Cluster Analysis for Symbolic Interval Data Using Linear Regression Method

by

Fei Liu

(Under the Direction of Professor Lynne Billard)

Abstract

Symbolic data records are becoming a more powerful instrument to deal with large size data sets. Interval-valued data are a special type of symbolic data, for which each observation is a vector of intervals. The typical K-means methods for interval-valued data suppose the data separate to spherical clusters. It usually cannot converge to the correct clusters if the data are not clustering spherically. We propose a K-regressions based clustering method for interval-valued data to recover a more complicated data structure. Assuming the response and predictor variables follow K different linear relationships, the data are initially split into K groups randomly. Then, we apply the new developed "symbolic variation" least squares to estimate the parameters of the K symbolic regressions. A data point is then relocated to its closest group in terms of its symbolic distance to the regression lines. This two-step dynamic clustering algorithm continues until the clusters are stable. Further, we introduce an orthogonal regression clustering algorithm (ORCA) for interval-value data to avoid specifying a response variable. Two orthogonal regression methods are proposed: the simple orthogonal regression method and the general orthogonal regression method. We utilize four different methods to determine the optimal number of clusters. Simulation study

is conducted to investigate the performance of the ORCA algorithm. We use the Iris data (Fisher, 1936) to test the effectiveness of the ORCA algorithm.

INDEX WORDS: Symbolic data analysis, Cluster analysis, Interval-valued data, Linear regression, Orthogonal regression, Measurement error model

Cluster Analysis for Symbolic Interval Data Using Linear Regression Method

by

Fei Liu

B.S., Huazhong University of Science and Technology, 2003

M.S., Georgia State University, 2010

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2016

Cluster Analysis for Symbolic Interval Data Using Linear Regression Method

by

Fei Liu

Approved:

Major Professors: Lynne Billard

Committee: Pengsheng Ji
William McCormick
Jaxk Reeves
Paul Schliekelman

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2016

Cluster Analysis for Symbolic Interval Data Using Linear Regression Method

Fei Liu

March 24, 2016

Dedication

To my dear parents, Meilian Liu and Zhicheng Liu;

my beloved wife, Ting Zhang;

and my soon to be born son, Albert C. Liu.

Acknowledgments

I would like to express my deepest appreciation and gratitude to Professor Lynne Billard, my major advisor, for her patient guidance, encouragement, and tremendous help through my research. Her enthusiasm, confidence, inspiration, and optimism have set a great example to be a successful scientific researcher. Without her persistent guidance and help this dissertation would not have been possible.

I would also like to extend my sincere thanks to Dr. Pengsheng Ji, Dr. William McCormich, Dr. Jaxk Reeves, and Dr. Paul Schliekelman to serve as my advisory committee. I appreciate their invaluable aids, thoughtful comments, and precious time they have spent to review my dissertation. I would also like to thank Dr. T.N. Sriram for his help and guidance during the SAS shootout. I appreciate all the faculty, staff, and my friends in the department for their tremendous instruction, help, and guidance throughout my years study in the program.

Lastly, special thanks to my wife, Ting Zhang, who has provided practical and emotional support and encouragement throughout my life and study for years. Her love and sacrifices are always the greatest motivation for me to complete this study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traditionally, statistical analysis deals with classical data where the values of a random variables are numbers or multiple levels. The values of symbolic data, in contrast, can be a list of numbers, a combination of numbers and factors, an interval, a histogram, or a distribution. Symbolic data usually come from two circumstances: the data are collected by a symbolic format, e.g., daily temperature for a particular city is [low temperature, high temperature]; or a large sized classical data set is aggregated into a symbolic format, e.g., the credit card transactions in a month for the accounts within a certain range of credit score can be aggregated as a histogram. More details about different types of symbolic data and their definitions can be found in Chapter 2. Billard and Diday (2006a) has given numerous examples about symbolic data and its applications. Nowadays, with the exponential growth of data size in all different areas, approaches that can extract information from large sized data sets are becoming more and more important. Since large sized classical data sets can be aggregated into a workable size of symbolic data, symbolic data become a promising method to deal with large sized data. Furthermore, the classical way of dealing with symbolic data, e.g., using only the interval center points for linear regression of interval-valued data, is not appropriate and misleading. Using symbolic methods to handle symbolic data is necessary.

This dissertation mainly focus on proposing approaches to cluster symbolic data, specifically for interval-valued data, by linear regression models. The clustering methodologies for spherical data structure with interval-value data have been well developed by Chavent and Lechevallier (2002), de Souza and de Carvalho (2004), de Souza et al. (2004), de Carvalho et al. (2006a,b), de Carvalho and Lechevallier (2009), where the algorithm proposed all adapt the K-means clustering algorithm for classical data. When observations in each cluster of a data set are clustering around a linear regression line, the clustering algorithm for spherical data structure can fail. In other words, when two or more variables in each cluster of a data set are highly correlated so that they follow a linear regression model, the clustering algorithm such as K-means algorithm does not work well. To recover a linear regression line based clusters for classical data is called the cluster-wise regression, a method that is well developed in multiple articles such as Späth (1979, 1981, 1982), DeSarbo and Cron (1988), Wedel and Kistemaker (1989), Zhang (2003), Van Aelst et al. (2006), García-Escudero et al. (2009), etc. However, the methodology of cluster-wise regression for symbolic data has not been studied. It is necessary to develop algorithms that can recover the clusters that are clustering around linear regression lines for interval-valued data.

The rest of this dissertation is organized as follows: Chapter 2 reviews the concept of symbolic data, the fundamental statistical definitions of interval-valued data, cluster-wise linear regression for classical data, previous studies about cluster analysis for symbolic data, the likelihood function and maximum likelihood estimation for symbolic data. In Chapter 3, we apply the symbolic variance method (Xu, 2010) and propose a K-regression algorithm to implement a cluster-wise regression for interval-valued data. We conduct simulation studies to compare the clustering results between the K-means based algorithm and the K-regressions algorithm we proposed. The performance of the K-regressions algorithm is also investigated through several simulated data sets. We propose two orthogonal regression methods for interval-valued data in Chapter 4: one applies the principal component analysis methodol-

ogy and the other adapts the measurement error model. The orthogonal regression methods are applied in a proposed algorithm, the orthogonal regression clustering algorithm, that is used to recover the clusters that are clustering around linear regressions lines. We apply four methods to determine the optimal number of clusters. Six different interval-valued data sets are simulated to study the performance of the orthogonal regression clustering algorithm and the performance of different approaches that determine the optimal number of clusters. The performance of the algorithm is also examined by a real data, the Iris data (Fisher, 1936). Finally, Chapter 5 discusses the possible future research based on the measurement error model for interval-valued data.

Chapter 2

Literature Review

This chapter reviews the concepts, methods, and theories that are relevant to the clustering of symbolic data. We review the concept of symbolic data in section 2.1. In section 2.2, we review methods developed to implement linear regression for interval-valued data. Section 2.3 summarizes the literature that considers cluster-wise linear regression for classical data. Cluster analysis of symbolic data using different methodologies and different dissimilarity measurements are reviewed in section 2.4. Section 2.5 reviews likelihood functions and some maximum likelihood estimators for symbolic data.

2.1 Symbolic Data

In statistical analyses, we typically deal with classical data for which the values of a random variable are numbers or multiple categories. In contrast, for symbolic data, the value or the realization for a random variable could be a list of numbers, a combination of numbers and factors, an interval, a histogram, or a distribution. The nature of symbolic data could depend on how the data are collected. For example, the blood pressure for a person is naturally an interval value, [low pressure, high pressure]. It could also originate from aggregating a large sized classical data set into manageable pieces so that the data set becomes one of workable

size. The complicated structure of symbolic data brings great challenges for statistical analyses. The basic concepts and descriptions for classical data would apply to symbolic data but their precise formulas do not apply directly. Billard and Diday systematically define different types of symbolic data and the descriptive statistics of these data (Billard and Diday, 2004, 2006b,a, Billard, 2007). In the remainder of this section, we give a brief introduction of some types of symbolic data.

Using the examples in Billard and Diday (2006a), suppose we have a data set that records the manufacturers and models of cars within households; for each household, the manufacturers and models of its cars are a list. For instance, one household may have

$$x_1 = \{\text{Honda Accord, Thunderbird}\};$$

another may have

$$x_2 = \{\text{Toyota Camry, Volvo, Renault}\}.$$

Denote X as the records of the manufacturers and models of cars within households; then, X is a random variable. The above x_1 and x_2 are realizations of X. Obviously, x_1, x_2 are not single values; instead, they are a list of categorical values.

The list of values could be numbers as well. For example, the records of weights in elementary and middle schools of Athens GA for certain age×gender combinations are a list of numbers. The realization of such a random variable might be like,

$$x_1 = x_1(\text{male at 14})=\{120, 95, 80, 90, 93, 102, 88, 113, 102\},$$
$$x_2 = x_2(\text{male at 15})=\{124, 115, 120, 98, 96, 92, 138, 111, 106\}.$$

The list could be a combination of numbers and categorical values. For example, the records of demographic information, city, age, and gender, in a clinic could be {Atlanta, 45, male}. All these random variables are **multi-valued** symbolic random variables.

Definition 2.1.1. *A random variable is called a* ***multi-valued*** *symbolic random variable if its possible values take one or more values from a list of values. The values in the list could be well-defined categorical or quantitative values.* □

For a multi-valued symbolic random variable, if each value of a particular realization is associated with a non-negative measurement, then, it is a **modal multi-valued** symbolic random variable. The measurements are typically weights, probabilities, relative frequencies, etc. The detailed definitions could be found in Billard and Diday (2006a).

We would like to introduce the next type of symbolic data by a example from Billard and Diday (2006a). Consider credit card expenses for a group of persons in a relatively long period, the expenses for a particular person in a certain month is an interval value. For example, Jon's expenses (in dollars) in January and February are

$$x_1 = x_1(\{\text{Jon, Jan}\})=[320.81,\ 538.29],$$
$$x_2 = x_2(\{\text{Jon, Feb}\})=[434.54,\ 598.12].$$

Another example of interval-valued data is the records of blood pressure. The blood pressure for a particular person is always an interval like x_1=[low pressure, high pressure]=[75, 120].

Definition 2.1.2. *A symbolic random variable is* ***interval-valued*** *if it takes values in an interval, i.e.,* $X = [a, b] \subset \mathfrak{R}^1$*, with* $a \leq b$*, and* $a, b \in \mathfrak{R}^1$*. The interval can be closed or open at either end.* □

A more complicated type of symbolic data is the **histogram interval-valued** symbolic random variable where the realization of the random variable is a histogram. Following the example from Billard and Diday (2006a), suppose we have the records of arrival delay (X_1), departure delay (X_2), and weather delay (X_3) for each airline carrier flying into New York's JFK Airport. The records for "Airline 1" may be,

$$x_{11} = x_{11}(\text{Airline 1})=\{(\leq 0], .42; (0, 60], .46; [> 60), .12\},$$

$$x_{12} = x_{12}(\text{Airline 1})=\{(\leq 0], .44; (0, 60], .47; [> 60), .09\},$$
$$x_{13} = x_{13}(\text{Airline 1})=\{(\leq 0], .92; (> 0), .08\},$$

where $x_{.j}$ are the realizations of the three histogram-valued random variables, $X_j,\ j = 1, 2, 3$, for "Airline 1". The number following each sub-interval for an particular realization is a percentage for that sub-interval. Generally, the number could be a weight, or relative frequency, or probability for that particular interval. Compared with interval-valued variables, usually histogram random variables provide more information by giving weights for each sub-interval within a particular realization. That can possibly lead to better statistical inference.

Definition 2.1.3. *Suppose X is a quantitative random variable taking values on a finite number of non-overlapping intervals, $\{[a_k, b_k], k = 1, ..., n\}$ with $a_k \leq b_k$, and $n < \infty$. A realization of X takes the form*

$$x_i = \{[a_{ik}, b_{ik}], p_{ik}; k = 1, ..., s_i\},$$

where $s_i < \infty$ is the number of intervals for the i^{th} realization of X, and p_{ik} is the weight of the k^{th} interval for x_i with $\sum_{k=1}^{s_i} p_{ik} = 1$. Either end of the interval for x_i could be open or closed. Then, X is a ***histogram interval-valued*** *symbolic random variable.* □

Since we will focus on interval-valued data in this dissertation, the formal definition of distribution and descriptive statistics of interval-valued data will be discussed in the remainder of this section. The descriptive statistics for interval-valued data follow the approach adopted by Bertrand and Goupil (2000). More details of the topic could be found in Billard and Diday (2006b,a), and Billard (2007).

Unlike classical data, there is no simple way to use one distribution to describe interval-valued data. Nevertheless, the concepts of descriptive statistics for interval-valued data are the same as for classical data, which include histogram, sample mean, sample variance and

covariance, etc. Before touching on these quantities, it is necessary to introduce a common notation of symbolic data and the concept of virtual extensions.

Let the random variables $X_j, j = 1, ..., p$, have domain $\mathcal{X} = \times_{j=1}^{p} \mathcal{X}_j$. Then, each point $\boldsymbol{x} = (x_1, ..., x_p)$ in $\mathcal{X}$ is called a **description vector**. Furthermore, let D_j be a subset of $\mathcal{X}_j$, or $D_j \subseteq \mathcal{X}_j$. Then, the p-dimensional subspace $D = (D_1, ..., D_p) \subseteq \mathcal{X}$ is called a **description set**. The symbolic description of an observation $i \in \Omega = \{1, ..., n\}$ for random variables $X_j, j = 1, ..., p$, is given by $\boldsymbol{d_i} = (x_{i1}, ..., x_{ip}), i = 1, ..., n$. The set of all possible descriptions is called the description space $\mathcal{D}$. In any particular case, x_{ij}, the i^{th} realization of x_j, could be classical data or symbolic data. When each $D_j, j = 1, ..., p$, is a set of one value only, then, the description vector d is defined as an **individual description**, i.e., $\boldsymbol{x} = (x_1, ...x_p) \equiv \boldsymbol{d} = (\{x_1\}, ..., \{x_p\})$, where $\boldsymbol{x} \in \mathcal{X} = \times_{j=1}^{p} \mathcal{X}_j$.

For symbolic data, there usually exist certain implicit logical dependencies between individual descriptions. A logical dependency can be represented by a rule v,

$$v : [x \in A] \Rightarrow [x \in B]$$

for $A \subseteq D, B \subseteq D$ and $x \in \mathcal{X}$, where v is a mapping of $\mathcal{X}$ onto $\{0, 1\}$ with $v(x) = 1$ if the rule is satisfied, and 0 if not.

Definition 2.1.4. *Let $\boldsymbol{x} \in D \subseteq \mathcal{X} = \times_{j=1}^{p} \mathcal{X}_j$ be the individual description vector, and let $V_{\mathcal{X}}$ be the set of all rules v operating on $\mathcal{X}$. Then, the **virtual description**, $vir(d)$, of the description vector d is the set of all individual description vectors $\boldsymbol{x} \in D$ such that $v(\boldsymbol{x}) = 1$ for $v \in V_{\mathcal{X}}$, denoted as*

$$vir(d) = \{\boldsymbol{x} : \boldsymbol{x} \in D, v(\boldsymbol{x}) = 1, \forall v \in V_{\mathcal{X}}\}. \tag{2.1}$$

□

The sample mean and variance for interval-valued data were given by Bertrand and Goupil (2000). Let X be an interval-valued random variable. The n observations of X are $x_1, ..., x_n$, with $x_i = [x_{ia}, x_{ib}], i = 1, ..., n$. The individual description vectors $x \in vir(d_i)$ are assumed to be uniformly distributed over the interval $[x_{ia}, x_{ib}]$. Therefore, for $\xi \in [x_{ia}, x_{ib}]$, the empirical density function is $f(\xi) = 1/(x_{ib} - x_{ia}), \forall \xi \in [x_{ia}, x_{ib}]$. Moreover, it is assumed that each object, x_i, $i = 1, ..., n$, is equally likely to be observed with probability $1/n$. Then, we have the following definition.

Definition 2.1.5. *The **empirical density function** for an interval-valued random variable X is defined as*

$$f(\xi) = \frac{1}{n} \sum_{i:\xi \in [x_{ia}, x_{ib}]} (\frac{1}{x_{ib} - x_{ia}}). \tag{2.2}$$

□

Let $I = [\min_{i \in \Omega} x_{ia}, \max_{i \in \Omega} x_{ib}] = [I_{min}, I_{max}]$, with $\Omega = \{1, ..., n\}$. Then, I is the interval that covers all the observed intervals for the random variable X. We partition the interval I into r bins $I_g = [\zeta_{g-1}, \zeta_g), g = 1, ..., r$, where $I_0 = I_{min}$ and $I_r = I_{max}$, and the r bins are usually equal in length. To construct the histogram, we define the **observed frequency** and **relative frequency** of the interval-valued variable as follows.

Definition 2.1.6. *The **observed frequency** of an interval-valued variable X, given the bins of the histogram I_g, is, for $g = 1, ...r$,*

$$f_g = \sum_{i \in \Omega} \frac{\|x_i \cap I_g\|}{\|x_i\|} \tag{2.3}$$

*and the **relative frequency** is*

$$p_g = f_g / n, \tag{2.4}$$

where x_i is the i^{th} observation of X, and $\|\cdot\|$ is the length of an interval. □

From the empirical density of an interval-valued random variable, we can obtain the **symbolic sample mean** and **symbolic sample variance** of an interval-valued variable.

Definition 2.1.7. *Let X be an interval-valued random variable. Suppose $x_1, ..., x_n$ is a random sample of X with $x_i = [x_{ia}, x_{ib}], i = 1, ..., n$. The* ***symbolic sample mean*** *of X is given by*

$$\bar{X} = \frac{1}{2n} \sum_{i=1}^{n} (x_{ia} + x_{ib}). \tag{2.5}$$

The ***symbolic sample variance*** *is given by*

$$S^2 = \frac{1}{3n} \sum_{i=1}^{n} (x_{ib}^2 + x_{ib}x_{ia} + x_{ia}^2) - \frac{1}{4n^2} \left[\sum_{i=1}^{n} (x_{ib} + x_{ia}) \right]^2. \tag{2.6}$$

□

The sample mean and sample variance in equations (2.5) and (2.6) were first derived in Bertrand and Goupil (2000). Billard (2008) showed that the sample variance could be decomposed as follows. By some reorganization, *TotalSS*$\equiv nS^2$ (of equation (2.6)) can be written as the sum of two parts, *TotalSS = Within Sum Squares(WithinSS) + Between Sum Squares(BetweenSS)*, where

$$WithinSS = \frac{1}{3} \sum_{i=1}^{n} \left[(x_{ia} - \bar{x}_i)^2 + (x_{ia} - \bar{x}_i)(x_{ib} - \bar{x}_i) + (x_{ib} - \bar{x}_i)^2 \right], \tag{2.7a}$$

$$BetweenSS = \sum_{i=1}^{n} \left[\frac{x_{ia} + x_{ib}}{2} - \bar{X} \right]^2 \tag{2.7b}$$

with $\bar{x}_i = (x_{ia} + x_{ib})/2$, $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_i$. We can show that

$$WithinSS = \sum_{i=1}^{n} \frac{(x_{ib} - x_{ia})^2}{12}, \tag{2.8}$$

which is the same as that of the uniform distribution variance for each individual interval

x_i. The *WithinSS* depends on the assumption of the distribution within the intervals of X. We have the *WithinSS* as in equation (2.8) since the individual realization, x_i, is assumed to be a uniform distribution within each interval. When the internal distribution of X is different, the *WithinSS* of X is different. Now, it is clear that the symbolic sample variance for an interval-valued random variable is the sum of two pieces, the total variance within each interval and the variance between the center points of each interval.

Definition 2.1.8. *Let X_1, X_2 be two interval-valued random variables, and let both have n observations with $x_{i1} = [x_{i1a}, x_{i1b}]$ and $x_{i2} = [x_{i2b}, x_{i2b}], i = 1, ..., n$, as the i^{th} observations of X_1 and X_2. The **symbolic sample covariance** between the two variables is given by*

$$\begin{aligned} Cov(X_1, X_2) =& \frac{1}{6n} \sum_{i=1}^{n} [2(x_{i1a} - \bar{X}_1)(x_{i2a} - \bar{X}_2) + (x_{i1a} - \bar{X}_1)(x_{i2b} - \bar{X}_2) \\ & + (x_{i1b} - \bar{X}_1)(x_{i2a} - \bar{X}_2) + 2(x_{i1b} - \bar{X}_1)(x_{i2b} - \bar{X}_2)], \end{aligned} \tag{2.9}$$

where $\bar{X}_1 = \frac{1}{n}\sum_{i=1}^{n}(x_{i1a} + x_{i1b})/2, \quad \bar{X}_2 = \frac{1}{n}\sum_{i=1}^{n}(x_{i2a} + x_{i2b})/2.$ □

Similarly as for the sample variance, we can denote the sample covariance as *(Total Sum Products)/6n*, or *TotalSP/6n*, and so the sample covariance could be decomposed as *TotalSP=WithinSP + BetweenSP* with

$$WithinSP = \sum_{i=1}^{n} (x_{i1b} - x_{i1a})(x_{i2b} - x_{i2a})/12, \tag{2.10a}$$

$$BetweenSP = \sum_{i=1}^{n} \left[(x_{i1a} + x_{i1b})/2 - \bar{X}_1\right] \left[(x_{i2a} + x_{i2b})/2 - \bar{X}_2\right]. \tag{2.10b}$$

The sample covariance is also composed into two parts. The *WithinSP* is the total within interval association between the two random variables, and the *BetweenSP* is the covariance between the interval center points of the two random variables.

Note. From the above equation (2.10), the *WithinSP* is always a positive addition on the

BetweenSP, even if X_1 and X_2 are actually negatively correlated or not correlated. The *WithinSP* is supposed to measure the association within the rectangles, but we do not have that information in fact. If the assumption is that the distributions within each rectangle are two independent univariate uniform distributions for X_1 and X_2, then, the association between the two random variables within rectangles should be 0. If the assumption is that the distribution within the rectangular is a bivariate uniform distribution with nonzero correlation, then, this nonzero correlation affects the within interval variation between the two random variables. For either scenario, further development is needed to measure the within interval variation.

2.2 Linear Regression for Interval-Valued Data

Linear regression is a common method used for classical data analysis, and so will also be an important method for symbolic data analysis. Since we are going to apply linear regression to interval-valued data clustering later, we review the development of linear regression for interval-valued data briefly in this section.

The first approach of fitting linear regression to interval-valued data was introduced by Billard and Diday (2000). They fitted the linear regression using the center point of the symbolic intervals, called the center method (CR). Later, Neto et al. (2005a,b), and Neto and de Carvalho (2008) transformed the intervals to center points and ranges, and fitted two separate linear regressions which is called the Center and Range method (CRM). The CRM method faces a problem that the lower bound of the predicted interval of response could be larger than the upper bound. To address the problem, Neto and de Carvalho (2010) added a constraint to the CRM, called CCRM, such that all the coefficients of the linear regression for the range points must be non-negative. In this way, the predicted lower bound will always be equal or smaller than the upper bound. There are some methods that built upon

the CRM, which are more robust with outliers. Domingues et al. (2010) used the CRM but adapted a Student-t distribution to the error term of the center points regressions but still used the normal error for the range regression component. Fagundes et al. (2013) applied a weighted regression to the CRM in order that the model is more resistant to extreme values.

The most recent development of linear regression for symbolic intervals was by Xu (2010) who took analogous multiple regression methods ideas for classical data, and then used the symbolic variance and covariance results for intervals to obtain the regression coefficient estimates. This is called the symbolic variance method (SVM). This is the first method that not only considers the information of the lower and upper bounds of the intervals, but also considers the variations within the intervals.

Some authors explored the method of using interval arithmetic for interval linear regression (Blanco-Fernández et al., 2011, 2013). Since there are problems with the use of interval arithmetic methods, we are not going to cover these ideas in this dissertation. Recently, Sun and Li (2014) introduced an approach which is similar to the CRM but which forces the coefficients of the range regression line to be the absolute value of the corresponding coefficients for the center point regression except for the intercept. They did not constrain the coefficients of the range regression line to be positive; instead, when the predicted range is negative, they shrunk it to be zero. We briefly review some of these models in the remainder of this section.

Suppose we have n observations in a data set with response variable Y and p predictor variables $X_1, ..., X_p$. For each observation, each variable is an interval-valued random variable. Let $x_{ij}, i = 1, ..., n, j = 1, ..., p$, be the i^{th} observation for the j^{th} predictor variable, denoted by $x_{ij} = [x_{ija}, x_{ijb}]$, with $x_{ija}, x_{ijb} \in \mathbb{R}$ and $x_{ija} \leq x_{ijb}$. Similarly, let y_i be the i^{th} observation for the response variable Y, denoted by $y_i = [y_{ia}, y_{ib}]$, with $y_{ia}, y_{ib} \in \mathbb{R}$ and $y_{ia} \leq y_{ib}$. Assume the response variable Y has a linear relationship with the predictors $\boldsymbol{X} = (X_1, ..., X_p)$.

Billard and Diday (2000) took an analogue of standard classical theory to obtain the regression coefficient estimates for the linear regression of interval-valued data. They defined the empirical joint density function for two interval-valued random variable X_1 and X_2 by

$$f(\xi_1, \xi_2) = \frac{1}{n} \sum_{i=1}^{n} \frac{I_i(\xi_1, \xi_2)}{||z_i||}, \tag{2.11}$$

where $z_i = x_{i1} \times x_{i2} = ([x_{i1a}, x_{i1b}], [x_{i2a}, x_{i2b}])$ is a rectangle on $X_1 \times X_2$, $||z_i||$ is the area of the rectangle, and $I_i(\xi_1, \xi_2)$ indicates whether (ξ_1, ξ_2) is in the rectangle z_i. Then, Billard and Diday (2000) derived the sample covariance between X_1 and X_2 as

$$\text{Cov}(X_1, X_2) = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_{i1} \bar{x}_{i2} - \frac{1}{n^2} \sum_{i=1}^{n} \bar{x}_{i1} \sum_{i=1}^{n} \bar{x}_{i2} \tag{2.12}$$

where $\bar{x}_{ij} = (x_{ija} + x_{ijb})/2$, $j = 1, 2$, is the center point of the interval $[x_{ija}, x_{ijb}]$. The sample variance of an interval-valued variable X is defined as

$$S^2 = \frac{1}{n} \sum_{i=1} \bar{x_i}^2 - \frac{1}{n^2} \left(\sum_{i=1}^{n} \bar{x}_i \right)^2 \tag{2.13}$$

with $\bar{x_i} = (x_{ia} + x_{ib})/2$, the center point of interval $[x_{ia}, x_{ib}]$. The empirical sample variance in equation (2.13) and sample covariance in equation (2.12) for the interval-valued data are the classical variance and covariance of the interval center points. The linear regression coefficient estimates could then be derived by the standard theory for classical data. For $p = 1$, the estimates are

$$\begin{aligned} \hat{\boldsymbol{\beta}}_1 =& \frac{\text{Cov}(Y, X)}{S_X^2}, \\ \hat{\boldsymbol{\beta}}_0 =& \bar{Y} - \hat{\boldsymbol{\beta}}_1 \bar{X}. \end{aligned} \tag{2.14}$$

From (2.14), the prediction for the CM method, given a new observation $x = [x_a, x_b]$, is

$\hat{y} = [(\boldsymbol{x}_a)^T\hat{\boldsymbol{\beta}}, (\boldsymbol{x}_b)^T\hat{\boldsymbol{\beta}}]$.

Note. Note that the definitions of sample variance and sample covariance in equations (2.12), (2.13) are the earlier versions of the definitions (2.1.7), (2.1.8). We will use the definitions (2.1.7), (2.1.8) in our research since they are more appropriate.

Later, Neto et al. (2005a,b) introduced an approach called the center and range method (CRM). Denote the center point of a interval $x = [x_a, x_b]$ with $x_a, x_b \in \Re$ and $x_a \leq x_b$, as $x^c = (x_a + x_b)/2$, the range of the interval as $x^r = x_b - x_a$, and the radius (half of the range) of the interval as $x^\delta = r/2$. Neto et al. (2005a) transformed the intervals of Y and $\boldsymbol{X}$ to their interval center points and interval ranges, Y^c, Y^r, $\boldsymbol{X}^c$, and $\boldsymbol{X}^r$. Then, they fitted separate linear regressions to the center points and ranges between Y and $\boldsymbol{X}$, respectively. The model is formulated as follows:

$$\begin{aligned} Y^c &= \boldsymbol{X}^c\boldsymbol{\beta}^c + \boldsymbol{\epsilon}^c, \\ Y^r &= \boldsymbol{X}^r\boldsymbol{\beta}^r + \boldsymbol{\epsilon}^r, \end{aligned} \tag{2.15}$$

where $\boldsymbol{X}^c = (X_1^c, \ldots, X_p^c)$, $X^r = (X_1^r, \ldots, X_p^r)$ are the interval center points and interval ranges for variables $X_1, \ldots, X_p$ with $X_j^c = (x_{1j}^c, \ldots, x_{nj}^c)^T$ and $X_j^r = (x_{ij}^r, \ldots, x_{nj}^r)^T$, $j = 1, \ldots, p$, being the interval center points and interval range for variable X_j; $\boldsymbol{\beta}^c = (\beta_1^c, \ldots, \beta_p^c)^T$ and $\boldsymbol{\epsilon}^c = (\epsilon_1^c, \ldots, \epsilon_n^c)$ are the regression coefficients and error terms for the interval center point regression model, while $\boldsymbol{\beta}^r = (\beta_1^r, \ldots, \beta_p^r)^T$ and $\boldsymbol{\epsilon}^r = (\epsilon_1^r, \ldots, \epsilon_n^r)$ are the regression coefficients and error terms for the interval range regression model.

To estimate the coefficients $\boldsymbol{\beta}^c$, $\boldsymbol{\beta}^r$, Neto et al. (2005a) minimized the following function

$$S = \sum_{i=1}^{n}((\epsilon_i^c)^2 + (\epsilon_i^r)^2),$$

which is equivalent to minimizing the two parts, $\sum_{i=1}^n(\epsilon_i^c)^2$ and $\sum_{i=1}^n(\epsilon_i^r)^2$, separately. The regression coefficient estimates, $\hat{\boldsymbol{\beta}}^c, \hat{\boldsymbol{\beta}}^r$, of the two regressions in equation (2.15) can be

obtained by least squares estimation. The predicted interval, given an observation of X, is

$$\hat{y}_a = \hat{y}^c - \hat{y}^r, \quad \hat{y}_b = \hat{y}^c + \hat{y}^r, \tag{2.16}$$

where $\hat{y}^c = (\boldsymbol{x}^c)^T \hat{\boldsymbol{\beta}}^c$, and $\hat{y}^r = (\boldsymbol{x}^r)^T \hat{\boldsymbol{\beta}}^r$.

A problem for the CRM methods is that it cannot guarantee that the lower bound is always smaller than the upper bound for the predicted intervals. To address the problem, Neto and de Carvalho (2010) added constraints to the CRM models. The constraints force the coefficients of the range regression to be always positive so that the predicted range will be always positive. Specifically, the model is

$$\begin{aligned} &Y^c = \boldsymbol{X}^c \boldsymbol{\beta}^c + \boldsymbol{\epsilon}^c, \\ &Y^r = \boldsymbol{X}^r \boldsymbol{\beta}^r + \boldsymbol{\epsilon}^r, \\ &\text{with constraints } \beta_j^r \geq 0, j = 0, 1, ..., p. \end{aligned} \tag{2.17}$$

The model in equation (2.17) is called the Constrained Center and Range method (CCRM). While we can still apply the least squares estimation method to the linear regression model for the center points, we cannot apply the least squares estimation method directly to the regression for the ranges subject to the constraints. Neto and de Carvalho (2010) used the Lawson and Hanson's algorithm (Lawson and Hanson, 1974) to estimate the $\boldsymbol{\beta}^r$ numerically. Given a new observation of $\boldsymbol{X}$, the prediction for CCRM is the same as in equation (2.16).

The positive constraints on $\boldsymbol{\beta}^r$ could be problematic. It is not necessary that all the ranges of X's have positive correlation with the range of Y. Thus, the CCRM solves one problem, but causes another problem.

Xu (2010) applied symbolic variance and covariance values to obtain the symbolic variation least squares estimators of the regression parameters for an interval data linear regression model. In this way, Xu (2010) utilized all the variation inherent to the data, the within and

between variation. The model is specified as follows:

$$Y = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.18}$$

where Y is the vector of response intervals, $\mathbf{X}$ is the design matrix of predictor variable intervals, $\boldsymbol{\epsilon}$ is the vector of error intervals, $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)$ is the intercept and the coefficients of the p predictor variables. Equation (2.18) can be rewritten as

$$Y - \bar{Y} = (\boldsymbol{X} - \bar{\boldsymbol{X}})\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\bar{Y}$ is the symbolic sample mean of Y, $\bar{\boldsymbol{X}} = (\bar{X}_1, ..., \bar{X}_p)$ with $\bar{X}_j, j = 1, ..., p$, as the symbolic sample mean of X_j. Then, the intercept β_0 in equation (2.18) is given by

$$\beta_0 = \bar{Y} - \beta_1\bar{X}_1 - ... - \beta_p\bar{X}_p.$$

Analogously with the methodology for classical data, the least squares estimators of $\boldsymbol{\beta}_1 \equiv (\beta_1, ..., \beta_p)^T$ are given by

$$\hat{\boldsymbol{\beta}}_1 = ((\boldsymbol{X} - \bar{\boldsymbol{X}})^T(\boldsymbol{X} - \bar{\boldsymbol{X}}))^{-1}(\boldsymbol{X} - \bar{\boldsymbol{X}})^T(Y - \bar{Y}). \tag{2.19}$$

This estimator in equation (2.19) is equivalent to

$$\hat{\boldsymbol{\beta}}_1 = (n \times \mathrm{Cov}(X_{j_1}, X_{j_2}))^{-1}_{p\times p} \times (n \times \mathrm{Cov}(X_j, Y))_{p\times 1}, \tag{2.20}$$

where $n \times \mathrm{Cov}(X_{j_1}, X_{j_2})$ is the $(j_1, j_2)^{th}$ element of the $p \times p$ matrix $(\boldsymbol{X} - \bar{\boldsymbol{X}})^T(\boldsymbol{X} - \bar{\boldsymbol{X}})$, $n \times \mathrm{Cov}(X_j, Y)$ is the j^{th} element of the $p \times 1$ vector $(\mathbf{X} - \bar{\boldsymbol{X}})^T(Y - \bar{Y})$, $j, j_1, j_2 = 1, ..., p$.

Accordingly, the estimator of the intercept β_0 is given by

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}_1 - ... - \hat{\beta}_p \bar{X}_p. \tag{2.21}$$

The estimator in equations (2.20) and (2.21) are the symbolic variation least squares estimators of the regression coefficients $\boldsymbol{\beta}$ in equation (2.18). The predicted interval given an observation of $\boldsymbol{X}$ is

$$\begin{aligned} \hat{y}_a &= \min_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^T \hat{\boldsymbol{\beta}}, \\ \hat{y}_b &= \max_{\mathbf{x} \in \mathbb{X}} \boldsymbol{x}^T \hat{\boldsymbol{\beta}}, \end{aligned} \tag{2.22}$$

where $\mathbb{X} = \{\boldsymbol{x} = (x_j) : x_{ja} \leq x_j \leq x_{jb}, j = 1, ..., p\}$.

Recently, Fagundes et al. (2013) introduced a weighted regression model for interval data. Sun and Li (2014) proposed a δ-distance and a δ-metric to measure the residuals of the model. The δ-metric of an interval variable X is defined as

$$\delta(X) = (X^c)^2 + (X^r)^2, \tag{2.23}$$

and the δ-distance between interval variables X and Y is defined as

$$\delta(X, Y) = \sqrt{(X^c - Y^c)^2 = (X^r - Y^r)^2}, \tag{2.24}$$

where X^c, Y^c are the interval center points of X and Y, X^r, Y^r are the interval ranges of X and Y. They estimated the regression coefficients by minimizing the average metric of the errors $\epsilon_i, i = 1, ..., n$. Both the methods are not essentially different from the CRM; thus, we will not include any more details here.

In summary, the CM, CRM, and CCRM all transfer the interval data regression model to

some forms of classical regression models for a solution. The SVM considers both the within and between interval variation for the estimation of regression coefficients, and it is a real symbolic linear regression method. Furthermore, the SVM provides more precise predictions than do the other four methods, and is verified by simulation results. In this dissertation, we will use the SVM for our research.

2.3 Cluster-wise Linear Regression for Classical Data

Cluster analysis is a common statistical tool that divides the population into different sub-populations such that the subjects within the same sub-population are similar while the subjects from different sub-populations are dissimilar. The fundamental and most well-known clustering method is the K-means clustering (MacQueen, 1967). For a fixed K, the K-means algorithm needs to form initial clusters to start the clustering process, which is called initialization. The initialization could be K seeds or K clusters; a detailed discussion can be found in Anderberg (1973), Cormack (1971). Then, the algorithm partitions the n objects into K clusters based on the rules under which an object belongs to a cluster with the nearest mean. A summary of the extension of the K-means algorithm could be found in Bock (2007, 2008).

The K-means clustering is also called center-based clustering because of its means-based algorithm. Similar to the K-means clustering, the cluster-wise linear regression method tries to recover the data structure where the objects are clustered into multiple linear regression models. Cluster-wise linear regression partitions the n objects into K subsets where each object belongs to its nearest linear regression model. The cluster-wise linear regression method is one of the most developed clustering methods in statistics. Analogously with the K-means algorithm, Späth (1979, 1981, 1982) partitioned the data into K subsets and fitted K linear regressions such that the total sum of squares of the errors is locally min-

imized. DeSarbo and Cron (1988) utilize the maximum likelihood methodology to choose the appropriate partition that maximizes the likelihood function, which resulted in a fuzzy cluster-wise linear regression method. The assumptions for ordinary linear regression modeling apply to the cluster-wise linear regression. Wedel and Kistemaker (1989) proposed another maximum likelihood methodology by which a particular object can belong to only one cluster. Later, Tibshirani et al. (2001) and Shao and Wu (2005) explored methods of determining the number of clusters for a cluster-wise linear regression clustering approach. Zhang (2003) introduced a K-harmonic means clustering for the cluster-wise linear regression method, which is less sensitive to the choice of the initialization. Rao et al. (2007) and Qian and Wu (2011) extended Späth's (1982) method to one that is more robust by applying an M-estimation for the linear regression modeling.

All of the above methods about the cluster-wise linear regression approach assume that we have a response variable and all other variables are predictor variables. However, for a more general scenario, the data are clustered to hyperplanes and there is not necessarily an identifiable response variable. Van Aelst et al. (2006) proposed the Linear Grouping Algorithm (LGA) using an orthogonal regression method. Van Aelst et al. (2006) also discussed multiple methods of determining the number of groups. To avoid the unexpected effect of extreme values in the data, García-Escudero et al. (2009, 2010) introduced a robust linear clustering using $(1-\alpha)100\%$ of the data (α-trimmed data) to fit the K orthogonal regression models. That is called the Robust Linear Grouping Algorithm (RLGA). We will briefly summarize some of these methods in the remainder of this section.

Given a data set with n independent observations, let Y be the response variable, and $\boldsymbol{X} = (X_1, ..., X_p)$ be the predictor variables. Suppose the true relationship between any given Y and $\boldsymbol{X}$ follows one of the K linear regression models

$$Y_k = \boldsymbol{X}_k \boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k, \quad k = 1, ..., K,$$

where $(\boldsymbol{X}_k, Y_k)$ is the set of observations in the k^{th} cluster. Denote $P = (C_1, ..., C_K)$ with $C_k \cap C_{k'} = \varnothing,\ \forall k \neq k'$ as a partition of the data set, where $C_k = \{1, ..., i_{n_k}\},\ k = 1, ..., K$, is the set of observation indices for the k^{th} cluster. Here, $n_k = |C_k|,\ k = 1, ..., K$, is the number of observations in the k^{th} cluster with $\sum_{k=1}^{K} n_k = n$. In Späth (1979), they solved the following optimization problem

$$\Delta = \underset{P;\ \hat{\boldsymbol{\beta}}_k}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i \in C_k} r_{ki}^2 = \underset{P;\ \hat{\boldsymbol{\beta}}_k}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i=1}^{n_k} r_{ki}^2, \tag{2.25}$$

where $\hat{\boldsymbol{\beta}}_k$ is the coefficient estimate of the k^{th} linear regression model, r_{ki} is the regression residual with $r_{ki} = y_i - \hat{y}_i = y_i - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k$ given $i \in C_k$. The target is to minimize the Δ in equation (2.25) by determining an appropriate partition P. The estimation for $\boldsymbol{\beta}$ follows the least squares estimation for a linear regression model. The optimal partition will satisfy

$$C_k = \{(\boldsymbol{x}, y) | (y - \boldsymbol{x}^T \hat{\boldsymbol{\beta}}_k) \leq (y - \boldsymbol{x}^T \hat{\boldsymbol{\beta}}_{k'})\},\ \forall k \neq k'\}.$$

Späth (1979) proposed an algorithm to find the local optima of equation (2.25) as follows:

(*i*) *Initialization*: Initialize the K regressions randomly or define them based on some prior knowledge.

(*ii*) *Clustering*: Calculate the residuals between all the observations and each of the K regression models, and allocate the observations to their closest model. The distance between the i^{th} observation and the k^{th} regression model is defined as $|y_i - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k|$. The updated partition is $P^{(l)} = (C_1^{(l)}, ..., C_K^{(l)})$, where l is the number of iterations.

(*iii*) *Regression*: For $k = 1, ..., K$, fit linear regression models within each of the K clusters for the partition $P^{(l)} = (C_1^{(l)}, ..., C_K^{(l)})$. Let $l = l + 1$.

(*iv*) *Stop*: Repeat (*ii*) and (*iii*) until there is no more data point changing its membership.

According to Zhang (2003), the steps (*ii*) and (*iii*) are a monotone decreasing process and the Δ in equation (2.25) eventually converges to a local minimum.

For the same problem, DeSarbo and Cron (1988) presented a maximum likelihood methodology using a mixture of conditional normal distributions. Let Y be the response variable of a data set with n independent observations, and let $\boldsymbol{X} = (X_1, ..., X_p)$ be the predictor variables. Let $\lambda_k, k = 1, ..., K$, be the unknown proportions of the conditional normal distributions that comprise the mixture distribution. Suppose $(\boldsymbol{x}_i, y_i), i = 1, ..., n$, is the i^{th} observation. Let $f_{ik}(y_i|\boldsymbol{x}_i, \sigma_k^2, \boldsymbol{\beta}_k)$ be the probability density function (*pdf*) of the conditional distribution of y_i. Here $f_{ik}(\cdot)$ is assumed to be the *pdf* of the k^{th} normal distribution $N(\boldsymbol{x}_i\boldsymbol{\beta}_k, \sigma_k^2)$, where $\boldsymbol{\beta}_k = (\beta_{0k}, ..., \beta_{pk})$ is the set of regression coefficients for the k^{th} linear regression model and σ_k^2 is the variance of the normal distribution. Then, y_i follows a finite sum or mixture of conditional univariate normal distributions:

$$\begin{aligned} y_i &\sim \sum_{k=1}^{K} \lambda_k f_{ik}(y_i|\boldsymbol{x}_i, \sigma_k^2, \boldsymbol{\beta}_k) \\ &= \sum_{k=1}^{K} \lambda_k (2\pi\sigma_k^2)^{-1/2} \exp\left\{-\frac{(y_i - \boldsymbol{x}_i\boldsymbol{\beta}_k)^2}{2\sigma_k^2}\right\}, \end{aligned} \tag{2.26}$$

where the λ_k and σ_k^2 satisfy $0 \leq \lambda_k \leq 1$, $\sum_{k=1}^{K} \lambda_k = 1$, and $\sigma_k^2 > 0$. Then, the log likelihood function of the mixture distribution is

$$\ln(L) = \sum_{i=1}^{n} \ln\left\{\sum_{k=1}^{K} \lambda_k (2\pi\sigma_k^2)^{-1/2} \exp\left[-\frac{(y_i - \boldsymbol{x}_i\boldsymbol{\beta}_k)^2}{2\sigma_k^2}\right]\right\}. \tag{2.27}$$

DeSarbo and Cron (1988) applied an EM algorithm to obtain the estimated optimal values of λ_k, σ_k^2, and $\boldsymbol{\beta}_k$ that maximize the log likelihood function in equation (2.27). Once the estimated optimal values $\hat{\lambda}_k$, $\hat{\sigma}_k^2$, and $\hat{\boldsymbol{\beta}}_k$ are obtained, each observation in the data set can be assigned to a cluster k based on its estimated posterior probability

$$\hat{p}_{ik} = \frac{\hat{\lambda}_k f_{ik}(y_i|\boldsymbol{x}_i, \hat{\sigma}_k^2, \hat{\boldsymbol{\beta}}_k)}{\sum_{k=1}^{K} \hat{\lambda}_k f_{ik}(y_i|\boldsymbol{x}_i, \hat{\sigma}_k^2, \hat{\boldsymbol{\beta}}_k)}. \tag{2.28}$$

Unlike the K-means cluster-wise linear regression algorithm of Späth (1979) where a particular observation i belongs to only one cluster, the maximum likelihood cluster-wise linear regression method gives each observation a probability of belonging to the cluster k. DeSarbo and Cron (1988) then discussed a method of determining the number of clusters by the Akaike Information Criterion (Akaike, 1973).

Hennig (1996, 1999) further developed DeSarbo and Cron (1988)'s maximum likelihood method for cluster-wise linear regression. The details could be found in Hennig (1996, 1999).

Wedel and Kistemaker (1989) developed another maximum likelihood methodology for cluster-wise linear regression methods. Instead of giving a proportion to each cluster k for an observation as in DeSarbo and Cron (1988), Wedel and Kistemaker (1989) assumed that a particular observation could belong to only one cluster. Let $n_k = |C_k|$ be the number of observations in the cluster C_K given a partition $P = (C_1, ..., C_K)$, and let $(\boldsymbol{X}_k, Y_k)$ be the set of observations in the k^{th} cluster, where $\boldsymbol{X}_k$ is an $n_k \times p$ matrix and Y_k is an $n_k \times 1$ vector. Within each cluster, the relationship between Y_k and $\boldsymbol{X}_k$ can be modeled by a linear regression model with coefficient $\boldsymbol{\beta}_k$ and the variance of error σ_k^2. Then, the log likelihood function is given by

$$\ln(L) = \sum_{k=1}^{K} \ln\left((2\pi\sigma^2)^{-\frac{n_k}{2}} \exp\left\{ \frac{(Y_k - \boldsymbol{X}_k\boldsymbol{\beta}_k)'(Y_k - \boldsymbol{X}_k\boldsymbol{\beta}_k)}{2\sigma_k^2} \right\} \right). \tag{2.29}$$

For a possible partition, the optimal values of $\boldsymbol{\beta}_k$ and σ_k^2 that maximize the log likelihood function in equation (2.29) are the ordinary least squares estimators $\hat{\boldsymbol{\beta}}_k$ and $\hat{\sigma}_k^2$. Given the number of observations of a data set, the number of possible partitions for the K clusters is finite. Thus, by comparing all the possible partitions, the one that maximizes the log likelihood function in equation (2.29) can be obtained. However, for a large number

of observations, the computation time would be huge and therefore sometimes it will be impossible to investigate all the possible partitions. Wedel and Kistemaker (1989) applied a transfer algorithm proposed by Banfield and Bassil (1977) which made this maximum likelihood estimation possible for a large number of observations.

Tibshirani et al. (2001) proposed a Gap statistic to determine the appropriate number of clusters K for cluster analyses. Let $D_k = \sum_{i,i' \in C_k} d_{i,i'}$ be the sum of pairwise distances for all points in cluster k, where $d_{i,i'}$ is the distance between observation i and i'. Define W_K as

$$W_K = \sum_{k=1}^{K} \frac{1}{2n_k} D_k. \tag{2.30}$$

In equation (2.30), W_K is a pooled within-cluster average of pairwise distances. Given an appropriate reference distribution of the n observations in the data set, the Gap statistic is defined as

$$\text{Gap}_n(K) = E_n^*(\log(W_K^*)) - \log(W_K), \tag{2.31}$$

where $E_n^*(\log(W_K^*))$ denotes the expectation of $\log(W_K^*)$ under a n-observation reference data set, W_K^* is obtained the same way as W_K in (2.30) but on the reference data set. The n-observation reference data set is usually draw from p uniform distributions where the ranges of each uniform distributions is the range of each of the p variables in the original data set. The Gap statistic compares the log pooled within-cluster average of pairwise distances with its expectation under a null reference distribution of the data. Then, the optimal value K will be the value that maximizes the Gap statistic. Though the Gap statistic is not particularly designed for cluster-wise regression methods, it is a very flexible method that could be easily applied to the cluster-wise regression method.

Shao and Wu (2005) developed an information-based criterion to determine the number of clusters for the cluster-wise linear regression method. They introduced a penalized term to the objective function that is the aggregated sum of squares of the errors of the linear

regression models. When minimizing the objective function, the penalized term, which is a increasing function of K, prevents K from being too large.

All the methods we have discussed in this section so far explicitly specify Y as the response variable, but this is not necessarily always the case. When there is no specific response variable, we can still implement a cluster-wise linear regression model by randomly assigning one of the variables as the response variable. However, the clustering results could depend on the choice of the response variable. Van Aelst et al. (2006) applied an orthogonal regression method for the cluster-wise linear regression methodology so that it is unnecessary to specify a response variable. For a data set, the orthogonal regression method obtains a hyperplane that is orthogonal to the smallest principle component of the whole data set. Specifically, given a data set Ω with variables $\boldsymbol{X} = (X_1, ..., X_p)$, the hyperplane is defined as

$$\boldsymbol{a}^T \boldsymbol{x} = b, \tag{2.32}$$

where $\boldsymbol{a}$ is the eigenvector corresponding to the smallest eigenvalue of the data set, while $b \equiv \boldsymbol{a}^T \bar{\boldsymbol{X}}$ with $\bar{\boldsymbol{X}} = (\bar{X}_1, ..., \bar{X}_p)$ being the mean vector of the data set. The process of seeking the K linear regressions is similar to the algorithm in Späth (1979). The algorithm is briefly summarized as follows:

(i) *Standardization*: All the variables are standardized so that they all have zero mean and unit variance (this is usually, but not always, necessary for implementing Principal Component Analysis (PCA)).

(ii) *Initialization*: Randomly draw K mutually exclusive subsets from the data set. For each subset, calculate the orthogonal regression hyperplane. The K hyperplanes are the initial regressions.

(iii) *Clustering*: Calculate the orthogonal distance between each data point and the K

hyperplanes. Assign a point to its closest hyperplane.

(*iv*) *Regression*: For $k = 1, ..., K$, update the orthogonal regression hyperplane for each group.

(*v*) *Stop*: Repeat (*iii*) and (*iv*) until there is no more data point changing its membership.

Van Aelst et al. (2006) then discussed how to determine the number of groups by using Gap statistics or a likelihood function with a penalty term.

García-Escudero et al. (2009) proposed a Robust Linear Grouping Algorithm (RLGA) that implements a linear grouping algorithm in the presence of outliers. Given a particular partition $P = (C_1, ..., C_K)$, the method uses a $(1 - \alpha)100\%$ proportion subsample that has the smallest orthogonal distance with its closest regression line. Here, $0 \leq \alpha < 1$ is a predetermined proportion. Then, update the K orthogonal regression methods based on the subsample. Given a data set with n observations, the RLGA is as follows:

(*i*) *Initialization*: Randomly draw K mutually exclusive subsets as the initial K clusters $P = (C_1, ..., C_K)$, and calculate the orthogonal regression hyperplane for each cluster.

(*ii*) *Clustering*: Assign each observation $\boldsymbol{x}_i, i = 1, ..., n$, to its closest cluster in terms of the orthogonal distance. Compute the orthogonal distance, d_i, between each observation and its closest cluster among the K clusters. The orthogonal distance d_i is defined as

$$d_i = \inf_{k=1,...,K} \|(I - U_k U_k')(\boldsymbol{x}_i - \bar{\boldsymbol{X}}_k)\|^2, \tag{2.33}$$

where U_k is a $p \times p$ matrix with columns being the p unit eigenvectors of the sample covariance matrix of the n_k observations in the k^{th} cluster, and $\bar{\boldsymbol{X}}_k$ is the mean vector of the k^{th} cluster, $k = 1, \ldots, K$. Here we assume that the matrix $\boldsymbol{X} = (X_1, ..., X_p)$ is of full rank. If the rank of $\boldsymbol{X}$ is $d < p$, then U_k has d columns.

Keep the set $H = (H_1, ..., H_K)$ with $n(1-\alpha)$ observations that have the smallest distance with its regression line, where $H_k \subseteq C_k,\ k = 1, ..., K$.

(*iii*) *Regression*: For $k = 1, ..., K$, calculate the orthogonal regression hyperplane for the observations within H_k.

(*iv*) *Stop*: Repeat (*ii*) and (*iii*) until there is no more data point changing its membership.

Based on the simulation results, the method performs much better than does the LGA in the presence of outliers.

2.4 Cluster Analysis for Symbolic Data

Cluster analysis is one of the most popular topics for symbolic data analysis, especially for interval-valued data. Gowda and Diday (1991a) proposed measurements of dissimilarities for interval-valued data (quantitative) and multi-valued data (qualitative), so that clustering could be easily implemented. Their dissimilarity comprises three components, dissimilarity components due to position, span, and content. The dissimilarity between two interval-valued objects is defined as the sum of the three components. The dissimilarity between two multi-valued objects is defined as the sum of dissimilarities due to span and content. At each step, a pair of symbolic objects is selected for agglomeration based on minimum dissimilarity. Analogously with the same methodology, Gowda and Diday (1991b) defined a similarity by three components, similarity components due to position, span and content. An agglomerative symbolic clustering was implemented to samples from a mixture of multivariate normal distributions. At each step, a pair of symbolic objects with highest similarity is agglomerated. Using the same three components, Gowda and Ravi (1995) modified the definition of dissimilarity and similarity, and used both measurements of similarity and dissimilarity for agglomerative clustering. This avoids disadvantages of the algorithm in Gowda

and Diday (1991a,b), for which the clustering results based on dissimilarity measures could be different from the results based on similarity measures.

By following the approach of Chavent (1998), Kim and Billard (2011) introduced a criterion-based divisive clustering method. At each divisive step, a cluster C_i is selected to be divided such that the difference between the inertia (or the within-cluster variance) of C_i and the sum of the inertia of the two partitioned clusters, C_i^1, C_i^2, is maximized. Then, for an interval $x \in C_i$, if its center, the average of upper and lower bounds, is smaller or equal to a cut point c, then, $x \in C_i^1$, else $x \in C_i^2$. This is a monothetic divisive clustering method where only one variable is considered at a time.

The K-means and adaptive K-means like clustering methods were developed in many publications. Chavent and Lechevallier (2002) proposed a dynamical clustering for interval-valued data using the Hausdorff distance. De Souza et al. (2004), de Souza and de Carvalho (2004), and de Carvalho et al. (2006a) used similar algorithms but extended them to the city-block distance, Mahalanobis distance, and L_2 distance. De Souza and de Carvalho (2004), de Carvalho et al. (2006b), and de Carvalho and Lechevallier (2009) applied the adaptive K-means like algorithm for clustering of interval-valued data. All these methods essentially implemented the clustering algorithms that were originally proposed by Diday and Simon (1976). We will briefly summarize some of these clustering methods.

The main process of the clustering algorithms proposed by Diday and Simon (1976) is iterative in two steps, the representation step and the allocation step. The representation step calculates the center of a cluster, such as its mean or median. The allocation step allocates an object to its closest cluster in terms of its distance to the cluster center, where the definition of distance could be different according to the data types. Diday and Simon's algorithm is an extension of the K-means algorithm. Given a data set Ω, a possible partition of the data set is denoted as $P = (C_1, ..., C_K)$. Then, the clustering algorithm is given by:

(i) *Initialization*: Choose a partition $P = (C_1, ..., C_K)$ of Ω randomly from all the possible

partitions, denote the center of the K clusters as $(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K)$.

(*ii*) *Allocation*: Assign each observation to its closest cluster in terms of its distance to the cluster centers, $\boldsymbol{\mu}_k$, $k = 1, ..., K$.

(*iii*) *Representation*: For $k = 1, ..., K$, calculate the center of cluster k such that $\boldsymbol{\mu}_k$ minimizes an objective function.

(*iv*) *Stop*: Repeat (*ii*) and (*iii*) until there is no more observation changing its membership.

When these clustering algorithms apply to interval-valued data, the main difference is in defining a distance between two intervals. Chavent and Lechevallier (2002) investigated the Hausdorff distance, defined between two intervals $x_1 = [x_{1a}, x_{1b}]$ and $x_2 = [x_{2a}, x_{2b}]$ as

$$d_H(x_1, x_2) = \max\{|x_{1a} - x_{2a}|, |x_{1b} - x_{2b}|\}. \tag{2.34}$$

The distance between two p-dimension observations, $\boldsymbol{x}_1 = (x_{11}, ..., x_{1p})$ and $\boldsymbol{x}_2 = (x_{21}, ..., x_{2p})$, is defined as the L_1 norm Hausdorff distance

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{j=1}^{p} \max\{|x_{1ja} - x_{2ja}|, |x_{1jb} - x_{2jb}|\} = \sum_{j=1}^{p} d_H(x_{1j}, x_{2j}). \tag{2.35}$$

Given a data set $X = (X_1, ..., X_p)$, let $\boldsymbol{x}_i = (x_{i1}, ..., x_{ip})$ be the i^{th} observation with $x_{ij} = [x_{ija}, x_{ijb}]$, $j = 1, ..., p$. Let $x_{ij}^c = (x_{ija} + x_{ijb})/2$ be the center point of the interval x_{ij}, and let $x_{ij}^r = x_{ijb} - x_{ija}$ be the range of x_{ij}, and let $x_{ij}^\delta = x_{ij}^r/2$ be the radius. Given the cluster center $\boldsymbol{\mu}_k = (\mu_{k1}, ..., \mu_{kp})$ with $\mu_{kj} = [\mu_{kja}, \mu_{kjb}]$, $j = 1, ..., p$, of the cluster k, the objective function, $f(\boldsymbol{\mu}_k)$, is the sum of the distances between each observation in the cluster and the cluster center $\boldsymbol{\mu}_k$, i.e.,

$$f(\boldsymbol{\mu}_k) = \sum_{i \in C_k} \sum_{j=1}^{p} d_H(x_{ij}, \mu_{kj}). \tag{2.36}$$

We need to find a $\boldsymbol{\mu}_k$ that minimizes the objective function $f(\boldsymbol{\mu}_k)$. The problem is equivalent to finding $\mu_{kj} = [\mu_{kja}, \mu_{kjb}]$, $j = 1, ..., p$, which minimizes

$$\tilde{f}(\mu_{kj}) = \sum_{i \in C_k} d_H(x_{ij}, \mu_{kj}) = \sum_{i \in C_k} \max(|x_{ija} - \mu_{kja}|, |x_{ijb} - \mu_{kjb}|). \tag{2.37}$$

Denote $\mu^c_{kj} = (\mu_{kja} + \mu_{kjb})/2$, and $\mu^\delta_{kj} = (\mu_{kjb} - \mu_{kja})/2$. Then, equation (2.37) is minimized when

$$\begin{gathered} \mu^c_{kj} = \text{median}\{x^c_{ij} | i \in C_k\},\ \mu^\delta_{kj} = \text{median}\{x^\delta_{ij} | i \in C_k\}, \\ k = 1, ..., K,\ j = 1, ..., p. \end{gathered} \tag{2.38}$$

For $j = 1, ..., p$, the optimal value for μ^c_{kj} is the median of the center points of intervals that are in the cluster k. Similarly, the optimal value for μ^δ_{kj} is the median of the radii of those intervals that are in the cluster k. The allocation step then assigns each object to its closest cluster in terms of the Hausdorff distance.

De Souza and de Carvalho (2004) applied the city-block distances to Diday and Simon's algorithm. The city-block distance between two intervals x_1 and x_2 for a given variable is defined as

$$d_C(x_1, x_2) = |x_{1a} - x_{2a}| + |x_{1b} - x_{2b}|. \tag{2.39}$$

The city-block distance between two p-dimension observations, $\boldsymbol{x}_1 = (x_{11}, ..., x_{1p})$ and $\boldsymbol{x}_2 = (x_{21}, ..., x_{2p})$ is defined as

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{j=1}^{p} d_C(x_{1j}, x_{2j}). \tag{2.40}$$

The objective function is the same as the equation (2.36) but using the city-block distance instead of the Hausdorff distance. The objective function is minimized when

$$\begin{gathered} \mu_{kja} = \text{median}\{x_{ija} | i \in C_k\},\ \mu_{kjb} = \text{median}\{x_{ijb} | i \in C_k\}, \\ k = 1, ..., K,\ j = 1, ..., p. \end{gathered} \tag{2.41}$$

De Souza and de Carvalho (2004) then developed an adaptive distance measure between two interval vectors based on the city-block distance. For the k^{th} cluster, the adaptive distance between an observation $\boldsymbol{x}_i$ and the cluster center $\boldsymbol{\mu}_k$ is given by

$$d_k(\boldsymbol{x}_i, \boldsymbol{\mu}_k) = \sum_{j=1}^{p} (\lambda_{kja}|x_{ija} - \mu_{kja}| + \lambda_{kjb}|x_{ijb} - \mu_{kjb}|), \tag{2.42}$$

where λ_{kja} and λ_{kjb} are weights for the j^{th} variable in the k^{th} cluster, $j = 1, ..., p$, $k = 1, ..., K$. If $\lambda_{kja} = \lambda_{kjb}$, equation (2.42) is a one-component adaptive distance; if $\lambda_{kja} \neq \lambda_{kjb}$, it is a two-component adaptive distance. The λ_{kja} and λ_{kjb} are subject to

$$\begin{aligned} &(1) \quad \lambda_{kja}, \lambda_{kjb} > 0, \\ &(2) \quad \prod_{j=1}^{p} \lambda_{kja} = \prod_{j=1}^{p} \lambda_{kjb} = 1. \end{aligned}$$

The objective function $f(\cdot)$ is defined similarly to that in equation (2.36). The optimal values of these three terms are obtained by fixing two of them each time and minimizing the objective function. The optimal values of $\boldsymbol{\mu}_k$ are still the medians of the upper bounds and the lower bounds of the intervals in the cluster k. The optimal values for λ_{kja}, λ_{kjb} are given by, for $k = 1, ..., K$,

$$\lambda_{kja} = \frac{\left[\prod_{h=1}^{p} (\sum_{i \in C_k} |x_{iha} - \mu_{kha}|)\right]^{\frac{1}{p}}}{\sum_{i \in C_k} |x_{ija} - \mu_{kha}|}, \qquad \lambda_{kjb} = \frac{\left[\prod_{h=1}^{p} (\sum_{i \in C_k} |x_{ihb} - \mu_{khb}|)\right]^{\frac{1}{p}}}{\sum_{i \in C_k} |x_{ijb} - \mu_{khb}|}. \tag{2.43}$$

From equation (2.43), the values of λ_{kja} and λ_{kjb} give more weight to the variables that are closer to the cluster center in terms of the upper and lower bounds. The involvement of the weights affects the cluster membership for each object and consequently the clustering results.

The same clustering algorithm and the idea of an adaptive distance can be expanded

to all other distance definitions that apply to interval-valued data. De Souza et al. (2004) applied the algorithm to a Mahalanobis distance for interval-valued data. Chavent et al. (2006) split n intervals of a variable x_j into elementary intervals and added a weight to each interval. Chavent et al. then used the elementary intervals and weights to create a two components dissimilarity measure with components due to position and weights. That dissimilarity measure was used for interval-valued data clustering. The details could be found in Chavent et al. (2006). De Carvalho et al. (2006a) applied a L_2 norm distance, while de Carvalho et al. (2006b) adopted an adaptive Hausdorff distance for interval-valued data clustering. De Carvalho and Lechevallier (2009) proposed an adaptive distance where the weights are the same among the K clusters but different between the p variables for each step.

De Carvalho (2007), analogously with the methodology for classical data, proposed a fuzzy c-means clustering method for symbolic interval data. For fuzzy c-means clustering, each object belongs to a cluster by a proportion (or a probability). The details can be found in de Carvalho (2007).

De Carvalho et al. (2010) developed a cluster-wise regression model for interval-valued data using the center and range method. The algorithm is analogous to the cluster-wise linear regression in Späth (1979). Within each cluster, de Carvalho et al. (2010) fitted one linear regression for the center points and one for the ranges of the intervals, respectively. As described in section 2.2, CRM minimizes the sum of squares of the errors $\sum_{i\in C_k}(\epsilon_i^c)^2$ and $\sum_{i\in C_k}(\epsilon_i^r)^2$, respectively. The objective function at the representation step of the algorithm is

$$f(\hat{\boldsymbol{\beta}}_k^c, \hat{\boldsymbol{\beta}}_k^r) = \sum_{i\in C_k}((\epsilon_i^c)^2 + (\epsilon_i^r)^2), \quad k = 1, ..., K,$$

where $\hat{\boldsymbol{\beta}}_k^c$ and $\hat{\boldsymbol{\beta}}_k^r$ are the coefficient estimates for the regression of the center points and the regression of ranges, respectively. For the allocation step, an observation is assigned to the

cluster that has the smallest sum of squares of the errors from the center point regression and range regression models, $((\epsilon_i^c)^2 + (\epsilon_i^r)^2)$.

Kim and Billard (2011) developed a polythetic divisive clustering algorithm for p-dimensional histogram-valued data. Kim and Billard (2012) proposed dissimilarity measures for the symbolic multimodal-valued data and a divisive clustering algorithm for the data. Kim and Billard (2013) gave some dissimilarity measures for histogram-valued data.

2.5 Likelihood Functions and Maximum Likelihood Estimators for Symbolic Data

Likelihood functions for interval-valued symbolic data were first developed by Le-Rademacher and Billard (2011). Let $x_i, i = 1, ..., n$, be a random sample of the interval-valued random variable X. Let f_i be the internal density of x_i. For an interval-valued random variable, f_i is usually assumed to be the probability distribution function (*pdf*) of a continuous uniform distribution. That is, for $x_i = [x_{ia}, x_{ib}]$, let $\xi \in [x_{ia}, x_{ib}]$ with $x_{ia}, x_{ib} \in \Re$. Then, if we assume $\xi \in [x_{ia}, x_{ib}]$ are uniformly distributed across $[x_{ia}, x_{ib}]$, we have $f_i(\xi) = 1/(x_{ib} - x_{ia})$. Let $\boldsymbol{\theta}_i$ be the parameter vector of f_i where $\boldsymbol{\theta}_i$ is defined to have a one-to-one relationship between x_i and $\boldsymbol{\theta}_i$. Suppose the distribution of $\boldsymbol{\Theta}$ is known with probability density function $g(\boldsymbol{\theta}_i; \boldsymbol{\tau})$, where $\boldsymbol{\theta}_i$ is the i^{th} realization of $\boldsymbol{\Theta}$ with $i = 1, ..., n$, and $\boldsymbol{\tau}$ is the parameter vector of the distribution. Since x_i is uniquely identified by $\boldsymbol{\theta}_i$, we have $P(X = x_i) = P(\boldsymbol{\Theta} = \boldsymbol{\theta}_i)$. The likelihood function can be obtained based on this relationship.

For interval-valued data, assume that $\boldsymbol{\Theta} = (\Theta_1, \Theta_2)$ with $\Theta_1 \sim \mathrm{N}(\mu, \sigma^2)$, a normal distribution, and $\Theta_2 \sim \exp(\beta)$, an exponential distribution. Let the i^{th} realization of X be x_i, and let the i^{th} realization of $\boldsymbol{\Theta} = (\Theta_1, \Theta_2)$ be $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2})$. Let θ_{i1} be the internal mean of x_i, and let θ_{i2} be the internal variance of x_i. Then, θ_i and x_i have a one-to-one correspondence with $\theta_{i1} = (x_{ia} + x_{ib})/2$ and $\theta_{i2} = (x_{ib} - x_{ia})^2/12$. Assume that Θ_1 and Θ_2

are independent, then, the probability density function of $\boldsymbol{\Theta}$ is given by

$$\begin{aligned} g(\boldsymbol{\theta}_i; \mu, \sigma^2, \beta) &= g_1(\theta_{i1}; \mu, \sigma^2) g_2(\theta_{i2}; \beta) \\ &= g_1(\frac{x_{ia} + x_{ib}}{2}; \mu, \sigma^2) g_2(\frac{(x_{ib} - x_{ia})^2}{12}; \beta), \end{aligned} \tag{2.44}$$

where $g_1(\cdot)$ is the *pdf* of the normal distribution and $g_2(\cdot)$ is the *pdf* of the exponential distribution.

The likelihood function of $\boldsymbol{\tau} = (\mu, \sigma^2, \beta)$ for a random sample $x_1, ..., x_n$ of X is given by

$$L(\mu, \sigma^2, \beta; \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n) = \prod_{i=1}^{n} g(\boldsymbol{\theta}_i; \mu, \sigma^2, \beta).$$

After some algebra, the maximum likelihood estimator (MLE) for the parameters are

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \frac{x_{ia} + x_{ib}}{2}, \tag{2.45}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (\frac{x_{ia} + x_{ib}}{2} - \hat{\mu})^2, \tag{2.46}$$

$$\hat{\beta} = \frac{1}{n} \sum_{i=1}^{n} \frac{(x_{ib} - x_{ia})^2}{12}. \tag{2.47}$$

Intuitively, we assume that the center points, $\theta_{i1} = (x_{ia} + x_{ib})/2$, of the random sample $x_1, ..., x_n$ follow a normal distribution with mean μ and variance σ^2, and the internal variance, $\theta_{i2} = (x_{ib} - x_{ia})^2/12$, follows an exponential distribution with mean β, and the two distributions are independent. Then, the MLE of the two parameters for the normal distribution are the sample mean and sample variance of the center points. The MLE of β for the exponential distribution is the sample mean of the internal variance. The MLEs are coincident with the classical cases. The details of the MLE given that the normal and exponential distributions are dependent could be found in Le-Rademacher and Billard (2011).

Chapter 3

Cluster-wise Regression for Interval-Valued Data

In chapter 2, we reviewed the development of cluster-wise regression methods for classical data and clustering analysis for symbolic interval-valued data. In this chapter, we introduce cluster-wise regression for symbolic interval-valued data. It can be misleading to assume there is only one linear regression model for the whole data set. In addition, if the population is clustered onto multiple hyperplanes, the K-means like clustering for symbolic intervals (Chavent and Lechevallier, 2002, de Souza and de Carvalho, 2004, de Souza et al., 2004, de Carvalho et al., 2006a,b, de Carvalho and Lechevallier, 2009) is not able to recover such a data structure. Unlike for classical data, the method of cluster-wise linear regression methodology for symbolic data is rarely studied. In this chapter, we propose methods for implementing cluster-wise regression methods for symbolic interval-valued data.

The remainder of this chapter is arranged as follows. Section 3.1 formally introduces the problem and gives the relevant assumptions. Section 3.2 proposes a algorithm to implement the cluster-wise regression methodology by adapting K-regressions clustering techniques. Section 3.3 gives a method to determine the optimal number of clusters K. The simulation

methodology of the interval-valued data is studied in section 3.4. In section 3.5, several simulated data sets are used to implement the cluster-wise regression algorithm by the K-regressions algorithm and the performance is evaluated.

3.1 Introduction

Suppose we have n observations in a data set with response variable Y and p predictor variables $X_1, ..., X_p$. Denote the p-dimensional predictor variables as $(X_1, ..., X_p) \equiv \boldsymbol{X}$. All the response and predictor variables are interval-valued random variables as defined in definition 2.1.2. Let x_{ij}, $i = 1, ..., n$, $j = 1, ..., p$, be the i^{th} observation for the j^{th} predictor variable X_j, denoted by $x_{ij} = [x_{ija}, x_{ijb}]$ with $x_{ija}, x_{ijb} \in \Re$ and $x_{ija} \leq x_{ijb}$. Similarly, let y_i be the i^{th} observation for the response variable Y, denoted by $y_i = [y_{ia}, y_{ib}]$ with $y_{ia} \leq y_{ib}$. Assume that the response variable Y has K different linear relationships with the predictor variables $\boldsymbol{X}$, where K is a fixed number. Let $(\boldsymbol{X}_k, Y_k)$, $k = 1, ..., K$, be the set of observations that follow the k^{th} regression model (or belong to the k^{th} cluster). Then,

$$Y_k = \boldsymbol{X}_k \boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k, \quad k = 1, ..., K, \tag{3.1}$$

where $\boldsymbol{\beta}_k$ is the set of linear coefficients of the p predictor variables for the k^{th} regression model, and $\boldsymbol{\epsilon}_k$ is the error interval vector.

Let n_k, $k = 1, ..., K$, be the number of observations in the k^{th} cluster with $\sum_{k=1}^{K} n_k = n$. We assume the following:

(A) The number of observations n_k satisfies $p < n_k \leq n$, $k = 1, ..., K$, where p is the number of predictor variables, and n is the total number of observations in the whole data set. It can be shown that $n_k = n$ only if $K = 1$.

(B) The individual error intervals in a particular cluster k are drawn independently from

a normal distribution with mean 0 and standard deviation σ_k^2, $N(0, \sigma_k^2)$. The error intervals $\boldsymbol{\epsilon}_k$ are independent from $\boldsymbol{\epsilon}_{k'}$, given $k \neq k'$, for $k, k' = 1, ..., K$.

The first assumption (A) avoids the situation with $n_k < p$ such that there is no linear regression solution for the k^{th} cluster; while the second assumption (B) reduces the computational complexity of the problem. Our goal is to find an optimal partition $P^* = (C_1^*, ..., C_K^*)$ that minimizes the overall residuals of the regression models given the number of clusters K.

3.2 Cluster-wise Regression by K-regressions Clustering: Algorithm

In this section, utilizing the Symbolic Variation Method (SVM) for linear regression of interval-valued data, we propose a K-regressions clustering algorithm to recover the data structure in equation (3.1).

Given a partition $P = (C_1, ..., C_K)$, we can fit a linear regression model for each cluster as in equation (3.1). Denote the coefficient estimate of $\boldsymbol{\beta}_k$ as $\hat{\boldsymbol{\beta}}_k$ for $k = 1, ..., K$. Then, the regression residuals for the k^{th} cluster are defined as

$$r_{ki} = d(y_i, \hat{y}_i) \tag{3.2}$$

given $i \in C_k$. Here, $d(y_i, \hat{y}_i)$ stands for the distance between the observation y_i and its predicted interval $\hat{y}_i$. Since $\hat{y}_i = \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k$, the equation (3.2) can be rewritten as $r_{ki} = d(y_i, \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k)$. The predictive interval $\hat{y}_i$ using the SVM method is

$$\hat{y}_i = [\hat{y}_{ia}, \hat{y}_{ia}] = [\min_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k, \max_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k], \tag{3.3}$$

for $i \in C_k$, where $\mathbb{X} = \{\boldsymbol{x} = (x_j) : x_{ja} \leq x_j \leq x_{jb}, j = 1, ..., p\}$. Our goal is to find an

optimal partition that minimizes the sum of squared residuals (SSR) given K:

$$SSR = \underset{P;\ \hat{\boldsymbol{\beta}}_k}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i \in C_k} r_{ki}^2 = \sum_{k=1}^{K} \sum_{i=1}^{n_k} r_{ki}^2. \tag{3.4}$$

Since r_{ki} in (3.4) is defined as the distance between two intervals, y_i and $\hat{y}_i$, different definitions of the distance between these two intervals variables will affect the clustering results of K-regressions algorithm. We consider three different distance definitions between two interval variables: center distance, Hausdorff distance and city-block distance. The center distance between two p-dimensional interval observations $\boldsymbol{x}_1 = (x_{11}, \ldots, x_{1p})$ and $\boldsymbol{x}_2 = (x_{21}, \ldots, x_{2p})$ with $x_{ij} = [x_{ija}, x_{jib}]$, $i = 1, 2$, $j = 1, \ldots, p$, is defined as

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{j=1}^{p} |x_{1j}^c - x_{2j}^c|, \tag{3.5}$$

where $x_{ij}^c = (x_{ija} + x_{ijb})/2$, is the center point of x_{ij}, $i = 1, 2$, $j = 1, \ldots, p$. The Hausdorff distance is defined as in equations (2.34); the city-block distance is defined as in equation (2.39). The algorithm of the K-regressions for the three distance definitions are the same. Analogously with the algorithm in Späth (1979), we propose the K-regressions cluster-wise regression for interval-valued data as follows:

(i) *Initialization*: Choose a partition $P = (C_1, ..., C_K)$ randomly from all the possible partitions, or partition the whole data set to K clusters based on some prior knowledge.

(ii) *Representation*: For $k = 1, ..., K$, fit regressions $Y_k = \boldsymbol{X}_k \boldsymbol{\beta}_k + \boldsymbol{\epsilon}$ to the observations in each of the K clusters for partition $P^{(l)} = (C_1^{(l)}, ..., C_K^{(l)})$, where l stands for the l^{th} iteration.

(iii) *Allocation*: For observation i, $i = 1, ..., n$, calculate its distance to each of the K regression lines, $d(y_i, \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k)$, $k = 1, ..., K$, and allocate the observation to its closest

line. The updated partition is now $P^{(l+1)} = (C_1^{(l+1)}, ..., C_K^{(l+1)})$.

(*iv*) *Stop*: Repeat (*ii*) and (*iii*) until the improvement of SSR in equation (3.4) is smaller than a predetermined criterion, or the number of iterations reaches a predetermined maximum number.

For the representation step, we apply the SVM to fit the linear regression model. For the allocation step, the observations are allocated such that, for $k = 1, ..., K$,

$$C_k = \{(\boldsymbol{x}, y) | d(y, \boldsymbol{x}^T \hat{\boldsymbol{\beta}}_k) \leq d(y, \boldsymbol{x}^T \hat{\boldsymbol{\beta}}_{k'}), \forall k \neq k'\}. \tag{3.6}$$

Given a data set, the algorithm cannot guarantee a global minimum of SSR. Thus, we repeatedly implement the steps (*i*)-(*iv*) a number of times and select the solution which has the lowest value of SSR. The selected partition can be further iterated until SSR cannot be reduced anymore.

3.3 Determine the Number of Clusters K

The K-regressions clustering algorithm is to implement the cluster-wise regression method given that K is known. However, if we do not have prior knowledge about K, a bad guess of K can mislead the clustering results. Xu (2010) gave a symbolic R-square (R^2) of the SVM for the linear regression of interval-valued data,

$$R^2 = \frac{\text{Var}(\hat{Y})}{\text{Var}(Y)}, \tag{3.7}$$

where $\hat{Y}$ is the predictive vector of response variable Y, and $\text{Var}(\cdot)$ is the symbolic variance. Using the symbolic R^2, we propose methods to determine the number of clusters K.

Given a predetermined maximum number of clusters K^{max}, for each $K = 1, \ldots, K^{max}$, calculate the R^2 for each cluster $k = 1, \ldots, K$, denoted by $R_k^{2(K)}$. For the whole data set,

the weighted average R^2 for the n observations given the number of clusters K is defined as

$$R^{2^{(K)}} = \sum_{k=1}^{K} w_k^{(K)} R_k^{2^{(K)}}, \tag{3.8}$$

where $w_k^{(K)} = n_k/n$ is the weight of the R^2 for the k^{th} cluster, and $n_k = |C_k|$ is the number of observations for the k^{th} cluster. From the plot of $(1 - R^{2^{(K)}})$ versus K, the elbow point is the optimal number of clusters, K^*.

To determine the optimal number of clusters K by looking for the elbow point can be subjective, especially when the elbow point is not obvious. Analogously with the adjusted R^2 for the linear regression model, we propose an adjusted R^2 to determine the optimal K for the K-regressions algorithm. We know that the R^2 for ordinary least square regression stands for the proportion of variation explained by the model. The R^2 is defined as

$$R^2 = 1 - SS_{res}/SS_{tot} = SS_{reg}/SS_{tot}, \tag{3.9}$$

where $SS_{tot} = \sum_i (y_i - \bar{y})^2$ is the total sum of squares, $SS_{reg} = \sum_i (\hat{y}_i - \hat{y})^2$ is the sum of squares of the regression, $SS_{res} = \sum_i (y_i - \hat{y}_i)^2$ is the sum of squares of the residuals, and $\hat{y} = \sum_i y_i/n$ is the sample mean of y. The R^2 in equation (3.9) can be rewritten as

$$R^2 = 1 - \text{var}_{res}/\text{var}_{tot}, \tag{3.10}$$

where $\text{var}_{res} = SS_{reg}/n$ and $\text{var}_{tot} = SS_{tot}/n$. The var_{res} and var_{tot} terms are both biased estimators of the residual variation and the population variation, respectively. The adjusted R^2 term adjusts these two variance estimators to be unbiased estimators, so that the adjusted R^2 is defined as

$$\bar{R}^2 = 1 - \frac{SS_{res}/df_e}{SS_{tot}/df_t}, \tag{3.11}$$

where $df_e = n - p - 1$ is the degree of freedom of the residuals, and $df_t = n - 1$ is the degree freedom of the population variation. The adjusted R^2 adjusts the R^2 in equation (3.9) so that it does not always increase.

The K-regressions algorithm fits K regressions on the whole data set, so that the total number of parameters is $K * p$. For each $K = 1, \ldots, K^{max}$, analogously with the idea of an adjusted R^2 for ordinary least square regression, we define the adjusted weighted R^2 for the K-regressions clustering as

$$\begin{aligned} \bar{R}^{2^{(K)}} &= R^{2^{(K)}} - (1 - R^{2^{(K)}}) \frac{K * p}{n - K * p - 1} \\ &\equiv \bar{R}^{2^{(K)}} - P^{(K)}, \end{aligned} \tag{3.12}$$

where $P^{(K)} = (1 - R^{2^{(K)}}) \frac{K*p}{n-K*p-1}$ is the penalty term.

The adjusted weighted R^2 in equation (3.12) penalizes the $R^{2^{(K)}}$ of equation (3.8) by the factor $P^{(K)}$ when the number of clusters increases. Since we fit K different linear regression models for the whole data set, $K*p$ is the number of parameters for the cluster-wise regression methodology.

From equation (3.12), $\bar{R}^{2^{(K)}}$ is always smaller than the $R^{2^{(K)}}$. The $\bar{R}^{2^{(K)}}$ increases only if the increase of K improves the $R^{2^{(K)}}$ more than the penalized term $P^{(K)}$. Usually when the number of clusters K increases, the $\bar{R}^{2^{(K)}}$ increases and reaches a maximum at a certain value of K, and decreases afterwards. The number of K that maximizes the $\bar{R}^{2^{(K)}}$ or minimizes the $1 - \bar{R}^{2^{(K)}}$ is the optimal number of clusters K^*. We compare the two methods of determining the optimal K by simulation results in the next section.

3.4 Simulation: Methodology

There are many ways to simulate interval-valued data sets where the response variable Y has a linear relationship with the predictor variables $\boldsymbol{X} = (X_1, ..., X_p)$ characterized by

$Y = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. We propose four simulation methods each of which has its advantages and disadvantages.

Simulation method I

A naive way is to randomly draw n samples from a multivariate normal distribution $N(\boldsymbol{\mu}, \Sigma)$ as the interval means or center points of the predictor variables, denoted as $\boldsymbol{X}^{(c)}_{n\times p}$. We then calculate the interval means of the response variable as $Y^{(c)} = \boldsymbol{X}^{(c)}_{n\times p}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_n)^T$ are independent and identically normal distributed (iid) random variables, i.e., $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, and where $\boldsymbol{\beta} = (\beta_0, ..., \beta_p)^T$ is given. Then, we simulate the interval ranges of each predictor variable $X^{(r)}_j$, $j = 1, ..., p$, and the response variable $Y^{(r)}$ independently. The values of the interval ranges can be drawn, e.g., from an exponential distribution, log-normal distribution, chi-square distribution or uniform distribution with positive support. The simulated X_j, $j = 1, ..., p$, is $X_j = [X^{(c)}_j - 0.5X^{(r)}_j, X^{(c)}_j - 0.5X^{(r)}_j]$, and $Y = [Y^{(c)} - 0.5Y^{(r)}, Y^{(c)} + 0.5Y^{(r)}]$.

There are two problems with this simulation method. First, the method makes the interval means of Y and the interval means of $\boldsymbol{X}$ follow a linear relationship, but we need that the interval variable Y and interval variables $\boldsymbol{X} = (X_1, \ldots, X_p)$ themselves follow a linear relationship. Second, from a data set obtained by this method, the interval ranges of Y would be independent of the interval ranges of $\boldsymbol{X}$, which is not true. Since $Y = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, given an observation $(\boldsymbol{x}_i, y_i)$, we have $y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$, $i = 1, ..., n$. The range of y_i is given by, for $i = 1, ..., n$,

$$\begin{aligned} y^{(r)}_i &= y_{ib} - y_{ia} \\ &= \max_{\boldsymbol{x}\in\mathbb{X}}(\boldsymbol{x}_i^T\boldsymbol{\beta} + \epsilon_i) - \min_{\boldsymbol{x}\in\mathbb{X}}(\boldsymbol{x}_i^T\boldsymbol{\beta} + \epsilon_i) \\ &= x^{(r)}_1|\beta_1| + ... + x^{(r)}_p|\beta_p| + \epsilon^{(r)}_i, \end{aligned} \tag{3.13}$$

for $i = 1, ..., n$, where $\mathbb{X} = \{\boldsymbol{x}_i = (x_i j) : x_{ija} \leq x_i j \leq x_{ijb}, j = 1, ..., p\}$. From equation (3.13), the ranges of Y should be positively correlated with the ranges of $\boldsymbol{X}$. Though the method I has these obvious problems, the advantage is that it is easy to implement and it guarantees the distributions within each interval of x_{ij} and y_i, $i = 1, ..., n$, $j = 1, ..., p$, are uniform distributions because we assume so. In practice, it is very unlikely that an interval data set with a linear relationship between response variable and predictor variables is obtained by the process of method I. We will not implement this method in our simulation study in section 3.5.

Simulation method II

Analogously with the simulation for classical data, we propose a second method to simulate symbolic intervals $\boldsymbol{X}$ and Y that satisfy $Y = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. Suppose the interval means of $\boldsymbol{X}$ follow a multivariate normal distribution $N(\boldsymbol{\mu}, \Sigma)$, and the interval ranges of X_j, $j = 1, ..., p$, follow an exponential distribution $\exp(\lambda_j)$. The exponential distributions are independent for $j \neq j'$, $j, j' = 1, ..., p$. Note that here the exponential distribution could be replaced by, e.g., a log-normal, chi-square, or uniform distribution with positive support. Then, the interval means, $X_j^{(c)}$, and interval ranges, $X_j^{(r)}$, for each predictor variable X_j, $j = 1, ..., p$, can be simulated given the parameters of the multivariate normal distribution and the exponential distributions. The simulated intervals of the predictor variables for a particular observation i is given by $x_{ij} = [x_{ij}^{(c)} - 0.5x_{ij}^{(r)}, x_{ij}^{(c)} + 0.5x_{ij}^{(r)}]$, $j = 1, ..., p$. The interval y_i consists of two intervals, $\boldsymbol{x}_i^T\boldsymbol{\beta}$ and ϵ_i, $i = 1, ..., n$. Since we already had the interval for $\boldsymbol{x}_i$, the interval $\boldsymbol{x}_i^T\boldsymbol{\beta}$

is given by, for $i = 1, ..., n$,

$$\begin{aligned} \boldsymbol{x}_i^T\boldsymbol{\beta} &= [a_i, b_i], \\ a_i &= \sum_{j:\beta_j>0} x_{ija}\beta_j + \sum_{j':\beta_{j'}<0} x_{ij'b}\beta_{j'}, \\ b_i &= \sum_{j:\beta_j>0} x_{ijb}\beta_j + \sum_{j':\beta_{j'}<0} x_{ij'a}\beta_{j'}. \end{aligned} \tag{3.14}$$

The equation (3.14) is the interval for y_i without the error term. To add the error term ϵ_i on y_i, suppose the interval means of ϵ_i follow a normal distribution, $\epsilon_i^{(c)} \overset{iid}{\sim} N(0, \sigma^2)$, and the interval ranges follow an exponential distribution, $\epsilon_i^{(r)} \overset{iid}{\sim} \exp(\lambda)$. We draw random samples from the two distributions. The error intervals are then given by $\epsilon_i = [\epsilon_{ia}, \epsilon_{ib}] = [\epsilon_i^{(c)} - 0.5\epsilon_i^{(r)}, \epsilon_i^{(c)} + 0.5\epsilon_i^{(r)}]$, $i = 1, ..., n$. The simulated intervals for the response variable are given by $y_i = [y_{ia}, y_{ib}] = [a_i + \epsilon_{ia}, b_i + \epsilon_{ib}]$, $i = 1, ..., n$, where a_i and b_i are as in equation (3.14).

This simulation method is like simulation for classical data, where we add an *iid* error term on the simulated $\boldsymbol{x}_i\boldsymbol{\beta}$ for $i = 1, ..., n$. There are drawbacks about this method. First, the range of the simulated y_i, $i = 1, ..., n$, can be derived as

$$\begin{aligned} y_i^{(r)} &= (b_i - a_i) + (\epsilon_{ib} - \epsilon_{ia}) \\ &= (\boldsymbol{x}_i\boldsymbol{\beta})^{(r)} + \epsilon_i^{(r)} \\ &\geq (\boldsymbol{x}_i\boldsymbol{\beta})^{(r)}, \end{aligned} \tag{3.15}$$

where $(\boldsymbol{x}_i\boldsymbol{\beta})^{(r)}$ stands for the range of the interval $\boldsymbol{x}_i\boldsymbol{\beta}$. From equation (3.15), the range of y_i is always not less than the range of $\boldsymbol{x}_i\boldsymbol{\beta}$, which is not true in practice. In addition, for a particular i, if we assume that the internal distributions of the intervals x_{ij}, $j = 1, ..., p$, and ϵ_i are uniform distributions, the obtained interval y_i is the sum of $p + 1$ uniform distributions. The internal distribution of y_i is generally not a uniform distribution anymore, which

violates the assumption of SVM method for the interval-valued data linear regression method.

Simulation method III

In practice, most of the interval data sets arise from aggregating classical data. From this perspective, we propose a third simulation method. The intervals of $\boldsymbol{X}$ are simulated as in method II where the interval means come from a multivariate normal distribution, and the interval ranges are from exponential distributions. The intervals of X_j, $j = 1, ..., p$, are given by $X_j = [X_j^{(c)} - 0.5X_j^{(r)}, X_j^{(c)} + 0.5X_j^{(r)}]$. The distributions within these intervals are assumed to be uniform. For a particular observation i, to obtain the interval y_i, we randomly draw m values from the uniform distribution $U(x_{ija}, x_{ijb})$ for each $j = 1, ..., p$, denoted by $x_{ij1}, ..., x_{ijm}$. The m is a predetermined number. Then the interval $y_i = [y_{ia}, y_{ib}]$ is determined by

$$\begin{aligned} y_{ia} &= \min_{l \in \{1,...,m\}} \{\beta_0 + \beta_1 x_{i1l} + ... + \beta_p x_{ipl} + \epsilon_{il}\}, \\ y_{ib} &= \max_{l \in \{1,...,m\}} \{\beta_0 + \beta_1 x_{i1l} + ... + \beta_p x_{ipl} + \epsilon_{il}\}, \end{aligned} \tag{3.16}$$

where $\epsilon_{il} \overset{iid}{\sim} N(0, \sigma^2)$ for $i = 1, ..., n$ and $l = 1, ..., m$.

This method is practically reasonable. For example, traffic on an particular intersection is recorded multiple times everyday; the minimum and maximum values are recorded as the traffic interval for a day. A more general case for this method is to assume the number m follows a certain distribution, say, an exponential distribution. For each observation i, the m's are the same for all the predictors X_j, $j = 1, ..., p$, but the m's are different for different observations. We have $m_i \overset{iid}{\sim} \exp(\lambda)$ for $i = 1, ..., n$. The interval y_i, $i = 1, ..., n$, is given by

$$\begin{aligned} y_{ia} &= \min_{l \in \{1,...,m_i\}} \{\beta_0 + \beta_1 x_{i1l} + ... + \beta_p x_{ipl} + \epsilon_{il}\}, \\ y_{ib} &= \max_{l \in \{1,...,m_i\}} \{\beta_0 + \beta_1 x_{i1l} + ... + \beta_p x_{ipl} + \epsilon_{il}\}. \end{aligned} \tag{3.17}$$

By allowing a random value for m, this simulation method fits more general scenarios. For instance, the daily price for a particular stock is an interval where the lower bound is the minimum price while the upper bound is the maximum price. The prices for the stock are recorded on a transaction base for every trading day, but the number of transactions on each day is not fixed. Instead, it is a random number that follows a certain distribution.

The problem for this simulation method is that it cannot guarantee the obtained intervals y_i, $i = 1, ..., n$, internally follow uniform distributions. The advantage is that it is close to how the interval data sets are collected in practice.

Simulation method IV

The fourth method tries to remedy the defect of the method III where the intervals y_i, $i = 1, ..., n$, are generally not uniform distributed internally. The intervals for X_j, $j = 1, ..., p$, are simulated the same way as in method III. Similarly as in method III, we assume the distribution within each interval of X_j is a uniform distribution. For each observation i, m values are randomly drawn from each interval x_{ij}, $j = 1, ..., p$, denoted by $x_{ij1}, ..., x_{ijm}$. The values of the response variable are calculated for $l = 1, ..., m$, as

$$y_{il} = \boldsymbol{x}_{il}^T \boldsymbol{\beta} + \epsilon_{il}, \tag{3.18}$$

where $\boldsymbol{x}_{il} = (1, x_{i1l}, ..., x_{ipl})^T$, and $\epsilon_{il} \overset{iid}{\sim} N(0, \sigma^2)$ for $i = 1, ..., n$ and $l = 1, ..., m$. So far, method IV is the same as method III, but we require the predetermined number m to be large. For observation i, the interval y_i is the interval from the first quartile to the third quartile of y_{il}, $l = 1, ..., m$. It can be verified that the interval y_i obtained by this way follows a uniform distribution for a relatively large m, say, $m \geq 3000$, see Xu and Billard (2014). The method can be easily extend to the scenario where m is a random variable.

Method IV solves the problem of method III where intervals y_i, $i = 1, ..., n$, are not

internally uniformly distributed. The disadvantage of method IV is that the number for m is not always large enough in practice to ensure that the y_i follows a uniform distribution. Furthermore, in practice when we aggregate classical data sets, we cannot simply cut the first quartile and the last quartile.

Each of the four methods has its advantages and disadvantages. In the following section of the simulation study, we will use the third method when m is relatively small. We use the fourth method if m is relatively large.

3.5 Simulation: Case Study

In this section, we conduct simulation studies to investigate the performance of the K-regressions algorithm. We try different data structures for the simulations study. The simulation methods of the interval-valued data follow the simulation methods II, II, and IV in section 3.4. Method I will not be implemented due to its unpractical process. We first compare the K-regressions clustering and the traditional K-means clustering methods, and investigate the convergence of the K-regressions algorithm. Then, we study the performance of the K-regressions algorithm for several different structures of data sets.

3.5.1 Comparison between the K-regressions Algorithm and K-means Algorithm

The K-means algorithm is designed for a spherical data structure. When each of the clusters in a data set is not spherical, the algorithm can fail. For example, if the variables are highly correlated within a cluster and the clusters are overlapped, it is difficult for the K-means algorithm to recover such clusters. In this section, we give two examples where the K-means algorithm fails to recover the true clusters while the K-regressions algorithm successds. The K-means clustering method for interval-valued data is based on the algorithms in Chavent

and Lechevallier (2002) and de Souza and de Carvalho (2004). We also study the convergence of the algorithm for these two examples.

Our first data set (I) is set to be composed of three clusters that follow the equations:

$$\begin{aligned} \text{cluster } (1): &\ y = 142 + 5x + \epsilon_1, \\ (2): &\ y = 53 - 3x + \epsilon_2, \\ (3): &\ y = -43 + 0.6x + \epsilon_3, \end{aligned} \tag{3.19}$$

respectively, where $\epsilon_1 \sim N(0, 15^2)$, $\epsilon_2 \sim N(0, 12^2)$, and $\epsilon_3 \sim N(0, 7^2)$. We apply the simulation method III and set $m = 25$ to simulate the data set (I) for each of its three regression models. The observations of these three regression models are simulated separately with 200 observations for each, and then the three data sets are stacked into one data set. Given $K = 3$, we implement the K-means clustering method based on each of the city-block distance (see equation (2.40)) and the Hausdorff distance (see equation (2.35)). For the K-regressions clustering method, we use the center distance (see equation (3.5)) for demonstration purposes.

Figure 3.1(a) shows the three true clusters with the three linear lines of equation (3.19), respectively. Figure 3.1(b) shows the clustering results based on the K-means algorithm when the city-block distance was used, while Figure 3.1(c) gives the K-means clustering results using Hausdorff distance of equation (3.19). From Figure 3.1 (b) and Figure 3.1 (c), we see that both these K-means algorithms cluster at the intersection areas between the three clusters in equation (3.19), which are clearly not the correct clusters.

We implement the K-regressions algorithm onto the same data set (I) given the number of clusters $K = 3$. After trying multiple initial partitions, the one with minimum SSR (see equation (3.4)) is said to be the convergence result for the K-regressions algorithm. We call an initial partition that converges to the minimum SSR as being a good initial partition.

Figure 3.1 (d), (e), and (f) show the clustering process of the K-regressions algorithm with a good initial partition. Figure 3.1 (d) shows the first (initialization) iteration of the K-regressions algorithm, while Figure 3.1 (c) shows the third iteration where the algorithm starts to converge to the true linear regression model in equation (3.19). Figure 3.1 (f) shows the tenth or the final iteration of the K-regressions algorithm where the algorithm converges to the three true linear regression model in equation (3.19). The three linear regression models obtained by the K-regressions algorithm are, respectively,

$$\begin{aligned} &(1) \quad y = 138.80 + 4.94x, \\ &(2) \quad y = 54.13 - 3.18x, \\ &(3) \quad y = -41.7 + 0.65x. \end{aligned} \tag{3.20}$$

The coefficients in equation (3.20) are estimated by the SVM method on the three clusters obtained by the K-regressions algorihtm. These coefficients are close to the true coefficients in equation (3.19). In addition, by comparing the true data set in Figure 3.1 (a) and the K-regression clustering results in Figure 3.1 (d), it is safe to say that the K-regression algorithm recovers the three true clusters for data set (I) in equation (3.19). A further investigation shows that all the misclassification observations are from the intersection areas between the three clusters.

Our second data set (II) is a two-dimensional data set that contains three clusters. We apply the simulation method IV in equation (3.4) and set $m = 3000$ and to simulate the data set. One hundred observations for each regression model are simulated and then all the observations are stacked into one data set. Then, the three linear regression models between

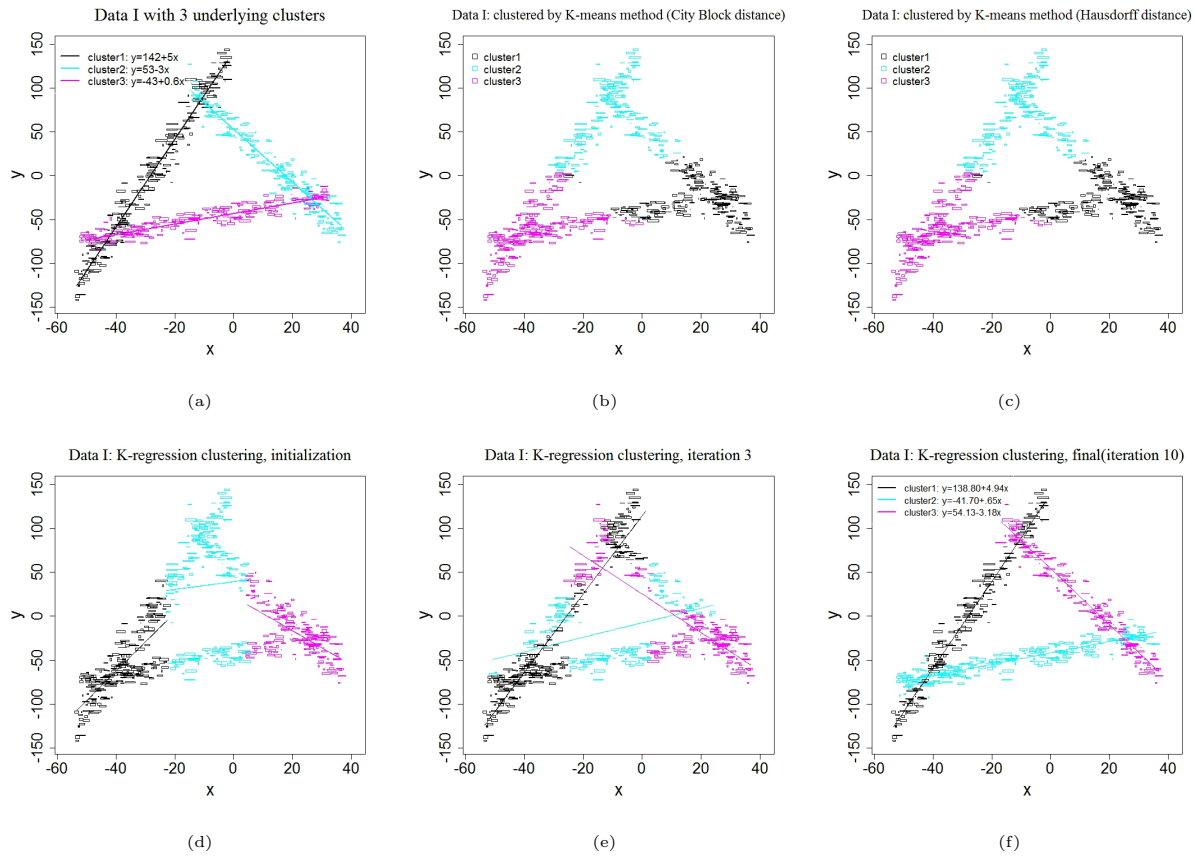


Figure 3.1: Comparison between clustering results of K-means algorithm and K-regressions algorithm for data set (I) of equation (3.19)

the two variables are as follows:

$$\begin{aligned} &(1) \quad y = 150.5 + 4.5x + \epsilon_1, \\ &(2) \quad y = 53 - 3x + \epsilon_2, \\ &(3) \quad y = -53 + 0.5x + \epsilon_3, \end{aligned} \tag{3.21}$$

where $\epsilon_1 \sim N(0, 15^2)$, $\epsilon_2 \sim N(0, 12^2)$, and $\epsilon_3 \sim N(0, 7^2)$.

The simulated data set (II) with total 300 observations and 3 clusters is visualized in

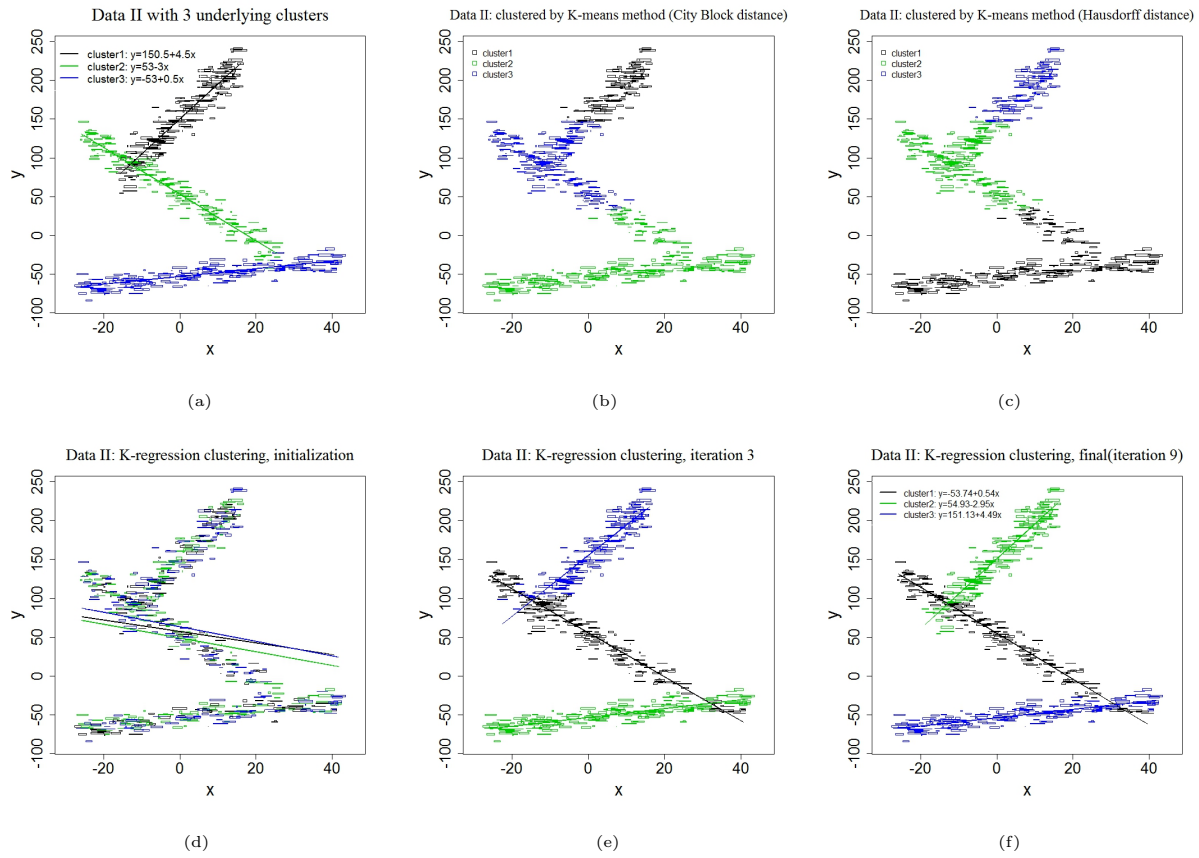


Figure 3.2: Comparison between clustering results of K-means algorithm and K-regressions algorithm for data set II of equation (3.21)

Figure 3.2(a) where the three true regression lines are also plotted. Given the number of clusters $K = 3$, we implement the K-means algorithm to cluster the data set (II). Figure 3.2 (b) and (c) give the clustering results by the K-means algorithm with the city-block distance of equation (2.40) and the Hausdorff distance of equation (2.35). We can see clearly from Figure 3.2 (b) and Figure 3.2 (c) that the K-means algorithms with both the city-block distance and the Hausdorff distance fail to recover the correct clusters .

The K-regression algorithm is applied to this data set (II) for the center distance given the number of clusters $K = 3$. Multiple initial partitions are tried and the one with the smallest

SSR is selected as the correct convergence for the K-regressions algorithm. Figure 3.2 (c), (d), and (e) show the convergence process of the K-regressions algorithm onto the data set (II) with the good initial partition. Figure 3.2 (a) is the plot of the three clusters for the first iteration (initialization). Figure 3.2 (b) shows the third iteration of the algorithm where the three clusters are already close to the three true clusters. The ninth (final) iteration is presented in Figure 3.2 (c) and shows clearly the converged three clusters by the algorithm. The estimated linear regression model by the SVM method for the three converged clusters are as follows:

$$\begin{aligned} &(1) \quad y = 151.13 + 4.49x, \\ &(2) \quad y = 54.93 - 2.95x, \\ &(3) \quad y = -53.74 + 0.54x. \end{aligned} \tag{3.22}$$

The estimated coefficients in equation (3.22) and the true coefficients in equation (3.21) are quite close. In addition, by comparing the plot of the original three linear regression models in Figure 3.2 (a) and the plot of the converged three clusters in Figure 3.2 (f), it is safe to say that the K-regression algorithm successfully recovered the true structure of the data set (II). Both the data set (I) and (II) in equation (3.19) and equation (3.21) are not spherical and the K-means algorithm failed to recover the true structure whereas our method did succeed.

In these two simulated examples using data set (I) and (II), we assume that the true number of clusters is known. However, in practice we usually do not have precise information about the number of clusters. To decide the optimal number of clusters, we use the method proposed in section 3.3. We calculate the weighted R-squared, $R^{2(K)}$, for $K = 1, ..., 8$, from equation (3.8). The elbow plot is the plot of $R^{2(K)}$ versus the number of clusters K. Figure 3.3(a) and (b) show the elbow plots for data set (I) from equation (3.19) and data set (II) from equation (3.21), respectively. For both data sets, the elbow plots show that the optimal

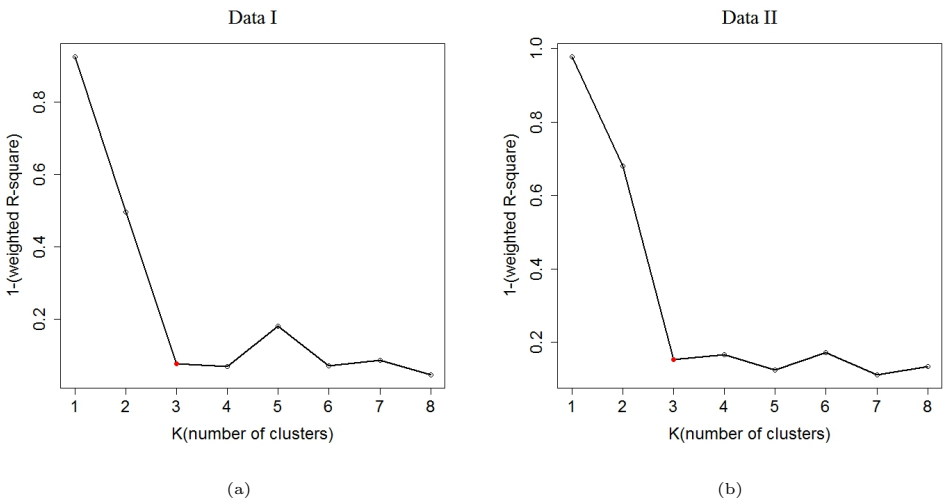


Figure 3.3: Determining the number of clusters K by an elbow plot

number of clusters is $K = 3$. That is, the elbow plot correctly determines the true number of clusters for both data sets.

3.5.2 Performance of the K-regressions Algorithm

In this section, we simulate several data sets with different structures to investigate the performance of the K-regressions algorithm. In particular, we consider the following three data sets. Data I and II are two-dimensional interval-valued data set with three clusters, while Data III is a two-dimensional interval-valued data set with four clusters. Table 3.1 provides the parameter setup for the three data sets. In Table 3.1, n is the sample size for each of the clusters; β_0 and β_1 are the coefficient of the linear relation for each cluster. The values μ_x and σ_x are the two parameters of the normal distribution $N(\mu_x, \sigma_x)$ from which the interval center points of the predictor variable X are drawn. The value λ_x is the parameter of the exponential distribution $\exp(\lambda_x)$ from which the interval ranges of X are

Table 3.1: Parameter setup for the Data I, II, and III

	Data I			Data II			Data III			
Cluster	1	2	3	1	2	3	1	2	3	4
n	100	100	100	100	100	100	60	60	60	60
β_0	1.0	45.0	45.0	142.0	33.0	-73.0	2.0	1.0	3.0	1.0
β_1	1.3	1.8	-2.5	5.0	-3.0	0.6	0.8	2.3	-1.8	4.3
μ_x	4.0	0.0	8.0	-28.0	12.0	-10.0	4.0	3.0	4.0	3.0
σ_x	12.0	9.6	9.0	10.0	17.0	20.0	4.0	3.0	4.0	3.0
λ_x	1.5	1.3	1.2	1.0	0.9	1.0	10.0	12.0	10.0	12.0
μ_e	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
σ_e	5.0	4.0	3.0	6.0	9.0	8.0	1.0	2.0	1.0	4.0
λ_e	2.5	2.0	2.0							

drawn. The error terms ϵ_i of a linear regression equation $y_i = \boldsymbol{x}_i^T\boldsymbol{\beta} + \epsilon_i$ are drawn from a normal distribution $N(0, \sigma_e)$ where the values of the parameter σ_e are shown in the row "σ_e" in Table 3.1. If a data set is simulated by the simulation method II, the interval ranges of the error terms ϵ_i are drawn from an exponential distribution $\exp(\lambda_e)$ where the values of parameter λ_e are shown in the row "λ_e" in Table 3.1. Data I is simulated by the simulation method II in section 3.4, while Data II and III are simulated by the simulation method III and IV, respectively.

We use the Data I as example to demonstrate the parameter setup in Table 3.1, the parameter set up for Data II and III are analogously with that for Data I. Data I has three clusters whose parameters are shown in the three columns under "Data I" in Table 3.1. Each of the clusters is composed of 100 observations that are indicated in the row "n". For cluster 1 of Data I, all parameters that are needed to simulate the 100 observations are in the column "1" under the "Data I" tab. The coefficients of the linear regression model of the cluster 1 for Data I is $\beta_0 = 1.0$ and $\beta_1 = 1.3$ with an equation $y = 1.0 + 1.3x$. According to the simulation method II, we generate 100 interval center points of the variable X from a normal distribution $N(\mu_x = 4.0, \sigma_x = 12.0)$, denoted as $\boldsymbol{x}^{(c)} = (x_1^{(c)}, \ldots, x_{100}^{(c)})^T$. The 100 interval ranges of the variable X are drawn from an exponential distribution $\exp(\lambda_x = 1.5)$,

denoted by $\boldsymbol{x}^r = (x_1^{(r)}, \ldots, x_{100}^{(r)})$. Then, a 100-observation sample of the interval-valued variable X is composed by

$$\begin{aligned} x_i &= [x_{ia}, x_{ib}] \\ &= [x_i^{(c)} - 0.5x_i^{(r)}, x_i^{(c)} - 0.5x_z^{(r)}], \end{aligned} \tag{3.23}$$

for $i = 1, \ldots, 100$. The 100 interval center points of the error term ϵ are generated from a normal distribution $N(\mu_e = 0, \sigma_e = 5.0)$, denoted with $\epsilon^{(c)} = (\epsilon_1^{(c)}, \ldots, \epsilon_{100}^{(c)})^T$. The 100 interval ranges of ϵ are drawn from an exponential distribution $\exp(\lambda_e) = 2.5$, denoted with $\epsilon^{(r)} = (\epsilon_1^{(r)}, \ldots, \epsilon_{100}^{(r)})^T$. The 100 interval-valued error terms are obtained by

$$\begin{aligned} \epsilon_i &= [\epsilon_{ia}, \epsilon_{ib}] \\ &= [\epsilon^{(c)} - 0.5\epsilon^{(r)}, \epsilon^{(c)} - 0.5\epsilon^{(r)}], \end{aligned} \tag{3.24}$$

for $i = 1, \ldots, 100$. Eventually, the interval-valued response variable Y is generated by

$$\begin{aligned} y_i &= [y_{ia}, y_{ib}] \\ &= [1.0 + 1.3x_{ia} + \epsilon_{ia}, 1.0 + 1.3x_{ib} + \epsilon_{ib}], \end{aligned} \tag{3.25}$$

for $i = 1, \ldots, 100$. The cluster 2 and 3 of Data I are simulated by a similar way as the cluster 1. The linear regression equation of cluster 2 for Data I is $y = 45 + 1.8x$. The 100 interval center points of X are drawn from a normal distribution $N(\mu_x = 0, 9.6)$, while the 100 interval range of X are generated from an exponential distribution $\exp(1.3)$. For the interval-valued error term ϵ, we have $\epsilon_i^{(c)} \overset{iid}{\sim} N(0, 4)$ and $\epsilon_i^{(r)} \sim \exp(2.0)$, $i = 1, \ldots, 100$. For cluster 3 of Data I, the linear regression equation is $y = 45 - 2.5x$. The interval center points of X follow a normal distribution $x_i^{(c)} \overset{iid}{\sim} N(8, 9)$, $i = 1, \ldots, 100$. The interval ranges of X follows an exponential distribution, $x_i^{(r)} \sim \exp(1.2)$. For the interval-valued error term ϵ, we have $\epsilon_i^{(c)} \overset{iid}{\sim} N(0, 3)$ and $\epsilon_i^{(r)} \sim \exp(2)$, $i = 1, \ldots, 100$. After simulating all the three clusters

for Data I, we stack the three 100-observation sample to obtain the Data I. Data I has the following structure:

$$\begin{aligned} &(1)\ \ y = 1.0 + 1.3x, \\ &(2)\ \ y = 45 + 1.8x, \\ &(3)\ \ y = 45 - 2.5x. \end{aligned} \tag{3.26}$$

The parameter setups for Data II and III in Table 3.1 are done analogously with Data I. Note that Data II is simulated by simulation method III so that the error terms are classical values. The error terms for Data III obtained by the simulation method IV are also classical values. The values of the parameter σ_e are for the normal distribution of the classical values of ϵ. For example, the error terms of cluster 2 of Data II follows a normal distribution, $\epsilon \sim N(0, 9)$, while the error term of cluster 3 of Data III follows a normal distribution, $\epsilon \sim N(0, 1)$. The true linear regression equations for the three clusters of Data II are as follows:

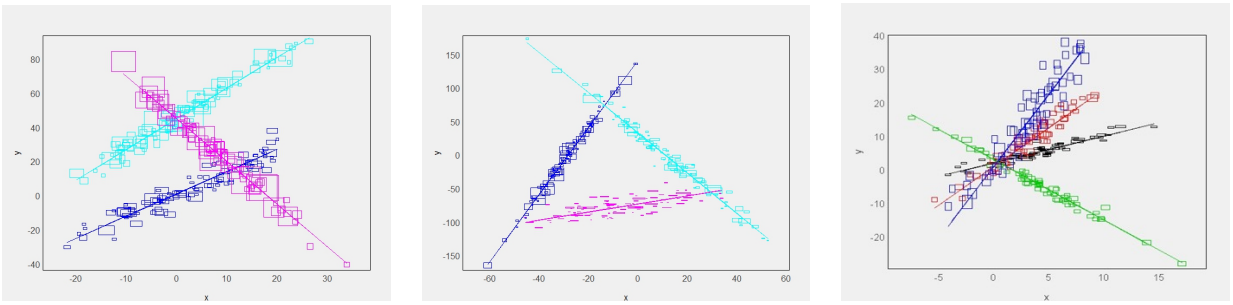
$$\begin{aligned} &(1)\ \ y = 142 + 5x, \\ &(2)\ \ y = 33 - 3x, \\ &(3)\ \ y = -73 + 0.6x. \end{aligned} \tag{3.27}$$

The true linear regression equations for the four clusters of Data III are as follows:

$$\begin{aligned} &(1)\ \ y = 2.0 + 0.8x, \\ &(2)\ \ y = 1.0 + 2.3x, \\ &(3)\ \ y = 3.0 - 1.8x, \\ &(4)\ \ y = 1.0 + 4.3x. \end{aligned} \tag{3.28}$$

Figure 3.4 shows the data structures for the three data sets, Data I, II, and III. From

Figure 3.4 (a), (b), and (c), we can observe the structure of Data I, II, and III, respectively. The regression lines in each plot are the recovered linear lines obtained by the K-regressions algorithm. We can see that different clusters overlap with each other for all three data sets. Especially, for Data III, a large proportion of the four clusters is overlapping. In addition, for a particular data set, each cluster is clustering around a linear regression line.



(a) (b) (c)

Figure 3.4: Data structure for the Data I (a), II (b), and III (c)

For the particular data structure of Data I, we generate a random sample that follows the Data I structure described in Table 3.1. Then, given the correct number of clusters $K = 3$ for Data I, we use the K-regressions algorithm to recover the data structure. We try a number of random initial partitions and repeat the K-regression algorithm on this random sample of Data I. Based on these different initial partitions, the clustering result with smallest sum of squared residuals of equation(3.4) is set to be the correct convergence for this random sample of Data I. For a random sample of Data I, we tried 50 different random initial partitions to recover its structure. This whole process is one replication for Data I and we implement 100 such replications for the structure of Data I to investigate the overall performance of the K-regression algorithm on Data I. The data set for each replication is a different random sample that follows the structure of Data I. Table 3.2 gives the clustering results of K-regression algorithm on 100 simulated random samples of Data I.

Table 3.2: K-regressions clustering results for Data I (number of replications=100)

	Parameters	True Values	center		city-block		Hausdorff	
			mean	std	mean	std	mean	std
Cluster 1	β_0	1.00	0.92	0.55	0.84	0.55	0.91	0.64
	β_1	1.30	1.31	0.04	1.29	0.05	1.29	0.05
Cluster 2	β_0	45.00	45.02	0.46	45.02	0.45	44.90	0.46
	β_1	1.80	1.80	0.04	1.80	0.04	1.80	0.04
Cluster 3	β_0	45.00	44.86	0.50	45.13	0.53	45.12	0.50
	β_1	-2.50	-2.49	0.04	-2.50	0.04	-2.51	0.04
n^* out of 50	-	-	28.48	8.77	24.76	12.25	25.87	12.15
SSR	-	-	890.26	39.18	2181.25	86.35	1458.43	50.66

In Table 3.2 the column "True Values" are the true coefficients for the three clusters of Data I. For example, the true values of the coefficients of the cluster 1 are $\beta_0 = 1.0$ and $\beta_1 = 1.3$, which indicates that the true linear relationship between the two interval-valued variables in the cluster 1 of Data I is $y = 1.0 + 1.3x$. Note that these true values correspond to the linear regression equations in equation (3.26). The columns "center", "city-block", and "Hausdorff" are the K-regressions algorithm clustering results using the center distance, the city-block distance, and the Hausdorff distance. For each replication, given the number of clusters $K = 3$, we apply the K-regressions algorithm to recover the true clusters of a particular random sample of Data I. We can obtain the estimated coefficients for each of the three clusters. After 100 replications, the mean and standard deviation are calculated for the coefficients of each clusters. The column "mean" under the tab "center" in Table 3.2 is the mean of the estimated coefficients out of 100 replications by applying the K-regressions algorithm using center distance. The column "std" is the standard deviation of the estimated coefficient out of the 100 replications. For example, the means estimated coefficients of β_0 and β_1 for the cluster 1 are 0.92 and 1.31, respectively. The standard deviation of the estimated β_0 and β_1 for the cluster 1 are 0.55 and 0.04, respectively. Small standard deviations of the coefficients indicate stable clustering results.

For each replication, we tried 50 different initial partitions when applying the K-regressions algorithm to a particular random sample of Data I. The number of good initial partitions out of 50, n^*, gives an idea about how difficult it is for the algorithm to converge to the correct cluster by a random initial partition. Out of the 100 replications, we can calculate the mean and standard deviation of n^*, which is shown in the row "n^* out of 50" in Table 3.2 for the three distances. The SSR (see equation (3.4)) is also calculated for each replication. The mean and standard deviation of the SSR out of the 100 replications are presented in the row "SSR".

For Data I, we compare the true values of the coefficients and the mean estimated coefficients in Table 3.2. The true coefficients and the mean of the estimated coefficients are close relative to their scales for all the three clusters and the three distances. For example, the mean estimated coefficient of β_1 for cluster 1 is 1.31 while the true value of β_1 is 1.3. The difference is 0.01 that is small relative to the value of 1.3. The standard deviations of the estimated coefficients are all small relative to the coefficient scales, which indicates the clustering results are stable. For example, the standard deviation of the 100 estimated β_1 for the cluster 2 is 0.04 that is relatively small given the mean estimated β_1 is 1.8.

Table 3.3: K-regressions clustering results for Data II (number of replications=100)

	Parameters	True Values	center		city-block		Hausdorff	
			mean	std	mean	std	mean	std
Cluster 1	β_0	142.00	141.30	2.09	140.77	1.89	141.75	2.01
	β_1	5.00	4.97	0.07	4.95	0.07	4.99	0.07
Cluster 2	β_0	33.00	33.06	1.25	33.39	1.30	33.42	1.28
	β_1	-3.00	-2.99	0.06	-3.00	0.06	-2.99	0.06
Cluster 3	β_0	-73.00	-72.93	0.98	-72.83	1.11	-72.64	1.14
	β_1	0.60	0.60	0.05	0.60	0.05	0.60	0.06
n^* out of 50	-	-	26.55	14.51	33.42	15.70	34.51	14.74
SSR	-	-	1757.41	85.17	4127.63	177.30	2689.33	90.65

Table 3.3 presents the clustering results for the Data II with 100 replications. Table 3.3 can be interpreted in a similar way as Table 3.2 for Data I. Given the number of clusters

$K = 3$, for all the three distances, the differences between the true coefficients and the mean estimated regression coefficients are all small relative to the coefficient scales. Small standard deviations for all the estimated coefficients imply a stable clustering results.

Table 3.4: K-regressions clustering results for Data III (number of replications=100)

	Parameters	True Values	center		city-block		Hausdorff	
			mean	std	mean	std	mean	std
Cluster 1	β_0	2.00	2.06	0.38	3.55	1.76	3.86	2.98
	β_1	0.80	0.81	0.05	0.68	0.17	0.73	0.18
Cluster 2	β_0	1.00	1.32	1.31	3.06	2.74	5.20	4.04
	β_1	2.30	2.36	0.22	2.28	0.50	1.97	0.73
Cluster 3	β_0	3.00	2.90	0.32	3.12	0.34	3.02	0.39
	β_1	-1.80	-1.78	0.04	-1.81	0.05	-1.80	0.05
Cluster 4	β_0	1.00	2.13	1.87	4.29	2.67	4.24	2.82
	β_1	4.30	4.27	0.35	4.04	0.46	4.05	0.48
n^* out of 200	-	-	43.95	45.77	14.73	28.70	24.77	33.35
SSR	-	-	296.25	20.52	777.90	44.84	521.13	27.75

The clustering results for the Data III are presented in the Table 3.4. The interpretation of Table 3.4 for Data III follows in a similar manner as for the Table 3.2 for Data I and Table 3.3 for Data II. Note that for Data III, a large proportion of the four clusters is overlapped, which makes it more difficult to converge to the correct clusters for the K-regressions algorithm. For each replication, we tried 200 different initial partitions. The differences between the true coefficients and the mean estimated coefficients are small relative to the scales of the coefficients. The standard deviations of the coefficients are small for all the estimated coefficients and all the three distances. However, the intercept estimates for clusters 2, 3, and 4 are not as accurate as it is for the cluster 1. This is not surprising given that cluster 1 is more separated from the other three clusters.

We have explored the performance of the K-regressions clustering algorithm for a given correct number of clusters, but usually we do not have information about the optimal number of clusters. Now, we use the same three data structures, Data I, Data II, and Data III presented in Table 3.1 to investigate the performance of determining the optimal number

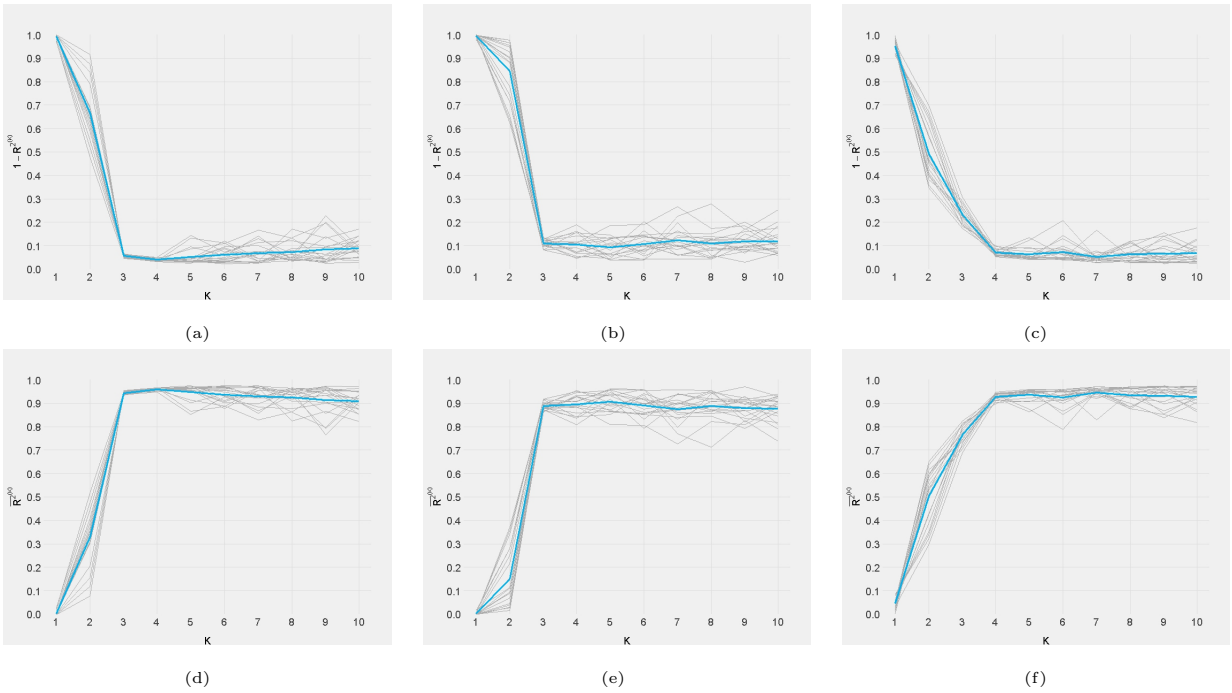


Figure 3.5: Elbow plots by weighted R^2 and adjusted R^2 for Data I (a) and (b), Data II (c) and (d), and Data III (e) and (f)

of clusters by the elbow method, and the adjusted R^2. For a particular data structure, we generate a random sample and implement the K-regressions algorithm for $K = 1, \ldots, 10$. For each of $K = 1, \ldots, 10$, we try a number of different initial partitions and select the results with smallest SSR as the correct clustering results. The $R^{2(K)}$ from equation (3.8) and $\bar{R}^{2(K)}$ from equation (3.12) are calculated for each of $K = 1, \ldots, 10$. The elbow plot is plotted as K versus $1 - R^{2(K)}$. We also plot the K versus $\bar{R}^{2(K)}$ where the maximum $\bar{R}^{2(K)}$ determines the optimal number of clusters. This whole process is for one replication, and we implement total 20 replication to test the performance of elbow method and the adjusted R^2.

Figure 3.5 shows the elbow plots and the plots of $\bar{R}^{2(K)}$. Figure 3.5 (a), (b), and (c) are the elbow plots for Data I, II, and III, respectively, where the grey lines are the elbow

plots for the 20 replications, the blue line is the average $R^{2^{(K)}}$ over 20 replications. We can see that the elbow plots identify the correct optimal number of clusters for all the three data sets, $K = 3$ for Data I and II, $K = 4$ for Data III. It is relatively difficult to determine the optimal number of clusters for Data III due to the overlapping of clusters, but the elbow plots correctly determined the number of clusters for all of the 20 replications nevertheless.

The Figure 3.5 (d), (e), and (f) show the plots of $\bar{R}^{2^{(K)}}$ from equation (3.8) for Data I, II, and III, respectively. For the plots of $\bar{R}^{2^{(K)}}$, we look for the largest value of $\bar{R}^{2^{(K)}}$ which corresponds to the optimal number of clusters. For each of the three data structures, the optimal number of clusters determined by the largest $\bar{R}^{2^{(K)}}$ is mostly larger than the true number of clusters for the 20 replications.

Generally, the elbow method is a stable and reliable method to determine the optimal number of clusters. There could be cases where the $R^{2^{(K)}}$ decreases gradually and consistently so that an elbow point is hard to find. Usually such scenarios indicate that there does not exist an optimal number of clusters to well separate the data and subjective judgment needs to be involved for a decision. Fixing a reasonable cutoff for the $R^{2^{(K)}}$ is a realistic option in practice. The $\bar{R}^{2^{(K)}}$, adjusted R^2, usually overestimates the optimal number of clusters and so is not a good method to determine the optimal number of clusters.

3.5.3 Appendix

R code for the K-regressions clustering

```
# ------------------------------------------------------------
#  Using SVM method to recover the relationship between x and y
# ------------------------------------------------------------

# ------------------------------------------------------------
#                        variance of interval data
# The function cov.int calculate the variance of a interval-valued
# variable, or covariance between two interval-valued variables

cov.int<-function(x,y=NULL)
{
        m<-nrow(x)
        if (is.null(y))
                {cov<-sum(x[,2]^2+x[,2]*x[,1]+x[,1]^2)/3/m-(sum(x[,1]+x[,2]))
                   ^2/4/m^2}

        else
        {       xbar<-sum(x[,1]+x[,2])/2/m
                ybar<-sum(y[,1]+y[,2])/2/m
                cov<-sum(2*(x[,1]-xbar)*(y[,1]-ybar)+(x[,1]-xbar)*(y[,2]-ybar)
                   +
                                (x[,2]-xbar)*(y[,1]-ybar)+2*(x[,2]-xbar)*(y
                                   [,2]-ybar))/6/m
        }
        return(cov)
}
# ------------------------------------------------------------
# The function corr.int calculate the correlation between two interval-valued
# variables
corr.int<-function(x,y)
{return(cov.interval(x,y)/sqrt(cov.interval(x))/sqrt(cov.interval(y)))}

# ------------------------------------------------------------
# The function calculate the variance covariance matrix of a
# multi-dimensional interval-valued data
varcov.interval<-function(x)
{
        #nobs<-nrow(x)
        nvar<-ncol(x)/2
        cov<-matrix(0,nvar,nvar)
```

```
        for(i in 1:(nvar-1))
        {       for(j in (i+1):nvar)
                {
                        cov[i,j]<-cov.int(x[,(2*i-1):(2*i)],x[,(2*j-1):(2*j)])
                }
        }
        for(i in 1:nvar) cov[i,i]<-cov.int(x[,(2*i-1):(2*i)])
        cov[lower.tri(cov)]<-t(cov)[lower.tri(cov)]
        return(cov)
}

# ---------------------------------------------------------
# The function lm.int calculate the coefficient estimate
# of linear regression between two interval-valued variables
# using the SVM method

lm.int<-function(y,x)
{  beta1<-cov.int(y,x)/cov.int(x)
   beta0<-mean.int(y)-beta1*mean.int(x)
   res<-cbind(beta0,beta1)
   colnames(res)<-c("intercept",deparse(substitute(x)))
   return(res)
}


# ---------------------------------------------------------
#       plot of x and y, x and y are both interval data
# The function plot.int draw a plot of variable x versus y.
# Each observation of the plot is a rectangle to reflect the
# interval-valued variables
library(graphics)
plot.int<-function(x,y,xlab=deparse(substitute(x)) ,ylab=deparse(substitute(y)
   ),
                main=NULL,sub=NULL,asp=NULL, density = NULL, angle = 45, col
                   = NA,
                border = NULL, lty = par("lty"), lwd = par("lwd"),...)
{       rangex<-max(x)-min(x)
        rangey<-max(y)-min(y)
        length<-nrow(x)
        plot(c(min(x)-.03*rangex,max(x)+.03*rangex), c(min(y)-.03*rangex,max(y
           )+.03*rangex),
             type="n",xlab=xlab,ylab=ylab,main=main,sub=sub,asp=asp,...)
        for (i in 1:length)
```

```
        {rect(x[i,1], y[i,1], x[i,2], y[i,2],density = NULL, angle = 45, col =
            NA,
                 border = NULL, lty = par("lty"), lwd = par("lwd"),...)}
}

# Function add.rect add more observations (rectangles) onto a plot
# that is plotted by the function plot.int.
add.rect<-function(x,y,col=NA,border=1,...)
{       rangex<-max(x)-min(x)
        rangey<-max(y)-min(y)
        length<-nrow(x)
        for (i in 1:length)
        {rect(x[i,1], y[i,1], x[i,2], y[i,2],col=col,border=border,...)}
}

# ---------------------------------------------------------
# The function hist.int plot the histogram of an interval-valued
# data set given the number of bin, num.bin. The default value
# of num.bin is 10.
hist.int<-function(x, num.bin=10)

{
  min.val <- min(x)
  max.val <- max(x)

  n <- nrow(x)
  bin.width <- (max.val-min.val)/num.bin
  freq <- numeric(num.bin)

  for (i in 1:n)
  {
    xbin1 <- (x[i,1] - min.val)/bin.width
    xbin2 <- (x[i,2] - min.val)/bin.width
    range <- x[i,2]-x[i,1]

    n1 <- ceiling(xbin1)
    n2 <- ceiling(xbin2)
    if (n2>num.bin) n2<-num.bin
    dec1 <- xbin1- n1+1
    dec2 <- n2-xbin2

    freq[n1:n2] <- freq[n1:n2]+1*bin.width/range
    freq[n1] <- freq[n1]-dec1*bin.width/range
    freq[n2] <- freq[n2]-dec2*bin.width/range
```

```
  }

  bins1 <- min.val+(0:(num.bin-1))*bin.width
  bins2 <- bins1+bin.width

  plot.int(cbind(bins1,bins2),cbind(0,freq),xlab=deparse(substitute(x)) ,ylab=
      "Frequency")

}

#   ------------------------------------------------------------------
#   Simulation methodology
#        The function simulate.int1, simulate.int2, and simulate.int3
#   correspond to the simulation method II, III, IV, respectively,
#   in chapter 3.
#   ------------------------------------------------------------------

#   ------------------------------------------------------------------
#   method II
#   ------------------------------------------------------------------
#   add a interval error on the interval of y which is calculated by linearly
    combining
#   the intervals of x's. The lower bound of y is the min of X*beta, while the
     upper
#   bound is the max of X*beta.

library(MASS)
simulate.int1 <- function(n,beta=c(1,1), x.mu=5,x.sigma=1.5,x.rate=15,e.mu=0,e
    .sigma=1.5,e.rate=1.5)
{
  p <- length(x.mu)

  if(p==1)  x.mean <- as.matrix(rnorm(n, x.mu, x.sigma),n,1)
  else x.mean <- mvrnorm(n, x.mu, diag(x.sigma))

  x<-NULL
  for (i in 1:p)
  {
    x.r <- rexp(n,x.rate[i])
    xa <- x.mean[,i]-.5*x.r
    xb <- x.mean[,i]+.5+x.r
    x <- cbind(x,xa,xb)
  }
```

```
  beta1<-beta[-1]

  #positive and negative positions
  pos<-which(beta1>0)
  neg<-which(beta1<0)
  if(length(pos)!=0)
  {
    a.pos<-as.matrix(x[,(2*pos-1)],n,length(pos))%*%beta1[pos]
    b.pos<-as.matrix(x[,(2*pos)],n,length(pos))%*%beta1[pos]
  }
  else
  {
    a.pos<-0
    b.pos<-0
  }

  if(length(neg)!=0)
  {
    a.neg<-as.matrix(x[,(2*neg)],n,length(neg))%*%beta1[neg]
    b.neg<-as.matrix(x[,(2*neg-1)],n,length(neg))%*%beta1[neg]
  }
  else
  {
    a.neg<-0
    b.neg<-0
  }

  ya<-beta[1]+a.pos+a.neg
  yb<-beta[1]+b.pos+b.neg

  e.mean <- rnorm(n,e.mu,e.sigma)
  e.r <- rexp(n,e.rate)
  e <- cbind(e.mean-.5*e.r,e.mean+.5*e.r)

  ya <- ya+e[,1]
  yb <- yb+e[,2]
  y <- cbind(ya,yb)

  data <- cbind(y,x)

  return(data)
}

#   ------------------------------------------------------------------
#   method III
```

```
#   ------------------------------------------------------------------
#   Randomly draw n.int points within each interval of xs, then calculate the
    ys that is
#   the linear combination of xs add a error term that follow a normal
    distribution.
#   The min of the ys would be the lower bound and max is the upper bound.

simulate.int2 <- function(n=10,beta=c(1,1),x.mu=0,x.sigma=10,x.rate=2,n.int
    =50,e.mu=0,e.sigma=1)
{
  p <- length(x.mu)

  if(p==1)  x.mean <- as.matrix(rnorm(n, x.mu, x.sigma),n,1)
  else x.mean <- mvrnorm(n, x.mu, diag(x.sigma))

  x<-NULL
  for (i in 1:p)
  {
    x.r <- rexp(n,x.rate[i])
    xa <- x.mean[,i]-.5*x.r
    xb <- x.mean[,i]+.5+x.r
    x <- cbind(x,xa,xb)
  }


  ya<-yb<-NULL

  for (i in 1:n)
  { x.int<-matrix(0,p,n.int)
    for(j in 1:p)
    {
      x.int[j,] <- runif(n.int,x[i,2*j-1],x[i,2*j])
    }
    y.int <- beta[1]+beta[-1]%*%x.int+rnorm(1,e.mu,e.sigma)
    ya.temp<-min(y.int)
    yb.temp<-max(y.int)

    ya<-c(ya,ya.temp)
    yb<-c(yb,yb.temp)
  }

  y <- cbind(ya,yb)

  data <- cbind(y,x)
  return(data)
```

```
}

#   ------------------------------------------------------------------
#   method IV
#   ------------------------------------------------------------------
#   Randomly draw n.int number for each interval of xs. The ys corresponding
    to these xs
#   are calculated as the linear combination of xs add a error term that
    follows a normal
#   distribution. Then the first quartile to the third quartile of these ys is
     the interval
#   y. The n.int need to be large.

simulate.int3 <- function(n=10,beta=c(1,1),x.mu=0,x.sigma=10,x.rate=2,n.int
   =1000,e.mu=0,e.sigma=1)
{
  x.mean <- rnorm(n, x.mu, x.sigma)
  x.r <- rexp(n,x.rate)
  xa <- x.mean-.5*x.r
  xb <- x.mean+.5+x.r
  x <- cbind(xa,xb)


  p <- length(x.mu)

  y<-NULL

  for (i in 1:n)
  {
    x.int <- runif(n.int,x[i,1],x[i,2])
    y.int <- beta[1]+beta[2]*x.int+rnorm(1,e.mu,e.sigma)
    y.temp<-quantile(y.int,c(.25,.75))

    y<-rbind(y,y.temp)
  }

  colnames(y) <- c("ya","yb")

  data <- cbind(y,x)
  #plot.int(x,y)
  return(data)

}
```

```
#   ------------------------------------------------------------------
#   The function K.regressions implement the K-regressions
#         clustering algorithm given the optimal number of cluster
#    K. The interval distance can be center, city-block, or
#    Hausdorff distance.
#   ------------------------------------------------------------------
library(caret)
K.regressions<-function(data,K,seed,distance="center",max.iter=100,list.group=
   FALSE)
{
  if (!missing(seed))
    set.seed(seed)

  p<-ncol(data)/2-1
  n<-nrow(data)

  data.fold<-createFolds(1:n, k = K)

  # initialization
  m<-matrix(0,K,2)
  group<-rep(0,n)
  for (k in 1:K)
  {
    group[data.fold[[k]]] <- k
    m[k,]<-lm.int(data[data.fold[[k]],1:2],data[data.fold[[k]],3:4])
  }

  cond<-1
  i<-0

  while(cond && i<=max.iter)
  {  i<-i+1
     #print(i)

     residual<-matrix(0,n,K)

     if (distance == "center")
     {
       for (k in 1:K)
       {
         residual[,k]<-abs(apply(data[,1:2],1,mean)-m[k,1]-m[k,2]*apply(data
            [,3:4],1,mean))
       }
     }
```

```
    if(distance == "city-block")
    {
      for (k in 1:K)
      {
        yhat<-m[k,1]+m[k,2]*data[,3:4]
        residual[,k]<-abs(data[,1]-apply(yhat,1,min))+abs(data[,2]-apply(yhat
            ,1,max))
      }
    }

    if (distance == "Hausdorff")
    {
      for (k in 1:K)
      {
        yhat<-m[k,1]+m[k,2]*data[,3:4]
        residual[,k]<-apply(cbind(abs(data[,1]-apply(yhat,1,min)),abs(data
            [,2]-apply(yhat,1,max))),1,max)
      }
    }

    regroup<-apply(residual,1,f<-function(x){return(which(x==min(x))[1])})

    if (!all(table(regroup)>p)||length(table(regroup))<K)
      stop("Number␣of␣observations␣is␣smaller␣than␣the␣number␣of␣parameters␣
          for␣one
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣or␣more␣clusters!")

    sum.residual<-0
    m<-matrix(0,K,2)
    for (k in 1:K)
    {
      m[k,]<-lm.int(data[regroup==k,1:2],data[regroup==k,3:4])
      sum.residual<-sum.residual+sum(residual[regroup==k,k])
    }

    cond<-!all(regroup==group)
    #print(sum(ifelse(group==regroup,1,0)))
    group<-regroup
  }

  #m<-m[order(m[,1]),]

  models<-matrix(t(m),1,2*K)

  if (list.group==T)
```

```
    {return(list(sum.residual = sum.residual, models = models, group = group))
       }
  else
    {return(list(sum.residual = sum.residual, models = models))}

}

#   ------------------------------------------------------------------
#   The function determineK calculate the R2 and adjusted
#   R2 of the K-regression clustering for K=1,...,Kmax.
#   An elbow plot can be draw to determine the optimal K
#   when the R2 is obtained.
#   ------------------------------------------------------------------
determineK<-function(data,rep.input,distance.input,Kmax)
{
  r2.K<-NULL
  for(K in 1:Kmax)
  {
    cluster<-NULL
    rep<-1
    seed<-sample(100000,1)
    seed.vec<-NULL
    while (rep <= rep.input)
    {
      t<-try(K.regressions(data=data,K=K,list.group=T, seed=seed, distance =
          distance.input))
      if("try-error" %in% class(t)) {seed<-seed+1; next}
      else
      {
        t<-c(t[[1]],t[[2]])
        cluster<-rbind(cluster,t)
        rep<-rep+1
        seed.vec<-c(seed.vec,seed)
        seed<-seed+123
      }
    }

    min.sr<-min(cluster[,1])

    opt.position<-which(cluster[,1]==min.sr)[1]
    result<-K.regressions(data,K,seed=seed.vec[opt.position],list.group=T)
    group<-result$group
    m<-result$models
    m<-matrix(m,K,2,byrow=T)
```

```
    yhat<-matrix(0,nrow(data),2)
    r2<-rep(0,K)
    for (k in 1:K)
    { pos<-which(group==k)
      yhat[pos,1]<-m[k,1]+m[k,2]*ifelse(rep(m[k,2]>0,length(pos)),data[pos,3],
          data[pos,4])
      yhat[pos,2]<-m[k,1]+m[k,2]*ifelse(rep(m[k,2]>0,length(pos)),data[pos,4],
          data[pos,3])
      r2[k]<-cov.int(yhat[pos,])/cov.int(data[pos,1:2])
    }

    weighted.r2<-weighted.mean(r2,table(group))
    r2.K<-c(r2.K,1-weighted.r2)
  }
  p<-ncol(data)/2-1
  adjusted.r2<-(1-r2.K)-r2.K*(1:K)*p/(nrow(data)-(1:K)*p-1)

  list(r2.K=r2.K,adjusted.r2=adjusted.r2)
}
```

Chapter 4

Linear Clustering Using Orthogonal Regression for Interval-Valued Data

In chapter 3, we proposed an algorithm to recover multiple linear relationships between a response variable and predictor variables for interval-valued data. That algorithm is a supervised learning algorithm. More commonly, it is of interest to group a data set into different clusters for which each cluster clusters around a linear line or a hyperplane of multiple dimensions. Under such a scenario, we have an unsupervised learning process and there is not necessarily a specific response variable. The proposed algorithm in chapter 3 can be applied to this kind of problem by trying each of the variables as the possible response variable. As Van Aelst et al. (2006) pointed out, however, it could be misleading using a supervised learning algorithm on this unsupervised problem, since the clustering results are usually different when using a different variable as the response variable. There is no way to determine which result is the correct one or the best one. In addition, it is computationally intensive to try each variable as the response variable.

In this chapter, we propose an unsupervised clustering algorithm by applying a proposed symbolic orthogonal regression methodology. The orthogonal regression method utilizes the

total least squares to minimize the sum of squares of error in terms of orthogonal distances between the observations and the predicted hyperplane. It does not require a response variable; thus, it is appropriate to our unsupervised clustering of the data clustered around multiple hyperplanes. In the reminder of this chapter, we will present the symbolic orthogonal regression methodology and orthogonal distance in section 4.1. The clustering algorithm using this symbolic orthogonal regression method is given in section 4.2. Section 4.3 proposes multiple methods of determining the optimal number of clusters K. A simulation study and an application are given in section 4.4 and section 4.5, respectively.

4.1 Symbolic Orthogonal Regression and Orthogonal Distance

Orthogonal regression (OR) methodology was originally derived for a measurement error model for classical data in Fuller (2009). For one dimension, we set up the model

$$y_i = \beta_0 + \beta_1 x_i,\ i = 1, \ldots, n, \tag{4.1}$$

where $x_1, \ldots, x_n$ is a random sample of the predictor variable x, and $y_1, \ldots, y_n$ is a random sample of the response variable y. For the measurement error model, x_i and y_i cannot be observed directly. Instead, we observe

$$\begin{aligned} \mu_i &= x_i + u_i, \\ \nu_i &= y_i + \epsilon_i, \end{aligned} \tag{4.2}$$

where u_i are independent and identically distributed (iid), $N(0, \sigma_u)$, $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon)$ for $i = 1, \ldots, n$, $\boldsymbol{u} = (u_1, \ldots, u_n)$ and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)$ are independent. Combining (4.1) and (4.2), we have $y_i = \beta_0 + \beta_1 x_i + \epsilon_i,\ i = 1, \ldots, n$. When x_i is an unknown fixed constant, the model is a functional model; while it is a structural model, if x_i is a random variable. If the error

variance ratio

$$\eta = \frac{\text{var}(\nu|x)}{\text{var}(\mu|x)} = \frac{\sigma_\epsilon^2}{\sigma_u^2} \tag{4.3}$$

is known, the estimators of the orthogonal regression model are obtained by solving

$$\min_{\beta_0,\beta_1,x_1,\ldots,x_n} \sum_{i=1}^{n} [(\nu_i - \beta_0 - \beta_1 x_i)^2/\eta + (\mu_i - x_i)^2]. \tag{4.4}$$

If $\eta = 1$, the solution of (4.4) minimizes the sum of squares of the orthogonal distance between the vector (y_i, x_i) and the linear line $y = \beta_0 + \beta_1 x$. The orthogonal distance between a vector $\boldsymbol{x}_i$ and a hyperplane $\boldsymbol{x}\boldsymbol{\beta} = \alpha$ given a constant vector $(\boldsymbol{\beta}, \alpha)$ is defined as

$$\|\boldsymbol{\beta}\|_2^{-1} |\boldsymbol{x}_i\boldsymbol{\beta} - \alpha|, \tag{4.5}$$

which is the Euclidean distance between $\boldsymbol{x}_i$ and its projected point onto the hyperplane $\boldsymbol{x}\boldsymbol{\beta} = \alpha$.

Unlike the classical linear regression method, orthogonal regression methods minimize the sum of squares of orthogonal distances between the data points and the fitted model. Thus, the fitted model is the same when treating either x or y as the response variable.

For multiple dimensions, the measurement model is

$$\begin{aligned} y_i &= \boldsymbol{x}_i\boldsymbol{\beta}, \\ (\nu_i, \boldsymbol{\mu}_i) &= (y_i, \boldsymbol{x}_i) + (\epsilon_i, \boldsymbol{u}_i), \end{aligned} \tag{4.6}$$

for $i = 1, \ldots, n$, where the parameter $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the set of coefficients that need to be estimated, $\boldsymbol{x}_i$ is the p dimensional predictor variables, $(\nu_i, \boldsymbol{\mu}_i)$ are the observed values while $(y_i, \boldsymbol{x}_i)$ are the true values, $\boldsymbol{e}_i = (\epsilon_i, \boldsymbol{u}_i)$ is the measurement error, and ϵ_i and $\boldsymbol{u}_i$ are independent. Let $\boldsymbol{e}_i \overset{iid}{\sim} N(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{e}})$, where $\boldsymbol{\Sigma}_{\boldsymbol{e}} = \boldsymbol{\gamma}_{\boldsymbol{e}}\sigma^2$ and $\boldsymbol{\gamma}_{\boldsymbol{e}}$ is known. Then, the orthogonal

regression estimator is obtained by minimizing

$$\sum_{i=1}^{n}(\nu_i - \boldsymbol{x}_i\boldsymbol{\beta}, \boldsymbol{\mu}_i - \boldsymbol{x}_i)\boldsymbol{\gamma}_e^{-1}(\nu_i - \boldsymbol{x}_i\boldsymbol{\beta}, \boldsymbol{\mu}_i - \boldsymbol{x}_i)'. \tag{4.7}$$

When $\boldsymbol{\gamma_e} = \boldsymbol{I}$, equation (4.7) is a $(p+1)$ dimension orthogonal distance between the data points and the fitted model $y_i = \boldsymbol{x}_i\boldsymbol{\beta}$; otherwise, it is a general orthogonal distance. Again, the fitted model is the same when treating any one variable of $(\nu_i, \boldsymbol{\mu}_i)$ as the response variable.

More generally, let $\boldsymbol{\mu}_i$ be p dimensional observations with

$$\boldsymbol{\mu}_i = \boldsymbol{x}_i + \boldsymbol{u}_i,\ i = 1, \ldots, n, \tag{4.8}$$

where $\boldsymbol{x}_i$ is the true unobserved value and $\boldsymbol{u}_i \overset{iid}{\sim} N(\boldsymbol{0}, \boldsymbol{\sigma_u})$ is the measurement error, and $\boldsymbol{\sigma_u}$ is the covariance matrix of the measurement errors. The unobserved $\boldsymbol{x}$ follows the linear relationship

$$\boldsymbol{x}\boldsymbol{\beta} = \alpha, \tag{4.9}$$

where $\alpha \in \mathbb{R}$, and $\boldsymbol{\beta} \in \mathbb{R}^p$. If $\boldsymbol{\sigma_u}$ is known, the orthogonal regression estimator is obtained by minimizing

$$\begin{aligned} &\sum_{i=1}^{n}(\boldsymbol{\mu}_i - \boldsymbol{x}_i)\boldsymbol{\sigma}_{\boldsymbol{u}}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{x}_i)' \\ &\text{s.t.}\ \ \boldsymbol{x}_i\boldsymbol{\beta} = \alpha, \end{aligned} \tag{4.10}$$

which is a minimization problem with a linear constraint. When $\boldsymbol{\sigma_u} = \sigma^2\boldsymbol{I}$ with σ^2 known, equation (4.10) is the simple orthogonal distance. The solution of equation (4.10) is a hyperplane that crosses the sample mean vector, $\bar{\boldsymbol{\mu}}$, of $\boldsymbol{\mu}$. Thus, $\boldsymbol{\beta}$ satisfies

$$(\boldsymbol{x} - \bar{\boldsymbol{\mu}})\boldsymbol{\beta} = 0. \tag{4.11}$$

Then, the $\boldsymbol{\beta}$ minimizing equation (4.10) is given by

$$(\boldsymbol{m_\mu} - \hat{\lambda}\boldsymbol{I})\hat{\boldsymbol{\beta}} = 0, \tag{4.12}$$

where $\hat{\lambda}$ is the smallest eigenvalue of $\boldsymbol{m_\mu}$, and $\boldsymbol{m_\mu}$ is the covariance matrix of $\boldsymbol{\mu}$. The estimator $\hat{\boldsymbol{\beta}}$ is also called total least squares estimator of $\boldsymbol{\beta}$.

Equation (4.11) is a hyperplane that crosses the mean of the data and is perpendicular to the eigenvector associated with the smallest eigenvalue of the covariance matrix of $\boldsymbol{\mu}$. The connection between orthogonal regression and principal component analysis (PCA) is created by the eigenvalue decomposition of the covariance matrix of $\boldsymbol{\mu}$. When applying the orthogonal regression method for clustering, it is not necessary to specify a response variable.

4.1.1 Orthogonal Regression for Interval-Valued Data

A perspective of symbolic PCA (simple symbolic orthogonal regression method) The orthogonal regression method for interval-valued data can be implemented based on the connection between the orthogonal regression and the principal component analysis (PCA) methods. Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be a p dimensional interval-valued data set with n observations where a realization of X_j is $[x_{ja}, x_{jb}]$, $j = 1, \ldots, p$. We assume that the p variables in $\boldsymbol{X}$ follow the linear relation $\boldsymbol{x\beta} = \alpha$ where $\alpha \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}$. The simple orthogonal regression coefficient estimate is obtained by minimizing

$$\begin{aligned} &\sum_{i=1}^{n} (\boldsymbol{x}_i - \ddot{\boldsymbol{x}}_i)(\boldsymbol{x}_i - \ddot{\boldsymbol{x}}_i)' \\ &\text{s.t.} \quad \ddot{\boldsymbol{x}}_i\boldsymbol{\beta} = \alpha, \end{aligned} \tag{4.13}$$

where $\ddot{\boldsymbol{x}}_i$ is the estimated value of $\boldsymbol{x}_i$, and $\boldsymbol{x}_i$ is the i^{th} observation of $\boldsymbol{X}$. Analogously with the solution of equation (4.10), the estimate of $\boldsymbol{\beta}$ is given by

$$(m_{\boldsymbol{X}} - \hat{\lambda}\boldsymbol{I})\hat{\boldsymbol{\beta}} = 0, \tag{4.14}$$

where $m_{\boldsymbol{X}}$ is the covariance matrix of $\boldsymbol{X}$, and $\hat{\lambda}$ is the smallest eigenvalue of the $m_{\boldsymbol{X}}$. The covariance matrix $m_{\boldsymbol{X}}$ can be constructed by the definition of sample variance and covariance for the interval-valued data in definitions 2.1.7 and 2.1.8, and $\hat{\lambda}$ can be derived accordingly. Then, the fitted simple orthogonal regression model for the interval-valued data $\boldsymbol{X}$ is as follows:

$$(\boldsymbol{x} - \bar{\boldsymbol{X}})\hat{\boldsymbol{\beta}} = 0, \tag{4.15}$$

where $\bar{\boldsymbol{X}} = (\bar{X}_1, \ldots, \bar{X}_p)$ is the mean vector of $\boldsymbol{X}$.

Usually, we standardize the variables before the principal component analysis to avoid the effect of different scales. The interval-valued random variable can be standardized in a similar way as for the standardization of classical values. Let $\bar{X}$ be the sample mean for an interval-valued variable X, and S be the sample variance; then, the standardized interval value of X is

$$Y = \frac{X - \bar{X}}{\sqrt{S}}. \tag{4.16}$$

It can be verified that the symbolic sample mean and variance of Y is 0 and 1, respectively. Once the variable is standardized, the orthogonal regression method can be implemented through the covariance matrix of Y. This orthogonal regression method obtained by implementing symbolic PCA is called a simple symbolic orthogonal regression method.

A perspective of a measurement error model (general symbolic orthogonal regression method)

The classical data regression method only considers the "between" variance and covariance in

equation (2.7) and equation (2.9) since there do not exist a "within" variance nor a "within" covariance. To additionally consider the internal or within variance and covariance (equation (2.7a), equation (2.10a)) is the main challenge for interval-valued data analysis. The simple symbolic orthogonal regression method is one way to integrate the internal variances and covariances of interval-valued data. Next, we propose an alternative way to construct an orthogonal regression method for interval-valued data from the perspective of a measurement error model.

Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be a p dimensional interval-valued data set with n observations. To look at the problem from a measurement error perspective, we treat $\boldsymbol{X}^{(c)}$, the center points of $\boldsymbol{X}$, as the observed values of the p variables. We assume that there exists a matrix $\ddot{\boldsymbol{x}} = (\ddot{\boldsymbol{x}}_1, \ldots, \ddot{\boldsymbol{x}}_n)^T$ such that

$$\begin{aligned} \boldsymbol{x}_i^{(c)} &= \ddot{\boldsymbol{x}}_i + \boldsymbol{u}_i, \\ \ddot{\boldsymbol{x}}_i \boldsymbol{\beta} &= \alpha, \end{aligned} \tag{4.17}$$

where $\ddot{\boldsymbol{x}}_i = (\ddot{x}_{i1}, \ldots, \ddot{x}_{ip})$, $\ddot{x}_{ij} \in [x_{ija}, x_{ijb}]$, $i = 1, \ldots, n$, $j = 1, \ldots, p$, and where $\boldsymbol{u}_i = (u_{i1}, \ldots, u_{ip})$ is the measurement error of $\ddot{\boldsymbol{x}}_i$. For an interval value $x_{ij} = [x_{ija}, x_{ijb}]$, the range of the interval can be seen as the measurement error of the true value $\ddot{x}_{ij}$. We assume the measurement error of x_{ij} follows a uniform distribution,

$$u_{ij} \sim U(x_{ija} - x_{ij}^{(c)}, x_{ijb} - x_{ij}^{(c)}). \tag{4.18}$$

We have $E(u_{ij}) = 0$ for $i = 1, \ldots, n$, $j = 1, \ldots, p$. Furthermore, we assume that the measurement error for a particular variable X_j follows a normal distribution overall. In other words, we assume that $\boldsymbol{u}_i \overset{iid}{\sim} N(0, \boldsymbol{\Sigma_u})$, $i = 1, \ldots, n$. Let $U_j = ([x_{1ja} - x_{ij}^{(c)}, x_{1jb} - x_{1j}^{(c)}], \ldots, [x_{nja} - x_{nj}^{(c)}, x_{njb} - x_{nj}^{(c)}])^T$. Then, the covariance matrix of the normal distribution

$\boldsymbol{\Sigma_u}$ can be estimated by the method of moments as follows:

$$\boldsymbol{\Sigma_u} = \begin{bmatrix} \text{var}(U_1) & \dots & \text{cov}(U_p, U_1) \\ \vdots & \ddots & \vdots \\ \text{cov}(U_1, U_p) & \dots & \text{var}(U_p)) \end{bmatrix}, \tag{4.19}$$

where $\text{cov}(U_j, U_{j'})$ is the symbolic covariance between U_j and $U_{j'}$, $j, j' = 1, \dots, p$.

Without loss of generality, we assume $\alpha = 0$; in other words, the hyperplane always crosses the origin of the coordinates. If that is not true, we just need to centralize the data so that $\boldsymbol{X}^* = \boldsymbol{X} - \bar{\boldsymbol{X}}$ where $\boldsymbol{X}^*$ is the centralized data, $\bar{\boldsymbol{X}} = (\bar{X}_1, \dots, \bar{X}_p)$ is the mean vector of $\boldsymbol{X}$.

Since we assume that $\boldsymbol{u}_i \sim N(\boldsymbol{0}, \boldsymbol{\Sigma_u})$, the density function of $\boldsymbol{u}_i$ is

$$(2\pi)^{-p/2} |\boldsymbol{\Sigma_u}|^{-1/2} \exp\{-\frac{1}{2} \boldsymbol{u} \boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1} \boldsymbol{u}'\}. \tag{4.20}$$

For the data $\boldsymbol{X}$ with n observations, the log likelihood function is

$$\log L = -\frac{np}{2} \log(2\pi) - \frac{n}{2} \log |\boldsymbol{\Sigma_u}| - \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{x}_i^c - \ddot{\boldsymbol{x}}_i) \boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1} (\boldsymbol{x}_i^c - \ddot{\boldsymbol{x}}_i)^T. \tag{4.21}$$

The estimation of $\boldsymbol{\beta}$ can be derived by solving the following minimization problem:

$$\begin{aligned} &\min(-\log L) \\ &\text{s.t.} \quad \ddot{\boldsymbol{x}}_i \boldsymbol{\beta} = 0. \end{aligned} \tag{4.22}$$

We adapt the derivation from Fuller (2009) a little to estimate the $\boldsymbol{\beta}$. To obtain a unique solution of $\boldsymbol{\beta}$, a constraint on $\boldsymbol{\beta}$ is needed. Without loss of generality, we set the first element of $\boldsymbol{\beta}$, $\beta_1 = -1$, and denote the rest of the elements $(\beta_2, \dots, \beta_p)$ as $\boldsymbol{\beta}_{-1}$. Similarly, we denote $(\ddot{x}_{i2}, \dots, \ddot{x}_{ip})$ as $\ddot{\boldsymbol{x}}_{i(-1)}$, and $(x_{i2}^{(c)}, \dots, x_{ip}^{(c)})$ as $\boldsymbol{x}_{i(-1)}^{(c)}$. For given $\boldsymbol{\beta}$, the problem in equation

(4.22) can be solved with respect to $\boldsymbol{x}_i^c$, $i = 1, \ldots, n$, which is equivalent to minimizing

$$(x_{i1}^{(c)} - \ddot{\boldsymbol{x}}_{i(-1)}\boldsymbol{\beta}_{-1}, \boldsymbol{x}_{i(-1)}^{(c)} - \ddot{\boldsymbol{x}}_{i(-1)})\boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1}(x_{i1}^{(c)} - \ddot{\boldsymbol{x}}_{i(-1)}\boldsymbol{\beta}_{-1}, \boldsymbol{x}_{i(-1)}^{(c)} - \ddot{\boldsymbol{x}}_{i(-1)})^T. \tag{4.23}$$

This is equivalent to a weighted least square estimation and the estimator of $\ddot{\boldsymbol{x}}_i^T$ is

$$\begin{aligned} \ddot{\boldsymbol{x}}_i^T &= (\boldsymbol{\beta}_{-1}, \boldsymbol{I}_{p-1})^T[(\boldsymbol{\beta}_{-1}, \boldsymbol{I}_{p-1})\boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1}(\boldsymbol{\beta}_{-1}, \boldsymbol{I}_{p-1})^T]^{-1})(\boldsymbol{\beta}_{-1}, \boldsymbol{I}_{p-1})\boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1}(\boldsymbol{x}_i^{(c)})^T \\ &= (\boldsymbol{x}_i^{(c)})^T - \boldsymbol{\Sigma}_{\boldsymbol{u}}\boldsymbol{\beta}(\boldsymbol{\beta}^T\boldsymbol{\Sigma}_{\boldsymbol{u}}\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^T(\boldsymbol{x}_i^{(c)})^T. \end{aligned} \tag{4.24}$$

By substituting equation (4.24) into equation(4.21), the estimation of $\boldsymbol{\beta}$ follows the derivation in Fuller (2009). The solution of $\boldsymbol{\beta}$ is given by

$$(\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}} - \hat{\lambda}\boldsymbol{\Sigma}_{\boldsymbol{u}})\hat{\boldsymbol{\beta}} = 0, \tag{4.25}$$

where $\hat{\lambda}$ is the smallest root of

$$|\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}} - \hat{\lambda}\boldsymbol{\Sigma}_{\boldsymbol{u}}| = 0, \tag{4.26}$$

and $\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}} = n^{-1}\boldsymbol{X}^{(c)\prime}\boldsymbol{X}^{(c)}$. The $\boldsymbol{\beta}$ obtained from equation (4.25) is usually constrained to be a unit vector. If we set $\beta_1 = -1$, then the estimation of $\boldsymbol{\beta}_{-1}$ is

$$\hat{\boldsymbol{\beta}}_{-1} = (\boldsymbol{M}_{\boldsymbol{X}_{-1}^{(c)}\boldsymbol{X}_{-1}^{(c)}} - \hat{\lambda}\boldsymbol{\Sigma}_{22})^{-1}(\boldsymbol{M}_{\boldsymbol{X}_{-1}^{(c)}X_1^{(c)}} - \hat{\lambda}\boldsymbol{\Sigma}_{12}), \tag{4.27}$$

where $\boldsymbol{X}_{-1}^{(c)} = (X_2^{(c)}, \ldots, X_p^{(c)})$, $\hat{\lambda}$ is the smallest root of equation (4.26), and $\boldsymbol{\Sigma}_{12}$ is the covariance matrix of the measurement errors between $X_1^{(c)}$ and $\boldsymbol{X}_{-1}^{(c)}$ while $\boldsymbol{\Sigma}_{22}$ is the covariance matrix of the measurement errors of $\boldsymbol{X}_{-1}^{(c)}$. Alternatively, we can rewrite $\boldsymbol{\Sigma}_{\boldsymbol{u}}$ as submatrices

in the form

$$\boldsymbol{\Sigma_u} = \begin{bmatrix} \underset{1\times1}{\boldsymbol{\Sigma}_{11}} & \underset{1\times(p-1)}{\boldsymbol{\Sigma}_{12}} \\ \underset{(p-1)\times1}{\boldsymbol{\Sigma}_{21}} & \underset{(p-1)\times(p-1)}{\boldsymbol{\Sigma}_{22}} \end{bmatrix}, \tag{4.28}$$

where $\boldsymbol{\Sigma}_{11}$ is the variance of the measurement error of X_1.

Note that the unit vector $\hat{\boldsymbol{\beta}}$ obtained from equation (4.25) and the vector $\hat{\boldsymbol{\beta}} = (1, \hat{\boldsymbol{\beta}}_{-1}^T)^T$ constructed from equation (4.27) are equivalent in terms of a constant multiplier. According to our assumption of $\ddot{\boldsymbol{x}}\boldsymbol{\beta} = 0$, we usually centralize the data first and then derive the coefficient estimate of $\boldsymbol{\beta}$. Eventually, when we transfer back to the original locations of the $\boldsymbol{x}$ vector, the hyperplane has the form $(\boldsymbol{x} - \bar{\boldsymbol{X}})\hat{\boldsymbol{\beta}} = 0$, $\boldsymbol{x} \in \mathbb{R}^p$.

From Fuller (2009), the smallest root of $|\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}} - \hat{\lambda}\boldsymbol{\Sigma_u}| = 0$ can be constructed by the following process. A unitary vector $\boldsymbol{t}$ that satisfies $(\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}} - \hat{\lambda}\boldsymbol{\Sigma_u})\boldsymbol{t} = 0$ is called the eigenvector of $\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}}$ in the metric of $\boldsymbol{\Sigma_u}$. Let the columns of matrix $\boldsymbol{Q}$ be the eigenvector of $\boldsymbol{\Sigma_u}$, and let $\boldsymbol{\Omega} = \text{diag}\{\omega_1, \ldots, \omega_p\}$, where ω_j is the j^{th} eigenvalue of $\boldsymbol{\Sigma_u}$. We can construct the matrix $\boldsymbol{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_p]$ as

$$\boldsymbol{T} = \boldsymbol{Q}\boldsymbol{\Omega}^{-1/2}\boldsymbol{P}, \tag{4.29}$$

where $\boldsymbol{\Omega}^{-1/2} = \text{diag}\{\omega_1^{-1/2}, \ldots, \omega_p^{-1/2}\}$, and the columns of $\boldsymbol{P}$ are the eigenvectors of the matrix $\boldsymbol{\Omega}^{-1/2}\boldsymbol{Q}^T\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}}\boldsymbol{Q}\boldsymbol{\Omega}^{-1/2}$. Then, we have

$$\boldsymbol{T}^T\boldsymbol{M}_{\boldsymbol{X}^{(c)}\boldsymbol{X}^{(c)}}\boldsymbol{T} = \text{diag}\{\lambda_1, \ldots, \lambda_p\}, \tag{4.30}$$

where $\lambda_1, \ldots, \lambda_p$ are the roots of equation (4.26) from which the smallest roots $\hat{\lambda}$ can be obtained and the associated $\hat{\boldsymbol{\beta}}$ can be calculated accordingly.

When the measurement errors from the p variables are independent, or the covariance matrix of the measurement error $\boldsymbol{\Sigma_u}$ is a diagonal matrix, the expression in equation (4.23)

is similar to that for a weighted least squares estimation. The objective function of equation (4.23) weights less if the measurement error for a variable is large. Generally, we call the measurement error model for an interval-valued data regression method a general symbolic orthogonal regression method.

4.1.2 Symbolic Orthogonal Distance

We have established the orthogonal regression methodology for interval-valued data in section 4.1.1. In this section, we define the distance between an interval-valued observation and a hyperplane. Since the principle of the simple and general symbolic orthogonal regression methods are different, the distances for the two regressions are defined separately. A symbolic orthogonal distance is defined for the simple symbolic orthogonal regression method, while a general orthogonal distance is defined for the general symbolic orthogonal regression method.

For classical data, the orthogonal regression residual of an observation x_i for a fitted hyperplane $\boldsymbol{x}\boldsymbol{\beta} = \alpha$ is

$$e_i = \frac{\boldsymbol{x}_i\boldsymbol{\beta} - \alpha}{\|\boldsymbol{\beta}\|_2}, \tag{4.31}$$

where $\boldsymbol{\beta}$ is a constant vector and α is a constant value. When the coefficient $\boldsymbol{\beta}$ is an unit vector, $\|\boldsymbol{\beta}\|_2 = 1$, the distance in equation (4.31) can be simplified as follows:

$$e_i = \frac{\langle \boldsymbol{\beta}, (\boldsymbol{x}_i - \bar{\boldsymbol{X}}) \rangle}{\|\boldsymbol{\beta}\|_2} = \langle \boldsymbol{\beta}, (\boldsymbol{x}_i - \bar{\boldsymbol{X}}) \rangle, \tag{4.32}$$

where $\langle \cdot, \cdot \rangle$ is the inner product between two vectors. The orthogonal distance between the observation $\boldsymbol{x}_i$ and the fitted hyperplane is the absolute value of the residual e_i.

For the simple symbolic orthogonal regression method, we define the simple orthogonal distance between an interval-valued observation and the fitted hyperplane in a similar way as for classical data. For interval-valued data, the coefficients $\boldsymbol{\beta}$ and α in equation (4.31) are both classical values, the only interval value is the observation $\boldsymbol{x}_i$. A natural way to define

the orthogonal distance between $\boldsymbol{x}_i$ and the hyperplane $\boldsymbol{x}\boldsymbol{\beta} = \alpha$ is the orthogonal distance between the center point of the $\boldsymbol{x}_i$ and the hyperplane

$$d_i^c = \|\boldsymbol{\beta}\|_2^{-1}|(\boldsymbol{x}_i^{(c)}\boldsymbol{\beta} - \alpha)| = \|\boldsymbol{\beta}\|_2^{-1}|((\boldsymbol{x}_{ia} + \boldsymbol{x}_{ib})\boldsymbol{\beta}/2 - \alpha)|, \tag{4.33}$$

where $\boldsymbol{x}_{ia}$ and $\boldsymbol{x}_{ib}$ are the end points vector of $\boldsymbol{x}_i$, $\boldsymbol{x}_i^{(c)}$ is the center points vector. We call this definition in equation (4.33) the center distance, denoted as d^c. Note the center distance defined in equation (4.33) is different from the center distance defined in equation (3.5). Equation (4.33) defines distance between an interval value and a hyperplane while equation (3.5) define the distance between two interval values. For the model of equation (4.15), the distance can be simplified as $|\langle\boldsymbol{\beta}, (\boldsymbol{x}_i^{(c)} - \bar{\boldsymbol{X}}\rangle|$, where $\bar{\boldsymbol{X}}$ is the mean vector of $\boldsymbol{X}$.

Now, we use an alternative way to define the simple orthogonal distance. Given a particular interval-valued observation $\boldsymbol{x}_i$ and a hyperplane $\boldsymbol{x}\boldsymbol{\beta} = \alpha$, we define the minimum and maximum orthogonal distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}\boldsymbol{\beta} = \alpha$ as follows:

$$\begin{aligned} D_i^{min} &= \min_{\boldsymbol{x}\in\boldsymbol{x}_i} \|\boldsymbol{\beta}\|_2^{-1}(\boldsymbol{x}^T\boldsymbol{\beta} - \alpha), \\ D_i^{max} &= \max_{\boldsymbol{x}\in\boldsymbol{x}_i} \|\boldsymbol{\beta}\|_2^{-1}(\boldsymbol{x}^T\boldsymbol{\beta} - \alpha), \end{aligned} \tag{4.34}$$

where $\boldsymbol{x} \in \boldsymbol{x}_i \equiv \{\boldsymbol{x} = (x_1, \ldots, x_p) : x_1 \in [x_{i1a}, x_{i1b}], \ldots, x_p \in [x_{ipa}, x_{ipb}]\}$. We can see that D_i^{min} is the minimum distance between any point within the hypercube of $\boldsymbol{x}_i$ and the hyperplane $\boldsymbol{x}^T\boldsymbol{\beta} = \alpha$, while D_i^{max} is the maximum distance. Note that the values of D_i^{min} and D_i^{max} can be either positive or negative. The definition in equation (4.34) can be rewritten as follows:

$$\begin{aligned} D_i^{min} &= \|\boldsymbol{\beta}\|_2^{-1}(\sum_{l:\beta_l>0} x_{ila}\beta_l + \sum_{m:\beta_m<0} x_m\beta_{imb} - \alpha), \\ D_i^{max} &= \|\boldsymbol{\beta}\|_2^{-1}(\sum_{l:\beta_l>0} x_{ilb}\beta_l + \sum_{m:\beta_m<0} x_m\beta_{ima} - \alpha). \end{aligned} \tag{4.35}$$

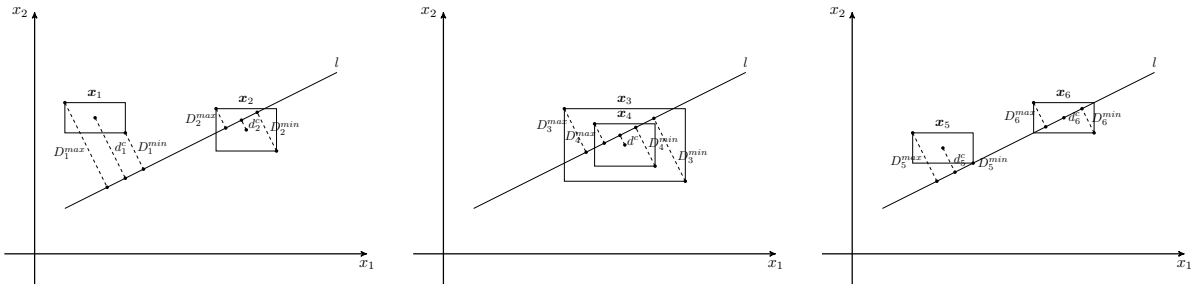


Figure 4.1: Examples of orthogonal distance defined by D^{min} and D^{max}

There are multiple ways to define the orthogonal distance by D_i^{min} and D_i^{max}. We consider three particular ways in the following context. First, we define the orthogonal distance as the absolute value of the average of D_i^{min} and D_i^{max},

$$d_i = \frac{1}{2}|D_i^{min} + D_i^{max}|, \tag{4.36}$$

which is equivalent to the center distance, d_i^c, defined in equation (4.33).

The second way is defined as the average of the absolute values of D_i^{min} and D_i^{max},

$$d_i^a = \frac{1}{2}(|D_i^{min}| + |D_i^{max}|). \tag{4.37}$$

We call the distance, d_i^a, in equation (4.37) average absolute distance to distinguish it form the center distance d_i^c of equation (4.33).

Figure 4.1 compares the center distance and the average absolute distance by consideration of 2-dimensional data examples. In Figure 4.1 (a), the observation $\boldsymbol{x}_1$ is generally further from the line l than is the observation $\boldsymbol{x}_2$. Both the center distance and the average absolute distance are able to measure that distance difference appropriately. The observations $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ in Figure 4.1 (b) have the same center points but the interval ranges are

larger for $\boldsymbol{x}_3$ for both dimensions. The average distance from l is larger for the points within $\boldsymbol{x}_3$ than are the points within $\boldsymbol{x}_4$. The average absolute distance can capture this distance difference, while the center distance for the two observations are the same. For this scenario, the average absolute distance is more appropriate to measure the orthogonal distance. The two observations $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$ in Figure 4.1 (c) have the same interval ranges but the positions are different. The average distance between the points within the observation and l is larger for $\boldsymbol{x}_5$ than for $\boldsymbol{x}_6$, which the center distance can capture but the average absolute distance cannot as this latter distance is the same for the two observations. For this example, the average absolute distance is more appropriate than is the center distance.

To address the inappropriateness of the center distance and average absolute distance under some certain scenario, we introduce a third distance measure. We revise the definition of equation (4.36) and equation (4.37) as follows

$$d_i^m = \frac{1}{2}[\min(|D_i^{min}|, |D_i^{max}|)\mathbf{1}_{D_i^{min}D_i^{max}>0} + \max(|D_i^{min}|, |D_i^{max}|)]. \tag{4.38}$$

The distance d_i^m is called the min max distance to distinguish it from the center distance and the average absolute distance. The definition in equation (4.38) is equivalent to

$$d_i^m = \frac{1}{2}\|\boldsymbol{\beta}\|_2^{-1}(\min_{\boldsymbol{x}\in\boldsymbol{x}_i}|\boldsymbol{x}^T\boldsymbol{\beta} - \alpha| + \max_{\boldsymbol{x}\in\boldsymbol{x}_i}|\boldsymbol{x}^T\boldsymbol{\beta} - \alpha|), \tag{4.39}$$

where $\boldsymbol{x} \in \boldsymbol{x}_i$ is defined the same as for equation (4.34). The interpretation of d_i^m is that when the hyperplane does not cross an observation, then the orthogonal distance d_i^m is the same as the distance d_i^a. When the hyperplane crosses an observation, the orthogonal distance d_i^m is half of the maximum absolute distance, D_i^{max} of equation (4.34), between points within the observation and the hyperplane. For the three examples in Figure 4.1, we can see that $d_1^m > d_2^m$, $d_3^m > d_4^m$, and $d_5^m > d_6^m$, which is appropriate for all the examples.

The min max distance has an advantage when comparing it with the center distance and

average absolute distance. We will apply the min max distance in sections 4.3, 4.4, and 4.5. Though the center distance has some disadvantages under certain situations, we would still use it due to its simplicity and ease of interpretation, but we will not implement the average absolute distance due to its obvious defects.

For the general symbolic orthogonal regression method, the objective is to minimize the sum of squares of the general orthogonal distances between an interval-valued observation $\boldsymbol{x}_i$ and the fitted hyperplane $\boldsymbol{x}\boldsymbol{\beta} = \alpha$. The general orthogonal distance is defined from the likelihood function of equation(4.21) as

$$d^w = \sqrt{(\boldsymbol{x}_i^c - \ddot{\boldsymbol{x}}_i)\boldsymbol{\Sigma}_{\boldsymbol{u}}^{-1}(\boldsymbol{x}_i^c - \ddot{\boldsymbol{x}}_i)^T}, \tag{4.40}$$

where $\ddot{\boldsymbol{x}}_i$ is obtained by equation (4.24).

4.2 Orthogonal Regression Clustering Algorithm

After defining the orthogonal regression methodology and orthogonal distances for interval-valued data in section 4.1, we now present an algorithm that recovers multiple linear regression structures of a data set without specifying a response variable. Given an interval-valued data $\boldsymbol{X} = (X_1, \ldots, X_p)$ with n observations, we assume that the X_j, $j = 1, \ldots, p$, follows K different linear relationships,

$$\boldsymbol{x}^T\boldsymbol{\beta}_k = \alpha_k, \ k = 1, \ldots, K, \tag{4.41}$$

where $(\boldsymbol{\beta}_k, \alpha_k)$ is different for different k not just by multiplying a constant. The different relationships can be due to different groups but we do not have the information of the groups. For example, the relationships for different age groups are known to be different, but the information of age is unavailable. The relationships can also be different due to unknown

underlying groups.

The Orthogonal Regression Clustering Algorithm (ORCA) uses the orthogonal regression method to identify the multiple linear relationships within an interval-valued data set $\boldsymbol{X}$. This ORCA is similar to the algorithm in section 3.2 by starting with a random partition for a given number of clusters K. Then, it fits an orthogonal regression method within each cluster. The observations are regrouped to their closest cluster in terms of the orthogonal distance. These two steps continue iterating until a local minimum is reached. The optimal partition minimizes the aggregated sum of squares of orthogonal distances between observations and their hyperplanes. The algorithm would be repeated many times to search for the optimal partition. We describe the ORCA by the following detailed steps:

(i) *Scale the variables (optional)*: If we apply the orthogonal regression method by PCA, to avoid the scale effect on PCA from different variables, we first divide each variable by its sample variance so that all the variables are on the same scale. In other words, the variances from different variables are all one.

(ii) *Initialization*: Randomly assign each observation into one of the K clusters with equal probability to obtain an initial partition $P^0 = (C_1^0, \ldots, C_K^0)$, or partition the whole data set to K clusters based on prior knowledge.

(iii) *Representation*: On the l^{th} iteration, for each cluster of the partition $P^l = (C_1^l, \ldots, C_K^l)$, derive the hyperplane of equation (4.11) or equation (4.15) by the methods described in section 4.1. Denote the hyperplane for each of the K as $\boldsymbol{x}\boldsymbol{\beta}_k^l = \alpha_k^l$, where $\boldsymbol{\beta}_k^l$ and α_k^l are the coefficient estimates for the k^{th} cluster.

(iv) *Allocation*: Calculate the orthogonal distance between the observation $\boldsymbol{x}_i$ with the hyperplane of each cluster, $\boldsymbol{x}\boldsymbol{\beta}_k^l = \alpha_k^l$. Allocate the observation to its closest cluster in terms of the orthogonal distance. The distance, d_i, between an observation and its

cluster is

$$d_i^{(l+1)} = \inf_{k=1,\ldots,K} d_{ik}^{(l+1)}, \tag{4.42}$$

where d_{ik}^l is the distance defined in equation (4.33) or equation (4.34) for the k^{th} cluster.

(*v*) *Stop*: Repeat the step (*iii*) and (*iv*) until the average aggregate orthogonal distance

$$\frac{1}{n}\sum_{k=1}^{K}\sum_{\boldsymbol{x}_i \in P_k^l} d_i^l \tag{4.43}$$

is smaller than a predetermined criterion or the time of iteration is greater than a predetermined maximum number. In practice, usually 50 is a practical number to be set as the maximum number of iterations.

The step (*ii*), initialization, is crucial for the algorithm to converge onto the optimal partition. Without prior knowledge, to start the algorithm by randomly splitting the whole data set into K roughly equal size groups is the simplest way. For such an initialization, however, since each group of the starting partition is a random sample of the whole data set, the groups and their fitted hyperplanes may be too close. It can be either slow or hard to converge to obtain the optimal partition.

To improve the efficiency of the algorithm, we would like to introduce more differences between groups for the initial partition. Ideally, the closer the initial partition is to the true partition, the faster the algorithm will converge. We revise the initialization step by adopting the concept of d-subsets from Rousseeuw and Driessen (1999). Instead of randomly splitting the data into K roughly equal sized groups, we randomly sample K mutually exclusive d-subsets where $d = p + 1$ is the minimum number of observations to estimate the hyperplane in a group and a d-subsets is a sample of $\boldsymbol{X}$ with sample size d. The benefit of the d-subsets is that there is a relatively higher probability each of the d-subsets is from a different true cluster. From Van Aelst et al. (2006), the probability that each d-subsets is from different

true cluster is

$$p = \frac{\binom{n_1}{d}\binom{n_2}{d}\cdots\binom{n_K}{d}}{\binom{n}{Kd}}. \tag{4.44}$$

Although a d-subsets from a true cluster cannot guarantee its fitted hyperplane is close to the true hyperplane, it does have a higher probability that the fitted hyperplane is close to the true hyperplane. The revised initialization step can be described as follows:

(ii') Generate the starting partition by randomly selecting K mutually exclusive d-subsets from the data set $\boldsymbol{X}$. Each of the d-subsets is one group of the partition $P^0 = (C_1^0, \ldots, C_K^0)$.

The *representation* in step (iii) accordingly fits the K hyperplanes based on the K d-subsets. The simulation results in section 4.4 show the efficiency of the algorithm is improved significantly.

The number of times to repeat the algorithm to obtain a good initial partition so that the algorithm converges to the optimal partition can be roughly estimated by equation (4.44). For example, to have a 95% probability of obtaining at least one initial partition that has d points from each group, the number of different initial partitions we need to try is $\log(.05)/\log(1-p)$ (Van Aelst et al., 2006).

4.3 Determine the Optimal Number of Clusters K

Given the number of clusters K for a data set, we can implement the ORCA algorithm proposed in section 4.2. Sometimes, the number of clusters is indicated by some background knowledge. More often, however, the optimal number of clusters K is unknown and needs to be determined by some criterion. In this section, we propose some possible criteria to determine the optimal number of clusters K. Some are relatively subjective while others utilize statistical tests.

4.3.1 The Elbow Method

Although it is somewhat subjective, the elbow plot is still an intuitive and feasible way to determine the number of clusters K. For the orthogonal regression method, there is no corresponding definition of R-square or the proportion of variance explained as exists in principal component analysis. Instead, we use the sum of squares of orthogonal distances (SSOD) to decide the elbow point. The SSOD is the sum of squares of orthogonal distances between each observation and the hyperplane of the cluster to which it belongs to. For a interval valued data set with K clusters, the SSOD is defined as

$$SSOD = \sum_{k=1}^{K} \sum_{i=1}^{n_k} {d_i^{(k)}}^2, \tag{4.45}$$

where n_k is the sample size of the k^{th} cluster, $d_i^{(k)}$ is the orthogonal distance between the i^{th} observation in the k^{th} cluster and the hyperplane fitted in the k^{th} cluster.

The idea is that before the number of clusters K reaches the optimal number, the increase of K will add much information. After K reaches its optimal number, to continue to increase K, the marginal information gain drops. Such a pattern would indicate an elbow point if we plot the SSOD versus K. We illustrate how the elbow method works by an example.

Given a 3-dimension interval-valued data set $\boldsymbol{X} = (X_1, X_2, X_3)$, suppose that the three variables in $\boldsymbol{X}$ follow the 3 different linear relationships:

$$\begin{aligned} \text{cluster } 1 &: X_1 - 1.3X_2 - 1.5X_3 - 1 = 0, \\ 2 &: 0.222X_1 + 0.400X_2 + 0.667X_3 - 1 = 0, \\ 3 &: 0.029X_1 + 0.100X_2 - 0.286X_3 - 1 = 0. \end{aligned} \tag{4.46}$$

Suppose a total of 300 observations are simulated with 100 observations for each of the clusters. If the maximum number of clusters is set to be $K = 8$, we run the clustering

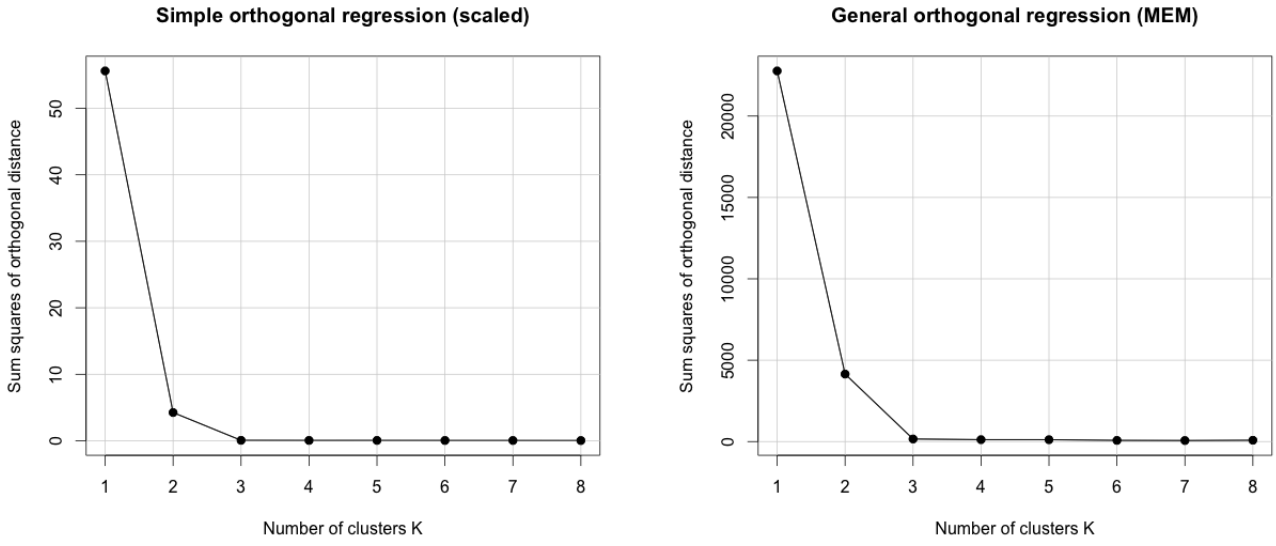
algorithm by the simple symbolic orthogonal regression method using the center distance and by the general symbolic orthogonal regression methods for $K = 1, \ldots, 8$. Further, the simple orthogonal regression method is based on the scaled data. The values of SSOD for the two methods for each K are summarized in Table 4.1.

Table 4.1: Sum of squares of orthogonal distances for $K = 1, \ldots, 8$

	K							
	1	2	3	4	5	6	7	8
Simple orthogonal regression	55.617	4.251	0.069	0.055	0.056	0.048	0.045	0.039
General orthogonal regression	22,772.304	4,148.512	159.032	117.521	115.488	74.082	64.699	82.121
SD_K (simple OR)	92.4%	98.4%	20.3%	-1.8%	14.3%	6.3%	13.3%	-
SD_K (general OR)	81.8%	96.2%	26.1%	1.7%	35.9%	12.7%	-26.9%	-

Unsurprisingly, we see from Table 4.1 that the SSOD decreases when the number of clusters K increases going from 55.617 for $K = 1$, down to 0.039 for $K = 8$, for the simple orthogonal regression method; and likewise for the general orthogonal regression method While both methods have decreasing values for SSOD as K increases, the values for the simple orthogonal regression method are considerably smaller than are those for the general orthogonal regression method; e.g., when $K = 3$, SSOD=0.069 for the simple orthogonal regression method, while SSOD=159.032 for the general orthogonal regression method. One reason for such a considerable difference is that the values of SSOD for simply orthgonal regression methods are calculated on a scaled data where the standard deviation of all the variables are scaled to be one. The second reason is the difference between the definitions of the simple orthogonal distance (see equation (4.36) and equation (4.38)) and the general orthogonal distance (see equation (4.40)). The general orthogonal distance involves the covariance matrix of the measurement errors.

Figure 4.2 (a) shows the plots of SSOD versus K, the elbow plots, for the simple orthogonal regression method and Figure 4.2 (b) for the general orthogonal regression method.



(a) (b)

Figure 4.2: Elbow plots using orthogonal regression for clustering

Both of the plots have a elbow point at $K = 3$, which indicates that the appropriate number of clusters is $K = 3$.

An equivalent approach to the elbow method without drawing the elbow plot itself is to calculate the percentage of SSOD decrements. The percentage of SSOD decrement is calculated as follows:

$$SD_K = \frac{SSOD_K - SSOD_{K+1}}{SSOD_K}, \tag{4.47}$$

where $SSOD_K$ is the SSOD when the number of clusters is K. The first SD_K value that is less than a predetermined cutoff value corresponds to the optimal K, which indicates that when the SSOD decrement is small by adding one more cluster, then, this added cluster is not necessary. The SD_K for the data set in equation (4.46) are calculated for each of $K = 1, \ldots, 7$. The results are shown in the row "SD_K (simple OR)" and row "SD_K (simple OR)" in Table 4.1, where "SD_K (simple OR)" is the SD_K for the simple orthogonal regression

method while "SD_K (general OR)" is the SD_K for the general orthogonal regression method. By SD_K definition of equation (4.47), the SD_K for $K = 1$ is defined, but the SD_K for $K = K^{max}$ is not defined for a given maximum number of clusters K^{max}. We can see that for the simple orthogonal regression method, there is a big drop of SD_K from 98.4% at $K = 2$ to 20.3% at $K = 3$. Any cutoff point of SD_K between 98% and 21% would determine the optimal number of clusters to be $K = 3$. Similarly, for the general orthogonal regression method, any cutoff point of SD_K between 27% and 96% would determine the optimal number of clusters $K = 3$. Similarly as for the elbow point, the determination of the cutoff point for SD_K is subjective.

The elbow point can be ambiguous under certain scenarios so that it is hard to determine the optimal number of clusters. For example, in Figure 4.2(a) for the simple orthogonal regression method, we might argue that $K = 2$ could be the elbow point. To overcome these disadvantages of the elbow method, we introduce other methods to determine K in sections 4.3.2, 4.3.3, 4.3.4.

4.3.2 Information Criterion Approach

The estimation for the measurement error model uses the maximum likelihood estimation method. Thus, we can utilize the information criterion to determine the optimal number of clusters K for ORCA using the general symbolic orthogonal regression method. In contrast, the simple symbolic orthogonal regression method use a symbolic PCA to estimate the coefficients of the linear regression model, which does not involve a likelihood function and so cannot apply the information criterion. We utilize the two commonly used information criteria, Akaike information criterion (AIC) (Akaike, 1973) and Bayesian information criterion (BIC) (Schwarz, 1978), to determine the optimal number of clusters.

For the general symbolic orthogonal regression method, the data are centralized before fitting a model. In other words, the model has the form of $(\boldsymbol{x} - \bar{\boldsymbol{X}})\boldsymbol{\beta} = 0$. For each

cluster, $k = 1, \ldots, K$, we need to estimate the $\boldsymbol{\beta}^{(k)}$ coefficients and $\bar{\boldsymbol{X}}^{(k)}$, which include a total of $2p - 1$ parameters. The log likelihood function for cluster k is given according to equation(4.21)

$$l^{(k)} = -\frac{n^{(k)}p}{2}\log(2\pi) - \frac{n^{(k)}}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{u}}^{(k)}| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i^{c(k)} - \ddot{\boldsymbol{x}}_i^{(k)})\boldsymbol{\Sigma}_{\boldsymbol{u}}^{(k)^{-1}}(\boldsymbol{x}_i^{c(k)} - \ddot{\boldsymbol{x}}_i^{(k)})^T, \quad (4.48)$$

where $n^{(k)}$ is the sample size of the k^{th} cluster such that $\sum_{k=1}^{K} n^{(k)} = n$, $\boldsymbol{\Sigma}_{\boldsymbol{u}}^{(k)}$ is the covariance matrix of measurement errors of the k^{th} cluster, and $\boldsymbol{x}_i^{c(k)}$ and $\ddot{\boldsymbol{x}}_i^{(k)}$ are the center points and the fitted values of the variables in the k^{th} clusters, respectively.

Note the number of parameters of the models for the K clusters is $m = K(2p - 1)$. The AIC and BIC critiera are, respectively,

$$AIC = -2\sum_{k=1}^{K} l^{(k)} + 2m, \quad (4.49)$$

$$BIC = -2\sum_{k=1}^{K} l^{(k)} + m\ln(n). \quad (4.50)$$

In equation (4.49) and equation (4.50) the penalty terms for the AIC and BIC are $2m$ and $m\ln(n)$, respectively. For both information criteria, the smaller the value is, the better the model fits. When the number of clusters K increases, the model has a better fit. However, while the term $-2\sum_{k=1}^{K} l^{(k)}$ in equations (4.49) and (4.50) decreases, the penalty terms for both criteria increase, which prevents the information criteria from decreasing steadily. For the information criteria, there is no theoretical obstacle to determining K even if the true optimal number of clusters is 1.

4.3.3 Gap Statistic

The Gap statistic was proposed by Tibshirani et al. (2001), and is a very flexible method to estimate the optimal number of clusters in a data set. The Gap statistic was originally proposed using a spherical clustering as an example, but there is no barrier to extending the method to a regression based clustering method. The Gap statistic calculates the log pooled within-cluster average distance to the cluster center, and compares that average distance with its expectation under a null reference distribution. We have given the definition of Gap statistics of Tibshirani et al. (2001) in equation (2.30) and equation(2.31). The idea is that the actual pooled within-cluster sum of squares of distances would decrease faster than its expected rate under the null distribution before the number of clusters K reaches its optimal value. The sum of squares of distances would decrease more slowly than its expected rate after K reaches its optimal value since unnecessary clusters are added. The Gap statistic is maximized when K is optimal. According to Tibshirani et al. (2001), a uniform distribution within ranges of the original data set is the best reference distribution.

For the ORCA algorithm, we are minimizing the orthogonal distances between the observations in an interval-valued data set and its fitted hyperplanes. The Gap statistic can be adapted to the ORCA by replacing the distance between an observation and its cluster center with the orthogonal distance between an observation and its fitted hyperplane. Suppose we partition the data set into K clusters, $P = (C_1, \ldots, C_K)$. The pooled within cluster sum of squares of orthogonal distances for the partition P is

$$W_K = \sum_{k=1}^{K} SSOD_k, \tag{4.51}$$

where $SSOD_k$ is the SSOD for the k^{th} cluster as defined in equation (4.45). The Gap statistic

is defined as follows:

$$\text{Gap}(K) = E[\log(W_K(B))] - \log(W_K), \tag{4.52}$$

where $E[\log(W_K(B))]$ is the expectation of W_K under the null reference distribution. In practice, $E[\log(W_K(B))]$ is obtained by randomly simulating B uniformly distributed samples and averaging the log pooled SSOD of these B samples, i.e.,

$$E[\log(W_K(B))] = \frac{1}{B}\sum_{b=1}^{B}\log(W_K(b)). \tag{4.53}$$

The reference samples are classical data samples generated from a uniform distribution over the ranges of the principal components of the data. The $W_k(b)$ is obtained by implementing a simple orthogonal regression clustering algorithm on the reference samples.

From Tibshirani et al. (2001), the optimal number of clusters K^* is the smallest K such that

$$\text{Gap}(K) \geq \text{Gap}(K+1) - s_{K+1}, \tag{4.54}$$

where $s_{K+1} = sd_{K+1}\sqrt{1+1/B}$, and sd_{K+1} is the standard deviation of SSOD of the B samples under the reference distribution given the number of clusters being $K+1$. Like the information criterion methods of section 4.3.2, the Gap statistic is defined for $K = 1$.

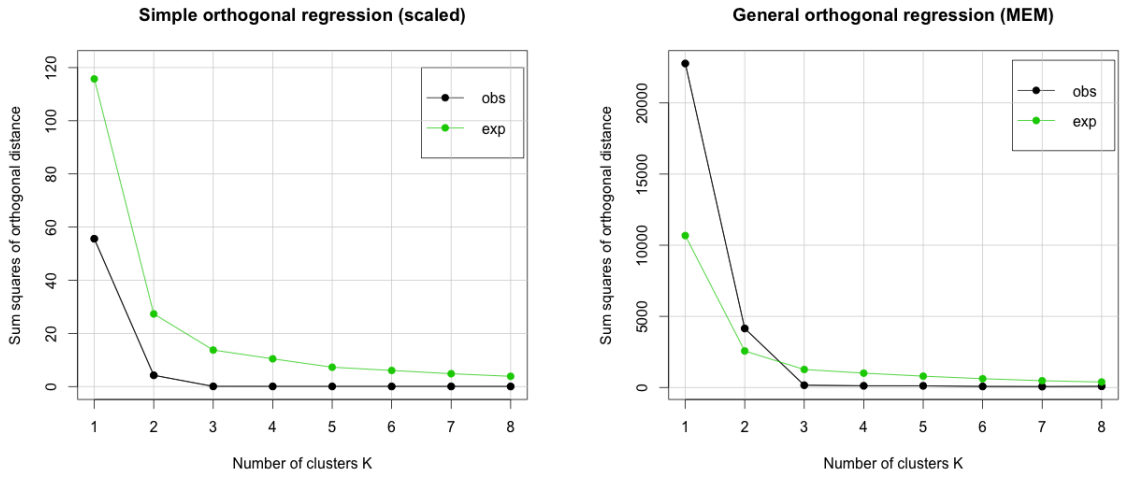
Table 4.2: Gap statistic for simple and general orthogonal regression clustering

Model	Statistic	K=1	2	3	4	5	6	7	8
Simple	SSOD	55.62	4.25	0.07	0.06	0.06	0.05	0.05	0.04
orthogonal	SSOD (ref)	115.75	27.32	13.73	10.44	7.28	6.06	4.83	3.86
regression	sd_K	0.07	0.07	0.15	0.13	0.17	0.13	0.13	0.16
	Gap	0.73	1.86	5.27	5.23	4.84	4.81	4.65	4.58
General	SSOD	22,772.30	4,148.51	159.03	117.52	115.49	74.08	64.70	82.12
orthogonal	SSOD (ref)	10,677.09	2,564.75	1,265.26	1,009.10	795.51	616.80	475.64	382.90
regression	sd_K	0.06	0.04	0.11	0.13	0.14	0.10	0.14	0.11
	Gap	-0.76	-0.48	2.07	2.14	1.92	2.12	1.99	1.53

Using the simulated data set generated by equation (4.46) as an example, the imple-

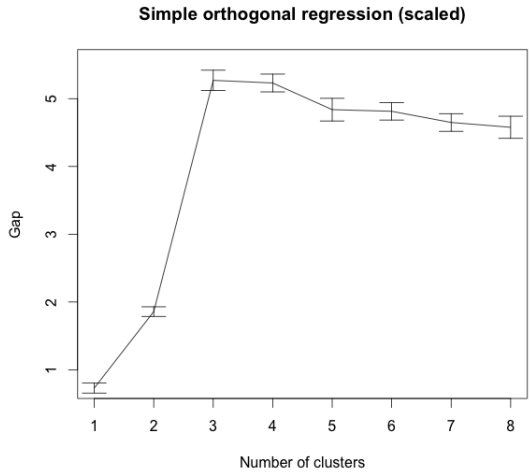
mentation results of the Gap statistics are shown in Table 4.2. Table 4.2 provides the Gap statistics by ORCA with the simple orthogonal regression using center distance and general symbolic orthogonal regression methods, respectively. For a particular orthogonal regression method, the row "SSOD" is the SSOD calculated based on the ORCA clustering results on the simulated interval-valued data set generated by equation (4.46). The row "SSOD(ref)" is the expected SSOD on the uniformly distributed reference data sets. The row "sd_K" is the standard deviation of the log SSOD(ref) calculated from 10 uniformly distributed reference data sets given the number of clusters being K. The row "Gap" shows the Gap statistics given the number of clusters being K. For the simple orthogonal regression method, we can see from Table 4.2 that $K = 3$ is the smallest number of K that satisfies the criterion equation (4.54): $\text{Gap}(3) = 5.27 \geq \text{Gap}(3+1) - s_{3+1} = 5.23 - 0.13\sqrt{1+1/10} = 5.09$. Thus, $K = 3$ is the optimal number of clusters for the simple orthogonal regression method. For the general orthogonal regression method, the optimal number of clusters is also $K = 3$ given that $K = 3$ is the smallest number that satisfies the equation (4.54): $\text{Gap}(3) = 2.07 \geq \text{Gap}(3+1) - s_{3+1} = 2.14 - 0.13\sqrt{1+1/10} = 2.00$.

Figure 4.3 (a) and (b) show the observed and expected SSOD values by the simple orthogonal regression method and the general orthogonal regression method, drawn by the black and the green lines, respectively. Figure 4.3 (c) and (d) show the Gap statistics with standard deviation bars for the simple regression method and the general orthogonal regression method, respectively. From the definition of Gap in equation (4.52) we can see that if the $W_K(B)$ decreases at a same rate as the W_K given the number of clusters K increases by one, then the value of the Gap statistic does not change. For example, when both $W_K(B)$ and W_K decrease by 10%, the Gap statistic is now $\text{Gap}(K+1) = E[\log(0.9W_K(B))] - \log(0.9W_K) = E[\log(W_K(B))] - \log(W_K) = \text{Gap}(K)$. When W_K decreases at faster rate than $W_K(B)$, the Gap statistic increases; otherwise, the Gap statistic decreases. We can see from Figure 4.3 (c) and (d) that the W_K decreases faster than its expectation on the

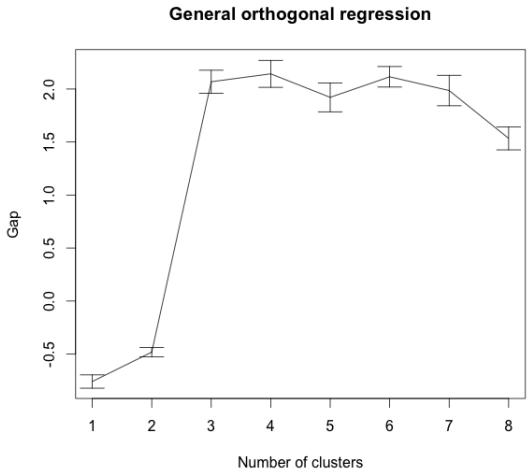


(a)

(b)



(c)



(d)

Figure 4.3: Gap statistic for the simple and general orthogonal regression clustering

reference distribution, $W_K(B)$, before K reaches its optimal value 3, given the fact that the Gap statistic keeps increasing before $K = 3$. The W_K decreases more slowly than does $W_K(B)$ after K reaches the optimal value.

4.3.4 Silhouette Statistic

Rousseeuw (1987) proposed a silhouette statistic to estimate the optimal number of clusters. The silhouette measures how strong an observation in a particular cluster is separated from its nearest cluster. For an observation $\boldsymbol{x}_i$, given a particular partition, let $a(i)$ be the average dissimilarity of $\boldsymbol{x}_i$ to all objects in its own cluster, and let $b(i)$ be the average dissimilarity of i to all the objects in its nearest cluster. The silhouette statistic, $s(i)$, is defined as follows:

$$s_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{4.55}$$

The range of s_i is from -1 to 1. When s_i is negative, the assignment of $\boldsymbol{x}_i$ is a misclassification. The closer the s_i value is to 1, the stronger is the indication that x_i belongs to its assigned cluster. A large value of the average silhouette over all the observations indicates a good separation of the clusters. The average silhouette statistic over the whole data set is maximized when the number of clusters K is optimized.

In spherical clusters, the dissimilarity is usually measured by the Euclidean distance. To adapt the silhouette statistic to the ORCA for interval-valued data, we revise the statistic as follows:

$$s_i = \frac{b(i) - a(i)}{b(i)}, \tag{4.56}$$

where $a(i)$ is the orthogonal distance between the observation $\boldsymbol{x}_i$ and the hyperplane to which it belongs, while $b(i)$ is the orthogonal distance between $\boldsymbol{x}_i$ and its nearest hyperplane fitted in the neighbor clusters. According to our algorithm, $b(i)$ would be always greater than $a(i)$; thus, the denominator in equation (4.55) would be always the $b(i)$ in equation

(4.56).

The optimal number of clusters is determined when the average silhouette statistic, $s = \sum_{i=1}^{n} s_i/n$, is maximized. This silhouette statistic describes how clearly the clusters are separated from each other. In addition, the average silhouette statistics on a particular cluster measure how strongly it is separated from other clusters.

4.4 Simulation Study

In this section, we use simulated data sets to test our algorithms. The simulation methods of the interval-valued data follow those of section 3.4 by treating the response variable y as one of the variables for the orthogonal regressions. We will first investigate the convergence and performance of the algorithm given the correct number of clusters. Then, we compare the different methods of determining the optimal number of clusters proposed in section 4.3.

4.4.1 Case Study

We study a total of six data sets with different structures and different dimensions to see if the ORCA algorithm is able to converge to the correct clusters given the correct number of clusters. In addition, we investigate what is an appropriate number of different initial partitions we need to try for a particular data set to converge to the correct clusters.

Our first example has the following structure:

$$\begin{aligned} \text{cluster } (1) &: x_2 = 8 + 1.3x_1 + \epsilon_1, \\ (2) &: x_2 = 45.5 + 2.8x_1 + \epsilon_2, \\ (3) &: x_2 = 65 - 2.5x_1 + \epsilon_3. \end{aligned} \tag{4.57}$$

We treat x_2 as response variable and x_1 as predictor variable and use the simulation method III in section 3.4 to generate data sets that satisfy the structure in equation (4.57). One

hundred interval-valued observations are simulated for each of the three clusters and then the observations from all the three clusters are stacked as one data set. Based on the simulation method III in section 3.4, the 100 interval center points of x_1 of the cluster 1, $x_{i1}^{(c)}$, $i = 1, \ldots, 100$, are independently generated from a normal distribution $N(4, 12)$. The 100 interval ranges of x_1 for the cluster 1, $x_{i1}^{(r)}$, $i = 1, \ldots, 100$, are generated from an exponential distribution $exp(1.5)$. The 100 interval values of x_1 are then obtained as $[x_{i1}^{(c)} - 0.5x_{i1}^{(r)}, x_{i1}^{(c)} + 0.5x_{i1}^{(r)}]$, $i = 1, \ldots, 100$. For each of $i = 1, \ldots, 100$, we randomly draw 5 values from a uniform distribution, $x_{i1l} \sim U(x_{i1a}, x_{i1b})$, $l = 1, \ldots, 5$. The interval value of $x_{i2} = [x_{i2a}, x_{i2b}]$ is obtained as

$$\begin{aligned} x_{i2a} &= \min_{l \in \{1,\ldots,5\}} \{8 + 1.3x_{i1l} + \epsilon_{il}\}, \\ x_{i2b} &= \min_{l \in \{1,\ldots,5\}} \{8 + 1.3x_{i1l} + \epsilon_{il}\}, \end{aligned} \tag{4.58}$$

where ϵ_{il}, $l = 1, \ldots, 5$, is the error term that follows a normal distribution $N(0, 7)$. The observations for the cluster 2 and 3 are drawn analogously as for the cluster 1. The interval center points of x_1 for the cluster 2 and 3 follow normal distributions $N(0, 11)$ and $N(8, 12)$, respectively. The interval ranges of x_1 for the cluster 2 and 3 are generated from exponential distributions $exp(1.3)$ and $exp(1.2)$, respectively. The error terms of the cluster 2 and 3 are generated from normal distributions $N(0, 7)$ and $N(0, 8)$, respectively.

We apply the ORCA using the simple orthogonal regression method with center distance, the simple orthogonal regression method with min max distance, and the general orthogonal regression method, to the data set generated by equation (4.57). The clustering results will be compared with the true clusters. Figure 4.4 (a) shows the plot of the simulated data with the three true clusters. Figure 4.4 (b), (c), and (d) show the clustering results by the simple orthogonal regression method (center), the simple orthogonal regression method (min max), and the general orthogonal regression method, respectively. For all the four figures,

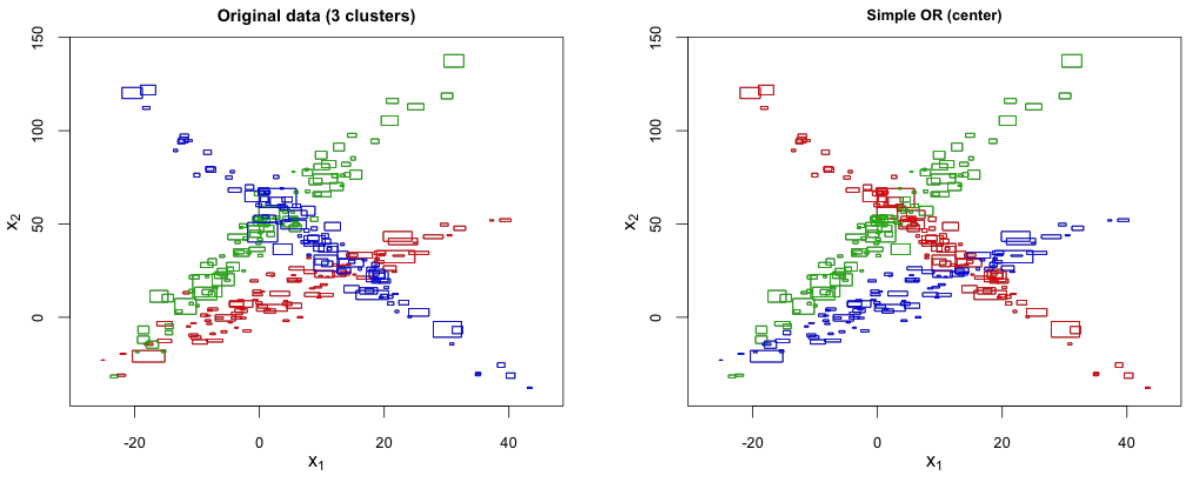
different colors differentiate the three clusters, but a particular color does not associate with a particular cluster. It is safe to say that the ORCA using all the three methods correctly recovers the true clusters by comparing the true clusters in Figure 4.4I (a) and the clustering results in Figure 4.4 (b), (c), and (d).

This example of equation (4.57) and Figure 4.4 are for demonstration purposes and verify that ORCA is able to converge to the correct clusters for this particular example. We conduct our simulation study on various examples with different data structures and a different number of clusters to look at the performance of the algorithm. In particular, we consider six examples for which a rough description for each example is shown below. For the purpose of focusing on the performance of ORCA algorithm, we defer the linear model equations for each example; the detailed parameter setups and the manner to read these setups are deferred to the Appendix in section 4.6. The simulation method for each example is analogous with the method for the example of equation (4.57).

I. Two-dimensional data with three clusters – The sample sizes for the three clusters are $n_1 = 100$, $n_2 = 50$, and $n_3 = 50$, respectively. The three clusters overlap with each other. Figure 4.5 (a) and (b) show the simulated three clusters and the ORCA clustering results using the general orthogonal regression method given the correct number of clusters $K = 3$.

II. Two-dimensional data with five clusters – The five clusters have equal sample size with $n_i = 100$, $i = 1, \ldots, 5$, in each cluster. Figure 4.6 (a) and (b) show the simulated data set with five clusters and ORCA clustering results on this data set using the general orthogonal regression methods given the correct number of clusters, $K = 5$.

III. Two-dimensional data with three clusters – The sample sizes for the three clusters are $n_1 = 100$, $n_2 = 50$, $n_3 = 25$, respectively. Two of the three clusters overlap a lot with each other. The third cluster has a relatively small sample size. Figure 4.7 (a) and



(a) (b)

(c) (d)

Figure 4.4: Clustering results of example of equation (4.57)

(b) give the simulated data set and the ORCA solution using the general orthogonal regression method given $K = 3$.

IV. Two-dimensional data with two clusters with equal sample sizes – The two clusters have equal sample sizes with $n_i = 50$, $i = 1, 2$. The two clusters are not well separated. The simulated data set and the ORCA clustering results using the general orthogonal regression method given $K = 2$ are shown in Figure 4.8 (a) and (b).

V. Three clusters in a three-dimensional data – The three clusters are equal sample sized with $n_i = 40$, $i = 1, 2, 3$. Visualization is difficult for this 3-dimensional data, but a relatively small number of mis-clustered observations indicates the ORCA solution converges to the correct clusters.

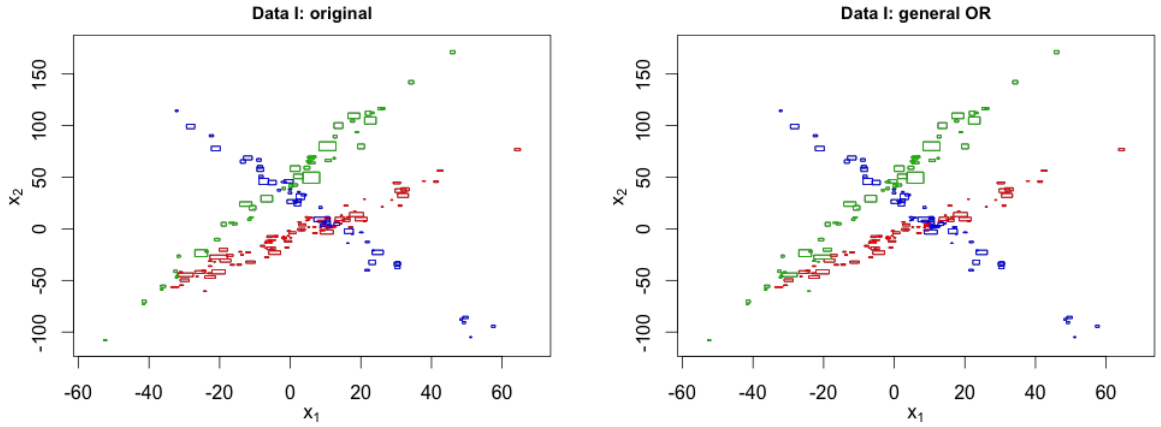
VI. Two clusters in a five-dimensional data – The sample sizes for the two clusters are equal $n_i = 50$, $i = 1, 2$. Again, a relatively small number of mis-clustered observations indicates correct convergence by the ORCA solution.

The examples V and VI are multi-dimensional data where it is hard to make plots for them to show visually the clusters and the linear relationship between variables. To verify that the ORCA is clustering to the correct clusters, we compare the ORCA clustering results with the true clusters. When a cluster obtained by the ORCA mostly overlaps with a true cluster, we say this true cluster is correctly recovered. If an observation in a true cluster is clustered by ORCA into different clusters, then this observation is mis-clustered. When the clusters in a data set partially overlap, we expect there are some observations that will be mis-clustered by the ORCA. However, most of the non-overlapped observations will be clustered correctly by ORCA. A low number of mis-clustered observations by ORCA for a particular data set is an indication that the algorithm successfully recovers the true clusters.

We compare the simulated data set and the ORCA clustering results by the general orthogonal regression method given the correct number of clusters in Figure 4.5, Figure 4.6,

Figure 4.7, and Figure 4.8 for example data sets I, II, III, and IV, respectively. It is safe to say that the ORCA using the general orthogonal regression method correctly recovers the true clusters for each of the example data sets.

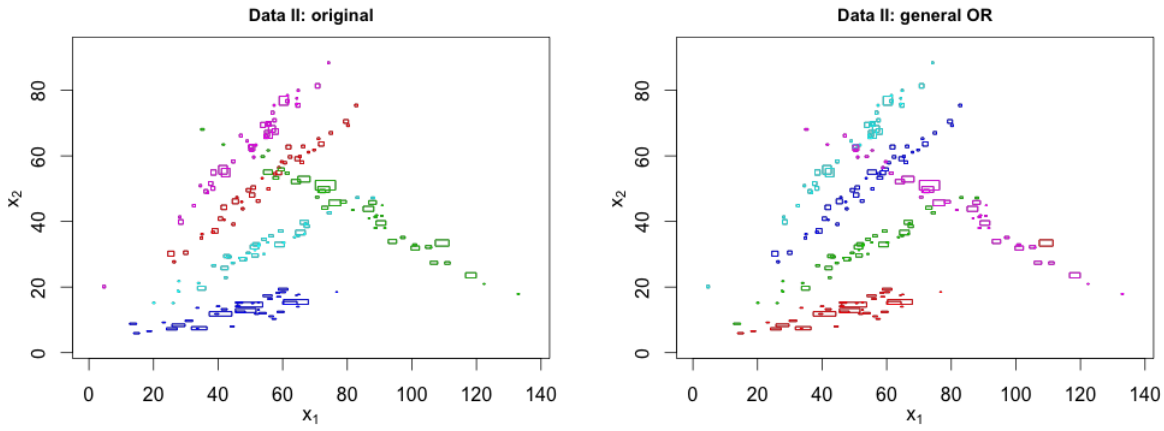
Note that the ORCA solutions in Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8 are all based on the general orthogonal regression method just for illustrative purposes. The ORCA solutions applying the simple orthogonal regression method with the center distance and the min max distance correctly recover the clusters for the example data sets I, II, III, and IV. (see Appendix 4.6.2).



(a) (b)

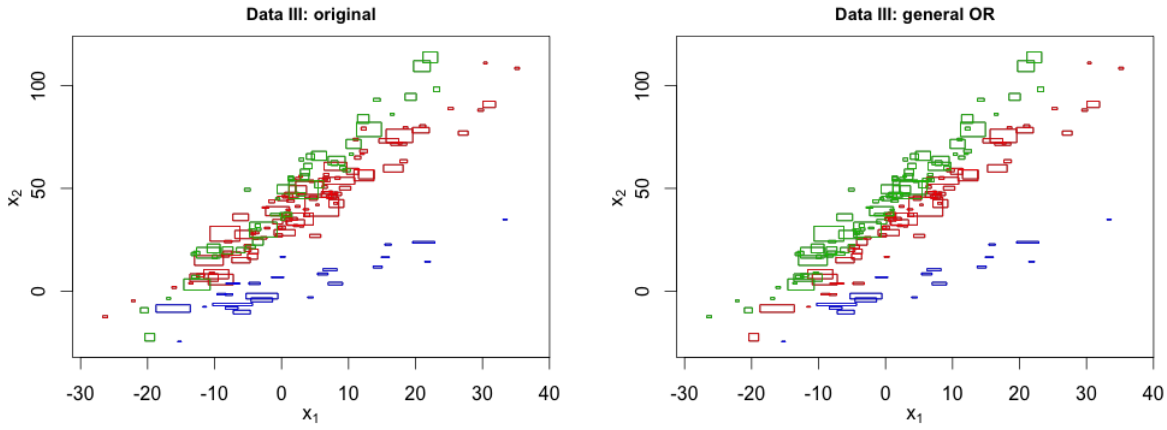
Figure 4.5: Example data set I and its ORCA clustering results

For the data set examples IV and V, we look at the number of mis-clustered observations in the clustering results, which are shown in Table 4.3 and Table 4.4. Table 4.3 and Table 4.4 are essentially frequency tables. The rows correspond to the true clusters in the simulated data sets, while the columns are the clusters obtained by the ORCA using the simple orthogonal regression method (center), the simple orthogonal regression method (min max), and the general orthogonal regression method. For a particular cell of the tables, if the row cluster does not agree with the column cluster, the cell is the number of mis-clustered observations.



(a) (b)

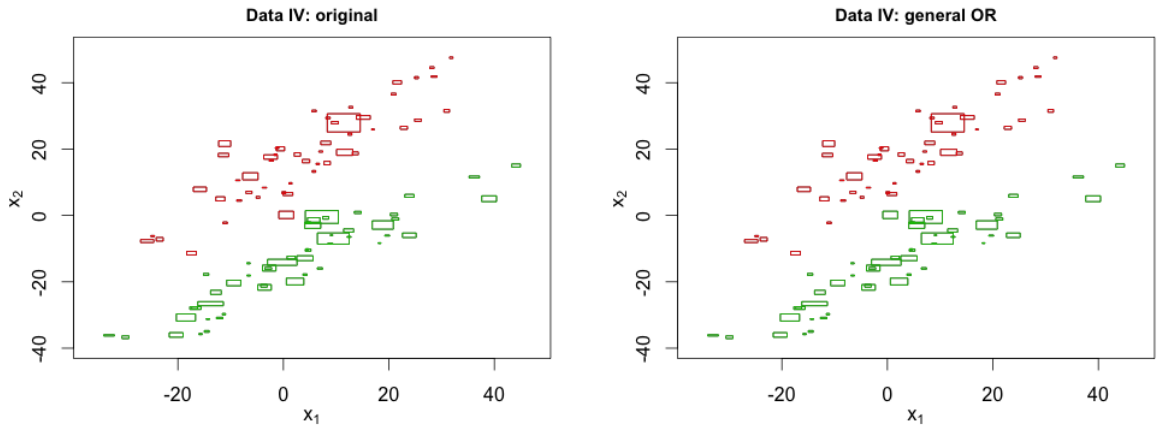
Figure 4.6: Example data set II and its ORCA clustering results



(a) (b)

Figure 4.7: Example data set III and its ORCA clustering results

For example, in Table 4.3, the value in the cell of row "cluster 1" and column "1" under tab "Simple OR (center)" is 40 that indicates all the 40 observations of cluster 1 in the simulated data set are correctly clustered as cluster 1 by the ORCA using the simple orthogonal regression method. The value in the cell of row "cluster 1" and column "3" under the tab "General OR" is 4. This implies that 4 of 40 observations in the cluster 1 of the simulated data set are clustered as cluster 3 by the ORCA using the general orthogonal



(a) (b)

Figure 4.8: Example data set IV and its ORCA clustering results

regression method. These four observations are mis-clustered. Table 4.4 and the other cells of the Table 4.3 can be interpreted in a similar manner. In general, the numbers of mis-clustered observations by ORCA for the data set examples IV and V are relatively small. It is safe to say that the ORCA correctly recovers the clusters for these two data sets.

Table 4.3: Number of mis-clustered observations for data set example V

		ORCA clustering results								
		Simple OR (center)			Simple OR (min max)			General OR		
		1	2	3	1	2	3	1	2	3
Simulated	cluster 1	40	0	0	40	0	0	35	1	4
cluseters	cluster 2	0	39	1	0	40	0	0	40	0
	cluster 3	0	0	40	0	0	40	0	0	40

For each of the above six simulated data sets, we want to understand what is the appropriate number of initial partitions needed for a good convergence. Given the correct number of clusters for each of the example data sets, we tried 1000 different random initial partitions for the ORCA using the simple orthogonal regression method (center), the simple orthogonal regression method (min max), and the general orthogonal regression method, respectively.

Table 4.4: Number of mis-clustered observations for data set example VI

		ORCA clustering results					
		Simple OR (center)		Simple OR (min max)		General OR	
		1	2	1	2	1	2
Simulated	cluster 1	50	0	50	0	50	0
cluseters	cluster 2	4	46	8	42	4	46

A random initial partition is generated by step (*ii'*) in section 4.2.

In Table 4.5, the percentages in columns "Simple OR (center)", "Simple OR (min max)", and "General OR" are the percentage of a good convergence by ORCA when trying one thousand different random initial partitions using the simple orthogonal regression method (center), the simple orthogonal regression method (min max), and the general orthogonal regression method, respectively. Here a correct convergence means that the ORCA converges to the correct clusters when starting with a particular initial partition. For example, 85.9% of the 1000 random initial partitions converge correctly by ORCA using the simple orthogonal regression (center). The column "Suggested Number" is the suggested number of initial partitions needed to obtain at least one partition that will converge to the correct clusters with 95% probability. As discussed in section 4.2, the suggest number of initial partitions is calculated as

$$\text{Suggested Number} = \frac{\log(0.5)}{\log(1-p)}, \tag{4.59}$$

where the probability p is obtained by the equation (4.44). If we try only the suggested number of different initial partitions, we want to know how many times the ORCA is able to converge correctly based on the simulated good convergence percentage out of the 1000 random initial partitions. Take Data I as an example; the suggested number of different initial partitions is 59. The expected good convergence rate from column "Simple OR (center)" is 85.9%. Then, the expected number of good convergences when applying ORCA using

the simple orthogonal regression method (center) on Data I is $59 \times 85.9\% = 50.4$. Given that we tried only the suggested number of different initial partitions, we can calculate the expected numbers of good convergences of ORCA on all Data I - VI. These are given by columns "Expected simple OR (center)", "Expected simple OR (min max)", and "Expected general OR" in Table 4.5. We can see that when trying the suggested number of different initial partitions, the ORCA can converge to the correct clusters multiple times for all Data I-VI. This indicates that to use the suggested number of initial partitions for the ORCA is a relatively convenient and safe choice to obtain the correct clusters.

Table 4.5: Number of initial partitions for good convergence

	Simple OR (center)	Simple OR (min max)	General OR	Suggested number	Expected simple OR (center)	Expected simple OR (min max)	Expected general OR
Data I (d=2, K=3)	85.9%	83.0%	72.9%	59	50.4	48.7	42.8
Data II (d=2, K=5)	9.1%	8.7%	2.7%	627	57.1	54.6	16.9
Data III (d=2, K=3)	20.0%	25.2%	11.9%	139	27.7	35.0	16.5
Data IV (d=2, K=2)	43.7%	43.3%	75.3%	8	3.6	3.6	6.2
Data V (d=3, K=3)	27.1%	23.7%	5.3%	49	13.3	11.6	2.6
Data VI (d=5, K=2)	89.8%	76.2%	75.7%	13	11.2	9.5	9.5

From Table 4.5, we can see that the ORCA for the general orthogonal regression method usually needs a higher number of initial partitions to obtain good convergence than do the two simple orthogonal regression methods. One of the reasons is that the general orthogonal regression method requires a more rigid assumption, equation (4.18), which constrains the measurement error to be within the interval ranges for each variable. This assumption can be violated sometimes. Furthermore, the fit of the general orthogonal regression method requires the information of the covariance structure of the measurement errors. To obtain convergence to the correct clusters for the general orthogonal regression method, not only a good initial partition is needed, but a good covariance structure that is close to the true covariance structure of the measurement errors is also needed.

4.4.2 Comparison of Different Methods to Determine the Optimal Number of Clusters

In section 4.3, we discussed several methods to determine the optimal number of clusters for the ORCA algorithm. In this section, we conduct a simulation study for various data structures and investigate the performance of the different methods. The same six data sets developed in section 4.4.1 will be considered for the simulation study in this section. Each method we proposed to determine the optimal number of clusters will be applied to the six data sets. Instead of visualizing the plot, the elbow method is implemented by evaluating the SD_K, percentage of change on SSOD, defined in equation (4.47). The optimal number of clusters is chosen as the smallest K such that SD_K is less than a predetermined constant c. In other words, when the number of clusters reaches its optimal value, the percentage of decrement of SSOD is less than c if a further cluster is added. The predetermined constant c is set to be 10%, 25%, and 50%, respectively, in our simulation study. Note that by such a rule on SD_K, the elbow method is defined for a single cluster. The Gap statistic described in section 4.3.3 and the information criterion approaches described in 4.3.2 are defined for a single cluster as well. The silhouette statistic is not defined for a single cluster. The information criterion approaches do not apply to the simple orthogonal regression methods since the likelihood function is not defined for these methods.

For a particular example of Data I-VI, we simulate a random sample based on the parameter setup in the Appendix 4.6. For each of $K = 1, \ldots, 8$, we tried the suggested number of initial partitions to apply the ORCA onto this random sample. The clustering results with smallest SSOD defined in equation (4.45) is set to be the correct cluster results given each of $K = 1, \ldots, 8$. We collect the information for the Gap statistics, the silhouette statistics, SSOD, AIC, and BIC of the correct clustering results for each of $K = 1, \ldots, 8$. The collected information is then used to determine the optimal number of clusters by each

of the proposed methods in section 4.3. We repeat this whole process 50 times to study the performance of each method to determine the optimal number of clusters.

The simulation results for the six data sets are summarized in the following tables. Table 4.6 shows the simulation results of the estimated optimal number of clusters by ORCA when applying the general orthogonal regression method. The rows of the table correspond to the different methods that were used to determine the optimal number of clusters for each of Data I-VI. The methods we used to estimate the optimal number of clusters are the Gap statistic defined in equation (4.54), the silhouette statistic defined in equation (4.56), the two information criterion approaches AIC (see equation (4.49))and BIC (see equation (4.50)), and the elbow method by evaluating SD_K (see equation (4.47)) using three different cutoff values, 10%, 25%, and 50%. The maximum number of the optimal number of clusters is set to be eight. The columns labeled $1, \ldots, 8$ are the estimated optimal number of clusters by these methods.

Table 4.6 is essentially a frequency table, e.g., for Data I the Gap statistics estimate 3 as the optimal number of clusters for 24 times out of the 50 replications, 4 as the optimal number of clusters for 25 times, and 5 as the optimal number of clusters for 1 time. The simulated Data I has three clusters; thus, the Gap statistics correctly estimate the optimal number of clusters 24 times out of the 50 replications. For each example data set, we mark the column with true number of clusters by † (*dagger*). For instance, the column $K = 3$ is marked by *dagger* since the true number of clusters of Data I is 3. We use the SD_K with cutoff values 10%, 25%, and 50% to implement the elbow method. For some cases, the SD_K is never smaller than the cutoff values; then, the number of optimal clusters estimated by the elbow method under such scenarios is recorded in the column "NA". For example, 40 times out of the 50 replications the SD_K for all of $K = 1, \ldots, 8$ are greater than cutoff value 10%. Thus, the last column of row "$SD_K(10\%)$" for Data I is 40. The other cells of Table 4.6 can be interpreted in a similar way as we explained for the Data I.

From Table 4.6, we can see that the silhouette statistics work consistently well to estimate the correct number of clusters except for the Data III. The silhouette statistics correctly estimated the optimal number of clusters for all of the 50 repetitions for Data I and VI, while it estimates the correct number of clusters 41, 48, and 47 times out of the 50 repetitions for Data II, IV, and V, respectively. For Data III, from Figure 4.7, we can see two of the three clusters are very close and a large proportion of their observations are overlapped. Thus, it is not surprising that the silhouette statistic determines the optimal number of clusters for the Data II to be two over all the 50 repetitions. The elbow method $SD_K(50\%)$ works fairly well to estimate the correct number of clusters for all the six data sets. It correctly estimates the number of clusters as 50, 32, 43, 36, 47, and 42 times out of the 50 replications for Data I-VI, respectively. The numbers of correct determinations out of the 50 repetition by $SD_K(50\%)$ are generally lower than those for the silhouette statistics. However, the elbow method $SD_K(50\%)$ correctly estimates the number of clusters for Data III 43 out of 50 times.

The elbow method $SD_K(50\%)$ is a more robust method to estimate the optimal number of clusters. The elbow method with criteria $SD_K(10\%)$ and $SD_K(25\%)$ mostly fails to estimate the correct number of clusters, which indicates that the thresholds 10% and 25% are too small most of the time. Note that the threshold of SD_K is predetermined subjectively, which can depend on the scale and structure of the data. In practice, an elbow plot can be helpful to make the decision about the optimal number of clusters. The information approach AIC only works for Data VI given that it correctly estimates the number of clusters as 47 out of the 50 replications. The BIC works well for Data V and VI given that it correctly estimates the number of clusters as 41 and 52 times out of the 50 replications.

The Gap statistic fails to estimate the correct number of clusters most of the time given that it correctly estimates the number of clusters only 24, 1, 1, 15, 21, and 50 times out of the 50 replications. The Gap statistic requires that the SSOD of the simulated data decreases

Table 4.6: Distribution of the estimated number of clusters by ORCA (general orthogonal regression method)

Method	Estimated number of clusters (K)								
	1	2	3	4	5	6	7	8	NA
Data I (K=3, p=2)									
Gap	0	0	24†	25	1	0	0	0	0
Silhouette	0	0	50†	0	0	0	0	0	0
AIC	0	0	0†	0	0	0	5	45	0
BIC	0	0	0†	0	0	0	16	34	0
$SD_K(10\%)$	0	0	0†	0	5	3	2	0	40
$SD_K(25\%)$	0	0	0†	15	31	4	0	0	0
$SD_K(50\%)$	0	0	50†	0	0	0	0	0	0
Data II (K=5, p=2)									
Gap	18	13	6	12	1†	0	0	0	0
Silhouette	0	0	0	0	41†	9	0	0	0
AIC	0	0	0	0	0†	0	1	49	0
BIC	0	0	0	0	0†	0	4	46	0
$SD_K(10\%)$	0	0	0	0	14†	15	3	0	18
$SD_K(25\%)$	0	0	0	0	33†	14	2	0	1
$SD_K(50\%)$	0	0	17	1	32†	0	0	0	0
Data III (K=3, p=2)									
Gap	35	14	1†	0	0	0	0	0	0
Silhouette	0	50	0†	0	0	0	0	0	0
AIC	0	0	0†	0	0	0	1	49	0
BIC	0	0	0†	0	0	0	3	47	0
$SD_K(10\%)$	0	0	0†	0	1	0	3	0	46
$SD_K(25\%)$	0	0	0†	4	18	18	10	0	0
$SD_K(50\%)$	0	7	43†	0	0	0	0	0	0
Data IV (K=2, p=2)									
Gap	35	15†	0	0	0	0	0	0	0
Silhouette	0	48†	2	0	0	0	0	0	0
AIC	0	0†	0	0	0	0	1	49	0
BIC	0	0†	0	0	0	0	3	47	0
$SD_K(10\%)$	0	0†	0	0	0	0	1	0	49
$SD_K(25\%)$	0	0†	0	3	17	19	5	0	6
$SD_K(50\%)$	0	36†	14	0	0	0	0	0	0
Data V (K=3, p=3)									
Gap	23	0	21†	6	0	0	0	0	0
Silhouette	0	1	47†	2	0	0	0	0	0
AIC	0	0	6†	29	11	4	0	0	0
BIC	0	0	41†	9	0	0	0	0	0
$SD_K(10\%)$	0	0	0†	9	15	15	5	0	6
$SD_K(25\%)$	0	0	3†	29	14	2	0	0	2
$SD_K(50\%)$	0	0	47†	3	0	0	0	0	0
Data VI (K=2, p=5)									
Gap	0	50†	0	0	0	0	0	0	0
Silhouette	0	50†	0	0	0	0	0	0	0
AIC	0	47†	3	0	0	0	0	0	0
BIC	0	50†	0	0	0	0	0	0	0
$SD_K(10\%)$	0	0†	1	5	9	13	9	0	13
$SD_K(25\%)$	0	0†	14	23	10	3	0	0	0
$SD_K(50\%)$	0	42†	8	0	0	0	0	0	0

† column corresponds to the true number of clusters.

Table 4.7: Distribution of the estimated number of clusters by ORCA (simple orthogonal regression method - center distance)

Method	Estimated number of clusters (K)								
	1	2	3	4	5	6	7	8	NA
Data I (K=3, p=2)									
Gap	0	0	50†	0	0	0	0	0	0
Silhouette	0	5	44†	1	0	0	0	0	0
$SD_K(10\%)$	0	0	0†	0	2	9	9	0	30
$SD_K(25\%)$	0	0	0†	9	25	15	1	0	0
$SD_K(50\%)$	0	0	49†	1	0	0	0	0	0
Data II (K=5, p=2)									
Gap	17	12	0	0	18†	3	0	0	0
Silhouette	0	0	0	10	38†	1	1	0	0
$SD_K(10\%)$	0	0	0	0	16†	13	9	0	12
$SD_K(25\%)$	0	0	0	0	41†	9	0	0	0
$SD_K(50\%)$	0	0	0	1	47†	2	0	0	0
Data III (K=3, p=2)									
Gap	0	36	12†	2	0	0	0	0	0
Silhouette	0	50	0†	0	0	0	0	0	0
$SD_K(10\%)$	0	0	0†	0	8	10	10	0	22
$SD_K(25\%)$	0	0	0†	20	25	4	1	0	0
$SD_K(50\%)$	0	0	50†	0	0	0	0	0	0
Data IV (K=2, p=2)									
Gap	0	50†	0	0	0	0	0	0	0
Silhouette	0	50†	0	0	0	0	0	0	0
$SD_K(10\%)$	0	0†	0	0	2	5	6	0	37
$SD_K(25\%)$	0	0†	0	9	19	12	7	0	3
$SD_K(50\%)$	0	49†	1	0	0	0	0	0	0
Data V (K=3, p=3)									
Gap	9	0	39†	2	0	0	0	0	0
Silhouette	0	0	50†	0	0	0	0	0	0
$SD_K(10\%)$	0	0	0†	7	14	14	9	0	6
$SD_K(25\%)$	0	0	11†	33	6	0	0	0	0
$SD_K(50\%)$	0	0	50†	0	0	0	0	0	0
Data VI (K=2, p=5)									
Gap	0	50†	0	0	0	0	0	0	0
Silhouette	0	50†	0	0	0	0	0	0	0
$SD_K(10\%)$	0	0†	1	6	16	17	6	0	4
$SD_K(25\%)$	0	0†	15	25	8	2	0	0	0
$SD_K(50\%)$	0	49†	1	0	0	0	0	0	0

† column corresponds to the true number of clusters.

Table 4.8: Distribution of the estimated number of clusters by ORCA (simple orthogonal regression method - min max distance)

Method	Estimated number of clusters (K)								
	1	2	3	4	5	6	7	8	NA
Data I (K=3, p=2)									
Gap	0	0	50†	0	0	0	0	0	0
Silhouette	0	0	50†	0	0	0	0	0	0
$SD_K(10\%)$	0	0	0†	0	2	13	8	0	27
$SD_K(25\%)$	0	0	0†	50	0	0	0	0	0
$SD_K(50\%)$	0	0	50†	0	0	0	0	0	0
Data II (K=5, p=2)									
Gap	11	5	0	0	34†	0	0	0	0
Silhouette	0	0	2	14	34†	0	0	0	0
$SD_K(10\%)$	0	0	0	0	2†	18	23	0	7
$SD_K(25\%)$	0	0	0	0	45†	5	0	0	0
$SD_K(50\%)$	0	0	0	0	50†	0	0	0	0
Data III (K=3, p=2)									
Gap	0	47	3†	0	0	0	0	0	0
Silhouette	0	50	0†	0	0	0	0	0	0
$SD_K(10\%)$	0	0	0†	0	10	15	11	0	14
$SD_K(25\%)$	0	0	0†	18	28	4	0	0	0
$SD_K(50\%)$	0	0	50†	0	0	0	0	0	0
Data IV (K=2, p=2)									
Gap	0	50†	0	0	0	0	0	0	0
Silhouette	0	50†	0	0	0	0	0	0	0
$SD_K(10\%)$	0	0†	0	0	5	5	5		35
$SD_K(25\%)$	0	0†	1	9	23	15	2	0	0
$SD_K(50\%)$	0	46†	4	0	0	0	0	0	0
Data V (K=3, p=3)									
Gap	13	0	37†	0	0	0	0	0	0
Silhouette	0	1	49†	0	0	0	0	0	0
$SD_K(10\%)$	0	0	47†	3	0	0	0	0	0
$SD_K(25\%)$	0	0	50†	0	0	0	0	0	0
$SD_K(50\%)$	0	0	50†	0	0	0	0	0	0
Data VI (K=2, p=5)									
Gap	2	48†	0	0	0	0	0	0	0
Silhouette	0	50†	0	0	0	0	0	0	0
$SD_K(10\%)$	0	4†	40	6	0	0	0	0	0
$SD_K(25\%)$	0	50†	0	0	0	0	0	0	0
$SD_K(50\%)$	0	50†	0	0	0	0	0	0	0

† column corresponds to the true number of clusters.

consistently faster than the reference distribution before the number of clusters reaches its optimal value. However, since the general orthogonal distance involves the covariance matrices of measurement errors, the decrement of SSOD can be similar or slower than it is for the reference distribution before the number of clusters reaches its optimal value. This is why the Gap statistic keeps underestimating the optimal number of clusters.

Generally, the silhouette statistics work well to determine the optimal number of clusters for ORCA using the general orthogonal regression method except for some difficult situation such as Data III. The elbow method is generally a good method to estimate the optimal number of clusters. The Gap statistic and information approaches mostly fail for ORCA using the general orthogonal regression method.

Table 4.7 shows the simulation results of ORCA when applying the simple orthogonal regression method with center distance. The elbow method $SD_K(50\%)$ outperforms the other methods given that it correctly estimates the optimal number of clusters as 49, 47, 50, 49, 50, and 49 times out of the 50 replications for Data I-VI, respectively. The silhouette statistic correctly estimates the optimal number of clusters most of the time except for Data III. From Table 4.7 we can see that the silhouette statistic correctly estimates the number of clusters as 44, 38, 50, 50, 50 times out of the 50 replications for Data I, II, IV, V, and VI, respectively, but 0 times for Data III. As we stated earlier, a large proportion of the two clusters for Data III overlaps. This is the reason that the silhouette statistic estimates the optimal number for clusters for Data III as two instead of 3, the true number of clusters, for all the 50 replications. Since the SSOD of the simple orthogonal regression method is comparable to that for the reference distribution, the Gap statistic works better for the simple orthogonal regression method (center) than does the general orthogonal regression method. The thresholds, 10% and 25%, for the elbow method are too small to estimate the correct number of clusters for all the data sets.

The simulation results of ORCA when applyed to the simple regression method with the

min max distance are presented in Table 4.8. We can see that the elbow method $SD_K(50\%)$ outperforms all other methods given that it correctly estimates the number of clusters as 50, 50, 50, 46, 50, 50 times out of the 50 replications for Data I-VI, respectively. The Gap statistic and the silhouette statistic correctly estimate the number of clusters most of the time except for the Data III. The numbers of correct determinations for the Gap statistic and the silhouette statistic are mostly smaller than that number for the SD_K (50%), e.g., the number of correct determinations by the Gap statistic for Data II is 34, smaller than the number of correct determinations by SD_K (50%) of 50. The performances of the Gap statistic and the silhouette statistic are similar considering that the numbers of correct determinations between the two statistics are close for all the six data sets. The thresholds, 10% and 25%, for the elbow method are mostly too small to estimate the correct number of clusters for the six data sets .

In summary, the elbow method with threshold 50% for SD_K is the best method to estimate the optimal number of clusters, but an appropriate threshold can be different for different data sets. An elbow plot can be helpful to choose an appropriate threshold for the SD_KS. The silhouette statistic performs well except for some difficult situations such as when there is a large proportion of overlaps between clusters as in Data III. The Gap statistic does not works consistently well especially for the general orthogonal regression method. The information approaches can only be applied to the general orthogonal regression method, but the performance is generally bad and not reliable. In our application in section 4.5, we will implement all the methods to determine the optimal number of clusters, but will rely on the elbow method and the silhouette statistics.

4.5 Application

In this section, the effectiveness of clustering by the ORCA algorithm is demonstrated on the iris data (Fisher, 1936). The Iris data have 150 observations consisting of three different species of Iris flowers: Setosa, Versicolor, and Virginica. Each species includes 50 observations where four attributes are recorded for each observation, sepal width, sepal length, petal width, and petal length, respectively. The two widths and two lengths are measured in centimeter and the measurements are shown in Table 4.9 where SepalL is the sepal length, SepalW is the sepal width, PetalL is the petal length, and PetalW is the petal width. Figure 4.9 gives a scatter plot matrix between the four attributes of 150 observations in the Iris data. In the numerical representation, two of the three species (Versicolor and Verginica) have substantial overlap, while the third species (Setosa) is relatively well separated from the other two. We can see that the observations from each species are not spherical. The Iris data have been a standard benchmark to test the effectiveness of a clustering algorithm since it was published in Anderson (1935) and Fisher (1936). Note that one can argue that both $K = 2$ or 3 could be the optimal number of clusters due to the large overlap between Iris Versicolor and Iris Virginica (Pal and Bezdek, 1997).

The Iris data are classical data but we will consider them from a symbolic data perspective and apply our ORCA algorithm to cluster the data. Sepal is part of the calyx of a flower, typically forming a whorl that encloses the petals and forms a protective layer around a flower in bud. The width and length of a sepal can be seen as the smallest and largest distance between any two points on the edge of the sepal. More precisely, the sepal width and sepal length are the smallest and the largest distances between any two points that are at the edge of the sepal and the line connecting the two points crosses the center of mass of the sepal. Similarly, the petal width and petal length are the smallest and the largest distances between any two points that are at the edge of the petal and the line connecting

Table 4.9: Iris data (Fisher, 1936)

	Setosa				Versicolor				Virginica			
N	SepalL	SepalW	PetalL	PetalW	SepalL	SepalW	PetalL	PetalW	SepalL	SepalW	PetalL	PetalW
1	5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
2	4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
3	4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4	4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5	5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
6	5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
7	4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
8	5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
9	4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
10	4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
11	5.4	3.7	1.5	0.2	5.0	2.0	3.5	1.0	6.5	3.2	5.1	2.0
12	4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
13	4.8	3.0	1.4	0.1	6.0	2.2	4.0	1.0	6.8	3.0	5.5	2.1
14	4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
15	5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
16	5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
17	5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
18	5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
19	5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
20	5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6.0	2.2	5.0	1.5
21	5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
22	5.1	3.7	1.5	0.4	6.1	2.8	4.0	1.3	5.6	2.8	4.9	2.0
23	4.6	3.6	1.0	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2.0
24	5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
25	4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
26	5.0	3.0	1.6	0.2	6.6	3.0	4.4	1.4	7.2	3.2	6.0	1.8
27	5.0	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
28	5.2	3.5	1.5	0.2	6.7	3.0	5.0	1.7	6.1	3.0	4.9	1.8
29	5.2	3.4	1.4	0.2	6.0	2.9	4.5	1.5	6.4	2.8	5.6	2.1
30	4.7	3.2	1.6	0.2	5.7	2.6	3.5	1.0	7.2	3.0	5.8	1.6
31	4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
32	5.4	3.4	1.5	0.4	5.5	2.4	3.7	1.0	7.9	3.8	6.4	2.0
33	5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
34	5.5	4.2	1.4	0.2	6.0	2.7	5.1	1.6	6.3	2.8	5.1	1.5
35	4.9	3.1	1.5	0.2	5.4	3.0	4.5	1.5	6.1	2.6	5.6	1.4
36	5.0	3.2	1.2	0.2	6.0	3.4	4.5	1.6	7.7	3.0	6.1	2.3
37	5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
38	4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
39	4.4	3.0	1.3	0.2	5.6	3.0	4.1	1.3	6.0	3.0	4.8	1.8
40	5.1	3.4	1.5	0.2	5.5	2.5	4.0	1.3	6.9	3.1	5.4	2.1
41	5.0	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
42	4.5	2.3	1.3	0.3	6.1	3.0	4.6	1.4	6.9	3.1	5.1	2.3
43	4.4	3.2	1.3	0.2	5.8	2.6	4.0	1.2	5.8	2.7	5.1	1.9
44	5.0	3.5	1.6	0.6	5.0	2.3	3.3	1.0	6.8	3.2	5.9	2.3
45	5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
46	4.8	3.0	1.4	0.3	5.7	3.0	4.2	1.2	6.7	3.0	5.2	2.3
47	5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5.0	1.9
48	4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3.0	5.2	2.0
49	5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	6.2	3.4	5.4	2.3
50	5.0	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3.0	5.1	1.8

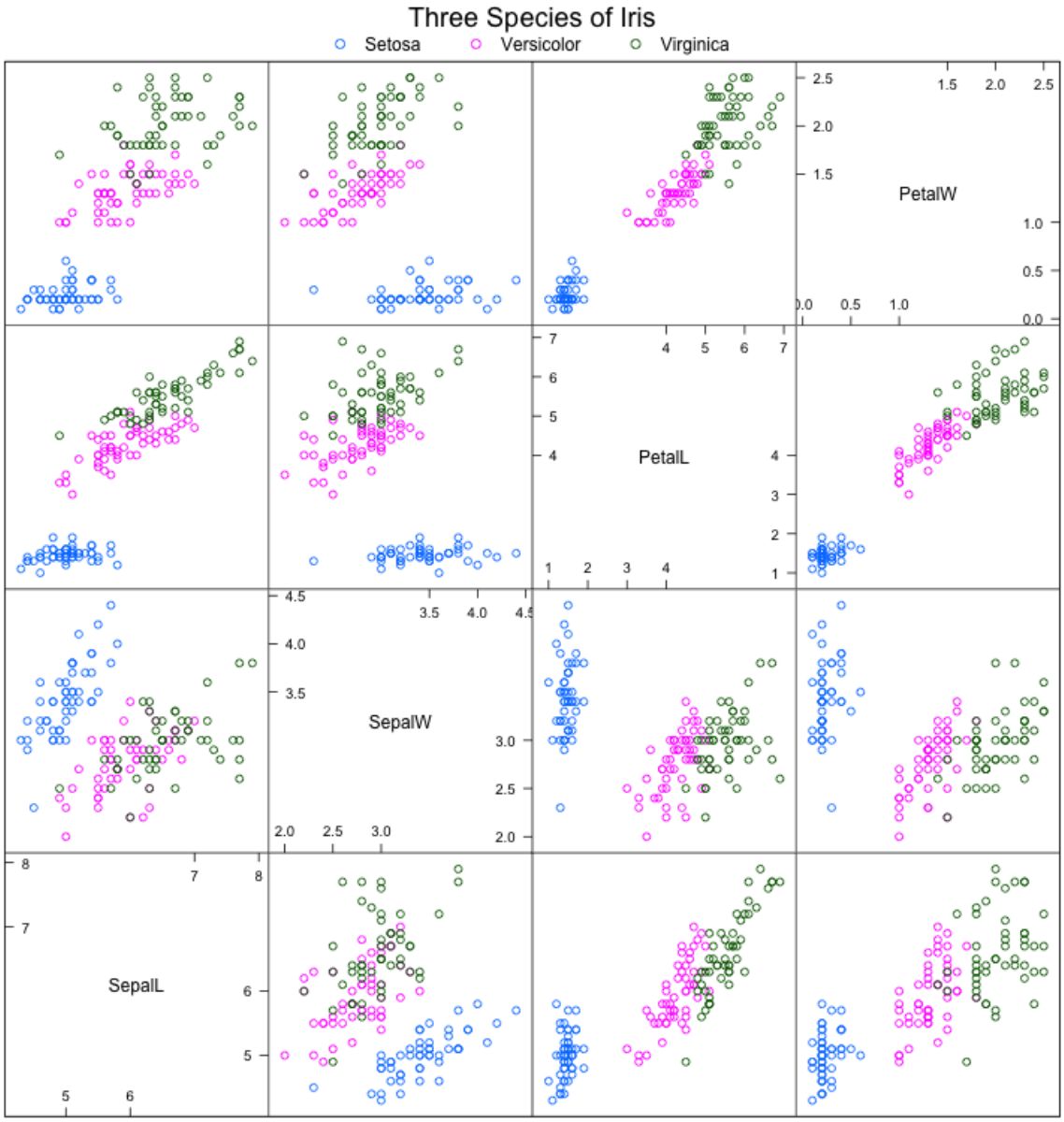


Figure 4.9: The scatter plot matrix of the Iris data

the two points crosses the center of mass of the petal. By such an interpretation, the sepal size can be described by interval-valued data where the lower point of an interval is the sepal width and upper point of the intervals is the sepal lengths. The interval-valued petal size can

be formed in a similar way. The interval-valued Iris data is presented in Table 4.10 where the sepal size is formed by the sepal width and sepal length, and petal size is formed by the petal width and petal length.

Table 4.10: Interval-valued Iris data

	Setosa		Versicolor		Virginica	
N	sepal size	petal size	sepal size	petal size	sepal size	petal size
1	[3.5, 5.1]	[0.2, 1.4]	[3.2, 7.0]	[1.4, 4.7]	[3.3, 6.3]	[2.5, 6.0]
2	[3.0, 4.9]	[0.2, 1.4]	[3.2, 6.4]	[1.5, 4.5]	[2.7, 5.8]	[1.9, 5.1]
3	[3.2, 4.7]	[0.2, 1.3]	[3.1, 6.9]	[1.5, 4.9]	[3.0, 7.1]	[2.1, 5.9]
4	[3.1, 4.6]	[0.2, 1.5]	[2.3, 5.5]	[1.3, 4.0]	[2.9, 6.3]	[1.8, 5.6]
5	[3.6, 5.0]	[0.2, 1.4]	[2.8, 6.5]	[1.5, 4.6]	[3.0, 6.5]	[2.2, 5.8]
6	[3.9, 5.4]	[0.4, 1.7]	[2.8, 5.7]	[1.3, 4.5]	[3.0, 7.6]	[2.1, 6.6]
7	[3.4, 4.6]	[0.3, 1.4]	[3.3, 6.3]	[1.6, 4.7]	[2.5, 4.9]	[1.7, 4.5]
8	[3.4, 5.0]	[0.2, 1.5]	[2.4, 4.9]	[1.0, 3.3]	[2.9, 7.3]	[1.8, 6.3]
9	[2.9, 4.4]	[0.2, 1.4]	[2.9, 6.6]	[1.3, 4.6]	[2.5, 6.7]	[1.8, 5.8]
10	[3.1, 4.9]	[0.1, 1.5]	[2.7, 5.2]	[1.4, 3.9]	[3.6, 7.2]	[2.5, 6.1]
11	[3.7, 5.4]	[0.2, 1.5]	[2.0, 5.0]	[1.0, 3.5]	[3.2, 6.5]	[2.0, 5.1]
12	[3.4, 4.8]	[0.2, 1.6]	[3.0, 5.9]	[1.5, 4.2]	[2.7, 6.4]	[1.9, 5.3]
13	[3.0, 4.8]	[0.1, 1.4]	[2.2, 6.0]	[1.0, 4.0]	[3.0, 6.8]	[2.1, 5.5]
14	[3.0, 4.3]	[0.1, 1.1]	[2.9, 6.1]	[1.4, 4.7]	[2.5, 5.7]	[2.0, 5.0]
15	[4.0, 5.8]	[0.2, 1.2]	[2.9, 5.6]	[1.3, 3.6]	[2.8, 5.8]	[2.4, 5.1]
16	[4.4, 5.7]	[0.4, 1.5]	[3.1, 6.7]	[1.4, 4.4]	[3.2, 6.4]	[2.3, 5.3]
17	[3.9, 5.4]	[0.4, 1.3]	[3.0, 5.6]	[1.5, 4.5]	[3.0, 6.5]	[1.8, 5.5]
18	[3.5, 5.1]	[0.3, 1.4]	[2.7, 5.8]	[1.0, 4.1]	[3.8, 7.7]	[2.2, 6.7]
19	[3.8, 5.7]	[0.3, 1.7]	[2.2, 6.2]	[1.5, 4.5]	[2.6, 7.7]	[2.3, 6.9]
20	[3.8, 5.1]	[0.3, 1.5]	[2.5, 5.6]	[1.1, 3.9]	[2.2, 6.0]	[1.5, 5.0]
21	[3.4, 5.4]	[0.2, 1.7]	[3.2, 5.9]	[1.8, 4.8]	[3.2, 6.9]	[2.3, 5.7]
22	[3.7, 5.1]	[0.4, 1.5]	[2.8, 6.1]	[1.3, 4.0]	[2.8, 5.6]	[2.0, 4.9]
23	[3.6, 4.6]	[0.2, 1.0]	[2.5, 6.3]	[1.5, 4.9]	[2.8, 7.7]	[2.0, 6.7]
24	[3.3, 5.1]	[0.5, 1.7]	[2.8, 6.1]	[1.2, 4.7]	[2.7, 6.3]	[1.8, 4.9]
25	[3.4, 4.8]	[0.2, 1.9]	[2.9, 6.4]	[1.3, 4.3]	[3.3, 6.7]	[2.1, 5.7]
26	[3.0, 5.0]	[0.2, 1.6]	[3.0, 6.6]	[1.4, 4.4]	[3.2, 7.2]	[1.8, 6.0]
27	[3.4, 5.0]	[0.4, 1.6]	[2.8, 6.8]	[1.4, 4.8]	[2.8, 6.2]	[1.8, 4.8]
28	[3.5, 5.2]	[0.2, 1.5]	[3.0, 6.7]	[1.7, 5.0]	[3.0, 6.1]	[1.8, 4.9]
29	[3.4, 5.2]	[0.2, 1.4]	[2.9, 6.0]	[1.5, 4.5]	[2.8, 6.4]	[2.1, 5.6]
30	[3.2, 4.7]	[0.2, 1.6]	[2.6, 5.7]	[1.0, 3.5]	[3.0, 7.2]	[1.6, 5.8]
31	[3.1, 4.8]	[0.2, 1.6]	[2.4, 5.5]	[1.1, 3.8]	[2.8, 7.4]	[1.9, 6.1]
32	[3.4, 5.4]	[0.4, 1.5]	[2.4, 5.5]	[1.0, 3.7]	[3.8, 7.9]	[2.0, 6.4]
33	[4.1, 5.2]	[0.1, 1.5]	[2.7, 5.8]	[1.2, 3.9]	[2.8, 6.4]	[2.2, 5.6]
34	[4.2, 5.5]	[0.2, 1.4]	[2.7, 6.0]	[1.6, 5.1]	[2.8, 6.3]	[1.5, 5.1]
35	[3.1, 4.9]	[0.2, 1.5]	[3.0, 5.4]	[1.5, 4.5]	[2.6, 6.1]	[1.4, 5.6]
36	[3.2, 5.0]	[0.2, 1.2]	[3.4, 6.0]	[1.6, 4.5]	[3.0, 7.7]	[2.3, 6.1]
37	[3.5, 5.5]	[0.2, 1.3]	[3.1, 6.7]	[1.5, 4.7]	[3.4, 6.3]	[2.4, 5.6]
38	[3.6, 4.9]	[0.1, 1.4]	[2.3, 6.3]	[1.3, 4.4]	[3.1, 6.4]	[1.8, 5.5]
39	[3.0, 4.4]	[0.2, 1.3]	[3.0, 5.6]	[1.3, 4.1]	[3.0, 6.0]	[1.8, 4.8]
40	[3.4, 5.1]	[0.2, 1.5]	[2.5, 5.5]	[1.3, 4.0]	[3.1, 6.9]	[2.1, 5.4]
41	[3.5, 5.0]	[0.3, 1.3]	[2.6, 5.5]	[1.2, 4.4]	[3.1, 6.7]	[2.4, 5.6]
42	[2.3, 4.5]	[0.3, 1.3]	[3.0, 6.1]	[1.4, 4.6]	[3.1, 6.9]	[2.3, 5.1]
43	[3.2, 4.4]	[0.2, 1.3]	[2.6, 5.8]	[1.2, 4.0]	[2.7, 5.8]	[1.9, 5.1]
44	[3.5, 5.0]	[0.6, 1.6]	[2.3, 5.0]	[1.0, 3.3]	[3.2, 6.8]	[2.3, 5.9]
45	[3.8, 5.1]	[0.4, 1.9]	[2.7, 5.6]	[1.3, 4.2]	[3.3, 6.7]	[2.5, 5.7]
46	[3.0, 4.8]	[0.3, 1.4]	[3.0, 5.7]	[1.2, 4.2]	[3.0, 6.7]	[2.3, 5.2]
47	[3.8, 5.1]	[0.2, 1.6]	[2.9, 5.7]	[1.3, 4.2]	[2.5, 6.3]	[1.9, 5.0]
48	[3.2, 4.6]	[0.2, 1.4]	[2.9, 6.2]	[1.3, 4.3]	[3.0, 6.5]	[2.0, 5.2]
49	[3.7, 5.3]	[0.2, 1.5]	[2.5, 5.1]	[1.1, 3.0]	[3.4, 6.2]	[2.3, 5.4]
50	[3.3, 5.0]	[0.2, 1.4]	[2.8, 5.7]	[1.3, 4.1]	[3.0, 5.9]	[1.8, 5.1]

The interval-valued Iris data now have two attributes, sepal size and petal size. We would like to apply the ORCA algorithm to the interval-valued Iris data and recover the three species. Figure 4.10 shows the relation between the sepal size and petal size for the interval-valued Iris data. In a similar manner as for the classical Iris data, the species Setosa is relatively separated from the other two species, while the species Versicolor and Virginica are largely overlapped.

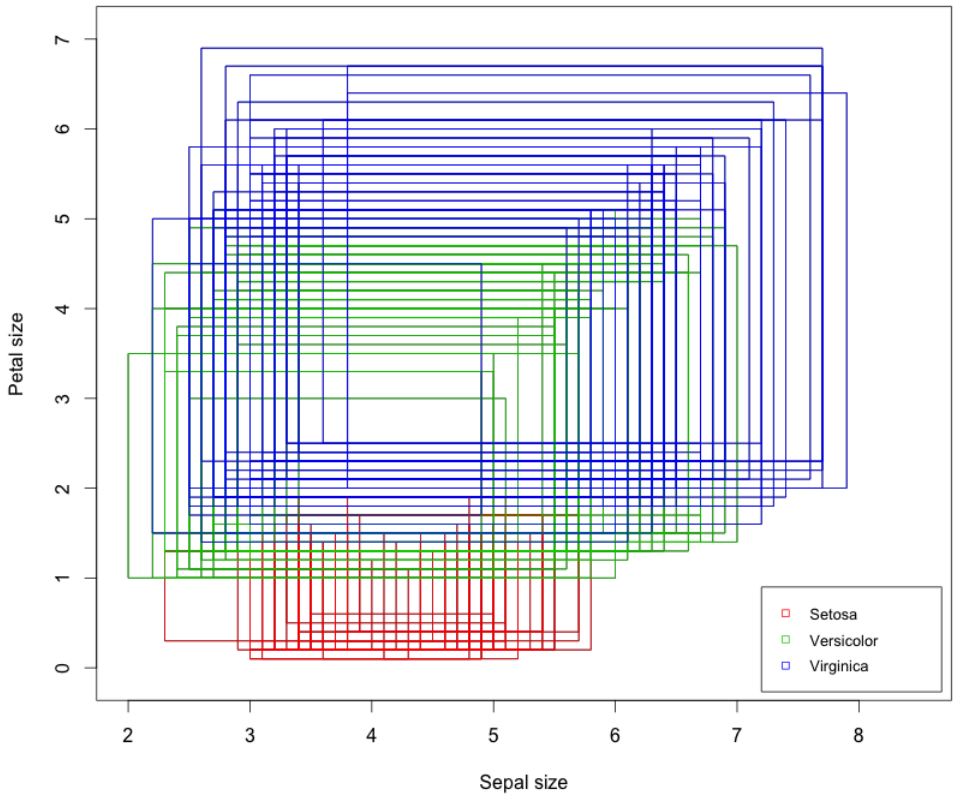


Figure 4.10: Sepal size and petal size of three species for interval-valued Iris data

We cluster the interval-valued Iris data by applying all the three orthogonal regression methods of ORCA, ORCA by the simple orthogonal regression method (center), the simple orthogonal regression method (min max), and the general orthogonal regression method. We first look at the clustering results given the number of clusters to be two and three. Then we apply the methods discussed in section 4.3 to determine an optimal number of clusters

for the Iris data.

Table 4.11 compares frequencies of the three true species with the clustered groups by the ORCA with three orthogonal regression methods. Note that the labels of the clustered group, 1, 2, 3, are merely to distinguish between different clusters. For each clustered group, we count its members as the frequency of each of the three species, Setosa, Versicolor, and Virginica. For example, given the number of clusters $K = 2$, all the members of the cluster 1 of the ORCA clustering results by simple OR (center) method are Iris Setosa. All the members in the second cluster are Iris Versicolor and Iris Virginica. For the same method, given $K = 3$, 50 Setosa are clustered into the first group, 41 Versicolor and 2 Virginica are clustered into the second group, while the remaining 9 Versicolor and 48 virginica are clustered into the third group.

Table 4.11: Comparison between the true species and the ORCA clustered groups for the Iris data

	Number of clusters	Clustered groups	Setosa	Versicolor	Virginica
Simple OR (center)	K=2	1	50	0	0
		2	0	50	50
	K=3	1	50	0	0
		2	0	41	2
		3	0	9	48
Simple OR (min max)	K=2	1	50	0	0
		2	0	50	50
	K=3	1	50	0	0
		2	0	44	2
		3	0	6	48
General OR	K=2	1	50	0	0
		2	0	50	50
	K=3	1	50	0	0
		2	0	38	2
		3	0	12	48

We can see from Table 4.11 that given $K = 2$, all the three OR methods of the ORCA

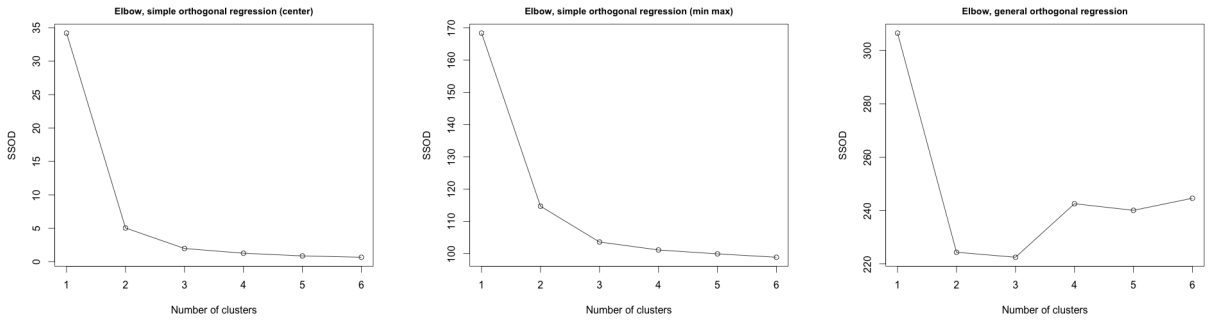
cluster the 50 Iris Setosa as one cluster, while they cluster the 50 Iris Versicolor and 50 Iris Virginica as the other cluster. Given the number of clusters $K = 3$, all the three OR methods of ORCA still cluster the 50 Iris Setosa as one cluster. While the most observations of the Iris Versicolor and Iris Virginica are clustered into two different clusters, there are observations in each of these two species that are clustered into a group where the majority is a different species. This is not surprising given the large overlap between the species Versicolor and Virginica. The clusters results from the three OR methods are a little different but they are comparable. Note that the ORCA is not a supervised algorithm which tries to classify an observation to its species based on the relation between predictor variables and the label of species. The ORCA algorithm clusters the observations into different groups based on the linear separability of the data without the species information. From the results in Table 4.11, the performance of the ORCA algorithm for all the three OR methods works well to cluster the data based on the linear separability of the data given appropriate number of clusters.

Table 4.12: Optimal number of clusters determined by different metrics

	Simple OR (center)	Simple OR (min max)	General OR
Gap	2	1	1
Silhouette	2	2	2
$SD(10\%)$	-	2	2
$SD(25\%)$	5	2	2
$SD(50\%)$	3	1	1
AIC	-	-	2
BIC	-	-	2

For a clustering problem, usually the information of the optimal number of clusters is not available. We apply the different metrics discussed in section 4.3 to determine the optimal number of clusters for the Iris data clustered by ORCA. Given the maximum number of clusters to be six, Table 4.12 shows the decision of the optimal number of clusters by different metrics for the ORCA by the three orthogonal regression methods. For the ORCA using the

simple orthogonal regression method (center), the optimal number of clusters is 2 by the Gap statistic and the silhouette statistic. The number of clusters reaches its optimal at 5 and 3 for the elbow method by SD (4.47) with cutoff 25% and 50%, respectively. None of the SD_K is less than 10% for $K = 1, \ldots, 5$. For the ORCA using the simple orthogonal method (min max), the number of optimal clusters given by the silhouette statistic, the elbow method with SD cutoff 10% and 25% are all 2, while this number is 1 for the Gap statistic and the elbow method with SD cutoff 50%. For the ORCA applying the general orthogonal regression method, the optimal number of clusters is 2 for the silhouette statistic, the elbow method with SD cutoff 10% and 20%, the AIC, and the BIC, while the Gap statistic and the elbow method with SD cutoff 50% reach their optimal at 1 cluster.



(a) (b) (c)

Figure 4.11: Elbow plots of ORCA results for the interval-valued Iris data

We understand that an appropriate SD cutoff of the elbow method can be different for different variables scales, data structures, or clustering methods. An elbow plot is helpful to determine the optimal number of clusters. Figure 4.11 (a), (b), (c) give the elbow plots for the ORCA using the simple orthogonal regression method (center), the simple orthogonal regression method (min max) and the general orthogonal regression method, respectively. It is arguable that 2 or 3 is the optimal number of clusters by Figure 4.11 (a) for the simple orthogonal regression method (center). We can see from the elbow plots in Figure 4.11 (b) that the simple orthogonal regression method (min max) is in favor of 3 clusters. The elbow

plot in Figure 4.11 indicates that the general orthogonal regression method is in favor of 2 clusters. For all the three methods from the elbow plots, we can see that there is a close competition between 2 clusters and 3 clusters.

Table 4.13: Different metrics for number of clusters $K = 1, \ldots, 6$

ORCA	Method	Number of clusters					
		1	2	3	4	5	6
Simpe OR	SSOD	34.21	5.04	1.97	1.26	0.86	0.66
(center)	Gap	0.82	1.31	1.34	1.3	1.26	1.29
		(0.05)	(0.08)	(0.11)	(0.12)	(0.12)	(0.12)
	Silhouette	-	0.69	0.58	0.56	0.5	0.49
	SD_K	0.85	0.61	0.35	0.32	0.23	-
Simpe OR	SSOD	168.35	114.68	103.59	101.12	99.91	98.86
(min max)	Gap	-0.82	-1.82	-2.6	-3.07	-3.51	-3.75
		(0.08)	(0.06)	(0.07)	(0.14)	(0.18)	(0.14)
	Silhouette	-	0.26	0.17	0.16	0.15	0.14
	SD_K	0.32	0.10	0.02	0.01	0.01	-
General OR	SSOD	306.52	224.37	222.50	242.56	240.09	244.62
	Gap	-1.62	-2.75	-3.56	-4.21	-4.5	-4.85
		(0.06)	(0.09)	(0.11)	(0.05)	(0.12)	(0.10)
	Silhouette	-	0.91	0.76	0.64	0.62	0.65
	SD_K	0.27	0.01	-0.09	0.01	-0.02	-
	AIC	312.52	236.37	240.50	266.56	270.09	280.62
	BIC	321.55	254.43	267.60	302.69	315.25	334.81

Table 4.13 presents the detailed values for each of the metrics given the number of clusters $K = 1, \ldots, 6$. For each orthogonal regression method, SSOD is the sum of squared orthogonal distances between observations and their closest regression line, Gap is the Gap statistic, silhoutte is the silhoutte statistic, SD_K is the percentage of SSOD decrement defined in equation (4.47). The values in parentheses under the Gap statistic are s_{K+1} (see equation (4.54)) that is used to determine the optimal number of clusters combined with the Gap statistic. We can see that an appropriate SD cutoff is much different between the three orthogonal regression methods. By the silhouette statistic for the general orthogonal regression method, 0.91 for two clusters and 0.76 for three clusters, the separation between the clusters is good by assuming normal distributed measurement error. The advantage of

2 clusters to 3 clusters is small by the silhouette statistics for all the three methods.

In summary, the ORCA using the three orthogonal regression methods can all recover the clusters given the number of clusters to be two or three. The optimal number of clusters by the elbow method, the information approaches, the Gap statistic, and the silhouette statistic are generally in favor of 2, though the competition between 2-cluster and 3-cluster is close. The determination of silhouette statistics and elbow methods are relatively more stable and more reliable.

4.6 Appendix

4.6.1 Parameter Setup of the Simulated Data Sets

This section gives the parameter setup for the six simulated data sets we have used to illustrate the ORCA algorithm in section 4.4. The simulation method for interval values of each data set follows the simulation method III in section 3.4. For each Data I-IV, the variable x_2 is set to be the response variable, x_1 is the predictor variable. For Data V, the variable x_3 is set to be the response variable, while all other variables are the predictor variables. For Data VI, the variable x_5 is the response variable and all other variables are the predictor variables.

In the following description of the parameter setup for the six data sets, each data name is followed by the number of clusters, data dimensions, and the sample size for each cluster. For example, Data I is a two-dimensional data set with three clusters where the sample sizes for the three clusters are $n_1 = 100$, $n_2 = 50$, and $n_3 = 50$. In the equations (4.61), (4.62), (4.63), (4.64), (4.65), and (4.66), "Cluster" gives the linear regression relations between all the variables for each cluster. The interval center points of the predictor variables for the k^{th} cluster of a particular data set are drawn from normal distribution $N(\mu_x, \sigma_x^2)$ where the μ_x and σ_x are the k^{th} row of $\boldsymbol{\mu_x}$ and $\boldsymbol{\sigma_x}$ for two-dimensional data. For multi-dimensional data set, σ_x is $\boldsymbol{\sigma_x^{(k)}}$ in the equations. The interval ranges of all the predictor variables for a particular cluster of a data set are drawn from exponential distributions $\exp \lambda$ where λ is the values in the k^{th} row of $\boldsymbol{\lambda}$. The error terms of k^{th} are generated from a normal distribution $N(0, \sigma_e)$ where σ_e is the value in the k^{th} row of $\boldsymbol{\sigma_e}$ in the equations.

We take the first cluster of Data I as example to demonstrate the simulation process. The sample size of the cluster 1 of Data I is 100. From the parameter setup in equation (4.61), the 100 interval center points of x_1 of the cluster 1, $x_{i1}^{(c)}$, $i = 1, \ldots, 100$, are independently

generated from a normal distribution $N(4, 18)$. he 100 interval ranges of x_1 for the cluster 1, $x_{i1}^{(r)}$, $i = 1, \ldots, 100$, are generated from an exponential distribution exp(1.5). The 100 interval values of x_1 are then obtained as $[x_{i1}^{(c)} - 0.5x_{i1}^{(r)}, x_{i1}^{(c)} + 0.5x_{i1}^{(r)}]$, $i = 1, \ldots, 100$. For each of $i = 1, \ldots, 100$, we randomly draw 5 values from a uniform distribution, $x_{i1l} \sim U(x_{i1a}, x_{i1b})$, $l = 1, \ldots, 5$. The i^{th} observation of response variable x_2 is $x_{i2} = [x_{i2a}, x_{i2b}]$, $i = 1, \ldots, 100$, that is obtained as

$$\begin{aligned} x_{i2a} &= \min_{l \in \{1,\ldots,5\}} \{8 + 1.3x_{i1l} + \epsilon_{il}\}, \\ x_{i2b} &= \min_{l \in \{1,\ldots,5\}} \{8 + 1.3x_{i1l} + \epsilon_{il}\}, \end{aligned} \tag{4.60}$$

where ϵ_{il}, $l = 1, \ldots, 5$, is the error term that follow a normal distribution $N(0, 4)$. The observations for the cluster 2 and 3 are analogously with the cluster 1. The interval center points of x_1 for the cluster 2 and 3 follow normal distributions $N(0, 19)$ and $N(8, 18)$, respectively. The interval ranges of x_1 for the cluster 2 and 3 are generated from exponential distributions exp(1.3) and exp(1.2), respectively. The error terms of the cluster 2 and 3 are generated from normal distributions $N(0, 5)$ and $N(0, 5)$, respectively. We stack the simulated observations of the three clusters to obtain the Data I. The simulation of Data II-VI are analogously with Data I.

The detailed parameter setup for the six data sets is as follows:

Data I ($K = 3$, $p = 2$, $n_1 = 100$, $n_2 = 50$, $n_3 = 50$):

$$\begin{aligned}
\text{Cluster } 1: \ & x_2 = -8 + 1.3x_1, \\
2: \ & x_2 = 45 + 2.8x_1, \\
3: \ & x_2 = 35 - 2.5x_1, \\
& \boldsymbol{\mu_x} = (4, 0, 8)', \\
& \boldsymbol{\sigma_x} = (18, 19, 18)', \\
& \boldsymbol{\lambda} = (1.5, 1.3, 1.2)', \\
& \boldsymbol{\sigma_e} = (4, 5, 5)'.
\end{aligned} \tag{4.61}$$

Data II ($K = 5$, $p = 2$, $n_i = 100$, $i = 1, \ldots, 5$):

$$\begin{aligned}
\text{Cluster } 1: \ & x_2 = 8 + 0.8x_1, \\
2: \ & x_2 = 85 - 0.5x_1, \\
3: \ & x_2 = 4 + 0.2x_1, \\
4: \ & x_2 = 5 + 0.5x_1, \\
5: \ & x_2 = 13 + x_1, \\
& \boldsymbol{\mu_x} = (50, 86, 47, 46, 45)', \\
& \boldsymbol{\sigma_x} = (15, 20, 16, 16, 15)', \\
& \boldsymbol{\lambda} = (2, 1.2, 1, 2, 2.2)', \\
& \boldsymbol{\sigma_e} = (2, 2, 2, 2, 2)'.
\end{aligned} \tag{4.62}$$

Data III ($K = 3$, $p = 2$, $n_1 = 100$, $n_2 = 50$, $n_3 = 25$):

$$\begin{aligned}
\text{Cluster } 1: \ x_2 =& 35 + 2x_1, \\
2: \ x_2 =& 45 - 2.8x_1, \\
3: \ x_2 =& 1 + 0.8x_1, \\
\boldsymbol{\mu_x} =& (4, 0, 8)', \\
\boldsymbol{\sigma_x} =& (12, 9.6, 12)', \\
\boldsymbol{\lambda} =& (1.5, 1.3, 1.2)', \\
\boldsymbol{\sigma_e} =& (7, 7, 7)'.
\end{aligned} \tag{4.63}$$

Data IV ($K = 2$, $p = 2$, $n_1 = 50$, $n_2 = 50$):

$$\begin{aligned}
\text{Cluster } 1: \ x_2 =& 15 + 0.9x_1, \\
2: \ x_2 =& -15 - 0.8x_1, \\
\boldsymbol{\mu_x} =& (4, 3)', \\
\boldsymbol{\sigma_x} =& (15, 14)', \\
\boldsymbol{\lambda} =& (1.5, 1.3)', \\
\boldsymbol{\sigma_e} =& (6, 6)'.
\end{aligned} \tag{4.64}$$

Data V ($K = 3$, $p = 3$, $n_i = 40$, $i = 1, 2, 3$):

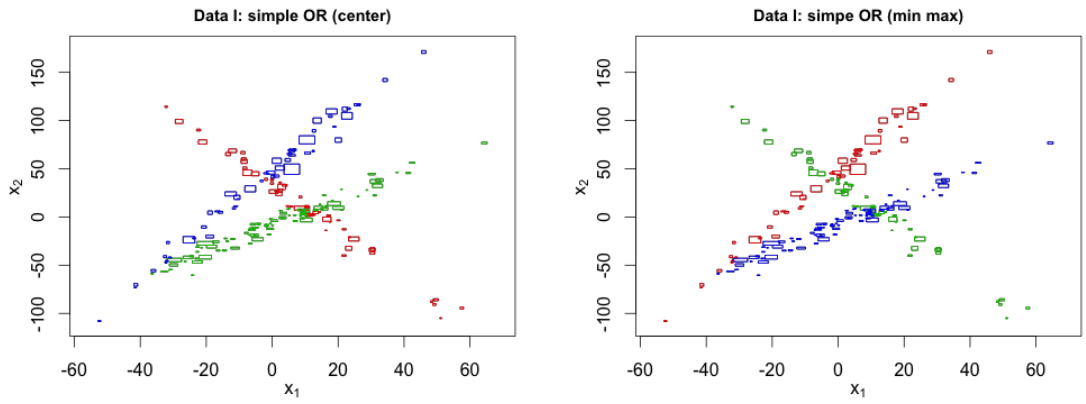
$$
\begin{aligned}
\text{Cluster } 1: \; x_3 =& 1 + 1.3x_1 + 1.5x_2, \\
2: \; x_3 =& 4.5 - 1.8x_1 - 3x_2, \\
3: \; x_3 =& 35 - 3.5x_1 + 10x_2, \\
\boldsymbol{\mu_x} =& \begin{bmatrix} 4 & 5 \\ -3 & -3 \\ 8 & 12 \end{bmatrix}, \\
\boldsymbol{\sigma}_{\boldsymbol{x}}^{(1)} =& \operatorname{diag}(20, 8), \\
\boldsymbol{\sigma}_{\boldsymbol{x}}^{(2)} =& \operatorname{diag}(20, 9), \\
\boldsymbol{\sigma}_{\boldsymbol{x}}^{(3)} =& \operatorname{diag}(20, 8), \\
\boldsymbol{\lambda} =& \begin{bmatrix} 15 & 12 \\ 13 & 12 \\ 12 & 12 \end{bmatrix}, \\
\boldsymbol{\sigma_e} =& (1, 1, 1)'.
\end{aligned}
\tag{4.65}
$$

Data VI ($K = 2$, $p = 5$, $n_1 = n_2 = 50$):

$$
\begin{aligned}
\text{Cluster } 1: \ x_5 =& 1 + 1.3x_1 + 1.5x_2 + 2x_3 + 4x_4, \\
2: \ x_5 =& 4.5 - 1.8x_1 - 3x_2 + 5x_3 + x_4, \\
\boldsymbol{\mu_x} =& \begin{bmatrix} 4 & 8 & 5 & 12 \\ -3 & 0 & -3 & 2 \end{bmatrix}, \\
\boldsymbol{\sigma}_{\boldsymbol{x}}^{(1)} =& \ \text{diag}(20, 20, 8, 8), \\
\boldsymbol{\sigma}_{\boldsymbol{x}}^{(2)} =& \ \text{diag}(20, 19, 9, 10), \\
\boldsymbol{\lambda} =& \begin{bmatrix} 2 & 1 & 2.3 & 2.4 \\ 1.5 & 2.5 & 1.5 & 2.4 \end{bmatrix}, \\
\boldsymbol{\sigma_e} =& (1, 1)'.
\end{aligned}
\tag{4.66}
$$

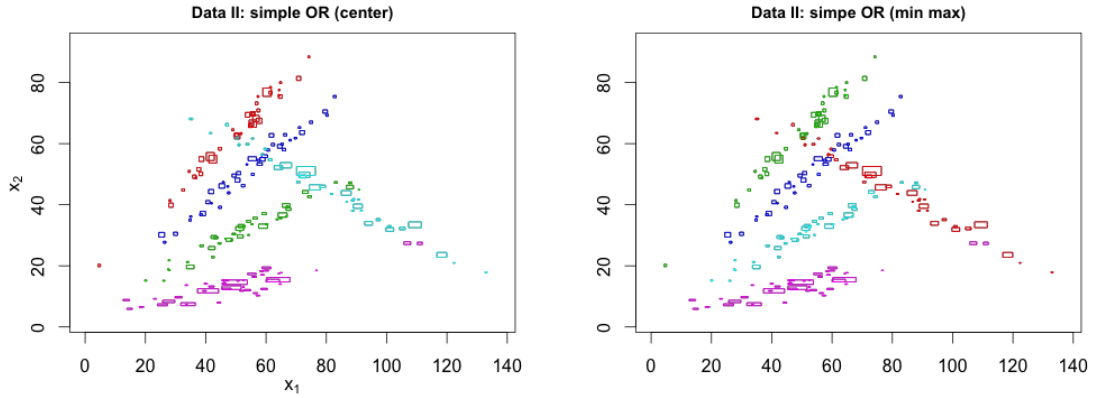
4.6.2 The ORCA Results for Data I-IV

We have presented the ORCA results of Data I-IV in section 4.4.1 implemented by the general orthogonal regression method. The clustering results when applying ORCA on Data I-IV with the simple orthogonal regression method center and the simple orthogonal regression method (min max) are ploted in Figure 4.12, 4.13, 4.14, and 4.15. Different colors represent different clusters, but a particular color does not necessarily associate with a particular cluster. The clustering results can be compared with the original clusters in Figure 4.5, Figure 4.6, Figure 4.7, Figure 4.8 for Data I-IV, respectively , to verify their correctness.



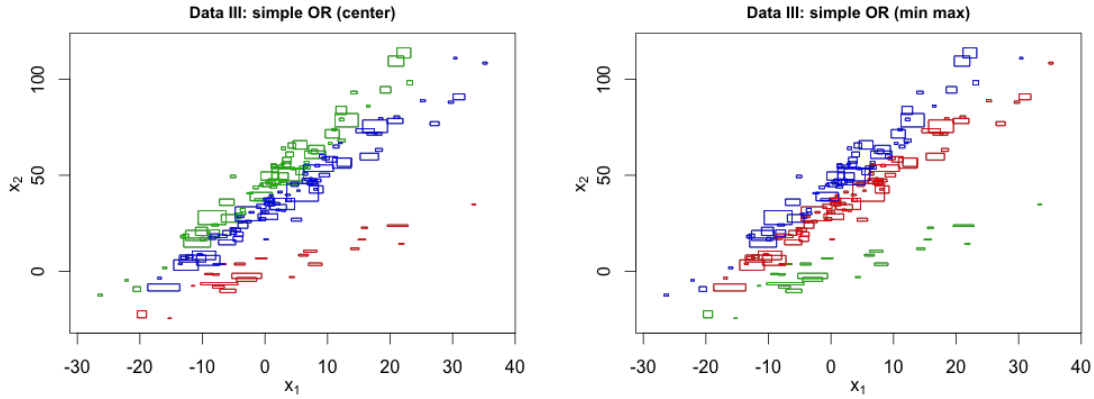
(a) Simple OR (center) (b) Simple OR (min max)

Figure 4.12: ORCA results for Data *I*



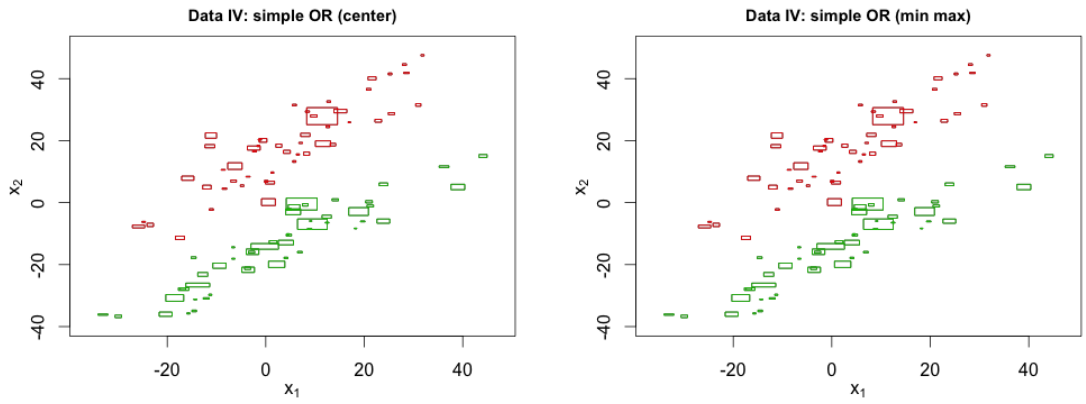
(a) Simple OR (center) (b) Simple OR (min max)

Figure 4.13: ORCA results for Data *II*



(a) Simple OR (center) (b) Simple OR (min max)

Figure 4.14: ORCA results for Data *III*



(a) Simple OR (center) (b) Simple OR (min max)

Figure 4.15: ORCA results for Data *IV*

4.6.3 R Code for the Implementation of the ORCA

```
# ------------------------------------------
# Standize the interval-valued data to be with variance=1
st.int <- function(data)
{ p <- ncol(data)/2
  st.d <- data
  m <- s <- NULL
  for (i in 1:p)
  {
    m <- c(m,mean(data[,(2*i-1):(2*i)]))
    s <- c(s,cov.int(data[,(2*i-1):(2*i)]))
    st.d[,(2*i-1):(2*i)] <- (data[,(2*i-1):(2*i)])/sqrt(s[i])
  }
  return(st.d)
}

# ------------------------------------------
# Calculate the variance covariance matrix and the correlation
# matrix given a multivariate interval-valued data
covMat<-function(d)
{ p<-ncol(d)/2
  cov <- corr <- matrix(0,p,p)
  cov[1,1] <- cov.int(d[,1:2])
  for (i in 1:p)
```

```
  {
    corr[i,i] <- 1
    for(j in (i+1):p)
    { if(j>p) break
      cov[j,j] <- cov.int(d[,(2*j-1):(2*j)])
      cov[j,i] <- cov[i,j] <- cov.int(d[,(2*i-1):(2*i)],d[,(2*j-1):(2*j)])

      corr[j,i] <- corr[i,j] <- cov[i,j]/sqrt(cov[i,i]*cov[j,j])
    }
  }
  return(list(covMat=cov,corrMat=corr))
}

# ------------------------------------------------------------------
#define the function to calculate the min max orthogonal distance
aod<-function(x,m,p,clusmean)
{ xa <- x[(1:p)*2-1]
  xb <- x[(1:p)*2]
  Dmin<-sum(apply(cbind(xa*m,xb*m),1,min))-sum(clusmean*m)
  Dmax<-sum(apply(cbind(xa*m,xb*m),1,max))-sum(clusmean*m)
  if (Dmin*Dmax<0)
    d<-sqrt(sum(m^2))^-1*max(abs(Dmin),abs(Dmax))/2
  else
    d<-sqrt(sum(m^2))^-1*(abs(Dmin)+abs(Dmax))/2
  return(d)
}

# ------------------------------------------
# PCA on the correlation matrix obtained from the covMat
# function, or the covariance matrix for the standardized
# data. They are equivalent

# Using orthogonal regression to do cluster-wise regression
# for interval-valued data

# input:
#   data: the interval-valued data that would be clustered
#   K: number of clusters
#
# ------------------------------------------------------------------
library(caret)
orthClus<-function(data,K,max.iter,distance="center",scale=T)
{ if(scale==T) data<-st.int(data)
  p<-ncol(data)/2
  n<-nrow(data)
```

```
#randomly choose K (p+1) subsets
intl<-sample(n,K*(p+1))

#initialize m that will save the linear models at each step
m<-vector("list",K)   # m saves the beta coefficients for each cluster
group<-rep(0,n)       # group saves the membership for each obs
residual<-regroup<-rep(0,n)
#residual saves the distance between an obs and its nearest hyperplane
#regroup saves the obs membership after each iteration

meanAll<-NULL #meanAll calculate the mean vector for each cluster
for (k in 1:K)
{ m[[k]] <- princomp(covmat=covMat(data[intl[((k-1)*(p)+1):(k*(p))],])[[1]])
   $loadings[,p]
      t<-NULL
  for (l in 1:p)
  {t <- c(t,mean(data[intl[((k-1)*(p+1)+1):(k*(p+1))],(2*l-1):(2*l)]))}
   meanAll<-rbind(meanAll,t)
}

for (j in 1:n)
{ if(distance=="center")
      { xm <- (data[j,(1:p)*2-1]+data[j,(1:p)*2])*.5
        res <-sapply(1:K,f<-function(x){abs(sum(m[[x]]*(xm-meanAll[x,])))},
            simplify=T)
        }
  else if(distance=="average")
       res <- sapply(1:K, function(x) aod(data[j,],m[[x]],p,meanAll[x,]))

  residual[j] <- min(res)
  regroup[j] <- which(res==min(res))[1]
}
group <- regroup

cond<-1
i<-0
while(cond && i<=max.iter)
{  i<-i+1

   if (!all(table(regroup)>p)||length(table(regroup))<K)
     stop("Number␣of␣observations␣is␣smaller␣than␣the␣number␣of␣parameters␣
        for␣one␣or␣more␣clusters!")

   meanAll <- NULL
```

```
    for(k in 1:K)
      {  t<-NULL
         for (l in 1:p)
         {t <- c(t,mean(data[group==k,(2*l-1):(2*l)]))}
         meanAll<-rbind(meanAll,t)

         m[[k]]<-princomp(covmat=covMat(data[regroup==k,])[[1]])$loadings[,p]
       }

    dis<-NULL # dis saves the distance between each obs and all fitted
       hyperplanes
    for (j in 1:n)
    { if(distance=="center")
       {xm <- (data[j,(1:p)*2-1]+data[j,(1:p)*2])*.5
        res <-sapply(1:K,f<-function(x){abs(sum(m[[x]]*(xm-meanAll[x,])))},
           simplify=T)
       }
      else if(distance=="average")
         res <- sapply(1:K, function(x) aod(data[j,],m[[x]],p,meanAll[x,]))

      dis<-rbind(dis,res)
      residual[j] <- min(res)
      regroup[j] <- which(res==min(res))[1]
    }

    cond<-!all(regroup==group)
    group<-regroup
  }
  if(K>1)
  {sortDis<-t(apply(dis,1,sort))
   s<-1-sqrt(sortDis[,1]/sortDis[,2])
  }
  else
  {s<-NA
   sortDis<-matrix(dis,n,1)
  }
  names(s)<-NULL

  return(list(sum.sq.residual = sum(residual^2), models = m, group = group,
     niter=i,silhouette=s))
}

# ------------------------------------------------------------
# orthogonal regression clustering - measurement error model (MEM)
# the orghReg function perform a general orthogonal regression for
```

```
# an interval-valued data.
# Input: interval-valued data
# Ouptputs: fitted model "m"; loglikelihood "loglike"; residual
#  multiplier "residualMultiplier".
# Left multiply the center point matrix of the interval-valued
# data by the residualMultiplier is the general orthogonal regression
# residual for each observation.
orthReg<-function(data)
{ p <- ncol(data)/2
  n<-nrow(data)
  st.d <- data
  meanp <- NULL
  for (i in 1:p)
  { meanp <- c(meanp,mean(data[,(2*i-1):(2*i)]))
    st.d[,(2*i-1):(2*i)] <- data[,(2*i-1):(2*i)]-meanp[i]
  }
  data <- st.d
  rm(st.d)

 # the interval mean is the observed valued for MEM: do
 # the measurement error is [x_ia-x_ib,x_ib-x_ia]: de
  do <- matrix(0,n,p)
  de <- matrix(0,n,2*p)
  for (i in 1:p)
  {do[,i]<-apply(data[,(2*i-1):(2*i)],1,mean)
   de[,(2*i-1):(2*i)] <- (data[,(2*i-1):(2*i)]-cbind(do[,i],do[,i]))*2
  }
  cove<-covMat(de)[[1]]
  cove<-diag(diag(cove))
  Mzz<-t(do)%*%do/p
 # model
  m<-eigen(solve(Mzz)%*%cove)[[2]][,1]
 #individual residual for each variables, a nxp matrix
  residualMultiplier<-t(cove%*%m%*%(m%*%cove%*%m)^(-1)%*%m)
 #loglike is the -2*loglikelihood, the smaller the better
  loglike<-n*log(det(cove))+sum(apply(do%*%residualMultiplier,1,function(x) x%
      *%solve(cove)%*%x))

  return(list(meanp=meanp,m=m,residualMultiplier=residualMultiplier,loglike=
      loglike,cove=cove))
}
# ------------------------------------------------------------------

# ------------------------------------------------------------------
# The function orthClusMEM perform ORCA by applying the general
```

```
# orthogonal regression method. The input is an interval-valued
# data, the given number of cluster K, and a maximum number of
# iteration.
# Outputs: SSOD; membership of each observation - group;
#   fitted models - m; log likelihood - loglike;
#   silhouette statistic - silhouette;
orthClusMEM<-function(data,K,max.iter)
{
  p<-ncol(data)/2
  n<-nrow(data)

  # the interval mean is the observed valued for MEM: do
  # the measurement error is [x_ia-x_ib,x_ib-x_ia]: de
  do <- matrix(0,n,p)
  de <- matrix(0,n,2*p)
  for (i in 1:p)
  {do[,i]<-apply(data[,(2*i-1):(2*i)],1,mean)
   de[,(2*i-1):(2*i)] <- (data[,(2*i-1):(2*i)]-cbind(do[,i],do[,i]))*2
  }

  # randomly choose K (p+1) subsets
  intl<-sample(n,K*(p+1))

  #initialize m that will save the linear models at each step
  m<-vector("list",K)
  group<-regroup<-rep(0,n)
  distance<-NULL

  for(k in 1:K)
  { #tt save a temp result
        tt<-orthReg(data[intl[((k-1)*(p+1)+1):(k*(p+1))],])
        m[[k]] <- tt$m
        residual<-sweep(do,2,tt$meanp)%*%tt$residualMultiplier
        distance<-cbind(distance,apply(residual,1,f<-function(x){x%*%solve(tt$
            cove)%*%x}))
  }
  regroup<-apply(distance,1,f<-function(x) {which(x==min(x))[1]})
  group <- regroup

  cond<-1
  i<-0
  while(cond && i<=max.iter)
  {  i<-i+1

        if (!all(table(regroup)>p)||length(table(regroup))<K)
```

```
       stop("Number of observations is smaller than the number of parameters
           for one or more clusters!")

         distance<-NULL
         for (k in 1:K)
         { tt<-orthReg(data[regroup==k,])
           m[[k]] <- tt$m
           residual<-sweep(do,2,tt$meanp)%*%tt$residualMultiplier
           distance<-cbind(distance,apply(residual,1,f<-function(x){x%*%solve(
              tt$cove)%*%x}))
          }
         regroup<-apply(distance,1,f<-function(x) {which(x==min(x))[1]})

     cond<-!all(regroup==group)
     group<-regroup
  }
  loglike<-sum(sapply(1:K,f<-function(x){orthReg(data[regroup==x,])$loglike}))

  if(K>1)
    {sortDis<-t(apply(distance,1,sort))
     s<-1-sqrt(sortDis[,1]/sortDis[,2])
    }
  else
    {s<-NA
     sortDis<-matrix(distance,n,1)
    }
  names(s)<-NULL
  return(list(sum.sq.residual=sum(sortDis[,1]),models=m,group=group,niter=i,
      loglike=loglike, silhouette=s))
}

# ------------------------------------------------------------------
# The RegClusGap function is to determine the optimal number of clusters by
    ORCA.
# The ORCA can be implemented by general OR, simple OR with center distance,
#   or simple OR with min max distance.
# Outputs: optimal K by Gap - optK; SSOR for K = 1~maxK - W;
#   Gap statistic - Gap; membership for each observation - group;
#   silhouette statistic for each observation - silhouette;
#   AIC - AIC; BIC - BIC.
# Given the outputs of the RegClusGap, to use the method of silhouette
    statistic,
#   information criterion approach, or elbow method to determine the optimal
#   number of clusters can also be implemented.
RegClusGap<-function(data,maxK,scale=FALSE,B,max.iter=50,nrep=50,method=
```

```
   orthClusMEM,...)
{       if(scale) data <- st.int(data)
        library(lga)
        n<-nrow(data)
        p<-ncol(data)/2
        # loadings of the principal component
        v<-princomp(covmat=covMat(data)[[1]])$loadings
        # combine the two end points of the same variable into one column
        x<-(data[,(1:p)*2-1]+data[,(1:p)*2])/2
        # transform the original values
        xx<-x%*%v
        bound<-cbind(apply(xx,2,min),apply(xx,2,max))
        # Gap: the Gap statistic
        # sdk: standard deviation of the log SSR of the B reference data
        # W: smallest SSR
        # Wk: average SSR for the B reference data
        Gap<-sdk<-W<-Wk<-group<-silhouette<-AIC<-BIC<-NULL
        for(K in 1:maxK)
        { logSSR<-NULL
                Wb<-Inf
                i<-0

                while(i < nrep) #nrep is the number of replication
                { t<-try(method(data,K,max.iter,...))
                  if(class(t)!="try-error")
                  { i<-i+1
                if(Wb>t$sum.sq.residual)
                { Wb<-t$sum.sq.residual
                   groupt<-t$group
                   silhouettet<-t$silhouette

                  AICt<-t$loglike+2*K*(2*p-1)
                  BICt<-t$loglike+log(n)*K*(2*p-1)
                 }
               }
             }
                W<-c(W,Wb)
                group<-cbind(group,groupt)
                if(K>1) silhouette<-cbind(silhouette,silhouettet)
                if(!is.null(AICt))
                {AIC<-c(AIC,AICt)
                 BIC<-c(BIC,BICt)
                }

                for(b in 1:B)
```

```
            { xb<-NULL
              for(j in 1:p)
              {xb<-cbind(xb,runif(n,bound[j,1],bound[j,2]))}
              zb<-xb%*%t(v) #transform back to the original scale
              SSRb<-lga(zb,k=K,scale=scale,silent=TRUE,biter=200)$ROSS
              logSSR<-c(logSSR,log(SSRb))
              }
            Wk<-c(Wk,mean(exp(logSSR)))
            Gap<-c(Gap,(sum(logSSR)/B-log(Wb)))
            sdk<-c(sdk,sqrt(sum((logSSR-mean(logSSR))^2)/B*(1+1/B)))
        }
        optK<-which(Gap[-K]>=(Gap[-1]-sdk[-1]))[1]
        return(list(optK=optK,W=W,Wk=Wk,sdk=sdk,Gap=Gap,group=group,silhouette
            =silhouette,AIC=AIC,BIC=BIC))
}
```

Chapter 5

Future Work

In section 4.1.1 we proposed an orthogonal regression model for interval-valued data by a measurement error model, the general orthogonal regression method. We used a rigid assumption in equation (4.18) that restrict the true value of an observation to be always within the interval of the observation. For convenience purposes, we rewrite the assumption in equation (4.18) here

$$u_{ij} \sim U(x_{ija} - x_{ij}^{(c)}, x_{ijb} - x_{ij}^{(c)}), \tag{5.1}$$

where u_{ij} is the measurement errors for the i^{th} observation of variable X_j, $x_{ij} = [x_{ija}, x_{ijb}]$, and $x_{ij}^{(c)}$ is the interval center point of x_{ij}. In other words, we assume the linear regression line always crosses the hypercube of each observation. Since our focus in this dissertation is to recover the clusters for interval-valued data that are clustering around linear regression lines, we did not study the properties of our general orthogonal regression method using measurement error model. The simulation study in section 4.4 shows the orthogonal regression clustering algorithm (ORCA) using the general orthogonal regression method can still converge to the correct clusters when the assumption in equation (4.18) is violated, though the convergence does usually require trying more initial partitions. For example, from Figure 4.4 for the data set in equation (4.57), it is impossible that a linear regression line can

cross all the rectangles for a particular cluster. The ORCA using the general orthogonal regression method can still recover the true structure for this data set. Future research about relaxing this assumption will make the measurement error model more appropriate to the linear regression method for the interval-valued data. The performance of ORCA using the general orthogonal regression method can possibly be improved if this assumption is relaxed.

One possible way to relax the assumption in equation (4.18) is to implement an iteration process. In particular, we describe the process as follows:

(i) Start with the assumption in equation (4.18), calculate the covariance matrix of the measurement errors by equation (4.19). Estimate the linear regression coefficients by equation (4.27) based on the obtained covariance matrix of measurement errors.

(ii) Calculate the true value for each observation by the estimated linear regression coefficients. Denote the calculated true value of an observation x_{ij} as $\ddot{x}_{ij}$; the measurement error for x_{ij} is updated as $u'_{ij} = [x_{ija} - \ddot{x}_{ij}, x_{jib} - \ddot{x}_{ij}]$. Update the covariance matrix for the measurement errors and go back to the step (i).

We can repeat these two steps until certain convergence criterion is satisfied. The criterion can be, for example, the Euclidean distance between the updated estimated true values of all observations and the estimated true values from the previous iteration is smaller than a predetermined value. However, further research is needed to study whether these two steps can eventually converge.

Using the measurement error model to estimate the linear regression coefficients for the interval-valued data provides a new way to make inference about the linear regression coefficients. The asymptotic properties of the linear regression coefficients for measurement error model were discussed in Fuller (2009). Future research can adapt these properties to the measurement error model for interval-valued data for inference.

The measurement error model for interval-valued data provides a maximum likelihood estimation (MLE) to the linear regression coefficients. Analogously with the maximum likelihood methodology for cluster-wise regression proposed by DeSarbo and Cron (1988) for classical data, the maximum likelihood estimation by measurement error model for the interval-valued data can be used to develop the model based cluster-wise regression for interval-valued data.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. and Csáki, F., editors, *Second International Symposium on Information Theory*, pages 267–281.

Anderberg, M. (1973). *Cluster Analysis for Applications*. Probability and Mathematical Statistics. Academic Press.

Anderson, E. (1935). The irises of the gaspe peninsula. *Bulletin of the American Iris society*, 59:2–5.

Banfield, C. and Bassil, L. (1977). A transfer algorithm for nonhierarchical classification. *Applied Statistics*, 26(2):206–210.

Bertrand, P. and Goupil, F. (2000). Descriptive statistics for symbolic data. In Bock, H. and Diday, E., editors, *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*, pages 106–124. Springer.

Billard, L. (2007). Dependencies and variation components of symbolic interval-valued data. In Brito, P., Cucumel, G., Bertrand, P., and de Carvalho, F. A. T., editors, *Selected Contributions in Data Analysis and Classification*, pages 3–12. Springer.

Billard, L. (2008). Sample covariance functions for complex quantitative data. In Mizuta, M.

and Nakano, J., editors, *World Congress, International Association Statistical Computing*, Yokohama, Japan.

Billard, L. and Diday, E. (2000). Regression analysis for interval-valued data. In Kiers, H. A. L., Rassoon, J.-P., Groenen, P. J. F., and Schader, M., editors, *Data Analysis, Classification, and Related Methods*, pages 369–374. Springer.

Billard, L. and Diday, E. (2003). From the statistics of data to the statistics of knowledge: symbolic data analysis. *Journal of the American Statistical Association*, 98(462):470–487.

Billard, L. and Diday, E. (2004). Symbolic data analysis: Definition and examples. Technical report.

Billard, L. and Diday, E. (2006a). *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. John Wiley & Sons, Chichester, Hoboken.

Billard, L. and Diday, E. (2006b). Descriptive statistics for interval-valued observations in the presence of rules. *Computational Statistics*, 21(2):187–210.

Blanco-Fernández, A., Colubi, A., and González-Rodríguez, G. (2013). Linear regression analysis for interval-valued data based on set arithmetic: A review. In Borgelt, C., Gil, M., Sousa, J., and Verleysen, M., editors, *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, pages 19–31. Springer.

Blanco-Fernández, A., Corral, N., and González-Rodríguez, G. (2011). Estimation of a flexible simple linear model for interval data based on set arithmetic. *Computational Statistics and Data Analysis*, 55(9):2568–2578.

Bock, H.-H. (2007). Clustering methods: A history of k-means algorithms. In Brito, P., Bertrand, P., Cucumel, G., and de Carvalho, F. A. T., editors, *Selected Contributions in Data Analysis and Classification*, pages 161–172. Springer.

Bock, H.-H. (2008). Origins and extensions of the k-means algorithm in cluster analysis. *Journal Electronique dHistoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistics*, vol 4.

Chavent, M. (1998). A monothetic clustering method. *Pattern Recognition Letters*, 19(11):989–996.

Chavent, M., de Carvalho, F. A. T., Lechevallier, Y., and Verde, R. (2006). New clustering methods for interval data. *Computational Statistics*, 21(2):211–229.

Chavent, M. and Lechevallier, Y. (2002). Dynamical clustering of interval data: optimization of an adequacy criterion based on hausdorff distance. In Sokolowsky and Bock, H., editors, *Classification, Clustering, and Data Analysis*, pages 53–60. Springer.

Cormack, R. M. (1971). A review of classification. *Journal of the Royal Statistical Society. Series A*, 134(3):321–367.

de Carvalho, F. A. T. (2007). Fuzzy c-means clustering methods for symbolic interval data. *Pattern Recognition Letters*, 28(4):423–437.

de Carvalho, F. A. T., Brito, P., and Bock, H.-H. (2006a). Dynamic clustering for interval data based on L_2 distance. *Computational Statistics*, 21(2):231–250.

de Carvalho, F. A. T., de Souza, R. M. C., Chavent, M., and Lechevallier, Y. (2006b). Adaptive hausdorff distances and dynamic clustering of symbolic interval data. *Pattern Recognition Letters*, 27(3):167–179.

de Carvalho, F. A. T. and Lechevallier, Y. (2009). Partitional clustering algorithms for symbolic interval data based on single adaptive distances. *Pattern Recognition*, 42(7):1223–1236.

de Carvalho, F. A. T., Saporta, G., and Queiroz, D. N. (2010). A clusterwise center and range regression model for interval-valued data. In Lechevallier, Y. and Saporta, G., editors, *Proceedings of COMPSTAT'2010*, pages 461–468. Springer.

de Souza, R. M. C. and de Carvalho, F. A. T. (2004). Clustering of interval data based on city–block distances. *Pattern Recognition Letters*, 25(3):353–365.

de Souza, R. M. C., de Carvalho, F. A. T., Tenório, C. P., and Lechevallier, Y. (2004). Dynamic cluster methods for interval data based on mahalanobis distances. In Banks, D., House, L., McMorris, F., Arabie, P., and Gaul, W., editors, *Classification, Clustering, and Data Mining Applications*, pages 351–360. Springer.

DeSarbo, W. S. and Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, 5(2):249–282.

Diday, E. and Simon, J. (1976). Clustering analysis. In Fu, K. S., editor, *Digital Pattern Recognition*, pages 47–94. Springer.

Domingues, M. A., de Souza, R. M. C., and Cysneiros, F. J. A. (2010). A robust method for linear regression of symbolic interval data. *Pattern Recognition Letters*, 31(13):1991–1996.

Fagundes, R. A., De Souza, R. M. C., and Cysneiros, F. J. A. (2013). Robust regression with application to symbolic interval data. *Engineering Applications of Artificial Intelligence*, 26(1):564–573.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Fuller, W. A. (2009). *Measurement Error Models*. John Wiley & Sons.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2010). A review of robust clustering methods. *Advances in Data Analysis and Classification*, 4(2-3):89–109.

García-Escudero, L. A., Gordaliza, A., San Martin, R., Van Aelst, S., and Zamar, R. (2009). Robust linear clustering. *Journal of the Royal Statistical Society: Series B*, 71(1):301–318.

Gowda, K. C. and Diday, E. (1991a). Symbolic clustering using a new dissimilarity measure. *Pattern Recognition*, 24(6):567–578.

Gowda, K. C. and Diday, E. (1991b). Unsupervised learning through symbolic clustering. *Pattern Recognition Letters*, 12(5):259–264.

Gowda, K. C. and Ravi, T. (1995). Agglomerative clustering of symbolic objects using the concepts of both similarity and dissimilarity. *Pattern Recognition Letters*, 16(6):647–652.

Hennig, C. (1996). *Identifiability of Finite Linear Regression Mixtures*. Citeseer.

Hennig, C. (1999). Models and methods for clusterwise linear regression. In Gaul, W. and Locarek-Junge, H., editors, *Classification in the Information Age*, pages 179–187. Springer.

Kim, J. and Billard, L. (2011). A polythetic clustering process and cluster validity indexes for histogram-valued objects. *Computational Statistics and Data Analysis*, 55(7):2250–2262.

Kim, J. and Billard, L. (2012). Dissimilarity measures and divisive clustering for symbolic multimodal-valued data. *Computational Statistics and Data Analysis*, 56(9):2795–2808.

Kim, J. and Billard, L. (2013). Dissimilarity measures for histogram-valued observations. *Communications in Statistics-Theory and Methods*, 42(2):283–303.

Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*, volume 161. SIAM.

Le-Rademacher, J. and Billard, L. (2011). Likelihood functions and some maximum likelihood estimators for symbolic data. *Journal of Statistical Planning and Inference*, 141(4):1593–1602.

MacQueen (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA.

Neto, L. and de Carvalho, F. A. T. (2008). Centre and range method for fitting a linear regression model to symbolic interval data. *Computational Statistics and Data Analysis*, 52(3):1500–1515.

Neto, L. and de Carvalho, F. A. T. (2010). Constrained linear regression models for symbolic interval-valued variables. *Computational Statistics and Data Analysis*, 54(2):333–347.

Neto, L., de Carvalho, F. A. T., and Freire, E. S. (2005a). Applying constrained linear regression models to predict interval-valued data. In Furbach, U., editor, *KI 2005: Advances in Artificial Intelligence*, pages 92–106. Springer.

Neto, L., de Carvalho, F. A. T., and Tenorio, C. P. (2005b). Univariate and multivariate linear regression methods to predict interval-valued features. In Webb, G. and Yu, X., editors, *AI 2004: Advances in Artificial Intelligence*, pages 526–537. Springer.

Pal, N. R. and Bezdek, J. C. (1997). Correction to on cluster validity for the fuzzy c-means model. *Fuzzy Systems, IEEE Transactions on*, 5(1):152–153.

Qian, G. and Wu, Y. (2011). Estimation and selection in regression clustering. *European Journal of Pure and Applied Mathematics*, 4(4):455–466.

Rao, C. R., Wu, Y., and Shao, Q. (2007). An m-estimation-based procedure for determining the number of regression models in regression clustering. *Advances in Decision Sciences*, 2007. Hindawi Publishing Corporation, 15 pages.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Rousseeuw, P. J. and Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464.

Shao, Q. and Wu, Y. (2005). A consistent procedure for determining the number of clusters in regression clustering. *Journal of Statistical Planning and Inference*, 135(2):461–476.

Späth, H. (1979). Algorithm 39 clusterwise linear regression. *Computing*, 22(4):367–373.

Späth, H. (1981). Correction to algorithm 39: Clusterwise linear regression. *Computing*, 26:275.

Späth, H. (1982). A fast algorithm for clusterwise linear regression. *Computing*, 29(2):175–181.

Sun, Y. and Li, C. (2014). Linear regression for interval-valued data: A new and comprehensive model. *arXiv preprint arXiv:1401.1831*.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*, 63(2):411–423.

Van Aelst, S., Wang, X. S., Zamar, R. H., and Zhu, R. (2006). Linear grouping using orthogonal regression. *Computational Statistics and Data Analysis*, 50(5):1287–1312.

Wedel, M. and Kistemaker, C. (1989). Consumer benefit segmentation using clusterwise linear regression. *International Journal of Research in Marketing*, 6(1):45–59.

Xu, W. (2010). *Symbolic Data Analysis: Interval-Valued Data Regression*. PhD thesis, University of Georgia.

Xu, W. and Billard, L. (2014). A Study of Interval-valued Observations and their Application to Linear Regression. Technical report, Wells Fargo, McLean VA and Department of Statistics, University of Georgia.

Zhang, B. (2003). Regression clustering. In Wu, X., Tuzhilin, A., and Shavlik, J., editors, *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 451–458. IEEE Computer Society.