15-826: Multimedia Databases and Data Mining

Lecture #8: Spatial Access Methods - V
Metric trees, kNN methods

C. Faloutsos

Must-read material

- Textbook, Chapter 5
- Roberto F. Santos Filho, Agma Traina, Caetano Traina Jr., and Christos Faloutsos: *Similarity search without tears: the OMNI family of all-purpose access methods* ICDE, Heidelberg, Germany, April 2-6 2001.
  (code at www.cs.cmu.edu/~christos/SRC/OmniUsrKit.tar.gz)

Outline

Goal: ‘Find similar / interesting things’
- Intro to DB
- Indexing - similarity search
- Data Mining
Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
  - problem dfn
  - z-ordering
  - R-trees
- misc
- fractals
- text

SAMs - Detailed outline

- spatial access methods
  - problem dfn
  - z-ordering
  - R-trees
  - misc topics
    - grid files
    - dimensionality curse; dim. reduction
    - metric trees
    - other nn methods
- fractals
- text, ...

Metric trees

- What if we only have a distance function \( d(o1, o2) \)?
- (Applications?)
Metric trees

• (assumption: d() is a metric: positive; symmetric; triangle inequality)
• then, we can use some variation of ‘Vantage Point’ trees [Yannilos]
• many variations (GNAT trees [Brin95], MVP-trees [Ozsoyoglu+] ...)

• Finally: M-trees [Ciaccia, Patella, Zezula, vldb 97]
  • M-trees = ‘ball-trees’: Minimum Bounding spheres
Metric trees

• Search (range and k-nn): like R-trees
• Split?

• Split? Several criteria:
  – minimize max radius (or sum radii)
  – (even: random!)
• Algorithm?

• eg., similar to the quadratic split of Guttman
**Metric trees - variations**

- Slim trees [Traina+, EDBT2000]
- OMNI tree [Filho+, ICDE2001]

**Metric trees - Slim trees**

- How to improve the structure?

**BEFORE**

**AFTER**

- Idea: give-away contents, if it decreases the radius - eg:
Metric trees - Slim trees

• How to accelerate the splitting time (O(N**3), currently)?

Metric trees - Slim trees

• Split using Minimum Spanning Tree (drop longest edge)
Metric trees - Slim trees

• Split using Minimum Spanning Tree (drop longest edge)

Metric trees - Slim trees

• result: at least as fast as M-trees for search
• MST: significantly faster for split, with tiny performance penalty

SAMs - Detailed outline

• spatial access methods
  – problem def
  – z-ordering
  – R-trees
  – misc topics
    • grid files
    • dimensionality curse; dim. reduction
    • metric trees
    • other nn methods
• fractals
• text, ...
**Metric trees - OMNI trees**

- How to turn objects into vectors?
- (assume that distance computations are expensive; we need to answer range/nn queries quickly)

A: pick n ‘anchor’ objects; record the distance of each object from them -> n-d vector

[Diagram of objects and distances]

A: pick n ‘anchor’ objects; record the distance of each object from them -> n-d vector

[Another diagram of objects and distances]
Metric trees - OMNI trees

• How to turn objects into vectors?
• A: pick $n$ ‘anchor’ objects; record the distance of each object from them -> $n$-d vector

Metric trees - OMNI trees

• we could put OMNI coordinates in R-tree (or other SAM, or even do seq. scan)
• and still answer range and nn queries! (see [Filho’01] for details)

Metric trees - OMNI trees

• Result: faster than M-trees and seq. scanning (especially if distance computations are expensive)
Metric trees - OMNI trees
• Q1: how to choose anchors?
• Q2: ... and how many?

SAMs - Detailed outline
• spatial access methods
  – problem dfn
  – z-ordering
  – R-trees
  – misc topics
    • grid files
    • dimensionality curse; dim. reduction
    • metric trees
    • other nn methods
• fractals
• text, ...

Other nn methods
• Problem#1: vector space
• Problem#2: metric space
Other nn methods - vector space

- Solution#1: k-d trees (or R-trees etc)
- Solution#2: Spiral search - put a grid; spiral out (O(1) on the average, but: exp(?) on dimension) [Bentley+, 80]

- what if we have no index (grid / kd-tree / R-tree)?
Other nn methods - vector space

• what if we have no index (grid / kd-tree / R-tree)?
• Solution#3: [Friedman+75] Project on 1-d & sort
Other nn methods - vector space

• what if we have no index (grid / kd-tree / R-tree)?
• Solution#3: [Friedman+75] Project on 1-d & sort

Other nn methods - vector space

• Solution#4: Voronoi diagrams (eg.,
  [Aurenhammer ‘91]
  - mainly, in comp. geometry
  - tricky for \( \geq 3 \) dimensions

Other nn methods

• Problem#1: vector space
• Problem#2: metric space
Other nn methods - metric space

• Solution#1: Branch+Bound [Fukunaga+, ‘73]. Very similar to nn in M-trees:

\[ \text{C1} \]
\[ \text{C2} \]

• Solution#2: Anchor points [Burkhard+, 73]

anchor

\[ q \]
Other nn methods - metric space

• Solution#2: Anchor points [Burkhard+, 73]
• variations: [Shapiro, ‘77], [Shasha+, ‘90]
• related to metric trees

Conclusions

• Metric trees (= “sphere/ball trees”) for metric spaces
  – M-trees / OMNI-trees
• several clever methods for nn search
  – branch + bound
  – anchors

SAMs - Detailed outline

• spatial access methods
  – problem dfn
  – z-ordering
  – R-trees
  – misc topics
    • grid files
    • dimensionality curse; dim. reduction
    • metric trees
    • other nn methods
• fractals
• text, ...
Conclusions for SAMs

- z-ordering and R-trees for low-d points and regions
- M-trees & variants for metric datasets
- beware of the ‘dimensionality curse’

References

References


References


References