Task Load Modeling for LTE Baseband Signal Processing with Regression Analysis Approach

CHANG LIU

KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden 2014

XR-EE-SB 2014:004

Abstract

In telecommunication baseband signal processing systems, thousands of tasks are executed every millisecond. These tasks take in different parameters and cause heavy load to the system. The aim of the thesis is to build proper mathematical models for these tasks, enabling the prediction of their load given the corresponding parameters.

For each task, data samples of task load measure and corresponding parameters are provided. No prior knowledge on the task load and its parameters is available. By studying the data samples, an explicit, accurate and simple model is expected. Graphical skills like scatter plots are used as a preliminary analysis of the data. Then first-order and second-order linear models, piecewise-linear models and tree-based models are taken as prototypes for the task modeling. Methods like stepwise linear regression and partial correlation analysis are applied to select proper parameters from many available parameters to simplify the models. An automatic tool is further developed to automate the whole modeling process.

There are 17 tasks in total. For 15 tasks, acceptable models are built with a $RMSE$ lower than 2 times of the estimated noise standard deviation with the assumption of a Gaussian noise, while for the other 2, no adequate models are given. Reasons for not getting acceptable models are discussed and suggestions on future work are proposed.

Sammanfattning

I telekommunikationssystem utförs tusentals uppgifter varje millisekund. Dessa uppgifter tar in olika parametrar och orsakar stor belastning på systemet. Syftet med avhandlingen är att bygga riktiga matematiska modeller för dessa uppgifter som gör det möjligt att förutsäga deras last givet motsvarande parametrar.

För varje uppgift har datasampel med lastutnyttjande och motsvarande parametrar tillhandahållits. En explicit, exakt och enkel modell önskas. Spridningsdiagram används som en preliminär analys av datat. Sedan används första ordningens och andra ordningens linjära modeller, styckvis-linjära modeller och trädbaserade modeller som prototyper för uppgiftsmodellering. Metoder som styckvis linjär regression och partiell korrelationsanalys tillämpas för att välja rätt parametrar från många tillgängliga parametrar för att förenkla modellerna. Ett automatisk verktyg har utvecklats för att automatisera hela modelleringsprocessen.

Det finns 17 uppgifter totalt. För 15 av 17 uppgifter hittades acceptabla modeller byggda med ett RMSE lägre än 2 gånger standardavvikelsen av det uppskattade bruset med antagandet om ett gaussiskt brus. För de andra två uppgifterna hittades inga adekvata modeller. Skäl till att inte få acceptabla modeller diskuteras och förslag på framtida arbete föreslås.

Acknowledgment

The thesis is done with Ericsson AB. Together with Lu Wang, we explored different approaches and each formed an individual thesis report. With the help of many people, the six-month long thesis has been a great experience for me.

First, I'd like to express my deepest gratitude to Dr. Henrik Olson. His views on the methodology and advise on the methods guide me to the right direction. And along the thesis, he has always been available even through he has a very heavy work schedule. Even on vacation, he was glad to spare time to review the thesis draft. His help has been invaluable and created a very nice environment for doing the thesis.

Also many thanks to John Nilson and Prof. Magnus Jansson. John kindly provided suggestions on methods and helped us with data preparation. Magnus has been giving key suggestions on the thesis, and helped us out of the school's thesis procedures. With their support, we saved quite a lot of time and were able to focus more on the thesis itself.

Next, I'd like to extend special thanks to Lu Wang, Johan Parin, Jianrong Zhang, Jonas Allander, and Mathias Ekwing. With my sincere respect and gratitude, I thank you all for the great experience I had.

Moreover, my master program study is fully sponsored by a scholarship under the State Scholarship Fund of Chinese government. I sincerely thank the government for such a valuable opportunity.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Telecommunication technology evolves all the time. In November 2004, the 3rd Generation Partnership Project (3GPP)[1] started work on the Long Term Evolution (LTE) as the access part of the Evolved Packet System (EPS). The LTE solution is developed to meet the main requirements like high spectral efficiency, high peak data rates, short round trip time as well as flexibility in frequency and bandwidth. Based on orthogonal frequency division multiple access(OFDMA), and in combination with higher order modulation (up to 64QAM), large bandwidths (up to 20 MHz) and spatial multiplexing in the downlink (up to 4x4), LTE is able to achieve high data rates. The highest theoretical peak data rate on the transport channel is 75 Mbps in the uplink, and as high as 300 Mbps in the downlink using spatial multiplexing. By the end of 2008, LTE specifications have been included in 3GPP Release 8 [2].

In the industry, as with all other protocols, LTE standards are mainly realized with sets of software programs. The programs execute on different hardware platforms. The software programs and hardware together make a complete system. Ericsson is one leading infrastructure vendor for radio network equipment. In its LTE base station, the 3GPP radio network specification together with Ericsson radio link and radio resource management algorithms are realized in control-, signal processing-, and radio software, and then executed on customized hardware platforms.

In the area commonly referred to as "baseband processing", several thousand software programs are executed every millisecond in a base station on customized digital signal processors (uplink or downlink DSPs). The software programs can be further grouped into different tasks, for example,

channel estimation, antenna combining, decoding, etc. Two schematic plots of the hardware platform and tasks are given in Figure 1.1 and Figure 1.2.

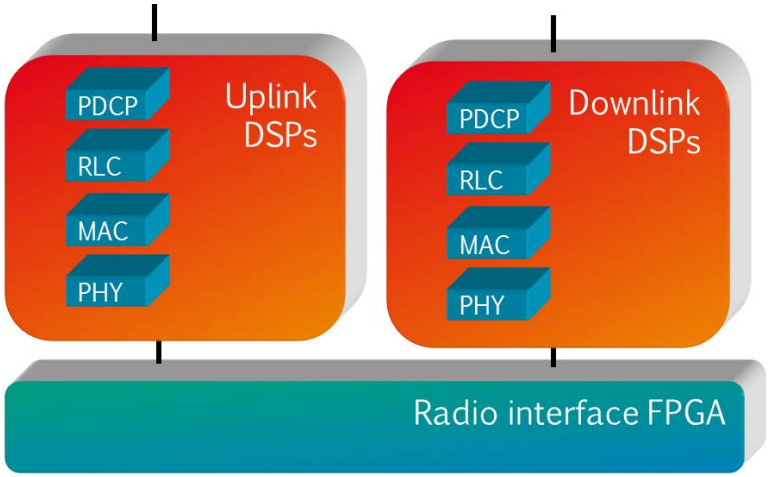

Figure 1.1: The hardware platform.

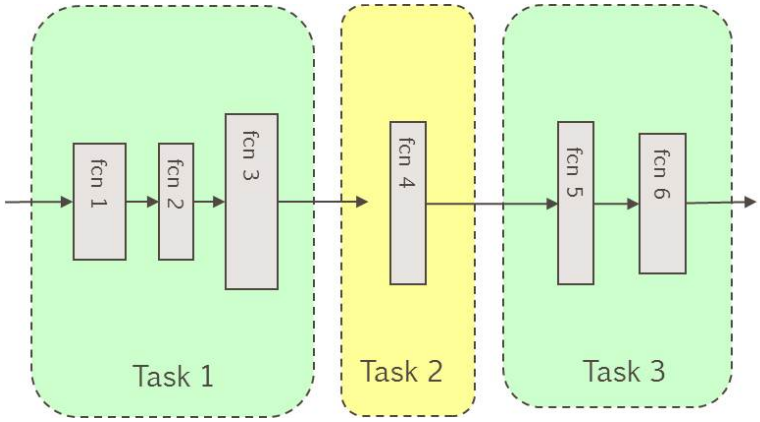

Figure 1.2: Software programs and tasks.

Running these tasks causes heavy load to the system. Meanwhile, most of these tasks are highly real-time critical in order to offer high quality of service, leading to a great interest for system designers to model the load of each task.

Research on task load modeling can be found in many fields. In computer science, CPU load gains a lot of interest. In [8], the authors evaluated linear models for predicting the Digital Unix host load. The analysis suggested that linear models like auto-regressive(AR) models, moving-average(MA) models [3] might be appropriate for predicting host load. Similar ideas of using

linear models show up in later research like [21]. Others may have nonlinear models like [17]. All these models involve previous task load data to predict current or future task load. One reason behind this is that these tasks do not change much for different runs; and CPU has its internal memory that will save data of the previous task to accelerate later runs. For LTE base station DSP hardware platform, the situation is different. First, DSP is a particular type of microprocessor designed to support numerically intensive tasks; little data will be saved for later runs, meaning little correlation should be expected between sequential runs. Thus in a model for tasks executing on DSPs, it is less possible to involve history items. The load of one task varies mainly with the task's input parameters, leading to a load model involving task parameters.

In this thesis, a specific type of data that is gathered from baseband signal processing in Ericsson base stations are provided. This is data stating how much computational load a certain task created, including traces of both task load and the tasks' input parameters. The task load is measured by the task running time. The input parameters can be the number of physical resource blocks, the modulation scheme, the number of antennas, etc. These parameters are believed to be related with the task load somehow but no clear knowledge is available. Also, there is no particular order for collecting the data. It is expected to analyze the data to explore the relationship between the task load and the parameters, and therefore be able to predict the task load.

The results are expected to be advantageous in several aspects:

- To better understand how the hardware is utilized, and estimate the resource headroom with current software program tasks.
- To give clues on more efficient use of the hardware resources with current software program tasks. For example, on a multi-core DSP platform, several tasks may be executed at the same time. A good schedule of these tasks that make the hardware fully used will give the best efficiency.
- To give clues on optimizing software design and hardware design.

1.2 Problem definition

A task usually takes in several parameters and then executes on some hardware platform. In this thesis, all tasks are always executed on the same digital unit hardware platforms (mainly DSPs), and software programs are fixed. Therefore, the task running time varies merely with its parameters, plus some noise from the system. By denoting a task as Task T, its

parameters as $p_1,\ p_2, \cdots$, its running time as Y and the noise as n_T, such a relationship is best expressed with a schematic model shown in Figure 1.3,

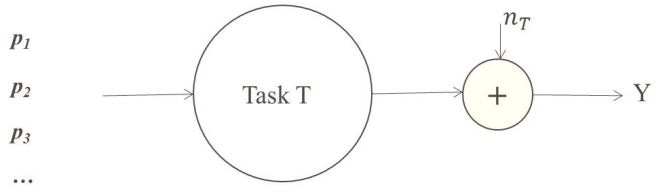

Figure 1.3: A block model for some specific Task T .

or mathematically,

$$Y = f_T(p_1, p_2, \cdots) + n_T \tag{1.1}$$

The running time Y is measured by hardware clock cycles. For example, with a clock speed of $1.2GHz$, a cycle is about $0.833ns$. The parameters $p_1,\ p_2, \cdots$ represent different physical parameters such as the modulation scheme and the number of receiving antennas and so on. These parameters may have different types: some being binary-valued, some being enumerating types, and others being best modelled as several values within a certain range. The number of parameters for some tasks can be quite large, mostly beyond 10. Different tasks usually have different sets of parameters, varying in both number of parameters and their types. No prior knowledge is available on the form of function f_T. The noise n_T is a random variable which models that the running time is affected by external factors (for example, interference from other tasks on other computing resources in the system). The noise is not necessarily to be the most well known white Gaussian noise. For each execution, the task takes in parameters $p_1,\ p_2, \cdots$, which can be treated as constant rather than random; however, the running time Y will be random due to the random noise n_T.

An explicit mathematical model with the task parameters to best estimate the running time for each task generally can be expressed as

$$\hat{Y} = \hat{f_T}(X_1, X_2, \cdots, X_m) \tag{1.2}$$

where $\hat{Y}$ is an estimate of the real measured value of the task T's running time. $X_1,\ X_2, \cdots,\ X_m$ are some or all the input parameters $p_1,\ p_2, \cdots$, or even their mathematical transforms. The function $\hat{f_T}$ can be of different forms.

Such an explicit model is expected to be accurate and simple. Basically, the model will be used to predict running time for a task given some parameters, therefore accuracy is necessary. A simple form of the model will make it easy for implementation. Also, the number of parameters taken

into the model is expected to be small. When not all these requirements can be met at the same time, some trade-off is necessary. So technically, to build such a model for some Task T, the following questions are to be answered:

- What is the proper form of $\hat{f_T}$?
  A proper form of $\hat{f_T}$ is expected. Since no prior assumptions on the relationships f_T is provided, a wide set of possible forms are available to choose, be it linear or nonlinear forms. It should be emphasized that $\hat{f_T}$ does not necessarily need to be exactly the same form as the real f_T. If there exists a f_T that is too complicated to grasp or describe in simple terms, it is best to approximate f_T by some simple mathematical function. As long as the simple $\hat{f_T}$ gives a $\hat{Y}$ close to Y, it is an acceptable form.
- Which parameters should be included into the model?
  When there are too many parameters for a task, probably a model with part of all parameters can be good enough to give a satisfactory model. This also makes it cost-efficient since it asks for less parameters as input. Sometimes transforms of the parameters (for example, high order items for a parameter) can be attractive to be included in the model as well.

1.3 Thesis objective

There are around 20 tasks in total, and each task has 16 to 20 parameters. Data on tasks' load and corresponding parameters are provided. Without prior knowledge of the parameters' relationship to the task's load, a pure mathematical relationship between the task's load and its parameters will be explored and a proper model will be built for each task. Also, an (guided) automatic tool will be designed, which takes the task data as input and gives a model as output for any task. An evaluation of the tool will be given with data extracted from real industrial systems. In the end a profile table containing all the tasks with their corresponding models will be reported as well.

1.4 Thesis outline

The rest of the thesis report is organized as follows. In Chapter 2, a preliminary analysis of the data is conducted, leading to some hints on methodology. Then in Chapter 3, techniques that can help build the models are introduced. Then in Chapter 4 further analysis on the data are carried

out for some typical tasks and proper models are built for them. Chapter 5 describes the design of an automatic tool to build models for tasks. Chapter 6 shows all the models for all given tasks. Chapter 7 gives a conclusion of our work and discusses future work on the topic.

Chapter 2

Preliminary analysis of the data

In this chapter, a preliminary analysis of the data is carried out. The data is extracted from real industrial devices. With no prior knowledge on the data, graphical methods are used as main approach to get an intuition of possible relationships behind the data, and lead us to techniques for further analysis.

2.1 Overview of the data

The data is extracted from baseband signal processing tasks. There are 17 tasks in total. Each task has different sets of parameters. A set of data samples for some typical task is shown in Figure 2.1.

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	Y
0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	18	2	0	1768
0	0	0	1	2	192	0	0	1	0	0	2	2	1	1	2	2	0	413
0	0	0	1	2	192	0	1	1	1	0	2	2	1	1	2	2	60	415
0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	2	2	0	413
0	0	0	1	2	196	0	0	0	0	0	2	0	1	1	32	2	0	2944
1	0	0	1	4	0	0	0	0	0	0	0	0	0	1	24	2	0	2271
0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	18	2	0	1773
0	0	0	1	2	192	0	0	1	0	0	2	2	1	1	2	2	0	415
0	0	0	1	2	160	0	0	0	0	0	2	0	1	1	20	2	0	1936
1	0	0	1	4	0	0	0	0	0	0	0	0	0	1	45	2	0	4040
0	0	0	1	2	192	0	0	1	0	0	2	2	1	1	2	2	0	415
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 2.1: A set of data samples for Task T.

This task has 18 input parameters denoted as $X_1, X_2, ..., X_{18}$, and running time is denoted as Y. A trace of the task running time and its

corresponding parameters is referred to as a data sample, and there are totally around 17000 samples. All the data only takes integer values. The ranges of the data set are summarized in Figure 2.2.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	Y
Min	0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	1	2	0	289
Max	1	0	0	1	4	838	0	1	1	1	0	2	2	1	1	100	2	104	9345

Figure 2.2: Data ranges of Task T.

It can be seen that:

- Parameters $X_2, X_3, X_4, X_7, X_{11}, X_{15}, X_{17}$ are constant;

- Parameters $X_1, X_8, X_9, X_{10}, X_{14}$ are binary;

- Parameters X_5, X_{12}, X_{13} vary within small ranges;

- Parameters X_6, X_{16}, X_{18} vary within big ranges.

Constant parameters do not contribute much help to a mathematical model, therefore are ignored for the current analysis; all other parameters are taken as normal numerical values at first.

A histogram of the input parameters are given in Figure 2.3. Here those constant parameters are ignored. It can been seen that most parameters do not show a uniform distribution. For example, there are more samples with $X_1 = 0$ than those with $X_1 = 1$. This is also a character of practice.

The data parameters are given without specific physical units. One reason is that the analysis of the data is purely from its mathematical properties, in the end making the automatic tool as a general tool for arbitrary tasks without relying too much on the real physical meanings; the other is that for some parameters there are no exact units, for instance, the value of X_3 here is a program flag taken from task software.

It should be also emphasized again that there is no particular order for these data samples. And there is no particular order in the data extracting process itself. This also means sorting or reordering these samples does not make any difference to the model building. All these is based on the fact that the output samples are independent to each other.

2.2 Scatter plot of the data

As is said in[5], "there is no statistical tool that is as powerful as a well chosen graph". Because graphs summarize data in ways that describe essential information more quickly and completely than tables of numbers, graphics

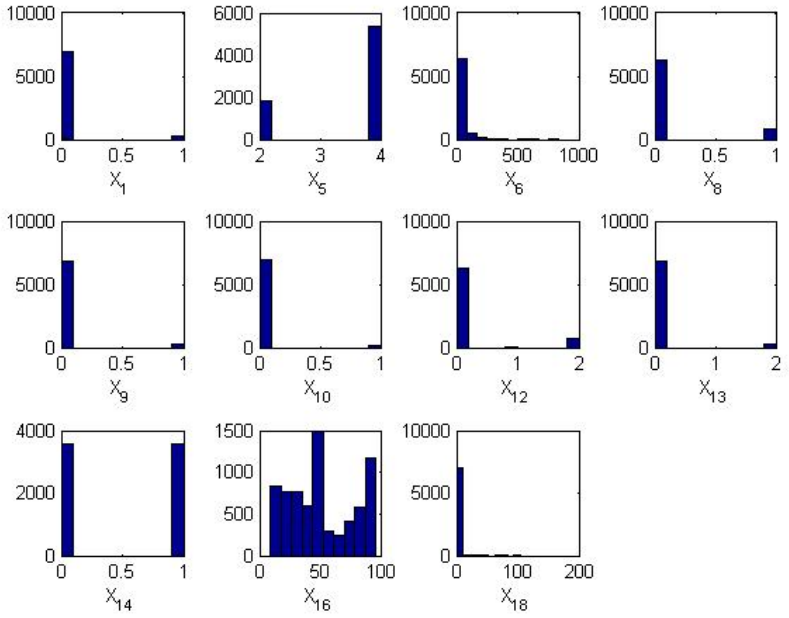

Figure 2.3: Histogram of parameters of Task T.

are important diagnostic tools for exploring data. Since no prior knowledge on the data is available, a graphic approach is an ideal choice for preliminary analysis. Among all plots, a scatter plot is our first choice. A scatter plot uses Cartesian coordinates to display values for two variables for a set of data. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis [19].

Figure 2.4 is a scatter plot of the data of a task, where constant parameters are ignored for the plot. With this plot, it seems obvious that there is probably a linear relationship between Y and X_1. This hints us to try linear regression methods with X_1 as a following step. For X_2, if $X_2 = 2$, the upper bound of Y is lower than that of $X_2 = 4$. Less clues are available on other parameters' relationship to Y. Further analysis is left to later chapters.

Another task has a scatter plot shown in Figure 2.5. It shows that X_1 affects Y significantly. The relationship is not likely a pure linear relationship. As X_1 goes beyond some value around 80, Y has a huge drop. If the data is separated into two groups according to the value of X_1, then for each group there is a linear relationship between X_1 and Y. This hints us to try a piecewise linear model.

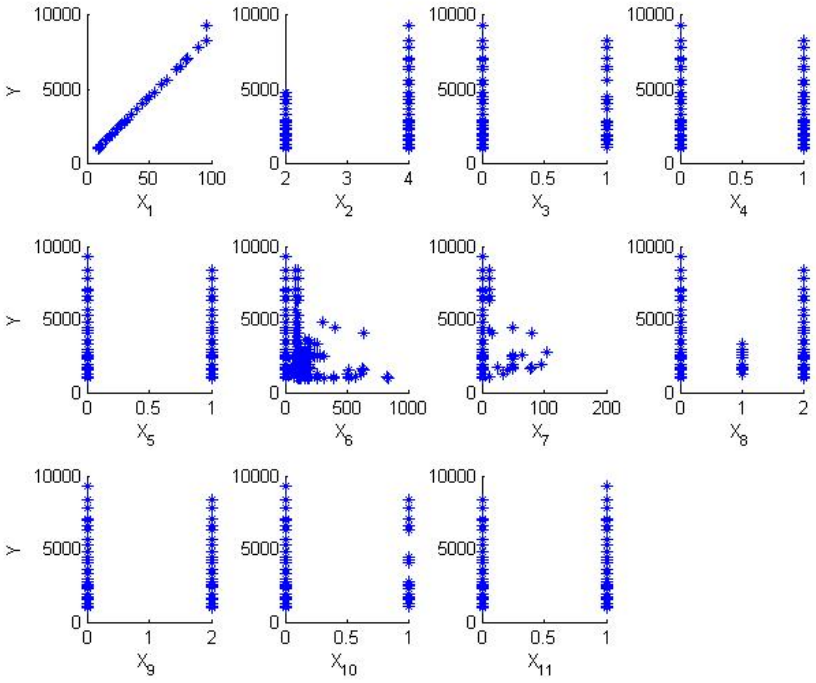

Figure 2.4: Scatter plot of Task T_1.

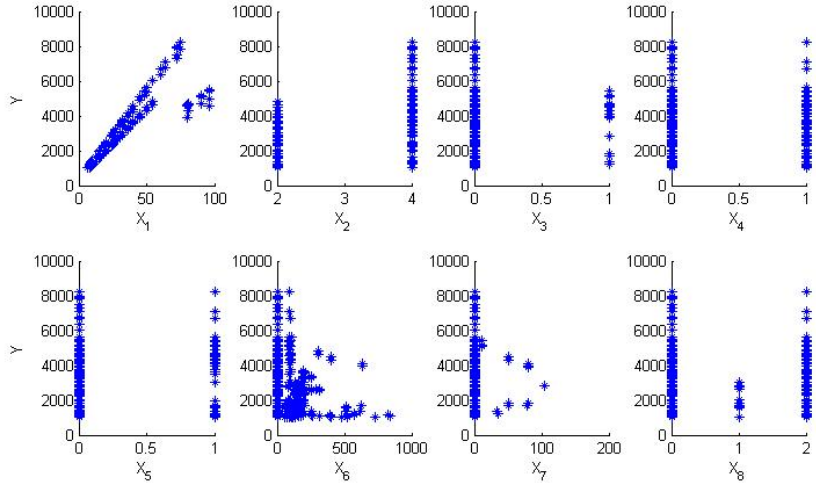

Figure 2.5: Scatter plot of Task T_2.

Another scatter plot is shown in Figure 2.6. In this plot, the relationship between Y and X_1 seems to be characterized by several rays. And in Figure 2.7, a similar relationship exists between Y and X_1, and between Y and X_3. The plots look pretty much like a tree with several branches, for each branch maybe a linear model can fit well. This gives us a hint that a tree-like model where data can be separated into several groups and then be handled separately may fit well.

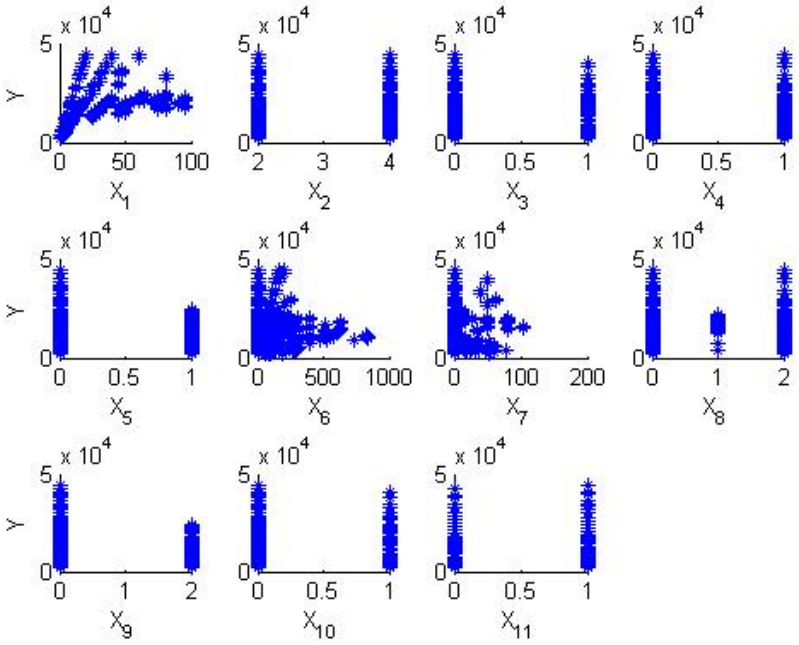

Figure 2.6: Scatter plot of Task T_3.

The scatter plots for the other tasks do not give more clear clues on possible relationships between task load and its parameters.

2.3 Samples with same parameters

Assuming that a model for one task is already built as well as an estimate $\hat{Y}$ of Y. One key measure of the model performance is the root-mean-square-error(RMSE), which can be expressed as

$$RMSE = \sqrt{E[(\hat{Y} - Y)^2]} \tag{2.1}$$

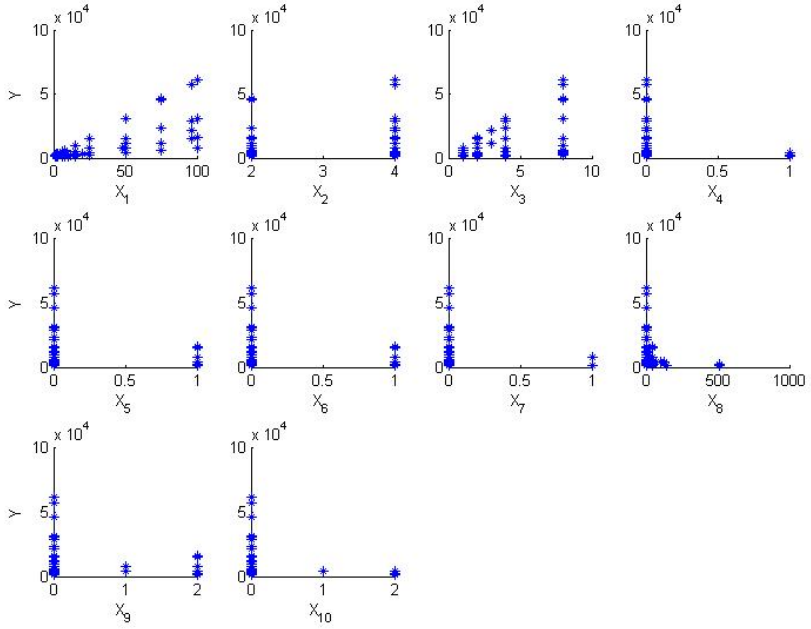

Figure 2.7: Scatter plot of Task T_4.

With an ideal model, RMSE should approach zero. But as mentioned in Equation 1.1, there is a noise n_T in the true relationship, so the zero RMSE rarely happens in practice.

No much information about the properties of the noise is available. According to the central limit theorem (CLT), it can be assumed that the noise is normally distributed, with zero mean and a variance of σ^2; furthermore the noise can be assumed to be independent from parameters, therefore the different parameters do not affect the distribution of the noise. Then for a good model, the RMSE is expected to approach the standard derivation σ.

The the standard derivation σ can be estimated from the data samples available. For most tasks, there is a great number of data samples available for analysis, and some samples have same parameters while different task running time. One example is shown in Figure 2.8.

The samples that have same parameters but different running time are caused mainly by this n_T. With the following notation,

- $Y_{1,1}, Y_{1,2}, ..., Y_{1,n_1}$ are n_1 samples with same parameters $(X_{1,1}, X_{1,2}, ...)$, and $\bar{Y_1}$ is the mean of these samples;

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	Y
1	4	0	0	0	0	0	0	0	0	1206
1	4	0	0	0	0	0	0	0	0	1280
1	4	0	0	0	0	0	0	0	0	1282
1	4	192	0	1	0	2	2	1	0	1278
1	2	0	0	0	0	0	0	0	0	1314
2	2	0	0	0	0	0	0	0	0	1654
2	2	0	0	0	0	0	0	0	0	1664
...	...	...	...	...	...	...	...	...	...	...

Figure 2.8: Samples with same parameters.

- $Y_{2,1}, Y_{2,2}, ..., Y_{2,n_2}$ are n_2 samples with same parameters $(X_{2,1}, X_{2,2}, ...)$, and $\bar{Y_2}$ is the mean of these samples;
- ...
- $Y_{m,1}, Y_{m,2}, ..., Y_{m,n_m}$ are n_m samples with same parameters $(X_{m,1}, X_{m,2}, ...)$, and $\bar{Y_m}$ is the mean of these samples;

an estimate s of σ can be calculated as

$$s = \sqrt{\frac{1}{n_1 + n_2 + \cdots + n_m} \sum_{j=1}^{m} \sum_{u=1}^{n_j} (Y_{j,u} - \bar{Y_j})^2} \tag{2.2}$$

which gives us a rough idea on how large a good model's RMSE should be.

2.4 Summary

In this chapter, a preliminary analysis is carried out on the data.

The input parameters can be constant, binary, or vary within some range. From the point of mathematical analysis, constant parameters are ignored for analysis; binary parameters and parameters that vary within a small range can be treated as categorical variables if necessary.

Scatter plots indicate that linear models, piecewise models and tree-based models will probably help.

Also the standard derivation of the noise is estimated from samples with exactly the same parameters, which can be helpful for later analysis on model's performance. It should be noted here that the assumption of a normally distributed noise is not solid and should be reexamined if necessary.

Based on the analysis in this chapter, linear regression, piecewise-linear regression and tree-based methods are candidate methods to analyze the

data to build proper models.

Chapter 3

Methodology study

In this chapter, several techniques are introduced to build models. Linear regression is explained first, followed by nonlinear methods including piecewise linear regression and tree-based methods.

3.1 Linear regression

Assuming variables (or predictors) $X_1, X_2, \cdots, X_m$ and a dependent variable or response Y are to be analyzed. A linear relationship between Y and the predictors can be described as

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \varepsilon \tag{3.1}$$

where $\beta_0, \beta_1, \cdots, \beta_m$ are constants referred to as the regression coefficients and ε is a random disturbance or error or noise. Equation 3.1 can be rephrased in matrix form,

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon} \tag{3.2}$$

where $\boldsymbol{X} = [1, X_1, X_2, \cdots, X_m]$ and $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_m]^T$.

Given data samples of $X_1, X_2, \cdots, X_m, Y$, where each of them is a $n \times 1$ vector,

$$\hat{\boldsymbol{Y}} = \boldsymbol{Xb} \tag{3.3}$$

and

$$\boldsymbol{b} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'Y} \tag{3.4}$$

where $\boldsymbol{b}$ is an estimate of the coefficients $\boldsymbol{\beta}$. It minimizes the error sum of squares $\varepsilon'\varepsilon$, irrespective of any distribution properties of the error. $\hat{\boldsymbol{Y}}$ is an estimate or fitted value of $\boldsymbol{Y}$. The calculation is based on the least squares method and details can be found in [9]. The residual is then given as $\boldsymbol{e} = \boldsymbol{Y} - \hat{\boldsymbol{Y}}$.

Equation 3.1 has a constant item β_0 and m first-order items. Linear models can also include higher order items, for example, second-order items like X_1^2, $X_1 \cdot X_2$, third-order items like X_1^3, $X_1^2 \cdot X_2$, etc. Generally, items like X^α where $\alpha \in \mathbb{R}$ are all allowed. These models are considered as linear models because the coefficients can be estimated in the same way as given in Equation 3.4.

3.1.1 The usefulness of the estimated model

To show the usefulness of a model described in Equation 3.3, a proper measure is necessary.

Assuming $\boldsymbol{X_0}$ is a specified $1\times(1+m)$ vector whose elements are of the same form as a row of $\boldsymbol{X}$ so that $\hat{Y_0} = \boldsymbol{X_0 b}$ is the fitted value at a specified point $\boldsymbol{X_0}$. For one sample, it is easy to see that

$$(Y_i - \bar{Y}) = (\hat{Y_i} - \bar{Y}) + (Y_i - \hat{Y_i}) \tag{3.5}$$

where $\bar{Y}$ is the average of $Y_i, i = 1, 2, \cdots$.

For all n samples together, the following equation holds:

$$\sum (Y_i - \bar{Y})^2 = \sum (\hat{Y_i} - \bar{Y})^2 + \sum (Y_i - \hat{Y_i})^2, \tag{3.6}$$

$$SST = SSR + SSE \tag{3.7}$$

and

$$\sum Y_i^2 = \sum (Y_i - \bar{Y})^2 + n\bar{Y}^2 \tag{3.8}$$

Here $\sum (Y_i - \bar{Y})^2$ is denoted as sum of squares about the mean(SST), $\sum (\hat{Y_i} - \bar{Y})^2$ as sum of squares due to regression(SSR) and $\sum (Y_i - \hat{Y_i})^2$ as sum of squares of residuals(SSE). Equation 3.7 shows that, of the variation in the Y's about their mean, some of the variation can be ascribed to the regression and some to the fact that samples do no all lie on the regression curve. In addition, $\frac{(\sum (Y_i))^2}{n} = n\bar{Y}^2$ is denoted as $SS(b_0)$, the sum of squares due to the existence of b_0, and $\sum Y_i^2$ as the total sum of squares.

The SSE can be a measure of model usefulness. If SSE is small, it means the difference between Y and $\hat{Y}$ is small. The ratio R^2=(Sum of squares due to regression)/(Sum of squares about the mean) gives similar measure too. If R^2 is close to unity, it means the sum of squares due to regression is much greater than the sum of squares of residuals, which is pleased to see. R^2 is formally named as coefficient of determination in statistics, measuring proportion of total variation about the mean $\bar{Y}$ explained by the regression.

Also an analysis of variance table can be constructed as Table 3.1. The last column is derived as the SS column divided by the df column, where

Source of Variation	Degrees of Freedom (df)	Sum of Squares (SS)	Mean Square (MS)
Due to regression	m	$\sum(\hat{Y}_i - \bar{Y})^2$	MS_R
Residual	n-(m+1)	$\sum(Y_i - \hat{Y}_i)^2$	MS_E
Total, about $\bar{Y}$	n-1	$\sum(Y_i - \bar{Y})^2$	
Due to b_0	1	$SS(b_0) = (\sum Y_i)^2/n = n\bar{Y}^2$	
Total	n	$\sum Y_i^2$	

Table 3.1: Analysis of Variance Table, the basic split

MS_R is the mean square due to regression and MS_E is the mean square due to due to error(residual). The table will be used for hypothesis testing later.

There are also criterions as Mallows C_p, Akaike Information Criterion(AIC), Bayes Information Criteria (BIC), etc. These criteria can be helpful to deal with the trade-off between the goodness of fit and the complexity of the linear model. Together with R^2 and SSE, these criteria offers various measures to quantify a model's usefulness. A more detailed discussion can be found in [6].

3.1.2 Hypothesis testing of linear regression

Hypothesis testing, or significance testing, is a method for testing a claim or hypothesis about a parameter in a population, using data measured in a sample. With this method, some hypothesis is tested by building a proper sample statistic and determining the likelihood that the hypothesis can be accepted. Hypothesis testing plays a fundamental role in the statistics[12].

In a linear model, many hypothesis is widely used to verify the model. With the assumption that the error $\boldsymbol{\varepsilon}$ is normally distributed as $\boldsymbol{\varepsilon} \sim N(0, \sigma^2)$, several different hypotheses are commonly considered in connection with the analysis of linear models. With the model given in Equation 3.1 and its estimated coefficients, one favourable investigated

hypothesis is "all the regression coefficients associated with the predictor variables are zero". In the following the procedure is explained in details.

First the null hypothesis H_0 and the alternative hypothesis H_1 are stated.

H_0: All the regression coefficients associated with the predictor variables are zero;
H_1: not H_0.

The hypothesis H_1 represents the model given in Equation 3.1, denoted as the full model(FM). The hypothesis H_0 represents a reduced model(RM) as

$$Y = \beta_0 + 0 \cdot X_1 + 0 \cdot X_2 + \cdots + 0 \cdot X_m + \varepsilon. \tag{3.9}$$

The statement can then be refined as

H_0: Reduced model is adequate;
H_1: Reduced model is not adequate compared to Full model.

With these two models FM and RM, their corresponding coefficients $\boldsymbol{b}$ and SSE can be estimated, denoted as $SSE(FM)$ and $SSE(RM)$. $SSE(RM)$ should be no less than $SSE(FM)$ because the additional parameters in the full model cannot increase the residual sum of squares, and $SSE(RM) - SSE(FM)$ represents the increase in the residual sum of squares due to fitting the reduced model. If this difference is large, then the reduced model is treated as inadequate. The decision is made with the help of a ratio

$$F = \frac{[SSE(RM) - SSE(FM)]/m}{SSE(FM)/(n - (m + 1))} \tag{3.10}$$

The ratio is referred to as the F-test. $SSE(RM) - SSE(FM)$ and $SSE(FM)$ each is divided by their respective degrees of freedom to compensate for the different number of parameters involved in the two models as well as to ensure that the resulting test statistic has a standard statistical distribution. For the full model, there are $1 + m$ parameters $(\beta_0, \beta_1, \cdots, \beta_m)$ to be estimated while for the reduced model there is only one β_0. (Note here β_0 for the two models may not be the same value.) Actually the F value can be calculated from the last column of Table 3.1.

Now the F value is compared with the tabulated value of F with $(m, n - (m + 1))$ degrees of freedom at the significance level α, which usually is set to some small value like 0.05, 0.01, etc. Accordingly, H_0 is rejected if

$$F \geq F_{(m,n-(m+1);\alpha)}, \tag{3.11}$$

or equivalently, if

$$p(F) \leq \alpha, \tag{3.12}$$

where $p(F)$ is the p-value for the F-test, which is the probability that a random variable having an F distribution with $(m, n-(m+1))$ degrees of freedom.

Another test is the T-test. If an estimate $\hat{\beta}$ of β is calculated, to test whether $\hat{\beta}$ is significantly different from a constant β_0, a t-statistic can be formed as

$$t_{\hat{\beta}} = \frac{\hat{\beta} - \beta_0}{s.e.(\hat{\beta})} \tag{3.13}$$

where $s.e.(\hat{\beta})$ is the standard error of the estimate $\hat{\beta}$, defined as

$$s.e.(\hat{\beta}) = \sqrt{\frac{\sum_{i=1}^{N} (\hat{\beta}_i - \bar{\hat{\beta}})^2}{N}} \tag{3.14}$$

Correspondingly the null hypothesis H_0 and the alternative hypothesis H_1 are

H_0: $\hat{\beta}$ equals β_0;
H_1: not H_0.

Then $t_{\hat{\beta}}$ is compared with tabulated value of T-test and make proper decision. A similar p-value for T-test can be associated with $t_{\hat{\beta}}$. More details can be found in [9].

Other tests can be conducted in the similar way.

3.1.3 Variables selection

Assuming in total m variables $X_1, X_2, \cdots, X_m$ and one response Y are available. A model as Equation 3.1 can then be built where all variables are included. Such a model will usually give a high precision. But if m is large, because of the cost involved in building a large model, a smaller model where some variables are excluded if it has similar performance as the model with all variables are preferred. Many methods can be used to select a subset of variables to build a small model, but there is no "best method" that handles all cases. Here correlation analysis, all-possible-regression method and stepwise method are introduced. Each method has its own advantages and disadvantages.

Correlation and partial correlation

Correlation is a measure of statistical dependence, and thus can be used to select the variables that are closely related to response. The most familiar

measure of dependence between two quantities is the Pearson product-moment correlation coefficient, commonly referred to as "the correlation coefficient"[18].

The population correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_X and standard deviations σ_X and σ_Y is defined as

$$\rho_{X,Y} \equiv \operatorname{corr}(X,Y) = \frac{\operatorname{Cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{3.15}$$

where E is the expected value operator and Cov means covariance.

$\rho_{X,Y}$ takes values between -1 and 1 and indicates the degree of linear dependence between the variables. As it approaches zero there is less of a relationship (closer to uncorrelated); the closer the coefficient is to either -1 or 1, the stronger the correlation between the variables are.

The population correlation can be estimated by the sample correlation coefficient. If measurements of X and Y are written as x_i and y_i where $i = 1, 2, \cdots, n$, then

$$r_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{3.16}$$

where $\bar{x}$ and $\bar{y}$ are the sample means and s_x and s_y are the sample standard deviations of X and Y.

Correlation has an important role in some variable selection methods. Assuming predictor variables are added to the model one by one. The first predictor variables can be chosen as the one which is most correlated with Y, i.e., the variable X_i whose $|r_{iY}|$ is the largest of all $|r_{lY}|, l = 1, 2, \cdots, m$. Suppose X_1 is chosen and the model

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon \tag{3.17}$$

is fitted. In the second stage, it is not wise to select the variable that has second largest correlation coefficient with Y. The reason is that the variables $X_i, i = 1, 2, \cdots, m$ could be correlated and one's correlation coefficient with Y can be affected by others.

To tackle this problem, new variables $X_{*,2}, X_{*,3}, \cdots, X_{*,m}, Y^*$ can be constructed by finding the residuals of X_2 after regressing it on X_1, that is, the residuals form fitting the model $X_2 = \alpha_0 + \alpha_1 X_1 + \varepsilon^{'}$, the residuals of X_3 after regressing it on X_1, and so on, respectively. The new variables represent those portions of the corresponding original data which have no

dependence on the values of the variable X_1. Now a new set of correlations which involve the starred variables is generated. These are called partial correlations, denoted as $r_{jY\cdot 1}$. In the second stage of the selection procedure the variable X_i whose partial correlation coefficient with Y is the greatest can be added to the model; that is, choose the variable X_i most correlated with Y after the effect of X_1 has been removed both from Y and X_i. After the second variable is chosen, say X_2, the third stage of the selection procedure involves partial correlations $r_{jY\cdot 12}$ between the residual of X_i regressed on X_1, X_2 and the residual of Y regressed on X_1, X_2. This process can be continued to any extent.

By comparing ordinary correlation and partial correlation, for example, r_{2Y} and $r_{2Y\cdot 1}$, three results can potentially occur. When partial and ordinary correlations are approximately equal, it suggests that the relationship between X_2 and Y cannot be explained by X_1; when partial correlations is closer to zero than ordinary correlations, it suggests no much improvement if both X_1 and X_2 are taken into the model. The case that partial correlations are farther from zero than ordinary correlations rarely happens.

All-possible-regressions

All-possible-regressions means checking all possible regression equations with all available sets of variables. The procedure try fitting each possible regression equation which involves a constant item plus any number of the m available variables. Each variable can be, or not be, in the equation. If there are m variables, there are $2^m - 1$ distinct regression equations. Each regression equation is assessed according to some criterion, be it R^2 ,C_p, AIC or other criterion. The best regression equation can be chosen as the final model.

This method guarantees that you will find the "best" model, since it looks at all the possible models. But there are cases where a small increase in the criterion, for example, R^2 increasing from 0.92 to 0.93, comes with the addition of several variables. In such situations, manually choice is always involved in the decision. Besides, the methods needs to deal with too many equations if m is large.

Stepwise linear regression

The general idea behind the stepwise regression procedure is to build our regression model from a set of candidate predictor variables by entering and removing predictors in a stepwise manner into our model until there is no justifiable reason to enter or remove any more[20]. By the "justifiable

reason" different criterions as stated above in Section 3.1.1 can be used. Since in this thesis, a regression model is constructed mainly for prediction, the criterion is chosen as SSE. Correspondingly, in each step, the p-value for an F-test of the change in the sum of squared error by adding or removing the term will be calculated. The procedure can go forward from a simple model and then try adding variables in, or go backward from a big model and then try removing variables out, or both way can be utilized. The bidirectional procedure is detailed here.

As a preliminary step, a starting model needs to be specified. Any model can be set as a starting model, while a constant model is usually preferred. Also an Alpha-to-Enter (α_E) significance level and an Alpha-to-Remove (α_R) significance level to help decide whether to enter or remove one variable into the model need to be specified.

Step 1 Fit each of the one-predictor models that is, regress Y on X_1, regress Y on X_2, ..., and regress Y on X_m. For each model a F-test that the coefficient of the variable to be included is zero is conducted. Of those predictors whose F-test p-value is less than α_E, the first predictor put in the stepwise model is the predictor that has the smallest F-test p-value. If no predictor has a F-test p-value less than α_E, stop.

Step 2 Assuming X_1 was chosen to be included in the model. Now, fit each of the two-predictor models that include X_1 as a predictor that is, regress Y on X_1 and X_2, regress Y on X_1 and X_3, ..., and regress Y on X_1 and X_m. Of those predictors whose F-test p-value is less than α_E, the second predictor put in the stepwise model is the predictor that has the smallest F-test p-value.

If no predictor has a F-test p-value less than α_E, stop.

If one predictor, say, X_2 is entered into the stepwise model, step back and see if entering X_2 into the stepwise model somehow affected the significance of the X_1 predictor. That is, check the F-test p-value for testing the coefficient of X_1 in the new two-predictor model is zero. If the F-test p-value is greater than α_R, remove X_1 from the stepwise model.

Step 3 Consider adding one more predictor from the other available predictors into the current model. Calculate the F-test p-value for the new predictors and choose the smallest one among those lower than α_E. If no predictor enters, stop; otherwise take the predictor into the model and conduct hypothesis on predictors that already in the model to see whether they can be removed.

Continue the steps as described above until adding an additional

predictor does not yield a F-test p-value below α_E and removing an existing predictor does not yield a F-test p-value above α_R, stop.

Step 4 When the procedure is stopped, the final model is chosen as our result.

Stepwise regression is a common choice when the number of variables is relatively large. Many mathematics software provides its implementation. It is easy to understand and easy to use. However, the method has some drawbacks. It does not guarantee an optimal result in many cases. It only provides a single final model in the end, although there are often several equally good models. The order that the variables are taken into the model should not be over-interpreted. Since the entering or removing of variables are based on hypothesis testing, one should not jump to the conclusion that all the important predictor variables for predicting Y have been identified, or that all the unimportant predictor variables have been eliminated.

Another point is that stepwise regression does not take into account a researcher's knowledge about the predictors. This is advantageous since no prior knowledge of the predictors is available. If some predictors are known to be important, then it is necessary to force the procedure to include such predictors.

3.2 Piecewise-linear regression

From the analysis in Chapter 2, nonlinear relationships may exist among data. To deal with such cases, nonlinear techniques are considered. Piecewise-linear(PWL) regression is a very old method and has been widely used in many fields including data mining, image processing, etc. By doing this, the basic problem is transformed from a single nonlinear equation into several linear equations, therefore linear theory can be applied.

A PWL function consists of a collection of linear mappings, for each segment of the function exactly one. Each mapping is only valid in a certain interval if there is only one variable. The PWL method works for high dimension cases too but in this thesis only one-variable situation is considered. The boundaries between the segments are therefore denoted as breakpoints.

If the breakpoints are known in advance, the data can be separated into several groups according to the breakpoints, and for each group linear regression methods may be applied, making it simple to solve. Figure 3.1 shows such a situation. The leftmost plot is the original data. If the breakpoint is known to be 6 alone, the data can be separated into two pieces

and a linear regression on each piece can be applied; if the breakpoints are known to be 6 and 10, 3 lines can be fitted.

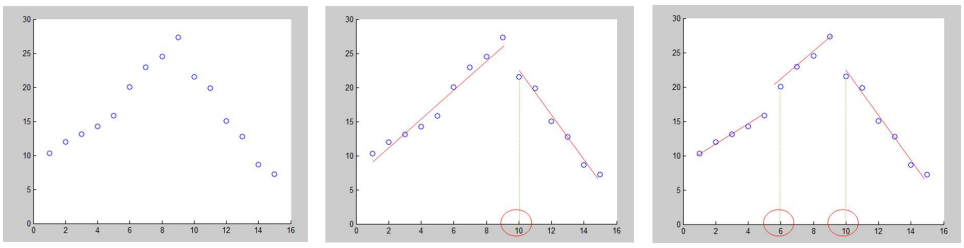

Figure 3.1: PWL with known breakpoints.

If the breakpoints are unknown, more analysis is needed to decide the breakpoints and build the model. In [11], PWL method is used in data mining for time series data. The author gives a summary of three major approaches to time series segmentation, named as Sliding-window, Top-down and Bottom-up, respectively. The data here is not time series data but the approaches are well worth considering.

The Top-down algorithm starts from a large interval and try splitting it into smaller intervals recursively until some user-specified threshold is met. A schematic plot is given in Figure 3.2, where it starts from one single piece and then split the piece into 2 and more pieces.

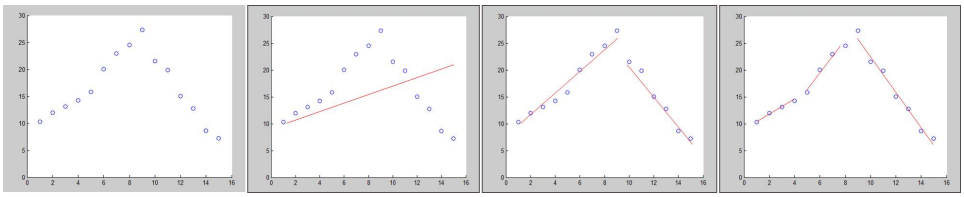

Figure 3.2: PWL with unknown breakpoints: Top-down approach.

The Bottom-up algorithm moves in the opposite direction, starting from many small pieces and try merging into larger pieces until some threshold is met. A schematic plot is given in Figure 3.3, where it starts from many small pieces and then merges the small pieces into larger pieces.

The Sliding-window algorithm works by taking the first data point of a time series as the starting point, then attempting to include the data to the right with increasing longer segments. If at some point, including the right data point results in an error beyond the predefined threshold, the point is set as the starting point of next segment; previous data points then forms a piece and is fixed. Form the new starting point, the same process continues until the entire data points have been handled. In Figure 3.4, it starts from

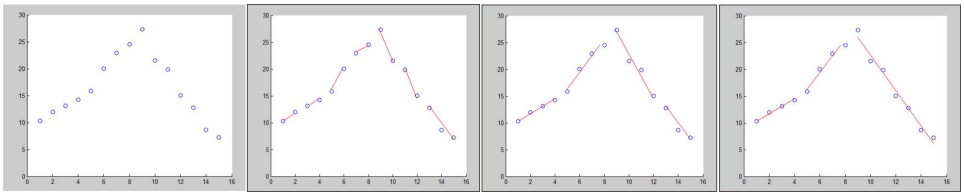

Figure 3.3: PWL with unknown breakpoints: Bottom-up approach.

the leftmost data point and tries taking the data points to the right in to do a linear regression; when trying to take the sixth data point, the error is beyond the threshold so the first five data points is set as a piece and applies a linear regression; the sixth data point is set as another starting point and the process continues.

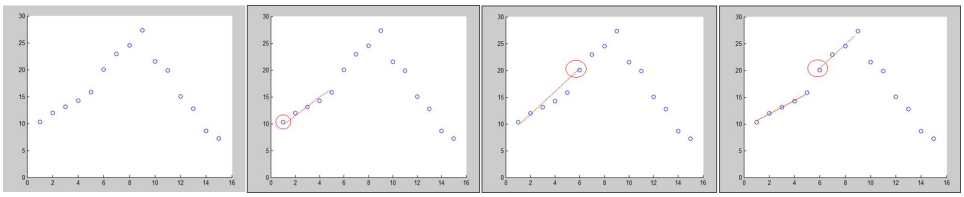

Figure 3.4: PWL with unknown breakpoints: Sliding-window approach.

Neither of these methods are guaranteed to give globally optimal solution in the end. The Sliding-window approach is attractive to us for its great simplicity and intuitiveness, and its good performance on noisy data.

PWL models were criticized by its lack of explicit analytical representation and the need to store an immense amount of information on functions in order that the linear equations over each interval can be retrieved for computation purposed. Now there are compact explicit PWL expressions to solve the problems, see[7],[10]. These expressions often ask for a clear knowledge of breakpoints, and in this thesis the compact expression does not provide much convenience for use, so it is not discussed here.

3.3 Tree-based models

Another particular kind of nonlinear method is predictive trees, with two basic varieties named as regression tree and decision tree. The models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. As a result, the partitioning can be represented graphically as a tree, with a simple model at each tree leaf[13].

Let us restudy the problem with m variables $X_1, X_2, \cdots, X_m$ and one response Y in the tree-based method approach. In theory, the solution is simply a partition of the $\boldsymbol{X}$ space into k disjoint sets, $A_1, A_2, \cdots, A_k$, such that the predicted value $\hat{Y}$ is close to Y for each set. There are different algorithms available for the partitioning, like CART[4] and M5[16], etc. In this thesis, due to limited time and the specific problem, only simple tree-based models are considered. Particularly, the thesis only considers splitting on discrete variables. As seen in Chapter 2, there are parameters which take only limited number of values, therefore making it easy to try all possible sets according to the variable values. In Figure 3.5, the data is first separated into two sets according to the value of X_2; then the data set with $X_2 = 2$ is further separated into three sets according to the value of X_3. For each small data set, a linear regression equation is built.

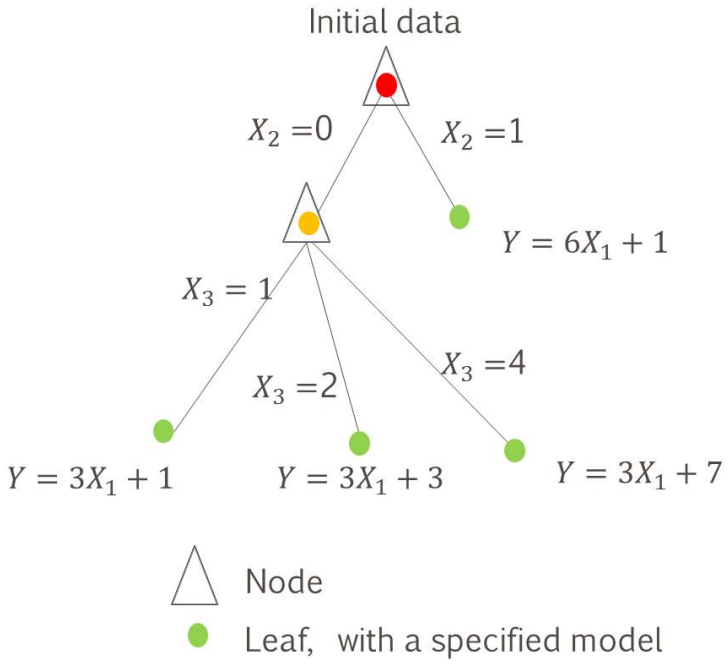

Figure 3.5: An example of tree-based model.

A key point in tree-based models is to decide which variables can be the nodes. The problem is similar to variable selection discussed earlier. As there are many variables available, it is not feasible to try every possible tree. Correlation and partial correlation together with variables' ranges can be applied as the criterion for node selection.

The tree-based method is known for its simplicity and efficiency when dealing with large number of variables and cases. The disadvantage is that

global optimal solution is not guaranteed.

3.4 Summary

This chapter studies several techniques that may help build feasible models for given tasks. Linear regression is the basic technique. Piecewise and tree-based methods extends the possibility to study nonlinear relationships. Piecewise regression can be seen as a simplified regression tree model where the branches are replaced by breaking points while with tree-based models, more complicated relationships can be explored where multiple variables are involved.

Variables selection is necessary for all methods. Correlation analysis provides a basic tool while hypothesis testing can help in many situations. It should be emphasized that the selection results based on these methods are not necessarily to be related to practical physical meaning, it is more likely a mathematical explanation. This satisfies our assumption that no prior knowledge on the data is available.

Chapter 4

Data analysis and model building for real tasks

In this chapter, the data of several tasks is fully analyzed and proper models are built, step by step. As a starting point, a general model building process is discussed and the performance criterion is decided to choose acceptable models. Then several typical tasks are analyzed to build models. The analysis also gives intuition on the design of an automatic model building tool. Matlab(R2014a) is the main tool and it offers many functions like $fitlm$ and *stepwiselm* to accelerate our process. The ideas behind these functions are basically the same as stated in Chapter 3. More details can be found in [14].

4.1 Model building procedure: general

The model building procedure should be carefully designed. Before trying models, graphical skills can help us get an intuition of what the data looks like. Correlation analysis also gives some impression on the data's relationship, and due to reasons stated in previous chapter, partial correlation will be our preference. As there may be many possible models, some criterion needs to be set for choosing acceptable models. The criterion should apply to different models. Then possible models can be tried, be it linear or nonlinear. Here first-order and second-order linear models, piecewise-linear models, or one-node-tree model will be tried. There is no specific order to try these models, but as the complexity increases from linear to nonlinear models, linear models will be explored first. After an acceptable model is built, new data is used to validate the model's usefulness. In our case, the sample data is separated into two parts at the very beginning, 70%

of the data will be used for building the model, while the remaining 30% will be used to validate the model. If the validation proves that the model works fine with validation data, the model is accepted as the final result of the corresponding task.

4.2 Model performance criterion

As discussed in Chapter 2, RMSE is a reasonable criterion for model performance. No matter which method is used to build a model, the outcome model is always associated with a RMSE. The RMSE is calculated as

$$RMSE = \sqrt{E[(\hat{Y} - Y)^2]} \approx \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n}} \tag{4.1}$$

A good model's RMSE should try to approach the estimated standard derivation s, which can be calculated from given data sample as shown in Section 2.3. Therefore in the analysis below, our main criterion for an acceptable model is set as

$$RMSE < 2s. \tag{4.2}$$

The threshold $2s$ is chosen by experiment, without much theoretical derivation. Within linear models, as introduced in Section 3.1.1 and Section 3.1.3, several other criteria are available. R^2 is widely used to compare linear models with same parameters; SSE, AIC, BIC are mostly used in stepwise linear regression procedure. These criteria act as supplement criteria in our model building process.

4.3 Task HD

In this section Task HD is analyzed. There are 17 parameters available, of which 11 are constant and therefore ignored. The remaining parameters are denoted as variables $X_1, X_2, \cdots, X_6$, and the measured running time is denoted as variable Y, and there are 213 samples in total. A small part of the samples is shown in Figure 4.1.

4.3.1 Setting criterion

First, calculate an estimate s of the error variance using all samples holding same parameters, as stated in Chapter 2. The calculation gives a result of

X_1	X_2	X_3	X_4	X_5	X_6	Y
2	2	0	192	0	0	3508
2	2	1	192	60	1	3554
2	2	0	192	0	0	3650
2	2	0	192	0	0	3621
8	4	0	92	0	0	2529
2	2	0	192	0	0	3545
45	2	1	636	80	1	7820
60	4	0	92	0	0	2668
40	4	0	88	0	0	2558
45	2	1	636	80	1	7816

Figure 4.1: A small part of the data set for Task HD.

$s = 41.29$. The criterion of an acceptable model for Task HD is then set as

$$RMSE < 2s = 82.58 \tag{4.3}$$

4.3.2 Scatter plot and correlation table

A scatter plot of the data set is given in Figure 4.2.

The plot obviously indicates a linear relationship between Y and X_4. To make sure of this, the partial correlation of the parameters with Y is calculated, shown as Figure 4.3.

It can be seen that X_4 and Y has a big partial correlation coefficient; X_1 has a medium one; other parameters have very small coefficients. This leads us to try linear models first.

4.3.3 Models trial

As there are 6 parameters in total, the all-possible-regression method can be applied. Figure 4.4 shows the drop of RMSE as the number of parameters in the model increases. At each point, the y-axis value is the smallest RMSE that can be reached with all possible models when the number of parameters is fixed as the corresponding x-axis value. The figure shows that all models can give a RMSE smaller than $2s$, and as the number of parameters increases from 1 to 6, the model's RMSE approaches s. But by adding parameters, the RMSE drop is small. A single parameter model holds a RMSE as

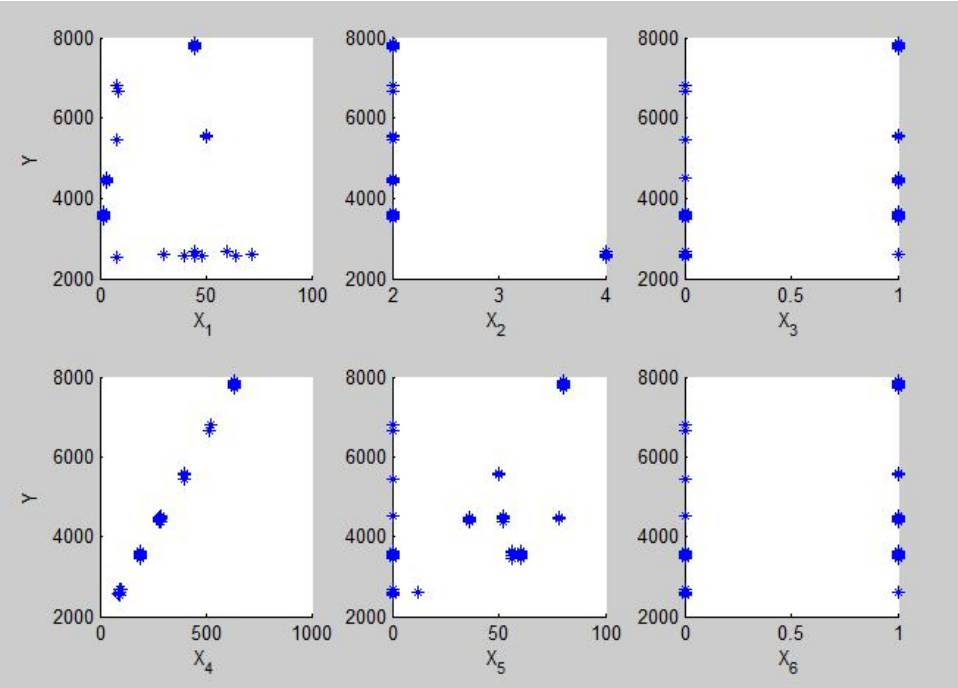

Figure 4.2: Scatter plot of the data set for Task HD.

	X_1	X_2	X_3	X_4	X_5	X_6
r_{iY}	0.0343	0.0266	0	0.9967	-0.1278	0

Figure 4.3: Correlation analysis of Task HD.

43.1661, and a two-parameter model holds 42.5411, giving a small drop of 1.45%. After 4 parameters have been included, adding more gives even less improvement. According to the criterion, a one-parameter model is already acceptable. Among all one-parameter models, the model that has X_4 gives the smallest RMSE. This matches the intuition of Figure 4.2 as well as the correlation analysis result.

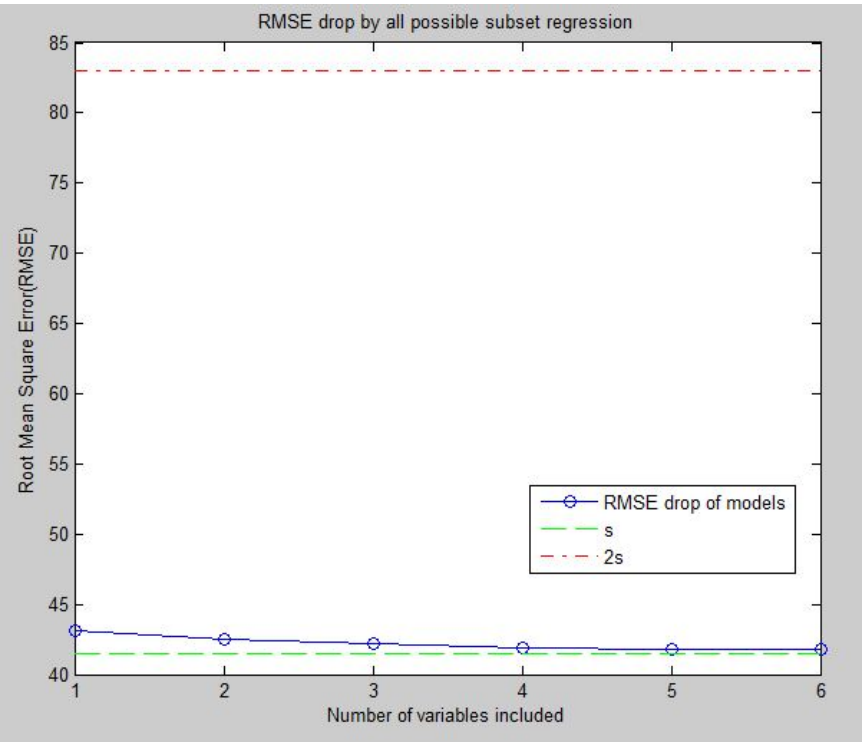

Figure 4.4: All-possible-regression modeling of Task HD.

The model turns out to be in the form

$$\hat{Y} = b_0 + b_4 X_4 \tag{4.4}$$

where coefficients are given in Figure 4.5. In Figure 4.5, "SE" represents the standard error of the estimated coefficient value, "tStat" represents the t-statistic for a test that the coefficient is zero, and the "pValue" is the corresponding p-value for the t-statistic. The smaller p-value the estimate has, the more reliable it is.

The model gives Figure 4.6. The model fits well with the real value. The residual plot is Figure 4.7 and no obvious pattern shows up, which is pleased to see.

Now stepwise linear regression method is applied. By setting $\alpha_E = 0.001$ and $\alpha_R = 0.05$, the method gives exactly the same result as above, and

	Estimate	SE	tStat	pValue
b_0	1701.7	7.4628	228.02	2.5316e-189
b_4	9.5845	0.027665	346.44	5.6321e-216

Figure 4.5: All-possible-regression coefficients of Task HD.

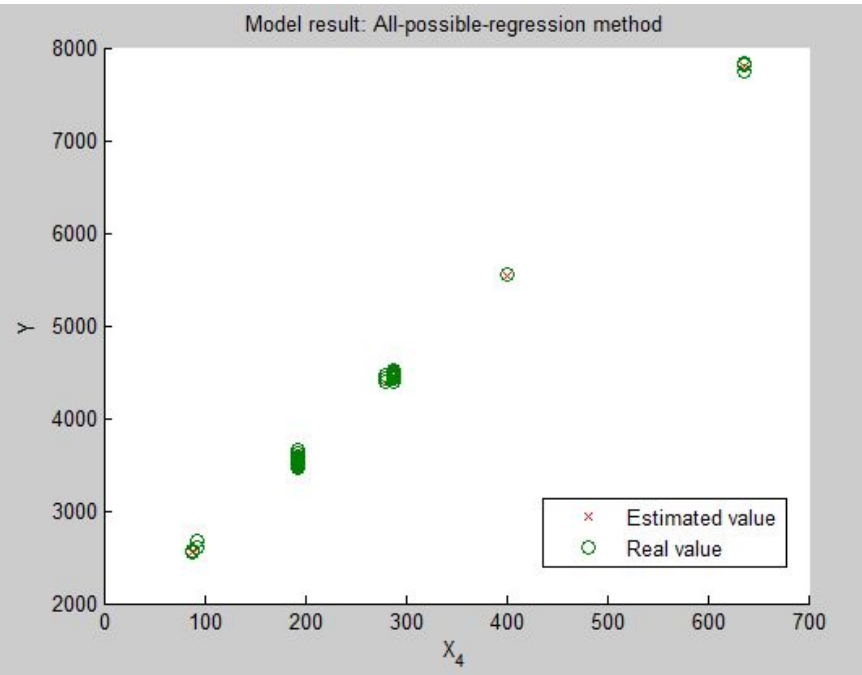

Figure 4.6: All-possible-regression result of Task HD.

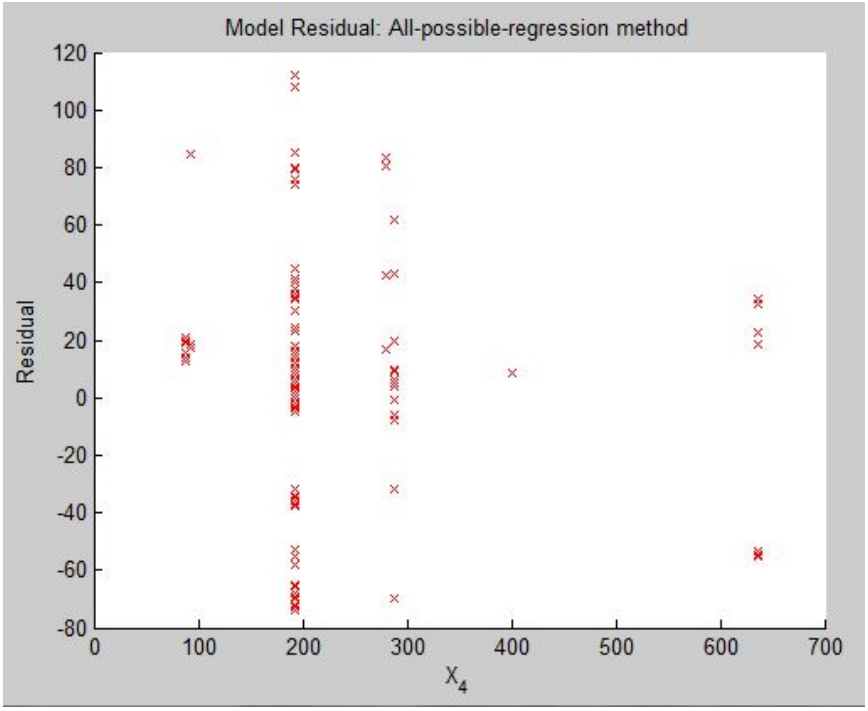

Figure 4.7: All-possible-regression residual of Task HD.

the selection is automatic except for our setting α_E and α_R at the very beginning.

4.3.4 Model validation

Let us further check the model with validation data. Figure 4.8 gives the validation plot, and the RMSE calculated for the model validation is $RMSE = 50.1574$, meeting the $2s$ criterion. The conclusion is that the linear model above is satisfactory, and it is taken as our model for Task HD.

4.4 Task HqD

In this section Task HqD is analyzed. There are 17 parameters available, of which 10 are constant and therefore ignored. The remaining parameters are denoted as variables $X_1, X_2, \cdots, X_7$, and the measured running time is denoted as variable Y. For this task, a very limited number of samples are provided: 40 in total. For Y, the smallest is 1488 while the largest is 13297.

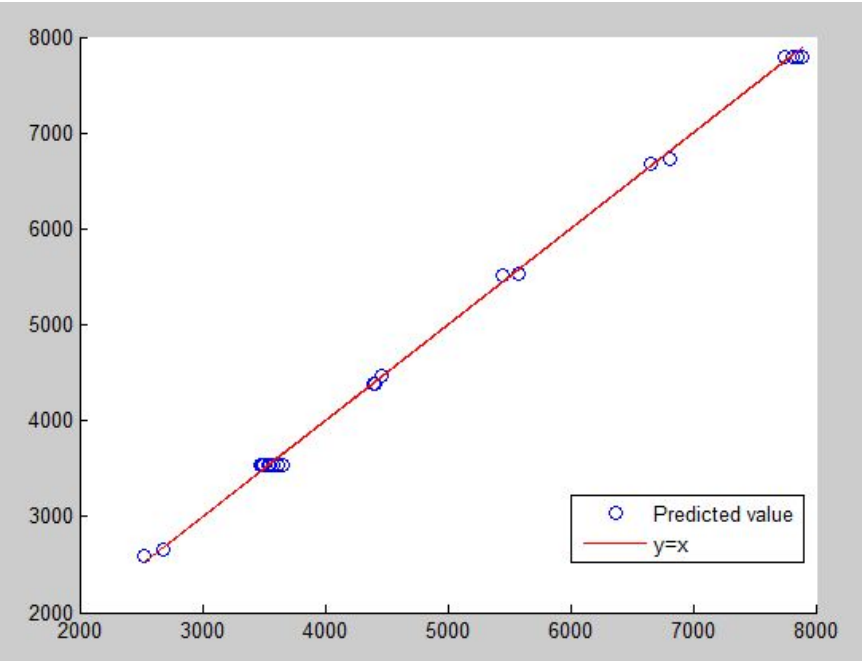

Figure 4.8: Validation of Task HD model.

4.4.1 Setting criterion

The samples with same parameters give an estimate as $s = 103.65$, therefore the criterion for an acceptable model is set as $RMSE < 2s = 207.30$.

4.4.2 Scatter plot and correlation table

The scatter plot of the data set is given in Figure 4.9. It seems that Y has a relative strong linear relationship with X_4. By checking the partial correlation table as in Figure 4.10, Y may has a strong relationship with X_4, X_2.

4.4.3 Models trial

Linear models are tried first. With the all-possible-regression method, Figure 4.11 shows that all the models give an RMSE beyond $2s$. Therefore even if all the parameters are included in the model, i.e., a 7-parameter first order linear model, it is still inadequate for the task. Stepwise linear regression gives similar results.

Then second-order linear models are considered, i.e., taking interactive

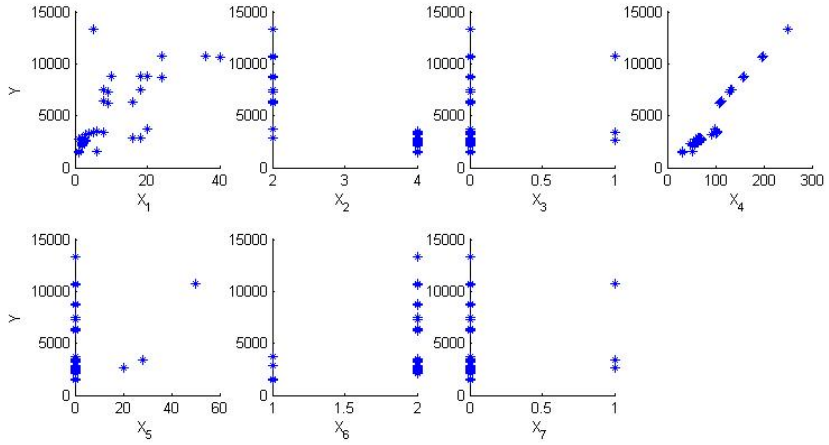

Figure 4.9: Scatter plot of the data set for Task HqD.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
r_{iY}	-0.1182	-0.6340	0	0.9228	0.1884	0.3447	0

Figure 4.10: Correlation of the data set for Task HqD.

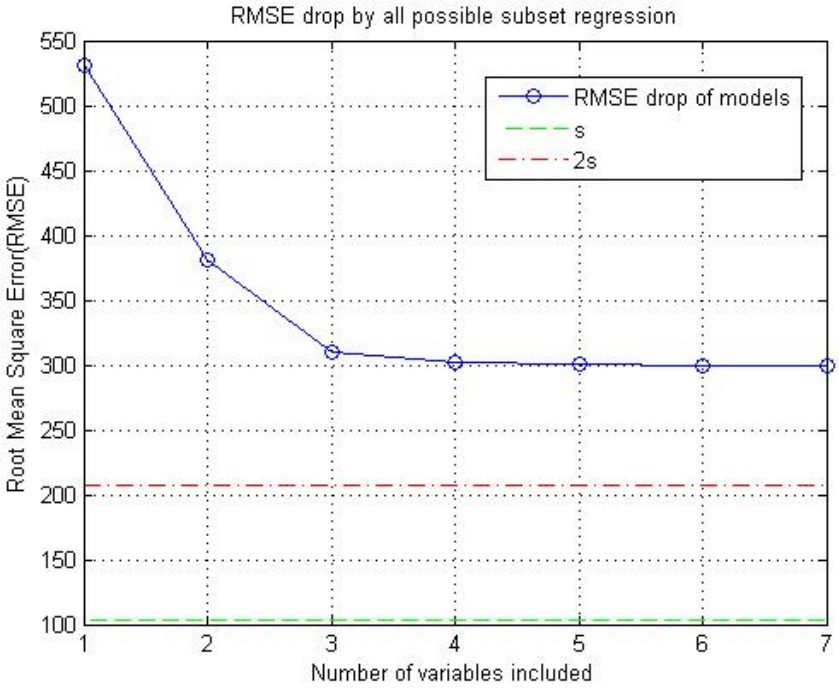

Figure 4.11: All-possible-regression of Task HqD.

items and square items into consideration. There will be 2^{56} possible models, making it infeasible to try all these models. Stepwise linear regression is used to try second-order linear models. Starting with a constant model and setting $\alpha_E = 0.001$ and $\alpha_R = 0.05$, the method ends with a second-order linear model. The model has an RMSE of 113.38, which satisfies the criterion. The model has a form as

$$\hat{Y} = b_0 + b_2 X_2 + b_4 X_4 + b_6 X_6 + b_{24} X_2 X_4 + b_{46} X_4 X_6 \tag{4.5}$$

where

	Estimate	SE	tStat	pValue
b_0	1508.5	438.45	3.4406	0.0015546
b_2	176.71	70.013	2.5239	0.016442
b_4	25.301	5.9447	4.2561	0.0001545
b_6	-556.84	151.95	-3.6646	0.00083699
$b_{2,4}$	-13.976	0.7225	-19.344	6.3321e-20
$b_{4,6}$	26.514	2.4263	10.928	1.1465e-12

Figure 4.12: All-possible-regression of Task HqD.

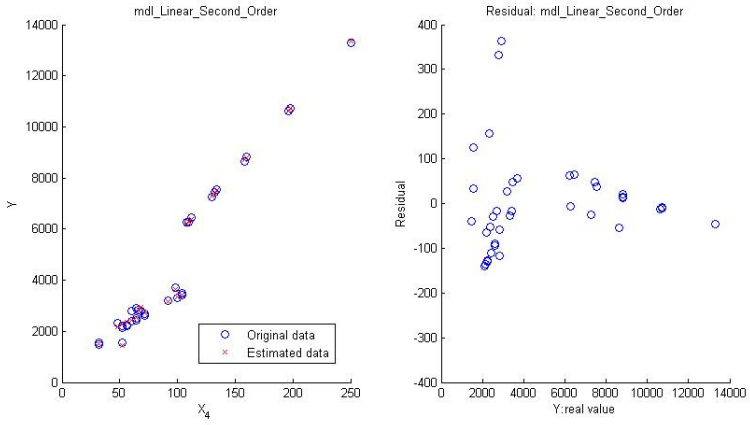

Figure 4.13: Second-order linear model of Task HqD.

The result is shown in Figure 4.13, as well as the residuals. The estimated value and the original value seems to match well. The residual plot shows that the model has relatively large residual when Y is small, and very small residual when Y is large. This may indicate a piecewise model. Let us try a piecewise linear model now. The variable according to which the data is

separated into pieces is decided by correlation coefficient with Y and the variable's range. A model relying on X_4 is built, in the form of

$$\hat{Y} = \begin{cases} 795.19 + 26.60X_4; \text{ if } X_4 < 108 \\ 845.85 + 49.81X_4; \text{ if } X_4 >= 108 \end{cases} \tag{4.6}$$

This model gives a RMSE of 165.80, which also satisfies our criterion while bigger than that of second-order linear model. Figure 4.14 shows the result.

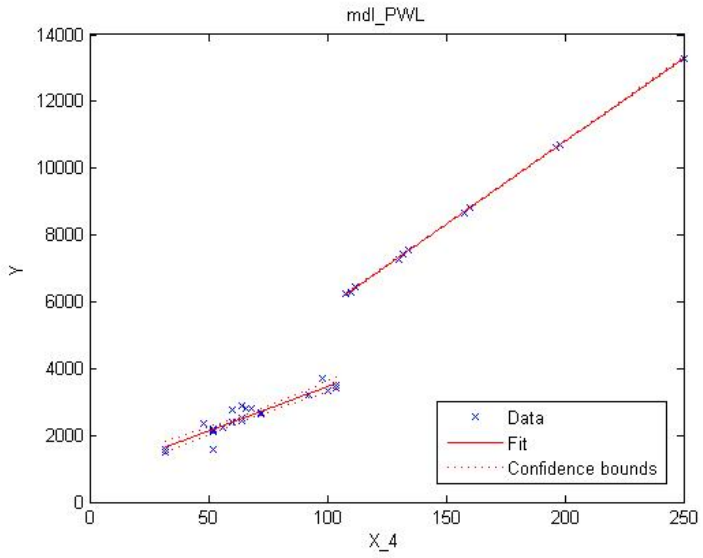

Figure 4.14: Piecewise-linear model of Task HqD.

Since there is no particular reason to reject any of the two models, here the second linear model is taken as the result simply because of its smaller RMSE, and do the validation.

4.4.4 Model validation

With validation data, the model gives the result shown in Figure 4.15, with a $RMSE = 72.36$, which is satisfactory. So Equation 4.5 is taken as the result for Task HqD.

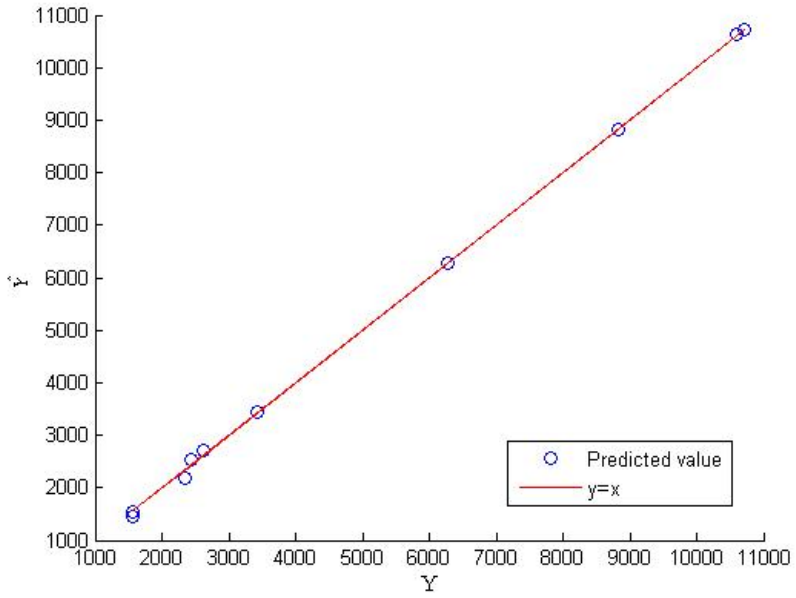

Figure 4.15: Validation of second-order linear model of Task HqD.

4.5 Task Ti

In this section Task Ti is analyzed. There are 17 parameters available, of which 7 are constant and therefore ignored. The remaining parameters are denoted as variables $X_1, X_2, \cdots, X_{10}$, and the measured running time is denoted as variable Y. There are 7301 samples in total. For Y, the smallest is 18097 while the largest is 19299.

4.5.1 Setting criterion

The samples with same parameters give an estimate as $s = 34.39$. The criterion for acceptable models is set as $RMSE < 2s = 68.78$.

4.5.2 Scatter plot and correlation table

The scatter plot is given in Figure 4.16. It seems quite obvious that there is a linear relationship between Y and X_1. The partial correlation table is given in Figure 4.17.

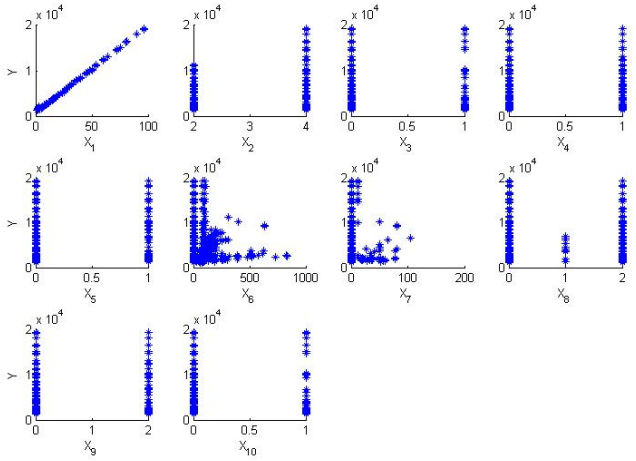

Figure 4.16: Scatter plot of the data set for Task Ti.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
r_{iY}	0.9993	0.0839	0.0317	-0.0776	0	-0.1837	-0.0113	0.0758	0	0.0231

Figure 4.17: Correlation of the data set for Task Ti.

4.5.3 Models trial

A linear model trial is carried out by stepwise linear method. With first-order models, $RMSE = 171.50$ is reached; with second-order models, $RMSE = 123.99$ is reached. Neither matches our criterion.

Piecewise-linear regression method is tried next. X_1 is chosen as the regressor, based on its largest absolute correlation value with Y or intuitions from the scatter plot. The idea of sliding-window algorithm is implemented, taking the data points that have minimum X_1 values(i.e., data points with $X_1 = 1$ and $X_1 = 2$) to build initial pieces. The algorithm gives a piecewise-linear model with five pieces, with the breakpoints of $4, 27, 54, 80$, shown in Equation 4.7.

$$\hat{Y} = \begin{cases} 907.3 + 363.03X_1; \text{ if } 1 \leq X_1 < 4 \\ 897.34 + 180.99X_1; \text{ if } 4 \leq X_1 < 27 \\ 1129.8 + 183.78X_1; \text{ if } 27 \leq X_1 < 54 \\ 1536.3 + 181.37X_1; \text{ if } 54 \leq X_1 < 80 \\ 1721.6 + 182.45X_1; \text{ if } 80 \leq X_1 < 96 \end{cases} \tag{4.7}$$

The piecewise-linear model has $RMSE = 38.39$, which perfect satisfies our criterion. The result plot is shown in Figure 4.18.

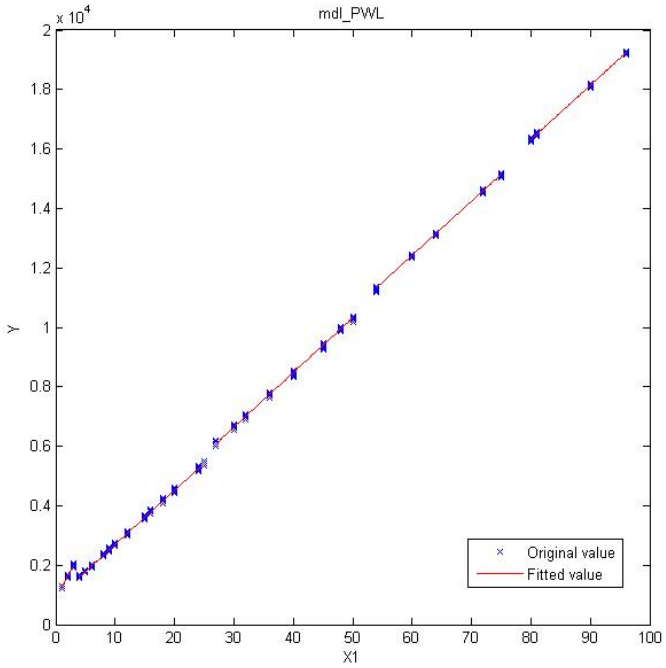

Figure 4.18: Piecewise-linear model for Task Ti.

4.5.4 Model validation

With validation data, the piecewise-linear model gives the result shown in Figure 4.19, with a $RMSE = 38.93$, which is satisfactory. So Equation 4.7 is taken as the result for Task Ti.

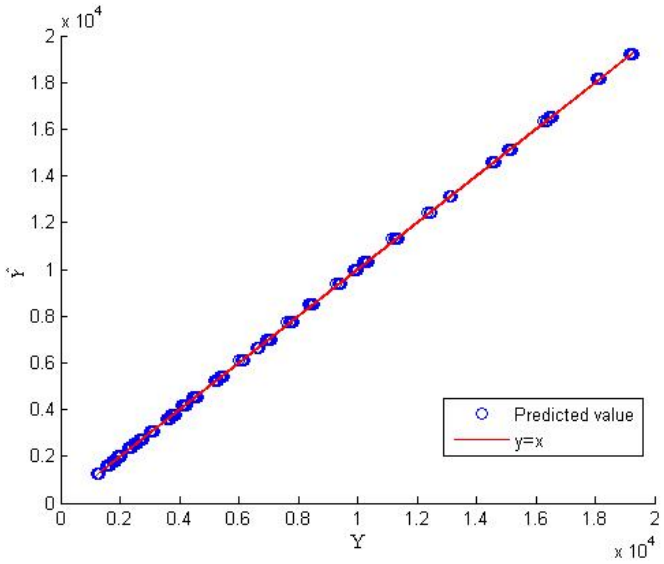

Figure 4.19: Validation of piecewise-linear model of Task Ti.

4.6 Task CE2

In this section Task CE2 is analyzed. There are 20 parameters available, of which 9 are constant and therefore ignored. The remaining parameters are denoted as variables $X_1, X_2, \cdots, X_{11}$, and the measured running time is denoted as variable Y. There are 30046 samples in total. For Y, the smallest is 409 while the largest is 5447.

4.6.1 Setting criterion

The samples with same parameters give an estimate as $s = 18.04$. The criterion for acceptable models is set as $RMSE < 2s = 36.08$.

4.6.2 Scatter plot and correlation table

The scatter plot is given in Figure 4.20. Again a linear relationship seems to exist between Y and X_1. The partial correlation table is given in Figure 4.21.

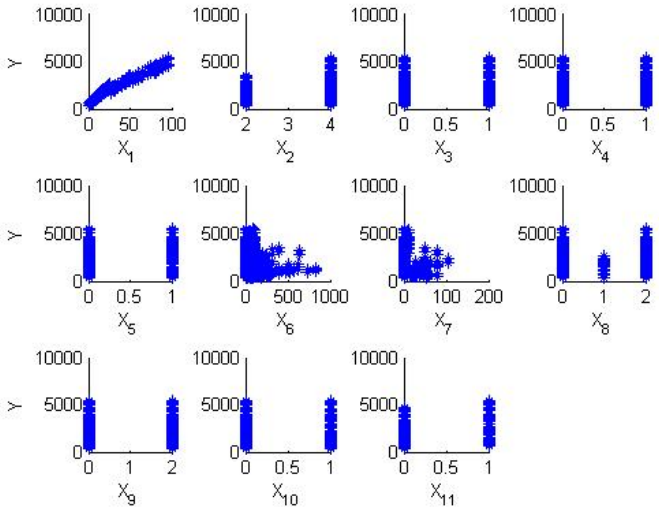

Figure 4.20: Scatter plot of the data set for Task CE2.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
r_{iY}	0.9856	0.0130	-0.0210	0.0589	0	0.1363	0.0154	-0.0586	0	-0.0181	0.6718

Figure 4.21: Correlation of the data set for Task CE2.

4.6.3 Models trial

With linear model approach, stepwise linear regression does not give an adequate model with either first-order or second-order models. The minimum RMSE it reaches is 100.94 with a second-order linear model. A single-variable piecewise linear model does not work well too. Then tree-based models are tried. One-node tree model is implemented here.

First the node variable is decided. According to the partial correlation table in Figure 4.21, X_1 and X_{11} are the two variables that have the largest partial coefficients. X_1 ranges from 1 to 96 while X_{11} is a binary variable, taking either 0 or 1. Therefore X_{11} is taken as the node and data is separated according to its value. For each small data set, a model is built with methods

mentioned above, be it linear or piecewise-linear regression. In this way, a tree model with $RMSE = 27.55$ is built, which meets the criterion well. The result is shown in Figure 4.22.

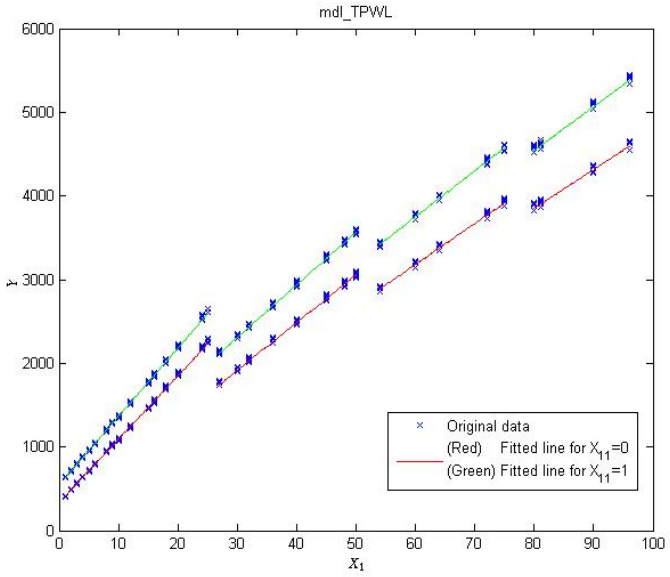

Figure 4.22: Tree-based model of Task CE2.

The model is in the form:

$$\hat{Y} = \begin{cases} f_1(x), \text{if } X_{11} = 0; \\ f_2(x), \text{if } X_{11} = 1. \end{cases} \tag{4.8}$$

where

$$f_1(x) = \begin{cases} 336.05 + 75.74X_1; \text{ if } 1 \leq X_1 < 27 \\ 210.01 + 56.89X_1; \text{ if } 27 \leq X_1 < 54 \\ 240.19 + 49.02X_1; \text{ if } 54 \leq X_1 < 80 \\ 143.16 + 46.36X_1; \text{ if } 80 \leq X_1 < 96 \end{cases} \tag{4.9}$$

and

$$f_2(x) = \begin{cases} 551.57 + 81.56X_1; \text{ if } 1 \leq X_1 < 27 \\ 422.24 + 62.89X_1; \text{ if } 27 \leq X_1 < 54 \\ 445.19 + 55.09X_1; \text{ if } 54 \leq X_1 < 80 \\ 363.72 + 52.25X_1; \text{ if } 80 \leq X_1 < 96 \end{cases} \tag{4.10}$$

4.6.4 Model validation

With validation data, the tree-based model gives the result shown in Figure 4.23, with a $RMSE = 19.33$, which is satisfactory. So Equation 4.8 is taken as the result for Task CE2.

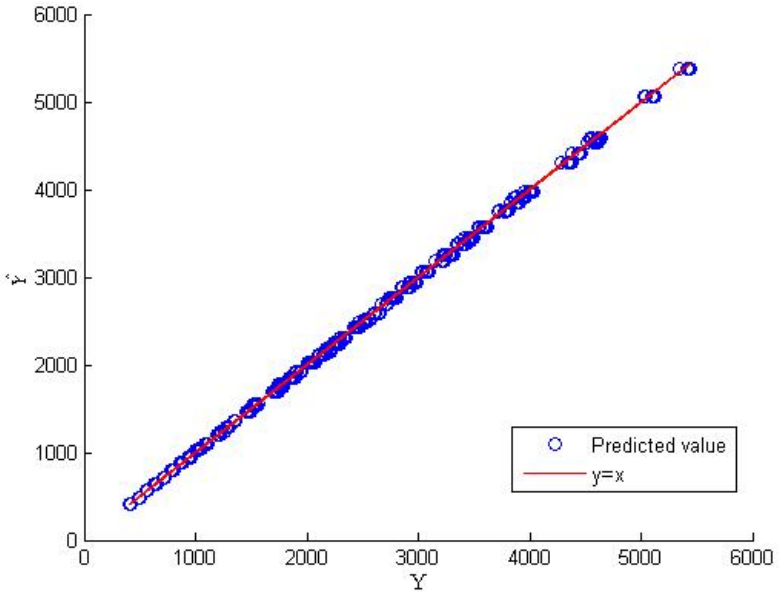

Figure 4.23: Validation of tree-based model of Task CE2.

4.7 Task CE5

In this section Task CE5 is analyzed. There are 20 parameters available, of which 9 are constant and therefore ignored. The remaining parameters are denoted as variables $X_1, X_2, \cdots, X_{11}$, and the measured running time is denoted as variable Y. There are 19807 samples in total. For Y, the smallest is 2056 while the largest is 44541.

4.7.1 Setting criterion

For this task, the samples with same parameters give an estimate as $s = 1799.96$. The criterion for acceptable models is set as $RMSE < 2s = 3599.90$.

4.7.2 Scatter plot and correlation table

The scatter plot indicates that X_1 is an important factor to Y. The partial correlation table is shown as Figure 4.25.

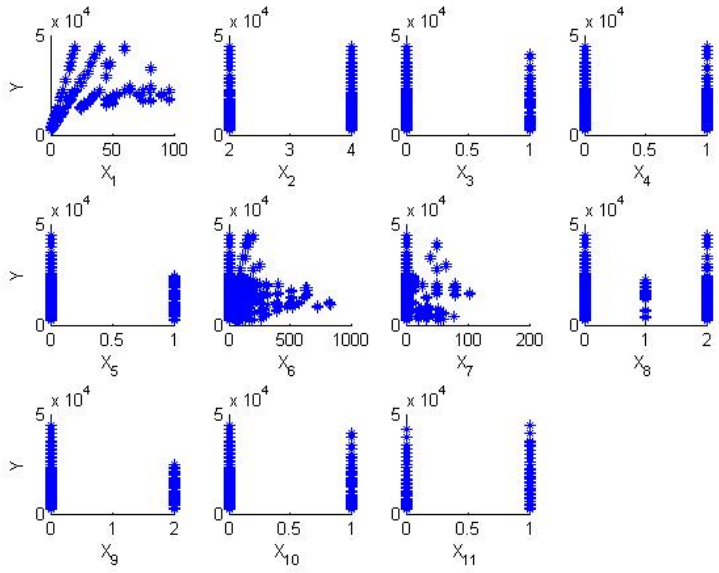

Figure 4.24: Scatter plot of the data set for Task CE5.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
r_{iY}	0.7617	0.0377	-0.0216	0.0731	0	0.1301	-0.0021	-0.0702	0	-0.0155	0.0698

Figure 4.25: Correlation of the data set for Task CE5.

4.7.3 Models trial

A second-order linear model is accepted according to our criterion, with a RMSE=2737.02. The result is shown in Figure 4.26.

With this plot, it is hard to say that the model is adequate model the task. Other methods are tried but none gives better models. Therefore for this task, no adequate model can be built.

To explain the reason behind this, the samples having the same parameters except X_1 is drawn as a plot of Y against X_1, shown as Figure 4.27.

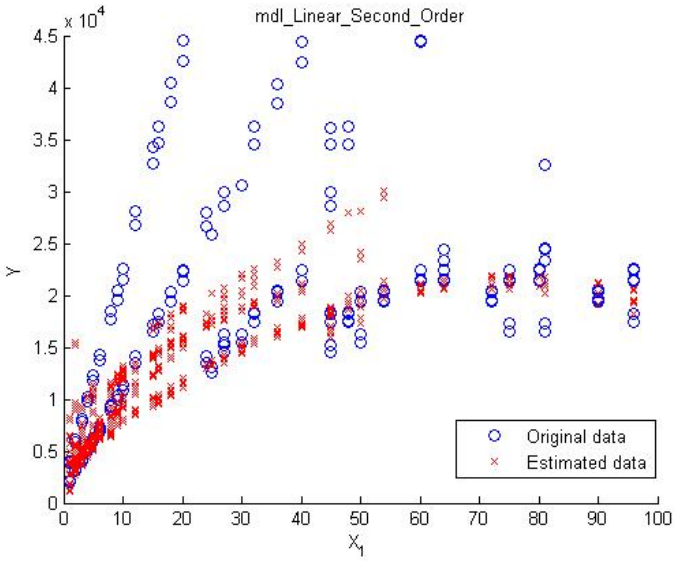

Figure 4.26: Second-order linear model for Task CE5.

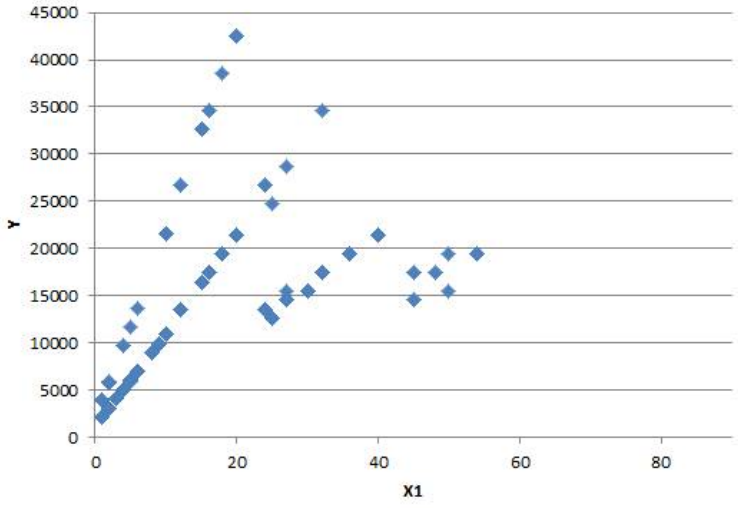

Figure 4.27: Data samples with varying X_1 for Task CE5.

The plots shows that, at almost all X_1 values, there are two or more different values, while all other parameters are exactly the same. This indicates a high possibility that some parameter which controls this variation is missed when collecting the data.

To detect this situation, review the estimated error variance $s = 1799.96$. Also the minimum value of available Y is 2056. $s/min(Y) = 87.55\%$. This means the noise variance is 87.55% of the real value. Compared to previous tasks data, the noise is too large. This ratio can be utilized to decide the quality of the data set before building models.

4.8 Summary

In this chapter, given the predefined criterion, several tasks are analyzed and proper models are built. The models vary in types and complexity, some being first-order or second-order linear, others being piecewise-linear or tree-based models. With validation data, the models' usefulness is validated. For some tasks like Task CE5, no adequate model is built with current approach.

This chapter analyzes the data and builds models in a step-by-step way. Graphical skills like scatter plot are used, and sometimes decision are made by human intuition (for instance, when choosing a "best" model with all-possible-regression method). With a lot of tasks, this approach is not efficient. An automatic tool is well worth considering.

Chapter 5

Design of the automatic tool

This chapter considers to automate the task modeling procedure, i.e., to build an automatic tool for task modeling. The tool is expected to involve less human intervention during the process, and be able to deal with large data set efficiently, therefore being able to handle more tasks in the future. The tool is implemented with Matlab object-oriented programming language[15].

5.1 The flow chart

The work flow of the automatic tool comes from the study of typical tasks shown in Chapter 4. As seen in previous chapter, several basic types of functions are able to model most given tasks, and procedures like stepwise linear regression can be utilized to avoid human intervention with proper predefined criteria.

The tool starts with pre-analysis of the data. Here the tool discards constant parameters, calculates correlation and partial correlation tables, and estimates the noise variance s from samples with same parameters for later use.

Then the criterion for an acceptable model is set as $RMSE < 2s$.

Next, different model types are tried. First-order linear models are tried first with stepwise linear regression approach, where the α_E and α_R are predefined. The default value is set as $\alpha_E = 0.001$ and $\alpha_R = 0.05$ based on our experience. If there is no model meeting the criterion, second-order linear models are tried, where interactive items and squared items are included.

If these linear models do not meet the criterion, piecewise-linear models are tried. The variable based on which the data is separated is decided by partial correlation and the variable's range. The sliding-window approach is implemented to give the piecewise-linear model. Here the initial length of a piece is set to 2, which means a first piece is built with the data points having the minimum two parameter values.

Finally tree-based models are tried, and the node is again decided by partial correlation and the parameter ranges. Only a one-layer tree is considered for the moment. At each leaf, a linear model or piecewise-linear model is given.

During these trials, if there is some model meeting the criterion, it is taken as the model and outputted; if none of these model types gives an acceptable model, the tool reports a failure.

The flow chart is given in Figure 5.1.

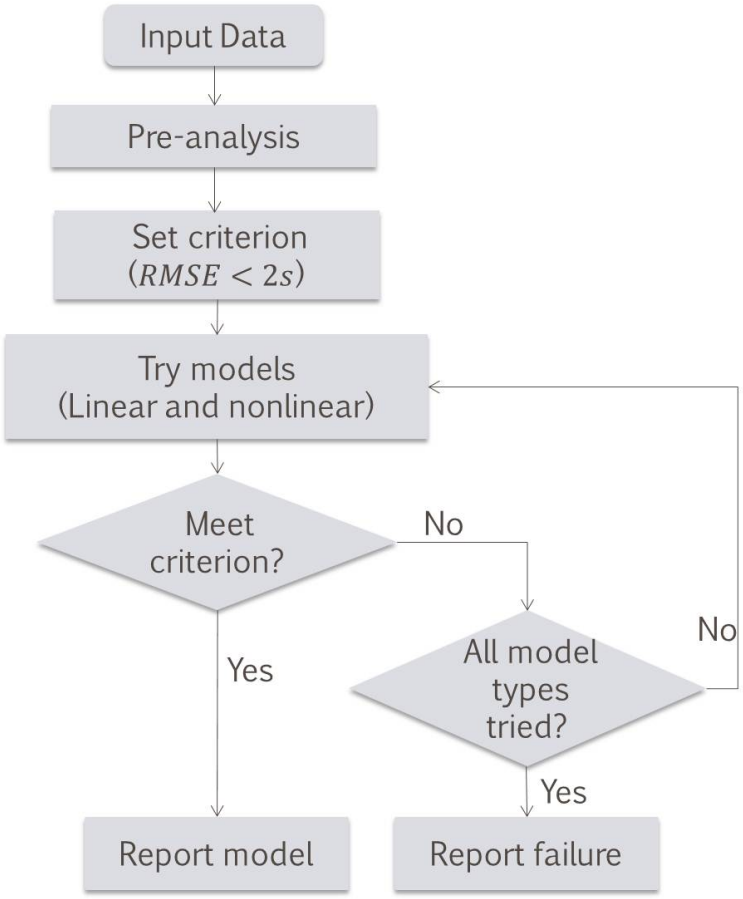

Figure 5.1: Flow chart of the automatic tool.

5.2 Summary

The tool is advantageous for its automatic procedure and thus avoids human intervention. The criterion is simple and easy to be compared by different types of models. Linear models, piecewise-linear models and tree-based models form a model pool. The tasks try different models to select a simple and acceptable one.

The disadvantage is obvious too. There might be cases where a scatter plot can give much intuitions while the tool cannot take advantage of it. The models have to be tried in some order(here linear models are tried first then nonlinear models), while this might not be the optimal order. The model pool has only a limited number of available model prototypes, which limits its flexibility and extensibility.

In the tool design, the validation procedure is not included. The validation procedure often involves people's judgement on plots and requires new data; and from several tasks analyzed in Chapter 4, the validation procedure does not conflict with our criterion with our model prototypes. The validation procedure is suggested to be manually carried out with new data, if necessary.

Chapter 6

Results

In this chapter each task is analyzed and a proper model is built. There are 17 tasks in total. With the help of the tool, a model is built for each task, with its RMSE as the performance measure. In addition, the time consumption for generating each task model is stated as a measure on how the tool works.

6.1 Linear models

Task CE1

For this task, the data set leads to an $s = 16.0884$. The tool gives a second-order linear model with 22 items in 8 parameters, and an $RMSE = 28.3$. The time consumption is 59.7372s with 30091 samples in total.

$$\begin{aligned}\hat{Y} &= 535.37 + 114.19X_1 - 3.0826X_2 + 12.125X_3 - 73.154X_4 \\ &\quad + 150.96X_5 + 0.6239X_6 + 0.0225X_7 - 46.967X_8 \\ &\quad + 0.852X_1X_2 - 0.2054X_1X_3 + 2.4119X_1X_4 - 2.685X_1X_5 \\ &\quad + 0.0024X_1X_6 + 0.0216X_1X_7 - 32.932X_2X_5 - 0.326X_2X_6 \\ &\quad + 26.861X_2X_8 - 103.32X_3X_4 + 0.4858X_3X_6 \\ &\quad - 0.0054X_6X_7 + 0.4767X_7X_8 + 0.046X_1^2 - 0.0002X_6^2 \end{aligned} \tag{6.1}$$

The too many items makes it less interesting. Meanwhile, a first-order linear model gives an $RMSE = 46.7$ with only 4 items.

$$\hat{Y} = 493.72 + 121.48X_1 - 28.518X_3 - 0.2133X_6 + 31.308X_9 \tag{6.2}$$

Figure 6.1 shows the first-order modeling.

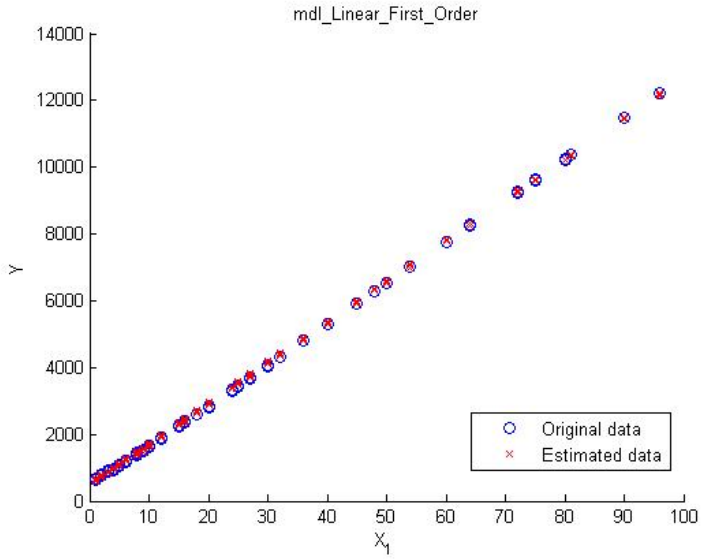

Figure 6.1: Modeling of Task CE1.

Equation6.2 is taken as the final model for Task CE1.

Task Co

For this task, the data set leads to an $s = 27.5384$. The tool gives a second-order linear model with 3 items in 2 parameters, and an $RMSE = 31.9$. The time consumption is 4.9964s with 363 samples in total.

$$\hat{Y} = 342.99 + 5.1485X_{10} + 167.12X_{11} + 7.5498X_{11}^2 \tag{6.3}$$

Figure 6.2 shows the modeling.

Task De

For this task, the data set leads to an $s = 78.2091$. The tool gives a second-order linear model with 20 items in 8 parameters, and an $RMSE = 97.2$. The time consumption is 17.3175s with 7484 samples in total.

$$\begin{aligned}\hat{Y} &= 2585.4 - 1.5782X_1 - 3.2307X_2 - 26.123X_3 + 1528.5X_4 \\ &+ 2389.6X_5 + 24.233X_6 + 16.156X_7 - 894.97X_8 \\ &+ 17.832X_1X_2 + 1.654X_1X_3 - 0.0134X_1X_6 \\ &+ 2.6275X_1X_8 - 594.92X_2X_4 - 1218.1X_2X_5 + 10.26X_2X_6 \\ &+ 4.9397X_2X_7 + 369.16X_2X_8 - 22.34X_6X_8 + 0.0342X_1^2\end{aligned} \tag{6.4}$$

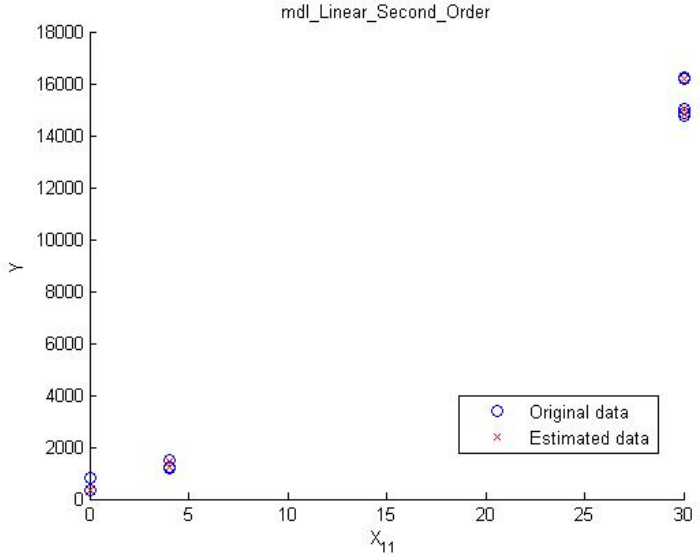

Figure 6.2: Modeling of Task Co.

Figure 6.3 shows the modeling.

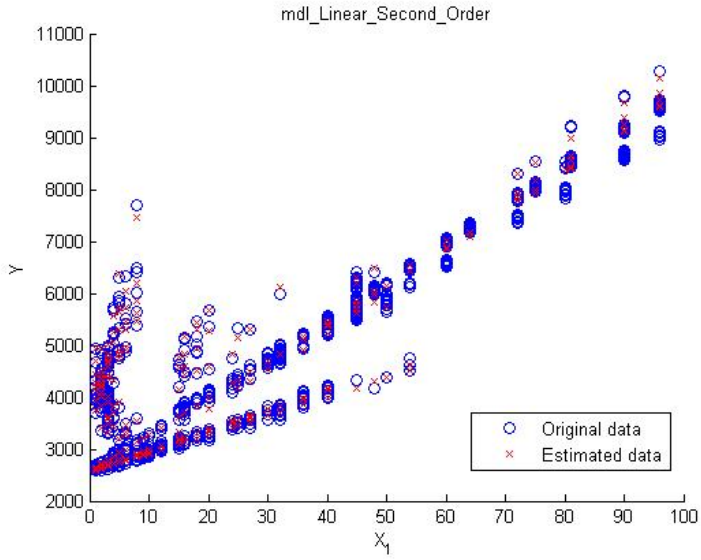

Figure 6.3: Modeling of Task De.

Task Ri

For this task, the data set leads to an $s = 30.2314$. The tool gives a first-order linear model with 7 parameters, and an $RMSE = 46.6$. The time consumption is 6.8317s with 1212 samples in total.

$$\begin{aligned}\hat{Y} = {} & 492.11 - 0.2973X_1 - 19.448X_2 + 1176.9X_4 + 7.5157X_6 \\ & + 12.208X_7 - 18.904X_9 + 1696.5X_{10}\end{aligned} \tag{6.5}$$

Figure 6.4 shows the modeling.

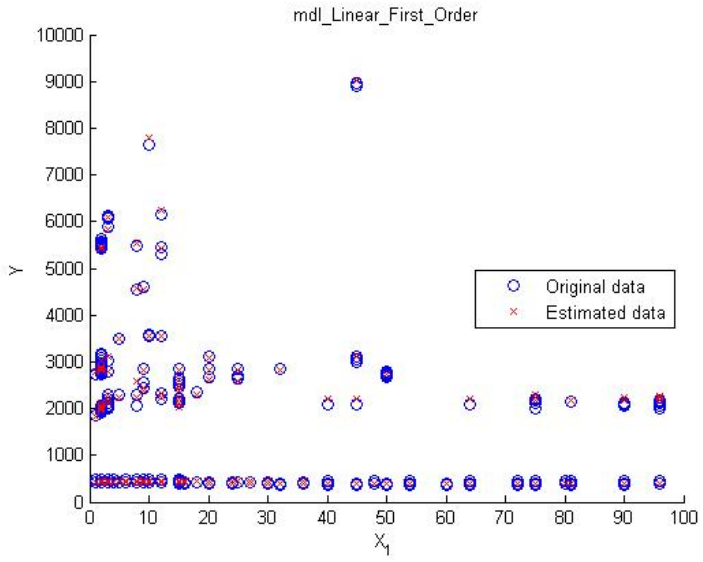

Figure 6.4: Modeling of Task Ri.

Task Rs

For this task, the data set leads to an $s = 9.6856$. The tool gives a first-order linear model with 6 parameters, and an $RMSE = 11.2723$. The time consumption is 16.4601s with 16993 samples in total.

$$\begin{aligned}\hat{Y} = {} & 249.95 + 84.135X_1 - 7.2931X_5 + 0.0265X_6 \\ & - 0.1403X_7 + 6.3504X_{10} - 1.9829X_{11}\end{aligned} \tag{6.6}$$

Figure 6.5 shows the modeling.

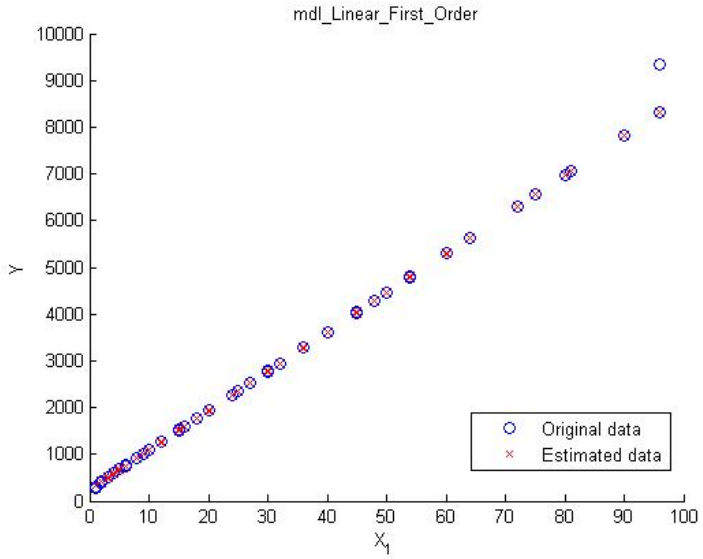

Figure 6.5: Modeling of Task Rs.

Task HD

Task HD has been analyzed in Section 4.3, giving a simple first-order linear model as shown in Equation 4.4.

Task HqD

Task HqD has been analyzed in Section 4.4, giving a second-order linear model as shown in Equation 4.5.

6.2 Piecewise-linear models

Task An

For this task, the data set leads to an $s = 46.7026$. The tool gives a piecewise-linear model with $RMSE = 68.4555$. The time consumption is about 86.7411s with 41540 samples in total.

$$\hat{Y} = \begin{cases} 518.05 + 136.11X_1; \text{ if } 1 \leq X_1 < 36 \\ 1226.5 + 134.6X_1; \text{ if } 36 \leq X_1 < 72 \\ 1824.9 + 134.94X_1; \text{ if } 72 \leq X_1 < 100 \end{cases} \tag{6.7}$$

Figure 6.6 shows the modeling.

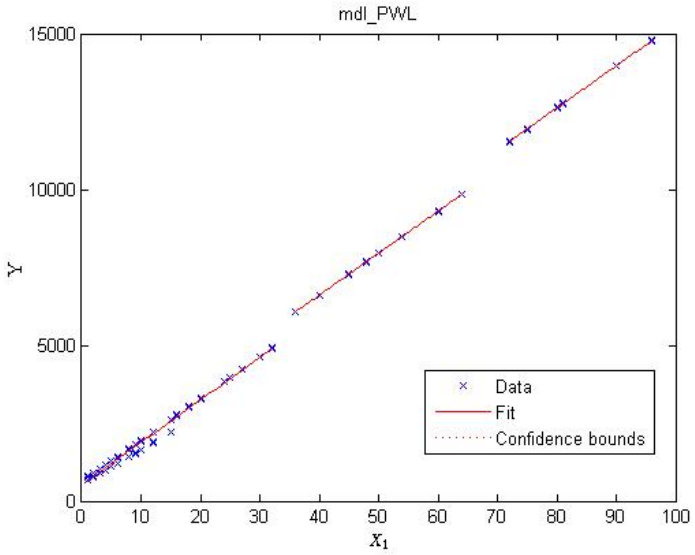

Figure 6.6: Modeling of Task An.

Task Po

For this task, the data set leads to an $s = 432.0566$. The tool gives a second-order linear model with 23 items in 8 parameters with an $RMSE = 791$. Also a piecewise-linear model can be given with an $RMSE = 458.8830$. The Piecewise-linear model is selected. The time consumption is about 48s with 26687 samples in total.

$$\hat{Y} = \begin{cases} 632.72 + 348.76X_1; \text{ if } 1 \leq X_1 < 4 \\ 659.89 + 355.31X_1; \text{ if } 4 \leq X_1 < 8 \\ 526.73 + 377.45X_1; \text{ if } 8 \leq X_1 < 12 \\ 821.36 + 379.74X_1; \text{ if } 12 \leq X_1 < 24 \\ 918.41 + 187.33X_1; \text{ if } 24 \leq X_1 < 45 \\ 966.54 + 123.99X_1; \text{ if } 45 \leq X_1 < 72 \\ 310.76 + 101.89X_1; \text{ if } 72 \leq X_1 < 81 \\ 10247 - 23.76X_1; \text{ if } 81 \leq X_1 < 100 \end{cases} \tag{6.8}$$

Figure 6.7 shows the modeling.

Task Ti

Task Ti has been analyzed in Section 4.5, giving a tree-based model as shown in Equation 4.7.

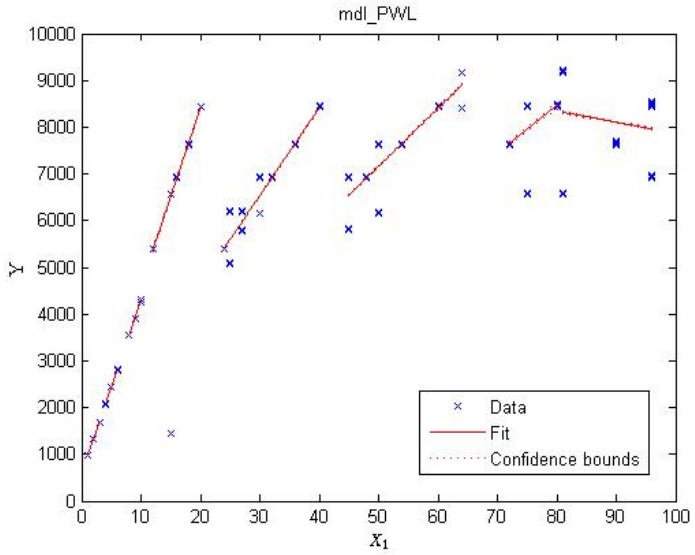

Figure 6.7: Modeling of Task Po.

6.3 Tree-based models

Task CE3

For this task, the data set leads to an $s = 163.4087$. The tool gives a tree-based model. Parameter 11 is chosen as the node and there are 2 branches. For each branch, a piecewise-linear model is given. The $RMSE = 258.8638$. The time consumption 79.2450s with 24278 samples in total.

The model is in the form of Equation 6.9:

$$\hat{Y} = \begin{cases} f_1(x), \text{if } X_{11} = 0; \\ f_2(x), \text{if } X_{11} = 1. \end{cases} \tag{6.9}$$

where

$$f_1(x) = \begin{cases} 484.9 + 108.25X_1; \text{ if } 1 \leq X_1 < 12 \\ 889.71 + 110.58X_1; \text{ if } 12 \leq X_1 < 20 \\ 6674.9 - 180.04X_1; \text{ if } 20 \leq X_1 < 27 \\ 869.23 + 56.00X_1; \text{ if } 27 \leq X_1 < 45 \\ 862.47 + 37.46X_1; \text{ if } 45 \leq X_1 < 72 \\ 811.49 + 28.81X_1; \text{ if } 72 \leq X_1 < 81 \\ 3714.8 - 7.74X_1; \text{ if } 81 \leq X_1 < 100 \end{cases}$$

and

$$f_2(x) = \begin{cases} 1083 + 162.93X_1; \text{ if } 1 \leq X_1 < 12 \\ 1627.3 + 168.10X_1; \text{ if } 12 \leq X_1 < 24 \\ 1121.6 + 91.31X_1; \text{ if } 24 \leq X_1 < 45 \\ 1276.3 + 57.70X_1; \text{ if } 45 \leq X_1 < 72 \\ 1534.5 + 40.04X_1; \text{ if } 72 \leq X_1 < 81 \\ 5230.3 - 6.86X_1; \text{ if } 81 \leq X_1 < 100 \end{cases}$$

The Figure 6.8 shows the modeling.

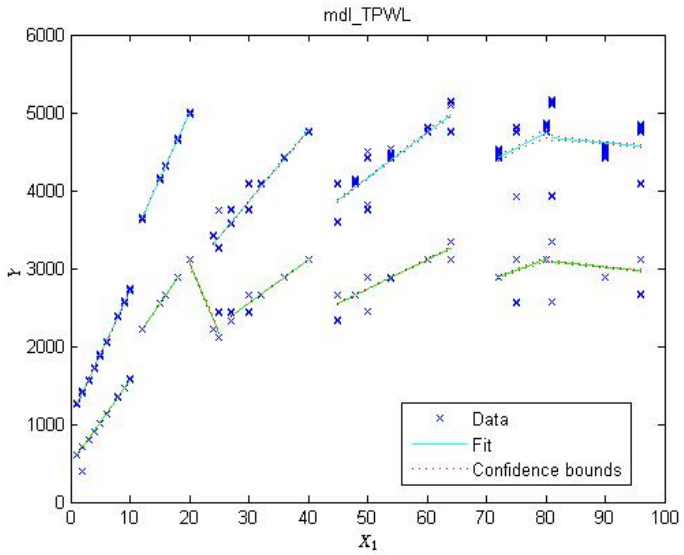

Figure 6.8: Modeling of Task CE3.

Task CE4

For this task, the data set leads to an $s = 58.8445$. The tool gives a tree-based model. Parameter 11 is chosen as the node and there are 2 branches. For each branch, a piecewise-linear model is given. The $RMSE = 103.2757$. The time consumption 55.7920s with 24079 samples in total.

The model is in the form of Equation 6.10:

$$\hat{Y} = \begin{cases} f_1(x), \text{if } X_{11} = 0; \\ f_2(x), \text{if } X_{11} = 1. \end{cases} \tag{6.10}$$

where

$$f_1(x) = \begin{cases} 767 + 222.55X_1; \text{ if } 1 \leq X_1 < 5 \\ 692.53 + 235.6X_1; \text{ if } 5 \leq X_1 < 24 \\ 715.4 - 117.71X_1; \text{ if } 24 \leq X_1 < 45 \\ 588.61 + 81.24X_1; \text{ if } 45 \leq X_1 < 64 \\ 9840.2 - 64.14X_1; \text{ if } 64 \leq X_1 < 80 \\ 9960.8 - 55.32X_1; \text{ if } 80 \leq X_1 < 100 \end{cases}$$

and

$$f_2(x) = \begin{cases} 1977.1 + 426.69X_1; \text{ if } 1 \leq X_1 < 24 \\ 899.37 + 223.93X_1; \text{ if } 24 \leq X_1 < 45 \\ 1447.1 + 138.96X_1; \text{ if } 45 \leq X_1 < 72 \\ 1792.6 + 100.01X_1; \text{ if } 72 \leq X_1 < 90 \\ 2439.9 + 73.35X_1; \text{ if } 90 \leq X_1 < 100 \end{cases}$$

Figure 6.9 shows the modeling.

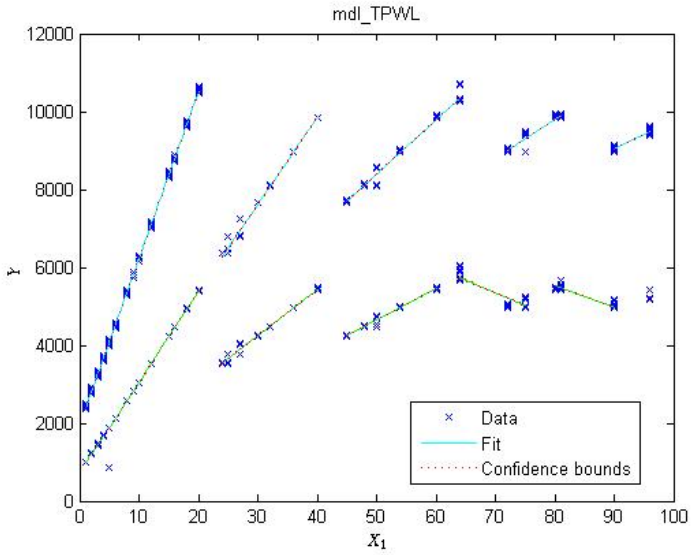

Figure 6.9: Modeling of Task CE4.

Task Dei

For this task, the data set leads to an $s = 143.5442$. The tool gives a second-order linear model with $RMSE = 221.1890$, but the plot is not satisfactory, shown in Figure 6.10.

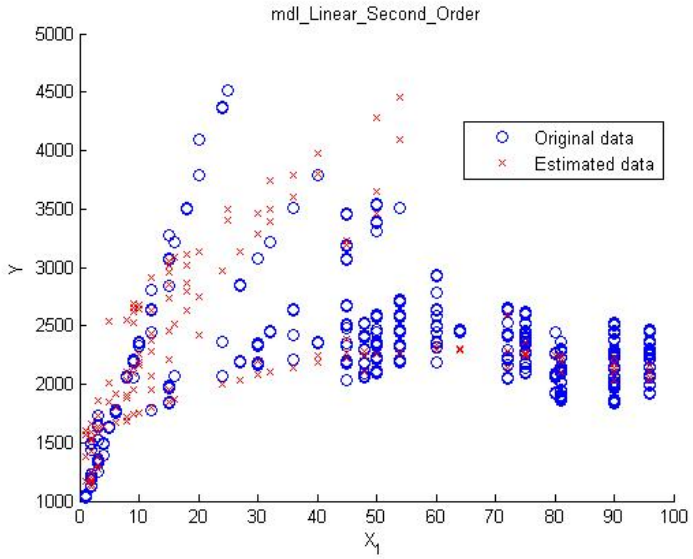

Figure 6.10: A second-order linear model of Task Dei.

The model can also give a tree-based model. If Parameter 2 is set as the node, then for each branch a piecewise-linear model is given. The $RMSE = 188.9782$.

The model is in the form of Equation 6.11:

$$\hat{Y} = \begin{cases} f_1(x), \text{if } X_2 = 2; \\ f_2(x), \text{if } X_2 = 4. \end{cases} \tag{6.11}$$

where

$$f_1(x) = \begin{cases} 1072.9 + 132.83X_1; \text{ if } 1 \leq X_1 < 27 \\ 896.08 + 72.44X_1; \text{ if } 27 \leq X_1 < 45 \\ 1852.9 + 30.87X_1; \text{ if } 45 \leq X_1 < 96 \end{cases}$$

and

$$f_2(x) = \begin{cases} 896.49 + 144.86X_1; \text{ if } 1 \leq X_1 < 12 \\ 1037.2 + 61.70X_1; \text{ if } 12 \leq X_1 < 24 \\ 1060.2 + 42.99X_1; \text{ if } 24 \leq X_1 < 36 \\ 2491 - 3.73X_1; \text{ if } 36 \leq X_1 < 96 \end{cases}$$

The Figure 6.11 shows the modeling.

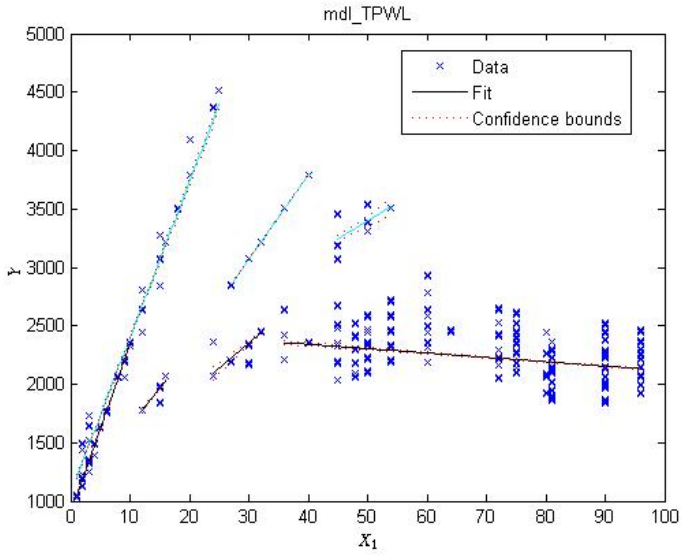

Figure 6.11: Modeling of Task Dei.

Task So

For this task, the data set leads to an $s = 24.8767$. The tool cannot give a proper model automatically. With the tree-based method, the tool takes Parameter 11 as the node. But if parameter 2 is set as the node then an acceptable model can be built. There are 2 branches. For one branch, a first-order linear model is given; for the other, a piecewise-linear model is given. The $RMSE = 40.7272$.

The model is in the form of Equation 6.12:

$$\hat{Y} = \begin{cases} f_1(x), \text{if } X_2 = 2; \\ f_2(x), \text{if } X_2 = 4. \end{cases} \tag{6.12}$$

where

$$f_1(x) = 368.98 + 82.38X_1$$

and

$$f_2(x) = \begin{cases} 377.91 + 104.89X_1; \text{ if } 1 \leq X_1 < 80 \\ 438.23 + 52.04X_1; \text{ if } 80 \leq X_1 < 96 \end{cases}$$

The Figure 6.12 shows the modeling.

There are two possible reasons behind the problem. One could be taking partial linear regression coefficients as our criterion to decide which parameter can be the node, which may not be well adequate. The other is

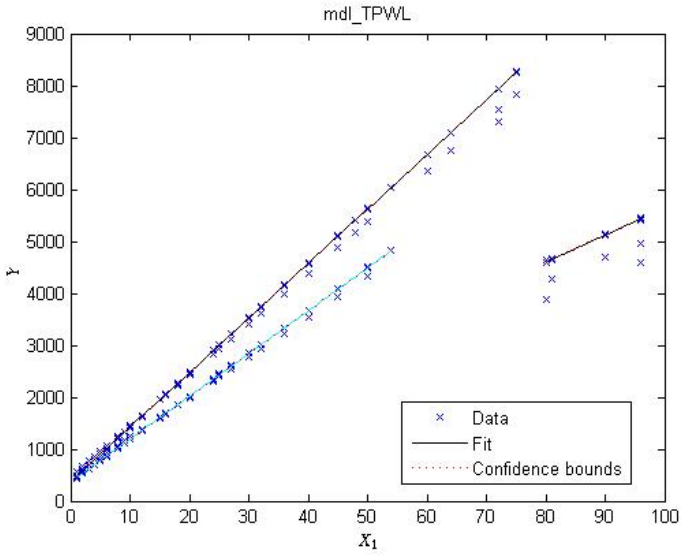

Figure 6.12: Modeling of Task So.

that only one-layer tree structure is considered in the design. If multi-layer tree can be designed, then the problem may be solved. Actually, Parameter 2 has a smaller partial coefficient than Parameter 1 and Parameter 11, but bigger than all others.

Task CE2

Task CE2 has been analyzed in Section 4.6, giving a tree-based model as shown in Equation 4.8.

6.4 Unexplained ones

Task Si

For this task, the data set leads to an $s = 181.1121$. The tool gives a first-order linear model, with $RMSE = 190.2107$.

$$\hat{Y} = 2226 + 175.51X_1 \tag{6.13}$$

Figure 6.13 shows the modeling.

The plots shows that the upper data points are all missed. So more study should be conducted on this task.

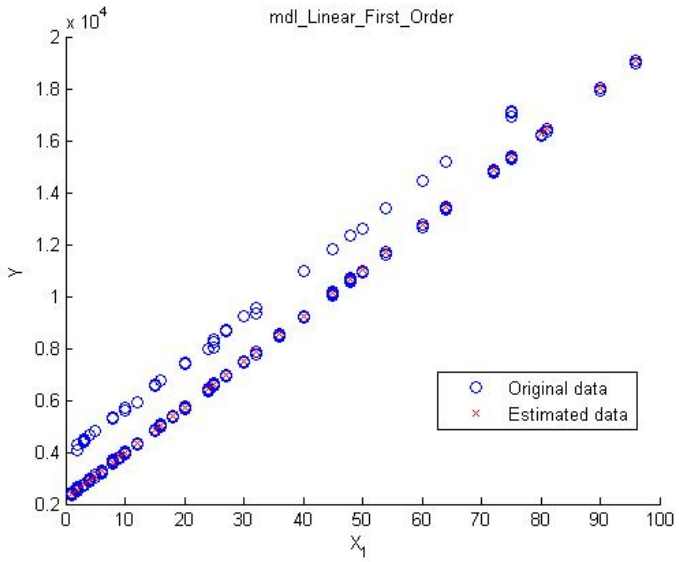

Figure 6.13: Modeling of Task Si.

Task CE5

Task CE5 has been analyzed in Chapter 4, with no satisfactory output models. The possible reason is also explained in Section 4.7.

6.5 Summary

The tool builds models for 14 tasks automatically. For Task CE1 and Task So, auxiliary steps are needed to build an acceptable model. This shows that the automatic tool works fine with many tasks but is still not fully capable of dealing with highly complex tasks.

The main limit of the tool results from the limited model prototypes, the order of the trial of the models, as stated in Section 5.2. Also taking partial correlation as a criterion for variables selecting is not well proved but only based on experiments. This does not always work fine, for example in Task So, automatically it takes Parameter 11 rather than Parameter 2 as the node, which does not lead to a good result.

Chapter 7

Summary and future work

This chapter summarizes the thesis work, discusses some problems during the work, and gives some suggestions on future work.

7.1 Summary of the thesis

During this thesis work, three basic function prototypes are proposed for modeling the relationship between the task load and its parameters, i.e., linear function, piecewise-linear function and tree-based function. These models varies in the number of parameters involved and complexity of the outcome model.

For linear models, stepwise linear regression is used as the main approach. The threshold to include or exclude parameters, α_E and α_E, are set by experience. The values have an influence on the result's complexity. The specific value of the parameters may need more trial for different tasks.

For piecewise-linear models, the parameter according to which the data is separated and how to decide the breakpoints are the key problems. The parameter is selected based on partial correlation and this works fine with most tasks. The sliding-window approach is used to separate the data and build the model. With this approach, the number of pieces is not specified in advance. The initial length of the first piece is set to 2 based on experiments, which may need adjustment in some cases.

For tree-based models, only one-layer tree is considered. The node is decided by partial correlation coefficient and the variable range.

There models can be considered as one unified modeling approach: a tree-based model with leaves of piece-wise linear modeling approach. Special cases could be zero node and linear models at leaves.

The criterion for an acceptable model is defined based on the intrinsic property of the data set, that is, samples with same parameters. The criterion $2s$ is also an empirical value, which may be flexibly adjusted if necessary.

The models built for the tasks are capable of predicting future task load with new parameters, and also can give some intuition on how the true relationship may be between the task load and its parameters.

The guided automatic tool means to help build models for new tasks faster. The tool works fine with many tasks, while for some tasks re-examination on the output result may be needed and the final model should be decided carefully.

7.2 Future work

On noise

With explicit modeling approach, an assumption of the noise is unavoidable. In this thesis it is assumed that the noise is normally distributed, independent with parameters or task load, i.e., $N(0, I\sigma^2)$. The assumption is mainly based on central limit theory and it turns out that the assumption does not harm the modeling. A more careful study may be carried out on the noise if necessary. For example, in some cases if the noise variance varies with parameters, another model like $N(0, V\sigma^2)$ may be applied and weighted least square method can then be used instead of ordinary least square method.

On model criterion

The criterion for acceptable models is calculated based on estimation of noise variance from samples with same parameters. The calculation here relies heavily on the quality of data set. If the number of repeat runs is very limited, the estimate of the noise variance may be unreliable. For later research, improvement can be carried out on this topic.

On correlation among input variables

In linear regression, the ideal situation is that the variables in the regressor are all independent variables, i.e., $X_1, X_2, ...$ are all independent variables. This is hardly true in practice, and in this thesis partial correlation and stepwise approach are used to avoid this problem. Other approaches are

available like ridge regression[6]. If the correlation among parameters are serious, these approaches can be considered.

On model forms

Three basic different function forms are designed in the thesis to fit different tasks. What has not been considered further in the thesis is a unified form of model functions. Based on our current work, there is a high possibility that interactive items is necessary for many tasks. The second-linear models and the tree-based models prove this. A new model prototype shown as Equation 7.1 is well worth considering.

$$\hat{Y} = f_1(X_1) \cdot f_2(X_2) \cdot \ \cdots \ \cdot f_m(X_m) \tag{7.1}$$

In this prototype, $f_1, f_2, ...$ are basic functions, mostly in the form of a simple first-order linear function form $y = \alpha x + \beta$, yet in some tasks, a piecewise-linear function is appropriate. In this way, the models shown in this thesis are included and can be extended to more complex models.

On researcher's knowledge

The thesis tries to avoid taking advantage of researcher's pre-knowledge on parameters when building the models. The analysis goes in a pure mathematical way. In this way the same work procedure applies to any future task. But for explicit modeling approach, researcher's knowledge can be of great help and improve the model's usefulness significantly. For instance, when building a piecewise linear model, if the breakpoints can be decided in advance, the model building process can be shorted and the result can gain more explanatory power. Careful use of researcher's knowledge is suggested with explicit modeling approach.

Bibliography

[1] 3GPP. About 3GPP Home. `http://www.3gpp.org/about-3gpp/about-3gpp`. Online: acessed 1-July-2014.

[2] 3GPP. Overview of 3GPP Release 8. `http://www.3gpp.org/specifications/releases/72-release-8`. Online: acessed 1-July-2014.

[3] George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.

[4] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[5] J.M. Chambers. *Graphical methods for data analysis*. Chapman & Hall statistics series. Wadsworth International Group, 1983.

[6] S. Chatterjee and A.S. Hadi. *Regression Analysis by Example*. Wiley Series in Probability and Statistics. Wiley, 2006.

[7] L.O. Chua and Sung Mo Kang. Section-wise piecewise-linear functions: Canonical representation, properties, and applications. *Proceedings of the IEEE*, 65(6):915–929, June 1977.

[8] Peter A Dinda and David R O'Hallaron. Host load prediction using linear models. *Cluster Computing*, 3(4):265–280, 2000.

[9] N.R. Draper and H. Smith. *Applied Regression Analysis*. Wiley Series in Probability and Statistics. Wiley, 1998.

[10] S. Kang and L.O. Chua. A global representation of multidimensional piecewise-linear functions with linear partitions. *Circuits and Systems, IEEE Transactions on*, 25(11):938–940, Nov 1978.

[11] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296, 2001.

[12] E.L. Lehmann and J.P. Romano. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer, 2006.

[13] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[14] MathWorks. Linear regression — multiple, stepwise, multivariate regression models, and more, 2014. [Online; accessed 11-July-2014].

[15] MathWorks. Object-oriented programming— advanced software development, 2014. [Online; accessed 14-July-2014].

[16] John R Quinlan et al. Learning with continuous classes. In *Proceedings of the 5th Australian joint Conference on Artificial Intelligence*, volume 92, pages 343–348. Singapore, 1992.

[17] Sena Seneviratne and David C Levy. Task profiling model for load profile prediction. *Future Generation Computer Systems*, 27(3):245–255, 2011.

[18] Wikipedia. Correlation and dependence — wikipedia, the free encyclopedia, 2014. [Online; accessed 8-July-2014].

[19] Wikipedia. Scatter plot — wikipedia, the free encyclopedia, 2014. [Online; accessed 1-August-2014].

[20] Derek Young. Stepwise regression. `https://onlinecourses.science.psu.edu/stat501/node/88`. Online: acessed 8-July-2014.

[21] Yuanyuan Zhang, Wei Sun, and Yasushi Inoguchi. Predict task running time in grid environments based on cpu load predictions. *Future Generation Computer Systems*, 24(6):489 – 497, 2008.

Appendix A

Here we list all the models with corresponding measures. The Time column is the time consumed of the automatic tool when building the model, in seconds. NA means the model is obtained by aid from human judgement thus no direct tool running time is available.

Task	Model	RMSE	s	Time(s)
CE1	$\hat{Y} = 493.72 + 121.48X_1 - 28.518X_3 - 0.2133X_6 + 31.308X_9$	46.7	16.0884	NA
Co	$\hat{Y} = 342.99 + 5.1485X_{10} + 167.12X_{11} + 7.5498X_{11}^2$	31.9	27.5384	4.9964
De	$\hat{Y} = 2585.4 - 1.5782X_1 - 3.2307X_2 - 26.123X_3 + 1528.5X_4$ $+ 2389.6X_5 + 24.233X_6 + 16.156X_7 - 894.97X_8$ $+ 17.832X_1X_2 + 1.654X_1X_3 - 0.0134X_1X_6$ $+ 2.6275X_1X_8 - 594.92X_2X_4 - 1218.1X_2X_5 + 10.26X_2X_6$ $+ 4.9397X_2X_7 + 369.16X_2X_8 - 22.34X_6X_8 + 0.0342X_1^2$	97.2	78.2091	17.3175
Ri	$\hat{Y} = 492.11 - 0.2973X_1 - 19.448X_2 + 1176.9X_4$ $+ 7.5157X_6 + 12.208X_7 - 18.904X_9 + 1696.5X_{10}$	46.6	30.2314	6.8317
Rs	$\hat{Y} = 249.95 + 84.135X_1 - 7.2931X_5 + 0.0265X_6$ $- 0.1403X_7 + 6.3504X_{10} - 1.9829X_{11}$	11.2723	9.6856	16.4601
HD	$\hat{Y} = 1701.7 + 9.5845X_4$	43.1661	41.29	NA
HqD	$\hat{Y} = 1508.5 + 176.71X_2 + 25.301X_4 - 556.84X_6 - 13.976X_2X_4 + 26.514X_4X_6$	43.1661	41.29	NA

Table 7.1: Tasks models: linear models

Task	Model	RMSE	s	Time(s)
An	$\hat{Y} = \begin{cases} 518.05 + 136.11X_1; \text{ if } 1 \leq X_1 < 36 \\ 1226.5 + 134.6X_1; \text{ if } 36 \leq X_1 < 72 \\ 1824.9 + 134.94X_1; \text{ if } 72 \leq X_1 < 100 \end{cases}$	68.4555	46.7026	86.7411
Po	$\hat{Y} = \begin{cases} 632.72 + 348.76X_1; \text{ if } 1 \leq X_1 < 4 \\ 659.89 + 355.31X_1; \text{ if } 4 \leq X_1 < 8 \\ 526.73 + 377.45X_1; \text{ if } 8 \leq X_1 < 12 \\ 821.36 + 379.74X_1; \text{ if } 12 \leq X_1 < 24 \\ 918.41 + 187.33X_1; \text{ if } 24 \leq X_1 < 45 \\ 966.54 + 123.99X_1; \text{ if } 45 \leq X_1 < 72 \\ 310.76 + 101.89X_1; \text{ if } 72 \leq X_1 < 81 \\ 10247 - 23.76X_1; \text{ if } 81 \leq X_1 < 100 \end{cases}$	458.8830	432.0566	NA
Ti	$\hat{Y} = \begin{cases} 907.3 + 363.03X_1; \text{ if } 1 \leq X_1 < 4 \\ 897.34 + 180.99X_1; \text{ if } 4 \leq X_1 < 27 \\ 1129.8 + 183.78X_1; \text{ if } 27 \leq X_1 < 54 \\ 1536.3 + 181.37X_1; \text{ if } 54 \leq X_1 < 80 \\ 1721.6 + 182.45X_1; \text{ if } 80 \leq X_1 < 96 \end{cases}$	34.39	38.39	NA

Table 7.2: Tasks models: piecewise-linear models

Task	Model	RMSE	s	Time(s)
CE3	$\hat{Y} = \begin{cases} f_1(x), \text{if } X_{11} = 0; \\ f_2(x), \text{if } X_{11} = 1. \end{cases}$ where $f_1(x) = \begin{cases} 484.9 + 108.25X_1; \text{ if } 1 \leq X_1 < 12 \\ 889.71 + 110.58X_1; \text{ if } 12 \leq X_1 < 20 \\ 6674.9 - 180.04X_1; \text{ if } 20 \leq X_1 < 27 \\ 869.23 + 56.00X_1; \text{ if } 27 \leq X_1 < 45 \\ 862.47 + 37.46X_1; \text{ if } 45 \leq X_1 < 72 \\ 811.49 + 28.81X_1; \text{ if } 72 \leq X_1 < 81 \\ 3714.8 - 7.74X_1; \text{ if } 81 \leq X_1 < 100 \end{cases}$ and $f_2(x) = \begin{cases} 1083 + 162.93X_1; \text{ if } 1 \leq X_1 < 12 \\ 1627.3 + 168.10X_1; \text{ if } 12 \leq X_1 < 24 \\ 1121.6 + 91.31X_1; \text{ if } 24 \leq X_1 < 45 \\ 1276.3 + 57.70X_1; \text{ if } 45 \leq X_1 < 72 \\ 1534.5 + 40.04X_1; \text{ if } 72 \leq X_1 < 81 \\ 5230.3 - 6.86X_1; \text{ if } 81 \leq X_1 < 100 \end{cases}$	258.8638	163.4087	79.2450

Table 7.3: Tasks models: tree-based models(1)

Task	Model	RMSE	s	Time(s)
CE4	$\hat{Y} = \begin{cases} f_1(x), \text{if } X_{11} = 0; \\ f_2(x), \text{if } X_{11} = 1. \end{cases}$ where $f_1(x) = \begin{cases} 767 + 222.55X_1; \text{ if } 1 \leq X_1 < 5 \\ 692.53 + 235.6X_1; \text{ if } 5 \leq X_1 < 24 \\ 715.4 - 117.71X_1; \text{ if } 24 \leq X_1 < 45 \\ 588.61 + 81.24X_1; \text{ if } 45 \leq X_1 < 64 \\ 9840.2 - 64.14X_1; \text{ if } 64 \leq X_1 < 80 \\ 9960.8 - 55.32X_1; \text{ if } 80 \leq X_1 < 100 \end{cases}$ and $f_2(x) = \begin{cases} 1977.1 + 426.69X_1; \text{ if } 1 \leq X_1 < 24 \\ 899.37 + 223.93X_1; \text{ if } 24 \leq X_1 < 45 \\ 1447.1 + 138.96X_1; \text{ if } 45 \leq X_1 < 72 \\ 1792.6 + 100.01X_1; \text{ if } 72 \leq X_1 < 90 \\ 2439.9 + 73.35X_1; \text{ if } 90 \leq X_1 < 100 \end{cases}$	103.2757	58.8445	55.7920

Table 7.4: Tasks models: tree-based models(2)

Task	Model	RMSE	s	Time(s)
Dei	$\hat{Y} = \begin{cases} f_1(x), \text{if } X_2 = 2; \\ f_2(x), \text{if } X_2 = 4. \end{cases}$ where $f_1(x) = \begin{cases} 1072.9 + 132.83X_1; \text{ if } 1 \leq X_1 < 27 \\ 896.08 + 72.44X_1; \text{ if } 27 \leq X_1 < 45 \\ 1852.9 + 30.87X_1; \text{ if } 45 \leq X_1 < 96 \end{cases}$ and $f_2(x) = \begin{cases} 896.49 + 144.86X_1; \text{ if } 1 \leq X_1 < 12 \\ 1037.2 + 61.70X_1; \text{ if } 12 \leq X_1 < 24 \\ 1060.2 + 42.99X_1; \text{ if } 24 \leq X_1 < 36 \\ 2491 - 3.73X_1; \text{ if } 36 \leq X_1 < 96 \end{cases}$	188.9782	143.5442	NA

Table 7.5: Tasks models: tree-based models(3)

Task	Model	RMSE	s	Time(s)
So	$\hat{Y} = \begin{cases} f_1(x), \text{if } X_2 = 2; \\ f_2(x), \text{if } X_2 = 4. \end{cases}$ where $f_1(x) = 368.98 + 82.38X_1$ and $f_2(x) = \begin{cases} 377.91 + 104.89X_1; \text{ if } 1 \leq X_1 < 80 \\ 438.23 + 52.04X_1; \text{ if } 80 \leq X_1 < 96 \end{cases}$	40.7272	24.8767	NA

Table 7.6: Tasks models: tree-based models(4)

Task	Model	RMSE	s	Time(s)
CE2	$\hat{Y} = \begin{cases} f_1(x), \text{if } X_{11} = 0; \\ f_2(x), \text{if } X_{11} = 1. \end{cases}$ where $f_1(x) = \begin{cases} 336.05 + 75.74X_1; \text{ if } 1 \leq X_1 < 27 \\ 210.01 + 56.89X_1; \text{ if } 27 \leq X_1 < 54 \\ 240.19 + 49.02X_1; \text{ if } 54 \leq X_1 < 80 \\ 143.16 + 46.36X_1; \text{ if } 80 \leq X_1 < 96 \end{cases}$ and $f_2(x) = \begin{cases} 551.57 + 81.56X_1; \text{ if } 1 \leq X_1 < 27 \\ 422.24 + 62.89X_1; \text{ if } 27 \leq X_1 < 54 \\ 445.19 + 55.09X_1; \text{ if } 54 \leq X_1 < 80 \\ 363.72 + 52.25X_1; \text{ if } 80 \leq X_1 < 96 \end{cases}$	27.55	18.04	NA

Table 7.7: Tasks models: tree-based models(5)