UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"
Polo delle Scienze e delle Tecnologie

Dottorato di ricerca in

Ingegneria delle Reti Civili e dei Sistemi Territoriali

XVIII Ciclo

Indirizzo "Infrastrutture Viarie e Sistemi di Trasporto"

Coordinatore Scientifico
Prof. Ing. Domenico Pianese

Coordinatore di Indirizzo
Prof. Ing. Renato Lamberti

TESI DI DOTTORATO

UN'APPLICAZIONE DELLE RETI BAYESIANE NELL'INDIVIDUAZIONE DELLE CAUSE DI INCIDENTE

Candidato: Ing. Lucia Sparavigna

Tutore: Prof. Ing. Bruna Festa

Anno Accamico 2004 2005

Indice

Capitolo 1: Introduzione

La sicurezza stradale rappresenta, al giorno d'oggi, uno degli argomenti di maggiore interesse, non solo per gli addetti ai lavori. L'aumento della sicurezza, intesa come riduzione del numero degli incidenti e della loro gravità, è un problema di non agevole soluzione. Va tenuto presente, infatti, che gli incidenti, anche se numerosi in senso assoluto, sono degli eventi rari se commisurati ai milioni di veicoli che ogni giorno percorrono le strade; le situazioni di pericolo che si verificano durante gli spostamenti sono ben numerose e, per fortuna, solo raramente sfociano in incidente. [ESP]

Ogni spostamento può essere visto come il funzionamento di un sistema le cui componenti sono la strada il veicolo, l'uomo e l'ambiente. Quando una o tutte le componenti manifestano anomalie di funzionamento, il sistema nel suo complesso comincia a funzionare male e si generano situazioni di pericolo. Le anomalie possono rientrare e il sistema riprende il suo normale funzionamento o accentuarsi, generando situazioni di pericolo più o meno forti, fino a giungere, in qualche caso, al guasto totale del sistema (incidente).

Inoltre per ciascun componente possono elencarsi numerosi fattori di rischio. Ad esempio per la strada possono aversi difetti di geometria dell'asse e della sezione, carenza di visibilità, pavimentazione con bassa aderenza, ecc., per il veicolo, cattivo stato di manutenzione dei freni e degli organi di guida, pneumatici usurati, eccesso di carico per mezzi pesanti, ecc., con riferimento al guidatore, velocità eccessiva, inosservanza del Codice della Strada, guida distratta o in cattive condizioni fisiche per affaticamento, abuso di alcol o droga, ecc., infine, per l'ambiente possono citarsi le condizioni di traffico, nebbia, ghiaccio, pioggia battente, ecc.

E' chiaro, quindi, che al verificarsi dell'incidente concorrano un insieme di fattori che è molto difficile individuare separatamente.

Scopo del presente lavoro di tesi è approfondire lo studio delle tecniche di Intelligenza Artificiale al fine di realizzare una metodologia che permetta di

individuare gli aspetti dell'ambiente stradale che maggiormente influiscono sul livello di incidentalità attuale, cioè di identificare i fattori di potenziale pericolo delle strade esistenti in modo tale che possano essere eliminati o attenuati prima che diano luogo a siti con elevata incidentalità.

L'adozione di misure in favore della sicurezza presuppone, infatti, che si sia in grado di riconoscere e valutare le condizioni di rischio che si accompagnano ad una determinata configurazione infrastrutturale, per la qual cosa il confronto delle caratteristiche tecniche dell'infrastruttura con gli standard suggeriti dalle norme di progettazione non sempre risulta esaustivo delle problematiche presenti.

In questa parte introduttiva si cerca di fornire una panoramica delle problematiche analizzate e sviluppate durante lo svolgimento del lavoro.

Lo sviluppo inarrestabile dei calcolatori e della loro interconnessione, ha portato ad una crescita esponenziale dei dati a disposizione dell'utente. E', quindi, necessario sviluppare delle tecniche che elaborino i dati secondo dei criteri *intelligenti*. La teoria della probabilità fornisce gli strumenti necessari per il ragionamento in presenza di incertezza, mentre le reti bayesiane permettono la creazione di classificatori molto efficaci.

1.1 Organizzazione del Lavoro

Ecco un'illustrazione sintetica del contenuto dei capitoli che seguiranno.

Capitolo 2 Nel secondo capitolo verrà illustrato brevemente il problema della sicurezza stradale con particolare attenzione verso quegli incidenti che non dipendono da un comportamento scorretto di guida dell'utente o da avarie del veicolo, ma dalle condizioni esterne alla guida (condizioni ambientali, visibilità, geometria del tracciato, ecc.).

Capitolo 3 Nel terzo capitolo è presente un'introduzione al data mining, i campi applicativi, le tecniche più utilizzate.

Capitolo 4 Nel quarto capitolo è presente una descrizione dettagliata di come viene costruita una rete bayesiana a partire da osservazioni campionarie su una certa realtà. In particolare, verrà presentato l'approccio bayesiano alla

probabilità ed alla statistica e si descriveranno i modelli grafici probabilistici di cui fanno parte le reti bayesiane, insieme alle tecniche per apprendere i parametri.

Capitolo 5 In questo capitolo verrà decritto come è stata realizzata la rete bayesiana che, una volta addestrata, consente la classificazione e verranno illustrati i risultati ottenuti sperimentando il sistema su un database reale.

Capitolo 2: Identificazione delle caratteristiche influenti sulla sicurezza stradale

Il tema della sicurezza in ambito urbano ed extraurbano, legato alla problematica dell'incidentalità stradale, è diventato negli ultimi anni di notevole attualità, a causa delle dimensioni sempre crescenti che il fenomeno dei sinistri sta assumendo. Ne sono una testimonianza i documenti, volti a promuovere la sicurezza stradale, che negli ultimi anni sono stati prodotti sia a livello di Unione Europea sia a livello nazionale.

Da qui nasce la necessità di predisporre nuove metodologie di studio e di analisi del fenomeno dell'incidente stradale. Tali analisi devono essere mirate ad un approccio preventivo e non solo correttivo, volte a perseguire una sicurezza fatta non solo dal rispetto delle norme di progettazione e dei limiti di velocità, ma da corrette scelte di carattere pianificatorio.

Il patrimonio infrastrutturale esistente risente di un'impostazione progettuale coerente con disposizioni normative che hanno subito nel tempo revisioni ed aggiornamenti anche in relazione alla migliorata consapevolezza delle conseguenze, in termini di sicurezza, dei diversi aspetti progettuali normati. Ciò rende le strade esistenti obsolete sotto lo specifico aspetto della sicurezza stradale, oltre che, spesso, inadeguate anche alle incrementate esigenze della domanda di traffico, aspetto, questo, strettamente connesso al primo.

Per ridurre l'incidentalità e le sue conseguenze è necessario intervenire sulle infrastrutture per eliminare quelle caratteristiche geometriche e infrastrutturali che maggiormente contribuiscono all'incidentalità.

Data l'estensione della rete viaria italiana, i tempi e i costi di intervento diffusi e generalizzati sono molto elevati. Per massimizzare i benefici del Piano Nazionale della Sicurezza Stradale, gli interventi di tipo specifico devono essere attuati in base al principio di eseguire le azioni con il miglior rapporto benefici-costi in modo tale da avere la massima riduzione di incidentalità in relazione alle risorse economiche impiegate.

Si deve, inoltre, tenere presente che, in generale, il fatto che una strada sia stata ben progettata non implica necessariamente che tutti gli utenti avvertiranno i limiti che l'andamento della strada impone al proprio comportamento ovvero che, pur avvertendoli, vi si adegueranno.

L'incidentalità sulle strade è, quindi, un fenomeno controllabile ma non eliminabile, poiché è una conseguenza diretta della libertà di guida, caratteristica principale del trasporto su strada. [GIA 01]

2.1 Catteri generali dell'incidentalità stradale

Il problema della sicurezza stradale è attualmente oggetto di particolare interesse da parte del Ministero dei Trasporti, che ha intrapreso una politica di sensibilizzazione dell'opinione pubblica e di canalizzazione degli investimenti nel settore della manutenzione e dell'adeguamento delle strade.

Tali interventi potranno migliorare tanto più efficacemente la sicurezza dell'infrastruttura quanto più riguarderanno quegli elementi critici che una preventiva analisi dell'incidentalità indica quali cause determinanti.

Molti studi hanno ormai dimostrato come gli incidenti stradale individuali siano eventi molto complessi causati da una notevole molteplicità di fattori che vedono coinvolti le caratteristiche geometriche della strada, il comportamento di guida dell'utente, le condizioni funzionali della pavimentazione, i fattori umani, etc..

Recenti studi eseguiti in vari paesi europei individuano la causa degli incidenti, per il 18-20% dei casi, nelle caratteristiche dell'infrastruttura. Un notevole numero di incidenti, circa il 50-60% sulle strade statali, provinciali e comunali ed il 30% sulle autostrade, avviene in tratti non rettilinei del tracciato.

Anche il fattore ambientale gioca un ruolo determinante; infatti se ci si riferisce ai dati ISTAT a disposizione si nota, ad esempio, che gli incidenti verificatisi in presenza di pioggia costituiscono circa il 17% dei complessivi.

E', inoltre, opinione comune che il tasso di incidentalità cresca con l'aumentare del volume di traffico, ma la relazione tra i due fattori non è di semplice proporzionalità, essa può variare notevolmente in funzione del tipo di strada.

Infine, vanno considerati i parametri prestazionali della pavimentazione. La diminuzione dell'aderenza disponibile in condizioni di strada bagnata, fa sì che il rischio di instabilità del veicolo aumenti significativamente; d'altra parte gli utenti, in tali circostanze, si comportano con maggiore prudenza. Quindi in presenza di entrambi i fattori un'attendibile previsione del livello di incidentalità diventa difficile. [CAR]

Alla luce di quanto detto, risulta opportuno ricercare correlazioni attendibili tra il livello di sicurezza di un'infrastruttura e i parametri suddetti.

Il lavoro propone una metodologia per valutare le condizioni che compromettono la sicurezza nella circolazione stradale (aumento della probabilità del verificarsi di un incidente) e che quindi sia di supporto per la localizzazione e la scelta degli interventi finalizzati al miglioramento della sicurezza dell'infrastruttura viaria.

2.2 Dati aggregati di incidentalità in Italia nel periodo 1995-2002

Gli incidenti stradali rappresentano uno dei più forti costi sociali che il nostro paese paga in termini non solo di danni materiali ma principalmente di vite umane. Qualsiasi studio per la riduzione di questo fenomeno deve necessariamente partire dall'analisi della sua entità.

I dati analizzati, riferiti al periodo 1995-2002, sono i dati ISTAT riportati nelle Statistiche degli incidenti stradali, relativi agli incidenti avvenuti in tutto il territorio nazionale, e i microdati ISTAT relativi agli incidenti nella Regione Campania.

Tale Istituto fornisce una serie di informazioni riguardanti gli incidenti in cui si sono verificati ferimenti o decessi di persone. Mancano dunque dati per gli incidenti con soli danni materiali. Inoltre, il dato fornito è inferiore a quello reale perché un certo numero di incidenti sfugge alle statistiche. L'Istituto stesso quantifica la sottostima, per il periodo temporale considerato, in un valore compreso tra il 12% e il 23%. Modelli predittivi di incidentalità, però, indicano

spesso un numero di incidenti molto maggiore di quello riportato dall'ISTAT (circa il doppio).

Nel periodo 1995-2000 in Italia si sono verificati 1.198.448 incidenti stradali, con 38.316 morti e 1.714.757 feriti.

2.2.1 Dati incidentalità della Regione Campania

Nel periodo 1995-2000 nella Regione Campania si sono verificati 38.067 incidenti stradali, con 1.639 morti e 62.341 feriti.

Capitolo 3: Il Data Mining

Negli ultimi vent'anni la disponibilità di dati ed informazioni digitali è cresciuta vertiginosamente grazie alla progressiva ed inesorabile diffusione dei computer e al drastico abbassamento del rapporto prezzo/capacità dei supporti di memorizzazione unito ad un aumento della potenza di calcolo dei moderni processori. Infatti non è insolito che un sistema di supporto alle decisioni contenga milioni o anche centinaia di milioni di record di dati.

I sistemi classici di memorizzazione dei dati, i DBMS (database management system), offrono un'ottima possibilità di memorizzare ed accedere ai dati con sicurezza, efficienza e velocità ma non permettono un'analisi per l'estrazione di informazione utili come supporto alle decisioni. Inoltre, le metodologie statistiche classiche sono state concepite con l'obiettivo di 'indurre' dal particolare al generale estendendo, attraverso un processo inferenziale, le informazioni ottenute da un ristretto insieme di dati, il campione, alla popolazione di riferimento.

Oggi la situazione appare capovolta. I dati considerati non rappresentano un campione ridotto ma l'intera popolazione, il problema quindi non è più di tipo inferenziale ma *esplorativo.* Ciò a cui si è interessati è la ricerca di una 'sintesi interessante' dei dati attraverso l'esplorazione multidirezionale e mulidimensionale degli stessi al fine di identificare relazioni non note a priori. Il ricercatore si pone in una condizione di completa ignoranza rispetto al fenomeno da analizzare e indirizza la propria ricerca verso numerose e diverse direzioni al fine di 'scandagliare' i dati alla ricerca della conoscenza.

Un approccio recente è il ***data mining*** che U.Fayyad, G.Piatetsky-Shapiro, P.Smyth, R.Uthurusamy definiscono "*l'insieme delle tecniche per l'analisi dei dati provenienti da grandi dataset finalizzata all'identificazione di relazioni non note, non banali, interpretabili e usufruibili dal soggetto fruitore dell'analisi"*. Questa definizione contiene alcuni concetti chiave che ben sintetizzano gli obiettivi primari di un processi di data mining. In particolare le relazioni scoperte nell'analisi devono essere:

non note, cioè non conosciute a priori;

non banali, cioè non intuibili senza l'ausilio del processo di data mining;

interpretabili, cioè sintetizzabili attraverso tecniche che rendano i risultati comprensibili non solo all'esperto di analisi dei dati, ma anche all'operatore che nel data mining cerca un supporto alla propria attività decisionale;

usufruibili, cioè tali da garantire un adeguato supporto all'attività decisionale del soggetto fruitore in modo da produrre un valore aggiunto che quanto meno giustifichi le risorse investite nel processo di data mining.

In sintesi, il data mining grazie ad un approccio esplorativo evidenzia relazioni che non solo erano nascoste e sconosciute, ma che spesso non si era nemmeno mai ipotizzato potessero esistere.

3.1 Cos'è il Data Mining

La crescita sempre più massiccia della quantità di informazione disponibile e l'aumentata "raggiungibilità" della stessa ha portato allo sviluppo di metodologie e strumenti che permettono di elaborare i dati e ricavarne informazioni non ovvie e di grande importanza per l'utilizzatore finale, sia esso un ricercatore che studia dei fenomeni scientifici o sperimentali, che il manager di una ditta che intende migliorare i processi decisionali nel suo business.

Realizzare questo obiettivo è stato reso particolarmente difficile dall'esplosiva crescita delle dimensioni delle basi dati commerciali, governative e scientifiche.

I sistemi di gestione delle basi di dati (*Data Base Management System*, DBMS) hanno certamente permesso di manipolare i dati in maniera efficace, ma non hanno risolto il problema di come supportare l'uomo nella "comprensione" e nell'analisi dei dati stessi.

Storicamente, lo sviluppo dei metodi statistici ha prodotto un certo numero di tecniche di analisi dei dati utili nel caso in cui si debbano confermare delle ipotesi predefinite. Tali tecniche risultano però inadeguate nel

processo di scoperta di nuove correlazioni e dipendenze tra i dati, che crescono in quantità, dimensione e complessità.

Orientativamente possiamo individuare tre fattori che hanno cambiato il panorama dell'analisi dei dati:

la disponibilità di grande potenza di calcolo a basso costo;

l'introduzione di dispositivi di raccolta automatica dei dati (si pensi, ad esempio, alla strumentazione ad alto rendimento per le rilevazione delle caratteristiche della pavimentazione) insieme alla disponibilità di vaste memorie di massa a basso costo;

l'introduzione di un nuovo insieme di metodi sviluppati nell'ultima decade dalla comunità dei ricercatori in Intelligenza Artificiale. Questi metodi permettono l'analisi e l'esplorazione dei dati, consentendo, tra l'altro, una più efficace rappresentazione della conoscenza

Nasce dunque il data mining[1] cioè "*il processo atto a scoprire correlazioni, relazioni e tendenze nuove e significative, setacciando grandi quantità di dati immagazzinati nei repository, usando tecniche di riconoscimento delle relazioni e tecniche statistiche e matematiche*"(Gartner Group). In altre parole, con il nome data mining si intende l'applicazione di una o più tecniche che consentono l'esplorazione di grandi quantità di dati, con l'obiettivo di individuare le informazioni più significative e di renderle disponibili e direttamente utilizzabili nell'ambito del decision making. L'estrazione delle informazioni significative avviene tramite individuazione delle associazioni, o "patterns", o sequenze ripetute, o regolarità nascoste nei dati. In questo contesto un "pattern" indica una struttura, un modello, o, in generale, una rappresentazione sintetica dei dati.

Vi sono una serie di fattori di natura tecnologica ed economica che sono alla base del fenomeno di data mining. Oltre ai fattori tecnologici già indicati

[1] *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.

precedentemente, è importante sottolineare un fattore economico relativamente nuovo: la *necessità di valorizzare il patrimonio informativo*. Con questo intendo dire che mai come in questo periodo è forte la volontà di far "fruttare" le informazioni che si hanno a disposizione. Questa necessità viene particolarmente sentita nelle organizzazioni a scopo commerciale, basti pensare alle molteplici iniziative di raccolta delle informazioni dei clienti, fatte in modo più o meno "occulto" dalle organizzazioni commerciali.

Un tipico esempio è quello della "tessera sconto": l'organizzazione propone la compilazione di una scheda, che permette alla stessa di ottenere un profilo informativo di ogni cliente.Il cliente, che compila la scheda credendo di ottenerne un beneficio (lo sconto), in realtà sta fornendo all'organizzazione qualcosa di infinitamente più prezioso, l'*informazione*.

Chi farà fruttare meglio l'informazione che ha a disposizione, anche grazie ai metodi di data mining, probabilmente sarà in grado di prevedere, e quindi guidare, o perlomeno non subire, il mercato.

3.2 Obiettivi e Tecniche del processo di Data Mining

Le attività di un processo di data mining possono essere diverse e fornire un elenco esaustivo non è sempre facile. Si può ritenere che un modo corretto di procedere sia quello di far riferimento alle finalità proprie del soggetto fruitore dell'analisi. In quest'ottica i diversi obiettivi e le tecniche impiegate per raggiungerli possono così sintetizzarsi:

1. **Analisi Esplorativa dei Dati (*Exploratory Data Analysis* EDA).** Così come suggerisce il nome, obiettivo è quello di esplorare i dati nelle direzioni più diverse senza avere informazioni a priori sugli stessi. Le tecniche tipicamente utilizzate in questo ambito sono quelle visuali e interattive che consentono di ridurre la dimensionalità di riferimento attraverso l'uso di metodi di proiezione e di rappresentazioni grafiche *ad hoc*. Il problema di questo tipo di tecniche è che, in presenza di dataset caratterizzati da un enorme numero di osservazioni, l'interazione e la rappresentazione visuale tende a diventare macchinosa e poco

comprensibile oltre al richiedere una elevata potenza di calcolo dell'elaboratore elettronico.

2. **Analisi descrittiva (*Descriptive Modelling*).** La finalità è quella di descrivere l'intero dataset (e il processo che ha generato i dati). Ne sono un esempio le tecniche che si occupano della stima della distribuzione che ha generato i dati (*stima della densità*), la partizione di uno spazio *p*-dimensionale in un certo numero di gruppi (*analisi dei gruppi e segmentazione*) e la descrizione, attraverso l'impiego di un modello, delle relazioni esistenti tra un set di variabili (*analisi della dipendenza*).
3. **Le tecniche di Classificazione e Regressione (*Predictive Modeling)*** sono, ad esempio, impiegate con buoni esiti per predire il valore di titoli mobiliari in momenti futuri sulla base dell'esperienza passata e delle determinanti individuate; per determinare la diagnosi di un paziente rispetto ai sintomi e alle caratteristiche socio-demografiche dello stesso; il termine predizione, per questi modelli, va inteso nella sua accezione generale e non di continuità temporale.
4. **Individuazione di pattern e regole ricorrenti (*Discovery Patterns and Rules*)**. Le prime tre tipologie di analisi appena illustrate hanno ad obiettivo principale la costruzione di un modello per finalità diverse. In questo caso, invece, il processo di data mining concerne l'individuazione di pattern[2].
5. **Recupero attraverso un contenuto (*Retrieval by Content*)**. In quest'ultimo caso, l'utente conosce il pattern e l'obiettivo dell'analisi è quindi quello di identificare l'esistenza nei dati di pattern simili. Questo tipo di processo è molto frequente nell'analisi di testi e immagini quando, ad esempio, si vogliono individuare i documenti relativi ad un certo argomento sulla base di parole-chiave.

Dall'analisi degli obiettivi e delle tecniche impiegate nei processi di data mining, è facile intuire che le metodologie statistiche classiche non sono in

grado di supportare al meglio le esigenze dell'analisi, specialmente se si pensa alle enormi dimensioni che le banche dati vanno assumendo giorno dopo giorno. In ogni caso comunque, la statistica gioca un ruolo importante nel data mining: essa infatti è una componente necessaria, ma non sufficiente, di un buon sistema di data mining.

Le principali differenze tra l'analisi statistica e un processo di data mining è proprio la dimensione del dataset. Nella statistica classica, un dataset è ritenuto 'grande' quando contiene dalle poche centinaia ad alcune migliaia di osservazioni. Nelle applicazioni di data mining, banche dati contenenti alcuni milioni di record sono la normalità. Queste banche dati hanno dimensioni nell'ordine di qualche gigabyte fino a raggiungere in alcuni casi i terabyte. Se, inoltre, aggiungiamo che tutti questi dati sono collezionati per scopi diversi dall'analisi statistica e sono quindi necessariamente affetti dalla presenza di valori mancanti, da fenomeni di contaminazione e corruzione dei dati, è facile intuire come il data mining non può essere inteso meramente come una classica esplorazione statistica dei dati, ma rappresenta una nuova disciplina che trae ragione nelle dimensioni e nella tipologia non tradizionale dei dati considerati.

3.3 Data Mining di tipi descrittivo e previsivo: Verification model e Discovery model

Gli approcci al data mining possono essere di due tipi: top-down e bottom-up. Nel primo caso si utilizza la statistica pura come guida per l'esplorazione dei dati, cercando di trovare conferme a fatti che l'utente ipotizza o già conosce, o per migliorare la comprensione di fenomeni parzialmente conosciuti. Tuttavia, un approccio di tipo top-down limita i compiti del data mining alla descrizione.

La sola descrizione dei dati non può fornire quelle informazioni di supporto alle decisioni, cui si fa riferimento quando si parla di potenzialità del

[2] Il termine pattern sta ad indicare l'esistenza di una 'regolarità' nei dati, dove per regolarità si intende la presenza di comportamenti simili delle unità analizzate rispetto alle caratteristiche considerate.

data mining. Di conseguenza, un approccio di tipo bottom-up, nel quale l'utente ricerca informazioni che a priori ignora, risulta sicuramente più interessante ma anche molto più difficile. Questo secondo approccio conduce ad un data mining di tipo previsivo in cui si costruisce uno o più set di modelli, si effettuano delle inferenze sui set di dati disponibili e si tenta di prevedere il comportamento di nuovi dataset. Nel data mining di tipo previsivo per di più si possono identificare due tipi, o modi, di operare, che possono essere usati per estrarre informazioni d'interesse per l'utente: i verification models e i discovery models.

I verification models utilizzano delle ipotesi formulate dall'utente e verificano tali ipotesi sulla base dei dati disponibili. Qui l'utente riveste un ruolo cruciale, in quanto deve formulare delle ipotesi sui possibili comportamenti delle variabili in questione. Tuttavia risultano di gran lunga più interessanti i discovery models, che costituiscono la parte più rilevante delle tecniche di data mining. In questi tipi di modelli all'utente non è affidato nessun tipo di compito specifico, è il sistema che scopre "automaticamente" importanti informazioni nascoste nei dati: si cerca di individuare pattern frequenti, regole di associazione, valori ricorrenti. L'utilizzo di queste tecniche richiede ovviamente enormi sforzi di ricerca ed in questo le maggiori performance dei sistemi informatici adeguati giocano un ruolo importante.

3.4 Sviluppi e campi applicativi

Un numero crescente di aziende ha sviluppato/adottato con successo applicazioni di data mining. Mentre, inizialmente, le prime ad adottare questa tecnologia furono principalmente le aziende operanti in settori definibili "information-intensive" quali, ad esempio, servizi finanziari e direct mail marketing, oggi questa tecnologia è praticamente applicabile ad ogni azienda che intenda trarre vantaggi competitivi dal sul suo patrimonio informativo. Il mercato del data mining ha fatto riscontrare nel 1996 un fatturato globale paria a circa 100 milioni di dollari; gli analisti del META Group, una società statunitense specializzata nell'analisi del mercato delle tecnologie

dell'informazione, prevedono che questo mercato esploderà nel corso degli anni.

In accordo con quanto riportato dal META Group, un recente studio del Gartner Group ha riportato che il data mining risulta oggi classificato in cima alla lista delle aree tecnologiche di interesse aziendale e che, inoltre, nei prossimi 3-5 anni esso assumerà sempre più importanza nel mondo commerciale e finanziario.

Il data mining è oggi applicabile ad una varietà di domini applicativi che spaziano dalla gestione degli investimenti di una finanziaria all'astronomia. La sua importanza è particolarmente riconosciuta in settori quali, vendita massiva al dettaglio, banche, assicurazioni, sanità, e telecomunicazioni. Applicazioni tipiche sono: analisi del mercato per pianificare campagne promozionali (market basket analysis), analisi della vulnerabilità dei clienti (risk analysis), gestione delle relazioni con il clienti (cross-selling), gestione del portafoglio finanziario (portfolio creation), individuazione delle truffe (fraud detection), l'analisi automatica di grosse quantità di testi liberi (text mining) ecc..

Ma il data mining è sfruttabile vantaggiosamente anche in settori quali la ricerca sismica e ambientale, l'elaborazione di immagini (image mining), l'analisi statistica, ecc., dove le necessità in termini di potenza computazionale sono drammaticamente più accentuate che nei settori elencati in precedenza, non solo per la quantità dei dati da analizzare (tipicamente dell'ordine dei giga-tera bytes) ma anche per il fatto che questi tendono ad essere generalmente più complessi e, spesso, richiedo l'applicazione di complessi modelli simulativi.

La crescita esponenziale di Internet ha generato scenari applicativi per il data mining assolutamente imprevedibili solo pochi anni fa: il web mining.

Il commercio elettronico e le altre attività ad esso correlate creeranno enormi quantità di dati utilizzabili per scopi commerciali e sociali. Il data mining, combinato con le applicazioni basate su Internet, assumerà un ruolo fondamentale nel sempre più complesso processo di supporto alle decisioni.

3.5 Data Mining and Knowledge Discovery

Il termine data mining è utilizzato come sinonimo di knowledge discovery in databases (K.D.D.), anche se sarebbe più preciso parlare di knowledge discovery quando ci si riferisce al processo, tipicamente interattivo ed iterativo, di scoperta ed interpretazione della conoscenza a partire dalle informazioni memorizzate in una base di dati e di data mining come una parte fondamentale di tale processo ed in particolare l'insieme di metodi ed algoritmi applicabili per la scoperta e l'estrazione, in modo automatico, dai dati, delle informazioni che non sono immediatamente visibili, data la grande mole e complessità dei dati stessi.

Il processo di KDD è caratterizzato dalle seguenti fasi:

1. *selezione*: partendo dai dati contenuti nel database, detti dati grezzi, si estrae l'insieme dei dati che si ritengono maggiormente significativi per il tipo di analisi che si vuole effettuare;
2. *pre-elaborazione*: in questa fase viene effettuata un'*integrazione* dei dati. In generale, i dati provengono da diverse fonti presentando quindi delle incongruenze quali, ad esempio, l'uso di diverse denominazioni per individuare uno stesso valore che può assumere un attributo. Inoltre questa fase prevede la *pulizia* dei dati, in cui si eliminano eventuali errori, ed il trattamento dei *dati mancanti*;
3. *processo di data mining*: questo processo ha come scopo quello di fornire all'utente finale una rappresentazione della conoscenza che ha acquisito, applicando uno o più metodi di data mining partendo dai dati risultanti dalla fase di pre-elaborazione.

La fase denominata *processo di data mining* è, a sua volta, concettualmente suddivisa in tre punti:

1. *trasformazione o preparazione*;
2. *data mining vero e proprio*;
3. *interpretazione, visualizzazione e previsione*.

3.5.1 Preparazione dei Dati

La fase di preparazione dei dati dipende dal metodo di mining che verrà utilizzato nell'applicazione.

Le attività tipiche di questa fase sono:

- l'introduzione di nuovi attributi mediante l'applicazione di operatori logici e matematici, aumentando così la quantità di informazioni utili disponibili;
- la trasformazione dei dati per essere adattati al metodo di Data Mining che verrà applicato;
- il campionamento o partizionamento dei dati;
- la discretizzazione dei dati.

Le operazioni di trasformazione dei dati sono, in genere, associate a particolari limitazioni dei metodi di mining che si intende adoperare, ad esempio incapacità di gestire contemporaneamente informazione numerica e categorica o valori mancanti.

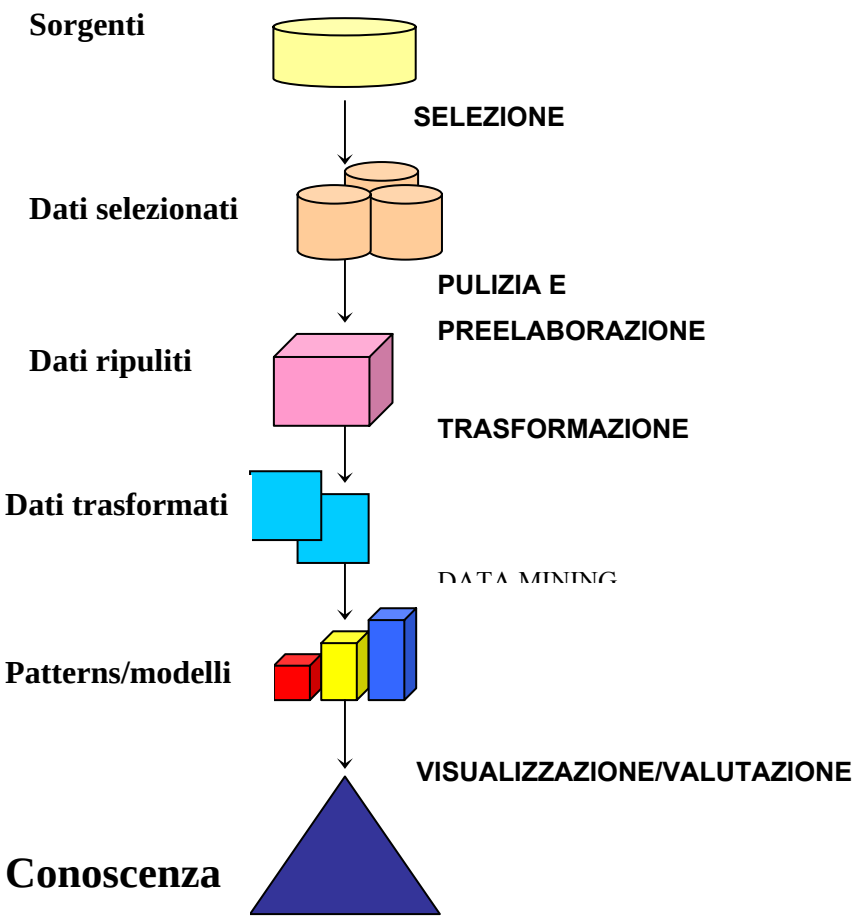


Figura 2.1: Processo di Knowledge Discovery in Databases

3.5.2 Data Mining vero e proprio

Mediante l'applicazione di uno o più metodi o algoritmi vengono determinati i percorsi, eventuali regole e le caratteristiche dei dati.

Se il risultato che si è ottenuto non è soddisfacente, si può operare una nuova trasformazione dei dati, tornando alla fase precedente e applicando nuovamente la fase di data mining, utilizzando lo stesso o un diverso metodo/algoritmo.

Questa fase può essere indicata anche come fase di apprendimento o di esplorazione e modellazione, in base alle denominazioni correntemente in uso del campo dell'Intelligenza Artificiale e della Statistica, rispettivamente.

3.5.3 Visualizzazione, Interpretazione e Previsione

Il risultato della fase di mining è costituito da "conoscenza" indotta dai dati rappresentata secondo un dato formalismo. Tuttavia, prima di utilizzare ai fini pratici tali informazioni, è necessario che queste ultime siano opportunamente validate, ossia si deve verificare se il mining ha prodotto risultati significativi.

Verificato questo aspetto, a seconda della tipologia di applicazione, si presentano almeno due possibili alternative:

- se l'applicazione di data mining era destinata alla previsione, il risultato passa, come si è soliti dire, "in produzione", ovvero viene utilizzato per analizzare nuove situazioni;
- altrimenti, se l'applicazione è di tipo interpretativo, la conoscenza acquisita dal sistema di data mining deve essere opportunamente trattata al fine di poter essere visualizzata ed interpretata da un analista, per ottenere, infine, le informazione necessarie al management in fase di supporto alle decisioni.

Si osservi che in entrambi i casi il risultato del data mining è adoperato per guidare un processo decisionale. La differenza è costituita dal livello e dalle modalità con cui ciò avviene.

3.6 Tecniche di data mining

Esistono differenti tecniche di data mining il cui utilizzo è funzione dell'ambito applicativo a cui è rivolta l'analisi, ma nonostante l'ampia disponibilità sul mercato di algoritmi, le tecniche più affermate e più dettagliatamente descritte nei paragrafi successivi sono:

1. Reti Neurali
2. Alberi di decisione
3. Algoritmi di clustering
4. Reti Bayesiane
5. Regole di associazione
6. Pattern sequenziale

3.6.1 Reti Neurali

L'idea alla base della tecnologia delle reti neurali è quella di tentare di imitare il funzionamento del cervello umano.Per questo motivo la struttura di una rete neurale ha come modello quella di un cervello, formato da *neuroni* collegati tra di loro attraverso *dendriti* ed *assoni*. Introduciamo la struttura funzionale del cervello umano, per comprendere meglio le scelte progettuali nella tecnologia delle reti neurali.

Il Cervello Umano

Da un punto di vista strutturale, e molto schematicamente, il cervello umano è composto da particolari cellule nervose, dette *neuroni*. I neuroni si compongono di due tipi di prolungamenti: numerose ramificazioni dette *dendriti* ed una singola lunga fibra detta *assone* (fig. 2.2). L'assone permette al neurone di comunicare verso altri neuroni, mentre i dendriti permettono al neurone di ricevere

comunicazioni dagli altri neuroni. La trasmissione avviene per mezzo della *sinapsi*. Nella sinapsi il neurone trasmettitore rilascia, per mezzo dell'assone, una sostanza (*neurotrasmettitore*) che può far aumentare o diminuire il potenziale elettrico del neurone ricevente (*sinapsi eccitatoria* o *sinapsi inibitoria*).In questo modo il potenziale di un neurone varia, istante per istante, a seconda del tipo di sinapsi che riceve.

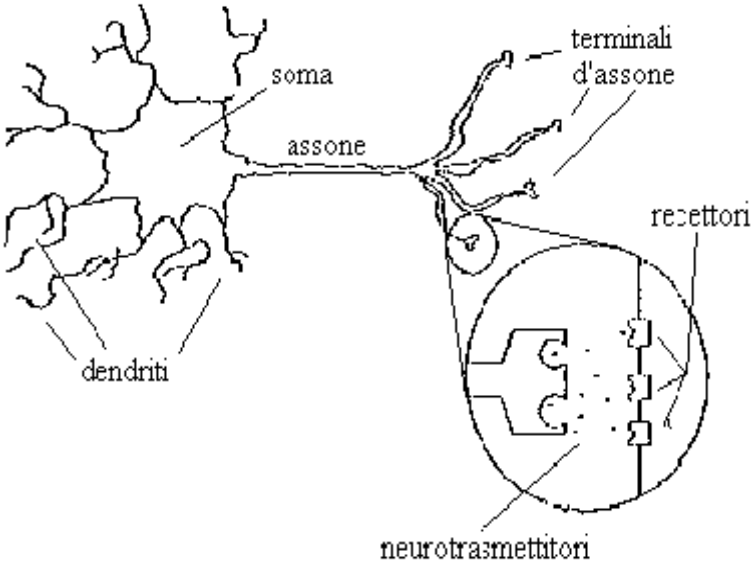


Figura 2.2: Le diverse parti di una cellula nervosa con,in evidenza,una sinapsi.

Nel momento in cui il potenziale di un neurone supera il proprio *livello di soglia*, esso si "attiva" generando nuove trasmissioni sinaptiche verso i neuroni ad esso collegati. Il numero dei neuroni che possono essere collegati, tramite le ramificazioni dell'assone, ad un altro neurone, possono variare da una dozzina a centinaia di migliaia. I neuroni formano anche nuove connessioni con altri neuroni e, talvolta, interi gruppi di neuroni possono migrare da un punto all'altro.

Si ritiene che questi meccanismi costituiscano i fenomeni di base nel processo di apprendimento del cervello.

Struttura Neurale Artificiale

Una rete neurale è costituita da un insieme di *nodi* di elaborazione (detti anche *unità*). Ciascuna unità è collegata alle altre unità attraverso uno o più archi di uscita ed uno o più archi di ingresso.

Ogni unità emette sugli archi di uscita il suo **livello di attivazione** a_i.

Ciascun arco ha associato un **peso** $W_{j,i}$. Il peso svolge la funzione che il neurotrasmettitore svolge nel neurone e viene utilizzato nel processo di apprendimento.

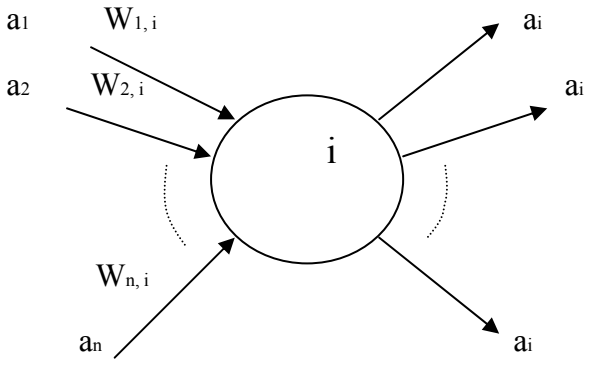


Figura 2.3: Un'unità i

Istante per istante (in realtà, ad ogni ciclo), ciascun nodo calcola il valore in_i di una **funzione di ingresso**, analogo al valore del potenziale elettrico in un neurone, infatti:

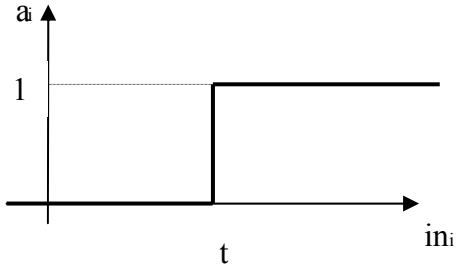
$$in_i = \sum_{j=1}^{n} W_{j,i} \cdot a_j$$

Il livello di attivazione a_i sarà funzione dello pseudo-potenziale interno al nodo:

$a_i = g(in_i)$

La funzione g() è detta **funzione di attivazione**, ed è, tipicamente, non lineare.

In genere è una funzione gradino:



Ossia:

$$step_t(x) := \begin{cases} +1 & se \quad x \geq t \\ 0 & se \quad x < t \end{cases}$$

Il valore *t*, detto di *soglia*, rappresenta il minimo valore che la somma pesata degli input (in_i) deve raggiungere affinché il nodo sia attivato.

Normalmente la soglia è sostituita con un peso aggiuntivo, in modo tale che nella fase di addestramento si devono modificare solo i pesi e non pesi e soglie.

In sostanza, in luogo della funzione $step_t(x)$ si usa una funzione $step_0(x)$, avendo cura di introdurre, per ogni nodo *i*, un ingresso fittizio:

$W_{0,i} \cdot a_0 = -t$

tale per cui:

$$step_t\left(\sum_{j=1}^{n} W_{j,i} \cdot a_j\right) = step_0\left(\sum_{j=0}^{n} W_{j,i} \cdot a_j\right)$$

In luogo della funzione gradino (*step*) è possibile utilizzare le funzioni segno e sigmoide:

- funzione segno:

$$sign(x) := \begin{cases} +1 & se \quad x \geq 0 \\ -1 & se \quad x < 0 \end{cases}$$

- funzione sigmoide:

$$sigmoid(x) := \frac{1}{1 + e^{-x}}$$

Interconnessioni

I nodi sono interconnessi e disposti su diversi livelli (o *strati*) in cui ogni nodo del livello *i* riceve come input il risultato dei nodi del livello (*i*-1) e l'output dei nodi del livello *i* costituisce l'input per i nodi del livello (*i*+1).

Alcuni di questi nodi (detti *nodi di input* e *nodi di output*) sono connessi con il mondo esterno, altri nodi invece si interfacciano solo con altri nodi, per questo sono chiamati *nodi nascosti*.

Abbiamo visto che ogni connessione ha associato un peso, questo rappresenta la "conoscenza" acquisita dal nodo.

.

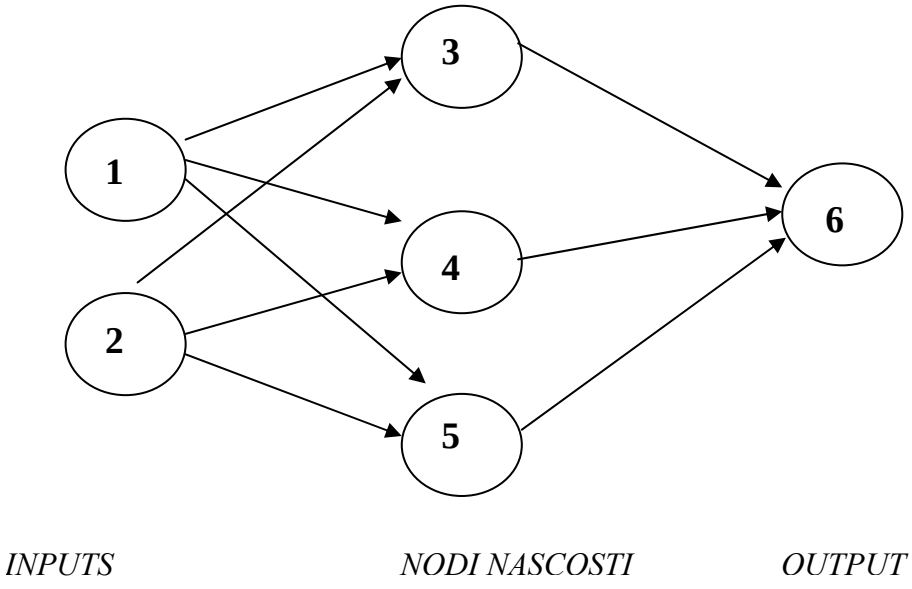


Figura 3.4: Rete neurale con uno strato interno

Algoritmo di Training

L'algoritmo di training della rete consiste sostanzialmente nell'individuazione di una buona stima per i pesi $W_{i,j}$: all'inizio la rete non contiene nessuna informazione, si assegnano i valori dei pesi in modo casuale, successivamente si presentano gli ingressi, si calcola l'uscita della rete, si calcola l'errore, come differenza tra il valore ottenuto e i valori che si volevano, e si

aggiustano i pesi in modo da far diminuire l'errore. Per aggiustare i pesi esistono diverse regole a seconda del tipo di rete, per una rete senza nodi nascosti si utilizzano la formula seguente:

$$W_j = W_j + \alpha \cdot I_j \cdot (T - O)$$

dove Wj è il peso del collegamento j-esimo, *alfa* è una costante chiamata "velocità di apprendimento" (learning rate), Ij è l'ingresso j-esimo, T è l'uscita i-esima desiderata e O è l'uscita i-esima della rete. La costante alfa viene utilizzata per diminuire la velocità di convergenza, in modo da limitare oscillazioni attorno al minimo di errore (l'errore è rappresentato da T-O). Reti di questo tipo con funzioni di attivazione lineari (chiamate *adalines*) vengono impiegate spesso nel riconoscimento dei caratteri o nella costruzione di filtri adattivi.

Se la rete contiene strati di neuroni nascosti è necessario utilizzare per l'addestramento la tecnica di "backpropagation", grazie alla quale si riesce a suddividere l'errore all'uscita tra i neuroni dei livelli nascosti e quindi a variare i pesi in modo da minimizzare l'errore. Nell'addestramento di una rete neurale è importante presentare tutte le coppie di ingresso-uscita diverse volte. Ciascuna presentazione viene chiamata "epoca" di addestramento. Se l'addestramento procede correttamente ad ogni epoca l'errore (quadratico) medio su tutte le uscite dovrebbe diminuire. L'algoritmo di training può essere schematizzato come segue:

Ripeti per n epoche
    Per ogni esempio di addestramento
        Presenta ingressi
            Propaga in avanti
            calcola errore (T-O)
        Ritocca pesi
    Fine ciclo esempi
Fine ciclo epoche

Per utilizzare una rete neurale nella risoluzione di un particolare problema è opportuno cercare di identificare gli ingressi e le uscite che riescano a caratterizzare bene il problema. Riguardo alla topologia della rete da usare non si può dire molto, in quanto non sono ancora disponibili teorie riguardo al numero ottimale di neuroni e collegamenti da utilizzare. Un risultato importante è che le reti con un solo livello nascosto riescono a rappresentare tutte le funzioni continue, mentre le reti con due livelli riescono a rappresentare anche quelle discontinue. Le prestazioni di tutte le reti neurali dipendono molto dal set di addestramento: più il set è rappresentativo del problema e più è completo più le prestazioni saranno migliori e la anche la capacità di generalizzare della rete sarà migliore.

Vantaggi e Svantaggi

Le reti neurali, con architetture specifiche, possono essere applicate, praticamente, a tutti i problemi di apprendimento supervisionato e non supervisionato, dimostrando, in molti casi, prestazioni, dal punto di vista dell'accuratezza e della generalizzazione, nettamente superiori ad altri formalismi.

Tuttavia, presentano anche una serie di svantaggi, in particolare:

- elevato costo computazionale dell'apprendimento anche nel caso di soluzioni approssimate;
- difficile trattamento dell'informazione non numerica;
- opacità del modello costruito, in quanto non è possibile, nella maggior parte delle architetture delle reti neurali, correlare direttamente i pesi delle connessioni con caratteristiche degli input.

Le reti RBF (*Radial Basis Function*) e SOM (*Self-Organized Map*) sono due particolari strutture di reti neurali diffuse nell'ambito del data mining.

3.6.2 Alberi di Decisione

Gli *alberi di decisione* sono un tecnica molto utilizzata per rappresentare modelli di classificazione e regressione.

Il modello risulta molto intuitivo e facilmente comprensibile, inoltre esistono diversi algoritmi, utilizzabili nelle applicazioni reali, per costruire automaticamente alberi di decisione.

Un albero di decisione contiene due tipi di nodi:

- i nodi foglia che sono associati ad una classe 'C';
- i nodi intermedi che sono associati ad un attributo 'F' ed hanno, al più, tanti nodi figli quanti sono i possibili valori che l'attributo 'F' può assumere.

Per classificare un elemento si parte dal nodo radice dell'albero seguendo il cammino contrassegnato dai valori dei vari attributi fino a giungere ad una foglia che indica la classe di appartenenza dell'elemento in esame.

Tipi e Costruzione di Alberi di Decisione

Tra le varianti del modello si segnalano gli *alberi di modelli* e gli *alberi di regressione* adoperati per trattare i casi in cui gli attributi e/o la classe siano espressi da valori numerici. Più precisamente un albero di regressione associa ad ogni foglia un valore numerico costante che approssimi adeguatamente la funzione, mentre un albero di modelli vi pone una funzione d'interpolazione lineare. Per quanto riguarda, invece, gli attributi, questi vengono discretizzati secondo varie tecniche. Dato un albero di decisione è banale l'estrazione di regole di classificazione del tipo "if-then" mediante una visita in profondità della struttura. I vari approcci proposti in letteratura per la costruzione di alberi di decisione possono essere visti in modo unificato come un partizionamento ricorsivo dell'insieme (o multi insieme) degli esempi.

Si possono distinguere due fasi nel processo di costruzione di un albero di decisione:

1. crescita (*growing* o *building*);

2. potatura (*pruning*).

Durante la fase di **crescita** si procede con la suddivisione ricorsiva degli esempi in base ad un determinato criterio (*criterio di splitting*) finché non si soddisfa una specifica condizione di terminazione (*criterio di stop*).

La fase di **potatura**, invece, prevede un'ulteriore manipolazione dell'albero al fine di migliorarne le capacità di generalizzazione.

Infatti questo formalismo presenta notevoli problemi legati al rischio di sovra-addestramento, pertanto è fondamentale non solo utilizzare un dataset di validazione per verificare le capacità di generalizzazione del modello, ma eventualmente correggerlo.

Vantaggi e Svantaggi

Concludendo si possono riassume i vantaggi e gli svantaggi di questo formalismo.Tra i vantaggi abbiamo:

- regole esplicite e modulari: la conoscenza acquisita è esprimibile facilmente sotto forma di costrutti 'if-then', inoltre ogni regola è analizzabile indipendentemente dalle altre;
- trattamento dei valori nulli;
- efficienza computazionale;
- robustezza: è possibile estrarre dei buoni modelli anche con un elevato livello di rumore nei dati;
- individuazione delle variabili di input rilevanti.

Mentre i punti deboli sono:

- gestione attributi numerici (sia target, sia di input): le prestazioni degli alberi di regressione non sono altrettanto buone quanto quelle degli alberi di classificazione, inoltre gli attributi numerici richiedono sovente una discretizzazione preliminare;
- formalismi di partizionamento: soprattutto nel caso di attributi numerici, il vincolo di partizionare lo spazio di input con piani paralleli agli assi coordinati può rappresentare una serie limitazione del potere espressivo in

quanto non consente di individuare la presenza dei cosiddetti effetti principali;

- instabilità: piccole variazioni nei dati possono produrre effetti notevoli sugli alberi prodotti, in quanto le differenze tra i vari split in competizione al momento della scelta non sono eccessivamente marcate.

3.6.3 Clusters ed Algoritmi di Clustering

Un *cluster* è un raggruppamento omogeneo di elementi realizzato in base ad un determinato criterio. Il processo di apprendimento non supervisionato (o autonomo) che porta alla costruzione dei cluster è indicato come clustering.

Sebbene il problema del raggruppamento possa essere risolto con diverse tecniche inquadrabili in contesti più ampi (per esempio reti neurali, problem solving), sono stati introdotti, dalla comunità scientifica, un elevato numero di algoritmi ad hoc per il clustering.

Si parla di *clustering geometrico* o spaziale nel caso in cui gli attributi siano di tipo numerico, pertanto è possibile associare ad ogni esempio un punto in un opportuno spazio coordinato.

Nel caso, invece, in cui gli attributi siano di tipo categorico si è soliti utilizzare la denominazione di *clustering concettuale*.

La differenza tra i due casi è riconducibile ad un'opportuna formulazione della funzione **metrica**.

Per quantificare in modo automatico la similarità tra due elementi o gruppi è necessario introdurre un concetto di distanza nello spazio degli esempi, ovvero bisogna ricorrere ad una qualche metrica.

In generale sia S lo spazio degli esempi allora una distanza è una funzione:

$$\|\cdot - \cdot\| : S \times S \to \Re$$

In particolare nel caso in cui lo spazio degli esempi sia $\Re^n$ (con n > 0), ovvero gli attributi siano di tipo numerico, si ricade nell'ambito trattato dalla

geometria, pertanto sono disponibili numerose proposte di distanze studiate in modo molto dettagliato in letteratura.

Esempi di esse sono:

- la distanza di *Minkowski* (con le varianti: distanza di *Hamming* o di *Manhattan,* distanza *euclidea* e distanza di *Tchebyshev*);
- La distanza di *Mahalamobis* (con le varianti: distanza di *Bhattaharaya* e distanza *euclidea*).

Queste metriche hanno delle caratteristiche comuni, in particolare:

- la tendenza ad individuare regioni convesse,
- inapplicabilità al caso di attributi non numerici (nominali),
- notevole peso computazionale (anche perché sono funzioni usate molto frequentemente negli algoritmi di clustering).

In letteratura esistono diversi algoritmi di clustering. Tra essi citiamo:

- **clustering gerarchico**: si parte da n (numero punti da raggruppare) cluster, ognuno contenente un solo punto. Al passo *i* si uniscono i due cluster più "vicini" secondo un opportuno criterio di distanza. Il ciclo si arresta quando il numero dei cluster è $k < n$ voluto;
- **K-Means**: adotta un approccio di tipo "greedy" (goloso) realizzando sostanzialmente una discesa lungo il gradiente, per raggruppare n vettori in $k < n$ insiemi. Rappresenta ogni gruppo con il suo baricentro o *punto medio*;
- **PAM** (Partitioning Around Medods): concettualmente simile al K-Means, rappresenta ogni gruppo con il suo elemento più vicino (il mediano o *medoids*). Introdotto da Kaufmann e Rousseeuw;
- **CLARA** (Clustering Large Applications): variante del PAM, per adattare questo algoritmo ai contesti di data mining. Viene preventivamente effettuato un campionamento degli elementi da raggruppare. Introdotto da Kaufmann e Rousseeuw;

Ognuno di questi algoritmi presenta proprie caratteristiche ed è applicabile in determinati contesti.

Questo suggerisce di analizzare in maniera completa il problema, prima di scegliere l'algoritmo di clustering più adatto.

3.6.4 Reti Bayesiane

Le reti di Bayes sono modelli grafici di probabilità in cui i nodi rappresentano variabili aleatorie e gli archi le dipendenze casuali fra le variabili.

Una rete bayesiana è dunque un modello grafico che codifica la distribuzione congiunta di probabilità di un insieme di variabili aleatorie X $=\{X_1,\ldots,X_n\}$. Questa consiste in:

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

La mancanza di un arco tra due nodi riflette la loro indipendenza condizionale. Al contrario, la presenza di un arco dal nodo X_i al nodo X_j può essere interpretata come il fatto che X_i sia causa diretta di X_j.

Data una struttura S e le distribuzioni locali di probabilità di ciascun nodo $p(X_i|Pa_i)$, dove Pa_i rappresenta l'insieme di nodi padri di X_i, la probabilità di distribuzione congiunta p(X) si ottiene da:

$$p(\mathrm{X})=\prod_{i=1}^{n} p(X_i|Pa_i)$$

ed è evidente come la coppia (S, P) codifichi in modo univoco p(X).

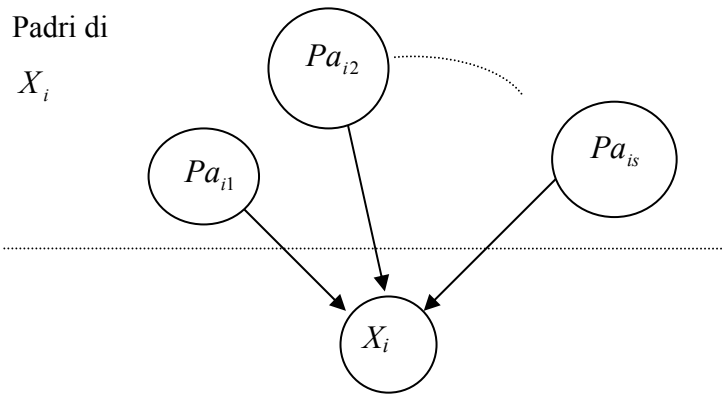


Figura 1.6: Schema della relazione fra un nodo ed i padri in una generica rete bayesiana

Apprendimento di una rete bayesiana

Il processo di learning di reti bayesiane comprende:

- **learning structure** (o **structural learning**): apprendere la struttura della rete ovvero le relazioni fra le variabili;
- **learning parameters**: apprendere i parametri[3]; apprendimento delle probabilità condizionate.

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

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

[3] 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 $\theta = p\ (X = x)$.

b) il database di informazioni

- sia *completo*
- presenti *missing value* (dati mancanti) o *hidden value* (variabile non presente nel database ma semplifica lo studio del dominio).

A seconda che i dati siano completi o incompleti e che la struttura sia nota si utilizza l'algoritmo più appropriato.

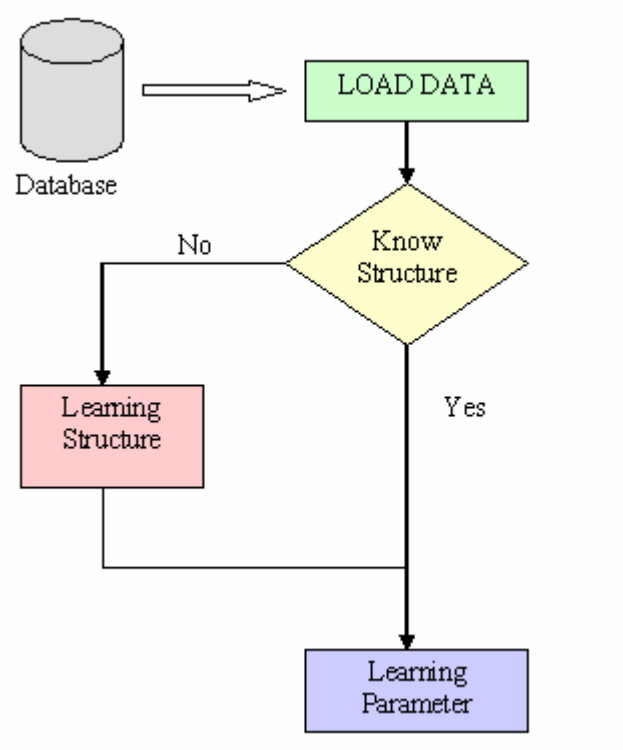


Figura 2.8: Scenario dell'apprendimento

.

	Unknown Structure	***Know Structure***
Complete database	**Structural &Parameter Learning** (Algoritmo Bayesiano, K2, MDL[4], Tpda[5])	**Learning parameter**
Missing value or Hidden variable	Algoritmo PC, Structural EM	EM[6] algorithm (learning parameter)

Tabella 2.3:Algoritmi per il learning: scenario

Lo 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[7].

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.

[4] Minimum description length
[5] Three phase dependence analysis
[6] Expectation maximizaion
[7] Un dominio indica un insieme di variabili (aleatorie) con cui modellare un problema in condizioni di incertezza.

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 sulle reti bayesiane, sullo structural learning e su alcuni algoritmi saranno approfonditi nel prossimo capitolo.

Perché le reti bayesiane

Le reti bayesiane sono uno strumento sempre più flessibile ed adatto per il problem solving e diversi sono i motivi che portano alla scelta delle stesse per estrarre informazioni dai dati:

1. 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. *L'uso di reti Bayesiane permette di risolvere problemi simili anche quando non è disponibile nessun esperimento a riguardo.*
2. 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) e la possibilità di apprendere le probabilità condizionate rendono la codifica della *priori knowledge* particolarmente chiara.
3. I metodi Bayesiani, in congiunzione con le reti Bayesiane, offrono un approccio efficiente per evitare l'overfitting[8] dei dati.

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

3.6.5 Regole di associazione

Siano LHS[9] e RHS[10] dei "pattern" di classificazione dei fatti, espressi con un linguaggio opportuno (ad esempio il formalismo logico).

Le regole di associazione vengono espresse con un formalismo del tipo:

LHS $\Rightarrow$ RHS

L'interpretazione di questa regola è: il verificarsi di un evento in accordo con LHS, rende probabile il verificarsi di un evento in accordo con RHS.

Ragionando in termini di record e tabelle, avere individuato nella base di dati un record (o una sequenza di records) che rispetta LHS, rende probabile trovare nella stessa base di dati un record (o un insieme di records) che rispetta RHS.

Le regole di associazione sono classificate utilizzando due parametri che misurano la "bontà" delle regole stesse.

I parametri in questione sono:

1. **confidenza**: rapporto tra il numero di eventi individuati da LHS e RHS ed il numero di eventi individuati da LHS. Esprime quanto è *attendibile* la regola;
2. **supporto**: rapporto tra il numero di eventi individuati da LHS e RHS ed il numero totale di eventi considerati. Esprime l'*importanza* della regola.

Questi parametri sono generalmente espressi in percentuale.

Il caso più noto in letteratura di studio di associazioni è il cosiddetto *market basket analysis*[11], ovvero l'analisi degli acquisti dei clienti. Ogni fatto è costituito dall'insieme degli articoli acquistati da un determinato cliente nel corso di una transazione commerciale, pertanto le regole trovate devono essere lette come: "se il cliente … ha acquistato … allora è probabile che acquisterà anche …". È facile immaginare come tali regole possano essere impiegate per effettuare predizioni in fase di pianificazione di manovre di marketing.

---

[9] Left Hand Side
[10] Right Hand Side
[11] La denominazione letteralmente traducibile con "analisi del carrello della spesa" ha una giustificazione storica riconducibile al fatto che le prime applicazioni di questo tipo erano destinate alle catene di grande distribuzione.

Tuttavia, affinché tali analisi abbiano successo è necessario seguire alcuni accorgimenti.

In particolare, siccome gli algoritmi sono particolarmente sensibili al rumore dei dati, bisogna scegliere in modo opportuno i valori minimi di confidenza e di supporto per le regole che si vogliono determinare, infatti, una scelta di valori troppo bassa porterà alla determinazione di regole inutili, insignificanti se non addirittura errate, viceversa valori troppo alti possono portare alla produzione di un numero basso, al limite nullo, di regole.

Per ottenere un buon livello di generalizzazione ed evitare di essere sommersi da troppe regole specifiche è consigliabile utilizzare delle gerarchie di classificazioni degli oggetti.

L'algoritmo di scoperta di regole di associazione più diffuso è *Apriori*.

3.6.6 Pattern Sequenziale

Con *pattern sequenziale* si intende un "motivo" ricorrente (detto anche *percorso*) all'interno di una sequenza di eventi. In altre parole si tratta di una sequenza di eventi che si ripete in maniera ricorrente nella base di dati.

Esistono vari modi per rappresentare un pattern, anche in funzione del formalismo adottato per rappresentare gli eventi.

Una possibile scelta è quella di rappresentare un pattern come una sequenza di elementi, che a loro volta possono essere utilizzati per individuare un evento. Ad esempio se gli eventi sono stringhe, un elemento del pattern può essere un'espressione regolare.

Un'altra scelta può essere quella di rappresentare sia gli eventi che gli elementi del pattern come insiemi di elementi su un qualche dominio.

Più formalmente, se S è una sequenza:

$S = <s_1, s_2, \ldots, s_n>$

e P è un pattern:

$P = <p_1, p_2, \ldots, p_m>$

allora P compare in S ($P \subseteq S$) se e solo se esiste una sequenza I di m interi:

$I = <i_1, i_2, \ldots, i_m>$

tali che:

$i_1 < i_2 < \ldots < i_m$ e,

$\forall\ i_j \in I$ (quindi $j = 1, 2, \ldots, m$): $p_j \subseteq s_{i_j}$.

Questa modellazione è particolarmente utile se si vogliono rappresentare le transazioni commerciali dei clienti, infatti, è sufficiente scegliere come dominio degli elementi l'insieme dei prodotti, quindi associare ad ogni cliente una sequenza avente per eventi l'insieme dei prodotti acquistati in una singola transazione.

Si introduce il concetto di *supporto* di un pattern s come la percentuale di sequenze c del database che contengono tale pattern. Un pattern con un elevato supporto viene detto *large sequence*.

L'obiettivo di un processo di mining è determinare tutte le large sequence con un supporto mining.

3.7 Tools di Data Mining

In questo paragrafo andremo ad elencare ed analizzare brevemente entrando più o meno nello specifico alcuni differenti software che permettono di fare data mining.

3.7.1 Enterprise Miner™

SAS System offre una soluzione di business completa e integrata per il data mining: il software Enterprise Miner™ che si è classificato positivamente in un checklist predisposto dal Gartner Group per la valutazione delle funzionalità di un prodotto di datamining.

Le funzionalità di Enterprise Miner comprendono un vasto repertorio che copre tutto il processo di data mining secondo la metodologia SEMMA.

SAS suggerisce come approccio pratico al processo data mining una metodologia suddivisa in cinque fasi rappresentate dall'acronimo SEMMA.

La metodologia SEMMA, che sta per Sample, Explore, Modify, Model e Assess, rende facile all'analista di business l'applicazione di tecniche di

esplorazione statistica e di visualizzazione, la selezione e la trasformazione delle variabili più importanti, la loro modellazione e la conferma della validità del modello scelto.

Fasi della metodologia SEMMA:

• **Sample:** è la fase nella quale viene estratta una porzione di dati abbastanza grande per contenere ancora informazioni significative, e abbastanza piccola per analizzarla velocemente;

• **Explore:** l'esplorazione dei dati serve per scoprire in anticipo relazioni e anomalie nei dati e per capire quali possono essere quelli di interesse;

• **Modify:** serve per creare, selezionare e trasformare le variabili, al fine di mettere a punto il processo di costruzione del modello;

• **Model:** in questa fase vengono ricercate automaticamente le variabili significative e i modelli che forniscono le informazioni contenute nei dati;

• **Assess**: è la fase finale in cui viene valutata l'utilità e l'affidabilità delle informazioni scoperte nel processo di data mining. In questa fase vengono portate nell'ambiente di produzione le regole estratte dai modelli.

Il processo SEMMA è di per sé un ciclo le cui fasi possono essere sviluppate interattivamente come desiderato. I progetti che seguono questa metodologia possono analizzare milioni di record e rivelare relazioni che permettono agli analisti di raggiungere gli obiettivi di datamining.

mobili.

3.7.2 Clementine

Clementine è un tool di analisi multistrategico, che utilizza la visualizzazione come supporto all'attività di data mining.

Questo tool, prodotto dalla SPSS S.P.A., risulta un sistema completo infatti supporta ogni aspetto del data mining, dall'accesso ai dati (Oracle, Ingres, Sysbase, Informix, fogli elettronici, ecc.), alla manipolazione e l'utilizzo di diverse tecniche di data mining.

Clementine viene paragonato, dagli stessi sviluppatori, come un Sistema Informativo Esecutivo (EIS), in quanto permette di estrarre dal database i dati

selezionati, manipolarli e visualizzare i trend e le relazioni. A differenza di molti EISs, comunque, Clementine è un sistema aperto e configurabile.

Il tool è stato studiato per essere utilizzato da una utenza non esperta, riesce così a nascondere bene la sua complessità, permettendo all'utente di concentrarsi solo sugli obiettivi che intende raggiungere.

Clementine è in grado di visualizzare i dati in diversi formati, come scatterplot, istogrammi e web relationships. La visualizzazione è interattiva e permette lo zooming dei dati.

3.7.3 i.Decide™ Web Success

i.Decide™ Web Success, proposto dall' ASCENTIAL SOFTWARE è la soluzione per monitorare, tracciare, analizzare il traffico degli accessi ai siti: permette la raccolta dei e l'analisi dei dati relativi all'utilizzo del sito (sia Internet che Intranet) e alla navigazione degli utenti, memorizzati nei Web Server e nei Proxi Server. Infatti, tutti i click degli utenti, in termini di pagine visitate o contenuti specifici richiesti (immagini, audio, ecc.), vengono registrati ed organizzati in file log insieme ad una serie di informazioni descrittive (data e ora, indirizzo IP dell'utente ecc.).

L'architettura della soluzione prevede un database relazionale opportunamente organizzato, che permette l'archiviazione e la storicizzazione di tutte le informazioni che saranno oggetto dell'analisi; componenti per estrarre, trasformare e caricare i dati e componenti per la visualizzazione intuitiva dei dati e delle analisi.

Avvalendosi di uno strumento ETL (Extract, Transform, Load, nello specifico Data Stage) i.Decide Web Success gestisce il processo di estrazione dei dati contenuti nei log dei Web Server e dei Proxy Server, di trasformazione degli stessi e di caricamento nella base dati specializzata per l'analisi. Il database garantisce a i.Decide un livello prestazionale adeguato per interrogazioni complesse su grandi volumi di dati ed al contempo garantisce agli utilizzatori della soluzione la massima flessibilità di analisi, con la possibilità di approfondire il livello di dettaglio fino al singolo "click" di ogni singolo visitatore. La visualizzazione gabellare e grafica delle informazioni, che sintetizzano le modalità

di consultazione del sito ed i comportamenti dei visitatori è immediata, intuitiva ed espressamente dedicata ai vertici aziendali che possono così disporre di un vero e proprio cruscotto decisionale. Inoltre, le funzionalità di visualizzazione possono essere ulteriormente arricchite grazie all'integrabilità di i.Decide con i più diffusi strumenti di visualizzazione analitica.

i.Decide si integra sia con siti e portali già esistenti, sia con le soluzioni orientate al commercio elettronico, con i sistemi di gestione dei contenuti multimediali e con qualsiasi sistema orientato al rapporto con i clienti ed al marketing personalizzato.

i.Decide™ Telco Success for Wireless è invece la soluzione progettata per la soddisfazione delle esigenze di analisi e supporto decisionale degli operatori delle telecomunicazioni.

3.7.4 FALCON™

FALCON™ e' il prodotto leader mondiale per il controllo e prevenzione in tempo reale di frodi con carte di credito e/o debito. Falcon, prodotto dal HNC SOFTWARE, aiuta quelle imprese che lo utilizzano a considerare tre fonti principali di informazione: la transazione, l'informazione sul proprietario ed i dati relativi alla spesa. L' informazione raccolta e' trasformata in formule matematiche intelleggibili solo al computer che delineando così solo le transazioni lascia anonimi i consumatori. Falcon utilizza in seguito un particolare tipo di rete neurale brevettato per "capire" quelle relazioni tra le centinaia di variabili possibili che hanno un senso e che si spingono spesso oltre l'intuito umano.

Lo scopo e' quello di abbassare al minimo il numero dei falsi allarmi e quindi aumentare il grado di efficienza del sistema in termini di precisione e velocità.

Un sistema aggiuntivo "ad hoc" permette inoltre ai responsabili di aggiustare ed adattare Falcon alle condizioni e caratteristiche specifiche dell'impresa che lo utilizza, per esempio creando o riaprendo casi sospetti basandosi sul punteggio o "score" del sistema od altri parametri chiave. Clienti di Falcon sono 16 dei 25 principali fornitori di carte di credito e/o debito al mondo

per un volume di circa 300 milioni di carte(circa 2/3del traffico mondiale). Si utilizza un server SUN/UNIX.

3.7.5 eFALCON™

eFALCON™, prodotto dal HNC SOFTWARE , e' un potente servizio anti frodi e di controllo del rischio per i negozi on-line. Sostanzialmente si utlizza la stessa tecnologia di Falcon per produrre uno punteggio (score) e quindi accettare o rifiutare un determinato acquisto. Funziona così. Per ogni transazioni si valutano le probabilità di frode in base alle conoscenze di diverse variabili "matematiche" pertinenti al cliente, al prodotto che si vuole comperare ed al venditore. Per esempio due transazioni per lo stesso prodotto possono essere una accettata e l'altra rifiutata perchè, tra l'altro (in realtà sono moltissime le relazioni che il sistema può "vedere") quest'ultima viene effettuata per esempio da un cliente il cui indirizzo di spedizione è cambiato troppo spesso nelle ultime settimane, oppure perchè proveniente da regioni o paesi noti per frodi su quel tipo di prodotto effettuate in quel modo, a quell' ora e con quell' indirizzo di spedizione. Come si può notare,e' un sistema probabilistico raffinato che cerca di eliminare al massimo "problemi" senza tagliare fuori troppo grossolanamente quegli acquisti inizialmente sospetti ma alla fine legittimi. Il sistema e' in grado di evitare oltre il 50% delle frodi online rifiutando meno del 5% delle transazioni buone.

Capitolo 4: Apprendimento di Reti Bayesiane

La rivoluzione dell'informazione avvenuta negli ultimi cinquanta anni ed ancora in atto, ha portato la civiltà verso una nuova era. La continua gara all'innovazione ha condotto a dei progressi mai visti in precedenza e la conseguente moltiplicazione delle potenzialità intellettive ha investito l'intero campo delle scienze. L'evoluzione degli elaboratori ha consentito la creazione di programmi di notevole utilità che rendono oramai il calcolatore una realtà onnipresente.

Il calcolatore, che in origine era considerato un dispositivo per le rilevazioni contabili, si è poi evoluto per elaborare tutti i tipi di informazione (parole, numeri, grafica, suoni).

I metodi tradizionali per il trattamento dell'informazione non sono più adatti a soddisfare le esigenze sempre più complesse e sofisticate dell'utente. La programmazione classica è basata sulla conoscenza che il progettista ha del dominio in esame. Spesso, però, il sistema da modellare è molto complesso ed in costante cambiamento ed il progettista può avere solo una conoscenza incompleta dell'ambiente che rappresenta. In un contesto così eterogeneo ed articolato, serve un qualche criterio intelligente in grado di emulare delle forme di ragionamento e di apprendere dall'esperienza fornita dal dominio dell'applicazione.

4.1 Intelligenza Artificiale e Apprendimento Automatico

E’ in questo contesto che si inserisce l'Intelligenza Artificiale ed in particolare il settore dell'Apprendimento Automatico che si occupa di realizzare dei programmi che imparano dall'esperienza. Infatti, da quando sono stati inventati i calcolatori, si è sempre cercato di capire se fosse stato possibile realizzare delle macchine in grado di imparare. L'intelligenza artificiale è una delle discipline più recenti, nata formalmente nel 1956 quando ne fu coniato il nome ma già operativa da circa cinque anni. Sono state proposte innumerevoli definizioni di cosa dovrebbe essere l'intelligenza artificiale, ognuna legata ad un particolare aspetto. Riportiamo la definizione di McCarthy del 1956:

L'intelligenza artificiale è una disciplina che studia metodologie e tecniche atte a concepire, progettare e sperimentare sistemi hardware o software le cui prestazioni possono essere assimilate a quelle di un essere umano.

Si può quindi affermare che l'intelligenza artificiale è l'attività rivolta alla costruzione di sistemi intelligenti. Attualmente l'AI comprende un'immensa varietà di sottocampi tra cui il settore dell'apprendimento automatico o *Machine Learning* che si occupa della costruzione di programmi che migliorano automaticamente le loro prestazioni con l'esperienza. Questo settore cerca di adattare a nuove circostanze la macchina, raffinando la sua conoscenza e studiando i meccanismi mediante i quali il comportamento di un agente risulta migliorato sulla base di esperienze o casi precedentemente trattati.

Una definizione più precisa di cosa si dovrebbe intendere per programma in grado di apprendere dall'esperienza è la seguente:

Definizione*: Si dice che un programma è in grado di apprendere dall'esperienza E rispetto ad un certo compito T ed ad una certa misura di prestazioni P, se le sue prestazioni nell'eseguire il compito T, misurate rispetto a P, migliorano con l'esperienza E.*

Questa definizione è abbastanza ampia in modo da includere la maggior parte dei compiti che vengono considerati comunemente compiti di apprendimento, nel senso in cui usiamo la parola apprendimento nel linguaggio di tutti i giorni.

Il progetto di un sistema di apprendimento coinvolge varie fasi. Innanzitutto si tratta di determinare il tipo di esperienza che deve essere usata per l'apprendimento. Poi si deve stabilire quale sia la funzione target f che deve essere appresa: infatti è utile ridurre il problema del miglioramento delle prestazioni P in un certo compito, al problema dell'apprendimento di una certa funzione. Determinata la funzione f, bisogna giungere ad una descrizione operativa della funzione target, cioè ad una descrizione che può essere usata dal nostro programma in una quantità limitata di tempo. Spesso può essere molto difficile apprendere una definizione operativa in modo esatto: il passo successivo è quello di determinare una rappresentazione, detta modello o ipotesi, che approssimi la funzione da apprendere: ad esempio, una rete neurale, una rete bayesiana, un

modello di Markov nascosto o un albero di decisione. L'ultimo passo consiste nello scegliere un algoritmo di apprendimento per approssimare la funzione target.

Ci sono varie questioni che devono essere risolte nel progetto di un sistema di apprendimento. Quando l'algoritmo di apprendimento converge alla funzione desiderata? Quanti dati di addestramento sono necessari e come devono essere scelti? Come può essere usata la conoscenza a priori per guidare il processo di generalizzazione dagli esempi? Qual è il miglior modo per ridurre un compito di apprendimento al problema dell'approssimazione di una funzione? Come può il processo di apprendimento modificare automaticamente la sua rappresentazione per migliorare la sua capacità di rappresentare ed apprendere la funzione obiettivo? La scelta di un modello rappresenta un passo critico nel progetto di un sistema di apprendimento. Innanzitutto si deve tenere conto del numero dei parametri per cercare di evitare fenomeni di sovrapprendimento (overfitting) o sottoapprendimento dei dati. Il fenomeno dell'overfitting è associato con la memorizzazione dei dati insieme al loro rumore fino ad un punto che è dannoso per la generalizzazione. Un approccio per questo problema è quello di pesare la funzione da apprendere con un termine che tiene conto della complessità del modello. Nel caso si abbia un modello con troppi parametri, l'approccio corretto sarebbe quello di modificare il modello. Un'altra strategia è quella di suddividere l'insieme dei dati in due parti: una parte servirà per addestrare il modello e verrà chiamata training set, mentre l'altra servirà per valutarne le prestazioni e sarà detta test set. Ognuno dei due insiemi di dati darà origine ad un certo errore: l'errore di addestramento decrescerà monotonicamente al crescere del numero delle epoche di addestramento, mentre l'errore sul test set raggiungerà un minimo e poi comincerà a crescere. A questo punto si tratta di fermare l'addestramento quando l'errore sul test set comincia a crescere o quando l'errore sul training set ha raggiunto una certa soglia. Comunque questa tecnica può lasciare un parziale overfitting dei dati. Inoltre, per ottenere un corretto addestramento, tutte le caratteristiche descrittive dei dati dovrebbero essere ugualmente rappresentate nel training set.

Un altro problema classico dell'apprendimento automatico riguarda la scelta dei dati. Ad esempio, se i dati usati per addestrare il modello sono molto correlati con quelli usati per il test set, si avrà una sovrastima della capacità predittiva. Per risolvere questo inconveniente, è necessario selezionare un insieme di dati in modo da rendere ugualmente rappresentate tutte le caratteristiche che li contraddistinguono. Una strategia alternativa consiste nel pesare i dati in accordo alla loro novità ma un rischio di questo approccio è quello di pesare molto i dati errati. Comunque i metodi di apprendimento automatico sono capaci di estrarre le caratteristiche essenziali dai vari esempi e di scartare l'informazione non desiderata quando presente; inoltre sono in grado di trovare correlazioni complesse e non linearità presenti negli esempi.

Per quanto riguarda gli algoritmi di apprendimento che fanno riferimento a modelli connessionisti, si possono suddividere in due classi, a seconda se l'aggiornamento avviene dopo ogni esempio oppure dopo l'intero insieme di esempi. L'addestramento è detto online se l'aggiustamento dei parametri del modello avviene dopo la presentazione di ogni esempio, mentre è detto batch se i parametri sono aggiornati dopo la presentazione di un grande numero di esempi, se non di tutti. L'apprendimento online non richiede di tenere in memoria molti esempi ed è più flessibile e più facile da implementare; può però introdurre un certo grado di casualità legato al fatto che l'aggiornamento avviene sulla base di un solo esempio. Può essere dimostrato che l'apprendimento online fatto con un tasso di apprendimento sufficientemente piccolo, approssima l'apprendimento batch.

Un'ultima osservazione riguarda i modelli che si ottengono dopo l'addestramento. Quando un modello complesso viene addestrato secondo un certo criterio di ottimizzazione, si ottengono dei parametri differenti se si variano certi fattori durante la procedura di apprendimento, come ad esempio, l'algoritmo di addestramento, l'ordine di presentazione degli esempi, il training set. Inoltre possono essere impiegate classi di modelli differenti. E’ naturale pensare che la migliore classificazione o predizione possa essere raggiunta mediando i vari modelli ottenuti in modi diversi.

Riassumendo, l'approccio dell'apprendimento automatico è quindi adatto in tutti quei domini caratterizzati dalla presenza di grandi quantità di dati anche rumorosi e dall'assenza di teorie generali. L'idea fondamentale di questo approccio è di imparare la teoria automaticamente dai dati, attraverso processi di inferenza, di adattamento di modelli e di apprendimento da esempi e rappresenta una metodologia alternativa ai metodi tradizionali. I metodi dell'apprendimento automatico sfruttano pesantemente la potenza dei calcolatori e traggono grandi benefici dal progresso in velocità delle macchine: la confluenza di tre fattori - dati, calcolatori e teoria - servirà allo sviluppo dell'apprendimento automatico. E' stato inoltre provato che gli algoritmi di machine learning sono di grande utilità pratica in molte applicazioni. In particolare, sono molto utili nei problemi di data mining dove si hanno a disposizione delle grandi basi di dati che contengono delle regolarità implicite che possono essere scoperte automaticamente, in quei domini poveri di teoria in cui una persona non potrebbe avere una conoscenza necessaria a sviluppare un algoritmo efficace ed in quegli ambienti in cui il programma si deve adattare dinamicamente al cambiamento delle condizioni.

Una critica spesso sollevata ai metodi di apprendimento automatico è che sono degli approcci a scatola nera: non si può sempre capire come mai un dato modello dia una certa risposta.

4.2 Ragionamento con Incertezza e Modelli Probabilistici

Nei sistemi basati sulla logica formale, si assume che le informazioni disponibili e le conclusioni derivate dalle regole di inferenza, siano vere o false. In molte situazioni però, non si può avere quasi mai accesso all'intera verità del dominio che si sta esaminando e vi saranno domande a cui non si potrà dare una risposta certa. Il sistema deve quindi agire in presenza di incertezza. Alcune fonti di incertezza sono i dati inaffidabili, mancanti o imprecisi. L'incertezza può anche sorgere dall'incompletezza e dalla mancanza di correttezza nella comprensione delle proprietà dell'ambiente. La teoria della probabilità fornisce le basi per il trattamento dei sistemi che ragionano con incertezza. La conoscenza che il nostro

sistema riesce a derivare, può fornire al massimo un grado di credenza sulla soluzione di un certo problema. La probabilità fornisce un modo per riassumere l'incertezza che deriva dalla non completa conoscenza del dominio.

Sono stati formalizzati e discussi vari approcci per il trattamento dell'incertezza: l'impiego di logiche non monotone (si rimettono in discussione delle informazioni giµa esistenti), l'inclusione di fattori di certezza nelle regole di produzione, la teoria di Dempster-Shafer e le reti bayesiane.

Nel nostro caso, verrà illustrato l'approccio bayesiano alla probabilità e alla statistica, mettendo in risalto le differenze con la visione classica e saranno descritte le reti bayesiane. Poichè i dati disponibili sono affetti da rumore e siamo quindi costretti a ragionare in un ambiente incerto, l'approccio bayesiano ci fornisce una teoria robusta che unifica differenti tecniche. Le principali caratteristiche dell'approccio bayesiano possono essere riassunte nei seguenti punti:

- utilizza delle ipotesi o modelli che si avvalgono dell'informazione a priori e dei dati disponibili;
- ricorre al linguaggio della teoria della probabilità per assegnare le probabilità a priori ai modelli;
- si avvale del calcolo della probabilità per valutare le probabilità a posteriori delle ipotesi alla luce dei dati disponibili e per fornire una risposta univoca a certi quesiti.

Può essere provato in senso matematico stretto che questo è un modo consistente di ragionare in presenza di incertezza e da un punto di vista teorico l'ambiente probabilistico bayesiano unifica molti metodi di apprendimento automatico.

Anche il modello di apprendimento deve tener conto della natura incerta dei dati e deve essere quindi probabilistico. Spesso in un insieme di dati, se qualcosa è difficile da apprendere, è molto probabile che sia un caso atipico o un errore. Una delle ragioni del successo dei metodi di apprendimento automatico su domini di dati incerti è la loro capacità di gestire il rumore presente nei dati. Una volta che l'ambiente probabilistico bayesiano è stato definito, l'idea successiva è quella di

utilizzare i modelli grafici. Poichè nell'analisi bayesiana il punto di partenza è per lo più sempre una distribuzione di probabilità di grado elevato, tale espressione deve essere decomposta e semplificata. La più comune semplificazione è quella di assumere che alcuni insiemi di variabili siano indipendenti, data la dipendenza condizionale con altri insiemi di variabili. Queste relazioni di indipendenza possono essere spesso rappresentate da un grafo dove le variabili sono associate con i nodi e un collegamento mancante rappresenta una particolare relazione di indipendenza. Le relazioni di indipendenza permettono la fattorizzazione della distribuzione di probabilità di grado elevato in un prodotto di semplici distribuzioni locali associate a piccoli gruppi di variabili correlate fra loro. I modelli grafici possono essere suddivisi in due grandi categorie a seconda se gli archi associati sono orientati o meno. I modelli grafici non orientati sono impiegati in quelle situazioni in cui le interazioni sono considerate completamente simmetriche, mentre i modelli grafici orientati sono utili in quelle circostanze in cui le interazioni non sono simmetriche e riflettono relazioni causali o irreversibilità temporale. Il linguaggio di rappresentazione dei modelli grafici è utile nella maggior parte delle applicazioni di apprendimento automatico; le reti bayesiane, le reti neurali, i modelli di Markov nascosti rappresentano un esempio di tali metodologie. In seguito verranno descritte la reti bayesiane, un modello grafico probabilistico per rappresentare le relazioni tra un insieme molto vasto di variabili aleatorie che descrivono il dominio di interesse. Questo formalismo codifica in modo molto efficiente la distribuzione congiunta di probabilità, sfruttando le relazioni di indipendenza condizionale tra le variabili. La combinazione delle reti bayesiane e della statistica bayesiana è la base per molte tecniche data mining: gestisce facilmente gli insiemi di dati incompleti e facilita la combinazione della conoscenza a priori e dei dati.

L'approccio bayesiano alla probabilità può essere esteso al problema dell'apprendimento: da questa unione scaturisce una teoria estremamente potente che fornisce una soluzione generale ai problemi di rumore, sovraddestramento e previsione ottima. L'apprendimento bayesiano si pone come obiettivo il problema di fare delle previsioni e ritiene il problema della formulazione di ipotesi a partire dai dati, come un suo sottoproblema. Un modo per specificare che cosa

intendiamo per la migliore ipotesi è quello di affermare che la migliore ipotesi è quella più probabile, avendo a disposizione dei dati ed una certa conoscenza iniziale delle probabilità a priori delle varie ipotesi. Le ipotesi elaborate dai dati e combinate in modo opportuno, portano alla formulazione di una previsione. Il metodo bayesiano non sceglie tra un insieme di ipotesi, ma le combina in base alloro capacità di rappresentare i dati.

4.3 Formulazione Generale del Problema dell'Apprendimento

L'apprendimento automatico si fonda sull'idea che l'esperienza possa migliorare la capacità dell'agente di agire in futuro. Ci sono vari paradigmi di apprendimento che si adattano alle varie condizioni che devono essere modellate. Qualsiasi situazione in cui sia l'ingresso che l'uscita di una componente possono essere percepiti, è detta di apprendimento supervisionato. La forma di apprendimento in cui non è disponibile alcuna informazione su quale sia l'output corretto è chiamata apprendimento non supervisionato. Con l'apprendimento non supervisionato si possono sempre predire delle relazioni tra le percezioni, ovvero si può imparare a predire le percezioni future a partire da quelle precedenti. Non si può invece apprendere cosa sia meglio fare in un certa occasione. Infine in certe circostanze può accadere che l'uscita corretta sia nota solo per certi input ma non per tutti: si parla allora di apprendimento parzialmente supervisionato. In generale si può affermare che il problema dell'apprendimento può essere visto come apprendimento di una funzione. Nell'apprendimento supervisionato all'elemento di apprendimento viene fornito il valore corretto o approssimativamente corretto della funzione per particolari input ed esso modifica la sua rappresentazione della funzione cercando di far collimare le informazioni fornite dal feedback. Risulta utile dare una definizione formale del problema dell'apprendimento supervisionato. Il punto di partenza è rappresentato dalla formulazione astratta di un qualunque problema di apprendimento induttivo di concetti detto concept learning:

Sia $D = \{(x_1; c_1 = f(x_1)); \ldots; (x_N; c_N = f(x_N))\}$ un insieme di esempi di addestramento che rappresentano gli ingressi e le uscite di una generica funzione

f: X → C, con $\{x_1; \ldots; x_N\} \subset$ X e $\{c_1; \ldots; c_N\} \subset$ C. Il compito dell'inferenza induttiva pura (o induzione) è quello di determinare una funzione h : X → C che meglio approssimi f su D secondo un certo criterio di ottimalità.

La funzione ignota f è chiamata funzione o concetto target mentre la funzione h è chiamata ipotesi e rappresenta la migliore approssimazione del concetto target che il processo di inferenza induttivo μe in grado di determinare. L'insieme D viene detto training set ed i vari $x_i \in X$ sono un sottoinsieme delle istanze o inputs della funzione f da apprendere. Un esempio non è altro che una coppia $(x_i; c_i = f(x_i))$ del training set e l'apprendimento viene detto supervisionato, perchè ad ogni ingresso x_i della funzione f viene associato il corrispondente valore $f(x_i)$ che assume la funzione da apprendere. Nel migliore dei casi l'algoritmo di apprendimento riesce a determinare un'ipotesi h tale che h = f. L'ipotesi viene determinata ricorrendo ad un criterio di ottimalità, di solito espresso come minimizzazione dell'errore (che può essere rappresentato in più modi) dell'ipotesi sul training set. Nella quasi totalità dei casi è impossibile ricavare l'esatta rappresentazione della funzione target ed il processo di apprendimento si riduce a determinare una approssimazione $h \in H$ tramite una ricerca nello spazio H delle ipotesi: in effetti ogni algoritmo di apprendimento può essere visto come una strategia di ricerca nello spazio delle ipotesi. Un algoritmo è ben progettato se tiene conto della struttura dello spazio che esplora per organizzare la ricerca in modo ottimale ed efficiente. Ciascuna preferenza per un'ipotesi piuttosto che per un'altra, al di là della semplice consistenza con gli esempi, è detta inclinazione o bias. Poichè c'è quasi sempre un gran numero di possibili ipotesi h consistenti, tutti gli algoritmi di apprendimento presentano un qualche genere di inclinazione.

La scelta della rappresentazione per la funzione target è probabilmente il punto più importante con cui il progettista si deve confrontare. Oltre ad influenzare l'algoritmo di apprendimento, può avere conseguenze sulla risolubilità stessa del problema. Inoltre nell'apprendimento c'è una contrapposizione fondamentale tra l'espressività e la funzione desiderata è rappresentabile con il linguaggio stesso - e l'efficienza - il problema di apprendimento è trattabile se si sceglie un certo linguaggio di rappresentazione. Questa definizione generale del

problema dell'apprendimento consente di modellare un gran numero di situazioni del mondo reale.

4.4 Le reti bayesiane

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[12] ha condotto, infatti, ad una rapida affermazione delle reti Bayesiane per comprendere le relazioni di causalità che intercorrono fra le variabili e per attuare un'analisi dei dati ad esse relative.

Le Belief Network sono *una diretta rappresentazione di un dominio*: i legami rappresentati nel modello grafico esprimono le reali dipendenze fra le variabili.

In ogni caso le BN consentono la comprensione di un problema complesso proprio grazie all'esplicazione dei legami fra le variabili.

Per ottenere le relazioni tra i dati bisogna apprendere la struttura della rete *(structure learning)* e apprendere i parametri *(parameter learning)*. Come già detto nel capitolo precedente, la struttura della rete può essere nota, perché ad esempio fornita da un esperto, o meno e il database può essere completo o può presentare dei dati mancanti o nascosti.

---

[12] 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 qual è la causa che ha provocato l'effetto B?).

Nei paragrafi successivi ci soffermeremo, dopo aver fatto un'analisi sul processo di inferenza nelle reti, soprattutto sullo *structure learning* analizzando i vari algoritmi presenti in letteratura. Nella parte finale del capitolo invece verrà affrontato il *parameter learning* e le metodologie per valutare quanto la rete appresa rispecchi la realtà che si vuole rappresentare. Ovviamente quanto più la struttura della rete si avvicina al modello effettivo tanto più i parametri sono ben determinati e la "divergenza informazionale" o "cross-entropia", che mi permette di classificare i dati contenuti in un archivio, tende a zero. Gli algoritmi sono presentati al fine non solo di rendere chiaro ed evidente al lettore quelle che sono le differenze specifiche tra gli algoritmi stessi, ma soprattutto per evidenziare quelli che sono gli usi specifici e le informazioni a priori necessarie per determinare la struttura della rete: questi sono appunto stati gli elementi che hanno portato ad una determinata scelta piuttosto che ad un'altra.

4.4.1 Definizione di rete bayesiana

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

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 $X = \{X_1, X_2, \ldots\ldots X_N\}$;
- con degli archi orientati fra i nodi, l'insieme di asserzioni di indipendenza condizionata relative ad X;

P (*rappresentazione quantitativa*) - un insieme di distribuzioni di probabilità locali associate ad ogni variabile.

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

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

I rami che collegano i nodi rappresentano le dipendenze causali fra le variabili. La mancanza di un arco tra due nodi riflette la loro indipendenza condizionale. Al contrario, la presenza di un arco dal nodo X_i al nodo X_j può essere interpretata come il fatto che X_i sia causa diretta di X_j.

La semantica probabilistica espressa da queste strutture consente di quantificare queste dipendenze con la CPT[13] di una variabile, dati i suoi padri[14]. 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* 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*. Se un nodo non ha padre (nessun collegamento punta ad esso), il nodo conterrà una tabella di probabilità marginale.

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.

4.4.2 Teoria dei grafi (cenni)

Essendo la rete bayesiana un grafo vediamo brevemente quelle che sono le caratteristiche e le varie tipologie esistenti di tale struttura.

Un grafo *diretto* (o grafo orientato) G è una coppia (V, E), dove V è un insieme ed $E \subseteq V \times V$ è una relazione binaria su V. Gli elementi di V sono detti vertici (o variabili o nodi), e gli elementi di E sono detti archi. Dato un nodo u, i successori (o nodi figli) di u sono i vertici v tali che $(u, v) \in E$. I predecessori (o nodi padri) di u sono i vertici v tali che $(v, u) \in E$. Se (u, v) è un arco, allora, diremo che v è adiacente a u e che vi è una *relazione di dipendenza condizionale* tra u e v.

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

Il grado uscente di un nodo u è il numero di successori di u; il grado entrante di u è il numero di predecessori di u. Il grado di un nodo è la somma tra grado uscente e grado entrante del nodo.

Un *cammino o path o adjacency path* di lunghezza $k \geq 0$ da un vertice u ad un vertice $u`$ è una sequenza di nodi $(v_0, v_1, ..., v_k)$ tale che $u = v_0$, $u` = v_k$, e $(v_i, v_{i+1}) \in E$ per ogni $i = 0, \ldots, k - 1$. La lunghezza del cammino è il numero di archi del cammino. Si noti che esiste sempre un percorso di lunghezza 0 da u a u. Un percorso semplice è un cammino senza nodi ripetuti. Un nodo v è raggiungibile da un nodo u se esiste un cammino da u a v.

Un *ciclo* è un percorso $(v_0, v_1, ..., v_k)$ tale che $v_0 = v_k$ e il cammino contiene almeno un arco. Un ciclo è semplice se inoltre $v_0, v_1, ..., v_k$ sono distinti. Un cappio è un ciclo di lunghezza 1 ed è il più piccolo ciclo possibile su grafi orientati. Un grafo è aciclico se non contiene cicli.

Per un qualsiasi nodo in un percorso se due archi si incontrano sullo stesso allora il nodo stesso è un *collider* del percorso, ad esempio in un grafo del tipo X →V←Y diremo che V è un collider del percorso. Il concetto di collider è precipuo per un percorso ossia un nodo può risultare collider in un path, ma non in un altro.

Un grafo *indiretto* (o non orientato) G è una coppia (V, E) tale che V è un insieme di vertici, ed E è un insieme di archi non orientati. Un arco non orientato è un insieme (u, v) dove u e v sono vertici distinti dove l'ordine dei nodi non è rilevante. Viene usata comunque la notazione (u, v) per indicare l'arco (u, v) assumendo che (u, v) e (v, u) siano lo stesso arco. Si noti che un grafo indiretto non ammette cappi.

Un grafo *pesato* è un grafo $G = (V, E)$ con una funzione di peso $w : E \rightarrow \mathrm{R}$ che associa un peso ad ogni arco.

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.

Un *DAG* è un grafo diretto e aciclico. Una *foresta* non è altro che un DAG dove ogni nodo può avere o un solo predecessore o non ne ha nessuno. Un *albero*

è invece una foresta dove ogni nodo ha un nodo padre tranne uno che è detto nodo radice e non ha predecessori.

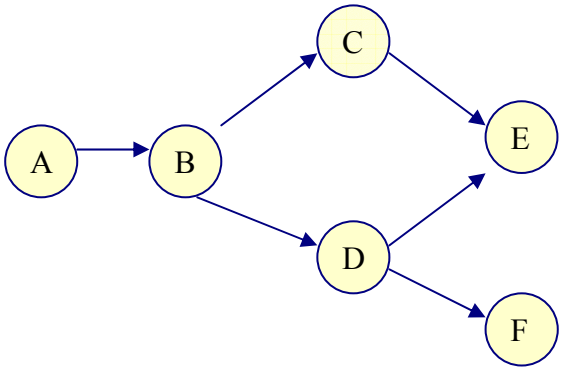


Figura 3.1: Un DAG.

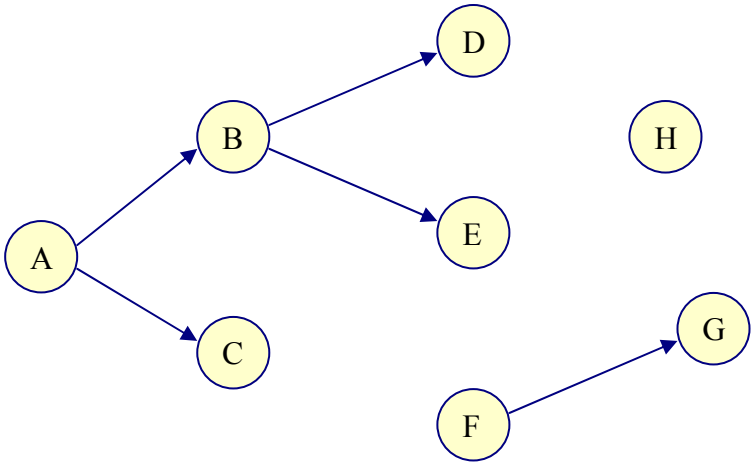


Figura 3.2: Una foresta

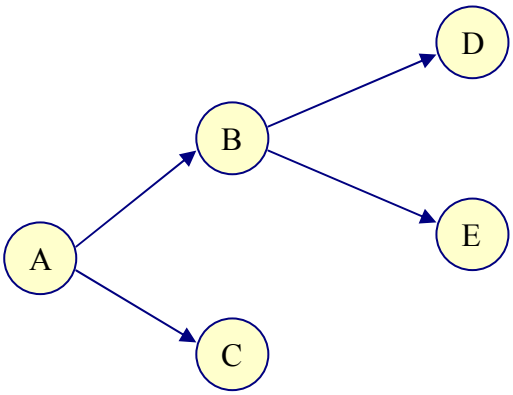


Figura 3.3: Un albero

4.4.3 Indipendenza nei grafi

Data una struttura di rete S, gli asserti di indipendenza condizionata possono essere letti da un grafo usando il concetto di *direction dependent separation o d-separation*[15]. 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 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 $C \subseteq N \setminus \{X, Y\}$, 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

1. Ogni collider in $P \subseteq C$ *non* ha discendenti in *C*;
2. 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.

In altre parole *la d-separation è la regola per individuare le indipendenze in una Bayesian Network.*

Una semplice analogia permetterà 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.

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. 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.

[15] d sta per directional

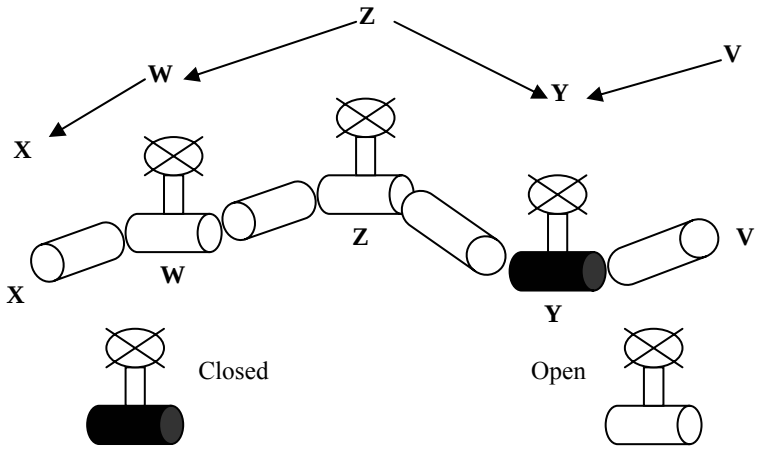


Figura 3.4:Analogia tra il flusso casuale e quello di un fluido

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 → C → B: A è una causa *indiretta* di B;
2. A ← C ← B: B è una causa *indiretta* di A;
3. A ← C → B: C è una causa comune di A e B;
4. A → C ← 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*.

Esaminiamo, ora, il caso in cui Z sia un insieme non vuoto. Inserire un vertice nel condition 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 cause sono dipendenti (mutuamente esclusive).

Riportando il discorso alla definizione di indipendenza per le 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.

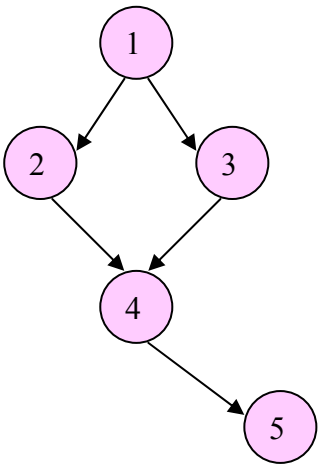


Figura 3.5 Rete bayesiana

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 ← 1 → 3 - il nodo Z non è un collider;
- 2 → 4 ← 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 → 4 ← 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

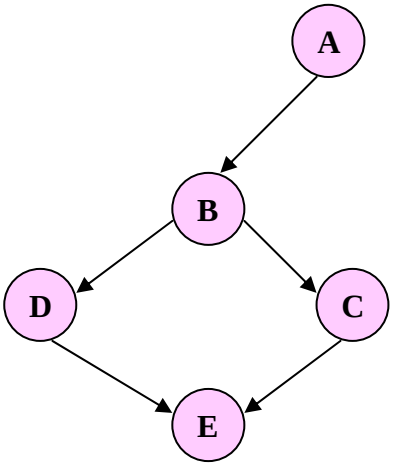


Figura 3.6:Rete Bayesiana

collega C e D dove E è un collider. Dato l'insieme vuoto {} C e D sono d-separated. Ritornando all'analogia delle valvole, inserendo il collider E nel cutset 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.

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 $I(X,Y|C') \leq I(X,Y|C)$.

4.4.4 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 di un evento[16] 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 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 $p(x \mid \xi)$.

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 Lazio 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

[16] Alcune nozioni (evento, ad esempio) e assiomi della teoria della probabilità sono riportati, in breve, in Appendice.

Lazio ha giocatori migliori rispetto alle altre squadre e alle sue vittorie 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, 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.

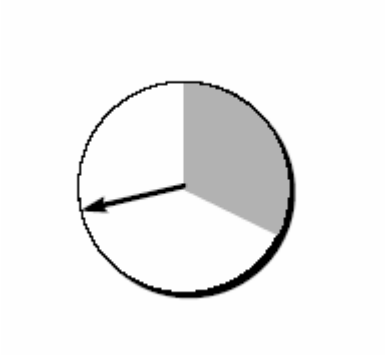


Figura 3.7: Probability assessment

Si consideri una ruota con solo due regioni (ombra e non ombra), come quella illustrata in Figura 3.7. 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.

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. [HEC95]

4.4.5 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.

Per comprendere meglio quanto detto vediamo un esempio 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.

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

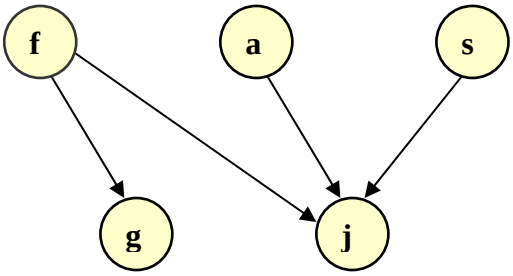


Figura 3.8:Rete Bayesiana che modella un problema di frode con carta di credito

Dalla rete 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 3.1 dalla quale 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".Quindi 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 3.1, 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.

PROBABILITY			CONDITIONS		
			FRAUD	AGE	SEX
Fraud=Yes	Fraud=No				
0.00001	0.9999				
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

Tabella 3.1:Probabilità associate alla rete bayesiana in figura 3.5

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

Tabella 3.2: I valori sono osservati per ogni variabile, relative alla rete in figura 3.8, eccetto Fraud. Questi possono essere usati per aggiornare le probabilità. .

In altre parole dobbiamo valutare la probabilità $P(f \mid j,g,s,a)$. 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

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

Poiché alcune variabili sono tra loro indipendenti (se X_1 non è figlio di X_n allora $P(X_1 \setminus X_2,\ldots,X_n,\ldots X_k) = P(X_1 \setminus X_2,\ldots,X_{n-1},X_{n+1},\ldots X_k)$), effettuando le opportune semplificazioni risulta

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

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

$$P(f = Yes \mid j = Yes, g = No, s = Male, a =< 30) =$$

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

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.

4.4.6 Algorimi per l'inferenza nelle reti Bayesiane

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

Howard e Matheson prima (1981), Olmsted, Shachter in seguito, hanno prodotto 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. 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.*

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 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].

4.5 Learning Bayesian Network

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; 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.

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.

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.

4.6 Lo structural learning

L'intento dello Structural Learning è esplicitare, dalle osservazioni su un insieme di variabili (dominio), "cosa è connesso 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 (dependance 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[17], per scoprire l'esistenza di relazioni di dipendenza fra le variabili del dominio mentre l'approccio bayesiano mi permette di determinare il modello *m* che codifica la struttura cercata massimizzando la probabilità $p(M=m\backslash D)$, dove *M* è l'insieme dei modelli che si ritiene contenere il *true model* di un dominio e *D* è il database di osservazioni campionarie.

Analizzeremo ora in maniera più dettagliata i due approcci allo structural learning per poi presentare i vari algoritmi che seguono l'uno o l'altro metodo.

4.7 Approccio bayesiano

L'approccio bayesiano codifica l'incertezza sulla struttura di un dominio $\boldsymbol{X}=\{X_1,\ldots,X_n\}$ introducendo una variabile aleatoria, $\boldsymbol{M}$, i cui stati sono proprio le possibili strutture (*structure hypothesis*) associate ad $\boldsymbol{X}$. Dopodiché si sceglie il modello $\boldsymbol{m}$ che massimizza la probabilità a posteriori $p(\boldsymbol{m}|D)$, 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 $\boldsymbol{m}$ 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)
- **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 struttura si adatti ai dati. Un'importante caratteristica di queste metriche è la *proprietà di scomposizione*, ovvero:

[17] 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.

$$\text{Score}(G,D)=\sum_i Score(X_i \mid Pa(X_i), N_{x_i, Pa(X_i)})$$

dove N indica le statistiche di *Xi* e **Pa**(*Xi*) (il numero di istanze in D delle possibili coppie *(Xi,**Pa**(Xi))*. La decomposizione ottenuta permette di computare facilmente l'espressione suddetta, 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.

L'apprendimento non è altro che 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 ovviamente le varie modifiche vanno effettuate verificando sempre che l'operazione effettuata non porti alla creazione di un ciclo[18]. Quindi è la "ricerca" il cuore di tale approccio.

Per poter applicare la metodologia illustrata deve essere calcolata la probabilità *p(m\D)*. È naturale utilizzare una funzione *SCORE(m)* derivata da *p(m\D)* più precisamente dal suo logaritmo che, per il teorema di Bayes, può essere messo in relazione con le distribuzioni a priori del modello e dei dati:

$$SCORE(m) = \log p(m\backslash D) = \log p(m) + \log p(D\backslash m) - \log p(D) \cong \log p(D\backslash m)$$

L'approssimazione compiuta deriva dal fatto che *log p(D)* è una costante e anche *log p(m)* può essere supposto costante se si fa l'ipotesi che ogni modello sia equiprobabile (completa ignoranza a priori sul modello). Quindi per poter massimizzare *SCORE(m)* bisogna calcolare le distribuzioni *p(D\m)*.

Il loro calcolo è compiuto in forma chiusa applicando le seguenti cinque ipotesi sulla rete bayesiana e sul database *D*.

1) Ogni variabile X_i è *discreta* (può assumere gli stati x_i^k, con k=1… r_i) e la sua distribuzione locale di probabilità è una collezione di *distribuzione multinomiali* $p(x_i^k \mid pa_i^j, \theta_i, m) = \theta_{ijk} > 0$, una per ogni stato pa_i^j delle variabili padri (j=1,…, q_i con $q_i = \prod_{X_s \in Pa_i} r_s$), tali che $\sum_{k=1}^{r_i} \theta_{ijk} = 1 \ \ \forall i, j$. Definiamo inoltre due vettori di parametri per semplificare la notazione: $\theta_{ij} = \{\theta_{ijk}\}_{k=1}^{r_i}$ e $\theta_i = \{\theta_{ij}\}_{j=1}^{q_i}$.
2) I parametri θ_{ij} sono mutuamente dipendenti. Ciò comporta che il problema diventa separabile nel senso espresso dall'equazione: $p(\theta_s \mid m) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\theta_{ij} \mid m)$
3) Ogni insieme di parametri θ_{ij} ha come distribuzione la coniugata della distribuzione della variabile X_i corrispondente. In questo caso è *la distribuzione di Dirichlet,* $p(\theta_{ij} \mid m) \propto \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1}$ dove gli α_{ijk} sono gli iperparametri della distribuzione, tali che $\alpha_{ijk} > 0 \ \ \forall i, j, k$ e che $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.
4) *D è completo*, quindi non ci sono osservazioni mancanti.
5) Il campione *D* deve essere estratto da un fenomeno il cui modello è una struttura *S* di una rete di Bayes.

Sotto queste ipotesi riesco ad ottenere i seguenti risultati

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

[18] In [KAN62] è proposto un algoritmo che mi permette di stabilire l'ordine topologico all'interno della rete:se questa operazione è possibile il grafo è aciclico.

- $p(x_i^k \mid pa_i^j, \theta_i, m) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$ dove N_{ijk} è il numero delle volte che nel database D si ha che $X_i = x_i^k$ e $Pa_i = pa_i^j$
- $SCORE(m) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$ (a)

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

4.8 Test di indipendenza condizionata

Se nel bayesian approach è la ricerca il cuore dell'algoritmo, nei metodi non bayesiani o contrait-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,…,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*.

4.8.1 Il test di 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):

$$SL = p(S \text{ nella zona di rifiuto} \mid H_0 \text{ vera})$$

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).

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 $p(A,B \mid C) = p(A \mid C)\ p(B \mid C)$ e il test verrebbe definito di ordine 1. Diventa così meno immediata la stessa definizione della tabella di contingenza[19] rispetto al test di ordine 0. Un modo pratico 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_1C_i	N_{A1B1Ci}	N_{A1B1Ci}
B_2C_i	N_{A1B2Ci}	N_{A2B2Ci}

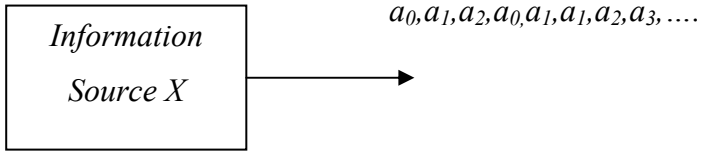
Tabella 3.3: Tabella di contingenza che esprime le occorrenze delle coppie (A_i,B_j) nel campione estratto dalla popolazione.

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).

4.8.2 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]



$$X=\{ a_0,a_1,a_2, \quad ,a_n\}$$

$$H(X)= -\sum_{i=0}^{n} p(X=a_i)\log(p(X=a_i))$$

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

---

[19] Il significato della tabella di contingenza verrà approfondito in appendice

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

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

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, definita come cross entropy nella letteratura delle reti bayesiane, fra i nodi A e B (i cui stati indichiamo simbolicamente con *a* e *b*) è definita come segue:

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

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

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 $P_D(x)$ 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 $P_D(x)$. 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 l'informazione su X portata da Y sia minore rispetto a quella portata da Z. D'altra parte si riesce a stimare P(X|Y) in modo più robusto rispetto a P(X|Z), perché implica l'elicitazione di meno parametri. Una miglioria si avrebbe inserendo l'entropia in un'opportuna funzione di score.

4.9 Differenze tra il bayesian approach e il constraint-based approach

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.

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. 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 ed analizzano un numero superiore di modelli rispetto ai test risultando, quindi, meno efficienti rispetto alla dimensione temporale.

In realtà le due metodologie, a volte, sono opportunamente integrate: ad esempio, l'approccio non bayesiano è usato per fornire o una rete o un possibile ordinamento delle variabili come input per un algoritmo bayesiano.

4.10 Algoritmi per lo structural learning

I vari algoritmi presenti in letteratura seguono o l'approccio bayesiano, dove la ricerca è il cuore dell'algoritmo, o l'approccio statistico, dove il risultato di un opportuno test statistico scopre le relazioni di indipendenza o dipendenza condizionata che sono implicite nelle osservazioni campionarie. Nei successivi paragrafi verranno analizzati:

- L'Algoritmo Bayesiano
- L'algoritmo K2
- L'algoritmo MDL

che seguono l'approccio bayesiano in particolar modo scenderemo maggiormente nel dettaglio per quanto riguarda ***l'algoritmo bayesiano.***

- L'algoritmo PC
- L'algoritmo TDMA

che seguono l'approccio statistico.

4.11 Algoritmo bayesiano

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

Dal teorema di Bayes

$$p(m_i)p(D \mid m_i) = p(m_j)p(D \mid m_j)$$

$$\frac{p(m_i \mid D)}{p(m_j \mid D)} = \frac{p(m_i)p(D \mid m_i)}{p(m_j)p(D \mid m_j)}$$

e il rapporto tra le evidenze $\frac{p(D \mid m_i)}{p(D \mid m_j)}$ è chiamato *fattore di Bayes.*

E' intuitivo scegliere come funzione di score *p(D|**m**);* nei paragrafi precedenti, si è

visto che sotto opportune ipotesi:

1. Il campione D è completo
2. I casi nel database sono indipendenti
3. la distribuzione a priori dei parametri è multinomiale
4. La coniugata a priori è una distribuzione di Dirichlet

si ottiene

- $$p(D \mid m) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \qquad (3.1)$$
- $$\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$$
- $$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

dove

- ***m*** rappresenta la struttura di rete candidata a rappresentare il dominio $\boldsymbol{X}=\{X_1,\ldots,X_n\}$ di ***n*** variabili/nodi, dato il database di campioni ***D***.
- Una variabile $\boldsymbol{X_i}$ presenta $\boldsymbol{r_i}$ stati mentre $\boldsymbol{q_i}$ sono le configurazioni in ***D*** dell'insieme dei padri di $\boldsymbol{X_i}$
- ***Γ(x)*** è la funzione gamma di Eulero
- $\boldsymbol{\alpha_{ijk}}$ 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.
- ***Nijk*** sono le occorrenze in ***D*** dei record aventi $\boldsymbol{X_i=x_i^k}$ (stato k-esimo di X_i) e $\boldsymbol{Pa(X_i)=pa_i^j}$ (configurazione j-esima dei padri di X_i).

Per la distribuzione dei parametri si ottiene:

$$p(x_i^k \mid pa_i^j, \theta_i, m) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \qquad (3.2)$$

La funzione *SCORE(**m**)* usata è il logaritmo[20] di *p(m\D)*

[20] 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.

$$SCORE(m) = \log p(m\backslash D) = \log p(m) + \log p(D\backslash m) - \log p(D) \cong \log p(D\backslash m)$$

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 precedenti è immediato ricavare che

$$SCORE(m) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\log\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} + \sum_{k=1}^{r_i}\log\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \qquad (3.3)$$

Per poter avere una formula computabile bisogna assegnare dei valori agli $\boldsymbol{\alpha_{ijk}}$. Questi iperparametri codificano la conoscenza a priori, "user confidence", che l'utente ha sul modello $\boldsymbol{m}$. Una possibilità è esprimere l'equiprobabilità di ogni istanza dello spazio delle probabilità con la relazione $\alpha_{ijk} = \frac{\alpha}{q_i r_i}$

$\alpha_i = \sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{\alpha}{q_i r_i} = \alpha$ per cui resta da assegnare un unico iperparametro α - *dimensione di un campione equivalente*.

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.

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(m\D)* sia localizzato nell'intorno di un particolare modello *m'*. Per selezionare *m'* si introduce una funzione il cui valore sia tanto più alto quanto il generico modello *m* è prossimo a *m'* . A tale scopo, un'eventuale metodologia di ricerca è l'"Hill-Climbing".

4.11.1 Procedura di ricerca "Hill Climbing"

La procedura di ricerca "*hill-climbing*" mi permette di massimizzare la funzione *SCORE(m).*

Scelta una struttura *S* è possibile valutare il guadagno di *SCORE* che si ha per ogni possibile variazione elementare degli archi, in modo da mantenere l'aciclicità del grafo. Queste variazioni sono l'aggiunta di un arco fra due nodi mutuamente indipendenti, la cancellazione di un arco fra due nodi dipendenti, il 4

Sfruttando il fatto che la funzione di costo descritta dall'equazione (3.3) può essere scomposta nella somma di *n* addendi, ciascuno associato ad un nodo X_i ed ai suoi padri Pa_i, la variazione di un solo arco della struttura *S* influirà al più su due addendi, relativi ai nodi sorgente e pozzo dell'arco variato. In particolare ciò accade soltanto se un arco della struttura viene invertito. Negli altri casi è sufficiente calcolare la variazione dell'addendo relativo al nodo pozzo del nuovo arco.

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

Questo tipo di approccio necessita di un grafo di partenza. Candidati per questo possono essere il grafo privo di archi, che codifica la completa ignoranza sulle relazioni che intercorrono fra le variabili, un grafo aciclico costruito

inserendo archi in modo casuale oppure una rete che rappresenti la conoscenza a priori posseduta sul dominio del problema.

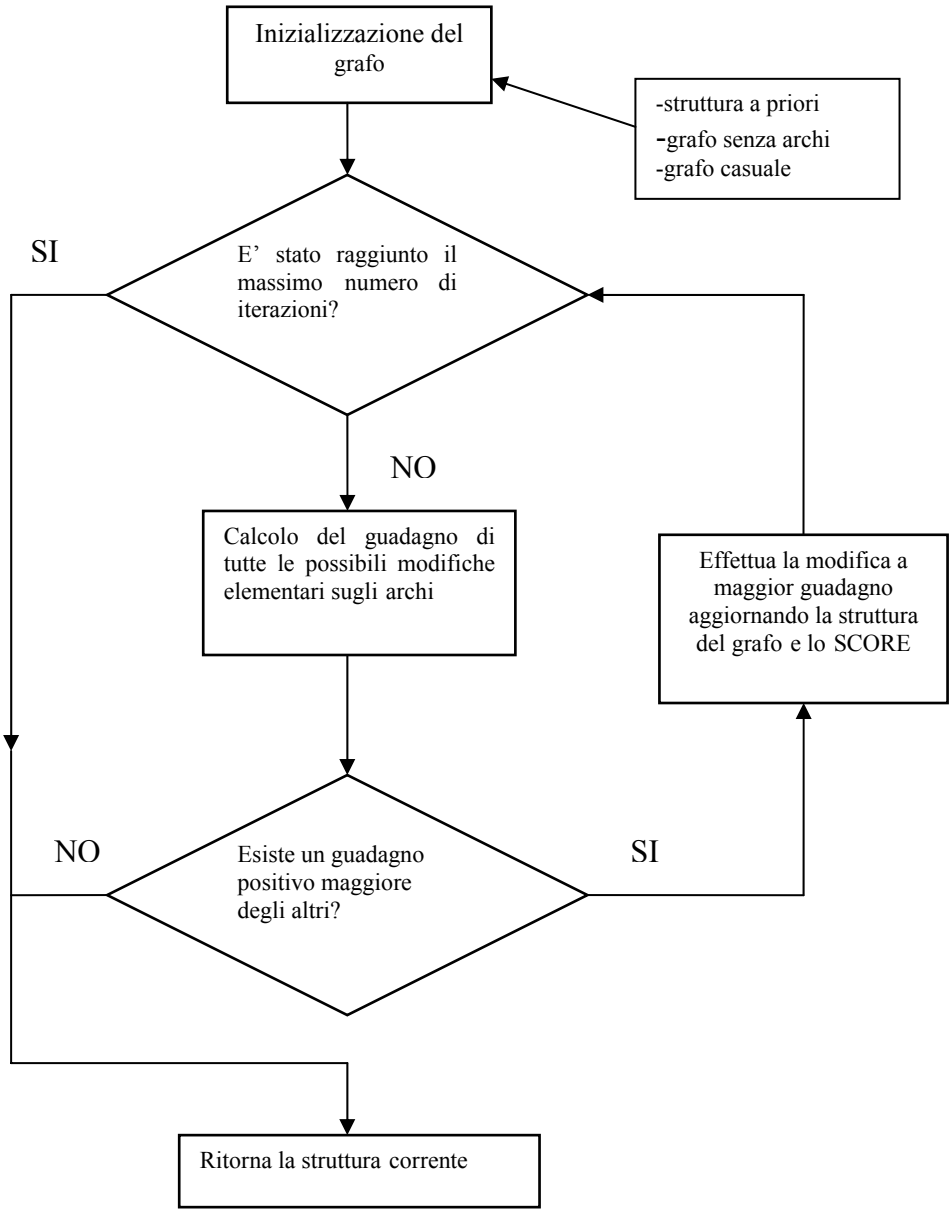


Figura 3.6: Procedura di ricerca "Hill Climbing"

4.12 L'algoritmo K2

Cooper e Herskovitz, in "A Bayesian Method for the Induction of Probabilistic Networks from Data" [COO92], hanno concepito un metodo bayesiano 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 e 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 $(v_{i_1}, v_{i_2}, ..., v_{i_{ri}})$. 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 la score function è

$$p(B_s, D) = p(B_s) \prod_{i=1}^{n} g(i, \pi_i) \qquad (3.4)$$

dove $g(i, \pi_i)$ è pari a

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

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

$$p(X_i = v_i^k \mid \pi_i = w_{ij}) = \frac{N_{ijk} + 1}{N_{ij} + r_i} \qquad (3.5)$$

Per la metrica K2 si assume $\alpha_{ijk} = 1$, ovvero completa ignoranza a priori sulla distribuzione di probabilità.

La procedura K2 si differenzia anche per la fase di inizializzazione. Mentre nell'algoritmo 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. 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 $p(X \mid D)$, 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 $X = \{X_1,\ldots,X_n\}$ in quanto

la distribuzione di probabilità congiunta $p(X_1,X_2, \quad X_n)$ 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. Se non è disponibile nessuna informazione a priori, la distribuzione di probabilità di $p(B_s)$ è *uniforme*, questa ulteriore ipotesi semplificativa, consente di ignorare (si elide) il valore $p(B_s)$ nella scoring function quando si confrontano due modelli.

In sintesi l'algoritmo K2 è così riassumibile:

1. Scegliere un ordinamento topologico sui nodi
2. Per ogni nodo X_i per i=1,….,n dove n è il numero dei 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 $\boldsymbol{X}=\{X_1,X_2,.....X_n\}$ e $\boldsymbol{Pa}=\{Pa(X_1),.....Pa(X_n)\}$.

L'algoritmo effettua una ricerca *greedy incrementale* che procede assumendo, all'inizio, che un nodo non abbia padri. Dopodiché, dato l'ordinamento $X_1<X_2<\ldots X_i\ldots<X_n$, l'insieme dei padri di ogni nodo X_i è determinato valutando se ogni variabile dell'insieme dei relativi predecessori, $pred(X_i) = \{X_1,\ldots,X_{i-1}\}$, 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 $\{X_1,\ldots,X_{i-1}\}$; 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.

Un altro svantaggio dell'algoritmo K2 è la necessità di designare un corretto ordinamento dei nodi il che, in assenza di informazioni a priori, non è semplice: l'ordine scelto influenza sia il risultato che la qualità della rete finale.

4.13 Agoritmo Minimum Description Lenght (MDL)

Il principio Minimum Description Length discende dal principio logico di William di Occam[21] noto anche come "Occam's razor", che postula " Pluralitas non est ponenda sine necessitate". L'immediata applicazione nel campo scientifico è quella di prediligere fra due teorie, o metodi, che conducano agli stessi risultati quello più semplice.

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:

- $L(B,D)=log(p(B)+N*H(B,D)-\frac{1}{2}klog(N)$ (3.6)
- $H(B,D)=\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}-\frac{N_{ijk}}{N}\log\frac{N_{ijk}}{Nij}$ (3.7)
- $k=\sum_{i=1}^{n}q_i(r_i-1)$

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 (3.6) rappresenta l'entropia condizionale della struttura di rete B che descrive quanto bene la rete rappresenti i

[21] Frate e filosofo francescano del XIV secolo

dati : “data description length”. L’entropia in (3.7) 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”, con cui quantificare la complessità della rete. Per essere più precisi, ogni probabilità stimata introduce un piccolo errore e il termine $\frac{1}{2} klog(N)$ rappresenta l’errore totale.

In tale algoritmo,dunque,si determina la rete minimizzando la metrica MDL e in questo modo 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 [BOU94] è dimostrato il seguente asserto che mostra come la metrica MDL è approssimativamente uguale alla metrica K2.

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*. Sia *p(B,D)* data dall’espressione (3.4), e sia *L(B,D)* la misura MDL di *B* dato *D*. Allora :

$$L(B,D) = \log p(B,D) + Costante \qquad (3.8)$$

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 K2; ne segue che l’MDL, avendo la funzione di score una soglia maggiore, tende ad apprendere una rete con meno archi rispetto all’algoritmo K2.

4.14 Algoritmo PC

L'algoritmo PC[22], dovuto a Spirtes, Glymour e Scheines, 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)
- 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[23] 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)\{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. In caso affermativo: si rimuove il legame XY; il condition set S viene memorizzato nella collezione di insiemi 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*.

---

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

[23] Conditional Indipendence

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 nelle pagine inizali del capitolo. 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, detta *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 $\{Y\}$.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:

1. 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.
2. 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.

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

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). 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 (≥ 0.664) 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]

4.15 Algoritmo TPDA

L'algoritmo "TPDA", acronimo per Three-Phase Dependence Analysis, nasce dal lavoro di J. Cheng, R. Greiner, J. Kelly, D. Bell e W. Liu ([CHE97]) e 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.

. 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.

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

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 $P_D(x)$, 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 $P_D(x) \approx$ 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.

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.

4.15.1 Algoritmo nel dettaglio

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.

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 va scelto un valore a seconda della dimensione del data set e della distribuzione dei dati.

Sebbene l'obiettivo di un algoritmo di Structural Learning sia apprendere un un Pmap, 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 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 impliciti nella distribuzione di probabilità codificata da una Bayesian network.

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.

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:

1. *Drafting* (schematizzare)
2. *Thickening* (infoltire)
3. *Thinning* (sfoltire)

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 .

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 1 richiede un solo test mentre nella fase 2 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).

4.16 Parameter learning

Vedremo ora come apprendere i parametri in una rete ovvero come determinare le tabelle di probabilità associate a ciascun nodo sia nel caso in cui il database è completo sia nel caso in cui ho missing value.

L'assunzione di database completo rende particolarmente chiara e semplice 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 $X_i = x^k_i$ e $Pa_i = pa_j$)

$$p(x_i^k \mid pa_i^j, \theta_i, m) = \frac{N_{ijk}}{\sum_k N_{ijk}}$$

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

$$p(x_i^k \mid pa_i^j, \theta_i, m) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$$

Uno degli algoritmi più diffusi per l'apprendimento dei parametri è l'algoritmo **EM** che viene utilizzato in presenza di missing values.

4.16.1 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; calcolare 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.

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 θ_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

$$E_{p(x \setminus D, \theta_s, S)}(N_{ijk}) = \sum_{i=1}^{N} p(x_i^k, pa_i^j \mid y_l, \theta_s, S) \qquad (3.9)$$

dove y_l è il possibile *l*-esimo caso incompleto in D e N il numero di record in D. Assumendo di massimizzare (*Maximization step*) secondo il criterio ML, si determina la configurazione θ_m che massimizza $p(D_c^{24}|\theta_m,m)$. Nel caso di variabili

multinomiali discrete segue

$$\theta_{ijk} = \frac{E_{p(x\backslash D,\theta_s,S)}(N_{ijk})}{\sum_{k=1}^{r_i} E_{p(x\backslash D,\theta_s,S)}(N_{ijk})} \qquad (3.10)$$

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

Dirichlet distribution)

$$\theta_{ijk} = \frac{\alpha_{ijk} + E_{p(x\backslash D,\theta_s,S)}(N_{ijk})}{\sum_{k=1}^{r_i} (\alpha_{ijk} + E_{p(x\backslash D,\theta_s,S)}(N_{ijk}))} \qquad (3.11)$$

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 θ di parametri, e dato un insieme D di osservazioni, risolve il problema del learning parameter apprendendo un nuovo vettore di parametri θ' per S da D.

Due fattori influenzano la scelta di θ' : il grado di adattamento a D e il non allontanarsi troppo dal modello preesistente $\hat{\theta}$. Per rispettare questi vincoli si introduce una funzione F da ottimizzare composta dal logaritmo dell'equazione (3.9) (log -likelihood) e dalla distanza fra $\hat{\theta}$ e θ'. 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 $\hat{\theta}-\theta'$. 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 θ' a partire da θ

[24] Database completo

(risultato dello step precedente) e da D fintantoché non si raggiunge un criterio di convergenza o un massimo numero di iterazioni. [KOL97][COZ01]

4.17 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:

1. 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:(le probabilità sono condizionate al modello di rete che rappresentano: m_p la rete gold e m_q la rete learned $H(p,q) = \sum_X p(X \setminus m_p) \log \frac{p(X \setminus m_p)}{q(X \setminus m_q)}$
2. per lo structural learning
   - *structural difference*: rappresenta il grado con il quale la learned netwok 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)* $\delta_i = |(Pa_i^Q \cup Pa_i^P) \setminus (Pa_i^Q \cap Pa_i^P)|$ 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]

Capitolo 5: Apprendimento di una rete bayesiana da un database di esempi

La fase di apprendimento, sia della struttura che dei parametri, di una rete Bayesiana è un processo di estrazione dell'informazione dai dati – Data Mining.

Gli algoritmi di learning 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 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'obiettivo di questo lavoro di tesi è valutare come e in che misura un opportuno algoritmo, partendo da conoscenza a priori nulla, sia in grado di apprendere le relazioni che intercorrono tra i dati. Per tale scopo è stato scelto l'algoritmo bayesiano che partendo da una rete priva di archi e da nessuna informazione a priori (non è necessario un ordinamento delle variabili come nel K2) è in grado di apprendere la struttura della rete e i parametri ad essa associata. Ovviamente anche questo metodo non mi permette di determinare perfettamente le relazioni tra le variabili in quanto il problema dell'apprendimento, sia

strutturale che dei parametri, rappresenta un problema di massimizzazione a più variabili.

5.1 Disponibilità dei dati

Uno degli obiettivi che si persegue quando ci si occupa di sicurezza stradale consiste nello sviluppare metodi per stimare il rischio per ciascuna categoria di utenti della strada in differenti condizioni di viaggio e periodi temporali. La capacità di misurare e confrontare i livelli di rischio, associati ai vari fattori correlati all'uomo, al veicolo, all'infrastruttura, al viaggio ed al tempo, fornisce le informazioni necessarie per individuare le aree che richiedono interventi per il miglioramento della sicurezza, ossia i fattori ad elevato rischio, e per valutare i risultati delle azioni di miglioramento della sicurezza.

Nel nostro Paese, i dati disponibili sul fenomeno incidentalità, sulle sue conseguenze e sull'esposizione al rischio necessari per studiare nel dettaglio l'incidentalità e implementare opportune strategie correttive, sono certamente insufficienti, per affidabilità, completezza e grado di dettaglio; basti pensare che per molte strade mancano dati affidabili sui flussi di traffico e ciò impedisce una valutazione attendibile del fenomeno.

Da quanto detto si evince che per lo studio dell'incidentalità è necessario acquisire un insieme di dati sufficientemente esteso e qualitativamente adeguato all'obiettivo. La scelta e il reperimento dei dati, siano essi di natura storica o dedotti attraverso attività di controllo e misure effettuate durante la fase di esercizio, rappresenta un tassello molto importante che influenza l'affidabilità delle fasi successive.

Come prima ipotesi si ritiene che il database debba contenere le seguenti informazioni:

- la dinamica dell'incidente (urto contro ostacolo fisso interno alla carreggiata, sbandata nell'ambito della carreggiata, urto con barriera di sicurezza, fuoriuscita, invasione della carreggiata opposta, urto con altro veicolo frontale – frontale, frontale – laterale, tamponamento, laterale – laterale, manovra di parcheggio, investimento di pedone);
- i veicoli coinvolti nell'incidente;

- le caratteristiche dei veicoli coinvolti;
- le modalità dell'incidente;
- le caratteristiche degli utenti coinvolti;
- il tipo di strada (autostrada extraurbana ed urbana, strada extraurbana principale, strada extraurbana secondaria, strada urbana di scorrimento, strada urbana di quartiere, strada extraurbana locale, strada urbana locale);
- le caratteristiche della strada (rettifilo, curva, clotoide, intersezione, pendenza longitudinale, viadotto, galleria, stato della pavimentazione, limite di velocità, numero di corsie, larghezza corsie e banchine, raggio della curva);
- le condizioni ambientali (pioggia, nebbia, illuminazione);
- le condizioni di traffico;
- la presenza di segnaletica;
- i danni ai veicoli coinvolti;
- danni alle persone coinvolte.

Questo lavoro di tesi è stato effettuato utilizzando i dati dell'ISTAT relativi agli incidenti verificatisi in Campania dal 1995 al 2000. Si è arricchita la banca dati con informazioni sul TGM (traffico giornaliero medio).

5.2 Analisi dell'apprendimento dal database

L'analisi dell'incidentalità affidata alle tecniche di Intelligenza Artificiale necessita di una corretta individuazione delle variabili che influenzano il fenomeno. La seguente tabella illustra in dettaglio le variabili e i possibili stati che queste possono assumere, nonché i codici numerici che il prototipo assegna loro.

X_i	**Significato**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
X_1	*Stagione*	Inverno	Primavera	Estate	Autunno								
X_2	*Fascia oraria*	Notte	Giorno										
X_3	*Localizzazione incidente*	Strada urbana	Provinciale entro l'abitato	Statale entro l'abitato	Comunale extraurbana	Provinciale	Statale	Autostrada	Altra strada				
X_4	*Tipo di strada*	Una carreggiata senso unico	Una carreggiata doppio senso	Due carreggiate	Più di due carreggiate								
X_5	*Pavimentazione*	Strada pavimentata	Strada pavimentata dissestata	Strada non pavimentata									
X_6	*Intersezione*	Incrocio	Rotatoria	Segnalata	Con semaforo o vigile	Non segnalata	Passaggio a livello						
	Non intersezione							Rettilineo	Curva	Dosso, strettoia	Pendenza	Galleria illuminata	Galleria non illuminata
X_7	*Fondo stradale*	Asciutto	Bagnato	Sdrucciolevole	Ghiacciato	Innevato							
X_8	*Segnaletica*	Assente	Verticale	Orizzontale	Verticale ed orizzontale								
X_9	*Condizioni meteorologiche*	Sereno	Nebbia	Pioggia	Grandine	Neve	Vento Forte	Altro					

Tale classificazione non prende in considerazione il veicolo come possibile causa di incidenti, né comportamenti patologici degli utenti, sia perché statisticamente irrilevanti (si pensi, ad es., che nel 1995 solo poco più dello 0,2% delle cause di incidente è riconducibile allo stato patologico del conducente del veicolo), sia perché si sono voluti considerare solo quelle variabili su cui è possibile intervenire per ridurre l'incidentalità.

5.2.1 Structural Learning

L'algoritmo di structural learning permette di determinare i vari collegamenti tra le variabili del dominio che si intende rappresentare.

La struttura della rete può essere rappresentata mediante una *matrice di adiacenza* analoga a quella mostrata in figura 5.2 per una rete la cui struttura è rappresentata in figura 5.1.

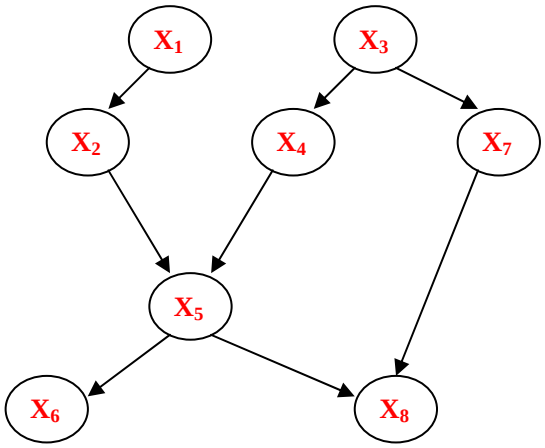


Figura 5.1: Rete Bayesiana

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
X_1	0	1	0	0	0	0	0	0
X_2	-1	0	0	0	1	0	0	0
X_3	0	0	0	1	0	0	1	0
X_4	0	0	-1	0	1	0	0	0
X_5	0	-1	0	-1	0	1	0	1
X_6	0	0	0	0	-1	0	0	0
X_7	0	0	-1	0	0	0	0	1
X_8	0	0	0	0	-1	0	-1	0

Figura 5.2: Matrice di adiacenza

La presenza di un 1 nella posizione [i,j] della matrice indica che il nodo i è padre del nodo j (ho un arco che va da i verso j); la presenza di un -1 nella posizione [i,j] indica che il nodo i è figlio del nodo j (ho un arco che va da j verso i); la presenza di un 0 nella posizione [i,j] indica, invece, che non vi è un arco tra i e j. Grazie a questa matrice si può facilmente visualizzare quali sono i padri e/o i figli di un determinato nodo.

Costruire la rete significa quindi determinare questa matrice di adiacenza.

Si parte da una rete vuota il che implica una matrice di adiacenza con tutti zeri. Vengono calcolati per ogni nodo i valori della SCORE Function e memorizzati in un apposito vettore di elementi pari al numero dei nodi. L'algoritmo entra così nei cicli di ricerca in cui vengono scanditi gli archi non ancora presenti nella struttura: si valuta il guadagno di SCORE ottenibile da un loro inserimento (se questo è possibile, mantenendo l'aciclicità del grafo). Se il guadagno è maggiore del massimo guadagno trovato nella corrente iterazione, si salva la mossa e si inserisce un 1 nella matrice di adiacenza. La stessa procedura viene utilizzata per ogni arco presente, valutando il guadagno della cancellazione e dell'eventuale inversione di verso mantenendo sempre l'aciclicità del grafo: se ho un incremento positivo del guadagno aggiorno gli SCORE relativi ai nodi pozzo e sorgente del nuovo arco e inserisco il valore opportuno nella matrice di adiacenza. Questa ricerca iterativa è proseguita fino a che non si raggiunge il massimo numero di iterazioni oppure fino a che non esiste più un'ulteriore mossa a guadagno positivo.

Per costruire la rete si percorre la matrice per la prima operazione partendo dal primo nodo e poi proseguendo secondo l'ordine topologico.

5.2.2 Parametr Learning

Una volta appresa la struttura bisogna apprendere quelle che sono le relazioni probabilistiche tra le variabili aleatorie che rappresentano i nodi della rete. Per ogni nodo avrò una tabella di probabilità che non è altro che una matrice le cui colonne sono i possibili stati del nodo considerato mentre il numero di righe rappresenta tutte le possibili combinazioni degli stati dei nodi padri.

L'apprendimento dei parametri avviene implementando la seguente formula:

$$p(x_i^k \mid pa_i^j, \theta_i, m) = \theta_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$$

calcolata per ogni riga di ciascuna tabella (che corrisponde ad una data configurazione dei padri del nodo in esame).

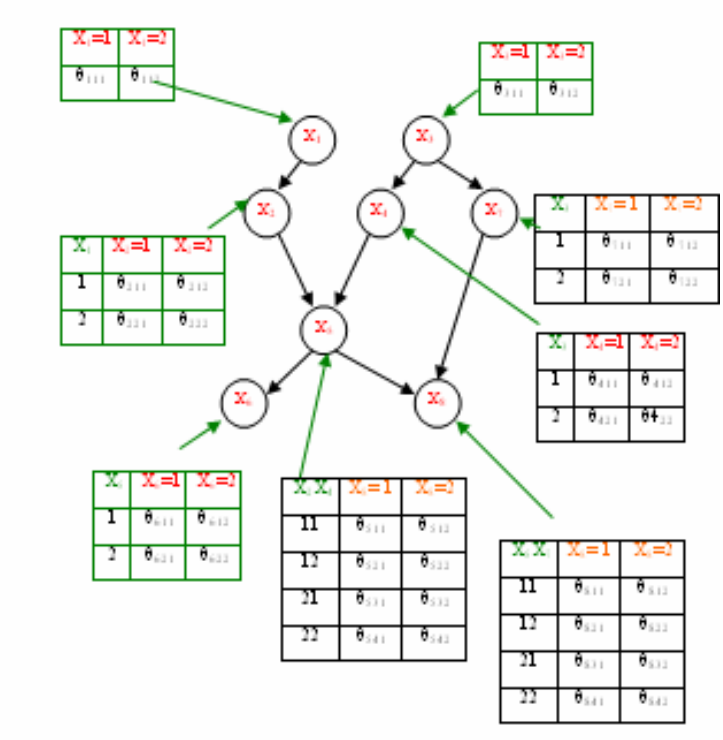


Figura 5.3: Rete Bayesiana con relative tabelle di probabilità

Gli N_{ijk} è il numero delle volte che nel database D si ha che $X_i = x_i^k$ e $Pa_i = pa_i^j$. Per poter avere una formula computabile bisogna assegnare dei valori

agli α_{ijk}. Questi iperparametri codificano la conoscenza a priori che l'utente ha sui parametri delle probabilità associate alla rete. Dato che abbiamo supposto una completa ignoranza, è logico porre $\alpha_{ijk} = \frac{\alpha}{q_i r_i}$.

Resta così da assegnare un unico "iper"iperparametro α che in letteratura è chiamato *dimensione di un campione equivalente*. Nel nostro caso poiché partiamo da una rete priva di archi senza dunque nessuna conoscenza a priori sui dati è necessario che il valore di α sia sufficientemente basso.

5.3 Implementazione

L'implementazione degli algoritmi per l'apprendimento della rete bayesiana partendo dal database di esempi è stata fatta utilizzando il linguaggio C++.

Segue una breve descrizione delle classi più interessanti necessarie all'apprendimento suddetto.

La classe *graph* implementa grafi aciclici orientati. I nodi sono rappresentati con dei numeri interi e gli archi con coppie di nodi, cioè con coppie di numeri interi. Un arco i → j è rappresentata da un 1 se è presente, da un -1 se è presente con il verso invertito e da uno 0 se non è presente. Per controllare l'aciclicità del grafo si utilizza la funzione *topologic_sort* che realizza un ordinamento topologico del grafo, attraversando per primi i nodi sorgente, poi i loro figli e così via. Se l'algoritmo riesce ad ordinare tutti i nodi, il grafo è aciclico.

La classe *query_database* serve per la gestione dei campioni che saranno prelevati da un file di testo e memorizzati in una matrice.

La classe *node* consente la costruzione della tabella delle probabilità condizionata associata ad ogni nodo. In essa sono presenti, tra le altre, la funzione *sijk* che restituisce il numero di volte in cui n_node = k e Pa(n_node) = j per ogni variabile X_i; *Mij* che restituisce il numero di volte in cui Pa(n_node) = j indipendentemente dal valore stesso di n_node; *aijk* che restituisce il singolo iper-parametro. Nella classe è presente anche la funzione *partial_score*, punto di partenza per il calcolo dello "score".

Nella classe *bn* si trovano le due funzioni principali: *structure_learning* e *learning_parameter*. Per lo structure_learning si fa riferimento all'algoritmo "Hill-Climbing" con un'opportuna modifica che ne migliora le prestazioni. L'Hill-Climbing è un algoritmo molto versatile, in quanto non presuppone nessuna conoscenza a priori riguardo l'ordinamento topologico dei nodi. Il suo spazio di ricerca è, quindi, il set di tutti i DAG contenenti le n variabili. Tale algoritmo si potrebbe fermare ad un massimo locale e, quindi, potrebbe non rilevare la struttura reale. Si può ovviare a tale problema avviando una procedura che preveda più partenze dell'algoritmo in parallelo inizializzando il grafo con più grafi casuali possibilmente abbastanza diversi tra loro.

La fase di training è particolarmente importante anche per evitare il problema dell'overfitting. Si parla di overfitting quando un modello statistico si adatta ai dati osservati. Nel caso specifico delle reti bayesiane, se il numero di esempi forniti in fase di allenamento è scarso, il modello potrebbe adattarsi a caratteristiche che sono specifiche solo del training set, ma non sono rappresentativi della realtà; perciò, in presenza di overfitting le prestazioni sui dati di allenamento aumentano, mentre le prestazioni sui dati non visionati saranno peggiori.

5.4 Conclusioni

Il sistema realizzato permette di individuare le situazioni a più alto rischio di incidente e quindi di gestire concretamente la sicurezza stradale in una qualunque rete. Ciò permetterà di individuare a priori, note le caratteristiche ambientali, geometriche, infrastrutturali, ecc. il livello di pericolosità di una strada.

Infine, si ritiene utile sottolineare che preresiquisito essenziale per lo studio e la comprensione dell'incidentalità stradale, indipendentemente dai criteri e dalle metodologie con cui si intende affrontare il problema, nonché per la valutazione dell'efficacia di interventi volti ad aumentare la sicurezza, è quello di disporre dei dati relativi agli incidenti accaduti. Questi ultimi, dovrebbero essere, oltre che affidabili, completi ed omogenei, quanto più possibile dettagliati, così da consentire analisi sia macroscopiche che microscopiche.

Purtroppo, allo stato attuale, la situazione in Italia non può dirsi affatto ottimale per diversi motivi ai quali si potrà solo in parte ovviare in futuro.

In generale, con il termine "incidente stradale" si intende ogni evento da cui conseguono danni a persone e/o cose per effetto della circolazione sulle strade di veicoli di qualsiasi tipo e persone; tuttavia se non c'è un intervento delle forze dell'ordine, questi incidenti non entrano nelle statistiche ufficiali.

Inoltre non vi è stata in passato, né è stata oggi completamente raggiunta, omogeneità nella redazione dei verbali da parte delle autorità di polizia. A partire dal 1991, adeguandosi a quanto avveniva già in molti paesi europei, gli organi che curano la raccolta e la diffusione dei dati di incidentalità (ISTAT e ACI) censiscono come incidenti stradali solo "gli eventi che si verificano su strade aperte alla circolazione pubblica in seguito ai quali una o più persone sono uccise o ferite e nei quali è implicato almeno un veicolo in movimento".

Un ulteriore adeguamento ai criteri seguiti nella comunità europea riguarda la mortalità: fino al 1998 in Italia un decesso era attribuito all'incidente se avveniva entro sette giorni; a partire dal 1 marzo 1998 il periodo necessario per l'attribuzione del decesso è passato a trenta giorni.

Va anche detto che le statistiche sanitarie segnalano una mortalità per incidenti stradali superiore al 25÷30% a quella dell'ACI-ISTAT, il che deriva dalla tendenza dei medici ad attribuire le morti agli incidenti stradali anche quando sono presenti concause (malore improvviso, spavento che provoca infarto, ecc.).

Per ovviare alle carenze appena accennate e ad altre per brevità non menzionate esplicitamente, in ottemperanza anche a quanto prescritto dal Nuovo Codice della Strada, è prevista la realizzazione di una Banca Dati da inserire nella sezione 3 (Incidenti) dell'Archivio Nazionale delle Strade; sarà così possibile disporre di dati sull'incidentalità con un elevato numero di informazioni, raccolti in modo uniforme su tutto il territorio nazionale, notevolmente affidabili e facilmente accessibili. [ESP]

BIBLIOGRAFIA

[AGR95]R. Agrawal e R. Srikant – Mining Sequential Patterns – Proc. of 11th Int'l Conf. on Data Engineering, IEEE CS Press, Los Alamitos, CA, 1995.

[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

[BUN94] W. L. Buntine, Operations for Learning with Graphical Models, 1994 AI Access Foundation and Morgan Kaufmann Publishers

[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

[CAR] Caroti, L., Lancieri, F., Losa, M., (1999) *Considerazioni su alcuni fattori di rischio dell'incidentalità stradale.* Rivista "Quarry and Construction"

[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

[CHEN96] M.S. Chen, J. Han, P.S. Yu – Data Mining: An Overview from a Database Perspective – IEEE Transaction on Knowledge and Data Engineering, Vol.8 N.6, Dec. 1996.

[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

[ESP] T. Esposito, R. Mauro – *La progettazione funzionale stradale* – Hevelius Edizioni

[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

[FAY96] U. Fayyad, G. Piatetsky - Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996

[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.

[FRI99] Nir Friedman, Iftach Nachman, Dana Peer, Learning Bayesian Network Structure from Massive datasets: the "sparse candidate" algorithm, Proceedings of

15th Conference on UAI, IEEE 1999

[GAN99]V. Ganti, J. Gehrke e R. Ramakrishnan, 1999 – Mining Very Large Database – IEEE Computer, Aug.1999, n.8, Vol.31.

[GIA 01] Giannattasio P., Domenichini L., Giuffrè O., Granà A., Grossi J. - "Linee guida per le analisi di sicurezza delle strade" – rivista "Circolazione e Sicurezza Stradale", n. 3/2001.

[GUO02] Hsu, W. H., Guo, H., Perry, B. B., & Stilson, J. A. A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In Proceedings of the Genetic and Evolutionary Computation Conference, New York, NY. San Francisco, CA: Morgan Kaufmann

[HAN99] J. Han, L. Lakshmanan e R. Ng, 1999 – Constraint-Based, Multidimensional Data Mining – IEEE Computer, Aug.1999, n.8, Vol.31.

[HEC94] David Heckerman, Dan Geiger, David M. Chickering, Learning Bayesian Network: the combination of Knowledge and statistical data, Machine Learning, 20:197-243,1995

[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

[HEC97] David Heckerman, Bayesian Networks for Data Mining, Journal of knowledge Discovery and Data Mining 1(1), pag. 79-119 (1997), Kluwer Acedemic Publishers

[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

[JEN96] Finn V. Jensen, An introduction to Bayesian networks, Springer (1996).

[KAN62] A.B.Kahn, Topological Sorting of Large Network, Comunication of the ACM, vol.5,558-562,1962

[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

[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

[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

[MAN] V.A.Manganaro, Reti neurali nel data mining, altre tecniche utilizzate nel DM e valutazione dei modelli,www.statistica.too.it

[MAN] V.A.Manganaro,Tecniche di DM: Alberi di decisione e algoritmi di classificazione, www.statistica.too.it

[MAT93] C.J.Matheus, P.K.Chan, and G.Piatetsky-Shapiro, "System for Knowledge Discovery in Databases", IEEE Transaction on Knowledge and Data Engineering, Vol. 5, NO.6, December,1993

[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.

[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

[PAP] Enrico Papalini, Michele Piccinini, Apprendimento di reti bayesiane da database di esempi, Università di Firenze

[PAP] Athanasios Papoulis, Probability, Random Variables, and Stochastic

[PIA91]G. Piatetsky-Saphiro, W. J. Frawley, (Eds.) – Knowledge Discovery in databases – AAAI/MIT Press, Menlo Park, CA, 1991.

[PRO] John Proakis, Masoud Salehi, Communication Systems Enginnering,

[SCH00]M.Schroder, H.Rehrauer, K.Seidel, and M.Datcu, Interactive Learning and Probabilistic Retrieval in Remote Sensing Image Archives,IEEE Transaction on Geoscience and Remote Sensing, Vol. 38, NO.5,September 2000

[SCH] M.Schroder, H.Rehrauer, K.Seidel, and M.Datcu, Query by Image Content from Remote Sensing Image Archives, IEEE

[SCH]M.Schroder, K.Seidel, and M.Datcu, Bayesian Modeling of Remote Sensing Image Content ,IEEE

[SIT] Manuali SITEB – 2004 *"Manutenzione delle pavimentazioni stradali"*

[STA] Standing Committee on Highways Traffic Safety: *"Highway safety strategic plan 1991-2000"* American Association of State Highway and Transportation Officials, Washington, D.C., 1990;

[STE00] Todd A. Stephenson, An introduction to Bayesian network theory and usage, IDIAP Research Report IDIAP – RR - 00-03, 2000

[TIG] S.Tighe, N.Li, L.C.Falls, R.Haas: *"Incorporating road safety in pavement management"*, Transportation Research Record 1699 pp.1-10

APPENDICE A

Richiami della Teoria della 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 delle relazioni 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

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 applicabile 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 dell'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 punto di vista probabilistico (si pensi ai pronostici sportivi).

Proprietà della misura della probabilità

$P(-A) = 1 - P(A)$

$P(\Phi) = 0$

$0 \leq P(A) \leq 1$

$P(A_1 \cup A_2) = P(A_1) + P(A_2) - P(A_1 \cap A_2)$

Dati due eventi A e B di S,valutiamo la probabilità condizionata ovvero la probabilità di B nell'ipotesi che A si sia già verificato.

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

Tale definizione ha senso se P(A) > 0 ed A è assunto come spazio campionario.

Segue la definizione di indipendenza, due eventi A e B sono indipendenti se P(A∩B) = P(A)P(B).

Ogni volta che un evento A può essere visto come effetto di uno tra k possibili eventi C_1, C_2, …, C_k incompatibili e tali che solo uno di essi si possa verificare ed interessa valutare la probabilità $P(C_i|A)$ è possibile invocare la formula di Bayes

$$P(C_i|A) = \frac{P(C_i \cap A)}{P(A)} = \frac{P(C_i)P(A|C_i)}{\sum_{j=1}^{k} P(C_j)P(A|C_j)}$$

dove l'espressione P(A) ottenuta al denominatore è nota come "Teorema della probabilità totale".

APPENDICE B

Test del Chi-Quadro

Si consideri una popolazione statistica le cui unità siano raggruppate secondo le classi $A = \{A_1, A_2, \ldots, A_r\}$ e $B = \{B_1, B_2, \ldots, B_t\}$ 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*.

	A						
B	$\mathbf{A_1}$	$\mathbf{A_2}$		$\mathbf{A_i}$		$\mathbf{A_r}$	
$\mathbf{B_1}$	p_{11}	p_{21}	…	p_{i1}	…	p_{ri}	p_1
$\mathbf{B_2}$	p_{12}	p_{22}	…	p_{i2}	…	p_{r2}	p_2
…	…	…	…	…	…	…	…
$\mathbf{B_j}$	p_{1j}	p_{2j}	…	p_{ij}	…	p_{rj}	p_j
…	…	…	…	…	…	…	…
$\mathbf{B_r}$	p_{1r}	p_{2r}	…	p_{ir}	…	p_{rr}	p_r
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 (Ai, Bj). 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_{ri}	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

$$p_{ij} = p(A_i , B_j)$$

Ma se consideriamo A_i e B_j come due eventi indipendenti allora risulta

$$p_{ij} = p(A_i,B_j) = p(A_i)\, p(B_j) = p_i\, p_j$$

essendo $p(A_i) = p_i$ e $p(B_j) = p_j$

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

$$\mathbf{H_0 : p_{ij} = p_i\, p_j} \quad i = 1, 2, \ldots, r;\ j = 1, 2, \ldots, t$$

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_ip_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

$$\hat{p}_i = \frac{n}{n}\hat{p}_i = \frac{n_j}{n}$$

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 la cui pdf[25] è:

$$f(x) = \frac{1}{2^{r72}\Gamma(r/2)} x^{r/2-1} e^{-x/2}$$

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):

$$SL = p(S \text{ nella zona di rifiuto} \mid H_0 \text{ vera})$$

[25] Probability density function

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).

Chiariamo il tutto 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

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:

$$\chi^2 = \frac{\left[141 - \left(\frac{150 \cdot 210}{230}\right)\right]^2}{\frac{150 \cdot 210}{230}} + \ldots + \frac{\left[11 - \left(\frac{80 \cdot 20}{230}\right)\right]^2}{\frac{80 \cdot 20}{230}} = 3,94$$

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).

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 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)

$\mathbf{C_i}$	$\mathbf{A_iC_i}$	$\mathbf{A_2C_i}$
$\mathbf{B_1C_i}$	n_{A1B1Ci}	n_{A2B1Ci}
$\mathbf{B_2C_i}$	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).