CMU SCS

15-826: Multimedia Databases and Data Mining

Time series mining and forecasting

Christos Faloutsos

CMU SCS

Thanks


Deepay Chakrabarti (CMU)


Prof. Dimitris Gunopulos (UCR)


Spiros Papadimitriou (CMU)


Mengzhi Wang (CMU)


Prof. Byoung-Kee Yi (Pohang U.)

CMU SCS

Outline

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

CMU SCS

Problem definition

- Given: one or more sequences
  x_1, x_2, … , x_t, …
  (y_1, y_2, … , y_t, …
  …)
- Find
  - similar sequences; forecasts
  - patterns; clusters; outliers

15-826 (c) C. Faloutsos, 2007 4

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, 2007 5

CMU SCS

Motivation - Applications (cont'd)

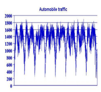
- 'Smart house'
  - sensors monitor temperature, humidity, air quality
- video surveillance

15-826 (c) C. Faloutsos, 2007 6

CMU SCS

Motivation - Applications (cont'd)

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

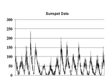


15-826 (c) C. Faloutsos, 2007 7

CMU SCS

Motivation - Applications (cont'd)

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



15-826 (c) C. Faloutsos, 2007 8

CMU SCS

Motivation - Applications (cont'd)

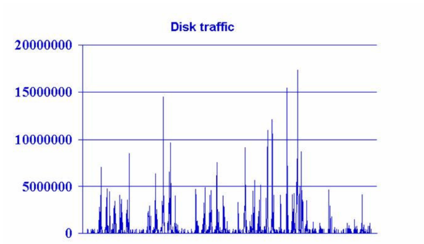
- Computer systems
  - 'Active Disks' (buffering, prefetching)
  - web servers (ditto)
  - network traffic monitoring
  - ...

15-826 (c) C. Faloutsos, 2007 9

CMU SCS

Stream Data: Disk accesses

#bytes



time

15-826 (c) C. Faloutsos, 2007 10

CMU SCS

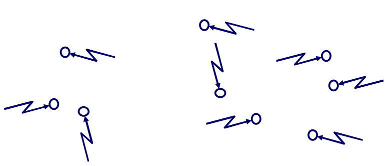
Settings & Applications

- One or more sensors, collecting time-series data

15-826 (c) C. Faloutsos, 2007 11

CMU SCS

Settings & Applications

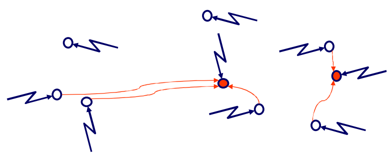


Each sensor collects data (x_1, x_2, …, x_t, …)

15-826 (c) C. Faloutsos, 2007 12

CMU SCS

Settings & Applications



Some sensors ‘report’ to others or to the central site

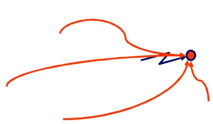
CMU SCS

Settings & Applications

Goal #1:
Finding patterns
in a single time sequence

CMU SCS

Settings & Applications



Goal #2:
Finding patterns
in many time
sequences

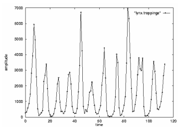
CMU SCS

Problem #1:

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

Find: patterns, periodicities, and/or compress

count



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

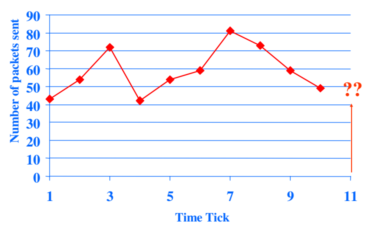
year

15-826 (c) C. Faloutsos, 2007 16

CMU SCS

Problem#2: Forecast

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

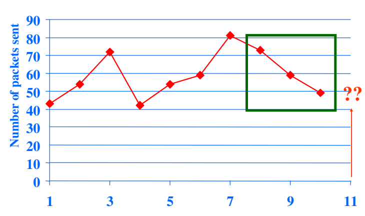


15-826 (c) C. Faloutsos, 2007 17

CMU SCS

Problem#2': Similarity search

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

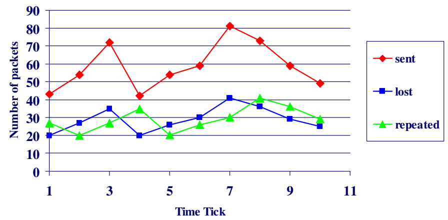


15-826 (c) C. Faloutsos, 2007 18

CMU SCS

Problem #3:

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



15-826 (c) C. Faloutsos, 2007 19

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, 2007 20

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, 2007 21

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)

15-826 (c) C. Faloutsos, 2007 22

CMU SCS

Outline

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

15-826 (c) C. Faloutsos, 2007 23

CMU SCS

Outline

- Motivation
- ➡ Similarity search and distance functions
  - Euclidean
  - Time-warping
- DSP
- ...

15-826 (c) C. Faloutsos, 2007 24

CMU SCS

Importance of distance functions

Subtle, but **absolutely necessary**:

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

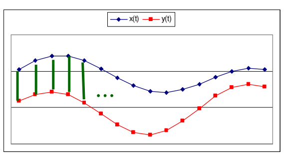
Two major families

- Euclidean and Lp norms
- Time warping and variations

15-826 (c) C. Faloutsos, 2007 25

CMU SCS

Euclidean and Lp



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

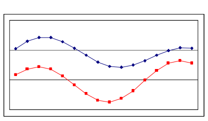
$$L_p(\vec{x}, \vec{y}) = \sum_{i=1}^{n} | x_i - y_i |^p$$

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

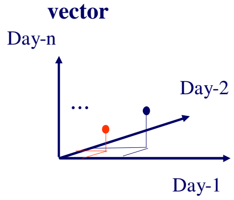
15-826 (c) C. Faloutsos, 2007 26

CMU SCS

Observation #1



- **Time sequence -> n-d vector**

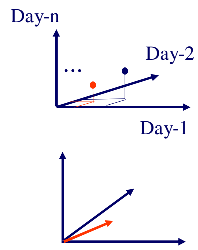


15-826 (c) C. Faloutsos, 2007 27

CMU SCS

Observation #2

Euclidean distance is closely related to
- cosine similarity
- dot product
- 'cross-correlation' function



15-826 (c) C. Faloutsos, 2007 28

CMU SCS

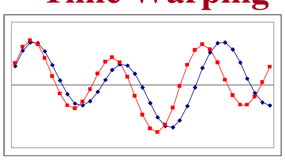
Time Warping

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

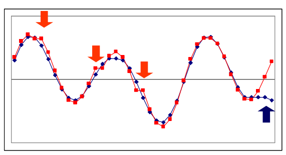
15-826 (c) C. Faloutsos, 2007 29

CMU SCS

Time Warping



'stutters':



15-826 (c) C. Faloutsos, 2007 30

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, 2007 31

CMU SCS

Time warping

Thus, with no penalty for stutter, for sequences

$x_1, x_2, \ldots, x_{i,;}$ $y_1, y_2, \ldots, y_j$

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

15-826 (c) C. Faloutsos, 2007 32

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, 2007 33

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]

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]

CMU SCS

Other Distance functions

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

CMU SCS

Conclusions

Prevailing distances:
- Euclidean and
- time-warping

15-826 (c) C. Faloutsos, 2007 37

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, 2007 38

CMU SCS

Linear Forecasting

15-826 (c) C. Faloutsos, 2007 39

CMU SCS

Forecasting

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

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

CMU SCS

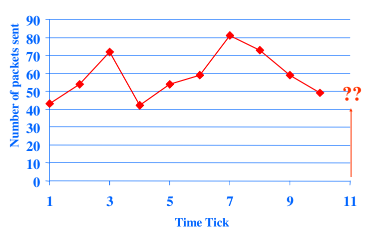
Outline

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

CMU SCS

Problem#2: Forecast

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



CMU SCS

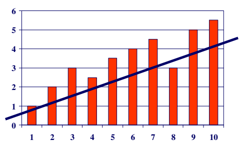
Forecasting: Preprocessing

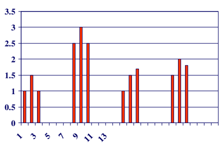
MANUALLY:

remove trends

spot periodicities

7 days





time

time

15-826 (c) C. Faloutsos, 2007 43

CMU SCS

Problem#2: Forecast

- Solution: try to express

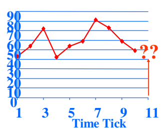
  x_t

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

  (up to a window of w)

Formally:

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



15-826 (c) C. Faloutsos, 2007 44

CMU SCS

(Problem: Back-cast; interpolate)

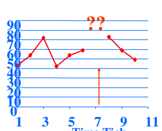
- Solution - interpolate: try to express

  x_t

  as a linear function of the past AND the future:

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

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

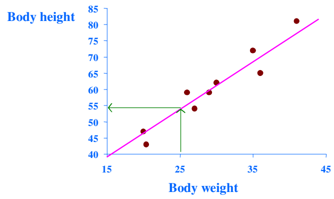


15-826 (c) C. Faloutsos, 2007 45

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)')

CMU SCS

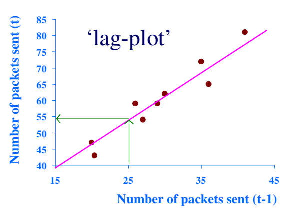
Linear Auto Regression:

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

CMU SCS

Linear Auto Regression:

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



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

CMU SCS

Outline

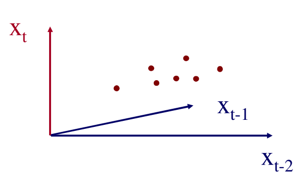
- Motivation
- ...
- Linear Forecasting
  - Auto-regression: **Least Squares; RLS**
  - Co-evolving time sequences
  - Examples
  - Conclusions

15-826 (c) C. Faloutsos, 2007 49

CMU SCS

More details:

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

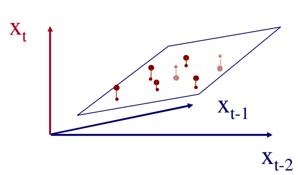


15-826 (c) C. Faloutsos, 2007 50

CMU SCS

More details:

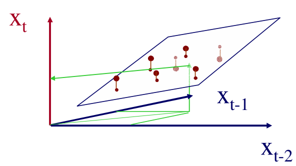
- Q1: Can it work with window *w*>1?
- A1: YES! (we'll fit a hyper-plane, then!)



15-826 (c) C. Faloutsos, 2007 51

More details:

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



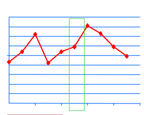
More details:

- 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:



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

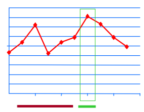
Ind-var1 Ind-var-w

time

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

CMU SCS

More details:



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

Ind-var1 Ind-var-w

time

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

15-826 (c) C. Faloutsos, 2007 55

CMU SCS

More details

- Q2: How to estimate $a_1, a_2, \ldots a_w = \mathbf{a}$?
- 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)
- **a** is the vector that minimizes the RMSE from **y**
- <**identical** math with 'query feedbacks'>

15-826 (c) C. Faloutsos, 2007 56

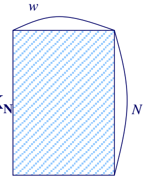
CMU SCS

More details

- Straightforward solution:

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

a : Regression Coeff. Vector
X : Sample Matrix



- Observations:
  - Sample matrix X grows over time
  - needs matrix inversion
  - **O**($N \times w^2$) computation
  - **O**($N \times w$) storage

15-826 (c) C. Faloutsos, 2007 57

CMU SCS

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).
- We can do the matrix inversion, WITHOUT inversion! (How is that possible?!)

CMU SCS

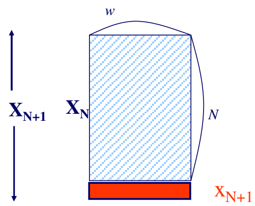
Even more details

- 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).
- We can do the matrix inversion, WITHOUT inversion! (How is that possible?!)
- A: our matrix has special form: (X^T X)

CMU SCS

More details

At the *N+1* time tick:

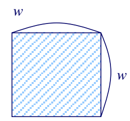


CMU SCS

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 $\mathbf{G}_N$

$\mathbf{G_N}$



15-826 (c) C. Faloutsos, 2007 61

CMU SCS

EVEN more details:

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

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, 2007 62

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, 2007 63

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 x 1] [(N+1) x w] [(N+1) x 1]

[w x (N+1)] [w x (N+1)]

15-826 (c) C. Faloutsos, 2007 64

CMU SCS

EVEN more details:

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

[(N+1) x w]

[w x (N+1)]

15-826 (c) C. Faloutsos, 2007 65

CMU SCS

EVEN more details:

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

'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, 2007 66

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, 2007 67

CMU SCS

EVEN more details:

1x1

1xw

wxw wxw wxw wx1 wxw

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

SCALAR!

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

15-826 (c) C. Faloutsos, 2007 68

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, 2007 69

CMU SCS

Altogether:

$$G_0 \equiv \delta I$$

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

CMU SCS

Comparison:

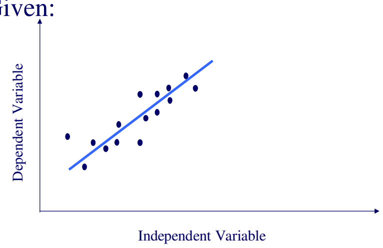
- Straightforward Least Squares
  - Needs huge matrix (**growing** in size) $O(N \times w)$
  - Costly matrix operation $O(N \times w^2)$
- Recursive LS
  - Need much smaller, fixed size matrix $O(w \times w)$
  - Fast, incremental computation $O(1 \times w^2)$
  - **no matrix inversion**

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

CMU SCS

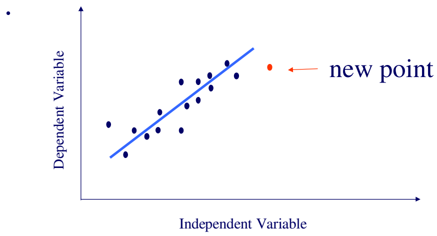
Pictorially:

- Given:



CMU SCS

Pictorially:

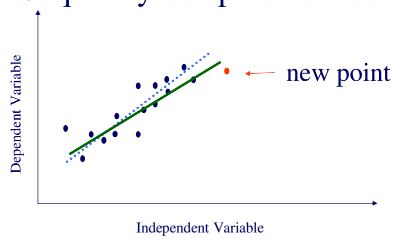


15-826 (c) C. Faloutsos, 2007 73

CMU SCS

Pictorially:

RLS: quickly compute new best fit



15-826 (c) C. Faloutsos, 2007 74

CMU SCS

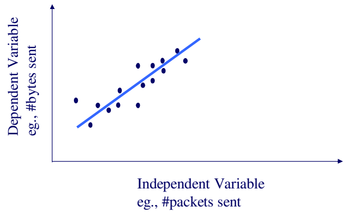
Even more details

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

15-826 (c) C. Faloutsos, 2007 75

CMU SCS

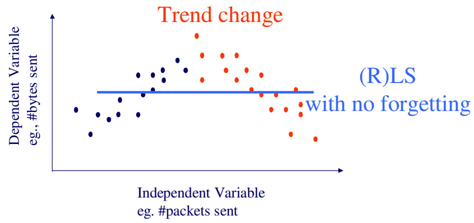
Adaptability - 'forgetting'



15-826 (c) C. Faloutsos, 2007 76

CMU SCS

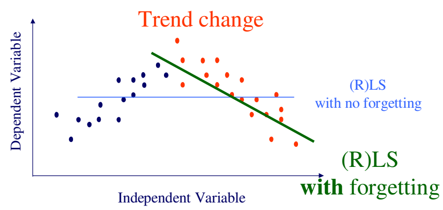
Adaptability - 'forgetting'



15-826 (c) C. Faloutsos, 2007 77

CMU SCS

Adaptability - 'forgetting'



- RLS: can *trivially* handle 'forgetting'

15-826 (c) C. Faloutsos, 2007 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, 2007 79

CMU SCS

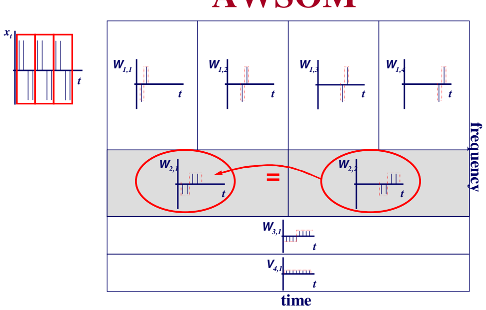
Answer:

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

15-826 (c) C. Faloutsos, 2007 80

CMU SCS

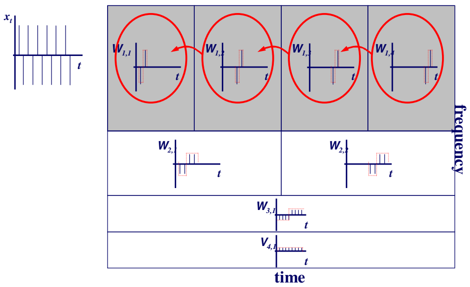
AWSOM



15-826 (c) C. Faloutsos, 2007 81

CMU SCS

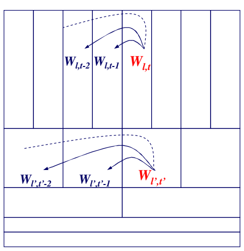
AWSOM



15-826 (c) C. Faloutsos, 2007 82

CMU SCS

AWSOM - idea



$W_{l,t} = \beta_{l,1} W_{l,t-1} + \beta_{l,2} W_{l,t-2} + \ldots$

$W_{l',t'} = \beta_{l',1} W_{l',t'-1} + \beta_{l',2} W_{l',t'-2} + \ldots$

15-826 (c) C. Faloutsos, 2007 83

CMU SCS

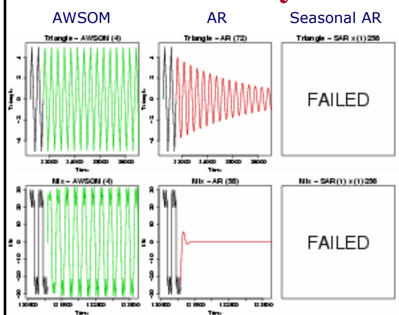
More details…

- 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")

15-826 (c) C. Faloutsos, 2007 84

CMU SCS

Results - Synthetic data

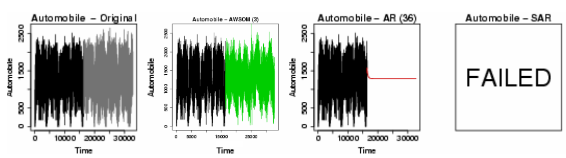


- Triangle pulse
- Mix (sine + square)
- AR captures wrong trend (or none)
- Seasonal AR estimation fails

15-826 (c) C. Faloutsos, 2007 85

CMU SCS

Results - Real data

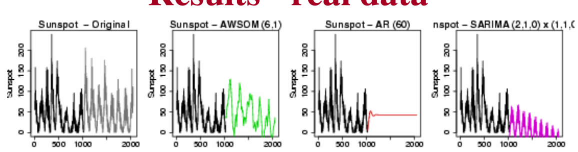


- Automobile traffic
  - Daily periodicity
  - Bursty "noise" at smaller scales
- AR fails to capture any trend
- Seasonal AR estimation fails

15-826 (c) C. Faloutsos, 2007 86

CMU SCS

Results - real data



- Sunspot intensity
  - Slightly time-varying "period"
- AR captures wrong trend
- Seasonal ARIMA
  - wrong downward trend, despite help by human!

15-826 (c) C. Faloutsos, 2007 87

CMU SCS

Skip

Complexity

- Model update

  Space: $O(lgN + mk^2) \approx O(lgN)$

  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; $O(lgN)$

15-826 (c) C. Faloutsos, 2007 88

CMU SCS

Outline

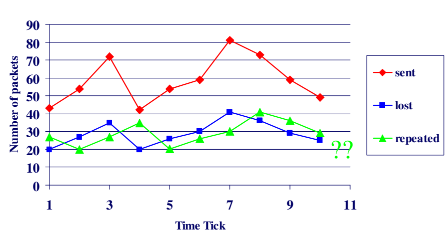
- Motivation
- ...
- Linear Forecasting
  - Auto-regression: Least Squares; RLS
  - Co-evolving time sequences
  - Examples
  - Conclusions

15-826 (c) C. Faloutsos, 2007 89

CMU SCS

Co-Evolving Time Sequences

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



15-826 (c) C. Faloutsos, 2007 90

CMU SCS

Solution:

Q: what should we do?

15-826 (c) C. Faloutsos, 2007 91

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, 2007 92

CMU SCS

Forecasting - Outline

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

15-826 (c) C. Faloutsos, 2007 93

CMU SCS

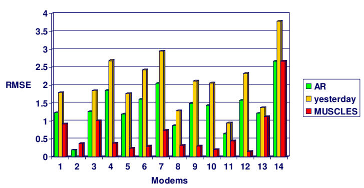
Examples - Experiments

- 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, 2007 94

CMU SCS

Accuracy - "Modem"

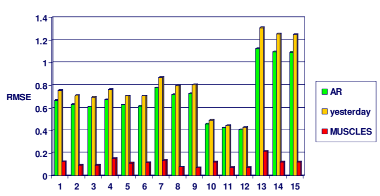


MUSCLES outperforms AR & "yesterday"

15-826 (c) C. Faloutsos, 2007 95

CMU SCS

Accuracy - "Internet"



MUSCLES consistently outperforms AR & "yesterday"

15-826 (c) C. Faloutsos, 2007 96

CMU SCS

B.II - Time Series Analysis - Outline

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

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)

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/**

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, 2007 100

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, 2007 101

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, 2007 102

CMU SCS

Bursty traffic and multifractals

CMU SCS

Outline

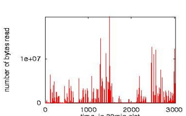
- Motivation
- ...
- Linear Forecasting
- Bursty traffic - fractals and multifractals
  - Problem
  - Main idea (80/20, Hurst exponent)
  - Results

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

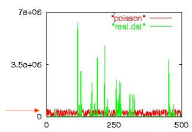
CMU SCS

Problem #1

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

bytes

Poisson



time

CMU SCS

Motivation

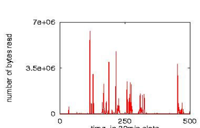
- predict queue length distributions (e.g., to give probabilistic guarantees)
- “learn” traffic, for buffering, prefetching, ‘active disks’, web servers

CMU SCS

Q: any ‘pattern’?

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

bytes

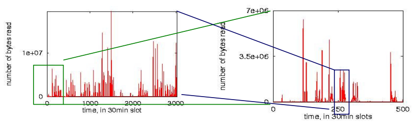


time

CMU SCS

Solution: self-similarity

bytes # bytes



time time

15-826 (c) C. Faloutsos, 2007 109

CMU SCS

But:

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

15-826 (c) C. Faloutsos, 2007 110

CMU SCS

Outline

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

15-826 (c) C. Faloutsos, 2007 111

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, 2007 112

CMU SCS

Approach

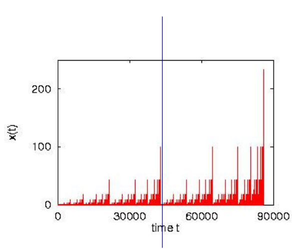
- A: ‘binomial multifractal’ [Wang+02]
- ~ 80-20 ‘law’:
  - 80% of bytes/queries etc on first half
  - repeat recursively
- *b*: bias factor (eg., 80%)

15-826 (c) C. Faloutsos, 2007 113

CMU SCS

Binary multifractals

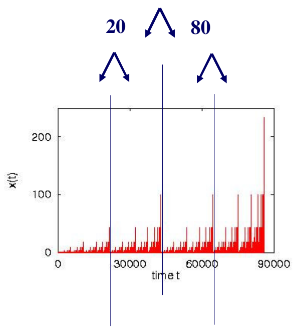
20 **80**



15-8 114

CMU SCS

Binary multifractals



CMU SCS

Parameter estimation

- Q2: How to estimate the bias factor *b*?

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]

CMU SCS

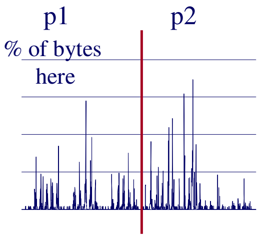
Entropy plot

- 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, 2007 118

CMU SCS

Entropy plot

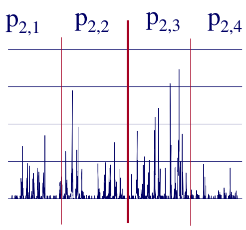


- Entropy *E(n)* after *n* levels of splits
- n=1: E(1)= - p1 $\log_2$(p1)- p2 $\log_2$(p2)

15-826 (c) C. Faloutsos, 2007 119

CMU SCS

Entropy plot



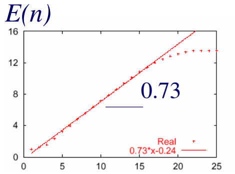
- Entropy *E(n)* after *n* levels of splits
- n=1: E(1)= - p1 log(p1)- p2 log(p2)
- n=2: $E(2) = - \Sigma_i \, p_{2,i} * \log_2 (p_{2,i})$

15-826 (c) C. Faloutsos, 2007 120

CMU SCS

Real traffic

Entropy
E(n)



# of levels (*n*)

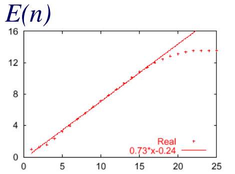
- Has linear entropy plot (-> self-similar)

15-826 (c) C. Faloutsos, 2007 121

CMU SCS

Observation - intuition:

Entropy
E(n)



# of levels (*n*)

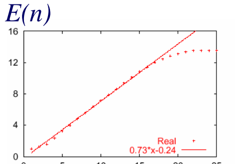
intuition: slope =
intrinsic dimensionality =
info-bits per coordinate-bit
- unif. Dataset: slope =?
- multi-point: slope = ?

15-826 (c) C. Faloutsos, 2007 122

CMU SCS

Observation - intuition:

Entropy
E(n)



# of levels (*n*)

intuition: slope =
intrinsic dimensionality =
info-bits per coordinate-bit
- unif. Dataset: slope =1
- multi-point: slope = 0

15-826 (c) C. Faloutsos, 2007 123

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, 2007 124

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, 2007 125

CMU SCS

Entropy plot

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

Dim = 1

Is MSB 0?

Is next MSB =0?

15-826 (c) C. Faloutsos, 2007 126

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 =1 bit = E(2) - E(1) = slope!

Is next MSB =0?

15-826 (c) C. Faloutsos, 2007 127

CMU SCS

Entropy plot

- Repeat, for all points at same position:

Dim=0

15-826 (c) C. Faloutsos, 2007 128

CMU SCS

Entropy plot

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

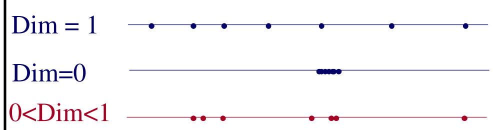
Dim=0

15-826 (c) C. Faloutsos, 2007 129

CMU SCS

Entropy plot

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



15-826 (c) C. Faloutsos, 2007 130

CMU SCS

(Fractals, again)

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

15-826 (c) C. Faloutsos, 2007 131

CMU SCS

Cantor dust

- Eliminate the middle third
- Recursively!

15-826 (c) C. Faloutsos, 2007 132

CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2007 133

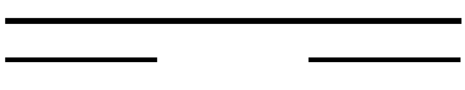
CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2007 134

CMU SCS

Cantor dust



15-826 (c) C. Faloutsos, 2007 135

CMU SCS

Cantor dust

15-826 (c) C. Faloutsos, 2007 136

CMU SCS

Cantor dust

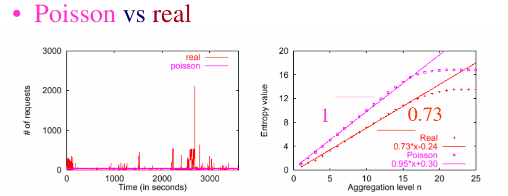
Dimensionality?
(no length; infinite # points!)
Answer: log2 / log3 = 0.6

15-826 (c) C. Faloutsos, 2007 137

CMU SCS

Some more entropy plots:

- Poisson vs real



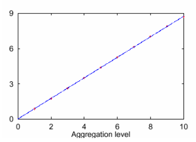
Poisson: slope = ~1 -> uniformly distributed

15-826 (c) C. Faloutsos, 2007 138

CMU SCS

b-model

$E(n)$



n

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

15-826 (c) C. Faloutsos, 2007 139

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, 2007 140

CMU SCS

Experimental setup

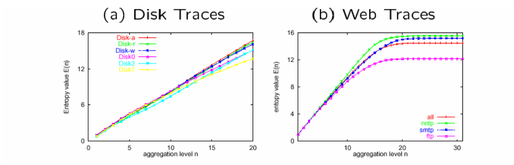
- Disk traces (from HP [Wilkes 93])
- web traces from LBL
  `http://repository.cs.vt.edu/`
  `lbl-conn-7.tar.Z`

15-826 (c) C. Faloutsos, 2007 141

CMU SCS

Model validation

- Linear entropy plots



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

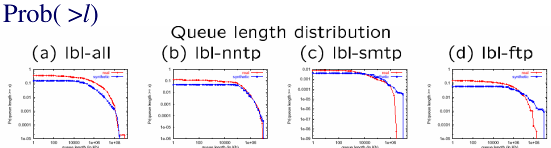
15-826 (c) C. Faloutsos, 2007 142

CMU SCS

Web traffic - results

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

Prob(>*l*)



How to give guarantees? (queue length *l*)

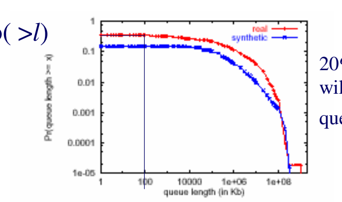
15-826 (c) C. Faloutsos, 2007 143

CMU SCS

Web traffic - results

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

Prob(>*l*)



20% of the requests will see queue lengths <100

(queue length *l*)

15-826 (c) C. Faloutsos, 2007 144

CMU SCS

Conclusions

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

15-826 (c) C. Faloutsos, 2007 145

CMU SCS

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, 2007 146

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, 2007 147

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.

15-826 (c) C. Faloutsos, 2007 148

CMU SCS

Outline

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

15-826 (c) C. Faloutsos, 2007 149

CMU SCS

Chaos and non-linear forecasting

15-826 (c) C. Faloutsos, 2007 150

CMU SCS

Detailed Outline

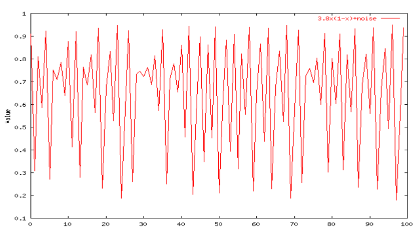
- Non-linear forecasting
  - Problem
  - Idea
  - How-to
  - Experiments
  - Conclusions

15-826 (c) C. Faloutsos, 2007 151

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}, ...

15-826 (c) C. Faloutsos, 2007 152

CMU SCS

How to forecast?

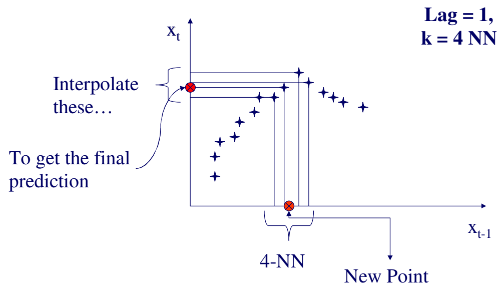
- ARIMA - but: linearity assumption

- ANSWER: 'Delayed Coordinate Embedding' = Lag Plots [Sauer92]

15-826 (c) C. Faloutsos, 2007 153

CMU SCS

General Intuition (Lag Plot)



15-826 (c) C. Faloutsos, 2007 154

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, 2007 155

CMU SCS

Q1: Choosing lag *L*

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

15-826 (c) C. Faloutsos, 2007 156

CMU SCS

Q2: Choosing number of neighbors *k*

- Manually (typically ~ 1-10)

CMU SCS

Q3: How to interpolate?

How do we interpolate between the *k* nearest neighbors?

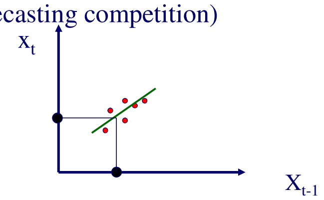
A3.1: Average

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

CMU SCS

Q3: How to interpolate?

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



CMU SCS

Q4: Any theory behind it?

A4: YES!

15-826 (c) C. Faloutsos, 2007 160

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, 2007 161

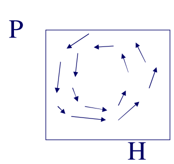
CMU SCS

Skip

Theoretical foundation

Example: Lotka-Volterra equations

$$dH/dt = r\,H - a\,H*P$$
$$dP/dt = b\,H*P - m\,P$$



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, 2007 162

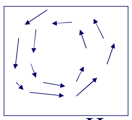
CMU SCS

Skip

Theoretical foundation

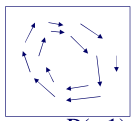
- But the delay vector space is a faithful reconstruction of the internal system state
- So prediction in **delay vector space** is as good as prediction in **state space**

P



H

P(t)



P(t-1)

15-826 (c) C. Faloutsos, 2007 163

CMU SCS

Detailed Outline

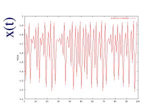
- Non-linear forecasting
  - Problem
  - Idea
  - How-to
  - Experiments
  - Conclusions

15-826 (c) C. Faloutsos, 2007 164

CMU SCS

Datasets

x(t)

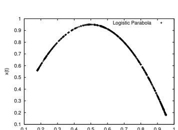


time

Logistic Parabola:

$x_t = a x_{t-1}(1-x_{t-1})$ + noise

Models population of flies [R. May/1976]

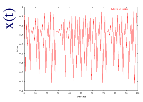


Lag-plot

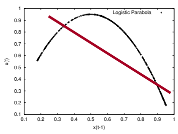
15-826 (c) C. Faloutsos, 2007 165

CMU SCS

Datasets



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

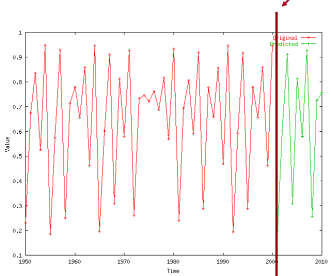


Lag-plot
ARIMA: fails

15-826 (c) C. Faloutsos, 2007 166

CMU SCS

Logistic Parabola

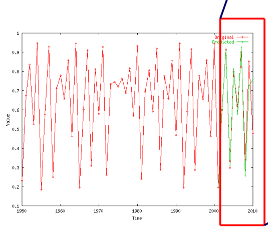


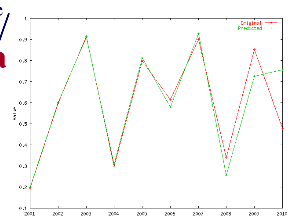
15-826 (c) C. Faloutsos, 2007 167

CMU SCS

Logistic Parabola

Comparison of prediction to correct values





15-826 (c) C. Faloutsos, 2007 168

CMU SCS

Value

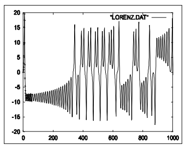
Datasets

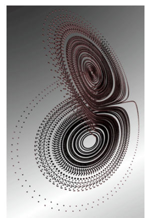
LORENZ: Models convection currents in the air

$dx / dt = a (y - x)$

$dy / dt = x (b - z) - y$

$dz / dt = xy - c z$





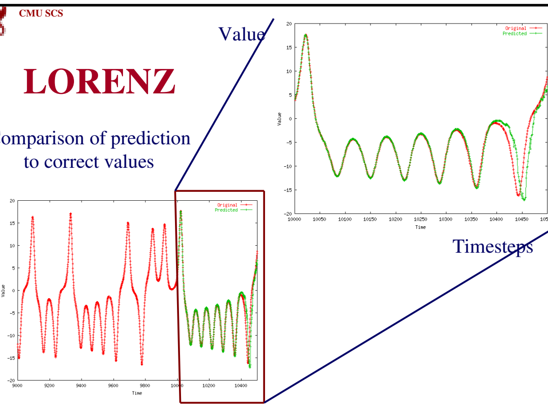
15-826 (c) C. Faloutsos, 2007 169

CMU SCS

Value

LORENZ

Comparison of prediction to correct values



Timesteps

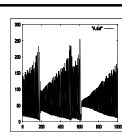
15-826 (c) C. Faloutsos, 2007 170

CMU SCS

Value

Datasets

- LASER: fluctuations in a Laser over time (used in Santa Fe competition)



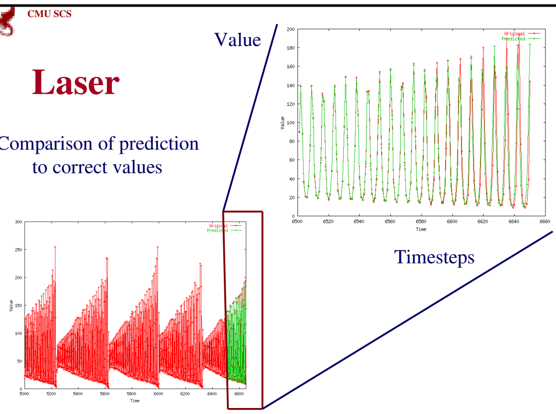
Time

15-826 (c) C. Faloutsos, 2007 171

CMU SCS

Laser

Comparison of prediction to correct values



15-826 (c) C. Faloutsos, 2007 172

CMU SCS

Conclusions

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

15-826 (c) C. Faloutsos, 2007 173

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, 2007 174

CMU SCS

References

- Weigend, A. S. and N. A. Gerschenfeld (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, 2007 175

CMU SCS

Overall conclusions

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

15-826 (c) C. Faloutsos, 2007 176

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, 2007 177

Overall conclusions

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

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')

Overall conclusions

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