Research on Database Parameters Tuning Method Based on Embedded Device

Xiaoli Geng1*, Wenchao Xu1 and Yonghong Yin1

1 South China Institute Of Software Engineering, Guangzhou University, Guangzhou, Guangdong, 510990, China
*Corresponding author’s e-mail: gxl@sise.com.cn

Abstract. Various kinds of embedded terminal devices have enriched our lives. As a consequence, a huge amount of data has been generated, which often results in slow operation of the devices and a dramatic drop in data processing capacity. Database parameter tuning will be an important way to maintain or improve the performance of the devices. As database performance is affected by multiple parameters, manual adjustment is more and more difficult. With the development of machine learning, database automatic parameter tuning technology has become one of the main choices to solve this problem. This paper proposes to use MARS regression algorithm to optimize the database parameters of embedded devices, to perform regression prediction and fitting on each part of the data space respectively, and obtain the optimal parameter through the processes of forward, backward, and model selection. Experiments show that compared with the traditional method, the parameter optimization results generated through the method proposed in this article are very nearly the same as those obtained through the traditional method, but the time efficiency is significantly improved, hence improving the efficiency of embedded project development.

1. Introduction

Database is an important component of the embedded system and an indispensable effective measure adopted in more and more individualized application, development and management. With the continuous networking of a large number of embedded terminal devices, the numbers of their adjustable parameters, both the database volume and the business volume will increase rapidly. As a consequence, the traditional manual database parameter tuning becomes more and more difficult, and the operating speed of the embedded devices will be affected.

Bayesian optimization is a classic algorithm in the field. However, Bayesian algorithm has some obvious drawbacks. For example, the process of Bayesian optimization involves choice of kernel functions, which itself has some parameters to be adjusted manually. Otherwise, the performance of the whole algorithm would be affected. On the other hand, when the amount of data increases, the inverse of the covariance matrix needs to be maintained and calculated (the time complexity is O(n³)) every time the regression model in the Gaussian process is renewed through Bayesian algorithm. Therefore, modelling in the process of algorithm implementation will consume more time. All this will greatly affect the operating efficiency of the device[1-2]. In view of this, this paper proposes to use MARS-based parameter tuning algorithm to optimize the database parameters of embedded devices, thus solving the problems of Bayesian algorithm.
2. MARS Regression Algorithm\textsuperscript{[3-4]}

MARS was proposed by Jerome H. Friedman in 1991. MARS takes tensor product of spline functions as the primary function, whose generation does not require manual operation. Therefore, compared with the other methods, MARS boasts not only strong adaptivity but also high precision for model prediction. In the case of many dimensions, due to the expansion of the sample space, how to divide the space becomes a crucial problem. MARS is a regression method dealing with high dimension data, featuring strong generalization ability. This regression method takes the tensor product of the spline functions as the primary function. The determining of the primary functions (the number of the state variables of tensor and the break point of the variables) and the number of the primary functions are automatically completed by the data without manual selection. With MARS model, this problem is solved satisfactorily. It has many advantages in multidimensional sample data processing that many response surface models do not have.

MARS not only combines the strong points of projection pursuit and recursive partitioning, but also introduces the spline as primary function. This method does not require space division is disjoint, and it is sufficient that their union set may cover the whole value space. Each small divided zone represents one coefficient. Input variable $x^*$, and the linear sum of the product of the coefficient of the zone and its primary function is the predicted value. In this way, the sequential estimated value of the functions may be obtained. Moreover, in the case of a few variables which are interactive, this method is featured by more flexibility. More importantly, with this method, the accumulated contributions of the variables and the interaction between different variables are easily determined.

The following three aspects constitute the process of MARS modelling. The first one is forward, second:backward, and the last one: model selection.

The following is a detailed introduction to the whole modelling process of MARS regression model, involving the definition of model and the generation of the finished model.

First, it should be known that generating predicted model through training of the training set is the target-related problem in dealing with the regression problem. In essence, multivariate adaptive regression splines is to conduct staged operation on all the sample space, and then do model fitting for the divided space. Each zone may be illustrated as in the following formula (1):

\[
y = a_0 + \sum_{m=1}^{M} a_m S_m(x) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{k_m} [S_{km}(x_{v(k,m)} - t_{km})]
\]  

In the above formula, $y$ is the model forecast value, $a_0, a_1, \ldots, a_m$ are the coefficients, and $S_m(x)$ and $M$ are primary function and the number of primary functions respectively. $k_m$ is the total number of samples.

$S_{km}$ is 1 or -1. The spline function is shown in formulas (2):

\[
[S_{km}(x_{v(k,m)} - t_{km})] = \begin{cases} (x - t_{km}) & \text{when } x \geq t_{km} \\ 0 & \text{others} \end{cases}
\]

\[
[S_{km}(x_{v(k,m)} - t_{km})] = \begin{cases} (t_{km} - x) & \text{when } x < t_{km} \\ 0 & \text{others} \end{cases}
\]  

In the above formulas, $t_{km}$, $x - t_{km}$ and $t_{km} - x$ refer to the node position and the functions on the left and right sides respectively. The model fitting result is, eventually, the linear combination as shown in the following formula (3):

\[
f(x) = a_0 + \sum_{m=1}^{m} a_m B_m(x)
\]
In the above formula (3), $a_m$ is to estimate coefficient $a_m$ through the minimum residual sum of squares. It may also be taken as $B_m(x)$ formed through the multiplication of many spline functions, based on standard linear regression. The algorithm process is illustrated in the following three aspects.

Forward process is to segment the sample space. However, after this process, many one-section spline functions will be generated, which runs the risk of over-fitting. Many spline functions generated in the above process need to go through the backward process. For each step, pruning should be done. The items deleted lead to the minimum increase in residual sum of squares, hence $f(x)$, the estimated value of the optimal model under each $\lambda$ (the number of items). Cross-validation may be done to estimate the optimal $\lambda$. However, to reduce the amount of calculation, with the result of cross-validation of MARS process in the broad sense as standard, the criterion for cross-validation of MARS process in the broad sense is defined as:

$$GVC(\lambda) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - \bar{f}_\lambda(x_i))^2$$

In the above formula (4), $M(\lambda)$ is the final number of terms in the regression model, $\hat{f}_\lambda(x_i)$ is the result of the $i$th sample through model prediction, and $N$ is the size of the sample space. Finally, the model with the minimum GCV value is chosen as the final execution result of the backward process [5-7].

3. MARS-based Regression Hyper-parameter Tuning Algorithm [8]

The above is the introduction of the modelling process of MARS regression model. The following is the hyper-parameter tuning algorithm based on multivariate adaptive regression spline, after which is the introduction of each step.

Hyper-parameter tuning algorithm based on multivariate adaptive regression spline (MARS)

Input: the function to be optimized.

1) Generate $n$ initial points, and assess them to establish database $D = \{x_i, y_i, i = 1, 2, \ldots, n\}$

2) While (not meeting the terminal condition)

3) Use data in $D$ to establish the model of multivariate adaptive regression spline, i.e., MARS regression model.

4) Generate a great number of random candidate solutions through the sampling method designed in the algorithm.

5) Generate the next iteration point from these candidate solutions based on MARS regression model.

6) Make authentic assessment of $\cdot x_{n+1}, y_{n+1} = f(x_{n+1})$ through the objective function.

7) Increase data $\{x_{n+1}, y_{n+1}\}$ to $D$, and renew MARS regression model.

Output: the optimal solution in $D$.

As this paper is devoted to the research on hyper-parameter tuning algorithm based on machine learning, some hyper-parameters are inevitably non-numeric parameters. From the above-mentioned introduction of algorithm, we learn that this type of non-numeric parameters cannot be directly substituted into the expression in the algorithm, nor related fitting can be done on the regression model. Therefore, before algorithm is done, these non-numeric hyper-parameters have to be encoded. For example, the hyper-parameter bootstrap (as for whether to establish the sample set based on the sampling with replacement, True means YES, and the default is True) in the random forest may be translated into a value to express its original meaning (0 refers to True, and 1 refers to False).

In Step 1), $n$ random initial samples are to be generated. In $n = 2^d + 1$, $d$ refers to the dimension of the hyper-parameter to be optimized. Though the initial point here is random sampling, to prevent
the result from falling into the locally optimal solution due to the over-dense distribution of initialized samples, sampling of the initial point needs to be distributed as equally much as possible.

In Step 3), modeling is done for the multivariate adaptive regression spline model.

In Step 4), A great amount of random sampling is done near the good solutions to generate the candidate solution set of the next iteration point.

So far, so much for the introduction of the specific procedures for the hyper-parameter tuning algorithm based on MARS regression proposed in this paper.

4. Experiment

First, do the experiment on comparing the method proposed in this paper with Bayesian algorithm. In this experiment, Bayesian optimization model takes proxy function as Gaussian Process. UCI data set is adopted, and the machine learning model to be tuned is GBM model. In doing the experiment, a comparison is done on the aspects: 1) the degree of accuracy after model optimization; 2) the execution time concerning tuning algorithm. The result of comparison concerning the two aspects is shown in Figure 1.

In practical application process, the purpose of optimization algorithm is to achieve a good result in the shortest possible time in the context of little difference in accuracy with the two methods adopted. It can be seen from Fig 2 that compared with Bayesian algorithm, the algorithm proposed in this paper may bring about a performance result after optimization of the target model which is nearly the same as that achieved through Bayesian algorithm. Moreover, with the proposed algorithm, the execution time witnesses remarkable improvement.

As Spark parameter has huge impact on the operating condition of database[9-10], we will do tuning for Spark parameters. In Spark, the parameters which have huge impact on the performance are mainly related to Executor resource allocation and Driver resource allocation as well as Shuffle movement[11]. To verify that this method is helpful to the operating efficiency of embedded devices, a comparison is made. The working condition of the Coordinator is observed on the basis of the changes in values in Endivice and thread. The embedded device used in this experiment is TINY4412. The specific test results are shown in Table 1.

![Figure 1. Result of comparison test](image)

| Number of terminals | Number of coordinator threads | Working condition of coordinator by default | Working condition of coordinator while optimizing |
|---------------------|-------------------------------|---------------------------------------------|-----------------------------------------------|
| 1                   | 1                             | smooth                                      | smooth                                        |
| 3                   | 1                             | smooth                                      | smooth                                        |
The experiment result proves that tuning parameters used, the working performance of the coordinator is greatly improved. With the increase of the numbers of both terminals and threads, the embedded devices suffer from obvious freeze by default, and sometimes crash. Comparatively, noticeably, freeze occurs less frequently after device optimization. This indicates that with out tuning method, the data performance of the embedded device may be improved, hence increased its working efficiency and operation safety.

5. Conclusion
This paper proposes a database tuning method for embedded devices. MARS regression algorithm is used to conduct parameter optimization of the embedded device. Regression prediction and fitting is done on each part of the data space respectively, and the optimal parameters are obtained through the process of forward, backward, and model selection. Compared with the traditional Bayesian approach, there is not much difference in accuracy through database tuning method proposed in this paper. However, it works well in execution time. Moreover, it may save the development time of embedded projects and improve development efficiency. The optimized parameter configured into the embedded device, the embedded device operates smoother. This indicates that database parameter optimization plays its role, and that this method is applicable to embedded device development and has some practical applications. However, the data sources of machine learning has some limitations that parameter recommendation can be done only through data configuration in the history vault. In the future, the author of this paper will explore a learning model which can predict the data size, on the basis of which parameter configuration recommendation is done.

Acknowledgments
South China Institute of Software Engineering, Guangzhou University Quality Engineering Project: Excellent resource sharing course of Database Principles And Applications (ZYGXK202006). Foundation for Young Innovative Talents in Higher Education of Guangdong, China (No.2018KQNCX389).

References
[1] Shahriari B, Swersky K, Wang Z, et al. Taking the human out of the loop: A review of Bayesian optimization[J]. Proceedings of the IEEE, 2016, 104(1): 148-175.
[2] Klein, Aaron, et al. "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets." Computer Science, eprint arXiv:1605.07079, 2016.
[3] Kushner H J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise[J]. Journal of Basic Engineering, 1964, 86(1): 97-106.
[4] Friedman J H. Multivariate adaptive regression splines[J]. The annals of statistics, 1991: 1-67.
[5] Gu C, Wahba G. Discussion: multivariate adaptive regression splines[J]. The Annals of Statistics, 1991, 19(1): 115-123.
[6] Leathwick J R, Rowe D, Richardson J, et al. Using multivariate adaptive regression splines to predict the distributions of New Zealand's freshwater diadromous fish[J]. Freshwater Biology, 2005, 50(12): 2034-2052.

[7] G. Wang, J. Xu and B. He, "A Novel Method for Tuning Configuration Parameters of Spark Based on Machine Learning," 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 586-593.

[8] Wu Haonan. Research on hyperparameter automatic tuning algorithm [D]. Harbin Institute of Technology, 2020:35-36.

[9] Li M, Tan J, Wang Y, et al. Sparkbench: a comprehensive benchmarking suite for in memory data analytic platform spark[C]//Proceedings of the 12th ACM international conference on computing frontiers. 2015: 1-8.

[10] Tous R, Gounaris A, Tripiana C, et al. Spark deployment and performance evaluation on the marenostrom supercomputer[C]//2015 IEEE International Conference on Big Data (Big Data). IEEE, 2015: 299-306.

[11] Chen Wu, The Influence of Mobile Phone Use on Teenagers’ Self-control [D].Central China Normal University, 2006.