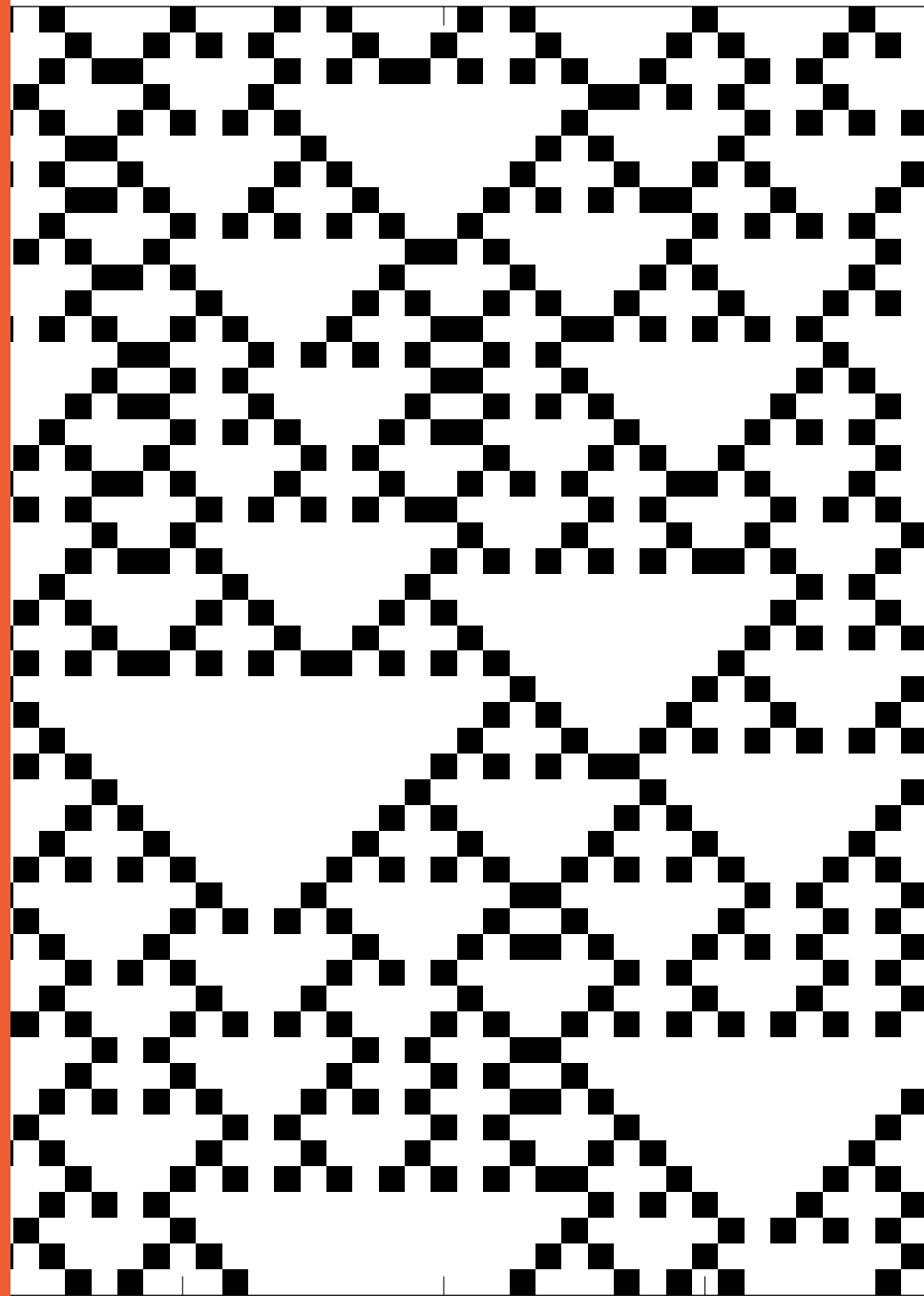


Lecture 5 – Undersampling and statistical significance

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

Statistical significance: session outcomes

- Understand estimation of information as a statistic, and how to test for significance of that statistic.
 - Understand where and how such tests can be performed analytically.
 - Apply statistical significance testing of MI, CMI etc., using JIDT
-
- 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.5

Mutual information

- Recall statistical interpretation of MI from lecture 3:

$$I(X; Y) = 0 \leftrightarrow X \text{ is independent of } Y$$

- In **theory** ...
- In **practice**, or from empirical data:
 - a. We can have X is independent of Y ,
 - b. But measure $I(X; Y) \neq 0$!
- **Q1**: Is a given estimate of $I(X; Y)$ different to or consistent with 0?
- **Q2**: How many samples do we need to determine this?

Aside (Q2) – how many samples do we need?

- Well, that depends on the question you want to answer ... 😊
- One heuristic (Lungarella et al.): have $\geq 3\times$ as many samples as possible state configurations
 - E.g. for $I(X;Y)$ with binary variables, there are 2×2 state configurations.
 - The number of state configurations increases as:
 - The variables have more discrete levels / larger alphabet size, or
 - The variables become multivariate. (Which is equivalent)
 - This assumes all possible state configurations are equally likely to be visited...

M. Lungarella, T. Pegors, D. Bulwinkle, and O. Sporns, "Methods for quantifying the informational structure of sensory and motor data," *Neuroinformatics*, vol. 3, no. 3, pp. 243–262, 2005.

J.T. Lizier, *"The local information dynamics of distributed computation in complex systems"*, Springer: Berlin/Heidelberg, 2013. Section 3.3.1

Aside (Q2) – how many samples do we need?

- Well, that depends on the question you want to answer ... 😊
- But for large multivariate spaces, only a subset of the state configuration space may be explored:
 - The “**typical set**” of state configurations is where the “sample entropy is close to the true entropy” of that joint state.
 - Think of as set of state configurations likely to be encountered frequently enough to contribute to that entropy.
 - This is the set we need to sample well enough, with the number of samples $N \geq 3 \times$ (or in general $\geq M \times$) the size of the typical set.
 - Good expressions for size of typical set for block entropy / entropy rate.

T. M. Cover and J. A. Thomas. Elements of Information Theory. Wiley-Interscience, New York, 1991. Chapter 3

K. Marton and P. C. Shields, “Entropy and the consistent estimation of joint distributions,” The Annals of Probability, vol. 22, no. 2, pp. 960–977, 1994

J.T. Lizier, “The local information dynamics of distributed computation in complex systems”, Springer: Berlin/Heidelberg, 2013. Section 3.3.1

Aside (Q2) – how many samples do we need?

- Well, that depends on the question you want to answer ... 😊
- That's easy enough to work with for plug-in discrete estimator.
- You can adapt it for box-kernel.

Q1: Is estimate $I(X;Y)$ consistent with 0?

- Taking a statistical view, we form a statistical test of $I(X;Y)$:
- **Null hypothesis** H_0 : X is independent of Y
- Alternative hypothesis: X is dependent on Y
- To test $H_0 \rightarrow$ Test probability of sampling the statistic $I(X;Y)$ assuming it is distributed under H_0 :
 1. Form **surrogate** distribution of $I(X;Y^s)$ where Y^s are **surrogates** for Y generated under H_0
 - Which have same statistical properties as Y , but potential relationship to X is destroyed.
 - With $p(x|y)$ distributed as $p(x)$ (whilst $p(y)$ retained)

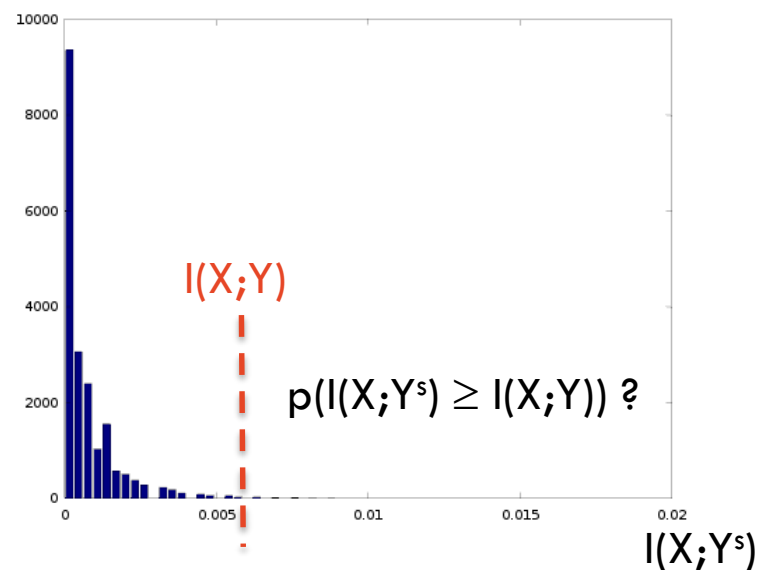
$$I(X;Y) = \sum_{x \in A_x, y \in A_y} p(x,y) \log_2 \frac{p(x|y)}{p(x)}$$

2. Measure p-value of $p(I(X;Y) \leq I(X;Y^s))$ and take one-tailed test against α

Q1: Is estimate $I(X;Y)$ consistent with 0?

- In practice, we generate the surrogate distribution **empirically**:
 - By resampling[†] the samples y of Y to create a surrogate variable Y^{s1} ; which has no per-sample relation to X ;
 - Then computing $I(X;Y^{s1})$;
 - And repeating many times (S) to get the distribution $I(X;Y^s)$.

X	Y	Y ^{s1}	Y ^{s2}	Y ^{s3}	Y ^{s4}	...	Y ^{sS}
x ₁	y ₁	y ₁₀	y ₈	y ₂₇	y ₄₅	...	y ₉₄
x ₂	y ₂	y ₄	y ₃₇	y ₅₈	y ₇₃	...	y ₂₉
x ₃	y ₃	y ₂₃	y ₈₈	y ₃₈	y ₅₅	...	y ₁₃
x ₄	y ₄	y ₅	y ₁₂	y ₄₄	y ₇₆	...	y ₈₉
x ₅	y ₅	y ₇₂	y ₅₁	y ₂₂	y ₁₁	...	y ₃
x ₆	y ₆	y ₁₆	y ₉₉	y ₈₁	y ₂₁	...	y ₆₅
...
	$I(X;Y)$	$I(X;Y^{s1})$	$I(X;Y^{s2})$	$I(X;Y^{s3})$	$I(X;Y^{s4})$...	$I(X;Y^{sS})$

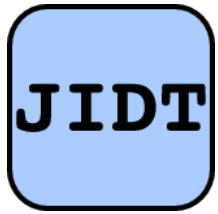


[†] Via permutation or bootstrap sampling.

Q1: Is estimate $I(X;Y)$ consistent with 0?

- Same principle holds for conditional MI $I(X;Y | Z)$, but
 - We generate surrogate distribution $p(x | y,z)$ as $p(x | z)$.
- Notice the difference to $I(X;Y)$?
 - Becomes a directional test here.
 - *Asymptotically* it doesn't matter if we resample x or y
 - Often we're interested in a directional test anyway (e.g. with transfer entropy – see next lecture).

Statistical significance test in JIDT

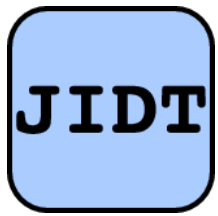


- Generate from AutoAnalyser by clicking the checkbox next to “Add stat. signif.?”
- Available on all MI, CMI and calculators based on these.

The screenshot shows the JIDT Mutual Information Auto-Analyser window. The 'Calculation parameters' section on the left includes a 'Calculator Type' dropdown set to 'Discrete', a 'Data file' field with a path and a 'Select' button, and a status message 'Valid data file with 1000 rows and 2 columns'. Below this are 'Source column' (0) and 'Destination column' (1) fields. A red circle highlights the 'Add stat. signif.?' checkbox, which is checked, next to an unchecked 'analytically?' checkbox. A table at the bottom left shows 'Property name' and 'Property value' for 'base' (2) and 'time difference' (0). At the bottom, there is a 'Compute result?' checkbox (checked) and a 'Generate code and Compute' button. The right side, 'Generated code', has tabs for 'Java', 'Python', and 'Matlab', with a message '...Awaiting new parameter selection (press compute) ...' and a large 'JIDT' watermark.

Property name	Property value
base	2
time difference	0

Statistical significance test in JIDT



- `Calculator.getSignificance(numPermutations)` returns an `EmpiricalMeasurementDistribution` object (see Javadocs) from which you can retrieve full surrogate distribution, mean, std dev, and p-value of statistic.

JIDT Mutual Information 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 pairs?

Source column:

Destination column:

☒ Add stat. signif. ☐ analytically?

Property name	Property value
base	2
time difference	0

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

Generated code

Matlab

```
% Add JIDT jar library to the path
javaaddpath('.../infodynamics.jar');
% Add utilities to the path
addpath('.../octave');

% 0. Load/prepare the data:
data = load('/home/joseph/temp/jidt-master/jidt/demos/data/2coupledBinaryColsUseK2.t
% Column indices start from 1 in Matlab:
source = octaveToJavaIntArray(data(:,1));
destination = octaveToJavaIntArray(data(:,2));

% 1. Construct the calculator:
calc = javaObject('infodynamics.measures.discrete.MutualInformationCalculatorDiscrete'
% 2. No other properties to set for discrete calculators.
% 3. Initialise the calculator for (re-)use:
calc.initialise();
% 4. Supply the sample data:
calc.addObservations(source, destination);
% 5. Compute the estimate:
result = calc.computeAverageLocalOfObservations();
% 6. Compute the (statistical significance via) null distribution (e.g. 100 permutations):
measDist = calc.computeSignificance(100);

fprintf('MI_Discrete(col_0 -> col_1) = %.4f bits (null: %.4f +/- %.4f std dev.; p(surrogate >
        result, measDist.getMeanOfDistribution(), measDist.getStdOfDistribution())
```

Status

MI_Discrete(col_0 -> col_1) = 0.0001 bits (null: 0.0007 +/- 0.0009 std dev.; p(surrogate > measured)=0.73000 from 100 surrogates)

Analytic surrogate distributions

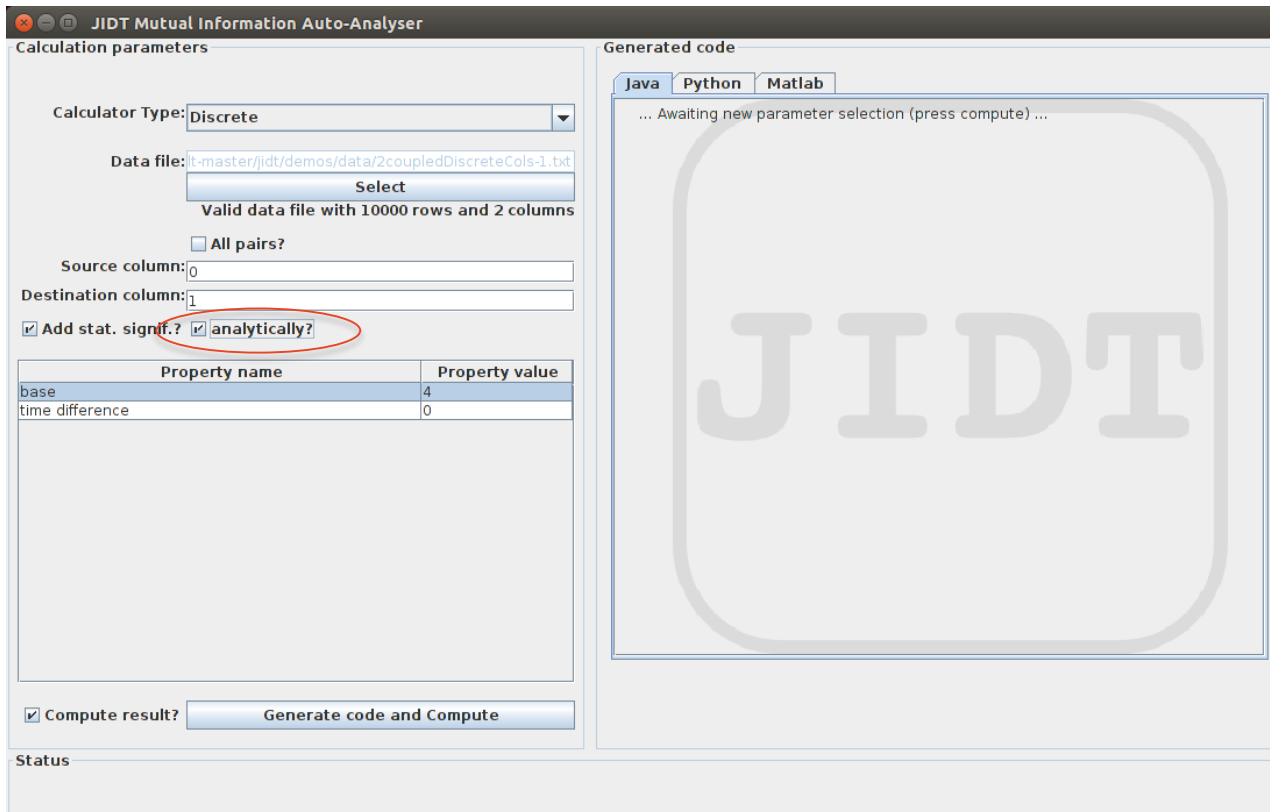
- For some estimators, we have an analytic representation of the surrogate distribution.
- Specifically, $2N \times I(X; Y^s)$ or $2N \times I(X; Y^s | Z)$ in *nats* follow χ^2 distributions with the following degrees of freedom:

Estimator	Mutual info $I(X; Y^s)$	Conditional Mutual Info $I(X; Y^s Z)$
Linear-Gaussian	$ X Y $	$ X Y $
Discrete (plug-in)	$(A_X - 1)(A_Y - 1)$	$(A_X - 1)(A_Y - 1) A_Z $

- Where:
 - $|X|$ means the number of dimensions in a multivariate X
 - $|A_X|$ means the alphabet size of discrete variable X .
 - Discrete estimate is converted to nats in this distribution!

Analytic statistical significance test in JIDT

- This is far faster than empirical surrogate generation.
- Generate from AutoAnalyser by clicking the checkbox next to “analytic?” near “Add stat. signif.?”



The screenshot shows the JIDT Mutual Information Auto-Analyser window. The 'Calculation parameters' section on the left contains the following settings:

- Calculator Type: Discrete
- Data file: t-master/jidt/demos/data/2coupledDiscreteCols-1.txt (with a 'Select' button)
- Valid data file with 10000 rows and 2 columns
- ☐ All pairs?
- Source column: 0
- Destination column: 1
- ☒ Add stat. signif.? ☒ analytically? (The 'analytically?' checkbox is circled in red)

Below these settings is a table with the following data:

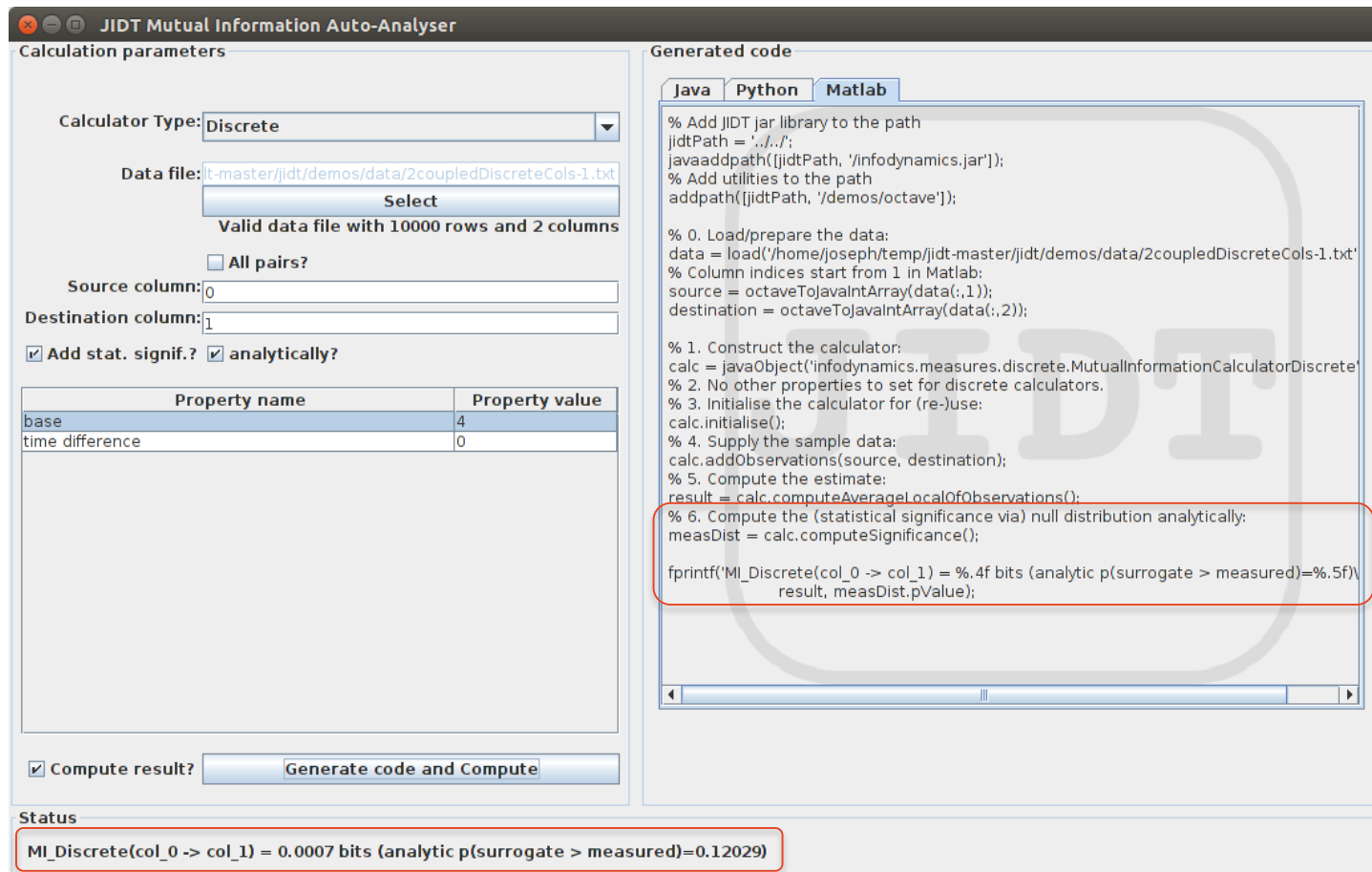
Property name	Property value
base	4
time difference	0

At the bottom of the 'Calculation parameters' section, there is a checkbox for 'Compute result?' which is checked, and a button labeled 'Generate code and Compute'.

The 'Generated code' section on the right has tabs for 'Java', 'Python', and 'Matlab'. The 'Java' tab is selected, and the text inside reads: '... Awaiting new parameter selection (press compute) ...'. A large, faint 'JIDT' watermark is visible in the background of the generated code area.

Analytic statistical significance test in JIDT

- `Calculator.getSignificance()` returns an `AnalyticMeasurementDistribution` object (see Javadocs) from which you can retrieve p-value of statistic, and convert between estimates \leftrightarrow p-values.



JIDT Mutual Information Auto-Analyser

Calculation parameters

Calculator Type: **Discrete**

Data file: `t-master/jidt/demos/data/2coupledDiscreteCols-1.txt`
 Select
 Valid data file with 10000 rows and 2 columns

☐ All pairs?

Source column: `0`

Destination column: `1`

☒ Add stat. signif.? ☒ analytically?

Property name	Property value
base	4
time difference	0

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

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/2coupledDiscreteCols-1.txt')
% Column indices start from 1 in Matlab:
source = octaveToJavaIntArray(data(:,1));
destination = octaveToJavaIntArray(data(:,2));

% 1. Construct the calculator:
calc = javaObject('infodynamics.measures.discrete.MutualInformationCalculatorDiscrete')
% 2. No other properties to set for discrete calculators.
% 3. Initialise the calculator for (re-)use:
calc.initialise();
% 4. Supply the sample data:
calc.addObservations(source, destination);
% 5. Compute the estimate:
result = calc.computeAverageLocalOfObservations();
% 6. Compute the (statistical significance via) null distribution analytically:
measDist = calc.computeSignificance();

fprintf('MI_Discrete(col_0 -> col_1) = %.4f bits (analytic p(surrogate > measured)=%.5f)\n',
        result, measDist.pValue);
```

Status

MI_Discrete(col_0 -> col_1) = 0.0007 bits (analytic p(surrogate > measured)=0.12029)

Analytic statistical significance

- Pros:
 - far faster than empirical
- Cons:
 - Is only completely correct asymptotically as $N \rightarrow \infty$. (But we only care about it when N is finite!)
 - Can be significantly away from empirical values when:
 - Distributions under analysis are highly multivariate (increasing undersampling effects), or
 - (for discrete estimator) where the distributions on the variables are heavily skewed.
- More details in demos/octave/NullDistributions (see on [wiki](#))

Aside: normalising measurements

- Sometimes we normalise information estimates by removing the component due to finite sample size:

$$I^n(X; Y) = I(X; Y) - \langle I(X; Y^s) \rangle$$

- This is equivalent to bias correction (so not necessary if bias correction works well).

Statistical significance : summary

- We've reviewed estimation of information as a statistic, and how to test for significance of that statistic:
 - Empirically, and
 - Analytically where possible.
- You know how to apply statistical significance testing of MI, CMI etc., using JIDT
- *Next lecture:* Information processing in complex systems.

Questions



THE UNIVERSITY OF
SYDNEY