Carnegie Mellon

Mining Time Series: Tools and Applications

Christos Faloutsos

CMU SCS

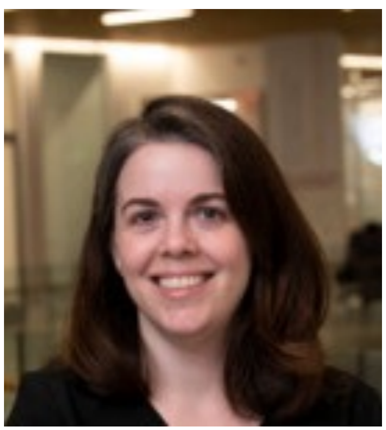
https://www.cs.cmu.edu/~christos/TALKS/21-09-exec-ed/

こんにちは

Carnegie Mellon

Thanks to:

- Prof. Zachary Lipton



- Susan Caplan

Outline

- Introduction - Motivation
- P1. Similarity Search and Indexing
- P2. DSP (Digital Signal Processing)
- P3. Linear Forecasting
- P4. Non-linear forecasting
- Conclusions

Carnegie Mellon

Problem definition

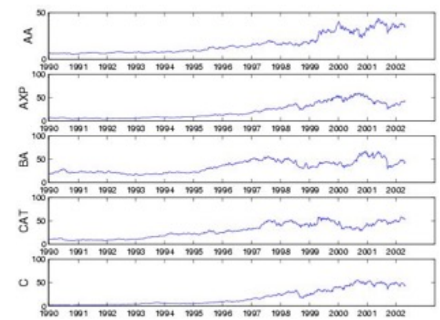
- Given: one or more sequences

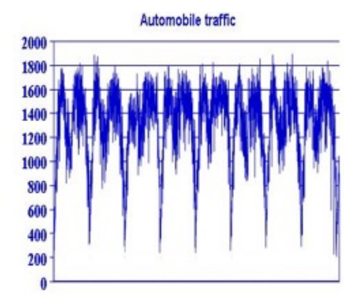
  $x_1 , \ x_2 , \ \dots , \ x_t , \ \dots$

  $(y_1, y_2, \dots , y_t, \dots$

  $\dots)$

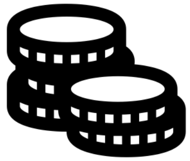
- Find
  - **Forecast**; similar sequences
  - patterns; clusters; outliers





Background of audience

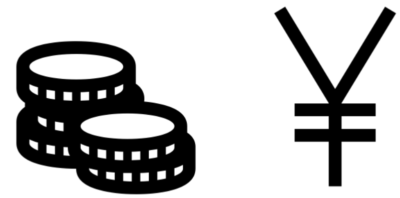
- Finance / bank




- Insurance company

- Manufacturer (chemical; electronic devices)
- Automotive

Background of audience



- Finance / bank
  - Forecasting; fraud/anomaly detection
- Insurance company
  - Same

- Manufacturer (chemical; electronic devices)
  - Sensor measurements; supply-chain forecasting
- Automotive
  - same

Questions from audience

- Are there any examples that have used DX/AI to innovate in BtoB sales, including new sales methods and the development of new markets or needs?

- [working on it, with a financial institution; 'recommendation']
- everybody is using forecasting (traditional AR etc; and Deep Learning)
- Cloud services offer such functionality, eg,
  - AWS: https://aws.amazon.com/forecast/
  - MS/Azure: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-auto-train-forecast
  - Google: https://cloud.google.com/blog/products/data-analytics/get-started-with-data-analytics-demand-forecasting-with-ml-models

Outline

- Introduction - Motivation
- P1. Similarity Search and Indexing
- P2. DSP (Digital Signal Processing)
- P3. Linear Forecasting
- P4. Non-linear forecasting
- Conclusions

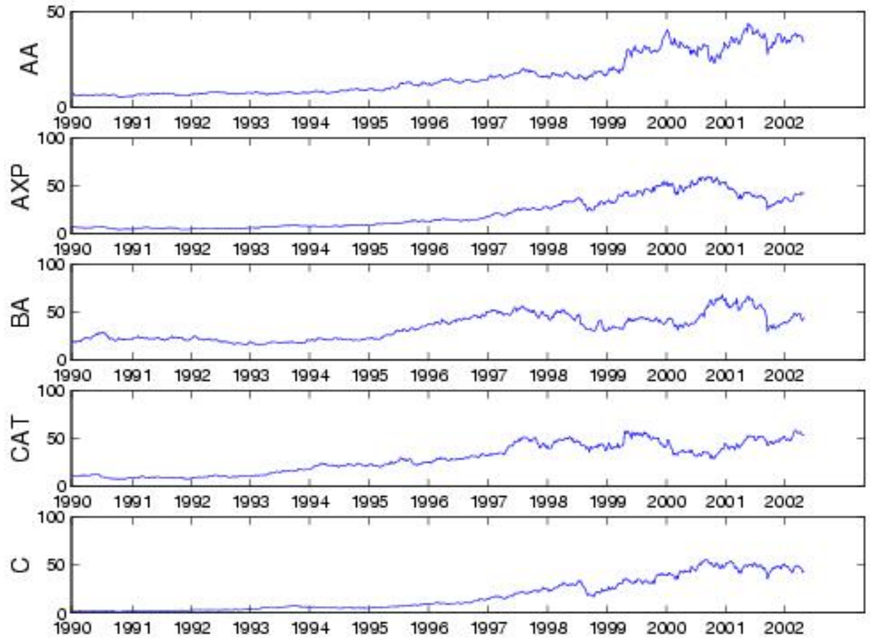
Motivation - Applications

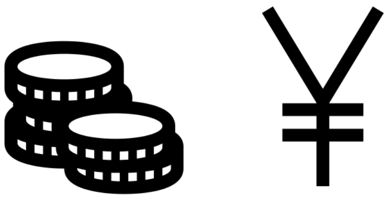
- Financial, sales, economic series

Alcoa Corp.

American express

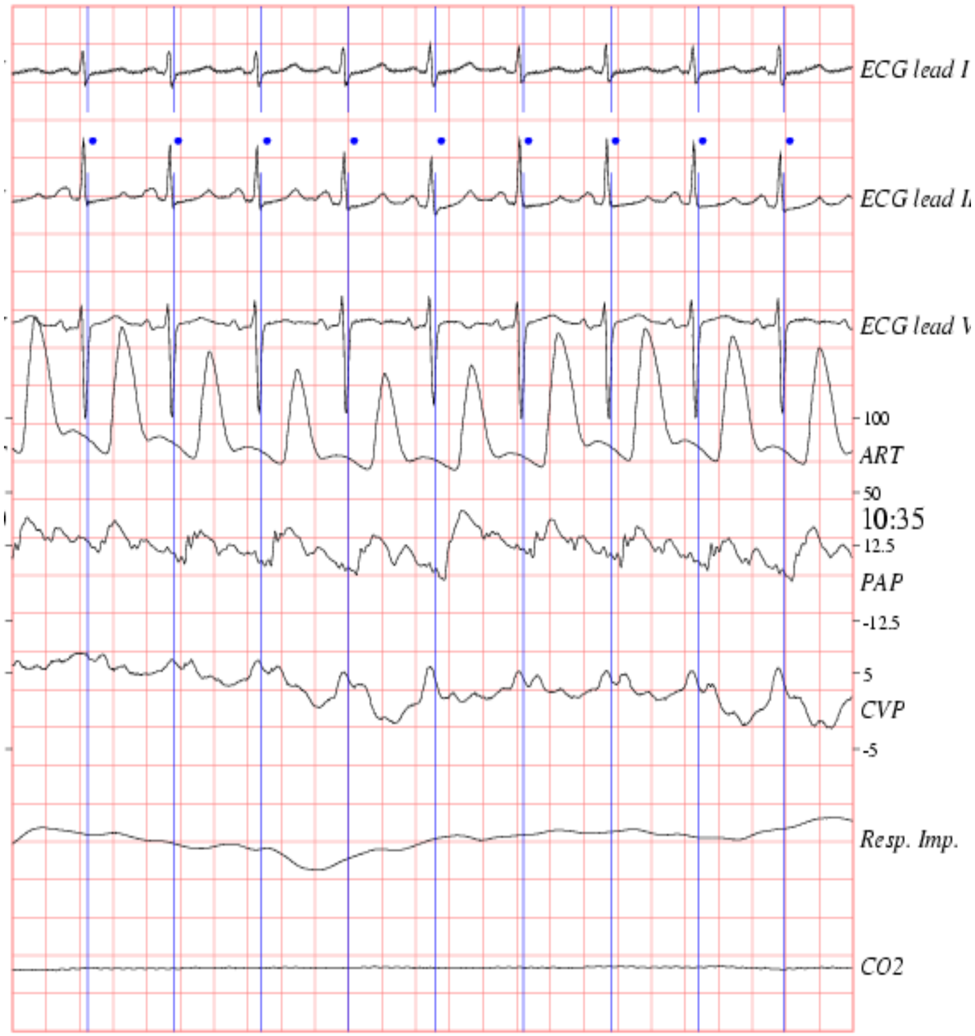
Citi group

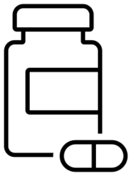




Carnegie Mellon

ECG - physionet.org





EEG - epilepsy

from wikipedia

Motivation - Applications (cont'd)

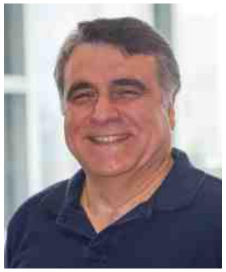
Tokyo Gate Bridge

- civil/automobile infrastructure
  - bridge vibrations [Oppenheim+02]

Motivation - Applications (cont'd)

Tokyo Gate Bridge

- civil/automobile infrastructure
  - bridge vibrations [Oppenheim+02]
  - https://www.dm.sanken.osaka-u.ac.jp/~yasuko/TALKS/17-KDD-tut/



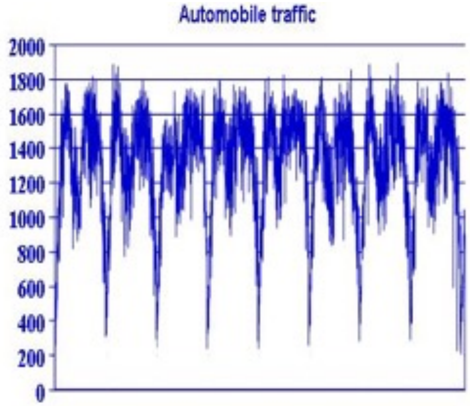
Prof. Yasushi Sakurai   Prof. Yasuko Matsubara

Carnegie Mellon

Motivation - Applications (cont'd)

- civil/automobile infrastructure
  - road conditions / traffic monitoring

From
www.transportation.gov

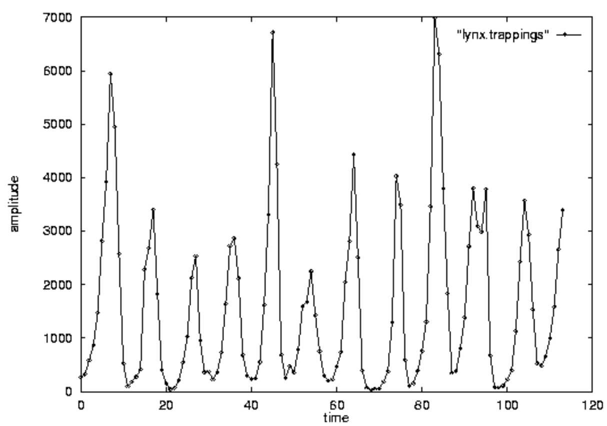


Overall Problem:

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

Find: patterns, periodicities, and/or compress

count

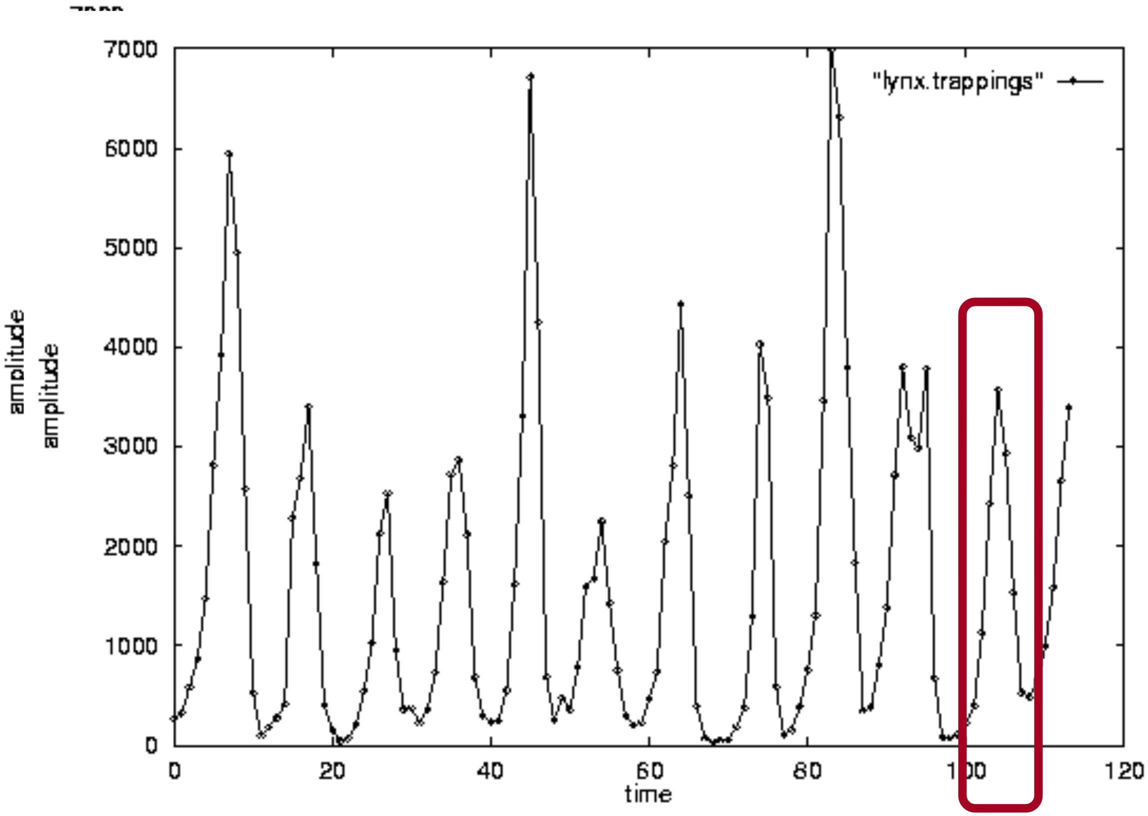


lynx caught per year
(packets per day;
temperature per day)

year

Problem#1: Similarity search

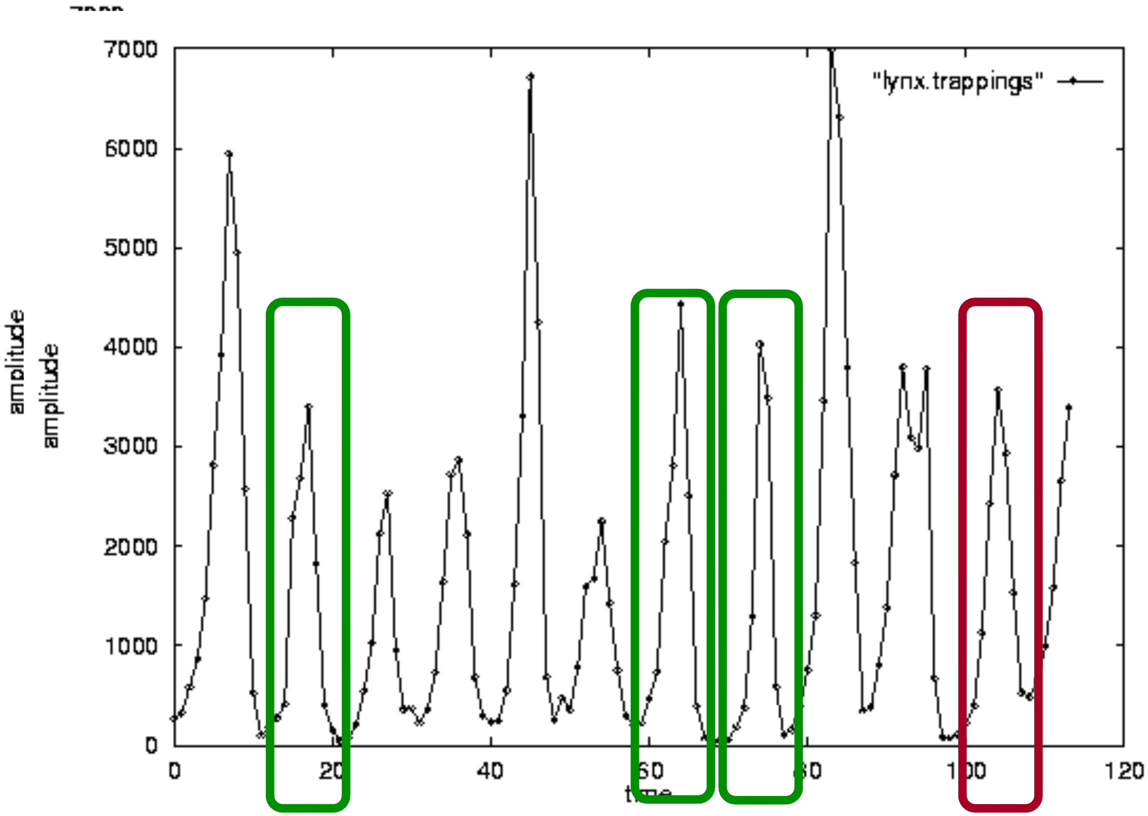
Eg., Find a10-tick pattern, similar to the last one



Carnegie Mellon

Problem#1: Similarity search

Eg., Find a10-tick pattern, similar to the last one

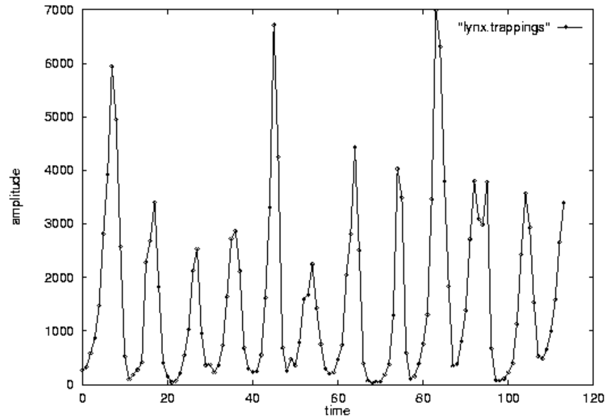


Problem #2: Patterns

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

Find: patterns, periodicities, and/or compress

count



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

year

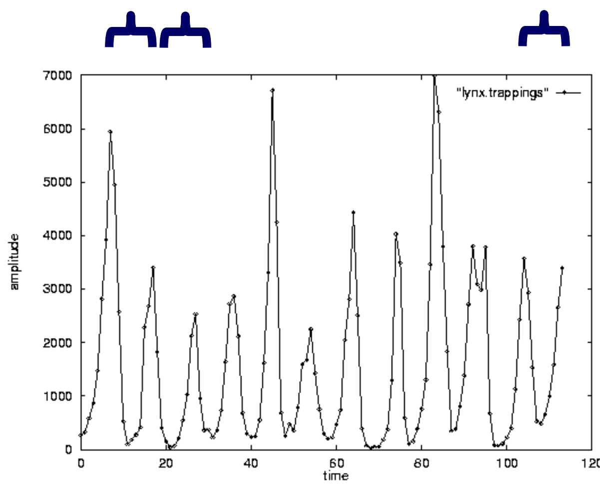
Problem #2: Patterns

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

Find: patterns, periodicities, and/or compress

count

11yr

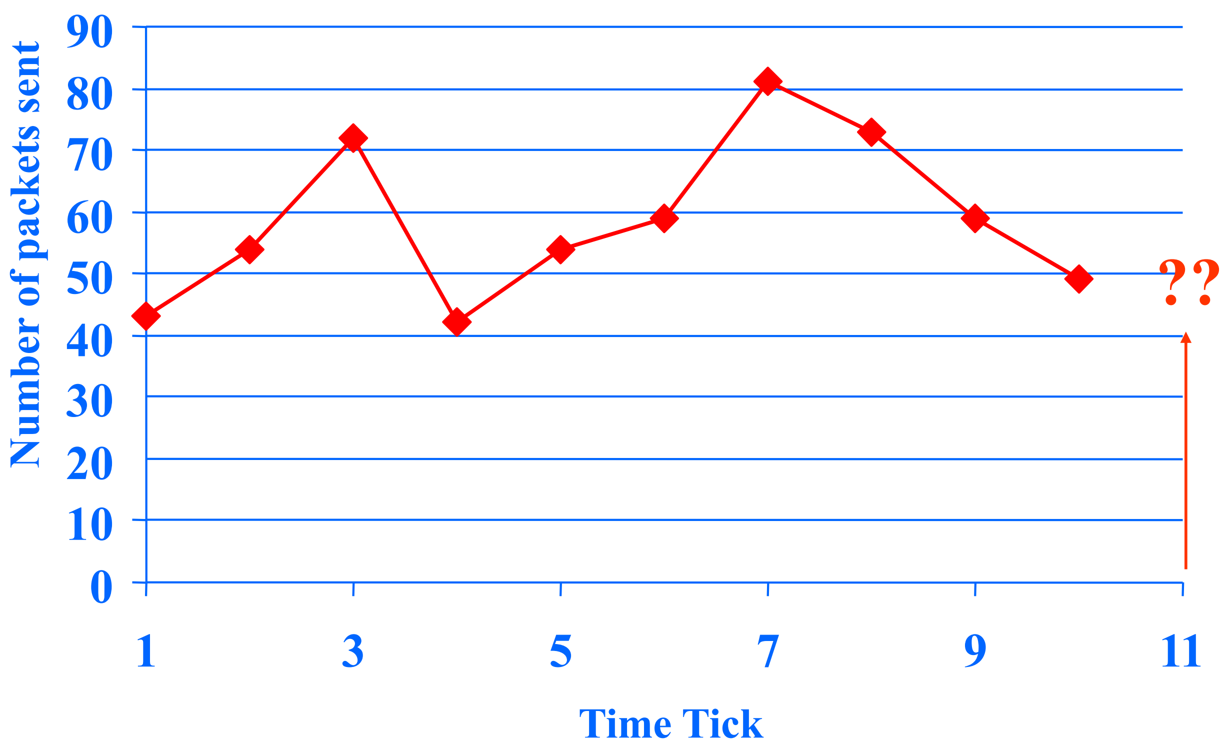


year

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

Problem #3: Forecast

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



Outline

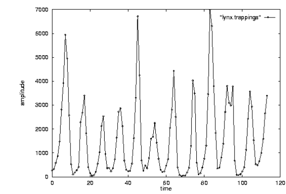
- Introduction - Motivation
  - Administrative announcements
- P1. Similarity Search and Indexing — Problem#1
- P2. DSP (Digital Signal Processing) — Problem#2
- P3. Linear Forecasting
- P4. Non-linear forecasting
- Conclusions

(P3 and P4: Problem#3)

Carnegie Mellon

Admin. ann. #1

Important observations



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

- To do **forecasting**, we need **Prob.#3**
  - to find **patterns**/rules **Prob.#2**
  - compress
  - to find **similar** past settings **Prob.#1**
- to find outliers, we need to have forecasts
  - (outlier = too far away from our forecast)

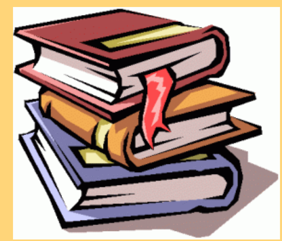
Admin. ann. #2

‘Recipe’ Structure:

- Problem definition

- Short answer/solution

- LONG answer – details



- Conclusion/short-answer

Admin. ann. #3

Check-point questions

With yellow back-ground

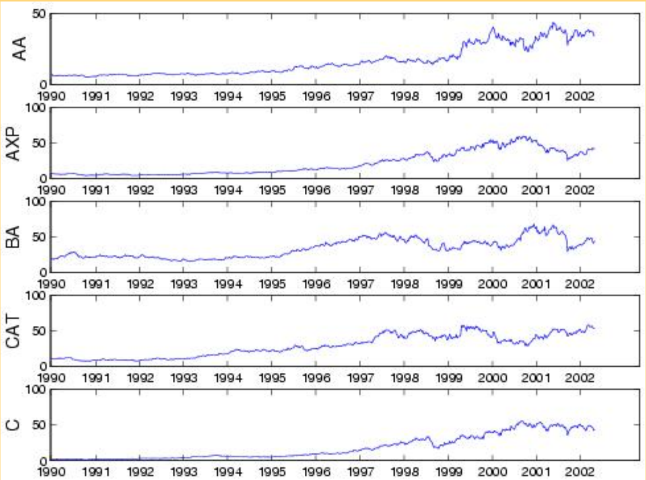
Carnegie Mellon

Outline

- Introduction - Motivation
  - 2-slide summary of this tutorial
- P1. Similarity Search and Indexing — Problem#1
- P2. DSP (Digital Signal Processing) — Problem#2
- P3. Linear Forecasting — Problem#3
- P4. Non-linear forecasting — Problem#3
- Conclusions

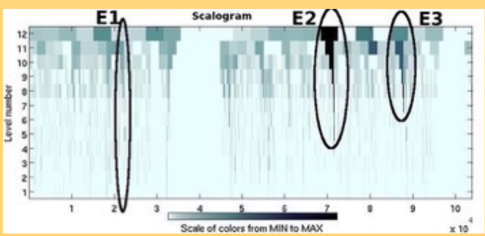
Problem:

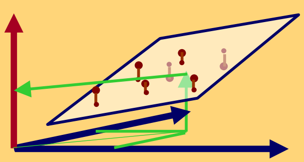
Q: mine/forecast (one, or more)
time sequences



Answers

- P1. Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- P2. Periodicities: **DFT/DWT**
- P3. Linear Forecasting: **AR** (Box-Jenkins)
- P4. Non-linear forecasting: **lag-plots**





Outline

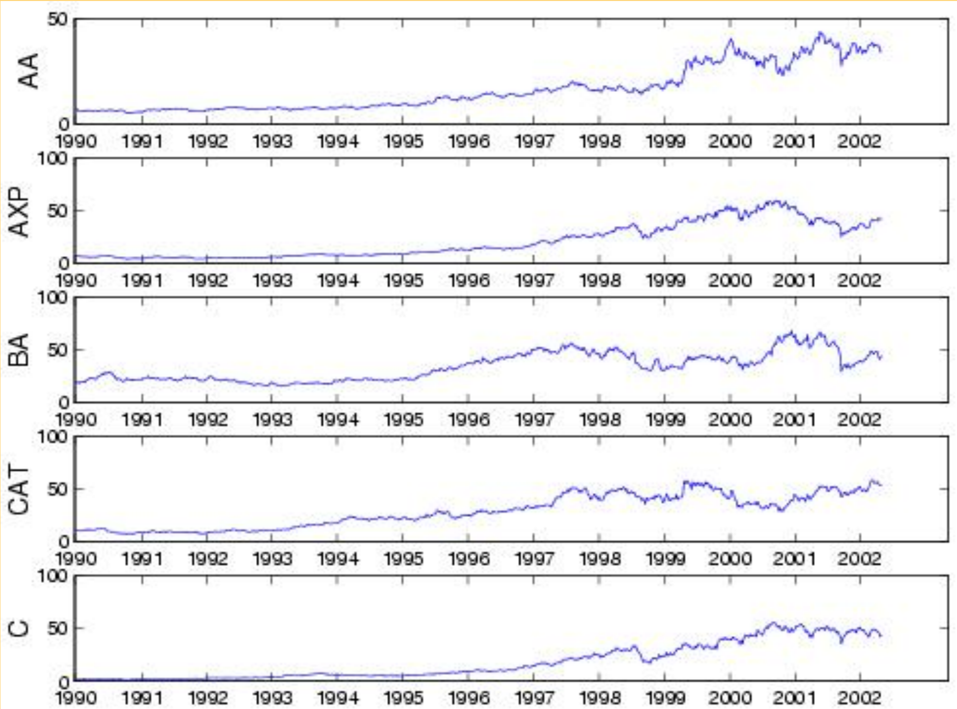
- Introduction - Motivation
- P1. Similarity Search and Indexing
- P2. DSP (Digital Signal Processing)
- P3. Linear Forecasting
- P4. Non-linear forecasting
- Conclusions

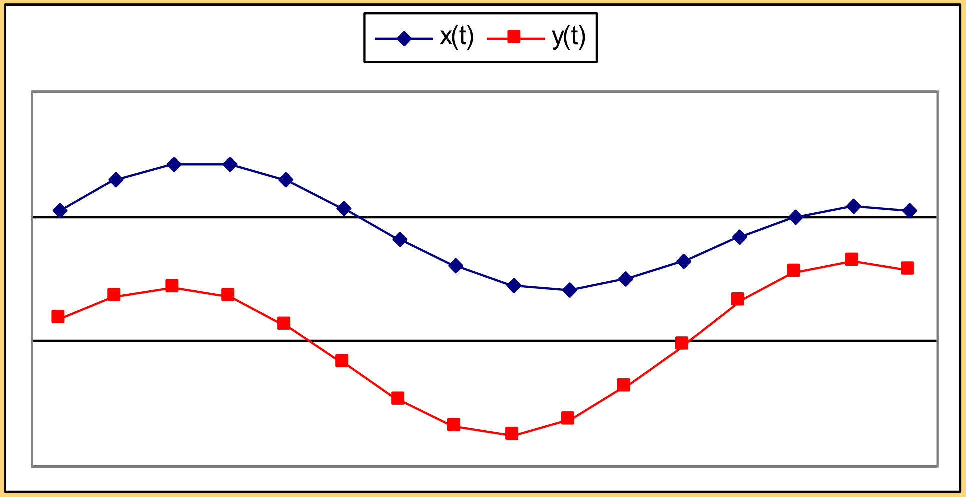
Outline

- Introduction - Motivation
- P1. Similarity Search and Indexing
  - P1.1. distance functions: Euclidean;Time-warping
  - P1.2. indexing
  - P1.3. feature extraction
- P2. DSP (Digital Signal Processing)
- ...

Problem:

Q: How similar are two sequences?

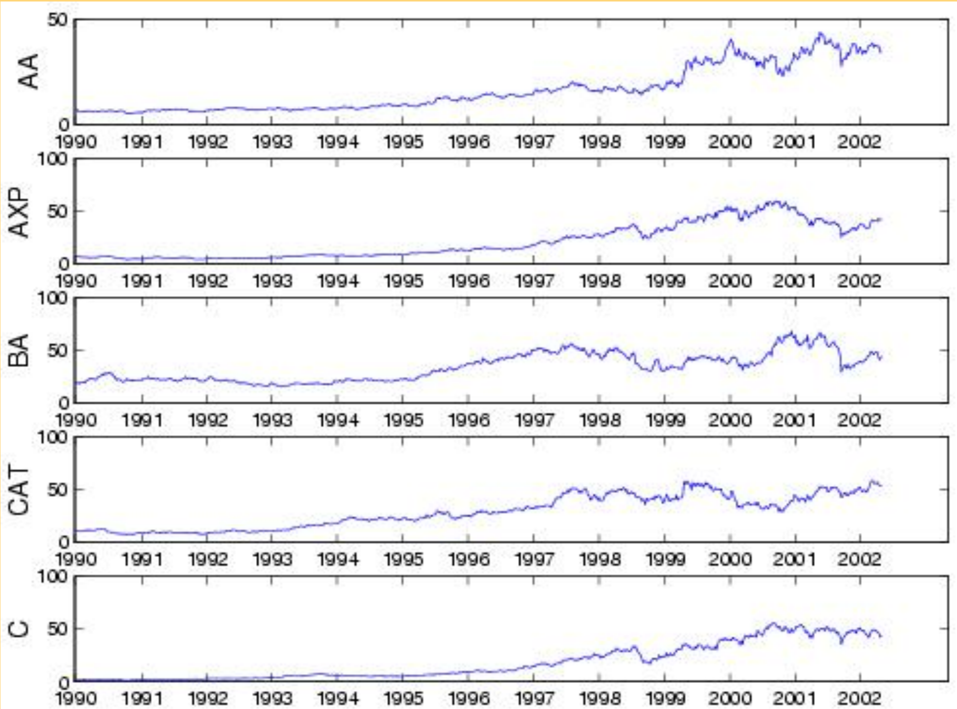




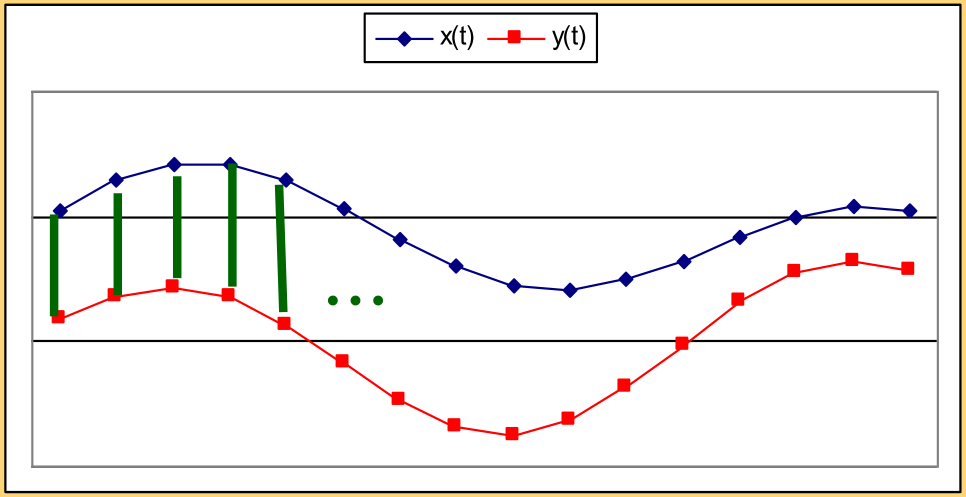
Answer:

Q: How similar are two sequences?

A: Euclidean distance (<-> cosine similarity)



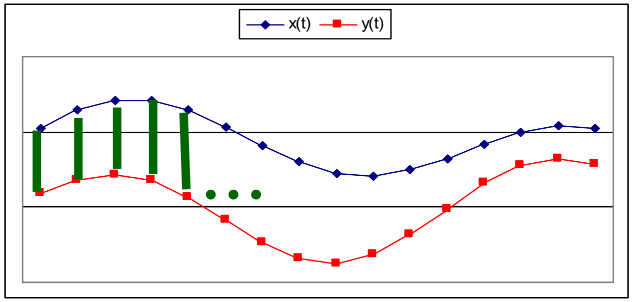
$$D(\vec{x}, \vec{y}) = \sum_{i=1}^{n} (x_i - y_i)^2$$



Importance of distance functions

Subtle, but **absolutely necessary**:

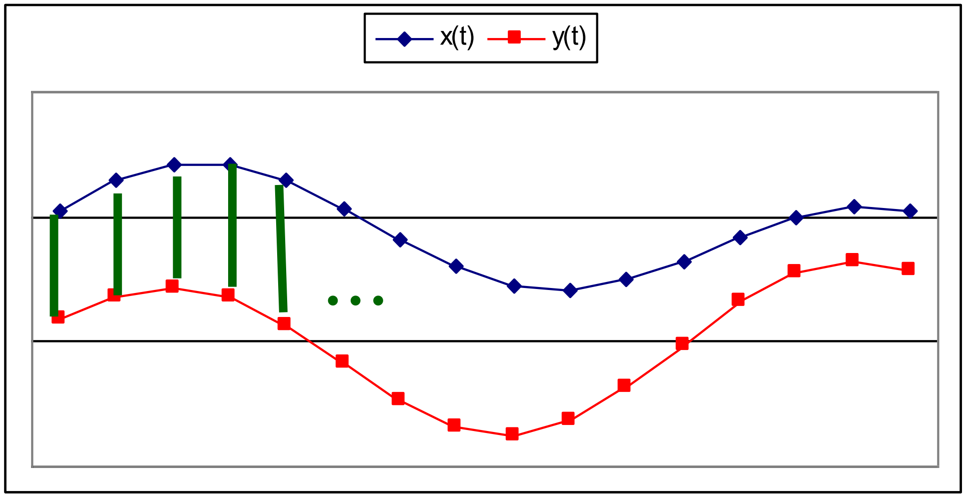
- A 'must' for similarity indexing (-> forecasting)
- A 'must' for clustering



Two major families

- – Euclidean and Lp norms
- – Time warping and variations

A1) Euclidean and Lp



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

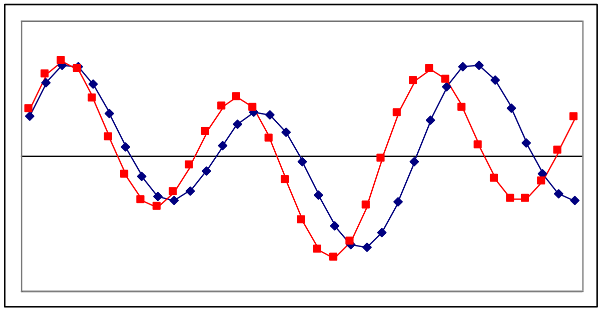
$$L_p(\vec{x}, \vec{y}) = \sum_{i=1}^{n} | x_i - y_i |^p$$

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

Carnegie Mellon

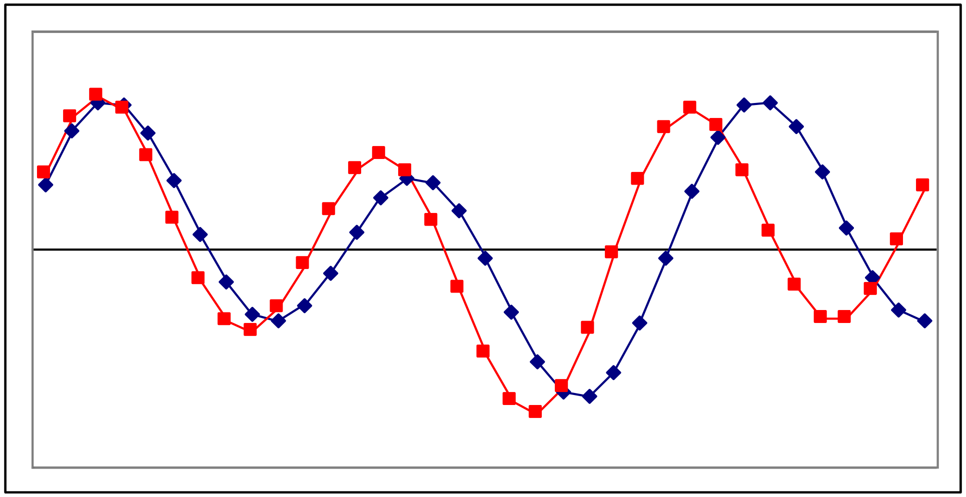
A2) Time Warping

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

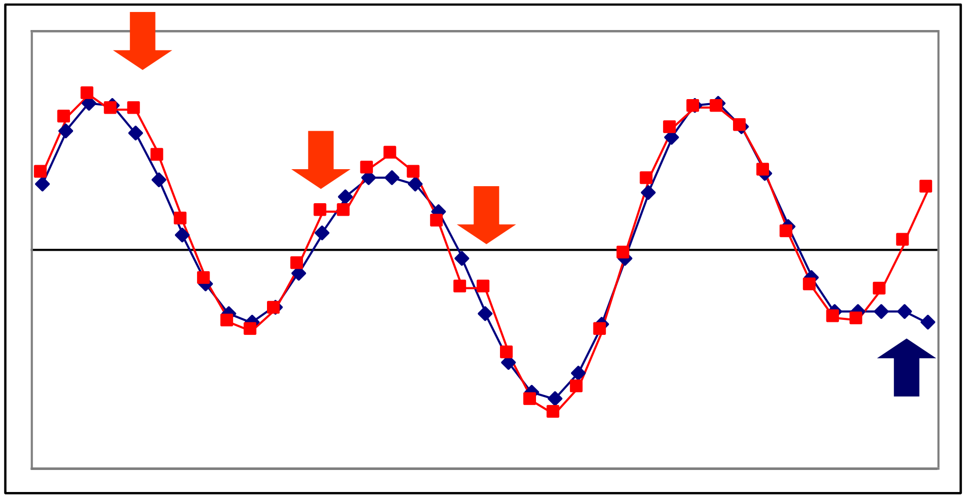


CarnegieMellon

Time Warping



'stutters':

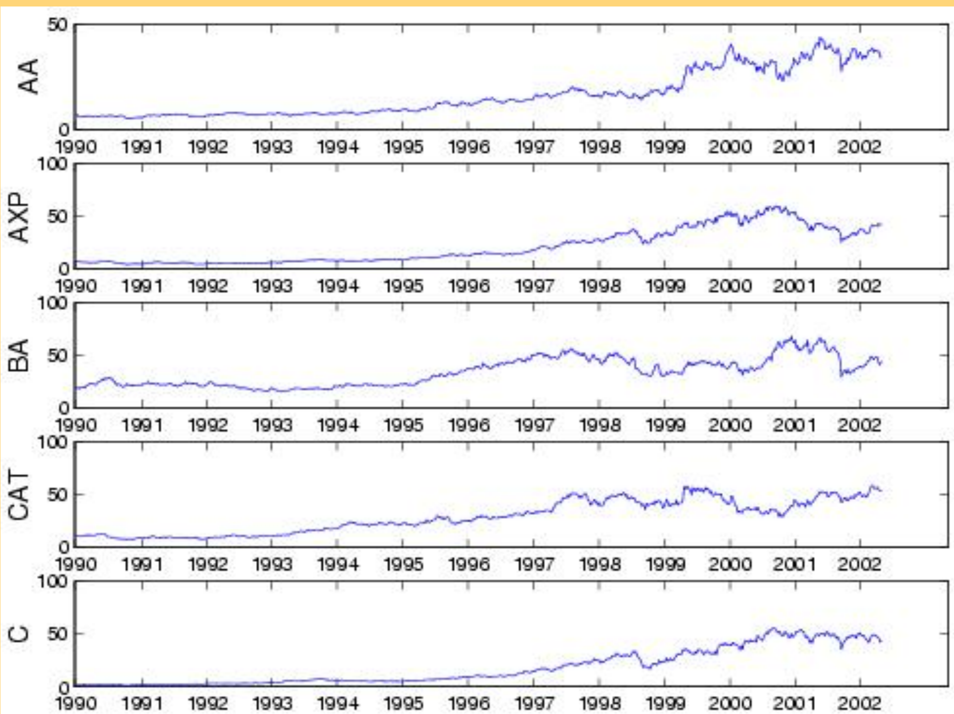


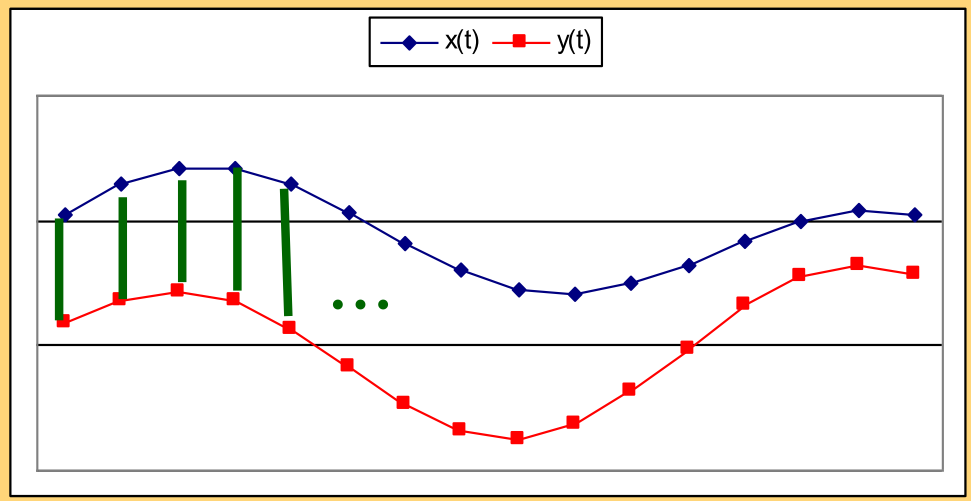
Answer:

Q: How similar are two sequences?

A: Euclidean distance

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



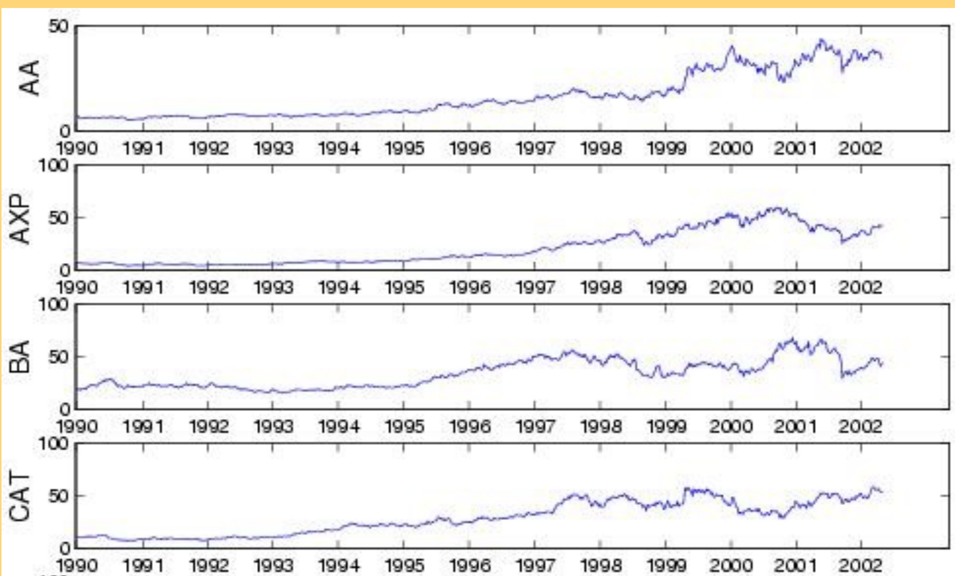


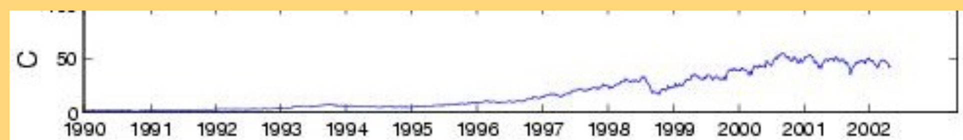
Outline

- Motivation
- P1. Similarity Search and Indexing
  - P1.1. distance functions: Euclidean;Time-warping
  - P1.2. indexing
  - P1.3. feature extraction
- P2. DSP (Digital Signal Processing)
- ...

Problem:

Q: find quickly stocks like ‘C’ (or customers like ‘smith’)

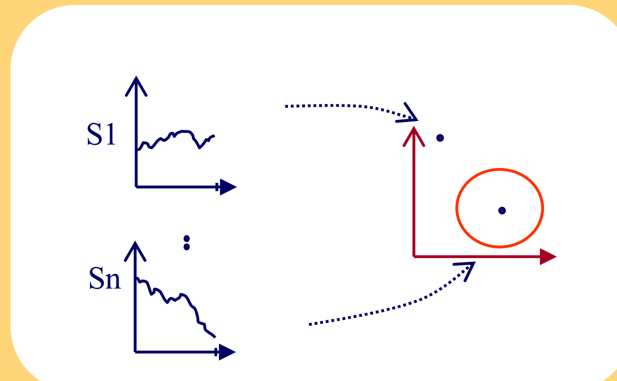


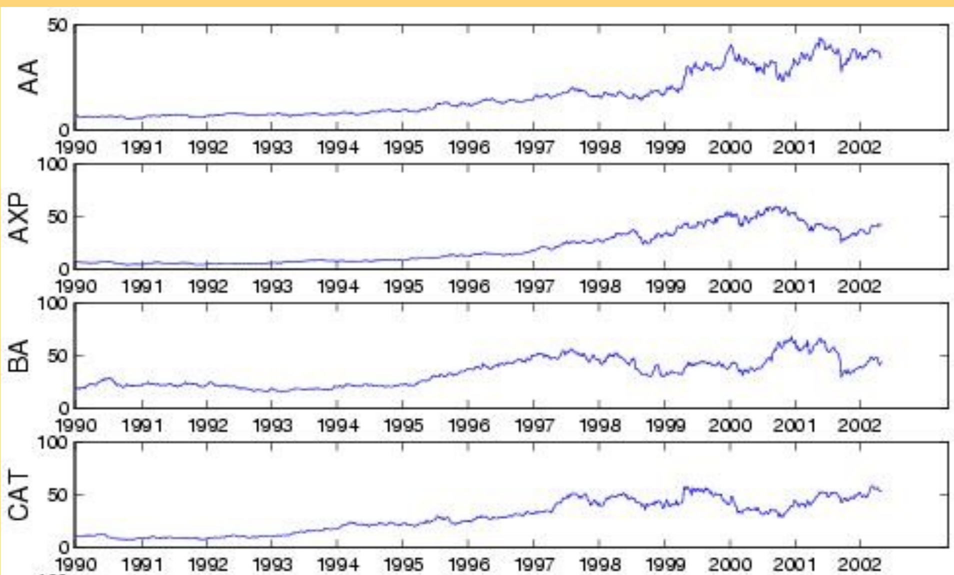


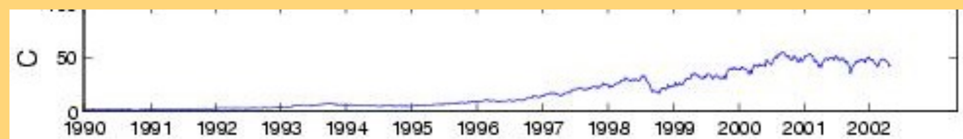
Answer:

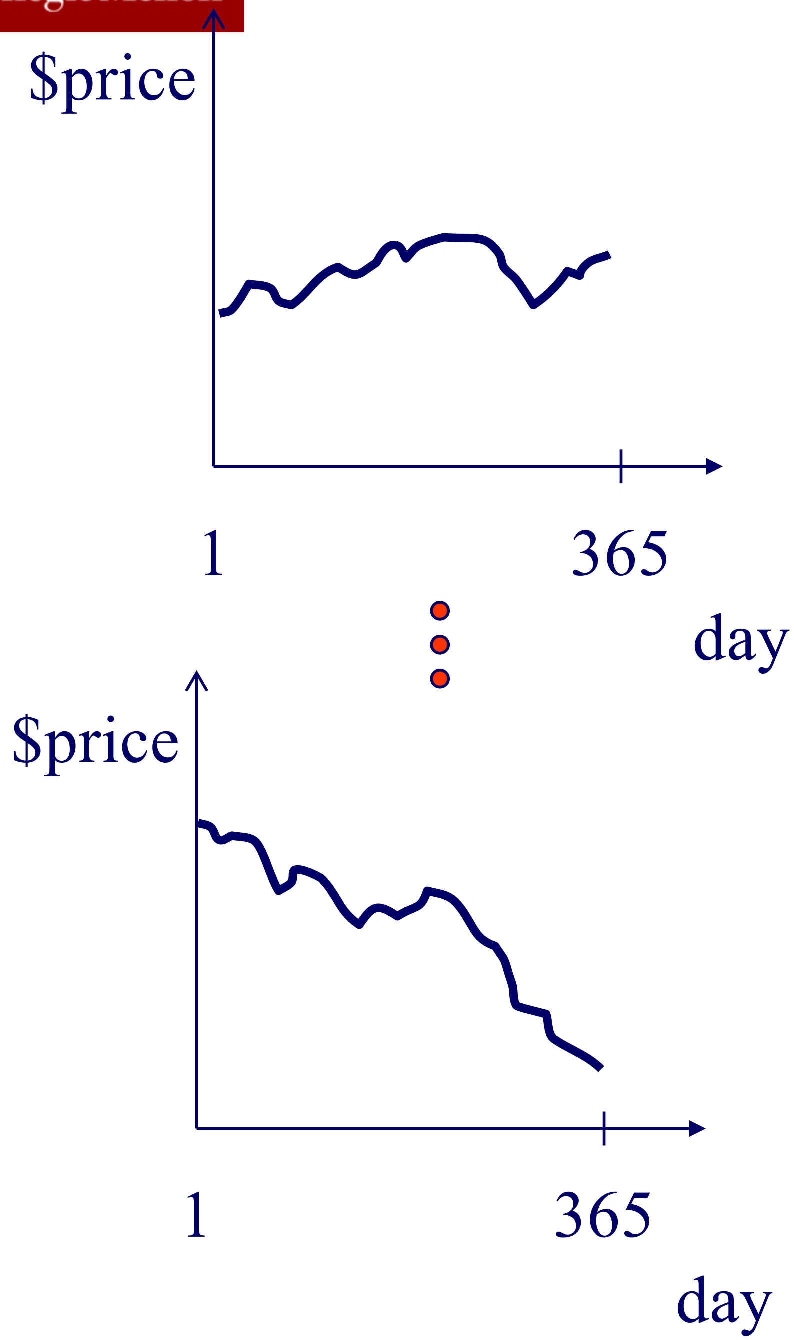
Q: find quickly stocks like ‘C’ (or customers like ‘smith’)

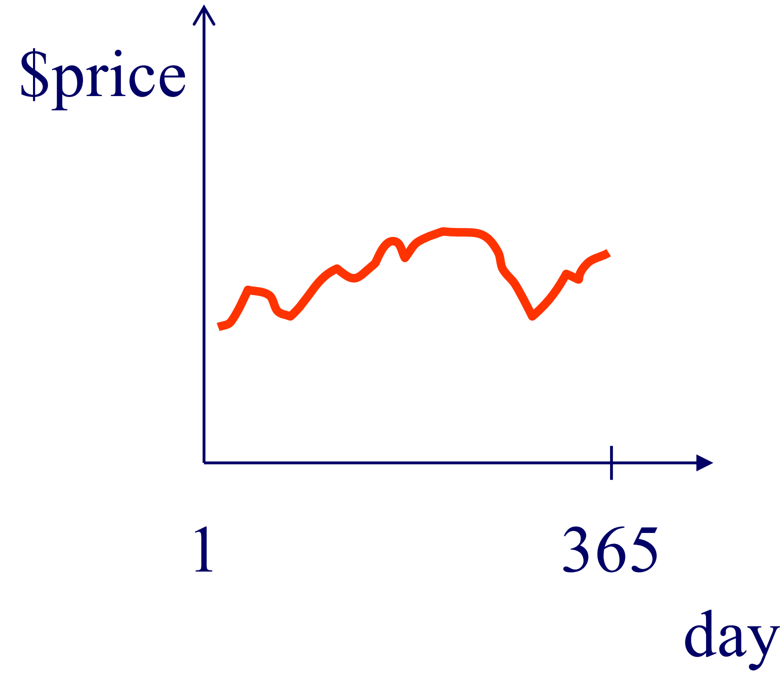
A: summarize seq. to a few numbers/features (eg., avg, stdv, Fourier coeff.)











distance function: by expert

Idea: 'GEMINI'

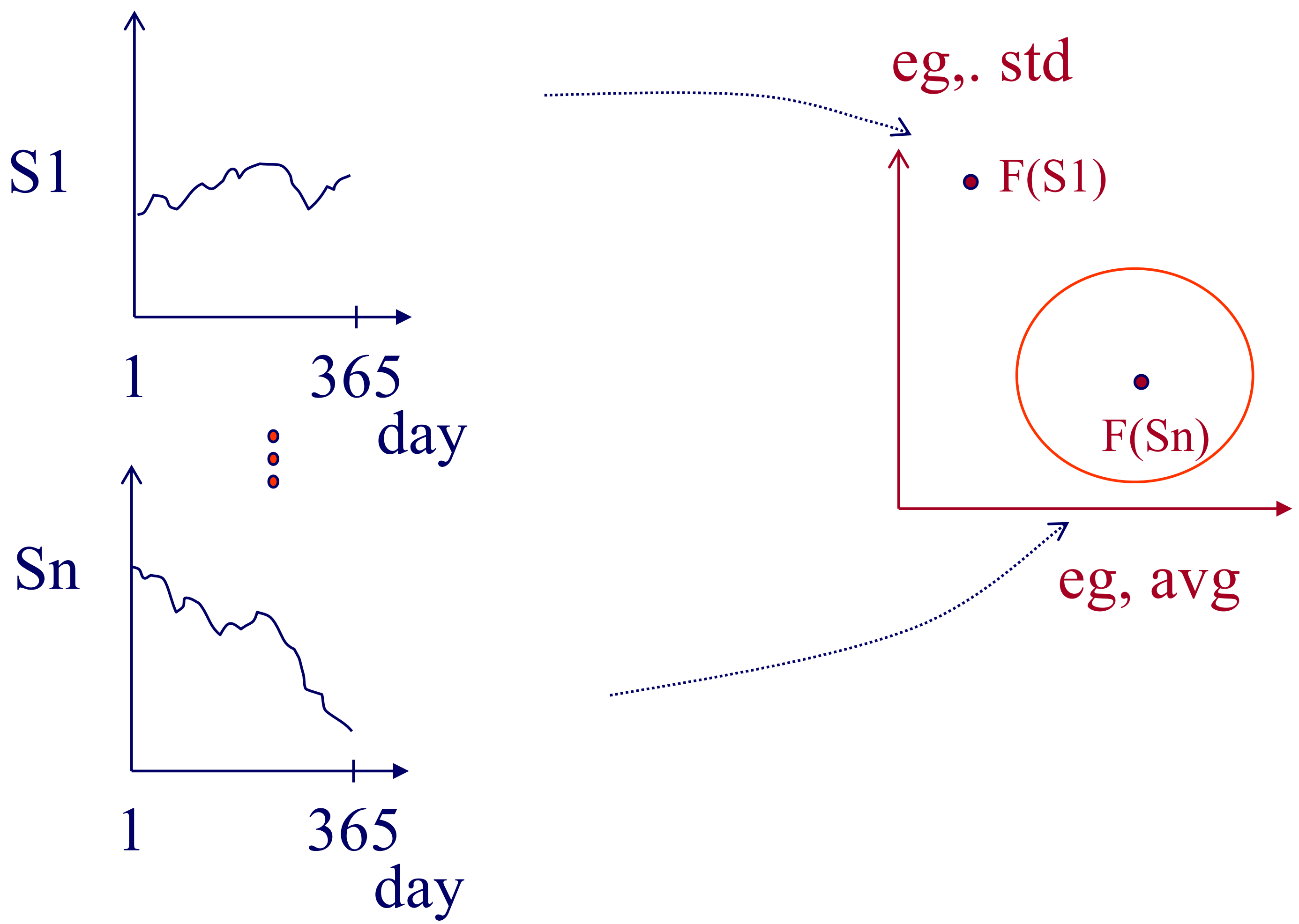
Eg., '*find stocks similar to MSFT*'

Seq. scanning: too slow

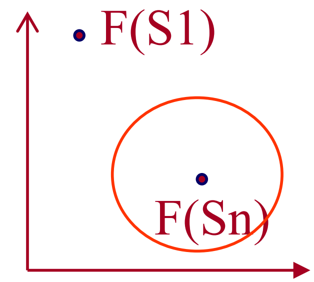
How to accelerate the search?

[Faloutsos96]

'GEMINI' - Pictorially



GEMINI

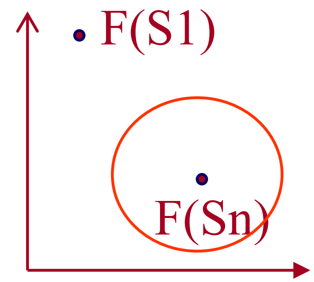


Solution: Quick-and-dirty' filter:

- extract *n* features (numbers, eg., avg., etc.)
- map into a point in *n*-d feature space
- organize points with off-the-shelf spatial access method ('SAM')
- discard false alarms

Carnegie Mellon

GEMINI



Solution: Quick-and-dirty' filter:

- extract *n* features (numbers, eg., avg., etc.)
- map into a point in *n*-d feature space
  - = feature extraction
  - = feature engineering
  - = dimensionality reduction
    - Singular value decomposition (SVD)
    - Independent Component Analysis (ICA)
  - = embedding (eg, with Deep Learning)

Examples of GEMINI

- Time sequences: DFT (up to 100 times faster) [Faloutsos, 1996]

Conclusions

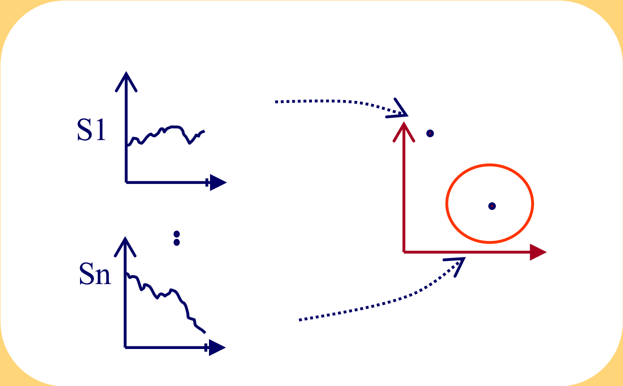
- Fast indexing: through GEMINI
  - feature extraction and
  - (off the shelf) Spatial Access Methods [Gaede+'98], or nearest-neighbor methods (Locality Sensitive Hashing LSH [Andoni+'08])

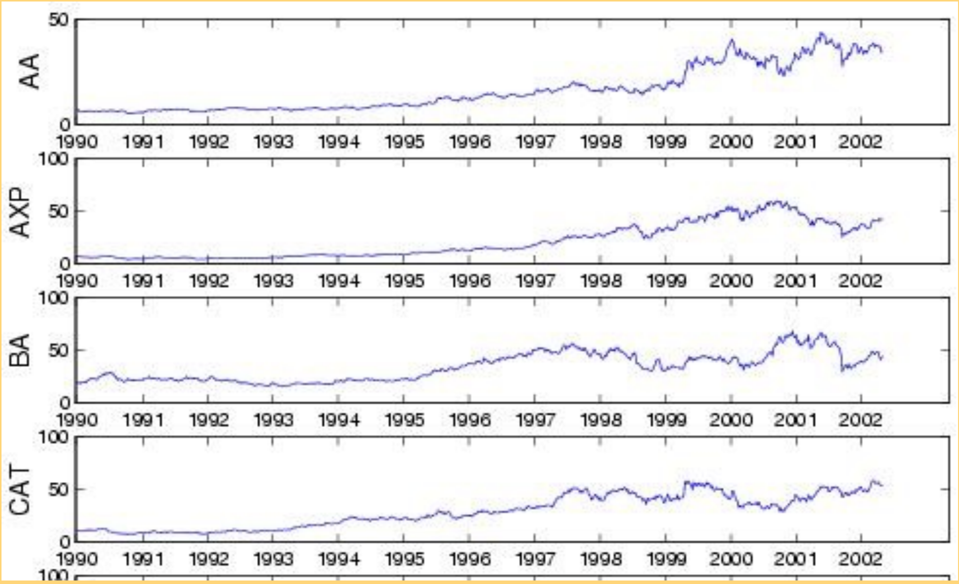
Carnegie Mellon

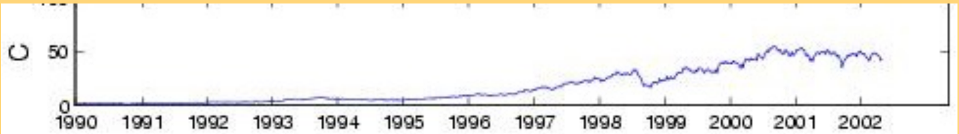
Answer:

Q: find quickly stocks like ‘C’ (or customers like ‘smith’)

A: summarize seq. to a few numbers/features (eg., avg, stdv, Fourier coeff.)







Books

- William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery: *Numerical Recipes in C*, Cambridge University Press, 1992, 2nd Edition. (Great description, intuition and code for SVD)
- ★ C. Faloutsos: *Searching Multimedia Databases by Content*, Kluwer Academic Press, 1996 (introduction to SVD, Fourier, Wavelets, and GEMINI)

References

- Gaede, V. and O. Guenther (1998). "Multidimensional Access Methods." Computing Surveys 30(2): 170-231.
- Alexandr Andoni, Piotr Indyk (2008). "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions", CACM 51, 1, 2008 https://doi.org/10.1145/1327452.1327494

References

- Oppenheim, I. J., A. Jain, et al. (March 2002). A MEMS Ultrasonic Transducer for Resident Monitoring of Steel Structures. SPIE Smart Structures Conference SS05, San Diego.

Carnegie Mellon

Part 2: DSP (Digital Signal Processing)

Carnegie Mellon

Outline

- Motivation
- P1. Similarity Search and Indexing
  - P1.1. distance functions: Euclidean;Time-warping
  - P1.2. indexing
  - P1.3. feature extraction
- P2. DSP (Digital Signal Processing)
- ...

Outline

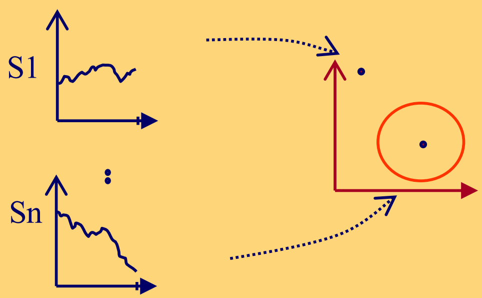
- Motivation
- P1. Similarity Search and Indexing
- P2. DSP (Digital Signal Processing)
  - – P2.1. Discrete Fourier Transform (DFT)
  - – P2.2. Discrete Wavelet Transform (DWT)
- P3. Linear Forecasting
- P4. Non-linear forecasting
- Conclusions

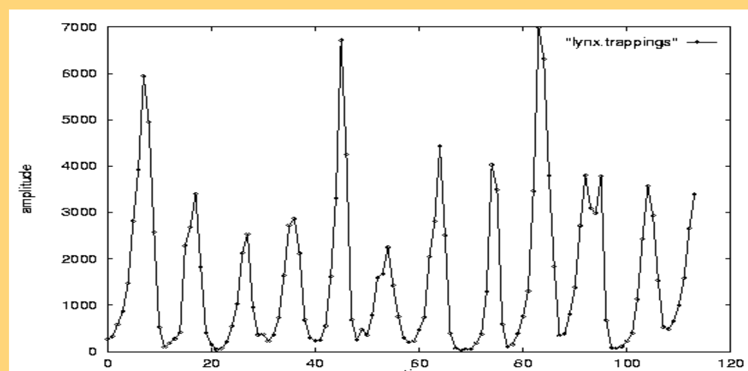
Detailed Outline

- P1. Indexing etc
- P2. DSP (Digital Signal Processing)
  - – P2.1. DFT
    - Definition of DFT and properties
    - how to read the DFT spectrum
  - – P2.2. DWT
    - Definition of DWT and properties
    - how to read the DWT scalogram

Problem

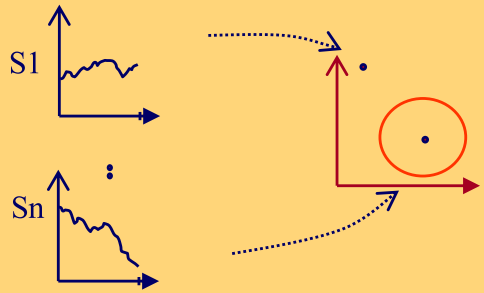
Q: How to summarize / extract few features

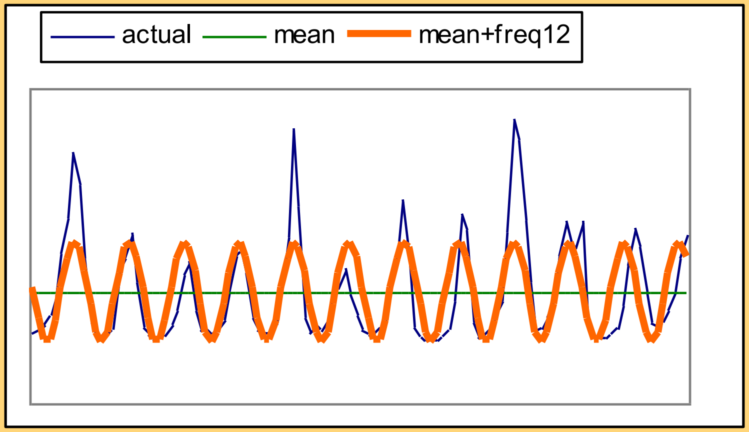




Answer:

Q: How to summarize / extract few features
A: Fourier; Wavelets
(A': SVD Singular Value Decomposition,
ICA = Independent Component Analysis)

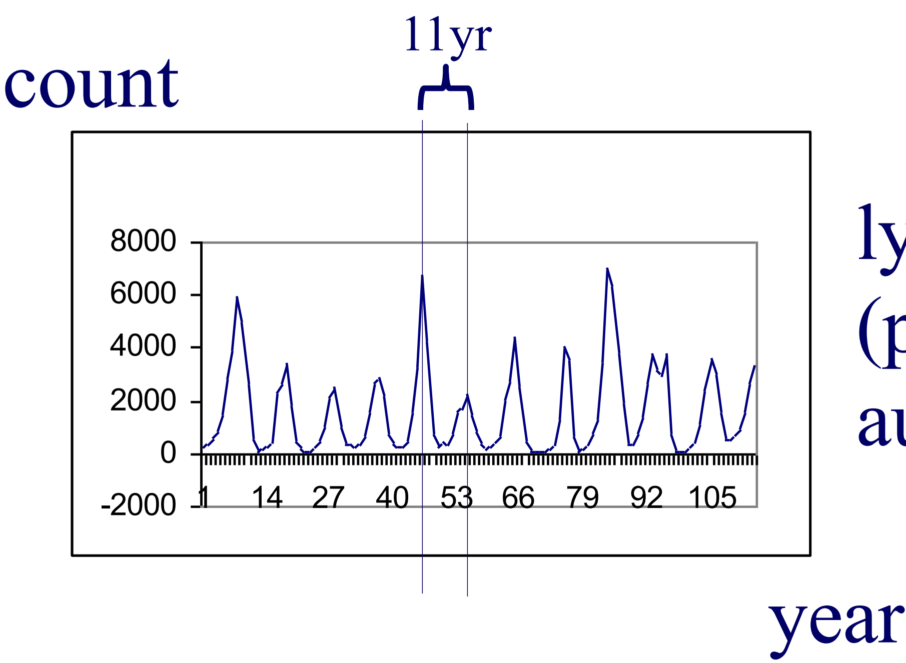




Introduction - Problem#2

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

Find: patterns and/or compress

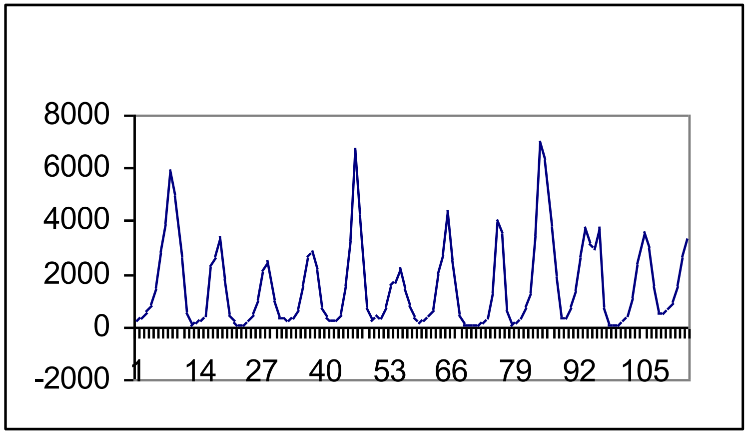


lynx caught per year
(packets per day;
automobiles per hour)

CarnegieMellon

Problem #2: patterns?

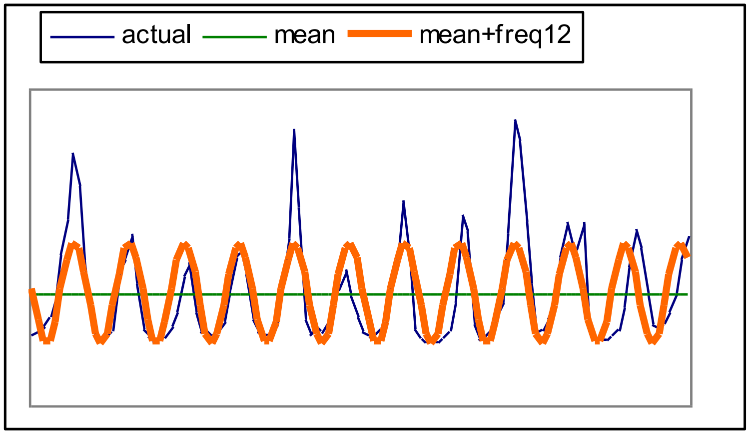
Q: How to automatically find patterns?



Problem #2: patterns?

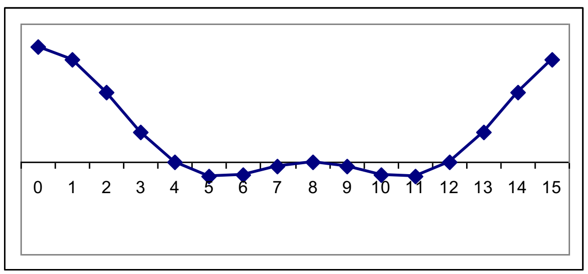
Q: How to automatically find patterns?

A: Discrete Fourier Transform (DFT)

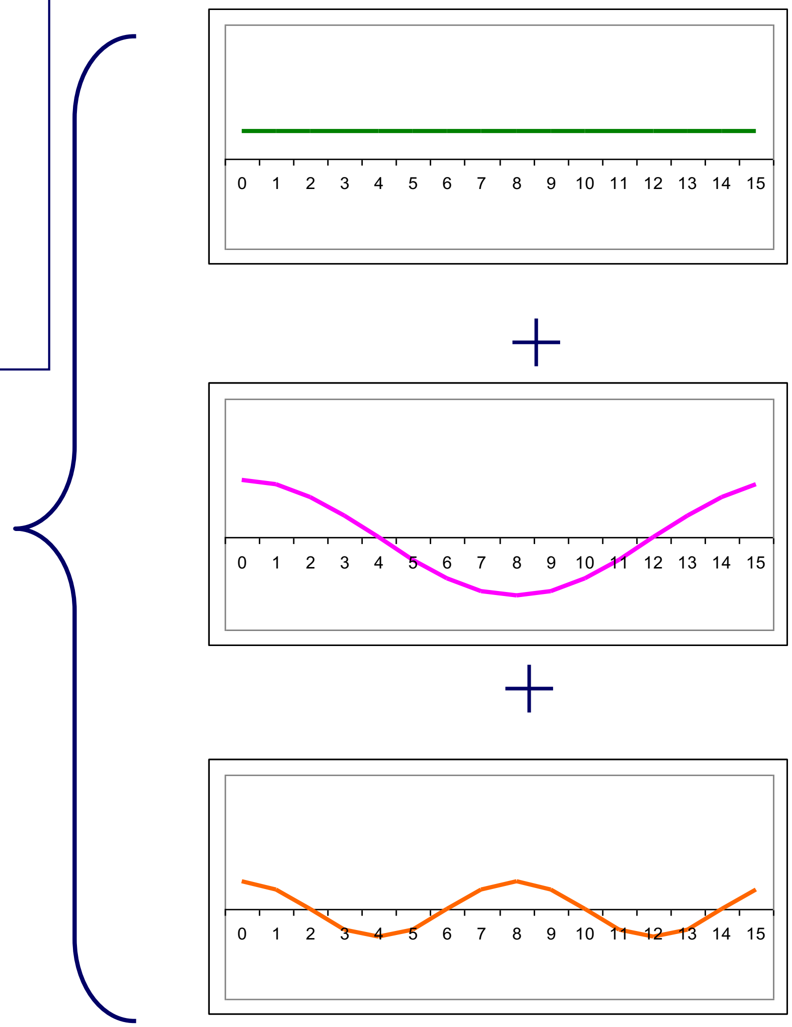


DFT: finds sinusoids

Given a signal,
DFT decomposes it
to sinusoids



=



Details

DFT: definition

- For a sequence $x_0, x_1, \ldots x_{n-1}$
- the (**n-point**) Discrete Fourier Transform is
- $X_0, X_1, \ldots X_{n-1}$:

$$X_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \exp(-j 2\pi \, tf / n) \qquad f = 0, \ldots, n-1$$

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

$$x_t = 1/\sqrt{n} \sum_{t=0}^{n-1} X_f * \exp(+j 2\pi \, tf / n)$$

inverse DFT

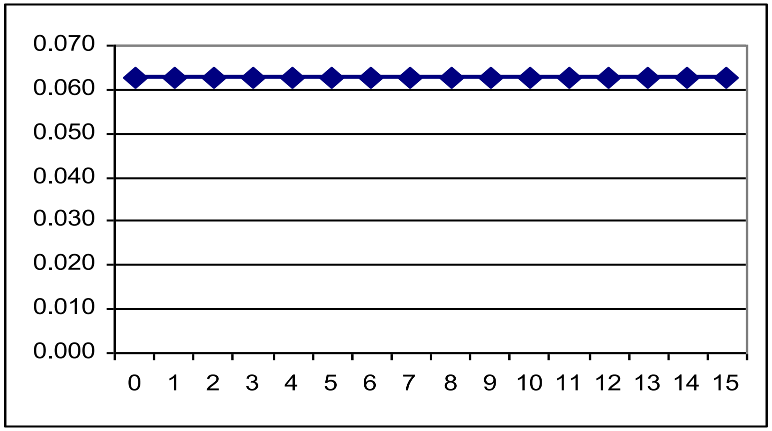
DFT: examples

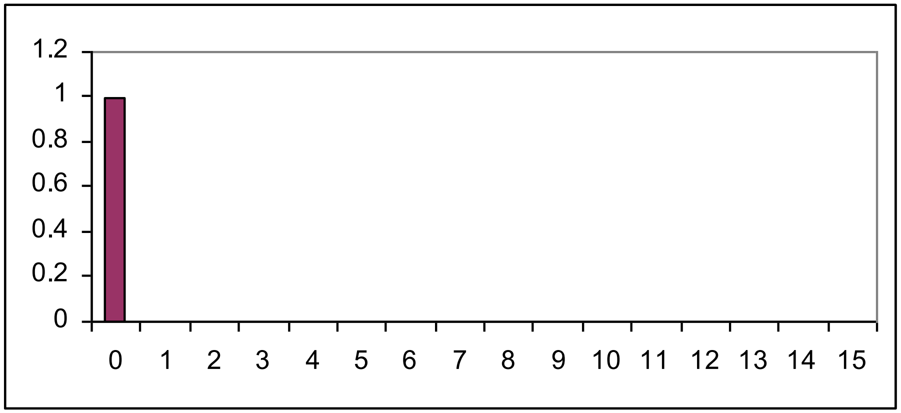
flat

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

Plot[Abs[Fourier[x]]];

Amplitude



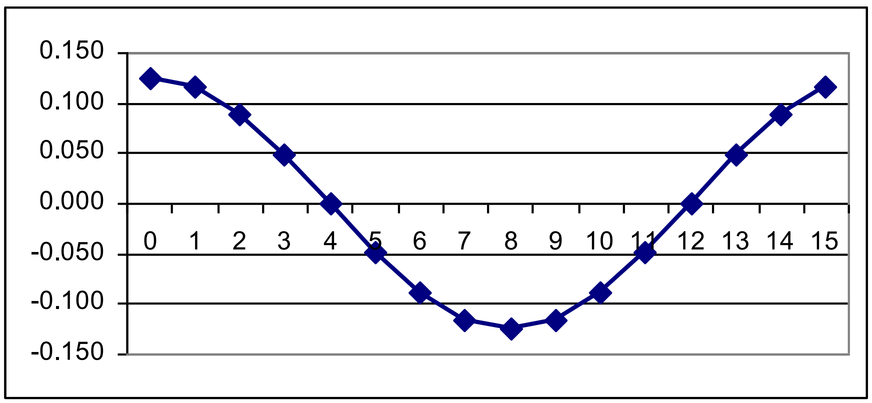


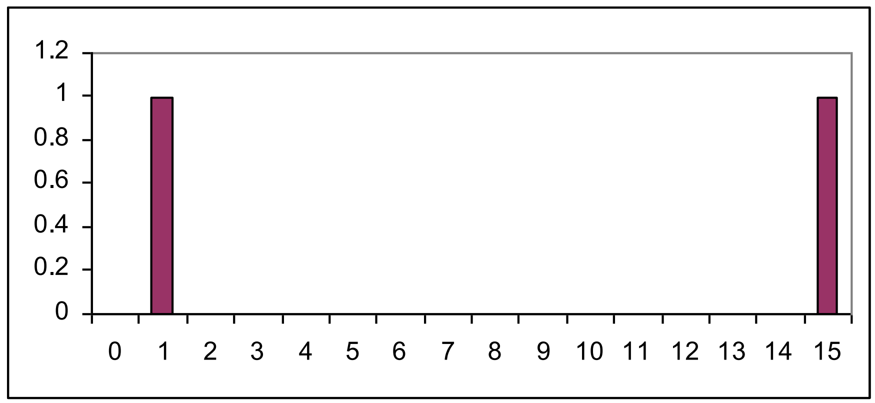
time

freq

DFT: examples

Low frequency sinusoid



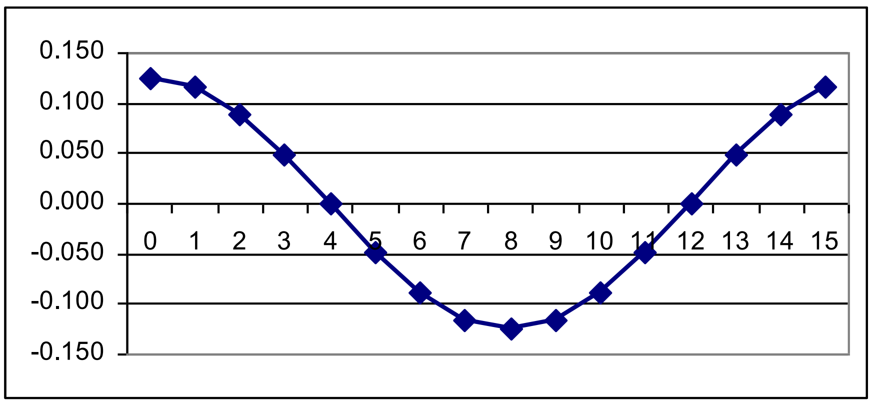


time

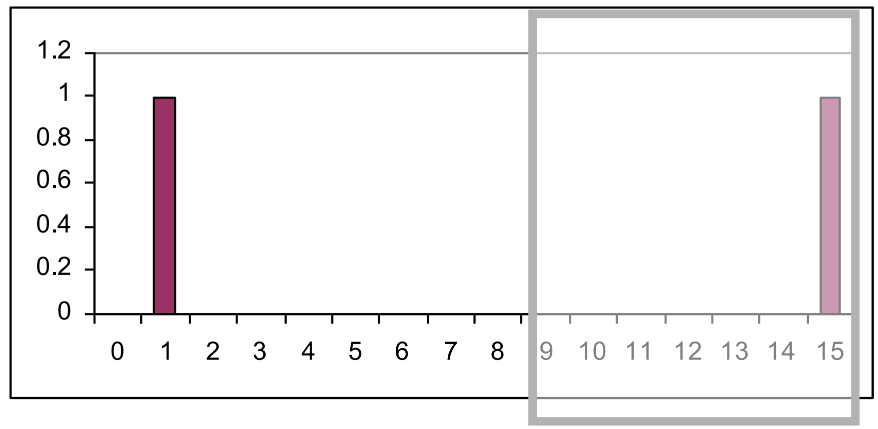
freq

DFT: examples

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



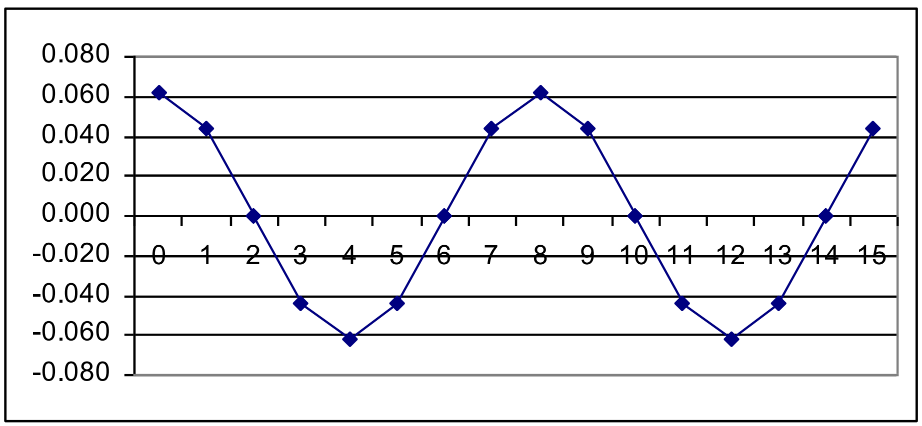
time



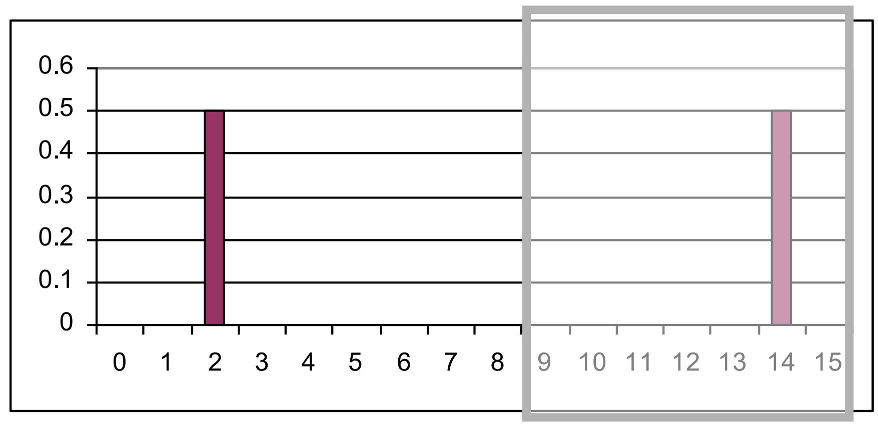
freq

DFT: examples

- Higher freq. sinusoid



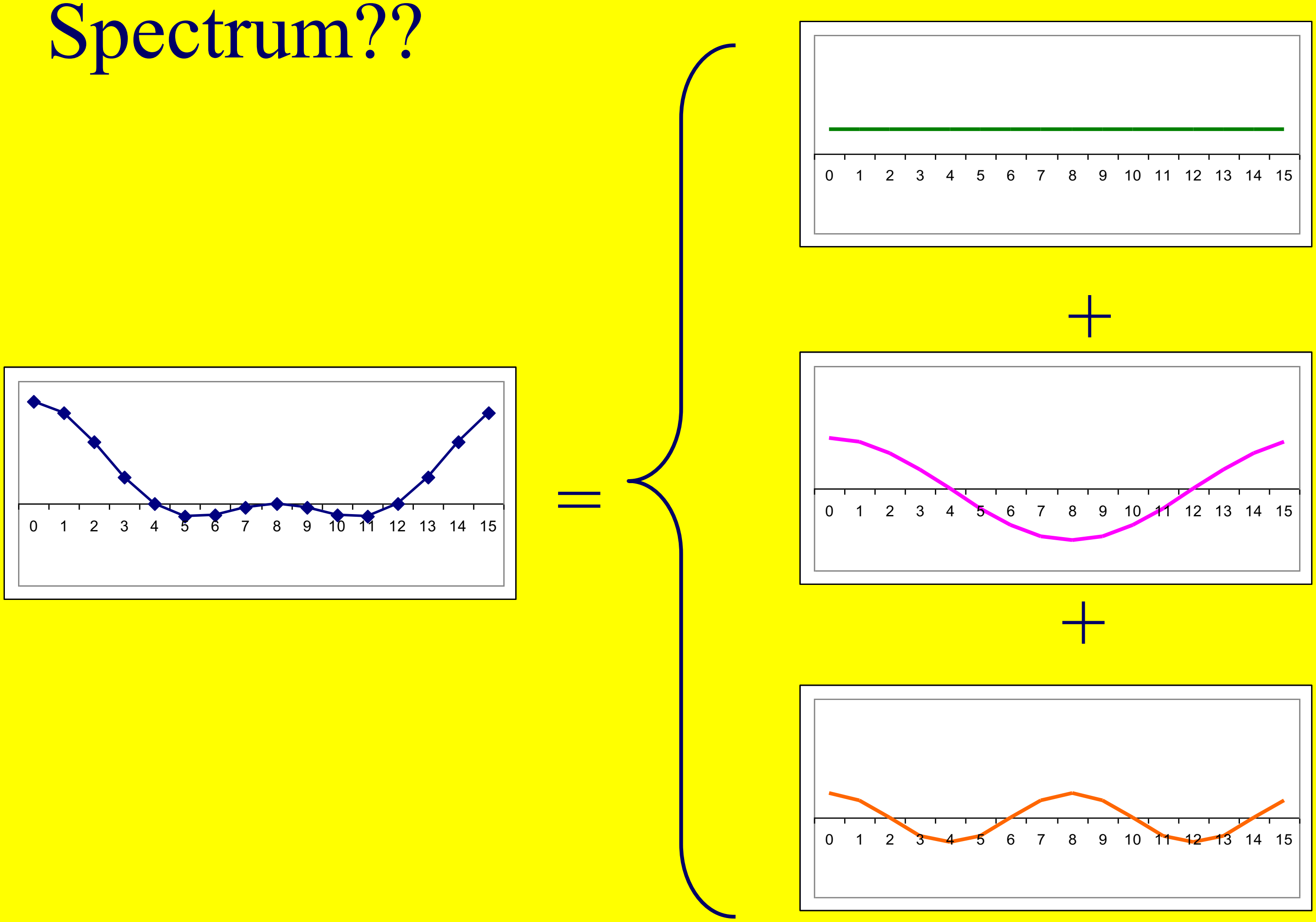
time



freq

DFT: check-point

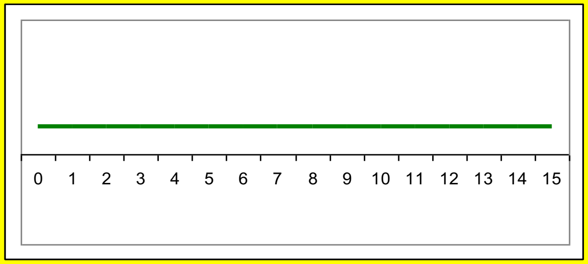
Spectrum??



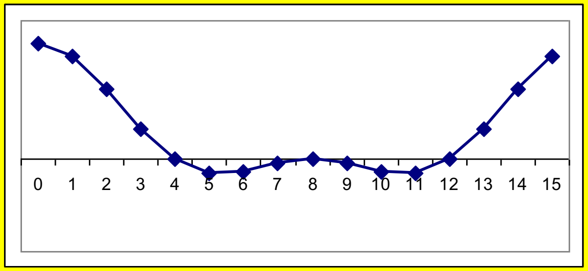
Carnegie Mellon

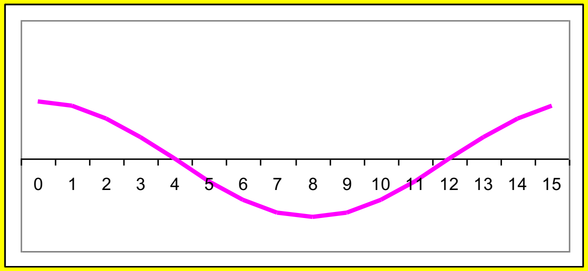
DFT: check-point

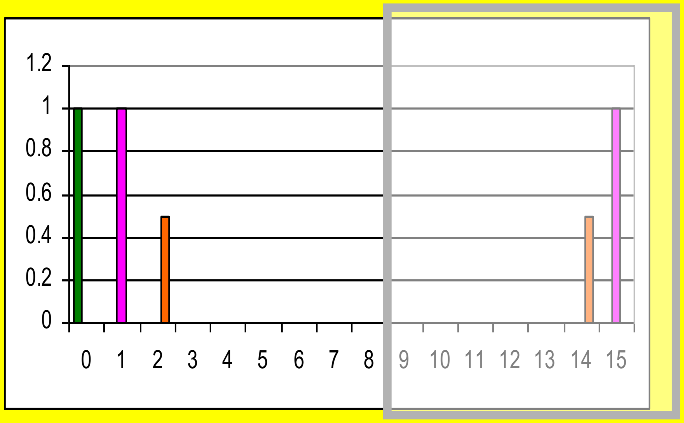
Spectrum:

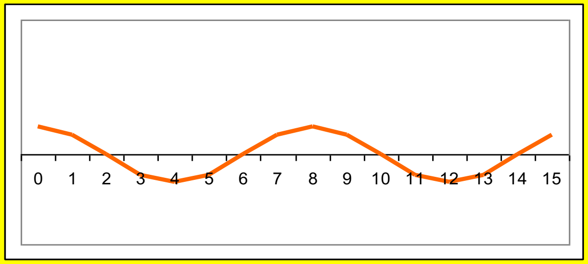


Ampl.







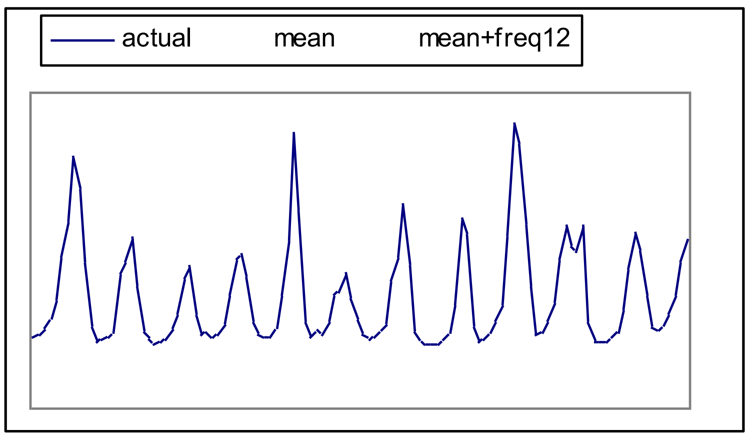


Freq.

DFT: Amplitude spectrum

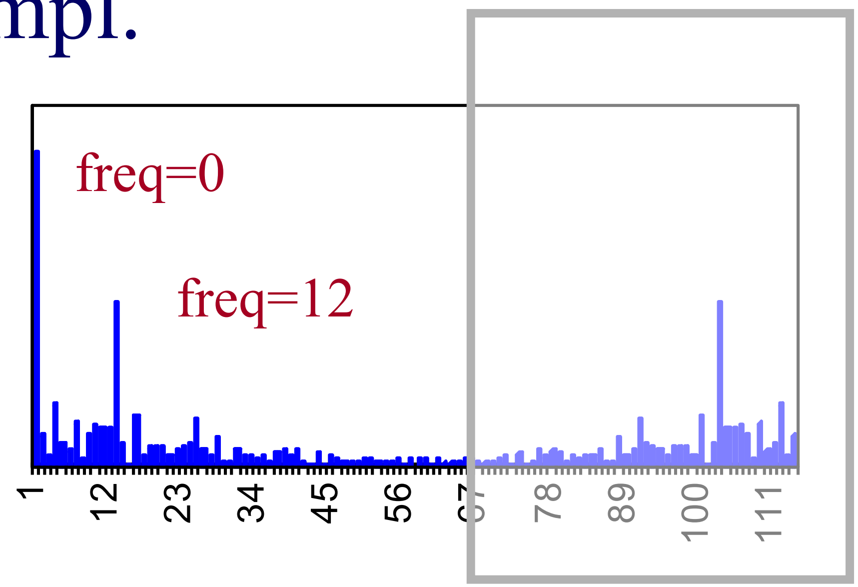
Amplitude: $$A_f^{\,2} = \mathrm{Re}^2(X_f) + \mathrm{Im}^2(X_f)$$

count



year

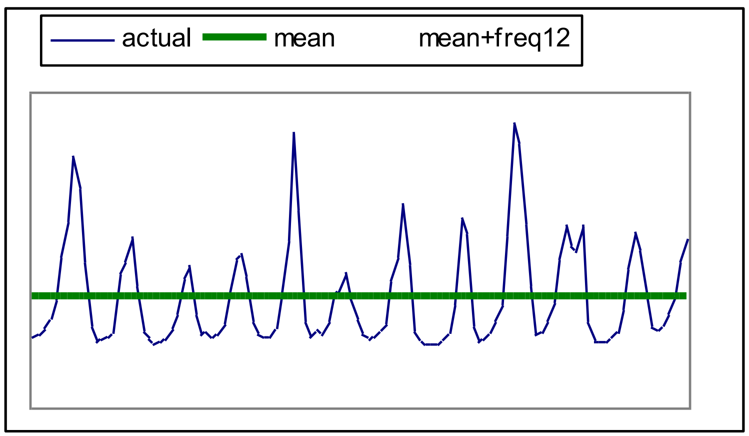
Ampl.



Freq.

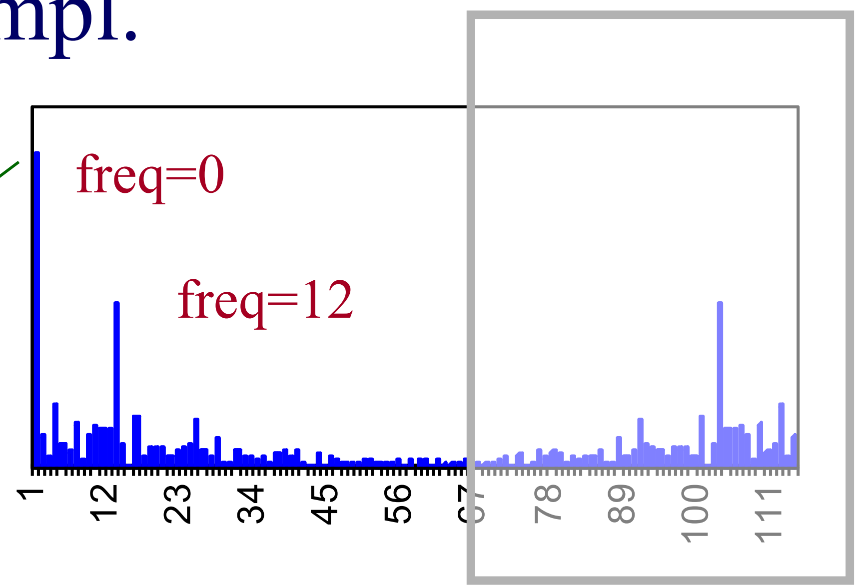
DFT: Amplitude spectrum

count



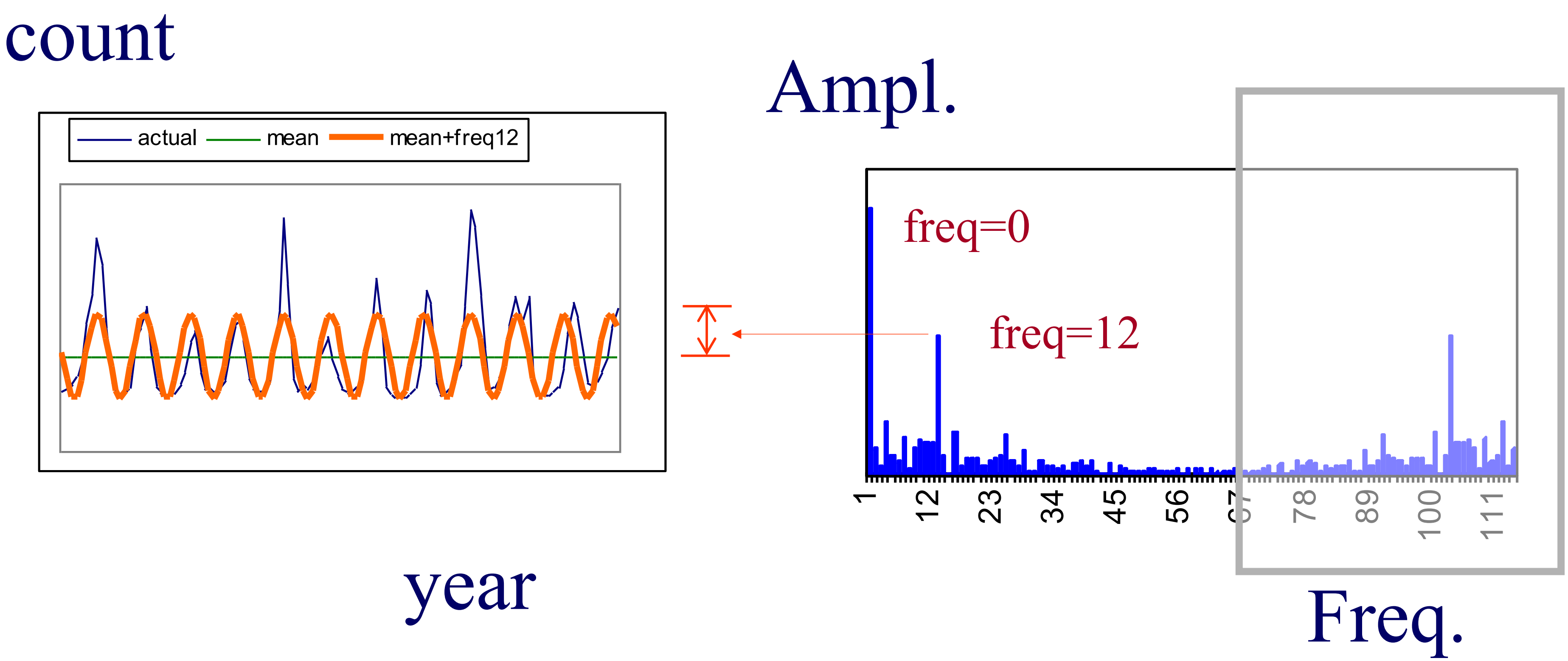
year

Ampl.



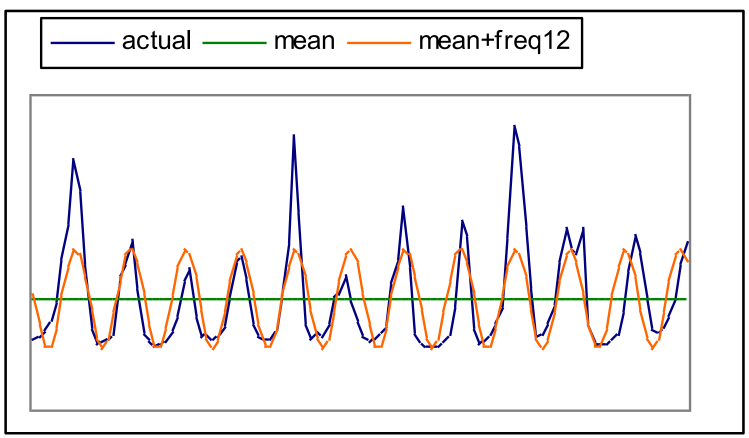
Freq.

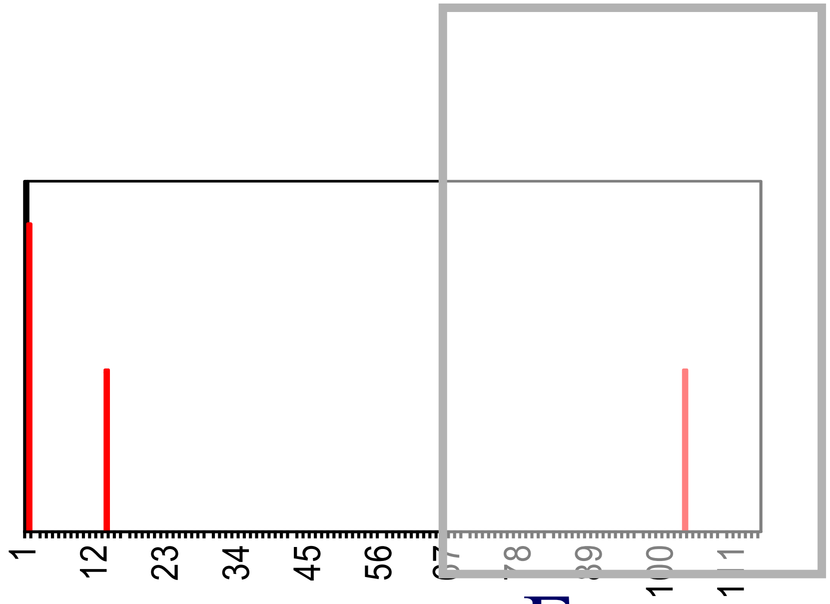
DFT: Amplitude spectrum



DFT: Amplitude spectrum

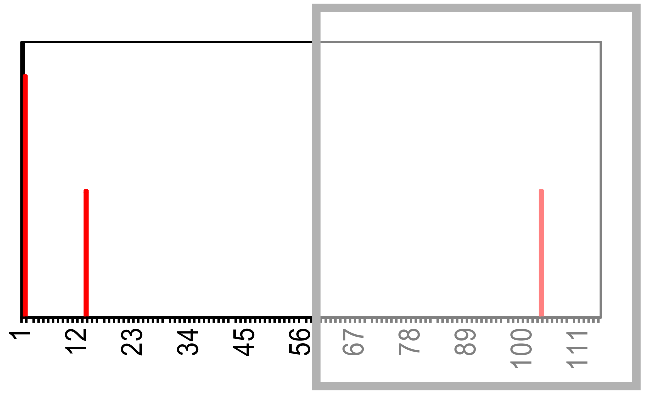
- excellent approximation, with only 2 frequencies!
- so what?





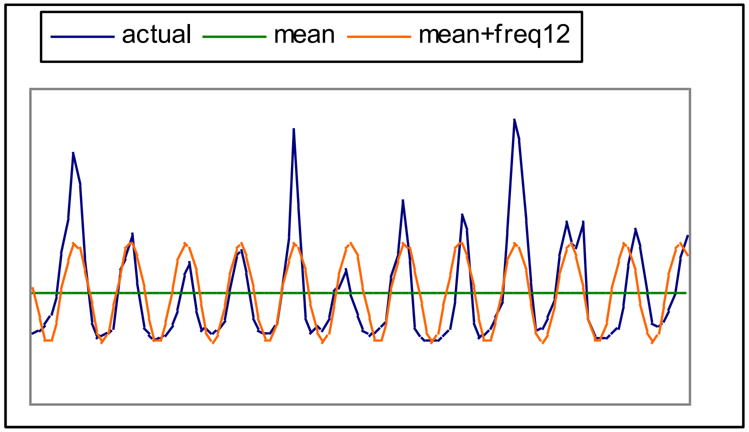
DFT: Amplitude spectrum

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

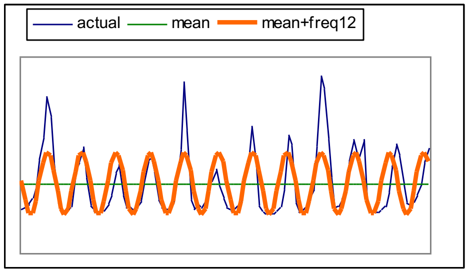


DFT: Amplitude spectrum

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



DFT - Conclusions



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

Detailed Outline

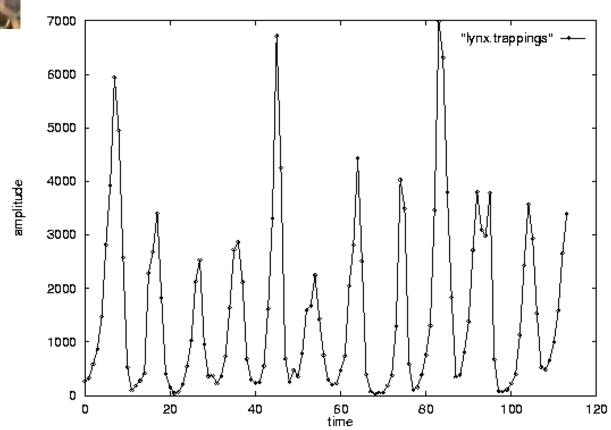
- P1. Indexing etc
- P2. DSP (Digital Signal Processing)
  - P2.1. DFT
    - Definition of DFT and properties
    - how to read the DFT spectrum
  - P2.2. DWT
    - Definition of DWT and properties
    - how to read the DWT scalogram

Problem #2:

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

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

count



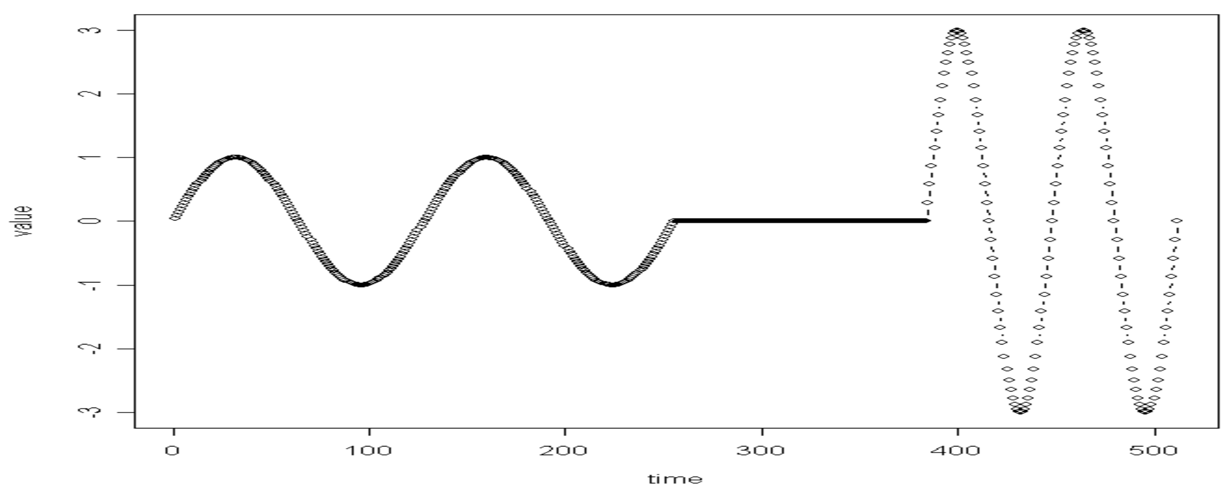
lynx caught per year
(packets per day;
virus infections per month)

year

Wavelets - DWT

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

value

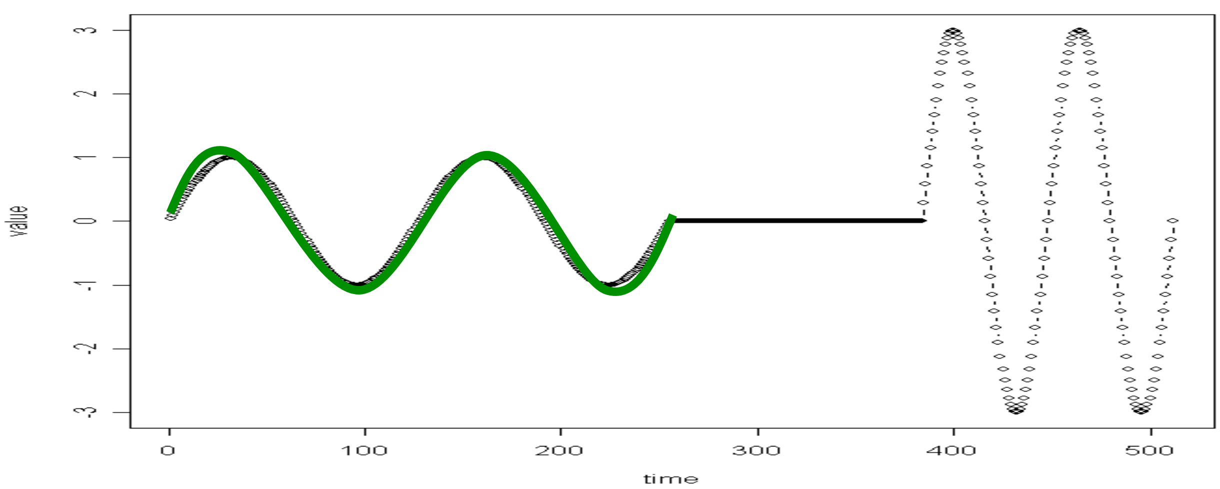


time

Wavelets - DWT

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

value



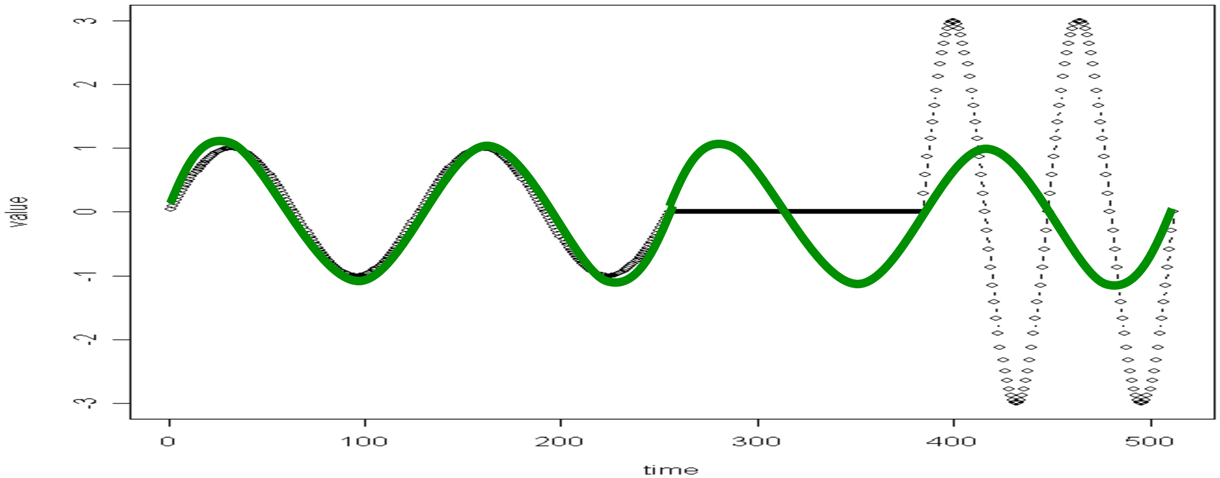
time

Carnegie Mellon

Wavelets - DWT

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

value

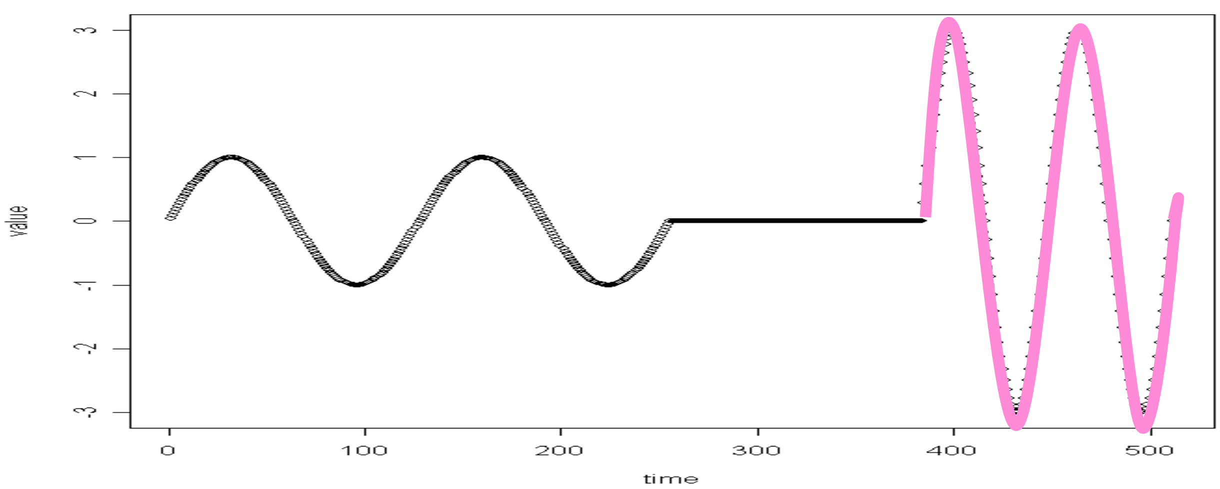


time

Wavelets - DWT

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

value

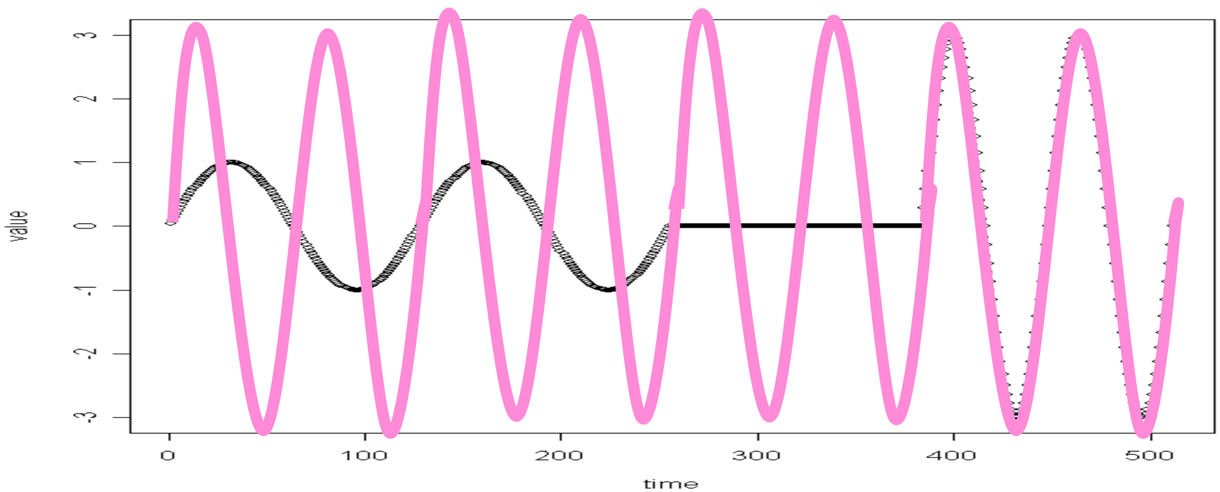


time

Wavelets - DWT

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

value

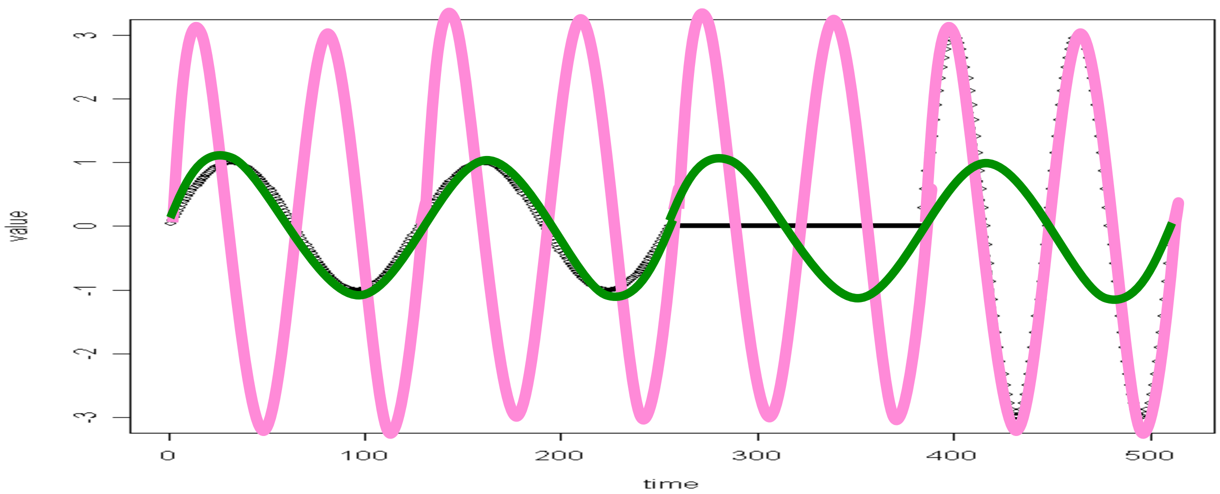


time

Wavelets - DWT

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

value

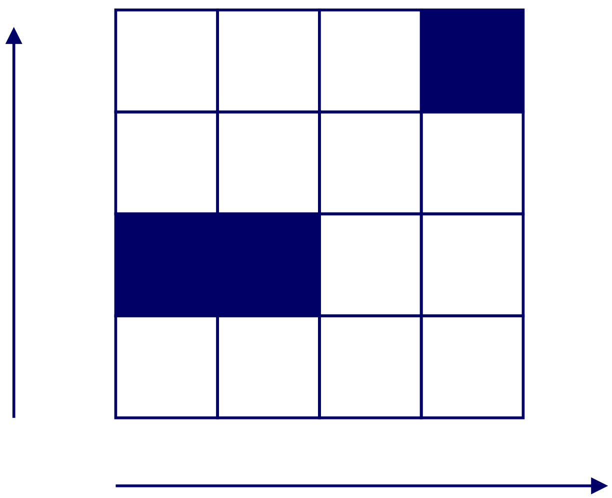


time

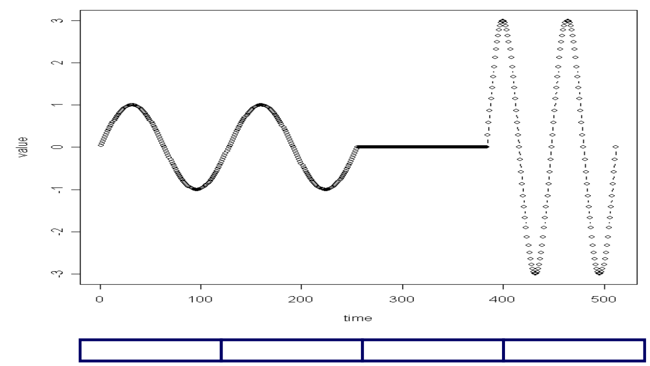
Short Window F.T.

- Solution#1: Short window Fourier transform (SWFT)

freq



value

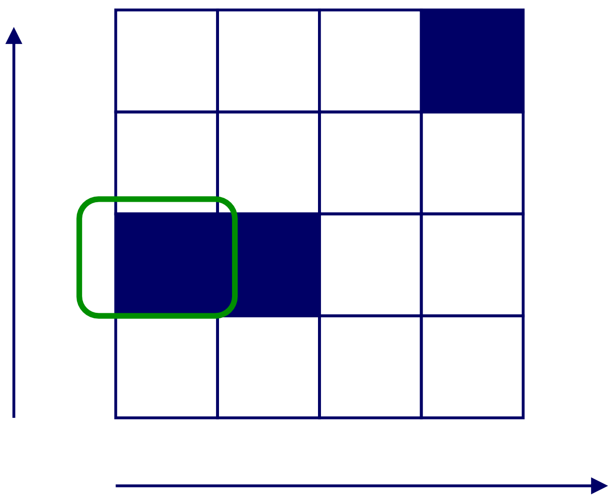


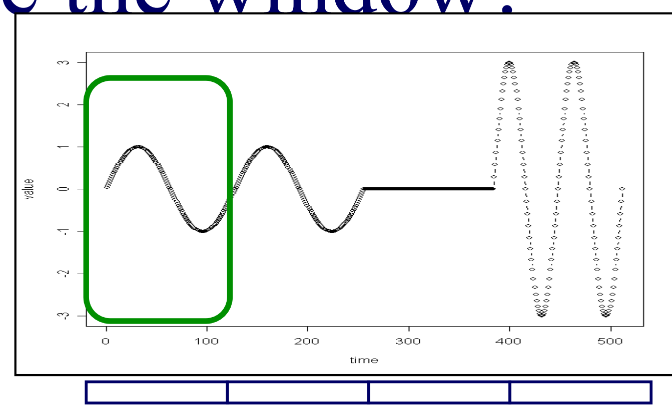
ime

time

Short Window F.T.

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

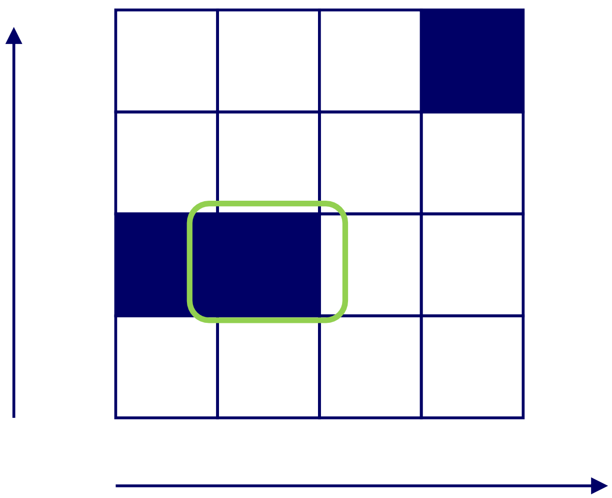




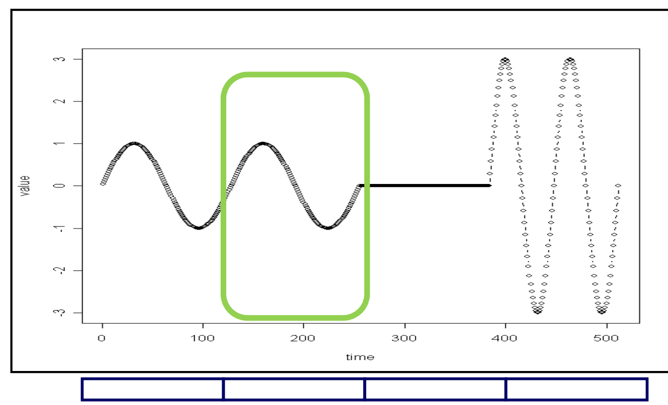
Short Window F.T.

- Solution#1: Short window Fourier transform (SWFT)

freq



value



ime

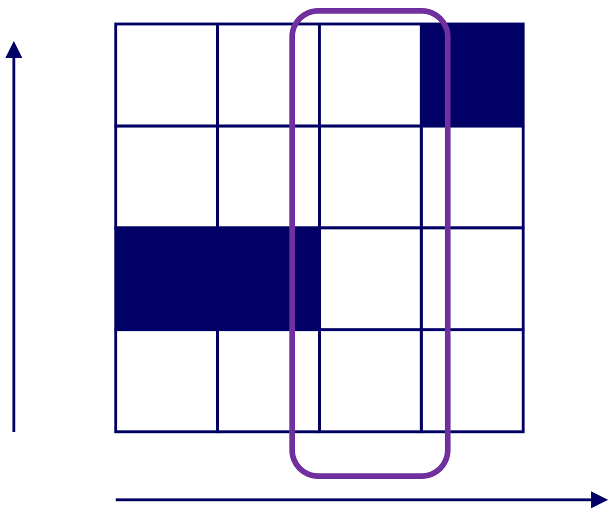
time

Carnegie Mellon

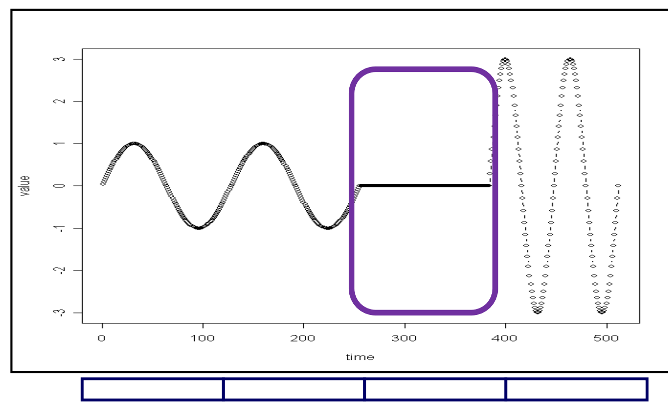
Short Window F.T.

- Solution#1: Short window Fourier transform (SWFT)

freq



value

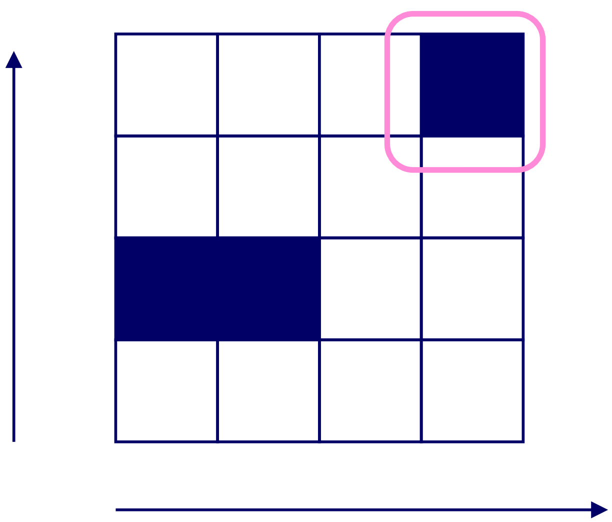


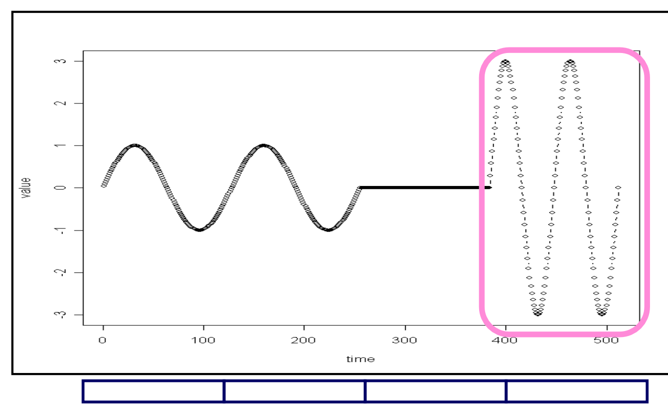
ime

time

Short Window F.T.

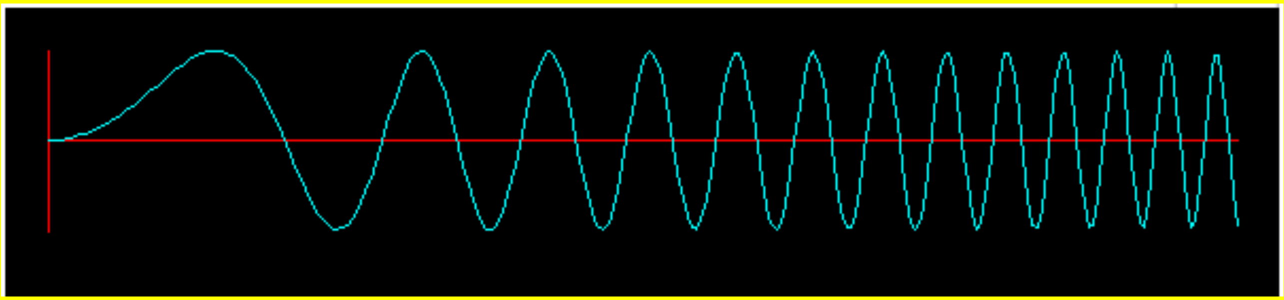
- Solution#1: Short window Fourier transform (SWFT)





Short window FT: check-point

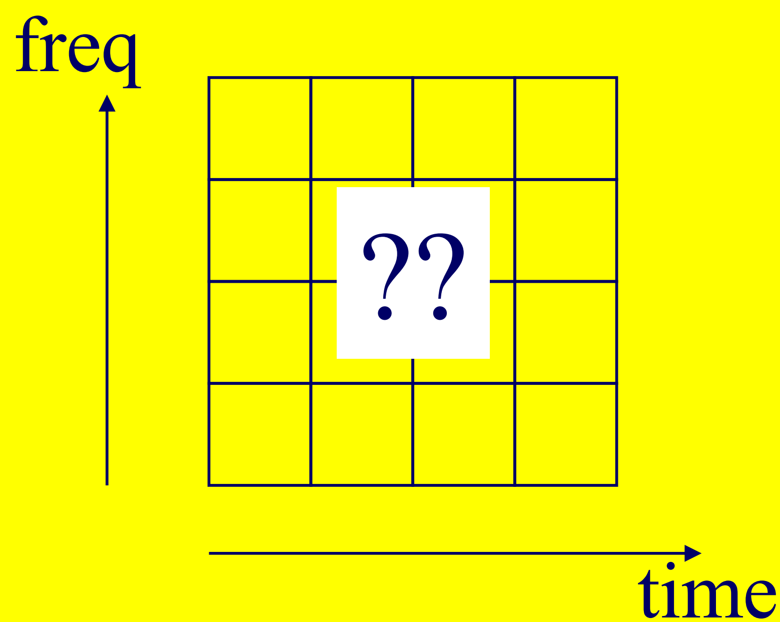
Chirp



$$x(t) = \sin (2 \pi t*t / N)$$

(usual sinusoids:

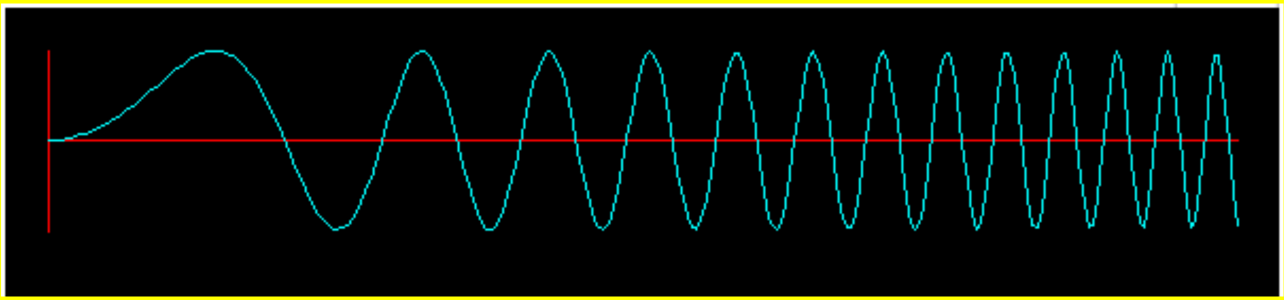
$$x(t) = \sin (2 \pi f*t / N)$$



Carnegie Mellon

Short window FT: check-point

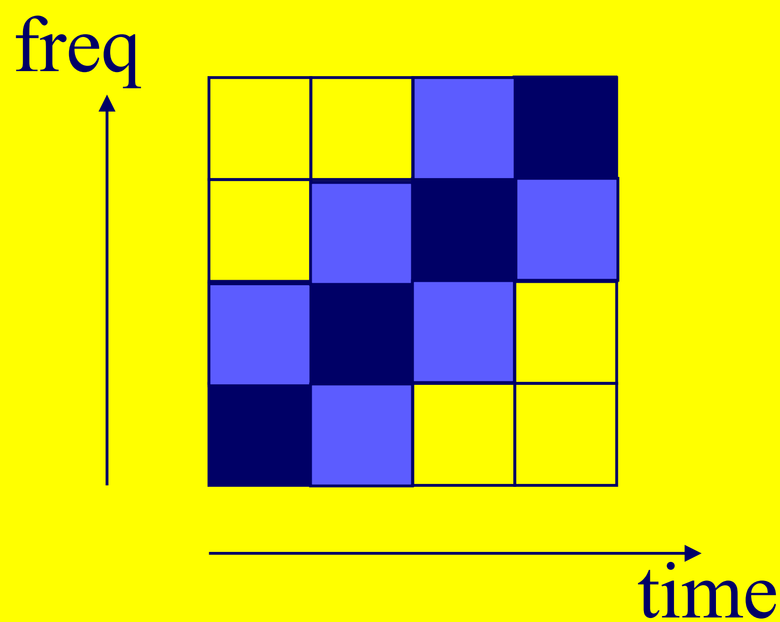
Chirp



x(t) = sin (2 pi t*t / N)

(usual sinusoids:

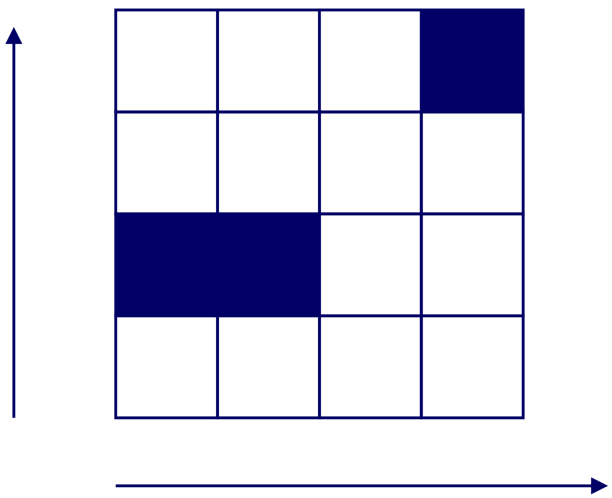
x(t) = sin (2 pi f*t / N)



Wavelets - DWT

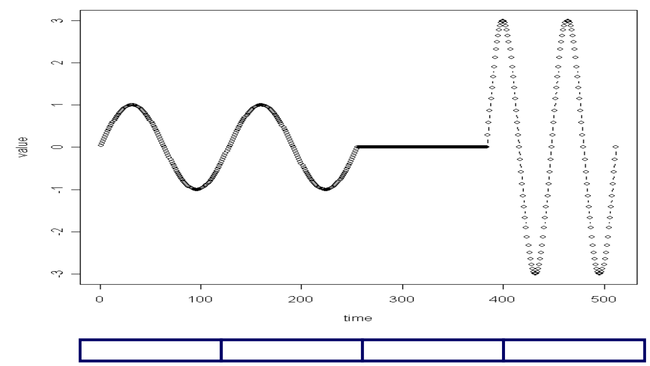
- Solution#1: Short window Fourier transform (SWFT)
- **But: how short should be the window?**

freq



time

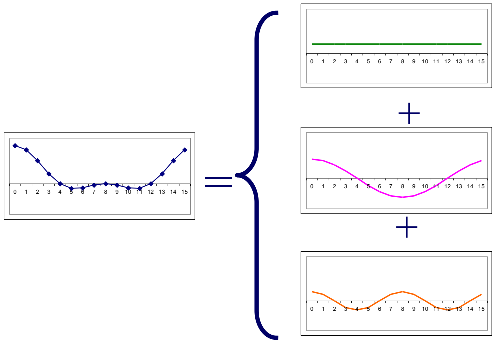
value



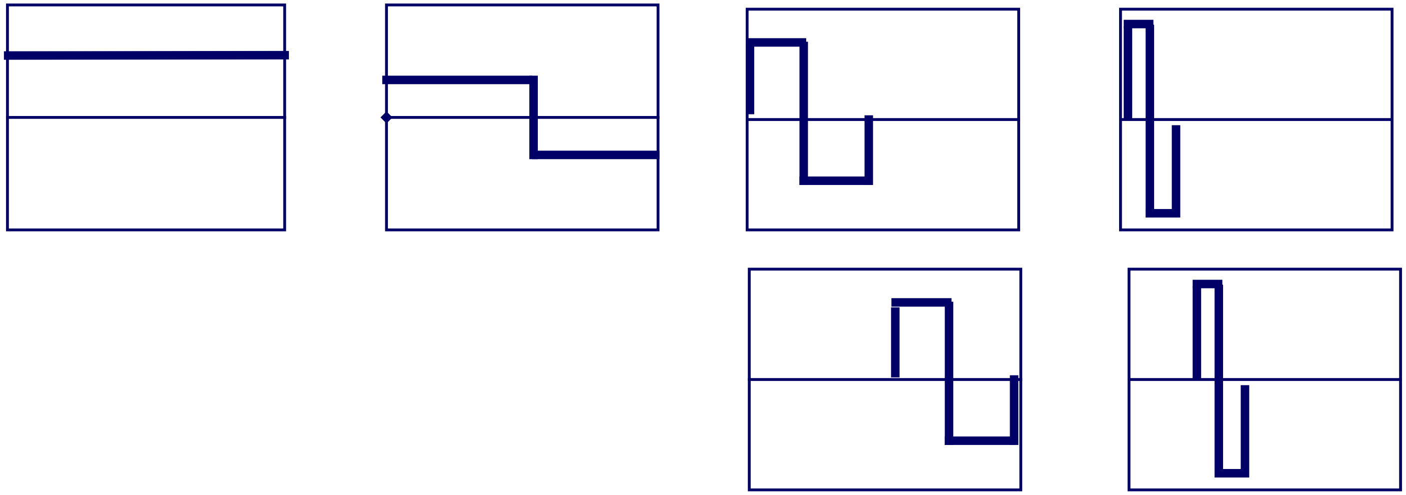
time

Carnegie Mellon

Haar Wavelets

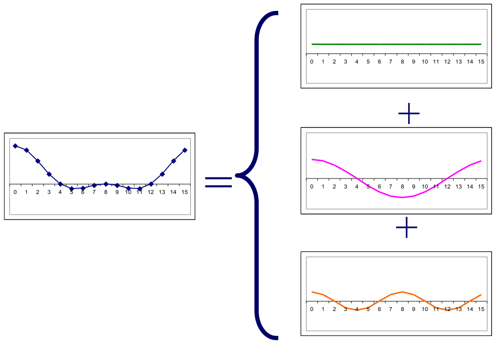


- Answer: **multiple** window sizes! -> DWT

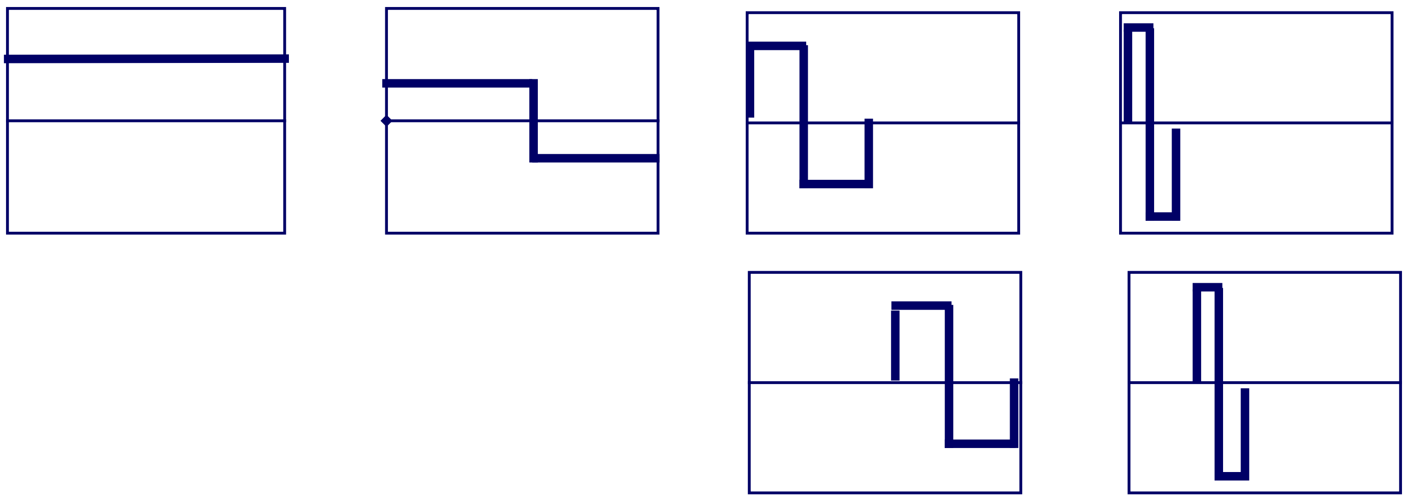


Carnegie Mellon

Haar Wavelets



- Basis functions: waves (of length 1, ½, ¼, etc)
- (vs DFT: sinusoids of full length)

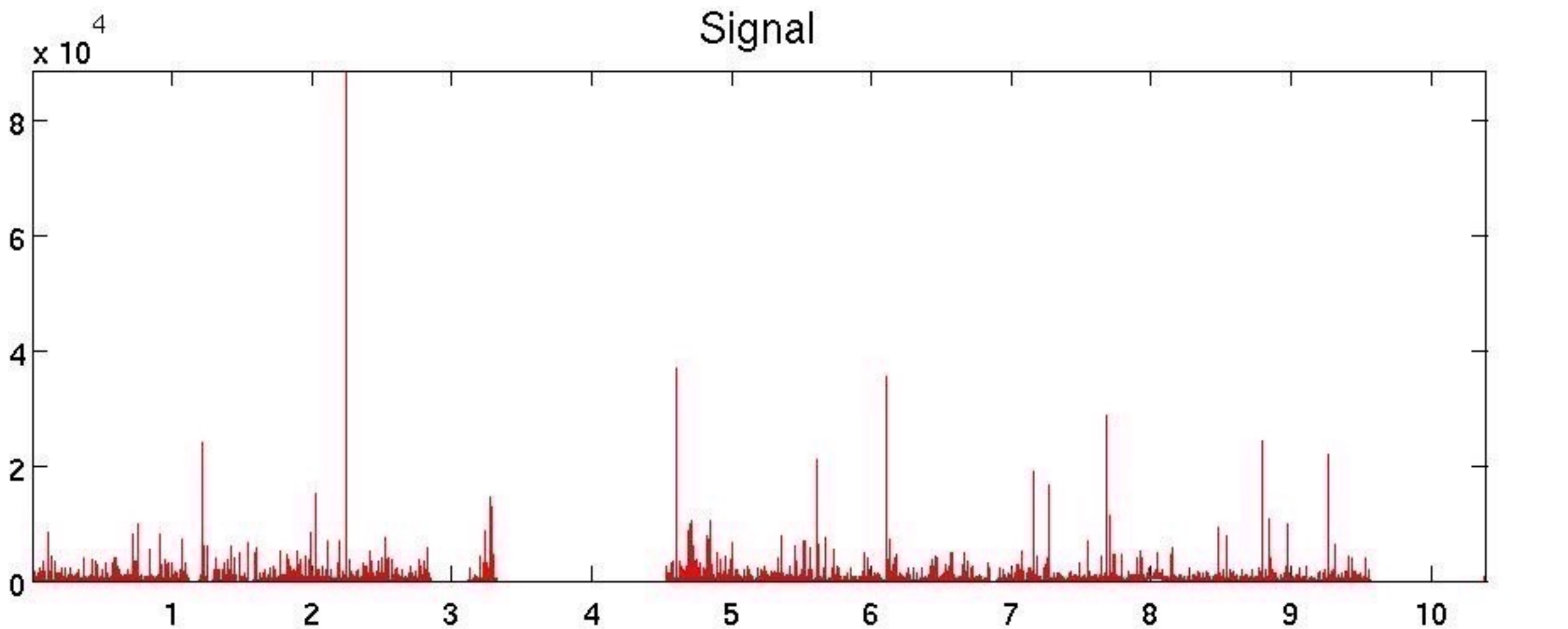


Detailed Outline

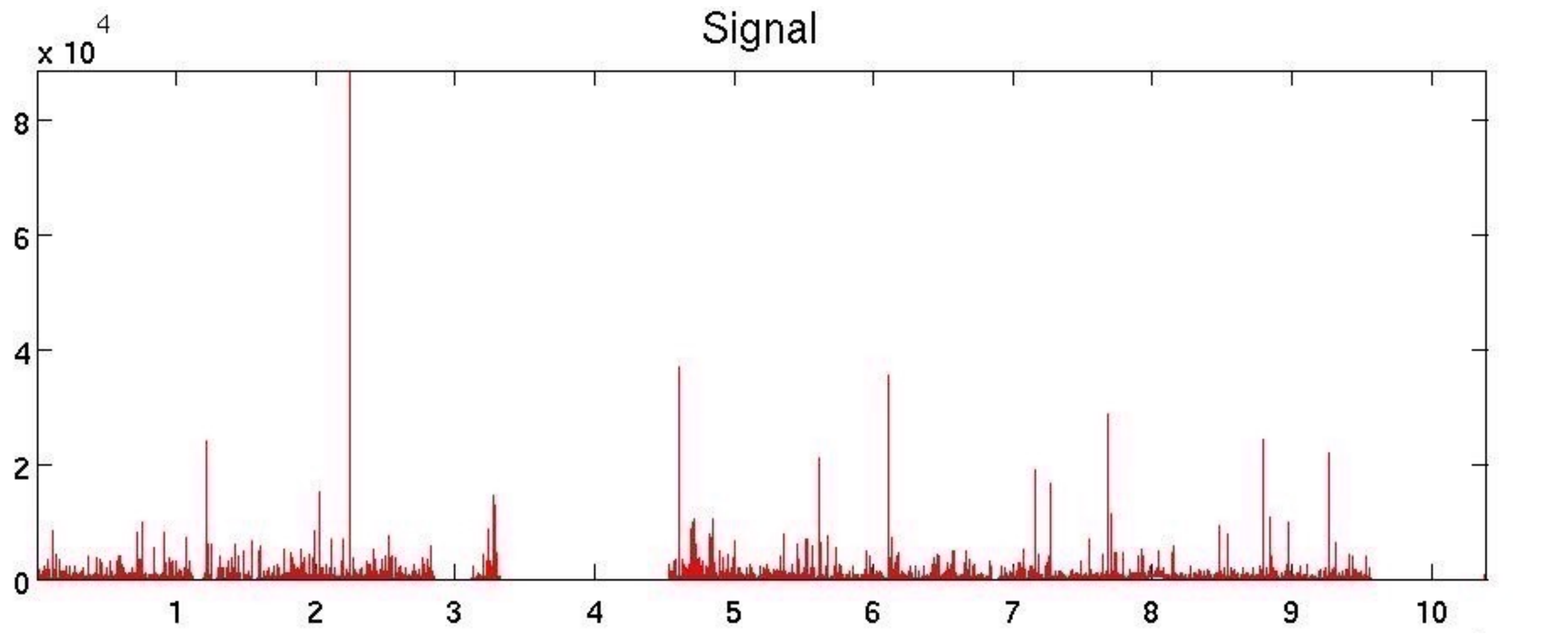
- P1. Indexing etc
- P2. DSP (Digital Signal Processing)
  - P2.1. DFT
    - Definition of DFT and properties
    - how to read the DFT spectrum
  - P2.2. DWT
    - Definition of DWT and properties
    - how to read the DWT scalogram

Wavelets in action

- # of correction packets vs time

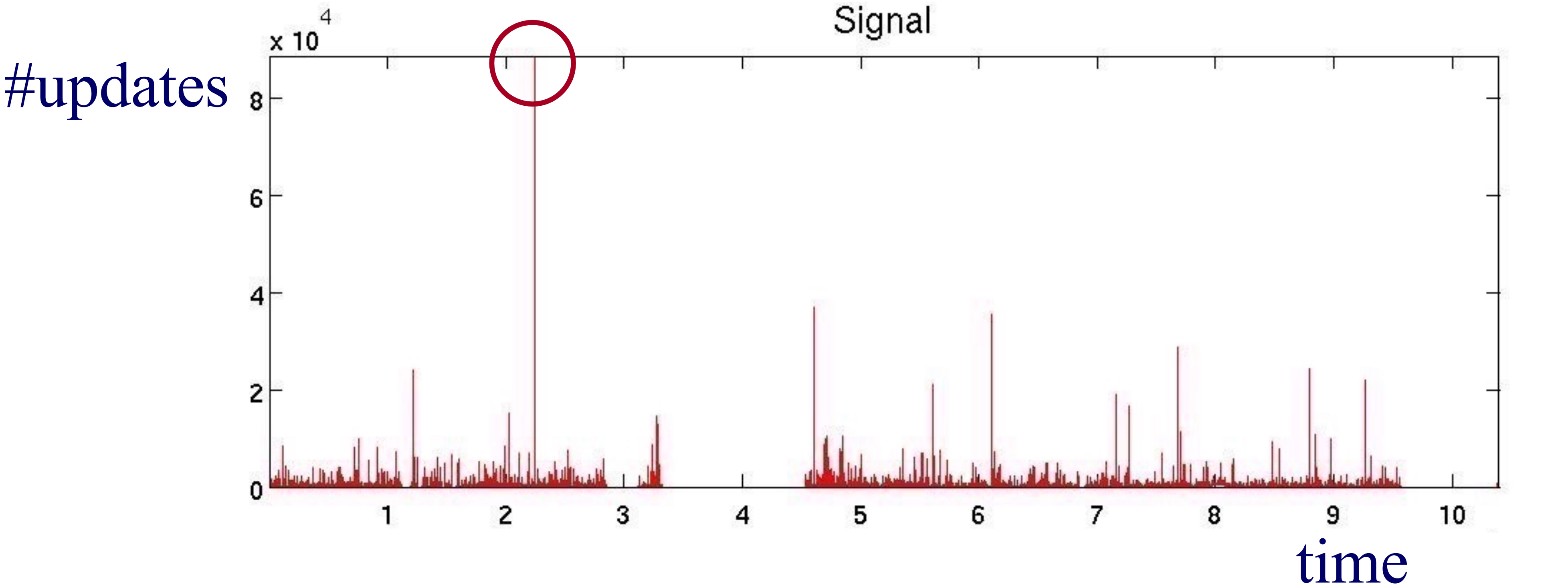


Case study: BGP updates



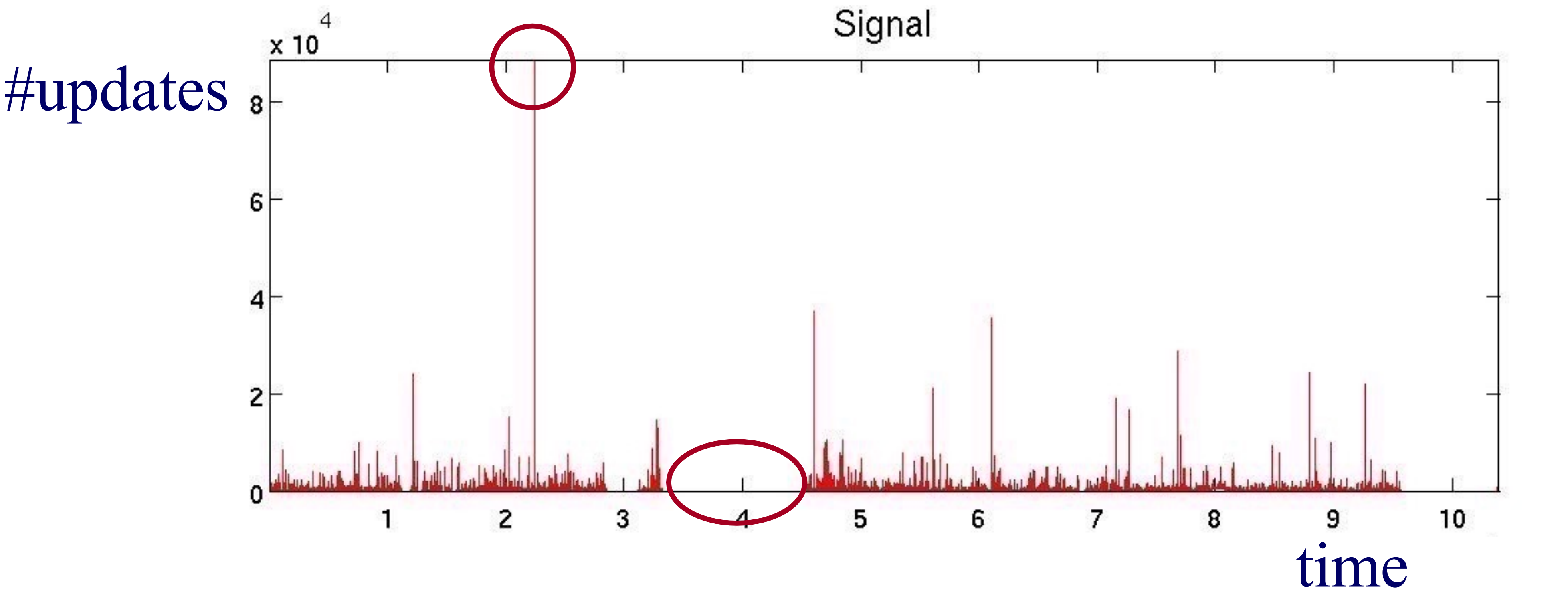
BGP-lens: Patterns and Anomalies in Internet Routing Updates B. Aditya Prakash et al, SIGKDD 2009

Case study: BGP updates



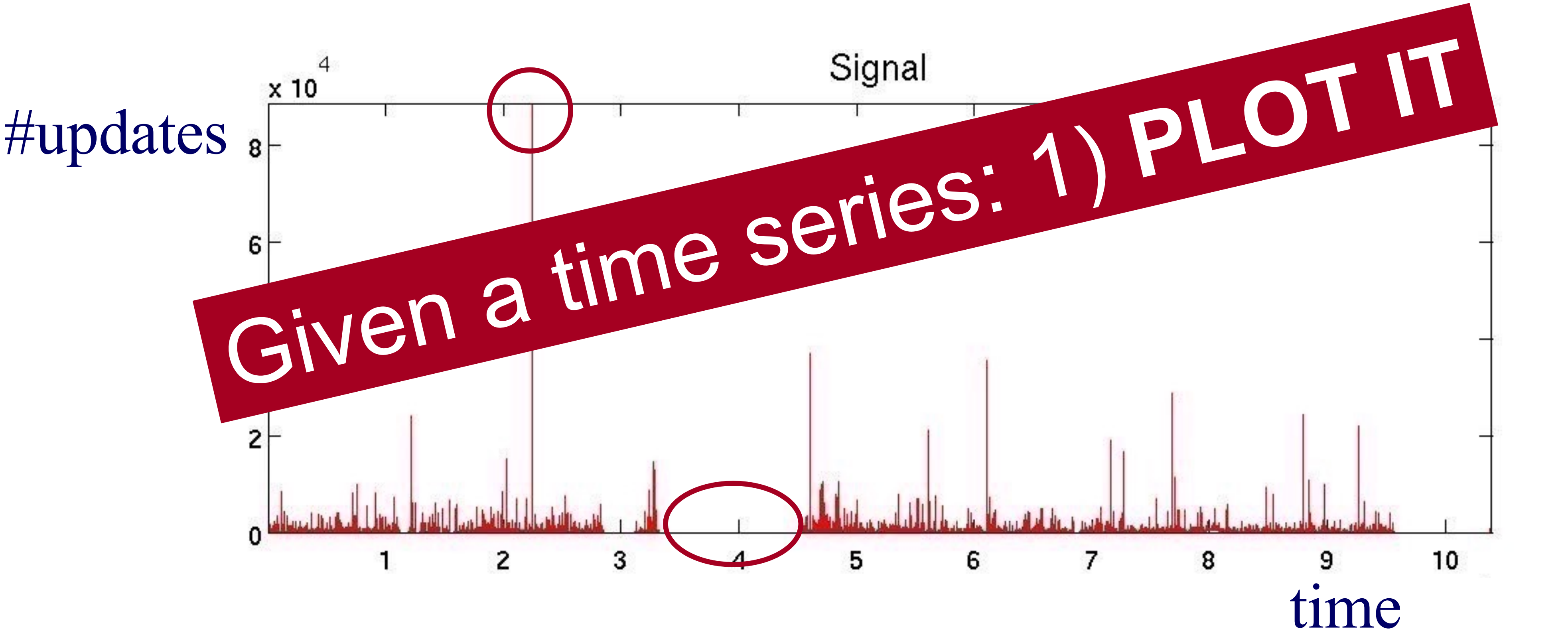
BGP-lens: Patterns and Anomalies in Internet Routing Updates B. Aditya Prakash et al, SIGKDD 2009

Case study: BGP updates



BGP-lens: Patterns and Anomalies in Internet Routing Updates B. Aditya Prakash et al, SIGKDD 2009

Case study: BGP updates



BGP-lens: Patterns and Anomalies in Internet Routing Updates B. Aditya Prakash et al, SIGKDD 2009

Carnegie Mellon

Case study: BGP updates

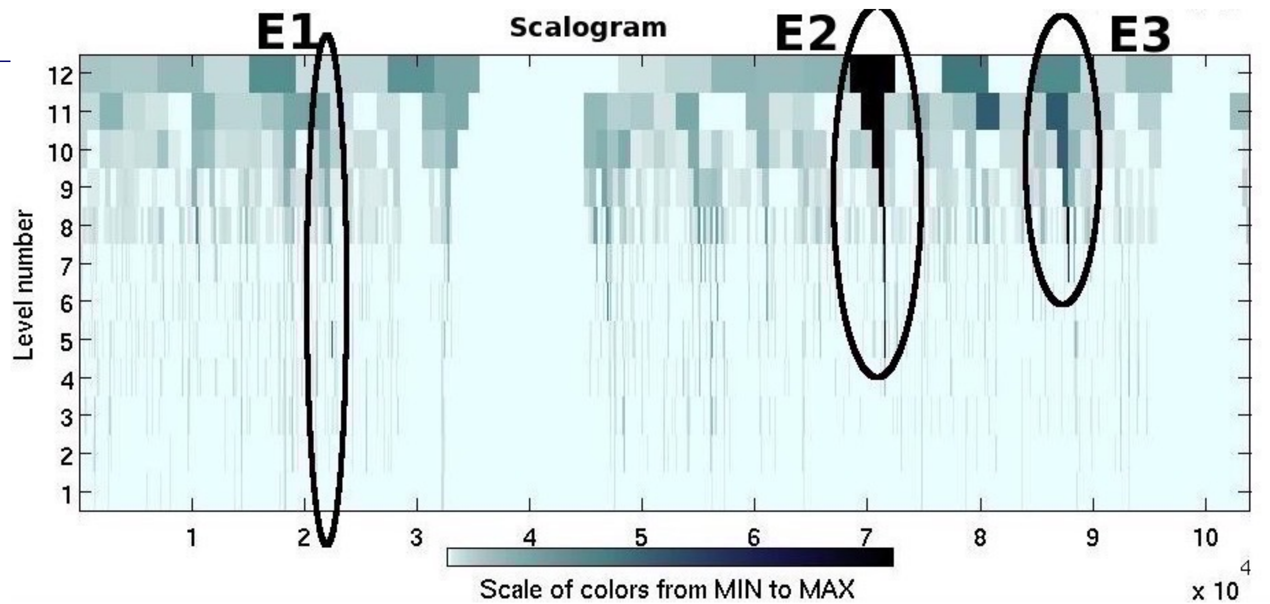
Given a time series: 1) PLOT IT

2) DFT / DWT

Case study: BGP updates

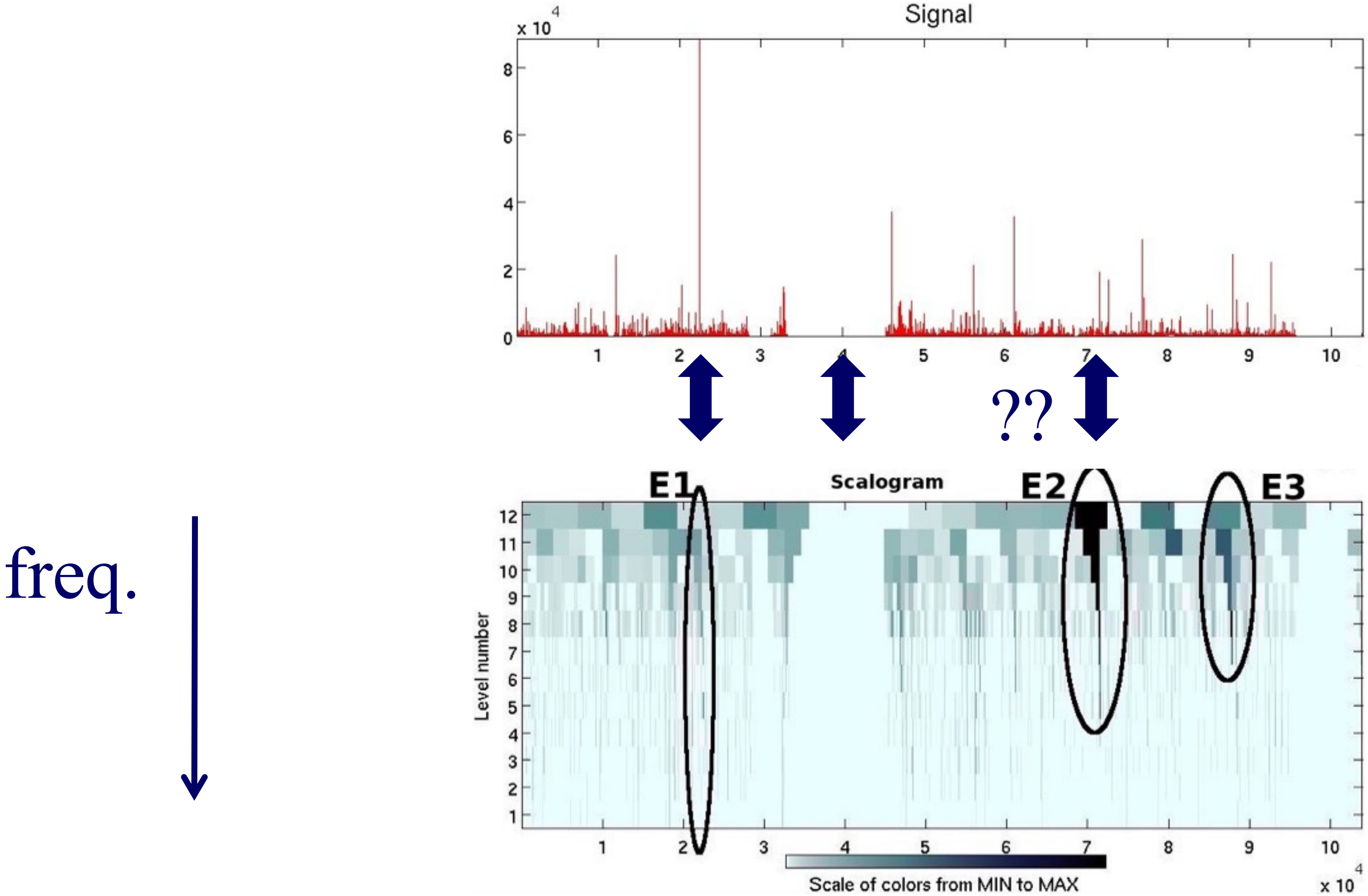
Low freq.: omitted

freq.



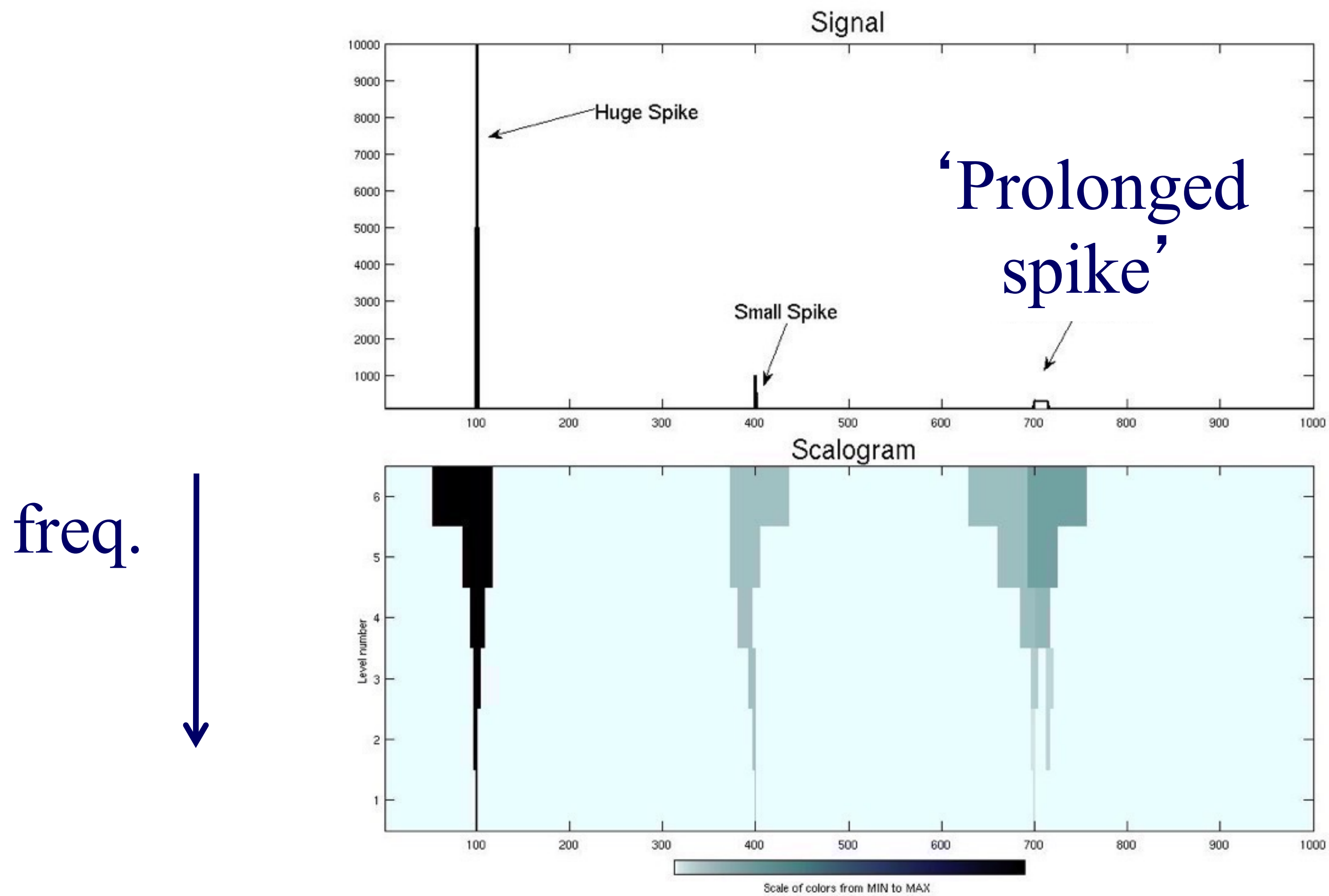
Carnegie Mellon

Case study: BGP updates



CarnegieMellon

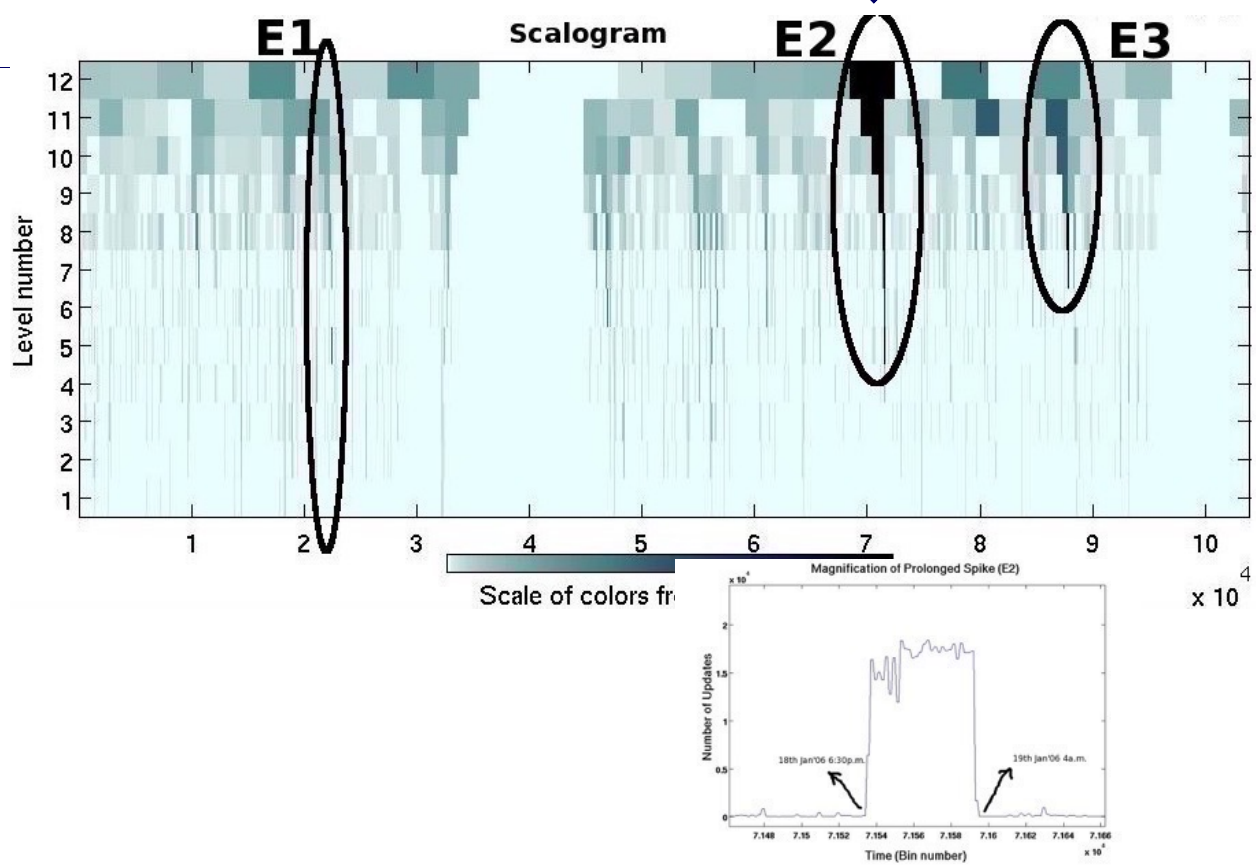
Case study: BGP updates



Case study: BGP updates

15K msgs, for several hours: 6pm-4am

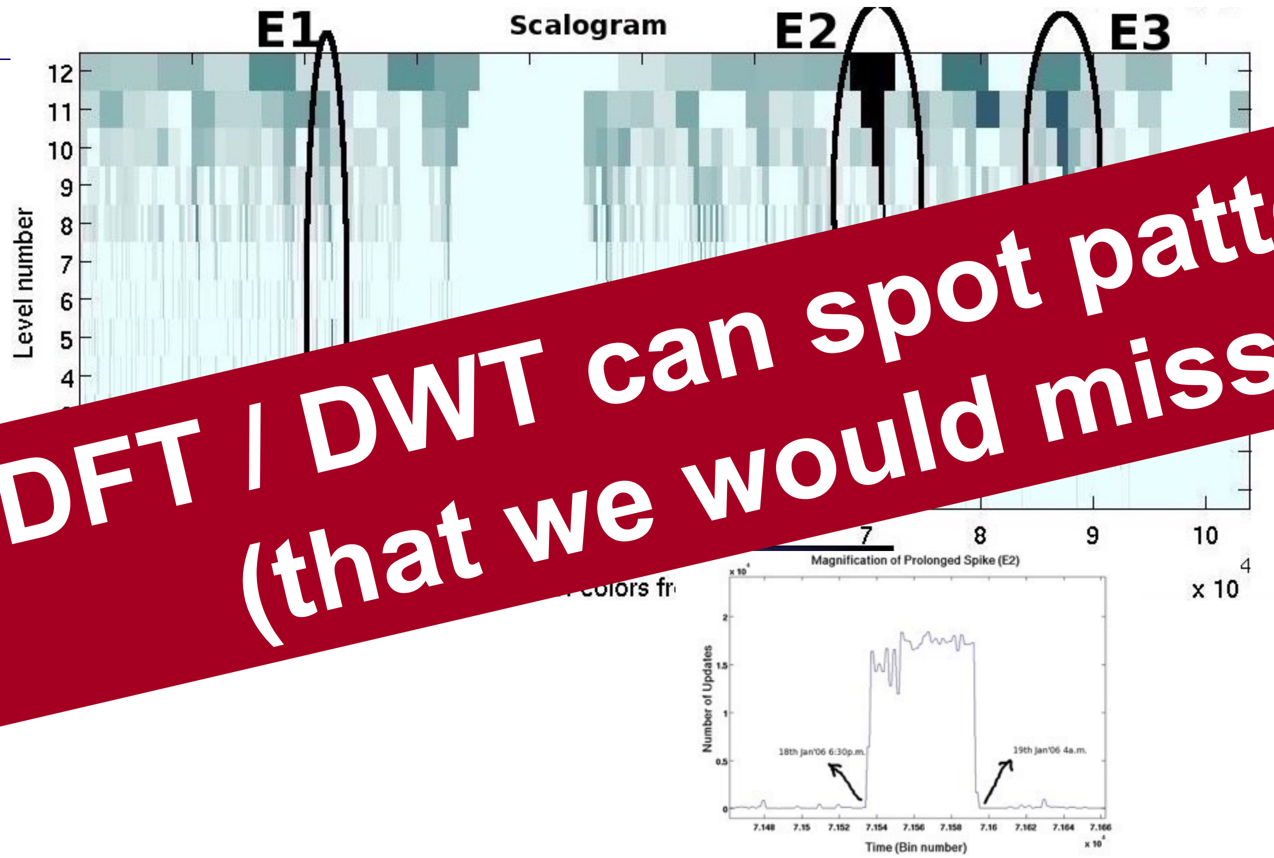
freq.



Case study: BGP updates

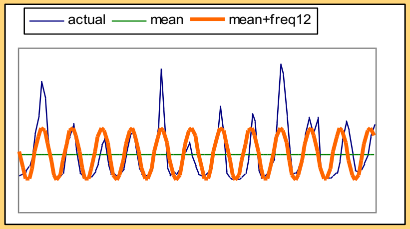
15K msgs, for several hours: 6pm-4am

freq.

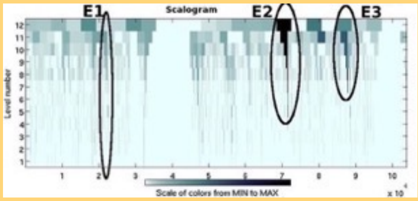


DFT / DWT can spot patterns (that we would miss)

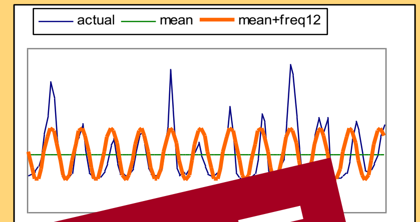
Conclusions for DSP



- DFT spots periodicities
- **DWT** : multi-resolution - matches processing of mammalian ear/eye better
- Both: powerful tools for **compression**, **pattern detection** in real signals

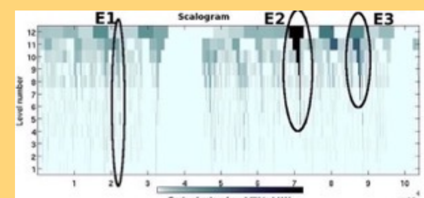


Conclusions for DSP



- DFT spots periodicities
- **DWT** : multi-resolution - mat[illegible]
  processing of m[illegible] [illegible]e bette[illegible]
- B[illegible] [illegible]ools for [illegible]
  pattern det[illegible] [illegible] signals

Given a time series: 1) PLOT IT
2) DFT / DWT



Resources: software and urls

- *xwpl:* open source wavelet package from Yale, with excellent GUI
- *pywavelets* (python library)
- Included in matlab™

Books

- William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery: *Numerical Recipes in C*, Cambridge University Press, 1992, 2nd Edition. (Great description, intuition and code for DFT, DWT)
- C. Faloutsos: *Searching Multimedia Databases by Content*, Kluwer Academic Press, 1996 (introduction to DFT, DWT)

Carnegie Mellon

Part 3: Linear Forecasting

Outline

- Motivation
- P1. Similarity Search and Indexing
- P2. DSP (Digital Signal Processing)
- ➡ P3. Linear Forecasting
- P4. Non-linear forecasting
- Conclusions

Carnegie Mellon

Forecasting

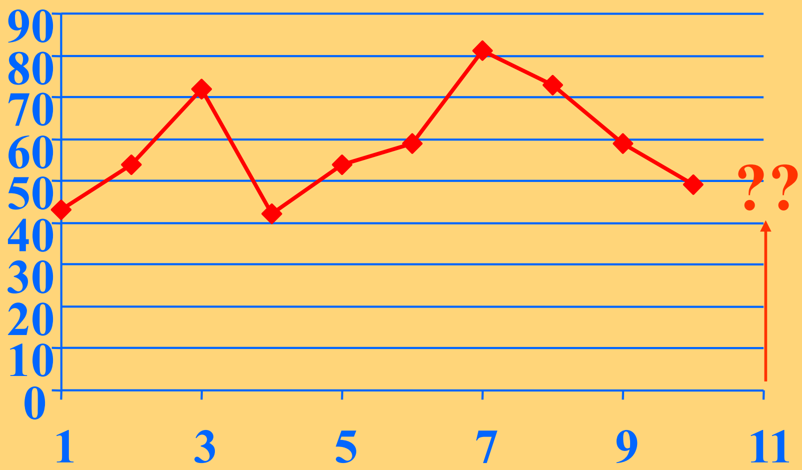
"Prediction is very difficult, especially about the future." - Nils Bohr

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

Carnegie Mellon

Problem#3: Forecast

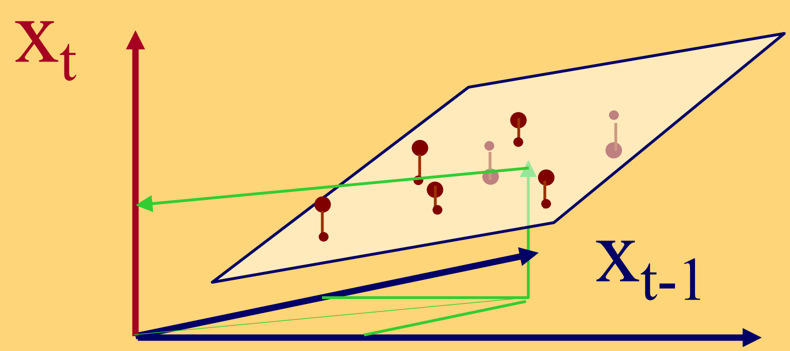
- given x_{t-1}, x_{t-2}, …,
- Q: forecast x_t



Solution: AR(IMA)

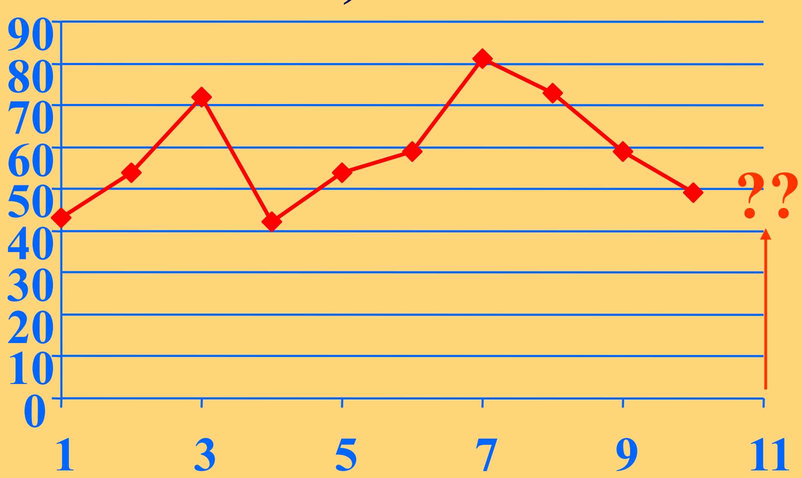
- given x_{t-1}, x_{t-2}, …,
- Q: forecast x_t
- A: AR(IMA) = Box-Je rs, Kalman)

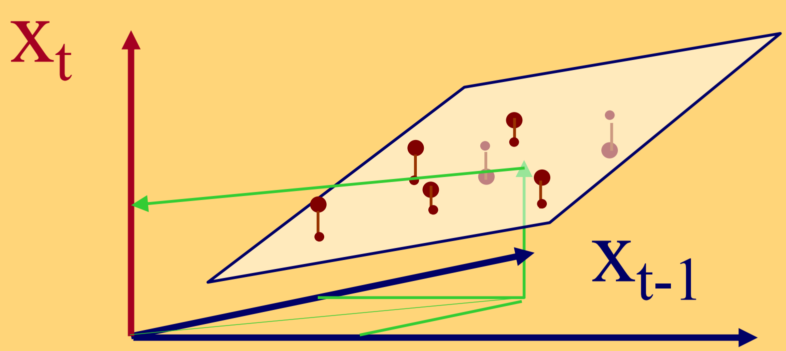
Auto **R**egressive
Integrated
Moving **A**verage



Solution: AR(IMA)

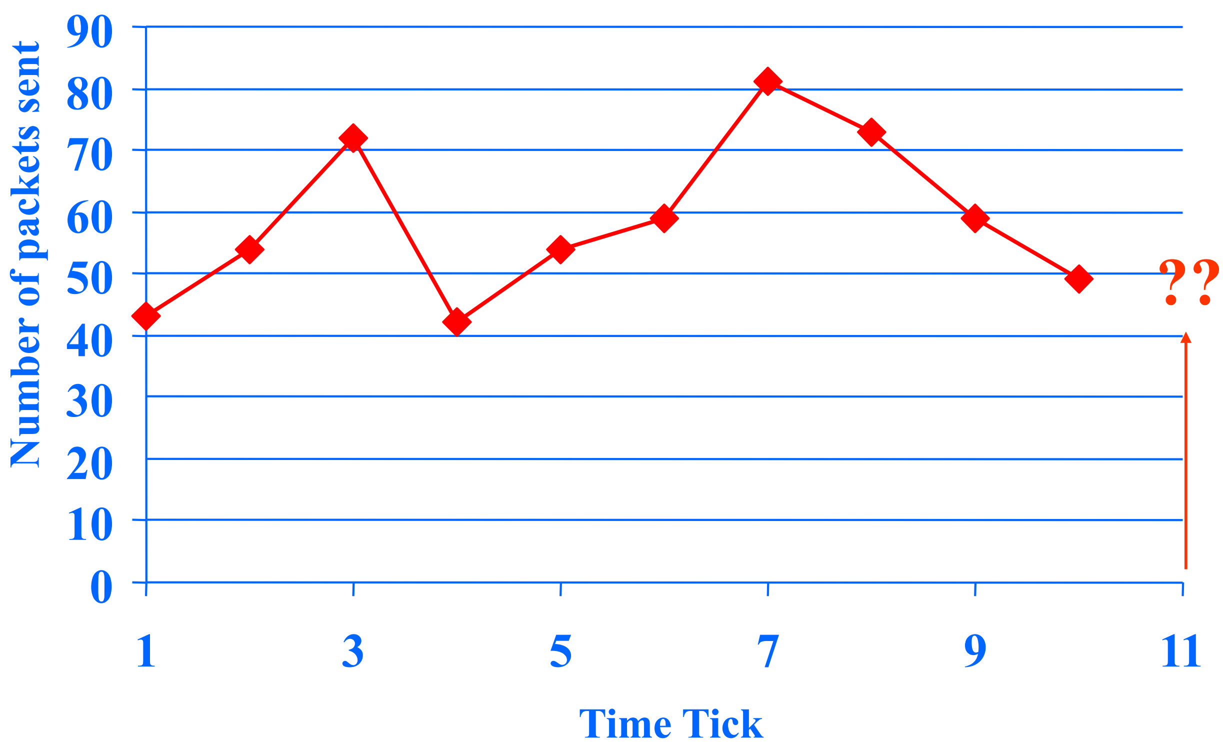
- given x_{t-1}, x_{t-2}, …,
- Q: forecast x_t
- A: AR(IMA) = Box-Jenkins (< Holt-Winters, Kalman)





Problem#3: Forecast

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



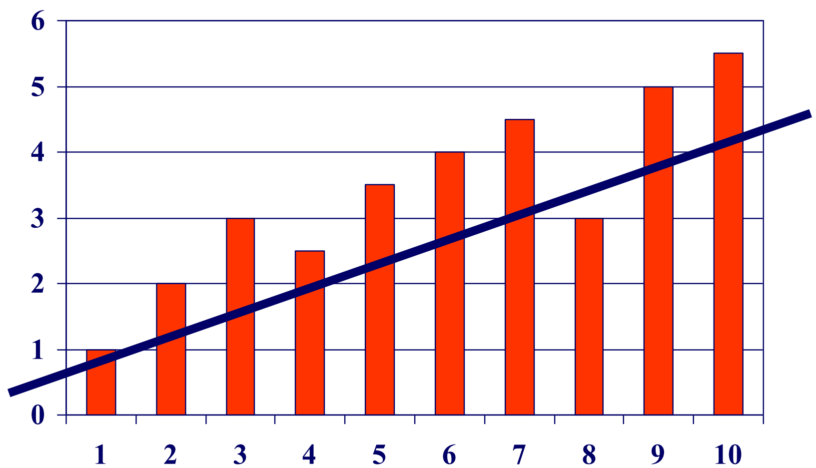
Forecasting: Preprocessing

MANUALLY:

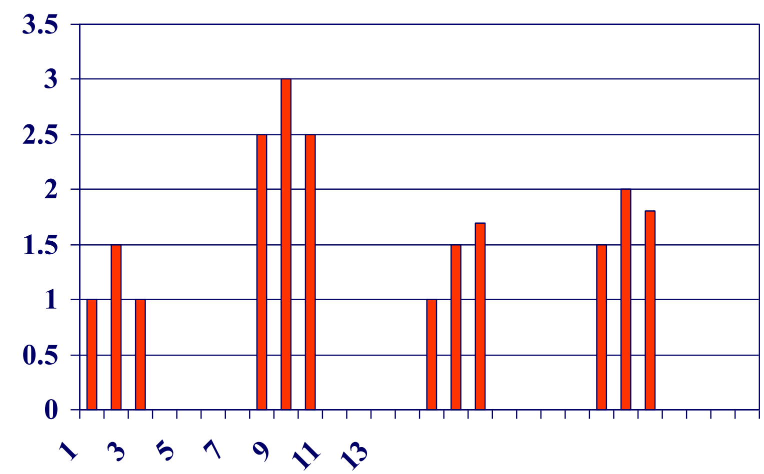
remove trends

spot periodicities

7 days



time



time

Problem#3: Forecast

- Solution: try to express

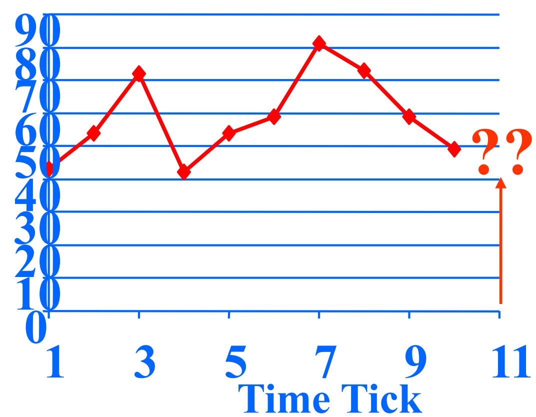
  x_t

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

  (up to a window of w)

Formally:

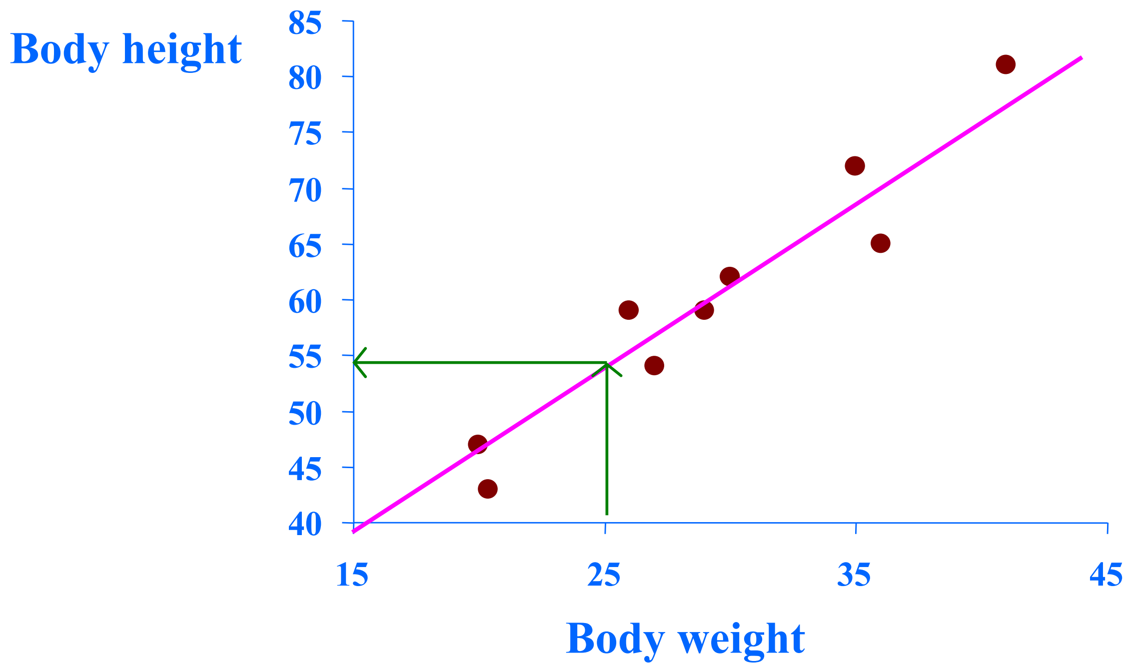
$$x_t \approx a_1 x_{t-1} + \ldots + a_w x_{t-w} + noise$$



Carnegie Mellon

Linear Regression: idea

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



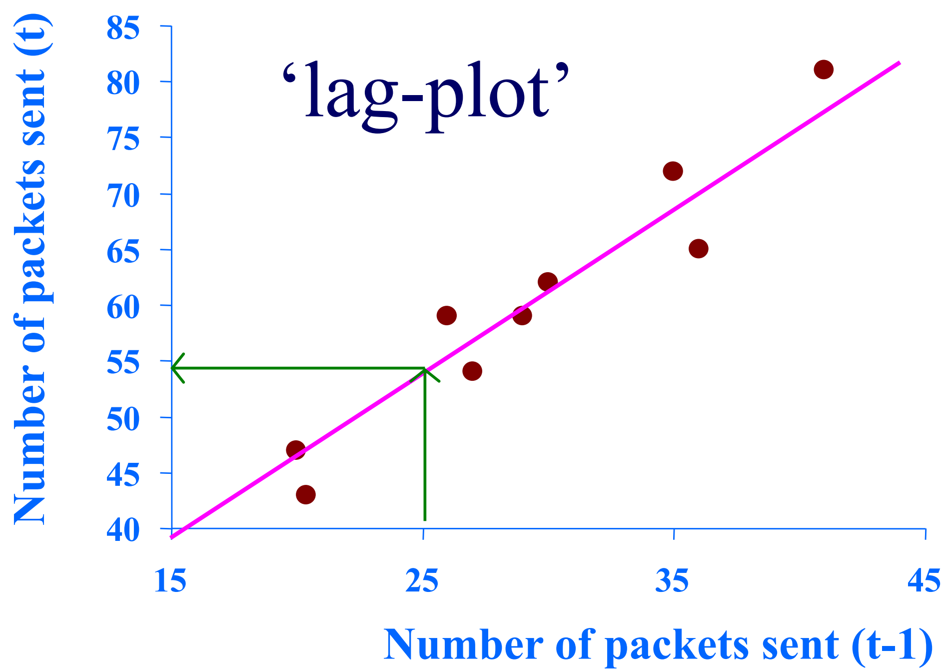
- express what we don't know (= 'dependent variable')
- as a linear function of what we know (= 'indep. variable(s)')

Linear Auto Regression:

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

Linear Auto Regression:

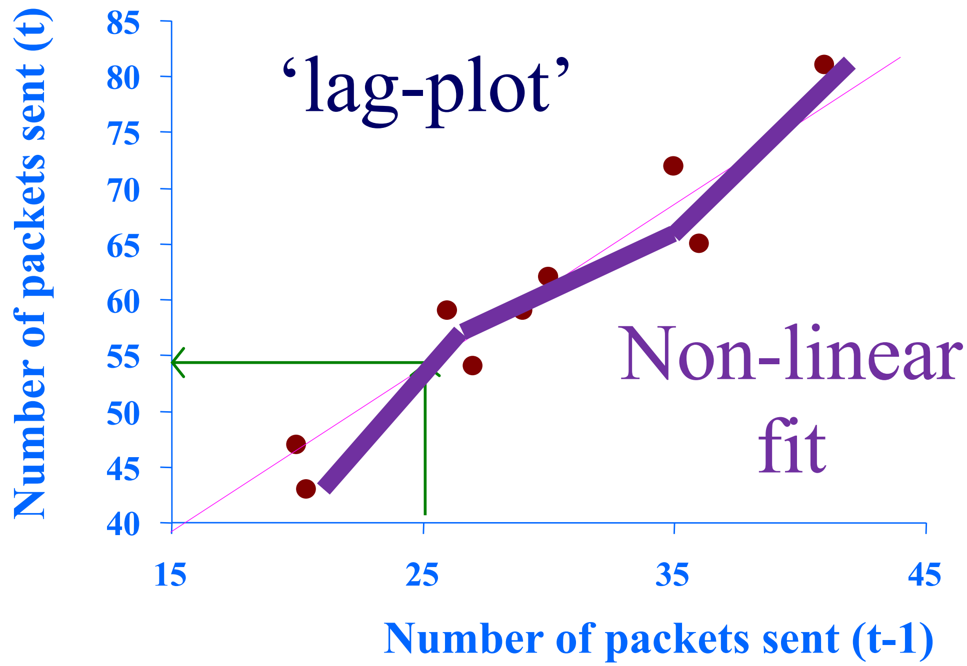
Time	*Packets Sent (t-1)*	*Packets Sent(t)*
1	-	43
2	43	54
3	54	72
	…	
…		…
N	25	??



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

Deep Learning:

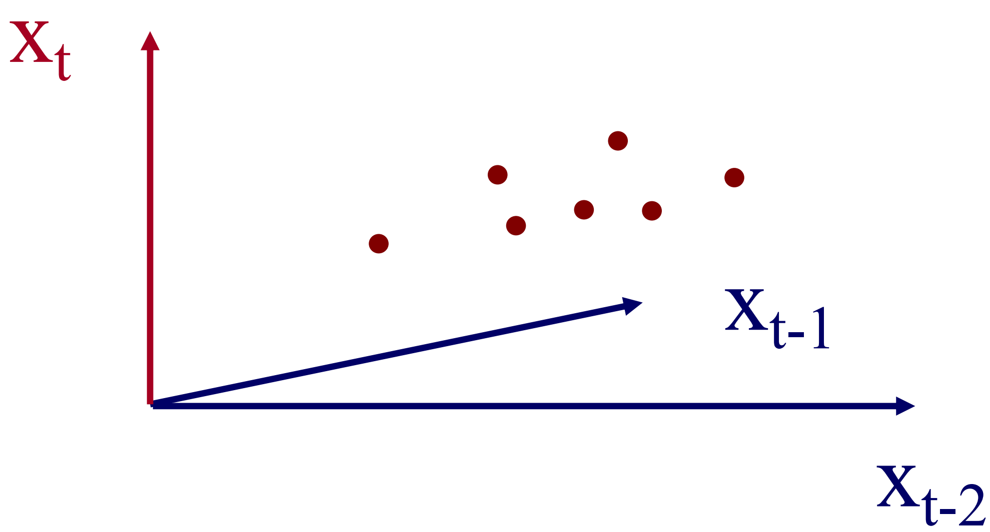
Time	*Packets Sent (t-1)*	*Packets Sent(t)*
1	-	43
2	43	54
3	54	72
	…	
…		…
N	25	??



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

More details:

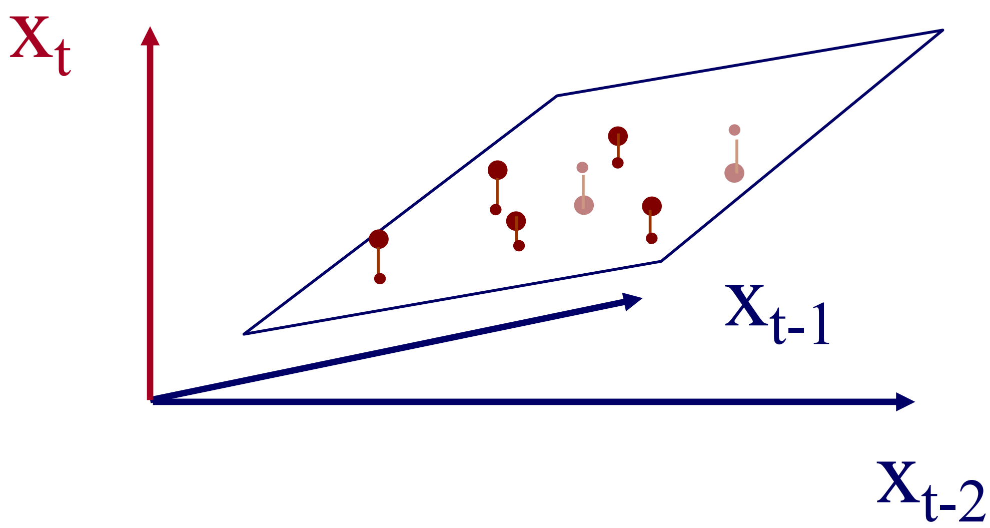
- Q1: Can it work with window w>1?
- A1: YES!



Carnegie Mellon

More details:

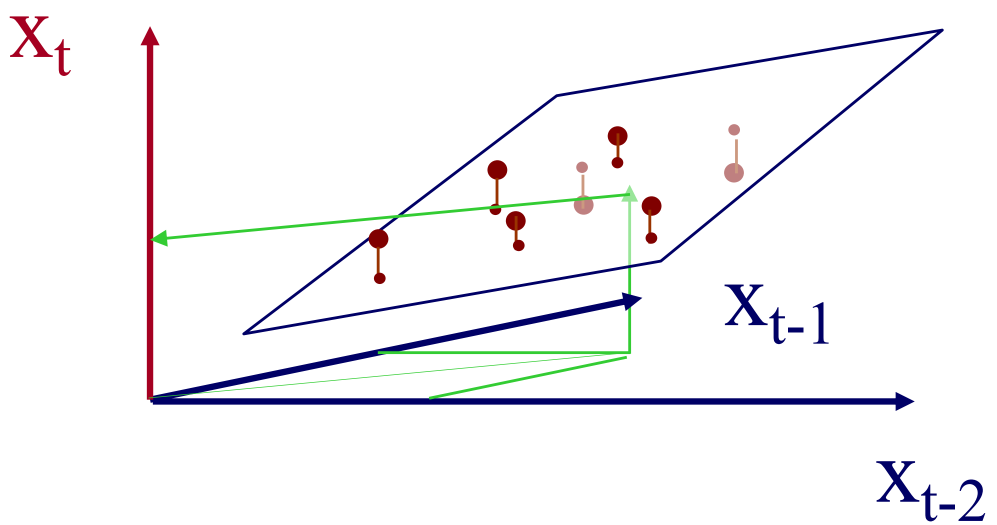
- Q1: Can it work with window w>1?
- A1: YES! (we'll fit a hyper-plane, then!)



Carnegie Mellon

More details:

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



DETAILS

More details:

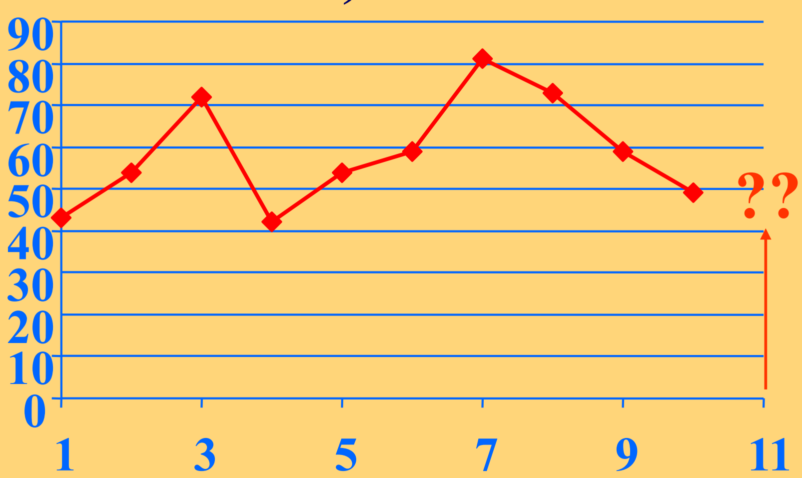
- Q1: Can it work with window *w*>1?
- A1: YES! The problem becomes:

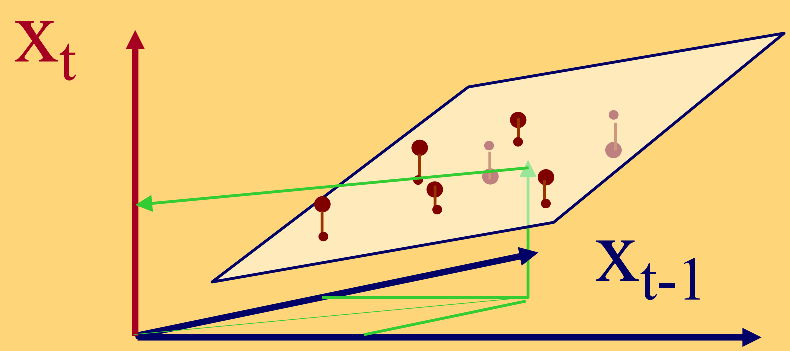
$$\mathbf{X}_{[N \times w]} \times \mathbf{a}_{[w \times 1]} = \mathbf{y}_{[N \times 1]}$$

- 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

Solution: AR(IMA)

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





Resources: software and urls

- http://cran.r-project.org/ ('R' system)
- https://www.statsmodels.org/ (python)
- …<many more>

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.

Additional Reading

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

Part 4: chaos and non-linear forecasting

Outline

- Motivation
- P1. Similarity Search and Indexing
- P2. DSP (Digital Signal Processing)
- P3. Linear Forecasting
- ➡ P4. Non-linear forecasting
- Conclusions

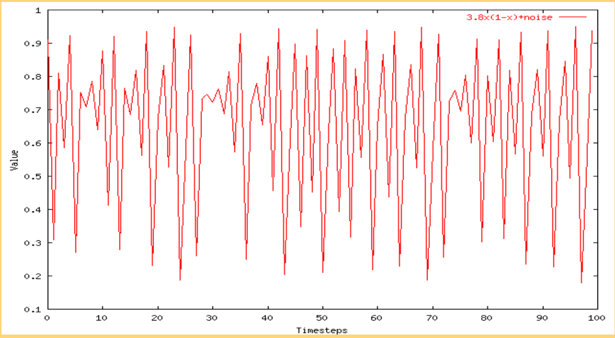
Carnegie Mellon

Detailed Outline

- …
- P3. Linear forecasting
- P4. Non-linear forecasting
  - Problem
  - Idea
  - How-to
  - Experiments

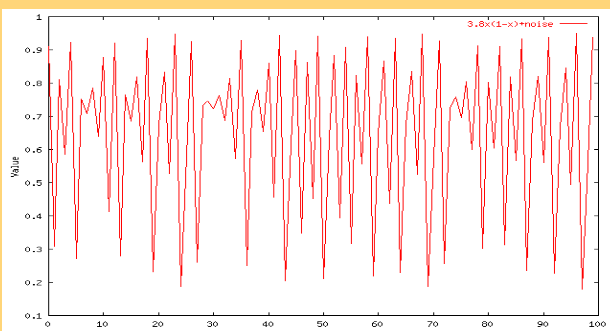
Problem: Forecast

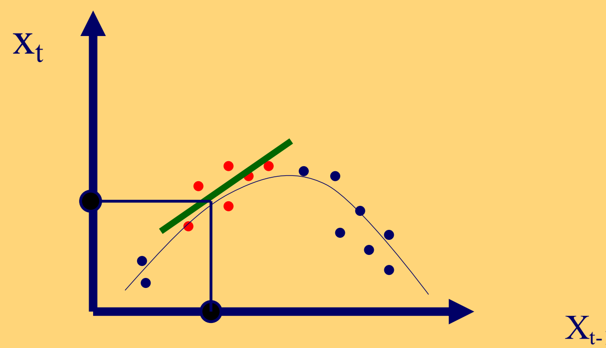
- given x_{t-1}, x_{t-2}, …,   (‘chaotic’/non-linear)
- Q: forecast x_t



Solution

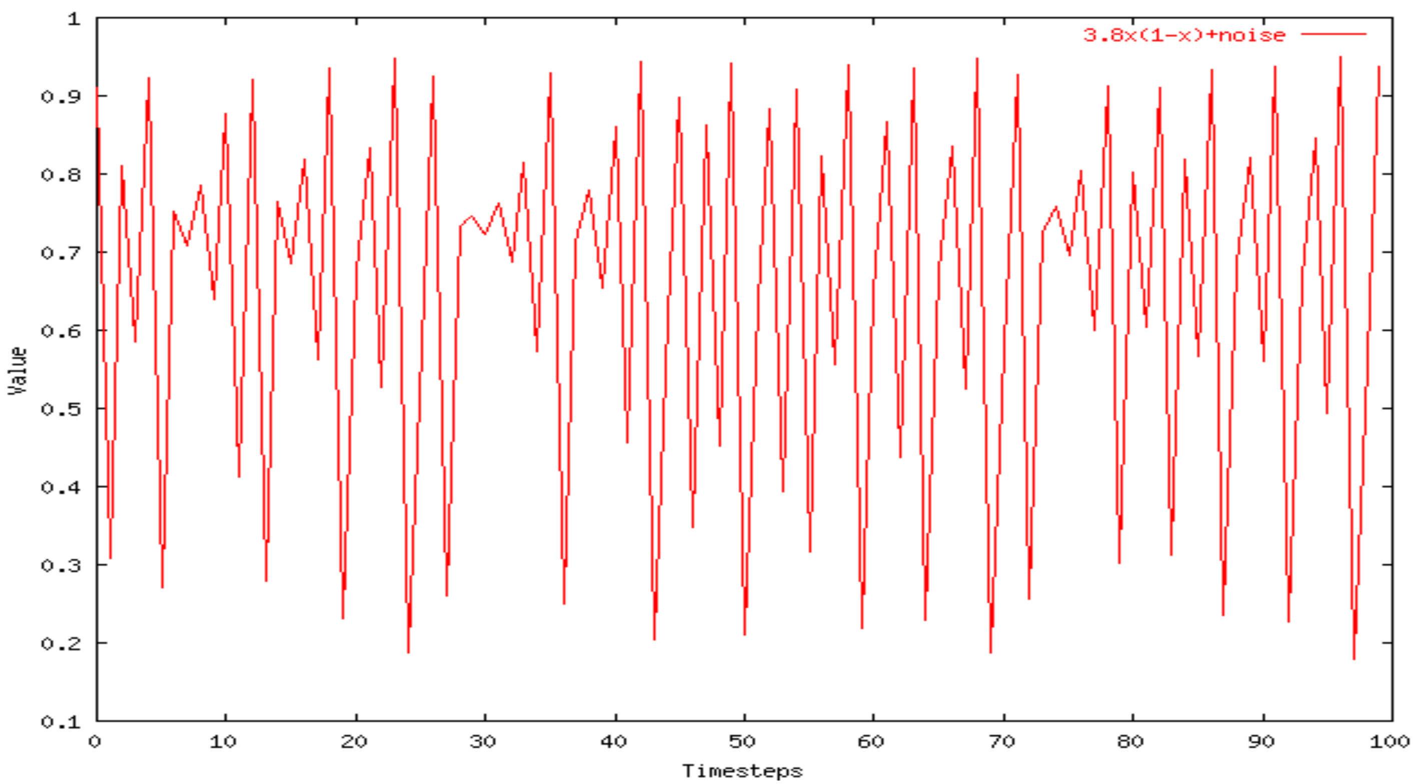
- given x_{t-1}, x_{t-2}, …,   (‘chaotic’/non-linear)
- Q: forecast x_t
- A: lag-plots + sim. search (= ‘Delayed Coordinate Embedding’)





Recall: Problem #3

Value

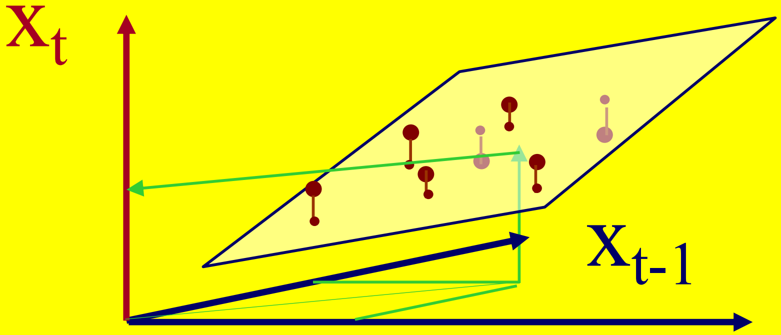


Time

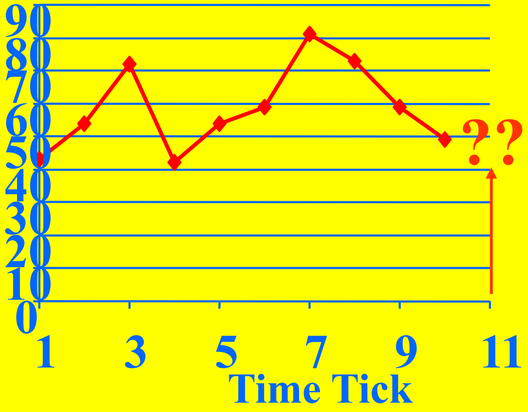
Given a time series $\{x_t\}$, predict its future course, that is, x_{t+1}, x_{t+2}, ...

How to forecast?

- what could go wrong with ARIMA?

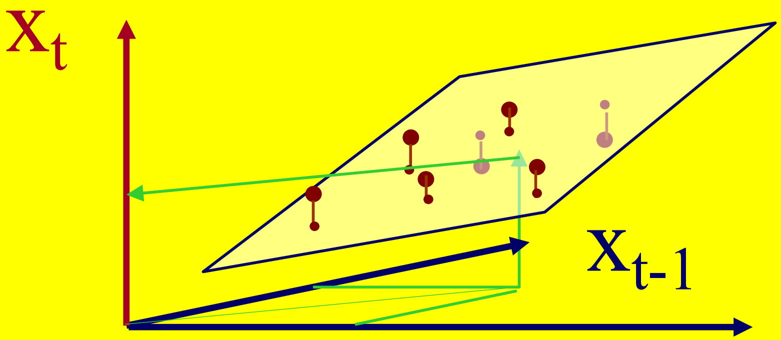


$$x_t \approx a_1 x_{t-1} + \ldots + a_w x_{t-w} + noise$$

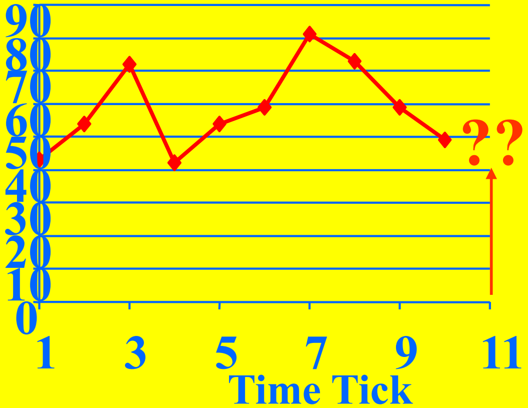


How to forecast?

- what could go wrong with ARIMA?
- A: **linearity** assumption might not hold!



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



CarnegieMellon

ARIMA pitfall

Example: logistic parabola

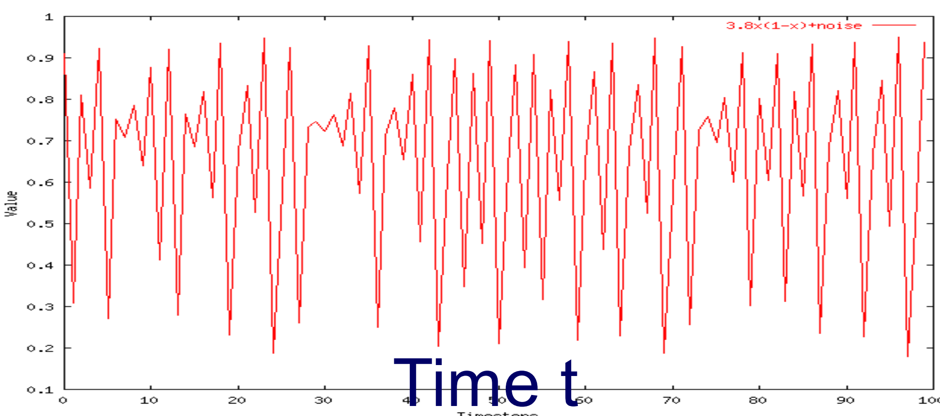
Models population of flies [R. May/1976]

#flies(t+1) | Reproductive rate | Max# flies

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

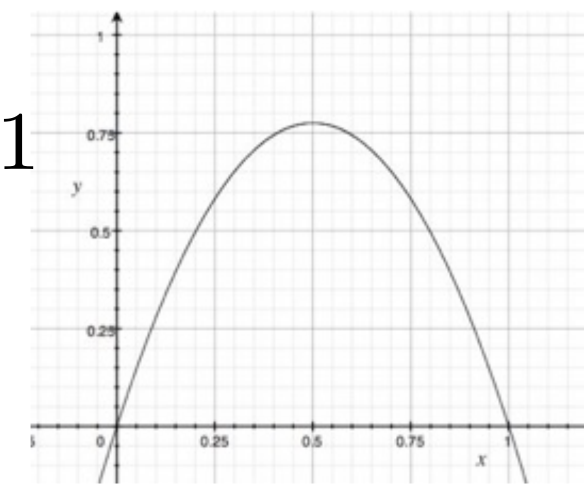
Time-series plot

x_t



Time t

x_{t+1}



x_t

ARIMA pitfall

Example: logistic parabola

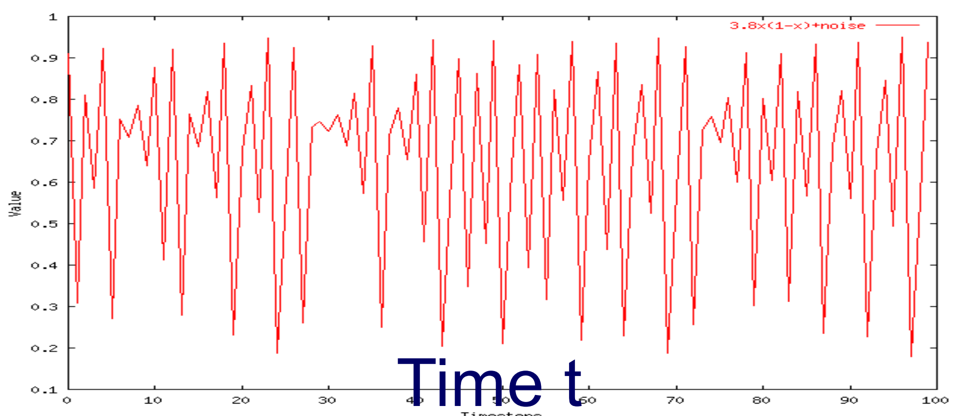
Models population of flies [R. May/1976]

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

Logistic map

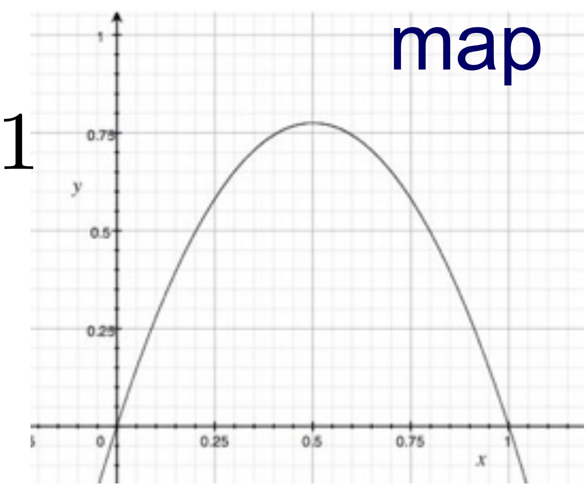
Time-series plot

x_t



Time t

x_{t+1}



x_t

ARIMA pitfall

Example: logistic parabola

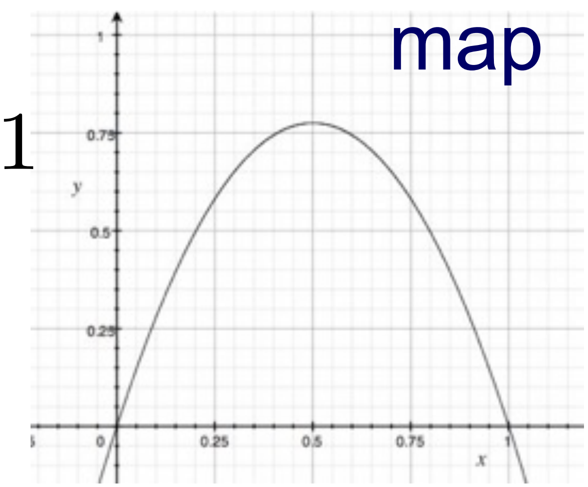
Models population of flies [R. May/1976]

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

Logistic map

- = SI virus prop. model (covid: S.I.R.)
- ~ Bass equation (market penetration)
- Special case of Lotka-Volterra
  - Competing species / products

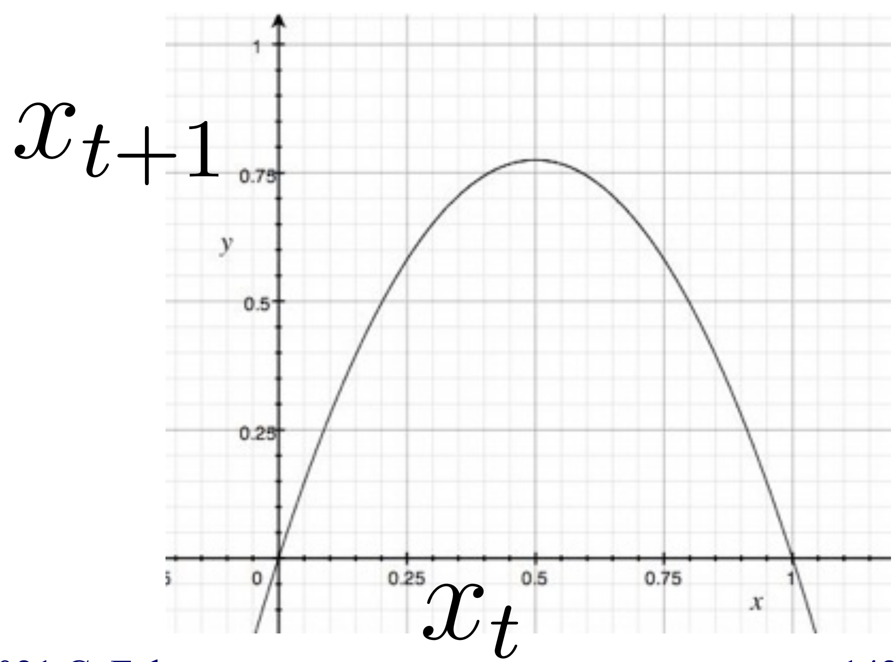
+1



x_t

ARIMA pitfall

Linear equations, e.g., AR, ARIMA, …

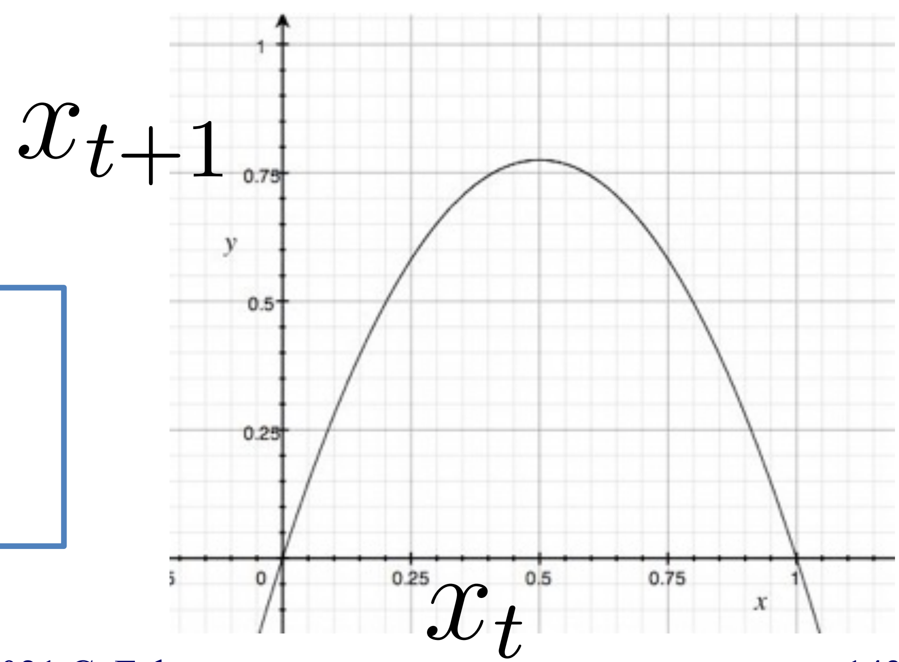


ARIMA pitfall

Linear equations, e.g., AR, ARIMA, …

e.g., AR(1)

$$x_{t+1} = a x_t + \epsilon$$

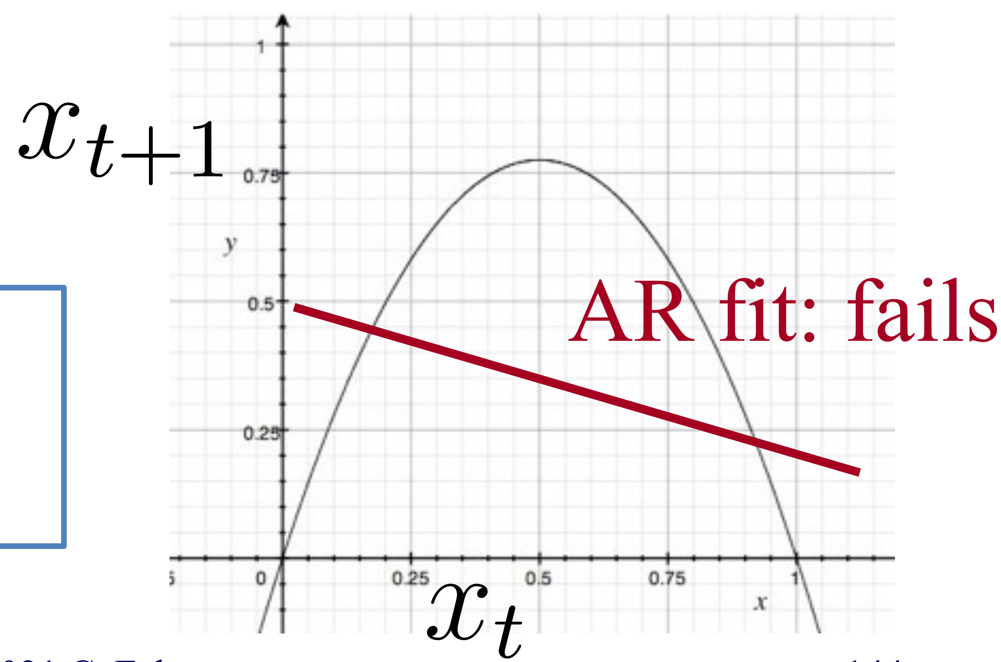


ARIMA pitfall

Linear equations, e.g., AR, ARIMA, …

e.g., AR(1)

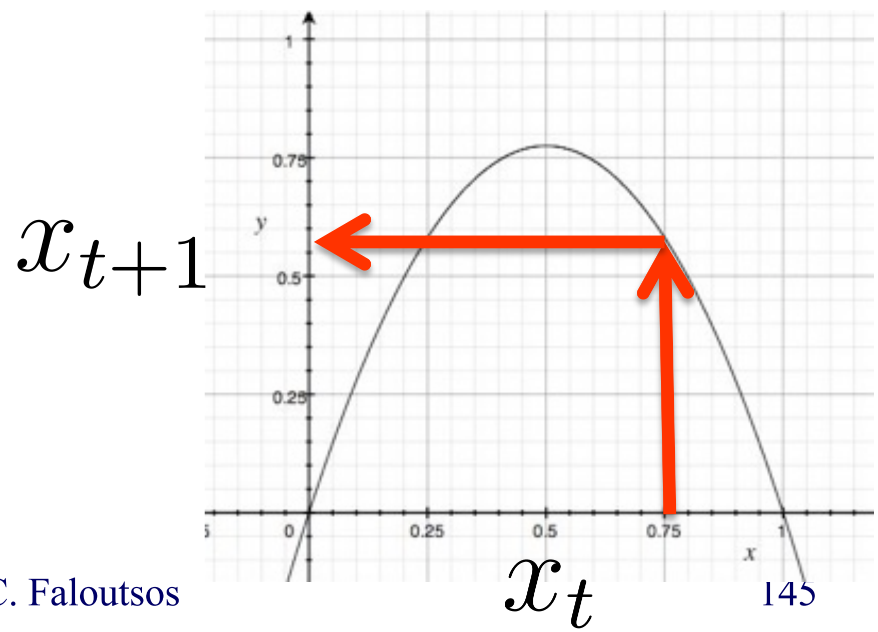
$$x_{t+1} = a x_t + \epsilon$$



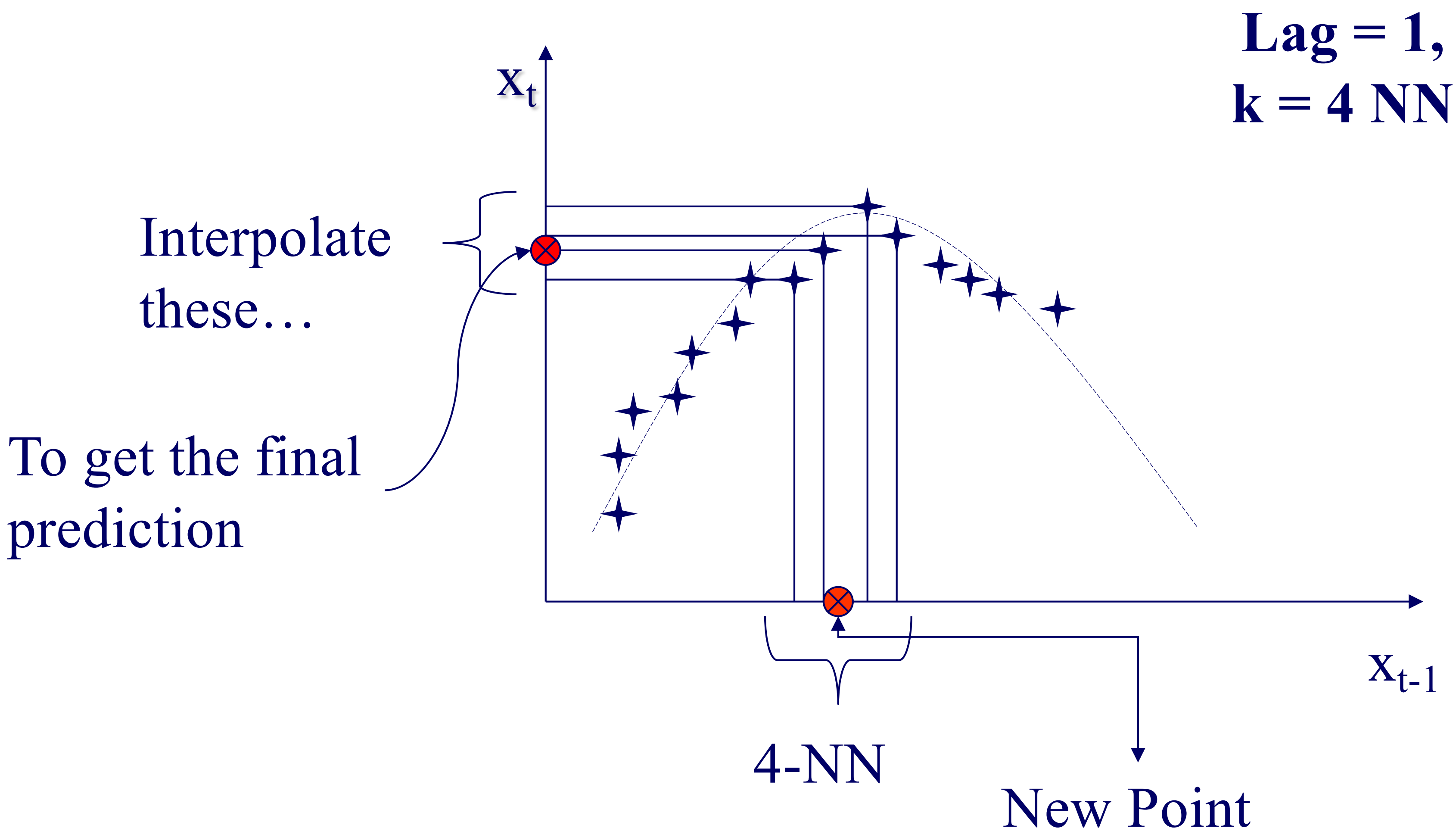
Solution?

- “Delayed Coordinate Embedding” = Lag Plots [Sauer94]

k-nearest neighbor search



General Intuition (Lag Plot)



Details

Q: How to interpolate?

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

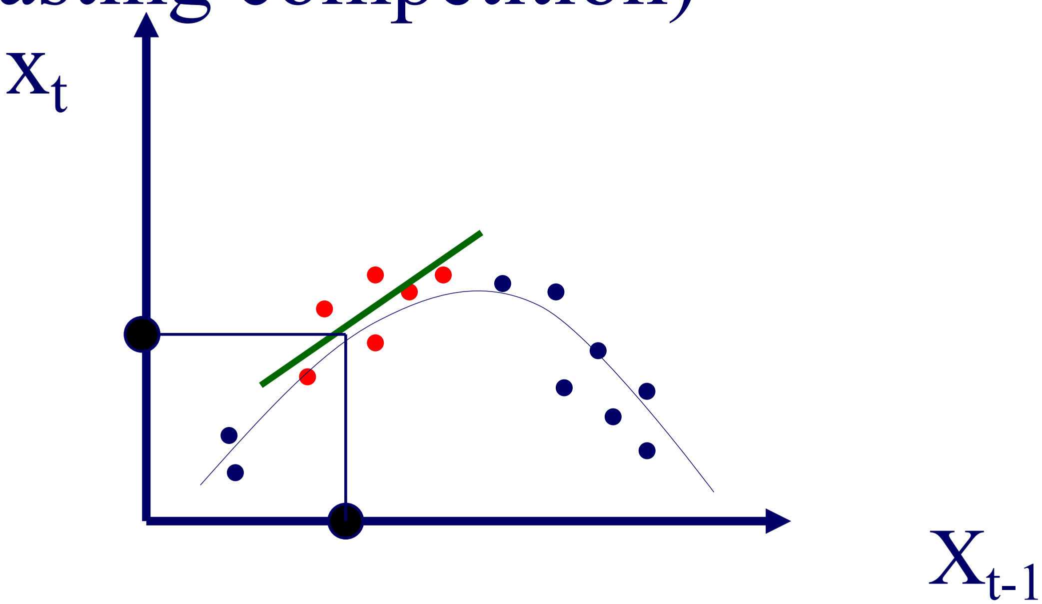
A1: Average

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

Details

Q: How to interpolate?

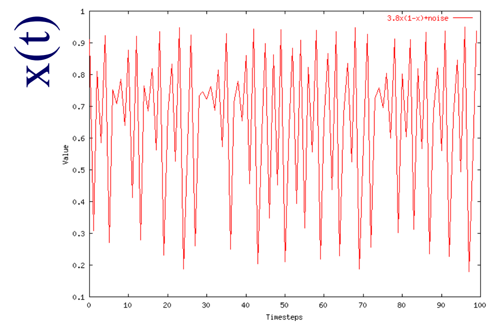
A3: Using SVD - seems to perform best ([Sauer94] - first place in the Santa Fe forecasting competition)



Detailed Outline

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

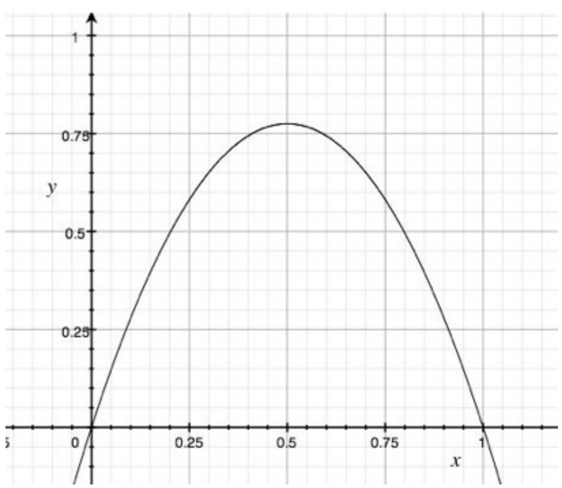
Datasets



Logistic Parabola:

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

Models population of flies [R. May/1976]



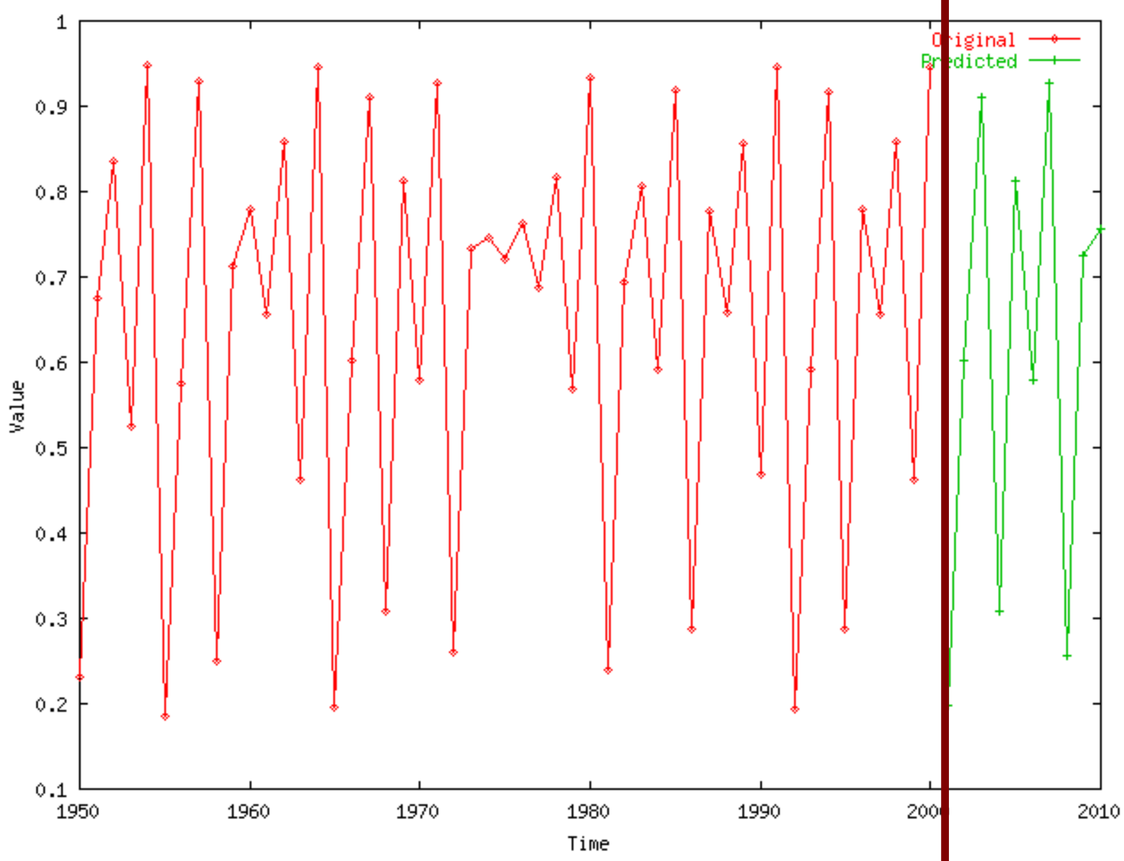
Lag-plot

Carnegie Mellon

Logistic Parabola

Our Prediction from here

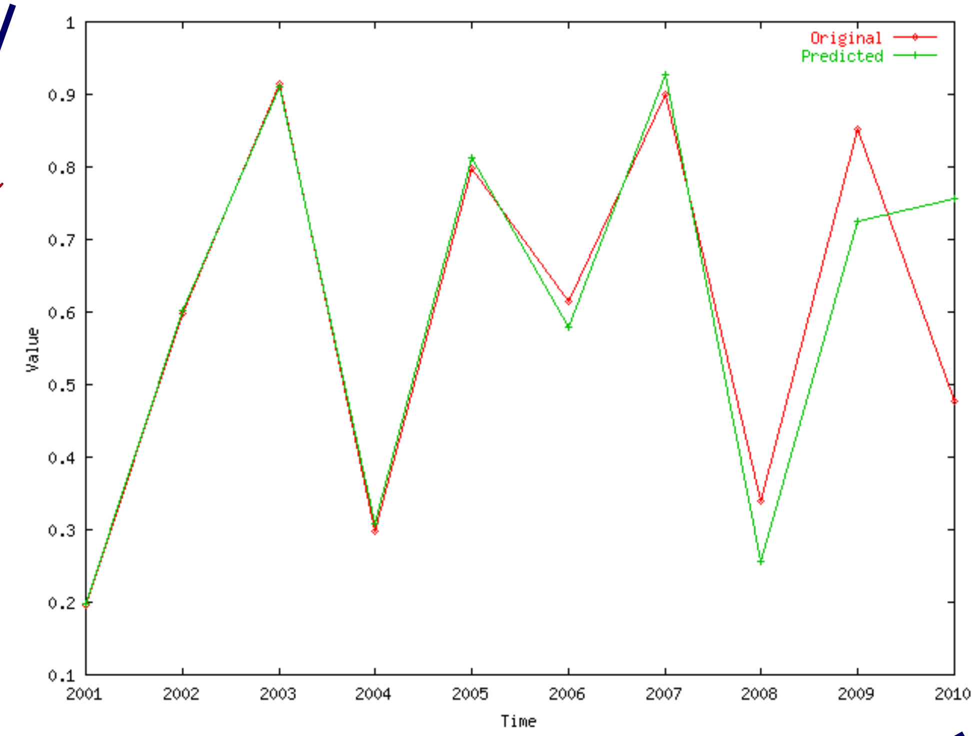
Value

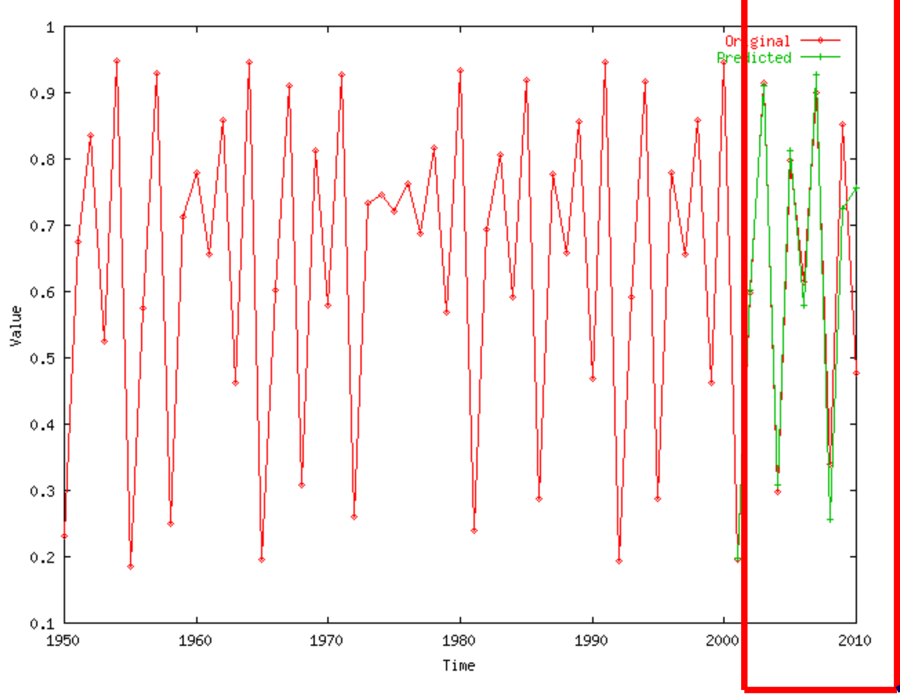


Timesteps

Logistic Parabola

Comparison of prediction
to correct values



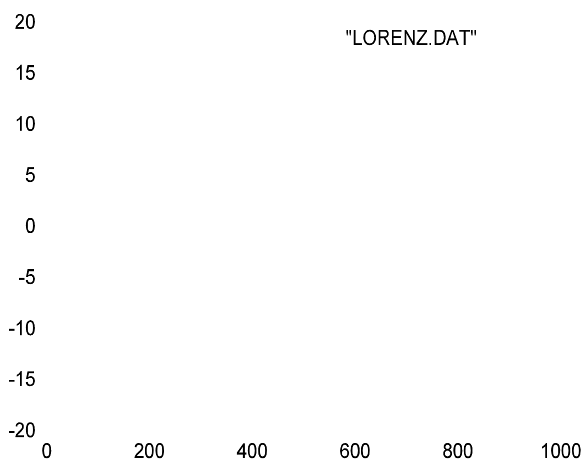


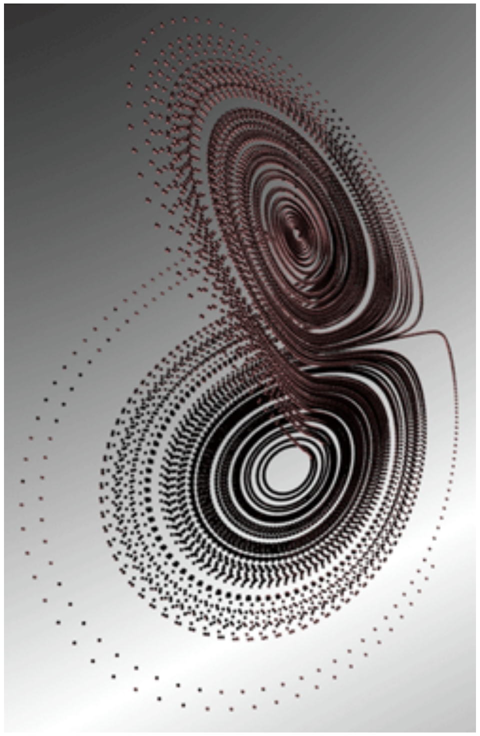
Edward Lorenz

Datasets

LORENZ: Models convection currents in the air

$$dx / dt = a (y - x)$$
$$dy / dt = x (b - z) - y$$
$$dz / dt = xy - c z$$



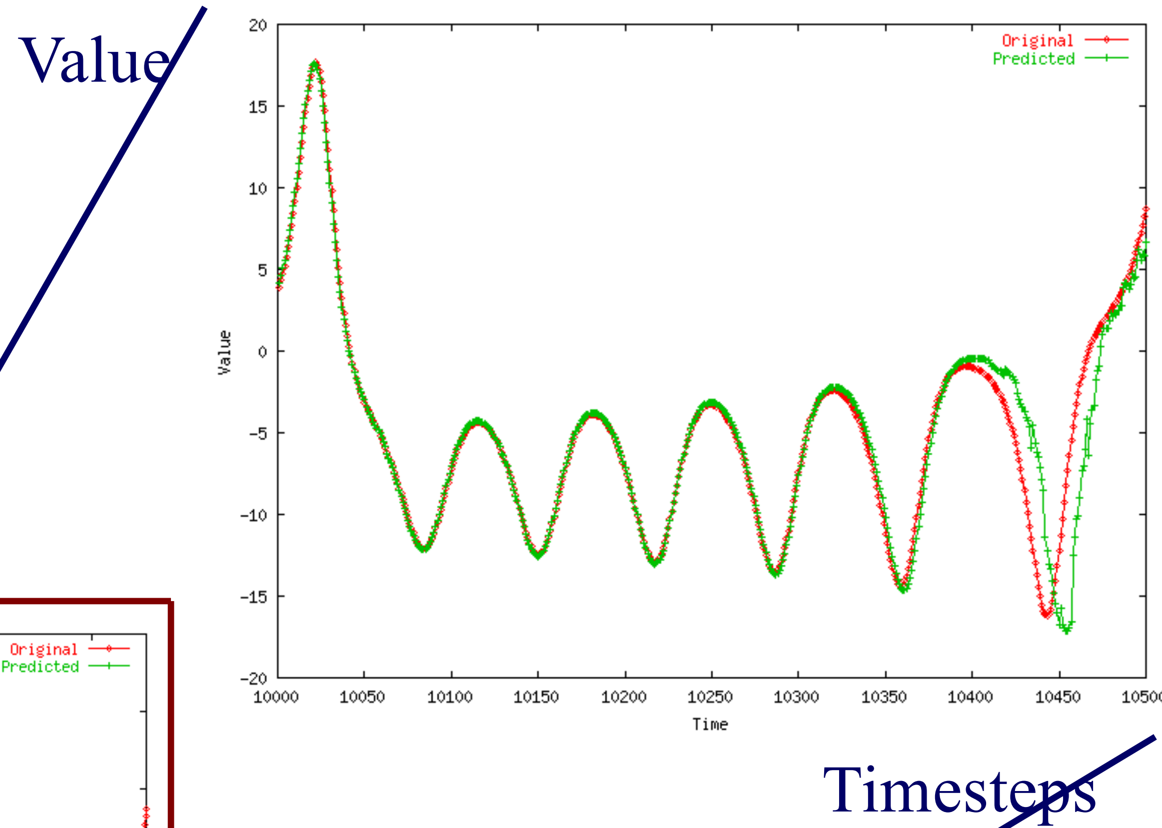


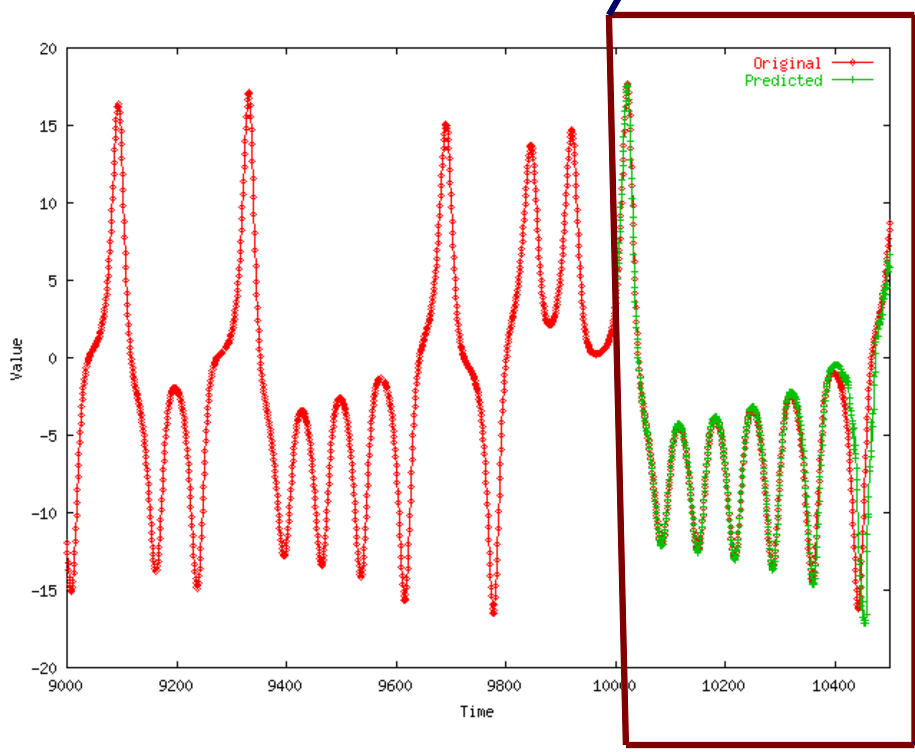
- **Deterministic chaos**
- **'butterfly effect'**

Carnegie Mellon

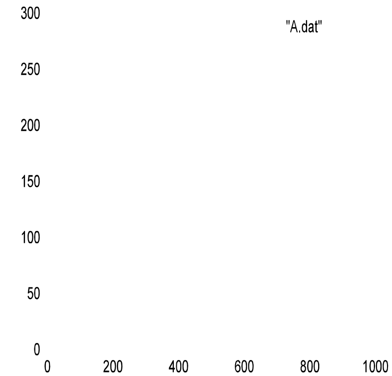
LORENZ

Comparison of prediction to correct values





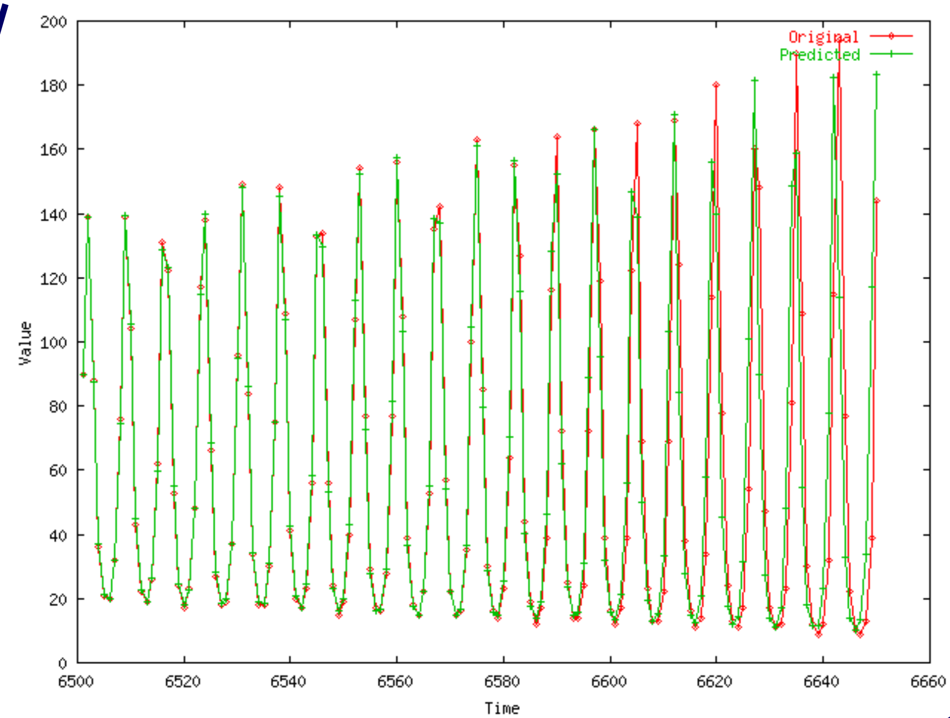
Datasets

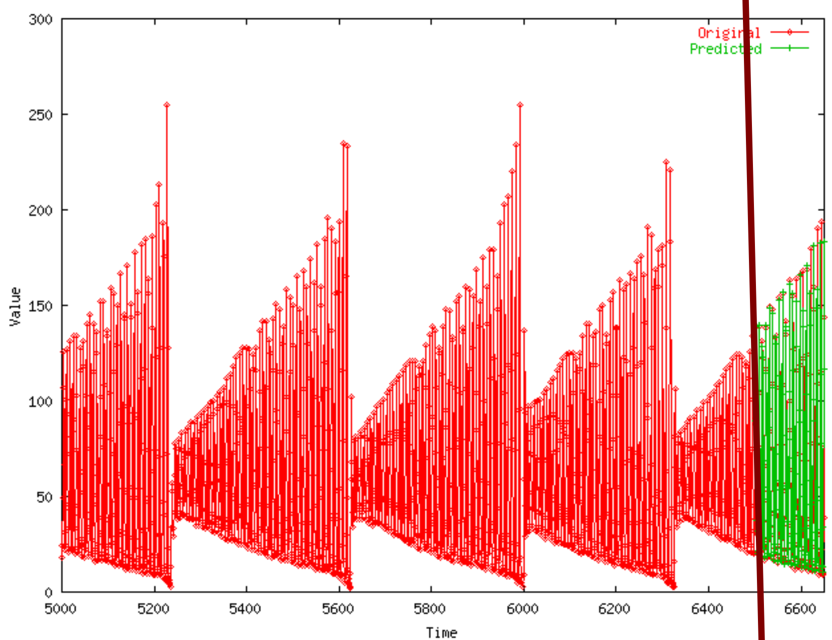


- LASER: fluctuations in a Laser over time (used in Santa Fe competition)

Laser

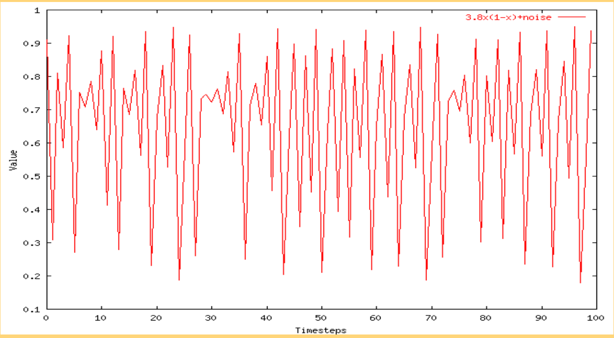
Comparison of prediction to correct values

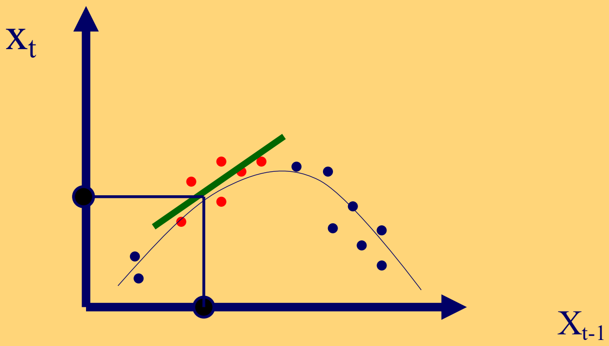




Solution

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





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.

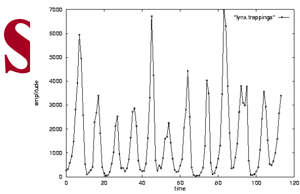
References

- 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.)

Almost Done

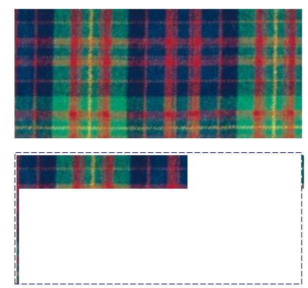
- Take-home messages:
- M1: equivalence of research problems
- M2: superset, 3h tutorial
- M3: summary of ‘recipes’

M1: 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)

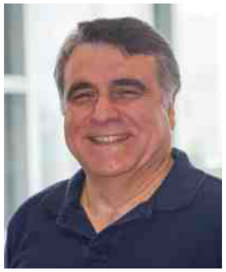
Carnegie Mellon

M2: Extended, 3 hour tutorial, KDD17

https://www.dm.sanken.osaka-u.ac.jp/~yasuko/TALKS/17-KDD-tut/

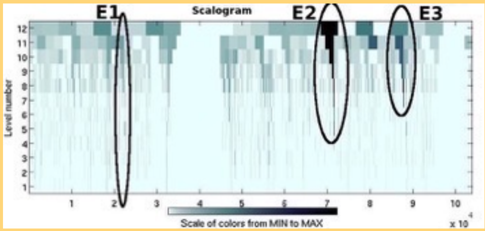
Prof. Yasushi Sakurai

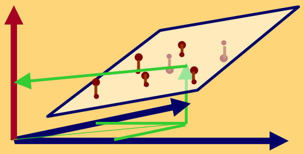
Prof. Yasuko Matsubara



M3: Time series - Answers

- P1. Similarity search: **Euclidean**/time-warping; **feature extraction** and **SAMs**
- P2. Periodicities: **DFT/DWT**
- P3. Linear Forecasting: **AR** (Box-Jenkins)
- P4. Non-linear forecasting: **lag-plots**





Time series - Answers

- P1. Similarity search

Thank you!

ありがとうございました

christos@cs.cmu.edu