CMU SCS

# 15-826: Multimedia Databases and Data Mining

Lecture #8: Fractals - introduction

*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

## Recommended Material

optional, but **very** useful:

- Manfred Schroeder *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise* W.H. Freeman and Company, 1991
  - Chapter 10: boxcounting method
  - Chapter 1: Sierpinski triangle



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

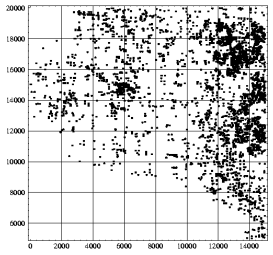
## Intro to fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots



CMU SCS

## Problem #1: GIS - points



Road end-points of Montgomery county:

•Q1: how many d.a. for an R-tree?

•Q2 : distribution?

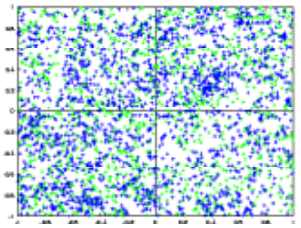
- •not uniform
- •not Gaussian
- •no rules??



CMU SCS

## Problem #2 - spatial d.m.

Galaxies (Sloan Digital Sky Survey w/ B. Nichol)



**- 'spiral' and 'elliptical' galaxies**

**(stores and households ...)**

**- patterns?**

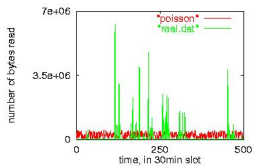
**- attraction/repulsion?**

**- how many 'spi' within r from an 'ell'?**

CMU SCS

# Problem #3: traffic

**# bytes**




**time**

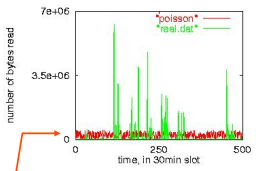
- disk trace (from HP - J. Wilkes); Web traffic - fit a model
- how many explosions to expect?
- queue length distr.?



CMU SCS

# Problem #3: traffic

**# bytes**




**time**

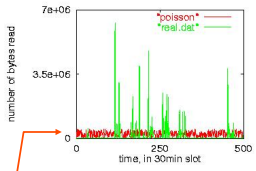
**Poisson indep., ident. distr**



CMU SCS

# Problem #3: traffic

**# bytes**




**time**

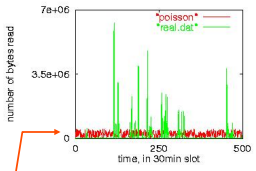
~~**Poisson**~~ **indep., ident. distr**



CMU SCS

# Problem #3: traffic

**# bytes**




**time**

~~**Poisson indep., ident. distr**~~

Q: Then, how to generate such bursty traffic?

CMU SCS

## Common answer:

- Fractals / self-similarities / power laws
- Seminal works from Hilbert, Minkowski, Cantor, Mandelbrot, (Hausdorff, Lyapunov, Ken Wilson, …)



CMU SCS

## Road map

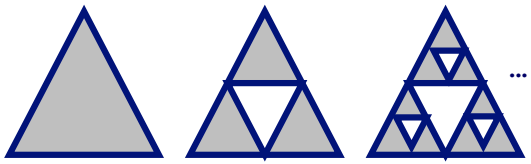
- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots



CMU SCS

## What is a fractal?

= self-similar point set, e.g., Sierpinski triangle:



... → **zero area; infinite length!**

Dimensionality??



CMU SCS

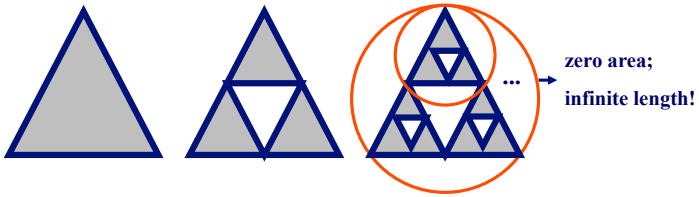
## Definitions (cont'd)

- Paradox: Infinite perimeter ; Zero area!
- 'dimensionality': between 1 and 2
- actually: Log(3)/Log(2) = 1.58...

CMU SCS

# Dfn of fd:

**ONLY** for a perfectly self-similar point set:



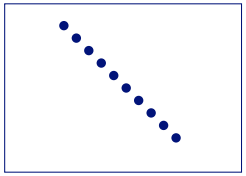

**=log(n)/log(f) = log(3)/log(2) = 1.58**



CMU SCS

# Intrinsic ('fractal') dimension

- Q: fractal dimension of a line?
- A: 1 (= log(2)/log(2)!)

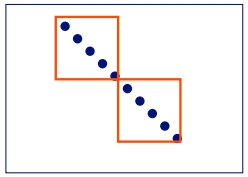




CMU SCS

# Intrinsic ('fractal') dimension

- Q: fractal dimension of a line?
- A: 1 (= log(2)/log(2)!)

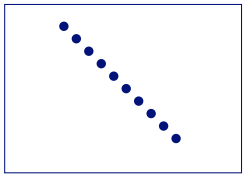




CMU SCS

# Intrinsic ('fractal') dimension

- Q: dfn for a given set of points?



| x | y |
|---|---|
| 5 | 1 |
| 4 | 2 |
| 3 | 3 |
| 2 | 4 |

CMU SCS

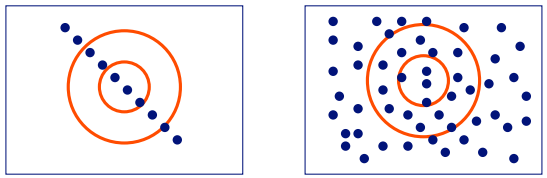
# Intrinsic ('fractal') dimension

- Q: fractal dimension of a line?
- A: nn ( <= r ) ~ r^1

( 'power law' : y=x^a)

- Q: fd of a plane?
- A: nn ( <= r ) ~ r^2

fd== slope of (log(nn) vs log(r) )





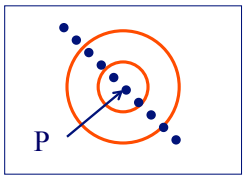


CMU SCS

EXPLANATIONS

# Intrinsic ('fractal') dimension

- **Local** fractal dimension of point 'P'?
- A: $nn_P$ ( <= r ) ~ r^1




- If this equation holds for several values of r,
- Then, the **local fractal dimension** of point P:
- Local fd = exp = 1

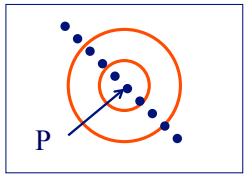


CMU SCS

EXPLANATIONS

# Intrinsic ('fractal') dimension

- **Local** fractal dimension of point 'A'?
- A: $nn_P$ ( <= r ) ~ r^1




- If this is true for all points of the cloud
- Then the exponent is the **global** f.d.
- Or simply the f.d.

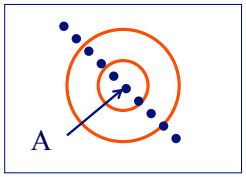


CMU SCS

EXPLANATIONS

# Intrinsic ('fractal') dimension

- **Global** fractal dimension?
- A: if
- $sum_{all\_P}$ [ $nn_P$ ( <= r ) ] ~ r^1
- Then: exp = global f.d.




- If this is true for all points of the cloud
- Then the exponent is the **global** f.d.
- Or simply the f.d.

CMU SCS

EXPLANATIONS

## Intrinsic ('fractal') dimension

- Algorithm, to estimate it?

Notice

- $Sum_{all\_P}$ [ $nn_P$ (<=r) ] is exactly *tot#pairs(<=r)*

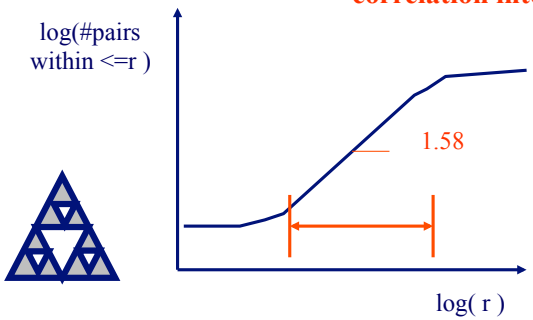
including 'mirror' pairs



CMU SCS

## Sierpinsky triangle

== 'correlation integral'






CMU SCS

## Observations:

- Euclidean objects have **integer** fractal dimensions
  - point: 0
  - lines and smooth curves: 1
  - smooth surfaces: 2
- fractal dimension -> roughness of the periphery



CMU SCS

## Important properties

- fd = embedding dimension -> uniform pointset
- a point set may have several fd, depending on scale

CMU SCS

## Important properties

- fd = embedding dimension -> uniform pointset
- a point set may have several fd, depending on scale

2-d



CMU SCS

## Important properties

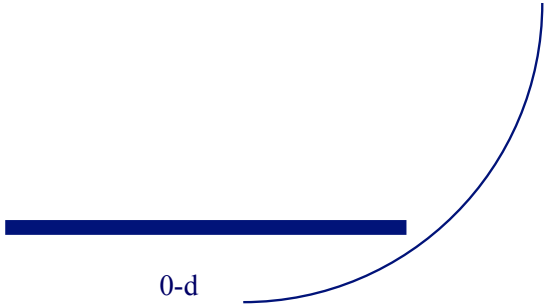
- fd = embedding dimension -> uniform pointset
- a point set may have several fd, depending on scale

1-d



CMU SCS

## Important properties



0-d



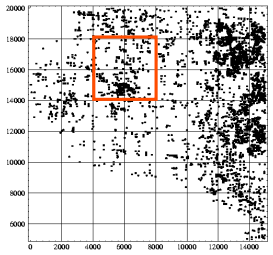
CMU SCS

## Road map

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots

CMU SCS

# Problem #1: GIS points



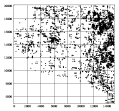
Cross-roads of Montgomery county:

•any rules?

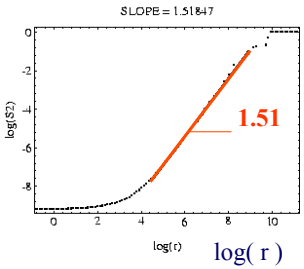


CMU SCS

# Solution #1



log(#pairs(within <= r))




log( r )

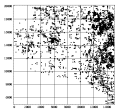
A: self-similarity ->

- <=> fractals
- <=> scale-free
- <=> power-laws (y=x^a, F=C*r^(-2))
- avg#neighbors(<= r ) = r^D

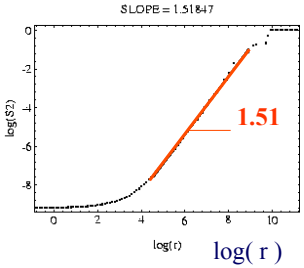


CMU SCS

# Solution #1



log(#pairs(within <= r))




log( r )

A: self-similarity

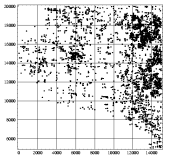
- avg#neighbors(<= r ) ~ r^(1.51)

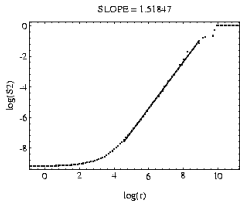


CMU SCS

# Examples:MG county

- Montgomery County of MD (road end-points)

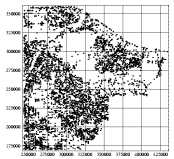


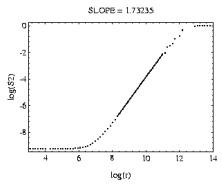


CMU SCS

# Examples:LB county

- Long Beach county of CA (road end-points)





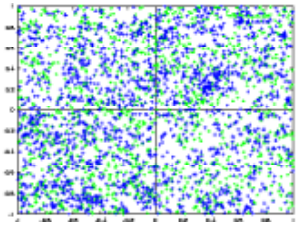


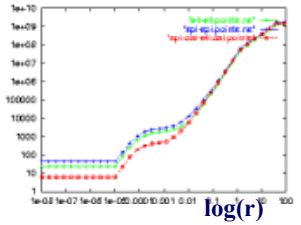

CMU SCS

# Solution#2: spatial d.m.

Galaxies ('BOPS' plot - [sigmod2000])

log(#pairs)





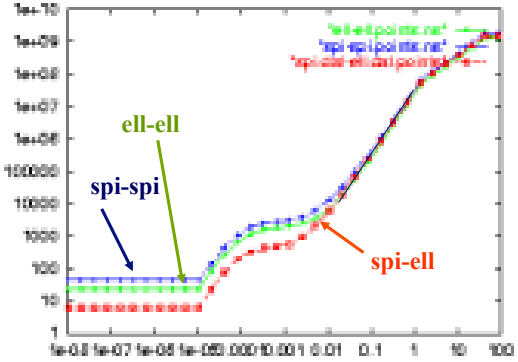
log(r)



CMU SCS

# Solution#2: spatial d.m.

log(#pairs within <=r )




- 1.8 slope
- plateau!
- repulsion!

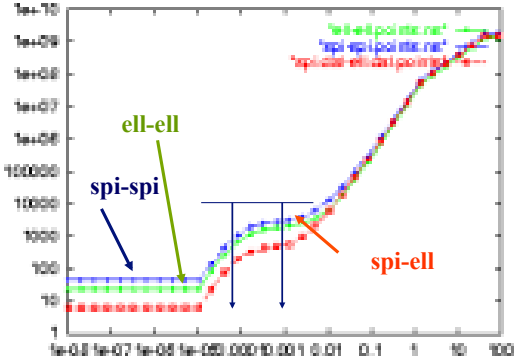
log(r)



CMU SCS

# Spatial d.m.

log(#pairs within <=r )

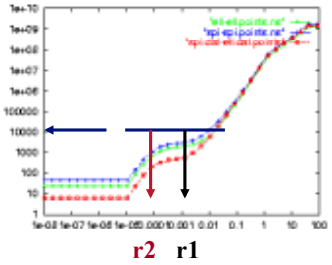



- 1.8 slope
- plateau!
- repulsion!

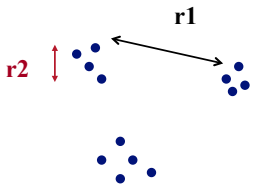
log(r)

CMU SCS

## Spatial d.m.



r2 r1



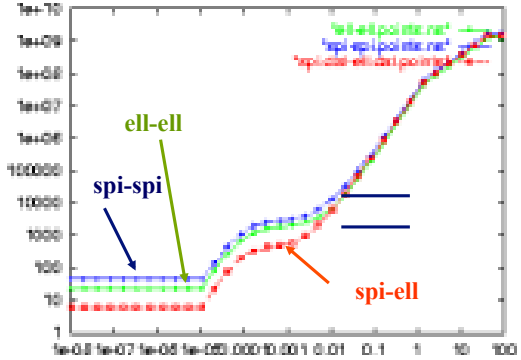

Heuristic on choosing # of clusters



CMU SCS

## Spatial d.m.

log(#pairs within <=r )




- 1.8 slope
- plateau!
- repulsion!

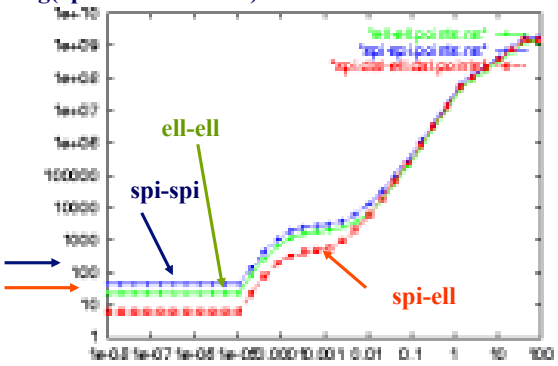
log(r)



CMU SCS

## Spatial d.m.

log(#pairs within <=r )




- 1.8 slope
- plateau!
-repulsion!!
-duplicates

log(r)

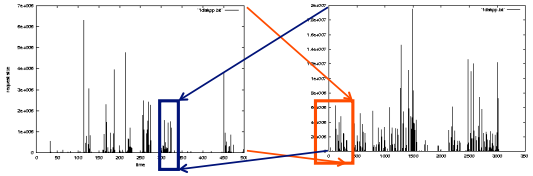


CMU SCS

## Solution #3: traffic

- disk traces: self-similar:

#bytes

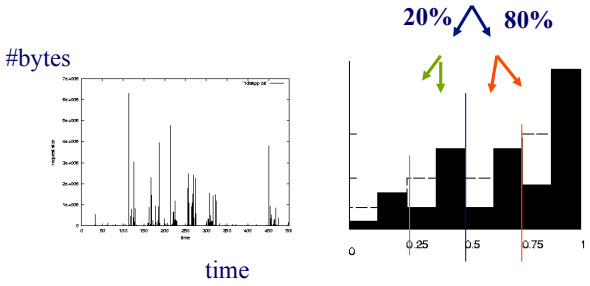


time

CMU SCS

## Solution #3: traffic

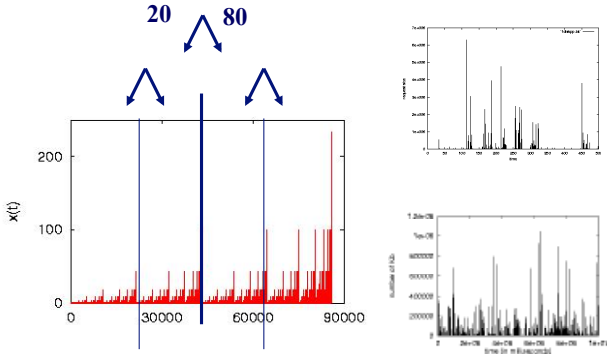
- disk traces (80-20 'law' = 'multifractal')






CMU SCS

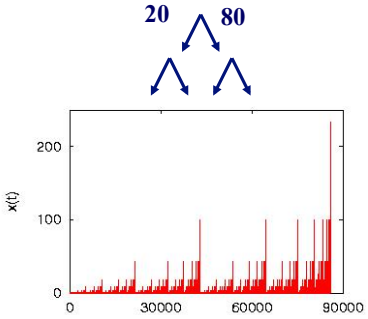
## 80-20 / multifractals






CMU SCS

## 80-20 / multifractals



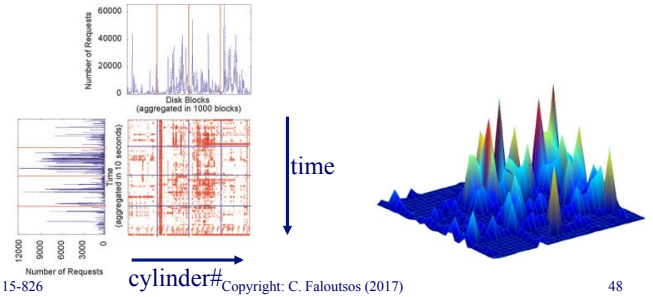

- p ; (1-p) in general
- **yes, there are dependencies**



CMU SCS

## More on 80/20: PQRS

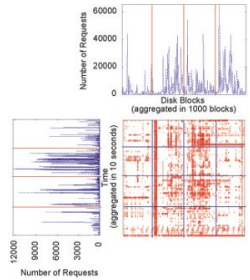
- Part of 'self-* storage' project [Wang+' 02]

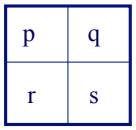


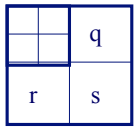
CMU SCS

## More on 80/20: PQRS

- Part of 'self-* storage' project [Wang+' 02]






CMU SCS

## Solution#3: traffic

Clarification:

- fractal: a set of points that is self-similar
- multifractal: a probability density function that is self-similar

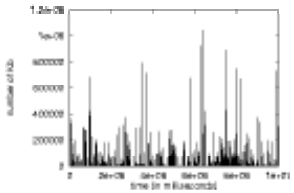
Many other time-sequences are bursty/ clustered: (such as?)



CMU SCS

## Example:

- network traffic



http://repository.cs.vt.edu/lbl-conn-7.tar.Z

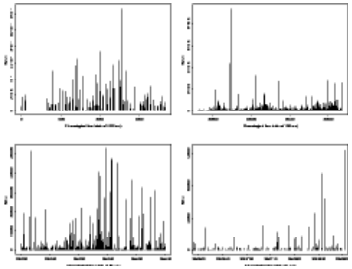


CMU SCS

## Web traffic

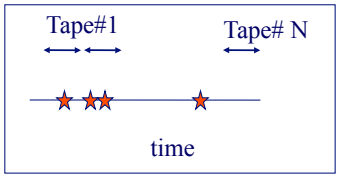
- [Crovella Bestavros, SIGMETRICS' 96]

1000 sec; 100sec
10sec; 1sec



CMU SCS

## Tape accesses




# tapes needed, to retrieve *n* records?

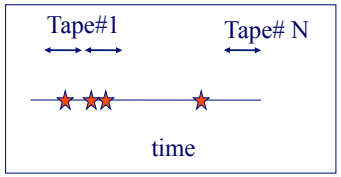
(# days down, due to failures / hurricanes / communication noise...)

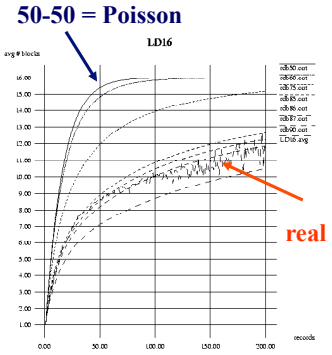


CMU SCS

## Tape accesses

# tapes retrieved




# qual. records



CMU SCS

## Road map

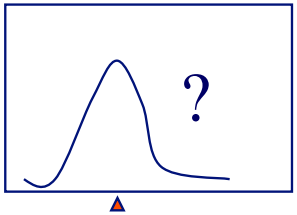
- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More **tools** and examples
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots



CMU SCS

## A counter-intuitive example

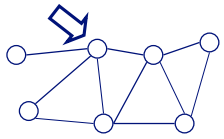
count




avg: 3.3

degree

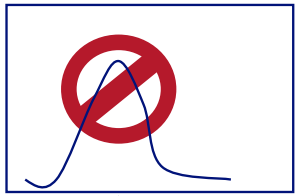
- avg degree is, say 3.3
- pick a node at random – guess its degree, exactly (-> "mode")



CMU SCS

## A counter-intuitive example

count



avg: 3.3

degree

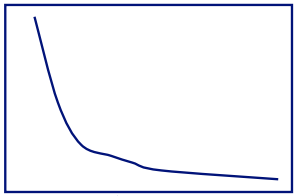
- avg degree is, say 3.3
- pick a node at random – guess its degree, exactly (-> "mode")
- A: 1!!



CMU SCS

## A counter-intuitive example

count



avg: 3.3

degree

- avg degree is, say 3.3
- pick a node at random - what is the degree you expect it to have?
- A: 1!!
- A': very skewed distr.
- Corollary: **the mean is meaningless**!
- (and std -> infinity (!))



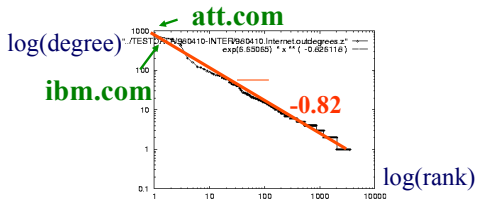
CMU SCS

## Rank exponent *R*

- Power law in the degree distribution [SIGCOMM99]

**internet domains**

log(degree)




log(rank)



CMU SCS

## More tools

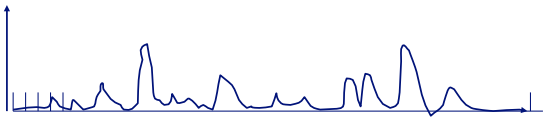
- Zipf's law
- Korcak's law / "fat fractals"

CMU SCS

## A famous power law: Zipf's law

• Q: vocabulary word frequency in a document - any pattern?

freq.



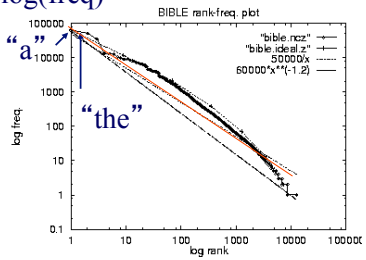
aaron zoo



CMU SCS

## A famous power law: Zipf's law

log(freq)




• Bible - **rank** vs **frequency** (log-log)

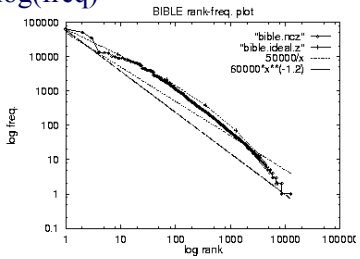
log(rank)



CMU SCS

## A famous power law: Zipf's law

log(freq)




log(rank)

• Bible - rank vs frequency (log-log)

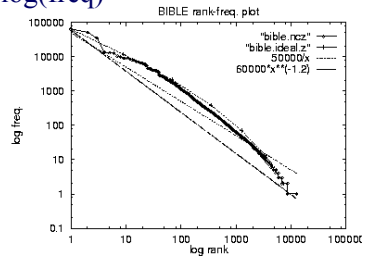
• similarly, in **many other** languages; for customers and sales volume; city populations etc etc



CMU SCS

## A famous power law: Zipf's law

log(freq)




log(rank)

•Zipf distr:

freq = 1/ rank

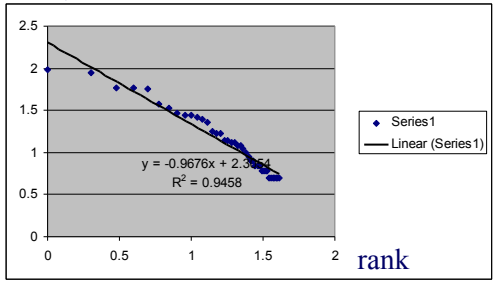
•generalized Zipf:

freq = 1 / (rank)^a

CMU SCS

## Olympic medals (Sydney):

log(#medals)



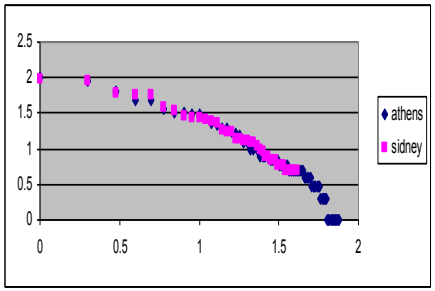

rank



CMU SCS

## Olympic medals (Sydney' 00, Athens' 04):

log(#medals)



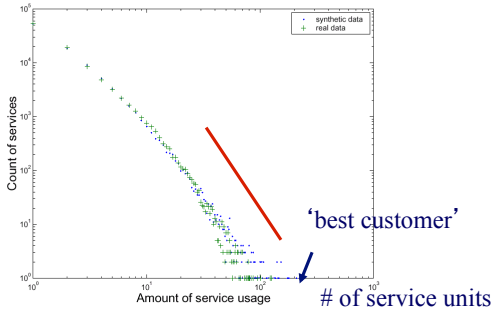

log( rank)



CMU SCS

## TELCO data

count of customers




'best customer'

# of service units

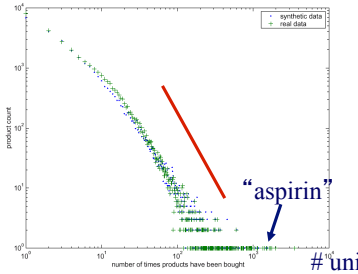
Count-frequency plot of real and synthetic data



CMU SCS

## SALES data – store#96

count of products




"aspirin"

# units sold

Count-frequency plot for store no. 96.

CMU SCS

# More power laws: areas – Korcak's law

Scandinavian lakes

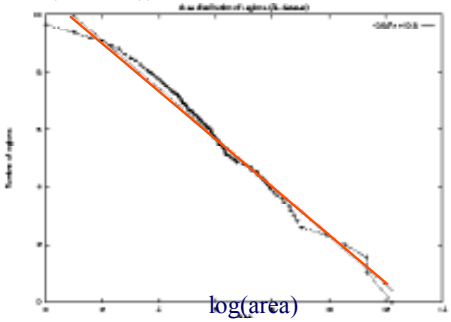
Any pattern?



CMU SCS

# More power laws: areas – Korcak's law

log(count( >= area))



Scandinavian lakes area vs complementary cumulative count (log-log axes)

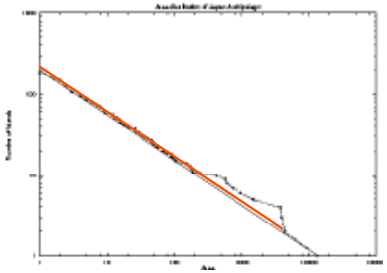
log(area)



CMU SCS

# More power laws: Korcak

log(count( >= area))



Japan islands;

area vs cumulative count (log-log axes)

log(area)



CMU SCS

# (Korcak's law: Aegean islands)


CMU SCS

# Korcak's law & "fat fractals"

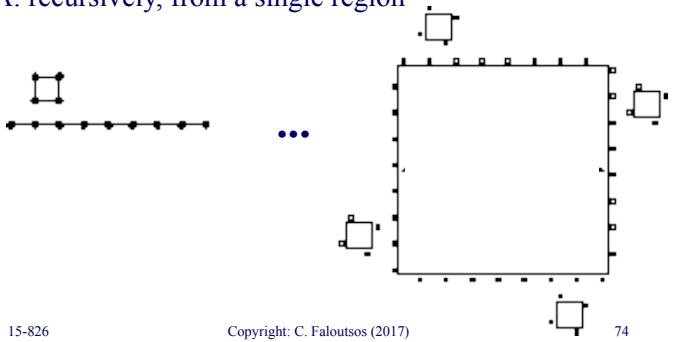

How to generate such regions?



CMU SCS

# Korcak's law & "fat fractals"

Q: How to generate such regions?
A: recursively, from a single region





CMU SCS

# so far we've seen:

- concepts:
  - fractals, multifractals and fat fractals
- tools:
  - correlation integral (= pair-count plot)
  - rank/frequency plot (Zipf's law)
  - CCDF (Korcak's law)

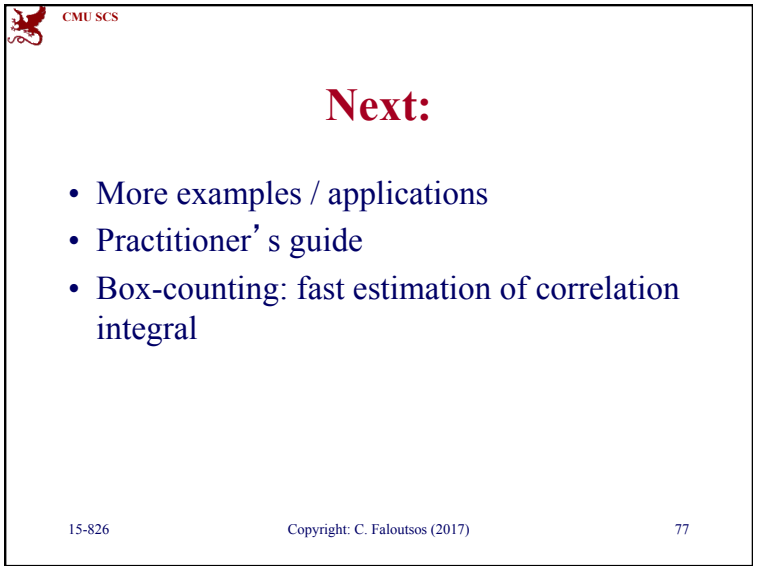


CMU SCS

# so far we've seen:

- concepts:
  - fractals, multifractals and fat fractals
- tools:
  - correlation integral (= pair-count plot)
  - rank/frequency plot (Zipf's law)
  - CCDF (Korcak's law)

same info


CMU SCS
Next:
• More examples / applications
• Practitioner's guide
• Box-counting: fast estimation of correlation integral
15-826
Copyright: C. Faloutsos (2017)
77