AN APPLICATION OF SIMPLICIAL INTERCEPT DEPTH (SID) METHOD

FOR FITTING LINEAR MODELS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Zhongxing Sun

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

April 2014

Fargo, North Dakota

North Dakota State University

Graduate School

Title

AN APPLICATION OF SIMPLICIAL INTERCEPT DEPTH (SID) METHOD FOR LINEAR FITTING

By

Zhongxing Sun

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Gang Shen

Chair

Dr. Seung Won Hyun

Dr. Shaobin Zhong

Approved:

4/23/2014	Dr. Ronda Magel
Date	Department Chair

ABSTRACT

This paper presents an application based on the Simplicial Intercept Depth method introduced by Liu (2004). We use this method to get the best linear fit of the phenotypic data for spot blotch resistant reaction of two different barley groups. The Simplicial Intercept Depth method is generalized by Simplicial Depth, also proposed by Liu in 1990. It provides a robust way for data analysis when outliers appear. In this paper, we use the Bootstrapping method, which is introduced by Bradley Efron (1979), to resample from the original dataset to get a distribution of the estimates. We also compare the SID with least squares regression and the Theil-type estimate which introduced by Shen (2009). The result shows that the SID is a robust method for estimating the coefficients of the linear regression model.

ACKNOWLEDGMENTS

During the process of writing my paper, I obtained a huge amount of help and encouragement from my advisor, Dr.Gang Shen. Thanks a lot for your careful review and effective suggestions. Also, I would also like to thank my whole committee, Dr.Seung Won Hyun and Dr.Shaobin Zhong. At last, I would like to thank everyone that had helped me in my past few years at NDSU.

I would like to express my deep thankfulness to my dear family members, they listened to me when I was in confusion, and talked to me when I felt down. These all mean a lot to me. Thanks all my friends, for never making me feel alone.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The least squares method is the most common statistical method used for analyzing related variables. It performs perfectly when the distribution of the dataset is normal. But if the outliers arise, which happens a lot in real data, this method usually delivers unsatisfied results. The Simplicial Intercept Depth (SID) method, introduced in section 2.2, provides a geometric way to find the best linear fit for a given dataset. It considers each pair of input and output variables as a single point in the plane, and finds the best fitted line formed by two of those points. The SID method is proposed by Liu (2004) based on the notion of Simplicial Depth by Liu (1990). This SID method shows robustness against outliers.

In section 3, we provide the phenotypic data for two different groups of inoculated barley. Spot blotch is a barley disease due to the fungus named *Cochlioblus sativus*. This disease is around all the places that barley is grown. It is one of the most important kinds of leaf disease of barley in North Dakota. It may cause significant yield losses in the warm temperature and moist humid climates. Under favorable conditions, the diseases could spread rapidly. The infections present as dark blotches, then the spots will spread to form dead dark patches on the leaves. The data shows the disease resistant reaction levels from 1 to 9, where 1 represents the most resistant reaction while 9 is the most susceptible one. In this paper, we are trying to use SID to estimate the difference between the two groups.

When we estimate the coefficients of the linear fit by SID, we use the Bootstrapping method, introduced in section 2.3, to resample the data. Thus we can construct a distribution of the estimated coefficients and then confidence intervals. The best linear fit is formed by the mean of the coefficients.

In order to show the robustness of the SID method, in section 5 we present some examples to compare the results with the least squares method and a Theil-type estimate (by Shen, 2009). We insert some significant outliers in the simulated data. The fitted lines by SID and the Theil-type estimate still pass through the bulk of the dataset, while the least squares method is attracted by the outliers. Though the SID method is computational expensive, it provides an alternative way to get the robust linear fit.

2. METHODOLOGY

2.1. Simplicial Depth

The notion of Simplicial Depth (SD) was proposed by Liu(1990). The idea is trying to find the probability that a point is inside a random simplex that formed by the independent observations from the given dataset. Statistical data depth started to play an increasing role in data analysis after that. The data depths represent how deep or central a given point is relative to the distribution or to the given data cloud. Liu introduced the Simplicial depth as a depth function that is robust and affine invariant.

Definition: Given a distribution F in R^p , the simplicial depth of the point x is the probability that x is inside a random simplex in R^p :

$$SD(F;x) = P_F(x \in S[X_1, \dots, X_{p+1}])$$

where $S[X_1, \dots, X_{p+1}]$ is a closed simplex formed by p + 1 random points from the distribution F.

If we consider F as a distribution on R^2, the simplicial depth function could be easily written as

$$SD = P_F(x \in \Delta(X_1, X_2, X_3))$$

where X_1, X_2, X_3 are any three independent points form F. It is easy to be understood through geometry. It means the probability the point *x* inside the triangles formed by the dataset is the so-called simplicial depth of x. The point near the center of the data cloud should be contained in more triangles while the point away from the center should be contained in less. In other words, the function SD should have higher values for the deeper points and should approach zero as the point leaving the center.

If the sample size is n, we could estimate the above equation as

$$\widehat{SD} = \binom{n}{3}^{-1} \sum_{*} I\,(x \in \Delta(X_i, X_j, X_k))$$

where * indicates that $1 \leq i < j < k \leq n$ and I( ) is the indicator function. To visualize the sample depth, we may imagine painting a layer with unit darkness on the region corresponding to each triangle $\Delta(X_i, X_j, X_k)$, until all the $\binom{n}{3}$ triangles are painted. The resulting degree of darkness thus represents the shape of the depth function. The points with larger depth values should show darker in color.

Here we can get some properties of the Simplicial Depth function,

(i) *Affine Invariance*. Assuming A( ) is an affine transformation function, the depth of point x respect to distribution F is equal to the depth of point A(x) respect to the distribution A(F). That is

$$SD(F; x) = SD(A(F); A(x))$$

(ii) *Maximality at Center*. The depth of the center point p should be always equal or greater than any other point q in the dataset.

$$SD(F; p) \geq SD(F; q)$$

(iii) *Monotonicity Relative to Deepest point.* The depth of any point between center point p and any other point q should be equal or greater than the depth of point q.

$$SD(F; p + r(q - p)) \geq SD(F; q)\ ; \quad r \in [0, 1]$$

(iv) *Vanishing at Infinity.* If the point q is far away from the data cloud, it is an outlier, and the depth of q becomes zero. That’s the motivation of using this method to detect outliers.

$$\lim_{||q||\to\infty} \mathrm{SD}(\mathrm{F};\mathrm{q}) = 0$$

Simplicial Depth mainly deals with the shape of data. It can be thought of as a measure of how well a point characterizes a dataset. This provides an alternative method to classical statistical analysis.

2.2. Simplicial Intercept Depth (SID)

When we try to study some related variables, the first and most commonly used statistical method that comes to mind is always linear fitting. Among all the existing linear fitting methods, the least squares regression has been used most extensively, for its mathematical convenience as well as for the optimality properties under normally distributed errors. However, it is less satisfactory when the error distributions are heavy-tailed or when the outliers are present, while SID shows more robustness in comparison.

SID was also introduced by Liu (2004). It may be viewed as a generation of simplicial depth. Unlike the usual concept of depth defined with respect to the point in a multivariate data cloud, the SID is presented as the depth for lines or for hyperplanes in general. It measures the depth of lines instead of points.

Although the SID method is suitable for the cases in any dimension, in this paper, we only focus on the multiple linear models in tow dimension,

$$y_i = \beta_0 + \beta_1 x_i + e_i \qquad \text{for} \qquad i = 1, \ldots\ldots, n$$

where n is the sample size, x_i is the i-th input variable, and y_i is the i-th output variable or response variable. e_i is the independent error and assumed to have mean zero and variance σ^2. β_0 and β_1 are to be estimated from the given dataset.

In R^2, the depth of a line is simply calculated as the average length ratio of the line intercept within each triangle to its longest edge, over all the triangles formed by the dataset. The depth here, in other words, presents how deep the line cuts through all the triangles. The SID value is 1 when all the data points fall on the line, and it decreases as data points move away. The line with the highest SID value will be considered as the best fit for the distribution.

Definition: For a given line L_{ij}, the criterion function of simplicial intercept depth is

$$\mathrm{SID}(L_{ij}) = \binom{n}{3}^{-1} \sum_{k=1}^{\binom{n}{3}} \{\frac{d_k(i,j)}{m(\Delta_k)}\},$$

for i, j = 1,……, n, i $\neq$ j

where n is the sample size. L_{ij} is the line formed by the 2 given points p_i, p_j from the dataset. Then the number of the lines L_{ij} is $\binom{n}{2}$. $d_k(i,j)$ is the intercept of L_{ij} within the triangle Δ_k. Δ_k is formed by any three points p_a, p_b and p_c from the data. $m(\Delta_k)$ is the longest side of Δ_k. The number of the triangles Δ_k is $\binom{n}{3}$. A larger SID value for L_{ij} means that the line L_{ij} cuts deeper into more triangles, which implies the line describe the dataset better.

There are two degenerate cases. One is when two vertices of the triangle are at the same position, the triangle will become a line. In this case, the ratio is 1 if the line L_{ij} overlaps with the line representing the triangle, 0 otherwise. Another case is that all the three vertices are same, then the ratio is defined to be 1 if the coincided vertex is on the line L_{ij}, 0 otherwise.

In this function, the line L_{ij} with the largest SID value is going to be considered as the best linear fit for the given dataset. That SID value of the fitted line can be considered as a robust condition of the coefficient of determination defined in the least squares regression. It provides a natural measurement of goodness fit for the line L_{ij}. It’s not hard to find that the

SID value of the fitted line is 1 if and only if every data point is on this line, and it will decrease when the data points scatter away from it.

Liu also showed the rotation and reflection invariance property of SID in his paper. Denote D_n ={p_1,…, p_n} be the dataset, where $p_i = (x_i, y_i)^T$. Consider H as a 2*2 orthonormal matrix, that is H^{t}H = HH^t = I. Let $\tilde{p}_i = Hp_i$, then $\widetilde{D_n}$ is the orthonormal transformation of the original data. So we can get

$$SID_{D_n}(L) = SID_{\tilde{D}_n}(\tilde{L})$$

where L is the given line with coefficients β_0, β_1. $\tilde{L}$ is the line with coefficients $\tilde{\beta}_0$, $\tilde{\beta}_1$, where $(\tilde{\beta}_0, \tilde{\beta}_1)^T =$ H$(\beta_{0,}\beta_1)^T$. $SID_{D_n}(L)$ implies the SID value of the line L with respect to the dataset D_n, while $SID_{\tilde{D}_n}(\tilde{L})$ is the SID value of the line $\tilde{L}$ with respect to the dataset $\tilde{D}_n$.

This suggests SID treats the input and the output variable symmetrically, and it is not usually shared by other regression methods. When the roles of the two variables as input and output variables are reversed, reflection invariance keeps the linear relationship between them unaffected.

2.3. Bootstrapping

Bootstrapping is a useful statistical method introduced by B. Efron(1979) for assigning measures of accuracy to sample estimates. It allows estimating the sampling distribution of almost any statistic using very simple methods. The basic idea of bootstrapping is to speculate about a population from sample data. This can be conducted by constructing a number of resamples, the number is usually 1,000 or 10,000. A moderate size resample usually contains 20 or 50 individuals. Each of these resamples is generated by random sampling with replacement from the given dataset. Bootstrapping may also be used to construct hypothesis tests.

In this paper, we only have one dataset. When we get the fitted model through SID (Section 4), we can't test the significance of the coefficients. Therefore, we use the bootstrapping to create a large number of datasets and compute the coefficients on each of these subsets. Thus we get a distribution of the coefficients. Also, we use bootstrapping to avoid the computational expensive of SID.

2.4. The Theil-type Estimate

This method is similar as the idea of SID. They both transfer the problem of regression to the one of location. Theil estimate is also affine invariance. In this paper, to simplify, we only consider the case in two-dimension. Therefore, the linear regression model will also be similar as (3)

$$y_i = \beta_0 + \beta_1 x_i + e_i \qquad \text{for} \qquad i = 1,\ldots\ldots,n$$

where n is the sample size, x_i is the i-th input variable or so-called deterministic in this case, and y_i is the i-th output variable. e_i is independent error with zero mean and variance σ^2. β_0 and β_1 are the estimators.

For the given x_i and x_j, we have corresponding y_i and y_j,where i and j are from 1 to n and i is different from j. There are b_i and b_j representing the coefficients of them respectively. In another word, $b_i = (\beta_{0i}, \beta_{1i})^T$ and $b_j = (\beta_{0j}, \beta_{1j})^T$. Then this Theil-type method gives us the estimate of β which minimizes the function

$$D_n(\beta) = \frac{1}{\binom{n}{2}\binom{n-2}{2}} \sum_i \sum_j \left|K_{ij}(\beta)\right|,$$

where

$$K_{ij}(\beta) = det\begin{pmatrix} 1 & 1 & 1 \\ b_i & b_j & \beta \end{pmatrix}.$$

This method, already proven by Shen (2009), provides more robust results than the least squares estimate.

3. DATA DESCRIPTION

In our work, the data was collected from the department of plant pathology of NDSU. It was read from two groups of inoculated barley lines. For group 1, each line with three replicates was planted in greenhouse room 1. The barley in group 2 was planted in greenhouse room 2 as four replicates.

The inoculated seedlings were read for disease resistant reaction at about ten days after the inoculation with a 1-9 scale method developed by Fetch and Steffenson (1999), where 1 represents the most resistant reaction and 9 represents the most susceptible reaction. Three independent inoculated plants were rated for each replicate and the averaged disease data were used to represent the reaction.

We had 1012 barley lines in group 1, and 1050 lines in group 2. We can see from Table 1 and Figure 1 that there is a difference between the mean of the two groups, but it is unclear if the difference is significant. The scatter points in Figure 1 are the outliers.

Table 1. Summary of data in group 1 and group 2.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Group 1	2.2	6.1	6.8	6.643	7.4	8.7
Group 2	3.3	6	6.5	6.418	6.9	8

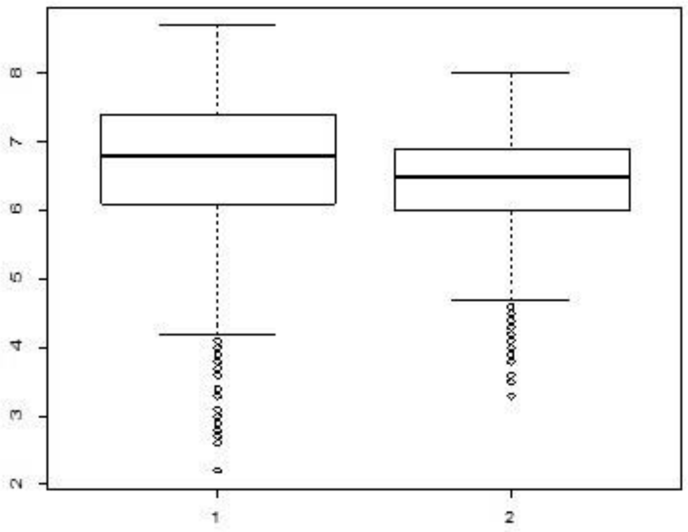

Figure 1. Boxplot for group 1 and group 2.

4. MODEL FITTING

The SID is a robust regression method that ignores the cluster of outliers, and the fitted line passes through the bulk of the major data cloud. This analysis in our paper is trying to find if there is a difference of the disease resistant reactions between the two groups. In this case, it is actually a simplified situation of SID in two-dimension. We use the disease resistant reaction levels as the response variables and consider 0 and 1 as input variables for group 1 and group 2, respectively. The input and output variables can be inversed, which was introduced in section 2.

However, our goal is to find the best linear fit by using the SID value to explain the dataset. The line with the largest SID value is estimated as the best fit.

We here assume the model as

$$y_i = \beta_0 + \beta_1 x_i + e_i \qquad \text{for} \qquad i = 1, \ldots, n,$$

where n is the sample size, which equals to the total of the barley lines in both groups. x_i is the i-th input variable, which equals to either 0 or 1. And y_i is the i-th output variable representing the disease resistant level with value from 1 to 9. e_i is the independent error and assumed to have zero mean and variance σ^2. β_0 and β_1 are the intercept and slope that need to be estimated from the dataset.

We define the whole dataset as $Q_n = \{q_1, \ldots, q_n\}$, where $q_i = (x_i, y_i)^T$. Then we consider 2 subsets $U_h = \{u_1, \ldots, u_h\}$, $V_l = \{v_1, \ldots, v_l\}$ for the two groups, respectively. Where $h = 1, \ldots, n_h$; $l = 1, \ldots, n_l$, then $u_h = (0, y_h)^T$, $v_l = (1, y_l)^T$.

In this case, when we are getting the pool of the lines using SID, the line x = 0 and x = 1 are useless since we are trying to compare U_h and V_l. Therefore, we only choose the line that passes through each pair of points u_h and v_l. That means each line here is formed by one point in group 1 and another point in group 2. We denote the pool of all the $\mathrm{n}_h \times \mathrm{n}_l$ lines

$$L_{hl} \equiv \{line_{hl} : h = 1, \ldots, \mathrm{n}_h; \quad l = 1, \ldots, \mathrm{n}_l\}.$$

Let Δ denote the collection of triangles. If all the three vertices of the triangle are from the same subset, then this triangle will be represented by a line, either x = 0 or x = 1. Recall that SID is the ratio between the length of the line's intercept within the triangle to the length of the longest side of that triangle. The SID value of any line with that triangle is always 0 since the length of the intercept is 0. Hence we define the subset of triangles with two vertices from U_h and another one from V_l as

$$\Delta \mathrm{u} \equiv \{\Delta u_1, \ldots, \Delta u_{kh}\}$$

where $kh = \binom{n_h}{2} \times n_l$. The format of each triangle in this subset should be like $\Delta(u_i, u_j, v_k)$. Let another subset for the triangles with one vertex from U_h and two from V_l as

$$\Delta \mathrm{v} \equiv \{\Delta v_1, \ldots, \Delta v_{kl}\}$$

where $kl = \binom{n_l}{2} \times n_h$. Each triangle here is represented as $\Delta(u_i, v_j, v_k)$. Therefore, the total number of the triangles we get in this research is

$$\mathrm{nt} = \mathrm{kh} + \mathrm{kl} = \binom{n_h}{2} \times n_l + \binom{n_l}{2} \times n_h$$

Then for a given line $line_{hl}$ in the pool L_{hl} and a triangle Δ_t in Δ, we can check if they intersect. If any point of intersection with the line representing any side of the triangle Δ_t has an X-axis position between 0 and 1, then the line $line_{hl}$ passes through the triangle Δ_t. Let $d_t(h, l)$ denote the distance between the intersection points. Then we can get the SID value of $line_{hl}$ using the following equation

$$\mathrm{SID}(line_{hl}) = \frac{1}{nt}\sum_{t=1}^{nt}\{\frac{d_t(h,l)}{m(\Delta_t)}\}$$

where $m(\Delta_t)$ is the length of the longest side of Δ_t.

There are several cases when we calculate $d_t(h,l)$:

(i) If $line_{hl}$ has the same coefficients with one of the lines representing the side of the triangle Δ_t, then $d_t(h,l)$ equals to the length of that side.

(ii) Clearly $d_t(h,l)$ = 0 if the line does not pass through the triangle, which means no intersection point has an X-axis value between 0 and 1.

(iii) If there are two intersections' X-axis position between 0 and 1, then $d_t(h,l)$ is the distance between those two points.

(iv) If there is only one intersection point's X-axis position between 0 and 1, and the triangle Δ_t is in the subset Δu, then $d_t(h,l)$ is the distance between that point and the intercept point of $line_{hl}$.

(v) If there is only one intersection point's X-axis position between 0 and 1, and the triangle Δ_t is in the subset Δv, then $d_t(h,l)$ is the distance between that point and the point v_l.

We calculate all the SID value for every single line in the pool L_{hl}. The best linear fit for the dataset Q_n is the line with maximum SID value. We denote the estimated coefficients as β_0^* and β_1^*.

In this research, we are using Bootstrapping method to construct U_h and V_l with resampling from the original datasets with the resample size 50 and 51, respectively. 1000 bootstrapping samples are created, we can get 1000 pairs of β_0^* and β_1^*. Their average values are used as the fitted coefficients for our model. Their summary is shown in Table 2. Then we get the fitted model

$$y_i = 6.72 - 0.6288x_i \qquad \text{for} \qquad \text{i} = 1, \dots, \text{n},$$

The 95% confidence interval for β_0^* is (5.2, 7.6), and for β_1^* is (-1.5, -0.1). From this result, we could find that the plants in group 2 shows more resistance to the spot blotch disease than those in group 1.

Table 2. Summary of estimated coefficients by SID.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Beta0	4.7	6.4	6.8	6.72	7.2	8.0
Beta1	-2.1	-1.0	-0.6	-0.6288	-0.4	1.0

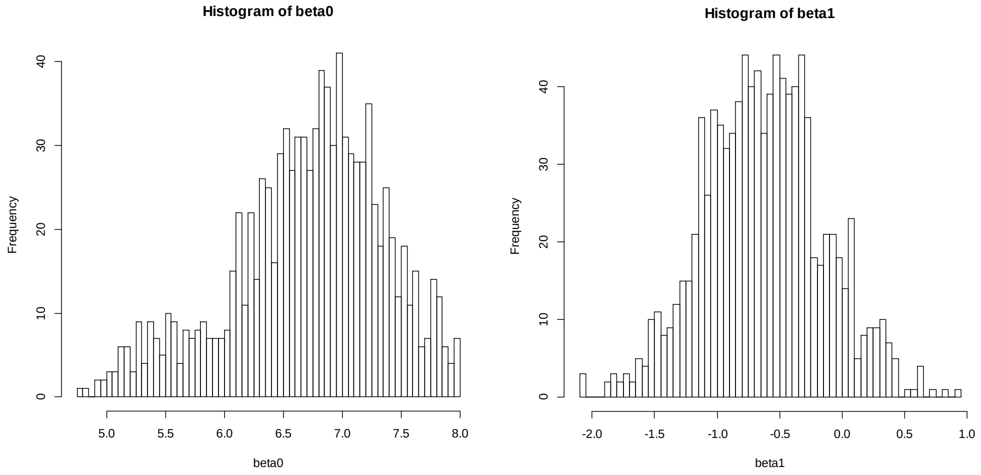

Figure 2. Histogram of estimated β_0^* and β_1^*.

The least squares regression provides the fitted linear model as $y_i = 6.64 - 0.22x_i$, with 95% confidence interval (6.586, 6.700) for β_0^* and (-0.304, -0.147) for β_1^*, while the linear model fitted by the Theil-type estimate is $y_i = 6.79 - 0.49x_i$. The 95% confidence intervals for β_0^* and β_1^* are (6.502, 7.086) and (-0.844, -0.134), respectively.

The results of all the methods are significant because the sample size is large enough. Note that when we use the Theil-type estimate to fit the model, we also use bootstrapping to get the resamples to avoid computational expense.

5. ROBUSTNESS COMPARISONS

To maintain the consistency of this paper, here we still use the assumption that the linear model is

$$y_p = \beta_0 + \beta_1 x_p + e_i \qquad \text{for} \qquad \text{p} = 1,\ldots\ldots,\text{n}$$

We apply the three methods, least squares estimate, the Theil-type estimate and SID, to the dataset we created in four different cases. In each case, we give specified values for β_0 and β_1 and a certain distribution for the error e_i. Let e_i generalize from the given distribution, then we can get the simulated response values y_p.

Denote D = $\{d_p = (x_p, y_p)^T\}$ as the dataset we created. Let n = 100, W_I = $\{w_i = (x_i, y_i)^T\}$ and W_J = $\{w_j = (x_j, y_j)^T\}$ be the two subsets of the variables, where i, j = 1,…, 50. Then we define $x_i = 0$ and x_j = 1. It is easy to find that $y_i = \beta_0 + e_i$ and $y_j = \beta_0 + \beta_1 + e_j$. And then we create three significant outliers in W_I, as $y_1 = y_2 = y_3$ = -100.

For the least squares method, the regression is simply the linear model for the response variable y_p with respect to the input variable x_p. We can get the fitted model, standard error for the estimated coefficients and the test score for it. Then we conduct the 95% confidence interval for the coefficients by using asymptotic normality.

As for the model fitting method of SID we mentioned in section 4, we consider the response variables y_i in W_I as the elements in the subset U_h, and let all the y_j in W_J be the variables in V_l. Then we can use SID to get our fitted linear model with the resample size 20, and find out the 95% interval for the bootstrapping coefficients.

Similar to SID, the Theil-type estimate considers $b_i = (\beta_{0i}, \beta_{1i})^T$ and $b_j = (\beta_{0j}, \beta_{1j})^T$ as the coefficients with respect to each w_i and w_j from the subsets W_I and $\text{W}_{\text{J},}$

respectively. Then we fit the b_i and b_j in the function $D_n(\beta)$ mentioned in section 2.4 to find the minimizer β. This simulation also runs 1000 times to get the distribution of β and the 95% confidence intervals.

The four cases include the situations that the given slope is both positive or negative, also the independent errors both from normal distribution or student-t distribution. To be fair, we set the variance of the errors to be the same in all the cases. The results are shown as follows.

Case 1: β_0 = 1, β_1= -2, e_i iid from n(0, 1). Then y_i has mean 1 and variance 1, y_j has mean -1 and variance 1.

The fitted lines by the three methods are shown respectively as below:

$$\begin{aligned} \text{LS:} &\quad \mathrm{y} = -4.903 + 3.796\mathrm{x} \\ \text{SID:} &\quad \mathrm{y} = 1.027 - 1.969\mathrm{x} \\ \text{Theil:} &\quad \mathrm{y} = 0.885 - 1.889\mathrm{x} \end{aligned}$$

Table 3. 95% confidence intervals for case 1.

	LS	SID	Theil
β_0	(-9.721, -0.080)	(0.6120, 1.436)	(0.691, 1.079)
β_1	(-3.053, 10.596)	(-2.494, -1.457)	(-2.096, -1.682)

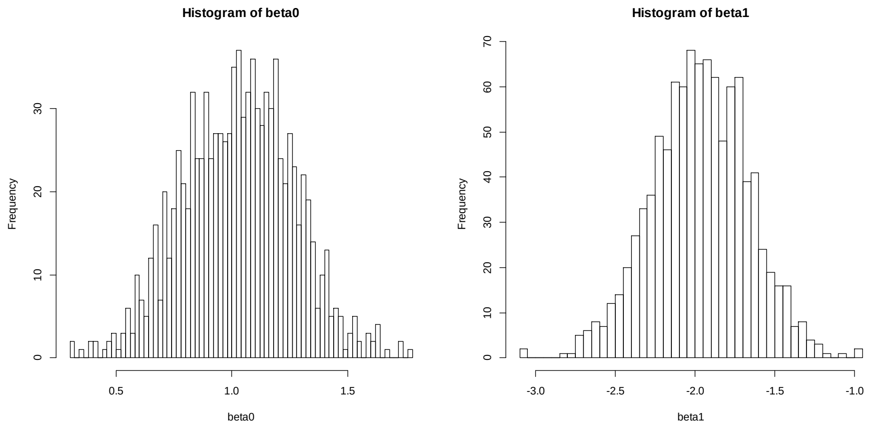

Figure 3. Coefficients estimated by SID for case 1.

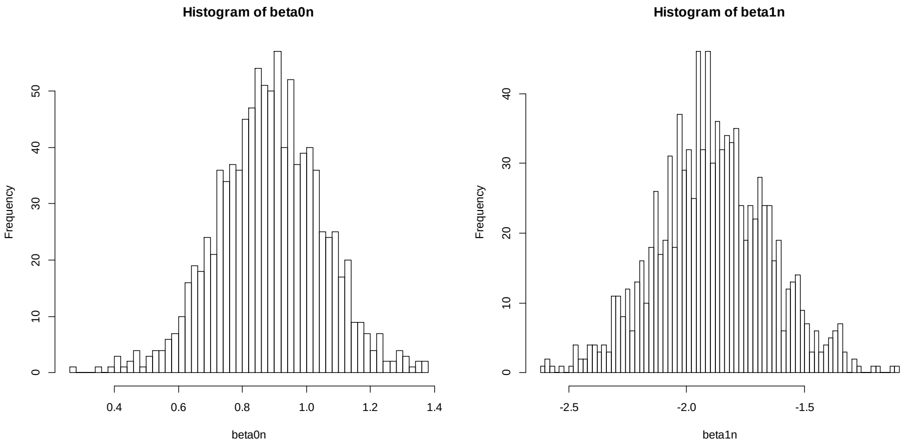

Figure 4. Coefficients estimated by Theil-type for case 1.

Case 2: β_0 = 1, β_1= -2, e_i iid from $\frac{t_4}{\sqrt{2}}$. Then y_i has mean 1 and variance 1, y_j has mean -1 and variance 1.

The fitted lines are

$$\begin{aligned} \text{LS:} &\quad \mathrm{y} = -4.826 + 3.827\mathrm{x} \\ \text{SID:} &\quad \mathrm{y} = 1.173 - 1.917\mathrm{x} \\ \text{Theil:} &\quad \mathrm{y} = 0.914 - 1.915\mathrm{x} \end{aligned}$$

Table 4. 95% confidence intervals for case 2.

	LS	SID	Theil
β_0	(-9.652, -0.000)	(0.561, 1.763)	(0.685, 1.143)
β_1	(-2.999, 10.654)	(-4.297, -0.379)	(-2.131, -1.699)

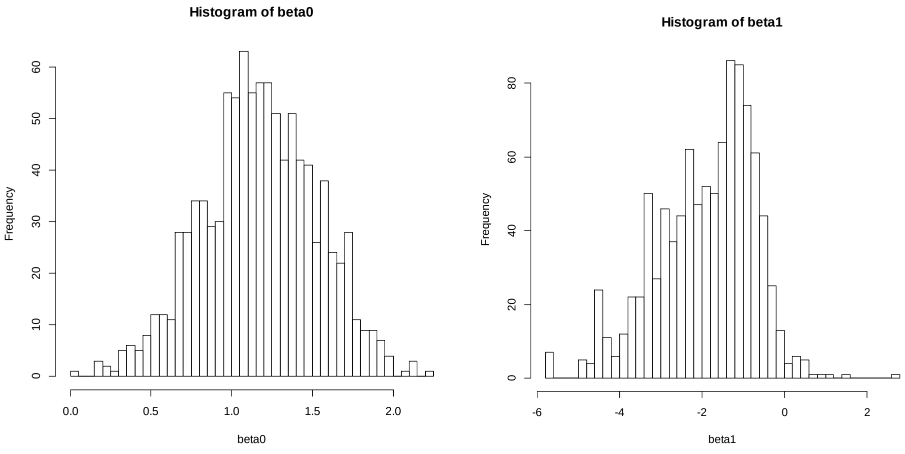

Figure 5. Coefficients estimated by SID for case 2.

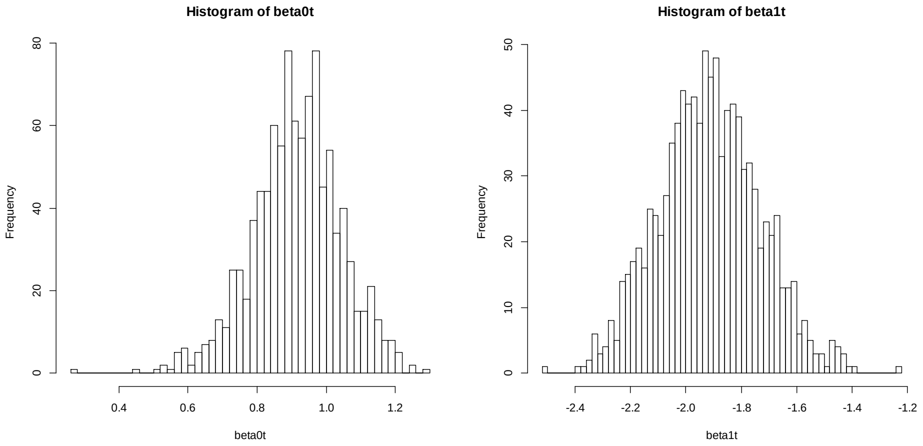

Figure 6. Coefficients estimated by Theil-type for case 2.

Case 3: β_0 = 1, β_1= 3, e_i iid from n(0,1). Then y_i has mean 1 and variance 1, y_j has mean 4 and variance 1.

We can get the fitted lines as these

$$\text{LS:} \quad y = -4.962 + 8.890x$$
$$\text{SID:} \quad y = 1.022 + 3.018x$$
$$\text{Theil:} \quad y = 0.904 + 3.092x$$

Table 5. 95% confidence intervals for case 3.

	LS	SID	Theil
β_0	(-9.782, -0.142)	(0.621, 1.439)	(0.615, 1.193)
β_1	(2.071, 15.709)	(2.611, 3.412)	(2.669, 3.515)

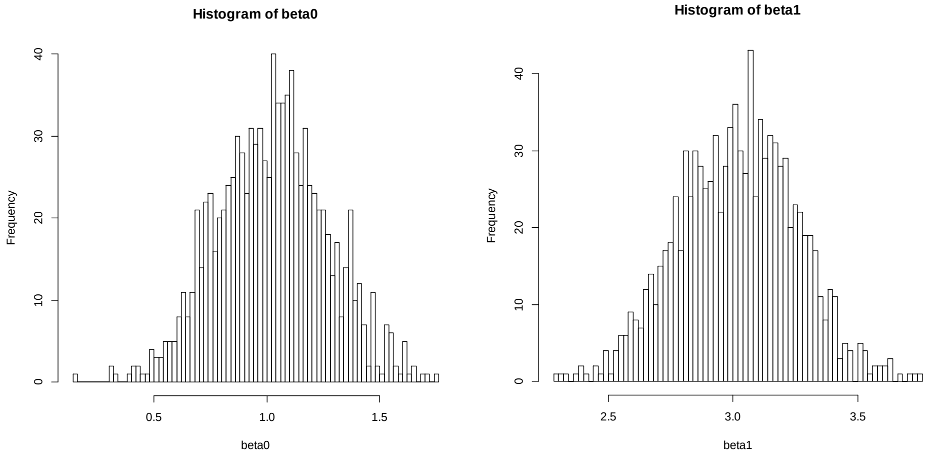

Figure 7. Coefficients estimated by SID for case 3.

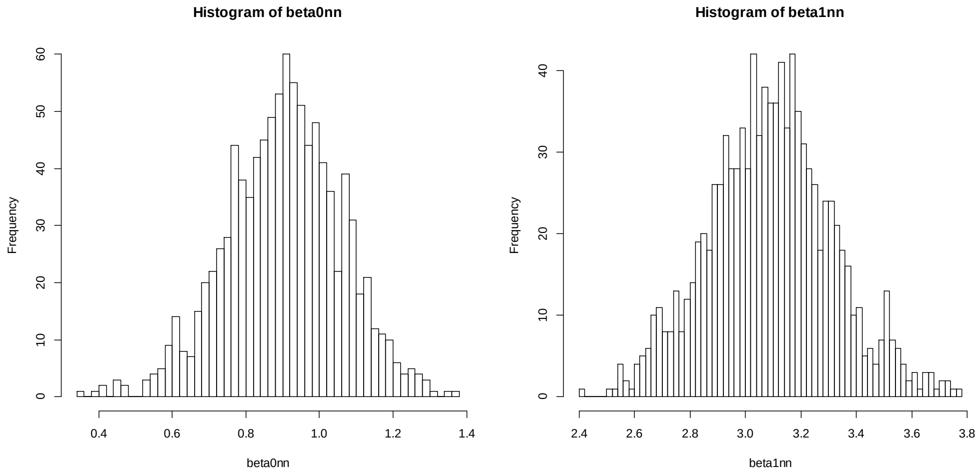

Figure 8. Coefficients estimated by Theil-type for case 3.

Case 4: β_0 = 1, β_1= 3, e_i iid from $\frac{t_4}{\sqrt{2}}$. Then y_i has mean 1 and variance 1, y_j has mean 4 and variance 1.

The results of the three methods are

$$\begin{aligned} \text{LS:} &\quad \mathrm{y} = -5.434 + 9.377\mathrm{x} \\ \text{SID:} &\quad \mathrm{y} = 0.973 + 3.110\mathrm{x} \\ \text{Theil:} &\quad \mathrm{y} = 0.917 + 3.080\mathrm{x} \end{aligned}$$

Table 6. 95% confidence intervals for case 4.

	LS	SID	Theil
β_0	(-10.236, -0.631)	(0.331, 1.635)	(0.723, 1.111)
β_1	(2.584, 16.170)	(2.431, 3.790)	(2.857, 3.303)

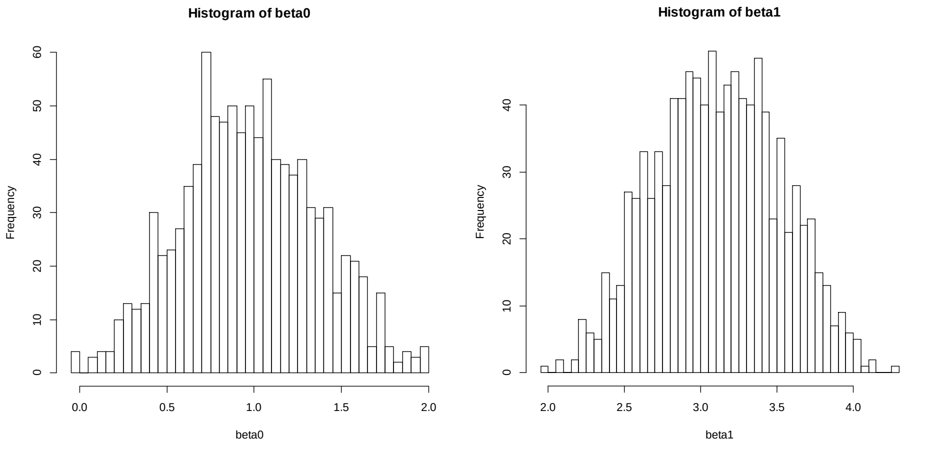

Figure 9. Coefficients estimated by SID for case 4.

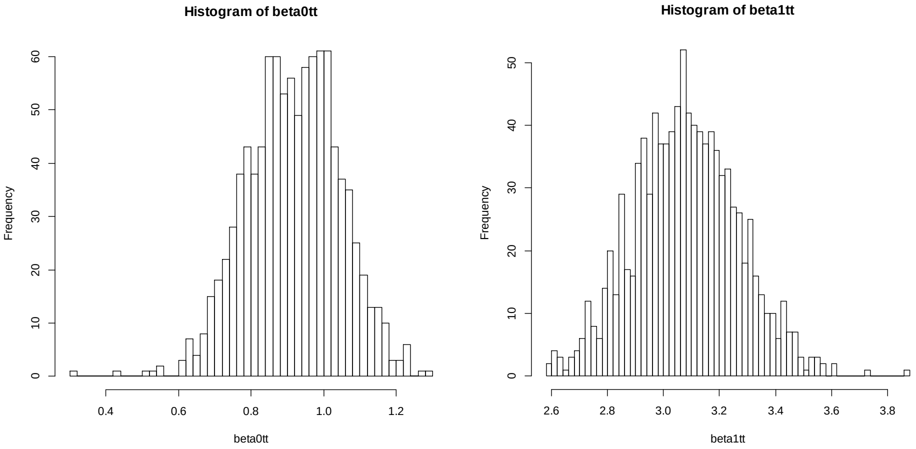

Figure 10. Coefficients estimated by Theil-type for case 4.

From the four different cases shown as above, we can observe that the lines fitted by least squares are affected by the outliers, and the regression models are actually not significant. Also, we could see from Figure 3 to 10 that the results of SID and the Theil-type method perform similarly. The estimated values of the two methods are very close to the assumptions. We could say they have almost no influence from the outliers.

6. DISCUSSION

In this research, we have used the SID method to find the best linear fit for the data of disease resistant reaction of spot blotch of barley. The result shows the group of barley lines planted in group 2 has more resistance to the disease. In this case, all the methods performed well because the sample size is large enough. In the simulation part, we reduced the sample size. The SID method shows less affect from the outliers than the common least squares method. It seems that SID gives more robust results in a small sample size. SID analyzes the numerical data in a geometric way. It provides an alternative to classical statistical analysis, and visualizes the problems while many measures are geometric in nature.

The resample size is set around 50 for each subset when we construct the bootstrapping samples to avoid computational expense. Imagine if we enlarge the size to 100, the number of lines in the pool will be 10,000, and the number of triangles will become about 1 million. We would need to calculate the SID of each the 10,000 lines with respect to all those triangles, and then bootstrapping for 1000 times. The calculation would need a really long time to complete. That’s a point that needs to be focused on in the future research. However, the SID method still works well on datasets with small sample size and provides a robust way to fit linear models against outliers.

REFERENCES

[1] Liu, R., Singh, K. and Teng, J. (2004). *Linear fitting by simplicial intercept depth (SID): reflection invariance and robustness.* Statistica Sinica **14,** 431-448

[2] Liu, R. (1990). *On a notion of data depth based on random simplices*. Ann. Statist. **18**, 405-414

[3] Katina, S., Wellmann, R. and Muller, C. (2008). *Simplicial depth estimators and tests in examples from shape analysis.* Tatra Mt. Math. Publ. **39**, 95-104

[4] Shen, G. (2009). *Asymptotics of a Theil-type estimate in multiple linear regression.* Statistics and Probability Letters **79**, 1053-1064

[5] Serfling, R. (2004). *Depth functions in nonparametric multivariate inference.* DIMACS.

[6] Burr, M., Rafalin, E. and Souvaine, D. (2003). *Simplicial depth: an improved definition, analysis, and efficiency for the finite sample case.* DIMACS

[7] Pederson, V. and Mcmullen, M. (1985). *Spot blotch of barley.* North Dakota State University.

[8] Ali, S., Wang, R. and Zhong, S. (2012). *Phenotypic reactions of 1050 barley accessions to a new spot blotch pathotype of Cochliobolus sativus.* NDSU Plant Pathology.

[9] Efron, B. (1979). *Bootstrap methods: Another look at the jackknife.* Ann. Statist. **7**, 1-26

[10] Varian, H. (2005). *Bootstrap tutorial*. Mathematica Journal, **9**, 768-775

}

APPENDIX

A.1. R Code for Simplicial Intercept Depth Method

```
#distance btw 2 points#

distance <- function(a, b){

  sqrt((a[1]-b[1])^2+(a[2]-b[2])^2)

}

#point of intersection of 2 lines#

poi <- function(x,y){                          #x,y indicate (intercept,slope) of each line#

  e <- (x[2]-y[2]);

if(e==0) {    no.inter<-c(0,0);       #parallel or overlap#

              no.inter

      }else

{yy <- -1*(y[2]*x[1]-x[2]*y[1])/e;

if(y[2]==0){

    xx <- (yy-x[1])/x[2];

}

else{

    xx <- (yy-y[1])/y[2];

}

p<-c(xx,yy);

p}

}
```

```
#read the data#

Group1 <-t(read.table(file.choose(),sep=" ")); #1012 group1 barly with 3 reps#

Group2 <-t(read.table(file.choose(),sep=" ")) ; #1050 group2 barly with 4 reps#

## make length of each group as 5% ##

nF <- 50    ;                              #NF=1012#

nG <- 51    ;                                #NG=1050#

F <- rep(0,nF)

G <- rep(0,nG)

xF <- 0;

xG <- 1;

##use Line(u) to get the (intercept, slope) of the no.u line##

##u from 1 to nG*nF      ##

k.line<-cbind(c(rep(c(1:nF),each=nG)),c(rep(c(1:nG),nF)));

Line <- function(u){

   intc <- F[c(rep(c(1:nF),each=nG))];

   slp <- rep(0,nF*nG)

   for(i in 1:(nF*nG)){

        slp[i] <- G[k.line[i,2]]-F[k.line[i,1]]

      }

   L<-cbind(intc, slp);

   L[u,]

}
```

```
nL <- nG*nF                    #number of lines#

#get the set of all triangles#

   #triangles with 2 vertices   form F#

nt2F <- nG*choose(nF,2);

kt2F <- cbind(matrix(rep(combn(c(1:nF),2),nG),ncol=2,byrow=TRUE),rep(c(1:nG),each=choose(nF,2)));

triangle.2F1G <- function(k){          #length of k is 3#

   line1<- c(F[k[1]],G[k[3]]-F[k[1]])

   line2<- c(F[k[2]],G[k[3]]-F[k[2]])

   edgecoef <-rbind(line1,line2)

   l1<-distance(c(0,F[k[1]]),c(0,F[k[2]]))

   l2<-distance(c(0,F[k[1]]),c(1,G[k[3]]))

   l3<-distance(c(0,F[k[2]]),c(1,G[k[3]]))

   loe<-c(l1,l2,l3)

   list("edgecoef"=edgecoef, "max.e"=max(loe))       #get the coef of 2 edges and the longest length of all 3 lines#

}

   #triangles with 2 vertices   form G#

nt2G <- nF*choose(nG,2);

kt2G <- cbind(rep(c(1:nF),each=choose(nG,2)),matrix(rep(combn(c(1:nG),2),nF),ncol=2,byrow=TRU
E));

triangle.1F2G <- function(k){

   line1<- c(F[k[1]],G[k[2]]-F[k[1]])

   line2<- c(F[k[1]],G[k[3]]-F[k[1]])

   ec<-rbind(line1,line2)

   l1<-distance(c(0,F[k[1]]),c(1,G[k[2]]))

   l2<-distance(c(0,F[k[1]]),c(1,G[k[3]]))

   l3<-distance(c(0,G[k[2]]),c(1,G[k[3]]))

   loe<-c(l1,l2,l3)

   list("edgecoef"=ec, "max.e"=max(loe))

}

#identify each triangle#

tr <- function(q){

   if(q<=nt2F){

      kt<-kt2F

      t<-triangle.2F1G(kt[q,])

   }else

   {

      kt<-kt2G

      t<-triangle.1F2G(kt[q-nt2F,])

   }

   t
```

```
}

nt <- nt2F+nt2G;        #no. of all triangles#

#SID function#

SID <- function(l){

    d<-0;s<-0;m<-1;pp1<-c(0,0);pp2<-pp1

    i<-0

    while(i<nt){

        i<- i+1

        m<- tr(i)$max

        if(sum(Line(l)==tr(i)$e[1,])==2 || sum(Line(l)==tr(i)$e[2,])==2){

               d<- distance(c(0,Line(1)[1]),c(1,sum(Line(l))))

               }

        else{

             if(i<=nt2F){xxx<-0;yyy<-Line(l)[1]}

             else{xxx<-1;yyy<-sum(Line(l))};

             pp1<- poi(Line(l),tr(i)$e[1,]);

             pp2<- poi(Line(l),tr(i)$e[2,]);

             if(pp1[1]>0 & pp1[1]<1 & pp2[1]>0 & pp2[1]<1){d<- distance(pp1,pp2)};

             if(!(pp1[1]>0 & pp1[1]<1) & !(pp2[1]>0 & pp2[1]<1)){d<- 0};

             if(pp1[1]>0 & pp1[1]<1 & !(pp2[1]>0 & pp2[1]<1)){d<-
distance(pp1,c(xxx,yyy))};

             if(!(pp1[1]>0 & pp1[1]<1) & pp2[1]>0 & pp2[1]<1){d<-
distance(c(xxx,yyy),pp2)};
                      }
              s<- (d/m) + s
        }
        s<- s/nt;
        s
}
####Bootstrapping with 1000 times####
beta0 <- rep(0,1000)
beta1 <- rep(0,1000)
b<-0
while(b < 1000){
    b <- b + 1
    F <- sample(Group1, size=nF, replace=TRUE)
    G <- sample(Group2, size=nG, replace= TRUE)
    #find the line with max SID value#
    SID.v <- c(rep(0,nL))
    for(z in 1:nL){
        SID.v[z] <- SID(z)
    }
    location<- (1:nL)[SID.v==max(SID.v)]
    beta <- Line(location[1])
```

```
   beta0[b] <- beta[1]

   beta1[b] <- beta[2]

}
```

A.2. R Code for the Theil-type Estimate.

```
options(expressions=100000)

library(gregmisc)

n = 50

ss = n*(n-1)/2

ind = combinations(n, 2, repeats=FALSE)

c1 = rep(ind[,1], each = ss, times =1)

c3 = rep(ind[,2], each = ss, times =1)

c2 = rep(ind[,1], each = 1, times=ss) + n

c4 = rep(ind[,2], each = 1, times=ss) + n

sim = 1000

betan.oja = matrix(NA, 2, sim)

b0=1

b00=3

for (i in 1:sim) {

en = rnorm(100,0,1)

for (w in 1:50){
  yn1[w] = b0+en[w]

  yn2[w] = b0+b00+en[w+50]

}

yn1[c(1:3)] = -100                    ## creat outliers##

yn = c(yn1,yn2)

y = yn

d1arg1 = y[c1]*y[c4]-y[c2]*y[c3]

d1arg2 = y[c4]-y[c2]

d1arg3 = y[c3]-y[c1]

d2arg1 = y[c1]*y[c2]-y[c3]*y[c4]

d2arg2 = y[c2]-y[c4]

d2arg3 = y[c3]-y[c1]

d1 = cbind(d1arg1, -d1arg2, d1arg3)

d2 = cbind(d2arg1, -d2arg2, d2arg3)

f<-function(b){

b1<- b[1]

b2<- b[2]

vb<- c(1, b1,b1+b2)

colSums(abs(d1%*%vb)+abs(d2%*%vb))/(ss^2)

}

est<-nlminb(c(0,0),f)

betan.oja[,i] = est$par
```

```
crit<-est$convergence

                        }
```