

**15-826: Multimedia Databases
and Data Mining**


Lecture #10: Fractals - case studies - I
C. Faloutsos



Must-read Material


- Christos Faloutsos and Ibrahim Kamel,
*Beyond Uniformity and Independence:
Analysis of R-trees Using the Concept of
Fractal Dimension*, Proc. ACM SIGACT
-SIGMOD-SIGART PODS, May 1994, pp.
4-13, Minneapolis, MN.

15-826 Copyright: C. Faloutsos (2010) 2



Optional Material

Optional, but **very** useful: Manfred Schroeder
*Fractals, Chaos, Power Laws: Minutes
from an Infinite Paradise* W.H. Freeman
and Company, 1991 (on reserve in the
WeH library)



15-826 Copyright: C. Faloutsos (2010) 3

CMU SCS

Outline

Goal: 'Find similar / interesting things'

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

15-826 Copyright: C. Faloutsos (2010) 4

CMU SCS

Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
 - z-ordering
 - R-trees
 - misc
- ➔ • fractals
 - intro
 - applications
- text

15-826 Copyright: C. Faloutsos (2010) 5

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 (2010) 6

CMU SCS

(Fractals mentioned before:)

- for performance analysis of R-trees
- fractals for dim. reduction

15-826 Copyright: C. Faloutsos (2010) 7

CMU SCS

Case study#1: R-tree performance

Problem

- Given
 - N points in E-dim space
- Estimate # disk accesses for a range query ($q_1 \times \dots \times q_E$)

(assume: 'good' R-tree, with tight, cube-like MBRs)

15-826 Copyright: C. Faloutsos (2010) 8

CMU SCS

Case study#1: R-tree performance

Problem

- Given
 - N points in E-dim space
 - with fractal dimension D
- Estimate # disk accesses for a range query ($q_1 \times \dots \times q_E$)

(assume: 'good' R-tree, with tight, cube-like MBRs)
Typically, in DB Q-opt: uniformity + independence

15-826 Copyright: C. Faloutsos (2010) 9

CMU SCS

Examples: World's countries

- BUT: area vs population for ~200 countries (1991 CIA fact-book).

pop log(pop)

area log(area)

15-826 Copyright: C. Faloutsos (2010) 10

CMU SCS

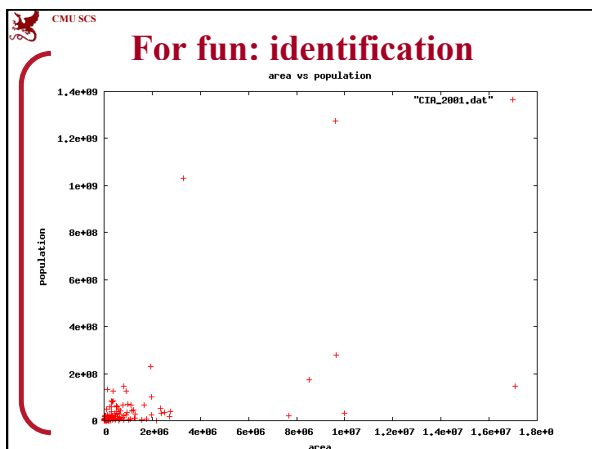
Examples: World's countries

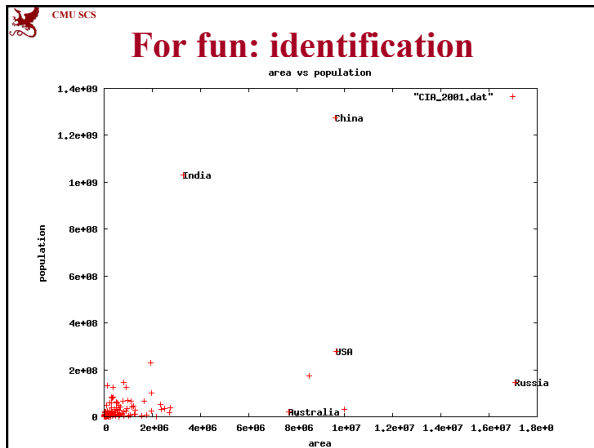
- neither uniform, nor independent!

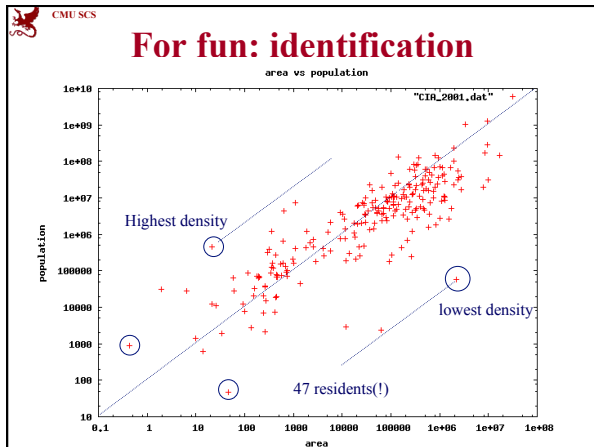
pop log(pop)

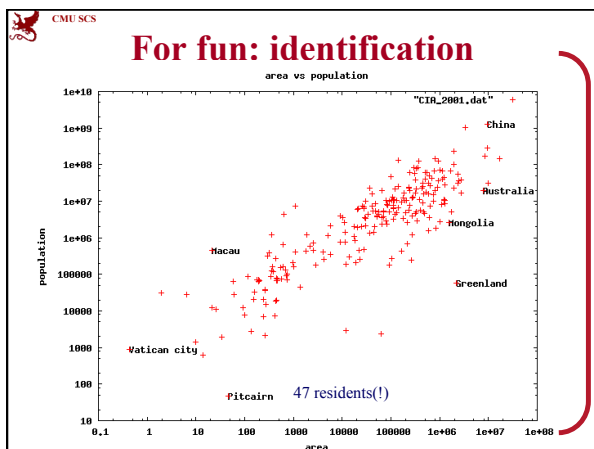
area log(area)

15-826 Copyright: C. Faloutsos (2010) 11







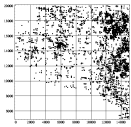


CMU SCS

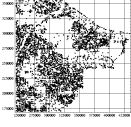
Examples: TIGER files

- neither uniform, nor independent!

MG county



LB county



15-826 Copyright: C. Faloutsos (2010) 16

CMU SCS

How to proceed?

- recall the [Page+]₁ formula, for range queries of size q1 x q2

$$\#DiskAccesses(q1, q2) = \sum (x_{i,1} + q1) * (x_{i,2} + q2)$$

But:
formula needs to know the $x_{i,j}$ sizes of MBRs!

15-826 Copyright: C. Faloutsos (2010) 17

CMU SCS


How to proceed?

But:
formula needs to know the $x_{i,j}$ sizes of MBRs!

Answer (jumping ahead):

$$s = (C/N)^{1/D0}$$

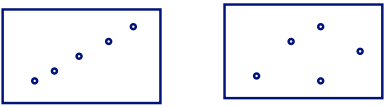
15-826 Copyright: C. Faloutsos (2010) 18

CMU SCS 


R-trees - performance analysis

Q: OK - so we are told that the **Hausdorff** fractal dim. = D_0 - Next step?
 (also know that there are at most C points per page)

$D_0=1$ $D_0=2$




15-826 Copyright: C. Faloutsos (2010) 22

CMU SCS 

R-trees - performance analysis

Hint: dfn of Hausdorff f.d.:



Felix Hausdorff (1868-1942)

15-826 Copyright: C. Faloutsos (2010) 23

CMU SCS

Reminder: Hausdorff or box-counting fd:

- Box counting plot: $\text{Log}(N(r))$ vs $\text{Log}(r)$
- r : grid side
- $N(r)$: count of non-empty cells
- (Hausdorff) fractal dimension D_0 :

$$D_0 = - \frac{\partial \log(N(r))}{\partial \log(r)}$$

15-826 Copyright: C. Faloutsos (2010) 24

CMU SCS proof

Reminder

- Hausdorff fd:

$r \sim \log(\#\text{non-empty cells})$

SLOPE = -1.5743

15-826 Copyright: C. Faloutsos (2010) 25

CMU SCS proof

Reminder

- dfn of Hausdorff fd implies that

$$N(r) \sim r^{-D_0}$$

↙

non-empty cells of side r

15-826 Copyright: C. Faloutsos (2010) 26

CMU SCS proof

R-trees - performance analysis

Q (rephrased): what is the side s_1, s_2, \dots of parent nodes, given N data points, packed by C , with f.d. = D_0

$D_0=1$

$D_0=2$

15-826 Copyright: C. Faloutsos (2010) 27

CMU SCS proof

R-trees - performance analysis

Q (rephrased): what is the side s_1, s_2, \dots of parent nodes, given N data points, packed by C , with f.d. = D_0

$D_0=1$

$D_0=2$

15-826 Copyright: C. Faloutsos (2010) 28

CMU SCS proof

R-trees - performance analysis

Q (rephrased): what is the side s_1, s_2, \dots of parent nodes, given N data points, packed by C , with f.d. = D_0

$D_0=1$

$D_0=2$

$s_1 = s_2 = s$

15-826 Copyright: C. Faloutsos (2010) 29

CMU SCS proof

R-trees - performance analysis

A: (educated guess)


- $s=s_1=s_2 (= \dots)$ - square-like MBRs
- N/C non-empty cells = $K * s^{(-D_0)}$

$D_0=1$

$D_0=2$

$\log(s)$

15-826 Copyright: C. Faloutsos (2010) 30


CMU SCS 

R-trees - performance analysis

Details of derivations: in [PODS 94].
 Finally, expected side s of parent MBRs:
 $s = (C/N)^{1/D0}$

Q: sanity check: how does s change with $D0$?
 A:

15-826 Copyright: C. Faloutsos (2010) 31

CMU SCS 

R-trees - performance analysis

Details of derivations: in [Kamel+, PODS 94].
 Finally, expected side s of parent MBRs:
 $s = (C/N)^{1/D0}$

Q: sanity check: how does s change with $D0$?
 A: s grows with $D0$
 Q: does it make sense?
 Q: does it suffer from (intrinsic) dim. curse?

15-826 Copyright: C. Faloutsos (2010) 32

CMU SCS

R-trees - performance analysis

Q: Final-final formula (# disk accesses for range queries $q1$ x $q2$ x ...):
 A:

15-826 Copyright: C. Faloutsos (2010) 33

CMU SCS

R-trees - performance analysis

Q: Final-final formula (# disk accesses for range queries $q_1 \times q_2 \times \dots$):

A: # of parent-node accesses:

$$N/C * (s + q_1) * (s + q_2) * \dots * (s + q_E)$$

A: # of grand-parent node accesses

15-826 Copyright: C. Faloutsos (2010) 34

CMU SCS

R-trees - performance analysis

Q: Final-final formula (# disk accesses for range queries $q_1 \times q_2 \times \dots$):

A: # of parent-node accesses:

$$N/C * (s + q_1) * (s + q_2) * \dots * (s + q_E)$$

A: # of grand-parent node accesses

$$N/(C^2) * (s' + q_1) * (s' + q_2) * \dots * (s' + q_E)$$

$$s' = (C^2/N)^{1/D_0}$$

15-826 Copyright: C. Faloutsos (2010) 35

CMU SCS

R-trees - performance analysis

Results: IUE (x-y star coordinates)

leaf accesses

(a) IUE - Leaf accesses vs. query side


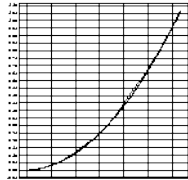
15-826 Copyright: C. Faloutsos (2010) 36

CMU SCS

R-trees - performance analysis

Results: LB County

leaf accesses



query side


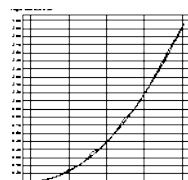
15-826 Copyright: C. Faloutsos (2010) 37

CMU SCS

R-trees - performance analysis

Results: MG-county

leaf accesses



query side

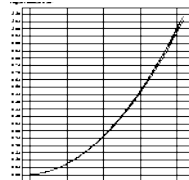
15-826 Copyright: C. Faloutsos (2010) 38

CMU SCS

R-trees - performance analysis

Results: 2D- uniform

leaf accesses



query side

15-826 Copyright: C. Faloutsos (2010) 39

CMU SCS

R-trees - performance analysis

Conclusions: usually, <5% relative error, for range queries

15-826 Copyright: C. Faloutsos (2010) 40

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 (2010) 41


CMU SCS

Case study #2: Dim. reduction


Problem definition: 'Feature selection'

- given N points, with E dimensions
- keep the k most 'informative' dimensions


[Traina+, SBBD'00]



Caetano
Traina



Agma
Traina



Leejay
Wu

15-826 Copyright: C. Faloutsos (2010) 42

CMU SCS

Dim. reduction - w/ fractals

(a) Quarter-circle (b) Line (c) Spike

not informative

15-826 Copyright: C. Faloutsos (2010) 43

CMU SCS

Dim. reduction

Problem definition: ‘Feature selection’

- given N points, with E dimensions
- keep the k most ‘informative’ dimensions

Re-phrased: spot and drop attributes with strong (non-)linear correlations

Q: how do we do that?

15-826 Copyright: C. Faloutsos (2010) 44

CMU SCS

Dim. reduction

A: Hint: correlated attributes do not affect the intrinsic/fractal dimension, e.g., if

$$y = f(x, z, w)$$

we can drop y

(hence: ‘*partial fd*’ (PFD) of a set of attributes = the fd of the dataset, when projected on those attributes)

15-826 Copyright: C. Faloutsos (2010) 45

CMU SCS

Dim. reduction - w/ fractals

global FD=1
PFD=1

(c) Spike

PFD=0

15-826 Copyright: C. Faloutsos (2010) 46

CMU SCS

Dim. reduction - w/ fractals

global FD=1
PFD=1

(b) Line

PFD=1

15-826 Copyright: C. Faloutsos (2010) 47

CMU SCS

Dim. reduction - w/ fractals

global FD=1
PFD~1

(a) Quarter-circle

PFD~1

15-826 Copyright: C. Faloutsos (2010) 48

CMU SCS

Dim. reduction - w/ fractals

- (problem: given N points in E -d, choose k best dimensions)
- Q: Algorithm?

15-826 Copyright: C. Faloutsos (2010) 49

CMU SCS

Dim. reduction - w/ fractals

- Q: Algorithm?
- A: e.g., greedy - forward selection:
 - keep the attribute with highest partial fd
 - add the one that causes the highest increase in pfd
 - etc., until we are within *epsilon* from the full f.d.

15-826 Copyright: C. Faloutsos (2010) 50

CMU SCS

Dim. reduction - w/ fractals

- (backward elimination: ~ reverse)
 - drop the attribute with least impact on the p.f.d.
 - repeat
 - until we are *epsilon* below the full f.d.

15-826 Copyright: C. Faloutsos (2010) 51

CMU SCS

Dim. reduction - w/ fractals

- Q: what is the smallest # of attributes we should keep?

15-826 Copyright: C. Faloutsos (2010) 52

CMU SCS

Dim. reduction - w/ fractals

- Q: what is the smallest # of attributes we should keep?
- A: we should keep at least as many as the f.d. (and probably, a few more)


15-826 Copyright: C. Faloutsos (2010) 53

CMU SCS

Dim. reduction - w/ fractals

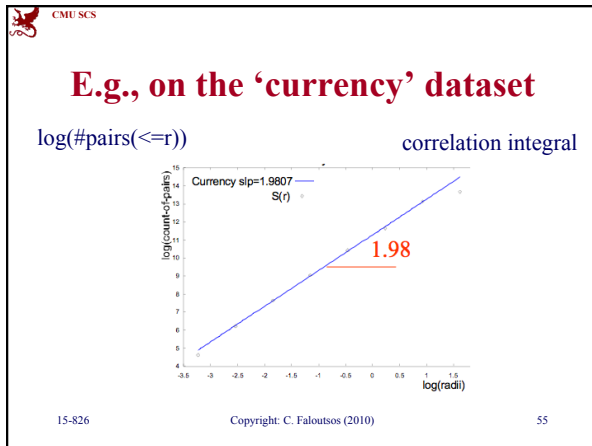
- Results: E.g., on the 'currency' dataset
- (daily exchange rates for USD, HKD, BP, FRF, DEM, JPY - i.e., 6-d vectors, one per day - base currency: CAD)

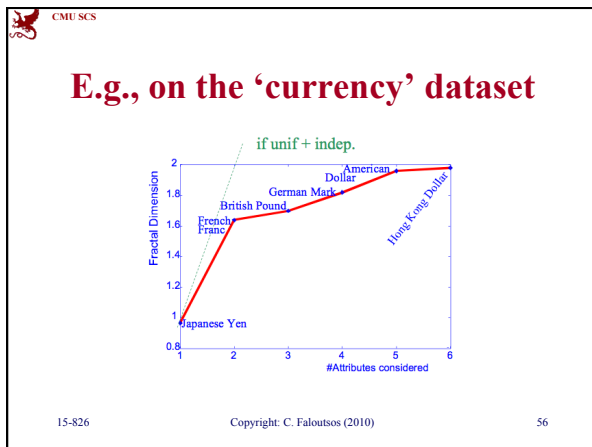
e.g.: FRF

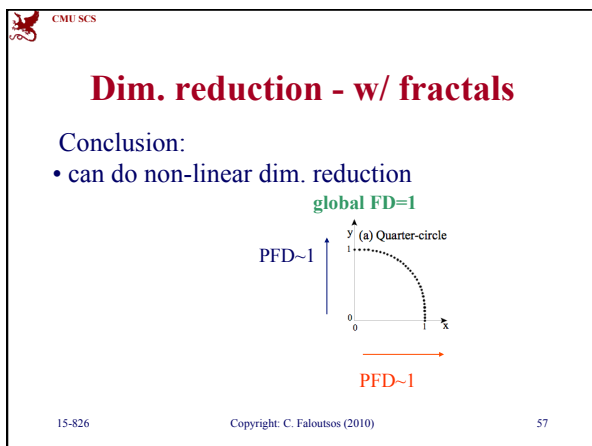



USD

15-826 Copyright: C. Faloutsos (2010) 54







 CMU SCS

References

- [PODS94] Faloutsos, C. and I. Kamel (May 24-26, 1994). *Beyond Uniformity and Independence: Analysis of R-trees Using the Concept of Fractal Dimension*. Proc. ACM SIGACT-SIGMOD-SIGART PODS, Minneapolis, MN.
- [Traina+, SBBD'00] Traina, C., A. Traina, et al. (2000). *Fast feature selection using the fractal dimension*. XV Brazilian Symposium on Databases (SBBD), Paraiba, Brazil.

15-826 Copyright: C. Faloutsos (2010) 58
