# Internet Research meets Data Mining

Part B: HOW TO FIND MORE

*C. Faloutsos*

# High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws



# Table Overview

| | Know | Don't Know | How to learn more |
|---|---|---|---|
| Topology | **Powerlaws, jellyfish** | **Growth pattern, Compare graphs** | |
| Link | **LRD, ON/OFF sources** | **Effect of topology and protocols** | |
| End-2-end | **LRD loss and RTT** | **Troubleshoot, cluster and predict** | |
| Traffic Matrix | **Skewness of location** | **Comprehensive model, troubleshoot** | |



# Table Overview

| | Know | Don't Know | How to learn more |
|---|---|---|---|
| Topology | **Powerlaws, jellyfish** | **Growth pattern, Compare graphs** | **SVD, fractals** |
| Link | **LRD, ON/OFF sources** | **Effect of topology and protocols** | **ARIMA, wavelets** |
| End-2-end | **LRD loss and RTT** | **Troubleshoot, cluster and predict** | **ARIMA, wavelets** |
| Traffic Matrix | **Skewness of location** | **Comprehensive model, troubleshoot** | **Power-laws; multifractals, clustering** |



# B.I - Traditional D.M. - Outline

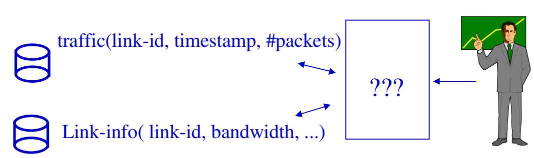
- Motivating Problems
- Supervised learning: decision trees
- Unsupervised learning: clustering
- Unsupervised learning: association rules
- Conclusions - practitioner's guide



# Problem

Given: (multiple) data sources

Find: patterns (classifiers, rules, clusters, outliers...)



# Problem 1: classification

- Eg. Given profiles of 'good' and 'bad' customers (clients, links, …)
- Classify the current customer (client, link, …)



# Problem 2: clustering

- Eg. Given profiles of several customers (clients, links, …)
- group them into 'natural' groups



# Problem 3: Association Rules

- Given a sequence of events (eg., 'server-A comes up', 'server-B goes down', …)
- Find events that occur together too often, eg.,
  - server-A-up, server-B-down -> server-C-down



# B.I - Traditional D.M. - Outline

- Motivating Problems
- Supervised learning: decision trees
- Unsupervised learning: clustering
- Unsupervised learning: association rules
- Conclusions - practitioner's guide



# Decision trees - Problem

| Avg packet size | Avg arrival rate | time | … | CLASS-ID |
|---|---|---|---|---|
| 30 | 150 | 13:30 | | + |
| | | | | … |
| | | | | - |

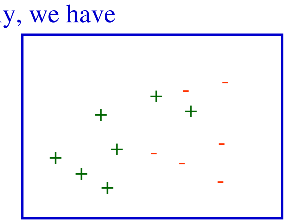
| | | | | |
|---|---|---|---|---|
| | | | | ?? |



# Decision trees

- Pictorially, we have

num. attr#2
(eg., avg rate)

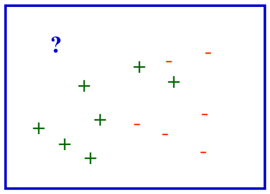


num. attr#1 (eg., 'avg size')

## Decision trees

- and we want to label '**?**'

num. attr#2
(eg., avg rate)



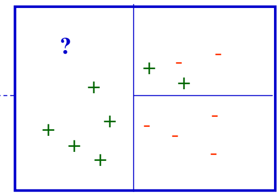

num. attr#1 (eg., 'avg size')



## Decision trees

- so we build a decision tree:

num. attr#2
(eg., avg rate)
40

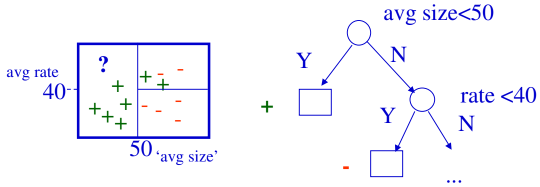



50
num. attr#1 (eg., 'avg size')



## Decision trees

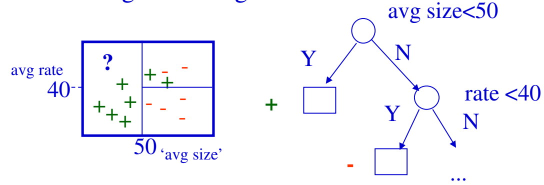
- so we build a decision tree:






## Decision trees

- Goal: split address space in (almost) homogeneous regions






## Conclusions -Practitioner's guide:

- **Many** available implementations
  - eg, C4.5 (freeware), C5.0
  - Also, inside larger stat. packages
- They usually hide **all** the details from us:
  - training / testing / tree pruning
  - 'boosting'
  - recent, scalable methods
  - see [Han+Kamber] for details



## High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - ➡ B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws

## B.I - Traditional D.M. - Outline

- Motivating Problems
- Supervised learning: decision trees
- Unsupervised learning: clustering
  - preliminaries
  - ‘sound’ methods
  - ‘iterative’ methods
- Unsupervised learning: association rules
- Conclusions - practitioner’s guide



## Problem 2: clustering

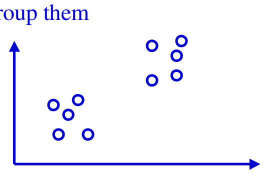
- Eg. Given profiles of several customers (clients, links, …)
- group them into ‘natural’ groups
- (and, optionally, report misfits as ‘outliers’)



## Cluster generation

- Problem:
  - given N points in D dimensions,
  - group them

avg packet rate

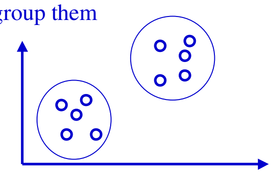


avg packet size



## Cluster generation

- Problem:
  - given N points in D dimensions,
  - group them





## Cluster generation

Short version:

- There are *numerous* clustering algorithms, available in free / open / commercial systems (eg., Splus, ‘R’ system)
- BUT: most algorithms require #-of-clusters and/or don’t scale up for large datasets
  - except for recent solutions...



## B.I - Traditional D.M. - Outline

- Motivating Problems
- Supervised learning: decision trees
- Unsupervised learning: clustering
  - preliminaries
  - ‘sound’ methods
  - ‘iterative’ methods
- Unsupervised learning: association rules
- Conclusions - practitioner’s guide

## Cluster generation

A: *many-many* algorithms - in two groups [VanRijsbergen]:

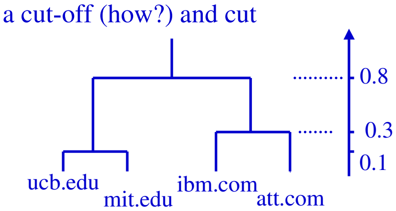
- theoretically sound (O($N$^2))
  - independent of the insertion order
- iterative (O($N$), O($N$ log($N$))





## Cluster generation - ‘sound’ methods

- Approach#1: dendrograms - create a hierarchy (bottom up or top-down) - choose a cut-off (how?) and cut

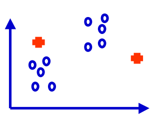







## Cluster generation - ‘sound’ methods

- Approach#2: min. some statistical criterion (eg., sum of squares from cluster centers)
  - like ‘k-means’

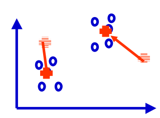






## Cluster generation - ‘sound’ methods

- Approach#2: min. some statistical criterion (eg., sum of squares from cluster centers)
  - like ‘k-means’
  - but how to decide ‘k’?

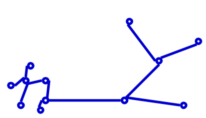






## Cluster generation - ‘sound’ methods

- Approach#3: Graph theoretic [Zahn]:
  - build MST;
  - delete edges longer than 2.5* std of the local average

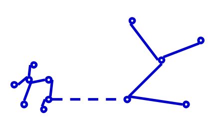






## Cluster generation - ‘sound’ methods

- Result:
  - why ‘2.5’?



## B.I - Traditional D.M. - Outline

- Motivating Problems
- Supervised learning: decision trees
- Unsupervised learning: clustering
  - preliminaries
  - ‘sound’ methods
  - ‘iterative’ methods
- Unsupervised learning: association rules
- Conclusions - practitioner’s guide



## Cluster generation - ‘iterative’ methods

general outline:

- Choose ‘seeds’ (how?)
- assign each vector to its closest seed (possibly adjusting cluster centroid)
- possibly, re-assign some vectors to improve clusters

Fast and practical, but ‘unpredictable’



## Cluster generation - ‘iterative’ methods

**Many**, recent, fast methods [see book by Han+Kamber]:

- BIRCH
- CURE
- CHAMELEON
- WaveCluster
- …



## Cluster generation- how many clusters?

- one way to estimate # of clusters *k*: X-means method [Moore+Pelleg]
- in general: AIC or BIC/MDL (= minimize not only error, but also model complexity, ie.: $\underline{RMSE + C * k}$ )
  - BIC: Bayesian Information Criterion
  - AIC: Akaike Inf. Criterion
  - MDL: minimum description language



## Conclusions - Practitioner’s guide

- **Many** clustering methods
- **Many** available implementations (BIRCH is free; all stat. packages include several versions of clustering algorithms)
- Usually need a ‘magic number’ (eg., # of clusters)



## High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws

## B.I - Traditional D.M. - Outline

- Motivating Problems
- Supervised learning: decision trees
- Unsupervised learning: clustering
- Unsupervised learning: association rules
- Conclusions - practitioner's guide



## Problem 3: Association rules

[Mannila+97]

- Given a stream of telecommunication events
- Find rules of the form

  *A,A,B -> C*

(within windows of 5')

Eg: A A C B

time

5'



## Association rules - idea

[Agrawal+SIGMOD93]

- Consider 'market basket' case:
  - (milk, bread)
  - (milk, bread, chocolate)
  - (milk, chocolate)
  - ...
  - (milk, bread)
- Find 'interesting things', eg., rules of the form:
  - milk, bread -> chocolate



## Association rules - example

INPUT:

(milk, bread)
(milk, bread, chocolate)
(milk, chocolate)
(milk, bread)

Sample rule:

milk, bread -> chocolate
(`confidence': 33%,
'support': 25%)

- '**confidence**' : how often people by chocolate, given that they have bought milk and bread
- '**support**': how often people buy bread, milk and chocolate



## Association rules - problem dfn

Problem definition:

- given
  - a set of 'market baskets' (=binary matrix, of N rows/baskets and M columns/products)
  - min-support 's' and
  - min-confidence 'c'
- find
  - all the rules with higher support and confidence



## Association rules

Association rules:

- Do NOT need the user to give 'hypotheses'
- because they discover automatically frequent items, pairs, triplets, ...
- They solve the problem, QUICKLY! (a few passes over the dataset)
  - 'A priori' algorithm of Agrawal+
  - faster algorithms (FP-trees - see [Han+Kamber])

# Association rules - Conclusions

Association rules: a new tool to find patterns

- easy to understand its output
- fine-tuned algorithms exist
- Many available implementations
  - IBM (IntelligentMiner)
    http://www-3.ibm.com/software/data/iminer/
  - Stand-alone ones



# Overall Conclusions

- Many, mature (and often, free!) tools for classification, clustering, and association rules



# Table Overview

| | Know | Don't Know | How to learn more |
|---|---|---|---|
| Topology | **Powerlaws, jellyfish** | **Growth pattern, Compare graphs** | |
| Link | **LRD, ON/OFF sources** | **Effect of topology and protocols** | **Association rules classification** |
| End-2-end | **LRD loss and RTT** | **Troubleshoot, cluster and predict** | **Association rules classification** |
| Traffic Matrix | **Skewness of location** | **Comprehensive model, troubleshoot** | **clustering** |



# Resources - software & urls

- Stat. Packages: SAS, Splus, 'R' (freeware!)
  - **www.r-project.org/**

  (all have SVD, ARIMA, clustering etc)
- Data Mining 'central': Software, datasets, conference announcements
  - **www.kdnuggets.com/**



# Resources - Books

- Machine Learning: Tom Mitchell: *Machine Learning*, McGraw Hill, 1997.
- Data mining: Jiawei Han and Micheline Kamber: *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.



# Additional Reading

- Agrawal, R., T. Imielinski, A. Swami, *Mining Association Rules between Sets of Items in Large Databases,* SIGMOD 1993.
- H. Mannila, H. Toivonen and I. Verkamo: Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1,3 (1997), 259-289.

## Additional Reading

- M. Mehta, R. Agrawal and J. Rissanen, `*SLIQ: A Fast Scalable Classifier for Data Mining'*, Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT), Avignon, France, March 1996
- Pelleg, Dan and Andrew Moore: *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*. In ICML-2000.



## Additional reading

- Van-Rijsbergen, C. J. (1979). Information Retrieval. London, England, Butterworths.
- Zahn, C. T. (Jan. 1971). "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters." IEEE Trans. on Computers C-20(1): 68-86.



# Part B.II: Time series , Fourier, wavelets and forecasting



## High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws



## B.II - Time Series Analysis - Outline

- Motivating problems
- DFT
- DWT
- AR(IMA) and forecasting

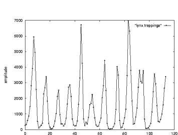


## Problem #1:

Goal: given a signal (eg., #packets over time)

Find: patterns, periodicities, and/or compress

count

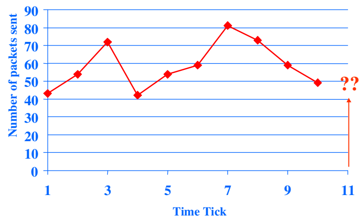


year

lynx caught per year
(packets per day;
virus infections per month)

## Problem#2: Forecast

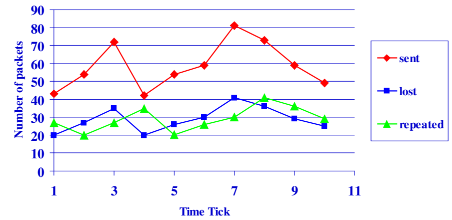
Given $x_t$, $x_{t-1}$, …, forecast $x_{t+1}$






## Problem #3:

- Given: A set of **correlated** time sequences
- Forecast 'Sent(t)'






## B.II - Time Series Analysis - Outline

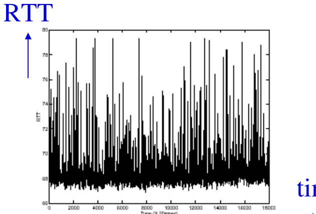
- DFT
  - Definition of DFT and properties
  - how to read the DFT spectrum
- DWT
- AR(IMA) and forecasting



## Recall from Part A:

UCR->CMU RTTs showed periodicity!

RTT



timestamp

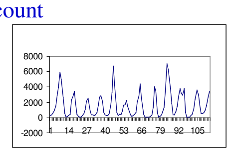


## Introduction - definitions

Goal: given a signal (eg., packets over time)

Find: patterns and/or compress

count




lynx caught per year
(packets per day;
virus infections per month)

year



## What does DFT do?

A: highlights the periodicities

## DFT: definition

- (**n-point**) Discrete Fourier Transform:

$$X_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \exp(-j2\pi tf/n) \qquad f = 0,\ldots,n-1$$

$$(j = \sqrt{-1})$$

inverse DFT

$$x_t = 1/\sqrt{n} \sum_{t=0}^{n-1} X_f * \exp(+j2\pi tf/n)$$



## DFT: definition

- **Good** news: Available in **all** symbolic math packages, eg., in 'mathematica'

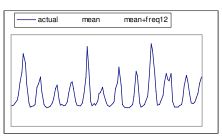
```
x = [1,2,1,2];
X = Fourier[x];
Plot[ Abs[X] ];
```



## DFT: Amplitude spectrum

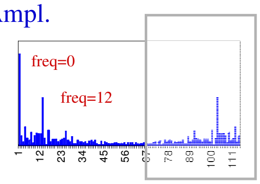
Amplitude: $A_f^{\,2} = \mathrm{Re}^2(X_f) + \mathrm{Im}^2(X_f)$

count




year

Ampl.



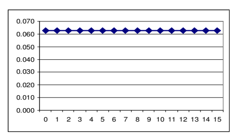

Freq.



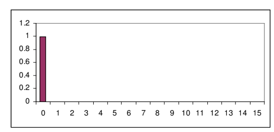
## DFT: examples

flat

Amplitude



time

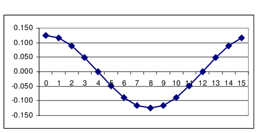


freq

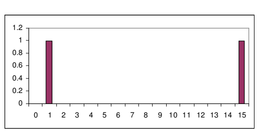


## DFT: examples

Low frequency sinusoid



time

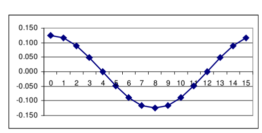


freq

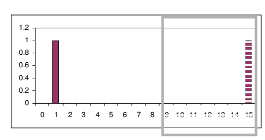


## DFT: examples

- Sinusoid - symmetry property: $X_f = X^*_{n-f}$



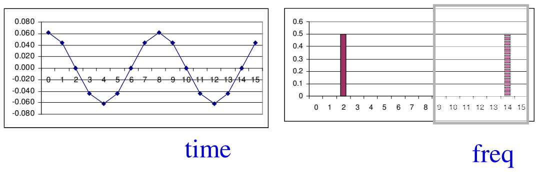
time



freq

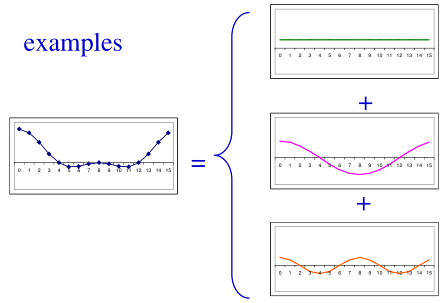
## DFT: examples

- Higher freq. sinusoid



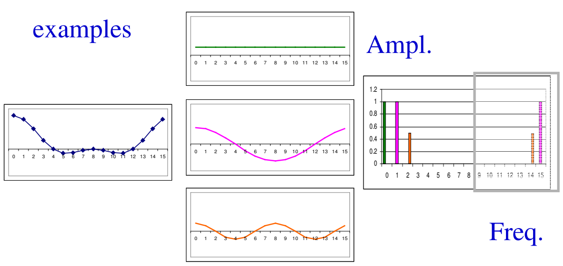



## DFT: examples






## DFT: examples





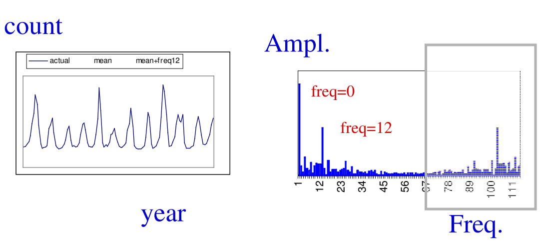

## B.II - Time Series Analysis - Outline

- DFT
  - Definition of DFT and properties
  - how to read the DFT spectrum
- DWT
- AR(IMA) and forecasting



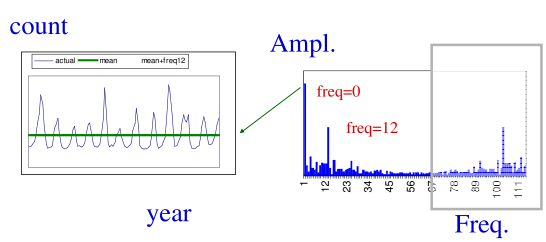
## DFT: Amplitude spectrum

Amplitude: $A_f^{\ 2} = \mathrm{Re}^2(X_f) + \mathrm{Im}^2(X_f)$

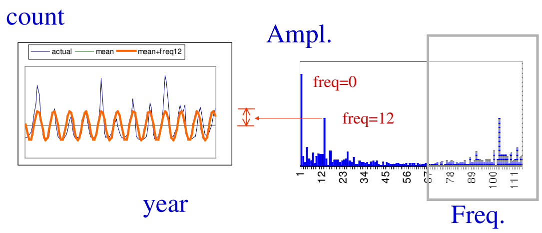





## DFT: Amplitude spectrum



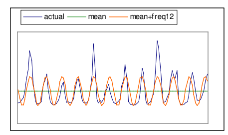
## DFT: Amplitude spectrum

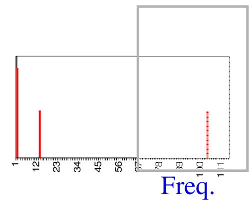





## DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?

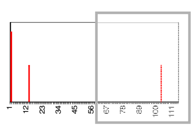





## DFT: Amplitude spectrum

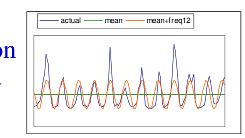
- excellent approximation, with only 2 frequencies!
- so what?
- A1: **(lossy) compression**
- A2: pattern discovery
- A3: forecasting





## DFT: Amplitude spectrum

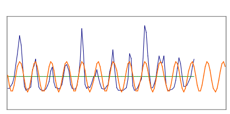
- excellent approximation, with only 2 frequencies!
- so what?
- A1: (lossy) compression
- A2: **pattern discovery**
- A3: forecasting






## DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
- A1: (lossy) compression
- A2: pattern discovery
- A3: **forecasting**
  (& outliers)





## DFT - Conclusions

- It spots periodicities (with the '**amplitude spectrum'**)
- can be quickly computed (O( *n* log *n*)), thanks to the FFT algorithm.
- **standard** tool in signal processing (speech, image etc signals)
- (closely related to DCT and JPEG)

# B.II - Time Series Analysis - Outline

- DFT
  - Definition of DFT and properties
  - how to read the DFT spectrum
- DWT
  - Motivation - definitions
  - How to read the 'scalogram'
- AR(IMA) and forecasting

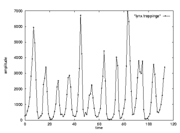


# Problem #1':

Goal: given a signal (eg., #packets over time)

Find: patterns, periodicities, and/or **compress**

count



lynx caught per year
(packets per day;
virus infections per month)

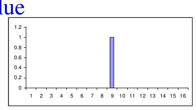
year



# Wavelets - DWT

- DFT is great - but, how about compressing a spike?

value



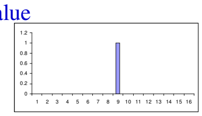
time



# Wavelets - DWT

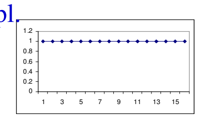
- DFT is great - but, how about compressing a spike?
- A: Terrible - all DFT coefficients needed!

value



time

Ampl.



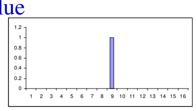
Freq.



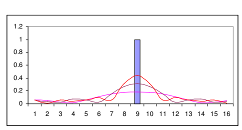
# Wavelets - DWT

- DFT is great - but, how about compressing a spike?
- A: Terrible - all DFT coefficients needed!

value



time

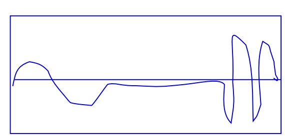




# Wavelets - DWT

- Similarly, DFT suffers on short-duration waves (eg., baritone, silence, soprano)

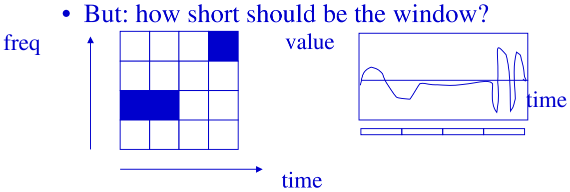
value



time

## Wavelets - DWT

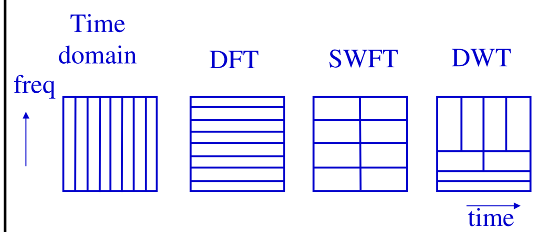
- Solution#1: Short window Fourier transform (SWFT)
- But: how short should be the window?






## Wavelets - DWT

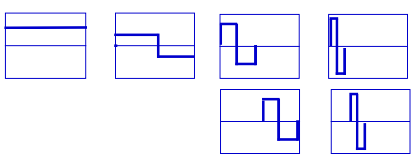
- Answer: **multiple** window sizes! -> DWT






## Haar Wavelets

- subtract sum of left half from right half
- repeat recursively for quarters, eight-ths, ...





## Haar wavelets - code

```
#!/usr/bin/perl5
# expects a file with numbers
# and prints the dwt transform
# The number of time-ticks should be a power of 2
# USAGE
#    haar.pl <fname>

my @vals=();
my @smooth; # the smooth component of the signal
my @diff;   # the high-freq. component

# collect the values into the array @val
while(<>){
        @vals = ( @vals ,  split );
}
```

```
my $len = scalar(@vals);
my $half = int($len/2);
while($half >= 1 ){
  for(my $i=0; $i< $half; $i++){
        $diff [$i] = ($vals[2*$i] - $vals[2*$i + 1] )/ sqrt(2);
        print "\t", $diff[$i];
        $smooth [$i] = ($vals[2*$i] + $vals[2*$i + 1] )/ sqrt(2);
  }
  print "\n";
  @vals = @smooth;
  $half = int($half/2);
}
print "\t", $vals[0], "\n" ;     # the final, smooth component
```



## Daubechies etc Wavelets

- Many more wavelets (Daubechies-4, -6 etc; Coifman; …)

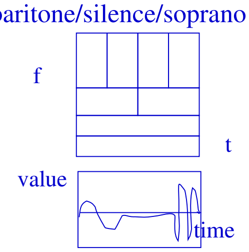


## B.II - Time Series Analysis - Outline

- DFT
  - Definition of DFT and properties
  - how to read the DFT spectrum
- DWT
  - Motivation - definitions
  - How to read the 'scalogram'
- AR(IMA) and forecasting

## Wavelets - Drill:

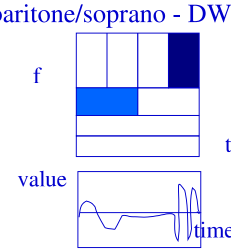
- Q: baritone/silence/soprano - DWT?






## Wavelets - Drill:

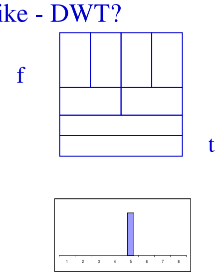
- Q: baritone/soprano - DWT?






## Wavelets - Drill:

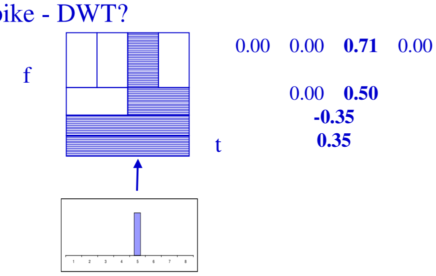
- Q: spike - DWT?






## Wavelets - Drill:

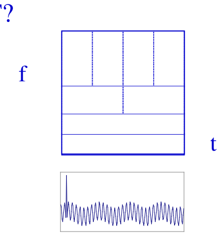
- Q: spike - DWT?






## Wavelets - Drill#2:

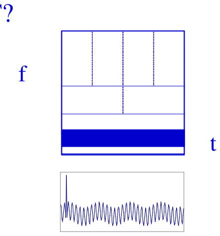
- Q: weekly + daily periodicity, + spike - DWT?





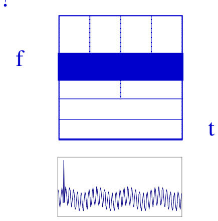

## Wavelets - Drill#2:

- Q: **weekly** + daily periodicity, + spike - DWT?



## Wavelets - Drill#2:

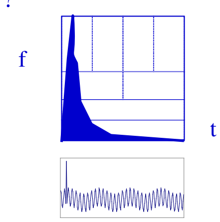
- Q: weekly + **daily** periodicity, + spike - DWT?






## Wavelets - Drill#2:

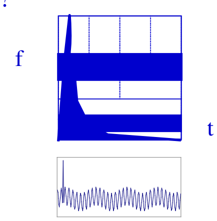
- Q: weekly + daily periodicity, + **spike** - DWT?






## Wavelets - Drill#2:

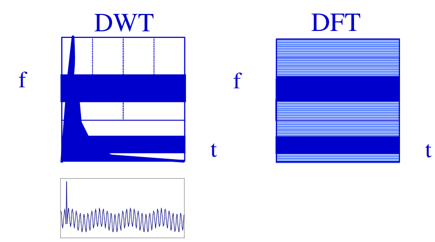
- Q: weekly + daily periodicity, + spike - DWT?






## Wavelets - Drill#2:

- Q: DFT?





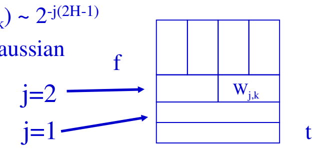

## Advantages of Wavelets

- Better compression (better RMSE with same number of coefficients - used in JPEG-2000)
- fast to compute (usually: O(*n*)!)
- very good for 'spikes'
- (mammalian eye and ear: Gabor wavelets)
- suitable for self-similar/LRD signals



## Advantages of Wavelets

- suitable for self-similar/LRD signals for fractional Gaussian Noise [Riedi+99]
  - var($W_{j,k}$) ~ $2^{-j(2H-1)}$
  - and ~ Gaussian



# Advantages of Wavelets

- suitable for self-similar/LRD signals for fractional Gaussian Noise [Riedi+99]
  - $var(W_{j,k}) \sim 2^{-j(2H-1)}$
  - and ~ Gaussian
- H: Hurst exponent (1/2 < H < 1)
- Fast generation of realistic LRD traffic



# Overall Conclusions

- DFT ( & DCT) spot periodicities
- DWT : multi-resolution - matches processing of mammalian ear/eye better; very suitable for self-similar traffic
- DWT: used for summarization of streams [Gilbert+01]



# Overal Conclusions - cont'ed

- All three: powerful tools for compression, pattern detection in real signals
- All three: included in math packages (matlab, mathematica, ... - DFT: even in spreadsheets!)



# B.II - Time Series Analysis - Outline

- Motivating problems
- DFT
- DWT
- AR(IMA) and forecasting



# Forecasting

"Prediction is very difficult, especially about the future." - Nils Bohr

**http://www.hfac.uh.edu/MediaFutures/thoughts.html**

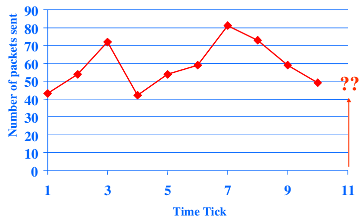


# ARIMA - Outline

- Auto-regression: Least Squares; recursive least squares
- Co-evolving time sequences
- Examples
- Conclusions

## Problem: Forecast

- Example: give $x_{t-1}$, $x_{t-2}$, …, forecast $x_t$






## Problem: Forecast

- Solution: try to express

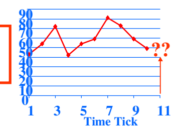
  $x_t$

  as a linear function of the past: $x_{t-2}$, $x_{t-2}$, …,

  (up to a window of $w$)

Formally:

$$\boxed{x_t \approx a_1 x_{t-1} + \ldots + a_w x_{t-w} + noise}$$






## (Problem: Back-cast; interpolate)

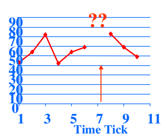
- Solution - interpolate: try to express

  $x_t$

  as a linear function of the past AND the future:

  $x_{t+1}$, $x_{t+2}$, … $x_{t+wfuture}$; $x_{t-1}$, … $x_{t-wpast}$

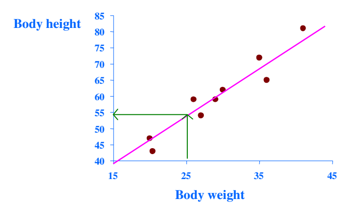
  (up to windows of $w_{past}$ , $w_{future}$)
- EXACTLY the same algo's






## Linear Regression: idea

| *patient* | *weight* | *height* |
|---|---|---|
| 1 | 27 | 43 |
| 2 | 43 | 54 |
| 3 | 54 | 72 |
| | … | |
| … | | … |
| N | 25 | ?? |




- express what we don't know (= 'dependent variable')
- as a linear function of what we know (= 'indep. variable(s)')



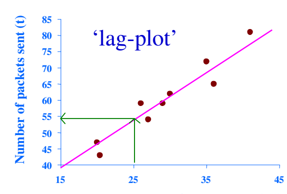
## Linear <u>Auto</u> Regression:

| *Time* | *Packets Sent(t)* |
|---|---|
| 1 | 43 |
| 2 | 54 |
| 3 | 72 |
| … | … |
| N | ?? |



## Linear <u>Auto</u> Regression:

| *Time* | *Packets Sent (t-1)* | *Packets Sent(t)* |
|---|---|---|
| 1 | - | 43 |
| 2 | 43 | 54 |
| 3 | 54 | 72 |
| | … | |
| … | | … |
| N | 25 | ?? |




- lag $w$=1
- <u>Dependent</u> variable = # of packets sent (S [t])
- <u>Independent</u> variable = # of packets sent (S[t-1])

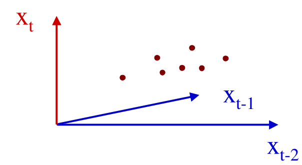
## B.II - Time Series Analysis - Outline

- Auto-regression
- Least Squares; recursive least squares
- Co-evolving time sequences
- Examples
- Conclusions



## More details:

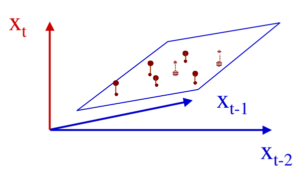
- Q1: Can it work with window *w*>1?
- A1: YES!






## More details:

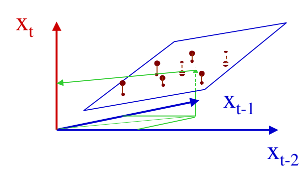
- Q1: Can it work with window *w*>1?
- A1: YES! (we'll fit a hyper-plane, then!)






## More details:

- Q1: Can it work with window *w*>1?
- A1: YES! (we'll fit a hyper-plane, then!)






## More details:

- Q1: Can it work with window *w*>1?
- A1: YES! The problem becomes:

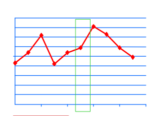
$$\mathbf{X}_{[N \times w]} \times \mathbf{a}_{[w \times 1]} = \mathbf{y}_{[N \times 1]}$$

- OVER-CONSTRAINED
  - **a** is the vector of the regression coefficients
  - **X** has the *N* values of the *w* indep. variables
  - **y** has the N values of the dependent variable



## More details:

- $\mathbf{X}_{[N \times w]} \times \mathbf{a}_{[w \times 1]} = \mathbf{y}_{[N \times 1]}$



Ind-var1 Ind-var-w

time

$$\begin{bmatrix} X_{11}, X_{12}, \dots, X_{1w} \\ X_{21}, X_{22}, \dots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{N1}, X_{N2}, \dots, X_{Nw} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_N \end{bmatrix}$$

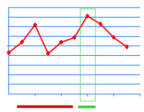
## More details:

- $\mathbf{X}_{[N \times w]} \times \mathbf{a}_{[w \times 1]} = \mathbf{y}_{[N \times 1]}$

Ind-var1 Ind-var-w

time

$$\begin{bmatrix} X_{11}, X_{12}, \cdots, X_{1w} \\ X_{21}, X_{22}, \cdots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{N1}, X_{N2}, \cdots, X_{Nw} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_N \end{bmatrix}$$







## More details

- Q2: How to estimate $a_1, a_2, \ldots a_w = \mathbf{a}$?
- A2: with Least Squares fit

$$\boxed{\mathbf{a} = (\mathbf{X}^T \times \mathbf{X})^{-1} \times (\mathbf{X}^T \times \mathbf{y})}$$

- (Moore-Penrose pseudo-inverse)
- **a** is the vector that minimizes the RMSE from **y**





## Even more details

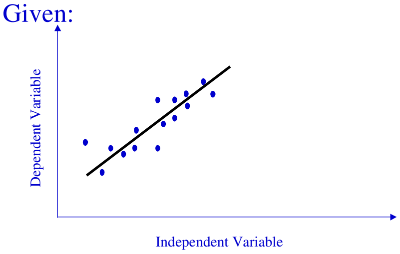
- Q3: Can we estimate **a** incrementally?
- A3: Yes, with the brilliant, classic method of 'Recursive Least Squares' (RLS) (see, e.g., [Yi+00], for details) - pictorially:





## Even more details

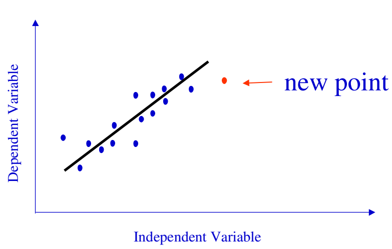
- Given:








## Even more details

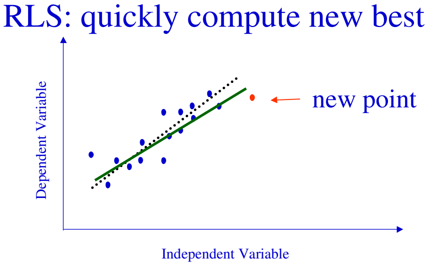







## Even more details

RLS: quickly compute new best fit

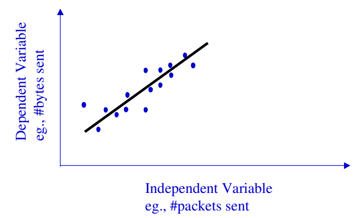


# Even more details

- Q4: can we ‘forget’ the older samples?
- A4: Yes - RLS can easily handle that [Yi+00]:

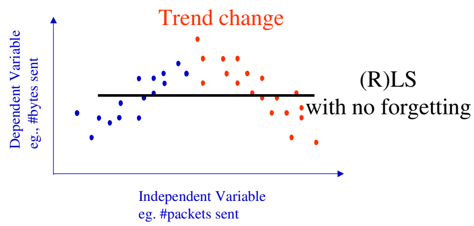


# Adaptability - ‘forgetting’



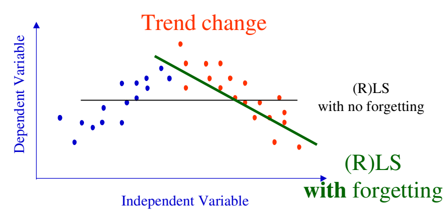



# Adaptability - ‘forgetting’






# Adaptability - ‘forgetting’




- RLS: can *trivially* handle ‘forgetting’



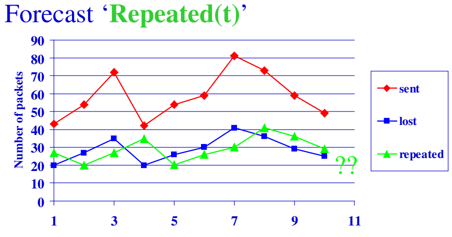
# B.II - Time Series Analysis - Outline

- Auto-regression
- Least Squares; recursive least squares
- Co-evolving time sequences
- Examples
- Conclusions



# Co-Evolving Time Sequences

- Given: A set of **correlated** time sequences
- Forecast ‘**Repeated(t)**’



## Solution:

Q: what should we do?

## Solution:

Least Squares, with

- Dep. Variable: Repeated(t)
- Indep. Variables: Sent(t-1) … Sent(t-w); Lost(t-1) …Lost(t-w); Repeated(t-1), ...
- (named: 'MUSCLES' [Yi+00])

## B.II - Time Series Analysis - Outline

- Auto-regression
- Least Squares; recursive least squares
- Co-evolving time sequences
- Examples
- Conclusions

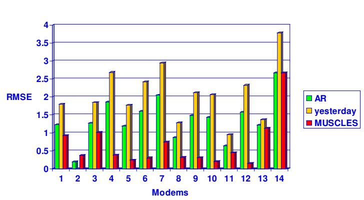
  

## Examples - Experiments

- Datasets
  - Modem pool traffic (14 modems, 1500 time-ticks; #packets per time unit)
  - AT&T WorldNet internet usage (several data streams; 980 time-ticks)
- Measures of success
  - Accuracy : Root Mean Square Error (RMSE)

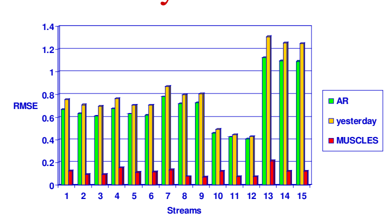
## Accuracy - "Modem"




MUSCLES outperforms AR & "yesterday"

## Accuracy - "Internet"




MUSCLES consistently outperforms AR & "yesterday"

## B.II - Time Series Analysis - Outline

- Auto-regression
- Least Squares; recursive least squares
- Co-evolving time sequences
- Examples
- Conclusions



## Conclusions - Practitioner's guide

- AR(IMA) methodology: prevailing method for linear forecasting
- Brilliant method of Recursive Least Squares for fast, incremental estimation.
- See [Box-Jenkins]



## Just a moment

Q: ARIMA - how about 'I' and 'MA'?

A1: 'I' - Integration (actually, differentiation - apply AR to $\Delta x_t$ ($= x_t - x_{t-1}$)

A2: 'MA': Moving Average (see book by Box-Jenkins - also: ARFIMA for 'F'ractional integration, GARFIMA etc)



## Table Overview

| | Know | Don't Know | How to learn more |
|---|---|---|---|
| Topology | **Powerlaws, jellyfish** | **Growth pattern, Compare graphs** | |
| Link | **LRD, ON/OFF sources** | **Effect of topology and protocols** | **ARIMA, wavelets** |
| End-2-end | **LRD loss and RTT** | **Troubleshoot, cluster and predict** | **ARIMA, wavelets** |
| Traffic Matrix | **Skewness of location** | **Comprehensive model, troubleshoot** | |



## Resources - software and urls

- http://www.dsptutor.freeuk.com/jsanalyser/FFTSpectrumAnalyser.html : Nice java applets for FFT
- http://www.relisoft.com/freeware/freq.html voice frequency analyzer (needs microphone)



## Resources: software and urls

- *xwpl:* open source wavelet package from Yale, with excellent GUI
- http://monet.me.ic.ac.uk/people/gavin/java/waveletDemos.html : wavelets and scalograms
- MUSCLES (`christos@cs.cmu.edu`)

## Books

- William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery: *Numerical Recipes in C*, Cambridge University Press, 1992, 2nd Edition. (Great description, intuition and code for DFT, DWT)
- C. Faloutsos: *Searching Multimedia Databases by Content*, Kluwer Academic Press, 1996 (introduction to DFT, DWT)
- George E.P. Box and Gwilym M. Jenkins and Gregory C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice Hall, 1994 (the classic book on ARIMA, 3rd ed.)



## Additional Reading

- [Gilbert+01] Anna C. Gilbert, Yannis Kotidis and S. Muthukrishnan and Martin Strauss, *Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries*, VLDB 2001
- [Riedi+99] R. Riedi, M. Crouse, V Ribeiro, R. Baraniuk, *A Multifractal Wavelet Model with Application to Network Traffic*, IEEE Trans. On Inf. Theory, 45,3, April 1999
- [Yi+00] Byoung-Kee Yi et al.: *Online Data Mining for Co-Evolving Time Sequences*, ICDE 2000. (Describes MUSCLES and Recursive Least Squares)



# Time for a break!



## Data Mining the Internet

Part B: HOW TO FIND MORE

*C. Faloutsos*

# Part B - III and IV new tools: SVD and fractals



## High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws

## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
- Solutions to posed problems
- Conclusions



## SVD - Motivation

- problem #1: find patterns in a matrix
  - (e.g., traffic patterns from several IP-sources)
  - compression; dim. reduction
- problem#2: find most ‘interesting’ node in a graph (google/Kleinberg-style)



## Problem#1

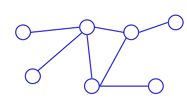
- ~10**6 rows; ~10**3 columns; no updates;
- Compress / find patterns

| day / customer | We 7/10/96 | Th 7/11/96 | Fr 7/12/96 | Sa 7/13/96 | Su 7/14/96 |
|---|---|---|---|---|---|
| ABC Inc. | 1 | 1 | 1 | 0 | 0 |
| DEF Ltd. | 2 | 2 | 2 | 0 | 0 |
| GHI Inc. | 1 | 1 | 1 | 0 | 0 |
| KLM Co. | 5 | 5 | 5 | 0 | 0 |
| Smith | 0 | 0 | 0 | 2 | 2 |
| Johnson | 0 | 0 | 0 | 3 | 3 |
| Thompson | 0 | 0 | 0 | 1 | 1 |



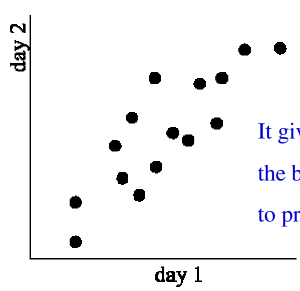
## Problem#2

Given a graph, find its most interesting/central node





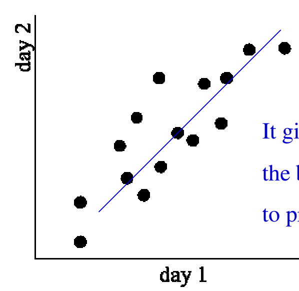
## SVD - in short:




It gives
the best hyperplane
to project on



## SVD - in short:




It gives
the best hyperplane
to project on

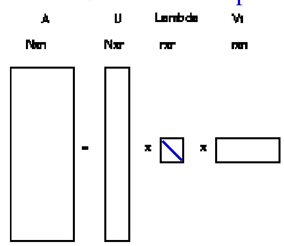
## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
- Solutions to posed problems
- Conclusions



## SVD - Definition

- **A = U Λ V**$^T$ - example:






## SVD - notation

Conventions:

- bold capitals -> matrix (eg. **A**, **U**, **Λ**, **V**)
- bold lower-case -> column vector (eg., **x**, $\mathbf{v}_1$, $\mathbf{u}_3$)
- regular lower-case -> scalars (eg., $\lambda_1$ , $\lambda_r$ )



## SVD - Definition

$$\boxed{\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} (\mathbf{V}_{[m \times r]})^T}$$

- **A**: n x m matrix (eg., n customers, m days)
- **U**: n x r matrix (n customers, r concepts)
- **Λ**: r x r diagonal matrix (strength of each 'concept') (r : rank of the matrix)
- **V**: m x r matrix (m days, r concepts)



## SVD - Properties

**THEOREM** [Press+92]: always possible to decompose matrix **A** into **A = U Λ V**$^T$ , where

- **U, Λ, V**: unique (*)
- **U**, **V**: column orthonormal (ie., columns are unit vectors, orthogonal to each other)
  - $\mathbf{U}^T \mathbf{U} = \mathbf{I}$; $\mathbf{V}^T \mathbf{V} = \mathbf{I}$ (**I:** identity matrix)
- **Λ:** eigenvalues are positive, and sorted in decreasing order



## SVD - example

- Customers; days; #packets

| | day / customer | We 7/10/96 | Th 7/11/96 | Fr 7/12/96 | Sa 7/13/96 | Su 7/14/96 |
|---|---|---|---|---|---|---|
| Comm. | ABC Inc. | 1 | 1 | 1 | 0 | 0 |
| | DEF Ltd. | 2 | 2 | 2 | 0 | 0 |
| | GHI Inc. | 1 | 1 | 1 | 0 | 0 |
| | KLM Co. | 5 | 5 | 5 | 0 | 0 |
| Res. | Smith | 0 | 0 | 0 | 2 | 2 |
| | Johnson | 0 | 0 | 0 | 3 | 3 |
| | Thompson | 0 | 0 | 0 | 1 | 1 |

# SVD - Example

- $\mathbf{A} = \mathbf{U}\,\Lambda\,\mathbf{V}^T$ - example:

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} \text{We} & \text{Th.} & \text{Fr} & \text{Sa} & \text{Su} \end{array} \\
\begin{array}{c} \uparrow \\ \text{Com.} \\ \downarrow \\ \uparrow \\ \text{Res.} \\ \downarrow \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\text{x}
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\text{x}
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$



# B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
  - #1: customers, days, concepts
  - #2: best projection - dimensionality reduction
  - #3: fixed point
- Solutions to posed problems
- Conclusions



# SVD - Interpretation #1

'customers', 'days' and 'concepts'

- **U**: customer-to-concept similarity matrix
- **V**: day-to-concept sim. matrix
- $\Lambda$: its diagonal elements: 'strength' of each concept



# SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\,\Lambda\,\mathbf{V}^T$ - example:

Rank=2

2x2

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} \text{We} & \text{Th.} & \text{Fr} & \text{Sa} & \text{Su} \end{array} \\
\begin{array}{c} \uparrow \\ \text{Com.} \\ \downarrow \\ \uparrow \\ \text{Res.} \\ \downarrow \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\text{x}
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\text{x}
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$



# SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\,\Lambda\,\mathbf{V}^T$ - example:

Rank=2

=2 'concepts'

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} \text{We} & \text{Th.} & \text{Fr} & \text{Sa} & \text{Su} \end{array} \\
\begin{array}{c} \uparrow \\ \text{Com.} \\ \downarrow \\ \uparrow \\ \text{Res.} \\ \downarrow \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\text{x}
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\text{x}
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$



# (reminder)

- Customers; days; #packets

| | day / customer | We 7/10/96 | Th 7/11/96 | Fr 7/12/96 | Sa 7/13/96 | Su 7/14/96 |
|---|---|---|---|---|---|---|
| Comm. | ABC Inc. | 1 | 1 | 1 | 0 | 0 |
| | DEF Ltd. | 2 | 2 | 2 | 0 | 0 |
| | GHI Inc. | 1 | 1 | 1 | 0 | 0 |
| | KLM Co. | 5 | 5 | 5 | 0 | 0 |
| Res. | Smith | 0 | 0 | 0 | 2 | 2 |
| | Johnson | 0 | 0 | 0 | 3 | 3 |
| | Thompson | 0 | 0 | 0 | 1 | 1 |

## SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$ - example:

U: customer-to-concept similarity matrix

weakday-concept, W/end-concept

$$\begin{array}{c|c} & \text{We Th. Fr Sa Su} \\ \begin{array}{c}\text{Com.}\\ \\ \\ \\ \text{Res.}\\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathrm{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathrm{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$



## SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$ - example:

U: Customer to concept similarity matrix

weekday-concept, W/end-concept

$$\begin{array}{c|c} & \text{We Th. Fr Sa Su} \\ \begin{array}{c}\text{Com.}\\ \\ \\ \\ \text{Res.}\\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathrm{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathrm{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$



## SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$ - example:

unit

$$\begin{array}{c|c} & \text{We Th. Fr Sa Su} \\ \begin{array}{c}\text{Com.}\\ \\ \\ \\ \text{Res.}\\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathrm{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathrm{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$



## SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$ - example:

weekday-concept

Strength of 'weekday' concept

$$\begin{array}{c|c} & \text{We Th. Fr Sa Su} \\ \begin{array}{c}\text{Com.}\\ \\ \\ \\ \text{Res.}\\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathrm{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathrm{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$



## SVD - Interpretation #1

- $\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$ - example:

weekday-concept

**V:** day to concept similarity matrix

$$\begin{array}{c|c} & \text{We Th. Fr Sa Su} \\ \begin{array}{c}\text{Com.}\\ \\ \\ \\ \text{Res.}\\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \mathrm{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \mathrm{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$



## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
  - #1: customers, days, concepts
  - #2: best projection - dimensionality reduction
  - #3: fixed point
- Solutions to posed problems
- Conclusions

# SVD - Interpretation #2

- best axis to project on: ('best' = min sum of squares of projection errors)

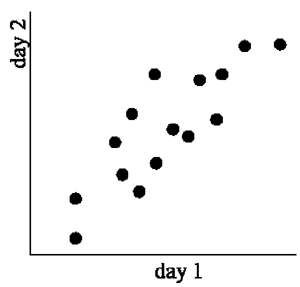


# SVD - Interpretation #2

| customer \ day | We 7/10/96 | Th 7/11/96 | Fr 7/12/96 | Sa 7/13/96 | Su 7/14/96 |
|---|---|---|---|---|---|
| ABC Inc. | 1 | 1 | 1 | 0 | 0 |
| DEF Ltd. | 2 | 2 | 2 | 0 | 0 |
| GHI Inc. | 1 | 1 | 1 | 0 | 0 |
| KLM Co. | 5 | 5 | 5 | 0 | 0 |
| Smith | 0 | 0 | 0 | 2 | 2 |
| Johnson | 0 | 0 | 0 | 3 | 3 |
| Thompson | 0 | 0 | 0 | 1 | 1 |



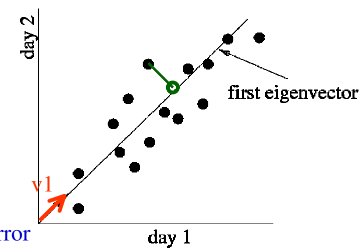
# SVD - Interpretation#2






# SVD - interpretation #2

SVD: gives best axis to project




- minimum RMS error



# SVD - Interpretation #2

- **A = U Λ V**$^T$ - example:

v1

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \text{x} \begin{bmatrix} \boxed{0.58 \;\; 0.58 \;\; 0.58 \;\; 0 \;\; 0} \\ 0 \;\; 0 \;\; 0 \;\; 0.71 \;\; 0.71 \end{bmatrix}$$



# SVD - Interpretation #2

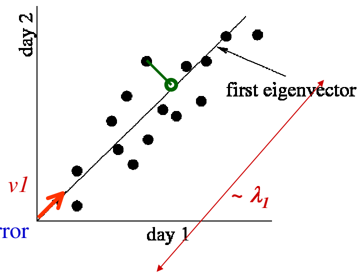
- **A = U Λ V**$^T$ - example:

variance ('spread') on the v1 axis

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

## SVD - interpretation #2

SVD: gives best axis to project




- minimum RMS error



## SVD, PCA and the v vectors

- how to 'read' the **v** vectors (= principal components)



## SVD

- Recall**:** **A** = **U** **Λ** **V**$^T$ - example:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \text{x} \begin{bmatrix} \boxed{0.58 \;\; 0.58 \;\; 0.58 \;\; 0 \;\; 0} \\ 0 \;\; 0 \;\; 0 \;\; 0.71 \;\; 0.71 \end{bmatrix}$$



## SVD

- First Principal component = **v1** -> weekdays are correlated positively
- similarly for **v2**
- (we'll see negative correlations later)

$$\begin{array}{c c} & \begin{array}{cc} \textbf{v1} & \textbf{v2} \end{array} \\ \begin{array}{c} \text{We} \\ \text{Th} \\ \text{Fr} \\ \text{Sa} \\ \text{Su} \end{array} & \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix} \end{array}$$



## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
  - #1: customers, days, concepts
  - #2: best projection - dimensionality reduction
  - #3: fixed point
- Solutions to posed problems
- Conclusions



## SVD - Interpretation #3

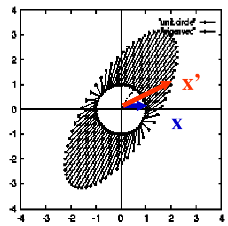
If **A** is symmetric,
**x** is an eigenvector of **A** if

$$\boxed{\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}}$$

## SVD - Interpretation #3

- **A** as vector transformation (assume **A** is symmetric)

$$\begin{array}{ccc} x' & A & x \\ \begin{bmatrix} 2 \\ 1 \end{bmatrix} = & \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array}$$

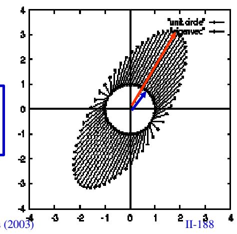





## SVD - Interpretation #3

- For a symmetric **A**, by defn. its eigenvectors remain parallel to themselves (‘**fixed points**’)

$$\begin{array}{cccc} \lambda_1 & \mathbf{v_1} & A & \mathbf{v_1} \\ 3.62 * & \begin{bmatrix} 0.52 \\ 0.85 \end{bmatrix} = & \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} & \begin{bmatrix} 0.52 \\ 0.85 \end{bmatrix} \end{array}$$





## SVD - Interpretation #3

- If **A** is not symmetric, then $\mathbf{A^T A}$ always is (= ‘day-to-day’ similarity matrix)



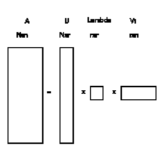
## SVD - Complexity

- O( n * m * m) or O( n * n * m) (whichever is less)
- less work, if we just want eigenvalues
- ... or if we want first k eigenvectors
- ... or if the matrix is sparse [Berry]
- Implemented: in *any* linear algebra package (LINPACK, matlab, Splus, mathematica ...)



## SVD - conclusions so far

- SVD: $\mathbf{A= U \Lambda V^T}$ : unique (*)
- **U**: row-to-concept similarities
- **V**: column-to-concept similarities
- **Λ**: strength of each concept



(*) see [Press+92]



## SVD - conclusions so far

- dim. reduction: keep the first few strongest eigenvalues (80-90% of ‘energy’ [Fukunaga])
- SVD: picks up linear correlations
- $\mathbf{v}_1$: fixed point (-> steady-state prob.)

## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
- Solutions to posed problems
  - P1: patterns in a matrix; **compression**
  - P2: most 'important' node in a graph
- Conclusions



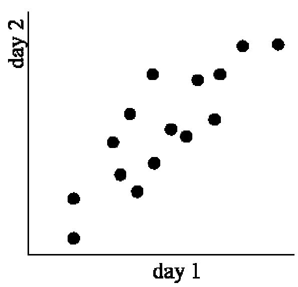
## Problem #1 - specs

- ~10**6 rows; ~10**3 columns; no updates;
- random access to any cell(s) ; small error: OK
- compress ; find patterns / rules

| day<br>customer | We<br>7/10/96 | Th<br>7/11/96 | Fr<br>7/12/96 | Sa<br>7/13/96 | Su<br>7/14/96 |
|---|---|---|---|---|---|
| ABC Inc. | 1 | 1 | 1 | 0 | 0 |
| DEF Ltd. | 2 | 2 | 2 | 0 | 0 |
| GHI Inc. | 1 | 1 | 1 | 0 | 0 |
| KLM Co. | 5 | 5 | 5 | 0 | 0 |
| Smith | 0 | 0 | 0 | 2 | 2 |
| Johnson | 0 | 0 | 0 | 3 | 3 |
| Thompson | 0 | 0 | 0 | 1 | 1 |

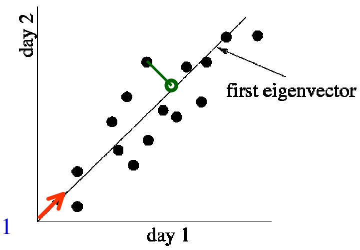


## Idea






## SVD to the rescue




- space savings: 2:1
- minimum RMS error

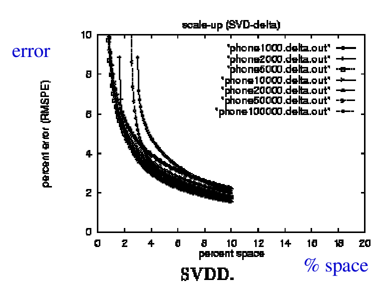


## Compression - Performance

- 3 pass algo (-> scalability)
- random cell(s) reconstruction
- 10:1 compression with < 2% error
- [Korn+, 97]



## Performance - scaleup



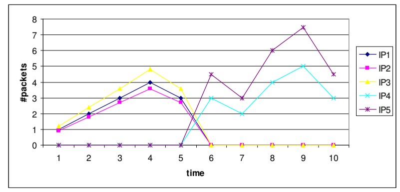
## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
- Solutions to posed problems
  - P1: **patterns** in a matrix; compression
  - P2: most ‘important’ node in a graph
- Conclusions



## SVD & visualization:

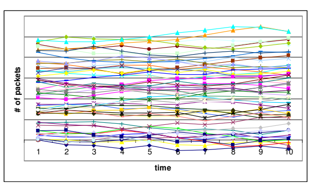
- Visualization for free!
  - Time-plots are not enough:






## SVD & visualization:

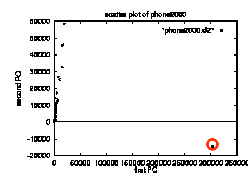
- Visualization for free!
  - Time-plots are not enough:






## SVD & visualization

- SVD: project 365-d vectors to best 2 dimensions, and plot:
- no Gaussian clusters; Zipf-like distribution



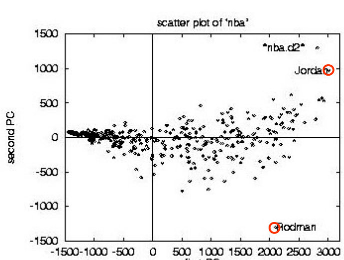

(a) ‘phone2000’

phonecalls



## SVD and visualization

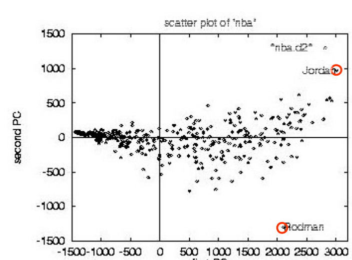
NBA dataset
~500 players;
~30 attributes
(#games, #points, #rebounds,…)






## SVD and visualization

could be network dataset:
- *N* IP sources
- *k* attributes (#http bytes, #http packets)



## Moreover, PCA/rules for free!

- SVD ~ PCA = Principal component analysis
- PCA: get eigenvectors **v1**, **v2**, ...
- ignore entries with small abs. value
- try to interpret the rest



## PCA & Rules

NBA dataset - **V** matrix (term to ‘concept’ similarities)

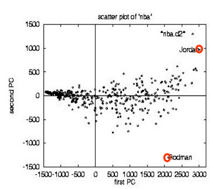
| field | $RR_1$ | $RR_2$ | $RR_3$ |
|---|---|---|---|
| minutes played | .808 | −.4 | |
| field goals | | | |
| goal attempts | | | |
| points | .406 | .199 | |
| total rebounds | | −.489 | .602 |
| assists | | | −.486 |
| steals | | | −.07 |

v1



## PCA & Rules

- (Ratio) Rule#1: minutes:points = 2:1
- corresponding concept?




v1



## PCA & Rules

- RR1: minutes:points = 2:1
- corresponding concept?
- A: ‘goodness’ of player
- (in a networks setting, could be ‘volume of traffic’ generated by this IP address)



## PCA & Rules

- RR2: points: rebounds negatively correlated(!)

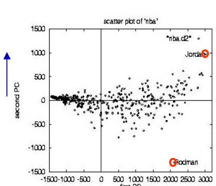
| field | $RR_1$ | $RR_2$ | $RR_3$ |
|---|---|---|---|
| minutes played | .808 | −.4 | |
| field goals | | | |
| goal attempts | | | |
| points | .406 | .199 | |
| total rebounds | | −.489 | .602 |
| assists | | | −.486 |
| steals | | | −.07 |



## PCA & Rules

- RR2: points: rebounds negatively correlated(!) - concept?

v2



# PCA & Rules

- RR2: points: rebounds negatively correlated(!) - concept?
- A: position: offensive/defensive
- (in a network setting, could be e-mailers versus gnutella-users)



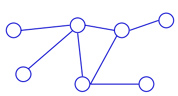
# B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
- Solutions to posed problems
  - P1: patterns in a matrix; compression
  - P2: most 'important' node in a graph
- Conclusions



# Problem#2

Given a graph, find its most interesting/central node

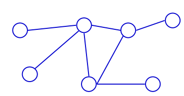




# Problem#2

Given a graph, find its most interesting/central node

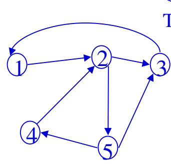
Proposed solution: Random walk; spot most 'popular' node (-> steady state prob.)





# google/page-rank algorithm

- Let **A** be the transition matrix (= adjacency matrix); let $\mathbf{A^T}$ become column-normalized - then




From

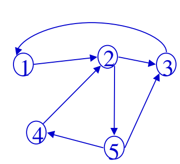
To

$$\mathbf{A^T} \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$



# google/page-rank algorithm

- $\mathbf{A^T\ p = p}$




$$\mathbf{A^T} \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix} \mathbf{p} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \mathbf{p} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$

## google/page-rank algorithm

- $A^T$ $p = 1 * p$
- thus, **p** is the eigenvector that corresponds to the highest eigenvalue (=1, since the matrix is column-normalized)

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-217

## google/page-rank algorithm

- In short: imagine a particle randomly moving along the edges (*)
- compute its steady-state probabilities

(*) with occasional random jumps and back-tracks

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-218

## Kleinberg's algorithm

- Kleinberg's algorithm of 'hubs' and 'authorities': closely related [Kleinberg'98]
- (and still based on SVD of the adjacency matrix)

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-219

## Kleinberg's algorithm - results

Eg., for the query 'java':

0.328 www.gamelan.com

0.251 java.sun.com

0.190 www.digitalfocus.com ("the java developer")

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-220

## B.III - SVD - outline

- Introduction - motivating problems
- Definition - properties
- Interpretation / Intuition
- Solutions to posed problems
  - P1: patterns in a matrix; compression
  - P2: most 'important' node in a graph
- Conclusions

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-221

## SVD - conclusions

SVD: a **valuable** tool , whenever we have a matrix, e.g.

- many time sequences
- many feature vectors
- graph (-> adjacency matrix)

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-222

# SVD - conclusions

SVD: a **valuable** tool , whenever we have a matrix, e.g.

- many time sequences
  - SVD finds groups
  - principal components
  - dim. reduction

| | #packets on day1 | #packets on day2 | ... | | |
|---|---|---|---|---|---|
| IP address1 | 1 | 1 | 1 | 0 | 0 |
| IP address2 | 2 | 2 | 2 | 0 | 0 |
| IP address3 | 1 | 1 | 1 | 0 | 0 |
| ... | 5 | 5 | 5 | 0 | 0 |
| | 0 | 0 | 0 | 2 | 2 |
| | 0 | 0 | 0 | 3 | 3 |
| | 0 | 0 | 0 | 1 | 1 |



# SVD - conclusions

SVD: a **valuable** tool , whenever we have a matrix, e.g.

- feature vectors
  - SVD finds groups
  - principal components
  - (Ratio) Rules
  - visualization

| | #packets sent | #bytes sent | #packets lost | ... | |
|---|---|---|---|---|---|
| IP address1 | 1 | 1 | 1 | 0 | 0 |
| IP address2 | 2 | 2 | 2 | 0 | 0 |
| IP address3 | 1 | 1 | 1 | 0 | 0 |
| ... | 5 | 5 | 5 | 0 | 0 |
| | 0 | 0 | 0 | 2 | 2 |
| | 0 | 0 | 0 | 3 | 3 |
| | 0 | 0 | 0 | 1 | 1 |



# SVD - conclusions

SVD: a **valuable** tool , whenever we have a matrix, e.g.

- adjacency matrix
  - source, dest, bandwidth
  - SVD -> 'most central node'

| | Dest. router1 | Dest. router2 | Dest. router3 | ... | |
|---|---|---|---|---|---|
| Source router1 | 1 | 1 | 1 | 0 | 0 |
| Source router2 | 2 | 2 | 2 | 0 | 0 |
| Source router3 | 1 | 1 | 1 | 0 | 0 |
| ... | 5 | 5 | 5 | 0 | 0 |
| | 0 | 0 | 0 | 2 | 2 |
| | 0 | 0 | 0 | 3 | 3 |
| | 0 | 0 | 0 | 1 | 1 |



# SVD - conclusions - cont'd

Has been used/re-invented **many times**:

- LSI (Latent Semantic Indexing) [Foltz+92]
- PCA (Principal Component Analysis) [Jolliffe86]
- KL (Karhunen-Loeve Transform)
- Mahalanobis distance
- ...



# Table Overview

| | Know | Don't Know | How to learn more |
|---|---|---|---|
| Topology | **Powerlaws, jellyfish** | **Growth pattern, Compare graphs** | **SVD** |
| Link | **LRD, ON/OFF sources** | **Effect of topology and protocols** | **SVD** |
| End-2-end | **LRD loss and RTT** | **Troubleshoot, cluster and predict** | **SVD** |
| Traffic Matrix | **Skewness of location** | **Comprehensive model, troubleshoot** | **SVD** |



# Resources: Software and urls

- SVD packages: in **many** systems (matlab, mathematica, LINPACK, LAPACK)
- stand-alone, free code: SVDPACK from Michael Berry

**http://www.cs.utk.edu/~berry/projects.html**

# Books

- Faloutsos, C. (1996). *Searching Multimedia Databases by Content*, Kluwer Academic Inc.
- Jolliffe, I. T. (1986). *Principal Component Analysis*, Springer Verlag.





# Books

- [Press+92] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery: *Numerical Recipes in C*, Cambridge University Press, 1992, 2nd Edition. (Great description, intuition and code for SVD)





# Additional Reading

- Berry, Michael: http://www.cs.utk.edu/~lsi/
- Brin, S. and L. Page (1998). *Anatomy of a Large-Scale Hypertextual Web Search Engine*. 7th Intl World Wide Web Conf.





# Additional Reading

- [Foltz+92] Foltz, P. W. and S. T. Dumais (Dec. 1992). "*Personalized Information Delivery: An Analysis of Information Filtering Methods*." Comm. of ACM (CACM) 35(12): 51-60.





# Additional Reading

- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press.
- Kleinberg, J. (1998). *Authoritative sources in a hyperlinked environment*. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms.





# Additional Reading

- Korn, F., H. V. Jagadish, et al. (May 13-15, 1997). *Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences*. ACM SIGMOD, Tucson, AZ.
- Korn, F., A. Labrinidis, et al. (2000). "*Quantifiable Data Mining Using Ratio Rules*." VLDB Journal 8(3-4): 254-266.

# Part B - IV fractals



## High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws

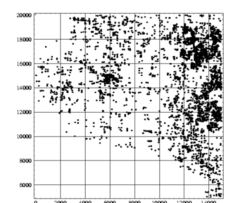


## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems
- More examples and tools
- Conclusions – practitioner's guide



## Problem #0: GIS - points



Road end-points of Montgomery county:

•Q1: # neighbors(*r*)?

•Q2 : distribution?

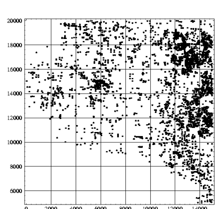
•not uniform

•not Gaussian

•no rules??



## Problem #0: GIS - points



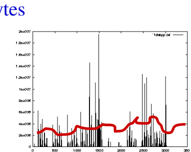
(could be: geo-locations of IP addresses launching DDoS attack)



## Problem #1: traffic

- disk trace (from HP - J. Wilkes); Web traffic - fit a model

#bytes



time

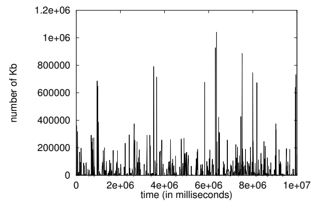
Poisson

**- how many explosions to expect?**

**- queue length distr.?**

## Problem #1’: traffic

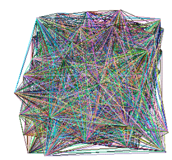
- Kb per unit time (requests on a web server)
  http://repository.cs.vt.edu/ lbl-conn-7.tar.Z




SIGMETRICS-03 © M. & C. Faloutsos (2003) II-241

## Problem #2 - topology

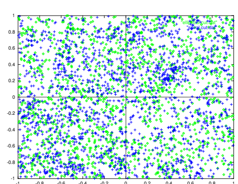
How does the Internet look like?



SIGMETRICS-03 © M. & C. Faloutsos (2003) II-242

## Problem #3 - spatial d.m.

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



**- ‘spiral’ and ‘elliptical’ galaxies**

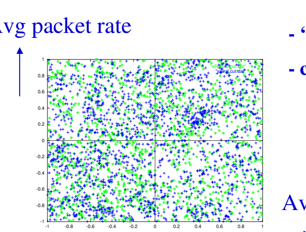
**- patterns?**

**- attraction/repulsion?**

**- separable?**

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-243

## Problem #3 - spatial d.m.

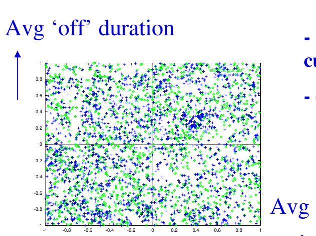



**- ‘good’ and ‘bad’ IP addresses**

**- can we separate them?**

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-244

## Problem #3 - spatial d.m.




**- ‘good’ and ‘bad’ customers / flows**

**- can we separate them?**

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-245

## Common answer:

Fractals / self-similarities / power laws

SIGMETRICS-03 © M. & C. Faloutsos (2003) II-246

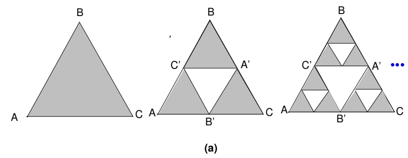
# B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems
- More examples and tools
- Conclusions – practitioner's guide



# What is a fractal?

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




**zero area;**

**infinite length!**



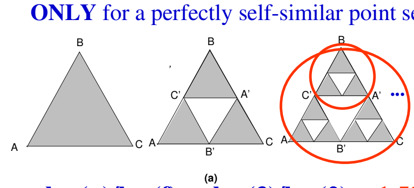
# Definitions (cont'd)

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



# Dfn of fd:

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




**zero area;**

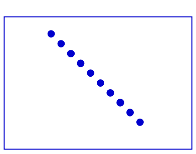
**infinite length!**

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



# Intrinsic ('fractal') dimension

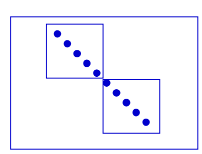
- Q: fractal dimension of a line?
- A: 1 (= log(2)/log(2)!)





# Intrinsic ('fractal') dimension

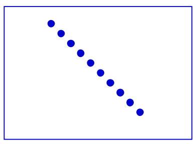
- Q: fractal dimension of a line?
- A: 1 (= log(2)/log(2)!)



## Intrinsic ('fractal') dimension

- Q: dfn for a given set of points?

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



SIGMETRICS-03  II-253

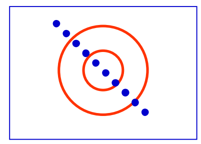
## Intrinsic ('fractal') dimension

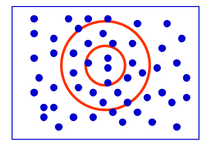
- Q: fractal dimension of a line?
- A: nn ( <= r ) ~ r^1

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

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

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





SIGMETRICS-03  II-254

## Intrinsic ('fractal') dimension

- Algorithm, to estimate it?

Notice

- *avg nn(<=r)* is exactly

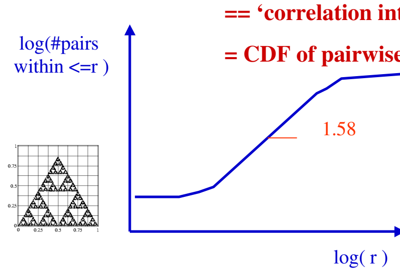
*tot#pairs(<=r) / (N)*

SIGMETRICS-03  II-255

## Sierpinsky triangle

**== 'correlation integral'**

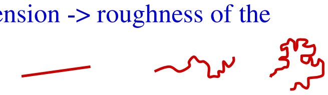
**= CDF of pairwise distances**




SIGMETRICS-03  II-256

## Observations:

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



SIGMETRICS-03  II-257

## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems
- More examples and tools
- Conclusions – practitioner's guide

SIGMETRICS-03  II-258

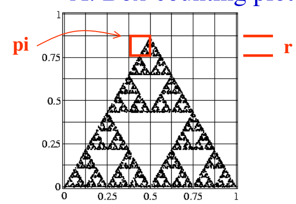
## Fast estimation

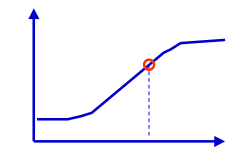
- Bad news: There are more than one fractal dimensions
  - Minkowski fd; Hausdorff fd; Correlation fd; Information fd
- Great news:
  - they can all be computed fast! (O(N); O(N logN))
  - Code is on the web (`www.cs.cmu.edu/~christos`)
  - they usually have nearby values



## Fast estimation of fd(s):

- How, for the (correlation) fractal dimension?
- A: Box-counting plot:



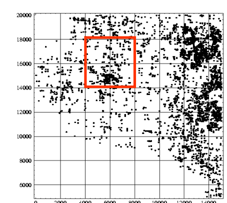



## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems: P#0 - points
- More examples and tools
- Conclusions – practitioner's guide



## Problem #0: GIS points

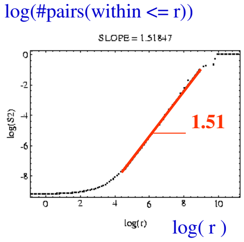


Cross-roads of Montgomery county:

•any rules?



## Solution #0



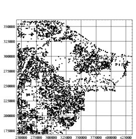

A: self-similarity ->

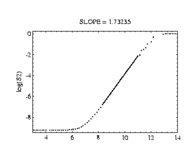
- **<=> fractals**
- **<=> scale-free**
- **<=> power-laws** (y=x^a, F=C*r^(-2))



## Examples:LB county

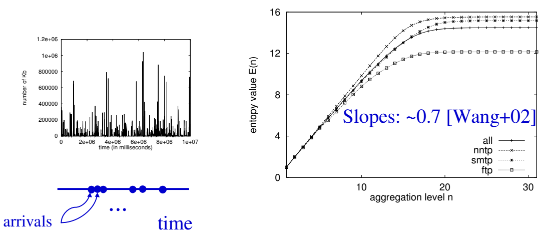
- Long Beach county of CA (road end-points)





## Example: traffic

- Kb per unit time (requests on a web server)





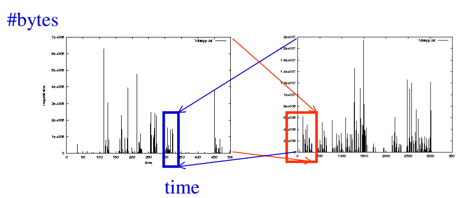

## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems: P#1- traffic
- More examples and tools
- Conclusions – practitioner's guide



## Solution #1: traffic

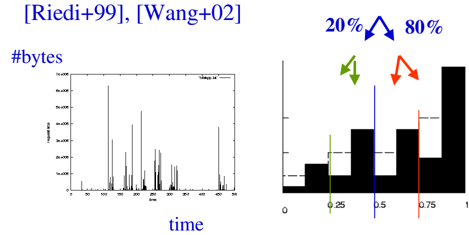
- disk traces: self-similar: (also: [Leland+94])
- How to generate such traffic?






## Solution #1: traffic

- disk traces (80-20 'law' = '**multifractal**') [Riedi+99], [Wang+02]





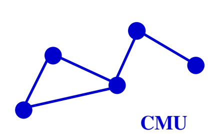

## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems: P#2 - topology
- More examples and tools
- Conclusions – practitioner's guide



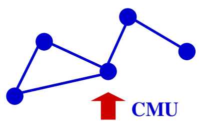
## Problem#2: Internet topology

- How does the internet look like?



## Problem#2: Internet topology

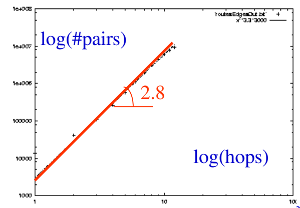
- How does the internet look like?
- Internet routers: how many neighbors within *h* hops?






## Problem#2: Internet topology

- Internet routers: how many neighbors within *h* hops? (= **correlation integral**!)



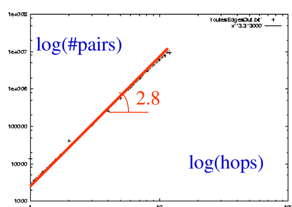

Reachability function: number of neighbors within r hops, vs r (log-log).

Mbone routers, 1995



## Problem#2: Internet topology

- Internet routers: how many neighbors within *h* hops?




Reachability function: number of neighbors within r hops

Q: How to compute it quickly?

A: [Palmer+01]



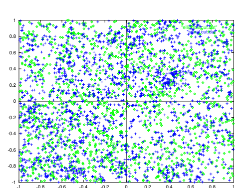
## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems: P#3: spatial d.m.
- More examples and tools
- Conclusions – practitioner's guide

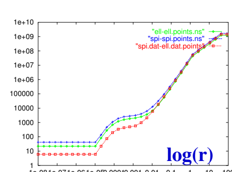


## Solution#3: spatial d.m.

Galaxies ( 'BOPS' plot - [sigmod2000])

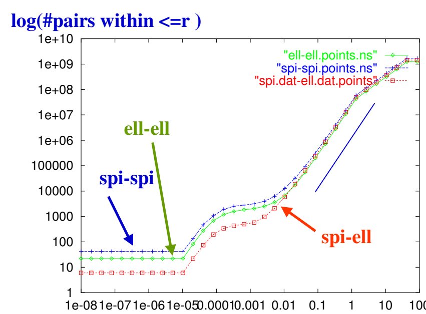


**log(#pairs(<=r))**






## Solution#3: spatial d.m.

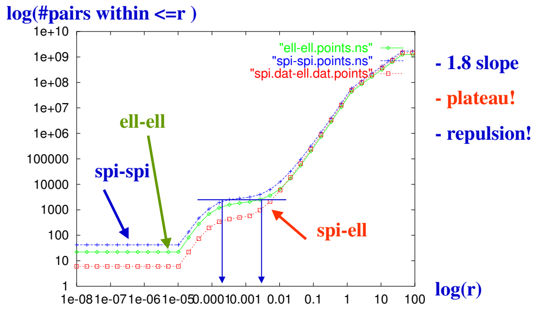



**- 1.8 slope**

**- plateau!**

**- repulsion!**

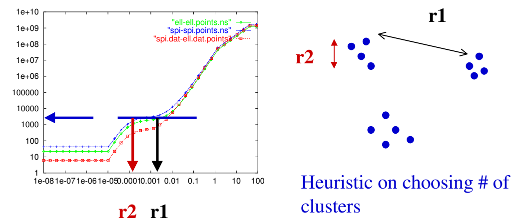
## spatial d.m.




- 1.8 slope
- plateau!
- repulsion!



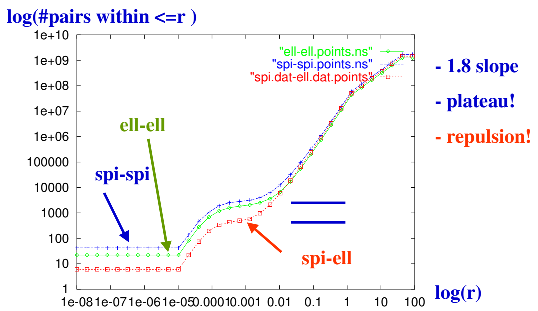
## spatial d.m.




Heuristic on choosing # of clusters



## spatial d.m.




- 1.8 slope
- plateau!
- repulsion!



## B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems
- More examples and tools
- Conclusions – practitioner's guide



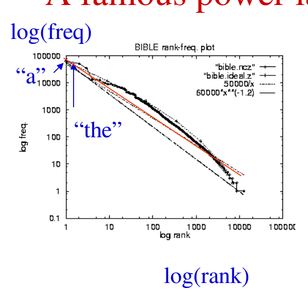
## Fractals and power laws

Recall that they are related concepts:

- fractals <=>
- self-similarity <=>
- scale-free <=>
- power laws ( $y= x^a$ )



## A famous power law: Zipf's law




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

# Power laws, cont'ed

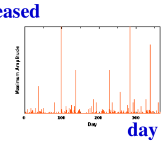
- In- and out-degree distribution of web sites [Barabasi], [IBM-CLEVER]
- length of file transfers [Bestavros+]
- Click-stream data [Montgomery+01]
- web hit counts [Huberman]



# More power laws

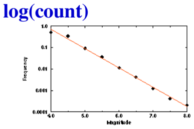
- duration of UNIX jobs; of UNIX file sizes
- Energy of earthquakes (Gutenberg-Richter law) [simscience.org]

**Energy released**



**day**

**log(count)**



**Magnitude = log(energy)**



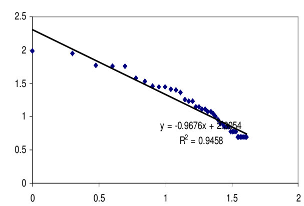
# Even more power laws:

- Income distribution (Pareto's law)
- publication counts (Lotka's law)



# Olympic medals (Sidney):

log(#medals)




log(rank)



# Fractals

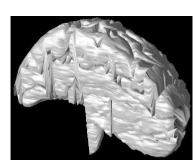
Let's see some fractals, in real settings:

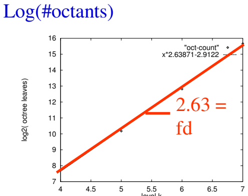


# Fractals: Brain scans

- Oct-trees; brain-scans

Log(#octants)






octree levels

# Fractals: Medical images

[Burdett et al, SPIE '93]:

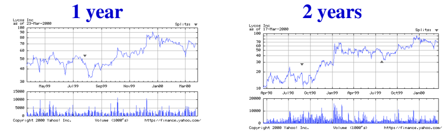
- benign tumors: fd ~ 2.37
- malignant: fd ~ 2.56



# More fractals:

- cardiovascular system: 3 (!) 
- stock prices (LYCOS) - random walks: 1.5

**1 year** **2 years**



- Coastlines: 1.2-1.58 (Norway!)






# B.IV - Fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Fast Estimation of fractal dimension
- Solutions to posed problems
- More examples and tools
- Conclusions – practitioner's guide



# Conclusions

- Real data often **disobey** textbook assumptions (Gaussian, Poisson, uniformity, independence)
  - avoid 'mean' - use median, or even better, use:
- fractals, self-similarity, and power laws, to find patterns



# Practitioner's guide:

- Fractals: help characterize a (non-uniform) set of points
- Detect non-homogeneous regions (eg., legal login time-stamps may have different fd than intruders')

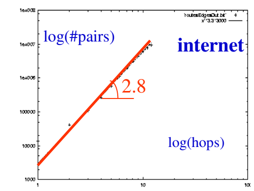
# Practitioner's guide

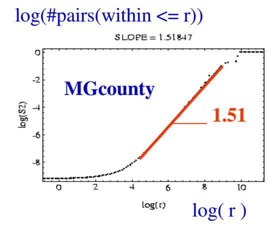
- **tool#1: (for points) 'correlation integral'**: (#pairs within <= *r*) vs (distance *r*)
- **tool#2: (for categorical values) rank-frequency** plot (a'la Zipf)



# Practitioner's guide:

- **tool#1**: correlation integral, for a **set of objects**, with a distance function (slope = intrinsic dimensionality)

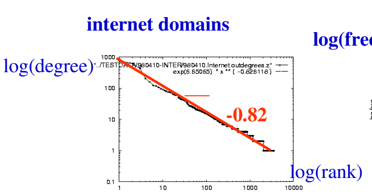


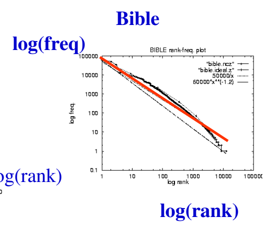






# Practitioner's guide:

- **tool#2**: rank-frequency plot (for **categorical attributes**)






# High-level Outline

- Part A - what we know about the Internet
- Part B - how to find more
  - B.I - Traditional Data Mining tools
  - B.II - Time series: analysis and forecasting
  - B.III - New Tools: SVD
  - B.IV - New Tools: Fractals & power laws
- 'Take-home' messages:

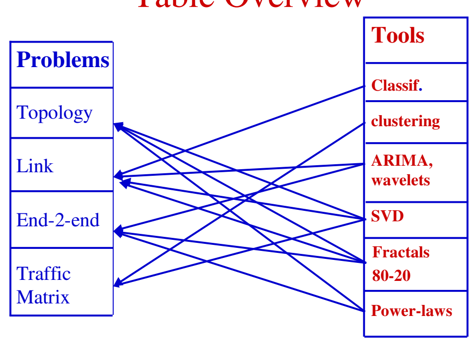


# Table Overview

| | Know | Don't Know | How to learn more |
|---|---|---|---|
| Topology | **Powerlaws, jellyfish** | **Growth pattern, Compare graphs** | **SVD, fractals** |
| Link | **LRD, ON/OFF sources** | **Effect of topology and protocols** | **ARIMA, wavelets, 80-20** |
| End-2-end | **LRD loss and RTT** | **Troubleshoot, cluster and predict** | **ARIMA, wavelets, 80-20** |
| Traffic Matrix | **Skewness of location** | **Comprehensive model, troubleshoot** | **Power-laws; multifractals, clustering** |



# Table Overview



## OVERALL CONCLUSIONS

- WEALTH of powerful, scalable tools in data mining (classification, clustering, SVD, fractals)
- traditional assumptions (uniformity, iid, Gaussian, Poisson) are often violated, when fractals/self-similarity/power-laws deliver.





## Resources: Software & urls

- Fractal dimensions: Software
  - **www.cs.cmu.edu/~christos**





## Books

- Fractals: Manfred Schroeder: *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise* W.H. Freeman and Company, 1991 (Probably the BEST book on fractals!)





## Further reading:


- [Barabasi+] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi, *Diameter of the World Wide Web*, Nature 401 130-131 (1999).
- [Kumar+99] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. *Extracting large scale knowledge bases from the web*. (VLDB) , September 1999.






## Further reading:


- [sigcomm99] Michalis Faloutsos, Petros Faloutsos and Christos Faloutsos, *What does the Internet look like? Empirical Laws of the Internet Topology,* SIGCOMM 1999
- [sigmod2000] Christos Faloutsos, Bernhard Seeger, Agma J. M. Traina and Caetano Traina Jr., *Spatial Join Selectivity Using Power Laws*, SIGMOD 2000
- [ieeeTN94] W. E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, *On the Self-Similar Nature of Ethernet Traffic,* IEEE Transactions on Networking, 2, 1, pp 1-15, Feb. 1994.






## Further reading


- [Montgomery+01] A. Montgomery and C. Faloutsos, *Identifying Web Browsing Trends and Patterns*, IEEE Computer, 2001
- [Palmer+01] Chris Palmer, Georgios Siganos, Michalis Faloutsos, Christos Faloutsos and Phil Gibbons: *The connectivity and fault-tolerance of the Internet topology* Workshop on Network Related Data Management (NRDM 2001), Santa Barbara, CA, May 25, 2001.

# Further reading

- [Riedi+99] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk, *A Multifractal Wavelet Model with Application to Network Traffic*, IEEE Special Issue on Information Theory, 45. (April 1999), 992-1018.
- [Wang+02] Mengzhi Wang, Tara Madhyastha, Ngai Hang Chang, Spiros Papadimitriou and Christos Faloutsos, *Data Mining Meets Performance Evaluation: Fast Algorithms for Modeling Bursty Traffic*, ICDE 2002, San Jose, CA, 2/26/2002 - 3/1/2002.





# THANK YOU!

michalis@cs.ucr.edu
www.cs.ucr.edu/~michalis

christos@cs.cmu.edu
www.cs.cmu.edu/~christos