A Multitasking Run Time Prediction Method based on GBDT in Satellite Ground Application System

Abstract: In satellite ground application system, it will cause resource constraints when running multiple tasks. To accurately measure out the task run time, this paper proposed a task running time estimation method based gradient boosting decision tree (GBDT). Firstly, according to various features of the time series variables classify applications. The predicted values are given for each node in the tree according to GBDT algorithm. Then through the establishment of multiple trees, the loss function is minimized. Finally, the most accurate predictions are calculated. After analysis, the proposed algorithm has a high accuracy ground.

Keywords: task run time; gradient boosting decision tree; various feature; loss function

1 Introduction

Fengyun meteorological satellite ground application system receives satellite data from the satellites, and depending on the business needs, according to the different levels and types of meteorological satellite data product to generate and distribute. At present, satellites are widely used in communications, meteorology and geology. Satellites have a large number of data and products. Satellite ground application system will face carry more tasks. Due to the limitations of the overall resources, often resulting in a decline multitasking operating efficiency. How to accurately estimate the task run time, thus providing support for the scheduling of the current study is needed to solve a problem.

Certain task run prediction methods have been proposed in the literatures. The paper [1] described and evaluated the Running Time Advisor (RTA), a system that can predict the running time of a compute-bound task on a typical shared, unreserved commodity host. The prediction is computed from linear time series predictions of host load and takes the form of a confidence interval that neatly expresses the error associated with the measurement and prediction processes, error that must be
captured to make statistically valid decisions based on the predictions. In paper [2], they proposed a method for predicting run-time resource consumption in multi-task component based systems based on a design of an application. The paper [3] proposed a running time prediction method for Grid tasks based on our previous work, which is a novel CPU load prediction method. The paper [4] proposed a novel approach that enables the construction models for predicting task running-times of data-intensive scientific workflows. Ensemble Machine Learning techniques are used to produce robust combined models with high predictive accuracy. Information derived from workflow systems and the characteristics and provenance of the data are exploited to guarantee the accuracy of the models. The paper [5] considered three objectives: expected time, long-run average, and timed (interval) reachability. Expected time objectives focus on determining the minimal (or maximal) expected time to reach a set of states. Long-run objectives determine the fraction of time to be in a set of states when considering an infinite time horizon. Timed reachability objectives are about computing the probability to reach a set of states within a given time interval. The paper [6] proposed two approaches utilizing some a priori knowledge and estimating it from scratch via a sparse structure assumption.

Although the researches of the cloud environment resource scheduling achieved good results, but there are still insufficient in error estimation. So, this paper introduces the GBDT algorithm, the prediction process as far as possible down to the minimum loss function, thereby enhancing the overall prediction accuracy.

2 Gradient Boosting Decision Tree Algorithm

GBDT is a very broad application of the algorithm, can be used for classification, regression. In many data prediction has a very good effect. GBDT decision is an iterative algorithm, the algorithm is composed by a number of trees tree, and the conclusion of all the trees make the final decision.

2.1 Algorithm Principle

The GBDT regression tree analysis resource consuming task to run time series to predict the various features of the task run time. GBDT algorithm based on time-series variables various features and applications to classify each node in the tree gives the predicted value. The GBDT algorithm by establishing multiple trees, to minimize the loss function, to get the most accurate predictions.

At the time of the beginning of the algorithm for each sample assigned a weight value, the initial time, each sample is the same important. At each step of the training obtained in model will make the estimated data points have to be wrong. We are at the end of each step, increase the weight of misclassification point weight, and reduce
weight division right on point. This makes is always some point if misclassification, it will be “serious concern.” It will be assigned a high weight on the right. Then after \( N \) iterations (specified by the user), user will get \( N \) simple classification. The final model may be obtained by a linear combination of ways.

If there is a sample \( x \), it may belong to \( K \) categories, the estimated values are \( F_i(x) \cdots F_i(x) \). Logistic transformation is a smooth and standardized data process (such as the length of the vector is 1). The probability \( p_k(x) \) of the result belonging to category \( K \) is as follows:

\[
p_k(x) = \frac{\exp(F_k(x))}{\sum_{l=1}^{K} \exp(F_l(x))}
\]

After logistic transformation, the loss function is as follows:

\[
L\left(\{y_k, F_i(x)\}\right) = -\sum_{k=1}^{K} y_k \log p_k(x)
\]

Where, \( y_k \) is the estimated function of the input sample data. And its derivative, there is the gradient of loss function:

\[
y_{ik} = -\left[\frac{\partial L\left(\{y_k, F_i(x)\}\right)}{\partial y_{ik}}\right]_{F_i(x)} = y_{ik} - p_{k,m-1}(x_i)
\]

### 2.2 Loss Function Gradient Analysis

The input data may belong to five categories \( \{c_1, c_2, c_3, c_4, c_5\} \). If \( x \) belongs to category \( c_3 \), there is \( y = (0, 0, 1, 0, 0) \). The resulting model estimation is \( F(x) = (f_1, f_2, f_3, f_4, f_5) \). Then after logistic transformation can be \( p(x) = (p_1, p_2, p_3, p_4, p_5) \). After the above, the conclusions can be as follows.

If \( g_i \) is the sample when the gradient of a dimension.

When \( g_i > 0 \), \( g_i \) smaller, the probability \( p(x) \) of this dimension should be greater. It should belong to the “right direction” forward. This estimate represents the smaller the more “accurate.”

When \( g_i < 0 \), \( |g_i| \) greater, the probability \( p(x) \) of this dimension should be smaller. It should belong to the “wrong direction” forward. This estimate represents the smaller the more “accurate.”

In general, for a sample of the best gradient is closer to 0.

### 3 Run Time Prediction Based on GBDT

GBDT regression tree based on the training set of feature vector sequence characteristics were divergent. Task run time as GBDT classification entropy, that is, the tree will try to make the crossing task to run every time the minimum entropy.
This approach makes the degree of aggregation run tasks on each node in the tree relatively high. Make the final decision based on the time relatively easy to predict the characteristics of the input operation time.

There are four tasks sample points in Figure 1. The running time of them are \((t_a, t_b, t_c, t_d)\). If \(t_a < t_b < t_c < t_d\), the running time can be divided into two sets \((t_a, t_b)\) and \((t_c, t_d)\). And \((t_a, t_b)\) can be divided into two sets \((t_a)\) and \((t_b)\), \((t_c, t_d)\) can be divided into two sets \((t_c)\) and \((t_d)\).

\[
\begin{align*}
(a+b+c+d)/4 & \quad (a,b,c,d) \\
(a+b)/2 & \quad (a,b) \\
(c+d)/2 & \quad (c,d) \\
\end{align*}
\]

\[
\begin{align*}
(a+b+c+d)/4 & \quad (a,b,c,d) \\
(a+b)/2 & \quad (a,b) \\
(c+d)/2 & \quad (c,d) \\
\end{align*}
\]

**Figure 1.** Multi node sample flow.

In the process of forecasting tasks running, we can go from the root all the way down the final decision condition according to the root node, and get the task running time of the root node. In GBDT workflow, there are M decision trees, obtained from the M trees like tree algorithm running time and the final prediction is GBDT run time.

As shown in Figure 2, tree generated every time according to the last tree prediction residuals of the tree all the variables residuals negligible so far. Multi-tree prediction value obtained by adding the result of addition makes forecasting more precise.

The main purpose of this section is based on the task of running time sequence characteristics to estimate the run time of the task. Enter the program requires the input feature time-series set. We need the help of feature extraction feature extraction module, but also need to use the features described above and converting the characteristic variable generation module.

Generate GBDT prediction module comprises two steps, the matrix (the training set) time series feature to generate variable GBDT decision tree and decision tree based on the task run time prediction.

When the estimated time sequence time, all the features need to enter a variable time series. By M trees forecasting and time series to generate the final run time.
4 Simulation

In order to verify the effectiveness of the proposed algorithm, we use the Visual Studio 6.0 platform to achieve our algorithm based on GBDT, and compare with the run time prediction algorithm based on fixed weights (FWA).

The simulation environment of PC configuration is: CPU Core i5-6500 3.20GHz, RAM=4GB, Windows 7 system. The simulation system through the C++ language, math calculations involved in the experiment were analyzed by calling math.h math library to calculate. Figure 3 shows the prediction accuracy of 10 experiments.

The proposed GBDT algorithm is higher than the FWA algorithm on prediction accuracy.
5 Conclusion

To accurately measure out the task run time, this paper proposed a task running time estimation method based gradient boosting decision tree (GBDT). Firstly, according to various features of the time series variables classify applications. The predicted values are given for each node in the tree according to GBDT algorithm. Then through the establishment of multiple trees, the loss function is minimized. Finally, the most accurate predictions are calculated. After analysis, the proposed algorithm has a high accuracy ground.

Acknowledgment: The work presented in this study is supported by National High-tech R&D Program (2011AA12A104) and the National Natural Science Foundation of China (NSFC61602126).

References