ES-GP: An Effective Evolutionary Regression Framework with Gaussian Process and Adaptive Segmentation Strategy

Shijia Huang and Jinghui Zhong (Corresponding author)

School of Computer Science and Engineering
South China University of Technology, Guangzhou, China
Email: jinghuizhong@gmail.com

Abstract. This paper proposes a novel evolutionary regression framework with Gaussian process and adaptive segmentation strategy (named ES-GP) for regression problems. The proposed framework consists of t-wo components, namely, the outer DE and the inner DE. The outer DE focuses on finding the best segmentation scheme, while the inner DE focuses on optimizing the hyper-parameters of GP model for each segment. These two components work cooperatively to find a piecewise gaussian process solution which is flexible and effective for complicated regression problems. The proposed ES-GP has been tested on four artificial regression problems and two real-world time series regression problems. The experiment results show that ES-GP is capable of improving prediction performance over non-segmented or fixed-segmented solutions.

Keywords: Gaussian Process · Differential Evolution

1 Introduction

Regression analysis is an active and important research topic in scientific and engineering fields. Traditional methods for regression analysis focus on choosing an appropriate model and adjusting model parameters to estimate the relationships between inputs and outputs. These methods usually make strong assumptions of data, which are ineffective if the assumptions are invalid. Gaussian Process (GP) is a powerful tool for regression analysis, which makes little assumption of data. Developed base on Statistical Learning and Bayesian theory, GP is flexible, probabilistic, and non-parametric. It has been shown quite effective in dealing with regression problems with high dimension and nonlinear complex data [1].

However, there are some drawbacks of GP. First, GP requires a matrix inversion which has a time complexity of $O(n^3)$ where n is the number of training data. Second, the covariance function and the hyper-parameters of GP model should be carefully fine-tuned to achieve satisfying performance. In the literature, many efforts have been made to solve the above problems, but most methods mainly focus on constructing a single type of GP model, not flexible enough for complicated regression data involving multiple significant different segments.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93713-7_71

To address the above issues, we propose an evolutionary segmentation GP framework named ES-GP, which can automatically identify the segmentation in regression data and construct suitable GP model for each segment. In ES-GP, there is an outer DE focuses on finding a suitable segmentation scheme, while an inner DE, embedded in the outer DE, focuses on tuning the GP model associated to each segment. Once the GP model for each segment is determined, the fitness of the given segmentation scheme can be evaluated. Guided by this fitness evaluation mechanism, the proposed framework is capable of evolving both segmentation scheme and the GP model for each segment automatically. Experimental results for six regression problems show that ES-GP is capable of improving prediction performance over non-segmented or fixed-segmented solutions.

2 Preliminaries

2.1 Gaussian Process for Regression

GP is a non-parametric model that generates predictions by optimizing a Multivariate Gaussian Distribution (MGD) over training data such that the likelihood of the outputs given the inputs is maximized[2]. Specifically, given a set of training data $s = [x, y]$ and predict output of a query input x_* is y_* , then we have:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mu, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{1}$$

where μ is the mean of the MGD which is commonly set to zero, T indicates matrix transposition, K , K_* and K_{**} are covariance matrixes, i.e.,

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \tag{2}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_n) \end{bmatrix} \tag{3}$$

$$K_{**} = k(x_*, x_*) \tag{4}$$

where $k(x, x^{'})$ is the covariance function used to measure the correlation between two points. There are a number of covariance functions in the literatures, "Squared Exponential" is a common one which can be expressed as:

$$k(x, x^{'}) = \sigma_f^2 exp\left[\frac{-(x - x^{'})^2}{2l^2}\right] \tag{5}$$

where σ_f and l are hyper-parameters of the covariance function.

Based on (1) and using the marginalization property, we can get that the conditional distribution of y_* given y also follows a Gaussian-distributed, i.e.,

$$y_*|y \sim \mathcal{N}(K_* K^{-1} y, K_{**} - K_* K^{-1} K_*^T) \tag{6}$$

Hence, the best estimate of y_* is the mean of this distribution and the uncertainty of the estimate is captured by the variance.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93713-7_71

2.2 Related Works on Enhanced GPs for Regression

Various efforts have been proposed to improve GP for regression. Generally, there are two major research directions. The first direction focuses on model calibration. Traditional methods of optimizing hyper-parameters have risks of falling into a local minima, Petelin[6] shown that evolutionary algorithms such as DE and PSO can outperform the deterministic optimization methods. Sundararajan and Keerthi[9] proposed some predictive approachs to estimate the hyper-parameters. Meanwhile, commonly used covariance functions may not model the data well, Kronberger[3] proposed a Genetic Programming to evolve composite covariance functions. Paciorek and Schervish[5] introduced a class of nonstationary covariance functions for GP regression. Seeger[7] proposed a variational Bayesian method for model selection without user interaction.

The second direction focuses on reducing the computational cost for GP model construction. Nguyen-Tuong[4] proposed the LGP which clusters data and establishes local prediction model. Williams and Seeger[10] proposed using *Nyström Method* to speed up kernel machines. Snelson and Ghahramani[8] proposed sparse GP whose covariance is parameterized by pseudo-input points.

3 The proposed method

3.1 General Framework

As illustrated in Figure 1, the proposed framework consists of two components. The outer DE for finding the best segmentation scheme, and the inner DE for optimizing the GP model associated to each segment. Accordingly, the data are divided into three parts to facilitate the search. The training data is used by the inner DE to calibrate the GP model. For each segment, the commonly used covariance functions are enumerated to construct the GP model, and the hyper-parameters are optimized by Inner DE. Once the best GP model in each segment is obtained, the validation data is used to evaluate the quality of the segmentation scheme. Guided by this fitness evaluation mechanism, the outer DE evolve a group of candidate solutions iteratively, until the termination condition is met. The testing data is used to test the performance of the final solution. JADE[12] is adopted as the solver in both outer DE and inner DE.

3.2 Framework Implementation

Chromosome Representation In this paper, we focus on dealing with one dimensional regression data. It can use a set of segment points to describe the segmentation scheme. Hence, in the proposed framework, we use an array of real numbers to represent the chromosome of the outer DE, as expressed by:

$$X_i = \{X_{i,1}, X_{i,2}, ..., X_{i,D}\} \tag{7}$$

where $\lfloor X_{i,j} \rfloor$ represents the length of the ith segment and D is the maximum number of segments set by users. When the length sum of the former segments is greater than or equal to the total length of the data, the latter parts of the chromosome is ignored.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93713-7_71

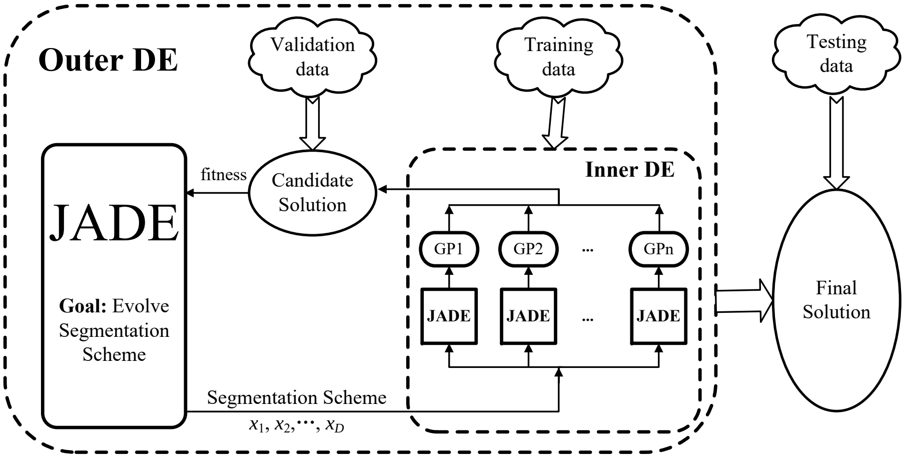


Fig. 1. Algorithm framework

Step 1 - Initialization The first step is to form the initial population. For each chromosome X_i, the jth dimension value is randomly set by:

$$X_{i,j} = rand(L_{min}, L_{max}),\ j = 1, 2, ..., D \tag{8}$$

where L_{min} and L_{max} are the lower bound and the upper bound of the segment length set by users, prevent producing extremely short (or long) segments.

Step 2 - Fitness Evaluation This step aims to evaluate the fitness of the segmentation scheme. Specifically, for each segment, we enumerate commonly used covariance function and optimize the hyper-parameters by the inner DE. The GP model with optimal hyper-parameters which has the maximum marginal likelihood will be considered as the most suitable model. In the inner DE, each chromosome is a set of real numbers, with each representing one hyper-parameter of the GP model. The marginal likelihood of the GP model is used as the fitness value of the individual. Guided by this, the inner DE is capable of fine-tuning the hyper-parameter setting of the GP models associated to the segmentation.

The validation data is used to test the quality of the entire solution. For each point, we firstly determine which segment it belongs to according to its x-coordinate and make prediction by the corresponding GP model. The average error is used as the fitness value of the segmentation scheme. Root-Mean-Square-Error (RMSE) is adopted to calculate the error value, i.e.,

$$f(S(\cdot)) = \sqrt{\frac{\sum_{i=1}^{N}(y_i - o_i)^2}{N}} \tag{9}$$

where y_i is the output of current solution $S(\cdot)$, o_i is the true output of the i_{th} input data, and N is the number of the samples be tested.

Step 3 - Mutation & Crossover The mutation and crossover is same as in JADE, so we omit the description of them, the details can be found in [12].

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93713-7_71

Step 4 - Evaluation and Selection The selection operation selects the better one between parent vector $x_{i,g}$ and trial vector $u_{i,g}$ to become a member of the next generation. In our method, the ones that have better prediction results have a smaller fitness value and will be retained to the next generation.

4 Experiment Studies

We test the effectiveness of ES-GP on four one-dimensional artificial data sets and two real-world time series regression problems. The four artificial data are generated by combining different kinds of functions (e.g., periodic functions and linear functions). The two real time series data sets are obtained from datamarket.com. The first one is ExRate-AUS[1]. The second one is ExRate-TWI[2]. Distribution of the six experiment data sets can refer to Figure 2.

We compare our algorithm with three other algorithms. The first one is the classic GP with single kernel function. The second one is a hybrid GP with multiple kernel functions (named FS-GP), in which data are divided into fixed length segments (set to 40 in this study). The third algorithm is SL-GEP [11], which has been show quite effective for symbolic regression.

JADE is adopted to optimize the hyper-parameters in GP for ES-GP, FS-GP and classic GP, related parameters of JADE are set according to author's recommended and the Maximum Evaluation Time is set to 5000. The kernel functions considered are: *Squared Exponential*, *Rational Quadratic* and *SE with Periodic Element*. Parameters setting in outer DE of ES-GP is same as in inner DE, except that MAXEVAL is 3000. Parameters of SL-GEP are set as suggested in [11].

4.1 Results

Table 1. RMSE of the six problems.

Algorithm	SLGEP [11]	GP	FS-GP	ES-GP
Artificial 1	5.5837889 -	0.0779977 -	0.1753696 -	**0.01658**
Artificial 2	10.8493016 -	1.1830659 -	1.41982 -	**1.0192136**
Artificial 3	21.4422843 -	2.2163847 -	3.2910243 -	**1.571685**
Artificial 4	23.2971078 -	4.5284973 -	5.2494532 -	**3.390701**
ExRate-AUS	12.337935 -	1.91834331 -	2.1590919 -	**1.7031436**
ExRate-TWI	9.5056589 -	1.8520972 ≈	2.0035029 -	**1.7073585**

Symbols -, ≈ and + represent that the competitor is respectively significantly worse than, similar to and better than ES-GP according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

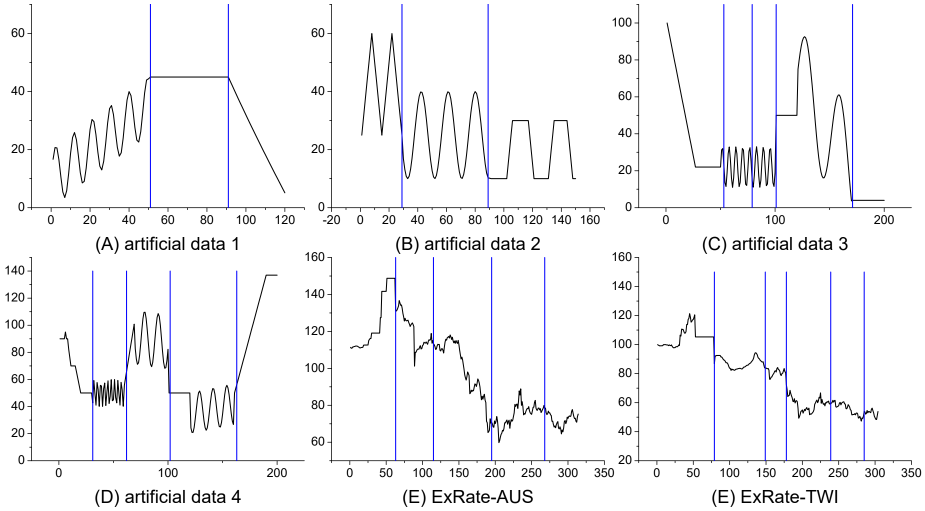
Table 1 shows the average RMSE of six data sets and the Wilcoxon's signed-rank test is conducted to check the differences. The statistical results indicate

[1] https://datamarket.com/data/set/22wv/exchange-rate-of-australian-dollar-a-for-1-us-dollar-monthly-average-jul-1969-aug-1995.

[2] https://datamarket.com/data/set/22tb/exchange-rate-twi-may-1970-aug-1995.

that ES-GP exhibited better performance than SLGEP, FS-GP and GP. Among them, ES-GP showed greater advantages in data set 3 and data set 4, indicating that ES-GP is very suitable for complex time series regression problem. As for the two real-world problems, ES-GP's advantage is not so obvious as in previous data sets because the segmentation characteristics are less obvious, but with the suitable segmentation scheme, the RMSE found by ES-GP is lower than other methods.

Figure 2 shows the example segmentation schemes found by ES-GP, which shows that ES-GP can find the appropriate segmentation. In artificial data set 1 and 2, the number of segments found by ES-GP is the same as we make the data and all segmentation points are almost sitting the right place. ES-GP finds more segments than originally setting in data set 3, however the original segments are within the set of these segments so the segmentation is successful. In data set 4, ES-GP finds less segments, but GP model in each segment can also perform well in such situation. The optimal segmentation is unknown for the two real-world problems. However, our method can find a promising segmentation scheme, which can help GP models perform better.



(A) artificial data 1 (B) artificial data 2 (C) artificial data 3

(D) artificial data 4 (E) ExRate-AUS (E) ExRate-TWI

Fig. 2. Result of segmentation in experiment data sets. The blue lines represent the boundary between segments.

5 Conclusion

In this paper, we have proposed a novel evolutionary regression framework with Gaussian process and adaptive segmentation named ES-GP. In ES-GP, a new chromosome representation is proposed to represent the data segmentation scheme. An outer DE is utilized to optimize the segmentation scheme and an inner DE is utilized to optimize the Gaussian process associated to each segment. The proposed ES-GP is tested on four artificial data sets and two real-world time series regression problems. The experimental results have demonstrated that the

DOI: 10.1007/978-3-319-93713-7_71

proposed ES-GP can properly divide the data and provide promising prediction performance.

6 Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61602181), and by the Fundamental Research Funds for the Central Universities (Grant No. 2017ZD053).

References

1. Sofiane Brahim-Belhouari and Amine Bermak. Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, 47(4):705–712, 2004.
2. Rasmussen Carl Edward, Williams, Christopher KI. *Gaussian processes for machine learning*. MIT press Cambridge, 2006
3. Gabriel Kronberger and Michael Kommenda. Evolution of covariance functions for gaussian process regression using genetic programming. In *International Conference on Computer Aided Systems Theory*, pages 308–315. Springer, 2013.
4. Duy Nguyen-Tuong, Jan R Peters, and Matthias Seeger. Local gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems*, pages 1193–1200, 2009.
5. Christopher J Paciorek and Mark J Schervish. Nonstationary covariance functions for gaussian process regression. In *Advances in neural information processing systems*, pages 273–280, 2004.
6. Dejan Petelin, Bogdan Filipič, and Juš Kocijan. Optimization of gaussian process models with evolutionary algorithms. *Adaptive and Natural Computing Algorithms*, pages 420–429, 2011.
7. Matthias Seeger. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In *Advances in neural information processing systems*, pages 603–609, 2000.
8. Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.
9. S Sundararajan and S Sathiya Keerthi. Predictive app roaches for choosing hyperparameters in gaussian processes. In *Advances in neural information processing systems*, pages 631–637, 2000.
10. Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.
11. Jinghui Zhong, Yew-Soon Ong, and Wentong Cai. Self-learning gene expression programming. *IEEE Transactions on Evolutionary Computation*, 20(1):65–80, 2016.
12. Zhang, Jingqiao and Sanderson, Arthur C. JADE: adaptive differential evolution with optional external archive *IEEE Transactions on evolutionary computation*, 13(5):945–58, 2009.

DOI: 10.1007/978-3-319-93713-7_71