NOVA IMS Information Management School

Neural and Evolutionary Learning

Class 2 - Genetic Programming

Prof.: Karina Brotto Rebuli

krebuli@novaims.unl.pt

Genetic Programming

	GA	*Tree-based GP
Genome	Constant length	Lisp-like tree
Task type	Optimization	Many tasks, including ML
Population initialization	Random values	**Random trees
Crossover	***“Blind” genotype variation	***“Blind” genotype variation
Mutation	***“Blind” genotype variation	***“Blind” genotype variation

* We will work woth tree-based GP, but GP solutions can have different structures, like grammars, ML pipelines. etc.

** Although many initialization methods exist, tree generation in the initial population involves inherent randomness.

*** Some advanced genetic operators account for additional metrics (solutions and populations).

Genetic Programming

- Important issues:
  - Bloat → Comparing solutions size and fitness evolution.
  - Premature convergence → Comparing diversity evolution.
  - Overfitting → Comparing train and test fitness.

How to track and overcome them?

Genetic Programming Implementations

- SRBench benchmarked methods and implementations (not all use Python):
  - Age-Fitness Pareto Optimization (Schmidt and Lipson 2009)
  - Age-Fitness Pareto Optimization with Co-evolved Fitness Predictors (Schmidt and Lipson 2009)
  - AIFeynman 2.0 (Udrescu et al. 2020)
  - Bayesian Symbolic Regression (Jin et al. 2020)
  - Deep Symbolic Regression (Petersen et al. 2020)
  - Fast Function Extraction (McConaghy 2011)
  - Feature Engineering Automation Tool (La Cava et al. 2017)
  - epsilon-Lexicase Selection (La Cava et al. 2016)
  - GP-based Gene-pool Optimal Mixing Evolutionary Algorithm (Virgolin et al. 2017)
  - gplearn (Stephens)
  - Interaction-Transformation Evolutionary Algorithm (de Franca and Aldeia, 2020)
  - Multiple Regression GP (Arnaldo et al. 2014)
  - Operon (Burlacu et al. 2020)
  - Semantic Backpropagation GP
- DEAP (Distributed Evolutionary Algorithms in Python - https://deap.readthedocs.io/en/master/) is also an interesting alternative.

slim_gsgp NOVA IMS library

- It is based on PyTorch tensor objects, which is a very robust and flexible framework.
- slim_gsgp includes:
  - standard Genetic Programming (GP);
  - standard Geometric Semantic Genetic Programming (GSGP);
  - all existing SLIM-GSGP variants, facilitating comparative analysis and benchmarking.
- Source code: https://github.com/DALabNOVA/slim.
- Manual: https://slim-library.readthedocs.io/en/latest/.
- Paper: Rosenfeld et al. 2025 Slim_gsgp: A Python Library for Non-Bloating GSGP, GECCO 2025.

slim_gsgp NOVA IMS library

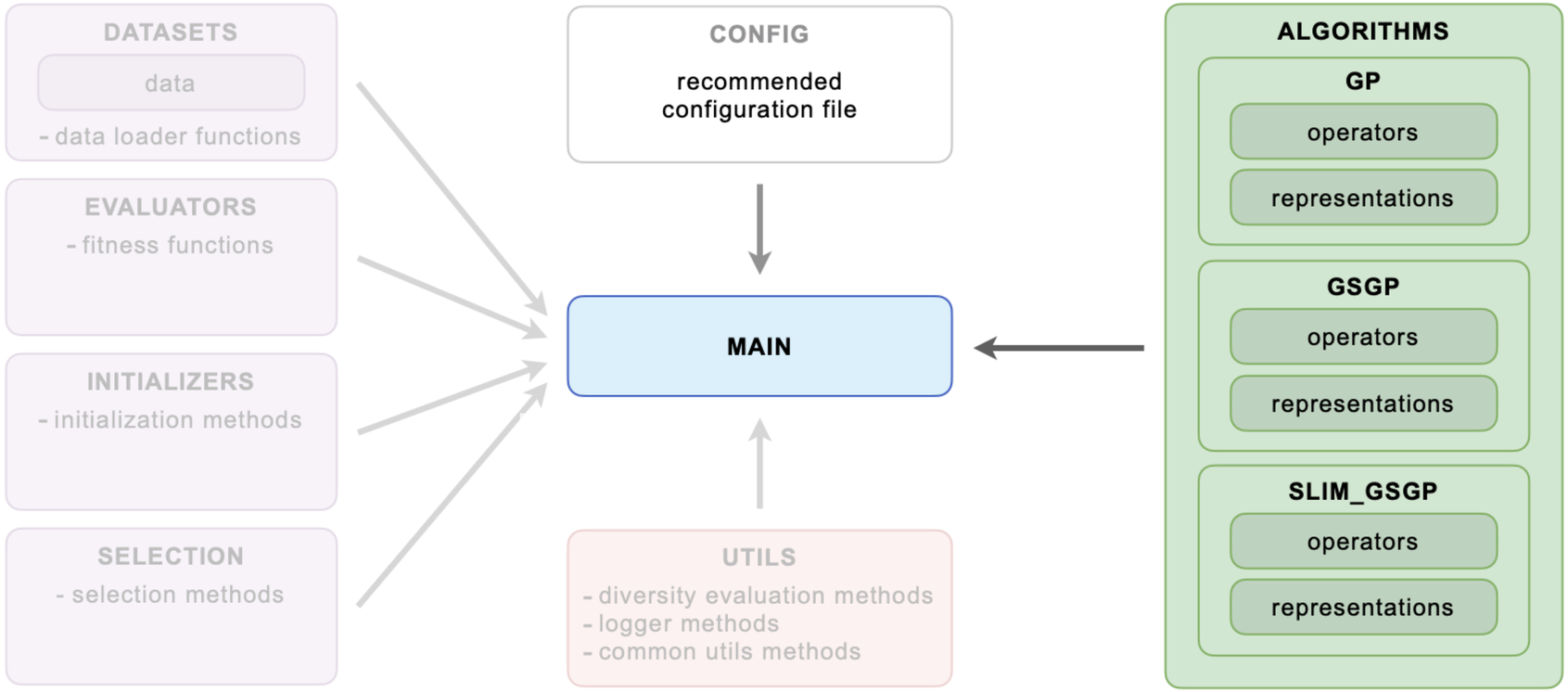

Figure 01. Overview of the slim_gsgp framework. Source: the author.

slim_gsgp NOVA IMS library

- Let's take a look at the codes.

NOVA IMS Information Management School

Questions?

https://forms.gle/EV9VkExNtfNckMSM8

Register your feedback