

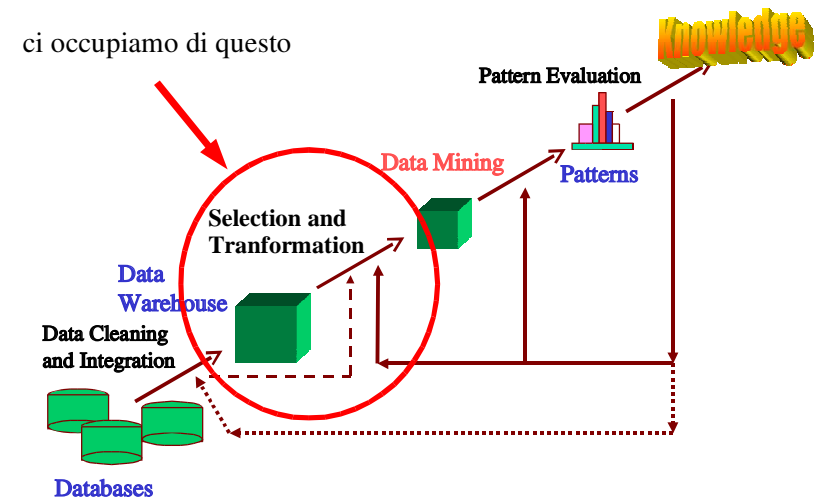
Pre-elaborazione dei dati

Gianluca Amato

Corso di Laurea Specialistica in Economia Informatica
Università "G. D'Annunzio" di Chieti-Pescara
anno accademico 2004-2005

1

Knowledge Discovery in Databases



2

Istanze e attributi

Input a un sistema di data mining

- Mentre nei sistemi OLAP la forma preferita di dati è il cubo multidimensionale, per i sistemi di data mining la visione tabellare è di solito più conveniente.
- L'input corrisponde essenzialmente a una tabella di un database relazionale:
 - ogni riga della tabella è una **istanza** (o **esempio**, o **tupla**)
 - ogni colonna è un **attributo**
- L'input è dunque un insieme di istanze, ognuna delle quali è un esempio indipendente dell'informazione che si vuole apprendere.

3

4

Un esempio di Input

	lun. sepal	larg. sepal	lun. petalo	larg. petalo	tipo
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3	1.4	0.2	Iris setosa
3	4.7	3.2	1.3	0.2	Iris setosa
4	4.6	3.1	1.5	0.2	Iris setosa
5	5	3.6	1.4	0.2	Iris setosa
..					
51	7	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
53	6.9	3.1	4.9	1.5	Iris versicolor
..					
103	7.1	3	5.9	2.1	Iris virginica
104	6.3	2.9	5.6	1.8	Iris virginica
105	6.5	3	5.8	2.2	Iris virginica

5

Attributi (1)

- Gli attributi possono essere distinti secondo il “**livello di misura**”
 - **nominali**
 - ogni valore è un simbolo distinto
 - l'unica operazione permessa è decidere se due valori sono uguali
 - ad esempio l'attributo **tipo** per il data set degli iris è un attributo nominale che assume tre possibili valori
 - **ordinali**
 - come gli attributi nominali, ma in più c'è un ordine
 - è possibile confrontare due valori con tutti gli operatori relazionali (<, >, = e derivati)
 - ad esempio, l'attributo **temperatura** può assumere i valori freddo, tiepido, caldo con $\text{freddo} < \text{tiepido} < \text{caldo}$.
 - spesso si usano i numeri interi per rappresentare i valori ordinali, dato che per essi è definito un ordinamento standard.
 - in questo caso, si potrebbe usare 0 per il freddo, 1 per il tiepido, 2 per il caldo.
 - la distinzione tra attributi nominali ed ordinali non è sempre chiara.

6

Attributi (2)

- **intervallo**
 - assumono valori ordinati e ottenuti da precise unità di misura
 - esiste il concetto di distanza, per cui è possibile sottrarre due valori
 - le altre operazioni aritmetiche come somma e prodotto non hanno senso, e non esiste un valore **zero** significativo.
 - ad esempio la **temperatura**, quando espressa in gradi Celsius o l'attributo **anno**.
- **ratio**
 - assumono valori ordinati e ottenuti da precise unità di misura, per cui esiste un valore **zero** ben definito.
 - tutte le operazioni aritmetiche hanno senso.
 - ad esempio la temperatura, quando espressa in gradi Kelvin.

7

Attributi (3)

- In pratica, la maggior parte delle volte gli algoritmi di data mining trattano solo due classi di attributi:
 - **nominali** (chiamati anche **categoriali** o **discreti**)
 - **numerici**: corrispondono ai tipi ordinale, intervallo o ratio a seconda del tipo di algoritmo
 - assumono un qualunque valore numerico
 - bisogna stare attenti al fatto che l'algoritmo non faccia delle operazioni che non hanno senso sul tipo di dato in questione

8

Metadati

- I sistemi di data mining possono usare altre informazioni oltre al tipo di attributi:
 - **informazioni dimensionali**, in modo da non confrontare dati espressi con unità di misura diverse (cosa vuol dire che 3 Km è minore di 5 Litri?)
 - **ordinamenti circolari**: indicare se un attributo è soggetto a particolare circolarità dei dati
 - gli angoli vanno da 0 a 360° (o da 0 a 2π) e poi ricominciano da 0.
 - ci si può riferire allo “stesso giorno nella prossima settimana” o alla “prossima domenica”
 - **gerarchie di concetti**: alcuni attributi possono essere trattati a vari livelli di dettaglio
- Tutte queste informazioni prendono il nome di **metadati** e consentono di aumentare l'efficienza del sistema di data mining.

9

Perché pre-elaborare i dati?

10

Pre-elaborazione dei dati (1)

- I dati nel mondo reale sono sporchi:
 - **incompleti**: manca il valore di alcuni attributi, o mancano del tutto alcuni attributi interessanti.
 - **inaccurati**: contengono valori errati o che si discostano sensibilmente da valori attesi.
 - Ad esempio, nel campo età di un impiegato si trova il valore di 120 anni.
 - **inconsistenti**: ad esempio, due filiali dello stesso negozio usano codici diversi per rappresentare la stessa merce venduta.
- Queste inesattezze non influivano sullo scopo iniziale per cui i dati sono stati raccolti, e vengono scoperti solo ora.
- **GIGO**: garbage in – garbage out
 - se i dati in input non sono di buona qualità, neanche le analisi basate su di questi lo possono essere!

11

Pre-elaborazione dei dati (2)

- Principali tecniche nella fase di pre-elaborazione dei dati:
 - **data cleaning** (pulitura dei dati)
riempire i campi con i valori mancanti, “lisciare” i dati rumorosi, rimuovere i valori non realistici.
 - **data integration** (integrazione dei dati)
integrare dati provenienti da database multipli risolvendo le inconsistenze.
 - **data transformation** (trasformazione dei dati)
preparare i dati per l'uso con alcuni particolari algoritmi di analisi.
 - **data reduction** (riduzione dei dati)
ridurre la mole dei dati in input, ma senza compromettere la validità delle analisi (campionamento, astrazione dei dati con le gerarchie di concetti, ...)

12

Data Cleaning

13

Data Cleaning

- Le attività eseguite durante il passo di **data cleaning** sono:
 - riempire gli attributi che hanno valori mancanti
 - identificare gli **outliers** (dati molto diversi dai valori attesi)
 - “lisciare” i dati rumorosi
 - correggere le inconsistenze
- Alcuni algoritmi di analisi hanno dei meccanismi per gestire dati con valori mancanti o con outliers.
 - essi operano però senza conoscenza del dominio applicativo
- I risultati migliori si ottengono con una pulizia a priori dei dati, con l'aiuto di esperti del dominio applicativo.

14

Dati mancanti (1)

- Varie ragioni per cui i dati mancano
 - malfunzionamento di qualche apparecchiatura.
 - dati inconsistenti con altri e quindi cancellati in una fase precedente.
 - dati non immessi.
- Mancanze casuali o no?
 - se un valore non è presente perché un determinato test non è stato eseguito in maniera deliberata, allora la presenza di un attributo mancante può veicolare una grossa mole di informazione.
 - le persone che studiano i database di natura medica hanno scoperto che spesso è possibile effettuare una diagnosi semplicemente guardando quali sono i test a cui è stato sottoposto

15

Dati mancanti (2)

- I possibili approcci quando si hanno dati con valori mancanti:
 - **ignorare le istanze con valori mancanti**
 - non molto efficace, in particolare se la percentuale di tuple con dati mancanti è alta.
 - si usa spesso quando il dato che manca è la classe in un problema di classificazione
 - **riempire i valori mancanti manualmente**
 - in generale è noioso, e potrebbe essere non fattibile
 - **usare un valore costante** come “Unknown” oppure 0 (a seconda del tipo di dati).
 - potrebbe alterare il funzionamento dell'algoritmo di analisi, meglio allora ricorrere ad algoritmi che gestiscono la possibilità di dati mancanti
 - è però utile se la mancanza di dati ha un significato particolare di cui tener conto

16

Dati mancanti (3)

- Altri possibili approcci:
 - usare la **media dell'attributo** al posto dei valori mancanti
 - per problemi di classificazione, usare la media dell'attributo per tutti i campioni della stessa classe
 - è una versione perfezionata del metodo della media per problemi di classificazione.
 - **predirre** il valore dell'attributo mancante sulla base degli altri attributi noti
 - la predizione può avvenire usando regressione lineare, alberi di classificazione, etc..
 - si usano algoritmi di data mining per preparare i dati in input ad altri algoritmi di data mining.

17

Dati inaccurati (1)

- Cause specifiche delle inesattezze
 - errori tipografici in attributi nominali: coca cola diventa coccola
 - il sistema di data mining pensa si tratti di prodotti diversi
 - sinonimi: pepsi cola e pepsi
 - errori tipografici o di misura in attributi numerici
 - alcuni valori sono chiaramente poco sensati, e possono essere facilmente riconosciuti
 - ma altri errori possono essere più subdoli
 - errori deliberati: durante un sondaggio, l'intervistato può fornire un CAP falso
 - errori causati da sistemi di input automatizzati
 - se il sistema insiste per un codice ZIP (come il CAP ma negli USA) e l'utente non lo possiede?

18

Dati inaccurati (2)

- Occorre imparare a conoscere i propri dati!
 - capire il significato di tutti i campi
 - individuare gli errori che sono stati commessi
- Semplici programmi di visualizzazione grafica consentono di identificare rapidamente dei problemi:
 - attributi nominali: istogrammi
 - la distribuzione è consistente con ciò che ci si aspetta?
 - attributi numerici: grafici
 - c'è qualche dato ovviamente sbagliato?
- Vediamo due tecniche tipiche:
 - **binning**: per “lisciare” i dati **rumorosi**
 - **clustering**: per riconoscere gli outliers

19

Dati rumorosi e binning

- Per **rumore** si intende un errore casuale su una variabile misurata (tipicamente numerica)
 - è una delle possibili cause di dati inaccurati
- Il rumore può essere dovuto a
 - apparati di misura difettosi
 - problemi con le procedure di ingresso dati
 - problemi di trasmissione
 - limitazioni tecnologiche
- Il binning è una tecnica per ridurre la variabilità (e quindi il rumore) nei dati

20

Equi-Depth Binning (1)

- si considerano tutti i possibili valori (con ripetizioni) assunti dall'attributo e li si ordina
 - chiamiamo a_i con $i \in [1..N]$ i dati input, già ordinati
- si fissa un valore d per la **profondità (depth)** e si divide l'intervallo $[a_0, a_N]$ in intervalli (**bin**) consecutivi disgiunti di ampiezza più o meno uguale ad d
 - quindi ci saranno circa N/d intervalli
 - chiamiamoli I_0, \dots, I_m
 - la corrispondenza tra i dati e gli intervalli è data da una funzione v tale che $a_i \in I_{v(i)}$
- ora sostituiamo ad ogni a_i un valore derivato dal corrispondente intervallo

21

Equi-Depth Binning (2)

- varie possibilità per questa sostituzione
 - **smoothing by bin means**
 - si sostituisce ad a_i la media del corrispondente intervallo
 - $a_i \rightarrow \text{media } I_{v(i)}$
 - **smoothing by bin medians**
 - si sostituisce ad a_i la mediana del corrispondente intervallo
 - $a_i \rightarrow \text{mediana } I_{v(i)}$
 - **smoothing by bin boundaries**
 - si sostituisce ad a_i uno dei due estremi dell'intervallo corrispondente, in particolare quello più vicino
 - se $a_i - \min I_{v(i)} < \max I_{v(i)} - a_i$, allora $a_i \rightarrow \min I_{v(i)}$
altrimenti $a_i \rightarrow \max I_{v(i)}$

22

Equi-Depth Binning (3)

- Prezzi (in euro): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
 - Partizionamento in intervalli di 4 elementi ($d=4$)
 - Bin 1: 4, 8, 9, 15
 - Bin 2: 21, 21, 24, 25
 - Bin 3: 26, 28, 29, 34
 - Smoothing by bin means:
 - Bin 1: 9, 9, 9, 9
 - Bin 2: 23, 23, 23, 23
 - Bin 3: 29, 29, 29, 29
 - Smoothing by bin boundaries
 - Bin 1: 4, 4, 4, 15
 - Bin 2: 21, 21, 25, 25
 - Bin 3: 26, 26, 26, 34
- Non sempre è possibile avere intervalli di esattamente d elementi.

23

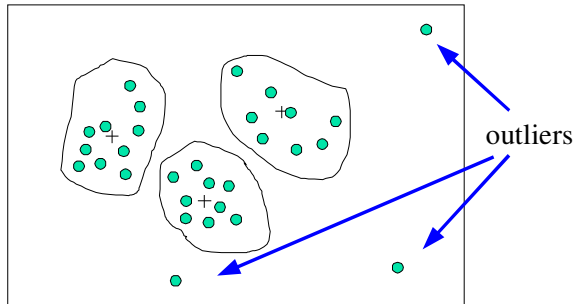
Equi-Width Binning

- simile al Equi-Depth Binning, ma gli intervalli sono ottenuti in modo da avere più o meno tutti la stessa **ampiezza (width)**
 - ovvero, se gli intervalli che otteniamo sono I_0, \dots, I_m , il valore $\max I_i - \min I_i$ è più o meno costante, al variare di $i \in [0..m]$
- Con i dati di prima, e una ampiezza per ogni intervallo più o meno fissata a 10, otteniamo i seguenti bin:
 - Bin 1: 4, 8, 9, 1
 - Bin 2: 21, 21, 24, 25, 26, 28, 29, 34

24

Clustering

- Con questo metodo è possibile riconoscere gli **outliers**.
 - si dividono i possibili valori degli attributi da pulire in gruppi;
 - eventuali valori che non ricadono in nessun gruppo sono degli outliers.
- Anche in questo caso usiamo algoritmi di data-mining come preparazione per altri algoritmi di data-mining



25

Data Integration

26

Data Integration (1)

- Si tratta di combinare dati provenienti da sorgenti diverse
- Primo aspetto: **schema integration**
 - integrazione dei metadati (ovvero dello schema relazionale) tra più database
 - il problema è capire che relazione c'è tra entità provenienti da diverse sorgenti
 - ad esempio, come fa il progettista a capire se l'attributo `customer_id` di un database e `cust_number` in un altro si riferiscono alla stessa entità
- Secondo aspetto: **integrazione dei dati vera propria**
 - ammesso di aver trovato che le tabelle `customer` di due diversi database si riferiscono alla stessa entità, come si fa a mettere assieme in una unica tabella le informazioni?
 - possibilità informazioni discordanti (errori, unità di misura diverse, etc..)

27

Data Integration (2)

- Terzo aspetto: **ridondanza**
 - alcuni attributi possono essere ricavati (perfettamente o in parte) da altri
 - inconsistenza negli attributi tra due database diversi può provocare ridondanza
 - ad esempio, non si capisce che i campi “categoria merceologica” di una tabella è “tipo prodotto” di un'altra si riferiscono allo stesso tipo di informazione.
 - quando si mettono assieme i dati, si creano due campi diversi nella tabella integrata.
 - i valori dei due campi sono strettamente collegati
 - per gli attributi numerici, è possibile provare a scoprire se due attributi sono tra loro ridondanti usando una **analisi di correlazione** (che vedremo in futuro)

28

Data Trasformation

29

Data Transformation

- I dati sono consolidati e trasformati in forme più appropriate per le analisi. Varie possibilità sono
 - **smoothing** (lisciamento) rimuovere i rumori nei dati (binning, clustering, regressione, ...)
 - già visti nella fase di Data Cleaning
 - **aggregazione**: costruire dati aggregati prima dell'analisi
 - nei sistemi OLAM, corrisponde a scegliere il cuboide appropriato
 - eventualmente usando le gerarchie di concetti
 - **costruzione degli attributi**: costruire nuovi attributi a partire da quelli presenti per aiutare l' algoritmo di analisi
 - per esempio, si aggiunge l'attributo derivato **area** come prodotto degli attributi **altezza** e **larghezza**.
 - **normalizzazione**: modificare la scala dei dati in modo che cadano in intervalli stabiliti (ad esempio da -1 ad 1)

30

Normalizzazione (1)

- Spesso gli attributi assumono valori in intervalli di ampiezza diversa.
 - può compromettere il funzionamento di alcune analisi
 - necessità di **normalizzare** i dati
- **min-max normalization**: si riscalda l'attributo A in modo che i nuovi valori cadano tra new_min_A e new_max_A .

$$v' = \frac{v - \min_A}{\max_A - \min_A} (new_max_A - new_min_A) + new_min_A$$

- il minimo e il massimo effettivo dell'attributo A potrebbero essere ignoti.
 - si verifica un superamento dei nuovi limiti se successivamente appare un dato con un valore di A oltre l'intervallo originario.
- molto influenzato dagli outliers

31

Normalizzazione (2)

- **z-score normalization** (anche **z-mean normalization**)

$$v' = \frac{v - \text{media}}{\sigma_A}$$

← media

← deviazione standard

- utile quando non si conosce minimo e massimo per A
- i valori normalizzati non hanno un minimo e un massimo fissato
- non influenzato dagli outliers (o almeno non altrettanto del metodo precedente)

32

Normalizzazione (3)

- **normalization by decimal scaling:**

- una variante del metodo min-max
- restringe i valori tra -1 ed 1 modificando la posizione della virgola

$$v' = \frac{v}{10^j} \quad \text{dove } j \text{ è il più piccolo intero tale che } \text{Max}(|v'|) < 1$$

- ad esempio, se l'attributo A varia da -986 a 917, per normalizzare dividiamo tutto per 1000. I nuovi valori andranno da -0.986 a 0.917.
 - il passaggio da valori di base a valori normalizzati è molto semplice

33

Data Reduction

34

Data Reduction

- I data warehouse possono memorizzare dati dell'ordine di terabyte: le analisi sono troppo complesse.
- Necessità di effettuare una **riduzione dei dati**
 - ottenere una rappresentazione ridotta dei dati con una occupazione molto inferiore di memoria ma che produce gli stessi (o comunque simili) risultati analitici.
- Varie strategie
 - **aggregazione**
 - usare un cuboide a più alto livello di aggregazione, purché sufficiente per il compito di analisi che dobbiamo svolgere.
 - **riduzione della dimensionalità** (dimensionality reduction)
 - **compressione dei dati**
 - **riduzione della numerosità** (numerosity reduction)
 - **discretizzazione e generazione delle gerarchie di concetto**

35

Riduzione della dimensionalità (1)

- Selezionare un insieme minimo di attributi che descrivano in maniera adeguata i dati in ingresso
 - ad esempio, eliminare gli attributi irrilevanti come può essere una chiave primaria
 -
- Può essere effettuata da un esperto del settore analizzato, ma qui parleremo di metodi automatici
- Serve una “**misura**” della **bontà** di un insieme di attributi, in modo che si possa scegliere l'insieme migliore.
- Una ricerca esaustiva è spesso impossibile:
 - se ho d attributi in totale, ci sono 2^d possibili sottoinsiemi
 - si usano quindi algoritmi euristici

36

Riduzione della dimensionalità (2)

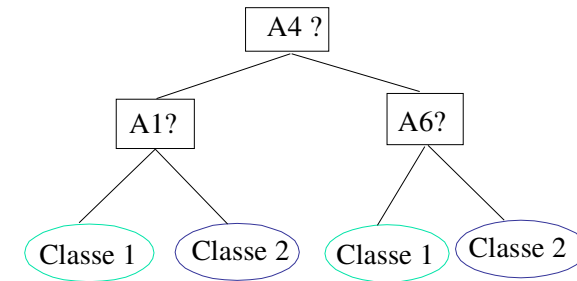
- Possibili algoritmi euristici
 - step-wise **forward selection**
 - parto da un insieme vuoto di attributi
 - ad ogni passo **aggiungo** l'attributo che massimizza la qualità dell'insieme risultante
 - Insieme di attributi : {A1, A2, A3, A4, A5, A6 }
 - Insiemi ridotti: {} → {A1} → {A1,A6} → {A1,A4,A6}
 - step-wise **backward selection**
 - parto da tutti gli attributi
 - ad ogni passo **tolgo** l'attributo che massimizza la qualità dell'insieme risultante
 - Insieme di attributi : {A1, A2, A3, A4, A5, A6 }
 - Insiemi ridotti: {A1, A2, A3, A4, A5, A6} → {A1, A3, A4, A5, A6} → {A1, A4, A5, A6} → {A1, A4, A6}
- combinazione di forward e backward selection

37

Riduzione della dimensionalità (3)

- Posso anche usare gli alberi di decisione
 - costruisco un albero di decisione e mantengo solo gli attributi che appaiono nell'albero

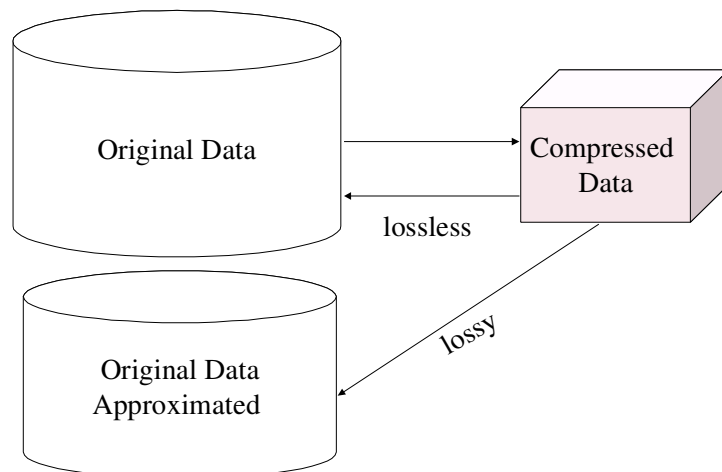
Insieme iniziale di attributi:
{A1, A2, A3, A4, A5, A6}



Insieme ridotto: {A1, A4, A6}

38

Compressione dei dati (1)



39

Compressione dei dati (2)

- Un insieme di tecniche che consentono di ridurre la dimensioni dei dati
 - **lossless** (senza perdita)
 - una enorme letteratura per quello che riguarda la compressione delle stringhe
 - esempio: formati ZIP, RAR, Gzip, BZ2 per file generici
 - esempio: formati GIF per immagini
 - **lossy** (con perdita)
 - tipicamente usate per contenuti multimediali
 - esempio: mp3, Ogg Vorbis (audio), MPEG (video), JPEG (immagini)

40

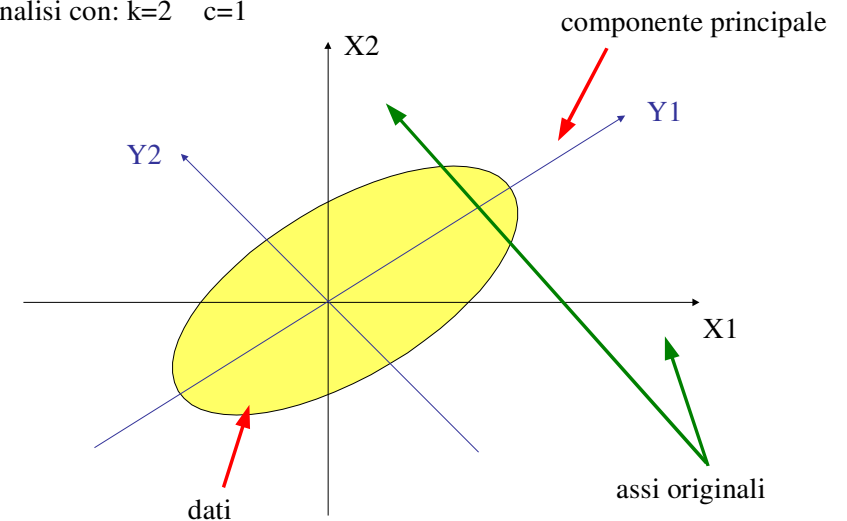
Compressione dei dati (3)

- **Analisi delle componenti principali**
 - dati N vettori in k dimensioni, trovare $c \leq k$ vettori **ortonormali** (le **componenti principali**) che possono essere usati per rappresentare i dati
 - l'insieme di dati originale viene ridotto ad un insieme di N vettori in c dimensioni
 - i nuovi dati sono combinazioni lineari delle c componenti principali
 - le componenti principali sono ordinati per “significatività”
 - i più significativi sono quelli che mostrano maggiore variabilità nei dati
 - i meno significativi sono quelli che mostrano minore variabilità nei dati
 - funziona solo su dati numerici

41

Compressione dei dati (4)

analisi con: $k=2$ $c=1$



42

Riduzione della numerosità

- Ridurre la mole dei dati scegliendo una rappresentazione diversa
- **Metodi parametrici**
 - assumere che i dati soddisfino un dato modello, stimare i parametri del modello e usare questi ultimi invece dei parametri originali
 - ad esempio, stimare una serie di numeri usando la regressione lineare..
 - od un insieme di numeri con una distribuzione gaussiana
 - assumiamo che siano ben noti dai corsi di statistica
- **Metodi non parametrici**
 - non si assume nessun modello particolare
 - famiglie principali di metodi:
 - raggruppamento
 - campionamento

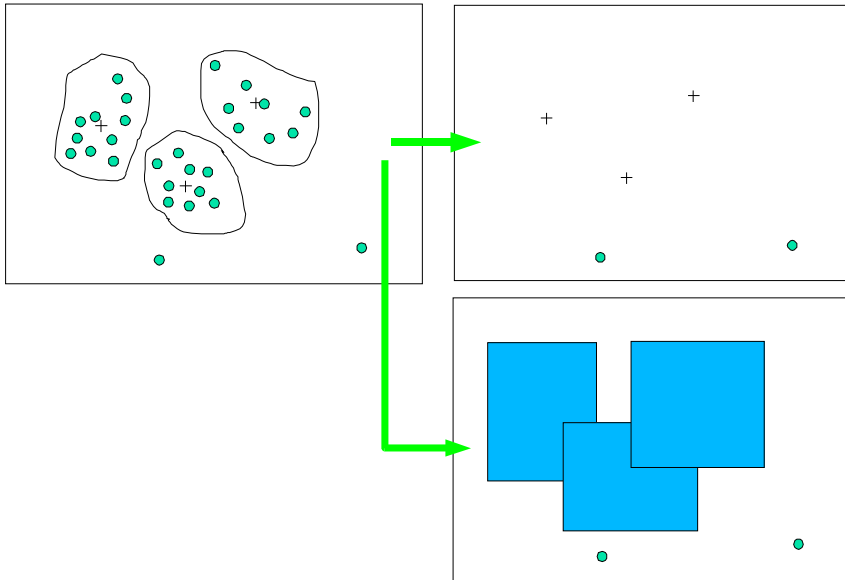
43

Raggruppamento (1)

- Si dividono i dati in cluster
- La rappresentazione dei cluster sostituisce la rappresentazione iniziale dei dati.
 - cosa si intende per rappresentazione dei cluster?
 - varie possibilità:
 - un **punto medio**,
 - una **figura geometrica** che approssima il cluster (poligoni, cerchi, etc..)

44

Raggruppamento (2)



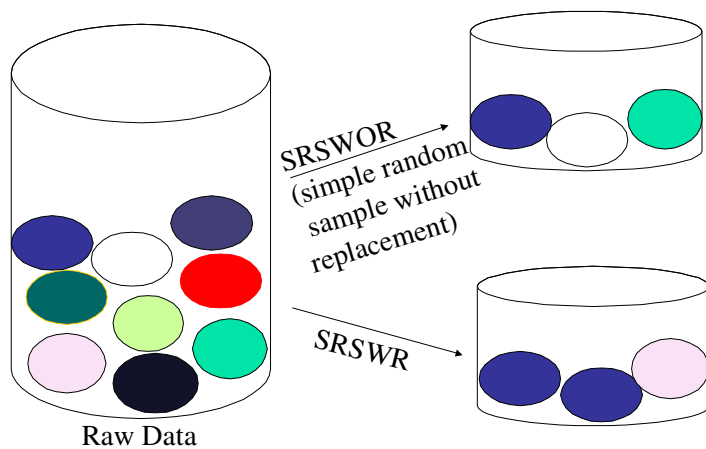
45

Campionamento (1)

- Scegliere un sottoinsieme dei dati (**campione**) per eseguire le analisi.
- Sia N il numero di istanze nel mio insieme di dati D e n il numero di istanze che voglio scegliere per il campione.
 - campionamento semplice senza rimpiazzo:
 - scelgo $n < N$ istanze da D . In ogni scelta, tutte le istanze hanno uguale probabilità, e non è possibile scegliere due volte la stessa istanza.
 - campionamento semplice con rimpiazzo
 - come sopra ma è possibile scegliere più volte la stessa istanza
- Rischiano di alterare i risultati dell'analisi
- Un vantaggio dei metodi di campionamento è che il costo per ottenere un campione è proporzionale alla dimensione del campione
 - ha dunque complessità **sub-lineare** rispetto ai dati in input

46

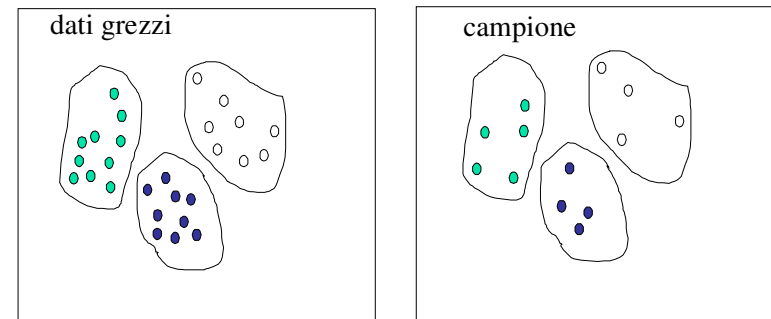
Campionamento (2)



47

Campionamento (3)

- Campionamento **stratificato**
 - i dati sono divisi in gruppi disgiunti $G_1 \dots G_k$ chiamati **strati**
 - scelgo da ogni gruppo G_i un campione proporzionale alla dimensione di G_i (ovvero scelgo $n * |G_i| / N$ elementi)
 - per problemi di classificazione, gli strati corrispondono alle classi.



48

Discretizzazione e gerarchie di concetti

49

Discretizzazione

- Consiste nel ridurre il numero di possibili valori diversi che assume un attributo
 - si divide il range dell'attributo in **intervalli** o comunque in **sottoinsiemi**
 - si sostituisce ai dati originali una etichetta che rappresenti l'intervallo o il sottoinsieme a cui appartiene
 - può migliorare l'efficienza di alcuni algoritmi di analisi
- Si può applicare la discretizzazione in maniera ricorsiva per ottenere una **gerarchia di concetti**
- Alcuni metodi per valori numerici
 - **binning** (già visto come algoritmo di data cleaning)
 - **discretizzazione basata sull'entropia**
 - **segmentazione per partizionamento naturale**

50

Binning

- Binning
 - si procede come nel caso della pulizia dati
 - oltre a rimpiazzare i dati con un valore caratteristico del bin a cui esso appartiene (media, mediana, estremi o altro), spesso lo si sostituisce con una “etichetta” che rappresenta l'intervallo di valori che esso assume
 - Esempio: 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
 - Partizionamento **equi-depth** in intervalli di 4 elementi ($d=4$)
 - Bin 1: 4, 8, 9, 15
 - Bin 2: 21, 21, 24, 25
 - Bin 3: 26, 28, 29, 34
 - Intervalli ottenuti: [-4,18],[18,25.5],[25.5,34]
 - gli estremi di ogni intervallo sono stati posti a metà tra il massimo valore del bin corrispondente e il minimo del successivo.

51

Discretizzazione ed entropia (1)

- Si applica tipicamente come preliminare alla classificazione
 - esiste un attributo “classe” per il calcolo dell'entropia
 - Ogni valore v di un attributo A è una possibile frontiera per la divisione negli intervalli $A < v$ e $A \geq v$.
 - Scelgo il valore che mi dà il maggiore **guadagno di informazione** $IG(S,v)$ definito come la differenza tra
 - l'entropia dell'insieme S dei dati iniziali: $E(S)$
 - l'entropia media dopo il partizionamento.
- $$E(S,v) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$
- dove S_1 ed S_2 sono gli insiemi corrispondenti alle condizioni $A < v$ e $A \geq v$ rispettivamente.
- $I(S,v) = E(S) - E(S,v)$

52

Discretizzazione ed entropia (2)

- Il processo si applica ricorsivamente ai sotto-intervalli così ottenuti, fino a che non si raggiunge una **condizione di arresto**
 - ad esempio, fino a che il guadagno di informazione che si ottiene diventa inferiore a una certa soglia d
- Supponiamo di avere le classi “s” ed “n” e l’insieme S costituito dalle seguenti coppie: (0, s), (2,n), (30,n), (31,n), (32,s), (40,s).
 - Calcoliamo le varie entropie condizionate:
 - $E(S,0)=1$ $E(S,2)=0.87$ $E(S,30)=1$ $E(S,31)=0.92$
 - $E(S,32)=0.54$ $E(S,40)=0.81$
 - Il guadagno di informazione maggiore si ha generando i sotto-intervalli $A < 30$ e $A \geq 30$
- Spesso, piuttosto che dividere sul valore ottenuto (30), si divide sul valore di mezzo tra quello ottenuto e il precedente
 - in questo caso si avrebbe $A < 16$ e $A \geq 16$

53

Partizionamento naturale (1)

- I metodi precedenti danno spesso intervalli innaturali
 - intervalli del tipo (~~€~~0.000-~~€~~0.000) sono più desiderabili di intervalli del tipo (~~€~~1.492-~~€~~0.872).
- la **regola 3-4-5** può essere usata per generare intervalli naturali
- in linea di massima, il metodo divide un dato intervallo in 3, 4 o 5 sotto-intervalli diversi, ricorsivamente a seconda dell’intervallo di valori assunti dalla **cifra più significativa**
 - se l’intervallo copre 3, 6, 7 o 9 valori distinti della cifra più significativa, partiziona l’intervallo in 3 intervalli
 - di uguale ampiezza per 3, 6 o 9 valori di stinti
 - nella proporzione 2-3-2 per 7 valori distinti
 - per 2, 4 o 8 valori distinti, partiziona i dati in 4 intervalli.
 - per 1, 5 o 10 valori distinti, partiziona i dati in 5 intervalli.

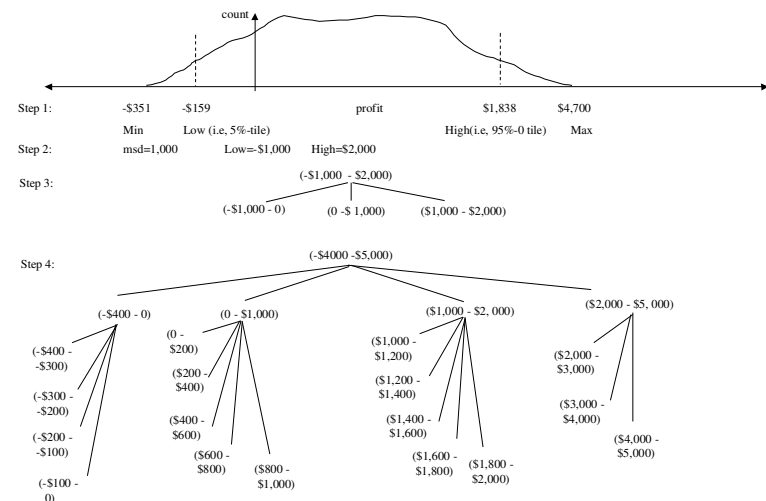
54

Partizionamento naturale (2)

- Se l’attributo A varia da -199 a 1838:
 - si arrotondano gli estremi dell’intervallo alla cifra più significativa, ottenendo l’intervallo (-1000,2000]
 - l’intervallo copre 3 cifre significative diverse
 - $(2000 - (-1000)) / 1000 = 3$
 - si divide in 3 sotto-intervalli (-1000,0], (0,1000], (1000,2000]
 - si procede ricorsivamente (se si vuole)
- la regola funziona male se ci sono valori estremi molto diversi dai valori medi.. in tal caso, si può usare per partizionare solo i dati dal 5° al 95° percentile

55

Partizionamento naturale (3)



56

Discretizzazione per attributi categoriali

- I metodi visti prima funzionano per attributi numerici
 - e per gli attributi categoriali?
- Abbiamo già visto i concetti di
 - **schema hierarchy**: la gerarchia è data specificando un ordine tra gli attributi
 - **set-grouping hierarchy**: la gerarchia è data specificando a mano la gerarchia
- Esistono possibilità intermedie
 - specificare gli attributi da usare per la gerarchia ma non l'ordinamento.
 - l'ordinamento è ricercato automaticamente, basandosi sul numero dei distinti valori assunti dagli attributi

57

Discretizzazione per attributi categoriali (2)



58