L-025

# Regression Based Execution Time Estimation for Scheduling in Distributed Computing Systems

Buyanmunkh VOLODYA Abhay GHATPANDE Hidenori NAKAZATO

Graduate School of Global Information and Telecommunication Studies, Waseda University

## 1 Introduction

Efficient scheduling is one of the most important factors to minimize the total execution time of an application in a distributed computing system (DCS). It requires the appropriate mapping of the application's component tasks to the machines in the DCS. Optimal mapping can be carried out if the execution time of the tasks on the machines can be predicted accurately.

The execution time depends not only on the physical machine (CPU) characteristics, but also the current machine *state*, i.e. CPU load, free memory, etc. By using the past values of execution time and the corresponding machine state, we can reasonably predict the execution time in another state by regression analysis. We present a new *cross validation criterion* to improve the prediction accuracy of the regression. We evaluate the relative importance of parameters such as CPU load, CPU type, main memory size, cache size, etc. that affect the execution time and also compare kernel functions that are used in regression. We found that the execution time can be predicted with reasonable accuracy, and that the exponential quadratic kernel is better than other kernels used in our simulation.

## 2 Problem Description

Let $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, be the stored data, where $x_{ip}$, $p = 1, \ldots, m$, is the value of machine feature $p$ in observation $\mathbf{x}_i$, and $y_i$ is the execution time. The query point is $\mathbf{q} = (q_1, \ldots, q_m)$, and it is required to estimate the execution time $y_{\mathbf{q}}$.

$\hat{y}_{\mathbf{q}}$, the regression estimator of $y_{\mathbf{q}}$, is defined as

$$\hat{y}_{\mathbf{q}} = \sum_{i=1}^{n} b_{i\mathbf{q}}\, y_i \tag{1}$$

where the regression coefficients $b_{i\mathbf{q}}$ are given by

$$b_{i\mathbf{q}} = \frac{K(D(\mathbf{x}_i, \mathbf{q}), h)}{\sum_{l=1}^{n} K(D(\mathbf{x}_l, \mathbf{q}), h)} \tag{2}$$

Here, $K(\cdot)$ is known as the kernel function, $D(\cdot)$ measures the distance between the query point and the observations, and $h$ is the kernel bandwidth.

The distance function $D(\mathbf{x}_i, \mathbf{q})$ is defined as

$$D(\mathbf{x}_i, \mathbf{q}) = \sqrt{\sum_{p=1}^{m} w_p\, d(x_{ip}, q_p)^2}$$

where $w_p$ is the feature weight and $d(x_{ip}, q_p)$ is the feature distance calculated as follows:

$$d(x_{ip}, q_p) = \begin{cases} \mathit{overlap}(x_{ip}, q_p) & p \text{ nominal} \\ \mathit{diff}(x_{ip}, q_p) & p \text{ real} \end{cases}$$

$$\mathit{overlap}(x_{ip}, q_p) = \begin{cases} 0 & \text{if } x_{ip} = q_p \\ 1 & \text{otherwise} \end{cases}$$

$$\mathit{diff}(x_{ip}, q_p) = \frac{|x_{ip} - q_p|}{\max_p - \min_p}$$

Here $(\max_p - \min_p)$ is the range of feature $p$.

The regression coefficients $b_{i\mathbf{q}}$ are affected to a large degree by the choice of feature weights $w_p$ and the kernel bandwidth $h$, and by the kernel function $K(\cdot)$ to a certain extent [1]. The problem reduces to finding a good kernel, and the optimal values of the feature weights and bandwidth to increase prediction accuracy.

## 3 Proposed Method

The optimal values of $w_p$ and $h$ to be used in the estimation on the query point are found from the experience data by using *leave one out cross validation* [2, 3]. This involves the minimization of a *cross validation criterion*, *PRESS**, Penalized Sum of Squares of Prediction Error. It was found that this criterion fails for certain kernels and hence to compare the performance of different kernels, we proposed a new minimization criterion, *STEPSS*, STandard deviation penalized Sum of Squares of Prediction Error, defined as:

$$\mathit{STEPSS} \equiv \frac{1}{n}\frac{1}{s_{-i}} \sum_{i=1}^{n} (y_i - \hat{y}_{i,-i})^2$$

$$\text{where, } s_{-i} = \sqrt{\frac{1}{n\,(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (b_{ij} - \bar{b})^2}$$

$$\bar{b} = \frac{1}{n\,(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} b_{ij}$$

The regressor $\hat{y}_{i,-i}$ of $y_i$ and the coefficients $b_{ij}$ are obtained from (1) and (2) by leaving out $y_i$.

## 4 Experiments and Analysis

The characteristics of machines used for experiments are shown in Table 1. The machines were loaded by running a background task that could spawn child tasks as required. A test application was created and executed on the machines under various conditions of CPU

| Machine | CPU | num of | CPU | Memory | CPU |
|---|---|---|---|---|---|
| num | type | CPU | Hz | size | cache |
| 1 | PM | 1 | 1GHz | 758M | 1024 |
| 2 | P4 | 1 | 3GHz | 256M | 1024 |
| 3 | P4 | 1 | 3GHz | 512M | 1024 |
| 4 | P4 | 1 | 3GHz | 1G | 1024 |
| 5 | P4 | 2 | 3GHz | 512M | 2048 |
| 6 | AMD | 2 | 2.4GHz | 3G | 1024 |

**Table. 1** Machine Characteristics

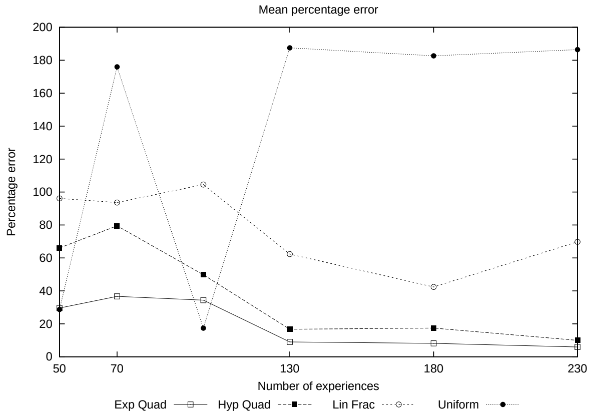

**Fig. 1** Mean percentage error of 50 queries

load and free memory. The execution time of the application for 280 different machine states was recorded, out of which 50 values were randomly picked to represent the queries, leaving 230 states as training data.

For $n = 50, 100, 130, 180, 230$, for four different kernels, the optimal feature weights and kernel bandwidth were found by using a genetic algorithm to minimize the *STEPSS* criterion. Using these values, the execution time for the 50 queries was estimated and the prediction errors were calculated. The different kernels used were:

| Kernel Type | Formula |
|---|---|
| Exponential quadratic | $K(D,h) = \mathrm{e}^{-(D/h)^2}$ |
| Hyperbola quadratic | $K(D,h) = (h/D)^2$ |
| Uniform | $K(D,h) = \begin{cases} 1 & D/h \leq 1 \\ 0 & D/h > 1 \end{cases}$ |
| Linear fraction | $K(D,h) = 1/(1+D/h)$ |

The values of the mean percentage error for the 50 queries for each kernel type and at different values of $n$ are plotted in Fig. 1. Referring to Fig. 1, the following conclusions can be drawn:

- Percentage prediction error in case of exponential quadratic kernel is the smallest among all kernels with a value of 6% at $n = 230$.
- As the value of $n$ increases, the percentage prediction error of the hyperbola quadratic kernel approaches the value of exponential quadratic kernel.
- Linear fraction kernel performs reasonably at higher values of $n$, but the errors are still too large to be practically acceptable.

| Feature weights | Number of experiences | | | | | |
|---|---|---|---|---|---|---|
| | 50 | 70 | 100 | 130 | 180 | 230 |
| Kernel bandwidth | 0.011 | 0.012 | 0.014 | 0.021 | 0.015 | 0.016 |
| CPU HZ | 0.08 | 0.05 | 0.67 | 0.45 | 0.59 | 0.37 |
| CPU # | 0.67 | 0.20 | 0.12 | 0.36 | 0.03 | 0.45 |
| CPU cache | 0.04 | 0.07 | 0.97 | 1.00 | 0.35 | 0.67 |
| CPU load | 0.93 | 1.00 | 0.77 | 0.94 | 0.70 | 0.63 |
| Memory | 0.72 | 0.01 | 0.02 | 0.13 | 0.12 | 0.04 |

**Table. 2** Optimal kernel bandwidth & feature weights

| Boundary | Boundary range | | | |
|---|---|---|---|---|
| | 10% | 5% | 3% | 1% |
| Lower | 16.70 | 19.25 | 23.26 | 36.00 |
| Center | 4.64 | 4.45 | 4.78 | 5.28 |
| Upper | 3.80 | 4.66 | 6.32 | 8.25 |

**Table. 3** Prediction error at boundaries

- Uniform kernel is the worst performing among all the kernels with percent error close to 200%.

The optimal kernel bandwidth and feature weights obtained with exponential quadratic kernel and the *STEPSS* criterion are shown in Table 2. CPU load and cache size are the most important features and receive large weights. The change in kernel bandwidth with the increase in number of experiences is minimal.

To evaluate the prediction error at the boundaries of the available data, we calculated the prediction error for query points that fell close to the lower and upper bounds of the CPU load. The results are shown in Table 3. E.g., the mean percentage prediction error for the queries that lie at the lower 1% of the CPU load range is 36%, for the queries at the upper 1% is 8.25%, and for the queries that lie in the rest of the 98% of the range is 5.25%. This shows the limitation of regression at the boundaries. We hypothesize that as more data is collected at the boundaries, this error will reduce.

## 5 Conclusion

We presented a method to estimate the execution time of tasks on machines using historical machine state information. We proposed a new criterion for cross validation in regression and evaluated its performance for different kernel types. We found that our technique has an average error less than 10%, and the execution time can be predicted with reasonable accuracy even with a small number of experiences.

## Bibliography

[1] E. A. Nadaraya. On estimating regression. *Theory of Probability and it's Applications*, 9:141–142, 1964.

[2] M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9:65–78, 1964.

[3] R. Einsporn. Atlink: A link between least squares regression and nonparametric curve estimation. Unpublished dissertation, 1987.