Genetic Programming

- Automatic induction of computer programs from samples
- Sample (pair of):
  - Set of arguments
  - Desired output value
- Program representation
  - Syntax tree
  - Linear (like assembler)
  - Graph
  - and more...

Genetic Programming

- Prefix notation:
  - (+ (/ x (* x x)) (* x x))
- No explicit memory storage

GP typical tasks

- Symbolic regression
- Classification
- Planning and control
- Logic circuit synthesis
- Evolvable hardware

Genetic operators: subtree crossover

- Is the result predictable?
  - Yes, but...
- Crossover is supposed to produce offspring between parents
  - Average in common sense
- Are \( x(x-2) \) or \( x^2 \) between \( x^2 + x^2 \) and \( x - x(x - 2) \)?
What does `between` mean for programs?

- Point may be between some other points only in a metric space
- We need a metric \( d: P \times P \to [0, +\infty) \) defined on program space \( P \):
  - \( d(a, b) = 0 \iff a = b \)
  - \( d(a, b) = d(b, a) \)
  - \( d(a, b) \leq d(a, c) + d(b, c) \)
- But... how to define a metric on pair of programs? We address this later.

Genetic operators: mutation

- Mutation is supposed to make an elementary change to the given solution
- Is replacement of whole subtree an elementary change?
- Is \( x^2 + x^2 \) similar to \( x^2 - x^4 + x^2 \)?

What does `similar` mean for programs?

- How similar is \( + \) to \( - \)?
- What about \( + \) and \( / \)?
- Again:
  - We need a metric
  - How to define a metric on instructions?

Semantics

- We induce programs from samples
- The samples are sets of numbers (in symbolic regression)
  - Set of function arguments
  - The desired output value
- Let us use similar representation as semantics
  - Set of function arguments
  - The calculated output value
- Call it sampled semantics

Semantics: example

Consider functions \( f(x) = \frac{x}{2} + x^2 \) and \( g(x) = \frac{2x}{3} + x^2 \)

Sample it equidistantly in range \([-1, 1]\) using 10 samples

<table>
<thead>
<tr>
<th>x</th>
<th>f(x)</th>
<th>g(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0</td>
<td>0.59</td>
<td>0.33</td>
</tr>
<tr>
<td>0.0</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.5</td>
<td>0.54</td>
<td>1.14</td>
</tr>
<tr>
<td>0.2</td>
<td>0.33</td>
<td>1.14</td>
</tr>
<tr>
<td>0.1</td>
<td>0.11</td>
<td>1.35</td>
</tr>
<tr>
<td>0.0</td>
<td>0.00</td>
<td>1.35</td>
</tr>
<tr>
<td>-0.3</td>
<td>1.89</td>
<td>1.35</td>
</tr>
<tr>
<td>-0.5</td>
<td>3.11</td>
<td>1.35</td>
</tr>
<tr>
<td>-0.7</td>
<td>5.00</td>
<td>1.35</td>
</tr>
<tr>
<td>-1.0</td>
<td>5.00</td>
<td>1.35</td>
</tr>
</tbody>
</table>

Again: How similar is \( f(x) \) to \( g(x) \)? Just chose a metric:

- Manhattan: 32.93
- Euclidean: 14.48
- Chebyshev: 15.33

Semantics in context of GP

- Computed every time a program is evaluated
  - The fitness function is some kind of distance measure
  - It is essentially free to obtain
  - A part of program is also a program, that can be executed
  - Semantics can be calculated in (almost) every node of the tree
Sampled semantics: properties

Advantages
- Similar representation to the way, how problem is posed
- Many distance metrics (any Minkowski distance $L_p$)
- Low computational costs (in context of GP)
- Extendable to any precision and any number of values (e.g. complex numbers)

Disadvantages
- Does not contain whole information about subject (it’s only a sample)
- Problem-dependent (arguments)

Geometric genetic operators

In a metric space
- The object may be between some other objects

- The object may be in a given perimeter of other object

“A recombination operator is a geometric crossover under the metric $d$ if all offspring are in the $d$-metric segment between its parents.”

“...”

Geometric crossover

- So, we can calculate (range of) semantics between semantics of parents $s(p_1)$ and $s(p_2)$

- But... how to obtain a program having desired semantics?
  - If it were easy, we would not need an optimization algorithm

How do we obtain a program having semantics intermediate between two other programs?

- We can build a library of programs
- How big should this library be?
  - Too few programs:
    - We may be not able to find the desired one
  - Too many programs:
    - We could not store the library in memory (slow access)
    - Infinite number of programs...

Why do we need the geometric crossover?

- Consider:
  - the Euclidean distance as a fitness/error function
  - fitness landscape spanned over k-dimensional space of program semantics
  - It must be a cone
  - The vertex is the global optimum
  - Programs lie on the edges of cone

Not possible for many real-world problems.
Why do we need the geometric crossover?

- It is guaranteed that:
  - An intermediate semantics between any pair of semantics must be not worse than the worst of the pair.
  - A sketch of proof:
    - If the pair lies on a single side of cone:
      - The fitness of intermediate solution must be between fitness values defined by the pair.
    - If the pair lies on opposite sides of cone:
      - The fitness of intermediate solution must be not worse than fitness of the worst of pair.

Locally Geometric Semantic Crossover (LGX)

- Choose a homologous crossover point (syntactically).
- Calculate average semantics between subtrees rooted at chosen point.
- Use library to find the closest procedure to the calculated semantics.
- Place the found procedure at crossover point in both parents.

Locally Geometric Semantic Mutation (LGM)

- Similar to LGX.
- Randomly choose mutation point.
- Choose a procedure from library according to the Poisson distribution (with given \( \lambda \)).
- Replace the subtree rooted at mutation point with the chosen procedure.
- Rationale:
  - The change cannot be too little.
  - The change cannot be too big.

Competition

- Semantic-Aware Crossover (SAC)
- Semantic Similarity-based Crossover (SSC)
- Semantic-Aware Mutation (SAM)
- Semantic Similarity-based Mutation (SSM)

Control methods

- Tree Swapping Crossover (GPX)
- One Point Crossover (GPH)
- Nonhomologous Geometric Crossover (NHX)
- Random Crossover (RX)
The experiment

Benchmark problems

Success rate (%)

Statistical significance

Friedman's test for multiple achievements of a series of subjects on the average of best-of-run fitness

<table>
<thead>
<tr>
<th>Problem</th>
<th>Definition (formula)</th>
<th>Training set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sextic</td>
<td>( \sin^2 x + \cos^2 x )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Septic</td>
<td>( \sin^2 x + \cos^2 x )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Nonic</td>
<td>( \sin^2 x + \cos^2 x )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>R1</td>
<td>( x + 1 )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>R2</td>
<td>( (x^2 + 1)/2 )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>R3</td>
<td>( x^2 + 1 )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Nguyen-5</td>
<td>( \sin(x)\cos(x) )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Nguyen-6</td>
<td>( \sin(x)\cos(x) )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Nguyen-7</td>
<td>( \sin(x)\cos(x) )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Keijzer-1</td>
<td>( \sin(x)\cos(x) )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Keijzer-4</td>
<td>( \sin(x)\cos(x) )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
<tr>
<td>Keijzer-7</td>
<td>( \sin(x)\cos(x) )</td>
<td>([{-1, 1}, 20])</td>
<td>([{-1, 1}, 20])</td>
</tr>
</tbody>
</table>

Success rate (%)

<table>
<thead>
<tr>
<th>Problem</th>
<th>GPX</th>
<th>GPH</th>
<th>LGX3</th>
<th>LGX4</th>
<th>NGX3</th>
<th>NGX4</th>
<th>RX3</th>
<th>RX4</th>
<th>SAC</th>
<th>SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sextic</td>
<td>85</td>
<td>3</td>
<td>31</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Septic</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonic</td>
<td>8</td>
<td>3</td>
<td>100</td>
<td>1</td>
<td>73</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>50</td>
<td>9</td>
<td>91</td>
<td>3</td>
<td>78</td>
<td>3</td>
<td>22</td>
<td>1</td>
<td>53</td>
<td>48</td>
</tr>
<tr>
<td>R2</td>
<td>6</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nguyen-3</td>
<td>41</td>
<td>91</td>
<td>24</td>
<td>71</td>
<td>34</td>
<td>63</td>
<td>56</td>
<td>75</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Statistical significance

- Friedman's test for multiple achievements of a series of subjects on the average of best-of-run fitness
- \( p = 2.589 \times 10^{-5} \)
- Post-hoc analysis (symmetry test)

P values

- GPX: 0.310, 0.899, 0.899, 1.000, 1.000
- GPH: 0.487, 0.034, 0.034, 0.034, 0.034
- LGX3: 0.149, 0.000, 0.000, 0.000, 0.000
- LGX4: 0.016, 0.000, 0.000, 0.000, 0.000
- NGX3: 0.214, 0.000, 0.000, 0.000, 0.000
- NGX4: 0.840, 0.002, 0.002, 0.002, 0.002
- RX3: 0.197, 0.017, 0.017, 0.017, 0.017
- RX4: 0.004, 0.004, 0.004, 0.004, 0.004
- SAC: 0.015, 0.015, 0.015, 0.015, 0.015
- SSC: 0.821, 0.000, 0.000, 0.000, 0.000

P values of Friedman's test are shown on the diagonal in bold. Lower values mean stronger significance.
Comparative analysis of performance of LGM

Future work

- Comparative analysis of performance of LGM
- Analysis of propagation of geometric changes done by LGX
- Geometric and semantically-based initialization of population
- Move the concept of semantically geometric operators outside GP
- Local search heuristics

References

Thank you
Questions?