RECONSTRUCTING LANGUAGE HIERARCHIES

James P. Crutchfield
Physics Department
University of California
Berkeley, California 94720, USA
chaos@gojira.berkeley.edu
chaos%gojira@jade.bitnet

citation: **Information Dynamics**, H. A. Atmanspracher *et al*, editors, Plenum Press, New York (1990).

SUMMARY

Within an assumed language class optimal models can be estimated using Gibbsian statistical mechanics. But how are model classes themselves related? We consider the problem of moving from less to more computationally capable classes in the search for finite descriptions of unpredictable data series.

COMPLEXITY: The Essential Tension between Order and Chaos

Complexity arises at the onset of chaos. Natural systems that evolve with and learn from interaction with their immediate environment exhibit both structural order and dynamical chaos. Order is the foundation of communication between elements at any level of organization, whether that refers to a population of neurons, bees, or humans. For an organism order is the distillation of regularities abstracted from observations. An organism's very form is a functional manifestation of its ancestor's evolutionary and its own developmental memory.

A completely ordered universe, however, would be dead. Chaos is necessary for life. Behavioral diversity, to take an example, is fundamental to an organism's survival. No natural environment can be modeled in its entirety, though. Approximation becomes essential to any system with finite resources. Chaos, as we now understand it, is the dynamical mechanism by which nature develops constrained and useful randomness from finite resources. And from it follow diversity and the ability to anticipate the uncertain future.

There is a tendency, whose laws we dimly comprehend, for natural systems to balance order and chaos, to move to the interface between structure and uncertainty. The result is increased complexity. This often appears as a change in a system's computational capability. The present state of evolutionary progress suggests that one need go even further and postulate a force that drives in time toward successively more sophisticated and qualitatively different computation. The evidence for this is immediate. We can look back to times in which there were no systems that attempted to model themselves, as we do now. This is certainly one of the outstanding puzzles: how can lifeless and disorganized matter exhibit such a drive? And the question goes to the heart of many disciplines, ranging from philosophy and cognitive science to evolutionary and developmental biology and particle astrophysics. The dynamics of chaos, the appearance of pattern and organization, and the complexity quantified by computation will be inseparable components in its resolution.

In the following we consider a restriction of this general problem to a tractable one that concerns modeling temporal sequences of measurements: i.e. inferring models of processes from their data series. The specificity gives a concrete picture of what we mean by a model, its class and language. Here a model will be a machine in the form of one of a variety of stochastic automata; its class, the list of architectural constraints; and its language, the full range of behavior it can describe. Our problem then becomes how to jump levels from a computationally less capable machine type to a more sophisticated one. Thus, we are not so much interested in the optimization of estimated models within any particular class, but rather the relation between levels. The relations consist of learning heuristics that indicate what needs to be innovated in order to define a new more powerful class.

ϵ-MACHINES

The central modeling abstraction is the ϵ-machine.[1] For a given, possibly infinite data stream this is the smallest deterministic machine at the least computationally powerful level that yields a finite description. ϵ here simply refers to the dependence on the nature of the data stream. In the past ϵ has denoted the measurement resolution with which a continuous state system is sampled. It is also consistent with the use in Shannon's dimension rate[2] and in the complexity theory of function spaces.[3] There are, of course, many other parameters that could be listed for a data stream; ϵ is a sufficient reminder.

There is no *a priori* reason to expect a hierarchy of finitely-describable models for nonlinear processes to extend ever upward or, for that matter, to terminate. The distinction between these alternatives is not unlike a Gödelian limitation. Computation theory indicates that the task of estimating optimal models becomes more difficult as the class capability increases. Nonetheless, we take ϵ-machines as a definition of what is humanly conceivable and humanly workable; and also as indicative of contemporary scientific method. From an operational viewpoint, for example, animal learning, scientific research, and biological evolution exhibit the property of compiling natural regularities into compact and usable structures. Physical degrees of freedom are reconfigured into analogical models of observed properties. These form predictive devices and a basis for generalization. The reconstruction of ϵ-machines gives a concrete approach to studying the common elements in these processes.

Table 1 summarizes the overall goal as a hierarchy of ever more sophisticated ϵ-machines. In the progression toward higher complexity less capable models form the representation basis for inference at the next higher level.

RECONSTRUCTION HIERARCHY

At each level of the hierarchy, there is a corresponding model class or representation that determines the language. Each representation has a notion of current state and of transition to a successor state. The set of sequences that each can recognize and generate is indicated by the grammar type. And that in turn gives the structure of the language, i.e. the range of expression of the model class.

A representation is a set of assumptions about the underlying process that produced the data stream. These assumptions we refer to as symmetries, since in many cases they are easily represented by particular semigroups. The procedure of estimating a model within a class corresponds to factoring out the appropriate symmetries from the data stream. Formally, the "factoring out" reconstruction is given by an equivalence relation, generally denoted $\sim$.

The complexity $C(S|\mathcal{M})$ of a data steam S with respect to a given model class $\mathcal{M}$ is the representation's size.[4,5,6] And this typically is taken to be the number of inferred $\mathcal{M}$-states. Just as with Shannon information, which is a special case as discussed below, complexity is a fundamentally relative concept. It is important to note that if, at a given

Machine Reconstruction Hierarchy								
Level	**Grammar**	**Representation**	**State**	**Reconstruction Relation**	**Complexity**	**Symmetry (factored out)**	**Memory**	**Abstraction**
0	subwords	Measurement sequence	observed symbol	S = E / ¯o	Total storage	Choice of experiment E and observables O	Infinite one-dimensional symbol string	Information Observation Measurement
1	common subwords	Tree	tree node	T = S / ¯t	#distinct words, Shannon information	time translation, subword invariance	Infinite tree of distinct subsequences	Probability Typicalness Ergodicity Stationarity
2	Definite	Subshift of finite type, Markov process	Markov state	M = T / ¯f	# of states	finite subword conditioning	Finite set of states	Process
3	Regular	Semi-group	semi-group element	G = T / ¯p	Semi-group complexity	predecessor conditioned future morph equivalence	Finite set of semi-group elements	Feedback Dynamics
4	Regular	Sofic system, Finite Automaton	dynamic state	FA = G / ¯m FA = T / ¯m	Finitary Complexity	future-morph equivalence	Finite set of states and labelled transitions	Parity
5	Regular	String automaton	branching state	SA = FA / ¯c	Branching Complexity	deterministic chain reduction	Finite sets of states and of finite transition strings	
6	Indexed Context Free	Register automaton	register and control state	RA = SA / ¯s	Production Complexity	future state-morph equivalence	Infinite pushdown stack	Recursion
...	...	...	...	...	...	...	...	...
	Phrase structure	Bernoulli-Turing machine	tape state and control state	BTM = S / ¯dr	Bernoulli -Kolmogorov	DTM computation and Randomness	Infinite tape	Universality

Table 1 Computational capability increases going down the table, i.e. with increasing level. See text for discussion.

level, a process P producing a data stream S appears to consist of independent processes $P_1 \otimes P_2 \otimes \cdots$, then the complexity is additive $C(S|\mathcal{M}) = \sum C(P_i|\mathcal{M})$.

The models in the hierarchy are deterministic in the sense of computation theory. This means they deterministically recognize the measurement sequence from which they are reconstructed. In other words, the machines cannot make transitions to different successor states on the same input. Nonetheless, when used as models to generate new sequences, including some possibly never observed in the original data, they employ a source of randomness to select a uniquely labeled transition from a state. That source of guesses is referred to as a random oracle. In this sense the hierarchy is stochastic.

The computational capability of the classes increases going down the table, i.e. with increasing level. For example, Markov processes are less computationally capable than stochastic semigroups and finite automata. This trend to increasing capability is concomitant with the increasing specificity and strength of the assumed statistical and deterministic structure of the process that produced the data. In moving up the hierarchy new abstractions come into play as the appropriate assumptions about the underlying process. Thus, only at the tree level do the notions of stationarity and sequence probability first appear; and only at the semigroup level does intrinsic dynamics arise.

Above, we discussed the complexity in terms of factoring out symmetries from the data stream. While true, the reconstruction process as a whole is incremental. Lower levels provide a representation basis on top of which the next level is reconstructed. Moving stepwise up the hierarchy is an example of incremental learning. Regular patterns of lower level states become the states at the next higher level. In fact, the progression in the table represents, fairly closely, the inference method used in building up finitary and nonfinitary

ϵ-machines; up to the level that is currently implemented: register production machines, the fifth level. The original question then is articulated in terms of the mathematical structure of incremental learning.

The first partial answer to the question of how to do incremental learning comes from considering the convergence properties of lower level representations. For example, as discussed below, the convergence of lower level complexity with increasingly accurate representation determines the complexity at the next higher level. For example, entropy convergence at the tree level determines the finite automaton complexity. And finite automata complexity convergence, in turn, affects the indexed context-free production complexity at the register automaton level.

The evolution to higher capability is simply described in statistical mechanical terms. There is a partition function for each class, $Z_\alpha = \Sigma_{\{\text{states}\}} e^{-\alpha \log p_{\text{state}}}$, that is the sum over that level's states. The complexity is the analog of the Helmholtz free energy, $C_\alpha = (1-\alpha)^{-1} \log Z_\alpha$. At each level one estimates an optimal model.[7,8,9,10] This is taken as the minimal representation consistent with the data to within some error. It is found by minimizing the analog of the Gibbs free energy which relates the change in complexity to representation accuracy and prediction error.

Various labels in the table are given in terms of forward time reconstruction. These can be modified, however, for reverse time reconstruction by consistently replacing "future" with "past" and "predecessor" with "follower". The difference in the corresponding complexities is a measure of statistical irreversibility. The first level at which a useful (nonzero) measure of reversibility occurs is that of semigroup representation.

DATA STREAMS AND PROCESSES

A data stream consists of a long measurement sequence $\mathbf{s} = \{s_0 s_1 s_2 \cdots : s_i \in \mathbf{A}\}$ whose elements are formal symbols in a finite set, the alphabet $\mathbf{A}$ of k symbols. Before discussing models it is natural to lay out the assumptions concerning what is being modeled, viz. a process that produces a symbol sequence. We are interested in *process* languages consisting of observable words and their substrings. One way to motivate this notion is to assume that there is some process P that emits an unending stream of symbols. We, the observers, have no control over how the process started and must simply take the data as it comes. If we have observed a particular word, then obviously we have observed its subwords. When we consider processes as machines, they can start in any state and can stop in any state. In contrast the machines we reconstruct have a single start state. This state represents the observer's condition of total ignorance of the process's state. The machine as a whole captures the temporal evolution of the observer's knowledge of the process as measurements are collected. The reconstruction problem is then how to infer properties of P at a specified level given an observed subset of the process's language $L(P)$. This is a central problem in the computational learning theory of inductive inference.[11]

Given a data stream consisting of a long measurement sequence $\mathbf{s} = \{s_0 s_1 \cdots s_{N-1} : s_i \in \mathbf{A}\}$ we can ask how much storage it requires. Consider the lexicographic ordering of all such sequences. Since there are k^N possible sequences, it will take $N \log_2 k$ bits of storage to specify the ordinal number that uniquely identifies the particular sequence observed. There may be regularities (both deterministic: such as particular sequences being disallowed; and statistical: some sequences being more probable) that will allow for substantially less storage to be used than this upper bound.

TREES

For example, rather than explicitly listing the sequence, if we assume that similar sequences observed at different positions in the data stream represent the same process states, then there is a more compact storage method based on histograms. The associated data structure is a tree, each level L of which represents the number each length L sequence observed anywhere in the data stream.

Given this assumption, the second step in machine inference is the construction of a parse tree. A tree $T = \{\mathbf{n}, \mathbf{l}\}$ consists of nodes $\mathbf{n} = \{n_i\}$ and directed, labeled links $\mathbf{l} = \left\{ l_i : \left(n \underset{s}{\rightarrow} n' \right), \quad n, n' \in \mathbf{n}, s \in \mathbf{A} \right\}$. connecting them in a hierarchical structure with no closed paths. An L-level subtree T_n^L is a tree that starts at node n and contains all nodes below n that can be reached within L links. The tree nodes are this representation's state; the links the state transitions. Trees, then, are machines with no feedback loops. There is no notion of recurrent sequences of tree states.

To construct a tree from a measurement sequence we simply parse the latter for all length D sequences and from this construct the tree with links up to level D that are labeled with individual symbols up to that time. We refer to length D subsequences $s^D = \left\{ s_i \cdots s_{i+D-1} : s_j = (\mathbf{s})_j \right\}$ as D-cylinders. Hence an D level tree has a length D path corresponding to each distinct observed D-cylinder. The picture here is that a particular D-cylinder is a name for that bundle of the underlying process's orbits each of which visited the sequence of measurement partition elements indexed by the D-cylinder. The basic assumption in building a tree is that symbol sequences observed at different times in the data stream approximate the **same** process state. Nonstationary processes are examples for which this assumption fails.

Every node n in the tree is associated with the sequence or word ω_n that leads to it starting from the top tree node n_λ, the tree state of total ignorance. Probabilistic structure is added to the tree T by recording for each node the number $N(\omega_n)$ of occurrences of the associated D-cylinder ω_n relative to the observed total number $N(l)$ of length $l = |\omega_n|$, $p_n^T = N^{-1}(l)\, N(\omega_n)$ where $n_\lambda \underset{\omega_n}{\rightarrow} n$. This provides estimates of the node proabilities $\vec{p}_n = \{p_n : n \in \mathbf{n}\}$. And this in turn gives a hierarchical approximation of the measure in sequence space $\Pi_{\mathrm{Time}} \mathbf{A}$. Tree representations of data streams are closely related to the hierarchical algorithm used for estimating dynamical entropies.[12,13]

ENTROPY

In fact, the notion of entropy as used in information theory is intimately related to tree data structures. For if we ask how much storage is required for a tree representation of a data stream, Shannon's entropy emerges as the most natural estimator for the growth rate of the number of nodes. This follows from consideration of the fundamental relationship between entropy, combinatorics, and probability.[14] Rather than specifying the particular data sequence as one out of k^N, it is identified via two parts: a histogram of counts and the ordinal number within the ensemble indexed by that histogram. For the given alphabet, we must specify k counts $\{v_i : i = 1, \ldots, k\}$ for each symbol and each count may be as large as N. This takes $k \log_2 N$ bits. Then the observed sequence is specified as one out the $N!\,(v_i! \cdots v_k!)^{-1}$ possible sequences described by the ensemble. The total required storage for large N is then dominated by $-N \sum_{i=1}^{k} \frac{v_i}{N} \log \frac{v_i}{N}$. This is clearly related to Shannon's entropy. But that connection follows only if we then assume that probabilities exist and can be determined by the relative frequency estimator, $p_i = N^{-1} v_i$. The analysis up to this point has made no such assumption.

We have considered the histogram of individual symbols. It can be easily extended to account for L-sequences and so it applies to each level of a tree representation. Within the class of tree models the notions of typical subsequence and of probability first come into play, moving up the hierarchy. The analysis shows in what sense entropy is dual to tree models of data streams and how it is the appropriate measure of a tree representation's average size. The tree is built using a sliding window to move through the data stream. It captures in this way the distinct sequences and summarizes their occurrence at different points in the data stream via a count or probability. Entropy measures the number of distinct sequences. That number increases if there is branching as one moves down the tree and forward in time.

Assuming that probabilities are appropriate descriptions, then the total Shannon entropy[2] of length L sequences is

$$H_{\text{Shannon}}(L) = - \sum_{\substack{n \in N \\ |\omega_n| = L}} p_n \log_2 p_n$$

The total Hartley entropy is given simply by the total number of distinct sequences independent of their (positive) probability $H_{\text{Hartley}}(L) = \log_2 N(L)$. If the probability distribution is uniform on the nonzero probability cylinders then these two entropies are equal. Any difference is thus a measure of deviation of the cylinder distribution from uniformity.

The latter observation leads to a parametrized generalization of the entropy introduced by Renyi. This we put into a statistical mechanical formalism by defining a partition function for the tree. The α-order total Renyi entropy,[15] or "Helmholtz free information", of the measurement sequence up to L-cylinders is $H_\alpha(L) = (1 - \alpha)^{-1} \log Z_\alpha(L)$, where the tree partition function is

$$Z_\alpha(L) = \sum_{\substack{n \in u \\ |\omega_n| = L}} e^{\alpha \log p_n}$$

with the probabilities p_n defined on the tree nodes. The Shannon entropy corresponds to the $\alpha = 1$ case, with the application of L'Hopital's rule, and the Hartley entropy to the $\alpha = 0$, with the appropriate definition of the zeroth power of a variable as its indicator function.

The average branching rate in the tree measures the growth rate of the number of new sequences of increasing length. And as such it is a measure of unpredictability in that a periodic process will at some length give rise to no more new cylinders and a random one will. The Renyi specific entropy, i.e. entropy per measurement, is approximated[12] from the L-cylinder distribution by $h_\alpha(L) = L^{-1} H_\alpha(L)$ and is given asymptotically by $h_\alpha = \lim_{L \to \infty} h_\alpha(L)$. The growth rate of total Shannon entropy is often referred to in information theory as the source entropy and in dynamical systems as the metric entropy. It is given by $h_\mu = \lim_{L \to \infty} L^{-1} H_{\text{Shannon}}(L)$. The corresponding Hartley entropy growth rate is called the topological entropy $h_0 = \lim_{L \to \infty} L^{-1} H_{\text{Hartley}}(L)$.

These entropy growth rates can be also given

$$h_\alpha = \lim_{L \to \infty} \{ H_\alpha(L) - H_\alpha(L - 1) \}$$

This form suggests a recursive algorithm for estimating them based on the difference in the cylinder distribution between successive levels in the tree. Consider the following information gain form for the metric entropy

$$h_\mu = \lim_{L \to \infty} \sum_{\substack{s \in \mathbf{A} \\ |\omega| = L-1}} p_{\omega s} \log_2 \frac{p_{\omega s}}{p_\omega}$$

The entropy rate is then the asymptotic information gain of the L-depth tree with respect to the $(L-1)$-depth tree representation. If a tree representation is good, then the information gain, or entropy rate, at some depth will vanish. This indicates that no further information need be stored to represent the process. This happens for a periodic process for trees deeper than the period. If the process is chaotic, with positive entropy, then the information contained in the tree representation will grow exponentially fast with modeling longer subsequences in the data stream. This indicates that the tree representation is inadequate and suggests that the modeler innovate a new class of representations. In the present case of temporal processes, the innovation is the notion feedback dynamics. As we will see many chaotic processes have a finite representation at this next level, whereas they have infinite tree representations.

DYNAMIC STATES

Let us assume then that the process is aperiodic, that is, that the information in the tree structure grows without bound. Then we must move up to the next level of representation whose central notions are feedback and dynamics. At this point we deviate a bit from following the increasing representation complexity given in table 1 by jumping ahead to what is essentially a more capable model class: finite automata.

We ask at this level if there is some dynamic process underlying the estimated tree structure. The reconstruction goal is to infer recurrent states. As a measurement is made, does the observer know what "state" the underlying process is in? The only window onto the process's state is through the intermediary of the chosen observables and measurement method. Isolated measurements do not necessarily correspond to direct detection of the process's state. They are only its indirect reflections.

At this computational level ϵ-machines are represented by a class of labeled, directed multigraph, or l-digraphs.[16] They are related to the Shannon graphs of information theory,[2] to Weiss's sofic systems in symbolic dynamics,[17] to discrete finite automata in computation theory,[18] and to regular languages in Chomsky's hierarchy.[19] Here we are concerned with stochastic versions of these. Their topological structure is described by an l-digraph $G = \{\mathbf{V}, \mathbf{E}\}$ that consists of vertices $\mathbf{V} = \{v\}$ and directed edges $\mathbf{E} = \{e\}$ connecting them, each of the latter is labeled by a symbol $s \in \mathbf{A}$. An edge is triplet consisting of an ordered pair of states $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V} \times \mathbf{A}$ and a symbol. An edge $e = \left(v \underset{s}{\rightarrow} v' \right)$ is interpreted as a transition from state v to state v' on symbol s.

To reconstruct a topological ϵ-machine we define an equivalence relation, subtree similarity, denoted $\underset{m}{\sim}$, on the nodes of a depth D tree T by the condition that the L-subtrees are identical: $n \sim n'$ if and only if $T_n^L = T_{n'}^L$. Naturally, we require that $L < D$ and in practice we take $2L = D$. Subtree equivalence means that the link structure is identical. We refer to the archetypal subtree link structure for each class as a "morph". This equivalence relation induces on T, and so on the measurement sequence s, a set of equivalence classes $\left\{ \mathbf{C}_m^L : m = 1, 2, 3, \ldots \right\}$ given by

$$\mathbf{C}_l^L = \left\{ n \in \mathbf{n} : n \in \mathbf{C}_l^L \text{ and } n' \in \mathbf{C}_l^L \text{ iff } n \sim n' \right\}$$

An l-digraph G_L is then constructed by associating a vertex to each tree node L-level equivalence class; that is, $\mathbf{V} = \left\{ \mathbf{C}_m^L : m = 0, \ldots, V - 1 \right\}$.

If the number of inferred states is finite, $\left\| \left\{ \mathbf{C}_m^L \right\} \right\| < \infty$, then the machine is finitary. The process associated with the machine states is a finite-order Markov chain, although the process associated with the symbol alphabet, i.e. the transitions, need not be finite Markovian.[20]

But this boundedness need not always be the case. It simply defines the finitary ϵ-machines. An infinite number of states can be inferred. This is what is found, in fact, at the onset of chaos via period-doubling,[1,21,22] quasiperiodicity,[21,22] or intermittency.

TRANSITIONS

Two vertices v_k and v_l are connected by a directed edge $e = (v_k \rightarrow v_l)$ if the transition exists in T between nodes in the equivalence classes: $n \rightarrow n' : n \in \mathbf{C}_k^L, n' \in \mathbf{C}_l^L$. The corresponding edge is labeled by the symbol associated with the tree links connecting the tree nodes in the two equivalence classes

$$\mathbf{E} = \left\{ e = \left(v_k \underset{s}{\rightarrow} v_l \right) : \; v_k \underset{s}{\rightarrow} v_l \text{ iff } n \underset{s}{\rightarrow} n' \text{: } n \in \mathbf{C}_k^L, n' \in \mathbf{C}_l^L, s \in \mathbf{A} \right\}$$

OCCAM'S RAZOR AND EXPLANATION

In this way, ϵ-machine reconstruction deduces from the diversity of individual patterns in the data stream "generalized states": the morphs, associated with the graph vertices, that are optimal for forecasting. If an observer knows the process's state then the average uncertainty is the minimum possible, that is, the metric entropy or, equivalently, the process's entropy rate. From a structural viewpoint, the topological ϵ-machines so reconstructed capture the essential computational aspects of the data stream by virtue of the following instantiation of Occam's Razor: Topological reconstruction of G_L produces the minimal and unique machine recognizing the language and the generalized states specified up to L-cylinders by the measurement sequence. This follows from the future morph equivalence relation which is the analog of right equivalence in finite automata minimization.[18] This type of equivalencing goes back to Huffman's DFA reduction[23] and was formalized by Nerode and Moore.[18] Although similar in principle, topological machine reconstruction differs from those approaches in that (i) they are not concerned with data series, rather they start with a given machine and minimize it and (ii) they are concerned with machines that are necessarily finite. Finally, we are ultimately interested stochastic versions of reconstruction: including ways to estimate probabilities for the topologically reconstructed machines and for machines with stochastically similar states.

Minimality guarantees that the model contains no structure and no more properties than the process. For example, there is a 128 state DFA that accepts all binary strings. But we wish to interpret the number of states as a measure of the amount of the process's memory. The 128 state DFA is consistent with the data, but indicates that the process has seven bits of storage. But the process clearly does not. It produces the most unpredictable, ideally random sequences possible. The minimal representation has a single state and so no information storage.

From the Bayesian viewpoint minimality is important since the minimal model consistent with the data maximizes, with respect to all other nonminimal, but consistent models, the posterior probability that the model could have produced that data $\Pr(S|M)$. Thus, via Bayes theorem, minimal machines are the most likely explanation: $\Pr(M|S)$ is maximized.

If the minimal machine is found to be independent of window length L then the storage requirement is $\mathcal{O}(|\mathbf{V}| + |\mathbf{E}|) = \mathcal{O}(V(1 + h_0))$, which is independent of the length of the data stream.

STOCHASTIC MACHINE FROM TREE

If we wish to interpret the tree and so also the machine as representing the average increase in the observer's knowledge with successively longer sequences, then the machine state and transition probabilities need to be estimated. The transition probability from a state $v \in \mathbf{V}$ is found by looking at the branching probabilities at the tree node labeled with the shortest path to the state. Due to the sliding window construction of the tree this gives estimates that use the highest count statistics since the tree node is the closest to the tree's top. Nodes further down in the tree in the state's equivalence class will have relatively diluted statistics. The increased counts in turn reduce the fluctuations and so the estimation error. So the transition probability from a state on symbol $s \in \mathbf{A}$ is given by

$$p_{v \underset{s}{\rightarrow} v'} = \frac{p_{n_{\omega s}}}{p_{n_\omega}}, \text{ where } n_\lambda \underset{\omega}{\rightarrow} n_\omega,\ n_\omega \in C_v, |\omega| \text{ smallest}$$

The state probabilities are estimated from the state-partitioning of tree nodes at morph depth level L. Recall that the probability distribution, and the total number of counts for that matter, on each level is normalized. Going down the tree then dilutes the given constant number of counts into more tree nodes. The deepest level at the tree nodes that have been classified gives the most refined partition. At that level the probability distribution has relaxed as close as possible to its asymptotic value. The state probabilities are then estimated via

$$p_v(L) = \left\| C_v^L \right\|^{-1} \sum_{n \in C_v^L} p_n$$

where C_v^L is the set of trees nodes at level L in the equivalence class of v.

In the case that the underlying process is k-order Markovian or k-periodic, then this method gives the appropriate estimates of the correct stochastic structure when $L > k$. Increasing the length of the data stream simply improves the probability estimates' accuracy. We have skipped over the details required to estimate the stochastic structure of the full class of DFAs or, equivalently, Sofic systems. But the gist of the estimation process is made clear by presenting the Markovian case.

FINITARY COMPLEXITY

With this representation for the data, the question becomes how an ϵ-machine captures a process's dynamics. Many of the important properties of these stochastic automata models, again, are given concisely using a statistical mechanical formalism that describes the coarse-grained scaling structure of orbit space. The statistical structure of an ϵ-machine is given by a parametrized stochastic connection matrix $T_\alpha = \Sigma_{s \in \mathbf{A}} T_\alpha^{(s)}$, that is the sum over each symbol of the state transition matrices $T_\alpha^{(s)} = \left\{ e^{\alpha \log p(v_i | v_j; s)} \right\}$. The α-order finitary complexity is the free energy $C_\alpha(L) = (1 - \alpha)^{-1} \log Z_\alpha^v(L)$ where the machine partition function is $Z_\alpha(L) = \sum_{v \in \mathbf{V}} e^{-\alpha \log p_v}$ and the probabilities p_v are defined on the ϵ-machine's vertices $v \in \mathbf{V}$. The finitary complexity is a measure of an ϵ-machine's information processing capacity in terms of the amount of information stored in the morphs. It is directly related to the mutual information of the past and future semi-infinite sequences and to the convergence[24,25,26] of the entropy estimates $h_\alpha(L)$. It can be interpreted, then, as a measure of the amount of mathematical work necessary to produce a fluctuation from asymptotic statistics. The units for complexity measures are bits of information. However, at this level we see that the complexity begins to more strongly reflect the degree of computational capability and so we refer to the units as Turings, rather than bits. At this low level the difference between bits and Turings is not as dramatic as at higher levels where each unit of machine structure is clearly associated with sophisticated computation.

The entropies and complexities are dual in the sense that the former is determined by the principal eigenvalue λ_α of T_α, $h_\alpha = (1-\alpha)^{-1} \log_2 \lambda_\alpha$ and the latter by the associated left eigenvector of T_α, $\vec{p}_\alpha = \{p_v^\alpha : v \in \mathrm{V}\}$ that gives the asymptotic vertex probabilities. A complexity based on the asymptotic edge probabilities $\vec{p}_e = \{p_e : e \in \mathbf{E}\}$ can also be defined $C_\alpha^e = (1-\alpha)^{-1} \log \Sigma_{e \in \mathbf{E}} \; p_e^\alpha$, where $\vec{p}_e$ is given by the left eigenvector of the ϵ-machine's edge graph. The transition complexity C_1^e is simply related to the entropy and graph complexity by $C_1^e = C_1 + h_1$. There are, thus, only two independent quantities for a finitary ϵ-machine.[27]

The two limits for α warrant explicit discussion. For the first, topological case $(\alpha = 0)$, T_0 is the l-digraph's connection matrix. The Renyi entropy $h_0 = \log \lambda_0$ is the topological entropy h. And the finitary complexity is $C_0 = \log |\mathrm{V}|$. This is $C(\mathrm{s}|\mathrm{DFA})$: the size of the minimal DFA description, or "program", required to *produce* sequences in the observed ensemble of which s is a member. This topological complexity counts all of the reconstructed states. It is similar to the regular language complexity developed for cellular automaton spatial patterns.[28] The DFAs in that case were constructed from known equations of motion and an assumed neighborhood template. In the second, metric case $(\alpha = 1)$, h_α becomes the metric entropy $h_\mu = \lim\limits_{\alpha \to 1} h_\alpha = -\frac{d\lambda_\alpha}{d\alpha}$. The metric complexity $C_\mu = \lim\limits_{\alpha \to 1} C_\alpha = -\Sigma_{v \in \mathrm{V}} \; p_v \log p_v$ is the Shannon information contained in the morphs. These measures have been discussed before as the "set complexity" version of the regular language complexity[26]. Following the preceding remarks, the metric entropy is also given directly in terms of the stochastic connection matrix

$$h_\mu = -\sum_{v \in \mathrm{V}} p_v \sum_{\substack{v' \in \mathrm{V} \\ s \in \mathrm{A}}} p(v|v';s) \log p(v|v';s)$$

KNOWLEDGE RELAXATION

ϵ-machines are representations of an observer's model of a process. Starting from the state of total ignorance about the process, successive steps through the machine correspond to a refinement of the observer's knowledge based on observations. The average increase is given by a diffusion of information throughout the given model. In the case of trees, we have a flow of probability downwards, in increasing time, toward the leaves. This is a unidirectional diffusion of information on an ultrametric structure.[29] The ultrametric distance on the tree is sequence length or, more simply, time itself. The tree and machine transition probabilities, especially those connected with transient states, govern the relaxation process of the observer gaining more information about the process with longer measurement sequences.

A measure of information relaxation on finitary machines is given by the length-dependent finitary complexity $C_\mu(t) = H(p_{\mathrm{V}}(t))$ where $H(P)$ is the Shannon entropy of the distribution P and $p_{\mathrm{V}}(t)$ is the probability distribution at time t beginning with the initial distribution $p_{\mathrm{V}}(0) = (1, 0, 0, \ldots)$ concentrated on the start state. The latter distribution represents the observer's state of total ignorance of the process's state, i.e. before any measurements have been made, and correspondingly $C_\mu(0) = 0$. The length-dependent complexity $C_\mu(t)$ is simply (the negative of) the Boltzmann H-function in the present setting. And we have the analogous result to the H-theorem for stochastic ϵ-machines: $C_\mu(t)$ is monotonically increasing. Furthermore, the observer has the maximal amount of information about the process, i.e. the observer's knowledge is in equilibrium with the process, when $C_\mu(t+1) - C_\mu(t)$ vanishes for $t > t_{\text{lock}}$. Generally, we have $C_\mu(t) \underset{t\to\infty}{\rightarrow} C_\mu$. That is, the length-dependent complexity limits on the metric complexity.

For finitary machines there are two convergence behaviors for $C_\mu(t)$: finite time convergence for periodic and Markovian (subshift of finite type: SSFT) processes and asymptotic convergence for strictly Sofic systems (SSS).[20] These are illustrated in figure 1. The effect of extrinsic noise on this type of convergence process was studied some time ago. It was found that extrinsic noise induces an effective Markovian process.[25]

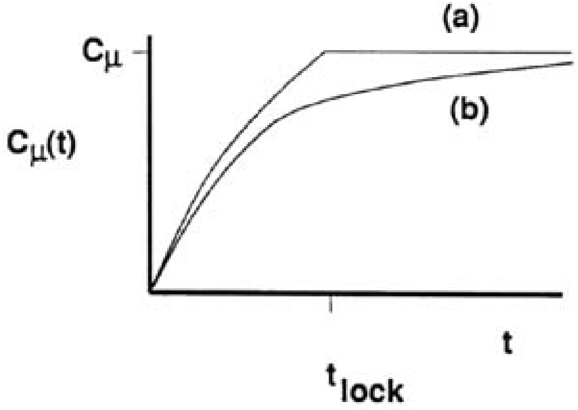

Figure 1 Temporal convergence of length-dependent complexity $C_\mu(t)$ for (a) periodic and Markovian (subshifts of finite type) processes and (b) for strictly Sofic systems.

Periodic and SSFT processes converge in finite time. SSSs asymptote but do not converge in finite time due to the relaxation of the initial distribution through an infinite number of Cantor sets. Through a structural analysis we see that the difference $\Delta C_\mu(t) = C_\mu - C_\mu(t)$ is a measure of the amount of information in the transient states. For SSSs this quantity only asymptotically vanishes since there are transient cycles that persist for all time, even though their probability decreases asymptotically. This leads to general definition of (chaotic or periodic) phase locking. We say that the observer has β-locked onto the process when $\Delta C_\mu(t_{lock}) < \beta$ this occurs at the locking time t_{lock}. When the process is periodic, this notion of locking is the standard one of engineering. But it also applies to chaotic processes and corresponds to the observer knowing what state the process is in, even if the next measurement cannot be predicted exactly.

MEANING

Enough structure has been developed up to this point to introduce a quantitative definition of the meaning of an observation. The meaning of a message, of course, depends on the context in which its information is made available. If the context is inappropriate the observation will have no basis with which to be understood. It will have no meaning. If appropriate, then the observation will be "understood". And if that which is understood, i.e. the content of the message, is largely unanticipated then the observation will be more significant than a highly likely, "obvious", message.

In the present discussion context is set by the model held by the observer at the time of a measurement. To take an example, assume that the observer is capable of modeling with respect to dynamic states. And, in particular, the observer has estimated a stochastic finite automaton and has been following the process sufficiently long to know the current state with certainty. Then at a particular time the observer measures symbol $s \in \mathbf{A}$. If that measurement forces a disallowed transition, then it has no meaning. Indeed, the response is for the observer to reset the machine to the initial state of its ignorance. If, however, the measurement is allowed, i.e. it is anticipated, then the amount of meaning is $-\log p_{\underset{s}{\rightarrow} v}$. Here $\underset{s}{\rightarrow} v$ denotes the machine state to which the measurement brings the observer's knowledge of the process's state and $p_{\underset{s}{\rightarrow} v}$ is the corresponding morph's probability. The meaning itself, i.e. the content of the measurement, depends on the morphs to which the model's states correspond.

Naturally, similar definitions of meaning can be developed between any two levels in a reconstruction hierarchy. Here we wish to emphasize the main components of meaning, as we have defined it: (i) it is an information-like quantity and (ii) it derives fundamentally from the relationship *across* levels. A given message has different connotations depending on the level. Meaning appears as a change in connotation. This definition answers a question posed previously.[30]

INFINITARY MACHINES

Recall that it is possible to infer machines with an infinite number of states or morphs from an infinite tree representation. As in the transition from an infinite tree representation to a finite machine, we can ask for a measure that indicates that the finitary machine representation is inadequate. Observing that breakdown we also look for some regularity in the infinite "finitary" machine that we can factor out.

The measure we use for this innovation step plays the same role as entropy in going from trees to finitary machines. We look at the growth rate at the total information contained in the representation[21,22] $c_\alpha = \lim_{L \rightarrow \infty} L^{-1} 2^{C_\alpha(L)}$. In the topological case, we consider the growth rate of the number of states $c_0 = \lim_{L \rightarrow \infty} L^{-1} \| \mathbf{V}(L) \|$. There is also an information gain version of this

$$c_\alpha = \lim_{L \rightarrow \infty} \left[2^{C_\alpha(L)} - 2^{C_\alpha(L-1)} \right]$$

$$c_\alpha = \lim_{L \rightarrow \infty} \sum_{\substack{v \in \mathbf{V}(L-1) \\ v' \in \mathbf{V}(L)}} p_v \log \frac{p_{v'}}{p_v}$$

In the last form, $v = \{v'\} / \underset{L-1}{\sim}$ is a distinct state at length $L-1$ that splits into states $\{v'\}$ at length L reconstruction. When this growth rate is positive then another higher level representation is called for. In this case we have register machines that perform string productions in registers whose lengths grow monotonically. Register machines are inferred from morphs consisting of regular patterns of state transitions in the "infinite" finitary machine.[21,22]

We should also note one of the highest levels, that of the universal Turing machine (UTM). As already noted, the more powerful the representation the more difficult it is to estimate minimal models. At the level of UTMs there is no general procedure for inferring the minimal program from a given data stream. An important difference with our approach is that we are, in effect, considering a class of computations based on deterministic Turing machines with access to a random register: the random oracle. This is the Bernoulli-Turing machine (BTM).[21] Modeling with respect to BTMs trades off deterministic computation against random guessing. Thus, at every level of the hierarchy, very regular and highly random processes are simply described and so have low complexity.

CONCLUDING REMARKS

This discussion has explored the idea of hierarchically reconstructing models by focusing of the particular example of moving from a data stream to a tree representation to a finitary machine representation, and finally to a string register machine representation. Each level has an equilibrium statistical mechanics that indicates via a minimization, or variational, principle what the optimal model at each level is. In going between levels the significant measures are growth rates of complexity, i.e. the growth of information contained in a representation. There is a statistical metamechanics in which the innovation from infinite representations at one level lead via a condensation or phase transition to a finite representation at a higher level. We gave a quantitative definition of meaning that is appropriate to the type of incremental learning considered here.

Hierarchical reconstruction hints at how to approach complexity on much larger scales. It indicates some of the necessary structure of evolutionary processes that appear to play off randomness and order. Some portion of natural language undoubtedly manifests such a developmental tension: the need for structural regularity, for example, in order to support communication between individuals and the need for a diverse and rich language to support expressiveness and specificity.

This essay has reviewed the attempt to weave together the structural framework of computation and formal language theories and the combinatorial notions of information theory into an approach to modeling periodic and chaotic nonlinear dynamical systems. The tools ultimately relie on semigroup theory, symbolic dynamics, statistical mechanics, and ergodic theory. Even Bayesian statistics appears in the parameter estimations and in the thermodynamic analogies. The statistical mechanics of inductive inference and learning appear to give a unified framework for these very different disciplines. Similar approaches to learning in neural networks have been developed recently.[31]

We close by reframing the question posed at the beginning concerning the naturalness of the drive toward higher complexity: Is it a physical property? Or is it the figment of an organism's need to model its environment? These are captured in the single philosophical question,[32] where does intentionality lay? From the present discussion's radical and shameless mechanistic bias it is difficult to see how intentionality and the complexity drive could not be at root a property of physical nature. The difficulty in understanding this remains, of course, in posing the question in a scientific manner that does not strip it of its philosophical content. We continue to wonder ... from what does the anticipation of knowledge spring?

ACKNOWLEDGEMENTS

The author thanks Karl Young and the participants of the seminar on Foundational Issues in Cognitive Science for useful discussions and Professor Carson Jeffries for his continuing support. This work was funded in part by ONR contract N00014-86-K-0154.

REFERENCES

1. J. P. Crutchfield and K. Young. Inferring statistical complexity. *Phys. Rev. Let.*, 63:105, 1989.
2. C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Champaign-Urbana, 1962.
3. A. N. Kolmogorov and V. M. Tikhomirov. ϵ-entropy and ϵ-capacity of sets in function spaces. *Uspekhi Mat. Nauk.*, 14:3, 1959. (Math. Rev. 22, No. 2890).
4. R. J. Solomonoff. A formal theory of inductive control. *Info. Control*, 7:224, 1964.
5. A. N. Kolmogorov. Three approaches to the concept of the amount of information. *Prob. Info. Trans.*, 1:1, 1965.
6. G. Chaitin. On the length of programs for computing finite binary sequences. *J. ACM*, 13:145, 1966.
7. J. G. Kemeny. The use of simplicity in induction. *Phil. Rev.*, 62:391, 1953.
8. E. T. Jaynes. Where do we stand on maximum entropy? In E. T. Jaynes, editor, *Essays on Probability, Statistics, and Statistical Physics*, page 210. Reidel, London, 1983.
9. J. Rissanen. Stochastic complexity and modeling. *Ann. Statistics*, 14:1080, 1986.
10. J. P. Crutchfield and B. S. McNamara. Equations of motion from a data series. *Complex Systems*, 1:417, 1987.
11. D. Angluin and C. H. Smith. Inductive inference: Theory and methods. *Comp. Surveys*, 15:237, 1983.
12. J. P. Crutchfield and N. H. Packard. Symbolic dynamics of one-dimensional maps: Entropies, finite precision, and noise. *Intl. J. Theo. Phys.*, 21:433, 1982.
13. J. P. Crutchfield. *Noisy Chaos*. PhD thesis, University of California, Santa Cruz, 1983. published by University Microfilms Intl, Minnesota.
14. R. E. Blahut. *Principles and Practice of Information Theory*. Addision-Wesley, 1987.
15. A. Renyi. On the dimension and entropy of probability distributions. *Acta Math. Hung.*, 10:193, 1959.
16. D. M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of Graphs*. Academic Press, New York, 1980.
17. R. Fischer. Sofic systems and graphs. *Monastsh. Math*, 80:179, 1975.
18. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 1979.
19. N. Chomsky. Three models for the description of language. *IRE Trans. Info. Th.*, 2:113, 1956.
20. J. P. Crutchfield and K. Young. Finitary ϵ-machines. in preparation, 1989.
21. J. P. Crutchfield and K. Young. Computation at the onset of chaos. In W. Zurek, editor, *Entropy, Complexity, and the Physics of Information*. Addison-Wesley, 1990.

22. J. P. Crutchfield and E. Friedman. Language cascades: A universal structure at the onset of chaos. preprint, 1990.
23. D. Huffman. The synthesis of sequential switching circuits. *J. Franklin Inst.*, 257:161, 275, 1954.
24. J. P. Crutchfield and N. H. Packard. Noise scaling of symbolic dynamics entropies. In H. Haken, editor, *Evolution of Order and Chaos*, page 215, Berlin, 1982. Springer-Verlag.
25. J. P. Crutchfield and N. H. Packard. Symbolic dynamics of noisy chaos. *Physica*, 7D:201, 1983.
26. P. Grassberger. Toward a quantitative theory of self-generated complexity. *Intl. J. Theo. Phys.*, 25:907, 1986.
27. J. P. Crutchfield. Inferring the dynamic, quantifying physical complexity. In N. B. Abraham, A. M. Albano, A. Passamante, and P. E. Rapp, editors, *Measures of Complexity and Chaos*, page 327. Plenum Press, 1990.
28. S. Wolfram. Computation theory of cellular automata. *Comm. Math. Phys.*, 96:15, 1984.
29. R. Rammal, G. Toulouse, and M. A. Virasoro. Ultrametricity for physicists. *Rev. Mod. Phys.*, 58:765, 1986.
30. J. P. Crutchfield. Information and its metric. In L. Lam and H. C. Morris, editors, *Nonlinear Structures in Physical Systems - Pattern Formation, Chaos and Waves*, Berlin, 1989. Springer-Verlag.
31. E. Levin, N. Tishby, and S. A. Solla. A statistical approach to learning in layered neural networks. In R. Rivest, D. Haussler, and M . K. Warmuth, editors, *Computational Learning Theory*, page 245. Morgan Kaufmann, 1989.
32. D. C. Dennett. *The Intentional Stance*. MIT Press, Cambridge, Massachusetts, 1987.