Delivering Data Science
In Resources & Energy

Part 1: Evolutionary Computing

Day 11: Special Data Types

Time Series and Networks

Michael Small & Thomas Stemler

CSIRO-UWA Chair of Complex Engineering Systems
Dept. Mathematics and Statistics / Complex System Group

michael.small@uwa.edu.au
thomas.stemler@uwa.edu.au

A CORE partnership

Program Timeline

Module	Day	Date	Topic
Prerequisite	Day 1	Thurs 30 Jan 2020	Introduction to the program tools - Where Data Science comes from
Prerequisite	Day 2	Fri 31 Jan 2020	Introduction to the program tools - The Why
Introduction to Data Projects	Day 3	Thurs 6 Feb 2020	Zero to Data Science in a day
Data Analysis	Day 4	Thurs 13 Feb 2020	Getting to know the program tools - data munging and exploratory data analysis
Data Analysis	Day 5	Thurs 20 Feb 2020	Simple predictions - regression and statistical model building
Data Analysis	Day 6	Thurs 27 Feb 2020	Multivariate analysis and model building
Data & Communication Sandbox	Day 7	Fri 28 Feb 2020	Effective data storytelling - communicating results to non-technical audiences
Data Fusion and Machine Learning	Day 8	Thurs 5 Mar 2020	Pros and cons of commonly used statistical and machine learning techniques I
Data Fusion and Machine Learning	Day 9	Thurs 12 Mar 2020	Pros and cons of commonly used statistical and machine learning techniques II
Data Fusion Sandbox	Day 10	Fri 13 Mar 2020	Sandbox - Consolidate approaches covered and test on datasets
Special Data Types - Time - series Data	Day 11	Thurs 19 Mar 2020	The 4th dimension and predictions
Special Data Types - Natural Language Processing and Text Mining	Day 12	Thurs 26 Mar 2020	Finding needles in wordstacks
Special Data Types - Spatial Data	Day 13	Thurs 2 Apr 2020	Spatial analytics and predictions
Capstone Project Development & Presentation	Day 14	Fri 3 Apr 2020	Pitching Capstone Projects
Capstone Propeller	Day 15	Thurs 2 July 2020	Project Review Day

Aims & Learning Outcomes – Day 11

Aims

1. Overview of Non-Smooth Optimisation
2. Basis of modelling from time series data
3. Introduction to computation on networks
4. Application to noisy, multivariate, non-stationary data

Learning Outcomes

1. Familiarity with main techniques required for each of the areas highlighted above
2. Ability to implement and/or use relevant methods in python

11.0 Overview

11.0 Overview

What is today about?

- Dealing with special data types
- What is optimisation and why do we need it?
- General-purpose optimisation techniques
- Time series data: prediction or simulation?
- Networks (in python)
- Case study: layer detection in geophysical data

What is a “special” data type? What data do you have that is special?

11.1 Evolutionary Computing

11.1 Evolutionary Computing

Bio-inspired computing

Computer algorithms ``inspired'' by natural processes:

- artificial intelligence, cellular automata, sensor networks, excitable media, genetic algorithms

Optimisation when differentiation (i.e. $\frac{d}{dt}(\cdot)$) is hard to do(objective function is not smooth):

- genetic algorithms, ant colony optimisation, particle swarm optimisation, simulated annealing, tabu search, cultural adaption, ...

What does optimization mean for your industry?

11.1 Evolutionary Computing

Objective function --- the thing to be optimized $f(x)$ where we want to find x that makes $f(x)$ as big/small as possible

Solution space $\mathcal{X}$ --- the set of all possible solutions x

Genetic Algorithm

1. Choose a collection of possible solutions $\{x_1, x_2, x_3, \ldots\} \subseteq \mathcal{X}$
2. Evaluate the fitness $f(x_i)$ of each x_i
3. Breed them, mutate them and apply evolutionary pressure (kill off less fit) to get a group of children $\{y_1, y_2, y_3, \ldots\} \subseteq \mathcal{X}$ from the parents $\{x_1, x_2, x_3, \ldots\}$
4. Replace the parents with the children and repeat from 2.

11.1 Evolutionary Computing

Genetic Algorithms

Good for:

- difficult to evaluate objective functions
- high dimensional systems or non-smooth system
- discrete, categorical, or mixed variables
- paralellisation
- problems when we don't know much about the objective
- “modular” problems

Bad for:

- smooth problems (use calculus)
- stochastic/noisy objective functions
- very high dimensional systems
- poorly parametrised problems (the “genes” don't work)
- Binary/categorical objectives
- naiveté

11.1 Evolutionary Computing

Implementation (example)

1. Define/determine objective function $f(x) = \frac{\cos\frac{1}{x}}{x}$
2. Encode $x \in \mathcal{X}$ as a gene consisting of building blocks $x \in [-1,1]$ where $x = (-1)^{b_0} \times b_1\ b_1\, b_2\, b_3 \dots b_{63}$ is a sign bit and a sequence of 63 binary bits.
3. Generate random initial population solutions $\{x_1, x_2, x_3, \dots\} \subseteq \mathcal{X}$
4. Compute the fitness $f(x_i)$ of each x_i
5. For each child y_i select two parents at random with probability proportional to $f(x_i)$ (their fitness) and perform cross-over mutation
6. Randomly mutate each gene of each child with probability m where $0 < m \ll 1$.
7. Replace the parents with the offspring and repeat from 4.

11.1 Evolutionary Computing

Pythonification

DIY is probably best, but if you insist:

- GAFT (on github)
- DEAP (on github)
- pyevolve (on github)
- Pyvolution (pypl.org)

11.1 Evolutionary Computing

Exercise

- Examine the provided code.
- What effect does mutation have on the speed of result?
- Modify to preserve the fittest
- 2D optimisation problem
- optimise networks and/or time series models

11.1 Evolutionary Computing

References

- Travelling salesman in python https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35
- Clinton Sheppard ``Genetic Algorithms with Python''.
- https://www.codeproject.com/Articles/1104747/Introduction-to-Genetic-Algorithms-with-Python-Hel also by Clinton Sheppard

11.2 Time Series

11.2 Time Series

Prediction

- Given a history $\{\ldots, x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t\}$ what is the future $\{x_{t+1}, x_{t+2}, x_{t+3}, \ldots\}$?
- … and, how good are these predictions?
- E.g. 5,-3,2,-1,1,0,1,1,2,3 --- what is next?
- The objective is to build $F(\ldots, x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t) \approx x_{t+1}$.

When would you need prediction in your industry? Prediction of what?

11.2 Time Series

Linear autoregressive models

- $F(x_{t-d+1}, x_{t-d+2}, \dots x_{t-1}) = a_1 x_{t-1} + a_2 x_{t-2} + \dots + a_d x_{t-d} \approx x_{t+1}$.
- … or $F(X) = X.A \approx y$ where X is a matrix and A and y are vectors:

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_d \\ x_2 & x_3 & \cdots & x_{d+1} \\ \vdots & & \ddots & \vdots \\ x_{n-d+1} & x_{n-d+2} & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix} \approx \begin{bmatrix} x_{d+1} \\ x_{d+2} \\ \vdots \\ x_n \end{bmatrix}$$

- I.e. $\hat{A} = X^{-1} y$
- Only question that remains is how to choose d (and whether models like this are useful at all).

11.2 Time Series

Local modelling --- "Weather Forecasting"

1. Construct a library of past observations and the past-futures of those past observations. (I.e. at time x_t the corresponding future was f_t).
2. For each new observation x find past similar observations and use their futures to predict the true future

   I.e. (last time it was cloudy, cold and windy it also rained)

- Also called *machine learning/deep learning/artificial intelligence*.
- Generalisation involve building more complex (or probabilistic) predictions by combining multiple similar points in the past

Nonlinear global models

1. let $v_t = (x_t, x_{t-\ell_1}, x_{t-\ell_2}, x_{t-\ell_3}, \dots x_{t-\ell_d})$
2. choose/select/guess a set of nonlinear functions ϕ_i of the data
3. build a nonlinear

$$F(x_t, x_{t-1}, x_{t-2}, \dots) = \sum_{i=1}^{M} \lambda_i \phi_i(v_t)$$

4. then this can be written as matrices $\Phi\Lambda = y$ and $\Lambda = \Phi^{-1}y$.
5. *but* we now need to make good choices of functions ϕ_i and dimension M.

11.2 Time Series

Pythonification

- An exercise for the reader…
- Do this.
- Which prediction schemes work best?
- When and why?

11.2 Time Series

References

- H. Kantz, T. Schreiber. ``Nonlinear time series analysis'' CUP.
- M. Small. ``Applied Nonlinear time series analysis'' World Scientific.
- https://towardsdatascience.com/time-series-analysis-in-python-an-introduction-70d5a5b1d52a
- https://jakevdp.github.io/PythonDataScienceHandbook/03.11-working-with-time-series.html

11.3 Networks

11.3 Networks

Exploring your LinkedIn Network

- Log into LinkedIn and go to Me->Settings and Privacy->Privacy->How LinkedIn uses your data->Getting a copy of your data
- Enter your details and Download archive (it'll take a while)
- Credit: towardsdatascience.com

You'll need this file at the end of this session.

11.3 Networks

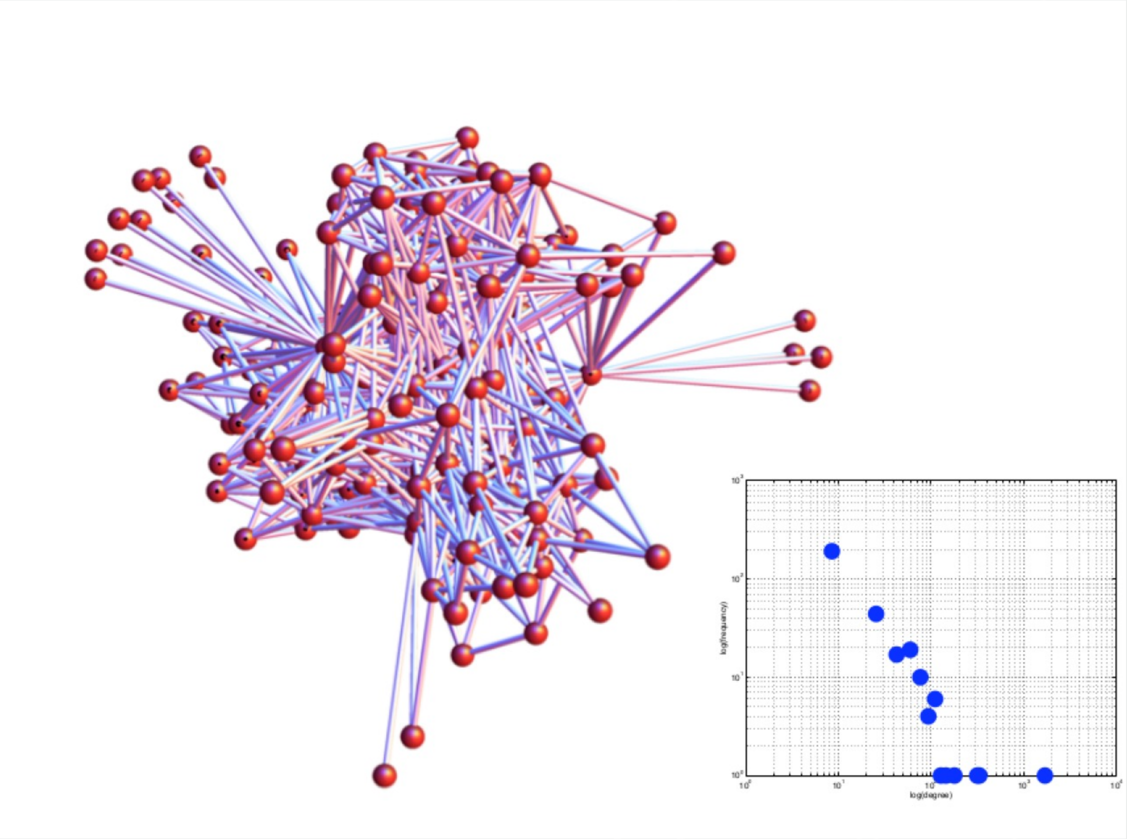
Discussion

- Where do you have networks in your business?

11.3 Networks

Network Quantification

- Degree sequence
- Degree histogram
- Path-length
- Clustering
- Assortativity
- Betweenness Centrality
- Eigenvalue Centralitiy
- Modularity and Communities
- Hubs and richclubs
- Robustness and fragility
- Motifs and superfamily



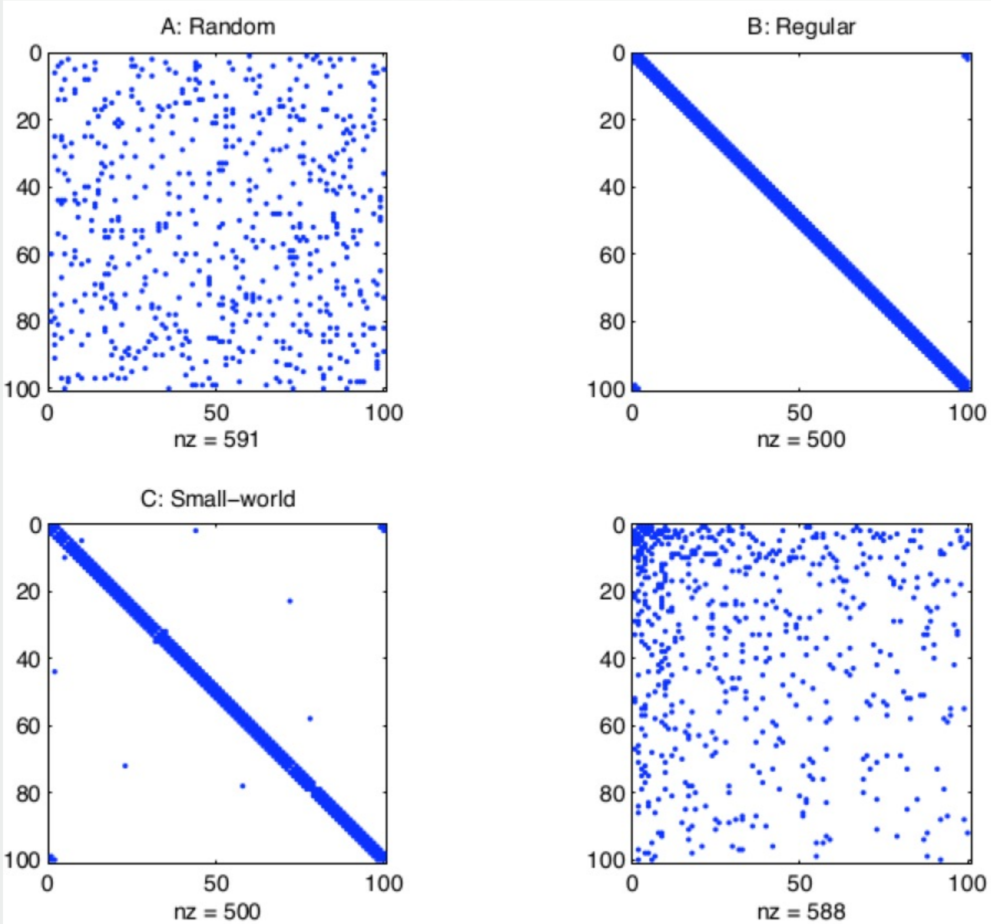
Michael Small, CSIRO & UWA

11.3 Networks

Adjacency Matrix

- Let A be a binary $N \times N$ matrix with $a_{ij} = (A)_{ij} := 1$ if and only if there is a link from node-i to node-j (otherwise $A_{ij} = 0$). Generalise to weighted networks where $a_{ij} = w_{ij}$ is the weight, and directed networks with an asymmetric A.
- The graph Laplacian L is defined so that $L = A$ except for the diagonal:

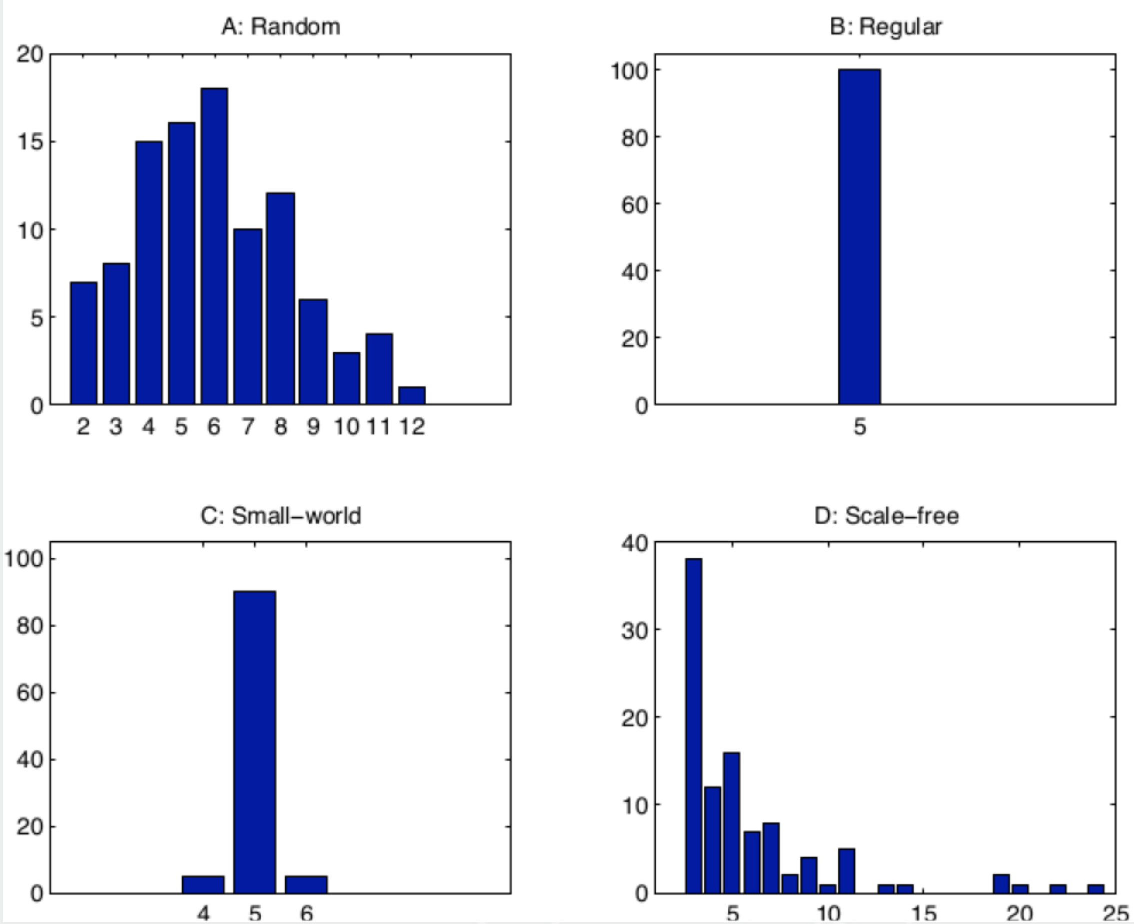
$$(L)_{ii} = \sum_{j \neq i} -a_{ij} \, .$$



11.3 Networks

Degree Distribution

- Compute the number of nodes n_k (or the probability p_k)with degree k.

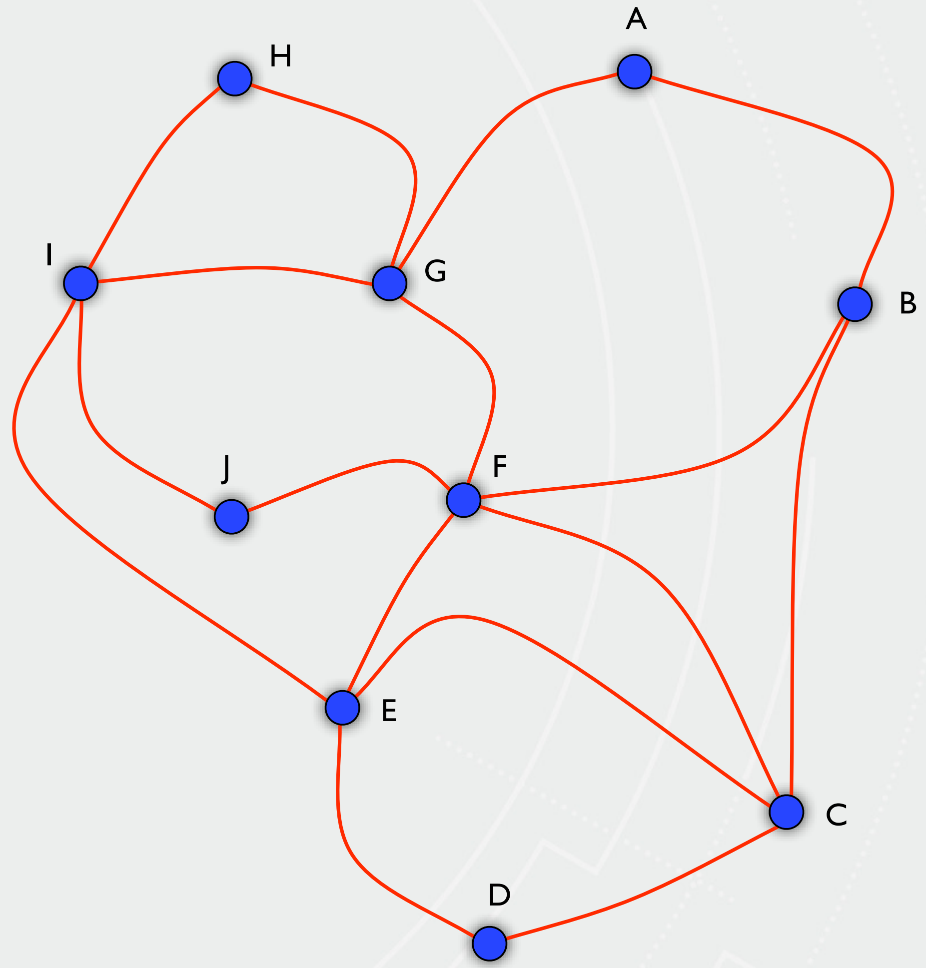


Michael Small, CSIRO & UWA

11.3 Networks

Path-length

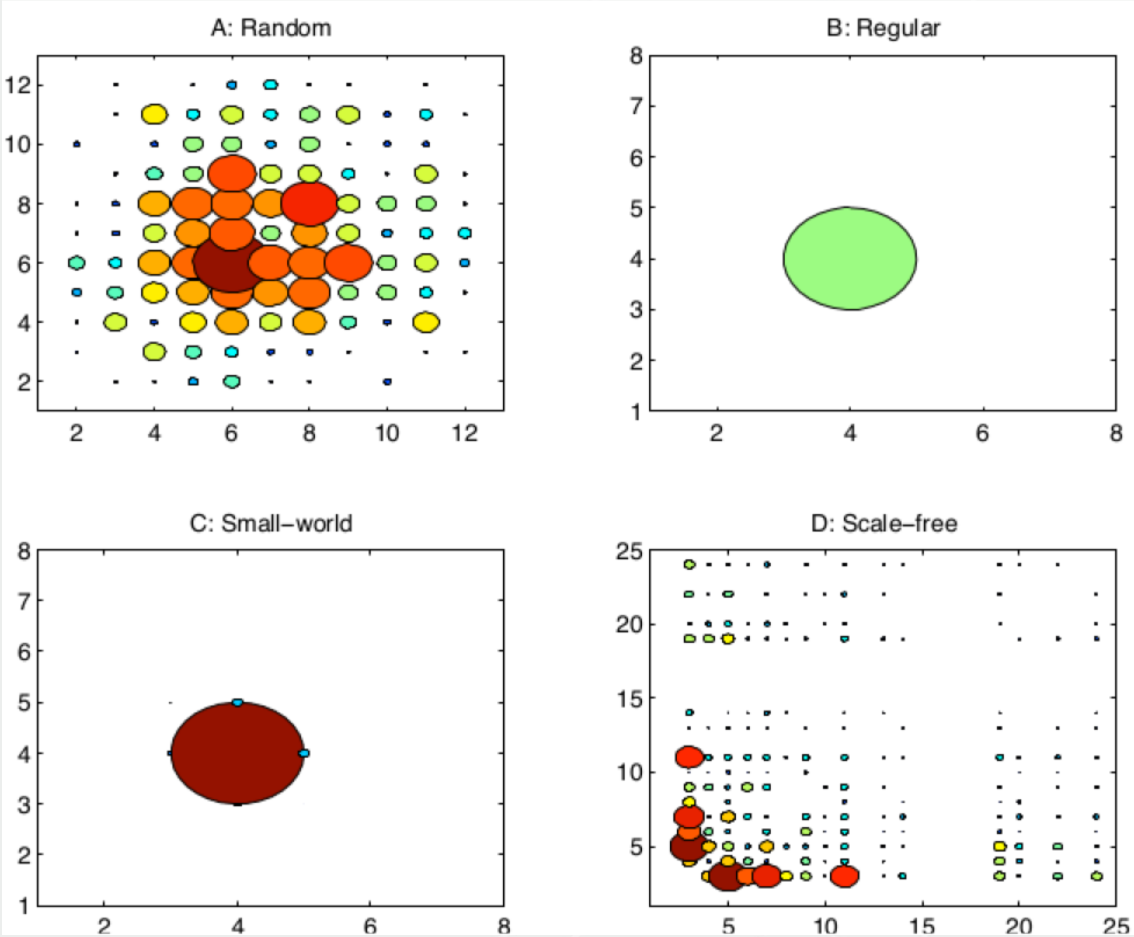
- *Path-length*: The path-length is the shortest path (number of edges traversed) between two nodes.
- *Diameter*: The maximum path-length



11.3 Networks

Clustering and Assortativity

- *Clustering*: The number of triangles --- the probability of neighbours being neighbours.
- *Assortativity*: The linear (Pearson) correlation between pairs of nodes with a give property

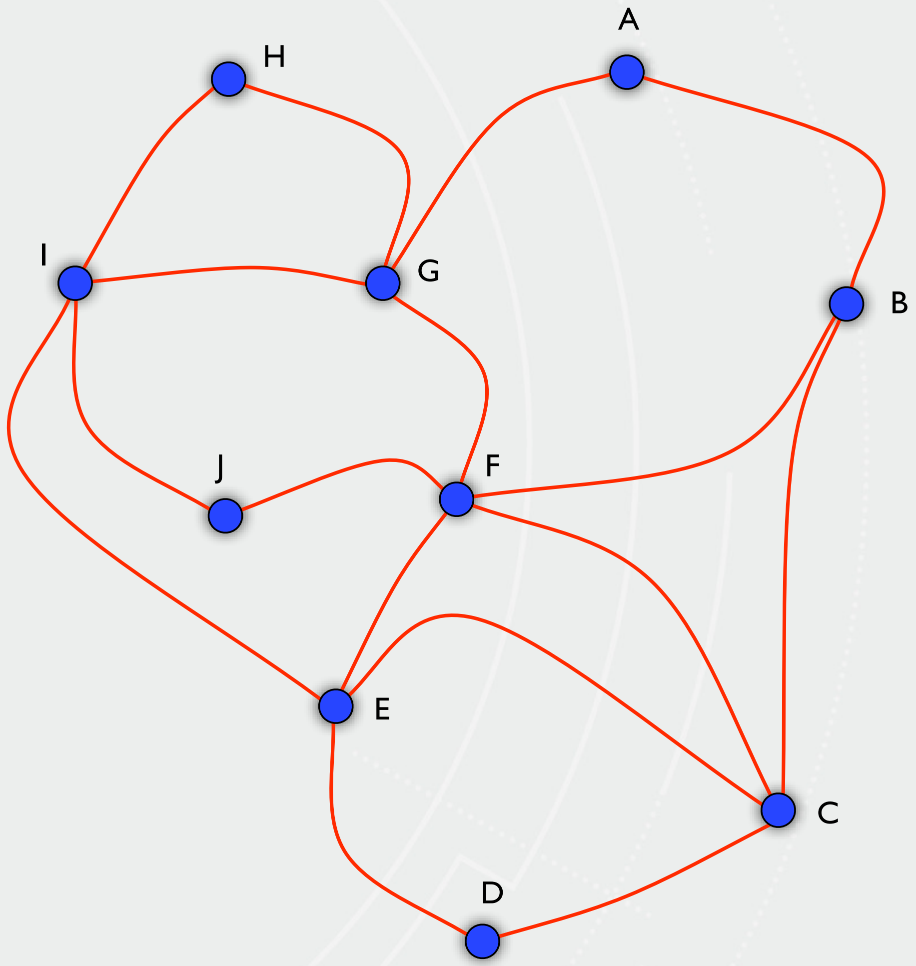


Michael Small, CSIRO & UWA

11.3 Networks

Centrality

- *Betweenness Centrality*: The number (fraction) of shorter paths passing through a given node
- *Eigenvalue Centrality*: (AKA: Google's PageRank) Eigenvalue decomposition of the Laplacian

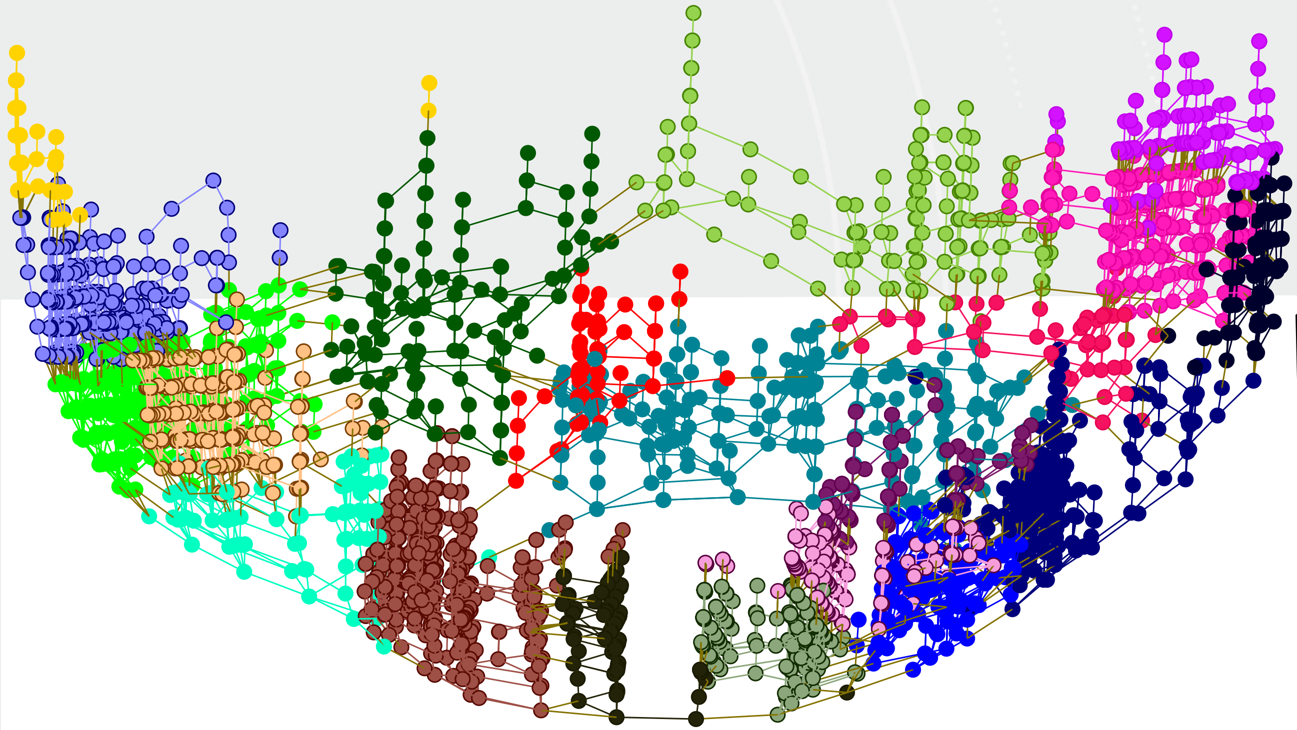


11.3 Networks

Community Detection

- *Communities* should have more links between members than between communities.
- *Modularity* Q measures this:

$$Q = \frac{1}{2m}\sum_{i,j}\left(a_{ij} - \frac{k_i k_j}{2m}\right)\delta(i,j)$$



Michael Small, CSIRO & UWA

11.3 Networks

Pile Higher and Deeper

- *Hubs* are nodes with the highest degree
- *Rich-club* is the connection between the hub nodes and the tendency of hub nodes to be connected to one another
- *Giant component*: The property that most of the nodes are connected (directly or indirectly) to one another.
- *Robustness*: The ability of a network to maintain it's giant component even after random removal of a relatively large number of edges (or, equivalently, nodes).
- *Fragility*: The corresponding inability of a network to maintain the rich club under targeted removal of edges (or nodes)
- *Motifs*: The structure of interconnection in sub-graphs of particular size k.
- *Super-family*: The relative frequency of all such sub-graphs for fixed k.
- *Random Graph Theory*, *Simplicial Complexes* and *Persistent Homology*: It's complicated.

11.3 Networks

Discussion

- Why would *you* need to know any of this stuff for your business?

Pythonification

- Everything we've discussed here is implemented and easily usable within the `networkx` package. Refer to the notebook

11.3 Networks

References

- C. Braham and M. Small. ``Complex networks untangle competitive advantage in Australian football'' *Chaos* **28** (2018) 053105.
- Barabasi, Network Science, http://networksciencebook.com/
- https://www.python-course.eu/networkx.php
- M. Small. Dynamics of biological systems (network chapter).

11.4 Multivariate, Noisy and Nonstationary

11.4 Multivariate, Noisy and Nonstationary

- Multivariate: multiple measures, several variables that represent complex system or phenomena.
- Noisy: errors in data collection, storage and processing which are meaningless information.
- Nonstationary: mean and variance change over time, difficult to model and predict.

Where and when do you face such type of data in your industry? Everywhere, always?

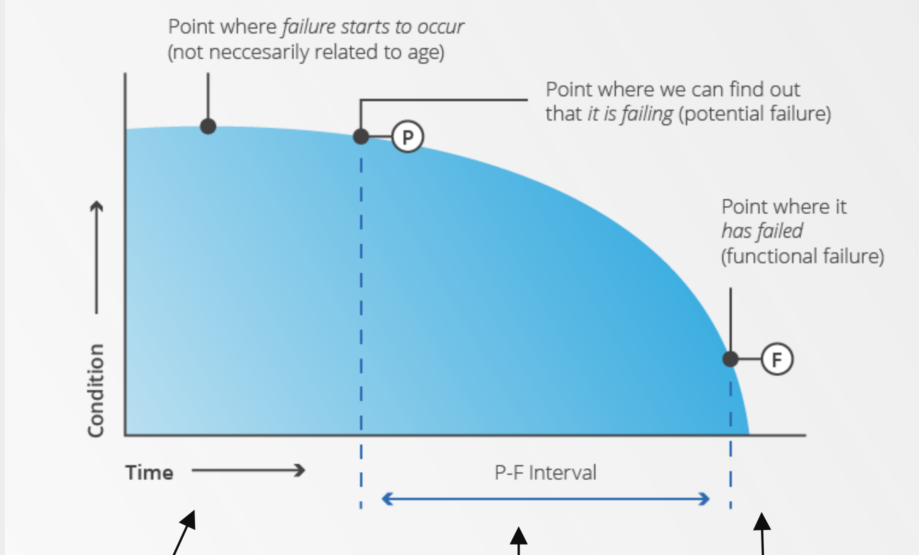
11.4 Motivation: Detect transition in multivariate, noisy and nonstationary data

Source: http://lase.de

Source: http://tenova.com

Michael Small, CSIRO & UWA

Source: http://maintworld.com



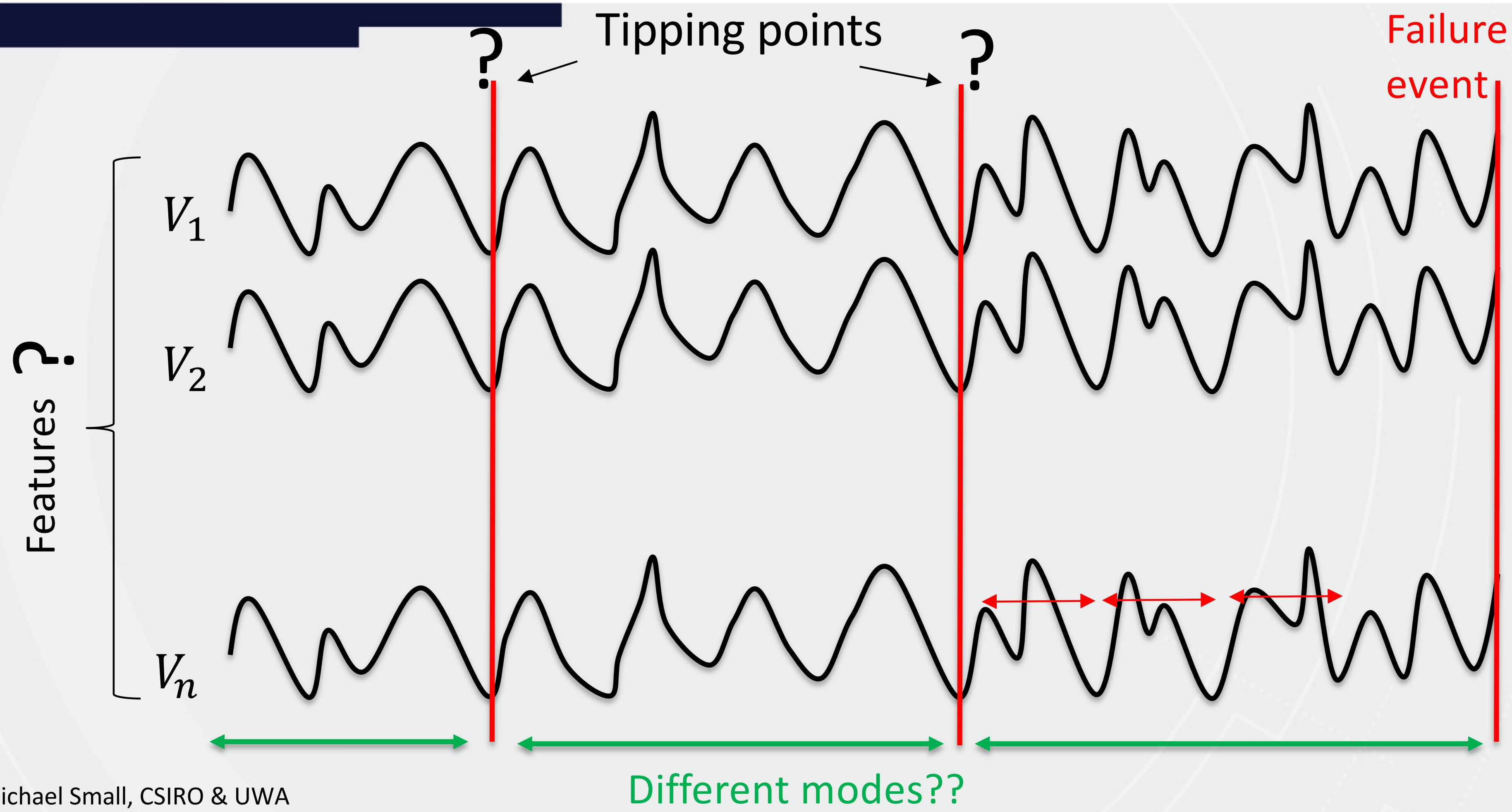
Predictive

Proactive

Reactive

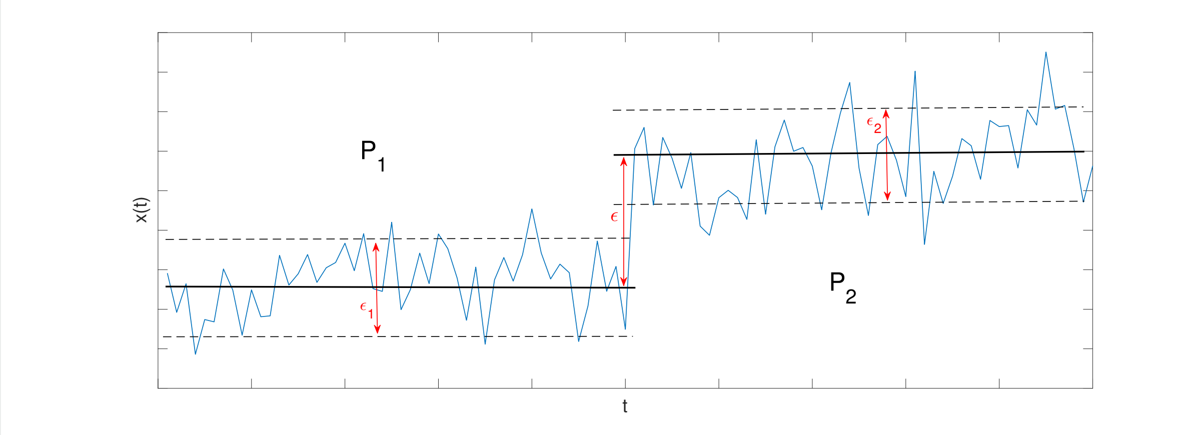
Objective: Develop a tool to allow early detection (prediction) of failures from condition/performance (nearly) real-time monitoring.

11.4 Motivation: Detect transition in multivariate, noisy and nonstationary data

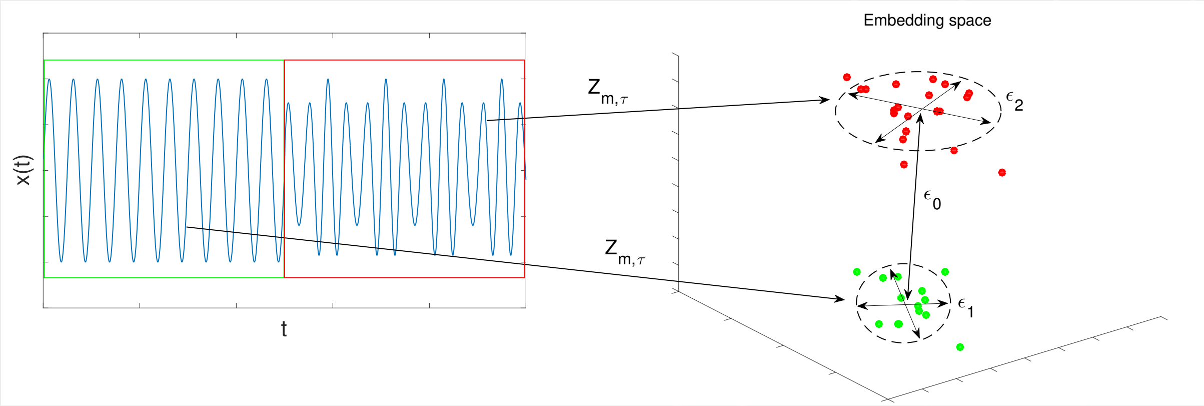


11.4 Type of transitions

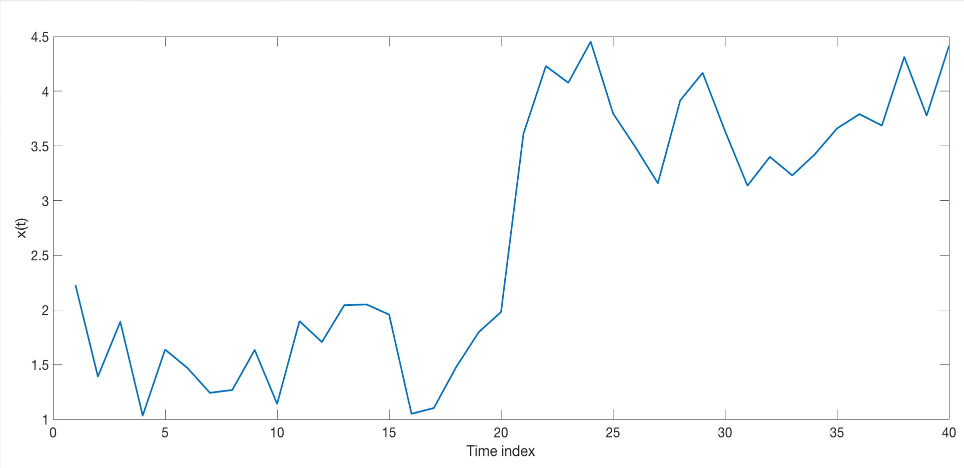
Type 1: State transition



Type 2: Dynamic transition

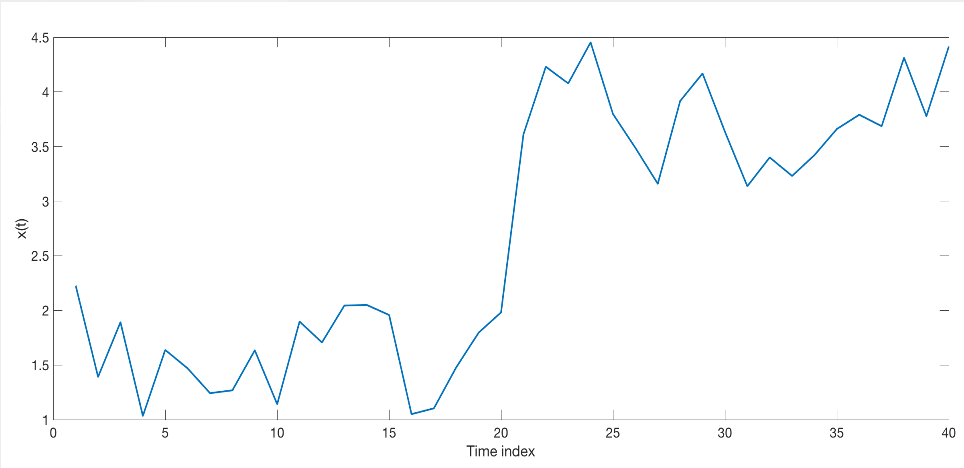


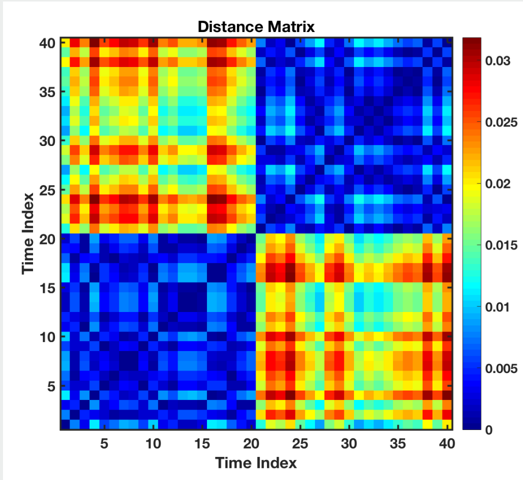
11.4 Recurrence Plot and Quadrant Scan to detect transition



Let $x(t) \in \mathbb{R}$ (could be $\mathbb{R}^m$ for embedded series or multivariate input) be the state of the system at time t where $\mathrm{t} = 1, 2, \ldots, N$.

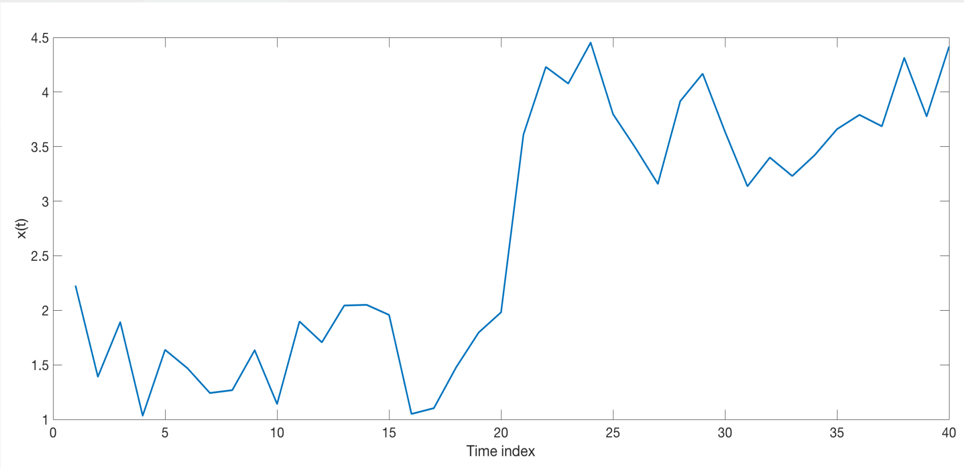
11.4 Recurrence Plot and Quadrant Scan to detect transition

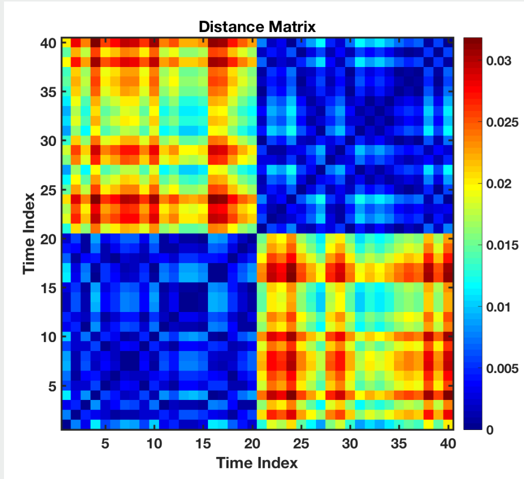


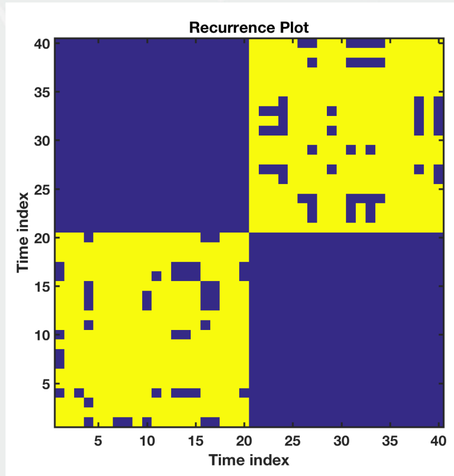


We construct an $N \times N$ norm matrix A with $a_{ij} = d(x(i), x(j))$. Here $d(.,.)$ is some norm (Euclidian norm in many applications). If $m > 1$ (multivariate input), then a sort of normalisation is required to avoid the dominance of measurements with large scale on the results.

11.4 Recurrence Plot and Quadrant Scan to detect transition





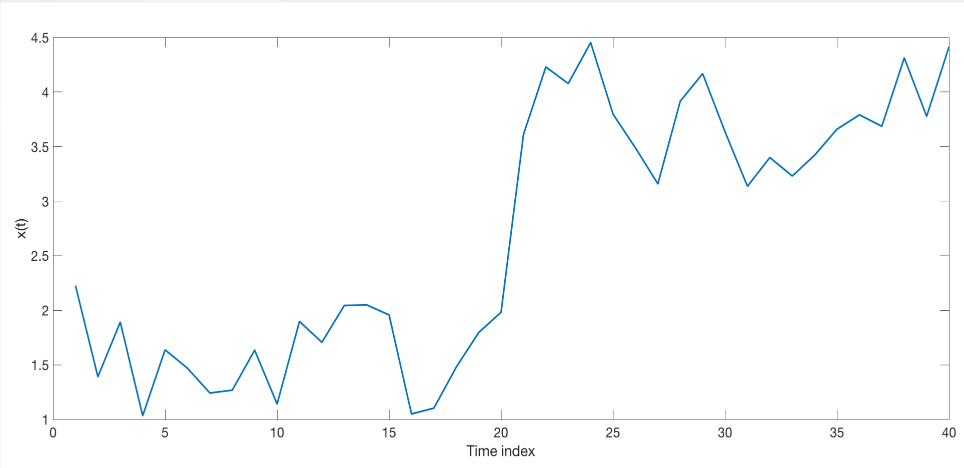


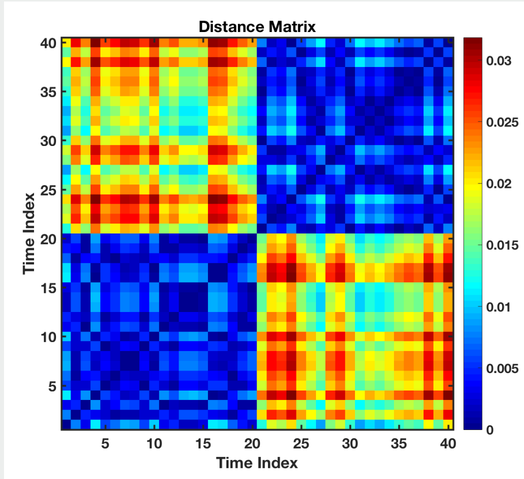
From the norm matrix, we construct an $N \times N$ ***Recurrence Plot*** matrix R with $r_{ij} = 1$ iff $a_{ij} < \epsilon$ and 0 otherwise. To consider multi-scale transition detection, we adopt the recurrence plot threshold by looking at the distribution of the elements of A:

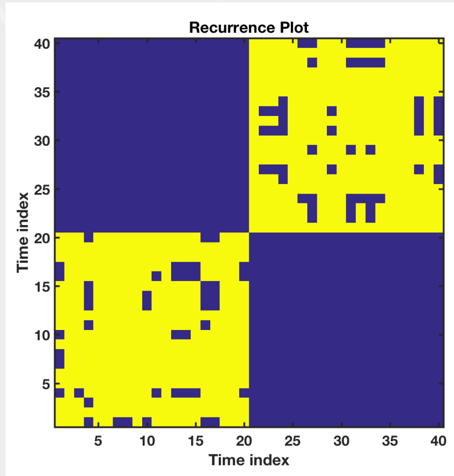
$$\epsilon = \alpha \times \left(mean(a_{ij}) + 3 \times std(a_{ij})\right), \text{where } 0 < \alpha < 1$$

Michael Small, CSIRO & UWA

11.4 Recurrence Plot and Quadrant Scan to detect transition

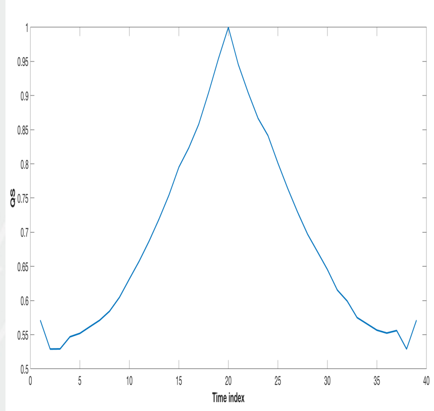






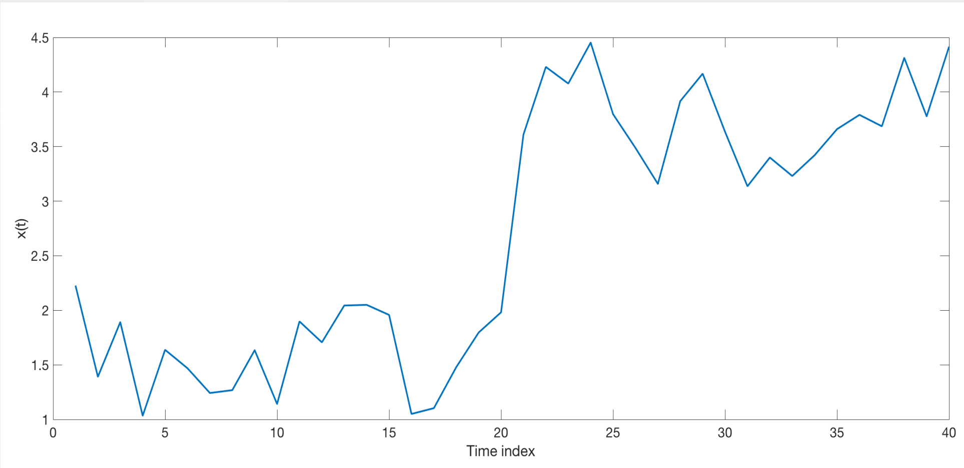
From the recurrence plot matrix R ($\epsilon = \alpha \times \left(mean\left(a_{ij}\right) + 3\times std\left(a_{ij}\right)\right)$, where $0 < \alpha < 1$), we extract the Quadrant Scan sequence $\mathrm{QS}(t) = \frac{D_{1,3}}{D_{1,3}+D_{2,4}}$

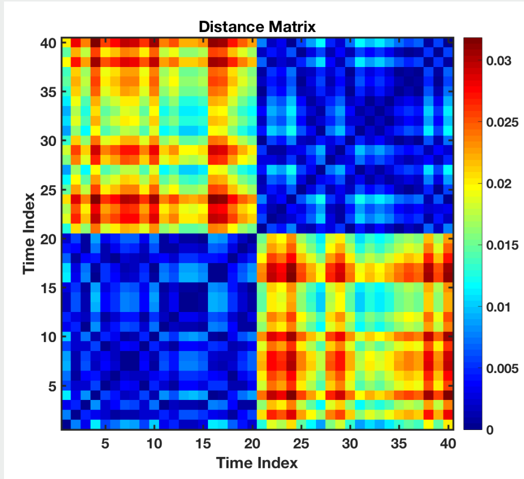
Where $D_{1,3} = \frac{\sum_{i,j\le t} r_{ij} + \sum_{i,j>t} r_{ij}}{t^2+(N-t)^2}$, $D_{2,4} = \frac{\sum_{i\le t,j>t} r_{ij} + \sum_{i>t,j\le t} r_{ij}}{t\times(N-t)\times 2}$

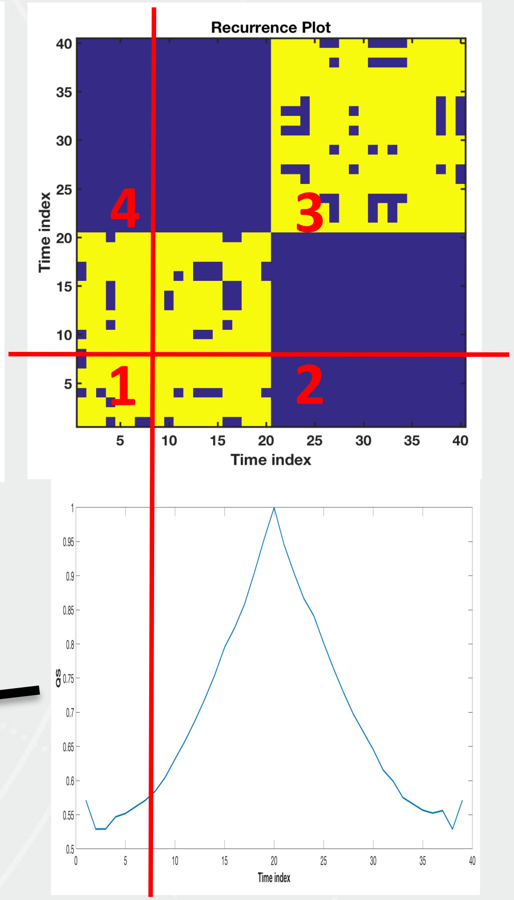


Michael Small, CSIRO & UWA

11.4 Recurrence Plot and Quadrant Scan to detect transition





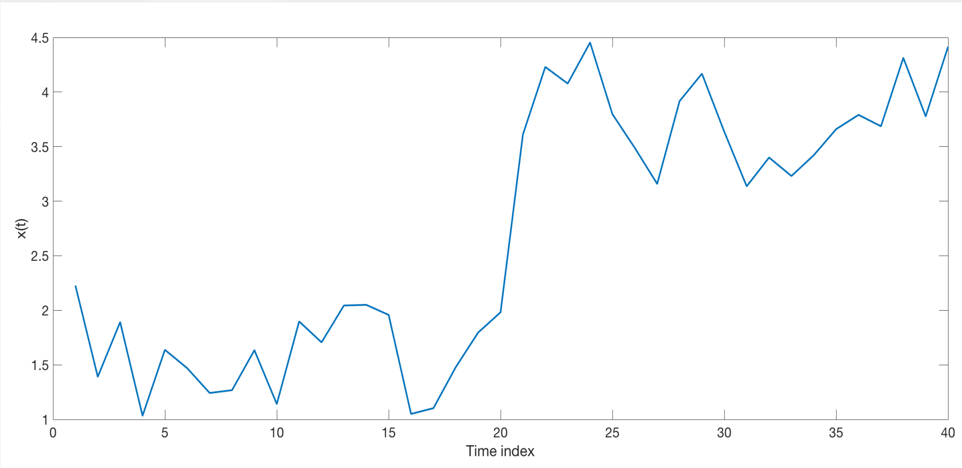


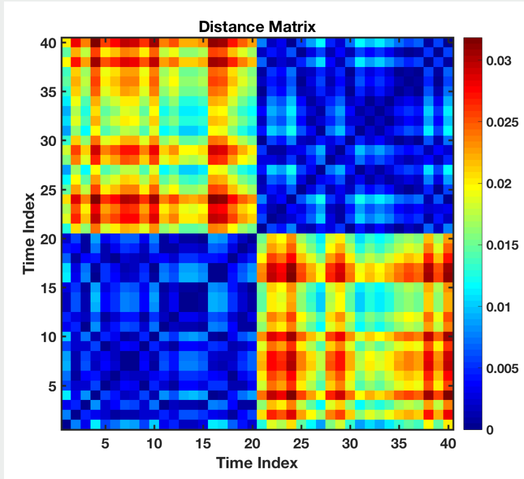
From the recurrence plot matrix R ($\epsilon = \alpha \times \left(mean\left(a_{ij}\right) + 3\times std\left(a_{ij}\right)\right)$, where $0 < \alpha < 1$), we extract the Quadrant Scan sequence $\mathrm{QS}(t) = \frac{D_{1,3}}{D_{1,3}+D_{2,4}}$

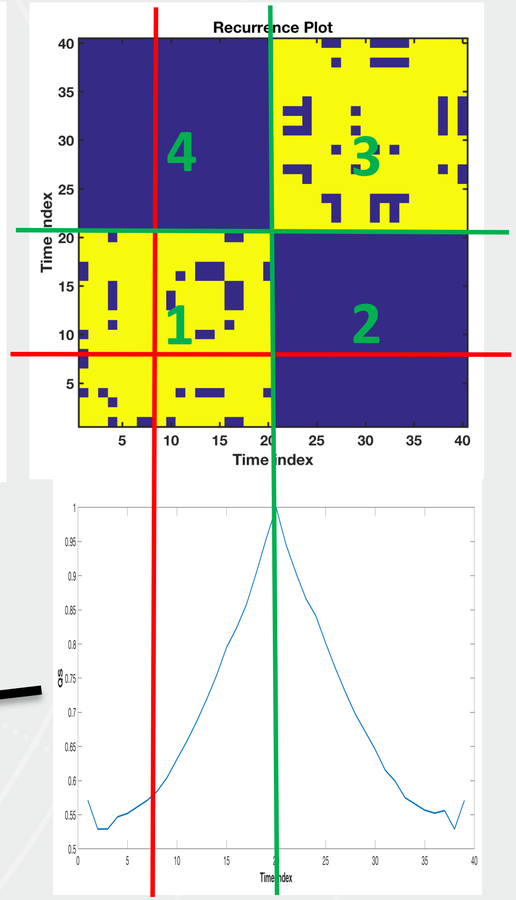
Where $D_{1,3} = \frac{\sum_{i,j\le t} r_{ij} + \sum_{i,j>t} r_{ij}}{t^2+(N-t)^2}$, $D_{2,4} = \frac{\sum_{i\le t,j>t} r_{ij} + \sum_{i>t,j\le t} r_{ij}}{t\times(N-t)\times 2}$

Michael Small, CSIRO & UWA

11.4 Recurrence Plot and Quadrant Scan to detect transition





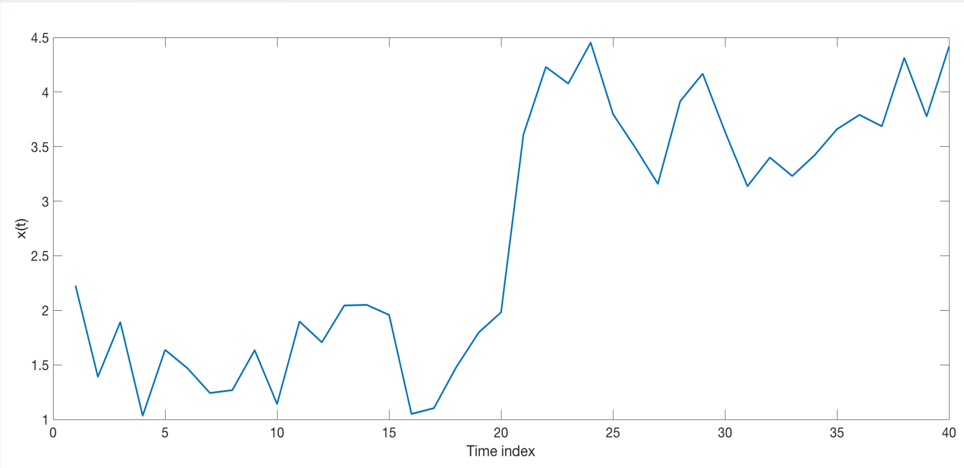


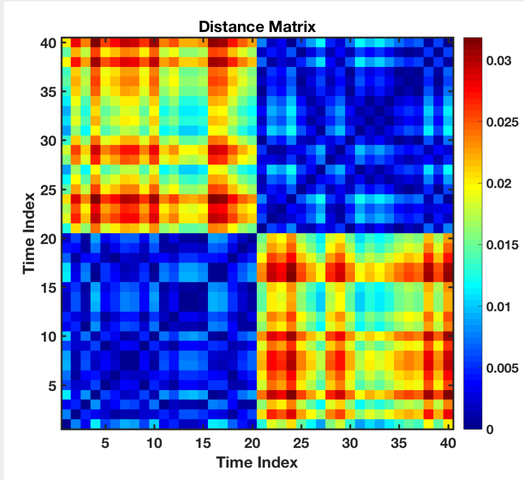
From the recurrence plot matrix R ($\epsilon = \alpha \times \left(mean(a_{ij}) + 3 \times std(a_{ij})\right)$, where $0 < \alpha < 1$), we extract the Quadrant Scan sequence $QS(t) = \frac{D_{1,3}}{D_{1,3}+D_{2,4}}$

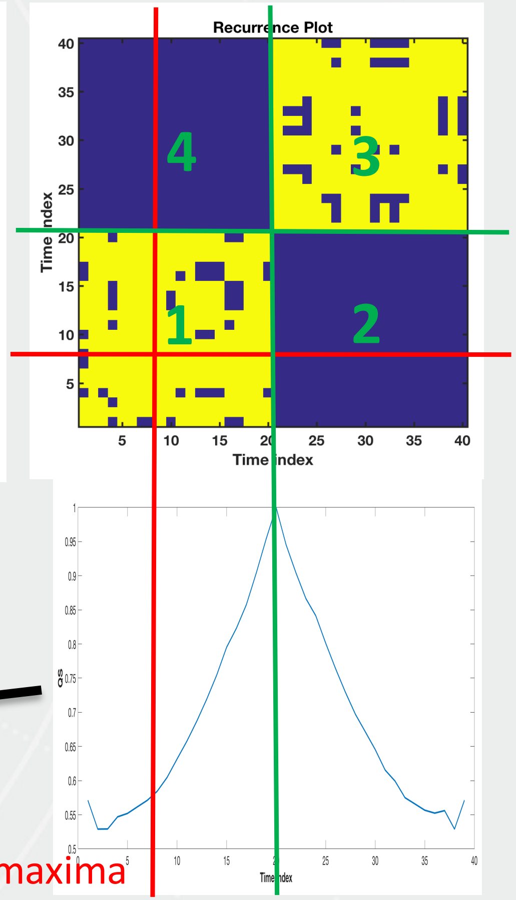
Where $D_{1,3} = \frac{\sum_{i,j \le t} r_{ij} + \sum_{i,j>t} r_{ij}}{t^2+(N-t)^2}$, $D_{2,4} = \frac{\sum_{i \le t, j>t} r_{ij} + \sum_{i>t, j \le t} r_{ij}}{t \times (N-t) \times 2}$

Michael Small, CSIRO & UWA

11.4 Recurrence Plot and Quadrant Scan to detect transition







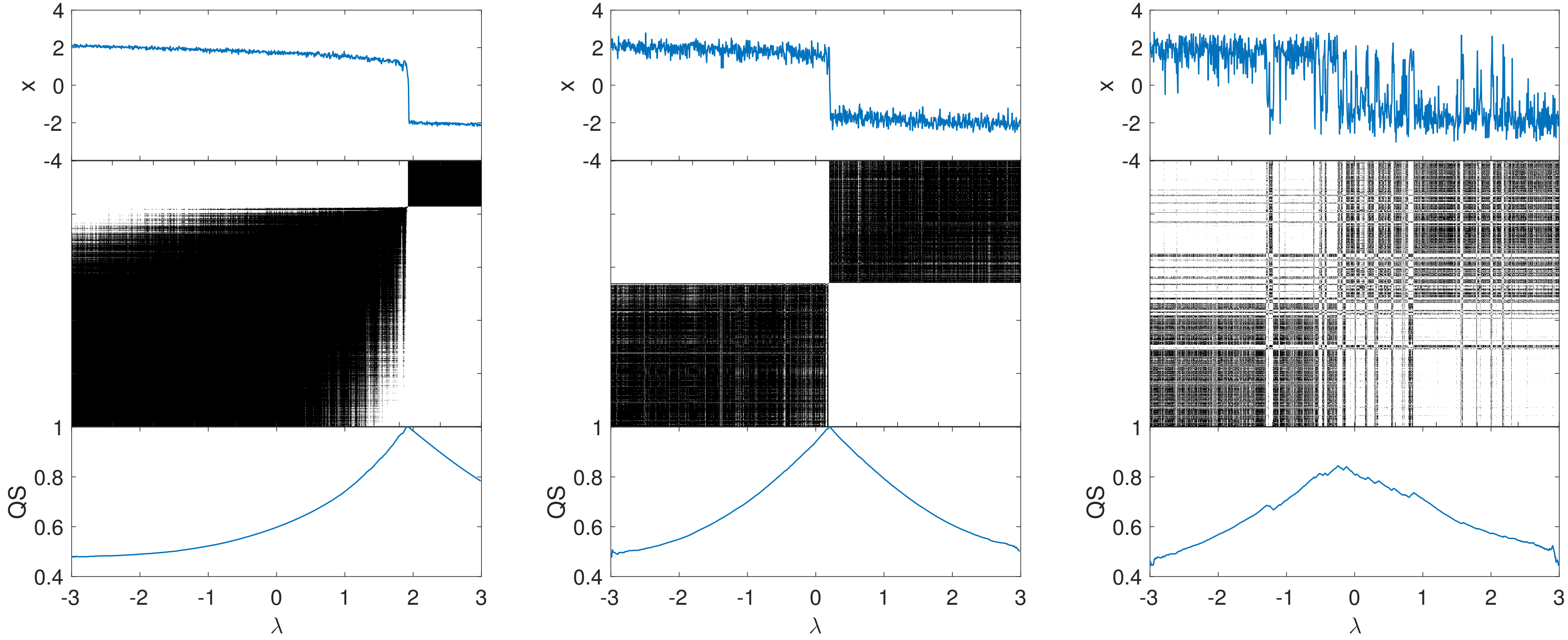
From the recurrence plot matrix R $\left(\epsilon = \alpha \times \left(mean(a_{ij}) + 3 \times std(a_{ij})\right)\right.$, where $0 < \alpha < 1$), we extract the Quadrant Scan sequence $QS(t) = \frac{D_{1,3}}{D_{1,3}+D_{2,4}}$

Where $D_{1,3} = \frac{\sum_{i,j \le t} r_{ij} + \sum_{i,j>t} r_{ij}}{t^2+(N-t)^2}, D_{2,4} = \frac{\sum_{i \le t, j>t} r_{ij} + \sum_{i>t, j \le t} r_{ij}}{t \times (N-t) \times 2}$

Michael Small, CSIRO & UWA

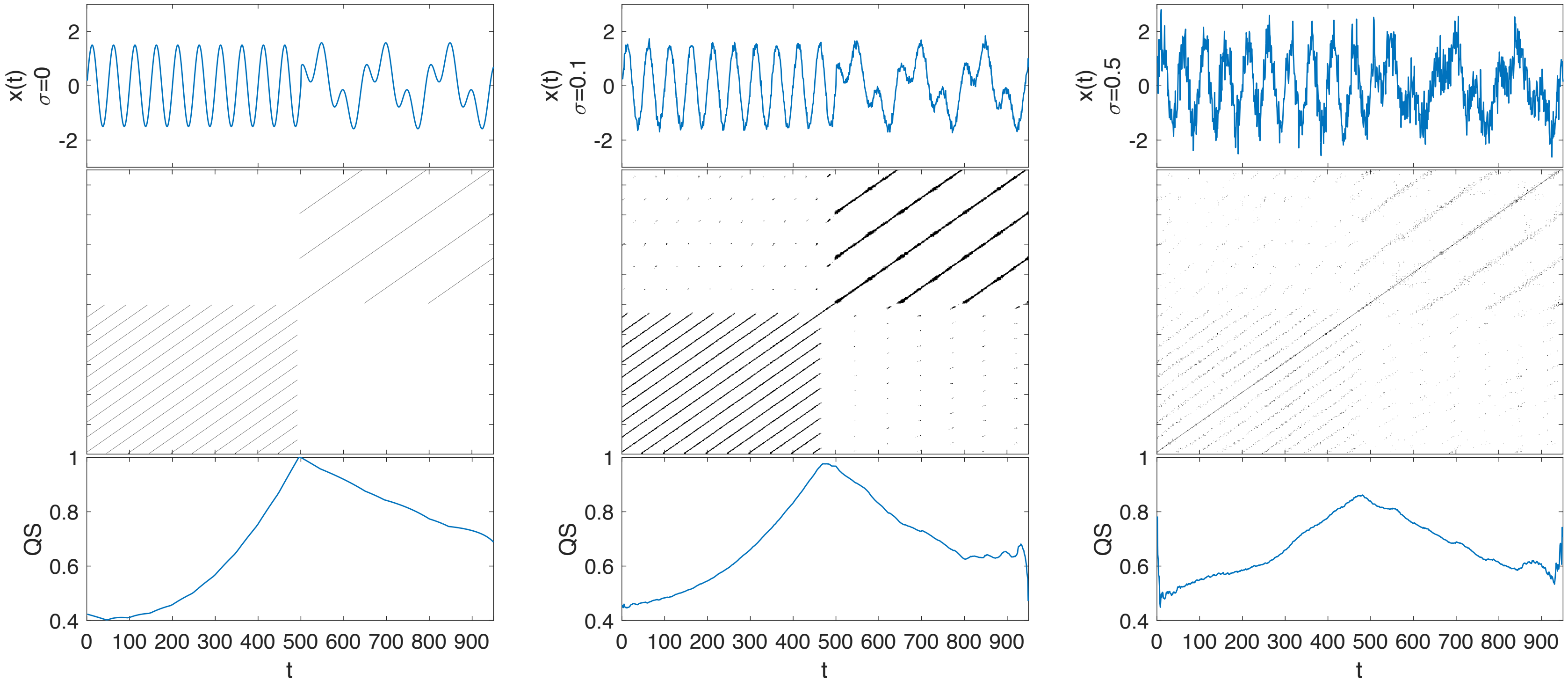
The value of the of the Quadrant Scan is between 0 and 1, maxima (peaks) of $QS(t)$ identify transitions in the system.

11.4 Synthetic example of type 1 (state transition): Noisy stochastic system



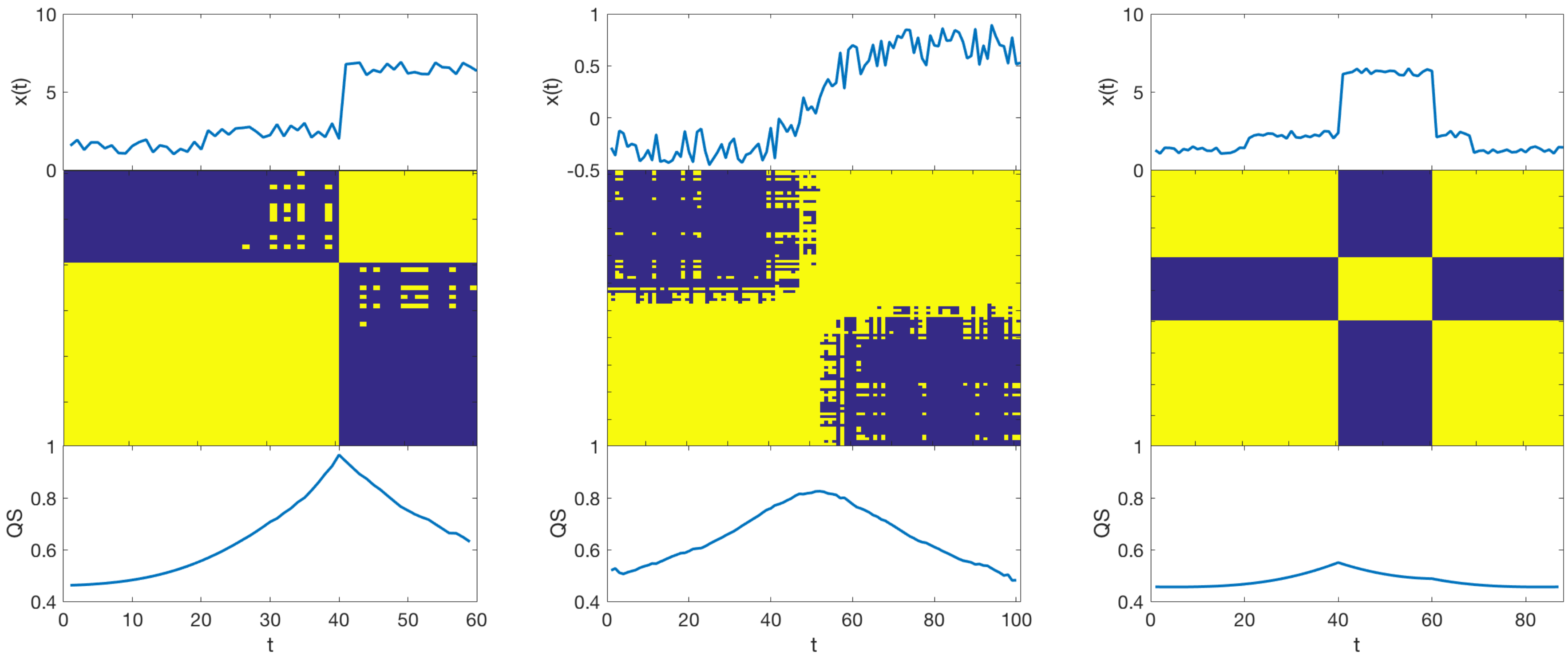
Michael Small, CSIRO & UWA

11.4 Synthetic example of type 2 (dynamic transition): Noisy periodic system



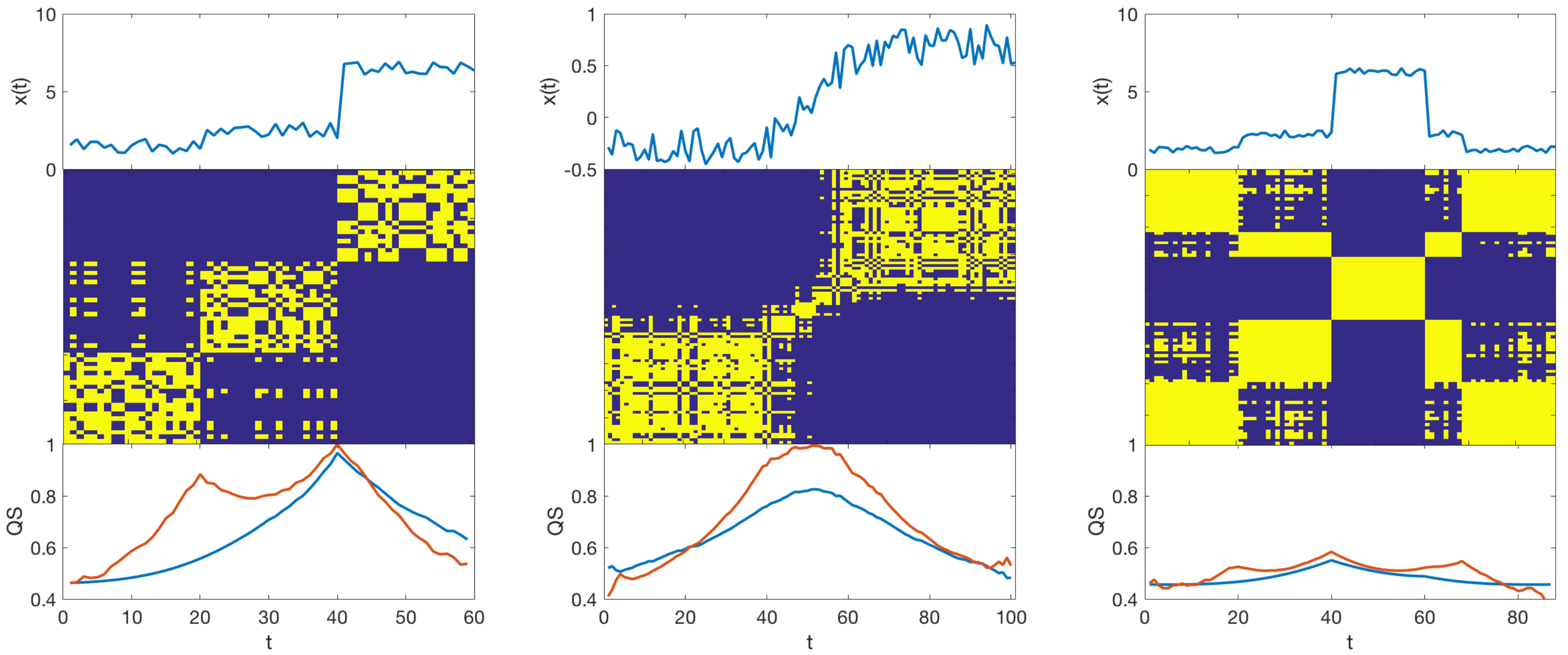
11.4 Testing different scenarios

Large threshold



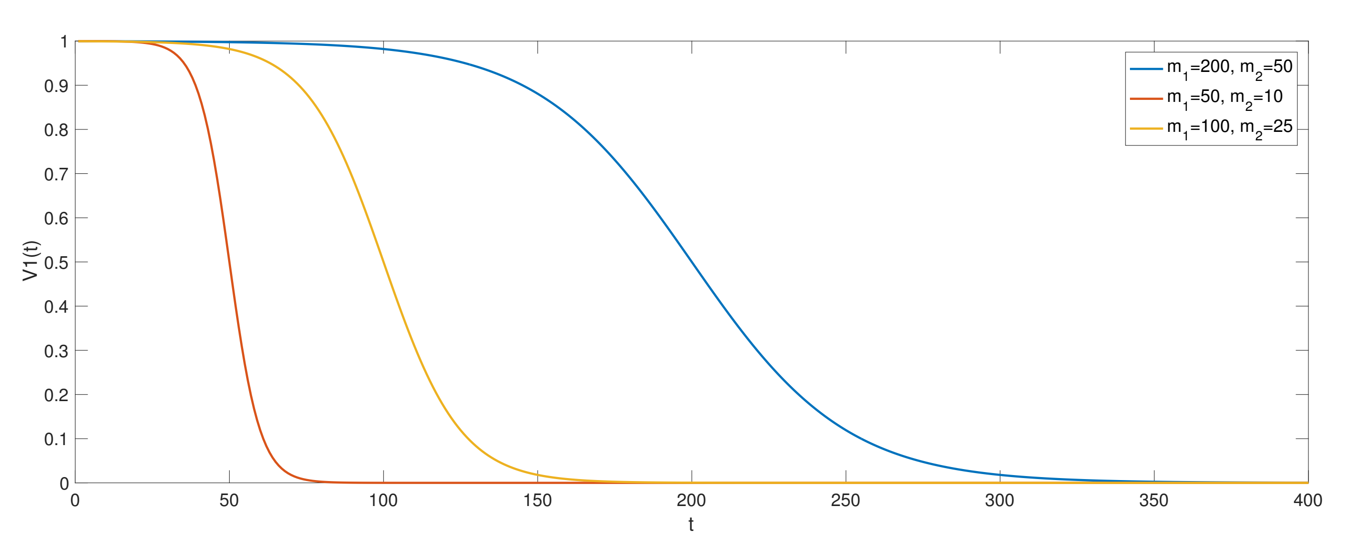
11.4 Testing different scenarios

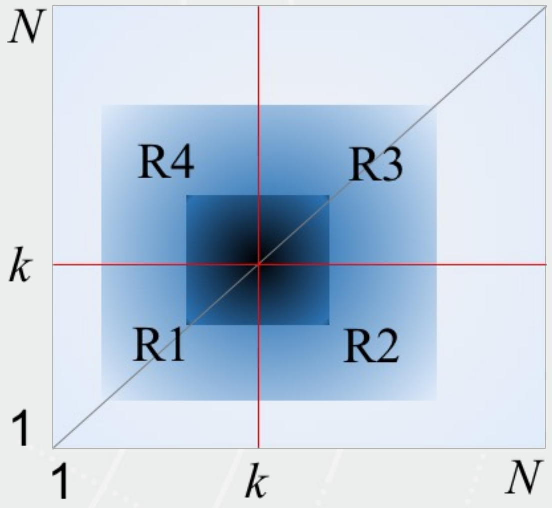
Smaller threshold



11.4 Weighted Quadrant Scan

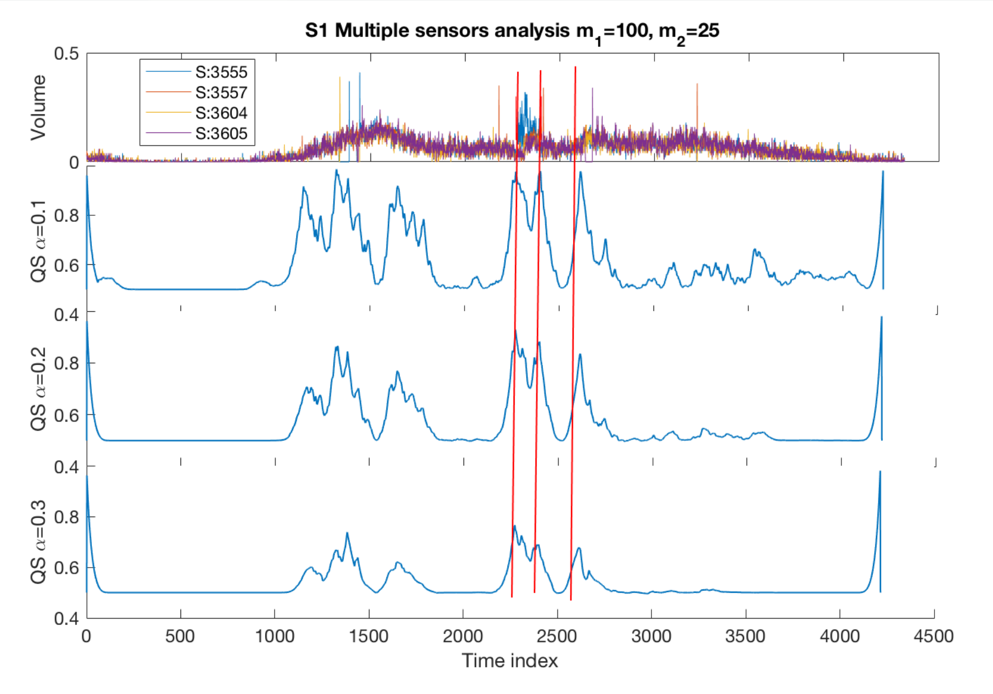
$$V1(t) = \frac{1}{2}(1 - \tanh(\frac{t - m_1}{m_2}))$$

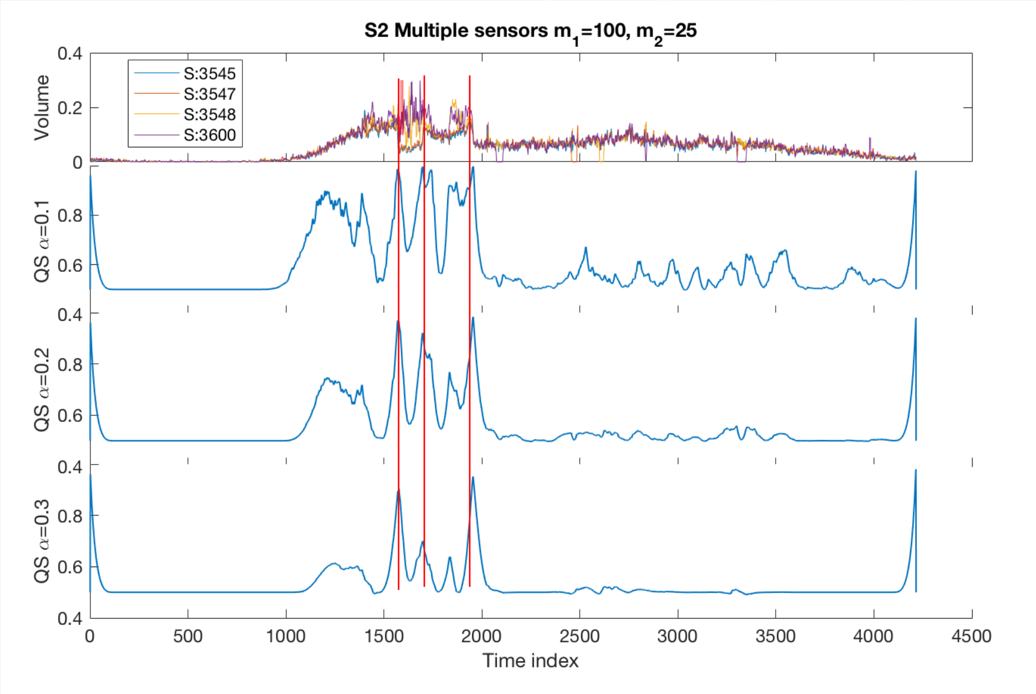




Michael Small, CSIRO & UWA

11.4 Real-world: Multivariate, noisy and nonstationary traffic data





Michael Small, CSIRO & UWA

11.4 Your turn

Case study

- Layer detection in multivariate geophysical data
- Refer to python notebook

11.4 Your turn

References

- Zaitouny, A., Walker, D.M. and Small, M., 2019. Quadrant scan for multi-scale transition detection. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *29*(10), p.103117.
- Zaitouny, A., Small, M., Hill, J., Emelyanova, I. and Clennell, M.B., 2020. Fast automatic detection of geological boundaries from multivariate log data using recurrence. *Computers & Geosciences*, *135*, p.104362.

CORE Skills

A CORE partnership

COREHUB.COM.AU/SKILLS