Non-negative Matrix Factorization for Derivation of Search Objectives in GP

Paweł Liskowski   Krzysztof Krawiec

Computational Intelligence Group
Institute of Computing Science
Poznan University of Technology, Poland

July 22, 2016

Fitness Evaluation

Observation 1

GP algorithms do not let programs know **which** tests they solve.

Typical fitness function in GP *aggregates* program's behavior on *tests* by

- counting the number of passed tests (discrete domains).

$$f(p) = |\{y_i \neq \hat{y}_i(p)\}|_i \tag{1}$$

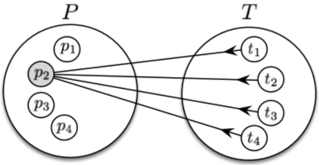
- summing the errors on individual tests (continuous domains).

$$f(p) = \sum_i (y_i - \hat{y}_i(p))^2 \tag{2}$$

Test-based Genetic Programming

Observation 2

Detailed information on particular interactions **is available** in GP.



$$g : P \times T \rightarrow [0, 1]$$

G	t_1	t_2	t_3	t_4
p_1	1	1	0	0
p_2	0	1	0	1
p_3	1	1	1	0
p_4	0	1	1	0

- P: set of m programs,
- T: set of n tests (fitness cases)
- $g(p, t)$: interaction function between $p \in P$ and $t \in T$
- G: $m \times n$ matrix of interaction outcomes between P and T
- Test-based problems (Pollack, Bucci, de Jong, Popovici)

Fitness Evaluation

Observation 3

Interaction outcomes for particular tests are *partially* dependent.

Can be used to **estimate** program-test interactions in G:

- Surrogate Fitness via Factorization of Interaction Matrix (*SFIMX*) (Liskowski & Krawiec, EuroGP 2016)

Fitness Evaluation

Observation 3

Interaction outcomes for particular tests are *partially* dependent.

Can be used to **estimate** program-test interactions in G:

- Surrogate Fitness via Factorization of Interaction Matrix (*SFIMX*) (Liskowski & Krawiec, EuroGP 2016)

Hypothesis

- $g(p, t)$ can be *explained* by decomposition into a set of **factors**,
- These factors can be used as **search drivers**.

The idea

Use **matrix factorization** to extract the factors from patterns in G.

Non-negative matrix factorization (NMF)

Given G, find W and H such that

$$G \approx WH \;\; s.t. \;\; W, H \geq 0,$$

or more precisely:

$$\underset{W,H}{min}\, f(W,H) \equiv \frac{1}{2}||G - WH||_F^2 \;\; s.t. \;\; W, H \geq 0,$$

- Effective, gradient-based algorithms exist (e.g., multiplicative update rule).
- Intensely used in machine learning (recommender systems, Netflix contest)

NMF: Example 1

$$G = \begin{array}{c c} & \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$W \times H = \begin{array}{c c} & \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 0.70 & 2.05 \\ 0.73 & 0.66 \\ 0.35 & 1.02 \end{pmatrix} \end{array} \times \begin{array}{c c} & \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{array}{c} f_1 \\ f_2 \end{array} & \begin{pmatrix} 0.70 & 0.70 & 2.70 & 2.70 \\ 0.74 & 0.74 & 0.06 & 0.06 \end{pmatrix} \end{array}$$

NMF: Example 2

$$G' = \begin{array}{c c} & \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ \color{blue}{2} & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$W' = \begin{array}{c c} & \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 0.96 & 1.51 \\ 0.39 & 1.84 \\ 0.86 & 0.38 \end{pmatrix} \end{array}, \quad H' = \begin{array}{c c} & \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{array}{c} f_1 \\ f_2 \end{array} & \begin{pmatrix} 2.16 & 1.20 & 0.72 & 0.72 \\ 0.05 & 0.35 & 0.90 & 0.90 \end{pmatrix} \end{array}$$

$$W' \times H' = \begin{array}{c c} & \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 2.17 & 1.70 & 2.07 & 2.07 \\ 0.95 & 1.13 & 1.96 & 1.96 \\ 1.88 & 1.17 & 0.97 & 0.97 \end{pmatrix} \end{array}$$

DOF: Discovery of Search **O**bjectives by **F**actorization

Algorithm

1. Calculate $m \times n$ interaction matrix G between P and T,
2. Factorize G into non-negative components W and H,
3. Define the derived objectives f'_j based on W and H,
4. Use f'_js for multiobjective evaluation/selection.

Two variants:

- DOF-W: $f_j(p) = w_{pj}$
- DOF-WH: $f_j(p) = w_{pj} \sum_{t \in T} h_{jt}$

The matrix G can be **sparse** (see Liskowski & Krawiec, EuroGP 2016)

- $\alpha \in (0, 1]$ - density of partial interaction matrix

Example

Example: $P = \{s_1, s_2, s_3\}$, $T = \{t_1, \dots, t_4\}$

$$G = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \xrightarrow[rank(G)=2]{NMF \quad k=2} W \times H = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{pmatrix} 0.70 & 2.05 \\ 0.73 & 0.66 \\ 0.35 & 1.02 \end{pmatrix} \end{array} \times \begin{array}{c} \\ f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 0.70 & 0.70 & 2.70 & 2.70 \\ 0.74 & 0.74 & 0.06 & 0.06 \end{pmatrix} \end{array}$$

$p_1 \succ p_2, \quad p_2 \succ p_3, \quad p_1 \succ p_3$

$p_1 \succ p_3$

	s_1	s_2	s_3
fitness:	4	2	0
ranking:	1	2	3

	s_1	s_2	s_3
ranking f_1	2	1	3
ranking f_2	1	3	2
average ranking	1.5	2	2.5

Example: $P = \{s_1, s_2, s_3\}$, $T = \{t_1, \dots, t_4\}$

$$G = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \xrightarrow[rank(G)=2]{NMF \quad k=2} W \times H = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{pmatrix} 0.70 & 2.05 \\ 0.73 & 0.66 \\ 0.35 & 1.02 \end{pmatrix} \end{array} \times \begin{array}{c} \\ f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 0.70 & 0.70 & 2.70 & 2.70 \\ 0.74 & 0.74 & 0.06 & 0.06 \end{pmatrix} \end{array}$$

$p_1 \succ p_2, \quad p_2 \succ p_3, \quad p_1 \succ p_3$

$p_1 \succ p_3$

	s_1	s_2	s_3
fitness:	4	2	0
ranking:	1	2	3

	s_1	s_2	s_3
ranking f_1	2	1	3
ranking f_2	1	3	2
average ranking	1.5	2	2.5

$$G' = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 2 & 1 & 1 & 1 \end{pmatrix} \end{array} \xrightarrow[rank(G)=3]{NMF \quad k=2} W' \times H' = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 2.17 & 1.70 & 2.07 & 2.07 \\ 0.95 & 1.13 & 1.96 & 1.96 \\ 1.88 & 1.17 & 0.97 & 0.97 \end{pmatrix} \end{array}$$

$$= \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{pmatrix} 0.96 & 1.51 \\ 0.39 & 1.84 \\ 0.86 & 0.38 \end{pmatrix} \end{array} \times \begin{array}{c} \\ f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 2.16 & 1.20 & 0.72 & 0.72 \\ 0.05 & 0.35 & 0.90 & 0.90 \end{pmatrix} \end{array}$$

$p_1 \succ p_2, \quad p_1 \succ p_3$

	s_1	s_2	s_3
fitness:	4	2	1
ranking:	1	2	3

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} f'_1 & f'_2 \end{array} \\ \begin{pmatrix} 4.61 & 3.22 \\ 1.87 & 4.05 \\ 4.13 & 0.84 \end{pmatrix} \end{array}$$

	s_1	s_2	s_3
ranking f'_1	1	3	2
ranking f'_2	2	1	3
average ranking	1.5	2	2.5

Experiment

- GP: 'vanila' GP, $|P| = 1000$
- DOF: Derived Objectives via Factorization
  - $\mathrm{DOF\text{-}W}$: derived objectives taken directly from W
  - $\mathrm{DOF\text{-}WH}$: weights features in H by program's factors in W
  - $|P|$ increased $(1 - \alpha)$ times $\implies$ same budget
  - Multiobjective selection: NSGA-II (Deb et al. 2002)
- DOC: Discovery of Objectives by Clustering
- $k \in 2, 3$ $\alpha \in 0.4, 0.6, \ldots, 1.0$

Domain	Instruction set	Problem	Variables	#tests
Boolean	and, nand or, nor	Cmp6	6	64
		Cmp8	8	256
		Par5	5	32
		Mux6	6	64
		Maj6	6	64
Algebras	$a_i(x, y)$	Disc-a1...a5	3	27
	$a_i(x, y)$	Malcev-a1...a5	3	15

Experiment

Average ranks on success rate (Friedman's $p \ll 0.001$)

	DOF		DOC	GP
	W	WH		
All problems	1.97	2.03	2.9	3.10
Boolean	2.4	2.6	3.15	4.4
Categorical	1.90	1.75	2.5	4.25

- Best results for DOF-W and DOF-WH,
- Pays off to derive objectives based on both W and H,
- High compression of G without affecting the performance
  - Interaction outcomes are indeed correlated.
- Low k $\rightarrow$ low computational overhead of factorization
  - only 6% of the total cost of $1,000|T|$ interactions

Conclusions

- A way to multiobjectivize GP,
- Applicable beyond GP:
  - Evolving controllers, two-player games, image analysis, ...
- Works with sparse interaction marices,
- Computationally efficient,
- Promising results for symbolic regression.

Conclusions

- A way to multiobjectivize GP,
- Applicable beyond GP:
  - Evolving controllers, two-player games, image analysis, ...
- Works with sparse interaction marices,
- Computationally efficient,
- Promising results for symbolic regression.

Thank You