Discovering and Describing Patterns and Organization: An Introduction to Computational Mechanics

David Feldman
University of California, Davis

11 June 1998

with:
Jim Crutchfield (Santa Fe Institute)

thanks to:
Karl Young (UCSF),
Richard T. Scalettar (UCD)

dpf@santafe.edu

http://leopard.ucdavis.edu/dave/

http://www.santafe.edu/projects/CompMech/

Motivation:

1. What are patterns and how can we discover them?
2. What does it mean to say a system is organized?

Central Idea:

Use computation theory to determine **how a system computes**.

Outline:

1. **Introduction to Information Theory.**
2. **Excess Entropy:** A measure of memory.
3. **Introduction to Computation Theory.**
4. ϵ**-Machines:** Using computation theory to describe patterns.
5. **Results for Ising Systems.**
6. **Conclusion.**

Information Theory

- Shannon, *Bell Sys. Tech. J.* 27:379-423. (1948).
- Shannon and Weaver, *The Mathematical Theory of Communication*, Univ. of Illinois Press, (1963).
- Cover and Thomas, *Elements of Information Theory*, Wiley, 1991.

The **Shannon entropy** of a random variable X:

$$H[X] \equiv -\sum_x \Pr(x) \log_2 \Pr(x) \ .$$

$H[X]$ measures the uncertainty associated with the variable X.

The **Mutual Information** between X and Y is defined as:

$$I[X;Y] \equiv H[X] - H[X|Y] \ .$$

Excess Entropy

Consider a sequence of discrete, random variables:

$$\overbrace{\cdots\ S_{-3}\, S_{-2}\, S_{-1}}^{\overleftarrow{S}}\ \overbrace{S_0\, S_1\, S_2\, S_3\ \cdots}^{\overrightarrow{S}}$$

We assume the distribution over sequences is stationary.

Examples:

1. A symbolic dynamical system.
2. A one-dimensional statistical mechanical system
3. A long time series measurements.

The **excess entropy** $\mathbf{E}$ is the mutual information between $\overrightarrow{S}$ and $\overleftarrow{S}$.

$$\mathbf{E} \equiv I[\overleftarrow{S};\overrightarrow{S}]\ .$$

- $\mathbf{E}$ is a measure of memory.
- $\mathbf{E}$ has units of *bits*.

Computation Theory

Idea: Classify sequences by considering how hard it is to recognize them.

Example 1: Accept all sequences of 1's and 0's except for those which have two or more 0's in a row.

$$1110101101$$

$$1101101001$$

This example is pretty easy. You don't need to remember much of the sequence as you scan it to figure out if it's ok.

Example 2: Accept all sequences with balanced parenthesis.

(( ) ( ) ( )) (( ))

(( ) (( ) ( ) ( ) ( )

This example is harder.

Example 3: Accept all sequences which consist of any number of 1's, and only a **prime** number of consecutive 0's!

$$1100011001$$

$$11000011$$

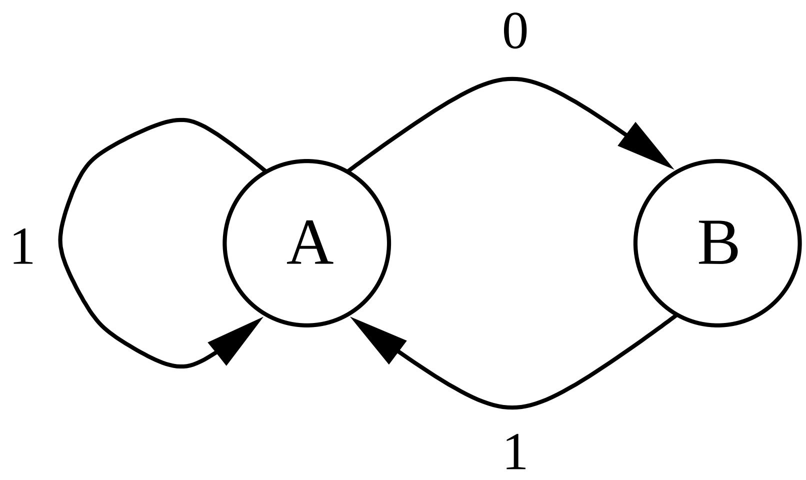
$$11010000000000001$$

$$11 \quad \overbrace{00\cdots00}^{1031 \text{ elements}} \quad 11$$

We can make this notion of difficulty precise by considering the type of "machine" you would have to build to perform this recognition task.

Example 1: Accept all sequences of 1's and 0's except for those which have two or more 0's in a row.

$$1101101001$$



Sequence is OK if there exists a path through this machine as one scans left to right. Machine works for an arbitrarily long sequence.

Example 2: Accept all sequences with balanced

parenthesis.

$$(\,(\,)\,(\,(\,)\,(\,)\,(\,)\,(\,)$$

To check the sequences you need to build a device with memory: a **Stack**.

Put a plate on the stack if you see a left paren $($ and take a plate off the stack if you see a right paren $)$.

If there are no plates left on the stack at the end of scanning the sequence, then sequence is ok.

Example 3: Accept all sequences which consist on any

number of 1's, and only a **prime** number of consecutive 0's.

$$11 \quad \overbrace{00\cdots00}^{1031\ \text{elements}} \quad 11$$

For this task you'll need to build a machine that can perform division: desktop computer or a calculator.

Adeleman, Pomerance and Rumely (1983) found an algorithm to test for the primeness of n whose run-time scales as:

$$\mathcal{O}((\log n)^{c \log\log\log n})$$

Summary of Computation Theory

Classify sequences by considering the type of "machine" needed to recognize them.

The machine reveals the "architecture" of the information processing.

Question:

How can we use use computation theory to discover and quantify patterns?

Consider a sequence of discrete, random variables:

$$\overbrace{\cdots\; S_{-3}\; S_{-2}\; S_{-1}}^{\overleftarrow{S}}\; \overbrace{S_0\; S_1\; S_2\; S_3\; \cdots}^{\overrightarrow{S}}$$

Causal States

- How much of the left half $\overleftarrow{S}$ is needed to predict the right half $\overrightarrow{S}$?
- Only need to distinguish between $\overleftarrow{S}$'s that give rise to different states of knowledge about $\overrightarrow{S}$.
- Two $\overleftarrow{S}$'s that give rise to the same state of knowledge are equivalent:

$$\overleftarrow{S}_i \sim \overleftarrow{S}_j \text{ iff } \Pr(\overrightarrow{S} \mid \overleftarrow{S}_i) = \Pr(\overrightarrow{S} \mid \overleftarrow{S}_j) \ .$$

- Equivalence classes induced by $\sim$ are **Causal States**, minimal sets of aggregate variables necessary for optimal prediction of $\overrightarrow{S}$.

ϵ-Machines

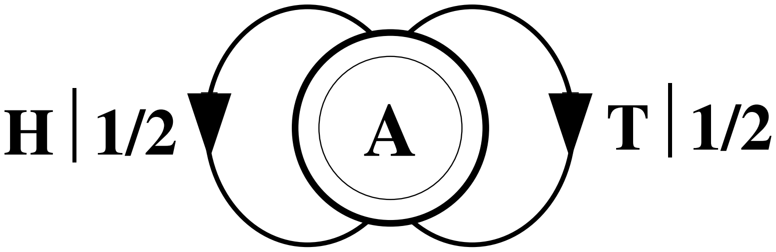
- The causal states together with the probability of transitions between causal states are an ϵ**-machine**, a minimal model capable of statistically reproducing the original configuration.
- Model is chosen from the least powerful computational model class that admits a finite description.
- The ϵ-machine tells us **how the system computes**.
- **Important Technical Note:** Unlike the previous examples, the ϵ-machines are models of **stochastic** computation.

References:

- Crutchfield and Young, Phys. Rev. Lett., **63** (1989).
- Crutchfield, Physica, **D 75** (1994).

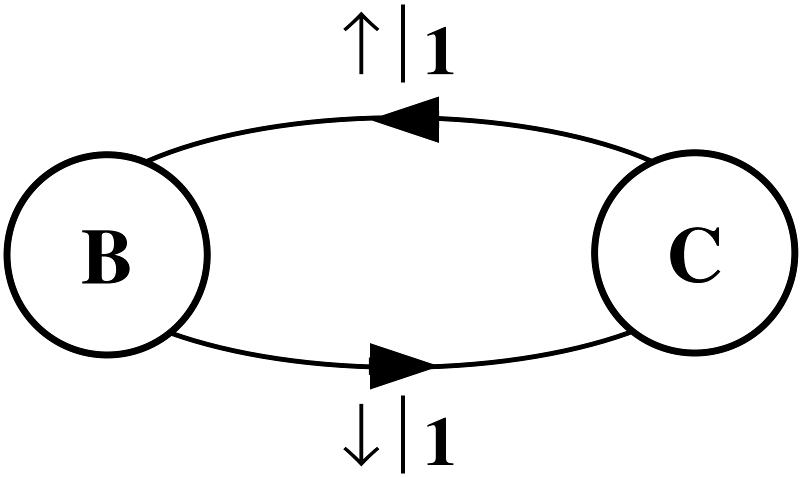
Examples

1. Fair Coin:



$\cdots$ HHTHTHTTTHTHTHTTHTHH $\cdots$

2. Period 2 Pattern:



$\cdots \uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow \cdots$

Examples

For the two examples just considered, the ϵ-machines had a finite number of states.

However, ϵ-machines are not always finite state machines.

- Crutchfield and Young, in *Complexity, Entropy and the Physics of Information*, Addison-Wesley, 1990.
- Crutchfield, *Physica D*, 75:11-54, 1994.
- Upper, *PhD Thesis*, University of California, Berkeley, 1997.

Ising Model Results

- Crutchfield and Feldman, Phys. Rev. E., **55** (1997).
- Feldman and Crutchfield, J. Stat. Phys, (Submitted).

A One-dimensional Ising system with nearest and next nearest neighbor interactions:

$$\mathcal{H} \equiv -J_1 \sum_i S_i S_{i+1} - J_2 \sum_i S_i S_{i+2} - B \sum S_i \ .$$

This gives rise to a Boltzmann distribution:

$$\Pr(\mathcal{C}) \propto e^{-\frac{1}{T}\mathcal{H}(\mathcal{C})} \ .$$

Structure Factors

The **structure factor** $S(q)$ is the Fourier transform of the correlation function $\Gamma(r)$:

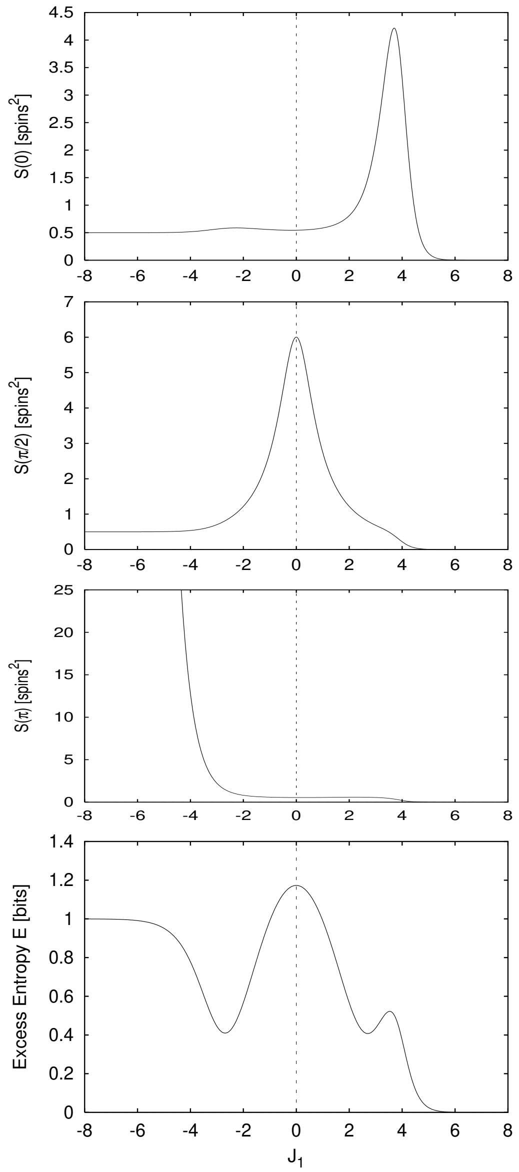
$$\Gamma(r) \equiv \langle (S_0 - \langle S_0 \rangle)(S_r - \langle S_R \rangle) \rangle \ ,$$

and

$$S(q) \equiv \sum_r e^{iqr} \Gamma(r) \ .$$

$S(q)$ **detects correlations of period** $2\pi/q$**.**

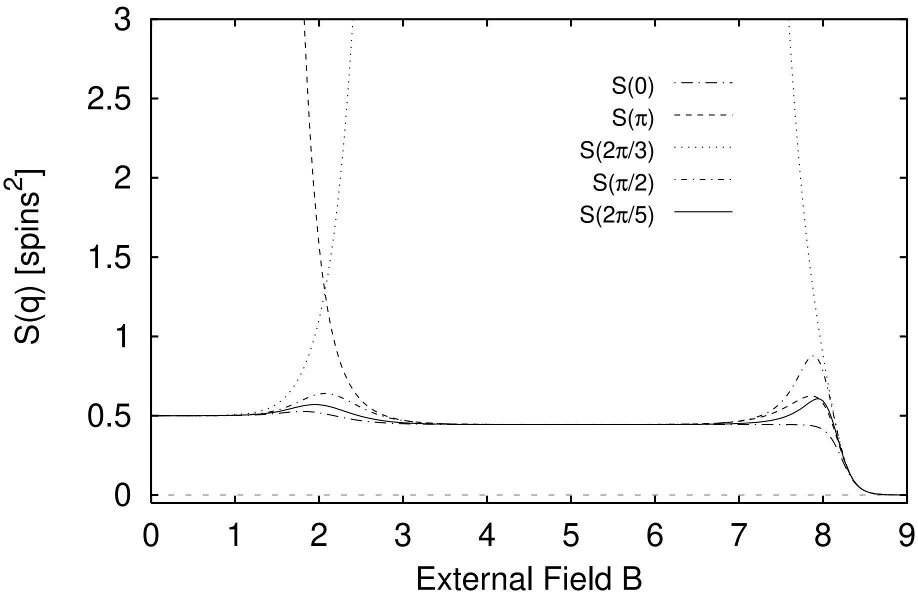
Excess Entropy is sensitive to Periodic patterns at any wavelength



$B = 0.05$, $T = 1.0$, and $J_2 = -1.2$.

A Puzzle: What is happening at $B = 8$?

Structure Factors vs. External Field for NNN Ising System:



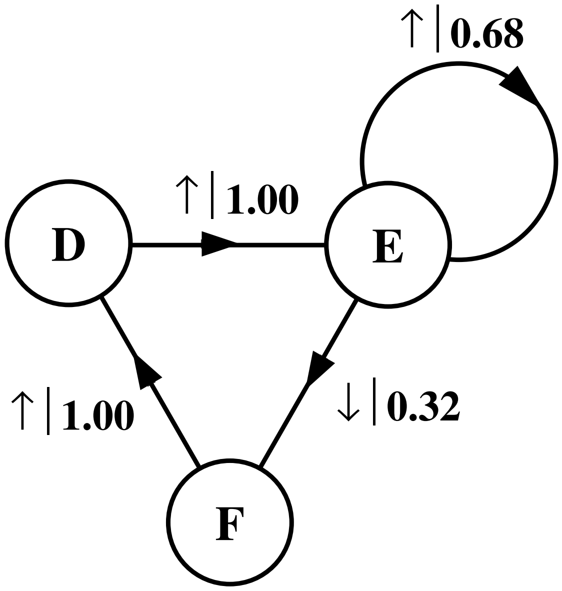
$$T = 0.2, J_1 = -3.0, J_2 = -1.0$$

What is going on at $B \approx 8$???

Presumably, period 1 and period 3 patterns "mix."

But how do the two patterns mix?

ϵ-machine for $B = 8$:



$$\cdots \uparrow\uparrow\downarrow\uparrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\downarrow\uparrow\uparrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow \cdots$$

- The machine tells us how the two patterns “mix” at $B = 8$.
- The machine is a minimal representation of all the regularities present in the configuration.
- In this sense **The Machine is the pattern**.

Conclusion:

Questions:

- What are patterns and how can we discover them?
- What does it mean to say a system is organized?

Summary:

1. Excess entropy, an information theoretic measure of memory, detects periodic structure of any periodicity.
2. Computation theory classifies sets of sequences by considering how difficult it is to recognize them.
3. Causal states and ϵ-machines adapt computation theory for use in a probabilistic setting.
4. The ϵ-machine provides an answer to the question: What patterns are present in a system?
5. I think that computation theory and information theory complement each other well, and can greatly enrich the vocabulary we use to describe complex systems.