

# Guessing Probability Distributions from Small Samples

Thorsten Pöschel,<sup>1,2</sup> Werner Ebeling,<sup>1</sup> and Helge Rosé<sup>1</sup>

Received February 16, 1995

---

We propose a new method for the calculation of the statistical properties, e.g., the entropy, of unknown generators of symbolic sequences. The probability distribution  $p(k)$  of the elements  $k$  of a population can be approximated by the frequencies  $f(k)$  of a sample provided the sample is long enough so that each element  $k$  occurs many times. Our method yields an approximation if this precondition does not hold. For a given  $f(k)$  we recalculate the Zipf-ordered probability distribution by optimization of the parameters of a guessed distribution. We demonstrate that our method yields reliable results.

---

**KEY WORDS:** Entropy estimation; information science.

## 1. INTRODUCTION

Given a statistical population of discrete events  $k$  generated by a stationary dynamic process, one of the most interesting statistical properties of the population and hence of the process is its entropy. If the sample space, i.e., the number of different elements which are allowed to occur in the population, is small compared with the size of a drawn sample, one can approximate the probabilities  $p(k)$  of the elements  $k$  by their relative frequencies  $f(k)$  and one finds for the observed entropy  $H_{\text{obs}}$

$$H = -\sum_k p(k) \log p(k) \approx -\sum_k f(k) \log f(k) = H_{\text{obs}} \quad (1)$$

---

<sup>1</sup> Humboldt-Universität zu Berlin, Institut für Physik, D-10099 Berlin, Germany. <http://summa.physik.hu-berlin.de:80/~thorsten/>.

<sup>2</sup> James Franck Institute, University of Chicago, Chicago, Illinois 60637.

If the number of the allowed different events is not small compared with the size of the sample the approximation  $p(k) \approx f(k)$  yields dramatically wrong results. In this case the knowledge of the frequencies is not sufficient to determine the entropy. The aim of this paper is to provide a method to calculate the entropy and other statistical characteristics for the case that the approximation (1) does not hold.

An interesting example of such systems are subsequences (words) of length  $n$  of symbolic sequences of length  $L$  written using an alphabet of  $\lambda$  letters. Examples are biosequences like DNA ( $\lambda = 4$ ,  $L \lesssim 10^9$ ), literary texts ( $\lambda \approx 80$  letters and punctuation marks,  $L \lesssim 10^7$ ), and computer files ( $\lambda = 2$ ,  $L$  arbitrary). For the case of biosequences there is a variety of  $\lambda^n = 1,048,576$  different words of length  $n = 10$ . To measure the probability distribution of the words directly by counting their frequencies we need at least a sequence of length  $10^8$  to have reliable statistics. Therefore the ensemble of subsequences of length  $n$  is a typical example where the precondition does not hold. For the entropy of  $N$  words of length  $n$  in a Bernoulli sequence with  $\lambda = 2$  where both symbols occur with the same probability, one finds the exact result  $H^{(n)} = n \log \lambda$ . When calculating the observed entropy  $H_{\text{obs}}^{(n)}$  based on counted relative frequencies, the observed entropy values are correct for small word length  $n$  when we can approximate the probabilities by the relative frequencies. From a certain word length on ( $n > n^*$ ) the observed entropies are significantly below the exact values, even for very large samples  $H_{\text{obs}}^{(n)} < H^{(n)}$ . For  $N = 10^4$  we found  $n^* \approx 12$ , for  $N = 10^6$  we found  $n^* \approx 18$ .

Under several strong preconditions the probabilities of words in sequences can be estimated from the frequencies using various correction methods.<sup>(1-3)</sup> The advanced algorithm proposed in ref. 3 is based on a theorem by McMillan and Khinchin<sup>(4)</sup> saying that for word length  $n \rightarrow \infty$  the frequencies of the admitted substrings of a sequence are equally distributed. If one is interested in the entropies for *finite* words, however, the theoretical basis to apply this theorem is weak and there is no evidence about the reliability of the results. Moreover, this theorem is proven for Markov sequences only. In sequences gathered from natural languages, biosequences and other natural or artificial sources it is very unlikely that the probabilities of the words of interesting length, e.g., words or sentences for languages, amino acids, or elements of the hidden "DNA language" for biosequences, are equally distributed. Otherwise we would have to assume that all English five-letter words are equally frequent. Certainly this is not the case.

## 2. DESCRIPTION OF THE METHOD

To calculate the entropy of a distribution it is not necessary to determine for each event  $k$  the probability  $p(k)$ . It is sufficient to determine the values of the probabilities without knowing which probability belongs to which event. Generally, if we assume we have  $K$  events, there are  $K!$  different relations  $k \leftrightarrow p$ . We need not determine one particular (the correct) relation but only one arbitrary one of them. Hence the calculation of the entropy is  $K!$  times easier than determining the probability  $p(k)$  for each event  $k$ . We assume a special order where the first element has the largest probability, the second one the second largest, etc. We call this distribution Zipf-ordered. Zipf ordering means that the probabilities of the elements are ordered according to their rank and therefore the distribution  $p(k)$  is a monotonically decaying function. The following procedure describes how to reconstruct the Zipf-ordered probability distribution  $p(k)$  from a finite sample, provided we have some reason to expect (to guess) the parametric form of the probability distribution. As an example we use a simple distribution  $p(k, \alpha, \beta, \gamma)$  with  $k = 1, 2, \dots$  consisting of a linearly decreasing and a constant part,

$$p(k) = \begin{cases} (2 - \alpha\beta)/2\gamma + \alpha(1 - k/\beta), & 1 \leq k < \beta \\ (2 - \alpha\beta)/(2\gamma), & \beta \leq k \leq \gamma \\ 0, & k > \gamma \end{cases} \quad (2)$$

Then the algorithm runs as follows:

i. Find the frequencies  $F(k)$  for the  $N$  events  $k$  and order them according to their value (Zipf order). The index  $k$  runs over all *different* events occurring in the sample ( $k \in \{1, \dots, K^{\text{MAX}}\}$ ). Note: there are  $N$  events but only  $K^{\text{MAX}}$  different ones. Normalize this distribution  $F_1(k) = F(k)/N$ . There are various sophisticated algorithms to find the frequencies of large samples and to order them (e.g., ref. 5). As in earlier papers<sup>(6)</sup> we applied for finding the elements a “hashing” method and for sorting a mixed algorithm consisting of “Quicksort” for the frequent elements and “Distribution Counting” for the long tail of elements with low frequencies.

ii. Guess initial conditions for the parameters (in our case  $\alpha$ ,  $\beta$ , and  $\gamma$ ).

iii. Generate  $M$  samples of  $N$  random integers ( $RI_k^m$ ,  $k = 1, \dots, N$ ,  $m = 1, \dots, M$ ) according to the parametric probability distribution  $p(k, \alpha, \beta, \gamma)$ . In the following examples we used  $M = 20$ . Order each of the

samples according to the ranks  $f_i(k, \alpha, \beta, \gamma)$  ( $i = 1, \dots, M$ ). Average over the  $M$  ordered samples

$$\overline{f(k, \alpha, \beta, \gamma)} = \frac{1}{M} \sum_{i=1}^M f_i(k, \alpha, \beta, \gamma) \tag{3}$$

with  $k \in \{1, k^{\max}\}$  and  $k^{\max} = \max(k_i^{\max}, (i = 1, \dots, M))$ . Since we want to determine the averaged or typical Zipf-ordered distribution, it is important to order the elements first and then to average. Normalize the averaged distribution of the frequencies

$$\overline{f_i(k, \alpha, \beta, \gamma)} = \left( \sum_{k=0}^{k^{\max}} \overline{f(k, \alpha, \beta, \gamma)} \right)^{-1} \overline{f(k, \alpha, \beta, \gamma)} \tag{4}$$

iv. Measure the deviation  $D$  between the normalized averaged simulated frequency distribution  $\overline{f_i(k, \alpha, \beta, \gamma)}$  and the frequency distribution  $F_1(k)$  of the given sample according to a certain rule, e.g.,

$$D = \sum_{k=1}^K \left( \frac{\overline{f_i(k, \alpha, \beta, \gamma)}}{F_1(k)} - 1 \right)^2, \quad K = \max\{k^{\max}, K^{\text{MAX}}\} \tag{5}$$

v. Change the parameters of the guessed probability distribution  $p(k)$  (in our case the parameters  $\alpha, \beta$ , and  $\gamma$ ) due to an optimization rule (e.g., ref. 7) which minimizes  $D$  and proceed with the third step until the deviation  $D$  is sufficiently small.

vi. Extract the interesting statistical properties out of the probability distribution  $p(k)$  using the parameters  $\alpha^*, \beta^*$ , and  $\gamma^*$  which have been gathered during the optimization process.

### 3. EXAMPLES

#### 3.1. Entropy of Artificial Sequences

We generated a statistical ensemble  $N = 10^4$  according to the probability distribution (2) with  $\alpha = 9.0 \times 10^{-6}$ ,  $\beta = 10,000$ , and  $\gamma = 50,000$ . Figure 1 (solid lines) shows the probability distribution  $p(k)$  and the Zipf ordered frequencies  $f(k)$ .

Optimizing the parametric guessed probability distribution using the proposed method, we find for the optimized parameters  $\alpha^* = 9.22 \times 10^{-6}$ ,  $\beta^* = 12,900$ , and  $\gamma^* = 50,000$ , i.e., the guessed and the actual distributions fall almost together. Since we know the original probability distribution [Eq. (2)] we can compare its exact entropy with the entropy of the guessed

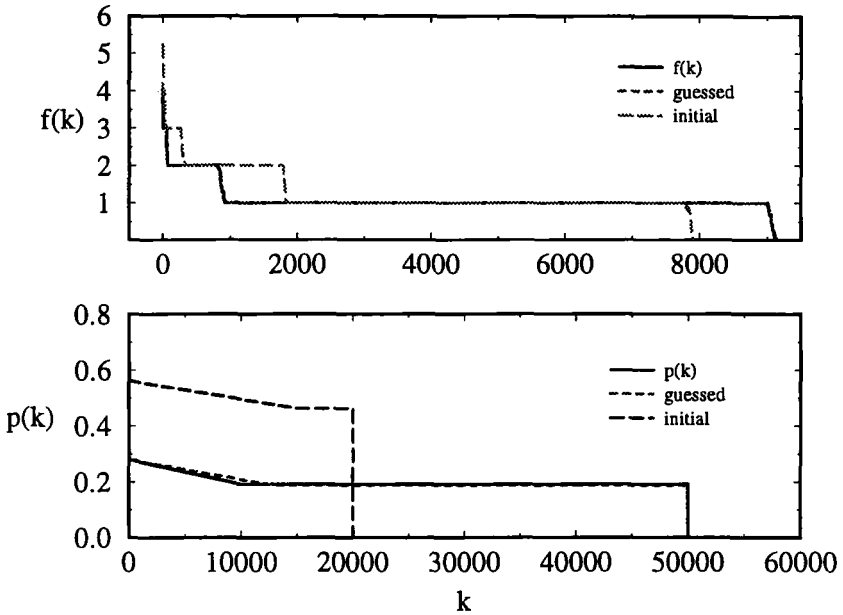


Fig. 1. The probability distribution  $p(k)$  [Eq. (2)] and the Zipf-ordered frequencies  $f(k)$  corresponding to this distribution. The dashed lines which almost cannot be distinguished from the solid lines display the guessed distributions. The initial distributions before optimization are drawn with wide dashes.

probability distribution  $H_{\text{guess}}$  and with the observed entropy  $H_{\text{obs}}$  due to Eqs. (6) and (7):

$$H_{\text{guess}}(k, \alpha^*, \beta^*, \gamma^*) = - \sum p(k, \alpha^*, \beta^*, \gamma^*) \log p(k, \alpha^*, \beta^*, \gamma^*) \quad (6)$$

$$H_{\text{obs}}(k) = - \sum F_1(k) \log F_1(k) \quad (7)$$

We found  $H_{\text{obs}} = 9.0811$  and  $H_{\text{guess}} = 10.8147$ ; the exact value according to  $p(k, \alpha, \beta, \gamma)$  [Eq. (2)] is  $H = 10.8188$ .

Now we try to guess a probability of a more complicated form:

$$p(k) = \begin{cases} \alpha(k - \varepsilon)^{-1/3}, & 1 \leq k < \beta \\ \phi k^{-\delta}, & \beta \leq k \leq \gamma \\ 0, & k > \gamma \end{cases} \quad (8)$$

(As we will show below, this function approximates the probability distribution of the words in an English text.) The variables  $\alpha$  and  $\phi$  can be eliminated due to the normalization and continuity conditions. The test

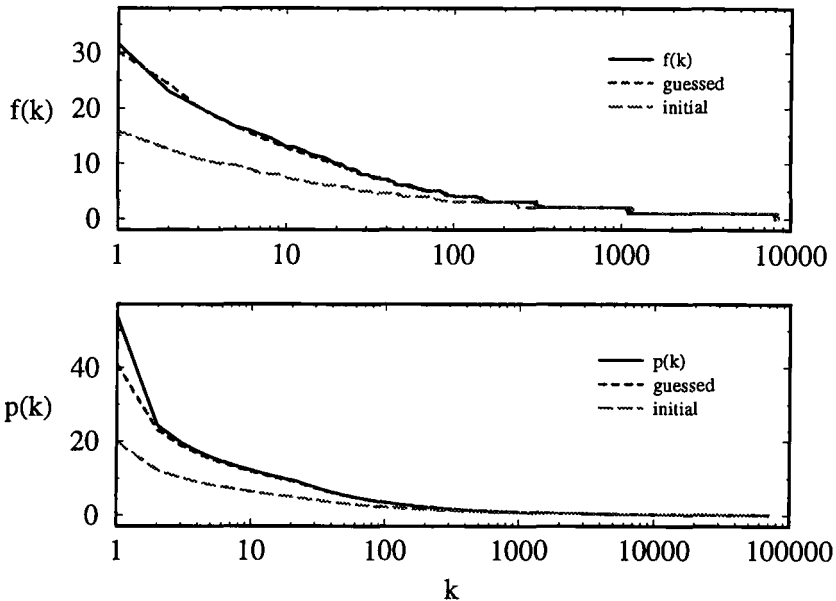


Fig. 2. The original and guessed probability distributions and the Zipf-ordered frequencies for the distribution in Eq. (8) ( $N=10^4$ ).

sample of size  $N=10^4$  was generated using  $\varepsilon=0.9$ ,  $\beta=22$ ,  $\delta=0.64$ , and  $\gamma=70,000$ . After the optimization we guess the parameters  $\varepsilon^*=0.79$ ,  $\beta^*=21.9$ ,  $\delta^*=0.63$  and  $\gamma^*=65,000$ . Figure 2 shows the original and the guessed probability distributions and the Zipf-ordered frequencies for both cases. The guessed entropy  $H_{\text{guess}}=10.5053$  approximates the exact value  $H=10.5397$  very well, while the observed entropy  $H_{\text{obs}}=8.8554$  shows a clear deviation from the correct value.

### 3.2. Words in an English Text

With the ansatz (8) we tried to guess the probability distribution of the words of different length  $n$  in the text *Moby Dick* by H. Melville.<sup>(8)</sup> The text was mapped to an alphabet of  $\lambda=32$  letters as described in ref. 9. Depending on overlapping or nonoverlapping counting of the words, we expect different results. We note that overlapping counting is statistically not correct since the elements of the sample are not statistically independent; however, only overlapping counting yields enough words to get somehow reasonable results for the observed entropy. We will show that our method works in both cases, overlapping and nonoverlapping.

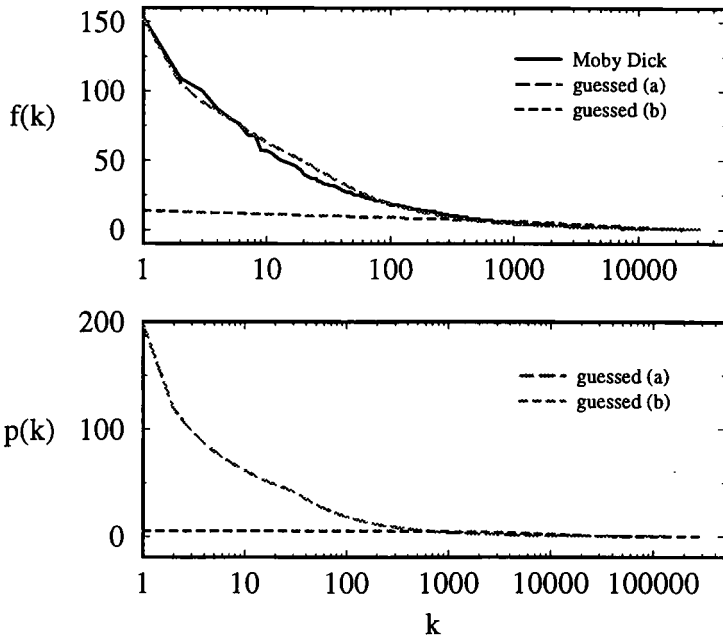


Fig. 3. Zipf-ordered frequencies of words of length  $n=6$  in *Moby Dick*. The curves guessed (a) and (b) (top) display the frequency distributions which have been reproduced using the guessed probability distributions in the bottom figure according to (a) Eq. (8) and (b) Eq. (9).

Figure 3 shows the ordered frequencies of  $N=5 \times 10^4$  words of the length  $n=6$ . The optimized distribution, Eq. (8), reproduces the original frequency distribution (*Moby Dick*) with satisfying accuracy.

Using the ansatz (8), we found  $\varepsilon^*=0.73$ ,  $\beta^*=31$ ,  $\delta^*=0.70$ , and  $\gamma^*=129,890$ . This calculation was carried out for various word lengths  $n$ . Figure 4 shows the entropies  $H_{\text{obs}}^{(n)}$  and  $H_{\text{guess}}^{(n)}$  according to Eqs. (6) and (7) as a function of  $n$ . All results obtained have been derived from a set of  $5 \times 10^4$  nonoverlapping words taken from the text of length  $L=10^6$ . When we count overlapping words, we find surprisingly that the entropy is quite insensitive (see curves using filled and empty diamonds in Fig. 4). The rather difficult problem of overlapping or nonoverlapping counting will be addressed in detail in ref. 10. Since the exact probability distribution for the words in *Moby Dick* is unknown, we compare the guessed entropy (crosses) with the observed entropy (empty diamonds: overlapping counting; full diamonds: nonoverlapping counting) and an estimation of the entropy using an extrapolation method,<sup>(6)</sup> all based on the same set of data

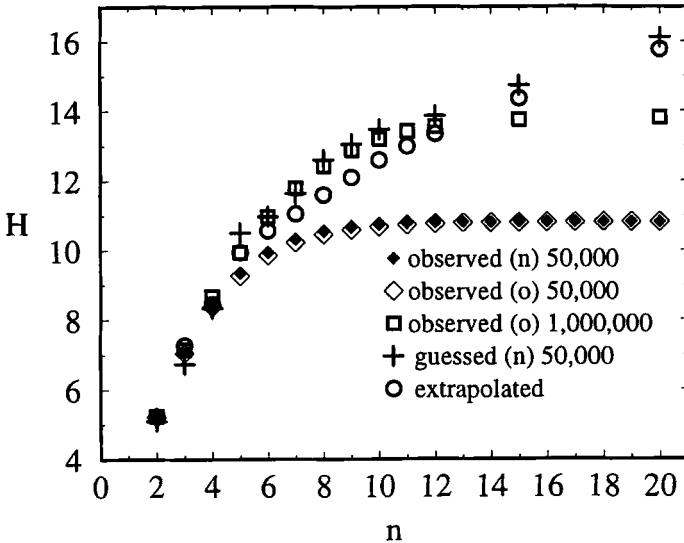


Fig. 4. Observed entropy  $H_{\text{obs}}^{(n)}$  and guessed entropy  $H_{\text{guess}}^{(n)}$  for the text *Moby Dick* ( $N = 5 \times 10^4$ ) over the word length  $n$ . The circles display the results of an extrapolation method described in ref. 6 and the squares show the observed entropy of the text using  $N = 10^6$  words. (o) Overlapping, (n) nonoverlapping counting.

( $N = 5 \times 10^4$ ), and with the observed entropy based on a 20 times larger set of data (boxes: overlapping counting). As expected, for longer word length  $n$  the observed values  $H_{\text{obs}}^{(n)}$  underestimate the entropy. For small  $n$  they are reliable due to the reliable statistics. The guessed entropy  $H_{\text{guess}}^{(n)}$  agrees for small  $n$  with the observed entropy and for large  $n$  with the extrapolated values.

The form of the guessed theoretical distribution  $p(k, \alpha, \beta, \gamma, \dots)$  is arbitrary as long as it is a normalized monotonically decreasing function (Zipf order). Suppose that one has no information about the mechanism which generated a given sample. Then one has to find the functional form of the guessed distribution which is most appropriate to a given problem, i.e., in the ideal case the guessed distribution contains the real probability distribution as special case without being too complicated. An ansatz  $p(k, \alpha, \beta, \gamma, \dots)$  is suited if the optimized guessed probability distribution reproduces the frequency distribution of the original sample with satisfactory accuracy.

The ansatz (8) looks rather artificial: in fact we tried several forms of the guessed probability distribution and the one proposed in Eq. (8) turned



out to be the best of them. None of the others reproduces the frequencies sufficiently correctly. For demonstration we assume the function

$$p(k, \alpha, \beta) = \begin{cases} \alpha \exp(-\beta k), & k \leq \gamma \\ 0, & k > \gamma \end{cases} \quad (9)$$

with the normalization  $\gamma = -\beta^{-1} \log(1 - \beta/\alpha)$ . The optimized function is drawn in Figure 3 [*guessed* ( $b$ )]. We find that the frequency distribution reproduced from this function differs much more from the original frequency distribution (*Moby Dick*) than that of the guess according to Eq. (8).

Admittedly, any similar ansatz showing a well-pronounced peak for low ranks (*frequent words*), a long plateau with slow decrease (*standard words*), and a long tail (*seldom words*) could give reliable results as well. Anyhow there is no wide choice for the parametric form of the probability distribution. Equation (8) belongs to the class of distributions fulfilling this *three-region criterion*. For a more detailed discussion of the statistics of words see, e.g. ref. 11 and many references therein.

#### 4. DISCUSSION

The problem addressed in this paper was to find the rank ordered probability distribution from the given frequency distribution of a finite sample. For finite samples (Bernoulli sequence and English text) we have shown that the calculation of the entropy using the relative frequencies instead of the (unknown) probabilities yields wrong results.

We could show that the proposed algorithm is able to find the correct parameters of a guessed probability distribution which reproduces the statistical characteristics of a given symbolic sequence. The method has been tested for samples generated by well-defined sources, i.e., by known probability distributions, and for an unknown source, i.e., the word distribution of an English text. For the sample sequences we have evidence that the algorithm yields reliable results. The deviations of the entropy values from the correct values are rather small and in all cases far better than the observed entropies. For the unknown source *Moby Dick* we have no direct possibility to check the quality of the method; however, the calculated entropy values agree for small word lengths  $n$  with the observed entropy and for larger  $n$  with the results of an extrapolation method.<sup>(6)</sup> In this sense both approaches support each other. The proposed algorithm can be applied to the trajectories of dynamic systems. Formally the trajectory of a discrete dynamics is a text written in a certain language using  $\lambda$  different letters. The rank ordered distribution of subtrajectories of length

$n$  belongs to the most important characteristics of a discrete dynamic system. In this way we consider the analysis of an English text as an example for the analysis of a very complex dynamic system.

In many cases there is a principal limitation of the available data, i.e., the available samples are small with respect to the needs of a reliable statistics, and hence there is a principal limitation for the calculation of the statistical properties using frequencies instead of probabilities. For such cases, using the proposed method, one can calculate values which could not be found otherwise. Given a finite set of data, the proposed method yields the most reliable values for the Zipf-ordered distributions and the entropies which are presently available. The method is not restricted to the calculation of the entropy, but all statistical properties which depend on the Zipf-ordered probability distribution can be estimated using the proposed algorithm.

## ACKNOWLEDGMENTS

We thank T. Aßelmeyer, H. J. Herrmann, and L. Schimansky-Geier for discussion and the Project Gutenberg Etext, Illinois Benedictine College, Lisle, Illinois, for providing the ASCII-text of *Moby Dick*.

## REFERENCES

1. J. H. Justice, ed., *Maximum Entropy and Bayesian Methods in Applied Statistics* (Cambridge University Press, Cambridge, 1986).
2. H. Herzel, *Syst. Anal. Mod. Simul.* 5:435 (1988); P. Grassberger, *Inf. J. Theor. Phys.* 25:907 (1986); *Phys. Lett. A* 128:369 (1988); *IEEE Trans. Inf. Theory* 35:669 (1989).
3. A. Schmitt, H. Herzel, and W. Ebeling, *Europhys. Lett.* 23:303 (1993).
4. B. McMillan, *Ann. Math. Stat.* 24:196–216 (1953); A. Khinchin, *Mathematical Foundation of Information Theory* (Dover, New York, 1967).
5. Donald E. Knuth, *The Art of Computer Programming* (Addison-Wesley Reading, Massachusetts, 1973), Vol. 3, pp. 506–570; Robert Sedgwick, *Algorithms* (Addison-Wesley, Reading, Massachusetts, 1991).
6. W. Ebeling, T. Pöschel, and K. Albrecht, *Bifurcation & Chaos*, in press.
7. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1987).
8. H. Melville, *Moby Dick* [provided as ASCII-text by Project Gutenberg Etext, Illinois Benedictine College, Lisle, Illinois].
9. W. Ebeling and T. Pöschel, *Europhys. Lett.* 26:241 (1994).
10. W. Ebeling and T. Pöschel, in preparation.
11. A. Apostolico and Z. Galil, *Combinatorial Algorithms on Words* (Springer, Berlin, 1985).