Networks Formed by Chaotic Units

NCASO/NLP Project, Spring 2010

Charlie Brummitt
cbrummitt@math.ucdavis.edu
Graduate Group in Applied Mathematics
University of California, Davis

May 31, 2010

Abstract

We first study a new link rewiring algorithm for a network of coupled one-dimensional maps that modifies the model in [2, 3]. Beginning from an Erdős-Rènyi random graph, the rewiring algorithm produces a graph with a large clustering coefficient and skewed degree distribution. Furthermore, we show that if the network's nodes consist of two different maps, these tend to coalesce into homogeneous communities under the rewiring algorithm. Second, we numerically show that, if given only the time series of the nodes' iterates converted to a discrete alphabet via a generating partition, one can recover the topology of a small, static network by calculating information-theoretic quantities.

Keywords: network evolution, chaotic maps, mutual information, transfer entropy

1 Part I: A new network rewiring algorithm for networks with chaotic units

1.1 Introduction

One of the most important questions in network science is how to generate networks with properties that characterize real world systems. For example, one of the earliest models, that of Watts and Strogatz, randomly and independently rewires the links in a regular grid, thus interpolating between regular graphs and random graphs—and the so-called small-world networks in between. In another popular model, Preferential Attachment, as nodes are successively added to the network they form links with other nodes with probability proportional to their degrees—a process that generates scale-free networks with power-law distributed degree distributions, a characteristic of many real world networks, from the Internet to neurons to power grids to social contacts [1].

What is common among many network models—including the Watts-Strogatz and Preferential Attachment models—is that their nodes are *static*; only the links are dynamic. But nodes in real world systems—from neurons to internet routers to airports to people—are dynamic in their connections *and* in their state. This motivates a model of network generation that incorporates both nodes' states and connections.

1.2 Previous network rewiring model ("Global Rewiring")

Two papers in Europhysics Letters in 2004 [2, 3] report on one such model of network structure formed by the dynamics on the nodes. Their model works as follows. $N = 300$ many logistic maps are coupled on a network (i.e., each node is a logistic map). Initially the network connections are random: $L = 5200$ many links are formed uniformly at random, which is sufficiently large in number to ensure that the network is fully connected (i.e., $L \gg \frac{N}{2} log(N)$). The dynamics at each node i are

$$x_i(t+1) = (1-\epsilon)f(x_i(t)) + \frac{\epsilon}{k_i}\sum_{j\in B(i)} f(x_j(t)) \tag{1}$$

where $B(i)$ are the neighbors of node i, k_i is the degree of node i, and

$$f(x) = 1 - ax^2 \tag{2}$$

is the quadratic map on $[-1, 1]$ (which is conjugate to the logistic map). In words, each node evolves according to its own logistic map and an evenly weighted sum over its neighbors. In [2, 3] they fix $a = 1.7$ (in the chaotic range) and the coupling parameter $\epsilon = 0.4$ (so that the majority of the dynamics, 0.6, depend on the node's own state). Next, choose initial conditions uniformly at random in $(-1, 1)$ for each node, and iterate some number of time steps T so that the system approaches its attractor.

Now begins the rewiring:

1. Choose a node at random (the "pivot")

2. Find which other node is most coherent with the pivot, i.e., which minimizes $d_{ij}(t) = |x_i(t) - x_j(t)|$, where t is the current time. (This node is called the "candidate.")

3. If the candidate is already connected to the pivot, do nothing.

4. If the candidate is not connected to the pivot, then form a link between the two, and sever the link between the pivot and its least coherent neighbor (i.e., the neighbor k that maximizes $d_{ik}(t) = |x_i(t) - x_k(t)|$).

Using this algorithm, [2] find that the network evolves from a random graph to a small-world network with Poisson degree distribution, large clustering coefficient (≈0.6 after 20,000 iterations). They note interesting dynamics in the histogram of coherences d_{ij}: as illustrated in Fig. 1, the coherences alternate between synchronization and desynchronization with intermittent behavior in between.

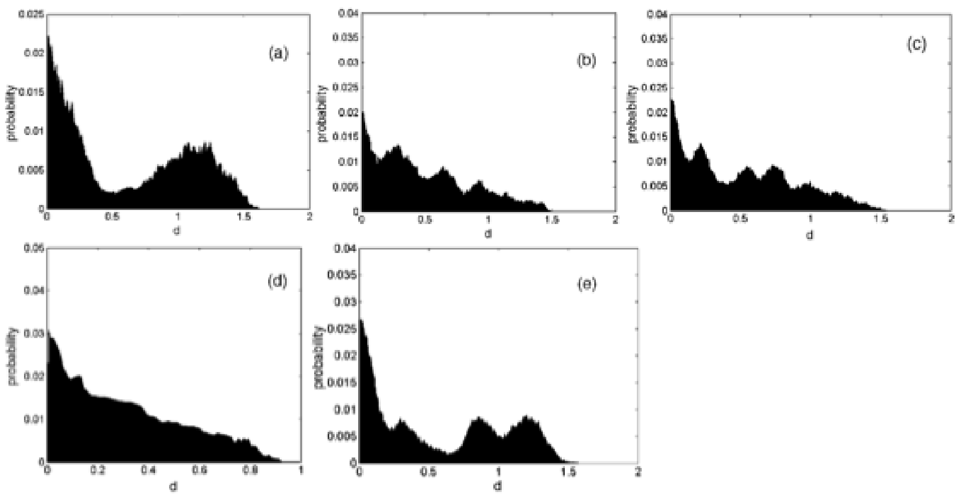


Figure 1: Histograms of coherences d_{ij} for different times: (a) after 7900 iterations; (b) after 8900 iterations; (c) after 9200 iterations; (d) after 9608 iterations; (e) after 9800 iterations. From reference [2].

The authors of [3] further explore the properties of this model. Intriguingly, they find that the network forms clusters (tightly connected communities), the number of which increases roughly linearly with the number of nodes N. Moreover, below a critical population of nodes the process leads to a topology that constantly changes, whereas above that critical population the topology settles onto that of a small-world network.

This model in [2, 3] has two particularly intriguing properties. First, the authors in both papers [2, 3] claim that this is a toy model for neurogenesis. That this may hold some truth is evidenced by another model of logistic maps on a network [5] that attempts to emulate the visual cortex in the brain. In this model, the nodes (i.e., logistic maps) start out with parameters in the chaotic regime, because chaos is thought to be the natural state of perception neurons, and then the nodes affect one another's parameter. Knocking nodes into limit cycles or fixed points corresponds to perception. The second virtue of this algorithm is the autonomy of the rewiring scheme: the nodes rewire connections depending on the nodes' dynamics, not some external source.

However, the model suffers from two drawbacks. First, the rewiring process is not entirely autonomous, as it requires choosing nodes at random and so needs a random number generator (which contradicts the claims in [2, 3] that the nodes are autonomous). Second, nodes form new links with the most coherent node, no matter where that node lies in the network, which is unrealistic for many real world systems.

1.3 Modified rewiring algorithm ("Local Rewiring")

We propose two changes to the rewiring algorithm in [2, 3] that fix both of these artificial properties (i.e., who rewires, and to whom do they form a link?). First, rather than choosing one node at random to attempt to rewire a connection, nodes will rewire connections whenever their total decoherence d_{ij} with its neighbors exceeds a certain threshold τ. This new parameter τ controls the rate at which nodes rewire their links: set it high, and nodes rarely rewire; set it low, and they frequently sever old connections and forge new ones. This threshold mechanism has the following interpretation: nodes sufficiently out of sync with their neighbors will get "fed up" and attempt to form a new link so as to achieve greater coherence. In this way, the nodes truly are autonomous; there is no external source that randomly chooses a node to rewire at each time step, as in [2, 3]. Instead, all nodes are endowed with the same threshold τ that dictates when they should rewire their links to other nodes.

The second change to the rewiring algorithm modifies to whom a node forms a new connection. Rather than forming a link with the most coherent node in the network, they form a link with the most coherent node in the network that are accessible by a path of length s, where $1 < s \leq \sigma$. The parameter σ dictates how many "hops" away in the network a node can "look" as it searches for a new node to link with. In many cases, we set σ to 2 but sometimes 3 (and not higher, as the diameter of most networks under consideration is usually 3 or 4). The interpretation of $\sigma = 2$ is as follows: after severing a link with its least coherent neighbor, a node in search of a new friend should only be able to form connections with his friends' friends, so to speak, since other nodes are too far away to know about them.

1.4 Implementation in Python

I heavily use NetworkX [4] in order to build and analyze networks. The basic building blocks of my code are

1. class MapNode: the common skeleton used for all one-dimensional maps

2. OneDimMaps.py: contains information about various maps (logistic, tent, cubic, cusp, exponential, cosine, bit-shift)

3. class NetworkOfMaps.py: uses NetworkX, MapNode and OneDimMaps.py to build graphs of maps

4. AnalyzeNetMapsGlobal.py (similarly for 'Local'): instantiates networks and calls routines to update, analyze and plot them.

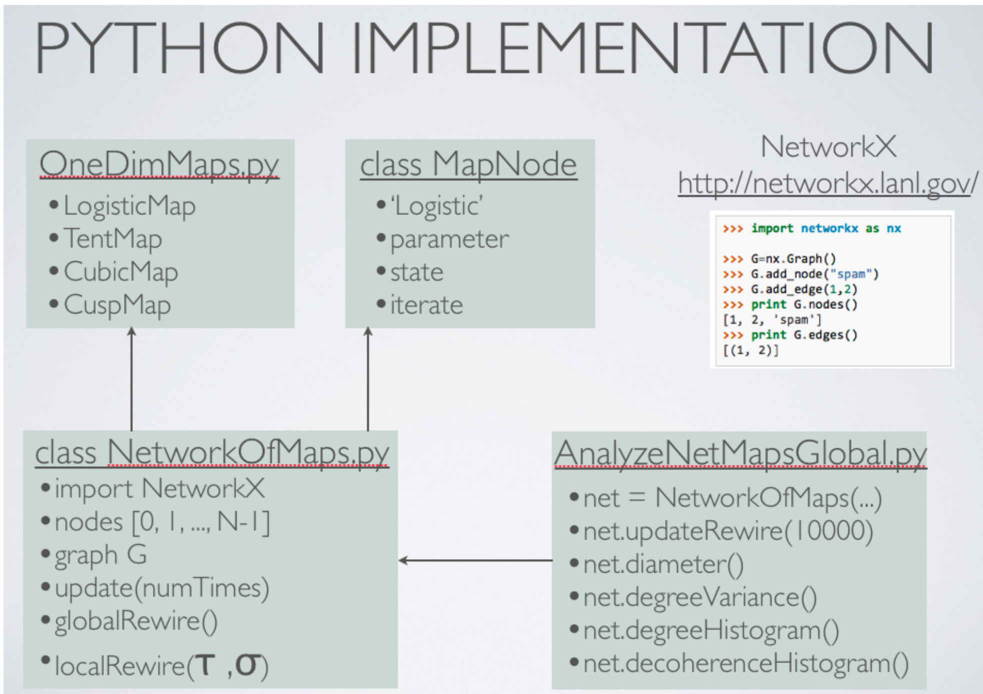


Figure 2: Main classes and Python files for the rewiring algorithm.

1.5 Global Rewiring Results

First I confirmed the results of [2, 3]. I begin with 300 logistic maps with $r = 4$ connected by 5,000 edges, initially chosen at random, and iterate for a transient of 1,000 time steps. For the next 10,000 time steps I record data on the clustering coefficient, degree variance, and the distance between the iterates of two nodes chosen at random. As in found in [2, 3], the clustering coefficient increases from about 0.10 to 0.20, and the degree distribution changes from Poisson distributed (since it's an Erdős-Rènyi random graph) to a skewed distribution with larger variance (50 versus 30).

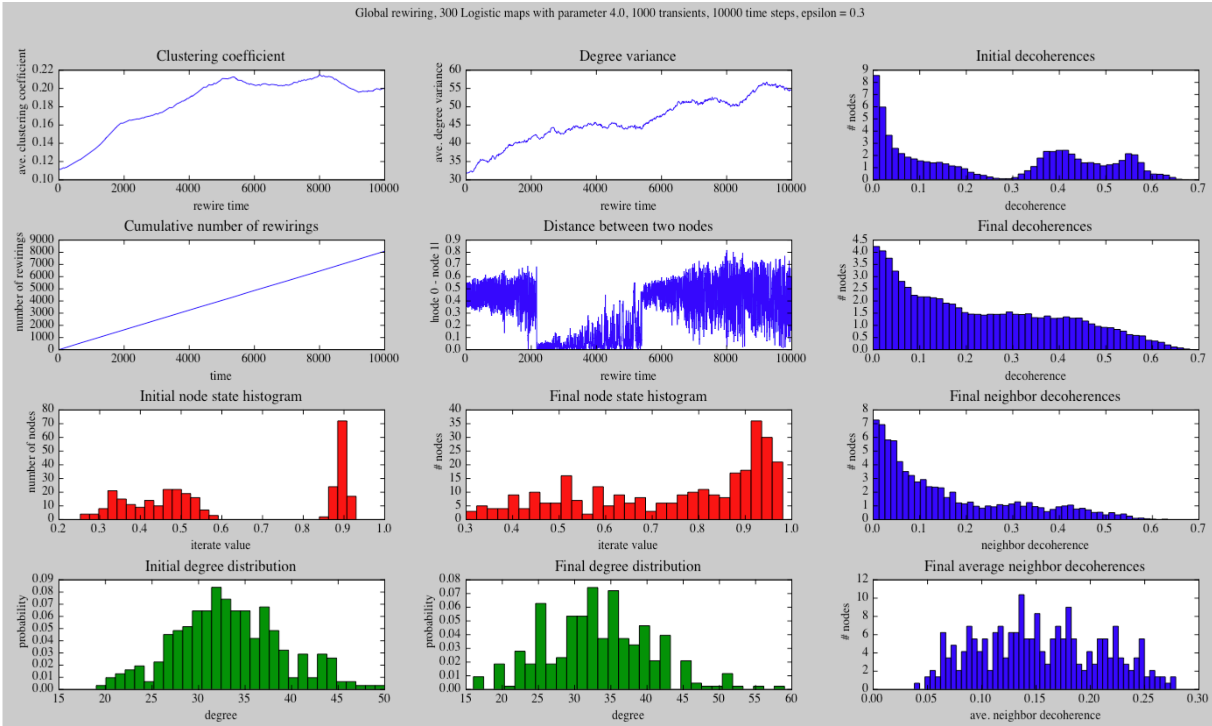


Figure 3: Global Rewiring, $\epsilon = 0.3$, 10,000 time steps.

1.6 Local Rewiring Results

Next I studied the new rewiring algorithm, Local Rewiring, with the same setup as for Global Rewiring (300 logistic maps with $r = 4$ connected by 5,000 edges, initially chosen at random). I found that σ, the maximum path length to which a node looks for a new connection, has little effect on the dynamics, so I fixed $\sigma = 2$. On the other hand, τ, the threshold of decoherence at which nodes rewire, has a strong effect on the dynamics. In a way, τ resembles a driving force: by lowering it, the nodes rewire more frequently, and the network's topology appears to converge more rapidly. The Global Rewiring algorithm, which selects one node at random per time step to rewire, has no analog of τ, a driving force, which may be a benefit for Local Rewiring.

Under Local Rewiring, the network undergoes many rewirings per time step (8 for $\tau = 0.25$, 60 for $\tau = 0.2$). As a result, measures of the network's topology, such as its clustering coefficient and variance of its degree distribution, appear to settle down more rapidly than for Global Rewiring. The final topologies, however, are similar to those of Global Rewiring: skewed degree distributions (degree variance 100 for Local, 200 for Global), high clustering (0.2), and interesting temporal dynamics in the decoherence histogram (see Fig. 7).

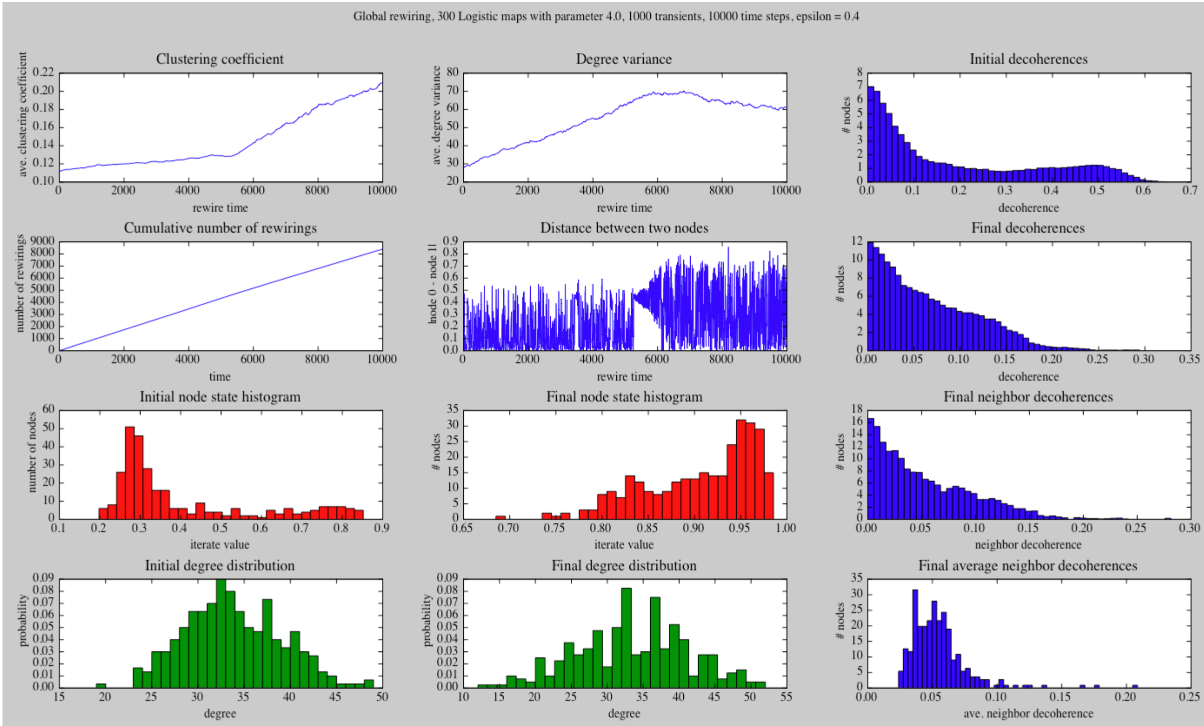


Figure 4: Global Rewiring, $\epsilon = 0.4$, 10,000 time steps. Note the knee in the clustering coefficient time series (top left).

1.7 Two Maps on a Network

In the results above, each node is a logistic map with $r = 4$. What if the network consists of *two* maps (say, logistic and tent)? Under the coupled dynamics and rewiring, would the two maps coalesce into homogeneous communities? Or would they be ambivalent to forming friendships with like and non-like maps?

To help quantify the answer, we define the *homogeneity* of a network of two maps to be the fraction of edges that connect identical maps. With 150 logistic maps

$$f(x) = 4x(1-x) \tag{3}$$

and 150 tent maps (with parameter = 1)

$$g(x) = \begin{cases} x & : 0 \le x \le \frac{1}{2} \\ 1 - x & : \frac{1}{2} < x \le 1 \end{cases}$$

we find that Global Rewiring achieves homogeneity $h \approx 0.9$, while Local Rewiring achieves homogeneity $h \approx 0.6$.

We posit two reasons why the network under Global Rewiring becomes more homogeneous. First, in Global Rewiring nodes can search further in the network—indeed, as

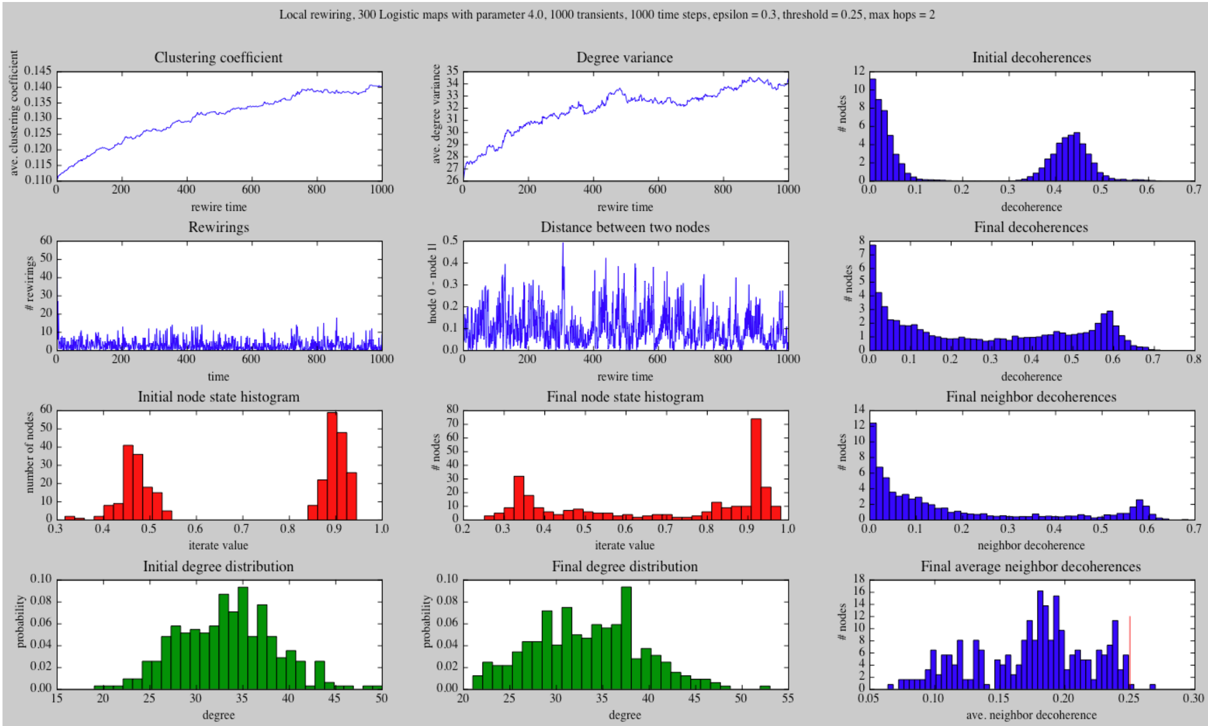


Figure 5: Local Rewiring, threshold $\tau = 0.25$, maximum path length $\sigma = 2$, 1,000 time steps.

far away as they wish—whereas in Local Rewiring nodes can only search at most σ hops away in the network, so they may have to settle for less coherent nodes, which are more likely to be a different map. A second reason that the network under Local Rewiring undergoes many more rewirings, effectively "stirring up" the system with additional energy that breaks homogeneous connections. This suggests that homogeneous connections are a sort of "ground state" of the system and that the network may be amenable to statistical mechanical analysis.

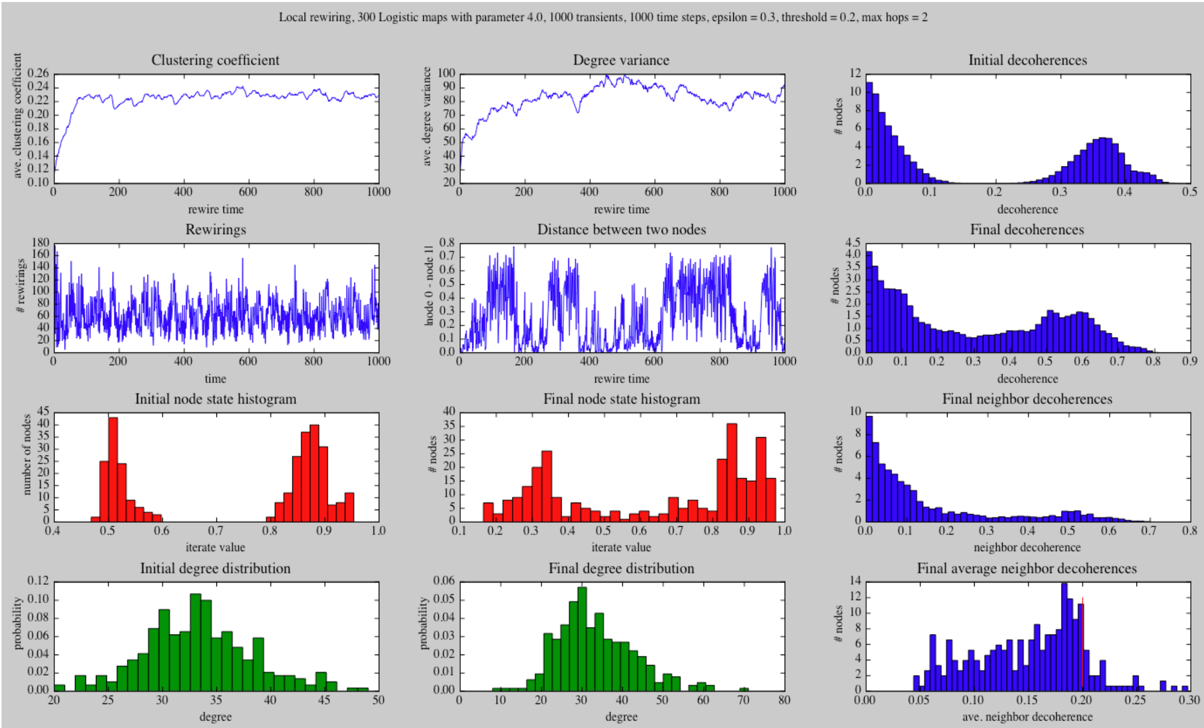


Figure 6: Local Rewiring, threshold $\tau = 0.2$, maximum path length $\sigma = 2$, 1,000 time steps.

2 Part II: Computational Mechanics of Networks of Maps

Our motivating question is the following: if the time series for each node's output is converted to a discrete alphabet (by a generating function, say), can one recover the topology of the network—such as who is connected to whom? We attack this question using two measures of information flow between two systems evolving in time: *time-delayed mutual information* and *transfer entropy*. We report on efforts to derive an analytical expression for the mutual information of two coupled logistic maps, applying causal state filtering to the symbolic time series, and inferring the topology of the network.

To convert time series of node output $x_i(t)$ to a discrete alphabet $\{A, B\}$, we use the generating partition for the logistic map with parameter $r = 4$ given by

$$\sigma_i(t) := \begin{cases} A & : 0 \le x_i(t) \le \frac{1}{2} \\ B & : \frac{1}{2} < x_i(t) \le 1 \end{cases}$$

In this way, a time series of floating point numbers $x_i(t)$ becomes a time series of discrete symbols, with which we can now compute information-theoretic quantities to study the

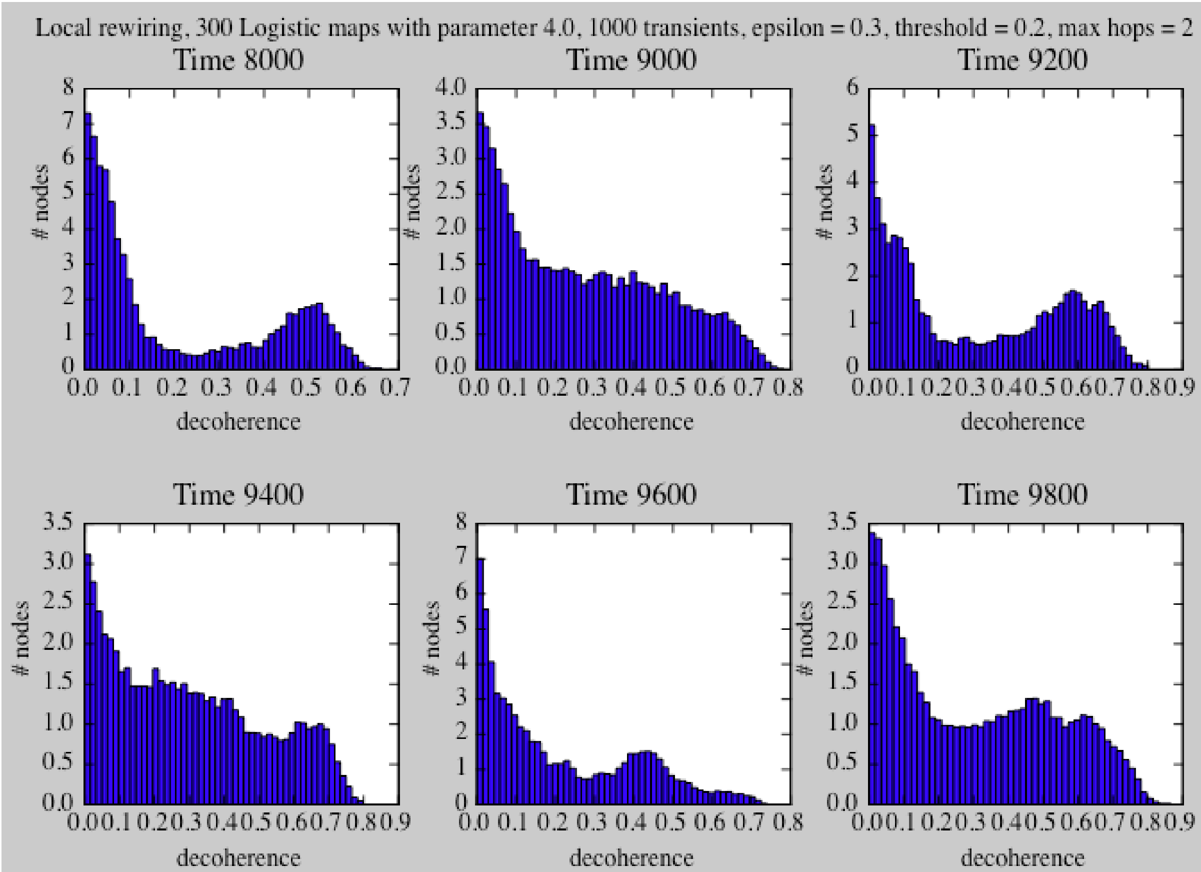


Figure 7: Temporal dynamics in the decoherences $d_{ij}(t) = |x_i(t) - x_j(t)|$ persist even after the network topology has settled down due to Local Rewiring. Plotted here are decoherence histograms at times 8000, 9000, 9200, 9400, 9600. Compare this to Fig. 1 for the Global Rewiring. (Recall from Fig. 6 that the clustering coefficient settles down after a few hundred time steps.)

coupling between nodes.

2.1 Mutual Information of Two Coupled Logistic Maps

Before tackling networks with at least three nodes, how well can we understand a network of two coupled logistic maps? In particular, can we derive an analytical expression for the mutual information $I(\sigma_1(t); \sigma_2(t))$ of the symbols at the same time?

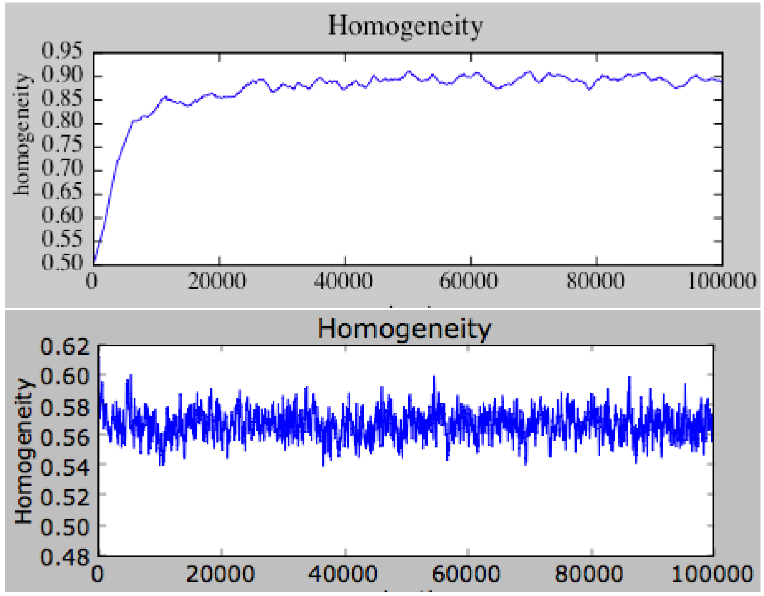


Figure 8: Homogeneity (fraction of edges connecting like maps) for global rewiring (top) and local rewiring (bottom).

For two nodes the dynamics are

$$\begin{aligned} x_1(t+1) &= (1-\epsilon)f(x_1(t)) + \epsilon f(x_2(t)); \\ x_2(t+1) &= (1-\epsilon)f(x_2(t)) + \epsilon f(x_1(t)). \end{aligned}$$

If the dynamics are ergodic, then the asymptotic mutual information $I(\sigma_1(t); \sigma_2(t))$ would be the area of the unit square that gets mapped above $\frac{1}{2}$—weighted by the asymptotic distribution over the unit square (see Fig. 9). However, in general the dynamics are quite complicated, as shown in Fig. 10. This suggests that an analytical expression for the asymptotic distribution over the unit square does not exist in general.

Nevertheless, there is one tractable (actually, trivial) case: $\epsilon = \frac{1}{2}$. The nodes synchronize at time $t = 1$ because of the perfect balance

$$x_1(1) = \frac{f(x_1(1)) + f(x_2(1))}{2} = x_2(1),$$

and hence the nodes are synchronized for all $t \geq 1$. So the system reduces to one logistic map with parameter $r = 4$ but stretched along the diagonal of the unit square. In this case the asymptotic distribution is known, and the (asymptotic) mutual information can be calculated analytically:

$$\begin{aligned} I(\sigma_1, \sigma_2) &= I(\sigma_1, \sigma_1) \\ &= H(\sigma_1) \\ &= 1. \end{aligned}$$

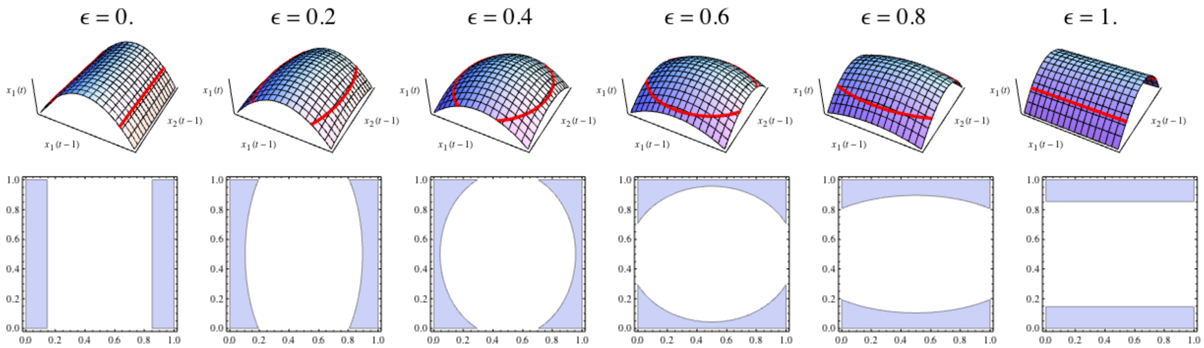


Figure 9: The mutual information between two coupled nodes $I(\sigma_1(t), \sigma_2(t))$ is the area of the unit square, weighted by the asymptotic distribution, that is mapped above $\frac{1}{2}$.

2.2 Transfer Entropy and Time-Delayed Mutual Information

We will use two different measures of information flow between two coupled systems that evolve in time: time-delayed mutual information and transfer entropy.

The time-delayed mutual information between two nodes $1, 2$ is the mutual information of node 1's symbol at time t with node 2's symbol at time $t - \tau$, where τ is the time-delay parameter:

$$M_{1,2}(\tau) := I(\sigma_1(t); \sigma_2(t - \tau)). \tag{4}$$

The transfer entropy, meanwhile, is a conditional mutual information:

$$T_{2\to 1} := I[\sigma_1(t); \sigma_2(t-1), ..., \sigma_2(t-\tau)|\sigma_1(t-1), ..., \sigma_1(t-\tau)], \tag{5}$$

where the parameter τ controls the length of the block of past random variables. These definitions are illustrated in Figures 11 and 12.

In a Physical Review Letter [**?**], Thomas Schreiber argues that transfer entropy is a more useful quantity than time-delayed mutual information because it distinguishes exchanged information from shared information due to common history and input signals. However, I found mixed results, and even concluded that the mutual information without time-delay was most useful for inferring network topology.

2.3 Results for Small Networks

We report on results for the mutual information without time delay, time-delayed mutual information with delay $\tau = 1$, and transfer entropy with block length $\tau = 1$, evaluated for 10^5 or 10^6 time steps (after a transient of $1,000$ time steps). In most cases, the mutual information without time delay is most informative of which nodes are connected to whom.

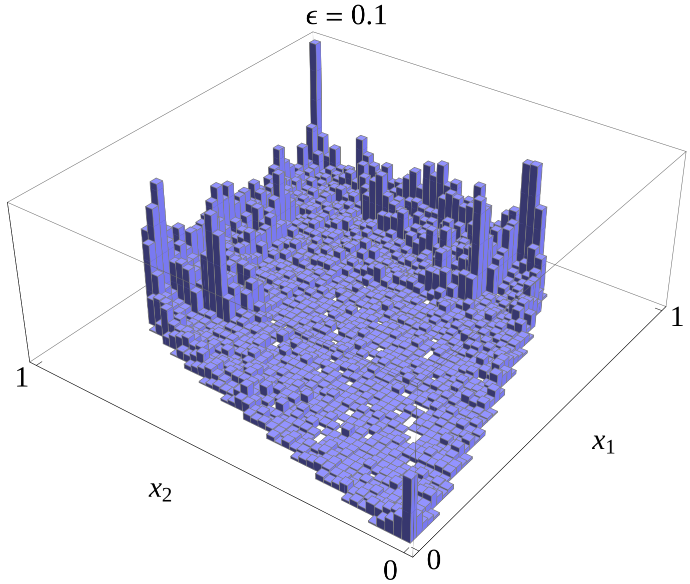


Figure 10: A histogram of 20,000 iterates of two coupled logistic maps with coupling parameter $\epsilon = 0.1$ shows that the dynamics are complicated and likely not ergodic.

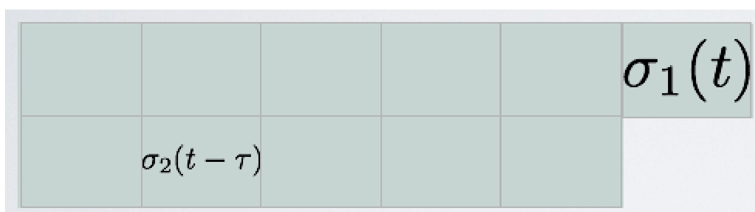


Figure 11: The time-delayed mutual information $M_{1,2}(\tau) := I(\sigma_1(t); \sigma_2(t-\tau))$.

2.3.1 Two Nodes, One Edge

As shown in Fig. 13, for two nodes with balanced coupling $\epsilon = \frac{1}{2}$, the nodes synchronize, and $I(\sigma_1(t), \sigma_2(t))$ converges to 1, while the other measures decay to zero. All three measures vanish when the nodes are uncoupled $\epsilon = 0$. With small coupling ($\epsilon = 0.1, 0.2$), both same-time mutual information and transfer entropy detect the coupling, while time-delayed mutual information only weakly detects it.

2.3.2 Three Nodes, Two Edges

As shown in Fig. 14, just as for the network of two nodes, when $\epsilon = \frac{1}{2}$, the network synchronizes, which same-time mutual information detects ($I(\sigma_1, \sigma_2) = 1$) but time-delayed mutual information and transfer entropy do not (because they condition on

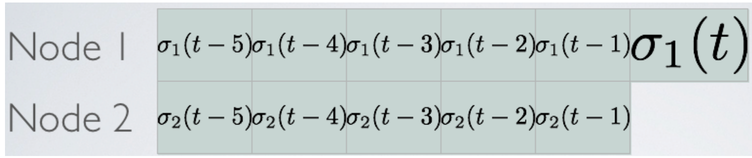


Figure 12: The transfer entropy $T_{2\to 1} := I[\sigma_1(t); \sigma_2(t-1), ..., \sigma_2(t-\tau)|\sigma_1(t-1), ..., \sigma_1(t-\tau)]$.

the past, which apparently fully determines the future in this case). With asymmetric coupling $\epsilon \neq \frac{1}{2}$, the results are mixed: for some ϵ the mutual information is stronger between connected nodes than between unconnected nodes whereas transfer entropy is the opposite, while for other ϵ the reverse holds: transfer entropy is stronger between connected nodes than between unconnected nodes, while mutual information is weaker between connected nodes than between unconnected nodes.

2.3.3 Three Hundred Nodes, Five Thousand Edges

Finally we test these measures on two connected nodes and two unconnected nodes chosen at random from a large network with 300 vertices and 5,000 edges (see Fig. 15). When $\epsilon = 0.5$, the network does not completely synchronize as it does for small networks with 2 or 3 nodes, but the information measures end up being very similar between connected and unconnected nodes, which makes inferring who is connected to whom difficult.

For $\epsilon = 0.2$, we find mixed results: the time-delayed mutual information is weaker between two unconnected nodes, while the transfer entropy is stronger by an order of magnitude between connected nodes than between unconnected ones. This suggests that inferring who is connected to whom via information measures on symbolic output is challenging.

2.4 Causal State Filtering

Before reporting on the grand finale, the results of inferring the topology of a moderately sized network, we comment on efforts to use ϵ-machines to convert symbol time series to time series of *causal states*, which may be more informative for inferring topology. This process, known as *causal state filtering*, consists of four steps:

1. Infer ϵ-machines from nodes' symbolic time series (using the algorithms TreeMerge or StateSplitting)

2. Feed symbolic time series into the ϵ-machines, synchronize to a state, then output the ensuing ϵ-machine states (see Fig. 16)

TWO NODES

- 1,000 transients
- 1,000,000 iterates
- r=4.0

			time-delayed mutual information	transfer entropy
Network	Coupling ϵ	$I[\sigma_1(t);\sigma_2(t)]$	$I[\sigma_1(t);\sigma_2(t-1)]$	$I[\sigma_1(t);\sigma_2(t-1)\vert\sigma_1(t-1)]$
	0.5	1.0	0.0000004	**0**
	0	0.000000001	0.0000001	0.0000001
	0.1	**0.005**	0.005	**0.05**
	0.2	**0.38**	0.04	**0.10**

Figure 13: Mutual information, time-delayed mutual information, and transfer entropy for two nodes and various coupling strengths ϵ.

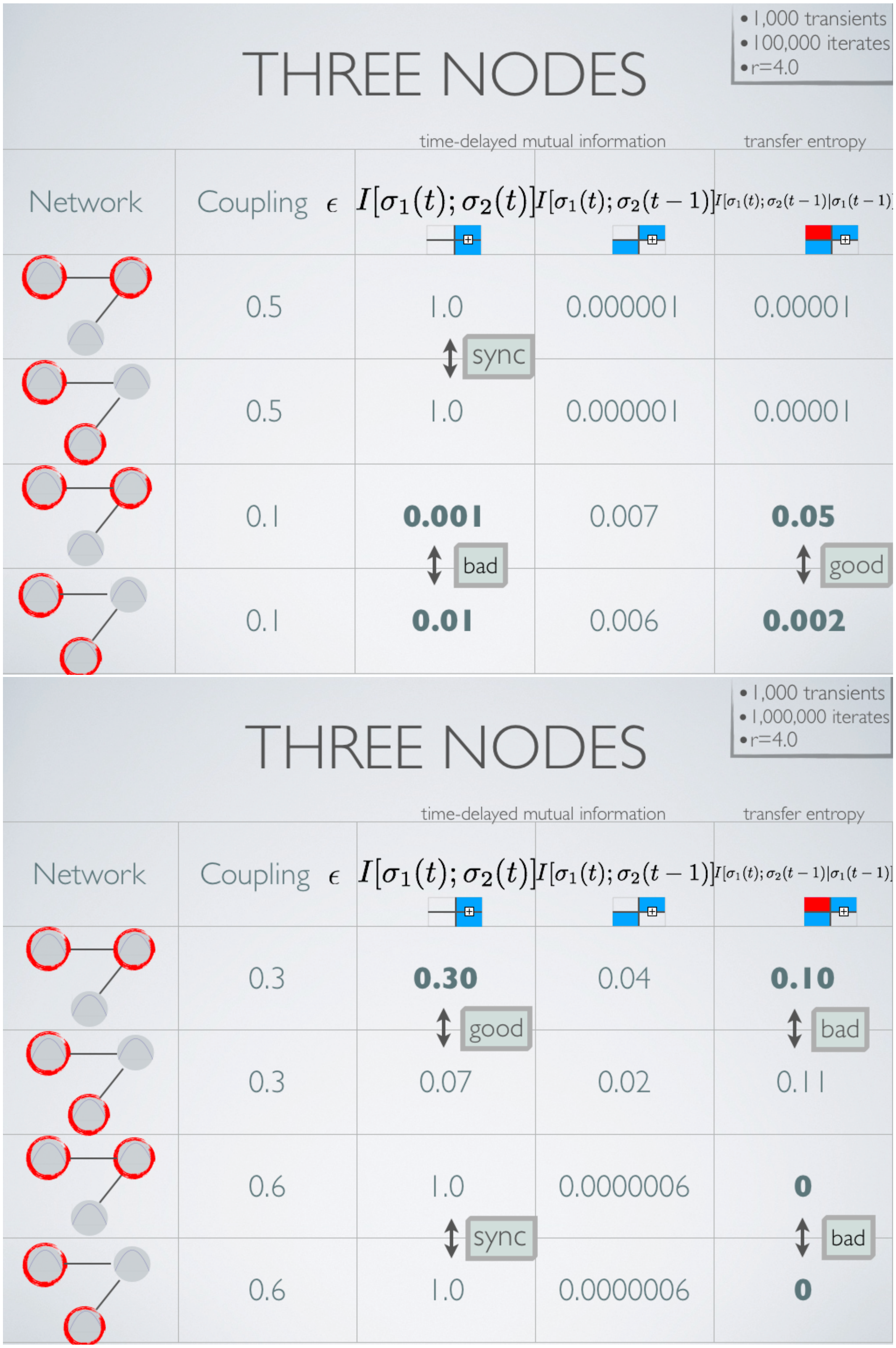
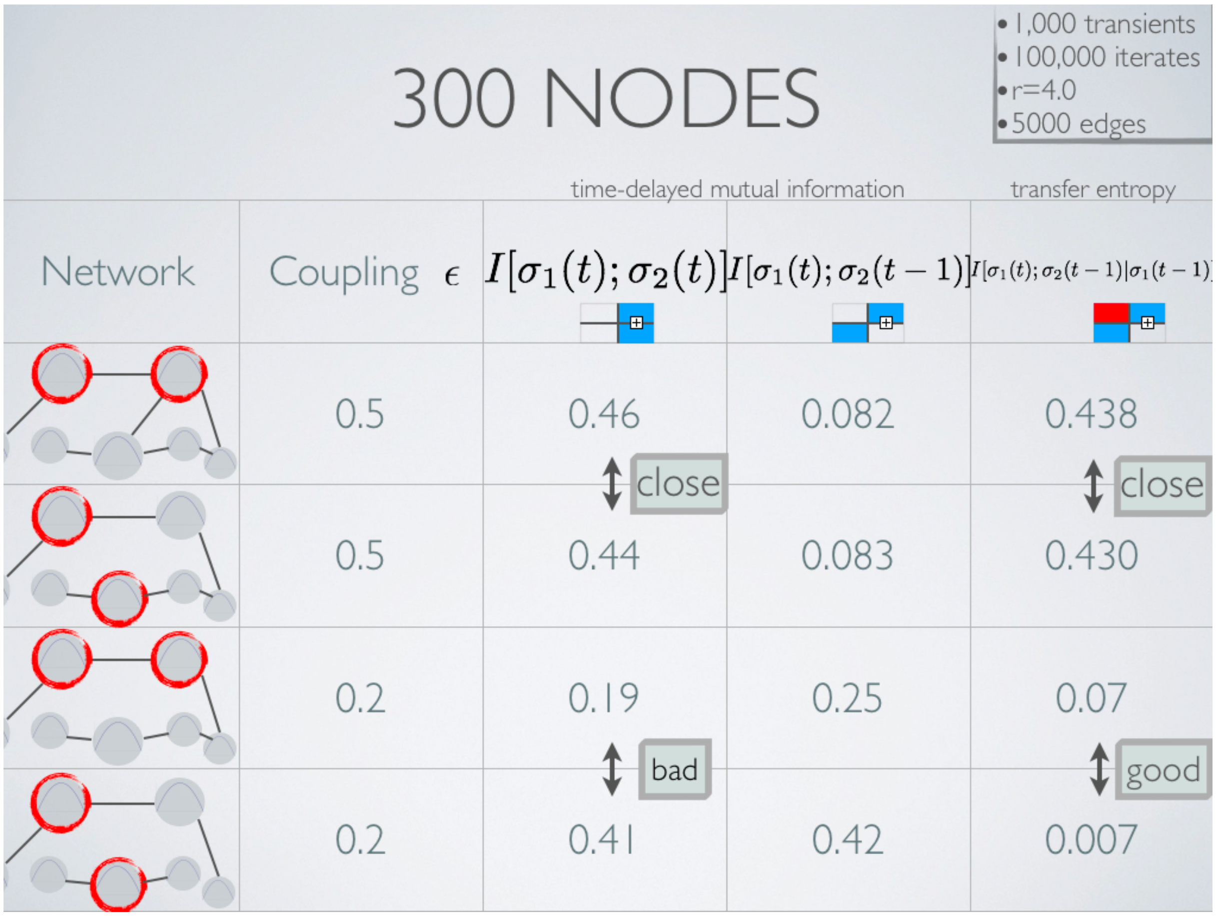


Figure 14: Mutual information, time-delayed mutual information, and transfer entropy for three nodes and two edges for various coupling strengths ϵ.



Network	Coupling ϵ	$I[\sigma_1(t);\sigma_2(t)]$	$I[\sigma_1(t);\sigma_2(t-1)]$	$I[\sigma_1(t);\sigma_2(t-1)\mid\sigma_1(t-1)]$
	0.5	0.46	0.082	0.438
	0.5	0.44	0.083	0.430
	0.2	0.19	0.25	0.07
	0.2	0.41	0.42	0.007

Figure 15: Mutual information, time-delayed mutual information, and transfer entropy for 300 nodes and 5000 edges for various coupling strengths ϵ.

3. Compute mutual information and transfer entropy on the time series of ϵ-machine states

We encountered difficulties with implementing this. With both inference algorithms, TreeMerge and StateSplitting, we invariably feed *forbidden words* into the ϵ-machine as we try to convert symbols to causal states. To fix this we tried resetting to the asymptotic distribution over causal states whenever a forbidden word is encountered; however, this makes it difficult to compare two time series, which must be truncated after each forbidden word. More careful attention to the parameters for the inference algorithms may solve these problems.

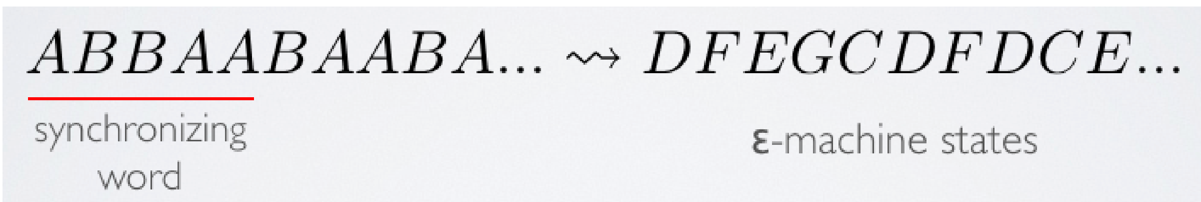


Figure 16: Step three of causal state filtering: after inferring an ϵ-machine from the symbols (with causal states labeled by C, D, E, F, G, in this example), feed the symbols into the machine to synchronize to a causal state, then use the ensuing time series of causal states.

2.5 Inferring the Topology of a Network of Maps

Since the mutual information without time delay, $I(\sigma_1(t), \sigma_2(t))$, appeared to be most useful for detecting who is connected to whom, as found in Section 2.3, we use this measure to infer the topology of a network of five nodes connected by five edges. As shown in Fig. 17, the mutual information is stronger between connected nodes (61, 65, 71, 72, 91, units: 10^{-3} bits) than between unconnected nodes (36, 51, 51, 53). However, the gap is small.

2.6 Conclusions and Future Work

Inferring the topology from a network of chaotic maps by computing information measures on the nodes' symbolic output is difficult, but it has hope. We found that the mutual information without time delay $I(\sigma_1(t), \sigma_2(t))$ appears to be more useful than time-delay mutual information and transfer entropy with $\tau = 1$. Moreover, the mutual information is stronger between connected nodes than between unconnected nodes, and the difference may become more pronounced if averaged over more time steps.

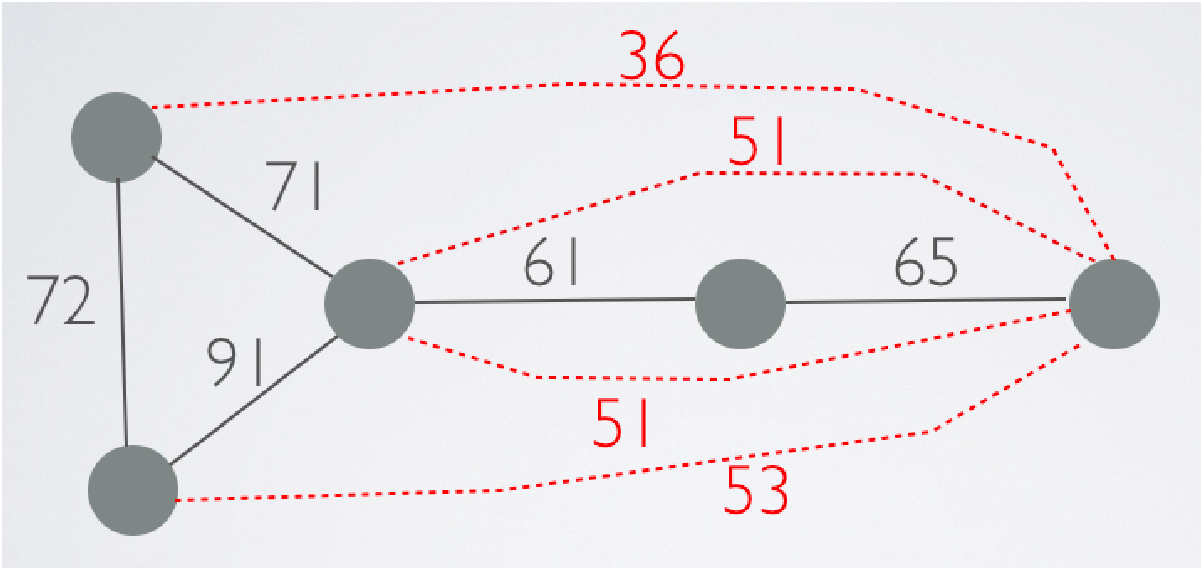


Figure 17: Inferring network topology: the edge labels are mutual information without time delay $I(\sigma_1(t), \sigma_2(t))$ ($\times 10^{-3}$ bits) computed for every pair of nodes in a network of five nodes and five edges. The mutual information is stronger between connected nodes than between unconnected nodes, but not by much.

In the future, rather than using generic information measures such as mutual information and transfer entropy, there may exist information measures more naturally tailored toward graphs—that incorporate, for instance, degree or centrality in the network—which may prove to be more useful. But before abandoning mutual information and transfer entropy, it is worth exploring whether using a longer time delay and more history than $\tau = 1$, as used here, may allow one to determine path lengths between nodes in a network. Furthermore, instead of aiming to infer who is connected to whom, a rather ambitious goal, other characteristics of the network may be more tractable, such as the degree of nodes or the presence of motifs (common structural patterns) in the network.

Of course, a whole zoo of other chaotic creatures could be placed on the nodes of a network, including other one-dimensional maps or even ordinary differential equations. In fact, chaotic ODEs may be more amenable to informational measures of statistical coherence because they do not jump around like discrete maps do.

A pie-in-the-sky, ultra-ambitious application of this work may be inferring the network topology of chaotic, coupled dynamical systems, such as neurons in the brain. In many such applications, the links would be dynamic, thus requiring a synthesis of rewiring models and information-theoretic measures such as those studied here.

References

[1] A.L. Barabási, R. Albert. Emergence of Scaling in Random Networks. Science, Vol. 286, No. 5439, October 1999, pp. 509 - 512

[2] P. Gong and C. Van Leeuwen. Evolution to a small-world network with chaotic units. Europhys. Lett., 67 (2), pp. 328-333 (2004)

[3] D. van den Berg and C. van Leeuwen. Adaptive rewiring in chaotic networks renders small-world connectivity with consistent clusters. Europhys. Lett., 65 (4), pp. 459464 (2004)

[4] NetworkX, Version 1.1, http://networkx.lanl.gov/

[5] R. Pashaie and N. H. Farhat. Self-Organization in a Parametrically Coupled Logistic Map Network: A Model for Information Processing in the Visual Cortex. IEEE Transactions on Neural Networks, Vol. 20, No. 4, April 2009

[6] T. Schreiber. Measuring Information Transfer. Physical Review Letters, Vol. 85, No. 2, July 2000.