The Learning Channel
Central questions:
What are the states?
What is the dynamic?
The Prediction Game

Rules:

1. I give you a data stream (an observed past sequence).
2. You predict its future.
3. You give a model (states & transitions) describing the process.
The Learning Channel ... 
The Prediction Game ...

Process I:
The Learning Channel ...
The Prediction Game ...

Process I:

Past: \ldots 111111111111
The Prediction Game ...

Process 1:

Past: \ldots 111111111111

Your prediction is?
Process I:

Past:   ...111111111111

Your prediction is?

Future: 111111111111...
Process I:

Past: \ldots 1111111111111

Your prediction is?

Future: 111111111111111\ldots

Your model (states & dynamic) is?
The Learning Channel ...
The Prediction Game ...

Process I:

Past: \ldots 111111111111

Your prediction is?

Future: 111111111111 \ldots

Your model (states & dynamic) is?

Complexity Lecture 2: Computational Mechanics (CSSS 2012); Jim Crutchfield
The Learning Channel ...

The Prediction Game ...

Process II:
The Learning Channel ...  
The Prediction Game ...  

Process II:  

Past: ... 10110010001101110
Process II:

Past: \ldots 10110010001101110

Your prediction is?
The Learning Channel ...
The Prediction Game ...

Process II:

Past: ... 10110010001101110

Your prediction is?

Analysis: All words of length L occur & equally often
The Prediction Game ...

Process II:

Past: \ldots 10110010001101110

Your prediction is?

Analysis: All words of length L occur & equally often

Future: Well, anything can happen, how about?
Process II:

Past: $\ldots 10110010001101110$

Your prediction is?

Analysis: All words of length $L$ occur & equally often

Future: Well, anything can happen, how about? $01010111010001101 \ldots$
The Learning Channel ...

The Prediction Game ...

Process II:

Past: ... 10110010001101110

Your prediction is?

Analysis: All words of length L occur & equally often

Future: Well, anything can happen, how about?

01010111010001101110 ...

Your model is?
Process II:

Past: \ldots 10110010001101110

Your prediction is?

Analysis: All words of length L occur & equally often

Future: Well, anything can happen, how about?

01010111010001101 \ldots

Your model is?
The Learning Channel ...
The Prediction Game ...

Process III:
The Learning Channel ...
The Prediction Game ...

Process III:

Past: ...1010101010101010
The Learning Channel ... 
The Prediction Game ... 

Process III: 

Past: \ldots 101010101010101010

Your prediction is?
Process III:

Past: ...1010101010101010

Your prediction is?

Future: 1010101010101011...
The Learning Channel ...

The Prediction Game ...

Process III:

Past: \ldots 101010101010101010

Your prediction is?

Future: 10101010101010101010101010101010101010101010101010101010101010101010 

Your model is?
The Learning Channel ...

The Prediction Game ...

Process III:

Past: \ldots 1010101010101010

Your prediction is?

Future: 101010101010101 \ldots

Your model is?

Complexity Lecture 2: Computational Mechanics (CSSS 2012); Jim Crutchfield
Goal: \[ \rightarrow \]
Predict the future \( S \) using information from the past \( S \)

But what “information” to use?

We want to find the effective “states” and the dynamic (state-to-state mapping)

How to define “states”, if they are hidden?

All we have are sequences of observations
Over some measurement alphabet \( A \)
These symbols only indirectly reflect the hidden states
Effective States:
Effective States:

```
1 0 1 1 0 1 1 0 1
```

Tree representation of the effective states.
Effective States:

\[
\begin{array}{cccccc}
1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 \\
\end{array}
\]

\[t \rightarrow t+1\]
Effective States:

\[
\begin{array}{cccccccc}
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\end{array}
\]

Process is in different “states” when futures look different

\[
\text{State}(t) \sim \text{State}(t+1)
\]
Effective States:

Process is in different “states” when futures look different

State(t) \sim State(t+1)
Effective States:

Process is in different “states” when futures look different:
State(t) \sim State(t+1)

Process is in the same “state” when the future looks the same:
State(t) \sim State(t+2)
Effective for what?
What’s a prediction?

A mapping from the past to the future.

Process \( \Pr(\vec{S}) : \vec{S} = \vec{S} \vec{S} \)

Future: \( \vec{S} \)                    Particular past: \( \vec{s} \)

Future Morph: \( \Pr(\vec{S}^L | \vec{s}) \) (the most general mapping)

Refined goal:
  Predict as much about the future \( \vec{S} \),
  using as little of the past \( \vec{S} \) as possible.
Histories leading to the same predictions are equivalent.

Effective States = **Partitions of History**:

\[ R = \{ R_i : R_i \cap R_j = \emptyset, \underleftarrow{S} = \bigcup_i R_i \} \]

Complexity Lecture 2: Computational Mechanics (CSSS 2012); Jim Crutchfield
Map from histories to partition elements:
\[ \eta : \vec{S} \rightarrow R \]
\[ \eta(\vec{s}) = R_i \]

Random variable:
\[ R = \eta(\vec{S}) \]

Distribution over Effective States:
\[ \Pr(R = R_i) = \sum_{\vec{s} : \eta(\vec{s}) = R_i} \Pr(\vec{s}) \]
How Effective are the Effective States?

**Effective Prediction Error**: Given a candidate partition $R$

$$H[\rightarrow^L S | R]$$

Uncertainty about future given effective states

**Effective Prediction Error Rate**:

$$h_\mu(R) = \lim_{L \to \infty} \frac{H[\rightarrow^L S | R]}{L}$$

Entropy rate given effective states
How Effective are the Effective States?

Statistical Complexity of the Effective States:

$$C_\mu(R) = H[R] = H(\text{Pr}(R))$$

Interpretations:

Uncertainty in state.

Shannon information one gains when told effective state.

Model “size” $\propto \log_2(\text{number of states})$

Historical memory used by $R$. 
Goals Restated:

Question 1:
Can we find effective states that give good predictions?

\[ H[\rightarrow^L S | R] = H[\rightarrow^L S | \leftarrow S] \]

or

\[ h_\mu(R) = h_\mu \]

Question 2:
Can we find the smallest such set?

\[ \min C_\mu(R) \]
Occam’s Pool: The Space of Models

Model = Partition of History Space

Model Space $\mathcal{R} =$ Space of all partitions

Rival Models:

$R_1, R_2 \in \mathcal{R}$
Causal States:

Set of pasts with same morph $\Pr(\vec{S} | \vec{s})$.
Set of histories that lead to same predictions.

Predictive equivalence relation:

$$\vec{s}' \sim \vec{s}'' \iff \Pr(\vec{S} | \vec{S} = \vec{s}') = \Pr(\vec{S} | \vec{S} = \vec{s}'')$$

$$\vec{s}' , \vec{s}'' \in \vec{S}$$
Causal State Components

Causal State = Pasts with same morph: $\Pr(\vec{S} \mid \vec{s})$

$$S = \{ \vec{s} : \vec{s}' \sim \vec{s} \}$$

Set of causal states:

$$\mathcal{S} = \mathcal{S} / \sim = \{ S_0, S_1, S_2, \ldots \}$$

Causal state map:

$$\epsilon : \mathcal{S} \rightarrow S$$

$$\epsilon(\vec{s}) = \{ \vec{s}' : \vec{s}' \sim \vec{s} \}$$
We’ve answered the first part of the modeling goal:

We have the effective states!

Now,

What is the dynamic?
Causal-state Filtering:

\[ \hat{s} = \ldots s_{-3} \quad s_{-2} \quad s_{-1} \quad s_0 \quad s_1 \quad s_2 \quad s_3 \quad \ldots \]

\[ \epsilon(\hat{s}) = \ldots \epsilon(\hat{s}_{-3}) \epsilon(\hat{s}_{-2}) \epsilon(\hat{s}_{-1}) \epsilon(\hat{s}_{0}) \epsilon(\hat{s}_{1}) \epsilon(\hat{s}_{2}) \epsilon(\hat{s}_{3}) \ldots \]

\[ \hat{S} = \ldots \quad S_{t=-3} \quad S_{t=-2} \quad S_{t=-1} \quad S_{t=0} \quad S_{t=1} \quad S_{t=2} \quad S_{t=3} \quad \ldots \]

Causal-state process:

\[ \Pr(\hat{S}) \]
Conditional transition probability:

\[ T_{ij}^{(s)} = \Pr(S_j, s | S_i) \]

\[ = \Pr(S = \epsilon(\leftarrow s s) | S = \epsilon(\leftarrow s)) \]

State-to-State Transitions:

\[ \{T_{ij}^{(s)} : s \in A, \ i, j = 0, 1, \ldots, |S|\} \]
The $\epsilon$-Machine ...

Process $\Rightarrow$ Predictive equivalence $\Rightarrow \epsilon -$ Machine

$\Pr(\vec{S}) \Rightarrow \vec{S} / \sim \Rightarrow \epsilon -$ Machine

$\mathcal{M} = \{S, \{T^{(s)}, s \in A\}\}$

Unique Start State:

$S_0 = [\lambda]$

$\Pr(S_0, S_1, S_2, \ldots) = (1, 0, 0, \ldots)$

Transient States

Recurrent States

Complexity Lecture 2: Computational Mechanics (CSSS 2012); Jim Crutchfield
The Learning Channel ... 

The $\epsilon$-Machine of a Process ...

Denumerable Causal States

Fractal

Continuous

Complexity Lecture 2: Computational Mechanics (CSSS 2012); Jim Crutchfield
Central questions:

- What are the states?  Causal States
- What is the dynamic?  The \( \epsilon \)-Machine

Complexity Lecture 2: Computational Mechanics (CSSS 2012); Jim Crutchfield
The $\epsilon$-Machine ...

**A Model of a Process** $\Pr(S)$:

$\epsilon$-Machine reproduces the process’s word distribution:

$$\Pr(s^1), \Pr(s^2), \Pr(s^3), \ldots$$

$$s^L = s_1 s_2 \ldots s_L \quad S(t = 0) = S_0$$

$$\Pr(s^L) = \Pr(S_0) \Pr(S_0 \rightarrow_{s=s_1} S(1)) \Pr(S(1) \rightarrow_{s=s_2} S(2)) \cdots \Pr(S(L - 1) \rightarrow_{s=s_L} S(L))$$

Initially, $\Pr(S_0) = 1$.

$$\Pr(s^L) = \prod_{l=1}^{L} T^{(s_l)}_{i=\epsilon(s^{l-1}), j=\epsilon(s^l)}$$
Causal shielding:
Past and future are independent given causal state

Process: \( \Pr(\vec{S}) = \Pr(\vec{S}\vec{S}) \)

\( \Pr(\vec{S}\vec{S} | S) = \Pr(\vec{S} | S) \Pr(\vec{S} | S) \)

Causal states shield past & future from each other.

Similar to states of a Markov chain, but for hidden processes.
The $\epsilon$-Machine ...

$\epsilon$Ms are **Unifilar**: $(S_t, s) \to$ unique $S_{t+1}$

(in automata theory, “deterministic”)

That is:

1. $S_i \in \mathcal{S}, \ s \in \mathcal{A}$, at most one $S_j \in \mathcal{S}$:

   $\leftarrow s \in S_i \Rightarrow \leftarrow s \in S_j$

2. If there is a next causal state $j$:

   $S_k \neq j \in \mathcal{S} \Rightarrow T_{ik}^{(s)} = 0$

3. If there is not:

   $T_{ij}^{(s)} = 0$
Consequence:

Unifilarity: 1-1 map between state-sequences & symbol-sequences.

Entropy rate expression requires this 1-1 mapping.

Can use $\epsilon M$ to calculate entropy rate $h_\mu$.

(Any unifilar presentation will do.)
The $\epsilon$-Machine ...

$\epsilon$Ms are Optimal Predictors:

Compared to any rival effective states $R$:

$$H \left[ \overrightarrow{s} \mid R \right] \geq H \left[ \overrightarrow{s} \mid S \right]$$

Proof sketch:  

$$H \left[ \overrightarrow{s} \mid S \right] = H \left[ \overrightarrow{s} \mid s \in S \right]$$  

(Causal equiv. rel’n)

$$= H \left[ \overrightarrow{s} \mid \overleftarrow{s} \in S \right]$$

$$\leq H \left[ \overrightarrow{s} \mid R \right]$$  

(Data processing inequality)

$$R = \eta(\overleftarrow{s})$$
Prescient Rivals $\hat{R}$: Alternative models that are optimal predictors

\[
\hat{R} \in \hat{R} \\
H[\overrightarrow{S} | \hat{R}] = H[\overrightarrow{S} | S]
\]

(Prescient rivals are sufficient statistics for process’s future.)
The $\epsilon$-Machine ...

**Minimal Statistical Complexity:**

For all prescient rivals, $\epsilon M$ is the smallest:

$$C_\mu(\hat{R}) \geq C_\mu(S)$$

Proof sketch:

1. Prescient rivals are refinements, so

$$\exists g : S = g(\hat{R})$$

2. But

$$H[f(X)] \leq H[X] \Rightarrow H[S] = H[g(\hat{R})] \leq H[\hat{R}]$$

3. So

$$C_\mu \leq H[\hat{R}]$$
The $\epsilon$-Machine ...

Minimal Statistical Complexity ...

Consequence:

(1) $C_\mu$ measures historical information process stores.

(2) This would not be true, if not minimal representation.

Remarks:

(1) Causal states contain every difference (in past) that makes a difference (to future) (Bateson “information”)

(2) Causal states are sufficient statistics for the future.
The $\epsilon$-Machine ...

Summary:

$\epsilon M$:

(1) Optimal predictor: Lower prediction error than any rival.

(2) Minimal size: Smallest of the prescient rivals.

(3) Unique: Smallest, optimal, unifilar predictor is equivalent.

(4) Model of the process: Reproduces all of process’s statistics.

(5) Causal shielding: Renders process’s future independent of past.
The $\epsilon$-Machine ...

Dynamical system's intrinsic computation:

(1) How much of past does process store?

(2) In what architecture is that information stored?

(3) How is stored information used to produce future behavior?
Dynamical system’s intrinsic computation:

(1) How much of past does process store?

\[ C_\mu \]

(2) In what architecture is that information stored?

(3) How is stored information used to produce future behavior?
The $\epsilon$-Machine ...

Dynamical system’s **intrinsic computation**:

1. How much of past does process store?
   
   $C_\mu$

2. In what architecture is that information stored?
   
   $\left\{ S, \{ T^{(s)} \}, s \in A \right\}$

3. How is stored information used to produce future behavior?
The $\epsilon$-Machine ...

Dynamical system’s intrinsic computation:

(1) How much of past does process store?

$$C_\mu$$

(2) In what architecture is that information stored?

$$\{S, \{T^{(s)}, s \in A\}\}$$

(3) How is stored information used to produce future behavior?

$$h_\mu$$