Prediction of Forest Aboveground Biomass: An Exercise on Avoiding Overfitting

Sara Silva[1,2], Vijay Ingalalli[1], Susana Vinga[1,3], João M.B. Carreiras[4] Joana B. Melo[4], Mauro Castelli[1,5], Leonardo Vanneschi[5,1], Ivo Gonçalves[2], and José Caldas[1]

[1] INESC-ID, IST, Universidade Técnica de Lisboa, 1000-029 Lisboa, Portugal
[2] CISUC, Universidade de Coimbra, 3030-290 Coimbra, Portugal
[3] FCM, Universidade Nova de Lisboa, 1169-056 Lisboa, Portugal
[4] Instituto de Investigação Científica Tropical, 1300-344 Lisboa, Portugal
[5] ISEGI, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal
sara@kdbio.inesc-id.pt

Abstract. Mapping and understanding the spatial distribution of forest aboveground biomass (AGB) is an important and challenging task. This paper describes an exercise of predicting the forest AGB of Guinea-Bissau, West Africa, using synthetic aperture radar data and measurements of tree size collected in field campaigns. Several methods were attempted, from linear regression to different variants and techniques of Genetic Programming (GP), including the cutting edge geometric semantic GP approach. The results were compared between each other in terms of root mean square error and correlation between predicted and expected values of AGB. None of the methods was able to produce a model that generalizes well to unseen data or significantly outperforms the model obtained by the state-of-the-art methodology, and the latter was also not better than a simple linear model. We conclude that the AGB prediction is a difficult problem, aggravated by the small size of the available data set.

1 Introduction

The importance of accurately estimating forest aboveground biomass (AGB) has been recognized in the literature (e.g. [13]). Forest AGB is a key component when assessing the carbon stocks of a given ecosystem, and mapping its distribution is paramount to monitor forests and capture deforestation processes, forest degradation, and the effects of conservation actions, sustainable management and enhancement of carbon stocks. Furthermore, it is a requirement of international conventions (e.g., United Nations Framework Convention on Climate Change, UNFCCC), especially on the basis of reporting mechanisms developed under the UNFCCC post-Kyoto Protocol and particularly the initiative focusing on Reducing Emissions from Deforestation and forest Degradation in developing countries (e.g. [1]). Remote sensing data acquired by sensors onboard orbital platforms provide the only means to assess and monitor the status and change

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2013, LNCS 7835, pp. 407–417, 2013.

of biophysical characteristics of tropical forests in a global and systematic way. Numerous studies have demonstrated that a relationship exists between forest AGB and low frequency (L- and P-band) Synthetic Aperture Radar (SAR) data (e.g. [10]), though a high level of uncertainty still remains.

Genetic Programming (GP) is the automated learning of computer programs, using Darwinian selection and Mendelian genetics as sources of inspiration [15]. It is now a mature technique that routinely produces human-competitive results. However, a few open issues remain, overfitting being one of them. For a review of the state-of-the-art in avoiding overfitting in GP the reader is referred to [5]. The problem of AGB prediction has recently been used as a test case to assess the performance of an overfitting control technique, the RST [5]. The results showed a clear improvement when compared to the results of standard GP, but their quality was not assessed from the point of view of the application.

In this paper we tackle the AGB prediction problem using a much larger variety of methods. From classical regression methods, including recent improvements, to different variants and techniques of GP, including bagging and boosting, two GP techniques aimed at avoiding overfitting, and the cutting edge geometric semantic GP approach, they were all used in the context of a private "contest" that was launched with the goal of obtaining good models that can generalize well to unseen data. The next section describes the data and the terms of the contest. Section 3 describes the long list of methods used, while Section 4 reports and discusses the results. Finally, Section 5 draws some conclusions from this study.

2 Data

The dataset is composed of a combination of 112 forest AGB estimates and corresponding Advanced Land Observing Satellite (ALOS) Phased Array L-band Synthetic Aperture Radar (PALSAR) data covering the forested areas of Guinea-Bissau (West Africa). Forest AGB was estimated in two field campaigns that took place in 2007 (43 observations) and 2008 (69 observations). It was based on a stratified sampling methodology using an available land cover map of 2007. Individual trees were measured following a three-nest sampling plot methodology (4, 14, and 20m concentric sub-plots) and used in combination with allometric equations to obtain forest AGB estimates. ALOS PALSAR data was acquired in 2008 in fine beam dual (FBD) mode (i.e., HH and HV polarizations). After image processing, several metrics were extracted for the same locations (112 plots) that were measured in the two field campaigns. Those metrics were the minimum, maximum, mean, and standard deviation of the HH and HV polarizations, expressed in decibel (dB) units. Therefore we have eight features, that we designate as $x_1, ..., x_8$, where some are highly correlated, such as $x_2 - x_3 - x_4$ and $x_6 - x_7 - x_8$. More information about the data set can be found in [2].

For the contest, only 75 of the 112 samples were given to the participants, and the remaining 37 samples were held as the unseen data where to measure the quality of each proposed model. With the 75 samples the participants were free to do as they wished.

The extended data used in method 8 below (EXT-REAL) was obtained by randomly selecting 65536 samples from the study area (pixels from the image), with the only constraint that the distance between any two points cannot be lower than 200 meters. The synthetic extended data used in method 9 below (EXT-SYNT) was obtained by attributing to each of the eight features four different values ($4^8 = 65536$), equidistant from each other and inside the ranges given by the minimum and maximum values of each feature in the 75 samples.

3 Methods

A large part of the methods used and described next are based on GP, namely methods 5–13. All of these used 30 random partitions of the data (the 75 samples) as training (50 samples) and validation data[1] (25 samples). These partitions were the same for methods 5–11. The partitions were used as cross-validation to calculate the expected error, and in some cases to tune the parameters of the method.

Method 1 (LIN). The first method to be tested is multiple linear regression, using a stepwise selection algorithm, iterating forward selection and backward elimination steps based on the statistical significance of the regression coefficients. This procedure aims at having the simplest model. The model obtained uses only one feature: $y = 154.0373 + 8.7676x_6$. From now on it will be designated as LIN (linear).

Method 2 (LIN-NO). The second method is basically the same as LIN, but the model is fitted without the three detected severe outliers which have high Cook's distances. The model obtained is $y = 174.6253 + 9.8750x_6$ and it will be designated as LIN-NO (linear with no outliers).

Method 3 (EXP). Due to the asymmetry of variable V9, which has an exponential distribution with parameter 64.8839, and the non-normality of the errors obtained with the LIN and LIN-NO models, the logarithm transformation is tested. This resulted in the model $y = exp(8.1390 + 0.2680x_8)$, designated as EXP (exponential).

Method 4 (REG). The fourth method is standard linear regression with elastic net regularization [20]. The elastic net penalty is a linear combination of L1 and L2 regularization terms that aims at obtaining sparse weight parameters and assign similar weights to correlated predictors. The model obtained was $y = 191.6389 - 1.8963x_1 + 0.5056x_2 - 1.0050x_3 + 0.2156x_4 + 3.6368x_6 + 3.9242x_7 + 3.4831x_8$ and it will be designated as REG (regularization).

Method 5 (STD-GP). The fifth method is a common implementation of tree-based GP, using Dynamic Limits [17] for bloat control and a fixed maximum depth of 10. A population of 500 individuals, initialized with the Ramped Half-and-Half procedure [8], was allowed to evolve for 50 generations with standard

[1] In GP the validation data is often called test data. We call "unseen data" to the 37 samples that were not given to the participants, to avoid name confusion.

crossover and mutation (probabilities 0.9 and 0.1, respectively) and a replication rate of 0.1. The function set was composed of the four binary operators +, -, ×, and /, protected as in [8] and the terminal set included ephemeral random constants. Selection for reproduction was made with lexicographic tournament [11] of size 5. Elitism guaranteed the survival of the best individual into the next generation.

The resulting model (not shown) is, after simplification, a 67-node tree where features x_2, x_4 and x_7 do not appear, and it will be designated as STD-GP (standard GP).

Method 6 (WTD). The sixth method is similar to STD-GP, but it uses a weighted fitness function. The weighted fitness function is defined in terms of the Root Mean Square Error (RMSE):

$$f^* = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(W_i \cdot (E_i - P_i))^2} \tag{1}$$

where N is the number training samples, $\mathbf{W}$ is the weight vector, $\mathbf{E}$ is the expected values for the training data, and $\mathbf{P}$ is the predicted values for the training data. The weights are updated on every generation $\mathcal{G}$ with Algorithm 1. Fitness is given by $f = f^*$.

In the above algorithm, when $\eta = 1$, the values for $\mathbf{P^1}$ are already available. That is, the prediction values for generation $\eta = 1$ have already been calculated with $\mathbf{W} = \mathbf{1}$, so that we can update the weights for the next generation fitness function.

We update the weights depending on whether there has been any improvement in the prediction values over the generations. The magnitude of increase or decrease in the prediction values is reflected in the error value ε. The choice of error function ε is so that the updated weights do not reach saturation values for a small error differences. In other words, the error function $\varepsilon = 1 - (\varepsilon^\eta + 1)^{-1/2}$ is a slowly growing function in terms of differences between the expected and predicted values. If it were not the case (e.g., using exponential functions), then, even for a small deviation of predicted values from the expected values, we would have $\varepsilon \approx 1$, which should be avoided.

This approach of weighing is different form the usual weighing procedures (e.g. [14]), where each sample is re-weighed with respect to other samples in the

Algorithm 1. `Update Weights`

DEFINE: $\boldsymbol{\varepsilon}^\eta = |\mathbf{P}^\eta - \mathbf{E}|$ and $\boldsymbol{\varepsilon}^{\eta-1} = |\mathbf{P}^{\eta-\mathbf{1}} - \mathbf{E}|$ for any $\eta \in 1 \ldots \mathcal{G}$
$\boldsymbol{\varepsilon} = \mathbf{1} - (\boldsymbol{\varepsilon}^\eta + \mathbf{1})^{-1/2}$
INITIALIZATION: $\mathbf{P^0} = \infty$ and $\mathbf{W^0} = \mathbf{1}$
for $\eta \in 1 \ldots \mathcal{G}$ **do**
  $W_i^\eta = W_i^{\eta-1} * \varepsilon_i$; if $\varepsilon_i^\eta < \varepsilon_i^{\eta-1}$, for all $i \in 1 \ldots N$
  $W_i^\eta = W_i^{\eta-1} + (1 - W_i^{\eta-1}) * \varepsilon_i$; if $\varepsilon_i^\eta > \varepsilon_i^{\eta-1}$, for all $i \in 1 \ldots N$
  $W_i^\eta = W_i^{\eta-1}$; otherwise, for all $i \in 1 \ldots N$ RETURN:
  $\mathbf{W} = \mathbf{W}^\eta$;

training data. In this approach, re-weighing of a sample solely depends on the magnitude of change in error values and is not reflected upon by the magnitude of change of values of other samples.

The model that resulted from this method is, after simplification, a 17-node tree, which is very short for GP standards. We represent it with the expression $y = 2x_2 - x_3 + 3x_5 + x_6 + 73.078x_5/x_6$ and designate it as WTD (weighted).

Method 7 (WTD-17). The seventh method is basically the same as WTD, but uses the Dynamic Limits [17] with a fixed maximum depth of 17. The model that resulted from this method (not shown) is, after simplification, a 54-node tree where all features except x_2 appear, and it will be designated as WTD-17 (weighted with maximum depth 17).

Method 8 (EXT-REAL). This method is inspired by the work of Robilliard and Fonlupt [16]. Since their validation set was very small, they gathered thousands of additional samples from which the expected output was unknown, only knowing what reasonable bounds they should have. With this extended data set a new validation criterion was used: the lowest number of samples out of bounds, the better the model.

In this method we use real extended data as described in Section 2. When using the extended data D_{ext} of size N_{ext}, we make slight modifications to Equation (1). Let $UB = \max(\mathbf{E})$ be the maximum expected value in the training data, $LB = \min(\mathbf{E})$ the minimum expected value in the training data, and P_{ext} the prediction values for the extended data. We now define a "confidence" parameter as $c = \left| P_{bnd} \right| * 100/N_{ext}$, where $P_{bnd} = \{P^i_{ext} \in [LB, UB]\}_{i=1}^{N_{ext}}$. The confidence parameter c quantifies the proportion of our predictions that are in the range of expected values. We modify the fitness function as $f = f^*/c$.

The model that resulted from this method (not shown) is, after simplification, a 83-node tree where features x_2 and x_8 do not appear, and it will be designated as EXT-REAL (extended real data).

Method 9 (EXT-SYNT). This method is similar to EXT-REAL, but the extended data D_{ext} is synthetic data as described in Section 2. The model that resulted from this method (not shown) is, after simplification, a 93-node tree where all the features appear, and it will be designated as EXT-SYNT (extended synthetic data).

Method 10 (BAG). This method is a bagging of GP models. Instead of taking the training data and obtaining one model from it, we perform τ trials to obtain τ models. Each trial uses a training set that is formed by randomly drawing, with replacement, the same number of samples as the original training set (n=75). Then the output of the model is the median of the τ outputs for each instance. We used the median instead of the mean because of frequent surges observed in the prediction values. τ was set to 10.

By construction, the model that results from this method (not shown) is an ensemble of models, hence complex and difficult to interpret. We will designate it as BAG (bagging).

Method 11 (BOOST). In the normal weighted approach (WTD) we update the weights on every generation for a given instance of training data. In this method we perform τ trials for the training data and update a weighted distribution $\mathbf{D}$ for *each trial*. Under this method, each trial τ uses the same set of training samples, which are drawn at random without replacement at the beginning of trial 1. We adopt the commonly used $Ada - Boost$ (e.g. [7,14]) to update our distribution. Let us define $\mathbf{P}^{t-1}$ to be the best prediction values for the previous trial. Since $\mathbf{D}$ is updated at the end of each trial, we have $\mathbf{W}^{\eta}$ updates available from Algorithm 1. For each trial we use fitness $f = f^*$.

The boosting approach usually employs evaluating a final hypothesis / function based on τ functions, evaluated for each trial [14]. We follow a naive approach of selecting a function whose RMSE is the best among the τ evaluated functions. It has also been observed that such a best hypothesis is obtained from the t^{th} trial, where $t > \tau/2$. This confirms that there is a good chance of improvement by re-weighing over the trials, than just re-weighing over the generations, as followed in WTD and WTD-17 approaches. τ was set to 10.

By construction, the model that results from this method (not shown) is an ensemble of models, hence complex and difficult to interpret. We will designate it as BOOST (boosting).

Method 12 (RST). This method is the Random Sampling Technique (RST). The RST was originally used to improve the speed of a GP run [4], however in [9] it was used to reduce overfitting in a classification task in the context of software quality assessment. With the RST the training set is never used as a whole in the search process. Instead, at each generation, a random subset of the training data is chosen and evolution is performed taking into account the fitness of the solutions in this subset. This implies that only individuals that perform well on various different subsets will remain in the population. Recently, Gonçalves *et al.* [5] have proposed a more flexible approach to the RST, where the size of the random subset and how often it is changed are parameters of the algorithm. The authors tested their technique on real-life datasets and found the best results by using only one random sample in each generation. They also showed that the RST with these settings produces parsimonious models.

Algorithm 2. `Boosting`

1 INITIALIZATION: $\mathbf{D^1} = 1/N$
2 **for** $t \in 2 \dots \tau$ **do**
3 $\quad D_i^t = (D_i^{t-1})^{1-L_i}$, for all $i \in 1 \dots N$
4 $\quad$ where

$$L_i = \frac{\left|P_i^{t-1} - E_i\right|}{\max\left|\mathbf{P}^{t-1} - \mathbf{E}\right|}$$

5 $\quad$ UPDATE: $\mathbf{W}^{\eta}$ from Algorithm 1
6 $\quad$ NORMALIZE: $\mathbf{D^t}$
7 $\quad$ RETURN: $\mathbf{W} = \mathbf{D^t} * \mathbf{W}^{\eta}$;

We used RST with these settings. As for the regular GP parameters, the settings were similar to STD-GP except for these differences: the population was allowed to run for 100 generations, the tournament size was 2% of the population size, no random constants were in the function set, elitism guaranteed the survival of the best individual into the next generation, and no bloat control was used except for the fixed depth limit of 17. The model that resulted from this method is, after simplification, a 29-node tree represented by the expression $x_2 - 3x_1 + x_4 - 4x_5 + 8x_6 + x_7 - 3x_8 - x_4/(x_2 - x_1 + x_7)$, from now on designated as RST (Random Sampling Technique).

Method 13 (GS-GP). This method is a GP system that uses the geometric semantic genetic operators recently created by Moraglio *et al.* [12]. By semantics it is meant 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. The geometric semantic operators directly search the semantic space, and they have a number of theoretical advantages compared to the ones of standard GP systems. In particular, as proven in [12], 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 or regression). This should facilitate evolvability [6], making these problems potentially easier to solve for GP. The geometric semantic operators also have a major drawback: they always create offspring that are larger than their parents, causing an exponential growth of the individuals. However, with the development of a novel implementation [19] we were able to use them efficiently. This new GP system evolves the semantics of the individuals without explicitly building their syntax, freeing us from dealing with exponentially growing trees during the evolution. Only the best individual found must be explicitly built. For more details see [19].

A population of 200 individuals was allowed to evolve until 10000 fitness evaluations were completed, using similar settings to the RST method with a few differences: the tournament was regular (not lexicographic) and absolutely no bloat control was used. Both semantic operators were used, with a higher than normal mutation rate (0.5) since it was recognized that the geometric semantic mutation requires a higher rate for good exploration of the search space [19]. The mutation step of the geometric semantic mutation was 0.001 as in [12]. The model that results from this method (not shown) is a very large individual that we have not attempted to simplify. We will designate it as GS-GP (geometric semantic GP).

Method 14 (BAG-SGB). Stochastic Gradient Boosting (SGB) [3] typically uses a base learner (in our case, decision trees) and constructs additive regression models by sequentially fitting the chosen base learner to current "pseudo"-residuals by least squares at each iteration [3]. At these iterations, a simple base learner is built using a random sub-sample of the training data (without replacement), which has been shown to substantially improve the prediction accuracy and execution speed, and makes the approach resilient to overfitting [3]. The final model is a linear combination of each simple base learner, which can be seen as a regression model whereby each term is a tree. Furthermore, Suen *et al.* [18] have demonstrated that

building and combining (by averaging in the case of regression) several SGB models on bootstrap samples of the training data set performs significantly better than an unique SGB model, and concluded that it was accomplished by variance reduction. Therefore, instead of building a single SGB model, several SGB models fitted to bootstrap samples (with replacement) of the original training set (n=75) were built (BagSGB). In this study 25 bootstrap replicates were used to build a BagSGB model. For more details see [2].

By construction, the model that results from this method (not shown) is an ensemble of models, hence complex and difficult to interpret. We will designate it as BAG-SGB (bagging of stochastic gradient boosting).

4 Results and Discussion

Table 1 shows the results obtained by each method. We report the results in terms of root mean square error (RMSE) and correlation (CORR) between predicted and expected outputs. For the methods that used some kind of cross-validation, e.g. all the GP methods (that did 30 runs, each one with a different data partition - see Section 3), the expected error was calculated as the mean or median RMSE and CORR obtained in the 30 runs. We have decided to report both mean and median because the variability between runs was very high, and hence the median becomes a better estimate of the error. We report the error obtained on the unseen data, and when available we also report the error obtained in the training data.

None of the methods was able to produce a model that generalizes well on the unseen data. All RMSEs are high and accompanied by low (negative for all GP models) CORRs. The model with lowest RMSE and highest CORR is the one produced by the best state-of-the-art method, BAG-SGB, matched by the first two linear models, LIN and LIN-NO, and followed by REG. The non-linear models behaved much worse. Among the GP models, STD-GP and WTD-17 were the ones with higher RMSE on the unseen data, surprisingly followed by RST. None of the GP models was able to accurately estimate the error, with exceedingly high values in the mean expected RMSE, median expected RMSE always too optimistic, and expected CORR showing similar values between mean and median but completely failing to guess the negative values obtained on the unseen data. The only models providing similar values for the mean and median expected RMSE were BOOST and GS-GP, GS-GP being the less optimistic one. GS-GP is also the one achieving, by far, lower RMSE and higher CORR on the training data. With such results on the training data we could expect GS-GP to generalize worse, but in fact it is also the best GP model on the unseen data. This is explained by the geometric properties of its operators [19]. However, if we take into consideration also the simplicity and interpretability of the models, GS-GP cannot be considered the best of the GP models; among all the models BAG-SGB also cannot be considered the best. That award goes to the most simple linear model, LIN. It is noteworthy that BAG-SGB is reported to achieve RMSE 26.62 and CORR 0.95 when using the entire original data set of 112 samples [2], and with the 75 samples it is not better than a linear model.

Table 1. Results of the different models. Best of each column is marked in bold.

	Expected RMSE		Expected CORR		Training Data		Unseen Data	
Techniques	Mean	Median	Mean	Median	(Median Values) RMSE	CORR	RMSE	CORR
LIN	n/a	n/a	n/a	n/a	54.27	0.47	**74.50**	0.11
LIN-NO	n/a	n/a	n/a	n/a	55.00	0.47	75.88	0.11
EXP	n/a	n/a	n/a	n/a	58.07	0.46	87.03	0.03
REG	n/a	n/a	n/a	n/a	53.62	0.49	76.17	0.05
STD-GP	225.04	74.14	0.16	0.15	52.69	0.53	1253	-0.20
WTD	313.28	68.81	0.09	0.06	54.94	0.51	81.42	-0.02
WTD-17	8853	68.57	0.14	0.15	50.36	0.53	115.02	-0.10
EXT-REAL	505.03	68.97	0.00	0.08	56.57	0.45	82.30	-0.02
EXT-SYNT	86537	64.64	0.08	0.10	57.16	0.47	82.02	-0.28
BAG	94.46	62.09	**0.24**	**0.25**	52.51	0.61	81.24	-0.21
BOOST	**59.41**	**57.98**	0.22	0.22	61.92	0.33	80.91	-0.14
RST	86.25	66.23	0.03	0.07	51.55	0.58	88.55	-0.30
GS-GP	67.09	64.48	0.15	0.15	**44.12**	**0.74**	79.56	-0.03
BAG-SGB	n/a	n/a	n/a	n/a	n/a	n/a	75.07	**0.14**

5 Conclusions

We have performed an exercise on predicting the forest AGB of Guinea-Bissau, West Africa. In the context of a privately launched "contest", 14 methods were used but none was able to produce a model that generalizes well to unseen data. Even the best state-of-the-art method was not better than a simple linear model, despite the literature reporting much better results when using a larger data set. We conclude that the AGB prediction is a difficult problem, aggravated by the small size of our data set.

Acknowledgments. The presented work was partially supported by national funds through FCT under contract Pest-OE/EEI/LA0021/2011 and projects PTDC/EIA-CCO/103363/2008 and PTDC/EEI-CTP/2975/2012, Portugal. The Secretaria de Estado para o Ambiente e Desenvolvimento Durável (SEAD) of Guinea-Bissau and the Ministry of the Environment of Portugal funded and logistically supported the development of the Carboveg-GB project. This research was conducted under the agreement of the Japan Aerospace Exploration Agency's (JAXA) Kyoto and Carbon (K&C) Initiative. JAXA is particularly thanked for their provision of the ALOS PALSAR data.

References

1. Campbell, B.: Beyond Copenhagen: Redd plus, agriculture, adaptation strategies and poverty. Global Environmental Change-Human and Policy Dimensions 19(4), 397–399 (2009)
2. Carreiras, J., Vasconcelos, M., Lucas, R.: Understanding the relationship between aboveground biomass and ALOS PALSAR data in the forests of Guinea-Bissau (West Africa). Remote Sensing of Environment 121, 426–442 (2012)
3. Friedman, J.: Stochastic gradient boosting. Computational Statistics & Data Analysis 38(4), 367–378 (2002)
4. Gathercole, C., Ross, P.: Dynamic Training Subset Selection for Supervised Learning in Genetic Programming. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 312–321. Springer, Heidelberg (1994)
5. Gonçalves, I., Silva, S., Melo, J.B., Carreiras, J.M.B.: Random Sampling Technique for Overfitting Control in Genetic Programming. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds.) EuroGP 2012. LNCS, vol. 7244, pp. 218–229. Springer, Heidelberg (2012)
6. Gustafson, S., Vanneschi, L.: Crossover-based tree distance in genetic programming. IEEE Transactions on Evolutionary Computation 12(4), 506–524 (2008)
7. Iba, H.: Bagging, boosting, and bloating in genetic programming. In: Proceedings of GECCO 1999, vol. 2, pp. 1053–1060 (1999)
8. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
9. Liu, Y., Khoshgoftaar, T.: Reducing overfitting in genetic programming models for software quality classification. In: Proceedings of the Eighth IEEE Symposium on International High Assurance Systems Engineering, Tampa, Florida, USA, March 25-26, pp. 56–65 (2004)
10. Lucas, R., Armston, J., Fairfax, R., Fensham, R., Accad, A., Carreiras, J., Kelly, J., Bunting, P., Clewley, D., Bray, S., Metcalfe, D., Dwyer, J., Bowen, M., Eyre, T., Laidlaw, M.: An evaluation of the alos palsar l-band backscatter – above ground biomass relationship over Queensland, Australia. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 3(4), 576–593 (2010)
11. Luke, S., Panait, L.: Lexicographic parsimony pressure. In: Proceedings of GECCO 2002, pp. 829–836. Morgan Kaufmann (2002)
12. 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)
13. Pan, Y., Birdsey, R., Fang, J., Houghton, R., Kauppi, P., Kurz, W., Phillips, O., Shvidenko, A., Lewis, S., Canadell, J., Ciais, P., Jackson, R., Pacala, S., McGuire, A., Piao, S., Rautiainen, A., Sitch, S., Hayes, D.: A large and persistent carbon sink in the world's forests. Science 333(6045), 988–993 (2011)
14. Paris, G., Robilliard, D., Fonlupt, C.: Applying Boosting Techniques to Genetic Programming. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) EA 2001. LNCS, vol. 2310, pp. 267–918. Springer, Heidelberg (2002)
15. Poli, R., Langdon, W.B., Mcphee, N.F.: A field guide to genetic programming (March 2008), `http://www.gp-field-guide.org.uk`
16. Robilliard, D., Fonlupt, C.: Backwarding: An Overfitting Control for Genetic Programming in a Remote Sensing Application. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) EA 2001. LNCS, vol. 2310, pp. 245–254. Springer, Heidelberg (2002)

17. Silva, S., Costa, E.: Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. Genetic Programming and Evolvable Machines 10(2), 141–179 (2009)
18. Suen, Y.L., Melville, P., Mooney, R.J.: Combining Bias and Variance Reduction Techniques for Regression Trees. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 741–749. Springer, Heidelberg (2005)
19. 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)
20. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B 67, 301–320 (2005)