Combining Conformal Prediction and Genetic Programming for Symbolic Interval Regression

Pham Thi Thuong
IT Department, University of Information and Communication Technology,
Thainguyen, Vietnam
ptthuong@ictu.edu.vn

Nguyen Xuan Hoai
HANU IT R&D Center,
Hanoi University,
Hanoi, Vietnam
nxhoai@hanu.edu.vn

Xin Yao
Department of Computer Science and Engineering,
Southern University of Science and Technology,
Shenzhen, China
xiny@sustc.edu.cn

ABSTRACT

Symbolic regression has been one of the main learning domains for Genetic Programming. However, most work so far on using genetic programming for symbolic regression only focus on point prediction. The problem of symbolic interval regression is for each input to find a prediction interval containing the output with a given statistical confidence. This problem is important for many risk-sensitive domains (such as in medical and financial applications). In this paper, we propose the combination of conformal prediction and genetic programming for solving the problem of symbolic interval regression. We study two approaches called black-box conformal prediction genetic programming (black-box CPGP) and white-box conformal prediction genetic programming (white-box CPGP) on a number of benchmarks and previously used problems. We compare the performance of these approaches with two popular interval regressors in statistic and machine learning domains, namely, the linear quantile regression and quantile random forrest. The experimental results show that, on the two performance metrics, black-box CPGP is comparable to the linear quantile regression and not much worse than the quantile random forrest on validity and much better than them on efficiency.

Keywords

Genetic Programming; Quantile Regression; Linear Quantile Regression; Quantile Regression Forests; Conformal Prediction; Interval Prediction; Symbolic Regression.

1. INTRODUCTION

Machine learning problems have constituted a popular application domain for Genetic Programming (GP). In particular, the symbolic regression is one of the most studied problems. The task for GP in this problem is to evolve a regressor in symbolic form given a training set of samples. Regression has traditionally been long studied in statistics and machine learning with numerous real-world applications. However, almost all of works on using GP to solve the symbolic regression problem only focus on point prediction that is to find a prediction point estimate Y for given X. In many risk-sensitive application domains such as biometrics, medicine, finance, reliability, etc point prediction would not be enough and interval prediction is needed. The problem of interval regression is to find a prediction interval covering the unknown value of Y (for a given X) with a specified probability (confidence level). To the best of our knowledge, there has almost been no general and systematic research on using GP for the task of symbolic interval regression.

Perhaps, using GP to evolve symbolic interval regressors has only been reported in [13] and [4]. In the former, Sánchez proposed an interval arithmetic-based GA-P model that uses interval arithmetic to solve the problem of interval prediction. However, the system based on the idea of combining interval arithmetic, genetic programming, and genetic algorithm is rather complicated and ad-hoc (with many components/parameters to be tuned). For instance, it must manually tune the parameters ϵ_1 and ϵ_2 to estimate the probability that Y falls in the prediction interval. Moreover, how to choose the number of intervals and their initializations (as well as genetic operators for them) for a given problem are not clear from this work. We argue that this limits the applicability of interval GA-P as well as makes their experimental results hard to be replicated. In the later work, Keijzer proposed the use of interval arithmetic, instead of using protected operators, to prevent peculiarities produced by some arithmetic operators (such as division by zero) used in GP expression trees. The output of each expression tree could be an interval but there is no statistical reliability (confident level) attached to it.

In this paper, we investigate a simpler and more general approach for building GP-based interval regressors. Our solution is based on combining Conformal Prediction (CP) and GP that can give the predictive range with statistical reliability for each given input X. In particular, we propose two methods called black-box conformal prediction GP (black-box CPGP) and white-box conformal prediction GP (white-box CPGP). We experimentally test our proposed methods on a number of benchmarks and previously used problems. We compare the performance of these methods with two well known range/interval prediction methods in statistics and machine learning domains, namely, the linear quantile regression(LQR) and the quantile random forrest (QRF).

GECCO '17, July 15-19, 2017, Berlin, Germany

ACM. ISBN 978-1-4503-4920-8/17/07...$15.00

DOI: http://dx.doi.org/10.1145/3071178.3071280

The remainder of the paper is organized as follows. Section 2 presents the background and related work including Quantile Regression and Conformal Prediction. Our proposed methods, black-box CPGP and white-box CPGP, are detailed in Section 3. Section 4 describes the settings of the experiments. The experimental results and analysis are given in section 5. Finally, section 6 summarize the paper and highlights some future work.

2. BACKGROUND AND RELATED WORK

In this section, we first give the definition of the interval symbolic regression, then the related problem, namely, quantile regression, is defined. Next, Linear Quantile Regression and Quantile Random Forrest, two well-known parametric and nonparametric quantile regression methods are described. Finally, the basic idea of conformal prediction is given.

2.1 Interval Symbolic Regression

The definition of the interval symbolic regression problem is adapted from [1, 15] as follows:

DEFINITION 2.1. *Let $Z = X \times Y$ be the sample space, where X is the predictor space of p-dimensions, Y is the real-value response space. Given $D = \{z_1, ..., z_n\}$ is the set of samples from Z, where $z_i = (x_i, y_i)$, find a function that maps X to a set (interval) Γ^{ϵ} that contains the unknown values of Y with a given probability $(1 - \epsilon)$, where $\epsilon \in [0, 1]$ is a significance level. Γ^{ϵ} said to be valid at a significance level ϵ, or valid with the statistical reliability $(1 - \epsilon)100\%$.*

2.2 Quantile Regression Problem

Quantile regression aims to estimate the conditional quantiles from a sample (training data) set D as follow [5, 2]: Given Y with the cumulative distribution function (CDF):

$$F_Y(y|X = x) = Prob(Y \leq y|X = x) \tag{1}$$

The quantile function is defined as the inverse of CDF:

$$Q_{Y|X=x}(\theta) = inf\{y : F_Y(y|X = x) \geq \theta\} \tag{2}$$

where $\theta \in [0, 1]$. Quantile regression is nicely linked to an optimization problem based on the observation that finding quantiles of a distribution could be casted as optimization (substituting order with optimization). The quantile q_θ is the solution of the following optimization problem [5]:

$$\min_c [E(\rho_\theta(Y - c))] \tag{3}$$

or for empirical quantiles on the sample set D of size n,

$$\min_c \sum_{i=1}^{n} \rho_\theta(y_i - c) \tag{4}$$

where, $\rho_\theta(.) = [(1 - \theta)I(y \leq 0) + \theta I(y > 0)]|y|$ is the loss function, $\theta \in [0, 1]$. For quantile regression problem, in the place of c, we need to find a function f of X that minimizes the expectation (or the sum for the empirical case).

2.3 Linear Quantile Regression (LQR)

Linear quantile regression is one of the most popular parametric quantile regression techniques in statistics. Given the sample set D as in 2.1, the model for linear quantile regression is defined as follows:

$$y = X\beta(\theta) + \epsilon \tag{5}$$

where, $\beta(\theta)$ is the vector of parameters for the generic conditional quantile θ and ϵ is the vector of stochastic errors. The parameters $\beta(\theta)$ are estimated from D by solving the following linear programming problem [5]:

$$\min_{\beta(\theta)} \sum_{i=1}^{n} \rho_\theta(y_i - x_i^T \beta(\theta)) \tag{6}$$

Using the estimated quantile function, the 95% prediction interval for the value of Y is given as follow:

$$\Gamma^{0.05} = [Q_{Y|X=x}(0.025), Q_{Y|X=x}(0.975)] \tag{7}$$

It is noted that the width of $\Gamma^{0.05}$ is dependent on x.

2.4 Quantile Regression Forest (QRF)

Quantile Regression Forests proposed by Nicolai Meinshausen [11] is considered as a common non-parametric quantile regression method in machine learning and statistics. It is a tree-based ensemble technique for estimating conditional quantiles. QRF grows k trees in a similar way to regression random forests. However, at each leaf node, it keeps all Y values (as an empirical distribution of Y values), instead of only the mean of Y as in normal quantile random forrest (more details can be seen in [11, 12]). Given an input $X = x$, we can find the leaf nodes from all k trees where X falls and the set of Y_i contained in these leaves. For each Y_i, a corresponding weights w_i is calculated as:

$$\omega_i(x) = \frac{1}{K} \sum_{k=1}^{K} \omega_i(x, \gamma_k) \tag{8}$$

where, $\omega_i(x, \gamma_k) = \frac{1}{N(t)}$ is the weight of tree γ_k, $N(t)$ is the number of cases in $l(x, \gamma_k, t)$ - the leaf t of the tree T_k that contains x. Then, the conditional distribution function of Y given X is estimated by

$$\widehat{F}_Y(y|X = x) = \sum_{i=1}^{n} \omega_i(x) I(Y_i \leq y) \tag{9}$$

where $I(.)$ is the indicator function. Given a probability θ, we can estimate the quantile $Q_{Y|X=x}(\theta)$ as:

$$\widehat{Q}_{Y|X=x}(\theta) = inf\{y : \widehat{F}_Y(y|X = x) \geq \theta\} \tag{10}$$

The prediction interval,

$$\Gamma^{0.05} = [Q_{Y|X=x}(.025), Q_{Y|X=x}(.975)] \tag{11}$$

made by QRF will contains the prediction Y with probability 95%.

2.5 Conformal Prediction (CP)

As shown in [15, 1], Conformal Prediction (CP) is the common measure to quantify the confidence of point predictors. It has a firm theoretical background developed over these years and has been applied successfully in many tasks in machine learning. Conformal Prediction could be used for both on-line and off-line learning, where the samples in D are only required to be exchangeable (weaker than the usual assumption of independently identically distributed in many

Algorithm 1: The Conformal Prediction Algorithm [1]

```
Set $z_{n+1} = (x_{n+1}, y)$;
for i=1 to n+1 do
    Set $\alpha_i = A(B, z_i)$;
Set $m = \#\{i = 1, .., n+1 | \alpha_i \geq \alpha_{n+1}\}$;
Set $p_y = \frac{m}{n+1}$;
if $p_y > \varepsilon$ then
    include $y$ in $\Gamma^{\varepsilon}(z_1, ..., z_n, x_{n+1})$ ;
```

statistical and machine learning techniques). Moreover, it has been proven that CP could produce a valid prediction interval with a given probability if D is sufficiently large [1]. The off-line version of CP is used in this paper.

Let D, ε and Γ^{ε} are as in section 2.1. We denote A as the non-conformity measure (e.g, the absolute error function); B is the bag of $z_1, ..., z_n$ except z_i; $z_{n+1} = (x_{n+1}, ?)$ is a new observation. Given a point predictor that gives a point prediction $\hat{y}$ for y_{n+1} given x_{n+1}. Assume that y_{n+1} is in fact equal to y, how Conformal Prediction defines Γ^{ε} is shown in **Algorithm 1**.

3. THE PROPOSED METHODS

In this section, we describe our proposed methods for combining conformal prediction with genetic programming to solve the problem of symbolic interval regression. In the first method, we use conformal prediction as the post-processing step for GP, which results in black-box Conformal Prediction GP (i.e GP is treated as black-box to conformal prediction). In the second method, we propose to embed conformal prediction into the fitness structure of GP individuals.

3.1 Computing Prediction Interval with Conformal Prediction

It is remarked that **Algorithm 1** in section 2.5 only gives us the testing condition for deciding to include y in Γ^{ε} that is valid at the $(1-\varepsilon)100\%$ level, but does not produce the Γ^{ε} itself. We observe that if y in Γ^{ε} then $p_y > \varepsilon$ must be true (it is called condition (1)). Then, Γ^{ε} could be constructed as in the following way:

Assume that A is the absolute error function, $A(B, z_i) = |y_i - \hat{y}_i|$, $\hat{y}_i$ is the obtained point prediction by GP, i.e., $\alpha_{n+1} = |y - \hat{y}_{n+1}|$ with $\hat{y}_{n+1}$ is the point prediction, given x_{n+1}. From condition (1), we have $\frac{m}{n+1} > \varepsilon$, or $m > \varepsilon * (n + 1)$. This requires at least k of α_i, $i = 1, .., n$ such that $\alpha_i \geq \alpha_{n+1}$, $k = \text{round}(\varepsilon * (n+1))$ (2). Let $E = \{\alpha_i | i = 1, .., n\}$, we sort E in decreasing order, call e_k is the k^{th} max element of the sorted E (3). From (2)&(3), we have $e_k \geq |y_{n+1} - \widehat{y}_{n+1}|$, or $\hat{y}_{n+1} - e_k \leq y_{n+1} \leq \hat{y}_{n+1} + e_k$. Given $e_k, \hat{y}_{n+1}$, so the Γ^{ε} is defined by the interval of $[\hat{y}_{n+1} - e_k, \hat{y}_{n+1} + e_k]$ (4). This will be the core part in calculating Γ^{ε} in our proposed methods.

3.2 Black-box CPGP

The leaning steps and evolution mechanism of black-box CPGP is just as standard genetic programming [7]. After a initial population is generated, the fitness evaluation are done for all individuals of the population. While the evolution progresses, genetic operators (crossover, and mutation) are performed on the chosen individuals to generate the new population. The best-of-the-run individual is chosen as the learned model at the end of the evolutionary process. After that, CP is used to compute the confidence interval Γ^{ε} given by this model in the way described in the previous subsection. The calculation of Γ^{ε} is independent of the learning process of GP, i.e it treats GP as a black-box learner and could be applied to any GP systems. The details of black-box CPGP are shown in Figure 1.

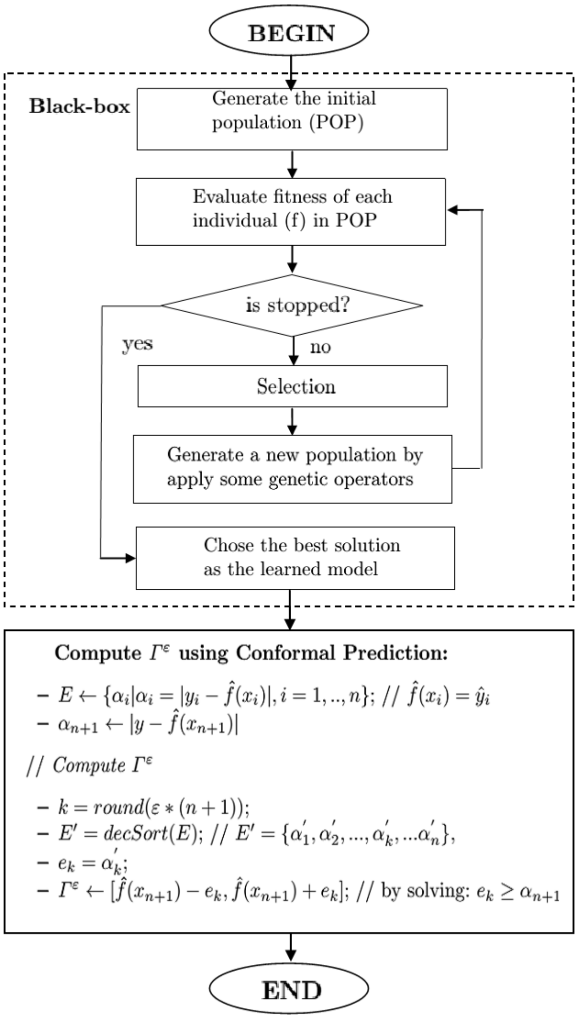
BEGIN
Black-box
Generate the initial population (POP)
Evaluate fitness of each individual (f) in POP
is stopped?
yes
no
Selection
Generate a new population by apply some genetic operators
Chose the best solution as the learned model
Compute Γ^{ε} using Conformal Prediction:
- $E \leftarrow \{\alpha_i | \alpha_i = |y_i - \hat{f}(x_i)|, i = 1, .., n\}$; // $\hat{f}(x_i) = \hat{y}_i$
- $\alpha_{n+1} \leftarrow |y - \hat{f}(x_{n+1})|$
// Compute Γ^{ε}
- $k = round(\varepsilon * (n+1))$;
- $E' = decSort(E)$; // $E' = \{\alpha_1^{'}, \alpha_2^{'}, ..., \alpha_k^{'}, ...\alpha_n^{'}\}$,
- $e_k = \alpha_k^{'}$;
- $\Gamma^{\varepsilon} \leftarrow [\hat{f}(x_{n+1}) - e_k, \hat{f}(x_{n+1}) + e_k]$; // by solving: $e_k \geq \alpha_{n+1}$
END

Figure 1: Black-box CPGP

By treating GP as a black-box learner, this method has some useful properties, such as (1) it allows to define how much variation of input can influence the output, i.e to measure the sensitivity of a GP learner; (2) it allows to indirectly analyze qualitatively describable aspects of the learned model such as generalization ability, bias, resistance to noise, and so on. As shown in [1], these properties are very useful for collaborative semi-supervised learning in which training data is the mixture of labeled and unlabeled, or on training data with perturbation. However, the analysis of these aspects of black-box CPGP is left for future work.

3.3 White-box CPGP

In the white-box approach, the objective is to embed conformal prediction into the evolutionary process of GP. Our

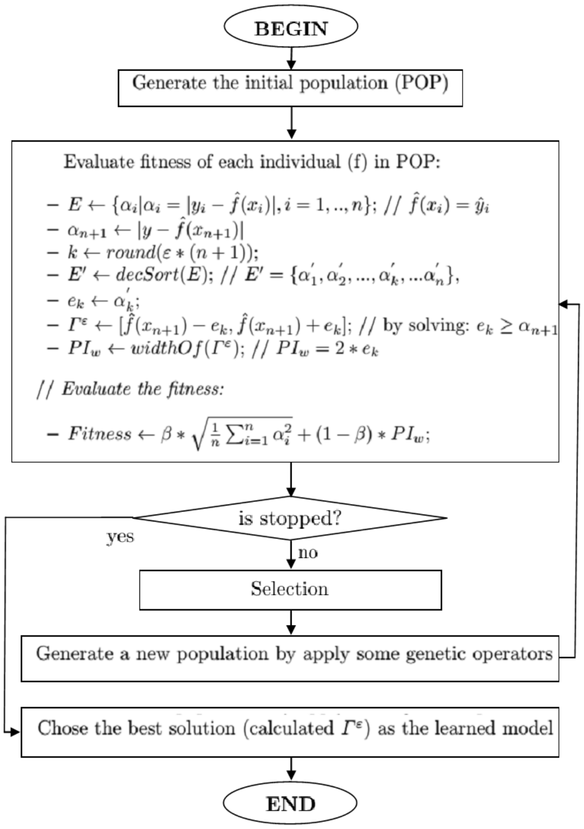
BEGIN
Generate the initial population (POP)
Evaluate fitness of each individual (f) in POP:
– $E \leftarrow \{\alpha_i | \alpha_i = |y_i - \hat{f}(x_i)|, i = 1, .., n\}$; // $\hat{f}(x_i) = \hat{y}_i$
– $\alpha_{n+1} \leftarrow |y - \hat{f}(x_{n+1})|$
– $k \leftarrow round(\varepsilon * (n+1))$;
– $E' \leftarrow decSort(E)$; // $E' = \{\alpha_1', \alpha_2', ..., \alpha_k', ...\alpha_n'\}$,
– $e_k \leftarrow \alpha_k'$;
– $\Gamma^\varepsilon \leftarrow [\hat{f}(x_{n+1}) - e_k, \hat{f}(x_{n+1}) + e_k]$; // by solving: $e_k \geq \alpha_{n+1}$
– $PI_w \leftarrow widthOf(\Gamma^\varepsilon)$; // $PI_w = 2 * e_k$
// Evaluate the fitness:
– $Fitness \leftarrow \beta * \sqrt{\frac{1}{n}\sum_{i=1}^{n} \alpha_i^2} + (1-\beta) * PI_w$;
is stopped?
yes
no
Selection
Generate a new population by apply some genetic operators
Chose the best solution (calculated Γ^ε) as the learned model
END

Figure 2: White-box CPGP

preliminary attempt in this paper is to allow the width of Γ^ε to be evolved in the evolutionary process of GP. This could be done by modifying the fitness function in GP so as it is the linear combination of training error, $Err(D)$, and the width of Γ^ε (called PI_w). The calculation of Γ^ε and its width is done as described in black-box CPGP. It is noted that PI_w depends on e_k but be independent of z_n. The fitness of each individual is defined as:

$$fitness = \beta * Err(D) + (1 - \beta) * PI_w \tag{12}$$

where: $\beta \in [0, 1]$ is a tunable parameter.

By embedding conformal prediction into the fitness structure of GP individuals, i.e. to incorporate the width of Γ^ε (or the efficiency) into the evolutionary process, white-box CPGP is capable of adapting Γ^ε dependent on X. It might also be useful for simultaneously optimizing the performance metrics (the efficiency and the validity) by combining them into the fitness function of the white-box CPGP. However, this possibility is left for future investigations.

4. EXPERIMENTAL SETTING

4.1 Test problems

To test our proposed methods, we experimented black-box and white-box CPGP on ten benchmark problems [10] and three UCI data sets that have been used to measure the performance of quantile random forrest in [11]. For comparison, we also run linear quantile regression (LQR) and quantile random forrest (QRF), the two most popular parametric and nonparametric quantile/interval regressors. For the benchmark problems, we follow [10] in generating the training and testing samples, with an exception that the size of the training sets are set uniformly as 200. For the UCI problems, the data are randomly divided into the training and testing samples with the ratio as $\langle$Train sample : Test sample$\rangle$ = $\langle 2 : 1 \rangle$, which is similar to the experiments in [11]. The detail information of the tested problems are shown in Table 1.

Table 1: Symbolic Regression Test Problems

UCIs:				
		#	**#**	**#**
ID	**Name**	**features**	**train**	**test**
U_1	Abalone	8	178	332
U_2	Housing	13	140	336
U_3	Ozone	12	122	244

Benchmarks:			
			#
ID	**Name**	**Definition**	**test**
B_1	Kei10	$x_1^{x_2}$	10000
B_2	Kei11	$x_1 x_2 + \sin((x_1 - 10)(x_2 - 1))$	361201
B_3	Kei12	$x_1^4 - x_1^3 + \frac{x_2^2}{2} - x_2$	361201
B_4	Kei13	$\sin(x_1)\cos(x_2)$	361201
B_5	Kei14	$\frac{8}{2+x_1^2+x_2^2}$	361201
B_6	Kei15	$\frac{x_1^3}{5} + \frac{x_2^3}{2} - x_2 - x_1$	361201
B_7	Vla1	$\frac{e^{(x_1-1)^2}}{1.2+(x_2-2.5)^2}$	194481
B_8	Vla5	$30\frac{(x_1-1)(x_3-1)}{x_2^2(x_1-10)}$	1200
B_9	Vla6	$6\sin(x_1)\cos(x_2)$	93636
B_{10}	Val8	$\frac{(x_1-3)^4+(x_2-3)^3-(x_2-3)}{(x_2-2)^4+10}$	1089

4.2 Performance Metrics

As shown in [1], the two most popular performance metrics for evaluating the quality of a range/interval predictor are validity (V) and efficiency (E).

4.2.1 *The validity*

Let Z, D, ε, Γ^ε and the example to be predicted (*test example*), z_{n+1} are as defined in section 2.5. Assume that the sequence $(z_1, ..., z_{n+1})$ of D is generated from a probability distribution P on Z^{n+1}. A prediction interval, Γ^ε, is said to be valid (how reliable they are) at a significance level ε if the probability (P) of $z^{n+1} \notin \Gamma^\varepsilon$ does not exceed ε. In other words, a Γ^ε is valid if it contains the truth at least $(1-\varepsilon)100\%$ of the time. The higher validity, the better range/interval predictor.

4.2.2 *The efficiency*

The efficiency is the width of the prediction interval (Γ^ε) where the smaller width is the better range/interval predictor.

4.3 System Configurations

The setting for the evolutionary parameters of black-box CPGP and white-box CPGP are shown in Table 2. These settings have often been used by GP researchers and practitioners [7]. In all experiments, the prediction decisions are fixed with 95% confidence (i.e the significant level $\varepsilon = 0.05$).

Table 2: Evolutionary parameters.

Parameters	Black/White-box CPGP
EA	Elitism, generational, expression tree
Function set	+, -, *, / (AQ)
Terminal set	Regression variables; one random constant $\in [0, 1]$
#Generations	151
Population size	500
Tour size	4
Tree creation	Ramped half-and-half (depths: 2 to 6)
Max. tree depth	15
Crossover rate	0.9
Mutation rate	0.1
#Runs	51
Fitness function	RMSE
β	Combinations with a step of 0.1 as (0.1,0.2, ..., 0.9)

We build our GP systems based on ECJ [9], a widely used evolutionary computation toolkit.

The code of QRF & LQR are available in the package quantreg of R, which can be freely downloaded from CRAN: http://cran.r-project.org. We run QRF & LQR with the default settings and the number of runs is the same as CPGP (51 runs). More detailed instruction for using LQR and QRF can be found in [8, 6, 14].

All of experiments were run on a PC with processor Intel(R) Core(TM) i5-4210 CPU @ 1.70 GHz; RAM (8.00GB) and 64-bit Operating System.

Table 3: p-values obtained by comparing the differences in the median of efficiency (E) and validity (V) by the black-box CPGP against white-box CPGP (with nine different settings includes *(0.1,0.9), (0.2,0.8), ..., (0.9,0.1)*) using the Mann-Whitney U-test on all problems from U_1 to B_{10}. Bold face indicates a confidence of at least 95% with (+) if the black-box CPGP is better than white-box CPGP $(\beta, (1-\beta))$ respectively, else with (-).

	(0.1,0.9)		*(0.2,0.8)*		*(0.3,0.7)*	
ID	**(E)**	**(V)**	**(E)**	**(V)**	**(E)**	**(V)**
U_1	**0-**	**0**+	**0**+	**0**+	**0-**	**0**+
U_2	0.32	**0**+	0.45	**0**+	0.06	**0.01**+
U_3	0.16	0.27	0.07	0.52	**0-**	0.73
B_1	0.17	**0**+	0.60	**0**+	0.16	**0**+
B_2	**0.01**+	0.13	0.10	0.42	**0.02**+	0.11
B_3	**0**+	**0**+	0.08	**0**+	0.06	**0**+
B_4	**0.05**+	**0**+	0.26	**0**+	0.07	**0**+
B_5	0.32	**0**+	0.95	**0**+	0.22	**0**+
B_6	**0.02**+	**0**+	**0.05-**	**0**+	**0.02**+	**0**+
B_7	**0**+	0.10	**0-**	**0.03-**	**0**+	0.63
B_8	**0.01**+	**0**+	**0-**	**0**+	0.12	**0**+
B_9	0.35	0.08	0.54	**0.05**+	0.98	**0.01**+
B_{10}	**0**+	0.73	**0-**	0.39	**0**+	0.49
	(0.4,0.6)		*(0.5,0.5)*		*(0.6,0.4)*	
U_1	**0-**	**0**+	**0-**	**0**+	**0-**	**0**+
U_2	0.15	**0**+	**0.01-**	**0.02**+	**0.04-**	**0.04**+
U_3	**0-**	0.62	**0-**	0.70	**0-**	0.77
B_1	0.91	**0**+	0.47	**0**+	0.27	**0**+
B_2	0.06	0.54	0.10	0.59	0.14	0.51
B_3	**0**+	**0**+	0.14	**0**+	0.49	**0**+
B_4	0.57	**0**+	0.87	**0**+	0.74	**0.01**+
B_5	0.47	**0**+	0.32	**0**+	0.13	**0**+
B_6	**0.03**+	**0**+	0.07	**0**+	0.75	**0.00**+
B_7	**0.01**+	0.66	0.07	0.84	0.22	0.25
B_8	**0.02**+	**0**+	0.25	**0**+	0.08	**0**+
B_9	0.82	0.13	0.40	**0.01**+	0.45	**0**+
B_{10}	**0**+	0.98	**0**+	0.92	**0**+	0.41
	(0.7,0.3)		*(0.8,0.2)*		*(0.9,0.1)*	
U_1	**0-**	**0**+	**0-**	**0**+	**0-**	**0**+
U_2	**0.05-**	**0**+	**0.01-**	**0.04**+	**0.02-**	**0.05**+
U_3	**0-**	0.98	**0-**	0.58	**0-**	0.94
B_1	0.34	**0**+	0.06	**0**+	0.31	0.07
B_2	0.41	0.57	0.69	0.31	0.66	0.08
B_3	0.73	**0**+	0.79	0.06	0.44	0.07
B_4	0.97	**0**+	0.95	0.08	0.16	**0**+
B_5	**0.02-**	**0**+	0.08	**0.01**+	0.41	0.12
B_6	0.75	**0**+	0.67	**0**+	0.55	**0**+
B_7	0.57	0.19	0.35	0.55	0.90	0.28
B_8	0.27	**0**+	0.14	**0.02**+	0.85	**0.02**+
B_9	0.51	**0**+	0.38	**0.02**+	0.49	0.06
B_{10}	**0**+	0.70	**0**+	0.23	0.35	0.21

5. RESULTS & ANALYSIS

5.1 Black-box CPGP&White-box CPGP comparison

In the first experiment, we compared the performance of black-box CPGP versus white-box CPGP (with 9 different settings of parameter β). Table 3 shows p-values obtained from the Mann-Whitney U-test for comparing the differences between black-box CPGP and white-box CPGP in the medians of efficiency and validity over all test problems [1]. In the table, the bold face indicates the statistical confidence of at least 95% with (+) if black-box CPGP is better than white-box CPGP or with (-) if white-box CPGP is better. From Table 3, it can be seen that black-box CPGP was usually better or equal to white-box CPGP on both validity and efficiency. For small values of β, black-box CPGP was better both in validity and efficiency (except for the case of (0.2,0.8)). When β is big (0.8 and 0.9), the validity of white-box CPGP tended to be similar to black-box CPGP, while

[1] Due to the limited space, the detailed median of validity and efficiency of the two systems are omitted

Table 4: p-value obtained by comparing the differences in the efficiency (E) and validity (V) of black-box CPGP against LQR & QRF using the Mann-Whitney U-test on all problems from U_1 to B_{10}. Bold face indicates a confidence of at least 95% with (+) if the black-box CPGP is better than LQR(QRF), else with (-).

	p value: (E)		**p value: (V)**	
ID	*LQR*	*QRF*	*LQR*	*QRF*
U_1	**0**(+)	**0**(+)	**0.0037**(+)	**0**(+)
U_2	0.6087	0.1060	**0**(+)	0.8962
U_3	**0**(+)	0.1262	**0**(-)	**0.0019**(-)
B_1	**0**(+)	**0**(+)	**0.0012**(+)	**0**(-)
B_2	**0**(+)	**0**(+)	**0.0265**(-)	**0**(-)
B_3	**0**(+)	**0**(+)	**0.0037**(+)	**0**(-)
B_4	**0**(+)	**0**(+)	0.8645	**0**(-)
B_5	**0**(+)	**0**(+)	**0.0105**(-)	**0**(-)
B_6	**0**(+)	**0**(+)	**0**(+)	**0**(-)
B_7	**0**(+)	**0**(+)	**0**(+)	**0**(-)
B_8	**0**(+)	**0**(+)	**0**(-)	**0**(-)
B_9	**0**(+)	**0**(+)	**0**(-)	**0**(-)
B_{10}	**0**(+)	**0**(+)	**0.0265**(-)	**0**(-)

the efficiency was better on UCI problems (however, the validity of black-box CPGP was better on those problems). Overall, these results show that tuning parameter β is not trivial task for white-box CPGP. An adaptive scheme for tuning β or a multi-objective approach for white-box CPGP seems necessary and we leave that for future work.

5.2 Black-box CPGP, LQR&QRF comparison

5.2.1 The validity & efficiency

Table 4 shows p-values obtained from the Mann-Whitney U-test for comparing the differences between black-box CPGP and LQR/QRF in the medians of efficiency and validity over all test problems. Table 5 contains the median values of all systems on all problems.

The results in Table 4 and Table 5 depict that black-box CPGP was comparable to LQR on validity being better on 6 problems and worse than on the other six (and square on U_2). However, the efficiency of black-box CPGP was better than LQR on almost all problems (12 out of 13). The validity of black-box CPGP was worse than QRF on almost all tested problems, even though it was comparable on the UCI problems and was not much worse than QRF in median values. The efficiency of black-box CPGP, however, was much better than QRF (often producing much narrower prediction intervals).

Figures 3 shows the predicted (x axis) versus the true value (y axis) for the prediction intervals produced by black-box CPGP, LQR, and QRF on the tests sets of some tested problems (using the the runs that achieved the median results). The diagonal line connects the points where the predicted values are equal to the true values. If a point is closer to this line, it represents a better point prediction. Green points indicate predictions that are within the predicted region, Γ^{ε}, (valid points), whereas the red points are predictions that are outside of this region.

From Figure 3, it can be seen that QRF produced the least number of red points. The number of red points produced by black-box CPGP and LQR are relatively equal. The prediction intervals of black-box CPGP is much smaller than LQR and QRF but more rigid (rather independent of X). We believe that the adaptation of the width of Γ^{ε} to X would be important for improving the validity of black-box CPGP.

Table 5: median obtained by comparing the differences in the efficiency (E) and validity (V) of black-box CPGP against LQR & QRF using the Mann-Whitney U-test on all problems from U_1 to B_{10}.

	median: (E)			**median: (V)**		
ID	*CPGP*	*LQR*	*QRF*	*CPGP*	*LQR*	*QRF*
U_1	8.56	9.26	9.14	0.93	0.92	0.84
U_2	17.73	17.79	17.33	0.95	0.92	0.95
U_3	236.9	214.7	232.6	0.89	0.95	0.90
B_1	0.12	0.35	0.43	0.94	0.93	0.95
B_2	1.93	12.27	8.87	0.92	0.93	0.96
B_3	6.01	131.9	94.66	0.94	0.93	0.99
B_4	2.24	10.46	8.42	0.93	0.92	0.95
B_5	0.25	2.83	1.50	0.93	0.94	0.97
B_6	2.70	7.10	5.52	0.93	0.89	0.98
B_7	0.12	0.46	0.34	0.85	0.79	0.92
B_8	0.42	2.30	2.12	0.83	0.85	0.92
B_9	7.20	10.78	8.56	0.92	0.94	0.96
B_{10}	1.78	5.00	3.04	0.86	0.87	0.94

Table 6: Fitness evaluation time (mean) of the best individual by black-box CPGP and the run-time (mean) by black-box CPGP, QRF & LQR over 51 runs on all problems from U_1 to B_{10}.

		Runtime		
ID	*Time (fitness)*	*CPGP*	*QRF*	*LQR*
U_1	0.5591	61.5690	0.1017	0.0127
U_2	0.3528	37.5800	0.1246	0.0190
U_3	0.2018	20.8526	0.0420	0.0132
B_1	0.4141	759.0627	0.9642	0.1953
B_2	0.4763	10087.9100	32.8770	5.3176
B_3	0.3801	9413.0030	34.5972	6.4849
B_4	0.5322	10982.6400	32.2264	5.5204
B_5	0.3981	9733.6550	31.9991	5.2519
B_6	0.5239	10793.7300	32.3510	5.2569
B_7	0.4796	6141.2850	16.9496	2.7553
B_8	0.4082	555.2053	0.1314	0.0223
B_9	0.6700	3684.9330	7.9389	1.3247
B_{10}	0.6287	774.4631	0.1158	0.0190

5.2.2 The computational cost

Table 6 shows the mean of fitness evaluation time of the best solution obtained by black-box CPGP and the mean of runtime of black-box CPGP, QRF and LQR over all runs. The time is measured in seconds. This table shows that high computational cost is a real issue of CPGP. As shown in [3], the most well-know problem in every application of GP is computational cost that makes the training time of GP can vary from seconds to days. This cost primarily due to the computation of the fitness of individuals which has to be repeatedly evaluated through the evolutionary process. So, the total of training time of GP is $O(n * m * k)$; where m is

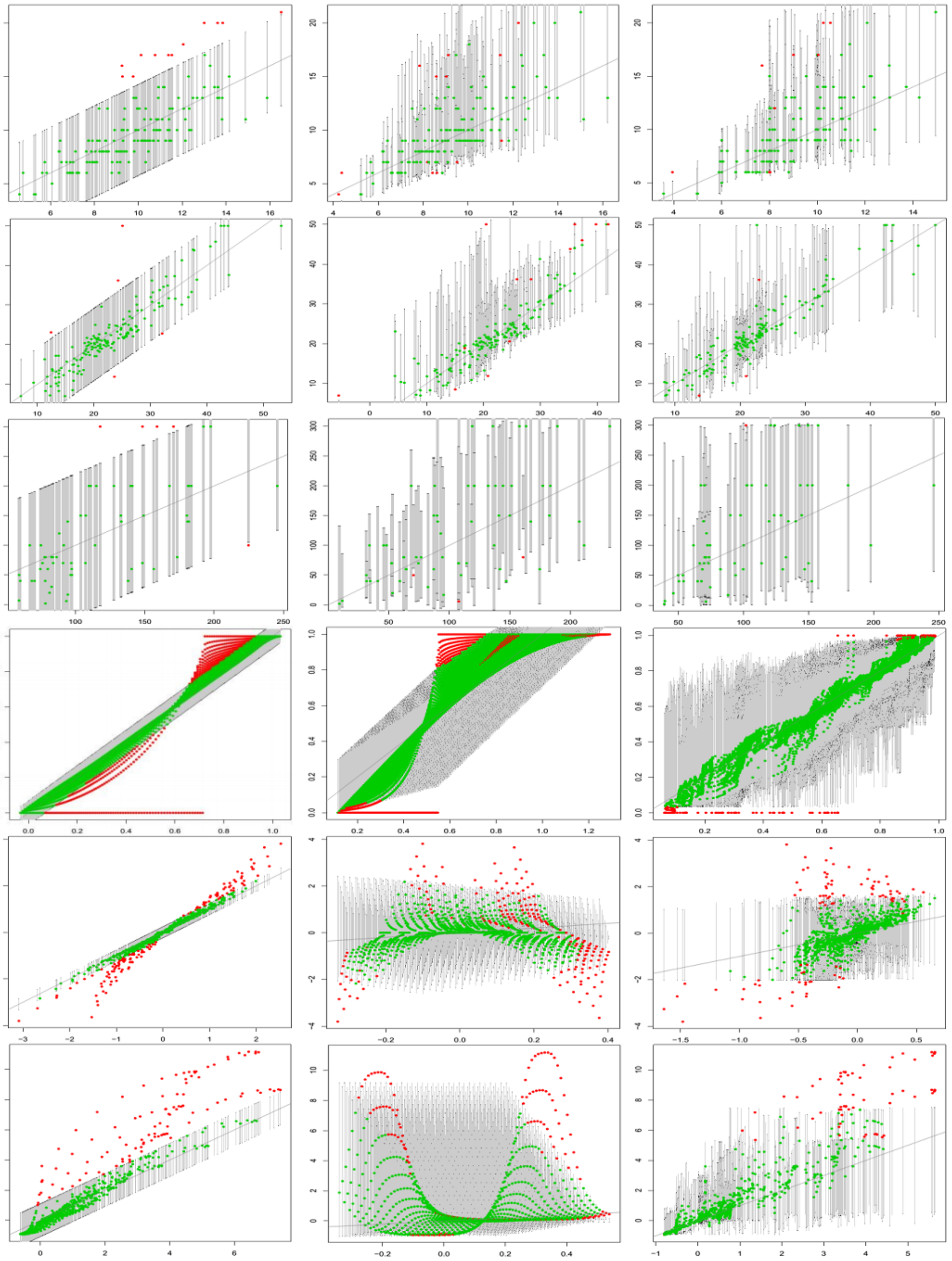

Figure 3: Predictive points & prediction interval on the problems: U_1, U_2, U_3, B_1, B_8 and B_{10} by the black-box CPGP (left), LQR(middle) and QRF (right), respectively. The solid line represent the set of points where the predicted values equal to the true values. The predictive values on the left are smaller than the true values while the predicted values are greater than the true values on the right.

number of generations, n is the size of population, and k is the average of size of individuals. Moreover, k can vary and be worse due to the bloating issue caused by introns or junk code in individuals.

6. CONCLUSIONS

In this research, we focused on solving the symbolic interval regression problem with GP. We argue that this problem is important when GP is used for risk sensitive learning domains. However, it has not been received much attention in the GP community. We proposed two approaches in making GP as a interval regressor by combining conformal prediction with GP. Our preliminary experimental results show that the approaches warrant further investigations in that our CPGP (black-box) is comparable to Linear Quantile Regression and not much worse than Quantile Random Forrest in terms of validity, but much better than them in terms of efficiency.

There are some open issues arisen from this paper. The first problem is how to adapt the prediction interval, Γ^{ε}, to be dependent on X in order to improve the validity of black-box CPGP. The second problem is to adaptively tune parameter β in white-box CPGP during the evolutionary process to improve the performance of white-box CPGP. Last but not least, reducing the computational time of CPGP is necessary and even vital if CPGP is to be used for online learning problems.

7. ACKNOWLEDGMENTS

This work is funded by The Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.09. The first author would like to thank the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen,China, for providing financial support and computing resources while she was doing this work at SUSTech. We thank Liyan Song for her useful comments and discussions related to this paper.

8. REFERENCES

[1] V. Balasubramanian, S.-S. Ho, and V. Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications.* Newnes, 2014.

[2] D. Cristina, F. Marilena, and V. Domenico. *Quantile Regression: Theory and Applications.* Wiley, 2014.

[3] P. G. Espejo, S. Ventura, and F. Herrera. A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 40(2):121–144, 2010.

[4] M. Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In *European Conference on Genetic Programming*, pages 70–82. Springer, 2003.

[5] R. Koenker. *Quantile regression.* Cambridge university press, 2005.

[6] R. Koenker. Quantile regression in r: A vignette. Technical report, 2015.

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

[8] M. Lipsitz, A. Belloni, V. Chernozhukov, and I. Fernández-Val. Nonparametric series quantile regression in r: A vignette. Technical report, 2015.

[9] S. Luke. The ecj owneŕs manual. Technical report, Department of Computer Science George Mason University, 2015.

[10] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. De Jong, et al. Genetic programming needs better benchmarks. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 791–798. ACM, 2012.

[11] N. Meinshausen. *Quantile regression forests. Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

[12] T.-T. Nguyen, J. Z. Huang, and T. T. Nguyen. Two-level quantile regression forests for bias correction in range prediction. *Machine Learning*, 101(1-3):325–343, 2015.

[13] L. Sánchez. Interval-valued ga-p algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1):64–72, 2000.

[14] L. Schiesser. Quantile regression forests - an r-vignette. Technical report, 2015.

[15] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421, 2008.