CMU SCS

# 15-826: Multimedia Databases and Data Mining

Lecture #9: Fractals - case studies - I

*C. Faloutsos*

CMU SCS

## Must-read Material

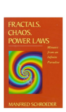
- Christos Faloutsos and Ibrahim Kamel, *Beyond Uniformity and Independence: Analysis of R-trees Using the Concept of Fractal Dimension*, Proc. ACM SIGACT-SIGMOD-SIGART PODS, May 1994, pp. 4-13, Minneapolis, MN.



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 (on reserve in the WeH library)



CMU SCS

## Reminder

- Code at

www.cs.cmu.edu/~christos/SRC/fdnq_h.zip

Also, in 'R'

> library(fdim);



CMU SCS

## Outline

Goal: 'Find similar / interesting things'

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



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
- ...

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) 7

CMU SCS

## (Fractals mentioned before:)

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

15-826 Copyright: C. Faloutsos (2013) 8

CMU SCS

## Case study#1: R-tree performance

Problem
- Given
  - N points in E-dim space

- Estimate # disk accesses for a range query
  ($q1$ x ... x $q_E$ )

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

15-826 Copyright: C. Faloutsos (2013) 9

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
  ($q1 \times ... \times q_E$ )

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

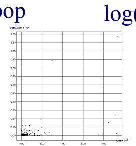
  

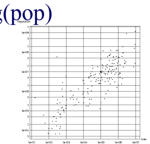
CMU SCS

## Examples:World's countries

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

pop log(pop)





area log(area)

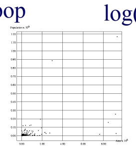
  

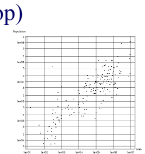
CMU SCS

## Examples:World's countries

- neither uniform, nor independent!

pop log(pop)



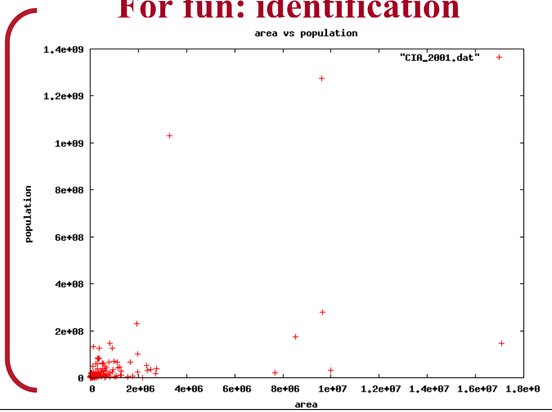


area log(area)

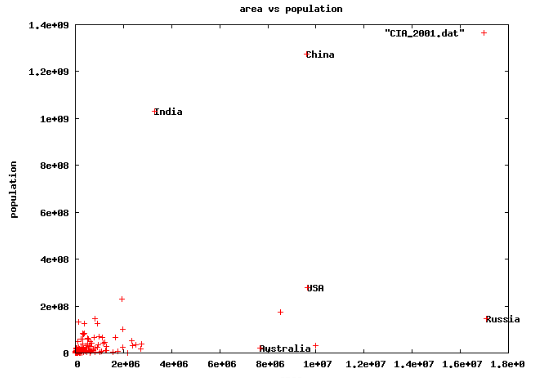
CMU SCS

## For fun: identification



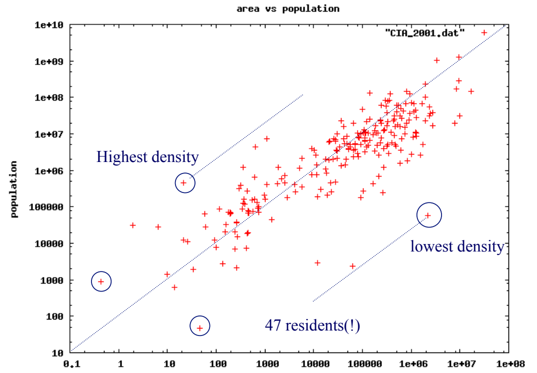

CMU SCS

## For fun: identification



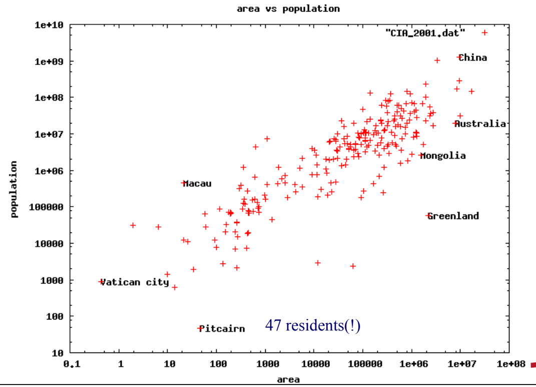

CMU SCS

## For fun: identification



CMU SCS

## For fun: identification

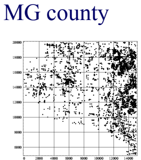



CMU SCS

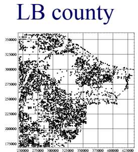
## Examples: TIGER files

- neither uniform, nor independent!

MG county



LB county



CMU SCS

## How to proceed?

- recall the [Pagel+] 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!

## How to proceed?

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

Answer (jumping ahead):

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



## How to proceed?

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

Answer (jumping ahead):

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

**Hausdorff fd**

side of (parent) MBR

# of data points

page capacity



## Let's see the rationale

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

CMU SCS

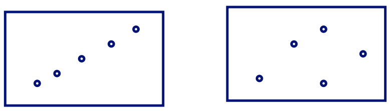
proof

## R-trees - performance analysis

I.e: for range queries - how many disk accesses, if we just now that we have

- *N* points in *E*-d space?

A: can not tell! need to know distribution



15-826 Copyright: C. Faloutsos (2013) 22

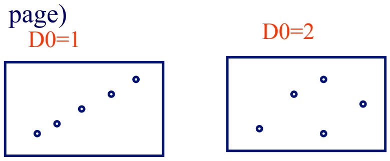
CMU SCS

proof

## R-trees - performance analysis

Q: OK - so we are told that the **Hausdorff** fractal dim. = D0 - Next step?

(also know that there are at most *C* points per page)




15-826 Copyright: C. Faloutsos (2013) 23

CMU SCS

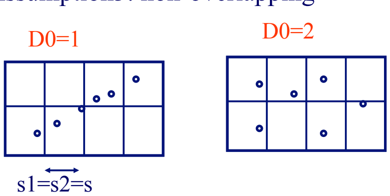
proof

## R-trees - performance analysis

Assumption1: square-like parents (s*s)

Assumption2: fully packed (C points each)

Assumption3: non-overlapping



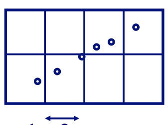

15-826 Copyright: C. Faloutsos (2013) 24

proof

## R-trees - performance analysis

Assumption1: square-like parents (s*s)
Assumption2: fully packed (N/C non-empty)
Assumption3: non-overlapping

D0=1



s1=s2=s

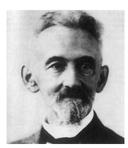




proof

## R-trees - performance analysis

Hint: dfn of Hausdorff f.d.:



Felix Hausdorff (1868-1942)





## Reminder: Hausdorff or box-counting fd:

- Box counting plot: Log( N ( r ) ) vs Log ( r)
- r: grid side
- N (r ): count of non-empty cells
- (Hausdorff) fractal dimension D0:

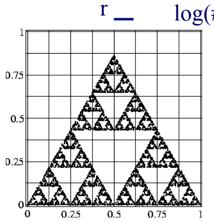
$$D_0 = -\frac{\partial \log(N(r))}{\partial \log(r)}$$

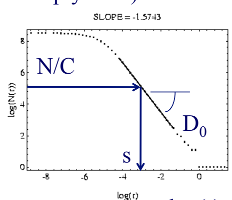
CMU SCS

proof

# Reminder

- Hausdorff fd:






CMU SCS

proof

# Reminder

- dfn of Hausdorff fd implies that

$$N(r) \sim r^{(-D0)}$$

\# non-empty cells of side *r*



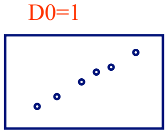
CMU SCS

proof

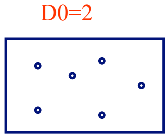
# R-trees - performance analysis

Q (rephrased): what is the side s1, s2, ... of parent nodes, given *N* data points, packed by *C,* with f.d. = *D0*

D0=1



D0=2

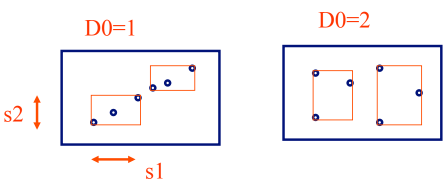


CMU SCS

proof

## R-trees - performance analysis

Q (rephrased): what is the side s1, s2, ... of parent nodes, given *N* data points, packed by *C,* with f.d. = *D0*



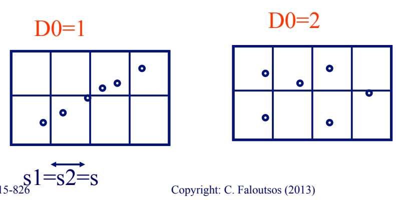

15-826 Copyright: C. Faloutsos (2013) 31

CMU SCS

proof

## R-trees - performance analysis

Q (rephrased): what is the side s1, s2, ... of parent nodes, given *N* data points, packed by *C,* with f.d. = *D0*




15-826 Copyright: C. Faloutsos (2013) 32

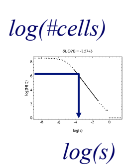
CMU SCS

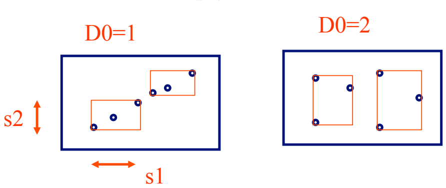
proof

## R-trees - performance analysis

A: (educated guess)

- s=s1=s2 (= ... ) - square-like MBRs
- N/C non-empty cells = K * $s^{(-D0)}$




15-826 Copyright: C. Faloutsos (2013) 33

CMU SCS

proof

## R-trees - performance analysis

Details of derivations: in [PODS 94].

Finally, expected side *s* of parent MBRs:

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

Q: sanity check: how does *s* change with *D0?*

A:

15-826 Copyright: C. Faloutsos (2013) 34

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 (2013) 35

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 (2013) 36

CMU SCS

## R-trees - performance analysis

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

A: # of parent-node accesses:

$N/C * (s + q1) * (s + q2) * \ldots (s + q_E)$

A: # of grand-parent node accesses

15-826 Copyright: C. Faloutsos (2013) 37

CMU SCS

## R-trees - performance analysis

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

A: # of parent-node accesses:

$N/C * (s + q1) * (s + q2) * \ldots (s + q_E)$

A: # of grand-parent node accesses

$N/(C^2) * (s' + q1) * (s' + q2) * \ldots (s' + q_E)$

$s' = ??$

15-826 Copyright: C. Faloutsos (2013) 38

CMU SCS

## R-trees - performance analysis

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

A: # of parent-node accesses:

$N/C * (s + q1) * (s + q2) * \ldots (s + q_E)$

A: # of grand-parent node accesses

$N/(C^2) * (s' + q1) * (s' + q2) * \ldots (s' + q_E)$

$s' = (C^2/N)^{1/D0}$

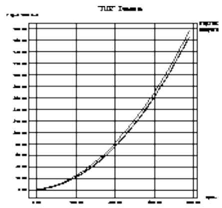
15-826 Copyright: C. Faloutsos (2013) 39

CMU SCS

## R-trees - performance analysis

Results: IUE (x-y star coordinates)

# leaf accesses



(a) IUE - Leaf accesses vs. query s

query side

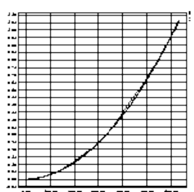
  

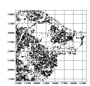
CMU SCS

## R-trees - performance analysis

Results: LB County

# leaf accesses





query side

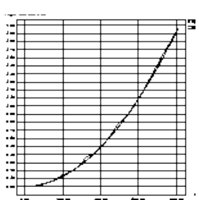
  

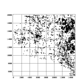
CMU SCS

## R-trees - performance analysis

Results: MG-county

# leaf accesses





query side

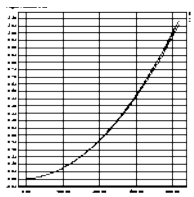
  

CMU SCS

## R-trees - performance analysis

Results: 2D- uniform

# leaf accesses



query side



CMU SCS

## R-trees - performance analysis

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



CMU SCS

Optional

## 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

Optional

## Case study #2: Dim. reduction

Problem definition: 'Feature selection'
- given *N* points, with *E* dimensions
- keep the *k* most 'informative' dimensions

[Traina+,SBBD'00]

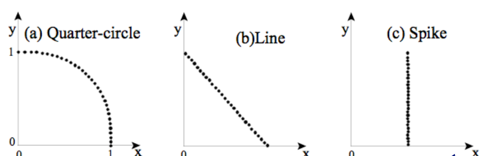
Caetano Traina Agma Traina Leejay Wu

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

CMU SCS

Optional

## Dim. reduction - w/ fractals




not informative

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

CMU SCS

Optional

## 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 (2013) 48

CMU SCS

Optional

## 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)



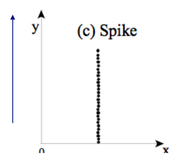
CMU SCS

Optional

## Dim. reduction - w/ fractals

**global FD=1**

PFD=1




PFD~0



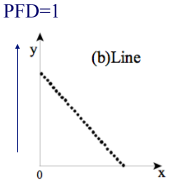
CMU SCS

Optional

## Dim. reduction - w/ fractals

**global FD=1**

PFD=1



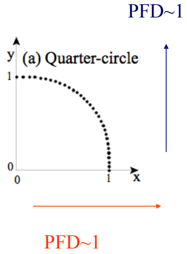

PFD=1

CMU SCS

Optional

## Dim. reduction - w/ fractals

global FD=1




15-826 Copyright: C. Faloutsos (2013) 52

CMU SCS

Optional

## Dim. reduction - w/ fractals

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

15-826 Copyright: C. Faloutsos (2013) 53

CMU SCS

Optional

## 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 (2013) 54

CMU SCS

Optional

## 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 (2013) 55

CMU SCS

Optional

## Dim. reduction - w/ fractals

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

15-826 Copyright: C. Faloutsos (2013) 56

CMU SCS

Optional

## 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 (2013) 57

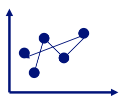
CMU SCS

Optional

## 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.:



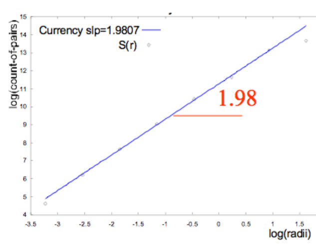

15-826 Copyright: C. Faloutsos (2013) 58

CMU SCS

Optional

## E.g., on the 'currency' dataset

log(#pairs(<=r)) correlation integral

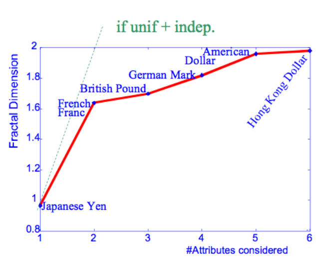



15-826 Copyright: C. Faloutsos (2013) 59

CMU SCS

Optional

## E.g., on the 'currency' dataset




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

Optional

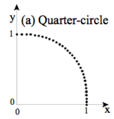
## Dim. reduction - w/ fractals

Conclusion:

- can do non-linear dim. reduction

global FD=1

PFD~1




PFD~1





## 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.