BRAIN. Broad Research in Artificial Intelligence and Neuroscience,
ISSN 2067-3957, Volume 1, July 2010,
Special Issue on Complexity in Sciences and Artificial Intelligence,
Eds. B. Iantovics, D. Rădoiu, M. Mǎruşteri and M. Dehmer

SOME CONSEQUENCES OF THE COMPLEXITY OF INTELLIGENT PREDICTION

Joel Ratsaby

Abstract. What is the relationship between the complexity of a learner and the randomness of his mistakes ? This question was posed in [4] who showed that the more complex the learner the higher the possibility that his mistakes deviate from a true random sequence. In the current paper we report on an empirical investigation of this problem. We investigate two characteristics of randomness, the stochastic and algorithmic complexity of the binary sequence of mistakes. A learner with a Markov model of order k is trained on a finite binary sequence produced by a Markov source of order k^* and is tested on a different random sequence. As a measure of learner's complexity we define a quantity called the *sysRatio*, denoted by ρ, which is the ratio between the compressed and uncompressed lengths of the binary string whose i^{th} bit represents the maximum *a posteriori* decision made at state i of the learner's model. The quantity ρ is a measure of information density. The main result of the paper shows that this ratio is crucial in answering the above posed question. The result indicates that there is a critical threshold ρ^* such that when $\rho \leq \rho^*$ the sequence of mistakes possesses the following features: (1) low divergence Δ from a random sequence, (2) low variance in algorithmic complexity. When $\rho > \rho^*$, the characteristics of the mistake sequence changes sharply towards a high Δ and high variance in algorithmic complexity. It is also shown that the quantity ρ is inversely proportional to k and the value of ρ^* corresponds to the value k^*. This is the point where the learner's model becomes too simple and is unable to approximate the Bayes optimal decision. Here the characteristics of the mistake sequence change sharply.

Keywords: *learning, sequence prediction, descriptive complexity*

Mathematics Subject Classification: 68Q19, 68Q30, 62M05, 68Q87, 94A17.

1. Overview

In computer science, the notion of computational complexity serves as a measure of how difficult it is to compute a solution for a given problem. Computations take time and complexity here means the time rate of growth to solve the problem. Another related kind of complexity measure (studied in theoretical computer science) is the so-called algorithmic (or Kolmogorov) complexity which measures how long a computer program (on some generic computational machine) needs to be in order that it produces a complete description of an object. Interestingly, the theory says that if we consider as an object a system that can process input information (available as a binary sequence of high entropy) and which produces another sequence as an output then the amount of randomness in the output sequence is inversely proportional to the algorithmic complexity of the system.

This has been known in the context of algorithmic randomness (see [1] and references within) and it has been only until recently unknown whether such a relationship between complexity and randomness exists for more general systems, for instance, those governed by physical laws. In [3] the complexity of a general static system (for instance, a physical solid) is modeled algorithmically, i.e., by its description length. Using the model it is proposed that the stability of a static system (from the physical perspective) is related to its level of algorithmic complexity. This is explained by the relationship between the complexity of a system and its ability to 'distort' the randomness in its environment. A proof of this concept appeared in several recent works [2, 6, 7] where it is shown that this inverse relationship between system complexity and randomness exists in a physical system. The particular system investigated consisted of a one-dimensional vibrating solid-beam to which a random sequence of external input forces is applied.

The current paper is yet another proof of concept of the model of [3]. We consider a *decision* system and study its influence on a random binary data sequence on which prediction decisions are made. The decision system is based on the maximum *a posteriori* probability decision where probabilities are defined by a statistical parametric model which is estimated from data. The learner of this model is a computer program that trains from a given random data sequence and then produces a decision rule by which it is able to predict (or decide) the value of the next bit in future (yet unseen) random binary sequences.

Our interest is in displaying a learning (and decision) system from the perspective of static system complexity (as in [3]) and determine its influence on a random input sequence.

2. Introduction

Let $X^{(n)} = X_1, \dots, X_n$ be a sequence of binary random variables drawn according to some unknown joint probability distribution $\mathbb{P}\left(X^{(n)}\right)$. Consider the problem of learning to predict the next bit in a binary sequence drawn according to $\mathbb{P}$. For training, the learner is given a finite sequence $x^{(m)}$ of bits $x_t \in \{0,1\}$, $1 \leq t \leq m$, drawn according to $\mathbb{P}$ and estimates a model $\mathcal{M}$ that can be used to predict the next bit of a partially observed sequence. After training, the learner is tested on another sequence $x^{(n)}$ drawn according to the same unknown distribution $\mathbb{P}$. Using $\mathcal{M}$ he produces the bit y_t as a prediction for x_t , $1 \leq t \leq n$. Denote by $\xi^{(n)}$ the corresponding binary sequence of mistakes where $\xi_t = 1$ if $y_t \neq x_t$ and is 0 otherwise. In [4] the following question was posed: how random is $\xi^{(n)}$? This question was answered for a particular learning setting where the teacher uses a probability distribution $\mathbb{P}$ based on a Markov model with a certain complexity. The learner has access to a hypothesis class of Boolean decision rules that are based on Markov models. Learning amounts to the estimation of parameters of a finite-order Markov model. The answer shows theoretically that the random characteristics of the subsequence of mistakes corresponding to the 0-predictions of a learner changes in accordance with the complexity of the learner's decision rule's complexity. The more complex the rule the higher the possibility of 'distortion' of randomness, i.e., the farther away it is from being truly-random.

In the current paper we take an experimental approach to answering the above question. As in [4] we focus on Markov source and a Markov learner whose orders may differ.

As this is only a short version of the paper in [5] in the next sections we briefly describe the setup and summarize the results.

3. Experimental setup

The learning problem consists of predicting the next bit in a given sequence generated by a Markov chain (model) $\mathcal{M}^*$ of order k^*. There are 2^{k^*} states in the model each represented by a word of k^* bits. During a learning problem, the source's model is fixed. A learner, unaware of the source's model, has a

Markov model of order k. We denote by $p(1|i)$ the probability of transiting from state i whose binary k-word is $b_i = [b_i(1), \ldots, b_i(k)]$ to the state whose word is $[b_i(2), \ldots, b_i(k), 1]$. Given a random sequence of length m generated by the source the learner estimates its own model's parameters $p(1|i)$ by $\hat{p}(1|i)$, $1 \leq i \leq 2^k$, which is the frequency of the event "b_i is followed by a 1" in the training sequence. We denote by $\hat{\mathcal{M}}$ the learnt model with parameters $\hat{p}(1|i)$, $1 \leq i \leq 2^k$. We denote by $p^*(1|i)$ the transition probability from state i of the source model, $1 \leq i \leq 2^k$.

A simulation run is characterized by the parameters, k and m. It consists of a training and testing phases. In the training phase we show the learner a binary sequence of length m and he estimates the transition probabilities. In the testing phase we show the learner another random sequence (generated by the same source) of length n and test the learner's predictions on it. For each bit in the test sequence we record whether the learner has made a mistake. When a mistake occurs we indicate this by a 1 and when there is no mistake we write a 0. The resulting sequence of length n is the generalization mistake sequence $\xi^{(n)}$. We denote by $\xi_0^{(n)}$ the binary subsequence of $\xi^{(n)}$ that corresponds to the mistakes that occured only when the learner predicted a 0.

For a fixed k denote by $N_{k,m}$ the number of runs with a learner of order k and training sample of size m. The experimental setup consists of $N_{k,m} = 10$ runs with $1 \leq k \leq 10$, $m \in \{100, 200, \ldots, 10000\}$ with a total of $100 \cdot 10 \cdot N_{k,m} = 10000$ runs. The testing sequence is of length $n = 1000$. Each run results in a file called *system* which contains a binary vector d whose i^{th} bit represents the maximum *a posteriori* decision made at state i of the learner's model, i.e.,

$$d_i = \begin{cases} 1 & \text{if } \hat{p}(1|i) > 1/2 \\ 0 & otherwise \end{cases} \tag{1}$$

for $1 \leq i \leq 2^k$.

Another file generated is the *errorT0* which contains the mistake subsequence $\xi_0^{(n)}$. At the end of each run we measure the lengths of the *system* file and its compressed length where compression is obtained via the Gzip algorithm (a variant of [8]) and compute the *sysRatio* (denoted as ρ) which is the ratio of the compressed to uncompressed length of the system file. Note that ρ is a measure of information density since it captures the number of bits of useful information (useful for describing the system) per bit of representation (in the uncompressed file).

We do similarly for the mistake-subsequence $\xi_0^{(n)}$ obtaining the length ℓ_0 of the compressed file that contains $\xi_0^{(n)}$ (henceforth referred to as the estimated algorithmic complexity of $\xi_0^{(n)}$ since it is an approximation of the Kolmogorov complexity of $\xi_0^{(n)}$, see [7]. We measure the KL-divergence Δ_0 between the probability distribution $P(w|\hat{p})$ of binary words w of length 4 and the empirical probability distribution $\hat{P}_m(w)$ as measured from the mistake subsequence $\xi_0^{(n)}$. Note, $P(w|\hat{p})$ is defined according to the Bernouli model with parameter $\hat{p}$, that is, $P(w|\hat{p}) = \hat{p}^i(1-\hat{p})^{4-i}$ for a word w with i ones, where $\hat{p}$ is the frequency of ones in the subsequence $\xi_0^{(n)}$. The distribution $\hat{P}_m(w)$ equals the frequency of a word w in $\xi_0^{(n)}$. Hence Δ_0 reflects by how much $\xi_0^{(n)}$ deviates from being random according to a Bernoulli sequence.

4. Results and conclusions

The first result of the experiment indicate that the mean of the sysRatio ρ decreases as the learner's model order k increases. As we discuss in the full paper [5] this is attributed to the lowering in entropy of the decision rule once the learner k surpasses k^*. We studied the characteristics of the mistake subsequence $\xi_0^{(n)}$. We observed that the mean of the estimated algorithmic complexity ℓ_0 of $\xi_0^{(n)}$ has a low spread in values when the mean of the system ratio ρ is low. There appears to be a sharp threshold at ρ^* where the spread around the mean value of ℓ_0 increases significantly. We saw a similar effect on the mean of the divergence Δ_0 of the mistake subsequence $\xi_0^{(n)}$. For low values of sysRatio the spread of Δ_0 is low and there exists a threshold at ρ^* where the standard deviation around the mean value of Δ_0 increases significantly. We conclude that the sysRatio ρ is a proper measure of complexity of a learner decision rule. It is with respect to ρ that the characteristics of the random mistake subsequence $\xi_0^{(n)}$ follow what the theory [4] predicts. The higher the sysRatio the more significant the deviation Δ_0 of $\xi_0^{(n)}$ compared to a pure Bernouli random sequence and the larger the possible fluctuations in its algorithmic complexity ℓ_0.

References

[1] L. Bienvenu. Kolmogorov-Loveland stochasticity and Kolmogorov complexity. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2007)*, volume LNCS 4393, pages 260–271, 2007.

[2] J. Chaskalovic and J. Ratsaby. Interaction of a self vibrating beam with chaotic external forces. *Comptes Rendus Mecanique*, doi:10.1016/j.crme.2009.11.001, 2009.

[3] J. Ratsaby. An algorithmic complexity interpretation of Lin's third law of information theory. *Entropy*, 10(1):6–14, 2008.

[4] J. Ratsaby. How random are a learner's mistakes ? *Technical Report # arXiv:0903.3667*, 2009.

[5] J. Ratsaby. Learning, complexity and information density. *Technical Report # arXiv:0908.4494v1*, 2009.

[6] J. Ratsaby and I. Chaskalovic. Random patterns and complexity in static structures. In *Proc. of International Conference on Artificial Intelligence and Pattern Recognition (AIPR'09)*, volume III of *Mathematics and Computer Science*, pages 255–261, 2009.

[7] J. Ratsaby and I. Chaskalovic. On the algorithmic complexity of static structures. *Journal of Systems Science and Complexity*, (doi:10.1007/s11424-010-8465-2), 2010.

[8] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Joel Ratsaby
Department of Electrical and Electronics Engineering
Ariel University Center
Ariel 40700, ISRAEL
email:*ratsaby@ariel.ac.il*