

## Il Clustering

*In questo capitolo illustreremo quel task di Data Mining noto come clustering. Il capitolo si apre con una introduzione al clustering; successivamente vengono esaminati i tipi di dati nel clustering; dopo di ciò verranno prese in considerazione alcune possibili tassonomie dei principali metodi di clustering. Il capitolo prosegue con la trattazione dei principali metodi di clustering, in particolare i metodi di partizionamento, i metodi di clustering gerarchico, i metodi di clustering basati sulla densità, i metodi di clustering basati sulla griglia e, infine, i metodi di clustering basati sul modello. Il capitolo si chiude con la trattazione del clustering di dati altamente dimensionali, del clustering basato sui vincoli e dell'analisi degli outlier.*

### 13.1 Introduzione al Clustering

Si immagini di avere *un insieme di oggetti* da analizzare in cui, a differenza della classificazione, *non è nota l'etichetta di classe* di ciascun oggetto.

Il processo di raggruppamento di un insieme di oggetti fisici o astratti in classi di oggetti simili è denominato *clustering*.

*Un cluster* è una collezione di oggetti che sono simili l'un l'altro e sono dissimili dagli oggetti di altri cluster. Un cluster di oggetti può essere trattato collettivamente come un gruppo in molte applicazioni.

Sebbene la classificazione è un mezzo efficace per distinguere gruppi o classi di oggetti *essa richiede una costruzione e un'etichettatura del training set* che risultano spesso costose.

Spesso può essere *desiderabile procedere in senso inverso* partizionando prima i dati in gruppi sulla base della loro similarità (ovvero, utilizzando il clustering) e, successivamente, assegnando le etichette al numero relativamente piccolo di gruppi così ottenuto.

*Ulteriori vantaggi di tale processo di clustering* riguardano il fatto che esso è adattabile ai cambiamenti e consente di scegliere quali sono le caratteristiche di interesse per distinguere i vari gruppi.

L'analisi dei cluster è *un'attività umana importante*. Persino nell'infanzia uno impara a distinguere tra gatti e cani o tra animali e piante migliorando continuamente i suoi schemi di classificazione subconsci. L'analisi dei cluster è stata largamente utilizzata in numerose applicazioni, comprese il riconoscimento dei pattern, l'analisi dei dati, il trattamento delle immagini e la ricerca di mercato.

*In economia*, il clustering può aiutare gli operatori a scoprire gruppi distinti di clienti caratterizzandoli in base ai loro acquisti.

*In biologia*, esso può essere utilizzato per derivare le tassonomie delle piante e degli animali, per categorizzare i geni con funzionalità simili e per esaminare varie caratteristiche delle popolazioni.

Il cluster può anche aiutare *nell'identificazione di aree terrestri con uso simile* in un database spaziale, *nell'identificazione di gruppi di assicuratori di macchine* che hanno la stessa politica come pure *nell'identificazione di gruppi di case in una città* a seconda del tipo di casa, del suo valore e della sua locazione geografica. Esso può anche aiutare a classificare i documenti sul Web.

Il clustering può anche essere *utilizzato per la ricerca degli outlier* (ovvero di valori molto lontani da ciascun cluster); in alcune applicazioni gli outlier sono ancora più importanti dei valori comuni. Si pensi, ad esempio, alla ricerca delle frodi nelle carte di credito oppure al monitoring delle attività

criminali nel commercio elettronico. Per esempio, casi eccezionali nelle transazioni delle carte di credito, ad esempio acquisti molto costosi e frequenti, possono essere indici di attività fraudolenta.

Come *funzionalità del data mining*, il clustering può essere utilizzato per esaminare le distribuzioni dei dati, per osservare le caratteristiche di ciascuna distribuzione e per focalizzarsi su quelle di maggiore interesse. Alternativamente, esso può essere utilizzato come un passo di preprocessing per altri algoritmi, quali la classificazione e la caratterizzazione, che operano sui cluster individuati.

Il clustering dei dati è una *disciplina scientifica giovane* che sta attraversando un enorme sviluppo. In esso convergono aree di ricerca quali il data mining, la statistica, il machine learning, la tecnologia dei database spaziali, la biologia e il marketing.

Il clustering è un *esempio di learning non supervisionato*; a differenza della classificazione (che è una forma di learning supervisionato), il clustering non si basa su classi predefinite e su campioni di training etichettati. Per tale ragione, il clustering è una forma di learning per osservazione piuttosto che di learning per esempio.

Il clustering è un campo di ricerca affascinante; *le potenziali applicazioni* sono molte ma il suo utilizzo richiede notevoli risorse. I seguenti sono requisiti tipici del clustering nel data mining.

1. *Scalabilità.* Molti algoritmi di clustering lavorano bene su piccoli insiemi di dati; tuttavia, un grande database può contenere milioni di oggetti. Si rendono, quindi, necessari algoritmi di clustering altamente scalabili.
2. *Capacità di trattare diversi tipi di attributi.* Molti algoritmi sono progettati per clusterizzare dati numerici. Tuttavia, le applicazioni possono richiedere il clustering di altri tipi di dati, quali dati binari, categorici e ordinali, oppure una combinazione di questi tipi.
3. *Individuazione di cluster con forme arbitrarie.* Molti algoritmi di clustering determinano i cluster basandosi sulle misure di distanza Euclidea o di Manhattan. Algoritmi basati su tali misure di distanza tendono a costruire cluster sferici con dimensioni e densità simili. Tuttavia, un cluster potrebbe avere una forma qualunque. È importante sviluppare algoritmi che possano individuare cluster di forma arbitraria.
4. *Richieste minime sulla conoscenza del dominio per determinare i parametri di input.* Molti algoritmi di clustering richiedono agli utenti di inserire alcuni parametri nell'analisi dei cluster (quali il numero di cluster desiderati). I risultati del clustering sono spesso piuttosto sensibili ai parametri di input. I parametri sono spesso difficili da determinare, specialmente per gli insiemi dei dati contenenti oggetti con molte dimensioni. Ciò non solo vincola gli utenti ma rende anche la qualità del clustering difficile da controllare.
5. *Capacità nel trattare dati rumorosi.* Gran parte dei database del mondo reale contengono dati mancanti, sconosciuti o errati. Alcuni algoritmi di clustering sono sensibili a questi dati e possono portare a cluster di scarsa qualità.
6. *Clustering incrementale.* Alcuni algoritmi di clustering non possono incorporare nuovi dati nei cluster già esistenti; pertanto, in presenza di nuovi dati, devono rifare daccapo tutta l'attività di clustering. Sarebbe importante sviluppare algoritmi capaci di aggiornare i clustering in modo incrementale.
7. *Insensibilità all'ordinamento dei record di input.* Alcuni algoritmi di clustering sono sensibili all'ordine dei dati di input; pertanto, lo stesso insieme dei dati, quando è presentato in un ordine diverso, può generare cluster drammaticamente differenti. È importante sviluppare algoritmi che siano insensibili all'ordinamento dell'input.
8. *Alta dimensionalità.* Un database o un Data Warehouse possono contenere diverse dimensioni o attributi. Molti algoritmi di clustering sono capaci di gestire dati, che coinvolgono solo due o tre dimensioni. Gli occhi umani sono capaci di giudicare la qualità del clustering fino a tre dimensioni. È affascinante clusterizzare oggetti in spazi con molte dimensioni, specialmente considerando che i dati nello spazio con molte dimensioni possono essere molto sparsi ed altamente asimmetrici.
9. *Clustering basato sui vincoli.* Le applicazioni del mondo reale possono dover operare tenendo conto di vari tipi di vincoli.
10. *Interpretabilità e usabilità.* Gli utenti si aspettano che i risultati del clustering siano interpretabili, comprensibili e usabili.

## 13.2 Tipi di dati nel Clustering

Si supponga che un *insieme di dati da clusterizzare* contenga  $n$  oggetti che possono rappresentare persone, cose, documenti, nazioni, ecc.

Gli algoritmi di clustering tipicamente operano su *una delle seguenti strutture dati*:

- *Una matrice di dati* (o struttura object-by-variable): questa rappresenta  $n$  oggetti, come ad esempio persone, con  $p$  variabili (chiamate anche misure o attributi), quali l'età, l'altezza, il peso, la razza, e così via. La struttura è nella forma di una tabella relazionale, o matrice  $n \times p$  ( $n$  oggetti per  $p$  variabili):

$$\begin{pmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{pmatrix}$$

- *Una matrice di dissimilarità* (o struttura object-by-object): questa memorizza il grado di dissimilarità di ciascuna coppia degli oggetti coinvolti. Essa è spesso rappresentata da una tabella  $n \times n$ , come di seguito specificato:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ d(2,1) & 0 & 0 & 0 & 0 \\ d(3,1) & d(3,2) & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{pmatrix}$$

dove  $d(i, j)$  è la differenza o dissimilarità misurata tra gli oggetti  $i$  e  $j$ . Si noti che  $d(i, j) = d(j, i)$  e che  $d(i, i) = 0$ .

La matrice dei dati è spesso denominata *matrice a due modi* mentre la matrice di dissimilarità è denominata *matrice ad un modo*, dal momento che le righe e le colonne delle prima rappresentano diverse entità, mentre quelle dell'ultima rappresentano la stessa entità.

Molti algoritmi di clustering operano sulla matrice delle dissimilarità. Se i dati sono presentati sotto forma di una matrice di dati è necessario trasformare quest'ultima in una matrice di dissimilarità prima di applicare tali algoritmi.

Il calcolo di  $d(i, j)$  può avvenire in svariati modi differenti; tali modi dipendono tanto dal tipo di variabili coinvolte quanto dal contesto di riferimento. Per ragioni di tempo non possiamo approfondire tale argomento per il quale, però, esiste una vasta gamma di articoli e testi di riferimento.

## 13.3 Una categorizzazione dei principali metodi di Clustering

Esiste un gran numero di algoritmi di clustering. La scelta dell'algoritmo da utilizzare in un dato contesto dipende dal tipo di dati disponibili, dal particolare scopo e dall'applicazione.

Se l'analisi dei cluster viene utilizzata come un tool descrittivo o esplorativo, è possibile *provare diversi algoritmi sugli stessi dati* per vedere cosa ciascuno di essi riesce a fare.

In generale, i principali metodi di clustering possono essere classificati come di seguito specificato.

- *Metodi di partizionamento*. Dato un database di  $n$  oggetti o tuple di dati, un metodo di partizionamento costruisce  $k$  partizioni dei dati, dove ciascuna partizione rappresenta un cluster, e  $k \leq n$ . In altre parole, l'algoritmo classifica i dati in  $k$  gruppi che, nel loro insieme, soddisfano i seguenti requisiti: (1) ciascun gruppo deve contenere almeno un oggetto, e (2) ciascun oggetto deve appartenere esattamente ad un gruppo.

Dato  $k$ , il numero di partizioni da costruire, un metodo di partizionamento crea innanzitutto un *partizionamento iniziale*. Su tale partizionamento viene, successivamente, applicata una tecnica di rilocalizzazione iterativa che tenta di migliorarlo spostando oggetti da un gruppo ad un altro.

Il criterio generale di un buon partizionamento è che gli oggetti nello stesso cluster devono essere "vicini", o correlati, l'un l'altro, mentre gli oggetti di cluster differenti sono molto distanti tra loro. Esistono vari altri criteri per giudicare la qualità delle partizioni.

Ottenere l'ottimalità globale nel clustering basato sul partizionamento richiederebbe l'enumerazione esaustiva di tutte le possibili partizioni. Per tale ragione, molto spesso, si adottano delle tecniche euristiche; tra esse citiamo: (1) l'algoritmo *k-means*, dove ciascun cluster è rappresentato dal valore medio degli oggetti nel cluster; (2) l'algoritmo *k-medoids*, dove ciascun cluster è rappresentato da uno degli oggetti localizzati vicino al centro del cluster.

Questi metodi euristici operano bene per trovare cluster a forma sferica in database piccoli o medi. Per trovare cluster con forme non sferiche e per clusterizzare insiemi di dati molto grandi è necessario estendere i metodi basati sul partizionamento.

- *Un metodo gerarchico* crea una decomposizione gerarchica di un dato insieme di oggetti. I metodi gerarchici possono essere classificati in agglomerativi o divisivi, basandosi su come viene effettuata la decomposizione gerarchica.

Nell'approccio agglomerativo, detto anche approccio "bottom up", ciascun oggetto forma inizialmente un gruppo separato. Successivamente gli oggetti o i gruppi vicini l'un l'altro vengono fusi fino a quando non si ottiene un unico gruppo (il livello più in alto della gerarchia), oppure fino a quando non si verifica una condizione di terminazione.

L'approccio divisivo, detto anche "approccio top-down", inizia con tutti gli oggetti posti nello stesso cluster. Durante ciascuna iterazione successiva, un cluster viene suddiviso in cluster più piccoli, fino a quando ciascun oggetto si trova in un cluster differente o fino a quando non si verifica una determinata condizione di terminazione.

I metodi gerarchici soffrono del fatto che, una volta che un passo (fusione o suddivisione) è stato effettuato, esso non può più essere disfatto. Tale rigidità è utile in quanto porta a dei costi computazionali minori, non consentendo un numero combinatorio di scelte differenti. Tuttavia, un grande problema di tali tecniche è che essi non possono correggere decisioni errate.

Vi sono due approcci per migliorare la qualità del clustering gerarchico: (1) effettuare un'analisi attenta dei collegamenti tra oggetti durante ciascun partizionamento gerarchico (come avviene in CURE e Chamaleon), oppure (2) integrare l'agglomerazione gerarchica e altri approcci utilizzando dapprima un algoritmo di agglomerazione gerarchica per raggruppare gli oggetti in microcluster e, successivamente, raggruppando i microcluster in cluster più grandi utilizzando la rilocalizzazione iterativa, come in BIRCH.

- *Metodi basati sulla densità.* Molti metodi di partizionamento clusterizzano gli oggetti basandosi sulla loro distanza. Tali metodi possono trovare solo cluster a forma sferica e incontrare difficoltà nell'individuare cluster di forma arbitraria.

Sono stati sviluppati altri metodi di clustering basandosi sulla nozione di densità. La loro idea generale è quella di far crescere un dato cluster fino a quando la densità (numero di oggetti o punti di dati) in un vicinato non eccede una determinata soglia; in altre parole, è necessario garantire che, per ciascun punto all'interno di un dato cluster, il vicinato di un determinato raggio debba contenere almeno un numero minimo di punti. Tale metodo può essere usato per filtrare rumore e scoprire cluster di forma arbitraria.

DBSCAN e la sua estensione OPTICS sono tipici metodi basati sulla densità che costruiscono i cluster in base ad un'analisi della connettività basata sulla densità. DENCLUE è un metodo che clusterizza gli oggetti basandosi sull'analisi delle distribuzioni dei valori delle funzioni di densità.

- *Metodi basati sulla griglia.* I metodi basati sulla griglia quantizzano lo spazio degli oggetti in un numero finito di celle che formano una struttura a griglia. Tutte le operazioni di clustering vengono, quindi, eseguite sullo spazio quantizzato.

Il principale vantaggio di tale approccio sta nella sua velocità di calcolo che è tipicamente indipendente dal numero degli oggetti dipendendo soltanto dal numero di celle in ciascuna dimensione dello spazio quantizzato.

STING è un tipico esempio di un metodo basato sulla griglia. CLIQUE e Wavecluster sono due algoritmi di clustering che sono basati sia sulla griglia che sulla densità.

- *Metodi basati sul modello.* I metodi basati sul modello ipotizzano un modello per ciascuno dei cluster e trovano la migliore disposizione dei dati rispetto al determinato modello.

Un algoritmo basato sul modello può localizzare i cluster costruendo una funzione di densità che riflette la distribuzione spaziale dei punti associati ai dati. Esso consente anche di determinare automaticamente il numero di cluster basandosi su statistiche standard, tenendo in considerazione il rumore e ottenendo, pertanto, metodi di clustering robusti.

EM è un algoritmo che effettua l'analisi expectation-maximization basata su modellazioni statistiche. COBWEB è un algoritmo di learning concettuale che effettua un'analisi delle probabilità

e prende i concetti come modelli per i cluster. *SOM* è un algoritmo basato sulle reti neurali che clusterizza mappando dati altamente dimensionali in una mappa di caratteristiche bidimensionali o tridimensionali.

*Alcuni algoritmi di clustering integrano le idee di vari metodi di clustering*, così che è spesso difficile classificare un dato algoritmo come appartenente unicamente ad uno dei metodi di clustering.

Inoltre, *alcune applicazioni possono avere dei criteri di clustering* che richiedono l'integrazione di diverse tecniche di clustering.

## 13.4 I metodi di partizionamento

Dato un database di  $n$  oggetti, e  $k$ , il numero di cluster da costruire, *un algoritmo di partizionamento organizza gli oggetti in  $k$  partizioni ( $k \leq n$ )*, dove ciascuna partizione rappresenta un cluster.

I cluster sono costruiti con il fine di *ottimizzare un criterio di partizionamento* oggettivo, spesso denominato funzione di similarità, come la distanza, in modo tale che gli oggetti all'interno di un cluster siano "simili" mentre gli oggetti di cluster differenti siano "dissimili".

### 13.4.1 Il metodo di partizionamento k-Means

*L'algoritmo k-means riceve in input un parametro  $k$  e partiziona un insieme di  $n$  oggetti in  $k$  cluster in modo tale che la similarità intra-cluster risultante sia alta mentre la similarità inter-cluster sia bassa. La similarità dei cluster è misurata rispetto al valore medio degli oggetti in un cluster; tale valore può essere visto come il centro di gravità del cluster.*

L'algoritmo k-means procede nel seguente modo. Innanzitutto, *seleziona randomicamente  $k$  degli oggetti*, ciascuno dei quali rappresenta inizialmente la media o il centro di un cluster. Ciascuno degli oggetti rimanenti viene associato al cluster più simile basandosi sulla distanza tra l'oggetto e la media del cluster.

Esso, quindi, *calcola la nuova media per ciascun cluster*.

*Tale processo viene iterato* fino a quando non si raggiunge la convergenza di una determinata funzione criterio. Tipicamente viene utilizzato come criterio l'errore quadratico, definito come:

$$E = \sum_{i=1..k} \sum_{p \in C_i} |p - m_i|^2$$

dove  $E$  è la somma dell'errore quadratico per tutti gli oggetti del database,  $p$  è il punto nello spazio che rappresenta il dato oggetto ed  $m_i$  è la media del cluster  $C_i$  (sia  $p$  che  $m_i$  sono multidimensionali).

Questo criterio cerca di rendere *i  $k$  cluster risultanti quanto più compatti e separati possibili*.

L'algoritmo tenta di determinare  *$k$  partizioni che minimizzano la funzione errore quadratico*. Esso si comporta bene quando i cluster sono compatti e piuttosto ben separati uno dall'altro.

Il metodo è relativamente scalabile ed efficiente nel processare grandi insiemi di dati perchè *la complessità computazionale dell'algoritmo è  $O(nkt)$* , dove  $n$  è il numero totale di oggetti,  $k$  è il numero di cluster e  $t$  è il numero di iterazioni. Normalmente,  $k \ll n$  e  $t \ll n$ .

*Il metodo spesso termina in un ottimo locale.*

Il metodo, tuttavia, *può essere applicato soltanto quando è definita la media di un cluster*. Questo può non accadere in molte applicazioni, ad esempio quando sono coinvolti dati con attributi categorici.

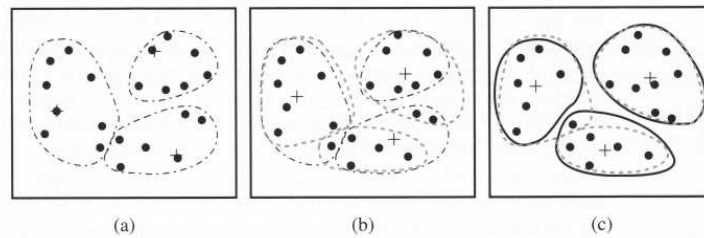
*La necessità per gli utenti di specificare all'inizio  $k$* , ovvero il numero dei cluster, può essere vista come uno svantaggio.

*k-means non è adatto per scoprire cluster con forme non convesse o cluster di dimensioni molto differenti.*

Inoltre, *è sensibile al rumore e agli outlier* dal momento che un piccolo numero di questi dati possono influenzare sostanzialmente il valore medio.

*Esempio 13.1.* Si supponga di avere un insieme di oggetti localizzati nello spazio secondo quanto mostrato in Figura 13.1. Si supponga che  $k = 3$ , ovvero che l'utente vorrebbe suddividere gli oggetti in 3 cluster.

Seguendo l'algoritmo k-means *scegliamo arbitrariamente 3 oggetti* come i 3 centri iniziali del cluster, dove i centri del cluster sono marcati con un "+". Ciascun oggetto viene distribuito su un cluster



**Figura 13.1.** Clustering di un insieme di oggetti basato sul metodo k-Means

basandosi sul centro del cluster a cui è più vicino. Tale distribuzione forma le silhouette separate da curve tratteggiate mostrate nella Figura 13.1(a).

*Al termine di tale raggruppamento è necessario ricalcolare i centri dei cluster* basandosi sugli oggetti che correntemente fanno parte del cluster. Usando i nuovi centri del cluster, gli oggetti vengono ridistribuiti tra i vari cluster basandosi sulle loro distanze dai nuovi centri. Tale ridistribuzione forma le silhouette circondate da curve tratteggiate, mostrate nella Figura 13.1(b).

*Il processo viene, quindi, iterato di nuovo* e si ottengono le silhouette mostrate nella Figura 13.1(c). Dopo di ciò non si ha più alcuna ridistribuzione degli oggetti e, pertanto, il processo termina. I cluster mostrati nella Figura 13.1(c) rappresentano, di conseguenza, il risultato finale del processo di clustering.

Vi sono *alcune varianti del metodo k-means*. Esse possono differire nella selezione delle  $k$  medie iniziali, nel calcolo della similarità e nelle strategie per il calcolo delle medie del cluster.

Una strategia interessante, che spesso restituisce buoni risultati, consiste *nell'applicare prima un algoritmo di agglomerazione gerarchica* per determinare il numero di cluster e per trovare una classificazione iniziale, e nell'utilizzare successivamente la rilocalizzazione iterativa per migliorare la classificazione.

Un'altra variante del k-means è il metodo *k-modes* che estende il paradigma k-means per clusterizzare i dati categorici, rimpiazzando le medie dei cluster con le mode, adottando nuove misure di dissimilarità per trattare oggetti categorici e utilizzando metodi basati sulla frequenza per aggiornare le mode.

I metodi k-means e k-modes possono essere integrati per clusterizzare i dati con valori misti numerici e categorici, ottenendo il metodo *k-prototypes*.

*Come si può rendere l'algoritmo k-means più scalabile?* Un approccio proposto che opera in tal senso è basato sull'idea di identificare tre tipologie di regioni nei dati: regioni che si possono comprimere, regioni che devono essere mantenute in memoria centrale e regioni che si possono scaricare.

*Un oggetto può essere scaricato* se la sua appartenenza ad un cluster è accertata.

*Un oggetto è comprimibile* se non si può scartare ma se appartiene ad un sotto-cluster ridotto. Una struttura dati conosciuta come clustering feature viene utilizzata per riassumere gli oggetti che sono stati scartati o compressi.

Se un oggetto non è né scartato né compresso, allora dovrebbe essere *mantenuto nella memoria principale*.

*Per ottenere la scalabilità*, l'algoritmo di clustering iterativo include solo le clustering feature degli oggetti comprimibili e degli oggetti che devono essere mantenuti nella memoria principale; in questo modo si trasforma un algoritmo basato sulla memoria secondaria in un algoritmo basato sulla memoria principale.

Un approccio alternativo per rendere scalabile l'algoritmo k-means utilizza l'idea dei microcluster, ovvero raggruppa in microcluster gli oggetti vicini e, successivamente, applica k-means sui microcluster.

□

#### 13.4.2 Il metodo di partizionamento k-Medoids

*L'algoritmo k-means è sensibile agli outlier*, dal momento che un oggetto con un valore estremamente grande può distorcere sensibilmente la distribuzione dei dati.



Per diminuire la sensitività, invece di prendere come punto di riferimento il valore medio degli oggetti in un cluster, può essere utilizzato il *medoid*, ovvero l'oggetto localizzato più centralmente in un cluster.

In questo modo è possibile comunque eseguire il metodo di partizionamento cercando di minimizzare la somma delle dissimilarità tra ciascun oggetto e il suo punto di riferimento corrispondente.

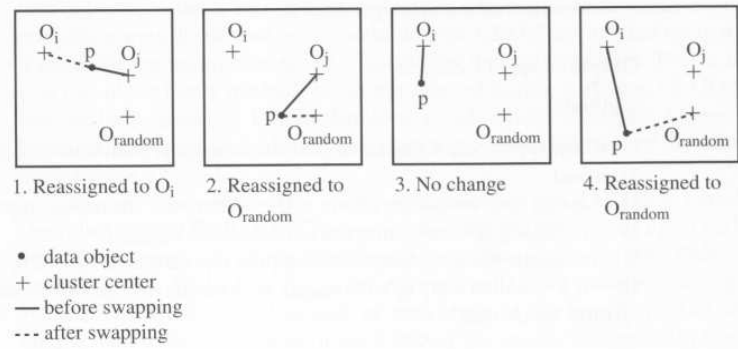
In altre parole, viene utilizzato il criterio di errore assoluto definito come:

$$E = \sum_{i=1}^k \sum_{p \in C_j} |p - o_j|$$

La strategia di base dell'algoritmo k-medoids è quella di raggruppare  $n$  oggetti in  $k$  cluster. Innanzitutto esso trova arbitrariamente un oggetto rappresentativo per ciascun cluster. Ciascun oggetto rimanente viene inserito nel cluster associato al medoid più simile.

La strategia sostituisce iterativamente medoidi con non-medoidi e termina quando la qualità del clustering risultante non può più essere migliorata. Questa qualità viene stimata utilizzando una funzione di costo che misura la dissimilarità media tra un oggetto e il medoide del suo cluster.

Per determinare se un oggetto non medoide,  $o_{random}$ , è un buon sostituto di un medoide corrente,  $o_j$ , è necessario esaminare i seguenti quattro casi per ciascuno degli oggetti non-medoidi  $p$  (Figura 13.2):



**Figura 13.2.** I quattro possibili casi della funzione di costo per il metodo k-Medoid

- Caso 1:  $p$  appartiene correntemente al medoid  $o_j$ . Se  $o_j$  è sostituito da  $o_{random}$  come medoid e  $p$  è più vicino ad uno degli  $o_i$ ,  $i \neq j$ , allora  $p$  viene riassegnato ad  $o_i$ .
- Caso 2:  $p$  appartiene correntemente al medoid  $o_j$ . Se  $o_j$  è sostituito da  $o_{random}$  come medoid e  $p$  è più vicino ad  $o_{random}$ , allora  $p$  viene riassegnato ad  $o_{random}$ .
- Caso 3:  $p$  appartiene correntemente al medoid  $o_i$ ,  $i \neq j$ . Se  $o_j$  è sostituito da  $o_{random}$  come medoid e  $p$  è ancora più vicino ad  $o_i$ , allora l'assegnamento non comporta cambiamenti per  $p$ .
- Caso 4:  $p$  appartiene correntemente al medoid  $o_i$ ,  $i \neq j$ . Se  $o_j$  è sostituito da  $o_{random}$  come medoid e  $p$  è più vicino ad  $o_{random}$ , allora  $p$  viene riassegnato ad  $o_{random}$ .

Ogni volta che viene effettuato un riassegnamento, si presenta una differenza nella funzione di costo. Tale differenza può essere positiva o negativa.

Definiamo *costo totale di swapping* la somma di tutte le differenze nelle funzioni di costo che si ottengono in seguito al riassegnamento degli oggetti non medoid.

Se il costo totale di swapping è negativo allora  $o_j$  viene sostituito con  $o_{random}$  dal momento che l'errore quadratico complessivo sarebbe ridotto. Se il costo totale di swapping è positivo, il medoid corrente  $o_j$  è considerato accettabile e non viene sostituito.

In un algoritmo di tipo k-medoid come PAM il costo di una singola iterazione è  $O(k(n-k)^2)$ . Per grandi valori di  $n$  e  $k$  tale calcolo diventa molto costoso.

Il metodo k-medoids è più robusto di k-means in presenza di rumore e di outlier perché un medoide è meno influenzato dagli outlier o da altri valori estremi rispetto ad una media.

Tuttavia, l'elaborazione di k-medoids è più costosa rispetto al metodo k-means.

Entrambi i metodi richiedono all'utente di specificare  $k$ , il numero di cluster.

### 13.4.3 Metodi di partizionamento in grossi database

Un tipico algoritmo di partizionamento k-medoids, come PAM, lavora efficientemente per piccoli insiemi di dati *ma non è molto scalabile*.

*Per trattare insiemi di dati più grandi* è possibile usare un metodo basato sul campionamento, denominato CLARA (Clustering LARge Applications).

*L'idea dietro CLARA è la seguente:* invece di prendere in considerazione l'intero insieme dei dati, viene scelta una piccola porzione dei dati effettivi supponendo che essa sia rappresentativa di tutti i dati. I medoidi vengono, quindi, scelti da questo campione usando PAM. Se i campioni vengono selezionati in maniera piuttosto randomica, dovrebbero rappresentare abbastanza fedelmente l'insieme dei dati originario e i medoidi rappresentativi individuati dovrebbero essere simili a quelli che si sarebbero costruiti utilizzando l'intero insieme dei dati.

CLARA sceglie più campioni dall'insieme dei dati, applica PAM su ogni campione e restituisce in output il suo clustering migliore. Chiaramente, CLARA può trattare insiemi di dati più grandi rispetto a PAM. *La complessità di ciascuna iterazione* diviene ora  $O(ks^2 + k(n - k))$ , dove  $s$  è la dimensione del campione,  $k$  è il numero di cluster ed  $n$  è il numero totale di oggetti.

*L'efficacia di CLARA dipende dalla dimensione del campione.* Si noti che PAM cerca i migliori  $k$  medoid tra un dato insieme di dati mentre CLARA cerca i migliori  $k$  medoid tra il campione selezionato dei dati. CLARA non può trovare il miglior clustering se ciascun medoid del campione non è tra i migliori  $k$  medoid. Per esempio, se un oggetto  $o_i$  è uno dei migliori  $k$  medoidi ma non è selezionato durante il campionamento, CLARA non troverà mai il miglior clustering. Esso, pertanto, è disposto ad accettare una diminuzione della qualità dei risultati finali a vantaggio, però, dell'efficienza.

Per migliorare la qualità e la scalabilità di CLARA è stato proposto un algoritmo di tipo k-medoids denominato CLARANS (Clustering Large Applications based upon RANdomized Search).

Mentre CLARA ha un campione fisso per ciascuno stadio della ricerca, CLARANS *costruisce un campione con alcune randomicità* per ciascun passo della ricerca.

Concettualmente, *il processo di clustering può essere visto come una ricerca in un grafo* dove ciascun nodo è una soluzione potenziale (un insieme di  $k$  medoidi).

*Due nodi sono vicini* (ovvero, connessi da un arco nel grafo) se i loro insiemi differiscono soltanto per un oggetto.

*A ciascun nodo può essere assegnato un costo* definito dalla dissimilarità totale tra ciascun oggetto e il medoid del suo cluster.

*A ciascun passo viene applicato l'algoritmo PAM* per esaminare tutti i vicini del nodo corrente nella ricerca della soluzione con costo minimo. Il nodo corrente viene sostituito dal vicino che presenta la massima discesa del costo. Mentre CLARA considera un campione di nodi all'inizio di una ricerca, CLARANS *considera dinamicamente* un campione randomico di vicini ad ogni passo della ricerca.

*Il numero di vicini* da campionare randomicamente è *ristretto da un parametro* selezionato dall'utente.

In questo modo CLARANS *non confina la ricerca* ad un'area localizzata.

*Se viene trovato un vicino migliore* (ovvero un vicino che ha un errore minore), CLARANS si sposta al nodo del vicino e il processo ricomincia; in caso contrario il cluster corrente produce un minimo locale.

*Se viene trovato un minimo locale*, CLARANS ricomincia con nuovi nodi selezionati randomicamente, sempre al fine di cercare un nuovo minimo locale.

*Una volta che un numero di minimi locali specificato dall'utente* è stato trovato, CLARANS restituisce il migliore tra i minimi locali.

È stato dimostrato sperimentalmente che CLARANS è *più efficace di PAM e CLARA*.

CLARANS *consente di cercare gli outlier*.

*La complessità computazionale di CLARANS*, tuttavia, è circa  $O(n^2)$  dove  $n$  è il numero di oggetti.

*Inoltre, la qualità dei risultati è dipendente* dal metodo di campionamento utilizzato.

La sua capacità di *gestire oggetti che risiedono su disco può essere migliorata ulteriormente* utilizzando tecniche che esplorano strutture dati spaziali, quali gli R\*-Tree.

## 13.5 I metodi gerarchici

*Un metodo di clustering gerarchico* lavora raggruppando gli oggetti in alberi di cluster.



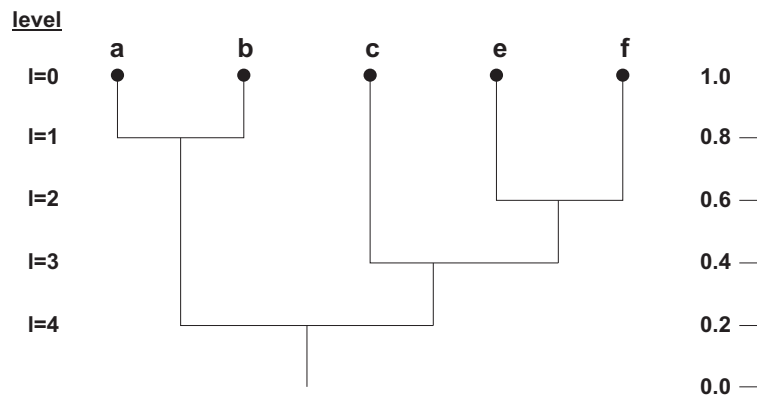
I metodi di clustering gerarchici possono essere ulteriormente classificati in:

- *Metodi di clustering gerarchico agglomerativi.* Questa strategia bottom-up parte inserendo ciascun oggetto nel proprio cluster e, successivamente, fondendo questi cluster atomici in cluster sempre più larghi, fino a quando tutti gli oggetti si trovano in un singolo cluster oppure fino a quando non vengono soddisfatte determinate condizioni di terminazione. I vari metodi di clustering appartenenti a questa categoria differiscono soltanto nella definizione della similarità intercluster.
- *Metodi di clustering gerarchico divisivi.* Questa strategia top-down opera in modo inverso rispetto ai metodi gerarchici agglomerativi. Inizialmente pone tutti gli oggetti in un cluster. Successivamente divide il cluster in porzioni sempre più piccole, fino a quando ciascun oggetto forma un cluster per conto proprio oppure fino a quando non vengono soddisfatte determinate condizioni di terminazione, legate al numero desiderato di cluster o alla distanza tra cluster.

Nei metodi di clustering gerarchici l'utente può specificare *il numero di cluster come condizione di terminazione*.

Una struttura ad albero, denominata *dendrogramma*, viene comunemente utilizzata per rappresentare il clustering gerarchico. Esso mostra come gli oggetti vengono raggruppati insieme passo dopo passo.

La Figura 13.3 mostra un esempio di dendrogramma. Al livello 0 vengono mostrati cinque cluster costituiti da un singolo oggetto. Al livello 1 gli oggetti *a* e *b* vengono raggruppati insieme per formare il primo cluster; essi rimarranno insieme per tutti i livelli successivi.



**Figura 13.3.** Un dendrogramma per il clustering gerarchico degli oggetti {a, b, c, d, e}

È possibile anche utilizzare un asse verticale per mostrare la scala di similarità tra cluster.

Quattro misure largamente usate per la distanza tra cluster; nelle formule  $|p - p'|$  è la distanza tra due oggetti o punti  $p$  e  $p'$ ,  $m_i$  è la media per il cluster  $C_i$  ed  $n_i$  è il numero di oggetti in  $C_i$ :

- *Minimum Distance:*  $d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'|$
- *Maximum Distance:*  $d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$
- *Mean Distance:*  $d_{\text{mean}}(C_i, C_j) = |m_i - m_j|$
- *Average Distance:*  $d_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|$

Il metodo di *clustering gerarchico*, sebbene semplice, incontra spesso *difficoltà relative alla selezione dei punti di merge o di split*. Tale decisione è critica perché una volta che un gruppo di oggetti viene fuso o suddiviso, il processo al passo successivo opererà sui nuovi cluster.

Esso non annulla mai quello che è stato fatto né effettua uno scambio di oggetti tra cluster. Pertanto, le decisioni di merge o di split, se vengono prese in modo sbagliato in qualche passo, possono portare a cluster di bassa qualità.

Inoltre il metodo non è molto scalabile dal momento che ciascuna decisione sul merge o sullo split richiede l'esame e la valutazione di un buon numero di oggetti o di cluster.

Una direzione promettente per migliorare la qualità del clustering dei metodi gerarchici è di integrare il clustering gerarchico con altre tecniche di clustering. Questo è quello che fanno alcuni metodi molto noti quali BIRCH, ROCK e Chamaleon.

### 13.5.1 BIRCH: Balanced Iterative Reducing and Clustering Using Hierarchies

BIRCH è stato progettato per *clusterizzare una grande quantità di dati* numerici integrando il clustering gerarchico (utilizzato nello stadio iniziale di microclustering) con altri metodi di clustering quali il partizionamento iterativo (utilizzato nello stadio di macroclustering successivo).

In questo modo è in grado di superare *le due principali difficoltà* dei metodi di clustering agglomerativo, ovvero la scalabilità e l'incapacità di annullare ciò che è stato fatto nei passi precedenti.

BIRCH è basato su *due concetti specifici*, ovvero quelli di clustering feature e clustering feature tree (CF tree), che vengono utilizzati per riassumere le rappresentazioni dei cluster. Queste strutture aiutano BIRCH ad ottenere una buona velocità ed una buona scalabilità in grossi database come pure di rendere efficace il clustering incrementale e dinamico degli oggetti coinvolti.

Esaminiamo più da vicino queste due strutture dati.

**Definizione 13.2.** Dati  $n$  oggetti di un cluster in uno spazio  $d$ -dimensionale, *definiamo il centroide*  $x_0$ , *il raggio*  $R$  e *il diametro*  $D$  del cluster nel seguente modo:

$$x_0 = \frac{\sum_{i=1}^n}{n}$$

$$R = \sqrt{\frac{\sum_{i=1}^n (x_i - x_0)^2}{n}}$$

$$D = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^n (x_i - x_j)^2}{n(n-1)}}$$

dove  $R$  è la distanza media tra gli oggetti del centroide e  $D$  è la distanza media delle coppie all'interno di un cluster. Sia  $R$  che  $D$  riflettono la ristrettezza del cluster attorno al centroide.  $\square$

**Definizione 13.3.** Una *Clustering Feature (CF)* è un vettore tridimensionale che riassume alcune informazioni sui cluster. Dati  $n$  oggetti  $d$ -dimensionali in un cluster,  $\{x_i\}$ , allora la *CF* del cluster è definita come:

$$CF = \langle n, LS, SS \rangle$$

dove  $n$  è il numero di punti nel cluster,  $LS$  è la somma lineare degli  $n$  punti (ovvero,  $\sum_{i=1}^n x_i$ ) ed  $SS$  è la somma quadratica dei punti (ovvero,  $\sum_{i=1}^n x_i^2$ ).  $\square$

Una clustering feature è, quindi, un *riassunto delle statistiche associate al sotto-cluster*: da un punto di vista statistico essa considera i momenti 0, 1 e 2 del sottocluster. Essa *registra misure cruciali* per le elaborazioni associate all'algoritmo e gestisce la memoria in modo efficiente dal momento che, invece di memorizzare gli oggetti, memorizza soltanto alcuni dati aggregati relativi ad essi.

Un *CF tree* è un albero bilanciato in altezza che memorizza le clustering feature per un clustering gerarchico. I nodi non foglia di un *CF Tree* memorizzano le somme delle *CF* dei loro figli e, pertanto, riassumono informazioni di clustering riguardo ai loro figli.

Un *CF tree* ha due parametri: il branching factor,  $B$ , e la threshold,  $T$ . Il *branching factor* specifica il massimo numero di figli per nodo non foglia. La *threshold* specifica il massimo diametro dei sottocluster memorizzati sui nodi foglia dell'albero. Questi due parametri influenzano la dimensione dell'albero risultante.

BIRCH opera in due fasi:

- *Fase 1:* Viene esaminato il database per costruire un *CF tree* iniziale su RAM; tale attività viene vista come una compressione multilivello dei dati che tenta di preservare la struttura di clustering intrinseca in essi.
- *Fase 2:* Viene applicato un algoritmo di clustering per clusterizzare i nodi foglia del *CF tree*.

Per quanto riguarda la Fase 1, il *CF* viene costruito dinamicamente mano a mano che gli oggetti vengono inseriti. Pertanto il metodo è incrementale.

*Un oggetto viene inserito nel cluster foglia più vicino.* Se, dopo tale inserimento, il diametro del cluster associato al nodo foglia è maggiore di una determinata soglia, allora il nodo foglia, ed eventualmente altri nodi, vengono suddivisi.

Dopo l'inserimento del nuovo oggetto, l'informazione su di esso viene propagata verso la radice dell'albero.

La dimensione del *CF tree* può essere modificata cambiando il valore della soglia. Se la quantità di memoria necessaria per memorizzare il *CF tree* è maggiore della dimensione della RAM, allora può essere specificato un valore di soglia più piccolo e, a partire da ciò, può essere costruito un nuovo *CF tree*. Tale processo viene effettuato costruendo un nuovo albero a partire dai nodi foglia del vecchio albero. Pertanto, esso non richiede il riesame dei vari oggetti già precedentemente analizzati. Di conseguenza, per costruire l'albero, i dati devono essere letti soltanto una volta.

Sono state proposte alcune metodologie per rendere BIRCH più robusto agli outlier e per migliorare la qualità complessiva dei *CF tree*.

Una volta che il *CF tree* è stato costruito, è possibile applicare ad esso un qualunque algoritmo di clustering, ad esempio il tipico algoritmo di partizionamento, durante la Fase 2.

BIRCH applica una *tecnica di clustering multifase*: una singola scansione dell'insieme dei dati restituisce un buon clustering di base; a questo punto possono essere utilizzate altre scansioni per migliorare ulteriormente la qualità dei risultati.

La complessità di calcolo dell'algoritmo è  $O(n)$ , dove  $n$  è il numero di oggetti da clusterizzare. Gli esperimenti hanno mostrato che l'algoritmo è linearmente scalabile rispetto al numero di oggetti e restituisce risultati di buona qualità.

Tuttavia, dal momento che ciascun nodo in un *CF tree* può memorizzare solo un numero limitato di oggetti, esso non sempre corrisponde ad un cluster reale.

Inoltre, dal momento che esso utilizza la nozione di raggio o di diametro per controllare i confini di un cluster, esso fornisce buoni risultati solo se i cluster in considerazione hanno forma sferica.

### 13.5.2 Chameleon: un algoritmo di clustering gerarchico che utilizza una modellazione dinamica

Chameleon è un algoritmo di clustering che utilizza una *modellizzazione dinamica per determinare la similarità tra coppie di cluster*.

In Chameleon, la similarità tra cluster viene stabilita basandosi su quanto sono ben connessi gli oggetti all'interno di un cluster e sulla prossimità dei cluster. In altre parole, due cluster vengono fusi se la loro interconnettività è alta e se essi, insieme, risultano vicini.

Pertanto, Chameleon non dipende da un modello statico fornito dall'utente e può adattarsi automaticamente alle caratteristiche interne dei cluster che devono essere fusi.

Il processo di fusione facilita la scoperta di cluster naturali ed omogenei e si applica a tutti i tipi di dati nella misura in cui è possibile specificare una funzione di similarità.

Chameleon è stato derivato tenendo conto dei punti deboli di due algoritmi di cluster precedenti, ovvero ROCK e CURE. ROCK e gli algoritmi correlati enfatizzano l'interconnettività tra cluster mentre ignorano le informazioni relative alla loro prossimità. CURE e gli algoritmi correlati considerano la prossimità tra cluster mentre ignorano la loro interconnettività.

La filosofia secondo cui opera Chameleon è illustrata in Figura 13.4.

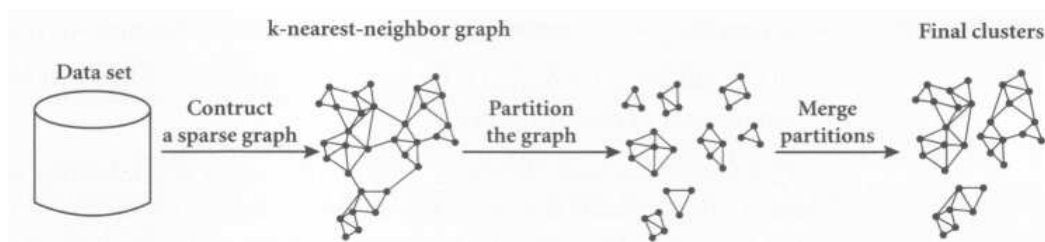


Figura 13.4. Funzionamento di Chameleon

*Esso utilizza un approccio basato sul  $k$ -nearest-neighbor graph* per costruire un grafo sparso; più specificatamente, ciascun vertice del grafo rappresenta un oggetto; esiste un arco tra due oggetti se un oggetto è tra i  $k$  oggetti più simili dell'altro. Gli archi sono pesati per riflettere la similarità tra oggetti.

Chameleon utilizza un *algoritmo di partizionamento dei grafi* per partizionare il  $k$ -nearest-neighbor graph costruito in un gran numero di sottocluster relativamente piccoli. Successivamente utilizza un algoritmo di clustering gerarchico agglomerativo che fonde ripetutamente i sottocluster basandosi sulla loro similarità.

Per determinare le coppie di sottocluster più simili, esso *prende in considerazione tanto l'interconnettività che la vicinanza tra cluster*.

Si noti che il  $k$ -nearest-neighbor graph *cattura dinamicamente il concetto di vicinato*: il raggio del vicinato di un oggetto viene determinato dalla densità della regione in cui risiede l'oggetto. In una regione densa il vicinato è stretto mentre in una regione sparsa esso è più largo.

Ciò tende a far *ottenere dei cluster più naturali rispetto ai metodi basati sulla densità* come DBSCAN che, invece, utilizzano un vicinato globale.

Inoltre, *la densità della regione viene registrata* dai pesi degli archi. Infatti, gli archi di una regione densa tendono a pesare di più rispetto a quelli di una regione sparsa.

L'algoritmo di partizionamento del grafo partiziona il  $k$ -nearest-neighbor graph in modo tale da *minimizzare il taglio degli archi*. In altre parole, un cluster  $C$  viene partizionato in sottocluster  $C_i$  e  $C_j$  in modo da minimizzare il peso degli archi che verrebbero tagliati se  $C$  fosse spezzato in  $C_i$  e  $C_j$ .

*Il taglio dell'arco viene denotato* come  $EC(C_i, C_j)$  e stabilisce l'Interconnettività Assoluta tra i cluster  $C_i$  e  $C_j$ .

Chameleon determina la similarità tra ciascuna coppia di cluster  $C_i$  e  $C_j$  in base alla loro Interconnettività Relativa  $RI(C_i, C_j)$  e alla loro Vicinanza Relativa  $RC(C_i, C_j)$ .

- *L'Interconnettività Relativa  $RI(C_i, C_j)$*  tra due cluster  $C_i$  e  $C_j$  viene definita come l'Interconnettività Assoluta tra  $C_i$  e  $C_j$  normalizzata rispetto all'Interconnettività Interna di  $C_i$  e di  $C_j$ .
- *La Vicinanza Relativa  $RC(C_i, C_j)$*  tra due cluster  $C_i$  e  $C_j$  è data dalla Vicinanza Assoluta tra  $C_i$  e  $C_j$  normalizzata rispetto alla Vicinanza Interna di  $C_i$  e di  $C_j$ .

Sono state definite opportune formule per calcolare  $RI(C_i, C_j)$  ed  $RC(C_i, C_j)$ .

*È stato dimostrato che Chamaleon* è più capace di trovare cluster di forma arbitraria e di alta qualità rispetto a molti algoritmi famosi, quali BIRCH e DBSCAN. Tuttavia, il costo di processing per dati altamente dimensionati può essere  $O(n^2)$ , dove  $n$  è il numero di oggetti da clusterizzare.

## 13.6 I metodi basati sulla densità

*Per individuare cluster di forma arbitraria* sono stati sviluppati i *metodi di clustering basati sulla densità*. Questi tipicamente considerano i cluster come regioni dense di oggetti nello spazio dei dati separate da regioni a bassa densità (che rappresentano rumore).

### 13.6.1 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) è un algoritmo di clustering basato sulla densità.

L'algoritmo *costruisce ciascun cluster mettendo insieme regioni con densità sufficientemente alta*; esso è in grado di individuare cluster di forma arbitraria e fornisce risultati interessanti anche in presenza di rumore.

In DBSCAN *un cluster viene definito come un insieme massimale di punti density-connected*.

Per capire il funzionamento del clustering basato sulla densità *sono richieste alcune nuove definizioni*. Esse verranno introdotte nel seguito.

**Definizione 13.4.** Il vicinato di raggio  $\varepsilon$  di un determinato oggetto è chiamato  $\varepsilon$ -neighborhood dell'oggetto.

□

**Definizione 13.5.** Se l' $\varepsilon$ -neighborhood di un oggetto contiene almeno un numero minimo,  $MinPts$ , di oggetti, allora l'oggetto è denominato *core object*. □

**Definizione 13.6.** Dato un insieme di oggetti  $D$  diciamo che un oggetto  $p$  è direttamente *density-reachable* dall'oggetto  $q$ , rispetto ad  $\varepsilon$  e  $MinPts$ , se  $p$  è all'interno dell' $\varepsilon$ -neighborhood di  $q$  e  $q$  è un *core object*. □

**Definizione 13.7.** Un oggetto  $p$  è *density-reachable* dall'oggetto  $q$  rispetto ad  $\varepsilon$  e  $MinPts$  in un insieme di oggetti  $D$  se vi è una catena di oggetti  $p_1, \dots, p_n$ ,  $p_1 = q$  e  $p_n = p$  tale che  $p_{i+1}$  è direttamente *density-reachable* da  $p_i$  rispetto ad  $\varepsilon$  e  $MinPts$ , per  $1 \leq i \leq n$ ,  $p_i \in D$ . □

**Definizione 13.8.** Un oggetto  $p$  è *density-connected* ad un oggetto  $q$  rispetto ad  $\varepsilon$  e  $MinPts$  in un insieme di oggetti  $D$  se esiste un oggetto  $o \in D$  tale che sia  $p$  che  $q$  sono *density-reachable* da  $o$  rispetto ad  $\varepsilon$  e  $MinPts$ . □

La *density-reachability* è la chiusura transitiva della *density reachability* diretta e questa relazione è asimmetrica. Solo i *core object* sono mutuamente *density-reachable*.

La *density-connectivity*, invece, è una relazione *simmetrica*.

Un cluster basato sulla densità è un insieme di oggetti *density-connected* che risulta essere massimale rispetto alla *density-reachability*. Se un oggetto non è contenuto in nessun cluster viene considerato rumore.

DBSCAN costruisce i cluster determinando l' $\varepsilon$ -neighborhood di ciascun punto nel database. Se l' $\varepsilon$ -neighborhood di un punto  $p$  contiene più di  $MinPts$ , viene creato un nuovo cluster con  $p$  come *core object*. Successivamente vengono individuati gli oggetti *density-reachable* dai vari *core object*; se due *core object* sono direttamente *density-reachable* i cluster corrispondenti vengono fusi.

Il processo termina quando nessun nuovo punto può essere aggiunto ad un cluster.

Se viene utilizzato un indice spaziale la complessità computazionale di DBSCAN è  $O(n \log n)$ , dove  $n$  è il numero di oggetti del database; in caso contrario tale complessità diventa  $O(n^2)$ .

Con una definizione appropriata dei parametri  $\varepsilon$  e  $MinPts$  definiti dall'utente, l'algoritmo è efficace nel trovare cluster di forma arbitraria.

## 13.7 I metodi basati sulla griglia

I metodi di clustering basati sulla griglia utilizzano una *struttura dati a griglia multirisoluzione*. Essi quantizzano lo spazio in un numero finito di celle che forma una struttura a griglia su cui vengono effettuate tutte le operazioni per il clustering.

Il principale vantaggio di questi approcci consiste nel loro tempo di elaborazione ridotto; tale tempo, infatti, è indipendente dal numero degli oggetti da clusterizzare essendo dipendente, soltanto, dal numero di celle in ciascuna dimensione dello spazio quantizzato.

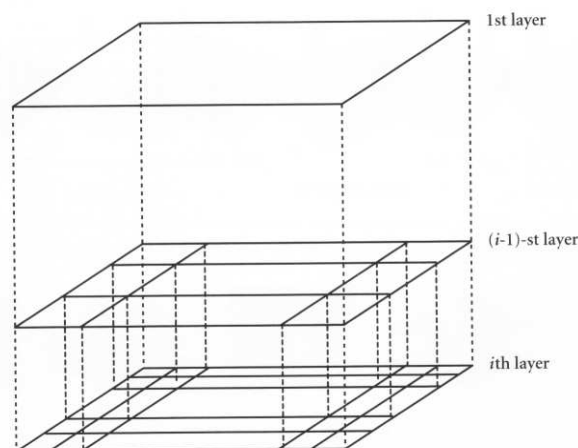
Alcuni esempi tipici di clustering basato sulla griglia includono:

- STING, che esplora le informazioni statistiche memorizzate nelle celle della griglia;
- WaveCluster, che clusterizza gli oggetti utilizzando un metodo basato sulla trasformata Wavelet;
- CLIQUE, che rappresenta un approccio basato, contemporaneamente, sulla griglia e sulla densità per il clustering di dati a grandi dimensioni.

### 13.7.1 STING: STatistical INformation Grid

STING è una tecnica di clustering multi-risoluzione basato sulla griglia in cui l'area spaziale è suddivisa in celle rettangolari. Vi sono, generalmente, diversi livelli di celle rettangolari che corrispondono a diversi livelli di risoluzione; queste celle formano una struttura gerarchica: ciascuna cella ad un livello elevato viene partizionata per formare un certo numero di celle ad un livello immediatamente inferiore.





**Figura 13.5.** Una struttura gerarchica per il clustering STING

L'algoritmo pre-calcola e memorizza alcune informazioni statistiche (quali la media, il valore massimo e il valore minimo) relative agli attributi di ciascuna cella della griglia. Tali parametri verranno utilizzati durante le attività di query processing, come sarà chiaro nel seguito.

La Figura 13.5 mostra una struttura gerarchica per il clustering basato su STING. I parametri statistici delle celle a più alto livello possono essere facilmente calcolati dai parametri delle celle a più basso livello. Alcuni parametri statistici particolarmente utilizzati sono: il conteggio, la media, la deviazione standard, il minimo, il massimo e il tipo di distribuzione seguita dagli attributi di una cella; i possibili tipi di distribuzione sono: normal, uniform, exponential oppure none (quando la distribuzione è sconosciuta).

Quando i dati vengono caricati nel database, i parametri statistici delle celle a livello più basso vengono calcolati direttamente da essi. I valori della distribuzione possono essere specificati dall'utente, se sono noti a quest'ultimo, oppure possono essere frutto di test di ipotesi, quali il test  $\chi^2$ .

Il tipo di distribuzione di una cella a più alto livello può essere determinato sulla base della maggioranza dei tipi di distribuzione delle corrispondenti celle a più basso livello. Se le distribuzioni delle celle a più basso livello sono in disaccordo tra loro e nessuna di esse prevale nettamente sulle altre, il tipo di distribuzione della cella ad alto livello è posto a none.

I parametri statistici possono essere utilizzati nel seguente modo. Innanzitutto, all'interno della struttura gerarchica, viene determinato un livello da cui far partire il processo di query answering. Questo livello tipicamente contiene un piccolo numero di celle.

Per ciascuna cella del livello corrente, viene calcolato l'intervallo di confidenza (o il range stimato di probabilità) che riflette la rilevanza della cella per quella query. Le celle irrilevanti non vengono prese in considerazione per le attività successive.

L'elaborazione del livello immediatamente inferiore esamina solo le celle rimaste al livello precedente.

Questo processo viene ripetuto fino a quando non si raggiunge il livello più basso.

A questo punto, se viene soddisfatta la specifica della query, vengono restituite le regioni di celle rilevanti che soddisfano la query stessa.

Altrimenti, vengono individuati i dati presenti nelle celle rilevanti e vengono ulteriormente elaborati fino a quando essi non soddisfano le richieste della query.

STING offre diversi vantaggi rispetto agli altri metodi di clustering:

- Il calcolo basato sulla griglia è indipendente dalla query dal momento che l'informazione statistica memorizzata in ciascuna cella rappresenta l'informazione riassuntiva dei dati della cella, indipendentemente dalla query.
- La struttura a griglia facilita l'elaborazione parallela e l'aggiornamento incrementale.
- L'efficienza del metodo è il principale vantaggio: STING esamina il database solo una volta per calcolare i parametri statistici delle celle; pertanto, la complessità temporale per la generazione dei cluster è  $O(n)$ , dove  $n$  è il numero totale di oggetti.

Dopo aver generato la struttura gerarchica, *la complessità temporale per il query processing è  $O(g)$* , dove  $g$  è il numero totale delle celle al livello più basso; tale valore è, usualmente, molto più piccolo di  $n$ .

Poiché STING è un approccio multi-risoluzione per l'analisi del clustering, *la qualità del clustering di STING dipende dalla granularità del livello più basso della struttura*. Se la granularità è molto fine, il costo dell'elaborazione crescerà sostanzialmente; tuttavia, se il livello più basso della struttura a griglia è troppo sparso, l'accuratezza dei risultati finali è molto bassa.

Infine, STING non considera la vicinanza tra oggetti appartenenti a celle vicine per la costruzione di una cella padre. Di conseguenza, *le forme dei cluster risultanti sono isotetiche*, ovvero tutti i confini del cluster sono orizzontali o verticali e non esiste alcun confine diagonale. Ciò può abbassare la qualità e l'accuratezza dei risultati, il che controbilancia il tempo di elaborazione veloce della tecnica.

## 13.8 I metodi basati sul Modello

I metodi di clustering basati sul modello tentano di *ottimizzare la corrispondenza tra i dati e un qualche modello matematico predefinito dall'utente*.

Tali metodi sono spesso basati *sull'assunzione che i dati sono generati da una composizione di distribuzioni di probabilità*.

I metodi di clustering basati sul modello seguono *due principali approcci*: l'approccio statistico e l'approccio con reti neurali.

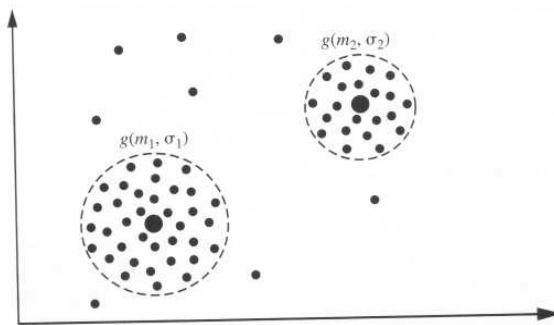
### 13.8.1 Expectation Maximization

Un cluster può essere *rappresentato matematicamente da una distribuzione di probabilità parametrica*.

I *dati complessivi* possono essere visti come una *composizione di tali distribuzioni* in cui ciascuna distribuzione viene tipicamente riferita come distribuzione componente.

Pertanto è possibile *clusterizzare i dati utilizzando un mixture density model* di  $k$  distribuzioni di probabilità in cui ciascuna distribuzione rappresenta un cluster. Il problema si riduce, quindi, a stimare le distribuzioni di probabilità che meglio di adeguano i dati.

Nella Figura 13.6 viene rappresentato *un esempio di un mixture density model* finito. In esso sono presenti due cluster ciascuno dei quali segue una distribuzione Gaussiana con la sua media e la sua deviazione standard.



**Figura 13.6.** Un esempio di mixture density model finito

L'algoritmo *EM* (Expectation Maximization) è un popolare *algoritmo di raffinamento iterativo* che può essere utilizzato per trovare le stime dei parametri.

Esso può essere visto come *un'estensione del paradigma k-means* soltanto che, invece di assegnare ciascun oggetto ad un cluster in modo rigido, *EM* assegna ciascun oggetto ad un cluster secondo cui un peso che rappresenta la probabilità di appartenenza. In altre parole, non ci sono confini stretti tra i cluster per cui le nuove medie vengono calcolate basandosi su misure pesate.

*EM* parte con una stima iniziale (o guess) dei parametri del modello (riferita collettivamente come vettore dei parametri).

Esso riassegna iterativamente gli oggetti tenendo conto del modello prodotto dal vettore dei parametri. Gli oggetti riassegnati vengono, quindi, usati per riassegnare le stime dei parametri.

Più specificatamente, l'algoritmo consiste nei seguenti passi:

1. *Viene fatta un'ipotesi iniziale del vettore dei parametri:* ciò comporta la selezione randomica di  $k$  oggetti per rappresentare le medie o i centri dei cluster nonché per fare delle ipotesi sui parametri addizionali.
2. *Vengono raffinati iterativamente i parametri basandosi sui seguenti passi:*
  - a) *Expectation Step:* ciascun oggetto  $x_i$  viene assegnato al cluster  $C_k$  con probabilità:

$$p(x_i \in C_k) = p(C_k|x_i) = \frac{p(C_k)p(x_i|C_k)}{p(x_i)}$$

dove  $p(x_i|C_k) = N(m_k, E_k(x_k))$  segue la distribuzione normale (cioè gaussiana), attorno alla media  $m_k$  con expectation  $E_k$ .

In altre parole, *questo passo calcola la probabilità di appartenenza* al cluster dell'oggetto  $x_i$ , per ciascuno dei cluster. Queste probabilità rappresentano le appartenenze ai cluster attese per l'oggetto  $x_i$ .

- b) *Maximization Step:* vengono utilizzate le stime di probabilità precedenti per *ri-stimare* (o *raffinare*) i parametri del modello. Per esempio:

$$m_k = \frac{1}{n} \sum_{i=1}^n \frac{x_i p(x_i \in C_k)}{\sum_j p(x_i \in C_j)}$$

Questo passo rappresenta la *massimizzazione della probabilità delle distribuzioni* rispetto ai dati.

*EM* è un algoritmo semplice e facile da implementare.

Nella pratica esso converge velocemente ma può non raggiungere l'ottimo globale. La convergenza è garantita per certe forme di funzione di ottimizzazione.

La complessità computazionale è lineare in  $d$  (il numero di caratteristiche dei dati di input presi in considerazione),  $n$  (il numero di oggetti) e  $t$  (il numero di iterazioni).

### 13.8.2 Autoclass

A livello commerciale è molto popolare *Autoclass*, un metodo di clustering bayesiano che utilizza una variante di *EM*.

I metodi di clustering bayesiano si focalizzano nel calcolo della densità di probabilità condizionale di classe.

In autoclass il clustering migliore massimizza la capacità di predire gli attributi di un oggetto dato il cluster corretto dell'oggetto.

*Autoclass* può anche stimare il numero di cluster.

Esso è stato applicato a diversi domini ed è stato capace di trovare una nuova classe di stelle basandosi su opportuni dati astronomici.

### 13.8.3 L'approccio statistico e COBWEB

Il clustering concettuale è una forma di clustering che deriva dal machine learning; esso, dato un insieme di oggetti non etichettati, ha lo scopo di produrre uno schema di classificazione per essi.

A differenza del clustering convenzionale, che identifica principalmente gruppi di oggetti simili, il clustering concettuale va oltre cercando di identificare anche le descrizioni di ciascun gruppo.

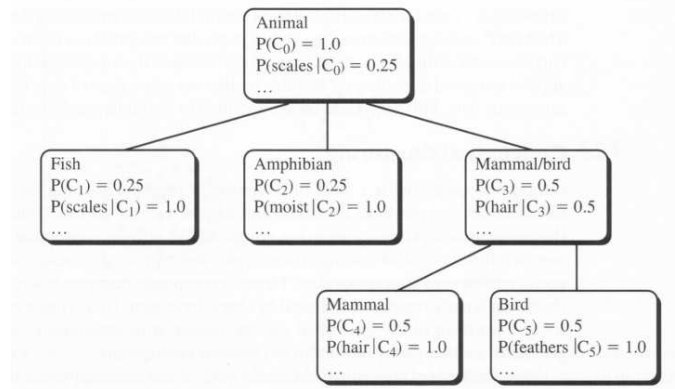
Pertanto, il clustering concettuale è un processo a due passi: dapprima viene effettuato il clustering; successivamente viene effettuata una caratterizzazione dei cluster prodotti.

Quindi, la qualità dei risultati non dipende soltanto dai cluster ottenuti, bensì tiene conto anche di altri fattori quali la generalità e la semplicità delle descrizioni dei concetti derivati.

Gran parte dei metodi di clustering concettuale *adottano un approccio statistico* che utilizza misure di probabilità per costruire i cluster.

Un metodo popolare e semplice di clustering concettuale incrementale è *COBWEB*. I suoi oggetti in input sono descritti da coppie attributo-valore categorici.

COBWEB crea un clustering gerarchico che consiste in un *albero di classificazione*. Un esempio di albero di classificazione, relativo a dati sugli animali, viene mostrato in Figura 13.7.



**Figura 13.7.** Un albero di classificazione

Un albero di classificazione differisce da un albero di decisione. Ciascun nodo in un albero di classificazione si riferisce ad un concetto e contiene una sua descrizione probabilistica che riassume gli oggetti ivi classificati. La descrizione probabilistica include la probabilità del concetto e probabilità condizionali della forma  $P(A_i = V_{ij} | C_k)$  dove  $A_i = V_{ij}$  è una coppia valore-attributo e  $C_k$  è una classe di concetti. In ciascun nodo vengono memorizzati vari contatori che servono per il calcolo delle varie probabilità.

Questa struttura è diversa, quindi, dagli alberi di decisione, che etichettano gli archi piuttosto che i nodi e usano descrittori logici piuttosto che probabilistici.

Si dice che i nodi fratelli ad un determinato livello di un albero di classificazione formano una partizione.

Per classificare un oggetto utilizzando un albero di classificazione, viene utilizzata una funzione di matching parziale che discende l'albero lungo il percorso con i migliori nodi di matching.

Per costruire un albero di classificazione, COBWEB utilizza una misura di valutazione statistica, denominata Category Utility, o *CU*. Essa viene definita come:

$$\frac{\sum_{k=1}^n P(C_k) [\sum_i \sum_j P(A_i = v_{ij} | C_k)^2 - \sum_i \sum_j P(A_i = v_{ij})^2]}{n}$$

dove  $n$  è il numero di nodi, concetti o “categorie” che formano una partizione  $\{C_1, C_2, \dots, C_n\}$  ad un determinato livello dell'albero.

In altre parole, la Category Utility rappresenta un incremento nel numero atteso di valori degli attributi che possono essere correttamente stabiliti, data una partizione, rispetto al numero di attributi stabilibili correttamente senza alcuna conoscenza.

Per ragioni di spazio e di tempo non mostreremo come viene derivata la Category Utility; diremo soltanto che essa tiene in considerazione la similarità intraclasse e quella interclasse. Infatti:

- La similarità intraclasse è la probabilità  $P(A_i = V_{ij} | C_k)$ ; maggiore è tale valore, più grande è la proporzione dei membri della classe che condividono questa coppia valore-attributo e, quindi, più predittiva risulterà la coppia.
- La dissimilarità interclasse è la probabilità  $P(C_k | A_i = V_{ij})$ ; maggiore è tale valore, minore è il numero di oggetti in classi diverse che condividono questa coppia attributo-valore e, quindi, più predittiva risulterà essere la coppia.

Vediamo, ora, come opera COBWEB. Esso incorpora incrementalmente oggetti in un albero di classificazione. In particolare, dato un nuovo oggetto, l'algoritmo discende l'albero lungo un opportuno percorso, aggiorna i contatori lungo il cammino alla ricerca del nodo migliore in cui classificare

l'oggetto. Più specificatamente, l'oggetto viene temporaneamente associato a ciascun nodo e viene calcolata la Category Utility della partizione risultante. Il nodo a cui viene associata la più alta Category Utility sarà il nodo destinato ad ospitare l'oggetto.

*Potrebbe capitare che l'oggetto non appartenga a nessuna delle classi esistenti.* Per affrontare questo problema COBWEB calcola anche la Category Utility della partizione che si avrebbe se si creasse un nuovo nodo associato a quell'oggetto. Questa viene confrontata con le Category Utility basate sui nodi pre-esistenti. L'oggetto viene associato ad una classe esistente oppure viene creata una nuova classe per esso a seconda di quale delle due possibilità consente di avere la più alta Category Utility.

Si noti che COBWEB *ha la capacità di individuare automaticamente il numero di classi di una partizione*. Pertanto, esso non ha la necessità di richiedere all'utente tale parametro.

*I due operatori menzionati precedentemente sono fortemente sensibili all'ordine di inserimento degli oggetti.*

COBWEB ha due operatori addizionali che lo aiutano ad essere meno sensibile all'ordine di inserimento dei dati. Tali operatori si chiamano *merging* e *splitting*. Quando un oggetto viene incorporato, viene presa in considerazione la possibilità di fondere in una singola classe i due nodi che lo ospitano meglio. Inoltre, viene presa in considerazione anche la possibilità di suddividere, tra le corrispondenti categorie, i figli del nodo che lo ospita meglio. Queste decisioni vengono assunte in base alla Category Utility. Gli operatori di merging e di splitting consentono a COBWEB di *effettuare una ricerca bidirezionale*; grazie a ciò sarà possibile che un merge annulli un precedente split.

COBWEB ha un certo numero di difetti. Innanzitutto è basato sull'assunzione che le distribuzioni di probabilità su attributi separati siano statisticamente indipendenti l'una dall'altra. Tale assunzione, tuttavia, non è sempre vera dal momento che può esistere una correlazione tra gli attributi.

Inoltre, l'utilizzo delle distribuzioni di probabilità sui cluster rende *piuttosto costoso il loro aggiornamento e la loro memorizzazione*. Ciò vale specialmente quando gli attributi hanno un gran numero di valori, dal momento che le complessità spaziali e temporali dipendono non solo dal numero di attributi ma anche dal numero di valori per ciascun attributo.

Inoltre, *l'albero di classificazione non è bilanciato in altezza* per dati in input asimmetrici; ciò può causare un drammatico degrado della complessità spaziale e temporale.

CLASSIT è un'estensione di COBWEB per il clustering incrementale di dati continui (o a valori reali). Esso memorizza una *distribuzione normale continua* (ovvero, la media e la deviazione standard) per ciascun attributo in ciascun nodo e utilizza una *misura di Category Utility modificata* che è un integrale sugli attributi continui invece di una sommatoria sugli attributi discreti, come avviene in COBWEB.

CLASSIT soffre degli stessi problemi di cui soffre COBWEB e, pertanto, non è adatto per il clustering di grandi database.

Il clustering concettuale è *popolare nella comunità di machine learning*; tuttavia, esso non è molto scalabile per grossi insiemi di dati.

## 13.9 Clustering di dati altamente dimensionali

*Gran parte dei metodi di clustering* sono stati progettati per clusterizzare dati con poche dimensioni e incontrano problemi quando la dimensionalità dei dati cresce di molto.

Ciò avviene perchè, quando la dimensionalità cresce, *generalmente soltanto un piccolo numero di dimensioni sono rilevanti per determinati cluster* ma i dati nelle dimensioni irrilevanti possono produrre molto rumore e mascherare i cluster reali.

Inoltre, quando la dimensionalità cresce, *i dati generalmente diventano sempre più sparsi*. Quando i dati diventano veramente sparsi, punti localizzati su diverse dimensioni possono essere considerati tutti ugualmente distanti e la misura di distanza, che è essenziale per l'analisi dei cluster, diventa poco significativa.

Per superare queste difficoltà è possibile considerare le tecniche di trasformazione degli attributi oppure le tecniche di selezione degli attributi.

- *Le tecniche di trasformazione delle caratteristiche* trasformano i dati in uno spazio più piccolo preservando generalmente la distanza relativa originaria tra gli oggetti. Essi riassumono i dati creando combinazioni lineari degli attributi e possono scoprire strutture nascoste nei dati.



Tuttavia, *tali tecniche non rimuovono dall'analisi nessuno degli attributi originari*. Ciò può diventare un problema quando vi è un gran numero di attributi irrilevanti. L'informazione irrilevante può mascherare i cluster reali anche dopo la trasformazione.

Inoltre, *gli attributi trasformati sono spesso difficili da interpretare* e ciò può rendere i risultati del clustering meno utili.

Pertanto, *la trasformazione degli attributi è adottata soltanto* per insiemi di dati in cui gran parte delle dimensioni sono rilevanti per l'attività di clustering. Tuttavia, nella pratica, molti insiemi di dati tendono ad avere molte dimensioni correlate o ridondanti.

- *La selezione degli attributi* ha lo scopo di rimuovere dimensioni irrilevanti o ridondanti. Dato un insieme di attributi, esso trova il sottoinsieme degli attributi più rilevanti per l'attività di mining. La selezione degli attributi comporta *la ricerca*, attraverso vari sottoinsiemi, *di attributi e la valutazione degli stessi sottoinsiemi* utilizzando opportuni criteri.

Essa viene effettuata molto comunemente mediante *il supervised learning* nel qual caso l'insieme degli attributi più rilevanti viene individuato rispetto alle determinate etichette di classe.

Tuttavia, *questa attività può essere effettuata anche mediante un processo non supervisionato*, quale l'analisi dell'entropia; quest'ultima è basata sulla proprietà che l'entropia tende ad essere bassa per dati che contengono cluster stretti.

*Il subspace clustering* è un'estensione della selezione degli attributi che si è dimostrato molto efficace in questo contesto. Esso è basato sull'osservazione di sottospazi diversi possono contenere cluster significativi differenti. Il subspace clustering cerca gruppi di cluster in sottospazi differenti dello stesso insieme dei dati.

*Tre approcci efficaci per il clustering di dati altamente dimensionali sono:*

- il dimension-growth subspace clustering, rappresentato, ad esempio, da CLIQUE;
- il dimension-reduction projected clustering, rappresentato, ad esempio, da PROCLUS;
- il frequent-based clustering, rappresentato, ad esempio, da pCluster.

## 13.10 Il clustering basato sui vincoli

Nelle discussioni precedenti *abbiamo assunto che il clustering fosse un processo automatico*, basato sulla valutazione delle funzioni similarità o distanza tra un insieme di oggetti da clusterizzare, con una guida o interazione dell'utente molto ridotta.

Tuttavia, gli utenti spesso hanno *una visione chiara delle richieste applicative* e vorrebbero utilizzare tale visione per guidare il processo di clustering e influenzare i suoi risultati.

Pertanto, in molte applicazioni, è desiderabile fare in modo che *il processo tenga in considerazione le preferenze e i vincoli dell'utente*. Esempi di tali informazioni includono il numero atteso di cluster, la dimensione minima e massima dei cluster, i pesi dei diversi oggetti e delle diverse dimensioni nonché altre caratteristiche desiderabili per i cluster risultanti.

Inoltre, quando un task di clustering coinvolge *uno spazio con un numero di dimensioni piuttosto elevato*, è molto difficile generare dei cluster significativi basandosi soltanto sui parametri di clustering. In questi casi, *gli input dell'utente* relativi alle dimensioni importanti oppure ai risultati desiderati servono *come suggerimenti* cruciali o come vincoli fondamentali per il clustering efficiente.

Il clustering basato sui vincoli ha lo scopo di *costruire cluster che soddisfano preferenze o vincoli specificati dall'utente*. A seconda della natura dei vincoli, il clustering basato sui vincoli può adottare approcci piuttosto differenti. Alcune categorie di vincoli sono le seguenti:

- *Vincoli su oggetti individuali*; questo vincolo limita l'insieme di oggetti da clusterizzare; esso può essere facilmente gestito tramite il pre-processing (ad esempio, effettuando una selezione tramite una query SQL), dopodiché il problema si riduce ad un'istanza di clustering non vincolato.
- *Vincoli sulla selezione dei parametri di clustering*; un utente può stabilire un intervallo desiderato per ciascun parametro di un metodo di clustering. Tali parametri sono generalmente specifici per l'algoritmo di clustering. Sebbene essi possono influenzare pesantemente i risultati di clustering, essi sono tuttavia confinati all'algoritmo stesso. Pertanto il loro tuning non viene generalmente considerato come una forma di constraint-based clustering.

- *Vincoli sulle funzioni di distanza o di similarità*; è possibile specificare diverse funzioni di distanza o di similarità per specifici attributi degli oggetti da clusterizzare oppure misure di distanza diverse per specifiche coppie di oggetti. Sebbene la loro scelta cambierà probabilmente i risultati, tuttavia ciò non altera il processo di clustering in sé e per sé. Tuttavia, in alcuni casi, questi cambiamenti possono rendere la valutazione della funzione distanza non banale, specialmente quando essa è strettamente correlata con il processo di clustering.
- *Vincoli sulle proprietà dei singoli cluster*; un utente può voler specificare un insieme di caratteristiche desiderate per i clustering risultanti; queste possono influenzare pesantemente il processo di clustering. Questa tipologia di problema è comune nella pratica.
- *Clustering basato su una supervisione parziale*; la qualità del clustering non supervisionato può essere significativamente migliorata utilizzando una qualche forma debole di supervisione. Ciò può avvenire nella forma di vincoli sulle coppie (ovvero, coppie di oggetti etichettati come appartenenti allo stesso cluster oppure a cluster differenti). Tale processo di clustering vincolato viene chiamato clustering semi-supervisionato.

### 13.11 Analisi degli outlier

Molto spesso esistono *dati che non sono in linea con le credenze* o il modello generale dei dati. Tali oggetti, che sono fortemente differenti o inconsistenti con il rimanente insieme di dati, vengono denominati outlier.

*Gli outlier possono essere causati da errori di misura* o di esecuzione. Per esempio, il fatto che venga mostrato come età di una persona il numero 999 potrebbe essere causato dalla presenza di un valore di default per tale attributo.

Alternativamente, *gli outlier potrebbero essere il risultato di una variabilità intrinseca* dei dati. Per esempio, lo stipendio del manager di un'azienda sarebbe un outlier rispetto agli stipendi degli altri impiegati.

*Molti algoritmi di data mining tentano di minimizzare l'influenza degli outlier* o, addirittura, di eliminarli. Ciò, tuttavia, *potrebbe comportare la perdita di importanti informazioni nascoste* perché “un rumore di una persona potrebbe essere un segnale per un'altra persona”.

In altre parole, *gli outlier potrebbero essere particolarmente interessanti*, come nel caso della ricerca delle frodi, dove gli outlier possono indicare attività fraudolenta. Pertanto, la ricerca e l'analisi degli outlier è un interessante task di data mining, noto come outlier mining.

*L'outlier mining ha moltissime applicazioni*. Come menzionato in precedenza, esso può essere utilizzato nella ricerca delle frodi, per esempio cercando un utilizzo insolito delle carte di credito o dei servizi di comunicazione. Inoltre, esso è utile nel marketing personalizzato, per identificare il comportamento di spesa dei clienti con redditi estremamente bassi o estremamente alti, oppure nell'analisi medica, per trovare risposte insolite a vari trattamenti medici.

*L'outlier mining può essere descritto nel seguente modo*: dati un insieme di  $n$  oggetti e un numero atteso di outlier  $k$ , trovare i  $k$  oggetti che sono considerevolmente dissimili, eccezionali o inconsistenti rispetto al resto dei dati.

*Il problema può essere decomposto in due sottoproblemi*: (i) definire quali dati possono essere considerati inconsistenti in un dato insieme; (ii) trovare un metodo efficiente per estrarre gli outlier così definiti.

*Il problema di definire gli outlier non è banale*.

Se viene utilizzato un *modello di regressione per modellare i dati*, l'analisi dei residui può dare una buona stima di quanto sono estremi i dati. Tale attività, tuttavia, diviene complicata quando è necessario trovare outlier in dati temporali, dal momento che essi possono essere nascosti in cambiamenti nel trend, in cambiamenti stagionali o in altri cambiamenti ciclici.

Quando vengono utilizzati *dati multidimensionali* potrebbe essere estrema non tanto una dimensione quanto una combinazione di dimensioni. Per dati categorici, la definizione degli outlier richiede una considerazione speciale.

Per la ricerca degli outlier *i metodi di visualizzazione potrebbero sembrare la scelta più ovvia*, dal momento che gli occhi umani sono molto veloci ed efficaci nel notare inconsistenze nei dati. Tuttavia, ciò non vale per i dati con rappresentazioni cicliche dove i valori che sembrano essere outlier potrebbero essere perfettamente validi nella realtà. I metodi di visualizzazione dei dati sono deboli nel cercare

outlier in dati con molti attributi categorici oppure in dati altamente dimensionali, dal momento che gli occhi umani sono capaci di visualizzare dati numerici con solo due o tre dimensioni.

*I metodi basati sul computer* per la ricerca degli outlier possono essere *suddivisi in quattro approcci*: l'approccio statistico, l'approccio basato sulla distanza, l'approccio agli outlier locali basato sulla densità e l'approccio basato sulla deviazione.

Si noti che mentre gli algoritmi di cluster scartano gli outlier come rumore, *essi potrebbero essere modificati per includere la ricerca di outlier* come un sotto-prodotto della loro esecuzione. In generale, gli utenti devono verificare che ciascun outlier scoperto da questi approcci sia in realtà un vero outlier.



## Il Web Mining

*In questo capitolo daremo una breve descrizione delle problematiche relative al Web Mining. Dopo una breve introduzione sull'argomento, esamineremo le tecniche di Web Structure Mining, quelle di Web Content Mining (nel cui ambito considereremo anche le tecniche di Text Mining) e, infine, le tecniche di Web Usage Mining.*

### 14.1 Introduzione al Web Mining

Il Web è un mezzo vivente, crescente, popolare e partecipativo che memorizza e diffonde *grandi quantità di informazioni* distribuite, interconnesse, eterogenee e dinamiche.

Il processo di creazione dei contenuti per il Web si basa su *diversi contributi indipendenti* (provenienti sia da individui che da imprese) che non sono soggetti ad alcuno standard o ad alcuna autorità centralizzata e agiscono in modo collaborativo.

Da una parte, ciò determina una partecipazione di massa nella realizzazione dei contenuti che, in ultima analisi, fa sì che il Web copra quasi ogni aspetto degli sforzi umani.

D'altro canto, tuttavia, la continua crescita del Web, insieme alla sua inerente eterogeneità e mancanza di organizzazione, pone *delle sfide severe* alla ricerca delle informazioni di alta qualità.

Alcune delle difficoltà sono le seguenti:

- *La dimensione del Web sta crescendo rapidamente.* Infatti, un numero crescente di librerie digitali, di cataloghi di prodotti, di riviste, di newsgroup, di report medici, di database, di applicazioni Web, di siti Web (sia per gli individui che per le organizzazioni), e così via, sono resi pubblicamente disponibili attraverso il Web.  
La vastità dei contenuti è alla base della *incapacità intrinseca* a navigare ed indicizzare l'intero Web per effettuare attività di Information Retrieval.
- *Le pagine Web sono unità informative fortemente dinamiche.* Non solo esse non hanno una struttura unificante, per cui è difficile categorizzarle o indicizzarle, ma anche informazioni quali notizie, quotazioni azionarie e annunci vengono continuamente aggiornati.  
Frequenti cambiamenti vengono apportati *sia ai Web service che agli hyperlink*. Inoltre, *un accesso uniforme* a pagine Web originariamente pensate per utenti con conoscenze, interessi e aspettative diverse non fornisce alcun supporto generale per soddisfare le loro specifiche richieste di informazioni.
- Il Web è un hypermedium popolare, in cui il contenuto viene creato da *diversi individui*, spesso con scopi diversi e contrastanti. Di conseguenza, anche se i creatori di contenuti impongono un ordine su base estremamente locale, *la struttura complessiva del Web appare caotica*.  
Una struttura ad alto livello può essere utilizzata nella ricerca di informazioni utili solo attraverso una qualche *analisi a posteriori* che sia capace di discriminare tra le diverse motivazioni che stanno dietro un hyperlink.  
Infine, per ragioni commerciali o competitive, *la rappresentazione sbagliata dei contenuti* (ovvero l'inserimento di termini senza significato tra i contenuti di pagine Web) viene tipicamente utilizzata per far sì che i motori di ricerca basati su keyword assegnino alle pagine Web prodotte un rank alto su più query.



- *Solo una piccola porzione del Web è tipicamente rilevante* per una determinata richiesta di informazioni.

Tuttavia, l'informazione di interesse viene spesso trovata come risultato di *tentativi di ricerca frustranti*. Questi hanno lo scopo di restringere opportunamente il Web ad una frazione di pagine che sono effettivamente correlate all'argomento di ricerca iniziale.

Successivamente, *un'ulteriore distillazione* di queste pagine Web permette l'identificazione di informazione di elevata qualità. Un compito talmente complesso scoraggia inevitabilmente gli utenti.

*I motori di ricerca* possono aiutare ad affrontare le sfide precedentemente menzionate. Tuttavia l'efficacia di tali servizi soffre di un *certo numero di limitazioni*.

Data la vastità dei contenuti e la scarsa accuratezza un problema fondamentale è la bassa precision e il basso recall. *La bassa precision* è una conseguenza del cosiddetto problema dell'abbondanza: sebbene un numero enorme di pagine Web viene tipicamente restituito in risposta ad una query dell'utente su qualche argomento vasto, la stragrande maggioranza di queste pagine o risulta marginalmente rilevante per l'argomento di ricerca oppure contiene informazioni di scarsa qualità.

*Il basso recall* è, invece, dovuto all'incapacità di indicizzare l'intero Web, che abbassa notevolmente la frazione di documenti rilevanti che vengono effettivamente recuperati.

Gli effetti della bassa precision e del basso recall sono ancora più frustranti quando il processo di ricerca incorre nel *problema della scarsità*, ovvero quando mancano le risposte a query che coinvolgono pochissime pagine.

Un'altra principale debolezza dei motori di ricerca tradizionali consiste nel loro *mero utilizzo di indici basati su keyword* per tenere traccia di sottoinsiemi di pagine Web contenenti determinati termini. Ciò dà luogo a due ulteriori complicazioni.

Innanzitutto, *la presenza di sinonimie e omonimie* può contribuire ulteriormente a diminuire la qualità dei risultati dei motori di ricerca se non si fa un notevole sforzo nel disambiguare il significato delle singole parole.

In secondo luogo, *molti documenti* fortemente rilevanti per una determinata query possono non essere restituiti per il fatto che *non contengono le stesse keyword della query utente*. L'analisi precedente mostra l'inadeguatezza dei motori di ricerca tradizionali per il recupero di informazioni sul Web.

*Il Web Mining* è una vasta area che vuole utilizzare le tecniche di Data Mining per scoprire automaticamente conoscenza nascosta, potenzialmente utile, sia dai dati che dai servizi su Web.

*Esso può essere visto come un passo* che consente di estendere l'intero processo di KDD all'ambiente Web per estrarre pattern interessanti, quali strutture nascoste, tendenze, associazioni, correlazioni e dipendenze statistiche tra le pagine Web, le loro parole e i loro hyperlink.

Il Web Mining è un'area in cui *convergono molti tentativi di ricerca* relativi a diversi campi che vanno dai database all'intelligenza artificiale, all'information retrieval e alla social network analysis.

Le tecniche di Web Mining possono essere *classificate in tre principali categorie*:

- *Web Structure Mining*. I metodi di questa categoria inferiscono informazioni esaminando la topologia della struttura dei link tra pagine Web.

Questo tipo di informazioni è *utile per un certo numero di scopi*: categorizzazione dei siti Web, capacità di individuare relazioni di similarità tra siti Web, capacità di sviluppare una metrica opportuna per la valutazione della rilevanza delle pagine Web.

- *Web Content Mining*. Lo scopo principale di questa categoria di metodi è quello di estrarre informazioni utili dal contenuto delle risorse Web.

Le tecniche di Content Mining *possono essere applicate* a sorgenti di dati eterogenei (quali documenti HTML/XML, librerie digitali oppure risposte a query sul database) e sono correlate a tecniche di information retrieval tradizionali.

Tuttavia, l'applicazione di tali tecniche alle risorse Web consente la definizione di nuovi domini applicativi interessanti, quali:

- *sistemi di query su Web*, che utilizzano l'informazione sulla struttura dei contenuti Web per gestire query di ricerca complesse;
- *agenti intelligenti di ricerca*, che lavorano per conto degli utenti basandosi sia su una descrizione del loro profilo che su una conoscenza specifica del dominio per filtrare opportunamente i risultati forniti dai motori di ricerca in risposta alle query dell'utente.

- *Web Usage Mining*. In questa categoria di metodi l'attenzione si rivolge all'applicazione delle tecniche di data mining per scoprire dei pattern di utilizzo dai dati Web allo scopo di capire e servire meglio le necessità delle applicazioni basate su Web e degli utenti finali.

Gli *access log* sono la principale sorgente di dati per una qualunque attività di Web Usage Mining: gli algoritmi di data mining possono essere applicati a tali log al fine di inferire informazioni che descrivono l'utilizzo delle risorse Web.

Il Web Usage Mining è *alla base di una varietà di applicazioni* che coinvolgono le statistiche per l'attività di un sito Web, le decisioni di business, la riorganizzazione della struttura dei link e/o dei contenuti di un sito Web, gli studi di usabilità, l'anomalia del traffico e la sicurezza.

Tuttavia, è bene enfatizzare che, nella pratica, le attività di Web Mining precedentemente menzionate si influenzano l'un l'altra. Per esempio, alcuni utilizzano la struttura dei link per una classificazione della pagine Web più accurata. Ancora, alcuni propongono un agente software, denominato Web Watcher, il cui scopo è quello di assistere gli utenti mentre navigano su un sito Web; esso si basa su una combinazione dei motori di Content Mining e Usage Mining.

## 14.2 Web Structure Mining

Le tecniche di *Information Retrieval tradizionali* sono pensate per collezioni finite di documenti, ovvero per unità di informazioni autocontenute che sono generalmente esplicative e veritiere in merito al loro contenuto.

Il Web, al contrario, è *un mezzo ipertestuale di dimensione sconosciuta*, le cui unità informative di base, ovvero le pagine Web, non sono spesso né veritiere né esplicative dei loro contenuti.

Ciò è dovuto, rispettivamente, all'esplosione enorme dei *contenuti non testuali*, quali immagini, e alla presenza di diverse keyword scorrelate che fanno in modo che i motori di ricerca assegnino un rank alto alle pagine Web con gran parte delle query comuni.

Una nozione fondamentale nel campo dell'Information Retrieval, *ovvero il recall*, assume poco significato nel Web dal momento che non vi è alcun modo fattibile per contare il numero complessivo di pagine Web rilevanti per una determinata query.

Alcuni studi sottolineano che ciò che sembra accadere sul Web è *una sorta di precision a basso recall*: i motori di ricerca organizzano le loro migliori risposte nella prima pagina di risultati e gli utenti tipicamente non richiedono ulteriori pagine.

Una migliore *comprensione delle peculiarità del Web* può fornire un supporto notevole per un Information Retrieval migliore.

Il focus è su ciò che distingue gli ipertesti dai documenti tradizionali, ovvero *gli hyperlink* che vengono utilizzati come un supplemento essenziale ai contesti testuali. Precisamente, un hyperlink da una pagina Web ad un'altra può essere interpretato come un giudizio di rilevanza dell'ultima pagina, stabilito dagli autori della prima.

Visti in questo modo, i link sono *un meccanismo potente per conferire autorevolezza* alle pagine Web. Infatti, ispezionare il vicinato di una certa pagina Web può essere visto come un approccio alternativo per valutare il livello di interesse della pagina; tale approccio può fornire risultati migliori sulla qualità dell'informazione rispetto ad un'analisi meramente testuale.

La ricerca nel campo dello Structure Mining *persegue i seguenti tre obiettivi*:

- utilizzo degli hyperlink e del testo per una *migliore classificazione degli argomenti*;
- sviluppo di modelli per *la creazione, la rimozione e la modifica* sia di nodi (ovvero di pagine Web) che di archi (ovvero di hyperlink) sul grafo associato al Web, il che fornisce una visione sulle dinamiche sottostanti al Web;
- sviluppo di tecniche efficaci che consentono di filtrare *un piccolo numero di pagine Web di alta qualità* da un numero enorme di risposte a query generiche.

## 14.3 Web Content Mining

Il Web Content Mining consente la scoperta di informazione interessante dal contenuto delle sorgenti informative. Il Web comprende *una grande varietà di tipi di dati*, quali i dati testuali, audio, video, immagini, metadati ed hyperlink e ciò ispira linee di ricerca distinte.

Sebbene, in letteratura, vi siano *tentativi di effettuare il Mining su più tipi di dati*, il focus è principalmente quello di estrarre contenuti testuali o ipertestuali. Questi possono essere ulteriormente *suddivisi in* non strutturati (ad esempio i testi liberi), semi-strutturati (ad esempio, i documenti HTML ed XML) e strutturati (ad esempio, le pagine HTML generate dinamicamente), che tipicamente gestiscono dati provenienti da tabelle relazionali o da librerie digitali.

L'atto di applicare tecniche di Data Mining a dati non strutturati è usualmente conosciuto come *Knowledge Discovery in Text*, o *Text Data Mining* o *Text Mining*.

La ricerca nel Web Content Mining può essere considerata *da due punti di vista*: quello dei database e quello dell'Information Retrieval.

*Dal punto di vista dei database*, il Web Content Mining vuole modellare i dati sul Web per sviluppare delle strategie di ricerca più sofisticate rispetto a quelle basate sulle keyword.

*Dal punto di vista dell'Information Retrieval*, invece, il Web Content Mining vuole sviluppare delle tecniche efficaci di Information Filtering e Retrieval che agiscono per conto dell'utente tenendo in considerazione qualche profilo delle sue richieste informative effettive.

### 14.3.1 Web Content Mining dal punto di vista dei database

L'applicazione di tecniche di database ai contenuti sul Web ha lo scopo di affrontare *tre principali problemi*: modellazione e querying del Web, estrazione e integrazione di conoscenza, progettazione e riorganizzazione dei siti Web.

*I primi due casi* rientrano, in realtà, nell'ambito del Web Content Mining. In questo caso lo scopo è quello di scoprire un'organizzazione più strutturata dei contenuti dei siti Web per una migliore gestione dell'informazione e del querying sul Web.

Sono stati proposti *diversi studi per questo scopo*; essi trattano principalmente i dati semi-strutturati, ovvero i dati che hanno una qualche struttura ma non possiedono uno schema ben definito.

In questo contesto, *l'OEM* (Object Exchange Model) è un importante modello di dati per rappresentare dati semi-strutturati nella forma di un grafo i cui nodi corrispondono ad oggetti e i cui link corrispondono a relazioni tra oggetti.

Gran parte delle *applicazioni che coinvolgono dati semi-strutturati* si basano sulla scoperta dello schema dei documenti Web o sulla costruzione di riassunti strutturali dei dati, noti come DataGuide, che sono spesso approssimati per ragioni computazionali.

### 14.3.2 Web Content Mining dal punto di vista dell'Information Retrieval

Il tentativo di recuperare e filtrare informazioni da documenti Web non strutturati o semi-strutturati ha determinato *l'esigenza di una modellazione efficiente dei loro contenuti testuali*.

A tal fine sono stati proposti molti *modelli alternativi per la rappresentazione dei contenuti*.

Gran parte degli sforzi di ricerca in tale campo utilizza il cosiddetto *bag of words o vector representation*, un modello che ignora l'ordine con cui le parole compaiono nei documenti Web e si focalizza sulle statistiche relative ai termini individuali.

Un consistente insieme di rappresentazioni alternative di contenuti si basa sul prendere in considerazione *la posizione delle parole all'interno dei documenti Web*. Esempi sono gli *n-grammi* (ovvero, le sequenze di *n* parole), le frasi, i termini e la rappresentazione relazionale (che consiste nell'utilizzo della logica del primo ordine per modellare le relazioni tra parole e le loro corrispondenti posizioni).

Nonostante l'abbondanza dei tipi di rappresentazione dei contenuti, *nessuna differenza rilevante nella loro efficacia* è stata trovata attraverso distinti domini.

Recentemente *i documenti Web semi-strutturati* hanno guadagnato molta attenzione consentendo una grande varietà di applicazioni che vanno dalla classificazione e il clustering degli ipertesti all'apprendimento di relazioni tra documenti Web e alla ricerca di pattern all'interno dei loro contenuti.

In particolare, il clustering degli ipertesti è un processo chiave per consentire attività di recupero delle informazioni di base, quali la ricerca, il browsing e la visualizzazione.

### 14.3.3 Text Mining

Gran parte degli *studi sul Data Mining* si sono focalizzati sui dati strutturati, quali i dati relazionali, quelli transazionali e i Data Warehouse.

Tuttavia, *nella realtà*, una porzione sostanziale dell'informazione disponibile viene memorizzata in *database testuali* (o database di documenti) che consistono in una grande collezione di documenti provenienti da varie sorgenti, quali nuovi articoli, articoli di ricerca, libri, librerie digitali, messaggi di e-mail e pagine Web.

*I database testuali stanno crescendo rapidamente* a causa della crescente quantità di informazione disponibile in formato elettronico; si pensi, ad esempio, alle pubblicazioni elettroniche, ai vari tipi di documenti elettronici, alle e-mail e al Web (che può essere visto come un enorme database testuale dinamico e interconnesso). Oggigiorno, gran parte delle informazioni nella Pubblica Amministrazione, nell'industria, nei servizi e nelle altre istituzioni è memorizzata elettronicamente, sotto forma di database testuali.

I dati memorizzati in gran parte dei database testuali sono *dati semi-strutturati* in quanto essi non sono né completamente privi di struttura né completamente strutturati. Per esempio, un documento può contenere pochi campi strutturati, quali il titolo, gli autori, la data di pubblicazione, la categoria, ecc., ma può anche contenere alcuni componenti testuali largamente non strutturati, ad esempio l'abstract e i contenuti.

Nella ricerca sui database sono stati recentemente effettuati *moltissimi studi per modellare e incrementare i dati semi-strutturati*. Inoltre, per gestire documenti non strutturati, sono state sviluppate tecniche di Information Retrieval, ad esempio tecniche di text indexing.

*L'Information Retrieval è un campo* che, per molti anni, si è sviluppato in parallelo con la ricerca sui database. A differenza dei sistemi di database che si sono focalizzati sull'interrogazione e sull'elaborazione di dati strutturati, l'Information Retrieval si è concentrato sull'organizzazione e il recupero di informazioni da un gran numero di documenti testuali.

Dal momento che l'Information Retrieval e i sistemi di database gestiscono tipi di dati differenti, *alcuni problemi tipici dei database sono generalmente non presenti nei sistemi di Information Retrieval* (si pensi, ad esempio, al controllo della concorrenza, al recovery, alla gestione delle transazioni e all'aggiornamento).

Analogamente, *alcuni problemi comuni di Information Retrieval non si incontrano generalmente nei sistemi di database tradizionali* (si pensi, ad esempio, ai documenti non strutturati, alla ricerca approssimata basata su keyword e alla nozione di rilevanza).

A causa dell'abbondanza dell'informazione testuale, *l'Information Retrieval ha trovato molte applicazioni*. Esistono molti sistemi di Information Retrieval, quali i cataloghi online delle librerie, i sistemi online per la gestione documentale, e, più recentemente, i motori di ricerca su Web.

*Un tipico problema di Information Retrieval consiste* nel localizzare i documenti rilevanti in una collezione di documenti basandosi su una query dell'utente, spesso costituita da alcune parole che descrivono un bisogno dell'utente oppure da un documento di esempio rilevante.

In tale problema di ricerca *l'utente prende l'iniziativa e specifica l'informazione* per lui rilevante; questo è molto appropriato quando l'utente ha un bisogno informativo specifico, ad esempio quello di trovare le informazioni per comperare una macchina usata.

Quando l'utente ha un bisogno informativo di lungo termine (ad esempio, interessi di ricerca), *un sistema di Information Retrieval può anche prendere l'iniziativa* per prelevare eventuali nuove informazioni arrivate presentandole all'utente, se queste vengono giudicate rilevanti per lui.

Tale processo di accesso all'informazione viene chiamato *Information Filtering* e i corrispondenti sistemi sono spesso chiamati *filtering system* o *recommender system*. Da un punto di vista tecnico, tuttavia, la ricerca e il filtraggio condividono molte tecniche comuni.

In via del tutto generale, *i metodi di Information Retrieval appartengono a due categorie*; in particolare, essi generalmente vedono il problema di retrieval come un problema di selezione dei documenti oppure come un problema di ranking dei documenti.

- *Nei metodi di selezione dei documenti*, la query specifica i vincoli che determinano l'interesse di un documento. Un tipico metodo di questa categoria è il *boolean retrieval method* in cui un documento viene rappresentato da un insieme di keyword e un utente fornisce un'espressione booleana di keyword, come "car and repair shops", "database system but not Oracle".
- *I metodi di ranking dei documenti* utilizzano la query per ordinare i documenti in base alla loro rilevanza.

*Le tecniche di Information Retrieval tradizionali stanno diventando inadeguate* a causa della continua crescita della quantità di dati testuali disponibili. Tipicamente, solo una piccola frazione dei molti documenti disponibili saranno rilevanti per un determinato utente.

Senza conoscere cosa potrebbe esserci nei documenti, è difficile formulare query efficaci per analizzare ed estrarre informazioni utili dai dati.

*Per poter effettuare query efficaci* sono necessari strumenti capaci di confrontare documenti diversi, di ordinarli in base alla loro importanza o di trovare pattern e trend in documenti multipli.

Per tutte queste ragioni, *il text mining è diventato un argomento sempre più popolare ed essenziale nel Data Mining.*

Vi sono *molti approcci per il Text Mining* che possono essere classificati da prospettive differenti tenendo conto degli input forniti al sistema e dei task di Data Mining da eseguire.

In genere, se si tiene conto del tipo di dati ricevuto in input, i principali sono:

- *Il keyword-based approach*, in cui l'input è un insieme di keyword o termini nel documento. Un semplice keyword-based approach può soltanto *scegliere relazioni ad un livello relativamente superficiale*; esso, ad esempio, può scoprire nomi composti (ad esempio, "database" e "system") oppure pattern coesistenti (ad esempio, "terrorist" e "explosion"). Esso non può portare a conclusioni più profonde sul testo.
- *Il tagging approach*, in cui l'input è un insieme di tag. Il tagging approach si può *basare sui tag ottenuti dal tagging manuale* (che è costoso e ingestibile in presenza di grosse collezioni di documenti), oppure da qualche algoritmo di categorizzazione automatico (che può processare un insieme relativamente piccolo di tag e può richiedere di definire le categorie in anticipo).
- *L'Information Extraction approach*, in cui l'input è rappresentato da informazioni semantiche, quali eventi e fatti, o da entità scoperte tramite l'Information Extraction. L'Information Extraction approach è più avanzato e *può portare alla scoperta di conoscenza profonda*; tuttavia esso *richiede un'analisi semantica del testo* per mezzo di metodi basati sulla comprensione del linguaggio naturale oppure sul machine learning. Questo risulta essere un task di Knowledge Discovery affascinante.

*Vari task di Data Mining possono essere effettuati sulle keyword*, sui tag o sull'informazione semantica estratta. Essi includono il clustering dei documenti, la classificazione, l'estrazione delle informazioni, l'estrazione di regole associative e l'analisi dei trend.

## 14.4 Web Usage Mining

Il Web Usage Mining ha lo scopo di scoprire *conoscenza nascosta in grossi volumi di Web Usage data*, ovvero di dati risultanti da attività di browsing degli utenti, per modellare e predire efficientemente la loro interazione con il Web.

A differenza del Web Content Mining che si applica ai dati Web primari, il Web Usage Mining comporta l'utilizzo di tecniche di Data Mining statistico e convenzionale per processare *dati secondari*.

Ciò nonostante, il Web Usage Mining è *al centro di una grande varietà di applicazioni*, alcune delle quali sono di seguito menzionate:

- *Statistiche per l'attività e l'inattività di un sito Web*. In questo caso lo scopo è quello di monitorare le operazioni sul sito Web per mezzo di misure periodiche, quali il numero totale di visite, il numero medio di accessi, il tempo di visita medio, la lunghezza media di un percorso di browsing attraverso il sito, gli errori del server, gli errori di pagine non trovate, le pagine più visitate, le pagine di entrata/uscita, e così via.
- *Iniziative di e-commerce*. In questo caso si cerca di segmentare gli utenti Web sulla base delle loro tendenze di acquisto e della loro permeabilità alla pubblicità. Ciò consente di identificare il target (ovvero il sottoinsieme di utenti) più adatto per ciascuna iniziativa di marketing, con un impatto positivo sui parametri cruciali, quali l'attrazione del cliente, il suo mantenimento o la sua perdita.
- *Progettazione di siti Web*. Le riorganizzazioni periodiche della struttura dei link o dei contenuti delle pagine di un sito Web vengono effettuate per riflettere meglio il suo attuale utilizzo.
- *Analisi del traffico*. Lo scopo è quello di capire quali siano le richieste minime, in termini di hardware e di replicazione dei dati, che consentono ad un sito Web di gestire efficientemente una certa percentuale di richieste dell'utente.



- *Studi sull'usabilità.* In tale contesto il focus sta nella valutazione del grado complessivo di efficacia (ovvero, precisione e completezza) e di efficienza con cui gli utenti possono raggiungere i loro obiettivi di browsing.
- *Sicurezza.* Scoprire politiche di sicurezza aiuta ad individuare ed impedire percorsi di browsing non autorizzati attraverso porzioni di un sito Web a cui viene negato un accesso pubblico.
- *Personalizzazione.* Questa attività consiste nel ritagliare i contenuti, i servizi e la struttura di un sito Web alle richieste informative, alle preferenze e alle aspettative di uno specifico utente, inferite dall'analisi del suo comportamento di browsing.

#### 14.4.1 Sorgenti dei dati di utilizzo

I dati di utilizzo possono essere collezionati a tre diversi livelli nel canale di comunicazione ideale tra un utente e il sito Web che egli sta correntemente accedendo: tali livelli sono il sito Web, il proxy e il cliente. A ciascun livello viene catturato il comportamento di browsing relativo ai diversi segmenti di utenti.

Spesso, i dati di utilizzo possono anche essere *estratti da un database di un'organizzazione*.

#### Sorgenti localizzate a livello di Web Site

A livello di Web site, i *Web log* rappresentano il modo più popolare di tracciare il comportamento dell'utente.

Registrando ogni richiesta di ingresso al Web server, i log catturano esplicitamente le attività di browsing dei visitatori in modo concorrente. Ciò vuol dire che i Web log catturano un comportamento *multi-user/single-site*.

Sono stati proposti svariati *formati ad hoc per organizzare i dati nei Web log*: due formati popolari sono il CLF (Common Log Format) e l'ECLF (Extended Common Log Format). Recentemente introdotto dal W3C, l'ECLF migliora il CLF aggiungendo a ciascuna entrata un certo numero di nuovi campi che si rivelano particolarmente utili per analisi demografiche e log summary.

*I principali campi di ciascuna entrata all'interno di un Web log ECLF sono i seguenti:*

- *IP Address* è l'indirizzo Internet dell'host remoto da cui è stata ricevuta una richiesta; può essere un indirizzo di un proxy server.
- *User ID* è riempito solo in quei casi in cui si richiede agli utenti di fornire la loro autenticazione per accedere dati sicuri sul server Web.
- *Time* è il timestamp che indica quando una richiesta in arrivo è stata ricevuta dal server Web.
- *Request* indica le principali componenti di una richiesta, ovvero il suo metodo (GET, POST o HEAD), l'URI della risorsa richiesta e il protocollo della richiesta stessa (tipicamente, http).
- *Status* viene utilizzato per registrare il tipo di risposta ad una particolare richiesta; esso può essere una redirectione, un rilascio con successo della risorsa richiesta, un errore interno al Web server, un errore durante il processo di rilascio.
- *Size* riflette la quantità di byte di una risposta ad una richiesta dell'utente.
- *Referrer* è l'URI della risorsa Web da cui si è originata la richiesta in ingresso.
- *Agent* è un campo che fornisce dettagli sulla natura del sistema operativo e del browser utilizzati a livello client.

I log ECLF possono anche includere *cookie*, ovvero pezzi di informazione generati unicamente dai server Web per identificare e tracciare gli utenti durante le loro attività di browsing.

Il principale limite relativo ai Web log è che essi non consentono di *catturare in modo affidabile il comportamento dell'utente*. Ad esempio, *il tempo di accesso percepito a livello di sito Web* può essere molto più lungo di quello misurato a livello di client. Ciò è tipicamente dovuto ad un numero di ragioni inevitabili, quali la banda del cliente, il tempo di trasmissione necessario al Web server per rilasciare la risorsa richiesta e lo stato di congestione della rete.

Inoltre, i Web log *possono non registrare alcune richieste dell'utente*. Tale perdita di informazione generalmente accade quando un utente accede ripetutamente una stessa pagina ed è presente il caching; tipicamente viene catturata solo la prima richiesta; quelle successive possono essere servite dalla cache che può essere locale al client o parte di un proxy server intermedio.

*Questi errori possono essere considerati* mentre si estraggono pattern di utilizzo dai Web log; svariate euristiche sono state definite per effettuare il pre-processing dei Web log al fine di diminuire gli effetti collaterali dei problemi precedentemente evidenziati.

I dati sul comportamento dell'utente possono essere collezionati mediante *due approcci alternativi*: l'utilizzo della tecnologia di raket sniffing o l'integrazione, all'interno degli application server, di meccanismi di tracking ad hoc. Entrambi gli approcci sono concepiti per catturare quelle informazioni che non sono considerate dai Web log, ad esempio i parametri di richiesta inviati al Web server attraverso il metodo nascosto POST.

### Sorgenti localizzate a livello di proxy server

I proxy server sono localizzati tra gli utenti finali e i siti Web e agiscono per conto di un visitatore *mentre naviga* attraverso un sito Web.

*Un'enorme quantità di dati di utilizzo* può essere collezionata a livello di proxy server, principalmente sotto forma di richieste da molti utenti a molti siti Web. Da questi repository può essere inferito uno sguardo su un comportamento di browsing *multi-user / multi-site*.

Precisamente, ciascuna collezione di dati di utilizzo consente di esaminare le attività di browsing *di un particolare segmento di utenti Web*: quelli serviti dallo stesso proxy server.

### Sorgenti localizzate a livello di client

A livello di client *le attività di un utente possono essere ricostruite* (in modo più affidabile rispetto a ciò che permette il tracking lato server) utilizzando agenti remoti che possono essere applicazioni ad hoc o una versione modificata di un browser Web standard.

I due approcci esibiscono diverse peculiarità. Più precisamente, *le applicazioni ad hoc* si focalizzano su un comportamento *single-user / single-site*, mentre *i browser modificati* vengono utilizzati per estendere il tracking lato client al fine di catturare il comportamento di browsing *single-user / multi-site*.

#### 14.4.2 Un tipico processo di Web Usage Mining

Idealmente, un tipico processo di Web Usage Mining può essere suddiviso in tre parti:

- *data pre-processing*, in cui i dati di utilizzo grezzi vengono trasformati in un insieme di dati pronti per il Mining (più precisamente, i dati grezzi di utilizzo vengono convertiti in astrazioni ad alto livello, quali viste, sessioni, transazioni e utenti);
- *pattern discovery*, in cui vengono utilizzate svariate tecniche provenienti da campi differenti (quali il machine learning e la statistica) per inferire pattern di utilizzo potenzialmente interessanti;
- *pattern analysis*, in cui i pattern identificati vengono ulteriormente ispezionati, aggregati e/o filtrati per ricavare dai pattern individuali una comprensione profonda relativa all'utilizzo del sito Web in esame.

## Data Mining e Oracle: Data Miner

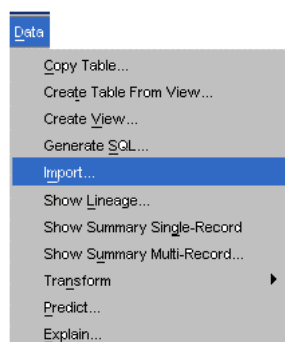
*Questo capitolo ha lo scopo di introdurre Oracle 10g Release 2 Data Mining per Oracle 10g. Attraverso questo tool di facile utilizzo, è possibile definire, implementare e gestire operazioni di Data Mining.*

### 15.1 Trasformare i dati

I dati utilizzati per le operazioni di data mining spesso vengono raccolti da locazioni differenti e spesso sono utili delle trasformazioni per rendere i dati adatti alle operazioni di data mining. A tale scopo vedremo come prelevare dati da differenti sorgenti e come effettuare trasformazioni per adattare meglio i dati alle specifiche richieste dei vari algoritmi di data mining.

#### 15.1.1 Importare i dati

Per importare un file csv in una tabella, selezionare Import dal menu Data e seguire i passi del Wizard.



#### 15.1.2 Visualizzatore di dati e di statistiche

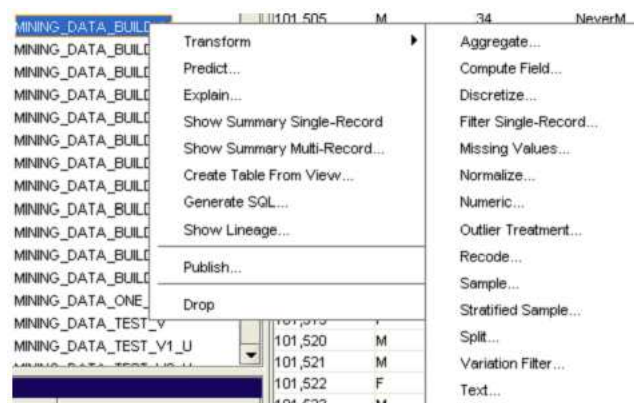
Cliccare sul nome di una tabella o di una vista per visualizzare la struttura.

PK	Name	Type	Size	Scale	Allow NULLS
X	CUST_ID	NUMBER	22		X
X	CUST_GENDER	CHAR	1		X
X	AGE	NUMBER	22		✓
X	CUST_MARITAL_STA...	VARCHAR2	20		✓
X	COUNTRY_NAME	VARCHAR2	40		X
X	CUST_INCOME_LEVEL	VARCHAR2	30		✓
X	EDUCATION	VARCHAR2	21		✓
X	OCCUPATION	VARCHAR2	21		✓
X	HOUSEHOLD_SIZE	VARCHAR2	21		✓
X	YRS_RESIDENCE	NUMBER	22		✓
X	AFFINITY_CARD	NUMBER	10	0	✓
X	BULK_PACK_DISNET...	NUMBER	10	0	✓
X	FLAT_PANEL_MONIT...	NUMBER	10	0	✓
X	HOME_THEATER_PA...	NUMBER	10	0	✓
X	BOOKKEEPING_APPL	NUMBER	10	0	✓
X	PRINTER_SUPPLIES	NUMBER	10	0	✓
X	Y_BOX_GAMES	NUMBER	10	0	✓
X	OS_DOC_SET_KANA	NUMBER	10	0	✓

Cliccare sulla voce Data per visualizzare i dati della tabella/vista.

Structure Data									
Fetch Size: 100	Fetch Next	Refresh							
CUST_ID	CUST_GEND...	AGE	CUST_MARI...	COUNTRY_N...	CUST_INCO...	EDUCATION	OCCUPATION	HOUSEHOLD...	YF
101,501	F	41	NeverM	United State...	J: 190,000 - ...	Masters	Prof.	2	4
101,502	M	27	NeverM	United State...	I: 170,000 - 1...	Bach.	Sales	2	3
101,503	F	20	NeverM	United State...	H: 150,000 - ...	HS-grad	Cleric	2	2
101,504	M	45	Married	United State...	B: 30,000 - 4...	Bach.	Exec.	3	5
101,505	M	34	NeverM	United State...	K: 250,000 - ...	Masters	Sales	9+	5
101,506	M	38	Married	United State...	K: 250,000 - ...	HS-grad	Other	3	4
101,507	M	28	Married	United State...	J: 190,000 - ...	< Bach.	Sales	3	3
101,508	M	19	NeverM	United State...	K: 250,000 - ...	HS-grad	Sales	2	2
101,509	M	52	Married	Brazil	K: 250,000 - ...	Bach.	Other	3	5
101,510	M	27	NeverM	United State...	L: 300,000 a...	Bach.	Sales	2	3
101,511	M	30	NeverM	United State...	H: 150,000 - ...	Bach.	Sales	2	5
101,512	F	30	NeverM	United State...	I: 170,000 - 1...	Profsc	Prof.	2	4
101,513	M	31	Married	United State...	J: 190,000 - ...	Bach.	Sales	3	3
101,514	M	45	NeverM	United State...	L: 300,000 a...	HS-grad	Sales	2	5
101,515	F	36	NeverM	United State...	J: 190,000 - ...	11th	Other	9+	2
101,516	M	33	Married	United State...	G: 130,000 - ...	< Bach.	Exec.	3	4
101,517	F	38	NeverM	United State...	I: 170,000 - 1...	HS-grad	Sales	9+	4

Cliccare col tasto destro del mouse sul nome della tabella o vista, verrà visualizzato un menu a tendina con altre opzioni.



Per visualizzare un resoconto statistico dei dati, cliccare Show Summary Single-Record.

Summary statistics for DMUSER2\MINING_DATA_BUILD_V								Attribute Count: 18	
Name	Mining Attr.	Attribute D.	Average	Max	Min	Sample Size	Variance		
AFFINITY_CARD	categorical	NUMBER	0.25	1	0	1500	0.19		
AGE	numerical	NUMBER	39.69	90	17	1500	195.95		
BOOKKEEPING_APPLICATION	categorical	NUMBER	0.98	1	0	1500	0.11		
BULK_PACK_DISKETTES	categorical	NUMBER	0.63	1	0	1500	0.23		
COUNTRY_NAME	categorical	VARCHAR2				1500			
CUST_GENDER	categorical	CHAR				1500			
CUST_ID	numerical	NUMBER	102,250.5	103,000	101,501	1500	197,625		
CUST_INCOME_LEVEL	categorical	VARCHAR2				1500			
CUST_MARITAL_STATUS	categorical	VARCHAR2				1500			
EDUCATION	categorical	VARCHAR2				1500			
FLAT_PANEL_MONITOR	categorical	NUMBER	0.58	1	0	1500	0.24		
HOME_THEATER_PACKAGE	categorical	NUMBER	0.58	1	0	1500	0.24		
HOUSEHOLD_SIZE	categorical	VARCHAR2				1500			
OCCUPATION	categorical	VARCHAR2				1500			
OS_DOC_SET_KANJI	categorical	NUMBER	0	1	0	1500	0		
PRINTER_SUPPLIES	categorical	NUMBER	1	1	1	1500	0		
YRS_RESIDENCE	categorical	NUMBER	4.09	14	0	1500	3.69		
Y_BOX_GAMES	categorical	NUMBER	0.29	1	0	1500	0.2		

Per visualizzare la distribuzione dei valori di un attributo occorre selezionarlo e cliccare Histogram.

I valori vengono divisi in range (detti *bin*). Gli attributi numerici vengono divisi in bin di uguale ampiezza, gli attributi categorici sono suddivisi utilizzando il metodo “Top N” (dove N è il numero totale di bin).

### 15.1.3 Trasformazione dei dati

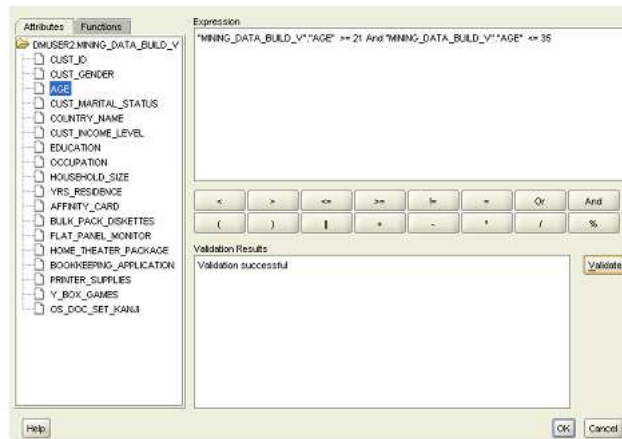
Cliccare col tasto destro del mouse sul nome della tabella per accedere al wizard che si occupa della trasformazione.

## Filtrare singoli record

Supponiamo che vogliamo concentrare l'attenzione sui clienti che hanno un'età compresa tra i 21 e i 35 anni. È possibile filtrare i dati per avere una tabella o vista che include soltanto questi dati.

Selezionare Transformations dal menu Data e quindi scegliere Filter Single-Record per lanciare il wizard.

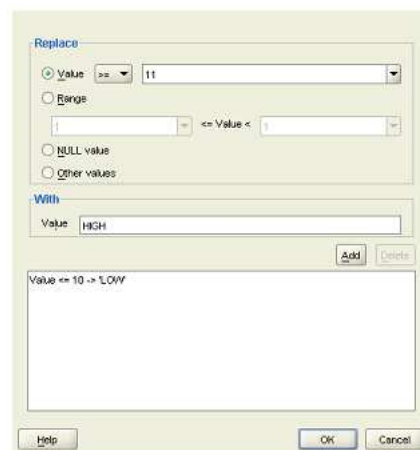
Il wizard ci aiuterà ad selezionare la tabella, dare un nome alla nuova tabella creata e specificare le condizioni di filtro tramite menu grafico.



## Trasformare i campi

Il processo di trasformazione dei dati permette di rimpiazzare i valori degli attributi con nuovi valori. Ad esempio supponiamo che l'attributo LENGTH\_OF\_RESIDENCE contenga valori numerici compresi tra 1 e 34. Attraverso tale processo è possibile rendere la tabella più adeguata alle operazioni di data mining considerando due classi di residenza: LOW per persone che hanno una residenza inferiore ai 10 anni e HIGH per le persone con residenza superiore ai 10 anni.

Per effettuare ciò selezionare Transform dal menu Data e quindi scegliere Recode. Seguire i passi del wizard per selezionare la tabella d'origine, scegliere il nome della nuova tabella, selezionare l'attributo da ridefinire e scegliere l'operazione di trasformazione attraverso un'interfaccia grafica.



Prestare comunque attenzione che non viene effettuato alcun controllo di verifica che la condizione di trasformazione sia consistente.

## Costruire nuovi campi

Quando vengono preparati i dati per un processo di data mining, è spesso necessario derivare una nuova colonna dalle colonne esistenti. Esempio tipico è quando si hanno due colonne che memorizzano delle date ma, ai fini delle nostre analisi, non è interessante il valore contenuto in sé, ma bensì l'intervallo di tempo trascorso tra una data e l'altra.

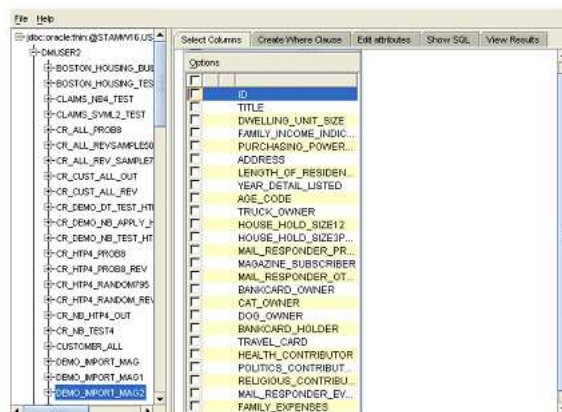
Per effettuare tale tipo di operazione selezionare Transform dal menu Data e quindi scegliere Compute Field.

I passi del wizard permetteranno di scegliere la tabella di origine, specificare il nome della nuova tabella costruita e definire (attraverso un'interfaccia grafica) la nuova colonna.



## Creare una vista dei dati

È spesso utile combinare i dati da differenti tabelle relazionali. Per effettuare ciò selezionare Create View dal menu Data; cliccare sul segno “+” accanto alla connessione di database ed espandere la lista ad albero degli schemi disponibili. Espandere gli schemi per identificare le tabelle e le viste disponibili. Fare doppio click sul nome della tabella o vista che si intende selezionare per portarla nell'area di lavoro a destra.



Cliccare sulle checkbox accanto ai nomi degli attributi per includere i corrispettivi campi sulla vista.

Sempre attraverso interfaccia grafica è possibile effettuare delle join tra tabelle.



## 15.2 Attribute Importance

Se i dati hanno troppi attributi, è probabile che non tutti sono utili per predire un modello; infatti, alcuni possono semplicemente contenere del “rumore”.

Oracle Data Mining fornisce una caratteristica chiamata Attribute Importance (AI) per classificare gli attributi per importanza nel caratterizzare un valore target. In tal modo è possibile accrescere l'accuratezza ed il tempo di esecuzione per i problemi di classificazione.

Per utilizzare tale funzione:

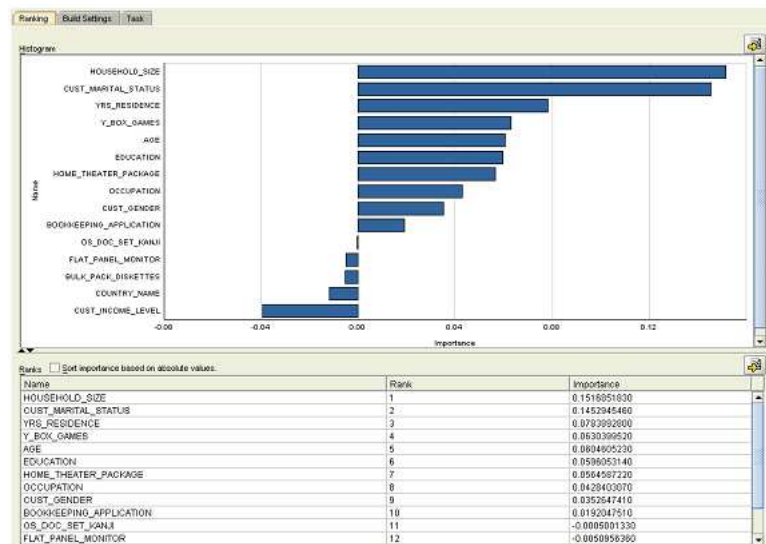
1. Cliccare su Build... dal menu Activity e selezionare Attribute Importance.
2. Selezionare la tabella che si intende analizzare e, successivamente, l'attributo Target.



3. Selezionare un nome per tale attività e cliccare su Finish.

Tale funzione classificherà gli attributi della tabella selezionata per importanza nel caratterizzare l'attributo Target.

Quando tutti i passi vengono completati, cliccare su Result nella voce Build per visualizzare il grafico contenente la lista degli attributi classificati per importanza.



### 15.3 Classificazione

Oracle Data Mining fornisce quattro metodi per risolvere i problemi di classificazione, in questo capitolo tratteremo l'algoritmo Naive Bayes. Tale algoritmo osserva i dati storici e calcola le probabilità condizionali per i valori target.

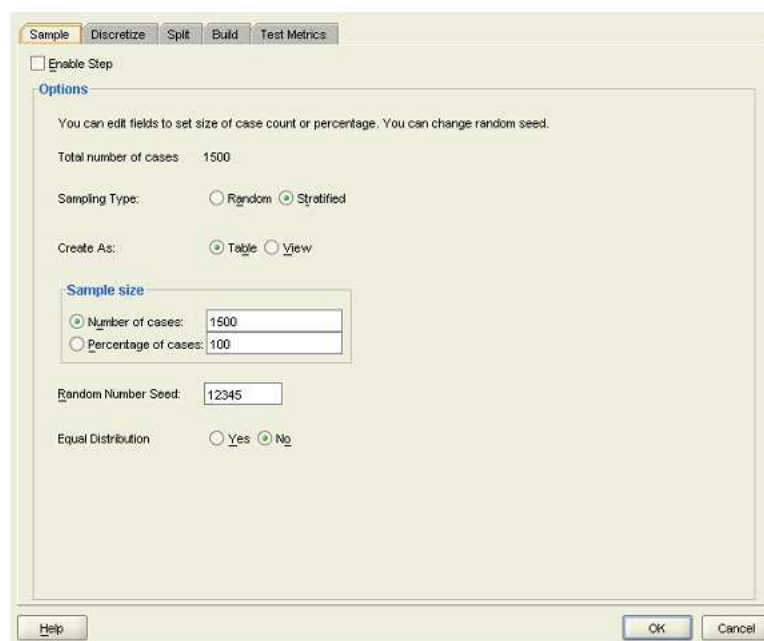
Per utilizzare l'algoritmo Naive Bayes occorre cliccare su Build... dal menu Activity e selezionare come funzione Classification e come algoritmo Naive Bayes.

Un wizard ci aiuterà a selezionare la tabella di origine, l'attributo target e, infine, assegnare un nome all'attività.

Terminato il wizard, cliccando su Advanced Settings... vengono gestiti altri parametri che per questioni di semplicità non sono stati presi in considerazione dal wizard.



La finestra che si apre cliccando su Advanced Settings... è la seguente:



Possiamo notare che la voce Sample non è abilitata. Infatti Oracle Data Mining è scalabile per tabelle di qualsiasi dimensione e spesso non vi è alcuna necessità di campionare i dati; tuttavia, se vi sono delle limitazioni hardware potrebbe essere preferibile abilitare tale voce.

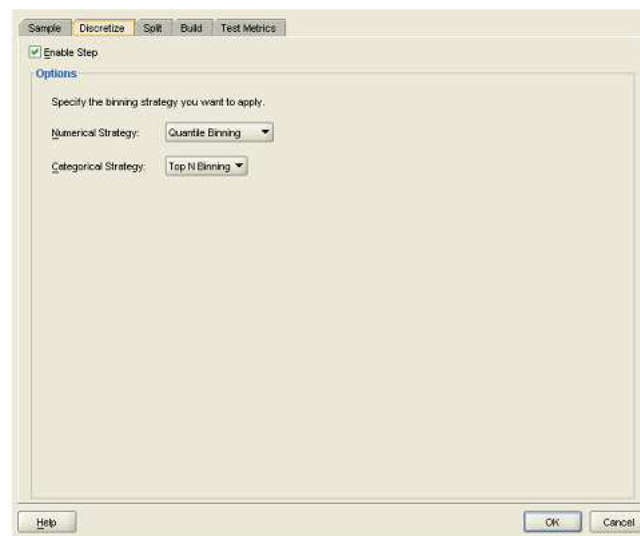
Se viene abilitata la voce Sample è possibile scegliere la dimensione dei campioni:

- Attribueno a Sampling Type il valore Random si sceglie un numero di casi che hanno approssimativamente la stessa distribuzione del valore target dei dati originali.
- Attribueno a Sampling Type il valore Stratified si sceglie un numero di casi che hanno approssimativamente lo stesso numero dei casi per ciascun valore target. È consigliabile utilizzare tale metodo solo in situazioni in cui il valore target interessante è raro (come nei problemi in cui si è interessati ad individuare attività illegali o malattie rare).

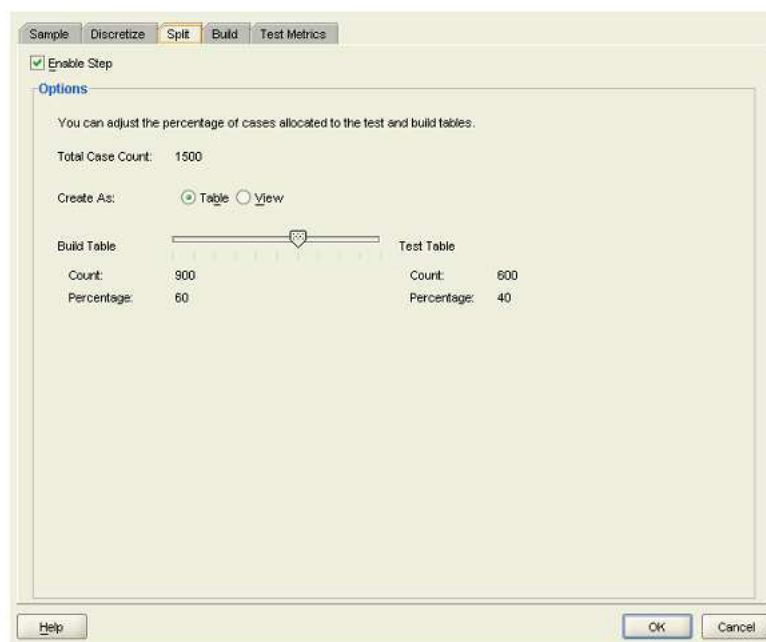
Nella voce Discretize è possibile gestire la discretizzazione dei dati. I dati numerici saranno raggruppati in range di valori, i dati categorici saranno divisi in gruppi (o bin), uno per ciascun valore.

È possibile scegliere il raggruppamento numerico:

- Quantile, crea bin con approssimativamente un numero di casi uguali su ciascun bin.
- Equi-width, crea bin di ampiezza identica, indipendentemente dal numero di casi di ciascun bin (tenere conto che tale strategia potrebbe anche creare dei bin vuoti).



Attraverso la voce Split è possibile modificare la percentuale dei casi destinati a testare il modello (detti dataset di test) e dei casi destinati alla costruzione del modello di classificazione.



Nella voce Build si trovano due sotto-voci:

- **General:** è possibile scegliere se preferire un'accuratezza media migliore o un'accuratezza globale migliore. Ad esempio, un modello potrebbe essere buono a predire i clienti con bassi valori di salario ma non buono a predire i clienti con alti valori di salario. Tipicamente si vuole un modello capace a predire tutti i casi, così Maximum Average Accuracy è il valore predefinito.



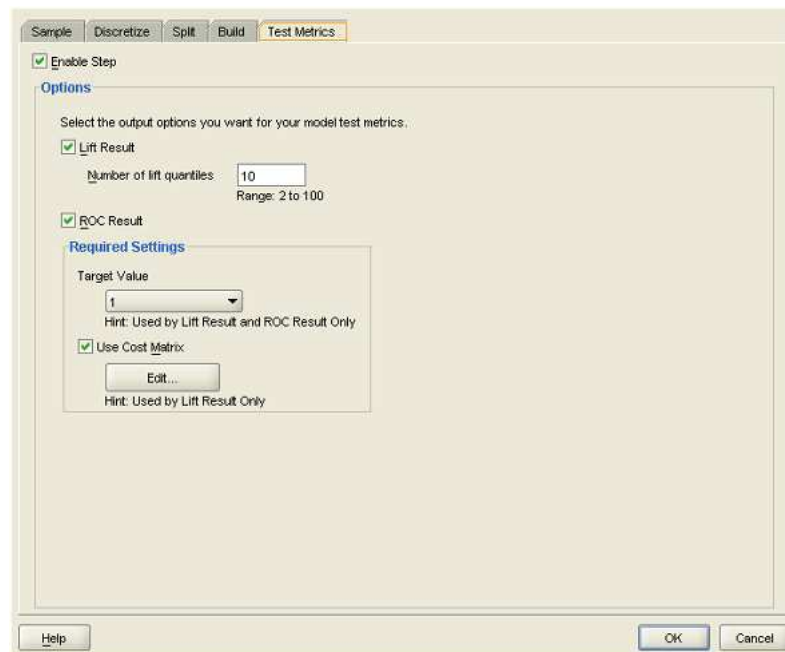
- **Algorithm Settings:** è possibile gestire ed eliminare rari e possibili casi di “rumore”.

Nella voce Test Metrics sono disponibili metriche di controllo per il problema di classificazione.

ROC è un metodo che sperimenta l'analisi “what if”: se la soglia di probabilità viene cambiata, come questa influenza il modello?

La Confusion Matrix indica il tipo di errori che il modello è predisposto a fare.

Lift è un tipo differente di test e misura quanto “velocemente” il modello trova i valori target realmente positivi. Tale analisi è adatta a rispondere la domanda: “Quanti clienti del database devo sollecitare per trovare il 50% dei clienti disposti a comprare il prodotto X?”



Cliccando OK si ritornerà nell'ultima finestra del wizard. Per avviare la classificazione è sufficiente cliccare su Finish. Verrà visualizzata la seguente finestra:

The screenshot shows the Orange3 workflow window with the following steps:

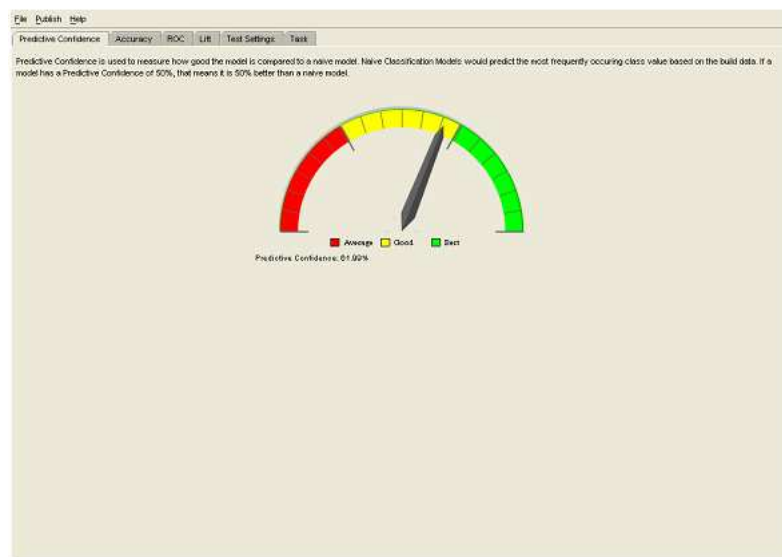
- Sample**: This step samples the mining data. Although not normally required, this step can be used to sample very large data sets. To complete this step manually, click Custom. (Status: Skipped)
- Discretize**: This transformation step discretizes the mining data. To complete this step manually, click Custom. (Status: Completed)
- Split**: This transformation step splits the mining data into build and test data sets. To complete this step manually, click Custom. (Status: Completed)
- Build**: This step builds the mining model. To complete this step manually, click Custom. (Status: Completed)
- Test Metrics**: This step creates a test metric result. To complete this step manually, click Custom. (Status: Completed)

Buttons for each step include Options..., Reset, and Custom... (or Skip for Sample). The Test Metrics step also includes a Select ROC Threshold button.

Quando tutti i passi vengono completati, cliccare su Result nella voce Test Metrics.

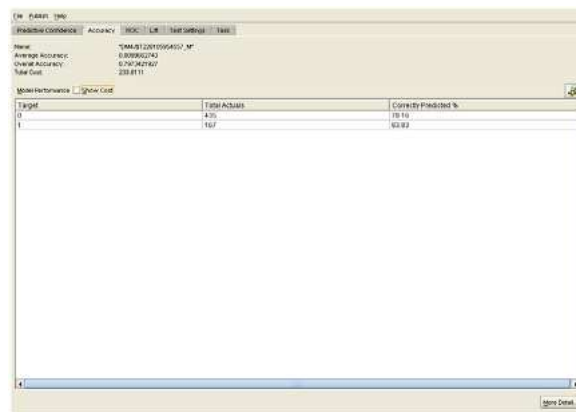
### 15.3.1 Test Metrics

La pagina iniziale mostra la Predictive Confidence, questa è una indice della efficacia del modello. Ad esempio, supponiamo di esaminare un database dove il 40% dei valori target è 1 e il 60% dei valori target è 0. Supponiamo che stiamo cercando i casi in cui il valore target sia 1. In tal caso, se non applicassimo il modello, ci aspetteremmo di avere successo nel trovare i casi di interesse circa il 40% di volte. Tuttavia, usando il modello predittivo, ci aspetteremmo di migliorare sensibilmente tale stima.



La pagina Accuracy mostra differenti interpretazioni dell'accuratezza del modello quando applicato ai dataset di test. Nei dataset di test il valore dell'attributo target è conosciuto, così le predizioni possono essere confrontate col valore reale.

La seguente schermata indica che nell'esempio ci sono 435 casi in cui il valore target è 0 e il modello predice correttamente il 78.16% di questi. Similmente il modello predice accuratamente l'83.83% dei 167 casi in cui il valore target è 1.

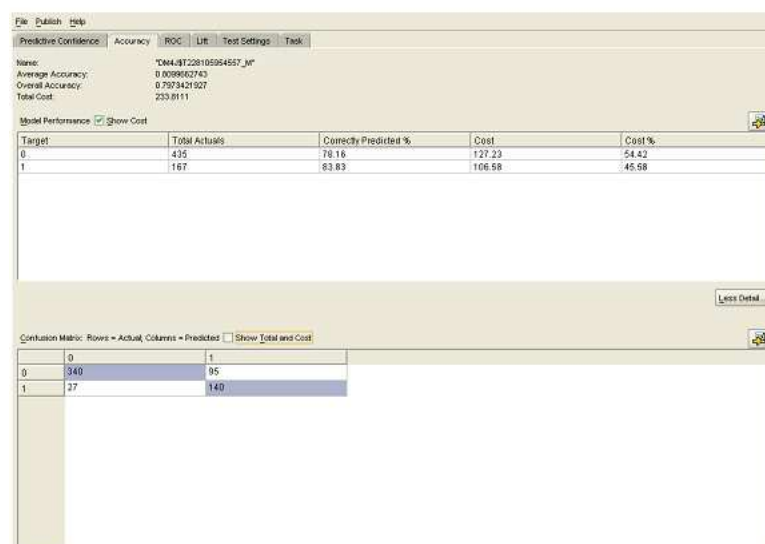


Target	Total Actuals	Correctly Predicted %
0	435	78.16
1	167	83.83

Cliccando sulla checkbox Show Cost si ha un'altra misura di accuratezza. Minore è tale misura, migliore sarà il modello.

Cliccando su More Detail verrà visualizzata la Confusion Matrix, questa mostra le tipologie di errori che possono capitare utilizzando il modello.

La Confusion Matrix viene calcolata applicando il modello al dataset di test. I valori dell'attributo target sono conosciuti e vengono rappresentati da righe, le colonne rappresentano predizioni effettuate dal modello. Ad esempio il numero 27 della figura in seguito indica le predizioni che sono falsi-negativi (predizioni di 0 quando il valore reale è 1), mentre il numero 95 indica i falsi-positivi (predizioni di 1 quando il valore reale è 0).



Target	Total Actuals	Correctly Predicted %	Cost	Cost %
0	435	78.16	127.23	54.42
1	167	83.83	106.58	45.58

Confusion Matrix: Rows = Actual, Columns = Predicted ☐ Show Total and Cost

	0	1
0	340	95
1	27	140

Cliccando su Show Total and Cost appariranno altre statistiche derivate dalla Confusion Matrix.

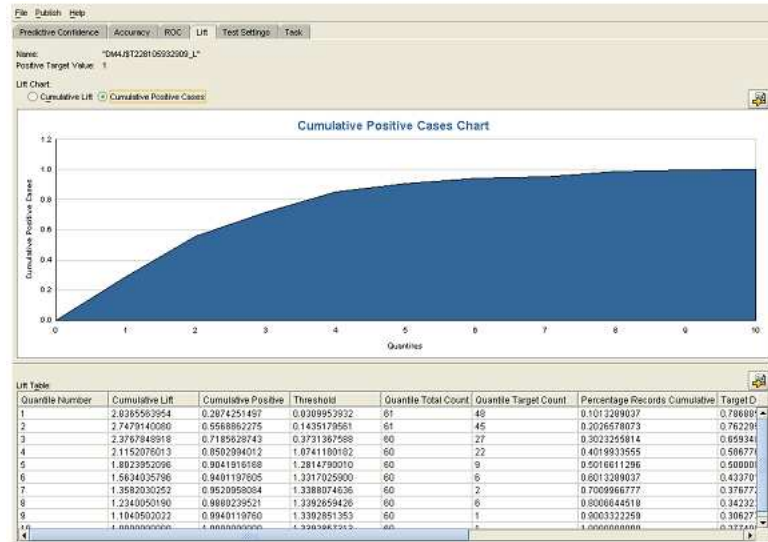
La pagina Lift mostra due grafici che denotano differenti interpretazioni dei calcoli lift. Il Cumulative Positive Chart è spesso denominato anche Lift Chart o Gains Chart.

Oracle Data Mining applica il modello sui dataset di test e ordina i risultati per probabilità e suddivide la lista in parti eguali (detti Quantile - il numero totale di Quantile di default è pari a 10) e conta i reali-positivi in ciascun Quantile.

I risultati di questo test indicano l'incremento di valori positivi che si ottengono considerando la percentuale di clienti che statisticamente, secondo le predizioni del modello, hanno probabilità maggiore di rispondere positivamente, rispetto al caso in cui viene selezionata una percentuale di clienti completamente random.

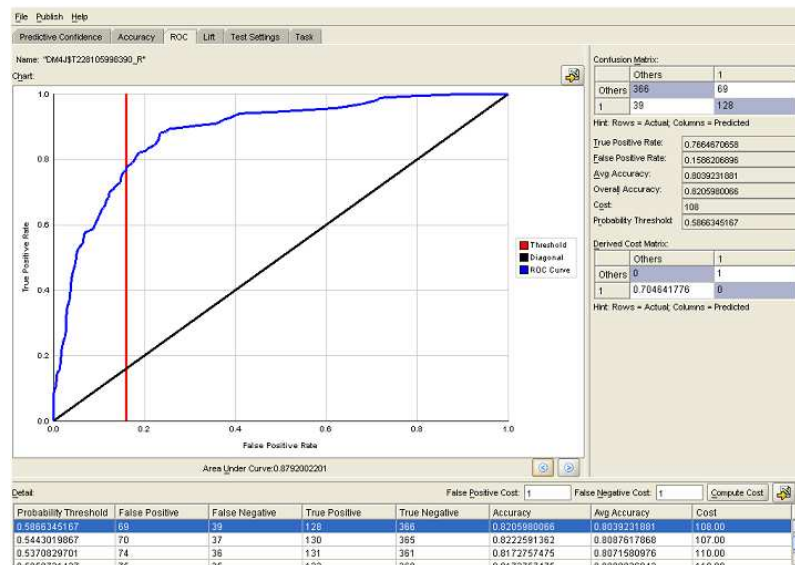
Nell'esempio sotto, la Lift Table mostra che per un Quantile Number pari a 3, Cumulative Lift è pari a 2.37; tali valori indicano che applicando il modello e selezionando da tale modello i migliori 30% clienti (ovvero, i migliori 3 Quantile su 10) avremo un responso almeno doppio rispetto al caso in cui viene selezionato il 30% di clienti su base random. La colonna successiva (Cumulative Positive) indica che oltre il 71% di risposte verosimili vengono trovate nei migliori 3 Quantile.





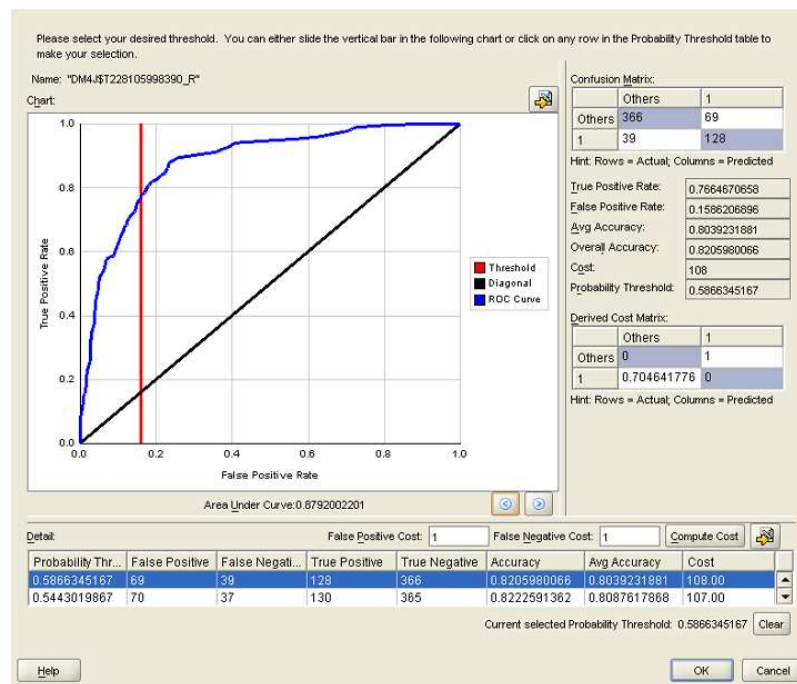
La finestra ROC esplora possibili cambiamenti nei parametri del modello.

La metrica ROC osserva come differenti impostazioni sul modello hanno effetto sulla Confusion Matrix. Ad esempio, supponiamo che si richiede che il numero di falsi-negativi sia ridotto il più possibile avendo fissato un massimo numero di predizioni positive. Muovendo la linea verticale rossa è possibile osservare i cambiamenti della Confusion Matrix.

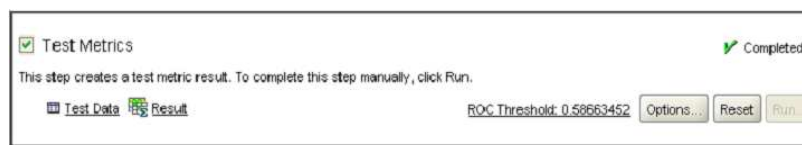


L'esempio mostra che i falsi-negativi possono essere ridotti a 39 tenendo i positivi (reali e non) sotto 200 ( $128 + 69 = 197$ ).

Tale pagine è di natura sperimentale, per rendere permanente il cambiamento al modello utilizzato, ritornare alla pagina Display e cliccare Select ROC Threshold. Si aprirà una nuova finestra, selezionare la riga contenente la soglia determinata prima e cliccare OK. Il modello è così modificato.



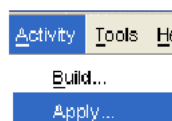
La soglia modificata apparirà nella voce Test Metrics:



### 15.3.2 Applicare il modello creato

Si supponga che dopo le analisi sperimentali si determina che il modello Naive Bayes è la migliore soluzione per risolvere il nostro problema. In tal caso il modello è pronto per essere applicato alla popolazione generale o a nuovi dati. Tale attività viene spesso denominata “scoring the data”.

Selezionare Apply... dal menu Activity.



Quando il modello viene applicato a nuovi dati, questi devono essere preparati e trasformati esattamente nello stesso modo di quanto fatto nelle sorgenti utilizzate nel passo Build.

Seguire i passi del wizard. Quando l'attività viene completata, cliccare su Result nella voce Apply per vedere un campione dei risultati ottenuti. La tabella contiene, per ciascuna riga, l'identificatore, il valore target più probabile e la probabilità della predizione. Cost è un'altra misura e rappresenta il costo di una predizione incorretta; sono preferibili bassi valori di Cost.

CASE_ID	PREDICTION	PROBABILITY	COST	RANK
100,001	0	0.9253	0.2947	1
100,002	0	0.8095	0.4143	1
100,003	0	0.9885	0.1323	1
100,004	0	0.9801	0.0787	1
100,005	1	0.987	0.0443	1
100,006	0	1	0.0001	1
100,007	0	0.9972	0.0109	1
100,008	0	0.9744	0.1009	1
100,009	1	0.4204	0.7762	1
100,010	0	0.9554	0.176	1
100,011	0	0.9998	0.0007	1
100,012	1	1	0	1
100,013	1	0.6208	0.5079	1
100,014	0	0.9829	0.1482	1
100,015	1	0.6789	0.4301	1
100,016	0	0.9962	0.015	1
100,017	0	1	0	1
100,018	0	1	0	1
100,019	1	0.7955	0.2739	1
100,020	0	0.9893	0.1212	1
100,021	1	0.9416	0.0742	1
100,022	1	0.8763	0.0218	1
100,023	1	0.950	0.0589	1
100,024	1	0.7771	0.2995	1
100,025	0	0.999	0.004	1
100,026	1	0.8964	0.4066	1
100,027	0	0.9516	0.1812	1
100,028	1	0.9843	0.062	1
100,029	1	0.9807	0.0526	1
100,030	0	1	0	1
100,031	0	1	0.0002	1
100,032	0	0.97	0.1184	1
100,033	0	0.9954	0.0559	1
100,034	1	1	0	1

## 15.4 Regressione

Oracle Data Mining può essere utilizzato per predire i valori di un valore continuo.

Per utilizzare tale caratteristica, selezionare Build... dal menu Activity, quindi scegliere Regression come funzione. Support Vector Machine è il solo algoritmo implementato. Selezionare Next. Seguendo i passi del wizard è possibile scegliere la tabella di origine e l'attributo target. Nella pagina finale è possibile cliccare su Advanced Settings per modificare i valori di default.

Support Vector Machine fa distinzione fra errori piccoli e grandi; la differenza tra questi viene definita attraverso il valore epsilon. L'algoritmo calcolerà e ottimizzerà un valore epsilon internamente; tuttavia è possibile fornire dall'esterno tale valore. Provare a diminuire epsilon nel caso in cui vi siano attributi categorici con alta cardinalità.

Sample Outlier Treatment Missing Values Normalize Split **Build** Test Metrics Residual Plot

☒ Enable Step

**Options**

Although the default settings are expected to work well, you may find it worthwhile to alter these settings based on the benefits outlined below.

Kernel function: Gaussian

Tolerance value:   
Range: > 0 and <= 0.1

Do you want to specify the complexity factors?

☐ Yes ☒ No

Complexity factor:   
Range: > 0 or not defined (system calculated)

Do you want Active Learning?

☒ Yes ☐ No

Do you want to specify the epsilon value?

☐ Yes ☒ No

Epsilon value:   
Range: > 0 or not defined (system calculated)

Do you want to specify the standard deviation for gaussian kernel?

☐ Yes ☒ No

Standard deviation:   
Range: > 0 or not defined (system calculated)

Cache size (M):   
Range: > 0

Help OK Cancel

Cliccare OK per ritornare nella pagina finale del wizard e Finish per avviare l'attività.

Support Vector Machine Regression Mining Activity - DEMO\_REGR\_BA1

This activity consists of the recommended steps to build and test a Regression model using the Support Vector Machine algorithm. The input for a step is the output of the previous completed step or, if no previous steps were completed, the input table. Click Run Activity to perform all selected steps.

**Summary**

[Activity Data](#)

Comment:  [Edit](#)

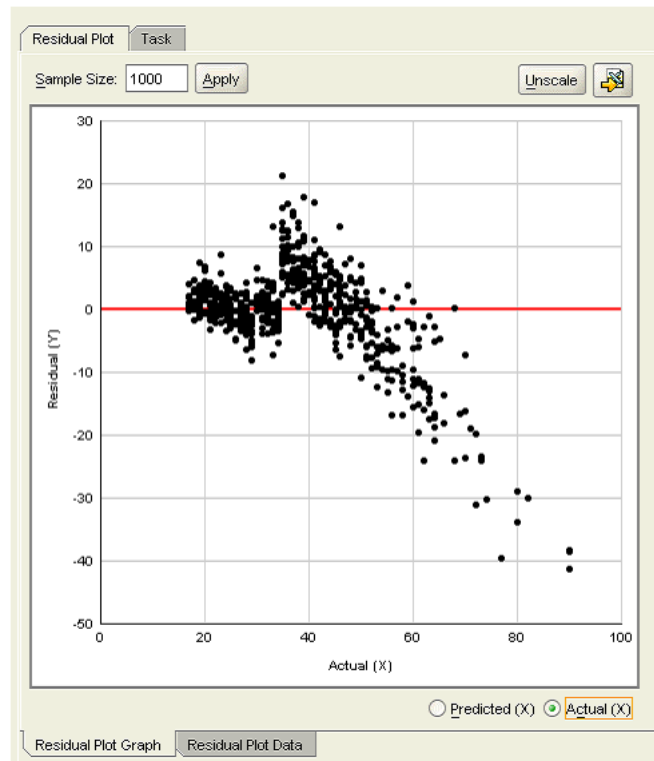
[Run Activity](#)

**Steps:**

<input type="checkbox"/> Sample	Skipped
This step samples the mining data. Although not normally required, this step can be used to sample very large data sets. To complete this step manually, click Custom.	
<a href="#">Options...</a>	<a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Outlier Treatment	Completed
This transformation step handles outliers in mining data. To complete this step manually, click Custom.	
<a href="#">Output Data</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Missing Values	Completed
This transformation step handles missing values in the mining data. To complete this step manually, click Custom.	
<a href="#">Output Data</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Normalize	Completed
This transformation step normalizes the mining data. To complete this step manually, click Custom.	
<a href="#">Output Data</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Split	Completed
This transformation step splits the mining data into build and test data sets. To complete this step manually, click Custom.	
<a href="#">Output Data</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Build	Completed
This step builds the mining model. To complete this step manually, click Custom.	
<a href="#">Build Data</a> <a href="#">Result</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Test Metrics	Completed
This step creates a test metric result. To complete this step manually, click Custom.	
<a href="#">Test Data</a> <a href="#">Result</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>
<input checked="" type="checkbox"/> Residual Plot	Completed
This step creates a Residual Plot result. To complete this step manually, click Custom.	
<a href="#">Residual Data</a> <a href="#">Result</a>	<a href="#">Options...</a> <a href="#">Reset</a> <a href="#">Custom...</a>

Cliccando su Result nella voce Test Metrics è possibile visualizzare le misure di accuratezza ottenute.

Cliccando su Result nella voce Residual Plot è possibile analizzare la differenza fra i valori reali e i valori predetti. Il grafico utilizza i valori reali nell'asse delle x e prefigge l'obiettivo di rispondere alla seguente domanda: "per quale range di valori reali il modello sembra essere accurato?".



Nella figura sopra è possibile visualizzare un cambiamento su Age=35. Una possibile tattica potrebbe essere quella di utilizzare due differenti modelli, uno sotto i 35 anni e l'altro sopra i 35 anni.

Cliccando sulla checkbox Predicted il grafico illustra i valori predetti nell'asse x e si prefigge di rispondere alla domanda: “quali predizioni sono più attendibili?”.

Una volta effettuate l'analisi sperimentale è possibile verificare se il modello è adatto a risolvere il nostro problema. In caso affermativo è possibile applicare il modello a nuovi dati esattamente come fatto nella sezione precedente con la Classificazione.

## 15.5 Clustering

Il processo di clustering viene utilizzato per identificare distinti segmenti di una popolazione e spiegare le caratteristiche comuni all'interno di ciascun segmento.

Oracle Data Mining fornisce due algoritmi di cluster, Enhanced k-means e O-cluster. Discuteremo in dettaglio sul secondo algoritmo.

Selezionare Build... dal menu Activity e selezionare Clustering come funzione e OCluster come algoritmo.

Seguire il wizard per selezionare la tabella su cui si vuole effettuare il clustering, visionare i dati e dare un nome significativo all'attività di clustering.

Nella pagine finale cliccando su Advance Settings è possibile visionare e cambiare le configurazioni di default. In tale pagina, tra le varie impostazioni, è anche possibile cambiare il numero di cluster.

Quando l'attività viene completata, cliccare Result nella voce Build per esaminare il modello.

o-Cluster Mining Activity - DEMO\_OC\_BA1

This activity consists of the recommended steps to build and test a Clustering model using the o-Cluster algorithm. The input for a step is the output of the previous completed step or, if no previous steps were completed, the input table. Click Run Activity to perform all selected steps.

[Summary](#)

[Activity Data](#)

Comment:  [Edit...](#)

[Run Activity](#)

Steps:

☐ Sample Skipped

This step samples the mining data. Although not normally required, this step can be used to sample very large data sets. To complete this step manually, click Custom.

[Options...](#) [Reset](#) [Custom](#)

☒ Outlier Treatment Completed

This transformation step handles outliers in mining data. To complete this step manually, click Custom.

[Output Data](#) [Options...](#) [Reset](#) [Custom](#)

☒ Discretize Completed

This transformation step discretizes the mining data. To complete this step manually, click Custom.

[Output Data](#) [Options...](#) [Reset](#) [Custom](#)

☒ Build Completed

This step builds the mining model. To complete this step manually, click Custom.

[Build Data](#) [Result](#) [Options...](#) [Reset](#) [Custom](#)

I cluster vengono mostrati anche con i sotto-cluster intermedi (similmente ad una struttura ad albero) così è possibile visionare come questi sono stati creati nel processo iterativo.

File Debug Help

Clusters Rules Results Build Settings Task

Leaf Clusters: 10  
Cluster Levels: 6  
Cases: 1,500

Clusters: ☐ Show Leaves Only [Bin](#) [Expand All](#)

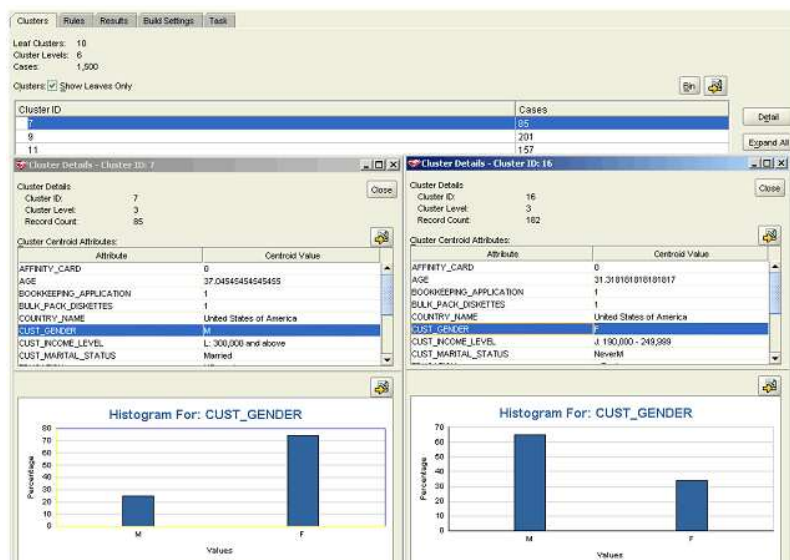
Cluster ID	Cases	Split Rule
1	1,500	
2	859	CUST_INCOME_LEVEL in (A: Below 30,000, B: 30,000 - 49,999, C: 50,000 - 69,999, D: 70,000 - 89,999, E: 90,000 - 109,999, F: 110,000 - 129,999, G: 130,000 - 149,999, H: 150,000 - 169,999, I: 170,000 - 189,999, J: 190,000 - 209,999, K: 210,000 - 229,999, L: 230,000 - 249,999, M: 250,000 - 269,999, N: 270,000 - 289,999, O: 290,000 - 309,999, P: 310,000 - 329,999, Q: 330,000 - 349,999, R: 350,000 - 369,999, S: 370,000 - 389,999, T: 390,000 - 409,999, U: 410,000 - 429,999, V: 430,000 - 449,999, W: 450,000 - 469,999, X: 470,000 - 489,999, Y: 490,000 - 509,999, Z: 510,000 - 529,999, AA: 530,000 - 549,999, AB: 550,000 - 569,999, AC: 570,000 - 589,999, AD: 590,000 - 609,999, AE: 610,000 - 629,999, AF: 630,000 - 649,999, AG: 650,000 - 669,999, AH: 670,000 - 689,999, AI: 690,000 - 709,999, AJ: 710,000 - 729,999, AK: 730,000 - 749,999, AL: 750,000 - 769,999, AM: 770,000 - 789,999, AN: 790,000 - 809,999, AO: 810,000 - 829,999, AP: 830,000 - 849,999, AQ: 850,000 - 869,999, AR: 870,000 - 889,999, AS: 890,000 - 909,999, AT: 910,000 - 929,999, AU: 930,000 - 949,999, AV: 950,000 - 969,999, AW: 970,000 - 989,999, AX: 990,000 - 1,000,000)
3	641	OCCUPATION in (A: Below 30,000, B: 30,000 - 49,999, C: 50,000 - 69,999, D: 70,000 - 89,999, E: 90,000 - 109,999, F: 110,000 - 129,999, G: 130,000 - 149,999, H: 150,000 - 169,999, I: 170,000 - 189,999, J: 190,000 - 209,999, K: 210,000 - 229,999, L: 230,000 - 249,999, M: 250,000 - 269,999, N: 270,000 - 289,999, O: 290,000 - 309,999, P: 310,000 - 329,999, Q: 330,000 - 349,999, R: 350,000 - 369,999, S: 370,000 - 389,999, T: 390,000 - 409,999, U: 410,000 - 429,999, V: 430,000 - 449,999, W: 450,000 - 469,999, X: 470,000 - 489,999, Y: 490,000 - 509,999, Z: 510,000 - 529,999, AA: 530,000 - 549,999, AB: 550,000 - 569,999, AC: 570,000 - 589,999, AD: 590,000 - 609,999, AE: 610,000 - 629,999, AF: 630,000 - 649,999, AG: 650,000 - 669,999, AH: 670,000 - 689,999, AI: 690,000 - 709,999, AJ: 710,000 - 729,999, AK: 730,000 - 749,999, AL: 750,000 - 769,999, AM: 770,000 - 789,999, AN: 790,000 - 809,999, AO: 810,000 - 829,999, AP: 830,000 - 849,999, AQ: 850,000 - 869,999, AR: 870,000 - 889,999, AS: 890,000 - 909,999, AT: 910,000 - 929,999, AU: 930,000 - 949,999, AV: 950,000 - 969,999, AW: 970,000 - 989,999, AX: 990,000 - 1,000,000)
4	720	OCCUPATION in (A: Below 30,000, B: 30,000 - 49,999, C: 50,000 - 69,999, D: 70,000 - 89,999, E: 90,000 - 109,999, F: 110,000 - 129,999, G: 130,000 - 149,999, H: 150,000 - 169,999, I: 170,000 - 189,999, J: 190,000 - 209,999, K: 210,000 - 229,999, L: 230,000 - 249,999, M: 250,000 - 269,999, N: 270,000 - 289,999, O: 290,000 - 309,999, P: 310,000 - 329,999, Q: 330,000 - 349,999, R: 350,000 - 369,999, S: 370,000 - 389,999, T: 390,000 - 409,999, U: 410,000 - 429,999, V: 430,000 - 449,999, W: 450,000 - 469,999, X: 470,000 - 489,999, Y: 490,000 - 509,999, Z: 510,000 - 529,999, AA: 530,000 - 549,999, AB: 550,000 - 569,999, AC: 570,000 - 589,999, AD: 590,000 - 609,999, AE: 610,000 - 629,999, AF: 630,000 - 649,999, AG: 650,000 - 669,999, AH: 670,000 - 689,999, AI: 690,000 - 709,999, AJ: 710,000 - 729,999, AK: 730,000 - 749,999, AL: 750,000 - 769,999, AM: 770,000 - 789,999, AN: 790,000 - 809,999, AO: 810,000 - 829,999, AP: 830,000 - 849,999, AQ: 850,000 - 869,999, AR: 870,000 - 889,999, AS: 890,000 - 909,999, AT: 910,000 - 929,999, AU: 930,000 - 949,999, AV: 950,000 - 969,999, AW: 970,000 - 989,999, AX: 990,000 - 1,000,000)
5	154	
6	596	OCCUPATION in (A: Below 30,000, B: 30,000 - 49,999, C: 50,000 - 69,999, D: 70,000 - 89,999, E: 90,000 - 109,999, F: 110,000 - 129,999, G: 130,000 - 149,999, H: 150,000 - 169,999, I: 170,000 - 189,999, J: 190,000 - 209,999, K: 210,000 - 229,999, L: 230,000 - 249,999, M: 250,000 - 269,999, N: 270,000 - 289,999, O: 290,000 - 309,999, P: 310,000 - 329,999, Q: 330,000 - 349,999, R: 350,000 - 369,999, S: 370,000 - 389,999, T: 390,000 - 409,999, U: 410,000 - 429,999, V: 430,000 - 449,999, W: 450,000 - 469,999, X: 470,000 - 489,999, Y: 490,000 - 509,999, Z: 510,000 - 529,999, AA: 530,000 - 549,999, AB: 550,000 - 569,999, AC: 570,000 - 589,999, AD: 590,000 - 609,999, AE: 610,000 - 629,999, AF: 630,000 - 649,999, AG: 650,000 - 669,999, AH: 670,000 - 689,999, AI: 690,000 - 709,999, AJ: 710,000 - 729,999, AK: 730,000 - 749,999, AL: 750,000 - 769,999, AM: 770,000 - 789,999, AN: 790,000 - 809,999, AO: 810,000 - 829,999, AP: 830,000 - 849,999, AQ: 850,000 - 869,999, AR: 870,000 - 889,999, AS: 890,000 - 909,999, AT: 910,000 - 929,999, AU: 930,000 - 949,999, AV: 950,000 - 969,999, AW: 970,000 - 989,999, AX: 990,000 - 1,000,000)
7	223	OCCUPATION equal (Crafts)
8	94	
9	129	
10	373	
11	221	OCCUPATION in (House-s, Machine, Other, Prof)
12	152	
13	139	
14	635	OCCUPATION in (A: Below 30,000, B: 30,000 - 49,999, C: 50,000 - 69,999, D: 70,000 - 89,999, E: 90,000 - 109,999, F: 110,000 - 129,999, G: 130,000 - 149,999, H: 150,000 - 169,999, I: 170,000 - 189,999, J: 190,000 - 209,999, K: 210,000 - 229,999, L: 230,000 - 249,999, M: 250,000 - 269,999, N: 270,000 - 289,999, O: 290,000 - 309,999, P: 310,000 - 329,999, Q: 330,000 - 349,999, R: 350,000 - 369,999, S: 370,000 - 389,999, T: 390,000 - 409,999, U: 410,000 - 429,999, V: 430,000 - 449,999, W: 450,000 - 469,999, X: 470,000 - 489,999, Y: 490,000 - 509,999, Z: 510,000 - 529,999, AA: 530,000 - 549,999, AB: 550,000 - 569,999, AC: 570,000 - 589,999, AD: 590,000 - 609,999, AE: 610,000 - 629,999, AF: 630,000 - 649,999, AG: 650,000 - 669,999, AH: 670,000 - 689,999, AI: 690,000 - 709,999, AJ: 710,000 - 729,999, AK: 730,000 - 749,999, AL: 750,000 - 769,999, AM: 770,000 - 789,999, AN: 790,000 - 809,999, AO: 810,000 - 829,999, AP: 830,000 - 849,999, AQ: 850,000 - 869,999, AR: 870,000 - 889,999, AS: 890,000 - 909,999, AT: 910,000 - 929,999, AU: 930,000 - 949,999, AV: 950,000 - 969,999, AW: 970,000 - 989,999, AX: 990,000 - 1,000,000)
15	329	OCCUPATION in (A: Below 30,000, B: 30,000 - 49,999, C: 50,000 - 69,999, D: 70,000 - 89,999, E: 90,000 - 109,999, F: 110,000 - 129,999, G: 130,000 - 149,999, H: 150,000 - 169,999, I: 170,000 - 189,999, J: 190,000 - 209,999, K: 210,000 - 229,999, L: 230,000 - 249,999, M: 250,000 - 269,999, N: 270,000 - 289,999, O: 290,000 - 309,999, P: 310,000 - 329,999, Q: 330,000 - 349,999, R: 350,000 - 369,999, S: 370,000 - 389,999, T: 390,000 - 409,999, U: 410,000 - 429,999, V: 430,000 - 449,999, W: 450,000 - 469,999, X: 470,000 - 489,999, Y: 490,000 - 509,999, Z: 510,000 - 529,999, AA: 530,000 - 549,999, AB: 550,000 - 569,999, AC: 570,000 - 589,999, AD: 590,000 - 609,999, AE: 610,000 - 629,999, AF: 630,000 - 649,999, AG: 650,000 - 669,999, AH: 670,000 - 689,999, AI: 690,000 - 709,999, AJ: 710,000 - 729,999, AK: 730,000 - 749,999, AL: 750,000 - 769,999, AM: 770,000 - 789,999, AN: 790,000 - 809,999, AO: 810,000 - 829,999, AP: 830,000 - 849,999, AQ: 850,000 - 869,999, AR: 870,000 - 889,999, AS: 890,000 - 909,999, AT: 910,000 - 929,999, AU: 930,000 - 949,999, AV: 950,000 - 969,999, AW: 970,000 - 989,999, AX: 990,000 - 1,000,000)
16	164	
17	142	
18	369	OCCUPATION in (Machine, Other, Prof, Protec)
19	186	
20	123	

[Detail](#) [Expand All](#) [Collapse All](#)

Per visualizzare solamente i cluster finali occorre spuntare la checkbox Show Leaves Only.

Selezionando un cluster e cliccando su dettagli è possibile vedere l'istogramma degli attributi per i membri del cluster. È possibile visualizzare anche gli istogrammi di più di un cluster alla volta aprendo più finestre, come visualizzato nella seguente figura.





Cliccando sulla voce Rules è possibile visualizzare le regole che definiscono ciascun cluster.

The screenshot displays the 'Rules' tab in the Oracle Data Mining interface. It shows a table of rules for Cluster ID 5. The table has three columns: Cluster ID, Confidence (%), and Support Count.

Cluster ID	Confidence (%)	Support Count
5	0.8129495403	113
6	0.8396451813	103
12	0.8325791895	194
13	0.7831578947	118
14	0.8097826087	149
15	0.8028169014	114
16	0.8494823856	150
17	0.7225572335	95
18	0.7978722404	75
19	0.7906976744	182

Below the table, the 'Rule Detail' for Cluster ID 5 is shown:

```

IF
HOUSEHOLD_SIZE in (1,2,3,4,5,6) and OCCUPATION in ('Cleric', 'Crafts', 'Exec.', 'Handler', 'Machine', 'Other', 'Prof.', 'Sales', 'Transp.') and OS_SOC_SET_KANAL = 0.0 and Y_BOX_OAKES in (0,1,2)
THEN
cluster equal 5
Confidence (%)=0.812949540377698
Support =113
  
```

## 15.6 Regole associative

L'algoritmo implementato da Oracle Data Mining per supportare le regole associative richiede che i dati siano nel formato in cui ciascun articolo sia rappresentato in un'unica riga, come mostra la seguente figura.

PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_S...
13	997	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,660	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,762	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,843	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	1,948	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,273	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,380	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,683	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	2,665	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	4,663	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	5,203	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	5,321	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	5,590	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	6,277	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	6,859	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	8,540	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	9,076	1998-01-10 00:00:00.0	3	999	1	1,232.16003...
13	12,099	1998-01-10 00:00:00.0	3	999	1	1,232.16003...

Selezionare Build... dal menu Activity e scegliere Association Rules come funzione, l'unico algoritmo che implementa le regole associative in Oracle Data Mining è Apriori.

Seguendo i passi del wizard sarà possibile selezionare l'identificatore dell'articolo su cui si vuole prestare attenzione; le colonne utilizzate per raggruppare i dati e il nome dell'attività.

Cliccare Advance Setting nella pagina finale del wizard per visionare ed, eventualmente, modificare i parametri disponibili per le regole associative. Attraverso tale finestra è possibile modificare il supporto minimo e la confidenza minima affinché la regola sia ritenuta valida. È anche possibile modificare il numero massimo di attributi che compongono una regola.

Cliccare OK per ritornare nell'ultima pagina del wizard e cliccare su Finish per avviare l'attività.

Name: DEMO\_AR\_BA1

Type: Association Rules Mining Activity

Input Table: SHSALES

Comment:  [Edit...](#)

[Mining Data](#)

Activity Steps: [Run Activity](#)

☐ Sample [Skipped](#)

This step samples the mining data. Although not normally required, this step can be used to sample very large data sets. To complete this step manually, click Run.

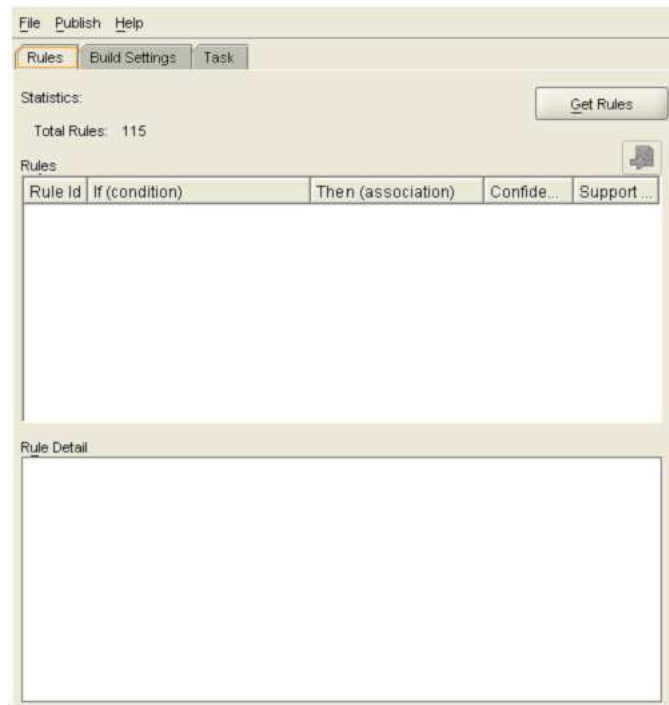
[Options...](#) [Reset](#) [Run](#)

☒ Build [Completed](#)

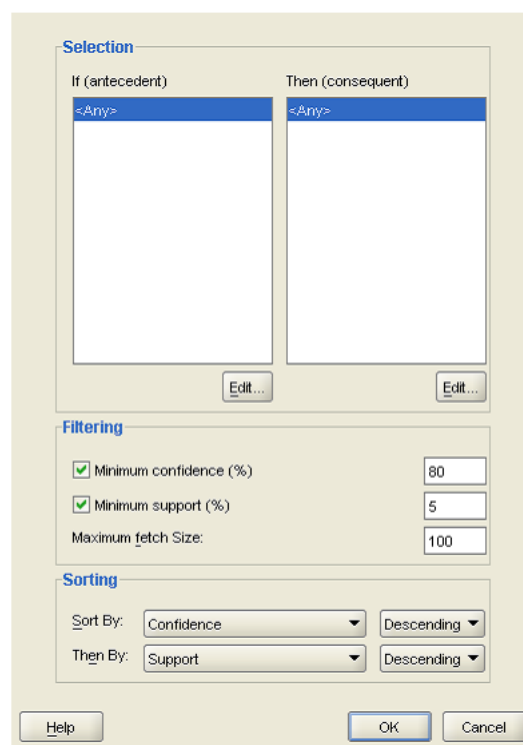
This step builds the mining model. To complete this step manually, click Run.

[Build Data](#) [Result](#) [Options...](#) [Reset](#) [Run](#)

Nell'esempio mostrato in figura sono state trovate 115 regole. Inizialmente non viene visualizzata nessuna regola, visto che molte di queste potrebbero essere non interessanti in quanto potremmo essere interessati solamente a regole contenenti soltanto particolari prodotti.



Per ottenere le regole occorre cliccare su Get Rules.



Dopo aver scelto i criteri preferiti, cliccare OK. Apparirà una lista delle regole come mostra la figura.

File Publish Help				
Rules Build Settings Task				
Statistics				
Total Rules: 115				
Get Rules				
Rules				
Rule ID	If (condition)	Then (association)	Confidence (%)	Support (%)
101	CD-R, Professional Grade, Pack of 10= 1 AND Music CD-R= 1	CD-R with Jewel Cases, pACK OF 12= 1	94.1794	5.4488
99	Music CD-R= 1 AND CD-RW, High Speed Pack of 5= 1	CD-R with Jewel Cases, pACK OF 12= 1	93.1260	5.4695
104	CD-R, Professional Grade, Pack of 10= 1 AND CD-RW, High Speed Pack of 5= 1	CD-R with Jewel Cases, pACK OF 12= 1	90.6486	6.1220
108	External 101-key keyboard= 1 AND SIMM 10MB PCMCIA card= 1	SIMM 8MB PCMCIA card= 1	90.3387	5.9250
96	CD-R, Professional Grade, Pack of 10= 1 AND Music CD-R= 1	CD-RW, High Speed Pack of 5= 1	90.1340	5.2145
112	PCMCIA modem/fax 19200 baud= 1 AND Keyboard Wrist Rest= 1	Mouse Pad= 1	89.6376	5.1679
95	Music CD-R= 1 AND CD-RW, High Speed Pack of 5= 1	CD-R, Professional Grade, Pack of 10= 1	89.3571	5.2145
99	CD-R with Jewel Cases, pACK OF 12= 1 AND Music CD-R= 1	CD-RW, High Speed Pack of 5= 1	86.1466	5.4695
102	CD-R with Jewel Cases, pACK OF 12= 1 AND Music CD-R= 1	CD-R, Professional Grade, Pack of 10= 1	85.8165	5.4486
106	CD-R with Jewel Cases, pACK OF 12= 1 AND CD-R, Professional Grade, Pack of 10= 1	CD-RW, High Speed Pack of 5= 1	85.6682	6.1220
109	SIMM 16MB PCMCIA card= 1 AND SIMM 8MB PCMCIA card= 1	External 101-key keyboard= 1	85.4767	5.9250
105	CD-R with Jewel Cases, pACK OF 12= 1 AND CD-RW, High Speed Pack of 5= 1	CD-R, Professional Grade, Pack of 10= 1	84.1221	6.1220
11	Music CD-R= 1	CD-R with Jewel Cases, pACK OF 12= 1	84.0703	6.3491
92	OS Documentation Set- French= 1	OS Documentation Set- English= 1	83.7690	6.0284
3	3.12T Bus Attachment, Bus of 100= 1	3.12T Bus Attachment, Bus of 60= 1	83.4836	6.3386
4				
Rule Detail				
IF				
CD-R, Professional Grade, Pack of 10= 1 AND Music CD-R= 1				
THEN				
CD-R with Jewel Cases, pACK OF 12= 1				
Confidence (%)=94.1794469268867				
Support (%)=5.44855010026635				