Evolvability as Learnability

Sumeet Agarwal
Evolution as Learning

- You've heard about machine learning, you've heard about biological evolution; what's the connection?
- From a computer science perspective, what Darwin described was an optimisation algorithm
- We seek to use statistical learning to try and figure out how biology works; wouldn't nature have needed to do similarly?
- If evolution is just a form of inter-generational learning, then is nature vs. nurture irrelevant?
Evolvability (Valiant 2009)

- The puzzle of evolvability is how complex mechanisms can evolve without unlikely events.
- To get to a specific target, there needs to be an evolutionary path consisting of small steps.
- What are the conditions in biology that allow such paths to be taken routinely and efficiently?
- If we view biological mechanisms as mathematical functions, considerations of computational complexity allow us to examine what sorts of function classes might be evolvable.
- Previous quantitative models of evolution (population dynamics, evolutionary game theory, genetic algorithms) have not quantified what can and can't be evolved with feasible resources.
Four Basic Notions for Valiant's Evolvability

- Evolution of *Many-argument functions*: $f(x_1, x_2, \ldots, x_n)$

- Measure of *performance* over a distribution of inputs: $\text{Perf}_f(r, D_n) = \sum_x f(x) r(x) D_n(x)$

- *Limited population* per generation

- Convergence to complex mechanisms in a *limited number of generations*
Evolution as Optimisation

- $\text{Perf}_f(r, D_n)$ defines a combinatorial fitness landscape over $r$ (let’s assume all elements/genes are binary: on/off)
- For instance, on two inputs there are $2^{2^2} = 16$ possible functions

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f$</th>
<th>$r_1$</th>
<th>$r_2$</th>
<th>$r_3$</th>
<th>$r_4$</th>
<th>$r_5$</th>
<th>$r_6$</th>
<th>$r_7$</th>
<th>$r_8$</th>
<th>$r_9$</th>
<th>$r_{10}$</th>
<th>$r_{11}$</th>
<th>$r_{12}$</th>
<th>$r_{13}$</th>
<th>$r_{14}$</th>
<th>$r_{15}$</th>
<th>$r_{16}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

$\text{Perf}_f(r, U_2)$ vs $i$
Conditions for feasible evolvability

• In practice, the search cannot be over all functions but must be restricted to some class C (e.g., disjunctions like \(x_1 \text{ OR } x_2\ldots\))

• We can assume appropriate small mutations, such as adding/deleting/swapping a single variable in a disjunction

• For a given candidate \(r()\), fitness relative to the ideal \(f()\) will actually be evaluated by sampling from \(D_n\), i.e., based on a finite set of ‘life experiences’ \(Y\):

  \[
  \text{Empirical Performance } v(r) = \frac{1}{|Y|} \sum_{x \in Y} f(x) r(x), \ Y \sim D_n
  \]

• Then there must exist a sequence \(r_0 \rightarrow r_1 \rightarrow r_2 \ldots\) such that each \(r_i\) follows from \(r_{i-1}\) after a single step of mutation and selection from a moderately sized population, and after a moderate number of steps \(i\), \(r_i\) is significantly better than \(r_0\) (i.e., detectable over a moderate number of experiences or samples)
Algorithmic Framework

- If we assume $n$ genes and allow for a performance error of $\varepsilon$, then we will bound the population size by $p(n, 1/\varepsilon)$, a polynomial function, and set a performance tolerance $t$.

- Within a neighbourhood of this size for a given $r_0$, evaluate empirical performance $v(r_1)$ for all candidates, and define sets $Bene = \{r_1 \mid v(r_1) \geq v(r_0) + t\}$ and $Neut = \{r_1 \mid v(r_1) \geq v(r_0) - t\} - Bene$.

- If $Bene$ is non-empty, pick $r_1$ randomly from it, else pick $r_1$ randomly from $Neut$. 

![Diagram showing the p(n,1/ε)-neighbourhood and the evaluation of empirical performance](image-url)
What is evolvable?

• In this framework, we will say a function class $C$ is *evolvable* for a given distribution $D_n$ over inputs if there exists an algorithm such that for any $f \in C$, $r_0$, $\epsilon > 0$, we have with probability at least $1 - \epsilon$ a sequence $r_0 \rightarrow r_1 \rightarrow r_2 \rightarrow \ldots \rightarrow r_N$, $N = g(n,1/\epsilon)$, another polynomial, and $v(r_N) > 1 - \epsilon$

• This notion of evolvability is a restricted case of a notion known in computational learning theory (COLT) as *PAC* (probably approximately correct) learnability

• Using results from COLT, it can be shown that certain classes of functions are evolvable in this sense, whilst others are not (or at least not unless P=NP)
<table>
<thead>
<tr>
<th>Function Class</th>
<th>Example</th>
<th>Evolvable for uniform distribution over instance space?</th>
<th>Evolvable for arbitrary distribution over instance space?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone Boolean conjunctions or disjunctions</td>
<td>$f(x) = x_{i[1]} x_{i[2]} x_{i[3]}$</td>
<td>Y</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>$f(x) = x_{i[1]} \lor x_{i[2]} \lor x_{i[3]}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parity functions</td>
<td>$f(x) = 1$ iff an odd number of ${x_{i[1]}, x_{i[2]}, ..., x_{i[k]}}$ are 1</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Decision lists</td>
<td>If $x_{i[1]} x_{i[2]}$, then $f(x)=1$</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>Else if $x_{i[1]} x_{i[3]}$, then $f(x)=-1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Else if $x_{i[2]} x_{i[4]}$, then $f(x)=-1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Else $f(x)=1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear threshold functions (aka halfspaces, perceptrons)</td>
<td><img src="image.png" alt="Graph" /></td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
Extensions: a role for ‘junk’ DNA?

• Evolution with optimisation is likely to work better
• This can be achieved within the same framework by having a redundant representation: i.r₀.rᵢ, where we continue to use r₀ to generate mutations for i=1…N, and then reset r₀=r_N
• It can be shown that N=3/t virtually ensures we will get within 2t of the best empirical performance in the p-neighbourhood of r₀
Limitations

• One could argue that the model is biologically unrealistic or too simplistic, as for other quantitative models of evolution.

• However the point is not so much to simulate actual evolution as to identify the sorts of things that are evolvable under natural resource constraints; these constraints may be more universal than any particular evolutionary algorithm.

• Also, some extensions such as allowing for population variability can be expressed fairly easily within the model.

• It is not yet known whether any non-trivial function class is evolvable for arbitrary distributions over input space.

• Assumptions such as linearity of genotype-fitness map, constancy of fitness landscape: more recent papers suggest results are robust to dropping at least some of these.
Conclusions and Outlook

• “Attempt to capture the Darwinian notion of random variation with selection in terms of a computationally compelling model”

• “Evolution of significant algorithmic structure is a predictable and analyzable phenomenon” (e.g., conjunctions and disjunctions)

• “The unified viewpoint on learning and evolution... suggests that cognitive systems can be viewed as pure learning systems”

• Multiphase learning (with varying target functions) can lead to arbitrarily complex circuits: links to Kashtan/Alon work?

• How has biology ‘chosen’ C? Is this where physics and chemistry come in? Do restrictions on C lead to modularity?

• What about the role of network structure? Can things like ‘scale-freeness’ be explained; can models of evolution help with biological network inference?
References


