Signal-to-Computation Ratio: A Measure of System Complexity and Information Extraction

J.L

September 3, 2024

Abstract

In this paper, we introduce the concept of the Signal-to-Computation Ratio (SCR), a new metric for evaluating the computational efficiency required to extract information from a system. SCR is an intrinsic property of the system and is independent of the algorithm used. We define SCR mathematically and extend the concept to time-evolving dynamical systems.

1 Introduction

In many computational tasks, the efficiency of extracting information from a system is of paramount importance. While the signal-to-noise ratio (SNR) is a well-known metric in communication theory, we propose an analogous concept, the Signal-to-Computation Ratio (SCR), to measure the computational cost relative to the amount of information extracted from a system, if at-all possible. The SCR is a system-specific property, reflecting how "difficult" it is to extract meaningful information from that system, irrespective of the specific algorithms used.

The goal of this paper is to formalize SCR, explore its properties, and examine its applications. Furthermore, we investigate it in the context of time-evolving dynamical systems.

2 Definitions

Definition 2.1. Let $X = \{x_t\}_{t \in T}$ be a family of systems indexed by T. The index set T can be finite, infinite, countable, or uncountable. Each element x_t represents a system at a particular index t.

Definition 2.2. A Boolean question is a function $q : X \to \{0, 1\}$. Given a x_t system, the question q returns a binary response. Let Q be the set of such questions chosen to be "meaningful" and "solvable".

Proposition 2.1.

$$q = q' \iff \forall x_t \in X, q(x_t) = q'(x_t) \tag{2.1}$$

Definition 2.3. An algorithm is a function $a : X \times D \to \{0, 1\}$, where D represents the available data the algorithm can use.

Definition 2.4. An algorithm a is said to implement a Boolean question q if, $\forall x_t \in X, a(x_t, d_{x_t}) = q(x_t)$, where d_{x_t} is the available data of x_t. Two algorithms are considered equivalent if they implement the same Boolean question q.

$$a \sim a' \iff \exists q \in Q, \forall x_t \in X, a(x_t, d_{x_t}) = q(x_t) = a'(x_t, d_{x_t}) \tag{2.2}$$

In such cases, we denote the equivalence class of algorithm that implements q as A_q. $A_q \neq \emptyset$, because we only consider "solvable" questions $q \in Q$.

Definition 2.5. The computation resource cost of an algorithm

$$s : A \times X \times D \to \mathbb{R}^+ \tag{2.3}$$

measures the computation resource cost associated with executing the algorithm.

Definition 2.6. The computation resource cost of a question q about system x_t,

$$c : Q \times X \to \mathbb{R}^+ \tag{2.4}$$

is defined as the minimum consumption over all algorithms A_q that implement the question q. Formally,

$$c(q, x_t) := \min_{a \in A_q} s(a, x_t, d_{x_t}) \tag{2.5}$$

Definition 2.7.

$$e : \mathbb{N} \to \mathbb{R}^+ \tag{2.6}$$

The function $e_x(n)$ represents the least computation resource cost to extract n bits of information from the system x_t. It is defined as:

$$e_{x_t}(n) := \min_{\{\mathfrak{Q} \in 2^Q : |\mathfrak{Q}| = n\}} \sum_{q \in \mathfrak{Q}} c(q, x_t) \tag{2.7}$$

The min is taken over all combination of n questions chosen from Q.

3   Boolean Questions and Algorithms

Let $X = \{x_t\}_{t \in T}$ represent a family of systems indexed by T, where T could be finite, infinite, countable, or uncountable. We define a Boolean question $q : X \to \{0, 1\}$, which represents a query about the system that yields a binary response.

An algorithm $a : X \times D \to \{0, 1\}$ is a process that uses available data D to answer the question q. We say that an algorithm a implements a question q if for all t, $a(x_t, d_t) = q(x_t)$. In such cases, we denote the algorithm as a_q. Two algorithms are considered equivalent if they implement the same question.

4   computation resource cost

We define the computation resource cost $s(a)$ as the resource cost associated with an algorithm a. Similarly, we define $c(q)$ as the least computational consumption needed to answer a question $q \in Q$, optimized over all possible algorithms that implement q:

$$c(q) = \min_{a_q} s(a_q)$$

Thus, $c(q)$ is a lower bound on the computational effort required to extract the information associated with question q.

5   Least Computation to Extract Information

The function $e_x(n)$ represents the least computation needed to extract n bits of information from the system x. Formally, it is defined as:

$$e_x(n) := \inf_{\{x \in 2^Q : |x| = n\}} \sum_{q \in x} c(q)$$

Here, Q is the set of all Boolean questions, and the infimum is taken over all subsets of Q that extract exactly n bits of information.

6   Time-Evolving Dynamical Systems

In many real-world applications, the systems of interest are time-evolving. Let x_t represent the state of a dynamical system at time t. In this context, information extraction often involves considering the entire trajectory of the system over a time interval, rather than a single snapshot.

We adapt the definition of SCR for such systems by considering the computational effort required to extract n bits of information from the system's

trajectory:
$$\text{SCR}(x_t) = \frac{n}{e_{x_t}(n)}$$
where $e_{x_t}(n)$ denotes the minimal computation required to extract n bits of information from the system's evolution over time.

7   Impact of Trivial Questions

Consider the scenario where the set of Boolean questions Q is flooded with trivial questions—questions that always return the same value (e.g., always 0 or always 1), regardless of the input system. These questions have minimal computation resource cost, as their answers do not depend on the system's state.

The presence of a large number of trivial questions can skew the calculation of the minimal computation $e_x(n)$. Specifically, it can artificially lower the perceived computational cost, leading to an inflated SCR. To address this issue, we need to carefully distinguish between trivial and non-trivial questions when calculating $e_x(n)$ to ensure that the SCR reflects meaningful information extraction rather than simple, uninformative queries.

8   Conclusion and Future Work

In this paper, we introduced the concept of Signal-to-Computation Ratio (SCR) as a measure of the computational efficiency required to extract information from a system. We defined SCR mathematically and explored its properties, especially in the context of time-evolving dynamical systems. Additionally, we discussed the impact of trivial questions on the computation resource cost and proposed methods to address these issues.

Future work in this area could involve refining the definition of SCR for more complex systems, including those with stochastic behavior, and exploring practical applications in fields such as machine learning, cryptography, and information theory. Moreover, further investigation into the optimization of algorithms for maximizing SCR could lead to new insights into efficient information extraction methods.

9   References

References

[1] Shannon, C. E., *A Mathematical Theory of Communication*, Bell System Technical Journal, vol. 27, pp. 379–423, 1948.

[2] Cover, T. M., and Thomas, J. A., *Elements of Information Theory*, Wiley-Interscience, 2006.

[3] Turing, A. M., *On Computable Numbers, with an Application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society, vol. 2, issue 42, pp. 230-265, 1936.