


CMU SCS

15-826: Multimedia Databases and Data Mining

Lecture #7: Spatial Access Methods -
Metric trees
C. Faloutsos



CMU SCS

Must-read material

- MM 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)

15-826 Copyright: C. Faloutsos (2017) #2




CMU SCS

Outline

Goal: 'Find **similar / interesting** things'

- Intro to DB
- ➔ • Indexing - similarity search
- Data Mining

15-826 Copyright: C. Faloutsos (2017) #3



CMU SCS

Indexing - Detailed outline

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

15-826 Copyright: C. Faloutsos (2017) #4

CMU SCS

SAMs - Detailed outline

- spatial access methods
 - problem defn
 - z-ordering
 - R-trees
 - misc topics
 - metric trees
- fractals
- text, ...

15-826 Copyright: C. Faloutsos (2017) #5

CMU SCS

Metric trees

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

15-826 Copyright: C. Faloutsos (2017) #6

CMU SCS

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+] ...)

15-826 Copyright: C. Faloutsos (2017) #7

CMU SCS

Metric trees

- Finally: M-trees [Ciaccia, Patella, Zezula, vldb 97]
- M-trees = 'ball-trees' : groups in spheres

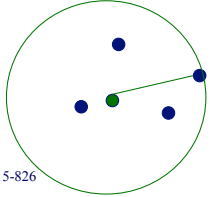


15-826 Copyright: C. Faloutsos (2017) #8

CMU SCS

Metric trees

- Finally: M-trees [Ciaccia, Patella, Zezula, vldb 97]
- M-trees = 'ball-trees' : Minimum Bounding spheres

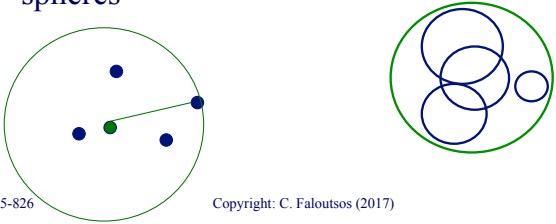


15-826 Copyright: C. Faloutsos (2017) #9

CMU SCS

Metric trees

- Finally: M-trees [Ciaccia, Patella, Zezula, vldb 97]
- M-trees = 'ball-trees' : Minimum Bounding spheres

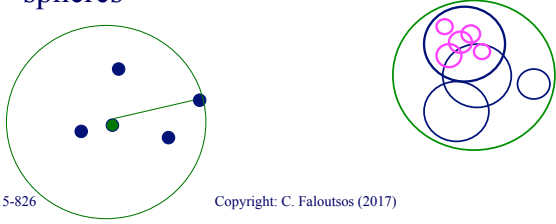


15-826 Copyright: C. Faloutsos (2017) #10

CMU SCS

Metric trees

- Finally: M-trees [Ciaccia, Patella, Zezula, vldb 97]
- M-trees = 'ball-trees' : Minimum Bounding spheres




15-826 Copyright: C. Faloutsos (2017) #11


CMU SCS

Metric trees


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



15-826 Copyright: C. Faloutsos (2017) #12




CMU SCS




Metric trees

- Search (range and k-nn): like R-trees
- Split? Several criteria:
 - minimize max radius (or sum radii)
 - (even: random!)
- Algorithm?

15-826 Copyright: C. Faloutsos (2017) #13




CMU SCS



Metric trees

- Search (range and k-nn): like R-trees
- Split? Several criteria:
 - minimize max radius (or sum radii)
 - (even: random!)
- Algorithm?
- eg., similar to the quadratic split of Guttman

15-826 Copyright: C. Faloutsos (2017) #14




CMU SCS

Metric trees - variations

- OMNI tree [Filho+, ICDE2001]

15-826 Copyright: C. Faloutsos (2017) #15



CMU SCS

Metric trees - OMNI trees


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

15-826 Copyright: C. Faloutsos (2017) #16

CMU SCS

Metric trees - OMNI trees

- How to turn objects into vectors?
- A: pick n 'anchor' objects; record the distance of each object from them $\rightarrow n$ -d vector

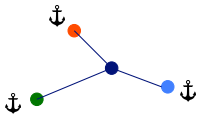


15-826 Copyright: C. Faloutsos (2017) #17

CMU SCS

Metric trees - OMNI trees

- How to turn objects into vectors?
- A: pick n 'anchor' objects; record the distance of each object from them $\rightarrow n$ -d vector

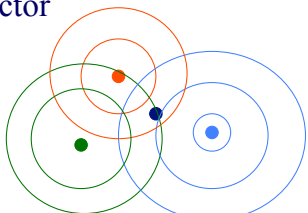


15-826 Copyright: C. Faloutsos (2017) #18

CMU SCS

Metric trees - OMNI trees

- How to turn objects into vectors?
- A: pick n 'anchor' objects; record the distance of each object from them $\rightarrow n$ -d vector

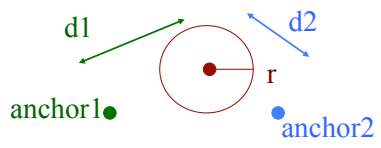


15-826 Copyright: C. Faloutsos (2017) #19

CMU SCS

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)



15-826 Copyright: C. Faloutsos (2017) #20

CMU SCS

OMNI trees – range queries

- and still answer range and nn queries! (see [Filho' 01] for details)

d1	d2
5.2	3.1
1.1	2.8
...	...

15-826
Copyright: C. Faloutsos (2017)
#21

CMU SCS

OMNI trees – range queries

- Result: faster than M-trees and seq. scanning (especially if distance computations are expensive)

d1	d2
5.2	3.1
1.1	2.8
...	...

15-826
Copyright: C. Faloutsos (2017)
#22

CMU SCS

Metric trees - OMNI trees

- Q1: how to choose anchors?
- Q2: ... and how many?


15-826
Copyright: C. Faloutsos (2017)
#23

CMU SCS

Conclusions for SAMs

- z-ordering and R-trees for low-d points and regions – **very successful**
- M-trees & variants for metric datasets
- beware of the ‘dimensionality curse’
 - Estimate ‘intrinsic’ dimensionality (‘fractals’)
 - Project to lower dimensions (‘SVD/PCA’)

15-826
Copyright: C. Faloutsos (2017)
#24




CMU SCS

References

- Aurenhammer, F. (Sept. 1991). "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure." *ACM Computing Surveys* 23(3): 345-405.
- Bentley, J. L., B. W. Weide, et al. (Dec. 1980). "Optimal Expected-Time Algorithms for Closest Point Problems." *ACM Trans. on Mathematical Software (TOMS)* 6(4): 563-580.
- Burkhard, W. A. and R. M. Keller (Apr. 1973). "Some Approaches to Best-Match File Searching." *Comm. of the ACM (CACM)* 16(4): 230-236.

15-826 Copyright: C. Faloutsos (2017) #25




CMU SCS

References

- Christian Böhm, Stefan Berchtold, Daniel A. Keim: [*Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases.*](#) *ACM Comput. Surv.* 33(3): 322-373 (2001)
- Edgar Chávez, Gonzalo Navarro, Ricardo A. Baeza-Yates, José L. Marroquín: [*Searching in metric spaces.*](#) *ACM Comput. Surv.* 33(3): 273-321 (2001)

15-826 Copyright: C. Faloutsos (2017) #26




CMU SCS

References

- Ciaccia, P., M. Patella, et al. (1997). M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. *VLDB*.
- Filho, R. F. S., A. Traina, et al. (2001). Similarity search without tears: the OMNI family of all-purpose access methods. *ICDE*, Heidelberg, Germany.
- Friedman, J. H., F. Baskett, et al. (Oct. 1975). "An Algorithm for Finding Nearest Neighbors." *IEEE Trans. on Computers (TOC)* C-24: 1000-1006.

15-826 Copyright: C. Faloutsos (2017) #27




CMU SCS

References

- Fukunaga, K. and P. M. Narendra (July 1975). "A Branch and Bound Algorithm for Computing k-Nearest Neighbors." *IEEE Trans. on Computers (TOC)* C-24(7): 750-753.
- Shapiro, M. (May 1977). "The Choice of Reference Points in Best-Match File Searching." *Comm. of the ACM (CACM)* 20(5): 339-343.

15-826 Copyright: C. Faloutsos (2017) #28



CMU SCS

References

- Shasha, D. and T.-L. Wang (Apr. 1990). “New Techniques for Best-Match Retrieval.” ACM TOIS 8(2): 140-158.
- Traina, C., A. J. M. Traina, et al. (2000). Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes. EDBT, Konstanz, Germany.

15-826 Copyright: C. Faloutsos (2017) #29