

Carnegie Mellon

Mining graphs and time series: patterns, anomalies, and fraud detection

Part 2: Time Series - Forecasting & Tensors

Christos Faloutsos

CMU SCS


<https://www.cs.cmu.edu/~christos/TALKS/19-Gol>



1

Carnegie Mellon

Roadmap



- Introduction
- Part#1: Graphs and Tensors
- Part#2: Time series
- Part#3: extras (visualization, etc)
- Conclusions

Gov. of India Copyright (C) 2019 C. Faloutsos

2

2

Carnegie Mellon

Outline

- Motivation
- Similarity Search and Indexing
- DSP
- ➔ • Linear Forecasting
 - Non-linear forecasting
- Tensors
- Conclusions

Gov. of India Copyright (C) 2019 C. Faloutsos

3

3

Carnegie Mellon

Part 2.3:

Linear Forecasting

Gov. of India
Copyright (C) 2019 C. Faloutsos
4


4

Carnegie Mellon

Forecasting

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

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



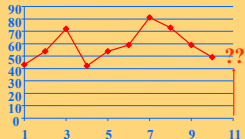
Gov. of India
Copyright (C) 2019 C. Faloutsos
5

5

Carnegie Mellon

Problem#2: Forecast

- given x_{t-1}, x_{t-2}, \dots ,
- Q: forecast x_t



Gov. of India
Copyright (C) 2019 C. Faloutsos
6

6

Carnegie Mellon

Solution: AR(IMA)

- given x_{t-1}, x_{t-2}, \dots ,
- Q: forecast x_t
- A: AR(IMA) = Box-Jenkins (< Holt-Winters, Kalman)

Gov. of India
Copyright (C) 2019 C. Faloutsos
7

7

Carnegie Mellon

Outline

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

Gov. of India
Copyright (C) 2019 C. Faloutsos
8

8

Carnegie Mellon

Problem#2: Forecast

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

Gov. of India
Copyright (C) 2019 C. Faloutsos
9

9

Carnegie Mellon

Forecasting: Preprocessing

MANUALLY:

remove trends

spot periodicities
7 days

Gov. of India Copyright (C) 2019 C. Faloutsos 10

10

Carnegie Mellon

Problem#2: Forecast

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

Formally:

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

Gov. of India Copyright (C) 2019 C. Faloutsos 11

11

Carnegie Mellon

(Problem: Back-cast; interpolate)

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

Gov. of India Copyright (C) 2019 C. Faloutsos 12

12

Carnegie Mellon

Linear Regression: idea

patient	weight	height
1	27	43
2	43	54
3	54	72
...
N	25	??

Body height

Body weight

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

Gov. of India

Copyright (C) 2019 C. Faloutsos

13

13

Carnegie Mellon

Linear Auto Regression:

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

Gov. of India

Copyright (C) 2019 C. Faloutsos

14

14

Carnegie Mellon

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 ($S[t-1]$)

Gov. of India

Copyright (C) 2019 C. Faloutsos

15

15

Carnegie Mellon

Outline

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

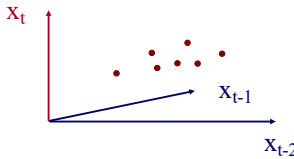
Gov. of India Copyright (C) 2019 C. Faloutsos 16

16

Carnegie Mellon

More details:

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



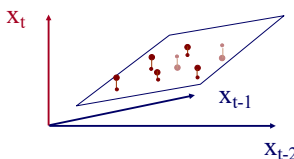
Gov. of India Copyright (C) 2019 C. Faloutsos 17

17

Carnegie Mellon

More details:

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



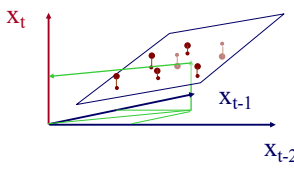
Gov. of India Copyright (C) 2019 C. Faloutsos 18

18

Carnegie Mellon

More details:

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



Gov. of India Copyright (C) 2019 C. Faloutsos 19

19

Carnegie Mellon

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

Gov. of India Copyright (C) 2019 C. Faloutsos 20

20

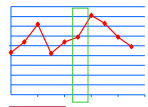
Carnegie Mellon

More details:

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

Ind-var1 Ind-var-w

time



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

Gov. of India Copyright (C) 2019 C. Faloutsos 21

21

Carnegie Mellon

More details: Skip

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

Ind-var1 Ind-var-w

time

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

Gov. of India Copyright (C) 2019 C. Faloutsos 22

22

Carnegie Mellon

More details Skip

- Q2: How to estimate $a_1, a_2, \dots, a_w = \mathbf{a}$?
- A2: with Least Squares fit
- $(\text{Model} = \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$
- \mathbf{a} is the vector that minimizes the RMSE from \mathbf{y}

Gov. of India Copyright (C) 2019 C. Faloutsos 23

23

Carnegie Mellon

Even more details Skip

- 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) - pictorially:

Gov. of India Copyright (C) 2019 C. Faloutsos 24

24

Carnegie Mellon

Even more details

Skip

- Given:

Gov. of India

Copyright (C) 2019 C. Faloutsos

25

25

Carnegie Mellon

Even more details

Skip

Gov. of India

Copyright (C) 2019 C. Faloutsos

26

26

Carnegie Mellon

Even more details

Skip

RLS: quickly compute new best fit

Gov. of India

Copyright (C) 2019 C. Faloutsos

27

27

Carnegie Mellon

Even more details

Skip

- 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
 $O(1 \times w^2)$

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

Gov. of India

Copyright (C) 2019 C. Faloutsos

28

28

Carnegie Mellon

Even more details

Skip

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

Gov. of India

Copyright (C) 2019 C. Faloutsos

29

29

Carnegie Mellon

Adaptability - ‘forgetting’

Skip

Dependent Variable
eg., #bytes sent

Independent Variable
eg., #packets sent

Gov. of India

Copyright (C) 2019 C. Faloutsos

30

30

Carnegie Mellon

Outline

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

Gov. of India
Copyright (C) 2019 C. Faloutsos
31

31

Carnegie Mellon

Co-Evolving Time Sequences

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

Time Tick	sent	lost	repeated
1	40	20	20
2	50	25	25
3	70	35	25
4	45	25	35
5	55	25	25
6	60	30	25
7	80	40	30
8	70	35	40
9	60	30	35
10	50	25	30
11	45	20	20

Gov. of India
Copyright (C) 2019 C. Faloutsos
32

32

Carnegie Mellon

Solution:

Q: what should we do?

Gov. of India
Copyright (C) 2019 C. Faloutsos
33

33

Carnegie Mellon

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])

Gov. of India Copyright (C) 2019 C. Faloutsos 34

34

Carnegie Mellon

Time Series Analysis - Outline

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

Gov. of India Copyright (C) 2019 C. Faloutsos 35

35

Carnegie Mellon

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]


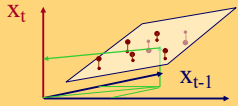
Gov. of India Copyright (C) 2019 C. Faloutsos 36

36

Carnegie Mellon

Solution: AR(IMA)

- given x_{t-1}, x_{t-2}, \dots ,
- Q: forecast x_t
- A: AR(IMA) = Box-Jenkins (< Holt-Winters, Kalman)

Gov. of India Copyright (C) 2019 C. Faloutsos 37

37

Carnegie Mellon

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/>

Gov. of India Copyright (C) 2019 C. Faloutsos 38

38

Carnegie Mellon

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.

Gov. of India Copyright (C) 2019 C. Faloutsos 39

39

Carnegie Mellon

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)

Gov. of IndiaCopyright (C) 2019 C. Faloutsos40

40

Carnegie Mellon

Part 2.4:
chaos and
non-linear forecasting

Gov. of IndiaCopyright (C) 2019 C. Faloutsos41

41

Carnegie Mellon

Outline

- Motivation
- Similarity Search and Indexing
- DSP
- Linear Forecasting
- ➡ • Non-linear forecasting
- Tensors
- Conclusions

Gov. of IndiaCopyright (C) 2019 C. Faloutsos42

42

Carnegie Mellon

Detailed Outline

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

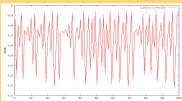
Gov. of India
Copyright (C) 2019 C. Faloutsos
43

43

Carnegie Mellon

Problem: Forecast

- given x_{t-1}, x_{t-2}, \dots , ('chaotic'/non-linear)
- Q: forecast x_t



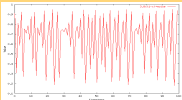
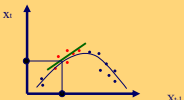
Gov. of India
Copyright (C) 2019 C. Faloutsos
44

44

Carnegie Mellon

Solution

- given x_{t-1}, x_{t-2}, \dots , ('chaotic'/non-linear)
- Q: forecast x_t
- A: lag-plots + sim. search (= 'Delayed Coordinate Embedding')

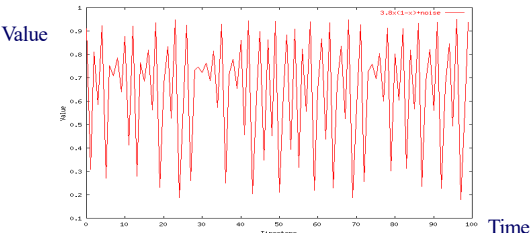



Gov. of India
Copyright (C) 2019 C. Faloutsos
45

45

Carnegie Mellon

Recall: Problem #1



Value

Time

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

Gov. of India Copyright (C) 2019 C. Faloutsos 46

46

Carnegie Mellon

How to forecast?

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


Gov. of India Copyright (C) 2019 C. Faloutsos 47

47

Carnegie Mellon

ARIMA pitfall

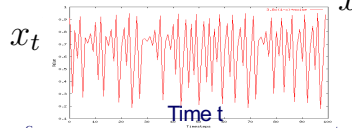
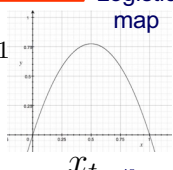
Example: logistic parabola
Models population of flies [R. May/1976]



$$x_{t+1} = ax_t \cdot (1 - x_t)$$

Logistic map

Time-series plot

Gov. of India Copyright (C) 2019 C. Faloutsos 48

48

Carnegie Mellon

ARIMA pitfall

Example: logistic parabola
Models population of flies [R. May/1976]

$$x_{t+1} = ax_t \cdot (1 - x_t)$$

Logistic map

- = SI virus prop. model
- ~ Bass equation (market penetration)
- Special case of Lotka-Volterra

Gov. of India
Copyright (C) 2019 C. Faloutsos

49

49

Carnegie Mellon

ARIMA pitfall

Linear equations, e.g., AR, ARIMA, ...

Gov. of India
Copyright (C) 2019 C. Faloutsos

50

50

Carnegie Mellon

ARIMA pitfall

Linear equations, e.g., AR, ARIMA, ...

e.g., AR(1)

$$x_{t+1} = ax_t + \epsilon$$

Gov. of India
Copyright (C) 2019 C. Faloutsos

51

51

Carnegie Mellon

ARIMA pitfall

Linear equations, e.g., AR, ARIMA, ...

e.g., AR(1)

$$x_{t+1} = ax_t + \epsilon$$

Gov. of India

Copyright (C) 2019 C. Faloutsos

52

52

Carnegie Mellon

Solution?

“Delayed Coordinate Embedding”
 = Lag Plots

[Sauer92]
 k-nearest neighbor search

Gov. of India

Copyright (C) 2019 C. Faloutsos

53

53

Carnegie Mellon

General Intuition (Lag Plot)

Lag = 1,
 k = 4 NN

Interpolate these...


To get the final prediction

Gov. of India

Copyright (C) 2019 C. Faloutsos

54

54

Carnegie Mellon


Q: How to interpolate?

How do we interpolate between the k nearest neighbors?

A1: Average

A2: Weighted average (weights drop with distance - how?)

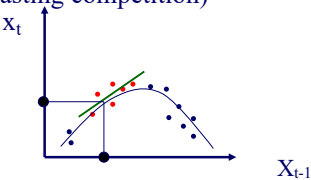
Gov. of India
Copyright (C) 2019 C. Faloutsos
55

55

Carnegie Mellon

Q: How to interpolate?

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



Gov. of India
Copyright (C) 2019 C. Faloutsos
56

56

Carnegie Mellon

Q: Any theory behind it?

A: YES!

Gov. of India
Copyright (C) 2019 C. Faloutsos
57

57

Carnegie Mellon

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)

Gov. of India

Copyright (C) 2019 C. Faloutsos

58

58

Carnegie Mellon

Theoretical foundation

Example: Lotka-Volterra equations

$$\frac{dH}{dt} = rH - aH*P$$

$$\frac{dP}{dt} = bH*P - mP$$

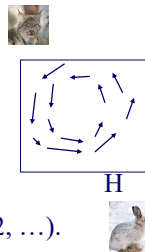
H is count of prey (e.g., hare)
P is count of predators (e.g., lynx)

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

Gov. of India

Copyright (C) 2019 C. Faloutsos

59



59

Carnegie Mellon

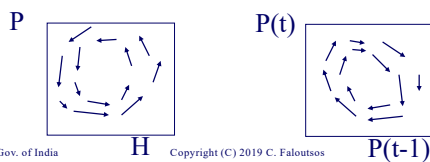
Theoretical foundation

- 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**

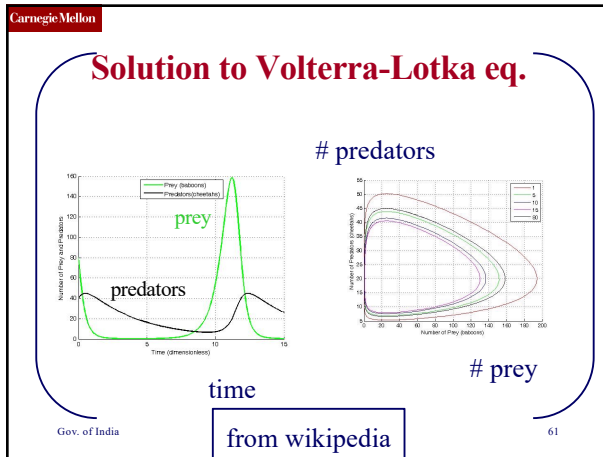
Gov. of India

Copyright (C) 2019 C. Faloutsos

60



60



61

Carnegie Mellon

Notice: LV are vital!

Example: Lotka-Volterra equations

$$\frac{dH}{dt} = rH - aH*P$$

$$\frac{dP}{dt} = bH*P - mP$$

- Prey-predator
- Competing animals (rabbits/goats)
- Self-competition (Bass model)
- **Competing products (stocks/bonds)**

P

H

Gov. of India

Copyright (C) 2019 C. Faloutsos

62

62

Carnegie Mellon

The Web as a Jungle: Non-Linear Dynamical Systems for Co-evolving Online Activities

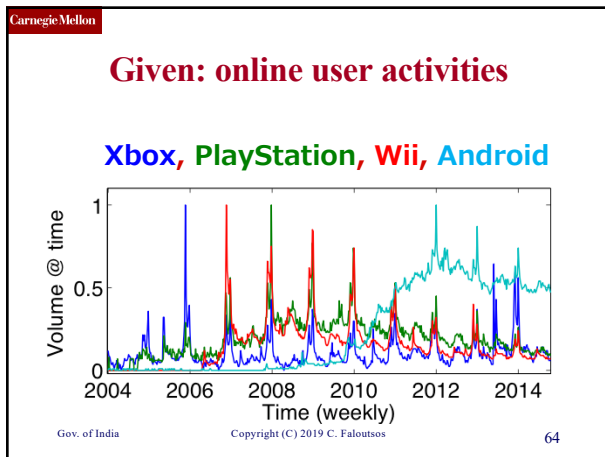
Yasuko Matsubara (Kumamoto University)

Yasushi Sakurai (Kumamoto University)

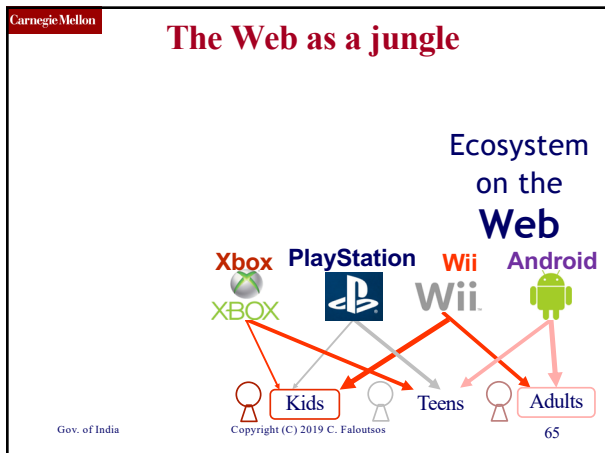
Christos Faloutsos (CMU)

Open source code: [here](#)

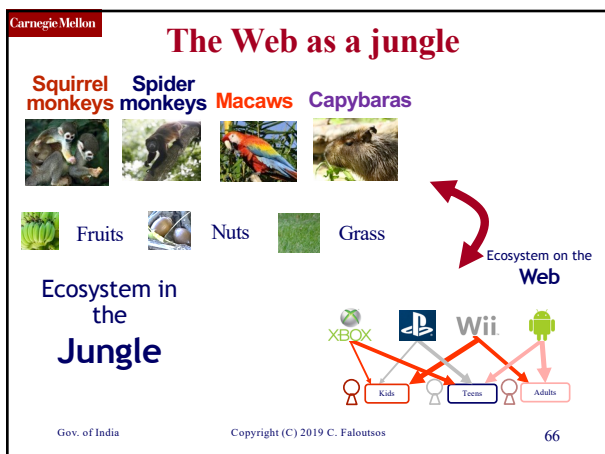
63



64



65



66

Carnegie Mellon

LV equations

Interaction between multiple (d) species/products/viruses

$$P_i(t+1) = P_i(t) \left[1 + r_i \left(1 - \frac{\sum_{j=1}^d a_{ij} P_j(t)}{K_i} \right) \right],$$

$(i = 1, \dots, d),$

a_{ij} - effect of species j on species i

- (positive: hurts)

Gov. of India
Copyright (C) 2019 C. Faloutsos
67

67

Carnegie Mellon

EcoWeb at work - forecasting

Train:
2/3 sequences
Forecast:
1/3 following years

Original sequences

Gov. of India
Copyright (C) 2019 C. Faloutsos
68

68

Carnegie Mellon

EcoWeb at work - forecasting

Train:
2/3 sequences
Forecast:
1/3 following years

EcoWeb

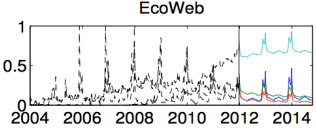
EcoWeb can capture future patterns

Gov. of India
Copyright (C) 2019 C. Faloutsos
69

69

Carnegie Mellon

EcoWeb at work - forecasting



The plot shows a time series from 2004 to 2014. The y-axis ranges from 0 to 1. There are multiple lines: a black line representing the actual data, and several colored lines (green, blue, red) representing different forecasting models. The data shows significant fluctuations, with a notable peak around 2008 and another around 2012.

Open source code: [here](#)

Gov. of India Copyright (C) 2019 C. Faloutsos 70

70

Carnegie Mellon

Detailed Outline

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

Gov. of India Copyright (C) 2019 C. Faloutsos 71

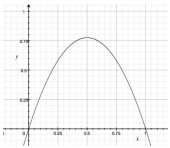
71

Carnegie Mellon

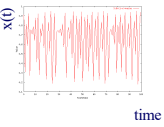
Datasets

Logistic Parabola:

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



The plot shows a parabolic curve on a grid. The x-axis ranges from 0 to 1, and the y-axis ranges from 0 to 1. The curve starts at (0,0), reaches a maximum of approximately 0.75 at x=0.5, and ends at (1,0).



The plot shows a time series x(t) over time. The y-axis ranges from -1 to 1, and the x-axis ranges from 0 to 100. The data is highly oscillatory, fluctuating between -1 and 1.

Lag-plot

Gov. of India Copyright (C) 2019 C. Faloutsos 72

72

Carnegie Mellon

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

Gov. of India

Copyright (C) 2019 C. Faloutsos

73

73

Carnegie Mellon

Logistic Parabola

Value

Our Prediction from here

here

TimeSteps

Gov. of India

Copyright (C) 2019 C. Faloutsos

74

74

Carnegie Mellon

Logistic Parabola

Value

Comparison of prediction to correct values

TimeSteps

Gov. of India

Copyright (C) 2019 C. Faloutsos

75

75

25

Carnegie Mellon

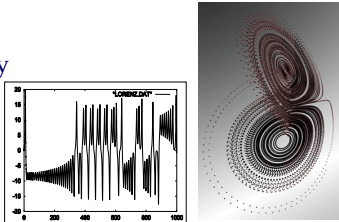
Datasets

LORENZ: Models convection currents in the air

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

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

$$dz / dt = xy - c z$$



Gov. of India

Copyright (C) 2019 C. Faloutsos

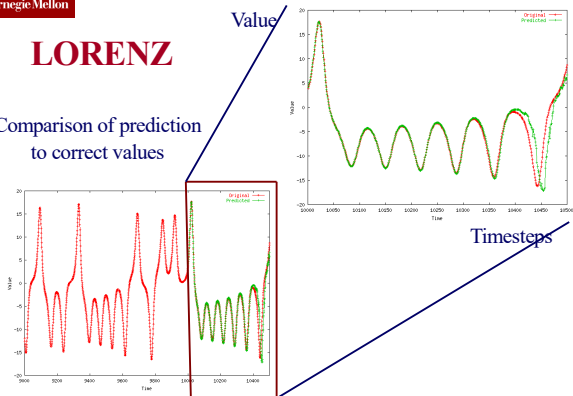
76

76

Carnegie Mellon

LORENZ

Comparison of prediction to correct values



Gov. of India

Copyright (C) 2019 C. Faloutsos

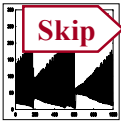
77

77

Carnegie Mellon

Datasets

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

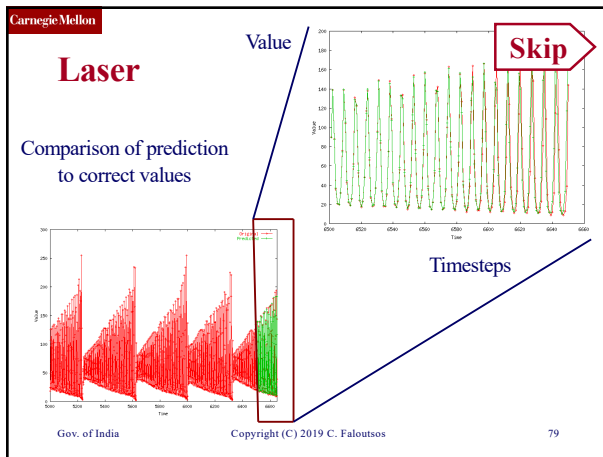


Gov. of India

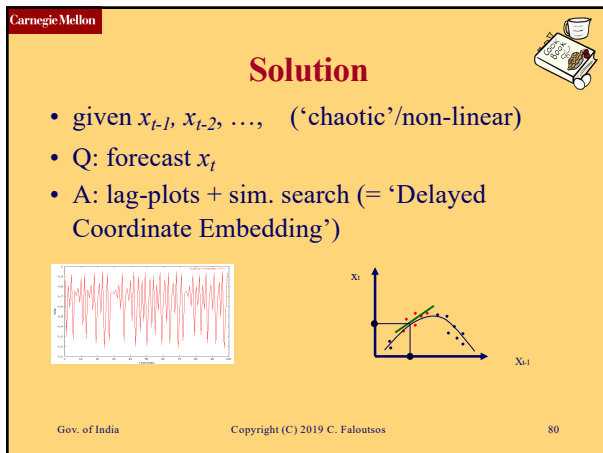
Copyright (C) 2019 C. Faloutsos

78

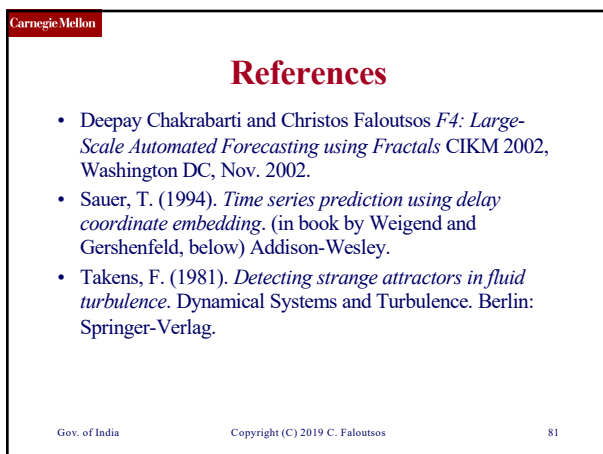
78



79



80



81

Carnegie Mellon

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.)

Gov. of India Copyright (C) 2019 C. Faloutsos 82

82

Carnegie Mellon

Part 2.5:
Tensors – time evolving
graphs

Gov. of India Copyright (C) 2019 C. Faloutsos 83

83

Carnegie Mellon

Outline

- Motivation
- Similarity Search and Indexing
- DSP
- Linear Forecasting
- Non-linear forecasting
- ➡ • Tensors
- Conclusions

Gov. of India Copyright (C) 2019 C. Faloutsos 84

84

Carnegie Mellon

Binary relationships: graph

- Who – buys - what

Gov. of India Copyright (C) 2019 C. Faloutsos 85

85

Carnegie Mellon

Ternary relationships – model?

- Who – buys – what - when

Gov. of India Copyright (C) 2019 C. Faloutsos 86

86

Carnegie Mellon

A: tensors

- ... for ternary and higher –order relationships

Gov. of India Copyright (C) 2019 C. Faloutsos 87

87

Carnegie Mellon

Problem: co-evolving graphs

- How to forecast?
 - 4M x 4M x 15 days

Gov. of India

Copyright (C) 2019 C. Faloutsos

88

88

Carnegie Mellon

A: tensors

- Q: what is a tensor?

Gov. of India

Copyright (C) 2019 C. Faloutsos

89

89

Carnegie Mellon

Tensor examples

- A: N-D generalization of matrix:

KDD' 17

	data	mining	classif.	tree	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary
Nick
...

Gov. of India

Copyright (C) 2019 C. Faloutsos

90

90

Carnegie Mellon

Tensor examples

- A: N-D generalization of matrix:

	data	mining	classif.	tree	...
John	13	11	22	55	...
Peter	5	4	6	7	...
Mary
Nick
...

Gov. of India Copyright (C) 2019 C. Faloutsos 91

91

Carnegie Mellon

Tensors are useful for 3 or more modes

Terminology: 'mode' (or 'aspect'):

	data	mining	classif.	tree	...
13	11	22	55	...	
5	4	6	7	...	
...	
...	
...	

Gov. of India Mode#3 Mode#2 Mode#1 (== aspect) #1 92

92

Carnegie Mellon

Tensor Basics

93

Tensor factorization

- Recall: (SVD) matrix factorization: finds blocks

Gov. of India Copyright (C) 2019 C. Faloutsos 94

94

Tensor factorization

- PARAFAC decomposition

Gov. of India Copyright (C) 2019 C. Faloutsos 95

95

Tensor factorization

- PARAFAC decomposition
- Results for who-calls-whom-when
– 4M x 15 days

Gov. of India Copyright (C) 2019 C. Faloutsos 96

96

Carnegie Mellon

Tensor factorization

- PARAFAC decomposition
- Results for who-calls-whom-when
 - 4M x 15 days

time
caller
callee

Gov. of India Copyright (C) 2019 C. Faloutsos 97

97

Carnegie Mellon

Tensor factorization

- PARAFAC decomposition
- Results for who-calls-whom-when
 - 4M x 15 days

time
caller
callee

Forecast in, eg, 3, instead of 1M*1M series

Gov. of India Copyright (C) 2019 C. Faloutsos 98

98

Carnegie Mellon

Important observations

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

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

Gov. of India Copyright (C) 2019 C. Faloutsos 99

99

Carnegie Mellon

Applications

- TA1: Phonecall
- TA2: Network traffic
- TA3: FaceBook
- TA4: KG Search/Annotation

Gov. of India

Copyright (C) 2019 C. Faloutsos

100

100

Carnegie Mellon

TA1: Anomaly detection in time-evolving graphs

- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks

1 caller

5 receivers

4 days of activity

~200 calls to EACH receiver on EACH day!

Gov. of India

Copyright (C) 2019 C. Faloutsos

101

101

Carnegie Mellon

TA1: Anomaly detection in time-evolving graphs

- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks

1 caller

5 receivers

4 days of activity

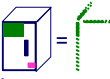
~200 calls to EACH receiver on EACH day!

Gov. of India

Copyright (C) 2019 C. Faloutsos

102

102

TA1: Anomaly detection in time-evolving graphs 

- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks

1 caller

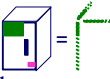
5 receivers

4 days of activity

~200 calls to EACH receiver on EACH day!

Gov. of India Copyright (C) 2019 C. Faloutsos 103

103

TA1: Anomaly detection in time-evolving graphs 

- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks

Miguel Araujo, Spiros Papadimitriou, Stephan Günnemann, Christos Faloutsos, Prithwish Basu, Ananthram Swami, Evangelos Papalexakis, Danai Koutra. *Com2: Fast Automatic Discovery of Temporal (Comet) Communities*. PAKDD 2014, Tainan, Taiwan.

104


Applications

- TA1: Phonecall
- ➡ TA2: Network traffic
- TA3: FaceBook

Gov. of India Copyright (C) 2019 C. Faloutsos 105

105

Carnegie Mellon



ParCube: Sparse Parallelizable Tensor Decompositions

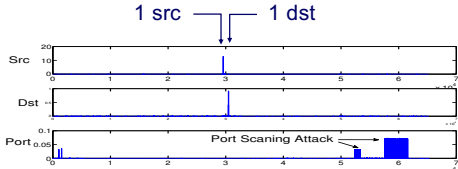
Evangelos E. Papalexakis, Christos Faloutsos, Nikos Sidiropoulos, ECML/PKDD 2012

Evangelos E. Papalexakis
 Email: epapalex@cs.ucr.edu
 Web: <http://www.cs.ucr.edu/~epapalex>

106

Carnegie Mellon

TA2: LBNL Network Data



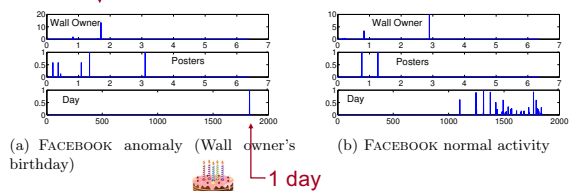
- Modes: src IP, dst IP, port #
- ~ Port Scanning Attack

Gov. of India Copyright (C) 2019 C. Faloutsos 107

107

Carnegie Mellon

TA3: FACEBOOK Wall posts



(a) FACEBOOK anomaly (Wall owner's birthday)

(b) FACEBOOK normal activity

- Modes: wall-owner, poster, timestamp
- Discovery: birthday-like event.

Gov. of India Copyright (C) 2019 C. Faloutsos 108

108

Carnegie Mellon

Conclusions (P2.5)

- Tensor analysis finds latent variables (market-segments, lockstep-groups, etc)
 - Deviations → anomalies
- Extends SVD/factorization, to higher-modes

timestamp

customer

product

=

shoes

+

Apple fans

+

jewelry

Gov. of India

Copyright (C) 2019 C. Faloutsos

109

109

Carnegie Mellon

Overall conclusions for time series:

Gov. of India

Copyright (C) 2019 C. Faloutsos

110

110

Carnegie Mellon

Overall conclusions

- P2.1. Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- P2.2. Signal processing: **DFT**, **DWT** are powerful tools
- P2.3. Linear Forecasting: **AR** (Box-Jenkins)
- P2.4. Non-linear forecasting: **lag-plots** (Takens)
- P2.5. **Tensors**: PARAFAC etc

Gov. of India


Copyright (C) 2019 C. Faloutsos

111


111

37

Carnegie Mellon






Important observations



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

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

Gov. of India Copyright (C) 2019 C. Faloutsos 112

112

Carnegie Mellon

P2 – Tensors - More references

Tensor survey

- Tamara G. Kolda and Brett W. Bader
[*Tensor Decompositions and Applications*](#)
 SIAM Rev., 51(3), pp 455–500, 2009.

Gov. of India Copyright (C) 2019 C. Faloutsos 113

113

Carnegie Mellon

P2 – Tensors - More references

Tensor survey #2

- Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos, [*Tensor Decomposition for Signal Processing and Machine Learning*](#), IEEE TSP, 65(13), July 1, 2017

Gov. of India Copyright (C) 2019 C. Faloutsos 114

114
