Context-tree modeling of observed symbolic dynamics

Matthew B. Kennel*
Institute for Nonlinear Science, University of California, San Diego, La Jolla, California 92093-0402

Alistair I. Mees
Centre for Applied Dynamics and Optimization, The University of Western Australia, Nedlands, Perth 6907, Western Australia

(Received 15 March 2002; revised manuscript received 16 August 2002; published 26 November 2002; publisher error corrected 20 December 2002)

Modern techniques invented for data compression provide efficient automated algorithms for the modeling of the observed symbolic dynamics. We demonstrate the relationship between coding and modeling, motivating the well-known *minimum description length* (MDL) principle, and give concrete demonstrations of the ''context-tree weighting'' and ''context-tree maximizing'' algorithms. The predictive modeling technique obviates many of the technical difficulties traditionally associated with the correct MDL analyses. These symbolic models, representing the symbol generating process as a finite-state automaton with probabilistic emission probabilities, provide excellent and reliable entropy estimations. The resimulations of estimated tree models satisfying the MDL model-selection criterion are faithful to the original in a number of measures. The modeling suggests that the automated context-tree model construction could replace fixed-order word lengths in many traditional forms of empirical symbolic analysis of the data. We provide an explicit pseudocode for implementation of the context-tree weighting and maximizing algorithms, as well as for the conversion to an equivalent Markov chain.

DOI: 10.1103/PhysRevE.66.056209 PACS number(s): 05.45.Tp, 02.50.Tt, 05.10.Gg

I. INTRODUCTION

We observe a time-ordered symbol stream $S=\{s_1,s_2,s_3,\ldots,s_N\}$, which is either quantized from continuous-valued observations or measured directly, each element from an alphabet A, and expressible as $s\in\{1,2,\ldots,|A|\}$. The distribution of multisymbol words provides information about time-dependent structure and correlation, just as, with continuous nonlinear data, time-delay embedding provides a vector space revealing dynamical information. Our goal is to construct compact and reliable models of the predictive probability distribution, the evolution law of the implied information source $P(s_{t+1}|s_t,s_{t-1},\ldots)$.

In any inductive inference of models from finite observed data, balancing complexity with apparent predictability, is a key issue. Excessive free parameters in a highly general model class (more complexity) overfit sample fluctuations and give models that fail to generalize to unobserved data despite low error on the fitted sequence. Beyond more conventional techniques such as cross validation or other forms of data withholding and testing, the minimum description length (MDL) principle [1] provides an information-theoretical solution. Though there are various implementations differing in detail, the central theme is to summarize overall performance in the *description length* as the information required to describe data relative to some model plus the information necessary to specify that model and its parameters are out of a broader class of models. Reduced to practice across a wide range of regression and modeling tasks (e.g., Ref. [2] in dynamical systems), the MDL principle has been demonstrated to give sensible and consistent results, usually equaling or outperforming traditional and often more *ad hoc* model selection criteria.

Given model $\hat{p}_\theta(\cdot|s_t,s_{t-1},\ldots)$, a two-part codelength is

$$DL(\theta)=L_{\hat{p}}(S)+L(\theta)=\sum_{i=0}^{N}-\log\hat{p}(s_{i+1}|s_i,s_{i-1},\ldots)+L(\theta), \tag{1}$$

with the MDL model $\hat{p}_{\mathrm{MDL}}=\arg\min_{\hat{p}_\theta}DL(\theta)$. The description length and entropies are in units of bits when logarithms are base 2, which will be assumed for the remainder of the paper unless denoted otherwise. The primary difficulty of MDL implementation is evaluating the complexity cost of model classes. Model classes frequently have both discrete (e.g., number and kinds of parameters) and continuous (parameter values themselves) degrees of freedom. Accounting for the model cost of continuous parameters is usually more difficult than for discrete parameters.

The technology of ''sequential'' coding techniques motivates a solution to account for parametric complexity. Such algorithms are adaptive, i.e., after processing some amount of observed data they reestimate models of the source to improve their performance. After encoding t symbols, we denote the best model having used *only the previously observed data* as $\hat{P}_t(\cdot|s_t,s_{t-1},\ldots,s_1)$. Then, the next symbol s_{t+1} may be encoded with, for example, an arithmetic coder [3], with cost $-\log\hat{P}_t(s_{t+1}|s_t,\ldots)$. The internal model is subsequently updated to reflect knowledge of s_{t+1}. The output of the coder may be transmitted over a hypothetical channel and causally decoded at the receiver. At time N, we have a good model of the source but more importantly, the codelength

*Electronic address: mkennel@ucsd.edu

$$L=\sum_{t=0}^{N-1} -\log \hat{P}_t(s_{t+1}|s_t,\ldots) \tag{2}$$

implicitly includes the complexity cost because all information necessary to replicate the input data has been transmitted, even though no explicit encoding of the model parameters or the structure was necessary! Minimizing L is the ''sequential minimum description length'' principle.

An example is helpful. Consider independent symbols drawn from an alphabet A with a fixed but unknown distribution $p_k, k\in 1\cdots|A|$. Having observed the counts c_k, the maximum-likelihood estimator of p is the obvious $\hat{p}_k = c_k/N$. The negative log likelihood is

$$L_{\mathrm{ML}}=\sum_k c_k(-\log p_{\hat{k}})=N\sum -\hat{p}_k \ln \hat{p}_k = NH(\hat{p}) \tag{3}$$

is not a fair codelength as it assumes that one can encode the early symbols already knowing $\hat{p}$. Instead encode sequentially with the distribution

$$\hat{p}_k=\frac{c_k+\beta}{\sum_j (c_j+\beta)}, k\in 1\cdots|A|, \tag{4}$$

where $\beta>0$, with c_k being the accumulated counts of previously observed symbols. Positive β ensures finite $-\ln\hat{p}$ when $c_k=0$. The net codelength

$$L_{\mathrm{PMDL}}=\sum_{j=1}^{N} -\ln \hat{p}(s_j) \tag{5}$$

is realizable. For the binary alphabet, $|A|=2$, $\beta=1/2$ is optimal, resulting in a parametric redundancy (excess codelength versus entropy, $L-Nh$) of $\rho\leqslant\frac{1}{2}\log N+1$ independent of the distribution, and is known as the Krichevsky-Trofimov (KT) estimator [4]. For $|A|>2$ and $\beta\neq 1/2$, the redundancy may depend on the underlying parameters, but in all cases there is a leading term proportional to $\log N$, so that the per symbol redundancy $\rho(s_1,\ldots,s_N)/N\to 0$ as $N\to\infty$ (see also Ref. [5]). Asymptotically, the codelength per symbol approaches the entropy rate, and thus this is *universal compression* for independent identically distributed (iid) discrete sources. Moreover, the asymptotic rate achieves the best possible leading term $(k/2)\log N$ [6] for any source with k parameters. $\hat{P}(S)\triangleq 2^{-L(S)}$ is a *coding distribution* for the sequence S itself, satisfying the Kraft inequality. (We use the symbol $\triangleq$ for definitions.)

II. CONTEXT TREES

We model more complex sources than the trivial one just discussed with *context trees*. A tree machine (Fig. 1) is a subclass of finite-state automata with stochastic emission probabilities and deterministic state transitions, given an emitted symbol. One follows recent symbols (the context) down the tree (deeper corresponding to more ancient symbols) and upon matching a terminal node, defines the state.

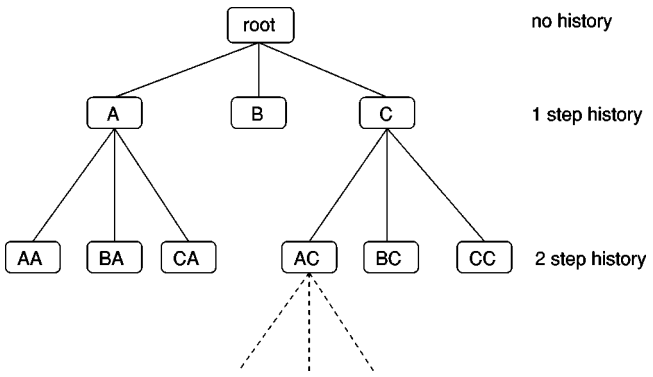

FIG. 1. Example of a small context tree for a three symbol alphabet. Internal nodes (nodes with deeper children) are the root node A, C, and AC, and terminal nodes AA,BA,CA,B,BC,CC. Descendants of AC continue off the figure. Each node accumulates counts of future symbols and internal code lengths.

The state emits independent symbols with a certain distribution. A tree with all nodes at depth D is a D-order Markov chain.

Consider estimating from the data a tree model whose topology alone is known. Every terminal node j retains counts c_k^j and with Eq. (4) an estimator $\hat{p}^j$. The tree model's estimate $\hat{p}(s_{t+1}|s_t,s_{t-1},\ldots)$ at each time is that $\hat{p}^m$ whose node m matches the recently processed symbols. Each node accumulates its codelength as Eq. (5) denoted L_e, with the sum $L=\Sigma_j L_e^j$ a fair codelength for the source—given an *a priori* topology.

The nontrivial issue, of course, is estimating a suitable topology for the data, as that directly addresses the complexity versus predictability issue, whether to use a shallow tree whose nodes collect more data and hence are better local estimators or to use a deeper tree because it is necessary to distinguish distinct states recognizable from the data. There are 2^{2^D} topologies of binary trees with maximum depth no larger than D; for any but the smallest D, choosing among them would appear to be prohibitively expensive. The *context-tree weighting* (CTW) algorithm by Willems, Shtarkov, and Tjalkens [7] provides a simple and clever recursive algorithm that performs an optimal *weighting* over trees in time proportional to D, resulting in a general coding algorithm with the excellent compression performance and acceptable computational cost. The upper bounds on redundancy (relative to any tree source) are pointwise for any sequence and not only in probability. Willems proves in Ref. [8] that the infinite-depth method is universal for stationary, ergodic, binary source, and achieves the Rissanen lower bound on redundancy $[(k/2)\log N]$ for finite memory tree sources (includes Markov chains). Although not stated in Ref. [8], universality is also true for nonbinary finite alphabets with an estimator like Eq. (4) [9]. CTW is further distinguished among the other universal context-tree methods by achieving this bound plus only a constant even for finite length strings and without arbitrary free parameters.

We present the CTW algorithm. One dynamically builds a tree for all previously observed contexts, retaining counts at every node. Consider weighting between a parent node with context s and its children cs. CTW recursively mixes the

local estimator at any node and the weighted estimators of *all* children:

$$P_w^s \triangleq \frac{1}{2}\left[P_e^s + \prod_{c\in A} P_w^{cs}\right]. \tag{6}$$

If no children are present, $P_w \triangleq P_e$. At the root λ, $P_w^\lambda(S)$ is the coding distribution for S with $L_{\mathrm{CTW}}(S) = -\log P_w^\lambda(s)$ its codelength. Nodes store L_e and L_w and implement Eq. (6) as $L_w^s = -\log P_w^s = 1 + \min(L_e^s, L_c) - \log(1 + 2^{-|L_e^s - L_c|})$, with $L_c = \sum_c L_w^{cs}$.

This code is incrementally updatable. Starting from the deepest matching node, one updates L_e with $-\log \hat{p}(s_{t+1})$, and subsequently increments the locally stored c_k with the knowledge of s_{t+1}. Then, each node's parent also updates its L_e with the new observation and its L_w from the updated child, accumulating the new observation, and proceeding to shallower nodes until the root node is reached. To ensure causal decodability, it is important to do so in this order. Each observation requires at most $O(D)$ computation as only nodes matching in the current context will change.

Reference [7] assumes a maximum depth D and its coder transmits D symbols verbatim to specify the starting context. Reference [8] extends CTW to an arbitrary depth. Before s_1 there is a semi-infinite past of an additional symbol: $S = \cdots \epsilon\epsilon s_1 s_2 \cdots s_N$. The tree is now $(|A|+1)$ary, but as ϵ is never coded, nodes still store $|A|$ counts. New nodes are added for the full history back to the ϵ, naively generating a large tree with storage complexity $O(N^2)$. However, most deep contexts will be part of a long ''tail'' of a single observation, where $L_e = L_w$. Avoiding explictly storing redundant nodes by retaining a ''tail flag'' and pointer, this optimization gives space complexity only slightly greater than $O(N)$ empirically. [Reference [8] provides a strictly $O(N)$ method that is more tricky to implement.] The EPAPS archive associated with this paper provides a pseudocode document and a software [10] demonstrating the tail-optimized infinite-depth CTW algorithm. Though it appeared simple, we found the correct concrete implementation was not particularly evident from the available literature sources that were essentially theoretical and concentrated exclusively on literal data compression.

Predictors of any quantity estimatable at any node may be weighted by the same formulas. Given an incrementally updated estimator $\hat{q}$ at each node n, define the weighted estimator,

$$\hat{q}_w(n) \triangleq \frac{(2^{-L_e})\hat{q}(n) + \left(2^{-\sum_c L_w}\right)\hat{q}_w(\mathrm{child})}{2^{-L_e} + 2^{-\sum_c L_w}}, \tag{7}$$

now extending codelengths into more general *loss functions* that must be calculated predictively. Reference [11] shows that this prediction method is nearly as good as the best possible pruning of a decision tree with reasonable and mild conditions on the loss functions and the predictors. When $\hat{q}$ is as Eq. (4) and L_e the usual codelength, this recovers CTW, providing an explicit conditional estimator. One could feed this distribution, calculated approximately in standard floating point, into an arithmetic coder. Contrary to Refs. [7,8], arbitrary precision arithmetic (for P_w) is thus not required for an explicit incremental coding with CTW.

CTW's mixture of trees, though it provides an excellent codelength, may be more cumbersome than a good single tree model. ''Context-tree maximizing'' [12] is a nonincremental pruning of the full context tree after having seen all the data. Define $P_p^s \triangleq \frac{1}{2}\max(P_e^s, \Pi_{c\in A} P_p^{cs})$, hence

$$L_p^s \triangleq 1 + \min\left[L_e^s, \sum_c L_p^{cs}\right]. \tag{8}$$

If the first term of the minimum is taken, then the tree is pruned at this node. For tail nodes (those with exactly one observation), $L_p = 1 + L_e$ terminating the recursion. Pruning must be applied to depth first. The ϵ-symbol trick similarly applies here. The description of the tree's topology is transmitted explicitly via the extra bit in Eq. (8). Like CTW, pruning is a universal compression algorithm [9] (though requires two passes) and provides the MDL model (with a reasonable structural prior) over the tree sources. Compression is often only modestly worse than CTW, though never better, as $L_w \leqslant L_p$.

There is one free parameter β. Empirically testing on the dynamical data, varying β by 75% about the value which minimizes the codelength L, changes l by perhaps 2–5%, but the pruned trees usually change little.'' As per the MDL principle, one may minimize L over β (there is almost always a smooth global minimum) provided that β is appropriately encoded with the cost added to L.

This pruning is different from the other context-tree source coding methods called ''state-selecting'' algorithms, e.g., Refs. [13–16]. With those, a codelength criterion similar to Eq. (8) selects a single *encoding node* from all matching nodes from the incrementally constructed tree. The problem is that the methods proven to be universal (e.g., Ref. [14]) are not the ones that are practically useful, the former require excess free parameters or may have poor finite sample performance. In our experience, the latter may have small occupation number pathologies. CTW and the pruned tree version have none of these issues. Ron *et al.* [17] presented a top-down (rather than bottom-up) context-tree estimation algorithm for stochastic sources, providing proven performance bounds, though again at the cost of a number of free parameters.

A pruned context tree represents a stochastic [18] symbolic information source terminal nodes are states. Each state retains a distribution for emission of symbols, inducing state transitions given that symbol and some past history of states. It is not a first-order Markov chain when the identity of the state and its emission distribution is insufficient to fully specify for the transitions to the next state. Figure 2 shows such a context tree. A tree machine's topology may be extended by appropriately grafting new children (with identical $\hat{p}$ as their parents) without altering the predictions of the model; the probability assigned to any sequence remains identical, hence it codes

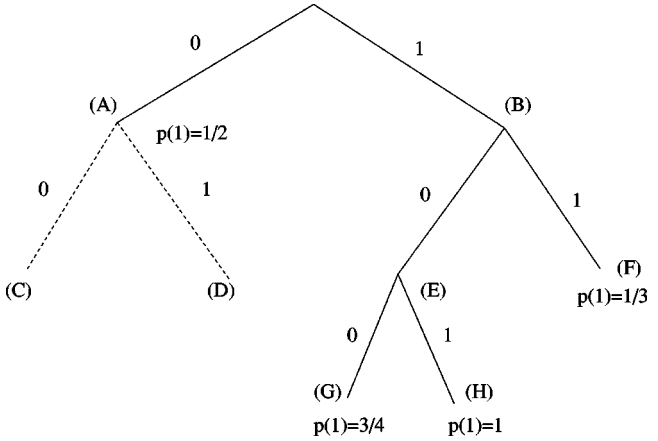

FIG. 2. Solid lines: a tree machine that is is not a Markov chain. Starting from state (A), emission of a "1" results in a history of "01" that only specifies enough history to get to (E), which is not a terminal node. Dotted lines: additional tree node split from (A) so that the full tree machine is equivalent to a Markov model. Starting from (C) or (D) emission of any symbol uniquely identifies the new state.

identically. The criterion for equivalence to a first-order Markov chain is that the topology of all subtrees must be extant if the subtree head is laid over the root. This topological criterion is easy to describe but obscure to implement, thus we present in Ref. [10] a more practical, string-manipulating algorithm to extend trees to the Markov form. Markovization may significantly enlarge the tree. The number of added states is finite, but the worst case expands it to a fully branched tree with depth equaling the maximum of the input tree. Fortunately, the expansion appears to be far less in practice.

Terminal nodes T_i matching contexts at time i are now states (enumerated $1,\ldots,N_s$) with a (sparse) transition matrix for the first-order Markov chain, $\mathbf{P}_{ij} \triangleq P(T_{t+1}=j|T_t=i)$. As ϵ will never be emitted, states with ϵ in contexts are removed. The *invariant distribution* μ of this chain (assumed to have one simply connected component) is the eigenvector with unit value $\mu=\mathbf{P}^T\mu$. One may sum over the appropriate μ_j of the split nodes to find the invariant density of the original tree machine. What transition probability should be used? Once a tree machine or Markov structure has been estimated from the data, generally the better estimate to use for bootstrapping data from the model is $\hat{p}=c/N$, not the smoothed estimate. Encoding new sequences with a fixed model, of course, requires $\hat{p}>0$.

III. APPLICATIONS AND RESULTS

The entropy rate is an obvious statistic to estimate from an observed symbolic sequence. When symbols X are discretized, with an ever finer partition, from an orbit on the invariant density of deterministic dynamics, the Shannon entropy rate $h(\mathcal{X})$ of the symbolic sequence divided by the discretization time step converges to the Kolmogorov-Sinai (ks) entropy h_{KS}, an invariant of a dynamical system. $h_{KS}>0$ defines chaos. Furthermore, for special sorts of partitions, termed "generating," $H(\mathcal{X})=h_{ks}$ even for low precision symbolic alphabets. Practical issues make the limit of infinitely large alphabets inadvisable for ks-entropy estimation from finite sized datasets with as the occupation in any bin tends to zero, making entropy estimation unreliable. Finding generating partitions is difficult enough for known dynamical systems, much less observed symbolic data. Nonetheless, symbolic analyses of observed data are now commonplace and we feel context-tree methods provide a reliable and general means to estimate entropy rates.

With universal compression, the codelength provides an obvious estimator of the entropy rate

$$\hat{h}_{\mathrm{CL}} \triangleq L/N, \tag{9}$$

since $\lim_{N\to\infty}\hat{h}_{\mathrm{CL}}\to h$ by definition. From fundamental theorems [19,20], $\hat{h}_{\mathrm{CL}}$ has non-negative bias because all universal codes have redundancy. We desire entropy estimators that reduce this bias. First, this means using a coder with a small redundancy. For sources compatible with a tree source with k free parameters (including finite-memory Markov), CTW asymptotically performs as $L_{\mathrm{CTW}}\approx hN+(k/2)\log N$ so that $\hat{h}_{\mathrm{CTW}}\approx h+(k/2)(\log N/N)$. By comparison, string-matching algorithms ubiquitous in the digital computer industry (two variants on Lempel-Ziv methods [21]), converge as $\hat{h}_{\mathrm{LZ77}}\approx h+h(\log\log N/\log N)$ and $\hat{h}_{\mathrm{LZ78}}\approx h+O(1/\log N)$, clearly slower than h_{CTW}. We assume $\log N$ redundancy convergence to derive an estimator with lower bias than Eq. (9). A sequential coder provides incremental codelengths: $C(k)\triangleq-\log\hat{p}(s_k)=L(k+1)-L(k)$. $C(k)$ is most conveniently extracted from CTW from the difference of L_w^{λ} before and after an observation. Assuming $L(s_1,\ldots,s_N)\approx h+A\log_e N$, we assert that, on average, $C(k)\approx[\partial L/\partial N]_{k=N}=h+A/k$. Strictly, this is not true for any specific location, but with appropriately averaged sums we assume the equality for present purposes. We define $M\triangleq\Sigma_{k=1}^{N}C(k)(2k/N)=h(N+1)+2A$ and eliminate A to give the estimator

$$\hat{h}_{\mathrm{CL2}}\triangleq\frac{\left(\frac{1}{2}\log_e N\right)M-L}{\left(\frac{1}{2}\log_e N\right)(N+1)-N}. \tag{10}$$

We present a third estimator. The entropy rate of a tree machine or Markov chain with stationary distribution μ is

$$h=\sum_i \mu_i H(P_{i\to *}), \tag{11}$$

with $P_{i\to *}$ the transition distribution, out from state i. Given observations of transitions, we use a standard estimator $\hat{h}$ for iid distributions and weight it by the estimated μ_i: $\hat{h}_{\mathrm{MC}}\triangleq\Sigma_i\mu_i\hat{h}_i$. The plug-in iid entropy estimator, $\hat{h}_{ML}=-\Sigma_k\hat{p}\log_2\hat{p}$ using $\hat{p}_k=c_k/N$, is biased from below. The correction unbiased to $1/N$ (and independent of distribution) has been derived independently numerous times [22]:

TABLE I. Compression performance of algorithm on various simple inputs: random and simple deterministic cases.

System	h	N	$\hat{h}_{\mathrm{SM}}-h$	$\hat{h}_{\mathrm{CL}}-h$	$\hat{h}_{\mathrm{CL2}}-h$
iid $\lvert A\rvert=2$	1	10^4	−0.097	0.00077	−0.00053
iid $\lvert A\rvert=5$	$\ln_2 5$	10^5	−0.386	0.00042	0.00013
Period 3	0	57	0.376	0.226	−0.036
Period 3	0	1824	0.0237	0.0112	−0.000803
Period 4	0	92	0.272	0.193	−0.01910
Period 4	0	2944	0.0156	0.00946	−0.00523

$\hat{h}_{\mathrm{MLC}} \triangleq \hat{h}_{\mathrm{ML}} + (\log_2 e)(M-1)/2N$ with M the number of nonzero $P(i \to *)$, which we estimate as the number of $c_k > 0$ actually observed. Our corrected estimator $\hat{h}_{\mathrm{MCC}}$ weights $\hat{h}_{\mathrm{MLC}}$ by μ. This corrects for the bias at leaves, but not for bias from estimating model structure and thus μ from the finite data. That would appear to be a very difficult quantity to correct in general circumstances; however, via simulation it typically appears to be less important than the bias on the leaf nodes.

We compare results to a recently proposed estimator [23] that operates on an entirely different principle, string matching, the core technology of dictionary-based universal coding algorithms, e.g., Lempel-Ziv [24]. With time index i and integer n, we define Λ_i^n as the length of the shortest substring starting at s_i that does *not* appear anywhere as a contiguous substring of the previous n symbols $s_{i-n}, \ldots, s_{i-1}$. The string-matching estimator is $\hat{h}_{\mathrm{SM}} \triangleq \log n/(n^{-1}\Sigma_{i=1}^n \Lambda_i^n)$ Since $\hat{h}_{\mathrm{SM}}$ does not arise from an actual lossless code, it is not necessarily biased from above, unlike h_{CL}. It is not necessarily unbiased for any finite n either, however. To implement it with N observed symbols, we first remove a small number Δ of symbols off the end and then split the remaining into two halves. String matching begins with the first element of the second half, $(N-\Delta)/2+1$, and examines the previous $n=(N-\Delta)/2$ characters. The length Δ excess padding is necessary to allow string matches from the end locations of the match buffer. Δ is presumed to be a few times longer than the expected match length $\langle \Lambda \rangle \approx \log n/h$.

Table I shows results on simple systems. The context-tree estimators perform well in all cases, whereas the $\hat{h}_{\mathrm{SM}}$ performs surprisingly poorly on high entropy cases. Even on deterministic systems where one might expect string matching to prevail, context-tree methods are superior. The pruned context tree induces a deterministic state machine on periodic data; $\hat{h}_{\mathrm{MC}}$ is thus zero on these sets. The next system is a first-order Markov chain with $|A|=3$, with two cases, the state directly emitted and not. The transition matrix is

$$\mathbf{M} = \begin{bmatrix} 0 & 1/3 & 2/3 \\ 1/4 & 0 & 3/4 \\ 2/10 & 0 & 8/10 \end{bmatrix}, \quad h(\mathbf{M}) \approx 0.7602. \tag{12}$$

We estimate entropies from finite samples of simulation. Either the state itself is emitted ($|A|=3$), or 0 or 1 is emitted ($|A|=2$) depending on whether the first or second nonzero transition is taken in each row. In the first case, the state structure is directly observable (and the pruning method finds the first-order Markov structure). In the the latter case—a hidden Markov model—approximate proxies for the state are automatically reconstructed by the modelers. Figure 3 shows estimators on resimulations. The context-tree methods outperform the match length estimator: the resampled distribution of estimates from tree methods includes the true value, whereas for the $\hat{h}_{\mathrm{SM}}$ truth lies significantly outside its distribution. The results are similar on other artificial chains, both time reversible and not.

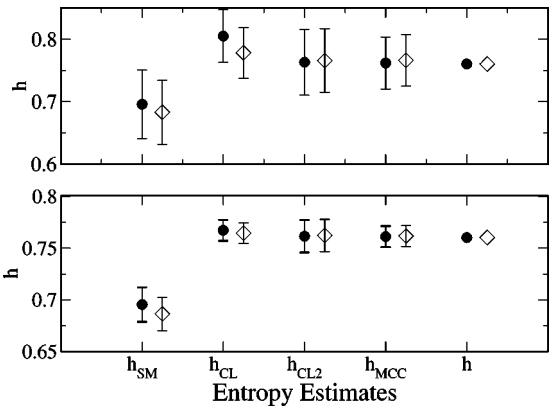

FIG. 3. Entropy estimators (mean ± sample standard deviation) evaluated from 100 samples of Markov chain $\mathbf{M}_1$ for $L=500$ (top), $L=5000$ (bottom). Solid circles are for $|A|=3$. From left to right, estimators are $\hat{h}_{\mathrm{SM}}, \hat{h}_{\mathrm{CL}}, \hat{h}_{\mathrm{CL2}}, \hat{h}_{\mathrm{MCC}}$ followed by the actual entropy rate.

The next example is the logistic map $x_{n+1}=f(x_n)=1-ax_n^2$. In continuous space, the map is so simple, it is never a challenge to model, but once discretized it is not trivial. Symbolizing at the critical point $x=0$ gives a generating partition for $0<a\leqslant 2$, and by the Pesin identity, $h=h_{KS}=\lambda$ with λ the Lyapunov exponent on an ergodic trajectory: $\lambda=\lim_{N\to\infty}\Sigma_{i=1}^N \log_2|f'(x_i)|$. We estimate h_{KS} via λ on a very long (10^6) trajectory. The lower panel of Fig. 4 shows results in a generic chaotic region, $a=1.8$. Compared to the

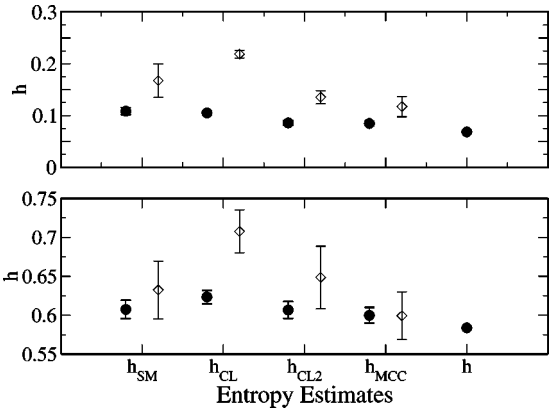

FIG. 4. Entropy estimators (mean ± sample standard deviation) evaluated from 100 samples of discretized logistic map $x(n+1)=1-ax(n)^2$, $a=1.405$ (upper), and $a=1.8$ (lower). $N=5000$ (circles) and $N=500$ (diamonds).

previous systems, the estimated model structure is substantially more complex (more important deep nodes) hence the difference between $\hat{h}_{\mathrm{CL}}$ and $\hat{h}_{\mathrm{CL2}}$ or $\hat{h}_{\mathrm{MCC}}$ is larger since $\hat{h}_{\mathrm{CL}}$ contains more model redundancy. The match length estimator performs well here, but the bias is still smaller with $\hat{h}_{\mathrm{MCC}}$. The upper panel of Fig. 4 shows results at $a=1.405$, only slightly above the period-doubling accumulation point, i.e., the ''edge of chaos.'' Here the entropy rate is positive but small, and thus the effective depth of the tree (complexity) grows much larger. Again $\hat{h}_{\mathrm{MCC}}$ provides the least biased estimate, somewhat surprising again as the dynamics exhibit many long near repetitions where string matching should be good.

Good compression performance implies an upper bound on relative entropy between the true and estimated distributions, hence good compressors are good modelers. The per-symbol coding redundancy $N^{-1}\rho = L/N - h$ is an estimate of the Kullback-Liebler distance $D(p\|q)$ between the true and the estimated probabilities, and thus the estimated and the true stochastic dynamical systems. Most applications of the empirical symbolic dynamics in the literature explicitly or implicitly use fixed-order Markov models, e.g., estimations of fixed-length word probabilities. The MDL context tree provides an explicit symbolic predictive model from the data and should be substitutable for fixed-order Markov models in most circumstances and usually provide equal or superior performance. We compare redundancy of the variable context trees to fixed-order Markov models, i.e., trees where all probability estimates occur at a fixed depth D. Its codelength is the sum of L_e at all extant nodes of the full depth D tree, plus that of the one structure parameter D, which the classical Elias delta code [25] may encode in no more than $L_{\mathrm{integer}}(D) = 1 + \log(D+1) + 2\log\log 2(D+1)$ bits. The Markov model codelength is thus $L_{\mathrm{MM}}(D) = L_{\mathrm{integer}}(D) + \Sigma_{\mathrm{Nodes}} L_e$. We include the startup costs similarly with the ϵ construction. This is a fair codelength because we could transmit the dataset with this number of bits. We thus also claim a MDL model selection criterion for fixed-order chains using sequential coding. If one wishes to examine distributions of length W words, then with $D^* = \arg\min_D L_{\mathrm{MM}}(D)$, the appropriate W is D^*+1.

Table II shows average redundancies from 100 samples of the logistic map for $a=1.8$, in a generic chaotic regime, and $a=1.405$ barely above the chaotic transition. Estimated from a long ergodic sample via the Lyapunov exponent the entropy rate is $h_{ks}=0.5838$ bits/symbol for $a=1.8$ and $h_{ks}=0.068\,050$ for $a=1.405$. The weighting tree estimator (CTW) is superior, followed closely by the MDL tree then the Markov chain with the optimal order D^*. For fixed-order Markov chains, D^* depends significantly both on N (not shown) and the structure of the dataset. When $a=1.405$, the typical D^* found is surprisingly deep, on an account of the significant stretches of ''near periodic'' orbits exhibited by the map close to the edge of chaos. For instance, $D^*=24$, implies a naive 2^{24} bins, far larger than the number of data. A typical ''rule of thumb'' for guessing at the order *a priori* by requiring a certain estimated minimum bin occupation such as $D \approx \log_2(2N)$ may be quite flawed. Selecting D with the MDL principle and sequential coding is a significant win, but it takes little extra effort to find the MDL variable-order context tree, which almost always outperforms the fixed-order tree.

TABLE II. Coding redundancy per symbol for logistic map. Mean and sample standard deviation over 100 independent samples of $N=10\,000$. Lower redundancy implies a superior probability model and/or a less complex model.

Compressor	$a=1.8$	$a=1.405$
CTW	0.0270 ± 0.0067	0.0253 ± 0.0028
MDL tree	0.0304 ± 0.0065	0.0275 ± 0.0028
Markov optimal	0.0473 ± 0.0070	0.0437 ± 0.0029
Markov $D=4$	0.1221 ± 0.0062	0.1156 ± 0.0004
Markov $D=8$	0.0478 ± 0.0070	0.0815 ± 0.0020
Markov $D=12$	0.1093 ± 0.0088	0.0593 ± 0.0088
Markov $D=16$	0.2094 ± 0.0102	0.0455 ± 0.0102
`gzip -9`	0.0990 ± 0.0043	0.2530 ± 0.0048
`bzip -9`	0.0752 ± 0.0041	0.3222 ± 0.0079

Unlike string matching, context-tree methods provide an explicit probability model and thus we may *simulate* from it. We fix the estimated model after all the input data have been observed, and resimulate from its probability distribution. That may be full CTW via Eq. (7) or, more easily, the state machine defined by the pruned tree or Markovized version thereof. We show examples of their ability to capture and to replicate nontrivial features of some observed streams. We compare various statistics from resimulations from an estimated model to those evaluated on a long sequence from the actual dynamical system. We demonstrate the successful generalizing modeling power for trees beyond simply minimizing the compression rate (for which they are explicitly designed) and that modeling and simulation appear to add few artifacts. We do not claim that most natural dynamical system ''are'' strictly in the class of finite-depth tree-structured information sources (tree machines), but suggest that such models may often be good approximations given a stable statistical estimation method. Simulation—using a MDL-pruned tree as a stochastic information source—is simple and rapid. We initialize the history by sampling a state from μ and recording its implied context. Iteratively, emit a symbol according to $\hat{p}$ at each deepest matching node, append to the buffer, and find the next context. One may also simulate even more rapidly from the equivalent state transition graph knowing **P**.

An immediate question is ''what distribution is appropriate for emitting symbols given a state''? We recommend using the naive estimator, i.e., Eq. (4) with $\beta=0$, so that the symbolic topology includes only transitions actually observed in the input data. There may be forbidden transitions, and in a resimulation it would generally be wrong to create them, which would happen with $\beta>0$ or generally any probability estimator that assigns a finite probability to a never before seen symbol emission. This is the moral equivalent of resimulating from a discrete iid distribution by randomly choosing *with replacement* from the set of observed symbols and no others. The density μ with the observed counts in **P** is

the limit of such a simulation. It may seem inconsistent to use $\beta>0$ for compression and $\beta=0$ for simulation, but it is not unreasonable. For the first case, one is computing a quantity that is nearly the log likelihood of the observed data and the estimated parameters give only a prior distribution for parameters before data have been seen. For the second, one maximizes the *post-hoc* likelihood of the emission parameters of the *given* data already observed and the topological structure already estimated, i.e.; a classical statistical procedure. A universal coding algorithm must assign a probability to *any possible* string in the alphabet, but a simulation is obligated to assign positive probability only to those strings that may be emitted from the model. Nevertheless, the issue of whether to use $\beta=0$ or $\beta>0$ for simulation has a universally correct answer, as choice is a matter of statistical assumption and viewpoint.

We may represent a symbolic sequence $s_i \in\{0,1,\ldots,|A|-1\}$ as a sequence of points in the symbolic plane $(x_i,y_i)\in[0,m]\times[0,m];m=(|A|-1)/(\alpha|A|-1)$ with

$$(x_i,y_i)\triangleq\left(\sum_{k=1}^{\infty}\frac{s_{i+1-k}}{\alpha^k|A|^k},\sum_{k=1}^{\infty}\frac{s_{i+k}}{\alpha^k|A|^k}\right).$$

We define the projection of a symbolic sequence on to this plane as a *symbologram*. The *x* coordinate of a point represents the past history with smaller deviations corresponding to more ancient symbols, and the *y* coordinate the future in the similar way. A symbolic information source produces a characteristic geometrical set, the symbologram, in rough analogy to the invariant set of a dynamical system in continuous space. For $\alpha=1$, there is guaranteed to be a unique correspondence between the points in the symbolic plane and the symbol sequences [26]. Furthermore, for $\alpha=1$, the fractal information dimension D_1 of the symbologram scales with the Shannon entropy: $D_1=2h/\log|A|$. The symbologram summarizes both the stationary distribution of the symbols and the conditional predictive distribution, i.e., the evolution law. Figure 5 shows $\alpha=1$ symbolograms for the logistic map, and a simulation from a tree estimated from a different sample of length 1000 of the original system. There is obviously a quite substantial resemblance between the apparent invariant densities between them, meaning that the probability of seeing a string in the original is quite close to that assigned to it by the tree model. That, of course, is the goal of source coding. Estimated from $N=10\,000$, the figures are nearly visually identical.

One might evaluate a Kolmogorov-Smirnov (KS) test for significance in the difference in cumulative distributions, given 10 000 points the KS test does indeed accept the null hypothesis for *x* or *y*. In practice, though, rejection is almost certain asymptotically. Given sufficiently long samples (and one may simulate arbitrarily long), there will be sufficient data that the test's null hypothesis is violated even though the deviation in the cumulative distributions is small in an absolute sense. Only if the model gives *exactly* the same probability assignment as truth would the test always be accepted, and that would occur only in the unlikely case where the system is a tree machine and the estimated probabilities happened to equal reality exactly. Statistical significance does not necessitate a large modeling deviation, however, and universal coding provides a guarantee that the model will converge to the truth in some useful measure.

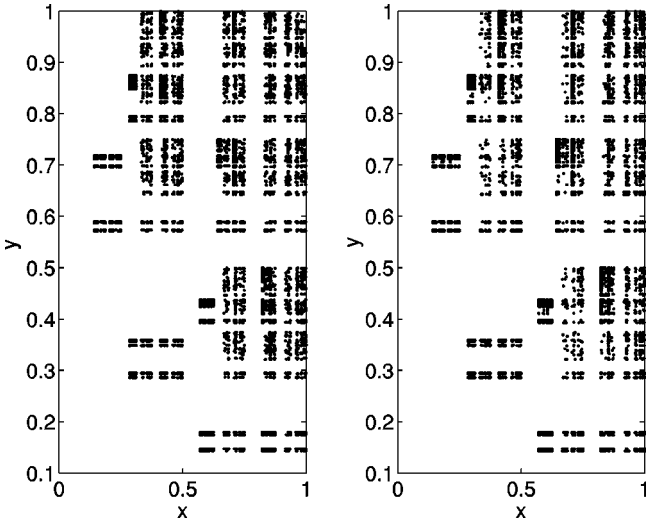

FIG. 5. Symbologram of a sample of discretized logistic map time series at $a=1.8$ (left), and from simulation from MDL tree model estimated on a distinct $N=1000$ length dataset.

We turn to a more complicated system, discretized samples from the "Lorenz 1984" attractor; a tiny geophysical model with attractor dimension $d\approx2.5$ [27]. Now the discretization is not the (unknown) generating partition, yet the projection down to a symbol stream still produces a nontrivial stochastic symbolic information source that we wish to model. Because it was sampled from a continuous ordinary differential equation and not a map, the entropy is rather low, $h/\log_2|A|\approx0.24$. Consequently, the symbologram with $\alpha=1$ is thus rather sparse and we thus display in Fig. 6 the original and simulacrum symbolograms with $\alpha=1/2$. The estimated context tree had approximately 200 terminal nodes; 400 after Markovization.

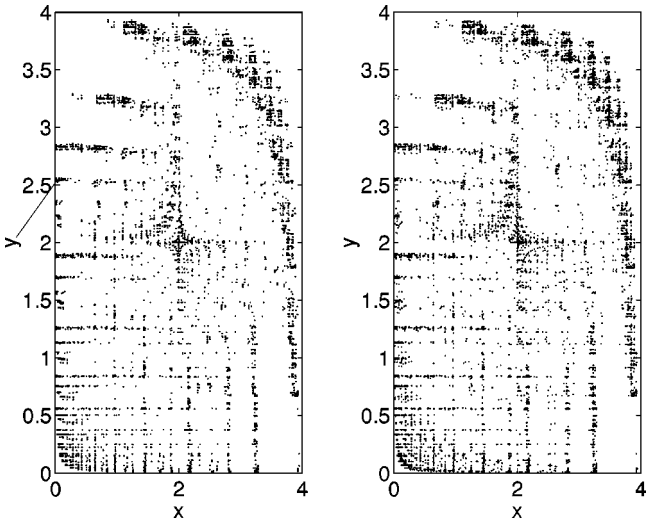

FIG. 6. Symbologram ($\alpha=1/2$) of output from Lorenz 1984 [27] model heavily discretized (left), and from simulation of context tree estimated from $N=10\,000$ symbols (right).

TABLE III. Kolmogorov-Smirnov test acceptance probabilities on observed distribution of recurrence times for various strings, comparing 500 000 symbols from logistic map to a simulation from a MDL tree estimated on 10 000 symbols. Recurrence distributions of most strings matched, except for consecutive 1's. The lower-right block of strings involves a deterministic transition so that sequences of $L=7,8,9$ strings have identical distributions as 0110100 is always followed by 10.

String	p value	String	p value
110	0.6048	101	0.4112
1101	0.5835	1010	0.4075
11011	0.4690	10101	0.5583
110111	0.1122	101010	0.9098
1101111	0.3081	1010101	0.7990
11011111	0.3172	10101010	0.3472
111	0.2270		
1111	0.0424	01101	0.2376
11111	0.7474	011010	0.1633
111111	0.0014	0110100	0.4412
1111111	5.8361×10^{-5}	01101001	0.4412
11111111	0.0541	011010010	0.4412
111111111	0.0443	0110100101	0.1309

Approximately matching symbolograms means that probabilities of words of consecutive symbols are successfully replicated. We now examine a more complicated task, matching a statistical distribution not closely modeled by the fitting procedure. From a long sample, we extract the *recurrence time* of some symbolic string, i.e., the time intervals between (nonoverlapping) observations of any arbitrary symbolic word. The distribution of recurrence times quantify longer-range dependences. Again for the logistic map at $a=1.8$, we estimated a tree from 10 000 observations and simulated a series of length 500 000, observed the distribution of recurrence times and compared to that observed on an identical-length set from the simulated dynamical system. We compare the acceptance likelihood from the Kolmogorov-Smirnov test, which compares the empirically observed cumulative distributions. The previously mentioned issue still applies, but the results are sufficient to show trends and provides assurance that the model does not necessarily fail to replicate the long-term as well as the short-term features. Table III shows KS rejection p values for recurrence times of some arbitrarily selected strings. As the test assumes independence, after the second match of a pair is found (their time difference being an observation), we apply a length 500 dead zone before searching for the next string match to begin a pair.

IV. DISCUSSION AND SUMMARY

The context-tree construction and, in particular, the equivalent Markov chain may provide a good estimate for the ''ϵ machine'' of Crutchfield and Shalizi [28]: a representation of a process consisting of the minimal *causal states* and their transitions. Their goal is to define firmly a useful and robust notion of ''complexity'' distinct from randomness; a quantification of the intuitive notion that both very deterministic (fixed point and periodic motions) and very random (iid noise) behaviors imply low complexity with the ''interesting dynamics'' possessing higher complexity. According to Ref. [28], a causal state is an equivalence class over histories so that all members of the class have the same conditional probabilistic distribution of observables for the infinite future. The complexity is then the Shannon entropy of the distribution of causal states observed weighted by the measure of the process. It quantifies ''how much information do I need to specify to set the current state so that I may optimally predict the future,'' i.e., the quantity of historical information stored in the system. White noise processes have one state, periodic processes have only as many states as the period, and thus low complexities. Complex processes have, of course, more internal states and thus complexity. Our point here is that the Markovized tree states satisfy the criteria of Ref. [28] for being a causal state, and we have a robust algorithm to find them from the observed time series. Furthermore, the minimum description length principle gives an explicit and attractive balance between ''prescience'' and ''complexity,'' which is essential to the inference of minimal causal state machines from finite amounts of data. The complexity measure is $C_\mu=-\Sigma_{i=1}^{N_s}\mu_i\log_2\mu_i$ over stationary probabilities μ_i for those states automatically discerned by the context tree. As expected, C_μ measure is zero for the logistic map for $a=2$ (when the signal becomes random binary) and increases as the bifurcation parameter decreases, and diverges at the period-doubling accumulation point, the edge of chaos, where the system can no longer be modeled by the regular languages of finite depth context trees.

Crutchfield and Young [29] and similarly, Carassco and Oncina [30], considered extracting probabilistic automata from observed data. Their constructions are quite different, relying on making equivalence classes among nodes in the *prefix tree* (depth is forward in time) of the observed sequences. Compared to the present MDL tree method there are significantly more free parameters, including an unknown ''depth'' over which one considers whether nodes form equivalent states or not. We feel that determining this depth automatically (and thus estimating word probabilities) is precisely the most difficult part of model inference. Some sort of criterion balancing statistics with complexity/depth is necessary, in addition. Checking the more general node equivalence condition requires more computational effort than context-tree pruning, but it does consider a wider class of models as internal nodes ''across'' branches may be found to be equivalent rather than only children versus parents. This might mean that a smaller machine could be found, implying lower complexity C_μ. Perhaps a MDL selection criterion could be derived for a more general class of context-type models providing the best of both methods.

We suggest some examples of how the context-tree methods presented here could potentially improve a number of existing algorithms and analyses of data in the literature, though it is beyond the scope of the present work to actually perform all these analyses with new methods and quantify the improvement or lack thereof. In general, any situation

requiring a conditional entropy estimate should be amenable to the context-tree estimator, if the dynamics are sufficiently well described by a regular language or an approximation thereof. Essentially, the influence of the past has to decrease sufficiently quickly. There are examples of dynamical systems, which do not appear to satisfy this criterion in various forms, but in general any data analysis on sufficiently nonstationary dynamics is dubious from the start. The boundary of exactly which classes of dynamics are efficiently estimatable with context trees is not presently known.

Fraser [31] studied information-theoretic quantities in observed noisy chaotic systems and pioneered the now-standard use of mutual information as a criterion for selecting time delays in continuous state-space reconstruction, given $x(t)$ one estimates mutual information $I(x_{t+\delta t};x_t) = H(x_{t+\delta t}) - H(x_{t+\delta t}|x_t)$. The first local minimum is deemed to be a good reconstruction parameter. Generalized beyond two scalars, the *differential redundancy*

$$R'(x_{t+\delta t}|\mathbf{x}_t) \triangleq I(x_{t+\delta t};\mathbf{x}_t) = H(x_{t+\delta t}) - H(x_{t+\delta t}|\mathbf{x}_t) \tag{13}$$

is the amount of information in the new observation $x_{t+\delta t}$, which is predictable from knowledge of the ''current state'' given by the vector $\mathbf{x}_t = [x_t, x_{t-1}, x_{t-2}, \ldots]$. Fraser shows [31] that for data observed from a chaotic dynamical system as the discretization intervals approach zero (increasing alphabet) and the dimension of the conditioning vector approaches infinity, $R'_{\delta t} \approx A - (\delta t) h_{KS}$ with h_{KS} the Kolmogorov entropy rate, a dynamical invariant of the continuous system. This formulation distinguishes the entropy caused by independent noise from chaos. Such noise is unpredictable but will have $h_{KS} = 0$, because $H(X_{t+1}) = H(X_{t+1}|X_t)$.

Although we do not presently deal with the issue of discretization to symbols s_i, we point out that it is trivial to estimate the conditional entropy δ steps ahead instead of one by replacing s_{i+1} with $s_{i+\delta}$ in Eq. (2) giving appropriate $\hat{h}_{\mathrm{CL}}$ and $\hat{h}_{\mathrm{CL2}}$ (but not $\hat{h}_{\mathrm{MC}}$!) estimators for the second conditional entropy term in Eq. (13). The first term is the zero-order entropy estimate, e.g., L_e/N at the root note of the tree. We can thus estimate $R'(x_{t+\delta t}|\mathbf{x}_t)$ safely taking the limit of the infinite conditioning context with arbitrary depth CTW. This circumvents the usual exponential explosion of bins, which practically limited the direct application of histogram-type information estimators in Ref. [31] beyond two dimensional.

We may similarly estimate the conditional entropy between the information sources, $h(R|S)$ provided simultaneous observations (s_i, r_i), and replacing s_{i+1} with $r_{i+\delta}$ in Eq. (2). This important generalization resulting from an explicit model is not easily available to string-matching methods. Reference [32] addresses detecting whether or not two signals, e.g., $x(t)$ and $y(t)$ are projections of the same dynamical system. When it is so, the conditional entropy $h(y_{t+\delta t}|x_t)$ has a minimum near $\delta t = 0$. In Ref. [32] the authors use fixed-length words and estimate entropies naively. We suggest using tree estimators in the symmetrized cross entropy

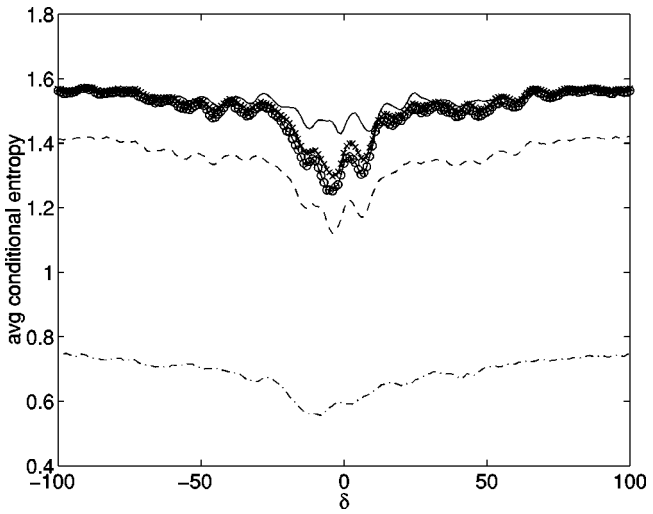

FIG. 7. Symmetrized cross entropy estimates flow-precision noisy x and z coordinates from Lorenz 1984 [27] model, 20 000 samples at $\delta t = 0.16$. Curves without symbols are with fixed-order Markov with depth $D = 4,8,12$. Curves with symbols are $\hat{h}_{\mathrm{CL}}$ and $\hat{h}_{\mathrm{MCC}}$. The context tree methods correctly find $h \approx \ln_2 |A|$ for $|\delta| \to \infty$, still preserving a comparatively deep relative dip at $\delta \approx 0$, demonstrating dynamical state correlation.

$$\overline{h}(\delta) = \frac{1}{2}[\hat{h}(r_{i+\delta}|s_i, s_{i-1}, \ldots) + \hat{h}(s_{i+\delta}|r_i, r_{i-1}, \ldots)].$$

Our example is again the Lorenz 1984 model. We examine correlation between x and z components, sampled at $\delta t = 0.16$. To ensure a challenge white Gaussian noise of standard deviation half of the clean time series was added, subsequently symbolized with $|A| = 3$ using equiprobable bins. Figure 7 compares results from context-tree methods to entropy estimates from fixed-order Markov models (like a fixed word length). An arbitrary word length results in a significant systematic bias. The shallow word $(D = 4)$ correctly finds conditional entropies tending to the unconditioned entropy $\ln_2 3 \approx 1.58$—no significant predictability—for large $|\delta|$, but has insufficient discriminating power to discern much correlation near $\delta t \approx 0$. Longer word lengths produce significant overall biases in the level. Context-tree methods best show power without large systematic bias.

The present authors previously used [16] a state selecting context-tree source modeler to test for dynamical stationarity by combining traditional frequentist tests at the ''encoding nodes.'' An undesirable artifact is that early points get encoded to shallowly because the choice of encoding node is made sequentially. The *post-hoc* MDL tree is a cleaner and reliable method of finding the appropriate states. In Ref. [33], Daw *et al* examined statistics comparing ''forward'' versus ''backward'' observations of symbolic words formed from a time series. Linear Gaussian processes and static nonlinear transforms thereof produce statistically time-reversible signals. Thus the observation of statistically significant temporal irreversibility, as found in chaos or other nonlinear dynamics, precludes that class of signals, the same class which is the null hypothesis of many ''surrogate data'' resimulation

methods. Reference [33] gives a direct statistic and test for reversibility, bypassing the need for Monte Carlo simulation. However, it used a fixed word length that was an arbitrary choice. Here we suggest using the MDL tree to find the proper choice of words, the terminal node contexts, and continuing with the procedure of Ref. [33]. One would estimate the tree structure from data comprising the time series run both forward and backwards. The MDL tree finds especially predictable nodes; these states correspond to more clear evidence of ''dynamics'' and hence we expect them to be more irreversible. Cover and Thomas [20] provide an explicit formula (4.40) for the entropy rate of a time-reversible Markov chain. As an alternate method for detecting irreversibility, one could compare the $\hat{h}_{\mathrm{MC}}$ on observed data to that expected under the reversible hypothesis (where any time-reversible Markov chain is equivalent to a random walk on an *undirected* but weighted graph). The transition probabilites on the discovered provides all the quantities needed.

In a biological application, Steuer *et al.* [34] examined the entropy rates derived from a binary symbolization of interspike intervals measured in paddlefish and crayfish mechanoreceptors. In this system predictability is modest, but the authors did find that certain higher-order Markov models (e.g., order 4 or 5) displayed superior predictability (lower conditional entropy) than the low-order models, and that this excess predictability was localized to a small number of symbolic contexts found to be reasonably experimentally stable across animal subjects. The authors had to validate the choice of Markov order with Monte Carlo surrogate data techniques. A context-tree method would avoid the tedium of some validation simulations, moreover the MDL tree algorithm intrinsically attempts to pull out of the data from those particular contexts that give excess predictability. Their choice of Markov order was based on comparing global statistics versus resimulation, but the object of desire is the indentification of individual states that are particularly prescient.

Lesher [35] symbolized interspike intervals in from a lamprey neuron via a more interesting technique, i.e., fitting, in continous space, a vector of successive observations to locally linear two-dimensional map and then reducing to discrete symbols the eigenvalue plane that results with the symbols corresponing to, e.g., ''stable node,'' ''stable focus,'' ''unstable focus,'' ''direct saddle,'' ''flip saddle,'' etc. There was a significant correspondence between interesting qualitative changes in the observed time series dynamics and the symbol transitions, but the authors found that a first-order model is insufficient, and suggested using a hidden Markov model in future investigation. We suggest that in situations where there is no obvious intuitive external guidance for the effective size of the structure or complexity of the data, an adaptive context tree could be superior. Classical estimation of hidden Markov models by expectation maximization requires the structure be designed *a priori* and then the transition parameters estimated, though there are now adaptive (but often slow) techniques in the literature.

We have introduced the use of modern source modeling techniques traditionally used for data compression for the purpose of analyzing observed symbolic time series from dynamical systems. The context-tree weighting and maximizing algorithms are theoretically attractive modeling techniques with good performance, one free parameter, and efficient implementation. Given a tree machine, we have an algorithm to convert it to an equivalent first-order Markov chain that opens additional opportunitites. We demonstrate a number of good performing entropy estimators and then show the success of the tree methods on modeling observed data via resimulation.

[1] J. Rissanen, *Stochastic Complexity in Statistical Inquiry* (World Scientific, Singapore, 1989); A. Barron, J. Rissanen, and B. Yu, IEEE Trans. Inf. Theory **44**, 2743 (1998).

[2] K. Judd and A. I. Mees, Physica D **82**, 426 (1995).

[3] J. Rissanen and G. G. Langdon, IEEE Trans. Inf. Theory **27**, 12 (1981); I. H. Witten, R. Neal, and J. G. Cleary, Commun. ACM **30**, 520 (1987).

[4] R. E. Krichevsky and V. K. Trofimov, IEEE Trans. Inf. Theory **28**, 199 (1981).

[5] Y. M. Shtarkov, T. J. Tjalkens, and F. M. J. Willems, Probl. Inf. Transm. **33**, 17 (1997). Although the Krischevsky-Trofimov (KT) estimator has nice and analytically computable asymptotic properties, other estimators may be superior for finite length observed data. We explored a number of ansatzes used by the text compression community but none were consistently superior the KT estimator (4) allowing β to vary as an optimization parameter. In empirical experimentation $\beta = 1/|A|$ gives results reasonably close to the minimum codelength.

[6] J. Rissanen, IEEE Trans. Inf. Theory **30**, 629 (1984).

[7] F. M. Willems, Y. M. Shtarkov, and T. J. Tjalkens, IEEE Trans. Inf. Theory **41**, 653 (1995).

[8] F. M. Willems, IEEE Trans. Inf. Theory **44**, 792 (1998).

[9] T. J. Tjalkens (private communication).

[10] See EPAPS Document No. E-PLEEE8-66-094211 for software, written in C, Perl, and Fortran 95, to perform the context-tree weighting, context-tree maximizing, conversion to Markov chain, estimation of the stationary probability density, and the various entropy estimates, is available in urlftp://lyapunov.ucsd.edu/pub/context/ and with the AIP Electronic Physics Auxiliary Publication service. A direct link to this document may be found in the online article's HTML reference section. The document may also be reached via the EPAPS homepage http://www.aip.org/pubservs/epaps.html or from ftp.aip.org in the directory/epaps. See the EPAPS homepage for more information. Documentation giving a pseudocode for implementatiion of the context-tree weighting, maximizing, and Markovization algorithms is also provided.

[11] D. P. Helmbold and R. E. Schapire, Mac. Learn. **27**, 51 (1997).

[12] P. A. J. Volf and F. M. J. Willems, in *Proceedings of the 1995 IEEE International Symposium on Information Theory* (IEEE, New York, 1995).

[13] J. Rissanen, IEEE Trans. Inf. Theory **29**, 656 (1983); S. Bun-

ton, in *Proceedings of the Data Compression Conference 1997*, edited by J. A. Storer and M. Cohn (IEEE Computer Society Press, Los Alamitos, 1997), pp. 32–41.
[14] M. J. Weinberger, J. J. Rissanen, and M. Feder, IEEE Trans. Inf. Theory **41**, 643 (1995).
[15] T. Schurmann and P. Grassberger, Chaos **6**, 414 (1996).
[16] M. B. Kennel and A. I. Mees, Phys. Rev. E **61**, 2563 (2000).
[17] D. Ron, Y. Singer, and N. Tishby, Mach. Learn. **25** 117 (1996).
[18] The computer science literature sometimes deems this a ''deterministic'' finite-state automaton, meaning that the identity of the subsequent state is determined completely given the previous state and emitted symbol.
[19] C. E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948).
[20] T. Cover and J. Thomas, *Elements of Information Theory* (Wiley Interscience, New York, 1991).
[21] A. D. Wyner, J. Ziv, and A. J. Wyner, IEEE Trans. Inf. Theory **44**, 2045 (1998).
[22] Miller & Madow, Air Force Cambridge Research Center Technical Report No. 54-75, 1954 (unpublished); B. Harris, Colloquia Mathematica Societatis Janos Bolya, p. 323 (1975); H. Herzel, Syste. Anal. Model Sim. **5**, 435 (1988). W. E. Caswell and J. A. Yorke, in *Dimensions and Entropies in Chaotic Systems*, edited by G. Mayer-Kress (Springer-Verlag, Berlin 1986).
[23] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner, IEEE Trans. Inf. Theory **44**, 1319 (1998).
[24] J. Ziv and A. Lempel, IEEE Trans. Inf. Theory **23**, 337 (1977).
[25] P. Elias, IEEE Trans. Inf. Theory **21**, 194 (1973).
[26] We conjecture that the unique correspondence remains for $\alpha < 1$ as long as $h \leqslant \ln(\alpha|A|)$.
[27] E. N. Lorenz, Tellus, Ser. A **36A**, 98 (1984). The model is $dx/dt = -y^2 - z^2 - a(x-F), dy/dt = xy - bxz - y + 1, dz/dt = bxy + xz - z, a = 1/4, b = 4, F = 8$. We recorded the x coordinate every $\delta t = 0.2$ and discretized by histogramming.
[28] J. P. Crutchfield and C. R. Shalizi, Phys. Rev. E **59**, 275 (1999).
[29] J. P. Crutchfield and K. Young, Phys. Rev. Lett. **63**, 105 (1989).
[30] R. C. Carassco and J. Oncina, in *Proceedings of the 2nd International Colloques on Grammatical Inference and Applications* (Springer, Berlin, 1994), pp. 139–152.
[31] A. M. Fraser, IEEE Trans. Inf. Theory **35**, 245 (1989); A. M. Fraser and H. L. Swinney, Phys. Rev. A **33**, 1134 (1986).
[32] M. Lehrman, A. B. Rechester, and R. B. White, Phys. Rev. Lett. **78**, 54 (1997).
[33] C. S. Daw, C. E. A. Finney, and M. B. Kennel, Phys. Rev. E **62**, 1912 (2000).
[34] R. Steuer, W. Ebeling, D. F. Russell, S. Bahar, A. Neiman, and F. Moss, Phys. Rev. E **64**, 061911 (2001).
[35] S. Lesher, L. Guan, and A. H. Cohen, Neurocomputing **32-33**, 1073 (2000).