Automatic Derivation of Search Objectives for Test-Based Genetic Programming

Krzysztof Krawiec   Paweł Liskowski

Computational Intelligence Group
Laboratory of Intelligent Decision Support Systems
Institute of Computing Science
Poznan University of Technology, Poland

evo* 2015

April 8, 2015

Evaluation Bottleneck

- Solving programming task using GP:

$$\arg\max_{s \in \mathbb{S}} f(s) \tag{1}$$

- Fitness function *aggregates* the behavior of s on *tests* by
  - Counting the number of passed tests (discrete domains).
  - Summing the errors on individual tests (continuous domains).

Evaluation Bottleneck

- Solving programming task using GP:

$$\arg\max_{s \in \mathbb{S}} f(s) \tag{1}$$

- Fitness function *aggregates* the behavior of s on *tests* by
  - Counting the number of passed tests (discrete domains).
  - Summing the errors on individual tests (continuous domains).
- Behaviorally rich evaluation process, yet limited feedback for the search algorithm.
  - Example: 6-bit multiplexer, $2^6 = 64$ tests.
  - Number of possible 'output behaviors': $2^{64} = 1.84 \times 10^{19}$
  - Number of possible fitness values: $2^6 + 1 = 65$
- Fitness conveys little information on s: **evaluation bottleneck**.

Implications

Implications of evaluation bottleneck:

- Compensation $\implies$ indiscernibility in selection
- All tests considered equally difficult (same rewards)
- Low fitness-distance correlation.

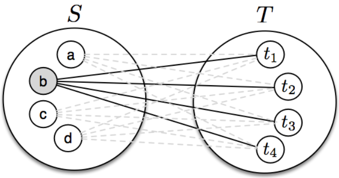
Question

Detailed information on solutions' interactions with individual tests **is available in interaction matrix**.
How to exploit that information?

Test-based Genetic Programming

Program synthesis = test-based problem.

- S: set of m programs, $S \subset \mathbb{S}$
- T: set of n tests (fitness cases), $T \subset \mathbb{T}$
- $g(s, t)$: interaction function between $s \in S$ and $t \in T$
  - passing test: $g(s, t) = 1$, failing test: $g(s, t) = 0$
- G: $m \times n$ matrix of interaction outcomes between S and T.



$g : S \times T \to [0, 1]$ $\longrightarrow$

G	t_1	t_2	t_3	t_4
a	1	1	0	0
b	0	1	0	1
c	1	1	1	0
d	0	1	1	0

See: (Bucci, Pollack, de Jong, 2000 and on), (Popovici et al. 2011)
Also: *behavioral GP* (Krawiec & Swan 2013; Krawiec & O'Reilly 2014)

Searching for structure in G

Idea

Identify groups of tests on which the programs behave **similarly**.

Hypothesis

- Interaction matrix can be clustered into a few derived objectives that approximately capture the skills exhibited by the programs.
- Objectives obtained in this way can be better **search drivers**.

DOC: **D**iscovery of Search **O**bjectives by **C**lustering

Algorithm

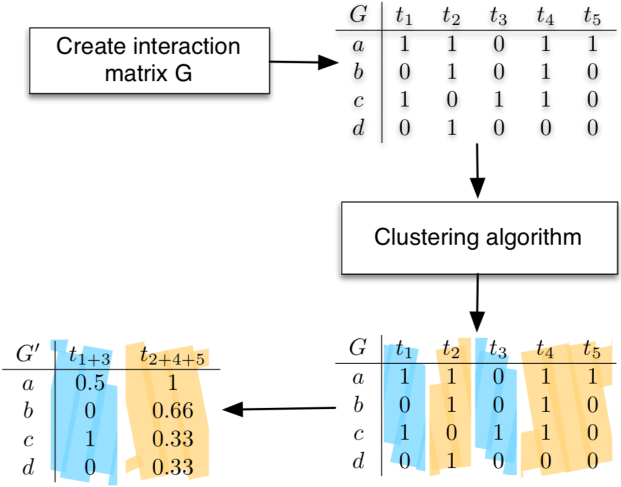
1. Calculate $m \times n$ interaction matrix between S and T.
2. Cluster n tests into clusters $\{T_1, \ldots, T_k\}$.
3. Define the derived objectives. For each T_j average row-wise the corresponding columns in G. The result is $m \times k$ matrix G':

$$g'_{i,j} = \frac{1}{|T_j|} \sum_{t \in T_j} g(s_i, t)$$

4. Use g'_js as **derived objectives**.

See also: (Liskowski & Krawiec, PPSN 2014)

Example



G	t_1	t_2	t_3	t_4	t_5
a	1	1	0	1	1
b	0	1	0	1	0
c	1	0	1	1	0
d	0	1	0	0	0

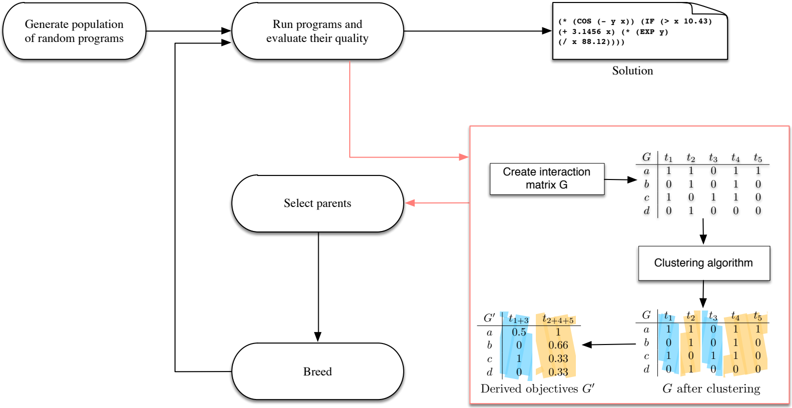
G	t_1	t_2	t_3	t_4	t_5
a	1	1	0	1	1
b	0	1	0	1	0
c	1	0	1	1	0
d	0	1	0	0	0

G after clustering

G'	t_{1+3}	t_{2+4+5}
a	0.5	1
b	0	0.66
c	1	0.33
d	0	0.33

Derived objectives G'

Example



- Black: convetional GP
- Red: GP with DOC

Experiment

Methods:

- GP: 'vanila' GP
- IFS: Implicit Fitness Sharing (Smith et al. 1993, McKay 2000)
- DOC: Discovery of Objectives by Clustering
  - Clustering algorithm: X-means (Pelleg et al. 2000), chooses k autonomously
  - Multiobjective selection: NSGA-II (Deb et al. 2002)
- RAND: As DOC, but tests clustered at random
  - Controls for the relevance of clustering.

Domain	Instruction set	Problem	Variables	Fitness cases	Space size
Boolean	and, nand, or, nor	Cmp6	6	64	2^{64}
		Cmp8	8	256	2^{256}
		Par5	5	32	2^{32}
		Mux6	6	64	2^{64}
		Maj6	6	64	2^{64}
Categorical	$a_i(x, y)$	Disc-a1...a5	3	27	3^{27}
	$a_i(x, y)$	Malcev-a1...a5	3	15	3^{15}

Experiment: Initial Results

Average ranks on success rate over all 15 benchmarks:

Population size: 500			
DOC	IFS	RAND	GP
1.93	2.20	2.50	**3.36**

Population size: 1000			
DOC	IFS	RAND	GP
1.76	2.33	2.60	**3.30**

(Friedman's p-value $\ll 0.001$)

DOC ranks better than IFS. Statistical significance?

Experiment: Initial Results

Average ranks on success rate over all 15 benchmarks:

Population size: 500			
DOC	IFS	RAND	GP
1.93	2.20	2.50	**3.36**

Population size: 1000			
DOC	IFS	RAND	GP
1.76	2.33	2.60	**3.30**

(Friedman's p-value $\ll 0.001$)

DOC ranks better than IFS. Statistical significance?

What's up, DOC?

Overspecialization? Focusing?

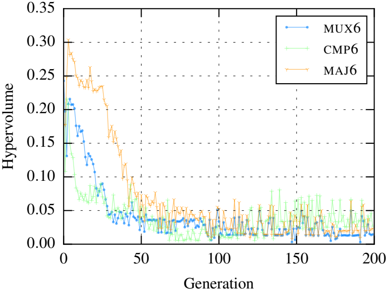
Hypothesis

Effect of **focusing** on some derived objectives.

Hypervolume of program's performance as characterized by the k derived objectives $g_1, \ldots, g_k$, i.e.,

$$h(p) = \prod_{j=1}^{k} g_j(p)$$

Maximized when the scores on g_js are balanced.



Average hypervolume of programs

- DOC-P: Hypervolume: $\prod_{j=1}^{k} g_j(p)$
- DOC-D: Weighs g_js by the number of tests: $\prod_{j=1}^{k} |T_j| g_j(p)$

Population size: 500					
DOC-D	DOC-P	IFS	DOC	RAND	GP
1.70	2.43	3.56	**3.63**	**4.33**	**5.33**

Population size: 1000					
DOC-P	DOC-D	DOC	IFS	RAND	GP
2.20	2.43	3.10	3.66	**4.50**	**5.10**

Success rate	$\|P\| = 500$						$\|P\| = 1000$					
	GP	IFS	RAND	DOC	DOC-P	DOC-D	GP	IFS	RAND	DOC	DOC-P	DOC-D
Cmp6	20	**100**	50	21	83	78	26	**97**	48	22	64	77
Cmp8	0	**56**	0	0	21	29	0	**7**	0	0	4	5
Disc1	0	0	0	7	3	**13**	0	0	0	**10**	**10**	7
Disc2	0	4	0	10	14	**37**	0	0	0	0	21	**40**
Disc3	0	0	0	18	53	**62**	0	0	0	56	71	**77**
Disc4	0	0	0	0	0	**7**	0	0	0	**4**	0	0
Disc5	0	0	0	0	**7**	3	0	0	0	0	**4**	**4**
Maj6	22	**100**	60	40	83	90	52	**100**	71	81	96	89
Malcev1	0	18	24	18	70	**76**	14	27	33	25	69	**93**
Malcev2	3	3	0	7	27	**30**	0	0	11	17	**32**	27
Malcev3	0	7	8	23	**83**	**83**	0	3	8	43	**93**	75
Malcev4	0	0	4	7	**10**	7	0	0	0	**25**	20	10
Malcev5	17	30	25	54	47	**57**	17	23	44	**100**	68	60
Mux6	77	**100**	83	73	**100**	**100**	90	**100**	96	**100**	**100**	**100**
Par5	0	14	14	**18**	7	12	4	6	0	**18**	3	0

Conclusions

Small picture:

- Derived search objectives effectively enhance conventional GP.
- DOC addresses some shortcomings of scalar evaluation:
  - Characterizes programs with multiple objectives ('skills')
  - Allows multiobjective approach to the problem.

Big picture:

- Derived objectives are examples of **search drivers**: measures designed to *guide* the search process.
  - Search drivers: relative, contextual, non-stationary
  - Objective functions: absolute, context-free, stationary
- Conventional objective functions are not necessarily good search drivers.
- Ongoing work on formalization and principled design of search drivers.

Conclusions

Small picture:

- Derived search objectives effectively enhance conventional GP.
- DOC addresses some shortcomings of scalar evaluation:
  - Characterizes programs with multiple objectives ('skills')
  - Allows multiobjective approach to the problem.

Big picture:

- Derived objectives are examples of **search drivers**: measures designed to *guide* the search process.
  - Search drivers: relative, contextual, non-stationary
  - Objective functions: absolute, context-free, stationary
- Conventional objective functions are not necessarily good search drivers.
- Ongoing work on formalization and principled design of search drivers.

Thank You