Land Cover/Land Use Multiclass Classification Using GP with Geometric Semantic Operators

Mauro Castelli[1,2], Sara Silva[1,3], Leonardo Vanneschi[2,1], Ana Cabral[4], Maria J. Vasconcelos[4], Luís Catarino[4,5], and João M.B. Carreiras[4]

[1] INESC-ID, IST, Universidade Técnica de Lisboa, 1000-029 Lisboa, Portugal
[2] ISEGI, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal
[3] CISUC, Universidade de Coimbra, 3030-290 Coimbra, Portugal
[4] Instituto de Investigação Científica Tropical, 1300-344 Lisboa, Portugal
[5] CIBIO, Universidade do Porto, 4485-661 Vairão, Portugal
sara@kdbio.inesc-id.pt

Abstract. Multiclass classification is a common requirement of many land cover/land use applications, one of the pillars of land science studies. Even though genetic programming has been applied with success to a large number of applications, it is not particularly suited for multiclass classification, thus limiting its use on such studies. In this paper we take a step forward towards filling this gap, investigating the performance of recently defined geometric semantic operators on two land cover/land use multiclass classification problems and also on a benchmark problem. Our results clearly indicate that genetic programming using the new geometric semantic operators outperforms standard genetic programming for all the studied problems, both on training and test data.

1 Introduction

A new integrated land science, joining environmental, human, and remote sensing sciences, is emerging. It addresses questions about the impacts of land use and land cover changes, both on the environment and on the livelihoods of people [8]. This recently developed discipline led to the development of a large amount of case studies and data sets, with a corresponding plethora of methodologies for analysis. However, the complexity of causes, processes and impacts of land change has, so far, impeded the development of an integrated theory. Land science studies require versatile data analysis tools that can solve multi-type pattern identification problems, ranging for instance from classification of satellite images into several land cover type classes in a map, to identifying land cover transition patterns in multi-temporal map data sets, or to prediction of pattern evolution through time.

Genetic Programming (GP) is the automated learning of computer programs, using Darwinian selection and Mendelian genetics as sources of inspiration [5]. In the last decade, GP has been extensively used both in Industry and Academia and it has produced a wide set of results that have been characterized as *human-competitive* [6]. Although in principle GP has the potential to evolve any kind of

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2013, LNCS 7835, pp. 334–343, 2013.

solution, including decision trees, it has never been particularly suited for multiclass classification problems. This inadequacy, in many cases, derives not from limitations of the algorithm, but from the particular representation used for the solutions. The interested reader is referred to [13] for discussions on the difficulties of GP in facing multiclass classification problems and to [3] for a complete review of the state of the art methods in using GP for classification problems. It is a common procedure to address two-class classification problems using GP as regression ones, applying a cutoff to the predicted output. This approach can also be used for multiclass classification, but the approach in general becomes less effective as a larger number of classes is considered, and the performance of GP degrades to the point where other machine learning techniques become the only reasonable option. Therefore, even though GP has the potential to address the complexity of land science studies, the "simple" task of land use/land cover multiclass classification represents a potential obstacle.

Research in GP has recently focused on an aspect that was only marginally considered up to some years ago: the definition of methods based on the semantics of the solutions (see for instance [1,9]), where by semantics we generally mean the behavior of a program once it is executed on a set of data or, more specifically, the set of outputs a program produces on the training data. Using this definition of semantics (which is also the one that we adopt here), Moraglio *et al.* have recently defined new genetic operators, called geometric semantic genetic operators [10]. They have a number of theoretical advantages compared to the ones of standard GP; in particular, as proven in [10], they induce a unimodal fitness landscape on any problem consisting in finding the match between a set of input data and a set of known outputs (like for instance classification and regression). This should facilitate evolvability [4], making these problems potentially easier to solve for GP. However, they have a major drawback that makes them unusable in practice: they always create offspring that are larger than their parents, causing an exponential growth of the code in the population. We have proposed a new and very efficient implementation of the geometric semantic operators [12]. This new GP system evolves the semantics of the individuals without explicitly building their syntax, freeing us from dealing with exponentially growing trees and thus allowing us to test, for the first time, the potentiality of the semantic operators on complex real-life problems.

In this paper we want to assess how much improvement the geometric semantic operators introduce when compared to standard GP operators, in particular when dealing with multiclass classification problems in a 'regression and cutoff' manner. We tackle three problems: two real-life land cover/land use applications of four and ten classes, and a well-known benchmark of three classes.

The paper is organized as follows: Section 2 describes the geometric semantic operators of Moraglio *et al.* used in this work. Section 3 presents the experimental study, describing the test problems and settings, and discussing the results obtained. Section 4 concludes and describes our intended future work.

2 Geometric Semantic Operators

Many semantically aware methods presented so far [1,9] are indirect: search operators act on the syntax of the parents to produce offspring that are only accepted if some semantic criterion is satisfied. As reported by Moraglio *et al.* [10], this has at least two drawbacks: (i) these implementations are very wasteful as heavily based on trial-and-error; (ii) they do not provide insights on how syntactic and semantic searches relate to each other. To overcome these drawbacks, Moraglio *et al.* introduced new operators that directly search the semantic space.

To explain the idea, we first provide an example using Genetic Algorithms (GAs). Let us consider a GA problem in which the target solution is known and the fitness of each individual corresponds to its distance to the target (our reasoning holds for any distance measure used). This problem is characterized by a very good evolvability and it is in general easy to solve for GAs. In fact, for instance, if we use point mutation, any possible individual different from the global optimum has at least one neighbor (individual resulting from its mutation) that is closer to the target than itself, and thus is fitter. So, there are no local optima. In other words, the fitness landscape is unimodal. Similar considerations hold for box mutation and for many types of crossover, including various kinds of geometric crossover [7].

Now, let us consider the typical GP problem of finding a function that maps sets of input data into known target outputs (regression and classification are particular cases). The fitness of an individual for this problem is typically a distance between its predicted output values and the expected ones (error measure). Now let us assume that we are able to find a transformation on the syntax of an individual whose effect is just a random perturbation of one of its predicted output values. In other words, let us assume that we are able to transform an individual G into an individual H whose output values are like the outputs of G, except for one value, that is randomly perturbed. Under this hypothesis, we are able to map the considered GP problem into the GA problem discussed above, in which point mutation is used. So, this transformation, if known, would induce a unimodal fitness landscape on every problem like the considered one (e.g. regressions and classifications), allowing GP to have a good evolvability, at least on training data. The same also holds for transformations that correspond to box mutation or semantic crossovers. Although not without limitations, the work of Moraglio *et al.* [10] accomplishes this task, defining the following operators.

Definition 1. (Geometric Semantic Crossover). *Given two parent functions* $T_1, T_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, *the geometric semantic crossover returns 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 values range in the interval* $[0, 1]$.

The interested reader is referred to [10] for a formal proof of the fact that this operator corresponds to a geometric crossover on the semantic space, in the sense that it produces an offspring that stands between its parents in this space. We do not report the proof here, but we limit ourselves to remark that, even without a formal proof, we can have an intuition of it considering that the (only) offspring

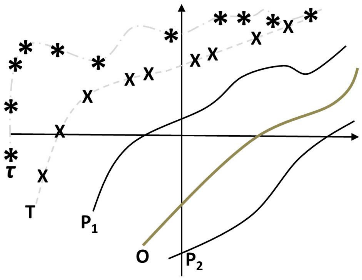

Fig. 1. Visual intuition of the fact that geometric semantic crossover creates an offspring that is at least not worse than the worst of its parents. In this toy example, offspring O (which stands between parents P_1 and P_2 in the semantic space by construction) is clearly closer to target T (training points represented by "×" symbols) than parent P_2. In Section 3 we also discuss the geometric properties of this operator on test data, represented by τ (test points represented by "$*$" symbols).

generated by this crossover has a semantic vector that is a linear combination of the semantics of the parents with random coefficients included in $[0, 1]$ and whose sum is equal to 1. Moraglio *et al.* [10] also prove an interesting consequence of this fact: the fitness of the offspring cannot be worse than the fitness of the worst of its parents. Also in this case, we do not replicate the proof here, but we limit ourselves to providing a visual intuition of this property: in Figure 1 we represent a simple two-dimensional semantic space in which we draw a target function T (training points are represented by "×" symbols), two parents P_1 and P_2 and one of their offspring O (which by construction stands between its parents), plus a test set (composed by test points represented by "$*$" symbols) that will be discussed in the final part of Section 3. It is immediately apparent from Figure 1 that O is closer to T than P_2 (which is the worst parent in this case). The generality of this property is proven in [10]. To constrain T_R in producing values in $[0, 1]$ we use the sigmoid function: $T_R = \frac{1}{1+e^{-T_{rand}}}$ where T_{rand} is a random tree with no constraints on the output values.

Definition 2. (Geometric Semantic Mutation). *Given a parent function* $T : \mathbb{R}^n \to \mathbb{R}$, *the geometric semantic mutation with mutation step ms returns the real function* $T_M = T + ms \cdot (T_{R1} - T_{R2})$, *where* T_{R1} *and* T_{R2} *are random real functions.*

Moraglio *et al.* [10] prove that this operator corresponds to a box mutation on the semantic space, and induces a unimodal fitness landscape. Even without a formal proof it is not difficult to have an intuition of it, considering that each element of the semantic vector of the offspring is a "weak" perturbation of the corresponding element in the parent's semantics. We informally define this perturbation as "weak" because it is given by a random expression centered in zero (the difference between two random trees). Nevertheless, by changing parameter ms, we are able to tune the "step" of this perturbation, and its importance.

3 Experimental Study

Test Problems. Two land cover/land use applications and a well-known benchmark have been used as test problems.

LANDMAP: Land cover mapping. The objective of this application is mapping land use/land cover types of Guinea-Bissau as function of six different metrics extracted from Landsat TM and ETM+ data for 2010. Mapping land use/land cover is one of the foremost requirements for planning, management and conservation of land and forest. This study considers 10 land cover types, among which three forest types (closed forest, open forest, savanna woodland) on a total of 6798 instances. Distinguishing between different forest types is a challenging task given the spectral similarity between them.

CASHEW: Cashew in West Africa. West Africa has one of the most modified tropical forest landscapes in the world, where tree cover is often part of a forest-savanna agriculture mosaic [11]. The objective of this application is to discriminate different land cover classes occurring in a forest and agriculture mosaic from 12 different metrics obtained from the RapidEye or Landsat Thematic Mapper data over Guinea-Bissau. The original data set contains 10 classes, on a total of 370 instances. However, as a preliminary study we used only four of these classes, on a total of 221 instances.

IRIS: Flower classification. This is a well-known benchmark problem available at the UCI Machine Learning Repository. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant and each instance is described by four attributes.

Experimental Setting. We tackle each of the test problems with the two different GP systems: standard GP (ST-GP) and GP that uses the geometric semantic operators described in Section 2 (GS-GP). In all cases GP is used as if we were dealing with regression problems, i.e. the numeric class label is interpreted as the expected output value of the function to be learned.

For each of the GP systems, 50 independent runs have been performed with a population of 200 individuals. For each run, different randomly generated partitions of the data sets into training (70%) and test (30%) sets where used. The evolution stopped after 10000 fitness evaluations for both GP variants. Tree initialization was performed with the Ramped Half-and-Half method [5] with a maximum initial depth of 6. The function set contained the four arithmetic operators $+$, $-$, $*$, and $/$ protected as in [5]. For each studied problem, the terminal set contained a number of variables equal to the number of features in the data set. Fitness was measured as the Root Mean Square Error (RMSE) between predicted and expected outputs, and tournament selection was used with tournament size of 4. The reproduction (replication) rate was 0.1, meaning that each selected parent has a 10% chance of being copied to the next generation instead of being engaged in breeding. ST-GP used standard subtree mutation and crossover (with uniform selection of crossover and mutation points among different tree levels), with probabilities 0.1 and 0.9 respectively. The new random branch created for mutation has maximum depth 6. Selection for survival was elitist, guaranteeing the survival of the best individual from one generation to the next. No maximum tree depth was imposed. GS-GP used a higher mutation rate of 0.5, which was found to be necessary in order for GS-GP to properly explore the search space. The mutation step ms was 0.001.

Experimental Results. We compare the results of GS-GP and ST-GP obtained on training data and, in order to compare the generalization ability of the two methods, on out-of-sample test data. We report the RMSE on the training data, and the accuracy, expressed as the proportion of correctly classified samples, on the test data. In order to calculate accuracy, each predicted output is rounded to its nearest integer value, which represents the class label.

On Figure 2, the plots on the left report, for each studied problem, the evolution of the mean RMSE of the best individual on the training set over the 50 runs. They clearly show that GS-GP reaches the lowest RMSE on all the considered test problems. The boxplots on the right report the RMSE of the best individual on the training set at the end of each run. It can be observed that GS-GP produces solutions with a lower dispersion of RMSE than ST-GP on both LANDMAP and CASHEW problems. To analyze the statistical significance of these results, a set of tests has been performed on the RMSE values. As a first step, the Kolmogorov-Smirnov test has shown that the data are not normally distributed and hence a rank-based statistic has been used. More precisely, we have used the Mann-Whitney test [2], considering a confidence of 95% with a Bonferroni correction. According to this test, the results produced by GS-GP are statistically different from the ones produced by ST-GP on all the considered test problems; the respective p-values are reported in Table 1.

On Figure 3, the plots on the left report, for each studied problem, the evolution of the mean accuracy on the test set of the best individual on the training set over the 50 runs. Also in this case it is clear that GS-GP reaches higher accuracy, i.e. generalizes better, than ST-GP. The boxplots on the right report the accuracy on the test data of the best individual at the end of each run. Also here GS-GP produces solutions with a lower dispersion of RMSE than ST-GP. According to the Mann-Whitney test, the results produced by GS-GP are statistically different from the ones produced by ST-GP on all the considered test problems, as reported in Table 1.

Table 2 summarizes the results obtained on the different studied problems with both GP variants, in terms of minimum, maximum, median, mean and standard deviation of both RMSE on the training set and accuracy on the test set.

Discussion. From the above results we realize that neither ST-GP nor GS-GP overfit, since the accuracy values on the test set do not degrade during the evolution. However, GS-GP has obtained much better results than ST-GP in both training and test data. The good results on the training data were expected: the geometric semantic operators induce an unimodal fitness landscape, which facilitates evolvability. However, this could have caused a loss of generalization

Table 1. The p-values obtained comparing the median fitness (RMSE on the training set and accuracy on the test set) of GS-GP and ST-GP, using the Mann-Whitney statistical test.

	LANDMAP	**CASHEW**	**IRIS**
TRAINING	7.066e-018	1.914e-009	6.0178e-018
TEST	2.194e-017	4.778e-011	5.248e-018

Table 2. Summary of the results obtained on 50 independent runs. Training fitness is the RMSE (optimal fitness 0), while testing fitness is the classification accuracy (optimal fitness 1).

LANDMAP					
	RMSE on TRAINING				
	Min	**Max**	**Median**	**Mean**	**Std Dev**
GS-GP	1.012	1.141	1.074	1.073	0.032
ST-GP	1.147	1.568	1.379	1.350	0.101
	ACCURACY on TEST				
	Min	**Max**	**Median**	**Mean**	**Std Dev**
GS-GP	0.612	0.743	0.696	0.690	0.033
ST-GP	0.337	0.645	0.518	0.511	0.082

CASHEW					
	RMSE on TRAINING				
	Min	**Max**	**Median**	**Mean**	**Std Dev**
GS-GP	0.839	0.946	0.913	0.906	0.024
ST-GP	0.879	1.260	1.211	1.136	0.142
	ACCURACY on TEST				
	Min	**Max**	**Median**	**Mean**	**Std Dev**
GS-GP	0.239	0.388	0.313	0.317	0.034
ST-GP	0.194	0.328	0.224	0.243	0.045

IRIS					
	RMSE on TRAINING				
	Min	**Max**	**Median**	**Mean**	**Std Dev**
GS-GP	0.096	0.176	0.143	0.145	0.016
ST-GP	0.380	0.494	0.445	0.444	0.030
	ACCURACY on TEST				
	Min	**Max**	**Median**	**Mean**	**Std Dev**
GS-GP	0.860	0.980	0.940	0.937	0.025
ST-GP	0.680	0.880	0.720	0.760	0.067

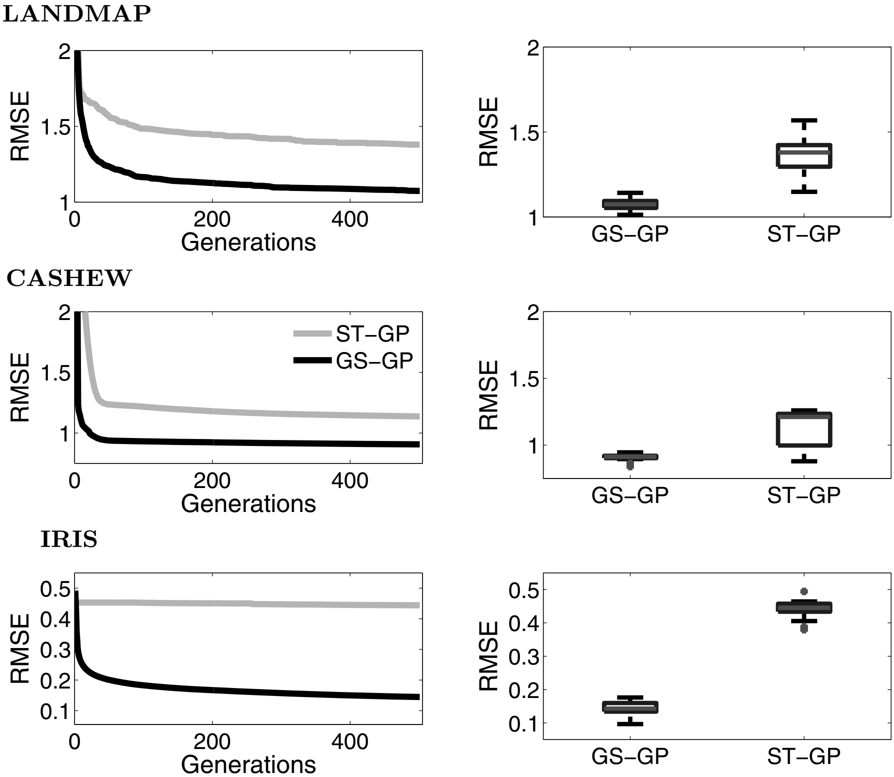

Fig. 2. Results obtained on the *training* set. Plots on the left: evolution of best fitness (RMSE), mean of 50 runs. Boxplots on the right: best fitness (RMSE) at the end of each run.

ability on the test data - it did not. Not so obvious at first sight, the geometric properties of the semantic operators hold *independently from the data* on which individuals are evaluated. In other words, geometric semantic crossover produces an offspring that stands between the parents also in the semantic space induced by test data. As a direct implication, following exactly the same argument as Moraglio *et al.* [10], each offspring is, in the worst case, not worse than the worst of its parents on the test set. This can be seen by looking back at Figure 1, where a simple test set τ is drawn (test points are represented by "$*$" symbols). Analogously, geometric semantic mutation produces an offspring that is a "weak" perturbation of its parent also in the semantic space induced by test data. This has an important consequence on the behavior of GS-GP on test data: even though the geometric semantic operators do not guarantee an improvement of test fitness each time they are applied (e.g. there is a very slight overfitting observed on the IRIS dataset with GS-GP, Figure 3), they at least guarantee that the possible worsening of the test fitness is "limited" (by the test fitness of the worst parent for crossover, and by the mutation step ms for mutation) [12].

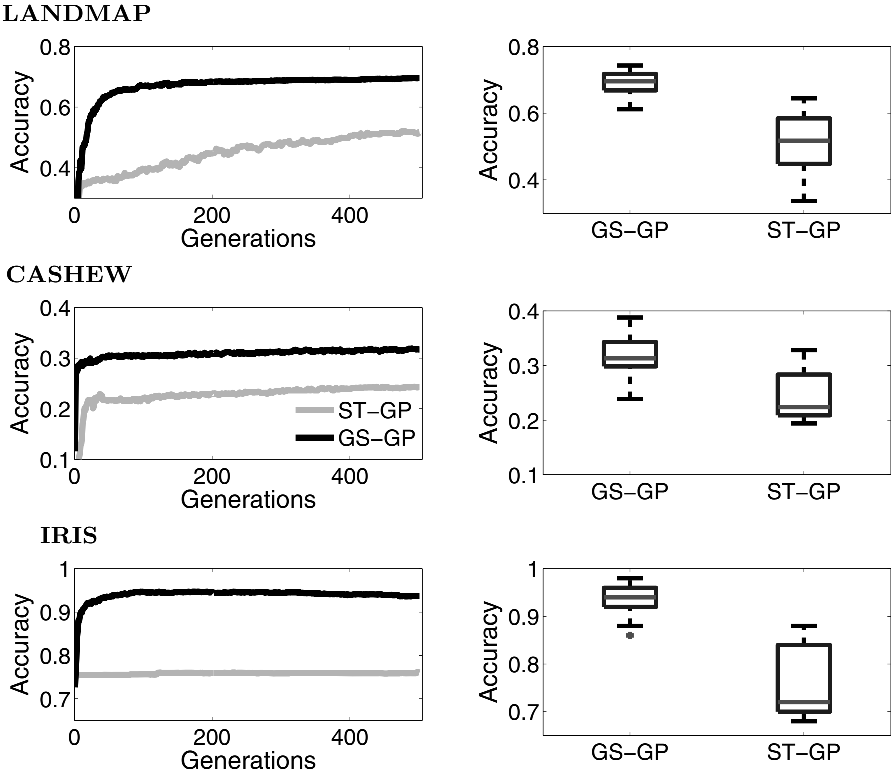

Fig. 3. Results obtained on the *test* set. Plots on the left: evolution of the accuracy on the test set of the best individual on the training set, mean of 50 runs. Boxplots on the right: accuracy on the test set at the end of each run.

4 Conclusions and Future Work

Multiclass classification is a common requirement of many land cover/land use applications, one of the pillars of land science studies. However, as reported in the literature, Genetic Programming (GP) is not particularly suited for this task. We have investigated the use of recently defined geometric semantic operators on multiclass classification problems, using two land cover/land use real-life applications, and one well-known benchmark, as test problems. Our results clearly indicate that GP that uses the geometric semantic operators (GS-GP) outperforms standard GP on all the studied problems. GS-GP returned much better results on training data without loss of generalization on test data. As future work we intend to qualitatively interpret the results achieved by GS-GP from the point of view of the applications, and compare its performance with other machine learning techniques using these and other multiclass classification problems, in order to determine if GS-GP is a competitive method for solving multiclass classification problems.

Acknowledgments. This work was partially supported by national funds through FCT under contract Pest-OE/EEI/LA0021/2011. The authors acknowledge projects "EnviGP - Improving Genetic Programming for the Environment and Other Applications" (PTDC/EIA-CCO/103363/2008), "Cashew in West Africa: socio-economic and environmental challenges of an expanding cash crop" (PTDC/AFR/117785/2010) and "MassGP - Improving Semantic Genetic Programming for Maritime Safety, Security and Environmental Protection" (PTDC/EEI-CTP/2975/2012) funded by the FCT, Portugal, and also project CarboVeg GB (funded by the Ministry of Environment, Portugal) and the Secretary of State of the Environment and Sustainable Development (SEAD, Guinea-Bissau).

References

1. Beadle, L., Johnson, C.G.: Semantic analysis of program initialisation in genetic programming. Genetic Programming and Evolvable Machines 10(3), 307–337 (2009)
2. Corder, G., Foreman, D.: Nonparametric statistics for Non-Statisticians. Wiley, New York (2009)
3. Espejo, P., Ventura, S., Herrera, F.: A survey on the application of genetic programming to classification. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40(2), 121–144 (2010)
4. Gustafson, S., Vanneschi, L.: Crossover-based tree distance in genetic programming. IEEE Transactions on Evolutionary Computation 12(4), 506–524 (2008)
5. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
6. Koza, J.R.: Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines 11(3/4), 251–284 (2010); Tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines
7. Krawiec, K., Lichocki, P.: Approximating geometric crossover in semantic space. In: GECCO 2009: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 987–994. ACM (2009)
8. Lambin, E.R., Geist, H., Rindfuss, R.R.: Local processes with global impacts. In: Lambin, E.R., Geist, H. (eds.) Land Use and Land Cover Change, pp. 1–8. Springer (2006)
9. McPhee, N.F., Ohs, B., Hutchison, T.: Semantic Building Blocks in Genetic Programming. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 134–145. Springer, Heidelberg (2008)
10. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric Semantic Genetic Programming. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 21–31. Springer, Heidelberg (2012)
11. Norris, K., Asase, A., Collen, B., Gockowksi, J., Mason, J., Phalan, B., Wade, A.: Biodiversity in a forest-agriculture mosaic – the changing face of west african rainforests. Biological Conservation 143, 2341–2350 (2010)
12. Vanneschi, L., Castelli, M., Manzoni, L., Silva, S.: A new implementation of geometric semantic GP applied to predicting pharmacokinetic parameters. In: Proceedings of EuroGP-2013. Springer (to appear, 2013)
13. Zhang, M., Smart, W.: Multiclass Object Classification Using Genetic Programming. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 369–378. Springer, Heidelberg (2004)