Complexity measures in Genetic Programming Learning: A Brief Review

Nam Le*, Hoai Nguyen Xuan*, Anthony Brabazon†, and Thuong Pham Thi*
*Hanu IT Research and Development Center
Hanoi University,
Email: namlehai90@gmail.com
nxhoai@gmail.com
ptthuong@ictu.edu.vn
†University College Dublin, Ireland
Email: Anthony.brabazon@ucd.ie

***Abstract*—Model complexity of Genetic Programming (GP) as a learning machine is currently attracting considerable interest from the research community. Here we provide an up-to-date overview of the research concerning complexity measure techniques in GP learning. The scope of this review includes methods based on information theory techniques, such as the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC); plus those based on statistical machine learning theory on generalization error bound, namely, Vapnik-Chervonenkis (VC) theory; and some based on structural complexity. The research contributions from each of these are systematically summarized and compared, allowing us to clearly define existing research challenges, and to highlight promising new research directions. The findings of this review provides valuable insights into the current GP literature and is a good source for anyone who is interested in the research on model complexity and applying statistical learning theory to GP.**

***Keywords*—Genetic Programming; Complexity measure; Model Selection; Vaknik-Chervonenkis dimension; Statistical Machine Learning; Rademacher complexity.**

I. Introduction

Genetic Programming (GP), first officially developed by John Koza [1] has recently been extensively applied to various machine learning (ML) problems with promising results (see Poli et al. [2]). Therefore, GP has gradually been deeply investigated and adopted as a machine learning method, which learns models from data.

It is not surprising that generalization has long been regarded as one of the most important and desirable properties of learning machines. This property implicitly looks at the phenomenon of overfitting; a phenomenon, which occurs when a learning system selects a model that fits a particular, set of training data but generalizes poorly on out-of-sample data. In addition, it is widely known that the generalization - overfitting paradox relates directly to the model selection task in machine learning and statistics [14]. That is to say, model selection based merely on the fit to observed data will tend to lead to the choice of an unnecessarily complex model that overfits the training data and generalizes poorly. To combat these problems, regularization [9] is often used in machine learning, to reduce overfitting by adding a complexity (penalty) term to the loss function. That term is used to control the complexity of a learning model to guarantee that model selection should be based not only on the goodness-of-fit to the observed data.

To perform regularization (model selection) in GP learning, it is necessary to define the ”complexity” of a GP learning model. Since the GP evolutionary learning process always maintains a population of learning models called individuals, it is individual complexity that needs to be measured. However, for the purpose of regularization, it is nontrivial to define and calculate the complexity of GP individuals for several reasons as follows: First, each GP individual, even though initialized in various ways (standard tree-based GP [1], linear GP [23], graph-based GP [24], and grammar-based GP [25]), always has a variable size. Second, GP conforms to theory of evolution to evolve computer programs as individuals. GP individual chromosomes must progress through several evolutionary steps of mutation and recombination and thus, they often increase in size. Thus, it follows that each GP individual chromosome is likely to have non-effective code (introns) [26] and suffer from bloat [2]. Recently, model selection and model complexity has drawn much attention from GP community in particular [52], [59]. The aim of this paper is to overview recent researches on model complexity in GP learning, showing the advantages and disadvantages of various approaches, and outlining some possible future research directions in this area

The rest of the paper is organized as follows: Section 2 introduces some background on statistical learning theory, model selection, GP, and the generalization problem. Section 3 provides a literature review. In section 4, a discussion of the literature is provided. Section 5 provides some conclusions.

II. Background

In this section, we first briefly review some important concepts of statistical learning theory. Then, a list of common model selection criteria is given. Finally, we discuss the issue of GP learning generalization.

A. Statistical learning theory

In [4], Vapnik and Chervonenkis proposed a remarkable family of upper bounds on the generalization error of a learning machine, in which the Vapnik-Chervonenkis dimension (VC-dimension) is a central concept for measuring the capacity of a family of functions (or learning machines) $f \in$H as classifiers. The VC-dimension of a class of (bounded) functions H is defined as the maximum number of points that can be shattered by H. For instance, the VC dimension of hyperplanes in R^n is n + 1 [4].

Generally, the error, ε(f), of a learning machine f is defined as:

$$\varepsilon(f) = \int Q(x, f; y) d\mu \tag{1}$$

where Q measures a loss between $f(x)$ and y, and μ is the (unknown) distribution from which sample (x, y) are drawn, usually *x* is called the instance and *y* the label. For instance, for classification problems, Q$(x, f; y)$ = $|y - f(x)|$, and the error is the misclassification rate. For regression problems, commonly Q$(x, f; y)$ = $(y - f(x))^2$ (mean square error). Many of the classic applications of learning machines can be explained within this formalism.

Since μ is usually unknown, in practice the theoretical error ε(f) is replaced by the empirical error which is estimated from a finite sample $\{x_i, y_i\}_{i=1}^n$ as:

$$\varepsilon(f) = \frac{1}{n} \sum_{i=1}^{n} Q(x_i, f; y_i) \tag{2}$$

The main results in [4] state that the error ε(f) can be bounded and independent of the distribution of μ(x, y) as in the following formula:

$$\varepsilon(f) \leq \varepsilon_n(f) + \sqrt{\frac{h(\log(2n/h) + 1 - \log(\eta/4)}{n}} \tag{3}$$

where η is the probability that bound is violated and *h* is the VC dimension of H from which function *f* is selected. The second term of the right hand side is called the VC confidence as it depends only the VC dimension given *h*, η, and *n*. (as example, for $h = 200, n = 100000$, and η = 0.95 the VC confidence is 0.12)

Statistical learning theory plays a major role in model selection in that one should select a model with lower bound on generalization error that conforms to one of machine learning theory mechanisms.

B. Model Selection and Model Complexity

Model selection is the task of selecting a statistical model from a set of candidate models given data drawn from an unknown distribution. Once the set of candidate models has been chosen, the statistical analysis allows us to select the best of these models. The terms "best" here can be interpreted in two ways: generalization error and training error. Generalization error is a model's error rate on unseen data, and training error is the error on the dataset used to generate the model. Overfitting the training set is a serious problem in machine learning that usually leads to low generalization capabilities. Thus, generalization and overfitting are two sides of the same coin. When comparing two models, the model with the lower training error is generally considered superior. When comparing models with similar training error, however, the model with lower complexity is usually preferred in the hope that it will have good prediction ability on future data. This is often described as Occam's razor in machine learning [10]. Roughly speaking, when applied to machine learning, Occam's razor can be stated in two forms:

i) Given two models with the same generalization error, the simpler one should be preferred because simplicity is desirable in itself.
ii) Given two models with the same training error, the simple one should be preferred because it is likely to have lower generalization error.

So, a central theme of model selection is that to avoid choosing unnecessarily complex models, a model should be selected based on its generalizability, rather than its goodness of fit. This goal is realized by defining a selection criterion that makes an appropriate adjustment to its goodness of fit by taking into account the contribution from model complexity. Further, model complexity is closely related to the bound on generalization error of learning model [10]. Thus, there are several different selection methods that are currently in use. They differ from one another in terms of how such adjustments are inserted to estimate a model's generalizability and complexity. The list of these methods include the Akaike Information Criterion (AIC) [11], the Bayesian Information Criterion (BIC) [12], the Root mean squared deviation (RMSD) [15], the Minimum Description Length (MDL) [19], Cross-Validation [21], Bayesian Model Selection (BMS) [22], Information-theoretic Measure of Complexity (ICOMP) [20]. In addition, regularization [9] should also be counted as a method used to prevent the model from overfitting the training sample by inserting a parameter (called regularization parameter) that is used to control the model complexity.

It is obvious that for any learning technique, how to define and control the complexity of the learning model is an important question.

C. GP and Generalization

Genetic programming recently has been considered a learning machine with some promising applications and results [27], [28], [29], [30]. As a learning technique, GP has to confront the generalization/overfitting even though this was not thoroughly considered in the early history of the field of GP. The initial work on GP mainly focused on solving problems just on the training data set, without considering the overfitting phenomenon. Before Kushchu published his seminal paper on the generalization capability of GP [5], there was rather little work in the literature to deal with the GP generalization aspect. Recently, this issue has attracted more attention from GP research community, with methods such as Sampling based approaches [32], Ensemble learning: Bagging and Boosting [33], Regularization/Early Stopping [31] being used.

Moreover, there has been a number of works on applying traditional machine learning techniques and practices to the learning process of GP to improve its generalization ability such as those presented in [15], [16], [17], [18]. However, if considered to be a machine learning technique, each GP individual is a model and the correlation between model complexity and generalization holds in the case of GP learning. This relationship has been widely studied by the machine learning community, and a strong theoretical understanding [4] has been obtained. However, defining and calculating model/individual complexity in GP is a nontrivial task. It has been seen in the literature that there has been some attempt to define and measure the model complexity in GP learning and the objective of the next section is to give an overview of this literature.

III. Model Complexity In GP Learning

In this section, we review a number of approaches proposed in the literature to measure the model/individual complexity for GP learning. It is noted that in GP, one could distinguish between the individual (model) genotype and its phenotype. While the genotype of an individual is its contents (typically compositions of functions and terminals from predefined sets), its phenotype is how it behaves (often on a set of inputs and outputs). Therefore, we will first classify the approaches to defining model/individual complexity in GP according to whether it is on genotype or phenotype, which we shall call (i) Structural complexity of model and (ii) Functional or Behavioral complexity. More recent approaches rely on complexity measures from statistical model selection, which give us the third group of model complexity in GP learning.

A. Structural Complexity

Different methods have been proposed to measure the structural complexity of a GP individual: (i) Number of nodes in a tree (ii) Number of levels in a tree (iii) Minimum description length (iv) Expressional complexity of a model determined by sum of number of nodes in all sub-trees of a given model.

The first approach in this type of complexity was proposed by Iba et al. in [16], which used MDL-based fitness functions to control the size of evolved GP trees. The authors reported that this method has desirable generalization ability not only for pattern recognition tasks but for other applications as well. But the disadvantages of this method were also noted. It only works with the initialization by decision tree structure along with two predefined conditions holds. The first one is "Size-based Performance", stating that "the more the tree grows, the better its performance". And the other is "Decomposition", expressing that "the fitness of a substructure is well-defined itself"; that means if the tree has good substructures, its fitness is necessarily high. Thus, extending MDL-based method to general applications may face considerable challenges. Moreover, this method is lack of flexibility in balancing accuracy with parsimony in unknown environments. Moreover, it is likely to converge prematurely if network size is penalized too much, despite using other diversity promoting mechanism such as having a large crossover rate.

To overcome this problem, Zhang and Heinz [35] continued Iba's idea and added Occam factor α(g) such that:

$$F_i(g) = E_i(g) + \alpha(g)C_i(g), \tag{4}$$

where $F_i(g)$, $E_i(g)$ and $C_i(g)$ are the fitness, the error and complexity values, respectively. In this MDL-based approach, the complexity has impacts on the selection process only when the candidates for selection have comparable performance. In other words, a tree (model) will be selected over others if and only if its error is smaller than those of other trees, or it has the same error but smaller size than others. This follows Occam's razor theory in machine learning, thus improving the generalizability of GP.

Vladislavleva et al. [39] defined genotypic measure that is related to counting number of nodes of a tree and its subtrees along with the number of layers. This method favors the flatter trees (i.e., trees with fewer layers and, hence, with fewer nested functions) over deep unbalanced trees (in the case of an equal number of nodes). We can clearly see an example of such a case in Fig. 1. The complexity can be interpreted as a size of the model obtained by substituting all inner functions of the model by their function bodies.

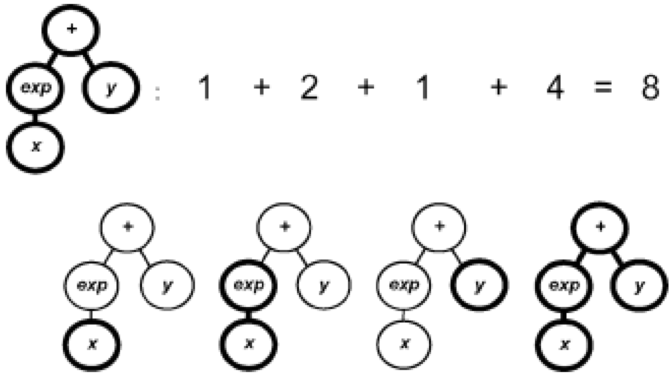

Figure 1: Complexity of a tree model is the total number of nodes in all subtree models

Another similar work should be mentioned is of M. Schmidt and Hod Lipson [63] in which they used Parato optimization to select individuals based on two dimensions: the age (how long the solution has been present in the population) and the fit to the data (overfitting). To measure complexity and control the fit to the data, they also used the number of nodes in the final expression tree for the target function in hopes of proving that the more the complexity, the more the chances of coupled nonlinear features, making it to be more likely overfitted to the given data. One different point of this method is that the authors generate a random equation using inputs and simplify the symbolic equation before measuring its complexity. Their experimental results showed that Age-fitness Pareto outperforms other available techniques in some regression problems.

The apparent shortcomings of node counting have motivated GP researchers to consider other complexity measures. A noticeable study is the one of Conte et al. [36], which proposed the use of Kolmogorov Complexity [37] as a complexity measure of GP individuals. The Occam's razor principle also held because it is closely related to Kolmogorov Complexity definition, as clearly stated in [37]. Shortly later, it was De Falco [38], who extended Conte's study to evolve a population of LISP programs. Their experimental results showed the effectiveness in obtaining a good approximation for complicated string compression, which then turn to for help in calculating edit distance between GP individuals and comparing their complexities.

In summary, methods based on size of GP program, especially focusing on counting number of nodes, might be deceptive. There are several reasons for this. First, the case when many nodes are introns is ubiquitous in GP. Moreover, even a large program tree in GP could also be compressed and simplified as a small expression [47]. The size of individual affects the dynamics of the evolutionary process, but bring us little in terms of the output of each program. Second, the genotype-phenotype discrepancy is not obvious in GP, where an explicit phenotype is normally not defined [48]. So, it is easier to concentrate on the functional output of a program to show its behavior. And thus, GP researchers have proposed the use of functional complexity measures for GP learning.

B. Functional Complexity (Behavioral Complexity)

A number of GP researchers believe that overfitting phenomenon in GP learning is associated with the functional complexity of the solution. Functional complexity of a model should be measured by computing model's behavior (output) over possible input space, thus called behavioral complexity.

In [39], Vladislavleva et al. introduced a new complexity measure called ”order of nonlinearity” to overcome the shortage of node counting based measures mentioned in the previous subsection. The order of nonlinearity of a model is computed by the order of the Chebyshev polynomial used to approximate the model. We refer to Fig. 2 to demonstrate how to compute the order of nonlinearity for a given tree (model/individual).

The concept behind the proposed measure is that over-fitted models are approximated by polynomial of high degree due to high oscillation in their behavior [40]. Parsimony pressure approach is suggested in [41] to reduce the complexity of models and thus to improve the generalization ability of the evolved models.

It should be noted that, the authors have not directly solved the model selection problem in GP learning but converted it into a model selection problem on the set of polynomial fits given an accuracy (the $epsilon$ in their definition). For functions of more than two variables, however, it is hard to construct Chebyshev polynomial approximation of a given accuracy, thus making the comparison of the order of nonlinearity become almost impossible.

To overcome these issues, Vanneschi et al. [43] proposed a new way to measure the nonlinearity based on the summation of partial complexity of each dimension inspired by the theory of generalized curvatures [44]. A noteworthy discovery in [43] is that, contrary to popular belief, bloat is independent from the overfitting phenomenon in GP learning. This refuted other research that had regarded bloat control as method of overfitting avoidance. However, as the authors used protected division in their GP formulation, it is not clear how the second-order derivative could be defined for this discontinuous function [45].

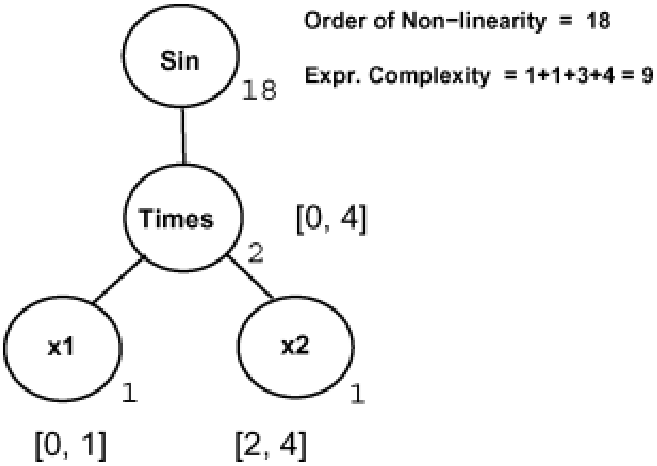

Figure 2: Example of nonlinearity calculation for a simple two-variable model. If x1 ∈ [0, 1] and x2 ∈ [2, 4], then ”x1 × x2” takes values from the interval [0, 4]. Nonlinearities of the terminal nodes are one. The nonlinearity of the ”Times” node is 1 + 1 = 2. Therefore, the nonlinearity of the Sin node is two times the degree of the Chebyshev approximation of function sin(x) on the interval [0, 4]. If the chosen approximation accuracy is 10^{-6}, then the order of nonlinearity of the root node is 2 × 9 = 18.

Following the concept of curvature, Vanneschi et al. [39] proposed a complexity measure called Slope-based Functional Complexity (SFC). This measure was computed by taking sum of differences of slope of consecutive line segments. This measure seems to be a pretty good approximation of a complexity measure to predict GP overfitting. However, the definition of SFC made it complicated for practice as noted by the authors. To conquer this limitation of SFC, the authors simultaneously proposed a new method called Regularity-based Functional Complexity (RFC). This measure based on the concept of Holderian regularity [49] not only captures the underlying advantage and notion of SFC, but also could be resistant to its practical difficulties. Experimental results of both methods showed almost no correlation between both complexity measures and program overfitting. And the RFC method only works well as overfitting indicator when considering highly overfitted solutions. On the other hand, the SFC measure could not achieve any useful correlation with program overfitting, and even resulting in unexpected outcomes. They also suggested that future research should focus more on comprehensive evaluation of these measures of complexity as indicators of GP overfitting.

To sum up, complexity measures in this section are mainly based on the non-linearity and/or curvature of the function defined by a GP individual. They are computed either directly (as in the approaches proposed by Vanneschi et al) or indirectly (as in the work by Vladislavleva et al.). It is unclear, however, the nonlinearity and/or curvature of the function is always related to overfitting/generalization of the model. It could be thought that more complex and nonlinear function should have more capacity to overfit the training data, however, as shown in [4], it is not necessary the case. In [4], Vapnik gave a counter-example for a class of functions that have arbitrary high degree of nonlinearity (curvature) but possesses very limited learning (overfitting) capacity (with small VC dimension). In the following section, we will cover some important researches on complexity measures using Statistical Learning Theory (SLT) for model selection in GP.

C. Statistical Machine Learning Complexity Measurement

In modern machine learning research, it is recommended that people should investigate the effects of machine learning theory anytime they want to research into the model selection problem as well as model complexity. This is due to the fact that model complexity has a close relationship with generalization error bound. This research trend has also affected GP research community and thus, several recent works on using statistical learning theory have been conducted with some promising results.

Perhaps, L. Alonso et al. [50] were among the first researchers in this research direction. First, they formulated a new encoding scheme using a data structure called straight-line program [51] (slp) to encode linear GP programs [50]. SLP have often been used as a parameter for complexity analysis in algebraic complexity theory because of its flexible representation for expressions (please see [51] for more details). Employing SLP data structure facilitates representing complex expressions with fewer amounts of instructions than the tree data structure. Another thing, experimental results showed that SLP-based GP has better performance than standard tree-based GP on some symbolic regressions in terms of convergence rate and solution quality. Moreover, this representation makes it easier to define Vapnik-Chervonenkis (VC) dimension [4] for GP individual than for complex tree expressions in standard tree-based GP. Therefore, it facilitates investigating machine learning techniques based on VC dimension for GP. Later, they conducted the first research on model selection along with a new complexity criterion for SLP-based GP trees [52]. In that research, the authors used three different model selection methods as follows:

- Akaike Information Criterion (AIC):

$$\varepsilon(f) = \varepsilon_n(f) + \frac{2h}{n}\sigma^2 \tag{5}$$

- Bayesian Information Criterion (BIC):

$$\varepsilon(f) = \varepsilon_n(f) + (\ln n)\frac{h}{n}\sigma^2 \tag{6}$$

As described in [53], the noise variance from the training data (x_i, y_i) is estimated as:

$$\sigma^2 = \frac{n}{n-h}\frac{1}{n}\sum_{1\le i\le n}(y_i - \hat{y}_i)^2 \tag{7}$$

$\hat{y}_i$ is the estimation of value y_i by model f, i.e, $\hat{y}_i = f(x_i)$. We can use the equation (7) in conjunction with AIC or BIC for each model complexity. The estimation of the model complexity h for both methods is the number of free parameters of the model f (for more details please see [55] on equation 3).

The third model selection method used in [52] is based on the Structural Risk Minimization (SRM) (see [4]).

$$\varepsilon(f) = \varepsilon_n(f).\left(1 - \sqrt{p - p\ln p + \frac{\ln n}{2n}}\right), \tag{8}$$

where $p = \frac{h}{n}$, and h stands for the VC-dimension as a measure of model complexity. This complexity is measured by the number of non-scalar nodes of the tree for SRM method (please refer to [52] on Theorem 1 for more details about how to compute VC-dimension for SLP-based GP genotype). Note that under SRM approach, we are not required to compute noise variance but the VC dimension instead.

Experimental results indicated that SRM model selection performed clearly better than AIC and BIC in almost all cases, even though affected by noise. AIC and BIC are of the same quality in general training.

After that, these authors extended their previous research ideas in [55] by adding Pfaffian operators [56] to the initialization function sets. They concluded that their approach using VC dimension (VCD) regularization is clearly the best. And also, they have found a theoretical upper bound of families of SLPs over Pfaffian operators, which is polynomial in the number of the non-scalar instructions of the family of the SLPs. This bound was stated to be distribution independent, thus improving generalization ability of GP for any data distribution. Finally, they also suggested a new research idea for future work that may compare with other non-statistical methods such as Tikhonov regularization.

Nevertheless, this method has a drawback that extending VC-based SRM approach for standard tree-based GP is not straightforward as noted in [62].

Another remarkable study undertaken by Ji Ni and Peter Rockett [57] employed Vicinal-Risk Minimization [58] for training GP classifiers. This approach expressed an advantage over SRM-based approach of Alonso et al. in that VRM can be applied to GP more easily than SRM and it is readily tractable. Therefore, it can be used to stabilize the training process of GP. Experimental results also pointed out that this approach obtained high probability of yielding better results than empirical risk minimization (ERM) which have usually been used as error estimate in GP, even though in a single run.

In summary, methods based on statistical learning theory described in this section showed promising results when applied to GP. Although still at a preliminary stage, they are potential indicators for future research into this field.

IV. Discussion

Over the past decade, model selection and complexity measures have been widely studied topics. All of methods surveyed in this paper show strengths and weaknesses. In our opinion, these approaches contribute to the literature in different ways.

Early approaches used to measure and control the complexity of GP models by size, node count, number of layers - or structural complexity for short, make it easier to control the code growth of GP individuals, facilitating the bloat control. These techniques be extended for other purposes, and are not just restricted to the complexity measure scenario, for example in tree compression technique used in several studies like [47]. However, these methods have the same drawback that the size of GP individual might be variable, making it difficult to control structure of GP model. Furthermore, if the tree size is significantly restricted, it will produce premature convergence.

Approaches based on functional or behavioral complexity seem to combat the problem of variable size. These methods share the same idea of using algebraic structures and polynomials in complexity analysis of GP learning models. In our opinion, the research of Vanneschi et al. [43] has contributed considerably. First, it not only indicated the use of curvature theory to overcome the obstacle of approximating Chebyshev polynomials, but also resulted in the significant discovery that bloat is independent from the overfitting issue. This will help GP researchers carefully and separately considering bloat issue and overfitting if they want to improve the generalization capability of GP learners. Another advantage that should be mentioned is that these methods can be applied to standard tree-based GP more easily than others. However, these approaches are still lack a strong theoretical base, making it hard to provide reliable model selection advice for GP systems. In fact, the theoretical results in [4] cast a doubt on whether the functional complexity is related to overfitting/generalization in statistical learning. This gap has increased the activity of the GP community into researching computational machine learning theory for GP model selection and complexity control.

Several model selection methods with model complexity measures based on statistical machine learning have also been studied. Preliminary research results from Alonso et al. suggest a potential future research interest and a novel data structure for GP with more a flexible representation and more suitable to statistical measurements. Despite the hopefulness, it was a big problem of computing VC dimension as well as using traditional model selection technique (AIC, BIC, for example) in standard tree-based GP. But it would be considered a prospective challenge for future research. From studying the complexity measure and model selection of Genetic Programming is very important, and statistical methods deserve further research because there might be a few new techniques have not yet been applied. We suggest that, based on the initial success of VC-dimension for SLP-GP, possibly Rademacher and Gaussian Complexity [60] might be of effectiveness for GP model selection. The reason is that Rademacher (for classification tasks), or Gaussian (for regression tasks) are distribution (data) dependent measure, which have been shown to yield a lower bound on generalization error than the distribution independent VC based approaches. Furthermore, the Probably Approximately Correct (PAC) approach of [54] could also be used as a learning framework that GP can conform to (as in [61], for example).

V. Conclusion

Model complexity measures and model selection for GP is currently attracting considerable interest from research community. The purpose of this review was to view the trends in model selection and model complexity measures for GP over the past decade and to help the reader understand different aspects posed by the research on Genetic Programming as learning machine. This is essential because model selection play an key role in improving the generalization ability and reducing the overfitting of any machine learning model. And furthermore, there are many GP researchers and practitioners, often not realizing the importance of model complexity control or ignoring this issue.

The scope of this review was on core techniques in complexity measures and control, including structural complexity, behavioral complexity, and statistical model selection based complexity measures. However, the practical application of these methods shows that each has their own advantages and disadvantages.

Our first conclusion is that considering the structure of GP tree could be a helpful method to control the GP model complexity, boosting it to be better generalizable. For simplicity, node counting gives satisfactory results on average but could be unreliable when the tree grows up and varies from generation to generation. For better generalization ability and reliability, Kolmogorov complexity that is related to Occam's razor theory is introduced and shows a good estimation for complicated functions along with other side-effects. In general, the choice should lie in how to control the shape and variation of GP tree (model) to make it better.

We must concede to surprise at how well the algebra help measure and control the model complexity of GP model. The simulation study has been based on non-linearity or curvature of a function expressed by a GP individual. By using some algebraic concepts such as Chebyshev polynomial, Slope-based Functional Complexity (SFC) [39] or Holderian regularity [49], authors tried to compute the non-linearity order and used the criteria to approximate the GP individual and link to GP model complexity, although just produced limited outcomes. And one important discovery from their works is that bloat control makes little sense in overfitting reduction. The idea, nevertheless, could be somewhat suspected when Vapnik (in [4]) showed a counter-example for the correlation between the high order of nonlinearity and the high level of overfitting (with small VC dimension). Despite this, these works opened a new prospect for further researches that could use advanced algebraic techniques to help control GP function (model).

The performance of statistical learning theory based complexity measure is most promising. Clearly this idea is worthy for future investigation. One obvious drawback of this method is that the parameter for model complexity requires complex computation. Most of the works mainly focused on one special data structure representing GP individuals called Straight Line Program and their results did not completely indicate that which statistical method may be quite robust. However, the great advantage of this technique is that it injected Statistical learning theory into controlling GP learning models like other machine learning algorithms, anyway. Moreover, it motivated a new trend for further research on other model selection techniques from machine learning with respect to GP, making it increasingly dependable.

The contributions of research works on each method are systematically encapsulated and compared above, which allows us to clearly define existing research challenges, and highlight promising new research directions. It is hoped that this survey can serve as a useful guide through the maze of the literature on these topics. In general, the expectation of the preliminary results on different model selection and complexity control techniques for GP could be a leverage. Also, we hope that, one day, GP could be completely understandable with respect to theoretical machine learning.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] Koza, J. R. (1992a). Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA

[2] R. Poli, W. B. Langdon, and N. F. McPhee. A Field Guide to Genetic Programming. lulu.com, 2008.

[3] N L Cramer: A representation for the Adaptive Generation of Simple Sequential Programs. Proceedings of an International Conference on Genetic Algorithms and the Applications , pp. 183-187, 24-26 July 1

[4] Vapnik, V. N. (1998), Statistical Learning Theory , Wiley-Interscience.

[5] Ibrahim Kushchu. Genetic programming and evolutionary generalization. IEEE Transactions on Evolutionary Computation, 6(5):431-442, October 2002. ISSN 1089-778X.

[6] Dan Costelloe and Conor Ryan. On improving generalisation in genetic programming. In Proceedings of the 12th European Conference on Genetic Programming, EuroGP

[7] C. Gagne, M. Schoenauer, M. Parizeau, and M. Tomassini. Genetic programming, validation sets, and parsimony pressure. In Proceedings of the 9th European Conference on Genetic Programming, volume 3905 of Lecture Notes in Computer Science,pages 109-120. Springer, 2006.

[8] L. Vanneschi and S. Gustafson. Using crossover based similarity measure to improve genetic programming generalization ability. In Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO 09), pages 1139-1146. ACM, 2009.

[9] L. Devroye, L. Gyorfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, New York, 1996.

[10] Partha Niyogi , Federico Girosi, On the Relationship between Generalization Error, Hypothesis Complexity, and Sample Complexity for Radial Basis Functions, Massachusetts Institute of Technology, Cambridge, MA, 1994

[11] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle, In B. N. Petrox and F. Caski (Eds.), Second international symposium on information theory, pp. 267. Akademiai Kiado, Budapest.

[12] Wasserman, L. (2000). Bayesian model selection and model averaging. Journal of Mathematical Psychology, 44, 92 - 107.

[13] Friedman, E., Massaro, D. W., Kitzis, S. N., 6 Cohen, M. M. (1995). A comparison of learning models. Journal of Mathematical Psychology, 39, 164 - 178.

[14] Myung, I. J. (2000). The importance of complexity in model selection.Journal of Mathematical Psychology,44, 190-204

[15] Christian Gagne, Marc Schoenauer, Marc Parizeau, and Marco Tomassini. Genetic programming, validation sets, and parsimony pressure. CoRR, abs/cs/0601044, 2006. URL http://arxiv.org/abs/cs/0601044. informal publication

[16] Hitoshi Iba, Hugo de Garis, and Taisuke Sato, 1994, Genetic programming using a minimum description length principle, Advances in Genetic Programming, chapter 12, pp. 265-284, MIT Press, Cambridge, MA, USA.

[17] David Jackson and Adrian P. Gibbons. Layered learning in boolean GP problems. In Marc Ebner, Michael O'Neill, Aniko Ekart, Leonardo Vanneschi, and Anna IsabelEsparcia-Alcazar, editors, Proceedings of the 10th European Conference on Genetic Programming, volume 4445 of Lecture Notes in Computer Science, pages 148-159,Valencia, Spain, 11-13 April 2007. Springer. ISBN 3-540-71602-5. doi: doi:10.1007/ 978-3-540-71605-1 14.

[18] Byoung-Tak Zhang. Bayesian methods for efficient genetic programming. Genetic Programming and Evolvable Machines, 1(3):217-242, July 2000. ISSN 1389-2576. doi: doi:10.1023/A:1010010230007.

[19] Grunwald, P. (2000). Model selection based on minimum description length. Journal of Mathematical Psychology, 44, 133 -152

[20] Bozdogan, H. (2000). Akaike information criterion and recent developments in information complexity. Journal of Mathematical Psychology, 44, 62 - 91.

[21] Browne, M. W. (2000). Cross-validation methods. Journal of Mathematical Psychology, 44, 108 - 132.

[22] Wasserman, L. (2000). Bayesian model selection and model averaging. Journal of Mathematical Psychology, 44, 92 - 107.

[23] J. P. Nordin. A Compiling Genetic Programming System that Directly Manipulates the Machine code. MIT Press, Cambridge, 1994.

[24] A. Teller and M. Veloso. Pado: A new learning architecture for object recognition. In Symbolic Visual Learning, pages 81 -116. Oxford University Press, 1996.

[25] N.X. Hoai, R.I. McKay, D. Essam, Representation and structural difficulty in genetic programming. IEEE Trans. Evol. Comput. 10(2), 157-166 (2006)

[26] P. Nordin, W. Banzhaf, and F. D. Francone, "Introns in nature and in simulated structure evolution", in Bio-Computation and Emergent Computation, Skovde, Sweden, D. Lundh, B. Olsson, and A. Narayanan (eds.), World Scientific Publishing, 1-2 September 1997.

[27] Tuan-Hao Hoang, R I (Bob) McKay, Daryl Essam, and Xuan Hoai Nguyen, Learning General Solutions through Multiple Evaluations during Development, ICES'2008, LNCS, Springer-Verlag, 201-212, 2008.

[28] M. Oltean and L. Diosan , "An autonomous GP-based system for regression and classification problems" , Applied Soft Computing. , vol. 9 , no. 1 , pp.49-60 , 2009

[29] I. D. Falco , E. Tarantino , A. D. Cioppa and F. Fontanella , "An innovative approach to genetic programming-based clustering"

[30] H. Guo , L. B. Jack and A. K. Nandi , "Feature generation using genetic programming with application to fault classification" , IEEE Trans. Syst., Man, Cybern. B , vol. 35 , no. 1 , pp.89 -99 , 2005

[31] Clodhna Tuite , Alexandros Agapitos , Michael O'Neill , Anthony Brabazon, Early stopping criteria to counteract overfitting in genetic programming, Proceedings of the 13th annual conference companion on Genetic and evolutionary computation, July 12-16, 2011, Dublin, Ireland

[32] Chris Gathercole, Peter Ross, Dynamic Training Subset Selection for Supervised Learning in Genetic Programming, Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, p.312-321, October 09-14, 1994

[33] Iba, H.: Bagging, boosting, and bloating in genetic programming. In: Proc. Of the Genetic and Evolutionary Computation Conference (GECCO 1999), pp. 1053-1060. Morgan Kaufmann, Orlando (1999)

[34] Ekaterina J. Vladislavleva, Guido F. Smits, and Dick Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. Trans. Evol. Comp, 13:333-349, April 2009.

[35] Byoung-Tak Zhang , Heinz Mhlenbein, Balancing accuracy and parsimony in genetic programming, Evolutionary Computation, v.3 n.1, p.17-38, Spring 1995 [doi:10.1162/evco.1995.3.1.17]

[36] Conte, M., Trautteur, G., De Falco, I., Della Cioppa, A., Tarantino, E., 1997. Genetic programming estimates of Kolmogorov complexity. In: Back, T. (Ed.), Proc. Seventh Int. Conf. on Genetic Algorithms. Morgan Kaufmann, San Francisco, CA, pp. 743-750.

[37] Li, M., Vit??anyi, P., 1993. In: An Introduction to Kolmogorov Complexity and Its Applications, Text and Monographs in Computer Science. Springer-Verlag, Berlin.

[38] A. Ekart and S. Z. Nemeth, ”Selection based on the Pareto nondomination criterion for controlling code growth in genetic programming”, Genet. Program. Evol. M., vol. 2, no. 1, pp. 61-73, 2001

[39] E. J. Vladislavleva, G. F. Smits, and D. den Hertog, ”Order of nonlinearity as a complexity measure for models generated by symbolic regression via Pareto genetic programming,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 333-349, Apr. 2009.

[40] Erwin Stinstra, Gijs Rennen, and Geert Teeuwen. Meta-modeling by symbolic regression and pareto simulated annealing. Internal report No. 2006-15, Tilburg University, Holland, March 2006

[41] Christian Gagn, Marc Schoenauer, Marc Parizeau, and Marco Tomassini. Genetic programming, validation sets, and parsimony pressure. In Pierre Collet, Marco Tomassini, Marc Ebner, Steven Gustafson, and Anik Ekrt, editors, Proceedings of the 9th European Conference on Genetic Programming, volume 3905 of Lecture Notes in Computer Science, pages 109-120, Budapest, Hungary, 10 - 12 April 2006. Springer

[42] Leonardo Trujillo, Sara Silva, Pierrick Legrand, and Leonardo Vanneschi. An empirical study of functional complexity as an indicator ofoverfitting in genetic programming. In Proceedings of the 14th European conference on Genetic programming, EuroGP’11, pages 262-273, Berlin, Heidelberg, 2011. Springer-Verlag.

[43] L. Vanneschi, M. Castelli, and S. Silva, ”Measuring bloat, overfitting and functional complexity in genetic programming,” in Proc. GECCO, Jul. 2010, pp. 877-884

[44] J.-M. Morvan, Generalized Curvatures. Berlin, Germany: Springer, 2008.

[45] J. Ni, R. H. Drieberg, and P. I. Rockett, ”The use of an analytic quotient operator in genetic programming,” IEEE Trans. Evol. Comput., vol. 17, no. 1, pp. 146-152, Feb. 2013.

[46] O. Giustolisi and D. A. Savic, ”Advances in data-driven analyses and modelling using EPR-MOGA,” J. Hydroinform., vol. 11, nos. 3-4, pp. 225-236, 2009.

[47] R. I. McKay, X. H. Nguyen, J. R. Cheney, M. Kim, N. Mori, and T. H. Hoang, ”Estimating the distribution and propagation of genetic programming building blocks through tree compression,” in Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference (GECCO ’09), pp. 1011-1018, ACM, 2009.

[48] McDermott, J., Galvan-Lopez, E., O’Neill, M.: A fine-grained view of GP locality with binary decision diagrams as ant phenotypes. In: Schaefer, R., Cotta, C., Kolodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 164-173. Springer, Heidelberg (2010)

[49] Trujillo, L., Legrand, P., Levy-Vehel, J.: The estimation of holderian regularity using genetic programming. In: Proceedings of GECCO 2010, pp. 861-868. ACM, New York (2010)

[50] Alonso, C., Montana, J., Puente, J., and Borges, C. (2009a). A new linear genetic programming approach based on straight lines programs: Some theoretical and experimental aspects. Int. Journal of Artificial Intelligence Tools, 18(5):757 - 781

[51] N. A. Lynch, ”Straight-line program length as a parameter for complexity analysis”, J. Comput. Syst. Sci. 21 (1980) 251-280

[52] Montana J L, Alonso C L, Borges C E, et al. Penalty functions for genetic programming algorithms[M]. Computational Science and Its Applications-ICCSA 2011. Springer Berlin Heidelberg, 2011: 550-562.

[53] V. Cherkassky and M. Yunkian. Comparison of Model Selection for Regression.Neural Computation, 15:1691-1714, 2003

[54] L. G. Valiant, A theory of the learnable, Communications of the ACM, v.27 n.11, p.1134-1142, Nov. 1984 [doi:10.1145/1968.1972]

[55] Alonso, C. L., Montana, J. L., Borges, C. E., 2013. Model complexity control in straight line program genetic programming. In: Rosa, A. C., Dourado,A., Correia, K. M., Filipe, J., Kacprzyk, J. (Eds.), IJCCI 2013 - Proceedings of the 5th International Joint Conference on Computational Intelligence, Vilamoura, Algarve, Portugal, 20-22 September, 2013. SciTePress, pp. 25-36.

[56] [Gabrielov and Vorobjov, 2004] A.N. Gabrielov and N. Vorobjov. Complexity of computations with pfaffian and noetherian functions. In Normal Forms, Bifurcations and Finiteness Problems in Differential Equations. Kluwer, 2004.

[57] J. Ni and P. Rockett, ”Training genetic programming classifiers by vicinal-risk minimization,” Genetic Programming and Evolvable Machines, vol. 16, no. 1, pp. 3-25, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10710-014-9222-4

[58] Vapnik, V.N.: The Nature of Statistical Learning Theory, 2nd edn. Springer, New York (2000)

[59] ”Tikhonov regularization as a complexity measure in multiobjective genetic programming,” IEEE Trans. Evolutionary Computation,vol. 19, no. 2, pp. 157-166, 2015.

[60] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities risk bounds and structural results. Journal of Machine Learning Research, 3:463-482, 2002.

[61] Timo Kotzing , Frank Neumann , Reto Spohel, PAC learning and genetic programming, Proceedings of the 13th annual conference on Genetic and evolutionary computation, July 12–6, 2011, Dublin, Ireland [doi¿10.1145/2001576.2001857]

[62] Amil, N.M., Bredeche, N., Gagne, C., Gelly, S., Schoenauer, M., Teytaud, O.: A statistical learning perspective of genetic programming. In: 12th European Conference on Genetic Programming (EuroGP 2009), pp. 327-338. Tubingen, Germany (2009)

[63] Schmidt, Michael, and Hod Lipson. ”Age-fitness pareto optimization.” Genetic Programming Theory and Practice VIII. Springer New York, 2011. 129-146