Lezione 11 - Strutture dati efficienti per la ricerca della similarità

https://www.youtube.com/watch?v=GJTqdAV0bR0

Introduzione

Per un insieme di oggetti multimediali, la fase di indicizzazione produce un insieme di vettori di caratteristiche, che possono contenere un numero grande di componenti

La fase di retrieving è fondamentalmente caratterizzata da un gran numero di confronti di caratteristiche tra Query e oggetti memorizzati

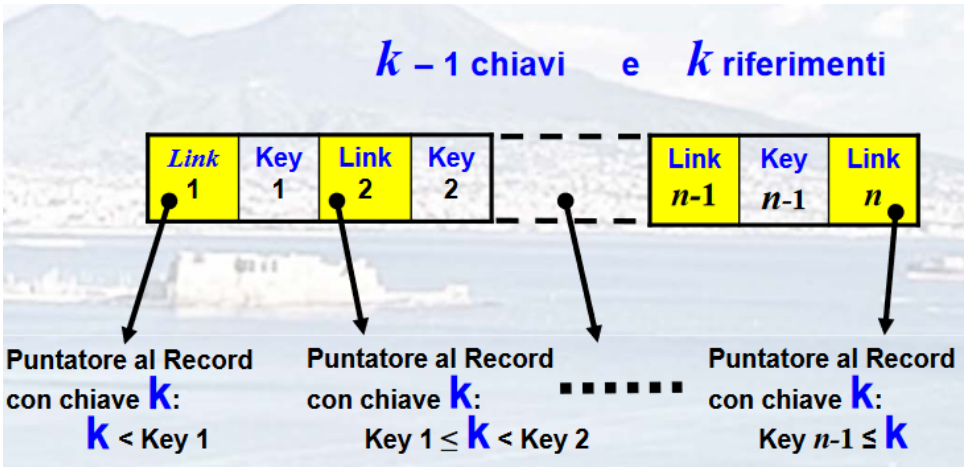
Alberi B

Un albero B di ordine m, in cui m rappresenta il massimo numero di figli che ogni nodo può avere, è un albero di ricerca generico

- La radice ha almeno 2 sottoalberi
- Ogni nodo che non sia la radice e che non sia una foglia contiene k-1 chiavi e k riferimenti a sottoalberi
- Ogni nodo foglia contiene k-1 chiavi
- Tutte le foglie sono sullo stesso livello

Secondo queste proprietà un albero B è sempre pieno per metà, ha pochi livelli ed è perfettamente bilanciato

Struttura nodo



Nodo di un albero

Ogni nodo contiene k-1 chiavi e dei Link che puntano al record con una data chiave K

Operazioni e Proprietà

Operazioni:

- Creazione
- Inserimento
- Cancellazione
- Ricerca

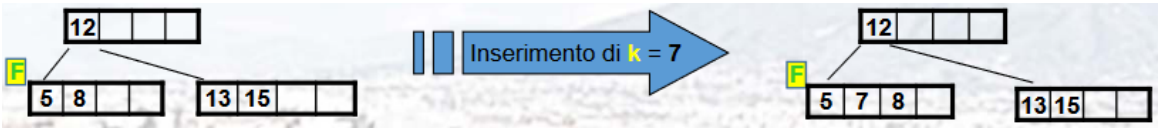
Porprietà strutturali:

- Albero bilanciato
  - Albero che ha la minor possibile altezza
- Prevedibilità per la complessità delle operazioni di ricerca e attraversamento dell'albero

▼ Inserimento (esempio)

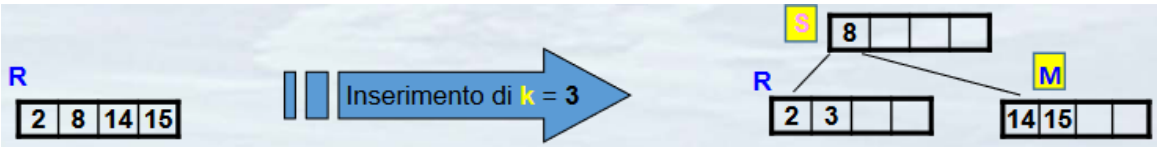
Per l'inserimento di una nuova chiave K nell'albero B si hanno 3 casi:

1. Se la foglia F in cui deve essere inserita K ha ancora spazio, allora viene inserito K



2. Se la foglia F in cui deve essere inserita K è piena allora:
    a. Si crea una nuova foglia G, e metà delle chiavi di F vengono inserite in G
    b. Una chiave di F si sposta nel proprio padre P
    c. Nel padre P si introducono i puntatori di G

3. Se la radice R dell'albero è piena si crea una nuova radice S ed un nuovo fratello M della radice R



Ricerca

La ricerca sugli alberi B è particolarmente efficiente, in quanto essi sono intrinsecamente bilanciati

Alberi B+

Sono considerati una variante dell'albero B

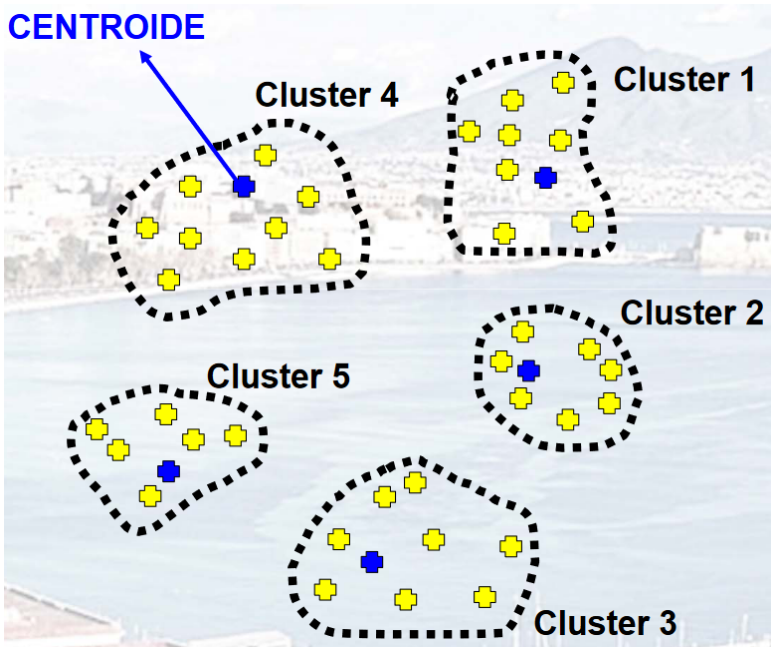
- I riferimenti ai dati sono contenuti solo nelle foglie
- Le foglie di un albero B+ contengono anche un campo puntatore aggiuntivo che permette di navigare tra le foglie

Clustering

Tecnica per ottimizzare i tempi di ricerca nello spazio di feature n-dimensionale

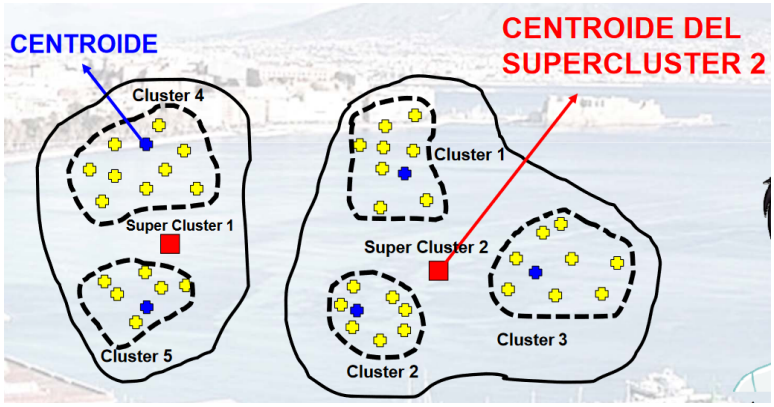
Vettori di features simili vengono raggruppati incluster, in base a misure di similarità

Ogni cluster è rappresentato dal proprio centroide, e il calcolo della similarità avviene tra la query e il centroide del singolo cluster



Clustering a più livelli

Quanto il numero di cluster è comunque alto, si utilizzano cluster a livelli multipli per ridurre il numero di calcoli di similarità



Cluster a più livelli

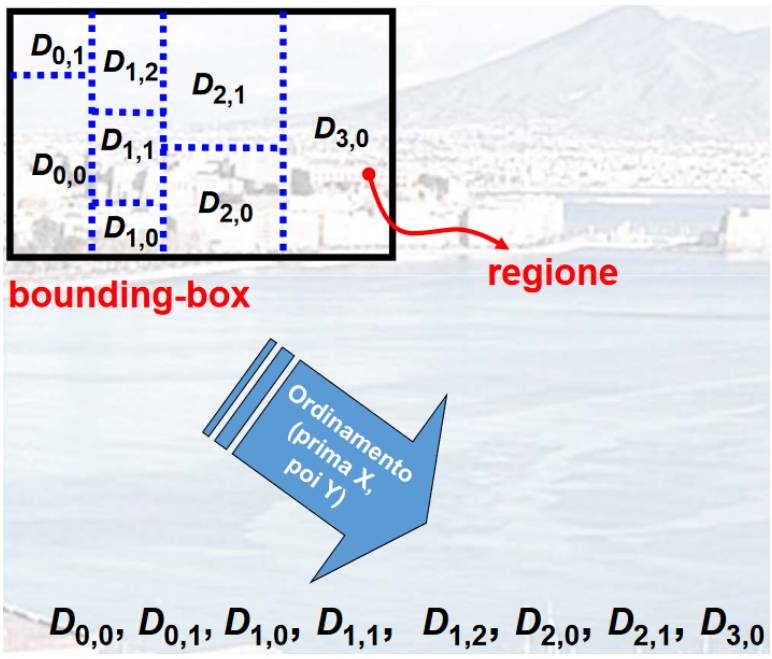
Alberi B+ Multidimensionali

Sono estensioni degli alberi B+

Esempio in 2D:

- Ogni feature vector è un punto dello spazio
- L'intero spazio delle feature "bounding-box" contenente tutti i punti è il rettangolo identificato dallo spigolo in basso a sinistra e dallo spigolo in alto a destra
- Dividiamo il bounding-box in regioni contenenti feature simili

- Ordiniamo le regioni secondo un criterio
- Ogni feature-vector ha un link con il dato multimediale di cui è una rappresentazione



Ricerca su Alberi MB+

Point Query:

- Ricerca di un vettore dato (x,y)
- Si parte dalla root e si trova la regione che contiene il vettore da ricercare
- Si scorre la lista di feature-vector associata alla regione

Range Query:

- Ricerca di tutti i vettori che ricadono in un rettangolo
- Partendo dalla root troviamo tutte le regioni che si sovrappongono al rettangolo di ricerca
- Si scorre la lista di feature-vector associata alla regione K

Nearest-Neighbour Query:

- Ricerca dei K vettori più vicini ad un vettore dato
- Si usa un procedimento iterativo, basato sulla ripetizione di Range query finché non si trova un numero sufficiente di vettori candidati

- Si usa il calcolo della distanza euclidea tra il vettore da ricerca e d i vettori candidati

K-d Trees

Sono considerati ulteriori estensioni degli alberi Binari

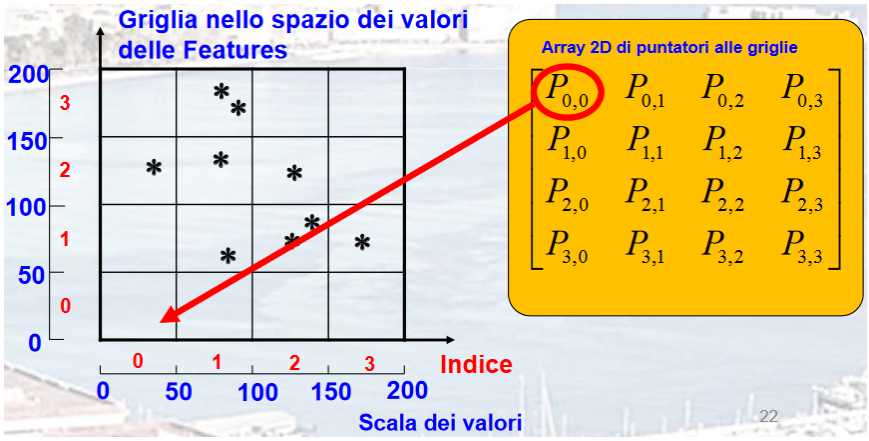
- Ogni chiave è costituita dal vettore K-dimensionale invece che da un solo valore
- Per generare l'albero occorre regolamentare la modalità di inserimento:
  - Al primo livello si decide basandosi sulla prima componente del vettore
  - Al secondo livello si decide basandosi sulla secondo componente
  - Etc…
  - Esaurite le K dimensioni si ricomincia dalla prima

Operazioni

- Inserimento → Per ciascun livello vale l'ordinamento relativo alla componente corrispondente del feature vector
- Ricerca → Simile al processo di inserimento. Per ogni livello la scelta dipende solo dal valore della relativa componente del vettore
- Eliminazione → Può risultare complicata quando occorre eliminare un nodo intermedio dell'albero, e si possono effettuare cancellazioni logiche senzsa modificare la struttura
- Range Query → Comporta la riduzione del numero dei nodi da visitare

Grid Files

Consistono nella suddivisione dello spazio n-dimensionale in ipercubi aventi tutti la stessa dimensione, dove ogni ipercubo contiene zero o più feature vector



Esempio di Grid files in 2D

Operazioni

- Inserimento → Molto semplice
- Ricerca → Con Point Query o Range Query

Considerazioni

Se i vettori di features sono distribuiti abbastanza uniformemente all'interno dello spazio dei valori allora tale metodo da buoni risultati, altrimenti alcune griglie risultano vuote (o quasi) e altre sovraffollate

Per risolvere questo problema si può modificare la grandezza delle griglie per renderle di dimensione variabile. Quindi zone più dense avranno griglie più piccole

R Tree

Generalizzazione dell'MB+ Tree, che identifica una famiglia di strutture indicizzate molto utilizzate per l'organizzazione dei dati multimediali

La struttura dati divide lo spazio in MBR (Minimum Bounding Rectangles), e ogni nodo dell'R-Tree ha un numero variabile di entry

Ogni entry che non sia una foglia contiene un'entità che identifica il nodo figlio, e una che contiene tutte le entry del nodo figlio

Operazioni

- Query (ricerca) → A partire dalla root si attraversa l'albero cercando i rettangoli che intersecano il Minimum Bounding Box
- Insert → Si attraversa l'albero selezionando il rettangolo più piccolo che include l'oggetto da inserire o quello che richiederebbe l'allargamento minore per

“coprire” il nuovo oggetto

- Delete → Si utilizza un processo simile a quello di ricerca