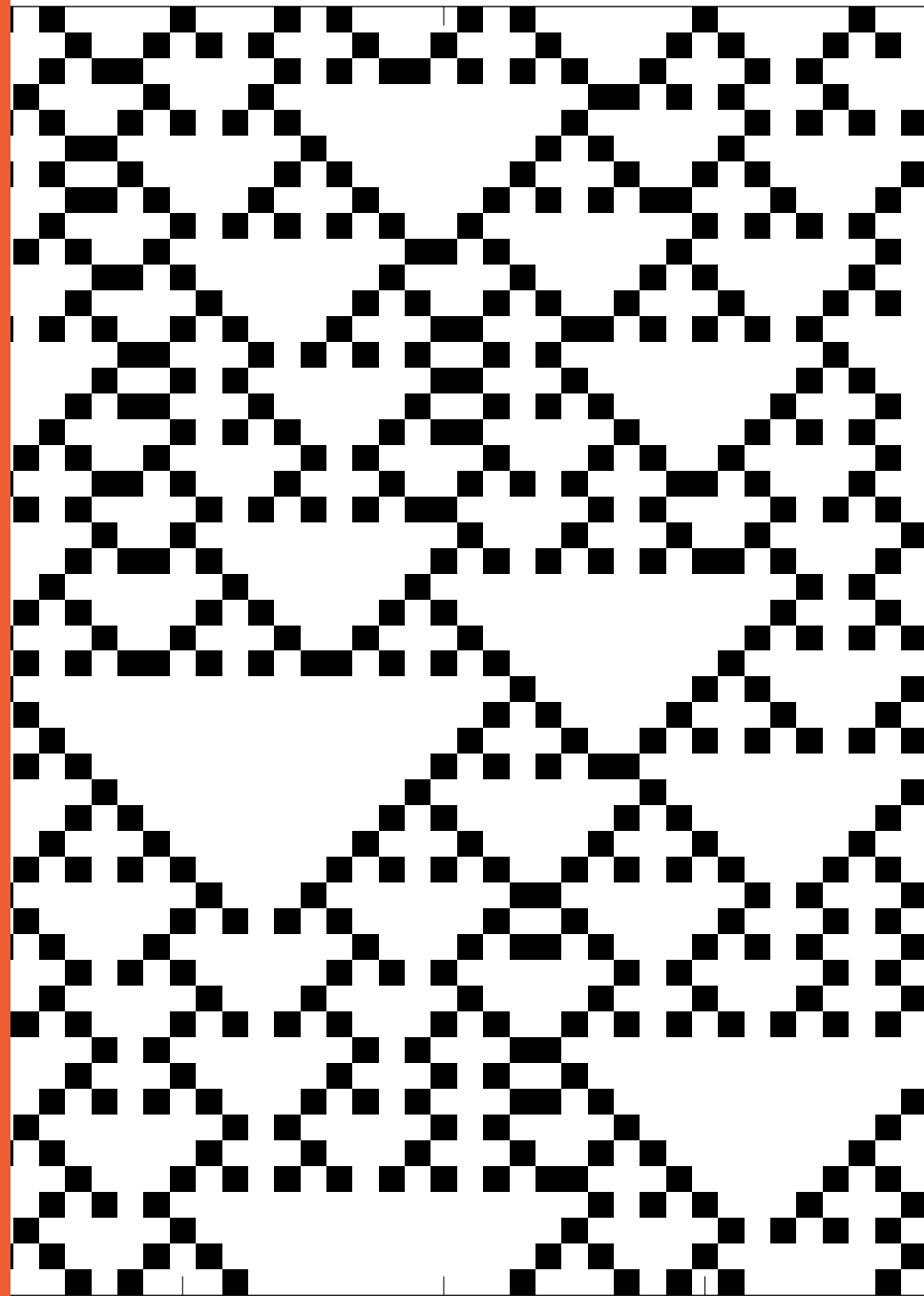


Lecture 6 – Information dynamics – part I

Dr. Joseph Lizier



COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by or on behalf of the **University of Sydney** pursuant to Part VB of the Copyright Act 1968 (the Act).

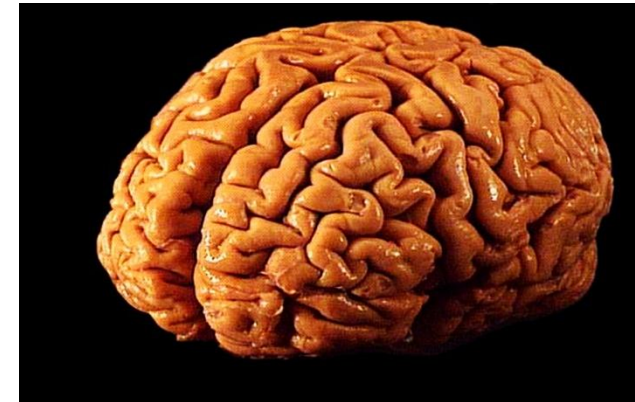
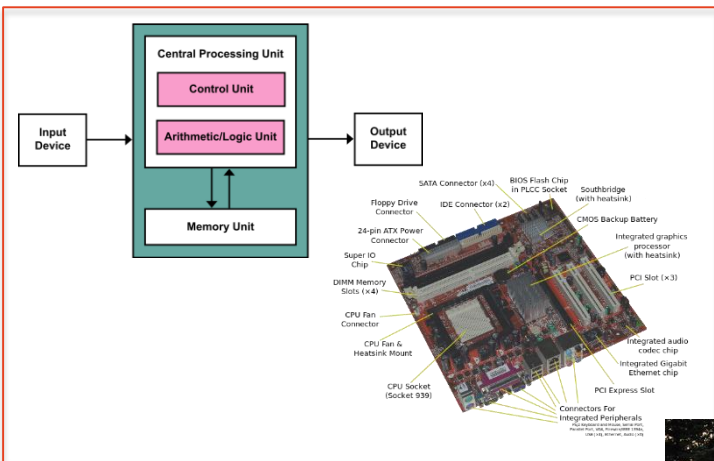
The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

Information dynamics Part I: session outcomes

- Understand philosophy behind information dynamics approach for analysing information processing in complex systems.
- Apply JIDT using AutoAnalyser and extensions of code it produces to analyse information storage in complex systems data sets.
- Primary references:
 - J.T. Lizier, "JIDT: An information-theoretic toolkit for studying the dynamics of complex systems", *Frontiers in Robotics and AI*, 1:11, 2014; appendix A.2 and A.3
 - J.T. Lizier, *"The local information dynamics of distributed computation in complex systems"*, Springer: Berlin/Heidelberg, 2013; chapter 3, 4
 - Bossomaier, Barnett, Harré, Lizier, "An Introduction to Transfer Entropy: Information Flow in Complex Systems", Springer, Cham, 2016; chapter 4 (sections 4.1-4.3); section 5.1

Using Turing machines, is computation easy to spot?



- For each image, consider whether the system is computing.
 - If so: What is it computing? How is it computing that: what are inputs/outputs/information? How are they manipulated?

[Von Neumann architecture](#) by Kapooht, CC BY-SA 3.0; [Motherboard](#) by Moxfyre at en.wikipedia, CC BY-SA 3.0; Fish by [Bruno de Giusti](#), CC BY SA 2.5 IT; Ants by [kodomut](#) @ flickr, CC BY 2.0; Fireflies by [s58y](#) @ flickr, CC BY 2.0; Brain by [aboutmodafinil.com](#) @ flickr, CC BY 2.0

Can we fit biological computation into dominant computer science paradigm of computation?

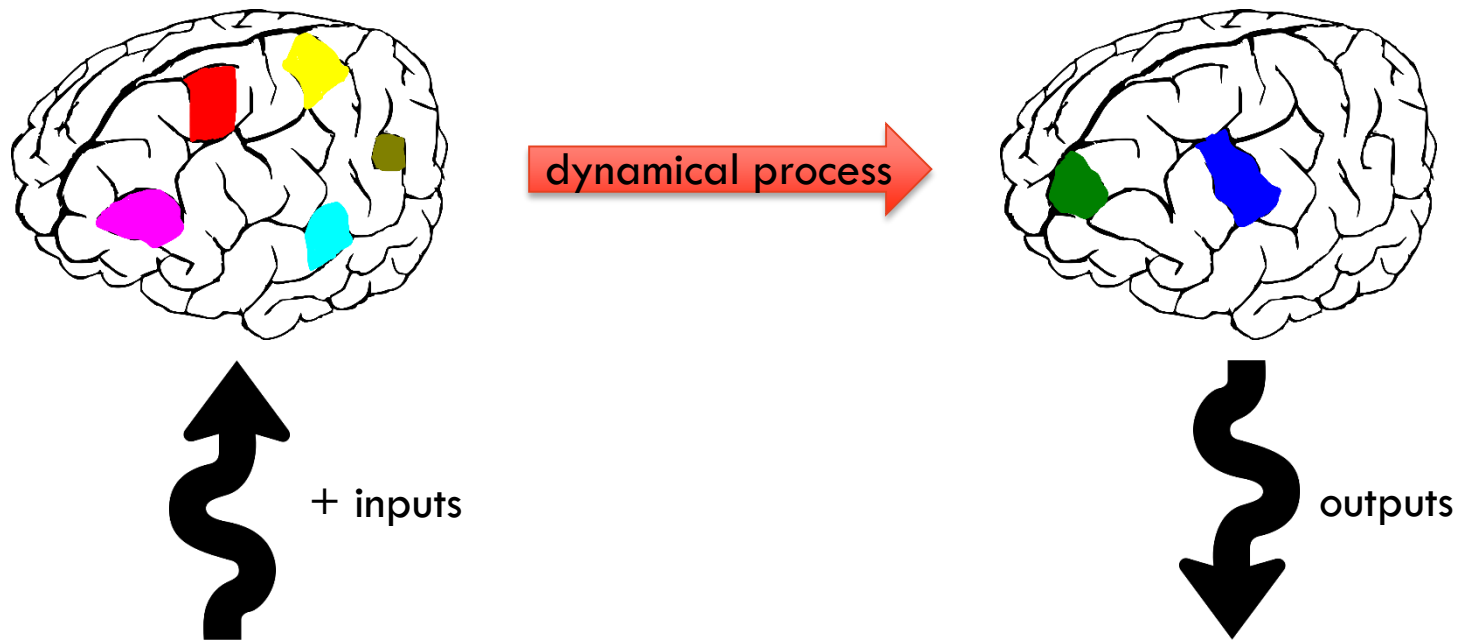
- What do you think?
- What would we look for?
- Mitchell:

	Computer science	Biological computation
What plays the role of information in the system?	Digital static tape	Analog states, patterns distributed in space and time. Gathered via statistical sampling
How is the information communicated and processed?	Deterministic, serial, error-free centralised rules	Decentralised, parallel, local, fine-grained stochastic interactions. Randomness utilised.
How does the information acquire function/purpose/meaning?	(Human) designer	Natural selection

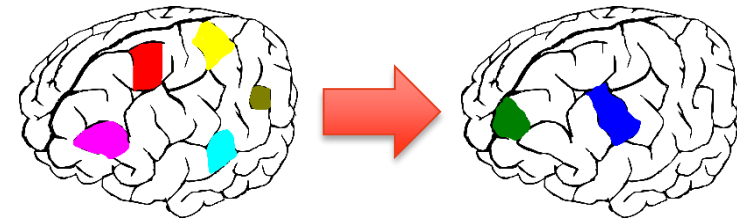
M. Mitchell, “Complexity: A guided tour”, New York: Oxford University Press, 2009 – chapter 12

Biological computation: we need a new perspective

- Mitchell: *“Language of dynamical systems may be more useful than language of computation.”*



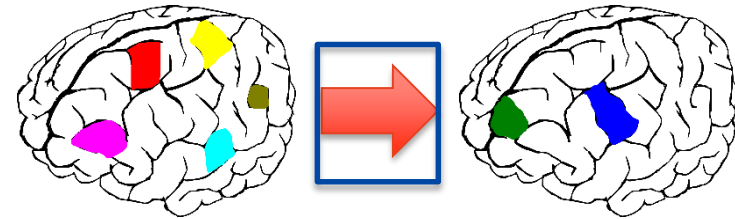
Biological computation: dynamical systems perspective



- What is happening in bio- and bio-inspired information processing?
 - It's **distributed**, unlike a Turing machine
 - It's **ongoing**, unlike a Turing machine
 - Intrinsic computation, or **information processing** doesn't necessarily finish
 - How can we describe it in computational or informational terms?
 - **Information storage, transfer and modification**
 - Easy to identify (elements performing) these operations on information in a traditional PC, not so easy in biological computation

Information dynamics and computation

- We *talk* about computation as:
 - Memory
 - Signalling
 - Processing
- **Distributed computation** is any process that involves these features:
 - Information processing in the brain
 - Time evolution of cellular automata
 - Gene regulatory networks computing cell behaviours
 - Flocks computing their collective heading
 - Ant colonies computing the most efficient routes to food
 - The universe is computing its own future!

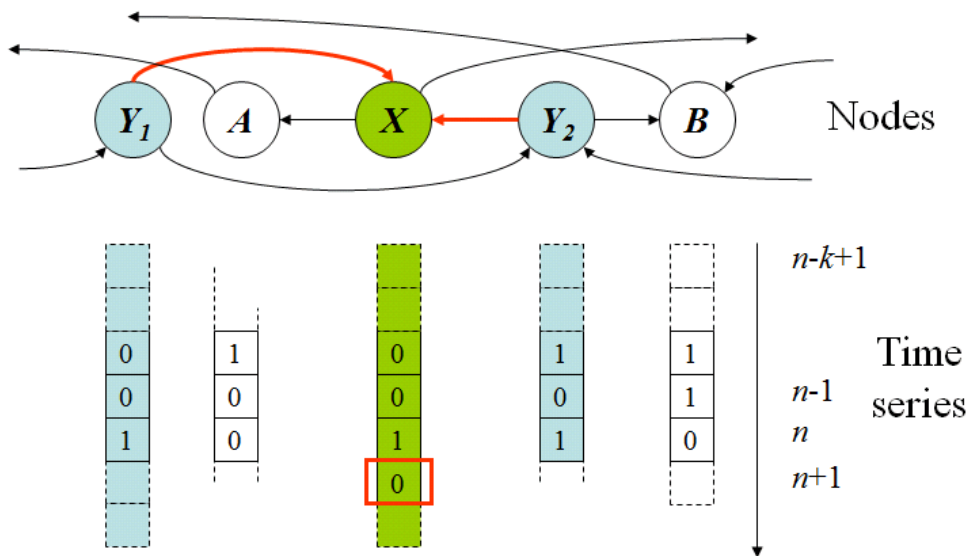


Information dynamics and computation

- We *talk* about computation as:
 - Memory
 - Signalling
 - Processing
- **Idea:** Quantify computation via:
 - Information **storage**
 - Information **transfer**
 - Information **modification**
- **Distributed computation** is any process that involves these features:
 - Information processing in the brain
 - Time evolution of cellular automata
 - Gene regulatory networks computing cell behaviours
 - Flocks computing their collective heading
 - Ant colonies computing the most efficient routes to food
 - The universe is computing its own future!
- **General idea:** by quantifying intrinsic computation in the language it is normally described in, we can understand how nature computes and why it is complex.

Information dynamics

- Key question: how is the next state of a variable in a complex system **computed**?



Complex system as a **multivariate time-series** of states

Q: Where does the information in x_{n+1} come from, and how can we measure it?
(Where might we look?)

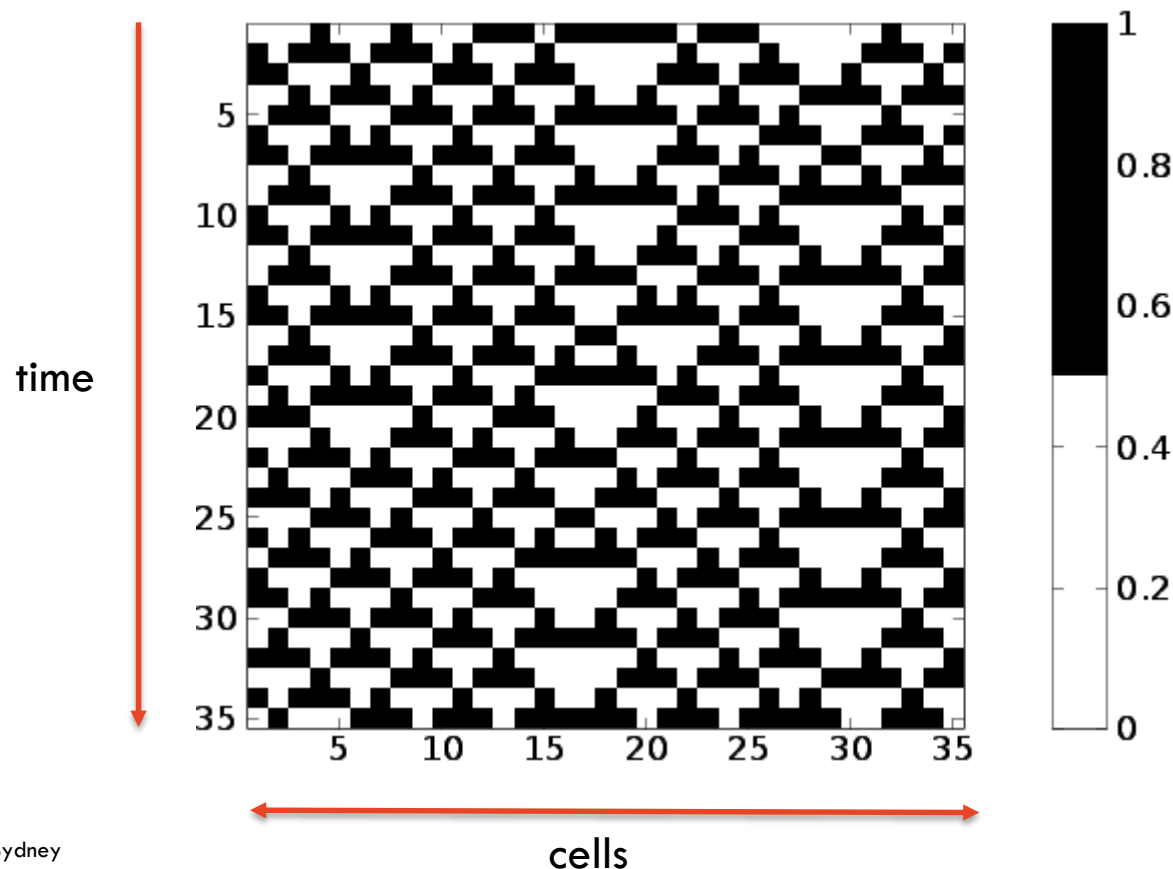
Q: Can we **model** the information processing in X in terms of:

- how much information was stored?
- how much was transferred?

Q: Can we partition them, do they overlap? etc.

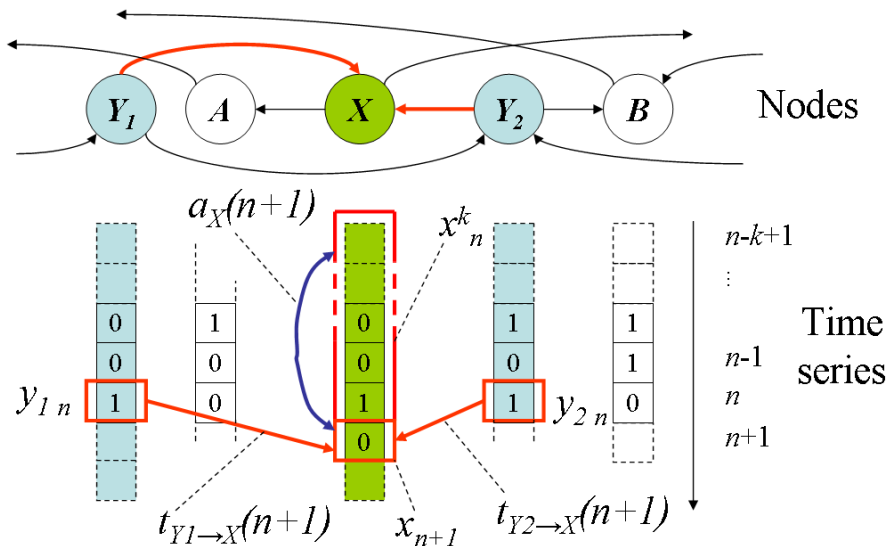
What kinds of multivariate time series could we analyse?

- How did we model behaviour in Scissors-Paper-Rock?
- How can we characterise the updates in cellular automata in terms of operations on information?



Information dynamics

- Studies computation of the next state of a target variable in terms of information **storage**, **transfer** and **modification**:



The measures examine:

- **State** updates of a target variable;
- **Dynamics** of the measures in space and time.

Notation

- We consider time-series **processes** X :
 - Which consist of random **variables** $\{\dots X_{n-1}, X_n, X_{n+1}, \dots\}$;
 - With process **realisations** $\{\dots x_{n-1}, x_n, x_{n+1}, \dots\}$;
 - For countable time indices n .
- Denote consecutive **block** vector: $X_n^{(k)} = \{X_{n-k+1}, \dots, X_{n-1}, X_n\}$
 - which has realisations $x_n^{(k)} = \{x_{n-k+1}, \dots, x_{n-1}, x_n\}$
- Formally, we ask: “where does the information in a random variable X_{n+1} come from, in terms of other variables Y_m, Z_m , etc. for $m \leq n$?”

Entropy rate

- Historically, **entropy rate** was first consideration here:

- Measures limiting rate at which block entropies scale with block length:

$$H'_{\mu X} = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n)$$

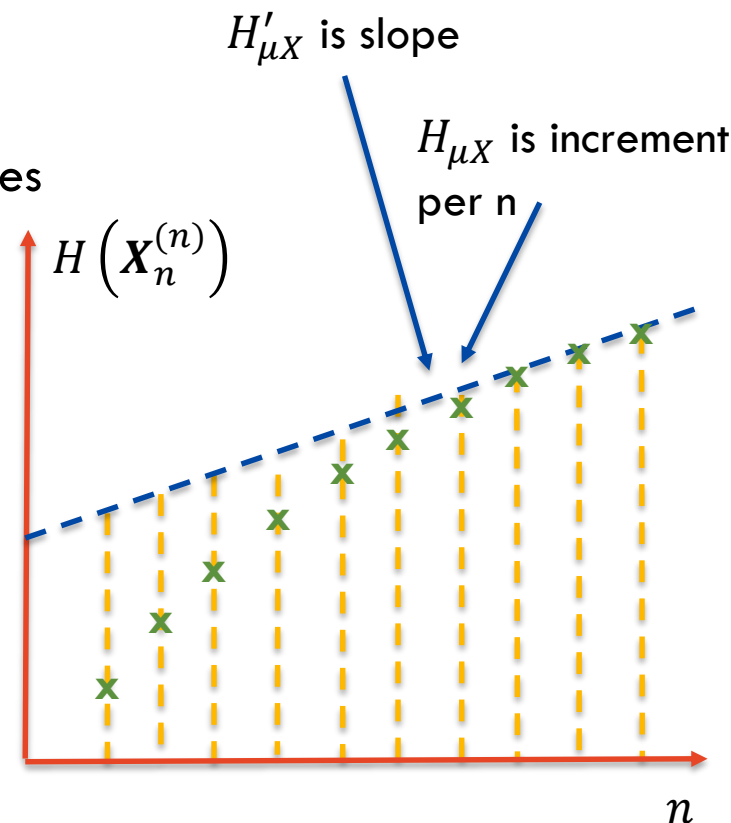
$$H'_{\mu X} = \lim_{n \rightarrow \infty} \frac{1}{n} H(\mathbf{X}_n^{(n)})$$

- Measures uncertainty of next R.V. given past of process:

$$H_{\mu X} = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_n | X_1, X_2, \dots, X_{n-1})$$

$$H_{\mu X} = \lim_{n \rightarrow \infty} H(X_n | \mathbf{X}_{n-1}^{(n-1)})$$

- $H'_{\mu X} = H_{\mu X}$ for stationary processes



- Implication is that we're using past of the process as first informative source, and asking how much uncertainty remains.

T. M. Cover and J. A. Thomas. "Elements of Information Theory". Wiley-Interscience, New York, 1991. Section 4.2. *Note: primes are reversed in our notation!*

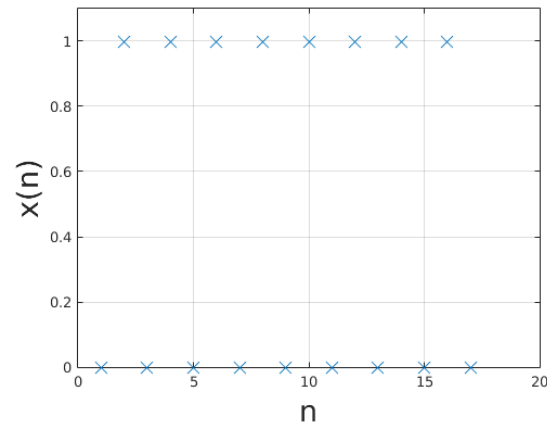
J. P. Crutchfield and D. P. Feldman, "Regularities unseen, randomness observed: Levels of entropy convergence", Chaos 13, 25 (2003).

Information storage

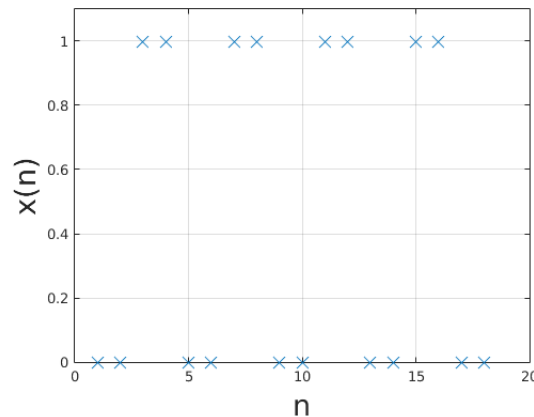
- How much information from the past of the variable helps us predict its next state?
- Or, in modelling the dynamics of the variable, how much information storage would we include in that model by accounting for the past influence of that variable?

Information storage – using our intuition

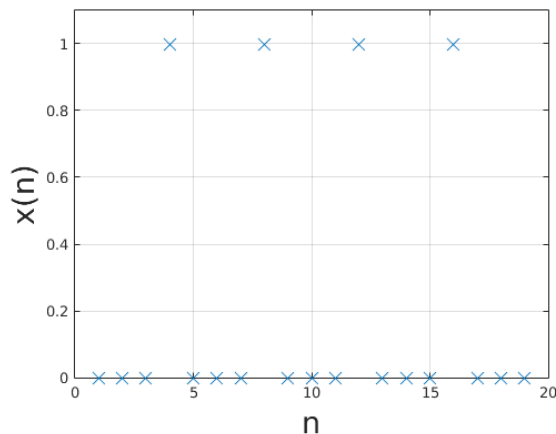
- In each example:
 - Try to predict the next value of the variable.
 - What assumptions did you make?
 - Where specifically did you take the information from to make that prediction?
 - How much information did you get from the past?



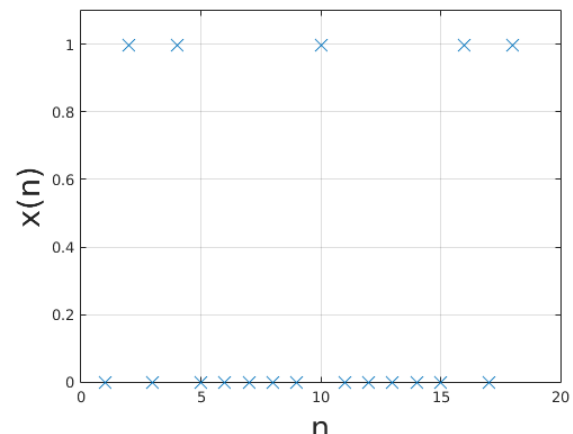
(a)



(b)



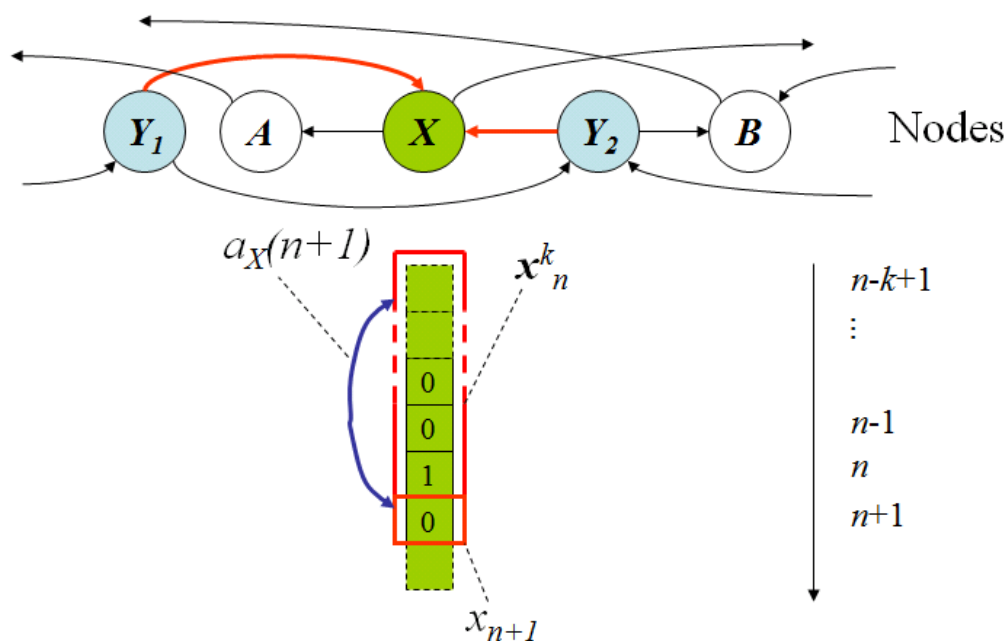
(c)



(d)

Active information storage

- How much information about the next observation X_n of process X can be found in its past state $X_n^{(k)} = \{X_{n-k+1}, \dots, X_{n-1}, X_n\}$?



Active information storage

$$A_X = \lim_{k \rightarrow \infty} I(X_n^{(k)}; X_{n+1})$$

$$A_X(k) = I(X_n^{(k)}; X_{n+1})$$

$$A_X = H(X_{n+1}) - H_{\mu X}$$

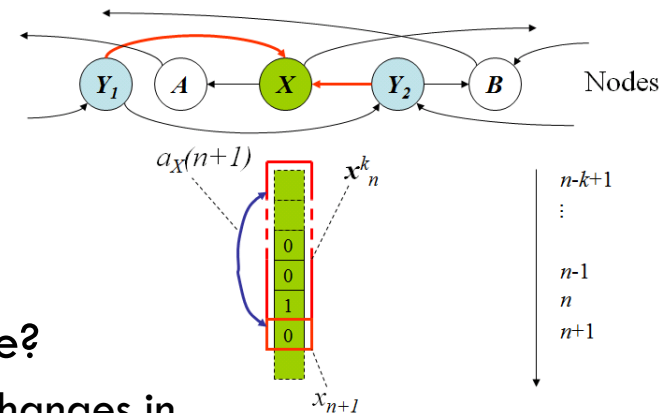
$$A_X(k) = \left\langle \log_2 \frac{p(x_{n+1} | x_n^{(k)})}{x_{n+1}} \right\rangle$$

$$a_X(k) = \log_2 \frac{p(x_{n+1} | x_n^{(k)})}{x_{n+1}}$$

- AIS: Average information from past state that is in use in predicting the next value
- Local AIS: information from a specific past state in use in predicting specific next value

Active information storage: interpretations

- Captures total memory and nonlinear effects
 - Autocorrelation just linear component from each past value separately.
- What types of information storage does A_X capture?
 - Active storage in dynamics (as opposed to passive changes in underlying structure);
 - Internal (causally) stored information;
 - Distributed information storage via feedback and feedforward loops, i.e. recurrent connections (network effects);
 - Input-driven storage: Patterns in input dynamics driving a variable
- All of these are intrinsically *modelled* as information storage to an observer when we account for the information here.



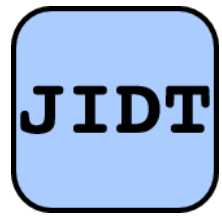
M. Wibral, J. T. Lizier, S. Vögler, V. Priesemann, and R. Galuske, “Local active information storage as a tool to understand distributed neural information processing”, *Frontiers in Neuroinformatics* 8, 1+ (2014).

Zipser, D., Kehoe, B., Littlewort, G., and Fuster, J. (1993), “A spiking network model of short-term active memory”, *Journal of Neuroscience*, 13, 8, 3406–3420

Lizier, J. T., Atay, F. M., and Jost, J. (2012), “Information storage, loop motifs, and clustered structure in complex networks”, *Phys Rev E*, 86, 2, 026110

Obst, O., Boedecker, J., Schmidt, B., and Asada, M. (2013), “On active information storage in input-driven systems”, *arXiv:1303.5526*

Active information storage in JIDT



- Start AIS AutoAnalyser
- Notice the important `k_HISTORY` parameter
- Has all types of underlying MI estimators available, same parameters as each and features (e.g. statistical significance, local values)

JIDT Active Information Storage Auto-Analyser

Calculation parameters

Calculator Type: **Discrete**

Data file: `master/jidt/demos/data/2coupledBinaryColsUseK2.txt`
Select
Valid data file with 1000 rows and 2 columns

☒ All variables?

Variable column: **0**

☒ Add stat. signif. ? ☐ analytically?

Property name	Property value
base	2
k_HISTORY	1

☒ Compute result? **Generate code and Compute**

Status
See console for all pairs calculation results

Generated code

Java **Python** **Matlab**

```
% Add JIDT jar library to the path
jidtPath = '././';
javaaddpath([jidtPath, '/infodynamics.jar']);
% Add utilities to the path
addpath([jidtPath, '/demos/octave']);

% 0. Load/prepare the data:
data = load('/home/joseph/temp/jidt-master/jidt/demos/data/2coupledBinaryColsUseK2.txt');
% 1. Construct the calculator:
calc = javaObject('infodynamics.measures.discrete.ActiveInformationCalculatorDiscrete', data);
% 2. No other properties to set for discrete calculators.

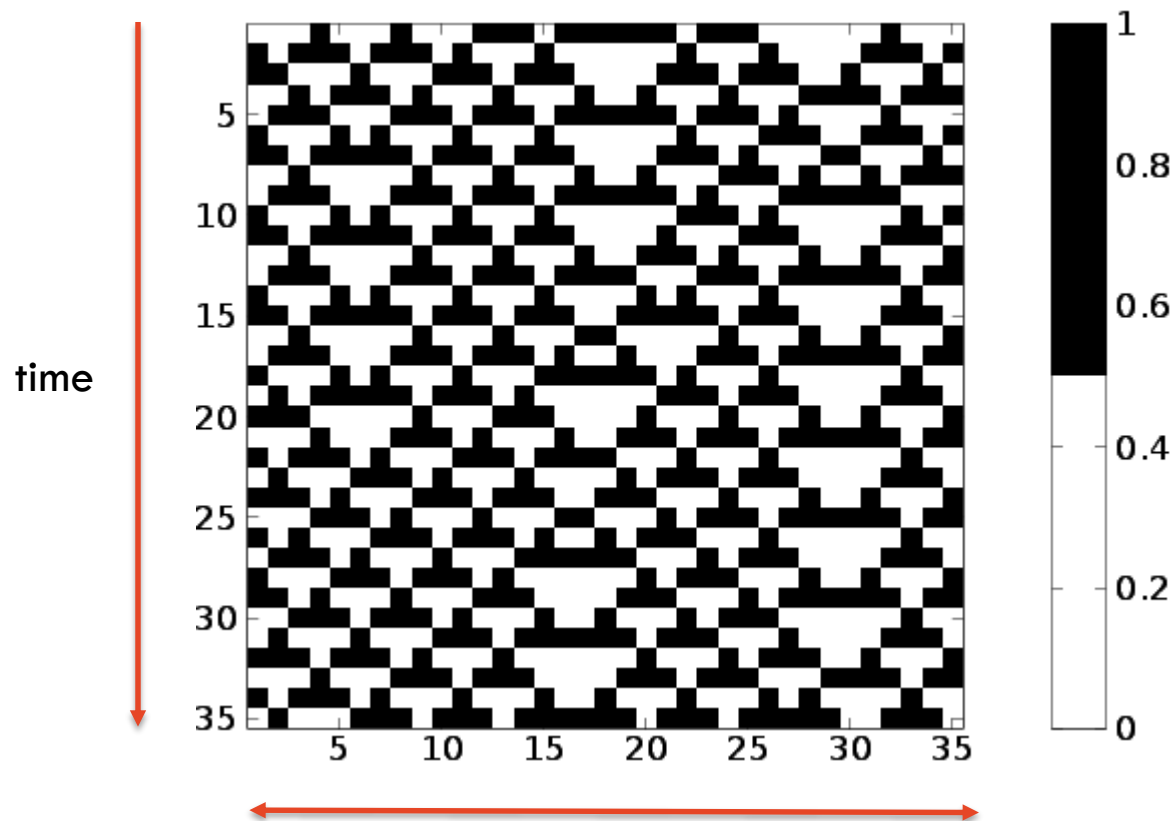
% Compute for all variables:
for v = 1:2
    % For each variable:
    % Column indices start from 1 in Matlab:
    variable = octaveToJavaIntArray(data(:, v));

    % 3. Initialise the calculator for (re-)use:
    calc.initialise();
    % 4. Supply the sample data:
    calc.addObservations(variable);
    % 5. Compute the estimate:
    result = calc.computeAverageLocalOfObservations();
    % 6. Compute the (statistical significance via) null distribution empirical:
    measDist = calc.computeSignificance(100);

    fprintf('AIS_Discrete(col_%d) = %.4f bits (null: %.4f +/- %.4f std dev.; p(s) = %.4f\n', v, result, measDist.getMeanOfDistribution(), measDist.getStdDev(), measDist.getProbabilityOfSignificance());
end
```

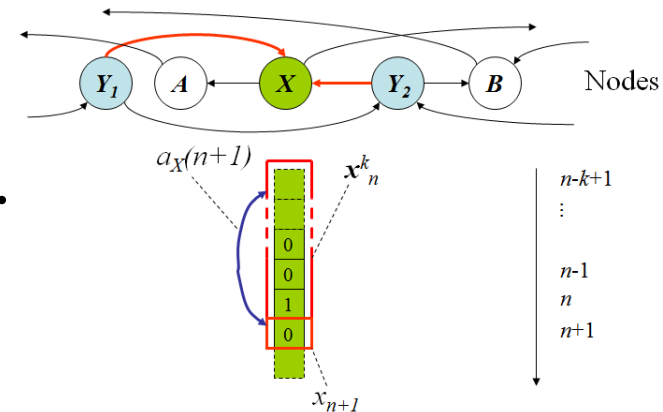
ALS: key question – how to set history length k ?

- For example in Elementary Cellular Automata rule 54?



ALS: key question – how to set history length k?

- $X_n^{(k)} = \{X_{n-k+1}, \dots, X_{n-1}, X_n\}$ is a **Takens' embedding** of the past state of X .
- Want to set embedding for optimal prediction of next value X_{n+1}
 - $p(x_{n+1} | x_n^{(k)}, x_{n-k}) = p(x_{n+1} | x_n^{(k)})$
- But we have competing concerns:
 - Want k as large as possible to capture all potential memory.
 - But increasing k increases our exposure to undersampling.
- There will be a “sweet spot” in between, either:
 - Where further values from past don't actually contribute, or
 - Where there is not enough data to validate their contribution.



ALS: key question – how to set embedding parameters?

- Can use embedding delay τ also: $X_n^{(k,\tau)} = \{X_{n-(k-1)\tau}, \dots, X_{n-\tau}, X_n\}$
- Option 1: Ragwitz criteria to minimize prediction error
 - Find K nearest neighbours for each $x_n^{(k)}$
 - Find mean of their corresponding x_{n+1}
 - Compute difference to actual x_{n+1}
 - Take mean over all points and minimize w.r.t history length k and embedding delay τ

ALS: key question – how to set embedding parameters?

- Option 2: Maximize bias-corrected ALS

$$A'_X = A_X - \langle A_X^S \rangle$$

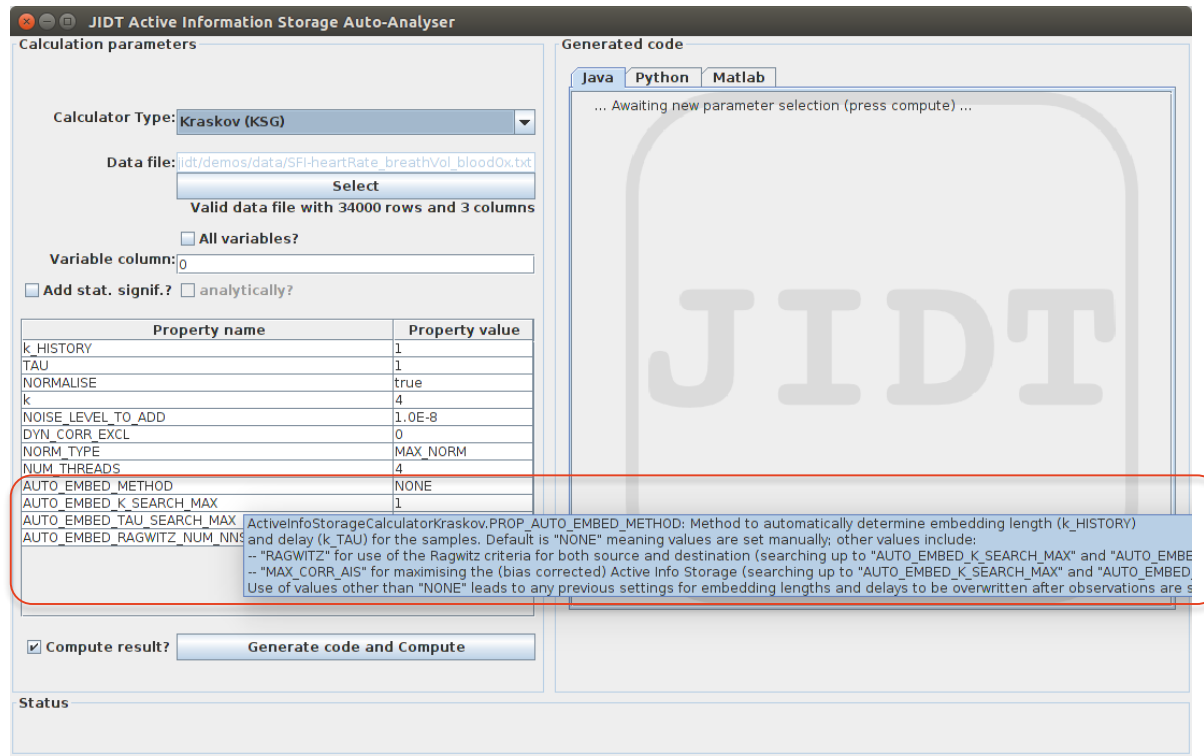
- (where A_X^S are surrogates, created by destroying the past-next relationship)
 - w.r.t history length k and embedding delay τ
- This means we use more points from history so long as they contribute more information about the next value than the increase in bias due to the higher dimensionality.
- For KSG estimator, bias correction is already built in!

ALS: key question – how to set embedding parameters?

- Option 3: **Non-uniform embedding**
 - Incrementally select points from past which make a statistically significant contribution beyond the points already selected.
 - Not yet implemented in JIDT but is in our higher-level IDTxl toolbox

AIS: setting embedding parameters in JIDT

- Select KSG estimator
- Hover on AUTO_EMBED_METHOD property to see options:
 - NONE: set `k_HISTORY` and `TAU` manually.
 - RAGWITZ: optimal parameters to minimise prediction error scanned up to `AUTO_EMBED_K_SEARCH_MAX` and `AUTO_EMBED_TAU_SEARCH_MAX`
 - MAX_CORR_AIS: optimal parameters to max. bias-corrected AIS scanned up to `AUTO_EMBED_K_SEARCH_MAX` and `AUTO_EMBED_TAU_SEARCH_MAX`



JIDT Active Information Storage Auto-Analyser

Calculation parameters

Calculator Type: **Kraskov (KSG)**

Data file: `jdt/demos/data/SFI-heartRate_breathVol_bloodOx.txt`
 Select
 Valid data file with 34000 rows and 3 columns

☐ All variables?
 Variable column: `0`

☒ Add stat. signif.? ☐ analytically?

Property name	Property value
<code>k_HISTORY</code>	1
<code>TAU</code>	1
<code>NORMALISE</code>	true
<code>k</code>	4
<code>NOISE_LEVEL_TO_ADD</code>	1.0E-8
<code>DYN_CORR_EXCL</code>	0
<code>NORM_TYPE</code>	MAX_NORM
<code>NUM_THREADS</code>	4
<code>AUTO_EMBED_METHOD</code>	NONE
<code>AUTO_EMBED_K_SEARCH_MAX</code>	1
<code>AUTO_EMBED_TAU_SEARCH_MAX</code>	
<code>AUTO_EMBED_RAGWITZ_NUM_NNS</code>	

Generated code

Java Python Matlab

... Awaiting new parameter selection (press compute) ...

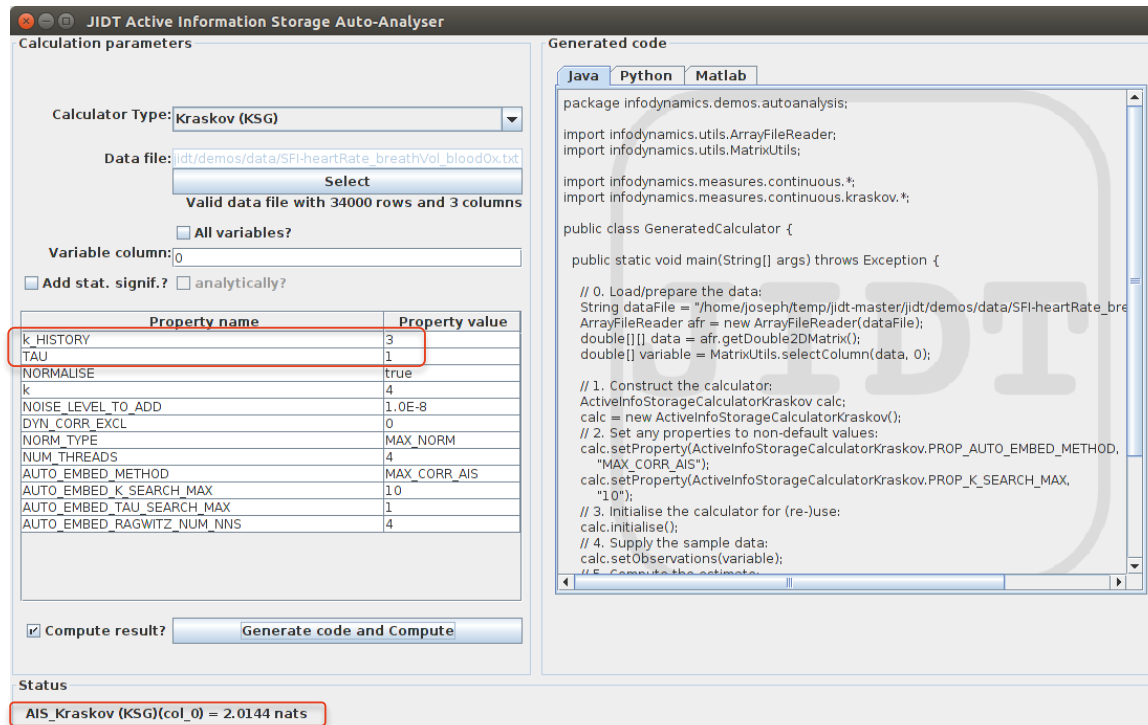
Tooltip for `AUTO_EMBED_METHOD`:
 ActiveInfoStorageCalculatorKraskov.PROP_AUTO_EMBED_METHOD: Method to automatically determine embedding length (`k_HISTORY`) and delay (`k_TAU`) for the samples. Default is "NONE" meaning values are set manually; other values include:
 -- "RAGWITZ" for use of the Ragwitz criteria for both source and destination (searching up to "`AUTO_EMBED_K_SEARCH_MAX`" and "`AUTO_EMBED_TAU_SEARCH_MAX`")
 -- "MAX_CORR_AIS" for maximising the (bias corrected) Active Info Storage (searching up to "`AUTO_EMBED_K_SEARCH_MAX`" and "`AUTO_EMBED_TAU_SEARCH_MAX`")
 Use of values other than "NONE" leads to any previous settings for embedding lengths and delays to be overwritten after observations are scanned.

☒ Compute result? **Generate code and Compute**

Status

AIS: setting embedding parameters in JIDT

- Select the SFI-heartRate_breathVol_bloodOx.txt data set
- Set AUTO_EMBED_METHOD to MAX_CORR_AIS and AUTO_EMBED_K_SEARCH_MAX to 10 and AUTO_EMBED_TAU_SEARCH_MAX to 1.
- Click Compute
- The result is returned with optimal parameters shown in k_HISTORY and TAU. You can retrieve them in code via a getProperty() call.



The screenshot shows the JIDT Active Information Storage Auto-Analyser interface. The 'Calculation parameters' section on the left includes a dropdown for 'Calculator Type' set to 'Kraskov (KSG)', a text field for 'Data file' pointing to 'jdt/demos/data/SFI-heartRate_breathVol_bloodOx.txt', and a 'Select' button. Below this, there are checkboxes for 'All variables?', 'Add stat. signif.?', and 'analytically?'. A table lists various 'Property name' and 'Property value' pairs, with 'k_HISTORY' and 'TAU' highlighted by a red box. The 'Generated code' section on the right shows a Java code snippet for the 'GeneratedCalculator' class, which includes comments and code for loading data, constructing the calculator, and setting properties like 'MAX_CORR_AIS', 'MAX_K_SEARCH_MAX', and 'MAX_TAU_SEARCH_MAX'. The 'Status' bar at the bottom indicates 'AIS_Kraskov (KSG)(col_0) = 2.0144 nats'.

Property name	Property value
k_HISTORY	3
TAU	1
NORMALISE	true
k	4
NOISE_LEVEL_TO_ADD	1.0E-8
DYN_CORR_EXCL	0
NORM_TYPE	MAX_NORM
NUM_THREADS	4
AUTO_EMBED_METHOD	MAX_CORR_AIS
AUTO_EMBED_K_SEARCH_MAX	10
AUTO_EMBED_TAU_SEARCH_MAX	1
AUTO_EMBED_RAGWITZ_NUM_NNS	4

```

package infodynamics.demos.autoanalysis;

import infodynamics.utils.ArrayFileReader;
import infodynamics.utils.MatrixUtils;

import infodynamics.measures.continuous.*;
import infodynamics.measures.continuous.kraskov.*;

public class GeneratedCalculator {

    public static void main(String[] args) throws Exception {

        // 0. Load/prepare the data:
        String dataFile = "/home/joseph/temp/jidt-master/jidt/demos/data/SFI-heartRate_breathVol_bloodOx.txt";
        ArrayFileReader afr = new ArrayFileReader(dataFile);
        double[][] data = afr.getDouble2DMatrix();
        double[] variable = MatrixUtils.selectColumn(data, 0);

        // 1. Construct the calculator:
        ActiveInfoStorageCalculatorKraskov calc;
        calc = new ActiveInfoStorageCalculatorKraskov();

        // 2. Set any properties to non-default values:
        calc.setProperty(ActiveInfoStorageCalculatorKraskov.PROP_AUTO_EMBED_METHOD,
            "MAX_CORR_AIS");
        calc.setProperty(ActiveInfoStorageCalculatorKraskov.PROP_K_SEARCH_MAX,
            "10");

        // 3. Initialise the calculator for (re-)use:
        calc.initialise();

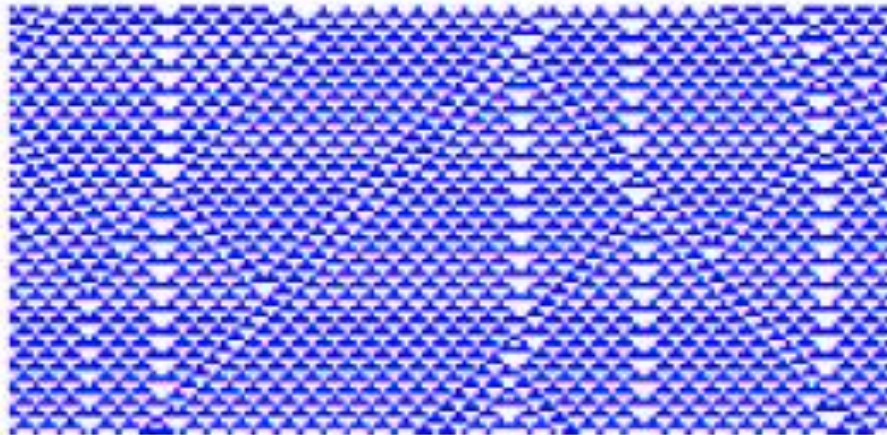
        // 4. Supply the sample data:
        calc.setObservations(variable);

        // 5. Compute the estimate:
    }
}
    
```

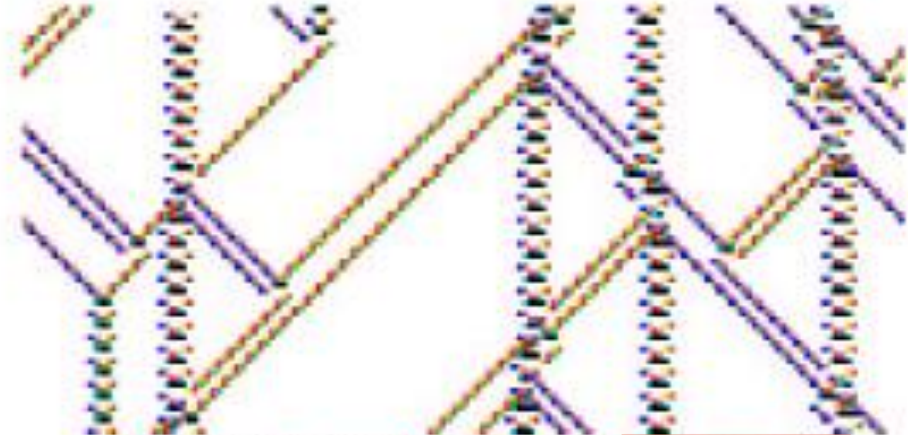
Compute result? ☒ Generate code and Compute

Status
AIS_Kraskov (KSG)(col_0) = 2.0144 nats

Example: Computational role of emergent structure in CAs



cells by value



cells by look-up and **filtered**

– Emergent structure:

- Domains, blinkers
- Particles
 - Gliders, Domain walls
- Collisions

– Conjectured to represent

- Information storage
- Information transfer
 - “
- Information modification

A. Wuensche, “Classifying cellular automata automatically: Finding gliders, filtering, and relating space-time patterns, attractor basins, and the Z parameter,” *Complexity*, vol. 4, no. 3, pp. 47–66, 1999. (plus image credit)

C. G. Langton, “Computation at the edge of chaos: phase transitions and emergent computation,” *Physica D*, vol. 42, no. 1-3, pp. 12–37, 1990.
The University of Sydney



Example: Computational role of emergent structure in CAs

- Go to tutorial sheet to try out our AIS calculator on CA data:
 - We'll compute appropriate embedding length
 - We'll compute local AIS values and see whether domains and blinkers do indeed have strong information storage values.

Predictive information

- How much information about the future $X_{n+1}^{(k+)} = \{X_{n+1}, X_{n+2}, \dots, X_{n+k}\}$ of process X can be found in its past state $X_n^{(k)} = \{X_{n-k+1}, \dots, X_{n-1}, X_n\}$?

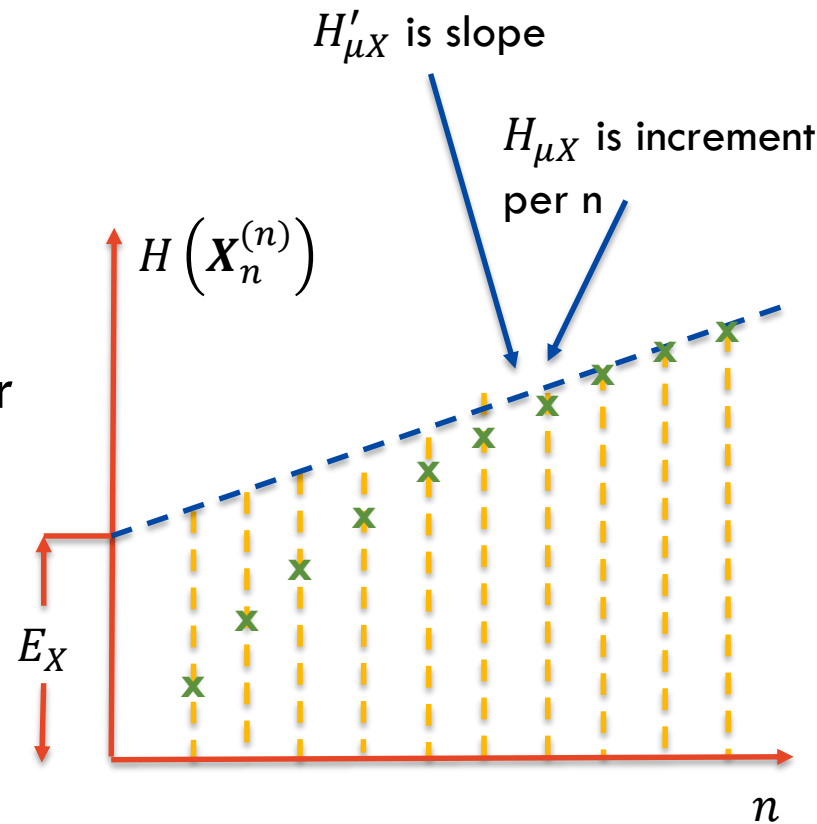
$$E_X = \lim_{k \rightarrow \infty} I(X_n^{(k)}; X_{n+1}^{(k+)})$$

$$E_X(k) = I(X_n^{(k)}; X_{n+1}^{(k+)})$$

- Captures *all* of the information stored in the past that is used *at some point* in the future.
 - Contrast to AIS which measures the part of the stored information in use in computing the next value.
 - We're more interested in AIS because it focusses on the computation of the next value and is complementary to information transfer.

Predictive information and excess entropy

- Excess entropy quantifies total structure or memory as slowness of the approach of the conditional entropy rate estimates to their limiting value:
- Is equal to predictive information for stationary processes.



Information dynamics Part I: summary

- We've looked at the philosophy behind the information dynamics approach for analysing information processing in complex systems.
- In particular, we've focussed on how information storage is characterised.
 - And used JIDT AutoAnalyser and extensions of code to analyse information storage in complex systems data sets.
- *Next lecture:* Information processing in complex systems Part II – information transfer.

Questions



THE UNIVERSITY OF
SYDNEY