Università degli Studi di Napoli
Federico II

Temporal Data Mining:
tecniche e algoritmi di clustering

Gabriella Milone

Tesi di Dottorato

in *Statistica*

XIX Ciclo

Temporal Data Mining:
tecniche e algoritmi di clustering

Napoli, 30 novembre 2007

Ringraziamenti

È difficile pensare di poter ringraziare tutti coloro che, in vario modo, hanno contribuito alla nascita e al completamento del presente lavoro.
Un primissimo ringraziamento, rivolgo al Prof. Natale Carlo Lauro per aver egli mostrato, da sempre, vivo interesse verso la tematica approfondita nel mio lavoro di tesi, nel palese convincimento che tale contesto di ricerca possa aprire nuovi orizzonti scientifici al "nostro", mi piace pensarlo così, Dipartimento.
Un ringraziamento particolare, rivolgo alla Professoressa Germana Scepi, che ha incoraggiato l'idea di questo lavoro, sin dalla sua fase "embrionale", risalente a tre anni fa, proponendomi di trasformarla in quella che poi sarebbe divenuta la mia tesi di dottorato, e che mi ha costantemente assistito, non risparmiando critiche costruttive ed inviti a calibrare taluni momenti analitici, in seno alla realizzazione del lavoro in parola.
Rivolgo un ringraziamento speciale al Prof. Luigi D'Ambra, per le sue circostanziate critiche e per l'approccio "entusiasmante" e l'indefessa dedizione, inculcatemi per la ricerca scientifica, nonché al Prof. Aurelio Pane - mio maestro da sempre, fin dall'epoca della mia tesi di laurea, con il quale collaboro da anni con instancabile dedizione - e al Prof. Antonio Perna che, prima ancora della mia attiva partecipazione al corso di Dottorato, mi ha trasmesso la passione per la Statistica e per le discipline quantitative in genere.

Considero fondamentale il contributo dei maestri sopra citati, avendo, gli stessi, inculcato in me il "metodo" e dispensato suggerimenti e incoraggiamenti utili alla mia ricerca, e, soprattutto, per aver onorato la mia persona del prestigioso "imprinting" accademico-scentifico, tipico della nostra "scuola napoletana".

Desidero anche ringraziare il Dott. Massimo Aria, per i numerosi suggerimenti relativi agli algoritmi di clustering e alle loro performance, che mi hanno consentito di comprendere appieno le problematiche legate agli aspetti computazionali, caratterizzanti gli algoritmi utilizzati nell'ambito del dominio temporale, oggetto del lavoro di tesi.

Un ringraziamento affettuoso, rivolgo all'amico, Dott. Enrico Ciavolino, dell'Università di Lecce, per avermi sorretto, soprattutto nelle fasi più complesse e delicate, con i suoi preziosi consigli.

Ringrazio, inoltre, l'Ing. Antonello Tamburrino, dell'Università di Cassino, per avermi accompagnato nell'apprendimento del linguaggio di programmazione MATLAB e per avermi fornito validi consigli, in merito al metodo di ricerca delle complesse tematiche scientifiche, legate alle problematiche della similarità, nonché alle proprietà sulle metriche e sulle distanze, presenti in letteratura.

Preziosissimo è stato il contributo del Geologo, Dott. Fabio Matano, per avermi fornito la base dati utile all'applicazione, nonché per avermi indirizzato sulla più opportuna chiave di lettura dei risultati ottenuti.

A tutti i dottorandi, passati e presenti, dedico un pensiero particolare per avermi ascoltato e dato fiducia, come pure, ai dottorandi che dopo di me vivranno questa entusiasmante e appassionante esperienza formativa, e auguro di poter usufruire, in egual misura, del contributo a me fornito dai maestri e colleghi anziani.

Ringrazio, infine, ma non per ordine di importanza, la mia famiglia, che, da sempre, appoggia le mie scelte, e, in particolare, colui del quale posso ormai dire di essere "figlia d'arte": mio padre.

Dedico questo lavoro, alla mia Martina, che, con i suoi “tre anni”, ed il suo sorriso, mi ha regalato la grinta e l’entusiasmo di cui avevo bisogno.

“L’uomo può credere all’impossibile,
ma non crederà mai all’improbabile”

(Oscar Wilde)

Indice

Lista delle figure

Lista delle tabelle

Introduzione

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

La realizzazione di un tale obiettivo di miglioramento è stato reso particolarmente arduo dall'esplosiva crescita delle dimensioni delle basi dati commerciali, governative e scientifiche. In ogni caso, i sistemi di gestione delle basi di dati (Data Base Management System, DBMS) hanno certamente permesso di manipolare le informazioni raccolte in maniera efficace, pur non avendo, appieno, risolto il problema di come supportare l'uomo nella "comprensione" e nell'analisi dei dati stessi. L'ambito scientifico a cui ci si riferisce è quello della gestione di grosse mole di dati, la cosiddetta ottica del Data Mining, al quale il presente lavoro di tesi si ispira, soffermandosi, in particolare, sull'estensione al Temporal Data Mining, riferito a strutture di dati temporali, espressi sia come sequenze che come serie storiche. In tale contesto, infatti, saranno confrontati, ambiti, metodi e algoritmi, con particolare attenzione alle tecniche di Clustering.

Uno dei principali fattori di criticità riscontrati nel Temporal Data Mining risiede nella scelta di tecniche efficienti per estrarre conoscenza da un gran

numero di dati temporali che sono, per loro natura, complessi da rappresentare e da trattare. A tale proposito, durante gli ultimi decenni, i grandi progressi nel campo dell'hardware, della tecnologia dei database, della grafica, hanno reso possibile la nascita di sistemi potenzialmente in grado di trattare grandi quantità d'informazioni complesse e multidimensionali, come, ad esempio, dati temporali, dati spaziali, e dati spazio-temporali. Ciò si traduce nello sviluppo di un ampio ventaglio di applicazioni: basti pensare ai Sistemi d'Informazione Geografica (GIS), ai sistemi di modellazione geometrica (CAD), alle applicazioni scientifiche, ai database catastali, e così via.

Il ragionamento temporale e spazio-temporale è, infatti, alla base di molte attività umane. Il problema di trattare con informazioni di questo tipo è diventato un aspetto sempre più rilevante, sul quale si è concentrata gran parte della ricerca scientifica, negli ultimi tempi. Nella realtà, infatti, spazio e tempo sono tra loro strettamente interconnessi: la maggior parte delle informazioni che si riferiscono allo spazio attengono anche al tempo. Ci sono, appunto, applicazioni per le quali è assolutamente indispensabile poter utilizzare dati spazio-temporali. Basti pensare ai sistemi spazio-temporali che forniscono grandi benefici in aree quali quelle del monitoraggio ambientale, dei settori amministrativi, dei sistemi di navigazione *real-time*, degli *scheduling dei trasporti*[1], ecc.

Si prendano, ad esempio, in considerazione le informazioni riguardanti alcuni tipi di dati ambientali: monitoraggio dello spostamento delle tempeste, della disposizione dei ghiacciai sul globo terrestre, dell'estensione delle superfici marine. Esse consistono, essenzialmente, di un insieme di dati spaziali con associate altrettante informazioni generali, da telerilevare sulla scorta di istanti temporali. Nel corso del tempo è possibile, infatti, che tali informazioni subiscano delle modifiche e, dal momento che alcune delle applicazioni possibili necessitano di utilizzare i dati originari, si rende necessaria la

[1]E' una componente fondamentale dei sistemi dei trasporti, in grado di eseguire un processo interrompendone temporaneamente un altro, realizzando così un cambiamento di contesto.

creazione di un *information system* che li contenga, che sia di facile accesso e che ne consenta il recupero ed il semplice utilizzo applicativo.

Anche in tali ambiti, ancora oggi, si fa riferimento a sistemi di raccolta del dato di tipo tradizionale, utilizzando database d'uso comune, che in realtà poco si prestano alle dinamiche dei dati cui si è appena fatto cenno. Strutture di database tradizionali non sembrano essere appropriate per memorizzare, manipolare e gestire dati complessi multidimensionali ad un alto livello d'astrazione, in particolare, a causa della complessità strutturale di tali dati che richiedono anche nuove funzionalità di *querying* e tecniche specifiche d'indirizzamento che molti degli attuali *DBMS* non sono in grado di fornire.

Sia i dati temporali che quelli spaziali differiscono dai dati convenzionali per il fatto che essi spesso modellano oggetti infiniti. I modelli relazionali falliscono nel trattamento di dati multidimensionali con estensioni possibilmente infinite e con operazioni associate molto complesse.

Dal momento che l'esigenza di dover trattare dati spazio-temporali è nata solo di recente, i modelli esistenti non sono ancora del tutto soddisfacenti a causa del fatto che non riescono, appunto, a fornire meccanismi di ragionamento espliciti e flessibili di cui l'informazione spazio-temporale necessita.

Uno dei limiti, al riguardo, sta nel fatto che non è presente un meccanismo ad alto livello che renda semplice l'interrogazione del database. Per raggiungere un alto grado di *performance*, infatti, l'utente deve essere un esperto del sistema e deve essere in grado di usare primitive efficienti e di basso livello.

In un tale contesto, dal punto di vista strettamente operativo, il processo globale di analisi ed elaborazione dell'informazione, allo scopo di estrarne della conoscenza di supporto alle decisioni, fa comunque riferimento al ben noto, agli esperti di Data Mining, processo di Knowledge Discovery in Databases (*KDD*). Si tratta di un processo, tipicamente interattivo ed iterativo, di scoperta ed interpretazione della *conoscenza* a partire dalle informazioni memorizzate in una base di dati.

Il processo di *KDD* è caratterizzato da una serie di fasi, di cui si discuterà in dettaglio nel primo capitolo, che riguardano essenzialmente *selezione* e *pre-*

elaborazione per, poi, confluire nel vero e proprio *processo di data mining*, la cui estensione ai dati temporali, rappresenta oggetto del presente lavoro di tesi.

In un contesto tipicamente temporale, infatti, per *Data Mining* (DM) si intende l'estrazione di dati temporali e, nella pratica diffusa, viene assimilato quale sinonimo di *Knowledge Discovery in Temporal Database* (KDTD, *scoperta della conoscenza contenuta in database temporali*). In realtà, il termine KDTD fa riferimento a tutto il processo di scoperta della conoscenza, mentre il Temporal Data Mining consiste nell'applicazione, ad alto livello, di particolari metodi per l'estrazione dei dati temporali.

Infatti, il processo di KDTD consiste in una sequenza di passi:

- *temporal data cleaning*, in cui si tenta di ridurre i dati errati, incompleti o rumorosi. Nella raccolta dati, infatti, possono presentarsi valori inconsistenti, causa la violazione di alcuni vincoli di integrità;
- *temporal data integration*, in cui si integrano le differenti sorgenti di dati in uno solo. I dati devono essere integrati risolvendo le possibili inconsistenze ed eliminando le ridondanze;
- *temporal data selection*, in cui si estraggono dai dati sorgente solo dati rilevanti per le analisi;
- *temporal data trasformation*, in cui si trasformano o consolidano i dati in forme più appropriate per il mining;
- *temporal data mining*, in cui si identificano e si caratterizzano relazioni tra insiemi di dati, senza richiedere necessariamente che l'utente ponga delle domande precise. Questa fase è quella in cui si opera con algoritmi specifici per estrarre modelli significativi dai dati;
- *temporal pattern evaluation*, in cui si identificano i *pattern* che rappresentano la conoscenza;
- *temporal knowledge rapresentation*, in cui si visualizzano le tecniche di rappresentazione della conoscenza.

Il Temporal Data Mining (TDM) racchiude in se sia tecniche descrittive che predittive. Le prime si occupano di descrivere i dati temporali in modo conciso e sommario, presentandone proprietà generali, le seconde hanno il compito di rappresentare inferenze su insiemi di dati temporali noti, provando a predirne il comportamento in nuovi insiemi.

Tra le tecniche particolarmente rilevanti nel TDM vi è l'*analisi esplorativa*, con il compito di individuare tutte le tecniche di analisi dei dati e di costruzione di modelli interpretativi che richiedano una forte supervisione da parte dell'operatore. Sebbene tutti i sistemi di *data mining* siano guidati, soprattutto nelle fasi iniziali e finali del processo, quelli basati sull'analisi esplorativa capovolgono l'approccio al problema, in quanto si basano sulla capacità dell'essere umano di riconoscere regolarità e *pattern* all'interno di un insieme di dati.

L'analisi esplorativa, come tecnica di TDM, risulta particolarmente interessante per i dati spazio-temporali e, più in generale, per tutti i dati di natura sperimentale/scientifica.

In merito ad una tale tipologia di dati, il vasto e repentino sviluppo del settore del Data Mining ha condotto all'esplorazione dei domini spazio-temporali e temporali, in particolare, implementando tecniche sempre più sofisticate, ma che potessero essere di ausilio agli utilizzatori.

Nella pratica corrente, infatti, si fa sempre più uso di domini temporali e/o spaziali, per cui le componenti tempo e/o spazio devono essere tenute in giusto conto nel processo di *mining*, allo scopo di interpretare correttamente, e sempre più efficientemente, i dati collezionati.

È quindi possibile estendere molte delle tecniche già esistenti di Data Mining a questa tipologia di domini, facendo particolare attenzione, così come si vedrà nel corpo della tesi, alle peculiarità delle tematiche del *Clustering temporale*.

Organizzazione della tesi

La mia tesi di dottorato, oltre a presentare una rassegna dello stato dell'arte in merito alle tecniche di *data mining* su *dati temporali*, con particolare riferimento alle problematiche di *Clustering*, vuole essere un valido supporto agli utilizzatori di dati, riferiti a domini temporali, presentando una dettagliata rassegna critica degli algoritmi che la letteratura fornisce, tanto da fungere quasi come una sorta di "protocollo applicativo", in riferimento al ventaglio scientifico-disciplinare coinvolto nei processi di *mining temporale*.

Il lavoro risulta essere suddiviso, a parte la breve introduzione al contesto scientifico trattato, in cinque capitoli:

- Il primo capitolo ha lo scopo di introdurre e presentare, nel dettaglio, l'evoluzione delle tecniche, dei formalismi e delle funzionalità associate a problematiche di grosse moli di dati appartenenti a domini temporali e spazio temporali, mediante un processo di *Knowledge Discovery in Temporal Databases* (quale estensione del *Knowledge Discovery in Databases*), senza tralasciarne, peraltro, la rappresentazione della sua architettura. In particolare, si affrontano le problematiche di trasformazione e campionamento di dati temporali e spazio-temporali; la rassegna dettagliata sulle tecniche in ambito di *data mining* nei domini trattati, con particolare riferimento al delicato aspetto della struttura di questo tipo di *dati temporali* (sotto forma di sequenze e di serie storiche), *spaziali* e *spazio-temporali*.

- Il secondo capitolo ha lo scopo, invece, di definire, approfondire e analizzare le problematiche di *Temporal Data Mining*, fornendo un'accurata rassegna sullo stato dell'arte dei processi ad esso collegati. In particolare, oltre a fornire le definizioni e le funzionalità del TDM, in merito ad obiettivi e ad ambiti disciplinari coinvolti, si effettua una prima disamina delle problematiche di questi contesti, legate alla similarità, alla periodicità, aspetti a cui sono peculiarmente legati una specifica tipologia di dati: le serie storiche.
- Il terzo capitolo ha lo scopo di porre l'attenzione su di un particolare ambito disciplinare del TDM, la "classificazione temporale" soffermandosi, nel dettaglio, sull'approccio del *clustering*, in ambito tipicamente temporale. In particolare, oltre a soffermarsi sulla disamina degli ambiti applicativi, si chiariscono i concetti di similarità e di ottimizzazione delle funzioni relative a set di dati temporali e spazio-temporali, fornendo una critica alle metriche (distanze) utilizzate per il clustering.
- Il quarto capitolo ha lo scopo di fornire una trattazione organica della tassonomia sugli algoritmi di clustering, interessati al contesto di riferimento. In particolare, ci si sofferma su come gli algoritmi di clustering in contesti di analisi di grosse moli di dati, assumano una connotazione di "tecniche di pre-processing", al fine di semplificare i database temporali di grosse dimensioni. Per ciascuno degli algoritmi presentati, viene fornita una critica in termini di vantaggi e svantaggi applicativi, legati alla cosiddetta "complessità computazionale", e ci si sofferma sulle performance di particolari famiglie di algoritmi: quelle divisive (es. *K-means* e sue estensioni, SOM - *Self Organization Map* - e sue estensioni) e quelle basate sulla densità (DBSCAN, OPTICS, DENCLUE).

- Il quinto capitolo ha lo scopo di fornire un'applicazione di algoritmi di clustering su un grosso data set forniti dal "Progetto TELLUS"[1]. I dati sono espressi sotto forma di serie storiche raccolte in base a rilevazioni di immagini *radar satellitari*.

  Si tratta di dati sottoposti ad una particolare tecnica di analisi (coperta da brevetto internazionale): la PS-InSAR^{TM} (T.R.E. s.r.l.), sviluppata e brevettata dal Politecnico di Milano e concessa in licenza esclusiva a TRE, primo *spin-off* commerciale del Politecnico, nel 2000. La *PS* rappresenta uno strumento molto efficace per il monitoraggio ad alta precisione di fenomeni di deformazione della superficie terrestre (in particolare, dati dei satelliti ERS-1/2 dell'ESA - *European Space Agency*). I sistemi radar satellitari coerenti e nello specifico i radar di tipo SAR (*Synthetic Aperture Radar*) utilizzati nell'elaborazione sono in grado di misurare la distanza tra il sensore e il bersaglio, registrando il tempo di volo tra l'onda trasmessa e la porzione retrodiffusa. Grazie alla loro periodicità di acquisizione (circa mensile) i dati SAR forniscono misure ripetute della distanza sensore-bersaglio, consentendo, mediante confronti successivi, di misurarne gli spostamenti nel tempo.

  La *Tecnica PS* si pone come obiettivo quello di sfruttare tutte le acquisizioni disponibili su una stessa area e individuare quei bersagli (*Permanent Scatterers - PS*) che mantengono inalterate nel tempo le proprie caratteristiche elettromagnetiche. Per ciascuno di essi è possibile stimare e rimuovere il disturbo atmosferico e, quindi, ricostruirne la storia dei movimenti, nell'intervallo di tempo analizzato con precisione millimetrica.

  L'applicazione degli algoritmi di cluster ha consentito di ottenere una mappatura del territorio di Benevento, in base a diverse forme di

[1]Unità di Supporto Locale n°6 Campania - Progetto Operativo Difesa Suolo (PODiS) – PON ATAS QCS 2000-2006, Direzione Generale Difesa Suolo - Ministero dell'Ambiente e della Tutela del Territorio e del Mare.

spostamento del terreno, utile per scopi di prevenzione di frane e terremoti.

La tesi è corredata, infine, di un capitolo di conclusioni, che possano condurre il lettore ad intravedere spunti di riflessione per eventuali sviluppi di carattere scientifico, nonché di due appendici in cui si riporta, rispettivamente il codice sorgente dell'algoritmo utilizzato nella applicazione (Appendice A) e una serie di note, utili per la comprensione dell'ambito scientifico di riferimento del dataset, utilizzato nella stessa, per il quale si sono adoperate tecniche di clustering nel Temporal Data Mining (Appendice B).

Capitolo 1

Tecniche, formalismi e funzionalità di Data Mining in contesti spazio-temporali

1.1 Processo di Knowledge Discovery in Databases

Senza dubbio, lo sviluppo dei metodi statistici ha prodotto un certo numero di tecniche di analisi dei dati utili nel caso in cui si debbano confermare delle ipotesi predefinite. Tali tecniche risultano però inadeguate nel processo di scoperta di nuove correlazioni e dipendenze tra i dati, che crescono in quantità, dimensione e complessità.

Orientativamente, si possono individuare tre fattori che hanno cambiato il panorama dell'analisi dei dati:

- la disponibilità di grande potenza di calcolo a basso costo;
- l'introduzione di dispositivi di raccolta automatica dei dati (si pensi, ad esempio, alla rilevazione di dati ambientali) insieme alla disponibilità di vaste memorie di massa a basso costo;
- l'introduzione, negli ultimi dieci anni, di un nuovo insieme di metodi sviluppati dalla comunità dei ricercatori in Intelligenza Artificiale.

Tali metodi permettono l'analisi e l'esplorazione dei dati, consentendo, tra l'altro, una più efficace rappresentazione della conoscenza.

Il processo globale di analisi ed elaborazione dell'informazione, allo scopo di estrarne della conoscenza di supporto alle decisioni, è noto come Knowledge Discovery.

Si vuole introdurre, in questo primo capitolo, la definizione di Data Mining e, più in generale, di Knowledge Discovery in Databases (*KDD*), facendo particolare riferimento ad uno specifico dominio di applicazione: lo spazio e il tempo.

Il processo di *KDD* è caratterizzato dalle seguenti fasi:

- *selezione*. Partendo dai dati contenuti nel database, detti dati grezzi, si estrae l'insieme dei dati che si ritengono maggiormente significativi per il tipo di analisi che si vuole effettuare;
- *pre-elaborazione*. In questa fase viene effettuata un'integrazione dei dati. In generale, i dati provengono da diverse fonti presentando quindi delle incongruenze quali, ad esempio, l'uso di diverse denominazioni per individuare uno stesso valore che può assumere un attributo. Inoltre, questa fase prevede la pulizia dei dati, in cui si eliminano eventuali errori, ed il trattamento dei dati mancanti;
- *processo di Data Mining*. Questo processo ha come scopo quello di fornire all'utente finale una rappresentazione della conoscenza che ha acquisito, applicando uno o più metodi di Data Mining, partendo dai dati risultanti dalla fase di pre-elaborazione. In particolare, il processo di Data Mining (*DM*) è concettualmente suddiviso in tre punti:
  - *trasformazione o preparazione;*
  - *data mining vero e proprio;*
  - *interpretazione, visualizzazione e previsione*.

Il Data Mining si presenta come il punto di confluenza dei risultati della ricerca di settori come l'*Intelligenza Artificiale* ed il *Machine Learning*; la *Statistica Inferenziale*; le *Basi di Dati* ed il *Data Warehousing*.

La difficoltà nell'esporre una visione unificata ed omogenea dei processi e delle tecniche di DM è dovuta alla vastità del settore. Nonostante la complessità del processo in questione, si è tutti d'accordo sull'individuare quali elementi tipici, che caratterizzano il settore della *KDD*: la grande dimensione degli archivi dei dati adoperati e l'automatizzazione spinta del processo di acquisizione della conoscenza.

Questo settore è, attualmente, in fase di grande sviluppo, e la sua importanza aumenterà sempre più rapidamente nel corso dei prossimi anni.

Si parlerà, nel seguito, di applicazione di Data Mining per intendere l'attuazione del processo, relativamente ad un caso specifico, utilizzando programmi *ad hoc* che implementino algoritmi specifici di *data mining*, o sfruttando *tool* generali, in maniera opportuna. Di seguito viene fornita una schematizzazione dell'intero processo di *KDD* (Figura 1.1).

Come già accennato, il processo di Data Mining si può suddividere in tre fasi principali: la trasformazione o preparazione; il data mining vero e proprio; l'interpretazione, visualizzazione e previsione. La fase di preparazione dei dati dipende dal metodo di mining che verrà utilizzato nell'applicazione.

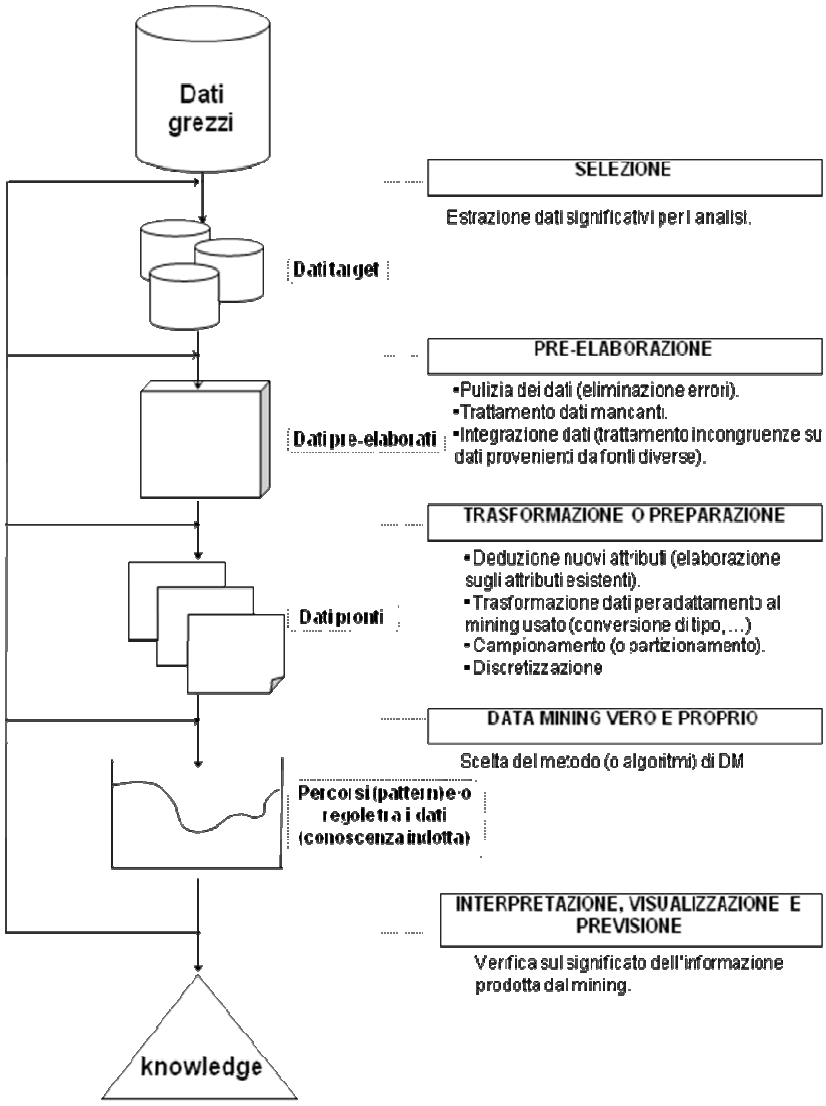


Figura 1.1: Processo di Knowledge Discovery in Database

Le attività tipiche di questa fase sono:

- l'*introduzione di nuovi attributi*, mediante l'applicazione di operatori logici e matematici, aumentando così la quantità di informazioni utili disponibili;
- la *trasformazione dei dati* per adattarli al metodo di Data Mining che verrà applicato;

- il *campionamento* o *partizionamento dei dati*;
- la *discretizzazione dei dati*.

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

Mediante l'applicazione di uno o più metodi o algoritmi vengono determinati i percorsi, le eventuali regole e le caratteristiche dei dati. Se il risultato che si ottiene in questa fase non è soddisfacente, si può operare una nuova trasformazione dei dati, tornando allo stadio precedente e applicando nuovamente il processo di Data Mining, utilizzando lo stesso o un diverso metodo/algoritmo.

Nello specifico, si tratta di operare il cosiddetto *apprendimento* o *esplorazione* e *modellazione*, in base alle denominazioni correntemente in uso del campo dell'Intelligenza Artificiale e della Statistica, rispettivamente.

Il risultato o l'obiettivo della fase di *mining* è costituito, appunto, dalla "conoscenza" indotta, rappresentando i dati secondo uno specifico formalismo. Tuttavia, prima di utilizzare ai fini pratici tali informazioni, è necessario che queste ultime siano opportunamente validate, ossia si deve verificare se il *mining* ha prodotto risultati significativi.

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

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

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

1.2 Trasformazione dei dati

La trasformazione, come già specificato, prevede la deduzione di nuovi attributi, il campionamento e l'adattamento dei dati al mining usato.
La discretizzazione sugli attributi numerici può essere utilizzata per consentirne la trattazione con algoritmi di data mining che trattano esclusivamente informazioni di tipo simbolico, oppure, più semplicemente, per diminuire la cardinalità di un attributo.
La fase di campionamento dei dati può essere, in generale, fondamentale per un esito positivo nell'uso di algoritmi di data mining: se il set di dati sui quali si applicano gli algoritmi è troppo grande, il costo dell'operazione di data mining è troppo elevato; se, invece, il set è troppo piccolo, può essere poco rappresentativo dell'insieme globale dei dati, pertanto, l'operazione di data mining può comportare l'apprendimento di nuova conoscenza di scarso utilizzo pratico.

1.2.1 Il campionamento

La fase di *campionamento* ha lo scopo di estrarre, dalla totalità dei dati a disposizione, un sottoinsieme (campione) sul quale verrà eseguito l'algoritmo di data mining.
Dal punto di vista economico si deve costatare che, sebbene il data mining voglia valorizzare la conoscenza latente contenuta nei dati, in certe situazioni, un approccio poco oculato potrebbe comportare costi tali da renderlo addirittura improduttivo.

Si deve, inoltre, osservare che la quantificazione delle risorse necessarie deve considerare anche delle attività a latere del data mining; si pensi al *data cleaning*, il cui costo è proporzionale alla massa di dati trattati.
In ultimo, si devono considerare i limiti delle attuali tecnologie che sono tali da rendere l'utilizzo del data mining, al disopra di certi livelli, del tutto impensabile.
Inoltre, è opportuno ridurre il numero di esempi da analizzare affinché l'algoritmo di apprendimento non sia *sovra-addestrato* o confuso dal rumore dei dati al punto di non essere in grado di produrre informazione facilmente interpretabile. Si consideri che in pratica tutti gli algoritmi di apprendimento sono sensibili a questi aspetti, sebbene alcuni lo siano in misura minore, ma in genere questi ultimi sono quelli con il minore potere espressivo.
Infine, si può pensare che un'opera di selezione dei dati sia utile al fine di evitare la scoperta di un numero eccessivo di regole di bassa qualità. In particolare, per evitare che venga prodotta della conoscenza già nota o per indirizzare la sensibilità del modello in una certa direzione (ad esempio a riconoscere prevalentemente esempi di una data classe).
Le tecniche di campionamento maggiormente usate sono:

- *Campionamento casuale*. Ogni elemento dell'insieme ha eguale probabilità di essere estratto per costituire l'insieme di *training*. Tale tipo di campionamento garantisce delle buone prestazioni sia dal punto di vista delle risorse computazionali necessarie, sia dal punto di vista della qualità del campione estratto;
- *Campionamento mediante clustering*. Si procede con un pre-raggruppamento dei dati, in base ad un opportuno criterio, ricadendo, così, nel problema più generale del *clustering*. Quindi, si estraggono, sempre casualmente, da ogni cluster un numero di campioni, proporzionale alla dimensione del cluster, sull'insieme di partenza. All'interno di ogni cluster, ogni campione ha la stessa probabilità di essere estratto. Se il clustering viene effettuato in modo opportuno, ovvero scegliendo in modo oculato l'algoritmo e la metrica, vi è la

possibilità di ridurre sostanzialmente il rumore dei dati, in quanto dati anomali e poco significativi possono venire esclusi dall'algoritmo di clustering e, pertanto, avranno una probabilità nulla di essere estratti. Rispetto al campionamento casuale, vi è, ovviamente, un maggiore costo computazionale;

- *Campionamento stratificato.* In presenza di attributi non numerici, gli algoritmi di clustering possono risultare inadeguati, pertanto, per migliorare la qualità del campionamento si può ricorrere alla stratificazione. Si selezionano una serie di attributi nominali di particolare interesse e si raggruppano i dati rispetto ad essi. Ogni gruppo viene denominato *strato* e si procede in modo analogo a quanto fatto per il campionamento mediante *clustering*, ovvero, ogni elemento all'interno di uno strato ha la medesima probabilità di essere estratto, mentre ogni strato contribuisce con un numero di campioni proporzionali al suo peso. Se uno strato ha un peso troppo basso, può essere scartato, in quanto assimilabile a rumore nei dati. Alla fine si ottiene un campione in cui le frequenze degli attributi utilizzati per la stratificazione sono simili a quelle dell'insieme originario. Si osservi che gli attributi utilizzati per la stratificazione devono assumere pochi valori distinti, e che ogni strato deve contenere un numero minimo di valori.

1.3 Tecniche e Formalismi del Data Mining

Un sistema di data mining è formato da un insieme di programmi *ad hoc* o da un *tool* generico di data mining, che permette di "eseguire" un'applicazione di data mining.

In letteratura, non esiste una visione unificata dei sistemi di data mining. Nonostante ciò, appare comune a tutte le classificazioni una suddivisione dei sistemi in due aspetti tipici:

- le funzioni offerte dal sistema all'utente (dette *funzionalità*);
- i metodi impiegati per realizzare le funzionalità.

Quando si parla di funzionalità, si intende tutto ciò che un sistema di data mining mette a disposizione dell'utente, affinché esso possa ottenere nuova conoscenza dai dati, in altre parole, "cosa" il sistema offre all'utente.

Il termine *metodi*, invece, si riferisce a "come" la funzionalità viene offerta. In generale, è possibile distinguere, all'interno del concetto di "metodi", tra formalismi e tecniche, nel senso che, a volte, un metodo di data mining (o tecnica) non determina direttamente la funzionalità, ma produce un formalismo che viene utilizzato dall'utente per realizzare la funzionalità.

Questa distinzione apparirà più chiara quando verranno introdotti, più dettagliatamente, funzionalità, tecniche e formalismi.

Le principali funzionalità di un sistema di data mining:

- *la Scoperta di Regole*. Questa funzionalità permette di individuare delle correlazioni tra i *record* di una base di dati, in base ad associazioni o a sequenze;
- *la Classificazione e Regressione*. Si tratta di un processo che porta a classificare un record (di una base di dati), a partire dal valore assunto dai suoi attributi. In sostanza, a partire da un insieme di record già classificati, si determinano certe regolarità che permettono di "predire" il valore di un nuovo attributo del record, la "classe". Si parla di classificazione quando la variabile da predire può assumere valori in un insieme discreto. Si parla, invece, di regressione quando deve essere predetto un valore continuo;
- *il Clustering* (o raggruppamento). Si tratta di un processo che porta a risultati simili a quelli della classificazione. A differenza della classificazione, però, il clustering è in grado di produrre una suddivisione dei record in gruppi (*cluster*) in maniera del tutto autonoma. Nella classificazione, invece, il raggruppamento era imposto attraverso l'insieme di addestramento. In altre parole, si tratta di una classificazione non addestrata. Il raggruppamento è fatto in

modo tale che i record in ogni cluster siano "simili" secondo certi criteri o metriche. Il clustering non fornisce un modello, cioè non permette di predire in quale cluster "cade" un nuovo record. Per "classificare" il nuovo record si rende necessario un nuovo raggruppamento. Per quanto visto, il clustering è detto "unsupervised learning".

Complessivamente, si possono individuare delle tecniche di utilizzo generico ed altre indirizzate ad un particolare formalismo e/o funzione.

Tra le tecniche di utilizzo generico vi sono:

- l'Analisi esplorativa (Esplorative Data Analysis);
- il Problem solving (ricerca euristica, discesa del gradiente, ecc.);
- la Computazione evolutiva.

Tra i formalismi possiamo individuare:

- le Reti neurali;
- le Reti bayesiane;
- gli Alberi di decisione;
- il Clustering;
- le Regole di associazione;
- i Pattern sequenziali;
- i Formalismi logici (logica proposizionale, dei predicati del primo ordine, VL-1, ecc.).

Accanto alle tecniche generiche, esistono delle tecniche specifiche che permettono di generare particolari formalismi. Queste tecniche sono, ad esempio, gli algoritmi o i metodi *ad hoc* per il clustering, per le reti neurali, per le reti bayesiane, per le regole di associazione, per i pattern sequenziali e per gli alberi di decisione, nonché i metodi basati sull'*inferenza logica*.

Una prima classificazione delle funzionalità in base alla tecnica/formalismo che la genera è rappresentata come segue.

Tabella 1.1: Tecniche, formalismi e funzionalità di Data Mining

Tecnica	**Formalismo**	**Funzionalità**
Metodi *ad hoc* per reti Neurali	Reti Neurali	Classificazione/ regressione Clustering
Algoritmi di Clustering	Clusters	Clustering
Metodi *ad hoc* per regole di associazione	Regole di Associazione	Associazioni
Metodi *ad hoc* per pattern sequenziali	Pattern sequenziali	Sequenze
Metodi *ad hoc* per alberi di decisione	Alberi di decisione	Classificazione
Metodi *ad hoc* basati su inferenza logica	Formalismi logici	*
Problem Solving	*	*
Computazione evolutiva	*	*

1.4 Data Mining nel dominio spazio-temporale

Il vasto sviluppo che il settore del Data Mining ha visto negli ultimi anni, ha condotto all'esplorazione di nuovi domini di applicazione, nonché alla specializzazione dei metodi ed algoritmi generici, a particolari domini di dati. Poiché molti di questi domini hanno un contesto inerentemente temporale e/o spaziale, le componenti tempo e/o spazio devono essere tenute in giusto conto nel processo di mining, allo scopo di interpretare correttamente, e più efficientemente, i dati collezionati. È quindi possibile estendere le tecniche già esistenti di Data Mining o svilupparne delle nuove, per adattare in modo adeguato le componenti spazio e tempo.

Si è pensato, inizialmente, di trattare le tecniche di Data Mining spaziale e temporale in modo separato, per via delle differenze in termini di dominio dei dati su cui si vuole operare, in modo da risolvere il problema dell'analisi nel

modo più completo possibile, cercando di applicare al dominio suddetto ciò che la ricerca specifica offre.

1.5 Struttura dei dati nel Temporal Data Mining

I concetti presentati in questo paragrafo fanno riferimento al Data Mining Temporale (Temporal Data Mining, TDM) e alla struttura i database temporali, in termini di sequenze o di serie storiche.

Il TDM concerne l'analisi di eventi ordinati sulla dimensione tempo.

Naturalmente il concetto di "tempo" può essere inglobato all'interno di un database relazionale, introducendo, ad esempio, un attributo che tenga conto della dimensione temporale.

I database temporali (TDB), invece, incorporano il concetto di tempo per creare un elevato livello di astrazione, utile in molte applicazioni, al fine di mantenere la "storia" degli oggetti trattati (non a caso i TDB sono anche chiamati *historical databases*).

Esempi tipici di database temporali sono: basi di dati finanziarie (prezzi azionari), basi di dati mediche (sequenze genetiche) e basi di dati provenienti dalla sperimentazione fisica, per citarne solo alcuni[1].

Nello specifico, un database temporale può contenere tre differenti tipi di oggetti:

- *time-invariant*: oggetti vincolati a non cambiare il loro valore (ad es. la data di nascita di una persona);
- *time-varying*: oggetti che possono cambiare valore con una frequenza arbitraria (ad es. il salario o il livello di un dipendente);

[1] Negli ultimi anni l'importanza della ricerca nel campo dei database temporali è stata riconosciuta dalla comunità scientifica internazionale tramite eventi quali: *int. Workshop on Temporal Databases Infrastructure* (1993), supportato da ARPA/NSF; VLDB-affiliated temporal Workshop (1995); istituzione di una speciale sezione dell'IEEE *Transaction on Knowledge and Data Engineering*, dedicata ai databases temporali e real-time (1995);inclusione di costrutti temporali nel linguaggio SQL3.

- *time-series* (serie storiche): oggetti che cambiano valore con frequenza stabilita e regolare (ad es. campionamenti regolari di dati scientifici o il valore giornaliero di un'azione in una certa borsa), il modo in cui cambiano i valori viene definito dal calendario.

In particolare, il concetto di time-series permette di manipolare, quindi, dati osservabili in periodi regolari.

Una serie storica è rappresentata attraverso una sequenza di eventi. Un evento è una coppia:

$$(t, v)$$

dove t è il valore temporale e v è il valore del dato osservato al tempo t.

Il valore del dato può essere semplice (valore singolo) o composto (multivariato). Il formato tipico di una serie storica multivariata è:

$$\{(t_1, <v_1,1, v_1,2, \ldots>), (t_2, <v_2,1, v_2,2, \ldots>), \ldots\}$$

Ogni serie storica è associata ad un calendario, che determina quale è la sequenza dei valori temporali (Tabella 1.2):

$$t_1, t_2, t_3, \ldots$$

Tabella 1.2: Esempio di calendario che definisce le "ore lavorative giornaliere"

Esempio di Calendar work hours	**Tipologia**	**Descrizione**
granularity	Ora	unità di tempo di default nel calendario
pattern	{[9,11];[13, 17]}	sottosequenza di unità di tempo espressa come elemento temporale
period	24	lunghezza di un intervallo di tempo nel quale ricorre ripetutamente un pattern
start time	4/1/2000	partenza del calendario
end time	∞	fine del calendario

È chiaro che il TDM può essere applicato a numerosi domini. In particolare, ad esempio, tutti quelli in cui si ha a che fare con dati in cui è importante legare il "fenomeno" all'istante o al periodo temporale in cui si verifica. Si pensi all'archivio delle nascite nelle anagrafi, ai file *.log* degli accessi ad un server da una rete esterna e così via.

Tutte queste collezioni di eventi sono significative fonti di informazione, non soltanto per ricercare particolari valori o eventi in un determinato istante o periodo temporale, ma anche, ad esempio, per l'analisi della frequenza di certi eventi o insiemi di eventi, associati da una relazione temporale (ad esempio: "la prima settimana di tutti i mesi di 31 giorni").

Questi tipi di analisi possono essere molto utili nella scoperta di informazione implicita nei dati grezzi e nella predizione di comportamenti futuri nei processi osservati.

Dall'analisi della letteratura si può distinguere tra due ampie direzioni su cui la ricerca si indirizza: la scoperta di "relazioni causali" tra eventi orientati temporalmente (analisi delle relazioni di causa-effetto); la scoperta di "pattern simili" nella stessa sequenza di tempo o tra diverse sequenze di tempo (analisi delle time-series o analisi dei *trend*).

1.5.1 La scoperta di relazioni causali

La scoperta di relazioni causali tra eventi orientati temporalmente, coincide con la scoperta del fatto che un certo evento, se si verifica, potrebbe implicare (sotto certe condizioni stabilite dalla regola) il verificarsi di un altro evento. Si potrebbe chiamare causa il primo evento ed effetto il secondo.
Nella terminologia che si intende utilizzare in questo contesto, il concetto di scoperta di relazioni causali tra eventi, corrisponde a quello di scoperta di regole di associazione sequenziale.
Un esempio di applicazione su basi di dati mediche può essere la predizione dei rischi per la salute (effetto) dovuti al verificarsi di certe condizioni (causa), riconosciuti dall'analisi dell'evoluzione temporale di certi parametri clinici del paziente.

1.5.2 Le serie storiche

Le "serie storiche" rappresentano una sequenza di osservazioni, di una certa variabile, nel tempo. Trovare un *trend* nei dati significa, sostanzialmente, determinare una regolarità nell'evoluzione temporale degli stessi.
L'analisi del trend si concentra nell'identificazione di pattern simili tra serie temporali, oppure nella ricerca di pattern simili all'interno della stessa serie storica, che si ripete con una certa regolarità.
Questo è un campo di attiva ricerca scientifica ormai da molto tempo.
Gli aspetti della ricerca, in questo settore, includono:

- l'approssimazione di curve con metodi matematici;
- la riduzione del rumore sui dati;
- la comparazione delle sequenze temporali utilizzando tecniche di *similarity matching* e tecniche di *predizione* (tramite metodi matematici o euristici).

Si noti che il matching di sequenze prevede:

- l'operazione di normalizzazione;

- l'analisi della periodicità;
- lo sviluppo di linguaggi di interrogazione flessibili su database temporali.

Nella ricerca di pattern simili, svolge un importante ruolo la misura di similarità utilizzata (metrica).

Le metriche più utilizzate sono la *distanza Euclidea* e la *correlazione*. La correlazione permette, a differenza della distanza Euclidea, di misurare la similarità, senza generare tutte le sottosequenze di lunghezza *n* (se la serie campione ha lunghezza *n*) di tutte le serie storiche nel database.

Da non sottovalutare sarebbe, in un tale contesto, l'importanza delle tecniche di visualizzazione e regressione nell'analisi del trend.

Un classico dominio di applicazione delle tecniche di analisi del trend su serie storiche, è quello dei dati finanziari. Gli analisti finanziari, infatti, cercano di riconoscere pattern sull'andamento dei prezzi nel mercato azionario, in modo da prevedere una discesa o salita del prezzo, sia sul breve che sul medio-lungo periodo, e quindi decidere se acquistare o vendere l'azione in questione.

1.6 Struttura dei dati nello Spatial Data Mining

In questo paragrafo si introduce il concetto di Data Mining Spaziale (Spatial Data Mining, *SDM*) e di database spaziali, facendo attenzione alla loro applicazione in contesti temporali. Ci si riferisce al Data Mining spaziale ed, in particolare, al DM spazio-temporale.

Lo *SDM* si riferisce all'estrazione di informazione da dati "orientati allo spazio". Con questo si intende dire che i dati presentano una forte caratterizzazione spaziale; in altre parole, i dati descrivono l'informazione presente in uno spazio ad *n* dimensioni.

Si pensi, ad esempio, all'informazione presente in una mappa geografica (bi-dimensionale) o al progetto di un edificio (tridimensionale).

La ricerca nel *SDM* nasce dallo sviluppo di particolari database che raccolgono oggetti adatti alla rappresentazione dei dati spaziali. Tale ricerca ha condotto, nel corso del tempo, allo sviluppo di strutture speciali di dati per l'accesso e la gestione degli stessi.

Entrambi i campi ricerca, che, sempre più spesso si integrano a vicenda, hanno un comune obiettivo: la gestione dei dati che consenta un'esecuzione efficiente delle interrogazioni.

Tipiche interrogazioni relative alla disposizione dei dati nello spazio sono: "estrarre tutti i punti che distano, meno di una certa distanza, da un punto particolare della mappa"; oppure: "determinare i punti che si trovano lungo una certa linea" (che, ad esempio, può modellare un fiume).

In particolare, la ricerca nelle strutture di accesso ai dati (chiamate anche SAM, dall'acronimo di Spatial Access Methods) nasce dall'idea di organizzare l'informazione, in modo che la contiguità geografica dei dati sia rappresentata nella struttura dati stessa, consentendo quindi una più efficiente esecuzione delle interrogazioni.

Nei paragrafi successivi si presenteranno diversi tipi di strutture dati ad albero che consentono di gestire efficientemente collezioni di punti (ad esempio, le città in una mappa geografica).

La specificità del problema ha condotto allo sviluppo di particolari sistemi dedicati, adatti alla gestione dei dati spaziali di tipo geografico, detti Sistemi Informativi Geografici (o *GIS* dall'acronimo di *Geographic Information System*). I *GIS* permettono, ovviamente, di descrivere, non solo la disposizione dei dati nello spazio, ma anche le caratteristiche di ciascun punto, linea o regione dello spazio. Ad esempio, i punti che rappresentano le città saranno dotati, oltre alle proprie coordinate geografiche, anche di informazioni non strettamente spaziali, come il numero di abitanti e l'altezza sul livello del mare.

Tramite i *GIS* sarà allora possibile esprimere interrogazioni in cui si mescolano aspetti spaziali e non: ad esempio si potranno estrarre tutte le città italiane con meno di centomila abitanti in cui si trova un castello medioevale.

Un particolare problema che affligge la *SDM* è la mancanza di un accordo sull'insieme comune dei formati dei dati.

Proprio per questo motivo, è in corso uno sforzo, ad opera del Consorzio "Open *GIS*" (O*GIS*), per la standardizzazione del formato per lo scambio dei dati spaziali, in modo da consentire interoperabilità fra i vari prodotti *GIS*.

Per quanto riguarda il Data Mining spazio-temporale, si possono identificare due direzioni di ricerca:

- l'introduzione del tempo sui sistemi spaziali;
- l'accomodamento dello spazio nei sistemi di mining temporali.

Probabilmente, la prima soluzione è quella migliore. In effetti, i sistemi di mining spaziale presentano una buona maturità e specializzazione (si pensi ai *GIS*), cosicché, sembra più corretto tenere adeguatamente conto della dimensione tempo sui database spaziali. Ad esempio, si può pensare di utilizzare il tempo come attributo non spaziale dei punti collezionati nel database spaziale. In questo modo, ciascun punto rappresenta, in realtà, non un oggetto spaziale vero e proprio, ma un "evento".

Forse il mining più pesante su dati spazio-temporali è quello sui dati legati alle condizioni atmosferiche, allo scopo di comprendere e prevedere i fenomeni geofisici.

All'Università della California di Los Angeles (UCLA) si è sviluppato un intero ambiente distribuito e parallelo per l'interrogazione e l'analisi di dati spazio-temporali, chiamato CONQUEST (CONtext-based Querying in Space and Time)[2].

Si rende necessario, a questo punto, approfondire il concetto di *SDM*, introducendo i concetti base ed una rassegna sulle soluzioni basate su un approccio statistico.

[2] A conferma del crescente interesse nel campo del mining spazio-temporale, si sono svolte, recentemente, diverse conferenze: NCGIA Varenius Workshop on Discovering Geographic Knowledge in Data-Rich environments, Redmond WA, USA, Marzo 1999; DEXA Workshop on Spatio-Temporal Data Models and Languages, Firenze, Italia, Agosto 1999; VLDB Workshop on Spatio-Temporal Database Management, Edinburgh, Scotland, Settembre 1999.

1.6.1 Le tecniche di Spatio-Temporal Data Mining

Con il termine "dati spazio-temporali" si indicano, come già accennato, dati relativi ad oggetti che occupano uno spazio, quindi caratterizzati da attributi spaziali di varia natura, osservati in istanti di tempo.

L'accesso ai dati in un database spazio-temporale avviene per mezzo di strutture dati di accesso speciali (i SAM, appunto).

Parlare di Spatial-Temporal Data Mining, *STDM* o Knowledge Discovery su database spazio-temporali significa estrarre della conoscenza implicita, relazioni spazio-temporali o altri pattern, non esplicitamente osservabili nei database spazio-temporali.

I precedenti lavori nell'ambito dei DBMS, del *machine learning* e della statistica, insieme ai progressi nei database spaziali, come le strutture dati spaziali, lo *spatial reasonong* e la geometria computazionale, formano le basi per lo studio del Data Mining spazio-temporali. Cruciale è il problema della efficienza negli algoritmi di Data Mining spazio-temporali, data la grande mole di dati spaziali su cui, generalmente, si opera e la complessità dei tipi di dati e dei metodi di accesso.

I metodi di *STDM* possono essere applicati per "comprendere" i dati spaziali, catturare relazioni implicite tra dati spaziali e non-spaziali, costruire basi di conoscenza spaziali (spatial knowledge bases), ottimizzare *query*, riorganizzare i database spaziali per ottenere maggiori performance, catturare caratteristiche generali in modo semplice e conciso, ecc.

Tutto ciò ha vaste applicazioni nei Sistemi Informativi Geografici, nel telerilevamento, nell'esplorazione di collezioni di immagini, nella grafica biomedica, ecc.

L'analisi statistica è, storicamente, l'approccio più comune per analizzare i dati spaziali e più nello specifico dati spazio-temporali.

Quella dell'analisi statistica è un'area matura, per questo motivo sono ormai stati individuati molti algoritmi e tecniche di analisi. Tali algoritmi

maneggiano bene dati numerici e, normalmente, costruiscono modelli realistici del fenomeno spazio-temporale.

Il maggiore svantaggio di questo tipo di approccio è l'assunzione della indipendenza statistica tra i dati. Questo causa diversi problemi, visto che spesso i dati spaziali sono, in realtà, "interconnessi", ad esempio un oggetto può essere influenzato dagli oggetti ad esso adiacenti.

I tentativi di attenuare tale problema hanno prodotto processi di modellizzazione estremamente complessi, manipolabili solo da utenti esperti.

In altre parole, non si tratta certamente del tipo di tecnica adatta all'utente finale. Oltretutto, l'approccio statistico non produce buoni modelli per regole non lineari, valori simbolici e dati incompleti.

Con l'avvento del Data Mining i ricercatori proposero vari metodi per scoprire informazioni da grandi database relazionali e transazionali.

Lo sforzo è sempre stato quello di combinare i risultati provenienti dalle aree più mature, come quelle del *machine learning*, delle basi di dati e della statistica, al più moderno settore del Data Mining, fino a porre le fondamenta per lo *SDM*. In tale ambito scientifico, esistono alcune tipiche terminologie di cui è bene chiarirne il significato.

Si parte dalla definizione delle regole che possono essere scoperte nei database spaziali:

- *Regole di caratterizzazione spaziale*. Sono delle descrizioni generali dei dati spaziali. Ad esempio, una regola che descrive la gamma dei prezzi delle case nelle diverse regioni geografiche di una città, è una regola di caratterizzazione;
- *Regole di discriminazione spaziale.* Sono delle descrizioni delle caratteristiche discriminanti o contrastanti di una classe di dati spaziali da un'altra. Ad esempio, la comparazione della gamma dei prezzi delle case in diverse regioni geografiche: residenziale e rurale;
- *Regole di associazione spaziale*. Sono delle regole che descrivono le implicazioni di una o un'insieme di caratteristiche, da un altro insieme di caratteristiche, in un database spaziale. Ad esempio, una

regola che associa il *range* dei prezzi delle case alle caratteristiche "vicine", come una spiaggia o un parco, è una regola di associazione.

Si può, a questo punto, spiegare il significato di Mappe Tematiche che hanno il compito di presentare la distribuzione nello spazio di uno o più attributi.

Le mappe tematiche differiscono dalle mappe generiche, in quanto, queste ultime hanno come obiettivo principale, quello di presentare la posizione di un oggetto, rispetto agli altri oggetti spaziali.

Le mappe tematiche possono essere usate per scoprire diversi tipi di regole, ad esempio, si può costruire una mappa tematica della temperatura nell'analisi del pattern spaziale generale, del tempo atmosferico in una regione geografica.

Riguardo la struttura dei dati per l'accesso ai dati spaziali, come già detto, è possibile oggi eseguire più efficientemente le interrogazioni, sfruttando la contiguità fisica dei dati, nella geometria stessa della struttura.

Più formalmente, i metodi di accesso (*spaziali* o *puntuali*) sono formati da una struttura dati ed un insieme di algoritmi associati, per la ricerca di punti o poligoni definiti in uno spazio multidimensionale.

Se nella struttura vengono rappresentati punti, si parla di PAM (Point Access Methods), se invece vengono rappresentati poligoni o comunque oggetti dotati di dimensione non nulla, si parla, più propriamente, di SAM (Spatial Access Methods).

Tra le strutture ad albero introdotte dai ricercatori vediamo quelle che consentono di gestire collezioni di punti (un punto può rappresentare, ad esempio, una città di una regione). Ciascun punto viene rappresentato da un nodo dell'albero. Ogni nodo comprende le sue coordinate (X e/o Y e/o Z), delle informazioni specifiche del nodo ed i puntatori ai nodi successori.

Nell'organizzazione ad albero bidimensionale (*2-d tree*) ogni nodo ha due successori, rispettivamente, destro e sinistro.

In questa organizzazione, il nodo radice rappresenta un'intera zona geografica, ed ogni nodo figlio suddivide la zona geografica rappresentata dal nodo padre in due zone, tramite una linea, che è verticale oppure orizzontale,

a seconda che il punto sia ad una distanza pari o dispari rispetto alla radice (se la distanza è pari, la linea è orizzontale).

Un esempio di una rappresentazione *2-d tree* la si trova nella Figura 1.2, la cui rappresentazione spaziale è indicata in Figura 1.3.

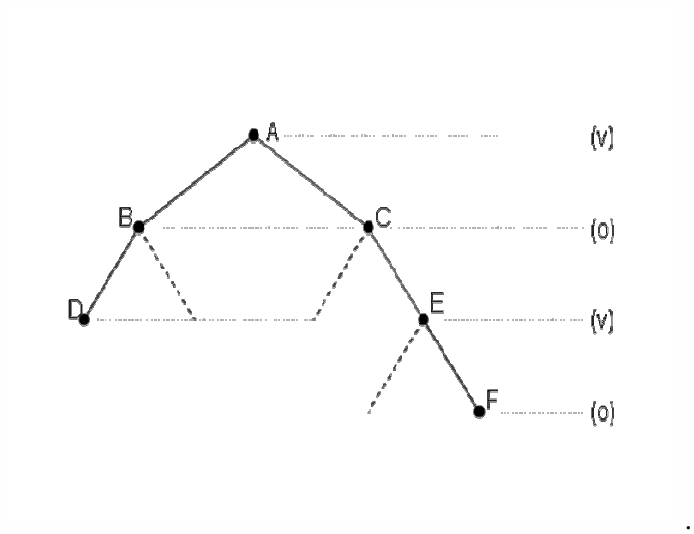


Figura 1.2: Esempio di *2-d tree*

Si noti che, nell'albero, il figlio di sinistra è quello in alto o a sinistra, spazio indicizzato dall'albero, a seconda che la retta suddivida verticalmente o orizzontalmente lo spazio.

Nell'organizzazione a *quad-tree* ogni nodo suddivide la zona spaziale da esso rappresentata, in quattro zone tramite due linee, orizzontale e verticale, che passano per il punto stesso.

Questo significa che ogni suo nodo ha quattro successori, che rappresentano i quattro quadranti (Figura 1.4 e 1.5).

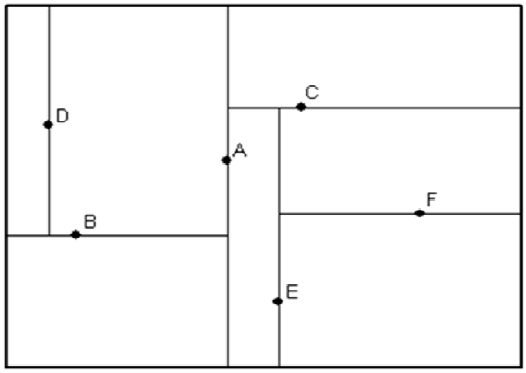


Figura 1.3: Rappresentazione di un *2-d tree*

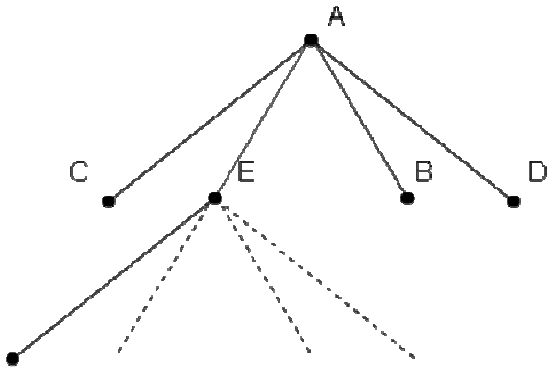


Figura 1.4: Esempio di *quad-tree*

In questo *quad-tree* A è il nodo radice ed ogni figlio suddivide la regione in cui è collocato in quattro parti.

Si noti che nell'albero, il figlio più a sinistra è quello in alto a destra, nello spazio indicizzato dall'albero. Per gli altri figli si procede in senso orario, cosicché, il figlio più a destra nell'albero, corrisponde al punto in alto a sinistra nello spazio.

I PAM più utilizzati sono, oltre ai *2-d tree* ed i *quad-tree* appena visti, i *k-d tree* e i *BSP-tree*.

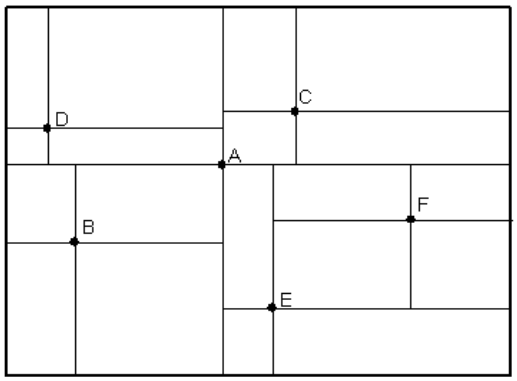


Figura 1.5: Rappresentazione di un *quad-tree*

1.7 Strutture e modelli di SDM

Varie architetture e modelli sono stati proposti per il processo di Data Mining. Autorevoli esempi sono: l'architettura parallela di Holsheimer (1994), oppure l'architettura multicomponente di Matheus (1993). Molte di queste sono state usate o estese per maneggiare il mining spaziale dei dati.

L'architettura di Matheus (1993) sembra essere molto generale ed è stata usata da altri ricercatori nel Data Mining spaziale (Figura 1.6), come Ester (1995).

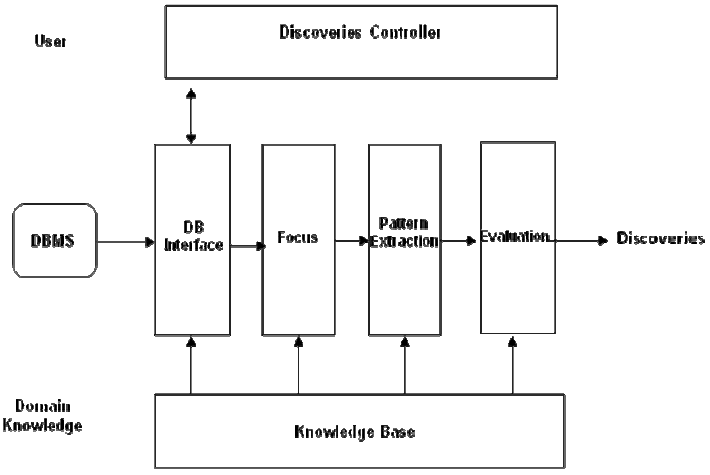


Figura 1.6: Architettura di un sistema di *KDD*

In questa architettura, l'utente può gestire ogni passo del processo di mining.
Eventuale conoscenza di base, come le gerarchie spaziali e non-spaziali di concetti, o altre informazioni sul database, è memorizzata in una base di conoscenza (*Knowledge base).*
I dati sono prelevati dal DBMS, usando un'opportuna interfaccia (DB interface) che consenta l'ottimizzazione delle interrogazioni. Le regole ed i modelli sono scoperti dal modulo *Pattern Extraction*. Questo può usare tecniche statistiche, *machine learning* e DM insieme con gli algoritmi di geometria computazionale per eseguire l'operazione di individuazione delle regole e delle relazioni e di eliminare quella parte della conoscenza considerata banale e ridondante.

1.7.1 I metodi per il *KDD* nelle Basi di Dati Spaziali

E' ben noto che i dati spaziali possono essere descritti usando due proprietà differenti: geometriche e topologiche.
Le proprietà geometriche riguardano ad esempio, posizione, area, perimetro, ecc., mentre le proprietà topologiche riguardano adiacenza (l'oggetto A è vicino all'oggetto B), inclusione (l'oggetto A è all'interno dell'oggetto B), ecc.
I dati geografici, come più volte ribadito, consistono di oggetti spaziali e della descrizione (non-spaziale) di questi oggetti.
Il problema di fondo risiede nella tipologia di organizzazione degli stessi all'interno della base dati spaziale al fine di agevolare il processo di estrazione della conoscenza. A tale proposito esistono differenti soluzioni previste in letteratura, tra cui quella introdotta da Aref (1991) che prevede che la descrizione non-spaziale di tali oggetti possa essere memorizzata in una tradizionale base di dati relazionale, in cui uno degli attributi è un puntatore alla descrizione spaziale dell'oggetto.

Quindi, i metodi per la scoperta della conoscenza possono essere messi a fuoco sulle proprietà non-spaziali e/o spaziali degli oggetti.

Naturalmente, in un contesto scientifico tanto ampio, molteplici sono le tecniche che possono essere utilizzate per lo *SDM*, fra cui:

- knowledge Discovery basato sulla generalizzazione (generalization-based);
- metodi che utilizzano il *Clustering*;
- metodi che esplorano le *Associazioni spaziali*;
- metodi di *Approssimazione* ed *Aggregazione*;
- metodi di mining su immagini e su *Raster Databases*.

Ci si sofferma, nel corso del lavoro di tesi, sui metodi che utilizzano il Clustering in ambiti temporali e spazio-temporali, e per il resto si rimanda alla letteratura.

Capitolo 2

Stato dell'arte sui processi di Temporal Data Mining

2.1 Generalità sul Temporal Data Mining

Il Temporal Data Mining (TDM) è un campo di ricerca interdisciplinare in piena evoluzione, basato, su metodologie di tipo statistico (ad esempio si pensi all'analisi di serie storiche), di riconoscimento di pattern temporali, di analisi di database temporali, di ottimizzazione, di visualizzazione, di *computing* ad alto rendimento e di *parallel computing*.

In questo capitolo, si vuole fornire al lettore una descrizione critica del TDM, soffermandosi sui metodi di ricerca e sui relativi contesti applicativi.

Il punto di partenza è quello di presentare una descrizione generale del problema, motivando l'importanza delle problematiche legate al TDM nel processo di estrazione della conoscenza, in riferimento a basi di dati temporali, (Kwnoledge Discovery Temporal Databases, *KDTD*).

Una volta inquadrate le problematiche generali di un tale contesto di ricerca, una scelta obbligata appare quella di produrre una ricognizione dei formalismi sui metodi, sui modelli e sulle tecniche di analisi nel TDM.

2.2 Definizioni e obiettivi del TDM

Una delle principali componenti di un processo di *KDTD* è quella di basare l'analisi su algoritmi specifici che consentano, appunto, l'estrazione e la ricognizione dei *pattern temporali*.
Le problematiche più comuni legate a tale contesto risiedono:

- nella difficoltà di accesso ai dati temporali, quando l'utente non sa o non è in grado di descrivere l'obiettivo in termini di domanda specifica;
- nella difficoltà del reperimento di tutte quelle informazioni relative al tempo, quando si abbia a che fare con un insieme di dati temporali di grosse dimensioni, e così via.

2.2.1 Gli obiettivi

Il Temporal Data Minig rappresenta un aspetto specifico di un processo di *KDTD*, con il compito di far emergere e classificare la struttura sottostante ai dati temporali, mettendo in luce funzionalità e modelli temporali, durante tutto l'intero processo di ricerca della conoscenza.
Appare evidente che il TDM è interessato all'analisi dei dati temporali al fine di individuarne gli schemi (funzionalità) e le regolarità temporali racchiuse negli insiemi di dati temporali. Inoltre, le tecniche di TDM si rifanno a tipologie di analisi esplorativa ed automatica dei dati.
Attualmente, la ricerca scientifica in tema di TDM sta assolvendo al delicato compito di condurre ad un nuovo sistema di interazione con basi di dati temporali, riuscendo, mediante tecniche sempre più sofisticate, a limitare gli

inconvenienti derivanti dalle, sempre più comuni, problematiche di multidimensionalità dei dati.

2.3 Tecniche e funzionalità del TDM

Le tecniche di TDM più note sono quelle sviluppate con un orientamento verso i grandi volumi di dati relativi al tempo, utilizzando i molteplici dati temporali raccolti, tentando, per quanto possibile, di pervenire a conclusioni certe. Il processo di acquisizione della conoscenza si articola nel seguente modo:

- raccolta e preparazione di un insieme di dati temporali in un database, che rappresenta il campione di riferimento;
- individuazione della metodologia di analisi che consenta di sviluppare una rappresentazione ottimale della struttura dei dati.

Una volta che la conoscenza temporale è stata acquisita, questo processo può essere esteso ad un più grande insieme dei dati, col presupposto di formare una struttura simile a quella dei dati del campione.

Una tra le forme più comuni delle tecniche di TDM è quella relativa alla scoperta di funzioni (regole). I vari tipi di funzioni temporali possono essere assimilate alla dipendenza dei dati dal loro dominio di applicazione. Tali funzioni temporali (o regole) possono essere, inoltre, costruite sulla base di:

- un'induzione di tipo *Bottom-Up* (dal basso verso l'alto), ovvero sulla base di un'analisi "attiva", in cui i dati stessi suggeriscono possibili ipotesi sul significato del loro contenuto. Si tratta di individuare fatti, relazioni, tendenze, pattern, associazioni, eccezioni e anomalie, che sfuggono all'analisi manuale;
- un'induzione di tipo *Top-Down* (dall'alto verso il basso), ovvero sulla base di un'analisi "passiva", atta a verificare se un certo modello (ipotesi) è coerente con i dati a disposizione. L'ipotesi o il modello

sono formulati dall'utente sulla base della sua esperienza e sono verificate mediante inferenza statistica o mediante estrazione di informazioni dal database, mediante interrogazione della base dei dati (*query language*).

2.3.1 Le funzionalità

Secondo le tecniche di estrazione della conoscenza temporale e, più in generale, dell'analisi statistica delle serie storiche, il TDM può coinvolgere tutta una serie di ambiti di ricerca, e, ad oggi, non sembra essere stato individuato una sorta di "protocollo" scientifico sull'argomento. In particolare, di seguito, saranno presentati i principali ambiti scientifici coinvolti, senza, però, trascurarne modelli e concetti sottostanti.
Gli ambiti disciplinari del TDM includono:

- *la caratterizzazione e la comparazione dei dati temporali, l'analisi dei cluster temporali;*
- *la classificazione temporale;*
- *le regole di associazione temporali;*
- *l'analisi dei pattern temporali;*
- *l'analisi dei trend e le previsioni temporali.*

In tali ambiti disciplinari, si pensi, ad esempio all'importanza delle indagini epidemiologiche, utili, appunto, a monitorare la crescita del bisogno, della domanda e della risposta di cure palliative a livello di popolazioni definite. L'impiego del TDM, in questo, come in innumerevoli ambiti disciplinari, rappresenta un utilissimo contributo alla ricerca, che se ben sfruttato può contribuire alla creazione di modelli di riferimento temporali utili all'interpretazione delle casistiche comuni.
Tuttavia, la creazione di un nuovo modello di dati temporali necessita, quasi sempre, di essere sviluppato sulla base della:

- struttura dei dati temporali (*temporal data structure*);
- semantica temporale (*temporal semantic*).

Ulteriori teorie di analisi su dati temporali possono avere bisogno di essere sviluppate sulla base di una avvenuta estrazione di una parte concettualmente rilevante, contenuta nella logica di base di un modello, cercando di realizzare un formalismo logico in grado di esprimere la conoscenza quantitativa ed una significativa approssimazione della realtà.

2.3.2 La classificazione

L'obiettivo di base della classificazione temporale è quello di predire connessioni nel tempo che si possono riscontrare all'interno di un database temporale.

Il problema, in genere, è diretto a determinare il più probabile dei valori da attribuire ad una variabile temporale, previsto per ogni campo, in base a dati di addestramento della variabile obiettivo, per ogni osservazione, fatte sulla scorta di una serie di ipotesi a priori.

Un punto focale delle tecniche di classificazione temporali risiede nelle problematiche di valutazione della densità.

Negli ultimi anni, molto lavoro è stato fatto nell'ambito della classificazione temporale, facendo ricorso ad un approccio statistico basato sulla previsione di modelli. In particolare, sono state previste tecniche per la valutazione di variabili categoriche, per la ricerca degli stimatori di densità e metodi del *K-means*. Queste tecniche sono basate su teorie statistiche applicabili a grosse basi di dati.

Finora, comunque, non è stato dato abbastanza risalto alle tecniche di classificazione temporali. Negli ultimi anni, l'idea di fondo della classificazione temporale è stato l'uso diretto di tecniche di campionamento nell'ambito dei metodi di analisi di serie cronologiche, al fine di sviluppare un modello per le sequenze temporali.

2.3.3 La Temporal Cluster Analysis

Il Clustering temporale (o Temporal Cluster Analysis) è una tecnica di analisi multivariata (l'oggetto in esame è formato da almeno due componenti) su dati temporali, volta alla selezione e al raggruppamento di elementi omogenei in un insieme di dati temporali. Il clustering temporale, così come il clustering, in generale, si basa sul concetto di distanza tra due elementi. Infatti, la bontà delle analisi ottenute dagli algoritmi di clustering dipende, essenzialmente, da quanto è significativa la metrica e, quindi, da come è stata definita la distanza. Quest'ultima, riveste un ruolo fondamentale: gli algoritmi di clustering raggruppano, infatti, gli elementi a seconda della distanza e, quindi, l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme. Le principali tecniche di clustering si basano essenzialmente su due filosofie:

- Dal basso verso l'alto: questa filosofia prevede che, inizialmente, tutti gli elementi siano considerati cluster a sé stanti. L'algoritmo provvede ad unire i cluster più vicini, successivamente. L'algoritmo continua ad aggiungere elementi al cluster fino ad ottenere un numero prefissato di cluster, oppure fino a che la distanza minima tra i cluster non supera un certo valore;
- Dall'alto verso il basso: all'inizio tutti gli elementi si trovano in un unico cluster; successivamente l'algoritmo inizia a dividere il cluster in tanti sotto-cluster di dimensioni inferiori. Il criterio che guida la suddivisione è quello di cercare di ottenere elementi omogenei. L'algoritmo procede fino a che non ha raggiunto il numero prefissato di cluster. Questo approccio è anche detto gerarchico. Gli algoritmi gerarchici possono essere agglomerativi (bottom-up) o divisivi (top-down). Negli algoritmi agglomerativi, ogni elemento forma inizialmente un cluster separato e mano a mano gli elementi più simili vengono uniti per formare cluster sempre più grandi. Gli algoritmi

divisivi, invece, partono dall'intero set di dati e proseguono dividendolo in cluster sempre più piccoli.

Inoltre, il *clustering temporale*, così come la ricerca di similarità è un aspetto che compare in molte branche della ricerca scientifica. In letteratura, due sono i fondamentali approcci di base per analizzarli:

- la misura della similarità temporale;
- il metodo della partizione temporale ottimale.

Riguardo alla misura della similarità temporale, si fa riferimento ai cosiddetti coefficienti di similarità, che hanno il compito di fornire la misura del grado di associazione fra osservazioni temporali, con un campo di variazione compreso, generalmente da *0* ad *1*. Tali valori limite corrispondono, rispettivamente, al caso di osservazioni del tutto disgiunte, prive di elementi comuni, ed al caso di osservazioni identiche fra loro.

Fra i molti coefficienti disponibili una importante distinzione è quella che deve essere fatta fra coefficienti simmetrici e coefficienti asimmetrici. All'interno di un vettore di misure relativo ad una osservazione può accadere che per uno o più descrittori siano stati rilevati dei valori nulli. E' evidente che in alcuni casi tali valori corrispondono ad un dato certo, almeno nei limiti dell'errore proprio dei metodi di campionamento e di determinazione (es. nell'ambito di un'analisi di dati temporali in ecologia, l'assenza di un certo inquinante), mentre in altri casi lo zero indica piuttosto l'assenza di informazione (es. tipologia di dato non è rinvenuta in un certo campione). Nel primo caso la scelta dovrà cadere su un coefficiente simmetrico, ai fini del cui calcolo i dati nulli hanno il medesimo valore comparativo degli altri, mentre nel secondo caso dovranno essere utilizzati coefficienti asimmetrici, in modo tale da evitare di definire una elevata similarità sulla base di informazioni non certe (quale ad esempio, la simultanea assenza di un elevato numero di specie in due stazioni che hanno poche o nessuna specie in comune).

Inoltre, si possono verificare casi specifici in cui un altro coefficiente, non compreso fra quelli descritti in precedenza, potrebbe risultare più adatto ad affrontare una particolare problematica, ma è bene sottolineare il fatto che la

scelta di un coefficiente di similarità rappresenta comunque, in qualche misura, un passo arbitrario in una procedura di analisi.

Un ulteriore approccio di analisi dei coefficienti di similarità è quello di convertirli in misure di distanza o, più propriamente, di dissimilarità, semplicemente considerandone il complemento ad 1 (cioè: D_{jk}=1-S_{jk}).

Qualunque sia la scelta del criterio di similarità da adottare (su cui si ritornerà nel capitolo 3), è da dire che una delle esigenze più comuni del *clustering temporale* è quella di raggruppare gli oggetti appartenenti ad un insieme dato, in modo tale da definire dei sottoinsiemi il più possibile omogenei. Per raggiungere questo risultato, identificando una partizione, cioè una collezione d'oggetti tale che ogni oggetto appartenga ad un solo sottoinsieme o classe, è necessario disporre di una procedura o di un algoritmo adatti alla natura dell'informazione disponibile, del problema da affrontare e degli oggetti stessi.

Le procedure di tipo soggettivo, in quest'ambito, hanno un ruolo molto più importante di quanto non si pensi comunemente. Basti considerare il fatto che un approccio di questo tipo, per quanto codificato in un quadro tassonomico di riferimento (vedi capitolo 4), delle attività fondamentali della ricerca nei contesti tipici del Temporal Data Mining, serve a classificare fenomeni che, all'apparenza, non sembrano differenziarsi di molto in istanti temporali contigui, come per le sequenze e per le serie storiche. Inoltre, prima che gli algoritmi di classificazione oggi disponibili venissero sviluppati, cioè fino a tutti gli anni '50, il modo più sofisticato di ottenere una partizione di un insieme di oggetti (o osservazioni) multivariati consisteva nel rappresentarli nello spazio dei loro descrittori o in quello definito da due o più assi principali, ricercando manualmente gli insiemi di punti più omogenei.

Nonostante, come appena accennato, le tecniche di classificazione siano tutte abbastanza recenti, esse costituiscono un insieme tanto ricco quanto diversificato. In linea di massima, tali tecniche si muovono su due fronti: quello di una suddivisione gerarchica, in cui si procede tipicamente per aggregazione successiva dei dati, e quello di una suddivisione di tipo non

gerarchica, in cui si procede alla divisione dell'insieme dei dati grezzi o per successivi aggiustamenti di una prima partizione.

Alcuni Autori preferiscono utilizzare il termine *clustering* per indicare i soli metodi non gerarchici, riservando il termine *classificazione* per quelli gerarchici.

Nell'ambito del seguente lavoro di tesi ci si sofferma maggiormente sul primo dei due criteri poiché esso è largamente utilizzato e compreso, indipendentemente dal contesto applicativo. La trattazione, si baserà principalmente sulle tecniche di clustering di dati temporali, ma è evidente che in alcuni casi può essere interessante e/o necessario ottenere, piuttosto, una partizione di un insieme di descrittori. E' importante sottolineare, inoltre, che una partizione dettata dall'impiego di una qualsivoglia tecnica di clustering è, a tutti gli effetti, un descrittore aggiuntivo (e sintetico) dell'insieme di oggetti in esame. L'appartenenza ad un cluster, infatti, se codificata in maniera appropriata, può essere utilizzata come una variabile di sintesi per ulteriori elaborazioni dell'informazione disponibile.

Inoltre, non è da sottovalutare l'esistenza di un approccio del tutto particolare ai problemi di clustering, che ben si adatta ai problemi più disparati, risultando molto efficace in alcuni contesti relativi a strutture di dati temporali. Si tratta dell'approccio basato sul concetto di *fuzzy set*, secondo cui l'appartenenza di un oggetto ad una classe (cioè ad un *fuzzy set*) non viene espressa in forma binaria, ma piuttosto in forma probabilistica. E' evidente che questo tipo di logica è molto più vicina a quella che tutti noi utilizziamo nella vita di tutti i giorni, quando ci riferiamo a categorie, i cui limiti sono, difficilmente, definibili in maniera univoca, poiché sfumano le une nelle altre, senza soluzione di continuità.

Qualunque sia l'approccio utilizzato, lo scopo della clusterizzazione resta sempre quello di determinare l'intrinseco raggruppamento in un set di dati temporali non classificati. La modalità di decisione circa la bontà della clusterizzazione, resta la scelta più delicata, dal momento che non esiste un criterio migliore in assoluto che sia indipendente dallo scopo finale della

clusterizzazione. Di conseguenza, è l'utente stesso che deve fornire questo criterio in modo che il risultato della clusterizzazione vada incontro alle sue necessità. Grazie ad una corretta clusterizzazione si può, ad esempio, essere interessati, ad esempio, a trovare:

- rappresentanti di gruppi omogenei (*data reduction*);
- cluster naturali e descrivere le loro proprietà sconosciute(*tipi di dato naturali*);
- raggruppamenti utili e convenienti (*classi di dati utili*);
- oggetti inusuali all'interno del set di dati (*outlier detection*).

Nell'ambito di analisi di dati temporali, inoltre, le tecniche di clustering utilizzate in un contesto di TDM, consentono di raggruppare i dati secondo criteri di similarità ed di ottimizzazione delle funzioni relative al set di dati. Se si conosce a priori il numero di classi, le tecniche di clustering, possono essere divise in tre categorie:

- tecniche basate sulla distanza-metrica;
- tecniche basate sui modelli;
- tecniche basate su un criterio di partizione.

Uno dei principali fondamenti di un processo di clustering è, come si è detto, l'elaborazione di misure di distanza o di similarità tra un set di sequenze. Tali misure si basano su metodologie standard, che si rifanno ad esempio al clustering agglomerativo.

Nell'ambito delle tecniche basate sulla distanza metrica si fa riferimento ai cosiddetti coefficienti di distanza metrici, sviluppati per trattare dati di tipo quantitativo e, con poche eccezioni, trattano lo zero come una misura, e non come una mancanza di informazione.

La più familiare fra le misure di distanza è certamente quella euclidea, di cui si tratterà nel capitolo successivo, che corrisponde esattamente a quella che si può calcolare o misurare nello spazio fra due oggetti fisici. Basti rilevare, per ora, che il quadrato della distanza euclidea, che non di rado viene utilizzato al posto di quest'ultima, rappresenta la cosiddetta *semimetrica*.

Sul problema della distanza ci si soffermerà, in particolare nel successivo.

Per quanto concerne le *Tecniche basate sui modelli*, esse si fondano su l'assunzione che è possibile associare un modello analitico per ogni cluster, allo scopo di trovare una serie di modelli che meglio si adattano alla serie di dati originaria.

Tra le tecniche basate sui modelli, assume rilevanza quella fondata sui cosiddetti *Metodi generativi* che assumono che esista, alla base del set di sequenze, un processo generatore, per cui il modello si basa sulla stima di parametri di somiglianza del modello, per ciascun cluster. Si pensi ad esempio alla combinazione di modelli Polinomiali (Gaffney and Smyth, 2003), ai processi ARMA (Piccolo, 1990; Xiong and Yeung, 2002), alle Catene di Markov e ai Modelli di Hidden Markov (Cadez et al., 2000; Alon et al., 2003).

Tale approccio, considerato, ormai, a pieno titolo, come una sorta di "approccio globale", consente di:

- integrare la conoscenza prima di trovare il numero corretto di cluster;
- fornire le basi la gestione del problema della modellazione, nonché per la creazione di cluster di sequenze, con diverse ampiezze;
- ottenere utili risultati per lo studio della dinamica e del comportamento di fenomeni a livello eterogeneo.

Inoltre, le *Tecniche basate su un criterio di partizione* sono basate su criteri di estrazione di una serie di caratteristiche, per ciascun individuo, consentendo l'acquisizione di informazioni circa i dati relativi alle sequenze temporali. Il problema risiede, quindi, nell'individuare la sequenza dei cluster mediante l'utilizzo di vettori di funzioni. Il problema, in tale approccio, è, appunto, quello di ridurre a più vettori di funzioni.

Talvolta, gli approcci sopracitati possono essere combinati tra loro, dando luogo alla cosiddetta *Probability-based* verso la *Distance-based cluster analysis* Se il numero dei gruppi non lo si conosce a priori, si fa ricorso ad algoritmi di clustering non gerarchico per trovare il numero dei *k* gruppi.

Di recente, le tecniche di TDM che sono state sviluppate hanno fatto ricorso all'algoritmo EM (trattato ne capitolo 4) e alla tecnica di *cross validation* col metodo Monte-Carlo.

2.3.4 L'induzione

Una database temporale è un deposito di informazioni riferite al tempo, ma ancora più interessante quanto si possa ricavare da esso, ovvero l'estrazione della conoscenza implicita.
A tale proposito, esistono due tecniche fondamentali per acquisire tale conoscenza:

- *Tecniche di deduzione temporale.* Tendono a desumere le informazioni in conseguenza di una logica temporale di fondo, relativa alle informazioni desunte dal database temporale di cui si dispone;
- *Tecniche induzione temporale.* Tendono a desumere le informazioni temporali che sono generalizzate nel database temporale e fanno uso degli *Alberi di decisione* e delle *Regole di associazione*.

2.4 Problematiche di TDM

Negli ultimi anni, si sono evidenziati due generi di problemi riguardanti il TDM. Il primo risiede nei problemi della similarità, ovvero nella ricerca di similarità all'interno di sequenze o dati temporali presenti nei grossi database temporali (TDB) o alla scoperta di tutte le coppie di sequenze simili. Il secondo risiede nella risoluzione del problema della periodicità, ovvero nella possibilità di trovare modelli "periodici" nei database temporali.

2.4.1 La similarità

Nelle applicazioni di TDM, è spesso necessario cercare, all'interno dei database temporali, le sequenze simili, rispondenti ad una query. In questi casi si parla di "ricerca della similarità", coinvolgendo nell'analisi tutte le tematiche legate alla multidimensionalità delle serie storiche nei TDB, per scoprire quali e quante serie sono simili tra loro. Questo è, senz'altro, uno dei problemi più delicati del TDM, che ha ispirato la ricerca scientifica, al fine fornire definizioni del problema della similarità e di delinearne metodologie standard di analisi in così ampio ventaglio applicativo.

In base a quanto, fin'ora esposto risulta ormai evidente che le tecniche di TDM possano essere applicate ai problemi di similarità. I principali passi da seguire per risolvere il problema della similarità sono:

- *Definire la similarità.* Questo passaggio consente di trovare la similarità fra le sequenze con differenti fattori di misura in base ai dati di partenza;
- *Scegliere una sequenza di query.* Questo passo consente di far luce sugli obiettivi da raggiungere quando si lavora su grosse mole di dati, mediante una classificazione delle sequenze (come ad esempio nel TDB);
- *Elaborazione dell'algoritmo per TDB.* In questa fase, si stabiliscono le metodologie statistiche da adottare (esempio, trasformazione di Wavelet, oppure rimozione dei dati affetti da rumore, interpolazione dei dati mancanti) a TDB;
- *Elaborazione di un algoritmo di approssimazione.* In questa fase, è possibile sviluppare uno schema di classificazione per il TDB, secondo la definizione di similarità, usando tecniche di estrazione di alcuni dati (per esempio, visualizzazione).

Tutti i possibili risultati prodotti dai problemi di similarità in TDB possono essere utilizzati o ricondotti a tecniche di associazione, previsione, ecc..

2.4.2 La periodicità

Il problema della periodicità è una questione legata all'individuazione di modelli periodici o di ricerca della periodicità all'interno di TDB. Il problema è collegato a due concetti: definizione del modello e dell'intervallo di tempo da considerare. Infatti, in tutta la sequenza selezionata del TDB, si è interessati all'individuazione di modelli che ripetono ricorsivamente intervalli di tempo (periodi), o all'individuazione di modelli in cui si ripete una sequenza in un TDB, così come all'intervallo che corrisponde al periodo del modello. Per risolvere i problemi periodicità nei TDB, è necessario procedere per fasi:

- *Definire di un problema*, in un dato periodo, in base ad alcuni presupposti. Questa fase permette di conoscere che genere di tecnica di ricerca della periodicità desideriamo effettuare nel TDB;
- *Sviluppare un insieme di procedure*. Questa fase permette di usare le proprietà di periodicità elle serie cronologiche, al fine dell'individuazione dei modelli periodici da un sottoinsieme di TDB usando appropriate procedure;
- *Effettuare procedure di simulazione*. Questa fase per la quale la precedente è propedeutica, consente di trovare modelli di estrazione da TDB.

2.5 Time Series Data Mining

Le serie cronologiche sono una raccolta elementi la cui variazione cambia rispetto al tempo di osservazione. Una caratteristica che distingue i dati, sotto forma di serie cronologiche, da altri tipi di dati è che, generalmente, i valori della serie, ad istanti differenti di tempo, saranno correlati.

L'applicazione delle tecniche di analisi di serie cronologiche nel TDM spesso è denominata Time Series Data Mining (*TSDM*). In tale ambito, gran parte della ricerca scientifica si è dedicata alla raccolta, all'individuazione, alla pulitura dei dati e all'individuazione della metodologia più appropriata per osservare, nonché scoprire, pattern temporali utili.

La metodologia di analisi delle serie storiche nell'ambito del TDM è stata applicata nelle seguenti categorie scientifiche:

- *Rappresentazione delle sequenze temporali*. Ci si riferisce alla rappresentazione dei dati prima che le tecniche di *KDD* su dati temporali reali vengano apportate, i cui metodi principali si riferiscono alla:
  - "Rappresentazione generale dei dati" (modelli stazionari/mobili; modelli continui o discontinui; modelli lineari/non lineari; modelli distributivi);
  - "Trasformazione generale dei dati". Ci si riferisce alla rappresentazione dei dati sotto forma di serie cronologiche con trasformazioni di tipo continue o discontinue (per esempio, trasformata di Fourier, trasformata di Wavelet e trasformata di discretizzazione).
- *Misura della sequenza temporale*. Si tratta di un elemento caratteristico del TDM che fa riferimento alla definizione di somiglianza e/o periodicità in una sequenza temporale (o, sub-sequenze di una sequenza temporale) o fra le sequenze temporali. In quest'ultimo contesto esistono due metodi:
  - "Misura della distanza caratteristica in un dominio temporale". Si tratta di trovare una misura caratteristica della distanza nei domini temporali continui o discontinui (ad esempio la funzione della distanza euclidea);
  - "Misura della distanza caratteristica in differenti domini che non siano quelli temporali". Si tratta di trovare una misura caratteristica della distanza in altri domini, continui o discontinui,

che non siano quelli temporali (per esempio, funzione di distanza fra due distribuzioni).

- *Previsione della sequenza temporale*. L'obiettivo principale della previsione è quello di predire alcuni campi in una base di dati, basata sul dominio temporale. Le tecniche possono essere classificate in due modelli:
  - "Modelli di classificazione temporale". L'obiettivo di base è di predire che il più probabile si riferisca ad una variabile categorica (quindi, che si predica ad esempio un "codice di categoria") nel dominio temporale.
  - "Modelli temporali di regressione". L'obiettivo di base è di predire una variabile numerica in un insieme, usando trasformazioni differenti presenti in letteratura (per esempio, lineare o non lineare) sulla base di dati, per trovare le informazioni temporali.

Capitolo 3

Principi alla base del Temporal Clustering Data Mining: misure di distanza e tecniche di trasformazione

3.1 Introduzione

Come mera fase preliminare della trattazione sul clustering, in un contesto di TDM, occorre, chiarire che cosa si intenda per clustering, quali sono le problematiche coinvolte, quali sono le applicazioni e, quindi, i requisiti in base ai quali valutare le diverse tecniche di clustering.

L'attività di clustering, o più formalmente la *cluster analysis*, può essere definita come quel processo di organizzazione di oggetti (*pattern*) in gruppi (cluster), i cui membri siano fra loro "simili". Ciò significa che i *pattern* di un cluster devono presentare una forte similarità fra loro e, al contempo, un'inferiore similarità con i *pattern* degli altri cluster.

La definizione fornita, sebbene sia intuitiva e apparentemente semplice, apre un ventaglio di problematiche operative legate alla definizione e alla

misurazione della similarità, nonché riguardo ai metodi e alle procedure con cui effettuare il raggruppamento dei *pattern* in base a detta similarità.

E' importante fin d'ora sottolineare quali siano i nuovi scenari applicativi della *cluster analysis*, allo scopo di comprendere il rinnovato interesse, sviluppatosi in questi ultimi anni, attorno ad un'attività tipica della statistica e già ampiamente studiata negli anni è'70.

Recentemente, le tecniche di clustering hanno trovato applicazione:

- nell'ambito della segmentazione delle immagini, per esempio per una più accurata risoluzione delle *query* di similarità in un database di immagini;
- nel riconoscimento di *pattern* e nel riconoscimento ottico dei caratteri (OCR);
- nelle analisi dei database temporali e spazio-temporali e delle enormi moli di dati provenienti da immagini satellitari del territorio;
- nello studio della biologia molecolare da parte dell'industria farmaceutica per la determinazione delle aree di aggancio fra proteine;
- nel recupero di informazioni (*information retrival*), con particolare relazione allo sviluppo del web;
- nell'estrazione della conoscenza da basi di dati, per eseguire ad esempio un'analisi di mercato o una profilatura della clientela (attività tipiche del data mining);
- nell'ambito generale del data mining, sia come attività indipendente (*stand-alone*), per le analisi della distribuzione dei dati, che come fase di pre-processing, prima di applicare successivi algoritmi di estrazione della conoscenza.

Quello che si evince dai domini applicativi delineati è di dover porre attenzione al fatto che si tratti di analisi condotte su grosse mole di dati, con cui, sempre più spesso, oggi ci si trova a dover avere a che fare e ad operare.

Si tratta di dati la cui complessità computazionale crea problemi di efficienza che insorgono all'aumentare della dimensionalità.

In virtù di queste considerazioni, la *cluster analysis* viene attualmente identificata come un'attività del DM (Figura 3.1) che, come si è visto in precedenza, prende il nome di *data clustering*, nell'ambito della quale sono stati presentati dalla metà degli anni '90, i nuovi algoritmi di clustering, con l'obiettivo di essere sufficientemente scalabili senza, per questo, potere rinunciare all'accuratezza del risultato prodotto.

E' da rilevare, inoltre, come la vastità dei dati sui quali si deve operare comporti tipicamente l'impossibilità di disporre a priori di un'insieme di classi predefinite, conducendo conseguentemente a considerare, in questo caso, la *cluster analysis* come un'attività non supervisionata. Quest'ultima osservazione ha notevoli implicazioni nella valutazione degli algoritmi di clustering, in quanto è legata alla possibilità di conoscere e stimare correttamente i parametri iniziali, con i quali detti algoritmi operano. Se da un lato, da detti parametri, talvolta in modo determinante, dipende la bontà del risultato, dall'altro, ragioni di efficienza non sempre consentono una loro modifica mediante successive iterazioni.

3.2 Clustering in un contesto di TDM

Il clustering temporale predispone gli elementi di una serie in gruppi (cluster) con *pattern* simili.

Differenti tecniche di clustering sono state sviluppate ed applicate all'analisi di dati sotto forma di serie storiche. Di qualunque tecnica si tratti, comunque, il clustering si occupa, principalmente, di raggruppare tutti gli elementi di una serie che mostrino simile comportamento tra i differenti punti.

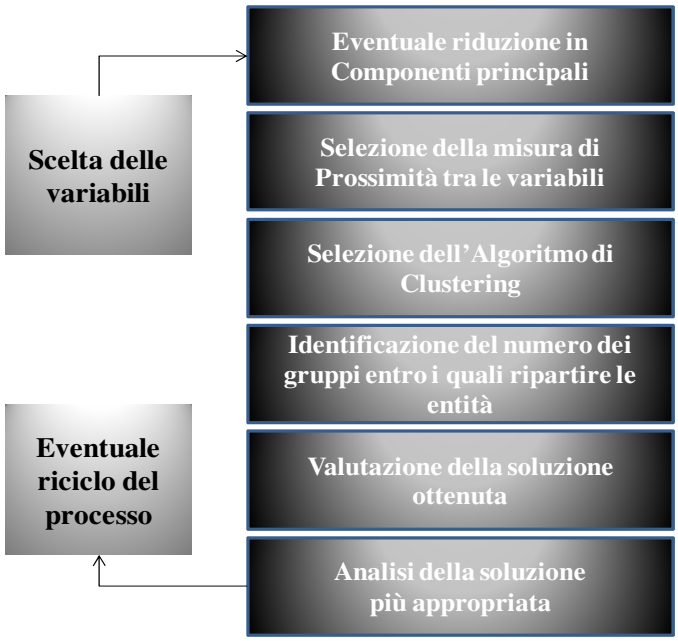


Figura 3.1: Fasi di un processo di clustering

Quindi, all'interno degli algoritmi di clustering bisogna inserire un unico dato, per ogni elemento della serie che appartenga ad un gruppo in cui si trovino forti correlazioni.

Nel seguito, saranno analizzati alcuni dei più rilevanti principi alla base del Clustering nel TDM, facendo riferimento alle problematiche della similarità e della distanza, fonte di ispirazione degli algoritmi che saranno poi trattati nel successivo capitolo.

Il principio alla base del clustering è quello delle distanze metriche. In un contesto di Temporal Data Mining, tale principio rappresenta una distanza tra un elemento della serie rispetto agli altri. Si tratta, quindi, di raggruppare nello stesso cluster tutti quegli elementi che mostrano distanza minima tra di loro. Naturalmente, però, il concetto di distanza e l'omogeneità tra i gruppi che si vengono a formare, non necessariamente devono essere interpretati in senso

univoco. Ciò significa che non sempre un adeguato raggruppamento sia sinonimo di omogeneità nei gruppi che si vengono a formare (Figura 3.2).

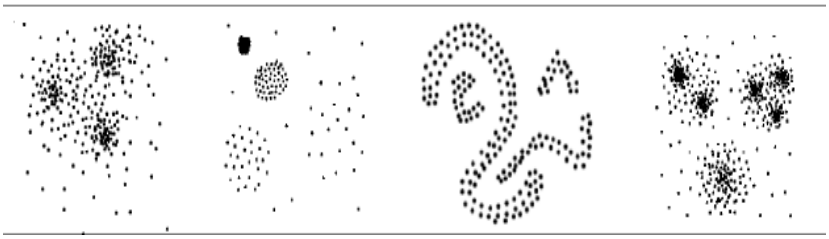


Figura 3.2: Diversità tra i tipi di raggruppamento

Anche per quanto riguarda il clustering nel TDM è possibile fare riferimento alla dicotomia classica di analisi, che vede la netta distinzione tra due tipi di clustering, a loro volta sotto classificati, quali:

- *Gerarchico*:
  - supervisionato, cioè quello che utilizza informazioni a priori sugli elementi della serie, per guidare l'algoritmo;
  - agglomerativo, cioè quello che non richiede di decidere a priori quanti saranno i cluster;
- *Non Gerarchico*:
  - non supervisionato;
  - divisivo.

Il clustering gerarchico è il più semplice da visualizzare; per far questo vengono utilizzati dei *dendrogrammi* che, oltre a formare i gruppi, mettono anche in relazione i singoli elementi della serie e consente di risalire alla radice che contiene tutta la gerarchia che si va, a poco a poco, ramificando fino ad arrivare al singolo elemento (Figura 3.3).

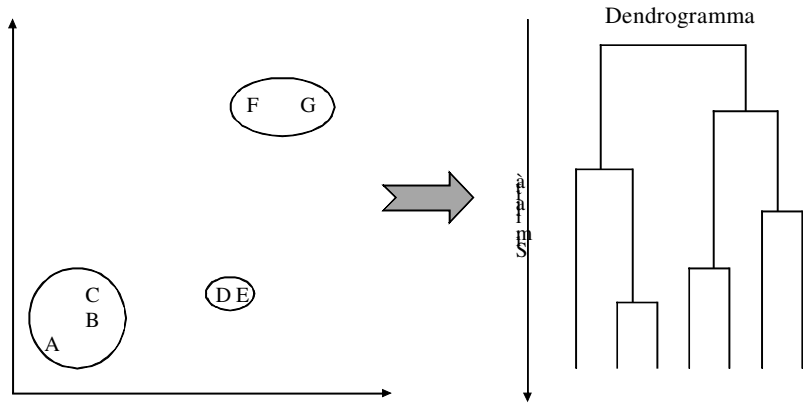


Figura 3.3: Trasformazione dei gruppi nel *dendrogramma*

Questo tipo di clustering consta delle seguenti fasi::

- Posizionare ogni osservazione in un cluster distinto,
- Fondere i due cluster più simili;
- Proseguire finché resta un solo cluster, ricordando la sequenza delle fusioni intermedie;
- Costruisce un albero che mostra le similarità fra osservazioni.

Nell'ambito del *clustering* non gerarchico, una famiglia di tecniche tra le più sofisticate è quella delle tecniche divisive, di cui fanno parte il *k-means* e le Self Organizing Map (SOM), per la cui discussione si rimanda al capitolo 4 della tesi. Basti per ora sapere che il *K-means* utilizza tecniche computazionalmente più impegnative; non è visualizzabile con *dendrogrammi* e, generalmente, non è supervisionato. Il motivo per il quale esso entra a far parte della categoria dei clustering divisivi sta nel fatto che, tali tecniche, partono dal totale degli elementi per dividerli successivamente in gruppi. Si noti che il numero di gruppi finale in cui dividere gli elementi deve essere impostato a priori dall'utente.

Le SOM, poi, si fondano sui principi delle reti neurali ed è quindi un tipo di procedura supervisionata. Basti sapere, per ora, che la letteratura prevede tutta

una serie di tecniche e di varianti ai temi, che saranno, appunto, approfonditamente trattate nel successivo capitolo.

Il punto su cui, invece, è bene attrarre l'attenzione del lettore, per ora, è l'indispensabilità di eseguire il clustering, solo dopo, naturalmente, aver filtrato statisticamente i risultati; costruendo una serie di gruppi attorno agli elementi della serie, per evitare di portarsi dietro, il più possibile, elementi con valori non affidabili; tutto ciò, al fine di diminuire il rischio di costruire *cluster* errati.

L'insieme dei dati da inserire nel clustering rappresenta una matrice caratteristica, cioè una serie di *t* tempi che misurino i livelli che caratterizzano i *t* differenti tempi sperimentali.

Si denota con la matrice *X* di dimensione (*n*-elementi per *t*-tempi) la matrice dei dati caratteristici. Dette matrici non saranno altro che tabelle in cui per ogni riga si avranno i valori di uno stesso elemento, mentre in ogni colonna si avranno i valori di uno stesso istante.

Nella matematica odierna, per vettore si intende, più in generale, un insieme ordinato di quantità dette componenti. In un contesto di clustering temporale, ogni vettore è rappresentato da un singolo elemento, le cui componenti sono i segnali delle caratteristiche di espressione temporale. Praticamente, ogni elemento è rappresentato da un vettore, all'interno della matrice ed il principio alla base del clustering non fa altro che associare vettori che mostrano componenti paragonabili (simili segnali di tempo).

Poiché una grossa lista di numeri è difficile da valutare, i dati grezzi sono convertiti in rappresentazioni grafiche in cui ogni dato è rappresentato con un colore che lo caratterizza quantitativamente e qualitativamente. Ciò permette al ricercatore di effettuare un'esplorazione e una comprensione del risultato più intuitiva ed immediata.

Generalmente, i colori utilizzati vanno dal verde "saturato" (valore max negativo) al rosso "negativo" (valore max positivo); per gli elementi nulli, il colore utilizzato è solitamente il nero. Questo tipo di rappresentazione grafica

è tipica delle sequenze genetiche, ma in altri contesti di dati sono possibili anche altri tipi di rappresentazioni.

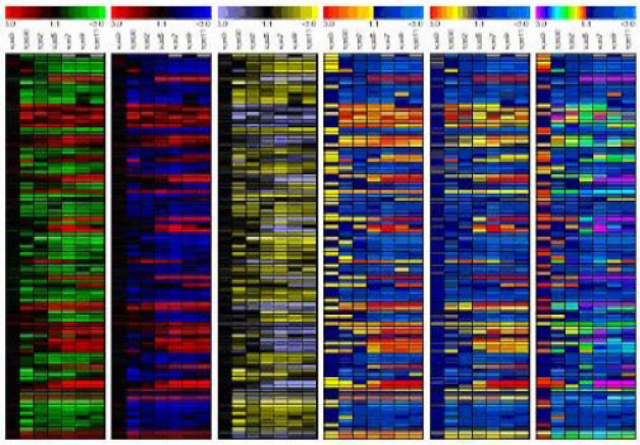


Figura 3.4: Esempio di sequenze genetiche

I cluster possono essere visualizzati anche in altre maniere:

- *Expression Graphs*, in cui ogni elemento in un cluster viene plottato separatamente
- *Centroid Graphs,* in cui vengono plottate la media e la deviazione standard di tutti i elementi del cluster.

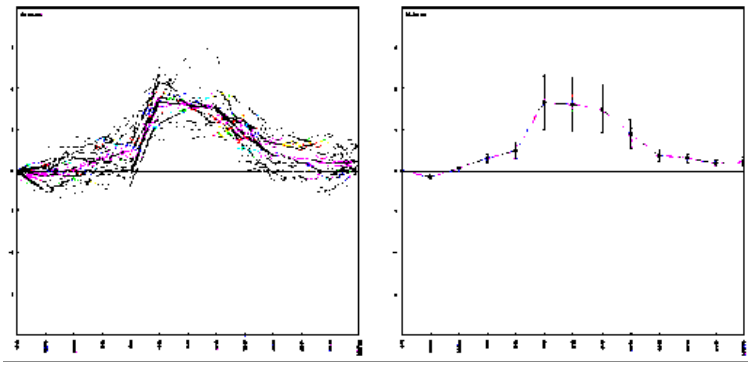


Figura 3.5: Rappresentazione di Expression e Centroid Graphs

3.3 Similarità e distanza nel tempo

Tutte le procedure di algoritmi di clustering usano misurare la somiglianza tra vettori per comparare i *pattern* simili.

Questo aspetto del problema va sotto il nome di "misura della similarità". Tale problematica rappresenta il cuore delle tecniche di clustering.

La misura della similarità va intesa come base di un problema di clustering ed, allo stesso tempo, fondamento per la risoluzione dello stesso. L'ottica della similarità è molto importante, dal momento che rappresenta la caratterizzazione degli ambiti su cui effettuare la ricerca.

Basti pensare alle differenti applicazioni possibili, che vanno:

- dalla biologia, in cui la si utilizza per derivare tassonomie di animali e piante;
- al marketing, per derivare e caratterizzare gruppi di consumatori;
- alla geologia, per derivare similitudini tra aree;
- all'analisi dei dati, che viene impiegata per studiare come i dati si distribuiscono nel tempo e nello spazio.

La distanza tra due elementi e/o esperimenti è computata sommando le distanze tra i loro rispettivi vettori.

Come si debbano determinare tali valori, dipende dalla misura di *distanza-similarità* utilizzata, nella determinazione della matrice delle distanze, tra cui, ad esempio: il *coefficiente di correlazione di Pearson*; l'R^2; la *covarianza*; la *distanza euclidea*; la *distanza di Manhattan*; il *Tau di Kendall*, ecc.

Per alcune di tali misure si rimanda agli opportuni approfondimenti. Infatti, se tratteranno solo alcune che, per altro, toccano tipicamente la sfera di analisi di un contesto tipicamente temporale.

In generale, comunque, ogni distanza misurata non sarà altro che una quantizzazione della relazione lineare tra due serie di misure x e y.

Mentre efficaci procedure di clustering sono ben note e sono accuratamente implementate in software commercialmente disponibili, quali MATLAB, SAS

e SPSS, ed altri; la definizione dei concetti di distanza tra serie è forse l'elemento più delicato su cui si concentra una buona parte della ricerca scientifica.

Dato un insieme di serie storiche, la funzione distanza fra di esse, nella sua definizione più generale, è una funzione non negativa, definita su ogni coppia di serie e tale che, un elevato livello di similarità fra due serie, è caratterizzato da un piccolo valore della loro distanza.

E' da dire che ogni funzione di distanza implementa un concetto differente di similarità. Per questo, non esiste una distanza ottimale che implementi il "vero" concetto di similarità; ogni distanza serve ad un obiettivo specifico.

Introducendo il concetto più generico di distanza, si può definire, preso T come tempo, la distanza come una funzione:

$$\left\| \cdot - \cdot \right\| : T \times T \rightarrow \Re \qquad [3.1]$$

In particolare, nel caso in cui lo spazio sia $\Re^n$ (con n>0), ovvero gli attributi siano di tipo numerico, si ricade nell'ambito trattato dalla geometria, pertanto, sono disponibili numerose proposte di distanze studiate in modo molto dettagliato in letteratura (la distanza "Euclidea" e di "Manhattan" o la distanza di "Hamming").

Uno spazio metrico è un insieme X, dotato di una funzione, stimata reale d: $XxX \rightarrow \Re$ (funzione di distanza o metrica), tali che per ogni $x, y,z \in X$, si ha:

$$\begin{aligned} &d(x,y) \geq 0; \\ &d(u,u) = 0; \\ &d(x,y) = d(y,x); \\ &d(x,z) \leq d(x,y) + d(y,z). \end{aligned} \qquad [3.2]$$

Si chiama "quasi-metrica" una funzione distanza che soddisfi le condizioni precedenti, eccetto la disuguaglianza triangolare. Si chiama ultrametrica ogni

funzione distanza che soddisfi le quattro condizioni precedenti, più la relazione:

$$d(a,c) \leq \max(d(a,b),d(b,c)) \qquad [3.3]$$

L'esempio di un prototipo di uno spazio metrico è definito come:

$$d(x,y) := |x - y| \qquad [3.4]$$

ed è valido in *R*.

A questo punto, è necessario introdurre le definizioni delle distanze cui si è accennato in precedenza.

Si parte dalla definizione di distanza "Euclidea", ovvero di quella in cui, presi due punti sul piano $u=(x_1, y_1)$ e $v=(x_2, y_2)$, la loro distanza sarà:

$$d(u,v) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad [3.5]$$

La funzione della distanza "Manhattan", invece, calcola la distanza tra due punti, misurata perpendicolarmente lungo gli assi, in un piano in cui *u* ha coordinate (x_1,y_1) e *v* ha coordinate (x_2, y_2), la distanza è:

$$d(u,v) = |x_1 - x_2| + |y_1 - y_2| \qquad [3.6]$$

Queste metriche hanno delle caratteristiche comuni: in particolare, la tendenza ad individuare regioni convesse, l'inapplicabilità al caso d'attributi non numerici (nominali) e un notevole peso computazionale anche perché sono funzioni usate molto frequentemente negli algoritmi di clustering (Figura 3.6).

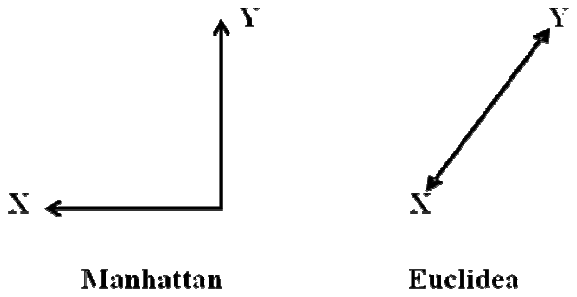


Figura 3.6: Esempio della distanza Manhattan ed Euclidea

Per rappresentare, invece, le soluzioni di un problema, avvalendosi di vettori booleani ad *n* componenti, interviene la distanza di "Hamming".

Si chiama distanza di Hamming, $d_H\ (s_1,s_2)$, fra due vettori booleani ad *n* componenti s_1 ed s_2, il numero delle componenti in cui essi differiscono (Figura 3.7).

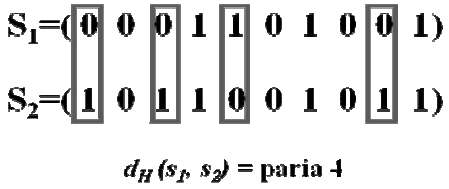


Figura 3.7: Esempio di una distanza di Hamming

Così come le problematiche di scelta della distanza da adottare per le tecniche di clustering, anche il problema della similarità appare di non meno rilevanza. E' per questo che si ritiene, in questa fase del lavoro di tesi, di esaminare alcuni tra i più diffusi concetti di similarità che sono stati proposti in letteratura.

Date due o più serie, prese negli stessi *N* intervalli temporali, la più semplice misura di similarità tra esse è la distanza L_p, considerando le serie come punti in uno spazio ad *N* dimensioni. L_1, talvolta chiamata distanza Manhattan, è

definita come la somma (o la media) dei moduli delle differenze fra le serie, L_2 è la distanza euclidea, L_q è la distanza di Minkowsky, definita come:

$$L_q = (\sum_{i=1}^{N} |a_i -_i b_i|^q)^{\frac{1}{q}}, L_\infty \qquad [3.7]$$

è la massima distanza tra punti omologhi.

Le distanze di cui sopra sono tutte metriche, in quanto rispettano le condizioni proprie della metrica. Esse hanno il vantaggio di essere semplici ma, al contempo, piuttosto restrittive. Innanzitutto, esse richiedono che le serie siano definite negli stessi punti. Inoltre, sono sensibili ad eventuali *outlier* e disallineamenti. Ad esempio, se due serie di prezzi con forti fluttuazioni sono molto simili nell'andamento, ma leggermente disallineate, la loro distanza euclidea può risultare molto grande.

Inoltre, in molte applicazioni si vogliono considerare simili serie a meno di fattori di scala e di spostamenti del valor medio. Ad esempio, le serie che rappresentano i prezzi di due azioni differenti possono avere lo stesso andamento e le stesse fluttuazioni percentuali, ma valori assoluti molto diversi. Perciò, si può estendere la definizione precedente, misurando la similarità fra serie, che sono state assoggettate a trasformazioni, sia per cambiarne i valori medi, sia per variarne la scala.

Un tale concetto di similarità è ancora troppo restrittivo per molte applicazioni finanziarie. Ad esempio, nel caso della valutazione del rischio creditizio, aziende differenti possono andare verso l'insolvenza, seguendo percorsi strutturalmente simili ma su un lasso di tempo differente. E' perciò importante identificare similarità anche fra serie che occupino intervalli di tempo differenti.

A tal fine è stato introdotto il concetto di *time-warping*. Una serie è soggetta a time-warping se, ai suoi elementi, vengono aggiunti elementi contigui

identici. Ad esempio, le serie (2; 21; 3; 4) e (2; 2; 1; 1; 3; 4; 4) sono identiche dopo un'operazione di time-warping.

Il time-warping consiste, perciò, nello stirare o comprimere localmente una serie. La distanza di time-warping è definita come la minore possibile distanza (ad esempio in senso L_p) dopo operazioni di time-warping. Una procedura di distanza time-warping allungherà o comprimerà localmente le serie fino a che la loro distanza non sia minima.

E' necessario, tuttavia, estendere ulteriormente il concetto di similarità per includere serie che sono simili solo in parte. Similarità di questo genere sono importanti quando si studiano, ad esempio, comportamenti particolari di fenomeni che abbiano seguito un *pattern* specifico in un certo intervallo (è il caso tipico di serie di titoli). Sono state avanzate varie proposte che conducono a considerare la distanza fra due serie solo in un sottoinsieme di punti.

Restando nell'ambito delle distanze metriche, un concetto differente di distanza è stato proposto da Bollobas, Das, Gunnopulos e Mannila (1997). Questi autori definiscono, innanzitutto, un intervallo di tolleranza che prescrive se punti omologhi sono considerati simili o meno. La distanza fra due serie è definita dal rapporto fra il numero di punti simili rispetto al numero di punti totale. Naturalmente, si possono assoggettare le serie ad operazioni di cambiamento di scala prima di calcolare la distanza in questo senso.

E' stata avanzata l'ipotesi che la similarità delle serie finanziarie possa essere meglio descritta da una ultrametrica, introducendo così una fondamentale gerarchizzazione nella similarità (Bonanno, Vandewalle e Mantegna, 2000 e Mantegna, 1999). E' stato mostrato (Ormerod e Mounfield, 2000) come si possa costruire una ultrametrica fra serie in modo molto semplice a partire dai consueti coefficienti di correlazione fra serie, definendo la distanza come:

$$d(a,b) = \sqrt{2(1 - C(a,b)} \qquad [3.8]$$

dove *C(a;b)* è il consueto coefficiente di correlazione.

Le distanze fra serie, definite nei precedenti paragrafi, sono esempi di distanze geometriche, calcolate direttamente sui punti delle serie, eventualmente dopo trasformazioni di scala o di base dei tempi. Sono stati proposti concetti di similarità, definiti su spazi correlati, formati da varie trasformate delle serie. La prima proposta di similarità fra serie, infatti, (Agrawal, Faloutsos e Swami, 1993) era basata sul proiettare le serie nello spazio delle Fast Fourier Transform (FFT). Si tratta di una trasformata integrale, fra le più note in analisi, con importantissime applicazioni in ambito scientifico. Particolare impiego se ne fa in fisica, nell'ambito delle scienze acustiche, ottiche, cristallografiche , in matematica, nelle scienze probabilistiche e statistiche, ecc., per trasmettere segnali in termini di frequenze e relative ampiezze. Si pensi alla sequenza delle note musicali ed alla loro scomposizione in onde sonore. La FFT è un algoritmo ottimizzato per calcolare la trasformata di Fourier discreta (detta DFT) e la sua inversa. La FFT è di grande importanza per una grande varietà di applicazioni, dall'elaborazione di segnali digitali alla soluzione di equazioni differenziali alle derivate parziali agli algoritmi per moltiplicare numeri interi di grandi dimensioni.

Sia $x_0, ..., x_{N-1}$ una *n*-pla di numeri complessi. La DFT è definita dalla formula:

$$X_q = F_d(x_n) = \sum_{k=0}^{N-1} x_k e^{-j\frac{2\pi}{N}kq} \qquad [3.9]$$

per $q = 0, 1, ..., N-1$

Calcolare direttamente questa somma richiede una quantità di operazioni aritmetiche $O(N^2)$. Un algoritmo FFT ottiene lo stesso risultato con un numero di operazioni O(nlog(n)). In generale questi algoritmi si basano sulla fattorizzazione di N, ma esistono algoritmi FFT per qualunque N, anche per numeri primi.

Poiché l'antitrasformata discreta di Fourier è uguale alla DFT, ma con esponente di segno opposto e 1/N a fattore, qualsiasi algoritmo FFT può essere facilmente invertito.

La parola wavelet, ondina, ha origine nei primi anni ottanta ed è dovuta a Morlet e Grossman che infatti usavano la parola francese ondelette - "piccola onda". Poco più tardi la parola venne convertita in inglese traducendo "onde" ("onda" in francese) in "wave" - ottenendo wavelet. Le trasformate wavelet sono classificate a livello generale nella trasformata wavelet discreta (Discrete Wavelet Transform, DWT) e nella trasformata wavelet continua (Continuous Wavelet Transform, CWT). La differenza di principio fra le due è il fatto che la trasformata continua opera su tutte le possibili scale e traslazioni, mentre la trasformata discreta usa un sottoinsieme discreto di tutti i valori possibili.

Proposte più recenti includono l'uso di wavelet (Huhtala, Karkkainen e Toivonen, 1999), approssimazioni lineari a pezzi (Keogh, Chakrabarti, Pazzani e Mehrotra, 2000) o l'estrazione di varie caratteristiche delle serie (Chang-Shing Perng, Wang, Zhang e Stott Parker, 2000).

Le wavelet sono definite dalla funzione wavelet $\psi(t)$ (cioè la wavelet madre) e dalla funzione di scalamento $\varphi(t)$ (detta anche wavelet padre) nel dominio del tempo.

La funzione wavelet è in effetti un filtro passa-banda e il suo scalamento ad ogni livello dimezza la sua banda. Questo crea il problema che, per coprire tutto lo spettro, sono richiesti un numero infinito di livelli. La funzione di scalamento filtra il livello più basso della trasformata e assicura che tutto lo spettro sia coperto.

Per una wavelet a supporto compatto, $\varphi(t)$ può essere considerata di lunghezza *finita* ed è equivalente al filtro di scalamento g.

Gli ambiti applicativi di questi due approcci sono fra i più disparati.

Generalmente, la DWT è usata nella codifica dei segnali mentre la CWT è usata nell'analisi dei segnali. Di conseguenza, la DWT è usata comunemente in ingegneria e informatica e la CWT è usata più spesso nella ricerca scientifica. Le trasformate wavelet sono oggi adottate in un gran numero di

applicazioni, spesso sostituendo la trasformata di Fourier convenzionale. Molte aree della fisica hanno visto questo cambiamento di paradigma, incluse dinamica molecolare, astrofisica, geofisica sismica, ottica, turbolenza e meccanica quantistica. Altre aree che stanno vedendo questo cambiamento sono elaborazione delle immagini, pressione del sangue, battito del cuore e analisi dell'ECG, analisi del DNA, analisi delle proteine, climatologia, elaborazione dei segnali in generale, riconoscimento vocale, computer graphics, e analisi multifrattale.

Una delle applicazioni delle wavelet è la compressione di dati. Come molte altre trasformate, la trasformata wavelet può essere usata per trasformare dati grezzi (come le immagini) per poi codificare i dati trasformati, ottenendo un'effettiva compressione. Il formato JPEG2000, ad esempio, è uno standard di immagini che usa le wavelet. Per i dettagli vedi compressione wavelet.

La trasformata wavelet è spesso paragonata alla trasformata di Fourier, dove i segnali sono rappresentati come somma di sinusoidi. La differenza principale è che le wavelet sono localizzate sia nel tempo che nella frequenza mentre la trasformata di Fourier standard è localizzata solo in frequenza. La trasformata di Fourier a tempo breve (STFT) è localizzata in tempo e in frequenza, ma ci sono dei problemi di risoluzione e le wavelet spesso offrono una migliore rappresentazione del segnale grazie all'uso dell'analisi multirisoluzione. La trasformata wavelet, inoltre, è anche meno complessa computazionalmente, richiedendo un tempo O(N) al contrario del tempo O(Nlog$N$) richiesto dalla trasformata di Fourier veloce (N indica la dimensione dei dati).

Esistono, poi, tutta una serie di "Trasformazioni" particolarmente importanti per la finanza e l'economia, che portano dallo spazio delle serie allo spazio delle loro distribuzioni di probabilità. In esse le distanze precedentemente definite non dipendono direttamente dalla distribuzione statistica dei punti delle serie. Naturalmente, tutte queste distanze possono essere interpretate statisticamente, ricavando appropriate statistiche, eventualmente con metodi

di simulazione. La loro motivazione, tuttavia, non è statistica ma è legata a vari concetti di somiglianza fra le forme delle serie, totali o parziali.

Se si lascia cadere la condizione di triangolarità e si fanno alcune ipotesi statistiche sulle serie, si possono adottare altri tipi di distanza che si sono rivelate vantaggiose in molte applicazioni.

In particolare, (Kakizawa, Shumway e Taniguchi, 1998), sono stati proposti concetti di similarità basati sulla teoria dell'informazione.

E' stato proposto, poi, di utilizzare l'entropia di Kullback-Leibler, definita come:

$$I(a,b) = E\left[\log\frac{p(x)}{q(x)}\right] \qquad [3.10]$$

o la misura di informazione di Chernoff definita come:

$$B_\alpha = \log E\left[\left(\frac{q(x)}{q(x)}\right)^\alpha\right] \qquad [3.11]$$

per definire la quasi-distanza fra due serie come:

$$J(a,b) = I(a,b) + I(b,a) \qquad [3.12]$$

oppure:

$$J(a,b) = B_\alpha(a,b) + B_\alpha(b,a) \qquad [3.13]$$

rispettivamente.

Sempre nell'ambito di distanze definite in termini di concetti probabilistici, sono state proposte distanze basate sui modelli di Markov nascosti (Hidden Markov Models - *HMM*). In questo caso, si cerca una rappresentazione delle

serie in termini di catene di Markov con un numero di stati che deve essere determinato con criteri statistici (Xianping Ge e Padraic Smyth, 2000).

3.4 Distanza, correlazione e cointegrazione

Il problema di stabilire relazioni di somiglianza fra serie non è nuovo in statistica. Infatti, il concetto classico di correlazione e di regressione è stato introdotto per caratterizzare processi che hanno un andamento simile.
Dato un insieme di serie storiche, la matrice di correlazione o di varianza-covarianza misura, infatti, il grado di similarità fra le varie serie. Si noti, tuttavia, che i coefficienti di correlazione non costituiscono una distanza perché possono assumere valori compresi fra -1 e +1.
Diverse motivazioni hanno spinto ad andare oltre il concetto di correlazione.
Una prima motivazione è l'applicabilità del concetto di correlazione. Infatti, si possono trovare similitudini fra *pattern* o spezzoni di serie, che sarebbero statisticamente non correlati. Un esempio potrebbe essere quello della ricerca di *pattern* di comportamento creditizio che conducono al fallimento. In questo caso, si possono trovare percorsi simili verso il fallimento anche tra aziende lontane o nel tempo o nello spazio, anche se non vi è alcuna correlazione diretta tra tali percorsi. I concetti di similarità si applicano perciò a contesti più generali della correlazione.
Una seconda, forte, motivazione che ha indotto a superare il concetto di correlazione è la fondamentale instabilità che può manifestarsi nelle relazioni di correlazione.
Se si considera un insieme abbastanza ampio di titoli, ad esempio i titoli appartenenti all'indice Standard & Poor 500 (S&P 500), si trova che la matrice di varianza-covarianza fra questi titoli non è stabile, a meno di valutare tale matrice su periodi estremamente lunghi. E' stato osservato empiricamente (Laloux, Cizeau, Bouchaud e Potters, 1999, Plerou, Gopikrishnan, Rosenow, Amaral e Stanley, 1999) che se il numero dei titoli è

dello stesso ordine di grandezza del numero dei punti su cui si valutano le correlazioni, la matrice delle correlazioni non è stabile, ma presenta fluttuazioni casuali.

Considerare periodi molto lunghi per valutare le correlazioni non è realistico dal punto di vista economico perché si valuterebbero correlazioni fra entità aziendali che nel frattempo hanno, in genere, subito importanti cambiamenti, ma potrebbe esserlo per altri domini applicativi. Limitando le finestre temporali a periodi dell'ordine di 1000 giorni, la matrice di correlazione su grandi aggregati è, comunque, instabile.

La rumorosità delle matrici di varianza-covarianza è solo un aspetto del problema della stabilità delle correlazioni. Infatti, il problema della debolezza intrinseca delle correlazioni era noto in statistica da parecchio tempo, anche prima dell'osservazione che le matrici di varianza-covarianza sono molto rumorose.

La possibilità di osservare correlazioni spurie era stata osservata da Yule (1926) e Granger e Newbold (1974), mostrando che è possibile trovare correlazioni spurie in fenomeni perfettamente non correlati, anche se i campioni osservati sono molto grandi. Infatti, vi sono casi in cui i metodi correnti di determinazione delle correlazioni convergono anche se le serie sono indipendenti. Le correlazioni trovate sono completamente spurie.

Questa osservazione è una delle motivazioni che hanno spinto all'introduzione delle tecniche di cointegrazione.

Due o più processi sono cointegrati se esiste una loro combinazione lineare che sia stazionaria. Processi cointegrati rimangono pertanto vicini, a meno di trasformazioni lineari. Le loro correlazioni sono stabili.

Dato un insieme di serie storiche empiriche, è possibile eseguire test di cointegrazione che servano a stabilire se è ragionevole considerare le serie generate da processi cointegrati, oltre che a determinare i processi stessi. Ad esempio, dato un insieme di serie storiche si possono stimare modelli a correzione d'errore (ECM) che producono processi cointegrati.

Non è agevole, tuttavia, adottare gli stessi metodi per analizzare grandi insiemi di serie empiriche in cui solo alcuni cluster sono eventualmente cointegrati. Per tornare ad un esempio finanziario, è evidente che su un grande aggregato come l'S&P 500 non tutti i titoli saranno cointegrati, anche se è possibile ipotizzare che esistano cluster di titoli fortemente correlati o anche cointegrati. Per risolvere questi inconvenienti sono state proposte varie strategie che conducono tutte a segmentazioni del mercato in segmenti internamente correlati.

Alcuni ricercatori tra cui Laloux, Cizeau, Bouchaud e Potters (1999) e Ormerod e Mounfield (2000) hanno proposto di utilizzare la teoria delle matrici casuali, una teoria matematica sviluppata nell'ambito della meccanica quantistica negli anni '50.

Se si considera la matrice di correlazione di serie storiche completamente indipendenti e casuali, si trova che gli autovalori di tale matrice hanno una distribuzione caratteristica che può essere calcolata teoricamente. E' stato perciò proposto di confrontare tale distribuzione teorica con la distribuzione degli autovalori di serie empiriche, per discriminare il rumore dall'informazione nella matrice di correlazione. Risultati sperimentali mostrano che la distribuzione empirica degli autovalori della matrice di correlazione del S&P 500 è sorprendentemente vicina a quella di una matrice casuale.

Esistono, tuttavia, importanti deviazioni che segnalano l'esistenza di correlazioni vere e stabili al di sopra del rumore. Tali risultati possono essere interpretati con la presenza di cluster di titoli fortemente correlati al loro interno mentre il rumore maschera le eventuali correlazioni fra altri titoli. L'analisi basata sulle matrici casuali mostra perciò che esiste una struttura di correlazione su insiemi del tipo S&P 500, distribuita su cluster correlati (Ormerod e Mounfield, 2000). E' naturale, allora, porrsi il problema di ricercare direttamente i cluster di titoli solidamente correlati. Come osservato in precedenza, i coefficienti di correlazione non costituiscono una distanza, in

quanto possono assumere valori negativi. Al fine della ricerca dei cluster è necessario introdurre una misura quantitativa della similarità fra serie che sia, appunto, una distanza. Data una distanza, possono essere applicate procedure di clustering che conducono a determinare cluster di elementi vicini. Si noti, quindi, che correlazione, cointegrazione e similarità sono nozioni che possono muoversi in varie direzioni.

Similarità è un concetto più generale di correlazione. Può essere applicato a processi che abbiano una somiglianza strutturale, opportunamente definita, ma che non sono necessariamente correlati. Ad esempio, i prezzi di due azioni diverse possono seguire percorsi strutturalmente simili in momenti diversi e su intervalli temporali diversi, senza essere necessariamente correlati.

Tuttavia, sotto opportune condizioni aggiuntive, si possono trovare cluster di processi simili che siano stabilmente correlati ed eventualmente cointegrati. Si noti che la matrice di similarità (o distanza) non è necessariamente più stabile della matrice di varianza-covarianza (o correlazione). La similarità, tuttavia, può essere definita in modo sufficientemente stringente per cui cluster di elementi simili mostrano correlazioni stabili.

3.5 Metodi di riduzione dimensionale

Come si è visto in più riprese, moltissimi algoritmi sono stati sviluppati per gestire database di dimensioni continuamente crescenti (come quelli presenti nelle datawarehouse delle grandi imprese). Tuttavia, poco si è potuto fare per creare algoritmi che resistano alla cosiddetta "curse of dimensionality", ovvero la barriera ineliminabile contro cui si infrangono i tentativi di fare clustering su dati ad alta dimensionalità, come quelli di cui si sta trattando.

Esiste una classe di algoritmi sviluppati appositamente per la riduzione dimensionale dei dati, ovverossia per cercare di proiettare uno spazio a N dimensioni in uno spazio a D dimensioni, con $D<N$, in modo tale che vengano preservate le caratteristiche di quei dati che interessano: in generale, parlando

di dati su cui si operino algoritmi basati sulla distanza euclidea, si vorrebbe preservare la distanza relativa, in modo che punti "lontani" nello spazio *N* dimensionale siano "lontani" anche dopo la proiezione.

Gli algoritmi principali sono:

- *Feature Selection***:** esistono una varietà di algoritmi di "feature selection", che scartano parte delle dimensioni dei dati ritenendole poco significative. Un esempio banale potrebbero essere dei dati che nello spazio a 3 dimensioni siano tutti disposti sul piano (*x,y*): in tal caso evidentemente la coordinata *z* potrebbe essere soppressa conservando tutte le informazioni;
- *Latent Semantic Indexing* **-** *LSI***:** con questo algoritmo le parole che compaiono frequentemente negli stessi documenti, in cui ne compaiono delle altre, vengono ritenute "simili" a queste. L'algoritmo crea una matrice di corrispondenza termine-documento (quindi uno spazio a moltissime dimensioni), pre-processandolo per eliminare desinenze, parole non significative (stop-word) e verbi comuni. Inoltre, l'algoritmo pesa nella matrice i termini inversamente alla loro diffusione totale, e suddivide la matrice mediante una Decomposizione per Valori Singolari (SVD), come avviene nella PCA (vedi sotto), ma senza passare per il calcolo della matrice di covarianza. L'LSI è, pertanto, utile nella compressione dei dati per analisi lessicali – ma non è assolutamente detto che sia utile per il tipo di problema di cui si sta trattando;
- *Principal Component Analysis - PCA:* tecnica molto nota, detta anche *trasformazione di Karhunen-Loève.* L'idea è quella di determinare la trasformazione da uno spazio originale *m*-dimensionale a uno *k*-dimensionale con $k<m$ che minimizza l'errore, ovvero la variazione nei rapporti di distanza tra punti. Si consideri un set di dati con *n* dati ed *m* variabili. L'algoritmo calcola la decomposizione a valori singolari della matrice di covarianza del set di dati *M(m,n)* e ne

estraggono i *k* autovettori principali. Questi autovettori sono le componenti principali dei dati, e sono delle combinazioni lineari delle componenti originarie. I dati vengono ora proiettati su queste direzioni e, pertanto, trasformati in uno spazio *k*-dimensionale. Esistono una serie di criteri per determinare il numero corretto di dimensioni basati sulle proporzioni della varianza o sulle radici caratteristiche della matrice di covarianza; tuttavia, si tratta soltanto di euristiche. L'algoritmo ha una complessità computazionale superiore a $O(m^2)$ e pari a $O(m^2)$ in memoria.

- Anche le *SOM* a volte sono considerate pre-processing, infatti dopo che la rete è stata addestrata, ogni punto viene proiettato sullo spazio bidimensionale dei neuroni. Le *SOM*., però, al contrario di quanto accade per trasformazioni di tipo geometrico come le precedenti, non danno una misura di quanto "buona" sia la trasformazione.
- *Multi Dimensional Scaling - MDS***:** questa tecnica considera le distanze tra coppie di punti e cerca di disporre i punti in uno spazio a dimensione inferiore, minimizzando un valore di stress. La funzione stress aumenta quando punti lontani nello spazio a molte dimensioni si vengono a trovare vicini nella proiezione a dimensioni inferiori. Questo algoritmo è una versione euristica di *PCA* e *LSI*, e viene spesso usato per ottenere dei rapide copie di più dataset, provenienti da file differenti (*snapshot*), su dati multidimensionali, rappresentabili su un piano o nello spazio.

Gli algoritmi di cui si è accennato son propriamente usati in contesti la cui dimensionalità sia altissima, fermo restando che essi risentano di una forte complessità computazionale. Ciononostante, essi vengono usati per "comprimere" le dimensioni sulla base di legami "quasi certi", che rendono alcune di queste dimensioni combinazioni "quasi lineari" di altre. Scartare le dimensioni che contengono pochi dati "non in linea" con queste relazioni predominanti è molto utile nel clustering puro, che interessa la maggior parte dei dati del cluster. Ma lo scopo di questi algoritmi in prima

approssimazione, è duplice: comprimere ciò che si può comprimere, evidenziando, però, la presenza di *outlier*, per quanto possibile.

I dati spuri vengono automaticamente "schiacciati" su quelli sani dalla trasformazione tra lo spazio a *N* dimensioni e quello ridotto a *D* dimensioni, perché le relazioni di distanza tra punti che vengono rispettate sono quelle della "maggioranza". Ciò, insinua il sospetto che un approccio di questo genere elimini la possibilità di fare poi una *outlier detection* interessante

Capitolo 4

Confronto tra algoritmi di clustering per il Temporal Data Mining

4.1 Introduzione

Nel trattare degli algoritmi di clustering, relativi ai domini temporali o spazio-temporali, non si può fare a meno di presentare una, se pur breve, ma densa, disamina sulle generali procedure di clustering. L'idea è quella di aiutare l'utilizzatore di database, in contesti tipicamente temporali, ad operare una scelta appropriata, in merito alle tecniche da utilizzare, partendo dall'identificazione delle principali problematiche di costruzione, in base alla tipologia di dati, su cui effettuare l'analisi, per giungere all'individuazione delle più opportune tecniche di risoluzione delle problematiche ad alta dimensionalità, tipiche dei contesti di *mining*.

In generale, ciascuno degli algoritmi di clustering, presenti in letteratura, gode di caratteristiche proprie, ed è applicabile ai più disparati contesti scientifici.

Risulta evidente che, dal momento che ogni fenomeno possa essere ricondotto ad una connotazione temporale, è necessario analizzare in maniera globale e,

pressoché, esaustiva il problema, prima di scegliere l'algoritmo di clustering più adatto.

Ad esempio, un algoritmo *non supervisionato* di apprendimento, riconduce ad un problema di *knowledge discovery* (algoritmi di *data mining*). Si tratta, infatti, di operare una riduzione dimensionale dei dati, facendo ricorso, appunto, alla classificazione *non supervisionata* o *clustering*. L'dea di fondo di tali algoritmi è quella di raggruppare gli oggetti in classi, in modo da minimizzare la distanza *intra-classe* e da massimizzare la distanza *inter-classe*. Tuttavia, un tale approccio dà luogo ad un ventaglio di problematiche sui metodi migliori da utilizzare per effettuare questa suddivisione, nonché, sui criteri in base ai quali misurare la similarità o la dissimilarità tra elementi, oggetto del clustering, criteri questi, che influiscono, senza dubbio sull'efficienza dell'algoritmo candidato per l'analisi.

4.2 Problematiche di dimensionalità

Gli algoritmi di clustering vengono usati nell'ambito del data mining come procedura di *pre-processing*, per semplificare la struttura dei dati.

Un requisito chiave che deve avere l'algoritmo da utilizzare è, innanzitutto, quello di consentire di trattare i dati a costi computazionali accettabili (esempio, mediante vettori di input da 1460 elementi). Non si deve credere, comunque, che tale requisito sia, in assoluto, estremo.

In molti dei domini applicativi, trattati nei precedenti capitoli, vi sono dati ad altissima dimensionalità. E' per questo che la maggior parte degli algoritmi di clustering, più semplici, presenti in letteratura, sono stati rivisitati e adattati, in termini implementativi, per poter essere applicati a dati ad alta dimensionalità, cercando di ridurre al minimo l'incombente problematica dell'*efficienza*.

Un'attenta riflessione va fatta, in primo luogo, su quegli algoritmi in grado di trattare grandi dataset. Indubbiamente, più "tempo di rete" l'algoritmo può osservare in addestramento, migliore sarà la qualità delle sue suddivisioni.
In secondo luogo, non si ha idea di "quante classi" possano servire per descrivere i dati. Per questo, si dovrebbe guardare, in veste critica, alle "assunzioni" iniziali, che si effettuano, nell'applicazione degli algoritmi, premesso che, tutti i relativi risultati andrebbero ulteriormente espansi, per cercare di determinare criteri meno "euristici" di quelli impiegati per inizializzare gli algoritmi.
In particolare, vi sono tre criteri per stabilire se la classificazione raggiunta da un algoritmo non supervisionato è soddisfacente:

- *inspection-based*. Tale criterio suggerisce l'esplorazione "manuale" delle suddivisioni create dall'algoritmo;
- *expert-based*. Tale criterio effettua un raffronto con una classificazione fatta da un esperto;
- *task-based*. Tale criterio suggerisce di misurare la performance dell'algoritmo, una volta calato nel suo "task" specifico.

Nell'impossibilità di disporre di una classificazione "da esperto", per uno solo dei due componenti dell'algoritmo, ovviamente, il criterio migliore sarebbe quello *task-based*.
Tuttavia, le ovvie limitazioni imposte da un'accurata analisi, richiederebbero di effettuare una valutazione ispettiva delle classi create dagli algoritmi.
L'algoritmo deve, altresì, rappresentare, in modo, possibilmente, intuitivo e semplice da utilizzare, i criteri di suddivisione che ha utilizzato. Sarebbe comodo poter esplorare tali criteri per avere una idea "umanamente significativa" della suddivisione da operare. Inoltre, sarebbe anche, estremamente, interessante che l'algoritmo prevedesse, tra i suoi compiti, l'individuazione degli *outlier*[1], ovvero di quei dati che ricadono di molto al di

[1] Un outlier è una osservazione che devia talmente tanto rispetto alle altre, tanto da far sospettare che sia stata generata da un meccanismo completamente diverso.

fuori dei cluster individuati. Si potrebbe, ad esempio, definire un *outlier* come un dato che dista da un cluster più di quanto distino, mediamente, il 99% degli altri dati osservati. Questo, però, non rende giustizia della presenza di *outlier* "locali", che sono più distanti, nella loro regione di spazio, da un cluster, rispetto a tutti gli altri punti della stessa regione.

Questo, perché, specialmente in domini in cui le metriche non sono così chiaramente definite, ci potrebbero essere grosse variazioni nella distribuzione locale dei punti. Inoltre, con l'aumentare della dimensionalità del problema alcune di queste metriche, basate sulla distanza, perdono il proprio valore, a causa dell'estrema dispersione dei punti.

In letteratura, sono stati proposti metodi specifici per l'individuazione di *outlier* in questi contesti. È interessante notare che alcuni algoritmi già prevedono l'individuazione di *outlier* come una caratteristica binaria (un dato è un *outlier* oppure non lo è), mentre altri metodi possono essere adattati, per esempio, dando una misura della distanza tra l'ingresso, appena classificato, e i suoi compagni di cluster.

In generale, si possono di*STING*uere insiemi di test "statistici", basati sulle distribuzioni, e test basati sulla "distanza". Per ciascuno di questi metodi, andrebbe sottolineato, se una valutazione degli *outlier* è già prevista o può essere facilmente realizzata.

Sarebbe, inoltre, interessante scegliere un algoritmo che dia la possibilità di essere "riaddestrato" o "tarato", man mano che si analizzino i dati. Inoltre, molti algoritmi, che per loro natura sono incrementali, possono anche essere, in qualche modo, *order-dependent*, il che potrebbe ingenerare grossi svantaggi.

4.3 Tipologie e caratteristiche dei dati

Secondo letteratura, tre sono le macrocategorie che fungono da dati in ingresso in un algoritmo di clustering:

- *dati quantitativi*, ossia semplici dati numerici. Ogni vettore rappresenta, quantitativamente, una serie di caratteristiche;
- *dati categorici,* ossia dati sui quali non ha senso eseguire operazioni matematiche (ad esempio, il calcolo della distanza) o predicati, se non l'uguaglianza. La misura di similarità, in questi casi, può essere definita in modo "creativo" o "non convenzionale", come ad esempio,

$$(T_1, T_2) = |T_1 \cap T_2| / |T_1 \cup T_2| ;$$

- *dati metrici*, ossia dati per i quali è definita una funzione distanza, cioè una funzione che rispetta i quattro assiomi delle metriche della [3.3], che associa, a due dati, un numero reale, che rappresenti la distanza fra essi. Ovviamente, i dati quantitativi rientrano in questa categoria che, però, risulta estendibile.

4.4 Proprietà degli algoritmi di clustering

Gli algoritmi di clustering godono delle seguenti proprietà:

- Un algoritmo può essere *gerarchico* o *piatto* (in inglese si usa il termine "partitioning" per gli algoritmi piatti, ma ciò potrebbe essere fuorviante, a causa del troppo simile concetto di "divisivo", collegato agli algoritmi gerarchici). Un algoritmo gerarchico suddivide, per raffinamenti successivi, i dati in classi, sempre meno generali. Un algoritmo piatto suddivide i dati in un insieme di classi, tutte dello stesso "livello". Un algoritmo gerarchico, spesso, produce risultati più

comprensibili, ma è meno efficiente, computazionalmente, di un algoritmo piatto. Inoltre, un algoritmo gerarchico, difficilmente può "correggere" un eventuale errore, commesso inizialmente. Gli algoritmi di clustering gerarchici possono essere *divisivi* oppure *agglomerativi*; possono, cioè, partire considerando tutti i dati come un unico grande insieme e, ricorsivamente, dividerlo, oppure, possono considerare ogni dato come elemento di un "singoletto" e cercare di aggregarli ricorsivamente.

- Un algoritmo di tipo piatto può essere *iterativo* (cioè ripetere l'assegnazione dei dati di *training*, nei vari cluster, per ogni passo dell'addestramento) oppure no.
- Si parla anche di *hard clustering* e di *soft clustering*, ovvero di algoritmi che suddividono i dati in classi in modo netto (ogni dato finisce per appartenere a una data classe), e algoritmi che invece attribuiscono ad ogni dato una certa probabilità di appartenere a una certa classe. Si è già accennato a questa categoria di tecniche nel capitolo 2. Appare necessario a questo punto, soffermarsi sul fatto che alcuni autori preferiscano parlare di *fuzzy clustering* e, quindi, di "gradi di appartenenza" alle varie classi, ma visto che la base di questi algoritmi è tipicamente statistica, ciò può apparire fuorviante.
- Gli algoritmi *disgiuntivi* prevedono che i cluster siano tutti separati; tuttavia, è possibile che l'algoritmo preveda la possibilità, per un dato, di appartenere a più cluster diversi.
- Alcuni algoritmi rappresentano i propri cluster mediante un *albero* di condizioni di divisione (in particolare, gli algoritmi gerarchici e divisivi), mentre altri li rappresentano, mediante una congiunzione di condizioni (algoritmi piatti, di vario tipo); altri ancora, li rappresentano mediante *centroidi*, e ne esistono parecchie varianti; infine, alcuni algoritmi rappresentano i cluster usando più *rappresentanti*. È ovvio, che gli algoritmi ad albero possono

rappresentare una varietà di regioni diverse, che le possano, poi, scoprire, dipende dall'algoritmo, non già dalla loro rappresentazione. Inoltre, una rappresentazione, basata su centroidi, funziona bene per regioni compatte di forma sferica o, comunque convessa.

Rappresentazioni mediante più punti descrivono bene *cluster* di forma generica, anche se, in generale, meglio se convessa.

Dal momento che si è parlato spesso di tecniche basate su problematiche legate alle "distanze", si rende necessario definire le procedure atte a misurare le distanze *inter-cluster* e *intra-cluster*:

Le prime, *Inter-Cluster*, si basano sull'assunzione che la distanza tra due sequenze è direttamente proporzionale al tempo che le separa dalla loro sequenza progenitrice comune, tipico degli algoritmi di "linkage" (Figura 4.1).

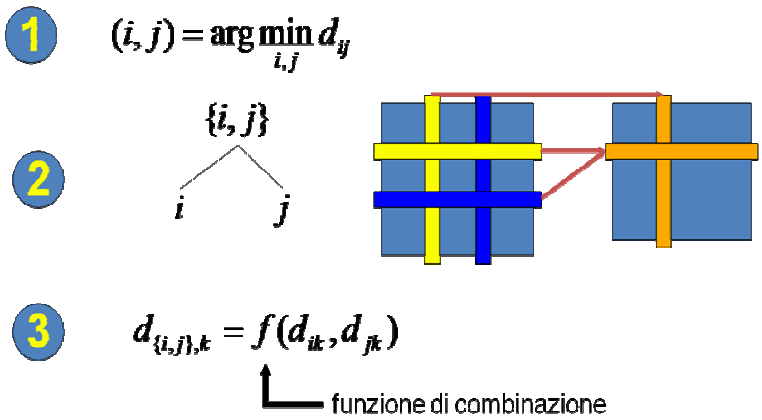


Figura 4.1: Fasi di un Algoritmo di "linkage"

Si pensi, ad esempio, alla distanza genetica tra due sequenze, direttamente proporzionale al tempo che le separa dalla loro sequenza progenitrice comune (Figura 4.2):

$$D = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1N} \\ d_{21} & 0 & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & 0 \end{bmatrix}$$

$$d_{ij} = d_{ji} \geq 0$$

$$d_{ij} \propto T_{ij}$$

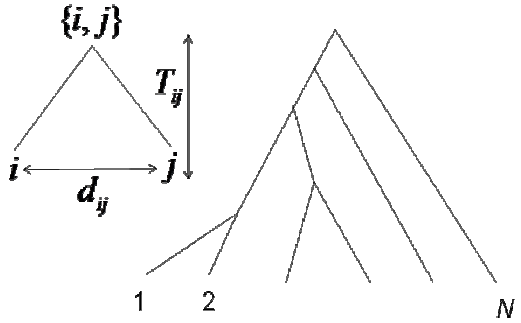


Figura 4.2: Esempio di distanza genetica tra due sequenze che utilizzerà un Algoritmo di "linkare

In particolare, un Algoritmo di "linkage" può essere:

- *Single linkage*, in cui la distanza minima tra i due cluster, ovvero la distanza dei due vettori più vicini, appartenenti, uno ad un cluster e l'altro al secondo.

$$d_{\{i,j\},k} = min\{ d_{ik}, d_{jk} \} \qquad \{i, j, \ldots, k, \ldots\}$$

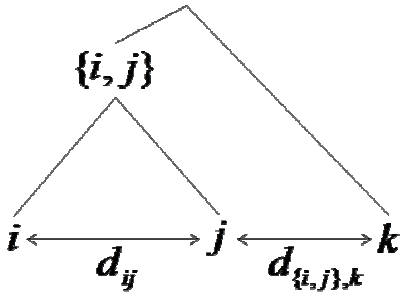


Figura 4.3: Esempio di *Single linkage*

- *Complete linkage*, in cui la distanza tra due cluster è massima,

$$d_{\{i,j\},k} = max\{ d_{ik}, d_{jk} \} \qquad \{i, j, \ldots, k, \ldots\}$$

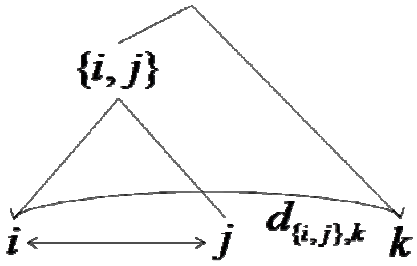


Figura 4.4: Esempio di *Complete linkage*

- *Average linkage*, in cui si effettua la media delle distanze tra ogni coppia di vettori appartenenti il primo e al secondo cluster,

$$d_{\{i,j\},k} = \frac{d_{ik} + d_{jk}}{2} \qquad \{i, j, \ldots, k, \ldots\}$$

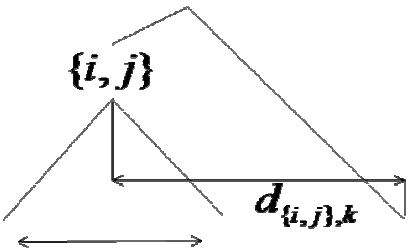
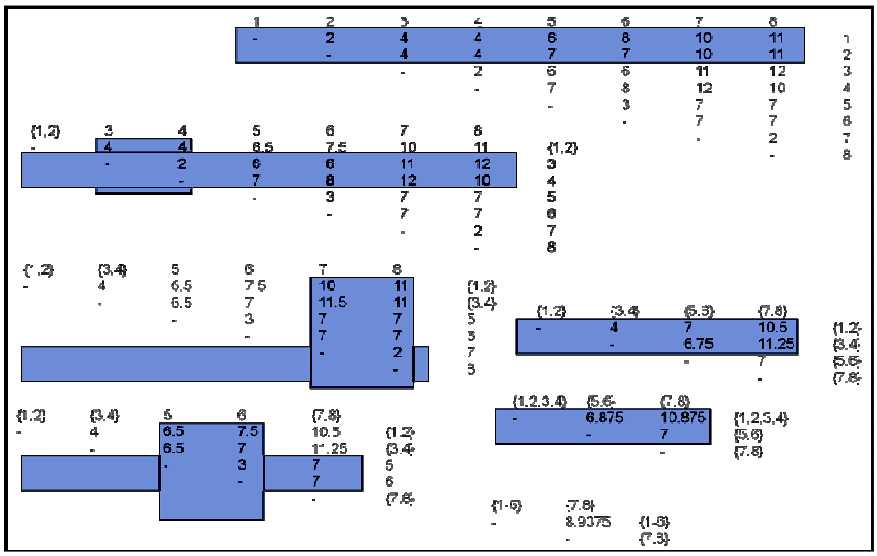


Figura 4.5: Esempio di *Average linkage*

Di seguito si riporta un esempio di relativo alla costruzione di alberi filogenetici, mediante Average linkage, partendo dalla matrice delle distanze:

1	2	3	4	5	6	7	8	
-	2	4	4	6	8	10	11	1
	-	4	4	7	7	10	11	2
		-	2	6	6	11	12	3
			-	7	8	12	10	4
				-	3	7	7	5
					-	7	7	6
						-	2	7
							-	8



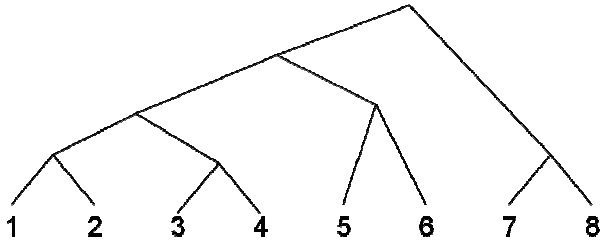
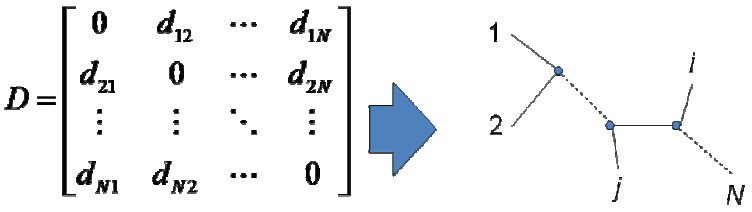


Figura 4.6: Esempio di costruzione di alberi filogenetici *mediante Average linkare*

- *Centroid linkage*, in cui si costruisce la distanza dei centroidi dei due cluster;
- *Centroid "Manhattan" distance*, in cui si costruisce la distanza secondo il metodo Manhattan, ovvero la somma semplice delle distanze lungo le varie dimensioni (invece della radice quadrata della somma quadratica).

Le *distanze Intra-Cluster*, invece, si di*STING*uono in:

- *Distanza media*, in cui si calcola la media delle distanze di ogni coppia di vettori ("diametro");
- *Varianza*, in cui si calcola la media degli scarti quadratici, rispetto al centroide ("raggio");
- *Distanza "nearest neighbor"(NJ)*, in cui si calcola la media delle distanze minime, ovvero, per ogni vettore, si considera la distanza dal vettore che gli sta più vicino.



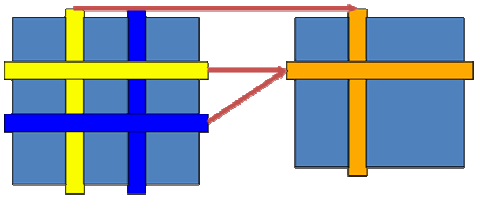
$$D = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1N} \\ d_{21} & 0 & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & 0 \end{bmatrix}$$

$$d_{ij} = d_{ji} \geq 0$$

Figura 4.7: Esempio di *NJ*

Con questo approccio la lunghezza degli archi deve essere "una buona approssimazione" delle distanze. Un esempio di reticolo della matrice delle distanze *NJ*, si può riscontrare di seguito (Figura 4.8)

$$d_{\{i,j\},k} = \frac{d_{ik} + d_{jk}}{2}$$



$$N \times N \quad (N-1) \times (N-1)$$

Figura 4.8: Esempio di reticolo della matrice delle distanze *NJ*

- *Distanza dal centroide*, in cui si calcola la media della distanza dal centroide.

4.5 Test su clustering

Prima di passare in rassegna gli algoritmi di clustering per dati temporali, che saranno presentati nel paragrafo successivo, è bene che ci si soffermi su un tratto caratteristico, comune a ciascuna tipologia di algoritmi di clustering, Si tratta di un aspetto molto delicato, relativo ai test da effettuare una volta, scelta la procedura, nonché dopo aver effettuato la partizione definitiva. Tale procedura consente di effettuare la *caratterizzazione* statistica dei gruppi ottenuti, mediante l'individuazione delle modalità e delle variabili più rilevanti ai fini di un'esauriente descrizione dei singoli gruppi.

La caratterizzazione viene operata in riferimento:

- a statistiche di scarto tra le frequenze delle modalità, all'interno dei gruppi, e le corrispondenti frequenze nell'insieme iniziale di unità statistiche;

- ai valori-test delle modalità.

Una modalità non risulta caratterizzare un dato gruppo se è presente, nel gruppo stesso, con una frequenza relativa che non si discosti, in modo statisticamente significativo, dalla frequenza relativa, assunta nell'insieme complessivo delle unità.

In termini inferenziali, parlare di differenze statisticamente non significative, rispetto a una data modalità, tra la frequenza relativa nel gruppo e quella nella "popolazione" di riferimento, equivale ad ipotizzare che le unità statistiche coinvolte nel gruppo siano estratte casualmente da tale popolazione. Per ogni *k*-esimo gruppo, quindi, l'ipotesi nulla H_0 da controllare è l'ipotesi di estrazione casuale, senza reintroduzione di n_k unità tra le n della popolazione di riferimento:

$$H_0 : \frac{n_{jk}}{n_k} = \frac{n_j}{n} \qquad [4.1]$$

dove NJ_k è la numerosità delle unità che nel gruppo *k*-esimo possiedono la modalità *j*-esima.

L'ipotesi alternativa H_1, viceversa, assume che la proporzione di unità che presentano la *j*-esima modalità è più elevata nel *k*-esimo gruppo, rispetto al collettivo:

$$H_1 : \frac{n_{jk}}{n_k} > \frac{n_j}{n} \qquad [4.2]$$

Più ci si allontana da H_0, diventando verosimile H_1, più la modalità è da considerarsi importante nella caratterizzazione del gruppo.

La numerosità NJ_k viene a corrispondere a una variabile aleatoria N, che sotto l'ipotesi H_0, segue una legge di probabilità ipergeometrica dai parametri noti. Essendo la distribuzione ipergeometrica convergente alla distribuzione binomiale e di conseguenza a quella normale (Paruolo, 1992), i valori-test

sono ricavati dalla normale standardizzata e individuano le probabilità critiche:

$$p_{jk} = Prob\{N \geq n_{jk} | H_0\} \quad [4.3]$$

All'aumentare del valore-test, diminuisce la probabilità critica e diventa sempre più inverosimile l'ipotesi nulla.

4.6 Tassonomia sugli algoritmi clustering

A questo punto, è d'obbligo soffermarsi sulla tassonomia degli algoritmi più noti, richiamando l'attenzione su alcuni principi su cui si fonderanno alcune scelte sperimentali di cui si tratterà nel capitolo successivo. Per una tassonomia più sviluppata si rimanda alla "Review" pubblicata dall'ACM [65].

- *Algoritmi Gerarchici*:
  - Algoritmi divisivi (*Principal Direction Divisive Partitioning*);
  - Algoritmi agglomerativi **(***Hierarchical Agglomerative Clustering*).
- *Algoritmi Non Gerarchici***:**
  - Neural Gas-NG (*NG; Growing NG- GNG*);
  - Self Organizing Map-SOM (*SOM; Recurrent SOM-RSOM; Growing Hierarchical SOM-GHSOM, Dissimilar-SOM-DTWC*);
  - *K-means* clustering-KMC (*KMC; Growing KMC-GKMC; GKMC with Maximum Size*);
  - Algoritmi basati su *K-medoids* (*PAM; CLARA; CLARANS*);
  - Sequential Leader Clustering-*SLC;*
  - Algoritmi basati sulla densità (*DBSCAN; OPTICS; DENCLUE*);
  - CURE;
  - *BIRCH;*

- Algoritmi basati sulla riduzione dimensionale (*CLIQUE; PROCLUS; MAFIA; OptiGrid; O-Cluster*);
- Algoritmi basati su griglie (*STING; WaveCluster*);
- Algoritmi basati su grafi (*Association Rules Hypergraph Partitioning - ARHP; Minimal Spanning Tree Clustering – MST*);
- AutoClass (*Bayesian Classification; EM Clustering*);
- Algoritmi Genetici;
- ART.

4.7 Analisi critica degli algoritmi di clustering

Di tale tassonomia, si tenga presente che per entrare nello specifico dei domini interessati al presente lavoro di tesi, sarebbe necessario tralasciare quei metodi che trattano esclusivamente dati categorici, in modo da prendere in considerazione esclusivamente quelli che possono essere ricondotti a contesti di domini temporali o spazio-temporali, ma questo sarà tenuto in considerazione nella parte dedicata alla sperimentazione nel successivo capitolo.

4.7.1 Hierarchical Agglomerative Clustering

Si tratta di un algoritmo di *hard clustering* di tipo gerarchico e agglomerativo, basato sul seguente principio: dapprima, ogni dato viene considerato l'unico elemento di una classe a se stante; questi elementi saranno le foglie dell'albero gerarchico di suddivisione. In seguito, cluster che risultano essere "vicini", vengono "fusi", unendoli in un nodo dell'albero gerarchico. La misura della "distanza" tra i cluster viene stabilita secondo una delle quattro definizioni di distanza inter-cluster illustrate in precedenza.

Il metodo in questione ha il vantaggio di essere semplice da realizzare e performante, pur avendo, lo svantaggio di essere molto sensibile al "tipo di distanza" prescelto. Si noti, che ognuna delle quattro definizioni di distanza genererà un albero diverso. In una situazione come quella in cui il concetto di "distanza" è già messo in discussione alla radice, questo non risulta essere incoraggiante per l'analista. Notoriamente, infatti, l'utilizzo della distanza "complete linkage" tende ad approssimare bene solo cluster solidamente compatti, mentre il "single linkage" presenta un problema di "chaining", ovvero, una densa catena di punti, tra due cluster, può indurre in errore l'algoritmo e farli condensare in uno solo. Inoltre, il metodo è sensibile alla presenza di *outlier* già nei dati di training.

Problema ancora più importante, appare la difficoltà di utilizzare il metodo per generare un "certo numero" di classi, poiché bisogna stabilire "dove" troncare l'albero.

L'algoritmo non prevede la possibilità di un addestramento *on-line*, né risulta semplice immaginare qualche metodo "euristico" per consentirlo.

Inoltre, estrarre da questo tipo di algoritmo un criterio per la successiva suddivisione di dati, non originariamente presenti nel set di addestramento, è un'operazione possibile, ma che presenta le sue complessità.

```
Input:
   D = {t1, t2, ...., tn}   // Set of elements
   A     // Adjacency matrix showing distance between elements.
Output:
   DE   // Dendrogram represented as a set of ordered triples.
Agglomerative Algorithm:
   d = 0;
   k = n;
   K = {{t1}, ...., {tn}};
   DE = {< d, k, K >}; // Initially dendrogram contains each element in its own cluster.
   repeat
      oldk = k;
      d = d + 1;
      Ad = Vertex adjacency matrix for graph with threshold distance of d;
      < k, K >= NewClusters(Ad, D);
      if oldk ≠ k then
         DE = DE∪ < d, k, K >; // New set of clusters added to dendrogram.
   until k = 1
```

Figura 4.9: Procedura di *Hierarchical Agglomerative Clustering*

4.7.2 Principal Direction Divisive Partitioning

Si tratta di un algoritmo di *hard clustering* ricorsivo di tipo *gerarchico* e *divisivo*. Dapprima, viene considerato un cluster formato da tutti i punti; in esso viene calcolato il centroide. Inoltre, si procede alla decomposizione in valori singolari della matrice degli elementi del cluster (SVD), di cui si prende in considerazione il primo valore e il relativo vettore associato, che è appunto detto *"direzione principale"*. Il centroide viene proiettato sul vettore, e il cluster viene diviso in due parti, con un iperpiano, perpendicolare al vettore, e, passante per la proiezione del centroide. Il cluster originario diventa un nodo padre, con le due metà come figli.

A questo punto, l'algoritmo si ripete ricorsivamente, sul più disperso degli insiemi foglia, fino al raggiungimento di un obiettivo che può essere un numero massimo di classi, oppure una soglia massima di dispersione prefissata. Si tenga presente che la misura della dispersione può essere effettuata in molti modi differenti.

Ulteriore problema, resta quello di stabilire, per il "tipo" di dati, a cui si fa riferimento, una soglia massima di dispersione e a dover prefissare un numero di classi. L'algoritmo, inoltre, produce per ogni "nodo" un criterio di classificazione molto semplice (costituito da una disuguaglianza).
L'interpretabilità del criterio, in spazi a grandi dimensioni, è tuttavia minima. L'algoritmo non si presta ad un addestramento o ad una calibrazione *on-line*, anche se esistono varianti proposte a questo fine.

4.7.3 *K-means* clustering (*KMC*)

K-means è un algoritmo di *hard clustering* di tipo *piatto* ed *iterativo*.
L'algoritmo inizia selezionando a caso un certo numero *k* di centroidi. Per ogni iterazione vengono svolte le seguenti operazioni:

- ogni vettore viene assegnato al centroide più vicino;
- per ogni cluster così formato viene calcolata la media dei punti che lo compongono come nuovo centroide;
- con questi *K* nuovi centroidi viene ripetuto il procedimento.

Il procedimento si ripete fino a quando i centroidi non si spostano più (Figura 4.1).

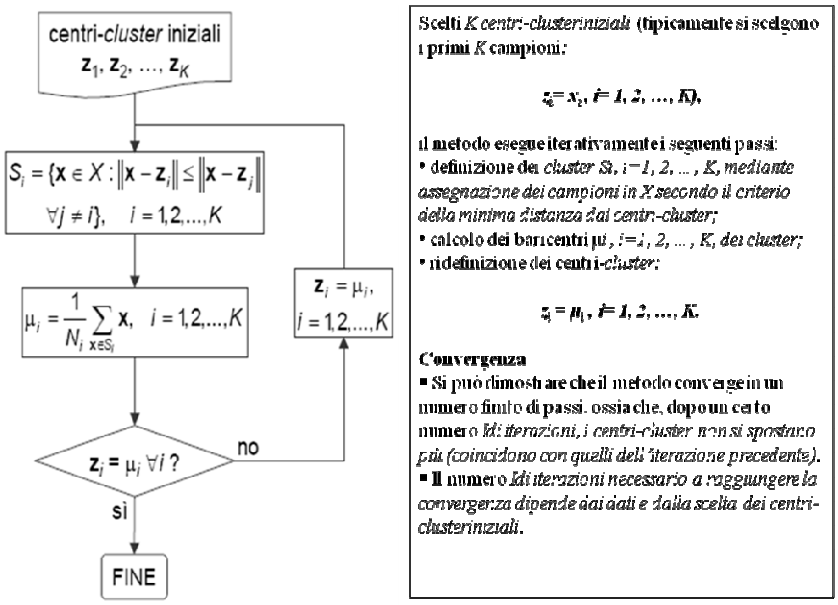


Figura 4.10: Diagramma di flusso di *K-means*

Vi sono un certo numero di presupposti in questo scenario. Innanzitutto, che i dati siano provvisti di una distanza metrica (l'algoritmo non può evidentemente funzionare per dati categorici, anche se alcune estensioni [66] sono state proposte a tale scopo). In secondo luogo, il parametro *k* va imposto a priori; ciò non è sempre facile, soprattutto per quegli algoritmi che operano su dati per i quali non si riesce nemmeno ad immaginare una possibile suddivisione.

Rispetto ad altri algoritmi, che operano con un numero predefinito di classi, *k-means* è più sensibile ad errori nella stima del parametro *k*.

Infine, uno dei grossi problemi dell'algoritmo è l'inizializzazione. Infatti, il procedimento converge, rapidamente, ad un ottimo locale nella distribuzione dei centroidi, ma è ampiamente possibile che inizializzazioni diverse conducano a soluzioni finali molto diverse tra loro. Inoltre, *k-means* non gestisce molto bene gli *outlier*.

Pur non esistendo un metodo vero per l'addestramento *on-line* dell'algoritmo, è possibile immaginare, senza difficoltà, delle soluzioni euristicamente accettabili. Uno dei grandi vantaggi è la relativa leggerezza computazionale dell'algoritmo.

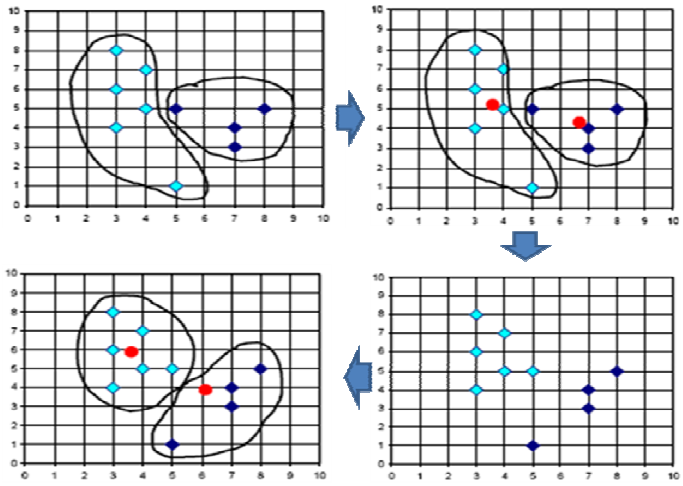


Figura 4.11: Esempio del processo d raggruppamento di *K-means*

Esistono delle varianti da tenere in considerazione:

- *Growing KMC (GKMC):* aggiunge all'algoritmo un elemento tratto dagli algoritmi di crescita. Se la distanza tra il punto in ingresso e il cluster più vicino è superiore di un valore soglia, viene creato un nuovo cluster in cui inserire il nuovo punto; alternativamente, l'algoritmo procede in modo normale. Questo sistema può, sia risolvere il problema della predeterminazione dei *k*, sia, in qualche modo, eliminare o ridurre la variabilità dovuta all'inizializzazione casuale dei centroidi. È, tuttavia, necessario determinare le dimensioni di questa "soglia", in base al problema affrontato.
- *GKMC with Maximum Size:* come per il precedente, con la variante che in ciascun cluster può essere inserito un numero massimo predeterminato di

elementi. In questo modo, invece di avere un singolo cluster, molto ampio, le zone con alta densità di punti saranno caratterizzate da molti cluster ravvicinati tra loro.

- *Global K-means:* questo algoritmo [67] è una versione di *k-means* sviluppata appositamente per contrastare il problema della località e della dipendenza forte dall'inizializzazione. L'algoritmo risolve ricorsivamente il problema di determinare *k* cluster, partendo dalla soluzione del problema con *k-1* cluster e utilizzando *k-means* come metodo di ricerca locale.
- *Scalable K-means:* questo algoritmo è una versione di *k-means* adattata ad ampi insiemi di dati. Viene stabilito un *buffer* di dati. Il database viene "pompato" nel buffer fino a riempirlo. Su questo sottoinsieme di dati viene eseguito l'algoritmo *k-means* normale. Quindi, l'insieme nel buffer viene "compresso", in due fasi:
  - Vengono eliminati i punti "molto vicini" ai centroidi (è difficile immaginare che questi finiscano in un cluster diverso) e le informazioni al loro riguardo vengono "riassunte" associandole al centroide.
  - Sui punti restanti viene eseguito un secondo *K-means*, con molti più centri rispetto al *k-means* principale, e su questi viene eseguito un algoritmo agglomerativo, con una certa soglia di dispersione. In tale modo, vengono individuate "nuvole dense" di punti, che si possono spostare da un cluster all'altro, ma probabilmente si sposteranno tutti assieme. Anche di questi punti vengono riassunte le informazioni fondamentali. I restanti punti vengono mantenuti, e viene caricata una nuova infornata di dati dal database. Questo tipo di algoritmo potrebbe essere usato anche per un aggiornamento di tipo incrementale di una suddivisione basata su *k-means*.

```
Input:
   D = {t1, t2, ..., tn}   // Set of elements
   A     // Adjacency matrix showing distance between elements.
   k     // Number of desired clusters.
Output:
   K     // Set of clusters.
K-Means Algorithm:
   assign initial values for means m1, m2, ..., mk ;
   repeat
      assign each item ti to the cluster which has the closest mean ;
      calculate new mean for each cluster;
   until convergence criteria is met;
```

Figura 4.12: Procedura di *K-means*

L'algoritmo *K-means* è stato applicato, ad un interessante caso, relativo all'ambito delle scienze sociali, riguardante uno studio sul clustering della popolazione disabile, di Rémi Gaudin, et all, dal titolo: "Clustering of Bi-Dimensional and Heterogeneous Time Series:Application to Social Sciences Data.

4.7.4 Self Organizing Maps (Kohonen Maps)

Una Self Organizing Maps - *SOM* [68] è una rete neurale, utilizzata per un apprendimento *non supervisionato*. Si tratta di un algoritmo di *hard clustering* di tipo *piatto* ed *iterativo*. L'algoritmo costruisce un insieme di nodi di dimensione prefissata, collegati tra loro, secondo uno schema deciso dall'implementatore (solitamente, rettangolare o esagonale), che hanno una loro posizione in uno "spazio dei nodi", solitamente bidimensionale (per questo viene chiamata "mappa"). Ogni nodo rappresenta una classe, e corrisponde, pertanto, ad un punto nello spazio *n*-dimensionale dei vettori di ingresso.

L'addestramento avviene, come nel caso dell'algoritmo *k-means*, mediante una ricorsiva approssimazione con i centroidi, e con la differenza che, in questo caso, vengono aggiornati, non solo i centroidi immediatamente più vicini ai vari input, ma anche tutti i centroidi che sono "adiacenti" ad esso nella mappa, con un peso relativo alla distanza (nello spazio dei nodi) dal nodo più prossimo. Questo la rende lievemente più pesante computazionalmente, ma soprattutto, fa sì che l'algoritmo non "converga" a stabilità, in modo rapido ed indolore. Inoltre, una inizializzazione *random* può far sì che anche la *SOM* converga a soluzioni non ottimali.

Questo tipo di mappatura, da uno spazio *n*-dimensionale a uno spazio, solitamente bidimensionale, fa sì che una *SOM* sia al confine tra un algoritmo di clustering e uno di riduzione dimensionale. Grazie ad algoritmi come LabelSOM [69] è anche possibile associare, ai nodi della mappa, dei campioni rappresentativi (etichette), che possono essere usati per esplorare più comodamente le suddivisioni.

Una *SOM* esibisce una discreta raccolta di problematiche da risolvere. Innanzitutto, anche per questo algoritmo va predeterminato il numero di nodi (cluster) necessari, anche se, rispetto a *k-means*, è più tollerante a scelte "eccessive" (grazie ai vincoli nello spazio dei nodi, è possibile che alcuni di essi risultino "vuoti", senza nessun vettore di addestramento associato).

Tuttavia, si paga, una tale flessibilità, con la necessità di precalcolare le epoche di addestramento, o di creare un qualche criterio di stop, più o meno plausibile, in quanto la "convergenza" non è prestabilita, come nel caso di *k-means*. In effetti, si può dimostrare che non esiste una funzione obiettivo per la *SOM*, ovvero, che le regole di addestramento utilizzate non sono il "gradiente" di nessuna funzione. Il criterio di convergenza è, dunque, soggettivo. Alcuni autori documentano che, per insiemi di dati con ampie dimensioni, anche tale convergenza soggettiva della rete potrebbe essere troppo lenta. Per essere più efficiente, a una *SOM,* potrebbe essere premesso un algoritmo di pre-processing, che elimini le dimensioni ridondanti dei dati e

li normalizzi, tuttavia, alcune obiezioni a questa strada sono presentate nel paragrafo sulla riduzione di dimensionalità più avanti.

Vi sono ricerche ed algoritmi che sembrano indicare la fattibilità di un addestramento "*on-line*" delle SOM, tuttavia, la pesantezza dell'algoritmo di addestramento, unita alla sperimentalità delle ricerche, non danno grandissime aspettative in proposito.

Trovando la distanza tra il vettore e il neurone più vicino, questo indica quanto "distante" sia il vettore dal neurone in cui la rete lo ha classificato. In questo modo, sarebbe possibile individuare alcuni *outlier* grazie al fatto che il centroide della classe, in cui la rete lo ha classificato (per mancanza di neuroni), dovrebbe, a logica, risultare "più distante" dall'*outlier* che da tutti gli altri vettori di input. Altri *outlier* possono essere individuati, per esempio, dal fatto che cadono in una classe fino a quel momento vuota.

Anche le *SOM* hanno alcune varianti "evolute" che dobbiamo tenere in considerazione:

- *Recurrent SOM (RSOM)*: Mantiene una memoria dell'output precedente, consentendo di presentare a una SOM sequenze temporali. Per quanto l'algoritmo abbia un suo fascino, proviamo concettualmente ad analizzare ciò che gli verrebbe presentato. I payload presentati alla rete appartengono a diversi flussi di dati, incongruenti: tra di essi non c'è una correlazione temporale non-stocastica.
- *Growing Hierarchical SOM (GHSOM)*: questo tipo di *SOM* cerca di risolvere il problema del dimensionamento della mappa. L'idea di base è quella di consentire alla mappa di crescere in profondità e larghezza: la mappa cresce in larghezza, aggiungendo nuove unità alla rete, durante l'addestramento, nelle aree dove vengono a mancare dei nodi, con un metodo simile a tutti gli algoritmi in crescita. Cresce, inoltre, in profondità, addestrando una nuova *SOM* nelle unità che rappresentano grossi cluster (pertanto, si distacca dal modello di base

dell'algoritmo piatto, assumendo caratteristiche gerarchiche). L'algoritmo consente, creando una struttura flessibile e gerarchica, di evolversi, basandosi sui dati. L'algoritmo è innovativo ed interessante, ma andrebbe estensivamente sperimentato prima di essere utilizzato (esiste ben poca letteratura sperimentale in proposito).

- *Dissimilar-SOM* (*DTWC*). Questo approccio consiste nella rasatura ogni serie da un piecewise lineare o spline cubica I dati di partenza sono una serie di curve

  L'algoritmo di clustering si basa su un adattamento dell'algoritmo SOM di Kohonen, algoritmo di dissomiglianza dei dati (*DSOM*; Golli et al., 2004), introducendo *DTWC* come misura di dissomiglianza.

  Come lo standard *SOM*, il *DSOM* genera mappe dei dati in ingresso su uno spazio. Ogni modello rappresenta un sottoinsieme da localizzare i dati di input

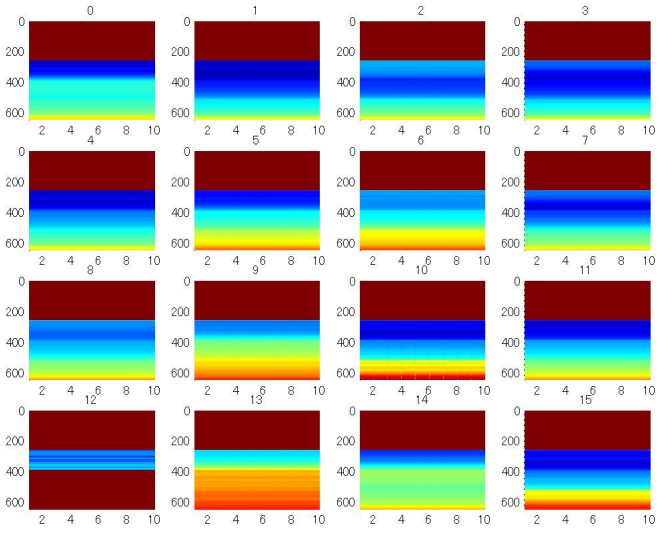


Figura 4.12: Esempio di visualizzazione secondo le *SOM*

Un'interessante applicazione nell'ambito del TDM si riscontra in "A remote sensing data classification method using self-organizingmap" di Hosokawa, M.; Ito, Y.; Hoshi, del 1999, dove si è effettuato il clustering su dati tele rilevati, applicando le SOM.

4.7.5 Sequential Leader Clustering (*SLC*)

L'algoritmo è estremamente semplice: il dato in ingresso viene analizzato, ne viene calcolata la distanza da ognuno dei cluster precedentemente trovati, e viene determinato il cluster più vicino: se la distanza tra il punto in ingresso e il cluster più vicino è inferiore ad una certa soglia, il punto viene associato al cluster, altrimenti il punto diventa il primo componente di un nuovo cluster.
La distanza tra punto e cluster può seguire varie metriche (*nearest neighbor*, *farthest neighbor*, *centroid*), ma l'algoritmo è sostanzialmente semplice e abbastanza rozzo nei risultati.
Questo algoritmo si presta anche in modo molto intuitivo ad un addestramento *on-line*. Si tratta di un algoritmo di *hard clustering* di tipo *piatto* e non *iterativo*.
Bisogna definire la soglia di distanza con attenzione, e l'ordine con cui i dati vengono presentati può causare variazioni enormi nella suddivisione. Questa estrema variabilità è ancora più preoccupante, in domini le cui conoscenze sono, relativamente, poche.

4.7.6 Neural Gas (*NG*)

Questo algoritmo prevede la presenza di un determinato numero di neuroni, che si comportano in modo molto simile a quelli di una *SOM* e ai centroidi di *K-means*.
Durante l'addestramento viene calcolata la distanza di ogni input da ogni cluster, nonché l'adattamento di ogni neurone ad ogni input che viene pesato,

in modo inversamente proporzionale a tale distanza. La distanza diventa sempre più importante, man mano che l'addestramento prosegue. È dunque un algoritmo di *soft clustering* di tipo *piatto* ed *iterativo.*

L'algoritmo condivide pregi e difetti con *K-means* e *SOM*; come in *K-means*, è necessario predeterminare il numero di neuroni; come in *SOM* la convergenza è più lenta (specialmente ad altissime dimensioni), anzi, è ancora più problematica, perché ogni input influenza, potenzialmente, tutti i neuroni.

Bisogna, dunque, fissare un numero di epoche di training o un parametro di stop. L'algoritmo non si presta in modo intuitivo ad un addestramento online.

Esiste, anche per Neural Gas, una variante evoluta nota come *Growing NG (GNG)*. L'algoritmo viene inizializzato con due cluster non connessi. Per ogni dato in ingresso vengono calcolati il primo e il secondo cluster più vicino ed il più vicino. La distanza quadratica del dato dal cluster più vicino viene sommata all'errore locale cumulato del cluster. Tra i due viene inserito un arco di età *0* (o viene azzerata l'età dell'arco esistente). Gli archi uscenti dal cluster vincente, la cui età supera una certa soglia, vengono rimossi. Se ciò produce dei cluster disconnessi da ogni altro cluster, questi vengono eliminati. Si provvede, poi, a ricalcolare i pesi del cluster più vicino, nonché di tutti quelli ad esso connessi. Inoltre, ogni *s* passi, si inserisce una nuova unità.

L'algoritmo in questione, non fa altro che ricercare quel cluster cui corrisponde il massimo errore cumulato, oltre al cluster, ad esso connesso, con i più alti errori cumulati. Successivamente, l'algoritmo rimuove l'arco tra i due cluster, inizializzando un nuovo cluster, interpolato tra questi due, su cui si calcolano, poi, i cluster originari, collegati mediante due archi.

Gli errori cumulati dei due cluster originari si riducono di un coefficiente, e al nuovo cluster si attribuisce un errore cumulato, basato su tali valori.

4.7.7 *K-medoids*

K-means è un algoritmo molto apprezzato, sia per la sua intuitività, sia per il fatto che esso converge con estrema velocità. Tuttavia, non è apprezzabile la

sua dipendenza dall'insieme di centri scelti casualmente, all'inizio del processo.

Gli algoritmi di tipo *K-medoids* sono *piatti* e *iterativi*, basati sull'idea fondamentale di *K-means*, ma con un approccio inverso. Scelti a caso i *K* punti rappresentativi, l'algoritmo *K-medoids* cerca di verificare se, per ogni punto, ne esiste un altro, all'interno del dataset con cui è vantaggioso sostituirlo.

Capostipite degli algoritmi *K-medoids* è *PAM*, Partitioning Around Medoids, che applica, letteralmente, il procedimento delineato sopra. Esso è in grado di generare dei buoni cluster, pur soffrendo di pesanti costi computazionali: $O(k(n-k)^2)$, con n numero di oggetti e k numero di mediane.

Un tentativo di migliorare *PAM* è rappresentato da *CLARA* [70] (Clustering Large Applications), che utilizza sottoinsiemi dei dati di *training*. Il costo computazionale è $O(kS^2+k(n-k))$, con *S,* numerosità dei sottoinsiemi estratti. L'efficienza dipende, però, da tale numerosità, e tentare di ridurre il parametro *S* significa esporsi al rischio di produrre un clustering "falsato", se l'insieme di dati estratti non fosse un campione statisticamente rappresentativo. Tale algoritmo, ben si adatta a dati di tipo spazio-temporale, come si vedrà nel capitolo 5.

CLARANS [71] (Clustering Large Applications based upon RANdomized Search) è il capostipite degli algoritmi di clustering per database di tipo spaziale, ed è un miglioramento di *CLARA*. Si tratta di un algoritmo che applica una metodo di ricerca casuale tra i punti adiacenti ogni *medoid,* per determinare un clustering ottimo localmente; successivamente, passa ad un set differente di mediane per determinare un altro ottimo locale, e così via.

CLARANS ha un costo che è di gran lunga superiore a $O(n)$, e, sebbene dipenda da variabili statistiche, si approssima a $O(n^2)$, richiedendo più di una scansione del dataset. Le sue performance sono state migliorate con l'applicazione di algoritmi *R**.

```
Input:
   D = {t_1, t_2, ..., t_n}   // Set of elements
   A      // Adjacency matrix showing distance between elements.
   k      // Number of desired clusters.
Output:
   K      // Set of clusters.
PAM Algorithm:
   arbitrarily select k medoids from D;
   repeat
      for each t_h not a medoid do
         for each medoid t_i do
            calculate TC_ih;
      find i, h where TC_ih is the smallest;
      if TC_ih < 0 then
         replace medoid t_i with t_h;
   until TC_ih ≥ 0;
   for each t_i ∈ D do
      assign t_i to K_j where dis(t_i, t_j) is the smallest over all medoids;
```

Figura 4.13: Procedura di *PAM*

4.7.8 Algoritmi basati sulla densità

DBSCAN (Density Based Spatial Clustering of Application with Noise) è stato il primo algoritmo ad utilizzare la nozione di *densità*. L'idea base di *DBSCAN* è che un cluster sia una regione di punti ad alta densità, circondata da una regione a bassa densità, che la separa dagli altri cluster. In altre parole, ogni dato, per appartenere ad un cluster, deve avere, entro un certo raggio, almeno un certo numero di altri pattern. In altre parole, la densità dei dati nelle vicinanze del punto considerato deve superare una certa soglia. Tale raggio viene denominato *Eps*, mentre la soglia minima di punti *MinPts*. Il "raggio" può essere stabilito usando una qualsiasi funzione di distanza che rispetti i quattro assiomi della metrica [3.2].

DBSCAN inizia da un punto arbitrario, procedendo, dapprima, calcolando il cosiddetto *Eps-neighborhood* del punto (ovvero, un insieme che contiene tutti i punti del database che stanno in una ipersferetta di raggio *Eps*, centrata nel

punto). Se tali punti sono più di MinPts, questo punto inizia a formare un cluster. La procedura viene ricorsivamente ripetuta per tutti i punti della Eps-neighborhood, e così via.

Successivamente, si determineranno dei dati che appartengono al cluster, ma che non hanno più di MinPts punti nel loro Eps-neighborhood. Tali punti sono i limiti del cluster o "border points", mentre, tutti gli altri sono detti "core point".

La complessità di questo algoritmo è $O(n^2)$; tuttavia, l'utilizzo di una struttura di accesso ad albero, paginato e bilanciato, per dati spaziali, quindi con altezza logaritmica nel numero di n elementi del database, (R*-Tree, MVP-tree, M-tree), può portare la complessità ad $O(n\log(n))$.

Dati sperimentali confermano la subquadraticità del costo, ma solo per dati di dimensionalità bassa. Infatti, in caso di alta dimensionalità, gli alberi perdono in efficienza e il costo torna a circa $O(n^2)$. Altri autori suggeriscono ulteriori miglioramenti di efficienza basati sull'uso della disuguaglianza triangolare. Si tratta di un algoritmo di *hard clustering***,** *piatto* e *non iterativo*.

Il vero problema di *DBSCAN*, tuttavia, come di tutti gli altri algoritmi di questa classe, è la sua sensibilità alla scelta dei parametri Eps e MinPts, per cui, non esistono valide prove euristiche. Inoltre, i parametri sono "globali", non adattandosi, quindi, alla presenza di cluster di varia forma e dispersione.

Tuttavia, l'algoritmo non soffre della tipica vulnerabilità agli *outlier* di cui gli algoritmi trattati sono afflitti. Ciò è possibile per il fatto che, contrariamente al solito, *DBSCAN* non considera tutti i punti del cluster, ma li seleziona mediante la funzione di densità, classificando come "noise" i punti tanto dispersi da non essere "density-connected" a nessun cluster. Esiste una versione incrementale dell'algoritmo. Si tratta di *OPTICS* (Ordering Points To Identify Clustering Structures), che nasce proprio allo scopo di superare i problemi collegati alla scelta dei parametri. Si osservi che, fissato un valore di Eps, diminuendo il valore di MinPts, aumenta il numero di punti che vengono

classificati come "core points". Viceversa, fissato il valore di MinPts, diminuire quello di Eps può causare due effetti:

- aumentare il numero di punti che vengono classificati come "border points";
- far sì che alcuni punti precedentemente appartenenti a un cluster divengano *outlier*.

Per ogni punto processato, *OPTICS* calcola una "core-distance" (il minimo valore di Eps, tale per cui la neighborhood di quel punto contiene esattamente MinPts), e una "reachability-distance" tra due dati (il minimo valore di Eps, tale che i due dati siano "directly-reachable"). L'algoritmo ha la stessa complessità di *DBSCAN*. Tuttavia, mediante considerazioni di tipo diverso è possibile utilizzare *OPTICS* come algoritmo di clustering "a se stante", risolvendo, anche parzialmente, il problema della forte dipendenza dai valori di Eps e MinPts, per cui le considerazioni sulla complessità permangono simili a quelle fatte per *DBSCAN*.

DENCLUE (DENsity CLUstEring), invece, è un metodo di clustering basato sulla densità, caratterizzato da una solida base matematica. Tuttavia, per non complicare eccessivamente l'esposizione, ci si limita a proporlo in termini intuitivi.

Innanzitutto, *DENCLUE* tratta esclusivamente vettori numerici nello spazio a D dimensioni, potendo, quindi, significativamente, parlare di "punti" al posto di "dati".

L'idea di base di *DENCLUE* consiste nell'immaginare che i punti si "influenzino" a vicenda e, ovviamente, che tale influenza sia tanto maggiore, quanto minore è la distanza che li separa; più specificatamente, si formula una "funzione di influenza" che indica con precisione l'influenza di un punto su di un altro.

In pratica, se la funzione che descrive l'influenza del punto y sul punto x è $f(x,y)$, ossia, in generale, la funzione della distanza tra x e y, la "funzione influenza" del punto sarà:

$$\overline{y} = f^{\overline{y}}(x) = f(x, \overline{y}) \qquad [4.4]$$

Intuitivamente, in un cluster denso di punti, la somma dell'influenza sarà maggiore che in altre regioni dello spazio. Si definisce, per ogni punto, la funzione densità, come somma delle influenze di tutti gli altri punti in quel punto:

$$f^D(x) = \sum_{i=1}^{N} f^{y_i}(x) \qquad [4.5]$$

con N = numero di punti.

È evidente la dipendenza dell'algoritmo dal tipo di "funzione di influenza". Alternativa, presenti in letteratura, sono riconducibili ad una "funzione a gradino":

$$f(x, y) = 1 \text{ , se } |x - y \le \sigma| \text{ , } f^D(x, y) = 0 \qquad [4.6]$$

Altrimenti, ad una funzione gaussiana:

$$f(x, y) = e^{\frac{|x-y|^2}{2\sigma^2}} \qquad [4.7]$$

pur, lasciando alla ricerca scientifica il compito di sperimentarne altre.

Si noti che il parametro σ, detto "density factor" governa l'estensione dell'influenza. Per trovare i cluster, *DENCLUE* inserisce la nozione di "density attractor", ovverossia dei massimi locali della funzione densità, e di "density attracted, ovvero la ricerca di una catena di punti da esso a un massimo locale della funzione, tali che per ogni punto della catena il "gradiente" della funzione densità è rivolto verso il punto successivo. Il gradiente viene definito da *DENCLUE* come:

$$\nabla f^{D}(x) = \sum_{i=1}^{N} (x_i - x) f^{x_i}(x) \qquad [4.8]$$

Il cluster "centrato" nel *density attractor* $\bar{x}$ viene definito come l'insieme dei punti x "attratti" da $\bar{x}$ e tali che $f^{x}(x)>\varepsilon$. Si può definire la "arbitrary shake clusters" come la funzione che unisce cluster di più centri, posto che tra i centri esista un "percorso" di punti, con densità superiore a ε .

Si noti, ad esempio, che se si utilizza una funzione a gradino, *DENCLUE* diventa identico a *DBSCAN*, con ε=*MinPts*,σ=*Eps*. Anche in questo caso, quindi, i parametri vanno stimati ad occhio, non esistendo delle euristiche.

L'algoritmo effettivamente usato utilizza una suddivisione dello spazio, mutuata dagli algoritmi "grid-based", suddividendo tutto in ipercubetti di lato pari a 2σ. La complessità totale risulta, in tal modo, ridotta a *O(nlog(n))*.

In generale, mancano in letteratura, per tutti questi algoritmi, prove applicative su problemi reali con alta dimensionalità dei dati.

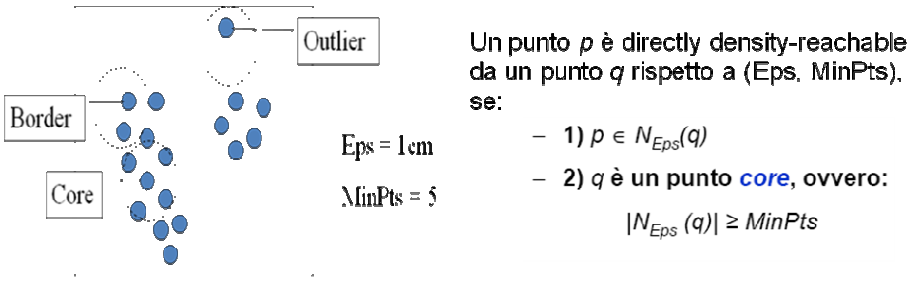


Figura 4.14: Procedura di *DBSCAN*

4.7.9 *BIRCH*: Balanced Iterative Reducing and Clustering using Hierarchies

BIRCH [72] è un algoritmo di *hard clustering* di tipo *gerarchico* e *divisivo* per la suddivisione di grandi quantità di dati, sviluppato per ottimizzare l'utilizzo della memoria e l'ammontare delle operazioni di I/O. L'osservazione su cui si

basa *BIRCH* è che lo spazio dei dati non è occupato in modo uniforme, e, pertanto, non tutti i dati sono ugualmente rilevanti.

BIRCH gestisce solo dati metrici, e tratta cluster sferici. Per ogni cluster, si mantiene in memoria una "clustering feature", che è una tripla (*N,LS,SS*) dove *N* è il numero di punti nel cluster, *LS* è la somma lineare di tutti i componenti del cluster, e *SS* è la somma quadratica. Si può dimostrare che mantenere l'elenco completo di tali *clustering features* è più che sufficiente per calcolare tutte le distanze *inter-cluster* che si sono definite finora, oltre alla misura del raggio e del diametro di ogni cluster. In aggiunta, se CF1 (N1,LS1,SS1) e CF2 (N2,LS2,SS2) sono due cluster disgiunti, la CF del cluster ottenuto dalla loro unione è ovviamente data da (N1+N2,LS1+LS2,SS1+SS2).

L'algoritmo costruisce un albero CF-tree, cioè un albero bilanciato in altezza. Vi sono due parametri, il *branching factor*, *B* e il *threshold*, *T*. In pratica, l'albero è costituito da nodi, ciascuno dei quali contiene al massimo *B* puntatori verso dei figli. Ad ogni puntatore (cioè, ad ogni figlio) si associata il relativo CF. Sulle foglie dell'albero si trovano una serie di CF di singoli sub-cluster. Con *T* si indica, appunto, la massima soglia di raggio o diametro di un cluster ammesso. Al crescere di *T,* l'algoritmo crea cluster sempre più ampi. L'impostazione rende evidente come *BIRCH* lavori al meglio su cluster di dimensioni e dispersione paragonabili.

La costruzione dell'albero avviene leggendo, ad uno ad uno, i record da suddividere, e piazzandoli, automaticamente, ciascuno in un sub-cluster appropriato.

BIRCH è un algoritmo locale, ogni scelta di clustering può essere presa basandosi sul singolo input in esame. In tal modo, il costo di I/O cresce linearmente con la dimensione, dando luogo a partizionamenti, pressoché corretti. Raggiunta una foglia, si scelgono i sub-cluster più prossimi al dato. Se aggiungendo il dato non si viola il vincolo di *T*, la procedura si arresta, e si può aggiornare il CF. In caso contrario, si aggiunge il cluster, dopodiché si crea una nuova foglia, separando, dagli altri, il cluster "più lontano",

ponendolo in una foglia a parte. In tal caso, si deve, ovviamente, aggiornare il nodo padre, e così via ricorsivamente, ricordandosi che nei nodi c'è anche il limite imposto dal *branching factor, B*.

Un possibile miglioramento dell'algoritmo, in caso di split, suggerisce che nel nodo dove lo splitting si ferma, ovvero il primo nodo che ha spazio per accogliere un nuovo ramo, si scelgano le due CF più vicine dove si fondano i relativi rami, dopodiché li si risuddivide, col criterio dei "più lontani", come prima, se si supera il *branching factor*. In ogni caso, i CF di tutti i nodi al di sopra della foglia vanno aggiornati: se non c'è uno split, semplicemente aggiungendo al CF padre, i valori del nuovo input; altrimenti si dovranno effettuare alcuni calcoli intuitivi per gli spostamenti. Inoltre, il valore di *T* può essere aumentato allo scopo di ridurre le dimensioni dell'albero.

In letteratura, è presente un apposito algoritmo di *re-scanning* che, a partire dal CF-tree, con un dato *T*, può costruire il CFtree con un *T* più elevato.

Tuttavia, vi sono altri accorgimenti, relativi al riordinamento delle foglie e al trattamento corretto di più copie di dati uguali, che si tralasciano in questa sede, ma che possono rappresentare utili spunti di ricerca. La letteratura, comunque, presenta una collezione di algoritmi di cluster, che possono essere implementati sulla base di clustering tradizionali, considerando, ciascuno dei centroidi dei cluster, pesato col numero di punti presenti, come una "rappresentazione concentrata" dei dati.

I costi dell'algoritmo *BIRCH* sono O(*dnBh*) per la costruzione dell'albero, e O(*rdLBh*) per il reinserimento di una "leaf entry", dove *d* è la dimensione dei dati; *n*, il numero di dati; *B* e *L* sono, rispettivamente, il numero di figli di un nodo e il numero di cluster nelle foglie; *h*, l'altezza dell'albero dal nodo a una foglia e *r*, il numero degli spostamenti.

BIRCH è stato modificato per essere implementato in degli spazi metrici generici, non necessariamente vettoriali, dando vita agli algoritmi BUBBLE e BUBBLE-FM [73], che sono tra i pochi applicabili a tali spazi.

4.7.10 *CURE*: Clustering Using Representatives

CURE [74] è un algoritmo di *hard clustering* di tipo *gerarchico* ed *agglomerativo* che cerca di superare almeno due delle limitazioni tipiche degli algoritmi tradizionali, ovvero i problemi derivanti dalla presenza di *outlier* nei dati di addestramento, e la necessità di identificare cluster di forma non sferica, senza utilizzare come descrizione tutti i punti del cluster (il che renderebbe l'algoritmo non scalabile).

CURE descrive ogni cluster con un insieme di n punti (con n, parametro scelto) generati scegliendo a campione quei punti del cluster che siano "ben dispersi". A ogni passo dell'algoritmo tali punti vengono "spostati" verso il centroide del cluster di una certa frazione α, della loro distanza da esso. In questo modo eventuali "*outlier*", presi erroneamente come punti descrittori, vengono rapidamente riportati in linea. Il coefficiente α può variare da *0* a *1*, con *0* l'algoritmo diventa sostanzialmente equivalente ad un algoritmo che descriva il cluster con tutti i punti, con *1*, diventa, sostanzialmente, un algoritmo basato su centroidi.

La complessità spaziale dell'algoritmo è $O(n)$ e quella temporale $O(n^2\log(n))$, che può ridursi a $O(n^2)$ se i dati sono di bassa dimensionalità (non è sicuramente il nostro caso). Nonostante alcune affermazioni degli autori dell'algoritmo, il fatto che la curva di crescita del tempo di elaborazione con la dimensionalità del data-set sia stata omessa dall'articolo in cui viene presentato l'algoritmo, e che i ragionamenti parlino di "high-dimensional dataset" riferendosi a un set a 40 dimensioni, fanno presupporre che l'aumento di dimensionalità richieda un numero molto alto di punti "rappresentativi", complicando oltremodo i calcoli e/o creando problemi di memoria.

1. Obtain a sample of the database.
2. Partition the sample into p partitions.
3. Partially cluster the points in each partition.
4. Remove outliers based on size of cluster.
5. Completely cluster all data in the sample (representatives).
6. Cluster entire database on disk using c points to represent each cluster. An item in the database is placed in the cluster which has the closest representative point to it.

Figura 4.14: Procedura di *CURE*

4.7.11 Algoritmi basati sulla riduzione dimensionale

CLIQUE [75] (Clustering In QUEst), appositamente studiato dai ricercatori di IBM presso il laboratorio di Almaden per lavorare in alta dimensionalità, si basa sulla ricerca di cluster in sottospazi dello spazio dei dati.
Più precisamente, l'ipotesi da cui parte *CLIQUE* è che, in determinati sottospazi dell'insieme dei dati, esistano cluster interessanti, più definiti che nello spazio completo. Al contrario, dei metodi di riduzione dimensionale, *CLIQUE* non scarta a priori alcune dimensioni, ma cerca in vari sottospazi e proiezioni cluster interessanti. È da notare che *CLIQUE* considera solo sottospazi proiettati mediante la "soppressione" di alcune dimensioni, e non mediante la ricombinazione lineare di dimensioni, onde evitare di creare cluster lungo dimensioni di difficile interpretazione umana.
CLIQUE deriva le sue intuizioni, sia dagli algoritmi basati sulla densità che da quelli basati sulla suddivisione dello spazio. I cluster sono, per *CLIQUE*, zone ad alta densità di punti, calcolate sfruttando la quantizzazione dello spazio dei dati in ipercubetti di lato ε , che è un parametro dell'algoritmo. La regione viene considerata densa se vi si trovano più di τ punti. Si omettono i dettagli che sono analoghi a quelli di altri algoritmi basati sulla densità.

Il punto focale, ovviamente, sta nell'identificare quali sono i sottospazi che contengono cluster. Il metodo procede in modo *bottom-up*, sfruttando il fatto che un cluster rilevato in un sottospazio a dimensioni *K* sarà un cluster anche in qualsiasi sottospazio di questo sottospazio a dimensioni *K-1*.

L'algoritmo è composto di passi che prendono in ingresso tutte le regioni dense di dimensione *K-1* (chiamiamolo insieme D_{k-1}), e restituisce un superset dell'insieme delle zone dense a dimensione *K*. Tali zone dense vengono analizzate per determinare quelle realmente interessanti, e una volta, ottenuto l'insieme delle regioni dense *K-dimensionali*, queste vengono ricorsivamente usate come input per calcolare i candidati sottospazi *K+1* dimensionali, e così via. Il passo consiste nel fare una operazione di *join* tra l'insieme D_{k-1} e se stesso, ponendo come condizione che le prime *K-2* dimensioni delle zone dense siano eguali. Le regioni, individuate da queste intersezioni, sono *K-dimensionali* e sono un superset di D_k . Sfruttando il teorema di monotonia, si proiettano tali zone a *K-1* dimensioni: se non sono presenti in D_{k-1}, non sono zone dense in *K* dimensioni e quindi si ottiene D_k.

L'algoritmo inizia, dunque, con una "passata" in cui analizza, dimensione per dimensione, trovando zone dense sulle singole dimensioni, e prosegue in modo ricorsivo. Sventuratamente, ciò significa che la complessità sia O(*ck*+*mk*), con *k* massima dimensionalità di un sottocluster denso, *m* il numero totale dei dati in ingresso, e *c* una costante. Il numero è evidentemente esplosivo, ma esistono ottime euristiche di *pruning*.

Gli output D_k rappresentano l'insieme delle regioni dense, che possono essere suddivise per ogni spazio *K-dimensionale*. A questo punto, *CLIQUE* con un passo di costo O(2*KN*), con *N*, numero di regioni dense del sottospazio, le unisce in un insieme minimale di cluster connessi presenti. Tuttavia, descrivere questi cluster in modo ottimo in base alla griglia di suddivisione dello spazio (ovvero, col numero minimo di ipercubi) è un problema *NP*-completo [76]. Il metodo approssimato generale descrive i cluster con regioni sovradimensionate, con un *bound* di *ln(K)* volte il minimo [77]. *CLIQUE* usa

un metodo in due passaggi (dapprima, copertura greedy e, in seguito, rimozione) ciascuno dei quali opera con un bound che è O(n^2).

Test approfonditi dimostrano performance reali che sono lineari rispetto alle dimensioni del database, e quadratiche rispetto al numero di dimensioni. Test di paragone con algoritmi di clustering standard mostrano che *BIRCH* per esempio "nota" cluster in sottospazi a 5 dimensioni se lo spazio originale ha meno di 10-15 dimensioni, dopodiché essi scompaiono in una nuvola di rumore. *DBSCAN* si comporta pure peggio. Meno chiara, infatti, appare la comparazione con una riduzione dimensionale SVD.

PROCLUS [78] è un altro algoritmo con le stesse caratteristiche, con performance simili, che parte da un approccio diverso, sostanzialmente estendendo il meccanismo dei *K-medoids* di *CLARANS* alla proiezione su sub-spazi. ORCLUS [79] applica lo stesso approccio alla determinazione di sottospazi arbitrari.

MAFIA [80] (Merging Adaptive Finite Intervals And is more than a *CLIQUE*) è una evoluzione di *CLIQUE* che implementa algoritmi adattativi per il dimensionamento non uniforme della griglia, risultando fino a 50 volte più performante. OptiGrid [81] cerca, a sua volta, di calcolare una griglia ottimizzata, con metodi diversi da quelli di *MAFIA* e cercando di usare iperpiani generici; purtroppo, si dimostra sensibile ai parametri in ingresso all'algoritmo che ne controllano il comportamento. Una versione più evoluta di OptiGrid è O-Cluster. Quest'ultimo algoritmo ritorna al concetto di partizionamento in celle parallelo agli assi (in quanto è dimostrabile che i benefici di un partizionamento in celle non parallelo agli assi diminuiscono con l'aumentare delle dimensioni, mentre invece i problemi computazionali aumentano).

La complessità di O-Cluster è O(*Nd*), con *N*, numero di dati e *d*, numero di dimensioni.

4.7.12 Algoritmi basati sulla suddivisione dello spazio (gridbased)

Questa categoria di algoritmi utilizza un approccio sostanzialmente distinto da tutti gli altri. Anziché ragionare sui dati, infatti, ragionano sullo *spazio*, che viene quantizzato in un numero finito di celle (generalmente, degli iperparallelepipedi) sulle quali vengono effettuate le operazioni di clustering.
Si può comprendere immediatamente che questa metodologia punti a rendere la velocità di computazione indipendente dal numero di pattern da classificare, ma dipendente dal numero di celle in cui lo spazio viene suddiviso. Inoltre, "riassumere" il contenuto delle celle in un singolo numero consente di mantenere anche grandi set di dati in memoria, velocizzando i calcoli.
Esistono molti algoritmi che adottano tecniche differenti in questo sottoinsieme, li esploriamo solo rapidamente:

- *STING* [82] (STatistical INformation Grid approach) adopera un approccio statistico, ed è stato studiato per problemi di data mining di tipo spaziale. *STING* può essere facilmente parallelizzato, e utilizzato in modo incrementale. In sostanza, crea una gerarchia di celle di "volume" sempre maggiore, e usa formule matematiche per calcolare i valori statistici delle celle, superiori rispetto a quelle inferiori. Su questi valori si possono, poi, effettuare varie operazioni, tra cui il clustering.
- WaveCluster [83] è una tecnica di clustering che si basa sull'applicazione di una trasformata wavelet ai dati in ingresso; tale approccio è impraticabile per dati ad alta dimensionalità
- *CLIQUE*, *MAFIA*, OptiGrid e O-Cluster sono algoritmi che riducono la dimensionalità. Questi due approcci si integrino molto bene.

Tutti questi algoritmi presentano, comunque, una complessità lineare nel numero di dati. Vi è, inoltre, una buona capacità di riconoscimento di cluster di forma arbitraria e una modesta sensibilità del risultato rispetto ai parametri

di ingresso. Il limite di questi metodi risiede nella qualità dei cluster forniti, che dipende in modo significativo (logicamente) dalla quantizzazione effettuata. Tuttavia, è da notare che con l'aumento delle dimensioni, il numero di sottocelle esplode esponenzialmente, anche se sono stati sviluppati algoritmi per creare una suddivisione in celle ottimale (ad esempio, nei citati OptiGrid, O-Cluster e *MAFIA*). Pertanto, gli algoritmi privi di tali accorgimenti, potrebbero benissimo trovarsi in difficoltà con l'altissima dimensionalità, tipica dei casi di TDM.

4.7.13 Algoritmi basati su grafi

Uno degli algoritmi, concettualmente più semplici, da utilizzare per il clustering di un insieme di punti è la costruzione e il taglio del albero minimo di copertura (Minimal Spanning Tree - *MST*) del grafo che li rappresenta; si tratta di una delle operazioni più studiate nell'algebra e nell'algoritmica di base.

Dato un grafo *G* formato da *V* nodi ed *E* archi, dotati di pesi, l'*MST* è definito come il minimo sottografo connesso e aciclico di *G* che contiene tutti i nodi *V*. Esistono dei ben noti algoritmi che producono un albero minimo di copertura, con un tempo estremamente efficiente, O(*E*log(*V*)), che sfruttano spazio in ragione di O(*E*+*V*). Una considerazione intuitiva riguarda il fatto che i punti appartenenti allo stesso cluster saranno uniti dagli archi di peso minimo all'interno dell'*MST*, mentre quelli appartenenti a cluster diversi saranno uniti da cluster di peso più alto.

Pertanto, ci si limita a disconnettere il grafo, rimuovendo progressivamente archi (in ordine di peso, dal più alto al più basso), fino ad ottenere il numero di cluster desiderato.

Un algoritmo decisamente più efficace ed efficiente è *Association Rules Hypergraph Partitioning* (*ARHP*). Questo metodo [84] rappresenta i dati di grandi dimensioni, in un modello a ipergrafo. In tale modello, ogni dato viene rappresentato come un vertice e le relazioni tra dati simili vengono

rappresentate come archi. Il peso dell'arco riflette la forza della vicinanza tra i vertici. Un algoritmo di partizionamento viene applicato all'ipergrafo per trovare una suddivisione tale che i punti contenuti in ogni cluster siano fortemente correlati e gli archi tagliati dalla suddivisione siano di peso minimo. È un algoritmo di *hard clustering* di tipo *piatto* e *non iterativo***.** In pratica, si utilizza la stessa schematizzazione dell'algoritmo *MST*, ma considerando anche la densità degli archi che connettono vertici appartenenti allo stesso cluster, non soltanto la minimalità degli archi tagliati dalla suddivisione.

Il partizionamento degli ipergrafi è un problema studiato nel contesto dello sviluppo dei circuiti VLSI ed esistono algoritmi (esempio, HMETIS) che consentono nel trovare con ottima efficienza una buona soluzione al problema (ancorché, non ottima). In particolare, la complessità di questo algoritmo è $O((E+V)\log(K))$, dove V è il numero dei vertici, E, il numero degli archi, e K, il numero di cluster.

L'algoritmo, ovviamente, non consente di fare addestramento *online*. Inoltre, il numero di classi deve essere necessariamente conosciuto a priori.

La correlazione tra i dati si può basare su regole di associazione (nei problemi di data mining di tipo commerciale, ad esempio), ma anche su una metrica di distanza, definita per ogni coppia di oggetti. L'efficacia di tale algoritmo viene messa in dubbio dalla mancanza di prove sperimentali che dimostrino che esso produca una suddivisione, di migliore qualità, per tipi di dati temporali; ma anche le considerazioni di efficienza conducono ad essere dubbiosi. Infatti, costruire un grafo euclideo (ovvero, che contiene archi tra tutti i punti, ognuno dei quali dotato di un peso crescente, in funzione della distanza) è relativamente semplice in due dimensioni: si tratta di costruire la *triangolazione di Delaunay*, per cui esistono algoritmi noti che hanno complessità temporale $O(V\log(V))$ e spaziale $O(V)$, ma che non scalano assolutamente a più di due dimensioni, in quanto, la triangolazione diventa un problema esponenziale. Oltre le due dimensioni la complessità diventa

quadratica, $O(V^2)$, inoltre l'algoritmo coinvolge il calcolo della distanza euclidea, che viene appesantito dalla crescita dimensionale.
Va ricordato, comunque, uno dei possibili punti di forza di questo tipo di algoritmo, che è la sua possibilità di descrivere regioni e cluster non sferici.

4.7.14 COBWEB

COBWEB è un sistema di clustering incrementale basato su alberi probabilistici. Tale algoritmo è molto peculiare nel fatto che supporta, di base, solo attributi nominali, e non numerici.
Ogni foglia è costituita da una classe. Per ogni possibile valore di ogni feature, vengono definite le percentuali di probabilità, all'interno della classe. Ogni nodo riporta i valori aggregati di tutti i suoi figli.
Per ogni nuovo dato, *COBWEB* parte dalla radice dell'albero ed esegue ricorsivamente l'algoritmo sintetizzato di seguito, *COBWEB* (dato,nodo):
1. Se il nodo è una foglia, crea due figli, in uno inserisce la foglia, nell'altro il nuovo dato, e poi si aggiorna;
2. Si aggiunge il nuovo dato al nodo e se ne aggiorna le probabilità;
3. Si valuta il miglioramento nella *Category utility*, per ciascuna delle seguenti alternative:
a. Creare una nuova classe per il dato
b. Inserire il dato nella migliore categoria figlia
c. Inserire il dato nella seconda migliore categoria figlia
d. Unire la prima e la seconda categoria in una categoria unica
e. Eliminare il nodo corrente e sostituirlo con le categorie figlie(split)
4. A seconda di quale sia il migliore metodo, completare il passo:
a. Creare la nuova categoria e inserire il dato
b. Eseguire COBWEB(dato, c_1)
c. Eseguire COBWEB(dato, c_2)
d. Sostituire c_1 e c_2 con $c_m = c_1+c_2$ ed eseguire COBWEB(dato, cm)

e. Eliminare il nodo corrente e sostituirlo con le categorie figlie e rieseguire COBWEB (dato, nodo_genitore).

La *category utility* è una funzione che cerca di massimizzare l'idea che si ha dei cluster: istanze della stessa classe avranno, frequentemente, valori uguali; istanze di classi diverse avranno, spesso, valori diversi:

$$CU = \sum_C \sum_A \sum_v P(A = v | C) P(C | A = v) P(A = v) \qquad [4.9]$$

Le tre parti della formula significano, rispettivamente, la probabilità che l'attributo *A* abbia valore *v*, dato che il vettore appartiene a *C*; la probabilità che il vettore appartenga a *C,* dato che *A* ha valore *v*, e la probabilità che *A* abbia, in generale, valore *v* (per "pesare" attributi comuni e non comuni). Per il teorema di Bayes:

$$CU = \sum_C \sum_A \sum_v P(A = v / C) \frac{P(C) P(A = v / C)}{P(A = v)} P(A = v) = \qquad [4.10]$$
$$= \sum_C \sum_A \sum_v P(A = v / C)^2 P(C)$$

Estraendo un termine invariante, otteniamo:

$$CU = \sum_C P(C) \sum_A \sum_v P(A = v / C)^2 \qquad [4.11]$$

Ma, $\sum_A \sum_v P(A = v / C)^2$ non è altro che il valore atteso degli attributi che possono essere indovinati correttamente, sapendo che un certo vettore appartiene alla classe *C*, cioè:

$$\sum_A \sum_v P(A = v)^2 \qquad [4.12]$$

In realtà, la *CU* si definisce come "l'incremento nel numero atteso di valori che possono essere indovinati correttamente, dato un insieme di *n* categorie, rispetto al numero atteso di quelli che possono essere indovinati correttamente senza tale conoscenza", e vale, pertanto:

$$CU = \frac{1}{N}\sum_C P(C)\sum_A \sum_v \left[P(A=v/C)^2 - P(A=v)^2\right] \quad [4.13]$$

dove il coefficiente *1/N* viene inserito per consentire di confrontare cluster di diversa numerosità.

4.7.15 AutoClass (Bayesian Classification, EM clustering)

AutoClass [85] è un algoritmo *piatto* di *soft clustering* basato sul modello statistico detto "finite mixture model", dotato di un metodo Bayesiano per determinare il numero ottimo di classi. Questo algoritmo cerca di trovare il più probabile insieme di classi che descrive l'insieme di dati di training, basandosi sulle aspettative a priori. La descrizione delle classi consiste in un tipo specifico di funzione di densità di probabilità, e l'assegnamento di ogni dato a una classe viene fornito, in termini di probabilità, piuttosto, che in termini deterministici, come normalmente avviene con gli altri algoritmi.
In pratica, il modello descrive i dati usando due tipi di funzione di densità di probabilità: innanzitutto, una bernoulliana che descrive la probabilità che un dato appartenga a una certa classe; un altra f.d.p. descrive, invece, per ogni classe, la probabilità di osservare un determinato vettore di valori, posto che il vettore appartenga a quella classe. Tale f.d.p. è il prodotto delle funzioni di densità di probabilità dei singoli attributi (per quella classe) che possono essere indipendenti o covariate (Poissoniane, Bernoulliane, Gaussiane o altre). Si può notare che la costruzione dello schema di queste funzioni richiede la conoscenza di modelli statistici, di cui a priori non si dispone, relativamente ai dati di rete che si vogliono classificare.

Per determinare i parametri, l'algoritmo effettua una stima mediante un processo di expectation-maximization (EM) che converge verso un massimo locale della funzione di verosimiglianza. Una delle approssimazioni che l'algoritmo impone è l'indipendenza tra le f.d.p. dei singoli attributi, ma ve ne sono altre, nascoste all'interno di passaggi di calcolo probabilistico, di per sé, troppo complessi.
L'algoritmo può determinare il numero più probabile di classi anche se questo non è noto a priori, ma funziona notoriamente in modo più efficace se questo dato viene fornito.
Uno dei grossi limiti di AutoClass è che non funziona bene con dati di alta dimensionalità: la complessità computazionale è infatti O(kd^2nI) con *k,* numero di cluster; *d,* numero di dimensioni; *n,* numero di oggetti da clusterizzare; *I,* numero medio di iterazioni dell'algoritmo.
Non sembrano esserci, ad oggi, metodi per il riaddestramento online, e, in ogni caso, la complessità computazionale pare proibitiva.

4.7.16 Algoritmi Genetici

Formulare un problema di clustering, nei termini tipici di un algoritmo, di tipo genetico - AG (o ad altri algoritmi evolutivi) è a prima vista semplice. Come è noto, gli algoritmi genetici operano generando soluzioni candidate, valutandone l'adeguatezza (fitness), e facendo "riprodurre" le funzioni migliori, mediante speciali meccanismi (crossover e mutazione) che tentano di emulare il processo evolutivo delle specie.
Evidentemente, si può immaginare che lo spazio delle soluzioni sia rappresentato da ogni possibile suddivisione dell'insieme di training. Basta imporre un criterio di fitness (per esempio, la minimizzazione dello scarto quadratico, nei singoli cluster) per applicare un algoritmo genetico al problema. Tuttavia, si devono codificare le singole soluzioni per poterle incrociare automaticamente. Sia *T*, l'insieme dei dati di test, con *t* elementi.

Per individuare *K* partizioni in *T*, il metodo intuitivo è quello di avere una stringa di *t* valori $x_1,x_2,\ldots,x_t$, con $1<x_i<K$. In poche parole, a ogni elemento di *T* si associa il numero del cluster a cui appartiene in quella soluzione. Questo crea un problema di ridondanza: vi sono *K!* soluzioni perfettamente identiche (con numeri diversi). Inoltre, si mostra, abbastanza semplicemente, che l'operatore di crossover, a punto singolo (si divide in due ciascuna stringa, in un punto fissato, ed il figlio prende parte dei dati del "padre" e parte di quelli della "madre"), in questa rappresentazione può facilmente generare "figli" di qualità, di gran lunga inferiore, rispetto al padre.

Per questo sono stati proposti schemi diversi di codifica ed operatori di crossover migliorati; ad esempio, una codifica, in cui gli elementi di *t* vengono inseriti nella stringa di soluzione, separati da un *, che possa marcare la divisione tra cluster, utilizzando l'operatore "permutazione" per il cross-over (con una soluzione simile a quella per il Traveling Salesman Problem [86]). Tuttavia, questo non risolve il problema della ridondanza, che è sempre fattoriale. Alcuni ricercatori [87] hanno anche proposto di usare un algoritmo genetico con un *encoding* dei dati su ipergrafo, simile a quello di *ARHP*, e usare un algoritmo di *edge-based crossover* [88]. Tuttavia, l'ordine di complessità dell'operatore, che è $O(K6+N)$, con *K*, numero dei cluster e *N*, numero dei dati, lo rende estremamente sensibile alla crescita del numero di cluster, improponibile per $K>10$.

Altri problemi che rendono difficilmente praticabile, per ora, l'uso di AG in problemi pratici di classificazione non supervisionata è l'estrema sensibilità a macroparametri, quali la densità della popolazione di soluzioni, i parametri di crossover, eccetera. L'uso di AG, in problemi di clustering, viene reso efficace proprio da quelle conoscenze di dominio che, in questo caso, si sa di non avere. Inoltre, uno studio [89] dimostra come le performance dei AG rispetto ad altri metodi decrescano con il numero di dimensioni in esame.

4.7.17 ART

ART (Adaptive Resonance Theory) è un termine che indica una particolare struttura di algoritmo neurale basato sulla risonanza. La teoria che sta alla base della rete *ART* è che un input sufficientemente vicino a uno dei cluster, riconosciuti dalla rete, la farà entrare in uno stato simile a quello del fenomeno fisico della risonanza, innescando un ciclo virtuoso di autoeccitazione.

Il modello originale (*ART*-1) è in grado solo di trattare dati booleani, mentre *ART*-2 [90] è in grado di gestire dati reali. *ARTMAP* è una implementazione supervisionata di *ART*, mentre *Fuzzy ART* estende il modello *ART*-1, per gestire dati fuzzy (e, incidentalmente, i dati reali, purché normalizzabili).

Le reti *ART* sono capaci di effettuare learning non supervisionato e incrementale. Tuttavia, sono order-dependent, cosa, questa che potrebbe ingenerare problemi.

Esiste, poi, un algoritmo denominato *IART*-1 che diminuisce tale dipendenza ordinale, ma funziona solo per i valori booleani. Per migliorare *ARTMAP* (che è formato dalla connessione di due moduli *ART*-1) e diminuirne la dipendenza dall'ordine degli ingressi, è stato presentato *Gaussian ARTMAP* [91], basato su una evoluzione di *ART*-1, denominata *Gaussian* ART. Altre estensioni recentissime [92] al paradigma *ART* sembrano superare alcuni di questi problemi, ma è ricerca aperta.

Inoltre, un parametro globale detto "vigilance threshold" stabilisce quanto deve essere "nuovo" un input per stimolare la creazione di un nuovo cluster.

Non sono disponibili in letteratura valutazioni effettive sulla complessità computazionale di questo tipo di algoritmi. Inoltre, ART-2 ha problemi di complessità computazionale, nonché problemi sulla adeguatezza a gestire dati di alta dimensionalità.

Capitolo 5

Applicazione su un dataset reale di serie temporali, provenienti da sistemi radar satellitari

5.1 Introduzione

La Tecnica PS (*Permanent Scatterers Technique* - PSInSAR™ - T.R.E. s.r.l.) è stata sviluppata e brevettata dal Politecnico di Milano e concessa in licenza esclusiva a TRE, primo *spin-off* commerciale del Politecnico, nel 2000. Si tratta di uno strumento molto efficace per il monitoraggio ad alta precisione di fenomeni di deformazione della superficie terrestre (Ferretti et al., 2000, 2001; Colasanti et al., 2003), basato sull'impiego di serie temporali di immagini radar satellitari (in particolare, si tratta di dati dei satelliti ERS-1/2 dell'ESA - *European Space Agency*).

I sistemi radar satellitari coerenti e, nello specifico, i radar di tipo SAR (*Synthetic Aperture Radar - Radar ad Apertura Sintetica*), utilizzati

nell'applicazione di seguito presentata, sono in grado di misurare la distanza tra il sensore e il bersaglio, registrando il tempo di volo, tra l'onda trasmessa e la porzione retro diffusa[1].

Grazie alla loro periodicità di acquisizione (all'incirca mensile), i dati SAR forniscono misure ripetute della distanza sensore-bersaglio, consentendo di misurarne gli spostamenti nel tempo, mediante confronti successivi.

La Tecnica PS si pone come obiettivo quello di sfruttare tutte le acquisizioni disponibili su una stessa area e di individuare quei bersagli (*Permanent Scatterers*) che mantengano inalterate nel tempo le proprie caratteristiche elettromagnetiche. Per ciascuno di essi, è possibile stimare e rimuovere il disturbo atmosferico e, quindi, ricostruirne la storia dei movimenti, in un intervallo di tempo analizzato con precisione millimetrica.

5.2 Tecnica dei diffusori permanenti con approccio PS

Il cuore dell'algoritmo della Tecnica PS consiste nella stima del contributo atmosferico, presente su ogni acquisizione, per procedere alla successiva compensazione dello stesso e, quindi, alla determinazione dei termini di fase che descrivono il moto dei *Permanent Scatterers*.

L'elaborazione risulta essere molto onerosa in termini computazionali, per questo, non viene condotta estensivamente su tutti i punti individuati nella prima fase, ma su di un loro sottoinsieme, costituito da quelli ritenuti migliori. A questo insieme di punti ci si riferisce, generalmente, con il nome di *cluster* relativo ai *Permanent Scatterers Candidates* (PSC).

Grazie al fatto che le caratteristiche del disturbo atmosferico variano lentamente, nell'intorno di ciascun bersaglio radar, l'informazione ricavata, per il sottoinsieme, può essere estesa a tutta l'immagine, consentendo, così, di stimare, correttamente, l'intero contributo atmosferico.

[1] Per un approfondimento sui sistemi radar si rimanda all'Appendice B.

Nel corso dell'analisi da condurre, si individua il punto, utilizzato come riferimento, che si ipotizza essere fermo ed esente da errori di quota, a cui sono riferite tutte le altre misure.

I risultati della Tecnica PS, infatti, sono differenziali, ovvero riferiti, temporalmente, alla data di acquisizione dell'immagine *master* e, spazialmente, al punto di riferimento.

Una volta stimato e rimosso il contributo atmosferico, per ogni immagine che costituisce il dataset, è possibile procedere con l'estrazione delle informazioni di movimento di ciascun *Permanent Scatterers*.

Il parametro che descrive la qualità delle misure effettuate è la "coerenza", indice normalizzato tra *0* e *1*, che risulta essere legato anche alla deviazione standard del rumore di fase del punto. La coerenza è funzione del numero di immagini elaborate e della distribuzione di *baseline* temporali e spaziali.

Si dice che un punto è considerato attendibile se il valore di coerenza associato è tale da garantire una probabilità di falso allarme inferiore a 10-5 per le analisi *Standard* e 10-4 per le analisi *Advanced*. Per falso allarme si intende la possibilità che una sequenza di rumore venga interpretata come movimento.

In base all'acquisizione, di volta in volta, avvenuta delle serie storiche di deformazione, è possibile ricostruire l'evoluzione della distanza sensore-bersaglio (nella direzione di LOS) del punto in esame. Lo spostamento, che viene associato a ciascuna acquisizione, espresso in millimetri, è di tipo differenziale, riferito allo stesso punto di riferimento, con cui si stima il contributo atmosferico. Si tratta di un'operazione di geocodifica, che consente di ricampionare i dati ottenuti in coordinate SAR, su una griglia geografica (latitudine/longitudine). Per questo motivo, è necessario disporre di un punto di coordinate note (*Ground Control Point*), in entrambe le geometrie, che garantisca una corrispondenza tra i due sistemi di riferimento.

5.2.1 Le basi della Tecnica PS

L'approccio PS è basato sull'osservazione che un piccolo sottoinsieme di bersagli radar, costituito appunto dai diffusori permanenti (*Permanent Scatterers*, PS), è di fatto immune agli effetti di decorrelazione. Essi mantengono la stessa "firma elettromagnetica", in tutte le immagini, al variare della geometria di acquisizione e delle condizioni climatiche, preservando, quindi, l'informazione di fase, nel tempo. I PS sono tipicamente parti di edifici, strutture metalliche, rocce esposte, comunque, elementi già presenti al suolo, per i quali, le caratteristiche elettromagnetiche non variano sensibilmente di acquisizione in acquisizione, cosa che non accade, ad esempio, per la vegetazione che muta di continuo.

La Figura 5.1 mostra una rappresentazione schematica della base teorica della tecnica *interferometrica* di un PS, nonché dei disturbi presenti nelle acquisizioni SAR (es.: variazione della componente di riflettività, che dà luogo a decorrelazione temporale; variazioni del *baseline normale* che dà luogo a decorrelazione geometrica; disturbi atmosferici).

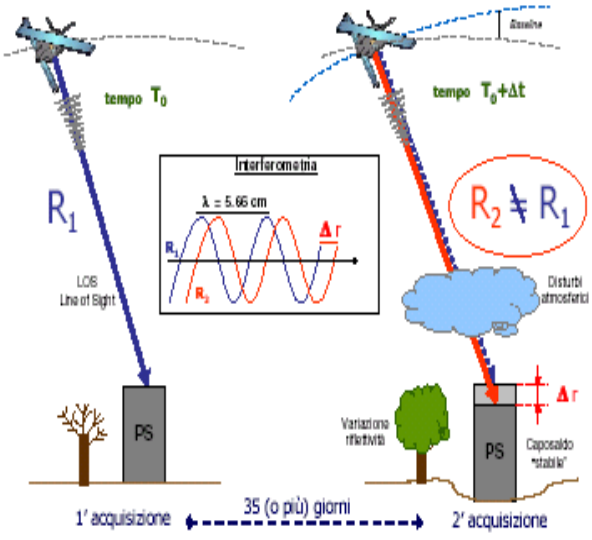


Figura 5.1: Rappresentazione schematica della base teorica della tecnica interferometrica di un PS e dei disturbi presenti nelle acquisizioni SAR

L'elaborazione prende origine da uno studio statistico delle immagini, che porta alla selezione dell'insieme dei PS, sostanzialmente, immuni ai fenomeni di *decorrelazione*. Tale proprietà dei PS consente di superare gran parte delle limitazioni legate all'analisi DInSAR (Appendice B), convenzionale.

Per i PS, infatti, utilizzando le serie storiche delle acquisizioni, è consentito stimare, sia l'entità del disturbo dovuto all'atmosfera terrestre, sia le possibili deformazioni superficiali della zona in esame[2]. Dopo aver rimosso il disturbo atmosferico dai dati, si è in grado di stimare accuratamente i movimenti dei PS, utilizzando il SAR, come un vero e proprio strumento di misura delle deformazioni del terreno.

Si può immaginare la griglia di PS come una rete di stazioni GPS (*Global Positioning System*) naturali per il monitoraggio di vaste aree di interesse, con una frequenza di aggiornamento del dato mensile e con una densità spaziale di punti di misura estremamente elevata (in aree urbane 100-400 PS/km^2).

Il cuore del brevetto della Tecnica PS risiede nella capacità di stimare correttamente e di compensare il disturbo atmosferico che limita le applicazioni interferometriche, svolgendo un'analisi dettagliata solo sul sottoinsieme di PS, accuratamente selezionato, secondo valutazioni statistiche.

Una volta stimato il contributo atmosferico, è stato possibile:

- individuare tutte le componenti che costituiscono la fase interferometrica;
- eliminare i termini spuri e isolare il termine che descrive la variazione di cammino ottico dell'onda elettromagnetica, nelle varie acquisizioni, consentendo di descrivere i movimenti che ha subito il bersaglio, nell'arco temporale, tra il primo e l'ultimo dato disponibile.

[2] Ciò è reso possibile da un insieme di algoritmi di elaborazione numerica dei segnali, frutto di oltre dieci anni di studi sviluppati dal gruppo SAR del Politecnico di Milano.

Per eseguire stime accurate dei disturbi atmosferici è necessario che la densità spaziale di PS sia sufficientemente elevata (maggiore di 5-10 PS/km^2), vincolo sempre verificato in aree urbane, utilizzando dataset di almeno 25-30 immagini ERS. In aree ad elevata urbanizzazione, la densità spaziale di PS raggiunge valori molto alti: 100-400 PS/km^2.

In corrispondenza di ciascun PS, si effettua una misura di deformazione, per ogni acquisizione disponibile, con precisione sino a 1-2 mm su ogni singola misura (per i punti migliori). Si è, quindi, in grado di ricostruire il *trend medio di deformazione annua*, con precisione compresa tra 0.1 e 1 mm/anno. La precisione è funzione del numero di immagini e della "qualità" del PS stesso, cioè di quanto l'informazione di fase disponibile, presso il PS, risulti immune ai fenomeni di disturbo.

Tutte le misure sono rilevate lungo la congiungente sensore-bersaglio LOS (*Line of Sight*), e sono di tipo differenziale, ottenute dopo avere determinato uno o più punti di riferimento a terra, di coordinate note e supposti fermi o espressamente indicati ad esempio da misure GPS o di livellazione ottica.

Per la visualizzazione delle stime ottenute, è possibile, poi, rappresentare il *trend medio di deformazione* su un qualsiasi background che aiuti un'interpretazione e una geolocalizzazione dei fenomeni in atto (l'ottimo si raggiunge operando in ambiente GIS, dove l'utente può selezionare il *layer* opportuno).

La Tecnica PS consente, quindi, di stimare e separare i vari contributi, sfruttandone i diversi comportamenti nella dimensione temporale (*inter* interferogramma) e spaziale (*intra* interferogramma).

Il primo obiettivo dell'analisi PS condotta è stato quello di individuare quei punti, detti *Permanent Scatterers*, che mantengono la propria "firma elettromagnetica" costante nel tempo e indipendente dal momento dell'acquisizione (*baseline* temporale), e che sono, inoltre, caratterizzati da un comportamento puntiforme, cioè quelli che presentano una scarsa sensibilità alle variazioni del punto di vista del sensore, nelle varie acquisizioni.

In base ai capisaldi individuati, è possibile, quindi, stimare successivamente il disturbo atmosferico e, una volta compensato, determinare le deformazioni in corso, ovvero le componenti della fase associabili al moto.
Lo studio condotto si focalizza sull'analisi statistica delle caratteristiche di *pixel* omologhi, nelle diverse immagini, evidenziando i parametri più opportuni da utilizzare come soglie, per discriminare gli effettivi *Permanent Scatterers* dagli altri *pixel*.

5.3 Caratteristiche del database utilizzato per l'analisi

Il database utilizzato per l'analisi e stato fornito dal Progetto TELLUS– Unità di Supporto Locale n. 6 Campania - Progetto Operativo Difesa Suolo (PODiS) – PON ATAS QCS 2000-2006, Direzione Generale Difesa Suolo - Ministero dell'Ambiente e della Tutela del Territorio e del Mare.
Il "file .DBF" comprende le caratteristiche dei punti di misura, identificati (PS), dei quali, ciascuno corrisponde ad un *record* del database, dando luogo alle seguenti informazioni:

- *Code*: codice che permette di identificare univocamente il punto di misura all'interno del file;
- *Lat*: Latitudine del PS, espressa mediante ellissoide di riferimento "WGS 84";
- *Lon*: Longitudine del PS, espressa mediante ellissoide di riferimento "WGS 84";
- *Vel*: Velocità media del PS, espressa in mm/anno (valutata rispetto al punto di riferimento);
- *Coherence*: indice di affidabilità delle misurazione (numero compreso tra 0 e 1);
- numero di campi aggiuntivi, pari al numero di immagini elaborate, contenenti, per l'immagine *i*-esima, lo spostamento in *mm* stimato per

il punto in esame, rispetto all'immagine *master* (riferimento temporale).

Tabella 5.1: Area di Benevento – Dataset Discendente

PS report	
Numero di scene elaborate	**72**
Intervallo temporale di analisi	**24 giugno 1992 23 dicembre 2000**
Master	**35448**
Dato di supporto per la geocodifica	**CTR**
Sistema di geocodifica utilizzato	**WGS84 - UTM 33N**
Posizione del punto di riferimento	**Nord: 4552673,80 Est: 481766,11**
Ipotesi sul moto del punto di riferimento	**Stabile**
Estensione dell area di interesse	**1200 ~ km^2**
Numero di PS identificati	**71557**
Soglia minima di coerenza	**0,043055556**
Densità media dei PS (PS/km^2)	**60 ~ PS/km^2**

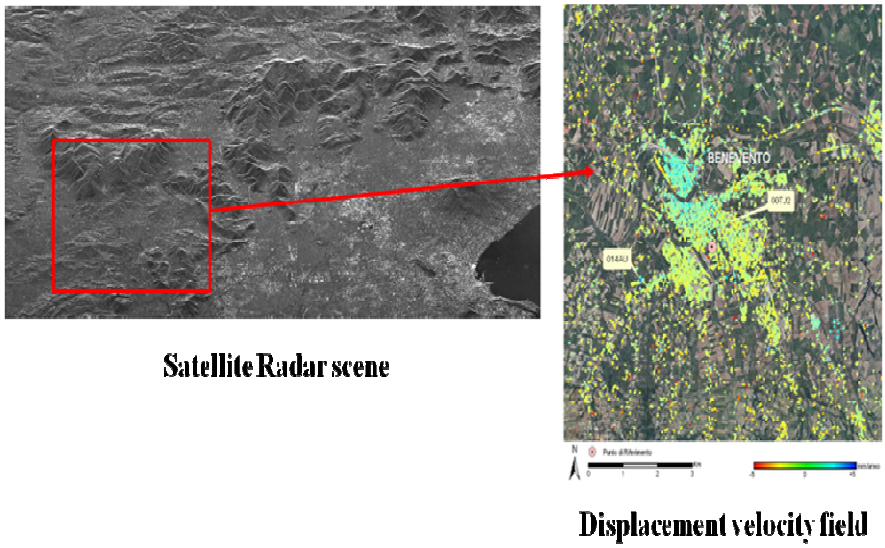


Figura 5.2: Visualizzazione delle aree osservate dal satellite

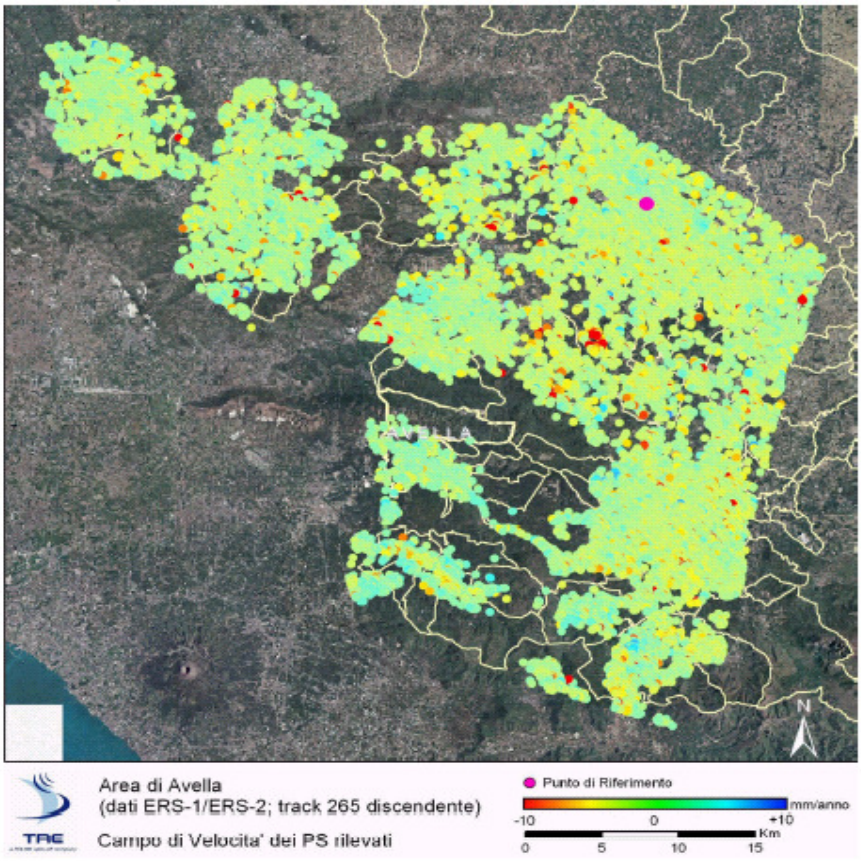


Figura 5.3: Velocità calcolate lungo la congiungente sensore bersaglio (LOS)

5.4 Algoritmo di clustering applicato ad un dataset relativo all'area di Avellino Benevento, rilevato secondo la tecnica PM

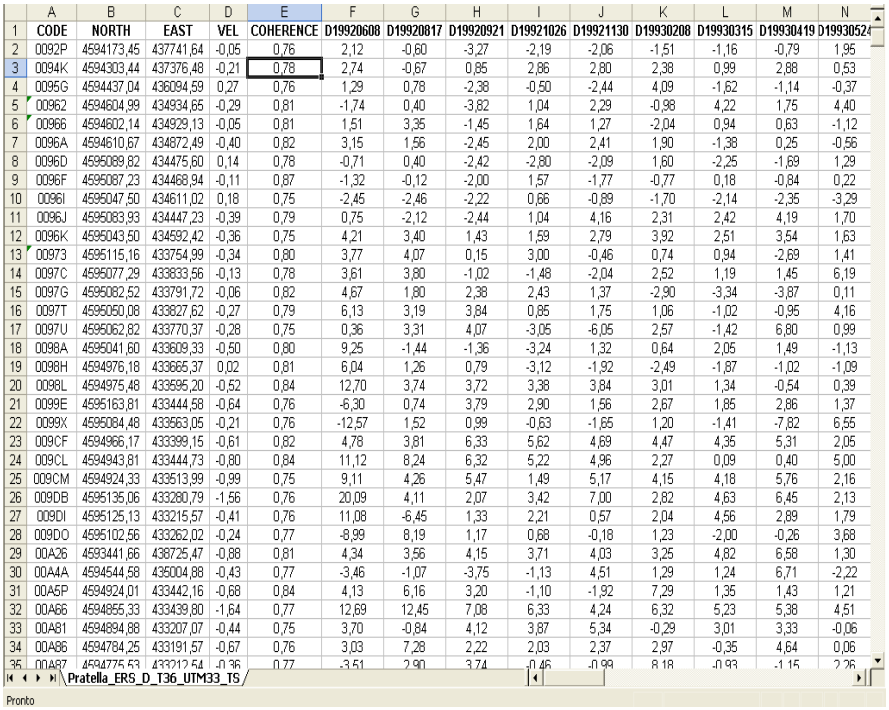
Il dataset utilizzato per l'analisi, denominato "Database Avella_ERS_D_1992_2001_PS_TS" è stato estratto da una matrice di dati, in cui è stata raccolta la descrizione del *trend* di deformazione del suolo, rispetto alla velocità media di un'area, su cui si effettuavano rilevazioni radio satellitari, che comprendeva il territorio Avellinese e Beneventano.

Dal dataset completo, che prevedeva 71.557 PS, si è passati ad effettuare un'estrazione dei dati, in base ad un test di "coerenza" che comportava l'esclusione di tutti i PS della matrice originaria che non superassero il valore di coerenza dello 0.76. In base a tale criterio, il numero di PS estratti è risultato 18.452. I dati estratti sono espressi sotto forma di serie storiche relative a 72 tempi di osservazione radar satellitare (Figura 5.4).

L'idea di base, relativa al trattamento di tali dati, è stata quella di generare dei gruppi, che oltre ad essere coerenti dal punto di vista statistico, potessero cogliere gli aspetti propri dell'interpretazione geologica da dare alla deformazione della porzione di territorio interessato. Pertanto, la procedura di analisi ha riguardato, innanzitutto, la fase di utilizzo di un algoritmo di clustering (*CLARA*), per generare, in base alla scelta a priori del numero delle classi, la classificazione migliore.

La scelta di tale algoritmo risiede nella sua accettabile complessità computazionale e del suo ridotto numero di parametri in ingresso, in confronto ad altri algoritmi utilizzati, ma che si è ritenuto di non dover riportare, per non appesantire il lavoro di tesi. Nello specifico, l'algoritmo richiedeva un unico parametro in ingresso, ovvero, lo stabilire a priori il numero delle K classi da ricercare.

La scelta di adottare un approccio *non supervisionato*, anche per la classificazione dell'immagine di cui è disponibile la relativa *mappa di realtà al suolo*, è dovuta al fatto che, sebbene una classificazione supervisionata sia più accurata di una non supervisionata, le classi fornite dalla mappa di realtà al suolo sono tipicamente multimodali, a causa della presenza di differenti tipologie di copertura del suolo corrispondenti a ciascuna classe.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CODE	NORTH	EAST	VEL	COHERENCE	D19920608	D19920817	D19920921	D19921026	D19921130	D19930208	D19930315	D19930419	D1993052
2	0092P	4594173,45	437741,64	-0,05	0,76	2,12	-0,60	-3,27	-2,19	-2,06	-1,51	-1,16	-0,79	1,95
3	0094K	4594303,44	437376,48	-0,21	0,78	2,74	-0,67	0,85	2,86	2,80	2,38	0,99	2,88	0,53
4	0095G	4594437,04	436094,59	0,27	0,76	1,29	0,78	-2,38	-0,50	-2,44	4,09	-1,62	-1,14	-0,37
5	00962	4594604,99	434934,65	-0,29	0,81	-1,74	0,40	-3,82	1,04	2,29	-0,98	4,22	1,75	4,40
6	00966	4594602,14	434929,13	-0,05	0,81	1,51	3,35	-1,45	1,64	1,27	-2,04	0,94	0,63	-1,12
7	0096A	4594610,67	434872,49	-0,40	0,82	3,15	1,56	-2,45	2,00	2,41	1,90	-1,38	0,25	-0,56
8	0096D	4595089,82	434475,60	0,14	0,78	-0,71	0,40	-2,42	-2,80	-2,09	1,60	-2,25	-1,69	1,29
9	0096F	4595087,23	434468,94	-0,11	0,87	-1,32	-0,12	-2,00	1,57	-1,77	-0,77	0,18	-0,84	0,22
10	0096I	4595047,50	434611,02	0,18	0,75	-2,45	-2,46	-2,22	0,66	-0,89	-1,70	-2,14	-2,35	-3,29
11	0096J	4595083,93	434447,23	-0,39	0,79	0,75	-2,12	-2,44	1,04	4,16	2,31	2,42	4,19	1,70
12	0096K	4595043,50	434592,42	-0,36	0,75	4,21	3,40	1,43	1,59	2,79	3,92	2,51	3,54	1,63
13	00973	4595115,16	433754,99	-0,34	0,80	3,77	4,07	0,15	3,00	-0,46	0,74	0,94	-2,69	1,41
14	0097C	4595077,29	433833,56	-0,13	0,78	3,61	3,80	-1,02	-1,48	-2,04	2,52	1,19	1,45	6,19
15	0097G	4595082,52	433791,72	-0,06	0,82	4,67	1,80	2,38	2,43	1,37	-2,90	-3,34	-3,87	0,11
16	0097T	4595050,08	433827,62	-0,27	0,79	6,13	3,19	3,84	0,85	1,75	1,06	-1,02	-0,95	4,16
17	0097U	4595062,82	433770,37	-0,28	0,75	0,36	3,31	4,07	-3,05	-6,05	2,57	-1,42	6,80	0,99
18	0098A	4595041,60	433609,33	-0,50	0,80	9,25	-1,44	-1,36	-3,24	1,32	0,64	2,05	1,49	-1,13
19	0098H	4594976,18	433665,37	0,02	0,81	6,04	1,26	0,79	-3,12	-1,92	-2,49	-1,87	-1,02	-1,09
20	0098L	4594975,48	433595,20	-0,52	0,84	12,70	3,74	3,72	3,38	3,84	3,01	1,34	-0,54	0,39
21	0099E	4595163,81	433444,58	-0,64	0,76	-6,30	0,74	3,79	2,90	1,56	2,67	1,85	2,86	1,37
22	0099X	4595084,48	433563,05	-0,21	0,76	-12,57	1,52	0,99	-0,63	-1,65	1,20	-1,41	-7,82	6,55
23	009CF	4594966,17	433399,15	-0,61	0,82	4,78	3,81	6,33	5,62	4,69	4,47	4,35	5,31	2,05
24	009CL	4594943,81	433444,73	-0,80	0,84	11,12	8,24	6,32	5,22	4,96	2,27	0,09	0,40	5,00
25	009CM	4594924,33	433513,99	-0,99	0,75	9,11	4,26	5,47	1,49	5,17	4,15	4,18	5,76	2,16
26	009DB	4595135,06	433280,79	-1,56	0,76	20,09	4,11	2,07	3,42	7,00	2,82	4,63	6,45	2,13
27	009DI	4595125,13	433215,57	-0,41	0,76	11,08	-6,45	1,33	2,21	0,57	2,04	4,56	2,89	1,79
28	009DO	4595102,56	433262,02	-0,24	0,77	-8,99	8,19	1,17	0,68	-0,18	1,23	-2,00	-0,26	3,68
29	00A26	4593441,66	436725,47	-0,88	0,81	4,34	3,56	4,15	3,71	4,03	3,25	4,82	6,58	1,30
30	00A4A	4594544,58	435004,88	-0,43	0,77	-3,46	-1,07	-3,75	-1,13	4,51	1,29	1,24	6,71	-2,22
31	00A5P	4594924,01	433442,16	-0,68	0,84	4,13	6,16	3,20	-1,10	-1,92	7,29	1,35	1,43	1,21
32	00A66	4594855,33	433439,80	-1,64	0,77	12,69	12,45	7,08	6,33	4,24	6,32	5,23	5,38	4,51
33	00A81	4594894,88	433207,07	-0,44	0,75	3,70	-0,84	4,12	3,87	5,34	-0,29	3,01	3,33	-0,06
34	00A86	4594784,25	433191,57	-0,67	0,76	3,03	7,28	2,22	2,03	2,37	2,97	-0,35	4,64	0,06
35	00A87	4594775,53	433212,54	-0,36	0,77	-3,51	2,90	3,74	-0,46	-0,99	8,18	-0,93	-1,15	2,26

Figura 5.4: Un estratto del database utilizzato per l'analisi

Al contrario, i cluster forniti da un classificatore non supervisionato, sono in generale, unimodali, quindi possono essere considerati come sottoclassi di una classe fornita dalla mappa di realtà al suolo.

Il primo problema metodologico incontrato nell'analisi è stato, dunque, la definizione del numero ottimale di cluster, da fornire come dato di ingresso all'algoritmo *CLARA*. Successivamente, ci si è concentrati sull'immagine per la quale è disponibile la realtà al suolo (per semplicità, tale immagine è stata identificata come quella alla prima data). L'informazione derivante dalla realtà al suolo è stata utilizzata per guidare l'ottimizzazione del numero di cluster *K*. In base ai risultati sperimentali effettuati sul dataset, si è ritenuto che la scelta di *K=4* fosse la migliore, pur non escludendo l'interessante apporto informativo che potesse essere riscontrato nella classificazione dei *PS* a 5 e 6 gruppi.

Per l'interpretazione dell'output dell'algoritmo si è fatto ricorso al contributo di esperti geologi che hanno verificato la coerenza delle scelte statistiche nell'analisi dei dati.

L'output sulla composizione dei gruppi, sia nella classificazione a 4, 5 e 6 fa riferimento alle seguenti misure (Tabb.: 5.2-5.4):

- N = *Numerosità di ciascun cluster*
- A = *Massima dissimilarità dalla mediana*. Percentuale di dissimilarità tra le osservazioni nei cluster e il parametro (valore mediano) dei cluster
- B = *Diametro del cluster*. Massima dissimilarità tra due osservazioni del cluster
- C = *Separazione del cluster*. Minima dissimilarità tra un'osservazione del cluster e un'osservazione di un altro cluster

Tabella 5.2: Composizione dei cluster con K=4

Cluster	N	A	B	C
1	10823	35,01	23,83	1,78
2	1564	35,47	26,24	1,80
3	2988	268,92	32,87	13,67
4	3077	108,61	27,68	5,52

Tabella 5.3: Composizione dei cluster con K=5

Tabella 5.4: Composizione dei cluster con K=6

Cluster	N	A	B	C
1	9166	37,92943	24,74399	1,777528
2	2793	260,8011	29,75662	12,2222
3	3192	48,4233	28,65344	2,269314
4	30	37,59098	30,26568	1,761667
5	3161	38,30629	24,9613	1,795189
6	110	101,1647	38,90655	4,740992

La composizione dei cluster con K=4 appare come la più equidistribuita.

A questo punto, in base ai risultati ottenuti, si è deciso di operare un campionamento di alcune serie del database, al fine di meglio rappresentare, sia pure con un approccio grafico, che come si sa ha i suoi limiti, se nei quattro gruppi potessero evidenziarsi dei *trend* che mettessero in luce la deformazione del suolo interessato dalla rilevazione, nonché gli andamenti delle rilevazioni nelle zone oggetto di analisi.

In base al campionamento effettuato per *K=4*, per ciascun gruppo, è possibile visualizzare una serie di andamenti che di seguito vengono presentati.

Nel primo cluster si evidenzia una stabile ma debole subsidenza del suolo, cioè di un rallentato abbassamento del suolo (Figura 5.5).

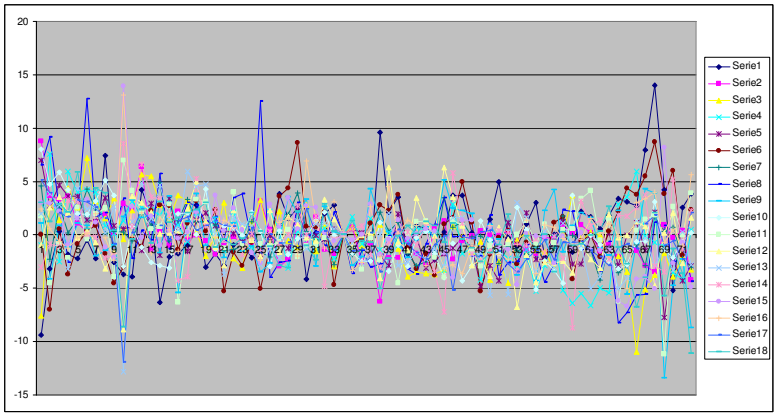


Figura 5.5: Andamento del cluster1 in base al campionamento

Nel secondo cluster si evidenzia una subsidenza variabile del suolo (Figura 5.6).

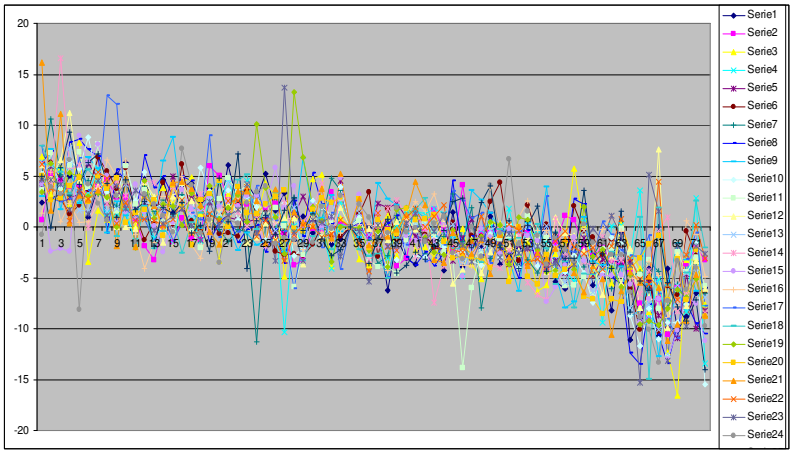


Figura 5.6: Andamento del cluster2 in base al campionamento

Nel terzo cluster si evidenzia una subsidenza costante del suolo (Figura 5.7).

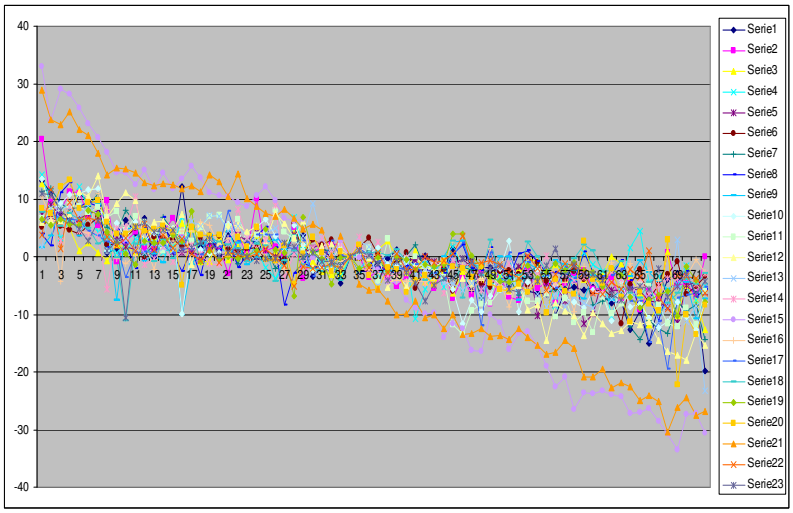


Figura 5.7: Andamento del cluster3 in base al campionamento

Nel quarto cluster si evidenzia una absidenza, ovvero di un innalzamento del suolo (Figura 5.8).

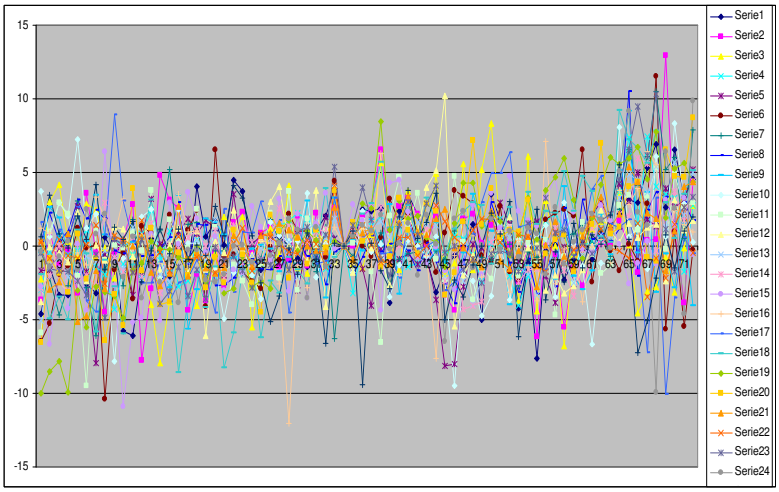


Figura 5.8: Andamento del cluster4 in base al campionamento

Verificata la coerenza dei cluster, in base al campionamento effettuato, l'analisi è proseguita, avvalendosi di tecniche di visualizzazione degli output

prodotti su opportune mappe che hanno consentito di effettuare una serie di considerazioni, evidenziando quanto accade al suolo nelle aree considerate.

Le mappe di seguito considerate fanno riferimento all'area dei Benevento e di Avellino e mettono in luce la coerenza della scelta del clustering con *K=4*, soprattutto sul territorio del Beneventano, relativamente alle quattro fasi, precedentemente discusse.

In particolare, la Figura 5.9, presenta la Mappatura del territorio di Benevento, riportando, su di essa, tutti i punti relativi al dataset completo che si riferiscono alla velocità media di rilevazione.

La Figura 5.10, invece, fa riferimento alla Mappatura del territorio di Benevento, riportando su di essa, però, solo i punti del dataset ridotto, in base al criterio di coerenza, che rilevano la velocità media. Un tale confronto è importante per meglio apprezzare il contributo della tecnica PS_TS all'analisi, che rendere, senza dubbio, più agevole la lettura dei punti e, quindi, l'interpretazione di ciò che accade al territorio.

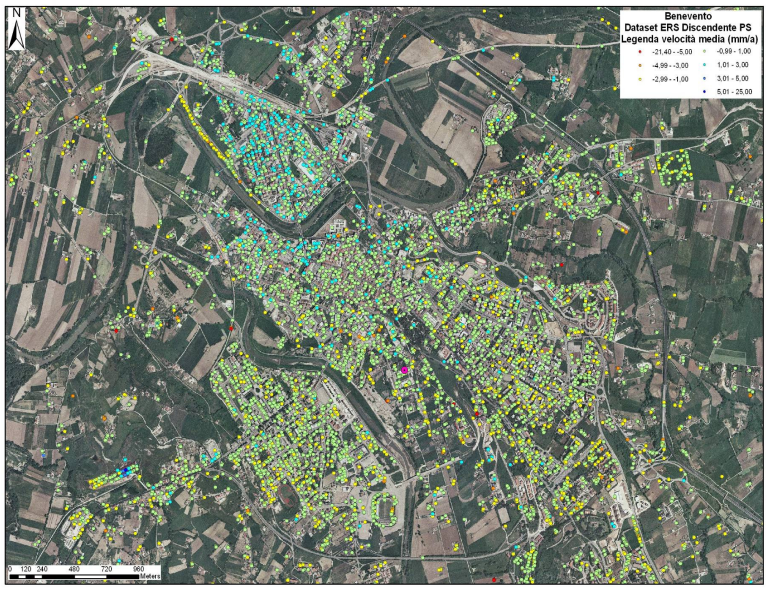


Figura 5.9: Mappatura dataset ERS Discendente PS nell'area di Benevento sulla velocità media

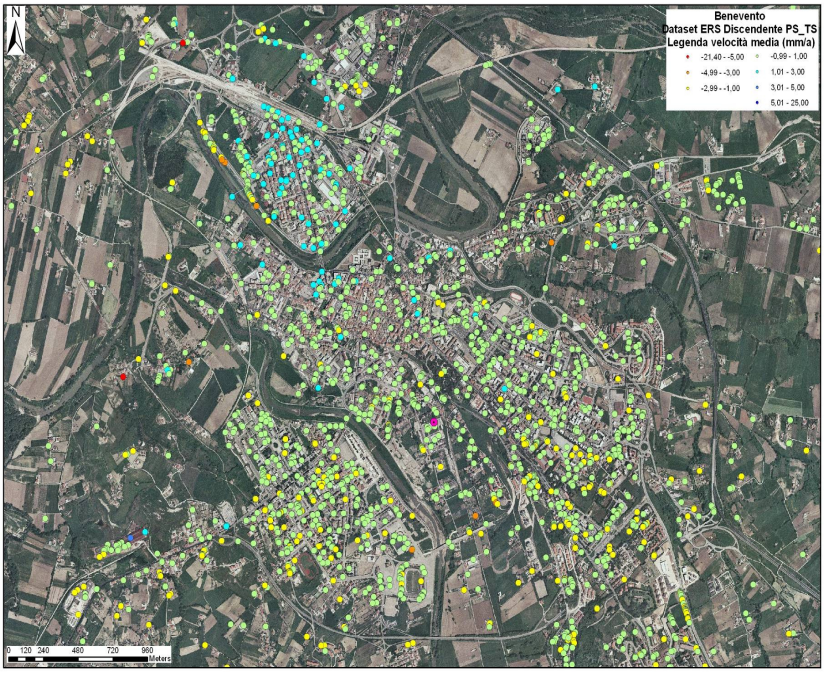


Figura 5.10: Mappatura dataset ERS Discendente PS_TS nell'area di Benevento sulla velocità media

Per poter meglio interpretare la suddivisione dei cluster si fa ricorso alla Figura 5.11 che riporta la mappa dell'Area di Benevento, su cui emergono i punti della serie storica del dataset ridotto, e il risultato della clusterizzazione mediante l'algoritmo CLARA con *K=4*, in cui, appunto, sono evidenti le quattro fasi emerse:

- stabile-debole subsidenza;
- variabile subsidenza;
- costante subsidenza;
- absidenza.

L'ultima fase sembra concentrarsi, per la maggior parte, nell'area nord occidentale ed è rappresentata dai punti in blu e quindi appare essere la prevalente.

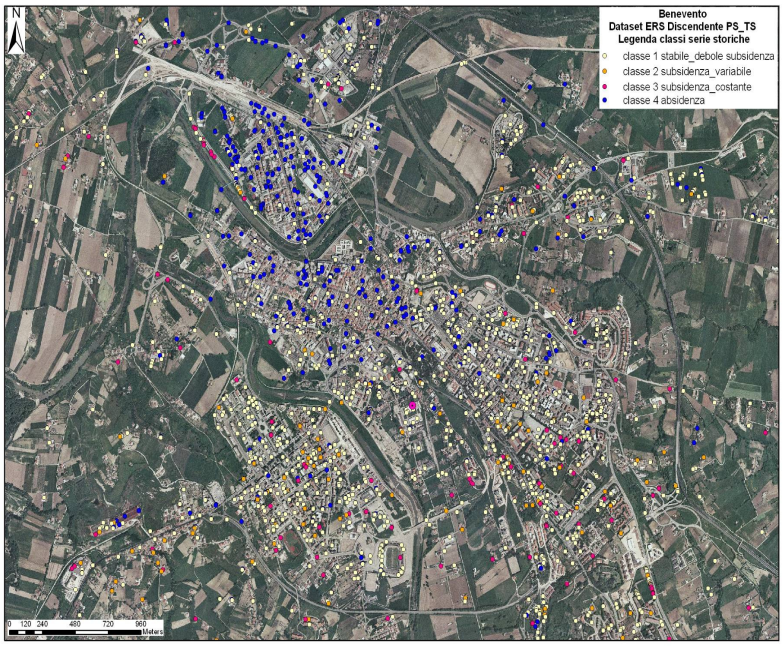


Figura 5.11: Mappatura dataset ERS Discendente PS_TS nell'area di Benevento sui quattro cluster della serie storica

La Figura 5.12 non è altro che la 5.11 su cui sono state apportate le linee di demarcazione settoriale, per meglio evidenziare le classi predominanti sul territorio osservato, che non si sparpagliano in maniera significativa, come si è visto per il cluster 4 (zona di absidenza del suolo).

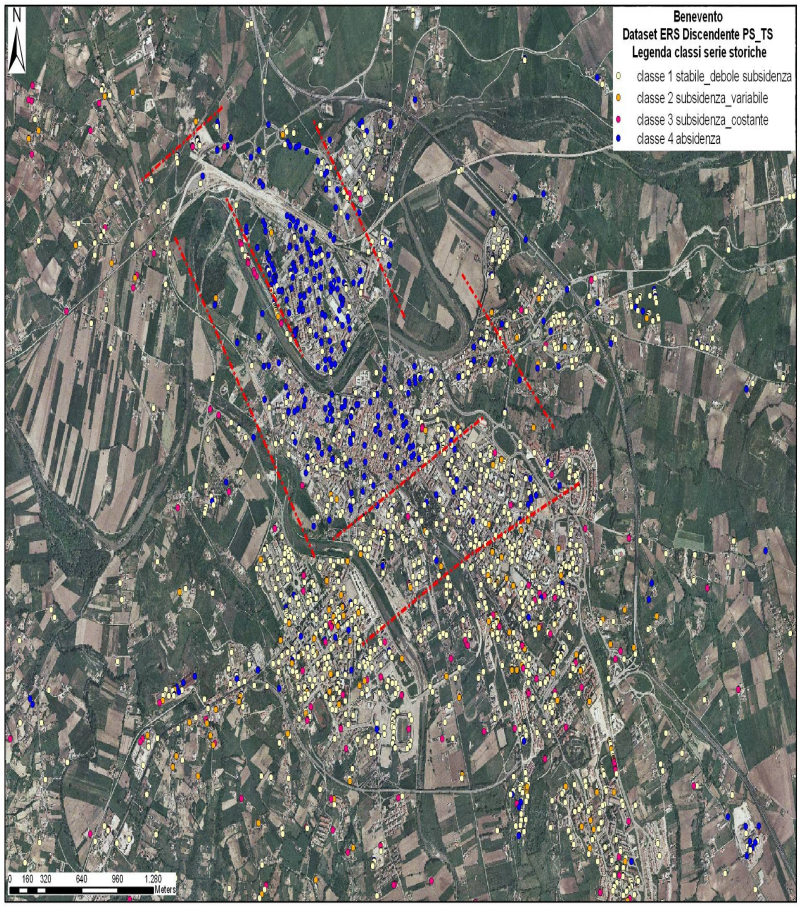


Figura 5.12: Mappatura dataset ERS Discendente PS_TS nell'area di Benevento sui quattro cluster della serie storica con linee settoriali

Analogo criterio è stato utilizzato sulla Figura 5.10, dando luogo alla Figura 5.13 dove si apportano le linee di demarcazione settoriale, anche alla visualizzazione del territorio, in cui emergono i punti di rilevazione della velocità media.

Stesso criterio di analisi è stato utilizzato per l'area di Avellino (Figure 5.14-5.16) per la quale si presentano, rispettivamente, solo:

- la mappatura che evidenzia i 4 cluster sulla serie storica proveniente dal dataset ridotto;

- la mappatura su cui vengono rappresentati i punti relativi alla velocità media, in base al dataset ridotto;
- la mappatura su cui vengono rappresentati i punti relativi alla velocità media, in base all'intero dataset.

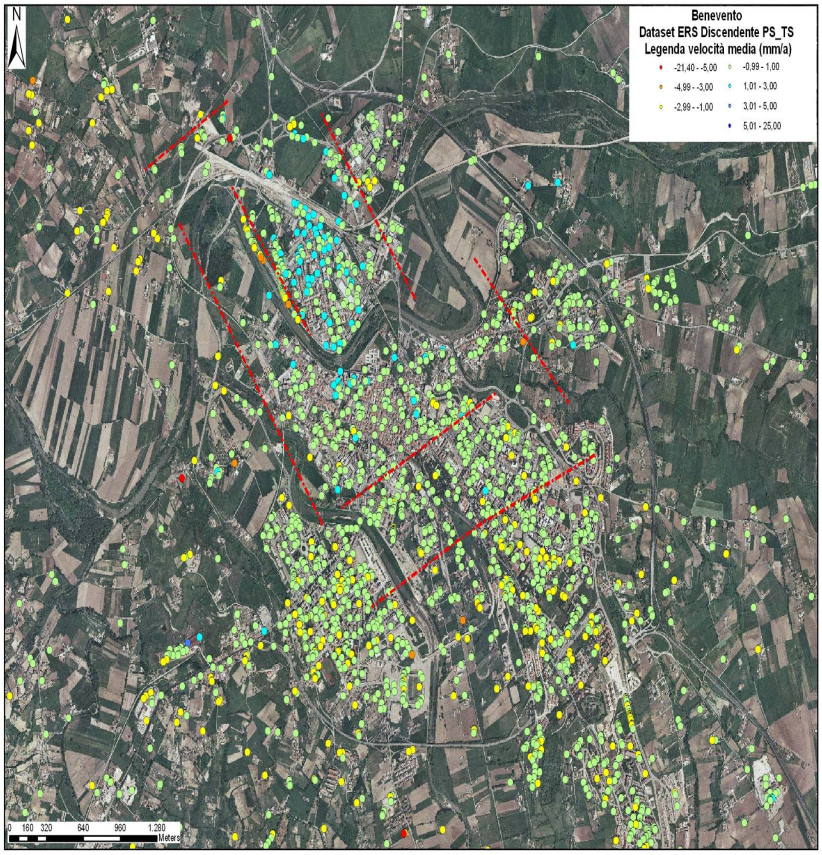


Figura 5.13: Mappatura dataset ERS Discendente PS_TS nell'area di Benevento sulla velocità media, con linee settoriali

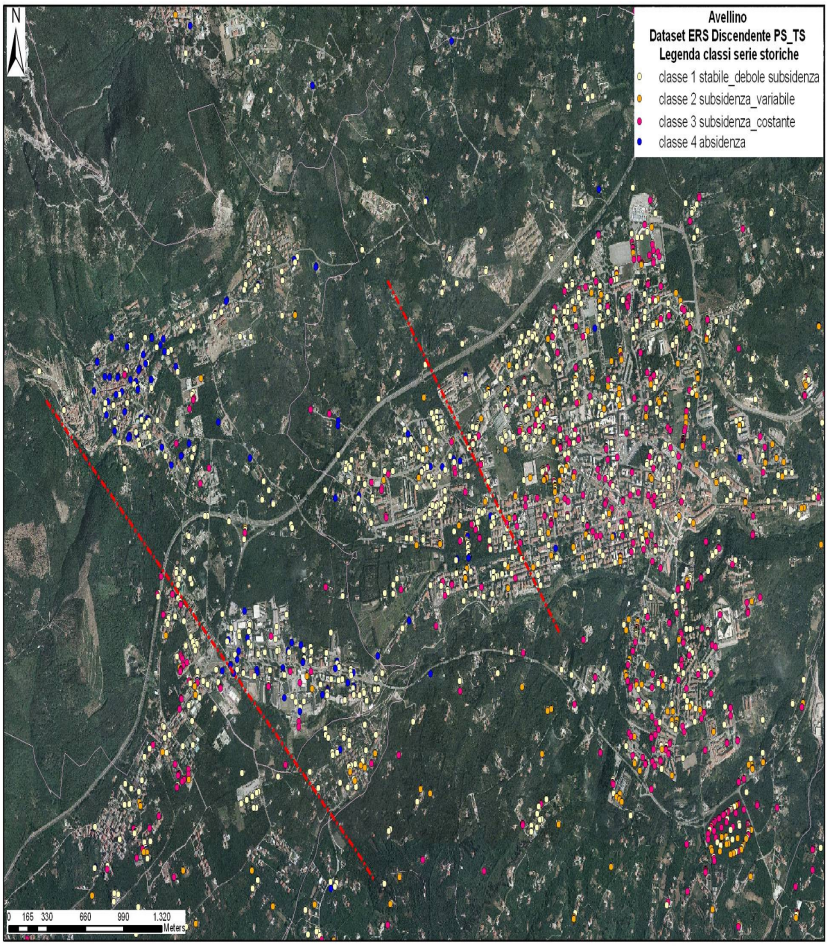


Figura 5.14: Mappatura dataset ERS Discendente PS_TS nell'area di Avellino sui quattro cluster della serie storica con linee settoriali

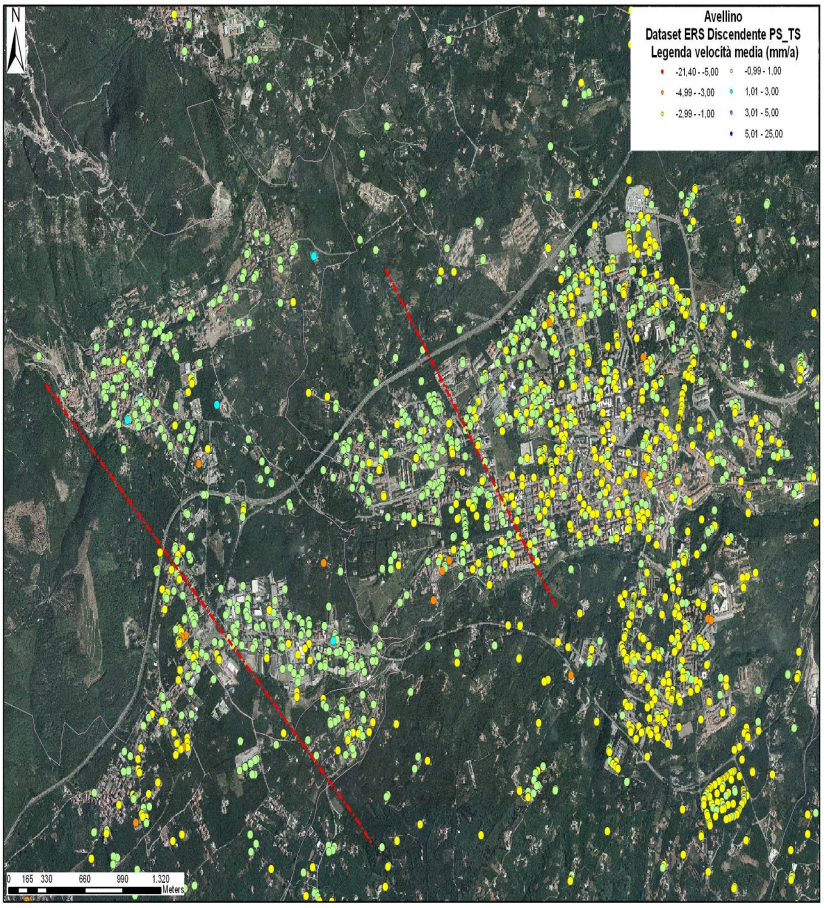


Figura 5.15: Mappatura dataset ERS Discendente PS_TS nell'area di Avellino sulla velocità media, con linee settoriali

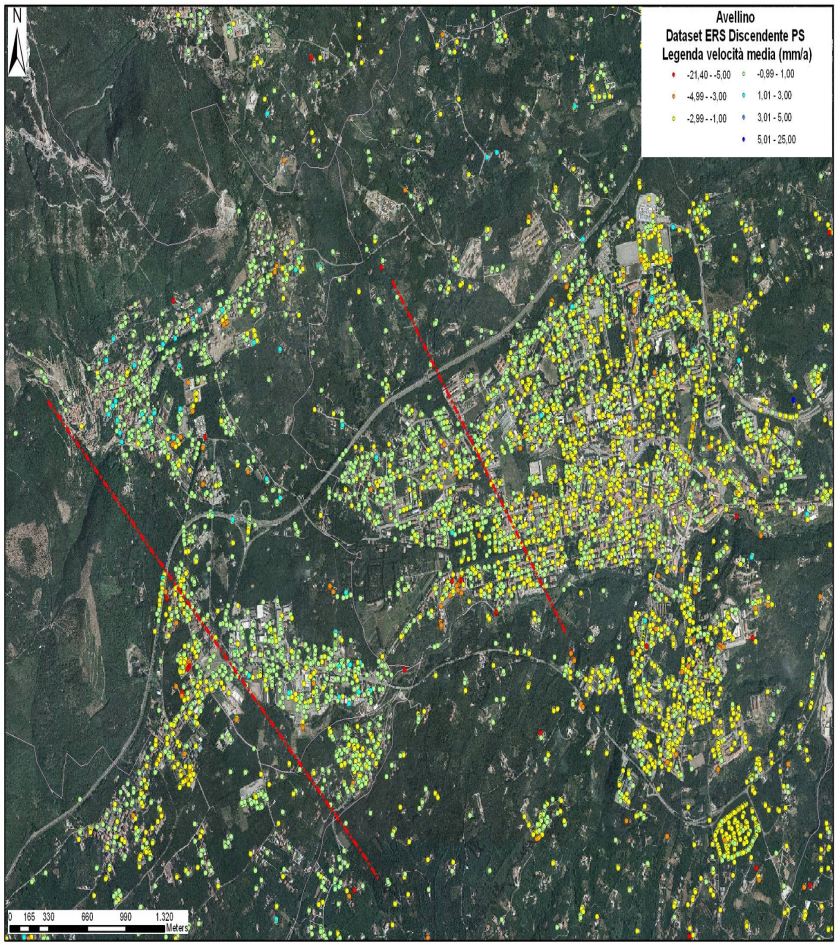


Figura 5.16: Mappatura dataset ERS Discendente PS nell'area di Avellino sulla velocità media con linee settoriali

Da quanto, finora esposto, appare evidente, quanto, il criterio di selezione in base alla coerenza, abbia apportato all'analisi un valido contributo alla lettura del dato e all'interpretazione delle mappe visualizzate, in base al tipo di output riscontrato.

Conclusioni e ulteriori sviluppi

Il Temporal Data Mining è un settore in rapida espansione, che presenta recentissimi risultati scientifici nei domini temporali. Nel presente lavoro di tesi, si è provveduto ad una critica dello stato dell'arte, segnalando molti nuovi metodi di analisi o prototipi, sviluppati di recente.

Il dominio a cui ci si è riferiti presenta, appunto, la caratteristica della multidisciplinarietà, avendo molte connessioni nei più disparati settori di conoscenza scientifica, permettendo, così, di percorrere uno studio esaustivo, che comprendesse contributi che vanno al di là del campo di applicazione di questo lavoro.

La discussione si è concentrata, in particolare, su quelli che sono i principali ambiti scientifici legati al Temporal Data Mining, potendo, così risalire agli obiettivi concreti, rispetto alle attuali tecnologie e possibilità computazionali.

Nell'ambito delle tecniche di TDM, appare abbastanza esaustiva la panoramica scientifica presente, relativamente ai metodi di estrazione di sequenze temporali, mediante procedure di clustering, pur riuscendo ad intravedere dei punti su cui la ricerca dovrebbe intervenire, per risolvere le debolezze di alcuni algoritmi.

La critica si è concentrata, in particolare, sugli aspetti della similarità e delle metriche, tipici di contesti temporali, che hanno, quindi, rappresentato l'elemento ispiratore di tale lavoro.

Partendo da tali premesse, si è evidenziato che, allo stato attuale, già esiste un numero consistente di algoritmi legati alla risoluzione di problemi di clustering temporale, per grosse moli di dati; tuttavia, l'utilizzatore delle stesse, necessita, talvolta, di essere guidato nella scelta di quelli che possano essere i più performanti per ciascuna esigenza empirica.

Dal momento che l'esigenza di dover trattare dati spazio-temporali è nata solo di recente, le tecniche esistenti non sono ancora del tutto soddisfacenti, a causa del fatto che non riescono, appunto, a fornire meccanismi di ragionamento espliciti e flessibili di cui l'informazione spazio-temporale necessita.

Uno dei limiti, al riguardo, sta nel fatto che non è presente un meccanismo ad alto livello che renda semplice l'interrogazione del database. Per raggiungere un alto grado di *performance*, infatti, l'utente deve essere un esperto del sistema e deve essere in grado di usare primitive efficienti e di basso livello.

Una questione molto importante, inoltre, nell'ambito del clustering temporale riguarda la scalabilità, cioè la possibilità di elaborare una maggiore numerosità o dimensionalità di dati, mantenendo l'efficacia e l'efficienza. In termini teorici, per complessità di un algoritmo si indica la funzione che associa alla dimensione del problema, il costo della sua risoluzione in base ad una misura quale, ad esempio, il tempo di esecuzione o lo spazio di memoria occupato. Con "classe di complessità" si indica la più piccola funzione, tra quelle che approssimano la misura ottenuta, secondo la metrica scelta. Per comprendere ciò, si pensi a quanto detto circa la complessità del *K-means*: dati K cluster, N pattern e T, numero massimo di iterazioni. Spesso K non è noto a priori, per cui possono risultare necessarie varie prove, per capire quale sia il suo valore ottimale.

Inoltre, un'altra, rilevante, problematica, comune ai contesti di clustering nel TDM, riguarda la complessità spaziale, cioè l'ingombro dei dati. La memoria principale, in genere, non sarà in grado di mantenere tutti i dati; occorre, quindi, fare in modo da minimizzare lo spostamento delle pagine dalla memoria di massa, e sono, relativamente, pochi gli algoritmi, allo stato attuale

che riescono nell'addestramento *on line*. Per fare ciò si considerano, attualmente, queste tre possibili strategie:

1. Si memorizzano i dati nella memoria di massa e si effettua il clustering su un sottoinsieme dei dati; infine, si aggiungono i pattern non elaborati al cluster più vicino (approccio *divide et impera*).
2. Si memorizzano i dati in memoria di massa e si trasferiscono uno alla volta in memoria principale, le informazioni sulla struttura dei cluster vengono anch'esse mantenute in memoria principale.
3. Si effettua l'elaborazione su una macchina parallela. In questo caso, i benefici che si ottengono, sono in funzione della struttura dell'algoritmo.

In conclusione, l'analisi condotta sulla tecniche di TDM, ha consentito di rilevare alcuni aspetti salienti dei dati temporali su cui intervenire, che risiedono: nella irregolarità, nell'asincronismo, nell'eterogeneità.

Inoltre, Il presente lavoro di tesi, vuol essere di auspicio per lo scrivente, affinché egli possa proseguire su tale filone di ricerca, non solo portando, avanti la parte applicativa (capitolo 5), potendo, così, fornire confronti utili alla sperimentazione nell'ambito degli studi sulla deformazione del suolo, in base ai dati radar rilevati sull'intero territorio campano; ma anche per valutare l'impiego di ulteriori tecniche da implementare, sulla scorta degli algoritmi di clustering presentati. Si ritiene, infatti, che le problematiche, finora trattate, possono essere, senza dubbio, dei filoni su cui la ricerca può e deve ancora fare molto, per consentire all'utilizzatore di apprendere, in modo sempre più immediato la conoscenza implicita nei dataset temporali di grandi dimensioni.

Appendice A

Si riportano i principali passi dell'algoritmo usato per l'applicazione sviluppata nel Capitolo 5 con i commenti relativi. Si tratta dell'algoritmo CLARA, sviluppato da Kaufman, L. and Rousseeuw, P.J. (1990), fatto girare con linguaggio MatLab™.

```
The CLARA algorithm is designed for large datasets
He returns a list representing a clustering of the
data into kclus clusters following.

Required input arguments:
   o x:     Data     matrix     (rows=observations,
     columns=variables)
   o kclus: Number of desired clusters
   o vtype: Variable type vector (length equals
     number of variables)

Possible values are:
   o Asymmetric binary variable (0/1)
   o Nominal variable (includes symmetric binary)
   o Ordinal variable
   o Interval variable

Optional input arguments:
   o metric: Metric to be used (default euclidian
     (eucli) or mixed (mixed))

Possible values are:
   o 'eucli' Euclidian (all interval variables)
   o 'manha' Manhattan
   o 'mixed' Mixed (not all interval variables)
```

- We define:
- nsamp: Number of samples to be drawn from the dataset
- sampsize: Number of observations in each sample (should be higher than the number of clusters and lower than the number of observations)
- I/O:
- result=clara(x,kclus,vtype,'eucli',5,40+2*kclus)

Example
(subtracted from the referenced book)

```
load obj200.mat
result=clara(obj200,3,[4 4])
```

The output of CLARA is a structure containing:

- result.dysobs: dissimilarities for each observation with the medoids
- result.metric: used metric
- result.number: number of observations
- result.idmed: Id of medoid observations
- result.ncluv: A vector with length equal to the number of observations, giving for each observation the number of the cluster to which it belongs
- result.obj: Objective function for the best subsample
- result.clusinf: Matrix, each row gives numerical information for one cluster. These are the cardinality of the cluster (number of observations), the maximal and average dissimilarity between the observations in the cluster and the cluster's medoid, the diameter of the cluster (maximal dissimilarity between two observations of the cluster), and the separation of the cluster (minimal dissimilarity between an observation of the cluster and an observation of another cluster).
- result.sylinf: Matrix based on the best subsample, with for each observation i of this subsample the cluster to which i belongs, as well as the neighbor cluster of I (the cluster, not containing i, for which the average dissimilarity between its observations

```
      and i is minimal), and the silhouette width of
      i.

This function is part of LIBRA: the Matlab Library
for Robust Analysis,available at:

http://wis.kuleuven.be/stat/robust.html
```

Appendice B

B1 Sistemi di telerilevamento radar satellitari

I sistemi radar satellitari forniscono immagini elettromagnetiche (a frequenze comprese tra 500 MHz e 10 GHz) della superficie terrestre con risoluzione spaziale di qualche metro.
Rispetto ai più noti sistemi ottici operano con continuità, potendo acquisire dati, in presenza di copertura nuvolosa, sia di giorno che di notte.
Il principio di funzionamento dei sistemi RADAR (acronimo di RAdio Detecting And Ranging) è il seguente: un apparecchio trasmittente illumina lo spazio circostante (ed eventuali oggetti) con un'onda elettromagnetica che, incidendo sulla superficie terrestre, subisce un fenomeno di riflessione disordinata (ossia di diffusione, definita *scattering*). Una parte del campo diffuso torna verso la stazione trasmittente, equipaggiata anche per la ricezione, dove vengono misurate le sue caratteristiche. Il dispositivo è in grado di individuare il bersaglio elettromagnetico (detecting) e, misurando il ritardo temporale tra l'istante di trasmissione e quello di ricezione, di valutare la distanza (ranging) a cui è posizionato, localizzandolo in modo preciso lungo la direzione di puntamento dell'antenna (direzione di range).
La direttività dell'antenna utilizzata per trasmettere e ricevere il segnale radar, cioè la selettività nell'illuminazione dello spazio circostante, consente di localizzare l'oggetto, anche lungo l'altra dimensione (detta di *azimuth*).

Quanto più grande è l'antenna, tanto più stretta è la sua impronta e, di conseguenza, tanto meglio viene localizzato il bersaglio.
Chiaramente, ciò avviene a scapito dell'estensione dell'area illuminata (Figura B.1).

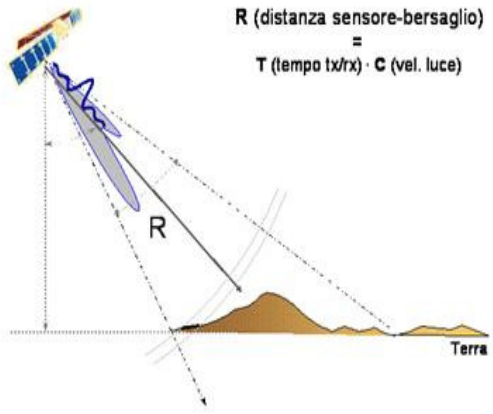


Figura B.1: Rappresentazione della direttività in entrata e in uscita dei segnali radar in base al bersaglio

Per ovviare a questo inconveniente, le antenne direttive usate per i radar militari e per applicazioni di aviazione civile ruotano, in modo da "spazzare" tutta l'area circostante alla loro posizione.

B2 Sistemi Synthetic Aperture Radar

I sistemi Radar ad Apertura Sintetica (Synthetic Aperture Radar, SAR) sono dispositivi di telerilevamento attivo, operanti nell'intervallo delle microonde (con frequenze comprese tra 1-10 GHz) e capaci di sintetizzare un'antenna di grandi dimensioni, osservando lo stesso bersaglio a terra da diversi angoli di vista. Per ottenere immagini ad alta risoluzione spaziale, sarebbero infatti necessarie antenne di grandi dimensioni, con ovvi problemi di messa in orbita.

Come mostrato nella Figura B.2, il SAR, lungo la sua traiettoria, osserva ripetutamente la stessa area e sintetizza un'antenna di dimensioni più grandi, combinando coerentemente i dati acquisiti nelle posizioni successive e ottenendo così un'elevata risoluzione nella direzione di *azimut* (parallela alla direzione orbitale). Il SAR è in genere montato a bordo di una piattaforma mobile, aereo o satellite.

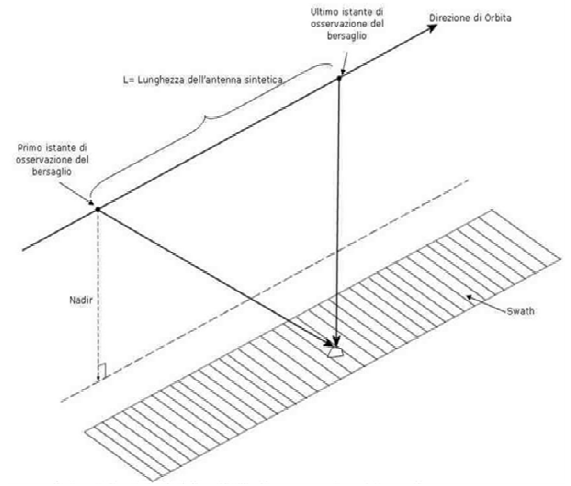


Figura B.2: Funzionamento dei SAR

L'idea alla base del SAR consente di risolvere brillantemente il compromesso risoluzione–estensione dell'area osservata. Combinando coerentemente i dati acquisiti dal sensore nelle posizioni successivamente occupate, si sintetizza un'antenna fittizia di grandi dimensioni detta, appunto, ad apertura sintetica. È proprio questo procedimento a garantire un'elevata risoluzione anche nella direzione di *azimuth*.

Poiché il sistema illumina lo spazio circostante, con radiazioni elettromagnetiche proprie, è definibile: sistema attivo. Non è richiesta, infatti, illuminazione solare e le frequenze utilizzate dal radar penetrano attraverso le nuvole, evitando, così, problemi di acquisizione dei sistemi ottici.

Il sistema di funzionamento di un sistema SAR, e, in generale, di una piattaforma radar per immagini, non è molto diverso dal classico radar per avvistamento: un impulso elettromagnetico viene emesso da un'antenna e il corrispondente segnale riflesso (o retrodiffuso) dai bersagli osservati al suolo, viene rilevato dalla stessa antenna, dopo un tempo proporzionale alla distanza dei bersagli dal radar, e con un'intensità proporzionale all'energia retrodiffusa in direzione del radar. La direzione parallela all'orbita è detta, quindi, di azimuth e coincide approssimativamente con la direzione Nord-Sud. La direzione della congiungente sensore-bersaglio (perpendicolare all'orbita ed inclinata di un angolo *teta*, o di *off-nadir*, rispetto alla verticale) è detta slant range o Line Of Sight - LOS (Figura B.3).

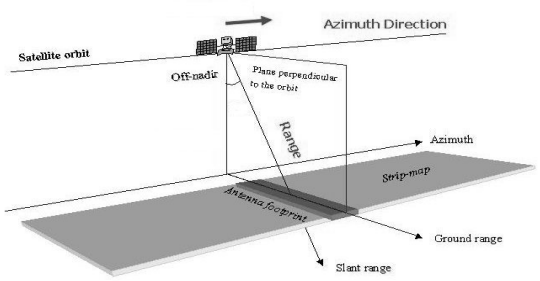


Figura B.3: Line Of Sight

La modalità di acquisizione dei sistemi radar, non perpendicolare al suolo ma secondo un angolo di vista *teta*, dà origine a differenti deformazioni prospettiche (Figura B.4) visibili nelle immagini SAR. In funzione della topografia del terreno, si distinguono in:

- *foreshortening*: si verifica quando la pendenza del terreno tende ad essere perpendicolare alla congiungente sensore-bersaglio (pendenza positiva pari all'angolo di *off-nadir*); in questi casi il contributo di più punti si concentra in poche celle producendo pixel molto luminosi nell'immagine di ampiezza.
- *layover*: si verifica quando la pendenza del terreno è maggiore dell'angolo di *off-nadir*; questo produce una forte distorsione

nell'immagine, impedendo la corretta interpretazione del segnale ed ogni analisi quantitativa.

- *shadowing*: si verifica quando alcune zone non possono essere illuminate dall'impulso radar, perché schermate da altri oggetti; si producono nell'immagine di ampiezza aree molto scure (in ombra).

Come illustrato in figura, la direzione della congiungente sensore-bersaglio (perpendicolare all'orbita ed inclinata di un angolo *teta* rispetto alla verticale) è detta *slant range* (o più semplicemente *range*). La sua proiezione al suolo è detta direzione di *ground range*. Per i satelliti ESA-ERS, la risoluzione in range vale circa 8 metri, mentre la corrispondente risoluzione in *ground range*, circa 20 metri (dato che il valore dell'angolo di *off-nadir* è più o meno 23 gradi al centro della scena).

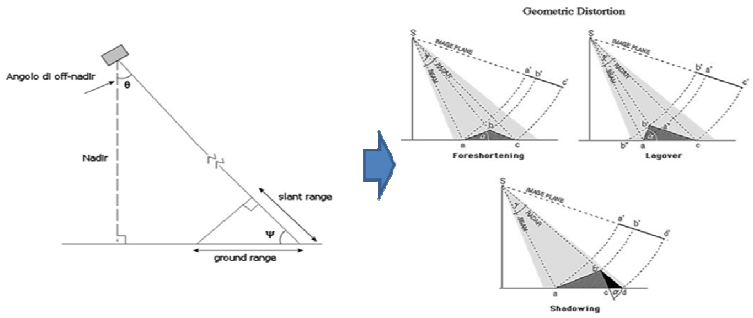
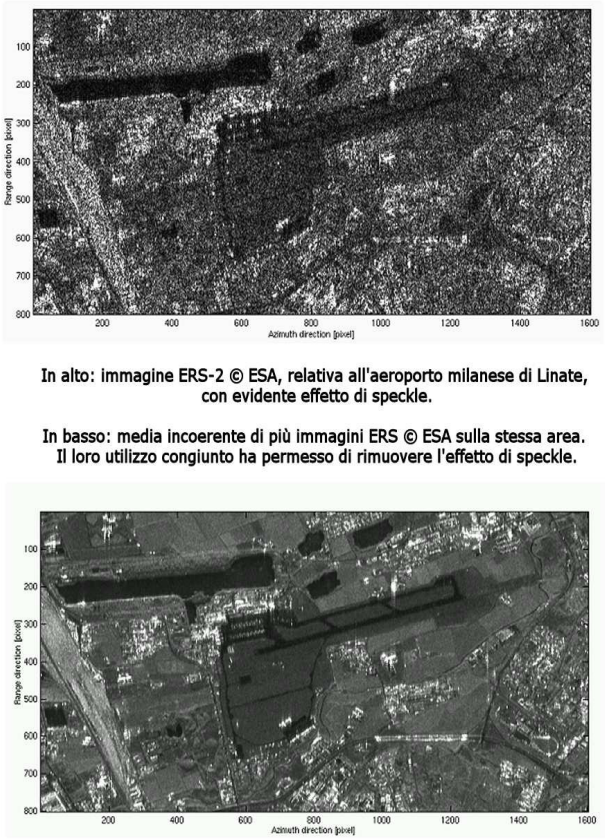


Figura B.4: Differenti deformazioni prospettiche in funzione della topografia del terreno

Oltre alle deformazioni prospettiche, le immagini radar sono affette dal cosiddetto rumore di *speckle*. Un oggetto, che nelle immagini ottiche risulta caratterizzato da un valore pressoché costante di riflettività appare, invece, "maculato" nell'immagine radar, con variazioni vistose dell'ampiezza del segnale. Questo effetto è tipico di tutti i sistemi coerenti, in cui la fase del segnale gioca un ruolo fondamentale, ma può essere facilmente rimosso mediando più immagini SAR della stessa area (Figura B.5).



In alto: immagine ERS-2 © ESA, relativa all'aeroporto milanese di Linate, con evidente effetto di speckle.

In basso: media incoerente di più immagini ERS © ESA sulla stessa area. Il loro utilizzo congiunto ha permesso di rimuovere l'effetto di speckle.

Figura B.5: Immagini radar caratterizzate dal rumore di speckle

I sistemi SAR, montati su piattaforme satellitari, acquisiscono (grazie alla combinazione del moto lungo l'orbita con il moto rotazionale terrestre) immagini in due diverse geometrie: *ascendente* e *discendente*. Il passaggio ascendente coincide approssimativamente con l'orbita Sud-Nord del satellite e consente di illuminare l'area di interesse da Ovest. Nel passaggio discendente, viaggiando da Nord a Sud, il sensore illumina il bersaglio da Est.

Le immagini SAR sono matrici di numeri complessi definiti dalle grandezze di ampiezza e fase (Figura B.6) del segnale retro-diffuso.

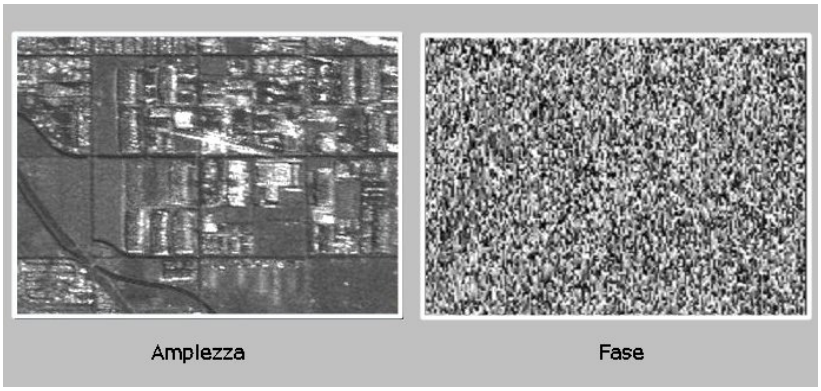


Figura B.6: Ampiezza e fase di un'immagine SAR

Ad ogni singolo pixel dell'immagine è associato un valore di ampiezza e un valore di fase: l'ampiezza individua la quantità di campo elettromagnetico, mentre la fase dipende da diversi fattori, tra cui la distanza sensore-bersaglio. L'ampiezza è funzione sia delle caratteristiche geometriche (tipicamente la rugosità) che fisiche degli oggetti. In generale, rocce esposte o aree urbanizzate mostrano un'ampiezza elevata, mentre specchi d'acqua calma mostrano un'ampiezza modesta perché la maggior parte dell'energia incidente viene riflessa specularmente lontano dl radar. La fase di una singola ripresa SAR non è in genere di alcuna utilità pratica. Tuttavia, assume una notevole importanza quando si confrontano immagini SAR riprese da angoli di vista differenti o in tempi successivi.

Nella fase di un'immagine SAR si possono distinguere quattro contributi principali: un termine dovuto alla riflettività del bersaglio (dipendente dal materiale e dalla sua geometria), un termine funzione della distanza sensore-bersaglio (usualmente definito *propagatore*), un contributo dovuto all'atmosfera e un rumore proprio del sistema di acquisizione. L'obiettivo delle tecniche interferometriche è quello di isolare gli effettivi contributi di fase dovuti al movimento del bersaglio e non imputabili a disturbi, ovvero stimare accuratamente la differenza di cammino ottico dell'onda

elettromagnetica trasmessa in due successive acquisizioni e retrodiffusa al sensore dal bersaglio a terra.

I dati SAR vengono utilizzati nelle applicazioni più varie:

- agricoltura (identificazione e monitoraggio di coltivazioni, estensione delle piantagioni, etc.);
- idrologia (stima dell'umidità del suolo);
- geologia (vulcanica, tettonica, strutturale);
- oceanografia;
- rilievo e zonazione del territorio;
- monitoraggio di foreste;
- monitoraggio di aree urbane.

B3 Satelliti ERS dell'Agenzia Spaziale Europea

Nel maggio 1991 l'Agenzia Spaziale Europea (ESA) lanciò il primo SAR europeo a bordo del satellite ERS-1 (Figura B.7), seguito nel 1995 dal gemello ERS-2 posto sulla sua stessa orbita ma con un ritardo di un giorno.

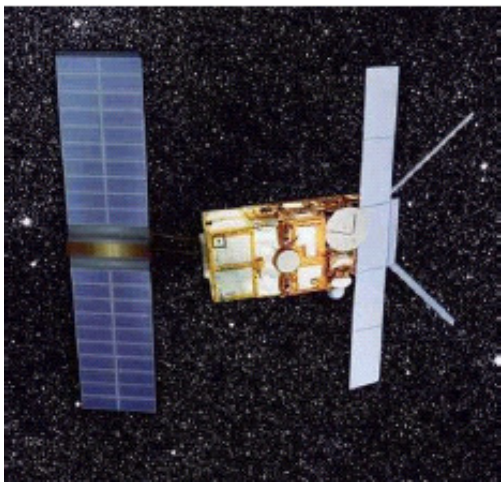


Figura B.7: Immagine del satellite ERS-1

I satelliti ERS seguono orbite elio sincrone lievemente inclinate rispetto ai meridiani, illuminando, da una quota attorno a 780 Km, una striscia di terreno (swath) larga circa 100 km con un sistema radar SAR operante nel dominio delle microonde alla frequenza di 5.3 GHz, ovvero con una lunghezza d'onda λ pari a 5.66 cm, caratteristica fondamentale per poter apprezzare movimenti millimetrici.

La stessa orbita nominale viene ripercorsa ogni 35 giorni (revisiting time), consentendo così di acquisire dati relativi alla stessa scena al suolo, in tempi differenti.

Grazie alla scelta dell'ESA di acquisire continuamente i dati a partire dal 1992, su vaste aree del pianeta, sono oggi disponibili i dataset dell'ultimo decennio composti da un'immagine radar ogni 35 giorni. Questi costituiscono un'informazione storica di enorme rilevanza, permettendo di studiare l'evoluzione della fase per ciascuna acquisizione e di ricostruire la storia delle deformazioni.

La direzione parallela all'orbita è detta azimuth e coincide approssimativamente con la direzione Nord-Sud. La risoluzione (ovvero la capacità di riconoscere come distinti due bersagli) in azimuth vale circa 5 m.

La direzione della congiungente sensore-bersaglio (perpendicolare all'orbita ed inclinata di un angolo θ - detto *off-nadir* - rispetto alla verticale pari mediamente a 23°) è detta *slant range* (o più semplicemente range) oppure Line Of Sight (LOS). La risoluzione in range vale circa 8 m.

Le immagini radar si sviluppano pertanto lungo le direzioni di *range* e di *azimuth*, dette usualmente coordinate SAR. In Figura B.8 è rappresentata schematicamente la geometria di acquisizione dei sistemi SAR-ERS.

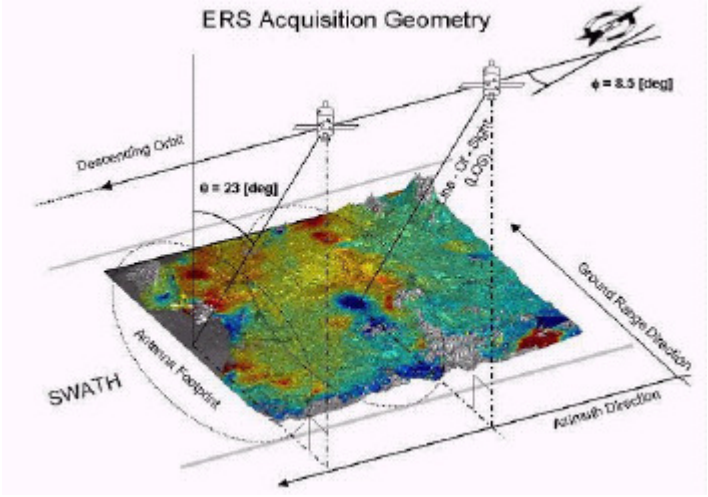


Figura B.8: Geometria di acquisizione SAR delle piattaforme ERS-1e ERS-2

Le immagini radar sono matrici di numeri complessi definiti dalle grandezze di ampiezza e di fase.

In esse: l'ampiezza individua la quantità di campo elettromagnetico retrodiffusa verso il satellite, la fase costituisce l'informazione chiave per le applicazioni interferometriche volte all'identificazione di aree soggette a fenomeni di movimento superficiale. La fase, inoltre, dipende da diversi fattori, tra cui la distanza sensore-bersaglio (Figura B.9).

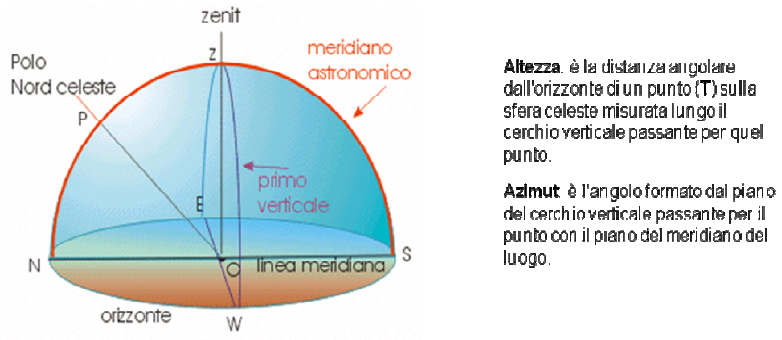


Figura B.9: Rappresentazione di un bersaglio mediante altezza e azimuth

In fase di acquisizione, ogni bersaglio a terra è colpito da più impulsi elettromagnetici emessi dal sensore lungo la sua orbita. ESA fornisce, su un opportuno supporto (CD), l'eco degli impulsi radar così come sono stati ricevuti dal satellite. Si parla, in questo caso, di dati grezzi (*raw data*).

Le immagini radar, propriamente dette, nascono solo a valle di un algoritmo di focalizzazione, che permette di associare alle varie celle di risoluzione (pixel), il relativo contributo di energia retrodiffusa: ad ogni elemento della matrice corrisponde una zona a terra di 20 per 4 m circa (su terreno piano). Ogni supporto contiene una quantità di dati relativa ad un'area di 100x100 Km (10000 Km2).

Durante i vari passaggi lungo la stessa orbita i satelliti si discostano leggermente dalla traiettoria nominale, di fatto vi sono delle variazioni nell'ordine delle centinaia di metri descritte dal parametro baseline geometrico (o normale); di conseguenza la geometria di acquisizione per la stessa zona varia di volta in volta di angoli θ leggermente diversi, creando matrici di pixel non corrispondenti alla medesima cella di risoluzione al suolo. Per effettuare l'analisi è necessario che, a pixel omologhi nelle varie immagini, corrisponda la stessa cella di risoluzione; si procede, quindi, con una fase di elaborazione dei dati detta di registrazione (o di ricampionamento). Operativamente, tra tutte le acquisizioni, si sceglie un'immagine come riferimento (detta *master*); tutte le rimanenti (dette *slave*), vengono ricampionate sulla geometria dell'immagine master, grazie ad un opportuno modello, in modo da ottenere la stessa griglia di riferimento per tutti i passaggi del satellite. Il modello utilizzato permette di compensare sia una rotazione sia una traslazione indotta sulle immagini a causa del differente angolo di vista.

B4 Analisi interferometrica

La generazione degli interferogrammi differenziali costituisce il primo passo dell'elaborazione di un'immagine SAR e si ottiene mediante la moltiplicazione complessa ("battimento") tra tutte le coppie formate dall'immagine master con le immagini slave, ottenendo così N-1 differenziali da N immagini del dataset.
L'informazione che si ottiene dagli interferogrammi differenziali racchiude ora diverse componenti: il movimento del bersaglio a terra, la topografia locale e il contributo dell'atmosfera.

B2.1 L'Interferometria radar differenziale (DInSAR)

La tecnica tradizionale per lo studio di dati SAR è l'interferometria differenziale (Differential SAR Interferometry - DInSAR).
Si è già detto che le immagini acquisite dai sistemi SAR sono matrici di numeri complessi definiti dalle grandezze di ampiezza e fase. Ogni valore corrisponde a un punto al suolo: l'ampiezza individua la quantità di campo elettromagnetico retro-diffusa verso il satellite, mentre la fase dipende da diversi fattori, tra cui la distanza sensore-bersaglio. Proprio la fase costituisce l'informazione chiave per le applicazioni interferometriche, volte all'identificazione di aree soggette a fenomeni di movimento superficiale.
Infatti, l'interferometria tradizionale si basa sull'analisi dell'evoluzione del valore di fase tra due distinte acquisizioni, in modo da mettere in luce eventuali differenze riconducibili a fenomeni di deformazione, topografia o disturbi atmosferici. Il valore di fase di un'immagine SAR è costituito da diversi contributi relativi alla riflettività del bersaglio (dipendente dal materiale e dalla sua geometria), alla presenza dell'atmosfera, alla distanza sensore-bersaglio e ad un inevitabile rumore proprio del sistema di acquisizione.

L'obiettivo della tecnica interferometrica è quello di isolare gli effettivi contributi di fase dovuti al movimento del bersaglio e non imputabili ad altri disturbi, ovvero stimare accuratamente la differenza di cammino ottico dell'onda elettromagnetica trasmessa in due successive acquisizioni e retro-diffusa dal medesimo bersaglio a terra. Sottraendo la fase di un'immagine a quella dell'altra, si genera un interferogramma, di cui si riporta un esempio nella Figura B.10. Se non avvengono particolari cambiamenti nel periodo tra le due acquisizioni, i contributi dovuti alla riflettività si elidono e la fase dell'interferogramma dipende, con buona approssimazione, solo dalla distanza sensore-bersaglio e quindi da eventuali movimenti intercorsi tra le due acquisizioni (a parte i contributi spuri dovuti all'atmosfera e al rumore).

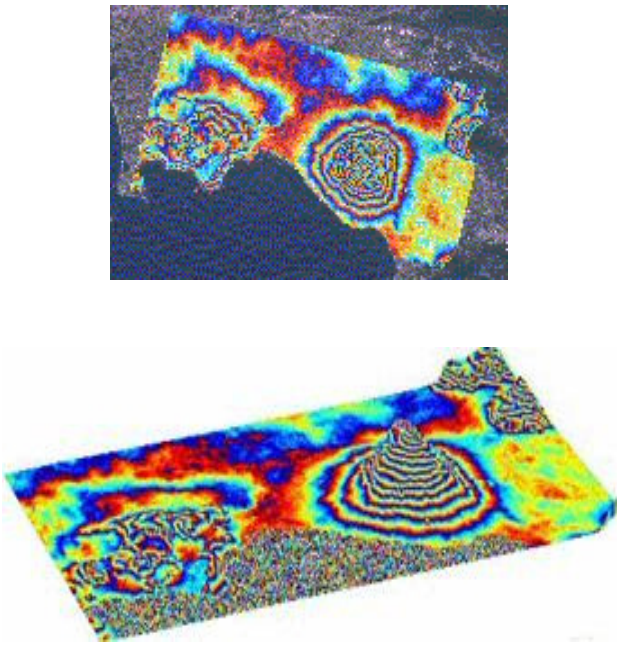


Figura B.10: Interferogrammi di fase rappresentati sul piano o in 3D di Napoli e Vesuvio

B2.2 I limiti dell'approccio tradizionale

L'analisi DInSAR non è diventata uno strumento operativo di analisi e monitoraggio molto diffuso per i diversi effetti che ne riducono (o addirittura compromettono) la qualità dei risultati.

In primo luogo, i fenomeni di decorrelazione temporale sono causati dalla variazione delle proprietà elettromagnetiche (riflettività) dei bersagli radar nel tempo. In questo caso l'ipotesi che il contributo di riflettività si elida, generando l'interferogramma, non è più verificata. Questi fenomeni risultano più marcati al crescere dell'intervallo di tempo - baseline temporale - tra le due acquisizioni utilizzate.

Le zone coperte da vegetazione, facilmente influenzabili dal vento e di diverso aspetto a seconda della stagione, sono fonte di decorrelazione, mentre i centri urbani e le rocce esposte rimangono maggiormente stabili nel tempo (cambiamenti possono essere causati anche da altri eventi atmosferici quali pioggia o neve).

La qualità dell'interferogramma dipende anche dalla distanza tra le due orbite effettivamente percorse dal sensore durante l'acquisizione delle due immagini; si parla in questo caso di *baseline normale* o *geometrico*. Si può dimostrare che maggiore è il valore assoluto del baseline, minore è la banda comune tra i due segnali e quindi minore è il rapporto segnale-rumore, relativo all'interferogramma da esse generato. Questo tipo di disturbo prende il nome di *decorrelazione geometrica.*

L'interpretazione dei dati interferometrici può essere ulteriormente complicata dalla variazione delle condizioni atmosferiche durante le due acquisizioni, tradotta in un ulteriore termine di fase difficile da discriminare dal contributo relativo a eventuali fenomeni di movimento.

Un'attenta analisi bibliografica porta comunque alla conclusione che, con l'interferometria SAR tradizionale, si è in grado di stimare movimenti dell'ordine del centimetro, ma in genere non è possibile effettuare stime

puntuali, bensì, solo analisi d'insieme per identificare fenomeni macroscopici in essere (estensione > 0.2 km²).

Bibliografia

[1] AGRAWAL R., BAYARDO R.J.JR. (1999), Mining the Most Interesting Rules, *in Proc. of the Fifth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 145-154.

[2] AGGARWAL C. C., PROCOPIUC C., WOLF J. L., YU P. S., PARK, J. S. (1999), Fast algorithms for projected clustering, *in Proc. of ACM SIGMOD International Conference on the Management of Data*, pagg. 61-72.

[3] AGRAWAL R., GEHRKE J., GUNOPULOS D., RAGHAVAN P. (1998), Automatic subspace clustering of high dimensional data for data mining applications, *in Proc. of ACM SIGMOD International Conference on the Management of Data, Seattle.*

[4] AGGARWAL C. C., YU P. S. (2000), Finding generalized projected clusters in high dimensional spaces, *in Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pagg. 70-81.

[5] AGRAWAL R., FALOUTSOS C. E SWAMI A.N. (1993), *Efficient Similarity Search in Sequence Databases* – FODO.

[6] AGRAWAL, SRIKANT (1994), Fast Algorithms for mining association Rules, *in Proc. 20th Int. Conf. Very Large Data Bases, VLDB*.

[7] AGRAWAL, SWAMI, IMIELINSKY (1993), Mining associations between sets of items in massive databases, *in Proc. of ACM SIGMOD, Washington D.C.*, pagg.207-216.

[8] ATLAS, COLE, MUTHUSAMY, LIPPMAN, CONDOR, PARK, EL SHARKAWI, MARKS (1990), *A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees*.

[9] BARALDI A., ALPAYDIN E. (2002), *Constructive Feedforward ART Clustering Networks—Part II*, IEEE Transactions on neural networks, vol. 13, nr. 3.

[10] BERRY M.J.A., LINOFF G. *Data Mining Techniques For Marketing, Sales and Customer support*. Wiley, 1997.

[11] BOLLOBAS B., DAS G., GUNOPULOS, D. E MANNILA H. (1997), Time-Series Similarity Problems and Well-Separated Geometric Sets, *in Proc. of the Association for Computing Machinery Thirteenth Annual Symposium on Computational Geometry*, pagg. 454–476,

[12] BRAIN D., WEBB G.I. (2002), The Need For Low Bias Algorithms in Classification Learning from Large Data Sets, *in Proc. of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pagg.:62-73.

[13] BREIMAN L., FRIEDMAN J., OLSHEN H., STONE R.A., CHARLES J. (1984), *Classification and Regression Trees,* Wadsworth.

[14] CARPENTER G.A., GROSSBERG S. (1987), *ART2:* Self-organization of stable category recognition codes for analog input patterns, *Applied Optics,* vol. 26, nr. 23, pagg. 4919–4930.

[15] CHEESEMAN P., STUTZ J. (1996), Bayesian Classification (AutoClass): Theory and Results, in *Proc. Advances in Knowledge Discovery and Data Mining,* U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds).

[16] CHOMICKI J., REVESZ P.Z. (1999), *Constraint-Based Interoperability of Spatiotemporal Databases* - Volume 3, Pagg. 211-243.

[17] CLEMENTINE (Integral Solutions) - http://www.spss.com/spssbi/clementine

[18] CORBETTA P. (2003), *La ricerca sociale: metodologia e tecniche* - il Mulino, (Vol. I: I paradigmi di riferimento, Vol. II: Le tecniche quantitative).

[19] DAS G., GUNOPULOS D.E, MANNILA H. (1997), Finding similar time series, *in Proc. of The Fourth International Conference on Knowledge Discovery and Data Mining*.

[20] DATTATREYA G.R., KANAL L.N. (1985), *Decision Trees in Pattern Recognition*.

[21] DI FONZO T., LISI F. (2001) *Complementi di statistica economica. Analisi delle serie storiche univariate*, Cleup Editrice, Padova.

[22] ESTER M., FROMMELT A., KRIEGEL H.P., E SANDER J. (1998) Algorithms for characterization and trend detection in spatial databases, *in Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining. Menlo Park,* CA: AAAI Press.

[23] ESTER M., KRIEGEL H.P., SANDER J. (1997), Spatial Data Mining: A Database Approach, *in Spatial Databases. LNCS 1262,* Springer Verlag, Berlin, pagg. 47-66.

[24] ESTER M., KRIEGEL H.P., SANDER J. (1999), *Knowledge Discovery in Spatial Databases*, KI, pagg. 61-74.

[25] EVERITT B. (1996) *Cluster Analysis* - Edward Arnold.

[26] FEIGE U. (1996), A threshold of ln(n) for approximating set cover, *in Proc. of the 28th Annual ACM Symposium on Theory of Computing*, pagg. 314-318

[27] FRANK, E., TRIGG, L., HOLMES, G., WITTEN, I. H. (2000), Naive Bayes for Regression - *Machine Learning,* Vol. 41.

[28] FURLETTI B., FORNASARI F. (2002), *Ambiente logico per il clustering spazio-temporale* - relatore prof. Franco Turini, A.A. 2001-2002.

[29] GANTI V., RAMAKRISHNAN R., GEHRKE J., POWELL A., FRENCH J. (1999), Clustering Large Datasets in Arbitrary Metric Spaces, *in Proc. of the 15th International Conference on Data Engineering (ICDE '99)*, Sydney.

[30] GAVRILOV M., ANGUELOV D., INDYK P. E MOTWANI R. (2000), Mining the Stock Market: Which Measure is Best? – *in Proc. of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining.*

[31] GE X., SMYTH P. (2000), *Deformable Markov Templates for Time Series Pattern Matching* - Technical Report N. 00-10, University of California at Irvine.

[32] GIUDICI P. (2001), *Data Mining, Metodi statistici per le applicazioni aziendali*, McGraw-Hill.

[33] GOLDBERG D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

[34] GRANGER C.W.J., NEWBOLD P. (1974), Spurious Regressions in Economics - *Journal of Econometrics*.

[35] GUHA S., RASTOGI R., SHIM K. (1998), *CURE: an efficient clustering algorithm for large databases*, ACM SIGMOD Record, vol. 27, n.2, pagg. 73-84.

[36] GUTTMAN A. (1984), *R-TREES: A Dynamic Index Structure for Spatial Searching*, SIGMOD Conference.

[37] HAN J., CAI Y., CERCONE N. (1992), Knowledge Discovery in Databases: An Attribute-Oriented Approach, *in Proc. of the 18th*

International Conference on Very Large Data Bases, pagg.547-559, August 23-27.

[38] HAN J., KAMBER M. (2000), *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers.

[39] HAN E. H., KARYPIS G., KUMAR V., MOBASHER B. (1997), Clustering in a high dimensional space using hypergraph models, *in Technical Report TR-97-063, Department of Computer Science, University of Minnesota, Minneapolis.*

[40] HAN, FU (1995), Discovery of Multiple-Level Association rules from Large Databases, *in Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95).*

[41] HAN J., KAMBER M., E TUNG A.K.H. (2001), Spatial Clustering Methods in Data Mining: A Survey – H. Miller and J. Han (eds.) - *Geographic Data Mining and Knowledge Discovery* - Taylor and Francis.

[42] HARVEY C. (1989), *Forecasting, structural time series models and the Kalman filter* - Cambridge University Press.

[43] HECKERMAN, MANNILA H., PREGIBON D. E UTHURUSAMY R. (1997), *in Proc. of the Third International Conference on Knowledge Discovery and Data Minig (KDD-97)*, AAAI Press.

[44] HECKERMAN, MANNILA H., PREGIBON D. e UTHURUSAMY R. (1997), *in Proc. of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, AAAI Press.

[45] HINNEBURG A., KEIM D.A. (1999), Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering, *in Proc. of the 25th International Conference on Very Large Data Bases*, pagg. 506–517.

[46] HUANG Z. (1998), Extensions to the K-means algorithm for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery*, vol.2, no.3, pagg. 283-304, Kluwer Academic Publishers.

[47] HUHTALA Y., KARKKAINEN J. E TOIVONEN H. (1999*),* Mining for Similarities in Aligned Time Series Using Wavelets - *Data Mining and Knowledge Discovery: Theory, Tools and Technology,* SPIE Proceedings Series Vol. 3695, Orlando, pagg. 150-160.

[48] Intelligent Miner (IBM): http://www-3.ibm.com/software/data/iminer.

[49] IULIANO S., MATANO F., NARDO' S., PISCITELLI E., RISI A., TERRANOVA C. (2007) - An integrated satellite interferometry (ps-insar) and field geology study in monitoring slow landslides in urban areas in Campania region (southern Italy). *Atti Convegno FIST Geoitalia* 2007, Rimini, Epitome, 2, 223.

[50] JAIN A.K., MURTY M.N., FLYNN P.J. (1999), *Data clustering: A survey*, ACM Computing Survey.

[51] JAMBU M., LEBEAUX M. (1983), *Cluster analysis and data analysis* - North Holland.

[52] JONES D., BELTRAMO M. A. (1991), Solving partitioning problems with genetic algorithms, *in Proc. of the Fourth International Conference on Genetic Algorithms*, pagg. 442–449.

[53] KAUFMAN L. (1990), *Finding groups in data: an introduction to cluster analysis,* Wiley, New York.

[54] KOHONEN T. (2001), *Self-Organizing Maps*, 3a edizione, Springer-Verlag, Berlino.

[55] LIKAS A., VLASSIS N., VERBEEK J. J. (2003), The Global K-means Clustering Algorithm, *Pattern Recognition* 36(2).

[56] MANNILA H. (1997), *Methods and problems in Data Mining*, ICDT.

[57] MANUNTA P., DEFLORIO A. M., PAGANINI M., PALAZZO F., FARINA P., MORETTI S., COLOMBO D., MENDUNI G., BRUGIONI M., SULLI L., MONTINI G., RISI A., TERRANOVA C., PISCITELLI E., MATANO F. & IULIANO S. (2005), Sviluppo di un servizio di mappatura e monitoraggio delle frane in Italia e Svizzera: integrazione di tecnologie satellitari ed analisi geologica. *Atti del Convegno A.I.T. "Telerilevamento e dissesto idrogeologico stato dell'arte e normativa"*, Cagliari 7-8 luglio 2005, 37-56.

[58] Mishra S. K., Raghavan V. V. (1994), An empirical study of the performance of heuristic methods for clustering, *Pattern Recognition in Practice*, E. S. Gelsema, L. N. Kanal (eds.), pagg. 425–436.

[59] NAGESH H., GOIL S., CHOUDHARY A.(1999), *MAFIA: Efficient and scalable subspace clustering for very large data sets*, Technical Report 9906-010, Northwestern University.

[60] NG R. T., HAN J. (1994), Efficient and Effective Clustering Methods for Spatial Data Mining, *in Proc. of the 20th VLDB Conference*, Santiago, pagg. 144-155.

[61] PICCOLO D. (2001), *Statistica* , Il Mulino.

[62] PICCOLO D. (2004), *Introduzione alla Statistica*, Il Mulino.

[63] PICCOLO D. (1990), *Introduzione all'analisi delle serie storiche*, NIS

[64] POVINELLI R.J., FENG X. (1999), Data Mining of Multiple Non stationary Time Series -*Proceedings of Artificial Neural Networks in Engineering* – St. Louis, Missuri.

[65] RAYMOND T. NG R.T., HAN J. (1994) *Efficient and Effective Clustering Methods for Spatial Data Mining*, pagg. 144-155.

[66] RAUBER A. (1999), LabelSOM: On the labeling of self organizing maps, *in Proc. of the International Joint Conference on Neural Networks*, Washington D .C.

[67] RECKHOW R. A., CULBERSON J. (1987), Covering simple orthogonal polygon with a minimum number of orthogonally convex polygons, *in Proc. of the ACM 3rd Annual Computational Geometry Conference*, pagg. 268-277.

[68] RISI A., TERRANOVA C., CASCONE E., COPPIN D., D'ARGENIO F., GELLI L., IULIANO S., MATANO F., NARDO' S., PISCITELLI E. (2007), Tellus project: a pilot project for developing a satellite based monitoring system for slope hazard detection in urban areas of Campania region. *Atti Convegno FIST Geoitalia,* Rimini, Epitome, 2, 201.

[69] SAFAVIAN S.R., LANDGREBE D. (1991) *A survey of Decision Tree Classifier Methodology*, IEEE Transactions on Systems, Man and Cybernetics, pagg. 660-674.

[70] SHEIKHOLESLAMI G., CHATTERJEE S., ZHANG A. (1998), WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases, *in Proc. of the 24th International Conference on Very Large Data Bases*, New York, pagg. 428-439.

[71] WANG W., YANG J., MUNTZ R. (1997), STING: A statistical information grid approach to spatial data mining, *in Proc. of the 23rd International Conference on Very Large Data Bases*, Atene, pagg. 186-195.

[72] WHITLEY D., STARKWEATHER T., FUQUAY D. (1989), Scheduling problems and traveling salesman: the genetic edge recombination, *in Proc. of the Third International Conference on Genetic Algorithms*, pagg. 133–140.

[73] WILLIAMSON J. R. (1996), *Gaussian ARTMAP: A neural network for fast in-cremental learning of noisy multidimensional maps*, Neural Networks, vol. 9, n. 5, pagg. 881–897.

[74] YULE G.W. (1926), Why Do We Sometimes Get Nonsense Correlations Between Time Series? A Study on Sampling and the Nature of Time Series, *Journal of the Royal Statistical Society*, 89, pagg. 1-64.

[75] ZHANG T., RAMAKRISHNAN R., LIVNY M. (1996), *BIRCH*: An Efficient Data Clustering Method for Very Large Databases, *in Proc. of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, pagg. 103-114.

[76] ZUUR A.F., FRYER R.J., JOLLIFFE I.T., Dekker R., Beukema J.J. (2003[a]), Estimating common trends in multivariate time series using dynamic factor analysis, *Environmetrics.*

[77] ZUUR A.F., TUCK I.D., BAILEY N.(2003[b]), Dynamic factor analysis to estimate common trends in fisheries time series, *Canadian Journal of Fisheries and Aquatic Sciences*.