Simulating Evolvability as a Learning Algorithm: Empirical Investigations on Distribution Sensitivity, Robustness, and Constraint Tradeoffs

Nico Fidalgo[1] and Puyuan Ye[1]

[1]Harvard University

April 24, 2025

Abstract

This project explores the theory of evolvability, as proposed by Valiant [1], through empirical simulations using genetic algorithms. We practically validate the evolvability of theoretically provably evolvable function classes such as monotone conjunctions and disjunctions, and investigate the practical limitations of evolvability for classes like parity. We then expand upon Valiant's framework by exploring three extensions: (1) whether significant classes are empirically provably evolvable for all distributions, (2) the tradeoff between robustness and complexity of evolvable mechanisms by relaxing arbitrary starting point assumptions, and (3) how constraining the evolution process (e.g., disallowing neutral mutations or fixing tolerance) impacts convergence. Our goal is to simulate a learning process grounded in biological evolution, deepen our understanding of evolvability, and quantify where empirical findings align or diverge from theoretical expectations.

1 Introduction

Evolutionary processes have long been studied as optimization mechanisms in biology. *Evolvability* by Valiant [1] formalizes this notion within the framework of computational learning theory, posing evolvability as a constrained variant of PAC learnability and the Statistical Query (SQ) model. This project simulates evolvability using a genetic algorithm (GA) and investigates the extent to which various Boolean functions are evolvable under theoretical and empirical constraints.

2 Motivation and Research Questions

We aim to validate and extend Valiant's theory by addressing the following questions:

1. Do classes that are theoretically evolvable (e.g., monotone conjunctions) also evolve in practice under GA-based simulation?
2. For which classes is evolvability empirically distribution-independent?
3. What is the tradeoff between robustness (arbitrary starting points) and the complexity of mechanisms that can evolve in practice?
4. How do model constraints (e.g., no neutral mutations, fixed tolerance) affect the evolvability of a class?

3 Theoretical Framework

Our implementation is inspired by the definitions and constraints introduced in Valiant's framework [1]:

- Representations (R) are binary encodings of Boolean hypotheses.
- Neighborhoods are defined by $p(n, 1/\epsilon)$-bounded mutation functions.
- Empirical performance is based on a fitness function measuring agreement with a target concept f on examples drawn from a distribution D_n.
- Mutators probabilistically select better or neutral variants from a candidate pool, mimicking selection in evolution.

4 Methodology

4.1 Target Functions

We use synthetic data generated using various Boolean functions:

- Monotone Conjunctions (e.g., $x_1 \wedge x_3$)
- Monotone Disjunctions (e.g., $x_2 \vee x_4$)
- Parity Functions (odd/even number of 1's)
- Majority Functions

4.2 Genetic Algorithm Implementation

- **Encoding:** Each candidate is a binary string representing a hypothesis.
- **Fitness Function:** Fraction of correctly labeled outputs on a sampled dataset.
- **Mutation and Crossover:** Applied with tunable parameters.
- **Selection:** Tournament or fitness-proportional selection.

4.3 Experimental Plan

Our experiments are organized into three phases:

Phase I: Baseline Validation

- Test empirical evolvability of conjunctions, disjunctions, parity.

Phase II: Distributional Sensitivity

- Evaluate performance under different distributions: uniform, biased Bernoulli, clustered.
- Analyze distribution-robustness of evolvable classes.

Phase III: Constrained Models

- **Fixed Starting Point:** Remove randomness in r_0 and analyze its effect.
- **Disallow Neutral Mutations:** Require strictly beneficial performance improvements.
- **Fixed Tolerance t:** Explore if convergence persists under constant thresholding.

5 Expected Outcomes

- Reproduce theoretical claims: conjunctions and disjunctions evolve, parity does not.
- Provide novel insights into when evolvability depends on the input distribution.
- Empirically map tradeoffs between constraints, robustness, and convergence.
- Contribute to understanding evolvability as a practical learning process beyond its theoretical definition.

6 Conclusion

This work bridges theory and practice in understanding evolvability, grounding our simulations in formal definitions and extending them with new empirical analyses of robustness and constraint. The framework developed here may also serve as a testbed for future theoretical questions in learning and evolution.

References

[1] L. G. Valiant. Evolvability. *Journal of the ACM (JACM)*, 56(1):3, 2009.