A divide and conquer method for symbolic regression [☆]

Changtong Luo[a,∗], Chen Chen[a,b], Zonglin Jiang[a,b]

[a]*State Key Laboratory of High Temperature Gas Dynamics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China*
[b]*School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing, 100049, China*

Abstract

Symbolic regression aims to find a function that best explains the relationship between independent variables and the objective value based on a given set of sample data. Genetic programming (GP) is usually considered as an appropriate method for the problem since it can optimize functional structure and coefficients simultaneously. However, the convergence speed of GP might be too slow for large scale problems that involve a large number of variables. Fortunately, in many applications, the target function is separable or partially separable. This feature motivated us to develop a new method, divide and conquer (D&C), for symbolic regression, in which the target function is divided into a number of sub-functions and the sub-functions are then determined by any of a GP algorithm. The separability is probed by a new proposed technique, Bi-Correlation test (BiCT). D&C powered GP has been tested on some real-world applications, and the study shows that D&C can help GP to get the target function much more rapidly.

Keywords: Mathematical modeling, Genetic programming, Symbolic regression, Artificial intelligence, Divide and conquer

1. Introduction

Symbolic regression (SR) is a data-driven modeling method which aims to find a function that best explains the relationship between independent variables and the objective value based on a given set of sample data (Schmidt and Lipson, 2009). Genetic programming (GP) is usually considered as a good candidate for SR since it does not impose a priori assumptions and can optimize function structure and coefficients simultaneously. However, the convergence speed of GP might be too slow for large scale problems that involve a large number of variables.

Many efforts have been devoted trying to improve the original GP (Koza, 2008) in several ways. Some works suggest replacing its tree-based method, with an integer string (Grammar Evolution) (O'Neill and Ryan, 2001), or a parse matrix (Parse-Matrix Evolution) (Luo and Zhang, 2012). These techniques can simplify the coding and decoding process but help little on improving the convergence speed. Some other works suggest confining its search space to generalized linear space, for example, Fast Function eXtraction (McConaghy, 2011), and Elite Bases Regression (Chen et al., 2017). These techniques can accelerate the convergence speed of GP, even by orders of magnitude. However, the speed is gained at the sacrifice of losing the generality, that is, the result might be only a linear approximation of the target function.

Fortunately, in many applications, the target function is separable or partially separable (see section 2 for definitions). For example, in gas dynamics (Anderson, 2006), the heat flux coefficient S_t of a flat plate could be formulated as

$$S_t = 2.274 \sin(\theta) \sqrt{\cos(\theta)} / \sqrt{Re_x}, \tag{1}$$

and the heat flux q_s at the stagnation point of a sphere as

$$q_s = 1.83 \times 10^{-4} v^3 \sqrt{\rho/R} (1 - h_w/h_s). \tag{2}$$

In equation (1), the two independent variables, θ and Re_x, are both separable. In equation (2), the first three variables v, ρ, and R are all separable, and the last two variables, h_w and h_s, are not separable, but their combination (h_w, h_s) is separable. The function in equation (2) is considered partially separable in this paper.

The feature of separability will be used in this paper to accelerate the optimization process of symbolic regression. Some basic concepts on function separability are defined in Section 2. Section 3 describes the overall

2

work flow of the proposed method, divide and conquer. Section 4 presents a special technique, bi-correlation test (BiCT), to determine the separability of a function. Numerical results are given in Section 5, and the concluding remarks are drawn in Section 6.

2. Basic concepts

The proposed method in this paper is based on a new concept referred to as partial separability. It has something in common with existing separability definitions such as reference (Berenguel et al., 2013) and (d'Avezac et al., 2011), but is not exactly the same. To make it clear and easy to understand, we begin with some illustrative examples. The functions as follows could all be regarded as partially separable:

$$z = 0.8 + 0.6 * (\boxed{u^2 + \cos(u)}) + \boxed{\sin(v + w) * (v - w)}; \tag{3}$$

$$z = 0.8 + 0.6 * (\boxed{u^2 + \cos(u)}) - \boxed{\sin(v + w) * (v - w)}; \tag{4}$$

$$z = 0.8 + 0.6 * (\boxed{u^2 + \cos(u)}) * \boxed{\sin(v + w) * (v - w)}; \tag{5}$$

where the boxed frames are used to indicate sub-functions, u is separable with respect to z, while v and w themselves are not separable, but their combination (v, w) is considered separable. A simple example of non-separable function is $f(\boldsymbol{x}) = \sin(x_1 + x_2 + x_3)$.

More precisely, the separability could be defined as follows.

Definition 1. *A scalar function with n continuous variables $f(\boldsymbol{x})$ ($f : \mathbb{R}^n \mapsto \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^n$) is said to be partially separable if and only if it can be rewritten as*

$$f(\boldsymbol{x}) = c_0 \otimes_1 \varphi_1(I_1 x) \otimes_2 \varphi_2(I_2 x) \otimes_3 \cdots \otimes_m \varphi_m(I_m x) \tag{6}$$

where the binary operator $\otimes_i$ could be plus $(+)$, minus $(-)$, times $(\times)$. I_i is a sub-matrix of the identity matrix, and $I_i \in \mathbb{R}^{n_i \times n}$. The set $\{ I_1, I_2, \cdots, I_m \}$ is a partition of the identity matrix $I \in \mathbb{R}^{n \times n}$, $\sum_{i=1}^{m} n_i = n$. The sub-function φ_i is a scalar function such that $\varphi_i : \mathbb{R}^{n_i} \mapsto \mathbb{R}$. Otherwise the function is said to be non-separable.

3

In this definition, the binary operator, division ($/$), is not included in $\otimes$ for simplicity. However, this does not affect much of its generality, since the sub-functions are not preset, and can be transformed as $\tilde{\varphi}_i(\cdot) = 1/\varphi_i(\cdot)$ if only $\varphi_i(\cdot) \neq 0$.

A special case is that all variables are separable, which could be defined as follows.

Definition 2. *A scalar function with n continuous variables $f(\boldsymbol{x})$ ($f : \mathbb{R}^n \mapsto \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^n$) is said to be completely separable if and only if it can be rewritten as equation (6) and $n_i = 1$ for all $i = 1, 2, \cdots, m$.*

3. Divide and conquer method

As above mentioned, many practical problems have the feature of separability. To make use of this feature to accelerate the optimization process of genetic programming, a new method, divide and conquer (D&C), is proposed. It works as follows.

First, a separability detection process is carried out to find out whether the concerned problem is separable (at least partial separable) or not. The variables are identified one by one, and then their combinations. Once it (or the variable combination) is identified as separable, a sub-function $\varphi_i(x_i)$ (or $\varphi_i(I_i\boldsymbol{x})$ for variable combinations) will be assigned. In this way, the structure of target function $f(\boldsymbol{x})$ could be divided into a set of sub-functions based on the separability information: $\varphi_i(I_i\boldsymbol{x})$, $i = 1, 2, \cdots, m$.

Then, the sub-functions $\varphi_i(I_i\boldsymbol{x})$ ($i = 1, 2, \cdots, m$) are optimized and determined one by one, using any of genetic programming algorithms, including classical GP, Grammatical Evolution (O'Neill and Ryan, 2001) and Parse-Matrix Evolution (Luo and Zhang, 2012). When optimizing one sub-function, variables not involved ($(I - I_i)\boldsymbol{x}$) are fixed as constants. That is, only a small number of variables ($I_i\boldsymbol{x}$, which is only a subset of $\{x_1, x_2, \cdots, x_n\}$) need be considered. This means the sub-function determination should be much easier than evolving the target function $f(\boldsymbol{x})$ directly.

For example, in Equation (3), Equation (4), or Equation (5), the sub-function $\varphi_1(u) = u^2 + \cos(u)$ (or $\varphi_2(v, w) = \sin(v + w) * (v - w)$) has less number of variables and complexity than the original function. Thus, optimizing them one by one is much easier for GP.

Finally, these sub-functions are properly combined to form the target function, which is referred to as a function recover process. This process
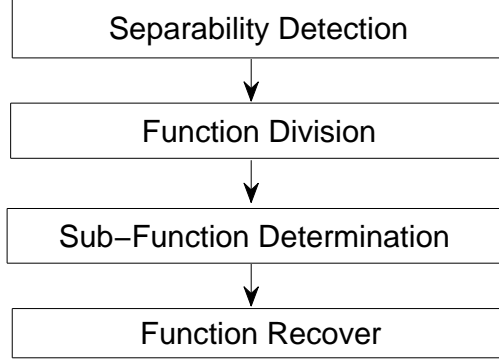
4

Figure 1: Work flow of divide and conquer for symbolic regression

is rather simple, and all traditional regression algorithms are qualified to accomplish this mission.

The work flow of D&C could be described in Figure 1.

4. Bi-correlation test

4.1. Description

The main idea of the proposed divide and conquer (D&C) method is to make use of the separability feature to simplify the search process. Therefore, the most important and fundamental step (the separability detection process in Fig. 1) is to determine whether the concerned problem is separable (at least partial separable) or not. To fulfill this task, a special technique, Bi-correlation test (BiCT), is provided in this section.

Consider independent variables $x_1, x_2, \cdots, x_n$ and the dependent f as n+1 random variables, and the known data as sample points. Recall that the linear relation and correlation coefficient of two random variables has the following relations.

Lemma 1. *The two random variables ξ and η are linearly related with correlation coefficient 1 (i.e., $\rho_{\xi\eta} = 1$) if and only if there exists two constants a, b ($b \neq 0$), such that $P\{\eta = a + b\xi\} = 1$.*

The correlation coefficient ρ could be estimated by the sample correlation

coefficient r, which is defined as follows.

$$r = \frac{1}{n-1} \sum_{i=1}^{N} \frac{(\xi_i - \bar{\xi})}{\sigma_\xi} \cdot \frac{(\eta_i - \bar{\eta})}{\sigma_\eta}$$

where N is the number of observations in the sample set, $\sum$ is the summation symbol, ξ_i is the ξ value for observation i, $\bar{\xi}$ is the sample mean of ξ, η_i is the η value for observation i, $\bar{\eta}$ is the sample mean of η, σ_ξ is the sample standard deviation of ξ, and σ_η is the sample standard deviation of η.

Only continuous model functions are considered in this paper. As a result, the conclusion of Lemma 1 could be simplified as follows.

- The two random variables $\boldsymbol{f}_A$ and $\boldsymbol{f}_B$ are linearly related ($\boldsymbol{f}_B = a + b\boldsymbol{f}_A$) if and only if the sample correlation coefficient $r = 1$ for any given sample set.

Studies shows that the functional separability defined in the above section (See equation 6) could be observed with random sampling and linear correlation techniques.

Without the loss of generality, a simple function with three variables ($f(\boldsymbol{x}) = f(x_1, x_2, x_3), x_i \in [a_i, b_i], i = 1, 2, 3$) is considered to illustrate the implementation of the bi-correlation test. To find out whether the first variable x_1 is separable, two correlation tests are needed.

The first correlation test is carried out as follows. A set of random sample points in $[a_1, b_1]$ are generated, then these points are extended to three dimensional space with the rest variables (x_2 and x_3) fixed to a point A. We get a vector of function values $\boldsymbol{f}^{(A)} = \boldsymbol{f}(x_1, A) = (f_1^A, f_2^A, \cdots, f_N^A)$, where N is the number of sample points. Then these points are extended to three dimensional space with fixed x_2 and x_3 to another point B, We get another vector $\boldsymbol{f}^{(B)} = \boldsymbol{f}(x_1, B) = (f_1^B, f_2^B, \cdots, f_N^B)$. It is obviously that the two vectors $f^{(A)}$ and $f^{(B)}$ will be linearly related if x_1 is separable. However, it is easy to show that this linear relation could NOT ensure its separability.

Then it comes to the second correlation test. Another set of random sample points in $[a_2, b_2] \times [a_3, b_3]$ are generated, then these points are extended to three dimensional space with the rest variable(s) (x_1 in this case) fixed to a point C, and get a vector $\boldsymbol{f}^{(C)} = \boldsymbol{f}(C, x_2, x_3)$. Similarly, another vector $\boldsymbol{f}^{(D)} = \boldsymbol{f}(D, x_2, x_3)$ is obtained. Again, the two vectors $\boldsymbol{f}^{(C)}$ and $\boldsymbol{f}^{(D)}$ needs to be linearly related to ensure the separability of x_1.

6

4.2. Proposition

Without the loss of generality, suppose we have a scalar function $f(\boldsymbol{x})$ with n continuous variables ($f : \mathbb{R}^n \mapsto \mathbb{R}$, $\boldsymbol{x} \in \Omega \subset \mathbb{R}^n$, and $\Omega = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$), and need to find out whether the first m variable combination $(x_1, x_2, \cdots, x_m)$ are separable. Let the matrix X_1 be a set of N random sample points from the subset $[a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_m, b_m] \subset \mathbb{R}^m$, and

$$
X_1 = \begin{bmatrix}
x_1^{(1)} & x_2^{(1)} & \cdots & x_m^{(1)} \\
x_1^{(2)} & x_2^{(2)} & \cdots & x_m^{(2)} \\
\cdots & \cdots & \cdots & \cdots \\
x_1^{(N)} & x_2^{(N)} & \cdots & x_m^{(N)}
\end{bmatrix}.
$$

The rest variables $x_{m+1}, x_{m+2}, \cdots, x_n$ are fixed to two given points A and B in the subset $[a_{m+1}, b_{m+1}] \times [a_{m+2}, b_{m+2}] \times \cdots \times [a_n, b_n] \subset \mathbb{R}^{n-m}$, i.e., $\boldsymbol{x}_A = (x_{m+1}^{(A)}, x_{m+2}^{(A)}, \cdots, x_n^{(A)})$, $\boldsymbol{x}_B = (x_{m+1}^{(B)}, x_{m+2}^{(B)}, \cdots, x_n^{(B)})$.

Let the matrix $X_2^{(A)} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix} \boldsymbol{x}_A = \begin{bmatrix}
x_{m+1}^{(A)} & x_{m+2}^{(A)} & \cdots & x_n^{(A)} \\
x_{m+1}^{(A)} & x_{m+2}^{(A)} & \cdots & x_n^{(A)} \\
\cdots & \cdots & \cdots & \cdots \\
x_{m+1}^{(A)} & x_{m+2}^{(A)} & \cdots & x_n^{(A)}
\end{bmatrix}$, and

$$
X_2^{(B)} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix} \boldsymbol{x}_B.
$$

Let the extended matrix $X_A = \begin{bmatrix} X_1 & X_2^{(A)} \end{bmatrix}$, and $X_B = \begin{bmatrix} X_1 & X_2^{(B)} \end{bmatrix}$. Let $\boldsymbol{f}_A$ be the vector of which the i-th element is the function value of the i-th row of matrix X_A, i.e., $\boldsymbol{f}_A = f(X_A)$, and $\boldsymbol{f}_B$ is similarly defined, $\boldsymbol{f}_B = f(X_B)$.

Lemma 2. *The two vectors $\boldsymbol{f}_A$ and $\boldsymbol{f}_B$ are linearly related if the function $f(\boldsymbol{x})$ is separable with respect to the first m variable combination $(x_1, x_2, \cdots, x_m)$.*

Proof. Since the first m variable combination $(x_1, x_2, \cdots, x_m)$ are separable, from definition 1, we have $f(\boldsymbol{x}) = \varphi_1(x_1, x_2, \cdots, x_m) \otimes \varphi_2(x_{m+1}, x_{m+2}, \cdots, x_n)$. Accordingly, the vector $\boldsymbol{f}_A = f(X_A) = \varphi_1(X_1) \otimes \varphi_2(X_2^A) = \varphi_1(X_1) \otimes k_A$, where $\otimes$ is a component-wise binary operation, and $k_A = \varphi_2(\boldsymbol{x}_A)$ is a scalar.

Similarly, the vector $\boldsymbol{f}_B = \varphi_1(X_1) \otimes k_B$. As a result,

$$\boldsymbol{f}_A = \begin{cases} k_A/k_B \cdot \boldsymbol{f}_B & \text{if } \otimes \text{ is times} \\ k_A - k_B + \boldsymbol{f}_B & \text{if } \otimes \text{ is plus} \\ k_B - k_A + \boldsymbol{f}_B & \text{if } \otimes \text{ is minus} \end{cases}$$

which means the two vectors $\boldsymbol{f}_A$ and $\boldsymbol{f}_B$ are linearly related. $\qquad\square$

On the other hand, if the first m variables are fixed to two given points C and D, and the rest of $n - m$ variables are randomly sampled. A similar proposition could be concluded as follows. Let

$$X_1^{(C)} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix} \begin{bmatrix} x_1^{(C)} & x_2^{(C)} & \cdots & x_m^{(C)} \end{bmatrix} = \begin{bmatrix} x_1^{(C)} & x_2^{(C)} & \cdots & x_m^{(C)} \\ x_1^{(C)} & x_2^{(C)} & \cdots & x_m^{(C)} \\ \cdots & \cdots & \cdots & \cdots \\ x_1^{(C)} & x_2^{(C)} & \cdots & x_m^{(C)} \end{bmatrix},$$

$$X_1^{(D)} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix} \begin{bmatrix} x_1^{(D)} & x_2^{(D)} & \cdots & x_m^{(D)} \end{bmatrix},$$

$$X_2 = \begin{bmatrix} x_{m+1}^{(1)} & x_{m+2}^{(1)} & \cdots & x_n^{(1)} \\ x_{m+1}^{(2)} & x_{m+2}^{(2)} & \cdots & x_n^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m+1}^{(N)} & x_{m+2}^{(N)} & \cdots & x_n^{(N)} \end{bmatrix}, \text{ the } N \times n \text{ matrix } X_C = \begin{bmatrix} X_1^{(C)} & X_2 \end{bmatrix},$$

and $X_D = \begin{bmatrix} X_1^{(D)} & X_2 \end{bmatrix}$. Let $\boldsymbol{f}_C$ be the vector of which the i-th element is the function value of the i-th row of matrix X_C, i.e., $\boldsymbol{f}_C = f(X_C)$, and $\boldsymbol{f}_D$ is similarly defined, $\boldsymbol{f}_D = f(X_D)$.

Lemma 3. *The two vectors $\boldsymbol{f}_C$ and $\boldsymbol{f}_D$ are linearly related if the function $f(\boldsymbol{x})$ is separable with respect to the first m variable combination $(x_1, x_2, \cdots, x_m)$.*

Proof. Since the first m variable combination $(x_1, x_2, \cdots, x_m)$ are separable, from definition 1, we have $f(\boldsymbol{x}) = \varphi_1(x_1, x_2, \cdots, x_m) \otimes \varphi_2(x_{m+1}, x_{m+2}, \cdots, x_n)$. Accordingly, the vector $\boldsymbol{f}_C = f(X_C) = \varphi_1(X_1^C) \otimes \varphi_2(X_2) = k_C \otimes \varphi_2(X_2)$, where $\otimes$ is a component-wise binary operation, and the scalar $k_C = \varphi_1(\boldsymbol{x}_C)$. Similarly, the vector $\boldsymbol{f}_D = k_D \otimes \varphi_2(X_2)$. As a result,

$$\boldsymbol{f}_C = \begin{cases} k_C/k_D \cdot \boldsymbol{f}_D & \text{if } \otimes \text{ is times} \\ k_C - k_D + \boldsymbol{f}_D & \text{if } \otimes \text{ is plus or minus} \end{cases}$$

which means the two vectors $\boldsymbol{f}_C$ and $\boldsymbol{f}_D$ are linearly related. $\qquad\square$

The above lemmas show that two function-value vectors must be linearly related if the target function has the separability feature, while the separable variables (or their complement variables) are fixed. These are necessary conditions for the separability identification of target function. The sufficient and necessary conditions are given as follows.

Theorem 1. *The function $f(\boldsymbol{x})$ is separable with respect to the first m variable combination $(x_1, x_2, \cdots, x_m)$ if and only if both of the flowing statements are true.*
(1) Any two function-value vectors with fixed $(x_1, x_2, \cdots, x_m)$ are linearly related;
(2) Any two function-value vectors with fixed $(x_{m+1}, x_{m+2}, \cdots, x_n)$ are linearly related.

Proof. From Lemma 2, and Lemma 3, we can conclude that the necessary conditions of the theorem hold. The sufficient conditions can be proved by contradiction. Suppose the separable form $f(\boldsymbol{x}) = \varphi_1(x_1, x_2, \cdots, x_m) \otimes \varphi_2(x_{m+1}, x_{m+2} \cdots, x_n)$ can not be derived from the above two conditions. Thus, there is at least one non-separable variable presented in both sub-functions, φ_1 and φ_2. Without loss of generality, we assume x_m to be this non-separable variable. That is,

$$f(\boldsymbol{x}) = \varphi_1(x_1, x_2, \cdots, x_m) \otimes \varphi_2(x_m, x_{m+1}, x_{m+2} \cdots, x_n). \tag{7}$$

Similarly, the process of sampling for the first correlation test can be given as

$$X_1 = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_m^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_m^{(2)} \\ \vdots & \vdots & & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_m^{(N)} \end{bmatrix}, \tag{8}$$

$$\tilde{X}_2^{(A)} = \begin{bmatrix} x_m^{(1)} & x_{m+1}^{(A)} & x_{m+2}^{(A)} & \cdots & x_n^{(A)} \\ x_m^{(2)} & x_{m+1}^{(A)} & x_{m+2}^{(A)} & \cdots & x_m^{(A)} \\ \vdots & \vdots & \vdots & & \vdots \\ x_m^{(N)} & x_{m+1}^{(A)} & x_{m+2}^{(A)} & \cdots & x_m^{(A)} \end{bmatrix}, \tag{9}$$

9

and

$$\tilde{X}_2^{(B)} = \begin{bmatrix} x_m^{(1)} & x_{m+1}^{(B)} & x_{m+2}^{(B)} & \cdots & x_n^{(B)} \\ x_m^{(2)} & x_{m+1}^{(B)} & x_{m+2}^{(B)} & \cdots & x_m^{(B)} \\ \vdots & \vdots & \vdots & & \vdots \\ x_m^{(N)} & x_{m+1}^{(B)} & x_{m+2}^{(B)} & \cdots & x_m^{(B)} \end{bmatrix}. \tag{10}$$

Let the extended matrix $X'_A = \begin{bmatrix} X_1 & \tilde{X}_2^{(A)} \end{bmatrix}$, and $X'_B = \begin{bmatrix} X_1 & \tilde{X}_2^{(B)} \end{bmatrix}$. Thus,

$$\boldsymbol{f}'_A = f(X'_A) = \varphi_1(X_1) \otimes \varphi_2\left(\tilde{X}_2'^{(A)}\right) = \varphi_1(X_1) \otimes \boldsymbol{\alpha}, \tag{11}$$

and

$$\boldsymbol{f}'_B = f(X'_B) = \varphi_1(X_1) \otimes \varphi_2\left(\tilde{X}_2^{(B)}\right) = \varphi_1(X_1) \otimes \boldsymbol{\beta} \tag{12}$$

are defined, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are function-value vectors of $\varphi_2\left(\tilde{X}_2^{(A)}\right)$ and $\varphi_2\left(\tilde{X}_2^{(B)}\right)$ respectively. As a result,

$$\boldsymbol{f}'_A = \begin{cases} \boldsymbol{\gamma} \cdot \boldsymbol{f}'_B & \text{if } \otimes \text{ is times} \\ \boldsymbol{\alpha} - \boldsymbol{\beta} + \boldsymbol{f}'_B & \text{if } \otimes \text{ is plus} \\ \boldsymbol{\beta} - \boldsymbol{\alpha} + \boldsymbol{f}'_B & \text{if } \otimes \text{ is minus} \end{cases} \tag{13}$$

where $\boldsymbol{\gamma} = (\alpha_1/\beta_1, \alpha_2/\beta_2, \cdots, \alpha_N/\beta_N)$. From the lemmas, we know that, two vectors $\boldsymbol{f}'_A$ and $\boldsymbol{f}'_B$ are linearly related if they are in the relation of $\boldsymbol{f}'_A = k_1 \cdot \boldsymbol{f}'_B + k_2$, where k_1 and k_2 are constant scalars, $k_1 \neq 0$. But, from the above discussion, the components of all the three vectors, $\boldsymbol{\gamma}$, $\boldsymbol{\alpha} - \boldsymbol{\beta}$ and $\boldsymbol{\beta} - \boldsymbol{\alpha}$, are not constant, due to the randomness of sample points $\left(x_m^{(1)}, x_m^{(2)}, \cdots, x_m^{(N)}\right)$. This contradicts the supposition that the two vectors $\boldsymbol{f}'_A$ and $\boldsymbol{f}'_B$ are linearly related, and so Equation (7) cannot hold. $\square$

4.3. Notes on BiCT

The proposed technique is called bi-correlation test (BiCT) since two complementary correlation tests are simultaneously carried out to determine whether a variable or a variable-combination is separable.

The above process is illustrated with two sub-functions, and it could be extended to determine the separability of a function with more sub-functions. However, if the binary operators $\otimes_1, \otimes_2, \cdots, \otimes_m$ in equation (6) are mutually different with mixed times and plus or minus, the extension might be a little

difficult. This issue will be left for the future work. Hereafter, we assume that the binary operators in equation (6) are the same, i.e, $\otimes_i=$ times or $\otimes_i=$ plus or minus for all $i = 1, 2, \cdots, m$, for simplicity. In this case, the extension process is very easy and omitted here.

To enhance the stability and efficiency of the algorithm, the distribution of sample points should be as uniform as possible. Therefore, controlled sampling methods such as Latin hypercube sampling (Beachkofski and Grandhi, 2002) and orthogonal sampling (Steinberg and Lin, 2006) are preferred for sample generation.

For the correlation test, any of correlation methods could be used. That is, Pearson's r method, Spearman's rank order correlation, and Kendall's τ correlation are all effective for BiCT.

Take the function $f(\boldsymbol{x}) = 0.8 + 0.6 * (x_1^2 + \cos(x_1)) * \sin(x_2 + x_3) * (x_2 - x_3), x \in [-3, 3]^3$, as an example, the first sample set consists of 13 uniformly distributed points in $[-3, 3]$, and the second sample set consists of 169 uniformly distributed points in $[-3, 3]^2$. The correlation tests could be illustrated as in Fig. 2. As can be seen that the function-value vectors $\boldsymbol{f}^A$ and $\boldsymbol{f}^B$ are linearly related (Fig. 2(b)), in which the variable x_2 and x_3 are fixed when considering the first variable x_1 (Fig. 2(a)). Similarly, to find out the separability of variable combination (x_2, x_3), the first variable x_1 is fixed (Fig. 2(c)). The corresponding function-value vectors $\boldsymbol{f}^C$ and $\boldsymbol{f}^D$ are linearly related (Fig. 2(d))

The D&C method with BiCT technique is described with functions of explicit expressions. While in practical applications, no explicit expression is available. In this case, some modifications need to adapt for D&C method. In fact, for data-driven modeling problems, a surrogate model of black-box type could be established as the underlying target function (Forrester et al., 2008) in advance. Then the rest steps are the same as above discussions.
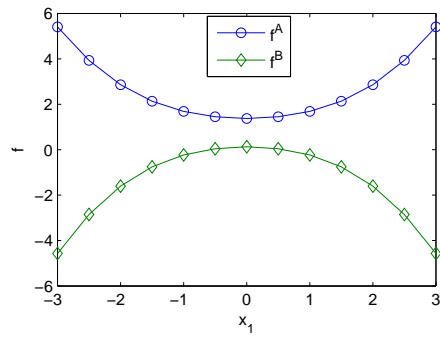
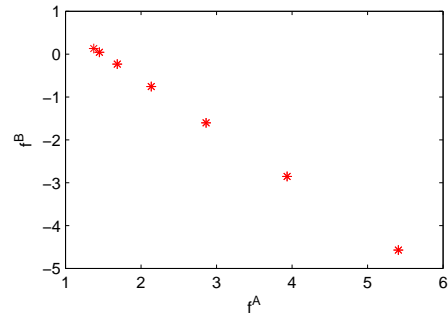## 5. Numerical results

5.1. Analysis on Computing time

The computing time (t) of a genetic programming with the proposed divide and conquer (D&C) method consists three parts:
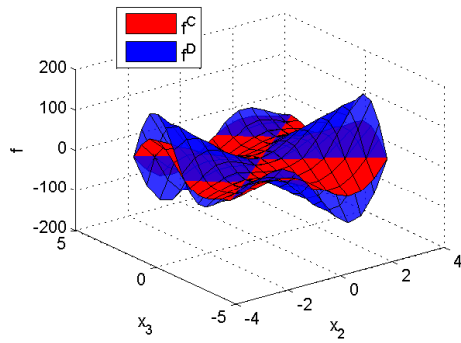
$$t = t_1 + t_2 + t_3 \tag{14}$$

where t_1 is for the separability detection, t_2 for sub-function determination, and t_3 for function recover. Note that both the separability detection and
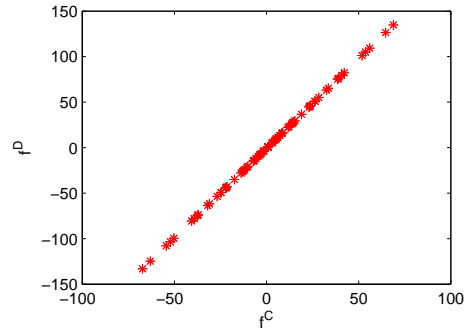
(a) (x_2, x_3) are fixed

(b) $\boldsymbol{f}^A \leftrightarrow \boldsymbol{f}^B$

(c) x_1 are fixed

(d) $\boldsymbol{f}^C \leftrightarrow \boldsymbol{f}^D$

Figure 2: Demo of separability detection process of BiCT

Table 1: A mapping table for parse-matrix evolution

$a_{.1}$	-5	-4	-3	-2	-1	0	1	2	3	4	5
T	$\sqrt{\cdot}$	ln	cos	/	-	skip	+	$*$	sin	exp	$(\cdot)^2$
$a_{.2}, a_{.3}$	-5	-4	-3	-2	-1	0	1	2	$\cdots$	d	
expr	λ_2	λ_1	f	f_2	f_1	1.0	x_1	x_2	$\cdots$	x_d	
$a_{.4}$	-1	0	1								
$f \to f_k$	skip	f_1	f_2								

function recover processes are double-precision operations and thus cost much less time than the sub-function determination process. That is, $t \approx t_2$.

It is obviously that the CPU time for determining all sub-functions (t_2) is much less than that of determining the target function directly (t_d). Next, a typical genetic programming, parse matrix evolution (PME), is taken as the optimization driver (other GP algorithms should also work) to show the performance of D&C.

Suppose that the dimension of the target function is d, the height of the parse-matrix h, and the mapping Table as in Table 1, then the parse-matrix entries $a_{.1} \in -5, -4, ..., 4, 5$, $a_{.j} \in -5, -4, \cdots, d$, $(j = 1, 2)$, and $a_{.4} \in -1, 0, 1$. Thus the parse-matrix $(a_{ij})_{h \times 4}$ have as many as $(11 * (6 + d) * (6 + d) * 3)^h$ possible combinations. Thus the CPU time of determining the target function directly satisfies

$$t_d \sim (11 * (6 + d) * (6 + d) * 3)^h. \tag{15}$$

This means that the searching time of determining a target function will increase exponentially with model complexity. Using D&C, only the sub-functions are needed to determinate, and each sub-function has less dimension d and less complexity h. Therefore, it will cost much less CPU time. In fact, by D&C powered GP, the CPU time will increase only linearly with dimensions, provided that the target function is completely separable.

Take equation (2) in Section 1 as an example. Without D&C, to search directly, the control parameters of PME should be set as follows: $d = 5$, $h \geq 9$. From equation (15), the order of required CPU time $t_d = O(2.58 \cdot 10^{32})$.

Using D&C powered PME, the required CPU time will be much less. In fact, after the separation detection, the function is divided into for sub-functions as follows.

$$q_s = 1.83 \times 10^{-4} \boxed{v^3} \cdot \boxed{\sqrt{\rho}} \cdot \boxed{1/\sqrt{R}} \cdot \boxed{(1 - h_w/h_s)}$$

For the sub-function v^3, $1/\sqrt{R}$, and $1 - h_w/h_s$, the control parameters of PME should be set as $d = 1$, $h \geq 2$. For the sub-function $\sqrt{\rho}$, $d = 1$, $h \geq 1$. As a result, $t_2 \approx 4*O(2.61 \cdot 10^6) = O(10^7)$ by equation (15), which means the D&C method could reduce the computational effort by orders of magnitude.

5.2. Program timing

Next, two illustrative examples are presented to show how much time the D&C technique could save in practical applications.

Again, equation (1) and equation (2) are set as the target function, respectively. For equation (1), the sample set consists 100 observations uniformly distributed in [1,10] degree and [1000, 10000] (i.e., θ=1:10; Re_x = 1000:1000:10000). The angle θ is fixed to 5 degree while detecting the sub-function $f_1(Re_x)$, and the Renold number Re_x is fixed to 5000 while detecting the sub-function $f_2(\theta)$.

For equation (2), the sample set consists 30000 observations uniformly distributed in a box in R^5 (i.e., $v = 500 : 100 : 1000; \rho = 0.0001 : 0.0001 : 0.001; R = 0.01 : 0.01 : 0.1; h_w = 10000 : 10000 : 50000; h_s = 100000 : 100000 : 1000000$). The velocity of free-stream v, density of air ρ, radius of nose R, Wall enthalpy h_w, and total enthalpy h_s are fixed to 800 m/s^2, 0.0005 kg/m^3, 0.05 m, 20000 J/kg, and 200000 J/kg, respectively, while detecting the sub-functions.

In both tests, the program stops when the current model is believed good enough: $1 - R^2 < 1.0 \cdot 10^{-10}$, where $R^2 = 1 - \frac{SSE}{SST}$ can be regarded as the fraction of the total sum of squares that has 'explained by' the model. To suppress the affect of randomness of PME, 10 runs are carried out for each target function, and the averaged CPU time on a PC using a single CPU core (Intel(R) Core (TM) i7-4790 CPU @3.60GHz) is recorded to show its performance. The test results (see Table 2, and Table 3) show that D&C technique can save CPU time remarkably. For equation (1), PME needs about 12 minutes and 26 seconds to get an alternative of the target function without D&C technique, while the D&C powered PME needs only about 11.2 seconds, which is much faster than the original algorithm. Similar conclusion could also seen from the test results of equation (2) (see Table 3). Note that the total time of D&C powered PME includes t_1, t_2 and t_3 (See Equ. 14), and $t_1 + t_3 \approx 0.2$ for Equ. (1), 0.3 for Equ. (2).

Table 2: Performance of PME on detecting Equ. (1) (with and without D&C)

Target Function	CPU time	Result Expression
For D&C powered PME		
$f_1(Re_x)$	3s	$St = 0.1978/\sqrt{Re_x}$
$f_2(\theta)$	8s*	$St = 0.03215 * \theta - 0.01319 * \theta^3$
Total time	11.2s	
PME without D&C		
$f(Re_x, \theta)$	12m26s*	$St = 2.274 * \theta * \cos(0.9116 * \theta)/\sqrt{Re_x}$
Total time	746s	

* PME failed to get the exact result, but always result in an alternative function with fitting error of zero in double precision (i.e., $1 - R^2 = 0.0$).

Table 3: Performance of PME on detecting target Equ. (2)(with and without D&C)

Target Function	CPU time	Result Expression
For D&C powered PME		
$f_1(v)$	3s	$q_s = 1.647 \cdot 10^{-5} * v^3$
$f_2(\rho)$	2s	$q_s = 3.77 \cdot 10^5 * \sqrt{\rho}$
$f_3(R)$	4s	$q_s = 6.49 \cdot 10^3 * \sqrt{0.08442/R}$
$f_4(h_w, h_s)$	9s	$q_s = 9370 - 9370 * h_w/h_s$
Total time	18.3s	
PME without D&C		
$f(v, \rho, R, hw, hs)$	85m43s	$q_s = 0.000183 * v^3 * \sqrt{\rho/R} * (1 - h_w/h_s)$
Total time	5143s	

6. Conclusion

The divide and conquer (D&C) method for symbolic regression has been presented. The main idea is to make use of the separability feature of the underling target function to simplify the search process. In D&C, the target function is divided into a number of sub-functions based on the information of separability detection, and the sub-functions are then determined by any of a genetic programming (GP) algorithms.

The most important and fundamental step in D&C is to identify the separability feather of the concerned system. To fulfill this task, a special algorithm, bi-correlation test (BiCT), is also provided for separability detection in this paper.

The study shows that D&C can accelerate the convergence speed of GP by orders of magnitude without losing the generality, provided that the target function has the feature of separability, which is usually the case in practical engineering applications.

References

References

Anderson, J., 2006. Hypersonic and High-Temperature Gas Dynamics (2nd ed.). American Institute of Aeronautics and Astronautics, Inc., Virginia.

Beachkofski, B., Grandhi, R., APRL 2002. Improved distributed hypercube sampling. In: 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Structures, Structural Dynamics, and Materials and Co-located Conferences. AIAA paper no. 2002-1274. Denver, Colorado.
URL https://doi.org/10.2514/6.2002-1274

Berenguel, L., Casado, L. G., Garca, I., Hendrix, E. M. T., Messine, F., 2013. On interval branch-and-bound for additively separable functions with common variables. Journal of Global Optimization 56 (3), 1101–1121.

Chen, C., Luo, C., Jiang, Z., 2017. Elite bases regression: A real-time algorithm for symbolic regression.
URL https://arxiv.org/abs/1704.07313

d'Avezac, M., Botts, R., Mohlenkamp, M. J., Zunger, A., 2011. Learning to predict physical properties using sums of separable functions. SIAM Journal on Scientific Computing 33 (6), 3381–3401.

Forrester, A., Sobester, A., Keane, A., 2008. Engineering design via surrogate modelling: a practical guide. John Wiley & Sons, Hoboken, New Jersey.

Koza, J. R., 2008. Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, MA.

Luo, C., Zhang, S.-L., 2012. Parse-matrix evolution for symbolic regression. Engineering Applications of Artificial Intelligence 25 (6), 1182–1193.

McConaghy, T., 2011. FFX: Fast, Scalable, Deterministic Symbolic Regression Technology. Springer New York, New York, NY, pp. 235–260.

O'Neill, M., Ryan, C., Aug. 2001. Grammatical evolution. IEEE Trans. Evol. Comp 5 (4), 349–358.
URL http://dx.doi.org/10.1109/4235.942529

Schmidt, M., Lipson, H., 2009. Distilling free-form natural laws from experimental data. Science 324 (59236), 81–85.

Steinberg, D. M., Lin, D. K. J., 2006. A construction method for orthogonal latin hypercube designs. Biometrika 93 (2), 279–288.