Algorithms **2012**, *5*, 50-55; doi:10.3390/a5010050

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

A Note on Sequence Prediction over Large Alphabets

Travis Gagie

Department of Computer Science and Engineering, Aalto University, 00076 Aalto, Finland;
E-Mail: travis.gagie@aalto.fi

Received: 14 November 2011; in revised form: 11 February 2012 / Accepted: 13 February 2012 / Published: 17 February 2012

Abstract: Building on results from data compression, we prove nearly tight bounds on how well sequences of length n can be predicted in terms of the size σ of the alphabet and the length k of the context considered when making predictions. We compare the performance achievable by an adaptive predictor with no advance knowledge of the sequence, to the performance achievable by the optimal static predictor using a table listing the frequency of each $(k + 1)$-tuple in the sequence. We show that, if the elements of the sequence are chosen uniformly at random, then an adaptive predictor can compete in the expected case if $k \leq \log_\sigma n - 3 - \epsilon$, for a constant $\epsilon > 0$, but not if $k \geq \log_\sigma n$.

Keywords: sequence prediction; alphabet size; analysis

1. Introduction

The relation between compression and prediction dates back at least as far as William of Ockham in the fourteenth century. This relation was not properly formalized, however, until the notion of Kolmogorov complexity was developed in the twentieth century [1–3]. Since then, there have been many efforts to harness compression algorithms for prediction, with a number of researchers focusing particularly on prediction for prefetching either disk pages or web pages. We refer the reader to the text by Cesa–Bianchi and Lugosi [4] and references therein for a thorough discussion. For example, Krishnan and Vitter [5] showed that a prefetcher based on LZ78 [6] is asymptotically competitive with the best finite-state prefetcher. For prefetching, however, the alphabet of possible elements—*i.e.*, all pages on the disk or web—is huge. In this paper we investigate, therefore, what effect the size of the alphabet has on predictability.

Krishnan and Vitter considered the problem of pure prefetching, in which the prefetcher can replace all the contents of the cache between each page request. They combined a predictor by Hannan [7], which is asymptotically competitive against any memoryless predictor, with an instantaneous version of LZ78, thus obtaining a predictor such that, for any finite-state predictor and any sequence, their predictor's success rate converges to or exceeds the finite-state predictor's success rate with probability approaching 1 as the sequence length increases. (As Krishnan and Vitter noted, a similar construction for predicting binary sequences was given by Feder, Merhav and Gutman [8].) Notice that this bound cannot be improved to hold with certainty instead of with high probability: the predictor must be randomized because, for any deterministic predictor, an adversary can choose each bit of the sequence to be the opposite of what the predictor guesses. Krishnan and Vitter's proof is based on the fact that, for any finite-state compressor and any sequence, LZ78's asymptotic compression ratio is at most that of the finite-state compressor; this necessarily involves the assumption that both the alphabet and the context length used in prediction are fixed. It is known what effect the size of the alphabet has on compressibility, both when we make certain assumptions about the source of the sequence [9,10] and when we want to bound the size of the encoding in terms of the kth-order empirical entropy of the sequence [11,12]. We will define a notion of predictability that is analogous to empirical entropy and use similar arguments to give nearly tight bounds on how large the alphabet can be before we cannot guarantee good prediction.

The rest of this paper is laid out as follows: in Section 2 we briefly review several notions of entropy in computer science—Shannon entropy, empirical entropy, Rényi entropy and min-entropy—before defining empirical predictability; in Section 3 we show that we can achieve good expected prediction in terms of a sequence's kth-order empirical predictability when $k \leq \log_\sigma n - 3 - \epsilon$, where k is the length of the contexts considered when making predictions, σ is the size of the alphabet, n is the length of the sequence and $\epsilon > 0$ is a constant; in Section 4 we show we cannot achieve the same bound when $k \geq \log_\sigma n$. A preliminary version of these results [13] was reported at the 9th Canadian Workshop on Information Theory (CWIT '05) while the author was at the University of Toronto.

2. Empirical Predictability

Shannon [14] defined the entropy of a random variable to be our uncertainty about its value. Specifically, if a random variable $\mathcal{X}$ takes on one of σ values according to a probability distribution $P = p_1, \ldots, p_\sigma$, then

$$H(\mathcal{X}) = \sum_{j=1}^{\sigma} p_j \log \frac{1}{p_j}$$

The base of the logarithm determines the unit of uncertainty; in computer science, the base is usually assumed to be 2 with the result that the unit is the bit (*i.e.*, our uncertainty about the outcome of flipping a fair coin). Throughout the rest of this paper we write $\log$ to mean $\log_2$. Given P, the expected number of bits needed to encode the value of $\mathcal{X}$ is at least $H(\mathcal{X})$ and less than $H(\mathcal{X}) + 1$.

The 0th-order empirical entropy $H_0(S)$ of a sequence $S[1 \ldots n]$ is simply our uncertainty about an element chosen uniformly at random from S, *i.e.*,

$$H_0(S) = \frac{1}{n} \sum_{a \in S} \mathsf{occ}(a, S) \log \frac{n}{\mathsf{occ}(a, S)}$$

where $a \in S$ means that element a occurs in S and $\mathsf{occ}(a, S)$ is its frequency. For $k \geq 1$, the kth-order empirical entropy $H_k(S)$ of S is our expected uncertainty about the random variable $s[i]$ in the following experiment: i is chosen uniformly at random between 1 and n; if $i \leq k$, then we are told $s[i]$; if $i > k$, then we are told $s[i-k \ldots i-1]$ and asked to guess $s[i]$. Specifically,

$$H_k(S) = \frac{1}{n} \sum_{|\alpha|=k} |S_\alpha| H_0(S_\alpha)$$

where S_α is the concatenation of the elements in S immediately following occurrences of the k-tuple α. Notice that $|S_\alpha| = \mathsf{occ}(\alpha, S)$ unless α is a suffix of S, in which case it is 1 less. Given a table listing the frequency of each $(k+1)$-tuple in S, it takes about $nH_k(S)$ bits to encode S. For further discussion of empirical entropy, we refer readers to Manzini's analysis [15] of the Burrows–Wheeler Transform.

The Rényi entropy of order t of $\mathcal{X}$ is defined as

$$\frac{1}{1-t} \log \left(\sum_{j=1}^{\sigma} p_j^t \right)$$

for $0 \leq t \neq 1$, where the random variable $\mathcal{X}$ again takes on values according to the probability distribution $P = p_1, \ldots, p_\sigma$. The Rényi entropy of order 0 of $\mathcal{X}$ is the logarithm of the size of the support of P. The limit of the Rényi entropy of order t of $\mathcal{X}$ as t approaches 1 is the Shannon entropy $H(\mathcal{X})$ of $\mathcal{X}$; as t approaches infinity, the limit is $-\log \sup_{1 \leq j \leq \sigma} p_j$, which is often called the min-entropy of X. Min-entropy is related to predictability because, given P and asked to guess the value of X, our best strategy is to choose the most probable value and, thus, guess correctly with probability $\max_{1 \leq i \leq \sigma}\{p_i\}$.

If we are asked to guess the value of an element $s[i]$ chosen uniformly at random from the sequence S with no context given, then our best strategy is to choose the most frequent element and, thus, guess correctly with probability $\max_{a \in S} \frac{\mathsf{occ}(a,S)}{n}$. Following the example of empirical entropy, we call this probability the *0th-order empirical predictability* $P_0(S)$ of S. We define the *kth-order empirical predictability* $P_k(S)$ of S to be the expected predictability of the random variable $s[i]$ in the following experiment: i is chosen uniformly at random between 1 and n; if $i \leq k$, then we are told $s[i]$; if $i > k$, then we are told $s[i-k \ldots i-1]$ and asked to guess $s[i]$. Specifically,

$$P_k(S) = \frac{1}{n} \left(k + \sum_{|\alpha|=k} |S_\alpha| P_0(S_\alpha) \right)$$

for $k \geq 1$, where S_α is again the concatenation of the elements in S immediately following occurrences of the k-tuple α. For example, if $S = \mathsf{TORONTO}$ then $P_0(S) = 3/7 \approx 0.429$,

$$\begin{aligned} P_1(S) &= \frac{1}{7}\left(1 + P_0(S_\mathsf{N}) + 2P_0(S_\mathsf{O}) + P_0(S_\mathsf{R}) + 2P_0(S_\mathsf{T})\right) \\ &= \frac{1}{7}\left(1 + P_0(\mathsf{T}) + 2P_0(\mathsf{RN}) + P_0(\mathsf{O}) + 2P_0(\mathsf{OO})\right) \\ &= \frac{6}{7} \approx 0.857 \end{aligned}$$

and all higher-order empirical entropies are 1. In other words, if someone asks us to guess an element chosen uniformly at random from TORONTO then, given no context, we should choose O, in which case the probability of our prediction being correct is $3/7$. If we are given the preceding element (or told there is no preceding element) and it is not an O, then we can answer with certainty; if it is an O, which has probability $2/7$, then we should guess either R or N and be right with probability $1/2$; overall, the probability of our prediction being correct is $6/7$. If we are given the two preceding elements (or told how many preceding elements there are), then we can always answer with certainty.

Given a table listing the frequency of each $(k+1)$-tuple in S, we can build a static predictor that, after seeing k elements, always predicts the element that follows that k-tuple most often in S; this predictor guesses correctly $nP_k(S)$ times when predicting all the elements in S, which is optimal for a static predictor that uses contexts of length at most k.

3. Upper Bound

Having defined the kth-order predictability $P_k(S)$ of S, it is natural to ask when an adaptive predictor with no advance knowledge of S can achieve success rate $P_k(S)$. Whenever both k and the size σ of the alphabet are fixed, Krishnan and Vitter's predictor [5] almost certainly achieves a success rate asymptotically approaching $P_k(S)$ as n goes to infinity. If S is a randomly-chosen permutation, however, then $P_1(S) = 1$ but the expected success rate of any predictor without advance knowledge of S, approaches 0 as n increases. In this section we show that if $k \leq \log_\sigma n - 3 - \epsilon$, for a constant $\epsilon > 0$, then an adaptive predictor can achieve expected success rate $P_k(S)$ on any sufficiently long sequence. For simplicity we assume that k is given although, in practice, a predictor should find an optimal or nearly optimal context length by itself.

The most obvious 0th-order predictor is the one that always guess that the next element will be the most frequent element seen so far. Hannan [7] randomized this predictor and obtained a predictor A whose expected success rate converges to $P_0(S)$ when $n = \omega(\sigma^3)$. We now consider the most obvious generalization A_k of Hannan's predictor to use contexts of a given length k: after seeing a k-tuple α, we apply Hannan's predictor to the subsequence of elements consisting of the concatenation of elements so far that immediately followed occurrences of α; *i.e.*,

$$A_k(S[1\ldots i]) = A((S[1\ldots i])_\alpha)$$

where $\alpha = S[i-k\ldots i-1]$ and $(S[1\ldots i])_\alpha$ is as defined in Section 2.

Fix $\epsilon > 0$ and assume $k \leq \log_\sigma n - 3 - \epsilon$. Consider the subsequences into which A_k partitions S before applying A, and let $\mathcal{L}$ be the subset of them that are each of length at least $\sigma^{3+\epsilon/2}$. Notice that A achieves expected success rate $P_0(S')$ on any subsequence $S' \in \mathcal{L}$ so, by linearity of expectation, A_k achieves expected success rate at least

$$\frac{1}{n} \sum_{\substack{|\alpha|=k,\\ S_\alpha \in \mathcal{L}}} |S_\alpha| P_0(S_\alpha)$$

On the other hand, the total length of the subsequences not in $\mathcal{L}$ is less than $\sigma^k \cdot \sigma^{3+\epsilon/2} < \frac{n}{\sigma^{\epsilon/2}} = o(n)$, so

$$\begin{aligned} P_k(S) &= \frac{1}{n}\left(k + \sum_{\substack{|\alpha|=k,\\ S_\alpha \in \mathcal{L}}} |S_\alpha| P_0(S_\alpha) + \sum_{\substack{|\alpha|=k,\\ S_\alpha \notin \mathcal{L}}} |S_\alpha| P_0(S_\alpha)\right) \\ &= \frac{1}{n} \sum_{\substack{|\alpha|=k,\\ S_\alpha \in \mathcal{L}}} |S_\alpha| P_0(S_\alpha) + o(1) \end{aligned}$$

Therefore, when S is sufficiently long, A_k achieves expected success rate $P_k(S)$.

Theorem 1 *If the n elements of a sufficiently long sequence S are chosen arbitrarily from an alphabet of size σ and $k < \log_\sigma n - 3 - \epsilon$, for a constant $\epsilon > 0$, then A_k achieves expected success rate $P_k(S)$.*

4. Lower Bound

Compression researchers (see, e.g., [16] for a survey) have shown how to store S in $nH_k(S) + o(n \log \sigma)$ for all $k \leq (1 - \epsilon) \log_\sigma n$ simultaneously, where ϵ is a positive constant. In a previous paper [11] we showed that it is impossible to prove a worst-case bound of this form when $k \geq \log_\sigma n$:

- in σ-ary De Bruijn cycles [17] of order k, each k-tuple appears exactly once, so such cycles have length σ^k and kth-order empirical entropy 0;
- there are $(\sigma!)^{\sigma^{k-1}}/\sigma^k$ such cycles [18] and $\log_2\left((\sigma!)^{\sigma^{k-1}}/\sigma^k\right) = \Theta(\sigma^k \log \sigma)$;
- by the pigeonhole principle, there is no injective mapping from σ-ary strings of length n with kth-order empirical entropy 0, to binary strings of length $o(n \log \sigma)$;
- therefore, if $k \geq \log_\sigma n$, then in the worst case we cannot store S in $\lambda n H_k(S) + o(n \log \sigma)$ bits for any coefficient λ.

In a recent paper [12] we used similar but more sophisticated arguments to show that, if $k \geq (1 + \epsilon) \log_\sigma n$ for some positive constant ϵ, then in the expected case we cannot store S in $\lambda n H_k(S) + o(n \log \sigma)$ bits for any coefficient $\lambda = o(n^\epsilon)$; if $k \geq (2 + \epsilon) \log_\sigma n$, then with high probability we cannot store S in that many bits for any coefficient λ.

We now turn our attention to proving lower bounds for prediction and show that, if $k \geq \log_\sigma n$ and the elements of S are chosen uniformly at random, then no predictor without advance knowledge of S can achieve an expected success rate close to $P_k(S)$. Notice that $nP_k(S)$ is, by definition, at least the number of distinct k-tuples in S minus 1: for any distinct k-tuple α that occurs in S and is not a suffix of S, the optimal static predictor described in Section 2 correctly guesses the element after at least one occurrence of α in S. Suppose $k \geq \log_\sigma n$—implying that $\sigma \geq 2$—and let $c = \sigma^k/n \geq 1$. Janson, Lonardi and Szpankowski [19] showed that the expected number of distinct k-tuples in S is

$$\sigma^k \left(1 - \frac{1}{e^{n/\sigma^k}}\right) + \mathcal{O}(k) + \mathcal{O}\left(\frac{nk}{\sigma^k}\right) = cn \left(1 - \frac{1}{e^{1/c}}\right) + \mathcal{O}(k) + \mathcal{O}(k/c)$$

so

$$\mathrm{E}[P_k(S)] \geq c \left(1 - \frac{1}{e^{1/c}}\right) \geq 1 - \frac{1}{e} > 0.632$$

On the other hand, no predictor without advance knowledge of S can achieve an expected success rate greater than $1/\sigma \leq 1/2$.

Theorem 2 *If the n elements of a sequence S are chosen uniformly at random from an alphabet of size σ and $k \geq \log_\sigma n$, then S's expected kth-order empirical predictability $\mathrm{E}[P_k(S)] \geq 1 - 1/e > 0.632$ but no predictor without advance knowledge of S can achieve an expected success rate greater than $1/\sigma \leq 1/2$.*

References

1. Solomonoff, R.J. A formal theory of inductive inference. *Inf. Control* **1964**, *7*, 1–22, 224–254.
2. Kolmogorov, A.N. Three approaches to the quantitative definition of information. *Probl. Inf. Transm.* **1965**, *1*, 1–7.
3. Li, M.; Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed.; Springer: Berlin, Germany, 2008.
4. Cesa-Bianchi, N.; Lugosi, G. *Prediction, Learning, and Games*; Cambridge University Press: Cambridge, UK, 2006.
5. Krishnan, P.; Vitter, J.S. Optimal prediction for prefetching in the worst case. *SIAM J. Comput.* **1998**, *27*, 1617–1636.
6. Ziv, J.; Lempel, A. Compression of individual sequences via variable-length coding. *IEEE Trans. Inf. Theory* **1978**, *24*, 530–536.
7. Hannan, J. Approximation of Bayes Risk in Repeated Plays. In *Contributions to the Theory of Games*; Dresher, M., Tucker, A., Wolfe, P., Eds.; Princeton University Press: Princeton, NJ, USA, 1957; Volume 3, pp. 97–139.
8. Feder, M.; Merhav, N.; Gutman, M. Universal prediction of individual sequences. *IEEE Trans. Inf. Theory* **1992**, *38*, 1258–1270.
9. Rissanen, J. Modeling by shortest data description. *Automatica* **1978**, *14*, 465–471.
10. Rissanen, J. Complexity of strings in the class of Markov sources. *IEEE Trans. Inf. Theory* **1986**, *32*, 526–532.
11. Gagie, T. Large alphabets and incompressibility. *Inf. Process. Lett.* **2006**, *99*, 246–251.
12. Gagie, T. Bounds from a card trick. *J. Discret. Algorithm.* **2012**, *10*, 2–4.
13. Gagie, T. A Note on Sequence Prediction. In *Proceedings of the 9th Canadian Workshop on Information Theory*, Montreal, Canada, 5–8 June 2005; pp. 304–306.
14. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423, 623–656.
15. Manzini, G. An analysis of the Burrows-Wheeler transform. *J. ACM* **2001**, *48*, 407–430.
16. Navarro, G.; Mäkinen, V. Compressed full-text indexes. *ACM Comput. Surv.* **2007**, *39*.
17. de Bruijn, N.G. A combinatorial problem. *K. Ned. Akad. Wet.* **1946**, *49*, 758–764.
18. van Aardenne-Ehrenfest, T.; de Bruijn, N.G. Circuits and trees in oriented linear graphs. *Simon Stevin* **1951**, *28*, 203–217.
19. Janson, S.; Lonardi, S.; Szpankowski, W. On average sequence complexity. *Theor. Comput. Sci.* **2004**, *326*, 213–227.

© 2012 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/.)