Unlocking Recursion: On Out-of-Distribution Generalization with a Simple but Powerful Principle

by

Lukas J. Rüttgers

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

at the

RWTH AACHEN UNIVERSITY

September 2024

Authored by: Lukas J. Rüttgers
Department of Computer Science
June 30, 2024

Certified by: Jingzhao Zhang
Assistant Professor of Computer Science, Tsinghua University, Thesis Supervisor

Certified by: Hector Geffner
Professor of Computer Science, RWTH Aachen University, Thesis Supervisor

Unlocking Recursion: On Out-of-Distribution Generalization with a Simple but Powerful Principle

by

Lukas J. Rüttgers

Submitted to the Department of Computer Science
on June 30, 2024 in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

ABSTRACT

Out-of-distribution generalization requires agents to induce from their experience to unseen environments. While recursion is a powerful tool for compressing algorithmic behaviour, the hardness of learning recursive descriptions has impeded its integration into modern machine learning pipelines.

This work puts forward a new complexity measure for algorithmic descriptions that captures the informational "simplicity" of a program and is based on Kolmogorov complexity. Intuitively, this measure characterizes the complexity of boolean functions by the minimum functional information required to ensure that the simplest algorithm that is consistent with the given information indeed computes the desired function.

With this measure at hand, it emphasizes the efficacy of learning recursive algorithmic descriptions for reasonable behaviour outside of the training distribution, and showcases drawbacks in existing frequently adopted models and optimization objectives in capturing elementary recursive patterns.

Thesis supervisor: Jingzhao Zhang
Title: Assistant Professor of Computer Science, Tsinghua University

Thesis supervisor: Hector Geffner
Title: Professor of Computer Science, RWTH Aachen University

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The i.i.d. Assumption

1.2 Domain Generalization

1.3 Outline of This Work

1.3.1 Purpose and Contributions

1.3.2 Content Organization

Chapter 2

Domain Generalization of Neural Networks

2.1 Extrapolation Drawbacks of Models

2.1.1 ReLU MLPs Extrapolate Linearly

Within the support of the training distribution, MLPs are universal function approximators. In the NTK regime, ReLU MLPs converge to linear functions outside the training distribution with a linear convergence rate. Instead, the non-linearities in the architecture are the crucial foundation for encoding task-specific non-linearities. Compare Graph Neural Networks and Dynamic Programming Problems.

2.2 Invariant Causal Mechanisms

2.3 Optimization Objective Reformulations

2.3.1 Risk Extrapolation

2.3.2 Invariant Risk Minimization

Fully Informative Invariant Features

But I argue that there is a far larger drawback.

Learning Prime Numbers

Consider the decision problem PrimeNumbers. It is widely believed that there is no efficient algorithm that decides prime numbers.

Empirically, none of the above algorithms perform better than ERM.

Chapter 3

Recursion and Simplicity

3.1 Kolmogorov Complexity

3.2 The Expressive Power of Recursion

3.3 The Descriptive Simplicity of Recursion

3.3.1 Representation in Numeral Systems

Humans do not perform operations on a number as a whole, as current mathematical models do. Instead, they represent numbers in numeral systems, in particular the decimal system. This transformation not only into a representation that allows Number is not regarded as a real number.

3.4 The Simplest Algorithm

By nature, our organism strives to minimize the energy it requires to perform a certain operation. This also applies to our brain. When trying to make sense of our world, our brain tries to do so in the most efficient way. But in terms of what kind of efficiency?

Let us illustrate this question with the example of inferring time series from few samples. Observing the sequence ... 2 _ _ _ 2..., our brains might assume the constant function $f(n) = 2$. But given ... 2 _ 4 _ _ ..., would the brain assume a linear sequence ... 2 3 4 5 ...? Or would it rather assume a constant function with one exception ... 2 2 4 2 .... And after the next sample extends the overall image to ... 2 _ 4 _ 2..., is the constant function with one exception now still the most plausible assumption? Or do we in fact deal with a symmetric piece-wise linear function ... 1 2 3 4 3 2 1 ... instead?

In general, both assumptions are reasonable, as they fit simple patterns on the observed sequences. But as more and more 2s are joining the overall image, the constant function with the exception becomes more and more plausible. While it remains a suitable candidate for the underlying pattern, alternative pattern classes become more and more complex with additional samples, and thus more and more energy-intensive. Within our inductive bias

that follows some principle of simplicity, such as Ockham's Razor, the simplest algorithm becomes more and more likely to generate the observed patterns.

3.4.1 Computational Simplicity and Complexity

The term "simplicity" might hint at the *descriptive* efficiency of an algorithm or concept. The less resources are needed to describe the algorithm, the more efficient will a machine or human be able to store this piece of information. However, this aspect does not fully capture the multi-sided shape of efficiency. Another side is the executive efficiency, that describes how many resources it needs to execute a certain algorithm. These resources include at least time, memory, but in a more general setting also communication costs between different involved units.

But given ... 2 4 8 16 ..., I do not assume a cubic polynomial, but an exponential function $f(n) = 2^n$. Although the function values are exponential in the input, the computational complexity need not be, depending on which operations the underlying architecture allows. An architecture that features bit shift operations in constant time will allow an algorithm that computes f with linear computational complexity. Moreover, its descriptive complexity is far lower than the growing complexity of the polynomial alternatives.

Chapter 4

Identifying the Simplest Consistent Algorithm

We fix a universal reference machine U and consider its induced enumeration of partial computable functions $f_1, f_2, \ldots$. Let $f : \Sigma^* \to \Sigma^*$ be an arbitrary, partial computable function over an arbitrary, but fixed, finite alphabet Σ. For any $D \subseteq \Sigma^*$, we define

$$K_U(f \mid D) := \min_{\mathcal{S} \subset (D \times \Sigma^*)^*} \{|S| \mid \text{The smallest } f_i \text{ consistent with } \mathcal{S} \text{ satisfies } f_i \equiv f\}, \text{ and} \tag{4.1}$$

$$K_U(f) := \min_{D \subseteq \Sigma^*} K(f \mid D). \tag{4.2}$$

4.1 Out-Sampling Erroneous Simpler Algorithms

What qualitative and quantitative criteria must the training sample meet to ensure that the simplest algorithm that is consistent with the sample truly coincides with the true function?

4.2 Hypothesis Certification

4.3 Retrievability of the Algorithm

We avail to the idea of [8] to derive sufficient conditions when the true algorithm is retrievable.

Appendix A

Theory and Proofs

A.1 Out-of-distribution limitations

A.1.1 Extrapolation of ReLU MLPs with L2-Regularization

A.1.2 Prime Numbers are not learnable by IRM

A.2 Properties of Kolmogorov Complexity

A.2.1 Unbounded Kolmogorov Complexity across Reference Machines

A.2.2 No Order Preservation between L2 Norm and Kolmogorov Complexity

A.3 Information Thresholds for Learnability

A.3.1 Abysmal Minimal Sample Count to Kolmogorov Complexity Ratio

We are interested if there exists a function m that - given the Kolmogorov complexity of a function f - serves a lower bound on the required sample count to identify f in the learning by enumeration framework. As it turns out, there is no effective lower bound. In other words, the sample count might become as small as one for arbitrarily high Kolmogorov complexities.

This demonstrates that it should not be the sample count we seek to relate the Kolmogorov complexity to, but instead quantify the training sample itself by its Kolmogorov complexity, too.

Proof.
Since Σ^* is countably infinite, consider an arbitrary order of Σ^* as $x_1, x_2 \ldots$. Fix an arbitrary definition range $D \subseteq \Sigma^*, D \neq \emptyset$, and an arbitrary $c \in D$. For any x_j, let f_{i_j} be the simplest function consistent with $\mathcal{S}_j := \{(c, x_j)\}$. Then, f_{i_j} is learnable from the Kolmogorov enumeration by a sample of size 1.

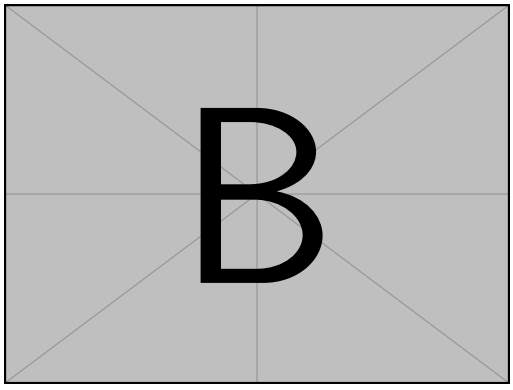

Figure A.1: Caption text [9].

Table A.1: The error function and complementary error function

x	erf(x)	erfc(x)	x	erf(x)	erfc(x)
0.00	0.00000	1.00000	1.10	0.88021	0.11980
0.60	0.60386	0.39614	1.8214	0.99000	0.01000
0.70	0.67780	0.32220	1.90	0.99279	0.00721

Obviously, $f_{i_j} \not\equiv f_{i_k}$ for any $j, k \in \mathbb{N}$ with $j \neq k$, since they disagree for input c.

Therefore, there are infinitely many functions over Σ^* that are learnable from the Kolmogorov enumeration by a sample of size 1. However, for each $n \in \mathbb{N}$, there are only finitely many objects with Kolmogorov complexity n.

A.3.2 Kolmogorov Complexity Upper Bound for K_U

A.3.3 Unbounded K_U across U for fixed finite D

A.3.4 Unbounded K_U across U for fixed countably infinite D

A.3.5 Lower Bound for K_U

A.4 Placeholder

References

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[2] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, e253, 2017.

[3] M. Li, P. Vitányi, *et al.*, *An introduction to Kolmogorov complexity and its applications*. Springer, 2008, vol. 3.

[4] J. Peters, P. Bühlmann, and N. Meinshausen, "Causal inference by using invariant prediction: Identification and confidence intervals," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 78, no. 5, pp. 947–1012, 2016.

[5] N. Pfister, P. Bühlmann, and J. Peters, "Invariant causal prediction for sequential data," *Journal of the American Statistical Association*, vol. 114, no. 527, pp. 1264–1276, 2019.

[6] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[7] K. Ahuja, E. Caballero, D. Zhang, J.-C. Gagnon-Audet, Y. Bengio, I. Mitliagkas, and I. Rish, "Invariance principle meets information bottleneck for out-of-distribution generalization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3438–3450, 2021.

[8] J. Richens and T. Everitt, "Robust agents learn causal world models," *arXiv preprint arXiv:2402.10877*, 2024.

[9] K. Gödel, "Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i," *Monatshefte für mathematik und physik*, vol. 38, pp. 173–198, 1931.