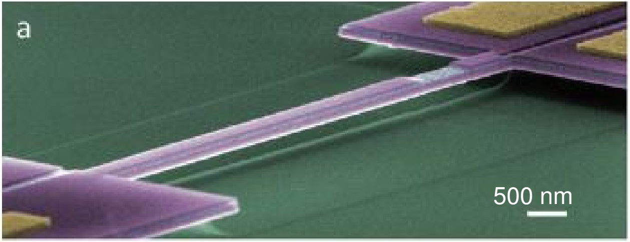
Permutation Entropies and Chaos

Russell Hawkins

Complexity Sciences Center
Physics Department
University of California, Davis

June 2nd, 2015

Nanoelectromechanical Systems

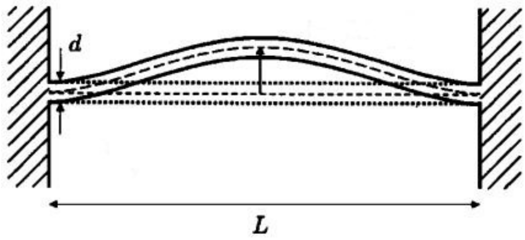


Roukes group Nano Letters 2011

- Sub-micron scale resonators
- Mechanical behavior electrically measure/controlled
- Subject to thermal fluctuations

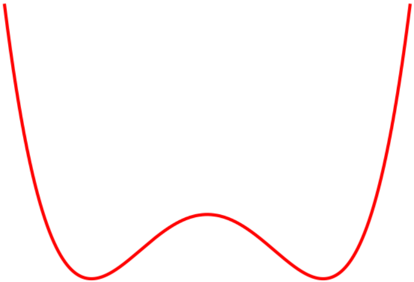
We want to make Maxwell Demons with them!

Nanobeams = Duffing Oscillators



- Compress a beam, eventually buckles up/down
- Can model bistable beam dynamics with Duffing equation

Duffing Equation



$$\ddot{x} + \gamma\dot{x} - \alpha x + \beta x^3 = G\cos\omega t$$

- Interested in mapping behavior over parameter space
- Also want to characterize behavior of experimental system
- Exploring utility of **Permutation Entropy**

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

- Doesn't need description of process

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

- Doesn't need description of process
- Simple algorithm

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

- Doesn't need description of process
- Simple algorithm
- Fast

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

- Doesn't need description of process
- Simple algorithm
- Fast
- Robust to noise

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

- Doesn’t need description of process
- Simple algorithm
- Fast
- Robust to noise
- Can get Shannon entropy rate of process in infinite limit

Permutation Entropy

Measure chaos directly from data, without equations of motion?

Permutation Entropy:

- Doesn't need description of process
- Simple algorithm
- Fast
- Robust to noise
- Can get Shannon entropy rate of process in infinite limit
- Data requirement grows quickly

Algorithm

- Look at length n window

Algorithm

- Look at length n window
- Rearrange elements in increasing order

Algorithm

- Look at length *n* window
- Rearrange elements in increasing order
- Record permutation of elements

Algorithm

- Look at length *n* window
- Rearrange elements in increasing order
- Record permutation of elements
- Repeat for all length n windows in time series

Algorithm

- Look at length *n* window
- Rearrange elements in increasing order
- Record permutation of elements
- Repeat for all length n windows in time series

PE = Shannon entropy of distribution over permutations:

$$H_n^* = -\sum_{\pi} p(\pi) \log_2 p(\pi)$$

where π = set of all n! permutations

A Simple Example

$$\mathbf{x} = (5, 8, 10, 11, 7)$$

Windows of length n = 3:

A Simple Example

$$\mathbf{x} = (5, 8, 10, 11, 7)$$

Windows of length n = 3:

- $(5, 8, 10)$: $x_1 < x_2 < x_3$, permutation is 123

A Simple Example

$$\mathbf{x} = (5, 8, 10, 11, 7)$$

Windows of length n = 3:

- $(5, 8, 10)$: $x_1 < x_2 < x_3$, permutation is 123
- $(8, 10, 11)$: $x_1 < x_2 < x_3$, permutation is 123

A Simple Example

$$\mathbf{x} = (5, 8, 10, 11, 7)$$

Windows of length n = 3:

- $(5, 8, 10)$: $x_1 < x_2 < x_3$, permutation is 123
- $(8, 10, 11)$: $x_1 < x_2 < x_3$, permutation is 123
- $(10, 11, 7)$: $x_3 < x_1 < x_2$, permutation is 312

A Simple Example

$$\mathbf{x} = (5, 8, 10, 11, 7)$$

Windows of length n = 3:

- $(5, 8, 10)$: $x_1 < x_2 < x_3$, permutation is 123
- $(8, 10, 11)$: $x_1 < x_2 < x_3$, permutation is 123
- $(10, 11, 7)$: $x_3 < x_1 < x_2$, permutation is 312

$$H_3^* = -2/3 \log_2 2/3 - 1/3 \log_2 1/3$$

$$H_3^* \approx 0.92 \text{ bits}$$

Permutations Entropy Rate

Define

$$h^*_\infty = \lim_{n\to\infty} \frac{H^*_n}{n-1}$$

For a broad class of systems,

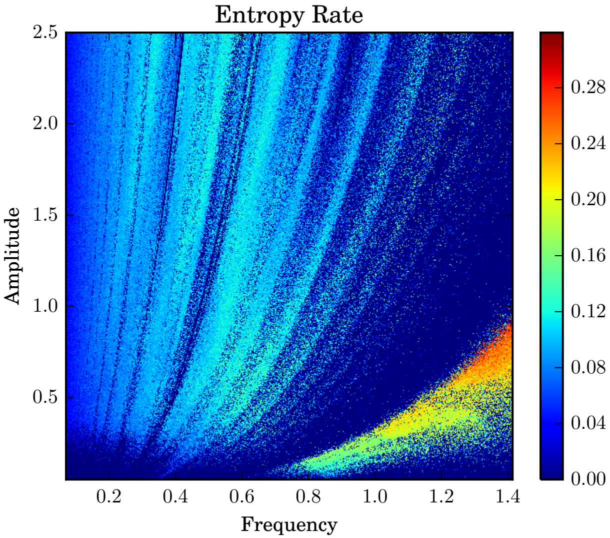
$$h^*_\infty = h_\mu$$

- Permutations partition time series by relative values of neighbors
- Derives partitions from the data itself
- Side steps issue of finding generating partition

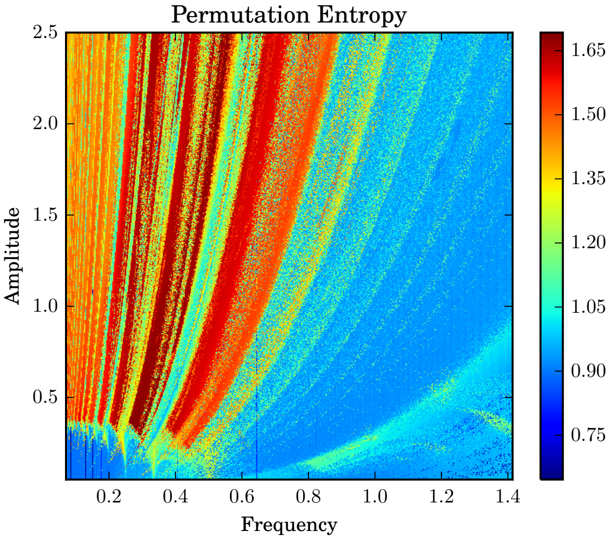
Duffing Oscillator Permutation Entropy Rate

- Focus on 1 degree of freedom (position)
- Sample continuous time data with at least twice Nyquist cutoff frequency
- Compare with Kolmogorov-Sinai Entropy (largest Lyapunov exponent)

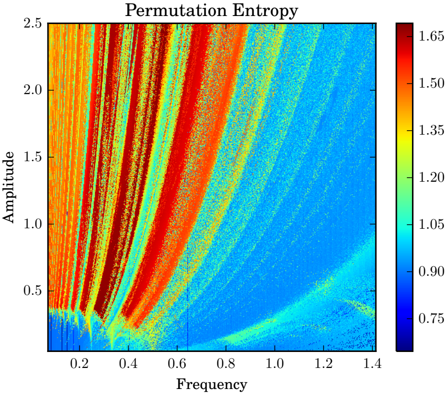
Duffing Oscillator Entropy Rate



Duffing Oscillator Permutation Entropy Rate



Permutation Entropy Rate vs KS Entropy Rate



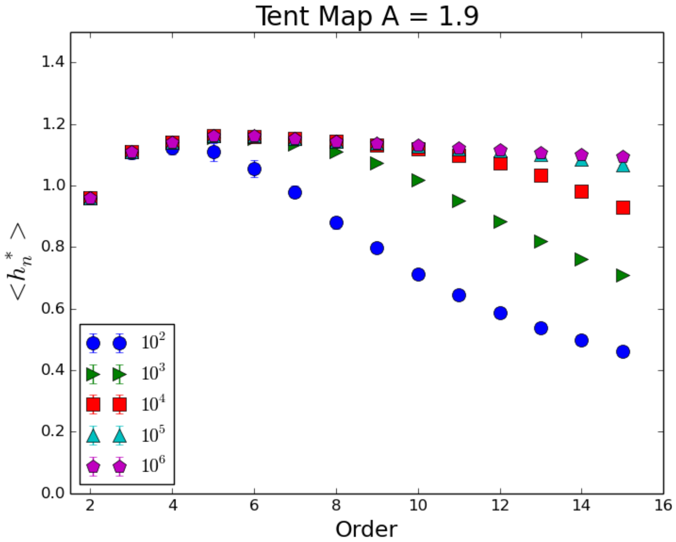
Permutation Entropy for 1-D Maps

To get a better grip on convergence properties, look at simpler systems:

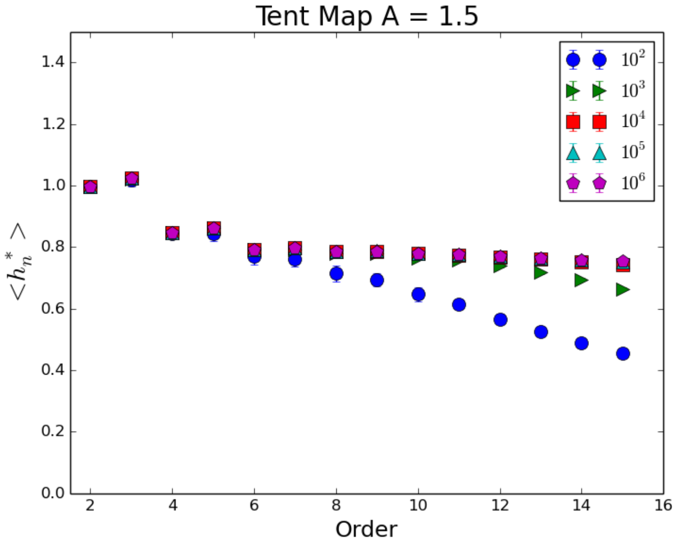
- Tent Map
- Logistic Map

Don't need to worry about sampling frequency issue, h_μ much easier to calculate.

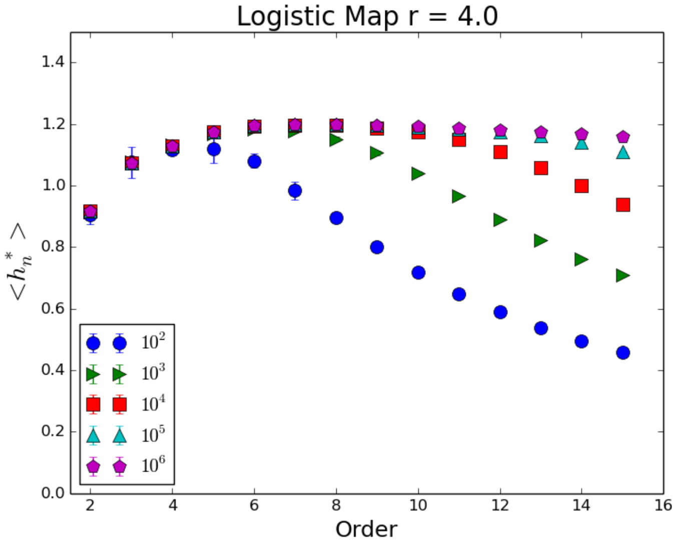
Tent Map h_n^* Convergence



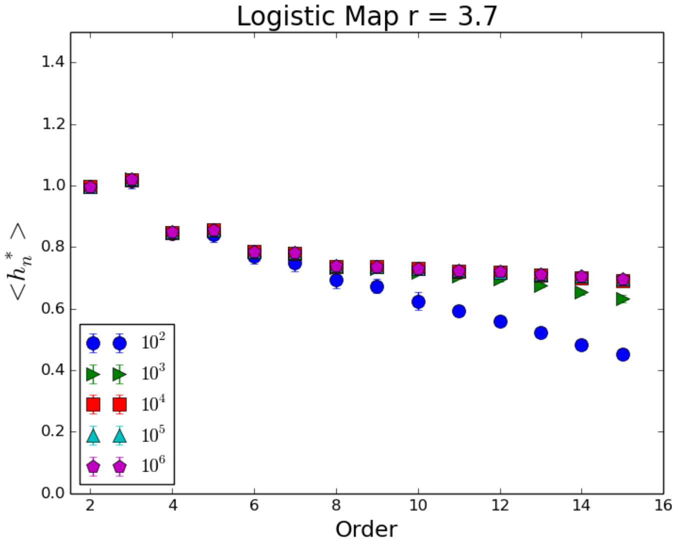
Tent Map h_n^* Convergence



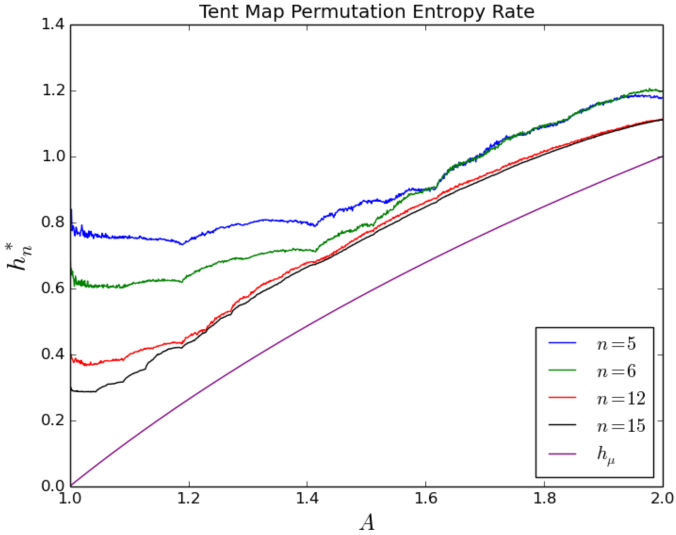
Logistic Map h_n^* Convergence



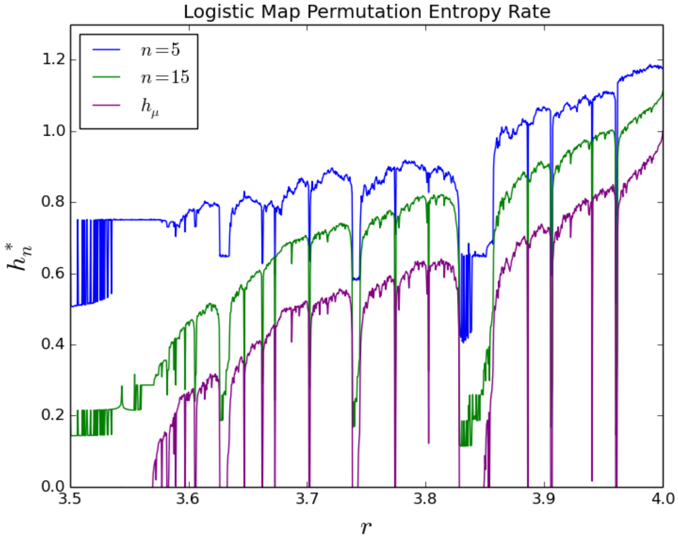
Logistic Map h_n^* Convergence



Tent Map h_n^* vs h_μ



Logistic Map h_n^* vs h_μ



Permutation Excess Entropy

Defined analogously to standard Excess Entropy:

$$E^* \equiv \lim_{n \to \infty} (H_n^* - h_\infty^* \cdot n)$$

Permutation Excess Entropy

Defined analogously to standard Excess Entropy:

$$E^* \equiv \lim_{n\to\infty} (H_n^* - h_\infty^* \cdot n)$$

- Converges to E for same class of processes for which $h_\infty^* = h_\mu$

Permutation Excess Entropy

Defined analogously to standard Excess Entropy:

$$E^* \equiv \lim_{n\to\infty} (H_n^* - h_\infty^* \cdot n)$$

- Converges to E for same class of processes for which $h_\infty^* = h_\mu$

Length n approximate:

$$E_n^* \equiv H_n^* - h_\mu \cdot n$$

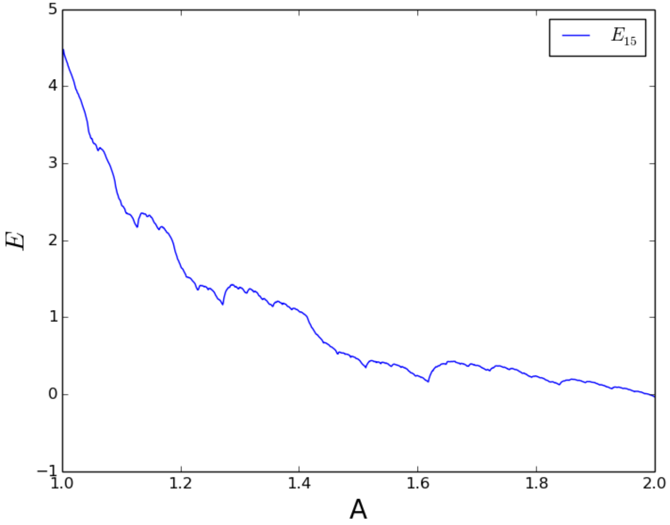
Wrinkle for 1-D Maps

1-D maps don't have a unique Excess Entropy!

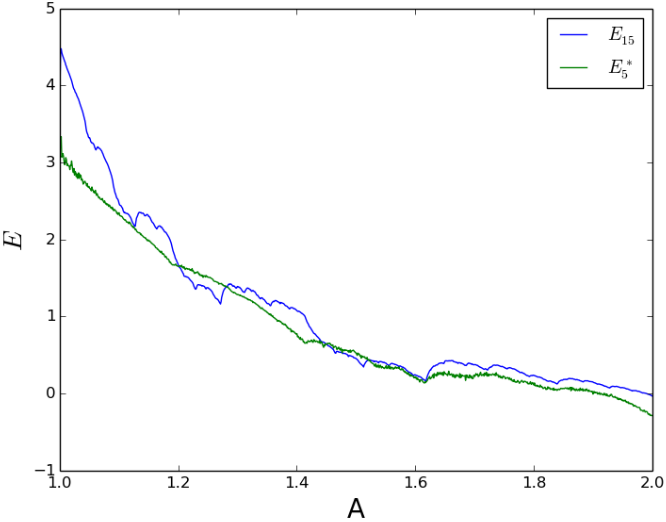
- Different generating partitions give different EE values
- All generating partitions converge to same h_μ, but may differ in how they converge

Which Excess Entropy does E^* converge to for 1-D maps?

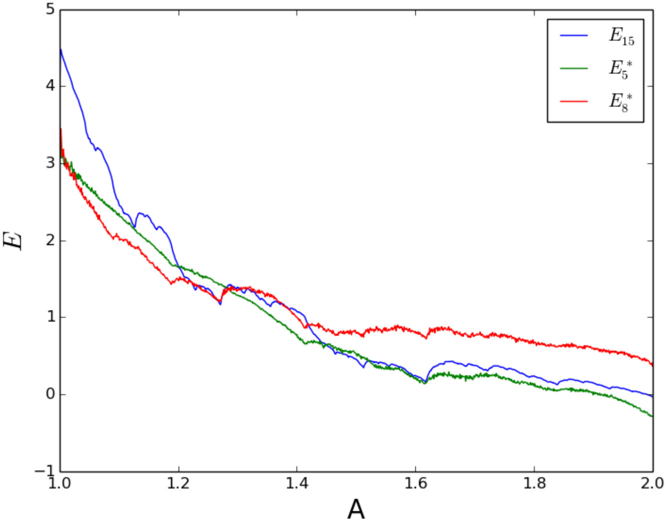
Tent Map Binary Excess Entropy



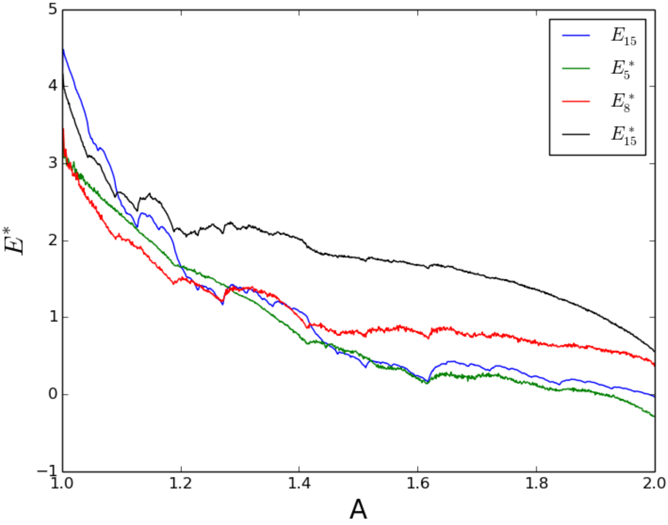
Tent Map E*



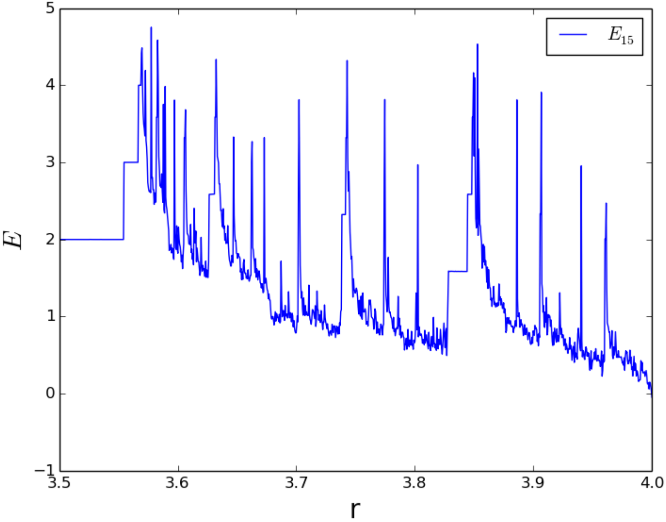
Tent Map E*



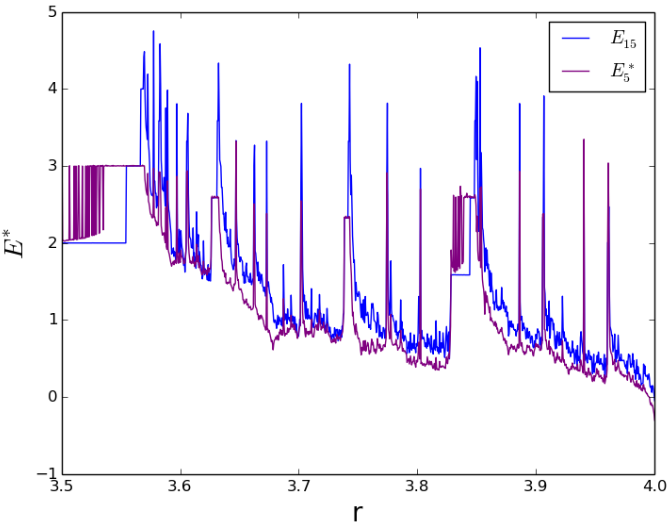
Tent Map E*



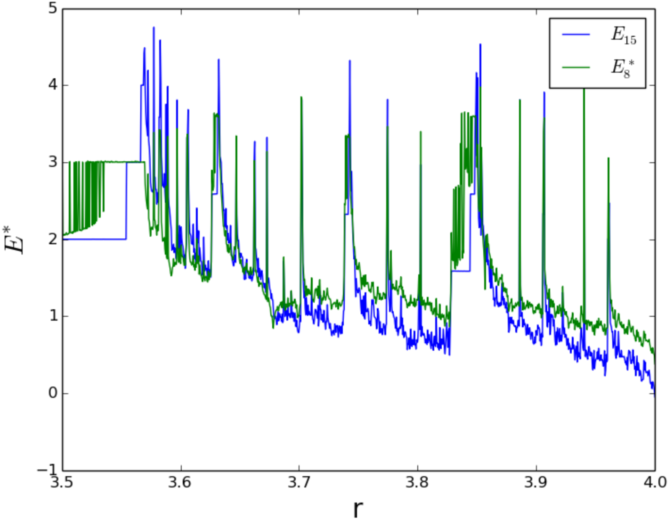
Logistic Map Binary Excess Entropy



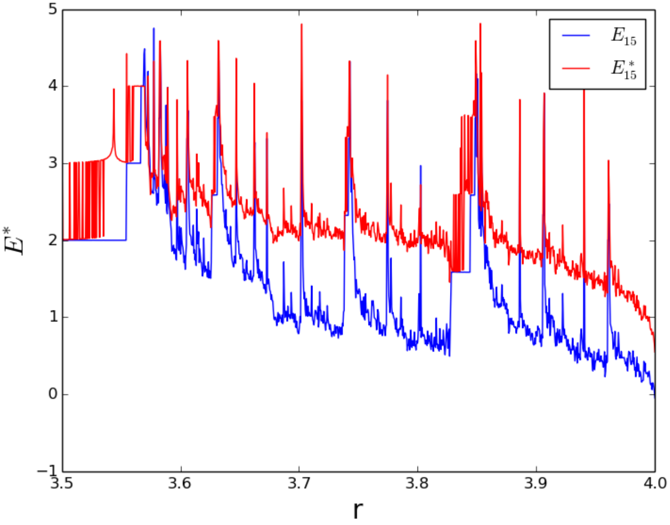
Logistic Map E*



Logistic Map E^*



Logistic Map E^*



Conclusions

Permutation Entropy is cool!

- Definitely captures qualitatively chaos/order in continuous time systems
- Wants to converge to h_μ for 1-D maps
- Takes its time in doing so

Permutation Excess Entropy is cool!

- Tantalizing, raises more questions than it answers
- Which Excess Entropy does it converge too?
- Reflects what kind of partition permutations are