K-d-tree

ALBERTO BELUSSI

ANNO ACCADEMICO 2012-2013

Caratteristiche di base

Il ***k-d-Tree*** (*K-dimensional tree*) è un albero binario di ricerca multi-dimensionale in cui ogni nodo indicizza un punto in k dimensioni.

Trattiamo il caso k=2

Introduciamo la definizione di livello di un nodo.

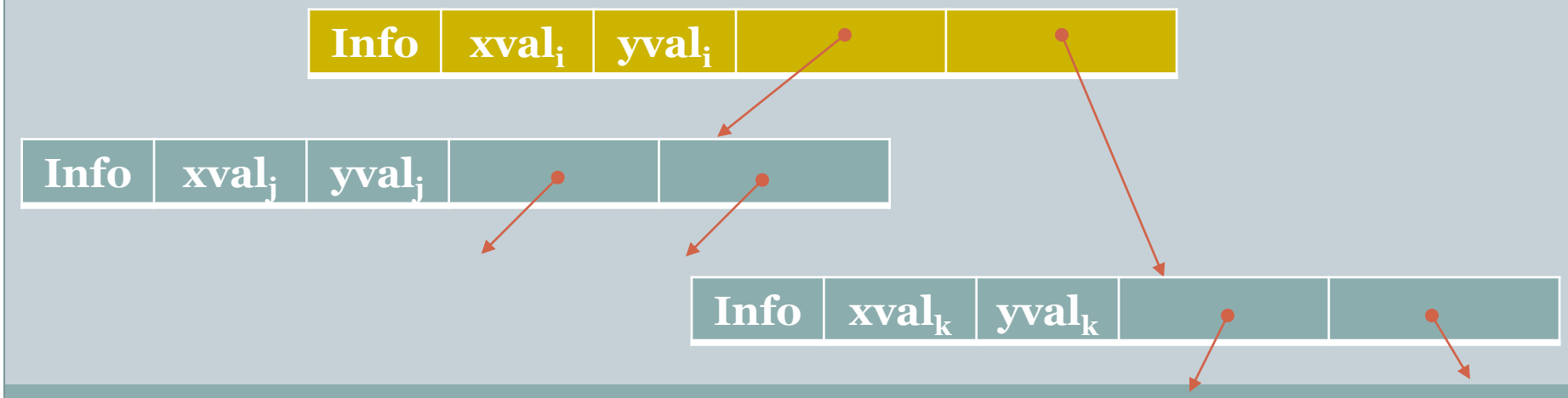
Sia T la radice dell'albero ed N un generico nodo, il livello di N è una funzione definita ricorsivamente come segue:

- level(N) = 0 se N è la radice T dell'albero.
- level(N) = level(P)+1 se N non è la radice e P è il padre di N.

Struttura di un nodo

Ogni nodo contiene:

- info: INFOTYPE;
- xval: REAL;
- yval: REAL;
- left-link: *NODE;
- right-link: *NODE;



Vincoli strutturali

- Sia N un generico nodo tale che level(N) sia pari, allora:
  - Per ogni nodo $CHILD_1$ appartenente al sottoalbero sinistro di N (N.left-link), si ha che $CHILD_1$.**xval** < N.**xval**.
  - Per ogni nodo $CHILD_2$ appartenente al sottoalbero destro di N (N.right-link), si ha che $CHILD_2$.**xval** ≥ N.**xval**.
- Sia N un generico nodo tale che level(N) sia dispari, allora:
  - Per ogni nodo $CHILD_1$ appartenente al sottoalbero sinistro di N (N.left-link), si ha che $CHILD_1$.**yval** < N.**yval**.
  - Per ogni nodo $CHILD_2$ appartenente al sottoalbero destro di N (N.right-link), si ha che $CHILD_2$.**yval** ≥ N.**yval**.

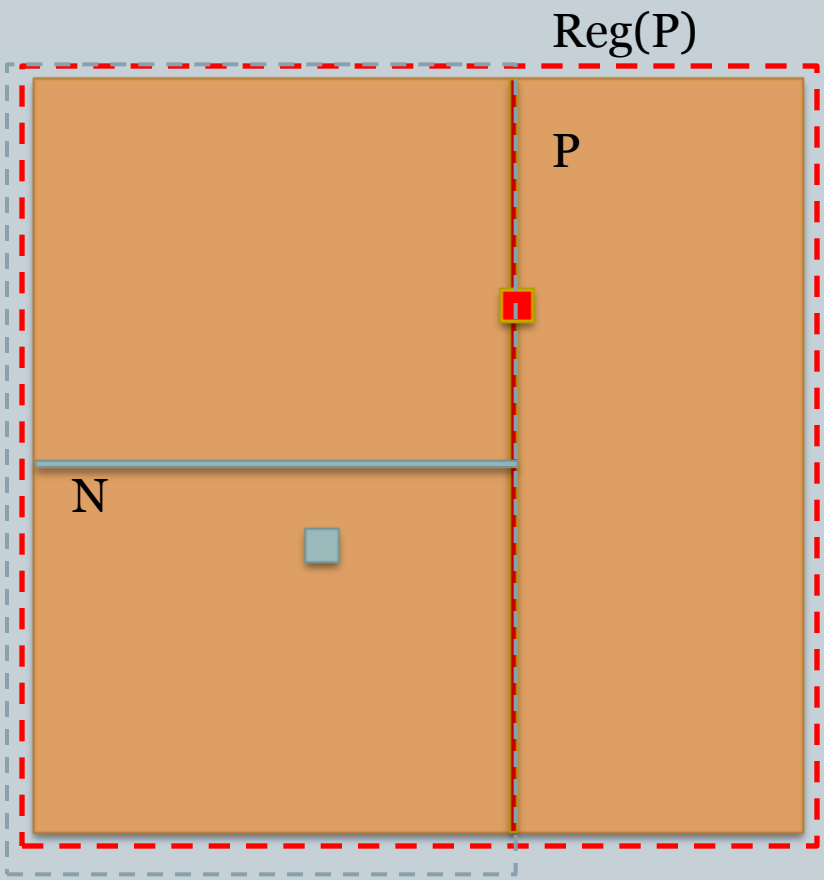
Vincoli strutturali

Ad ogni punto si può pensare che sia associata una regione rettangolare, la quale viene implicitamente divisa in due parti tramite una linea verticale (livello pari) oppure orizzontale (livello dispari) passante per il punto.

Alla radice dell'albero è associata l'intera regione di partenza.

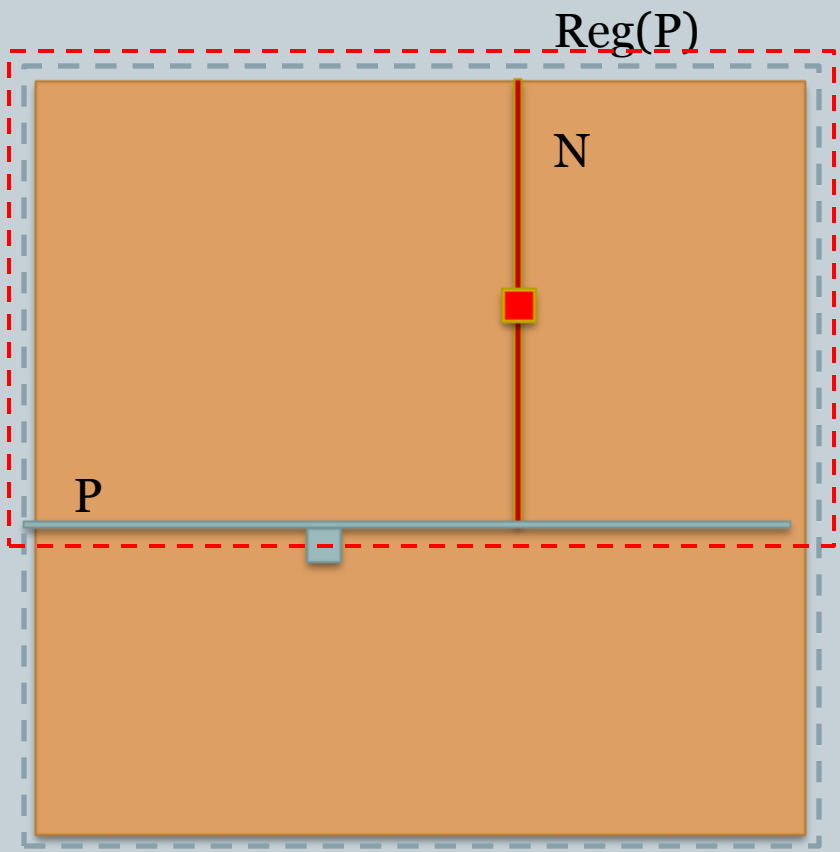
Regione di spazio associata ad un nodo

- Alla radice è associato tutto lo spazio di riferimento
- Dato un nodo N di padre P:
  - se level(P) è pari, allora:
    - se N = P.left-link, allora
      - N.xlb = P.xlb
      - N.xub = **P.xval**
    - se N = P.right-link, allora
      - N.xlb = **P.xval**
      - N.xub = P.xub
    - N.ylb = P.ylb
    - N.yub = P.yub

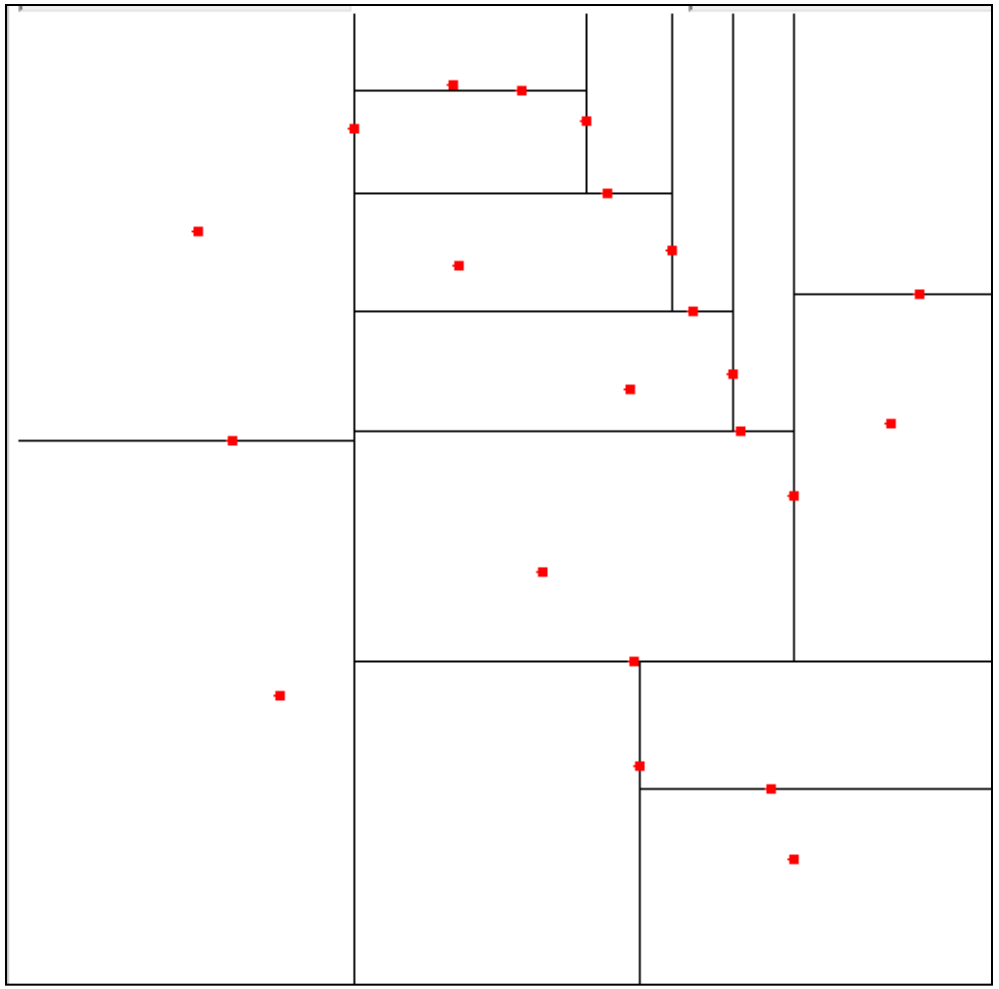


Regione di spazio associata ad un nodo

- Alla radice è associato tutto lo spazio di riferimento
- Dato un nodo N di padre P:
  - se level(P) è dispari, allora:
    - se N = P.left-link, allora
      - N.ylb = P.ylb
      - N.yub = **P.yval**
    - se N = P.right-link, allora
      - N.ylb = **P.yval**
      - N.yub = P.yub
    - N.xlb = P.xlb
    - N.xub = P.xub



Esempio



Caratteristiche strutturali

- Questi alberi non sono bilanciati in generale e, nel caso peggiore, possono degenerare in una lista.
- Per ovviare a questa limitazione i punti dovrebbero essere inseriti casualmente in modo che il valore atteso della profondità dell'albero sia logaritmico
- oppure si potrebbero applicare delle tecniche di ri-bilanciamento per mantenere il tempo di ricerca logaritmico.

Operazioni su un K-d-tree

Algoritmi di ricerca

Point query, ovvero ricerca di un punto P = (x, y):

- Sia Q il nodo corrente (inizialmente il nodo radice che si assume essere di livello pari).
- Se (Q.xval,Q.yval) = P, il punto è stato trovato e la ricerca termina, altrimenti:
  - se Q è di livello pari:
    - se x < Q.xval allora Q = Q.left-link, altrimenti Q = Q.right-link.
  - se Q è di livello dispari:
    - se y < Q.yval allora Q = Q.lelft-link, altrimenti Q = Q.right-link.
- Si riparte dal secondo passo fino a che non si trova P o si raggiunge la fine dell'albero (puntatori a NULL).

Operazioni su un K-d-tree

- Il tempo di una **Point query** è logaritmico nel numero totale di nodi nel caso di un albero bilanciato, tuttavia, come già accennato, potrebbe anche diventare lineare nel caso in cui il k-d-Tree sia degenerato in una lista.

Operazioni su un K-d-tree

Algoritmi di ricerca

- **Range query**, ovvero ricerca di tutti i punti P_i che appartengono ad una regione R passata in ingresso che assumiamo essere rettangolare. Supponiamo inoltre che sia nota la regione di spazio associata ad un nodo Reg(N).
- Sia N il nodo corrente (inizialmente il nodo radice che si assume essere di livello pari).
- se N è un nodo interno allora:
  - se $Reg(N) \subseteq R$ ritorna tutti i punti dei sottoalberi di N compreso (N.xval,N.yval),
  - se $Reg(N) \cap R \neq \emptyset$: se $(N.xval,N.yval) \in R$ ritorna (N.xval,N.yval) e visita i figli di N che intersecano R,
  - se $Reg(N) \cap R = \emptyset$ allora la ricerca termina.

Operazioni su un K-d-tree

- se N è una foglia e (N.xval,N.yval) $\in$ R, allora ritorna (N.xval,N.yval).

Operazione di inserimento

Inserimento del nodo relativo ad un punto P = (xval?, yval?):

- Ricerca del nodo foglia che dovrebbe contenere P.
- Creazione del nodo e aggiornamento dei campi del record relativo al punto P.
- I puntatori ai figli presenti nel nodo foglia vanno messi a NULL.

La creazione di un k-d-Tree consiste nella creazione del nodo radice con relativo aggiornamento dei campi statici e puntatori a NULL, seguita dall'inserimento dei nodi successivi. Con una procedura ricorsiva si costruisce un k-d-Tree in O(n log n), dove n è il numero di punti.

Operazione di cancellazione

Cancellazione del nodo relativo ad un punto P = (xval?, yval?):

- la cancellazione di un nodo foglia non porta ad alcun problema poiché basta assegnare a NULL il puntatore del padre.
- se invece il nodo da cancellare è interno, bisogna trovare una riorganizzazione dell'albero in modo tale che siano nuovamente rispettati i vincoli strutturali.
  Esistono diversi modi di procedere.

Operazione di cancellazione

- La prima soluzione consiste nel cancellare il nodo e reinserire tutti quelli dei suoi sottoalberi ma è troppo onerosa.
- La seconda soluzione, consiste in una cancellazione logica, nel senso che il nodo da rimuovere viene marcato come “eliminato” in modo che non sia preso in considerazione nelle successive ricerche: quando l’albero dovrà essere ricostruito tutti i nodi marcati saranno rimossi.

Operazione di cancellazione

- La terza soluzione consiste nel cercare un nodo candidato R nel sottoalbero destro di N (o sinistro, in questo caso si sposta il sottoalbero a destra). L'idea è di mettere R al posto di N, che quindi viene cancellato.
  - Rimpiazzare i campi statici (quelli non link) di N con quelli di R.
  - Cancellare R.

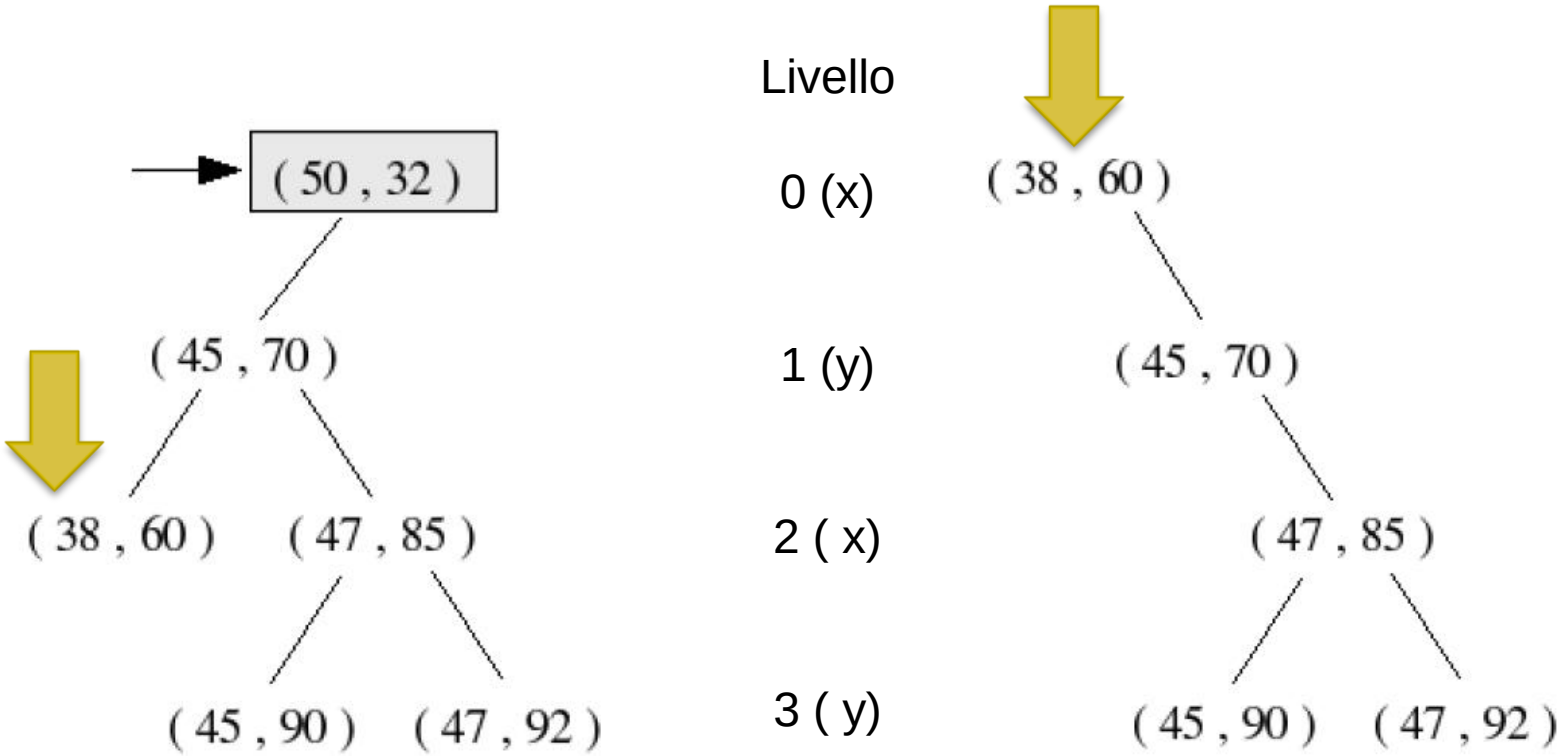
Operazione di cancellazione

- Come trovare il candidato giusto?
  Sia N il nodo interno da cancellare:
  - se level(N) è pari:
    - se esiste il sottoalbero destro di N il candidato è il nodo R tale che
      - R.xval è minore o uguale del valore xval presente in tutti i nodi del sottoalbero destro,
    - se invece N.right-link = NULL, si scambiano i puntatori left-link ed right-link di N, in modo tale che quello che prima era il sottoalbero sinistro di N ora sia quello destro; si ritorna al passo precedente.

Operazione di cancellazione

- se level(N) è dispari:
  - se esiste il sottoalbero destro di N, il candidato è il nodo R tale che R.yval è minore o uguale del valore yval presente in tutti i nodi del sottoalbero destro,
  - se invece N.right-link = NULL, si scambiano i puntatori left-link ed right-link di N, in modo tale che quello che prima era il sottoalbero sinistro di N ora sia quello destro; si ritorna quindi al passo precedente.

Esempio di cancellazione



Sito di riferimento per indici multi-dimensionali

VASCO Spatial Index demo (Prof. Hanan Samet):

http://donar.umiacs.umd.edu/quadtree/points/kdtree.html