CMU SCS

# 15-826: Multimedia Databases and Data Mining

Lecture #10: Fractals: M-trees and dim. curse (case studies – Part II)

*C. Faloutsos*

CMU SCS

## Must-read Material

- Alberto Belussi and Christos Faloutsos, Estimating the Selectivity of Spatial Queries Using the `Correlation' Fractal Dimension Proc. of VLDB, p. 299-310, 1995



CMU SCS

## Optional Material

Optional, but **very** useful: Manfred Schroeder *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise* W.H. Freeman and Company, 1991

## 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
  - z-ordering
  - R-trees
  - misc
- fractals
  - intro
  - applications
- text
- ...





## Indexing - Detailed outline

- fractals
  - intro
  - applications
    - disk accesses for R-trees (range queries)
    - dimensionality reduction
    - selectivity in M-trees
    - dim. curse revisited
    - "fat fractals"
    - quad-tree analysis [Gaede+]

CMU SCS

## What else can they solve?

- ✓ separability [KDD’02]
- forecasting [CIKM’02]
- ✓ dimensionality reduction [SBBD’00]
- non-linear axis scaling [KDD’02]
- ✓ disk trace modeling [Wang+’02]
- ➡ selectivity of spatial/multimedia queries [PODS’94, VLDB’95, ICDE’00]
- ...



CMU SCS

Optional

## Metric trees - analysis

- Problem: How many disk accesses, for an M-tree?
- Given:
  - N (# of objects)
  - C (fanout of disk pages)
  - r (radius of range query - BIASED model)



CMU SCS

Optional

## Metric trees - analysis

- Problem: How many disk accesses, for an M-tree?
- Given:
  - N (# of objects)
  - C (fanout of disk pages)
  - r (radius of range query - BIASED model)
- NOT ENOUGH - what else do we need?

CMU SCS

Optional

## Metric trees - analysis

- A: something about the distribution



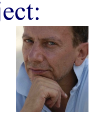
CMU SCS

Optional

## Metric trees - analysis

- A: something about the distribution

[Ciaccia, Patella, Zezula, PODS98]: assumed that the distance distribution is the same, for every object:



Paolo Ciaccia

Marco Patella



CMU SCS

Optional

## Metric trees - analysis

- A: something about the distribution

[Ciaccia+, PODS98]: assumed that the distance distribution is the same, for every object:

F1(d) = Prob(an object is within d from object #1)

= F2(d) = ... = F(d)

CMU SCS

Optional

## Metric trees - analysis

- A: something about the distribution
- Given our ‘fractal’ tools, we could try them - which one?



CMU SCS

Optional

## Metric trees - analysis

- A: something about the distribution
- Given our ‘fractal’ tools, we could try them - which one?
- A: Correlation integral [Traina+, ICDE2000]



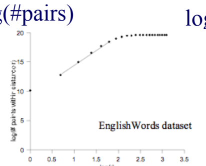
CMU SCS

Optional

## Metric trees - analysis

English dictionary

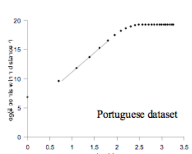
log(#pairs)




log(d)

Portuguese dictionary

log(#pairs)




log(d)

CMU SCS

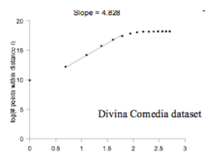
Optional

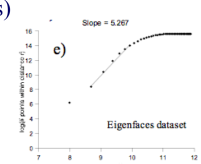
# Metric trees - analysis

Divina Comedia

Eigenfaces

log(#pairs) log(#pairs)

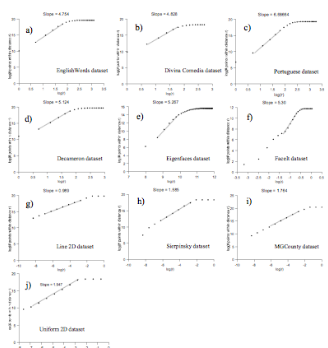



log(d) log(d)



CMU SCS






CMU SCS

Optional

# Metric trees - analysis

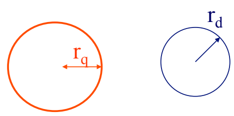
| | Data Set | N (# Objects) | Dimension | Distance Function | *Distance Exponent D* |
|---|---|---|---|---|---|
| Real Metric datasets | English | 25,143 | NA | $L_{Edit}$ | 4.753 |
| | Divina Commedia | 12,701 | NA | $L_{Edit}$ | 4.827 |
| | Decamerone | 18,719 | NA | $L_{Edit}$ | 5.124 |
| | Portuguese | 21,473 | NA | $L_{Edit}$ | 6.686 |
| | FaceIt | 1,056 | NA | Not divulged | 6.821 |
| Real vector datasets | MGCounty | 15,559 | 2 | $L_2$ | 1.752 |
| | Eigenfaces | 11,900 | 16 | $L_2$ | 5.267 |
| Synthetic datasets | Sierpinsky | 9,841 | 2 | $L_2$ | 1.584 |
| | 2D Line | 20,000 | 2 | $L_2$ | 0.989 |
| | Uniform 2D | 10,000 | 2 | $L_2$ | 1.947 |

CMU SCS

Optional

## Metric trees - analysis

- So, what is the # of disk accesses, for a node of radius $r_d$, on a query of radius $r_q$?



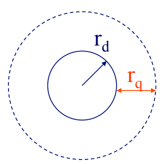



CMU SCS

Optional

## Metric trees - analysis

- So, what is the # of disk accesses, for a node of radius $r_d$, on a query of radius $r_q$?
- A: ~ $(r_d+r_q)$....



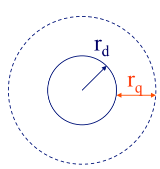



CMU SCS

Optional

## Metric trees - analysis

- So, what is the # of disk accesses, for a node of radius $r_d$, on a query of radius $r_q$?
- A: ~ $(r_d+r_q)$^***D***

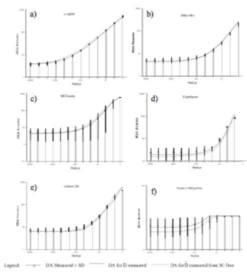


CMU SCS

Optional

## Accuracy of selectivity formulas

log(#d.a.)



log(rq)



CMU SCS

Optional

## Fast estimation of *D*

- Normally, ***D*** takes O(N^2) time
- Anything faster? suppose we have already built an M-tree

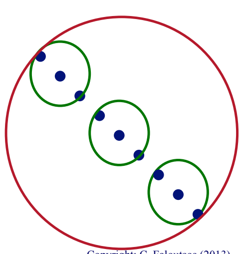


CMU SCS

Optional

## Fast estimation of *D*

- Hint:



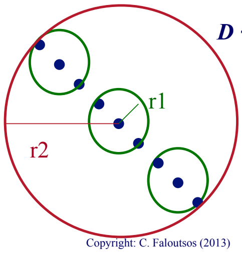
CMU SCS

Optional

## Fast estimation of D

- Hint:

ratio of radii:
r1^***D*** * C = r2^***D***
***D*** ~ log(C) / log(r2/r1)




15-826 Copyright: C. Faloutsos (2013) 25

CMU SCS

## Indexing - Detailed outline

- fractals
  - intro
  - applications
    - ✔ disk accesses for R-trees (range queries)
    - ✔ dimensionality reduction
    - ✔ selectivity in M-trees
    - ➡ dim. curse revisited
    - "fat fractals"
    - quad-tree analysis [Gaede+]

15-826 Copyright: C. Faloutsos (2013) 26

CMU SCS

## Dimensionality 'curse'

- Q: What is the problem in high-d?

15-826 Copyright: C. Faloutsos (2013) #27

CMU SCS

## Dimensionality 'curse'

- Q: What is the problem in high-d?
- A: indices do not seem to help, for many queries (eg., k-nn)
  - in high-d (& uniform distributions), most points are equidistant -> k-nn retrieves too many near-neighbors
  - [Yao & Yao, '85]: search effort ~ O( $N^{(1-1/d)}$ )



CMU SCS

## Dimensionality 'curse'

- (counter-intuitive, for db mentality)
- Q: What to do, then?



CMU SCS

## Dimensionality 'curse'

- A1: switch to seq. scanning
- A2: dim. reduction
- A3: consider the 'intrinsic'/fractal dimensionality
- A4: find approximate nn

CMU SCS

## Dimensionality 'curse'

- A1: switch to seq. scanning
  - X-trees [Kriegel+, VLDB 96]
  - VA-files [Schek+, VLDB 98], 'test of time' award



CMU SCS

## Dimensionality 'curse'

- A1: switch to seq. scanning
- ➡ A2: dim. reduction
- A3: consider the 'intrinsic'/fractal dimensionality
- A4: find approximate nn



CMU SCS

## Dim. reduction

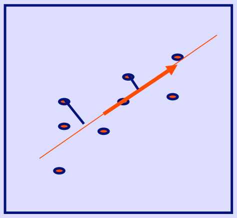
a.k.a. feature selection/extraction:

- SVD (optimal, to preserve Euclidean distances)
- random projections
- using the fractal dimension [Traina+ SBBD2000]

CMU SCS

## Singular Value Decomposition (SVD)

- SVD (~LSI ~ KL ~ PCA ~ spectral analysis...)



LSI: S. Dumais; M. Berry

KL: eg, Duda+Hart

PCA: eg., Jolliffe

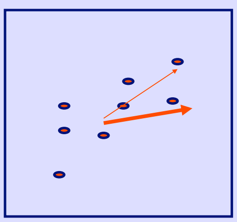
MANY more details: soon



CMU SCS

## Random projections

- random projections(Johnson-Lindenstrauss thm [Papadimitriou+ pods98])





CMU SCS

## Random projections

- pick ‘enough’ random directions (will be ~orthogonal, in high-d!!)
- distances are preserved probabilistically, within epsilon
- (also, use as a pre-processing step for SVD [Papadimitriou+ PODS98]

CMU SCS

## Dim. reduction - w/ fractals

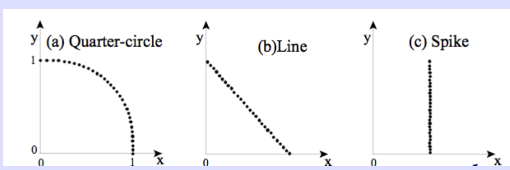
- Main idea: drop those attributes that don't affect the intrinsic ('fractal') dimensionality [Traina+, SBBD 2000]

15-826  #37

CMU SCS

## Dim. reduction - w/ fractals

global FD=1




15-826  #38

CMU SCS

## Dimensionality 'curse'

- A1: switch to seq. scanning
- A2: dim. reduction
- A3: consider the 'intrinsic'/fractal dimensionality
- A4: find **approximate** nn

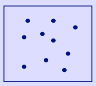
15-826  #39

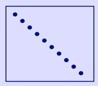
CMU SCS

## Intrinsic dimensionality

- before we give up, compute the intrinsic dim.:
- the lower, the better... [Pagel+, ICDE 2000]
- more details: under ‘fractals’

intr. d = 2





intr. d = 1

15-826  #40

CMU SCS

## Dimensionality ‘curse’

- A1: switch to seq. scanning
- A2: dim. reduction
- A3: consider the ‘intrinsic’/fractal dimensionality
- A4: find approximate nn

15-826  #41

CMU SCS

## Approximate nn

- [Arya + Mount, SODA93], [Patella+ ICDE 2000]
- Idea: find k neighbors, such that the distance of the k-th one is guaranteed to be within epsilon of the actual.

15-826  #42

CMU SCS

## Dimensionality ‘curse’

- A1: switch to seq. scanning
- A2: dim. reduction
- A3: consider the ‘intrinsic’/fractal dimensionality
- A4: find approximate nn



CMU SCS

## Dim. curse revisited

- (Q: how serious is the dim. curse, e.g.:)
- Q: what is the search effort for k-nn?
  - given N points, in E dimensions, in an R-tree, with k-nn queries (‘biased’ model)

[Pagel, Korn + ICDE 2000]




CMU SCS

## (Overview of proofs)

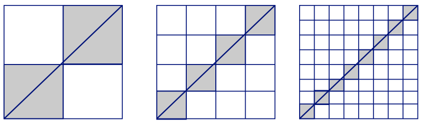
- assume that your points are uniformly distributed in a *d*-dimensional manifold (= hyper-plane)
- derive the formulas
- substitute *d* for the fractal dimension

CMU SCS

proof

## Reminder: Hausdorff Dimension ($D_0$)

- $r$ = side length (each dimension)
- $B(r)$ = # boxes containing points $\propto r^{D0}$



$r = 1/2 \;\; B = 2$ — $\log r = -1$, $\log B = 1$

$r = 1/4 \;\; B = 4$ — $\log r = -2$, $\log B = 2$

$r = 1/8 \;\; B = 8$ — $\log r = -3$, $\log B = 3$

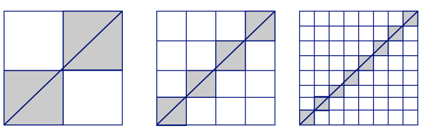
15-826 Copyright: C. Faloutsos (2013) 46

CMU SCS

proof

## Reminder: Correlation Dimension ($D_2$)

- $S(r) = \sum p_i^2$ (squared % pts in box) $\propto r^{D2}$
  $\propto$ #pairs( within <= r )



$r = 1/2 \;\; S = 1/2$ — $\log r = -1$, $\log S = -1$

$r = 1/4 \;\; S = 1/4$ — $\log r = -2$, $\log S = -2$

$r = 1/8 \;\; S = 1/8$ — $\log r = -3$, $\log S = -3$

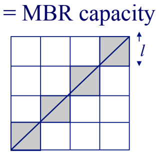
15-826 Copyright: C. Faloutsos (2013) 47

CMU SCS

proof

## Observation #1

- How to determine avg MBR side $l$?
  - $N$ = #pts, $C$ = MBR capacity



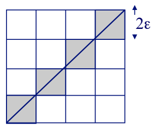

Hausdorff dimension: $B(r) \propto r^{D0}$

$B(l) = N/C = l^{-D0} \Rightarrow l = (N/C)^{-1/D0}$

15-826 Copyright: C. Faloutsos (2013) 48 proof

## Observation #2

- $k$-NN query → ε-range query
  - For $k$ pts, what radius ε do we expect?




Correlation dimension: $S(r) \propto r^{D2}$

$$S(\varepsilon) = \frac{k}{N-1} = (2\varepsilon)^{D2}$$

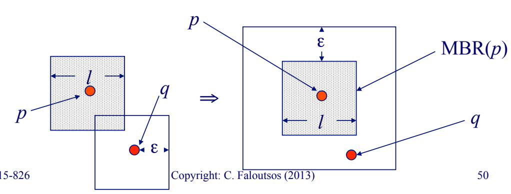




proof

## Observation #3

- Estimate avg # query-sensitive anchors:
  - How many **expected** $q$ will touch **avg** page?
  - Page touch: $q$ stabs ε-dilated MBR($p$)








## Asymptotic Formula

- $k$-NN page accesses as N → ∞
  - $C$ = page capacity
  - $D$ = fractal dimension (=$D0 \sim D2$)

$$P_{all}^{L\infty}(k) \approx \sum_{j=0}^{h} \left\{ \frac{1}{C^{h-j}} + \left[ 1 + \left( \frac{k}{C^{h-j}} \right)^{1/D} \right]^{D} \right\}$$

## Asymptotic Formula

$$P_{all}^{L\infty}(k) \approx \sum_{j=0}^{h} \left\{ \frac{1}{C^{h-j}} + \left[ 1 + \left( \frac{k}{C^{h-j}} \right)^{1/D} \right]^{D} \right\}$$

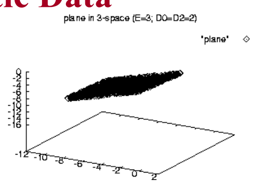
- NO mention of the embedding dimensionality!!
- Still have dim. curse, but on f.d. *D*

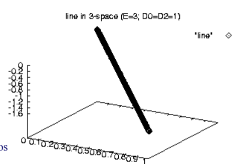




## Synthetic Data

- `plane`
  - $D_0 = D_2 = 2$
  - embedded in *E*-space
  - $N = 100K$
- `manifold`
  - $E = 8$
  - $D_0$=$D_2$ varies from 1-6
  - line, plane, *etc.* (in 8-d)

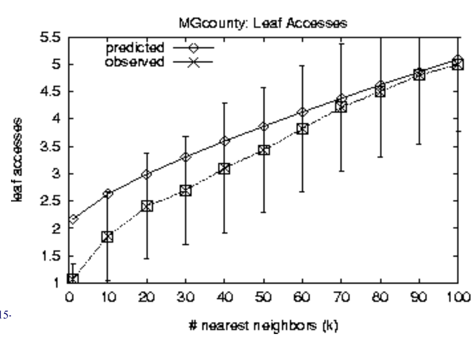


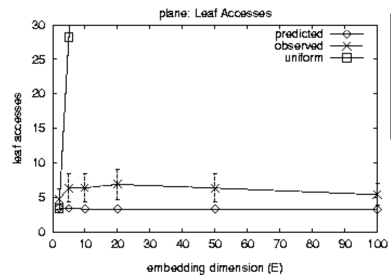





## Accuracy of $L_\infty$ Formula



## Embedding Dimension



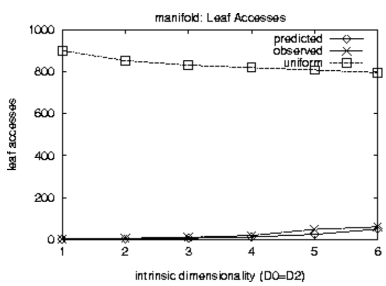

| E | unif ind | f ractal | leaf |
|---|---|---|---|
| 2 | 3.49 | 3.49 | 4.75 |
| 5 | 28.26 | 3.45 | 6.40 |
| 10 | 847.26 | 3.34 | 6.42 |
| 20 | All | 3.36 | 6.9 |
| 50 | All | 3.32 | 6.37 |
| 100 | All | 3.32 | 5.43 |

`plane`
$k = 50$
$L_\infty$ dist





## Intrinsic Dimensionality




| D | unif ind | f ractal | leaf |
|---|---|---|---|
| 1 | 899.31 | 1.79 | 3.55 |
| 2 | 851.1 | 3.36 | 6.56 |
| 3 | 829.89 | 6.53 | 11.04 |
| 4 | 818.32 | 12.85 | 20.21 |
| 5 | 806.89 | 25.30 | 48.9 |
| 6 | 794.35 | 49.77 | 60.94 |

`manifold`
$k = 50$
$L_\infty$ dist





## Non-Euclidean Data Set

| E | unif ind | f ractal | leaf |
|---|---|---|---|
| 2 | 3.49 | 2.53 | 4.72±1.81 |
| 10 | 847.26 | 2.53 | 6.42±2.11 |
| 20 | all | 2.53 | 7.76±4.12 |
| 50 | all | 2.53 | 6.15±2.82 |
| 100 | all | 2.53 | 5.64±2.32 |

`sierpinski`, $k = 50$, $L_\infty$ dist

CMU SCS

## Conclusions

- Dimensionality 'curse':
  - for high-d, indices slow down to ~O(N)
- If the **intrinsic** dim. is low, there is hope
- otherwise, do seq. scan, or sacrifice accuracy (approximate nn)

15-826 Copyright: C. Faloutsos (2013) #58

CMU SCS

## Conclusions – cont'd

- Worst-case theory is **over-pessimistic**
- High dimensional data can exhibit good performance if **correlated, non-uniform**
- Many real data sets are **self-similar**
- Determinant is **intrinsic** dimensionality
  - multiple fractal dimensions ($D_0$ and $D_2$)
  - indication of how far one can go

15-826 Copyright: C. Faloutsos (2013) 59

CMU SCS

## References


- Sunil Arya, David M. Mount: *Approximate Nearest Neighbor Queries in Fixed Dimensions*. SODA 1993: 271-280
  ANN library:
  http://www.cs.umd.edu/~mount/ANN/


15-826 Copyright: C. Faloutsos (2013) #60

CMU SCS

## References


- Berchtold, S., D. A. Keim, et al. (1996). The X-tree : An Index Structure for High-Dimensional Data. VLDB, Mumbai (Bombay), India.
- Ciaccia, P., M. Patella, et al. (1998). *A Cost Model for Similarity Queries in Metric Spaces*. PODS.


15-826  #61

CMU SCS

## References cnt'd


- Nievergelt, J., H. Hinterberger, et al. (March 1984). "The Grid File: An Adaptable, Symmetric Multikey File Structure." ACM TODS 9(1): 38-71.
- Pagel, B.-U., F. Korn, et al. (2000). *Deflating the Dimensionality Curse Using Multiple Fractal Dimensions*. ICDE, San Diego, CA.
- Papadimitriou, C. H., P. Raghavan, et al. (1998). Latent Semantic Indexing: A Probabilistic Analysis. PODS, Seattle, WA.


15-826  #62

CMU SCS

## References cnt'd


- Traina, C., A. J. M. Traina, et al. (2000). *Distance Exponent: A New Concept for Selectivity Estimation in Metric Trees*. ICDE, San Diego, CA.
- Weber, R., H.-J. Schek, et al. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in high-dimensional spaces. VLDB, New York, NY.


15-826  #63

CMU SCS

## References cnt'd

- Yao, A. C. and F. F. Yao (May 6-8, 1985). A General Approach to d-Dimensional Geometric Queries. Proc. of the 17th Annual ACM Symposium on Theory of Computing (STOC), Providence, RI.

15-826 Copyright: C. Faloutsos (2013) #64