THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

SDS 385: Stat Models for Big Data

Lecture 9: KD trees

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

https://psarkar.github.io/teaching

Background

- Has a long history–invented in 1970 by Jon Bentley
- k represents the number of dimensions
- Idea is to partition the data spatially, by using only one dimension at any level.
- While searching, this helps pruning most of the search space.

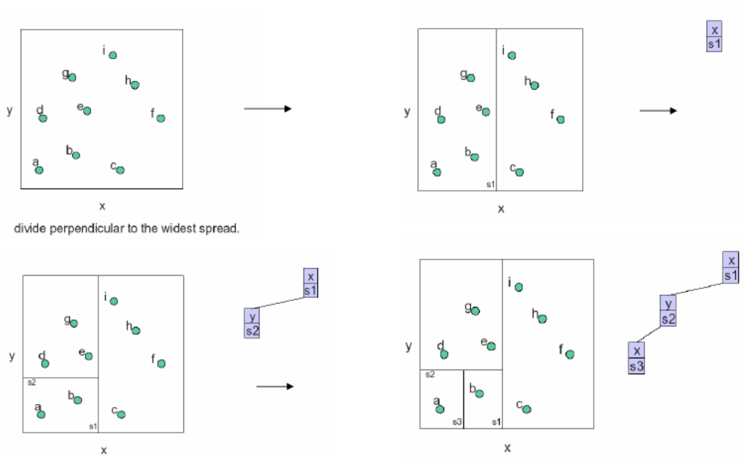
General idea

- Cycle through the dimensions for each level
- Call this cut-dim (cutting dimension)
- Node in tree contains $P = (x, y)$
- So, to find a point, only need to compare the cutting dimension.

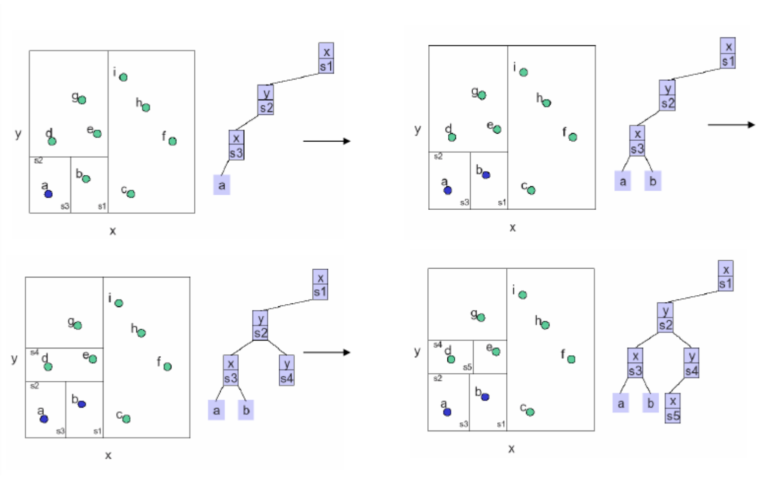
Construct

- If there is one point, just form a leaf node
- Otherwise divide the points in half along the cutting axis
    - Find the axis with the widest spread
    - divide in alternative/round robin fashion
- recursively build kdtrees from each half
- Complexity $dn \log n$

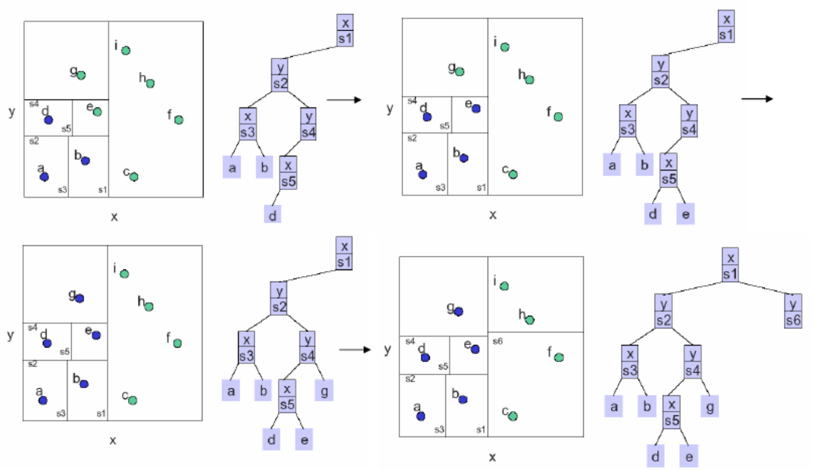
Insert



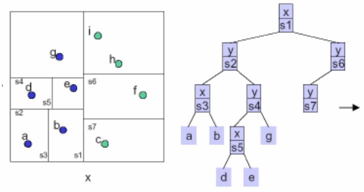
Insert



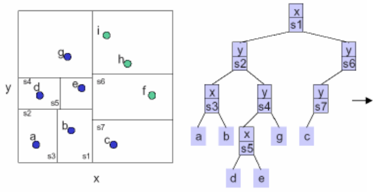
Insert



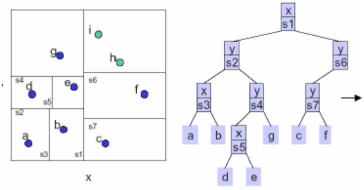
Insert



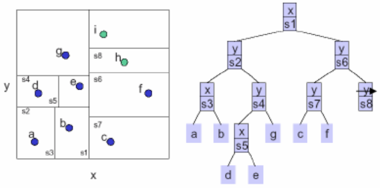
i
g
h
s4
d
e
s6
f
s5
s2
b
s7
a
c
s3
s1
x
x s1
y s2
y s6
x s3
y s4
y s7
a
b
x s5
g
d
e



i
g
h
y
s4
d
e
s6
f
s5
s2
b
s7
a
c
s3
s1
x
x s1
y s2
y s6
x s3
y s4
y s7
a
b
x s5
g
c
d
e



i
g
h
s4
d
e
s6
f
s5
s2
b
s7
a
c
s3
s1
x
x s1
y s2
y s6
x s3
y s4
y s7
a
b
x s5
g
c
f
d
e



i
g
s8
h
y
s4
d
e
s6
f
s5
s2
b
s7
a
c
s3
s1
x
x s1
y s2
y s6
x s3
y s4
y s7
y s8
a
b
x s5
g
c
f
d
e

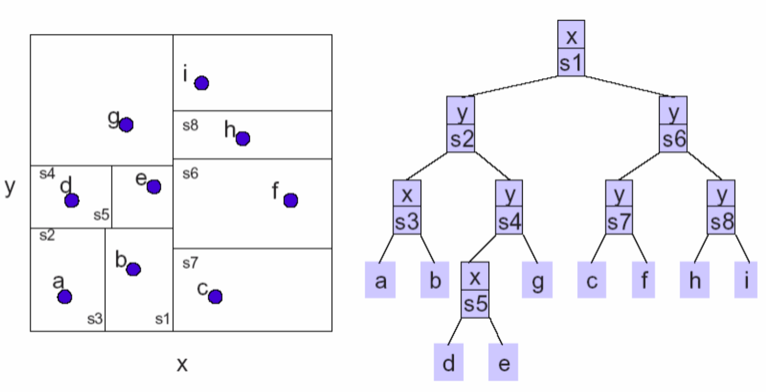
Find point with the smallest element in dimension a

- If cutdim at current node equals a,
  - the min cannot be in the right subtree
  - recurse on the left subtree

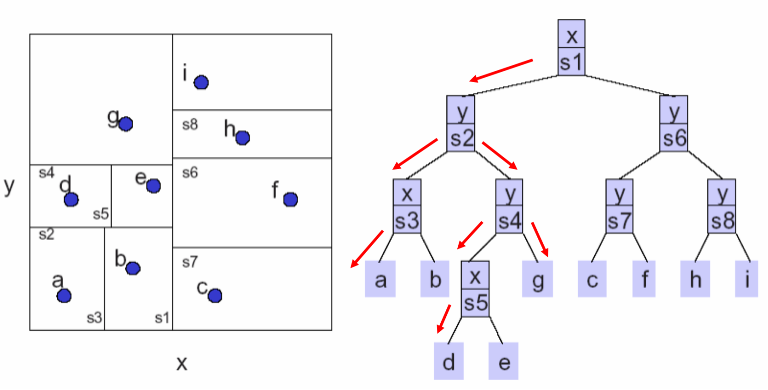
Base case: if there are no left children, stop and return current point.

- Otherwise
  - the min could be in either
  - recurse on both left and right subtrees

Find point with the smallest element in dimension x



Find point with the smallest element in dimension x



Nearest neighbor queries

- Given point Q, find the closest point R
- Have to be careful, because its possible that two points are far away in the tree but close in the Eucidean space.
- For each node store a bounding box

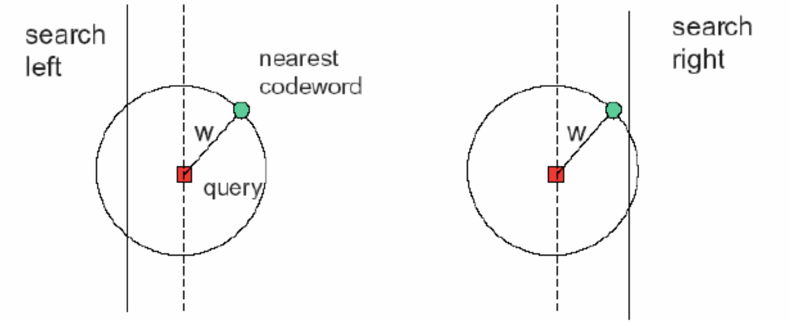
Nearest neighbor queries

- Given point Q, find the closest point R
- Have to be careful, because its possible that two points are far away in the tree but close in the Eucidean space.
- For each node store a bounding box
- Remember the closest point to Q seen so far (call this R')

Nearest neighbor queries

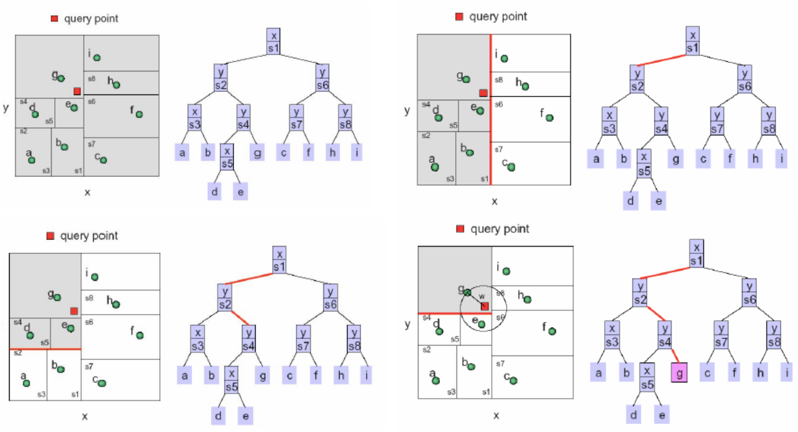
- Given point Q, find the closest point R
- Have to be careful, because its possible that two points are far away in the tree but close in the Eucidean space.
- For each node store a bounding box
- Remember the closest point to Q seen so far (call this R')
- Prune subtrees where bounding boxes cannot contain R'

Nearest neighbor queries

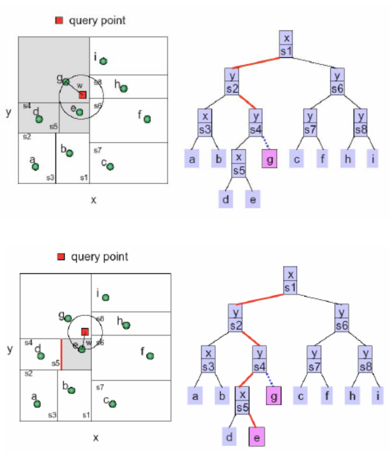


- If circle overlaps with left subtree, search left subtree
- If circle overlaps with right subtree search right subtree
- Has been shown to work in about $O(\log n)$ time.

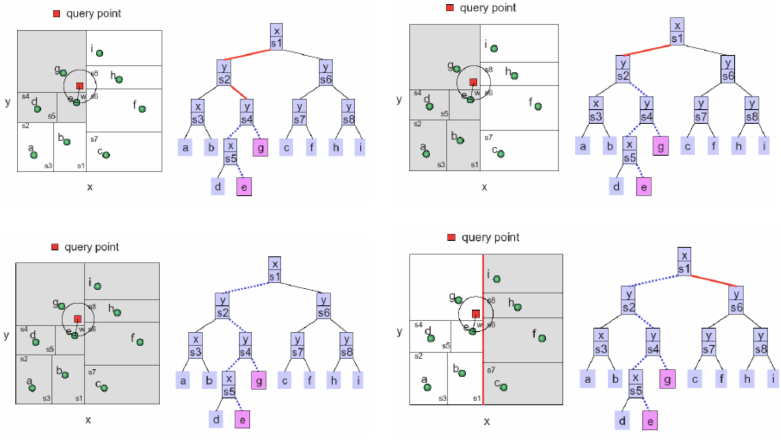
NN search



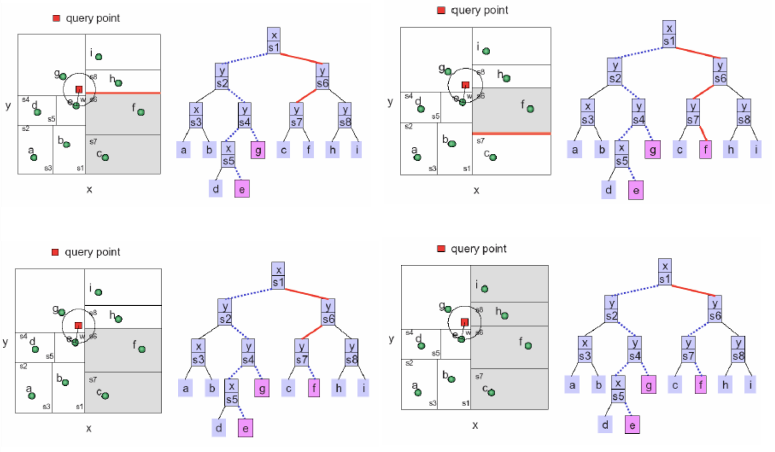
NN search



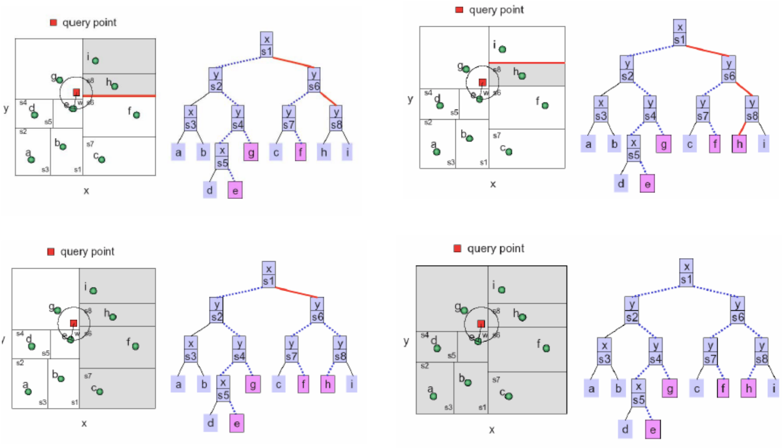
NN search



NN search


query point
x
y
s1
s2
s3
s4
s5
s6
s7
s8
a
b
c
d
e
f
g
h
i
query point
query point
query point

NN search


query point
x
y
s1
s2
s3
s4
s5
s6
s7
s8
a
b
c
d
e
f
g
h
i
query point
query point
query point

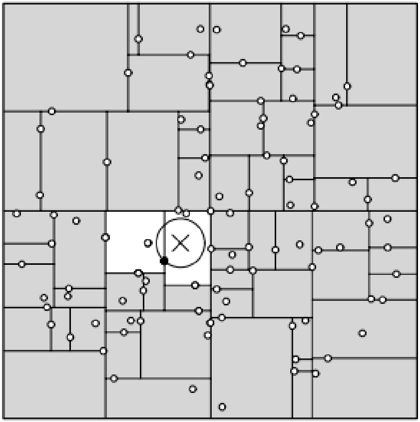


Figure 6.5

Generally during a nearest neighbour search only a few leaf nodes need to be inspected.

NN search

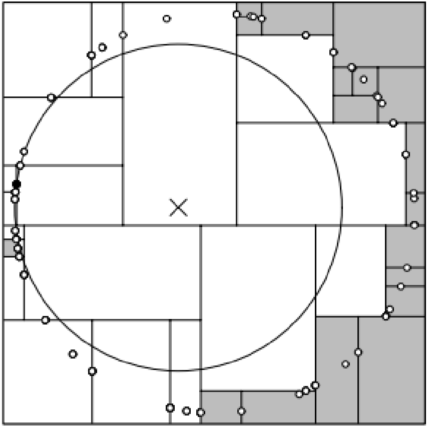


Figure 6.6

A bad distribution which forces almost all nodes to be inspected.

Timing vs tree size

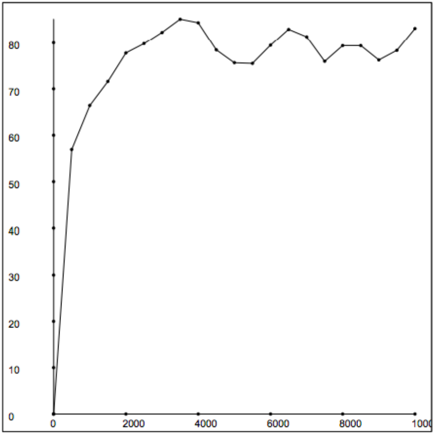


Figure 6.8

Number of inspections against kd-tree size for an eight-dimensional tree with an eight-dimensional underlying distribution.

Timing vs dimensions

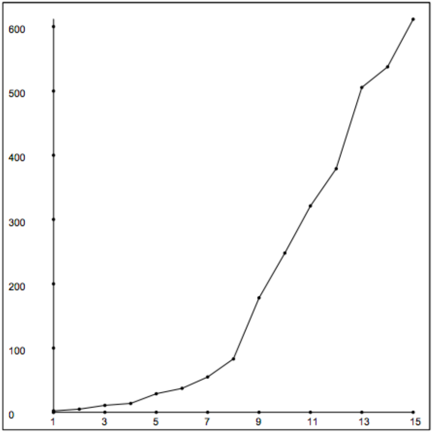
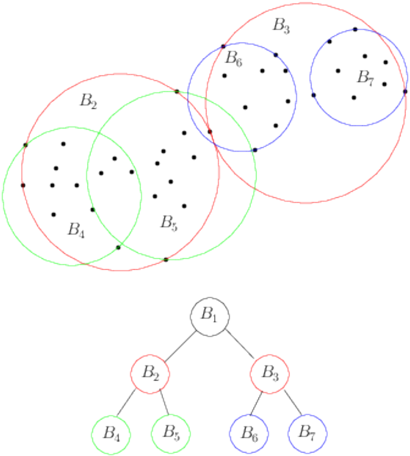


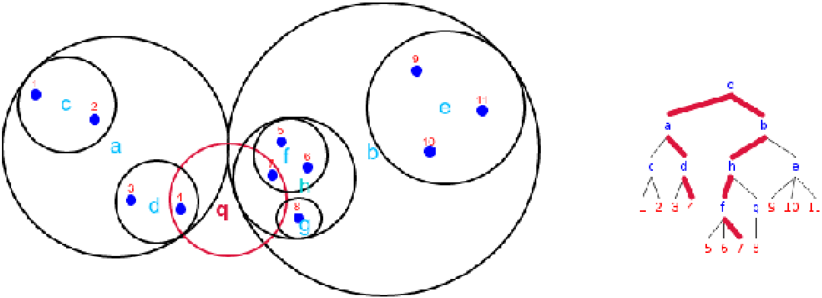
Figure 6.9

Number of inspections graphed against tree dimension. In these experiments the points had an underlying distribution with the same dimensionality as the tree.

Ball trees



Ball tree search



(a) (b)

Acknowledgment

- The kdtrees animations were borrowed from
    - Thinh Nguyen's slides
    - Carl Kingsford's slides
- Andrew moore's tutorial