Measuring Information Transfer – Thomas Schreiber

Entropy Discussion Group
3/5/2006

Joseph Lizier
PhD student
The University of Sydney – Prof. Albert Zomaya
CSIRO ICT Centre - Dr. Mikhail Prokopenko

CSIRO

Measuring Information Transfer – Thomas Schreiber

VOLUME 85, NUMBER 2 PHYSICAL REVIEW LETTERS 10 JULY 2000

Measuring Information Transfer

Thomas Schreiber
Max Planck Institute for the Physics of Complex Systems, Nöthnitzer Strasse 38, 01187 Dresden, Germany
(Received 19 January 2000)

An information theoretic measure is derived that quantifies the statistical coherence between systems evolving in time. The standard time delayed mutual information fails to distinguish information that is actually exchanged from shared information due to common history and input signals. In our new approach, these influences are excluded by appropriate conditioning of transition probabilities. The resulting *transfer entropy* is able to distinguish effectively driving and responding elements and to detect asymmetry in the interaction of subsystems.

- *Aim*: Improve on standard use of Mutual Information (MI) to quantify info transfer, addressing MI's:
  - Symmetric nature
  - Static nature
  - lack of discrimination against common history and input signals

CSIRO

Contents

1. Define Mutual Information, and describe inadequacies
2. Derive Transfer Entropy metric
3. Clarify use with examples, considering:
   1. Discrete/discretized systems
   2. Continuous systems

Build definition of Mutual Information (1)

- Shannon entropy – $H_I = -\sum_i p(i) \log_2 p(i)$
  - *"Average number of bits needed to optimally encode independent draws of the discrete variable I following probability distribution p(i)"*
- Kullback entropy – $K_I = \sum_i p(i) \log p(i)/q(i)$
  - *"Excess number of bits coded if a different distribution q(i) is used"* … for the coding instead of the actual underlying distribution p(i)
  - Can also be defined for conditional probabilities:

$$K_{I|J} = \sum_{i,j} p(i,j) \log \frac{p(i|j)}{q(i|j)}$$

CSIRO

Build definition of Mutual Information (2)

- Mutual Information – $$M_{IJ} = \sum p(i,j) \log \frac{p(i,j)}{p(i)p(j)}$$
  - View via Kullback entropy as "*Excess amount of code produced by erronously assuming that the two systems are independent*" … i.e. using q(i,j) = p(i)p(j) for a coding.
  - Can show – $M_{IJ} = H_I + H_J - H_{IJ} \geq 0$.
  - Clearly MI is completely symmetric in i, j
    - No directional sense to define information transfer
  - Ad-hoc method of adding directional sense: MI with time lag –

$$M_{IJ}(\tau) = \sum p(i_n, j_{n-\tau}) \log \frac{p(i_n, j_{n-\tau})}{p(i)p(j)}$$

    - Still isn't accounting for system dynamics

CSIRO

Aside: Conditional entropy

- Conditional entropy –

$$H_{I|J} = -\sum p(i,j) \log p(i|j)$$

  - Is asymmetric $= H_{IJ} - H_J$
  - But as $H_{I|J} - H_{J|I} = H_I - H_J$
    - "*nonsymmetric only due to different individual entropies and not due to information flow*"

Build definition of Transfer Entropy (1)

- Move to transition (dynamic) probabilities rather than static probabilities.
- Entropy rate $h_I = -\sum p(i_{n+1}, i_n^{(k)}) \log p(i_{n+1} \mid i_n^{(k)})$
  - Is a conditional entropy – "*average number of bits needed to encode one additional state of the system if all previous states are known*"
  - Is also the difference in Shannon entropy in using k+1 and k delay vectors

$$h_I = H_{I^{(k+1)}} - H_{I^{(k)}}$$

Build definition of Transfer Entropy (2)

- Direct generalisation of entropy rate to two process is still symmetric.
- Transfer entropy –

$$T_{J \to I} = \sum p(i_{n+1}, i_n^{(k)}, j_n^{(l)}) \log \frac{p(i_{n+1} \mid i_n^{(k)}, j_n^{(l)})}{p(i_{n+1} \mid i_n^{(k)})}$$

  - Construct via Kullback entropy – Excess amount of code produced in coding "destination" variable *i* by assuming that the next state of the destination variable is independent of the "source" variable *j*, i.e. using $q = p(i_{n+1} \mid i_n^{(k)})$ instead of $p(i_{n+1} \mid i_n^{(k)}, j_n^{(l)})$.
  - Quantifies influence of process J on transition probabilities of system I.
  - $T_{J \to I}$ Is explicitly asymmetric.
  - Exclude influence of common driving force Z by conditioning probabilities under logarithm to z_n also.

CSIRO

Transfer entropy in discrete(-ized) systems

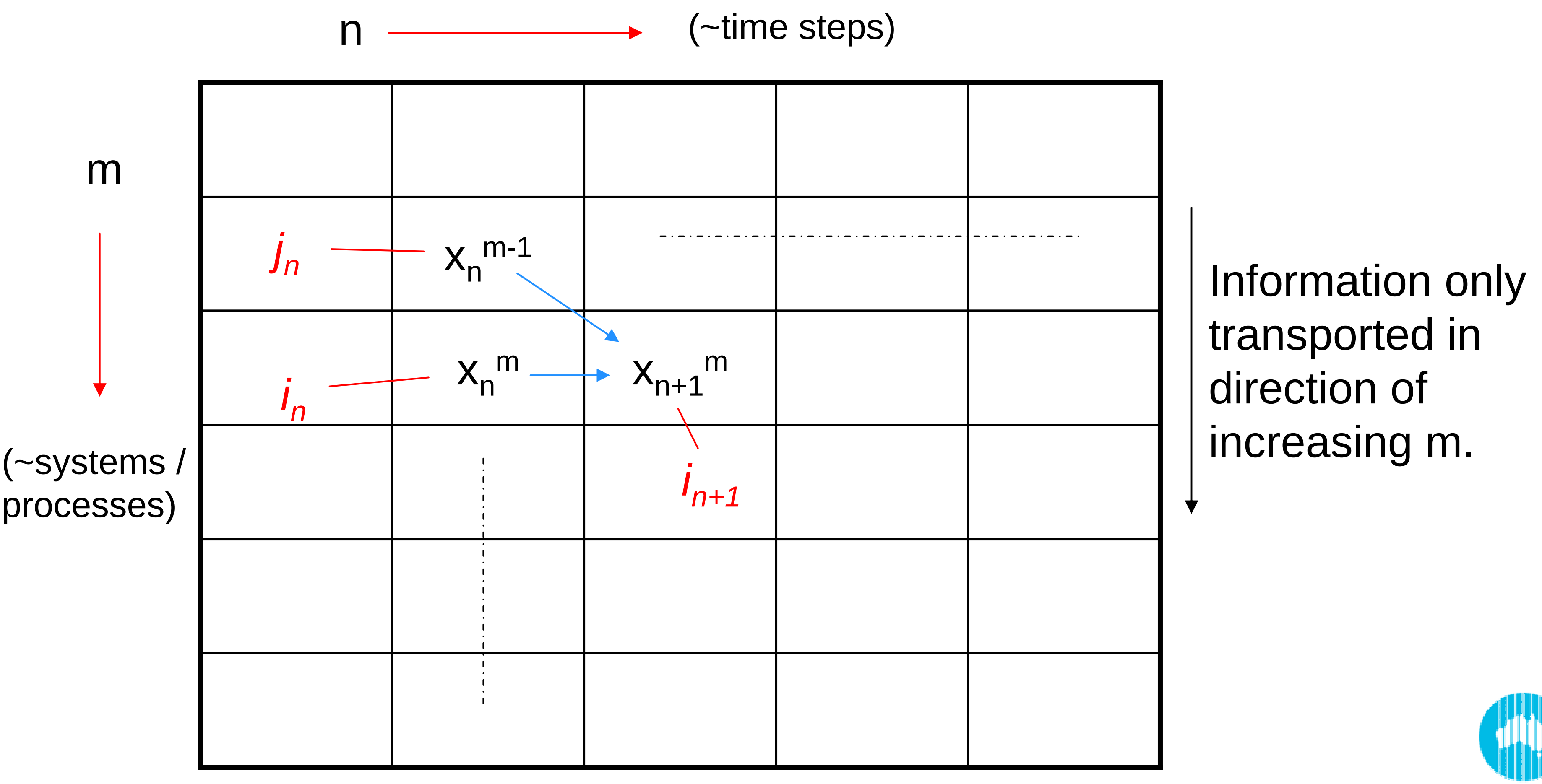
- Computation of $T_{J \to I}$ is relatively straightforward because probabilities are straightforward.
- For each $(i_{n+1}, i_n^{(k)}, j_n^{(l)})$ *possible* discrete valued tuple:
  - $p(i_{n+1}, i_n^{(k)}, j_n^{(l)})$ – count of the number of occurrences of the given transition divided by the total number of transitions observed.
  - $p(i_{n+1}| i_n^{(k)}, j_n^{(l)}) = p(i_{n+1}, i_n^{(k)}, j_n^{(l)}) / p(i_n^{(k)}, j_n^{(l)})$ – for $p(i_n^{(k)}, j_n^{(l)})$ count of the number of occurrences of $(i_n^{(k)}, j_n^{(l)})$ divided by the total number of system states observed
  - $p(i_{n+1}| i_n^{(k)}) = p(i_{n+1}, i_n^{(k)}) / p(i_n^{(k)})$ – for $p(i_{n+1}, i_n^{(k)})$ as for $p(i_n^{(k)}, j_n^{(l)})$, for $p(i_n^{(k)})$ count of the number of occurrences of $i_n^{(k)}$ divided by the total number of system states observed.
- Compute $T_{J \to I}$ summing over all *possible* transition tuples.

$$T_{J \to I} = \sum p(i_{n+1}, i_n^{(k)}, j_n^{(l)}) \log \frac{p(i_{n+1} \mid i_n^{(k)}, j_n^{(l)})}{p(i_{n+1} \mid i_n^{(k)})}$$

CSIRO

Examples – Unidirectional coupled maps

- One dimensional lattice: $x^m_{n+1} = f(\epsilon x^{m-1}_n + (1-\epsilon)x^m_n)$

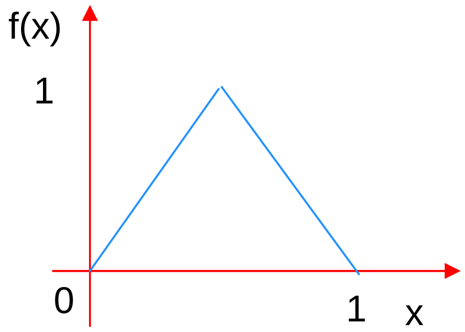


Discrete example – Tent map (1)

- Use of tent mapping for f(x):

$$f(x < 0.5) = 2x;\ f(x \geq 0.5) = 2 - 2x$$

$0 \leq x \leq 1$



- Using continuous x to propagate mapping, but
- Discretizing x into $x \leq 0.5 \rightarrow 0$ and $x \geq 0.5 \rightarrow 1$ to record state transition tuples $(i_{n+1}, i_n^{(1)}, j_n^{(1)})$
- e.g.

$\epsilon=0.02$

0.66	0.69
0.34	0.69
0.29	0.59
0.17	0.34
0.58	0.87

discretize →

1	1
0	1
0	1
0	0
1	1

Tuples :

1, 0, 1

1, 0, 0

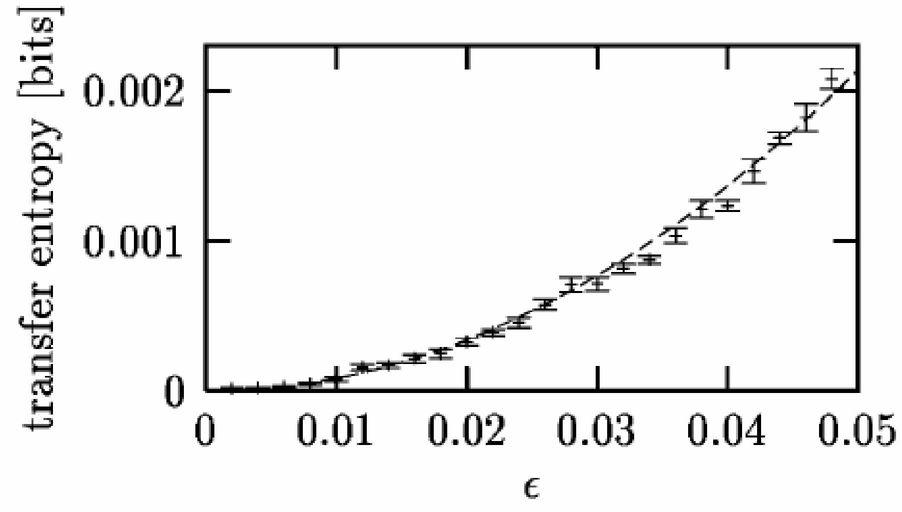
0, 0, 0

1, 1, 0

CSIRO

Discrete example – Tent map (2)

- Results:
  - $M_{m,m-1}(\tau=1) = M_{m-1,m}(\tau=1) = 0 \leftarrow$ Not useful, doesn't deal with dynamics of system.
  - $T_{l^m \rightarrow l^{m-1}} = 0$. As expected, no information transfered in reverse direction.
  - $T_{l^{m-1} \rightarrow l^m} > 0$, and rises with coupling as expected.
    - Analytical prediction: $T \propto \epsilon^2$ for low coupling.
    - Numerically verified:



CSIRO

Transfer entropy in continuous systems (1)

- Continuous system (X,Y) needs to be coarse grained (I,J) at resolution r to compute entropies.

In limit as $r \to 0$ for:	Deterministically dynamical system / coupled processes	Generic noisy interdependence
Entropy and entropy rate:	$\lim_{r \to 0} h_X(r)$ may exist (Kolmogorov-Sinai entropy)	Depends on partitioning, diverges like –log r.
Mutual information and transfer entropy:	Diverges	$\lim_{r \to 0}$ is finite and indpt of partition.

- $r \to 0$ not practically attainable: either look at $T_{X \to Y}$ as a function of r, or fix r.

Transfer entropy in continuous systems (2)

- Computation of $T_{J \to I}$ is less straightforward because the probabilities are less straightforward.
- For each $(i_{n+1}, i_n^{(k)}, j_n^{(l)})$ tuple *realized* in the given observations:
  - $p(i_{n+1}, i_n^{(k)}, j_n^{(l)})$ – use kernel estimation to obtain (e.g. for $p(i_{n+1},i_n,j_n)$):

$$\hat{p}_r(x_{n+1}, x_n, y_n) = \frac{1}{N}\sum_{n'} \Theta\left(\left| \begin{pmatrix} x_{n+1} - x_{n'+1} \\ x_n - x_{n'} \\ y_n - y_{n'} \end{pmatrix} \right| - r \right)$$

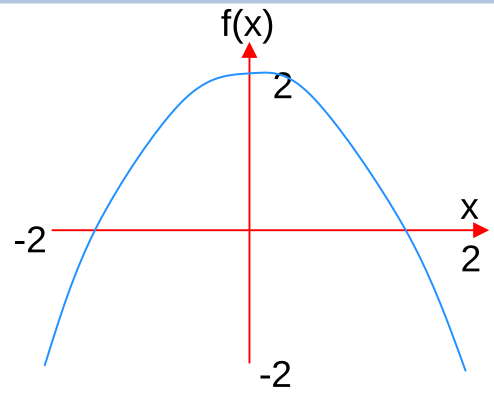
× – reverse the terms to $r - |t_n - t_{n'}|$

  - Perform similar kernel estimation for the transition probabilities in:
    - $p(i_{n+1}| i_n^{(k)}, j_n^{(l)}) = p(i_{n+1}, i_n^{(k)}, j_n^{(l)}) / p(i_n^{(k)}, j_n^{(l)})$, and
    - $p(i_{n+1}| i_n^{(k)}) = p(i_{n+1}, i_n^{(k)}) / p(i_n^{(k)})$
- Compute $T_{J \to I}$ summing over all *realized* transition tuples:
  - Renormalise for $p(i_{n+1}, i_n^{(k)}, j_n^{(l)})$ in sum (but don't renormalise each component in the transition probabilities – their ratios to eachother are important.)

$$T_{J\to I} = \sum p(i_{n+1}, i_n^{(k)}, j_n^{(l)}) \log \frac{p(i_{n+1} \mid i_n^{(k)}, j_n^{(l)})}{p(i_{n+1} \mid i_n^{(k)})}$$

CSIRO

Continuous example – Ulam map (1)



- Use of Ulam mapping for f(x):

$$f(x) = 2 - x^2$$

$-2 \le x \le 2$

- Using continuous x to propagate mapping, and
- Using kernel estimation to compute probabilities from recorded state transition tuples $(i_{n+1}, i_n^{(1)}, j_n^{(1)})$
- e.g.

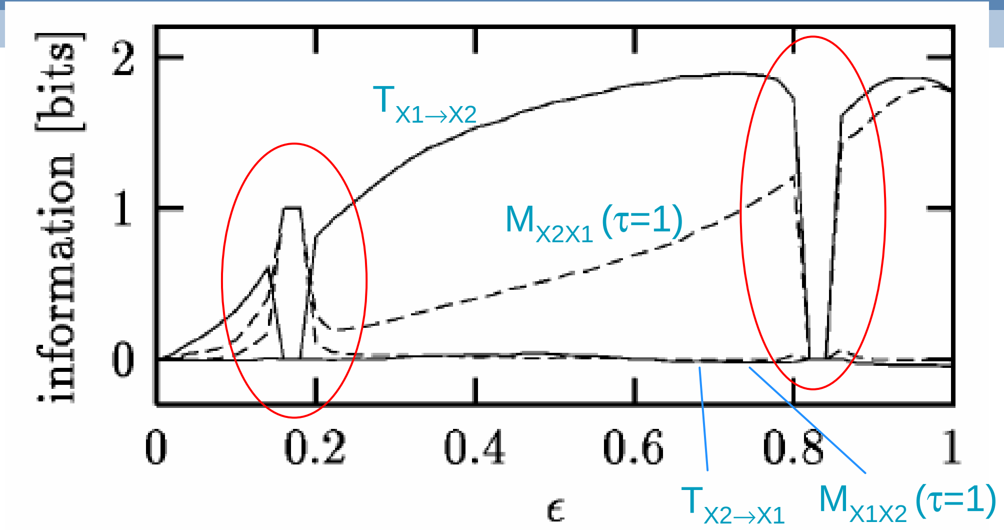
$\epsilon=0.3$

-1.17	1.67
1.91	1.03
1.96	-1.79
1.97	-1.86
0.83	0.62

Tuples :		$\hat{p}_{0.2}(i_{n+1}, i_n^{(1)}, j_n^{(1)})$:		
1.03, 1.91, -1.17		1 / 4		0.166
-1.79, 1.96, 1.91	Kernel est. →	2 / 4	renorm →	0.333
-1.86, 1.97, 1.96		2 / 4		0.333
0.62, 0.83, 1.97		1 / 4		0.166

CSIRO

Continuous example – Ulam map (2)

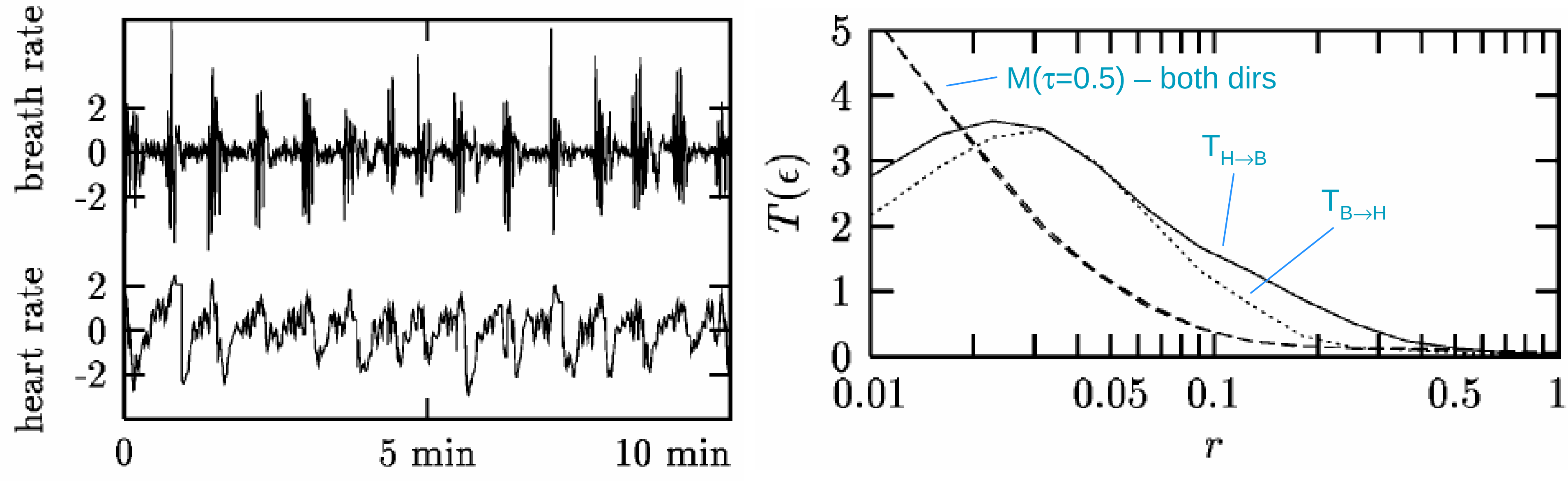


- Results (fixed *r*=0.2):
  - $T_{X^1 \rightarrow X^2}$ and $M_{X^2X^1}(\tau=1)$ (denoting direction of info flow m-1 to m) being generally larger reflect the general direction of information flow.
  - Different behaviours shown at system bifurcations:
    - $\epsilon \approx 0.18$ – System $\rightarrow$ temporal & spatial period 2
      - M = 1 (both directions) – correctly represents static correlation between the maps
      - T = 0 – as desired, because there is no information being transported
    - $\epsilon \approx 0.82$ – System $\rightarrow$ fixed temporal state, spatial period 2
      - M = 0 (both directions) and T = 0, both correctly showing zero info transfer
  - Only transfer entropy consistently gives intuitively correct results for information transfer.

CSIRO

Continuous example – Heart and breath

- Study of M and T of breath and heart rates



- Time delayed M shows no difference.
- T indicates a stronger flow of information from heart rate to breath rate:
  - Consistent with observation.
  - Be wary of applying labels “drive” and “response”.

Summary table – dynamic vs static info

- Transfer entropy designed here to:
  - “detect directed exchange of information between two systems”
  - “ignore static correlations”

	Static probabilities	Dynamic probabilities
Single process	Entropy	Entropy rate ?
Joint processes	Mutual information	Transfer entropy ?

CSIRO

Octave / Matlab code of examples

www.ict.csiro.au

- http://www.it.usyd.edu.au/~jlizier/infotx/