Constructing finite-context sources from fractal representations of symbolic sequences

Peter Tiňo* **Georg Dorffner†**
Austrian Research Institute for Artificial Intelligence
Schottengasse 3
A-1010 Vienna, Austria
Phone: +43-1-5336112-15
Email: petert,georg@ai.univie.ac.at

Abstract

We propose a novel approach to constructing predictive models on long complex symbolic sequences. The models are constructed by first transforming the training sequence n-block structure into a spatial structure of points in a unit hypercube. The transformation between the symbolic and Euclidean spaces embodies a natural smoothness assumption (n-blocks with long common suffices are likely to produce similar continuations) in that the longer is the common suffix shared by any two n-blocks, the closer lie their point representations. Finding a set of prediction contexts is then formulated as a resource allocation problem solved by vector quantizing the spatial representation of the training sequence n-block structure. Our predictive models are similar in spirit to variable memory length Markov models (VLMMs). We compare the proposed models with both the classical and variable memory length Markov models on two chaotic symbolic sequences with different levels of subsequence distribution structure. Our models have equal or better modeling performance, yet, their construction is more intuitive (unlike in VLMMs, we have a clear idea about the size of the model under construction) and easier to automize (construction of our models can be done in a completely self-organized manner, which is shown to be problematic in the case of VLMMs).

1 Introduction

Statistical modeling of complex sequences is a fundamental goal of machine learning due to its wide variety of applications (Ron, Singer, & Tishby, 1996): in genetics (Prum, Rodolphe, & deTurkheim, 1995), speech recognition (Nadas, 1984), finance (Bühlman, 1998), or seismology (Brillinger, 1994).

One of the models for sequences generated by stationary sources, assuming no particular underlying mechanistic system, are Markov models (MMs) of finite order (Bühlmann, 1997). The only implicit assumption made is about the finite memory of the process.

*also with *Department of Computer Science and Engineering, Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovakia*

†also with *Dept. of Medical Cybernetics and Artificial Intelligence, Univ. of Vienna*

These statistical models define rich families of sequence distributions and give efficient procedures for both generating sequences and computing their probabilities. However, MMs can become very hard to estimate due to the familiar explosive increase in the number of free parameters (yielding highly variable estimates) when increasing the model order. Consequently, only low order MMs can be considered in practical applications.

Approaches proposed in the literature (Ron, Singer, & Tishby, 1996; Laird, & Saul, 1994; Nadas, 1984; Rissanen, 1983; Weinberger, Rissanen, & Feder, 1995; Willems, Shtarkov, & Tjalkens, 1995) to overcome the curse of dimensionality in MMs share the same basic idea: instead of MMs consider variable memory length Markov models (VLMMs) with a "deep" memory just where it is really needed (Ron, Singer, & Tishby, 1994).

Prediction contexts of variable length in VLMMs are often represented as prediction suffix trees (PSTs) (Rissanen, 1983). The relevant prediction context is defined as the deepest node in the PST that can be reached from the root when reading the input stream in reversed order.

Prediction suffix trees can be constructed in a top-down (Ron, Singer, & Tishby, 1994; Ron, Singer, & Tishby, 1996; Weinberger, Rissanen, & Feder, 1995), or bottom-up (Bühlmann, 1997; Guyon, & Pereira, 1995) fashion. Both schemes strongly depend on the construction parameters regulating candidate context selection and growing/pruning decisions (Bühlmann, 1997; Guyon, & Pereira, 1995). The appropriate values for those parameters are derived only under asymptotic considerations. In practical applications, the parameters must be set by the modeler, which can be, as we will see, quite inconvenient. Bühlmann (1997) suggests to optimize the construction parameters' values through minimization of model complexity measured, for example, by the Akaike information criterion (Akaike, 1974).

We introduce finite-context predictive models similar in spirit to VLMMs. The key idea behind our approach is a spatial representation of candidate prediction contexts, where contexts with long common suffices (i.e. contexts that are likely to produce similar continuations) are mapped close to each other, while contexts with different suffices (and potentially different continuations) correspond to points lying far from each other. Selection of the appropriate prediction contexts is left to a vector quantizer. Dense areas in the spatial representation of potential prediction contexts correspond to contexts with long common suffices and are given more attention by the vector quantizer.

The paper has the following organization: Section 2 brings a brief introduction to methods of quantifying and representing subsequence distributions in symbolic sequences. In section 3, we use the framework of finite memory sources to introduce our predictive models as well as the classical and variable memory length Markov models. Section 4 contains a detailed comparison of the studied model classes on a symbolic sequence obtained by quantizing activity changes of a laser in a chaotic regime, and on the Feigenbaum sequence generated from the logistic map with the period-doubling accumulation point parameter value. Discussion summarizes the empirical results and outlines directions in our current and future research.

2 Quantifying and representing subsequence structure in symbolic sequences

We consider sequences $S = s_1 s_2 ...$ over a finite alphabet $\mathcal{A} = \{1, 2, ..., A\}$ (i.e. every symbol s_i is from $\mathcal{A}$) generated by stationary information sources (Khinchin, 1957). The sets of all sequences over $\mathcal{A}$ with a finite number of symbols and exactly n symbols are denoted by $\mathcal{A}^+$ and $\mathcal{A}^n$, respectively. By S_i^j, $i \leq j$, we denote the string $s_i s_{i+1} ... s_j$, with $S_i^i = s_i$. For each sequence $S = s_1 s_2 ... s_n \in \mathcal{A}^+$, S^R denotes the reversed sequence $S^R = s_n s_{n-1} ... s_1$.

2.1 Statistics on symbolic sequences

Denote the (empirical) probability of finding an n-block $w \in \mathcal{A}^n$ in S by $\hat{P}_n(w)$. A string $w \in \mathcal{A}^n$ is said to be an allowed n-block in the sequence S, if $\hat{P}_n(w) > 0$. The set of all allowed n-blocks in S is denoted by $[S]_n$.

Statistical n-block structure in a sequence S is usually described through generalized entropy spectra. The spectra are constructed using a formal parameter β that can be thought of as the inverse temperature in the statistical mechanics of spin systems (Crutchfield, & Young, 1990).

The original distribution of n-blocks, $\hat{P}_n(w)$, is transformed to the "twisted" distribution (Young, & Crutchfield, 1993) (also known as the "escort" distribution (Beck, & Schlögl, 1995))

$$Q_{\beta,n}(w) = \frac{\hat{P}_n^\beta(w)}{\sum_{v \in [S]_n} \hat{P}_n^\beta(v)}. \tag{1}$$

The entropy rate

$$h_{\beta,n} = \frac{-\sum_{w \in [S]_n} Q_{\beta,n}(w) \log Q_{\beta,n}(w)}{n} \tag{2}$$

of the twisted distribution $Q_{\beta,n}$ approximates the thermodynamic entropy density (Young, & Crutchfield, 1993)

$$h_\beta = \lim_{n \to \infty} h_{\beta,n}. \tag{3}$$

When $\beta = 1$ (metric case), $Q_{1,n}(w) = \hat{P}_n(w)$, $w \in \mathcal{A}^n$, and h_1 becomes the metric entropy of subsequence distribution in the sequence S, that gives the asymptotic growth rate of the block entropy $H_n = n h_{1,n}$.

The infinite temperature regime ($\beta = 0$), also known as topological, or counting case, is characterized by the distribution $Q_{0,n}(w)$ assigning equal probabilities to all allowed n-blocks. The topological entropy h_0 gives the asymptotic exponential growth rate of the number of distinct n-blocks in S as $n \to \infty$.

Varying the parameter β amounts to scanning the original n-block distribution $\hat{P}_n$: the most probable and the least probable n-blocks become dominant in the positive zero ($\beta = \infty$) and the negative zero ($\beta = -\infty$) temperature regimes respectively. Varying β from 0 to ∞ amounts to a shift from all allowed n-blocks to the most probable ones by accentuating still more and more probable subsequences. Varying β from 0 to $-\infty$ accentuates less and less probable n-blocks with the extreme of the least probable ones.

We note that the thermodynamic entropy densities are closely related to the β-order Rényi entropy rates (Rényi, 1959) (cf. (Young, & Crutchfield, 1993)). In particular, the

two quantities are known to coincide in the topological and metric cases (Grassberger, 1991).

2.2 Geometric representations of subsequence structure

In (Tiňo, 1998) we formally study a geometric representation of subsequence structure, called the chaos game representation, originally introduced by Jeffrey (1990) to study DNA sequences (see also (Oliver, Galván, Garcia, & Roldan, 1993; Roldan, Galván, & Oliver, 1994; Li, 1997)). The basis of the chaos game representation of sequences over an alphabet $\mathcal{A} = \{1, 2, ..., A\}$ is an iterative function system (IFS) (Barnsley, 1988) consisting of A affine contractive maps[1] $1, 2, ..., A$, acting on the N-dimensional unit hypercube[2] $X = [0, 1]^N$, $N = \lceil \log_2 A \rceil$:

$$i(x) = kx + (1 - k)t_i, \quad t_i \in \{0, 1\}^N, \quad t_i \neq t_j \text{ for } i \neq j. \tag{4}$$

The contraction coefficient of the maps $1, ..., A$, is $k \in (0, \frac{1}{2}]$.

The chaos game representation $CGR_k(S)$ of a sequence $S = s_1 s_2 ...$ over $\mathcal{A}$ is obtained as follows (Tiňo, 1998):

1. Start in the center $x_* = \{\frac{1}{2}\}^N$ of the hypercube X, $x_0 = x_*$.
2. Plot the point $x_n = j(x_{n-1})$, $n \geq 1$, provided the n-th symbol s_n is j.

A useful variant of the chaos game representation, that we call the chaos n-block representation (Tiňo, 1998), codes allowed n-blocks as points in X. The chaos n-block representation $CBR_{n,k}(S)$ of the sequence S is constructed by plotting only the last points of the chaos game representations $CGR_k(w)$ of allowed n-blocks $w \in [S]_n$.

Formally, let $u = u_1 u_2 ... u_n \in \mathcal{A}^n$ be a string over $\mathcal{A}$ and $x \in X$ a point in the hypercube X. The point

$$u(x) = u_n(u_{n-1}(...(u_2(u_1(x)))...)) = (u_n \circ u_{n-1} \circ ... \circ u_2 \circ u_1)(x)$$

is considered a geometrical representation of the string u under the IFS (4). For a set $Y \subseteq X$, $u(Y)$ is then $\{u(x) |\ x \in Y\}$.

Given a sequence $S = s_1 s_2 ...$ over $\mathcal{A}$, the chaos n-block representation of S is defined as a sequence of points

$$CBR_{n,k}(S) = \left\{ S_i^{i+n-1}(x_*) \right\}_{i \geq 1}, \tag{5}$$

containing a point $w(x_*)$ for each n-block w in S. The map $w \to w(x_*)$ is one-to-one.

The chaos n-block representation has many useful properties. First of all, as shown in (Tiňo, 1998), the estimates of Rényi generalized dimension spectra (McCauley, 1994) quantifying the multifractal scaling properties of $CBR_{n,k}(S)$, directly correspond to the estimates of the Rényi entropy rate spectra (Rényi, 1959) measuring the statistical structure in the sequence S. In particular, for infinite sequences S, as the block length n grows, the box-counting fractal dimension (Barnsley, 1988) and the information dimension (Beck,

[1] To keep the notation simple, we slightly abuse mathematical notation and, depending on the context, regard the symbols $1, 2, ..., A$, as integers, or as referring to maps on X.

[2] for $x \in \Re$, $\lceil x \rceil$ is the smallest integer y, such that $y \geq x$

& Schlögl, 1995) estimates of the chaos n-block representations $CBR_{n,k}(S)$, tend to the sequences' topological and metric entropies, respectively, scaled by $(\log \frac{1}{k})^{-1}$.

Second, the chaos n-block representation codes the suffix structure in allowed n-blocks in the following sense (Tiňo, 1998): if $v \in \mathcal{A}^+$ is a suffix of length $|v|$ of a string $u = rv$, $r, u \in \mathcal{A}^+$, then $u(X) \subset v(X)$, where $v(X)$ is an N-dimensional hypercube of side length $k^{|v|}$. Hence, the longer is the common suffix shared by two n-blocks, the closer the n-blocks are mapped in the chaos n-block representation $CBR_{n,k}(S)$. On the other hand, the Euclidean distance between points representing two n-blocks u, v, that have the same prefix of length $n-1$ and differ in the last symbol, is at least $1-k$.

3 Finite memory sources

An information source (Khinchin, 1957; Weinberger, Rissanen, & Feder, 1995) over an alphabet $\mathcal{A} = \{1, 2, ..., A\}$ is defined by a family of probability measures P_n on n-blocks over $\mathcal{A}$, $n = 0, 1, 2, ...$ Consistent measures satisfy the marginality condition: for all[3] $s \in \mathcal{A}$, $w \in \mathcal{A}^n$, $n = 0, 1, 2...$,

$$\sum_{s \in \mathcal{A}} P_{n+1}(ws) = P_n(w).$$

In applications it is useful to consider probability functions P_n that are both consistent and easy to handle. This can be achieved, for example, by assuming a finite source memory of length at most L, and formulating the conditional measures

$$P(s|w) = \frac{P_{L+1}(ws)}{P_L(w)}, \quad w \in \mathcal{A}^L,$$

using a function $c : \mathcal{A}^L \to \mathcal{Q}$, from L-blocks over $\mathcal{A}$ to a (presumably small) finite set $\mathcal{Q}$ of prediction contexts,

$$P(s|w) = P(s|c(w)). \tag{6}$$

Finite memory sources can be used as sequence generators by initiating them with the first L-block and letting them produce a continuation according to the next-symbol distribution (6).

Consider a stationary ergodic source Σ and a typical long sequence $S = s_1 s_2 ... s_m$, $m >> L$, generated by that source. Denote the empirical n-block frequency counts in S by $\hat{P}_n$. Let $\mathcal{M}$ be a finite memory source built on S. The probability that the model $\mathcal{M}$, initiated with the first L-block S_1^L, generates the continuation S_{L+1}^m is

$$P_{\mathcal{M}}\left(S_{L+1}^m | S_1^L\right) = \prod_{i=L+1}^{m} P\left(s_i | c\left(S_{i-L}^{i-1}\right)\right)$$

and the likelihood of the sequence S with respect to the model $\mathcal{M}$ is determined as

$$P_{\mathcal{M}}(S) = \hat{P}_L\left(S_1^L\right) P_{\mathcal{M}}\left(S_{L+1}^m | S_1^L\right). \tag{7}$$

In this paper, the fitted sources $\mathcal{M}$ are compared by

[3] $\mathcal{A}^0 = \{\Lambda\}$ and $P_0(\Lambda) = 1$, where Λ denotes the empty string.

1. letting the sources, initiated with the training sequence S, generate sequences G on their own: copy the first L-block S_1^L from the training sequence to the model generated sequence $G = g_1 g_2 ...$, i.e. $G_1^L = S_1^L$. Then, for $i > L$ iteratively generate the i-th symbol g_i with respect to the model distribution $P(g_i|c(G_{i-L}^{i-1}))$.

2. evaluating the statistical distances[4] between the model generated sequences G and the training sequence S.

In what follows, we present two specific examples of finite memory sources and later introduce a novel approach for constructing finite memory sources from geometric representations of training sequences.

In *Markov models* (MMs) of order $n \leq L$, for all L-blocks $w \in \mathcal{A}^L$, $c(w)$ is the length-n suffix of w, i.e. $c(uv) = v$, $v \in \mathcal{A}^n$, $u \in \mathcal{A}^{L-n}$.

In *variable memory length Markov models* (VLMMs), the suffices $c(w)$ of L-blocks $w \in \mathcal{A}^L$ can have different lengths, depending on the particular L-block w. We briefly review strategies for selecting and representing the prediction contexts.

Suppose we are given a long training sequence S over $\mathcal{A}$. Let $w \in [S]_n$ be a potential prediction context of length $n < L$ used to predict the next symbol $s \in \mathcal{A}$ according to the empirical estimates $\hat{P}(s|w) = \hat{P}_{n+1}(ws)/\hat{P}_n(w)$. If for a symbol $a \in \mathcal{A}$, such that $aw \in [S]_{n+1}$, the prediction probability of the next symbol s, $\hat{P}(s|aw) = \hat{P}_{n+2}(aws)/\hat{P}_{n+1}(aw)$, with respect to the extended context differs "significantly" from $P(s|w)$, then adding the symbol $a \in \mathcal{A}$ in the past helps in the next-symbol predictions. Several decision criteria have been suggested in the literature. For example, one can extend the prediction context w with a symbol $a \in \mathcal{A}$, if

- there exists a symbol $s \in \mathcal{A}$, such that (Ron, Singer, & Tishby, 1996)

$$\hat{P}(s|aw) \geq \frac{1}{A}(1+\epsilon_1)\epsilon_1 \quad \text{and} \quad \frac{\hat{P}(s|aw)}{\hat{P}(s|w)} > 1 + 3\epsilon_1. \tag{8}$$

- the Kullback-Leibler divergence between the next-symbol distributions for the candidate prediction contexts w and aw, weighted by the prior distribution of the extended context aw, exceeds a given threshold (Ron, Singer, & Tishby, 1994; Guyon, & Pereira, 1995),

$$\hat{P}_{n+1}(aw) \sum_{s \in \mathcal{A}} \hat{P}(s|aw) \log \frac{\hat{P}(s|aw)}{\hat{P}(s|w)} \geq \epsilon_2. \tag{9}$$

The (small, positive) construction parameters ϵ_1, ϵ_2 are supplied by the modeler. For other variants of decision criteria see (Weinberger, Rissanen, & Feder, 1995).

A natural representation of the set $\mathcal{Q}$ of prediction contexts, together with the associated next-symbol probabilities, has the form of a prediction suffix tree (PST) (Ron, Singer, & Tishby, 1996; Rissanen, 1983). The edges of PST are labeled by symbols from $\mathcal{A}$. From every internal node there is at most one outgoing edge labeled by each symbol. The nodes of PST are labeled by pairs $(s, \hat{P}(s|v))$, $s \in \mathcal{A}$, $v \in \mathcal{A}^+$, where v is a string associated with the walk starting from that node and ending in the root of the tree. For

[4] expressed in terms of cross-entropies and L_1 distances – see sections 4.1 and 4.2

each L-block $w \in \mathcal{A}^L$, the corresponding prediction context $c(w)$ is then the deepest node in the PST reached by taking a walk labeled by w^R, starting in the root.

The algorithm for building the PST has the following form[5] (Ron, Singer, & Tishby, 1996; Ron, Singer, & Tishby, 1994; Guyon, & Pereira, 1995):

- the initial PST is a single root node and the initial set of candidate contexts is $W = \{s \in \mathcal{A} |\ \hat{P}_1(s) > \epsilon_{st}\}$.
- while $W \neq \emptyset$, do:
  1. pick any $v = aw \in W$, $a \in \mathcal{A}$, and remove it from W
  2. add the context v to the PST by growing all the necessary nodes, provided the condition (8) (or (9)) holds[6]
  3. provided $|v| < L$, then for every $s \in \mathcal{A}$, if $\hat{P}(sv) > \epsilon_{add}$, add sv to W.

The depth of the resulting PST is at most L. The tree is grown from the root to the leaves. If a string v does not meet the criterion (8) (or (9)), it is not definitely ruled out, since its descendants are added to W in step 3. The idea is to keep a provision for the future descendents of v which might meet the selection criterion.

Variable memory length Markov models (VLMMs) are described as stochastic machines (SMs). Briefly, SMs are like finite state machines except that the state transitions take place with probabilities prescribed by a distribution T. The generating process is started in an initial state[7] and then, at any given time step, the machine is in some state i, and at the next time step moves to another state j outputting some symbol s, with the transition probability $T_{i,j,s}$.

Given a PST, the set Q of prediction contexts (states) of the corresponding VLMM contains the leaves of the PST plus contexts added so that the symbol driven state transition probabilities $T_{i,j,s}$ are properly defined (see (Ron, Singer, & Tishby, 1996; Ron, Singer, & Tishby, 1994; Guyon, & Pereira, 1995)). SMs representing VLMMs have suffix-free state sets Q and are known as probabilistic suffix automata (PSA) (Ron, Singer, & Tishby, 1996; Weinberger, Rissanen, & Feder, 1995).

Although VLMMs can be emulated with the corresponding PSTs, PSA representations of VLMMs give higher processing speed. In PSA, the longest suffices are precomputed into states, whereas in PSTs the longest suffices must be dynamically determined (Guyon, & Pereira, 1995).

3.1 Prediction fractal machines

Note that the prediction context function $c : \mathcal{A}^L \to \mathcal{Q}$ in Markov models of order $n \leq L$, can be interpreted as a natural homomorphism $c : \mathcal{A}^L \to \mathcal{A}^L|_{\mathcal{E}}$ corresponding to the equivalence relation $\mathcal{E} \subseteq \mathcal{A}^L \times \mathcal{A}^L$ on L-blocks over $\mathcal{A}$: $(u, v) \in \mathcal{E}$, if the L-blocks u, v share the same suffix of length n. The factor set $\mathcal{A}^L|_{\mathcal{E}} = \mathcal{Q} = \mathcal{A}^n$ consists of all n-blocks over $\mathcal{A}$.

[5] ϵ_{st} and ϵ_{add} are small positive construction parameters

[6] $\hat{P}(s|\Lambda) = \hat{P}_1(s)$, Λ is the empty string.

[7] in the experiments described in this paper, the initial state always corresponds to the first L-block of the training sequence

As already mentioned in the introduction, for large suffix lengths n, the estimation of prediction probabilities $P(s|c(w))$ can become infeasible. In what follows, we show a method for constructing an equivalence relation $\mathcal{E}$ on $\mathcal{A}^L$ (and hence the corresponding natural homomorphism $c : \mathcal{A}^L \to \mathcal{A}^L|_{\mathcal{E}}$) in the context of limited resources (limited number of equivalence classes).

Suppose our model cannot have more than M prediction contexts, i.e. the number of elements in the set $\mathcal{Q}$ is upper bounded by M. We impose a natural smoothness constraint on our model by assuming that L-blocks with long common suffices are likely to produce similar continuations, whereas L-blocks with different suffices may lead to a range of different future scenarios.

Given a sequence S over $\mathcal{A}$ on which we want to build our model, the smoothness constraint implies that the equivalence $\mathcal{E} \subseteq \mathcal{A}^L \times \mathcal{A}^L$ should factorize the set $[S]_L$ of allowed L-blocks into the set $[S]_L|\mathcal{E}$ of M equivalence classes, such that blocks within each equivalence class share as long common suffix as possible.

At this point, we suggest the reader to briefly return to section 2.2 and recall the basic notions in geometric representations of subsequence structure.

The equivalence relation $\mathcal{E}$ on the set $\mathcal{A}^L$ of L-blocks over $\mathcal{A}$ induces an equivalence relation $\mathcal{E}'$ on the set of their geometrical codings, $\mathcal{A}^L(x_*) = \{u(x_*)|\ u \in \mathcal{A}^L\}$,

$$\forall u, v \in \mathcal{A}^L,\ (u, v) \in \mathcal{E} \text{ iff } (u(x_*), v(x_*)) \in \mathcal{E}'.$$

Recalling the suffix structure coding properties of the chaos L-block representation $CBR_{L,k}(S)$, we reformulate the constraints on the equivalence $\mathcal{E}$ (restricted to the set $[S]_L$ of allowed L-blocks in the training sequence), as constraints on the induced equivalence $\mathcal{E}'$ (restricted to the set $[S]_L(x_*)$ of points appearing in $CBR_{L,k}(S)$).

For an n-block $w \in \mathcal{A}^n$, $n < L$, dense clusters of points from $CBR_{L,k}(S)$ in the set $w(X)$ show that there are many allowed L-blocks in S sharing the same suffix w. Sparsely inhabited areas in $w(X)$ reflect the presence of only few allowed L-blocks having the suffix w. Partitioning the set of allowed L-blocks into M equivalence classes, each containing L-blocks with as long a common suffix as possible, corresponds to partitioning the chaos L-block representation $CBR_{L,k}(S)$ into M subsets, each of diameter as small as possible. In practical terms, this means allocation of points from $CBR_{L,k}(S)$ to M codebook vectors $b_1, ..., b_M \in X$, such that the loss

$$E(S) = \sum_{w \in [S]_L} \hat{P}_L(w)\ d_E^2(w(x_*), c(w)) \tag{10}$$

is minimal, where $c(w) \in \mathcal{Q} = \{b_1, ..., b_M\}$ is the codebook vector to which the point $w(x_*)$ is allocated, and d_E is the Euclidean distance.

In other words, the problem of finding the set $\mathcal{Q}$ of M prediction contexts is formulated as a vector quantization task, where we construct a set of M codebook vectors $b_1, ..., b_M$ minimizing the loss function (10).

We refer to finite memory sources with the context function $c : \mathcal{A}^L \to \mathcal{Q}$, where $c(w)$ is the codebook vector representing the point $w(x_*)$, as the *prediction fractal machines* (PFMs). The prediction fractal machines are constructed as follows:

1. partition the hypercube X into M regions $V_1, ..., V_M$ by running a vector quantizer on the chaos L-block representation $CBR_{L,k}(S)$ of the training sequence $S = s_1...s_m$.

The regions V_i, $i = 1, ..., M$, are the Voronoi compartments (Aurenhammer, 1991) of the codebook vectors $b_1, ..., b_M$,

$$V_i = \{x \in X |\ d_E(x, b_i) = \min_j d_E(x, b_j)\}.$$

All points in V_i are allocated[8] to the codebook vector b_i.

2. set the counters $N(i, a)$, $i = 1, ..., M$, $a = 1, ..., A$, to zero
3. for $1 \leq t \leq m - L$
   - code the L-block S_t^{t+L-1} by a point $S_t^{t+L-1}(x_*)$
   - if $S_t^{t+L-1}(x_*) \in V_i$, increment the counter $N(i, s_{t+L})$ by one
4. with each prediction context (codebook vector) $b_1, ..., b_M$, associate the next symbol probabilities

$$P(s|b_i) = \frac{N(i, s)}{\sum_{a \in \mathcal{A}} N(i, a)}, \quad s \in \mathcal{A}.$$

We also present a simple method for constructing SMs, called *stochastic fractal machines* (SFMs), from quantized chaos L-block representations $CBR_{L,k}(S)$ of training sequences $S = s_1...s_m$.

1. the SFM has M states 1,...,M, indexing the codebook vectors $b_1, ..., b_M$
2. set the counters $N(i, j, a)$, $i, j = 1, ..., M$, $a = 1, ..., A$, to zero
3. for $1 \leq t \leq m - L$
   - code the consecutive L-blocks S_t^{t+L-1} and S_{t+1}^{t+L} by points $S_t^{t+L-1}(x_*)$ and $S_{t+1}^{t+L}(x_*)$, respectively
   - if $S_t^{t+L-1}(x_*) \in V_i$ and $S_{t+1}^{t+L}(x_*) \in V_j$, increment the counter $N(i, j, s_{t+L})$ by one
4. the symbol driven state transition probabilities are given by

$$T_{i,j,s} = \frac{N(i, j, s)}{\sum_{a \in \mathcal{A}} \sum_{r=1}^{M} N(i, r, a)}, \quad i, j \in \{1, ..., M\}, \quad s \in \mathcal{A}.$$

Compared to PFMs, SFMs have an advantage of directly representing, in the state transition diagrams, the topological structure of sequences they generate. A PFM can emulate the corresponding SFM built on the same codebook, provided the SFM has a deterministic state transition structure, i.e. for every state q and each symbol s, there is at most one transition from q labeled with s.

[8] Ties as events of measure zero (points land on the border between the compartments) are broken according to index order

4 Experiments

We compared the prediction fractal machines (PFMs) with both the classical and variable memory length Markov models referred to as MM and VLMM, respectively. The experiments were performed on two symbolic sequences with different levels of subsequence distribution structure.

4.1 Laser data

The first data set is a long sequence[9] (10.000 items) of differences between the successive activations of a real laser in a chaotic regime. The sequence was quantized into a symbolic stream over four symbols corresponding to low and high positive/negative laser activity change. More precisely, the activity differences in the range $[-200, 200]$ were partitioned into four regions $[0, 50)$, $[50, 200]$, $(-64, 0]$ and $[-200, -64)$ represented by symbols 1,2,3 and 4 respectively.

The training sequence S is spatially represented as a two-dimensional chaos L-block representation $CBR_{L,k}(S)$, with $L = 20$ and $k = \frac{1}{2}$.

Using the training sequence S and its geometric representation $CBR_{L,k}(S)$, we construct the predictive models PFMs, SFMs, VLMMs and MMs that are compared through the Lempel-Ziv entropy and cross-entropy[10] estimates (Ziv, & Merhav, 1993) on the model generated sequences.

For a stationary ergodic process that has generated a sequence $Q = q_1 q_2 ... q_m$, the Lempel-Ziv codeword length for Q, divided by m, is a computationally efficient and reliable estimate $h_{LZ}(Q)$ of the source (metric) entropy (Ziv, & Merhav, 1993). Let $n(Q)$ denote the number of phrases in Q resulting from the incremental parsing of Q, i.e. sequential parsing of Q into distinct phrases such that each phrase is the shortest string which is not a previously parsed phrase. The Lempel-Ziv codeword length for Q is approximated with $n(Q) \log n(Q)$ (Ziv, & Merhav, 1993).

Let P and R be two Markov probability measures, each of some (unknown) finite order. The cross-entropy (also known as the Kullback-Leibler divergence) between P and R,

$$d^{KL}(R|P) = \limsup_{n \to \infty} \frac{1}{n} \sum_{w \in \mathcal{A}^n} R_n(w) \log \frac{R_n(w)}{P_n(w)}$$

measures the expected additional code length required when using the ideal code for P instead of the ideal code for the "right" distribution R.

Given length-m realizations S_P and S_R of P and R, respectively, the cross-entropy $D^{KL}(R|P)$ is estimated using the Lempel-Ziv sequential parsing of S_R with respect to S_P. First, find the longest prefix of S_R that appears in S_P, i.e. the largest integer r such that the r-blocks $(S_R)_1^r$ and $(S_P)_i^{i+r-1}$ are equal, for some i. $(S_R)_1^r$ is the first phrase of S_R with respect to S_P. Next, start from the r-th position in S_R and find, in a similar manner, the longest prefix $(S_R)_r^j$ that appears in S_P, and so on. The procedure terminates when S_R is completely parsed with respect to S_P. Denoting the number of phrases in S_R with

[9] taken from `http://www.cs.colorado.edu/~andreas/Time-Series/SantaFe.html`

[10] with respect to the training sequence

respect to S_P by $n(S_R|S_P)$, the Lempel-Ziv estimate of $d^{KL}(R|P)$ is computed as[11] (Ziv, & Merhav, 1993)

$$d_{LZ}^{KL}(S_R|S_P) = \frac{n(S_R|S_P)\log m}{m} - h_{LZ}(S_R). \tag{11}$$

Due to finite sequence length effects, the estimates of the cross-entropy $h_{LZ}^{KL}(S|G)$ between the training and model generated sequences S and G, respectively, may be negative. Since the only nonconstant term in $h_{LZ}^{KL}(S|\cdot)$ is the number of cross-phrases in S with respect to G, we use $n(S|\cdot)$ as the relevant performance measure. The higher is the number of cross-phrases, the bigger is the estimated statistical distance between the sequences S and G.

For each $M \in \{1, 2, ..., 15, 20, 30, ..., 300\}$, we quantized the $CBR_{L,k}(S)$ into M Voronoi compartments corresponding to M codebook vectors $b_1, ..., b_M$, and constructed both the PFM and the SFM. To test the dependence of PFM construction on the particular vector quantizer, we used both the K-means (MacQueen, 1967; Buhmann, 1995) and dynamic cell structures (DCS) (Bruske, & Sommer, 1995) techniques.

The aim of K-means clustering is a minimization of the loss (10), whereas DCS attempt to jointly minimize (10) and preserve the input space topology in the cells' neighborhood structure. Unlike in the traditional self-organizing feature maps (SOFM) (Kohonen, 1990), the number of quantization centers and the cells' neighborhood structure in DCS is not fixed, but as the learning process proceeds, the codebook is gradually grown and the corresponding codebook topology is adjusted to mimic to topology of the training data.

Each model was used to produce 10 model generated sequences G of length equal to the length of the training sequence S. For every sequence G, we computed the Lempel-Ziv entropy and cross-entropy estimates $h_{LZ}(G)$ and $h_{LZ}^{KL}(S|G)$, respectively.

The results are presented in figure 1. Shown are the mean quantities across 10 sequence generation realizations. The numbers $n(S|G)$ of cross-phrases are scaled down by a factor 10^{-3}. The mean performances of the PFMs and the SFMs are almost identical, irrespective of the codebook construction technique (apart from small fluctuations due to instability of the K-means quantization with respect to random codebook initialization).

We built VLMMs using both context selection criteria (8) and (9) (see section 3). Maximal memory length L was set to 20. Varying the construction parameters, we obtained a sequence of VLMMs of growing size.

Each VLMM was used to generate 10 sequences G, on which the measure-theoretic quantities $h_{LZ}(G)$ and $h_{LZ}^{KL}(S|G)$ were computed. The results can be seen in figure 2. Both VLMM construction schemes yield comparable performances.

Finally, we constructed the classical MMs of order 1,2,...,5. Figures 3 and 4 bring a comparison, with respect to the mean Lempel-Ziv performance measures, of the MMs, PFMs constructed via dynamic cell structures on $CBR_{L,k}(S)$ and VLMMs built using the scheme (9). Dotted continuations of the MM and VLMM lines correspond to the fifth-order MM with $4^5 = 1024$ prediction contexts and a VLMM with 560 prediction contexts, respectively.

The results deserve several comments.

Obviously, as the model size grows, the VLMMs outperform the standard MMs, but they do so significantly later then the PFMs. After an initial phase of being too simple to

[11] Ziv and Merhav (1993) proved that if S_R and S_P are independent realizations of two finite order Markov processes R and P, $d_{LZ}^{KL}(S_R|S_P)$ converges (as $m \to \infty$) to $d^{KL}(R|P)$ almost surely.

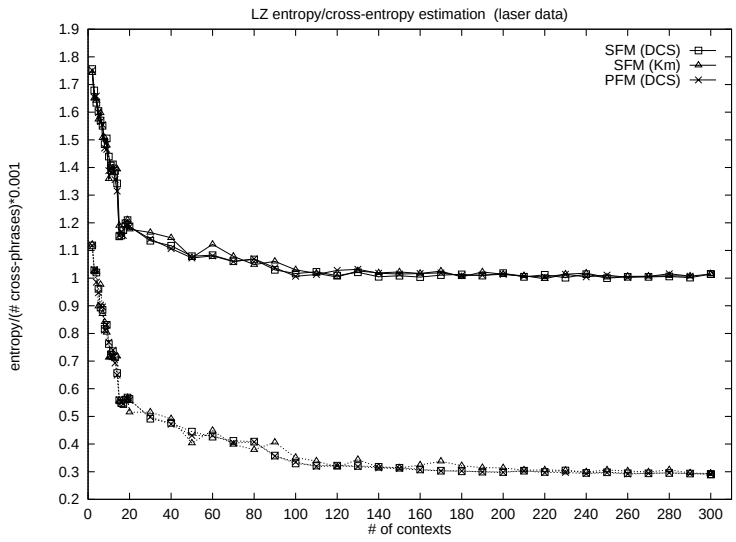

Figure 1: Entropy and cross-entropy Lempel-Ziv estimates on sequences G generated by PFMs and the corresponding SFMs. The models are built on the laser sequence S. Shown are the mean entropies $h_{LZ}(G)$ (solid lines) across 10 sequence generation realizations and the mean numbers $n(S|G)$ of cross-phrases (dotted lines) scaled down by a factor 10^{-3}. Technique used to quantize the chaos L-block representation of S is indicated by `(DCS)` – dynamic cell structures, and `(Km)` – K-means clustering.

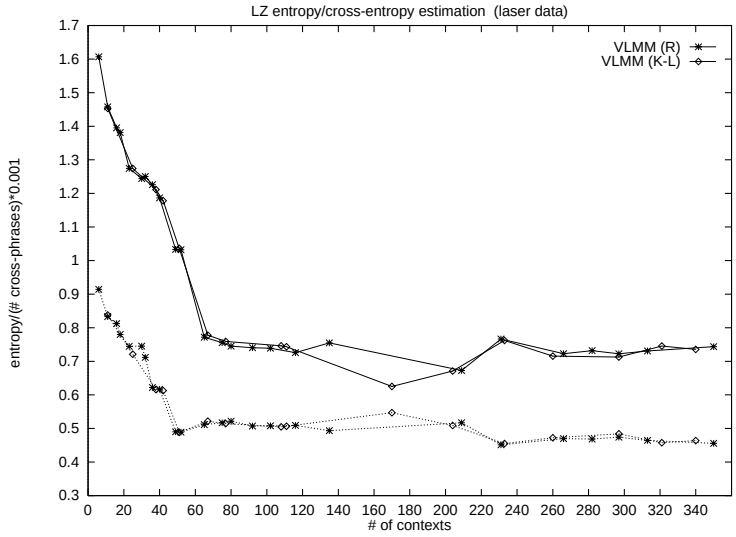

Figure 2: Mean entropy estimates $h_{LZ}(G)$ (solid lines) and numbers of cross-phrases $n(S|G)$ (with respect to the laser sequence S, dotted lines) on sequences generated by VLMMs constructed using schemes (8) and (9) (indicated by `(R)` and `(K-L)`, respectively).

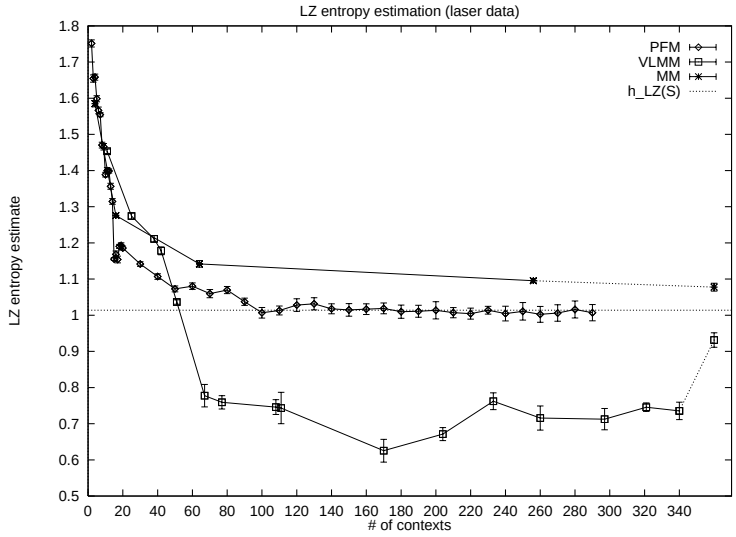

Figure 3: LZ entropy estimates on sequences generated by MMs, PFMs constructed via dynamic cell structures and VLMMs built using the scheme (9). Dotted continuations of the MM and VLMM lines correspond to the fifth-order MM with 1024 prediction contexts and a VLMM with 560 prediction contexts, respectively. Shown are the mean values and standard deviations across 10 sequence generation realizations. The horizontal dotted line corresponds to the LZ estimate $h_{LZ}(S)$ of the training sequence entropy.

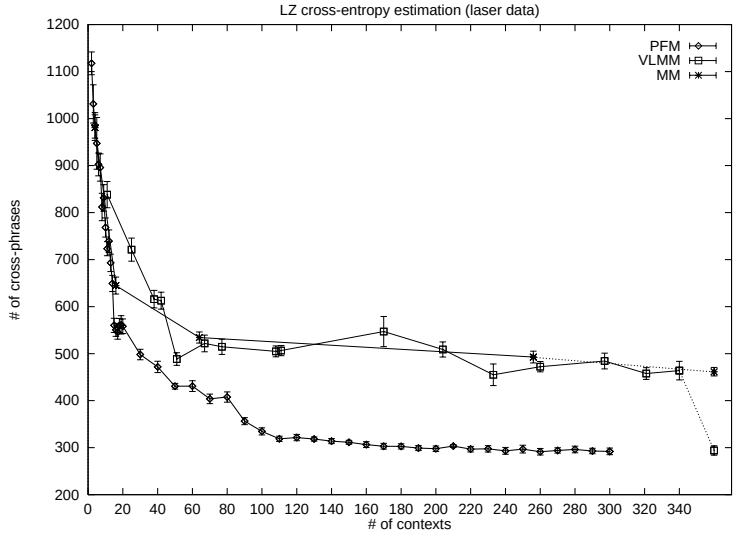

Figure 4: Numbers of cross-phrases $n(S|G)$ with respect to the laser sequence S, calculated on model generated sequences G. The experimental setting is described in caption to the previous figure.

account for details in the training sequence subsequence distribution (entropy estimates of model generated sequences G are higher than the entropy estimate of the training sequence S), the VLMMs start to include specialized contexts that introduce more statistical structure into the model generated sequences than can actually be found in the training sequence ($h_{LZ}(G) < h_{LZ}(S)$). We will come back to this point when analysing the thermodynamic entropy rate spectra of sequences generated by selected model class representatives.

On the other hand, increasing the number of prediction contexts in PFMs results in an underfitting stage, followed by a stage of fairly leveled performance. The next symbol uncertainty in sequences G generated at this stage match the training sequence entropy estimate $h_{LZ}(S)$ almost perfectly.

The cross-entropy estimates $h_{LZ}^{KL}(S|G)$ do not show any deterioration in the PFM-generated subsequence distribution when increasing the number of prediction contexts[12]. Once enough codebook vectors cover the training sequence representation $CBR_{L,k}(S)$, so that the corresponding Voronoi compartments are sufficiently small regions with almost constant next-symbol probabilities, further refinement of those regions produces just "nonminimal" versions of the PFMs.

We used the thermodynamic entropy rate spectra (eq.(2)) to study in a greater detail the subsequence distribution of the training and model generated sequences. To this end, we first selected three model class representatives AIC(PFM), AIC(VLMM) and AIC(MM), one from each model class, using the Akaike information criterion (AIC) (Akaike, 1974). The AIC method penalizes oversized models in a model class Γ:

$$AIC(\Gamma) = \arg\min_{\mathcal{M}\in\Gamma}\{-2\ \log P_{\mathcal{M}}(S) + 2\ \mathrm{Par}(\mathcal{M})\},$$

where $P_{\mathcal{M}}(S)$ is the likelihood of the training sequence given the model $\mathcal{M}$ and $\mathrm{Par}(\mathcal{M})$ is the number of free parameters in the model $\mathcal{M}$. Using the AIC for selection of optimal size VLMMs was suggested in (Bühlmann, 1997).

The selected representatives AIC(PFM), AIC(VLMM) and AIC(MM) have 150, 233 and 256 prediction contexts, respectively[13].

Each representative was let to generate sequences G, on which we subsequently computed the thermodynamic entropy rate spectrum. As an example, figure 5 shows the typical spectra for the 6-block statistics. The positive temperature parts unveil an overestimation of high probability 6-blocks by the VLMM representative. More extremal peaks in the 6-block distribution can be found in the sequences generated by AIC(VLMM) than are actually present in the training sequence S. The opposite is true about the MM representative. The MM of order 4 is too simple to account for all details in the training sequence 6-block distribution. The positive temperature branch of the PFM-generated spectrum suggests an almost perfect coincidence of the 6-block training sequence and model generated distributions, across all high probability levels.

The negative temperature part of entropy rate spectra concentrates on rare 6-blocks. Flat regions in the spectra appear because the sequences are too short to reveal any significant probabilistic structure in low probability 6-blocks. The representatives do not introduce any strong additional structure into rare 6-blocks and the performances

[12] at least up to 300 contexts

[13] AIC(MM) is of order 4

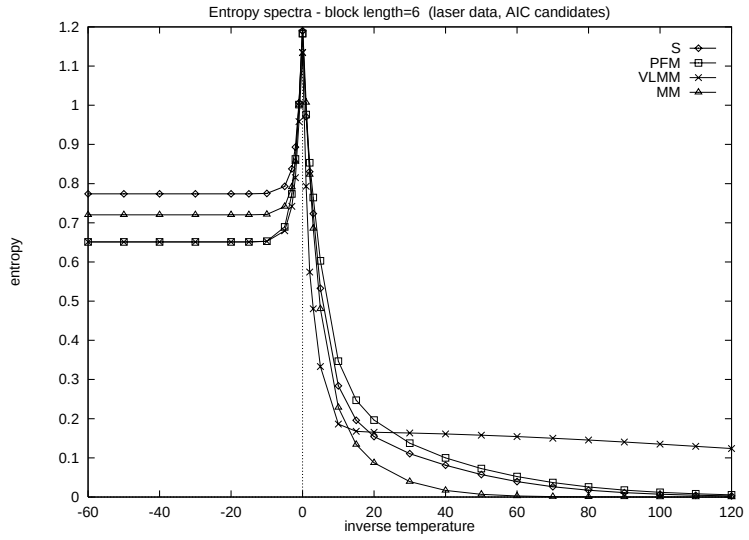

Figure 5: Estimation of thermodynamic entropy rate spectra (based on 6-block statistics) of sequences generated by the AIC-selected representatives AIC(PFM), AIC(VLMM) and AIC(MM). Spectrum of the training sequence S is plotted with the bold line.

of AIC(PFM) and AIC(VLMM) are only slightly worse than that of the less sophisticated model AIC(MM).

Next, we analysed the models' behavior across various block lengths by evaluating the entropy rate spectra $h_{\beta,n}$ for block lengths $n = 1, 2, ..., 11$, and nonnegative inverse temperatures from $B = \{0, 1, 2, 3, 5, 10, 15, 20, 30, 60, 100\}$. The entropy rate spectra of the training sequence S and the model generated sequences G are compared through the distances

$$D_n(S, G) = \sum_{\beta \in B} |h_{\beta,n}(S) - h_{\beta,n}(G)|.$$

The distance $D_n(S, G)$ approximates the L_1 entropy spectra distance $\int_0^\infty |h_{\beta,n}(S) - h_{\beta,n}(G)| dq(\beta)$, corresponding to high probability n-blocks. The measure q is concentrated on high temperature values. Because of the finite sequence length, the high temperature statistics are better determined than the low temperature ones.

The entropy spectra distances D_n for the models AIC(PFM), AIC(VLMM) and AIC(MM) are shown in figure 6. The PFM representative is clearly superior to the other two model class representatives. Its modeling performance on small block lengths almost equals that of the MM and continues to be good for higher block lengths as well. The performance of the VLMM representative improves with increasing block length, but never equals that of AIC(PFM).

Table 1 summarizes the performance of the studied model classes for models of sizes comparable to the sizes of the AIC candidates. Both PFMs and SFMs are the most favorable candidates in all cases.

It is instructive to plot the chaos L-block representation $CBR_{L,k}(S)$ of the training sequence, together with points $w(x_*)$ representing the prediction contexts $w \in \mathcal{Q}$ of the VLMM and MM representatives.

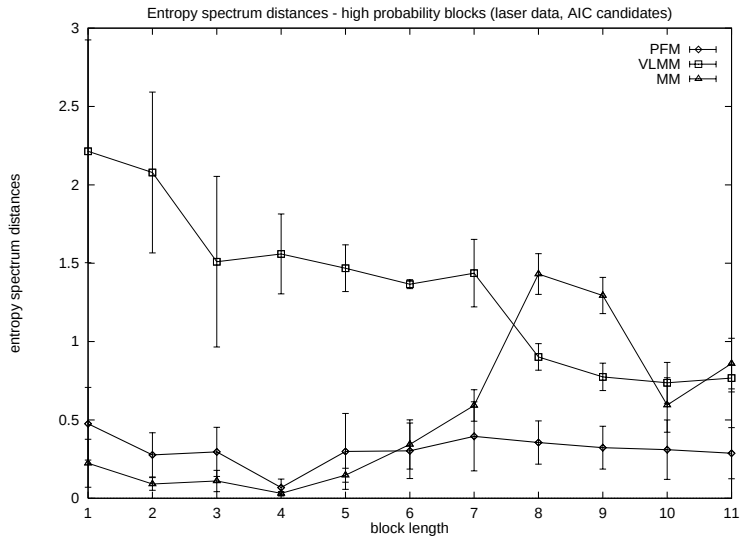

Figure 6: Entropy spectra distances corresponding to positive temperatures evaluated on 10 sequence generation realizations. Shown are the mean values and standard deviations. AIC(PFM) modeling performance on high probability n-blocks is superior to that of the representatives AIC(VLMM) and AIC(MM).

Model	# contexts	$h_{LZ}(G)$	$n(S\|G)$	$D_n^{min}(S,G)$	$D_n^{max}(S,G)$
PFM	**150**	**1.015 (0.017)**	**311.4 (4.17)**	**0.069 (0.05)**	**0.476 (0.23)**
SFM	150	1.009 (0.024)	314.1 (6.13)	0.059 (0.03)	0.501 (0.28)
VLMM	170	0.625 (0.031)	547.0 (32.0)	1.011 (0.06)	3.289 (0.08)
PFM	**230**	**1.014 (0.011)**	**297.5 (6.80)**	**0.067 (0.03)**	**0.493 (0.19)**
SFM	230	1.001 (0.017)	304.3 (4.69)	0.041 (0.01)	0.546 (0.29)
VLMM	233	0.762 (0.023)	455.1 (22.9)	0.736 (0.08)	2.214 (0.51)
PFM	**260**	**1.002 (0.022)**	**291.4 (6.93)**	**0.050 (0.02)**	**0.481 (0.20)**
SFM	260	1.004 (0.014)	294.7 (4.90)	0.072 (0.04)	0.609 (0.34)
VLMM	260	0.716 (0.033)	472.2 (11.0)	0.712 (0.09)	2.914 (0.31)
MM	256	1.095 (0.005)	492.7 (12.5)	0.091 (0.01)	1.431 (0.13)

Table 1: Model performance evaluated on 10 sequence generation realizations for models of sizes comparable to those of the AIC candidates. Shown are the mean values (together with standard deviations in parenthesis) for the Lempel-Ziv entropy estimates $h_{LZ}(G)$ on the model generated sequences G, the number $n(S|G)$ of cross-phrases in the training sequence S with respect to G, minimal and maximal entropy distances $D_n^{min}(S,G) = \min_{1\leq n\leq 11} D_n(S,G)$ and $D_n^{max}(S,G) = \max_{1\leq n\leq 11} D_n(S,G)$, respectively. Lempel-Ziv entropy estimate of the training sequence entropy is 1.0137.

As seen in figure 7c, the points $w(x_*)$ (shown as diamonds) corresponding to the prediction contexts $w \in \mathcal{A}^4$ of the 4th-order MM blindly cover the unit square X, regardless of the actual distribution of points (shown as dots) in the allowed L-blocks' geometric representation.

Prediction contexts of the VLMM representative are suffices of the allowed L-blocks, and so geometric representations of the prediction contexts concentrate in the areas inhabited by $CBR_{L,k}(S)$. The context selection criteria favor prediction contexts whose probability exceeds some pre-set "acceptance" threshold and whose next-symbol probabilities do not significantly differ from those of the extended contexts. The result (see figure 7b) is a sort of "conditional" vector quantization of the geometric training sequence representation, whose aim is to cover the set of "accepted" allowed blocks with a minimal set of prediction contexts, taking into account the associated next-symbol probabilities.

Prediction contexts (codebook vectors) of the PFM representative are shown in figure 7a.

4.2 Feigenbaum sequence

The second data set is an artificial symbolic sequence over a binary alphabet $\mathcal{A} = \{1, 2\}$, also known as the Feigenbaum sequence (Freund, Ebeling, & Rateitschak, 1996). The sequence is generated by iterating the logistic map $y_{t+1} = r y_t(1 - y_t)$, $y \in [0, 1]$, with the control parameter r set to the period doubling accumulation point value[14] (McCauley, 1994), and partitioning the iterands y_t into two regions[15] $[0, \frac{1}{2})$ and $[\frac{1}{2}, 1]$, corresponding to symbols 1 and 2, respectively.

The topological structure of the sequence (i.e. the structure of allowed n-blocks not regarding their probabilities) can only be described using a context sensitive tool – a restricted indexed context-free grammar (Crutchfield, & Young, 1990). The metric structure of the Feigenbaum sequence is organized in a self-similar fashion (Freund, Ebeling, & Rateitschak, 1996). The transition between the ranked distributions for block lengths $2^g \rightarrow 2^{g+1}$, $3\ 2^{g-1} \rightarrow 3\ 2^g$, $g \geq 1$, is achieved by rescaling the horizontal and vertical axis by a factor 2 and $\frac{1}{2}$, respectively. Plots of the Feigenbaum sequence n-block distributions, $n = 1, 2, ..., 8$, can be seen in figure 8. Numbers above the plots indicate the corresponding block lengths. The arrows connect distributions with the $(2, \frac{1}{2})$-scaling self-similarity relationship.

We represented the binary Feigenbaum sequence S containing 260.000 symbols through a one-dimensional chaos L-block representation $CBR_{L,k}(S)$, with $L = 30$ and $k = \frac{1}{2}$. Then, we built a series of PFMs using codebook vectors obtained via dynamic cell structures technique run on the $CBR_{L,k}(S)$.

By varying the construction parameters, we obtained a series of VLMMs of growing size (construction scheme (9)). Unlike in the previous experiment, constructing the series of increasingly complex VLMMs appeared to be a troublesome task. Due to the Feigenbaum sequence organization, the construction procedure did not work "smoothly" with varying construction parameters. Instead, we experienced a highly non-regular behavior with intervals of parameter values yielding unchanged VLMMs, and tiny regions in parameter space corresponding to a large spectrum of VLMM sizes. Therefore, it was impossible

[14] r=3.56994567...

[15] this partition is a generating partition defined by the critical point

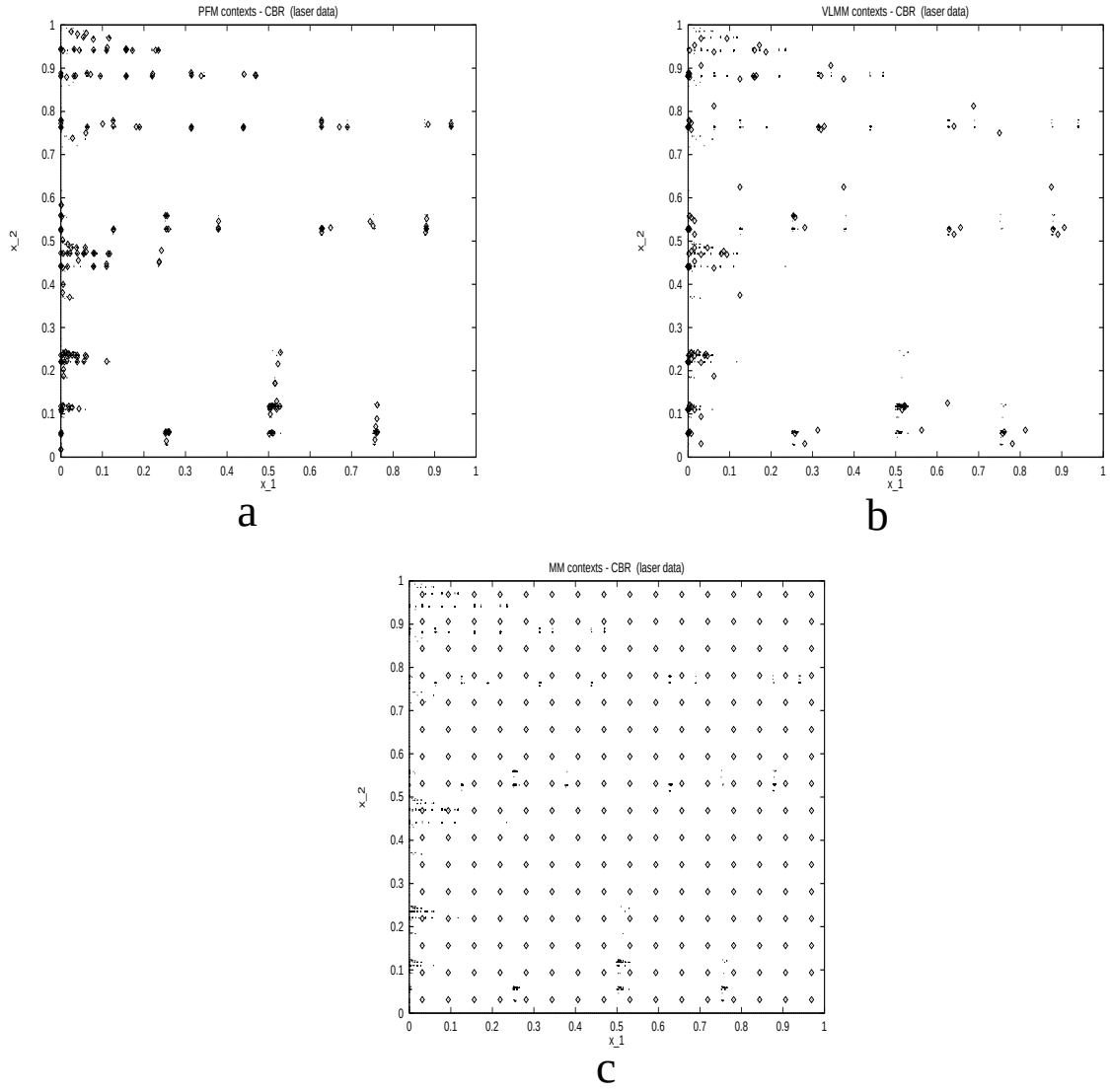

Figure 7: Chaos L-block representation $CBR_{L,k}(S)$ of the laser training sequence S (dots), $L = 20$, $k = \frac{1}{2}$. Diamonds represent the block representations of prediction contexts of the PFM (a), VLMM (b) and MM (c) AIC representatives.

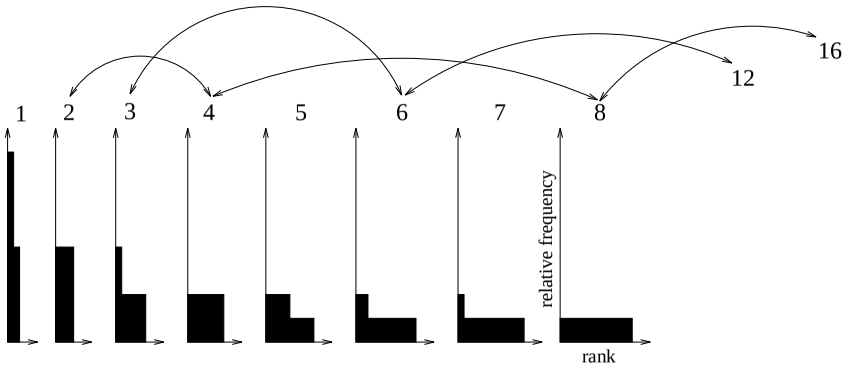

Figure 8: Plots of self-similar rank-ordered block distributions of the Feigenbaum sequence for different block lengths (indicated by the numbers above the plots). The self similarity relates block distributions for block lengths $2^g \rightarrow 2^{g+1}$, $3\ 2^{g-1} \rightarrow 3\ 2^g$, $g \geq 1$ (connected by arrows).

to simply iteratively change the parameters by a small amount and save the resulting VLMMs (as done in the laser data experiment). Instead, one had to spent a fair amount of time to find the critical parameter values.

In contrast, building PFMs and SFMs proceeded as naturally as in the previous experiment. Dynamic cell structures covered the training sequence geometric representation $CBR_{L,k}(S)$ with increasing number of codebook vectors that later became the prediction contexts and states of PFMs and SFMs, respectively.

As in the previous experiment, the models were used to generate sequences G of length equal to the length of the training sequence S. Then, we computed L_1 distances[16] between the n-block distributions on S and G,

$$d_n(S,G) = \sum_{w \in \mathcal{A}^n} |\hat{P}_{S,n}(w) - \hat{P}_{G,n}(w)|,$$

where $\hat{P}_{S,n}$ and $\hat{P}_{G,n}$ are the empirical n-block frequencies in S and G respectively.

With each model $\mathcal{M}$, we associated its modeling horizon $n(\mathcal{M})$ by letting the model $\mathcal{M}$ generate 10 sequence generation realizations $G_1, ..., G_{10}$, and evaluating

$$\delta_n(\mathcal{M}) = \max_{i=1,2,...,10} d_n(S, G_i).$$

The modeling horizon $n(\mathcal{M})$ of the model $\mathcal{M}$ is defined by

$$\forall n \leq n(\mathcal{M}),\ \delta_n(\mathcal{M}) \leq \Delta \quad \text{and} \quad \delta_{n(\mathcal{M})+1}(\mathcal{M}) > \Delta.$$

In our experiments, the threshold[17] Δ was set to 0.005.

Figure 9 interprets the growing ability of PFMs, SFMs and VLMMs to model the metric structure of allowed blocks in the Feigenbaum sequence S.

[16] Feigenbaum sequence n-block distributions have just one or two probability levels. Therefore, in this case, it is not necessary to use the thermodynamic entropy rate spectra (as done in the previous experiment) to scan the n-block distributions across all probability levels and compute the spectra distances concentrating on statistically well-determined higher temperature statistics. We use the L_1 distance to measure the disproportions between the Feigenbaum and the model generated distributions.

[17] We observed that either $\delta_n(\mathcal{M}) \in (0, 0.005]$ (the model mimics the n-block distribution successfully), or $\delta_n(\mathcal{M}) >> 0.005$ (the model fails in modeling the n-block distribution).

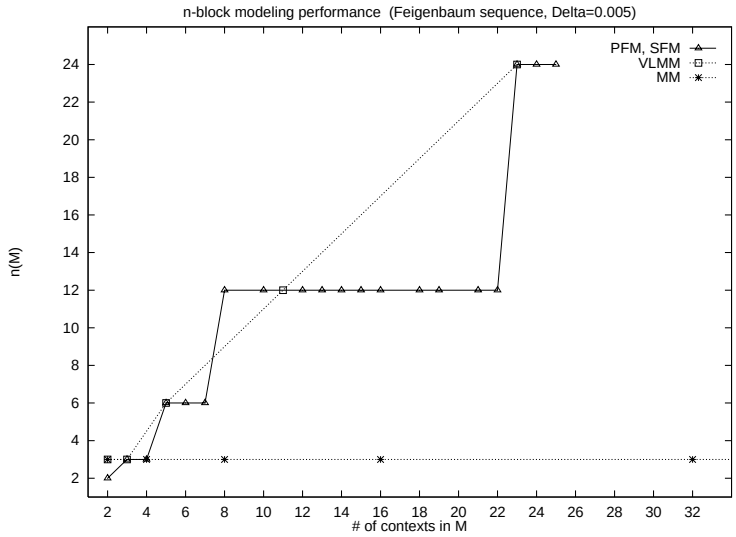

Figure 9: Modeling horizons $n(\mathcal{M})$ of models $\mathcal{M}$ built on the Feigenbaum sequence as a function of the number of prediction contexts in $\mathcal{M}$.

The classical MM totally fails in this experiment, since the context length 5 is far too small to enable the MM to mimic the complicated subsequence structure in S. PFMs, SFMs and VLMMs quickly learn to explore a limited number of deep prediction contexts and perform comparatively well.

The jumps in the modeling horizon graph of PFMs and SFMs on figure 9 can be understood through state transition diagrams of the SFMs.

While the machine M_4 in figure 10a can model only blocks of length 1,2 and 3, the introduction of an additional transition state in the machine M_5 shown in figure 10b enables the latter machine to model blocks of length up to 6.

Only three consecutive `2`'s are allowed in the training Feigenbaum sequence S. The loop on symbol `2` in the state 1 of the machine M_4 is capable of producing blocks of consecutive `2`'s of any length. So, the n-block distribution, $n \geq 4$, cannot be properly modeled by the machine M_4.

The state 1 in the machine M_4 is split into two machine M_5 states 1.a and 1.b. Any number of 4-blocks `2212` can be followed by any number of 2-blocks `12` and vice versa. This is fine as long as we study structure of the 6-block distribution.

Moving to higher block lengths, we find that once the 4-block `2212` is followed by the 2-block `12`, another copy of the 2-block `12` followed by the 4-block `2212` must appear. This 12-block rule is implemented by the machine M_8 in figure 11b. The machine M_8 is created from the machine M_7 in figure 11a by splitting the state 3.a into two states 3.a and 3.c. The machine M_7 with 7 states is equivalent to the machine M_5 (figure 10a) with 5 states: states 2.a, 2.b and 3.a, 3.b in M_7 are equivalent to states 2 and 3, respectively, in M_5.

State splitting responsible for the third jump in the modeling horizon graph between the SFMs M_{22} and M_{23} with 22 and 23 states, respectively, is illustrated in figure 12. Symbols `A` and `B` stand for the 4-blocks `1212` and `2212`, respectively. The machine M_{22} is equivalent to the machine M_8. State splitting in the middle left branch of the machine M_{22} removes the two lower cycles `BAB, B`, and creates a single larger cycle `BBAB` in the machine

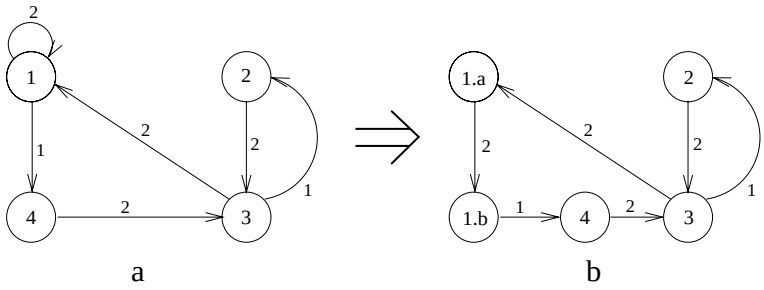

Figure 10: State transition diagrams of the SFMs constructed on one-dimensional chaos L-block representation $CBR_{L,k}(S)$, $L = 30$, $k = \frac{1}{2}$, of the training sequence S. The machines M_4 (a) and M_5 (b) were obtained by quantizing $CBR_{L,k}(S)$, via dynamic cell structures with 4 and 5 centers respectively. State transitions are labeled only with the corresponding symbols, since the transition probabilities $T_{i,j,s}$ are uniformly distributed, i.e. $T_{i,j,s} = 1/N_i$, where N_i is the number of arcs leaving the state i.

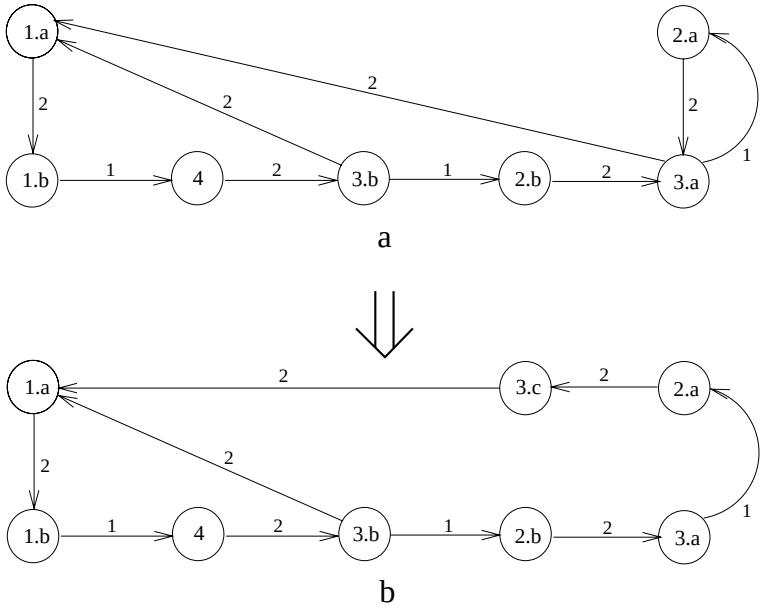

Figure 11: SFMs M_7 and M_8 built on geometric representation $CBR_{L,k}(S)$ of the Feigenbaum sequence S quantized into 7 (a) and 8 (b) compartments, respectively. Construction details are described in caption to the previous figure.

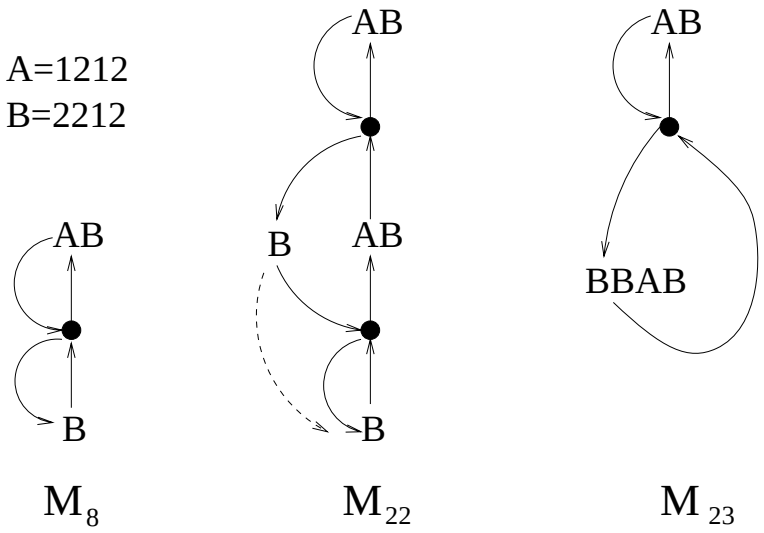

Figure 12: Schematic representation of SFMs built on geometric representation of the Feigenbaum sequence. Symbols `A` and `B` stand for the 4-blocks `1212` and `2212`, respectively. The machine M_{22}, obtained from a codebook with 22 centers, is equivalent to the machine M_8 (see also the previous figure). State splitting in the middle left branch of the machine M_{22} (dashed line) removes the two lower cycles `BAB, B`, and creates a single larger cycle `BBAB` in the machine M_{23}.

M_{23}. This machine correctly implements the training sequence block distributions for blocks of length up to 24.

Variable memory length Markov models implement the same subsequence constraints as their fractal counterparts SFMs. Figures 13a and 13b present VLMMs N_5 and N_{11} with 5 and 11 prediction contexts, respectively. The VLMMs are shown as probabilistic suffix automata with states labeled by the corresponding suffices. The VLMM N_5 is isomorphic to the SFM M_5 in figure 10b, and the VLMM N_{11} is equivalent to the SFM M_8 in figure 11b. Although not shown here, the VLMM with 23 prediction contexts is isomorphic to the SFM M_{23} schematically presented in figure 12.

5 Discussion

The main advantage of our approach is the self-organizing character of constructing fractal-based predictive models PFMs and SFMs with increasing model size. Algorithms like dynamic cell structures (Bruske, & Sommer, 1995), or growing cell structures (Fritzke, 1995) cover the spatial training sequence L-block representation with increasingly large codebooks in a natural and self-organized manner. Predictive models constructed on such codebooks can be compared through a model selection criterion (e.g. Akaike information criterion (AIC) used in this study) selecting a model class representative.

This is an important issue that has attained little attention in the VLMM literature. Usually, the results are presented only for a few selected models, stressing the memory requirement advantage of VLMMs over the classical MMs. Little is said about how a particular model was selected and how difficult it was to arrive at the presented solution

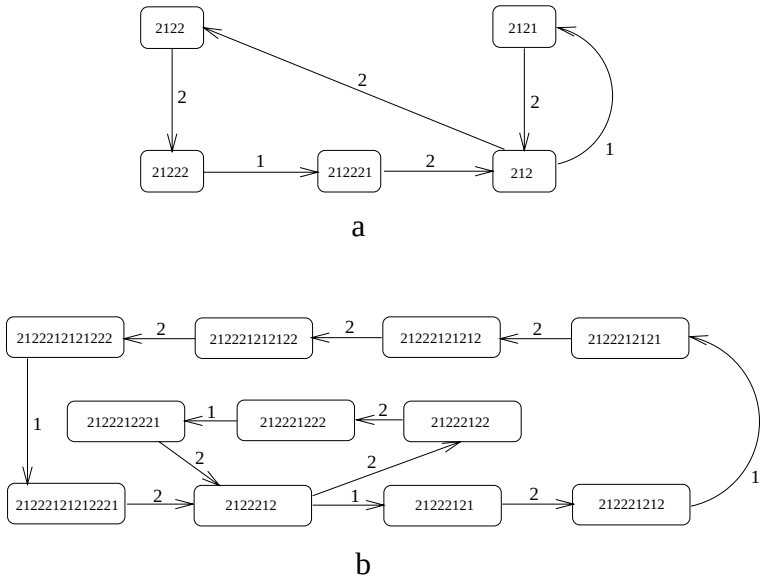

Figure 13: VLMMs N_5 (a) and N_{11} (b) built on the Feigenbaum sequence. The VLMMs are are shown as probabilistic suffix automata with states labeled by the corresponding suffices. As with SFMs in this experiment, the state transition probabilities are uniformly distributed.

(see, for example (Ron, Singer, & Tishby, 1996; Ron, Singer, & Tishby, 1994)). Bühlmann (1997) mentions difficulties in selecting the VLMM construction parameter values and suggests to use the AIC as a model selection tool.

Guyon and Pereira (1995) study two ways of constructing increasingly complex VLMMs: by increasing the source memory L with other construction parameters kept fixed, or by fixing a (long enough) source memory L and gradually changing a single parameter on which all the other construction parameters functionally depend. The latter scheme was experimentally shown to yield a superior performance (Guyon, & Pereira, 1995). We used the latter scheme in the experiments described in this paper.

As shown in section 4.2, the construction parameters' selection is a non-intuitive task that may require a lot of interactive steps. In this respect, the construction of PFMs and SFMs is more intuitive (the number of codebook vectors directly corresponds to the number of predictive contexts), easier to automatize (growth of predictive models is directed by the codebook growth in the self-organizing quantization algorithms) and often faster.

For example, on a Pentium linux machine, it takes less than a minute to construct the chaos L-block representation $CBR_{L,k}(S)$ of the Feigenbaum sequence. Dynamic cell structures (DCS) quantize the $CBR_{L,k}(S)$ into a series of codebooks with 2,3,...,30, centers in about 5 minutes. Construction of a series of fractal-based predictive models PFM (SFM) is finished in 5 minutes. On the other hand, we spent three hours of interactive work to obtain the VLMM series used in the Feigenbaum sequence experiment.

More importantly, our fractal-based predictive models PFMs and SFMs outperform the classical and variable length Markov models in the laser data experiment, and achieve the VLMM modeling performance on the Feigenbaum sequence.

Compared with the Feigenbaum sequence one- or two-level n-block distributions, the range of probability levels corresponding to blocks contained in the laser-produced training sequence is quite rich. We employ the thermodynamic entropy rate spectra to scan the laser sequence high probability n-block structure. The modeling behavior of the studied models for blocks of increasing length is assessed using the entropy rate spectra distances computed on the training and model generated sequences. With respect to these measures, as well as with respect to the block length independent Lempel-Ziv entropy and cross-entropy estimates, the PFMs and SFMs outperform the MMs and VLMMs.

The experiment with the artificially generated Feigenbaum sequence tests, on each model size level, how quickly can the models find a set of specialized deep prediction contexts needed to model a rather peculiar training sequence subsequence structure. In contrast to the laser data experiment, where the training sequence is generated by a real laser in a chaotic regime, the setting of this experiment is different in that

- the one- and two-level n-block distributions allow us to use the L_1 distance between the training and model generated sequences as a modeling performance criterion.

- the predictive models need deep prediction contexts. This is the case where the classical Markov models cannot succeed and the full power of admitting a limited number of variable length contexts can be exploited.

- the topological and metric structures of the Feigenbaum sequence are well-understood (Crutchfield, & Young, 1990; Freund, Ebeling, & Rateitschak, 1996). This enables us to monitor the construction of fractal-based models of growing size by analysing the SFM state transition diagrams for models corresponding to jumps in the modeling horizon graph and comparing them to the theoretical Feigenbaum sequence model. Indeed, The machines M_5, M_8 and M_{23} constitute increasingly better approximations to the infinite self-similar Feigenbaum machine described by Crutchfield and Young (1990).

5.1 Final remarks

1. We also tried other quantization techniques like the classical Kohonen self-organizing feature maps (SOFM) (Kohonen, 1990), SOFM with the star topology of neuron field (Tiňo, & Šajda, 1995), or deterministic annealing based hierarchical clustering (Rose, Gurewitz, & Fox, 1990). In all cases, the quantization of the training sequence chaos L-block representation $CBR_{L,k}(S)$ yielded predictive models of quality comparable to that of the models obtained via the K-means or DCS clustering. Clustering via deterministic annealing took enormous time without any apparent improvement in the resulting predictive models.

2. In some cases, topological ordering of codebook vectors in vector quantization of the training sequence spatial representation may be beneficial. For example, in the Feigenbaum sequence experiment, dynamic cell structures created new codebook vectors, one at a time, without much change to the codebook distribution and topology. Speaking in terms of the extracted SFMs, the state transition diagrams and the state labeling before and after inclusion of a new codebook vector differed only locally and the state splitting effect was immediately detectable.

3. The fractal-based models PFMs and SFMs depend on the cluster density in the $CBR_{L,k}(S)$, that is controlled with the contraction parameter $k \in (0, \frac{1}{2}]$. Smaller k's yield more dense clusters. Furthermore, quantization of the $CBR_{L,k}(S)$ is controlled by the magnification factor (Ritter, & Schulten, 1986; Bauer, Der, & Herrmann, 1996) of the used vector quantization scheme. The magnification factor relates[18] the frequency of codebook vectors in a quantized region with the frequency of points from $CBR_{L,k}(S)$ in that region. One can find a formal relationship among the CBR-contraction factor k, magnification factor of the vector quantizer and the topology of the SFM state transition diagrams. This and other related issues are currently under investigation and will be published elsewhere (Tiňo, & Dorffner, 1998).

4. For k close to $\frac{1}{2}$, geometric representations of completely different n-blocks may lie close to each other. This happens, for example, for blocks 444...41 and 333...32 over the alphabet $\{1, 2, 3, 4\}$, geometrically represented through the iterative function system (4) acting on $[0, 1]^2$, with $t_1 = (0, 0)$, $t_2 = (1, 0)$, $t_3 = (0, 1)$ and $t_4 = (1, 1)$. As a remedy, one may lower the contraction ratio k. In our experiments, however, we did not notice any serious downfall in the model quality for $k = \frac{1}{2}$. The issue of optimal contraction ratio with respect to a given training sequence and vector quantizer is also being currently investigated.

5. Is is only fair to note that even though the predictive models PFMs and SFMs emerge from our experiments as potentionally interesting and favorable alternatives to VLMMs, so far, they lack a sound theoretical background comparable to that supporting the use of VLMMs (Ron, Singer, & Tishby, 1996; Weinberger, Rissanen, & Feder, 1995; Bühlmann, 1997). Proceeding in this direction, we have theoretically analysed the multifractal properties of the basis for our predictive models' construction – the chaos n-block sequence representation (Tiňo, 1998), and found a relationship among the chaos block representation contraction factor, magnification factor of the vector quantizer and the topology of the SFM state transition diagrams (Tiňo, & Dorffner, 1998).

Acknowledgements

We wish to thank Owen Kelly for helpful comments on variable length Markov models. We are grateful to Igor Farkaš for performing the dynamic cell structures simulations. This work was supported by the Austrian Science Fund (FWF) within the research project "Adaptive Information Systems and Modeling in Economics and Management Science" (SFB 010). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Science and Transport.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19, 716–723.

Aurenhammer, F. (1991). Voronoi Diagrams - Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, 3, 345–405.

[18] under asymptotic considerations

Barnsley, M.F. (1988). *Fractals everywhere*. Academic Press, New York.

Bauer, H.U., Der, R., & Herrmann, M. (1996). Controlling the magnification factor of self-organizing feature maps. *Neural Computation*, 8, 757–771.

Beck, C., & Schlögl, F. (1995). *Thermodynamics of chaotic systems*. Cambridge University Press, Cambridge, UK.

Brillinger, D.R. (1994). Examples of scientific problems and data analysis in demography, neurophysiology and seismology. *J.Comp. and Graph. Statistics*, 3, 1–22.

Bruske, J., & Sommer, G. (1995). Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, 4, 845–865.

Bühlmann, P. (1997). Variable length markov models. Technical Report 479, University of California, Berkeley.

Bühlmann, P. (1998). Extreme events from return-volume process: adiscretization approach for complexity reduction. *Applied Financial Economics*, to appear.

Buhmann, J.M. (1995). Learning and data clustering. In Arbib, M. (Eds.), *Handbook of Brain Theory and Neural Networks*. Bradfort Books, MIT Press.

Crutchfield, J.P., & Young, K. (1990). Computation at the onset of chaos. In Zurek, W.H. (Eds.), *Complexity, Entropy, and the physics of Information, SFI Studies in the Sciences of Complexity* (Vol 8). Addison-Wesley.

Freund, J., Ebeling, W., & Rateitschak, K. (1996). Self-similar sequences and universal scaling of dynamical entropies. *Physical Review E*, 5, 5561–5566.

Fritzke, B. (1995). Growing cell structures - a self-organizing network for unsupervised and supervised training. *Neural Networks*, 9, 1441–1460.

Grassberger, P. (1991). Information and complexity measures in dynamical systems. In Atmanspacher, H., & Scheingraber, H. (Eds.), *Information Dynamics*. Plenum Press, New York.

Guyon, I., & Pereira, F. (1995). Design of a linguistic postprocessor using variable memory length markov models. In *Proceedings of International Conference on Document Analysis and Recognition* (pp. 454–457), Monreal, Canada: IEEE Computer Society Press.

Jeffrey, J. (1990). Chaos game representation of gene structure. *Nucleic Acids Research*, 8, 2163–2170.

Khinchin, A.I. (1957). *Mathematical Foundations of Information Theory*. Dover Publications, New York.

Kohonen, T. (1990). The self–organizing map. *Proceedings of the IEEE*, 9, 1464–1479.

Laird, P., & Saul, R. (1994). Discrete sequence prediction and its applications. *Machine Learning*, 15, 43–68.

Li, W. (1997). The study of correlation structures of dna sequences: a critical review. *Computer and Chemistry*, 4, 257–272.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297), Berkeley, CA: University of California Press.

McCauley, J.L. (1994). *Chaos, Dynamics and Fractals: an algorithmic approach to deterministic chaos*. Cambridge University Press.

Nadas, A. (1984). Estimation of probabilities in the language model of the ibm speech recognition system. *IEEE Trans. on ASSP*, 4, 859–861.

Oliver, J.L., Bernaola-Galván, P., Guerrero-Garcia, J., & Román Roldan, R. (1993). Entropic profiles of dna sequences through chaos-game-derived images. *Journal of Theor. Biology*, 160, 457–470.

Prum, B., Rodolphe, F., & deTurckheim, E. (1995). Finding words with unexpected frequencies in deoxyribonucleic acid sequences. *Journal of Royal Statistical Society*, B 57, 205–220.

A. Rényi, A. (1959). On the dimension and entropy of probability distributions. *Acta Math. Hung.*, 10, 193.

Rissanen, J. (1983). A universal data compression system. *IEEE Trans. Inform. Theory*, 5, 656–664.

Ritter, H., & Schulten, K. (1986). On the stationary state of the kohonen's self-organizing sensory mapping. *Biol. Cybern.*, 54, 99–106.

Román-Roldan, R., Bernaola-Galván, P., & Oliver, J.L. (1994). Entropic feature for sequence pattern through iteration function systems. *Pattern Recognition Letters*, 15, 567–573.

Ron, D., Singer, Y., & Tishby, N. (1994). The power of amnesia. In *Advances in Neural Information Processing Systems 6*: Morgan Kaufmann.

Ron, D., Singer, Y., & Tishby, N. (1996). The power of amnesia. *Machine Learning*, 25.

Rose, K., Gurewitz, E., & Fox, G.C. (1990). Statistical mechanics and phase transitions in clustering. *Physical Review Letters*, 8, 945–948.

Tiňo, P., & Šajda, J. (1995). Learning and extracting initial mealy machines with a modular neural network model. *Neural Computation*, 4, 822–844.

Tiňo, P. (1998). Spatial representation of symbolic sequences through iterative function systems. Technical Report TR-98-17, Austrian Research Institute for Artificial Intelligence, Austria.

Tiňo, P. & Dorffner, G. (1998). Recurrent neural networks with iterated function systems dynamics. *International ICSC/IFAC Symposium on Neural Computation*, accepted.

Weinberger, M.J., Rissanen, J.J., & Feder, M. (1995). A universal finite memory source. *IEEE Transactions on Information Theory*, 3, 643–652.

Willems, F.M.J., Shtarkov, Y.M., & Tjalkens, T.J. (1995). The context tree weighting method: basic properties. *IEEE Trans.Info.Theory*, 3, 653–664.

Young, K., & Crutchfield, J.P. (1993). Fluctuation spectroscopy. In Ebeling, W. (Eds.), *Chaos, Solitons, and Fractals, special issue on Complexity.*

Ziv, J., & Merhav, N. (1993). A measure of relative entropy between individual sequences with application to universal classification. *IEEE Transactions on Information Theory*, 4, 1270–1279.