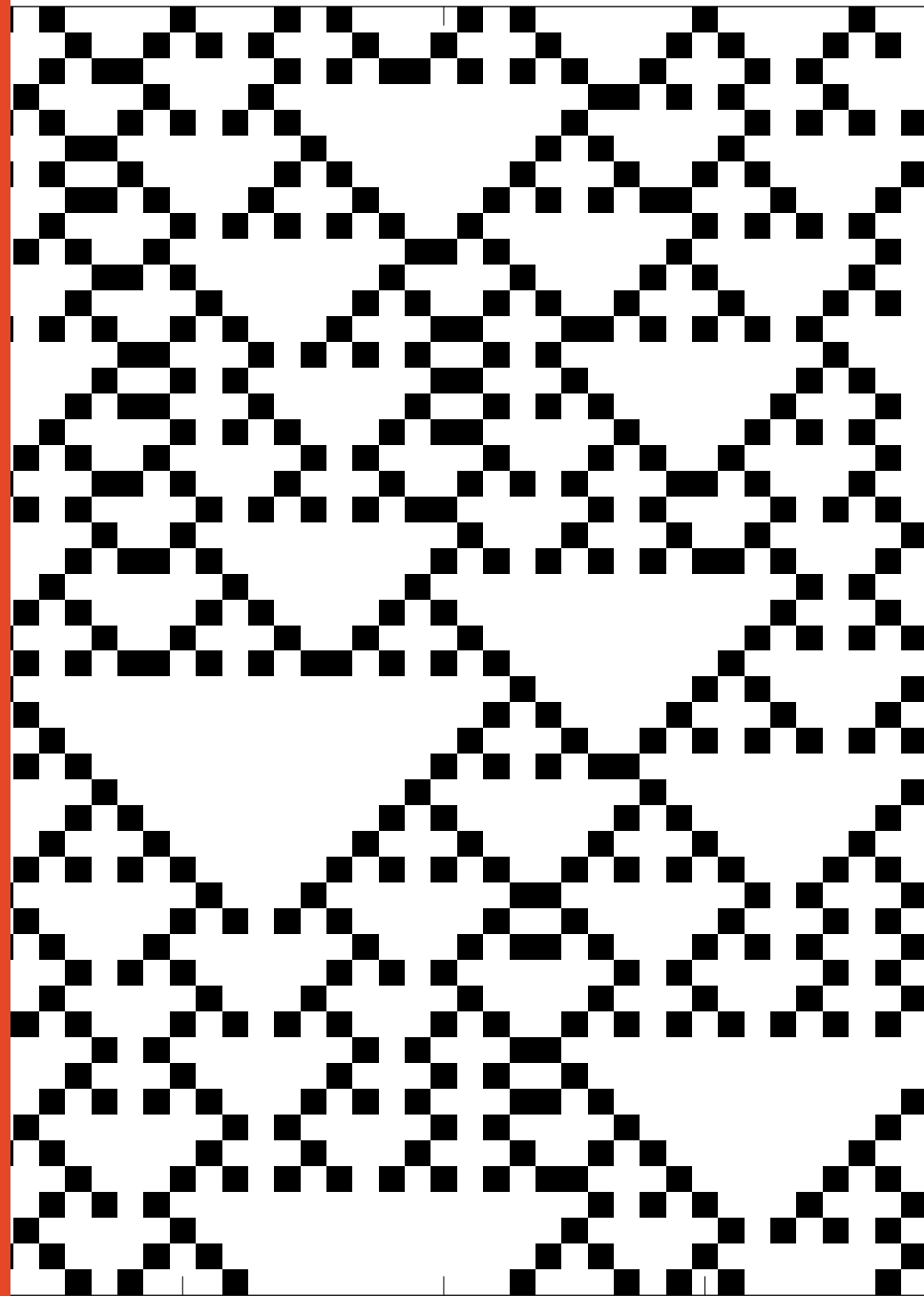


Information dynamics – Part II – Information storage

Dr. Joseph Lizier



THE UNIVERSITY OF
SYDNEY



Information dynamics Part II: session outcomes

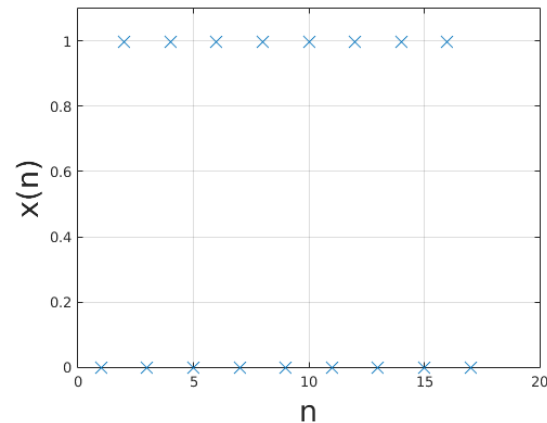
- Understand measures for **information storage**
- Apply JIDT using AutoAnalyser and extensions of code it produces to analyse information storage in complex systems data sets.
- Primary references (for Info Dynamics II and III sessions):
 - J.T. Lizier, "JIDT: An information-theoretic toolkit for studying 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

Information storage

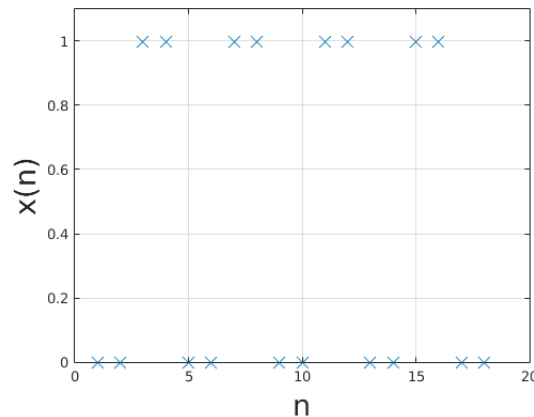
- How much information from the past of the variable helps us predict its next state?
- Or, in **modelling** the dynamics of the variable, how much information storage would we include in that model by accounting for the past influence of that variable (as a first step)?

Information storage – using our intuition

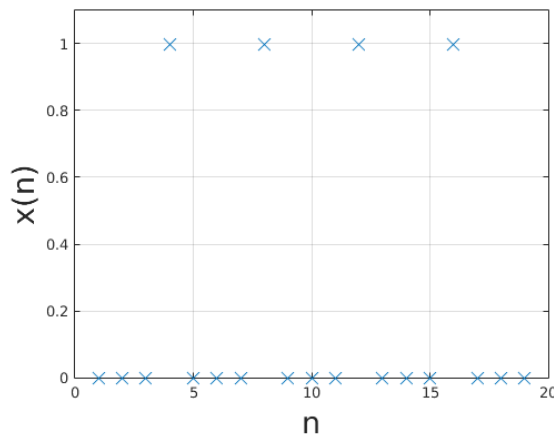
- In each example:
 - Try to predict the next value of the variable (beyond what is shown).
 - What assumptions did you make?
 - Where specifically did you take the information from to make that prediction?
 - How much information did you get from the past?



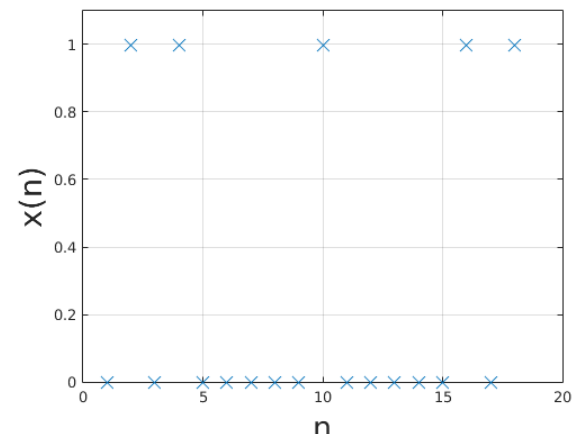
(a)



(b)



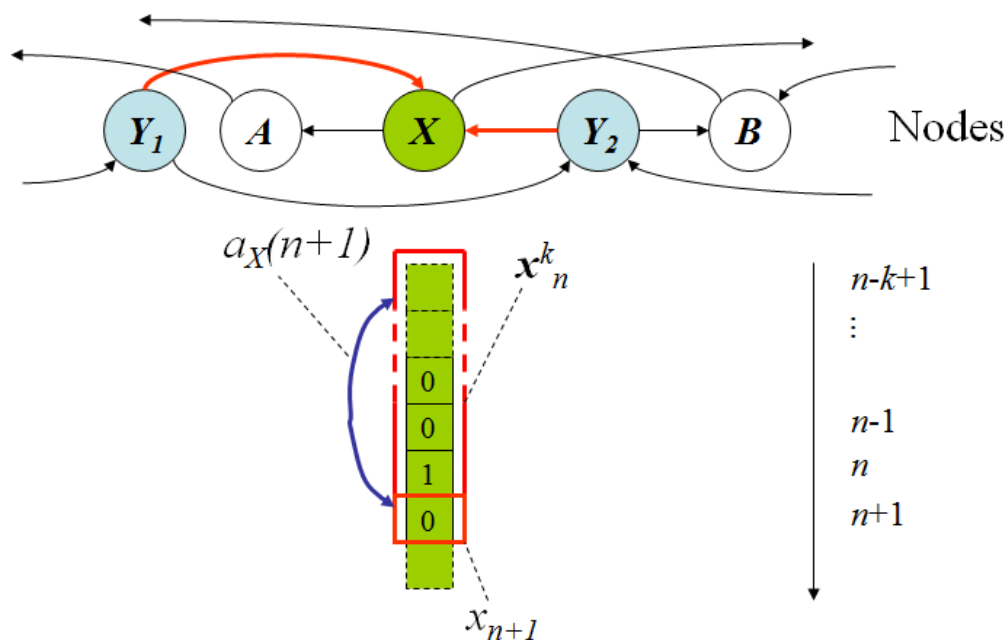
(c)



(d)

Active information storage

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



Active information storage

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

$$A_X(k) = I(\mathbf{X}_n^{(k)}; X_{n+1})$$

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

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

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

- 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

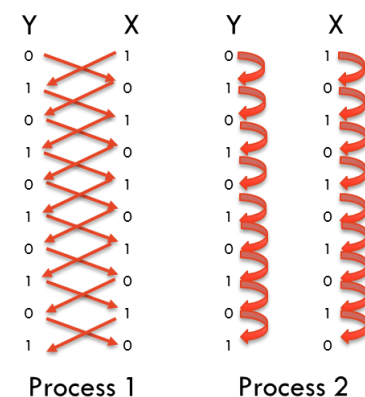
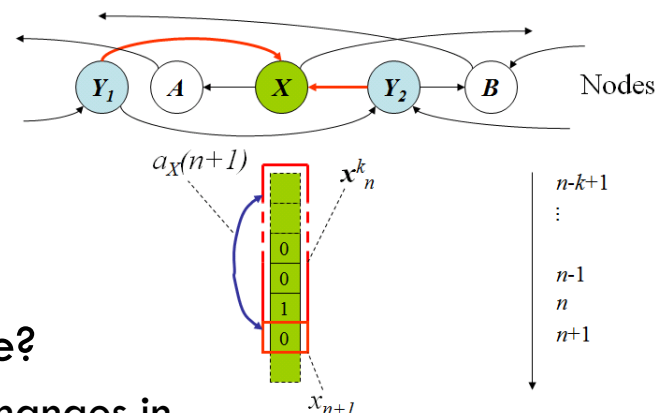
J.T. Lizier, M. Prokopenko, A.Y. Zomaya, "Detecting Non-trivial Computation in Complex Dynamics", Proc. ECAL, pp. 895-904 (2007).

J.T. Lizier, M. Prokopenko, & A.Y. Zomaya, "Local measures of information storage in complex distributed computation", Information Sciences 208, 39 (2012)

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

Active information storage: interpretations

- Captures total memory and nonlinear effects
 - Autocorrelation is 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).

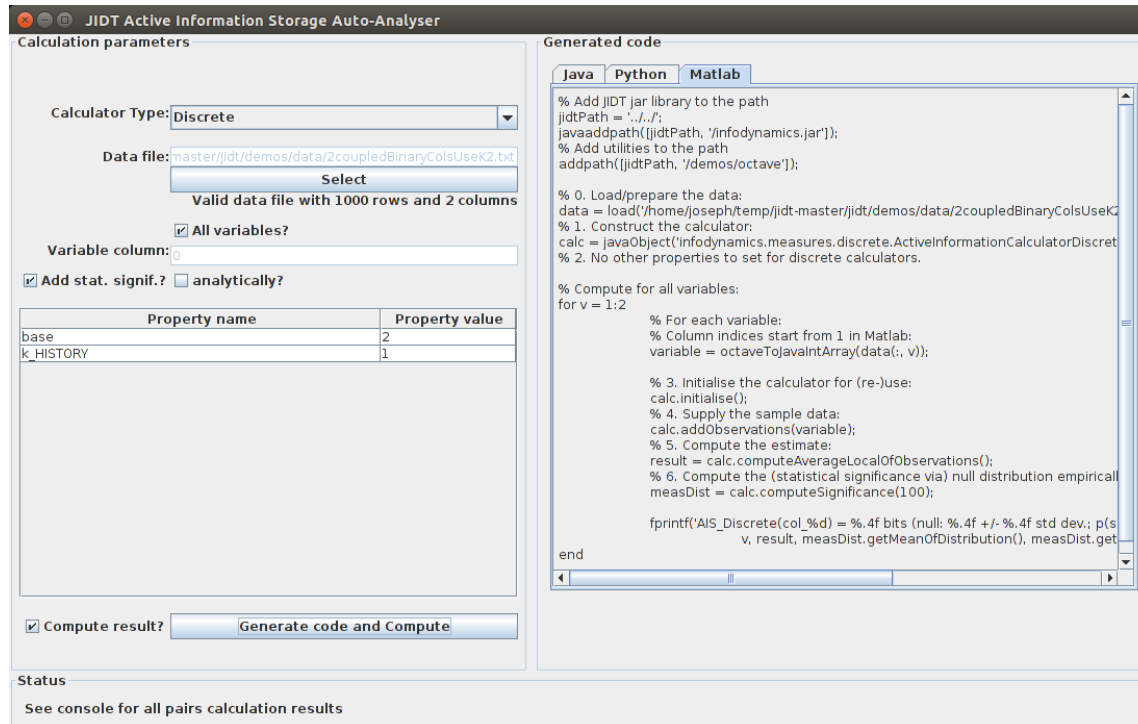
Zipser, D., Kehoe, B., Littlewort, G., and Fuster, J. (1993), "A spiking network model of short-term active memory", *Journal of Neuroscience*, 13, 8, 3406–3420

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



The screenshot shows the JIDT Active Information Storage Auto-Analyser window. The left pane contains the 'Calculation parameters' section, and the right pane shows the 'Generated code' for the selected calculator.

Calculation parameters:

- Calculator Type: Discrete
- Data file: master/jidt/demos/data/2coupledBinaryColsUseK2.txt (with a 'Select' button)
- 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?

Generated code (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');
% 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);

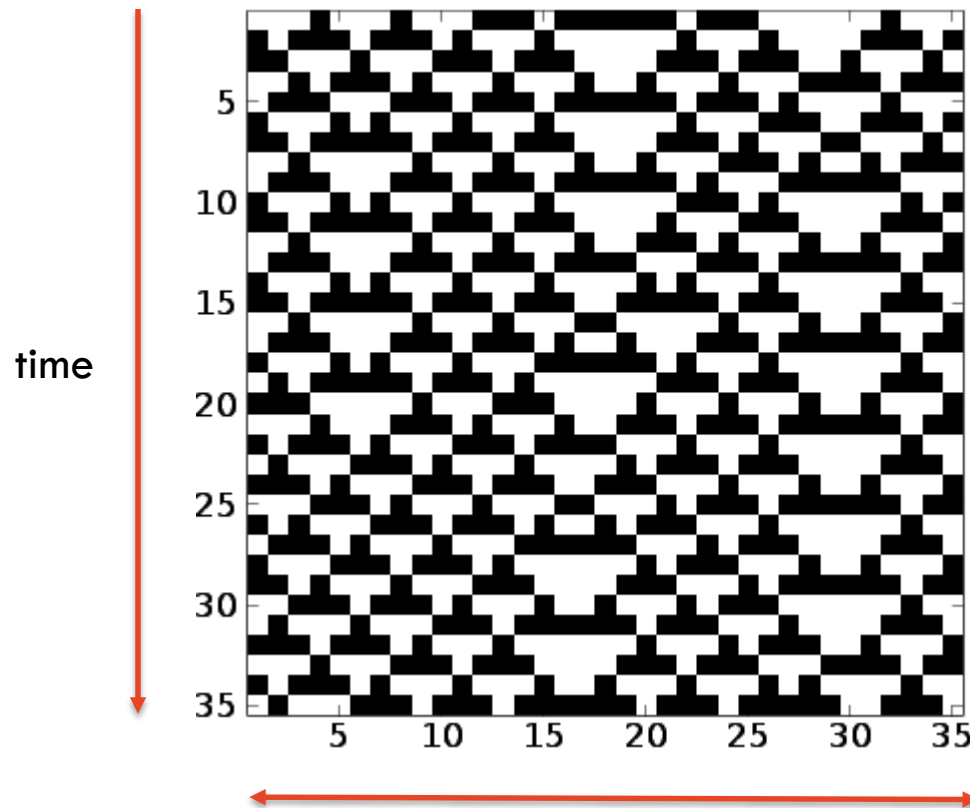
    fprintf('AIS_Discrete(col %d) = %.4f bits (null: %.4f +/- %.4f std dev.; p/s
            v, result, measDist.getMeanOfDistribution(), measDist.get

end
```

Status: See console for all pairs calculation results

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?

– $\mathbf{X}_n^{(k)} = \{X_{n-k+1}, \dots, X_{n-1}, X_n\}$ is a **Takens' embedding** of the past state of X .

– Can use embedding delay τ also: $\mathbf{X}_n^{(k,\tau)} = \{X_{n-(k-1)\tau}, \dots, X_{n-\tau}, X_n\}$

– Want to set embedding for optimal prediction of next value X_{n+1}

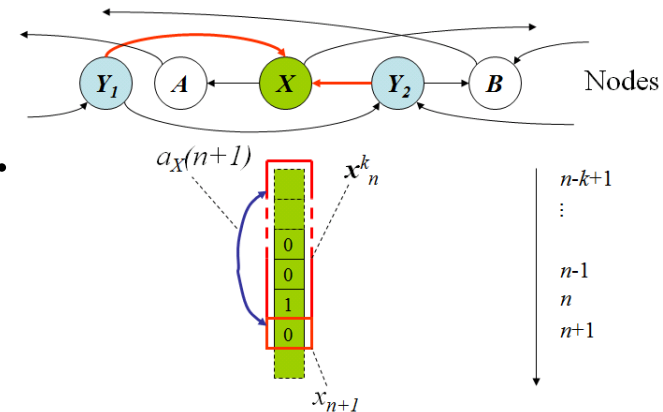
$$p(x_{n+1} | \mathbf{x}_n^{(k)}, x_{n-k}) = p(x_{n+1} | \mathbf{x}_n^{(k)})$$

– 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?

- 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?

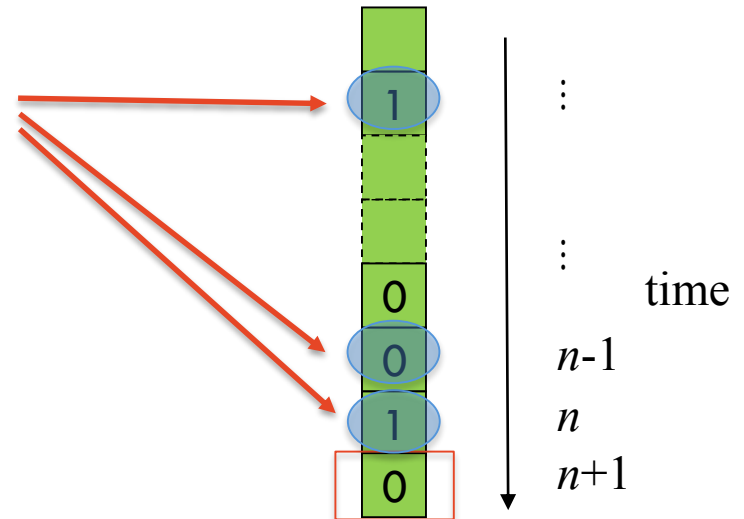
- Option 2: Maximize bias-corrected ALS

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

- (where A_X^S are surrogates, created by destroying the past-next relationship)
 - w.r.t history length k and embedding delay τ
- This implies we include 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 **IDTxI** toolbox

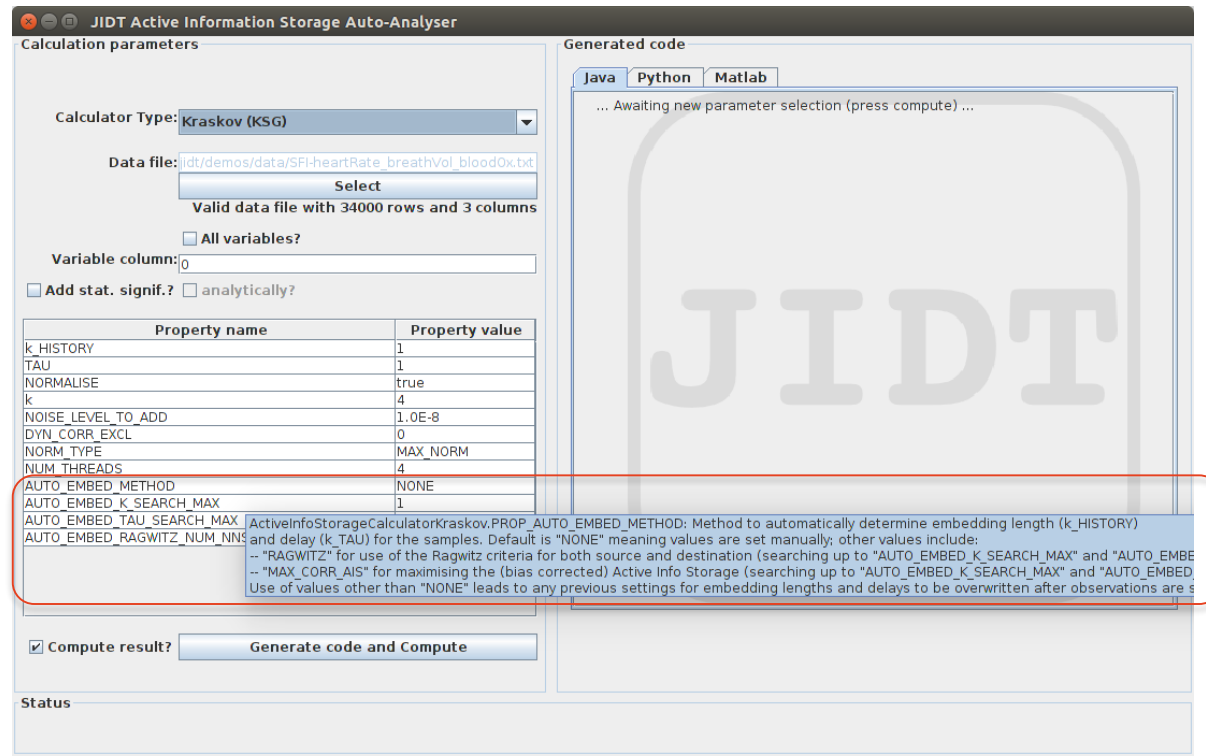


L. Faes, G. Nollo, and A. Porta, “Information-based detection of nonlinear Granger causality in multivariate processes via a nonuniform embedding technique”, Physical Review E 83, 051112+ (2011)

P. Wollstadt, J.T. Lizier, R. Vicente, C. Finn, M. Mart/inez-Zarzuela, P. Mediano, L. Novelli and M. Wibral, “IDTxI: The Information Dynamics Toolkit xI: a Python package for the efficient analysis of multivariate information dynamics in networks”, arXiv:1807.10459, 2018.

AIS: setting embedding parameters in JIDT

- Select KSG or Gaussian 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
k_HISTORY	1
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	NONE
AUTO_EMBED_K_SEARCH_MAX	1
AUTO_EMBED_TAU_SEARCH_MAX	
AUTO_EMBED_RAGWITZ_NUM_NNS	

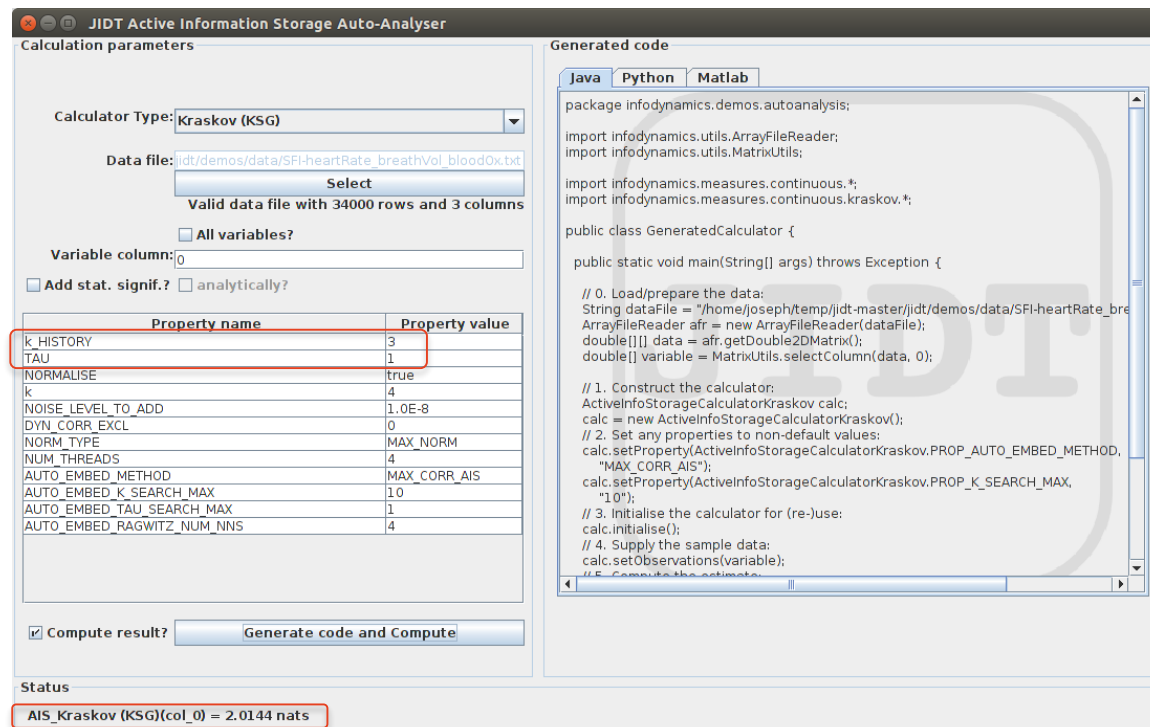
AUTO_EMBED_METHOD tooltip:
ActiveInfoStorageCalculatorKraskov.PROP_AUTO_EMBED_METHOD: Method to automatically determine embedding length (k_HISTORY) and delay (k_TAU) for the samples. Default is "NONE" meaning values are set manually; other values include:
-- "RAGWITZ" for use of the Ragwitz criteria 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_extract.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' with the path '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 properties and their values, 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 imports for 'infodynamics' and 'ArrayFileReader', and a 'main' method that loads data and constructs the calculator.

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_extract.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");

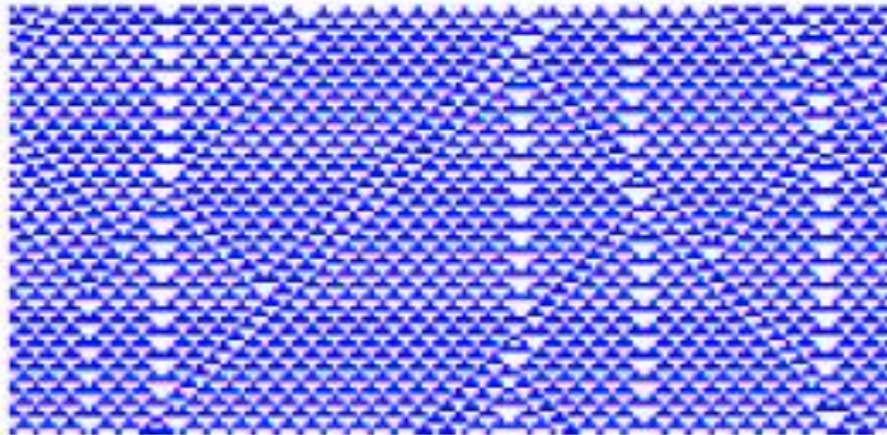
        // 3. Initialise the calculator for (re-)use:
        calc.initialise();

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

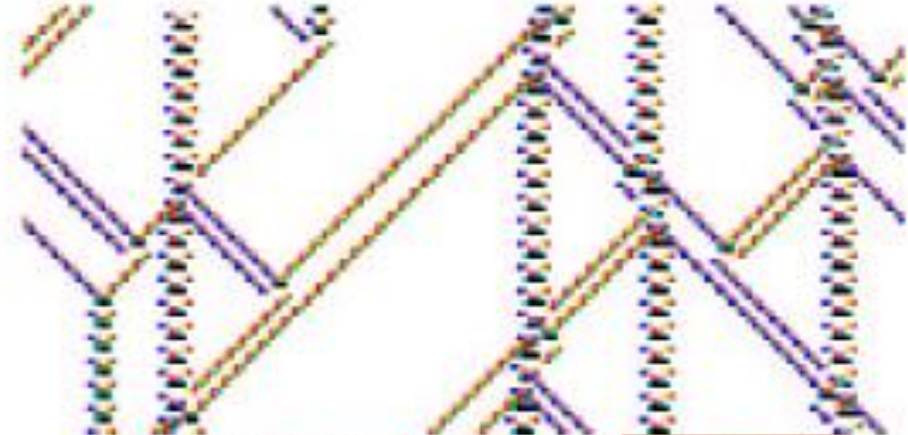
        // 5. Compute the estimate:
    }
}
    
```

At the bottom, the 'Status' section shows the result: `AIK_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, © Wiley, used with permission)

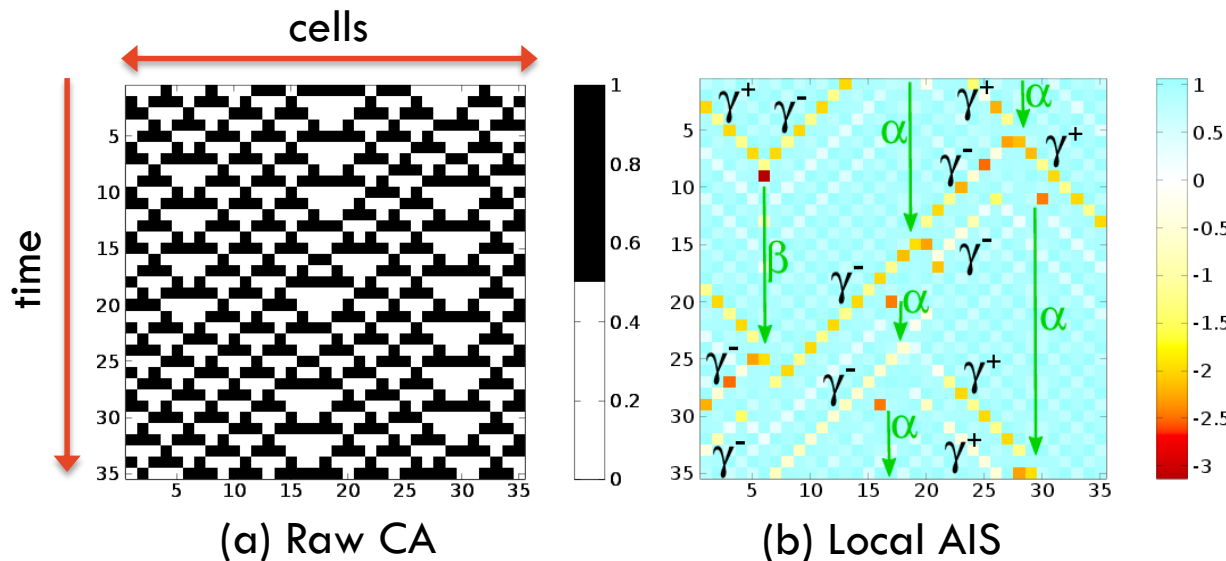
C. G. Langton, “Computation at the edge of chaos: phase transitions and emergent computation,” *Physica D*, vol. 42, no. 1-3, pp. 12–37, 1990.

M. Mitchell, J.P. Crutchfield, R. Das, “*Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work*”, Proc. 1st Int. Conf. Evolutionary Computation and Its Applications (EvCA’96), 1996. (see p. 1/10)

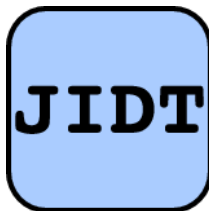


Example: Computational role of emergent structure in CAs

- Go to activities to try the 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.



Spoiler alert:
Blinkers and domains are
dominant storage entities!



J.T. Lizier, M. Prokopenko, A.Y. Zomaya, "Detecting Non-trivial Computation in Complex Dynamics", Proc. ECAL, pp. 895-904 (2007).

J.T. Lizier, M. Prokopenko, & A.Y. Zomaya, "Local measures of information storage in complex distributed computation", Information Sciences 208, 39 (2012)

J.T. Lizier, "JIDT: An information-theoretic toolkit for studying the dynamics of complex systems", Frontiers in Robotics and AI, 1:11, 2014

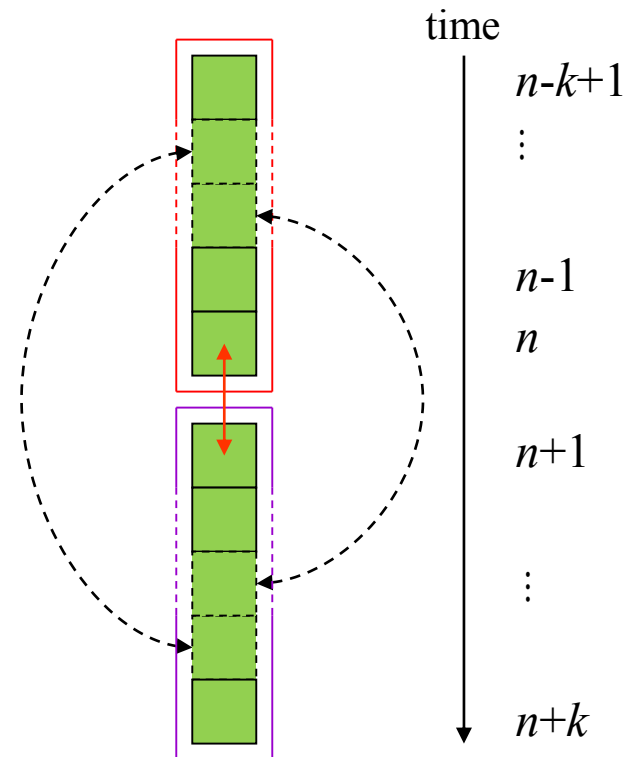
Predictive information

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

$$E_X = \lim_{k \rightarrow \infty} I(\mathbf{X}_n^{(k)}; \mathbf{X}_{n+1}^{(k+)})$$

$$E_X(k) = I(\mathbf{X}_n^{(k)}; \mathbf{X}_{n+1}^{(k+)})$$

- 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 (so is first order component).
 - We're more interested in AIS because it focusses on the computation of the next value and is complementary to information transfer.



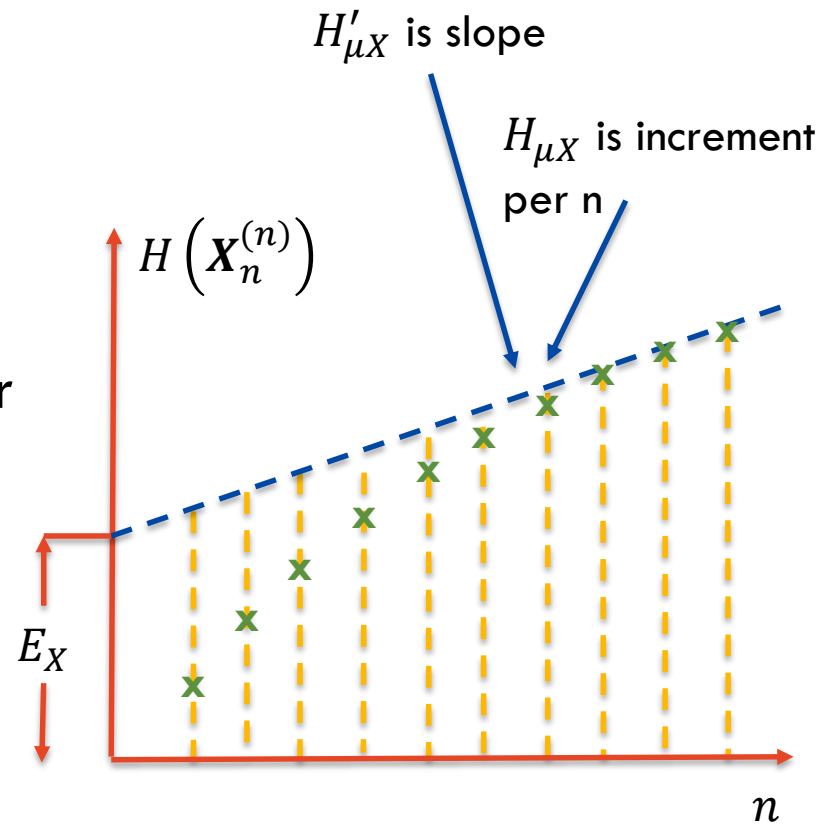
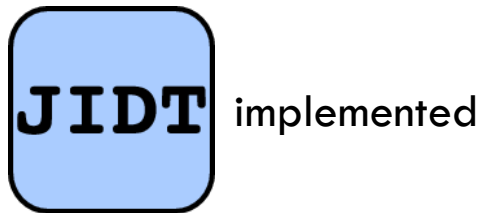
W. Bialek, I. Nemenman, and N. Tishby, "Complexity through nonextensivity", Physica A 302, 89 (2001)

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

J.T. Lizier, "The local information dynamics of distributed computation in complex systems", USyd, 2010. Section 3.2.2.2

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 storage: summary

- We've examined how information storage is characterised.
 - And used JIDT AutoAnalyser and extensions of code to analyse information storage in complex systems data sets.
- *Coming up:* Information transfer in complex systems

Questions



THE UNIVERSITY OF
SYDNEY