SYMMETRIC AND TRIMMED SOLUTIONS OF SIMPLE LINEAR REGRESSION

by

Chao Cheng

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(STATISTICS)

December 2006

Dedication

to my grandmother

to my parents

Acknowledgments

First and foremost, I would like to thank my advisor, Dr. Lei Li, for his patience and guidance during my graduate study. Without his constant support and encouragement, the completion of the thesis would not be possible. I feel privileged to have had the opportunity to work closely and learn from him.

I am very grateful to Dr. Larry Goldstein and Dr. Jianfeng Zhang, in the Department of Mathematics, for their most valuable help in writing this thesis. Most of my knowledge about linear regression is learned from Dr. Larry Goldstein's classes.

I would also like to thank Dr. Xiaotu Ma, Dr. Rui Jiang, Dr. Ming Li, Huanying Ge, and all the past and current members in our research group for their suggestions and recommendations. I greatly benefit from the discussion and collaboration with them.

Finally, I would like to thank my little sister and my parents for their love, support and encouragement. I would also like to thank my grandmother, who passed away in 2002, soon after I began my study in USC. I feel sorry that I did not accompany her in her last minute. This thesis is dedicated to her.

Table of Contents

List of Tables

List of Figures

Abstract

Least trimmed squares (LTS), as a robust method, is widely used in linear regression models. However, the ordinary LTS of simple linear regression treats the response and prediction variable asymmetrically. In other world, it only considers the errors from response variable. This treatment is not appropriate in some applications. To overcome these problems, we develop three versions of symmetric and trimmed solutions that take into consideration errors from both response and predictor variable.

In the thesis, we describe the algorithms to achieve the exact solutions for these three symmetric LTS. We show that these methods lead to more sensible solutions than ordinary LTS. We also apply one of the methods to the microarray normalization problem. It turns out that our LTS based normalization method has advantages over other available normalization methods for microarray data set with large differentiation fractions.

Chapter 1

INTRODUCTION

1.1 Simple linear regression model

Regression is a methodology for studying relations between variables, where the relations are approximated by functions. In the late 1880s, Francis Galton was studying the inheritance of physical characteristics. In particular, he wondered if he could predict a boy's adult height based on the height of his father. Galton hypothesized that the taller the father, the taller the son would be. He plotted the heights of fathers and the heights of their sons for a number of father-son pairs, then tried to fit a straight line through the data (Figure 1.1). If we denote the son's height by y and the father's height by x, the relation between y and x can be described by a simple linear model:

$$Y = \alpha + \beta X + \varepsilon \tag{1.1}$$

In the simple linear regression model, y is called dependent variable, x is called predictor variable, and ε is called prediction error or residual. The symbols α and β are called regression parameters or coefficients.

Simple linear regression model is a special case of multiple linear regression model, which involves in more than one predictor variables. i.e., multiple linear regression model describes the relation of response variable y with a number of predictor variables, say $x_1, x_2, ..., x_p$. The model can be formulated as the following:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon \tag{1.2}$$

In this thesis, we are interest in the relation between two variables x and y, and therefore will focus on simple linear regression model.

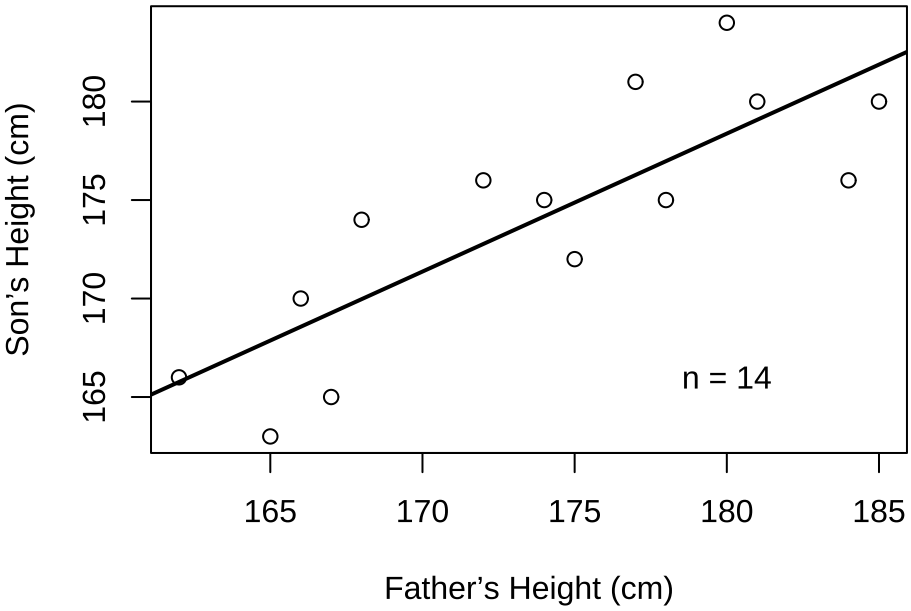


Figure 1.1: Hight of fathers and sons for 14 father-son pairs. The strait line is the least square fitted line.

1.2 Least square fitting

Once a model has been specified, the next task in regression analysis is to fit the model to the data. For the model (1.1) the task is to find specific values for α and β, to represent the data as well as possible. As long as α and β have been specified, a specific straight line can be determined to represent the relation between x and y as shown in Figure 1.1. By far the most important and popular method for doing this is what so called least square fitting. It aims to find values for α and β that minimize the sum of squares of the vertical deviations of the data points (x, y) from the fitted straight line. Namely,

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

The minimizing values $\hat{\alpha}$ and $\hat{\beta}$ are called least squares estimates of α and β or least squares regression coefficients. The corresponding line is called the least square regression line. The $\hat{y} \triangleq \hat{\alpha} + \hat{\beta}x$ represents the fitted value calculated from the regression.

Let's define $F(\alpha, \beta) = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$, then we can obtain least squares regression coefficients $\hat{\alpha}$ and $\hat{\beta}$ by solving the following equations:

$$\left\{ \begin{array}{rcl} \dfrac{\partial\, F(\alpha, \beta)}{\partial\, \alpha} & = & 0 \\ \dfrac{\partial\, F(\alpha, \beta)}{\partial\, \beta} & = & 0 \end{array} \right. \tag{1.3}$$

The solution is:

$$\left\{ \begin{array}{rcl} \hat{\beta} & = & \dfrac{S_{xy}}{S_{xx}} \\ \hat{\alpha} & = & \bar{y} - \hat{\beta}\bar{x} \end{array} \right. \tag{1.4}$$

where $S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$ and $S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$.

It should be note that the regression model itself and the least square fitting is not symmetric with respect to the predictor variable x and the response variable y. In the least square fitting, it is the vertical, and not the horizontal deviations that are minimized. The horizontal deviations should be minimized if we are interested in estimating the relation of x to y rather than that of y to x. The asymmetry can be rationalized if the model is used to make prediction. If we are interested in predicting y from x using a linear relation, we should minimize the vertical deviations $\sum (y_i - \hat{y}_i)^2$. On the other hand, if we want to predict x from y, we should minimize the horizontal deviations $\sum (x_i - \hat{x}_i)^2$ and consistently the model $X = \alpha + \beta Y + \varepsilon$ should be used.
Sometimes we are interested in estimating the linear relation between x and y rather than making prediction of one variable from the other. In these cases, x and y are of equal importance and should be treated symmetrically. Thereby, the least square fitting is not appropriate to be used any more, because it consider only the errors from the so-called response variable and the errors from predictor variable are completely ignored. As a matter of fact, in the canonical simple linear regression model given by model (1.1), only the response variable y is considered as a random variable, x is regarded as a constant variable. For the above reasons, the least square regression line estimated using x as predictor are distinct with the one using y as predictor. This is unreasonable when x and y are of the same importance and may lead to troubles in practise. We will investigate this problem in details in latter part.

1.3 Robust regression

One can prove that least squares estimates are the most efficient unbiased estimates of the regression coefficients under the assumption that the errors are normally distributed. However, they are not robust to the outliers that are present in the data. To obtain sensible estimates of the regression coefficients in such circumstances, robust regression that are not sensitive to outliers should be used.

The sensitivity of least squares to outliers is due to two factors [SL03].First, when the squared residual is used to measure size, any residue with a large magnitude will have a very large size relative to the others. Second, by using a measure of location such as mean that is not robust, any large square will have a very strong impact on the criterion, which will result in the extreme data point having a disproportionate influence on the regression.

Consistently, we may have two strategies to achieve robust regression. First, we can replace the square ε^2 by some other function $\rho(\varepsilon)$, which reflects the size of the residual in a less extreme way. The idea lead to the notion of M estimation [jH73],S estimation [RY84], MM estimation [Yoh87] and least absolute value (LAV) regression [STA01]. As an example, in LAV regression, the regression coefficients are obtained by minimize $\sum_{i=1}^{n} |y_i - \alpha - \beta x_i|$. Second, we can replace the sum by a more robust measure of location such as the median or a trimmed mean, which lead to least median of squares estimates (LMS) [Rou84, VR99] and least trimmed squares estimates (LTS) [RL99, RD99], respectively. Weighted least squares (WLS) is another approach that follow this strategy [CR88]. The idea is to assign smaller weights for observations with larger residuals. In the thesis, we will propose a new algorithm to calculate exact solutions of a revised version of LTS, which takes into account errors from both predictor and response variables. Thus, we will introduce LTS in the next section.

1.4 Least trimmed squares (LTS)

LTS is first proposed by Rousseeuw in 1984 [Rou84]. The breakdown values is often used to measure the robustness of a method. It is defined as the proportion of contamination (outliers) that a procedure can withstand and still achieve accurate estimates. The breakdown value of ordinary least squares fitting is $\frac{1}{n}$, which means one outlier in the observations may cause incorrect estimation of the regression coefficients. The LTS is widely used for its high breakdown value and other good properties.

We denote the squared residuals in the ascending order by $|r^2(\alpha, \beta)|_{(1)} \leq |r^2(\alpha, \beta)|_{(2)} \leq \cdots |r^2(\alpha, \beta)|_{(n)}$, where $r(\alpha, \beta) = y - \alpha - \beta x$ Then the LTS estimate of coverage h is obtained by

$$\arg\min_{\alpha,\beta} \sum_{i=1}^{h} |r^2(\alpha, \beta)|_{(i)} .$$

This definition implies that observations with the largest residuals will not affect the estimate. The LTS estimator is regression, scale, and affine equivariant; see Lemma 3, Chapter 3 in Rousseeuw and Leroy [RL99]. In terms of robustness, we can roughly achieve a breakdown point of ρ by setting $h = [n(1 - \rho)] + 1$. In terms of efficiency, $\sqrt{n}$-consistency and asymptotic normality similar to M-estimator exist for LTS under some conditions; see Víšek for example [V9́6, V0́0]. Despite its good properties, the computation of LTS remains a problem.

The problem of computing the LTS estimate of α and β is equivalent to searching for the size-h subset(s) whose least squares solution achieves the minimum of trimmed squares. The total number of size-h subsets in a sample of size n is $\binom{n}{h}$. A full search through all size-h subsets is impossible unless the sample size is small. Several ideas have been proposed to compute approximate solutions. First, instead of exhaustive search we can randomly sample size-h subsets. Second, Rousseeuw and

Van Driessen [RD99] proposed a so-called C-step technique (C stands for "concentration"). That is, having selected a size-h subset, we apply the least squares estimator to them. Next for the estimated regression coefficients, we evaluate residuals for all observations. Then a new size-h subset with the smallest squared residuals is selected. This step can be iterated starting from any subset. In the case of estimating a location parameter, Rousseeuw and Leroy [RL99], described a procedure to compute the exact LTS solution. Rousseeuw and Van Driessen [RD99] applied this idea to adjust the intercept in the regression case. The idea of subset search is also relevant to LMS; see Hawkins [Haw93]. Hössjer described an algorithm for computing the exact least trimmed squares estimate of simple linear regression, which requires $O(n^3 \log n)$ computations and $O(n^2)$ storage [H9̈5]. A refined algorithm, which requires $O(n^2 \log n)$ computations and $O(n)$ storage, was also sketched. However, Hössjer remarked that the refined algorithm is not stable.

Li introduced an algorithm that computes the exact solution of LTS for simple linear regression with constraints on the slope [Li05]. The idea is to divide the range of slope into regions in such a way that within each region the order of residuals is unchanged. As a result, the search for LTS within each such region becomes a problem of linear complexity. Without constraints, the overall complexity of the algorithm is $O(n^2 \log n)$ in time and $O(n^2)$ in storage. In practical regression problems, constraints are often imposed on slopes, which will reduce computing load substantially. Like the canonical least squares fitting, this algorithm is asymmetric with respect to predictor and response variable in a way that only errors from response variable are considered. In practice, we are often need to study the relation between two variables that are of equal importance and errors from both should be taken into account. In this thesis we will introduce three algorithms which are developed from Li's method. The new methods take into account errors from both predictor variable x and response variable y by considering both of them as random variables.

Chapter 2

Model and algorithm

2.1 Bivariate least squares (BLS)

Let us consider a error-in-variables version of simple linear regression model which regards both of the two variables as random. Suppose we have n independent observations $(x_i, y_i), i = 1, ..., n$, and the underlying linear relation between x and y is

$$\mu_{y_i} = \alpha + \beta\mu_{x_i}, \;\; i = 1, 2, ..., n.$$

In each observation, we have:

$$\begin{cases} x_i &= \mu_{x_i} + \varepsilon_{x_i}, \\ y_i &= \mu_{y_i} + \varepsilon_{y_i}. \end{cases} \tag{2.1}$$

We assume equal precisions of measurement for both x_i and y_i at all observations, i.e. $\varepsilon_{x_1}, \varepsilon_{x_2}, \cdots, \varepsilon_{x_n}$ and $\varepsilon_{y_1}, \varepsilon_{y_2}, \cdots, \varepsilon_{y_n}$ are independently and identically distributed. To calculate the regression coefficient estimates, the ordinary least squares (OLS) fitting minimize the following target function (Figure 2.1A):

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

To simplify the formulas in later sections, we define:

$$\begin{aligned} S_{xx} &= \textstyle\sum_{i=1}^{n}(x_i-\bar{x})^2, \\ S_{yy} &= \textstyle\sum_{i=1}^{n}(y_i-\bar{y})^2, \\ S_{xy} &= \textstyle\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}), \end{aligned}$$

The solution for OLS is:

$$\begin{cases} \hat{\beta} = S_{xy}/S_{xx}, \\ \hat{\alpha} = \bar{y}-\hat{\beta}\bar{x}. \end{cases}$$

In order to take into account errors from both x and y, we need construct a target function that is symmetric with respect to x and y. Three possible target functions can be used to achieve such a purpose:

(1) $(\hat{\alpha},\hat{\beta}) = \arg\min_{\alpha,\beta}\sum_{i=1}^{n}|(y_i-\hat{y_i})(x_i-\hat{x_i})|$
$= \arg\min_{\alpha,\beta}\sum_{i=1}^{n}\dfrac{(y_i-\alpha-\beta x_i)^2}{|\beta|}$

(2) $(\hat{\alpha},\hat{\beta}) = \arg\min_{\alpha,\beta}\sum_{i=1}^{n}[(y_i-\hat{y_i})^2+(x_i-\hat{x_i}^2)]$
$= \arg\min_{\alpha,\beta}\sum_{i=1}^{n}(1+\frac{1}{\beta^2})(y_i-\alpha-\beta x_i)^2$

(3) $(\hat{\alpha},\hat{\beta}) = \arg\min_{\alpha,\beta}\sum_{i=1}^{n}[(y_i-\hat{y_i})^2\cos^2(\arctan\beta)]$
$= \arg\min_{\alpha,\beta}\sum_{i=1}^{n}\dfrac{(y_i-\alpha-\beta x_i)^2}{1+\beta^2}$

In (1), we aims to minimize the sum of the areas circled by (x_i, y_i), $(\hat{x_i}, y_i)$ and $(x_i, \widehat{y_i})$: $\sum|(y_i-\hat{y_i})(x_i-\hat{x_i})|$, so we define the method as Bivariate Multiplicative

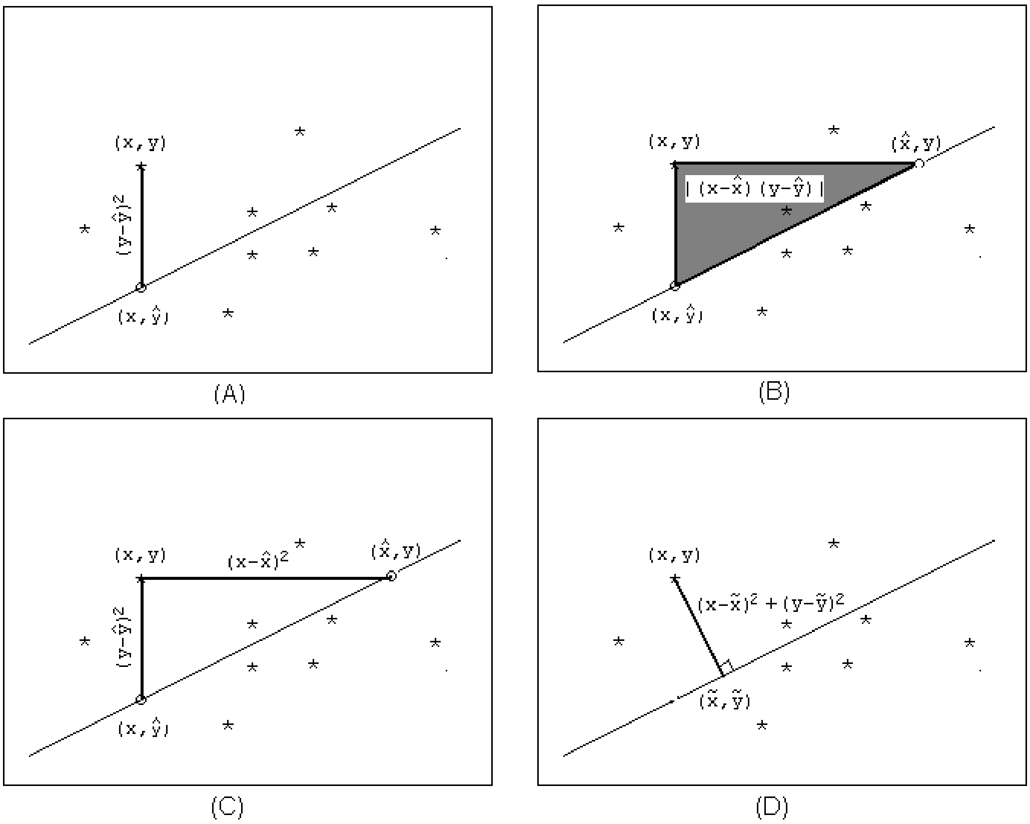


Figure 2.1: Illustration of BLS. (A) Ordinary Least Squares (OLS) minimize the sum of vertical distances from (x_i, y_i) to $(x_i, \hat{y_i})$: $\sum(y_i - \hat{y_i})^2$. (B)Bivariate Multiplicative Least Squares (BMLS) minimize the sum of the areas circled by (x_i, y_i), $(\hat{x_i}, y_i)$ and $(x_i, \hat{y_i})$: $\sum|(y_i - \hat{y_i})(x_i - \hat{x_i})|$. (C)Bivariate Symmetric Least Squares (BSLS) minimize the sum of summation of vertical distances from (x_i, y_i) to $(x_i, \hat{y_i})$ and horizontal distances from (x_i, y_i) to $(\hat{x_i}, y_i)$: $\sum[(y_i - \hat{y_i})^2 + (x_i - \hat{x_i}^2)]$. (D)Bivariate Perpendicular Least Squares (BPLS) minimize the the sum of perpendicular distances from (x_i, y_i) to $(\tilde{x_i}, \tilde{y_i})$: $\sum(y_i - \hat{y_i})^2 \cos^2(\arctan\beta)$.

Least Squares (BMLS), see Figure 2.1B. In (2), we aims to minimize the sum of summation of vertical distances from (x_i, y_i) to $(x_i, \hat{y_i})$ and horizontal distances from (x_i, y_i) to $(\hat{x_i}, y_i)$, so we define the method as Bivariate Symmetric Least Squares (BSLS), see Figure 2.1C. In (3), we aims to minimize the the sum of perpendicular distances from

(x_i, y_i) to $(\tilde{x_i}, \tilde{y_i})$: $\sum (y_i - \hat{y_i})^2 \cos^2(\arctan \beta)$, so we define the method as Bivariate Perpendicular Least Squares (BPLS), see Figure 2.1D.

We note that BPLS has been introduced previously as Total Least Squares (TLS) together with Error-In-variables model (EIV) [vHL02]. A similar method to BSLS has also been proposed which minimize the following equation with respect to α and β.

$$\chi^2(\alpha, \beta) = \sum_{i=1}^{N} \frac{(y_i - \alpha - \beta x_i)^2}{\sigma_{y_i}^2 + \beta^2 \sigma_{x_i}^2}$$

where σ_{x_i} and σ_{y_i} are, respectively, the x and y standard deviations for the *i*th point [WBS99]. In this thesis, we would derive several algorithms that are robust to outliers to solve linear regression problems based on these Bivariate Least Squares (BLS) methods.

2.2 Solutions of BLS

In this section, we will use the Bivariate Perpendicular Least Squares (BPLS) as example to show how to solve the three types of Bivariate Least Square fitting. Let us define the target function for BPLS as:

$$F(\alpha, \beta) = \sum_{i=1}^{n} \frac{(y_i - \alpha - \beta x_i)^2}{1 + \beta^2}$$

To achieve the least square estimates $\hat{\alpha}$ and $\hat{\beta}$, we aim to find the $(\hat{\alpha}, \hat{\beta})$ that minimize the function, namely:

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha, \beta} \sum_{i=1}^{n} \frac{(y_i - \alpha - \beta x_i)^2}{1 + \beta^2}$$

By solving the following equation:

$$\begin{cases} \dfrac{\partial\, F(\alpha,\beta)}{\partial\, \alpha} &= 0 \\ \dfrac{\partial\, F(\alpha,\beta)}{\partial\, \beta} &= 0 \end{cases}$$

we obtain two solutions:

$$\begin{cases} \hat{\beta} &= -B \pm \sqrt{B^2+1} \\ \\ \hat{\alpha} &= \bar{y} - \hat{\beta}\bar{x} \end{cases}$$

where $B = \dfrac{1}{2}\dfrac{[\sum_{i=1}^{n} x_i^2 - n\bar{x}^2] - [\sum_{i=1}^{n} y_i^2 - n\bar{y}^2]}{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}} = \dfrac{1}{2}\dfrac{S_{xx} - S_{yy}}{S_{xy}}$.

Then we calculate the second order derivatives: $A(\alpha,\beta) \triangleq \dfrac{\partial^2\, F(\alpha,\beta)}{\partial\, \alpha^2}$, $U(\alpha,\beta) \triangleq \dfrac{\partial^2\, F(\alpha,\beta)}{\partial\, \beta^2}$, and $C(\alpha,\beta) \triangleq \dfrac{\partial^2\, F(\alpha,\beta)}{\partial\, \alpha\, \partial\, \beta}$. As we know, the solution $(\hat{\alpha},\hat{\beta})$ that achieve the minimum of the target function has to satisfy the following inequations: $A > 0$, $C > 0$ and $U^2 - AC < 0$. After calculation, we can prove that only one of the two solutions achieve the minimum of the target function [Wei]:

$$\begin{cases} \hat{\beta} &= -B + \sqrt{B^2+1}, \quad if S_{xy} \geq 0 \\ \\ \hat{\beta} &= -B - \sqrt{B^2+1}, \quad if S_{xy} < 0 \end{cases}$$

.

Above is the solution for BPLS, similarly we obtain the solutions for BMLS and BSLS. For the BMLS we have:

$$\begin{cases} \hat{\beta} & = & \pm\sqrt{S_{yy}/S_{xx}}, \quad if S_{xy} \geq 0, take''+''; \; if S_{xy} < 0, take''-'', \\ \hat{\alpha} & = & \bar{y} - \hat{\beta}\bar{x}. \end{cases}$$

For the BSLS we have $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$ and $\hat{\beta}$ can be obtained by solving the following quartic equation: $\hat{\beta}^4 S_{xx} - \hat{\beta}^3 S_{xy} + \hat{\beta} S_{xy} - S_{yy} = 0$. It is hard to derive the formula solutions for above quartic equation. But we can obtain the numeric solutions by using various method such as Ferrari's method [Qua]. Generally, the quartic equation has four roots, including real roots and imaginary roots. The solution for $\hat{\beta}$ is the real root that minimize the target function.

2.3 Bivariate Least Trimmed Squares (BLTS)

Least Trimmed Squares (LTS) is one of the most frequently used robust regression method. Several algorithms have been introduced for LTS such as fast-LTS [RD99]. Recently Li et al proposed a new algorithm to compute exact solution of LTS for simple linear regression [Li05]. In this thesis, we take into account the experimental errors from both x and y by using the above introduced target functions (BMLS, BSLS and BPLS) for least squares estimation. Consistently, we designed three bivariate least trimmed squares algorithms, which are called BMLTS, BSLTS, and BPLTS, respectively. In the following section, we will describe the algorithm using BPLTS as example. The algorithms of BMLTS and BSLTS are similar to BPLTS.

2.4 Algorithm for BPLTS

We consider a simple linear regression problem with a response variable y and an explanatory variable x. Let (x_i, y_i), $i = 1, \cdots, n$, be data.

Proposition 1. *Consider a simple linear regression with restricted slope range, namely, we impose a constraint on the slope: $\beta \in [b_1, b_2]$. Denote the unrestricted least squares solution by $(\hat{\alpha}, \hat{\beta})$.*

- *If $\hat{\beta} \in [b_1, b_2]$, then the unrestricted solution is also the restricted solution.*
- *If $\hat{\beta} \notin [b_1, b_2]$, then the restricted solution of the slope is either b_1 or b_2.*

Proof: If $\hat{\beta} \in [b_1, b_2]$, then the unrestricted solution satisfies the requirement. Otherwise, according to the Kuhn-Tucker condition, the solution must be on the boundary; see Luenberger [Lue87] and Lawson and Hanson [LH74]. Stark and Parker provided another algorithm for the problem of bounded-variable least squares [SP95].

Hereafter we refer to any subset of the data by the set of their indices. For the sake of completeness and convenience, we write down the formulas of the least squares estimate for any size-h subset H.

$$\left\{\begin{array}{rcl} \overline{x}_H & = & \frac{1}{h}\sum_{i\in H} x_i \, , \\ \overline{y}_H & = & \frac{1}{h}\sum_{i\in H} y_i \, , \\ \overline{xx}_H & = & \frac{1}{h}\sum_{i\in H} x_i^2 \, , \\ \overline{yy}_H & = & \frac{1}{h}\sum_{i\in H} y_i^2 \, , \\ \overline{xy}_H & = & \frac{1}{h}\sum_{i\in H} x_i y_i \, , \\ \overline{cxy}_H & = & \overline{xy}_H - \overline{x}_H \overline{y}_H \, , \\ \hat{\beta}_H & = & \overline{cxy}_H / [\overline{xx}_H - \overline{x}_H^2] \, , \\ \hat{\alpha}_H & = & \overline{y}_H - \hat{\beta}_H \overline{x}_H \, , \\ SS_H & = & [\overline{yy}_H - \overline{y}_H^2] - \hat{\beta}_H \overline{cxy}_H \, . \end{array}\right. \tag{2.2}$$

The last quantity is the averaged sum of squares. $\overline{cxy}_H$ represents the centerized version of $\overline{xy}_H$. To proceed, we need some notation and definitions.

Definition 1. *For any (α, β),*

1. *Their residuals are defined by $r(\alpha, \beta)_i = y_i - (\alpha + \beta\, x_i)$;*

2. *$H_{(\alpha,\beta)}$ is a size-h index set that satisfies the following property: $|r(\alpha, \beta)_i| \leq |r(\alpha, \beta)_j|$, for any $i \in H_{(\alpha,\beta)}$ and $j \notin H_{(\alpha,\beta)}$.*

On the other hand, given any size-h subset H, we can find the least squares estimate (α_H, β_H) for the subsample $\{(x_i, y_i),\, i \in H\}$. Thus we have established a correspondence between the parameter space and the collection of all size-h subsets. This leads to our following definition of LTS. Please notice that we consider a general case with possible constraints on the slope. For convenience, hereafter in this section we use coverage h instead of a trimming fraction. The breakdown value for coverage h is $\frac{n-h}{n}$.

Definition 2. *For a given coverage h, the least trimmed squares estimate subject to $b_1 \leq \beta \leq b_2$, where $-\infty \leq b_1 < b_2 \leq \infty$, is defined by*

1. *(α, β) that minimizes $\sum_{i \in H_{(\alpha,\beta)}} r(\alpha, \beta)_i^2$ subject to $b_1 \leq \beta \leq b_2$;*
2. *or equivalently, the least squares estimate (α_H, β_H) of a size-h subset H that minimizes SS_H subject to $b_1 \leq \beta_H \leq b_2$; cf. (2.2).*

Namely, we can either search in the parameter space, or search in the space of size-h subsets. This dual form of the BPLTS solution is the key to the development of the algorithms in this paper.

Proposition 2. *For any parameter (α, β), we sort its residuals in the ascending order, namely, $r(\alpha, \beta)_{\pi(1)} \leq r(\alpha, \beta)_{\pi(2)} \leq \cdots \leq r(\alpha, \beta)_{\pi(n)}$, where $\{\pi(1), \pi(2), \cdots, \pi(n)\}$ is a permutation of $\{1, 2, \cdots, n\}$. Then $H_{(\alpha,\beta)}$ must be from the $n - h + 1$ size-h subsets: $\{\pi(k), \pi(k+1), \cdots, \pi(k+h-1)\}$, $k = 1, \cdots, n-h+1$.*

Proof: This is true because of the following fact: after taking absolute values, one or more residuals become the smallest; As moving away from the smallest, residuals increase monotonically leftwards or rightwards.

The significance of the result is that the complexity for the search of $H_{(\alpha,\beta)}$ reduces to $O(n)$ from $\binom{n}{h}$. We notice that for fixed β, the order of residuals keeps unchanged whatever intercept α is chosen. This leads to the following definition.

Definition 3. *For a fixed β, denote $r(\beta)_i = y_i - \beta\, x_i$. Let $r(\beta)_{(i)}$, $i = 1, \cdots, n$, be their ordered residuals. Their partial ordered averages and partial sums of squares are defined by:*

$$\left\{ \begin{array}{rcl} \bar{r}(\beta)_{(l:m)} & = & \frac{1}{m-l+1} \sum_{k=l}^{m} r(\beta)_{(k)}\,, \\ SS(\beta)_{(l:m)} & = & \frac{1}{m-l+1} \sum_{k=l}^{m} (r(\beta)_{(k)} - \bar{r}(\beta)_{(l:m)})^2,\ 1 \le l \le m \le n\,. \end{array} \right. \tag{2.3}$$

This structure and Proposition 2 are the basis of the following algorithm for computing the BPLTS estimate of a location parameter; see Page 171–172 in Rousseeuw and Leroy [RL99].

Definition 4. *For $\{y_i,\ i = 1, \cdots, n\}$, we define partial ordered averages and partial sums of squares respectively by*

$$\left\{ \begin{array}{rcl} \bar{y}_{(l:l+h-1)} & = & \frac{\sum_{k=l}^{l+h-1} y_{(k)}}{h}\,, \\ SS_{(l:l+h-1)} & = & \frac{\sum_{k=l}^{l+h-1} [y_{(k)} - \bar{y}_{(l:l+h-1)}]^2}{h}\,. \end{array} \right.$$

Corollary 1. *The BPLTS estimate of the location of $\{y_i,\ i = 1, \cdots, n\}$ is given by the partial ordered average(s) that achieves the smallest among $SS_{(l:l+h-1)}$, $l = 1, \cdots, n - h + 1$. In case there are ties, the solution is not unique.*

This case corresponds to $\beta = 0$ in the simple linear regression. Proposition 2 implies that we only need to check $n - h + 1$ partial ordered averages for searching the BPLTS estimate of the location. Partial ordered averages and sums of squares can be calculated in a recursive way; see Rousseeuw and Leroy (1987). Go back to the case of simple linear regression. The next result is the basis of our algorithms.

Proposition 3. *Suppose that for $\beta \in [b_1, b_2]$, the order of $\{r(\beta)_i\}$ keeps unchanged. Namely, there exists a permutation $\{\pi(1), \pi(2), \cdots, \pi(n)\}$ such that $r(\beta)_{\pi(1)} \leq r(\beta)_{\pi(2)} \leq \cdots \leq r(\beta)_{\pi(n)}$ for any $\beta \in [b_1, b_2]$.*

- *Then the size-h subset(s) for the BPLTS must be from the $n-h+1$ subsamples: $\{\pi(k), \pi(k+1), \cdots, \pi(k+h-1)\}$, where $k = 1, \cdots, n-h+1$.*
- *Moreover, for each subsample $\{\pi(k), \pi(k+1), \cdots, \pi(k+h-1)\}$, we compute its regular least squares estimate denoted by $\hat{\beta}_{(k:k+h-1)}$. Then the BPLTS estimate of β subject to the constraint $\beta \in [b_1, b_2]$ must be from the following: 1. $\{b_1, b_2\}$; 2. or $\{\hat{\beta}_{(k:k+h-1)}$, satisfying $b_1 < \hat{\beta}_{(k:k+h-1)} < b_2$, $k = 1, \cdots, n-h+1\}$.*

Proof: Let $(\hat{\alpha}, \hat{\beta})$ be one solution. According to the assumption, we have $r(\hat{\beta})_{\pi(1)} \leq r(\hat{\beta})_{\pi(2)} \leq \cdots \leq r(\hat{\beta})_{\pi(n)}$. Using the same argument leading to Proposition 2, we conclude that the size-h subset $H_{(\hat{\alpha},\hat{\beta})}$ must be from the $n-h+1$ size-h subsets: $\{\pi(k), \pi(k+1), \cdots, \pi(k+h-1)\}$, $k = 1, \cdots, n-h+1$. Then we apply Proposition 1 to the subset $H_{(\hat{\alpha},\hat{\beta})}$ and the proof is completed. According to this result, we only need to check the sum of squares for the $n-h+1$ candidates subject to the constraint.

Our next move is to divide the range of β into regions in such a way that within each region the order of residuals is unchanged and thus we can apply the above proposition. It is motivated by two technical observations. First we consider straight lines connecting each pair $\{(x_i, y_i), (x_j, y_j)\}$ satisfying $(x_i, y_i) \neq (x_j, y_j)$. The total number of such pairs is at most $\frac{n(n-1)}{2}$. The regions defined by these dividing lines satisfy the above order condition. Second, to compute the least squares estimates for the $n-h+1$ size-h subsets within each region, we do not have to repeatedly apply Formulas (2.2). Instead, from one region to the next, we only need to update estimates if a few residuals change their orders. The algorithm can be described as the following.

1. For any pair (i,j) such that $x_i \neq x_j$, compute $b^{(i,j)} = \frac{y_j - y_i}{x_j - x_i}$. Sort $\{b^{(i,j)}\}$ in the range $[b_1,\, b_2]$ and denote them by $b_1 < b^{[1]} \leq b^{[2]} \leq \cdots \leq b^{[L]} < b_2$, where $L \leq \frac{n(n-1)}{2}$. Save the pairs of indices corresponding to these slopes.

   Remark: Here we exclude all the slopes outside (b_1, b_2) when sorting slopes. We first restrict the scope of this algorithm to the case without slope ties, namely: $b_1 < b^{[1]} < b^{[2]} < \cdots < b^{[L]} < b_2$. To avoid slope ties, we perturb the original data by adding a tiny random number to each $x's$ and $y's$ before computing pairwise slope $b^{(i,j)}$. We have a complex version of algorithm which can deal with slope ties without perturbing the data [Li05]. In this paper, we only show the simple version considering its efficiency.

2. If $b_1 = -\infty$, go to Step 3. If $b_1 > -\infty$, then we consider the residuals along the lower bound.

   (a) Compute $r(b_1)_i = y_i - b_1\, x_i$, $i = 1, \cdots, n$. Sort them in the ascending order and denote the ordered residuals by $\{r(b_1)_{(i)}, i = 1, \cdots, n\}$.

   (b) Compute the following quantities, for $l = 1, \cdots, n-h+1$,

$$\left\{ \begin{array}{rcl} \overline{r}(b_1)_{(l:l+h-1)} & = & \frac{1}{h}\sum_{k=l}^{l+h-1} r(b_1)_{(k)}\,, \\ \overline{rr}(b_1)_{(l:l+h-1)} & = & \frac{1}{h}\sum_{k=l}^{l+h-1} r(b_1)^2_{(k)}\,, \\ SS(b_1)_{(l:l+h-1)} & = & \dfrac{1}{1+b_1^2}[\overline{rr}(b_1)_{(l:l+h-1)} - \overline{r}(b_1)^2_{(l:l+h-1)}], \end{array} \right. \tag{2.4}$$

   by the recursion

$$\left\{ \begin{array}{rcl} \overline{r}(b_1)_{(l+1:l+h)} & = & \overline{r}(b_1)_{(l:l+h-1)} + \frac{1}{h}[r(b_1)_{(l+h)} - r(b_1)_{(l)}]\,, \\ \overline{rr}(b_1)_{(l+1:l+h)} & = & \dfrac{1}{1+b_1^2}\{\overline{rr}(b_1)_{(l:l+h-1)} + \frac{1}{h}[r(b_1)^2_{(l+h)} - r(b_1)^2_{(l)}]\}\,. \end{array} \right. \tag{2.5}$$

(c) Save the size-h subset(s) that achieves the minimum of sum of squares.

3. Take a value β such that $b_1 < \beta < b^{[1]}$.

   (a) Compute $r(\beta)_i = y_i - \beta x_i$, $i = 1, \cdots, n$. Sort them in the ascending order. For ties, we arrange them in the ascending order of x-values. We denote the ordered residuals by $r(\beta)_{\pi(i)}$, $i = 1, \cdots, n$, where $\{\pi(1), \pi(2), \cdots, \pi(n)\}$ is a permutation of $\{1, 2, \cdots, n\}$. We also denote the inverse of $\{\pi(1), \pi(2), \cdots, \pi(n)\}$ by $\{\lambda(1), \lambda(2), \cdots, \lambda(n)\}$.

   (b) Compute the following quantities, for $l = 1, \cdots, n - h + 1$,

$$\left\{\begin{array}{rcl}
\overline{x}_{(l:l+h-1)} & = & \frac{1}{h}\sum_{i=l}^{l+h-1} x_{\pi(i)}\,, \\
\overline{y}_{(l:l+h-1)} & = & \frac{1}{h}\sum_{i=l}^{l+h-1} y_{\pi(i)}\,, \\
\overline{xx}_{(l:l+h-1)} & = & \frac{1}{h}\sum_{i=l}^{l+h-1} x^2_{\pi(i)}\,, \\
\overline{yy}_{(l:l+h-1)} & = & \frac{1}{h}\sum_{i=l}^{l+h-1} y^2_{\pi(i)}\,, \\
\overline{xy}_{(l:l+h-1)} & = & \frac{1}{h}\sum_{i=l}^{l+h-1} x_{\pi(i)}y_{\pi(i)}\,, \\
\overline{cxy}_{(l:l+h-1)} & = & \overline{xy}_{(l:l+h-1)} - \overline{x}_{(l:l+h-1)}\,\overline{y}_{(l:l+h-1)}\,, \\
B_{(l:l+h-1)} & = & \dfrac{[\overline{xx}_{(l:l+h-1)} - \overline{x}^2_{(l:l+h-1)}] - [\overline{yy}_{(l:l+h-1)} - \overline{y}^2_{(l:l+h-1)}]}{2\overline{cxy}_{(l:l+h-1)}}\,, \\
\hat{\beta}_{(l:l+h-1)} & = & \pm\sqrt{B^2_{(l:l+h-1)} + 1} - B_{(l:l+h-1)} \\
 & & (if\ \ \overline{cxy}_{(l:l+h-1)} \geq 0\ \ take\ \ '+',\ \ esle\ \ take\ \ '-'), \\
\hat{\alpha}_{(l:l+h-1)} & = & \overline{y}_{(l:l+h-1)} - \hat{\beta}_{(l:l+h-1)}\overline{x}_{(l:l+h-1)}\,, \\
SS_{(l:l+h-1)} & = & \dfrac{1}{1+\beta^2}\{[\overline{yy}_{(l:l+h-1)} - \overline{y}^2_{(l:l+h-1)}] - 2\hat{\beta}_{(l:l+h-1)}\overline{cxy}_{(l:l+h-1)} \\
 & & +\hat{\beta}^2_{(l:l+h-1)}[\overline{xx}_{(l:l+h-1)} - \overline{x}^2_{(l:l+h-1)}]\}\,,
\end{array}\right. \tag{2.6}$$

by the recursion

$$\begin{cases} \overline{x}_{(l+1:l+h)} & = & \overline{x}_{(l:l+h-1)} + \frac{1}{h}[x_{\pi(l+h)} - x_{\pi(l)}] \,, \\ \overline{y}_{(l+1:l+h)} & = & \overline{y}_{(l:l+h-1)} + \frac{1}{h}[y_{\pi(l+h)} - y_{\pi(l)}] \,, \\ \overline{xx}_{(l+1:l+h)} & = & \overline{xx}_{(l:l+h-1)} + \frac{1}{h}[x^2_{\pi(l+h)} - x^2_{\pi(l)}] \,, \\ \overline{yy}_{(l+1:l+h)} & = & \overline{yy}_{(l:l+h-1)} + \frac{1}{h}[y^2_{\pi(l+h)} - y^2_{\pi(l)}] \,, \\ \overline{xy}_{(l+1:l+h)} & = & \overline{xy}_{(l:l+h-1)} + \frac{1}{h}[x_{\pi(l+h)}y_{\pi(l+h)} - x_{\pi(l)}y_{\pi(l)}] \,. \end{cases} \tag{2.7}$$

(c) Update the BPLTS solution.

4. For $k = 1, 2, \cdots, L$, do the following.

   (a) Consider $b^{[k]}$ and the corresponding pair of indices (i, j). Update the permutation π by letting $\pi(\lambda(i)) = j$ and $\pi(\lambda(j)) = i$, and update λ by swapping $\lambda(i)$ and $\lambda(j)$.

   (b) Update the quantities in (2.6).

   (c) Update the BPLTS solution.

5. If $b_2 < \infty$, go through Step 2, replace b_1 by b_2, and update the BPLTS solution.

Proposition 4. *The output of the above Algorithm is the BPLTS solution.*

Proof: First, subject to the constraint $b_1 < \beta < b^{[1]}$, $r(\beta)_{\pi(1)} \leq r(\beta)_{\pi(2)} \leq \cdots \leq r(\beta)_{\pi(n)}$ is always true for the permutation $\{\pi(1), \pi(2), \cdots, \pi(n)\}$. Otherwise, we can find two indices (i, j) such that $y_i - \beta\, x_i < y_j - \beta\, x_j$ for some β values and $y_i - \beta\, x_i > y_j - \beta\, x_j$ for some other β values. Then there exists at least one value $b \in (b_1, b^{[1]})$ such that $y_i - b\, x_i = y_j - b\, x_j$ and $x_i \neq x_j$. This leads to $b = \frac{y_j - y_i}{x_j - x_i}$, a contradiction to the assumption that no other $b^{(i,j)}$ exists between b_1 and $b^{[1]}$. Hence we can apply Proposition

3 to the interval $[b_1, b^{[1]}]$. As a consequence, we only need to check the $n-h+1$ subsample $\{\pi(l), \pi(l+1), \cdots, \pi(l+h-1)\}$, where $1 \leq l \leq n-h+1$. This is exactly Step 2. Next we consider the order structure as β reaches $b^{[1]}$. Remember that we have assumed $b^{[1]} < b^{[2]}$. As β passes $b^{[1]}$ into the interval $(b^{[1]}, b^{[2]})$, the order structure is preserved except for the pair of indices (i,j) such that $b^{(i,j)} = \frac{y_j - y_i}{x_j - x_i} = b^{[1]}$. Otherwise, a self-contradiction would occur by the same argument as in the interval $(b_1, b^{[1]})$. If we do the swap as in Step 4(a): $\pi(\lambda(i)) = j$ and $\pi(\lambda(j)) = i$, and exchange $\lambda(i)$ and $\lambda(j)$, then the residuals under the new permutation π are still in the ascending order subject to $b^{[1]} \leq \beta < b^{[2]}$. Next we update the quantities in (2.6) and apply Proposition 3 to this region. Recursively, we rotate the slope around the origin and repeat this procedure for each region $b^{[k]} \leq \beta < b^{[k+1]}$.

Proposition 3 guarantees that we can find the size-h subset(s) for the optimal solution at the end of search. Next we evaluate the complexity of the algorithm. This requires a more detailed picture of the order structure.

Proposition 5. *If we assume that*

1. *Data contain no identical observations.*

2. *No tie exists in the slopes $\{b^{[l]}\}$, namely, $b_1 < b^{[1]} < b^{[2]} < \cdots < b^{[L]} < b_2$.*

Then

- *as β goes from an interval $[b^{[k-1]}, b^{[k]})$ to the next $[b^{[k]}, b^{[k+1]})$, the pair of indices (i_k, j_k) involved in the swap, cf. Step 4(a) of the algorithm, must be adjacent to one another in the permutations π in the two intervals $[b^{[k-1]}, b^{[k]})$ and $[b^{[k]}, b^{[k+1]})$;*

- *Consequently, we only need to adjust two partial regressions given in Equation (2.6) and thus the complexity of the algorithm is quadratic in time except for the part of sorting $\{b^{(i,j)}\}$.*

Proof: When $\beta \in [b^{[k-1]}, b^{[k]})$, we have either $y_{i_k} - \beta\, x_{i_k} < y_{j_k} - \beta\, x_{j_k}$ or $y_{i_k} - \beta\, x_{i_k} > y_{j_k} - \beta\, x_{j_k}$. Without loss of generality, we assume the former is true. That is,

$$\begin{cases} y_{i_k} - \beta\, x_{i_k} < y_{j_k} - \beta\, x_{j_k}, & b^{[k-1]} < \beta < b^{[k]}\,, \\ y_{i_k} - \beta\, x_{i_k} = y_{j_k} - \beta\, x_{j_k}, & \beta = b^{[k]}\,, \\ y_{i_k} - \beta\, x_{i_k} > y_{j_k} - \beta\, x_{j_k}, & b^{[k]} < \beta < b^{[k+1]}\,. \end{cases}$$

Suppose we have another term (x_u, y_u) whose residual is between these two terms for $\beta \in [b^{[k-1]}, b^{[k]})$. That is, $y_{i_k} - \beta\, x_{i_k} < y_u - \beta\, x_u < y_{j_k} - \beta\, x_{j_k}$. This implies

$$0 < (y_u - y_{i_k}) - \beta\,(x_u - x_{i_k}) < (y_{j_k} - y_{i_k}) - \beta\,(x_{j_k} - x_{i_k})\,.$$

When $\beta \longrightarrow b^{[k]}$, we have

$$0 \leq (y_u - y_{i_k}) - \beta\,(x_u - x_{i_k}) \leq (y_{j_k} - y_{i_k}) - \beta\,(x_{j_k} - x_{i_k}) = 0\,.$$

Hence we have $y_u - y_{i_k} = \beta\,(x_u - x_{i_k})$ and $b^{[k]} = \frac{y_u - y_{i_k}}{x_u - x_{i_k}}$. This conflicts the assumption. Similarly, the two terms indexed by i_k, j_k are still next to each other in the interval $[b^{[k]}, b^{[k+1]})$. The only change in π is the swap of their positions. This exchange involves only two partial regressions given in (2.6).

It should be note that the above algorithm gives the exact solution only if there is no tie exists in the slope $b^{(i,j)}$'s. In practise, we can avoid slope tie by introducing tiny random perturbations to the data points (x_i, y_i), namely, we add a tiny random number to each x_i or y_i. In addition, we also have a version of algorithm that can deal with general case in which ties may occur in $b^{(i,j)}$'s. But it is not as efficient as the above simple version in terms of time complexity. Further more, we find that the simple version can obtain almost the same result as the general version, if random perturbation is performed. So in the thesis, we always use the results from the simple version.

Chapter 3

Mathematical Properties of BLTS Methods

3.1 Symmetry of BLTS methods with respect to x and y

The target function of ordinary least trimmed squares (OLTS) only considers the errors from response variable. Regression of y on x ($y \sim x$) and regression of x on y ($x \sim y$) will give distinct h-size subset and regression line. On the other hand, the target functions for bivariate multiplicative least trimmed squares (BMLTS), bivariate symmetric least trimmed squares (BSLTS), and bivariate perpendicular least trimmed squares (BPLTS) are all symmetric with respect to x and y. Therefore, regression $y \sim x$ and $x \sim y$ will result in the same h-size subsets and regression lines. We applied OLTS, BMLTS, BSLTS, and BPLTS regression with both $y \sim x$ and $x \sim y$ to a data with 200 points, the result is show in Figure 3.1. The trimming fraction for all the LTS are set to 30%. The OLTS lead to two distinct regression lines corresponding to $y \sim x$ and $x \sim y$, as shown in Figure 3.1A. While the other three bivariate least trimmed squares (BMLTS, BSLTS and BPLTS) achieve the same result for $y \sim x$ and $x \sim y$, as shown in Figure 3.1B-D.

Figure 3.2 shows the results of ordinary least squares (OLS), OLTS and BPLTS using the same data set. This time a trimming fraction of 20% are used. As can be seen, the h-size subsets of $y \sim x$ (Figure 3.2A) and $x \sim y$ (Figure 3.2B) identified by OLTS are different with each other. In Figure 3.2D, the regression lines by OLS are

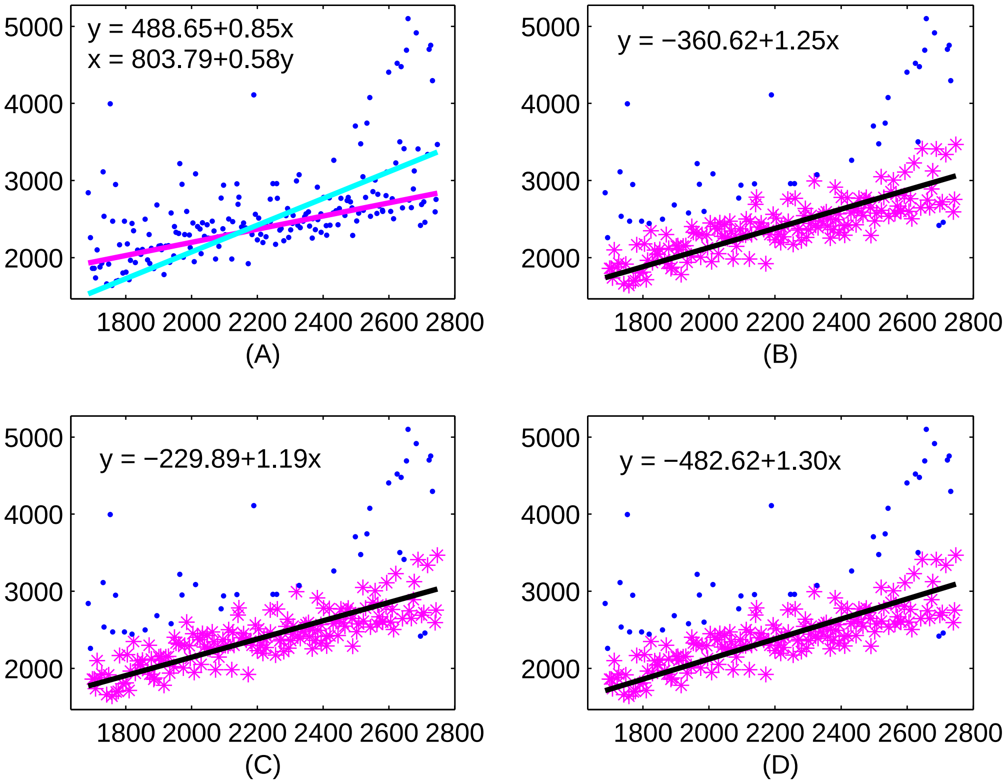
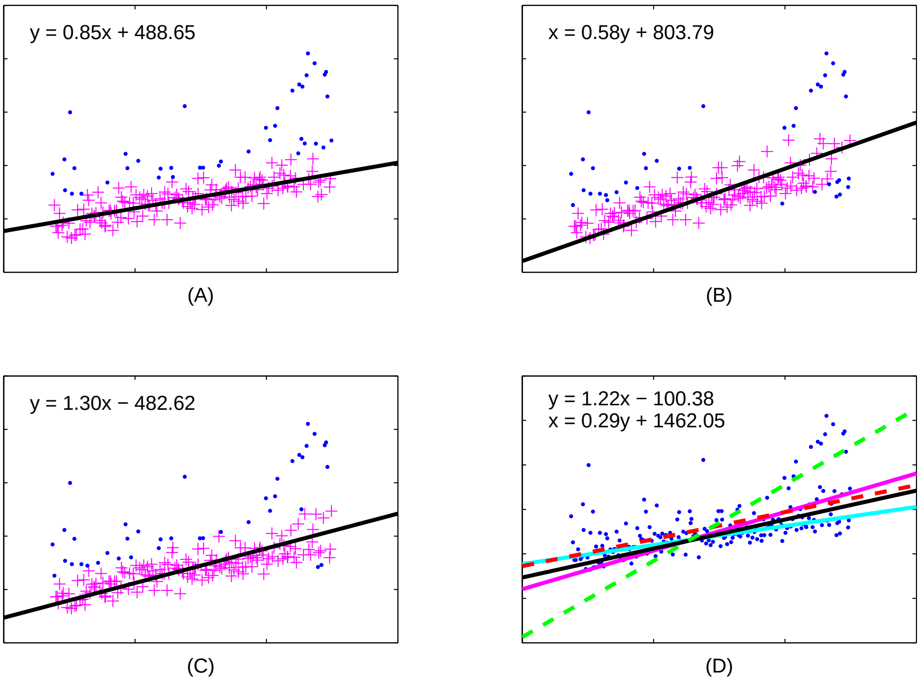


Figure 3.1: Fitted regression lines using (A)OLTS (B)BMLTS (C)BSLTS (D)BPLTS. In (A), the magenta and cyan lines are the best fitted lines of regression $y \sim$ x and $x \sim y$, respectively when OLTS is used. In (B)-(D), the blue points are excluded outliers. In (A)-(D), a trimming fraction of 30% is used.

shown as dashed lines. Since OLS estimate the correlation coefficients based on all the data points without trimming, it will affected by the outliers in the data. As shown, the regression lines of $y \sim x$ and $x \sim y$ by OLS are more different in comparison with those by OLTS. Despite the difference, all the LTS methods give more reliable estimation of the regression coefficients than the OLS method.



(A) (B) (C) (D)

Figure 3.2: The first simulated data set. (A) OLTS with $y \sim x$ (B) OLTS with $x \sim y$ (C) BLTS (D) Ordinary least square fitting. In (A)(B)(C): the trimming fraction is 20%; the outliers are shown as blue dots. in (D): the dashed lines are from ordinary least squares(red: $y \sim x$; green: $x \sim y$); the solid lines are from OLTS (cyan: $y \sim x$; purple: $x \sim y$) and BLTS (black).

3.2 Comparison of BLTS methods

We then investigate the accuracy of OLTS and our three BLTS methods for estimate linear relation between two variables. When we estimate the relation between father's height and son's height, we measure the height of all the father-son pairs. Certainly, we would expect experiment errors for both father's height and son's height. However, the OLS and OLTS only take in consideration the errors from the response variable, that is, the errors from the father's height when regress father's height on son'height. The three

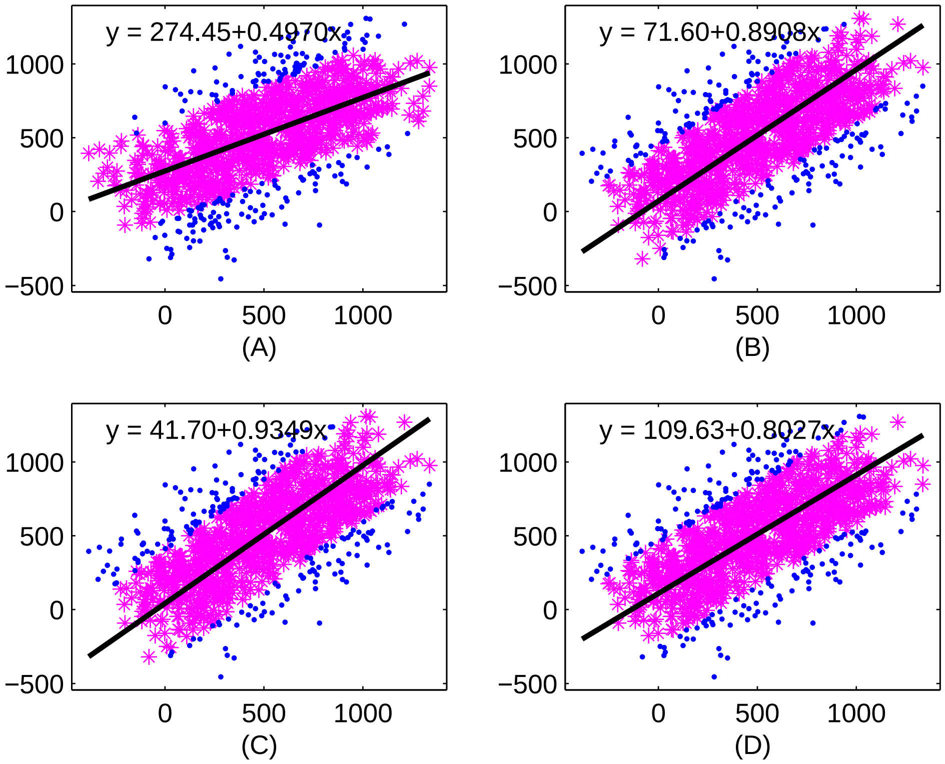


Figure 3.3: Comparison of OLTS(A), BMLTS(B), BSLTS(C) and BPLTS(D) using simulated data set with errors in both x and y. The magenta stars mark the data points in the h-size subset. The blue dots indicate the outliers.

methods BMLTS, BSLTS and BPLTS we propose in the thesis consider errors from both variables. As such, we may expect a more accurate estimation of the linear relation between the two variables. To investigate this problem, we simulate a data set of size 1000 using the following procedure. First, we generate a vector $X = [1, 2, ..., 1000]$, Second we generate another vector Y, where $Y_i = 100 + 0.8X_i, i = 1, 2, ..., 1000$. Finally, we introduce errors for both X and Y by adding a random number ϵ_x or ϵ_y for each X_i and Y_i, where $\epsilon_x \sim (0, 200)$, and $\epsilon_y \sim (0, 200)$. That is for the 1000 observations (x_i, y_i), we have $x_i = \mu_{x_i} + \varepsilon_{x_i}$ and $y_i = \mu_{y_i} + \varepsilon_{y_i}$, where the ε_{x_i} and

ε_{y_i} are independently and identically distributed as N(0,200). The underlying relation between x and y in the simulated data is $\mu_{y_i} = 100 + 0.8\mu_{x_i}$.

We applied OLTS, BMLTS, BSLTS and BPLTS to the simulated set using a trimming fraction of 20%. As shown in Figure 3.3, the regression lines are

1. $y = 279.45 + 0.4970x$, OLTS ($y \sim x$)

2. $y = 71.60 + 0.8908x$, BMLTS

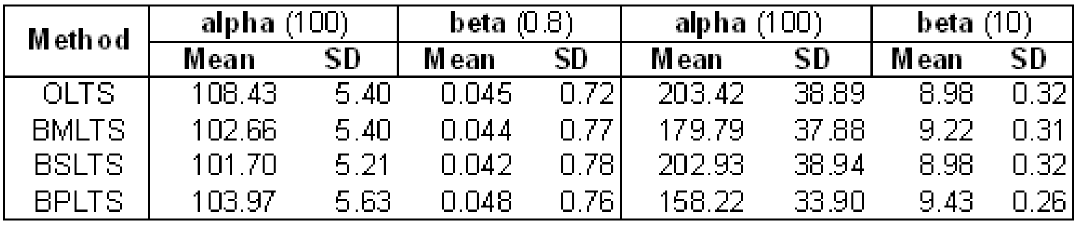
3. $y = 41.70 + 0.9349x$, BSLTS

4. $y = 109.63 + 0.8027x$, BPLTS

The result shows that all the three bivariate LTS method that consider errors from both x and y achieve more accurate result than OLTS method. Among the three bivariate LTS methods, the one that minimize the sum of perpendicular distances gives the best result. The estimated slope and intercept are 0.8027 and 109.63 respectively, which are approximately equal to the real ones.

Given that $\varepsilon_x = \varepsilon_y$, The BMLTS and BSLTS can achieve good estimate of β only if the two variables x and y have comparable scales. When the scale of one variable is much larger than that of the other one, say $y \gg x$ (the real slope $\beta \gg 1$), we will tend to have $|y_i - \hat{y_i}| \gg |x_i - \hat{x_i}|$ because the measurement errors of x and y are equal. Therefore BMLTS and BSLTS tend to result in a $\hat{\beta}$ smaller than the real β to reduce the deviations in y. Conversely, if $y \ll x$ (the real slope $\beta \ll 1$), BMLTS and BSLTS result in a $\hat{\beta}$ larger than the real β. We have simulated data set with various β values, BPLTS method achieves the most accurate estimate of β in most cases. In practice, the variable with a smaller scale has the larger relative measurement error, given that the two variables x and y have the same absolute measurement errors $\varepsilon_x \cong \varepsilon_y$ and hence errors in this variable play more roles in BLTS regression. This is reasonable, because the variable with smaller precision should be treated with much more attention.

Table 3.1: Comparison of different approaches.



Method	alpha (100)		beta (0.8)		alpha (100)		beta (10)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
OLTS	108.43	5.40	0.045	0.72	203.42	38.89	8.98	0.32
BMLTS	102.66	5.40	0.044	0.77	179.79	37.88	9.22	0.31
BSLTS	101.70	5.21	0.042	0.78	202.93	38.94	8.98	0.32
BPLTS	103.97	5.63	0.048	0.76	158.22	33.90	9.43	0.26

To obtain the bias and variation of $\hat{\alpha}$ and $\hat{\beta}$, we simulate 1000 data sets using above described methods. Each for data set includes 200 observations and ε_x, ε_y follows a normal distribution N(0,10). The results of OLTS and the three BLTS are shown in Table 3.1. For all of the methods, the trimming fraction is set to 20$\%$. As can be seen, when $\beta = 0.8$, BMLTS, BSLTS, and BPLTS are comparable. When $\beta = 10$, the $\hat{\beta}$ by BPLTS has smaller bias and variation and thereby outperforms BMLTS and BSLTS. In both cases, the three BLTS methods achieve more sensible results than OLTS.

3.3 Breakdown value of BPLTS method

In this section, we investigate the robustness of BPLTS method using simulated data set. The simulated data set is obtained following the instructions of Rousseeuw and Leroy to investigate the breakdown points of our BPLTS estimate [RL99]. The procedure is described as following:

1. Generate 100 samples from: $Y = X + 20 + \epsilon$, where x is from 10.5 to 60 at equally spaced 0.5 units and $\epsilon \sim N(0, 100^2)$;

2. Generate another N samples that are considered to be outliers from(X, Y), where $X \sim Uniform(60, 80)$ and $Y \sim Uniform(10, 30)$;

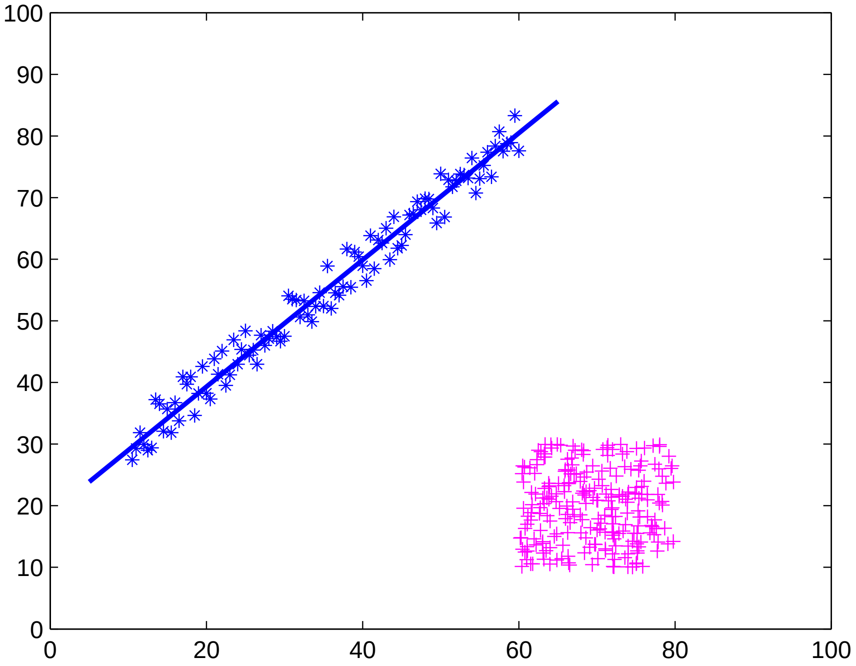
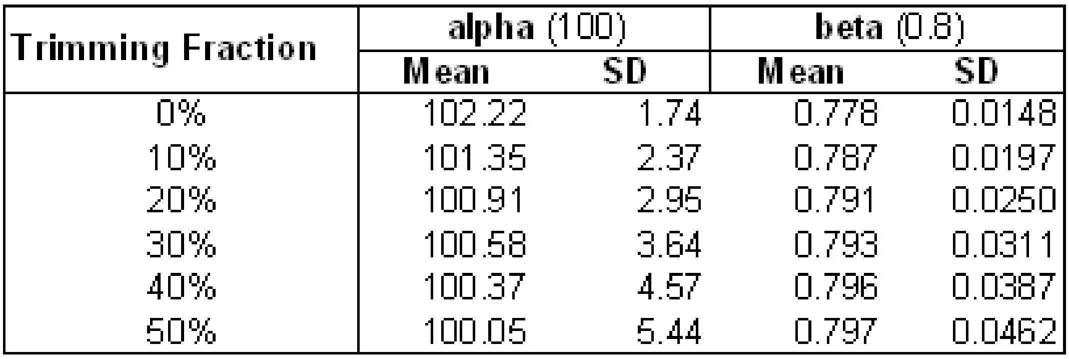


Figure 3.4: Robustness of BPLTS method. The purple "+"s in the bottom left are "outliers".

3. We genearate 8 data sets with $N = 400, 233, 150, 100, 67, 43, 25, 11$, corresponding to outlier percentage of $80\%, 70\%, 60\%, 50\%, 40\%, 30\%, 20\%$ and 10%, respectively.

Theoretically,we can achieve any break point τ by setting the trimming fraction to τ in the BPLTS method. Figure 3.4 shows that the BPLTS method can give the correct best fitted line from a simulated data with as much as 70% outliers. It should be noted here the "outliers" in these simulated data sets are not real outliers, because they all follow into a small region. So if there are enough number of "outliers", it will generate an artificial fitted lines in the outlier region and the BPLTS doesn't give the correct regression line any more.

Table 3.2: Trimming fraction effect on coefficient estimation using BPLTS.



Trimming Fraction	alpha (100)		beta (0.8)	
	Mean	SD	Mean	SD
0%	102.22	1.74	0.778	0.0148
10%	101.35	2.37	0.787	0.0197
20%	100.91	2.95	0.791	0.0250
30%	100.58	3.64	0.793	0.0311
40%	100.37	4.57	0.796	0.0387
50%	100.05	5.44	0.797	0.0462

A larger trimming fraction lead to a higher break down value, but at the cost of efficiency. As shown by simulations in Table 3.2, higher trimming fractions result in smaller biases for both $\hat{\alpha}$ and $\hat{\beta}$ but the variations are increased.

3.4 Analysis of multi-component data set using BPLTS

Since the BPLTS method can estimate the correct linear relation between variable regardless of the outliers, it can be used to analyze the data set with multiple components. The idea is estimate the relation of the variables in the first component using BPLTS, then remove points in the identified h-size subset and estimate the relation between variables in the second component by applying BPLTS to the rest of data points, and then analyze the third component, and so on. The key is to set an appropriate trimming fraction in each BPLTS regression. We simulate a data set with two component and the result of analysis for this data set is shown in Figure 3.5. We simulated a data set of size 600 from a mixture model with two components. The model is illustrated as follows:

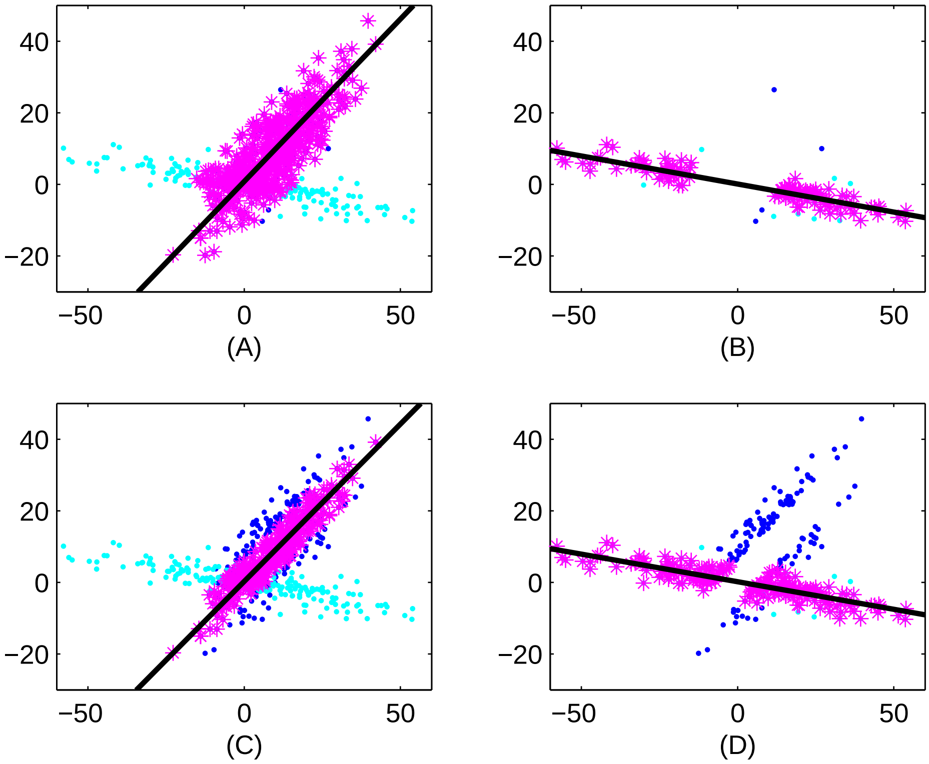


Figure 3.5: Analysis of multi-component data set using BPLTS. (A) Identification of the first component. The trimming fraction is 20%.(B) Identification of the second component. The data points in the h-size subset identified by the first BPLTS regression are excluded from the second BPLTS regression. The trimming fraction is 10%. (C)Same as in (A), but the trimming fraction is 50%. (D)Same as in (B), but the trimming fraction is 40%. The data points from the first and the second component are marked by blue and cyan dots, respectively. The magenta stars indicates the data points in the h-size subset for the corresponding BPLTS regression.

1. The main component contained 400 samples. They are collected from the bivariate normal distribution $(X, Y) \sim (\mu_1, \sum_1)$, where $\mu_1 = [10, 10]$ and $\sum_1 = \begin{pmatrix} 100 & 80 \\ 80 & 100 \end{pmatrix}$. So $\rho_{X,Y} = 0.8$ in the major component;

2. The minor component contained 200 samples. They are collected from the bivariate normal distribution $(X, Y) \sim (\mu_2, \sum_2)$, where $\mu_1 = [0, 0]$ and $\sum_2 = \begin{pmatrix} 500 & -80 \\ -80 & 20 \end{pmatrix}$. So $\rho_{X,Y} = -0.8$ in the minor component.

As shown in Figure 3.5, the relation between x and y is correctly identified in the first BPLTS regression(see Figure 3.5A and C). Then the relation between x and y in the second component is correctly estimated by the second BPLTS regression(see Figure 3.5B and C). The choosing of trimming fraction is whatsoever flexible. The trimming fractions for the first BPLTS are 20% and 50% in Figure 3.5A and Figure 3.5C respectively, both give the correct estimation. In the second BPLTS regression, a trimming fraction of either 50% in Figure 3.5B or 40% in Figure 3.5D gives the correct estimation. These results show that BPLTS can be used to analyze those data sets with multiple components.

3.5 Subset correlation analysis using BPLTS

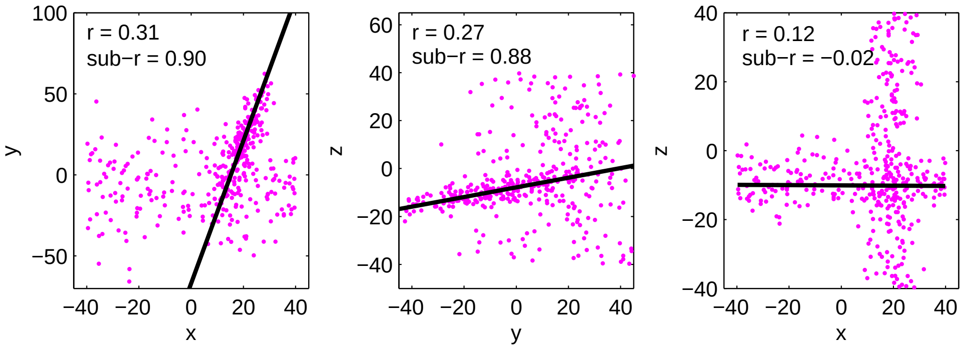


Figure 3.6: Subset correlation analysis using BPLTS. (A) 50%-subset correlation between x and y. (B) 50%-subset correlation between y and z. (C) 50%-subset correlation between x and z.

In the data set with multiple components, we may not expect the correlation between two variable across all the data points. For example, two variables x and y may correlated with each other in only one component that represents 50% data points. We can define P-subset correlation as the correlation of two variables in the subset identified by BPLTS with trimming fraction 1-P, where $0 < P \leq 1$. For instance, we may have variables X, Y and Z, where X and Y are correlated in some conditions; Y and Z are correlated in some other conditions. Therefore, if we have n observation of (X, Y, Z), we would expect X and Y are correlated in a subset of data points, Y and Z are correlated in another subset of data points,but no correlation between X and Z. Using the ordinary correlation measurement that estimate the liner relationship across all the data points would fail to identify the underlying relations between them. But subset correlation works in this situation. To simulate this situation, we perform the following procedures:

1. Generate $(X_i, Y_i), i = 1...200$ from the bivariate normal distribution (X, Y), where $\mu = (20, 20)$ and $\Sigma = \begin{pmatrix} 0.25 & 0.8 \\ 0.8 & 4 \end{pmatrix}$.
2. Generate $(Y_i, Z_i), i = 201...400$ from the bivariate normal distribution (Y, Z), where $\mu = (-10, -10)$ and $\Sigma = \begin{pmatrix} 4 & 0.8 \\ 0.8 & 0.25 \end{pmatrix}$.
3. Generate $X_i,\ i = 201...400$ and $Z_j,\ j = 1...200$ from uniform distribution $Unif(-40, 40)$.

Then we obtain the data set in which X and Y are correlated ($PCC = 0.8$) in 50% data points; Y and Z are correlated ($PCC = 0.8$) in the other 50% data points.

The result is shown in Figure 3.6, we can find that the ordinary Pearson correlation coefficients are $\rho_{xy} = 0.31$, $\rho_{yz} = 0.27$, and $\rho_{yz} = 0.12$, respectively. But the 50% subset Pearson correlation coefficient are $\rho_{xy}^{50\%} = 0.90$, $\rho_{yz}^{50\%} = 0.88$, and $\rho_{yz}^{50\%} = -0.02$, respectively. This indicates that the subset correlation can correctly expose the linear relationships among the three variables.

3.6 Scale dependency of BPLTS

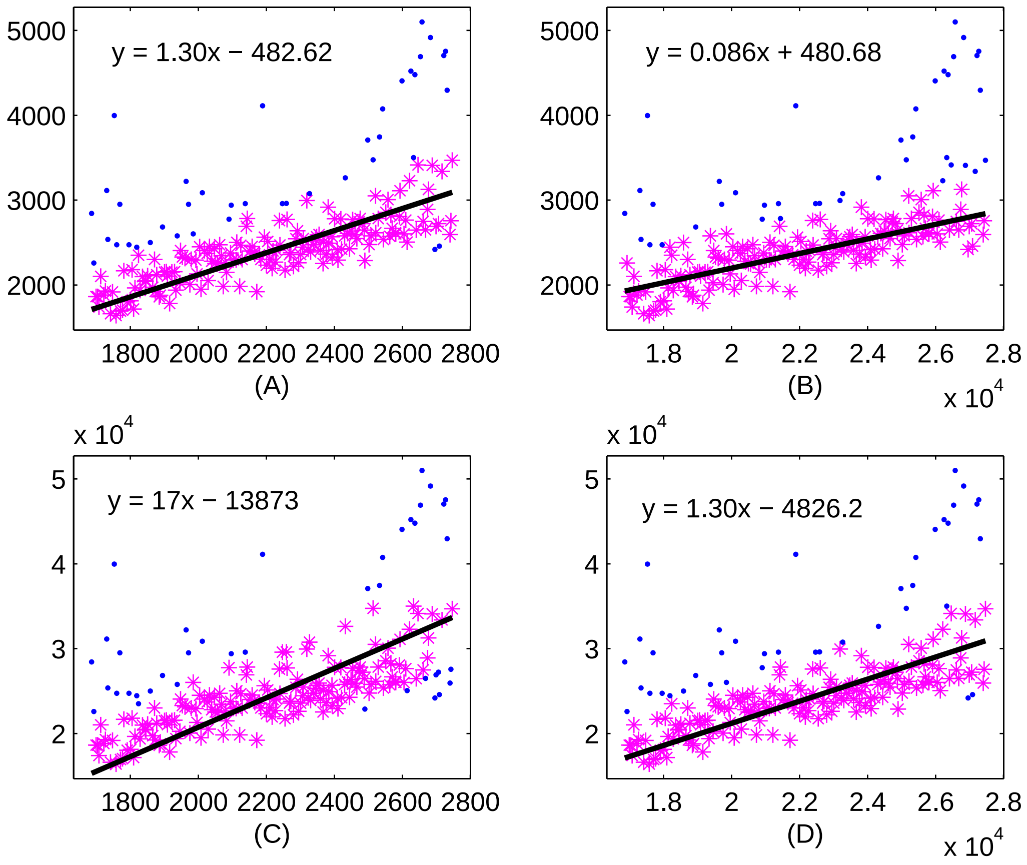


Figure 3.7: BPLTS is scale dependent. (A) Regression of y on x. (B) Regression of y on $10x$. (C) Regression of $10y$ on x. (D) Regression of $10y$ on $10x$. The magenta stars mark the h-size subset identified by the BPLTS regression. The blue dots are those outliers.

As we know, the ordinary least trimmed squares (OLTS) is scale equivariant. That is, it results in the same subset when the scale of x or y is changed. However, the scale equivariant is not hold by BPLTS regression. As shown in Figure 3.7, the BPLTS regression gives different h-size subset of data points if we change the scale of y or x. But if we change the scales of x and y at the same level, BPLTS regression results in the same h-size subset as the one obtained in the original data.

This bring about the question: what scale should we use for BPLTS regression? For the observations (x_i, y_i) of size n, if we have: $x_i = \mu_{x_i} + \varepsilon_x$ and $y_i = \mu_{y_i} + \varepsilon_y$, where $\varepsilon_x \sim N(0, \sigma_x^2)$ and $\varepsilon_y \sim N(0, \sigma_y^2)$, we should make transformation for either x or y, say $y' = cy$, such that $\sigma_{y'} = \sigma_x$. After this transformation, the errors from the two variables are equally considered and thereby the BPLTS regression gives a reasonable results.

Chapter 4

Application of BPLTS Method to Microarray Normalization

Microarray technologies have been widely used in recent years, which provide expression measurement of tens of thousand genes at the same time. One of the key step for microarray data analysis is normalization. After normalization, gene expression measurements from various arrays are comparable and can be used to detect gene expression changes. In this chapter, we will introduce a novel microarray normalization method that is based on our new developed perpendicular least trimmed squares (BPLTS).

4.1 Introduction to microarray technologies and normalization

Microarray is a key technique in the study of functional genomics. It measures abundance of mRNAs by hybridization to appropriate probes on a glass chip. The current technique can hold hundreds of thousands of probes on a single chip. This allows us to have snapshots of expression profiles of a living cell. In the thesis, we mainly consider high-density oligonucleotide arrays. The Affymetrix GeneChip® uses 11-20 probe pairs, which are short oligonucleotides of 25 base pairs, to represent each gene, and as a whole they are called a probe set [Aff01, LDB$^+$96]. Each probe pair consist of a perfect match (PM) and a mis-match (MM) probe that differ only in the middle (13-th) base.

Mis-match probes are designed to remove the effects of non-specific binding, cross-hybridization, and electronic noise. Ideally, probes are arranged on a chip in a random fashion. But in customized arrays, this is not always true.

From microarray measurements, we seek differentiation of mRNA expression among different cells. However, each array has a "block effect" due to variation in RNA extraction, labeling, fluorescent detection, etc. Without statistical treatment, this block effect is confounded with real expression differentiation. The statistical treatment of reducing the block effect is defined to be normalization. It is usually done at the probe level. Several normalization methods for oligonucleotide arrays have been proposed and practiced. One approach uses *lowess* [Cle79] to correct for non-central and non-linear bias observed in M-A plots [YDLS01]. Another class of approaches correct for the non-linear bias seen in Q-Q plots [SHG$^+$02, WJJ$^+$02, BIS03]. As discussed in [ZAZL01, WJJ$^+$02], several assumptions must hold in the methods using quantiles. First, most genes are not differentially regulated; Second, the number of up-regulated genes roughly equals the number of down-regulated genes; Third, the above two assumptions hold across the signal-intensity range.

Our perspective of normalization is that of blind inversion [Li03]. The basic idea is to find a transformation for the target array so that the joint distribution of hybridization levels of the target and reference array matches a nominal one. Two different ideas exist to achieve this goal. First, quantiles allows us to compare distributions and the Q-Q plot is the standard graphical tool for the purpose. The normalization proposed in [SHG$^+$02, WJJ$^+$02, BIS03] aims to match the marginal distribution of hybridization levels from the target with that from reference. Although slight and subtle difference exists between the two principles, quantile methods work well for arrays with little differentiation. The second idea is regression, either linear or non-linear [YDLS01, SLSW01].

4.2 Application of BPLTS to Microarray normalization

When we compare two arrays in which a substantially large portion of genes are differentially expressed, we need to identify a "base" subset for the purpose of normalization. This subset should exclude those probes corresponding to differentially expressed genes and abnormal probes due to experimental variation. A similar concept "invariant set" is defined in [SLEW, TOR$^+$01, KCM02]. We use perpendicular least trimmed squares (BPLTS) to identify the base for normalization and to estimate the transformation in a simultaneous fashion. Substantial differentiation is protected in BPLTS by setting an appropriate trimming fraction. The exact BPLTS solution is computed by the algorithm we described in Chapter 2.

Array-specific spatial patterns may exist due to uneven hybridization and measurement process. For example, reagent flow during the washing procedure after hybridization may be uneven; scanning may be non-uniform. We have observed different spatial patterns from one array to anther. To taken this into account, we divide each array into sub-arrays that consist of a few hundred probes, and normalize probe intensities within each sub-array.

Statistical principle of normalization Suppose we have two arrays: one reference and one target. Denote the measured fluorescence intensities from the target and reference arrays by $\{U_j, V_j\}$. Denote true concentrations of specific binding molecules by ($\tilde{U}_j$, $\tilde{V}_j$). Ideally, we expect that (U_j, V_j)=($\tilde{U}_j$, $\tilde{V}_j$). In practice, measurement bias exists due to uncontrolled factors and we need a normalization procedure to adjust measurement. Next we have another look at normalization. Consider a system with ($\tilde{U}_j$ $\tilde{V}_j$) as input and $(U_j,\ V_j)$ as output. Let $\mathbf{h} = (h_1, h_2)$ be the system function that accounts for all uncontrolled biological and instrumental bias; namely,

$$\begin{cases} U_j &= h_1(\tilde{U}_j)\,, \\ V_j &= h_2(\tilde{V}_j)\,. \end{cases}$$

The goal is to reconstruct the input variables $(\tilde{U}_j,\ \tilde{V}_j)$ based on the output variables $(U_j,\ V_j)$. It is a blind inversion problem [Li03], in which both input values and the effective system are unknown. The general idea is to find a transformation that matches the distributions of input and output. This leads us to the question: what is the joint distribution of true concentrations $(\tilde{U}_j,\ \tilde{V}_j)$? First, let us assume that the target and reference array are biologically undifferentiated. Then the differences between the target and reference are purely caused by random variation and uncontrolled factors. In this ideal case, it is reasonable to assume that the random variables $\{(\tilde{U}_j, \tilde{V}_j), j = 1, \cdots\})$ are independent samples from a joint distribution $\tilde{\Psi}$ whose density centers around the straight line $\tilde{U} = \tilde{V}$, namely, $E(\tilde{V}|\tilde{U}) = \tilde{U}$. The average deviations from the straight line measures the accuracy of the experiment. If the effective measurement system $\mathbf{h}$ is not an identity one, then the distribution of the output, denoted by Ψ, could be different from $\tilde{\Psi}$. An appropriate estimate $\hat{\mathbf{h}}$ of the transformation should satisfy the following: the distribution $\hat{\mathbf{h}}^{-1}(\Psi)$ matches $\tilde{\Psi}$, which centers around the line $\tilde{V} = \tilde{U}$. In other words, the right transformation straightens out the distribution of Ψ.

Next we consider the estimation problem. Roughly speaking, only the component of h_1 relative to h_2 is estimable. Thus we let $v = h_2(\tilde{v}) = \tilde{v}$. In addition, we assume that h_1 is a monotone function. Denote the inverse of h_1 by g, then we expect the following is valid.

$$E[\tilde{V}|\tilde{U}] = \tilde{U}, \quad \text{or} \quad E[V|g(U)] = g(U)\,.$$

Proposition 6. *Suppose the above equation is valid. Then g is the minimizer of* $\min_l E(V - l(U))^2$.

According to the well known fact of conditional expectation, $E[V|g(U)] = g(U)$ minimizes $E[V - l_1(g((U))]^2$ with respect to l_1. Next write $l_1(g(U)) = l(U)$. This fact suggests that we estimate g by minimizing $\sum_j (v_j - g(u_j))^2$. When necessary, we can impose smoothness on g by appropriate parametric or non-parametric forms.

Differentiation fraction and undifferentiated probe set Next we consider a more complicated situation. Suppose that a proportion λ of all the genes are differentially expressed while other genes are not except for random fluctuations. Consequently, the distribution of the input is a mixture of two components. One component consists of those undifferentiated genes, and its distribution is similar to $\tilde{\Psi}$. The other component consists of the differentially expressed genes and is denoted by $\tilde{\Gamma}$. Although it is difficult to know the form of $\tilde{\Gamma}$ as *a priori*, its contribution to the input is at most λ. The distribution of the input variables $(\tilde{U}_j, \tilde{V}_j)$ is the mixture $(1 - \lambda)\,\tilde{\Psi} + \lambda\,\tilde{\Gamma}$. Under the system function $\mathbf{h}$, $\tilde{\Psi}$ and $\tilde{\Gamma}$ are transformed respectively into distributions denoted by Ψ and Γ; That is, $\Psi = \mathbf{h}(\tilde{\Psi})$, $\Gamma = \mathbf{h}(\tilde{\Gamma})$. This implies that the distribution of the output (U_j, V_j) is $(1 - \lambda)\,\Psi + \lambda\,\Gamma$. If we can separate the two components Ψ and Γ, then the transformation $\mathbf{h}$ of some specific form could be estimated from the knowledge of $\tilde{\Psi}$ and Ψ.

Spatial pattern and sub-arrays Normalization can be carried out in combination with a stratification strategy. On the high-density oligonucleotide array, tens of thousands of probes are laid out on a chip. To take into account any plausible spatial variation in h, we divide each chip into sub-arrays, or small squares, and carry out normalization for probes within each sub-array. To get over any boundary effect, we allow sub-arrays to overlap. A probe in a overlapping regions gets multiple adjusted values from sub-arrays it belongs to, and we take their average.

In each sub-array contains only a few hundred probes, we choose to parameterize the function g by a simple linear function $\alpha + \beta\, u$, in which the background α and scale β represent respectively uncontrolled additive and multiplicative effects. Therefore, we apply BPLTS method to estimate α and β. Furthermore, by setting a proper trimming fraction ρ, we expect the corresponding size-h set identified by BPLTS is a subset of the undifferentiated probes explained earlier. Obviously, the trimming fraction ρ should be larger than the differentiation fraction λ. We call this BPLTS based normalization method SUB-SUB normalization. The details of sub-array normalization can be found in [CL05]

Implementation and SUB-SUB normalization We have developed a module to implement the normalization method describe above, referred as SUB-SUB normalization. The core code is written in C, and we have an interfaces with Bioconductor in R. The input of this program is a set of Affymetrix CEL files and output are their CEL files after normalization. Three parameters need to be specified: sub-array size, overlapping size and trimming fraction. The sub-array size specified the size of the sliding window. The overlapping size controls the smoothness of window-sliding. Trimming fraction specifies the break down value in LTS. An experiment with an expected higher differentiation fraction should be normalized with a higher trimming fraction.

4.3 Application of SUB-SUB normalization

Microarray data sets To test the effectiveness of sub-array normalization, we apply it to two microarray data sets. The first one is Affymetrix Spike-in data set includes fourteen arrays obtained from Affymetrix HG-U95 chips.

Fourteen genes in these arrays are spiked-in at given concentrations in a cyclic fashion known as a Latin square design. The data are available from *http://www.affymetrix.com/support/technical/sample_data/datasets.affx*. In the following analysis, we chose eight arrays out of the complete data set and split them into two groups. The first group contains four arrays: *1521m99hpp_av06*, *1521n99hpp_av06*, *1521o99hpp_av06*, *1521p99hpp_av06*. The second group contains four arrays: *1521-q99hpp_av06*, *1521r99hpp_av06*, *1521s99hpp_av06* and *1521t99hpp_av06*. For convenience, we abbreviate these arrays by M, N, O, P, Q, R, S, T. As a result, the concentrations of thirteen spiked-in genes in the second group are two-fold lower. The concentrations of the remaining spike-in gene are respectively 0 and 1024 in the two groups. In addition, two other genes are so controlled that their concentrations are also two-fold lower in the second group compared to the first one.

The second data set is also from the same Affymetrix web site, which contains two technical replicates using yeast array YG-S98. We will use them to study the variation between array replicates.

Spike-in data Figure 4.1 shows the distribution of log transformed expression levels of probes and genes in each of the eight arrays before and after SUB-SUB normalization. The expression level for a gene is calculated by combining all the probe levels corresponding to a genes using ”medianpolish” summarization method provided by Bioconductor. As shown, the expression measurement have various distribution without normalization, which is result from systematic errors and indicates that normalization is required for comparison of these arrays. After normalization the distributions of expression measurement are comparable both in probe level and in gene level. This result show that SUB-SUB can effectively reduce the systematical errors.

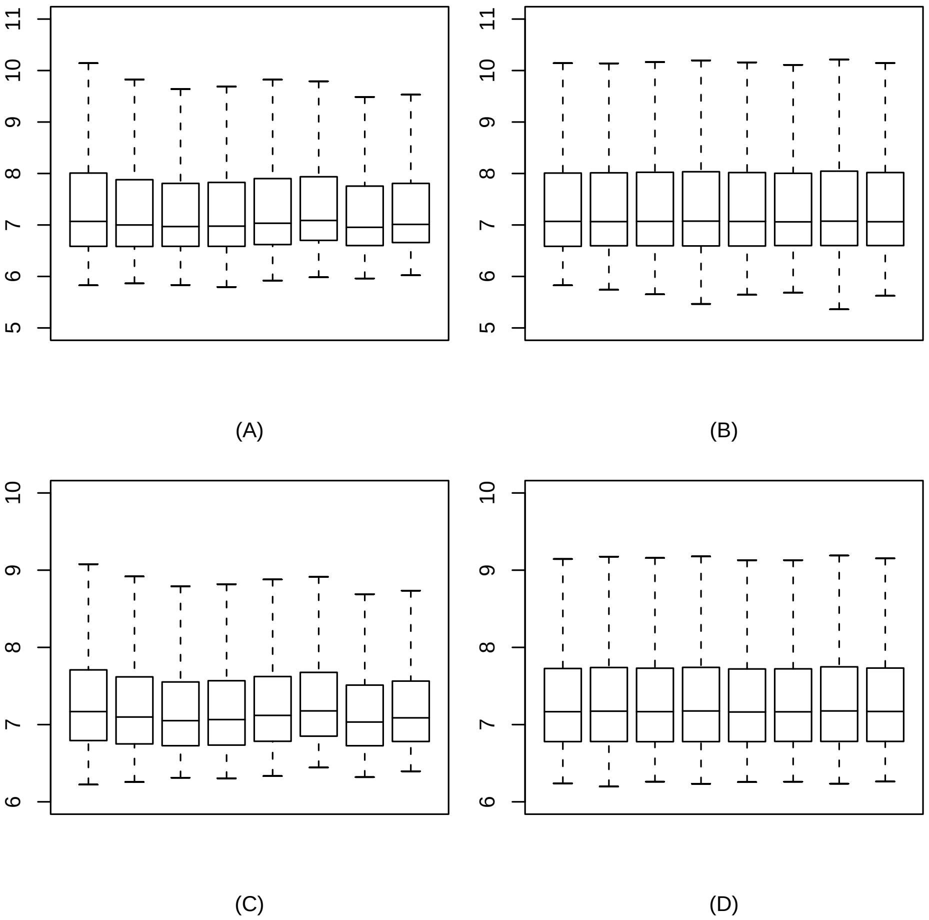
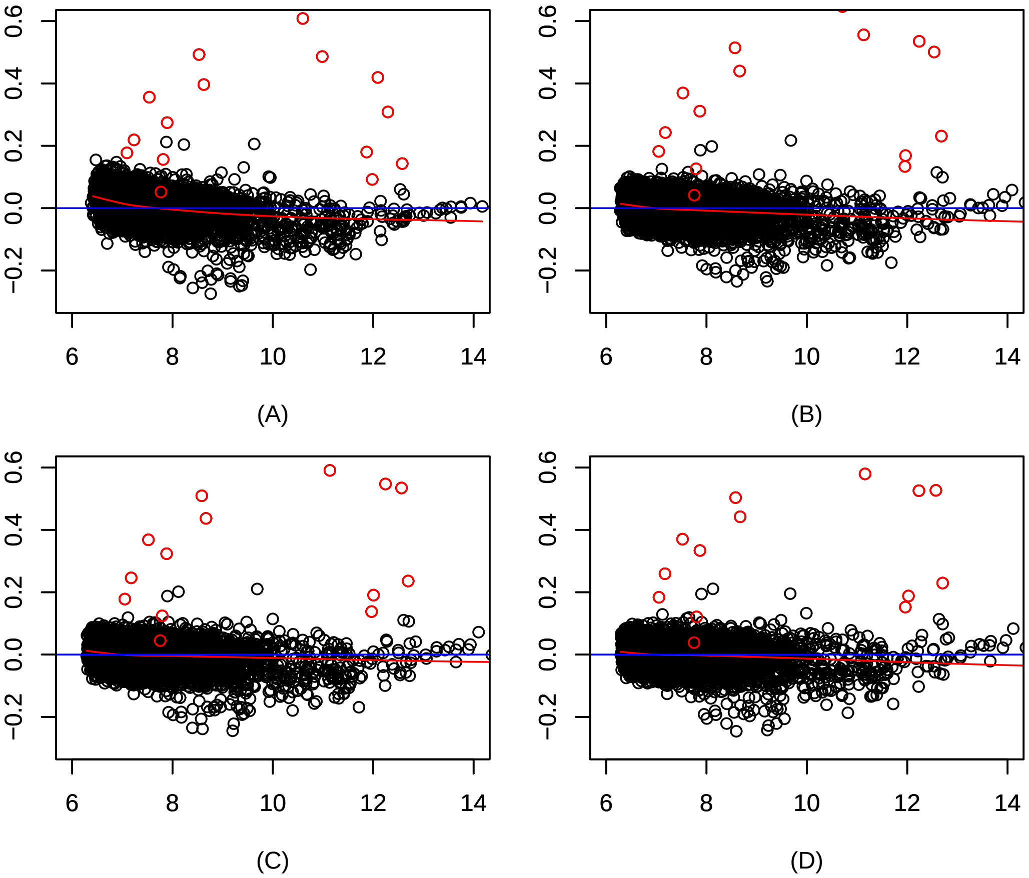


Figure 4.1: Distribution of log transformed expression measurement. (A) probe level before SUB-SUB normalization. (B)probe level after SUB-SUB normalization. (C) gene level before SUB-SUB normalization. (D)gene level after SUB-SUB normalization. SUB-SUB parameters: window size 20×20; overlapping size 10; trimming fraction: 80%.

We carry out SUB-SUB normalization to each of the eight arrays using Array M as the reference. We experimented with different sub-array sizes, overlapping sizes and trimming fractions. Figure 4.2 shows the M-A plots summarized from the eight arrays after normalization, namely, the log-ratios of expressions between the two groups versus the abundance. The sub-array size is 20×20, the overlapping size is 10 and the trimming factor ranging from 0 to 40% are used. Our result indicates that both the sub-array



(A)

(B)

(C)

(D)

Figure 4.2: M-A plots of spike-in data after Sub-Sub normalization. The x-axis is the average of log-intensities from two arrays. The y-axis is their difference after normalization. (A)trimming fraction is 0. (B)trimming fraction is 10%. (C)trimming fraction is 20%. (D)trimming fraction is 40%. The red points represent those Spike-in genes. Window size 20×20; Overlapping size: 10.

size (data not shown) and trimming fraction matter substantially for normalization. In other words, stratification by spatial neighborhood and selection of break down value in BPLTS do contribute a great deal to normalization. When the trimming fraction is set to zero, BPLTS degenerates into ordinary least squares (OLS). As shown in Figure 4.2A, the OLS doesn't make a good normalization. Overlapping size has a little contribution in this data set.

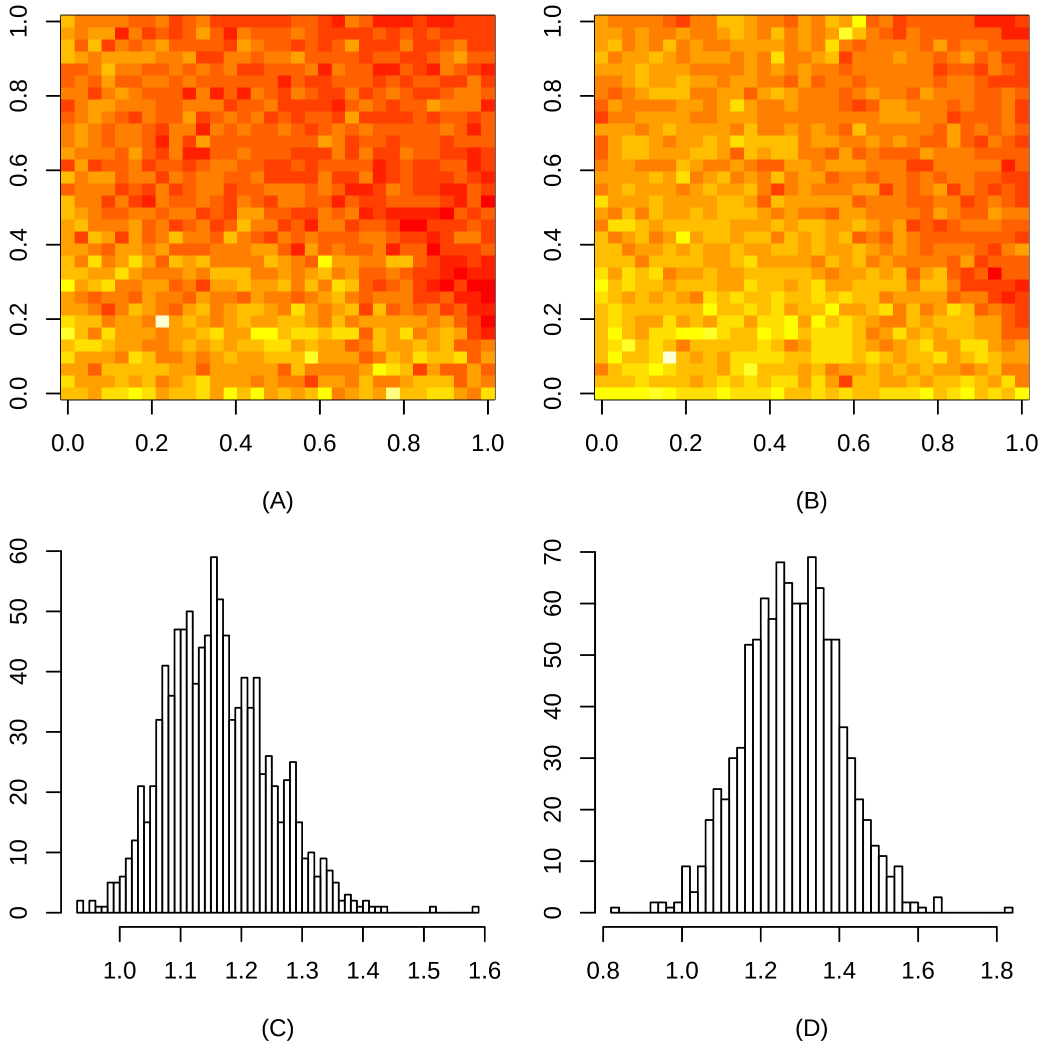


Figure 4.3: The slope matrices of two arrays show different spatial patterns in scale. The common reference is Array M. (A) Array P versus M; (B) Array N versus M. Their histograms are shown at bottom correspondingly in (C) and (D).

We then investigate the existence of spatial pattern. The HG-U95 chip has 640×640 spots on each array. We divided each array into sub-arrays of size 20×20. We run simple LTS regression on the target with respect to the reference for each sub-array. This results in an intercept matrix and a slope matrix of size 32×32, representing the spatial difference between target and reference in background and scale. We first take

Array M as the common reference. In Figure 4.3, the slope matrices of Array P and M are respectively shown in the subplots at top left and top right, and their histograms are shown in the subplots at bottom left and bottom right. Two quite different patterns are observed. Similar phenomenon exists in patterns of α. The key observation is that spatial patterns are array-specific and unpredictable to a great extent. This justifies the need of adaptive normalization.

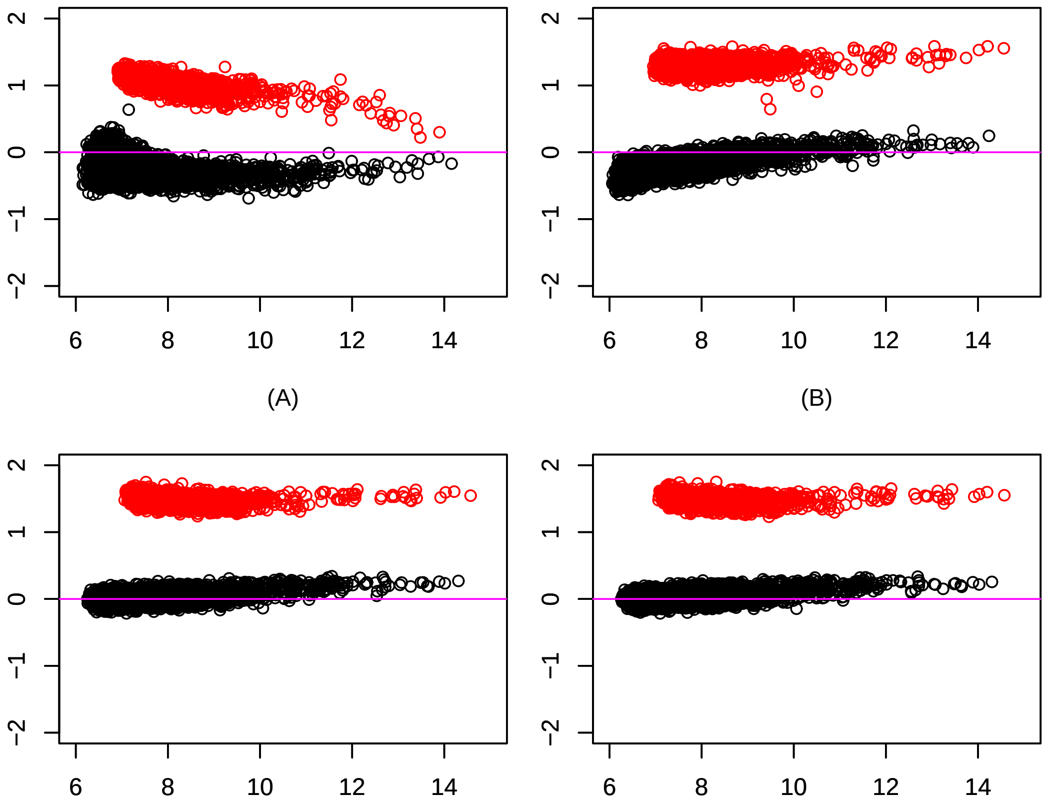


Figure 4.4: M-A plots of perturbed spike-in data set after SUB-SUB normalization. The x-axis is the average of log-intensities from two arrays. The y-axis is their difference after normalization. In the top four subplots, 20% randomly selected genes have been artificially up-regulated by 2.5 fold in Array Q, R, S, and T. The differentiated genes are marked red, and undifferentiated genes are marked black. The trimming fractions in the subplots are: (A) 0%; (B) 10%; (C) 30%; (D) 50%.

Perturbed Spike-in data SUB-SUB normalization protects substantial differentiation by selecting an appropriate trimming fraction in LTS. To test this, we generate an artificial data set with relatively large fraction of differentiation by perturbing the HG-U95 spike-in dataset. Namely, we randomly choose 20% genes and increase their corresponding probe intensities by 2.5 fold in the four arrays in the second group. We then run SUB-SUB normalization on the perturbed data set with various trimming fractions. The results are shown in Figure 4.4 for four trimming fractions, 50%, 30%, 10%, and 0%. AS shown, the normalization is satisfactory when the trimming fraction is above 30%, or 10% larger than the nominal differentiation fraction.

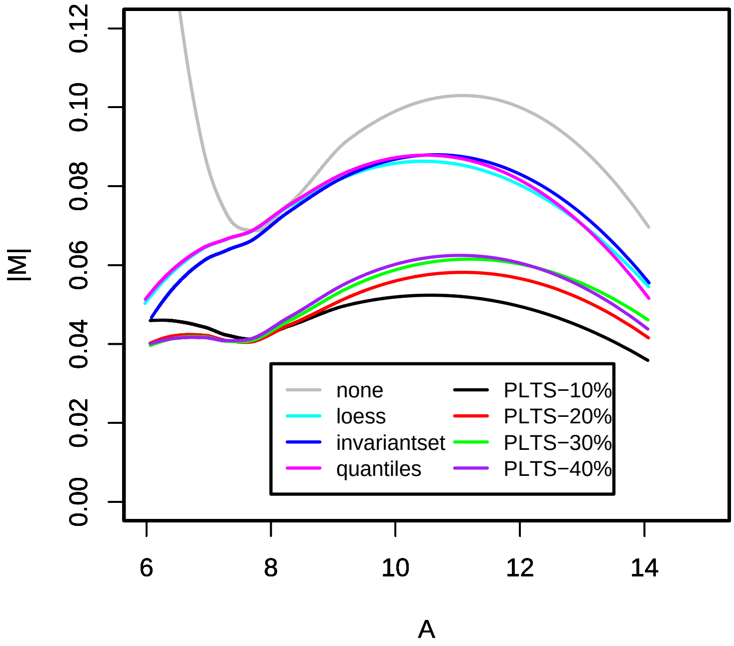


Figure 4.5: The lowess curves of $|M|$ versus A values by various normalization methods.

Variation reduction by sub-array normalization Stratification is a statistical technique to reduce variation. Sub-array normalization can be regarded as a way of stratification. We normalize the yeast array 2-121502 versus 2-121501 by various normalization methods available from Bioconductor. Since the two arrays are replicates, the difference between them is due to experimental variation. In the resulting M-A plots, we fit lowess [Cle79] curves to the absolute values of M, or $|M|$. These curves measure the variation between the two arrays after normalization, see Figure 4.5. The sub-array normalization achieves the minimal variation. As variation is reduced, signal to noise ratio is enhanced and power of significance tests is increased. Furthermore, a smaller trimming fraction in sub-array normalization reduces more variation. This suggests the following rules to determine trimming fraction for sub-array normalization : (1) One must select a trimming fraction that is large enough to ensure the robustness of the BPLTS estimation. (2) If the first rule is satisfied, a smaller trimming fraction is more efficient in terms of variation reduction.

References

[Aff01] *Affymetrix Microarray Suite User Guide Santa Clara, CA: Affymetrix. 5th edn*. 2001.

[BIS03] B. Bolstad, R. A. Irizarry, and M. Astrand T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, 19:185–193, 2003.

[CL05] C. Cheng and L. M. Li. Sub-array normalization subject to differentiation. *Nucleic Acids Research*, 37:55655573, 2005.

[Cle79] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.*, 74:829–836, 1979.

[CR88] R. J. Carroll and D. Ruppert. *Transformation and Weighting in Regression*. Chapman and Hall, New York, 1988.

[H9̈5] O. Hössjer. Exact computation of the least trimmed squares estimate in simple linear regression. *Computational Statistics and Data Analysis*, 19:265–282, 1995.

[Haw93] D. M. Hawkins. The feasible set algorithm for least median of squares regression. *Computational Statistics and Data Analysis*, 16:81–101, 1993.

[jH73] P. j. Huber. Robust regression: asymptotics, conjectures and monte carlo. *Ann. Stat.*, 1:799–821, 1973.

[KCM02] T. Kepler, L. Crosby, and K. Morgan. Normalization and analysis of DNA microarray data by self-consistensy and local regression. *Genome Biology*, 3:1–12, 2002.

[LDB$^+$96] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee M. Mittmann, C. Want, M. Kobayashi, H. Horton, and E. L. Brown. DNA expression monitoring by hybridization of high density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

[LH74] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice-Hall, 1974.

[Li03] L. M. Li. Blind inversion needs distribution (BIND): the general notion and case studies. *Science and Statistics: A Festschrift for Terry Speed*, 40:273–293, 2003.

[Li05] M. L. Li. An algorithm for computing exact leat-trimmed squares estimate of simple linear regression with constraints. *Comutational Statistics and Data Analysis*, 48:717–734, 2005.

[Lue87] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley Publishing Company, 1987.

[Qua] http://mathworld.wolfram.com/quarticequation.html.

[RD99] P. J. Rousseeuw and K. Van Driessen. *Computing LTS Regression for Large Data Sets.* Technical Report, University of Antwerp, 1999.

[RL99] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, New York, 1999.

[Rou84] P. J. Rousseeuw. Least median squares regression. *Journal of American Statistical Association*, 88:1279–1281, 1984.

[RY84] P. J. Rousseeuw and V. Yohai. *Robust regression by means of S-estimators.* Springer Verlag, New York, 1984.

[SHG$^+$02] I. A. Sidorov, D. A. Hosack, D. Gee, J. Yang, M. C. Cam, R. A. Lempicki, and D. S. Dimitrov. Oligonucleotide microarray data distribution and normalization. *Information Sciences*, 146:67–73, 2002.

[SL03] G. A. Seber and A. J. Lee. *Linear regression analysis. 2nd edn.* A John Wiley & Sons Publication, New York, 2003.

[SLEW] E. E. Schadt, C. Li, B. Ellis, and W. H. Wong. Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *Journal of Cellular Biochemistry*, Supplement 37.

[SLSW01] E. E. Schadt, C. Li, C. Su, and W. H. Wong. Analyzing high-density oligonucleotide gene expression array data. *Journal of Cellular Biochemistry*, 80:192–202, 2001.

[SP95] P. B. Stark and R. L. Parker. Bounded-variable least-squares: an algorithm and applications. *Computational Statistics*, 10:129–141, 1995.

[STA01] *STATA 7 Reference Manual A-G*. STATA Press, College Station, 1995-2001.

[TOR+01] G. C. Tseng, M. Oh, L. Rohlin, J. C. Liao, and W. H. Wong. Issues in cDNA microarray analysis: quality filtering,channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Research*, 29:2549–2557, 2001.

[V9́6] J. Á. Víšek. Sensitivity analysis of m-estimates. *Ann. Ins. Statist. Math.*, 48:469–495, 1996.

[V0́0] J. Á. Víšek. On the diversity of estimates. *Computational Statistics and Data Analysis*, 34:67–89, 2000.

[vHL02] S. van Huffel and P. Lemmerling. *Total Least Square and Errors-in-Variables Modelling*. Kluwer Academic Pulisher, Dordrecht/Boston/London, 2002.

[VR99] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S-PLUS. 4th edn.* Springer, New York, 1999.

[WBS99] H. P. William, P. F. Brian, and A. T. Saul. *Numerical Recipes in Fortran. 2nd edn.* Press syndicate of the university of cambridge, 1999.

[Wei] E. W. Weisstein. http://mathworld.wolfram.com/leastsquaresfitting perpendicularoffsets.html.

[WJJ+02] C. Workman, J. Jensen, H. Jarmer, R. BerkaAND L. Gautier, B. Nielsen, H. Saxild, C. Nielson, S. Brunak, and S. Knudsen. A new nonlinear normalization method for reducing variability in DNA microarray experiments. *Genome Biology*, 3:1–16, 2002.

[YDLS01] Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed. Normalization for cDNA microarray data. *Microarrays: Optical Technologies and Informatics*, 4266, 2001.

[Yoh87] V. J. Yohai. High breakdown point and high efficiency robust estimates for regression. *Annals of Statistics*, 1:253–270, 1987.

[ZAZL01] A. Zien, T. Aigner, R. Zimmer, and T. Lengauer. Centralization: a new method for the normalization of gene expression data. *Bioinformatics*, 17:323–331, 2001.