, nei metodi constrait-based, o dependence-based, sono i test di indipendenza il fulcro dello Structural Learning: per scegliere i candidati padri di un generico nodo X_i si determina il grado di dipendenza con $X_j \neq X_i$, $j = 1,\ldots,n$. In particolare, nel considerare ogni candidato, si assume che non ci siano indipendenze spurie nei dati: se Y è padre di X allora X non deve risultare indipendente da Y fissato un sottoinsieme di padri diverso da Y.

Il test di indipendenza è una delle possibili applicazioni delle metodologie statistiche per la verifica di ipotesi. I manuali di statistica sono un'ottima fonte per tale studio; di seguito si fa riferimento alla verifica di indipendenza statistica fra variabili aleatorie illustrata in "Probabilità e Statistica" di G. Cicchitelli [CIC] e riportata in dettaglio nell'Appendice.

3.2.1.3.1 Il test χ^2 come test di indipendenza

Si consideri una popolazione statistica le cui unità siano raggruppate secondo le classi A = { A_1,A_2,…, A_r } e B = { B_1,,B_2,…,B_t } le quali modellano due caratteristiche qualitative (come professione e sesso di una persona) o quantitative (peso e statura, ad esempio). Si voglia identificare l'ipotesi di indipendenza tra A e B; se consideriamo A_i e B_j come due eventi indipendenti allora risulterà

$$p(A_i,B_j) = p(A_i)p(B_j)$$

Se tale relazione è valida per ogni coppia (A_i,B_j) si dice che le caratteristiche A e B sono tra loro indipendenti. Dunque l'ipotesi (detta nulla) da verificare è

$$\mathbf{H_0} \rightarrow p(A_i,B_j) = p(A_i)p(B_j),\ i = 1,2,\ldots,r;\ j = 1,2,\ldots,t$$

Dovendo impostare un test di ipotesi statistica, bisogna esprimere il livello di significatività, o fiducia, ovvero la probabilità con cui si determina la "zona di rifuto del test", che in genere è fissata a livelli convenzionali 0,05, 0,01, 0,001. Il livello di significatività (SL), altro non è che la probabilità che la generica statistica S, nel nostro caso χ^2, cada nella zona di rifiuto quando l'ipotesi è vera (in pratica la probabilità che il test fornisca un risultato errato):

$$SL = p(S \text{ nella zona di rifiuto } |H0 \text{ vera})$$

Quanto minore è il valore di SL tanto maggiore è la credibilità di un eventuale rifiuto dell'ipotesi nulla (in quanto si avrebbe bassa possibilità di sbagliare).

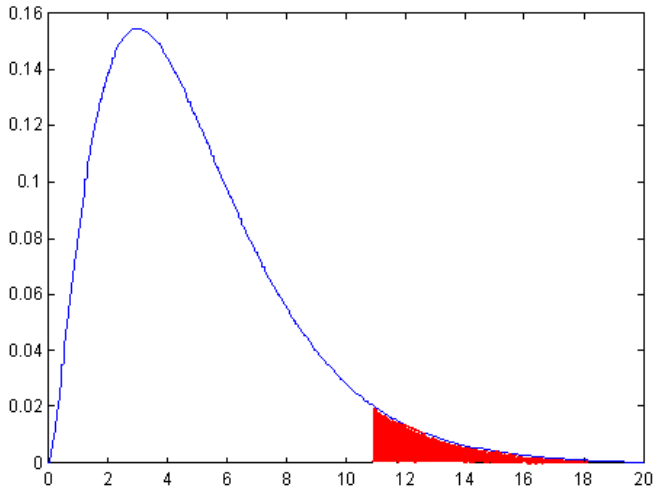


Figura 20 - Distribuzione chi quadro per 5 gradi libertà. In rosso è indicata la zona di rifiuto a SL = 5% = 0.05. L'ipotesi nulla va rifiutata se $\chi^2 > \chi^2_{0.05} = 11.07$

Il test di indipendenza diventa meno agevole quando bisogna considerare l'eventualità di variabili condizionate. In tale proposito, definiamo *condition set* l'insieme delle variabili condizionanti e *ordine del test* la cardinalità di tale insieme. Ad esempio se le variabili A,B fossero condizionate da una terza classe (ovvero variabile casuale) C l'indipendenza sarebbe espressa dalla relazione $p(A,B|C) = p(A|C)\ p\ (B|C)$ e il test verrebbe definito di ordine 1. Diventa così meno immediata la stessa definizione della tabella di contingenza[52] rispetto al test di ordine 0. Un modo pratico, adoperato in questa tesi, per schematizzare il condition set è calcolare la tabella delle contingenze fissando il condition set. Riconsiderando l'esempio A,B|C e, per semplicità, ipotizzando di avere tre variabili binarie, l'idea è di fare riferimento, per il test, ad una tabella di contingenza come la seguente (i = 1,2)

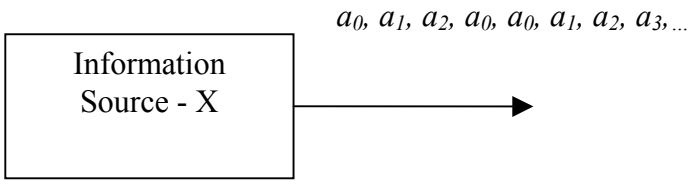
C_i	$A_1\ C_i$	$A_2\ C_i$
$B_1\ C_i$	n_{A1B1Ci}	n_{A2B1Ci}
$B_2\ C_i$	n_{A1B2Ci}	n_{A2B2Ci}

[52] Il significato di tabella delle contingenze è illustrato nell'Appendice: in breve, è una matrice delle occorrenze delle coppie (A_i,B_j) nel campione estratto dalla popolazione.

Si intuisce che i test, all'aumentare dell'ordine, richiedono un maggiore impegno sia computazionale che in termini di tempo (per consultare ogni volta il database di campioni).

3.2.1.3.2 Mutua Informazione

Il test di indipendenza può essere condotto anche valutando la *cross entropia (divergenza di Kullback-Leibler)* o *mutua informazione* fra due variabili.
Nella teoria dell'informazione l'entropia misura il contenuto informativo di una sorgente. [PRO]



$$X = \{a_0, a_1, ..., a_n\}$$
$$H(X) = -\sum_{i=1}^{n} p(X = a_i) \log p(X = a_i)$$

Per essere più precisi, la mutua informazione estende, dalla teoria dell'informazione, la nozione di cross (o joint) entropy e di entropia condizionata.

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y)$$
$$H(X \mid Y) = -\sum_{x,y} p(x,y) \log p(x \mid y) \qquad (3.2)$$

joint e cross entropy

L'entropia condizionata è usata per rappresentare l'informazione attesa, X, dopo avere osservato Y. Nelle reti bayesiane, in modo analogo, se due nodi sono dipendenti, la conoscenza del valore di un nodo fornirà qualche informazione sul valore dell'altro a cui è legato. La mutua informazione[53] fra i nodi A e B (i cui stati indichiamo simbolicamente con *a* e *b*) è definita come segue

[53] Nella letteratura sulle reti Bayesiane, la mutua informazione così espressa è anche indicata come cross entropy.

$$I(A\ ,B)\ =\ \sum_{a,b} P(a,b)\ \log\frac{P(a,b)}{P(a)\,P(b)}$$

$$I(A,B\mid C)\ =\ \sum_{a,b,c} P(a,b,c)\ \log\frac{P(a,b\mid c)}{P(a\mid c)\,P(b\mid c)}$$

Data la reale distribuzione di probabilità *P(x), diremo che A e B* sono indipendenti se e solo se *I(A,B)=0*. Sfortunatamente, spesso non si dispone della reale distribuzione di probabilità bensì, in base ai campioni D estratti da una popolazione, di una stima empirica $\hat{P}_D(x)$ elicitata dalle frequenze relative (principio Maximum Likelihood per lo stimatore della probabilità). Perciò è corretto, in tal caso, usare una $I_D(A,B)$, che approssima la I(A,B) in quanto definita rispetto a $\hat{P}_D(x)$. Per la stessa ragione, non è opportuno considerare come condizione di indipendenza $I_D(A,B) = 0$ bensì A è indipendente da B quando $I_D(A,B) < \varepsilon$, dove $\varepsilon > 0$ è una soglia arbitraria prossima allo zero.

In particolare, la mutua informazione non dice soltanto se le variabili sono dipendenti ma quantifica anche l'entità della dipendenza (un valore elevato di I(A,B) indica una forte dipendenza). Ovviamente il problema della computazione delle probabilità condizionate può essere affrontato come accentato sopra.

La cross entropia è utilizzata anche nei metodi score based: in tale proposito bisogna osservare che la definizione precedente può risultare errata, specie se non si considera la cardinalità delle variabili. Per esempio, se sia Y che Z fossero possibili padri di X, ma Y presenta due valori e Z otto (rispettivamente uno e tre bit di informazione), ci si aspetta che Y sia meno informativa su X rispetto a Z. Però si riesce a stimare P(X|Y) in modo più robusto di P(X|Z) perché implica l'elicitazione di meno parametri. Una miglioria si avrebbe inserendo l'entropia in un'opportuna funzione di score. [CHE97]

3.3 METODOLOGIE DI VALUTAZIONE

Una volta che l'algoritmo di learning (sia della struttura che dei parametri) ha prodotto un risultato come verificarne l'efficacia? Una metodologia consiste nel testare l'algoritmo su una rete nota, definita *gold-standard network*, da cui generare un database D di campioni (in genere con metodi di campionamento). La learning accuracy, cioè la differenza fra *learned* e *gold* network, viene stimata:

⇒ per il parameter learning

- *mean square errore* (errore quadratico medio);
- *cross entropy*: sia p(U) la distribuzione di probabilità congiunta della gold-standard e q(U) quella della learned network. La cross entropy H(p,q) è definita come detto prima

$$H(p,q) = \sum_{X_1,\ldots,X_n} p(X_i,\ldots,X_n) \log \frac{p(X_i,\ldots,X_n)}{q(X_i,\ldots,X_n)}$$

$$H(p,q) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_r} p(X_i = k, Pa_i = j) \log \frac{p(X_i = k, Pa_i = j)}{q(X_i = k, Pa_i = j)}$$

⇒ per lo structural learning

- *structural difference*: rappresenta il grado con il quale la learned network rappresenta le relazioni causali presenti in D rispetto alla rete gold. Definita la differenza simmetrica δ_i fra i padri di X_i in due differenti reti P *(gold)* e Q *(learned)* $\delta_i = \left| \left(\boldsymbol{Pa}_i^{\boldsymbol{Q}} \cup \boldsymbol{Pa}_i^{\boldsymbol{P}} \right) \setminus \left(\boldsymbol{Pa}_i^{\boldsymbol{Q}} \cap \boldsymbol{Pa}_i^{\boldsymbol{P}} \right) \right|$, la diffrenza strutturale $\delta = \sum_{i=1}^{n} \delta_i$ misura il numero di archi in cui le reti P e Q differiscono, contando due volte gli archi che sono stati invertiti nel passaggio da P a Q;
- semplicemente confrontando la gold e la learned network e verificando il numero di archi non rilevati (missing edge), in più (extra edge) e inverititi.

[HEC95] [BUN96] [PAP]

3.4 GLI ALGORITMI DI STRUCTURAL LEARNING

3.4.1 GLI ALGORITMI BAYESIANI

3.4.1.1 L'ALGORITMO BAYESIANO

L'algoritmo bayesiano risolve il problema dello "Structural Learning from data" determinando la struttura **m** che massimizza la probabilità p(**M** = **m** | D), dove **M** = {$\mathbf{m}_1, \ldots, \mathbf{m}_{G(n)}$} è l'insieme dei modelli che si ritiene contenere il *true model* di un dominio **X** e D è il database di osservazioni campionarie. Dati due modelli $\mathbf{m}_i$ e $\mathbf{m}_j$ candidati a rappresentare il dominio **X** dato D, si sceglie $\mathbf{m}_i$ se $p(\mathbf{m}_i|D) > p(\mathbf{m}_j|D)$. Dal teorema di Bayes

$$p(\mathbf{m}_i)p(D|\,\mathbf{m}_i) > p(\mathbf{m}_j)p(D|\,\mathbf{m}_j)$$

$$\frac{p(\mathbf{m}_i \mid D)}{p(\mathbf{m}_j \mid D)} = \frac{p(\mathbf{m}_i)}{p(\mathbf{m}_j)} \frac{p(D \mid \mathbf{m}_i)}{p(D \mid \mathbf{m}_j)} \qquad (3.3)$$

ed il rapporto delle evidenze $\frac{p(D \mid \mathbf{m}_i)}{p(D \mid \mathbf{m}_j)}$ in (3.3) è chiamato *fattore di Bayes*.

E' intuitivo scegliere come funzione di score p(D|**m**); nel capitolo precedente, si è visto che secondo opportune ipotesi (il campione D è completo, i casi nel database sono indipendenti, la distribuzione a priori dei parametri è multinomiale e la coniugata a priori è una distribuzione di Dirichlet) si ottiene

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

$$\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} \qquad (3.4)$$

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

m rappresenta la struttura di rete candidata a rappresentare il dominio $\mathbf{X} = \{X_1,\ldots,X_n\}$, di *n* variabili/nodi, dato il database di campioni D. Una variabile X_i presenta r_i stati mentre q_i sono le configurazioni in D dell'insieme dei padri di X_i.

$\Gamma(x)$ è la funzione gamma di Eulero; α_{ijk} sono gli iperparametri della distribuzione di Dirichlet e assumono un valore elevato quanto maggiore è la conoscenza a priori sulla struttura **m** e sulla distribuzione dei parametri.

N_{ijk} sono le occorrenze in D dei record aventi $\underbrace{\{X_i = x_i^k}_{\text{stato k – esimo di } X_i}, \quad \underbrace{Pa(X_i) = Pa_i^j\}}_{\text{configurazione j – esima dei padri di } X_i}$

mentre per la stima della probabilità

$$p\left(x_i^k \mid pa_i^j, D, \mathbf{m}\right) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \qquad (3.5)$$

La funzione SCORE(**m**) usata è il logaritmo[54] di p(D|**m**)

$$\begin{aligned} \text{SCORE}(\mathbf{m}) &= \text{logp}(\mathbf{m} \mid D) \\ &= \text{logp}(\mathbf{m}) + \text{logp}(D \mid \mathbf{m}) - \text{logp}(D) \\ &\cong \text{logp}(D \mid \mathbf{m}) \end{aligned} \qquad (3.6)$$

L'approssimazione compiuta è ammissibile in quanto log(p(D)) è una costante come anche il prior sul modello, log(p(**m**)), è costante nell'ipotesi che ogni modello sia equiprobabile (completa ignoranza a priori sul dominio). Da queste considerazioni, il criterio statistico di riferimento è il Maximum Likelihood; invece, nel caso in cui log(p(**m**)) non sia trascurabile si fa un implicito riferimento al criterio Maximum a Posteriori (MAP).

Dall' equazioni (3.4) e (3.6) è immediato ricavare che

$$\text{SCORE}(\mathbf{m}) = \sum_{i=1}^{n}\sum_{j=1}^{q_i} \log\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log\frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \qquad (3.7)$$

[54] Il logaritmo ha il pregio di essere più agevole per il computo matematico specie per il confronto fra valori prossimi allo zero, come accade con eventi poco probabili, e perché riduce la computazione in somme o differenze anziché prodotti e divisioni.

Per poter avere una formula computabile bisogna assegnare dei valori agli α_{ijk}. Questi iperparametri codificano la conoscenza a priori, "user confidence", che l'utente ha sulla distribuzione dei parametri p(**θ**|**m**) e, quindi, sul modello **m**. Una possibilità è esprimere l'equiprobabilità di ogni istanza dello spazio delle probabilità con la relazione $\alpha_{ijk} = \frac{\alpha}{q_i r_i}$; $\alpha_i = \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \alpha_{ijk} = \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{\alpha}{q_i r_i} = \alpha$ per cui resta da assegnare un unico iperparametro α - *dimensione di un campione equivalente*. Nelle metriche bayesiane rientrano delle versioni "penalizzate", indicate in letteratura come *Penalized Maximum Likelihood*, quali l'AIC e il BIC[55].

$$\log p(D \mid \mathbf{m}, \boldsymbol{\theta}_m) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - pen(N) \dim(\mathbf{m})$$

Penalized Maximum Likelihood - N è la dimensione del database (numero di record), dim(**m**) è una misura della complessità (numero di legami, ad esempio) del modello **m**.
AIC - Akaike Information Criterion: pen(N) = 1;
BIC - Bayeisan Information Criterion: pen(N) = ½ logN.

Determinata la scoring function bisogna scegliere la metodologia di ricerca; l'approccio bayesiano completo del model averaging è improponibile. In effetti gli statistici hanno delineato due possibili approssimazioni

- *model selection*: selezionare un "buon" modello fra tutti i possibili ed usarlo come se fosse quello corretto;
- *selective model averaging*: selezionare un certo numero di buoni modelli.

Questi approcci fanno emergere alcuni interrogativi: si riesce a fornire risultati accurati? Se sì, come ricercare il miglior modello? E come indicare se un modello sia "buono" o meno? La questione dell'accuratezza e della ricerca, in teoria, sono difficili da risolvere. Ciononostante, ricercatori quali Cooper, Herskovits, Heckerman, Spirtes, Meek, Chickering hanno mostrato, sperimentalmente, che la selezione di un singolo modello, usando una greedy search (ricerca esaustiva), fornisce predizione accurate.

[55] Estensione dell'approssimazione gaussiana accennata nel Capitolo 2.

Il selective model averaging, invece, deve essere applicato con metodi di campionamento di Monte Carlo e risulta talvolta più efficiente e con predizioni migliori.

Nel seguito si farà riferimento al "model selection": l'ipotesi su cui si fonda la scelta del *best model* è che il massimo della distribuzione $p(\mathbf{m} \mid D)$ sia localizzato nell'intorno di un particolare modello $\hat{\mathbf{m}}$. Per selezionare $\hat{\mathbf{m}}$ si introduce una funzione il cui valore sia tanto più alto quanto il generico modello **m** è prossimo a $\hat{\mathbf{m}}$. A tale scopo, un'eventuale metodologia di ricerca è l'"hill-climbing"; scelta una struttura *S* di partenza (un grafo privo di archi, che codifica la completa ignoranza sulle relazioni fra le variabili, un grafo aciclico costruito inserendo archi in modo casuale oppure una rete che rappresenti la conoscenza a priori), è possibile valutare il guadagno di *SCORE* che si ha per ogni possibile variazione elementare degli archi (aggiunta di un arco fra due nodi mutuamente indipendenti, eliminazione o inversione della direzione di un arco fra due nodi dipendenti) che non alteri l'aciclicità del grafo. Dopo la computazione di tutte le variazioni elementari possibili si effettua, se esiste, quella che apporterebbe un guadagno positivo maggiore. Il nuovo *SCORE* viene aggiornato e si reitera il procedimento. La procedura termina nel caso in cui nessuna modifica faccia aumentare lo *SCORE* oppure, in modo da terminare e velocizzare l'algoritmo nonostante la ricerca esaustiva, se è stato raggiunto un limite massimo, definibile a priori, di iterazioni.

La computazione della metrica è resa agevole dalla proprietà di scomposizione delle scoring function. La variazione di un solo arco della struttura influirà al più su due addendi della metrica, relativi ai nodi sorgente e destinazione dell'arco modificato (in particolare ciò accade soltanto se un arco della struttura viene invertito, negli altri casi è sufficiente la stima della variazione relativa del solo nodo destinazione in quanto l'insieme dei padri del sorgente non varia).

Il pregio di questo algoritmo è nell'esprimere la priori knowledge con il grafo di partenza e con il parametro α il quale sarà tanto maggiore quanto più informazioni a priori sono disponibili. La natura greedy dell'algoritmo, invece, rappresenta un limite sull'efficienza. [HEC94] [HEC95] [KRA98] [PAP]

1. Inizializzazione del grafo: si sceglie una struttura di partenza fra una random structure (grafo casuale), o una priori network (rete che codifica la conoscenza a priori), o un grafo vuoto senza legami. In base alla conoscenza a priori espressa in questa fase, fissare un valore per l'iperparametro α.
2. Se non è stato ancora raggiunto il massimo numero di iterazioni
   - ☑ Calcolare la score relativa ad ogni possibile modifica elementare sugli archi (inversione, rimozione, aggiunta).
   - ☑ Se esiste un guadagno positivo maggiore degli altri effettuare la modifica aggiornando la struttura del grafo e la score totale ed iterare il passo 2; altrimenti step 3.
3. Ritorna la struttura corrente.

3.4.1.2 L'ALGORITMO K2

Cooper e Herskovitz, in "A Bayesian Method for the Induction of Probabilistic Networks from Data" [COO92], hanno concepito un metodo bayesiano, quindi non basato sui test di indipendenza, per l'apprendimento: l'algoritmo K2. Questa denominazione deriva da una prima versione, denominata Kutato, della quale si è conservata l'inziale "K". Dato un insieme di assunzioni:

1. le variabile sono discrete (multinomiali);
2. i casi del database occorrono indipendentemente;
3. non ci sono missing values;
4. non si conosce la probabilità numerica da assegnare alla struttura;

Cooper e Herskovitz hanno determinato la metrica K2 in virtù del seguente asserto:

TEOREMA. Si consideri un insieme Z di *n* variabili discrete. Ogni variabile $X_i \in Z$ ha r_i possibili valori $(v_{i_1},...,v_{i_{r_i}})$. Sia D un database di m casi completi dove ogni caso contiene il valore da attribuire ad ogni variabile in Z. Sia B_s la struttura

di una rete bayesiana contenente proprio le variabili in Z, ogni variabile X_i in B_s ha un insieme di padri π_i. Indichiamo con w_{ij} la j-esima unica istanza di π_i in D e con q_i il totale delle istanze π_i. N_{ijk} rappresenti il numero di casi in D nel quale X_i è istanziato con valore v_{i_k}, mentre π_i ha valore da w_{ij}, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Allora

$$p(B_s, D) = p(B_s)\prod_{i=1}^{n} g(i, \pi_i)$$

dove $g(i, \pi_i)$ è fornita dalla seguente

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_t - 1)!} \prod_{k=1}^{r_t} (N_{ijk})!$$

Una volta che la struttura della rete è nota, l'elicitazione delle probabilità condizionate (parametri) è resa dalla seguente[56]

$$\theta_{ijk} = p(X_i = v_i^k \mid \pi_i = w_{ij}) = E[\theta_{ijk} \mid \mathbf{m}, D, \xi] = \frac{N_{ijk} + 1}{N_{ij} + r_i}$$

In effetti Cooper e Herskovits hanno ricavato anche l'espressione generale (3.4) illustrata per l'algoritmo bayesiano: per la metrica K2, in particolare, assumono $\alpha_{ijk} = 1$, ovvero completa ignoranza a priori sulla distribuzione di probabilità.

La procedura K2 si differenzia anche per la fase di inizializzazione. Mentre nell'approccio bayesiano vi è la possibilità di un grafo iniziale di partenza che codifichi la conoscenza a priori, in questo algoritmo, invece, è richiesto l'ordinamento topologico (prima i padri e poi i figli) dei nodi, in modo da ridurre la cardinalità dello spazio di ricerca dei modelli. Infatti, come accennato in precedenza, il numero delle possibili strutture cresce esponenzialmente in funzione del numero di variabili del dominio, cosicché uno screening su un'enumerazione esaustiva di tutti i modelli ammissibili è inefficiente sia dal punto di vista computazionale che in termini di tempo. Ovviamente è desiderabile che si scelga un ordinamento tale da permettere di rappresentare graficamente quante più indipendenze condizionate espresse dalla distribuzione di probabilità $p(\mathbf{X}|D)$, ovvero che descriva il dominio di interesse e le relazioni presenti nei dati.

Nonostante il requisito dell'ordinamento, il numero di modelli da considerare resta comunque elevato al crescere della cardinalità di $\mathbf{X} = \{X_1,\ldots,X_n\}$ in quanto

[56] E[X] valore atteso della variabile aleatoria X.

la distribuzione di probabilità congiunta $P(X_1,X_2,\ldots,X_n)$ può essere riscritta in modo diverso, a seconda dei legami, fissata una qualsiasi delle *n*! configurazioni.
Il termine $p(B_s)$, introdotto nella metrica K2, permette di introdurre la conoscenza a priori sulla struttura: se un esperto suggerisce l'esistenza di uno specifico arco è opportuno assegnare una maggiore probabilità alle strutture che soddisfano questo vincolo. In particolare, ci sono anche situazioni nelle quali alcuni modelli sono chiaramente da preferire; nel campo delle applicazioni diagnostiche, è evidente che non bisogna considerare strutture per le quali le malattie siano effetto dei sintomi! Se non è disponibile nessuna informazione a priori, la distribuzione di probabilità di $p(B_s)$ è *uniforme*, questa ulteriore ipotesi semplificativa, assunta nel seguito del discorso, consente di ignorare (si elide) il valore $p(B_s)$ nella scoring function quando si confrontano due modelli.
Un esempio permette di chiarire il discorso condotto finora. Si supponga di disporre del database in figura, in cui X_1 rappresenta un guasto in un sistema, mentre X_2 e X_3 due possibili conseguenze.

Case	Variable values for each case		
	x_1	x_2	x_3
1	*present*	*absent*	*absent*
2	*present*	*present*	*present*
3	*absent*	*absent*	*present*
4	*present*	*present*	*present*
5	*absent*	*absent*	*absent*
6	*absent*	*present*	*present*
7	*present*	*present*	*present*
8	*absent*	*absent*	*absent*
9	*present*	*present*	*present*
10	*absent*	*absent*	*absent*

Quale relazione di dipendenza qualitativa insiste fra le variabili? Ad esempio, X_1 e X_3 si influenzano l'una con l'altra (figura 3) o soltanto attraverso X_2 (figura 4)? Quale è la probabilità che x_3 sia *present* sapendo che anche x_1 è present? Le risposte dipendono da molti fattori: il modello usato, la conoscenza a priori sui dati e quali relazioni fra le variabili siano note.

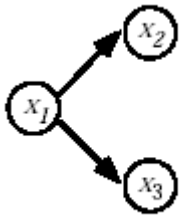


Figura 21 - B_{S2}

Figura 22 - B_{S1}

Supponiamo di generare i campioni, con metodi di campionamento Monte Carlo, in base alle probabilità espresse di seguito relative alla struttura B_{S1}, ovvero ogni stato di x_3 è condizionalmente indipendente dal valore x_1 noto x_2.

$P(x_1 = present)$	$= 0.6$	$P(x_1 = absent)$	$= 0.4$
$P(x_2 = present \mid x_1 = present)$	$= 0.8$	$P(x_2 = absent \mid x_1 = present)$	$= 0.2$
$P(x_2 = present \mid x_1 = absent)$	$= 0.3$	$P(x_2 = absent \mid x_1 = absent)$	$= 0.7$
$P(x_3 = present \mid x_2 = present)$	$= 0.9$	$P(x_3 = absent \mid x_2 = present)$	$= 0.1$
$P(x_3 = present \mid x_2 = absent)$	$= 0.15$	$P(x_3 = absent \mid x_2 = absent)$	$= 0.85$

Applicando la metrica K2 si ha

$$p(B_{S1}, D) = P(B_{S1}) \frac{(2-1)!}{(10+2-1)!} 5!5! \frac{(2-1)!}{(5+2-1)!} 1!4! \frac{(2-1)!}{(5+2-1)!} 4!1! \frac{(2-1)!}{(5+2-1)!} 0!5! \frac{(2-1)!}{(5+2-1)!} 4!1! =$$
$$= p(B_{S1}) 2.23 \cdot 10^{-9}$$

invece $p(Bs_2,D) = p(Bs_2)2.2310^{-10}$. Se si assume $p(Bs_1) = p(Bs_2)$, l'apprendimento dai i dati segnala che Bs_1 è 10 volte più probabile rispetto a Bs_2: in questo caso il risultato non è tanto sorprendente poiché i campioni sono stati generati proprio da B_{S1}.

L'approccio illustrato è basato sul best model; è opportuno dire però che lo stesso Herskovits ha concepito un algoritmo K2-multiscore che fornisce i modelli più probabili corrispondenti ad un particolare ordinamento sui nodi.

In sintesi, l'algoritmo K2 è riassumbile come esposto nel riquadro seguente.

1. Scegliere un ordinamento topologico ottimale sui nodi.
2. Per ogni nodi X_i, con i = 1,…,*n* (*n* ≡ numero di nodi):
   - Costruire l'insieme dei predecessori *pred(X_i)* , secondo l'ordinamento, di X_i.
   - Per ogni nodo X_j presente in *pred(X_i)*, se massimizza la funzione di score allora considerare X_j come padre di X_i aggiungendolo all'insieme $Pa(X_i)$.
3. La rete appresa è rappresentata da **X** = {X_1, …, X_n } e **Pa** = { $Pa(X_i)$, …, $Pa(X_n)$}.

Il nucleo dell'approccio suggerito da Cooper e Herskovits è un **greedy search algorithm incrementale** che procede assumendo, all'inizio, che un nodo non abbia padri. Dopodiché, dato l'ordinamento $X_1<X_2<…X_i…<X_n$, l'insieme dei padri di ogni nodo X_i è determinato valutando se ogni variabile dell'insieme dei relativi predecessori, $pred(X_i) = \{X_1,…,X_{i-1}\}$, possa essere padre di X_i, ovvero quale $X_j \in pred(X_i)$ massimizza la metrica K2 espressa da $g(X_i,\pi_i)$. La fase di ricerca, per l'insieme dei padri di X_i, termina dopo avere esaminato tutte le variabili in $pred(X_i)$ o se è 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 $\{X_1,…,X_{i-1}\}$; d'altronde non ha senso considerare i nodi da X_{i+1} in poi in quanto, in base all'ordinamento, rappresentano dei figli. 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. Si evince, però, anche uno degli svantaggi di tale approccio in quanto non è possibile "ritornare indietro" dopo l'aggiunta di un arco come accade, invece, nell'algoritmo bayesiano.

Algoritmo K2

per ogni nodo X_i, con $i = 1, .. ,n$ trovare l'insieme di padri π_i di X_i come segue:

π_i = insieme vuoto

$P_{old} = g(i, \pi_i)$

NotDone = true

While NotDone do

Per ogni X_l che viene prima di X_i nell'ordine, con $X_l \notin \pi_i$, $g_l = g(X_i, \pi_i \ U \ \{ X_l \})$

$P_{new} = max_{Xl} \ g(X_i, \pi_i \ U \ \{ X_l \})$

Sia X_z la variabile che massimizza g_l precedente, allora

If Pnew > Pold then

Pold = Pnew

$\pi_i = \pi_i \ U \ \{ X_z \}$

Else NotDone = false;

end {while};

Il maggior svantaggio dell'algoritmo K2, nonostante fornisca risultati lusinghieri per lo Structural Learning, è però 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. Interessanti lavori sull'ordinamento sono "A permutation genetic algorithm for variabile ordering in learning Bayesian Networks from data" di W.H.Hsu, H.Guo, B.B. Perry, J,A.Stilson [HSU02], in cui è un algoritmo genetico a determinare l'ordinamento ottimale, e l'algoritmo CB illustrato fra breve. Un'alternativa è iniziare con una sequenza casuale e adoperare dei raffinamenti successivi sulla struttura con altri algoritmi o grazie alla user's knowledge.

3.4.1.2.1 Modifiche al K2: l'algoritmo CB

La procedura K2 illustrata può essere estesa anche al caso di missing values e hidden variables determinando, con metodi statistici, il valore dei dati non noti, ottenendo così un database completo da considerare come input per l'algoritmo

K2, oppure valutando le stime N_{ijk} per i record di D che sono completi (questo ultimo approccio è senza dubbio inefficiente nel caso di numerosi missing value).
M.Singh e M.Valtorta in "Construction of bayesian network structures from data: a brief survey and efficient algorithm" [SIN94] illustrano un approccio ibrido fra l'algoritmo K2 ed i metodi basati sui test di indipendenza condizionata (CI). Lo stesso Herskovitz, d'altronde, suggerisce l'uso della metrica K2 con un metodo CI-based per ovviare alla richiesta dell'ordinamento dei nodi. I test CI, identificata una possibile struttura, permettono di generare un ordinamento sui nodi: l'algoritmo implementato da Sigh e Valtorta, denominato CB (il nome è suggerito dalla due fasi dell'algoritmo - Conditional independence Bayesian learning), usa un algoritmo CI per generare un pdag (dag parziale: non tutti gli archi risultano orientati) da cui determinare l'ordinamento dei nodi da elaborare, in una seconda fase, secondo gli step fondamentali dell'algoritmo K2.
[COO92] [BOU94] [SIN94] [SUZ99]

3.4.1.3 MDL - BASED ALGORITHM: L'ALGORITMO K3

Il principio Minimum Description Length discende dal principio logico di William di Occam (frate e filosofo francescano del XIV secolo), noto anche come "Occam's razor", che postula << Entities should not be multiplied unnecessarily >> o, per gli appassionati dei latinismi, << Pluralitas non esta ponenda sine necessitate >>. L'immediata applicazione nel campo scientifico è quella di prediligire fra due teorie, o metodi, che conducano agli stessi risultati quello più semplice. Nello specifico, la metrica MDL - Minimum Description Length – è spesso applicata nel machine learning e si basa sull'omonimo principio: "il migliore modello per un insieme di dati è quello che minimizza la somma fra 1) la lunghezza di codifica[57] del modello, 2)la lunghezza di codifica della sorgente dato il modello". Una tipica applicazione sono gli algoritmi di compressione dei dati: il codificatore descrive la regola usata per generare la sequenza di simboli con cui comprimere l'informazione cosicché il decodificatore possa ricostruire la sequenza ricevuta in modo univoco. Generalizzando, sia *G* l'insieme delle

[57] Il termine "lunghezza di codifica" viene anche indicato come "description length" e si misura in bit.

possibili regole e A l'insieme dei possibili esempi (sequenze), $g \in G$ una regola e $x^n \in A$ una sequenza (con $n \geq 1$ ed intero); il principio MDL suggerisce che la g migliore è quella che minimizza la description length $L(g,x^n)$. Applicando il formalismo della compressione dei dati al caso dello Structural Learning di BN, le osservazioni del database rappresentano la sequenza x^n emessa da una qualche struttura $g \in G$, dove G rappresenta l'insieme dei modelli che codificano l'informazione **X**.

Nell'ambito dello Structural Learning delle reti bayesiane, l'approccio MDL predilige la learned network che minimizza la *total desription length* definita dalla

1) decription length dei campioni (sorgente);
2) description length di una struttura di rete (modello) pre-esistente (fornita da un esperto o generata in un processo di apprendimento precedente. I dati e la struttura pre-esistente sono assunti indipendenti l'uno dall'altro per cui le lunghezze di codifica sono elaborate separatamente).

La learned network rappresenterà un compromesso fra l'accuratezza rispetto ai dati, la vicinanza con un'eventuale struttura pre-esistente e la complessità (numero di legami) della struttura. Dati un dominio di n variabili **X**, una struttura di rete B, un database di N campioni D, la description length L(B,D) della struttura di rete B dato D è espressa dalla seguente relazione:

$$L(B,D) = \log p(B) + N \cdot H(B,D) - \frac{1}{2} k \log N$$

$$H(B,D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=}^{r_i} -\frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}} \qquad (3.8)$$

$$k = \sum_{i=1}^{n} q_i (r_i - 1)$$

B rappresenta la struttura di rete candidata a rappresentare il dominio $\mathbf{X} = \{X_1,\ldots,X_n\}$, di n variabili/nodi, dato il database di campioni D. Una variabile X_i presenta r_i stati mentre q_i sono tutte le configurazioni possibili dell'insieme dei padri di X_i.

N_{ijk} sono le occorrenze in D dei record aventi $\{X_i = x_i^k,\ \text{stato k – esimo di } X_i \quad Pa(X_i) = Pa_i^j\ \text{configurazione j – esima dei padri di } X_i\}$,

$N_{ij} = \sum_{j=1}^{r_i} N_{ijk}$ occorrenze dei record $\{X_i,\ Pa(X_i) = Pa_i^j\}$

A differenza del K2, si osservi che q_i indica il numero di tutte le possibili configurazioni (teoriche) di padri del nodo X_i mentre in K2 sono quelle osservate in D (empiriche). Il secondo termine della formula rappresenta l'entropia condizionale della struttura di rete B che descrive quanto bene la rete rappresenti i dati - "data description length". L'entropia, come è definita sopra, rappresenta una misura non negativa dell'incertezza ed è massima quando l'incertezza è elevata, zero quando vi è completa conoscenza; più informazione è disponibile e minore sarà l'entropia. Ad esempio, aggiungendo dei padri ad un nodo il termine relativo all'entropia diminuirà poiché la distribuzione di probabilità può essere descritta in modo più accurato (sono individuabili maggiori dipendenze). Nel terzo termine, il fattore *k* indica proprio il numero di probabilità che devono essere stimate dal database D per ottenere i parametri della rete ed indica la "network description length"[58], con cui quantificare la complessità della rete. Per essere più precisi, ogni probabilità stimata introduce un piccolo errore e il termine 1/2klogN rappresenta l'errore totale. Assumendo p(B) uniforme per tutte le strutture di rete, il seguente grafico rappresenta l'interazione fra i termini della MDL score. Sull'asse delle ascisse è riportato *k* che è proporzionale al numero di archi presenti nella struttura; la misura MDL è sull'asse delle ordinate.

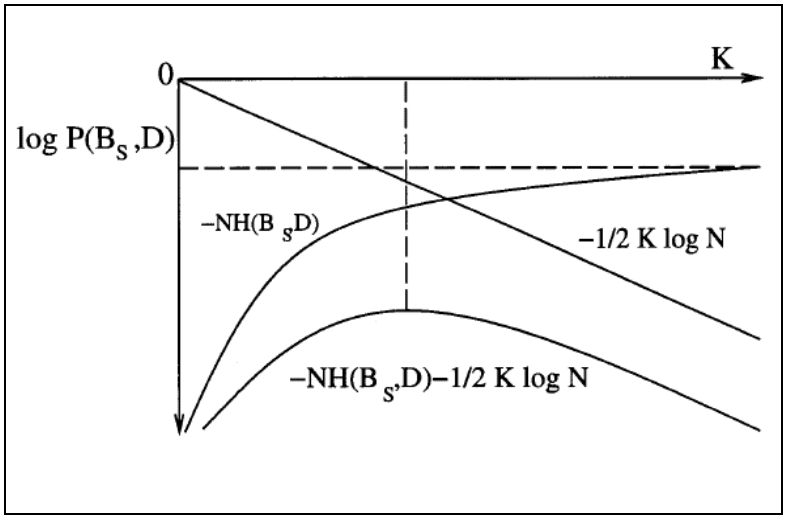


Figura 23 – Relazioni fra i vari termini della misura MDL

[58] In "A distributed learning algorithm for Bayesian inference networks" [LAM02] è riportata una formula più accurate per questo termine, in cui $s_i = N_{ij}$ e d è il numero di bit richiesti per rappresentare un valore numerico.

$$L_{network}(X_i, Pa(X_i)) = |Pa(X_i)| \log_2(n) + d(s_i - 1) \prod_{j \in pa(X_i)} s_j$$

L'equiprobabilità della priori è rappresentata da una retta costante. Con un numero maggiore di archi, una struttura di rete sarà più accuratamente descritta perché sono indicate più dipendenze condizionate per cui il termine relativo all'entropia diminuisce. D'altro canto, il termine 1/2klogN aumenta con l'aggiunta di archi: la struttura di rete migliore bilancerà entrambi i contributi.

In sintesi, minimizzando la metrica MDL si preferisce una struttura di rete con meno archi rispetto ad una con più archi a meno che l'entropia condizionale del modello più complesso sia minore di quello più semplice.

In "Probabilistic Network construction using the Minimum Description Length Principle" di R.R. Bouckaert [BOU94], è dimostrato il seguente asserto che evidenzia l'importanza dell'approccio bayesiano.

TEOREMA. Sia **X** un insieme di variabili, B una struttura di rete e D un database relativo a **X** con N casi tali che *tutte le istanze degli insiemi di padri di B occorrano nel database*[59]. Sia p(B,D) la misura bayesiana di B dato D, secondo l'espressione fornita da Cooper e Herskovits, e sia L(B,D) la misura MDL di B dato D. Allora L(B,D) = log p(B,D) + Costante.

Una differenza sostanziale si presenta nel momento in cui le osservazioni in D non contengono tutte le possibili configurazioni dei padri: in tal caso, la misura MDL assegna un peso maggiore al termine relativo alla stima dei parametri rispetto alla stima Bayesiana; ne segue che l'MDL, avendo la funzione di score una soglia maggiore, tende ad apprendere una rete con meno archi rispetto all'approccio Bayesiano. [BOU94][LAM98][SUZ99][LAM02]

In letteratura l'approccio MDL è spesso usato con altre tecniche quali algoritmi di Evolutionary Programming - metodo MDLEP. In " Bayesian Network refinement via machine learning approach" di W. Lam [LAM98], il principio MDL è lo strumento impiegato per raffinare la struttura di una BN, eventualmente acquisendo solo una parte dei dati e non tutti quelli disponibili. Dei miglioramenti e un diverso approccio sono presenti in "Learning Bayesian Belief Networks based on the MDL principle: an efficient algortihm using the branch and bound technique" di J. Suzuki [SUZ99], in cui l'autore illustra una ricerca diversa dalla greedy search, branch & bound, con risultati incoraggianti relativamente al local maxima. Altre applicazioni usano la metrica MDL, al posto della BD o K2,

[59] Si ricordi la differenza, su menzionata, fra il valore di q_i per la metica K2 e la MDL score.

concependo un approccio bayesiano basato su MDL: un esempio è l'algoritmo K3, versione modificata del K2, per il quale vanno fatte le stesse ipotesi di lavoro. In particolare, per quanto esposto sulla metrica MDL, il K3 rispetto al K2 termina prima la fase di ricerca ed è, quindi, meno accurato. Come per la metrica K2, il principio MDL fornisce un *natural stopping criterion* per la ricerca euristica.

Algoritmo K3

Siano le variabili del dominio $\boldsymbol{X}$ *ordinate,* $X_1 < \ldots < X_n$

for i =1 , ... , n do $\pi_{i,new} = \pi_{i,old} = 0$

for i = 2, ..., n do

repeat

$\pi_{i,old} = \pi_{i,new}$

sia B_S *la struttura definite da* $\pi_{1,old}, \ldots, \pi_{n,old}$

$z = argmax_y \{L(B_{Sy},D) - L(B_S,D)\}$ *con* $y \in \{X_1, \ldots, X_{i-1}\} \setminus \pi_{i,old}$ *dove*

B_{Sy} *è* B_S *con* $\pi_i = \pi_{i,old} \cup \{y\}$

If $(L(B_{Sz},D) - L(B_S,D)) > 0$ *then* $\pi_{i,new} = \pi_{i,old} \cup \{z\}$

Until $\pi_{i,new} = \pi_{i,old}$ *or* $|\pi_{i,new}| = i-1$

Risultato: B_S *è definita da* $\{(X_1, \pi_{1,new}), \ldots, (X_n, \pi_{n,new})\}$.

3.4.2 I METODI CONSTRAINT BASED

I metodi constraint-based o dependance-based deducono, con opportuni test statistici, le relazioni di indipendenza condizionata, implicite nelle osservazioni campionarie, con cui modellare i leagami della struttura di una BN. Alla base di questi algoritmi vi è, comunque, il formalismo espressivo della teoria dei grafi. Di seguito si richiamano alcuni concetti per rendere più comprensibile la descrizione degli algoritmi.

3.4.2.1 PREMESSA: L'INDIPENDENZA NEI GRAFI

Dalla teoria della probabilità, due variabili casuali X e Y sono indipendenti - *I(X,Y)* - se p(X,Y)=p(X)p(Y); introdotta un'ulteriore variabile Z, I(X,Y|Z) denota che X è condizionalmente indipendente da Y dato Z, ovvero p(X,Y|Z) = p(X|Z)p(Y|Z).

Sia $\mathbf{X} = \{X_1,\ldots,X_n\}$ il dominio associato al DAG G. La distribuzione di probabilità congiunta p(**X**) ammette la seguente scomposizione:

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_i \mid Pa(X_i)) \qquad (3.9)$$

in cui $pa(X_i)$ denota una configurazione di valori per l'insieme dei padri di X_i. La (3.9) assume differenti espressioni in riferimento alle relazioni di indipendenza codificate dal DAG. Seguono alcuni semplici esempi che permettono di comprendere meglio la codifica dell'indipendenza nei modelli grafici quali le BN.

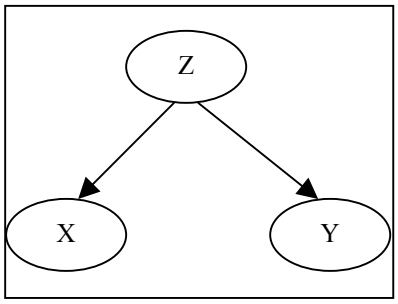


Figura 24 – X e Y indipendenti noto Z

In riferimento alla figura precedente, sia Z una malattia quale il morbillo e le variabili X e Y invece eventuali sintomatologie, "puntini rossi" e "Koplik's spot". Se viene diagnosticato il morbillo, sarà nota la probabilità di entrambi i sintomi, per cui la conferma di un sintomo non altererà l'occorrenza dell'altro.

La decomposizione della distribuzione di probabilità congiunta per X,Y,Z è $p(X,Y,Z) = p(X,Y|Z)p(Z) = p(X|Z)p(Y|Z)\ p(Z)$. Uno scenario ed una fattorizzazione differente, $p(X,Y,Z) = p(Z|X,Y)p(X)p(Y)$, invece, sono codificati dalla BN successiva in cui X e Y sono marginalmente indipendenti ma dipendenti una volta che risulti noto Z.

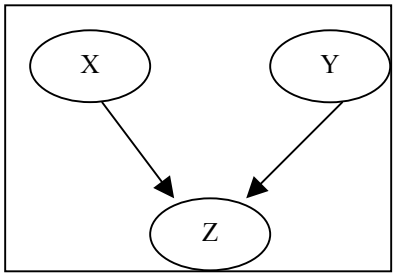


Figura 25 – X e Y marginalmente indipendenti, condizionalmente dipendenti da Z

Ad esempio, siano "rain" (X) e "sprinkler on"(Y) possibili cause dell'evento "prato bagnato"(Z). Prima che venga fatta qualsiasi osservazione sul prato, le probabilità di rain e sprinkler sono indipendenti; una volta che si osserva se il prato sia bagnato o meno, l'osservazione su X ("sta piovendo?") influenzerà la probabilità che lo sprinkler sia "on" (sono cause alternative).

Un ultimo esempio completa la casistica. La patologia X ("sindrome di Kawasaki") è una causa di Z ("ischemia del miocardio"), che, a sua volta, ha un sintomo Y ("dolore al torace").

Se da test aggiuntivi, si diagnostica la presenza dell'ischemia del miocardio, l'osservazione del dolore al torace non influenza la malattia di Kawasaki. Cioè, X e Y sono condizionalmente indipendenti dato Z.

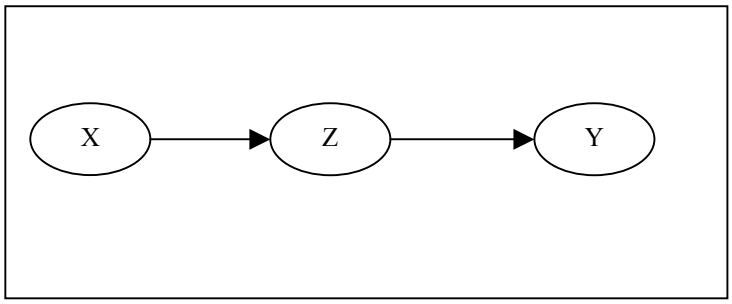


Figura 26 - X e Y sono condizionalmente indipendenti dato Z

La distribuzione di probabilità fattorizza come p(X,Y,Z) = p(Y|Z)p(Z|X)p(X).
A questo punto, è opportuno fissare alcune definizioni che saranno ricorrenti nel seguito del discorso.

Definizione. Due nodi sono definiti **adiacenti** (adjacent) o vicini (neighbour) se esiste un arco che li unisce. Un nodo senza padri è detto radice.

Definizione. Per qualsiasi coppia di nodi X e Y, un percorso $P = \langle a_1, a_2, ..., a_k \rangle$ – path o **adjacent path** – fra $a_1=X$ e $a_k=Y$ è una sequenza continua di archi, di cui si ignora la direzione, che collegano X e Y. Per distinguere questa situazione da quella in cui ha importanza considerare la direzione, ci riferiremo al primo caso con il nome di adjacency paths o chain, applicabili a grafi diretti, parzialmente orientati o non orientati, al secondo come **directed path**.

Definizione. Per qualsiasi nodo in un percorso di adiacenze, se due archi si incontrano su uno stesso nodo V, ad esempio X→V←Y o X-V-Y, diremo che V è un **collider** del percorso, altrimenti è detto non collider. Il concetto di collider è precipuo per un percorso, ossia un nodo può risultare collider in un path ma non esserlo per un un altro.

Data una struttura di rete S, gli asserti di indipendenza condizionata possono essere letti da un modello grafo usando il concetto di **direction dependent separation** o **d-separation**, dovuto a Pearl, (d[60] sta per directional). Il criterio *d*-separation è usato per decidere, dato un grafo causale, se una collezione X di variabili è indipendente da un'altra Y, dato un terzo insieme Z. L'idea è di

[60] In letteratura, la "d" è indicata è anche l'iniziale di "dependence", soprattutto per identificare la d-separation in grafi non completamente orientati.

associare "dipendenza" e "connessione" (esistenza di un percorso), "indipendenza" e "assenza di connessione" - "separation".

Definizione. Sia N un insieme di nodi per un DAG *G*. Per qualsiasi coppia di nodi *X,Y*, con $X \neq Y$, dato un sottoinsieme $C \subseteq N\backslash\{X,Y\}$ *(*"evidence"*)*, diremo che "*X* e *Y* sono **d-separated** da *C* in *G* se e solo se **non** esiste un *adjacency path P* fra *X* e *Y*, per cui

(i) Ogni collider in $P \in C$ o ha discendenti in *C*;

(ii) Nessun altro nodo del path $P \in C$.

C è chiamato **cut-set**. In caso contrario, X e Y sono **d-connected** da C. La definizione di d-separation di due nodi può essere facilmente estesa alla d-separation fra insieme di nodi.

I primi studi sulla d-separation sono stati condotti da Judea Pearl, Dan Geiger e Thomas Verma, nell'ambito di un lavoro per la memorizzazzione e gestione di informazioni in condizioni di incertezza con *artificial intelligent agent*. Dai loro studi hanno intuito l'efficacia dei directed acyclic graphs per codificare le relazioni di indipendenza in condizioni di incetezza. Successivamente, nei primi anni '90, Peter Spirtes, Clark Glymour e Richard Scheines, studiando *l'inferenza causale* al Philosopy Department alla Carnegie Mellon University, hanno usufruito del lavoro di Pearl e colleghi per esaminare e scoprire strutture causali nelle scienze del comportamento. In particolare, è stato provato da J.Pearl e D.Geiger in "Logical and algorithmic properties of conditional independence" (Technical Report TR-97, Cognitive Systems Laboratori, UCLA, 1988) che il concetto di d - separation può rilevare tutte le relazioni di indipendenza condizionale codificate in una rete bayesiana. In sintesi, *la d-separation è "la" regola per individuare le indipendenze in una Bayesian Network*.

Gli asserti di Markov rappresentano una delle alternative per riconoscere l'indipendenza:

- *directed local Markov property* - "qualsiasi variabile è condizionalmente indipendente dai suoi non discendenti, dati i padri"
- *global directed Markov property* - "se due insiemi di variabili A e B sono condizionalmente indipendenti dato un terzo C, allora C separa A e B nel grafo G".

Il Lauritzen ha dimostrato l'equivalenza fra la d-separation e la proprietà globale di Markov. Geiger e Pearl[61] hanno dimostrato che "Per ogni dag G, esiste una distribuzione di probabilità P tale che per ogni tripla X,Y,Z di insiemi disgiunti di variabili X e Y sono d-separated da Z se e solo se I(X,Y|Z) – indipendenza di X e Y dato Z".

Se due variabili sono *d-separated* rispetto ad un insieme di variabili Z in un directed graph G, allora esse sono condizionalemtnte indipendenti da Z in tutte le distribuzioni di probabilità codificate da G. Cerchiamo di comprendere meglio questa affermazione: due variabili X e Y sono condizionalmente indipendenti dato Z se la conoscenza su X non fornisce ulteriori informazioni su Y una volta noto il valore di Z (si ricordino gli esempi illustrati prima).

Una semplice analogia permette di chiarire il concetto di d-separation. Una Bayesian network è paragonabile ad una rete di canali idraulici: ogni nodo è una **valvola** che può presentarsi negli stati "**attiva**" (open) o "**inattiva**" (closed), le valvole sono connesse dai canali/**archi**.

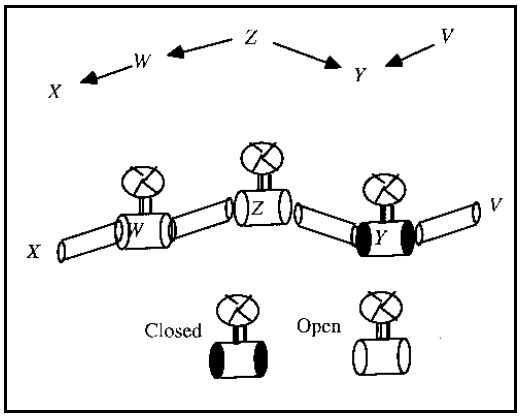


Figura 27- Analogia fra il flusso causale e quello di un fluido

Il flusso informativo (il fluido) passa attraverso una valvola attiva ma è bloccato da una valvola inattiva. Se tutte le valvole su un percorso di adiacenze fra due nodi (X,Y) sono attive, c'è flusso informativo fra X e Y, diremo che il path è **open**.

[61] Geiger D., Pearl J. 1988. On the logic of causal models. Proc. 4tyh Workshop on Uncertainity in AI, St. Paul, Minn, 136-147.

Se qualsiasi valvola nel percorso è inattiva, il path è **closed**: un nodo collider, quindi, è assimilabile ad valvola inattiva; viceversa una valvola attiva è un non collider.

Inserire un nodo in un condition set equivale ad alterare lo stato della valvola stessa e di quelle ad essa collegate (nello specifico l'alterazione influenza gli antenati di un nodo). X e Y sono d-separated da un condition set C quando tutti i percorsi da X a Y sono chiusi da C, "l'insieme delle valvole inattive". Viceversa, X e Y sono d-connected da C se tutti i percorsi fra X e Y sono attivi.

Riepilogando, un path fra X e Y è **attivo** o **open** se conduce informazione fra le due variabili, ovvero c'è dipendenza. Poiché in una rete Bayesiana due nodi X e Y possono essere connessi da numerosi percorsi, dei quali tutti, alcuni o nessuno può essere attivo, la d-separation implica che, fissato un condition set Z, tutti i percorsi che uniscono X e Y, siano **inattivi**. A questo punto, è opportuno evidenziare cosa renda un percorso inattivo o attivo. A tale scopo, consideriamo Z = {} (insieme vuoto) ed identifichiamo le varie situazioni per una terna A,B,C. La semantica causale permette di asserire che nel caso:

1) $A \rightarrow C \rightarrow B$: A è una causa *indiretta* di B;
2) $A \leftarrow C \leftarrow B$: B è una causa *indiretta* di A;
3) $A \leftarrow C \rightarrow B$: C è una causa comune di A e B;
4) $A \rightarrow C \leftarrow B$: A e B implicano l'effetto comune C.

Le configurazioni 1)-2)-3) evidenziano la dipendenza fra A e B, vi è un flusso di informazione, per cui tutti questi percorsi devono essere considerati attivi. Invece, nel caso 4) C è l'effetto (non la causa) comune fra A e B per cui non vi è nessuna connessione fra loro: il percorso è inattivo. Quando il conditioning set, Z, è vuoto sono attivi quei percorsi a cui corrisponde una connessione causale; la caratteristica comune ai tre percorsi, e che differenzia il quarto, è che nei primi tre C è un *non-collider* mentre nel quarto è un *collider* (i collider non trasmettono informazione, ovvero non c'è dipendenza). Quindi, *se il conditioning set è l'insieme vuoto i collider sono inattivi*. In tal caso, X e Y sono d-separated da Z se ci sono percorsi fra X e Ycon nodi collider.

Esempio 1

$$x \rightarrow r \rightarrow s \rightarrow t \leftarrow u \leftarrow v \rightarrow y$$

Questo grafo contiene un solo collider in t; il path *x-r-s-t* è aperto per cui *x* e *t* sono *d*-connected. Allo stesso modo *t* e *y*, come anche le coppie *u* e *y, t* e *v, t* e *u, x* e *s* sono d-connected. Invece, *x* e *y* non sono *d*-connected; non c'è nessun modo di tracciare un path da *x* a *y* senza attraversare il collider in *t*. Perciò si conclude che *x* e *y* sono *d*-separated, come anche *x* e *v, s* e *u, r* e *u*.
Esaminiamo il caso in cui Z sia un insieme non vuoto. Inserire un vertice nel conditon set Z significa alterare il suo stato che passa da attivo ad inattivo e viceversa. Sia Z = {C}; considerando i percorsi precedenti, C diventa inattivo per 1)-2)-3) ed attivo per il 4).

Secondo la semantica causale, infatti segue che

1) A → **C** → B: il path da A - B è bloccato da C;
2) A ← **C** ← B: il path da B - A è bloccato da C;
3) A ← **C** → B: nota la causa, gli effetti sono indipendenti;
4) A → **C** ← B: noto l'effetto le cuase sono dipendenti (mutuamente esclusive). Ad esempio, come spiega lo stesso Pearl, sapere che la batteria dell'auto è carica non fornisce informazioni sullo stato del carburante, ma se batteria è carica *sapendo anche* che l'auto non parte, molto probabilmente il serbatoio è vuoto. Le cause diventano dipendenti introducendo l'evidenza sull'effetto comune - collider.

dead battery → car won't start ← no gas

Esempio 2

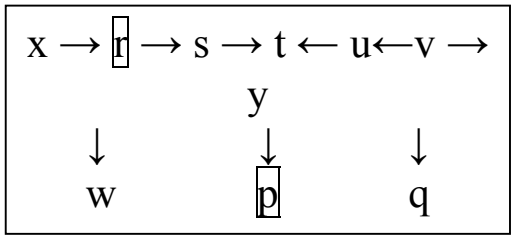
$$x \rightarrow \boldsymbol{r} \rightarrow s \rightarrow t \leftarrow u \leftarrow \boldsymbol{v} \rightarrow y$$

Quando si introduce l'evidenza, Z non vuoto, cambia lo stato delle variabili in Z e quindi anche di quelle rimanenti (indipendenti diventano dipendenti, dipendenti diventano indipendenti).

Sia infatti $Z = \{r, v\}$, x e y sono *d*-separated da Z. I percorsi *x-r-s*, *u-v-y* e *s-t-u* sono bloccati da Z. Le uniche coppie di nodi che restano connesse scegliendo come condition set Z sono *s* e *t, u* e *t*; si osserrvi che benchè *t* non sia in Z, il path *s-t-u* è comunque bloccato da Z, poiché *t* è un collider (Si confrontino le variazioni rispetto all'Esempio 1).

Esempio 3

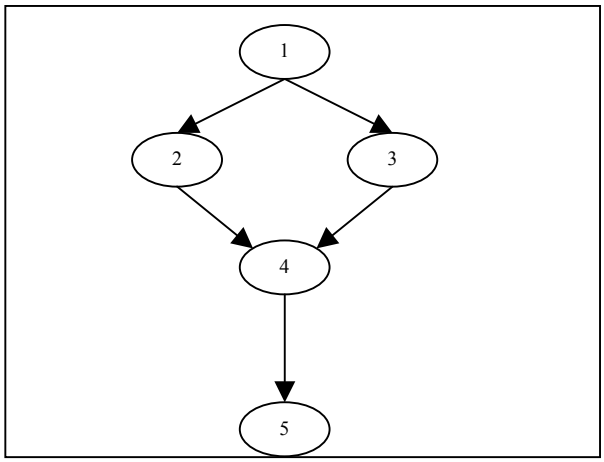
Se un collider appartiene ad un conditioning set Z, o ha un discendenti in Z, allora Z non blocca qualsiasi percorso che attraversa il collider.



Sia Z = {r, p}. s e y sono d-connected da Z, perché il collider in t ha il discendente (p) in Z, per cui il path s-t-u-v-y è open. x e u sono ancora d-separated da Z, in quanto r è in Z[62]. Infine una maggiore attenzione richiedono alcuni casi particolari: gli archi bi-directed. Per esempio, se si aggiunge al grafo precedente un arco bidirezionale fra x e t, allora y e x non saranno più d-separated da Z={r, p}, poiché il path x-t-u-v-y è d-connected - il collider t è aperto in virtù del discendente p in Z. Riportando il discorso alla definizione di indipendenza per le

[62] Per comprendere l'utilità della d-separation in un ambito più generale, si consideri la regressione di *y* su *p, r* e x: $y = c_1 p + c_2 r + c_3 x$. Consideriamo per quali coefficienti la regressione è nulla. In riferimento all'esempio 3, si conclude che $c_3 = 0$, perché *y* e *x* sono *d*-separated dati *p* e *r*; c_1 e c_2, invece, in generale non saranno nulli; infatti dal grafo l'insieme $Z=\{r, x\}$ non separa *y* da *p*, e $Z=\{p, x\}$ non separa *y* da *r*.

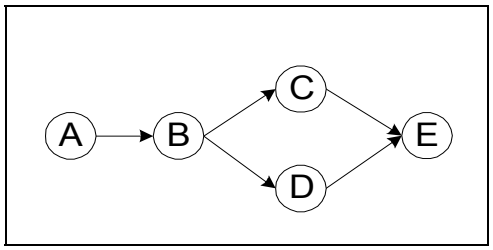
reti Bayesiane, un percorso fra i nodi X e Y è bloccato, data l'evidenza C, se X e Y sono condizionalmente indipendenti dato C. Consideriamo ora un'applicazione ad una Bayesian Network.



Siano X = 2, Y = 3 e sia Z = 1. Esaminiamo se X e Y sono d-separated da Z; ricordando la definizione di d-separation, dobbiamo considerare tutti i possibili percorsi fra i due nodi X e Y.

- $2 \leftarrow 1 \rightarrow 3$ - il nodo Z non è un collider;
- $2 \rightarrow 4 \leftarrow 3$ – il collider 4 e il suo discendente 5 sono al di fuori di Z. Di conseguenza, in virù della la d-separation, X e Y sono d-separated da Z = 1 (ovvero X e Y sono bloccati da Z).

Consideriamo un diverso insieme Z = {1,5}. Il discendente di 4, il nodo 5, è in Z ed apre un percorso fra X e Y; cioè l'evidenza sul nodo 5 influenza 2 e 3, Z apre il path $2 \rightarrow 4 \leftarrow 3$. Di conseguenza X e Y non sono d-separati da Z = {1,5}.
Ancora, nella figura seguente i nodi C-E-D formano un percorso di adiacenze che collega C e D: che E è un collider in C-E-D. Dato l'insieme vuoto {} C e D sono d-separated. Ritornando all'analogia delle valvole, inserendo il collider E nel cut-set apriremo il percorso fra C e D in quanto abbiamo alterato lo stato della valvola E. Invece, inserendo B nel cut-set chiuderemo i path A-B-C-E e A-B-D-E, cosicché A-E risultano d- separated da B.



Gli algoritmi dependance-based cercano di chiudere tutte le connesioni realizzabili fra due nodi X e Y per indagare, dal data set, se vi è un flusso informativo addizionale fra X e Y: in caso positivo, bisogna aggiungere l'arco X-Y. Per misurare il volume del flusso di informazione si usano i test di indipendenza che indicheranno se esiste un sottoinsieme di variabili, il condition set, che riduce ed eventualmente blocca il flusso informativo fra X e Y.

Un modello di indipendenza M è un insieme di relazioni di indipendenza. Sia M l'insieme di asserzioni di indipendenza espresse da una distribuzione di probabilità P. Un grafo G è un *dependence map*, **D-map**, di M se ogni relazione di dipendenza derivata da G è vera in M. Viceversa, un grafo G è un *independence map*, **I-map**, di M se ogni relazione di indipendenza espressa in G è vera in M. Un grafo Independence map è un *minimum I-map* di M se la rimozione di qualsiasi arco non lo rende un I-map. Se un grafo G è sia D-map che I-map di M, è definito *pefect-map*, **P-map** e la distribuzione P è definita **DAG-Isomorph** di G ed in tal caso la distribuzione P e il grafo G sono *faithful* l'una con l'altro. Difatti non è detto che una rete bayesiana rappresenti tutte le indipendenze condizionate della distribuzione di probabilità P.

Un dataset D è *DAG-faithful* se il modello probabilistico ad esso associato (cioè da cui si suppone che siano estratti i campioni) è DAG structured.

Definizioni. $paths_G(X,Y)$ è l'insieme di tutti gli adjacency path da X a Y in un grafo G. $open_G(X,Y|C)$ è il sottoinsieme di $paths_G(X,Y)$ che sono "open" fissato il cut-set C. Un modello DAG-Faithful G è **Monotone DAG-faithful** se e solo se per tutti i nodi X, Y in G se $open_G(X, Y \mid C') \subseteq open_G(X, Y \mid C)$ allora risulta anche $I(X,Y|C') \leq I(X,Y|C)$.

3.4.2.2 L'ALGORITMO PC

L'algoritmo PC, dovuto a Spirtes, Glymour e Scheines[63], richiede in input le osservazioni, relative ad un insieme di variabili casuali discrete multivariate, e un livello di fiducia (Level of Significance) con il quale manifestare la significatività del test, ovvero la probabilità che il test rifiuti l'ipotesi nulla seppure questa è corretta. I passi fondamentali della procedura PC sono una serie iterativa di test di indipendenza condizionata. L'output della procedura è un pattern, ovvero un grafo parzialmente orientato. In particolare, si ottiene il true pattern se i campioni sono estratti da un DAG *G*, le variabili sono state tutte misurate (no missing values), e la distribuzione P contiene le sole indipendenze condizionate dovute alla fattorizzazione di P secondo G.

Indichiamo con *n* la cardinalità di un sottoinsieme S di nodi di **X**; se S è usato come condition set nei CI test *n* indica anche l'ordine del test. Fatta questa semplice premessa, la procedura del PC consta di

- una fase di inizializzazione in cui si provvede a creare un DAG completamente connesso, associato al dominio **X**, e si pone $n = 0$;
- una fase iterativa di screening delle relazioni di indipendenza implicite nei campioni.

Ad ogni iterazione si considera, per la coppia di variabili (X,Y), l'insieme C dei nodi adiacenti ad X escluso Y, $C = Adjacencies(C,X)\backslash\{Y\}$, che abbia cardinalità maggiore o uguale al valore corrente di *n*. Per ogni sottoinsieme S di cardinalità *n* estratto da C si effettua il test statistico di ordine *n* per determinare se X e Y sono d-separated da S, ovvero se I(X,Y|S). In caso affermativo: si rimuove il legame X-Y; il condition set S viene memorizzato nella collezione di insiemi (SepSet) che separano i due nodi, si procede ad esaminare un nuovo S reiterando il procedimento. Una volta considerati tutti gli S in C si incrementa *n* e si itera l'algoritmo finché vi sono C con cardinalità (numero nodi) maggiore o uguale a *n*.

[63] Il nome PC non è un acronimo: sono semplicemente le iniziali dei nomi degli autori Peter Spirtes e Clark Glymour.

Dopo lo screening dei test di indipendenza, i legami rilevati formano un grafo non orientato. La definizione della direzione degli archi è effettuata in virtù delle considerazioni sull'indipendenza condizionata, illustrate nella premessa.
In particolare, fra i nodi di una Bayesian network, soltanto i collider possono consentire che il flusso di informazione passi attraverso di loro quando sono attivi. Per qualsisasi terna di nodi X,Y,Z della forma X-Y-Z ("-" adiacenza) ci sono tre soluzioni:

$$(1)\ X \rightarrow Y \rightarrow Z$$
$$(2)\ X \leftarrow Y \rightarrow Z$$
$$(3)\ X \rightarrow Y \leftarrow Z.$$

Fra queste, solo la terza tipologia, *v-structure*, può consentire il passaggio di informazione da *X* a *Z* quando Y è noto. In altre parole, soltanto la v-structure rende *X* e *Z* condizionalmente dipendenti da $\{Y\}$ ($I(X,Z|\{Y\}) > 0$).
Usando questa caratteristica delle reti bayesiane, possiamo identificare tutte le v-structure in una rete per orientarle usando collider identificati nelle precedenti fasi dell'algoritmo. Il numero di archi orientabili è limitato dalla struttura della rete (se non ci sono v-structure non si riesce ad orientare); il meccanismo per identificare i collider è descritto di seguito:
a) Individuare le coppie di nodi che possano essere gli estremi di una v-structure. *Per una terna di nodi X, Y e Z in cui X e Y, Y e Z, sono, rispettivamente, coppie adiacenti, mentre X e Z non sono adiacenti, allora se* $Y \notin C$ *(insieme dei nodi che separano X e Z) sia X padre di Y e Z padre di Y* (se Y è un collider in X-Y-Z, Y non dovrebbe appartenere all'insieme che sepapara X da Z).
I collider e le v-structure identificati permettono di inferire la direzione degli archi non orientati.
b) Per qualsiasi terna di nodi X,Y,Z, se

(i) X è padre di Y,

(ii) Y e Z sono adiacenti,

(iii) X e Z non sono adiacenti,

(iv) L'arco (Y, Z) non è orientato,

allora sia Y padre di Z.

c) Per qualsiasi arco non orientato (X,Y), se c'è un percorso orientato da X a Y, sia X padre di Y.

In merito all'individuazione della direzione di un arco, oltra alle altre opere di Pearl, sono interessanti le relazioni riportate nel lavoro di T. Verma, J. Pearl "An algorithm for deciding if a set of observed independencies has a casual explanation" [VER92].

Uno degli inconvenienti dell'algoritmo PC è che i test CI richiedono, nel caso peggiore, la determinazione delle relazioni di indipendenza di ordine *n-2* (se *n* sono le variabili). Verma e Pearl hanno osservato in proposito che "in generale, l'insieme di tutte le possibili assunzioni di indipendenza aumenta in modo esponenziale all'aumentare del numero di variabili"[VER92]. L'affidabilità del risultato del test è nel numero di campioni a disposizione: all'aumentare del numero di variabili, aumentano le dipendenze da rilevare e quindi maggiore deve essere la quantità di campioni per avere un risultato attendibile. Per quanto concerne il livello di significatività, più è elevato e maggiori sono le dipendenze che vengono estrapolate dal database di campioni. Ciò non deve sorprendere in quanto aumentando la soglia si incrementa la probabilità che il test di indipendenza possa fornire un risultato errato, cioè seppure due variabili sono, nella realtà, indipendenti si rifiuta l'ipotesi di indipendenza. In relazione alla scelta di un corretto valore del significance level è opportuno sottolineare che un valore elevato (≥ 0.6[64]) è indicato quando sono pochi i campioni a disposizione o vi siano da evidenziare delle relazioni molto deboli; al contrario un valore basso è opportuno in presenza di un numero considerevole di osservazioni.

[BUN96] [CHE97]

Come accennato per l'algoritmo CB, la metodologia CI – based è un valido supporto per realizzare approcci ibridi. Un'applicazione è riportata nel contributo di Meek e Spirtes, "Learning Bayesian Networks with discrete variables from data" [SPI], in cui si descrive il Greedy Bayesian Pattern Search algorithm (GBPS). Il GBPS è generalmente più accurato, ma più lento, del PC

[64] Il livello di significatività è una probabilità e come tale assume valori compresi nell'intervallo (0,1) - 0 e 1 esclusi.

nell'apprendere un pattern: l'aproccio è simile all'algoritmo bayesiano, descritto nelle opere di Heckerman, però anziché cercare un DAG - Greedy Bayesian DAG Search (GBDS) - si desidera individuare un pattern. L'idea è di usare la procedura PC per generare un grafo iniziale, ragionevolmente più vicino all'ontologia corretta di un random o empty graph, così da abbreviare la fase di searching del GBPS.

Algoritmo PC

Costruisci il grafo completo non orientato G costituito dai vertici V= {X_1,…,X_n}
n=0 ***{Inizializzazione}***
Repeat {Searching - CI test}
    Repeat
    Seleziona una coppia ordinata di variabili X e Y adiacenti in G (vi è un legame X-Y), tali che il numero di vertici in Adjacencies(G,X)\{Y}[65] *sia ≥ n;*
        Repeat
            Seleziona un sottoinsieme S di Adjacencies(G,X)\{Y} con n vertici;
            Se il test statistico di dipendenza fallisce cancella l'arco X-Y da G e poni Sepset(X,Y) = S e SepSetY,X) = S;
        Until *ogni sottoinsieme S di Adjacencies(G,X)\{Y} con n vertici sia stato selezionato o sia stato trovato un subset S;*
    Until *siano state selezionate tutte le coppie ordinate di vertici adiacenti X e Y tali che il numero di nodi in Adjacencies(G,X)\{Y} sia maggiore o uguale a n;*
    n = n+1
Until *per ogni coppia ordinata di vertici adiacenti X,Y, Adjacencies(G,X)\{Y} ha meno di n vertici*
{Orienting edge} *Per ogni tripla di vertici X,Y,Z tali che la coppia X,Y e Y,Z siano ognuna adiacente in G ma la coppia X,Z non sia adiacente in G, orientare X-Y-Z come X→Y←Z se e soltanto se Y non appartiene nel Sepset(X,Z).*
Repeat
    If X→Y-Z in G, e X e Z non sono adiacenti in G, allora orienta Y→Z
Until *nessun arco può essere orientato.*
Repeat
    If X-Y in G, e vi è un directed path da X a Y orienta Y→Z
Until *nessun arco può essere orientato.*

[65] *Adjacencies(G,X)\{Y}*: insieme dei nodi adiacenti, che presentano quindi un legame esplicito e non indiretto, con X escluso Y.

3.4.2.3 L'ALGORITMO TPDA

L'algoritmo descritto in questo paragrafo nasce dal lavoro di J. Cheng, R. Greiner, J. Kelly, D. Bell e W. Liu ([CHE97]) i quali affermano che, sebbene molti degli algoritmi di Structural Learning producano buoni risultati su diversi data set di riferimento, sono presenti ancora alcuni inconvenienti per una completa automatizzazione del learning from data:

- *Richiesta dell'ordinamento dei nodi*. Molti algoritmi, come illustrato per K2 e K3, richiedono come informazione aggiuntiva un ordinamento sui nodi per ridurre lo spazio di ricerca. Sfortunatamente questa informazione non è sempre disponibile.
- *Complessità computazionale*. Tutti i BN learner sono lenti, per domini di notevoli dimensioni, sia in teoria che nella pratica; ad esempio la maggior parte degli algortimi basati sull'analisi delle dipendenze richiede un numero esponenziale di test.
- *Mancanza di learning tool disponibili*. Anche se ci sono numerosi algoritmi per il learning task, pochi sono pubblicamente disponbili e applicabili ad applicazioni reali di data-mining che spesso contengono centinaia di variabili e milioni di record.

L'algoritmo "TPDA", acronimo per Three-Phase Dependence Analysis, è corretto (nel senso che produce il perfetto modello associato ad una distribuzione di probabilità) allorquando è fornita una quantità sufficiente di training data ed il modello è *monotone DAG faithful*. Il pregio di questo approccio è che richiede al massimo $O(N^4)$ CI test per apprendere N variabili.

L'algoritmo è integrato nel sistema *Bayesian Network Power Constructor*[66] (BNPC), che è disponibile gratuitamente in Internet (http://www.cs.ualberta.ca/~jcheng/bnpc.htm) dall'Ottobre 1997. Il tool comprende anche il software per la discretizzazione di variabili continue e per l'inferenza nelle BN.

[66] Consultare la tabella dei software disponibili in Appendice. Con l'ambiente sviluppato, gli autori hanno concorso e vinto alla "KDD Cup 2001 data mining competition (task one)" per la risoluzione di un reale problema di data mining.

L'interfaccia grafica è user-friendly; in particolare BNPC permette di indicare relazioni derivabili dalla conoscenza a priori, definire un particolare tipo di struttura (tree-structured, classificatore), fornire l'ordinamento per le variabili presenti nel database di campioni.

3.4.2.3.1 Learning Bayesian Network e Information Theory

Essendo il TPDA un dependence - based algorithm, l'idea è apprendere la BN structure estrapolando, dai dati, quali siano le relazioni di indipendenza. Nello specifico si usa la mutua informazione fra coppie di variabili per esaminare l'indipendenza: difatti la cross entropia, come è noto nella teoria dell'informazione, fra due variabili aleatorie A e B, misura l'informazione attesa su B, dopo avere osservato il valore della variabile A. Quindi, in una Bayesian network, se due nodi sono dipendenti, la conoscenza dello stato di un nodo fornirà informazioni anche sull'altro, cosicchè la mutua informazione non soltanto esprime la dipendenza ma quantifica, a differenza del test del Chi quadro, l'entità del legame: per tale ragione gli autori del TPDA definiscono **quantitativi** i test cross entropy based.

$$I(A,B) = \sum_{a,b} P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

$$I(A,B \mid C) = \sum_{a,b,c} P(a,b,c) \log \frac{P(a,b \mid c)}{P(a \mid c)P(b \mid c)}$$

Come già accennato, data la reale distrbuzione di probabilità *P(x), diremo che A e B* sono indipendenti se s solo se *I(A,B)=0*. Sfortunatamente, non si dispone della reale distribuzione di probabilità ma di una stima empirica $\hat{P}_D(x)$, basata sul data set *D*, elicitando le probabilità con le frequenze relative (principio Maximum Likelihood per lo stimatore della probabilità). E' più preciso definire allora una $I_D(A,B) \approx I(A,B)$ poiché $\hat{P}_D(x) \approx$ P(x). Per la stessa ragione, A sarà indipendente da B quando $I_D(A,B) < \varepsilon$, dove $\varepsilon>0$ è una soglia arbitraria prossima allo zero. Il costo computazionale di $I_D(A,B|C)$ è:

- esponenziale nella dimensione di C – richiede un tempo proporzionale al prodotto delle dimensioni del dominio A,B e di tutti i nodi in C;
- lineare nella dimensione del data set *D,* in quanto per eseguire i test statistici bisogna scorrere tutto il data set.

La scelta di usare la mutua informazione come test di indipendenza è legata sia allo stretto legame con la teoria dell'informazione ma anche perché si presta ad un confronto agevole con i metodi entropy based dell'MDL nell'approccio bayesiano: in futuro, gli stessi autori del TPDA, affermano di volere migliorare l'algoritmo con un approccio ibrido.

3.4.2.3.2 Il Three-Phase Dependence Analysis Algorithm

L'algoritmo TPDA trae origine da una versione semplificata dell'algoritmo SLA (Simple Learning Algorithm); i due algoritmi differiscono per la presenza di una fase di inzializzazione che, nel TPDA, consente di avere un'opportuna struttura di partenza e delle informazioni utili per una ricerca euristica. Entrambi gli algoritmi possono essere modificati ed ottimizzati, specie per l'identificazione della direzione degli archi, se è assegnato, a priori, l'ordinamento delle variabili. Di seguito viene illustrata l'idea generale del TPDA.
L'input dell'algoritmo, come nel PC, è costituito dalla tabella di campioni, in cui ogni record è un'istanza completa delle variabili del dominio, e la soglia ε del test. Per ε è valido il discorso accennato per il livello di significatività del PC: va scelta a seconda della dimensione del data set e della distribuzione dei dati. Gli stessi autori del TPDA hanno formulato una relazione empirica per ε; dai risultati sperimentali hanno però rilevato che il valore ε = 0.01 fornisce sovente buoni risultati al punto da assumerlo come valore di default.

Sebbene l'obiettivo di un algoritmo di Structural Learning sia apprendere un un P-map, ciò non è sempre possibile poiché, per alcune distribuzioni di probabilità, non sono formalizzabili tutte le relazioni di indipendenza. Per esempio, sia Z una variabile che indichi il suono di un campanello quando due monete, X e Y, esibiscono la stessa faccia: la struttura che identifica questo dominio è $X \rightarrow Z \leftarrow$

Y ma tale notazione non è perfetta perchè non rispecchia il fatto che X e Z , Y e Z sono tuttavia marginalmente indipendenti. Per di più, l'insieme di indipendenze condizionali implicate da una distribuzione di probabilità non è sufficente per definire un singolo modello BN, difatti ogni distribuzione rappresentata dal grafo $A \rightarrow B$ può anche essere rappresentata da $A \leftarrow B$. Le relazioni di indipendenza sono comunque sufficienti per definire un **essential graph** (o "pattern") ossia un grafo con gli stessi nodi e le stesse "v-structure" impliciti nella distribuzione di probabilità codificata da una Bayesian network. In particolare, in letteratura[67], è strato dimostrato che << Every DAG-faithful distribution has a unique essential graph >>.

L'algoritmo TPDA lavora con le seguenti assunzioni:

- i record nel data set occorrono indipendentemente (iid - "independent and identically distributed");
- gli attributi di una tabella hanno valori discreti e non ci sono missing value in nessun record;
- la quantità dei dati è tale da considerare i test CI affidabili, $I_D(...) \approx I(...)$.

3.4.2.3.3 Le tre fasi dell'algoritmo

Di principio, tutti gli algoritmi dependence-based indagano sulla necessità di un arco per ogni coppia di nodi e, affinché tali decisioni siano corrette fin dall'inizio, sarebbe necessario un numero esponenziale di test CI. A differenza, il TPDA divide il processo di apprendimento in tre parti.

Le tre fasi dell'algoritmo *TPDA* sono

I. *Drafting* (schematizzare)

II. *Thickening* (infoltire)

III. *Thinning* (sfoltire)

[67] Spirtes, P., Glymour, C. and Scheines, R., "Causation, Prediction, and Search", Springer Lecture Notes in Statistics, 1993.Chickering, D. M., "Learning equivalence classes of Bayesian network structures". Proceedings of the twelfth conference on uncertainty in artificial intelligence, 1996.

Nella prima e nella seconda sono consentite alcune decisioni non corrette. La "drafting" phase produce un insieme iniziale di relazioni dopo avere eseguito dei semplici test, con la mutua informazione, sulle coppie di variabili del dominio. Il draft ottenuto è un grafo senza cicli, detto anche *single connected*, dove è presente al più un percorso fra due nodi (come accade nell'algoritmo di Chow-Liu citato in Appendice[68]).

La seconda fase, "thickening", provvede ad aggiungere archi al grafo single connected se non è possibile *d-separare*, in base al risultato di un insieme di CI test, due nodi.

Il grafo risultante alla fine, se è rispettata l'ipotesi DAG faithful, contiene tutti gli archi del true model e in più degli extra-link dovuti ad una mancata individuazione della condizione di d-separation o agli errori del test. Ogni decisione nella fase I richiede un solo test mentre nella fase II il numero di test da valutare è dell'ordine $O(N^2)$, con N numero di nodi della rete.

La terza fase,"thinning", consiste nell'esaminare ogni arco e rimuoverlo se due nodi sono condizionalmente indipendenti; sono necessari $O(N^2)$ CI test per verificare l'esistenza di ogni legame riscontrato nelle due fasi precedenti, per cui, in totale, al massimo sono richiesti $O(N^4)$ CI test, quindi una quantità polinomiale e non esponenziale, per determinare il pattern.

Infine l'algoritmo provvede ad orientare, ove possibile, i legami (essential graph). Nel dettaglio, l'algoritmo TPDA è illustrato, come in [CHE97], nelle righe seguenti in cui si menzionano alcune procedure illustrate in seguito.

[68] In [CHE97] è specificato che l'algoritmo di Chow-Liu può essere visto come un caso speciale del TPDA per le BN con struttura ad albero.

TPDA

Begin [Drafting]

1. Sia $V = \{X_1,...,X_n\}$, le cui osservazioni sono in D, $E = \{\}$ l'insieme dei legami fra le variabili in V; $L = \{\langle X,Y\rangle \mid I(X,Y) > \varepsilon\}$ la lista delle coppie di nodi $\langle X,Y\rangle$ - $X,Y \in V$ e $X \neq Y$ - aventi una mutua informazione almeno pari ad ε.
2. Ordinare *L* in ordine decrescente rispetto a *I(X,Y)*
3. Per ogni coppia $\langle X,Y\rangle$ in L:

   Se **non** c'è un adjacency path fra *X* e *Y* nel grafo corrente *(V,E)* aggiungere $\langle X,Y\rangle$ a *E* e rimuovere $\langle X,Y\rangle$ da *L*.

Begin [Thickening]

4. Per ogno coppia $\langle X,Y\rangle$ in *L*:

   Se è necessario un arco fra X e Y [procedura *EdgeNeeded_H((V,E), X, Y; D,* **ε** *)* segnala true]

   Aggiungere l'arco che unisce la coppia $\langle X,Y\rangle$ a *E*

Begin [Thinning]

5. Per ogni coppia $\langle X,Y\rangle$ in *E*:

   Se non ci sono altri percorsi, oltre questo arco, che connettono *X* e *Y*,

   sia *E' = E* - $\langle X,Y\rangle$ *(si rimuove temporanemaente l'arco che unisce X e Y)*

   Se **non** è necessario un legame fra X e Y

   [procedura *EdgeNeeded_H((V,E'), X,Y; D,* **ε** *)* segnala false] allora *E = E'*

   (se X può essere separata da Y nel grafo "ridotto" (V,E') si rimuove definitivamente il legame X-Y da E)
6. Per ogni coppia $\langle X,Y\rangle$ in *E*:

   Se *X* ha almeno tre vicini oltre *Y*, o *Y* ha almeno tre vicini oltre *X*,

   sia E' = E - $\langle X,Y\rangle$ *(si rimuove temporanemaente l'arco che unisce X e Y)*

   Se **non** è necessario un legame fra X e Y

   [procedura *EdgeNeeded((V,E'), X,Y; D,* **ε** *)* segnala false] allora *E = E'*
7. Orienta i legami in *E* [procedura *OrientEdges*((V,E), D)]

In [CHE97] è illustrato questo esempio che permette di comprendere, a grandi linee, come opera la procedura del TPDA.

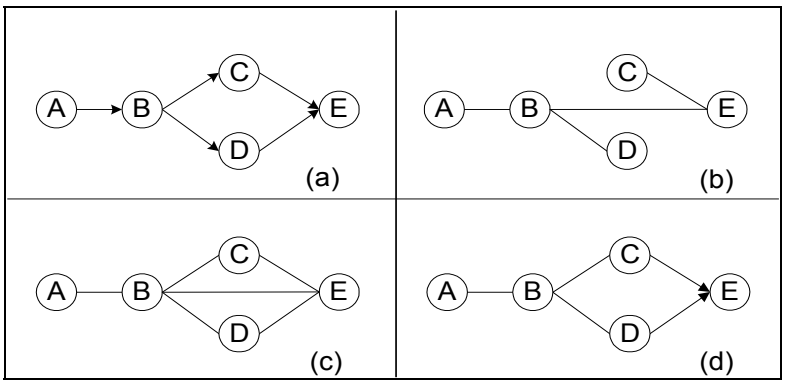


Figura 28 - Fasi dell'algoritmo TPDA

a) Il grafo da apprendere e da cui sono generati i campioni.

b) Supponiamo che dopo avere eseguito i test su tutte le coppie risulti L = { $I(B,D) \geq I(C,E) \geq I(B,E) \geq I(A,B) \geq I(B,C) \geq I(C,D) \geq I(D,E) \geq I(A,D) \geq I(A,E) \geq I(A,C) \geq \varepsilon$} (coppie che presentano un legame di dipendenza). L'ordinare la mutua informazione in ordine decrescente è una euristica giustificata dal fatto che se una coppia presenta un valore della cross entropy elevato è più probabile che vi sia un legame rispetto ad un'altra coppia con minore mutua informazione, la cui connessione, ad esempio, potrebbe essere indiretta. Durante la drafting phase, in modo iterativo, si esamina una coppia di nodi (X,Y) in L e se non è stato già creato un percorso di adiacenze fra X e Y si inserisce il legame X-Y, rimuovendo (X,Y) da L. Al termine di questa fase, siano $L = [\langle B,C\rangle, \langle C,D\rangle, \langle D,E\rangle, \langle A,D\rangle, \langle A,E\rangle, \langle A,C\rangle]$ le coppie di nodi non direttamente connesse nella fase di draft ma che hanno, comunque, una mutua informazione maggiore di ε. In effetti il grafo ottenuto dopo questa fase già rispecchia il modello di riferimento (a), le sole discrepanze sono $\langle B,E\rangle$, un extra link, e la mancanza di $\langle D,E\rangle$ e $\langle B,C\rangle$ a causa degli adjacency path (*D*-*B*-*E*) e (*B*-*E*-*C*) che già forniscono un percorso utile, rispettivamente, da *D* a *E* e da *B* a *C*. La drafting phase cerca di minimizzare il numero di archi mancanti e il numero di archi aggiunti in modo errato: poiché i test statistici eseguiti in questa fase sono effettutati su coppie, ridurre un tipo di errore significa aumentare l'altro. Si termina quando ogni coppia è legata da un percorso

di adiacenze; questa stopping condition è una sorta di trade-off fra i due tipi di errore.

c) Il drafting può fornire qualsiasi struttura fra un grafo vuoto e un grafo completo quindi bisogna raffinare la struttura. Nella seconda fase, "thickening", è usato un test più elaborato, *EdgeNeeded_H*, che sfrutta l'eurstica insita in L per determinare se è necessario connettere una coppia di nodi. La figura (c) mostra il grafo dopo la Thickening Phase: gli archi $\langle B,C \rangle$ e $\langle D,E \rangle$ sono aggiunti perché *EdgeNeeded_H* non può separare queste coppie di nodi secondo i test CI. L'arco $\langle A,C \rangle$ non è stato aggiunto perché la mutua informazione rivela che *A* e *C* sono indipendenti dato, ad esempio, il cut-set $\{B\}$; i link $\langle A,D \rangle$, $\langle C,D \rangle$ e $\langle A,E \rangle$ non sono presenti per ragioni simili. Il grafo risultante da questa fase può includere alcuni extra edge perché il test fallisce nell'individuare coppie di nodi che sono realmente indipendenti. Ciò accade o perché alcuni legami reali possono essere non rilevati fino alla fine di questa fase, impedendo alla procedura *EdgeNeeded_H* di trovare l'esatto cut-set.

d) Poichè le prime due fasi possono aggiungere legami non necessari, la terza fase, "thinning", cerca di identificare questi archi aggiunti in modo errato per rimuoverli. La procedura utilizzata è *EdgeNeeded*: sebbene *EdgeNeeded_H* e *EdgeNeeded* presentino la stessa funzionalità e richiedano lo stesso numero di test, in pratica gli ideatori dell'algoritmo hanno riscontrato che *EdgeNeeded_H*, di solito, usa meno test CI e richiede condition-set più piccoli. La thinning phase, quindi, esegue una prima analisi usando l'euristica *EdgeNeeded_H* come filtro iniziale. Infine si effettua un'ulteriore controllo sui link rimanenti usando la procedura *EdgeNeeded*, più corretta ma lenta[69]. Il grafo 'thinned' è mostrato in figura (d) ed ha la stessa struttura del grafo originale: l'arco $\langle B,E \rangle$ è rimosso perché *B* e *E* sono indipendenti dati $\{C,D\}$. Se il modello di dipendenza sottostante ha una distribuzione di probabilità DAG-faithful e si hanno

[69] La procedura *EdgeNeeded* è chiamata di rado: solo se un nodo X ha almeno altri tre nodi adiacenti oltre Y (o Y ha almeno altri tre nodi oltre X). Difatti se sia X che Y avessero al massimo due adicenti la *EdgeNeeded_H* tenderebbe ad esaminare ogni subset dei "nodi vicini" per prendere una decisione a riguardo.

sufficienti dati per i test statistici, la struttura generata contiene esattamente i legami del true model. In (d) solo due archi sono orientati in base alle informazioni (cutset e collider) ottenute durante l'elaborazione.

In sintesi, le procedure "*EdgeNeeded"* cercano di determinare se c'è un "information flow" fra due variabili *X* e *Y* una volta che sia stato bloccato ogni possible percorso che le unisca. A tale scopo, si valutano i cut-set *C* che blocchino ogni percorso fra *X* e *Y*; la procedurà indicherà la necessità di un legame se la mutua informazione condizionale $I_D(X,Y \mid C)$ supera la soglia ε. Nell'ipotesi che sia noto l'ordinamento topologico delle variabili è facile individuare il cut set appropriato e computare un solo test, in caso contrario bisogna eseguire più test CI per diversi *C*. Nel seguito sono presentate, nel dettaglio, le procedure impiegate per verificare la necessità di un legame fra due nodi come esposto in [CHE97].

3.4.2.3.3.1 Subroutine EdgeNeeded*

Per meglio comprendere ed apprezzare le procedure EdgeNeeded_H e EdgeNeeded, è illustrata in breve la EdgeNeeded* che conduce un'analisi esaustiva ma più lenta. Considerati due nodi *X* e *Y*, assumiamo che *Y* non sia antenato di *X*. La procedura si basa quindi su un approccio "try each subset" (ceca ogni sottoinsieme) come fatto anche dagli altri algoritmi dependence-analysis based quali SGS, PC e Verma e Pearl (citati in Appendice): naturalmente è richiesto un numero esponenziale di test di indipendenza.

Nello step 3, la procedura cerca ogni sottoinsieme *C'* di *C* e se *C'* blocca il flusso informativo fra X e Y allora è un cut-set per la coppia (X,Y). La procedura ritorna valore false se non è necessario un legame fra X e Y. Le informazioni sul cut-set sono memorizzate in una struttura globale *CutSet* in cui è inserita la terna <(X,Y),C'>.

EdgeNeeded* (*G: graph, X, Y: node,* D: Data set, ε : threshold***)***

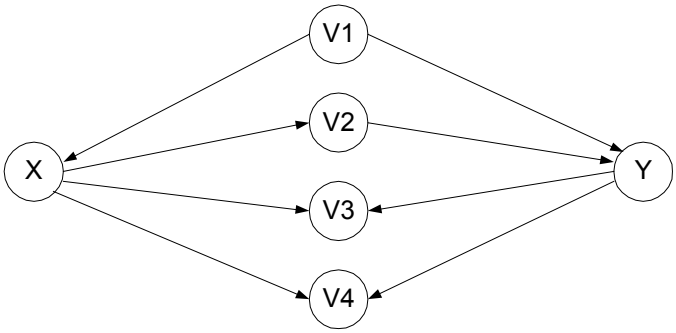
Ritorna il valore true se e solo se dal data set D si evince che è necessario un arco fra X e Y. All'occorrenza, aggiornare l'insieme dei CutSet

1. Sia S_X = Ngbr(X) ∩ AdjPath(X,Y) l'insieme dei nodi adiacenti ad X ma presenti anche in uno dei percorsi di adiacenze fra X e Y*;* in modo simile S_Y = Ngbr(Y) ∩ AdjPath(X,Y). Inizializzare il *CutSet* := {}.
2. Rimuovere da S_X qualsiasi nodo figlio di X*;* e da S_Y qualsiasi nodo figlio di Y. (Se sono note le direzioni degli archi)
3. Per ogni condition-set $C \in \{S_X, S_Y\}$
   Per ogni sottoinsieme $C' \subseteq C$ do
   Sia $s := I_D(X, Y|C')$ - valutazione della mutua informazione
   Se $s < \varepsilon$, (X,Y indipendenti)
   $CutSet := CutSet \cup \{ \langle \{X,Y\}, C' \rangle \}$;
   return ('false').
4. Return ('true') (X e Y sono dipendenti)

3.4.2.3.3.2 Subroutine EdgeNeeded_H (Heuristic)

Poiché non c'è modo di evitare un numero esponenziale di test CI se il risultato di ogni tentativo è semplicemtne "si" o "no", come nel test del Chi quadro, il TPDA usa un *approccio quantitiativo* per i test di indipendenza. La cross entropia, infatti, misura quanta informazione c'è fra due nodi X e Y, fissato un cut-set C. La procedura EdgeNeeded_H (H per heuristic), per una data struttura G ed una coppia di nodi X e Y, inizia con un insieme C che comprende tutti i cut-set. Poi si cerca di identificare e rimuovere i nodi inappropriati da C, uno alla volta, applicando l'euristica che predilige il cut-set con la mutua informazione minima (step 3 - 2) in seguito). L'intero processo richiede un numero di test CI polinomiale. Se *X* e *Y* non sono adiacenti, allora o i padri di X o i padri di Y formano un proprio cut-set; l'idea è trovare un cut-set identificando i padri di X dall'insieme di nodi adiacenti ad X (o i padri di Y dai suoi vicini).

Per illustrare come si opera consideriamo la rete seguente.



Supponiamo di volere determinare la necessità di un legame fra X e Y. Se è dato un ordinamento sui nodi, ovvero sono note le direzioni degli archi, dal grafo si evince che V1 e V2 sono padri di Y e Y non è padre di X. Quindi P={V1,V2} è un opportuno cut set che separa X e Y.

Non è detto, però, che si conosca la direzione degli archi per cui non possiamo determinare se un nodo è padre di Y o meno. A tale scopo, la procedura *EdgeNeeded_H* prima determina S_X e S_Y, entrambi sono dati dall'insieme $\{V1,V2,V3,V4\}$.

Nello step "3 1)", si usa $C=\{V1,V2,V3,V4\}$ come condition-set per esegutie i CI test – determinando cioè se $I_D(X,Y|C) > \varepsilon$. Sebbene questo condition-set chiuda i percorsi *X-V1-Y* e *X-V2-Y,* esso apre anche *X-V3-Y* e *X-V4-Y* (in virtù della definzione di d-separation). Ciò significa che non separa X e Y e così il test fallisce per cui si va allo step "3 2)". In questo step, invece, si considera ogni sottoinsieme di 3 nodi di $\{V1,V2,V3,V4\}$ come possibile condition-set: $\{V1,V2,V3\}$, $\{V1,V2,V4\}$, $\{V1,V3,V4\}$ e $\{V2,V3,V4\}$. Nell'ipotesi monotone DAG-faithful, o $\{V1,V2,V3\}$ o $\{V1,V2,V4\}$ darà il più piccolo valore del test poiché questi sottoinsiemi lasciano un solo percorso aperto (*X-V3-Y* o *X-V4-Y* rispettivamente) mentre i restanti condition-set lasciano aperti tre percorsi. Considerando che sia $\{V1,V2,V3\}$ quello che esibisce il valore minore, si conclude che *V4* è un collider per cui va rimosso dal condition-set in modo da non considerare più questo nodo nell'analisi delle dipendenze fra X e Y.

EdgeNeeded_H (*G: struttura del grafo corrente, X, Y: nodi,* D: Data set, ε : soglia del test ***)***

Ritorna il valore true se e solo se dal data set D si evince che è necessario un arco fra X e Y. All'occorrenza, aggiornare l'insieme dei CutSet

1. Sia S_X = Ngbr(X) ∩ AdjPath(X,Y) l'insieme dei nodi adiacenti ad X ma presenti anche in uno dei percorsi di adiacenze fra X e Y*;* in modo simile S_Y = Ngbr(Y) ∩ AdjPath(X,Y).
2. Rimuovere da S_X qualsiasi nodo figlio di X*;* e da S_Y qualsiasi nodo figlio di Y. (Se sono note le direzioni degli archi)
3. Per ogni condition-set $C \in \{S_X, S_Y\}$ do
   - **1)** Sia $s := I_D(X, Y|C)$. (X e Y indipendenti)
     Se $s < \varepsilon$, $CutSet := CutSet \cup \{ \langle \{X,Y\}, C' \rangle \}$; return ('false').
   - **2)** Finchè il condition set ha un numero di nodi > 1 ($|C| > 1$)
     a. Per ogni i, sia $C := C \setminus \{ i^{th} \text{ nodo di } C \}$, $s_i = I(X, Y \mid C_i)$.
     b. Sia (l'indice) $m = argmin_i \{ s_1, s_2, \ldots \}$
     c. Se $s_m < \varepsilon$, % $s_m = \min(s_1, s_2, \ldots)$
        allora return ('false');
        altrimenti se $s_m > s$ allora considera il prossimo C (Itera Step 3)
        altrimenti sia $s := s_m$, $C := C_m$, e itera lo Step "3.2)".
4. Return ('true') (X e Y dipendenti)

Nelle successive iterazioni, la procedura *EdgeNeeded_H* considera ogni coppia di nodi del sottoinsieme {*V1,V2,V3*} come possibile conditon set: {*V1,V2*}, {*V1,V3*} e {*V2,V3*}. Dopo i test, assumendo che il cut-set {*V1,V2*} separi *X* e *Y*, cioè $I_D(X,Y \mid \{V1,V2\}) \approx 0$, e che *EdgeNeeded_H* ritorni "false", non avviene l'aggiunta dell'arco X-Y perché sarebbe inappropriata. Invece, se da un altro data set di campioni si evincesse un ulteriore dipendenza fra *X* e *Y*, *EdgeNeeded_H* continuerebbe a cercare fra gli altri sottoinsiemi. Infine, potrebbe risultare un flusso significante di informazione fra X e Y per tutti i condition set considerati: in tal caso *EdgeNeeded_H* ritornerebbe valore "true" il che implicherebbe l'aggiunta del legame X-Y.

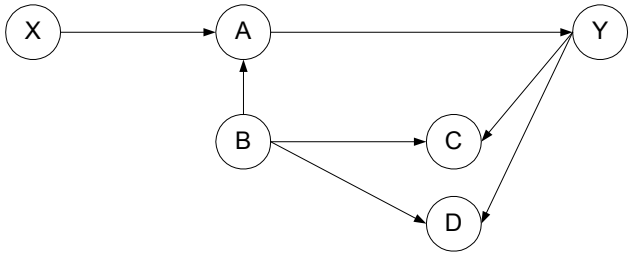
3.4.2.3.3.3 Subroutine EdgeNeeded (Guaranteed)

La procedura *EdgeNeeded_H* usa un'euristica che rimuove i padri di *X* (o *Y*) dal condition-set. In particolari strutture è più complicato separare i nodi X e Y se:

(1) Esiste almeno un percorso fra X e Y attraverso un figlio di Y che a sua volta è un collider per il percorso.

(2) In tali percorsi, vi sono uno o più collider, oltre al nodo figlio, che sono antenati di Y.

In tal caso, la procedura *EdgeNeeded_H* può individuare in modo errato un padre di Y come figlio di Y e rimuoverlo erroneamente dal condizionamento. In sintesi, la procedura fallisce nel separare i due nodi.



Ad esempio, nella figura precedente, si vuole separare X e Y usando il sottoinsieme dei nodi vicini a Y, N_2={*A*,*C*,*D*}. La procedura *EdgeNeeded_H* considererà {*A*,*C*,*D*} come condition-set. Poichè tale insieme lascia due percorsi open, *X-A-B-C-Y* e *X-A-B-D-Y,* si valutano i sottoinsiemei di 2 elementi {*A*,*C*}, {*A*,*D*} e {*C*,*D*} come possibili condition-set. Ognuno di questi, a sua volta, lascia un percorso aperto, rispettivamente *X-A-B-C-Y*, *X-A-B-D-Y* e *X-A-Y*. Se la mutua informazione fra X e Y è più piccola quando *X-A-Y* è open, la procedura rimuoverà il nodo *A* nei successivi tentativi. Chiaramente ciò induce ad un errore nel separare X e Y. In questo esempio accade che il neighbour-set di *X*, C={*A*}, può separare *X* e *Y*, ma ci sono modelli più complessi per i quali la procedura fallisce. In ogno caso tali strutture sono piuttosto rare.

La Procedura *EdgeNeeded* risolve l'ambiguità anche in tali situazioni problematiche. La principale differenza fra *EdgeNeeded_H* e *EdgeNeeded* è che oltre all'aggiunta/esclusione di ogni neighbour di X – S_X – *EdgeNeeded* considererà anche l'inclusione/esclusione dei vicini di quei neighbour - chiamati S_X' – (in modo simile per S_Y e S_Y'). Si osservi in proposito che la procedura

considera solo uno di questi possibili sottoinsiemi – o $S_X \cup S_X'$ o $S_Y \cup S_Y'$; ma non entrambi.

3.4.2.3.3.4 Orienting Edges

Si procede come accennato nell'algoritmo PC.

EdgeNeeded (*G: struttura del grafo corrente, X, Y: nodi,* D: Data set, ε : soglia derl test ***)***

Ritorna il valore true se e solo se dal data set D si evince che è necessario un arco fra X e Y. All'occorrenza, aggiornare l'insieme dei CutSet

2. Sia S_X = Ngbr(X) ∩ AdjPath(X,Y) l'insieme dei nodi adiacenti ad X ma presenti anche in uno dei percorsi di adiacenze fra X e Y*;* in modo simile S_Y = Ngbr(Y) ∩ AdjPath(X,Y).
3. Sia $S_X' = AdjPath(X,Y)$ I $(\bigcup_{x \in S_X} Ngbr_G(x) - S_X)$ l'insieme dei nodi in S_X che sono sia su un percorso di adiacenze fra X e Y ma che non appartengano a S_X; allo stesso modo $S_Y' = AdjPath(X,Y)$ I $(\bigcup_{y \in S_y} Ngbr_G(y) - S_Y)$
4. Sia C il più piccolo fra $\{ S_X \cup S_X', S_Y \cup S_Y' \}$
5. Sia $s = I_D(X,Y|C)$.
6. Se $s < \varepsilon$, $CutSet := CutSet \cup \{\langle \{X,Y\}, C\rangle\}$; return ('false') (X e Y indipendenti)
7. Finchè il condition set ha un numero di nodi > 1 ($|C| > 1$)
    a. Per ogni i, sia $C := C \setminus \{ i^{th} \text{ nodo di } C\}$, $s_i = I(X, Y \mid C_i)$.
    b. Sia (l'indice) $m = argmin_i \{ s_1, s_2, \ldots \}$
    c. Se $s_m = \min(s_1, s_2, \ldots) < \varepsilon$,
    allora $CutSet := CutSet \cup \{\langle \{X,Y\}, C_m\rangle\}$ return ('false');
    altrimenti se $s_m > s$ allora Step 7
    altrimenti $s := s_m$, $C := C_m$, e itera lo Step 6.
7. Return ('true') (X e Y dipendenti)

3.5 CENNI AI METODI CHE GESTISCONO MISSING VALUES

Nell'apprendimento dai dati si presenta il problema dei missing values: infatti una delle principali assunzioni negli algoritmi illustrati è che il database dei casi sia completo.

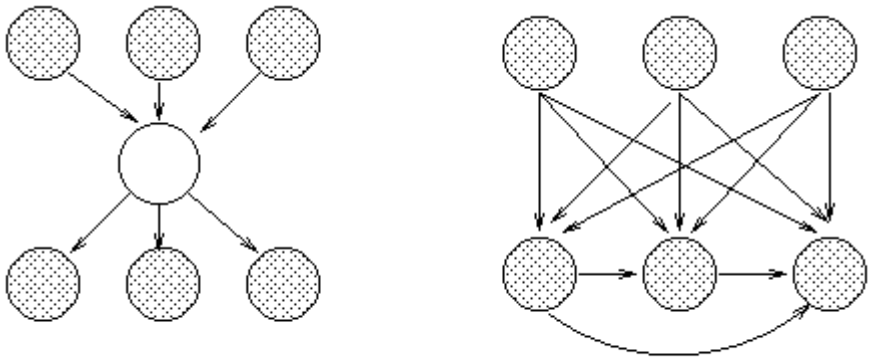
Quando invece è incompleto, ogni dato disperso "x?", per una generica variabile X_i, potrebbe essere sostituito da uno qualsiasi degli stati $\{x_i^1,...,x_i^{r_i}\}$ assumibili da X_i.

Dal punto di vista puramente teorico, si potrebbero considerare tutti i database ottenuti assegnando un valore ai missing value auspicando un averaging sui risultati: un'analisi bayesiana rigorosa diventerebbe però irrealizzabile. Identificando l'insieme dei campioni, il data set, con C_h e l'insieme dei record con missing value con $C_h^{'}$, la probabilità che C_h sia attribuibile ad una distribuzione di probabilità B_p codificata da una rete Bayesiana B_s è

$$p(C_h \mid B_s, B_P) = \sum_{C_h^{'}} p(C_h, C_h^{'} \mid B_s, B_P)$$

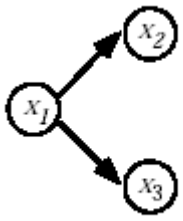
La sommatoria rispetto a $C_h^{'}$ esplora tutti i possibili valori per i missing value; la complessità del problema cresce in modo esponenziale all'aumentare dei dati non rilevati.

Un variabile latente - hidden variable - rappresenta invece un'entità presupposta circa la quale però non si hanno osservazioni. Nell'ipotesi di generare i campioni per l'apprendimento dalla struttura nella figura (a) seguente, in cui il nodo non ombreggiato è una hidden variable, si ottiene la figura (b) che è meno espressiva e lontana dal true model.



Una variabile nascosta può essere schematizzata con record di missing value. Ad esempio, date tre variabili binarie, X_1,X_2,X_3, se X_1 fosse una hidden variabe si avrebbe

Case	x_2	x_3
1	*absent*	*absent*
2	*present*	*present*



$$
\begin{aligned}
P(B_{S2}, D) \;=\; & \\
& \int_{B_P} P(x_1=0, x_2=0, x_3=0 \mid B_{S2}, B_P)\, P(x_1=0, x_2=1, x_3=1 \mid B_{S2}, B_P)\, f(B_P \mid B_{S2})\, P(B_{S2})\, dB_P \\
+ & \int_{B_P} P(x_1=0, x_2=0, x_3=0 \mid B_{S2}, B_P)\, P(x_1=1, x_2=1, x_3=1 \mid B_{S2}, B_P)\, f(B_P \mid B_{S2})\, P(B_{S2})\, dB_P \\
+ & \int_{B_P} P(x_1=1, x_2=0, x_3=0 \mid B_{S2}, B_P)\, P(x_1=0, x_2=1, x_3=1 \mid B_{S2}, B_P)\, f(B_P \mid B_{S2})\, P(B_{S2})\, dB_P \\
+ & \int_{B_P} P(x_1=1, x_2=0, x_3=0 \mid B_{S2}, B_P)\, P(x_1=1, x_2=1, x_3=1 \mid B_{S2}, B_P)\, f(B_P \mid B_{S2})\, P(B_{S2})\, dB_P
\end{aligned}
$$

La difficoltà principale è che, seconda questa schematizzazione, vi sarebbe un numero illimitato di hidden variable e quindi anche un numero illimitato di strutture che possano contenerle. Le soluzioni al problema sono diverse: limitare il numero di variabili nascoste, usare la conoscenza a priori o degli indicatori statistici che suggeriscano la presenza di tali variabili. [COO92] [HEC94]
I metodi presenti in letteratura, risolvono il problema dei missing values e delle hidden variable, effettuando un'approssimazione asintotica del database, con metodi iterativi quali l'algoritmo EM, o approssimazioni di tipo stocastico. Di recente Friedman [FRI97] ha applicato l'algoritmo EM (Expectation – Maximization) per lo Structural Learning di reti bayesiane concependo l'algoritmo **SEM - Structural Learning EM**. Sia *O* l'insieme delle variabili osservabili, *H* l'insieme delle variabili con missing value (i cui elementi sono indicati con *h*), *M* una collezione di strutture candidate a rappresentare un dominio e M un elemento di tale insieme: in particolare, con l'espressione M^h si ipotizza che la struttura M possa rappresentare il dominio; M_n rappresenta la struttura all'n-sima iterazione; Θ^M è il vettore di parametri associato a M.

Ripeti per $n = 0,1,\ldots$ finché non c'è convergenza o si raggiunge un limite, fissato a priori, su *n*

Computare la probabilità a posteriori $p(\Theta^{M_n} \mid M_n^h, O)$

E- step: Per ogni M, computare la quantità

$$Q(M, M_n) = E[\log(p(H, O, M^h) \mid M_n^h, O) = \sum_h p(h \mid O, M_n^h) \log p(h, O, M^h)$$

M - step: Scegliere M_{n+1} che massimizza $Q(M,M_n)$

Criterio di convergenza: Se $Q(M,M_n) = Q(M_{n+1},M_n)$ allora restituisci M_n

4 LA NOSTRA PROPOSTA:L'APPROCCIO MULTI-ESPERTO

La fase di apprendimento, sia della struttura che dei parametri, di una rete Bayesiana è un processo di estrazione della conoscenza dai dati - Knowledge Discovery from data.

L'intento dell'Intelligenza Artificiale è di automatizzare il learning, quindi, come accennato nei capitoli precedenti, nello Structural Learning la "human experience" è sostituita dai learning algorithm.

Questi algoritmi presentano svantaggi e vantaggi; i dependence-based algorithm consentono un'analisi efficiente dal punto di vista del tempo impiegato ma il risultato è dipendente dalla robustezza del test di indipendenza. Gli algoritmi che usano l'approccio bayesiano (massimizzazione di una opportuna probabilità a posteriori o di una metrica), invece, sono più lenti ma maggiormente affidabili nel riconoscimento dei legami.

D'altronde, in letteratura, non è menzionato un metodo che ricostruisca, perfettamente, una rete dai campioni; il risultato dell'apprendimento è in genere una struttura con archi mancanti, aggiunti e invertiti. In alcuni casi, vincoli molto restrittivi, quali l'ordinamento delle variabili nel K2, contribuiscono a migliorare la ricostruzione.

Tuttavia lo scopo dell'analisi di problemi reali, ad esempio con tecniche di data mining, è proprio l'estrazione della conoscenza: un vincolo, quale l'ordinamento, dovrebbe essere insito nell'output del processo di apprendimento più che rappresentarne un input. Anche l'inserimento di un livello di fiducia per il test di indipendenza (algoritmo PC) potrebbe essere visto come una sorta di vincolo; bisogna però osservare che la scelta di tale parametro è legata principalmente al numero di record presenti nel database di campioni più che ad informazioni a priori sul dominio.

L'idea nata da questo lavoro di tesi è di esaminare la validità di un approccio multi-esperto che, confrontando i risultati dei diversi algoritmi e prescindendo da ipotesi a priori molto restrittive, determini i legami di una Bayesian network dai dati.

Il capitolo è organizzato in tre parti: nella prima vengono forniti gli elementi teorici sui sistemi ME (Multiple Expert), nella seconda è illustrato, in breve, il tool implementato sia per integrare gli esperti/algoritmi di structural learning che per condurre la sperimentazione, e nella terza si riportano alcuni dei risultati ottenuti (altri sono consultabili in un allegato alla tesi).

4.1 ASPETTI TEORICI DEL MULTIPLE EXPERTS (ME)

Negli ultimi anni diversi gruppi di ricerca hanno concentrato la loro attenzione su un approccio di tipo multi-esperto (noto anche come *Combination of Multiple Experts - CME*) soprattutto in discipline dell'Intelligenza Artificiale quali la pattern recognition. Questo approccio trova la sua ragion d'essere nell'assunzione che, combinando opportunamente i risultati forniti da un insieme di esperti è possibile ottenere delle prestazioni migliori di quelle ottenibili da ogni esperto preso singolarmente. Il consenso di un insieme di esperti può compensare la debolezza di ogni singolo esperto, mantenendo inalterate le capacità di ognuno di essi.

La combinazioni di più esperti è un aspetto particolare di quel campo che va sotto il nome di *sensor fusion*. In generale si parla di sensor fusion quando si combinano le informazioni provenienti da sensori di diverso tipo che osservano uno stesso fenomeno (un fenomeno fisico, una scena, un oggetto). Volendo affrontare il progetto di un sistema ME, alcune regole generali da tener presente possono essere desunte da quelle date da Nahan e Pokoski[70]. Queste consentono di evitare combinazioni di esperti che generalmente *non* producono complessivamente un significativo vantaggio:

1. combinare i risultati che provengono da più esperti che hanno un basso grado di accuratezza (ovvero che hanno una probabilità di riconoscimento corretto minore dello 0.5) ;
2. combinare i risultati che provengono da molti esperti estremamente accurati (ovvero con probabilità di riconoscimento corretto maggiore dello 0.95);
3. in presenza di un gran numero di esperti (cioè maggiore di 8 o 10), l'aggiunta di ulteriori esperti dello stesso tipo di quelli già presenti.

[70] Queste regole sono riportate in B. Ackermann, H. Bunke, "Combination of Classifiers on the Decision Level for Face Recognition", *Technical Report IAM-96-002*, Institut für Informatik und angewandte Mathematik, Universität Bern, January 1996.

Va detto tuttavia che esperti di tipo diverso possono avere un impatto significativo sul sistema, anche se non forniscono prestazioni, in termini di percentuale di riconoscimento, particolarmente buone. Altre considerazioni generali, di cui tener conto in fase di progetto, riguardano la sequenza temporale del processo di combinazione. E' infatti possibile pensare sia ad architetture seriali che parallele, così come sono possibili schemi di combinazione multi-stadio, in cui ogni stadio combina le uscite degli esperti intervenuti nello stadio precedente.

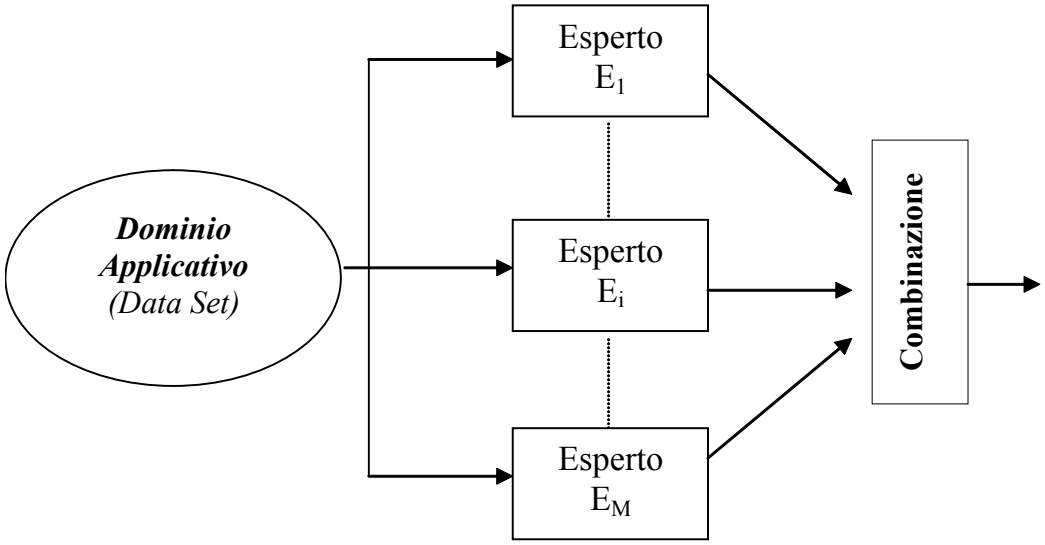


Figura 29 – Schema a blocchi dell'approccio multi-esperto

REGOLE DI COMBINAZIONE

Le regole di combinazione presenti in letteratura sono molteplici; alcune sono pensate per applicazioni specifiche, mentre altre sono applicabili in generale. I metodi possono essere raggruppati in due categorie: approcci euristici, come strategie basate sul voto o somme dei punteggi, e metodi basati sulla teoria statistica della decisione, come la combinazione Bayesiana e le regole da essa derivate ("*Behavior-Knowledge Space*").

Inoltre, poiché il combinatore può essere visto come un classificatore il cui vettore di ingresso è formato dalle uscite degli M esperti, ogni metodo di classificazione basato su un vettore di feature può essere applicato a questo tipo di problema, come, ad esempio, i classificatori neurali, gli alberi di decisione e la logica fuzzy.

4.1.1.1 STRATEGIE BASATE SUL VOTO

Questo tipo di strategie fanno riferimento ai tipici processi decisionali umani: ogni esperto ha a sua disposizione un voto, che attribuisce al risultato che ritiene più probabile.

Quindi si conta il numero di voti ricevuti ed il sistema ME prende la sua decisione in base ad un criterio a maggioranza. Se la votazione finisce in pareggio, il campione può essere rigettato o si possono utilizzare dei criteri per dirimere la questione. In genere, in caso di pareggio, si pesano i voti assegnati dagli esperti tramite coefficienti calcolati sulla base delle prestazioni ottenute dai vari esperti su di un opportuno insieme di dati.

I criteri di maggioranza per la decisione finale possono essere più o meno stretti: in genere si parla di *voto di consenso* se basta una maggioranza relativa o *voto di maggioranza* se è necessaria una maggioranza assoluta.

Varianti di questo tipo di strategia sono quelli basati sui metodi di votazione pesata, in cui ogni esperto pesa il proprio voto tramite un opportuno coefficiente. I pesi devono essere evidentemente determinati prima del processo di combinazione, come nel caso dei coefficienti utilizzati per dirimere i pareggi nel caso di voto di consenso.

4.1.1.2 STRATEGIE BASATE SUL PUNTEGGIO

Se ciascun esperto assegna un punteggio ad un'eventuale configurazione (ad esempio, per le BN alla presenza di un arco nella struttura), in modo che i vari punteggi siano tra loro comparabili o esista quantomeno una trasformazione efficace che li renda tali, è possibile applicare una strategia di combinazione basata sui punteggi. Tali strategie sfruttano, quindi, informazioni più ricche rispetto alle strategie basate sul voto. Il modo più semplice di effettuare la combinazione in questo caso è quello di sommare i punteggi attribuiti al caso esaminato. [MOL02]

4.1.1.3 L'APPROCCIO MULTI-ESPERTO APPLICATO ALLE BAYESIAN NETWORK

Per lo Structural Learning di reti Bayesiane, in prima analisi, il metodo più immediato è quello basato sulle strategie di voto ed è, quindi, quello adottato in questo lavoro. Il nostro scopo è di combinare M esperti E^i, i=1,…,M, con un *combinatore* K; ciascun esperto acquisisce in ingresso i dati che provengono da un dominio e delle informazioni aggiuntive peculiari dell'algoritmo (ad esempio livello di significatività del test per l'algoritmo PC) dopodiché il combinatore K analizzerà le reti Bayesiane apprese dai singoli esperti, individuando quali legami siano effettivamente esistenti e quali invece possano essere considerati assenti. Un legame fra due nodi X - Y che *non* abbia ricevuto un numero di voti pari almeno alla metà degli esperti più uno, non sarà esplicitato nella struttura. Se $v(n_i)$ è una funzione che restituisce il numero di voti attribuiti da M esperti alla presenza dell'arco n_i in una rete Bayesiana, la regola di combinazione nel caso di voto di maggioranza, ha esito positivo, se $v(n_i) > \frac{M}{2}$.

La nostra scelta su quanti e quali esperti utilizzare per lo studio belle BN, è nata dall'analisi dello stato dell'arte nelle tecniche di Structural Learning e dall'individuare gli algoritmi più interessanti proposti nella letteratura scientifica. Abbiamo così considerato un numero dispari di esperti, 5, in modo da evitare l'eventualità che si presenti un caso di pareggio: algoritmo K2, K3, PC, TPDA e Bayesiano (descritti nel capitolo precedente). Nel paragrafo dedicato alla sperimentazione sarà chiarito graficamente, con alcuni esempi, l'approccio seguito.

4.2 IL TOOL REALIZZATO: *BAYEXPERT*

L'obiettivo di questo lavoro di tesi è quello di fornire una panoramica sugli algoritmi di Structural Learning e di saggiare la validità di un approccio multi-esperto, in particolare laddove i singoli algoritmi forniscano risultati deludenti.
Inizialmente, abbiamo studiato le procedure di apprendimento esposte dalle comunità scientifica concentrando l'attenzione sull'algoritmo Bayesiano, K2, K3, PC, TPDA ed individuando, così, una rosa di cinque esperti.
Nella fase operativa, abbiamo sviluppato un tool, in Java, che consentisse sia di effettuare lo structural learning usando i singoli algoritmi ma che mettesse anche a disposizione un prototipo per l'approccio multi-esperto. Per dare una maggiore organicità al discorso, senza i tecnicismi tipici dell'implementazione, una descrizione approfondita delle strutture utilizzate (per la rappresentazione delle reti bayesiane) e della realizzazione dei learning algorithm è riportata, separatamente, in un allegato.
In sintesi, il lavoro è stato svolto secondo i seguenti step:

- implementare i singoli algoritmi e testarli su database di riferimento, reperiti in appositi repository di data set per Bayesian network e machine learning;
- constatato il corretto funzionamento degli algoritmi, abbiamo sviluppato un semplice tool che ne consentisse l'utilizzo con un'opportuna interfaccia grafica. L'idea nata durante il lavoro è stata proprio di creare un ambiente, un piccolo laboratorio, sulle reti Bayesiane che abbiamo battezzato ***BayExpert***;
- concepire un approccio multi-esperto: la scelta operata è di seguire una strategia a maggioranza.

Utilizzando le funzionalità di Java per la grafica, in particolare l'API JSwing, abbiamo realizzato un'interfaccia dotata di menu con le funzioni illustrate nei successivi paragrafi.

4.2.1 BAYEXPERT: ACQUISIZIONE E MEMORIZZAZIONE DI UNA BAYESIAN NETWORK

L'acquisizione di una rete può avvenire "manualmente", cioè l'utente inserisce il numero di nodi del dominio, poi il nome, gli stati e l'insieme dei padri di ogni variabile e le CPT (tabelle di probabilità condizionate).
In alternativa, il loading di una BN avviene da file con estensione .xml o avviando un processo di structural learning (in particolare ogni algoritmo provvede, implicitamente, ad apprendere anche le probabilità in modo da disporre immediatamente di una rete su cui potere inferire).

La scelta dell'XML[71] (eXtensible Markup Language) non è del tutto casuale: la tecnologia XML permette di rappresentare informazioni in formato testuale e strutturato con un metalinguaggio pensato per una rappresentazione comoda e snella dei documenti, organizzati in gerarchie e quindi anche delle reti Bayesiane.
In tale ottica il centro di ricerca della Microsoft che si occupa dello studio delle reti Bayesiane, ha provveduto a stipulare uno standard di riferimento per lo scambio di informazioni sulle BN: il formato BNIF.

[71] Nato nel febbraio 1998 come raccomandazione del W3C, un consorzio di aziende che si occupa della definizione e dell'aggiornamento della maggior parte degli standard "de facto" del mondo di Internet. Anche se la tecnologia XML è stata studiata per i documenti fruibili in Internet, sono numerose altre applicazioni in cui l'XML consente uno scambio di informazioni immune da interpretazioni erronee ed agevole (ad esempio, descrizioni di database).

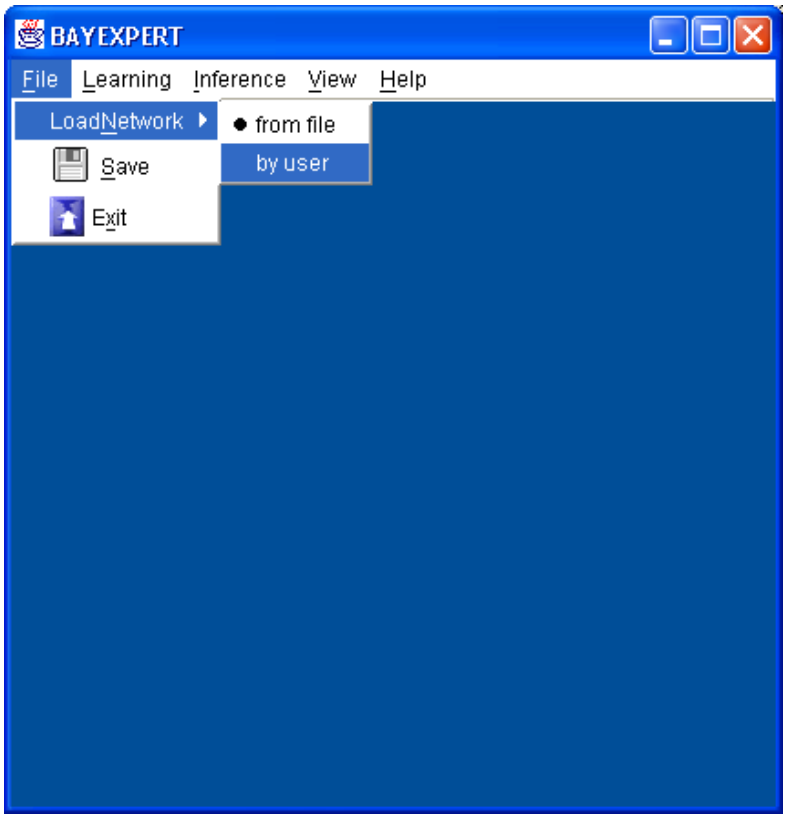


Figura 30 - BayExpert: Loading di una BN

4.2.1.1 IL FORMATO BIF

Negli ultimi anni c'è stata una discussione crescente sul potenziale valore di un ***Bayesian Network Interchange Format*** **(BNIF)** per promuovere la collaborazione fra i ricercatori della Uncertainty and Artificial Intelligence (UAI) community che studiano le reti Bayesiane. L'indubbio vantaggio di poter creare del software sulle BN secondo le proprie esigenze ma di riuscire anche ad usufruire delle reti sviluppate da terzi, favorendo l'interazione e lo scambio di informazioni, ha fatto sì che una proposta di BNIF fosse disponibile in Internet dal 1996 in seguito alla Conference on Uncertainty in Artificial Intelligence tenutasi in quell'anno (www.research.microsoft.com/research/dtg/bnformat/). Invece, durante la Conference on Uncertainty in Artificial Intelligence del 1998, ci fu una discussione sul futuro del Bayesian Network Interchange Format che designò la tecnologia XML come valido supporto per il formato BNIF. A riguardo, La proposta suggerita da Fabio Cozman (www.cs.cmu.edu/~fgcozman), Marek Drudzel (www2.sis.pitt.edu/~drudzel) e Daniel Garcia

(www2.sis.pitt.edu/~drudzel) è l'**XMLBIF** (**XML**-based **B**ayesNets **I**nterchange **F**ormat)[72]. Questo formato è utilizzato in JavaBayes, un software sviluppato dallo stesso Cozman, e i sistemi GeNie, un progetto a cui partecipano sia Drudzel che Garcia. Inoltre, anche società affermate nel campo delle reti Bayesiane, quali Netica e Hugin, sembrano interessate all'utilizzo dell'XMLBIF.

Poiché l'XMLBIF è più immediato e impiega meno tag, rispetto all'XBN, per la descrizione di una rete Bayesiana, è il formato di riferimento in questo lavoro anche in virtù degli sviluppi su menzionati.

4.2.2 BAYEXPERT: STRUCTURAL LEARNING

L'apprendimento di una struttura implica i seguenti step.

1. Scelta dell'algoritmo (K2,K3,PC,TPDA,Bayesiano) o dell'approccio multi-esperto.

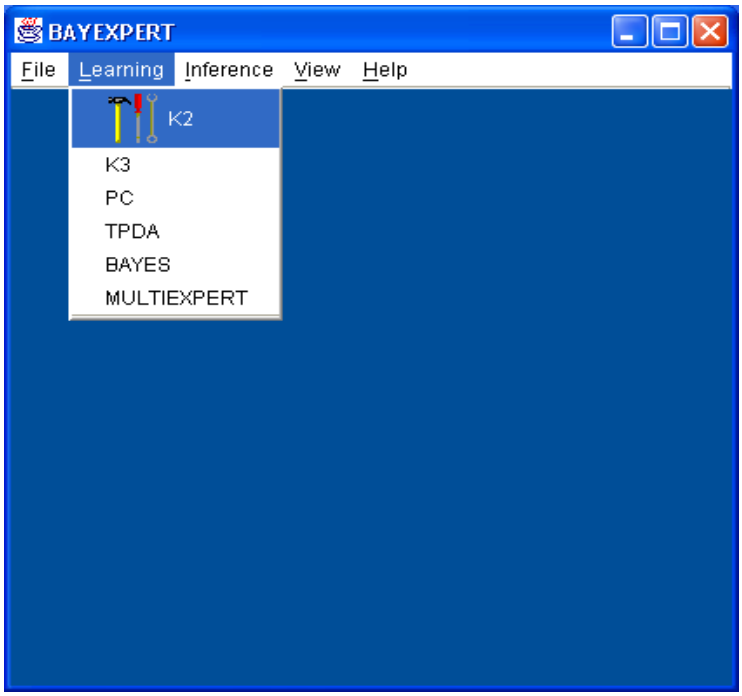


Figura 31 – BayExpert: scelta dell'algoritmo di Structural Learning

[72] Alla stesura dell'XMLBIF format hanno contribuito altri ricercatori creando l'XMLBIF working group: Fabio Cozman, Marek Druzdzel, Daniel Garcia, Akihiro Shinmori (esperto delle problematiche dell'XML), Brent Boerlage, Frank Jensen ed infine Bruce D'Ambrosio che ha sostenuto il progetto.

2. Selezione del data set: file .dat. Il file di input .dat per lo Structural Learning ha la seguente struttura, in cui il carattere "|" ha la funzione di separatore per l'intestazione, la quale riassume il dominio, mentre lo spazio separa le singole osservazioni che formano un campione.

Numero di nodi della rete Bayesiana <Nome $nodo_1$>|<Nome $nodo_2$>|…<Nome $nodo_n$> <Numero stati $nodo_1$>|<Numero stati $nodo_2$>|…<Numero stati $nodo_n$> Numero di campioni (record) Campioni	4 UT_1|BT_1|Ho_1|Pr_1 2|2|2|2 10000 yes no no yes no no no yes yes yes yes yes

3. Indicare l'ordinamento delle variabili. In tale proposito un'opportuna interfaccia, illustrata nel seguito del capitolo, consente di lasciare la disposizione delle variabili inalterata, ordinarla a proprio piacimento o secondo un probabile ordinamento topologico fornito dai test di indipendenza).

4. Se l'algoritmo lo richiede, inserire i parametri necessari (nel multi-esperto è richiesto per PC, TPDA, Bayesiano). Ad esempio, per il PC il livello di fiducia nel test; l'algoritmo Bayesiano, invece, necessita del numero di iterazioni e del valore α che quantifica la conoscenza a priori (in effetti, ci siamo posti nel caso peggiore, completa ignoranza sul dominio, considerando come grafo iniziale una empty network[73]).

Oltre alla struttura, gli algoritmi provvedono a determinare le CPT.

[73] Il tool, in un lavoro di tesi successivo, potrebbe essere migliorato con funzionalità che consentano di far intervenire un esperto sia durante il processo di apprendimento che a priori, specificando eventuali archi presenti o proprietà peculiari dei nodi (se un nodo deve essere padre, ad esempio).

Algoritmo	Parametri	Valori usati (sperimentazione)	Selezionabile dall'utente
K2	vincolo sul numero dei padri di un nodo	5	No
K3	vincolo sul numero dei padri di un nodo	5	No
PC	Significance Level: rappresenta la probabilità che il test fornisca un risultato errato. Deve essere scelto piccolo nel caso i record nel database siano numerosi o si vogliano evidenziare le sole relazioni più forti. Altrimenti è opportuno un valore più elevato, sempre compreso fra 0 e 1.	0.01 0.1 0.3 0.5 0.7	Si
TPDA	Soglia ε del test: è un valore prossimo allo zero. E' indicato un valore piccolo per evidenziare relazioni deboli (basso valore della mutua informazione), un valore elevato segnala relazioni più forti. E' opportuno scegliere il valore anche in base al numero di record del database come esposto nel PC.	0.01 (valore suggerito anche dagli ideatori dell'algoritmo)	Si
Bayesiano	Valore α: quantifica la conoscenza a priori; indicare un valore elevato quante più informazioni sono disponibili a priori (ad esempio se il grafo di partenza presenta degli archi noti).	1, 3, 10 50, 100	Si
	Numero di iterazioni (nel caso l'algoritmo non raggiunga a convergenza permette di diminuire il run-time)	20	Si

4.2.3 BAYEXPERT: INFERENZA

Una volta appresa una rete o acquisita da file, è possibile avviare un processo di inferenza. In tale proposito, un'interfaccia permette di selezionare le variabili in cui inserire l'evidenza (fissare lo stato).
L'algoritmo di bucket elimination provvederà a fornire le informazioni richieste da opportune query di inferenza probabilistica (realizzate molto semplicemente selezionando il tipo di query e la variabile rispetto alla quale eseguirla).

4.2.4 BAYEXPERT: VISUALIZZAZIONE DI UNA BN

Per visualizzare una rete Bayesiana graficamente il file di origine deve essere un file xml (formato XMLBIF).

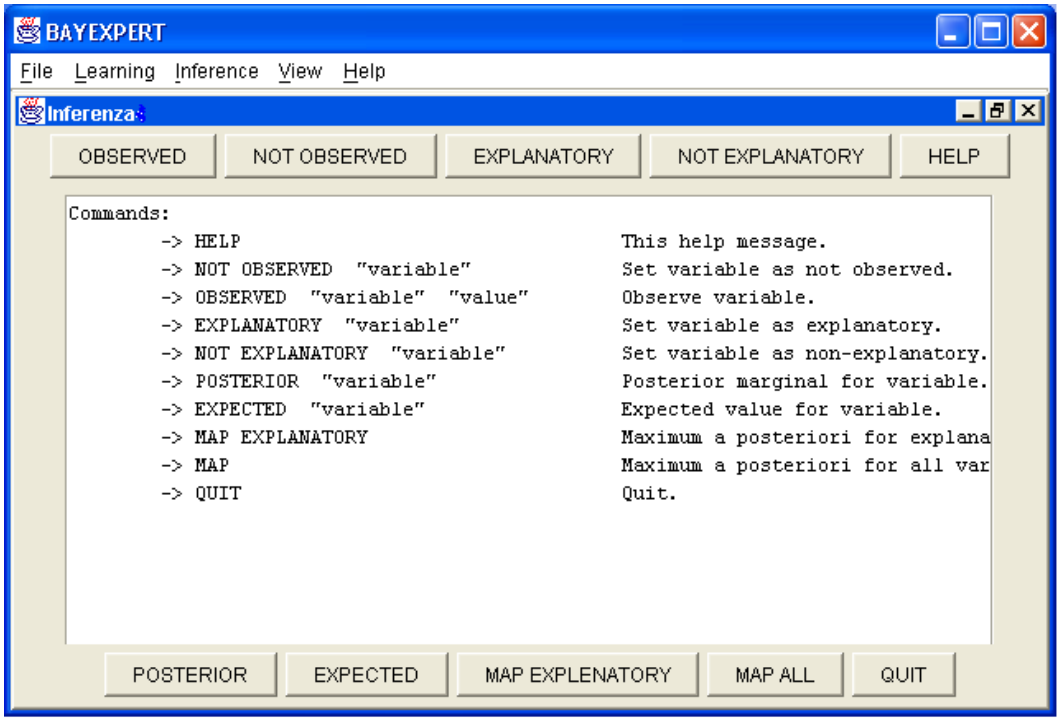


Figura 32 – BayExpert: inferenza

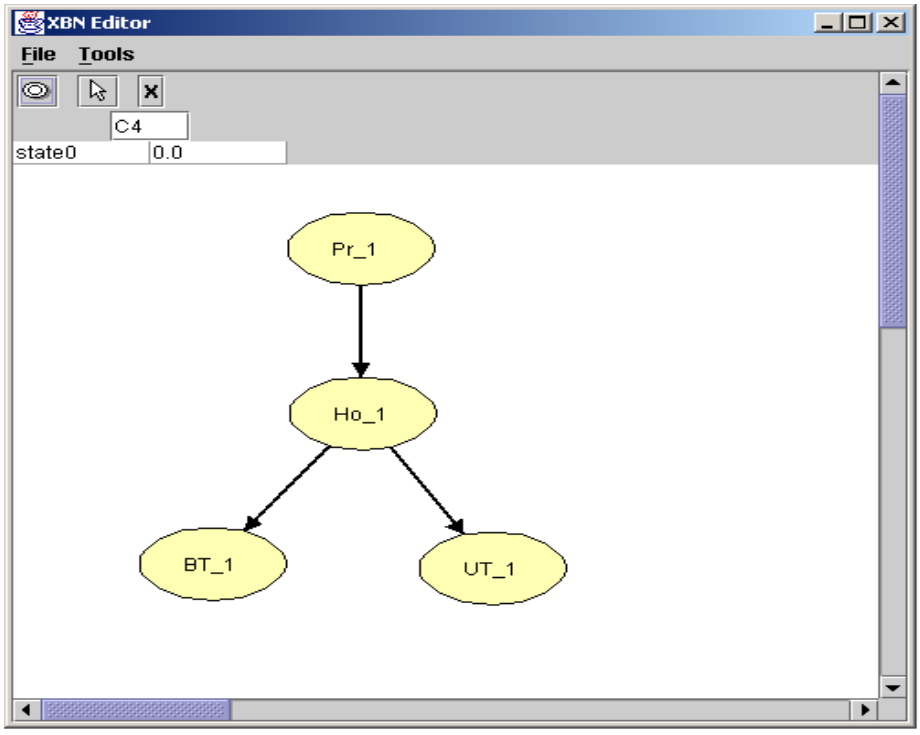


Figura 33 – Visualizzatore file XMLBIF formed. In figura è rappresentata Rete Pregnancy[74].

[74] Consultare il paragrafo in cui sono descritte le reti usate per la sperimentazione.

In effetti è possibile memorizzare una BN anche in un file .dat che è un semplice file di testo nel quale è descritta la rete in modo immediato (per tutti i nodi: nome variabile, stati, insieme dei padri e CPT).

File di output .dat

```
node:Pr_1
values: [yes, no]
parents: []
                        cpt
0.5981803639272145      Pr_1 = "yes"
0.4018196360727854      Pr_1 = "no"

node:Ho_1
values: [no, yes]
parents: [Pr_1]
                        cpt
0.09642379679144385     Ho_1 = "no" | Pr_1 = "yes"
0.9930348258706467      Ho_1 = "no" | Pr_1 = "no"
0.9035762032085561      Ho_1 = "yes" | Pr_1 = "yes"
0.006965174129353234    Ho_1 = "yes" | Pr_1 = "no"

node:UT_1
values: [yes, no]
parents: [Ho_1]
                        cpt
0.1076822061720289      UT_1 = "yes" | Ho_1 = "no"
0.8014719411223551      UT_1 = "yes" | Ho_1 = "yes"
0.8923177938279712      UT_1 = "no" | Ho_1 = "no"
0.19852805887764488     UT_1 = "no" | Ho_1 = "yes"

node:BT_1
values: [no, yes]
parents: [Ho_1]
                        cpt
0.9023856423725104      BT_1 = "no" | Ho_1 = "no"
0.30193192272309105     BT_1 = "no" | Ho_1 = "yes"
0.0976143576274896      BT_1 = "yes" | Ho_1 = "no"
0.6980680772769089      BT_1 = "yes" | Ho_1 = "yes"
```

4.3 LA SPERIMENTAZIONE

4.3.1 LE MODALITÀ DELLA SPERIMENTAZIONE

Nella prima fase del lavoro, è stato necessario testare gli algoritmi implementati confrontando i risultati ottenuti con quelli esibiti da versioni già funzionanti e disponibili in Internet: ad esempio, l'algoritmo PC è presente nell'applicativo Hugin (www.hugin.com), di cui abbiamo usato la versione Lite 6.0 (che presenta delle limitazioni rispetto a quella commerciale, specie per il numero di nodi da considerare), mentre l'algoritmo TPDA è alla base del tool BNPC (www.cs.ualberta.ca/~jcheng/bnpc.htm).

Dopo avere implementato i cinque esperti/algoritmi, l'algoritmo K2, K3, PC, TPDA e Bayesiano, abbiamo provveduto allo sviluppo del tool BayExpert.

Con la sperimentazione, utilizzando gli esperti su indicati, ci siamo posti i seguenti obiettivi:

a) indagare sull'importanza dell'ordinamento sia per algoritmi bayesiani, quali K2 e K3, che dependence – based;
b) esaminare la validità di un approccio multi-esperto allo structural learning;
c) vagliare la possibilità di impiego delle reti Bayesiane nell'ambito di un Intelligent Tutorial System.

Per il punto a), per perturbare l'ordine delle variabili dei database di riferimento abbiamo sviluppato una semplice interfaccia grafica.

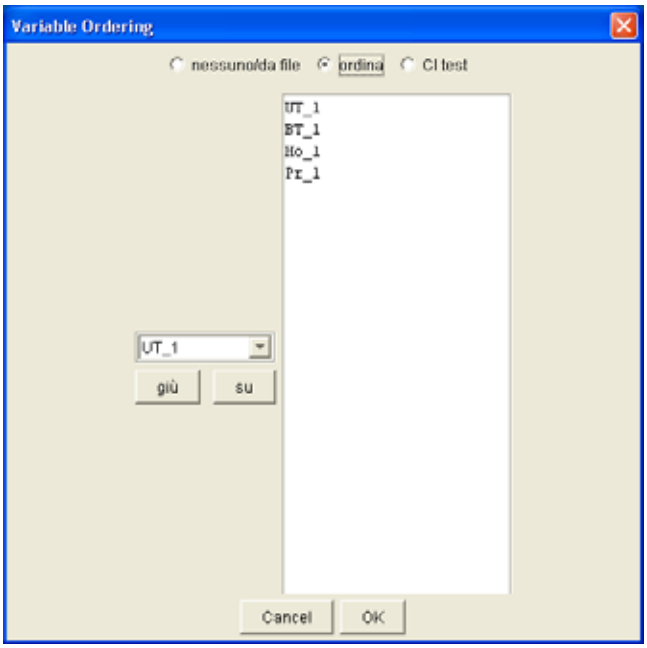
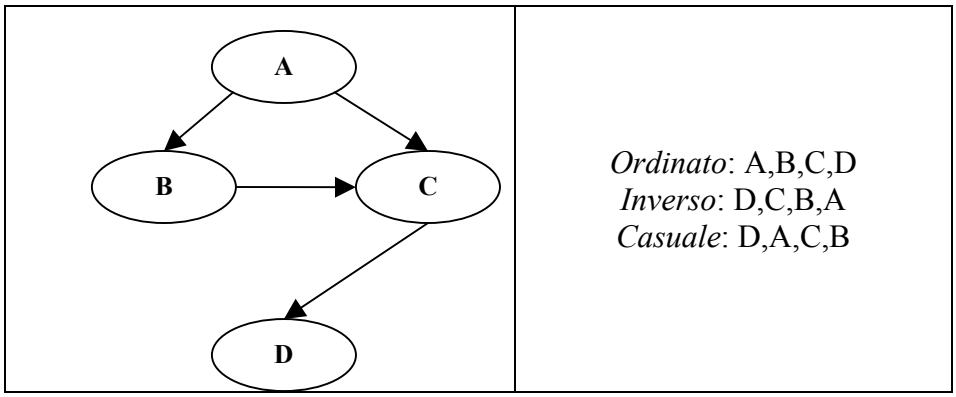


Figura 34 – Interfaccia per determinare un ordinamento per le variabili

Come da figura, è possibile lasciare la disposizione delle variabili inalterata (*nessuno/da file*), ordinare (*ordina*) i nodi secondo le proprie conoscenze agendo sui tasti su/giù o ricorrere ai test di indipendenza (*CI test*) per avere una possibile configurazione. Gli ordinamenti scelti per la sperimentazione sono denominati:

- "*Ordinato*": ordinamento topologico (della struttura) dai padri i figli; determinabile se si dispone di alcune informazioni a priori;
- "*Inverso*": dai figli ai padri;
- "*Casuale*": una delle *n*! disposizioni delle *n* variabili del dominio.



4.3.2 LA DESCRIZIONE DEI DATABASE

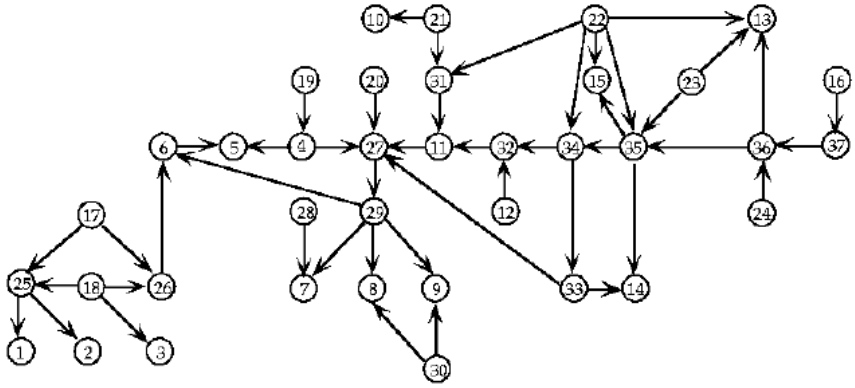
Nel paragrafo seguente sono descritte brevemente le Bayesian network esaminate durante la sperimentazione, citando le fonti sia dei data set di campioni che delle gold network (reti di riferimento create, in genere, grazie alla expert's knowledge).

4.3.2.1 LA RETE ALARM

ALARM è un acronimo di "A Logical Alarm Reduction Mechanism", questa rete Bayesiana è una delle gold network[75] più utilizzate in letteratura e modella un sistema diagnostico per il monitoraggio di un paziente in sala operatoria. Il dominio ALARM, studiato dal Beinlich [BEI89], è caratterizzato da 8 variabili "diagnosi", 16 "osservazioni" e 13 variabili intermedie per un totale di 37 nodi e 46 archi (è una *sparse network* perché non vi sono molti legami rispetto al numero di variabili).
Gli studi su questo dominio sono stati condotti con gli algoritmi di Structural Learning, quali TPDA e K2, in [BEI89][COO92][CHE97].

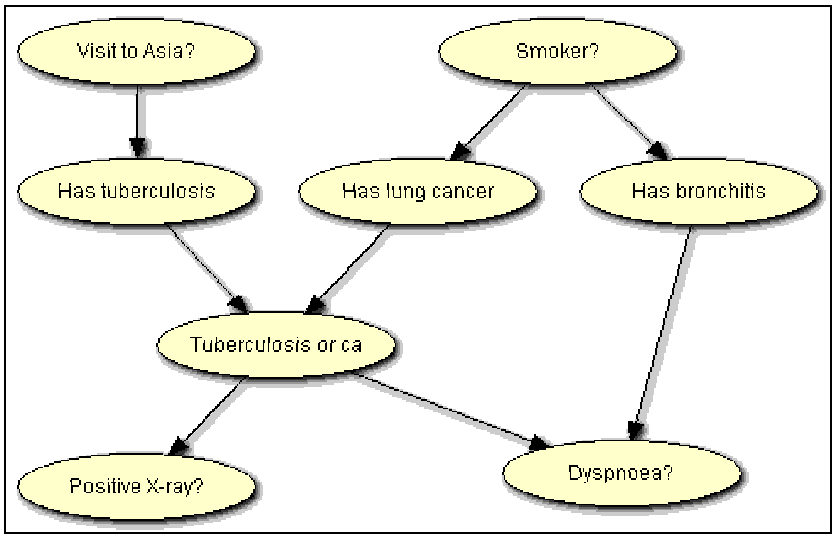
Il data set utilizzato per la rete ALARM è quello allegato al tool Belief Network Power Constructor – BNPC -, sviluppato da Cheng, J., Bell, D.A., Liu, W. – un software disponibile, gratuitamente, all'url www.cs.ualberta.ca/~jcheng/bnpc.htm. I campioni sono 10000 e le variabili **non** sono ordinate secondo la topologia della rete, ma in modo casuale.

[75] Reti di riferimento per il testing.



1 – central venous pressure
2 – pulmonary capillary wedge pressure
3 – history of left ventricular failure
4 – total peripheral resistance
5 – blood pressure
6 – cardiac output
7 – heart rate obtained from blood pressure monitor
8 – heart rate obtained from electrocardiogram
9 – heart rate obtained from oximeter
10 – pulmonary artery pressure
11 – arterial-blood oxygen saturation
12 – fraction of oxygen in inspired gas
13 – ventilation pressure
14 – carbon-dioxide content of expired gas
15 – minute volume, measured
16 – minute volume, calculated
17 – hypovolemia
18 – left-ventricular failure
19 – anaphylaxisr
20 – insufficient anesthesia or analgesia
21 – pulmonary embolus
22 – intubation status
23 – kinked ventilation tube
24 – disconnected ventilation tube
25 – left-ventricular end-diastolic volume
26 – stroke volume
27 – catecholamine level
28 – error in heart rate reading due to low cardiac output
29 – true heart rate
30 – error in heart rate reading due to electrocautery device
31 – shunt
32 – pulmonary-artery oxygen saturation
33 – arterial carbon-dioxide content
34. – alveolar ventilation
35 – pulmonary ventilation
36 – ventilation measured at endotracheal tube
37 – minute ventilation measured at the ventilator

4.3.2.2 LA RETE ASIA

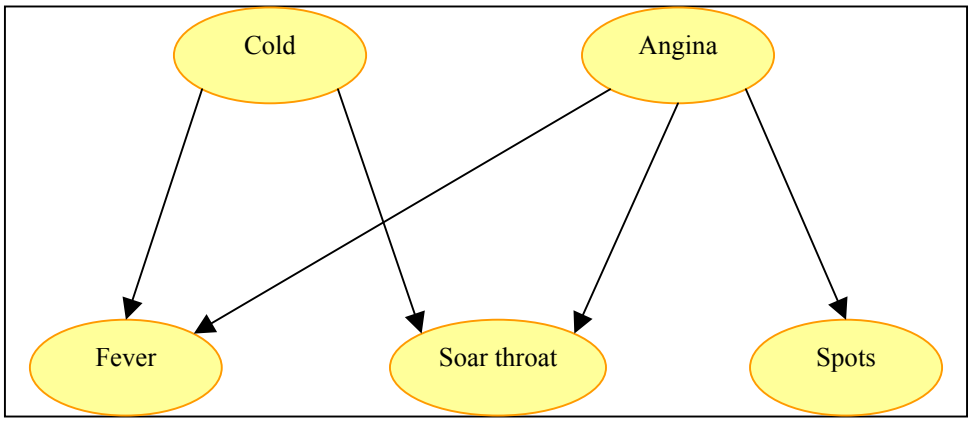


E' una Bayesian network di piccole dimensioni che non codifica un dominio reale bensì un problema medico fittizio con cui inferire sulla possibilità che un paziente abbia la tubercolosi, il cancro ai polmoni (lung cancer) o la bronchite in base alla diagnostica offerta da radiografie e presenza di dispnea (difficoltà respiratoria) e a cause aggravanti quali il fumo o una recente visita nel continente asiatico. Difatti, è noto che il fumo è causa sia della bronchite che del tumore ai polmoni; un viaggio in Asia può, invece, aumentare la possibilità di contrarre la tubercolosi. La rete Asia, in letteratura nota anche come rete "Chest Clinic", è stata concepita dal Lauritzen nel 1988 [LAU88]. Numerose sono le applicazioni che adoperano la rete Asia come gold network, ad esempio l'articolo di Cheng [CHE97].
Il data set utilizzato, fornito dal gruppo di lavoro Probabilistic-Reasoning - http://www.kddresearch.org/Groups/Probabilistic-Reasoning/k2.html, consta di 5000 campioni con le variabili già ordinate secondo la topologia della gold network.

Nome variabile	**Valori assunti (gold network)**	**Valori assunti (nel database)**
Visit to Asia?	yes, no	a1,a2
Smoker?	yes, no	f1,f2
Has Tubercolosis	yes, no	b1,b2
Has lung cancer	yes, no	e1,e2
Tubercolosis or lung cancer (either)	yes,no	c1,c2
Has bronchitis	yes, no	g1,g2
Positive X-ray	yes, no	d1,d2
Dysponea	yes, no	h1,h2

4.3.2.3 LA RETE ANGINA

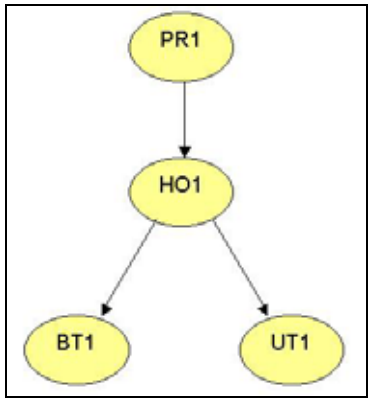
Descrizione del dominio - Il mal di gola (Soar Throat) può essere sintomo dell'inizio di un raffreddore (Cold) o il sintomo di un'Angina (infiammazione alle tonsille). La misura della febbre (Fever) e la verifica della presenza di puntini in gola (Spots) permette di acquisire una maggiore conoscenza sulla possibile causa. I dati su questa rete, citata in [FIN99], sono presenti al sito http://www.cs.auc.dk/~marta/datamine.htm e sono circa 10000 campioni (le variabili sono disposte dai nodi figli ai padri – ordinamento inverso).



Nome variabile	Valori assunti (anche nel database)
Cold	yes, no
Angina	mild, severe, no
Fever	low, high, no
Soar throat	yes, no
Spots	yes,no

4.3.2.4 LA RETE PREGNANCY

Il dominio codificato da questa BN, che ha solo uno scopo esemplificativo, consente di prevedere la riuscita dell'inseminazione di una mucca in un'azienda agricola. Infatti, sei settimane dopo l'inseminazione, l'analisi del sangue (Blood Test – BT) e l'analisi delle urine (Urine Test – UT) consentono di determinare se una mucca sia gravida. I risultati delle analisi del sangue e delle urine sono condizionati dallo stato ormonale (HO) che influenza la riuscita della fecondazione.

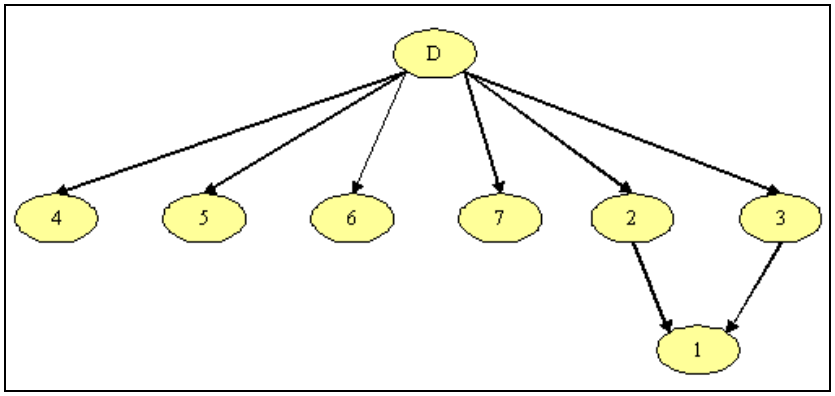


La rete Pregnancy è menzionata in [JEN96]. Il data set, di 10000 campioni, è disponibile all'indirizzo http://www.cs.auc.dk/~marta/datamine.htm. Le variabili del dominio sono memorizzate nel database in ordine inverso rispetto alla topologia della rete (dai padri ai figli).

Nome variabile	Valori assunti
PR1	yes, no
HO1	yes, no
BT1	yes, no
UT1	yes, no

4.3.2.5 LA RETE LED

Una rete semplice, impiegata per testare l'algoritmo proposto in [SIN94], è quella denominata LED.

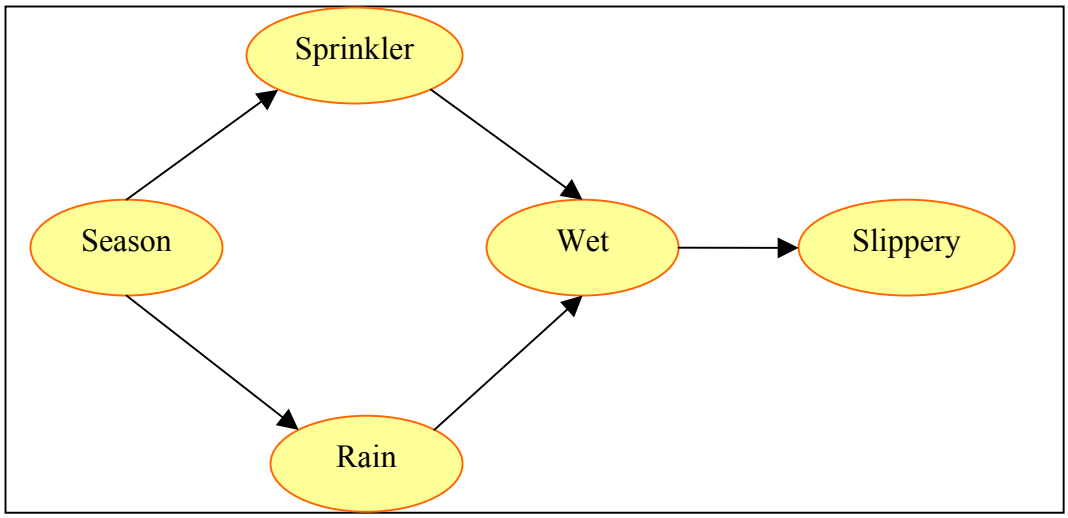


E' un esempio di classificatore, più che una BN vera e propria. Il LED display domain contiene 7 attributi booleani (variabili da 1 a 7 con stati 0 e 1) rappresentanti i 7 segmenti di un display a led, ed una variabile D con 10 stati ad indicare le 10 cifre decimali (da 0 a 9) visualizzabili. Il problema della classificazione è semplice a meno della presenza di rumore (ad esempio eventuali segmenti bruciati) nel dispositivo che altererebbe la visualizzazione.

I campioni del database sono 5000 generati come indicato all'url http://kdd.ics.uci.edu.

4.3.2.6 LA RETE SPRINKLER

A seconda della stagione dell'anno (Season) c'è una maggiore o minore di probabilità di pioggia. L'entità della pioggia (Rain) o l'annaffiatore (Sprinkler) rendono la strada bagnata (Wet) ed anche scivolosa (Slippery).



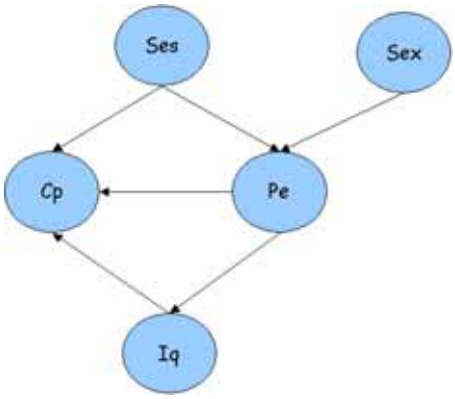
Questa rete bayesiana è talvolta presente, in letteratura, con una notazione diversa (cambiano i nomi delle variabili del dominio). In questo articolo consideriamo la rete Sprinkler illustrata in [PEA00].

I dati, 400 campioni, già ordinanti secondo la topologia, sono stati presi all' url http://www.kddresearch.org/Groups/Probabilistic-Reasoning/k2.html.

Nome variabile	**Valori assunti (anche nel database)**
Season	fall, winter, spring, summer
Rain	downpour, drizzle, steady, none
Sprinkler	on, off
Wet	wet, dry
Slippery	slippery, unslippery

4.3.2.7 LA RETE COLLEGE

Per la rete college non c'è una gold network di riferimento effettiva proprio perché nasce dagli studi sull'apprendimento di reti bayesiane condotti da Heckerman, Geiger, Chickering in [HEC94] su un database reale. I fattori che influenzano la scelta di uno studente nell'intraprendere la carriera universitaria costituiscono il dominio di questa BN: il sesso (SEX) dello studente e i suoi progetti sull'università (College Plans – CP), la condizione socioeconomica (SocioEconomic Status – SES), il quoziente di intelligenza (Intelligence Quotient - IQ), il condizionamento familiare (Parental Encorougement – PE).



Nome variabile	**Valori assunti (anche nel database)**
SEX	male, female
SES	low, lowermiddle, uppermiddle, high
IQ	low, lowermiddle, uppermiddle, high
PE	low, high
CP	yes, no

La rete rappresenta e lega tra loro, quindi, le cause che influenzano la scelta sul prosieguo degli studi dopo il liceo: come BN di riferimento si considera quella ottenuta da Heckerman nel suo tutorial. I dati, 10318, sono stati costruiti dalle informazioni presentate dallo stesso Heckerman.

4.3.2.8 L'ONTOLOGIA DEL CORSO FONDAMENTI DI INFORMATICA (CFI)

Uno degli obiettivi di questo lavoro è esaminare la possibilità di impiego delle reti Bayesiane come supporto ad un Intelligent Tutoring System (ITS) in un ambiente per l'e-learning. In lavori di tesi precedenti, è stato avviato il discorso e lo sviluppo di componenti da integrare in un ITS sia per metadare i contenuti di un corso e le caratteristiche di uno studente. In tale ottica, una BN consente di organizzare i contenuti di un corso in modo propedeutico e quindi rappresenterebbe un valido supporto per un ITS.

In particolare, siamo interessati a studiare le relazioni di propedeuticità tra i diversi argomenti che compongono un corso universitario di Fondamenti di Informatica (CFI) presso la Facoltà di Ingegneria Elettronica.

Il termine ontologia nel campo dell'informatica, in particolare nell'ambito dell'AI (Artificial Intelligence - per l'Information System (IS), il semantic web, knowledge-based system) e della Knowledge Engineering, indica un "engineering artefact"[76] costituito, come afferma anche il Neches, da un vocabolario, usato per descrivere una certa realtà, ed un insieme di assunzioni esplicite che riguardano il significato dei termini del vocabolario, espresse in forma di assunzioni di livello logico del primo ordine (relazioni). Gruber, nel contesto della condivisione della conoscenza[77] (Knowledge Sharing), definisce l'ontologia come una *specificazione di una concettualizzazione*, in quanto esplica tutte le possibili relazioni fra i concetti che appartengono ad un dominio: si intuisce come le reti Bayesiane rappresentino un valido riferimento per formalizzare un'ontologia. [GUA98][DES03]

Grazie alla consulenza di alcuni docenti (esperti), sono stati indicati i legami di propedeuticità tra i diversi argomenti trattati durante lo svolgimento del corso di Fondamenti di Informatica. L'ontologia proposta è in figura.

[76] Letteralmente: un manufatto/prodotto dell'ingegneria.
[77] Fra persone o agenti software.

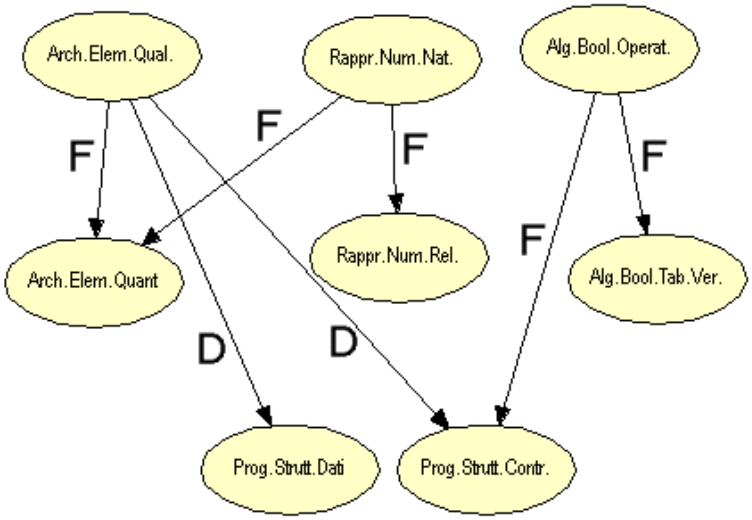


Figura 35 - Le relazioni di propedeuticità in un corso di Fondamenti di Informatica

Le lettere in corrispondenza degli archi indicano l'intensità del legame: D debole, F forte. La fonte di informazioni da cui apprendere la Rete Bayesiana è costituita da un database di risposte date da 231 studenti di un corso accademico di Fondamenti di Informatica.

Di seguito è riportato uno stralcio di questionario.

QUESTIONARIO

1) Con 4 bit nella rappresentazione degli interi in segno e modulo, si possono rappresentare i numeri:

[a] compresi tra – 8 e + 8 inclusi
[b] compresi tra – 7 e + 7 inclusi
[c] i positivi da 0 a 15

2) La rappresentazione in segno e modulo del numero decimale 5 è:

[a] 1101
[b] 0101
[c] 1010

3) La rappresentazione in complementi alla base del numero –5 su 4 bit è:

[a] 1011
[b] 1101
[c] 0101

Tali questionari *non* sono stati costruiti in previsione del loro futuro utilizzo come dati per l'apprendimento Bayesiano, e quindi hanno necessitato di un'opportuna elaborazione che li rendesse idonei a tale scopo.

Ciò ha comportato l'attuazione di una serie di scelte:

1) valutazione dell'omogeneità dei diversi questionari
2) modalità della valutazione della risposta alla singola domanda;
3) raggruppamento delle domande in argomenti;
4) modalità della valutazione della risposta all'argomento nel suo complesso.

Ogni nodo della rete (cioè ogni variabile aleatoria), rappresenta quindi la probabilità che il generico discente risponda correttamente e quindi conosca l'argomento associato allo stesso nodo.

In questa rappresentazione supporremo che ogni nodo possa assumere solo due stati:

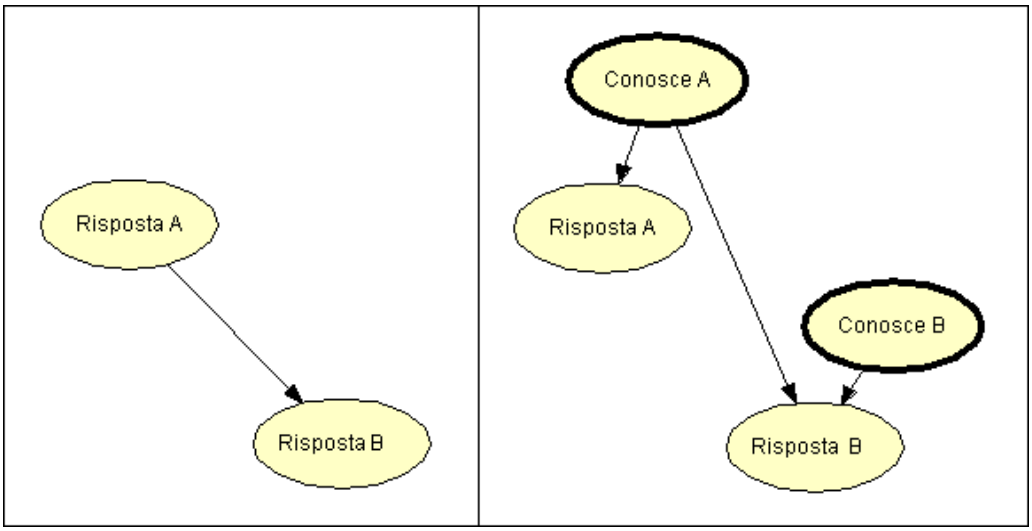
- stato 'Yes': conoscenza completa dell'argomento,
- stato 'Not': totale ignoranza dell'argomento.

Come osservato in un lavoro di tesi precedente[78], sarebbe più corretto aggiungere un nodo che identifichi l'evento "conoscenza dell'argomento". In effetti nell'ontologia illustrata sopra, un nodo indica se: "lo studente risponde bene/male alla domanda sull'argomento", ignorando le possibili cause come la non conoscenza dell'argomento.

Lo studente sbaglierà la risposta se non conosce uno degli argomenti propedeutici ovvero sbaglierà, molto probabilmente, se ha risposto in maniera errata ad uno degli argomenti propedeutici.

Anziché avere un'implicita doppia interpretazione -"lo studente risponde bene/male alla domanda sull'argomento" / "lo studente conosce/non conosce l'argomento", sarebbe opportuno distinguere i due eventi con due variabili aleatorie distinte come in figura.

[78] "Reti Bayesiane per il data mining in ambienti di e-learning" di Raffaele Albanese (Università degli Studi di Salerno, Tesi di laurea in Sistemi di Elaborazione, anno accademico 2001-2002).



I nodi "Conosce A" e "Conosce B" sono nascosti, cioè in termini di variabili aleatorie, gli stati assunti da tali variabili non sono mai osservati: siamo di fronte a dei *missing-values*. Sebbene esistano metodi per il trattamento dei missing-values, essi sono poco efficaci per il caso (ed è il nostro) di missing-values sistematici, cioè quando tali variabili non possono essere *mai* osservate. La natura di tale inosservabilità è dovuta al fatto che non siamo in grado di sapere se lo studente conosce l'argomento X se non attraverso le risposte date dallo stesso ai questionari proposti; per ottenere una simile informazione insieme alle domande relative all'argomento X, dovrebbe apparire una domanda del tipo: "Conosci l'argomento X ?", che per ovvi motivi è improponibile in sede di esame. Però, organizzando in maniera opportuna il questionario, forse sarebbe possibile ottenere delle indicazioni sul grado di preparazione dello studente attribuendo, per esempio, un punteggio ad una non-risposta onde condizionare l'esaminando a rispondere solo se è convinto.

Oltre al database su menzionato, sono stati esaminati altri tre data set relativi al corso di Fondamenti di Informatica svolto presso la facoltà di Ingegneria Elettronica e la facoltà di Lingue e Letterature straniere. Purtroppo, anche in questo caso i dati a disposizione sono pochi per cui gli algoritmi di structural learning diventano poco attendibili. Ulteriori dettagli sui risultati ottenuti sono presentati nell'allegato.

4.4 I RISULTATI

Di seguito sono riportati i risultati della sperimentazione. Poiché i dati sono numerosi, seguono solo quelli più importanti sia perché spesso riferiti in letteratura (rete ASIA e ALARM) sia per la semplicità della rappresentazione grafica (PREGNANCY). Dove la visualizzazione della rete complicasse la raffigurazione, ad esempio per l'elevato numero di nodi (ALARM), si è preferito usare una tabella. In tal caso, per ogni algoritmo e rispetto a diversi parametri, nella prima colonna della tabella, in grassetto, sono indicati i nomi dei nodi del dominio e nell'ultima colonna, in grassetto e corsivo, è riportata la sequenza corretta dei padri in base alla gold network.

	Algoritmo e Parametro	**…**	**Algoritmo e Parametro**	
nodo 1				***padri nodo 1 nella gold network***
…				**…**
nodo i				***padri nodo i nella gold network***
…				**…**
nodo n				***padri nodo n nella gold network***

Figura 36 – Formato tabella

Quindi, per ogni singolo database sono presentati:

- la gold network ed un'eventuale legenda;
- per le reti ALARM, ASIA e PREGNANCY una rappresentazione del multi-esperto;
- per l'ontologia del "Corso di Fondamenti di Informatica" (CFI) i risultati sono esposti in forma di tabella.

L'approccio multi-esperto è stato valutato considerando i seguenti parametri:

☑ Algoritmo PC - Il livello di significatività del test 0.3; è un valore sufficiente per le reti Asia, Alarm, Prgnancy perché i data set hanno dai 5000 ai 10000 record. Il valore non è indicato per l'ontologia CFI per la quale i campioni sono 231; d'altronde valori elevati del livello di fiducia del test evidenziavano molti legami inesistenti, quindi abbiamo comunque considerato soglia 0.3.

- ☑ Algoritmo TPDA - La soglia per la determinazione dell'indipendenza è quella di default, indicata dagli ideatori dell'algoritmo, 0.01.
- ☑ Algoritmo Bayesiano - Il numero di iterazioni è N = 20 mentre $\alpha = 3$ in quanto siamo partiti da un empty graph (completa ignoranza a priori). Per la rete ALARM il numero di iterazioni designato è 200 in quanto, per N = 20, dall'output abbiamo riscontrato che l'algoritmo terminava non per il raggiungimento di una convergenza bensì per avere superato il numero di iterazioni.
- ☑ K2 e K3 - per le reti ASIA, PREGNANCY e CFI sono presentati i risultati anche al variare dell'ordinamento delle variabili (ordinato, inverso, casuale) mentre per la rete ALARM si è considerato direttamente un ordinamento casuale.

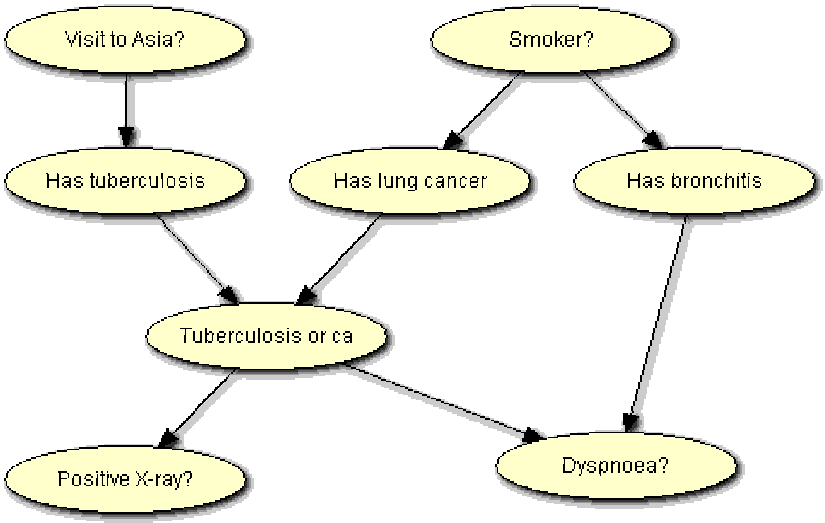
Nella sperimentazione sono stati considerati altri database; i risultati in forma di tabelle, sono consultabili in un allegato. Nell'ultimo capitolo saranno riassunti i punti più rilevanti della sperimentazione.

Rete	nel data set	Ordinato	Inverso	Casuale
Alarm	Casuale	-	-	-
Angina	Inverso	Angina|Cold|Fever| Soar| Spots|	Fever|Spots|Sore|Angina|Cold	Fever|Sore|Angina|Cold|Spots
Asia	Ordinato	visit to Asia?|smoking?|tuberculosis?|lung cancer?| tub. or lung cancer?|bronchitis?|positive X-ray?|dyspnoea?	dysponea?| positive X-ray?| tub. or lung cancer?| bronchitis?| lung cancer?| tuberculosis?| smoking? | visit to Asia?|	dysponea?| tub. or lung cancer?| positive X-ray?| visit to Asia?| lung cancer?| tuberculosis?| bronchitis?| smoking?
College	Ordinato[79]	Sex|Ses|Iq|Pe|Cp	Cp|Iq|Pe|Ses|Sex	Sex|Cp|Iq|Pe|Ses
Led	Inverso	D|7|6|5|4|2|3|1	1|2|3|4|5|6|7|D	1|7|6|5|D|4|2|3
ontCorretta	Casuale	aeql|rnn|abo|psd|aeqt|rnr|psc|abtv	abtv|psc|rnr|aeqt|psd|abo|rnn|aeql	aeql|aeqt|rnn|rnr|psd|abo|abtv|psc
ontHardware	Ordinato	hardware|processore|memoria| dispositivi	processore|memoria|dispositivi| hardware	processore|hardware|memoria| dispositivi
ontSoftware	Ordinato	software|s_operativo|web|word| p_elettronica|excel	word|p_elettronica|excel|software| s_operativo|web|	word|web|s_operativo|software|excel| p-elettronica
Pregnancy	Inverso	PR1|HO1|UT1|BT1	UT_1|BT_1|Ho_1|Pr_1	UT1,PR1,BT1,HO1
Sprinkler	Ordinato	Season|Sprinkler|Rain|Wet|Slippery	Slippery|Wet|Rain|Sprinkler|Season	Rain,Slippery,Wet,Sprinkler,Season

Tabella 3 - Ordinamenti usati durante la sperimentazione

[79] Per essere più precisi, Pe è padre di Iq quindi nell'*ordinato* andrebbe prima: poiché la rete di riferimento non è una gold network abbiamo comunque lasciato inalterata la disposizione.

RETE ASIA

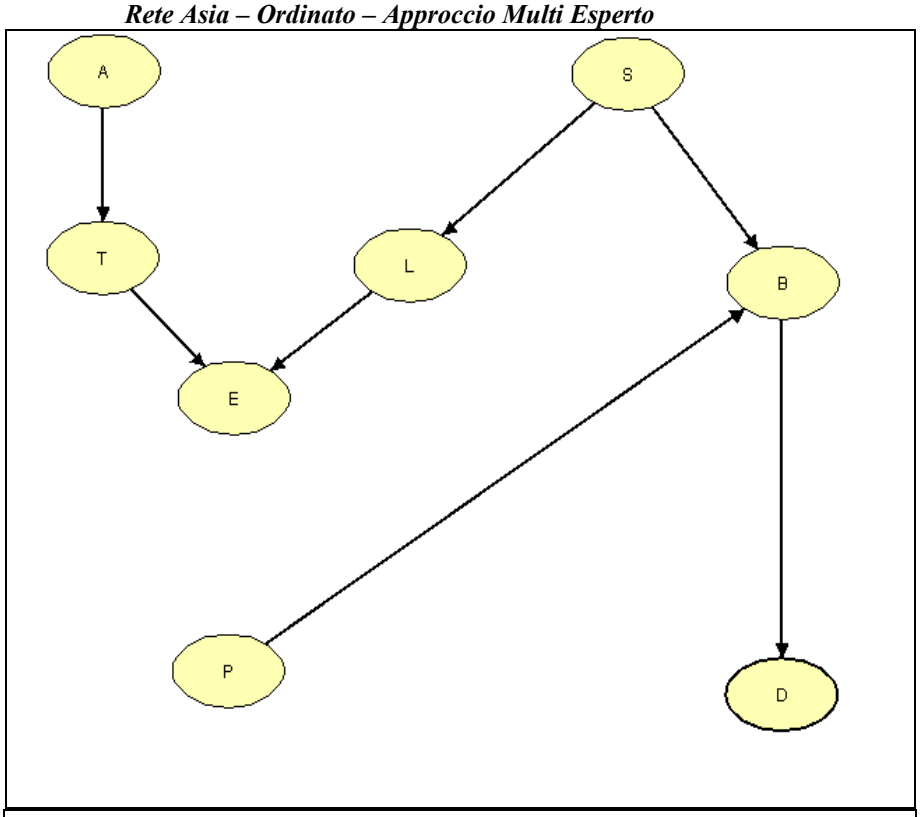


Legenda

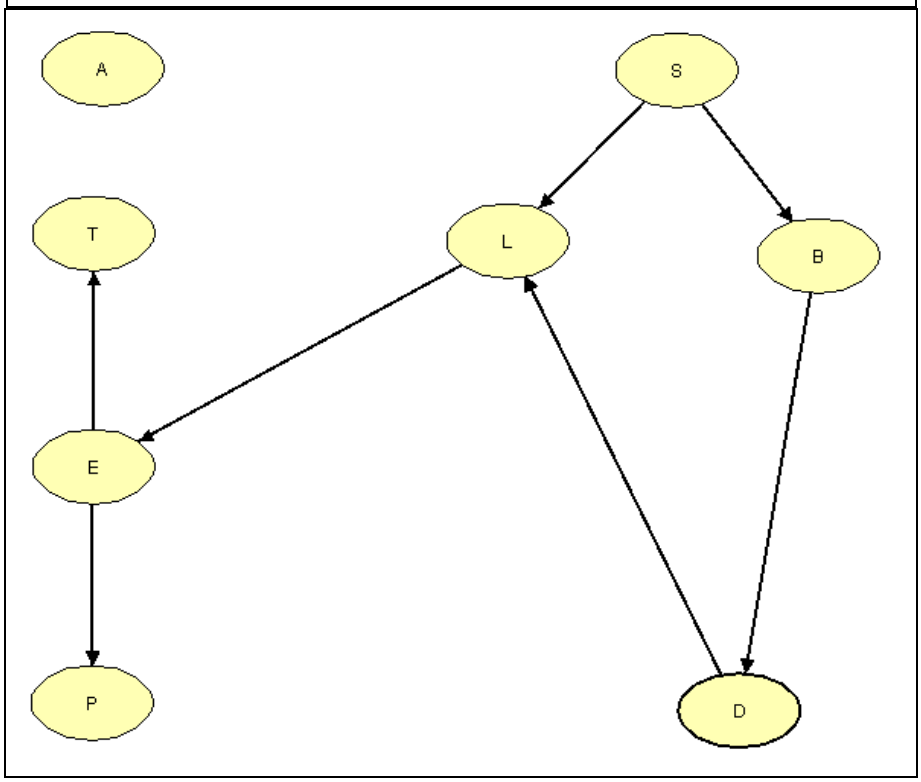
Visit to Asia?	→ **A**
Smoker?	→ **S**
Has Tubercolosis	→ **T**
Has Bronchitis	→**B**
Has Lung Cancer	→ **L**
Tuercolosis or Lung Cancer	→ **E**
Positive P – Ray?	→ **P**
Dysponea?	→ **D**

Rete Asia – Ordinato – Approccio Multi Esperto

PC
0.3

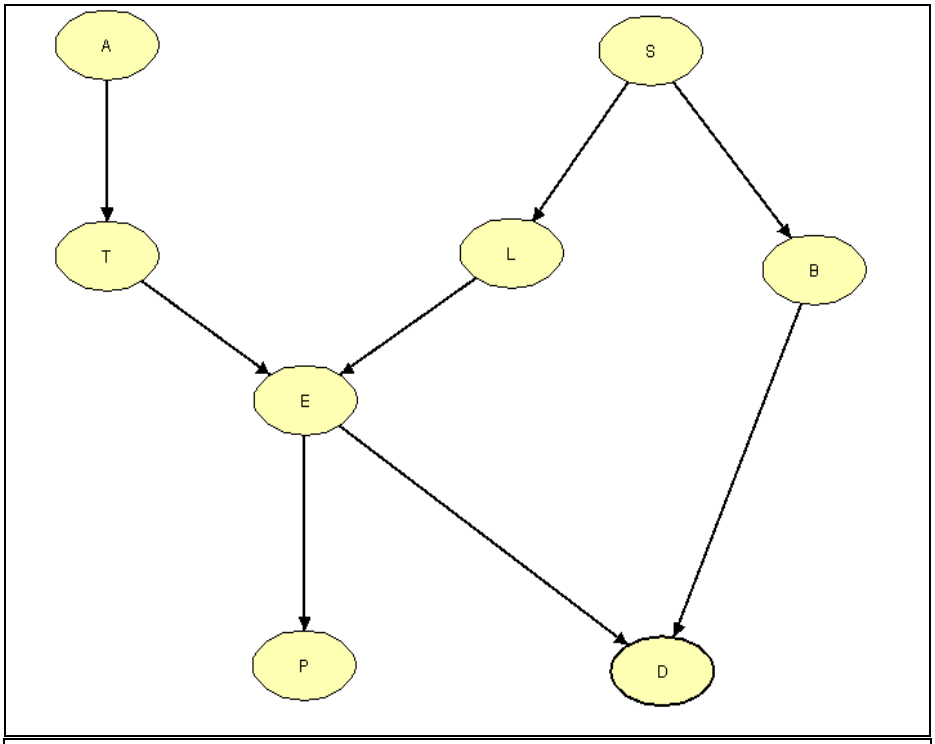


TPDA
0.01

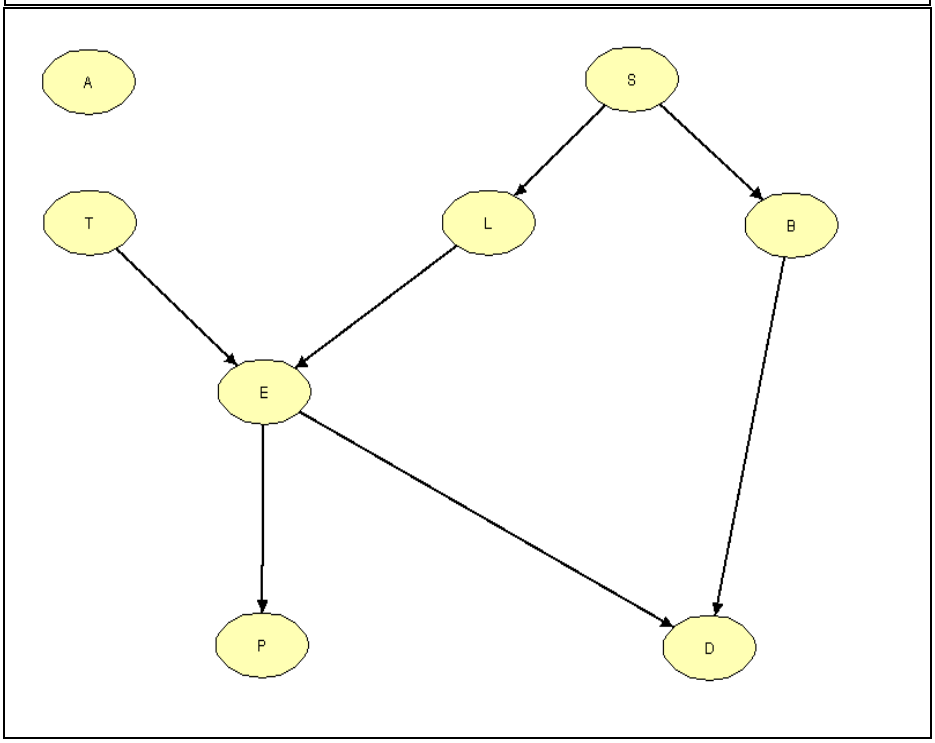


Rete Asia – Ordinato – Approccio Multi Esperto

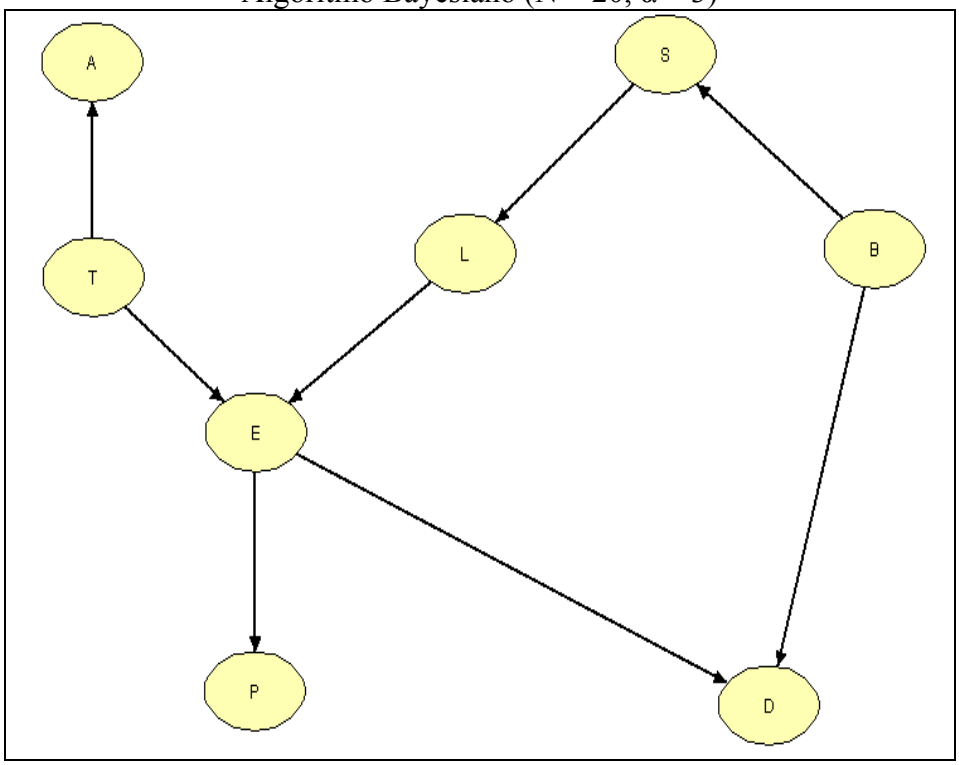
K2



K3



Algoritmo Bayesiano (N = 20, α = 3)



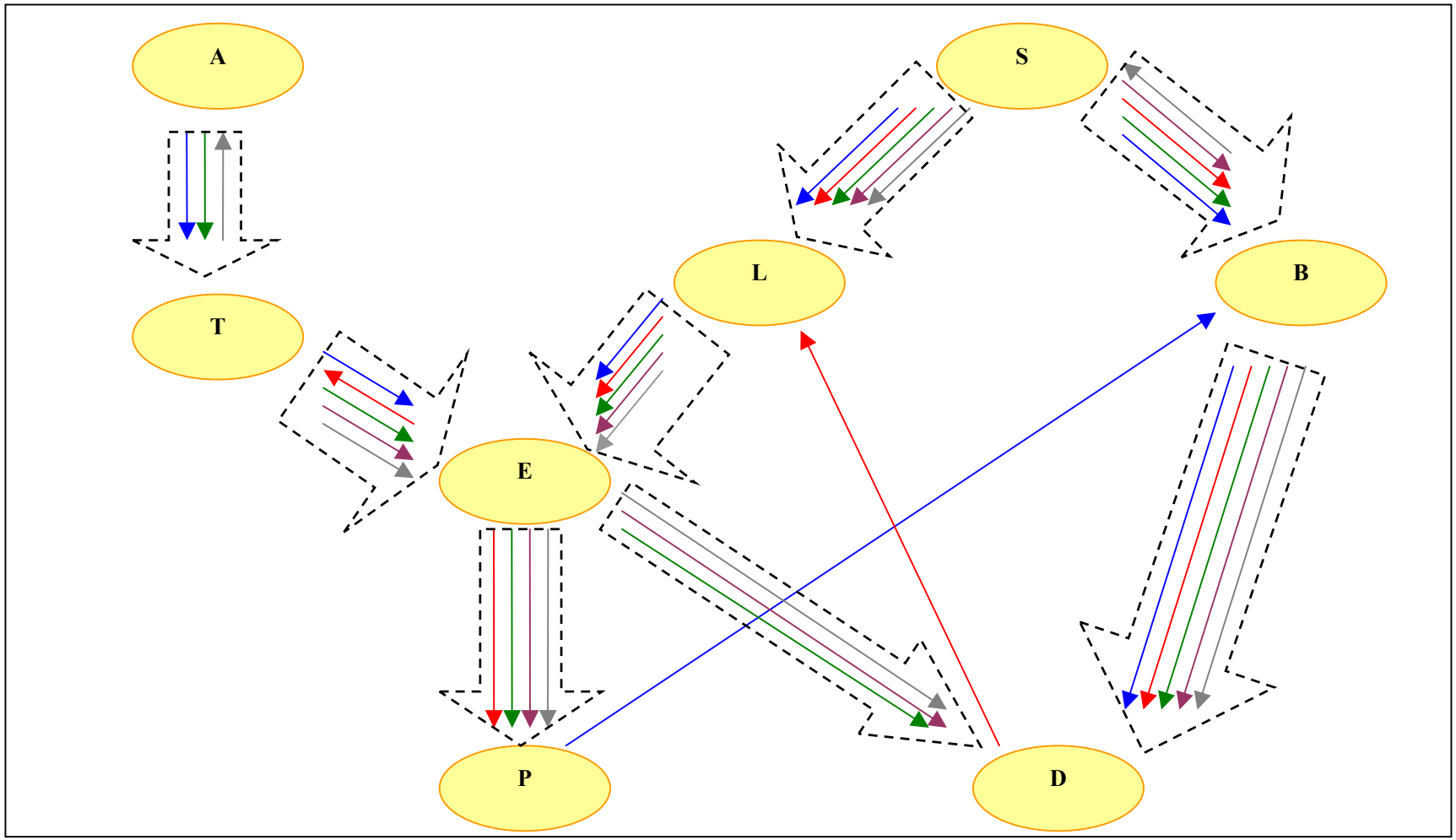
Rete Asia – Ordinato – Approccio Multi Esperto

Nome rete	Asia
Numero di nodi	8
Numero di archi	8
Descrizione dei Nodi	
Nome Nodi	**Stati**
visit to Asia?	2
smoking?	2
tuberculosis?	2
lung cancer?	2
either tub. or lung cancer?	2
bronchitis?	2
positive X-ray?	2
dyspnoea?	2

Database di riferimento	
Nome Database	Asia5000.dat
Numero Campioni	5000
Fonte	http://www.kddresearch.org/Groups/Probabilistic-Reasoning/k2.html
Ordinato (dai padri ai figli)	si

	Parametro					
Algoritmi di ricostruzione		**Archi Corretti**	**Archi Mancanti**	**Archi Aggiunti**	**Orientamenti Corretti**	**Orientamenti Errati**
Bayesiano	N = 20, α = 3	8	0	0	6	2
K2		8	0	0	8	0
K3		7	1	0	7	0
PC	0,3	6	2	1	6	0
TPDA	0,01	6	2	1	5	1
Multi-esperto a Maggioranza (tratteggio)		8	0	0	8	0

Rete Asia – Ordinato – Approccio Multi Esperto



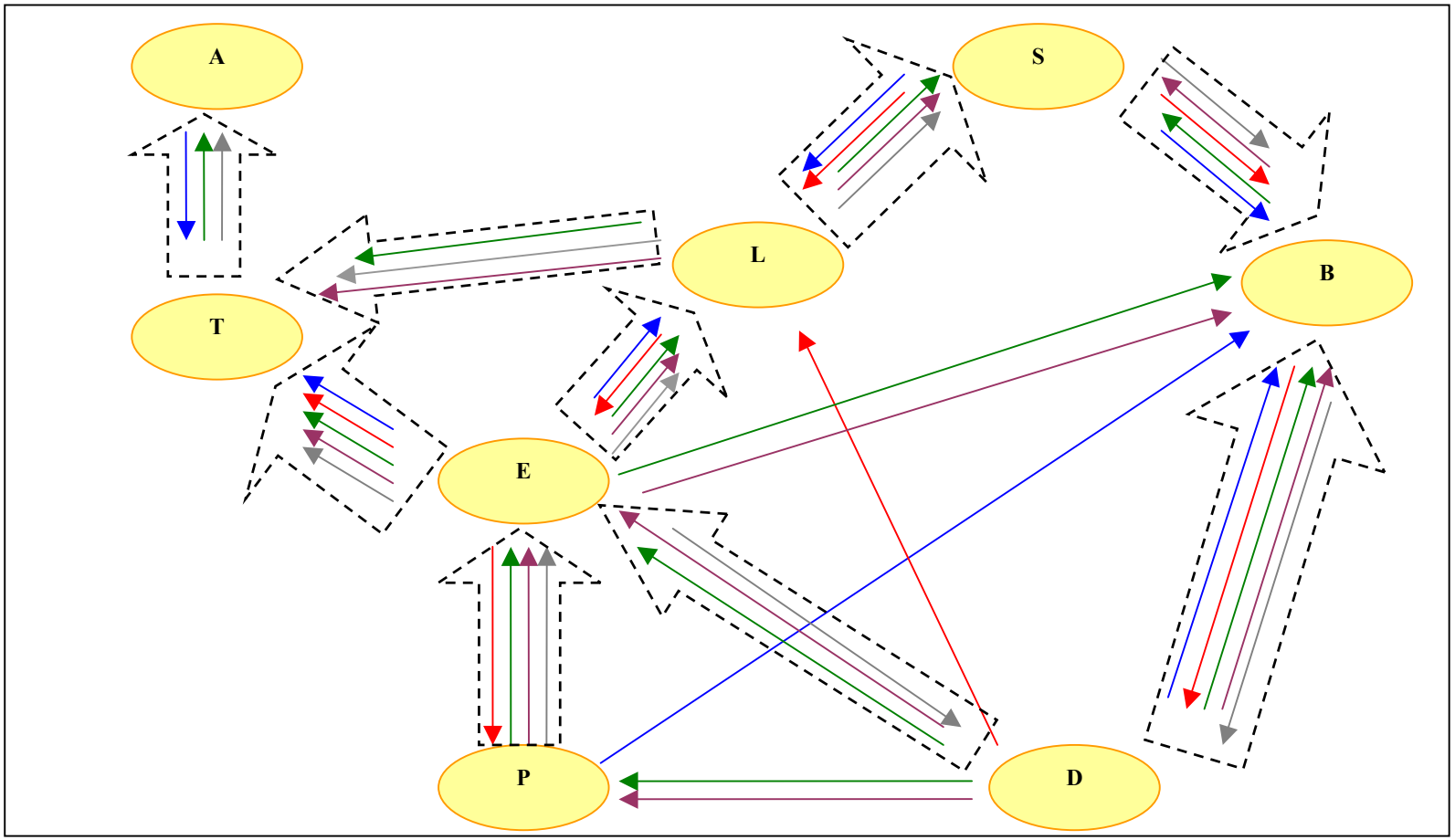
Rete Asia – Inverso – Approccio Multi Esperto

Nome rete	Asia
Numero di nodi	8
Numero di archi	8
Descrizione dei Nodi	
Nome Nodi	**Stati**
visit to Asia?	2
smoking?	2
tuberculosis?	2
lung cancer?	2
either tub. or lung cancer?	2
bronchitis?	2
positive X-ray?	2
dyspnoea?	2

Database di riferimento	
Nome Database	Asia5000.dat
Numero Campioni	5000
Fonte	http://www.kddresearch.org/Groups/Probabilistic-Reasoning/k2.html
Ordinato (dai padri ai figli)	si

	Parametro					
Algoritmi di ricostruzione		**Archi Corretti**	**Archi Mancanti**	**Archi Aggiunti**	**Orientamenti Corretti**	**Orientamenti Errati**
Bayesiano	N = 20, α = 3	8	0	1	3	5
K2		8	0	3	0	8
K3		7	1	3	0	7
PC	0,3	6	2	1	3	3
TPDA	0,01	6	2	1	5	1
Multi-esperto a Maggioranza (tratteggio)		8	0	1	1	7

Rete Asia – Inverso – Approccio Multi Esperto



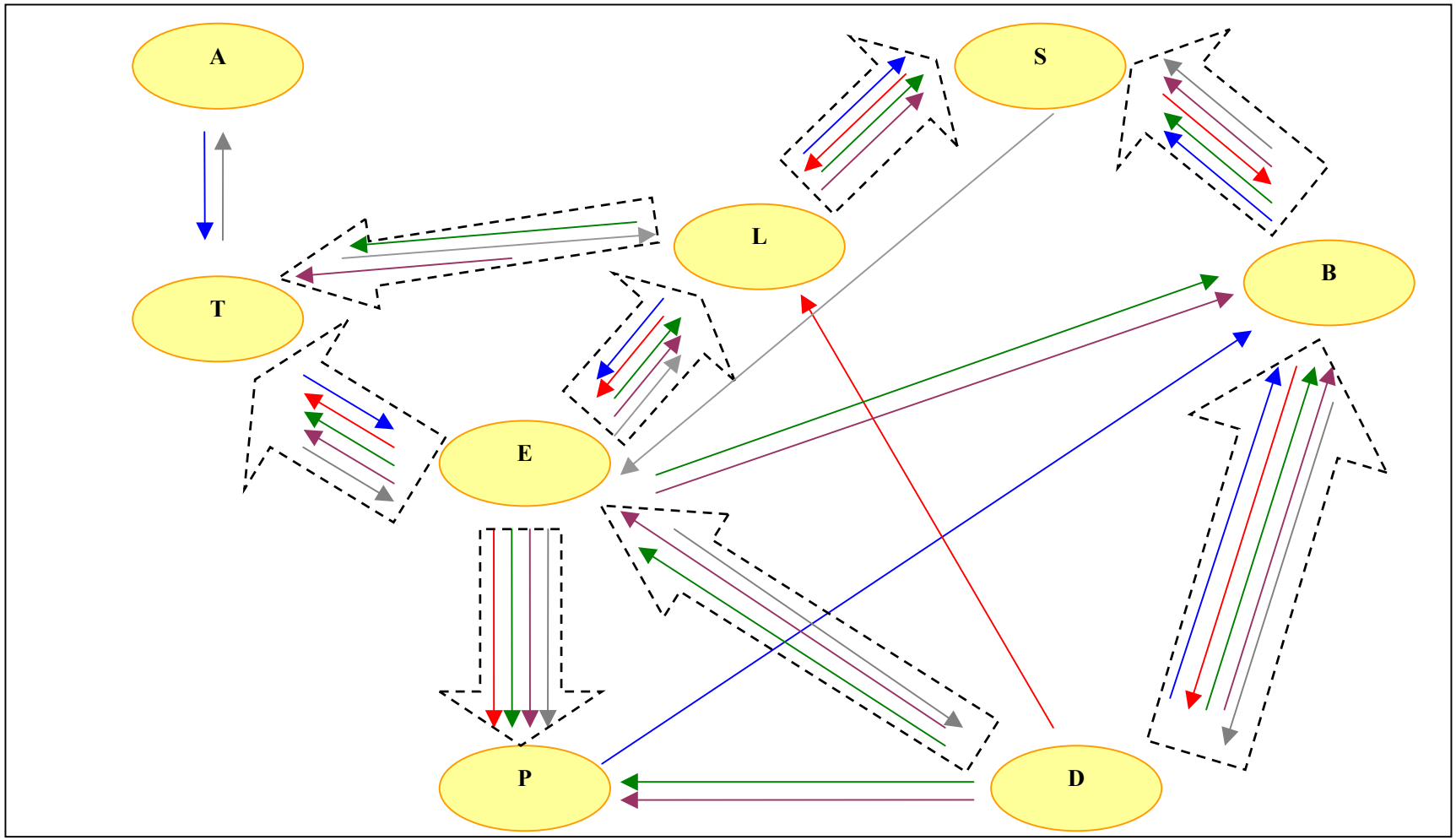
Rete Asia – Casuale – Approccio Multi Esperto

Nome rete	Asia
Numero di nodi	8
Numero di archi	8
Descrizione dei Nodi	
Nome Nodi	**Stati**
visit to Asia?	2
smoking?	2
tuberculosis?	2
lung cancer?	2
either tub. or lung cancer?	2
bronchitis?	2
positive X-ray?	2
dyspnoea?	2

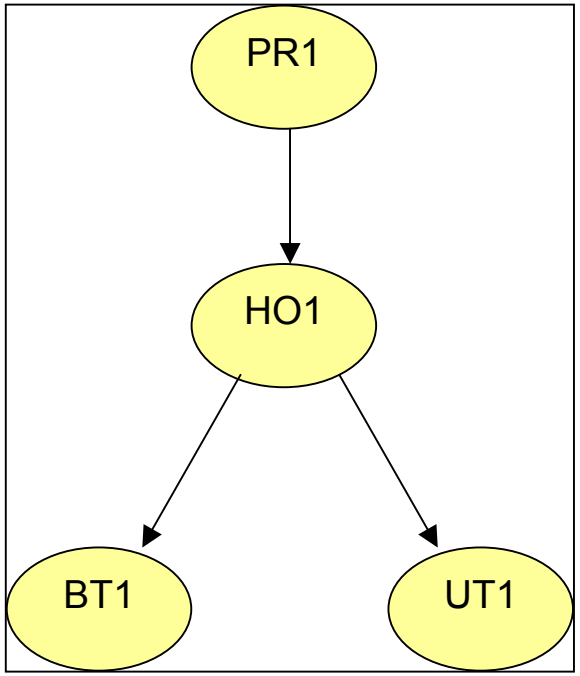
Database di riferimento	
Nome Database	Asia5000.dat
Numero Campioni	5000
Fonte	http://www.kddresearch.org/Groups/Probabilistic-Reasoning/k2.html
Ordinato (dai padri ai figli)	si

Algoritmi di ricostruzione	Parametro	Archi Corretti	Archi Mancanti	Archi Aggiunti	Orientamenti Corretti	Orientamenti Errati
Bayesiano	N = 20, α = 3	7	1	2	4	3
K2		7	1	2	1	6
K3		7	1	2	1	6
PC	0,3	6	2	1	3	3
TPDA	0,01	6	2	1	5	1
Multi-esperto a Maggioranza (tratteggio)		7	1	1	1	6

Rete Asia – Casuale – Approccio Multi Esperto

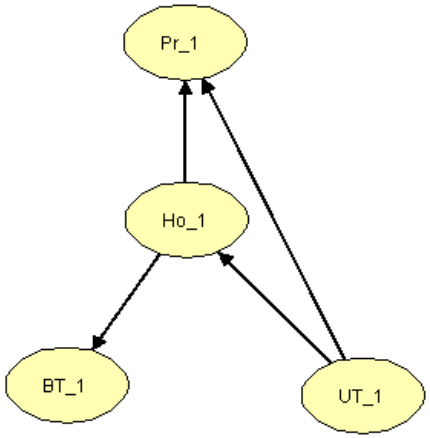


RETE PREGNANCY

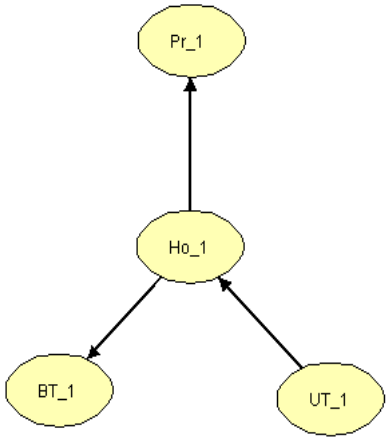


Rete Pregnancy – Ordinato – Approccio Multi Esperto

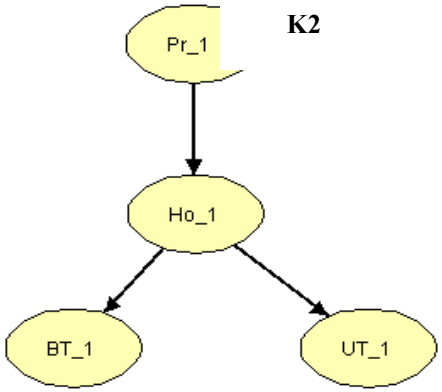
PC (0.3)



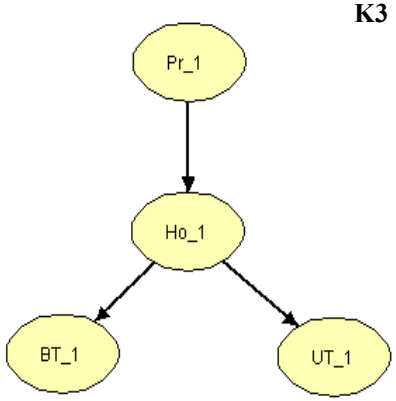
TPDA (0.01)



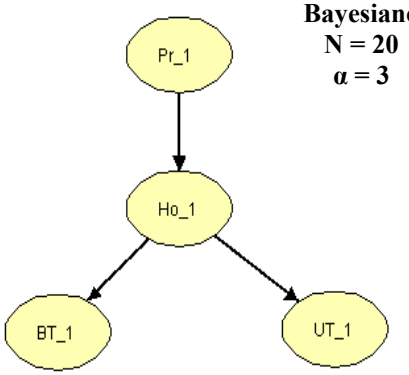
K2



K3



Bayesiano N = 20 α = 3



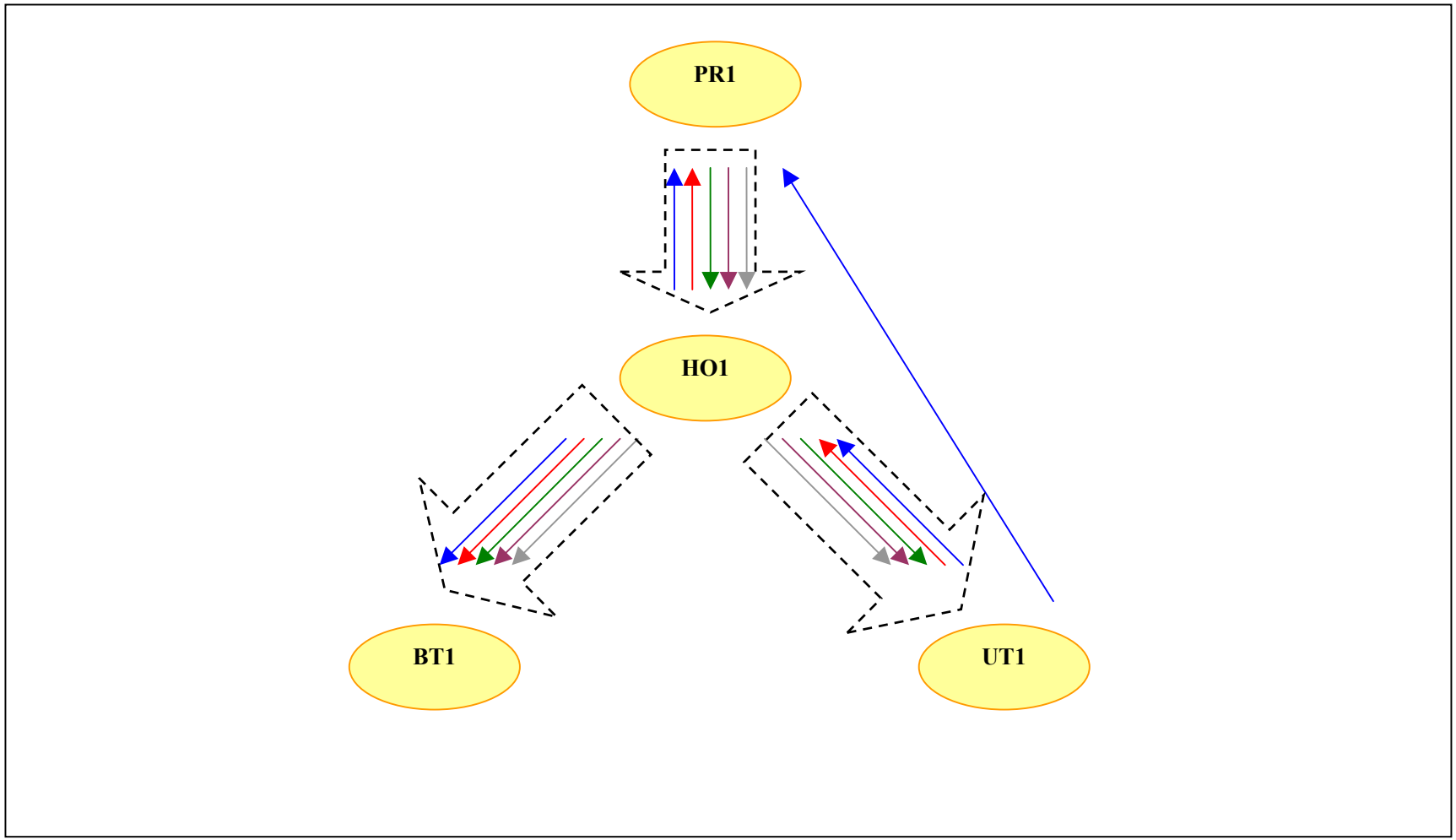
Rete Pregnancy – Ordinato – Approccio Multi Esperto

Nome rete	Pregnancy
Numero di nodi	4
Numero di archi	3
Descrizione dei Nodi	
Nome Nodi	Stati
UT_1	2
BT_1	2
Ho_1	2
Pr_1	2

Database di riferimento	
Nome Database	pregnancy.dat
Numero Campioni	10000
Fonte	http://www.cs.auc.dk/~marta/datamine.htm
Ordinato (dai padri ai figli)	no

Algoritmi di ricostruzione		Archi Corretti	Archi Mancanti	Archi Aggiunti	Orientamenti Corretti	Orientamenti Errati
Bayesiano	N = 20, α = 3	3	0	0	3	0
K2		3	0	0	3	0
K3		3	0	0	3	0
PC	0.3	3	0	1	1	2
TPDA	0.01	3	0	0	1	2
Multi-esperto a Maggioranza (tratteggio)	3	3	0	0	3	0

Rete Pregnancy – Ordinato – Approccio Multi Esperto



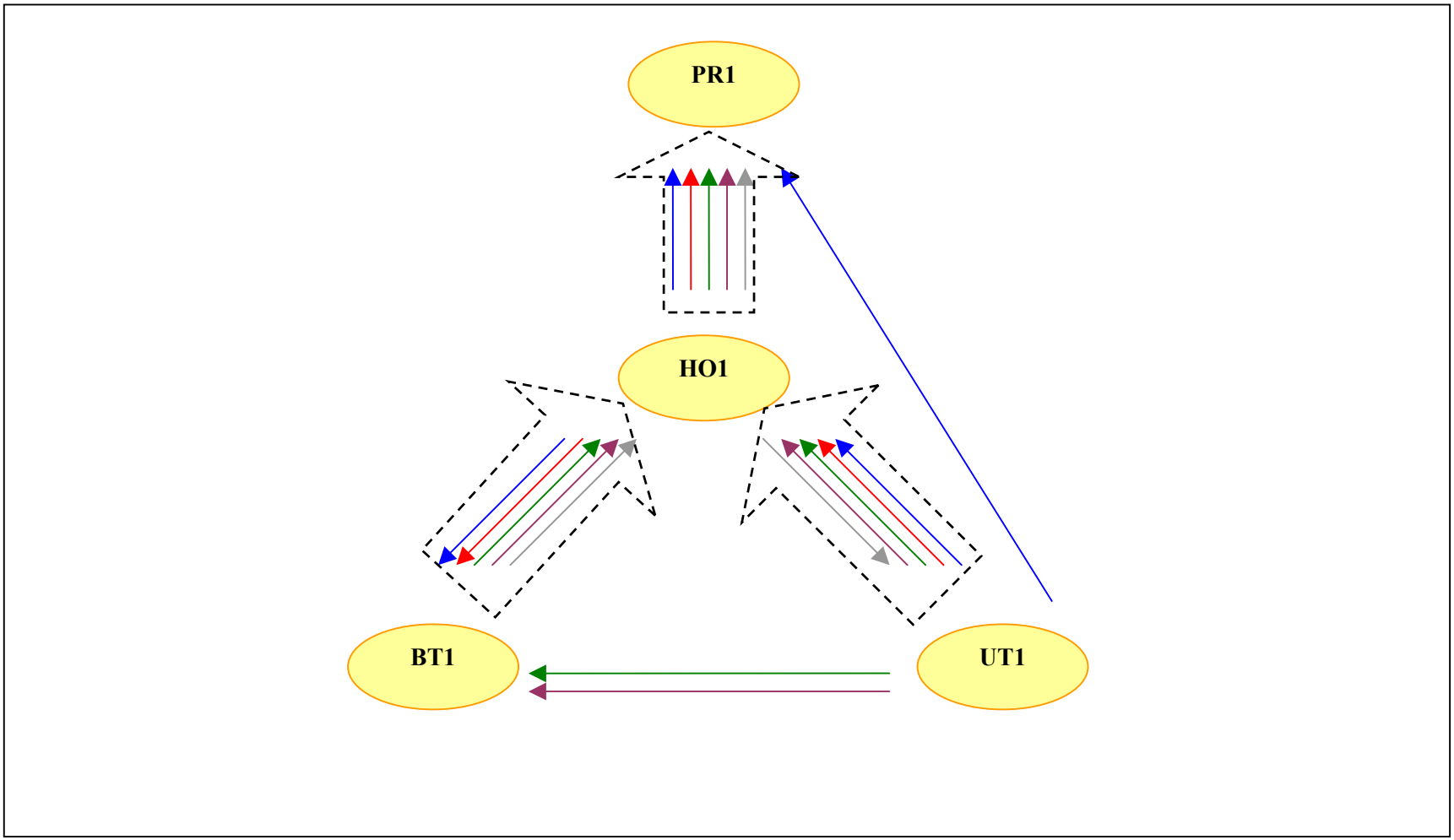
Rete Pregnancy – Inverso – Approccio Multi Esperto

Nome rete	Pregnancy
Numero di nodi	4
Numero di archi	3
Descrizione dei Nodi	
Nome Nodi	**Stati**
UT_1	2
BT_1	2
Ho_1	2
Pr_1	2

Database di riferimento	
Nome Database	pregnancy.dat
Numero Campioni	10000
Fonte	http://www.cs.auc.dk/~marta/datamine.htm
Ordinato (dai padri ai figli)	no

Algoritmi di ricostruzione		Archi Corretti	Archi Mancanti	Archi Aggiunti	Orientamenti Corretti	Orientamenti Errati
Bayesiano	N = 20, α = 3	3	0	0	1	2
K2		3	0	1	0	3
K3		3	0	1	0	3
PC	0.3	3	0	1	1	2
TPDA	0.01	3	0	0	1	2
Multi-esperto a Maggioranza (tratteggio)		3	0	0	1	2

Rete Pregnancy – Inverso – Approccio Multi Esperto



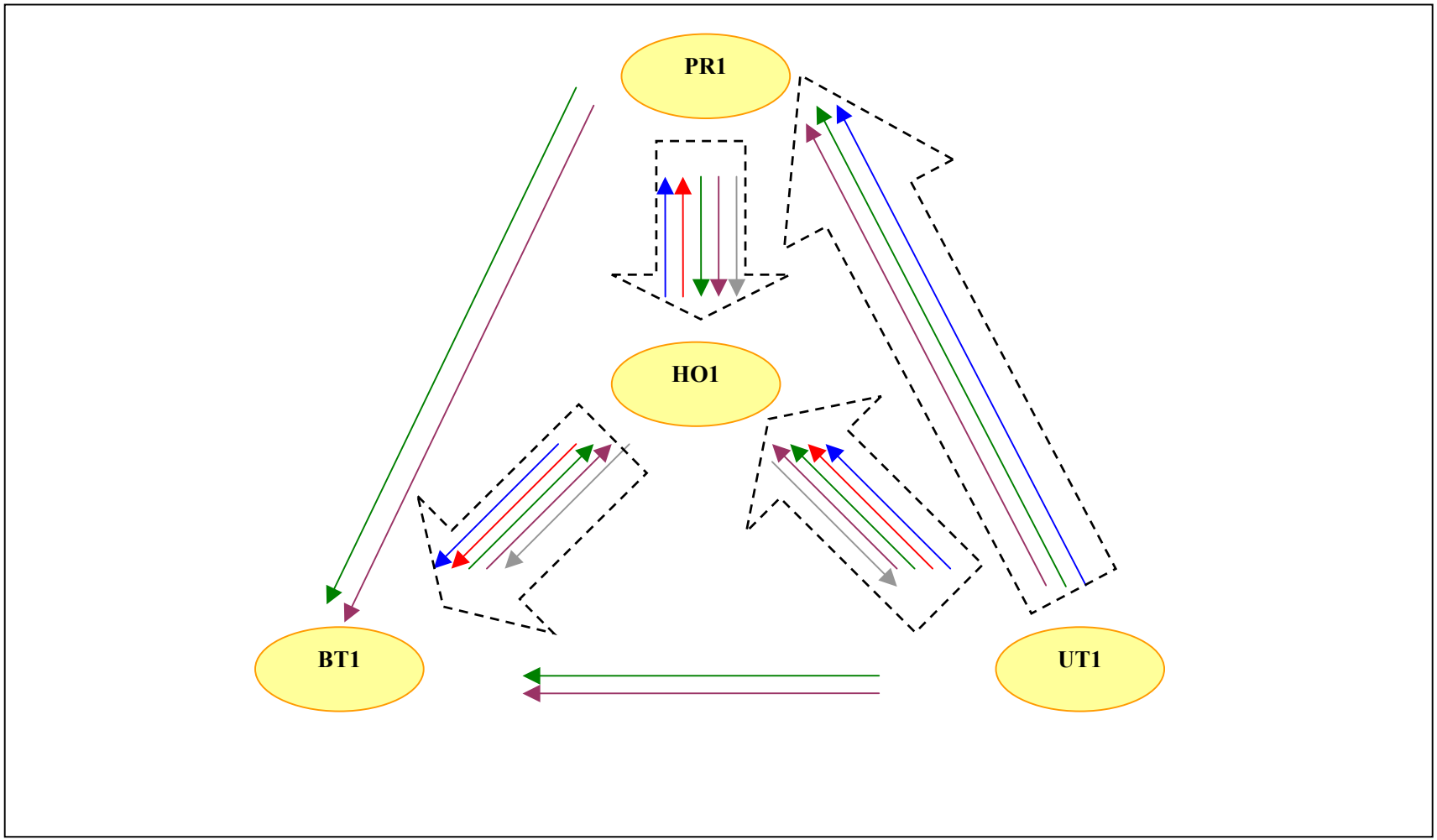
Rete Pregnancy – Casuale – Approccio Multi Esperto

Nome rete	Pregnancy
Numero di nodi	4
Numero di archi	3
Descrizione dei Nodi	
Nome Nodi	Stati
UT_1	2
BT_1	2
Ho_1	2
Pr_1	2

Database di riferimento	
Nome Database	pregnancy.dat
Numero Campioni	10000
Fonte	http://www.cs.auc.dk/~marta/datamine.htm
Ordinato (dai padri ai figli)	no

Algoritmi di ricostruzione		Archi Corretti	Archi Mancanti	Archi Aggiunti	Orientamenti Corretti	Orientamenti Errati
Bayesano	N = 20, α = 3	3	0	0	3	0
K2		3	0	3	1	2
K3		3	0	3	1	2
PC	0.3	3	0	1	1	2
TPDA	0.01	3	0	0	1	2
Multi-esperto a Maggioranza (tratteggio)		3	0	0	2	1

Rete Pregnancy – Casuale – Approccio Multi Esperto



RETE ALARM

Di seguito sono elencati i risultati della rete ALARM. Visto l'elevato numero di nodi, al posto della rappresentazione grafica è riportata una tabella in cui per ogni algoritmo sono illustrati gli archi individuati correttamente (si), invertiti (direzione inversa), aggiunti (in giallo), mancanti (in rosso).

Archi aggiunti	
Archi mancanti	

	Algoritmi	PC	TPDA	Bayesiano	K2	K3	Multi-Esperto a maggioranza
	Parametri	0,3	0,001	N = 200, a = 3			
Nodo origine DA	Nodo Destinazione A						
4	27	si	si	direzione inversa	si	si	**si**
4	5	si	si	si	si	si	**si**
6	5	si	si	si	si	si	**si**
11	27	si	si	direzione inversa	si	si	**si**
11	4						
11	20						
12	10						
12	32	si	si	si	si	si	**si**
16	37	si	direzione inversa	direzione inversa	si	si	**si**
17	25	si	si	si	si	si	**si**
17	26	si	si	si	si	si	**si**
18	2						
18	21						
18	25	si	si	si	si	si	**si**
18	30						
18	26	si	si	si	si	si	**si**
18	3	direzione inversa	direzione inversa	direzione inversa	si	si	**direzione inversa**
19	4	direzione inversa	direzione inversa	direzione inversa	si	si	**direzione inversa**
20	4						
20	27	si	si	direzione inversa	si	si	**si**
20	5						
20	26						
21	7						

21	31	si	si	si	si	si	**si**
21	27						
21	10	si	direzione inversa	direzione inversa	si	si	**si**
22	15	si	direzione inversa	si			**si**
22	31	si	si	si	si	si	**si**
22	35	si		direzione inversa	si	si	**si**
22	34	si	direzione inversa	direzione inversa	si	si	**si**
22	36						
22	13	si	si	si	si	si	**si**
23	31						
23	35	si	si	direzione inversa	si	si	**si**
23	36						
23	13	si	si	direzione inversa	si	si	**si**
24	9						
24	36	si	si	direzione inversa	si	si	**si**
25	1	direzione inversa	si	si	si	si	**si**
25	2	si	si	si	si	si	**si**
26	1						
26	37						
26	6	si	si	si	si	si	**si**
27	12						
27	29	direzione inversa	si	direzione inversa	si	si	**si**
27	14						
27	32						
27	31						
27	34						
28	10						
28	29						
28	7	si	si	direzione inversa	si	si	**si**
29	6	si	si	si	si	si	**si**
29	8	si	si	si	si	si	**si**
29	9	si	si	si	si	si	**si**
29	7	si	si	direzione inversa	si	si	**si**
30	8	si	si	si	si	si	**si**
30	9	si	si	si	si	si	**si**
31	11	si	si	si	si	si	**si**
32	11	si	si	si	si	si	**si**
33	20						
33	27	si	si		si	si	**si**
33	14	direzione inversa	direzione inversa	direzione inversa	si	si	**direzione inversa**
34	14						
34	32	si	si	si	si	si	**si**
34	33	si	direzione inversa	si	si	si	**si**

34	*13*						
35	*13*						
35	*34*	*si*	*direzione inversa*	*direzione inversa*	*si*	*si*	***si***
35	*14*		*direzione inversa*	*direzione inversa*	*si*	*si*	***si***
35	*15*	*si*	*direzione inversa*	*si*	*si*	*si*	***si***
36	*35*			*direzione inversa*	*si*	*si*	***si***
36	*13*	*si*	*si*	*direzione inversa*	*si*	*si*	***si***
37	*24*						
37	*36*	*si*	*si*	*direzione inversa*	*si*	*si*	***si***

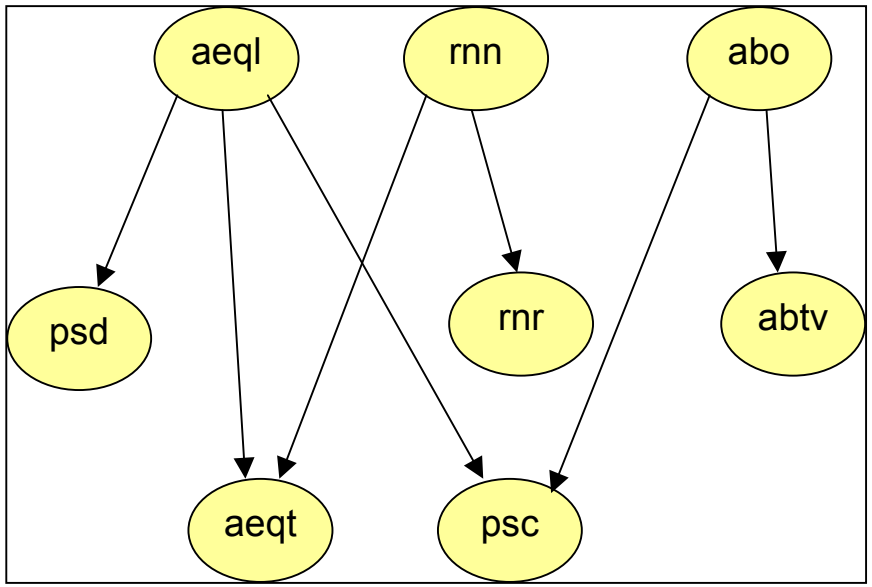
Rete Alarm (Sintesi)

Nome rete	Alarm
Numero di nodi	37
Numero di archi	46

Database di riferimento	
Nome Database	Alarm
Numero Campioni	10000
Fonte	http://www.cs.ualberta.ca/~jcheng/bnpc.htm
Ordinato (dai padri ai figli)	no

Algoritmi di ricostruzione		Archi Corretti	Archi Mancanti	Archi Aggiunti	Orientamenti Corretti	Orientamenti Errati
Bayes		45	1	19	24	21
K2		45	1	1	45	0
K3		45	1	4	45	0
PC		44	2	7	39	5
TPDA		44	2	3	33	11
MultiEsperto a Maggioranza		46	0	0	43	3

RETE FONDAMENTI DI INFORMATICA



Seguono i risultati ottenuti per ogni singolo algoritmo; avendo pochi campioni non sono stati rilevati tutti i legami della rete ma solo quelli più forti.

# archi rete 7	# archi appresi	Archi corretti	Archi mancanti	Archi aggiunti	Orientamenti corretti	Orientamenti errati
PC 0.01	5	2	5	3	0	2
PC 0.1	6	1	7	5	1	0
PC 0.3	7	1	7	6	1	0
PC 0.5	10	3	4	7	2	1
PC 0.7	15	2	5	13	0	2
TPDA	11	3	4	8	1	2

# archi rete 7	# archi appresi	Archi corretti	Archi mancanti	Archi aggiunti	Orientamenti corretti	Orientamenti errati
K2	13	4	3	9	4	0
K3	7	2	5	5	2	0
Bayes 1	9	3	4	6	1	2
Bayes 3	11	3	2	8	0	3
Bayes 10	11	2	5	9	1	1
Bayes 50	18	3	4	15	3	0
Bayes 100	19	5	2	14	5	0

# archi rete 7	# archi appresi	Archi corretti	Archi mancanti	Archi aggiunti	Orientamenti corretti	Orientamenti errati
PC 0.01	6	2	5	4	0	2
PC 0.1	8	2	5	6	0	2
PC 0.3	9	2	5	7	0	2
PC 0.5	10	2	5	8	0	2
PC 0.7	16	3	4	13	1	2
TPDA	9	2	5	7	1	1

# archi rete 7	# archi appresi	Archi corretti	Archi mancanti	Archi aggiunti	Orientamenti corretti	Orientamenti errati
K2	11	2	5	9	0	2
K3	8	2	5	6	0	2
Bayes 1	9	3	4	6	1	2
Bayes 3	11	3	4	8	0	3
Bayes 10	13	3	4	10	2	1
Bayes 50	17	3	4	14	3	0
Bayes 100	19	4	3	15	4	0

# archi rete 7	# archi appresi	Archi corretti	Archi mancanti	Archi aggiunti	Orientamenti corretti	Orientamenti errati
PC 0.01	5	2	5	3	0	2
PC 0.1	6	1	6	5	1	0
PC 0.3	7	1	6	6	1	0
PC 0.5	10	3	4	7	2	1
PC 0.7	15	2	5	13	2	0
TPDA	12	3	4	8	1	2

# archi rete 7	# archi appresi	Archi corretti	Archi mancanti	Archi aggiunti	Orientamenti corretti	Orientamenti errati
K2	15	5	2	10	4	1
K3	9	2	5	7	1	1
Bayes 1	9	3	4	6	1	2
Bayes 3	11	2	5	9	2	0
Bayes 10	11	2	5	9	1	1
Bayes 50	18	3	4	15	3	0
Bayes 100	19	5	2	14	5	0

5 CONCLUSIONI

A conclusione di questo lavoro si valutano i risultati ottenuti durante la sperimentazione traendo alcune considerazioni sia di carattere generale che per i singoli data set.

Abbiamo constatato che l'algoritmo che meglio apprende la struttura di una BN nel caso sia designato l'ordinamento topologico delle variabili è il K2; il K3 lavora, nelle stesse ipotesi, altrettanto bene anche se, talvolta, evidenzia meno legami del K2 (come già citato nel capitolo sullo Structural Learning), ad esempio nella rete ASIA il link Asia - Tubercolosi non è rilevato dal K3.

In merito all'ordinamento, l'algoritmo Bayesiano, che opera con una ricerca di tipo esaustivo, ne è influenzato in misura minore rispetto a K2 e K3; per gli algoritmi dependance - based, come riportato anche in letteratura, è invece preferibile l'ordinamento inverso in modo da considerare i nodi padre solo nelle fasi finali di screening dei test di indipendenza.

Per quanto concerne la scelta dei parametri, l'algoritmo Bayesiano, partendo da un grafo senza legami, ha appreso le strutture in modo soddisfacente (a meno di qualche direzione errata) per valori di α piccoli da 1 a 10 (come ci si aspettava perché abbiamo usato un empty graph iniziale); per $\alpha = 50$ già venivano aggiunti archi non necessari. Per l'algoritmo PC dei risultati accettabili, per buona parte delle BN considerate, sono stati ottenuti fissando una soglia di errore del test al 30% o al 50% (0.3 o 0.5). Un livello di significatività troppo basso, 0.01, è risultato spesso insoddisfacente anche nel caso di database di notevoli dimensioni (10000 campioni). Un valore elevato, 0.7, utile nel caso di database con pochi campioni e legami deboli, come l'ontologia del corso di Fondamenti di Informatica, introduce però ulteriori link fra nodi inesistenti nella gold network.

Nel dettaglio, scegliendo in modo oculato i parametri con cui elaborare i campioni (considerando la dimensione dei data set, le conoscenze a priori, la designazione di un ordinamento corretto per le variabili), l'apprendimento della rete ASIA ha fornito risultati deludenti nel caso "inverso" e "casuale". Per quanto concerne la rete ANGINA, notevole è l'individuazione della gold network specie da parte degli algoritmi dependance - based, quali il PC.

In proposito, è opportuna una precisazione. Algoritmi quali PC e TPDA non sono in grado di orientare tutti gli archi se non sono evidenziate le v-structure e, in tal caso, la learned network è un pdag - partial dag - in cui alcuni legami archi non risultano orientati. Per orientare un grafo, quindi, vi sono varie alternative che devono rispettare sempre il vincolo di aciclicità: adoperare l'entropia condizionata (o la mutua correlazione) o un approccio Bayesiano ($p(X \rightarrow Y|\text{data set})$ come metrica per la determinazione della direzione di un arco). Nel tool BayExpert, per il PC e il TPDA, abbiamo seguito un metodo empirico: stimando le probabilità condizionate valutare quale occorrenza si presenti più volte, P(A|B) o P(B|A). In caso di parità, stimare l'entropia considerando l'euristica che un nodo padre ha minore contenuto informativo (si presenta più volte, specie nei condizionamenti).

Per la rete SPRINKLER i risultati non sono stati molto soddisfacenti per gli algoritmi constraint based : infatti i campioni del data set sono 400. Tuttavia, abbiamo riscontrato che un legame più volte appreso è quello dal nodo Rain a Sprinkler. In effetti, l'esistenza di questa relazione non sarebbe del tutto sbagliata in quanto la pioggia condiziona l'accensione dell'annaffiatore (semplice esempio di data mining).

Per la rete PREGNANCY e COLLEGE sia l'approccio bayesiano che constraint-based hanno fornito risultati soddisfacenti nell'individuare i legami nonostante le direzioni, talvolta, risultassero errate.

Per il classificatore bayesiano LED buona parte degli archi sono stati evidenziati, fatta eccezione per la v-structure 2-1-3, rilevata dal PC con parametro 0.1.

Purtroppo la mancanza di sufficienti campioni a disposizione non ha consentito di osservare risultati lusinghieri sull'ontologia del corso di Fondamenti di Informatica (CFI): difatti un risultato da sottolineare è l'assenza del legame *rnn-rnr* in quasi tutte le sperimentazioni effettuate.

Alla luce dei risultati precedenti, l'approccio multi-esperto è risultato vantaggioso poiché in tutti i database studiati, il multi-expert segue sempre l'indicazione dell'algoritmo che fornisce la struttura più simile alla gold network (eccezione sono la rete SPRINKLER e CFI proprio perché i campioni sono pochi). Il risultato più importante del multi - esperto è proprio la rete ALARM rispetto alla quale abbiamo individuato tutti i 46 legami (le direzioni non sono tutte corrette)! L'unico inconveniente osservato è il dispendio in termini di tempo richiesto

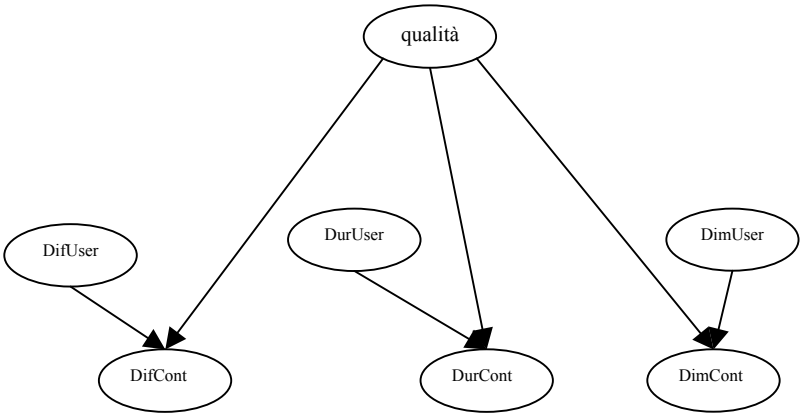
poiché si deve comunque attendere l'elaborazione dei cinque esperti prima di potere combinare i risultati e fornire la learned network (d'altronde lo stesso Java non nasce come linguaggio di programmazione performante per elaborazioni numeriche). La validità dell'approccio multi - esperto è notevole nel rilevare i legami, però le direzioni degli archi sono influenzate dall'ordinamento scelto. Bisogna infatti osservare che gli algoritmi K2, K3 peggiorano notevolmente quando è alterata la corretta disposizione delle variabili del dominio. L'ordinamento *inverso*, ad esempio, tende a prediligere tutte le direzioni all'incontrario rispetto a quelle della gold network mentre quello *casuale* evidenzia numerosi legami inesistenti (specie fra i padri). Un successivo approfondimento a riguardo potrebbe essere lo studio di una strategia "a punteggio" per il multi - esperto, attribuendo un valore maggiore al risultato fornito da un esperto in base anche all'ordinamento scelto (quindi se c'è un ordinamento far pesare di più K2, K3), al numero di campioni del data set e al valore del parametro (specie per gli algoritmi dependence-based).

Uno degli obiettivi della tesi, rappresentato dagli studi sul data set dell'ontologia degli argomenti del corso di Fondamenti di Informatica (CFI), era esaminare l'opportunità di utilizzare le BN in un Sistema di Tutoring Intelligente da integrare in un ambiente di e-learning. L'interesse per le BN è dovuto sia dall'opportunità di costruire, con lo Structural Learning, un'ontologia dalle numerose informazioni a disposizione in un ambiente di e-learning, che dall'agevole processo inferenziale delle reti Bayesiane. Per quanto riguarda la costruzione automatizzata di un'ontologia, i risultati ottenuti dagli algoritmi di Structural Learning e, in particolare, dall'approccio multi-esperto, sono incoraggianti specie se è possibile disporre di informazioni a priori sul dominio. Tuttavia è necessario osservare che il numero di campioni a disposizione deve essere appropriato per l'impiego del learning from data (gli stessi test di indipendenza sono più affidabili nel caso si abbiano disposizione molti campioni). In merito, per l'ontologia CFI i campioni erano insufficienti ciononostante sono state rilevati tre legami definiti "forti" dall'esperto. Data una rete Bayesiana, invece, l'inferenza permetterebbe di risolvere alcuni problemi di decision making relativi alla gestione di un ambiente di e-learning. Ad esempio, in virtù di lavori di tesi precedenti sull'e-learning, abbiamo concepito un prototipo di BN con cui

prevedere le caratteristiche di un contenuto formativo più adatte alle esigenze fornite da un discente/utente. Supponiamo infatti che sia i contenuti informativi che le esigenze dell'utente siano classificate in:

- *difficoltà*: low, medium, high, very high;
- *durata*: (in minuti) 0-10, 10-20, 20-30, 30-60, 60-90, over90;
- *dimensione*: (in Kbyte) 0-100, 100-200, 200-300, 300-400, over400.

Il livello di adattamento fra le caratteristiche dei contenuti e quelle manifestate dall'utente sia rappresentato dalla variabile *qualità* avente stati low, medium, high, very high; allora per schematizzare il problema, potremmo immaginare che in base al grado di adattamento e alle caratteristiche specificate dall'utente (User) si scelga un opportuno contenuto formativo (Cont). Una proposta di BN che modelli questo dominio è illustrata di seguito.



Ci sono quindi dei validi presupposti per potere continuare le ricerche per l'applicazione delle BN in un Intelligent Tutorial System. In tal proposito, il software realizzato segue una struttura modulare che ne possa garantire sia ulteriori sviluppi che l'integrazione in altri moduli. Interessanti sviluppi futuri sarebbero lo studio dell'algoritmo EM (citato nei capitoli precedenti) per gestire missing value e hidden variable e quindi aggiungere, l'algoritmo Structural EM, alla rosa degli *structural learning experts*, ed automatizzare anche la valutazione dei risultati.

In conclusione, le reti Bayesiane offrono una vera e propria tecnologia per la gestione dell'incertezza nell'analisi dei problemi e per il decision making e rappresentano sicuramente una delle nuove avanguardie dell'Intelligenza Artificiale, tali da interessare aziende del calibro della Microsoft.

APPENDICE

⇒ Teoria delle probabilità

Definiamo esperimento casuale ogni atto o processo la cui singola esecuzione (prova) fornisce un risultato non prevedibile. L'insieme dei possibili risultati di un esperimento si chiama spazio campionario S. Nel caso in cui il numero di possibili eventi sia finito o un'infinità numerabile lo spazio campionario è detto discreto, altrimenti continuo. Ciascuno elemento o sottoinsieme (combinazioni di elementi) di S è chiamato evento elementare. Lo studio della relazione tra eventi è riconducibile allo studio della releazioni tra insiemi. In tale ottica, definiamo la probabilità.

Definizione Assiomatica. Una misura di probabilità P è una funzione d'insieme a valori reali definita nello spazio campionario S ed avente le seguenti proprietà (Assiomi)

1. $P(A) \geq 0$, per ogni evento (insieme) A.
2. $P(S) = 1$.
3. $P(A_1 \cup A_2 \cup \ldots) = p(A_1) + p(A_2) + \ldots$ per ogni serie finita o infinita di eventi disgiunti $A_1, A_2, \ldots$

Definizione classica. La probabilità di un evento A è il rapporto tra numero di casi favorevoli al verificarsi di A e il numero totale dei casi possibili, ammesso che questi siano equiprobabili. Tale definizione è insoddisfacente sia perché si definisce la probabilità in termini di casi equiprobabili, nell'ambito della definizione ricorre il concetto da definire, sia perché è appilicabile solo se gli eventi elementari hanno tutti la stessa probabilità.

Definizione frequentista. La probabilità dell'evento A è la frequenza relativa con cui si verifica A in una lunga serie di prove ripetute sotto condizioni simili. La frequenza relativa, come tale, rispetta i tre assiomi della probabilità. Un semplice esempio è il lancio di una moneta n volte. Si otterrà una serie di risultati (C croce - T testa) CTCCTTTTCTTT… dove la probabilità di T o C in ciascuna prova non è influenzata dai risultati delle prove precedenti ed è costante. L'esperienza

suggerisce che, all'aumentare di n, tale frequenza relativa diventa sempre meno oscillante e tende a stabilizzarsi al valore della probabilità. E' necessario la ripetibilità delll'esperimento cui la proposizione probabilistica si riferisce.

Definizione soggettivistica. La probabilità è la valutazione che il singolo individuo può coerentemente formulare in base alle proprie conoscenze sul grado di avverabilità di un evento. Difatti molti eventi che non sono ripetibili sono comunque valutabili dal puinto di vista probabilistico (si pensi ai pronostici sportivi).

Proprietà della misura della probabilità

$$p(-A) = 1 - p(A)$$
$$p(\phi) = 0$$
$$0 \le P(A) \le 1$$
$$P(A_1 \cup A_2) = P(A_1) + P(A_2) - (P(A_1 \cap A_2)$$

Dati due eventi A e B di S, valutiamo la probabilità condizionata ovvero la probabilità di B nell'ipotesi che A si sia ià verficato.

$$p(B \mid A) = \frac{p(A \cap B)}{p(A)}$$

Tale definzione ha senso se p(A) > 0 ed A è assunto come spazio campionario. Segue la definzione di indipendenza, due eventi A e B sono indipendenti se $p(A \cap B) = p(A)p(B)$.

Ogni volta che un evento A può essere visto come effetto di uno tra k possibili eventi C1, C2, …, Ck incompatibili e tali che solo uno di essi si possa verificare ed interessa valutare la probabilità p(Ci|A) è possibile invocare la formula di Bayes

$$p(C_i \mid A) = \frac{p(C_i \cap A)}{p(A)} = \frac{p(C_i)p(A \mid C_i)}{\sum_{j=1}^{k} P(C_j)P(A \mid C_j)}$$

dove l'espressione di p(A) ottenuta al denominatore è ottenuta in virtù degli assiomi della probabilità. [CIC]

⇒ SOFTWARE PER LE RETI BAYESIANE

Sebbene le reti Bayesiane rappresentino un campo di ricerca giovane, il Web offre molti spunti e tool da utilizzare per approfondire lo studio di tali strumenti. In un primo approccio, è difficile districarsi fra i vari software anche perché alcuni non comprendono tutte le funzionalità dell'apprendimento, come lo structural learning (la maggior parte dispone di motori per l'inferenza).
Fondamentale è, quindi, l'elenco di software disponibili fornito dal sito dell'Università di Berkley, www.cs.berkeley.edu/~murphyk/Bayes/bnsoft.html, il quale certamente non sarà completo ma garantisce un'ampia ed esauriente panoramica sull'offerta disponibile. L'elenco è riportato nelle pagine seguenti.
In questo lavoro di tesi è stato principalmente usata la versione free di Hugin ed altri applicativi divulgati da gruppi di ricerca in ambito accademico. Nell'elenco, in grigio chiaro, sono evidenziati i software consultati sia per una maggiore comprensione dell'argomento che degli algoritmi di Structural Learning ed Inferenza.

Legenda

Sebbene molti termini fanno riferimento ad algoritmi o approcci di cui si farà cenno nei capitoli successivi, segue una breve legenda.

- Src = codice sorgente incluso? Y/N
- API = application program interface incluse? (N significa che il programma non può essere integrato con il proprio codice ovvero è concepito come eseguibile standalone.)
- Exe = Eseguibile sui sistemi W = Windows (95/98/NT), U = Unix, M = Mac, o qualsiasi macchina con un compilatore (tale opzione è indicata con "-").
- Cts = variabile continue supportate? (D = discrete)
- GUI = Graphical User Interface inclusa?
- Learns parameters?

- Learns structure? CI = usa test di indipendenza condizionata
- Sample = metodi di campionamento (ad esempio: likelihood weighting, MCMC) supportati?
- Utility = nodi utility e di decisione (diagrammi di influenza) supportati?
- Free = E' disponibile una versione free? Yes, No o Restricted. (I prodotti commerciali spesso hanno versioni free che sono ristrette in vari modi, ad esempio, la massima dimensione di un modello è limitata[80], i modelli non possono essere salvati, o non ci sono API; in questo caso, si usa R per "restricted". I prodotti concepiti per scopi accademici sono spesso solo free per uso non-commeriale)
- Undir = sono supportati grafici a catena o non orientati ?
- Inference = quale algoritmo di inferenza è usato? jtree = junction tree, varelim = variable (bucket) elimination, MH = Metropols Hastings, G = Gibbs sampling, IS = importance sampling, sampling = altri metodi di Monte Carlo, polytree = algoritmo di Pearl ristretto ai grafici senza cicli, none = il program è progettato soltanto per lo structural learning da dati completamente osservati.

[80] Come la versione lite di Hugin.

Name	Authors	Src	API	Exe	Cts	GUI	Params	Struct	Sample	Utility	Free	Undir	Inference	Comments
Analytica	Lumina	N	N	W	Y	W	N	N	N	Y	R	N	?	Spread sheet compatible.
Bassist	U. Helsinki	C++	Y	U	Y	N	Y	N	Y	N	Y	N	MH	Generates C++ for MCMC.
Bayda	U. Helsinki	Java	Y	WUM	Y	Y	Y	N	N	N	Y	N	?	Bayesian Naive Bayes classifier.
BayesBuilder	Nijman (U. Nijmegen)	N	N	W	N	Y	N	N	Y	N	R	N	?	-

Bayesware Discoverer	Bayesware	N	N	WUM	D	Y	Y	Y	N	N	R	N	?	Uses "bound and collapse" for learning with missing data.
B-course	U. Helsinki	N	N	WUM	D	Y	Y	Y	N	N	Y	N	?	Runs on their server: view results using a web browser.
Bayonnet	Motomura (ETL)	Java	Y	WUM	N	Y	Y	N	N	N	Y	N	?	For learning, represents BN as a neural net.
Belief net power constructor	Cheng (U.Alberta)	N	W	W	N	Y	Y	CI	N	N	Y	N	?	-
BNT	Murphy (U.C.Berkeley)	Matlab /C	Y	WU	Y	N	Y	Y	Y	Y	Y	N	Many	Also handles dynamic models, like HMMs and Kalman filters.
BNJ	Hsu (Kansas)	Java	-	-	N	Y	N	Y	Y	N	Y	N	jtree and IS sampling	-

BN Toolkit	Gowans (Imperial)	Visual Basic	Y	W	N	Y	N	Y	N	N	Y	N	Polytree	Parser and GUI for the XML-BIF format.
BucketElim	Rish (U.C.Irvine)	C++	Y	WU	N	N	N	N	N	N	Y	N	Varelim	-
BUGS	MRC/Imperial College	N	N	WU	Y	W	Y	N	Y	N	Y	N	Gibbs	-
Business Navigator 5	Data Digest Corp	N	N	W	D	Y	Y	Y	N	N	R	N	Jtree	-
CABeN	Cousins et al. (Wash. U.)	C	Y	WU	N	N	N	N	Y	N	Y	N	5 Sampling methods	-
CoCo+Xlisp	Badsberg (U. Aalborg)	C/lisp	Y	U	N	Y	Y	CI	N	N	Y	Only	Jtree	Designed for contingency tables.
CIspace	Poole et al. (UBC)	Java	N	WU	N	Y	N	N	N	N	Y	N	Varelim	-
Ergo	Noetic Systems	N	N	WM	N	Y	N	N	N	N	R	N	?	-
FLoUE/BIFtoN	ENS Lyon	Java	Y	WUM	N	N	N	N	N	N	Y	N	Jtree	-

GDAGsim	Wilkinson (U. Newcastle)	C	Y	WUM	Only	N	N	N	Y	N	Y	N	Exact	Useful as a subroutine for Bayesian analysis of large linear Gaussian directed models.
GMRFsim	Rue (U. Trondheim)	C	Y	WUM	Only	N	N	N	Y	N	Y	Only	MCMC	Bayesian analysis of large linear Gaussian undirected models.
Genie/Smile	U. Pittsburgh	N	WU	WU	N	W	N	N	Y	Y	Y	N	Jtree	-
Hugin Expert	Hugin	N	Y	W	Y	W	Y	CI	Y	Y	R	Y	Jtree	-
Hydra	Warnes (U.Wash.)	Java	-	-	Y	Y	Y	N	Y	N	Y	Y	MCMC	-
Ideal	Rockwell	Lisp	Y	WUM	N	Y	N	N	N	Y	Y	N	Jtree	GUI requires Allegro Lisp.
Java Bayes	Cozman (CMU)	Java	Y	WUM	N	Y	N	N	N	Y	Y	N	Varelim/ jtree	-
MIM	HyperGraph Software	N	N	W	Y	Y	Y	Y	N	N	R	Y	Jtree	Up to 52 variables.

MSBNx	Microsoft	N	Y	W	N	W	N	N	N	Y	R	N	Jtree	-
Netica	Norsys	N	WUM	W	Y	W	Y	N	Y	Y	R	N	?	-
PMT	Pavlovic (BU)	Matlab /C	-	-	N	N	Y	N	Y	N	Y	N	special purpose	-
Pulcinella	IRIDIA	Lisp	Y	WUM	N	Y	N	N	N	N	Y	N	?	Uses valuation systems for non-probabilistic calculi.
RISO	Dodier (U.Colorado)	Java	Y	WUM	Y	Y	N	N	N	N	Y	N	Polytree	Distributed implementation.
Tetrad	CMU	N	N	WU	Y	N	Y	CI	N	N	Y	Y	None	-
UnBBayes	?	Java	-	-	N	Y	N	Y	N	N	Y	N	jtree	K2 for struct learning
Web Weaver	Xiang (U.Regina)	Java	Y	WUM	N	Y	N	N	N	Y	Y	N	?	-
WinMine	Microsoft	N	N	W	Y	Y	Y	Y	N	N	Y	Y	None	Learns BN or dependency net structure.
XBAIES 2.0	Cowell (City U.)	N	N	W	N	Y	Y	N	N	Y	Y	Y	Jtree	-

⇒ L'ALGORITMO EM IN DETTAGLIO

In molte situazioni di domini reali, i dati disponibili per l'apprendimento sono incompleti: può essere difficile o anche impossibile osservare alcune variabili. E' perciò importante che un algoritmo di apprendimento sappia fare un uso efficienti dei dati osservati. In presenza di missing value il problema dell'apprendimento diventa molto più difficile; la stima dei parametri è un punto importante sia a causa della difficoltà di effettuare accurate stime numeriche della probabilità, sia perché l'apprendimento dei parametri è talvolta parte integrante di applicativi dedicati all'apprendimento della struttura (nel capitolo sullo Structural Learning si accennerà difatti ad allo Structural EM).

Il problema del learning parameter, secondo l'approccio EM, in breve considera una rete bayesiana S, descritta da un vettore $\overline{\theta}$ di parametri e un insieme D di osservazioni, per poi apprende un nuovo vettore di parametri $\tilde{\theta}$ per S da D.

Due fattori influenzano la scelta di $\tilde{\theta}$: il grado di adattamento a D e il non allontanarsi troppo dal modello preesistente $\overline{\theta}$. Per rispettare questi vincoli si introduce una funzione F da ottimizzare composta dal logaritmo della marginal likelihood (log - likelihood) e dalla distanza fra $\overline{\theta}$ e $\tilde{\theta}$. La forma esatta della funzione F dipende dai pesi che forniamo al log - likelihood e alla distanza ed anche dalla scelta di come valutare $\overline{\theta} - \tilde{\theta}$. Possibili misure per la distanza sono, ad esempio, *relative entropy* (o Kullback-Leibler \ KL-divergence), *chi - quadro* (che è un'approssimazione lineare della precedente).

Il processo di apprendimento dell'EM è iterativo: ad ogni step, si migliora la computazione di $\tilde{\theta}$ a partire da $\overline{\theta}$ (risultato dello step precedente) e da D fintantoché non si raggiunge un criterio di convergenza o un massimo numero di iterazioni. [KOL97][COZ01]

EM Standard

L'utilizzo dell'algoritmo di **E**xpectation - **M**aximization (EM) per il parameter learning è dovuto a A. Dempster (1977).

L'approccio seguito è il criterio MAP o ML: si inizia assegnando una configurazione di parametri $\boldsymbol{\theta}_m$ ad una BN con struttura **m** (l'assegnazione può essere casuale, o scegliendo una distribuzione equiprobabile, per esempio). Poi, si elaborano le statistiche sufficienti (Expectation) in modo da sostituire i missing values ed ottenere un data set completo. L'expectation è condotta considerando la distribuzione di partenza e il database D. Nel caso di variabile discreta si ha

$$E_{p(x|D,\theta_s,S)}(N_{ijk}) = \sum_{i=1}^{N} p(x_i^k, pa_i^j \mid y_l, \theta_s, S)$$

dove $\mathbf{y}_l$ è il possibile *l*-esimo caso incompleto in D e N il numero di record in D. Quando non ci sono missing value il valore da assegnare è banale: è zero o uno. In alternativa, è possibile usare un algoritmo di inferenza bayesiana per valutare i termini mancanti. Le statistiche sufficienti diventano quindi le statistiche per creare un database completo D_c.

Assumendo di massimizzare (Maximization) secondo il criterio ML, determiniamo la configurazione $\boldsymbol{\theta}_m$ che massimizza $p(D_c| \boldsymbol{\theta}_m, \mathbf{m})$. Nel caso di variabili multinomiali discrete segue

$$\theta_{ijk} = \frac{E_{p(x|D,\theta_s,S)}(N_{ijk})}{\sum_{k=1}^{r_i} E_{p(x|D,\theta_s,S)}(N_{ijk})}$$

mentre seguendo l'approccio MAP

$$\theta_{ijk} = \frac{\alpha_{ijk} + E_{p(x|D,\theta_s,S)}(N_{ijk})}{\sum_{k=1}^{r_i} (\alpha_{ijk} + E_{p(x|D,\theta_s,S)}(N_{ijk}))}$$

Il problema dell'apprendimento, sia strutturale che dei parametri, rappresenta un problema di ottimizzazione a più variabili. In tale proposito è opportuno ricordare che le funzioni a più variabili sono dotate di massimi locali (nell'intorno di un intervallo) e massimo assoluto (il valore massimo della funzione nel campo di esistenza). Gli algoritmi di learning, quindi, presentano l'inconveniente di determinare un massimo locale e non globale, determinando un possibile buon modello ma non il migliore. [HEC95]

Nei paragrafi che seguono si illustrano i risultati ottenuti da Bauer, Koller e Singer in "Update rules parameter estimation in Bayesian Networks". [KOL97] In sintesi l'algoritmo da loro proposto, chiamato EM(η) determina una generalizzazione dell'algoritmo EM standard; anzi sono menzionate dei miglioramenti nei tempi di convergenza verso l'effettiva distribuzione delle probabilità scegliendo un valore $\eta = 1.8$.

EM(η): Le equazioni di base

Per fornire maggiore chiarezza all'esposizione che segue esplicitiamo le notazioni usate. Data una BN con struttura S, i parametri sono descritti, per un singolo nodo, da un vettore di conditional probability table (CPT), per cui si ha una cella per ogni stato della variabile ed ogni istanza dei nodi padre. Inoltre, denotiamo con X_i un nodo, $\mathbf{Pa}_i$ l'insieme dei padri di X_i in S, x_i^k per k=1,...,ri i possibili valori assunti da X_i, Pa_i^j una configurazione, per j=1..q_i, di $\mathbf{Pa}_i$ e θ_{ijk} il parametro $p(X_i = x_i^k \mid Pa_i^j)$. Infine $\boldsymbol{\theta}$ indica l'intero vettore di θ_{ijk} mentre l'insieme di osservazioni è $D = \{\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N\}$ per il quale $\mathbf{y}_l$ è un possibile parziale (cioè non completo) assegnamento di valori alle variabili nella rete. Sia $\overline{\boldsymbol{\theta}}$ la corrente assegnazione di $\boldsymbol{\theta}$: si vuole costruire un nuovo modello $\widetilde{\boldsymbol{\theta}}$ basato su $\overline{\boldsymbol{\theta}}$ e D. Poiché il modello corrente $\overline{\boldsymbol{\theta}}$ è già il risultato di qualche processo di apprendimento precedente (da D o da altro campione), non vogliamo ignorarlo completamente. Bisogna quindi bilanciare il potenziale incremento al log-likelihhod ($L_D(\boldsymbol{\theta})$) causato dai dati con le informazioni pre-sistenti ($\overline{\boldsymbol{\theta}}$). Introduciamo la funzione F che bilancia i due fattori precedenti:

$$F(\boldsymbol{\theta}) = \eta \cdot L_D(\boldsymbol{\theta}) - d(\overline{\boldsymbol{\theta}}, \boldsymbol{\theta})$$

Il primo termine $L_D(\boldsymbol{\theta})$ è il log likelihood normalizzato di D, ovvero

$$L_D(\theta) = \frac{1}{N} \sum_{l=1}^{N} \log p(x_i^k, pa_i^j \mid y_l, \theta)$$

La normalizzazione elimina la dipendenza dal numero di campioni durante la computazione. Il termine di penalizzazione $d(\overline{\boldsymbol{\theta}}, \boldsymbol{\theta})$ stima la distanza fra il nuovo ed il vecchio parametro; l'obiettivo è rendere $\overline{\boldsymbol{\theta}}$ e $\boldsymbol{\theta}$ non troppo distanti. Il parametro $\eta>0$ è definito *learning rate* e determina il livello con cui i parametri appresi dai campioni divergono dal modello dell'iterazione precedente. Il learning rate esprime "quanto considerare il passato": se η tende a 1, il passato conta poco e l'update dei parametri si basa principalmente sui dati correnti nel data set mentre se η tende a zero, i parametri della rete cambiano di poco rispetto al modello precedente.

Infine $\widetilde{\boldsymbol{\theta}}$ è

$$\widetilde{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} [F(\boldsymbol{\theta})] = \arg \max_{\boldsymbol{\theta}} [\eta L_D(\boldsymbol{\theta}) - d(\boldsymbol{\theta}, \overline{\boldsymbol{\theta}})]$$

Poiché dal punto di vista computazionale è difficile massimizzare la F, si preferisce massimizzare una versione semplificata, ottenuta linearizzando il termine log - likelihood.

Sia $\nabla L_D(\boldsymbol{\theta})$ il gradiente del vettore $L_D(\boldsymbol{\theta})$ e $\nabla_{ijk} L_D(\boldsymbol{\theta})$ la componente del gradiente corrispondente al $\boldsymbol{\theta}_{ijk}$. Quindi, nell'intorno di vicinanze di $\overline{\boldsymbol{\theta}}$ approssimiamo il log - likelihhod con il primo termine dello sviluppo in serie di Taylor

$$L_D(\theta) \approx L_D(\overline{\theta}) + \nabla L_D(\overline{\theta})(\theta - \overline{\theta})$$

L'approssimazione lineare precedente peggiora quanto più $\boldsymbol{\theta}$ dista da $\overline{\boldsymbol{\theta}}$: interviene il fattore penalizzante *d*. Assumendo $\widetilde{\boldsymbol{\theta}}$ sia non troppo lontano da $\overline{\boldsymbol{\theta}}$, segue

$$\hat{F}(\widetilde{\boldsymbol{\theta}}) = \eta[L_D(\overline{\boldsymbol{\theta}}) + \nabla L_D(\overline{\boldsymbol{\theta}})(\widetilde{\boldsymbol{\theta}} - \overline{\boldsymbol{\theta}})] - d(\widetilde{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}})$$

Una massimizzazione della F, secondo l'approccio Maximum Likelihood, è data, utilizzando la stima della probabilità in base ai campioni, ovvero dai parametri θ_{ijk}

$$\nabla_{ijk} L_D(\theta) = \frac{1}{\theta_{ijk}} \frac{\sum_{l=1}^{N} p_\theta(x_i^k, pa_i^j \mid y_l)}{N} = \frac{E_\theta(x_i^k, pa_i^j \mid D)}{\theta_{ijk}}$$

Il problema della massimizzazione presenta dei vincoli, ovvero per ogni i,j risulta $\sum_{i=1}^{r_i} \theta_{ijk}$, quindi deve essere risolto con le opportune metodologie[81].

Delle possibili soluzioni per la scelta di *d*, esaminiamo la distanza chi - quadro. La distanza chi - quadro fra due distribuzioni di probabilità *p* e *q* è

$$\chi^2(p \,||\, q) = \frac{1}{2} \sum_x \frac{(p(x) - q(x))^2}{q(x)}$$

Nel caso in esame, la relazione precedente diventa

$$\hat{\chi}^2(\widetilde{\boldsymbol{\theta}} \,||\, \overline{\boldsymbol{\theta}}) = \sum_i \sum_j p_{\tilde{\theta}}(pa_i^j) \chi^2(\widetilde{\boldsymbol{\theta}}_{ij} \,||\, \overline{\boldsymbol{\theta}}_{ij})$$

da cui si ricava l'espressione finale per la stima dei parametri

[81] Moltiplicatori di Lagrange, ad esempio.

$$\tilde{\theta}_{ijk} = \frac{\eta}{p(pa_i^j)} E_{\bar{\theta}} p(x_i^k, pa_i^j \mid D) - \frac{\eta}{p(pa_i^j)} E_{\bar{\theta}} (pa_i^j \mid D) \bar{\theta}_{ijk} + \bar{\theta}_{ijk}$$

Batch update rules

Fissato il data set D, il nostro obiettivo è di cercare il parametro $\tilde{\theta}$ che meglio rappresenti D. Per applicare la (33) bisogna stimare $p(pa_I^j)$; una soluzione ragionevole è la stima basata sul campione:

EXPECTATION STEP

$$p(pa_i^j) = \frac{1}{N} \sum_{l=1}^{N} P_{\bar{\theta}} (Pa_i = pa_i^j \mid y_l) = E_{\bar{\theta}} (pa_i^j \mid D)$$

che inserita nell'espressione ricavata per la distanza chi quadro fornisce

MAXIMIZATION STEP

$$\tilde{\theta}_{ijk} = \frac{\eta E_{\bar{\theta}} (x_i^k, pa_i^j \mid D)}{E_{\bar{\theta}} (pa_i^j \mid D)} - \frac{\eta E_{\bar{\theta}} (a_i^j \mid D)}{E_{\bar{\theta}} (pa_i^j \mid D))} \bar{\theta}_{ijk} + \bar{\theta}_{ijk} = \frac{\eta E_{\bar{\theta}} (x_i^k, pa_i^j \mid D)}{E_{\bar{\theta}} (pa_i^j \mid D)} + (1 - \eta) \bar{\theta}_{ijk}$$

Per una fissata iterazione, l'equazione esprime una media pesata fra il parametro stimato dal database di campioni e il parametro computato dall'iterazione precedente $\bar{\theta}_{ijk}$.

Quando $\eta = 1$ l'espressione ricavata altro non è che la regola di update rule dell'algoritmo EM standard.

Per $\eta < 1$, EM(η) il parametro $\tilde{\theta}_{ijk}$ è una combinazione pesata: un valore numerico compreso fra $\bar{\theta}_{ijk}$ e quello stimato dai dati. In particolare l'update dei parametri risulta più lento rispetto all'EM standard. Per $\eta > 1$, piuttosto che interpolare, i parametri indotti dai dati influiscono maggiormente sull'apprendimento che è inoltre più veloce.

Proprietà di convergenza

Poiché la funzione di likelihood di una rete bayesiana con missing values ha più massimi locali, è impossibile derivare dei limiti per la convergenza globale dell'apprendimento dei parametri. Bauer, Koller e Singer, hanno verificato il seguente asserto.

TEROREMA Per qualsiasi data set D, e qualsiasi $0< \eta <2$, la regola di update (34) converge ad un massimo locale della funzione di verosimiglianza $p(D|\theta^*)$, all'interno di un intorno $|| \theta - \theta^*|| < \delta$.

Sebbene non sia possibile valutare esattamente η, dal loro lavoro si evince che per il valore $\eta = 1.8$ l'algoritmo risulta più efficiente: richiede la metà delle iterazioni per convergere. Questo risultante è importante in quanto, ad ogni iterazione, la stima dei parametri implica la consultazione del dataset.
In particolare gli autori affermano che "l'optimal learning rate" in un intorno di θ^* è maggiore di 1, per cui l'EM(η) è migliore dello standard EM in termini di velocità di convergenza. Tuttavia lo standard EM garantisce la convergenza da qualsiasi punto di partenza, mentre i risultati ottenuti per l'EM(η) riguardano un intorno del local maxima.
La procedura dell'EM(η) si presta anche per l'on-line learning dove la scelta di η può essere adattata al numero di campioni acquisiti. In particolare una versione dedicata all'on-line learning, chiamata Voting EM, è illustrata in "Online learning of Bayesian Network parameters" di Cohen, Bronstein e Cozman. [COZ01]

⇒ ALGORITMI DI STRUCTURAL LEARNING

I modelli grafici probabilistici includono le reti bayesiane, oggetto di questo lavoro di tesi, e le reti di Markov. Negli ultimi anni, il *graphical model learning* è divenuto un campo di ricerca molto attivo: segue quindi una breve panoramica degli algoritmi di Structural Learning presente nel contributo di J.Cheng, D. Bell, W.Liu "Learning Bayesian Network from data: an efficient approach based on information theory" [CHE97].

Metodi Search & Score

a. **Chow-Liu Tree Construction Algorithm (1968)**: Chow e Liu svilupparono un algoritmo per la costruzione di grafi ad albero che ha avuto una grande influenza sui graphical model algorithm. Dati N nodi, con $O(N^2)$ step, viene ricostruito l'albero dai dati relativi ad una distribuzione di probabilità P. In particolare, Chow e Liu mostrarono che quando il grafo sottostante alla distribuzione P era effettivamente un albero, l'algoritmo ricostruiva perfettamente la struttura. L'idea è trovare un modello con la migliore score, Kullback-Leibler cross-entropy, fra tutte le possibili coppie di variabili. In particolare, se i dati non sono generati da un albero allora la cross-entropy predilige la struttura la cui distribuzione di probabilità si avvicini a P. L'approccio è un ibrido fra constraint - based e search & score: considerare le possibili coppie di variabili evita una ricerca di tipo euristico mentre i test di cross-entropy hanno una base statistica. Si intuisce quindi come l'algoritmo non funzioni bene per reti molteplicemente connesse (*multiply connected belief network*) per le quali il numero dei test diverrebbe improponibile.

b. **Rebane - Pearl Polytree Construction algorithm (1987)**: è una diretta estensione dell'algoritmo precedente. Un *politree* (chamato anche grafo singly connected) è una struttura che non presenta cicli ma tale per cui c'è al più un percorso fra due nodi. Rebane e Pearl concepirono anche un metodo, ancora usato in molti algoritmi, per determinare le direzioni degli

archi identificando i nodi "collider" (questo concetto sarà approfondito nel paragrafo relativo alla descrizione di alcuni structural algorithm).

c. **K2 algorithm di Cooper e Herskovits (1992)**: è il maggior rappresentante degli algoritmi per il bayesian learning delle BN. Dati in input un data set ed un ordinamento delle variabili, si applica un Bayesian scorig method con l'obiettivo di trovare la struttura di rete S più probabile dato D, massimizzando p(S|D). Il K2 è conosciuto per l'accuratezza dei risultati sulla rete ALARM, considerata come BN di riferimento dalla comunità scientifica [COO92].

d. **HGC (Heckerman, Geiger e Chickering) algorithm (1994)**: questo algoritmo è basato sull'approccio bayesiano puro. In particolare nel lavoro di questi studiosi, [HEC94], risultano due nuove assunzioni chiamate *parameter modularity* e *event equivalence*, ignorate da altri ricercatori, in base alle quali si integra la conoscenza a priori (ad esempio un grafo di partenza con alcuni legami noti) con i dati.

e. **Kutato (o Kutatò) algorithm di Cooper e Herskovits (1991)**: anziché usare le metriche del K2 o dell'approccio HGC, è usata una entropy-based score, approssimando la vera distribuzione di probabilità congiunta dei dati con quella implicita in una BN che abbia la minima perdita di informazione (massima entropia). Il metodo di ricerca usato nell'algoritmo è simile a quello del K2 per cui necessita di un ordinamento sui nodi. E' uno degli algoritmi più lenti. [COO92]

f. **Wong - Xiang algorithm (1994)**: algoritmo entropy - based per le reti di Markov.

g. **BENEDICT algorithm di Acid e Campos (1996)**: altro algoritmo entropy-based. Utilizza un metodo di ricerca euristico e richiede l'ordinamento dei nodi. A differenza del Kutato, usa un approccio differente: dopo aver ottenuto una struttura di rete dalla ricerca euristica, analizza anche le indipendenze condizionate espresse dalla struttura (concetto di d-separation illustrato nel seguito del capitolo) confrontandole con le indipendenze implicate dai dati.

h. **CB algorithm di Singh e Valtorta (1995)**: rimedia al problema dell'ordinamento sui nodi del K2. Poiché l'analisi delle dipendenze nell'approccio constraint-based consente comunque di orientare gli archi, Singh e Valtorta hanno concepito una versione ibrida. Con un algoritmo constraint - based si determina un possibile ordinamento topologico sui nodi da fornire, con il database di campioni, al K2. [SIN94]

i. **Suzuki's algorithm (1996)**: basato sul principio del minimum description length (MDL) il quale provvede a selezionare una regola che meglio bilanci la semplicità della rete con l'adattamento ai dati. Non usa un metodo di ricerca euristico ma si basa una tecnica, chiamata *branch and bound* che permette di ridurre lo spazio di ricerca. Per pochi campioni l'algoritmo risulta addirittura migliore del K2 ma i risultati peggiorano con database di migliaia di campioni. [SUZ99]

j. **Lam - Bacchus algorithm (1994)**: algoritmo basato sul principio MDL, senza richiedere ordinamento dei nodi. La direzione degli archi si avvale di un metodo di search & score (è considerata la direzione che massimizza lo score).

k. **Friedman - Goldszmidt algorithm (1996)**: L'algoritmo usa due differenti metriche: MDL e Bayesiana. Come nell'algoritmo di Lam e Bacchus, la direzione di un arco è determinata con un metodo search e score. L'importanza di questo lavoro è nell'individuare, oltre alla struttura della BN, le *local structure* che permettono di rifinire localmente le relazioni di un nodo e le relative CPT.

l. **WKD (Wallace, Korn e Dai) algorithm (1996)**: Usa il principio MML (minimum message length) simile all'MDL.

Algoritmi Dependence Analysis - based

a. **The Wermuth - Lauritzen algorithm (1983)**: Dato un ordinamento sulle variabili, per ogni coppia X_k e X_i, con $X_i<X_k$, si applica un test di indipendenza. Se X_i e X_k sono dipendenti allora $X_i \rightarrow X_k$ (in virtù dell'ordinamento). Richiede un elevato numero di test.

b. **Boundary DAG algorithm - Pearl (1988)**: richiede comunque un ordinamento sui nodi ed un numero esponenziale di test, ma non proibitivo come nel Wermuth - Lauritzen algorithm. Definito Boundary perché sfrutta un principio detto *Markov Boundary*.

c. **SRA (Srinivas, Russell e Agogino) algorithm (1990)**: estensione del precedente. Permette di avere sia un ordinamento parziale dei nodi che conoscenza a priori sul dominio. Richiede un numero esponenziale di test.

d. **Constructor Algorithm - Fung e Crawford (1990)**: contempla le reti di Markov. Non richiede ordinamento. A differenza di altri metodi simili, non fissa una singola soglia per i test bensì sceglie la rete fra quelle ottenute inserendo soglie diverse, in modo da evitare l'overfitting quando i dati a disposizione non sono numerosi.

e. **SGS (Spirtes, Glymour e Scheines) algorithm (1990)**: No ordinamento, orienta gli archi, numero di test esponenziale.

f. **PC algorithm - Spirtes e Glymour (1991)**: deriva dal precedente, versione migliorata, per BN con sparse model ("sparse" - non molti legami).

Altri approcci

a. **Model averaging**: alcuni ricercatori (Buntine; Madigan e Raftery) usano le tecniche di *model averaging* perché ritengono che talvolta i dati non identifichino esattamente la struttura implicita in essi. Perciò invece di un singola soluzione è opportuno che l'algoritmo consideri più reti.

b. **hidden variables o latent variables**: in certe circostanze alcune variabili non sono mai presenti nel data set (hidden) o non sono presenti i loro valori (missing). In tale ambito sono stati compiuti alcuni progressi per lo Structural Learning (compito già impegnativo con database completi!) grazie a Spirtes; Verma e Pearl; Ramoni e Sebastiani (metodo *Bound and Collapse*); Friedman (*Structural EM*).

$\Rightarrow$ TEST CHI QUADRO PER L'INDIPENDENZA

Si consideri una popolazione statistica le cui unità siano raggruppate secondo le classi A = { A_1,A_2,…, A_r } e B = { B_1,,B_2,…,B_t } le quali modellano due caratteristiche qualitative (come professione e sesso di una persona) o quantitative (peso e statura, ad esempio). Sia p_{ij} la frequenza relativa delle unità aventi A_i come modalità di A e B_j come modalità di B: il tutto è formalizzato nella seguente *tabella di contingenze*.

B	**A**						
	A_1	**A_2**		**A_i**		**A_r**	
B_1	p_{11}	p_{21}	…	p_{i1}	…	p_{r1}	$p_{.1}$
B_2	p_{12}	p_{22}	…	p_{i2}	…	p_{r2}	$p_{.2}$
…	…	…	…	…	…	…	…
B_j	p_{1j}	p2j	…	p_{ij}	…	p_{rj}	$p_{.j}$
…	…	…	…	…	….	…	…
B_t	p_{1t}	p_{2t}	…	p_{it}	…	p_{rt}	$p_{.t}$
Totale	$p_{1.}$	$p_{2.}$	…	$p_{i.}$	…	$p_{r.}$	1

dove si è posto $p_{i.} = \sum_j p_{ij}$, $p_{.j} = \sum_i p_{ij}$. La quantità p_{ij} può essere interpretata come la probabilità di osservare, in un'estrazione casuale dalla popolazione, una unità appartenente alla coppia (A_i,B_j). Si consideri ora l'estrazione di un campione di *n* unità dalla popolazione in oggetto e classificato secondo le stesse classi A e B.

B	**A**						
	A_1	**A_2**	**A_i**			**A_r**	
B_1	n_{11}	n_{21}	…	n_{i1}	…	n_{r1}	$n_{.1}$
B_2	n_{12}	n_{22}	…	n_{i2}	…	n_{r2}	$n_{.2}$
…	…	…	…	…	…	…	…
B_j	n_{1j}	n_{2j}	…	n_{ij}	…	n_{rj}	$n_{.j}$
…	…	…	…	…	….	…	…
B_t	n_{1t}	n_{2t}	…	n_{it}	…	n_{rt}	$n_{.t}$
Totale	$n_{1.}$	$n_{2.}$	…	$n_{i.}$	…	$n_{r.}$	1

in cui $n_{i.} = \sum_j n_{ij}, n_{.j} = \sum_i n_{ij}$.

Si voglia ora identificare l'ipotesi di indipendenza tra A e B. In simboli, dalla tabella delle contingenze, segue

$$p_{ij} = p(A_i \text{ e } B_j)$$

Ma se consideriamo A_i e B_j come due eventi indipendenti allora risulta

$$p_{ij} = p(A_i,B_j) = p(A_i)p(B_j) = (\text{essendo } p(A_i) = p_{i.} \text{ e } p(B_j) = p_{.j}\ ) = p_{i.}\ p_{.j}$$

Se tale relazione è valida per ogni coppia i,j si dice che le caratteristiche A e B sono tra loro indipendenti.

Dunque l'ipotesi (detta nulla) da verificare è

$$\mathbf{H_0 : p_{ij} = p_{i.}\ p_{.j}},\ i = 1,2,\ldots,r;\ j = 1,2,\ldots,t$$

Sono possibili due situazioni:

1. Le frequenze marginali $p_{i.}$ e $p_{.j}$ sono note; in questo caso la verifica dell'ipotesi si riduce al giudizio di conformità delle frequenze osservate n_{ij} alle frequenze attese $np_{i.}p_{.j}$ (più che di indipendenza si parla di *test di adattamento*);
2. Le frequenze marginali non sono note e allora è necessario stimarle dai dati del campione.

Nella seconda situazione, stimando $p_{i.}$ e $p_{.j}$ con il metodo della massima verosimiglianza (Maximum Likelihood), si ottiene

$$\hat{p}_{i.} = \frac{n_{i.}}{n}, \hat{p}_{.j} = \frac{n_{.j}}{n}$$

che inserita nella statistica da valutare per il test $\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{t} \frac{(n_{ij} - np_{i.}p_{.j})^2}{np_{i.}p_{.j}}$, fornisce

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{t} \frac{(n_{ij} - \frac{n_{i.}n_{.j}}{n})^2}{\frac{n_{i.}n_{.j}}{n}}$$

Si noti che essendo $\sum_i p_{i.} = \sum_j p_{.j} = 1$ i parametri da stimare sono (r - 1 + t - 1); quindi i gradi di libertà della distribuzione χ^2[82] sono rt - - (r -1 + t -1) = (r - 1) (t - 1).

Infine bisogna esprimere il livello di significatività, o fiducia, nel test ovvero la probabilità con cui si determina la "zona di rifuto del test", che in genere è fissata a livelli convenzionali 0,05, 0,01, 0,001. Il livello di significatività (SL), quindi, altro non è che la probabilità che la generica statistica S, nel nostro esempio χ^2, cada nella zona di rifiuto quando l'ipotesi è vera (in pratica la probabilità che il test fornisca un risultato errato):

SL = p(S nella zona di rifiuto |H0 vera)

Quanto minore è il valore di SL tanto maggiore è la credibilità di un eventuale rifiuto dell'ipotesi nulla (in quanto si avrebbe bassa possibilità di sbagliare).

[82] Una variabile casuale X ha distribuzione chi-quadrato (con $x \geq 0$ e r gradi libertà) se $f(x) = \frac{1}{2^{r/2}\Gamma(r/2)} x^{r/2-1} e^{-x/2}$

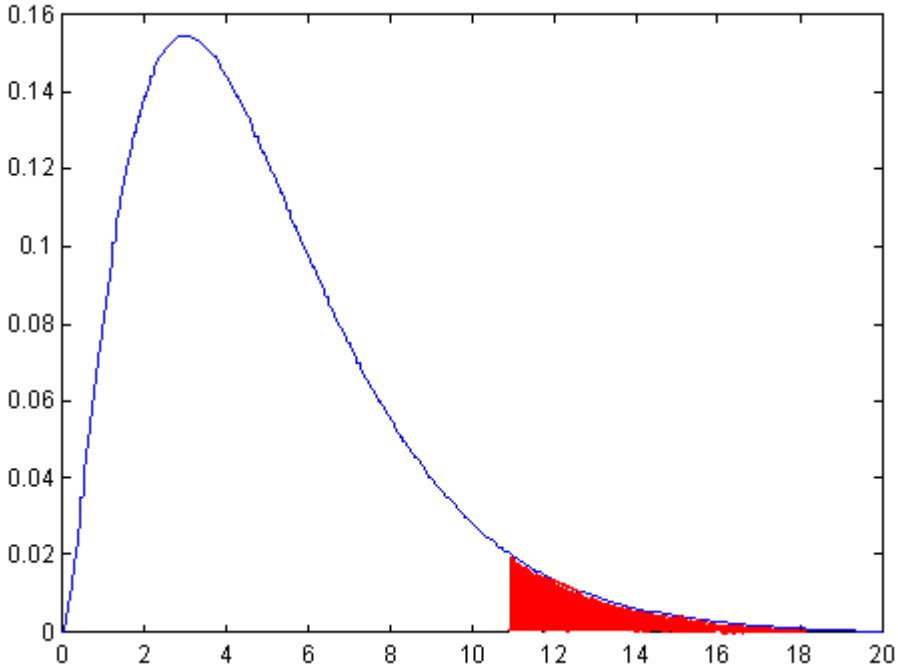


Figura 37 - Distribuzione chi quadro per 5 gradi libertà. In rosso è indicata la zona di rifiuto a SL = 5% = 0.05. L'ipotesi nulla va rifiutata se $\chi^2 > \chi^2_{0.05} = 11.07$.

Chiariamo il utto con un esempio. Si consideri il seguente campione di persone appartenenti alle forze di lavoro classificate secondo il sesso e la condizione occupazionale:

Condizione occupazionale	**Sesso**		**Totale**
	M	**F**	
Occupati	141	69	210
In cerca di occupazione	9	11	20
Totale	150	80	230

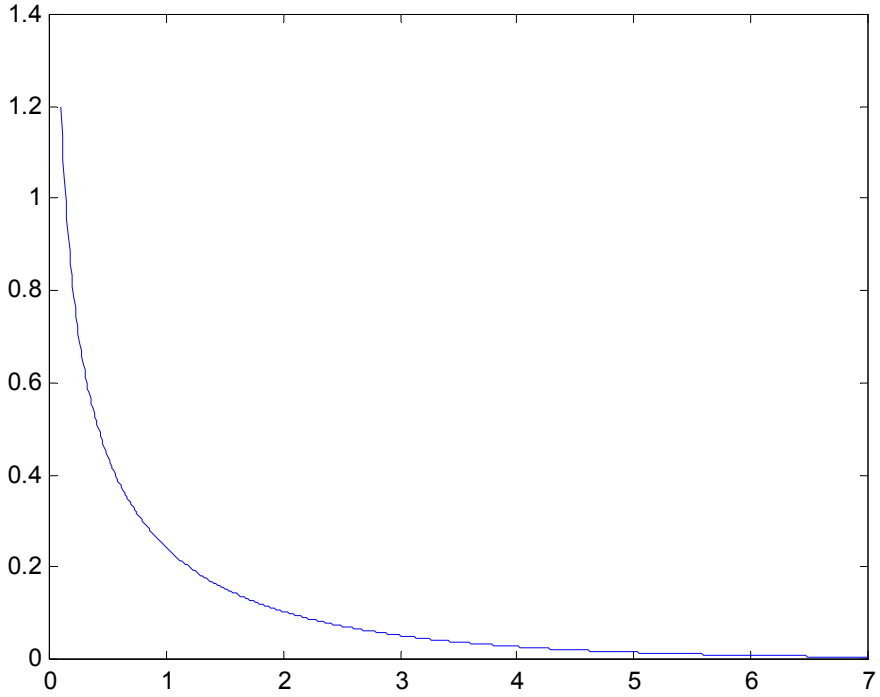


Figura 38 - Distribuzione chi quadro gradi di libertà = 1.

Si vuole verificare l'ipotesi nulla che vi sia indipendenza tra sesso e condizione occupazionale a livello di significatività dello 0.01. Dalle relazioni precedenti si ricava

$$\chi^2 = \frac{[141 - (\frac{150 \cdot 20}{230})]^2}{\frac{150 \cdot 20}{230}} + \ldots + \frac{[11 - (\frac{80 \cdot 20}{230})]^2}{\frac{80 \cdot 20}{230}} = 3{,}94$$

poiché $\chi^2_{0,01} = 6.63$ (gradi di libertà = 1) si conclude che, al livello di significatività 1%, l'ipotesi di indipendenza *non* va rifiutata (il valore non cade nella zona di rifiuto). [CIC]

Il test di indipendenza diventa meno agevole quando bisogna considerare l'eventualità di variabili condizionate. In tale proposito, definiamo *condition set* l'insieme delle variabili condizionanti e *ordine del test* la cardinalità di tale insieme. Ad esempio se le classi A,B fossero condizionate da una terza classe

(ovvero variabile casuale) C l'indipendenza sarebbe espressa dalla relazione $p(A,B|C) = p(A|C)\ p\ (B|C)$ e il test verrebbe definito di ordine 1. Diventa così meno immediata la stessa definizione della tabella di contingenza rispetto al test di ordine 0. Un modo pratico, adoperato in questa tesi, per schematizzare il condition set è calcolare la tabella delle contingenze fissando il condition set. Riconsiderando l'esempio A,B|C e, per semplicità, ipotizzando di avere tre variabili binarie, l'idea è di fare riferimento, per il test, ad una tabella di contingenza come la seguente (i = 1,2)

$\mathbf{C_i}$	$\mathbf{A_1\ C_i}$	$\mathbf{A_2\ C_i}$
$\mathbf{B_1\ C_i}$	n_{A1B1Ci}	n_{A2B1Ci}
$\mathbf{B_2\ C_i}$	n_{A1B2Ci}	n_{A2B2Ci}

Si intuisce che i test, all'aumentare dell'ordine, richiedono un maggiore impegno sia computazionale che in termini di tempo (per consultare ogni volta il database di campioni).

BIBLIOGRAFIA

⇒ PROBABILITÀ E STATISTICA

[CIC] Giuseppe Cicchitelli, Probabilità e Statistica, Maggioli Editore - Rimini

[PAP] Athanasios Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw hill Kogakusha, Ltd.

[PRO] John Proakis, Masoud Salehi, Communication Systems Enginnering, Prentice - Hall Internayional, Inc.

⇒ RETI BAYESIANE

[BEI89] Beinlich, Ingo, H. J. Suermondt, R. M. Chavez, and G. F. Cooper, The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks*, Proc. of the Second European Conf. on Artificial Intelligence in Medicine (London, Aug. 1989), 38, 247-256*

[BHA93] Raj Bhatnagar, Laveen N. Kanal, Structural and probabilistic knowledge for abductive reasoning, *IEEE Transactions on pattern analysis and machine intelligence, vol. 15, no. 3, march 1993*

[BOU94] R.R. Bouckaert, Probabilistic Network construction using the Minimum Description Length principle, *Utrecht University, Department of Computer Science, UU–CS–1994-27*
(Esposizione chiara su reti bayesiane, metrica bayesiana e MDL)

[BUN94] W. L. Buntine, Operations for Learning with Graphical Models, *1994 AI Access Foundation and Morgan Kaufmann Publishers*
(Lavoro ampio sui modelli grafici in generale)

[BUN96] Wray Buntime, A guide to the literature on learning probabilistic network from data, *IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 2, April 1996 page(s) : 195 - 210*
(Un articolo che permette di avere una visione completa, ma non approfondita ed esaustiva, delle BN dal punto di vista matematico, dello structural learning e dell'inferenza con una buona bibliografia e i riferimenti agli articoli più rilevanti)

[CHE97] Cheng, J., Bell, DA, Liu, W., Learning belief networks from data: an information theory based approach, *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management, 1997*
(algoritmo TPDA)

[CHI96] D. M. Chickering, Learning Bayesian Networks is NP – Complete, *Learning from Data: AI and Statics. Edited by D. Fisher and H.J. Lenz, 1996 Springer Verlag, Cap. 12*
[COO92] Cooper and Herskovits, 1992 Gregory F. Cooper , Edward Herskovits, A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning, v.9 n.4, p.309-347, Oct. 1992*

[COZ01] Ira Cohen, Alexander Bronstein, Fabio G. Cozman, Online learning of bayesian network parameters, *Internet Systems and Storage Laboratory HP Laboraotires Palo Alto – HPL-2001-55(R.1) June 5, 2001*

[DAS99] D. Dash, M.J. Druzdzel, A hybrid anytime algorithm for the construction of Causal Models from sparse data, *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI – 99), pages 142 – 149, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1999*

[HSU02] Guo, H. & Hsu, W. H. (2002). A Survey of Algorithms for Real-Time Bayesian Network Inference. *In Guo, H., Horvitz, E., Hsu, W. H., and Santos, E., eds.* Working Notes of the Joint Workshop (WS-18) on Real-Time Decision Support and Diagnosis*, AAAI/UAI/KDD-2002. Edmonton, Alberta, CANADA, 29 July 2002. Menlo Park, CA: AAAI Press*

[DEC96] R. Dechter, Bucket elimination: a unifying framework for probabilistic inference, *Proceedings of twelthth Conference on Uncertainty in Artificial Intelligence, pages : 211 – 219, Portland, Oregon, 1996 – E. Horviots and F. Jensen editors*

[FRI98] N. Friedman, The Bayesian structural EM algorithm, *Uncertainty in Artificial Intelligence: Proceedings of th Fourteenth Conference, pages 129 – 138, Madison, Wisconsin, 1998. Morgan Kaufmann*.
(L'algoritmo descritto rappresenta uno degli ultimi lavori apprezzati dal mondo scientifico riguardo allo Structural Learning)

[FRI99] Nir Friedman, Iftach Nachman, Dana Peer, Learning Bayesian Network Structure from Massive datasets: the "sparse candidate" algorithm, *Proceedings of 15th Conference on UAI (Uncertainty in Artificial Intelligence) , IEEE 1999*

[GUO02] Hsu, W. H., Guo, H., Perry, B. B., & Stilson, J. A. (2002). A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, New York, NY. San Francisco, CA: Morgan Kaufmann

[HEC94] David Heckerman, Dan Geiger, David M. Chickering, Learning Bayesian Network: the combination of Knowledge and statistical data, *Machine Learning, 20:197-243,1995* (anche Microsoft Research - MSR-TR-94-09 March 1994)

[HEC95] David Heckerman, A tutorial on learning with Bayesian networks, *Learning in Graphical Models - Adaptive Computation and Machine Learning The MIT Press, Cambridge, Massachusetts - M.I. Jordan Editor - 1999*
(anche Microsoft Research – MSR – TR – 95 – 06)
(Una delle opere più accreditate. Per il formalismo usato e il rigore matematico, non è consigliato per una fase introduttiva all'argomento)

[HEC97] David Heckerman, Bayesian Networks for Data Mining, *Journal of knowledge Discovery and Data Mining 1(1), pag. 79-119 (1997), Kluwer Acedemic Publishers*
(Sul data mining dice poco; simile alle opere precedenti di Heckerman)

[HUG] Introduzione a Hugin (dall'help dell'omonimo tool)
(Da considerare non solo per apprendere le funzionalità di Hugin ma anche per la semplicità di esposizione dei concetti sulle reti bayesiane)

[JEN96] Finn V. Jensen, An introduction to Bayesian networks, Springer (1996).
(Uno dei testi (non numerosi) sulle BN. Rappresenta un valido supporto per la comprensione, in fase iniziale, dell'argomento. Illustra l'algoritmo di inferenza junction tree e sono presenti numerosi esempi)

[JEN99] Finn V. Jensen, Lecture notes on Bayesian networks and influence diagrams, *Department of Computer Science – Aalborg University – Frederik Bajers Vej 7 – Denmark, September 1999*

[KOL97] Eric Bauer, Daphne Koller, Yoram Singer, Update rules for parameter estimation in Bayesian Networks, *Proceedings of the Thirteenth Annual Conference on uncertainity in Artificial Intelligence (UAI-97) pages 3-13, Providence, Rhode Island, August 1-3,1997*

[KOL01] Daphne Koller, Nir Friedman, Learning Bayesian Networks from data – *NIPS 2001 Tutorial – Relevant Readings*
(un'ampia bibliografia e riferimenti sulle BN)

[LAM98] Wai Lam, Bayesian Network refinement via machine learning approach, *IEEE Transactions on Pattern Analysis and Machine Learning Intelligence, Vol. 20, N. 3, March 1998*
(Esposizione chiara del principio MDL)

[LAM02] Wai Lam, Alberto Maria Segre, A distributed learning algorithm for bayesian inference networks, *IEEE Transactions on Knowledge and Data Engineering, Vol. 14, N. 1, January/February 2002*

[LAU88] Lauritzen, Steffen L. and David J. Spiegelhalter (1988), Local computations with probabilities on graphical structures and their application to expert systems, *in J. Royal Statistics Society B, 50(2), 157-194*

[MEE97] D. Heckerman, C. Meek, and G. Cooper A Bayesian Approach to Causal Discovery. *In C. Glymour and G. Cooper, editors, Computation, Causation, and Discovery, pages 141-165. MIT Press, Cambridge, MA, 1999. (Anche Technical Report MSR-TR-97-05, Microsoft Research, February, 1997*)

[MUR01] K. Murphy, Learning Bayes net structure from sparse data sets. *Technical report, Computer Science Division, University of California, Berkeley, 2001*

[PEA00] J. Pearl, and S. Russell, Bayesian Networks, *In M. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, second edition, forthcoming, 2001*
(Una breve panoramica sull'utilità e le applicazioni delle BN. Utile perché viene presentata la gold network della rete sprinkler su cui sono state effettuate delle sperimentazioni)

[PAP] Enrico Papalini, Michele Piccinini, Apprendimento di reti bayesiane da database di esempi, Università di Firenze
(algoritmo Bayesiano)

[RAM98] Marco Ramoni, Paola Sebastiani, Bayesian methods for intelligent data analysys, *Technical Report TR-67, Knowledge Media Institute Technical Report, KMI – TR - 67, The Open University, Milton Keynes, United Kingdom. July 1998*

[SPI] Peter Spirtes, Christopher Meek, Learning Bayesian Networks with Discrete Variables from Data, *Department of Philosophy, Carnegie Mellon University Pittsburgh*

[SAH01] Ferat Sahin, John S. Bay, Structural Bayesian network learning in a biological decision theoretic intelligent agent and its application to a herding problem in the context of distributed multi agent system, *2001 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, pages: 1606 - 1611*

[SIN94] Moninder Singh, Marco Valtorta, Construction of bayesian network structures from data: a brief survey and efficient algorithm, *International Journal of Approximate Reasoning 1994 – 11: 1 -158, 1994 Elsevier Science inc.*
(L'articolo presenta l'algoritmo K2 e come modificarlo per risolvere il problema dell'ordinamento: algoritmo CB)

[STE00] Todd A. Stephenson, An introduction to Bayesian network theory and usage, *IDIAP Research Report IDIAP – RR - 00-03, 2000*
(Nonostante il formalismo matematico, è indicato per introdurre la teoria delle BN. Breve panoramica sulla teoria dei grafi. Illustra l'algoritmo junction tree per l'inferenza)

[SUZ99] Joe Suzuki, Learning Bayesian Belief Networks based on the MDL principle: an efficient algorithm using the branch and bound technique, *IEICE TRANS. INF. & SYST., VOL. E82, NO.2 February 1999*

[TON01] Simon Tong, Daphne Koller, Active learning for structure in Bayesian networks, *International Joint Conference on Artificial Intelligence 2001*

[VER92] T. Verma, J. Pearl, An algorithm for deciding if a set of observed independencies has a casual explanation, *Proceedings of the Eighth Conference on Uncertainty in Artificial IntelligenceMorgan Kaufmann Publishers, Inc., San Francisco, 1992*

⇒ APPLICAZIONI DELLE BN

[ANT02] P. Antal, B. De Moor, D. Timmerman, T.Meszaros, T.Dobrowiecki, Domain knowledge based information retrieval language: an application of annotated baysesian networks in ovarian cancer domain, *Proceedings of the 15 th ieee symposium on computer based medical system (CBMS 2002)*
(Si presenta il modello ABN - Annotated Bayesian Network - ovvero modelli probabilistici arricchiti con annotazioni testuali. In tale proposito viene illustrato un nuovo query language che possa sfruttare queste potenzialità).

[BEN01] Souad Souafi Bensafi, Marc Parizeu, Franck Lebourgeois, Hubert Emptoz, Logical labeling using Bayesian Network, *Proceedings Sixth International Conference on Document Analysis and Recognition, 2001, page(s) : 832 - 836*
(Nell'articolo è illustrato un prototipo di logical labeling, ovvero il costruire la struttura logica di un documento. Le BN sono usate per la classificazione.)

[CHA01] Sung – Hyuk Cha, Sargur N. Srihari, A priori algorithm for sub-category classification analysis of handwriting, *Proceedings of sixth International Conference on Document Analysis and Recognition, 2001, page(s) : 1022 - 1025*
(Applicazione del data mining per un problema di classificazione. Algoritmo Apriori.)

[BUT02] C.J. Butz, Exploiting contextual indipendencies in web search and user profiling, *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Vol. 2 page(s): 1051 - 1056*

[HUE00] Luis M. de Campos, Juan M. Fernandez, Juan F. Huete, Building bayesian network – based Information retrieval systems, *Proceedings of 11th International Workshop on Database and Expert Systems Application, IEEE 2000, page(s); 543 - 550*

[KAR01] R.Chen, K. Sivakumar, H. Kargupta, Distributed Web mining using bayesian networks from multiple data streams, *Proceedings IEEE International Conference on Data Mining, 2001, page(s); 75 - 82*

[EZA96] K. J. Ezawa, S. W: Norton, Constructing bayesian network to predict uncollectible telecommunications accounts, *Expert, IEEE, Vol. 11, Issue: 5, Oct. 1996 page(s): 45 – 51*
(Il sistema APRI - Advanced Pattern Recognition and Identification - sfrutta le BN per la classificazione di un insieme di dati)

[HAG01] C. O'Hagan, The integration of television and internet, *Proceedings of IEEE International Conference on Advanced Learning Technologies, 2001, pages: 475 - 477*
(Si accenna alle università virtuali nell'ambito dell'e-learning)

[LAM97] Wai Lam, Alberto Maria Segre, Distributed data mining of probabilistic knowledge, *Proceedings of the 17th International Conference on Distributed Computing Systems, 1997, pages 178-185*

[LIU96] Kuo-Chu Chang, Jun Liu, Efficient algorithms for learning probabilistic networks, *IEEE International Conference on Systems, Man, and Cybernetics 1996, pages 1274-1279, VOL. 2*
(Sono menzionati gli approcci dell'MDL e K2 come metodi migliori per il learning)

[MYL01] Petri Myllymaki, Tomi Silander, Henry Tirri, Pekka Uronen - *CoSCo Group (Complex System Computation Group), University of helsinky, Finland*, *B-*course: a web service for bayesian data analysis, Proceedings of the 13th International Conference on Tools with Artificial Intelligence, pages 247-256, 2001
(B-course è un applicativo web-based che permette all'utente di analizzare i dati e rilevare le dipendenze fra le variabili, rappresentandole con reti bayesiane. Il servizio si esplica attraverso una procedura in tre passi: upload dei dati, ricerca del modello, analisi del modello).

[PRZ00] K. Wojtek Przytula, Don Thompson, Construction of Bayesian Networks for diagnostics, *Proceedings of Aerospace Conference, IEEE 2000, Vol.5 page(s): 193 - 200*

[SHY00] Mei-Ling Shyu, Shu-Ching Chen, A bayesian network-based expert query system for a distributed database system, *2000 IEEE International Conference on Systems, Man and Cybernetics, Vol.3 page(s): 2074 - 2079*

[TAN96] Jiming Liu, Michel C. Desmarais, Yuan Y. Tang, A method of learning implication networks from empirical data: algorithms and Monte Carlo simulation based validation, *IEEE International Conference on Systems, Man,, and Cybernetics, 1996, pages 1291-1296, VOL. 2*

[WON99] Man Leung Wong, Wai Lam, Kwong Sak Leung, Jack C. Y. Cheng, Applying Evolutionary algorithms to discover knowledge from medical databases, *Proceddings of 1999 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 5 page(s) : 936 - 941 pag. 936-941*

⇒ DATA MINING

[FAY96] U. Fayyad, G. Piatetsky - Shapiro, P. Smyth, R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining", *AAAI/MIT Press, 1996*

[HAN99] J. Han, Data Mining*, J. Urban and P. Dasgupta (eds.), Encyclopedia of Distributed Computing, Kluwer Academic Publishers, 1999*.

[DIN] Torgeir Dingsøyr - Integration of Data Mining and Case-Based Reasoning - http://www.idi.ntnu.no/~dingsoyr/diploma/
(Una tesi svolta alla fine del Master al dipartimento di Informatica e Scienze dell'Informazione dell'Università della Scienza e della Tecnologia in Norvegia)

⇒ INFORMATION RETRIEVAL

[JAC] P.S. Jacobs, *Part II: "Traditional" IR.* in *Text-Based Intelligent Systems: current research and practice in information extraction and retrieval*

[TUR] W. B. Croft e H. R. Turtle Text Retrieval and Inference., in *Text-Based Intelligent Systems*, op. cit.

⇒ ONTOLOGIE

[DES03] M. De Santo, M. Vento, F. Colace, P. Foggia, Ontology learning through Bayesian networks, *Proceedings of ICEIS 2003, Angers, April 2003*

[GUA98] N. Guarino, Formal Ontology in Information Systems, *Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15*

⇒ APPROCCIO MULTI - ESPERTO

[MOL02] Mario Molinara, Individuazione degli Shot in Filmati MPEG Mediante un Approccio Multi-Esperto, *Tesi di Dottorato di Ricerca in Ingegneria dell'Informazione, Elettromagnetismo e Telecomunicazioni, Università degli Studi di Salerno, A.A.2001/2002*

⇒ LINK UTILI

- http://ww.acm.org
- http://auai.org
- http://www.hugin.com
- http://www.agena.co.uk
- http://www.norsys.com
- http://www.cs.berkeley.edu/~murphyk/Bayes/
- http://www.cs.ualberta.ca/~jcheng/bnpc.htm
- http://www.kddresearch.org/Groups/Probabilistic-Reasoning/
- http://www.cs.Helsinki.FI/research/cosco
- http://www.research.microsoft.com/research/dtg/bnformat/
- http://www.cs.cmu.edu/~fgcozman/Research/Interchangeformat
- http://www.cs.auc.dk/~marta/datamine.htm
- http://www.cs.huij.ac.il/~galel
- http://www.phil.cmu.edu/projects/tetrad/publications.html
- http://bndev.sourceforge.net/
- http://www.ics.uci.edu/~mlearn/MLRepository.html
- http://leo.ugr.es/~elvira/
- http://web.tiscali.it/mmariotti