Lezioni di Data Mining

Ivan Diliso

Indice

Capitolo 1

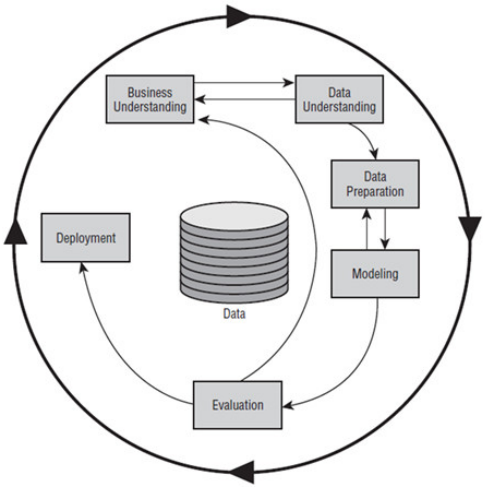
Processo KDD (Knowledge Discovery in Databases)

1.1 Introduzione: Il processo KDD

Il processo di knowledge discovery in databases è un processo interattivo e iterativo con possibilità di backtracking dove l'utente interviene nei diversi passaggi. Il processo origina dai dati, la selezione permette di filtrare solo i dati utili in fase di analisi, il preprocessing trasforma i dati in una forma computabile, la trasfromazione in una forma utilizzabile da un algoritmo di data mining, passare da unità di osservazione (i miei dati osservati) ad unità di analisi (dati su cui si svolge lo studio obiettivo). Successivamente applico gli algoritmi di data mininig e interpreto i risultati, in base alla valutazione e all'interpretazine valuto di tornare agli step precedenti.

1.1.1 CRISP-DM

Cross Industry Standard Process for Data Mining. Standardizzazione del processo di KDD. Il processo viene decomposto in fasi con step definenedo input e ouput di ogni passo. Vengono inoltre definite delle scelte intermedie.



1.2 Business understanding

Determinare gli obiettivi aziendali, i requisitit minimi da soddisfare, presenza di problemi e opportunità e eventuale sponsorizzazione dirigenziale. Eseguito da data analyst e business analyst. Vengono definite le aspettative del processo di data mining. Produce:

- Conoscenza di fondo
- Obiettivi di business
- Criteri di successo

1.2.1 Assess situation

Informazioni su risorse, vincoli, assumptions, disponibilità dei dati. Produce un glossario della terminologia e una analisi dei dati e dei costi e benefici.

1.2.2 Obiettivi data mining

Gli obiettivi di business vengono trasformati in obiettivi del data mining (termini tecnici) vengono quindi definiti i criteri di successo tecnici del data mining (ad esempio un certo livello di accuratezza).

- Predizione: Classificazione, regressione, stima (supervisionato o diretto)
- Descrizione: Gruppi affinità, clustering, profilazione (non supervisionato)
- Prescrittiva: Non effettuo predizione, ma definisco delle azioni in funzione del valore di una variabile, viene combinata la predizione con la descrizione.

Nella predizione ho variabili dipendenti e indipendenti, mentre nella descrizione queste non sono presenti e vengono scoperte relazioni senza definizione di dipendenza a monte, viene studiata la correlazione tra le variabili. Alcuni dei task sono:

- Classificazione: Mappare un dato in classi predefinite
- Regressione: Variabile dipendente numerica
- Gruppi affinità: Associazioni tra gruppi di variabili
- Clustering: Scoprire sottopololazioni omogenee, divisione in gruppi di un popolazione
- Summarization: Descrizione compatta di un insieme di variabili
- Dependency modeling: Scoprire dipendenza tra variabili, dipendenza sia strutturale sia quantitativa in caso di variabili pesate
- Change and deviation detection: Scoperta del cambiamento dei dati dalle misurazioni o dati precedenti, socperta di outlier o pattern emergenti

1.2.3 Produce project plan

Produrre un piano degli step successivi, la durata, input e output, durata, e altre dipendenza da ulteriori risultati.

1.3 Data understanding

1.3.1 Raccolta dati iniziali

Accedere ai dati rilevanti indicati nell'inventario delle risorse (aziende che vendono dati consorziali etc). Produce una descrizione della collezione di dati iniziali che elenca i dati, la posizione e le proprietà iniziali, metodo di accesso ed eventuali problemi.

1.3.2 Descrizione dei dati

Esame delle proprietà, produco un report descrittivo in termini di formato, valore potenziale, quantità, nome degli attributi e altre caratteristiche superficiali. Può essere utile ricorrere ai metadati ottenuti da fonti come data warehouse.

- Categorici: finiti valori possibili, applicabili operatori relazionali ma non aritmetici.
  - Nominali: Non è presente un ordine
  - Ordinali: Presente una relazione d'ordine anche gerarchica (strutture ad albero)
- Quantitativa: Posso applicare operatori aritmetici
  - Discreti: Valori interi numerabili
  - Continui: Valori reali non numerabili. Se sono definiti a livello di rapporto (ratio level) in molti casi hanno più senso operazioni di moltiplicazione e divisione.

1.3.3 Ispeziono la qualità del dato

Controllo se il dato è conforme al valore effettivo

- Accuratezza, confomità del valore osservato con il valore effettivo
- Completezza, assenza di valori mancanti
- Consistenza, rappresenzazione uniforme
- Aggiornamento, i dati non sono obsoleti

Problemi più grandi nei progetti KDD sono di qualità e integrità del dato. Se nell'analisi vengono utilizzati dati esterni secondari (non nati o modellati per obiettivi di data mining) far coincidere dati esterni ed interni può risultare difficile. Insieme al data managemente specialist viene valutata anche la sensibilità temporale del dato. Produco un report della qualità dei dati.

1.3.4 Esplorazione dei dati

Esploro i dati a disposizione tramite metodi statistici e visualizzazione dei dati. Permette di riconoscere assimettrie nelle distribuzioni dei dati (eventuali categorie possono essere sovra o sottorappresentate), questo non indica poca qualità ma un indicatore per le fasi di analisi di utilizzare tecniche robuste rispetto all'assimetria. Per variabili categoriche strumenti come istogrammi o diagrammi a torta, per dati quantitativi misure statistiche come media, mediana, massimo minimo o strumenti grafici come scatterplot o boxplots.

1.4 Data preparation

La fase di data preparation è un processo multifase composto da molti step individuali che riguardano la pulizia dei dati, la selezione, l'estrazione delle feature rilevanti e la riduzione e trasformazione dei dati.

1.4.1 Selezione dei dati

Riguarda le tuple delle relazione del database, utilizza criteri di selezione: rilevanza, qualità e vincoli tecnini, limitazioni su volumi o tipi di dato. Produce un report con le motivazioni per inclusione o esclusione dei dati. La selezione può essere automatica tramite sampling (campionamento) o feature selection (selezione delle feature).

1.4.1.1 Campionamento

Rimozione di alcune delle tuple del database in modo da create un dataset più piccolo e maneggevole

Random sampling Scelta casuale, ogni instanza ha la stessa probabilità di essere selezionata, si assume una distribuzione uniforme dell'importanza dei dati. Importante tenere conto della dimensione per valutare quanto un dataset è rappresentativo. Nella forma semplice si assume di poter enumerare le istanze. Nella forma con rimpiazzamento ogni elemento può essere considerato più volte (posso ottenere un dataset campionato con elementi ripetuti) nella forma senza rimpiazzzamento viene utilizzato un meccnaismo di controllo delle istanza considerate per evitare duplicati. Questa forma è inutile se la provavilità di considerare due volte la stessa istanza è molto bassa (dataset di dimensione molto elevata)

Stratified random sampling Divisione della popolazione in gruppi in modo da campionare ogni strato (gruppo) separatemente. In questo modo posso stabvilire a pripri la dimensione rispetto agli strati di un campione. Uno dei vantaggi principali è dare maggiore rilevanza a strati poco rappresentati. Non è possibile utilizzare questa tecnica per analizzare proprietà complessive del dataset (sovrarappresentazione di singoli strati, poca generalità). Gli strati devono

essere scelti in modo da sottolineare le differenze. Questa metodologia minimizza la varianza intera (tra gli individui di uno strato) e massimizza la varianza esterna (tra gli individui di diversi strati) (questo è anche un criterio di scelta per gli strati). Le stime all'interno di uno strato saranno più precise.

Cluster sampling In alcuni casi famiglie di instanze formano naturalmente dei cluster, è quindi possibile effettuare un campionamento sui cluster creati. Questo metodo porta vantaggi in terminiti di sostenibilità sia economica sia di calcolo. Quando questa metodologia è applicata a blocchi di partizione del disco è chiamata block sampling. Se i cluster contengono istanze molto simili si perdono infomazioni sulla varianza.

Two-stage sampling Scelta randomica dei cluster e successivamente campionamento all'interno dei cluster scelti.

Systematic sampling Utilizzabile quando la popolazione di un dataset è numerata, il numero di elementi può anche essere variabile o inizialmente non conosciuto. Anche chiamata kth-sampling in quanto viene campionato un elemento ogni k istanze. Metodo vulnerabile ad eventuali periodicità presenti nei dati creando un campione polarizzato su un periodo che non riflette la popolazione.

Two phase sampling Organizzo il campionamento su una o più variabili la cui distribuzione non è conosciuta. Nella prima fase si stima la distribuzione del parametro, nella seconda viene effettuato il campionamento.

- *FASE 1*: Seleziono dalla popolazione un campione S_1 di dimensione n_1 e osservo X
- *FASE 2*: Utilizzando probabilità di scelta sulla base del parametro X osservato nella fase 1, utilizzo S_1 com popolazione e creo un nuovo campionamento S_2 di dimensione n_2

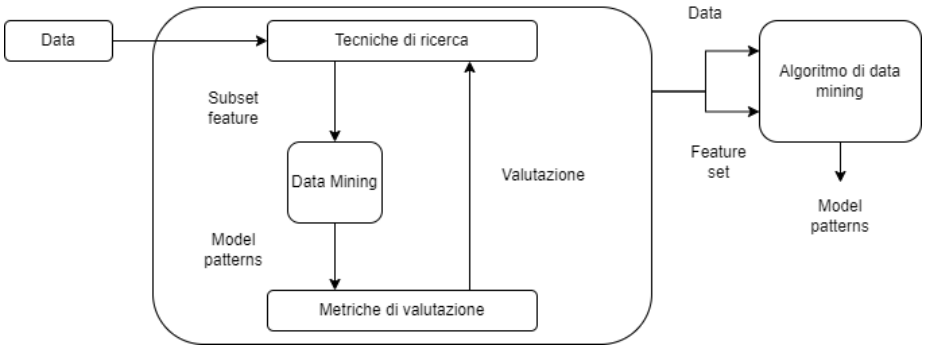
Questo approccio può anche essere utilizzato per capire la dimensione del campionamento. Per stimare la grandezza del campione sono anche presenti delle tecniche statistiche che rieantrano nella branca delle *power analysis*

Dimensione del campione e progressive sampling La dimensione del campione non è indice di ottimalità del sistema. Molte informazioni con poca varianza non portano informazione. E possibile avere campioni di dimensione crescete per capire se c'è una convergenza verso un moodello. Nel progressive sampling estendo il campione progressivamente fino a dei criteri di stop o alla convergenza ad un valore (parametro in cui il modello si stabilizza).

1.4.1.2 Feature selection

Ridurre gli attributi del dataset per ridurre lo spazio di ricerca (riduco lo spazio delle ipotesi, le soluzioni). Necessario per la *curse of dimensionality*: è difficile lavorare con dati ad alta dimensionalità in quanto il numero dei campioni dovrebbe crescere al crescere dei campioni, creando però cosi stime meno accurate. Le prestazioni del sistema degradano al creascere delle feature a parità di dimensione del campione

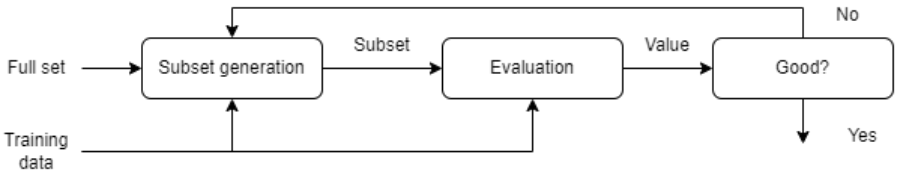
Wrapper models Utilizzo l'algoritmo di data mining per determinare se un subset di feature è buono. Dato un algoritmo di data mining che crea un iperpiano di separazione tra i campioni, utilizzo questa separazione come parte integrante del criterio di selezione delle feature. La selezione viene quindi effettuata attorno al modello di data mining.



Nella fase centrale si usa un set ridotto di dati in modo da non portare bias al modello. Può essere costoso in quanto sto applicando più volte un algormito di data mining e necessita di criteri di stop (anche provenienti da cross validation)

Filter models Indipendente dalla tecnica di data mining, non performante su applicazioni specifiche ma meno suscettibile a cambiamenti delle tecniche adottate, si basa sulle proprietà intrinsiche dei dati, sulla base di misure di informazione e distanza.

Embedded models Modello ibrido wrapper e filter model in un unico modello generale. La feature selection viene vista come insieme di feature subset selection, valutazione delle feature e criterio di stop.



Differenze tra modelli I modelli wrapper permettono di risolvere un problema speicfico, permettendo di ottimizzare il criterio di scelta delle feature tramite gli algoritmi di datamining, ma risulta particolamente dispendioso in termini di tempo e risorse in quanto richiedono cross validazione nella valutazione. I metodi filter sono molto più veloci in quanto non utilizzano data mining ma i risultati non sono ottimali. I metodi embedded sono parte integrante del sistema di data mining.

1.4.1.3 Feature subset selection

Strategie per la selezione del sottoinsieme di feature rilevanti:

- Enumerazione: Tutti i possibili subset, computazionalmente pesante
- Random: Generazione casuale e scelta del migliore
- Sequenziali: Utilizzando uno dei seguenti criteri di selezione
  - Forward: partendo da un insieme vuoto aggiundo ad ogni passo
  - Backward: partendo dal dataset totale rimuovo un elemento ad ogni passo

Queste strategie sono basate su una ricerca *greedy* con complessità 2^N con N feature presenti

1.4.2 Valutazione delle feature

Misure per decidere e valutare quali feature dovrebbero essere aggiunte o rimosse

1.4.2.1 Teoria dell'informazione

L'informazione deve essere inversamente proporzionale alla probabilità del verificarsi di un evento. Se un evento è certo l'informazione è 0. Viene considerato il logaritmo della probabilità di un evento:

$$I_p = -log_2(p) \quad [0,1] \rightarrow [0,\infty]$$

Si sceglie la base due per favorire una rappresentazione binaria. Abbiamo K eventi c_i con una distribuzione di probabilità tale che la somma delle probabilità degli eventi è uguale a 1. Definiamo quindi il guadagno d'ingormazione:

$$IG(X) = \sum_i U(P(c_i)) - E\left[\sum_i U(P(c_i|X))\right]$$

La funzione U è una qualsiasi funzinoe di incertezza. Se la funzione di incertezza scelta è I_p allora $\sum_i U(P(c_i))$ si definisce misura di *entropia*

1.4.2.2 Misure di distanza

Densità di probabilità condizionata Se l'obiettivo è la classificazione si può utilizzare la distanza tra funzioni di densità di probabilità condizionata alla classe:

$$P(X|C) \quad C = C_1 \dots C_n \quad P(X|C) \quad P(X|C_1) \dots P(X|(C_n)$$

Sia $D(X)$ distanza tra $P(X|C_1)$ e $P(X|C_2)$ allora preferisco X ad Y se $D(X) > D(Y)$. Conoscere X separa meglio le probabilità, portando quindi ad una maggiore separazione tra le classi.

Kullback-Leibler divergence Siano P,Q distribuzioni di probabilità sul dominio V

$$m_{kl}(P,Q) = \sum_{v \in V} q(v) log \left(\frac{q(v)}{p(v)} \right)$$

Misura quanto la distanza P approssima Q. Infatti se $P(v) = Q(v)$ allora il logaritmo si annulla. P viene chiamato *soggetto* mentre Q viene definito il *riferimento* per la valutazione. Misura asimmetrica (i risultati possono cambiare se inverto P e Q. Non calcolabile se Q dovesse essere 0, tendente a infinito se P tende a 0. Range di valori non limitato. Posso definire chi meglio approssima P tra due altre misure Q_1 e Q_2 ma non permette di determianre la bonta di Q in assoluto in quanto non ha limite superiore, utilizzabile quindi solo per confronto.

χ^2 divergence

$$m_{\chi^2}(P,Q) = \sum_{y \in Y} \frac{|p(y - q(y)|^2}{p(y}$$

Topologicamente più forte di KL in quanto la divergenza chi quadro implica la convergenza KL. Misura asimmetrica.

$$m_{kl}(P,Q) <= m_{\chi^2}(P,Q)$$

Variation distance (Manhattan distance)

$$m_l(P,Q) = \sum_{v \in V} |p(v) - q(v)|$$

Minkowski L_2 (Distanza euclidea)

$$m_2(P,Q) = \sum_{v \in V} |p(v) - q(v)|^2$$

Tutte le misure della forma $m_p(P,Q)$ soddisfano le proprietà *metriche*. Potrebbero non essere adegueate in caso di distribuzioni rumorose dei dati.

1.4.2.3 Misure di dipendenza

Quanto una feature è dipendende da un altra feature sia R(X) la dipendenza di X dalla classe C. Scelgo X su Y se $R(X) > R(Y)$. Queste misure non forniscono una misura globale di bontà della feature. In caso di feature ridondanti non ho modo di scegliere tra le due.

1.4.2.4 Misure di inconsistenza

Cercare un sottoinsieme delle feature minimo che permette di separare le classi in modo consistente con la separazione del dataset completo

$$P(C|fullset) = P(C|subset)$$

1.4.2.5 Approcci in letteratura

Approccio esaustivo

- Focus: Misura di inconsistenza con valutazione esaustiva sequenziale forward
- Branch and bound: Enumerazione di tutte le soluzioni candidate. Rimuove sottinsiemi di candidati poco promettenti usando un limite superiore e inferiore della misura da ottimizzare.

Approccio euristico

- SFS (sequential forward search) e SBS (sequential backward search)
- DTM: Addestro un classificatore e uso le feature trovate nel classificatore (wrapper)

Approccio nondeterministico

- LVF (Las Vegas Filter) e LVW (Las Vegas Wrapper): Generazione random del subset e valutazione con inconsistenza il primo e classificatore il secondo
- Algoritmi genetici e simulated annealing

Approccio basato su insanze

- Relief: Feature pesate sulla loro capacità di differenziare istanze di classi differenti.

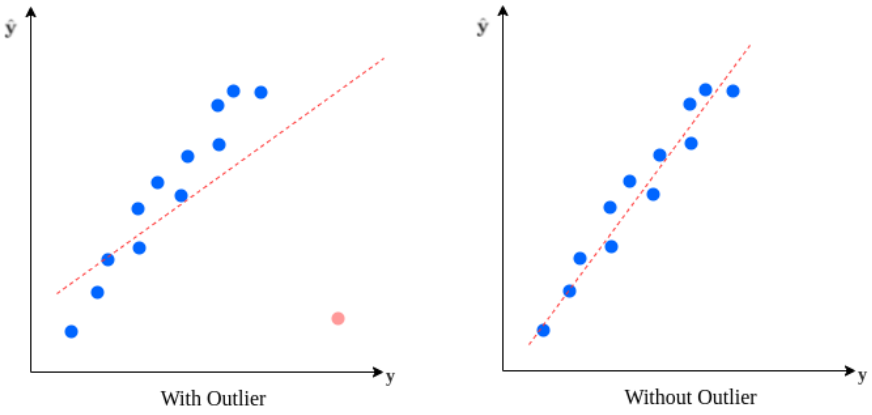
1.4.3 Pulizia dei dati

Esaminare i dati per la ricerca d i mancanze, errori e non osservanza dei vincoli posti sui dati. Viene creato un rapport con decisione, azioni e elenco delle trasformazioni applicate ai dati. Possiamo ritrovare tre tipologie di errori:

- Dati rumorosi
- Outlier e anomalie (fuori norma, rari ma verificabili, altrimenti sarebbero dati rumorosi)
- Valori mancanti

1.4.3.1 Dati rumorosi e outlier o anomalie

Dati osservati diversti dai dati effettivi. Dovuto ad errori nel processo di osservazione, errore di fattori latenti che influenza la variabile osservata o errore umano. Distribuzioni sbilanciate portrbbero indicare la presenza di anomalie o outlier.



1.4.3.2 Valori mancanti

Valori mancanti o valori non validi:

- Rimozione del dato: Potrebbe portare a perdita di dati importanti. Va bene in base alla percentuale di dati persi e di dati mancanti.
- Rimozione della feature: Se è presente un numero significativo di istanze con questa feature non avvalorata
- Sostituzione con valore probabile: Valore mediano o media per valori quantitativi o moda per valori categorici. POssibilità di utilizzare modelli predittivi per sostituire il dato (utilizzo le informazioni presenti per sostituire il valore mancante)

- Creazione di una fase di analisi in grado di gestire dati mancanti. Questo metodo permette di eliminare il bias che potrebbe subentrare nella scelta di eliminazione o avvaloramento dei dati mancanti.

1.4.4 Costruzione dei dati

Creazione di variabili derivate o nuove istanze a aprtire da quelle raccolte: conversione di valori in diversi formati, aggredazione di dati e computazioni di nuovi valori a partire dai presenti

1.4.4.1 Scaling e normalization

Normalizzazione è il processo di trasformare i dati in modo da modificarne il range di valori. Permette di migliorare l'interpretabilità del modello. Una normalizzazione può essere forzata dal modello di data mining scelto. Si definisce standardizzazione un cambiamento dei valori che porta il dato ad avere media 0 e deviazione standard 1.

min-max normalization

$$v' = \frac{v - min_a}{max_a - min_a}(newmax_a - newmin_a) + newmin_a$$

0-1 normalization

$$v' = \frac{v - min}{max}$$

Queste due normalizzazioni utilizzano i valori di min e max trovati durante la fase di data understanding o basate sui dati osservati. Nuove istanze protrebbero trovarsi fuori dagli intervalli portando ad errori di tipo *out of bound*

z-score

$$v' = \frac{v - mean_a}{std_a}$$

Il nuovo valore atteso è 0. 90% dei dati ricade tra $-\delta$ e δ (deviazione standard). Standardizzazione utile se non conosco l'intervallo di partenza dei valori. Utile anche von valori di outlier che potrebber dominare la valutazione min max (outlier agli estremi dell'intervallo)

Scalamento decimale

$$v' = \frac{v}{10^j} \quad dove \quad j = min_j : max(|v'|) < 1$$

1.4.4.2 Discretizzazione

Variabili quantitative in categoriche

- Equal width: Definisco N intervalli. Trovo B valore massimo e A valore minimo. Creo intervalli di ampiezza $W = (B - A)/N$. Formula diretta, sensibile agli outlier e con valori distribuiti in modo asimettrico (skewed)
- Equal depth: Definisco N invervalli tali da avere ognuno lo stesso numero di elementi. Riduco la perdita di informazioni, buono scalamento dei dati, in caso di dati ripetuti una distribuzione asimettriche non è evitabile
- Equi-log ranges: Utile quando un attributo mostra una distribuzione esponenziale all'interno di un intervallo. Ogni range $[b, a]$ è scelto in modo che $log(b) - log(a)$ è uguale in tutti gli intervalli

In entrambi i casi $\#bins \geq \#classes$. Una regola euristica per il numero di bins $\#bins = M/(3C)$ con M numero di esempi di training e C numero delle classi. L'intuito è che se la variabile fosse utile a classificare una delle classi C allora ci sarà una parte che andrebbe associata a C, una ad un intervallo più piccolo ed una ad un intervallo più grande.

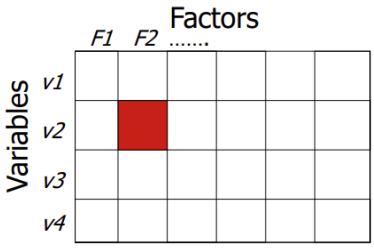
One of N (discreto a numerico) Vettore *one-hot* con un solo bit settato a 1 e gli altri a 0. Migliore rispetto ad una codifica numerica ordinata perchè non crea differenze tra i valori.

1.4.4.3 Riduzione dei dati

Creazione di indici che rappresentano un grande numero di dati, è sia data reduction sia data construction. Teniche utilizzate sono factor analysis e principal component analysis (branca della statistica multivariata)

Factor analysis Analisi della struttura della interdipendenza tra grandi numeri di valriabili definendo un insieme di dimensioni comuni chiamate fattori. Tutte le variabili sono considerate simultaneamente, ognuna messa in correlazione con l'altra. Descrivo gli individui con un numero ridotto di fattori che rappresentano cluster di variabili. Possibile creare una serie di indici che rappresentano le variabili originali.

$Y = X_1(variable_1) + X_1(variable_2) + \ldots \quad dove \quad X_1, X_2, \cdots = factor\,loadings$

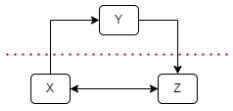


I fattori finali sono combinazione lineare delle variabili con dei valori chiamati factor loading (con valori nell'intervallo [-1,1] dove un valore $\leq \pm 0.3$ indica importanza bassa $= \pm 0.4$ indica un valore normale e $\geq \pm 0.5$ indica un valore importante).Viene quindi eseguito un ranking dei fattori e scartati i fattori non rilevanti riducendo al minimo la perdita di informazione.

- VANTAGGI: Insiemi più piccoli di variabili, modelli più performanti
- SVANTAGGI: Computazionalmente pesante, perdita di ifnormazioni, difficile interpretabilità. Variabili aggregate di difficile comprensione, passaggio da categorio a fattore numerico.

1.4.5 Integrate data

Combinazione di dati da più sorgenti per creare nuovi record. Produco un modello analitico dei dati una ristrutturazione consolidata, integrate e dipendente dal tempo dei dati. Divisione della unità di anlisi (entità che vengno analizzate nello studio) e unità di osservazione (dati collezionati, dove solo effettuate le osservazioni sistematiche). Ad esempio uno studio potrebbe avere l'unità di osservazione sui singoli individui, ma effettuare una analisi su gruppi.Le due unità possono coincidere. Si parla di *fallacia ecologica* quando vengono effettuate inferenze sul singolo individuo a partire da dati aggregati. L'operazione di aggregazione non può essere annullata. Alcune correlazioni possono non essere genuine



In realtà è Y a creare la correlazione da X a Z, esempio di correlazione non genuina. Y è il fattore latente che crea la correlazione.Le unità di analisi possono essere composte da diverse unità di osservazione. Ad esempio nei CRM viene applicata una tecnica di householding, dove i dati vengono aggregati sulla base dell'abitazione.

1.4.5.1 Single table assumption

In generale si trattano dati in single table assumption, ma in molti casi può essere una limitazione. Se aggiungo nuove feature non crea problemi, si potrebbero avere molti dati ripetuti (ad esempio nella relazione acquirente acquista le informazioni del cliente sono ripetute, inoltre in questo caso sto cambiando l'unità di analisi che diventa l'acquisto e non il cliente). Le ripetizioni possono creare delle distribuzioni non bilanciate. Tabelle non normalizzate possono portare a dei cambiamenti nell'unità di analisi. Posso quindi:

- Creare tabelle separate
- Aggregare dei dati che non sono unità di analisi (ad esempio aggrego gli acquisti se l'unità di analisi è il cliente)

1.4.5.2 Multi relational data mining

Dati distribuiti su più tabelle. Abbiamo relazioni che definiscono dei target (unità di analisi) e altre che descrivono informazioni ulteriori del target. Studiate nella inductive logic programming (si può mettere in relazione la tupla con l'istanza di un predicato di ILP)

- Nodelli relazionali per la ricerca di patter relazionali (rispetto ai modelli proposizionali delle STA)
- Predicate calculus, SQL, grafi
- REgole di classificazione regressione e associazione relazionale

1.4.6 Formattazione dei dati

Permette di modellare dati e valori possibili, gestire dati sparsi e dati mancanti. Alcuni formati sono CSV e ARFF (proprietario Weka)

1.5 Modeling

1.5.1 Introduzione

Scelta dell'algoritmo di data mining, produce l'algoritmo. Fase influenzata da:

- Obiettivo (classificazione, regressione, etc...)
- Abilità nel trattare i dati
- Abilità nel trattare relazioni multiple
- Spiegabilità (le reti neurali ad esempio sono poco spiegabili)
- Scalabilità

- Familiarità

Un algortimo di data mining è quindi composto da tre componenti principali:

- Rappresentazione: Relazionale o proposizionale, utente che utilizzerà il sistema, quantitativo o qualitativo
- Valutazione: Incertezza basata su test, stime di probabilità o fuzzy theory
- Ricerca: Esaustiva o greedy

1.5.2 Rappresentazione

Un modello di rappresentazione è un linguaggio L che descrive il modello, linguaggio dipendende dal tipo di pattern che il modello deve esprimere. La capacità descrittiva può essere qualitativa o quantitativa ed espressa con semplici regole, l'insieme di queste regole crea una teoria logica. Il linguaggio dipende anche dall'utente finale del modello.I pattern possono essere intercampo, relazioni di attributi nello stesso record o intercampo-record che mettono in relazione valori aggregati su gruppi di record.

1.5.2.1 Utenti dell'algoritmo

- Umani:
  - Rappresentazione con linguaggio naturale o formalismo logico, facilmente interpretabile anche per la macchina. (alberi di decisione)
  - Possono essere utilizzate anche rappresentazioni visuali, testuali o a forma di albero. Rappresentazione grafica non solo per pattern appresi ma anche dell'analisi dei dati. Ricade nell'area del *information visualization* e del *visual data mining*. La rappresentazione visuale si può applicare sia ai dati che ai modelli estratti. Informazioni su forma e densità sono meglio rappresentate in forma visuale.
- Computer: Linguaggi di programmazione funzionali o formalismi dichiarativi
- Sistema di KD: Scoperte date in input al modello come conoscenza del dominio

Scegliendo un linguaggio troppo limitato non sarà possibile addestrare in maniera ottimale un algoritmo causando *underfitting*, mentre una rappresentazione più complessa può causare un sovradattamento ai dati di input chiamato *overfitting* (aggiunta di gradi di libertà)

1.5.2.2 Rappresentazione dell'incertezza

Scegliere se rappresentare l'incertezza, fornire un grado di probabilità della decisione con metodi quali:

- Probabilità
- Deviazione standard o intervalli di fiducia
- Misure di belief
- Misure fuzzy set
- Valutazione visiva: grandezza densità e shading

L'intercetta può derivare da un campinamento. Costrutisco il modello non su tutti i dati ma su un campione, poartando ad una incertezza della replicabilità della costruzionedel modello. Tecniche statistiche possono essere utilizzare per valutare il grado di incertezza e per determinare quanti sample aggiuntivi sono necessari per raggiungere il livello di confidenza richiesto nei risultati.

1.5.3 Valutazione del pattern

Metriche di valutazine permettono di dstimare l'incertezza di un pattern specifico. Una delle tecniche di valutazione è la *cross validation*

1.5.4 Ricerca

La ricerca può essere condotta su:

- Parametri: Migliore combinazione di parametri che ottimizza il modello. Si ricerca una forma chiusa (soluzione a problema analitico) o si utilizza una ricerca iterativa (algoritmi come discesa del gradiente)
- Modello: Loop sulla scelta del modello
- Modello e parametri

Molti algoritmi si basano sull'assunzione di un modello prefissato. Gli algoritmi tendono ed essere euristici a causa dello spazio proibitivo dei parametri (greedy seach) che non permette di ottenere soluzioni in forma chiusa.

1.5.5 Ulteriori fasi

1.5.5.1 Progettazione test

Procedura per valutare la qualità del modello, questa fase descrive di dati di training e di test, i parametri da valutare, metriche e metodi di valutazione

1.5.5.2 Build model

Modello viene applicato ai dati di training. Produce i setting dei parametri e motivi delle scelte effettuate. Vengono inoltre forniti i dati attesi di accuratezza o altre metriche.

1.5.5.3 Output in PMML

Produco l'output del modello seguendo gli standard industriali, come il Predictive Model Markup Language PMML basato su XML. Permette di avere un unico standard di rappresentazione dell'output risolvendo problemi di incompatibilità e formati proprietari.

1.5.5.4 Valutazione del modello

I risultati del modello vengono valutati e interpretati tramite metriche di accuratezza, precisione, confidenza della precisione, comprensibilità. Prevede anche una revisione dei parametri del modello di data mining.

Minimum Description Lenght (MDL) minimo numero di bit necessari per rappresentare la regola e le sue eccezioni (basata sulla teoria dell'informazione). Questa metriche permette di decidere tra due regole equivalenti, scegliendo la regola più espressiva.

La valutazione del modello avviene su dati mai visti prima, viene valutata non solo la singola predizione ma l'intero modello. Vengono confrontati più modelli con la stessa accuratezza (protrebbero avere livelli di varianza diversi sulle singole predizioni).

1.6 Valutazione

1.6.1 Introduzione

Viene valutato il modello rispetto agli obiettivi di business. Il data analyst discute i risultati con il business analysit. L'obiettivo è presentare nuove scoperte in ottica di business intelligence guidata dai dati. Produrre scoperte in una forma che si lega direttamente all'obiettivo di business che si era posti sul progetto. Vengono scoperte e prodotte soluzioni data drive e iniziative oriented.

1.6.2 Processo di review

Dopo aver valutato il risultato valuto il processo che ha prodotto i risultati. Produce una raccomandazione di ulteriori attività da svolgere. La domanda principale in questa fase è *"è stato trovato qualcosa di interessante valido e utilizzabile?"*

In caso di risposta negativa è possibile tornare alle fasi precedenti

Customer Data Segmentation Si tenta di etichettare ogni cluster e dare una interpretazione di business a ciascuno di essi. Questo è possibile con una rappresentazione omogenea dei cluster. Se ho cluster troppo piccoli o troppo grandi vuol dire che ho fallito nel differenziare i clienti. Posso decidre d ivariare le variabili alla base della segmentazione, producendo una segmentazione equilibrata.

Predictive models Valutazione dell'accuratezza, un fallimento della valutazione guida il team verso la ricerca dei fattori influenti nascosti, l'eventuale presenza di overfitting or la ricerca di nuove variabili di input. In caso di overfitting si cerca di semplificare il modello.

1.7 Deployment

Piano d'impiego, stategie di rilascio del modello. Nel paino di sviluppo l'obiettivo è quello di portare a compimento gli obiettivi definiti nel business modeling. Importante l'assimilazione della conoscenza, formulare modi in cui le nuove informazioni possono essere sfruttate e utilizzate. Alcuni esempi:

- Segmentation databse: Integrato in politiche di campagne di marketing
- Modelli predittivi: Identificazione dei clienti e sistemi di direct mailing
- Regole di associazione: Cross sell di prodotti esistenti e promozioni

Una delle difficoltà e l'integrazione della nuova conoscenza e modelli con le infrastrutture pre esistenti. A seguito del deployment segue una fase di monitoraggio e manuntenzione su lunghi periodi. Viene infine prodotto un reort finale che indinca il risultato complessivo e la valutazione degli aspetti positivi e negativi del progetto.

Capitolo 2

Similarità e distanze

2.1 Introduzione: Similarità e distanze

Molti dei problemi di data mining richiedono il calcolo della similarità. Nella analisi dei dati le misure di similarità si chiamano misure di somiglianza.

Misura di sogmiglianza Sia E insieme di elementi. A misura di somiglianza $A : E \times E \to \mathbb{R}$ tale che $r(a,b) \in \mathbb{R}$ definisco $d(a,b)$ dissimilarità e $s(a,b)$ similarità con le seguenti proprietà:

1. Simmetria $r(a,b) = r(b,a) \quad \forall a,b \in E$
2. $\forall a \in E \quad r_a^* = r(a,a)$:
   - Dissimilarità $\quad r(a,n) \geq r_a^* \quad \forall b \in E$
   - Similarità $\quad r(a,n) \leq r_a^* \quad \forall b \in E$

In situazioni classiche $r_a^* = r_b^* \quad \forall a,b \in Ea \neq b$ quindi ho un minimo di dissimilarità $r^* = d^* = 0$ e un massimo di similarità $r^* = s^* = 1$

- Dissimilarità

$$d : E \times E \to \mathbb{R} \quad \forall a,b \in E \quad 0 = d(a,a) \leq d(a,b) = d(b,a) < \infty$$

- Similarità

$$s : E \times E \to \mathbb{R} \quad \forall a,b \in E \quad 1 = s(a,a) \geq s(a,b) = s(b,a) \geq 0$$

Una funzione di similarità puà essere trasformata in una di dissimilarità e viceversa utilizzando funzioni strettamente decrescenti:

$$s = max(d) - d \quad oppure \quad s = \sqrt{max(d) - d}$$

Quasi ordering Le misure di somiglianza inducono un quasi ordinamento sul set $E \times E$. Sia S e $\leq$ è quasi ordinamento se $\forall x,y \in S$:

1. Riflessiva $\quad x \leq x$
2. Transitiva $\quad x \leq y \wedge y \leq z \implies x \leq z$

Misura definita Una dissimilarità si dice *definita* se:

$$\forall a,b \in E \quad d(a,b) = 0 \implies a = b$$

Distanza Una distanza è una dissimilarità definita che soddisfa la disuguaglianza triangolare. La divergenza d iKullback-Liebler non soddisfa la disuguaglianza triangolare, quindi non è una distanza.

2.1.1 Contesto di utilizzo

Bisogna sceglire la metrica d i distanza il base al dominio applicativo (misura di similarità o distanza). Le misure possono essere espresse in forma chiusa o algoritmicamente (distanza su serie temporali)

- Grafi di similarità: Pesi su gli archi, viene calcolata la similarità tra nodi
- Spectral embedding: Conversione dei grafi di similarità in dati multidimensionali in modo che la distanza tra i punti multidimensinai rispecchi la connettività dei nodi.

Le misure di distanza sono sensibli a fattori quali:

- Distribuzione dei dati
- Dimensionalità
- Tipologia dei dati

2.2 Misure di distanza su dati multidimensionali

Misure di distanza e impatto di fattori esterni per misure multidimensionali categoriche e quantitative

2.2.1 Norma L_p

Misura di distanza più comune per dati quantitativi, dati $\overline{X} = (x_1 \dots x_d)$ e $\overline{Y} = (y_1 \dots y_d)$ si definisce norma L_p:

$$L_p(\overline{X}, \overline{Y}) = Dist(\overline{X}, \overline{Y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Alcune forme notevoli sono la distanza Euclidea per $p = 2$ e la distanza di Manhattan per $p = 1$, queste due infatti hanno una interpretazione geometrica, la prima è la distanza in linea retta tra i due punti nello spazio, la seconda può essere vista come la minore distanza che dovrebbe essere percorda daun automobile per mouoversi da due punti situati in una città suddivisa in isolati quadrati. La norma euclidea è invariante alle rotazioni, questo permette di applicare PCA e SVD senza intaccare la distanza. Per $p = \infty$ viene scelta la dimensone in cui i due punti sono più distanti (scelgo una sola feature per valutare la distanza), questa misura enfatizza troppo la differenza tra i due vettori se questa differenza è dovuta a rumore.

Svantaggi Nonostante la facile interpretabilità questa misura non performa bene con dati ad alta dimensionalità a causa di fattori quali sparsità dei dati (i dati sembrano in apparenza molto simili tra loro) ,distribuzione, rumore e rilevanza delle feature.

2.2.2 Impatto della rilevanza specifica del dominio

In alcuni casi la conoscenza del dominio ci permette di sapere quali feature sono più importanti di altre, in questi casi è possibile pesare le varie feature in base alla rilevanza nel dominio specifico. In questo caso è possibile utilizzare una misura chiamata *norma L_p generalizzata* anche chiamata *distanza di Minkowki generalizzata*

$$Dist(\overline{X}, \overline{Y}) = \left(\sum_{i=1}^{d} a_i |x_i - y_i|^p \right)^{\frac{1}{p}}$$

con a_i coefficente associato alla i-esima feature.

2.2.3 Impatto dell'alta dimensionalità

Applicazioni di data mining basate sulla distanza perdono la loro efficacia all'aumentare della dimensionalità, questo perchè le misure di distanza non riflettono bene le differenze semantiche intrisiche dei dati ad alta dimensionalità, si dice infatti che gli algoritmi di data mining basati su distanza sono qualitativamente inefficenti.

Esaminiamo un cubo unitario nel quadrate positivo con un vertice nell'origine degli assi. Essendo il cubo unitario ognuna delle cordinate di un qualsiasi punto all'interno del cubo avra distribuzione uniforme in $[0, 1]$. Dato un punto $\overline{X} = (x_1 \dots x_d)$ ognuna delle x_i sarà una variabile aleatoria uniforme in $[0, 1]$. Consideriamo quindi la distanza di manhattan dall'origine di un qualisasi punto $\overline{X}$ interno al cubo.

$$Dist(\overline{O}, \overline{X}) = \left(\sum_{i=1}^{d} (x_i - 0) \right)$$

Il risultato è una variabile aleatoria con:

- Media $\mu = d/2$
- Varianza $\sigma = \sqrt{d/12}$
- Intervallo di distanza: Ognuno dei punti nel cubo avrà una distanza dall'origine compresa tra $[Dist_{min}, Dist_{max}] = [\mu - 3\sigma, \mu + 3\sigma]$

Contrasto Si definisce constrasto il rateo di variazione della distanza tra due punti:

$$Contrast(d) = \frac{D_{max} - D_{min}}{\sigma}$$

Interpretabile come quanto sono distanti i punti rispetto alla distanza media. Nel caso della distanza di manthattan nel cubo unitario questo contrasto è pari a $\sqrt{12/d}$ possiamo quindi notare che all'aumentare della dimensinoalità porta ad una diminiuzione del contrasto rendendo la variazione di distanza tra punti piccolissima (tutti i punti sono indistinguibili tra loro, distanze molto ridotte).

2.2.4 Impatto delle feature localmente non rilevanti

La norma euclidea può portare ad un maggiore contributo da parte di feature rumorose a causa del suo approccio tramite somma di quadrati, risulta quindi necessaria una selezione delle feature rilevanti. In alcuni casi le feature rilevanti per la distanza potrebbero essere sensitive alla particolare coppia di oggetti che viene controntata, questo problema non è risolvibile tramite selezione delle feature in quanto la rilevanza delle feature in questo caso è determinata localmente, questo può avvennire anche quando tutte le feature sono rilevanti localemente. Il rumore additivo può portare ad una concentrazione delle distanze.

2.2.5 Impatto delle differenti norme L_p

All'aumentare del valore p nella norma L_p è più probabile che attributi irrilevanti diventino determinanti della distanza. Nel caso estremo con $p = \infty$ perdo completamente informazioni sulle similarità locali, basta una sola feature irrilevante con valori rumorosi può portare ad una dissimilarità tra gli oggetti anche se le feature mostrano una similarità. Le norme L_p degradano all'aumentare della dimensonalità ma mostrano degradazioni molto più veloci ad alti valori di p. In due dimensoni la degradazione è molto bassa, questo in quanto a bassa dimensionalità il valore di p è meno rilevante.

2.2.5.1 Metriche frazionali

Questo problema è alla base della nascita delle *metriche frazionali* con $p \in (0, 1)$ permettendo metriche migliori in casi di alta dimensionalità:

$$Dist_d^f(\overline{X}, \overline{Y}) = \left(\sum_{i=1}^{d} (x_i - y_i)^f \right)^{\frac{1}{f}}$$

2.2.6 Match based similarity computation

Nel calcolare una misura di similarità o distanza sarebbe desiderabile poter selezionare le feature localmente rilevanti, un approccio semplice è quello di mettere in evidenza feature vicine (rilevanti, corrispondenza cumulativa su più valori di attributi, ad esempio più record che hanno diversi attributi rilevanti simili e un solo attributo rumoroso) e tenere sotto controllo il peso delle feature irrilevanti. Vengono quindi de enfatizzate le variazioni rumorose su gli attributi individuali mentre vengono contate le corrispondenze comulative su più dimensioni. Questo approccio è di difficile implementazione su dati a bassa dimensione, in quanto non è possibile definire un impatto cumulativo con metodi statistici robusti in poche dimensioni.

La norma L_p produce un effetto contrario invece, andando ad enfatizzare più gli attributi rumorosi che quelli rilevanti. Con questo approccio vengono quindi enfatizzate non le differenze (distanze) tra i dati, ma gli effetti cumulativi di attributi vicini tra loro. Nel testo si usa ad esempio la *cosine similarity*:

$$similarity = cos(\theta) = \frac{AB}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sum_{i=1}^{n} A_i^2 \sum_{i=1}^{n} B_i^2}$$

Con A,B vettori e A_i, B_i rispettive compotenti. Questa misura varia in $[0, 1]$ ed è specialmente indicata quando il magnitudo dei vettori non ha importanza.

2.2.6.1 Soglie di prossimità

Le soglie di prossimità permettono di de-enfatizzare variazioni rumorose su certi attributi (de-enfatizzo il livello di dissimilarità). I dati vengono discretizzati in bucket di uguale frequenza (stesso numero di osservazioni per ogni bucket). Ogni dimensione viene divisa in k_d bucket (il numero totale di bucket per ogni dimensione dipenderà dalla dimensionalità del dataset) ognuno contentente $1/k_d$ dei record totali.

1. Siano quindi $\overline{X} = (x_1 \dots x_d)$ e $\overline{Y} = (y_1 \dots y_d)$ due record del dataset
2. Per la dimensione i se x_i e y_i apparendono allo stesso bucket, i due record si diranno in *prossimità* per la dimensione i
3. Il sottoinsieme di dimensioni in cui $\overline{X}$ e $\overline{Y}$ sono in prossimità si definisce *insieme di prossimità* e si indica con $S(\overline{X}, \overline{Y}, k_d)$
4. $\forall i \in S(\overline{X}, \overline{Y}, k_d)$ siano m_i e n_i estremo minimo e massimo del bucket della dimensione i in cui i due record sono simili

Posso allora definire la similarità *PSelect*:

$$PSelect(\overline{X}, \overline{Y}, k_d) = \left[\sum_{i \in S(\overline{X}, \overline{Y}, k_d)} \left(1 - \frac{|x_i - y_i|}{m_i - n_i} \right)^p \right]^{1/p}$$

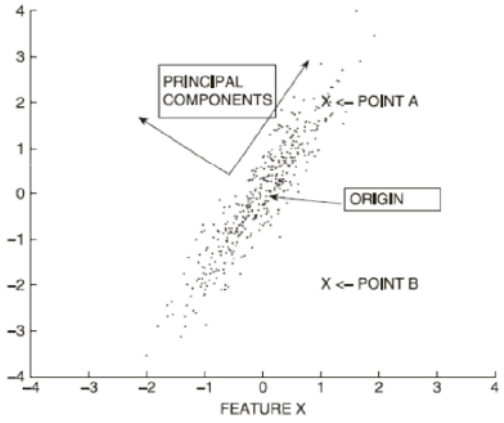
Dato che ogni elemento nella sommatoria varia tra $[0, 1]$ la similarità PSelect avrà valori in $[0, \#S(\overline{X}, \overline{Y}, k_d)]$. La probabilità che due record condividano un bucket è $1/k_d$ e in media si avranno d/k_d componenti non nulle. Il grado di dissimilarità su dimensioni distanti è ignorato, misura robusta rispetto al rumore. Si dimostra che scegliere valori di $k_d \propto d$ fornisce un livello di contrasto costante in contesti di alta dimensionalità per alcune distribuzioni dei dati.

I risultati ottenuti dimostrano che in uno spazio altamente dimensionale è meglio mirare ad avere limiti di qualità più elevati sulle dimensoni, in modo da utilizzare un minor numero di dimensioni nel computare la similarità

A basse dimensoni una scelta di k_d in funzione di d ci assicura un comportamento simile alle norme L_p, ad alte dimensioni invece questa misura di similarità si comporta come le misure di similarità su testo. Risulta particolarmente adatta per algoritmi di nearest-neighbor.

2.2.7 Impatto della distribuzione dei dati

La norma L_p dipende soltanto dalla posizione dei record nello spazio ed è quindi invariante alla statistiche globali dei dati restanti, sarebbe utile invece andare a considerare anche la distribuzione di base dei restanti dati.



Il punto A è posto in una direzione di alta varianza, mentre il punto B è posto in una direzione di bassa varianza (la distanza dall'origine è uguale per i due punti, ma il punto B si discosta molto dalla distribuzione dei dati). La distanza tra 0 e A dovrebbe essere minore della distanza tra 0 e B, possiamo risolvere questo problema utilizzando la *Mahalanobis distance*. Sia Σ la matrice di covarianza (matrice simmetrica $cov(i,j) = cov(j,i)$) sulle dimensioni $(d \times d)$ con $\Sigma_{(i,j)} = cov(i,j)$ allora:

$$Maha(\overline{X}, \overline{Y}) = \sqrt{(\overline{X} - \overline{Y})\Sigma^{-}1(\overline{X} - \overline{Y})^T}$$

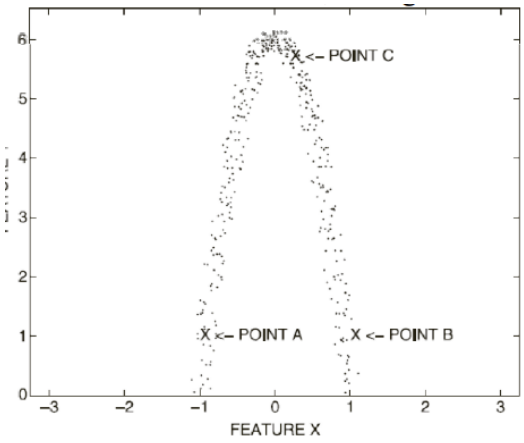
Questa distanza è simile alla distanza euclidea ma normalizza i dati sulla base delle correlazioni interattributo. Se la matrice Σ è una matrice diagonale questa distanza è chiamata *distanza euclidea standardizzata* con σ deviazione standard:

$$d(\overline{x}, \overline{y}) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{s_i^2}} \quad con \quad s_i = \sigma(x_i, y_i)$$

Una ulteriore interpretazione della distanza di Mahalanobis avviene in termini di PCA, in cui vengono calcolati gli autovettori della matrice di covarianza dei dati, questi autovettori sono le componenti principali. Utilizzando questi vettori possiamo effettuare una rotazione sui dati in modo che la direzione di massima varianza sia allineata con l'asse delle ascisse e la seconda con l'asse delle ordinate (proiezione dei dati sulle componenti pricipali). In questo nuovo piano la matrice di covarianza è zero e la distanza di mahalanobis diventa la distanza euclidea.

2.2.8 Impatto di distribuzioni non lineari

Consideriamo il caso di dati con distribuzione non lineare di forma arbitraria.



Considerando la distanza euclidea i punti A e B sono i più vicini tra loro rispetto al punto C, ma osservando la distribuzione dei dati questa mostra il contrario. Una intuizione per risolvere questo problema è che solo piccoli salti da punto a punto possono misurare accuratemente la distanza tra punti seguendo la distribuzione muovendomi verso la direzione che minimizza la distanza collettiva (somma della distanza percorda ad ogni salto). Questa misura di distanza si chiama *geodesica*. Si assume che le distribuzioni non lineari siano solo localmente euclidee (non globalmente).

2.2.8.1 ISOMAP

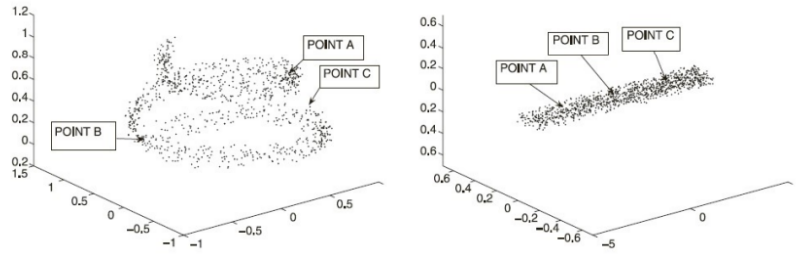
Misura di distanza derivata da riduzione di dimensionalità non lineare e modelli embedding:

- Calcolo i KNN per ogni punto
- Creo un grafo G con i punti come nodi e come archi le distanza dei KNN
- La distanza tra due punti sarà il percorso più breve tra i due punti nel grafo G

2.2.8.2 Embedding con multi dimensional scaling (MDS)

L'aggiunta di un ulteriore passo di embedding permette di proiettare i dati su uno spazio multidimensionale (Multi Dimensional Scaling) più piccolo cercando di presevare le distanze che valgono sullo spazio di dimensione più grande. Questo processo rende calcoli ripetuti di distanza più efficenti a costo della accuratezza. Questo provesso viene applicato dopo aver costruito le distanze per

ogni coppia di nodo in G. Questo produce una linearizzazione della forma non lineare riportata in uno spazio multidimensionale più piccolo.

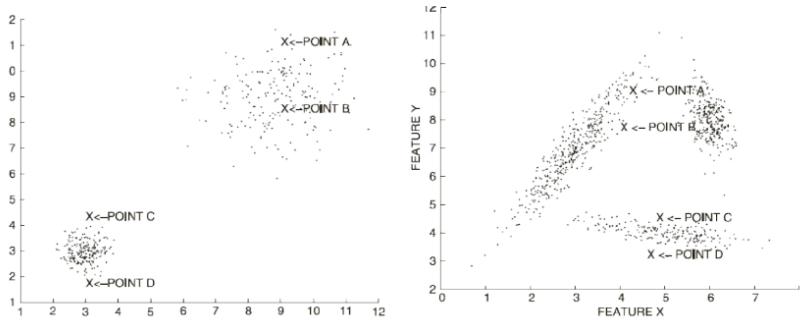


In generale i dati ad alta dimensionalità sono allineati lungo delle forme non lineari a più bassa dimensione chiamati *manifolds*, questi possono essere appiattiti in una nuova rappresentazione in modo da poter utilizzare metriche di distanza in modo efficace. La PCA può essere estesa alla scoperta di embedding non lineari tramite l'utilizzo di metodi chiamati kernel tricks.

2.2.9 Impatto delle distribuzioni locali

La distribuzione dei dati e l'accuratezza della distanza dipendono molto dalla distribuzine locale dei dati, questa variazione dalla distribuzione locale può essere di due forme:

- Densità dei dati in una regione, questa puà variare molto con la distribuzione locale (distanze diverse nel contesto della densità della distribuzione locale).
- Variazione dell'orientamento delle distribuzini locali dei dati.



(a) local density variation (b) local orientation variation

Metodi come il Local Outlier Factor (LOF) permettono di regolare le variazioni delle distribuzioni locali per incrementare le performance del modello.

2.2.9.1 Shared Nearest Neighbor Similarity

Il primo problema (densità della distribuzione locale) può essere risolto tramite l'utilizzo di SNNS. Diviso nelle seguenti fasi:

1. In una fase di preprocessing viene calcolato il KNN di ogni punto
2. Il valore SNNS sarà uguale al numero dei neighbor comuni tra due punti

Questa misura è localmente sensibili dato che dipende sul numero di nighbor comuni e non su proprietà globali. Questo metodo può essere utilizzato per definire dei grafi di similarità dei punti sottostanti ai punti che condividono almeno un neighbor comune connessi tra loro tramite archi.

2.2.9.2 Metodi generici

L'idea di principale di questi metodi si divide in due step:

1. Partitziono i dati in un insieme di regioni locali
2. Per ogni coppia di oggetti viene calcolata la regione più appropriata e calcolata la ditanza tra i due punti utilizzando le statistiche locali della regione

Possono essere utilizzati diversi metodi di clustering per dividere i dati i regioni, una problematica risiede nel dover definire in ogni caso una misura di distanza per la divisione in regioni locali.

2.2.10 Considerazioni computazionali

Le misure di distanza sono implementate come subroutine che sono ripetutamente utilizzare all'interno di un modello di data mining. Metodi come ISOMAP e metodi generici cono computazionalmente pesanti e hanno una complessità di almeno N^2 rispetto al numero di dati (difficili da implementare su grandi quantità di dati) ma essendo delle trasformazioni one time permettono di creare delle rappresentazioni che possono poi essere utilizzate efficentemente dagli algoritmi di data mining.

2.2.11 Dati categorici

I dati categorici pongono una ulteriore difficoltà nel calcolo della distanza in quanto a differenza dei dati quantitavi non sono ordinati. Un approccio possibile è la binarizzazione, che potrebbe generare dati altamente sparsi, si possono quindi utilizzare delle misure di similarità robuste rispetto alla sparsità dei dati (come le misure di similarità per il testo).

Considerando due record $\overline{X} = (x_1 \dots x_d)$ e $\overline{Y} = (y_1 \dots y_d)$ definiamo similarità:

$$Sim(\overline{X}, \overline{Y}) = \sum_{i=1}^{d} S(x_i, y_i)$$

Con $S(x_i, y_i)$ misura di similarità sui singoli attributi.

Overlap

$$S(x_i, y_i) = \begin{cases} 1 & x_i = y_i \\ 0 & otherwise \end{cases}$$

Questa misura non tiene congo delle frequenze relative degli attributi. Nei dati categorici è importante tenere conto delle proprietà statistiche aggregate nel calcolare la similarità, l'idea principale è quella di dare più importanza (peso maggiore) ad attributi categorici meno frequenti e meno importanza agli attributi frequenti.

Inverse Occurance Frequency La similarità viene pesata con una funzione inversa del valore di matching

$$S(x_i, y_i) = \begin{cases} \frac{1}{p_k(x_i)^2} & \text{if } x_i = y_i \\ 0 & otherwise \end{cases}$$

Con $p_k(x_i)$ frazione dei record che contengono l'attributo x_i

Goodall

$$S(x_i, y_i) = \begin{cases} 1 - p_k(x_i)^2 & \text{if } x_i = y_i \\ 0 & otherwise \end{cases}$$

2.2.12 Dati misti categorici e numerici

La misura di similarità viene divisa in due parti

$$Sim(\overline{X}, \overline{Y}) = \lambda \cdot Sim_{num}(\overline{X}, \overline{Y}) + (1 - \lambda) \cdot Sim_{cat}(\overline{X}, \overline{Y})$$

Il parametro λ permette di regolare l'importanza relaiva degli attributi categori e numerici. In assenza di conoscenza di dominio sull'importanza degli attributi è possibile scegliere un valore di λ uguale alla frazione degli attributi numerici nei dati. Può essere necessario un ulteriore step di normalizzazione dato che i valori numerici e categorici utilizzano scale di valori completamente diverse, viene quindi utilizzata la deviazione standard di entrambe le tipologie di attributi:

$$Sim(\overline{X}, \overline{Y}) = \frac{\lambda \cdot Sim_{num}(\overline{X}, \overline{Y})}{\sigma_{num}} + \frac{(1 - \lambda) \cdot Sim_{cat}(\overline{X}, \overline{Y})}{\sigma_{cat}}$$

In questo modo il valore λ rappresenta un vero peso relativo.

2.2.13 Dati binari

Dati binari multidimensionali sono rappresentazioni di dati basati su insiemi, dove il valore 1 rappresenta la presenza di un element in un insieme. Consideriamo una matrice di dati binaria $X = (x_{kj}) \in \{0, 1\}^{n \times p}$ formata da n vettori di granzezza p. La similarità di due vettori k e l viene definita da quattro indici:

$$\begin{cases} a = \sum_{j=1}^{p} x_{kj} x_{lj} & \text{numero di attributi uguali } x_{kj} = x_{ij} = 1 \\ b = \sum_{j=1}^{p} x_{kj}(1 - x_{lj}) & \text{numero di attributi diversi } x_{kj} = 1, x_{ij} = 0 \\ c = \sum_{j=1}^{p} (x_{kj} - 1) x_{lj} & \text{numero di attributi diversi } x_{kj} = 0, x_{ij} = 1 \\ d = \sum_{j=1}^{p} (x_{kj} - 1)(1 - x_{lj}) & \text{numero di attributi uguali } x_{kj} = x_{ij} = 0 \end{cases}$$

$$S_\theta = \frac{a+d}{a+d+\theta(b+c)}$$

$$T_\theta = \frac{a}{a+\theta(b+c)}$$

$$\boldsymbol{T_1 = |S_k \cap S_l| / |S_k \cup S_l|}$$

Notation	Definition	Range	Authors
S_1	$\frac{a+d}{p}$	[0,1]	Kendall, Sokal-Michener (1958)
S_2	$\frac{a+d}{a+d+2(b+c)}$	[0,1]	Rogers and Tanimoto (1960)
$S_{1/2}$	$\frac{2(a+d)}{2(a+d)+b+c}$	[0,1]	
T_1	$\frac{a}{a+b+c}$	[0,1]	Jaccard (1900)
T_2	$\frac{a}{a+2(b+c)}$	[0,1]	Sokal & Sneath
$T_{1/2}$	$\frac{2a}{2a+b+c}$	[0,1]	Dice (1945), Czekanowski (1913)

similarity	definition	range
Kulczynski	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}\right)$	[0,1]
Sokal & Sneath (1963)	$\frac{1}{4}\left(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{d+b}+\frac{d}{d+c}\right)$	[0,1]
Driver & Kroeber (1932), Ochiai (1957)	$\frac{a}{\sqrt{(a+b)(a+c)}}$	[0,1]
Sokal & Sneath (1963)	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	[0,1]
Yule (1927)	$\frac{ad-bc}{ad+bc}$	[-1,1]
Pearson	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	[-1,1]

Con S_k, S_l rappresentazion insiemistiche dei vettori k, l tali che $a = |S_k \cap S_l|$ (numero di elementi) e $a + b + c = |S_k \cup S_l|$

2.2.14 Dati ordinali

Sono valori categorici con ordinamento naturale, ma con distanza tra le categorie non conosciuta (come ad esempio la scala di Likert). La trasformazione in valori binari deve mantenere l'ordine dei valori. Una trasformazione possibile proposta da Sokal e Sneath (1963): Sia A un attributo ordinale con j_A valori distinti ordinati. Se il valore a assume la posizione numero h delle j_A posiziono possibili

allora le prime h posizioni del vettore di lunghezza j_A avranno valore 1 mentre le restanti $j_A - h$ avranno valore 0.

$$x_a = \underbrace{(\overbrace{1, 1, \ldots, 1}^{h}, \overbrace{0, 0, \ldots, 0}^{j_A - h})}_{j_A}$$

Se i valori della feature ordinale possono essere messi in corrispondenza con i primi r_k numeri interi $0 \leq x_k \leq r_k$ con x_k numero intero associato al valore ordinale, posso definire la sequente distanza:

$$d_0(x, y) = \frac{\sum_{k=1}^{n} x_k + \sum_{k=1}^{n} y_k + 2\sum_{k=1}^{n} min(x_k, y_k)}{\sum_{k=1}^{n} x_k + \sum_{k=1}^{n} y_k - \sum_{k=1}^{n} min(x_k, y_k)}$$

2.3 Misure di distanza temporali

I dati temporali contengono:

- Un attributo contestuale che rappresenta il tempo
- Uno o più attributi comportamental iche misurano le proprietà che variano in un certo periodo di tempo

I dati temporali possono essere rappresentati come serie temporali continue o come sequenze discrete (che possono essere viste come una discretizzazione della forma continua). Il design di misure di similarità su serie temporali è specifico in base all'applicazione. La distanza euclidea può lavorare bene in molti contesti ma non tiene conto dei fattori di distorsione della serie quali:

- Scaling e traslazione degli attributi comportamentali: Serie temporali diverse con scale diverse. I pattern simili presenti nelle due serie non possono essere confrontati in maniera efficace utilizzando i valori assoluti.
- Traslazione (contestuale) dell'attributo temporale: In alcune applicazioni il valore assoluto dell'attributo contestuale (quando la lettura dei valori comportamentali è stata effettuata, come nel caso di dati medici) non ha importanza, in questi casi l'attributo temporale va spostato (shifted) in una delle due serie in modo da permettere un matching più accurato.
- Scaling (contestuale) dell'attributo temporale: In questo caso la serie andrebbe allungata o compressa lungo l'asse temporale per migliorare il matching, questa tecnica è chiamata time warping. Una ulteriore complicazione è che differenti segmenti della serie potrebbero necessitare di time warping diversi.
- Discontinuità nel matching: Presenza di segmenti rumorosi in lunge serie temporali, necessità di una misura di distanza robusta rispetto ai dati rumorosi.

2.3.1 Normalizzazione dell'attributo comportamentale

Normalizzazione e scalatura più facile su attributi comportamentali rispetto ai contestuali in questo è possibile effettuare queste azioni nella fase di preprocessing:

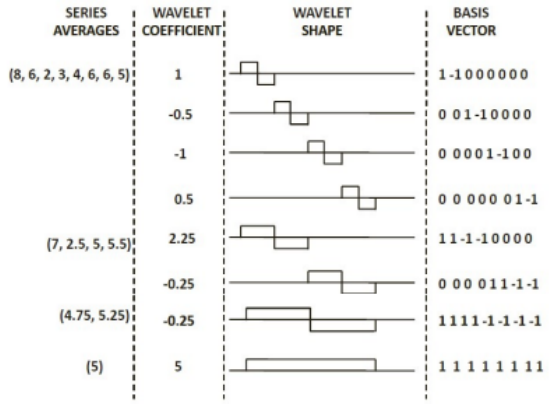
- Traslazione: L'attributo è centrato sulla media
- Scalatura: La deviazione standard dell'attributo è scalata a 1

Queste normalizzazioni potrebbero non essere rilevanti su tutte le applicazioni, che potrebbero richiedere una sola delle due o entrambe, una scelta errata nella normalizzazione potrebbe portare ad una perdità di interpretabilità dei risultati.

2.3.1.1 Serie temporali come dati multidimensionali

Vengono utilizzati metodi come la *discrete fourier transform* e la *discrete wavelet transform*. Questi metodi possono essere visti come rotazioni del sistema di assi definito dai timestamp dell'attributo contestuale (ogni dimensione rappresenta un timestamp). La trasformazione wavelet trasforma la serie temporale in un dato multidimensionale come un set di coefficenti che rappresenta differenze in media tra differenti porzioni della serie, è possibile selezionale un subset di questi coefficenti per ridurre la quantità di dati.

Haar wavelet transform Se utilizzo come coefficenti solo la media di un intero periodo di misurazioni perdo l'informazione della variazione degli attributi comportamentali nel corso del periodo di tempo. Se invece uso come coefficenti la differenza delle temperature media della prima metà e della seconda metà del periodo posso derivare la media di entrambe le metà da questi valori. Applico questo processo ricorsivamente fino ad un livello di granularità delle misurazioni (ad esempio un sensore che legge un dato al secondo nel corso di una giornata). Valori di differenza più grandi forniscono informazioni sulle variazioni dei dati, sono quindi più importanti dei valori di differenza più piccoli. Questa tecnica crea una decomposizione di una serie temporale in un insieme di wavelate basis vector pesati con i coefficenti. Ogni coefficente rappresenta una variazione della serie temporale tra due metà di un particolare periodo di tempo. Il wavelate basis vector è una funzione a gradino che rappresenta il range temporale di questa variazione. La serie originale viene quindi decomposta in una somma di serie temporali più semplici (pesate con i coefficenti) ortogonali tra loro.



2.3.1.2 Norma L_p

La serie temporale viene trattata come un punto multidimensionale in cui ogni istante di tempo è una dimensione. Date due serie $\overline{X} = (x_1 \dots x_n)$ e $\overline{Y} = (y_1 \dots y_n)$:

$$Dist(\overline{X}, \overline{Y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Può essere applicata a trasformazioni in wavelet. Nel caso p=2 la distanza è accurata con la rappresentazione wavelete se i coefficenti wavelet più grandi vengono mantenuti. Si dimostra che se nessun coefficente viene rimosso allora la distanza è uguale tra due rappresentazioni diverse, inoltre la norma euclidea è invariante alle rotazioni degli assi. Lo svantaggio principale è che questa norma è definita per serie di grandezza uguale, non può quindi tenere conto delle distorsioni dell'attributo temporale.

2.3.1.3 Dynamic Time Warping

La serie temporale viene allungata o accorciata dinamicamente su diverse porzioni in modo da permettere un matching più efficace. Questa tecnica può essere usata sia per serie contiue e discrete, in quanto modifica soltanto l'attributo contestuale e non è legata alla natura degli attributi comportamentali. Il DTW permette un mapping uno a molti nel time warping, questo può essere visto come ripetere alcuni degli elementi più volte in una delle due serie temporali, in modo da creare artificialmente due serie con la stessa lunghezza che hanno un mapping uno a uno tra di loro (la distanza tra le nuove serie può essere quindi calcolata usando la norma L_p). La scelta degli elementi da ripetere viene effettuata in modo da minimizzare la distanza DTW, questo viene calcolato tramite programmazione dinamica (Appendice 1). Conseiderate la due serie $\overline{X} = (x_1 \dots x_m)$ e $\overline{Y} = (y_1 \dots y_n)$:

$$DTW(i,j) = distance(x_i, y_i) + min \begin{cases} DTW(i, j-1) & \text{repeat } x_i \\ DTW(i-1, j) & \text{repeat } y_i \\ DTW(i-1, j-1) & \text{repeat neither} \end{cases}$$

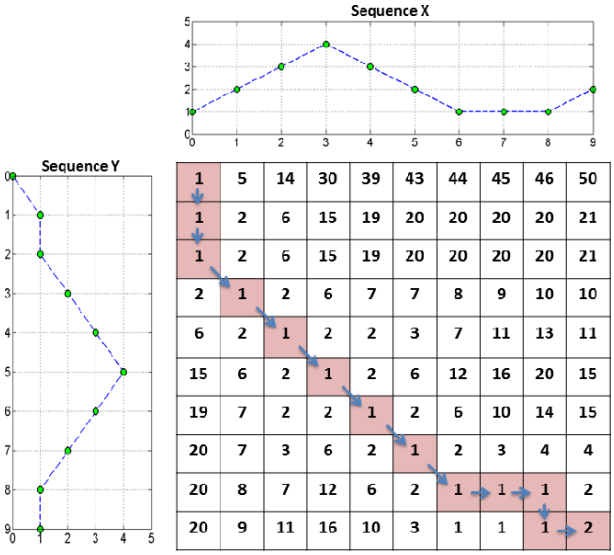
Viene ridotto di 1 solo uno dei due indici in quanto è possibile un mapping uno a molti (in un mapping uno a uno entrambi gli indici dovrebbero diminuire), la scelta dell'indice da diminuire dipende in base al miglior matching tra le due serie. L'indice non ridotto corrisponde all'elemento ripetuto. La distanza può essere definita come $|x_i - y_i|^p$ nel caso di serie continue (o qualsiasi altra misura di distanza). Il DTW può essere esteso a attributi comportamentali multipli utilizzando distanza su più attributi (distanza basata su vettori) nella ricorsione. I valori iniziali sono:

$$\begin{cases} DTW(0,0) & = 0 \\ DTW(0,j) & = \infty \\ DTW(i,0) & = \infty \end{cases}$$

Interpretazione geometrica Questo approccio richiede di computare:

$$DTW(i,j) \quad \forall i \in 1 \dots m \text{ e } \forall j \in 1 \dots n$$

Creando una tabella di $m \times n$ valori, quindi l'approccio può richiedere $O(mn)$ iterazioni. Il time warping ottimale può essere visto come il percorso ottimale tra i diversi valori di i e j nella tabella dei valori.



1	5	14	30	39	43	44	45	46	50
1	2	6	15	19	20	20	20	20	21
1	2	6	15	19	20	20	20	20	21
2	1	2	6	7	7	8	9	10	10
6	2	1	2	2	3	7	11	13	11
15	6	2	1	2	6	12	16	20	15
19	7	2	2	1	2	6	10	14	15
20	7	3	6	2	1	2	3	4	4
20	8	7	12	6	2	1	1	1	2
20	9	11	16	10	3	1	1	1	2

Window constraint Viene imposto un livello minimo w di allineamento posizionale tra due elementi in matching.

$$DTW_{window}(i,j) = \begin{cases} DTW(i,j) & |i-j| < w \\ \infty & \text{altrimenti} \end{cases}$$

In questo modo il loop interno dell'indice varia in $min\{0, i-w\}$ a $max\{n, i+w\}$

2.3.1.4 Metodi basati su finestre

Se due serie htanno molti segmenti corrispondenti contigui dovrebbero essere considerate simili. Siano $\overline{X}$ e $\overline{Y}$ due serie temporali e siano $\overline{X_1} \dots \overline{X_r}$ e $\overline{Y_1} \dots \overline{Y_r}$ le r finistre non sovrapponibili ordinate temporalmente estratte dalle rispettive serie temporali. Alcune finistre potrebbero non essere incluse nell'elenco delle finestre considerate, queste corrispondono a segmenti rumorosi che sono rimossi.

$$Sim = (\overline{X}, \overline{Y}) = \sum_{i=1}^{r} Match(\overline{X_i}, \overline{Y_i})$$

Si presentano difficoltà nella scelta della fuznione di matching in quanto match contigui su lunghe finestre temporali sono più rari che su piccoli segmenti della stessa lunghezza (scelta dipendente dall'applicazione). Anche la decomposizione ottimale in finestre può essere una scelta difficile.

2.3.2 Misure di similarità per serie discrete

Se è possibile un mapping uno ad uno è possibile utilizzare molte delle misure di distanze per dati multidimensionali categorici, in caso contrario sono disponibili diversi approcci.

2.3.2.1 Edit distance

Minor quantità di "effort" (o costo) per trasfroamare una sequenza in un altra tramite una serie di trasformazioni chiamate edits, anche chiamata distanza di Levenshtein. Le operazioni sono inserimento, cancellamento e rimpiazzamento, ognuna con costi specifici. In molte applicazioni il costo di rimpiazzamento è più alto rispetto ad inserimento e cancellamento. Esistono diversi modi per trasformare una stringa in un altra, la distanza di edit è definita come il costo ottimale per la trasformazione. La distanza di edit è calcolata tramite programmazione dinamica (Appendice 1) Si considerino i primi i simboli come nei metodi precedenti (stessa notazione):

$$Edit(i,j) = min \begin{cases} Edit(i-1,j) & + \text{ Costo cancellazione} \\ Edit(i,j-1) & + \text{ Costo cancellazione} \\ Edit(i-1,j-1) & + I_{ij} \cdot (\text{Costo rimpiazzamento}) \end{cases}$$

Con I_{ij} uguale a 0 se $x_i = y_i$ 1 altrimenti. I valori finali per la ricorsione sono:

$$\begin{cases} Edit(i,0) & = i \cdot \text{ Costo cancellazione} \\ Edit(0,j) & = j \cdot \text{ Costo cancellazione} \end{cases}$$

La misura di edit così definita non è simmetrica. Nella pratica però viene assunto un costo uguale per inserimento e cancellazione rendendo quindi la misura simmetrica. La misura di edit può essere estesa a dati numerici modificando le operazioni di inserimento cancellazione e rimpiazzamento con operazioni create per serie temporali continue. Questa metodologià è più complessa in quanto richiede di modellare un set di trasformazioni della forma di una serie temporale e i relativi costi.

2.3.2.2 Longest common subsequence

Una sottosequenza è un insieme di simboli estratti da una sequenza nello stesso ordine della sequenza originale (i simboli devono essere contigui). Sottosequenze uguali tra due serie temporali sono un indice di similarità tra le due serie, dove sottosequenze corrispondenti tra loro più lunghe indicano un più alto grado di similarità. Nonostante il numero di sottosequenze estraibili è esponenziale rispetto alla lunghezza della sequenza la distanza LCSS può essere calcolata in tempo polinomiale tramite programmazione dinamica (Appendice 1). Si considerino i primi i simboli come nei metodi precedenti (stessa notazione), nel metodo LCSS o esiste un matching sull'ultimo elemento della sequenza oppure viene eliminato l'ultimo elemento in una delle due sequenze.

$$LCSS(i,j) = max \begin{cases} LCSS(i-1,j-1)+1 & se \quad x_i = y_i \\ LCSS(i-1,j) & \text{altrimenti (nessun match su } x_i) \\ LCSS(i,j-1) & \text{altrimenti (nessun match su } y_i) \end{cases}$$

I valori finali della ricorsione sono entrambi settati a 0. Può essere applicato a serie continue dopo aver applicato una discretizzazione della serie continua in una sequenza di valori categorici.

2.4 Misure di distanza su grafi

Esistono due tipi di misure di similarità per grafi, nella prima la similarità è calcolata tra i due grafi presi nella loro interezza, nella seconda tipologia viene calcolata la similarità tra due nodi nello stesso grafo. Si assume che i grafi siano indiretti.

2.4.1 Similarità tra due nodi di un grafo

Sia $G = (N, A)$ un grafo indiretto con nodi N e archi A. Solitamente le funzioni di distanza utilizzano funzioni di costo, mentre le similarità utilizzano dei pesi,

è possibile passare da una misura all'altra tramite l'utilizzo di funzioni kernel euristiche scelte in base all'applicazione, come l'heat kernel $K(x) = e^{-x^2/t^2}$. La similarità tra due nodi nello stesso grafo è basata sul concetto di omofilia (due nodi sono più simili tra loro quando sono connessi tra loro tramite archi). Se due nodi sono collegati tra loro tramite molti percorsi e percorsi brevi, questi vengono considerati più simili tra di loro.

2.4.1.1 Misure di distanza strutturale

Sia $SP(s, j)$ la distanza del percorso più breve per arrivare dal nodo s ad un nodo j. Questo valore è uguale a 0 se $j = s$ altrimenti a ∞. Allora la distanza tra s e tutti gli altri nodi del grafo viene calcolata tramite:

- Tra tutti i nodi esaminati fino ad ora, selezionare il nodo i tale che $i = min_i(SP(s, i))$ e aggiorna le distanza da tutti i suoi neighbor j tramite:

$$SP(s, j) = min\{SP(s, j), SP(s, i) + c_{ij}\}$$

Questo metodo è alla base dell'algoritmo di Dijkstra. Approccio lineare sul numero di archi nel grafo (ogni nodo esaminato una sola volta). Questo tipo di misura non utilizza informazioni riguardanti la molteplicità dei percorsi tra i due nodi ma si basa solo sulla distanza strutturale (non funzionano benen quando due nodi hano un numero variabile di percorsi che li uniscono).

2.4.1.2 Random walk similarity

Utilizzata quando due nodi sono fortemente connessi con percorsi multipli. Si immagina di camminare randomicamente all'interno del grafo partendo dal nodo s procedendo verso i nodi adiacenti con una probabilità pesata sul peso dell'arco, è presente inoltre la possibilità di tornare indietro verso il nodo precedente con una probabilità chiamata *restart probability*. Questo creerà una distribuzione di probabilità con un grande bias verso il nodo s. Nodi simili a s avranno una più alta probabilità di essere visitati (posizioni che sono più vicine e che possono essere raggiunte in più modi sono più facilmente accesseibili in una random walk). Questa metodologia è legata al concetto di PageRank, infatti questa modifica all'algoritmo per calcolare la similarità tra nodi viene anche chiamata *personalized page rank*. Come nell'algoritmo di PageRank dopo un certo numero di step le probabilità convergeranno verso una distribuzione stazionaria.

2.4.2 Similarità tra due grafi

La similarità tra due grafi è più complessa in quanto molti nodi potrebbero avere le stesse etichette, rendendoli indistinguibili tra di loro, rendendo estremamente complesso anche il solo caso di determinare se due grafi sono identici. Questo problema NP-HARD è chiamato il problema dell'isomorfismo dei grafi. L'idea alla base di queste misure di similarità è:

1. Maximum common subgraph distance: Grafi sono considerati simili se contengono grandi sottografi in comune
2. Similarità strutturale: Difficile su grandi grafi ma più semplice su sottostrutture più piccole. L'idea è di contare le sottostrutture frequenti all'interno dei due grafi e riportare il numero come misura di similarità
3. Misura di graph-edit: Numero di edit necessari per trasformare una grafo in un altro
4. Graph kernels: Funzioni kernel utilizzate per misuarer la similarità tra grafi come il shortest path kernel e il random welk kernel.

Grafi dotati di label rappresentano formule relazionli nella logica del primo ordine, quindi la similarità tra grafi può essere basata nel matching parziale delle formule relazionali. Il matching è asimmetrico, è invece simmettrica la distanza:

$$Matching(A, B) + Matching(B, A)$$

2.5 Misure di distanza supervisionate

Le misure non supervisionate viste fin'ora trattano tutte le feature equalmente, non vengono utilizzate informazioni riguardo il significato semantico della distanza (le intenzioni dell'utente). Un modo per incorporare queste informazioni nella funzione di similarità è utilizzare un feedback esplicito riguardo la similarità e dissimilarità degli oggetti da confrontare. Dato un insieme di feedback è possibile utilizzare una forma chiusa della funzione di similarità con dei parametri associati che vanno appresi. Un esempio è la norma L_p pesata:

$$Dist = (\overline{X}, \overline{Y}) = \left(\sum_{i=1}^{d} a_i \cdot |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Con $\Omega = (a_1, \ldots, a_d)$ parametri da apprendre. Bisogna definire quindi una funzione di similarità $f(O_i, O_j, \Omega)$ con Ω parametri da apprendere. Si assume che valori più grandi della funzione rappresentino una maggiore dissimilarità (distanza). Definito il feedback come:

$$S = \{(O_i, O_j) : O_i \text{ è simile a } O_j\} \quad D = \{(O_i, O_j) : O_i \text{ non è simile a } O_j\}$$

Si determinano i parametri seguendo le seguenti condizioni:

$$f(O_i, O_j, \Omega) = \begin{cases} 0 & if \quad (O_i, O_j) \in S \\ 1 & if \quad (O_i, O_j) \in D \end{cases}$$

Posso quindi esprimere la scelta dei parametri come una problema di ottimizzazioni dei minimi quadrati con il seguente errore:

$$E = \sum_{(O_i,O_j)\in S} (f(O_i, O_j, \Omega) - 0)^2 + \sum_{(O_i,O_j)\in D} (f(O_i, O_j, \Omega) - 1)^2$$

Capitolo 3

Associazioni di variabili

3.1 Introduzione

I metodi per la ricerca di dipendenza tra variabili si distinguono in base a:

1. Numero di variabili: due o multiple variabili
2. Tipo delle variabili: quantitative o qualitative

3.2 Dipendenza tra variabili quantitative

3.2.1 Coefficente correlazione di Pearson

Un metodo per valutare la dipendenza tra due variabili quantitative è il coefficente di correlazione di Pearson, con ρ_{XY} coefficente relativo alla popolazione e r_{XY} coefficente relativo al campione. (Fare riferimento all'Appendice 2 per il significato dei simboli utilizzati)

$$\rho_{XY} = \frac{\sigma_{xy}}{\sqrt{\sigma_x^2 \sigma_y^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \qquad r_{XY} = \frac{S_{xy}}{\sqrt{S_x^2 S_y^2}} = \frac{S_{xy}}{S_x S_y}$$

Il valore varia tra -1 e +1, dove un valore vicino ai due estremi indica una forte associazione lineare, mentre un valore pari a 0 indica totale assenza di correlazione.

Interpretazione geometrica

$$x' = \begin{bmatrix} x_1 - \overline{x} \\ \dots \\ x_N - \overline{x} \end{bmatrix} \quad y' = \begin{bmatrix} y_1 - \overline{y} \\ \dots \\ y_N - \overline{y} \end{bmatrix} \quad r_{xy} = \frac{x' \cdot y'}{\|x'\| \, \|y'\|} = cos(\alpha)$$

Con α angolo tra i due vettori x' e y'

3.2.2 Matrici di correlazione

Nel caso di n variabili quantitative $X_1, \dots, X_n$ è possibile calcolare la correlazione di ogni coppiadi variabili creando una matrice $n \times n$ simmetrica chiamata matrice di correlazione

$$\rho \in \mathbb{R}^{n \times n} \quad \{\rho\}_{ij} = \rho_{x_i x_j}$$

3.2.3 Correlazione parziale

Quando si hanno più variabili si può incontrare il fenomeno della correlazione parziale, che si verifica quando la correlazione tra due variabili è influenzata dalla presenza di una terza (o più) varibile. La correlazione parziale è la misura di associazione tra due variabili dopo aver controllato l'effeto di una o più variabili addzionali. Questa può essere calcolata a partire dai coefficenti di correlazione delle coppie di variabili e non necessità dei dati iniziali.

$$r_{xy.z} = \frac{r_{xy} - (r_{xz} \cdot r_{yz})}{\sqrt{1 - r_{xz}^2} \cdot \sqrt{1 - r_{yz}^2}}$$

L'utilizzo della correalazione parziale permette di evitare correlazioni errate tra variabili e permette di avere una visione migliore delle correlazioni di base dei dati. é possibile l'utilizzo di modelli grafici per esprimere indipendenze condizionali delle variabili. Il coefficente di correlazione parziale tra le variabili X_1 e X_2 quando l'effetto di $X_j, X_{j+1}, \ldots, X_n$ è ricorsivamente:

$$\rho_{12.j,j+1,\ldots,n} = \frac{\rho_{12.j,j+1,\ldots,n} - (\rho_{1j.j,j+1,\ldots,n} \cdot \rho_{2j.j+1,\ldots,n})}{\sqrt{1 - \rho_{1j.j+1,\ldots,n}^2} \cdot \sqrt{1 - \rho_{2j.j+1,\ldots,n}^2}}$$

3.2.4 Regressione e correlazione

La regressione lineare permette di descrivere la relazione lienare tra una variabile predittrice (asse x) e una variabile di risposta (asse y)

$$Y = \beta_0 + \beta_1 X \quad Y = \mu_Y + \rho_{XY} \frac{\sigma_Y}{\sigma_X}(X - \mu_X)$$

$$\beta_0 = \mu_Y - \beta_1 \mu_X \quad \beta_1 = \frac{\sigma_{XY}}{\sigma_X^2} = \rho \frac{\sigma_Y}{\sigma_X}$$

Se le variabili X e Y sono normalizzate allora:

$$Y' = \rho_{XY} X'$$

Nei modelli di regressione multipla:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

I coefficenti β_i sono legati alla correlazione parziale, in particolare, nel caso di variabili standardizzate a varianza unitaria (anche non normalizzate) $\beta_i = \rho_{Yi.1,2,\ldots,i-1,i+1,\ldots,p}$

3.3 Dipendenza tra variabili qualitative

3.3.1 Tabelle di contingenza

Date due varibili qualitative X con n possibili valori e Y con m possibili valori, una tabella di contingenza è una matrice $n \times m$ tale che C_{ij} contiene la frequenza assoluta delle occorrenze del valore x_i e y_j contemporaneamente.

	y_1	y_2	y_3
x_1	12	12	56
x_2	45	23	90
x_3	84	8	11

3.3.2 Chi quadro

Sia f_o^i le frequenze osservate in ogni cella e f_e^i le frequenze attese calcolate con $f_e^i = \left(\frac{c_i r_i}{N}\right)$ con c_i somma delle frequenze sulla i-esima colonna e r_i somma delle frequenze sulla i-esima riga, si definisce la misura chi quadro:

$$\chi^2 = \sum_i \frac{(f_o^i - f_e^i)^2}{f_e^i}$$

Misura simmetrica, ma non fornisce una misura della forza della relazione statistica tra le due variabili. Date due tabelle con la stessa grandezza di campionamento, un valore più alto del chi quadro indica una correlazione più forte, questo non è vero con tabelle di dimensione diversa, portando a indicatori errati della grandezza delle relazioni tra le variabili. La misura chi quadro in questo caso è dipendente da modifiche lineari ai valori delle tabelle (moltiplicazioni, etc) che portano ad un aumento della misura chi quadro (anche se la correlazione non è aumentata)

3.3.3 Lambda

Misura di associazione asimmetrica, misura la percentuale di miglioramento nell'abilità di predire il valore della variabile dipendente una volta conosciuto il valore della variabile indipendente. Questa misura si basa sull'assunzione che la migliore strategia per la predizione è quella di selezionare il valore con più occorrenze, in quanto minimizzerà il numero di pridizioni errate.

$$\lambda(y|x) = \frac{\sum_x max_y f_{xy} - max_y f_{\cdot y}}{N - max_y f_{\cdot y}}$$

Il valore massimo è 1.0 che avviene quando la predizione può essere fatta senza errore (ogni variabile indipendente è associata con un singolo valore della variabile dipendente). Un valore di 0 indica nessun miglioramento nella predizione.

3.3.4 Spearman rank correlation coefficient

Indice di correlazione tra due variabili ordinate per ranghi. Ogni valore della variabile ha un rango associato che permette di definire un ordinamento. Sia D_i la differenza la il rango del valore X_i e Y_i (X e Y variabili aleatorie). Allora:

$$r_s = 1 - \frac{6\left[\sum_i D_i^2\right]}{n(n^2-1)}$$

Il valore lambda varia tra -1 e +1 con un valore assoluto vicino ad 1 che indica una associazione forte tra le variabili. Si dimostra che la misura lambda equivale al coefficente di correlazione di spearman quando i valori delle osservazioni sono sostituiti da ranghi ordinati. Viene fornito quindi un nuovo modello alternativo per il calcolo:

1. Ordina i valori delle variabili X e Y
2. Rimpiazza i valori con i ranghi ordinati
3. Calcola il coefficente di correlazione di Pearson sui ranghi

3.3.5 Variabili multiple

Nel trattare variabili multiple a stato dell'arte è preferibile modelli log-lineari quando le variabili sono di tipo nominale o ordinale. In questi modelli non è fatta differenza tra variabili dipendenti e indipendenti, vengono quinidi solo mostrate le associazioni tra le variabili. La strategia alla base dei modelli log-lineari è di addestrare un modello sullo frequenze osservate nella tabulazione delle variabili categoriche. Una volta stimati i parametri delle misure permettono di valutare l'associazione delle varibili (in generale valori maggiori di uno indicano correlazione positiva, minori di uno associazione negativa).

Limitazioni Difficile interpretazione a causa delle molte variabili. In media è necessario avere almeno un numero di campioni pari a 5 volte il numero di celle della tabella. Complessità computazionale.

Capitolo 4

Association Pattern Mining

4.1 Introduzione

Il goal dell'association pattern mining è di scoprire associazioni tra gruppi di oggetti, che possono essere viste come correlazioni a k vie tra gli oggetti. I modelli più popolari utilizzano le frequenze dei set di item per quantificare il livello di associazione. L'association pattern mining viene utilizzato nei seguenti campi:

- Dati supermercati o negozi online: Prima applicazione che ha motivato la nascita di questa tipologia di mining.
- Text mining: Permette di indentificare co occorezza di termini e parole chiave.
- Generalizzazione di dati orientati alle dipendenze (dependency oriented): Il modello di association pattern mining è stato esteso a tipi di dato come serie temporali, serie sequenziali, dati spaziali e grafi, questi modelli vengono utilizzati in applicazioni come web log analysis, bug detection e ritrovamento di eventi spazio temporali.
- Utilizzato come subroutine in altri problemi di data mining come clustering classificazione e outlier analysis.

I dati vengono chiamati *transazioni* mentre gli output sono chiamati *itemset*, questa terminologia viene presa in prestito dalla market basket analysis (prima vera applicazione dei modelli di pattern mining). Un pattern frequente viene quindi definito come un sottoinsieme frequente dell'universo di tutti i possibili insiemi. Gli itemset frequenti possono essere utilizzati per generare regole di associazione come $X \implies Y$ dove X e Y sono set di items. E' presente una direzionalità dell'implicazione che è quantificata come una probabilità condizionale. I modelli basati su frequenza sono popolari grazie alla loro semplicità, ma la frequenza grezza dei pattern non è rilevante quanto il significato statistico delle correlazioni sottostanti (correlazioni nascoste).

4.2 Modello frequent pattern mining

Dataset Il problema è definito su dati insiemistici non ordinati. Sia T il database contenente n transazioni chiamata $T_1, \ldots, T_n$. Dove $T_i \in U$ universo degli item, che può essere visto come un record multidimensionale di dimensione $d = |U|$ contentente solo attributi binari ognuno rappresentate un particolare item, il valore binario sarà 1 se l'item è contenuto nella transazione, 0 altrimenti. Nelle applicazioni reali $|U| >> |T_i|$ (una transazione contiene un sottoinsieme molto ridotto di tutti gli item). Ogni transazione possiede un *transaction identifier* o *tid* identificatore univoco della transazione. Un database di transazione può essere rappresentato come un database multidimensionale binario la cui dimensione è uguale al numero di items in U

Itemset Un itemset è un insieme di item. Un k-itemset è un itemset che contiene esattamente k item.

Supporto Il supporto di un itemset I è definito come la frazione di transanzioni nel database T che contengono I come sottoinsieme. Si denota con $sup(I)$

Supporto minimo Dato un insieme di transazioni T, si definisce frequent itemset mining il processo di determinare tutti gli itemset I che occorono con supporto almeno *minsup* all'interno delle transazioni di T. Il valore minsup può essere espresso sia come valore frazionale relativo sia come valore intero assoluto (numero grezzo di transazioni). Il numero di itemset frequenti è molto sensibile al livello di supporto minimo, un supporto minimo molto basso genererà molti itemset frequenti, una soglia più alta produrrà un insieme di itemset frequenti ridotto.

Proprietà: Monotonia del supporto Se un itemset è contenuto in una transazione, lo saranno anche tutti i suoi sottoinsiemi, quindi il supporto di ogni sottoinsieme J di I avrà la seguente proprietà

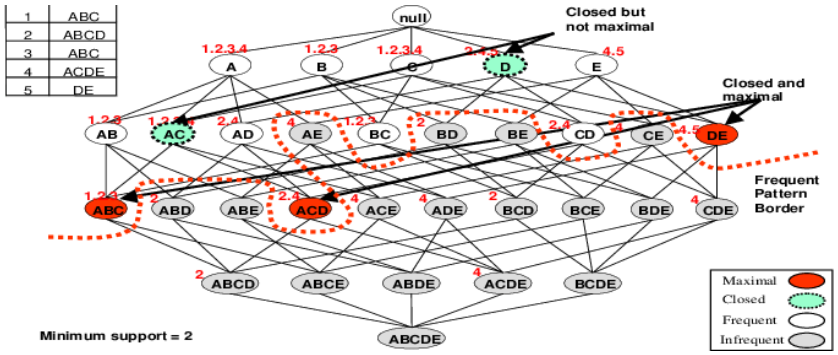
$$sup(J) \geq sup(I) \quad \forall J \subseteq I$$

La proprietà di monotonia implica la seguente proprietà:

Proprietà: Chiususra verso il basso Ogni sottoinsieme J di un itemset frequente I è anch'esso frequente. Questa proprietà rappresetna un vincolo nella struttura dei pattern frequenti, permettendo quindi di effettuare potature del processo di ricerca degli itemset frequenti. Può essere inoltre utilizzata per fornire rappresentazioni concise dei pattern frequenti.

Itemset frequenti massimali Un itemset è massimale ad un dato livello di minsup se è frequente e nessun suo superset è frequente. Tutti gli itemset frequenti possono essere derivati dai pattern massimali tramite enumerazione dei loro sottinsiemi. Un pattern massimale può essere quindi considerato una rappresentazione compatta dei pattern frequenti che non mantiene però informaizoni sui valori di supporto dei suoi sottoinsiemi.

Lattice degli itemset Tutti gli itemset possono essere organizzati in una struttura chiamata lattice degli itemset. Questa struttura a grafo contiene un nodo per ognuno dei $2^{|U|}$ sottoinsiemi dell'universo U, un arco connette due nodi se questi differiscono per un solo item. Gli algoritmi di pattern mining effettuano la loro ricerca sul lattice degli itemset per determinare i pattern frequenti. Il lattice separa l'insieme degli itemset frequenti da quelli non frequenti tramite un confine. Gli itemset frequenti massimali sono adiacenti al confine.



4.3 Generazione delle regole di associazione

Gli itemset frequenti possono generare regole di associazione tramite l'utilizzo di una misura di *confidenza*

Confidenza Siano X e Y due insiemi di item, la confidenza $conf(X \cup Y)$ della regola $X \cup Y$ è la probabilità condizionale che $X \cup Y$ avvenga in una transazione, conoscendo che la transazione contiene X. Quindi la confidenza di $X \implies Y$ è definita:

$$conf(X \implies Y) = \frac{sup(X \cup Y)}{sup(X)}$$

I due itemset X e Y si chiamano rispettivamente antecedente e conseguente della regola di associazione.

Proprietà: Monotonia della confidenza Siano X_1, X_2, I itemset tali che $X_1 \subset X_2 \subset I$. Allora:

$$conf(X_2 \implies I - X_2) \geq conf(X_1 \implies I - X_1)$$

Questa proprietà permette di generare solo regole di associazioni non ridondanti.

Regola di associazione Siano X e Y due insiemi di item, allora la regola $X \implies Y$ è definita una regola di associazione per un deterimnato supporto minimo minsup e confidenza minimia minconf se:

1. Il supporto di $X \cup Y$ è $sup(X \cup Y) \geq minsup$
2. La confidenza della regola $X \implies Y$ è $conf(X \implies Y) \geq minconf$

La confidenza minima permette di generare le regole di associazione più rilevanti. Il primo criterio assicura che esita un numero sufficente di transazioni rilevanti per la regola, il secondo criterio si assicura che la regola sia abbastanza forte in termini di probabilità condizionata.

4.3.1 Framework generale

Il framework generale per la generazione di regole di associazione è diviso in due fasi:

1. Vengono generati tutti gli itemset frequenti con supporto almeno minsup
2. Le regole di associazione con confidenza almeno minconf sono generate dagli itemset frequenti

La prima fase è computazionalmente più pesante. Per ogni $I \in F$ insieme degli itemset frequenti è possibile generare regole di associaizone partizionando I in tutte le possibili combinazioni di set $X, Y = I - X$ tali che $I = X \cup Y$, le regole di associazione $X \implies Y$ possono essere generete utilizzando la confidenza minima e la proprietà di monitonia della confidenza.

4.4 Algoritmi di pattern mining

Gli algoritmi che verranno presentati utilizzano dei tricks per aumentare l'efficenza della generazione degli itemset frequenti che possono essere riutilizzani in algoritmi diversi in quanto quasi tutti gli algoritmi utilizzano (o possono essere ricondotti) allo stesso framework di enumerazione dell'albero degli itemset.

4.4.1 Approccio generale

Gli algoritmi di pattern mining ricercano il lattice degli itemset per gli itemset candidati utilizznado il database delle transazioni per calcolare il supporto. Il calcolo del supporto chiamato *support counting* richiede di confrontare un itemset con tutte le transazioni per controllare se questa è contenuta al suo interno. Una maggiore efficenza può essere raggiunta utilizzando uno dei seguenti approcci:

1. Ridurre la grandezza dello spazio di ricerca attraverso strategie di pruning dei candidati (molto spesso viene utilizzata la proprietà di chiusura verso il basso)
2. Applicare un pruning delle transazioni in modo da rendere il support counting più efficente.
3. Utilizzare strutture dati compatte per rappresentare i candidati o la transazioni, sia per la generazione degli itemset frequenti sia per il support counting.

4.4.2 Brute force

Dato un universo degli item U ci sono $2^{|U|} - 1$ possibili combinazioni di itemset creabili (escludendo l'itemset vuoto). Una possibile soluzione è generare tutti questi itemset candidati (itemset che potrebbero essere degli itemset frequenti)

e calcolare il loro supporto per trovare gli itemset frequenti. . Un modo per rendere questo metodo più veloce è l'utilizzo della chiusura verso il passo, che ci permette di escludere tutti i pattern di grandezza $k+1$ se nessun pattern di grandezza k è frequente. Vengono quindi generati itemset fino alla grandezza l portando lo spazio di ricerca a $\sum_{i=1}^{l} \binom{|U|}{i} << 2^{|U|}$

4.4.3 Apriori

Viene utilizzate la proprietà di chiususra verso il basso per effettuare un pruning dello spazio di ricerca dei candidati. Gli itemset non frequenti vengono utilizzati per filtrare la generazione dei superset candidati (non ha senso contare il supporto dei superset candidati di un itemset non frequente). L'algoritmo apriori genera candidati di grandezza k e conta il loro supporto prima di generare i candidati di grandezza k+1, i k-itemset frequenti vengono utilizzati per restringere il numero di k+1 itemset superset candidati che vengono generati. Si assume un ordine lessicografico degli item dell'universo U. La generazione dei candidati e il support counting dei pattern sono interlacciati in questo algoritmo. Gli algoritmi che contano il supporto dei candidati di dimensione crescente vengono chiamati *level-wise*.

Algoritmo Apriori Sia F_k l'insieme dei k-itemset frequenti e sia C_k l'insieme dei k-itemset candidati. I candidati di lunghezza k+1 in C_{k+1} vengono generati a partire dagli itemset frequenti F_k già trovati dall'algoritmo, le frequenza dei k+1 candidati vengono calcolate tramite il database delle transazioni. Lo spazio di ricerca per la generazione dei k+1 candidati può essere potato controllando se tutti i k-subset di C_{k+1} sono contenuti in F_k.

Se $X, Y \in F_k$ hanno (k-1) item in comune allora un join dei due itemset utilizzando i (k-1) item comuni viene utilizzato per generare un candidato di dimensioni (k+1). L'utilizzo di un ordine lessicografico permette di evitare ridondanze nella generazione degli itemset tramite join dei primi (k-1) item in comune. La proprietà di chiusura verso il basso ci assicura che vengano generati tutti gli itemset candidati (non verrà escluso nessun itemset davvero frequente)(il join viene effettuato solo tra itemset frequenti).

Questo approccio esaustivo e non ripetitivo di generazione degli itemset può essere interpretato tramite l'utilizzo di alberi di enumerazione. Il join degli itemset è molto efficente in quanto, grazie all'ordinamento lessicografico, tutti gli itemset con i primi (k-1) item comuni appariranno in posizioni contigue, permettendo una ricerca facile.

Level wise pruning trick Tutti i k-subset di un itemset $I \in C_{k+1}$ devono essere presenti in F_k per la proprietà di chiusura verso il basso, altrimenti è garantito che I non sia frequente.

Costo computazionale L'algoritmo apriori è efficente dal punto di vista di accessi al disco. In quanto il support counting dei candidati C_{k+1} può essere

effettuato in un singolo passaggio sui dati senza accessi randomici al disco. Il numero di passaggi sui dati è uguale alla cardinalità dell'itemset frequente più lungo.

4.4.3.1 Efficient support counting

Per rendere più efficente la ricerca degli itemset all'interno delle transazioni l'algoritmo apriori utilizza una struttura dati chiamata *hash tree*. Una struttura dati ad albero un prefissato grado di nodi interni. Ogni nodo interno ha associata una funzione hash che permette di indicizzare i diversi nodi figlio del nodo interno. Ogni nodo interno contiene una hash table che permette di indicizzare i nodi successivi (figli), ogni nodo foglia contiene un'insieme di itemset ordinati lessicograficamente. Ogni itemset in C_{k+1} è contenuto in esattamente una foglia dell'albero. Le funzioni hash permettono di definire a che foglia appartiene ogni itemset.

1. Viene fissato il valore h, massimo numero di figli per ogni nodo interno (grado di ramificazone dell'albero).
2. Ad ogni nodo interno viene associata una funzione $f(\cdot) : U \rightarrow [0, h-1]$ con U universo degli item. Ogni itemset viene assegnato ad una foglia tramite un percorso dalla radice alla foglia.
3. All'i-esimo livello dell'albero, la funzione $f(\cdot)$ hash viene applicata all'i-esimo elemento dell'itemset $I \in C_{k+1}$ in modo da decidere che percorso seguire per assegnare l'itemset ad un nodo foglia.
4. Viene imposta una soglia massima di itemset in un nodo foglia.

L'albero viene costruito in maniera top-down tramite i seguenti passaggi applicati ad ogni $I \in C_{k+1}$:

1. Sia k il livello attuale dell'albero (a partire da 1), I_k k-esimo elemento di I
2. Applico la funzione hash all'k-esimo elemento $f(I_k) = n$ e ho come risultato il nodo n
3. Mi sposto all'n-esimo nodod figlio del cnodo corrente
4. Se l'n-esimo nodo è un nodo foglia allora inserisco l'itemset I, altrimenti incremento k e torno allo step 2
5. Se il nodo foglia è pieno viene divide il nodo utilizzando il metodo descritto nei passaggi precedenti.

In una sola esplorazione dell'albero vengono trovati tutti gli k-itemset candidati di C_{k+1} che sono subset della transazione $T_j \in T$. Vengono esplorate tutti i possibili percorsi dell'albero le cui foglie potrebbero contenere dei subset candidati attraverso una ricerca ricorsiva sull'albero.

Ad un dato nodo interno, se l'ultimo elemento sottoposto ad hash è l'i-esimo elemento della transazione T_j allora tutti gli item successivi verranno sottoposti ad hash per determinare i possibili nodi figlio da seguire (dato l'item corrente, vengono cosi provate tutte le combinazioni con gli item successivi, in modo da cercare tra tutti i possibili subset)

4.4.4 Alberi di enumerazione

Questi algoritmi generano gli itemset candidati tramite un struttura ad albero chiamata *albero di enumerazione* che è un sottografo del lattice degli itemset. Questo è anche chiamato albero lessicografico in quanto impone un ordine lessicografico degli item. La caratteristica principale di questa struttura è che fornisce una rappresentazione astratta gerarchica degli itemset, questa viene utilizzata dagli algortimi di pattern mining per effettuare una esplorazione non ripetitiva e sistematica dei pattern candidati.

1. Il nodo dell'albero rappresetna l'itemset null (vuoto). L'albero ha un nodo per ogni itemset frequente.
2. Sia $I = \{i_1, \ldots, i_k\}$ un itemset frequente con i suoi elementi ordinati in ordine lessicografico. Il genitore del nodo I è l'itemset $\{i_1, \ldots, i_{k-1}\}$. Quindi un nodo figlio rappresenta un estensione del nodo gentitore effettuato solo con item che occorrono lessicograficamente dopo tutti gli item presenti nel nodo genitore. Per questo l'albero di enumerazione può essere visto come un albero di prefissi delle stringhe che rappresentano gli itemset.

Un item utilizzando per estendere un nodo nel suo nodo figlio p chiamato *frequent tree extension*. La principale differenza con il lattice degli itemset che è questo genera diversi percorsi per estendere un nodo (rappresetnare un itemset), mentre l'albero di enumerazione genera un solo percorso per ogni itemset. L'ordine lessicografico impone una struttura gerarchica degli itemset permettendo una ricerca esaustiva e non ridondante.

4.4.4.1 Framework generico enumeration tree

Il nodo radice viene esteso cercando gli 1-itemset frequenti, sucessivamente questi possono essere estesi per generare itemset candidati, questi vengono poi confrontati con il database delle transazioni per il support counting. Sia P la notazione per indicare un itemset e il rispettivo nodo all'interno dell'albero.

Per la proprietà di chiususra verso il basso, un item i viene condiderato un candidato all'estensione del nodo P a $P \cup i$ solo se è una frequent tree extension del Q genitore di P.

- Sia $F(Q)$ l'insieme delle frequent tree extensions di Q
- Sia $i \in F(Q)$ item che estende il nodo Q in $P = Q \cup I$

- Sia $C(P)$ il sottoinsieme di item di $F(Q)$ che occorrono lessicograficamente dopo l'item i (utilizzato per estendere Q in P)
- $C(P)$ rappresenta gli item che sono estensioni candidate del nodo P

Possiamo notare che $F(P) \subseteq C(P) \subset F(Q)$. Un insieme C di estensioni candidate può ovviamente contenere estensioni non frequenti. Si procede dunque:

1. Sia ET l'albero di enumerazione
2. Seleziono uno o più nodi $P \in ET$
3. Determino le estensioni candidate $C(P) \quad \forall P$ scelti
4. Support counting
5. Aggiungo i candidati frequenti a ET

Metodo generale di crescita dell'albero, eseguito finchè non è più possibile estendere nodi.

4.4.4.2 Strategie di crescita

Nelle strategie di ricerca in ampiezza l'insieme dei nodi considerati al passo 2 dell'algoritmo sono tutti i nodi di un livello dell'albero. Questo approccio è rilevante per database allocati su disco in quanto tutti i nodi di un livello possono essere estesi un una sola passata di counting sul database. Nelle strategie in profondità selezionano un singolo nodo al livello più profondo al passo 2 dell'algoritmo. Queste strategie permettono di esplorare l'albero in profondità e trovare fin da subito lunghi pattern frequenti, questa scoperta iniziale e veloce di pattern frequenti lunghi è particolamente utile per la sua efficenza nel maximal pattern mining e per migliorare la gestione della memoria negli algoritmi basati su proiezioni.

4.4.4.3 Interpretazione di Apriori con enumeration tree

L'algoritmo apriori può essere visto come una costruzione per livelli dell'albero di enumerazione tramite stratezione in ampiezza. Il join effettuato da apriori per generare (k+1) itemset utilizzando solo i primi (k-1) item dei k-itemset frequenti può essere visto come il join di tutte le coppie di sibling immediatemente contigue ad un livello k dell'albero di enumerazione. Queste operazione di join sul nodo P permette quindi di generare dei candidati che sono l'estensione di P con ognuno degli item in C(P), il pruning dell'algoritmo apriori permette poi di eliminare i nodi dell'albero di enumerazione che è sicuro non siano frequenti. Un singolo passaggi sul database delle transazione permette quindi di generare le estensioni frequenti F(P). L'algoritmo apriori implicitamente estende l'albero di enumerazione livello per livello attraverso l'uso di join.

4.4.4.4 TreeProjection e DepthProject

TreeProjection è una famiglia di metodi che utilizza proiezioni ricorsive delle trnsazioni sull'albero di enumerazione. L'obiettivo di queste proiezioni è di riutilizzare il support counting fatto ad un dato nodo dell'albero di enumerazione per i suoi nodi discendenti. TreeProjection è un framework generale che mostra come utilizzare projizioni di database nelle strategie di costruzione dell'albero di enumerazione. L'approccio depthproject è una forma di questo framework che utilizza una strategia in profondità. L'osservazione principale dei metodi basati su proiezioni è che se una transazione non contiene l'itemset corrispondente di un nodo, allora la transazione non sarà rilevante per il support counting dei nodi figlio (o dei genitori). Queste strategie utilizzano la nozioni di *projeted databaes*, in cui ogni projected transaztion è legata (specifica) ad un nodo dell'albero di enumerazione. Le transazioni che non includono l'itemset P non sono incluse nel projected database del nodo P e dei suoi discendenti.

1. Sia E(P) l'insieme delle frequent lexicographic tree extensions del nodo P.
2. Data una transazione T, la transazione proiettata T(P) sarà uguale a $T \cap E(P)$, ma se T non contiene l'itemset corrispondente al nodo P allota T(P) è vuoto.
3. Per un insieme di transazioni T, l'insieme delle transazioni proiettate $\mathbb{T}(P)$ è definito come l'insieme delle transazioni proiettate T rispetto all'itemset frequente E(P).

4.4.5 Recursive suffix-based pattern growth

Gli alberi di enumerazione sono costruiti tramite estensioni dei prefissi degli itemset, questi metodi invece permettono di generare gli itemset tramite l'estensione dei suffissi, questi metodi possono essere visti come caso speciale degli algoritmi basati su alberi di enumerazione. I metodi di estensione del suffisso utilizzano una speciale struttura dati chiamata *Frequent Pattern Tree* (FP-Tree) ma possono essere implementati anche tramite l'utilizzo di array e puntatori.

4.4.5.1 Approccio agnostico alla struttura dati

Sia T il database delle transazioni espresso solo in termini di 1-item frequenti (da ogni transazione vengono rimossi gli item singoli che non sono frequenti). Gli item nel databse vengono ordinati con supporto decrescente, sarà questo l'ordine lessicografico usato per definire l'ordine degli itemset e delle transazioni. L'algoritmo ha come input il dataset T, il suffiso frequente corrente P (alla prima chiamata è vuoto) e il valore di support minimo minsup. L'obiettivo della chiamata ricorsiva è trovare tutti i pattern frequenti che hanno come suffisso P. In ogni chiamata ricorsiva gli step sono:

1. Costruisco l'itemset $P_i = P \cup \{i\}$ concanenando P con ogni item i (i viene aggiunto all'inizio di p) nel database delle transazioni T e riportando

l'itemset come frequente. L'itemset P_i è frequente in quanto T è definito in termini di item frequenti del databse proiettato del suffisso P.

2. Creo il database proiettato T_i di P_i. Vengono estratte tutte le transazioni da T che contengono i. Tutti gli item che sono lessicograficamente maggiori o uguali di i vengono rimossi dalle transazioni in T_i. La frequenza di ogni item in T_i viene contata e gli item non frequenti vengono rimossi dalle transazioni.

3. Per ogni item i, viene esteso ulteriormente P_i tramite una chiamata ricorsiva.

4.4.5.2 FPGrowth

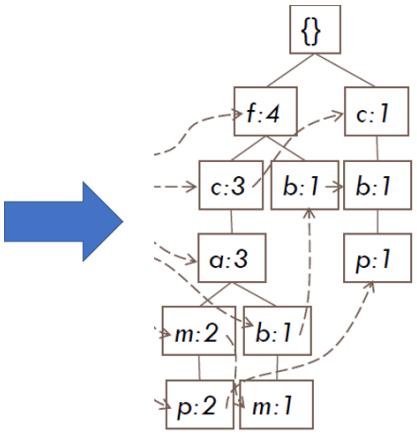
La struttura dati FP-Tree è stata create con l'obiettivo principale di efficenza di spazio nei databse proiettati. Il percorso dal nodo radice ad ogni nodo interno rappresenta una transazione o il prefisso di una transazione nel database. Ogni nodo interno ha associato un valore che rappresenta il numero di transazioni nel database che contengono il prefisso rappresentato dalla path. Viene utilizzato il supporto degli 1-item come ordine lessicografico degl iitem, dal più frequente al meno frequente, questo permette di sfruttare al meglio la compressione basata su suffissi.

1. Sia FPT la struttura dati iniziale FP-Tree

2. Vengono rimossi dalle transaction del database tutti gli item non frequenti

3. Le transazioni sono inserite nell'albero, ogni nodo rappresenta un item, ad ogni successiva transazione inserita, si segue il percorso da nodo in nodo per ogni item nella transazione, se un nodo è stato già creato viene aumetnato il contatore di uno per quel nodo, altrimeni si crea un nuovo nodo figlio.

TID	Items bought
100	{a, c, d, f, g, i, m, p}
200	{a, b, c, f, i, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, c, e, f, l, m, n, p}



Vengono effettuate solo due scan del database, la prima per trovare gli 1-itemset frequenti e la seconda per costruire la struttura dati. Per ogni item $i \in FPT$ si costruisce l'albero condizionale FPT_i che rappresenta il database condizionale T_i, questo per inizializzare la chiamata ricorsiva dell'algoritmo per generare gli itemset frequenti.

1. Viene estratto l'albero delle conditional prefix paths per ogni item i. Questo albero è formato da tutte le path dall'item i fino alla radice. Tutti gli altri percorsi vengono potati

2. I contatori dei nodi del nuovo albero condizionale vengono aggiornati in modo da tener conto dei percorsi potati, questo può essere fatto aggregando i contatori proseguendo dalle foglie in su. L'insieme delle path dalla radice al nodo i sono chiamate conditional pattern base.

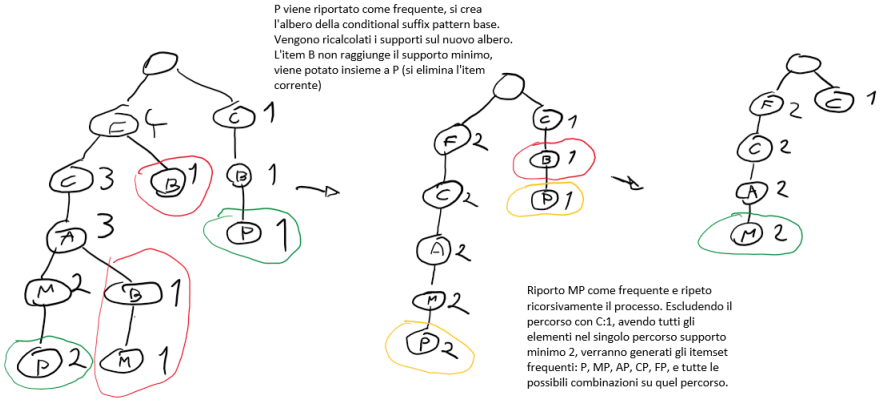
3. La frequenza di ogni item è contata aggregando i contatori di tutte le occorrezze dell'item nell'albero. Ogni item che non supera la soglia di minsup viene rimosso dalla conditional pattern base, viene inoltre rimosso l'item i da ogni prefix path.

4. Viene ricreato l'albero utilizzando la nuova conditional pattern base.

5. Viene controllato se l'albero è vuoto (questo accade quando non ci sono item frequenti), se non è vuoto si procede con la successiva chiamata ricorsiva con il suffisso $P_i = P \cup i$

L'utilizzo di FP-Tree permette di definire una condizione di restrizione che permette di estrarre pattern frequenti velocemente ai livelli di ricorsione profondi. Se tutti i nodi gioaciono in un singolo percorso allora i pattern frequenti possono essere estratti direttamente estraendo tutte le combinazioni di nodi sul percorso con i relativi conteggi del supporto aggregati.

Vantaggi e confronto con Apriori Utilizza il principio di pruning di apriori, il database viene scannerizzato solo due volte, non c'è una fase di generazione dei candidati ne test sui candidati, il database viene espresso in forma compressa tramite gli fp-tree, apriori a differenza di fpgrowth applicate scansioni ripetutre del database, le operazioni utilizzate sono basilari, conteggio locale degli item frequenti e costruzioni dei sottoalberi condizionali, non viene effettuato pattern search e matching.



4.5 Interesting patterns models

Gli algoritmi esposti fin'ora utilizzano la proprietà di chiusura verso il basso per definire algoritmi efficenti per il pattern mining, questa efficenza e convenienza algoritmica non significa che questi algortimi siano performanti da un prospettiva specifica dell'applicazione, in quanto le frequenze grezze degli item e itemset non corrispondono sempre a un indicatore di pattern interessante (si frequente, ma non interessante).

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

Ad esempio nella figura l'item milk appare in tutte le transazioni del database, questo significa che ogni regola di associazione del tipo $X \implies \{milk\}$ avrà confidenza del 100%. Ovviamente utilizzando metodi basati su confidenza e support counting questa regola di associazione è si vera, ma non interessante da un punto di vista informativo (non può essere utilizzata in alcun modo). Questo è un esempio che dimostra le limitazioni dei modelli basati su supporto e confidenza. In alcune applicazini è desiderabile crare delle misure che posano addattarsi alla skew nei valori di supporto dei singoli item, queste misure sono particolarmente importanti nel negative pattern mining.

Bit-Simmetria Dato un itemset e il corrispetivo itemset negato ($\{mik, butter\}$, $\{\neg milk, \neg butter\}$) , il coefficente di correlazione statistico tra i due itemset sarà identico, quindi una misura di associazione dobrebbe misurare l'associazione tra le due tipologie di itemset nello stesso modo. o Gli 0 e gli 1 nella matrice binaria

dei dati sono trattati nello stesso modo.

Questa tipologia di modelli è chiamata interestingness-based models, con la particolarità che non è più valida la proprietà di chiusura verso il basso, questo non permette di progettare algortimi efficenti in spazi di ricerca molto grandi, in molti casi queste misure sono definite solo per i 2-itemset.

4.5.1 Coefficente statistico di correlazione

Siano X, Y due variabili:

$$\rho = \frac{E[X \cdot Y] - E[X]E[Y]}{\sigma(X)\sigma(Y)}$$

Nella market baskeet analysis X,Y sono variabili binarie che indicano la presenza o assenza di item. E rappresentea l'attesa e σ la deviazione standard. Definiti $sup(i), sup(j)$ supporti degli item individuali i,j e $sup(i, j)$ il supporto dell'itemset $\{i, j\}$:

$$\rho_{ij} = \frac{sup(i, j) - sup(i) \cdot sup(j)}{\sqrt{sup(i) \cdot sup(j) \cdot (1 - sup(i)) \cdot (1 - sup(j))}}$$

Il valore varia tra -1 e +1, dove +1 indica correlazione positiva, -1 correlazione negativa e valori vicini allo 0 indica dati scarsamente correlati. Questa misura soddisfa la proprietà di bit-simmetria. Viene considerata la misura più robusta per misurare la correlazione ma ha una difficile interpretazione quando gli item hanno supporti bassi ma con alta varianza.

4.5.2 Misura χ^2

Dato un insieme di k variabili aleatorie binarie (item) definite da X, ci sono 2^k possibili stati che rappresentano la presenza o assenza degli item di X nella transazine. La freazione attesa degli item presenti in ognuna di questa combinazioni può essere quantificata come il prodotto dele supporto degli stati degli item individuali. Sia O_i il valore osservato e E_i il valore atteso del supporto in valore assoluto dello stato i, allora:

$$\chi^2 = \sum_{i_1}^{2^{|X|}} \frac{(O_i - E_i)^2}{E_i}$$

Questa misura è bit-simmetrica. Un valore vicino allo 0 indica indipendenza statistica degli item, valori più grandi indicano il grado di dipendenza delle variabili, questo valore però non da informazioni su correlazione positiva o negativa, questo perchè questa viene misurata la dipendenza tra le variabili, non la correlazione tra gli specifici stati delle variabili. Questa misura soddisfa la proprietà di chiusura verso il basso, permettendo di progettare algoritmi efficenti, ma la complessità della misura aumenta esponenzialmente in base a $|X|$.

4.5.3 Interest ratio

Misura semplice e facilmente interpretabile. Sia $\{i_1, \ldots, i_k\}$ un set di item:

$$I(\{i_1, \ldots, i_k\}) = \frac{sup(\{i_1, \ldots, i_k\})}{\prod_{j=1}^{k} sup(i_j)}$$

Se gli item sono statisticamente indipendenti, il supporto del set di item a numeratore saraà uguale al prodotto dei supporto al denominatore, quindi un valore più alto di 1 indica un correlazione positiva, minore di uno una correlazione negativa. L'interest ratio produce dei risultati ingannevoli quando alcuni item sono molto rati. Se una item appare in una sola transazione del database ogni item che co occore con esso nella stessa transazione può essere utilizzato per creare un 2-itemset con un valore di interest ratio molto alto. Inoltre in quanto non viene soddisfatta la proprietà di chiusura verso il basso è difficile creare algortimi efficenti.

4.5.4 Misure di confidenza simmetriche

La tradizionale misura di confidenza è asimmetrica, ma la misura di supporto è simmetrica, questo ci permette di definire un nuovo tipo di misura di confidenza simmetrica che può sostituire il calcolo del supporto-confidenza con una sola misura. Siano X,Y 1-itemset, una misura simmetrica può essere una funzione delle due confindenze $X \implies Y$ e $Y \implies X$:

- Minimo: Non è desiderabile quando X o Y non sono molto frequenti, in quanto la misura combinata sarebbe troppo bassa.
- Massimo: Non è desiderabile quanod X o Y sono molto frequenti, in quanto la misura combinata sarebbe troppo alta.
- Media: La media può essere un tradeoff robusto.

La misura può essere generalizzata a k-itemset utilizzando tutti i possibili k item individuali come consequenti nella computazione della misura. Gli itemset rilevanti scoperti con questa misura (che superano una specifica soglia) non soddisfano la proprietà di chiusura verso il basso.

4.5.5 Coefficente coseno su colonne

Normalmente applicato sulle righe (transazioni) è possibile applicarlo sulle colonne per calcolare la similarità tra item. Questa misura viene computata utilizzando la rappresentazione tramite lista verticale dei tid dei corrispondenti vettori binari.

$$cosine(i, j) = \frac{sup(\{i, j\})}{\sqrt{sup(i)} \cdot \sqrt{sup(j)}}$$

Il numeratore può essere calcolato come la lunghezza dell'intersezione dei due vettori verticali tid. Può essere interpretata come la media geometrica della confidenza delle regole $i \implies j$ e $j \implies i$, quindi è una misura simmetrica.

4.5.6 Coefficente di Jaccard

La lista di TID in una colonna può essere vista come un insieme, e il coefficente di Jaccard permette di computare la similarità tra questi due insiemi. Siano $S_1, \ldots, S_k$ insiemi di tid di k items:

$$J(S_1, \ldots, S_k) = \frac{|\cap S_i|}{|\cup S_i|}$$

Gli itemset rilevanti possono essere trovati imponendo una soglia minima. Il coefficente di jaccard soddisfa la proprietà di set-wise monotonicity:

$$J(S_1, \ldots, S_k) \geq J(S_1, \ldots, S_{k+1})$$

Questo perchè il numeratore è non crescente all'aumentare di k, mentre il denominatore è non decrescente all'aumentare di k, quindi questa misura non può aumentare all'aumentare di k. Se viene imposta una soglia, gli itemset rilevanti soddisfano la proprietà di chiusura verso il basso. Per questa proprietà gli algoritmi tradizionali di pattern mining possono essere generalizzati al coefficente di Jaccard facilmente.

4.5.6.1 Sampling

Il sampling permette di velocizzare la computazione trasformando la computazione del coefficente di jaccard in una algortimo di pattern mining standard. Il sampling utilizzato simula dei sample randomici dei dati tramite funzioni hash. Sia D la matrice binaria $n \times d$ con n righe e d colonne e sia da calcolare il coefficente di jaccard sulle prime k colonne.

Si suppone che vengano ordinate le righe di D e viene estratta la prima riga in cui almeno una delle prime k colonne ha valore 1. Allora la probabilità delle evente che tutte le k colonne hanno valore 1 è uguale al coefficente di jaccard su k elementi (il denominatore è l'unione di tutti i tid delle transazioni, il numeratore sono solo i tid comuni a tutti gli item, quindi la probabilità che tutte le colonne abbiano valore uno, che significa che una transazione contiene tutti gli item interessati, è la frazione tra l'intersezione dei tid con l'unione dei tid, quindi il coefficente di jaccard).

Il processo di sorting delle righe viene effettuato più volte in modo da stimare la probabilità che le prime k colonne abbiano valore 1 come la frazione tra il numero di sort in cui questo evento avviene. Questo metodo è inefficente in quanto ogni sort richiede una scansione del database. Inoltre questo approccio stima la probabilità solo per un particolare set di colonne ma non permette di scoprire i k-itemset che soddisfano il criterio minimo del coefficente di jaccard.

4.5.6.2 Min-hash trick

Il min-hash trick permette di effettuare la selezione dei dati in maniera implicita transformando id ati in una rappresentazione campionata concisa, sui cui applicare gli algoritmi di pattern mining standard.

1. Una funzione di hashing random $h(\cdot)$ viene applicata ad ogni tid
2. Per ogni colonna di valori binari, viene selezionato il tid con il valore minore della funzione hash tra tutti i tid di quella colonna che hanno valore 1.
3. Viene prodotto un vettore di d differenti tids
4. La probabilità che i tids delle prime k colonne siano uguali è uguale al coefficente di jaccard in quanto il processo di hashing simula una selezione (ordinameto) e produce l'indice del primo elemento non zero nella matrice binaria
5. Ripeto il processo r volte creando un matrice categorica $r \times d$ di tids.
6. Determinando il sottoinsieme di colonne con tid uguali e con supporto uguale al livello minimo di supporto è possimire stimare tutti i sottinsiemi di k-item il cui coefficente di jaccard è almeno uguale al supporto minimo.

Utilizzando diverse funzioni hash indipendenti per generare sample multipli è possibile stimare il coefficente di jaccard. Estraendo gli identificatori delle colonne con tid uguali per ogni riga è possibile creare una nuove transazioni con gli identificatori di colonna come item. In questo modo ogni riga della tabella saraò associata a transazioni multiple, questo metodo permette di trasformare la matrice categorica in una matrice binaria e poter applicare algoritmi di pattern mining off the shelf. (Se una riga ha le colonne C1 e C3 con tid uguale e le colonne C2 e C4 con tid diversi la nuova transazione avrà gli item {C1,C3}).

4.5.7 Collective strenght

La collective strenght di un itemset è definita in termini di violation rate. Un itemset I è in violazioe di una transazione se alcuni dei suoi item sono presenti in una transazione ed altri no. Il violation rate $v(I)$ di un itemset I è la frazione delle transazioni con cui esso è in violazione, quindi:

$$C(I) = \frac{1 - v(I)}{1 - E[v(I)]} \cdot \frac{E[v(I)]}{v(I)} \in [0, \infty]$$

Un valore 0 indica un correlazione negativa perfetta, mentre un valore infinito indica correlazione positiva perfetta. Il valore 1 è il breakpoint. $E[v(I)]$ è calcolato assumendo l'indipendenza degli item individuali. Se p_i è il numero di transazioni in cui occorre l'item i:

$$E[v(I)] = 1 - \prod_{i \in I} p_i - \prod_{i \in I} (1 - p_i)$$

La violazione di una transazione può esser vista come un evento negativo, il contrario un evento positivo ($1 - v(I)$ percentuale eventi positivi, $v(I)$ percentuale negativi).

Strongly collective items Un itemset è è detto strongly collective al livello s se:

1. $C(I) \geq s$
2. $\forall J \subset I : C(J) \geq s$ Chiusura

La proprietà di chisusura è necessaria per assicurare che item non correlati siano presenti nell'itemset. Questa definizione permette di definire un algoritmo simile ad apriori per calcolare la collective strenght.

4.5.8 Negative pattern mining

Vengono scoperti pattern tra item o sull'assenza degli item. Il negative pattern mining necessita di misure bit-simmetriche in modo da trattare presenza e assenza di un item allo stesso modo. Le misure di supporto confidenza non sono progettate per gestire l'assenza di item. Possono essere usate misure quali chi quadro, coefficente di correlazione e collective strenght, ma hanno difficile utilizzo in pratica a causa dei problemi computazionali (assenza della chiusura verso il basso).

4.6 Meta algoritmi

Un meta algortimo è un algortimo che utilizza un altro particolare algortimo come un sub-routine per rendere l'algortimo principale più efficente o trarre nuove informazioni. Due tipologie:

- Utilizzo di campionamento per migliorare l'efficenza del pattern mining
- Utilizzo di pre e post processing come subroutine per applicare l'algoritmo in nuovi scenari

4.6.1 Metodi di campionamento

Database molto grandi non possono essere contenuti nella RAM, possono quindi essere utilizzati solo algoritmi che operano su livelli (level-wise), algoritmi con strategie di ricerca in prodondità su alberi di enumerazioni trovano grande difficoltà in questi scenari a causa della richiesta di accessi random alla memoria. Il campionamento permette di applicare questi algoritmi in maniera efficenza pagando un piccolo costo in accuratezza. Si presentano due problemi:

- Falsi positivi: Patter che superano la soglia di supporto sul campione ma non sul database. Queti possono essere rimossi tra mite una singola scansione dell'intero database.

- Falsi negativi: Patter che non superano la soglia di support sul campione ma la superano sul databse. Questi possono essere gestiti abbassando la soglia di supporto, garantendo un livello garantito di perdita di accuratezza a soglie specifiche. Ridurre troppo la soglia porterà alla creazione di molti itemset inutili rendendo più pesante il post processing.

4.6.2 Ensemble di dati partizionati

Questo approccio permette di garantier l'assenza di falsi positivi e negativi e può essere usato sia per ridurre i costi di accesso al disco sia per ridurre i requisiti di memoria per i database proiettati. Il database delle transazioni viene diviso in k segmenti disgiunti che possono essere inseriti nella memoria principale. Viene applicato l'algoritmo di pattern mining su ogni segmento. L'unione degli itemset frequenti generati dai diversi segmenti genera un superset dei pattern frequenti. Una fase di postprocessing con support counting può essere applicata per la rimozione dei falsi positivi. Una proprietà importante è che ogni pattern frequente deve apparire in almeno uno dei segmenti, altrimenti il supporto cumulativo tra diversi segmenti non raggiungerà la soglia minima.

4.6.3 Approximate frequent pattern sets

1. Sampling: Vengono generati n cambioni randomici dal dataset originale. Viene applicato *random multi sampling*, vengono generati n campioni con ripetizioni con una percentuale p per determinare la grandezza del campione, questi campioni essendo stati creati con rimpiazzamento non sono ne mutualmente esclusivi ne esaustivi. La probabilità che una transazione non sia inserita nel sample di grandezza m dato un dataset di N transazioni è:
$$\left(1 - \frac{1}{N}\right)^{nm}$$

2. Mining: Viene eseguito Apripri per ogni campione su una griglia. La grid computing si definisce come la condivisione di risorse in maniera coordinata e problem solving in un ambiente distribuito.

3. Merging: Unione dei risultati per ricostruire il frequent pattern set approssimato. Le coppie di itemset frequenti vengono unite iterativamente, indicizzando ogni itemset frequente e raccogliendo il supporto medio relativo e il numero di occorrenze, questi poi vengono filtrati sulla base del supporto minimo. I pattern frequenti cosi ottenuti sono una approssimaizone dei pattern set originale che può essere ottenuto dall'intero dataset.

Test su diversi dataset mostrano che i parametri di valutazione incrementano per qualsiasi valore di nmin all'aumentare della percentuale elementi nel campione, questo aumento porta anche ad una diminuizione del numer odi falsi positivi. Aumentare il valore di nmin porta ad una perdita di recall e numero di pattern trovati con un aumento della precisione (diminuisce l'effetto dei falsi

positivi). Un risultato importante è che con un sampling del 0.1 su 10 differenti campioni si ha una precisione che sfiora il 100%.

4.6.4 Generalizzazione ad altre tipologie di dato

Nel quantitative association rule mining i valori quantitativi possono essere discretizzati e convertiti in valori binari, rappresentando l'intero database con una matrice binaria. Regole di associazione di range adiacenti possono essere unite per creare regole riassuntive. Nel caso di variabili categoriche queste possono essere binarizzate, se inoltre è disponibile conoscenza del dominio è possibile utilizzazare cluster dei valori categorici come attributi binari.

Capitolo 5

Clustering

5.1 Introduzione

Partizionamento dei dati in gruppi intuitivamente simili, questo partizionamento di grandi quantità di dati in gruppi più piccoli aiuta nel sintetizzare e capire i dati in molte applicazioni di data mining.

Clustering Dato un insieme di data points, partizionare i dati in gruppi contentenenti data point simili tra loro.

Uno dei primi problemi incontrati in questa categoria di modelli è la quantità delle feature che possono essere rumorose e non informative. Queste vanno rimosse nelle prime fasi di analisi tra mite feature selection.

5.2 Feature selection per il clustering

La feature selection è più difficile per problemi non supervisionati a causa della mancanza di una validazione esterna (come le label per i problemi supervisionati). La feature selection è legata al problema di determinare la tendenza al clustering implicita di un insieme di feature, questi metodi cercano quindi di determinare il subset delle fearture che massimiza la tendenza al clustering. Le tipologie sono:

- Filter: Utilizzando una misura di similarità viene associato un punteggio ad ogni feature. In alcuni casi il modello quantifica la qualità delle feature come combinazione di feature, permettendo di tenere in considerazione l'impatto incrementale di aggiungere una feature ad un altra.
- Wrapper: Viene utilizzato un algoritmo di clustering per valutare la qualità di un sottinsieme di feature, questa informazione viene poi utilizzata per raffinare il set di feature su cui verrà applicato l'algoritmo di clustering principale. Le feature selezionare saranno dipendenti dall'algortimo di clustering scelto per la valutazione, la feature selection verrà quindi ottimizzata per per la specifica tecnica di clustering scelta. Lo svantaggio è che l'informatività intrinseca delle feature specifiche potrebbe perdersi a causa dell'impatto della tecnica di clustering specifica utilizzata.

5.3 Algoritmi representative-based

Gli algortimi representative-based utilizzano la nozione di distanza o similarità per il clustering dei dati attraverso l'uso di un insieme di rappresentati del partizionamento, creati come funzione dei dati nel cluster o scelti tra i dati stessi presenti nel cluster. L'intuizione è che la scoperta di un buon partizionamento dei dati è equivalente alla scoperta di un buon set di dati rappresentatitvi, una volta definiti questi dati viene utilizzata una funzione di distanza per assegnare

ogni data point al rappresentante più vicino. Questa categoria di algoritmi effettua il clustering in una sola passata e non è presente una struttura gerarchica tra i cluster.

Tipicamente si assume che il numero di cluster da creare sia definito dall'utente a priori, e sarà definito k. Sia D un insieme contenente n data points chiamati $\overline{X}_1, \ldots, \overline{X}_n, \overline{X_i} = (x_1, \ldots, x_d)$. L'obiettivo è determinare l'insieme $\overline{Y}_1, \ldots, \overline{Y}_k$ dati rappesentativi che minimizzano:

$$O = \sum_{i=1}^{n} \left[min_j Dist(\overline{X}_i, \overline{Y}_j) \right]$$

Viene quindi minimizzata la distanza dei diversi data points dal rappresentatne più vicni. In alcune variazioni come l'algoritmo k-medoid i rapprsentati sono ottenuti direttamente dal database dei dati (sono elementi del databse). Il problema di minimizzare O si risolve attraverso l'utilizzo di un metodo iterativo dove i rappresentati candidati e rappresentanti assegnati al momento vengono utilizzati per migliorarsi a vicenda (dipendenza circolare dei rappresentati). L'approccio generico inizia quindi assegnando k rappresentatnti casuali S attraverso l'utilizzo di euristiche e vengono poi raffinati tramite:

- Assegnamento: Assegna ogni data point al rappresentatnte più vicino in S utilizzando la fuznione di distanza e siano $C_1, \ldots, C_k$ i cluster cosi creati.
- Ottimizzazione: Determina il rappresentante ottimale per ogni cluster C_j che minimizza la funzione obiettivo locale:

$$\sum_{\overline{X}_i \in C_j} \left[Dist(\overline{X}_i, \overline{Y}_j) \right]$$

Il secondo step è semplificato dal fatto che non dipende più da una assegnazione sconosciuta del set di rappresentanti. Tipicamente il rappresentante ottimale è una misura centrale dei punti nel cluster. Ad esempio nel caso della distanza euclidea o della similarità del coseno si dimostra che il rappresentante ideale è la media. Diverse tipologie di distanza definiscono diversi rappresentanti ideali per ogni cluster. L'idea principale della famiglia di algoritim k-representative p di migliorare la funzione obiettivo in maniera iterativa, fino alla convergenza (il miglioramento della funzione dopo una iterazione è minore di una soglia definita dall'utente). Uno dei principali problemi di questi metodi è il dover applicare la funzione di distanza tra tutte le coppie di punti e rappresentanti portando ad una complessità di $O(k, n, d)$ per un modello di k rappresentanti, n dati e dimensionalità d. Gli algoritmi k-representative tendono a convergere molto velocemente ad un buon clustering dei dati, può capitare però una convergenza a soluzioni subottimali quando un outlier viene assegnato come rappresentante iniziale dell'algoritmo, causando la creazione di un cluster che contiene un solo elemento o il merge di due cluster.

5.3.1 k-Means

La funzione di distanza è la somma dei quadrati della distanza euclidea dei data point dal rappresentante più vicino, questa può essere interpretata come l'errore quadratico dell'approssimare un data point con il rappresentante.

$$Dist(\overline{X}_i, \overline{Y}_j) = \left\|\overline{X}_i - \overline{Y}_j\right\|_2^2$$

Il rappresentante ottimale per ogni cluster è la media dei data point nel cluster (si dimostra ponendo a 0 la derivata parziale della funzione obiettivo rispetto a $\overline{Y}_j$).

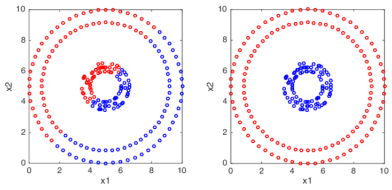
Variante con Mahalanobis Si utilizza la distanza di Mahalanobis per l'assegnazione dei data points ai cluster.Ogni cluster C_j ha la propria matrice di covarianza $\sum_j \in \mathbb{R}^{d\times d}$ che viene calcolata utilizzando i data point assegnati ad ogni cluster nell'iterazione precedente.

$$Dist(\overline{X}_i, \overline{Y}_j) = (\overline{X}_i - \overline{Y}_j)\Sigma_j(\overline{X}_i - \overline{Y}_j)^T$$

L'utilizzo di questa distanza è particolarmente utile quanod i cluster sono ellitticamente allunganti lungo alcune direzioni. La matrice di covarianza applica una normalizzazione della densità locale che è di aiuto per dataset con densità locale variabile.

Svantaggi In generale l'agoritmo k-means non funziona bene con cluster di forma arbitraria (ad esempio forme non convesse), in quanto ha un bias verso cluster sferici. Anche la versione con la distanza di Mahalanobis non funzione bene in questo scenario nonostante l'abilità di adattarsi all'allungamento dei cluster

Kernel k-Means L'algoritmo k-means può essere esteso in modo da trovare cluster di forma arbitraria utilizzando dei metodi chiamati kernel tricks. L'idea alla base è di trasformare implicitamente i dati in modo che cluster di forma arbitraria siano mappati a cluster euclidei in un nuovo spazio. Il problema principale è che la complessità computazionale dell'elaborare la matrice kernel è in funzione quadratica del numero dei dati.



5.3.2 k-Medians

Viene utilizzata la distanza di Manhattan come distanza per la fuznione obiettivo, si dimostra che il rappresentante ottimale per ogni cluster è la mediana dei datapoints lungo ogni dimensione nel cluster. Questo in quanto, dato un insieme di punti distributiti su una lina, il punto che minimizza la somma delle distanze L_1 è proprio la mediana dei punti. Una perturbazione e in una delle direzioni dalla mediana non può ridurre la somma delle distanza L_1, implicando che la mediana ottimizza la somma delle distanze L_1 dei data points in un insieme. Dato che la mediana è scelta in maniera indipendente su ogni dimensione il rappresentante tipicamente non appartiene al dataset D. k-Medians seleziona cluster in maniera più robusta di k-Means in quanto la distanza di Manhattan non è sensibile agli outlier quando la media.

5.3.3 k-Medoids

La funzione obiettivo ha la stessa forma dell'algoritmo k-representative, ma i rappresentanti sono sempre scelti dal database D. Le motivazionni della scelta sono:

- Il rappresentante di un cluster creato tramite k-means può essere distorto dalla presenza di outlier nel cluster (il rappresentante potrebbe trovarsi in una zona non popolata, quindi non rappresentativa dei datapoint nel cluster) portando all'unione di diversi cluster. Questo problema può essere parzialmente risolto usando delle variazioni robuste agli outlier come k-Medians
- Per tipi di dato complessi è difficile computare il rappresentante centrale ottimale.

Una proprietà importante di questo algoritmo è che può essere applicato su ogni tipo di dato se è possibile applicare su esso la misura di distanza scelta. Viene utilizzato un approccio hill-climbing:

1. Il set di rappresentanti iniziali S è inizializzato ad un insieme di punti del database D.
2. Il set S viene ottimizzato scambiando un singolo punto in S con un nuovo punto in D.

S implicitamente definisce una soluzione al problema di clustering e ogni scambio può essere visto come uno step nell'hill-climbing. Lo scambio può essere effettuato tramite:

- Vengono provate tutte le $|S| \cdot |D|$ possibilità per il rimpiazzamento e viene selezionato il migliore. Soluzione computazionalmente costosa.
- Vengono randomicamente scelte r coppie di punti candidate per lo scambio, la migliore tra le coppie viene poi scelta e utilizzata. Il tempo richiesto è proporzionale a r volte la grandezza del database ma è facilmente implementabile per databse di medie dimensioni.

Svantaggi Più lento di k-Means ma maggiore applicabilità per tipi di dati diversi.

5.3.3.1 Partitioning around medoids (PAM)

PAM utilizza una strategia di ricerca greedy che potrebbe non trovare la soluzione ottimale ma che risulta più veloce della ricerca esaustiva:

1. Inizializza: Vengono selezionati k dei n data points come medoids per minimizzare il costo
2. Associa: Ogni data point viene associato al medoid più vicino
3. Ottimizzazione: Finchè la funzine di costo non dimiuisce, per ogni coppia di punti m, o (con m medoid e o punto non medoid) viene calcolata la variazione nella funzione di costo se viene effettuato lo scambio usando quella coppia. Viene selezionata la coppia con la migliore variazione della funzione di costo.

La complessità dell'algoritmo è $O(k(n-k))$.

5.4 Algoritmi per clustering gerarchico

Gli algoritmi gerachici tipicamente effettuano il clustering tramite distanze e molti di questi utilizzano altri metodi di clustering basati su densità o su grafi come subroutine per costruire la gerarchia. Questa categoria di algoritmi è importante in quanto i diversi livelli di granularità dei cluster possono portare alla scoperta di nuova conoscenza da un punto di vista specifico dell'applicazione. Viene quindi crata una tassonomia dei cluster che può essere navigata alla ricerca di conoscenza semantica dei cluster. I metodi gerarchici tipicamente non necessitano di definire a priori il numero di cluster, ma possono essere usati anche per la creazione di cluster "flat" (numero di cluster specificato). Esistono due strategie di clustering gerarchico:

- Bottom up (agglomerativo): I data point individuali sono agglomerati in cluster di livelloi più alto. I vari algoritmi di questa categoria differiscono sulla scelta della funzione obiettivo per il merging dei cluster.
- Top down (divisivo): I data point sono partizionati in una struttura ad albero, ad ogni livello è possibile utilizzare clustering flat per il partizionamento. Questi metodi forniscono flessibilità nella scelta del trade off tra bilanciamento della struttura ad albero e numero di data points in ogni nodo.

5.4.1 Metodi agglomerativi (bottom up)

L'algoritmo inizia con tutti i data point che rappresentano dei cluster singleton. In ogni iterazione i due cluster più vicini tra loro vengono uniti in un nuovo

cluster, è quindi necessario definire una funzione in grado di misurare il grado di prossimità tra cluster.

1. Sia n il numero di data points nel database d dimensionale D
2. Sia $n_t = n - t$ il numer odi cluster dopo t agglomerazioni.
3. L'algoritmo mantiene per ogni step una matrice di distanza $n_t \times n_t$ tra tutti i cluster presenti.
4. In ogni iterazione viene selezionano l'elemento della matrice (non diagonale) con la ditanza minore e i relativi cluster vengono uniti.
5. La matrice viene aggiornata in una nuova matrice di dimensione $n_t - 1 \times n_t - 1$

Il merge di due cluster all'eliminazione delle righe e colonne relative ai due cluster uniti, e all'inserimento di una nuova riga e colonna per il nuovo cluster creato. L'update incrementale della matrice è più efficente di computare tutte le distanze da 0, questo se la matrice può risiedere in memoria, altrimenti la matrice verrà computata nuovamente da 0 ad ogni step. L'algoritmo termina una volta:

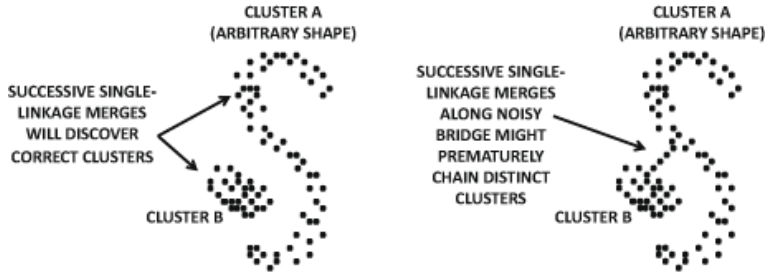
- Soglia massima sulle distanze tra cluster. In questo caso viene determinato automaticamente il numero naturale di cluster, ma richiede la scelta di una soglia ottimale.
- Soglia minima sul numero di cluster creati. Ha il vantaggio di essere intuitivo e facilmente interpretabile.

Questo metodo crea una struttura ad albero gerachica che rappresenza le relazioni tra cluster chiamata *dendogram*

5.4.1.1 Distanza tra cluster: Group-based statistics

Metodi per calcolare la distanza tra due cluster. Siano i, j i due cluster candidati per il merge, siano m_i, m_j rispettivamente il numero di elementi nei due cluster. Questa categoria di metodi esprime la distanza come una funzinoe delle $m_i \cdot m_j$ coppie di distanze tra gli item presenti nei due cluster.

Best (single) linkage Distanza minima tra tutte le coppie di oggetti. Corrisoponde alla coppia di oggetti più vicina tra loro nel cluster. Dopo il merge la nuova riga e colonna inserite nella matrice possono essere comutate usando il minimo dei valori della riga e colonna eliminate. Questo in quanto la distanza del nuovo cluster da tutti gli altri cluster sarà il minimo delle distanze dei cluster individuali. Questo metodo è ottimo per trovare cluster di forma arbitraria, in quanto la forma arbitraria si crea con unioni successive dei datapoint che la compongono. Questo metodo può però portare all'unione di cluster distinti a causa della presenza di punti rumorosi.



(a) Good case with no noise (b) Bad case with noise

L'algoritmo DBSCAN può essere visto come una variante robusta di questo metodo in quanto permette di escludere punti rumorosi tra cluster nel processo di merge.

Worst (complete) linkage Distanza massima tra tutte le coppie di oggetti. Corrisponde alla coppia di oggetti più lontani tra loro nei cluster. Questo criterio cerca implicitamente di minimizzare il diametro massimo di un cluster. Anche chiamato metodo del *complete linkage*. Uno dei principali svantaggi è che cercherà di creare cluster tutti con un diametro simile, quindi se il dataset contiene naturalemente cluster di grandezza variable potrebbe portare a dividere i cluster più grandi. Presenta inolte un bias nel crere cluster di forma sferica oltre a dare troppa importanza all'eventuale frontiera rumorosa di un cluster in quanto prende la distanza massima tra due punti nel cluster (questi punti è molto probabile che siano sulla frontiera)

Group-average linkage Media delle distanze tra tutte le coppie di oggetti. La nuova riga (colonna) nella matrice viene calcolata con una media pesata della i-esima e j-esima riga (colonna) della matrice M (matrice prima dell'eliminazione). Per ogni $k \neq i, j$ il nuovo valore è uguale a:

$$\text{Righe: } \frac{m_i \cdot M_{ik} + m_j \cdot M_{jk}}{m_i + m_j} \quad \text{Colonne: } \frac{m_i \cdot M_{ki} + m_j \cdot M_{kj}}{m_i + m_j}$$

Closest centroid Definito il centroide come la media dei punti del cluster, questo criterio unisce i due cluster con i centroidi più vicini. Uno degli svantaggi è che il centroide perde informazione sulla sparsità degli oggetti nel cluster, questo metodo infatti non trova differenze tra cluster di dimensione diversa finchè i loro centroidi sono vicini tra loro. Questo metodo ha un bias verso il merge di cluster grandi in quanto i centroidi di cluster di dimensione maggiore hanno statisticamente più probabilità di essere vicini tra loro.

Variance-based criterion Minimizza la variazione della funzione obiettivo dopo a causa del merging dei cluster. Il merging causa sempre un peggioramento della funzione obiettivo a causa della perdita di granularità. Per minimizzare

la variazione della funzione obiettivo vengono mantenute per ogni cluster le statistiche del momento di ordine zero, primo e secondo:

- Momento del 0-ordine: Numero m_i di punti nel cluster
- Momento del 1-ordine: F_{ri} La somma dei datapoint nel cluster i lungo ogni dimensione r.
- Momento del 2-ordine: S_{ir} La somma quadratica dei datapoint nel cluster i lungo ogni dimensione r.

$$SE_i = \sum_{r=1}^{d} \left(\frac{S_{ir}}{m_i} - \frac{F_{ir}^2}{m_i^2} \right)$$

L'algoritmo BIRCH utilizza la seguente misura di distanza. Queste statistiche sono facilmente gestibili dopo un merge in quanto le statistiche del nuovo cluster saranno la somma delle statistiche dei due cluster i, j usati per il merge. La variazione che avviene eseguendo un merge dei due cluster i, j è:

$$\Delta SE_{i \cup j} = SE_{i \cup j} - SE_i - SE_j$$

Si dimostra che questa quantità è sempre positiva. Viene mantenuta una matrice delle variazioni per ogni coppia di cluster. Dopo il merge, la k-esima riga (colonna) tale che $k \neq i, j$ della nuova colonna in M è uguale a:

$$SE_{i \cup j \cup k} - SE_{i \cup j} - SE_k$$

Metodo Ward Si utilizza come criterio di merge la somma dell'errore quadratico non scalata utilizzando i momenti definiti nel paragrafo precedente

$$\sum_{r=i}^{d} (m_i S_{ir} - F_{ir}^2)$$

Questo approccio è una variante dell'utilizzo dei centroidi. La funzione obiettivo si ottiene moltiplicando l'errore quadratico della distanza euclidea tra centroidi con la media armonica del numero di punti in ogni coppia di cluster. Questo fattore penalizza cluster più grandi rendendo questo metodo più efficente del metodo dei centroidi.

5.4.1.2 Considerazioni computazionali

I metodi ward, variance e group average sono più robusti al rumore grazie all'utilizzo del multiple linkage nel calcolo della distanza. I metodi agglomerativi richiedono di mantentere un insieme di distanze ordinate in modo da calcolare in maniera efficente la distanza minima nella matrice. La computazione iniziale della matrice richiede $O(n^2 \cdot d)$, mentre il mantenimento di una sorted heap richiede $O(n^2 \cdot log(n))$ nel corso del tempo in quanto verranno effettuate $O(n^2)$

inserimenti e cancellazioni nella struttura. La complessità temporale totale è quindi $O(n^2 \cdot d + n^2 \cdot log(n))$ con complessità in spazio $O(n^2)$. Quest'ultima è problematica per dataset molto grandi in quanto la computazinoe per il merge dovrà essere fatta nuovamente al momento del merge, portando la complessità a $O(n^3 \cdot d)$. Questo processo può essere reso più efficente tramite l'uso di approssimazioni del criterio di merge. L'algoritmo CURE implementa il single-linkage dei metodi gerarchici in modo efficente effettuando una scelta accurata dei rappresentanti che vengono poi usati per computare approsimativamente il criterio single-linkage.

5.4.2 Metodi divisivi (top down)

Possono essere visti come meta-algoritmi general purpose che possono utilizzare qualsiasi algoritmo di clustering come subroutine. Permettono un maggior controllo della struttura globale dell'albero in termini del suo grado (numero di nodi figli per nodo interno) e bilanciamento tra i diversi rami.

1. Viene scelto un algoritmo flat A come subrouting
2. Il nodo dell'albero viene inizializzato con tutti i dati del dataset.
3. In ogni iterazione il dataset di punti in un nodo viene diviso in più nodi figlio.

Se l'algoritmo scelto è randomico come k-Means è possibile effettuare tentativi multipli con l'algoritmo e selezionare la suddivisione migliore.

5.4.2.1 Bisecting k-Means

Ogni nodo viene diviso in esattamente due figli utilizzando l'algoritmo 2-means. Per eseguire lo split vengono effettuate diverse trial utilizzando 2-means e viene scelta la suddivisione che ha l'impatto migliore della funzione obiettivo generale del clustering. La scelta di un diverso criterio di scelta del nodo da dividere porta ad un bilanciamento del peso di cluster o dell'altezza dell'albero.

5.5 Algoritmi basati su modello probabilistico

Gli algoritmi discussi fin ora sono algoritmi di tipo "hard" in cui ad ogni data point viene assegnato ad un solo cluster in maniera deterministica. Gli algoritmi probabilistici sono algoritmi "soft" in quando definiscono una distribuzione di probabilità che un data point appartenga a tutti (o alcuni) dei cluster. Un algoritmo soft può essere trasformato in hard assegnando ad ogni data point il cluster con la probabilità più alta per quel specifico data point.

5.5.1 Mixture based generative model

Si assume che i dati siano stati generati da una miscela (mixture) di k distribuzioni con distribuzioni di probabilità $G_1, \ldots, G_k$ dove ogni distribuzione rappresenta un cluster ed è definito *mixture component*. Ogni data point $\overline{X}_i, i \in [1, n]$ viene generato dal mixture model tramite:

1. Sia $\alpha_i = P(G_i), i \in [1, k]$ la probabilità a priori assegnata ad ogni cluster (probabilità di selezionare uno specifico mixture componente). Si assume sia stato scelto il cluster r.
2. Genera data point da G_r

Si definisce M il modello generativo. Le probabilità a priori e i parametri delle distribuzioni sono iperparametri non conosciuti. Si assume che ogni distrubuzione dei mixture componente sia gaussiana. La scelta della distribuzione riflette la conoscenza dell'utente a priori della distribuzione naturale e della forma naturale dei cluster.

5.5.2 Algoritmo expectation-maximization (EM)

Utilizzato per stimare i parametri delle mixture component in modo da massimizzare la verosimiglianza che i dati siano generati dal modello. Una volta stimati i parametri, le probabilità generative a posteriori dei dati rispetto ai cluster possono essere calcolate.

Sia $f^i(\cdot)$ la fuznione di densità di probabilità della mixture compnete G_i. Allora la probabilità che il data point $\overline{X}_j$ sia generato dal modello è data salla somma pesata delle densità di probabilità dei diversi cluster con le probabilità a priori dei cluster come peso.

$$f^{point}(\overline{X}_j|M) = \sum_{i=1}^{k} \alpha_i \cdot f^i(\overline{X}_j)$$

Perr un insieme di dati D la probabilità che il modello generi tutti i dati è:

$$f^{data}(D|M) = \prod_{j=1}^{n} f^{point}(\overline{X}_j|M)$$

La verosimiglianza logaritmica è invece (espressa anche in forma di somme per fini computazionali):

$$L(D|M) = log(\prod_{j=1}^{n} f^{point}(\overline{X}_j|M)) = \sum_{j=1}^{n} \left(\sum_{i=1}^{k} \alpha_i \cdot f^i(\overline{X}_j) \right)$$

Se le probabilità che i dati siano stati generati da ogni mixture componente sono conosciute risulta facile stimare i paremetri per ogni cluster. Esiste una

dipendenza circolare tra le probabilità e i parametri da stimare. Nel caso degli algoritmi di partizionamento la conoscenza di un hard clustering fornisce la possibilità di determinare dei rappresentanti ottimali per ogni cluster, la conoscenza invece di un soft clustering permette di stimare i parametri del modello localmente per ogni cluster. Questo permette di definire un processo iterativo in cui i parametri del modello e le assegnazione probabilistiche ai cluster sono utilizzati per stimarsi a vicenda.

1. Sia Θ il vettore che rappresenta l'intero insieme dei parametri che descrivono il mixture model
2. Si inizializza Θ a dei valori iniziali (assegnazione random dei data point ai mixture component)
3. Passo Expectation: Si stima la probabilità a posteriori $P(G_i|\overline{X}_j, \Theta)$ cioè la probabilità soft che il data point sia generato da un mixture component. Questo step viene eseguito per ogni dato e ogni mixture component.
4. Passo Maximization: Date le probabilità degli assegnamenti dei data point ai cluster, si utilizza la massima verosimiglianza per determinare i valori dei parametri in Θ
5. L'algoritmo termina se la funzione obiettivo non migliora in maniera significativa

Expectation Tramite la densità di probabilità si calcola la probabilità bayesiana che il data point $\overline{X}_j$ sia generato dalla mixutre component G_i. Il subscript Θ nella densità di probabilità indica che queste sono valutate per i parametri correnti in Θ

$$P(G_i|\overline{X}_j, \Theta) = \frac{P(G_i) \cdot P(\overline{X}_j|G_i, \Theta)}{\sum_{r=1}^{k} P(G_r) \cdot P(\overline{X}_j|G_r, \Theta)} = \frac{\alpha_i \cdot f^{i,\Theta}(\overline{X_j})}{\sum_{r=1}^{k} \alpha_r \cdot f^{r,\Theta}(\overline{X_j})}$$

Maximization Si assume che l'assegnazione nello step Expectation sia corretta. L'ottimizzazione dei parametri avviene calcolata e settata a zero la derivata parzila della log-verosimiglianza calcolata sui parametri del modello attuale. Il valore delle probabilità a priori dei cluster viene calcolata con la frazione pesata dei punti assegnati al cluster:

Probabilità a priori del cluster: Frazione pesata dei punti assegnati al cluster. Nella pratica per ottenere dei risultati più robusti per piccoli dataset si effettua il *laplace smoothing*, si aumenta il numero di data points attesi nel cluster di 1 e il numero dei dati totali di k:

$$\alpha_i = P(G_i) = \frac{1 + \sum_{j=1}^{n} P(G_i|\overline{X_j}, \Theta)}{n + k}$$

5.5.2.1 Gaussian mixture model

Siano d le dimensioni, la distribuzione dell'i-esimo componente è:

$$f^{i,\Theta}(\overline{X_j}) = \frac{1}{\sqrt{|\Sigma_i|}(2\pi)^{d/2}} \cdot e^{-1/2(\overline{X_j}-\overline{\mu_i})\Sigma_i^{-1}(\overline{X_j}-\overline{\mu_i})}$$

Con μ vettore d-dimensionale delle media della i-esima componente gaussiana e Σ matrice di covarianza $d \times d$ della distribuzione gaussiana generale della i-esima componente. $|\Sigma|$ indica il determinante delal matrice. Si dimostra che la stima della massima verosigmiglianza del vettore media e matrice di covarianza produce la media e la matrice di covarianza dei dati in quel mixture component. Questo è lostesso metodo usato da Mahalanobis k-means per calcolare la matrice di covarianza ma in questo caso i data points non sono pesati dato che viene utlizzato un assegnamento hard dei cluster. Il termine nell'esponente della distribuzione gaussiana è il quadrato della distanza di Mahalanobis

Nella pratica per la stima dei parametri le celle non diagonali della matrice di covarianza sono settate a 0 in modo da semplificare il calcolo del determinante (diventa il prodotto delle varianza lungo ogni dimensione). Questo equivale ad utilizzare il quadrato della distanza di Minkwoski nell'esponente. Se invece tutte i valori diagonali shanno lo stesso valore allora è equivalente alla distanza euclidea e ogni compontente del modello genera cluster sferici.

5.5.3 Vantaggi e svantaggi

Nei mixture model i cluster sono caratterizzati da un piccolo numero di parametri, sono più generali degli algoritmi di partitioning e i risultati possono soddisfare le assunzioni statistiche del modello generativo. D'altro canto possono convergere a minimi locali (può essere risolto con trial multipli con inizializzazione randomica), è computazionalmete pesante se il numero di cluster è grande o se il dataset è piccolo ed è difficile stimare il numero di parametri

5.5.4 Relazione tra EM e altri metodi rappresentativi

Siano le probabilità a priori dei cluster $\alpha_i = 1/k$ e sia σ il raggio di tutti i componenti in ogni direzione e sia la media del j-esimo cluster $\overline{Y_j}$. Essendo $\sigma, \overline{Y_1}, \ldots, \overline{Y_k}$ gli unici parametri da stimare la distribuzione di probabilità è:

$$f^{j,\Theta}(\overline{X_i}) = \frac{1}{(\sigma\sqrt{2\pi}^d} \cdot e^{-\left(\frac{\|\overline{X_i}-\overline{Y_j}\|}{2\sigma^2}\right)}$$

Ogni cluster sarà sferico e l'esponente della distribuzione è il quadrato della distanza euclidea scalata. Allora rispetto all'algoritmo k-mean:

- E-Step: Nell'algoritmo k-means viene scelta la migliore distanza euclidea dal rappresentante $\overline{Y_j}$ per assegnare un punto i al cluster j, in EM c'è una

distribuzione di probabilità su tutti i cluster propozionale alla distanza euclidea esponenziata e scalata.

- M-Step: In EM $\overline{Y_j}$ è la media pesta di tutti in datapoint dove il peso è la proabilità dell'assegnamento al cluster j, in k-means ogni elemento ha probabilità 1 per un cluster e 0 per le altre.

Casi speciali dell'algoritmo EM sono equivalenti a versioni soft dell'algoritmo k-Means dove le distanze esponenziate dal rappresentante di k sono usate per definire degli assegnaenti soft delle probabilità in EM. Molti distribuzioni di mixture componente possono essere definite come:

$$K_1 \cdot e^{-K_2 \cdot Dist(\overline{X_i}, \overline{Y_j})}$$

Con K1 e K2 regolati dalla distribuzioni dei parametri. La versione logaritmica di questa distribuzioni esponenziata si mappa direttamente al termine additivo $Dist(\overline{X_i}, \overline{Y_j})$ nello step M della funzione obiettivo che è strutturalmente identica alla corrispondente ottimizzazione additiva nel metodo k-representativa. Dato quindi un mixture model con questa distribuzione è possibile definire un equivalente modello k-representative usando come distanza la distanza dal rappresentante media $\overline{Y_j}$

5.5.5 Condiderazioni pratiche

Principale problema pratico è modellare il livello di flessibilià nel definire i mixture compontente. Se questi sono gaussiani generalizzati saranno più efficaci nel trovare cluster di forma e orientamento arbitrario ma allo stesso tempo richiedono l'apprendiento di un elevato numero di parametri come la matrice di covarianza, in questo caso se il numero di dati è piccolo si può cadere nell'overfitting. Se viene scelto invece una gaussiana sferica il modello EM funzionerà bene su piccoli dataset in quanto verrà appreso un solo parametro ma in caso di orientamento e forma arbitraria le performance saranno basse anche su grandi dataset.

5.6 Algoritmi basati su griglia e densità

Il principale problema degli algoritmo basati su distanza e probabilistici è che la forma dei cluster viene definita implicitamente dalla funzione di distanza utilizzata o dalla distribuzione di probabilità.

Negli algoritmi basati su densità vengono prima identificate delle zone di densità a grana fine (fine grained) che vengono poi usare come blocchi per costruire cluster di forma arbitraria. Questi algoriti possono essere considerati algoritmi gerarchici a due livelli (livello dei blocchi e dei dati). Una volta trovare le regioni di densità queste possono essere organizzate in forme complesse tramite una anlaisi simile al single linkage degli algoritmi agglomerativi

5.6.1 Metodi basati su griglia

I dati vengono discretizzati in p intervalli tipicamente di uguale larghezza.

- Per un dataset d-dimensionale vengono creati p^d ipercubi sui dati sottostanti, questi ipercupi sono i blocchi con cui verranno costruiti i cluster.
- Una soglia τ viene usata per definire un sottoinsieme degli ipercubi che sono densi
- Due regioni si dicono *adiacentemente connessi* se condividono un lato in comune, una forma più depoble di connessione definisce due regioni connesse se condividono uno spigolo (corner). I cluster di forma arbitraria saranno formati da ipercubi adiacentemente connessi.
  - In uno spazio k-dimensionale due cubi si dicono adiacenti se hanno una superfice di dimensionalità almeno r in comune, con il parametro r definito dall'utente tale che $r < k$
- Due ipercubi (regioni) si dicono connesse per densità (density connected) se è possibile definire un percorso tra le due regioni utilizzando soltanto regioni adiacentemente connesse

Clustering su griglia tramite grafi L'obiettivo del clustering grid-based è di scoprire regioni della griglia connesse nel modo descritto, questo risulta facile utilizzando modelli basati su grafi. Ogni cella densa deglla griglia è associata con un nodo nel grafo, un arco tra due nodi indica che le due celle sono adiacentemente connesse. Una volta costruito il grafo è possibili utilizzare algoritmi di ricerca in ampiezza o profondità per trovare tutte le regioni connesse.

Iperparametri e dettagli Questi metodi non necessitano di definire a priori il numero di cluster da generare, inoltre hanno solo due parametri da ottimizzare, i range per la divisione dei dati e la soglia di densità:

- Se il range della griglia è troppo piccolo i data point di più cluster saranno presenti nella stessa regione, al contrario se il valore è troppo grande si formeranno molte regioni non dense (vuote) anche all'interno dei cluster, portando i cluster naturali ad essere disconnessi.
- Se la soglia di densità è troppo bassa tutti i cluster più il rumore verranno uniti in un unico grande cluster, se invece è troppo grande si potrebbero perdere alcuni cluster completamente o parzialmente.

Problemi pratici La scelta del grid-range è particolarmente problematica in quanto non è chiaro come questa è collegata alla soglia di densità. Un altro problema con gli algoritmi su densità è che utilizzano una singola soglia di densità globalmente, non tenendo conto di cluster che potrebbero avere densità variabili, in questi casi algoritmi basati su distanza sono più efficaci, questa

problematica non è specifica degli algoritmi su griglia ma è generale per gli algoritmi di densità. Anche l'utilizzo di regioni rettangolari è una approssimazione usata da questa classe di metodi, questa approssimazione degrada all'aumentare della dimensionalità, in regioni rettangolari ad alta dimensionalità sono una approssimazione povera dei clustersottostanti. Inoltre nel caso di alta dimensionalità questi algoritmi non sono utilizzabili in quanto il numero di celle cresce esponenzialmente con la dimensione.

5.6.2 DBSCAN

In questo metodo la densità dei data point è usata per unirli in cluster, quindi i dati individuali nelle regioni dense sono usati come blocchi di costruzione dei cluster dopo averli classificati in base alla densità. Questo risolve il problema di definire un parametro di densità globale. La densità di un dato è dato dal numero di punti che occorrono entro un raggio *Eps* di quel punto. La densità di queste zone sferiche sono utilizzare per classificare i punti in :

- Core point: Se contiene almeno τ datapoints
- Border point: Contiene meno di τ point ma almeno un core point all'interno del raggio
- Noise point: Non è ne core point ne border point

Si procede quindi in questo modo:

1. Si classificano tutti i punti
2. Si crea un grafo di connettività tra i core points dove i nodi sono i core point e un arco tra due nodi esiste solo se questi sono entro una distanza *Eps* l'uno dall'altro
3. Vengono identificate le regioni connesse in questo grafo questi saranno i cluster
4. I border point vengono assegnati al cluster con cui hanno il più alto livello di connettività
5. I gruppi creati vengono riportati come cluster e noise point come dati rumorosi

DBSCAN come single-linkage agglomerativo Il primo passo di classificazione è identico al single linkage dei metodi agglomerativi con criterio di terminazione di distanza *Eps* applicata ai core point, questo metodo può essere visto come una miglioria dei metodi agglomerativi single linkage in cui vengono trattati in maniera specifica i punti di frontiera e rumorosi, questa miglioria permette di mantenere l'abilità di trovare cluster di forma arbitraria riducendo la sensibilità agli outlier

5.7 Validazione dei cluster

Valutazione della qualità dei cluster trovati, difficile in quanto il problema è non supervisionato (non è presente una label ground truth nel dataset) è necessario quindi definire dei criteri interni per la valutazione. Il principale problema della scelta di questi criteri è che questi potrebbero avere dei bias verso uno specifico algoritmo. Se sono presenti delle label o se è possibile conoscere a priori i cluster naturali le misure sono definite esterne.

5.7.1 Criteri di validazione interni

Questi criteri sono definiti direttamente dalla funzione obiettivo che è ottimizzata dal particolare modello di clustering, si pone quindi il problema di non poter confrontare algoritmi che utilizzano metodologie diverse, questi criteri andranno a favorire un algoritmo che tulizza una funzione obiettivo simile nella ottimizzazione. Sono efficaci quando vengono confrontati due algoritmi che utilizzano lo stesso approccio.

Sum of square distances dai centroidi Valori bassi di questa misura indicano un migliore qualità del clustering. Ottimizzata per gli algoritmi basati su distanza. Il valore assoluto non da informaizoni sul riguardo la qualità dei cluster (è da utilizzare come misura di confronto)

Rateo di distanza intracluster a intercluster Più dettagliata della SSQ dai centroidi. Si effettua un campionamento di r datapoint, di questi sia P l'insieme delle coppie di punti che appartengono allo stesso cluster, i rimanenti punti sono l'insieme Q

$$Intra = \sum_{\overline{X_i},\overline{X_j} \in P} \frac{dist(\overline{X_i},\overline{X_j})}{|P|}$$

$$Inter = \sum_{\overline{X_i},\overline{X_j} \in Q} \frac{dist(\overline{X_i},\overline{X_j})}{|Q|}$$

Il valore finale sarà $Intra/Inter$. Piccoli valori indicano migliore qualità dei cluster.

Coefficente di Silhouette Sia $Davg_i^{in}$ la ditanza media di $\overline{X_i}$ dagli altri punti del suo cluster (distanza interna). Sia $Dmin_i^{out}$ il minimo della distanza media del punto $\overline{X_i}$ dai punti presenti in tutti gli altri cluster (il minimio delle distanze esterne). Il coefficente di Silhouette specifico per il dato i è:

$$S_i = \frac{Dmin_i^{out} - Davg_i^{in}}{max\{Dmin_i^{out}, Davg_i^{in}\}}$$

Il coefficente di Silhouette è la media dei coefficent specifici di tutti i dati. Ha valori in (-1,+1). Valori positivi grandi indicano cluster altamente separati, valori negativi indicano un qualche livello di mescolamento dei dati in diversi cluster. Questo perchè la misura esterna sarà minore di quella esterna solo quando il data point analizzato è più vicino ad almeno un altro cluster che al suo. Il valore assoluto di questa misura è un buon indicatore intuitivo della qualità del clustering.

Misure probabilistiche Si utilizza un mixture model per stimare la qualità del clustering. Si assume che il centroide di ogni mixture component sia il centroide di ogni cluster scoperto mentre gli altri parametrti (come la matrice di covarianza) sono stimate utilizzando metodi simili al Maximizatio step dell'algoritmo EM. Viene riportata la log-verosimiglia globale della misura. Questa misura è utile quando è disponibile conoscenza del dominiio riguardo la forma che ogni cluster dovrebbe avere.

5.7.1.1 Vantaggi e svantaggi

Misure basate su distanza come il coefficente di Silhoutette non funzioneranno bene per cluster di forma arbitraria. Nel caso del coefficent di silhouette i coefficenti specifici di alcuni punti potrebbero avere un impatto negativo anche in caso di clustering corretto, portando il coefficente globale a essere non tanto alto quanto un clustering k-means scorretto.

Il problema maggiore nel effettuare confronti con questi misure è che cercano di creare un "prototipo" di modello ottimale, quindi molte volte queste misure ci dicono solo quanto il modello che viene valutato è vicino (matches) con il modello prototipo della misrua. Questa può essere vista come una forma di overfitting, creando incertezza sulla affidabilità della misura.

Queste misure sono sensibili al numero di cluster trovati dall'algoritmo, infatti due algoritmi dello stesso approccio che trovano un numero di cluster differento non possono essere confrontati

5.7.1.2 Tuning dei parametri con misure interne

L'idea è che la variazione della misure di validazione possa mostrare un punto di flesso alla corretta scelta dei parametri. La forma di questo punto di flesso può variare con la natura dei parametri che vengono ottimizzati o della misura di validazione usata. Se il modello di validazione non riglette la forma naturale dei cluster o il modello algoritmico utilizzato per creare i cluster questi punti di flesso potrebbero essere fuorvianti o non osservabili. Nel caso del k-means SSQ cercherà di ridurre il numero di cluster ma con una riduzione più lenta al punto di flesso, intra-inter ratio diminuirà fino al punto di flesso per proseguire con un leggero aumento.

5.7.2 Criteri di validazione esterni

Vengono utilizzate delle classi associate ai dati per valutare i cluster. Il rischio maggiore utilizzando questi metodi è che le label associate ai dati siano basate su proprietà della specifica applicazione e non rispecchino i cluster naturali dei dati, questi criteri sono comunque preferibili ai criteri interni perchè evitano il bias nella valutazione. Il numero di cluster naturali potrebbe non riflettere il numero di classi. Definiamo k_t, k_d rispettivamente il numero di classi e il numero di cluster.

5.7.2.1 Matrici di confusione

Quando $k_d = k_t$ le matrici di confusione permettono di mappare le classi con i cluster generati dall'algoritmo. Ogni riga i corrisponde alla class label i e ogni colonna j corrisponde al cluster j. Ogni cella (i,j) della matrice contiene il numero di punti della classe i che sono stati mappati al cluster j. La somma dei valori di ogni riga sarà uguale al variare dell'algoritmo di clustering essendo il totale dei punti in ogni classe ground truth. Se il clustering è di alta qualità è possibile permutare righe e colonne in modo che i valori sulla diagonale siano più grandi, quando invece il clustering è di bassa qualità i valori saranno più uniformemente distribuiti. Difficile ispezione visiva quanod $k_d \neq k_t$ e non pratico per matrici troppo grandi.

5.7.2.2 Purity

Sia $m_{ij} \quad i \in [1, k_t] \quad j \in [1, k_d]$ il numero di datapoint della classe i, mappati al cluster j (cella i,j della matrice di confusione). E siano:

$$N_i = \sum_{j=1}^{k_d} m_{ij} \quad \forall i = 1 \dots k_t$$

Numero di elementi nella classe i (somma della riga i della matrice di confusione) e sia:

$$M_j = \sum_{i=1}^{k_t} m_{ij} \quad \forall j = 1 \dots k_d$$

Numero di elementi nel cluster j (somma della colonna j della matrice di confusione). Un cluster j di alta qualità dovrebbe contenere punti che sono largamente dominati da una singola classe, si definisce quindi P_j il numero di data point nella classe dominante di un certo cluster:

$$P_j = max_i\{m_{ij}\}$$

Un cluster di alta qualità avrà un valore di $P_j \leq M_j$ molto vicino al valore d iM_j, si definisce quindi purezza:

$$Purity = \frac{\sum_{j=1}^{k_d} P_j}{\sum_{j=1}^{k_d} M_j}$$

Si desiderano vlori alti di purezza. La purezza può essere computata in due modi:

1. Si calcola la purezza di ogni cluster per poi calcolare la purezza aggregata (formula di sopra)

2. Si calcola la purezza di ogni classe rispetto ai cluster trovati dall'algoritmo

I due metodi NON producono gli stessi risultati soprattutto quando il numero di classi è il numero di cluster è molto diverso tra loro, è possibile utilizzare la media dei due valori come singola misura.

5.7.2.3 Gini Index

La purezza tiene conto solo della label dominante nel cluster e ignore la distribuzione dei dati rimanenti. A parità di purezza, un cluster i cui dati hanno due classi dominanti dovrebbe risultare migliore di uno con tante classi dominanti. Il gini index tiene conto la variazione rispetto alle diverse classi, questa misura è collegata alla nozione di entropia e calcola il livello di differenza (confusione) nella distribuzione dei dati in una riga (o colonna) della matrice di confusione. Può essere calcolata sia per righe sia per colonne producendo valori diversi.

$$G_j = 1 - \sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j} \right)^2$$

Questo valore sarà vicino a 0 se i dati in una colonna della matrice di confusione sono skewed (sparsi), se invece sono distribuiti uniformemente il valore sarà uguale a $1 - \frac{1}{k_t}$ (estremo destro del valore). Il gini index globale è media pesata dei diversi index per colonna:

$$G_{average} = \frac{\sum_{j=1} k_d G_j \cdot M_j}{\sum_{j=1} k_d M_j}$$

Sono desiderabili bassi valori del gini index.

5.7.2.4 Entropia

L'entropia misura le stesse caratteristiche intuitive del gini index:

$$E_j = -\sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j} \right) \cdot log \left(\frac{m_{ij}}{M_j} \right)$$

Bassi valori di entropia indicano clustering di alta qualità. L'entropia generale è:

$$E_{average} = \frac{\sum_{j=1} k_d E_j \cdot M_j}{\sum_{j=1} k_d M_j}$$

5.7.2.5 Recall e precision

Vengono generate tutte le coppie di dati per ogni cluster generato dall'algoritmo. La precision è la frazione di coppie di dati che appartengono alla stessa classe. Per la recall vengono calcolate tutte le coppie di dati per ogni classe ground truth e la frazione di queste che appare nello stesso algoritmo rappresenta la recall.

$$Precision: \quad \frac{m_{ij}}{\sum_k m_{ik}}$$

$$Recall: \quad \frac{m_{ij}}{\sum_k m_{kj}}$$

5.7.2.6 Fowlkes-Mallows e F1-Score

L'index di Fowlkes-Mallows (F_β score) permette di unire precisione e richiamo, regolata dal parametro β:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Per $\beta = 1$ equivale alla media geometrica di precisione e richiamo e si chiama $F_1 score$:

$$F_1 = 2 \frac{precision \cdot recall}{precision + recall}$$

Capitolo 6

Appendice

6.1 Programmazione dinamica

In ottimizzazione matematica per programmazione dinamica si intende il processo di semplificazione di una decisione suddividendola in una sequenza di decisioni nel tempo. Sia y lo stato del sistema, vengono definite delle funzioni valore $V_1, \ldots, V_n$ con $V_n(y)$ che rappresenta il valore ottenuto con lo stato y all'ultimo istante di tempo n. I valori di V_i per $i = n-1, n-1, \ldots, 1$ possono essere trovati tramite una ricerca all'indietro, usanto una relazione ricorsiva chiamata *equazione di Bellman*

Il valore $V_i - 1$ ad ogni stato i è calcolato da V_i massimizzando una funzione del guadagno da una decisione effettuata al tempo $i-1$ e del valore V_i del sistema quando la decisione viene effettuata. Il valore V_1 (stato iniziale del sisteam) è il valore della soluzione ottimale.

Un problema per poter essere risolto tramite programmazione dinamica necessita di:

- Sottostruttura ottimale: La soluzione al problema di ottimizzazione può essere ottenuta tramite combinaione di soluzioni a suoi sottoproblemi. Se un problema può essere risolto tramite combinazioni di soluzioni ottimali non sovrappobili la strategia è chiamata *divide and conquer*.
- Sottoproblemi sovrapponibili: Lo spazio dei sottoproblemi deve essere piccolo. L'algoritmo ricorsivo non deve generare nuovi sottoproblemi ma risolvere lo stesso sottoproblema ogni volta.

6.2 Statistica di base

Siano X,Y due variabili aleatorie, si definiscono:

Varianza della variabile aleatoria X

$$\sigma_Y^2 = E[(X - E[X])^2]$$

Covarianza delle variabili aleatorie X e Y

$$\sigma_{XY} = E[(X - E[X])(Y - E[Y])]$$

con la seguente proprietà

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$$

Varianza campione dato $(x_i, y_i)_{i \in [1,\dots,N]} \in (X, Y)$

$$S_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2 \cong \sigma_X^2$$

Covarianza campione delle variabili X e Y

$$S_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y}) \cong \sigma_{XY}$$

6.3 Similarità su testo

Il testo può essere considerato un dato quantitativo multidimensinale quando è rappresentato in forma di *bag of words*. Dato un testo viene estratta ogni parola (possono essere rimosse stop word) e ad ognuna di essere è associata la frequenza nel documento (o nei documenti). Questa è una rappresentazione sparsa (se prendiamo l'intero vocabolario delle parole come insieme dei possibili valori, un testo in media avrà molte meno parole, settando molte di queste a frequenza 0). Tutte le frequenze sono non negative. Misure come la norma L_p non si adattano bene alla lunghezza variabile dei documenti. La misura del coseno permette di normalizzare queste irregolarità (angolo tra due documenti).

Similarità del coseno Siano $\overline{X} = (x_1, \dots, x_d)$ e $\overline{Y} = (y_1, \dots, y_d)$ due documenti definiti sul lexicon di dimensione d. La similarità del coseno tra i due documenti è:

$$cos(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}$$

Questa misura usa le frequenze grezze degli attributi, per altri tipi di dato è possibile usare misure statistiche globali per migliorare la misura.

Inverse document frequency Funzione decrescente sul numero di documenti n_i in cui occore la i-esima parola. Usata con normalizzazione logaritmica:

$$id_i = log(\frac{n}{n_i})$$

con n numero di documenti. Questo è utile quando ad esempio due documenti hanno un match su una parola non comune, questo evento deve essere più indicativo di um match su una parola comune.

Damping function Usata per evitare che la presenza di una sola parola non alteri negativamente la funzine di similarità. Si applica la radice quadrata o il logaritmo delle frequenza prima di calcolare la similarità. La frequenza normalizzata sarà quindi:

$$h(x_i) = f(x_i) \cdot id_i$$

Posso sostituire quindi la dicitura x_i, y_i nella similarità del coseno con $h(x_i), h(y_i)$

Coefficente di Jaccard Usato raramente nel computare similarità del testo ma più comune per dati binari sparsi:

$$J(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{d} h(x_i) \cdot h(y_i)}{\sum_{i=1}^{d} h(x_i)^2 + \sum_{i=1}^{d} h(y_i)^2 - sum_{i=1}^{d} h(x_i) \cdot h(y_i)}$$