

CMU SCS

15-826: Multimedia Databases and Data Mining

Lecture # 25: Time series mining and
forecasting
Christos Faloutsos

CMU SCS

Reading Material

- Byong-Kee Yi, Nikolaos D. Sidiropoulos, Theodore Johnson, H.V. Jagadish, Christos Faloutsos and Alex Biliris, [*Online Data Mining for Co-Evolving Time Sequences*](#), ICDE 2000, Feb. 2000.

15-826 (c) C. Faloutsos, 2008 2

CMU SCS

Thanks



Deepay Chakrabarti (CMU)



Prof. Dimitris Gunopulos (UCR)



Spiros Papadimitriou (CMU)



Mengzhi Wang (CMU)



Prof. Byoung-Kee Yi (Pohang U.)

15-826 (c) C. Faloutsos, 2008 3

CMU SCS

Outline

- ➔ • Motivation
 - Similarity search – distance functions
 - Linear Forecasting
 - Bursty traffic - fractals and multifractals
 - Non-linear forecasting
 - Conclusions

15-826 (c) C. Faloutsos, 2008 4

CMU SCS

Problem definition

- Given: one or more sequences
 - $x_1, x_2, \dots, x_t, \dots$
 - $(y_1, y_2, \dots, y_p, \dots$
 - $\dots)$
- Find
 - similar sequences; forecasts
 - patterns; clusters; outliers

15-826 (c) C. Faloutsos, 2008 5

CMU SCS

Motivation - Applications

- Financial, sales, economic series
- Medical
 - ECGs +; blood pressure etc monitoring
 - reactions to new drugs
 - elderly care

15-826 (c) C. Faloutsos, 2008 6

CMU SCS

Motivation - Applications (cont'd)

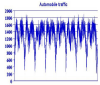
- 'Smart house'
 - sensors monitor temperature, humidity, air quality
- video surveillance

15-826 (c) C. Faloutsos, 2008 7

CMU SCS

Motivation - Applications (cont'd)

- civil/automobile infrastructure
 - bridge vibrations [Oppenheim+02]
 - road conditions / traffic monitoring

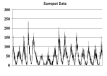


15-826 (c) C. Faloutsos, 2008 8

CMU SCS

Motivation - Applications (cont'd)

- Weather, environment/anti-pollution
 - volcano monitoring
 - air/water pollutant monitoring



15-826 (c) C. Faloutsos, 2008 9

CMU SCS

Motivation - Applications (cont'd)

- Computer systems
 - ‘Active Disks’ (buffering, prefetching)
 - web servers (ditto)
 - network traffic monitoring
 - ...

15-826 (c) C. Faloutsos, 2008 10

CMU SCS

Stream Data: Disk accesses

15-826 (c) C. Faloutsos, 2008 11

CMU SCS

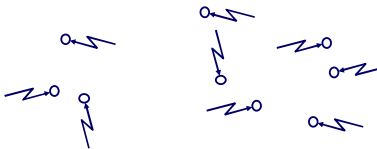
Settings & Applications

- One or more sensors, collecting time-series data

15-826 (c) C. Faloutsos, 2008 12

CMU SCS

Settings & Applications

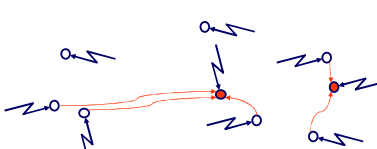


Each sensor collects data $(x_1, x_2, \dots, x_p, \dots)$

15-826 (c) C. Faloutsos, 2008 13

CMU SCS

Settings & Applications




Some sensors 'report' to others or to the central site

15-826 (c) C. Faloutsos, 2008 14

CMU SCS

Settings & Applications

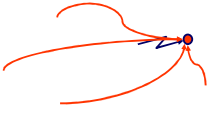


Goal #1:
Finding patterns
in a single time sequence

15-826 (c) C. Faloutsos, 2008 15

CMU SCS

Settings & Applications



Goal #2:
Finding patterns
in many time
sequences

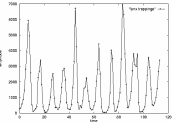
15-826 (c) C. Faloutsos, 2008 16

CMU SCS

Problem #1:

Goal: given a signal (e.g., #packets over time)
Find: patterns, periodicities, and/or compress

count



year

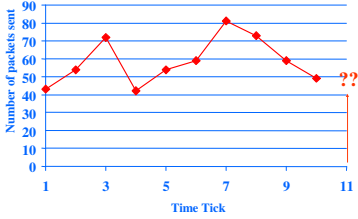
lynx caught per year
(packets per day;
temperature per day)

15-826 (c) C. Faloutsos, 2008 17

CMU SCS

Problem#2: Forecast

Given x_p, x_{t-1}, \dots , forecast x_{t+1}



15-826 (c) C. Faloutsos, 2008 18

CMU SCS

Problem#2': Similarity search

E.g., Find a 3-tick pattern, similar to the last one

Time Tick	Number of packets sent
1	45
2	55
3	75
4	45
5	55
6	65
7	85
8	75
9	65
10	55
11	45

15-826 (c) C. Faloutsos, 2008 19

CMU SCS

Problem #3:

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

Time Tick	sent	lost	repeated
1	45	25	30
2	55	30	25
3	75	40	35
4	45	25	30
5	55	30	25
6	65	35	30
7	85	45	40
8	75	40	35
9	65	30	25
10	55	25	20
11	45	20	15

15-826 (c) C. Faloutsos, 2008 20

CMU SCS

Differences from DSP/Stat

- Semi-infinite streams
 - we need on-line, 'any-time' algorithms
- Can not afford human intervention
 - need automatic methods
- sensors have limited memory / processing / transmitting power
 - need for (lossy) compression

15-826 (c) C. Faloutsos, 2008 21

CMU SCS

Important observations

Patterns, rules, forecasting and similarity indexing are closely related:

- To do forecasting, we need
 - to find patterns/rules
 - to find similar settings in the past
- to find outliers, we need to have forecasts
 - (outlier = too far away from our forecast)

15-826 (c) C. Faloutsos, 2008 22

CMU SCS

Important topics NOT in this presentation:

- Continuous queries
 - [Babu+Widom] [Gehrke+] [Madden+]
- Categorical data streams
 - [Hatonen+96]
- Outlier detection (discontinuities)
 - [Breunig+00]
- Related (see D. Shasha’s tutorial, SIGMOD 2004)

15-826 (c) C. Faloutsos, 2008 23

CMU SCS

Outline

- Motivation
- ➔ • Similarity Search and Indexing
- Linear Forecasting
- Bursty traffic - fractals and multifractals
- Non-linear forecasting
- Conclusions

15-826 (c) C. Faloutsos, 2008 24

CMU SCS

Outline

- Motivation
- ➔ • Similarity search and distance functions
 - Euclidean
 - Time-warping
- ...

15-826 (c) C. Faloutsos, 2008 25

CMU SCS

Importance of distance functions

Subtle, but **absolutely necessary**:

- A 'must' for similarity indexing (-> forecasting)
- A 'must' for clustering

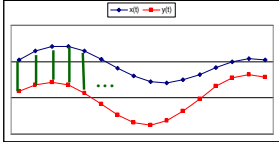
Two major families

- Euclidean and Lp norms
- Time warping and variations

15-826 (c) C. Faloutsos, 2008 26

CMU SCS

Euclidean and Lp



$$D(\vec{x}, \vec{y}) = \sum_{i=1}^n (x_i - y_i)^2$$

$$L_p(\vec{x}, \vec{y}) = \sum_{i=1}^n |x_i - y_i|^p$$

- L_1 : city-block = Manhattan
- L_2 = Euclidean
- L_∞

15-826 (c) C. Faloutsos, 2008 27

CMU SCS

Observation #1

- Time sequence -> n-d vector

15-826 (c) C. Faloutsos, 2008 28

CMU SCS

Observation #2

Euclidean distance is closely related to

- cosine similarity
- dot product
- 'cross-correlation' function

15-826 (c) C. Faloutsos, 2008 29

CMU SCS

Time Warping

- allow accelerations - decelerations
 - (with or w/o penalty)
- THEN compute the (Euclidean) distance (+ penalty)
- related to the string-editing distance

15-826 (c) C. Faloutsos, 2008 30

CMU SCS

Time Warping

'stutters':

15-826 (c) C. Faloutsos, 2008 31

CMU SCS

Time warping

Q: how to compute it?
 A: dynamic programming
 $D(i, j)$ = cost to match
 prefix of length i of first sequence x with prefix
 of length j of second sequence y

15-826 (c) C. Faloutsos, 2008 32

CMU SCS

Time warping

Thus, with no penalty for stutter, for sequences
 $x_1, x_2, \dots, x_i, \dots$ y_1, y_2, \dots, y_j

$$D(i, j) = \|x[i] - y[j]\| + \min \begin{cases} D(i-1, j-1) & \text{no stutter} \\ D(i, j-1) & \text{x-stutter} \\ D(i-1, j) & \text{y-stutter} \end{cases}$$

15-826 (c) C. Faloutsos, 2008 33

CMU SCS

Time warping

VERY SIMILAR to the string-editing distance

$$D(i, j) = \|x[i] - y[j]\| + \min \begin{cases} D(i-1, j-1) & \text{no stutter} \\ D(i, j-1) & \text{x-stutter} \\ D(i-1, j) & \text{y-stutter} \end{cases}$$

15-826 (c) C. Faloutsos, 2008 34

CMU SCS

Time warping

- Complexity: $O(M*N)$ - quadratic on the length of the strings
- **Many** variations (penalty for stutters; limit on the number/percentage of stutters; ...)
- popular in voice processing [Rabiner+Juang]

15-826 (c) C. Faloutsos, 2008 35

CMU SCS

Other Distance functions

- piece-wise linear/flat approx.; compare pieces [Keogh+01] [Faloutsos+97]
- ‘cepstrum’ (for voice [Rabiner+Juang])
 - do DFT; take log of amplitude; do DFT again!
- Allow for small gaps [Agrawal+95]

See tutorial by [Gunopulos + Das, SIGMOD01]

15-826 (c) C. Faloutsos, 2008 36

CMU SCS

Other Distance functions

- recently: parameter-free, MDL based [Keogh, KDD'04]

15-826 (c) C. Faloutsos, 2008 37

CMU SCS

Conclusions

Prevailing distances:

- Euclidean and
- time-warping


15-826 (c) C. Faloutsos, 2008 38

CMU SCS

Outline


- Motivation
- Similarity search and distance functions
- ➔ Linear Forecasting
- Bursty traffic - fractals and multifractals
- Non-linear forecasting
- Conclusions

15-826 (c) C. Faloutsos, 2008 39



Linear Forecasting


15-826 (c) C. Faloutsos, 2008 40



Forecasting

"Prediction is very difficult, especially about the future." - Nils Bohr
<http://www.hfac.uh.edu/MediaFutures/thoughts.html>

15-826 (c) C. Faloutsos, 2008 41



Outline

- Motivation
- ...
- Linear Forecasting
 - ➔ - Auto-regression: Least Squares; RLS
 - Co-evolving time sequences
 - Examples
 - Conclusions

15-826 (c) C. Faloutsos, 2008 42

CMU SCS

Reference

[Yi+00] Byoung-Kee Yi et al.: *Online Data Mining for Co-Evolving Time Sequences*, ICDE 2000.
(Describes MUSCLES and Recursive Least Squares)

15-826 (c) C. Faloutsos, 2008 43

CMU SCS

Problem#2: Forecast

- Example: give x_{t-1}, x_{t-2}, \dots , forecast x_t

15-826 (c) C. Faloutsos, 2008 44

CMU SCS

Forecasting: Preprocessing

MANUALLY:

remove trends

time

spot periodicities
7 days

time

15-826 (c) C. Faloutsos, 2008 45

CMU SCS

Problem#2: Forecast

- Solution: try to express x_t as a linear function of the past: x_{t-2}, x_{t-3}, \dots (up to a window of w)

Formally:

$$x_t \approx a_1 x_{t-1} + \dots + a_w x_{t-w} + \text{noise}$$

15-826 (c) C. Faloutsos, 2008 46

CMU SCS

(Problem: Back-cast; interpolate)

- Solution - interpolate: try to express x_t as a linear function of the past AND the future: $x_{t+1}, x_{t+2}, \dots, x_{t+w_{future}}; x_{t-1}, \dots, x_{t-w_{past}}$ (up to windows of w_{past}, w_{future})
- EXACTLY the same algo's

15-826 (c) C. Faloutsos, 2008 47

CMU SCS

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)')

15-826 (c) C. Faloutsos, 2008 48

CMU SCS

Linear Auto Regression:

Time	Packets Sent(t)
1	43
2	54
3	72
...	...
N	??

15-826 (c) C. Faloutsos, 2008 49

CMU SCS

Linear Auto Regression:

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

Number of packets sent (t)

Number of packets sent (t-1)

'lag-plot'

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

15-826 (c) C. Faloutsos, 2008 50

CMU SCS

Outline

- Motivation
- ...
- Linear Forecasting
 - ➔ - Auto-regression: **Least Squares; RLS**
 - Co-evolving time sequences
 - Examples
 - Conclusions

15-826 (c) C. Faloutsos, 2008 51

CMU SCS

More details:

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

15-826 (c) C. Faloutsos, 2008 52

CMU SCS

More details:

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

15-826 (c) C. Faloutsos, 2008 53

CMU SCS

More details:

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

15-826 (c) C. Faloutsos, 2008 54

CMU SCS

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
 - \mathbf{a} is the vector of the regression coefficients
 - \mathbf{X} has the N values of the w indep. variables
 - \mathbf{y} has the N values of the dependent variable

15-826 (c) C. Faloutsos, 2008 55

CMU SCS

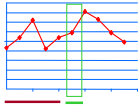
More details:

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

Ind-var1

time

Ind-var-w



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

15-826 (c) C. Faloutsos, 2008 56

CMU SCS

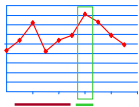
More details:

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

Ind-var1

time

Ind-var-w



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

15-826 (c) C. Faloutsos, 2008 57

CMU SCS

More details

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

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

- (Moore-Penrose pseudo-inverse)
- \mathbf{a} is the vector that minimizes the RMSE from \mathbf{y}
- <identical math with 'query feedbacks'>

15-826 (c) C. Faloutsos, 2008 58

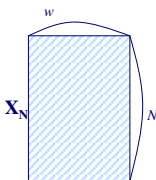
CMU SCS

More details

- Straightforward solution:

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

\mathbf{a} : Regression Coeff. Vector
 \mathbf{X} : Sample Matrix



• Observations:

- Sample matrix \mathbf{X} grows over time
- needs matrix inversion
- $\mathcal{O}(N \times w^2)$ computation
- $\mathcal{O}(N \times w)$ storage

15-826 (c) C. Faloutsos, 2008 59

CMU SCS

Even more details

- Q3: Can we estimate \mathbf{a} incrementally?
- A3: Yes, with the brilliant, classic method of 'Recursive Least Squares' (RLS) (see, e.g., [Yi+00], for details).
- We can do the matrix inversion, WITHOUT inversion! (How is that possible?!)

15-826 (c) C. Faloutsos, 2008 60

CMU SCS

Even more details

- Q3: Can we estimate \mathbf{a} incrementally?
- A3: Yes, with the brilliant, classic method of ‘Recursive Least Squares’ (RLS) (see, e.g., [Yi+00], for details).
- We can do the matrix inversion, **WITHOUT** inversion! (How is that possible?!)
- A: our matrix has special form: $(\mathbf{X}^T \mathbf{X})$

15-826 (c) C. Faloutsos, 2008 61

CMU SCS

More details

At the $N+1$ time tick:

The diagram shows a blue hatched rectangle representing a matrix. The top width is labeled w . The left height is labeled N . A vertical double-headed arrow on the left is labeled \mathbf{X}_{N+1} . A horizontal double-headed arrow at the bottom is labeled \mathbf{X}_N . A red shaded horizontal bar at the bottom of the matrix is labeled \mathbf{X}_{N+1} .

15-826 (c) C. Faloutsos, 2008 62

CMU SCS

More details

- Let $\mathbf{G}_N = (\mathbf{X}_N^T \times \mathbf{X}_N)^{-1}$ (“gain matrix”)
- \mathbf{G}_{N+1} can be computed recursively from \mathbf{G}_N

The diagram shows a blue hatched square representing a matrix. The top width is labeled w . The right height is labeled w . The label \mathbf{G}_N is placed to the left of the square.

15-826 (c) C. Faloutsos, 2008 63

CMU SCS

EVEN more details:

$$G_{N+1} = G_N - [c]^{-1} \times [G_N \times x_{N+1}^T] \times x_{N+1} \times G_N$$

\swarrow
1 x w row vector

$$c = [1 + x_{N+1} \times G_N \times x_{N+1}^T]$$

Let's elaborate
(VERY IMPORTANT, VERY VALUABLE!)

15-826 (c) C. Faloutsos, 2008 64

CMU SCS

EVEN more details:

$$a = [X_{N+1}^T \times X_{N+1}]^{-1} \times [X_{N+1}^T \times y_{N+1}]$$

15-826 (c) C. Faloutsos, 2008 65

CMU SCS

EVEN more details:

$$a = [X_{N+1}^T \times X_{N+1}]^{-1} \times [X_{N+1}^T \times y_{N+1}]$$

$[w \times 1]$ $[(N+1) \times w]$ $[(N+1) \times 1]$
 $[w \times (N+1)]$ $[w \times (N+1)]$

15-826 (c) C. Faloutsos, 2008 66

CMU SCS

EVEN more details:

$$a = \underbrace{[X_{N+1}^T \times X_{N+1}]^{-1}}_{[(N+1) \times w]} \times \underbrace{[X_{N+1}^T \times y_{N+1}]}_{[w \times (N+1)]}$$

15-826 (c) C. Faloutsos, 2008 67

CMU SCS

EVEN more details:

$$a = [X_{N+1}^T \times X_{N+1}]^{-1} \times [X_{N+1}^T \times y_{N+1}]$$

'gain matrix' $G_{N+1} \equiv [X_{N+1}^T \times X_{N+1}]^{-1}$ 1 x w row vector

$$G_{N+1} = G_N - [c]^{-1} \times [G_N \times x_{N+1}^T] \times x_{N+1} \times G_N$$

$$c = [1 + x_{N+1} \times G_N \times x_{N+1}^T]$$

15-826 (c) C. Faloutsos, 2008 68

CMU SCS

EVEN more details:

$$G_{N+1} = G_N - [c]^{-1} \times [G_N \times x_{N+1}^T] \times x_{N+1} \times G_N$$

$$c = [1 + x_{N+1} \times G_N \times x_{N+1}^T]$$

15-826 (c) C. Faloutsos, 2008 69

CMU SCS

EVEN more details:

$$G_{N+1} = G_N - [c]^{-1} \times [G_N \times x_{N+1}^T] \times x_{N+1} \times G_N$$

SCALAR! $c = [1 + x_{N+1} \times G_N \times x_{N+1}^T]$

15-826 (c) C. Faloutsos, 2008 70

CMU SCS

Altogether:

$$a = [X_{N+1}^T \times X_{N+1}]^{-1} \times [X_{N+1}^T \times y_{N+1}]$$

$$G_{N+1} \equiv [X_{N+1}^T \times X_{N+1}]^{-1}$$

$$G_{N+1} = G_N - [c]^{-1} \times [G_N \times x_{N+1}^T] \times x_{N+1} \times G_N$$

$$c = [1 + x_{N+1} \times G_N \times x_{N+1}^T]$$

15-826 (c) C. Faloutsos, 2008 71

CMU SCS

Altogether:

$$G_0 \equiv \delta I$$

where
 I: w x w identity matrix
 δ: a large positive number

15-826 (c) C. Faloutsos, 2008 72

CMU SCS

Comparison:

- **Straightforward Least Squares**
 - Needs huge matrix (**growing** in size) $O(N \times w)$
 - Costly matrix operation $O(N \times w^2)$
- **Recursive LS**
 - Need much smaller, fixed size matrix $O(w \times w)$
 - Fast, incremental computation $O(1 \times w^2)$
 - **no matrix inversion**

$N = 10^6, \quad w = 1-100$

15-826 (c) C. Faloutsos, 2008 73

CMU SCS

Pictorially:

- **Given:**

15-826 (c) C. Faloutsos, 2008 74

CMU SCS

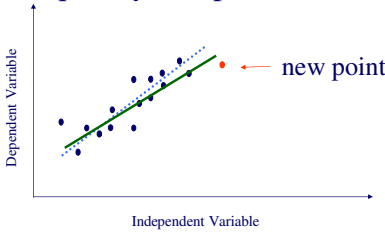
Pictorially:

15-826 (c) C. Faloutsos, 2008 75

CMU SCS

Pictorially:

RLS: quickly compute new best fit



Dependent Variable

Independent Variable

new point

15-826 (c) C. Faloutsos, 2008 76

CMU SCS

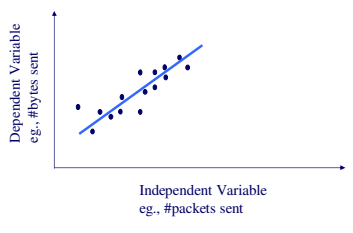
Even more details

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

15-826 (c) C. Faloutsos, 2008 77

CMU SCS

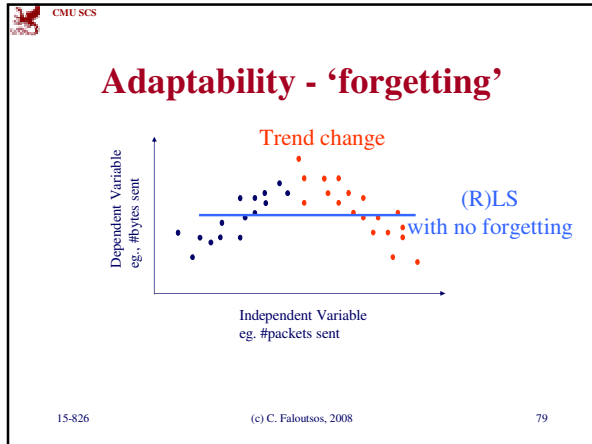
Adaptability - ‘forgetting’

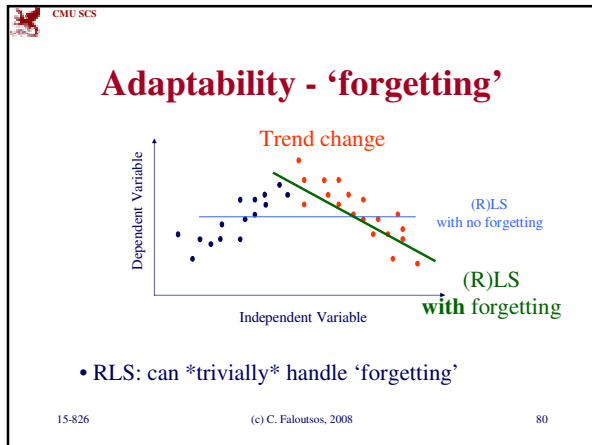


Dependent Variable
eg., #bytes sent

Independent Variable
eg., #packets sent

15-826 (c) C. Faloutsos, 2008 78





CMU SCS

How to choose 'w'?

- goal: capture arbitrary periodicities
- with NO human intervention
- on a semi-infinite stream

15-826 (c) C. Faloutsos, 2008 81

CMU SCS

Reference

[Papadimitriou+ vldb2003] Spiros Papadimitriou, Anthony Brockwell and Christos Faloutsos *Adaptive, Hands-Off Stream Mining* VLDB 2003, Berlin, Germany, Sept. 2003

15-826 (c) C. Faloutsos, 2008 82

CMU SCS

Answer:

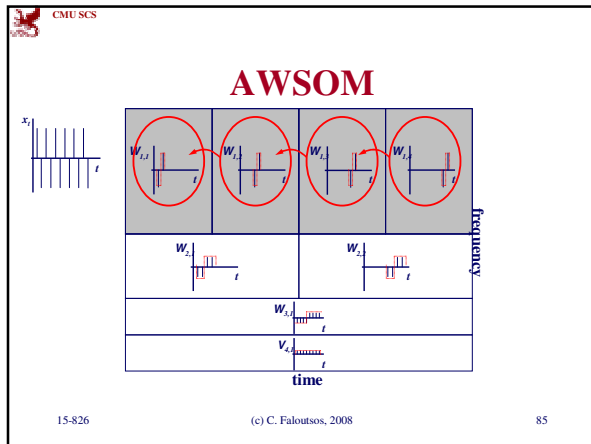
- ‘AWSOM’ (Arbitrary Window Stream fOrecasting Method) [Papadimitriou+, vldb2003]
- idea: do AR on each wavelet level
- in detail:

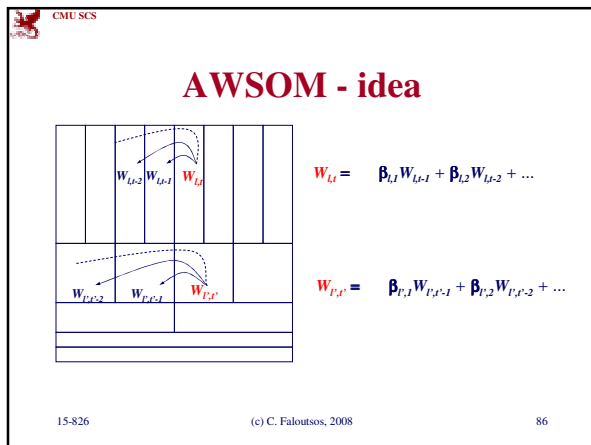
15-826 (c) C. Faloutsos, 2008 83

CMU SCS

AWSOM

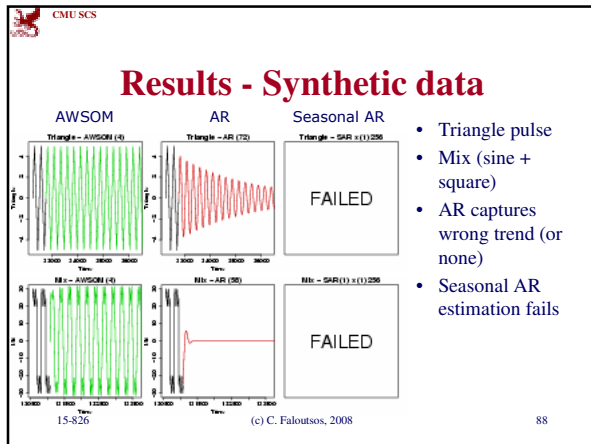
15-826 (c) C. Faloutsos, 2008 84

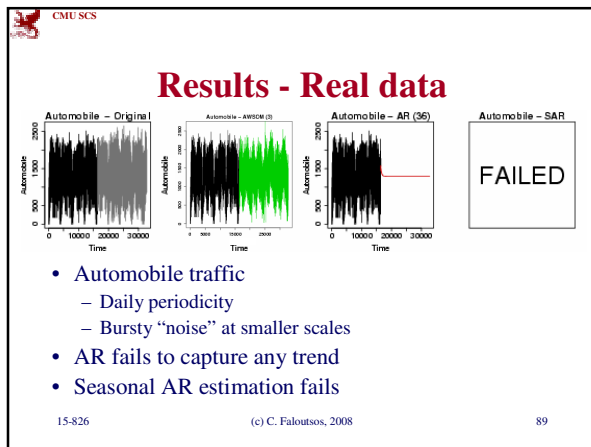


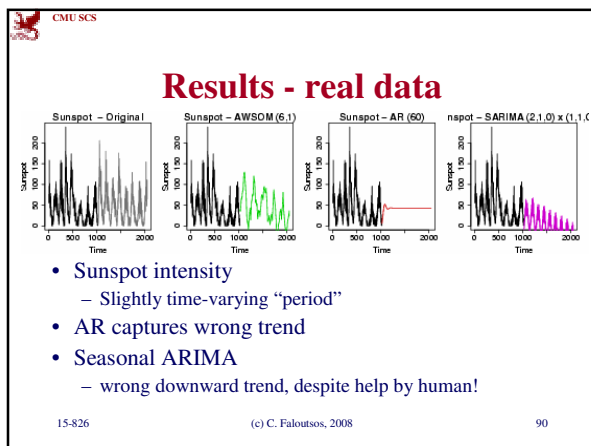


More details...

- Update of wavelet coefficients (incremental)
- Update of linear models (incremental; RLS)
- Feature selection (single-pass)
 - Not all correlations are significant
 - Throw away the insignificant ones (“noise”)







CMU SCS Skip

Complexity

- Model update
 - Space: $O(\lg N + mk^2) \approx O(\lg N)$
 - Time: $O(k^2) \approx O(1)$
- Where
 - N : number of points (so far)
 - k : number of regression coefficients; fixed
 - m : number of linear models; $O(\lg N)$

15-826 (c) C. Faloutsos, 2008 91

CMU SCS

Outline

- Motivation
- ...
- Linear Forecasting
 - Auto-regression: Least Squares; RLS
 - ➔ - Co-evolving time sequences
 - Examples
 - Conclusions

15-826 (c) C. Faloutsos, 2008 92

CMU SCS

Co-Evolving Time Sequences

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

Time Tick	sent	lost	repeated
1	40	20	25
2	55	25	20
3	70	35	30
4	45	20	35
5	55	25	20
6	60	30	25
7	85	40	35
8	75	35	40
9	60	30	35
10	50	25	30
11	??	??	??

15-826 (c) C. Faloutsos, 2008 93

CMU SCS

Solution:

Q: what should we do?

15-826 (c) C. Faloutsos, 2008 94

CMU SCS

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])

15-826 (c) C. Faloutsos, 2008 95

CMU SCS

Forecasting - Outline

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

15-826 (c) C. Faloutsos, 2008 96

CMU SCS

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)

15-826 (c) C. Faloutsos, 2008 97

CMU SCS

Accuracy - "Modem"

The chart displays RMSE values for 14 different modems. For each modem, three bars are shown: AR (green), yesterday (yellow), and MUSCLES (red). MUSCLES consistently shows the lowest RMSE across all modems, indicating superior accuracy.

Modem	AR	yesterday	MUSCLES
1	1.8	1.5	0.8
2	1.5	1.2	0.5
3	1.8	1.5	0.8
4	2.5	2.2	1.2
5	2.2	1.8	1.0
6	2.8	2.5	1.5
7	2.5	2.2	1.2
8	2.2	1.8	1.0
9	2.5	2.2	1.2
10	2.2	1.8	1.0
11	2.5	2.2	1.2
12	2.8	2.5	1.5
13	2.5	2.2	1.2
14	3.8	3.5	2.5

MUSCLES outperforms AR & "yesterday"

15-826 (c) C. Faloutsos, 2008 98

CMU SCS

Accuracy - "Internet"

The chart displays RMSE values for 15 different internet streams. For each stream, three bars are shown: AR (green), yesterday (yellow), and MUSCLES (red). MUSCLES consistently shows the lowest RMSE across all streams, indicating superior accuracy.

Streams	AR	yesterday	MUSCLES
1	0.8	0.7	0.4
2	0.7	0.6	0.3
3	0.7	0.6	0.3
4	0.7	0.6	0.3
5	0.7	0.6	0.3
6	0.7	0.6	0.3
7	0.8	0.7	0.4
8	0.7	0.6	0.3
9	0.7	0.6	0.3
10	0.5	0.4	0.2
11	0.5	0.4	0.2
12	0.5	0.4	0.2
13	1.3	1.2	0.8
14	1.2	1.1	0.7
15	1.2	1.1	0.7

MUSCLES consistently outperforms AR & "yesterday"

15-826 (c) C. Faloutsos, 2008 99

CMU SCS

Linear forecasting - Outline

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

15-826 (c) C. Faloutsos, 2008 100

CMU SCS

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]
- very recently: AWSOM (no human intervention)

15-826 (c) C. Faloutsos, 2008 101

CMU SCS

Resources: software and urls

- MUSCLES: Prof. Byoung-Kee Yi:
<http://www.postech.ac.kr/~bkyi/>
 or christos@cs.cmu.edu
- free-ware: 'R' for stat. analysis (clone of Splus)
<http://cran.r-project.org/>

15-826 (c) C. Faloutsos, 2008 102

CMU SCS

Books

- 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.)
- Brockwell, P. J. and R. A. Davis (1987). *Time Series: Theory and Methods*. New York, Springer Verlag.

15-826 (c) C. Faloutsos, 2008 103

CMU SCS

Additional Reading

- [Papadimitriou+ vldb2003] Spiros Papadimitriou, Anthony Brockwell and Christos Faloutsos *Adaptive, Hands-Off Stream Mining* VLDB 2003, Berlin, Germany, Sept. 2003
- [Yi+00] Byoung-Kee Yi et al.: *Online Data Mining for Co-Evolving Time Sequences*, ICDE 2000. (Describes MUSCLES and Recursive Least Squares)

15-826 (c) C. Faloutsos, 2008 104

CMU SCS

Outline

- Motivation
- Similarity search and distance functions
- Linear Forecasting
- ➔ **Bursty traffic - fractals and multifractals**
- Non-linear forecasting
- Conclusions

15-826 (c) C. Faloutsos, 2008 105

CMU SCS

Bursty traffic and multifractals

15-826 (c) C. Faloutsos, 2008 106

CMU SCS

Outline

- Motivation
- ...
- Linear Forecasting
- Bursty traffic - fractals and multifractals
 - ➔ - Problem
 - Main idea (80/20, Hurst exponent)
 - Results

15-826 (c) C. Faloutsos, 2008 107

CMU SCS

Reference:

[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.

Full thesis: CMU-CS-05-185
Performance Modeling of Storage Devices using Machine Learning Mengzhi Wang, Ph.D. Thesis
[Abstract](#), [.ps.gz](#), [.pdf](#)

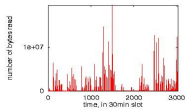
15-826 (c) C. Faloutsos, 2008 108

CMU SCS

Recall: Problem #1:

Goal: given a signal (eg., #bytes over time)
 Find: patterns, periodicities, and/or compress

#bytes



Bytes per 30'
(packets per day;
earthquakes per year)

time

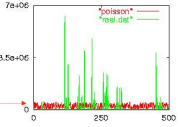
15-826 (c) C. Faloutsos, 2008 109

CMU SCS

Problem #1

- model bursty traffic
- generate realistic traces
- (Poisson does not work)

bytes



time

15-826 (c) C. Faloutsos, 2008 110

CMU SCS

Motivation

- predict queue length distributions (e.g., to give probabilistic guarantees)
- “learn” traffic, for buffering, prefetching, ‘active disks’, web servers

15-826 (c) C. Faloutsos, 2008 111

CMU SCS

Q: any 'pattern'?

- Not Poisson
- spike; silence; more spikes; more silence...
- any rules?

bytes

number of bytes read

time

15-826 (c) C. Faloutsos, 2008 112

CMU SCS

Solution: self-similarity

bytes

number of bytes read

time

bytes

number of bytes read

time

15-826 (c) C. Faloutsos, 2008 113

CMU SCS

But:

- Q1: How to generate realistic traces; extrapolate; give guarantees?
- Q2: How to estimate the model parameters?

15-826 (c) C. Faloutsos, 2008 114

CMU SCS

Outline

- Motivation
- ...
- Linear Forecasting
- Bursty traffic - fractals and multifractals
 - Problem
 - ➔ – Main idea (80/20, Hurst exponent)
 - Results

15-826 (c) C. Faloutsos, 2008 115

CMU SCS

Approach

- Q1: How to generate a sequence, that is
 - bursty
 - self-similar
 - and has similar queue length distributions

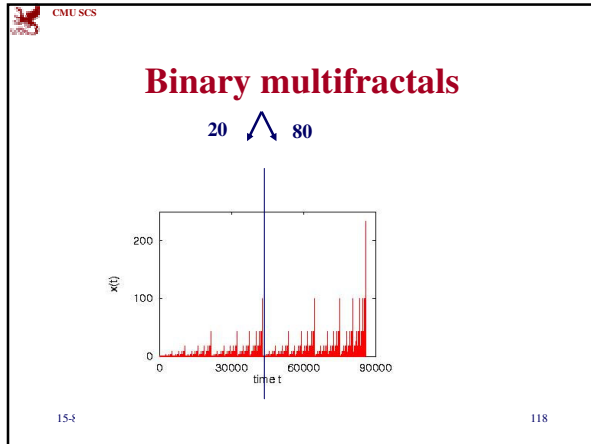
15-826 (c) C. Faloutsos, 2008 116

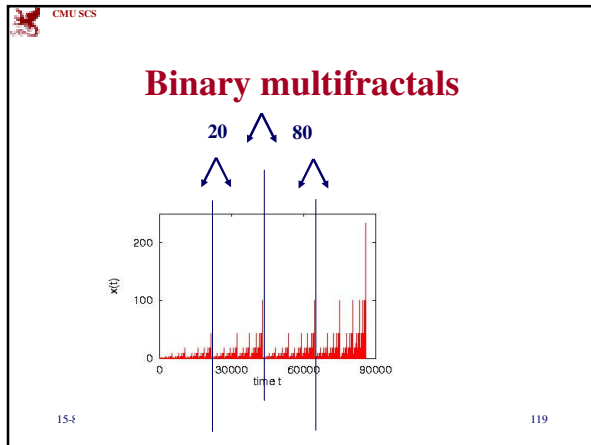
CMU SCS

Approach

- A: 'binomial multifractal' [Wang+02]
- ~ 80-20 'law':
 - 80% of bytes/queries etc on first half
 - repeat recursively
- *b*: bias factor (eg., 80%)

15-826 (c) C. Faloutsos, 2008 117





CMU SCS

Parameter estimation

- Q2: How to estimate the bias factor b ?

15-826 (c) C. Faloutsos, 2008 120

CMU SCS

Parameter estimation

- Q2: How to estimate the bias factor b ?
- A: MANY ways [Crovella+96]
 - Hurst exponent
 - variance plot
 - even DFT amplitude spectrum! ('periodogram')
 - More robust: 'entropy plot' [Wang+02]

15-826 (c) C. Faloutsos, 2008 121

CMU SCS

Entropy plot

- Rationale:
 - burstiness: inverse of uniformity
 - entropy measures uniformity of a distribution
 - find entropy at several granularities, to see whether/how our distribution is close to uniform.

15-826 (c) C. Faloutsos, 2008 122

CMU SCS

Entropy plot

p1 p2

% of bytes here

- Entropy $E(n)$ after n levels of splits
- $n=1: E(1) = -p1 \log_2(p1) - p2 \log_2(p2)$

15-826 (c) C. Faloutsos, 2008 123

CMU SCS

Entropy plot

- Entropy $E(n)$ after n levels of splits
- $n=1: E(1) = -p_1 \log(p_1) - p_2 \log(p_2)$
- $n=2: E(2) = -\sum_i p_{2,i} \log_2(p_{2,i})$

15-826 (c) C. Faloutsos, 2008 124

CMU SCS

Real traffic

Entropy $E(n)$

- Has linear entropy plot (\rightarrow self-similar)

of levels (n)

15-826 (c) C. Faloutsos, 2008 125

CMU SCS

Observation - intuition:

Entropy $E(n)$

of levels (n)

intuition: slope =
intrinsic dimensionality =
info-bits per coordinate-bit

- unif. Dataset: slope = ?
- multi-point: slope = ?

15-826 (c) C. Faloutsos, 2008 126

CMU SCS

Observation - intuition:

Entropy $E(n)$

intuition: slope =
intrinsic dimensionality =
info-bits per coordinate-bit

- unif. Dataset: slope = 1
- multi-point: slope = 0


of levels (n)

15-826 (c) C. Faloutsos, 2008 127

CMU SCS

Entropy plot - Intuition

- Slope ~ intrinsic dimensionality (in fact, 'Information fractal dimension')
- = info bit per coordinate bit - eg

Dim = 1 

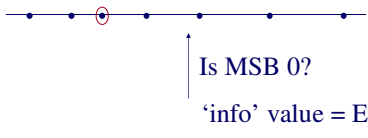
Pick a point;
reveal its coordinate bit-by-bit -
how much info is each bit worth to me?

15-826 (c) C. Faloutsos, 2008 128

CMU SCS

Entropy plot

- Slope ~ intrinsic dimensionality (in fact, 'Information fractal dimension')
- = info bit per coordinate bit - eg

Dim = 1 

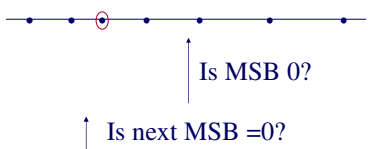
Is MSB 0?
'info' value = $E(1)$: 1 bit

15-826 (c) C. Faloutsos, 2008 129

CMU SCS

Entropy plot

- Slope ~ intrinsic dimensionality (in fact, 'Information fractal dimension')
- = info bit per coordinate bit - eg

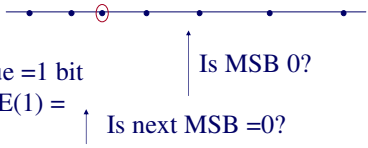
Dim = 1 

15-826 (c) C. Faloutsos, 2008 130

CMU SCS

Entropy plot

- Slope ~ intrinsic dimensionality (in fact, 'Information fractal dimension')
- = info bit per coordinate bit - eg

Dim = 1 


Info value = 1 bit
= $E(2) - E(1)$ = slope!

15-826 (c) C. Faloutsos, 2008 131

CMU SCS

Entropy plot

- Repeat, for all points at same position:


Dim=0 

15-826 (c) C. Faloutsos, 2008 132

CMU SCS

Entropy plot

- Repeat, for all points at same position:
- we need 0 bits of info, to determine position
- -> slope = 0 = intrinsic dimensionality


Dim=0 


15-826 (c) C. Faloutsos, 2008 133


CMU SCS

Entropy plot

- Real (and 80-20) datasets can be in-between: bursts, gaps, smaller bursts, smaller gaps, at every scale

Dim = 1 

Dim=0 

$0 < \text{Dim} < 1$ 

15-826 (c) C. Faloutsos, 2008 134

CMU SCS

(Fractals, again)

- What set of points could have behavior between point and line?

15-826 (c) C. Faloutsos, 2008 135

CMU SCS

Cantor dust

- Eliminate the middle third
- Recursively!

15-826 (c) C. Faloutsos, 2008 136

CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2008 137

CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2008 138

CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2008 139

CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2008 140

CMU SCS

Cantor dust

Dimensionality?
(no length; infinite # points!)
Answer: $\log_2 / \log_3 = 0.6$

15-826 (c) C. Faloutsos, 2008 141

CMU SCS

Some more entropy plots:

- Poisson vs real

Poisson: slope = ~1 -> uniformly distributed

15-826 (c) C. Faloutsos, 2008 142

CMU SCS

b-model

$E(n)$

- b-model traffic gives perfectly linear plot
- Lemma: its slope is $\text{slope} = -b \log_2 b - (1-b) \log_2 (1-b)$
- Fitting: do entropy plot; get slope; solve for b

15-826 (c) C. Faloutsos, 2008 143

CMU SCS

Outline

- Motivation
- ...
- Linear Forecasting
- Bursty traffic - fractals and multifractals
 - Problem
 - Main idea (80/20, Hurst exponent)
 - ➔ – Experiments - Results

15-826 (c) C. Faloutsos, 2008 144

CMU SCS

Experimental setup

- Disk traces (from HP [Wilkes 93])
- web traces from LBL
<http://repository.cs.vt.edu/lbl-conn-7.tar.Z>

15-826 (c) C. Faloutsos, 2008 145

CMU SCS

Model validation

- Linear entropy plots

(a) Disk Traces

(b) Web Traces

Bias factors b : 0.6-0.8
 smallest b / smoothest: nntp traffic

15-826 (c) C. Faloutsos, 2008 146

CMU SCS

Web traffic - results

- LBL, NCDF of queue lengths (log-log scales)

Prob($>l$)

(a) lbl-all

(b) lbl-nntp

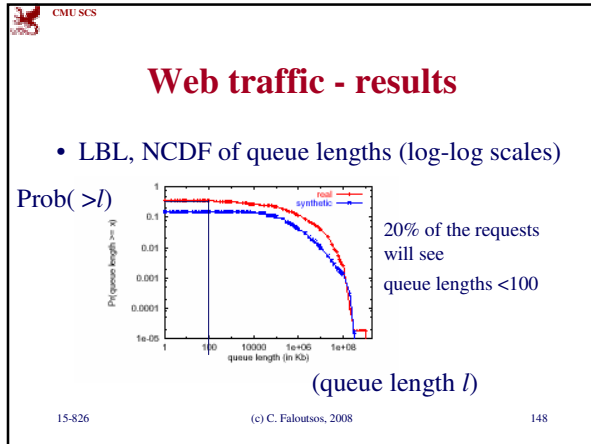
(c) lbl-smtp

(d) lbl-ftp

Queue length distribution

How to give guarantees? (queue length l)

15-826 (c) C. Faloutsos, 2008 147



Conclusions

- Multifractals (80/20, 'b-model', Multiplicative Wavelet Model (MWM)) for analysis and synthesis of bursty traffic

15-826 (c) C. Faloutsos, 2008 149

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!)

15-826 (c) C. Faloutsos, 2008 150

CMU SCS

Further reading:

- Crovella, M. and A. Bestavros (1996). Self-Similarity in World Wide Web Traffic, Evidence and Possible Causes. *Sigmetrics*.
- [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.

15-826 (c) C. Faloutsos, 2008 151

CMU SCS

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.

Entropy plots

15-826 (c) C. Faloutsos, 2008 152

CMU SCS

Outline

- Motivation
- ...
- Linear Forecasting
- Bursty traffic - fractals and multifractals
- ➔ Non-linear forecasting
- Conclusions

15-826 (c) C. Faloutsos, 2008 153

CMU SCS

Chaos and non-linear forecasting

15-826 (c) C. Faloutsos, 2008 154

CMU SCS

Reference:

[Deepay Chakrabarti and Christos Faloutsos
*F4: Large-Scale Automated Forecasting
using Fractals* CIKM 2002, Washington
DC, Nov. 2002.]

15-826 (c) C. Faloutsos, 2008 155

CMU SCS

Detailed Outline

- Non-linear forecasting
 - Problem
 - Idea
 - How-to
 - Experiments
 - Conclusions

15-826 (c) C. Faloutsos, 2008 156

CMU SCS

Recall: Problem #1

Value

Time

Given a time series $\{x_t\}$, predict its future course, that is, x_{t+1}, x_{t+2}, \dots

15-826 (c) C. Faloutsos, 2008 157

CMU SCS

How to forecast?

- ARIMA - but: linearity assumption
- ANSWER: 'Delayed Coordinate Embedding' = Lag Plots [Sauer92]

15-826 (c) C. Faloutsos, 2008 158

CMU SCS

General Intuition (Lag Plot)

Interpolate these...

To get the final prediction

4-NN

New Point

x_t

x_{t-1}

Lag = 1,
k = 4 NN

15-826 (c) C. Faloutsos, 2008 159

CMU SCS

Questions:

- Q1: How to choose lag L ?
- Q2: How to choose k (the # of NN)?
- Q3: How to interpolate?
- Q4: why should this work at all?

15-826 (c) C. Faloutsos, 2008 160

CMU SCS

Q1: Choosing lag L

- Manually (16, in award winning system by [Sauer94])

15-826 (c) C. Faloutsos, 2008 161

CMU SCS

Q2: Choosing number of neighbors k

- Manually (typically ~ 1-10)

15-826 (c) C. Faloutsos, 2008 162

CMU SCS

Q3: How to interpolate?

How do we interpolate between the k nearest neighbors?

A3.1: Average

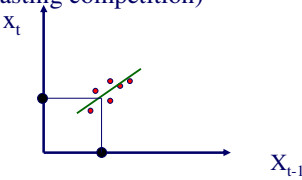
A3.2: Weighted average (weights drop with distance - how?)

15-826 (c) C. Faloutsos, 2008 163

CMU SCS

Q3: How to interpolate?

A3.3: Using SVD - seems to perform best ([Sauer94] - first place in the Santa Fe forecasting competition)



15-826 (c) C. Faloutsos, 2008 164

CMU SCS

Q4: Any theory behind it?

A4: YES!

15-826 (c) C. Faloutsos, 2008 165

CMU SCS

Theoretical foundation

- Based on the “Takens’ Theorem” [Takens81]
- which says that **long enough delay vectors can do prediction**, even if there are unobserved variables in the dynamical system (= diff. equations)

15-826 (c) C. Faloutsos, 2008 166

CMU SCS

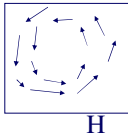
Theoretical foundation

Skip

Example: Lotka-Volterra equations

$$\begin{aligned} dH/dt &= r H - a H*P \\ dP/dt &= b H*P - m P \end{aligned}$$

H is count of prey (e.g., hare)
P is count of predators (e.g., lynx)



Suppose only P(t) is observed (t=1, 2, ...).

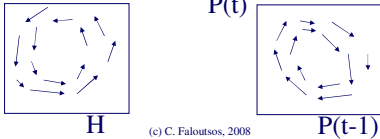
15-826 (c) C. Faloutsos, 2008 167

CMU SCS

Theoretical foundation

Skip

- But the delay vector space is a faithful reconstruction of the internal system state
- So prediction in **delay vector space** is as good as prediction in **state space**



15-826 (c) C. Faloutsos, 2008 168

CMU SCS

Detailed Outline

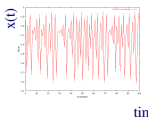
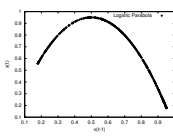
- Non-linear forecasting
 - Problem
 - Idea
 - How-to
 - ➔ – Experiments
 - Conclusions

15-826 (c) C. Faloutsos, 2008 169

CMU SCS

Datasets

Logistic Parabola:
 $x_t = ax_{t-1}(1-x_{t-1}) + \text{noise}$
 Models population of flies [R. May/1976]

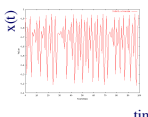
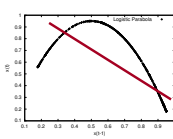
Lag-plot

15-826 (c) C. Faloutsos, 2008 170

CMU SCS

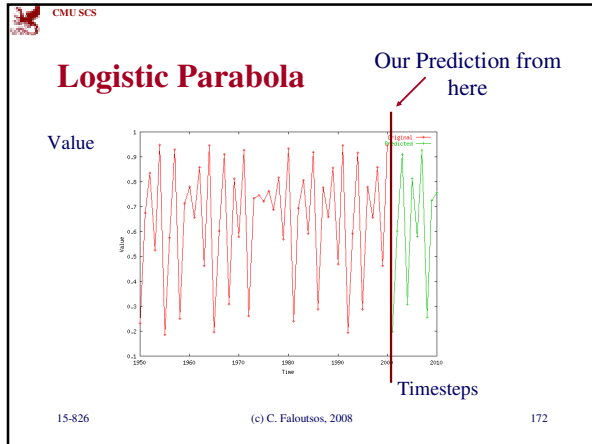
Datasets

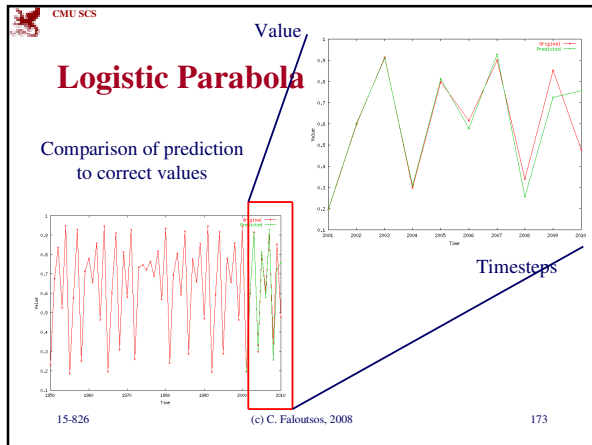
Logistic Parabola:
 $x_t = ax_{t-1}(1-x_{t-1}) + \text{noise}$
 Models population of flies [R. May/1976]

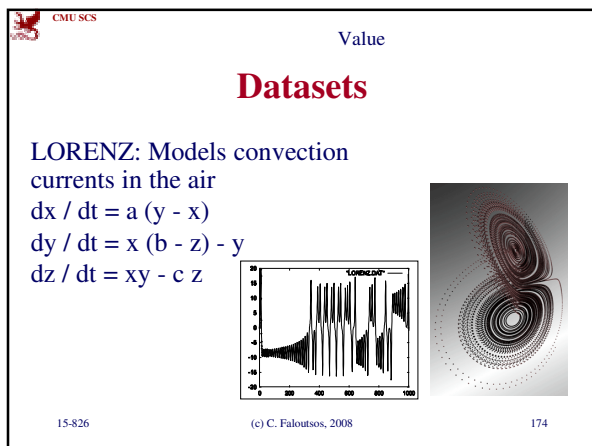



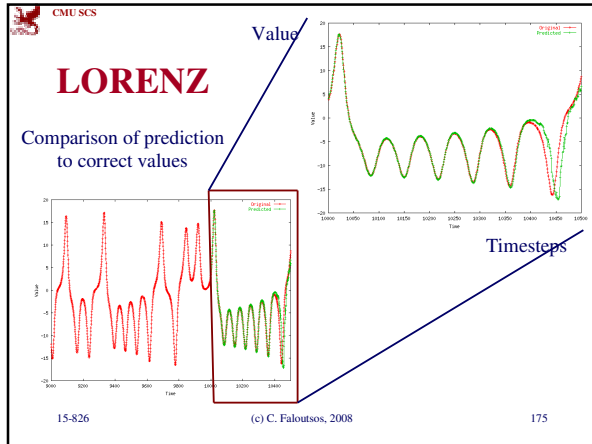
Lag-plot
ARIMA: fails

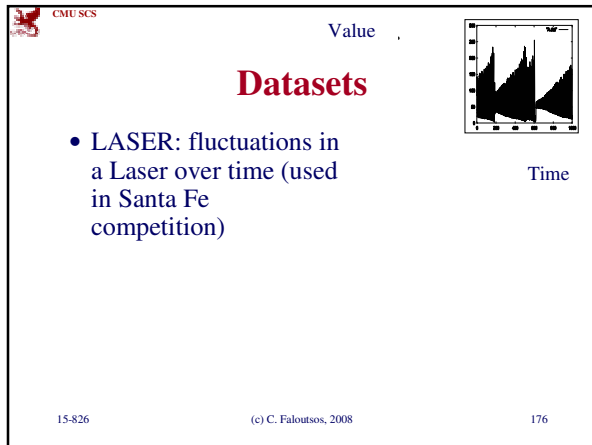
15-826 (c) C. Faloutsos, 2008 171

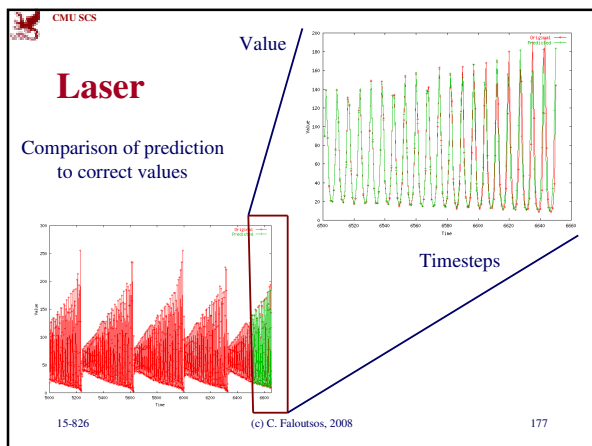












CMU SCS

Conclusions

- Lag plots for non-linear forecasting (Takens' theorem)
- suitable for 'chaotic' signals

15-826 (c) C. Faloutsos, 2008 178

CMU SCS

References

- Deepay Chakrabarti and Christos Faloutsos *F4: Large-Scale Automated Forecasting using Fractals* CIKM 2002, Washington DC, Nov. 2002.
- Sauer, T. (1994). *Time series prediction using delay coordinate embedding*. (in book by Weigend and Gershenfeld, below) Addison-Wesley.
- Takens, F. (1981). *Detecting strange attractors in fluid turbulence*. Dynamical Systems and Turbulence. Berlin: Springer-Verlag.

15-826 (c) C. Faloutsos, 2008 179

CMU SCS

References

- Weigend, A. S. and N. A. Gershenfeld (1994). *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison Wesley. (Excellent collection of papers on chaotic/non-linear forecasting, describing the algorithms behind the winners of the Santa Fe competition.)

15-826 (c) C. Faloutsos, 2008 180

CMU SCS

Overall conclusions

- Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**

15-826 (c) C. Faloutsos, 2008 181

CMU SCS

Overall conclusions

- Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- Signal processing: **DWT** is a powerful tool

15-826 (c) C. Faloutsos, 2008 182

CMU SCS

Overall conclusions

- Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- Signal processing: **DWT** is a powerful tool
- Linear Forecasting: **AR** (Box-Jenkins) methodology; **AWSOM**

15-826 (c) C. Faloutsos, 2008 183

CMU SCS

Overall conclusions

- Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- Signal processing: **DWT** is a powerful tool
- Linear Forecasting: **AR** (Box-Jenkins) methodology; **AWSOM**
- Bursty traffic: **multifractals** (80-20 'law')

15-826 (c) C. Faloutsos, 2008 184

CMU SCS

Overall conclusions

- Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- Signal processing: **DWT** is a powerful tool
- Linear Forecasting: **AR** (Box-Jenkins) methodology; **AWSOM**
- Bursty traffic: **multifractals** (80-20 'law')
- Non-linear forecasting: **lag-plots** (Takens)

15-826 (c) C. Faloutsos, 2008 185
