Combining Geometric Semantic GP with Gradient-descent optimization

Gloria Pietropolli, Luca Manzoni, Alessia Paoletti, Mauro Castelli

University of Trieste

21/04/2022

Outline

Outline

Introduction

Geometric semantic genetic programming (GSGP) is a well-known variant of genetic programming (GP) where **recombination** and **mutation** operators have a clear semantic effect.

Introduction

Geometric semantic genetic programming (GSGP) is a well-known variant of genetic programming (GP) where **recombination** and **mutation** operators have a clear semantic effect.

Both kind of operators have randomly selected parameters that are **not optimized** by the search process.

Introduction

Geometric semantic genetic programming (GSGP) is a well-known variant of genetic programming (GP) where **recombination** and **mutation** operators have a clear semantic effect.

Both kind of operators have randomly selected parameters that are **not optimized** by the search process.

IDEA : combine GSGP with a well-known gradient-based optimizer, *Adam*, in order to leverage:

- the ability of **GSGP** to operate structural changes of the individuals
- the ability of **gradient-based methods** to optimize the parameters of a given structure.

Outline

Geometric Semantic GP

improve the performance of GP $\Rrightarrow$ integration of **semantic awareness** in the evolutionary process

Geometric Semantic GP

improve the performance of GP ⤞ integration of **semantic awareness** in the evolutionary process

Thus, we can represent a GP individual as a **point** in a real finite-dimensional vector space: the ***Semantic Space***.

Geometric Semantic GP

improve the performance of GP $\ggg\!\!\longrightarrow$ integration of **semantic awareness** in the evolutionary process

Thus, we can represent a GP individual as a **point** in a real finite-dimensional vector space: the ***Semantic Space***.

Geometric Semantic Genetic Programming (GSGP)

GSGP is an evolutionary technique originating from GP that directly searches the semantic space of the programs

Geometric Semantic Operators

Geometric Semantic Operators

Geometric Semantic Crossover (GSC)

Given two parents functions $T_1, T_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, **GSC** generates the real function

$$T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$$

where T_R is a random real function whose output range in the interval $[0, 1]$.

Geometric Semantic Operators

Geometric Semantic Crossover (GSC)

Given two parents functions $T_1, T_2 : \mathbb{R}^n \to \mathbb{R}$, **GSC** generates the real function

$$T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$$

where T_R is a random real function whose output range in the interval $[0, 1]$.

Geometric Semantic Mutation (GSM)

Given a parent function $T : \mathbb{R}^n \to \mathbb{R}$, **GSM** generates the real functions

$$T_M = T + ms \cdot (T_{R1} - T_{R2})$$

where T_{R1} and T_{R2} are random real functions whose output range in the interval $[0, 1]$ and ms is a parameter called mutation step.

The Adam Algorithm

Adam (Adaptive Moment Estimation) is an algorithm for **first-order gradient-based optimization** of stochastic objective functions, based on adaptive estimates of lower-order models.

The Adam Algorithm

Adam (Adaptive Moment Estimation) is an algorithm for **first-order gradient-based optimization** of stochastic objective functions, based on adaptive estimates of lower-order models.

- efficient
- easy to implement
- little memory usage for its execution
- well suited for problems dealing with a vast amount of data and/or parameter

The Adam Algorithm

Algorithm 1 Pseudocode of the *Adam* algorithm.

Require: $f(\theta),\ \theta_0,\ N,\ \alpha,\ \beta_1 \in [0,1),\ \beta_2 \in [0,1),\ \epsilon$

$m_0 \leftarrow 0$

$v_0 \leftarrow 0$

for $i = 0 \cdots N$ **do**

  $d_{i+1} \leftarrow \nabla_\theta f_{i+1}(\theta_i)$ ▷ gradient of stochastic function f

  $m_{i+1} \leftarrow \beta_1 \cdot m_i + (1-\beta_2) \cdot d_{i+1}$ ▷ update first moment estimates

  $v_{i+1} \leftarrow \beta_2 \cdot v_i + (1-\beta_2) \cdot d_{i+1}^2$ ▷ update second moment estimates

  $\bar{m}_{i+1} \leftarrow m_{i+1}/(1-\beta_1^{i+1})$ ▷ contrast intrinsic initialization bias

  $\bar{v}_{i+1} \leftarrow v_{i+1}/(1-\beta_2^{i+1})$ ▷ contrast intrinsic initialization bias

  $\theta_{i+1} \leftarrow \theta_i - \alpha \cdot \bar{m}_{i+1}/(\bar{v}_{i+1})$ ▷ update parameters

end for

WHY?

Why?

Why?

Geometric Semantic GP

- Allows **big jump** on the solutions space
- New areas of the solution space can be explored

Why?

Geometric Semantic GP

- Allows **big jump** on the solutions space
- New areas of the solution space can be explored

Adam Optimizer

- Perform **small shift** in the local area of the solution space

Why?

Geometric Semantic GP

- Allows **big jump** on the solutions space
- New areas of the solution space can be explored

Adam Optimizer

- Perform **small shift** in the local area of the solution space

Good jump in promising areas of the solution space (**GSGP**) and subsequent refinement of the solution (**ADAM**).

HOW?

How?

let's consider an equivalent definition for GSOs:

1. GSM: $T_M = T + ms \cdot (R_1 - R_2)$, where $0 \leq m \leq 1$
2. GSC: $T_{XO} = (T_1 \cdot \alpha) + ((1 - \alpha) \cdot T_2)$, where $0 \leq \alpha \leq 1$

where T, T_1 and T_2 are the parent function; R_1 and R_2 are random real functions whose output range in $[0, 1]$.

How?

let's consider an equivalent definition for GSOs:

1. GSM: $T_M = T + ms \cdot (R_1 - R_2)$, where $0 \leq m \leq 1$
2. GSC: $T_{XO} = (T_1 \cdot \alpha) + ((1 - \alpha) \cdot T_2)$, where $0 \leq \alpha \leq 1$

where T, T_1 and T_2 are the parent function; R_1 and R_2 are random real functions whose output range in $[0, 1]$.

- generate initial random population of functions

How?

let's consider an equivalent definition for GSOs:

1. GSM: $T_M = T + ms \cdot (R_1 - R_2)$, where $0 \leq m \leq 1$
2. GSC: $T_{XO} = (T_1 \cdot \alpha) + ((1 - \alpha) \cdot T_2)$, where $0 \leq \alpha \leq 1$

where T, T_1 and T_2 are the parent function; R_1 and R_2 are random real functions whose output range in $[0, 1]$.

- generate initial random population of functions
- perform **evolutionary steps** (GSM + GSC)

How?

let's consider an equivalent definition for GSOs:

1. GSM: $T_M = T + ms \cdot (R_1 - R_2)$, where $0 \leq m \leq 1$
2. GSC: $T_{XO} = (T_1 \cdot \alpha) + ((1 - \alpha) \cdot T_2)$, where $0 \leq \alpha \leq 1$

where T, T_1 and T_2 are the parent function; R_1 and R_2 are random real functions whose output range in $[0, 1]$.

- generate initial random population of functions
- perform **evolutionary steps** (GSM + GSC)
- obtain $T = (T_1, T_2, \cdots, T_N)$ new population, composed of derivable functions

How?

let's consider an equivalent definition for GSOs:

1. GSM: $T_M = T + ms \cdot (R_1 - R_2)$, where $0 \leq m \leq 1$
2. GSC: $T_{XO} = (T_1 \cdot \alpha) + ((1 - \alpha) \cdot T_2)$, where $0 \leq \alpha \leq 1$

where T, T_1 and T_2 are the parent function; R_1 and R_2 are random real functions whose output range in $[0, 1]$.

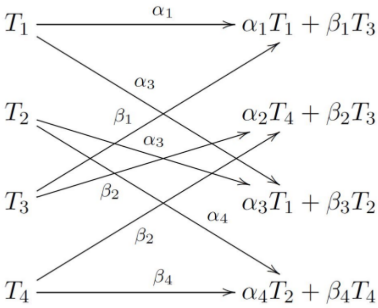
- generate initial random population of functions
- perform **evolutionary steps** (GSM + GSC)
- obtain $T = (T_1, T_2, \cdots, T_N)$ new population, composed of derivable functions
- apply **Adam optimizer**:
  1. objective function $f(\theta) \rightarrow$ the generation considered T
  2. parameter vector $\rightarrow \theta = (\alpha, \beta = (1 - \alpha), ms)$

A practical example

A practical example

- initial random pop T_1, T_2, T_3, T_4
- only crossover operator

1. 4 new individuals created as linear combination of previous generations through parameters α_i and β_i $(i = 1, \ldots, 4)$
2. perform Adam optimization to update parameters α_i and β_i $(i = 1, \ldots, 4)$



WHO?

Who?

Who?

1. **HYB-GSGP**: (*Hybrid Geometric Semantic Genetic Programming*) one step of GSGP is alternated to one step of the Adam optimizer.

Who?

1. **HYB-GSGP**: (*Hybrid Geometric Semantic Genetic Programming*) one step of GSGP is alternated to one step of the Adam optimizer.

2. **HeH-GSGP**: (*Half et Half Geometric Semantic Genetic Programming*) initially, all the GSGP genetic steps are performed, followed by an equal number of Adam optimizer steps.

Outline

Dataset

Have been considered and tested dataset:

- real-world, complex, ranging from different areas
- widely used as benchmarks for GP

Dataset

Have been considered and tested dataset:

- real-world, complex, ranging from different areas
- widely used as benchmarks for GP

Dataset	Variables	Instances	Area	Task
%F	242	359	Pharmacokinetic	Regression
LD50	627	234	Pharmacokinetic	Regression
%PPB	627	131	Pharmacokinetic	Regression
yac	7	308	Physics	Regression
slump	10	102	Physics	Regression
conc	9	1030	Physics	Regression
air	6	1503	Physics	Regression

Experimental Study

Experimental Study

- Results obtained with **HYB-GSGP** and **HeH-GSGP** are compared to classical **GSGP** ones.

Experimental Study

- Results obtained with **HYB-GSGP** and **HeH-GSGP** are compared to classical **GSGP** ones.
- Two hyperparameters settings with different **learning rate** value for the Adam algorithm:
  - 0.01
  - 0.001

Experimental Study

- Results obtained with **HYB-GSGP** and **HeH-GSGP** are compared to classical **GSGP** ones.
- Two hyperparameters settings with different **learning rate** value for the Adam algorithm:
    - 0.01
    - 0.001
- Total number of fitness evaluations must be equal for every method considered (200).

Experimental Study

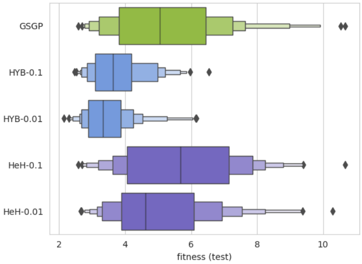
- Results obtained with **HYB-GSGP** and **HeH-GSGP** are compared to classical **GSGP** ones.
- Two hyperparameters settings with different **learning rate** value for the Adam algorithm:
  - 0.01
  - 0.001
- Total number of fitness evaluations must be equal for every method considered (200).
- 100 runs for each problem.

Experimental Study

- Results obtained with **HYB-GSGP** and **HeH-GSGP** are compared to classical **GSGP** ones.
- Two hyperparameters settings with different **learning rate** value for the Adam algorithm:
  - 0.01
  - 0.001
- Total number of fitness evaluations must be equal for every method considered (200).
- 100 runs for each problem.
- $70:30$ training-test partition.

Outline

Slump



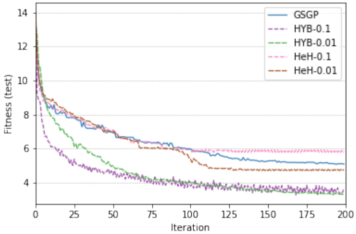
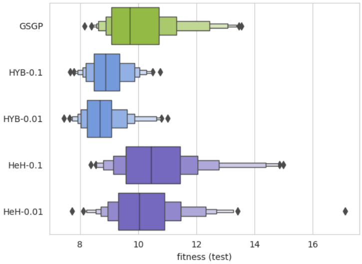


Figure: Boxplots of testing RMSE and median of the fitness over epochs obtained over 100 independent runs for **slump** problems.

Concrete



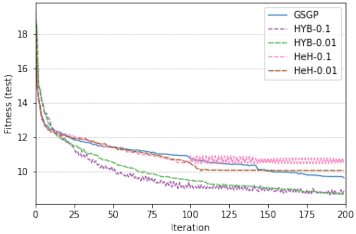
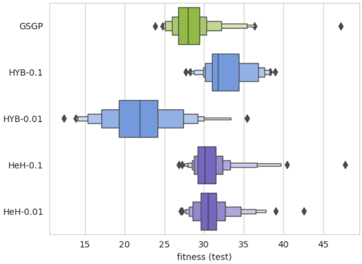


Figure: Boxplots of testing RMSE and median of the fitness over epochs obtained over 100 independent runs for **concrete** problems.

Airfoil



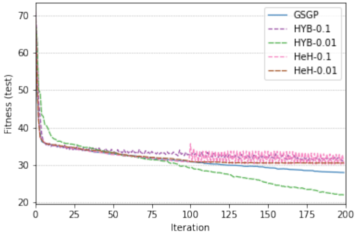


Figure: Boxplots of testing RMSE and median of the fitness over epochs obtained over 100 independent runs for **airfoil** problems.

Results

		GSGP	HYB-0.1	HYB-0.01	HeH-0.1	HeH-0.01
%F	Train	38.08	37.74	**36.80**	39.61	40.60
	Test	40.15	40.48	**39.61**	40.85	41.23
LD50	Train	2118.00	**2086.56**	2128.22	2144.27	2161.00
	Test	2214.78	**2203.25**	2229.87	2221.72	2215.09
%PPB	Train	30.15	27.00	**24.32**	34.79	33.26
	Test	328.1	401.43	263.81	**213.86**	235.53
yac	Train	**11.83**	11.92	12.48	12.28	12.31
	Test	11.92	**11.83**	12.52	12.38	12.48
slump	Train	4.56	3.47	**2.92**	5.19	4.41
	Test	5.08	3.63	**3.32**	5.77	4.76
conc	Train	9.62	8.86	**8.50**	10.59	10.05
	Test	9.65	8.88	**8.69**	10.47	10.07
air	Train	27.76	31.54	**21.98**	30.37	30.46
	Test	27.94	31.71	**21.97**	30.15	30.53

Table: Training and testing fitness (RMSE) for the considered benchmark problems. **Bold** font indicates the best results.

Outline

Conclusion

Combination of GSGP and Adam optimizer can improve the performance of GSGP.

- **HYB-GSGP outperforms classic GSGP** in both training and test sets.

Conclusion

Combination of GSGP and Adam optimizer can improve the performance of GSGP.

- **HYB-GSGP outperforms classic GSGP** in both training and test sets.
- **HYB-GSGP converges** to good-quality solutions **faster** than classical GSGP.

Conclusion

Combination of GSGP and Adam optimizer can improve the performance of GSGP.

- **HYB-GSGP outperforms classic GSGP** in both training and test sets.
- **HYB-GSGP converges** to good-quality solutions **faster** than classical GSGP.
- **HeH-GSGP does not outperform GSGP** even if it generally ensures good quality results on the test set.

Conclusion

Combination of GSGP and Adam optimizer can improve the performance of GSGP.

- **HYB-GSGP outperforms classic GSGP** in both training and test sets.
- **HYB-GSGP converges** to good-quality solutions **faster** than classical GSGP.
- **HeH-GSGP does not outperform GSGP** even if it generally ensures good quality results on the test set.

Future work:

- to consider a different kind of crossover where more than 2 parents are involved to get a structure more similar to a neural network one.

Conclusion

Combination of GSGP and Adam optimizer can improve the performance of GSGP.

- **HYB-GSGP outperforms classic GSGP** in both training and test sets.
- **HYB-GSGP converges** to good-quality solutions **faster** than classical GSGP.
- **HeH-GSGP does not outperform GSGP** even if it generally ensures good quality results on the test set.

Future work:

- to consider a different kind of crossover where more than 2 parents are involved to get a structure more similar to a neural network one.
- to test this technique considering other optimizer.

thanks for your attention!!