Improving Generalisation of Genetic Programming for Symbolic Regression with Structural Risk Minimisation

Qi Chen* Bing Xue* Lin Shang† Mengjie Zhang*
*School of Engineering and Computer Science
Victoria University of Wellington, PO Box 600, Wellington 6400, New Zealand
{Qi.Chen, Bing.Xue, Mengjie.Zhang}@ecs.vuw.ac.nz
†State Key Laboratory of Novel Software Technology
Nanjing University, Nanjing 210046, China
shanglin@nju.edu.cn

ABSTRACT

Generalisation is one of the most important performance measures for any learning algorithm, no exception to Genetic Programming (GP). A number of works have been devoted to improve the generalisation ability of GP for symbolic regression. Methods based on a reliable estimation of generalisation error of models during evolutionary process are a sensible choice to enhance the generalisation of GP. Structural risk minimisation (SRM), which is based on the VC dimension in the learning theory, provides a powerful framework for estimating the difference between the generalisation error and the empirical error. Despite its solid theoretical foundation and reliability, SRM has seldom been applied to GP. The most important reason is the difficulty in measuring the VC dimension of GP models/programs. This paper introduces SRM, which is based on an empirical method to measure the VC dimension of models, into GP to improve its generalisation performance for symbolic regression. The results of a set of experiments confirm that GP with SRM has a dramatical generalisation gain while evolving more compact/less complex models than standard GP. Further analysis also shows that in most cases, GP with SRM has better generalisation performance than GP with bias-variance decomposition, which is one of the state-of-the-art methods to control overfitting.

Keywords

Genetic Programming; Symbolic Regression; Generalisation; Structural Risk Minimisation; VC Dimension

1. INTRODUCTION

In machine learning, the task of a learning algorithm is to find a learnt machine that can minimise the expected prediction/test error. Generalisation error is the expected prediction error of a model over an unseen test set for a given training set [11], where on both the training set and the test set, the inputs X and the outputs Y are drawn randomly from their joint distribution $P(X,Y) = P(Y|X)P(X)$. The expected test error needs to be taken with respect to $P(X,Y)$. The problem is that, for real-world tasks, $P(X,Y)$ is generally unknown, thus a widely accepted way to minimise the generalisation error is through the empirical risk minimisation principle [11]. The principle consists of computing the errors of a set of candidate models over the training set and selecting the one which has a minimum training error among all the candidate models. However, the empirical risk is not alway a good indicator of the expected risk/generalisation error, such as when the number of training samples is small.

Genetic Programming (GP) [17] addresses the regression problems by means of evolving toward an optimal model structure along with the best fitted coefficients of the model. The evolutionary process, which is guided by purchasing models having lowest empirical error on the training set, can prone to severe overfitting when the number of instances is small or when the over-complex models are selected.

Structural risk minimisation (SRM) [27] is an approach from the learning theory. It provides a powerful framework to assess the generalisation ability of a learning machine by predicting the distance between the training error and the test error. SRM gives a definition of the upper bound of the generalisation error based on the empirical risk and the confidence interval, which measures the difference between the empirical risk and the real expected risk. The confidence interval of a set of models relies on two quantities — the size of the training set and the Vapnik-Chervonenkis dimension (VC-dimension) [26], which characterises the complexity of the model. Given a fix size of training set, the generalisation bound is characterised purely by the VC-dimension, thus this generalisation bound is also called VC generalisation bound or VC bound. The influence of the complexity of a model on its generalisation ability has been investigated and confirmed by a number of contributions [8, 25, 29]. A common agreement among these contributions is that, for a given number of training samples, models with higher complexity generally have bigger difference between the training error and the test error. Under SRM, the learning process aims to select the optimal model that minimises the upper bound on the generalisation error. The optimal model is expected to achieve a good tradeoff between the empirical risk and the model complexity.

Despite its accuracy in predicting the expected error and

GECCO '16, July 20-24, 2016, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/2908812.2908842

the solid theoretical foundation, SRM is seldom considered in GP for symbolic regression. One key problem is that it is difficult to estimate the VC-dimension of the models, since the theoretical estimation of the VC-dimension can be obtained only for a set of linear models. In [25], Vapnik et al. proposed an empirical approach to measure the VC-dimension of a learning machine for classification, which can be easily extended for regression.

Goals: This work aims to develop a new GP method to enhance the generalisation of GP for symbolic regression. This will be accomplished by introducing the SRM, which uses an empirical measurement of the VC-dimension of the candidate models, into GP. For the measurement of the VC-dimension of the candidate models in GP, we extend the approach in [25]. Specifically, this work has the research objectives as follows:

- how SRM influences the learning ability of GP in terms of the training performance,
- whether GP with SRM can have a dramatic generalisation gain for GP, and
- how GP with SRM influences the complexity of the model evolved by GP.

2. BACKGROUND

This section gives a brief introduction on the background of GP for symbolic regression and reviews some state-of-the-art development in the area of improving the generalisation on GP for symbolic regression. A brief introduction of the key concept used in this work — VC-dimension and structural risk minimisation from learning theory will also be given.

2.1 GP for Symbolic Regression

Symbolic regression is a kind of regression analysis. The task of it is to discover the relationship between the input variables and the output variables for a given dataset and express this relationship in a mathematical model.

When tackling symbolic regression problems, GP generally starts from a population of randomly generated models. This population is progressively evolved over generations in an iterative mechanism through evaluation, selection and mating until a predefined termination criterion has been met. GP's ability to address the problems automatically without any prior assumption of the form and the size of the solutions makes it a good approach to symbolic regression. It has been used to solving many symbolic regression problems [4, 16, 22].

2.2 Generalisation in GP for Symbolic Regression

Generalisation is one of the most important performance criteria for learning algorithms. In many areas in machine learning, generalisation has been treated as the most important issue for a long time [3, 10].

Despite the importance of generalisation in many other fields, it did not receive enough attention it deserves in GP for symbolic regression in the past for quite a long time. During those years, symbolic regression was treated as an optimisation problem, not much attention was payed to the regression performance of the models on unseen data. Until recent years (late 2000s), a plethora of approaches to improve the generalisation of GP for symbolic regression have been investigated in a number of works [5, 12, 13, 20, 23].

Many approaches control the complexity of models, thus to reduce overfitting and improve the generalisation performance of GP. In [13], a variance-based layered learning GP had been proposed. In the layered learning GP, models are trained from lower layers, which contain less complex training data, to higher layers that consist of more complex training data. These different training sets are all drawn from the original dataset using the variance of the target output as a complexity measure. During the evolutionary process, the model complexity of the population are controlled by means of discarding offspring which have greater complexity value than a fixed threshold in each layer. The variance of the output of the model on a set of validation is used as a measure of model complexity. Mousavi et al. [20] implemented a multiobjective GP to enhance the generalisation ability of GP. In their multiobjective GP, the first order derivative of the GP models is treated as a measure of model complexity. The RMSE of the first order derivative of the models is considered as anther objective in addition to the accuracy of the models. The results showed that their method can have better generalisation performance than standard GP on four symbolic regression problems. [29] introduced a new complexity measure named order of nonlinearity for GP to generate more smoother model, thus can have good generalisation gain. Order of nonlinearity is based on the degree of the Chebyshev polynominals approximation [18] of a certain accuracy. A Pareto GP is implemented by taking the control of the order of nonlinearity of models as a second objective. They claimed that their Pareto GP has extra generalisation capability. In [21, 24], an equalization operator was introduced to control the distribution of model length, thus as a method for controlling bloat and reduce overfitting. It is shown to be effective to promote the generalisation of GP.

2.3 Vapnik-Chervonenkis (VC) Dimension and Structural Risk Minimisation

The learning theory presents a general measure for the capacity/complexity of a learning machine, which is Vapnik-Chervonenkis dimension (VC-dimension) [26]. The original definition of the VC-dimension is for a set of indicator functions $\{I(X, \alpha)\}$, where X are the input vectors and α is a parameter vector. The VC-dimension h of a set of indicator functions $\{I(X, \alpha)\}$ is defined as the maximum number of vectors X_1, X_2, $\cdots$, X_h that can be seperated into two classes in all 2^h possible ways by the set of functions [28]. This definition is then extended to a set of real-value functions $\{R(X, \alpha)\}$. It is defined to be the VC-dimension of the indicator function $\{I(R(X, \alpha) - \beta > 0)\}$ [27], where β is the value of the range of R.

Using the VC-dimension, various estimations on the expected risk are constructed. Structural risk minimisation (SRM) is one of these approaches. In [9, 27], a practical form of the VC generalisation bound for regression problems is given. It is a derivative from the general analytical VC bound with appropriately selected values of theoretical parameters. For a regression model, the practical form of VC generalisation bound is defined as:

$$R_{exp}(h) \leq R_{emp}(h) \left(1 - \sqrt{p - p \ln p + \frac{\ln n}{2n}}\right)_{+}^{-1} \quad (1)$$

where the bound of the expected risk/error $R_{exp}(h)$ is char-

acterised by two values: $R_{emp}(h)$ which is the empirical risk with respect to the training error of the model, and the term $\left(1-\sqrt{p-p\ln p+\frac{\ln n}{2n}}\right)_+^{-1}$, which is interpreted as a confidence interval measuring the difference between the empirical risk and the real expected risk ("+" denotes the positive part of the term). In this term, $p = h/n$, h is the VC-dimension of the model, and n is the size of the training set. With a fix number of training samples, the confidence interval is characterised purely by the VC-dimension of the model. The higher h indicates the smaller value of the term $\left(1-\sqrt{p-p\ln p+\frac{\ln n}{2n}}\right)_+^{-1}$. Thus, given the same/similar values of $R_{exp}(h)$, the higher h means the larger generalisation bound. For detail derivation of Equation (1), readers are referred to [9, 27].

Under this definition, SRM aims to choose the model with optimal VC-dimension thus minimising the upper generalisation bound. This is accomplished by fitting the candidate models to a nested sequence that has increasing estimated expected risk $R_{exp1}<R_{exp2}<\cdots$ and choosing the model having the smallest value of the generalisation bound. The only problem of the SRM is the difficulty in estimating the VC-dimension of the models.

2.4 SRM in GP

Despite the solid theoretical foundation and reliability of SRM and VC-dimension, their application to GP has seldom been seen. The only works that can be found is in [7, 19], which introduced SRM as a new fitness function to GP. In comparison with GP using two classical statistical model selection methods Akaike Information Criterion [2] and Bayesian Information Criterion [6] as fitness functions, they demonstrated the advantage of SRM in promoting the generalisation ability of GP. A new simplified estimation of VC-dimension of the GP models are used in both works, which is counting the number of non-scalar nodes in a GP tree. The non-scalar nodes are nodes that not operated with $\{+,-\}$. They argued that the number of non-scalar nodes has exact relationship with the VC-dimension of the model. However, it is still an approximation of the VC-dimension of the models but not a direct measure that seems to be more reliable. This is a major difference between these two works and the work to be presented in this paper.

3. THE PROPOSED METHOD

This work aims to introduce the SRM into GP for symbolic regression. The proposed method is named *genetic programming with structural risk minimisation* (GPSRM). The major difference between GPSRM and standard GP is the evaluation method, i.e. fitness function.

3.1 Introducing SRM into GP

In GPSRM, the evaluation method is changed from the empirical risk, which is generally used in standard GP, to the SRM framework. The underlying assumption of GPSRM is straightforward. The VC generalisation bound defined by SRM provides an estimation of the test errors of the candidate models during the evolutionary process. SRM, as a new kind of evaluation criterion, is expected to guide the evolutionary process toward models which can achieve a good balance between the empirical risk and model complexity in terms of the VC-dimension. These models are expected to have smaller difference between the test error and the training error, thus can have better generalisation gain.

The empirical risk of standard GP for symbolic regression can take the form of various error between the target outputs and the outputs given by the candidate models, such as, *sum of absolute error* (SAE), *mean absolute error* (MAE), *mean square error* (MSE), *root mean square error* (RMSE). In this work, the RMSE is used to measure the empirical risk of models. The definition of VC generalisation bound of SRM is describe in Equation (1). According to the bound and the assumption that the bound is tight, the new fitness function in GPSRM is designed as Equation (2):

$$R_{exp}=\frac{RMSE}{\left(1-\sqrt{p-p\ln p+\frac{\ln n}{2n}}\right)_+} \tag{2}$$

where $RMSE$ is the empirical error over the training set, $p = h/n$, h is the VC-dimension of the model and n is the number of training samples. A key underlying component in the new fitness function of GPSRM is the method to estimate the VC-dimension of the models.

3.2 Measuring the VC-dimension of Models

The exact theoretical value of the VC-dimension can be obtained only for a set of linear models (there exists many theoretical definitions of the bound of VC-dimension, which are often very loose). However, during the evolutionary process of GP, a bunch of models consisting of linear and nonlinear ones are generated. Thus, it is difficult to have a theoretical estimation of the VC-dimension of GP models.

An empirical method to measure the VC-dimension of models was proposed in [25]. The key component of the method is to observe the empirical maximum derivation $\epsilon(n)$ of a model on two independent datasets with n instances, which is defined as:

$$\epsilon(n)=\frac{1}{n}\left(\sum_{j=1}^{n}\left|Y_{D_{1j}}-f(X_{D_{1j}},\alpha)\right|-\sum_{i=1}^{n}\left|Y_{D_{2i}}-f(X_{D_{2i}},\alpha)\right|\right) \tag{3}$$

where $\frac{1}{n}\sum_{j=1}^{n}|Y-f(X,\alpha)|$ denotes the frequency of error of the model on one dataset, D_1 and D_2 are two datasets having n instances.

According to the theoretical derivation of Vapnik et al. [25], $\epsilon(n)$ has an upper bound $\Phi(\frac{n}{h})$. The formula description of $\Phi(\frac{n}{h})$ is given as:

$$\Phi(\tfrac{n}{h})=\begin{cases}1 & \text{if } (\frac{n}{h}<0.5)\\ a\frac{\ln(2\frac{n}{h})+1}{\frac{n}{h}-k}\left(\sqrt{1+\frac{b\left(\frac{n}{h}-k\right)}{\ln(2\frac{n}{h})+1}}+1\right) & \text{otherwise.}\end{cases} \tag{4}$$

$\Phi(\frac{n}{h})$ is characterised by the number of training samples n and the VC-dimension of the model h. According to [25], the bound is tight. Thus, $\epsilon(n) \approx \Phi(\frac{n}{h})$ holds. As the VC-dimension h is the only unknown variable in the approximate equation, it can be measured by best fitting the empirical measure ϵ and theoretical value Φ on a group of two datasets having various n. The flowchart for describing the method is shown in Figure 1. The detail derivation of the definition of Φ and its parameters will not be presented in this work due to page limit. In short, the values of parameters a, b in Φ have been determined by fitting Φ to the empirically obtained maximum deviation of a linear model with known VC-dimension. The values are : $a = 0.16$ and $b = 1.2$. The value of k is determined from the continuity of Φ at the point $n/h = 0.5$, i.e. $\Phi(0.5) = 1$, thus the parameter

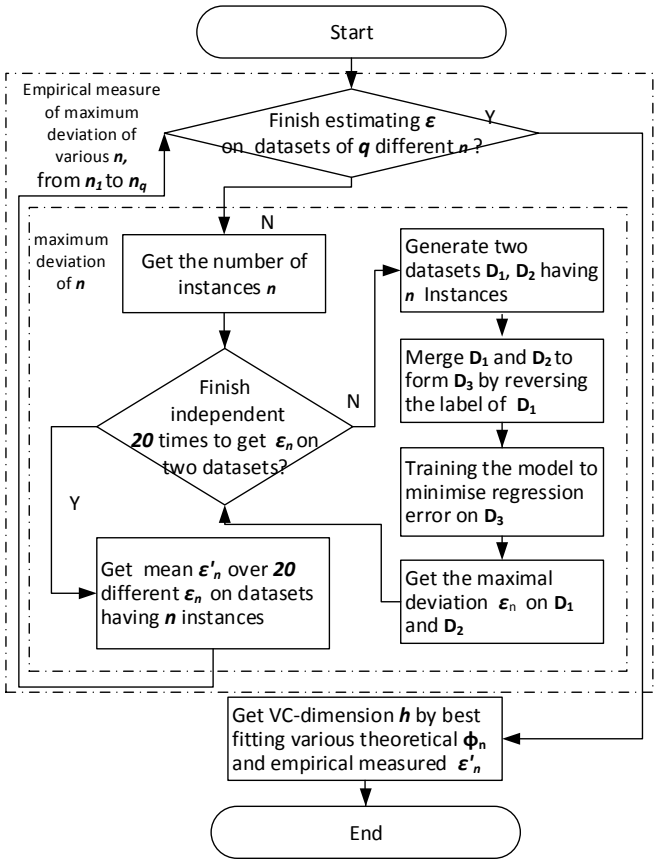

Figure 1: Flowchart of method to measure VC-dimension.

$k = 0.14928$. We have confirmed the validity of Φ with a number of linear models having known VC-dimensions. Here, a brief description of the procedure for measuring the VC-dimension of a model is given as follows (the model is denoted as $f(X, \alpha)$, where X are the input vectors and α is the parameter vector):

1. Generate two random datasets D_1, D_2 each having n instances. The input vectors, which have the same number of components as input variables in the model $f(X, \alpha)$, are drawn randomly within a uniform distribution over the interval $[-1, 1]$. The labels are generated according to the conditional probability distributions as $P(Y|X)=0.5$, $Y=0$ and $P(Y|X)=0.5$, $Y=1$.

2. To get the maximum deviation of the two datasets, merge the two datasets into a new dataset D_3 by reversing the labels of the first dataset D_1. The dataset D_3 will have $2n$ instances.

3. Training the model $f(X, \alpha)$ on dataset D_3 to get a minimum mean square error.

4. Calculate the $\epsilon(n)$ between the two datasets D_1 and D_2 according to Equation (3).

5. Repeat Steps 1-4 m times (m is set to 20, the same as that in [25]) independently, average the m different $\epsilon(n_m)$ to get an approximation of $\epsilon(n)$ which is the maximum derivation of the model $f(X, \alpha)$ on two random dataset of the size n.

This procedure is shown in the internal loop in Figure 1. In Step 3, *mini-batch gradient descent* is used to train the model to minimise the mean square error on D_3. The relationship between minimising the error on D_3 and getting the maximum derivation ϵ on D_1 and D_2 has been proved in Appendix 3 in [25]. The deviation is largely independent of the distribution $P(Y|X)$, thus it can be any other values. This is also proved in [25].

Then the whole procedure will repeat for q times on various training samples n within the range $0.5 \leq \frac{n}{h'} \leq 32$ (it is the setting from [25], where 0.5 is the start point of the definition of Φ and 32 is set to make sure that the range of $\frac{n}{h'}$ is big enough for various n), where h' is an initial guess of the VC-dimension of the model, generally it is the number of free parameters in the model. This process is shown in the outside loop in Figure 1.

After getting all the empirical values of maximum deviation on various training samples, $\epsilon(n_1), \ldots, \epsilon(n_q)$, the VC-dimension of the model can then be approximated by choosing the optimal value to parameter h, thus the best fit between the set of $\epsilon(n_j)$ and the function $\Phi(\frac{n}{h})$ can be obtained: $h=\arg\min \sum_{j=1}^{q} \left[\epsilon(n_j) - \Phi(n_j/h)\right]^2$.

4. EXPERIMENTAL SETUP

To demonstrates GP with SRM as a mechanism for enhancing the generalisation ability of GP, a set of experiments are performed. Standard GP is used as a baseline for comparison. We also compare the performance of GPSRM with GP with bias-variance decomposition (BVGP) [1, 15], which is one of the state-of-the-art approaches to control overfitting in GP. BVGP is based on introducing a statistical concept of bias/variance error decomposition to GP. Under this concept, the generalisation error is estimated by adding the square of bias error to the variance error. BVGP intends to obtain a model that can achieve a tradeoff between a good fit to the training data and less complexity. Since BVGP also aims to estimate the generalisation error during the evolutionary process, which is the same as GPSRM, this work will also compare the generalisation performance difference between BVGP and GPSRM.

4.1 Test Problems

There is no established suit of benchmarks, which is specially designed for testing the overfitting problem in GP. In this work, we will tackle a set of synthetic symbolic regression problems, which are the same as those in [1]. The eight target problems are defined as follows:

$$f_1(x)=e^{-x}x^3 cosx sinx(cosx sin^2 x-1) \quad (5)$$

$$f_2(x_1,x_2,x_3)=30\frac{(x_1-1)(x_3-1)}{x_2^2(x_1-10)} \quad (6)$$

$$f_3(x_1,x_2)=6sinx_1 cosx_2 \quad (7)$$

$$f_4(x_1,x_2)=\frac{(x_1-3)^4+(x_2-3)^3-(x_2-3)}{(x_2-2)^4+10} \quad (8)$$

$$f_5(x_1,x_2)=x_1x_2+sin((x_1-1)(x_2-1)) \quad (9)$$

$$f_6(x_1,x_2)=x_1^4-x_1^3+x_2^2/2-x_2 \quad (10)$$

$$f_7(x_1,x_2)=\frac{8}{2+x_1^2+x_2^2} \quad (11)$$

$$f_8(x_1,x_2)=x_1^3/5+x_2^3/2-x_2-x_1 \quad (12)$$

Table 1: Sampling strategy for the training and the test data.

The notation *rnd[a,b]* denotes the variable is randomly sampled from the interval *[a,b]*, while the notation *mesh([start:step:stop])* defines the set is sampled using regular intervals.

Benchmark	Training	Test
f_1	50 points x=rnd[0.05,10]	221 points x=mesh([-0.5:0.05:10.5])
f_2	50 points x_1, x_3=rnd[0.05,2] x_2=rnd[1,2]	2701 points x_1, x_3= mesh([-0.05:0.15:2.1]) x_2=mesh([0.95:0.1:2.05])
f_3	50 points x_1, x_2=rnd[0.1,5.9]	961 points x_1, x_2= mesh([0.05:0.02:6.05])
f_4	50 points x_1, x_2=rnd[0.05,6.05]	1157 points x_1, x_2= mesh([-0.25:0.2:6.35])
$f_5, f_6,$ f_7, f_8	20 points x_1, x_2=rnd[-3,3]	361,201 points x_1, x_2=mesh([-3:0.01:3])

Table 2: Parameters for GP, BVGP and GPSRM

parameter	Values
Population Size	500
Generations	50
Crossover Rate	0.9
Mutation Rate	0.1
Elitism Rate	0.01
Maximum Tree Depth	10
Initialisation	Ramped-Half&Half
Minimum Initialisation Depth	2
Maximum Initialisation Depth	6
Basic Function Set	$+$, $-$, $*$, %protected, $Square$, $Sqrt$, $Negative$
f_1	e^x, e^{-x}, $sinx$, $cosx$
f_3	e^x, e^{-x}
Percentage of Top Individuals — γ	20%

The detail of the sampling strategy of the training data and the test data can be found in Table 1. The first four benchmarks are selected from [29], as they seem to be difficult problems to GP. The rest four problems are chosen from [14]. The number of training points are deliberately designed to be a small value for all these eight problems, which is to simulate the real-world situation where the methods are more likely to prone to overfitting. The number of training points is *50* points for the first four problems and *20* points for the other four problems. These values are set to be smaller or the same as in [14, 29].

4.2 Parameter Settings

The parameter settings for standard GP (SGP), BVGP and GPSRM can be found in Table 2. Following the setting in [14, 29], for different benchmarks, the function set is different. For the same benchmark, all the three methods have the same setting.

For GPSRM, it does not necessary to measure the VC-dimension of the whole population, since the empirical risk (i.e. RMSE) difference between the top individuals and their worse counterparts are generally very big, the difference between the confidence interval, which ranges within the interval $[0, 1]$, is too small to work well. At the same time, during the whole evolutionary process, these worse individuals have very low probability to select to breed the new generation. Since the method to measure the VC-dimension of the models is very time consuming and due to the reasons mentioned before, to make GPSRM more efficient, we set a parameter γ to GPSRM. It denotes that GPSRM only measures the top γ percentage of individuals in the candidate population according to their empirical risk values. For the rest $1 - \gamma$ percentage of individuals, a random big value is given to the VC-dimension of these individuals, i.e. 100 in this work, since it is big enough to make the evolutionary process focusing on the comparison of estimated generalisation error between the top γ percentage of individuals.

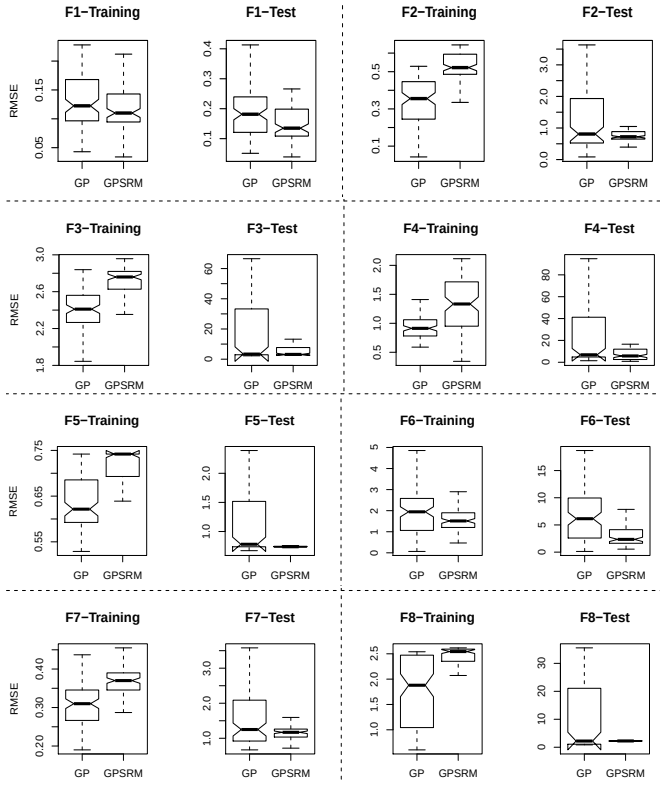

Figure 2: Distribution of RMSE of the 100 best-of-the-run individuals.

The experiments of each method have been conducted for 100 independent runs for each problem. Therefore, 2400 (i.e. 3*8*100) experiments have been run for the three methods on eight datasets and 4800 (i.e. 2400*2) training and test results are used here to discuss the learning ability and generalisation performance of the three methods.

5. RESULTS AND DISCUSSION

The experimental results of GPSRM, SGP and BVGP are presented and discussed in this section. The major comparison is presented between SGP and GPSRM. The results will be presented in terms of comparisons of RMSEs on the training sets (the fitness value of models in GPSRM is calculated using VC-bound, however for the comparison with SGP, the RMSE of models are also recored) and the test sets, the size of the 100 best-of-the-run models. We also compare the effect of GPSRM on promoting the generalisation performance of GP with the effect of BVGP. Wilcoxon test is used to evaluate the statistical significance of the difference on the RMSEs on both the training sets and the test sets. The significant level is set to be 0.05.

5.1 Overall Results

The results of the eight benchmarks are shown in Figure 2 (scales are different for the training and test RMSEs to save space), which display the distribution of RMSE of the 100 best-of-the-run individuals on the training sets and the

Table 3: Program size of 100 best-of-run models

Problem	SGP		GPSRM	
	mean	min	mean	min
f_1	29.45	13	27.88	10
f_2	44.05	14	16.28	4
f_3	35.07	3	23.45	3
f_4	39.39	11	16.35	3
f_5	27.99	3	7.33	3
f_6	38.03	16	27.7	9
f_7	24.95	11	16.68	7
f_8	32.51	12	18.25	7

test sets. As it shows, for six of the eight benchmarks SGP has better training performance than GPSRM (except for f_1 and f_6). The RMSE difference between SGP and GPSRM are statistical significant on these six benchmarks. On the other two benchmarks — f_1 and f_6, GPSRM outperforms SGP on the training data, which are also both statistically significant. In fact, the results on the training sets is not out of the expectation. On the training sets, SGP is expected to achieve better performance than GPSRM, since in GPSRM, the evolutionary process is guided by an additional objective implicitly, which is the lower VC-dimension of the models. This objective which measures the complexity of the models can sometimes conflict with the reduction of the training error, especially when the models overfit the training data.

Comparing to the training performance, GPSRM is rather expected to achieve better generalisation performance on the test sets. Figure 2 shows that GPSRM has a much smaller RMSE than standard GP on all the test sets of the benchmarks. The statistical significant results show that GPSRM has significant smaller RMSE than SGP on all the test sets. At the same time, the standard deviation of RMSE over the 100 runs in GPSRM is also much smaller than standard GP on the test sets, which means the generalisation of GPSRM outperforms standard GP on the test benchmarks in a very stable way. A further evidence of the better generalisation ability of GPSRM is provided by the difference between the training errors and the test errors. While GPSRM has a very similar RMSE value on the training data and the test data, the error values on the test set are much larger than on the training set for standard GP.

The program size of the 100 best-of-the-run models in terms of number of nodes are also examined. We assume that, although program/model size is not the same as model complexity, the comparison of the size of two groups of model can reflect the trend of model complexity to some extent. The mean and the minimum size of the evolved models can be found on Table 3. It can be observed that GPSRM produces much more compact individuals than standard GP on most of the problems, except for the problem f_1, where GPSRM has slightly smaller mean model size. GPSRM is expected to guide the evolutionary process toward models having a good tradeoff between the empirical error and the model complexity. When overfitting occurs in GP, SRM should have the ability to discard higher complexity models, while it should not preconverge to some over simple models when no overfitting occurs. This might be a reason why GPSRM has a comparative less reduction of the model size on problem f_1 than standard GP.

5.2 Evolution of the Test Errors

In Figure 3, the evolutionary plots on the test sets are reported per generation on both methods, which is based on the mean RMSE of the 100 best-of-the-generation models

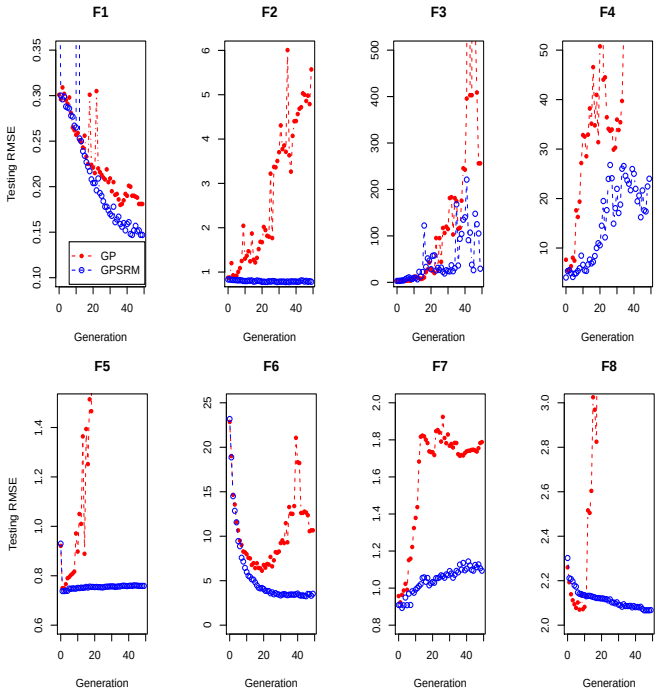

Figure 3: The evolution plots of the mean RMSE of 100 runs of the best individual on every generation on the *Test Sets*.

on the test data. The evolution plots confirm that GPSRM has a dramatic generalisation gain than SGP over all the test data of the considered benchmarks. It can be observed from the increasing test errors of GP over generations that, on most of the problems (except for f_1), standard GP, which guides the evolutionary process only by the empirical risk on the training data, overfits the training sets, while GPSRM generalise well on all these benchmarks. On six of the problems (except for the problem of f_1 and f_6), standard GP overfits very quickly on the first several generations, while GPSRM guiding the evolutionary process by the estimated generalisation error can eliminate overfitting effectively on three problem (i.e. f_2, f_5, f_8) and at least overfit less on the other three problems (i.e. f_3, f_4, f_7). The trend of less overfitting or eliminate overfitting to the training data of GPSRM confirms the expectation of the SRM's estimation ability of generalisation error.

On f_6 where standard GP does not overfit very dramatically and f_1 where standard GP does not overfit the training data at all, GPSRM can still performs better than standard GP on the test sets. This might due to GPSRM's less greedy to the lower training error and a better exploration of the search space which is guided by the implicit objective of the complexity of the models.

The trend on the test errors also confirms the reason of the different trends on the change of the model size. As mentioned before, the reduction of model complexity (model size) should be related to the trend of overfitting. For f_1, where no overfitting occurs, the reduction of model complexity of GPSRM is smaller than on other overfitted problems.

The results also indicate that GPSRM is better than using a validation set to control overfitting. Regarding the "test set" for SGP as the "validation set", then a "stopping point" for training should be picked at the lowest error on the vali-

Table 4: Example of best-of-the-run models for Problem f_6

	Method	Evolved Model	Simplified Model
1)	GPSRM	$(+(+(sqrt(0-(*(0-x_2)$ $(0-(0-x_2)))))(0-x_2))$ $(*(+(*x_1x_1)(*(0-x_1)$ $(*(0-x_1)$ $(0-x_1))))(0-x_1)))$	$x_1^4 - x_1^3$
	SGP	$(+(*(*x_1 8.75)x_1)$ $(*(*(sqrtx_1)(sqrtx_1))$ $(0-(+(*(*(\%0.905x_1)$ $(+x_1x_1))(+(*x_1 1.538)$ $(sqrt 17.205)))(\%$ $(*1.538(sqrtx_1))x_1)))))$	$11.54x_1^2 + 7.51x_1$ $+1.54/\sqrt{x_1}$
2)	GPSRM	$(+(+(*(*x_1x_1)$ $(+(*x_1x_1)(0-x_1)))$ $(0-x_2))(sqrtx_2))$	$x_1^4 - x_1^3 + \sqrt{x_1} - x_2$
	SGP	$(*(*x_1x_1)(+(\%(*$ $(*(+1.204(0-x_2))$ $(sqrt(\%x_1 8.20)))$ $(sqrt(\%(\%x_1 8.20)$ $(0-x_2))))(*x_1x_1))$ $(+(0-x_1)(*x_1x_1))))$	$x_1^4 - x_1^3 - \frac{(1.21-x_2)x_1}{8.2\sqrt{x_2}}$

Table 5: Generalisation performance (RMSE on the test sets) of standard GP, BVGP, and GPSRM

	SGP (median)[1]	BVGP (median)[1]	SGP (median±std)	BVGP (median±std)	GPSRM (median±std)	Significant Test
f_1	0.32	0.28	0.18±0.07	0.28±0.67	0.13±0.05	–
f_2	0.22	0.25	0.81±2.76	0.58±0.42	0.72±0.14	=
f_3	3.66	3.03	3.38±113.1	9.18±62.9	3.17±18.1	–
f_4	2.65	1.98	6.83±66.29	392.9±472.1	5.776±20.6	–
f_5	56.9	0.68	0.87±2.27	1.04±0.89	0.74±0.01	–
f_6	13.57	10.52	6.15±3.95	7.01±4.37	2.34±1.74	–
f_7	35.36	27.27	1.25±0.95	1.91±0.39	1.17±0.22	–
f_8	16.36	5.25	2.21±41.71	4.87±14.28	1.22±0.41	–

dation set for SGP. In fact, in the usual case, the true error on an independent test set should be slightly larger (at least not smaller) than on the validation set. As can be seen from the figure, except for problems f_5 and f_7, the test errors of GPSRM continues to decrease after the "stopping points", which suggests that GPSRM has the potential to achieve better performance than using a validation set to control overfitting. At the same time, compared with using a validation set, GPSRM is more suitable when the number of available data is small, since it does not require additional samples for validating the candidate models.

5.3 Analysis of the Evolved Programs

To study the behavioural difference between models evolved by both methods, as an example, two groups of best-of-the-run models on the f_6 problem are given in Table 4 for each method. The target model is shown in Equation (10). Each group of evolved models are the results of the same run of the two methods. To make the study of the behaviour of the models easier, the mathematical simplified form of the models are also provided. From the form of original models, it can be observed that the model produced by GPSRM is less complex than standard GP. The simplified models, which shows the behaviours of the models, indicates that the behaviour of the models evolved by GPSRM are more similar to the target model, although none of the evolved models has exactly the same form as the target model.

5.4 Comparison with BVGP

The results of the comparison of the generalisation ability between GPSRM and BVGP can be found in Table 5. The first two columns of results are taken directly from [1]. These are the best median generalisation error of BVGP under different tuning parameters in [1]. Since the parameter settings in [1] is not very common (like extremely large maximum tree depth and the higher mutation rate but lower crossover rate), the training samples in this work are different from [1], and [1] also used the validation sets that contain the same number of samples as the training sets to select the best-of-the-run models among a group of best-of-the generation models, this work tries to re-implement the bias-variance decomposition method without using a validation set. The purely comparsion between the generalisation performance of the two methods under the same and more common parameter settings are presented in this work.

The results of this work are shown in the 3rd to 5th columns in Table 5. The last column shows the statistical significant test result (Wilcoxon test is used, with a significant level of 0.05). While "- (+)" means BVGP has worse (better) generalisation performance than GPSRM, "=" means no significant difference. It can be observed that on seven of the eight problems, under the setting of this work, GPSRM performs better generalisation than BVGP. Even using the best results from [1], GPSRM still outperform BVGP on four benchmarks and has similar performance on three benchmarks of the eight benchmarks. This can confirm the advance generalisation of GPSRM over BVGP to some extent. Another advantage of GPSRM is that, it does not require any tuning parameter (for the only parameter γ is optional and also has a fix value), while BVGP needs to tune at least two parameters to achieve a good balance between bias error and variance error.

6. CONCLUSIONS

This work developed a new GP method — genetic programming with structural risk minimisation (GPSRM) by introducing SRM into GP to estimate the difference between the generalisation error and the empirical error. The goal of GPSRM is to improve the generalisation ability of GP for symbolic regression. A set of experiments have been conducted to investigate the influence of SRM on the learning and generalisation ability of GP on eight synthetic symbolic regression problems.

The results show that GPSRM has huge generalisation gain than standard GP and BVGP on the considered problems. This dramatical generalisation improvement depends on the accurate estimation of generalisation error during the evolutionary process of GP and not only purchasing lower empirical/training errors, which might lead to a better exploration of GP. Furthermore, the size of the models evolved by GPSRM are generally much smaller than standard GP.

However, the expensive computational cost of GPSRM and uniform setting of the parameters in measuring the VC-dimension are problems that need to be addressed in the following work. At the same time, the effectiveness of GPSRM on improving generalisation has not been compared with some other multi-objective GP methods, like GP with order of nonlinearity in [29] and GP with first order derivate of the model in [20]. This will also be part of our future work. We also plan to introduce some other model selection approaches like Akaike Information Criterion [2] and Bayesian Information Criterion [6] from statistical learning theory into GP and compare them with GPSRM on improving the generalisation performance.

7. ACKNOWLEDGMENTS

This work was supported in part by the Marsden Fund of the New Zealand Government under contract VUW1209, administrated by the Royal Society of New Zealand, and the University Research Fund (210375/3557, 209861/3580) at Victoria University of Wellington in New Zealand, and State Key Laboratory for Novel Software Technology (Overseas Collaboration Grant KFKT2014A28) at Nanjing University in China.

8. REFERENCES

[1] A. Agapitos, A. Brabazon, and M. O'Neill. Controlling overfitting in symbolic regression based on a bias/variance error decomposition. In *Parallel Problem Solving from Nature-PPSN XII*, pages 438–447. Springer, 2012.

[2] H. Akaike. Statistical predictor identification. *Annals of the Institute of Statistical Mathematics*, 22(1):203–217, 1970.

[3] S.-i. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[4] I. Arnaldo, K. Krawiec, and U.-M. O'Reilly. Multiple regression genetic programming. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 879–886. ACM, 2014.

[5] R. Azad and C. Ryan. Variance based selection to improve test set performance in genetic programming. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1315–1322. ACM, 2011.

[6] J. M. Bemardo and A. Smith. Bayesian theory. *Chichester: JohnWiley and Sons, Ltd*, 1994.

[7] C. E. Borges, C. L. Alonso, and J. L. Montaña. Model selection in genetic programming. In *Proceedings of the 12th annual conference on genetic and evolutionary computation*, pages 985–986. ACM, 2010.

[8] V. Cherkassky and Y. Ma. Comparison of model selection for regression. *Neural computation*, 15(7):1691–1714, 2003.

[9] V. Cherkassky and F. M. Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

[10] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[11] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[12] I. Gonçalves, S. Silva, and C. M. Fonseca. On the generalization ability of geometric semantic genetic programming. In *Genetic Programming*, pages 41–52. Springer, 2015.

[13] M. A. Haeri, M. M. Ebadzadeh, and G. Folino. Improving gp generalization: a variance-based layered learning approach. *Genetic Programming and Evolvable Machines*, 16(1):27–55, 2015.

[14] M. Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In *Genetic programming*, pages 70–82. Springer, 2003.

[15] M. Keijzer and V. Babovic. *Genetic programming, ensemble methods and the bias/variance tradeoff–Introductory investigations*. Springer, 2000.

[16] M. Kommenda, M. Affenzeller, G. Kronberger, B. Burlacu, and S. Winkler. Multi-population genetic programming with data migration for symbolic regression. In *Computational Intelligence and Efficiency in Engineering Systems*, pages 75–87. Springer, 2015.

[17] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[18] J. C. Mason and D. C. Handscomb. *Chebyshev polynomials*. CRC Press, 2002.

[19] J. L. Montaña, C. L. Alonso, C. E. Borges, and J. De La Dehesa. Penalty functions for genetic programming algorithms. In *Computational Science and Its Applications-ICCSA 2011*, pages 550–562. Springer, 2011.

[20] S. S. Mousavi Astarabadi and M. M. Ebadzadeh. Avoiding overfitting in symbolic regression using the first order derivative of GP trees. In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, pages 1441–1442. ACM, 2015.

[21] S. Silva, S. Dignum, and L. Vanneschi. Operator equalisation for bloat free genetic programming and a survey of bloat control methods. *Genetic Programming and Evolvable Machines*, 13(2):197–238, 2012.

[22] N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene, and P. Demeester. Constructing a no-reference h. 264/avc bitstream-based video quality metric using genetic programming-based symbolic regression. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(8):1322–1333, 2013.

[23] N. Q. Uy, N. T. Hien, N. X. Hoai, and M. O'Neill. Improving the generalisation ability of genetic programming with semantic similarity based crossover. In *Genetic Programming*, pages 184–195. Springer, 2010.

[24] L. Vanneschi, M. Castelli, and S. Silva. Measuring bloat, overfitting and functional complexity in genetic programming. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 877–884. ACM, 2010.

[25] V. Vapnik, E. Levin, and Y. Le Cun. Measuring the vc-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

[26] V. N. Vapnik and S. Kotz. *Estimation of dependences based on empirical data*, volume 40. Springer-verlag New York, 1982.

[27] V. N. Vapnik and V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[28] V. N. Vladimir and V. Vapnik. The nature of statistical learning theory, 1995.

[29] 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. *Evolutionary Computation, IEEE Transactions on*, 13(2):333–349, 2009.