國立交通大學

統計學研究所

碩士論文

迴歸樹在半導體良率提升之應用

Modified Regression Tree

and their Applications in Semiconductor Yield Improvement

研究生：賴政言

指導教授：盧鴻興 博士

中華民國九十七年六月

國立交通大學

統計學研究所

碩士論文

迴歸樹在半導體良率提升之應用

Modified Regression Tree

and their Applications in Semiconductor Yield Improvement

研 究 生：賴政言

指導教授：盧鴻興 博士

中華民國九十七年六月

Modified Regression Tree
and their Applications in Semiconductor Yield Improvement

研 究 生：賴政言 Student: Zheng-Yen Lai

指導教授：盧鴻興 博士 Advisor: Dr. Horng-Shing Lu

國立交通大學

統計學研究所

碩士論文

A Thesis
Submitted to Institute of Statistics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Statistics
June 2008
Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

改良的迴歸樹在半導體良率提升之應用

研究生：賴政言　　　　　　　　　　　　指導教授：盧鴻興　博士

國立交通大學統計學所　　碩士班

摘　　要

在半導體產業中，產品之良率高低將影響公司之營運成本與競爭力，故提升良率是每一間公司的重要目標，然而技術的進步固然重要，確保其產品之良率維持在應有之水準更為重要。由於半導體產業製程相當複雜，產品中之檢驗站往往是需經過數百個製程站才可執行，若其良率發生變異，要從中找出有問題的製程站，對於工程師而言為一大挑戰。

在現有文獻中，對於解決檢驗良率是否發生變異並找出其正確位置，尚未有一較佳的方式，因此開發偵測良率之自動化系統極為重要。在本篇論文中，將利用數學方式建立模型，提供偵測良率之方式，使工程師更有效率的解決有問題的製程站。

本篇論文所提供之方式，主要想法是來自於 CART (Classification And Regression Tree)中之迴歸樹作法，並對其做一改良，進而將此應用至半導體產業界上。由於半導體產業中製造過程，常會出現離群值，其對於數學上建立模型為一困擾，因此在本文中對於離群值之出現，亦提供一方式來解決離群值對所建立之模型影響。

而在本文中，將會與 2000 年 Wayne A. Taylor 博士所發表的方法作比較。利用模擬的方式，建立均值平移之模型，模擬產品良率變動之情形，並且以偵測出其變異所發生之位置來比較其正確率。

Modified Regression Tree
and their Applications in Semiconductor Yield Improvement

student：Zheng-Yan Lai Advisors：Dr. Horng-Shing Lu

Institute of Statistics
National Chiao Tung University

ABSTRACT

In the semiconductor industry, a yield rate will affect the cost of business and the competitive power the company. Therefore, promoting a yield rate contributes to each company's profitable target. However, not only is the technical progress undoubtedly important, but a company's guarantee that its product will have a standard yield rate is also important. Because the semiconductor industry's system regulation is quite complex, a product must pass through hundreds of process stations to completely manufacture the product. After completing a system of ownership regulation, the product will be able to detect its yield rate. Therefore when the yield rate varies, it is an enormous challenge for engineers.

In current literature, there is no good way to solve the process of detecting whether to have variation and to discover a correct position. Therefore it is very important to develop an automatic system to detect the variation of the yield rate. In this paper, we will establish a model using mathematics, provide a way to detect the yield rate, and provide engineers a more effective solution to find the problem station.

The main ideal of this paper comes from CART (Classification And Regression Tree). This paper improves on it and then applies this method to the semiconductor industry. In the manufacturing process in the semiconductor industry, the regular session presents the outlier, and it is confusing to use mathematics in the model. Therefore the appearance of an outlier also provides a way to solve it.

Also, in this paper, our method will compare the accuracy rate with CPD (statistical Change-Point Detection), which was proposed by Dr. Wayne A. Taylor (2000a). Using a simulation, we set up models of the mean shift to simulate situations of changing product yield rates and use the detection of its varying position to compare its accuracy.

誌謝

於交大的研究生活即將告一段落，在這兩年裡，讓我成長了許多，獲得很多的經驗，包含做人處事的道理、求學的態度、做學問的精神等等。而在這段日子裡，首先我要感謝我的指導教授—盧鴻興博士的諄諄教誨，老師給我很多的發展空間，讓我在求學過程中，能嘗試用不同的思維來解決問題。再來要感謝交大的每一位老師，在求學過程中給予我很多的幫助。接著感謝博士班涂凱文學長，讓我有機會可以進到科學園區，體驗了工程師的生活，也在這一年中，啟發了我很多對事情的概念與想法，讓我更懂得如何去瞭解事情。感謝陳泰賓學長分享了很多生活的經驗，以及程式的教導，開啟了我對程式的興趣。更感謝班上的每一位同學，在這兩年中，能和樂的相處做學問，共同奮鬥與生活，讓我每一天都可以過的很開心。最後感謝我的家人，給予我生活上與心靈上的支柱，讓我可以專心求學順利完成論文。

同時感謝口試委員洪志真博士、許文郁博士和陳君厚博士提供諸多寶貴的建議，使得本論文更加完善。

賴政言　謹　誌于
國立交通大學統計學研究所
中華民國九十七年六月

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1. 1 Motivation and objectives

In the semiconductor industry, a better wafer yield is equivalent not only to the quality of the products in the company but also to operation costs and completion. In a good company, high quality is good and often increases its competitive power. Therefore each company's management promotes yield as its production profit target.

But a semiconductor's system is quite complex, frequently passing through hundreds of process stations to be able to completely manufacture the product. And after completing a system of ownership regulation, the product will be able to detect its yield rate. Therefore when the yield rate varies, it is an enormous challenge for engineers.

If some system regulation takes place the problem at time t, there will occur two distributions of yield rates around time t (Figure 1). According to the product in this station's production order, engineers draw the trend chart of a product's yield rate. He will find that the trend shifts the mean if there is a problem at the time t. An engineer would think that the problem which influenced the yield rate is at time t and need to check it.

The traditional SPC method is not suitable for this question. There are two primary causes: first, the product must be able to get the yield rate detection through hundreds of system regulation stations and then obtain the product's yield rate. Second, the semiconductor industry's production pattern is not necessarily in accordance with FIFO (first-in and first-out), such that the first product produced may not necessarily complete the whole manufacturing process first and detect its yield rate. Hence, we must find a new way to monitor the yield rate in the process.

On different fields, there are many methods to solve the mean-shift problem. CPD (statistical Change-Point Detection) is used widely. For example, CPD may be used to discover the nerve where the fission in biosphere appears [11]; it also applied to monitor the semiconductor yield rate in semiconductor industry. [15]

Moreover, we can use CART (Classification And Regression Tree) to detect the mean-shift problem. However, it is difficult to select the cost-complexity, which describes the order of complexity of the model. In Statistics, we will use cross-validation (Seymour Geisser, 1929 – 2004) to determine the cost-complexity, but its shortcoming lies in the its complex computation, and it results in small samples, which is bad.

Because in the semiconductor industry, manufacturing processes are quite complex, people sometimes avoid some outlier materials that are produced because of artificial mistakes. Engineers hoped to understand a regulation system with an overall tendency to exclude the effect of outliers. As a result of the outliers' appearance, the outcome usually changes tremendously. Thus, there needs to be a method to deal with outliers. In Figure 2, and Figure 3 we can find that the outliers affect the outcomes.

Finally, in the semiconductor field, the components are often measured by different instruments, so methods must be effective to detect the location of the mean shift suitably in different situations for engineers and be able not to influence the unit of measurement.

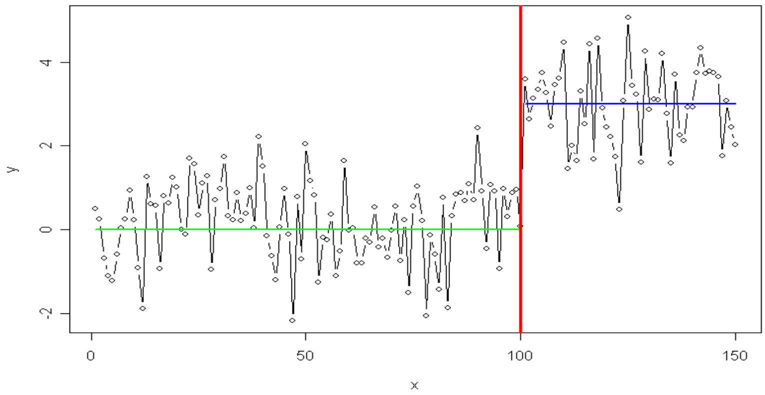


Figure 1 Some system regulation takes place the problem at time t.

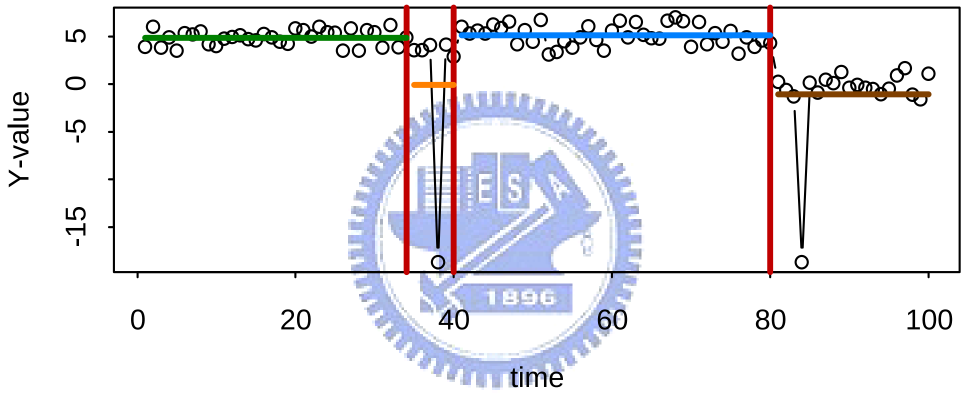


Figure 2 The outliers affect the outcomes.(1)

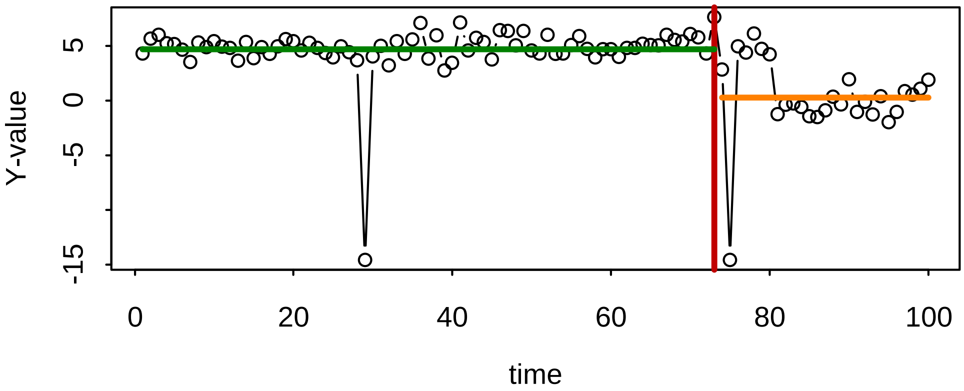


Figure 3 The outliers affect the outcomes.(2)

1. 2 The procedure of research

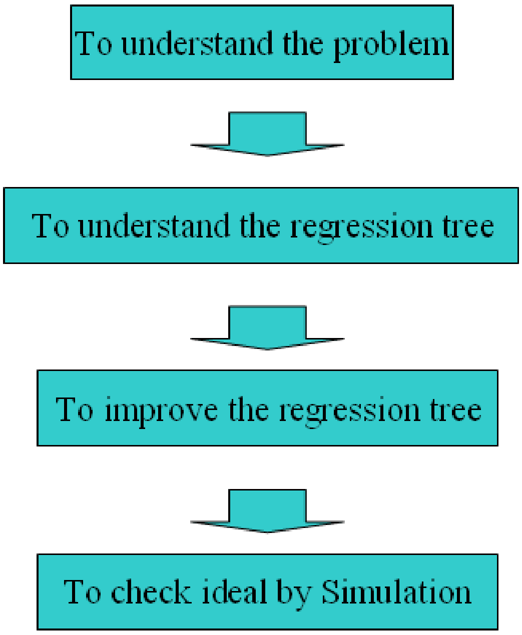

To understand the problem
To understand the regression tree
To improve the regression tree
To check ideal by Simulation

Figure 4 The procedure of research

1. To understand the problem

In the semiconductor field, solving the mean-shift problem most often involves using CPD, which is a method Taylor proposed in 2000 to find a change-point, mainly using a cumulated sum (CUSUM) method. However, the simulated yield rate was unable to satisfy the expectations of an engineer. Therefore we need to research and develop a method to get a high simulated yield rate for the mean-shift problem and widely to apply it to different places.

2. To understand the regression tree

A regression tree is a fast calculating method which uses dichotomy to quickly divide data. In this way, it will clearly understand the whole properties of data. But, there is no good way in literature to choose the size of model. Moreover, we discover that the result of partitioning is very easily wrong if outliers exist. Thus, to deal with outliers is also an important link.

3. To improve the regression tree

We propose a method to improve the regression tree. The main quotation of the new method is Bayesian, and the concept of an influence point in regression analysis is to choose the infection of outliers.

4. To check ideal by simulation

Using the different models, we discuss the simulated classification rate of the new method. Then from simulation results, the new method's simulated classification rate is high. And it does not affect unbalanced data, different units, and less affect the outliers.

1. 3 Organization

This thesis is organized as follows. Chapter 1 outlines the procedure of this research, motivation and objective. In Chapter 2, we describe literature review, which contains two methods to detect the location of mean-shift. In Chapter 3, we propose our new method, which is improved by CART and deals with outliers by the view of influence point. In addition, we verify our method by using simulation result in Chapter 4 and conclude the thesis in Chapter 5.

Chapter 2: Literature review

2. 1 Using CPD to detect the mean-shift problem

CPD (statistical Change-Point Detection) was proposed by Dr. Wayne A. Taylor (2000a). This method can detect the change-point problem. Refer to this homepage: http://www.variation.com/cpa/tech/changepoint.html. In this paper, it was introduced by an example for US trade deficit data.

Taylor (2000a) uses the procedure to find out the location of change for execution change analysis. It mainly uses the tools cumulative sum chart (CUSUM) and permutation test.

In this paper, the significant level for permutation tests is 95%. If the test result was significant, Taylor would use CUSUM to find out the location of mean-shift.

Suppose the data is Y_i ， $i=1,\ ...,\ N$, and the significant level is α. At first Taylor calculates the cumulative sums.

$$S_i = S_{i-1} + (Y_i - \bar{Y}),\ \ S_0 = 0 \tag{2.1}$$

Then he calculates the value S_{diff}^{0} , where

$$S_{diff}^{0} = S_{\max} - S_{\min}\text{ , and }\begin{array}{l} S_{\max} = \max\limits_{i=0,\ ...,\ N} S_i \\ S_{\min} = \min\limits_{i=0,\ ...,\ N} S_i \end{array}$$

Taylor repeats the above movement B times by permuting the data, and gets the values $S_{diff\,1}$, $S_{diff\,2}$, ..., $S_{diff\,B}$. By calculating the counts of $S_{diff\,i} > S_{diff}^{0}$, he gets its confidence level: $100\times\frac{\text{count}}{B}\%$. If $100\times\frac{\text{count}}{B}\% > \alpha$, he will conclude the trend is changed in some position. Then he will find out the location of mean-shift by CUSUM and separate the data into two sections.

$$S_m = \max |S_i| \tag{2.2}$$

At last, he finds out all the changes in this trend repeatedly for each section. The algorithm is as follows.

1. Given the significant level α, all data are named resource data.

2. Take resource data to input data0.

3. Calculate $S_{diff}^{0} = S_{\max} - S_{\min}$, where $S_i = S_{i-1} + Y_i - \bar{Y}$.

4. By permuting the input data B times, get B sequences of the new input data.

5. Repeat step 4 ~ step 5 to get $S_{diff\,1}, S_{diff\,2}, \ldots, S_{diff\,B}$.

6. Calculate the counts $S_{diff\,i} > S_{diff}^{0}$.

7. The significant level is $100 \times \frac{\text{count}}{B} \%$.

8. If $100 \times \frac{\text{count}}{B} \% \geq \alpha$, go to step 11 ~ step 12. Else conclude that there is no shift in this section and break.

9. By CUSUM chart $S_m = \max |S_i|$, to find out the location of mean-shift.

10. Record the location of mean-shift, and partitioned the two input data1 and input data 2.

11. Take input data 1 and input data 2 individually to input data 0, and go to step 3 ~ step 12.

12. Get all of the locations of mean-shift.

2. 2 Using regression trees to detect the mean-shift problem

2. 2. 1 Introduction of regression trees

In 1963, Morgan and Sonquist proposed Automatic Interaction Detection (AID) to get the optimum model by minimizing the mean square error. The regression tree was a development traced back to Morgan (1964), Sonquist (1970), Sonquist, Morgan (1973), Fielding (1977), Van Eck (1980), and Leo Breiman, Jerry Friedman, Charles J, .Stone, Richard Olshen, who proposed CART in 1984.

CART is an algorithm to separate data by using a binary decision tree. The algorithm for the material divides the parental node to two child nodes, using recursive partitioning from top to down to establish a complete tree. The following figure, **Figure 5**, makes an introduction using a graphical representation to construct the tree.

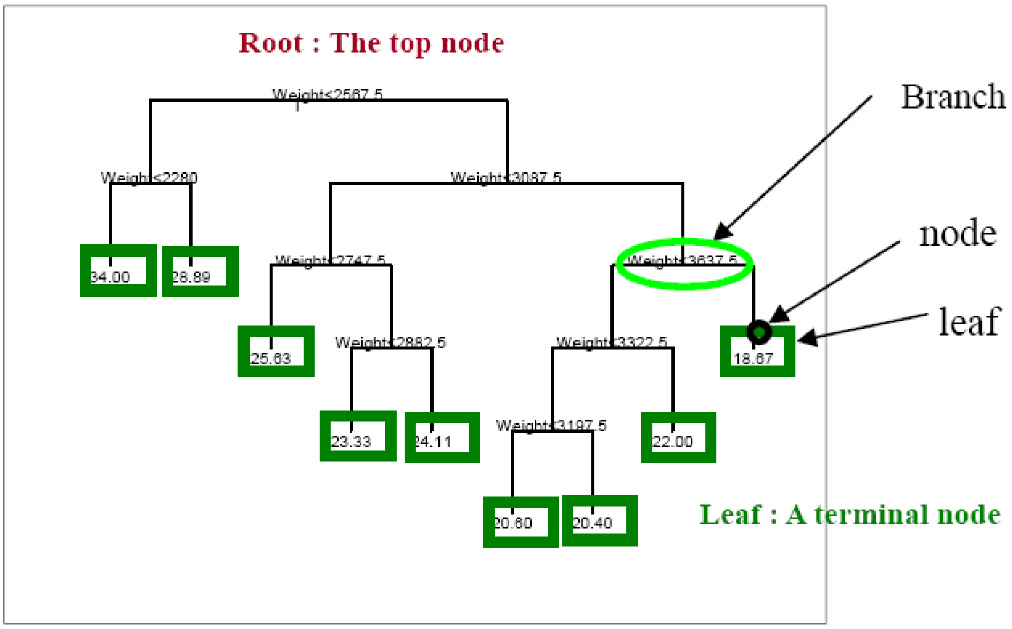


Figure 5 Construction of a tree

Suppose $(x_i,\ y_i),\ i=1,\ ...,\ N,$ with $x_i=(x_{i1},\ x_{i2},\ ...,\ x_{in})$. The algorithm needs to automatically decide the split points. Suppose we have a partition of K regions $R_1,\ R_2,\ ...,\ R_K$. Then our response model is denoted by

$$f(x)=\sum_{k=1}^{K} c_m I(x \in R_k) \tag{2.3}$$

where c_m is a constant in each region.

If we adopt minimization of the sum of squares as the criterion for the split rule, we will use $\hat{c}_m$ to estimate c_m, where $\hat{c}_m$ is the average of y_i in the region R_k.

$$\hat{C}_{\mathrm{m}}=\text{average}(y_i|x_{\ i} \in R_m) \tag{2.4}$$

We will illustrate the regression tree in three sections. We will say how to find the best point to split the data in 2.2.2 and how to select the tree size in 2.2.3.

2. 2. 2 Partition

We find the best binary partition by minimizing the sum of squares. The goal of the partition is to decrease the error in each group. We seek the splitting variable *j* and split point *s* by solving as follows below.

$$R_1(j,s)=\{X|X_j \le s\} \text{ and } R_2(j,s)=\{X|X_j > s\} \tag{2.5}$$

$$\arg\min_{j,s}[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i-c_2)^2] \tag{2.6}$$

For any choice *j* and *s*, the solution of c_1 and c_2 are estimated by $\hat{c}_1$ and $\hat{c}_2$, where $\hat{c}_1$ and $\hat{c}_2$ are as follows below.

$$\begin{aligned}\hat{c}_1 &= \text{average}(y_i|x_i \in R_1(j,\ s)) \\ \hat{c}_2 &= \text{average}(y_i|x_i \in R_2(j,\ s))\end{aligned} \tag{2.7}$$

We partition the data into two resulting regions and repeat the splitting process on each of the two regions. Then the process will split the data into individual sections.

2. 2. 3 Pruning

How large should we grow the tree? A large tree might over-fit the data; on the contrary, a small tree might not describe the important structure. Tree size will be a parameter to control the model's complexity, so how do we choose the tree size?

Traditionally, there are two ways to prune trees for choosing a tree size. One is pre-pruning, and the other is post-pruning. The algorithm of pre-pruning is setting some criteria to determine how to stop the tree from growing, and the algorithm of post-pruning is pruning a tree by some criteria after growing a complete tree. The criteria of pre-pruning are too short-sighted, however, since a seemingly worthless split might lead to a good split below it. So, we will use post-pruning to choose the tree size in this paper.

Venables and Ripley proposed to choose the terminal node by these two criteria:

1. $\max_{s} \Delta R(s,\ t) \le 0.006R(t)$, i.e. the sum of squares after a spilt is smaller than the original data by 0.006 times.
2. The size of terminal node is at least 5.

Then this large tree is pruned by using cost-complexity pruning. Suppose T is a subtree of T_0, and with $|T|$ terminal nodes, where $|T|$ is the number of terminal nodes in T. We index terminal nodes by k, and we represent a region by R_k. Suppose

$$\hat{c}_k = \frac{1}{N_k} \sum_{x_i \in R_k} y_i \tag{2.8}$$

$$Q_k(T) = \frac{1}{N_k} \sum_{x_i \in R_k} (y_i - \hat{c}_k)^2, \tag{2.9}$$

the cost complexity criterion is represented by

$$C_{\alpha}(T)=\sum_{k=1}^{|T|}N_{k}Q_{k}(T)+\alpha\times|T| \qquad (2.10)$$

where

N_k is the number of the observation data falling in the region R_k.

k is the index of terminal nodes on the binary tree T.

$|T|$ is the number of terminal nodes in T, and.

α is the cost-complexity ($\alpha \geq 0$).

For given a cost-complexity α, we can get a subtree T_α to minimize $C_\alpha(T)$. From this formula, we can find the larger value α gets, the smaller subtree T_α gets. For given each value α, we can get a unique smallest subtree T_α. If $\alpha = 0$, we can get a full tree.

2. 2. 4 The challenge of using regression trees to detect mean-shift

In Figure 6, we can find the relation between the mean-shift trend chart and regression tree. We can detect the mean-shift by this way, but there are many challenges in this question.

For a given data, and we can plot its trend chart (**Figure 7**). If using different cost-complexity α in this data, the results of regression tree will different as in **Figure 8**. Given the same α using different scales to change the data, we can find different results of regression trees in **Figure 9**. If the number of mean-shifts is more than one, a major mean-shift would dominate the decision of a minor mean-shift if the major mean-shift is large. The challenge of a regression tree is how to give an adequate cost-complexity value for the data.

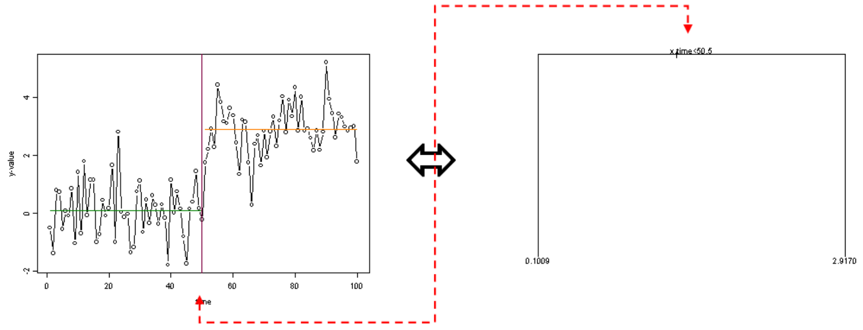


Figure 6 The relation between the mean-shift trend chart and regression tree

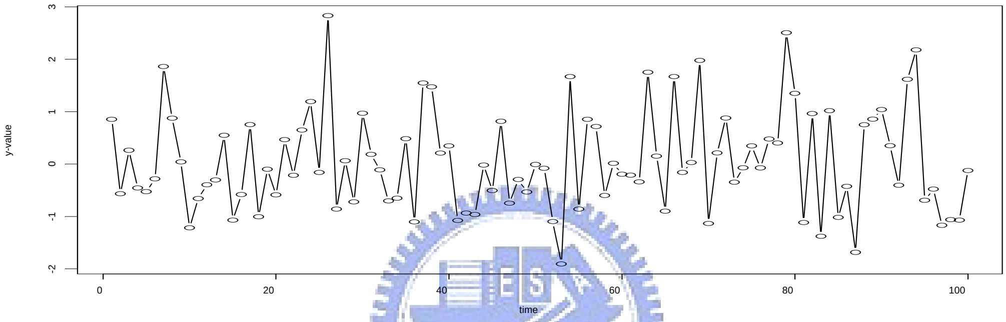


Figure 7 Trend chart of row data

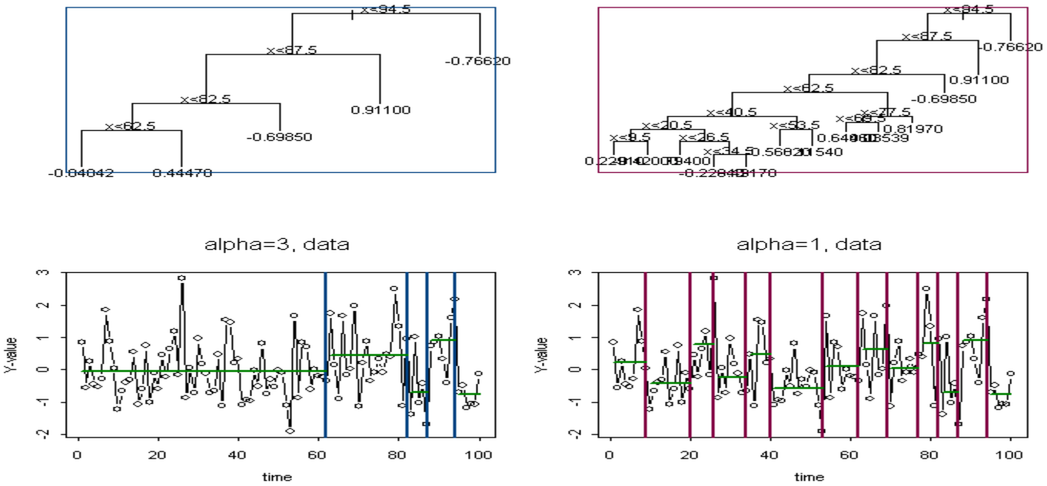


Figure 8 The results of tree with different cost-complexity values are different.

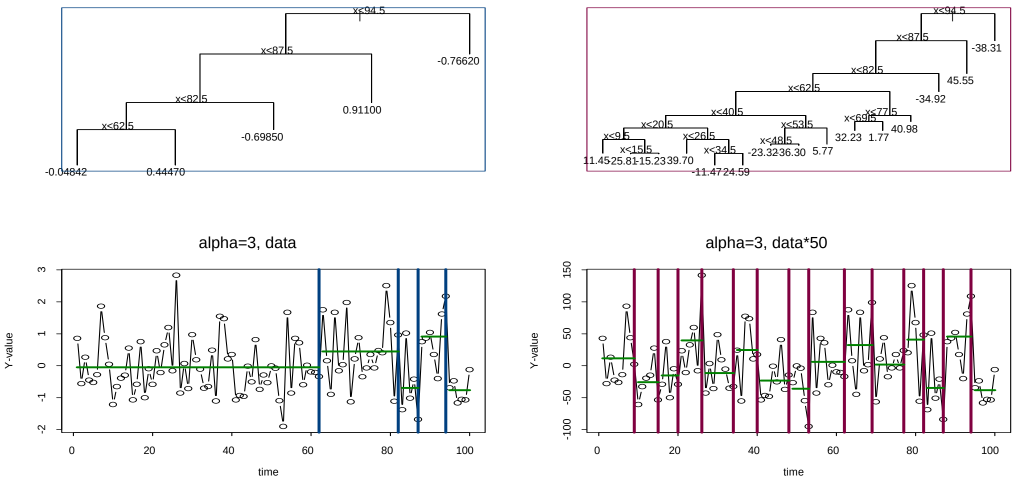


Figure 9 The results of tree with different scales are different.

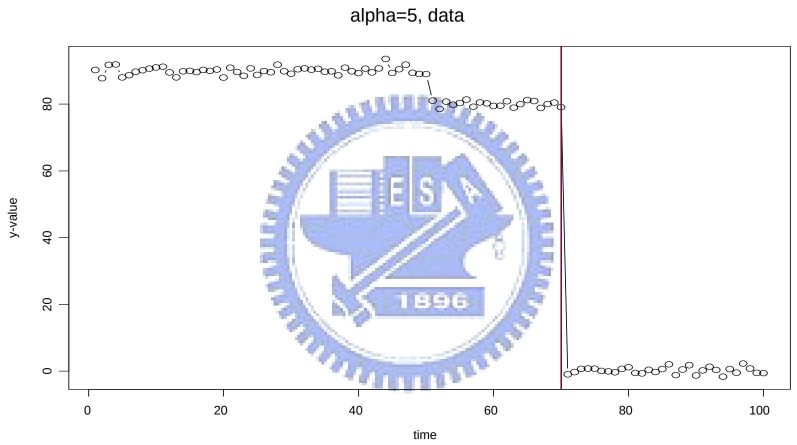


Figure 10 A major mean-shift would dominate the decision of a minor mean-shift.

2. 2. 5 Cross-Validation

Cross-validation (Stone, 1974, Stone 1977, and Allen 1977) is the most widely used method for estimating prediction errors in machine learning. Also, it is used in regression trees to choose the optional model.

Suppose Y is a target variable, X is a vector of inputs, and a prediction model $\hat{f}(X)$ has been estimated from a training sample. Then this method estimates the extra sample error

$$\mathrm{Err} = E\left[L\left(Y,\ \hat{f}(X)\right) \right], \tag{2.11}$$

where $L\left(Y,\ \hat{f}(X)\right)$ is the loss function for measuring errors between Y and $\hat{f}(X)$, and is described as follows:

$$L(Y,\ \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ \left| Y - \hat{f}(X) \right| & \text{absolute error} \end{cases}$$

Suppose we split the data into K equal-sized parts. For the $k-\mathrm{th}$ part, we fit the model with the other $K-1$ parts of the data and then calculate the prediction error of the $k-\mathrm{th}$ part of the data, where $k = 1,\ \ldots,\ K$. Let $\kappa : \{1,\ \ldots,\ N\} \mapsto \{1,\ \ldots,\ K\}$ be an indexing function, and let $\hat{f}^{-k}(x)$ denote the fitted function for removing the $k-\mathrm{th}$ part of the data. Then the cross-validation estimate of prediction error is

$$CV = \frac{1}{N}\sum_{i=1}^{N} L(y_i,\ \hat{f}^{-\kappa(i)}(x_i)). \tag{2.12}$$

If $K = N$, it equals leave-one-out cross-validation.

Given a set of models $f(x,\ \alpha)$ indexed by a tuning parameter α, we denote the fitted function for removing the $k-\mathrm{th}$ part of the data as $\hat{f}^{-k}(x,\ \alpha)$. Then the cross-validation estimate of prediction error is

$$CV(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i,\ \hat{f}^{-\kappa(i)}(x_i,\ \alpha)). \tag{2.13}$$

Then we find the tuning parameter $\hat{\alpha}$ to minimize it and choose the model $\hat{f}(x,\ \hat{\alpha})$ to fit the data. Traditionally, tools like five-fold cross-validation or ten-fold cross-validation are widely used to estimate the error. The algorithm is as follows below.

Algorithm of Cross-Validation Tree

1. Suppose Y is a target variable, X is a vector of inputs, and α is a tuning parameter.
2. Split the data into K groups by random chance.
3. Take $K-1$ groups as training set, and group k as testing set.
4. Set up the model by the $K-1$ groups.
5. Predict the group k by the model from step 4.
6. Repeat from the step 3 to step 5 to calculate $CV(\alpha)$, where $CV(\alpha)=\frac{1}{N}\sum_{i=1}^{N}L(y_i,\ \hat{f}^{-\kappa(i)}(x_i,\ \alpha))$, $\kappa:\{1,\ \ldots,\ N\}\mapsto\{1,\ \ldots,\ K\}$, and
$$L(Y,\ \hat{f}(X))=\begin{cases}(Y-\hat{f}(X))^2 & \text{squared error}\\ \left|Y-\hat{f}(X)\right| & \text{absolute error}\end{cases}$$
7. Find a tuning parameter $\hat{\alpha}$ and minimize it.
8. In program, repeat r times from step 1 to step 7 to get $\left\{\hat{\alpha}_1^*,\ \ldots,\ \hat{\alpha}_r^*\right\}$.
9. Use $\bar{\hat{\alpha}}^* = \dfrac{\hat{\alpha}_1^* +\ \ldots +\ \hat{\alpha}_r^*}{r}$ to set up the model.

2.3 Outlier

In general, an outlier is an observed value that is numerically distant from the rest of the data.

However, an outlier appearance will create many puzzles. First, you must suspect whether this outlier is there because of some kind of mistakes, perhaps such as external factors. And maybe we can consider giving up this outlier, according to the least squares error method principle, as the outlier will change the model if it exists. But the outlier could also possibly be directed to contain some important information, such as perhaps some key points existing in this discovery. Therefore we suggest deleting it when we were certain the outlier is due to other reasons.

In regression analysis, there are some points called high leverage points [12], if they have the influence to change the model. We can find that the model has a large change if point A existed in **Figure 11**.

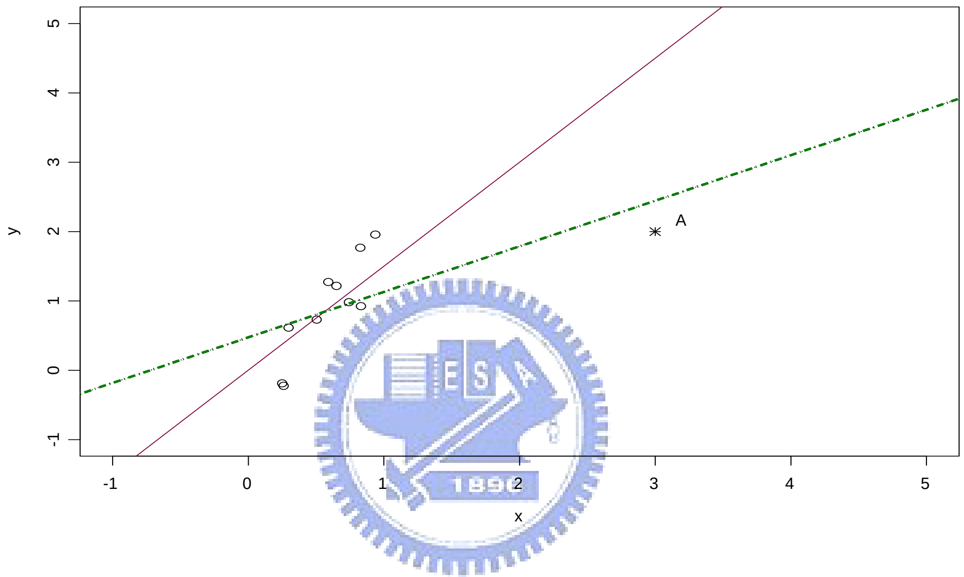


Figure 11 Example of an influential point

Chapter 3: New Method

3. 1 Introduction of a new method

The new method is an improvement on the Bayesian method for pruning in the algorithm of CART. By this method, it will be easy to choose the cost-complexity for data.

Let Y be a piecewise stationary mean process and be as follows according to mathematical symbols.

$$Y_i = \mu_i + \varepsilon_i,\ 1 \le i \le N,\ \varepsilon_i \sim N(0,\ \sigma_\varepsilon^2), \tag{3.1}$$

where μ_i is the mean of i-th component

N is the sample size, and

σ_ε^2 is Within-variance

Suppose the data has K steps, and the location of mean-shift are at $t_1,\ ...,\ t_k,\ ...,\ t_{K-1}$, $t_0 = 1,\ t_K = N$, and $1 = t_0 < t_1 < ... < t_k < ... < t_K = N$. The data is split into these $k-1$ points, $t_1,\ ...,\ t_k,\ ...,\ t_{K-1}$, the mean in each section is $\theta_1,\ ...,\ \theta_k,\ ...,\ \theta_K$ in order, and is as follows according to mathematical symbols.

$$\mu_{\text{i}} = \theta_k \text{ for } t_k \le i \le (t_{k+1} - 1) \tag{3.2}$$

In this question, K and t are a random variables, because we don't know how many times the mean shifted in the data or where the locations of mean-shift are.

We suppose that there exists a binary sequence R with length N to represent the data, where

$$R_i = \begin{cases} 1 \text{ if } i = t_k,\ \text{for } 1 \le k \le K,\ 1 \le i \le N \\ 0 \text{ if } i \ne t_k,\ \text{for } 1 \le k \le K,\ 1 \le i \le N \end{cases}. \tag{3.3}$$

Then we can say that the number of times of mean-shift equals the sum of the sequence R, and the mean-shift happens when $R_i = 1$. Therefore we will want to find the most probable sequence R to describe the trend of the data. If the sequence is established, then $\vec{\mu}$ ($\vec{\mu} = (\mu_i)_{i=1,\ldots,N} = \vec{\mu}(\theta)$, where $\theta = (\theta_0,\ \theta_1,\ \ldots,\ \theta_K)$) become parameters.

We can suppose that R is a Bernoulli sequence with probability λ, $0 < \lambda < 1$. Given the data Y, we can find the prior probability $P(R \mid Y)$, which is the most probable mean-shift model.

$$P(R \mid Y) = \frac{P(Y \mid R)P(R)}{P(Y)} \propto P(Y \mid R)P(R) \tag{3.4}$$

$$\propto \left[\prod_{k=0}^{K} \prod_{i=t_k}^{t_{k+1}-1} \frac{e^{-\frac{(y_i-\theta_k)^2}{2\sigma_\varepsilon^2}}}{\sqrt{2\pi\sigma_\varepsilon^2}} \right] \times \lambda^K (1-\lambda)^{N-K} \tag{3.5}$$

And the most probable mean-shift model is

$$\underset{k,\theta}{Max}\big(P(R \mid Y)\big) \propto \underset{k,\theta,R}{Max}\left(\left[\prod_{k=0}^{K-1} \prod_{i=t_k}^{t_{k+1}-1} \frac{e^{-\frac{(y_i-\theta_k)^2}{2\sigma_\varepsilon^2}}}{\sqrt{2\pi\sigma_\varepsilon^2}} \right] \times \lambda^K (1-\lambda)^{N-K} \right) \tag{3.6}$$

$$\propto \underset{k,\theta}{Min}\left(-\log\left(\left[\prod_{k=0}^{K-1} \prod_{i=t_k}^{t_{k+1}-1} \frac{e^{-\frac{(y_i-\theta_k)^2}{2\sigma_\varepsilon^2}}}{\sqrt{2\pi\sigma_\varepsilon^2}} \right] \times \lambda^K (1-\lambda)^{N-K} \right) \right)$$

$$= \underset{k,\theta}{Min}\left\{ -\log\left(\prod_{k=0}^{K-1} \prod_{i=t_k}^{t_{k+1}-1} e^{-\frac{(y_i-\theta_k)^2}{2\sigma_\varepsilon^2}} \times \left(\frac{\lambda}{1-\lambda} \right)^K \times \left(\frac{1-\lambda}{\sqrt{2\pi\sigma_\varepsilon^2}} \right)^N \right) \right\} \tag{3.7}$$

$$\propto \underset{k,\theta}{Min}\left(\sum_{k=0}^{K} \sum_{i=t_k}^{t_{k+1}-1} \frac{(y_i-\theta_k)^2}{2\sigma_\varepsilon^2} + K\log(\frac{1-\lambda}{\lambda}) \right)$$

$$= \underset{k,\theta}{Min}\left(\sum_{k=0}^{K} \sum_{i=t_k}^{t_{k+1}-1} \frac{(y_i-\theta_k)^2}{\sigma_\varepsilon^2} + 2\log(\frac{1-\lambda}{\lambda}) \times K \right)$$

$$= \underset{k,\theta}{Min}\left(\sum_{k=0}^{K} \sum_{i=t_k}^{t_{k+1}-1} (y_i-\theta_k)^2 + 2 \times \sigma_\varepsilon^2 \times \log(\frac{1-\lambda}{\lambda}) \times K \right) \tag{3.8}$$

, where K is the number of segments.

To compare this equation with the formula of cost-complexity pruning (CCP),

$$Max\left(P(R\,|\,Y)\right)$$
$$\propto Min\left(\sum_{k=0}^{K}\sum_{i=t_k}^{t_{k+1}-1}\left(y_i-\theta_k\right)^2+2\sigma_\varepsilon^2\log(\frac{1-\lambda}{\lambda})\times K\right),$$

where K is the number of segements

$$D_\alpha(T')=\sum_{j=1}^{|T'|}\sum_{i=1}^{n_j}(y_{ij}-\bar{y}_{.j})^2+\alpha\times\left|T'\right|$$

, where $\left|T'\right|$ is the number of terminal nodes of T'.

We can estimate α by $2\sigma_\varepsilon^2\log(\frac{1-\lambda}{\lambda})$.And the way to choose λ is by using the probability for the normal distribution shift. In this paper, we use the probability over three standard deviations to estimate λ. In the data, if we choose a larger λ, the amount of subgroups becomes more complex.

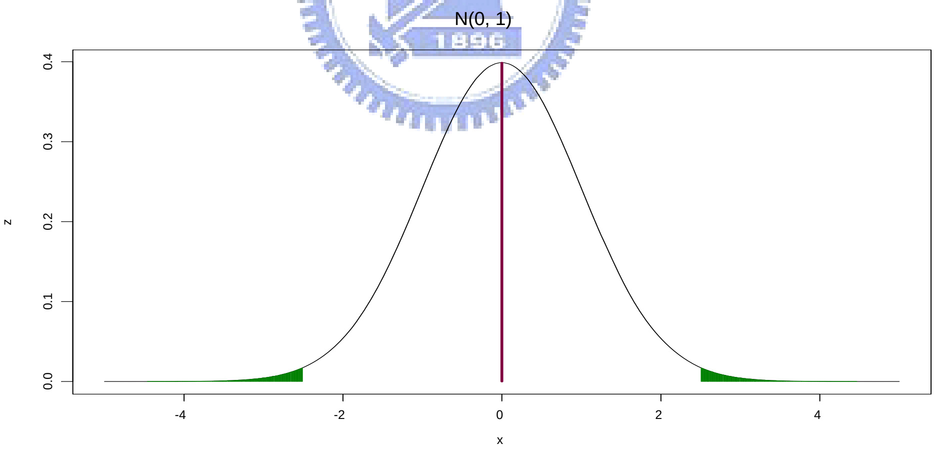


Figure 12 The probability of the sequence that happened

And we use $\hat{s}^2$ to estimate σ_ε^2,

$$\text{where}\quad \hat{s}^2=\frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k}(x_{ij}-\bar{x}_{.j})^2}{N-K} \tag{3.9}$$

N is the number of data

K denotes it has K steps, and

n is the number of the terminal node.

If we suppose the prior probability of λ is the probability over three standard deviations (0.0027), then $\hat{\alpha} = 2\sigma_{\varepsilon}^{2}\log(\frac{1-0.0027}{0.0027}) = 11.8236 \times \hat{s}^{2}$.

3. 2 The influence of outliers

If some are outliers existing in the data, it would maybe be a mistake to estimate the model. For example, if the location of mean-shift is at 80, we will find the result changed for outliers added in **Figure 13** and **Figure 14**. Because of adding outliers, in **Figure 13**, the change-point changes from 80 to 72; and in **Figure 14**, the result shows more variation.

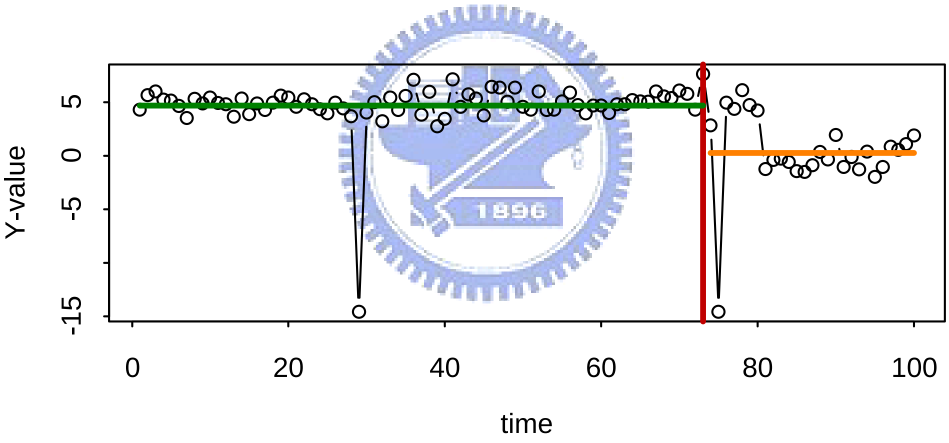


Figure 13 The location for mean-shift is shifted when the outlier exists.

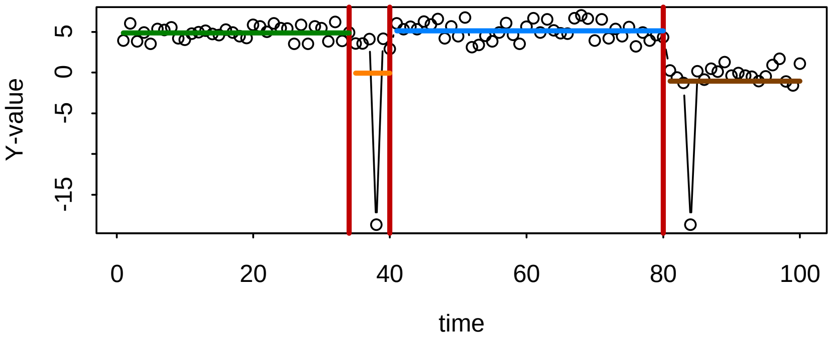


Figure 14 The model is more complex when the outlier exists.

By this viewpoint, we will know that we should examine first whether the data have the appearance of outliers. In this section, we will illustrate a method to deal with outliers. First we define the outlier, which satisfies the below:

1. Any data observation which lies more than 1.5*IQR lower than the first quartile or 1.5*IQR higher than the third quartile is considered an outlier.
2. Any data observation is an influence point.

In this paper, we will make the symbol to the outliers (**Figure 15**). Then it will be easy to see the trend of the data without losing any information.

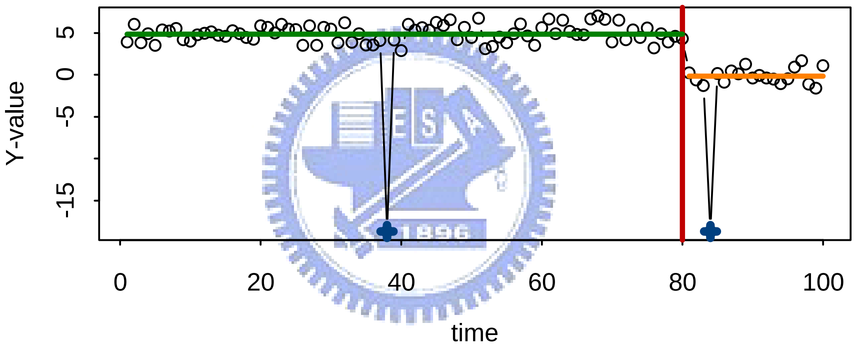


Figure 15 We will make a symbol to the outliers.

3. 3 Finding different level mean shifted by Multi-resolution

Because we need to estimate the variation within groups, but $\hat{s}^2$ contains variation within groups and variation between groups, and the estimator $\hat{s}^2$ may be larger than σ_ε^2, so we would use multi-resolution to adjust the estimator until no new mean-shift is found.

We will show our new method's flow as follows bellow.

The algorithm of new method

1. Take resource data to input data
2. Calculate the $\hat{s}$ of input data
3. Grow the tree and to prune the tree by estimator $\hat{\alpha}$.
4. Record every terminal node
5. Find whether this section has observations more than 1.5*IQR lower than the first quartile or 1.5*IQR higher than the third quartile. If true, go to step 6, else go to 7.
6. Delete the point which is recorded in step 5 to set up a new tree from step 1 to step 4 to get the new terminal node * step by step.
7. If the terminal nodes * are different from the terminal nodes, then we will delete them and go to step 1. Else go to step 8.
8. If we get new terminal nodes, take each terminal node to input data, and repeat from step 2 to step 7. Else break.

The process flow of new method is as follows below. (**Figure 16** and **Figure 17**)

Process flow

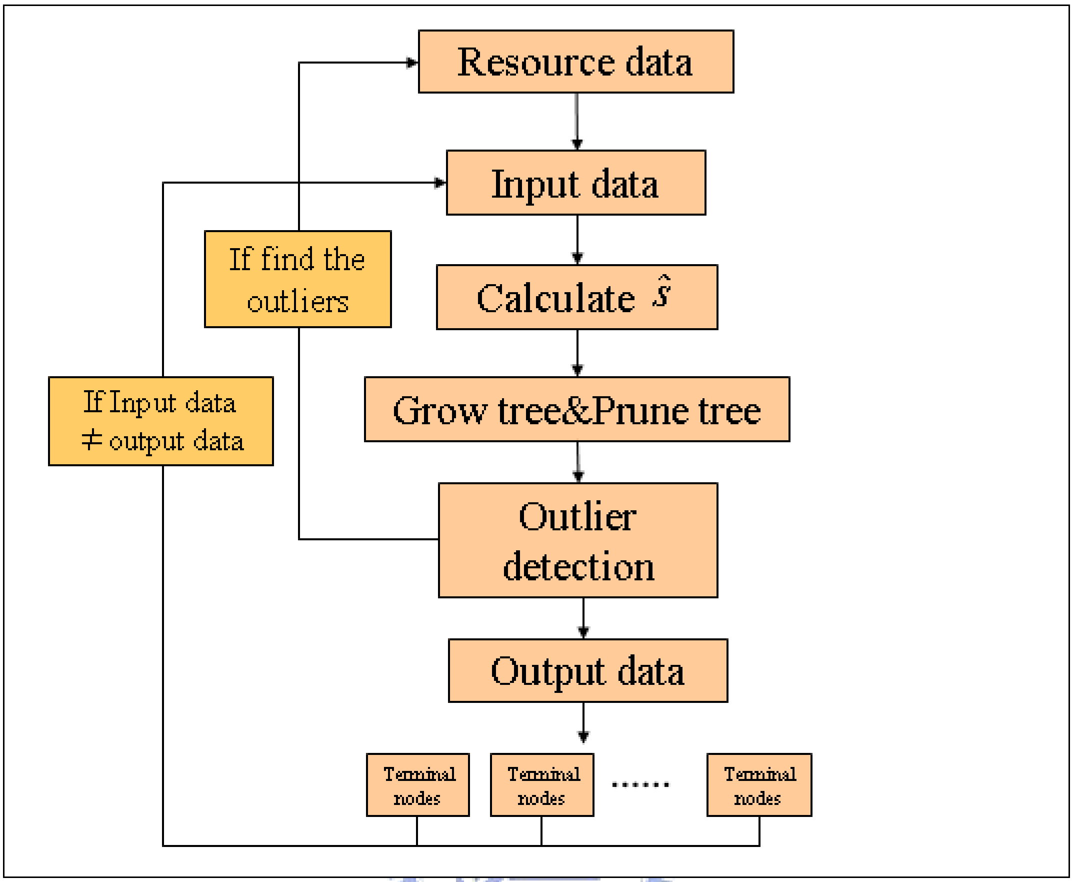
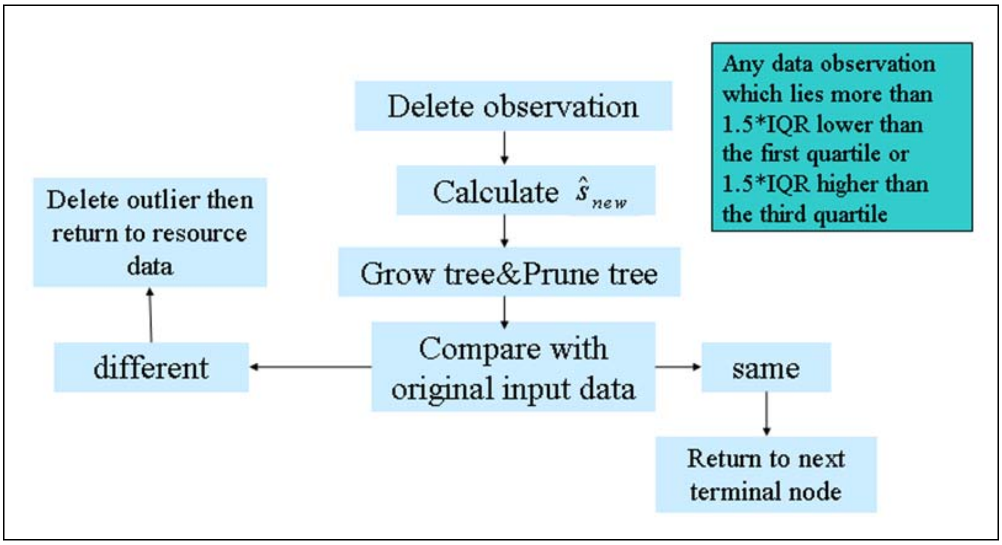


Figure 16 The process flow of the new method

Outlier detection flow



Delete observation
Calculate $\hat{s}_{new}$
Grow tree&Prune tree
Compare with original input data
same
Return to next terminal node
different
Delete outlier then return to resource data
Any data observation which lies more than 1.5*IQR lower than the first quartile or 1.5*IQR higher than the third quartile

Figure 17 The process flow of the outlier detection

Chapter 4: Experiment

In this chapter, we will discuss many cases by simulation results. In each case, the data is simulated by the joint distribution of several distributions (M_1, M_2, ...), where M_k is a normal distribution with σ_ε^2, and M_1, M_2, ... have different means.

The main cases we want to discuss follow below, and in the table's last row, we also show the simulated classification rate of all the main cases.

1. No shift, if $k=0$
2. Shifted one time, if $k=1$
3. Shifted several times, if $k>1$
4. The influence of different scales

In detail, we will discuss different levels for shifting, including one standard deviation, two standard deviations, three standard deviations, and five standard deviations. In our experiment, we take $\sigma_\varepsilon^2=1$ for every variation. We will discuss balanced data and unbalanced data. In addition, we will discuss the influence of an outlier, so we will replace two outliers from the data whose values are 5 and 6 times IQR lower than the first quartile.

Because the data size in the semiconductor industry is about of 50-200 units, our experiment is designed around 50-200, and in each case we simulate 50 times to get the simulated classification rate. In the result of simulation, we will compare three methods: CPD (bootstrap 1000 times, 95% confidence level), Cross-Validation tree, and a new method. We define the simulated classification rate by four results. First, we discuss the simulated classification rate of detecting the number of times of mean-shift; then we discuss the simulated classification rate of detecting the location of mean-shift. The rules follow below:

1. Result 1: The number of mean-shifts in the simulation result is same as the setting.
2. Result 2: The location of the mean-shift in the simulation result is same as the setting. (Appendix 4.1)
3. Result 3: The location of the mean-shift in the simulation result and the setting are close, within 3.
4. Result 4: The location of the mean-shift in the simulation result and the setting are close, within 5. (Appendix 4.2)
5. Result 5: Change the scale larger in the same data.
6. Result 6: Change the scale smaller in the same data.

4. 1 No shift

Table 1 Result1 of no shift

Result 1: *The number of mean-shifts* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
1	0	30	94%	52%	100%	96%	62%	100%
2	0	50	94%	68%	100%	90%	70%	100%
3	0	100	94%	90%	100%	92%	84%	100%
4	0	200	98%	96%	100%	98%	92%	100%
			95%	**76.5%**	**100%**	**94%**	**77%**	**100%**

Table 2 Result3 of no shift

Result 3: *The location of the mean-shift* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
1	0	30	94%	52%	100%	96%	62%	100%
2	0	50	94%	68%	100%	90%	70%	100%
3	0	100	94%	90%	100%	92%	84%	100%
4	0	200	98%	96%	100%	98%	92%	100%
			95%	**76.5%**	**100%**	**94%**	**77%**	**100%**

4. 2 Shifted one time

Table 3 Result1 of mean-shift one time

Result 1: *The number of mean-shifts* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
5	(5, 0)	(50, 50)	94%	90%	98%	94%	86%	100%
6	(3, 0)	(50, 50)	86%	84%	100%	80%	78%	100%
7	(2, 0)	(50, 50)	84%	82%	96%	90%	84%	100%
8	(1, 0)	(50, 50)	98%	90%	94%	76%	58%	70%
9	(5, 0)	(90, 10)	88%	84%	94%	68%	92%	100%
10	(3, 0)	(90, 10)	60%	86%	100%	48%	88%	100%
11	(2, 0)	(90, 10)	52%	94%	92%	32%	60%	74%
12	(1, 0)	(90, 10)	24%	30%	50%	18%	16%	8%
13	(2, 0)	(15, 15)	96%	62%	100%	46%	46%	24%
14	(2, 0)	(20, 10)	86%	62%	98%	30%	36%	8%
15	(2, 0)	(25, 25)	92%	68%	100%	92%	56%	92%
			78.18%	**75.64%**	**92.91%**	**61.27%**	**63.64%**	**70.55%**

Table 4 Result3 of mean-shift one time

Result 3: *The location of the mean-shift* in the simulation result and the setting are within 3.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
5	(5, 0)	(50, 50)	94%	90%	98%	94%	86%	98%
6	(3, 0)	(50, 50)	86%	84%	100%	74%	76%	100%
7	(2, 0)	(50, 50)	80%	76%	94%	78%	74%	94%
8	(1, 0)	(50, 50)	62%	50%	54%	46%	36%	50%
9	(5, 0)	(90, 10)	86%	84%	100%	64%	92%	96%
10	(3, 0)	(90, 10)	56%	86%	100%	28%	80%	96%
11	(2, 0)	(90, 10)	32%	92%	88%	14%	48%	70%
12	(1, 0)	(90, 10)	6%	24%	26%	4%	14%	8%
13	(2, 0)	(15, 15)	88%	60%	84%	40%	30%	24%
14	(2, 0)	(20, 10)	82%	62%	80%	26%	24%	8%
15	(2, 0)	(25, 25)	92%	68%	100%	76%	48%	86%
			69.45%	**70.55%**	**84%**	**49.45%**	**55.27%**	**66.36%**

4. 3 Shifted several times

Table 5 Result1 of mean-shift several times

Result 1: *The number of mean-shifts* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
16	(6, 3, 0)	(40, 40, 40)	36%	84%	96%	44%	84%	96%
17	(15, 10, 5, 0)	(30, 30, 30, 30)	86%	82%	96%	84%	66%	98%
18	(9, 6, 3, 0)	(30, 30, 30, 30)	88%	84%	96%	76%	70%	100%
19	(6, 4, 2, 0)	(30, 30, 30, 30)	80%	68%	96%	86%	76%	98%
20	(20, 15, 10, 5, 0)	(25, 25, 25, 25, 25)	14%	100%	96%	16%	70%	100%
21	(12, 9, 6, 3, 0)	(25, 25, 25, 25, 25)	20%	60%	86%	28%	74%	94%
22	(8, 6, 4, 2, 0)	(25, 25, 25, 25, 25)	34%	60%	86%	46%	56%	92%
23	(25, 20, 15, 10, 5, 0)	(20,20,20,20,20,20)	8%	100%	96%	20%	74%	98%
24	(15, 12, 9, 6, 3, 0)	(20,20,20,20,20,20)	26%	82%	96%	36%	54%	100%
25	(10, 8, 6, 4, 2, 0)	(20,20,20,20,20,20)	38%	50%	90%	42%	36%	84%
26	(10, 3, 0)	(40, 40, 40)	86%	78%	96%	80%	84%	100%
27	(10, 2, 0)	(40, 40, 40)	80%	74%	92%	88%	74%	98%
28	(10, 1, 0)	(40, 40, 40)	84%	72%	92%	74%	56%	76%
29	(10, 5, 2, 0)	(30, 30, 30, 30)	72%	72%	98%	78%	70%	98%
30	(10, 5, 1, 0)	(30, 30, 30, 30)	70%	64%	68%	50%	26%	62%
31	(10, 2, 0)	(50, 40, 30)	92%	72%	96%	88%	74%	98%
32	(10, 5, 2, 0)	(50, 35, 25, 10)	72%	78%	86%	68%	58%	82%
33	(10, 6, 4, 2, 0)	(40, 30, 25, 15, 10)	38%	64%	72%	60%	40%	74%
34	(5, 0, 5)	(40, 40, 40)	92%	84%	98%	94%	80%	98%
35	(3, 0, 3)	(40, 40, 40)	88%	88%	100%	90%	74%	100%
36	(2, 0, 2)	(40, 40, 40)	88%	68%	88%	82%	82%	92%
37	(5, 0, 5)	(20,60,40)	84%	82%	100%	86%	86%	100%
38	(2, 0, 2)	(20,60,40)	82%	76%	94%	78%	78%	100%
39	(5, 2, 0, 2, 4, 6)	(20,30,20,30,20,20)	54%	56%	74%	62%	66%	72%
			63%	**74.92%**	**91.17%**	**64.83%**	**67%**	**92.08%**

Table 6 Result3 of mean-shift several times

Result 3: *The location of the mean-shift* in the simulation result and the setting are within 3.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
16	(6, 3, 0)	(40, 40, 40)	20%	78%	90%	24%	76%	86%
17	(15, 10, 5, 0)	(30, 30, 30, 30)	86%	82%	96%	74%	64%	92%
18	(9, 6, 3, 0)	(30, 30, 30, 30)	86%	82%	94%	66%	56%	90%
19	(6, 4, 2, 0)	(30, 30, 30, 30)	66%	56%	74%	54%	40%	74%
20	(20, 15, 10, 5, 0)	(25, 25, 25, 25, 25)	8%	98%	96%	6%	66%	94%
21	(12, 9, 6, 3, 0)	(25, 25, 25, 25, 25)	18%	56%	80%	14%	58%	76%
22	(8, 6, 4, 2, 0)	(25, 25, 25, 25, 25)	6%	34%	34%	6%	18%	46%
23	(25, 20, 15, 10, 5, 0)	(20,20,20,20,20,20)	8%	90%	94%	16%	70%	92%
24	(15, 12, 9, 6, 3, 0)	(20,20,20,20,20,20)	14%	68%	82%	16%	36%	86%
25	(10, 8, 6, 4, 2, 0)	(20,20,20,20,20,20)	8%	30%	42%	10%	16%	40%
26	(10, 3, 0)	(40, 40, 40)	84%	78%	94%	76%	78%	98%
27	(10, 2, 0)	(40, 40, 40)	78%	72%	86%	80%	68%	92%
28	(10, 1, 0)	(40, 40, 40)	46%	38%	54%	34%	24%	46%
29	(10, 5, 2, 0)	(30, 30, 30, 30)	64%	68%	92%	58%	56%	88%
30	(10, 5, 1, 0)	(30, 30, 30, 30)	50%	38%	42%	32%	16%	32%
31	(10, 2, 0)	(50, 40, 30)	90%	64%	86%	82%	62%	88%
32	(10, 5, 2, 0)	(50, 35, 25, 10)	68%	76%	66%	58%	52%	62%
33	(10, 6, 4, 2, 0)	(40, 30, 25, 15, 10)	18%	46%	56%	14%	22%	46%
34	(5, 0, 5)	(40, 40, 40)	92%	84%	98%	86%	76%	98%
35	(3, 0, 3)	(40, 40, 40)	78%	80%	86%	74%	62%	88%
36	(2, 0, 2)	(40, 40, 40)	80%	60%	68%	60%	52%	74%
37	(5, 0, 5)	(20,60,40)	80%	82%	100%	82%	84%	98%
38	(2, 0, 2)	(20,60,40)	64%	64%	76%	56%	56%	84%
39	(5, 2, 0, 2, 4, 6)	(20,30,20,30,20,20)	34%	32%	32%	22%	28%	30%
			51.92%	**64.83%**	**75.75%**	**45.83%**	**51.5%**	**75%**

4. 4 The influence of different scales

In this main case, we expand the no shift data to see the result, and we take the data from the Case 1.

Table 7 Result5 of scales changed

		No adding outlier	Adding outliers
Case	data	New	New
1	*1	100%	100%
40	*1.5	100%	100%
41	*2	100%	100%
42	*5	100%	100%
43	*10	100%	100%
44	*20	100%	100%
		100%	**100%**

In this main case, we contract the shifted data to see the result, and we take the data from Case 6. The location of the mean-shift in the simulation result and the setting are within 3.

Table 8 Result6 of scales changed

		No adding outlier	Adding outliers
Case	data	New	New
6	*1	100%	98 %
45	*0.5	100%	98 %
46	*0.2	100%	98 %
47	*0.1	100%	98 %
48	*0.05	100%	98 %
49	*0.01	100%	98 %
		100%	**98 %**

Chapter 5: Conclusions

In this paper, we used a regression tree to set up the model and apply it in the semiconductor industry to detect where the yield rate changed. From chapter 4, we can find that our new method is efficient in detecting the mean-shift problem, and we can conclude that

1. The new method has the highest simulated classification rate among these three methods.
2. When adding some outliers, the result of the simulation is less affected.
3. For the different scales, the result of the simulation is also less affected.
4. For the unbalanced data, the result of the simulation is also less affected.
5. The new method must reduce the time.

Therefore we may say that the new method is a useful application in the semiconductor industry to determine whether in detecting yield rates helps to find where the process has variation.

The new method may be applied in other fields. For example:

1. We can use the new method to set up a model for classification.
2. We can also use the new method to solve problems that used CPD before.

In future work:

1. Because this article solves problem of the detecting semiconductor yield rates, so we just discuss one dimension by this method. Therefore

we may study the new method to apply to other problems with high dimensions in this way.

2. In this paper, our condition is laid in normal distribution with unchanged variance; therefore, in the future, we may discuss the classification of random assignment material. If the variance has changed, we need to discuss how to detect the position of its changed variance.

3. Because of the semiconductor industry's demand, the size of its detection rate material is approximately 50-200. Therefore in this article, we mainly simulate small samples. It is adaptable to have a sample size larger than 50 in this new method.

References

[1] Abu-Taleb, A.A., Alawneh, A.J., and Smadi, M.M., Statistical analysis of recent changes in relative humidity in Jordan. American Journal of Environmental Sciences 3 (2), 2007, 75-77.

[2] Bergeret, F. and Le Gall, C., Yield Improvement using Statistical Analysis of Process Dates, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 3, 2003, 535-542.

[3] Besse P., Le Gall, C., Application and reliability of change-point analyses for detecting a defective in intragated circut manufacturing series, Communication in Statistics, Simulation and Computation , 2006.

[4] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., Classification and Regression Trees, Wadsworth, Belmont, California, 1984.

[5] Carslaw, D.C., Ropkins, K., and Bell, M.C., Change-Point Detection of Gaseous and Particulate Traffic-Related Pollutants at a Roadside Location, Environmental Science and Technology, Vol. 40. Issue 22, 2006, 6912-6918.

[6] Esposito, F., Malerba, D., and Semeraro, G., A Comparative Analysis of Methods for Pruning Decision Trees, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 19, NO. 5, 1997, 476-491

[7] Esposito, F., Malerba, D. and Semeraro, G., A Comparative Analysis of Methods for Pruning Decision Trees, IEEE Transactions on Pattern Analysis and Machine

Intelligence, Vol. 19, No. 5, May 1997, 476-491.

[8] Kucera, J., Barbosa, P., Strobl, P., Cumulative sum charts - A novel technique for processing daily time series of MODIS data for burnt area mapping in Portugal, IEEE Proceedings Multitemp2007, Leuven, Belgium.

[9] Lai, T.L. Sequential change point detection in quality control and dynamical systems, J. Royal Statistical Society Soc. Ser. B **57**, 1995, 613-658.

[10] Lavielle, M., Optimal segmentation of random processes, IEEE Transactions on signal processing 46, May 1998, 1365-1373.

[11] Neretti, N., Remondini, D., Tatar, M., Sedivy, J.M., Pierini, M., Mazzatti, D., Powell, J., Franceschi, C., and Castellani, G.C., Correlation analysis reveals the emergence of coherence in the gene expression dynamics following system perturbation, BMC Bioinformatics 2007, 8(Suppl 1):S16 (8 March 2007).

[12] Neter, J., Wasserman, W., and Kutner, M.H., Applied Linear Regression Models, Second Edition, Richard D. Irwin. Inc., Boston, Massachusetts, 1989.

[13] Smadi, M.M., Zghoul, A.A., A Sudden Change In Rainfall Characteristics In Amman, Jordan During The Mid 1950s, American Journal of Environmental Sciences 2 (3), 2006, 84-91.

[14] Taylor, W.A. Change-Point analysis: A powerful new tool for detecting changes, 2000. http://www.variation.com/cpa/tech/changepoint.html

[15] Williams, D., Kuhn, A., Kupsch, A., Tijssen, M., van Bruggen, G., Speelman, H., Hotton, G., Yarrow, K., and Brown, P., Behavioural cues are associated with modulations of synchronous oscillations in the human subthalamic nucleus. Brain, September 1, 2003; 126(9): 1975-1985.

[16] Windeatt T., Ardeshir G., An empirical comparison of pruning methods for ensemble classifiers, Proc. of Int. Conf Intelligent Data Analysis, Sept 13-15, 2001, Lisbon, Portugal, Lecture notes in computer science, Springer-Verlag, 208-217

Appendix 4.1

Table 9 Result2 of no shift

Result 2: *The location of the mean-shift* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
1	0	30	94%	52%	100%	96%	62%	100%
2	0	50	94%	68%	100%	90%	70%	100%
3	0	100	94%	90%	100%	92%	84%	100%
4	0	200	98%	96%	100%	98%	92%	100%
			95%	**76.5%**	**100%**	**94%**	**77%**	**100%**

Table 10 Result2 of mean-shift one time

Result 2: *The location of the mean-shift* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
5	(5, 0)	(50, 50)	92%	88%	96%	88%	80%	92%
6	(3, 0)	(50, 50)	66%	64%	76%	60%	58%	76%
7	(2, 0)	(50, 50)	60%	58%	66%	56%	54%	66%
8	(1, 0)	(50, 50)	24%	16%	22%	20%	16%	22%
9	(5, 0)	(90, 10)	58%	84%	98%	42%	86%	90%
10	(3, 0)	(90, 10)	30%	66%	80%	14%	62%	72%
11	(2, 0)	(90, 10)	22%	66%	62%	10%	30%	50%
12	(1, 0)	(90, 10)	4%	16%	20%	2%	6%	4%
13	(2, 0)	(15, 15)	56%	30%	52%	20%	14%	14%
14	(2, 0)	(20, 10)	58%	40%	60%	18%	10%	8%
15	(2, 0)	(25, 25)	54%	40%	62%	42%	30%	48%
			47.64%	**51.64%**	**63.09%**	**33.82%**	**40.55%**	**49.27%**

Table 11 Result2 of mean-shift several times

Result 2: *The location of the mean-shift* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	Mean	size	CPD	Cvtree	New	CPD	Cvtree	New
16	(6, 3, 0)	(40, 40, 40)	6%	56%	60%	12%	44%	52%
17	(15, 10, 5, 0)	(30, 30, 30, 30)	82%	80%	94%	58%	52%	68%
18	(9, 6, 3, 0)	(30, 30, 30, 30)	60%	54%	68%	44%	34%	60%
19	(6, 4, 2, 0)	(30, 30, 30, 30)	20%	16%	28%	16%	12%	22%
20	(20, 15, 10, 5, 0)	(25, 25, 25, 25, 25)	6%	78%	78%	0%	46%	74%
21	(12, 9, 6, 3, 0)	(25, 25, 25, 25, 25)	4%	26%	30%	0%	26%	28%
22	(8, 6, 4, 2, 0)	(25, 25, 25, 25, 25)	0%	8%	8%	0%	2%	6%
23	(25,20,15,10,5, 0)	(20, 20, 20, 20, 20, 20)	4%	72%	70%	2%	40%	48%
24	(15, 12, 9, 6, 3, 0)	(20,20,20,20,20,20)	4%	24%	24%	2%	8%	22%
25	(10, 8, 6, 4, 2, 0)	(20, 20, 20, 20, 20, 20)	0%	4%	4%	0%	0%	4%
26	(10, 3, 0)	(40, 40, 40)	64%	72%	90%	54%	68%	86%
27	(10, 2, 0)	(40, 40, 40)	64%	56%	74%	56%	46%	66%
28	(10, 1, 0)	(40, 40, 40)	22%	16%	22%	16%	8%	20%
29	(10, 5, 2, 0)	(30, 30, 30, 30)	28%	28%	38%	34%	24%	44%
30	(10, 5, 1, 0)	(30, 30, 30, 30)	14%	8%	12%	6%	0%	6%
31	(10, 2, 0)	(50, 40, 30)	66%	46%	64%	52%	34%	58%
32	(10, 5, 2, 0)	(50, 35, 25, 10)	38%	46%	54%	26%	26%	28%
33	(10, 6, 4, 2, 0)	(40, 30, 25, 15, 10)	6%	16%	20%	2%	6%	20%
34	(5, 0, 5)	(40, 40, 40)	86%	70%	84%	78%	58%	78%
35	(3, 0, 3)	(40, 40, 40)	60%	54%	60%	50%	44%	56%
36	(2, 0, 2)	(40, 40, 40)	38%	24%	28%	30%	20%	28%
37	(5, 0, 5)	(20,60,40)	72%	82%	100%	64%	76%	90%
38	(2, 0, 2)	(20,60,40)	22%	38%	42%	16%	24%	38%
39	(5, 2, 0, 2, 4, 6)	(20,30,20,30,20,20)	4%	6%	6%	0%	2%	4%
			32.08%	**40.83%**	**48.25%**	**25.75%**	**29.17%**	**41.92%**

Appendix 4.2

Table 12 Result4 of no shift

Result 4: *The location of the mean-shift* in the simulation result is same as the setting.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
1	0	30	94%	52%	100%	96%	62%	100%
2	0	50	94%	68%	100%	90%	70%	100%
3	0	100	94%	90%	100%	92%	84%	100%
4	0	200	98%	96%	100%	98%	92%	100%
			95%	**76.5%**	**100%**	**94%**	**77%**	**100%**

Table 13 Result4 of mean-shift one time

Result4: *The location of the mean-shift* in the simulation result and the setting are close, within 5.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
5	(5, 0)	(50, 50)	94%	90%	98%	94%	86%	100%
6	(3, 0)	(50, 50)	86%	84%	100%	74%	76%	100%
7	(2, 0)	(50, 50)	84%	82%	100%	86%	80%	100%
8	(1, 0)	(50, 50)	74%	62%	68%	54%	38%	58%
9	(5, 0)	(90, 10)	88%	84%	100%	68%	92%	100%
10	(3, 0)	(90, 10)	60%	86%	100%	36%	84%	96%
11	(2, 0)	(90, 10)	38%	94%	92%	16%	54%	74%
12	(1, 0)	(90, 10)	6%	26%	28%	6%	14%	8%
13	(2, 0)	(15, 15)	96%	62%	90%	44%	32%	24%
14	(2, 0)	(20, 10)	86%	62%	82%	30%	34%	8%
15	(2, 0)	(25, 25)	92%	68%	100%	82%	50%	90%
			73.09%	**72.73%**	**87.09%**	**53.64%**	**58.18%**	**68.91%**

Table 14 Result4 of mean-shift several times

Result 4: *The location of the -shift* in the simulation result and the setting are close, within 5.								
			No adding outlier			Adding outliers		
Case	mean	size	CPD	Cvtree	New	CPD	Cvtree	New
16	(6, 3, 0)	(40, 40, 40)	36%	84%	92%	38%	82%	94%
17	(15, 10, 5, 0)	(30, 30, 30, 30)	86%	82%	98%	82%	66%	96%
18	(9, 6, 3, 0)	(30, 30, 30, 30)	88%	84%	100%	70%	62%	98%
19	(6, 4, 2, 0)	(30, 30, 30, 30)	78%	66%	98%	70%	52%	88%
20	(20, 15, 10, 5, 0)	(25, 25, 25, 25, 25)	14%	100%	100%	14%	70%	100%
21	(12, 9, 6, 3, 0)	(25, 25, 25, 25, 25)	20%	60%	86%	24%	72%	86%
22	(8, 6, 4, 2, 0)	(25, 25, 25, 25, 25)	14%	46%	56%	8%	28%	62%
23	(25,20,15,10,5,0)	(20,20,20,20,20,20)	8%	96%	88%	20%	74%	96%
24	(15,12,9,6,3,0)	(20,20,20,20,20,20)	24%	82%	90%	28%	50%	98%
25	(10,8,6,4,2,0)	(20,20,20,20,20,20)	26%	46%	68%	20%	26%	68%
26	(10, 3, 0)	(40, 40, 40)	86%	78%	98%	80%	84%	100%
27	(10, 2, 0)	(40, 40, 40)	80%	74%	94%	84%	70%	96%
28	(10, 1, 0)	(40, 40, 40)	68%	54%	72%	52%	38%	64%
29	(10, 5, 2, 0)	(30, 30, 30, 30)	72%	72%	98%	70%	64%	94%
30	(10, 5, 1, 0)	(30, 30, 30, 30)	56%	46%	52%	34%	16%	46%
31	(10, 2, 0)	(50, 40, 30)	92%	68%	96%	84%	68%	94%
32	(10, 5, 2, 0)	(50, 35, 25, 10)	72%	78%	94%	66%	58%	70%
33	(10, 6, 4, 2, 0)	(40, 30, 25, 15, 10)	28%	56%	68%	24%	32%	60%
34	(5, 0, 5)	(40, 40, 40)	92%	84%	98%	94%	80%	98%
35	(3, 0, 3)	(40, 40, 40)	88%	88%	100%	82%	68%	98%
36	(2, 0, 2)	(40, 40, 40)	88%	64%	78%	64%	54%	80%
37	(5, 0, 5)	(20,60,40)	84%	82%	100%	86%	86%	100%
38	(2, 0, 2)	(20,60,40)	72%	68%	84%	66%	60%	86%
39	(5, 2, 0, 2, 4, 6)	(20,30,20,30,20,20)	48%	44%	58%	32%	42%	54%
			59.17%	**70.92%**	**86.08%**	**53.83%**	**58.42%**	**84.42%**