Dynamical Systems and Information Theory

Information Theory Lecture 4

Let's consider systems that evolve with time

- That is, systems that can be described as the evolution of a set of *state variables*
- Such evolution can be in discrete or continuous
- The former is governed by *difference* or *recurrence* equations, the later by *differential equations*

$$\mathbf{x}_{t+1} = F\left(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \ldots\right)$$

or

$$\frac{d^n\mathbf{x}}{dt^n} = F_n\left(\mathbf{x}\right)$$

Some Vocabulary

- If *F* is linear, the system is a linear system, likewise nonlinear
- The order of the system is the number of historical terms in the difference equations, or the highest order *n* in the differential equations

$$\mathbf{x}_{t+1} = F\left(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \ldots\right)$$

or

$$\frac{d^n\mathbf{x}}{dt^n} = F_n\left(\mathbf{x}\right)$$

Differential equations in first-order form

- In general, a system of differential equations can be converted to a first order system through the addition of variables
- Here's an example for a second order, linear system

$$0 = -m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx$$

$$\frac{d^2x}{dt^2} = \frac{b}{m}\left(\frac{dx}{dt}\right) + \frac{k}{m}x$$

$$\mathbf{x} = \begin{bmatrix} x \\ dx/dt \end{bmatrix}$$

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} 0 & 1 \\ \left(k/m\right) & \left(b/m\right) \end{bmatrix}\mathbf{x}$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$$

Eigenvalues and Eigenvectors

- Eigen is a German word, which roughly translates to "characteristic"
- For a mathematical transformation of some vector of variables
  - An *eigenvector* of the transformation is a characteristic shape for that transformation
  - An *eigenvalue* is a corresponding magnitude for that shape
- A transformation may have several eigenvalues and eigenvectors
- Representing behaviors of transformations as a combination of eigenvectors is a form of data compression
- We will examine eigenvalues and vectors in continuous dynamical systems *as an example*

An example

- Consider solving a ordinary, linear differential equation
- We solve by assuming a solution form
- Which reduces to the problem of finding eigenvectors

$$0 = -m\ddot{x} + b\dot{x} + kx$$

$$x = Ce^{\lambda t}$$

$$\dot{x} = \lambda x$$

$$\ddot{x} = \lambda^2 x$$

$$0 = -m\lambda^2 x + b\lambda x + kx$$

ignoring the trivial $x = 0$ solution

$$0 = -m\lambda^2 + b\lambda + k$$

$$\lambda = \frac{b \pm \sqrt{b^2 - 4mk}}{2m}$$

In first-order form

- This is the standard eigenvalue problem for **A**
- Solutions are the eigenvalues for the matrix (transformation) **A**
- For a given *λ*, the solution for **x** in *λ* **x=Ax** is an eigenvector

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$$
$$\mathbf{x} = \mathbf{c}e^{\lambda t}$$
$$\lambda\mathbf{x} = \mathbf{A}\mathbf{x}$$
ignoring the trivial $\mathbf{x} = \mathbf{0}$ solution
$$\mathbf{0} = \mathbf{A} - \lambda\mathbf{I}$$
$$0 = |\mathbf{A} - \lambda\mathbf{I}|$$

In dynamical systems

- Eigenvectors (shapes) represent *modes* of the characteristic (unforced) behavior of the system
- Eigenvalues (magnitudes) are related to these shape's *durations through time*

Behold the wonder of Euler

- Eigenvalues come in *complex conjugate pairs*
- Thus
  - positive real parts indicate growth
  - negative real parts indicate decay
  - Imaginary parts indicate frequency of oscillation
- Of the associated eigenvector (shape)

$$e^{it} = \cos t + i \sin t$$
$$e^{(r+\omega i)t} = e^{rt}\left(\cos \omega t + i \sin \omega t\right)$$
for complex conjugate pairs
$$e^{(r\pm\omega i)t} = e^{rt}\left(\cos \omega t\right)$$

In summary

- For a transformation, eigenvectors are characteristic shapes, eigenvalues of their characteristic magnitudes
- For dynamical systems, these the durations through time of modes of behavior
- We can describe continuous linear dynamical systems with a matrix, via first order form
- Eigenvectors of this matrix indicate one of several characteristic "shapes" of a dynamical systems evolution
- For corresponding eigenvalues:
  - Positive real parts indicate that shape grows exponentially
  - Negative real parts indicate that shape dies off exponentially
  - Imaginary parts indicate the speed of oscillation around that shape ("*natural frequency*")

Attractors

- In general, we can say that dynamical systems have *transient behavior* (that which dies out over time) and *steady-state behavior*
- Any steady state behavior is also known as an *attractor* of that system
- Systems can also "diverge" (one of more of their state variables can go to infinity)

Three kinds of attractors

- Fixed points
  - An equilibrium value of the state vector
- Periodic attractors
  - A repeating sequence of state vector values
- Chaotic attractors
  - A sequence that never diverges, but never repeats (!?)
- Attractors can also be stable or unstable

Examining attractors

- As an experiment, let's construct a matrix describing a dynamical systems behavior using *the method of delays*
- This method allows is a non-analytical way of examining system behavior without having to have the system equations
- We can treat either discrete or continuous systems with this method

$$\mathbf{X} = \left[\mathbf{x}_t \,|\, \mathbf{x}_{t-\Delta} \,|\, \mathbf{x}_{t-2\Delta} \,|\ldots \mathbf{x}_{t-M\Delta}\right]$$

Singular value decomposition

- Is a generalization of eigen decomposition (which we'll talk about in more detail later)
- Let's get the singular values *σ* of ***X***
- Then normalize them to 0-1
- The distribution indicates the complexity of system dynamics
- Let's take the entropy of the resulting distribution

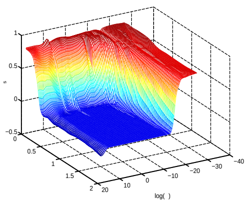
$$\sigma_i' = \frac{\sigma_i}{\sum_j \sigma_j}$$

$$H = -\sum_i \sigma_i' \log \sigma_i'$$

$$\Omega = 2^H$$

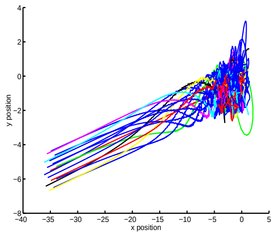
An Example

- Let's consider a set of particles connected with nonlinear springs and dampers
- We can think of this as a sort of "particle swarm"
- Let's look at how *Ω* varies with the spring and damper strength



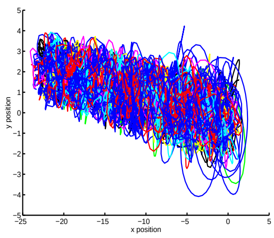
Low *Ω*

- Motion in this figure is largely right to left
- This is the case where the long term behavior is for the particles to "lock" and behave like a single particle
- Relative to the particle's center of mass, this is a fixed point



"Medium" *Ω*

- Is the situation where the particles do not diverge, but do not "coalesce"
- It is likely that this is a chaotic attractor (but I haven't technically proven that)
- We might call the behavior "complex", "emergent" or "self organized"
- We'll look a bit more at "complexity" measures



Symbolic Dynamics

- Let's assume that we are taking measurements of a dynamical system in discrete time, and that each measurement results in one symbol from an alphabet *A*, consisting of *k* possible symbols
- The underlying system might be a discrete or continuous dynamical system
- With or without stochastic elements
- *Note that we are brushing over details of stochastic processes at this point*

Let's consider a symbolic dynamical system (Crutchfield and Shalizi)

- Generating a sequence of symbols
  - $\ldots S_{-2}S_{-1}S_0S_1S_2\ldots$
- For a given time *t*, we will label the past and future sequences
- And we define the notion of *a stationary stochastic process,* if the probability of any measurable future event sequence (taken from the possible set *F*) is independent of time

$\overleftarrow{S}_t$ is the past

$\overrightarrow{S}_t$ is the future

the system is stationary

if $P\left(\overrightarrow{S}_{t_1} \in A \mid \overleftarrow{S}_{t_1} = s\right)$

$= P\left(\overrightarrow{S}_{t_2} \in A \mid \overleftarrow{S}_{t_2} = s\right)$ for all

t_1 and t_2

$\overleftarrow{S}^L$ are the last L symbols

$\overrightarrow{S}^L$ are the next L symbols

Predicting the future

- We want to look at previous symbols, and predict the probability distribution of future symbol sequences
- We are going to *partition* the set of possible previous symbols such that all the elements in a given cell of this partition are matched to the same predicted distribution over the set of possible future sequences
- If the function mapping a past history to a future distribution is η, past sequences s_1 and s_2, are in the same partition cell if and only if $\eta(s_1)=\eta(s_2)$

Effective states

- We will call each cell in this partition an *effective state* of the underlying process, for a given prediction function η
- We will call ***R*** the set of effective states induced by η

Learning

- *We would like to learn the partition, and the predicted distributions, based on past sequences*
- Let's concentrate on getting the right partitions
- We'd like to maximize the mutual information between the partition ***R*** and the possible sequences of future states
- Any prediction that is as good as one could do remembering all past states is called *prescient*

$$I\left(\overrightarrow{S}^L;R\right) = H\left(\overrightarrow{S}^L\right) + H\left(\overrightarrow{S}^L \mid R\right)$$

$$H\left(\overrightarrow{S}^L \mid R\right) \le H\left(\overrightarrow{S}^L \mid \overleftarrow{S}\right)$$

Statistical Complexity

- *C(**R**)* is the number of bits needed to represent the partition
- Note that while this is computed in bits, and is based on a statistical model, it is a different sort of complexity measure than *H*
- It is a sort of "machine size"

Causal states

- We will call the (unique) set of prescient states that minimizes statistical complexity the *causal states* of the system
- Let's recap: this is the most efficient set of sets of previous symbols that predict the probability distribution of future sequences

But there's more

- Given one causal state, and a symbol from the real process, we move to another causal state
- We want to find those transitions, as well
- It turns out that this gives a *deterministic dynamical system* in the following sense
  - For a causal state, and current symbol *s*, the machine moves to another particular causal state, with probability 1
- However, recall the system we are modeling is stochastic,
  - so the model is stochastic, in the sense that the sequence of symbols *s* that are "input" is stochastic
- Also recall that the causal states are mapped to probability distributions over the future states by the function η
- *Whew!*

The system's ε-machine

- Is defined by the symbol set of the original symbolic dynamical system, that system's causal states, and the transition probability matrices $\boldsymbol{T}^{(s)}$

$$T_{ij}^{(s)} = P\left(\vec{S}^{1} = s, S'_{t+1} = \sigma_j \mid S'_t = \sigma_i\right)$$

Markov Process

- The causal states form a *Markov process*
- That is you only need to know the current state to completely determine the probability distribution over all possible future states
- We call also this the *Markov property*

Recurrent, Transient, and Synchronization States

- In a Markov process, states are either
  - Recurrent – visited over and over again in an infinite loop
  - Transient – visited once, and never returned to again
- In an ε-machine, transient states are also called *synchronization states* since the represent the history of symbols you have to see before you can fix yourself into the appropriate recurrent state
- Crutchfield's complexity measures will ignore synchronization states, in general
- We might also call a set of connected recurrent states and *attractor* of the process

Complexity metrics

- We need two numbers to characterize the complexity of the system, given the ε-machine
  - *C(**R**),* the statistical complexity
    - The variable memory needed to represent the machine
  - *H*, the entropy of the state transitions
- *This is rather profound!*

Two kinds of predictable

- Weather that is wildly variable is *predictable in its variability* (high *H*)
  - Well treated with probabilistic models
- Weather that is very periodic is very predictable (high *C*)
  - Well treated with deterministic models
- *Complex* weather is neither of these things
  - (complexity in this sense is characterized by bounded randomness and relatively high size of the machine used to describe dynamics)
  - Hard to get a good model of either kind

Causal state splitting reconstruction (CSSR)

- A somewhat exhaustive algorithm for finding a system's ε-machine
- We start by assuming only one causal state, and the largest possible
- It's very interesting to look at the complexity metrics inferred for various systems

The CSSR algorithm

- Given data from a system of symbol dynamics
  - Start with one causal state and the assumption that symbols are uniformly randomly generated (maximum H)
  - Test statistically to see if causal states should be added
    - If so, add a state, and compute appropriate distributions and transition probabilities from the given data, and repeat
    - If not, stop

Slightly more detail…

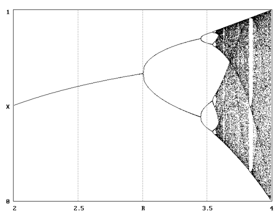
- Set $L=0$, $S'=\{\sigma_0\}$ (the null causal state)
- While $L<L_{max}$
  - For each causal state σ in S'
    - Calculate the conditional probability distribution of all future state sequences of length L
    - For each history in σ
      - Consider each sequence that consists of this history and one more previous character
      - Calculate the conditional probability distribution of all future state sequences of length L
      - Use a statistical test to see if this distribution is the same as that for any existing causal state

If

  - The new history gives a distribution that is statistically the same as that of an existing causal state
    - Add this history to that state
  - Else
    - Create a new state that contains just this history
- Calculate the causal state transitions corresponding to any given symbol
- *I have simplified this terribly!*

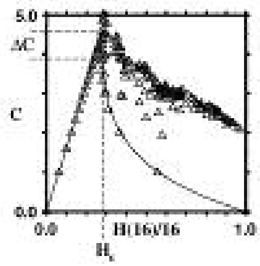
A CSSR Example

- Consider the famous logistic equation
- $X(t+1)=rX(t)(1-X(t))$
- This is the primary example of *deterministic chaos*
- We convert it to a symbolic dynamical system by outputting 1 if $X(t)>0.5$, 0 otherwise



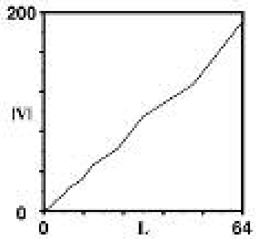
CSSR gives an ε-machine

- For each value of *r,* and $L_{max}=16$
- These are plotted in the space of the two complexity measures C ("machine size") and H ("randomness")
- The phase transition occurs at the Feigenbaum number



At the phase transition

- Adding more inference to CSSR (increasing L_{max}) just leads to larger and larger machine size (*V is approximately 2^C*)
- This is the so-called *edge of chaos*
- It also indicates a jump up Chomsky's hierarchy of grammars

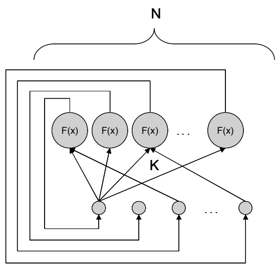


The Edge of Chaos

- Is a phenomena often discussed in the field of Complexity
- It seems to indicate an region of system dynamics bounded by "simple" and "simply random" behaviors, where
- Interesting developmental or accidental patterns and phenomena occur in the system
- It's what I was trying to capture with "medium" Ω

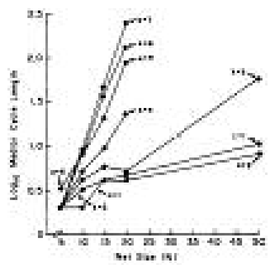
Another study of the edge

- Consider Kaufman's Random Boolean Networks
- Recurrent networks (dynamical systems) with binary outputs/inputs, and random Boolean functions at the nodes
- Characterized by N (number of nodes) and K (connectivity)
- Started with some bit string, they settle towards one of (possibly many) attractors



Attractor Length

- As a function of N and K
- For K < 3 (ish), length of attractors expands as sqrt(N)
- For K > 5 (ish), length of attractors expands exponentially with N
- For K around 3 length of attractors is sub-linear in N…

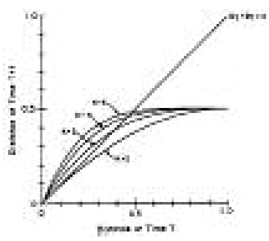


Number of distinct attractors

- As a function of N and K
- For K < 3 (ish), number of attractors expands exponentially with N
- For K > 5 (ish), number of attractors expands as a low-order polynomial of N
- For K around 3 number of attractors expands sub-linearly in N

Stability of attractors

- That is, whether small random perturbations return to a given attractor, or go to some other attractor
- For N<3 (ish) attractors are fairly unstable
- For N>5 attractors unstable
- For N around 3, attractors are stable



Summary of this edge

- K<3: many simple unstable behaviors
- K>5: few complicated unstable behaviors
- K around 3: few medium complicated *stable* behaviors
- This is another edge of chaos
- But is it the same one

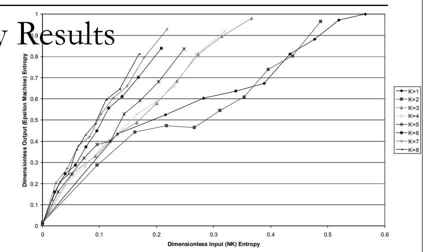
Uniting Crutchfield and Kaufman's Edges?

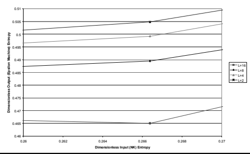
- Procedure
  - Generate large numbers of RBNs, with various levels of *ongoing* perturbation (mutations of the output)
  - Use CSSR to find *ε-machines* for the results
  - Find a unified method of examining the results

"Dimensionless Entropy"

- Consider H/C, the "random" complexity relative to the "machine" complexity
- We examine this for the input and the output of the RBNs:
  - At the input, C is the number of bits necessary to describe the RBN, and H is the entropy of the "mutations"
  - At the output, C and H are as given by CSSR
- We are measuring the complexity of what we can infer, versus what is actually there

Preliminary Results





Take Home Messages

- Dynamical system (including symbolic dynamics) behavior can be characterized by (compressed into)
  - Eigen decomposition (and similar)
  - Attractor description
  - And in a broader sense, information theoretic approaches
  - Which can be characterized by Markov chains
- Such examination reveals, among other things
  - Two distinct *kinds* of complexity: randomness and machine size
  - The edge of chaos phenomena
- These remain active research topics