A Study of Semantic Geometric Crossover Operators in Regression Problems

Julio Albinati, Gisele L. Pappa, Fernando E. B. Otero, Luiz Otávio V. B. Oliveira

In traditional genetic programming, operators used for crossover and mutation are typically *syntactic*, in the sense that they operate in a representation-level, ignoring the semantic impact they cause. Considering we are mostly interested in generating semantically useful solutions, there has been an increased interest in semantic-aware operators in GP. However, it was only in [3] that operators working *directly* in the semantics of solutions were proposed and they have been successfully applied in several scenarios [2, 4] when compared to traditional GP.

For regression problems, the semantic crossover operator proposed in [3] was defined as convex combination. Given two previously generated functions f_1 and f_2, a new function f_3 is created as

$$f_3(x) = c(x) \cdot f_1(x) + (1 - c(x)) \cdot f_2(x) \ \ , \tag{1}$$

where $c(x)$ is a randomly generated expression with outputs in the interval $[0, 1]$. If $c(x)$ is a constant in relation to x, it corresponds to the Euclidean based semantic geometric crossover operator; if $c(x)$ is a random function, it corresponds to the Manhattan based semantic geometric crossover operator. A function created this way has the property of being semantically intermediate between its parents (characterizing the operator as *geometric*). A direct consequence of this fact is that the new solution is not worse than the worst of its parents. More interestingly, this property holds even when considering test data. While in [3], the experiments were carried out using the Euclidean based operator, more recent works have used the Manhattan based operator [1]. In this paper we study the difference in performance of these operators, including two new variations of the Euclidean based operator, which analytically determine the constant to be used by the crossover operator.

In the particular case of semantic crossover between two expressions, f_1 and f_2, where $c(x)$ is replaced by a constant value c in the interval $[0, 1]$ (Euclidean based crossover), one can define the error obtained by f_3 as a function of the coefficient used in the convex combination. When considering the root mean squared error in relation to the desired output, we obtain

$$RMSE(c) = \sum_{(x_i, y_i) \in \mathcal{T}} [y_i - c \cdot f_1(x_i) - (1 - c) \cdot f_2(x_i)]^2 \ \ , \tag{2}$$

where $\mathcal{T}$, f_1 and f_2 are the training set consisting in (x_i, y_i) pairs—where x_i is the input and y_i is the desired output—and two previously generated functions, respectively. Since Equation (2) is continuous, one can derive it with respect to the coefficient c and equal the derivative to zero, finding the value c^* that minimizes the equation. This strategy leads to

$$c^* = \frac{\sum\limits_{(x_i, y_i) \in \mathcal{T}} [y_i - f_2(x_i)] \cdot [f_2(x_i) - f_1(x_i)]}{[f_1(x_i) - f_2(x_i)]^2} \ \ . \tag{3}$$

A function generated using such optimized coefficient is guaranteed to be not worse than the best of its parents in the training set, strengthening the original properties of the Euclidean based semantic geometric crossover operator. To understand why this property is valid, we must remember that this optimization is over all possible convex combination, including those that uses $c = 1$ or $c = 0$. Therefore, if Equation (3) returns a value between 0 and 1, the function generated through the convex combination will be better than its parents; otherwise, it will be simply replicate the best of its parents. A similar strategy can lead to the definition of a semantic geometric crossover operator that uses linear combination (not necessarily convex) and still generates solutions that are better than its parents considering the training set—this allows a larger number of combinations and, at the same time, includes every combination allowed by the operator using convex combinations.

In order to assess the impact of operators proposed, we evaluated four algorithms: a Koza-style standard GP, a semantic GP (SGP) using the Euclidean based semantic geometric operator, a semantic GP using the Euclidean based semantic geometric operator with optimised coefficients given by Equation 3 (SGP-O), a semantic GP using optimized linear combinations as crossover (SGP-L) and a semantic GP using the Manhattan based semantic operator (SGP-M). The Manhattan based operator used a random generated function r and used a sigmoid function $s(x) = (1 + e^{-r(x)})^{-1}$ to constrain the output to the interval $[0, 1]$ as in [1]. Table 1 shows the median RMSE obtained in the test data by each version of SGP in four datasets after 30 executions. Each implementation had a population size of 1000 individuals and evolved 2000 generations; mutation rate and step were set to 0.4 and 0.01 for SGP, 0.1 and 0.1 for SGP-O and SGP-M, 0.4 and 0.001 for SGP-L. In terms of the RMSE achieved in the test data, our experiments show that it is possible to improve the performance of the SGP by using optimised constants in the semantic geometric crossover. There is not much difference between SGP-O and SGP-L, therefore, it is not clear if the linear combination has any impact in the performance of the algorithm. However, the use of the Manhattan based operator in the SGP-M has clear advantages over Euclidean ones, the only exception is the keijzer-6 dataset. These results suggest that the Manhattan based semantic geometric crossover operator is more suitable for regression domains.

Table 1: Results for all algorithms in four datasets: ▲ indicates that the method on the column is statistically better than SGP, while ▽ indicates that the method is statistically worse.

Dataset	**SGP**	**SGP-O**	**SGP-L**	**SGP-M**
bioavailability	95.737	114.505 ▽	110.555 ▽	36.720 ▲
keijzer-5	0.243	0.214	0.303 ▽	0.142 ▲
keijzer-6	1.823	0.642 ▲	0.019 ▲	0.332 ▲
tower	47.123	38.518 ▲	38.268 ▲	23.670 ▲

References

[1] Mauro Castelli, Sara Silva, and Leonardo Vanneschi. A C++ framework for geometric semantic genetic programming. *Genetic Programming and Evolvable Machines*, pages 1–9, 2014.

[2] Mauro Castelli, Leonardo Vanneschi, and Sara Silva. Prediction of high performance concrete strength using genetic programming with geometric semantic genetic operators. *Expert Systems with Applications*, 40(17):6856–6862, 2013.

[3] Alberto Moraglio, Krzysztof Krawiec, and Colin G Johnson. Geometric semantic genetic programming. In *Parallel Problem Solving from Nature-PPSN XII*, pages 21–31. Springer, 2012.

[4] Leonardo Vanneschi, Sara Silva, Mauro Castelli, and Luca Manzoni. Geometric semantic genetic programming for real life applications. In *Genetic Programming Theory and Practice XI*, pages 191–209. Springer, 2014.