Tecniche di DM: Link analysis e Association discovery

Vincenzo Antonio Manganaro

vincenzomang@virgilio.it, www.statistica.too.it

Indice

La *link analysis* (analisi di legame) è un approccio descrittivo di esplorazione dei dati, che può aiutare ad identificare relazioni fra gli attributi in un database, attraverso l'analisi dei valori del database. I due approcci più comuni alla link analysis sono *l'association discovery* (scoperta di associazioni) e la *sequence discovery* (scoperta di sequenze). Tratteremo l'association discovery, cioè la scoperta di *regole di associazione*, in questo capitolo mentre la scoperta di sequenze, che rientra nell'ambito dei modelli sequenziali/temporali, sarà trattata nel capitolo 4. L'applicazione di *market basket analysis*, che avremo modo di considerare nell'ultimo capitolo della tesi, è un esempio ben noto di uso di regole di associazione.

1 Architettura di un generico algoritmo di DM.

Abbiamo più volte affermato come l'obiettivo principale delle tecniche di DM consista nella ricerca di regolarità e/o modelli che servono per descrivere meglio i dati. Per cui data una classe P di modelli o proposizioni che descrivono le proprietà dei dati, attraverso tali tecniche, dovremmo poter specificare se un particolare modello $p \in P$ si verifica con elevata frequenza e se, d'altra parte, è un modello interessante per i fini pratici che si propone l'utilizzatore. Il generico obiettivo di una procedura di DM consiste quindi nella ricerca del set di modelli o proposizioni $PI(d, P) = \{p \in P | p$ si verifica un numero sufficiente di volte nei dati d e p risulta interessante$\}$, cioè quel set di modelli, sottoinsieme di P, costituito da quei modelli p che si adattano ai dati d con una determinata frequenza e che sono interessanti per l'utilizzatore. Un formalismo diverso da quello già visto potrebbe consistere nel considerare un insieme di proposizioni espresse in un determinato linguaggio L e considerare il DM come un problema di ricerca di frasi nel linguaggio L che sono sufficientemente vere per i dati e successivamente seguire i criteri dell'utilizzatore, per quanto riguarda l'interesse di tali proposizioni. D'altro canto, mentre le frequenze con cui i modelli riescono a spiegare "bene" i dati o la verità di una proposizione possono essere rigorosamente definite e determinate, l'interesse di un modello o di una proposizione sembra più difficile da spiegare e misurare. L'obiettivo generale di una qualsiasi procedura di DM è dunque la ricerca dell'insieme $PI(d, P)$. L'algoritmo generale per trovare tale insieme consiste nel trovare prima tutti i modelli che si verificano (nei dati) con una certa frequenza e poi selezionare dall'output ottenuto i modelli ritenuti più interessanti. La sua struttura può essere più o meno schematizzata in tal modo:

- *assumiamo che ci sia un ordinamento $\prec$ definito tra i modelli dell'insieme P per cui per ogni $p \in P$ e $q \in P$ con $q \neq p \Rightarrow p \prec q$ oppure $q \prec p$, cioè un ordinamento tale che se accade che $q \prec p$, allora p accade più di frequente di q ed è più idoneo a descrivere il comportamento dei dati* -.

Si possono dunque definire le seguenti istruzioni:

1. $C := \{p \in P |$ **per nessun** $q \in P$ **abbiamo che** $q \prec p\}$;
   si tratta di una istruzione di assegnazione . L'insieme C viene assegnato come un insieme di modelli $p \in P$ per cui non esiste nessun modello $q \in P$ tale che

$q \prec p$, vale a dire C in questo primo passo è costituito da tutti i modelli di P sui quali ancora non vi è ordinamento.

2. **Mentre** $C \neq \oslash$ **fai**
   {
3. **per ogni** $p \in C$
4. **trova il numero delle volte che il modello p si verifica nei dati d;**
   finché $C \neq \oslash$ l'algoritmo trova per ogni $p \in C$ il numero di volte che il modello si verifica nei dati (cioè la frequenza con cui ciascun modello riesce a spiegare i dati).
5. $F := F \cup \{p \in C | p$ **è sufficientemente frequente in** $d\}$;
   si tratta di un'altra istruzione di assegnazione che definisce iterativamente l'insieme F come costituito dall'insieme F precedentemente ottenuto al quale si aggiungono i modelli p che si verificano con sufficiente frequenza nei dati d.
6. **C:=**$\{p \in P|$ **ogni** $q \in P$ **con** $q \prec p$ **è stato già considerato e** p **è frequente**$\}$;
   in questa istruzione di assegnazione si modifica il contenuto di C, che viene ad essere costituito dai soli modelli $p \in P$ tali che tutti i modelli $q \in P$ con $q \prec p$ sono già stati considerati ed esclusi dall'insieme C e che ciascun modello $p \in P$ sia frequente. Si tratta del passo dell'algoritmo che mi assicura il fatto che l'insieme C tende all'insieme vuoto. Se cosi non fosse l'algoritmo non avrebbe fine e si cadrebbe in un loop.
7. **END;**
   }
8. **Output F.** Sull'output F, costituito da modelli certamente frequenti, l'utilizzatore può ricavare attraverso determinati criteri i modelli ritenuti interessanti. L'algoritmo descritto è inoltre estendibile con le relative modifiche a tutte le procedure di DM. Vedremo un caso particolare di questo algoritmo nei successivi paragrafi e nel contesto specifico delle regole di associazione.

2 Regole di associazione: definizioni ed esempi.

Una regola di associazione è rappresentata dai simboli $X \Rightarrow Y$ dove X è chiamato *antecedente* e Y *conseguente*. Per esempio, nella regola di associazione "se le persone comprano un martello, allora comprano chiodi", l'antecedente è "comprare un martello" e il conseguente è "comprare chiodi". Formalmente, se $L = A_1, A_2, \ldots, A_p$ è un insieme di letterali o attributi che rappresentano le qualità di ciascun record nel database, indicato con $P(L)$ l'insieme delle parti di L, $P(L)$ rappresenterà l'insieme di tutti i sottoinsiemi di L e dunque l'insieme di tutte le combinazioni di attributi. Un record del database potrà essere rappresentato da un opportuno $r \in P(L)$ e potremmo inoltre indicare con $R \subseteq P(L)$ l'insieme di tutti i records del database (R può ovviamente contenere records uguali). Nella regola di associazione $X \Rightarrow Y$, individuata attraverso l'esame dell'insieme R, si ha $X \subseteq L$ e $Y \in L|X$ cioè X è un sottoinsieme di attributi e Y è un singolo attributo che non coincide con nessuno degli attributi del sottoinsieme X, ovvero, in termini insiemistici, $\{X \cap Y\} = \oslash$.

- *Esempio 1*

Supponiamo che il nostro database sia il seguente:

A	B		D	E	F
A	B	C		E	F
	B	C	D	E	
A	B		D		F
A	B	C	D	E	F
A	B			E	
A	B	C		E	F
	B	C			
	B	C	D	E	F
A	B		D	E	

In base a quanto definito avremmo $L = \{A, B, C, D, E, F\}$ insieme di attributi, $P(L) = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}, \{A, B\}, \{A, C\}, \ldots, \{A, B, C, D, E, F\}, \oslash\}$ ed $R = \{r_1, r_2, \ldots, r_{10}\}$ (R in tal caso non ha elementi uguali). Si possono determinare ad esempio le seguenti regole di associazione:

- la regola $A, B \Rightarrow E$ con $X = \{A, B\}$ e $Y = E \in L|X$
- la regola $B \Rightarrow E$ con $X = \{B\}$ e $Y = E \in L|X$
- la regola $A, B, C \Rightarrow F$ con $X = \{A, B, C\}$ e $Y = F \in L|X$
- la regola $D, E \Rightarrow F$ con $X = \{D, E\}$ e $Y = F \in L|X$.

Files di dati cui possiamo applicare regole di associazione possono essere, ad esempio, i dati raccolti tramite i lettori dei codici a barre nei supermercati: in tal caso le colonne rappresentano gli attributi costituiti dalla presenza/assenza di un prodotto, mentre ciascuna riga rappresenta una singola transazione, ovvero un insieme di prodotti acquistati nella transazione (vedi il dataset di esempio sopra). Un altro esempio dove invece è più utile l'approccio della sequence discovery[1] può essere rappresentato da un insieme di misurazioni relative al comportamento di un sistema. Potremmo pensare ai centralini in una rete telefonica; in tal caso le colonne corrispondono alla presenza/assenza di determinate condizioni ed ogni riga corrisponde ad una determinata misurazione. Se la cella (m, c) del dataset vale 1 significa che alla misurazione m era presente la condizione c. E' chiaro che in un database si possono individuare moltissime regole di associazione, in teoria alle regole precedenti si potrebbe aggiungere la regola $A, D \Rightarrow F$ con $X = \{A, D\}$ e $Y = F \in L|X$ (si tratta di una regola scritta a caso che potrebbe avere una sua validità!). Risulta altrettanto chiaro che non tutte le regole sono importanti; alcune possono non avere rilevanza statistica nei termini che adesso spiegheremo, altre pur avendo una forte rilevanza statistica sono ovvie, altre ancora potrebbero rilevare una perfetta associazione, ma spiegata dal fatto che abbiamo a che fare con prodotti venduti insieme. Il problema è dunque trovare delle regole che siano innanzitutto statisticamente rilevanti, cercare poi di esaminare le informazioni per determinare le ragioni delle relazioni esistenti e infine verificare la possibilità di sfruttare in qualche modo le relazioni indicate.

[1] La scoperta di sequenze sarà affrontata nel capitolo 4 della tesi poichè esse riguardano per lo più i modelli sequenziali temporali.

3 Regole di associazione: criteri per una rilevanza statistica.

Il primo obiettivo, ovvero l'individuazione di relazioni statisticamente valide nel database, è raggiunto con l'introduzione dei concetti di *supporto* o *prevalenza* e di *confidenza:*

- **si definisce supporto[2] (s) della regola di associazione $X \Rightarrow Y$ il numero di records nel database per i quali si verifica la presenza contemporanea degli attributi che compongono la regola di associazione.**
- **si definisce confidenza (c) della regola di associazione $X \Rightarrow Y$ la frazione di records del database per i quali la presenza di X implica la presenza di Y.**

Nell'esempio 1 si ha:

- la regola $A, B \Rightarrow E$ ha supporto 6 (6/10 pari al 60%) e confidenza 6/7.
- la regola $B \Rightarrow E$ ha supporto 8 (8/10 pari al 80%) e confidenza 8/10.
- La regola $A, B, C \Rightarrow F$ ha supporto 3 (3/10 pari al 30%) e confidenza 3/3.
- la regola $D, E \Rightarrow F$ ha supporto 3 (3/10 pari al 30%) e confidenza 3/5.

Il supporto di una regola di associazione dà l'idea della base su cui poggia la regola stessa: un basso valore del supporto (per esempio, in un database di transazioni solo una transazione su un milione rivela la contemporanea presenza di due prodotti) indica che la regola non è statisticamente rilevante e quindi importante. La confidenza risponde al quesito: dato il verificarsi dell'antecedente X, quanto spesso si verifica il conseguente Y? Un'altra misura della potenza di un'associazione è il *Lift*. Il Lift misura l'influenza che il verificarsi dell'antecedente X ha sul verificarsi del conseguente Y. Esso è dato dalla seguente espressione:

$$Lift(X \Rightarrow Y) = \frac{confidenza(X \Rightarrow Y)}{frequenza(Y)} \tag{1}$$

[2]Useremo i corrispondenti termini in italiano laddove sarà possibile individuarli (vedi il caso di supporto e confidenza). In caso contrario useremo i termini in lingua inglese (vedi misure quali il lift).

Più grande è il Lift, più grande è l'influenza che il verificarsi di X ha sul verificarsi di Y. Da notare inoltre l'analogia dell'espressione del Lift con la classica formula delle probabilità condizionate, secondo cui se A e B sono due eventi si ha che $P(A|B) = P(A \cap B)/P(B)$. La ricerca di regole di associazione si può ricondurre alla ricerca, fra tutte le possibili regole costruibili tra gli attributi che compaiono nel database, di quelle che soddisfano un minimo di supporto e un minimo di confidenza: in pratica si vogliono trovare tutte le regole di associazione $X \Rightarrow Y$ con supporto maggiore od uguale ad un dato s e con confidenza maggiore od uguale ad un dato c.

- *Esempio 2*

Supponiamo che il seguente sia un database di transazioni:

TRANSAZIONI	**OGGETTI ACQUISTATI**
1	A,B,C
2	A,C
3	A,D
4	B,E,F

Se assumiamo come supporto minimo il 50% e come confidenza minima il 50% otteniamo:

TRANSAZIONI	**OGGETTI ACQUISTATI**
1	**A**,B,**C**
2	**A**,**C**
3	**A**,D
4	B,E,F

Regole ottenute:

- $A \Rightarrow C$ con supporto del 50% e confidenza del 66,6%
- $C \Rightarrow A$ con supporto del 50% e confidenza del 100%.

Gli algoritmi di associazione trovano queste regole calcolando il supporto e la confidenza per ciascuna regola trovata ed escludendo quelle regole che non rientrano

nei parametri di supporto e confidenza. L'efficienza con cui i vari algoritmi arrivano ad individuare le regole che soddisfano tali requisiti è uno degli elementi differenziatori fra i vari algoritmi ed in un contesto di DM, trattando con grossi database, il problema dell'efficienza diventa ancora più rilevante. Quasi tutti gli algoritmi di ricerca di regole di associazione inclusi nei software per il DM permettono, inoltre, di trovare tutte le regole che rispondono a requisiti ancora più specifici dei classici vincoli di supporto e confidenza. Ad esempio, si può chiedere al software di trovare tutte quelle regole in cui un particolare attributo Y sia il conseguente e tali che le regole trovate abbiano un supporto minimo e una confidenza minima. Al fine di evitare la scelta di regole con un basso valore di supporto e un basso valore di confidenza, si considera un ulteriore limite inferiore sotto il quale la regola viene esclusa, anche se rientra nei limiti di supporto e confidenza minimi prescritti. Questo ulteriore parametro indicato con m e denominato *comunanza*, è un valore sotto il quale il prodotto dei valori di supporto e confidenza non deve scendere.

4 L'algoritmo per la ricerca di regole di associazione.

Nella scoperta delle regole di associazione l'obiettivo è trovare tutte le regole $X \Rightarrow B$, tali che il supporto della regola sia almeno uguale ad un dato valore soglia s e la confidenza della regola sia almeno uguale ad una dato valore soglia c. Nelle applicazioni di DM il numero delle righe di un database può essere dell'ordine di 10^6 o persino 10^8 e il numero di colonne attorno a 5000. Il valore soglia del supporto (s) tipicamente va da 10^{-2} a 10^{-4} e il valore soglia di confidenza è generalmente compreso tra 0 e 1[3]. In queste condizioni non è raro ottenere da un tale database centinaia o centinaia di migliaia di regole di associazione che rientrano nei parametri richiesti. La crescita esponenziale del numero di regole di associazione è inoltre legata al fatto che il numero di attributi dell'antecedente non è fisso, né d'altro canto è fisso il conseguente; questo modo di procedere, se da un lato permette l'individuazione di regole inaspettate, che in tal modo non vengono automaticamente escluse dalla procedura di ricerca (se si fissasse il conseguente, determinate regole di associazione

[3]Non potrebbe essere altrimenti visto che la confidenza è definita come una frazione dei records sul totale.

importanti verrebbero escluse a priori), dall'altro determina la crescita esponenziale del numero di regole che soddisfano i valori soglia cui accennavamo precedentemente. Se diciamo che un insieme X è frequente quando il supporto di X $s(X) \geq s$, una volta individuati tutti gli insiemi frequenti, trovare le regole di associazione è semplice, dato che per ogni insieme X frequente e per ogni $B \in X$ si può agevolmente verificare se la regola $X|B \Rightarrow B$ ha un valore di confidenza sufficientemente alto o al limite $\geq c$. Il problema si riduce allora alla necessità di individuare tutti gli insiemi X di attributi frequenti e tale problema viene risolto adattando l'algoritmo generale di una tecnica di DM al caso specifico delle regole di associazione. L'algoritmo sfrutta inoltre l'importante proprietà (tra l'altro molto intuitiva) che "tutti i sottoinsiemi di insiemi frequenti sono anch'essi frequenti"; esso opera nel seguente modo: in un primo momento si trovano tutti gli insiemi di ampiezza 1, ovvero da una prima lettura del database si registra il numero di volte in cui ogni attributo A si verifica. Quindi gli insiemi di ampiezza 1 frequenti, cioè quelli per cui $s(A) \geq s$, sono candidati ad essere insiemi di ampiezza 2, ovvero insiemi costituiti da coppie di attributi $\{B, C\}$ che sono entrambi frequenti o tali che $s(\{B, C\}) \geq s$. Non appena si individuano tutti gli insiemi frequenti di ampiezza 2 tali insiemi diventano candidati ad essere insiemi di ampiezza 3, ovvero insiemi costituiti da terne di attributi $\{B, C, D\}$ tali che $s(B, C, D) \geq s$. Sembra quindi chiaro il meccanismo con cui l'algoritmo procede. Inoltre, per quanto detto precedentemente e cioè che tutti i sottoinsiemi di insiemi frequenti sono anch'essi frequenti, se $s(\{B, C, D\}) \geq s$ segue necessariamente che $s(\{B, C\}) > s$, $s(\{B, D\}) > s$ e $s(\{C, D\}) > s$; per cui se si verifica che, ad esempio, $s(C, D) < s$ non può mai verificarsi che $s(\{B, C, D\}) \geq$ s e l' insieme $\{C, D\}$ non è candidato a formare insiemi di grandezza 3. In generale nel momento in cui non vi sono più insiemi di grandezza $n - 1$ candidati a formare insiemi di grandezza n il procedimento si arresta. Diamo qui una semplice rappresentazione dei passi dell'algoritmo descritto:

1. $C := \{\{A\}|A \in R\}$;
2. $F := \oslash$;
3. $i := 1$;
4. mentre $C \neq \oslash$ fai

5. $\overline{F}$:=gli insiemi $X \in C$ che sono frequenti;
6. aggiungi $\overline{F}$ a F;
7. C:= insiemi Y di grandezza $i+1$ tali che
8. ogni sottoinsieme W di Y di grandezza i è frequente;
9. $i := i + 1$;
10. END;

Questo algoritmo è costretto a leggere il database almeno $k+1$ volte essendo k la grandezza del più grande insieme frequente. Nelle applicazioni pratiche k è piccolo, generalmente $k = 10$, per cui il numero di passi dell'algoritmo attraverso i dati è computazionalmente ragionevole.

5 Regole di associazione interessanti e non.

Sebbene determinate regole di associazione siano statisticamente rilevanti (nel senso che soddisfano i valori soglia di supporto e confidenza) esse possono non essere interessanti. Esistono varie ragioni per cui tali regole possono fallire in riferimento al loro grado di interesse per l'utilizzatore.
Consideriamo i seguenti esempi:

- **esempio 1:** In un database di iscrizioni a corsi di informatica la regola di associazione *chi è iscritto al corso di Introduzione a Unix* $\Rightarrow$ *è iscritto al corso di Programmazione in C* ha una coppia di valori (s, c) pari a (0.34, 0.84), cioè stabilisce che il 34% degli studenti frequentano entrambi i corsi e che l'84% degli studenti che si sono iscritti al corso di Introduzione a Unix si sono iscritti al corso di Programmazione in C.
  La regola relativa a questo esempio è statisticamente rilevante dal momento che ha dei valori di supporto e confidenza che sono certamente superiori a possibili valori soglia stabiliti a priori, ma non è per nulla interessante poiché è assolutamente normale che l'84% degli studenti che hanno frequentato il corso

base di Introduzione a Unix frequentino il corso più avanzato di Programmazione in C[4]. Questo tipo di regole, peraltro statisticamente rilevanti, **sono poco interessanti in quanto prevedibili e per certi versi scontate** (è opportuno ricordare come il DM si proponga di individuare regole che non siano *scontate* ...).

- **esempio 2:** Nel medesimo database di iscrizioni a corsi di informatica la regola di associazione *chi è iscritto al corso di Fondamenti di Informatica* ⇒ *è iscritto al corso di Programmazione in Pascal* ha una coppia di valori (s, c) pari a (0.60, 0.95).
  La regola relativa a questo esempio è senza dubbio statisticamente rilevante ma dal momento che i due corsi sono corsi base (e questo l'utilizzatore lo sa, perché prima ha suddiviso i corsi in classi) se l'utilizzatore è interessato alle relazioni tra corsi base e corsi avanzati, o più in generale alle relazioni tra corsi appartenenti a diverse classi, non troverà per nulla importante questa regola.

Vi possono essere inoltre casi di **regole di associazione ridondanti**, ovvero di regole che, in virtù di altre statisticamente rilevanti e interessanti, risultano scontate, dal momento che discendono logicamente da queste ultime. Ma se è vero che i motivi per cui determinate regole non sono interessanti vanno oltre quelli sin qui esaminati, il problema è ora trovare dei metodi che descrivano come trovare le regole più interessanti tra quelle statisticamente rilevanti.
Classi di regole interessanti possono essere specificate con l'utilizzo di *templates*, cioè di basi su cui poggiare una regola. Un template descrive un insieme di regole specificando quali attributi debbano verificarsi nell'antecedente e quale nel conseguente. Se l'utilizzatore è interessato a quelle regole che come conseguente hanno un determinato attributo, può chiedere al software di individuare tutte le regole statisticamente rilevanti con quell'attributo come conseguente. Le regole così trovate, per definizione, saranno regole interessanti per l'utilizzatore. Di contro l'uso di templates non permette di trovare regole inaspettate poiché, una volta stabilite le basi su cui poggiare le regole interessanti, vengono a priori escluse dall'output dell'algoritmo.

[4]Considerato soprattutto che il sistema operativo UNIX è costruito prevalentemente utilizzando il linguaggio C.

In generale, un template è un'espressione del tipo $A_1, A_2, \ldots, A_k \Rightarrow A_{k+1}$ essendo A_i una classe comprendente vari attributi o un singolo attributo. Ad esempio, possiamo raggruppare vari attributi nelle varie classi e trovare quelle regole di associazione il cui conseguente è A_{k+1} e il cui antecedente è costituito da attributi la cui combinazione è $(A_1, A_2, \ldots, A_k)$. L'utilizzatore specifica quali regole sono per lui importanti e quali no definendo un template nel modo visto. In generale i software di DM sono dotati di tools che prevedono, per chi li usa, la possibilità di esplicitare fondamentalmente due diversi tipi di templates:

- **templates di tipo inclusivo**
- **templates di tipo restrittivo.**[5]

Con il primo tipo si sceglieranno regole che soddisfano quel template, con il secondo tipo si escludono tutte le regole descritte da quel template. Riportiamo per concludere degli esempi in riferimento ai due tipi di templates:

- **esempio 3:** Se consideriamo tre classi di corsi, cioè CORSI BASE, CORSI MEDI e CORSI AVANZATI, e una classe comprendente qualunque corso dei precedenti, denominata QUALUNQUE CORSO, supposto che il corso di DISEGNO E ANALISI DI ALGORITMI appartenga ai corsi avanzati, il seguente è un esempio di template di tipo inclusivo:
  CORSI AVANZATI, QUALUNQUE CORSO ⇒DISEGNO E ANALISI DI ALGORITMI
  con il quale l'utilizzatore sceglie una qualunque tra le regole che abbiano come conseguente il corso di DISEGNO E ANALISI DI ALGORITMI e come antecedente la combinazione di un qualunque corso con un corso avanzato. L'utilizzatore considera come non interessanti le regole che, ad esempio, contengono solo corsi base nell'antecedente o altre che non rientrano nel tipo di template considerato.

- **esempio 4:** Se consideriamo la suddivisione dei corsi in classi dell'esempio precedente il seguente è un esempio di template di tipo restrittivo:
  CORSI BASE, QUALUNQUE CORSO ⇒ QUALUNQUE CORSO

[5] In realtà non esistono templates inclusivi e restrittivi in assoluto: un medesimo template, nel software di riferimento, può essere usato in modo inclusivo o restrittivo.

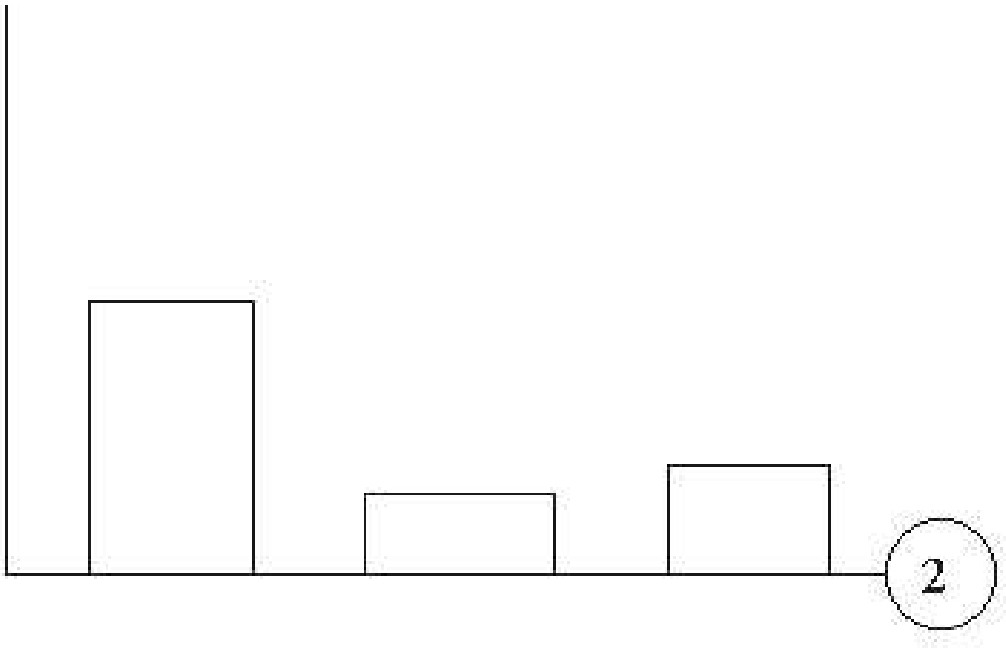

Figura 1: Esempio di visualizzazione di una singola regola di associazione con grafico a barre.

con il quale l'utilizzatore esclude tutte quelle regole che hanno almeno un corso base nell'antecedente.

6 Visualizzazione di regole di associazione.

Consideriamo ora il problema della visualizzazione di regole di associazione. Vi è la possibilità di rappresentare graficamente sia una regola di associazione sia un insieme di regole di associazione. La figura 2.1 mostra un esempio di rappresentazione di una singola regola di associazione per mezzo di un grafico a barre. Le barre all' estrema sinistra e alla estrema destra rappresentano rispettivamente il supporto e la confidenza della regola, mentre la barra centrale (più bassa delle laterali) rappresenta la comunanza. Il numero a destra del grafico rappresenta il numero di attributi dell'insieme X nella generica regola $X \Rightarrow Y$. Un altro modo di visualizzare singole regole di associazione consiste nel creare dei rettangoli, nei quali ogni lato rappresenta il supporto e la confidenzaconfidenza della regola, cosicché l'area ne rappresenta chiaramente la comunanza.

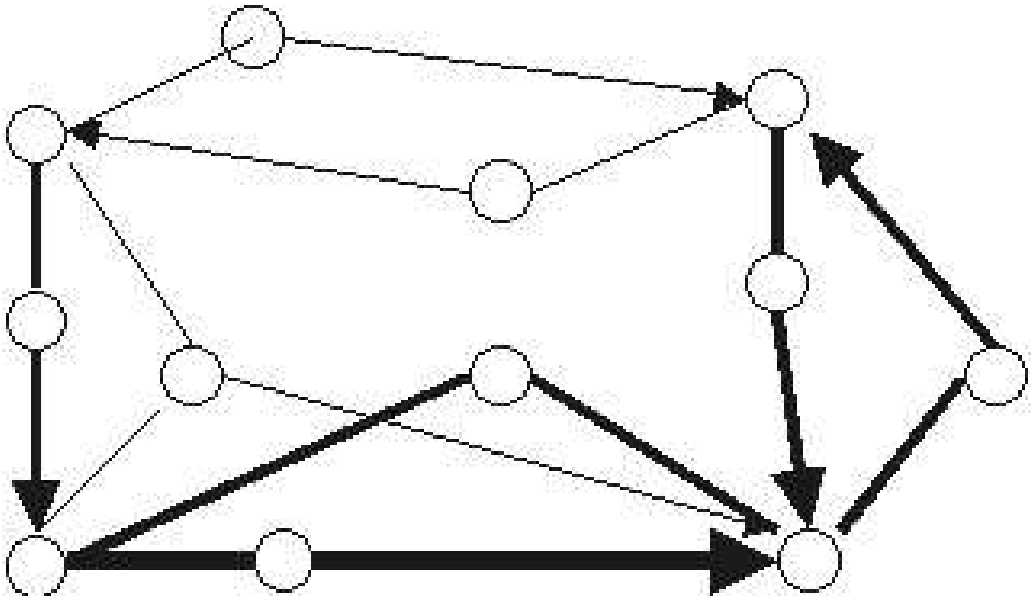

Figura 2: Esempio di visualizzazione di un set di regole di associazione. Le frecce più spesse rappresentano i legami più forti.

Tuttavia spesso è opportuno, nonché più intuitivo, rappresentare molte regole di associazione contemporaneamente. Una rappresentazione molto semplice ha come idea base quella di rappresentare gli attributi attraverso dei nodi e le associazioni attraverso delle frecce orientate in un diagramma tipo quello di fig. 2.2, nel quale le frecce più spesse rappresentano le associazioni più forti.

7 Conclusioni.

La scoperta di sequenze è formalmente analoga alla scoperta di associazioni. L'elemento che li differenzia è il fattore tempo: infatti, mentre le regole di associazione riguardano gli attributi che si verificano insieme in un evento, quale ad esempio una transazione commerciale, le sequenze costituiscono delle regole che si manifestano, ad esempio, nel corso di due o più transazioni commerciali avvenute in tempi successivi. Affronteremo comunque più dettagliatamente la scoperta di sequenze nell'ambito dei modelli sequenziali/temporali. A questo punto ci potremmo porre i seguenti quesiti: quanto reali sono le regole di associazione trovate? In che modo è possibile sfruttare le regole trovate per fini di tipo commerciale o in generale per fini

di business? Per rispondere al primo quesito diremo che le associazioni o le frequenze trovate non esistono realmente; piuttosto si tratta di descrizioni di relazioni in un particolare database. Non esiste nessuna possibilità di estendere i risultati trovati per quel database ad altri dati, in maniera tale da estendere la potenza predittiva di queste regole. Quindi una regola di associazione, statisticamente rilevante ed interessante per un database di un centro commerciale, può essere statisticamente non rilevante per un altro centro e anche nell'ambito dello stesso database l'utilizzo di regole di associazione per fini commerciali si basa sull'assunzione implicita che il comportamento passato continuerà in futuro. Premesso, inoltre, che questa ipotesi implicita sia soddisfatta e cioè che il comportamento passato continui nel futuro, una regola di associazione può, ad esempio, contribuire ad un aumento del valore del "cesto della spesa" medio. Infatti scoprire che due prodotti sono fortemente associati può spingere gli addetti alla gestione del centro commerciale a posizionare i due prodotti lontani fra di loro allo scopo di "rendere la vita più difficile al cliente", massimizzando in tal modo la probabilità di acquisto di prodotti, che nel caso precedente non si sarebbero potuti trovare nel percorso teorico che un cliente segue nel centro commerciale.