I-Complexity and Discrete Derivative of Logarithms: A Group-Theoretic Explanation

Vladik Kreinovich and Jaime Nava

Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
Emails: vladik@utep.edu, jenava@miners.utep.edu

1. Kolmogorov Complexity

- The best way to describe the complexity of a given string s is to find its *Kolmogorov complexity* $K(s)$.
- $K(s)$ is the shortest length of a program that computes s.
- For example, a sequence is random if and only if its Kolmogorov complexity is close to its length.
- We can check how close are two DNA sequences s and s' by comparing $K(ss')$ with $K(s)+K(s')$:
  - if they are *unrelated*, the only way to generate ss' is to generate s and then generate s', so

$$K(ss') \approx K(s)+K(s');$$

  - if they are *related*, we have $K(ss') \ll K(s)+K(s')$.

2. Need for Approximate Complexity

- The big problem is that the Kolmogorov complexity is, in general, *not* algorithmically *computable*.
- Thus, it is desirable to come up with *computable* approximations.
- At present, most algorithms for approximating $K(s)$:
  - use some loss-less compression technique to compress s, and
  - take the length $\widetilde{K}(s)$ of the compression as the desired approximation.
- However, this approximation has limitations: for example,
  - in contrast to $K(s)$, where a change (one-bit) change in x cannot change $K(s)$ much,
  - a small change in s can lead to a drastic change in $\widetilde{K}(s)$.

3. I-Complexity

- Limitation of $\widetilde{K}(s)$: a small change in $s = (s_1 s_2 \ldots s_n)$ can lead to a drastic change in $\widetilde{K}(s)$.
- To overcome this limitation, V. Becher and P. A. Heiber proposed the following new notion of *I-complexity*.
- For each position i, we find the length $B_s[i]$ of the largest repeated substring within $s_1 \ldots s_i$.
- For example, for *aaaab*, the corresponding values of $B_s(i)$ are 01233.
- We then define $I(s) \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} f(B_s[i])$, for an appropriate decreasing function $f(x)$.
- Specifically, it turned out that the *discrete derivative of the logarithm* works well: $f(x) = \text{dlog}(x+1)$, where

$$\text{dlog}(x) \stackrel{\text{def}}{=} \log(x+1) - \log(x).$$

4. Good Properties of I-Complexity

- *Reminder:* $I(s)=\sum_{i=1}^{n} f(B_s[i])$, where:
  - $B_s[i]$ is the length of the largest repeated substring within $s_1 \ldots s_i$, and
  - $f(x)=\log(x+1)-\log(x)$.
- *Similarly to* $K(s)$:
  - If s starts s', then $I(s) \le I(s')$.
  - We have $I(0s) \approx I(s)$ and $I(1s) \approx I(s)$.
  - We have $I(ss') \le I(s)+I(s')$.
  - Most strings have high I-complexity.
- *In contrast to* $K(s)$*:* I-complexity can be computed in linear time.
- *A natural question*: why this function $f(x)$?

5. Towards Precise Formulation of the Problem

- We view the desired function $f(x)$ as a discrete analogue of an appropriate continuous function $F(x)$:

$$f(x) = \int_x^{x+1} g(y)\,dy = F(x+1) - F(x).$$

- Which function $F(x)$ should we choose?
- In the continuous case, the numerical value of each quantity depends:
  - on the choice of the measuring unit and
  - on the choice of the starting point.
- By changing them, we get a new value $x' = a \cdot x + b$.
- For length x, the starting point 0 is fixed.
- So, we only have re-scaling $x \to x' = a \cdot x$.

6. Our Result

- By changing a measuring unit, we get $x' = a \cdot x$.
- When we thus re-scale x, the value $y = F(x)$ changes, to $y' = F(a \cdot x)$.
- It is reasonable to require that the value y' represent the same quantity.
- So, we require that y' differs from y by a similar re-scaling:

  $y' = F(a \cdot x) = A(a) \cdot F(x) + B(a)$ for some $A(a)$ and $B(a)$.

- It turns out that all monotonic solutions of this equation are linearly equivalent to $\log(x)$ or to x^α, i.e.:

$$F(x) = \widetilde{a} \cdot \ln(x) + \widetilde{b} \text{ or } F(x) = \widetilde{a} \cdot x^\alpha + \widetilde{b}.$$

- So, symmetries do explain the selection of the function $F(x)$ for I-complexity.

7. Proof

- *Reminder:* for some monotonic function $F(x)$, for every a, there exist values $A(a)$ and $B(a)$ for which

$$F(a \cdot x) = A(a) \cdot F(x) + B(a).$$

- *Known fact:* every monotonic function is almost everywhere differentiable.
- Let $x_0 > 0$ be a point where the function $F(x)$ is differentiable.
- Then, for every x, by taking $a = x/x_0$, we conclude that $F(x)$ is differentiable at this point x as well.
- For any $x_1 \neq x_2$, we have $F(a \cdot x_1) = A(a) \cdot F(x_1) + B(a)$ and $F(a \cdot x_2) = A(a) \cdot F(x_2) + B(a)$.
- We get a system of two linear equations with two unknowns $A(a)$ and $B(a)$.

8. Proof (cont-d)

- We get a system of two linear equations with two unknowns $A(a)$ and $B(a)$:

$$F(a\cdot x_1)=A(a)\cdot F(x_1)+B(a).$$
$$F(a\cdot x_2)=A(a)\cdot F(x_2)+B(a).$$

- Thus, both $A(a)$ and $B(a)$ are linear combinations of differentiable functions $F(a\cdot x_1)$ and $F(a\cdot x_2)$.
- Hence, both functions $A(a)$ and $B(a)$ are differentiable.
- So, $F(a\cdot x)=A(a)\cdot F(x)+B(a)$ for differentiable functions $F(x)$, $A(a)$, and $B(a)$.
- Differentiating both sides by a, we get

$$x\cdot F'(a\cdot x)=A'(a)\cdot F(x)+B'(a).$$

- In particular, for $a=1$, we get $x\cdot\dfrac{dF}{dx}=A\cdot F+B$, where $A\stackrel{\text{def}}{=}A'(1)$ and $B\stackrel{\text{def}}{=}B'(1)$.

9. Proof (final part)

- *Reminder:* $x \cdot \dfrac{dF}{dx} = A \cdot F + B.$
- So, $\dfrac{dF}{A \cdot F + b} = \dfrac{dx}{x}$; now, we can integrate both sides.
- *When $A = 0$:* we get $\dfrac{F(x)}{b} = \ln(x) + C$, so
$$F(x) = b \cdot \ln(x) + b \cdot C.$$
- *When $A \neq 0$:* for $\widetilde{F} \stackrel{\text{def}}{=} F + \dfrac{b}{A}$, we get $\dfrac{d\widetilde{F}}{A \cdot \widetilde{F}} = \dfrac{dx}{x}$, so $\dfrac{1}{A} \cdot \ln(\widetilde{F}(x)) = \ln(x) + C$, and $\ln(\widetilde{F}(x)) = A \cdot \ln(x) + A \cdot C$.
- Thus, $\widetilde{F}(x) = C_1 \cdot x^A$, where $C_1 \stackrel{\text{def}}{=} \exp(A \cdot C)$.
- Hence, $F(x) = \widetilde{F}(x) - \dfrac{b}{A} = C_1 \cdot x^A - \dfrac{b}{A}$.
- The theorem is proven.

10. Acknowledgments

This work was supported in part:

- by the National Science Foundation grants HRD-0734825 and DUE-0926721, and
- by Grant 1 T36 GM078000-01 from the National Institutes of Health.

11. References

- V. Becher and P. A. Heiber, "A better complexity of finite sequences", *Abstracts of the 8th Int'l Conf. on Computability and Complexity in Analysis CCA'2011 and 6th Int'l Conf. on Computability, Complexity, and Randomness CCR'2011*, Cape Town, South Africa, January 31 – February 4, 2011, p. 7.
- M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, Berlin, Heidelberg, New York, 2008.