Symbolic Data
Regression and Clustering Methods

by

Yi Chen

(Under the Direction of Lynne Billard)

Abstract

With the development of computing and internet technology, data sets with stupendously large numbers of observations are more and more common. One technique to handle the big data is to aggregate classical data to symbolic data, like lists, intervals, lists with probabilities and intervals with probabilities (histograms). Building clustering methods for symbolic data has been an active area over the past decade. In this dissertation, we first review regression and clustering methods for interval data. Then, we develop a regression approach to single-factor analysis of variance and implement it in the software R. Finally, the clustering method proposed by Chavent (1998, 2000) is coded and implemented in R and applied to both simulated and practical data. Advantages and disadvantages of using different distances for clustering are also discussed.

Index words: Symbolic data, Interval-valued data, Symbolic covariance method, ANOVA, Divisive monothetic clustering, Hausdorff distance

Symbolic Data
Regression and Clustering Methods

by

Yi Chen

B.S., Shanghai Jiao Tong University, 2007

M.S., Michigan State University, 2009

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

2014

Symbolic Data
Regression and Clustering Methods

by

Yi Chen

Approved:

Major Professors:	Lynne Billard
Committee:	Cheolwoo Park Lynne Seymour T.N. Sriram Xiangrong Yin

Electronic Version Approved:

Julie Coffield
Interim Dean of the Graduate School
The University of Georgia
August 2014

Symbolic Data
Regression and Clustering Methods

Yi Chen

June 15, 2014

Acknowledgments

I would like to highly appreciate my major advisor Dr. Lynne Billard. Dr. Billard has provided tremendous help through my research. Her patient guidance, willingness to help, understanding and encouragement help me finish the graduate research and dissertation writing. She opens the door to research for me.

Also, I want to thank my committee members, Dr. Cheolwoo Park, Dr. Lynne Seymour, Dr. T.N. Sriram and Dr. Xiangrong Yin for their help on reviewing my work and giving valuable suggestions. Moreover, I would like to appreciate the faculty members and staff from the department of statistics. They have provided continuous help and make me feel living in a big family during my years at The University of Georgia.

Last but not least, I want to thank my parents and wife for their support and encouragement all the time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Data we analyze are classical data most of the time. Observations of classical data on p random variables are points in p-dimensional space $\mathcal{R}^p$. For instance, the i^{th} observation from a data set with n observations and p variables $X_1, ..., X_p$ is $(x_{i1}, ..., x_{ip})_{1\times p}$, where $x_{i1}, ..., x_{ip}$ are points in p-dimensional space. In contrast, symbolic data with p variables are p-dimensional hypercubes in $\mathcal{R}^p$. From Billard and Diday (2006), they can be multi-valued, interval-valued, modal multi-valued, modal interval-valued, etc.

The concept of symbolic data was first raised by Diday (1987). Diday (1995), Diday and Emilion (1996, 1998), Diday et al. (1996) and Emilion (1997) set up a mathematical framework for symbolic data. There are three important types of symbolic data: multi-valued data, i.e., list data (e.g., {black, grey}), interval-valued data (e.g., [10, 25]) and modal interval-valued data, i.e., histogram interval-valued data (e.g., {[0, 10), 1/10; [10,20), 7/10; [20,30), 1/5}).

Typically, symbolic data come from two circumstances: the data are collected in a symbolic data format, or the data are classical observations aggregated to become symbolic data. An example of inherently symbolic data is the price of a certain item in an area. Since the price can vary from store to store, the format used to record the price can be either in a list format, e.g., {Price 1,...,Price n}, or in an interval-valued format, e.g., [Lowest price, Highest price]. The other circumstance under which symbolic data occurs is that data sets nowadays are becoming larger and larger, such as data sets from the internet, population data, geographic data, etc. There can be millions or even billions of observations in one data set. Using current methods to perform analyses on these kinds of data sets can be incredibly time consuming, even with modern computers. Moreover, many of them will fail to give useful results (e.g., a paired t-test on a matched-pair sample with millions of observations may always be rejected due to the large sample size, which results in type I error). However, our main interests are usually in features of certain groups rather than each individual, so it is more reasonable to aggregate the original data which results in symbolic data. For example, a merchant may record hundreds of pieces of visiting histories for a single person visiting its website, but the merchant may only be interested in shopping habits of people from different areas. Therefore, all those records related to one area can be aggregated. More examples of symbolic data can be found in Billard and Diday (2006).

This dissertation will mainly focus on interval-valued data, especially regression and clustering methods for interval-valued data. Interval-valued data play an important role among all types of symbolic data. It is the most common form of symbolic data, and the techniques used to analyze it can be generalized to other types of symbolic data more readily. Unlike classical data, interval-valued data have an internal structure which will cause extra variation which must be included when the data are analyzed. In fact, classical data are a special case of interval-valued data by setting the lower bound equal to the upper bound for each observation from interval-valued data.

Billard and Diday (2000), Lima Neto et al. (2004, 2008), de Carvalho et al. (2004), Billard and Diday (2006) and Lima Neto et al. (2005, 2010) proposed several methods to implement linear regression analyses for interval-valued data, but they all used points as classical surrogates only and hence could not capture the internal variance of each interval.

Later, a new method was presented by Xu (2010) using the symbolic sample covariance suggested by Billard (2008). The new method utilized the total variation in the data and was proved, by Xu (2010), to be superior to former methods. However, analysis of variance (ANOVA) methodology for interval-valued data has not been discussed yet; this is needed when we have data with one or more categorical predictor variables and an interval-valued outcome.

Developing clustering methods for symbolic data is another area of interest. Chavent (1998, 2000) introduced a divisive monothetic clustering method for interval data by using Hausdorff (1937) distance. Kim (2009), and Kim and Billard (2011, 2012) developed both the divisive monothetic and divisive polythetic clustering methods for histogram data by extending the Gowda and Diday (1991, 1992) and Ichino and Yaguchi (1994) distances. In this dissertation, more Hausdorff related distances are added for the divisive monothetic clustering method and comparisons between different Hausdorff distances are made by both simulation studies and using the practical data. The method is implemented in R with options being able to choose different Hausdorff distances.

Chapter 2 gives a literature review of symbolic data: the concept and important types of symbolic data are introduced; descriptive statistics of two important types of symbolic data are given; current regression methods for symbolic data are reviewed; necessary concepts for clustering methods for interval-valued data are also reviewed. In Chapter 3, one-way ANOVA for interval-valued data and a regression approach to one-way ANOVA are proposed, and are applied to a real data set. The clustering algorithm and the comparison among different Hausdorff distances are given in Chapter 4. Finally, Chapter 5 discusses the possible future research including more work in ANOVA for interval-valued data and extension of the clustering method to histogram data.

References

Billard, L. (2007). Dependencies and Variation Components of Symbolic Interval-Valued Data. In: *Selected Contributions in Data Analysis and Classification.* Springer-Verlag, Brelin, 3-13.

Billard, L. (2008). Sample Covariance Functions for Complex Quantitative Data. In: *Proceedings, World Conferences International Association of Statistical Computing* 2008, Yokohama, Japan, 157-163.

Billard, L. and Diday, E. (2000). Regression Analysis for Interval-Valued Data. In: *Data Analysis, Classification and Related Methods, Proceedings of the Seventh Conference of the International Federation of Classification Societies(IFCS'00)*, Springer, Belgium, 369-374.

Billard, L. and Diday, E. (2006). *Symbolic Data Analysis: Conceptual Statistics and Data Mining.* John Wiley and Sons, Chichester.

Chavent, M. (1998). A Monothetic Clustering Method. *Pattern Recognition Letters*, 19, 989-996.

Chavent, M. (2000). Criterion-Based Divisive Clustering for Symbolic Data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data.* (Eds. Bock, H.-H. and Diday, E.). Springer-Verlag, Berlin, 299-311.

de Carvalho F.A.T., Lima Neto, E.A. and Tenorio, C.P. (2004). A New Method to Fit a Linear Regression Model for Interval-valued Data. In: *Lecture Notes in Computer Science, KI2004 Advances in Artifical Inteligence.* Springer-Verlag, 295-306.

Diday, E. (1987). Introduction à l'Approche Symbolique en Analyse des Données. *Premières Journées Symbolique-Numérique,* CEREMADE, Université Paris, Dauphine, 21-56.

Diday, E. (1995). Probabilist, Possibilist and Belief Objects for Knowledge Analysis. *Annals of Operations Research*, 55, 227-276.

Diday, E. and Emilion, R. (1996). Lattices and Capacities in Analysis of Probabilist Objects. In: *Studies in Classification* (eds. E. Diday, Y. Lechevallier, and O. Opilz), 13-30.

Diday, E. and Emilion, R. (1998). Capcities and Credibilities in Analysis of Probabilistic Objects by Histograms and Lattices. In: *Data Science, Cllassification, and Related Methods* (eds. C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H.-H. Bock, and Y. Baba), 353-357.

Diday, E., Emilion, R. and Hillali, Y. (1996). Symbolic Data Analysis of Probabilistic Objects by Capacities and Credibilities. *Societea' Italianadi Statistica*, 5-22.

Emilion, R. (1997). Différentiation des Capacitiés. *Comptes Rendus de l'Academie des Sciences - Series I - Mathematics*, 324, 389-392.

Gowda, K.C. and Diday, E. (1991). Symbolic Clustering Using a New Dissimilarity Measure. *Pattern Recognition*, 24, 567-578.

Gowda, K.C. and Diday, E. (1992). Symbolic Clustering Using a New Similarity Measure. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 368-378.

Hausdorff, F. (1937). *Set Theory* (translated into English by Aumann, J. R. 1957). Chelsey, New York.

Ichino, M. and Yaguchi, H. (1994). Generalized Minkowski Metrics for Mixed Feature Type Data Analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 24, 698-708.

Kim, J. (2009). *Dissimilarity Measures for Histogram-Valued Data and Divisive Clustering of Symbolic Objects.* Doctoral Dissertation, University of Georgia.

Kim, J. and Billard, L. (2011). A polythetic clustering process and cluster validity indexes for histogram-valued objects. *Computational Statistics and Data Analysis*, 55, 2250-2262.

Kim, J. and Billard, L. (2012). Dissimilarity Measures and Divisive Clustering for Symbolic Multimodal-Valued Data. *Computational Statistics and Data Analysis*, 56, 2795-2808.

Lima Neto, E.A. 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, 1500-1515.

Lima Neto, E.A 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.

Lima Neto, E.A., de Carvalho F.A.T. and Freire, E.S. (2004). Univariate and Multivariate Linear Regression Methods to Predict Interval-valued Features. In: *Lecture Notes in Computer Science, AI 2004 Advances in Artificial Intelligence.* Springer-Verlag, Berlin, 526-537.

Lima Neto, E.A., de Carvalho F.A.T. and Freire, E.S. (2005). Applying Constrained Linear Aggression Models to predict Interval-Valued Data. In: *Lecture Notes in Computer Science, KI: Advances in Artificial Inteligence* (ed. U. Furbach). Springer-Verlag, Brelin, 92-106.

Xu, W. (2010). *Symbolic Data Analysis: Interval-valued Data Regression.* Doctoral Dissertation, University of Georgia.

Chapter 2

Literature Review

In order to set up some groundwork for later chapters, we need review some of the current literature. In Section 2.1, the definition of symbolic data is introduced along with some examples. Section 2.2 displays four types of symbolic data and their descriptive statistics. In Section 2.3, several currently available regression methods for symbolic data and their advantages and disadvantages are discussed. Some preliminaries for clustering methods for symbolic data are considered in Section 2.4.

2.1 What Are Symbolic Data?

Classical data with p variables are points in a p-dimensional space $\mathcal{R}^p$. For example, for a data set with n observations and p variables $X_1, ..., X_p$, the i^{th} observation is $(x_{i1}, ..., x_{ip})_{1\times p}$, where $x_{i1}, ..., x_{ip}$ are points in p-dimensional space. However, symbolic data with p variables are p-dimensional hypercubes in $\mathcal{R}^p$, or a Cartesian product of p distributions. Lists (e.g., $x_{ij} = \{\text{fair, worse}\}$), intervals (e.g., $x_{ij} = [1.5, 2.5]$) and histograms (e.g., $x_{ij} = \{[1, 3), 0.1; [3, 6], 0.9\}$) are all examples of symbolic data. The concept of symbolic data was first raised by Diday (1987) and since then has been applied to principal component analysis, regression analysis and cluster analysis.

Three Important Types of Symbolic Data

Let us start with three important kinds of symbolic data. Expanded details can be found in Bock and Diday (2000) and Billard and Diday (2006), with a non-technical introduction in Billard (2011).

Definition 2.1.1. *A* ***multi-valued symbolic random variable*** *X takes one or more values from those values' domain $\mathcal{X}$ as a list. The possible number of values in $\mathcal{X}$ is finite, and values can be categorical or quantitative values.*

Definition 2.1.2. *An* ***interval-valued symbolic random variable*** *X takes values from an interval. That is, $X = \xi = [a, b] \subset \mathcal{R}^1$, with $a \leq b, a, b \in \mathcal{R}^1$. The interval can be closed or open at either end, i.e., $(a, b), [a, b], [a, b)$, or $(a, b]$.*

Definition 2.1.3. *Let X be a quantitative random variable that can take values on a finite number of non-overlapping intervals $[a_i, b_i), i = 1, 2, ...,$ with $a_i \leq b_i$. An outcome for observation w_u for a* ***histogram interval-valued random variable*** *takes the form*

$$X(w_u) = \xi_u = \{[a_{ui}, b_{ui}), p_{ui}; i = 1, ..., s_u\}$$

where $s_u < \infty$ is the number of intervals forming the support for the outcome $X(w_u)$ for observation w_u, and p_{ui} is the weight for the particular subinterval $[a_{ui}, b_{ui}), i = 1, ..., s_u$, with $\sum_{k=1}^{s_u} p_{ui} = 1$. The intervals $[a_i, b_i)$ can be open or closed at either end.

Let us look at some examples. Suppose we have a dataset from a high school counseling program. Each row represents a student from that high school. For each student, personal information like Student ID, Gender and Race are recorded. Concerns that a student may talk to his/her counselor may be: College to attend/Career (A), Stress/Anxiety (B), Academic difficulties (C), Family problems (D) and Other (E). The grade point average (GPA)

Table 2.1: Sample Counseling Program Survey Dataset

ID	Gender	Race	Concerns	GPA	...
1	Female	White	A	[3.5,4]	...
2	Male	Asian	A,B,C	[3.1,3.5]	...
3	Male	White	A,C,D	[2.8,3.5]	...
4	Female	Hispanic	A,B,D,E	[2.5,2.9]	...
5	Male	African American	A	[3,3.8]	...
⋮	⋮	⋮	⋮	⋮	⋮

of the student for each term is also recorded. Other variables like age of the student, whether they feel the counselor helpful, their plans after high school, etc., are also of interest. We will not discuss them here.

Table 2.1 is a dataset that contains typical symbolic data. For instance, the variable Concerns has five potential values and may take more than one value from these five values, since a student may have several concerns when consulting with his/her counselor. Another example is GPA. Usually, a school will give the counselor a student's cumulative GPA. However, it will be more informative if a student's GPAs of each term are collected. This will be discussed again in more details later.

Table 2.2: Sample Credit Card Holders Dataset

Name	Gender	Annual Income	Financial Assets	Own House	...
D. Mike	Male	[$12,000, $15,000)	C,S	No	...
J. Susan	Female	[$60,000, $80,000)	C,S,R	Yes	...
L. Charles	Male	[$100,000, $200,000)	C,S,R,O	Yes	...
K. Richard	Female	[$15,000, $20,000)	C	Yes	...
W. John	Female	[$20,000, $40,000)	C,S	No	...
⋮	⋮	⋮	⋮	⋮	⋮

Table 2.2 is another example of symbolic data. When applying for a credit card, a person's financial condition will be asked, such as the total annual income and types of financial assets owned elsewhere. It is always sensitive to ask people's annual incomes and people tend not to tell the exact amount of their incomes, so it will make them more comfortable if several income ranges are provided. Therefore, Annual Income is an interval-valued variable. Also, people usually have more types of financial assets other than just saving or checking accounts. Financial assets include Checking Account (C), Savings Account (S) including Money Market and CDs, Retirement Account (R) and Other Investments (O) including stocks, bonds, brokerage accounts, etc. Financial Assets is a typical list data. Both of them are symbolic data.

The following two examples are from Billard and Diday (2006).

Table 2.3: Sample Mushroom Data

Species	Pileus Cap Width	Stipe Length	Stipe Thickness
arorae	[3.0, 8.0]	[4.0, 9.0]	[0.50, 2.50]
arvenis	[6.0,21.0]	[4.0, 14.0]	[1.00, 3.50]
benesi	[4.0, 8.0]	[5.0, 11.0]	[1.00, 2.00]
bernardii	[6.0, 7.0]	[4.0, 7.0]	[3.00, 4.50]
bisporus	[5.0, 12.0]	[2.0, 5.0]	[1.50, 2.50]
⋮	⋮	⋮	⋮

The data of Table 2.3 provide measurements (in cm) of certain features of some species of mushrooms, like the Width of the Pileus Cap, and the Length and Thickness of the stipe. These measurements are interval valued. For example, the width of the pileus cap for mushrooms of the *arorae* species is from 3.0 to 8.0 cm. Since we are only interested in the species of mushrooms instead of each individual mushroom, it is more reasonable and manageable to aggregate the data of each mushroom into intervals by their species.

The data of Table 2.4 are aggregated from classical data of medical records of individuals

Table 2.4: Health Insurance

Type × Gender	Age	Weight	...
Dental Males	{[0, 40), 4/9; [40, 99], 4/9 }	{[150, 200), 4/9; [200, 275], 5/9}	...
Dental Females	{[0, 40), 1/3; [40, 99], 2/3}	{[140, 160), 2/3; [160, 180], 1/3}	...
Medical Males	{[0, 20), 1/6; [20, 40], 2/89; [40, 60), 2/9; [60,99], 7/18}	{[0, 120), 1/9; [120, 180), 7/8; [180, 240), 1/2;}	...
Medical Females	{[0, 20), 1/7; [20, 40], 2/7; [40, 60), 3/14; [60, 99], 5/14}	{[50, 100), 1/7; [110, 140], 5/14; {[140, 170), 1/2}	...
Optical Males	{[40, 60), 1/2; [60, 99], 1/2 }	{[140, 160), 1/2; [160, 200], 1/2}	...
Optical Females	{[60, 99])}	{[140, 160), 1/2; [160, 180], 1/4}	...

retained by a health insurance company. For each individual, there are usually demographic variables such as gender, marital status, age, information on parents (such as the number alive), siblings, number of children, employer, health provider, etc. Many other variables such as a record of geographical location variables, basic medical variables and other health related variables are recorded. For illustration, only age and weight are shown in Table 2.4. This is a typical histogram interval-valued dataset. For males receiving dental care, 4/9 of them are between age 0 and 40 and 4/9 are between age 40 and 99. Also, 4/9 of them weigh between 150 and 200 pounds and the rest weigh between 200 and 275 pounds.

Comparison With Classical Data

Typically, symbolic data come from two situations: the data are collected in a symbolic data format (lists, intervals, histograms, etc.) or the data are aggregated to symbolic data from classical data during later processing.

The data of Table 2.1 and 2.2 are collected in a symbolic data format. In the past, we analyzed these data by replacing them with surrogate values, e.g., point values. The GPA of student 1 in Table 2.1 will be replaced by the cumulative GPA, 3.75, for example. The first person in Table 2.2 will have the income as \$13,500, which is the midpoint of the range or he will have the income level as 1, which is the categorical data we create for the lowest range. However, by doing so, we lose important information in the original data. The range of the interval and the internal variation of the interval no longer exist. For the counseling program example, the GPAs of each term are collected in the form of lists, or intervals stating the minimum and maximum term GPA since entry to the school. If student M has term GPAs as 3.4, 3.5, 3.5 and 3.6 while student N has term GPAs as 2.0, 4.0, 4.0 and 4.0, they will have the same cumulative GPAs. However, the variation of student N's GPAs is larger than student M's GPAs, which reveals that student N is an excellent student most of the time except for one semester. This may be due to his/her temporary naughtiness, illness, family accident, etc. If it is due to an illness, we can conclude that student N is better than M in academia in spite of their having the same cumulative GPAs.

Again, for the credit card example, the lowest range and the highest range in Table 2.2 are $[\$12,000, \$15,000)$ and $[\$100,000, \$200,000)$, respectively. If they are just replaced by categorical data like 1 and 5, they will be processed equally as points during later analysis. However, it is obvious that the range of $[\$12,000, \$15,000)$ is \$3,000 while the range of $[\$100,000, \$200,000)$ is \$100,00. Both the range and the internal variance of $[\$100,000, \$200,000)$ are larger than those of $[\$12,000, \$15,000)$. This information should be applied into any analysis rather than being ignored.

The second situation is when the number of the original data is so huge that they must be summarized and organized before being analyzed. With the development of computer technology, large datasets are becoming more and more common nowadays. It is too time consuming and usually hard to calculate statistics from large datasets by classical statistical algorithms and methodologies; and can be unnecessary especially when our interests are in units of certain groups instead of each individually observed data. Under these circumstances, large datasets can be aggregated into symbolic data and then be analyzed. The data of Table 2.3 and 2.4 are formed in this way.

2.2 Density Function and Descriptive Statistics

Some definitions and notations need to be introduced to derive density functions and descriptive statistics of symbolic data. Details can be found in Bock and Diday (2000), Billard and Diday (2006). Many of the contents below and in this chapter can be found in Le-Rademacher (2008), Kim (2009) and Xu (2010).

Suppose we have a data matrix $\mathbf{X}$ with n observations and p variables. Let $\mathbf{X}_j$ denote the j^{th} variable and $\mathbf{X}(i)$ denote the i^{th} observation, where $j = 1, ..., p$ and $i = 1, ..., n$. The realization of the j^{th} variable of the i^{th} observation is denoted by x_{ij} for classical data and is denoted by ξ_{ij} for symbolic data.

Definition 2.2.1. *Let $\mathcal{X}_j$ be the domain of $\mathbf{X}_j$. Then the random variables $\mathbf{X}_j, j = 1, ..., p$, have domain $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_p$. Every point $\mathbf{x} = (x_1, ..., x_p)$ in $\mathcal{X}$ is called a **description vector**.*

Definition 2.2.2. *Let the random variables $\mathbf{X}_j, j = 1, ..., p$, have domain $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_p$. The p-dimensional subspace $D = (D_1, ..., D_p) \subseteq \mathcal{X}$ is a **description set**, where $D_j \subseteq \mathcal{X}_j$. If $D = D_1 \times ... \times D_p$ is the Cartesian product of the sets D_j, then D is called a **Cartesian description set**.*

Definition 2.2.3. *The symbolic description of an observation is given by the description vector* $\mathbf{d}$. *If each* D_j *is a set of one value only, i.e.,* $\mathbf{x} = (x_1, ..., x_p) \equiv \mathbf{d} = (\{x_1\}, ..., \{x_p\})$, *then* $\mathbf{x}$ *is called an* ***individual description***.

A logical dependency rule v can be written as

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

for $A \subseteq D, B \subseteq D$, and $x \in \mathcal{X}$ where v is a mapping of $\mathcal{X}$ onto $0, 1$, with $v(x) = 0$ if the rule is not satisfied by x and 1 otherwise. The set of all rules v operating on $\mathcal{X}$ is denoted as $V_{\mathcal{X}}$.

Definition 2.2.4. *The* ***virtual description, vir(d)****, of the description vector* ***d*** *is the set of all individual description vectors* x *that satisfy all the logical dependency rules* v *in* $\mathcal{X}$. *This can be written as*

$$vir(d) = \{x \in D | v(x) = 1 \textit{ for all } v \textit{ in } V_{\mathcal{X}}\}.$$

Multi-Valued Variables

Suppose there are n observations for the random variable X_j where X_j is a multi-valued random variable and ξ is a realization of X_j. If Y is a value in X_j, then the observed frequency of Y taking value ξ is

$$O_Y(\xi) = \sum_{i=1}^{n} \frac{|\{x \in vir(d_i) | x_j = \xi\}|}{|vir(d_i)|} \tag{2.1}$$

where $|A|$ is the number of individual descriptions in the space A. In Equation (2.1), any i for which $vir(d_i)$ is empty is ignored.

Then, the empirical distribution function of Y is

$$F_Y(\xi) = \frac{1}{n'} \sum_{\xi_k \leq \xi} O_Y(\xi_k)$$

where $n' = (n - n_0)$ with n_0 being the number of i for which $|vir(d_i)| = 0$.

When the variable Y is quantitative, the symbolic sample mean is

$$\bar{Y} = \frac{1}{n'} \sum_{\xi_k \in \mathcal{X}_j} \xi_k O_Y(\xi_k),$$

and the symbolic sample variance is

$$S_j^2 = \frac{1}{n'} \sum_{\xi_k \in \mathcal{X}_j} (\xi_k - \bar{Y})^2 O_Y(\xi_k).$$

Interval-Valued Variables

Among all the types of symbolic data, interval-valued symbolic data play an important role. Not only is it the most common type, but also the techniques used to analyze them can be applied to other types of data more readily.

In Section 2.1, it has been mentioned that symbolic data usually arise from two aspects: the data are collected in a symbolic format, or the data are aggregated to symbolic data from classical data. For interval-valued data, if they are collected in a symbolic format, sometimes it is due to its original format (like income range, GPA range) while sometimes it is due to the impossibility of measuring some characteristic accurately as an exact value. Instead, an interval format like $[x - \Delta_1, x + \Delta_2], \Delta_1 \neq \Delta_2$ is usually more reasonable.

Besides the empirical density function, sample mean and sample variance (from Bertrand and Goupil, 2000), the sample covariance for interval-valued data will also be discussed in this section. The sample covariance was obtained by Billard (2007, 2008). All of these definitions are based on the assumption that values across each interval are uniformly distributed.

Again, let X_j denote the j^{th} variable and $X(i)$ denote the i^{th} observation of an interval-valued data sample, where $j = 1, ..., p$ and $i = 1, ..., n$. The realization of the j^{th} variable of the i^{th} observation is denoted by ξ_{ij} taking an interval of values $[a_{ij}, b_{ij}]$, where $a_{ij} \leq b_{ij}$.

We assume Y is uniformly distributed over the interval $X_{ij} = [a_{ij}, b_{ij}]$ for all the individual description vectors $\mathbf{x} \in vir(\mathbf{d_i})$. Then, for each ξ,

$$P\{Y \leq \xi | \mathbf{x} \in vir(\mathbf{d_i})\} = \begin{cases} 0, & \xi < a_{ij}, \\ \frac{\xi - a_{ij}}{b_{ij} - a_{ij}}, & a_{ij} \leq \xi < b_{ij}, \\ 1, & b_{ij} \leq \xi. \end{cases}$$

Also, it is assumed that each observation is equally likely to be observed with probability $1/n$. Then, the empirical distribution function of Y is

$$\begin{aligned} F_Y(\xi) &= \frac{1}{n} \sum_{i=1}^{n} P\{Y \leq \xi | \mathbf{x} \in vir(\mathbf{d_i})\} \\ &= \frac{1}{n} \left\{ \sum_{i:\xi \in \xi_{ij}} (\frac{\xi - a_{ij}}{b_{ij} - a_{ij}} + |(i|\xi \geq b_{ij})|) \right\}. \end{aligned}$$

By taking the derivative with respect to ξ, the empirical density function of Y is

$$f_Y(\xi) = \frac{1}{n} \sum_{i:\xi \in \xi_{ij}} (\frac{1}{b_{ij} - a_{ij}}); \tag{2.2}$$

see Bertrand and Goupil (2000). Also, from Bertrand and Goupil (2000), the symbolic sample mean can be achieved from Equation (2.2) as

$$\begin{aligned}
\bar{X}_j &= \int_{-\infty}^{\infty} \xi f(\xi) d\xi \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{b_{ij}-a_{ij}}\int_{-\infty}^{\infty}\xi d\xi\right] \\
&= \frac{1}{2n}\sum_{i=1}^{n}\left[(\frac{1}{b_{ij}-a_{ij}})(\xi^2|_{a_{ij}}^{b_{ij}})\right] \\
&= \frac{1}{2n}\sum_{i=1}^{n}\frac{b_{ij}^2-a_{ij}^2}{b_{ij}-a_{ij}} \\
&= \frac{1}{2n}\sum_{i=1}^{n}(b_{ij}+a_{ij}). \qquad (2.3)
\end{aligned}$$

The symbolic sample variance is

$$\begin{aligned}
S_j^2 &= \int_{-\infty}^{\infty}(\xi-\bar{X}_j)^2 f(\xi)d\xi \\
&= \int_{-\infty}^{\infty}\xi^2 f(\xi)d\xi - \bar{X}_j{}^2. \qquad (2.4)
\end{aligned}$$

Replacing $f(\xi)$ in Equation (2.4) with Equation (2.2), we have

$$\begin{aligned}
\int_{-\infty}^{\infty}\xi^2 f(\xi)d\xi &= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{b_{ij}-a_{ij}}\int_{-\infty}^{\infty}\xi^2 d\xi\right] \\
&= \frac{1}{3n}\sum_{i=1}^{n}\left[\frac{1}{b_{ij}-aij}\xi^3|_{a_{ij}}^{bij}\right] \\
&= \frac{1}{3n}\sum_{i=1}^{n}\left[\frac{b_{ij}^3-a_{ij}^3}{b_{ij}-a_{ij}}\right] \\
&= \frac{1}{3n}\sum_{i=1}^{n}(a_{ij}^2+a_{ij}b_{ij}+b_{ij}^2). \qquad (2.5)
\end{aligned}$$

Substituting Equations (2.5) and (2.3) into Equation (2.4), we have

$$\begin{aligned} S_j^2 &= \frac{1}{3n}\sum_{i=1}^{n}(a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{1}{4n^2}\left[\sum_{i=1}^{n}(a_{ij} + b_{ij})\right]^2 \\ &= \frac{1}{3n}\sum_{i=1}^{n}(a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \bar{X}_j^2. \end{aligned} \tag{2.6}$$

Note that when $a_{ij} = b_{ij} = x_{ij}$, Equation (2.6) is equal to that of classical sample variance, i.e.,

$$S_j^2 = \frac{\sum_{i=1}^{n} x_{ij}^2}{n} - \left(\frac{\sum_{i=1}^{n} x_{ij}}{n}\right)^2.$$

In Billard (2007, 2008), Total Sum of Squares (TotalSS) is further divided into Within Sum of Squares (WithinSS) and Between Sum of Square (BetweenSS) terms. That is,

$$TotalSS = WithinSS + BetweenSS; \tag{2.7}$$

the Total Sum of Squares, $TotalSS$, is simply

$$TotalSS = nS_j^2.$$

The S_j^2 in Equation (2.6) can be rewritten as

$$S_j^2 = \frac{1}{3n}\sum_{i=1}^{n}[(a_{ij} - \bar{X}_j)^2 + (a_{ij} - \bar{X}_j)(b_{ij} - \bar{X}_j) + (b_{ij} - \bar{X}_j)^2]. \tag{2.8}$$

When $n = 1$, Equation (2.8) is

$$S_j^2(n = 1) = \frac{1}{3}[(a_{ij} - \bar{X}_j)^2 + (a_{ij} - \bar{X}_j)(b_{ij} - \bar{X}_j) + (b_{ij} - \bar{X}_j)^2]. \tag{2.9}$$

Then, Equation (2.9) is the internal variation of a single observation X_{ij}. By summing over

$i = 1, ..., n$, the overall Within Sum of Squares can be obtained as

$$WithinSS = \frac{1}{3}\sum_{i=1}^{n}[(a_{ij} - \bar{X}_j)^2 + (a_{ij} - \bar{X}_j)(b_{ij} - \bar{X}_j) + (b_{ij} - \bar{X}_j)^2]. \tag{2.10}$$

Substituting Equation (2.3) into Equation (2.10), we have

$$WithinSS = \frac{1}{12}\sum_{i=1}^{n}(b_{ij} - a_{ij})^2. \tag{2.11}$$

The Between Sum of Squares is the variation between observations, i.e., the variation of interval midpoints, i.e., we have

$$BetweenSS = \sum_{i=1}^{n}[\frac{1}{2}(a_{ij} + b_{ij}) - \bar{X}_j]^2 = \sum_{i=1}^{n}(\bar{X}_{ij} - \bar{X}_j)^2. \tag{2.12}$$

Therefore, the result of Equation (2.7) follows by substituting Equations (2.11) and (2.12) into Equation (2.7).

Note that the $WithinSS$ is the sum of n variations of a uniform distribution, which coincides with the assumption that Y is uniformly distributed over the interval $X_{ij} = [a_{ij}, b_{ij}]$.

Similarly, the covariance function between two interval-valued symbolic variables can be achieved. Without loss of generality, choose (X_1, X_2) as a pair of random variables from the sample data. The cross-variation between observations (X_{i1}, X_{i2}), $i = 1, ..., n$, i.e., the Sum of Products (SP) is

$$WithinSP = \sum_{i=1}^{n}\frac{(b_{i1} - a_{i1})(b_{i2} - a_{i2})}{12}$$

and

$$BetweenSP = \sum_{i=1}^{n}\left(\frac{a_{i1} + b_{i1}}{2} - \bar{X}_1\right)\left(\frac{a_{i2} + b_{i2}}{2} - \bar{X}_2\right)$$

where $\bar{X}_1$ and $\bar{X}_2$ can be found by Equation (2.3). Therefore,

$$\begin{aligned} TotalSP &= WithinSP + BetweenSP \\ &= \frac{1}{6}\sum_{i=1}^{n}[2(a_{i1}-\bar{X}_1)(a_{i2}-\bar{X}_2)+(a_{i1}-\bar{X}_1)(b_{i2}-\bar{X}_2) \\ &\quad +(b_{i1}-\bar{X}_1)(a_{i2}-\bar{X}_2)+2(b_{i1}-\bar{X}_1)(b_{i2}-\bar{X}_2)]. \end{aligned} \tag{2.13}$$

The covariance function between X_{i1} and X_{i2} is

$$Cov(X_1, X_2) = TotalSP/n. \tag{2.14}$$

It is easy to verify that when $X_1 = X_2$, i.e, $a_{i1} = a_{i2}$ and $b_{i1} = b_{i2}$, Equation (2.14) is equal to $S_1^2 = S_2^2$, which can be derived from Equation (2.6). Also, when $a_{i1} = b_{i1}$ and $a_{i2} = b_{i2}$, Equation (2.14) becomes $\frac{1}{n}\sum_{i=1}^{n}(a_{i1}-\bar{X}_1)(a_{i2}-\bar{X}_2)$, which is the covariance function for classical data.

After obtaining the variance and covariance, it is obvious that the correlation coefficient between X_1 and X_2 is

$$Corr(X_1, X_2) = \frac{Cov(X_1, X_2)}{S_1 S_2}.$$

2.3 Regression Methods for Symbolic Data

Several methods have been proposed to implement linear regression analyses for interval-valued symbolic data. The first method was raised by Billard and Diday (2000) using centers of intervals when fitting regression models (CM), and parameter estimates from models using centers were used to calculate the prediction interval when a new observation is available. Then, a center and range method (CRM) (Lima Neto et al., 2004, de Carvalho et al., 2004 and Lima Neto and de Carvalho, 2008) utilized not only centers but also ranges of intervals to fit regression models. Centers and ranges are used separately to do the fitting. Later on, Billard and Diday (2006) improved the CRM by fitting centers and ranges simultaneously as a bivariate model either without interaction (BCRMO) or with interaction (BCRMI).

Center Method

The first approach to fit a linear regression model to interval-valued data was introduced by Billard and Diday (2000). They obtained the centers of each interval and fitted regression models using the centers as for a classical method. After obtaining parameter estimates, they applied the fitted model to both lower and upper bounds of a new observation to achieve an interval predicted response. This approach is called the center method (CM).

Let X_j denote the j^{th} variable of an interval-valued data sample, let $\mathbf{X}(i)$ denote the i^{th} observation, and let Y be the response, where $i = 1, ..., n$, and $j = 1, ..., p$. The i^{th} observed value of X_j is $X_{ij} = [a_{ij}, b_{ij}]$ and the i^{th} observed value of Y is $Y_i = [c_i, d_i]$. Hence,

$$X^c_{ij} = \frac{a_{ij} + b_{ij}}{2}, Y^c_i = \frac{c_i + d_i}{2}.$$

The regression model is

$$\mathbf{Y}^c = \mathbf{X}^c \boldsymbol{\beta}^c + \boldsymbol{\epsilon}^c$$

where $\mathbf{X}^c = (\mathbf{X}^c(1), ..., \mathbf{X}^c(n))'$, $\mathbf{Y}^c = (Y^c_1, ..., Y^c_n)'$, $\boldsymbol{\beta}^c = (\beta^c_0, \beta^c_1, ..., \beta^c_p)'$, $\boldsymbol{\epsilon}^c = (\epsilon^r_1, ..., \epsilon^r_n)'$ and $\mathbf{X}^c(i) = (1, X^c_{i1}, ..., X^c_{ip})$ for $i = 1, ..., n$.

Now, the least squares estimator of $\boldsymbol{\beta}^c$ is achieved by using the classical method, as

$$\hat{\boldsymbol{\beta}}^c = ((\mathbf{X}^c)'\mathbf{X}^c)^{-1}(\mathbf{X}^c)'\mathbf{Y}^c.$$

Then, for a given $\mathbf{X}^* = (1, [a^*_1, b^*_1], ..., [a^*_p, b^*_p])$, the predicted value of Y is $\hat{Y}^* = [\hat{Y}^*_a, \hat{Y}^*_b]$

where

$$\hat{Y}_a^* = \hat{\beta}_0^c + \hat{\beta}_1^c a_1^* + \cdots + \hat{\beta}_p^c a_p^*, \quad \hat{Y}_b^* = \hat{\beta}_0^c + \hat{\beta}_1^c b_1^* + \cdots + \hat{\beta}_p^c b_p^*.$$

Center and Range Method

Lima Neto et al. (2004, 2008) and de Carvalho et al. (2004) put forward the center and range method (CRM) to estimate parameter $\boldsymbol{\beta}$ using both centers and ranges of intervals separately. After having parameter estimates for both centers and ranges, these estimates can be applied to the calculation of predicted response when given a new observation.

The regression model on centers is the same as CM's,

$$\mathbf{Y}^c = \mathbf{X}^c \boldsymbol{\beta}^c + \boldsymbol{\epsilon}^c,$$

where $\mathbf{X}^c$, $\mathbf{Y}^c$, $\boldsymbol{\beta}^c$ and $\boldsymbol{\epsilon}^c$ are defined in Section 2.3.

In addition to building a model on centers of intervals, CRM also builds a model on ranges of intervals. Let $X_{ij}^r = b_{ij} - a_{ij}$ and $Y_{ij}^r = d_i - c_i$ be the ranges of the interval-valued data, where $i = 1, ..., n, j = 1, ..., p$.

The regression model on ranges is

$$\mathbf{Y}^r = \mathbf{X}^r \boldsymbol{\beta}^r + \boldsymbol{\epsilon}^r$$

where $\mathbf{X}^r = (\mathbf{X}^r(1), ..., \mathbf{X}^r(n))'$, $\mathbf{Y}^r = (Y_1^r, ..., Y_n^r)'$, $\boldsymbol{\beta}^r = (\beta_0^r, \beta_1^r, ..., \beta_p^r)'$, $\boldsymbol{\epsilon}^r = (\epsilon_1^r, ..., \epsilon_n^r)'$ and $\mathbf{X}^r(i) = (1, X_{i1}^r, ..., X_{ip}^r)$ for $i = 1, ..., n$. Hence, the least squares estimators of $\boldsymbol{\beta}^c$ and $\boldsymbol{\beta}^r$ are, respectively,

$$\hat{\boldsymbol{\beta}}^c = ((\mathbf{X}^c)'\mathbf{X}^c)^{-1}(\mathbf{X}^c)'\mathbf{Y}^c, \quad \hat{\boldsymbol{\beta}}^r = ((\mathbf{X}^r)'\mathbf{X}^r)^{-1}(\mathbf{X}^r)'\mathbf{Y}^r.$$

For a given $\mathbf{X}^* = (1, [a_1^*, b_1^*], ..., [a_p^*, b_p^*])$, the predicted value of Y is $\hat{Y}^* = [\hat{Y}_a^*, \hat{Y}_b^*]$, where

$$\hat{Y}_a^* = \hat{Y}^{c*} - \frac{\hat{Y}^{r*}}{2}, \quad \hat{Y}_b^* = \hat{Y}^{c*} + \frac{\hat{Y}^{r*}}{2}, \tag{2.15}$$

with $\hat{Y}^{c*} = \hat{\beta}_0^c + \hat{\beta}_1^c(\frac{a_1^*+b_1^*}{2}) + \cdots + \hat{\beta}_p^c(\frac{a_p^*+b_p^*}{2})$, $\hat{Y}^{r*} = \hat{\beta}_0^r + \hat{\beta}_1^r(b_1^* - a_1^*) + \cdots + \hat{\beta}_p^r(b_p^* - a_p^*)$.

Bivariate Center and Range Method

The CRM assumes that centers and ranges are independent and fits models on them separately. In order to break this assumption, Billard and Diday (2006) fitted centers and ranges simultaneously as a bivariate model, either with (BCRMI) or without (BCRMO) interaction terms between the center and range variables.

The BCRMO gives the model

$$\begin{pmatrix} \mathbf{Y}^c \\ \mathbf{Y}^r \end{pmatrix} = \begin{pmatrix} \mathbf{X}^{cr}\boldsymbol{\beta}^c + \boldsymbol{\epsilon}^c \\ \mathbf{X}^{cr}\boldsymbol{\beta}^r + \boldsymbol{\epsilon}^r \end{pmatrix} \tag{2.16}$$

where $\mathbf{X}^{cr} = (\mathbf{X}_1^{cr}, ..., \mathbf{X}_n^{cr})'$, $\mathbf{X}_i^{cr} = (1, X_{i1}^c, ..., X_{ip}^c, X_{i1}^r, ..., X_{ip}^r)$ for $i = 1, ..., n$. The $\mathbf{X}^c$, $\mathbf{Y}^c$, $\boldsymbol{\epsilon}^c$, $\mathbf{X}^r$, $\mathbf{Y}^r$ and $\boldsymbol{\epsilon}^r$ are defined in Section 2.3 and Section 2.3. The least squares estimators of $\boldsymbol{\beta}^c$ and $\boldsymbol{\beta}^r$ are

$$\hat{\boldsymbol{\beta}}^c = (\mathbf{X}^{cr'}\mathbf{X}^{cr})^{-1}\mathbf{X}^{cr'}\mathbf{Y}^c, \hat{\boldsymbol{\beta}}^r = (\mathbf{X}^{cr'}\mathbf{X}^{cr})^{-1}\mathbf{X}^{cr'}\mathbf{Y}^r. \tag{2.17}$$

Then, the predicted response for a given $\mathbf{X}^{cr*} = (1, X_1^{c*}, ..., X_p^{c*}, X_1^{r*}, ..., X_p^{r*})'$ is $\hat{Y}^* = [\hat{Y}_a^*, \hat{Y}_b^*]$ as defined in Equation (2.15), but now

$$\begin{pmatrix} \hat{\mathbf{Y}}^{c*} \\ \hat{\mathbf{Y}}^{r*} \end{pmatrix} = \begin{pmatrix} \mathbf{X}^{cr*}\hat{\boldsymbol{\beta}}^c \\ \mathbf{X}^{cr*}\hat{\boldsymbol{\beta}}^r \end{pmatrix}. \tag{2.18}$$

The BCRMI adds interaction terms between the center and range variables into the model. The model given by BCRMI is the same as Equation (2.16) except

$$\mathbf{X}_i^{cr} = (1, X_{i1}^c, ..., X_{ip}^c, X_{i1}^r, ..., X_{ip}^r, X_{i1}^c \times X_{i1}^r, ..., X_{ip}^c \times X_{ip}^r) \text{ for } i = 1, ..., n.$$

Similarly, parameter estimates and predicted response given a new observation can be derived from Equations (2.17) and (2.18), respectively.

It is obvious that the CRM and BCRM methods have some improvements by taking into account both centers and ranges of interval-valued data while the CM method concerns center points only. However, there are two main problems with both of these methods. First, these methods are essentially the same as the classical method by converting intervals to points and fitting the model afterward. They can not fully capture the internal variance of each interval using range only, hence they lose information. Secondly, there is no reason that the predicted response of a range is guaranteed to be positive. If a negative predicted range is achieved, the lower bound will be bigger than the upper bound of the predicted response.

Constrained Method

In order to solve the second problem mentioned at the end of Section 2.3, Lima Neto et al. (2005, 2010) suggested the constrained method. The constrained method includes the constrained center method (CONCM) and the constrained center and range method (CONCRM). The regression models are the same as those in Section 2.3 and Section 2.3. The only difference is that constraints $\beta_j^c, \beta_j^r \geq 0,\ j = 0, 1, ..., p$, were added into the models.

To ensure the positiveness of least squares estimates of the parameters $\boldsymbol{\beta}$, Lima Neto et al. (2005, 2010) used an algorithm introduced by Lawson and Hanson (1974). The algorithm identifies the values which coincide with the constraints and changes them to non-negative values using a re-weighting process. However, the fact that all parameters are compelled to be non-negative can not always reveal the nature of data and will produce inaccurate estimates of the regression parameters, if the true $\beta_j < 0$.

Symbolic Covariance Method

A new method was proposed by Xu (2010) describing a symbolic covariance method (SCM) to address the two main issues mentioned in Section 2.3. This new method is based on the symbolic sample covariance in Equation (2.14) suggested by Billard (2008). Due to the use of the symbolic sample covariance, the new method can utilize all the variations in the data. Also, the new method suggested a min/max step to address the issue of a lower bound being bigger than an upper bound in predictions.

We first consider the classical case where there are p predictor variables $X_1, ..., X_p$. The regression model is

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i, \ i = 1, ..., n, \tag{2.19}$$

where Y_i is the dependent response, $\mathbf{X}(i) = (X_{i1}, ..., X_{ip})'$ are predictor variables for the i^{th} observation, β_0 is the intercept parameter, $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$ are regression parameters and ϵ_i are independent error terms following $N(0, \sigma^2)$. The intercept parameter in Equation (2.19) is

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}_1 - \cdots - \hat{\beta}_p \bar{X}_p \tag{2.20}$$

where $\bar{Y}$ and $\bar{X}_j$, $j = 1, ..., p$, are the sample means of Y_i, $i = 1, ..., n$, and X_{ij}, respectively.

Equation (2.19) can be centered as

$$Y_i - \bar{Y} = \beta_1 (X_{i1} - \bar{X}_1) + \cdots + \beta_p (X_{ip} - \bar{X}_p) + \epsilon_i, \ i = 1, ..., n.$$

Therefore, the least squares estimator of $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = ((\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{X} - \bar{\mathbf{X}}))^{-1}(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{Y} - \bar{\mathbf{Y}}) \tag{2.21}$$

where $\mathbf{X} = (\mathbf{X}_1, ..., \mathbf{X}_p)_{n\times p}$, $\mathbf{Y} = (Y_1, ..., Y_n)'$, $\bar{\mathbf{X}} = (1, ..., 1)'_{1\times n}(\bar{X}_1, ..., \bar{X}_p)_{1\times p}$ and $\bar{\mathbf{Y}} = (1, ..., 1)'_{1\times n}\bar{Y}$.

It is easy to show that

$$\begin{aligned}(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{X} - \bar{\mathbf{X}}) &= \begin{pmatrix} \sum_{i=1}^{n}(X_{1i} - \bar{X}_1)^2 & \cdots & \sum_{i=1}^{n}(X_{1i} - \bar{X}_1)(X_{pi} - \bar{X}_p) \\ \vdots & \vdots & \vdots \\ \sum_{i=1}^{n}(X_{pi} - \bar{X}_p)(X_{1i} - \bar{X}_1) & \cdots & \sum_{i=1}^{n}(X_{pi} - \bar{X}_p)^2 \end{pmatrix}_{p\times p} \\ &= \left(\sum_{i=1}^{n}(X_{j_1 i} - \bar{X}_{j_1})(X_{j_2 i} - \bar{X}_{j_2})\right)_{p\times p} \\ &= (n \times Cov(X_{j_1}, X_{j_2}))_{p\times p}, \quad j_1, j_2 = 1, ..., p,\end{aligned}$$

and

$$\begin{aligned}(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{Y} - \bar{\mathbf{Y}}) &= \left(\sum_{i=1}^{n}(X_{ji} - \bar{X}_j)(Y_i - \bar{Y})\right)_{p\times 1} \\ &= (n \times Cov(X_j, Y))_{p\times 1}, \quad j = 1, ..., p.\end{aligned}$$

Hence, Equation (2.21) is equivalent to

$$\hat{\boldsymbol{\beta}} = (n \times Cov(X_{j_1}, X_{j_2}))^{-1}_{p\times p} \times (n \times Cov(X_j, Y))_{p\times 1} \tag{2.22}$$

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

Under the situation of classical data, $\hat{\boldsymbol{\beta}}$ can be obtained by putting data into Equation

(2.21). However, under the situation of interval-valued data, we have to use the symbolic covariance function given in Equations (2.13)-(2.14) in Section 2.2. Again, suppose the observed realizations of (Y, X_j) are the intervals $Y_i = [c_i, d_i]$ and $X_{ij} = [a_{ij}, b_{ij}]$, $i = 1, ..., n$, $j = 1, ..., p$. Then, the symbolic covariance function from Billard (2008) is

$$\begin{aligned} Cov(X_{j1}, X_{j2}) &= \frac{1}{6n}\sum_{i=1}^{n}[2(a_{ij1} - \bar{X}_{j1})(a_{ij2} - \bar{X}_{j2}) + (a_{ij1} - \bar{X}_{j1})(b_{ij2} - \bar{X}_{j2}) \\ &\quad +(b_{ij1} - \bar{X}_{j1})(a_{ij2} - \bar{X}_{j2}) + 2(b_{ij1} - \bar{X}_{j1})(b_{ij2} - \bar{X}_{j2})] \qquad (2.23) \end{aligned}$$

and

$$\begin{aligned} Cov(Y, X_j) &= \frac{1}{6n}\sum_{i=1}^{n}[2(c_i - \bar{Y})(a_{ij} - \bar{X}_j) + (c_i - \bar{Y})(b_{ij} - \bar{X}_j) \\ &\quad +(d_i - \bar{Y})(a_{ij} - \bar{X}_j) + 2(d_i - \bar{Y})(b_{ij} - \bar{X}_j)] \qquad (2.24) \end{aligned}$$

where

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n}(c_i + d_i)/2, \quad \bar{X}_j = \frac{1}{n}\sum_{i=1}^{n}(a_{ij} + b_{ij})/2, \quad j_1, j_2, j = 1, ..., p.$$

Substituting Equation (2.23) and Equation (2.24) into Equation (2.22), we can obtain the estimators of $\boldsymbol{\beta}$; and substituting $\hat{\boldsymbol{\beta}}$ into Equation (2.20), we can obtain the estimator of β_0.

When given a predictor $\mathbf{X}^* = ([a_1^*, b_1^*], ..., [a_p^*, b_p^*])$, the predicted response is

$$\hat{Y} = \hat{\beta}_0 + \hat{\boldsymbol{\beta}}(\mathbf{X}^*)'.$$

Suppose $\hat{Y} = [\hat{Y}_a, \hat{Y}_b]$, $\hat{Y}_a \leq \hat{Y}_b$, it is easy to show that

$$\hat{Y}_a = \min_{X \in \mathcal{X}} \left(\hat{\beta}_0 + \hat{\boldsymbol{\beta}}(\mathbf{X}^*)'\right), \quad \hat{Y}_b = \max_{X \in \mathcal{X}} \left(\hat{\beta}_0 + \hat{\boldsymbol{\beta}}(\mathbf{X}^*)'\right) \qquad (2.25)$$

where $\mathcal{X} = \{a_j^* \leq x_j \leq b_j^*, \quad j = 1, ..., p\}$.

The new symbolic covariance method utilizes the total variation in the data and its performance is superior to classical methods (CM, CRM, BCRM) according to the simulation study by Xu (2010). Also, the assumption that observations within an interval are uniformly distributed across that interval can not always hold in real life. If the assumption is broken, it is very likely that classical methods will meet even greater problems. Therefore, the SCM method is preferred and the research in Chapter 3 of this dissertation is based on this method.

2.4 Preliminary for Clustering Methods for Interval-Valued Data

One technique used to analyze data sets is to partition observations into r clusters so that observations within one group are as homogeneous as possible while observations between groups are as heterogeneous as possible. Gowda and Diday (1991, 1992) suggested dissimilarity measures for multi-valued and interval-valued data; Ichino and Yaguchi (1994) proposed dissimilarity measures for multi-valued and interval-valued data with extensions to Minkowski distances; De Carvalho (1994, 1998) gave dissimilarity measures for interval-valued data by extending the Ichino-Yaguchi measure; Hausdorff (1937) distance for interval-valued data was used by Chavent (2000) and Billard and Diday (2006) for clustering analysis.

Williams and Lambert (1959) and Lance and Williams (1968) introduced divisive monothetic clustering methods for classical binary data. A divisive monothetic clustering method for symbolic data was first developed by Chavent (1998, 2000) for interval data. Kim (2009, 2012) extended the divisive monothetic clustering method to histogram-valued data. This dissertation focuses on the hierarchical divisive monothetic clustering method for interval-valued data.

Dissimilarity and Distance Measures

More details and examples of the contents in Section 2.4 can be found in Billard and Diday (2006) and Kim (2009). Let X_j denote the j^{th} variable and $\mathbf{X}(i)$ denote the i^{th} observation of an interval-valued data sample, where $j = 1, ..., p$ and $i = 1, ..., n$. The realization of the j^{th} variable of the i^{th} observation is denoted by ξ_{ij} taking an interval of values $[a_{ij}, b_{ij}]$, where $a_{ij} \leq b_{ij}$. We assume the X values are uniformly distributed over the interval $X_{ij} = [a_{ij}, b_{ij}]$.

Definition 2.4.1. *If $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$ are two observations in Ω where $\Omega = \{\mathbf{X}(1), ..., \mathbf{X}(n)\}$, $i_1, i_2 = 1, ..., n$, a* ***dissimilarity measure*** *between $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$, $d(\mathbf{X}(i_1), \mathbf{X}(i_2))$, is a measure satisfying the following conditions:*

1. *$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = d(\mathbf{X}(i_2), \mathbf{X}(i_1))$;*

2. *$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) > d(\mathbf{X}(i_1), \mathbf{X}(i_1)) = d(\mathbf{X}(i_2), \mathbf{X}(i_2))$ for all $\mathbf{X}(i_1) \neq \mathbf{X}(i_2)$;*

3. *$d(\mathbf{X}(i_1), \mathbf{X}(i_1)) = 0$ for all $\mathbf{X}(i_1) \in \Omega$.*

Condition 1 tells us that dissimilarity measures are symmetric, conditions 2 and 3 show that a dissimilarity measure between two different observations is positive while it is zero between the same two observations.

Definition 2.4.2. *A* ***distance measure*** *is a dissimilarity measure as defined in Definition 2.4.1 and further satisfies the following conditions:*

4. *If $d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = 0$, then $\mathbf{X}(i_1) = \mathbf{X}(i_2)$;*

5. *$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) \leq d(\mathbf{X}(i_1), \mathbf{X}(i_3)) + d(\mathbf{X}(i_3), \mathbf{X}(i_1))$ for all $\mathbf{X}(i_1), \mathbf{X}(i_2), \mathbf{X}(i_3) \in \Omega$.*

Condition 5 is called the triangular inequality.

Definition 2.4.3. *The* ***Hausdorff distance*** *between two interval-valued observations $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$, $i_1, i_2 = 1, ..., n$, for the variable X_j is*

$$d_j(\mathbf{X}(i_1), \mathbf{X}(i_2)) = max\{|a_{i_1j} - a_{i_2j}|, |b_{i_1j} - b_{i_2j}|\}, \quad j = 1, ..., p. \tag{2.26}$$

Definition 2.4.4. *The* ***Euclidean Hausdorff distance*** *between two interval-valued observations $X(i_1)$ and $X(i_2)$, $i_1, i_2 = 1, ..., n$, is*

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \{\sum_{j=1}^{p}[d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))]^2\}^{1/2} \tag{2.27}$$

where $d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))$, $j = 1, ..., p$, are the Hausdorff distances defined in Equation (2.26).

Definition 2.4.5. *The* ***Span Normalized Euclidean Hausdorff distance*** *between two interval-valued observations $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$, $i_1, i_2 = 1, ..., n$, is*

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{\sum_{j=1}^{p}\left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_j|}\right]^2\right\}^{1/2} \tag{2.28}$$

where the span is $|\mathcal{Y}_j| = max_i(b_{ij}) - min_i(a_{ij})$ and $d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))$ is the Hausdorff distance defined in Equation (2.26).

The Span Normalized Euclidean Hausdorff distance is based on the length of the maximum deviation. This is also called the span normalization.

Definition 2.4.6. *The* ***Normalized Euclidean Hausdorff distance*** *between two interval-valued observations $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$, $i_1, i_2 = 1, ..., n$, is*

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{\sum_{j=1}^{p}\left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_j}\right]^2\right\}^{1/2} \tag{2.29}$$

where

$$H_j^2 = \frac{1}{2n^2}\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}[d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))]^2, \quad j = 1, ..., p,$$

and where $d_j(X(i_1), X(i_2))$ is the Hausdorff distance defined in Equation (2.26).

If the data are classical, then the Normalized Euclidean Hausdorff distance is equivalent to a Normalized Euclidean distance on $\mathcal{R}^2$, with H_j corresponding to the standard deviation of X_j. This is also called the dispersion normalization.

Suppose we have interval-valued observations $\mathbf{X}(i)$, $i = 1, ..., n$, $\mathbf{X}(i) \in \Omega$, where $\Omega = \{\mathbf{X}(1), ..., \mathbf{X}(n)\}$.

Definition 2.4.7. *A **partition** of Ω is a set of subsets $\{C_1, ..., C_r\}$ that satisfies*

1. *$C_u \cap C_v = \phi$, for all $u \neq v = 1, ..., r$;*

2. *$\bigcup_{u=1}^{r} C_u = \Omega$.*

*The subsets of a partition are also called **clusters**.*

Definition 2.4.8. *A **hierarchy** on Ω is a set of subsets $H = \{C_1, ..., C_r\}$ that satisfies*

1. *$\Omega \in H$;*

2. *for all single observations $\mathbf{X}(u)$ in Ω, $\{\mathbf{X}(u)\} \in H$;*

3. *$C_u \bigcap C_v \in \{\phi, C_u, C_v\}$ for all $u \neq v = 1, ..., r$.*

Condition 3 tells us that either two clusters are disjoint, or one is contained in the other.

Definition 2.4.9. ***Divisive clustering*** *is a top-down clustering process. It divides the entire dataset into as many clusters as necessary to produce the hierarchy $H = \{C_1, ..., C_r\}$.*

Clustering Criteria

This hierarchical divisive clustering criteria can also be found in Billard and Diday (2006). Suppose we have n interval-valued observations $\mathbf{X}(i) \in \Omega = \{\mathbf{X}(1), ..., \mathbf{X}(n)\}$, with $\mathbf{X}(i) = (X_{i1}, ..., X_{ip})$, $i = 1, ..., n$. Let C_u^r be the u^{th} cluster from the r^{th} stage of clustering. In the first stage, we start by dividing $C_1^1 = \Omega$ into two clusters (C_1^2, C_2^2). In the second stage, we divide one of these two clusters, e.g., C_1^2, into two sub-clusters, (C_1^3, C_2^3), hence producing three clusters for the third stage. The third cluster in the third stage, C_3^3, inherits from C_2^2

in the second stage. This procedure can be followed as many times as necessary until there is only one observation left in each cluster. Let $P_r = \{C_1^r, ..., C_r^r\}$ represent the partition of Ω at the r^{th} stage.

Definition 2.4.10. *Suppose we have the cluster $C_u^r = \{\mathbf{X}(1)^{ru}, ..., \mathbf{X}(n_{ru})^{ru}\}$, where n_{ru} is the number of observations in the u^{th} cluster from P_r. Then, the* ***within-cluster variation*** *$I(C_u^r)$ is given by*

$$I(C_u^r) = \frac{1}{2\lambda} \sum_{i_1=1}^{n_{ru}} \sum_{i_2=1}^{n_{ru}} w_{i_1} w_{i_2} d^2(i_1, i_2) \equiv \frac{1}{\lambda} \sum_{i_1<i_2=2}^{n_{ru}} \sum_{i_2=1}^{n_{ru}} w_{i_1} w_{i_2} d^2(i_1, i_2) \qquad (2.30)$$

where $d^2(i_1, i_2)$ is a distance or dissimilarity measure between the observations $\mathbf{X}(i_1)^{ru}$ and $\mathbf{X}(i_2)^{ru}$ in C_u^r, $i_1, i_2 = 1, ..., n_{ru}$, and where w_i is the weight associated with the observation $\mathbf{X}(i)^{ru}$, and where $\lambda = \sum_{i=1}^{n_{ru}} w_i$.

When observations are evenly weighted, i.e., $w_i = 1/n$, $i = 1, ..., n$, Equation (2.30) becomes

$$I(C_u^r) = \frac{1}{2n_{ru}n} \sum_{i_1=1}^{n_{ru}} \sum_{i_2=1}^{n_{ru}} d^2(i_1, i_2). \qquad (2.31)$$

Definition 2.4.11. *The* ***total within-cluster variation*** *for partition $P_r = \{C_1, ..., C_r\}$ is*

$$W(P_r) = \sum_{u=1}^{r} I(C_u^r) \qquad (2.32)$$

where $I(C_u^r)$ is the within-cluster variation for C_u^r given in Equation (2.30).

Suppose at stage r, we divide $C_{u^*}^r$ into two clusters $(C_{u^*}^{r+1}, C_{u^*+1}^{r+1})$; then,

$$P_{r+1} = P_r \cup \{C_{u^*}^{r+1}, C_{u^*+1}^{r+1}\} - \{C_{u^*}^r\}.$$

Hence, at stage $r + 1$, from Equation (2.32),

$$W(P_{r+1}) = W(P_r) + I(C_{u^*}^{r+1}) + I(C_{u^*+1}^{r+1}) - I(C_{u^*}^r).$$

Therefore,

$$W(P_r) - W(P_{r+1}) = I(C_{u^*}^r) - I(C_{u^*}^{r+1}) - I(C_{u^*+1}^{r+1}). \tag{2.33}$$

The purpose of clustering is that within the same group, distances among observations are as small as possible, i.e., total within-cluster variation is as small as possible. The total within-cluster variations at stage r and $r+1$ are $W(P_r)$ and $W(P_{r+1})$, respectively. When moving from stage r to $r+1$, we want to minimize $W(P_{r+1})$, which is equivalent to maximizing $W(P_r) - W(P_{r+1})$. Maximizing $W(P_r) - W(P_{r+1})$ is equivalent to maximizing $I(C_{u^*}^r) - I(C_{u^*}^{r+1}) - I(C_{u^*+1}^{r+1})$ according to Equation (2.33).

References

Billard, L. (2007). Dependencies and Variation Components of Symbolic Interval-Valued Data. In: *Selected Contributions in Data Analysis and Classification.* Springer-Verlag, Brelin, 3-13.

Billard, L. (2008). Sample Covariance Functions for Complex Quantitative Data. In: *Proceedings, World Conferences International Association of Statistical Computing* 2008, Yokohama, Japan, 157-163.

Billard, L. (2011). Brief Overview of Symbolic Data and Analytic Issues. *Statistical Analysis and Data Mining*, 4, 149-156.

Bertrand, P. and Goupil, F. (2000). Descriptive Statistics for Symbolic Data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data* (eds. H.-H. Bock and E. Diday). Springer-Verlag, Berlin, 103-124.

Billard, L. and Diday, E. (2000). Regression Analysis for Interval-Valued Data. In: *Data Analysis, Classification and Related Methods, Proceedings of the Seventh Conference of the International Federation of Classification Societies(IFCS'00)*, Springer, Belgium, 369-374.

Billard, L. and Diday, E. (2006). *Symbolic Data Analysis: Conceptual Statistics and Data Mining.* John Wiley and Sons, Chichester.

Bock, H.-H. and Diday, E. (eds.) (2000). *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data.* Springer-Verlag, Berlin.

Chavent, M. (1998). A Monothetic Clustering Method. *Pattern Recognition Letters*, 19, 989-996.

Chavent, M. (2000). Criterion-Based Divisive Clustering for Symbolic Data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data.* (Eds. Bock, H.-H. and Diday, E.). Springer-Verlag, Berlin, 299-311.

de Carvalho, F.A.T. (1994). Proximity Coefficients between Boolean Symbolic Objects. In: *New Approaches in Classification and Data Analysis, Series: Studies in Classification, Data Analysis, and Knowledge Organisation* (Eds. Diday, E., Lechevallier, Y., Schader, M. and Bertrand, P.). Springer-Verlag, Berlin, 387-394.

de Carvalho, F.A.T. (1998). Extension Based Proximity Coefficients between Constrained Boolean Symbolic Objects. In: *Proceedings of the 5th Conference of the International Federation of Classification Societies (IFCS'96)* (Eds. Hayashi, C., Ohsumi, N., Yajima, K., Tanaka, Y., Bock, H.-H. and Baba, Y.). Springer-Verlag, Berlin, 370-378.

de Carvalho F.A.T., Lima Neto, E.A. and Tenorio, C.P. (2004). A New Method to Fit a Linear Regression Model for Interval-valued Data. In: *Lecture Notes in Computer Science, KI2004 Advances in Artifical Inteligence.* Springer-Verlag, 295-306.

Diday, E. (1987). Introduction à l'Approche Symbolique en Analyse des Données. *Premières Journées Symbolique-Numérique,* CEREMADE, Université Paris, Dauphine, 21-56.

Gowda, K.C. and Diday, E. (1991). Symbolic Clustering Using a New Dissimilarity Measure. *Pattern Recognition*, 24, 567-578.

Gowda, K.C. and Diday, E. (1992). Symbolic Clustering Using a New Similarity Measure. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 368-378.

Hausdorff, F. (1937). *Set Theory* (translated into English by Aumann, J. R. 1957). Chelsey, New York.

Ichino, M. and Yaguchi, H. (1994). Generalized Minkowski Metrics for Mixed Feature Type Data Analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 24, 698-708.

Kim, J. (2009). *Dissimilarity Measures for Histogram-Valued Data and Divisive Clustering of Symbolic Objects.* Doctoral Dissertation, The University of Georgia.

Kim, J. and Billard, L. (2012). Dissimilarity Measures and Divisive Clustering for Symbolic Multimodal-Valued Data. *Computational Statistics and Data Analysis*, 56, 2795-2808.

Kuntner, M.H., Nachtsheim, C.J., Neter, J. and Li, W. (2005). *Applied Linear Statistical Models, 5th Edition.* McGraw-Hill.

Lance, G.N. and Williams, W.T. (1968). Note on a New Information Statistic Classification Program. *The Computer Journal*, 11, 195-197.

Le-Rademacher, J.G. (2008). *Principal Component Analysis for Interval-Valued and Histogram-Valued Data and Likelihood Functions and Some Maximum Likelihood Estimators for Symbolic Data.* Doctoral Dissertation, The University of Georgia.

Lima Neto, E.A., de Carvalho F.A.T. and Freire, E.S. (2004). Univariate and Multivariate Linear Regression Methods to Predict Interval-valued Features. In: *Lecture Notes in Computer Science, AI 2004 Advances in Artificial Intelligence.* Springer-Verlag, Berlin, 526-537.

Lima Neto, E.A., de Carvalho F.A.T. and Freire, E.S. (2005). Applying Constrained Linear Regression Models to predict Interval-Valued Data. In: *Lecture Notes in Computer Science, KI: Advances in Artificial Inteligence* (ed. U. Furbach). Springer-Verlag, Brelin, 92-106.

Lima Neto, E.A. 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, 1500-1515.

Lima Neto, E.A 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.

Williams, W.T. and Lambert, J.M. (1959). Multivariate Methods in Plant Ecology. *Journal of Ecology*, 47, 83-101.

Xu, W. (2010). *Symbolic Data Analysis: Interval-valued Data Regression.* Doctoral Dissertation, University of Georgia.

Chapter 3

Analysis Of Variance For Interval-Valued Data

In Section 2.3, we have put forward regression methods for symbolic data. However, when dealing with several categorical predictor variables (or factors), another useful method is analysis of variance (ANOVA).

In this chapter, Section 3.1 will give a brief review of ANOVA for classical data, and of ANOVA being done as a multiple regression. The ANOVA for interval-valued data and how it can be done as a multiple regression will be shown in Section 3.2. Finally, a practical data set to which the methods will be applied is described briefly in Section 3.3 in order to illustrate how to do ANOVA for interval-valued data.

3.1 Preliminaries

One-way ANOVA

Most of the contents in Section 3.1 are from Wu and Hamada (2000). An ANOVA is a sophisticated way to analyze data with one or more categorical predictor variables and a continuous outcome. Classically, the linear model for a one-way layout is

$$y_{ij} = \mu + \tau_i + \epsilon_{ij}, \quad i = 1, ..., p,\ j = 1, ..., n_i, \tag{3.1}$$

where y_{ij} is the j^{th} observation with treatment i, μ is the overall mean, τ_i is the ith treatment effect, the errors ϵ_{ij} are independent $N(0, \sigma^2)$ with mean 0 and variance σ^2, p is the number of treatments, and n_i is the number of observations with treatment i. Table 3.1 shows the ANOVA table summarizing relevant statistics for the linear model Equation (3.1), where $N = \sum_{i=1}^{p} n_i$.

Table 3.1: One-way ANOVA Table for Classical Data

	Degrees of	Sum of	Mean	
Source	Freedom	Squares	Squares	F
Treatment	$p-1$	$\sum_{i=1}^{p} n_i(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2$	$MST = \frac{SST}{p-1}$	$\frac{MST}{MSE}$
Residual	$N-p$	$\sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i\cdot})^2$	$MSE = \frac{SSE}{N-p}$	
Total	$N-1$	$\sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{\cdot\cdot})^2$		

The ANOVA statistics for the one-way layout can be derived using the decomposition

$$\begin{aligned} y_{ij} &= \hat{\mu} + \hat{\tau}_i + \epsilon_{ij} \\ &= \bar{y}_{\cdot\cdot} + (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}) + (y_{ij} - \bar{y}_{i\cdot}) \end{aligned}$$

where

$$\hat{\mu} = \bar{y}_{\cdot\cdot}, \quad \hat{\tau}_i = \bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}, \quad \epsilon_{ij} = y_{ij} - \bar{y}_{i\cdot};$$

we have

$$y_{ij} - \bar{y}_{\cdot\cdot} = (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}) + (y_{ij} - \bar{y}_{i\cdot}). \tag{3.2}$$

By squaring both sides of Equation (3.2) and summing over i and j, it can be verified that

$$\sum_{i=1}^{p}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_{\cdot\cdot})^2 = \sum_{i=1}^{p} n_i(\bar{y}_{i\cdot}-\bar{y}_{\cdot\cdot})^2 + \sum_{i=1}^{p}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_{i\cdot})^2,$$

which is

$$SSTotal = SST + SSE$$

where

$$SSTotal = \sum_{i=1}^{p}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_{\cdot\cdot})^2, \tag{3.3}$$

$$SST = \sum_{i=1}^{p} n_i(\bar{y}_{i\cdot}-\bar{y}_{\cdot\cdot})^2, \tag{3.4}$$

and

$$SSE = \sum_{i=1}^{p}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_{i\cdot})^2. \tag{3.5}$$

Note that the SSE of Equation (3.5) and $SSTotal$ of Equation (3.3) here can also be written, respectively, as

$$SSE = \sum_{i=1}^{p} n_i s^2(y_i) \tag{3.6}$$

and

$$SSTotal = Ns^2(y) \tag{3.7}$$

where $s^2(y_i)$ is the sample variance of the y_{ij}'s with treatment i and $s^2(y)$ is the sample variance of all y_{ij}'s. Rewriting SSE and $SSTotal$ in this way is very useful when deriving the ANOVA table for interval-valued data. The statistics of Table 3.1 can be rewritten as in Table 3.2.

Table 3.2: Rewritten One-way ANOVA Table for Classical Data

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F
Treatment	$p-1$	$\sum_{i=1}^{p} n_i(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2$	$MST = \frac{SST}{p-1}$	$\frac{MST}{MSE}$
Residual	$N-p$	$\sum_{i=1}^{p} n_i s^2(y_i)$	$MSE = \frac{SSE}{N-p}$	
Total	$N-1$	$Ns^2(y)$		

Regression Approach to a Single-Factor ANOVA

An ANOVA model Equation (3.1) is a linear model (see, e.g., Kutner et al., 2005), i.e.,

$$y_{ij} = \mu + \tau_1 x_{ij1} + \cdots + \tau_p x_{ijp} + \epsilon_{ij}, \quad i = 1, ..., p, \; j = 1, ..., n_i, \tag{3.8}$$

where y_{ij} is the j^{th} observation with treatment i, μ is the overall mean, τ_i is the i^{th} treatment effect, the errors ϵ_{ij} are independent $N(0, \sigma^2)$ with mean 0 and variance σ^2, p is the number of treatments, n_i is the number of observations with treatment i and

$$x_{ij1} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level 1,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\vdots$$

$$x_{ijp} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level } p, \\ 0 & \text{otherwise.} \end{cases}$$

By Christensen (1996), Equation (3.8) can also be written in matrix form as

$$E\begin{pmatrix} \mathbf{y}_{(1)} \\ \mathbf{y}_{(2)} \\ \vdots \\ \mathbf{y}_{(p)} \end{pmatrix}_{N\times 1} = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{1}_{n_2} & \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{1}_{n_p} & \mathbf{0}_{n_p} & \mathbf{0}_{n_p} & \cdots & \mathbf{1}_{n_p} \end{pmatrix}_{N\times(p+1)} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_p \end{pmatrix}_{(p+1)\times 1}$$

$$= \mathbf{X}_{N\times(p+1)}\boldsymbol{\beta}_{(p+1)\times 1}$$

where $\mathbf{y}_{(i)} = (y_{i1}, ..., y_{in_i})'$ and $N = \sum_{i=1}^{p} n_i$.

It is easy to show that $rank[\mathbf{X}_{N\times(p+1)}] = p < p+1$; hence,

$$rank[\mathbf{X'X}_{(p+1)\times(p+1)}] \leq rank(\mathbf{X}_{N\times(p+1)}) < p+1.$$

Therefore, $\mathbf{X'X}$ is not invertible. In order to solve this problem, the constraint

$$\sum_{i=1}^{p} \tau_i = 0 \tag{3.9}$$

has to be added. Note that Equation (3.9) can also be expressed as $\tau_p = -\tau_1 - \tau_2 - \cdots \tau_{p-1}$. Thus, we shall use only the parameters $\mu, \tau_1, ..., \tau_{p-1}$ for the linear model. Then, Equation (3.8) is now

$$y_{ij} = \mu + \tau_1 x_{ij1} + \cdots + \tau_{p-1} x_{ij,p-1} + \epsilon_{ij}, \quad i = 1, ..., p, \;\; j = 1, ..., n_i,$$

with

$$x_{ij1} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level 1,} \\ -1 & \text{if observation } y_{ij} \text{ from factor level } p, \\ 0 & \text{otherwise,} \end{cases}$$

$$\vdots$$

$$x_{ij,p-1} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level } p-1, \\ -1 & \text{if observation } y_{ij} \text{ from factor level } p, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix form is now

$$\begin{aligned} E\begin{pmatrix} \mathbf{y}_{(1)} \\ \mathbf{y}_{(2)} \\ \vdots \\ \mathbf{y}_{(p)} \end{pmatrix}_{N\times 1} &= \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{1}_{n_2} & \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{1}_{n_{p-1}} & \mathbf{0}_{n_{p-1}} & \mathbf{0}_{n_{p-1}} & \cdots & \mathbf{1}_{n_{p-1}} \\ \mathbf{1}_{n_p} & -\mathbf{1}_{n_p} & -\mathbf{1}_{n_p} & \cdots & -\mathbf{1}_{n_p} \end{pmatrix}_{N\times p} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{p-1} \end{pmatrix}_{p\times 1} \\ &= \mathbf{X}_{N\times p}\boldsymbol{\beta}_{p\times 1} \qquad (3.10) \end{aligned}$$

where $\mathbf{y}_{(i)} = (y_{i1}, ..., y_{in_i})'$ and $N = \sum_{i=1}^{p} n_i$. Therefore, the least squares estimator of $\boldsymbol{\beta}_{p\times 1}$ in Equation (3.10) is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are defined in Equation (3.10).

3.2 ANOVA for Interval Data

In order to distinguish between ANOVA tables for classical and interval-valued data, some notation has to be introduced first. Let $Y_{ij} = [a_{ij}, b_{ij}]$ be the j^{th} observation with treatment i. Again, we assume that the spread of observations within the interval $[a_{ij}, b_{ij}]$ is uniform.

One-Way ANOVA

For interval-valued data Y_{ij}, the linear model for the one-way layout is

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij},\ i = 1, ..., p,\ j = 1, ..., n_i, \tag{3.11}$$

where μ is the overall mean, τ_i is the ith treatment effect, the errors ϵ_{ij} are independent $N(0, \sigma^2)$ with mean 0 and variance σ^2, p is the number of treatments, and n_i is the number of observations with treatment i.

By analogy with Table 3.2, Table 3.3 is the ANOVA table for the linear model Equation (3.28), where $N = \sum_{i=1}^{p} n_i$, $\bar{Y}_{i\cdot}$ is the mean of observations for the ith treatment, $\bar{Y}_{\cdot\cdot}$ is the mean of all observations, $S^2(Y_i)$ is the sample variance of Y_{ij}'s with treatment i and $S^2(Y)$ is the sample variance of all Y_{ij}'s. We observe that the entries in both tables are similar. However, in contrast to the entries in Table 3.2 for classical data, those in Table 3.3 are calculated for interval data, as follows.

Table 3.3: One-way ANOVA Table for Interval Data

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F
Treatment	$p-1$	$\sum_{i=1}^{p} n_i(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2$	$MST = \frac{SST}{p-1}$	$\frac{MST}{MSE}$
Residual	$N-p$	$\sum_{i=1}^{p} n_i S^2(Y_i)$	$MSE = \frac{SSE}{N-p}$	
Total	$N-1$	$NS^2(Y)$		

From Bertrand and Goupil (2000), we can obtain

$$\begin{aligned}
\bar{Y}_{i\cdot} &= \frac{1}{2n_i}\sum_{j=1}^{n_i}(a_{ij}+b_{ij}), \\
\bar{Y}_{\cdot\cdot} &= \frac{1}{2N}\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij}+b_{ij}), \\
S^2(Y_i) &= \frac{1}{3n_i}\sum_{j=1}^{n_i}(a_{ij}^2+a_{ij}b_{ij}+b_{ij}^2)-\frac{1}{4n_i^2}[\sum_{j=1}^{n_i}(a_{ij}+b_{ij})]^2, \\
S^2(Y) &= \frac{1}{3N}\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij}^2+a_{ij}b_{ij}+b_{ij}^2)-\frac{1}{4N^2}[\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij}+b_{ij})]^2. \qquad (3.12)
\end{aligned}$$

Therefore,

$$\hat{\mu}=\bar{Y}_{\cdot\cdot},\ \hat{\tau}_i=\bar{Y}_{i\cdot}-\bar{Y}_{\cdot\cdot}.$$

It can be proved that

$$NS^2(Y)=\sum_{i=1}^{p}n_i(\bar{Y}_{i\cdot}-\bar{Y}_{\cdot\cdot})^2+\sum_{i=1}^{p}n_iS^2(Y_i), \qquad (3.13)$$

which is

$$SSTotal=SST+SSE$$

where

$$SSTotal=NS^2(Y), \qquad (3.14)$$

$$SST=\sum_{i=1}^{p}n_i(\bar{Y}_{i\cdot}-\bar{Y}_{\cdot\cdot})^2, \qquad (3.15)$$

and

$$SSE=\sum_{i=1}^{p}n_iS^2(Y_i). \qquad (3.16)$$

The detailed proof is in this chapter's appendix.

For the particular case of classical observations, with $a_{ij} = b_{ij} = y_{ij}$, we can show that

$$\begin{aligned} SSE &= \sum_{i=1}^{p} n_i S^2(Y_i) \\ &= \sum_{i=1}^{p} n_i \{ \frac{1}{3n_i} \sum_{j=1}^{n_i} (a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{1}{4n_i^2} [\sum_{j=1}^{n_i} (a_{ij} + b_{ij})]^2 \} \\ &= \sum_{i=1}^{p} n_i \{ \frac{1}{3n_i} \sum_{j=1}^{n_i} (y_{ij}^2 + y_{ij}y_{ij} + y_{ij}^2) - \frac{1}{4n_i^2} [\sum_{j=1}^{n_i} (y_{ij} + y_{ij})]^2 \} \\ &= \sum_{i=1}^{p} \sum_{j=1}^{n_i} y_{ij}^2 - \sum_{i=1}^{p} n_i \bar{y}_i^2 \\ &= \sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i\cdot})^2, \end{aligned} \tag{3.17}$$

and

$$\begin{aligned} SSTotal &= NS^2(Y) \\ &= N\{ \frac{1}{3N} \sum_{i=1}^{p} \sum_{j=1}^{n_i} (a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{1}{4N^2} [\sum_{i=1}^{p} \sum_{j=1}^{n_i} (a_{ij} + b_{ij})]^2 \} \\ &= N\{ \frac{1}{3N} \sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij}^2 + y_{ij}y_{ij} + y_{ij}^2) - \frac{1}{4N^2} [\sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij} + y_{ij})]^2 \} \\ &= \sum_{i=1}^{p} \sum_{j=1}^{n_i} y_{ij}^2 - N\bar{y}_{\cdot\cdot}^2 \\ &= \sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{\cdot\cdot})^2. \end{aligned} \tag{3.18}$$

It is not surprising that SSE and $SSTotal$ now are the same as those with classical data as defined in Section 3.1.

Regression Approach to a Single-Factor ANOVA

The linear model for interval-valued data is

$$Y_{ij} = \mu + \tau_1 x_{ij1} + \cdots + \tau_p x_{ijp} + \epsilon_{ij}, \quad i = 1, ..., p, \ j = 1, ..., n_i, \tag{3.19}$$

where Y_{ij} is the j^{th} observation with treatment i, μ is the overall mean, τ_i is the i^{th} treatment effect, the errors ϵ_{ij} are independent $N(0, \sigma^2)$ with mean 0 and variance σ^2, p is the number of treatments, n_i is the number of observations with treatment i and

$$x_{ij1} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level 1,} \\ 0 & \text{otherwise,} \end{cases}$$
$$\vdots$$
$$x_{ijp} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level } p\text{,} \\ 0 & \text{otherwise.} \end{cases}$$

By analogy with the symbolic covariance method (SCM) proposed by Xu (2010, Section 2.3), Equation (3.19) can be centered as

$$Y_{ij} - \bar{Y} = \tau_1(x_{ij1} - \bar{x}_1) + \cdots + \tau_p(x_{ijp} - \bar{x}_p) + \epsilon_{ij}, \quad i = 1, ..., p, \ j = 1, ..., n_i, \tag{3.20}$$

where $\bar{Y}$ is the sample mean of Y_{ij} and $\bar{x}_l$, $l = 1, ..., p$, is the sample mean of x_{ijl} for a given l. The matrix form of Equation (3.20) is

$$E(\mathbf{Y} - \bar{\mathbf{Y}}) = (\mathbf{X} - \bar{\mathbf{X}})\boldsymbol{\beta} \tag{3.21}$$

where

$$\mathbf{Y} = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{pn_p} \end{pmatrix}_{N\times 1} \qquad \bar{\mathbf{Y}} = \begin{pmatrix} \bar{Y} \\ \vdots \\ \bar{Y} \end{pmatrix}_{N\times 1} \qquad \boldsymbol{\beta} = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_p \end{pmatrix}_{p\times 1}$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & \vdots & & \vdots \\ \mathbf{0}_{n_p} & \mathbf{0}_{n_p} & \cdots & \mathbf{1}_{n_p} \end{pmatrix}_{N\times p} \qquad \bar{\mathbf{X}} = \begin{pmatrix} \frac{n_1}{N} & \cdots & \frac{n_p}{N} \\ \vdots & & \vdots \\ \frac{n_1}{N} & \cdots & \frac{n_p}{N} \end{pmatrix}_{N\times p}$$

with $N = \sum_{i=1}^{p} n_i$. Hence,

$$\mathbf{X} - \bar{\mathbf{X}} = \begin{pmatrix} (1-\frac{n_1}{N})\mathbf{1}_{n_1} & -\frac{n_2}{N}\mathbf{1}_{n_1} & \cdots & -\frac{n_p}{N}\mathbf{1}_{n_1} \\ -\frac{n_1}{N}\mathbf{1}_{n_2} & (1-\frac{n_2}{N})\mathbf{1}_{n_2} & \cdots & -\frac{n_p}{N}\mathbf{1}_{n_2} \\ \vdots & \vdots & & \vdots \\ -\frac{n_1}{N}\mathbf{1}_{n_p} & -\frac{n_2}{N}\mathbf{1}_{n_p} & \cdots & (1-\frac{n_p}{N})\mathbf{1}_{n_p} \end{pmatrix}_{N\times p}.$$

It is obvious that $rank(\mathbf{X} - \bar{\mathbf{X}}) < p$, so that

$$rank[(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{X} - \bar{\mathbf{X}})_{p\times p}] \leq rank(\mathbf{X} - \bar{\mathbf{X}}) < p.$$

Hence, $(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{X} - \bar{\mathbf{X}})$ is not invertible. The constraint

$$\sum_{i=1}^{p} \tau_i = 0,$$

i.e.,

$$\tau_p = -\tau_1 - \tau_2 - \cdots - \tau_{p-1}, \tag{3.22}$$

has to be added. Now, Equation (3.20) becomes

$$Y_{ij} - \bar{Y} = \tau_1(x_{ij1} - \bar{x}_1) + \cdots + \tau_{p-1}(x_{ij,p-1} - \bar{x}_{p-1}) + \epsilon_{ij}, \quad i = 1, ..., p,\ j = 1, ..., n_i, \tag{3.23}$$

with

$$x_{ij1} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level 1,} \\ -1 & \text{if observation } y_{ij} \text{ from factor level } p, \\ 0 & \text{otherwise,} \end{cases}$$

$$\vdots$$

$$x_{ij,p-1} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level } p-1, \\ -1 & \text{if observation } y_{ij} \text{ from factor level } p, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix form of Equation (3.23) is $E(\mathbf{Y} - \bar{\mathbf{Y}}) = (\mathbf{X} - \bar{\mathbf{X}})\boldsymbol{\beta}$ still, except that now

$$\mathbf{Y} = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{pn_p} \end{pmatrix}_{N\times 1} \quad \bar{\mathbf{Y}} = \begin{pmatrix} \bar{Y} \\ \vdots \\ \bar{Y} \end{pmatrix}_{N\times 1} \quad \boldsymbol{\beta} = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_{p-1} \end{pmatrix}_{(p-1)\times 1}$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{1}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \vdots & & \vdots \\ \mathbf{0}_{n_{p-1}} & \cdots & \mathbf{1}_{n_{p-1}} \\ -\mathbf{1}_{n_p} & \cdots & -\mathbf{1}_{n_p} \end{pmatrix}_{N\times(p-1)} \quad \bar{\mathbf{X}} = \begin{pmatrix} \frac{n_1-n_p}{N} & \cdots & \frac{n_{p-1}-n_p}{N} \\ \vdots & & \vdots \\ \frac{n_1-n_p}{N} & \cdots & \frac{n_{p-1}-n_p}{N} \end{pmatrix}_{N\times(p-1)} \tag{3.24}$$

Therefore, the least squares estimator of parameter $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = [(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{X} - \bar{\mathbf{X}})]^{-1}(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{Y} - \bar{\mathbf{Y}}) \tag{3.25}$$

where $\mathbf{X}$, $\bar{\mathbf{X}}$, $\mathbf{Y}$ and $\bar{\mathbf{Y}}$ are from Equation (3.24).

Since $(\mathbf{X}-\bar{\mathbf{X}})$ is a matrix with classical entries, so $[(\mathbf{X}-\bar{\mathbf{X}})'(\mathbf{X}-\bar{\mathbf{X}})]^{-1}$ can be obtained easily. The way to obtain $(\mathbf{X}-\bar{\mathbf{X}})'(\mathbf{Y}-\bar{\mathbf{Y}})$ is similar to that in Xu (2010, Section 2.3), i.e.,

$$(\mathbf{X}-\bar{\mathbf{X}})'(\mathbf{Y}-\bar{\mathbf{Y}}) = (n \times Cov(X_l, Y))_{(p-1)\times 1}, \quad l = 1, ..., p-1. \tag{3.26}$$

Recall that x_{ijl} for Equation (3.23) is

$$x_{ijl} = \begin{cases} 1 & \text{if observation } y_{ij} \text{ from factor level } l, \\ -1 & \text{if observation } y_{ij} \text{ from factor level } p, \quad i = 1, ..., p,\ j = 1, ..., n_i,\ l = 1, ..., p-1, \\ 0 & \text{otherwise;} \end{cases}$$

so

$$x_{ijl} = I_{i=l} - I_{i=p}, \quad i = 1, ..., p,\ l = 1, ..., p-1,$$

which is a special case of interval-valued data with both lower and upper bounds equal to x_{ijl}. From Billard (2008),

$$\begin{aligned} Cov(X_l, Y) &= \frac{1}{6N}\sum_{i=1}^{p}\sum_{j=1}^{n_i}[2(a_{ij}-\bar{Y})(x_{ijl}-\bar{X}_l) + (a_{ij}-\bar{Y})(x_{ijl}-\bar{X}_l) \\ &\quad +(b_{ij}-\bar{Y})(x_{ijl}-\bar{X}_l) + 2(b_{ij}-\bar{Y})(x_{ijl}-\bar{X}_l)]. \end{aligned} \tag{3.27}$$

Substituting Equation (3.27) into Equation (3.26) and substituting Equation (3.26) into Equation (3.25), we obtain the estimators of $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}} = (\hat{\tau}_1, ..., \hat{\tau}_{p-1})'$. According to Equation (3.22) and Equation (3.19), it follows that

$$\hat{\tau}_p = -\hat{\tau}_1 - \hat{\tau}_2 - \cdots \hat{\tau}_{p-1},$$

and

$$\hat{\mu} = \bar{Y} - \hat{\tau}_1 \bar{x}_1 - \cdots - \hat{\tau}_p \bar{x}_p.$$

3.3 Application

The original classical data set contains information for 144 chickens. The data are collected starting from the 18^{th} week since the chicken's birth. Three different diets are fed to the chicken layers. Each diet is fed to 48 chickens. The diets consist of differing amounts of protein (low, medium, high) added to the diet. The different diets represent different costs of producing layers. The layer body weight (variable Y_1) is measured for each chicken once every three to four weeks (16 times in total) from week 18 to 75. The average daily feed intake (variable Y_2) is recorded for each chicken weekly from week 18 to week 76. The egg weight (variable Y_3) is recorded weekly from week 20 to 76. The number of eggs per week (variable Y_4) is also recorded weekly from week 19 to 76. The percent of egg production (number of eggs per 100 hens per day) (variable Y_5) is recorded from week 19 to 76. The producer is interested in whether different diets will cause differences among chickens by variable Y_1 to Y_5. Details of the experiment and the data can be found in Shim et al. (2013).

Except for the missing and unrecorded data, there are 2274 values for Y_1, 8329 values for Y_2, 7916 values for Y_3, 8199 values for Y_4 and 8191 values for Y_5. For Y_1 (see Table 3.4), all the values that are measured in the same week from the same diet are aggregated into an interval. Further, the interval's lower bound is the 5^{th} percentile and the upper bound is the 95^{th} percentile of the values so that extreme values in two tails can be excluded and the spread within the interval can be more similar to a uniform distribution. For Y_2 to Y_5 (see Table 3.5), they are aggregated into intervals by each diet and by every four weeks. Since Y_2 to Y_5 are measured weekly, aggregating them by every four weeks instead of weekly can keep the number of intervals at a reasonable magnitude (not too big and not too small).

Five one-way ANOVAs are applied on the interval-valued chicken data, with Y_1, Y_2, Y_3, Y_4 and Y_5 being responses and different diets being treatments. Table 3.6 to Table 3.10 are the ANOVA tables of these five variables.

Table 3.6: One-Way ANOVA Table for Body Weight Y_1

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F	p-value
Diet	2	0.686	0.343	9.03	0.0005
Residual	45	1.725	0.038		
Total	47	2.411			

Table 3.7: One-Way ANOVA Table for Daily Feed Intake Y_2

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F	p-value
Diet	2	1583.06	791.53	3.52	0.038
Residual	45	10128.09	225.07		
Total	47	11711.15			

Table 3.8: One-Way ANOVA Table for Egg Weight Y_3

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F	p-value
Diet	2	302.82	151.41	7.20	0.002
Residual	42	883.16	21.03		
Total	44	1185.98			

Table 3.9: One-Way ANOVA Table for Number of Eggs per Week Y_4

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F	p-value
Diet	2	6	3	1.10	0.34
Residual	45	122.25	2.72		
Total	47	128.25			

Table 3.10: One-Way ANOVA Table for Egg Production Y_5

Source	Degrees of Freedom	Sum of Squares	Mean Squares	F	p-value
Diet	2	1175.60	587.80	1.06	0.35
Residual	45	24954.29	554.54		
Total	47	26129.90			

The p-values from Tables 3.6, 3.7 and 3.8 indicate significant treatment effects. This means different diets will cause different layer body weights (Y_1), different average daily feed intakes (Y_2) and different egg weights (Y_3). However, diet does not have a significant effect on the number of eggs per week (Y_4) and egg productions (Y_5). The predicted overall means and treatment effects are listed in Table 3.11.

Table 3.11: Estimation of Overall Mean and Treatment Effects

Response	$\hat{\mu}$	Diet 1	Diet 2	Diet 3
Y_1	1.466	0.149	-0.005	-0.144
Y_2	94.109	5.659	2.216	-7.875
Y_3	56.213	3.057	0.227	-3.285
Y_4	5.125	0.250	0.250	-0.500
Y_5	73.065	3.720	3.274	-6.994

3.4 Appendix

Definition of ϵ_{ij}

Recall that $Y_{ij} = [a_{ij}, b_{ij}]$ is the j^{th} observation with treatment i. For interval-valued data Y_{ij}, the linear model for the one-way layout is

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij},\ i = 1, ..., p,\ j = 1, ..., n_i, \tag{3.28}$$

where μ is the overall mean, τ_i is the ith treatment effect, the errors ϵ_{ij} are independent $N(0, \sigma^2)$ with mean 0 and variance σ^2, p is the number of treatments, and n_i is the number of observations with treatment i.

In Section 3.2, ϵ_{ij} is not defined. In fact, ϵ_{ij} can be defined as

$$\epsilon_{ij} \triangleq Y_{ij} - \mu - \tau_i = [a_{ij} - \mu - \tau_i, b_{ij} - \mu - \tau_i]. \tag{3.29}$$

Then,

$$\hat{\epsilon}_{ij} = [a_{ij} - \bar{Y}_{i\cdot}, b_{ij} - \bar{Y}_{i\cdot}].$$

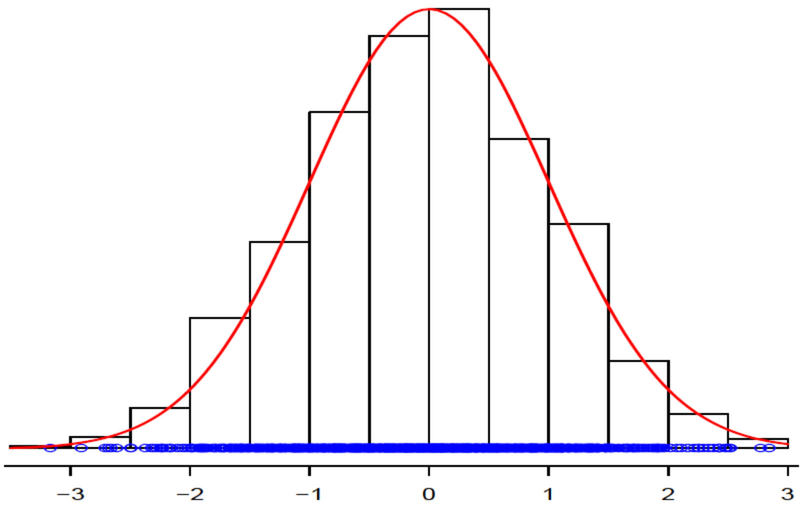


Figure 3.1: Histogram of A Variable Simulated from $N(0,1)$

That is, as for the Y_{ij}, the realizations of the error terms are also intervals. Also, as for Y_{ij}, it is assumed that the error terms aggregated into each interval are uniformly spread within each such interval. Note however the probability distribution of the errors and Y_{ij}'s are still normally distributed. See Figure 3.1 for the histogram of a variable simulated from $N(0,1)$. Values are uniformly spread within each small segment while they still follow a normal distribution with mean as zero and variance as one.

Also, let us define the square of an interval as

$$[a_{ij}, b_{ij}]^2 \triangleq \frac{a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2}{3}. \tag{3.30}$$

When $a_{ij} = b_{ij}$, Equation (3.30) is the square of a classical point.

Therefore, the sum of squares of residuals can be written as

$$\begin{aligned} SSE &= \sum_{i=1}^{p}\sum_{j=1}^{n_i}\hat{\epsilon}_{ij}^2 \\ &= \sum_{i=1}^{p}\sum_{j=1}^{n_i}\frac{1}{3}[(a_{ij}-\bar{Y}_{i\cdot})^2+(a_{ij}-\bar{Y}_{i\cdot})(b_{ij}-\bar{Y}_{i\cdot})+(b_{ij}-\bar{Y}_{i\cdot})^2] \qquad (3.31) \end{aligned}$$

based on Equations (3.29) and (3.30).

On the other hand, the SSE from ANOVA Table 3.3 is

$$SSE = \sum_{i=1}^{p} n_i S^2(Y_i)$$

where $S^2(Y_i)$ can be rewritten as

$$S^2(Y_i) = \frac{1}{3n_i}\sum_{j=1}^{n_i}[(a_{ij}-\bar{Y}_{i\cdot})^2+(a_{ij}-\bar{Y}_{i\cdot})(b_{ij}-\bar{Y}_{i\cdot})+(b_{ij}-\bar{Y}_{i\cdot})^2]$$

according to Equation (2.8). Hence, the SSE from Table 3.3 is

$$\begin{aligned} SSE &= \sum_{i=1}^{p} n_i S^2(Y_i) \\ &= n_i\frac{1}{3n_i}\sum_{j=1}^{n_i}[(a_{ij}-\bar{Y}_{i\cdot})^2+(a_{ij}-\bar{Y}_{i\cdot})(b_{ij}-\bar{Y}_{i\cdot})+(b_{ij}-\bar{Y}_{i\cdot})^2] \\ &= \frac{1}{3}\sum_{j=1}^{n_i}[(a_{ij}-\bar{Y}_{i\cdot})^2+(a_{ij}-\bar{Y}_{i\cdot})(b_{ij}-\bar{Y}_{i\cdot})+(b_{ij}-\bar{Y}_{i\cdot})^2]. \qquad (3.32) \end{aligned}$$

Equations (3.31) and (3.32) are the same term. Therefore, definitions from Equations (3.29) and (3.30) hold here.

Proof of Equation (3.13)

We have

$$NS^2(Y) = \sum_{i=1}^{p} n_i(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2 + \sum_{i=1}^{p} n_i S^2(Y_i).$$

Then, from Equation (3.12) and simplifying the left hand side (LHS) of Equation (3.13), we have

$$\begin{aligned} NS^2(Y) &= N\{\frac{1}{3N}\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{1}{4N^2}[\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij} + b_{ij})]^2\} \\ &= \frac{1}{3}\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{N}{4}(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2. \qquad (3.33) \end{aligned}$$

Substituting the values for $\bar{Y}_{i\cdot}$, $\bar{Y}_{\cdot\cdot}$ and $S^2(Y_i)$ from Equation (3.12) into the right hand side (RHS) of Equation (3.13), we can show that

$$\begin{aligned} &\sum_{i=1}^{p} n_i(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2 + \sum_{i=1}^{p} n_i S^2(Y_i) \\ = &\sum_{i=1}^{p} n_i[\frac{1}{2}(\bar{a}_{i\cdot} + \bar{b}_{i\cdot}) - \frac{1}{2}(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})]^2 \\ &+\sum_{i=1}^{p} n_i\{\frac{1}{3n_i}\sum_{j=1}^{n_i}(a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{1}{4n_i^2}[\sum_{j=1}^{n_i}(a_{ij} + b_{ij})]^2\} \\ = &\frac{1}{4}\sum_{i=1}^{p} n_i[(\bar{a}_{i\cdot} + \bar{b}_{i\cdot}) - (\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})]^2 \\ &+\frac{1}{3}\sum_{i=1}^{p}\sum_{j=1}^{n_i}(a_{ij}^2 + a_{ij}b_{ij} + b_{ij}^2) - \frac{1}{4}\sum_{i=1}^{p} n_i(\bar{a}_{i\cdot} + \bar{b}_{i\cdot})^2. \qquad (3.34) \end{aligned}$$

We want to show Equations (3.33) and (3.34) are equal. After cancelling out common terms, this is equivalent to showing

$$-N(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2 = \sum_{i=1}^{p} n_i[(\bar{a}_{i\cdot} + \bar{b}_{i\cdot}) - (\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})]^2 - \sum_{i=1}^{p} n_i(\bar{a}_{i\cdot} + \bar{b}_{i\cdot})^2. \qquad (3.35)$$

The right hand side of Equation (3.35) is

$$
\begin{aligned}
& \sum_{i=1}^{p} n_i[(\bar{a}_{i\cdot} + \bar{b}_{i\cdot})^2 - 2(\bar{a}_{i\cdot} + \bar{b}_{i\cdot})(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot}) + (\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2] - \sum_{i=1}^{p} n_i(\bar{a}_{i\cdot} + \bar{b}_{i\cdot})^2 \\
= \; & -2\sum_{i=1}^{p} n_i(\bar{a}_{i\cdot} + \bar{b}_{i\cdot})(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot}) + \sum_{i=1}^{p} n_i(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2 \\
= \; & -2(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})\sum_{i=1}^{p} n_i(\bar{a}_{i\cdot} + \bar{b}_{i\cdot}) + N(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2 \\
= \; & -2(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})\sum_{i=1}^{p}(\sum_{j=1}^{n_i} a_{ij} + \sum_{j=1}^{n_i} b_{ij}) + N(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2 \\
= \; & -2(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})(N\bar{a}_{\cdot\cdot} + N\bar{b}_{\cdot\cdot}) + N(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2 \\
= \; & -N(\bar{a}_{\cdot\cdot} + \bar{b}_{\cdot\cdot})^2. \qquad (3.36)
\end{aligned}
$$

Thus, Equation (3.36) is equal to the left hand side of Equation (3.35), and our result is proved.

■

R Code For One-Way ANOVA for Interval-Valued Data

The code is adapted from Xu (2010) and it has been modified to be used for one-way ANOVA for interval-valued data. The F-test for ANOVA has also been added.

```
intANOVA <- function(Y,X)
  # The code is adapted from Xu (2010).
  # It has been modified to be used for ANOVA.
  # The F-test for one-way ANOVA is added.
  # Y: Response, interval-valued
  # X: Treatment, integer or character
  # The data must be sorted by treatment (1,2,3,... or a,b,c,...).
{
```

```
n=length(unique(X)) # number of levels within the factor

m=length(X) # number of observations

XX=rep(0,m*2*(n-1)) # Design matrix

dim(XX)=c(m,2,n-1) # Transfer design matrix into interval format

for (i in 1:(n-1))

{

  XX[,,i][X==i,]=1

  XX[,,i][X==n,]=-1

}

dim(Y)=c(m,2,1)

d=abind(Y,XX,along=3);

p=length(d[1,1,]); #number of variables including y and xi's.

x=rep(0,m*p*2);

dim(x)=c(m,2,p);

cov=matrix(0,p,p); #covariance matrix;

corr=matrix(0,p,p); #correlation coefficient matrix;

temp=matrix(0,p,2);

xbar=c(1:p); #mean for variables:x1, x2, ..., xp;


#read in data to x;

for (i in 1:p)

{

  x[,,i]=matrix(0,m,2);

  x[,,i]=d[,,i];

}


#calcualte mean of variables;

for (i in 1:p)
```

```
{
  for (j in 1:2)
  {
    temp[i,j]=mean(x[,j,i]);
  }
  xbar[i]=mean(temp[i,]);
}
for (k in 1:p)
{
  for (l in 1:p)
  {
    sum=0;
    for (i in 1:m)
    {
      sum=sum+2*(x[i,1,k]-xbar[k])*(x[i,1,l]-xbar[l])+
        (x[i,1,k]-xbar[k])*(x[i,2,l]-xbar[l])+
        (x[i,2,k]-xbar[k])*(x[i,1,l]-xbar[l])+
        2*(x[i,2,k]-xbar[k])*(x[i,2,l]-xbar[l]);
    }
    cov[k,l]=sum/6/m;
  }
}

#get lower triangle part of corr matrix;
for (k in 1:p)
{
  for (l in k:p)
  {
    corr[k,l]=cov[k,l]/sqrt(cov[k,k])/sqrt(cov[l,l]);
  }
}

#get upper triangle part of corr matrix;
for (k in 2:p)
{
  for (l in 1:(k-1))
  {
    corr[k,l]=corr[l,k]
  }
}
covxy=matrix(cov[-1,1],(p-1),1);
covxx=matrix(cov[-1,-1],(p-1),(p-1));
covyy=matrix(cov[1,1])
beta=solve(covxx)%*%covxy;
dbar=c(1:p);
for (i in 1:p)
{
  dbar[i]=mean(d[,,i])
}
ybar=dbar[1];
xbar=dbar[2:p];
beta0=ybar-t(beta)%*%xbar;
beta0=as.numeric(beta0)
tau_p=-(sum(beta))
betaall=rbind(beta,tau_p);
rownames(betaall)=c(1:n)
```

```
  # F test
  sstotal = as.numeric(m*covyy)
  sst = 0
  for (i in 1:n)
  {
    sst = sst + table(X)[i]*betaall[i]^2
  }
  sst = as.numeric(sst)
  sse = sstotal-sst
  mst = sst/(n-1)
  mse = sse/(m-n)
  F = mst/mse
  p.value = 1-pf(F,n-1,m-n)
  anova = c(sst, sse, sstotal, mst, mse, p.value)
  list(mu=beta0,tau=betaall,"SST SSE SSTotal MST MSE Pr(>F)"=anova)
}
```

Table 3.4: Chicken Data Variable Y_1

Diet	Week	Y_1
1	18	[1.04,1.29]
1	21	[1.30,1.69]
1	25	[1.30,1.73]
1	29	[1.26,1.82]
1	33	[1.30,1.82]
1	37	[1.32,1.85]
1	41	[1.35,2.02]
1	45	[1.35,1.96]
1	49	[1.28,1.96]
1	53	[1.35,2.00]
1	57	[1.39,2.09]
1	61	[1.37,2.03]
1	65	[1.36,2.04]
1	69	[1.38,2.09]
1	72	[1.38,2.11]
1	75	[1.35,2.12]
2	18	[1.04,1.27]
2	21	[1.35,1.59]
2	25	[1.30,1.60]
2	29	[1.23,1.59]
2	33	[1.21,1.66]
2	37	[1.17,1.62]
2	41	[1.17,1.64]
2	45	[1.19,1.68]
2	49	[1.18,1.71]
2	53	[1.23,1.74]
2	57	[1.25,1.81]
2	61	[1.26,1.77]
2	65	[1.26,1.81]
2	69	[1.32,1.82]
2	72	[1.31,1.83]
2	75	[1.30,1.83]
3	18	[1.06,1.28]
3	21	[1.27,1.64]
3	25	[1.16,1.69]
3	29	[1.08,1.70]
3	33	[1.02,1.61]
3	37	[0.99,1.55]
3	41	[1.01,1.58]
3	45	[1.00,1.62]
3	49	[1.02,1.60]
3	53	[1.03,1.50]
3	57	[1.09,1.59]
3	61	[1.04,1.59]
3	65	[1.06,1.62]
3	69	[0.99,1.62]
3	72	[1.06,1.65]
3	75	[0.97,1.62]

Table 3.5: Chicken Data Variable Y_2 - Y_5

Diet	Week	Y_2	Y_3	Y_4	Y_5
1	16-19	[43.24,103.27]		[0.00,2.00]	[0.00,28.57]
1	20-23	[77.59,117.77]	[41.20,57.98]	[0.00,7.00]	[0.00,100.00]
1	24-27	[78.00,114.26]	[49.41,62.48]	[5.00,7.00]	[71.43,100.00]
1	28-31	[81.64,120.07]	[52.19,64.83]	[4.00,7.00]	[57.14,100.00]
1	32-35	[82.20,114.53]	[51.61,65.07]	[5.00,7.00]	[71.43,100.00]
1	36-39	[82.10,118.56]	[50.29,65.82]	[5.00,7.00]	[71.43,100.00]
1	40-43	[90.77,116.86]	[52.73,66.46]	[6.00,7.00]	[85.71,100.00]
1	44-47	[88.77,116.96]	[51.79,65.61]	[5.00,7.00]	[71.43,100.00]
1	48-51	[90.20,115.33]	[52.74,65.59]	[6.00,7.00]	[85.71,100.00]
1	52-55	[89.76,121.60]	[53.36,66.15]	[5.00,7.00]	[71.43,100.00]
1	56-59	[88.59,121.63]	[53.02,67.49]	[5.00,7.00]	[71.43,100.00]
1	60-63	[84.63,118.70]	[54.33,69.33]	[5.00,7.00]	[71.43,100.00]
1	64-67	[83.97,118.29]	[55.43,68.27]	[4.00,7.00]	[57.14,100.00]
1	68-71	[84.40,117.93]	[55.16,69.56]	[3.00,7.00]	[42.86,100.00]
1	72-75	[84.39,116.53]	[55.29,69.13]	[4.00,7.00]	[57.14,100.00]
1	76-79	[89.70,120.36]	[55.54,70.26]	[3.00,7.00]	[42.86,100.00]
2	16-19	[33.71,99.34]		[0.00,1.00]	[0.00,14.29]
2	20-23	[76.04,110.97]	[41.70,55.49]	[0.00,7.00]	[0.00,100.00]
2	24-27	[80.40,114.57]	[50.07,58.71]	[5.00,7.00]	[71.43,100.00]
2	28-31	[77.84,117.76]	[49.81,59.70]	[5.00,7.00]	[71.43,100.00]
2	32-35	[70.37,114.83]	[49.32,60.69]	[5.00,7.00]	[71.43,100.00]
2	36-39	[74.93,117.59]	[49.32,60.22]	[4.00,7.00]	[57.14,100.00]
2	40-43	[76.59,114.69]	[49.90,60.89]	[4.00,7.00]	[57.14,100.00]
2	44-47	[82.77,116.44]	[50.74,61.06]	[5.00,7.00]	[71.43,100.00]
2	48-51	[81.66,117.70]	[50.53,61.63]	[5.00,7.00]	[57.14,100.00]
2	52-55	[84.36,121.70]	[50.49,62.88]	[5.00,7.00]	[71.43,100.00]
2	56-59	[83.53,121.87]	[50.97,63.86]	[5.00,7.00]	[71.43,100.00]
2	60-63	[85.20,120.06]	[54.09,64.66]	[5.00,7.00]	[71.43,100.00]
2	64-67	[83.17,116.99]	[55.05,66.48]	[5.00,7.00]	[71.43,100.00]
2	68-71	[82.71,115.29]	[54.50,64.76]	[5.00,7.00]	[71.43,100.00]
2	72-75	[77.37,115.76]	[54.00,64.99]	[4.00,7.00]	[57.14,100.00]
2	76-79	[75.63,120.59]	[53.00,63.73]	[4.00,7.00]	[57.14,100.00]
3	16-19	[36.64,94.37]		[0.00,2.00]	[0.00,28.57]
3	20-23	[73.00,109.51]	[40.70,54.35]	[0.00,7.00]	[0.00,100.00]
3	24-27	[66.73,120.83]	[45.94,56.13]	[5.00,7.00]	[71.43,100.00]
3	28-31	[59.01,112.67]	[46.76,56.92]	[4.00,7.00]	[57.14,100.00]
3	32-35	[53.37,105.29]	[46.35,57.53]	[3.00,7.00]	[42.86,100.00]
3	36-39	[58.14,111.41]	[46.86,56.94]	[3.00,7.00]	[42.86,100.00]
3	40-43	[60.77,112.46]	[48.10,57.93]	[3.00,7.00]	[42.86,100.00]
3	44-47	[60.11,112.17]	[48.18,58.33]	[3.00,7.00]	[42.86,100.00]
3	48-51	[55.36,110.49]	[48.03,58.43]	[3.00,6.00]	[42.86,85.71]
3	52-55	[63.43,118.29]	[47.77,58.92]	[3.00,7.00]	[42.86,100.00]
3	56-59	[74.30,119.36]	[48.43,59.98]	[3.00,7.00]	[42.86,100.00]
3	60-63	[70.77,118.36]	[49.38,60.68]	[3.00,7.00]	[42.86,100.00]
3	64-67	[61.50,114.53]	[49.62,60.32]	[3.00,7.00]	[42.86,100.00]
3	68-71	[56.40,111.46]	[48.88,59.35]	[2.00,7.00]	[28.57,100.00]
3	72-75	[60.34,107.49]	[48.36,60.23]	[2.00,6.00]	[28.57,85.71]
3	76-79	[57.37,113.59]	[48.83,59.63]	[3.00,7.00]	[42.86,100.00]

References

Bertrand, P. and Goupil, F. (2000). Descriptive Statistics for Symbolic Data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data* (eds. H.-H. Bock and E. Diday). Springer-Verlag, Berlin, 103-124.

Billard, L. (2008). Sample Covariance Functions for Complex Quantitative Data. In: *Proceedings, World Conferences International Association of Statistical Computing* 2008, Yokohama, Japan, 157-163.

Christensen, R. (1996). *Plane Answers to Complex Questions: The Theory of Linear Models.* Springer, New York.

Kutner, M.H., Nachtsheim, C.J., Neter, J. and Li, W. (2005). *Applied Linear Statistical Models, 5th Ed.* McGraw-Hill/Irwin, Boston.

Shim, M.Y., Song, E., Billard, L., Aggrey, S. E., Pesti, G. M. and Sodsee P. (2013). Effects of balanced dietary protein levels on egg production and egg quality parameters of individual commercial layers. *Poultry Science*, 92, 2687-2696.

Wu, C.F.J. and Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization.* John Wiley and Sons, New York.

Xu, W. (2010). *Symbolic Data Analysis: Interval-valued Data Regression.* Doctoral Dissertation, University of Georgia.

Chapter 4

Hierarchical Divisive Monothetic Clustering Method for Interval-Valued Data

As discussed in Billard and Diday (2006), there are numerous clustering methods for both classical data and symbolic data. Clustering structures can be either hierarchical or agglomerative. If the clustering process starts with each observation being a cluster of size one and aggregates clusters at each stage, which is a bottom-up process, then it is agglomerative clustering. If two clusters from the same stage have at least one observation in common, i.e., their intersection is not the null set, condition 3 in Definition 2.4.8 will not be met and the agglomerative clustering structure will be pyramidal. This agglomerative process is in contrast to the top-down divisive clustering defined in Definition 2.4.9. In this dissertation, the focus in on divisive clustering. When performing divisive clustering, we can partition the data according to either one single variable (monothetic method) or all variables simultaneously (polythetic method). Also, we need to determine the distance measure to be used.

Chavent (1998, 2000) developed the first divisive monothetic clustering method for interval data by using Hausdorff (1937) distances. Billard and Diday (2006) demonstrated the algorithm in-depth by analyzing practical data using different Hausdorff distances. However, the algorithm has not been implemented in the software package R, and comparisons among different Hausdorff distances have never been made.

In this chapter, the hierarchical divisive monothetic method is applied to interval-valued data using different Hausdorff distances defined in Section 2.4. There are other distances like Gowda-Diday (1991, 1992) and Ichino-Yaguchi (1994) distances which can also be used in divisive clustering method. Gowda-Diday distance utilizes three terms to capture the relative size, relative content and relative measure between two intervals. Ichino-Yaguchi distance also calculates the difference between two intervals by using three terms. Besides that, meet and join operators are applied on these three terms by Ichino-Yaguchi distance. Although Gowda-Diday and Ichino-Yaguchi distances are able to capture the difference between two intervals in a comprehensive way by doing calculation on three terms, the drawback in computing time is obvious. Hausdorff distance, on the other hand, is easy to understand and calculate, and more importantly, needs less computing time when it is implemented. This dissertation focuses on the Hausdorff distance.

Section 4.1 gives the detailed algorithm of applying the method to interval-valued data. Simulations are run in Section 4.2 in order to compare different Hausdorff distances and learn their advantages and disadvantages. In Section 4.3, the method is applied to some practical data sets. Finally, some notes to which attention needs to be paid given taken in Section 4.4.

4.1 Clustering Algorithm

Let $\mathbf{X} = (X_1, ..., X_p)$ be an interval-valued data set with $X_{ij} = [a_{ij}, b_{ij}]$, $a_{ij} \leq b_{ij}$, $i = 1, ..., n$, $j = 1, ..., p$, where X_j is the j^{th} variable. Let $\mathbf{X}(i)$ denote the i^{th} observation of an interval-valued data set and $\mathbf{X}(i) = (X_{i1}, ..., X_{ip})$.

Suppose we have n interval-valued observations $\mathbf{X}(i) \in \Omega = \{\mathbf{X}(1), ..., \mathbf{X}(n)\}$ where Ω is a set of all observations. Let C_u^r be the u^{th} cluster from the r^{th} stage of clustering. At stage r, $\Omega = \bigcup_{u=1}^{r} C_u^r$ with C_u^r containing observations $C_u^r = \{\mathbf{X}^{ru}(1), ..., \mathbf{X}^{ru}(n_{ru})\}$ where $\mathbf{X}^{ru}(i')$, $i' = 1, ..., n_{ru}$, is the i'^{th} observation in the u^{th} cluster and n_{ru} is the number of observations in the u^{th} cluster.

Recall the four different Hausdorff distances between two interval-valued observations $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$, $i_1, i_2 = 1, ..., n$, introduced in Section 2.4:

- **Hausdorff distance**

$$d_j(\mathbf{X}(i_1), \mathbf{X}(i_2)) = max\{|a_{i_1j} - a_{i_2j}|, |b_{i_1j} - b_{i_2j}|\}, \quad j = 1, ..., p; \tag{4.1}$$

- **Euclidean Hausdorff distance**

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \{\sum_{j=1}^{p} [d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))]^2\}^{1/2}; \tag{4.2}$$

- **Span Normalized Euclidean Hausdorff distance**

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{ \sum_{j=1}^{p} \left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_j|} \right]^2 \right\}^{1/2} \tag{4.3}$$

where the span is $|\mathcal{Y}_j| = \max_i(b_{ij}) - \min_i(a_{ij})$;

- **Normalized Euclidean Hausdorff distance**

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{ \sum_{j=1}^{p} \left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_j} \right]^2 \right\}^{1/2} \tag{4.4}$$

where

$$H_j^2 = \frac{1}{2n^2} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} [d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))]^2, \quad j = 1, ..., p.$$

We have introduced the clustering criteria in Section 2.4. The hierarchical divisive monothetic clustering algorithm for interval-valued data is as follows (an illustration can be found in Billard and Diday, 2006):

Step 1.

Within the u^{th} cluster at the r^{th} stage, sort observations according to the j^{th} variable as follows:

First, calculate the n_{ru} means by

$$\bar{X}_{ij}^{ru} = \frac{a_{ij}^{ru} + b_{ij}^{ru}}{2}, \ i = 1, ..., n_{ru},$$

where $X_{ij}^{ru} = [a_{ij}^{ru}, b_{ij}^{ru}]$ is the i^{th} observation for variable j from the u^{th} cluster at the r^{th} stage, and n_{ru} is the number of observations in the u^{th} cluster at the r^{th} stage. Then, reorder the observations $\{\mathbf{X}^{ru}(1), ..., \mathbf{X}^{ru}(n_{ru})\}$ by increasing mean values $\bar{X}_{ij}^{ru}$ to obtain: $\{\mathbf{X}_{(1)}^{ru}, ..., \mathbf{X}_{(n_{ru})}^{ru}\}$.

Step 2.

After Step 1, cut the cluster at the q^{th} cut-point:

Cut $C_u^r = \{\mathbf{X}_{(1)}^{ru}, ..., \mathbf{X}_{(n_{ru})}^{ru}\}$ into $C_u^{r+1} = \{\mathbf{X}_{(1)}^{ru}, ..., \mathbf{X}_{(q)}^{ru}\}$ and $C_{u+1}^{r+1} = \{\mathbf{X}_{(q+1)}^{ru}, ..., \mathbf{X}_{(n_{ru})}^{ru}\}$.

Calculate

$$\Delta_{jq}^{ru} \triangleq I(C_u^r) - I(C_u^{r+1}) - I(C_{u+1}^{r+1})$$

where Δ_{jq}^{ru} is the amount that the total within-cluster variation decreased from P_r to P_{r+1} if we use the j^{th} variable and cut-point q for the u^{th} cluster at the r^{th} stage, and where $I(C_u^r)$ is given in Equation (2.30). In Equation (2.30), $d(i_1, i_2)$ can be one of the distances given in Equations (4.1), (4.2), (4.4) and (4.3).

Note: If the Hausdorff distance (Equation (4.1)) is used, we use the same variable in Step 1 for calculating distances; if the Euclidean Hausdorff distance (Equation (4.2)) is used, distances between observations do not depend on the chosen variable, and hence remain the same across all variables; if the Span Normalized Euclidean Hausdorff distance (Equation (4.3)) and Normalized Euclidean Hausdorff distance (Equation (4.4)) are used, either the distances can be recalculated at each cluster since, from Equation (4.3) and Equation (4.4), the normalization factors $\mathcal{Y}_j$ and H_j can rely on the (sub)cluster size, or the distances can keep invariant like the Euclidean Hausdorff distance since the normalization factors can rely on the data size. More details will be discussed in later sections.

Step 3.

Within the u^{th} cluster at the r^{th} stage, repeat Step 1 and Step 2 for all $j = 1, ..., p$, $q = 1, ..., n_{ru} - 1$, and obtain

$$\Delta^{ru} = max_{j,q}(\Delta_{jq}^{ru}),\ j = 1, ..., p,\ q = 1, ..., n_{ru}.$$

Step 4.

At the r^{th} stage, repeat Step 3 for $u = 1, ..., r$, and obtain

$$\Delta^{r} = max_{u}(\Delta^{ru}),\ u = 1, ..., r.$$

Step 5.

At the r^{th} stage, choose (u^*, j^*, q^*) such that $\Delta_{j^*q^*}^{ru^*} = \Delta^r$.

If $|\{\Delta_{j^*q^*}^{ru^*}\}| > 1$, i.e., if $\Delta_{j^*q^*}^{ru^*}$ is not unique, then select (u^{**}, j^{**}, q^{**}) such that

$$d_{j^{**}}(X_{q^{**}j^{**}}^{ru^{**}}, X_{q^{**}+1,j^{**}}^{ru^{**}}) = \underset{u^*,j^*,q^*}{max}[d_{j^*}(X_{q^*j^*}^{ru^*}, X_{q^*+1,j^*}^{ru^*})]$$

where $d_j(i_1, i_2)$ is the Hausdorff distance given in Equation (4.1);
if $|\{\Delta_{j^*q^*}^{ru^*}\}| = 1$, i.e., $\Delta_{j^*q^*}^{ru^*}$ is unique, then $(u^{**}, j^{**}, q^{**}) = (u^*, j^*, q^*)$. The cut-point is

$$c_r = \frac{\bar{X}_{q^{**}j^{**}}^{ru^{**}} + \bar{X}_{q^{**}+1,j^{**}}^{ru^{**}}}{2}$$

where $\bar{X}_{ij}^{ru}$ is given in Step 1.

Step 6.

At the r^{th} stage, cut the u^{**th} cluster. The criterion is

$$Is\ \bar{X}_{ij^{**}}^{ru^{**}} \leq c_r?,\ i = 1, ..., n_{ru^{**}}.$$

If Yes, then the observation i goes into cluster $C_{u^{**}}^{r+1}$. If No, then it goes into cluster $C_{u^{**}+1}^{r+1}$.

Step 7.

At the $(r+1)^{th}$ stage, the rest of the clusters, except for $C_{u^{**}}^{r+1}$ and $C_{u^{**}+1}^{r+1}$, inherit from the clusters at the r^{th} stage, respectively, that is,

$$C_1^{r+1} = C_1^r, ...,\ C_{u^{**}-1}^{r+1} = C_{u^{**}-1}^r,\ C_{u^{**}+2}^{r+1} = C_{u^{**}+1}^r, ...,\ C_{r+1}^{r+1} = C_r^r.$$

For easier understanding, Step 1 to Step 7 can be summarized as:

Step 1 Sort interval observations within the cluster according to one variable by increasing mean values;

Step 2 Separate the cluster into two clusters according to the variable used in Step 1;

Step 3 Try all the variables and all the separations (repeat Step 1 and Step 2) in order to find a separation causing the maximum decrease of the total within-cluster variation (see Equation (2.33));

Step 4 Repeat Step 3 for all the clusters in order to find a separation causing the maximum decrease of the total within-cluster variation (see Equation (2.33));

Step 5 If there exist more than one separation causing the same maximum decrease, choose the separation causing the maximum Hausdorff distance between two separated intervals;

Step 6 Separate the cluster found in Step 4 and Step 5;

Step 7 Those clusters that are not separated in Step 6 remain the same.

Step 1 to Step 7 can be repeated from $r = 1$ until as many as necessary, as long as there is more than one observation in any cluster.

The code in R to implement this algorithm is given in Appendix 4.5.

4.2 Simulation Studies

We consider six Hausdorff distances. We have mentioned that there are Hausdorff distance (H), Euclidean Hausdorff Distance (EH), Span Normalized Euclidean Hausdorff distance (SNEH) and Normalized Euclidean Hausdorff distance (NEH). As noted in Step 2 from Section 4.1, either the normalization factors for normalized distances, $|\mathcal{Y}_j|$ and H_j from

Equation (4.3) and Equation (4.4), can be recalculated at each cluster, or they can keep invariant at each cut. If they rely on the (sub)cluster size ($n =$ (sub)cluster size) and hence should be recalculated at each step, let us call the two normalized distances Local Span Normalized Euclidean Hausdorff distance (LSNEH) and Local Normalized Euclidean Hausdorff distance (LNEH). Otherwise, if they rely on the data set size ($n =$ data set size) and therefore are invariant from the beginning, let us call them Global Span Normalized Euclidean Hausdorff distance (GSNEH) and Global Normalized Euclidean Hausdorff distance (GNEH). Simulation studies to compare these six distances will be performed in this section, and advantages and disadvantages of each distance will also be discussed.

Case 1: An Intuitive Example

In the first simulation study, we consider a common situation where observations can be easily separated into their corresponding clusters if interval-valued data are used. The process of the simulation is as follows:

Simulation Step 1.

First, 1000 random samples are simulated from bivariate normal distributions

$$N_2\left(\begin{pmatrix} 10 \\ 15 \end{pmatrix}, \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}\right) \tag{4.5}$$

and

$$N_2\left(\begin{pmatrix} 20 \\ 15 \end{pmatrix}, \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}\right), \tag{4.6}$$

respectively. Therefore, there are 2000 classical sample points generated in total. Figure 4.1 shows the scatter plot of these points.

Simulation Step 2.

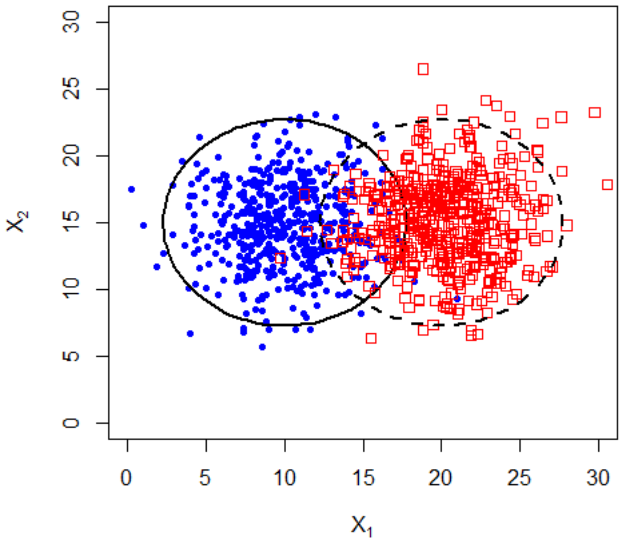


Figure 4.1: Simulated Data from Equations (4.5) and (4.6)

Now, we need to determine the number of classical points to be aggregated in order to form an interval. For example, if we decide to use 100 points for one interval, we take the first 100 points out of the 1000 points simulated according to Equation (4.5) in Simulation Step 1, and set the 5^{th} percentile of 100 points as the lower bound of the interval and the 95^{th} percentile of 100 points as the upper bound of the interval. This aggregates the first 100 points into an interval $\mathbf{X}(1)$. The process can be continued until all the classical points have been aggregated and we therefore have interval-valued observations

$$\{\mathbf{X}(1), ..., \mathbf{X}(10)\} \sim N_2\left(\begin{pmatrix} 10 \\ 15 \end{pmatrix}, \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}\right)$$

and

$$\{\mathbf{X}(11), ..., \mathbf{X}(20)\} \sim N_2\left(\left(\begin{array}{c} 20 \\ 15 \end{array}\right), \left(\begin{array}{cc} 10 & 0 \\ 0 & 10 \end{array}\right)\right).$$

The number of points to be aggregated into an interval determines the number of intervals inherited from a classical data set. In order to learn the impact of the number of intervals on clustering results, the same 2000 classical points simulated in Simulation Step 1 are aggregated to 4, 10, 20, 40 and 100 intervals each time, with 500, 200, 100, 50 and 20 points to constitute an interval, respectively.

Simulation Step 3.

Then, the hierarchical divisive monothetic clustering method is applied to 4, 10, 20, 40 and 100 interval-valued observations using each of the six different Hausdorff distances and the clustering stops when the data are divided into two clusters (i.e., at the 2^{nd} stage of the divisive clustering process) since there are two underlying clusters. Both accuracy and computing time are reported for each combination ($5 \times 6 = 30$ combinations in total).

Definition 4.2.1. *For a given data set, the* ***accuracy*** *of the clustering result is the proportion of observations that are correctly allocated to its underlying cluster. It ranges from 0.5 to 1.*

Let us consider the example in Simulation Step 2, where $\{\mathbf{X}(1), ..., \mathbf{X}(20)\}$ are from two different distributions and thus should be grouped into two underlying clusters
$C_1 = \{\mathbf{X}(1), ..., \mathbf{X}(10)\}$ and $C_2 = \{\mathbf{X}(11), ..., \mathbf{X}(20)\}$.

Example 4.2.2. *Suppose the two clusters from the result are* $C_1^* = \{\mathbf{X}(1), ..., \mathbf{X}(10)\}$ *and* $C_2^* = \{\mathbf{X}(11), ..., \mathbf{X}(20)\}$*, then the accuracy is 1.*

Example 4.2.3. *Suppose the clustering result is* $C_1^* = \{\mathbf{X}(1), ..., \mathbf{X}(10), \mathbf{X}(11), ..., \mathbf{X}(15)\}$ *and* C_2^**=\{**X**(16), ..., **X**(20)\}. In* C_1^**,* 66.7% ($\frac{10}{15} = 66.7\%$) *and* 33.3% ($\frac{5}{15} = 33.3\%$) *of the* *observations are from C_1 and C_2, respectively. In C_2^*,* 100% *of the observations are from C_2. Hence, C_1^*'s underlying cluster is C_1 and C_2^*'s underlying cluster is C_2. Accordingly, $\{\mathbf{X}(11), ..., \mathbf{X}(15)\}$ do not belong to C_1 and are wrongly grouped. Therefore, the accuracy* $= 1 - \frac{5}{20} = 0.75$.

Example 4.2.4. *Suppose the clustering result is $C_1^* = \{\mathbf{X}(1), ..., \mathbf{X}(3), \mathbf{X}(11), ..., \mathbf{X}(14)\}$ and $C_2^* = \{\mathbf{X}(4), ..., \mathbf{X}(10), \mathbf{X}(15), ..., \mathbf{X}(20)\}$. In C_1^*,* 42.9% ($\frac{3}{7} = 42.9\%$) *and* 57.1% ($\frac{4}{7} = 57.1\%$) *of the observations are from C_1 and C_2, respectively. In C_2^*, $\frac{7}{13} = 53.8\%$ of the observations are from C_1 and $\frac{6}{13} = 46.2\%$ of the observations are from C_2. Hence, C_1^*'s underlying cluster is C_2 and C_2^*'s underlying cluster is C_1. Accordingly, $\{\mathbf{X}(1), ..., \mathbf{X}(3)\}$ do not belong to C_2 and $\{\mathbf{X}(15), ..., \mathbf{X}(20)\}$ do not belong to C_1, and are wrongly grouped. Therefore, the accuracy* $= 1 - \frac{9}{20} = 0.55$.

Example 4.2.5. *If the observations are randomly grouped into two clusters, the expected number of observations within each cluster will be 10 and the expected percentage of observations from C_1 and C_2 within each cluster will be 50%. Hence, the accuracy* $= \frac{10\times 50\% + 10\times 50\%}{20} = 0.5$.

Simulation Step 4.

Finally, Simulation Steps 1 to 3 are repeated 1000 times. The mean and standard deviation of accuracy and computing time are calculated.

Table 4.1 shows the means and standard deviations of accuracy and computing time of doing clustering on 4, 10, 20, 40 and 100 intervals by using the six different Hausdorff distances for 1000 replications. Table 4.2 counts the number of times each variable (either X_1 or X_2) is used for cutting among 1000 replications (a cutting variable is the variable chosen in Step 5 and 6 from Section 4.1).

It can be seen that all the intervals, except for a few under GNEH and LNEH, are correctly grouped into its underlying clusters by using the different Hausdorff distances. In

Table 4.1: Mean and Standard Deviation of Accuracy and Computing Time

# of Intervals	H	EH	GSNEH	GNEH	LSNEH	LNEH
Mean (SD) of Accuracy						
4	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	0.976 (0.075)	1.000 (0.000)	0.976 (0.075)
10	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	0.999 (0.008)	1.000 (0.000)	0.999 (0.008)
20	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.003)	1.000 (0.000)	1.000 (0.003)
40	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.001)	1.000 (0.000)	1.000 (0.001)
100	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Mean (SD) of Computing Time (s)						
4	0.008 (0.009)	0.008 (0.010)	0.009 (0.010)	0.008 (0.010)	0.011 (0.011)	0.010 (0.010)
10	0.011 (0.012)	0.014 (0.009)	0.014 (0.009)	0.014 (0.010)	0.022 (0.010)	0.022 (0.010)
20	0.019 (0.009)	0.029 (0.010)	0.029 (0.010)	0.029 (0.011)	0.060 (0.013)	0.061 (0.013)
40	0.050 (0.011)	0.082 (0.012)	0.086 (0.012)	0.086 (0.013)	0.201 (0.017)	0.200 (0.018)
100	0.290 (0.023)	0.528 (0.031)	0.529 (0.033)	0.529 (0.035)	1.235 (0.068)	1.212 (0.063)

this example, it is obvious that X_1 should be the right variable to use for cutting in order to separate the two clusters. However, by looking at Table 4.2, we can see that when the number of intervals is small, like 4, GSNEH, GNEH, LSNEH and LNEH will use X_2 to do the cutting occasionally. This is due to the normalization step when calculating these distances.

Notice that in Figure 4.1, the variation along X_1 is larger than that along X_2. Hence, the normalization factor $|\mathcal{Y}_j|$ and H_j introduced in Equation (4.3) and Equation (4.4) are larger when $j = 1$, i.e.,

$$|\mathcal{Y}_1| > |\mathcal{Y}_2| \text{ and } H_1 > H_2.$$

For any interval observations $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$ from the left cluster (the cluster with solid dots in Figure 4.1) and the right cluster (the cluster with hollow squares in Figure 4.1), respectively,

$$d_1(\mathbf{X}(i_1), \mathbf{X}(i_2)) > d_2(\mathbf{X}(i_1), \mathbf{X}(i_2))$$

Table 4.2: Number of Times Each Variable Is Used For Cutting

# of Intervals	Variable	H	EH	GSNEH	GNEH	LSNEH	LNEH
4	X_1	1000	999	995	918	995	918
	X_2	0	1	5	82	5	82
10	X_1	1000	1000	1000	999	1000	999
	X_2	0	0	0	1	0	1
20	X_1	1000	1000	1000	1000	1000	1000
	X_2	0	0	0	0	0	0
40	X_1	1000	1000	1000	1000	1000	1000
	X_2	0	0	0	0	0	0
100	X_1	1000	1000	1000	1000	1000	1000
	X_2	0	0	0	0	0	0

where $d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))$ is the Hausdorff distance between $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$ for variable X_j. However, it is possible that when $|\mathcal{Y}_1|$ and H_1 become much larger than $|\mathcal{Y}_2|$ and H_2,

$$\frac{d_1(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_1|} < \frac{d_2(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_2|}$$

and

$$\frac{d_1(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_1} < \frac{d_2(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_2}.$$

Hence, the Normalized distances between $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{ \sum_{j=1}^{2} \left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_j|} \right]^2 \right\}^{1/2}$$

and

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{ \sum_{j=1}^{2} \left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_j} \right]^2 \right\}^{1/2}$$

will be dominated by

$$\frac{d_2(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_2|} \text{ and } \frac{d_2(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_2};$$

i.e., variable X_2 will be used as the cutting variable since the clustering process believes that the two clusters are more dissimilar to each other along variable X_2 while this is not true according to Figure 4.1. Here, Span Normalized distances perform better than Normalized distances since H_1 is much larger than H_2 in this case while $|\mathcal{Y}_1|$ is just a little larger than $|\mathcal{Y}_2|$.

In Table 4.1, the computing time increases quadratically when the number of intervals increases. For example, by doing a linear regression of Computing Time on (Number of Intervals)^2, the relationship between computing time and the number of intervals when using the Hausdorff distance is

$$\text{Computing Time} = 0.0071 + 0.000028 \times (\text{Number of Intervals})^2.$$

Clustering using the Hausdorff distance is the fastest method, which is not surprising since, unlike other distances, the Hausdorff distance utilizes only one variable at a time when calculating distances. The computing times for EH, GSNEH and GNEH distances are longer and are similar to each other. That is because they utilize all the variables when calculating distances. The LSNEH and LNEH distances need the most amount of time doing clustering since they not only utilize all the variables when calculating distances, but also recalculate distances at each cluster.

Case 2: A Bad Example For Hausdorff Distance

Now, let us consider a situation where the scales of two variables are of different magnitudes. The steps for this simulation study are the same as those in Section 4.2, except that the first

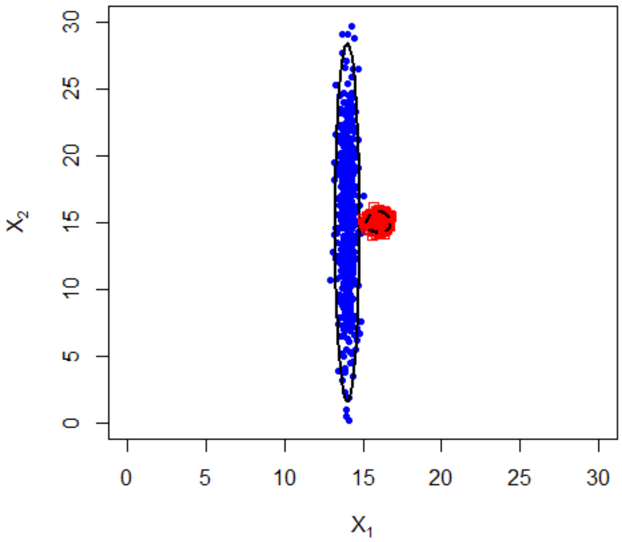


Figure 4.2: Simulated Data from Equations (4.7) and (4.8)

1000 classical random samples are simulated from the bivariate normal distribution

$$N_2\left(\begin{pmatrix} 14 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 30 \end{pmatrix}\right) \tag{4.7}$$

and the other 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix} 16 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}\right). \tag{4.8}$$

See Figure 4.2 for their distributions. Therefore, the corresponding interval valued observations are

$$\{\mathbf{X}(1), ..., \mathbf{X}(\frac{k}{2})\} \sim N_2\left(\begin{pmatrix} 14 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 30 \end{pmatrix}\right)$$

and

$$\{\mathbf{X}(\frac{k}{2}+1),...,\mathbf{X}(k)\} \sim N_2\left(\left(\begin{array}{c}16\\15\end{array}\right),\left(\begin{array}{cc}0.1 & 0\\0 & 0.1\end{array}\right)\right)$$

where $k = 4, 10, 20, 40$ and 100. Table 4.3 lists the means and standard deviations of accuracy and computing time of doing clustering on 4, 10, 20, 40 and 100 intervals by using different distances for 1000 replications. Table 4.4 lists the number of times each variable is used for splitting.

Table 4.3: Mean and Standard Deviation of Accuracy and Computing Time

# of Intervals	H	EH	GSNEH	GNEH	LSNEH	LNEH
Mean (SD) of Accuracy						
4	0.873 (0.125)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
10	0.826 (0.071)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
20	0.795 (0.055)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
40	0.769 (0.042)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
100	0.746 (0.029)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Mean (SD) of Computing Time (s)						
4	0.008 (0.010)	0.009 (0.010)	0.009 (0.010)	0.009 (0.010)	0.010 (0.010)	0.011 (0.010)
10	0.012 (0.010)	0.015 (0.010)	0.015 (0.011)	0.014 (0.010)	0.023 (0.010)	0.024 (0.011)
20	0.021 (0.010)	0.030 (0.011)	0.030 (0.011)	0.031 (0.012)	0.063 (0.013)	0.063 (0.014)
40	0.052 (0.012)	0.086 (0.013)	0.088 (0.013)	0.089 (0.013)	0.207 (0.016)	0.208 (0.017)
100	0.301 (0.018)	0.544 (0.023)	0.547 (0.022)	0.546 (0.023)	1.273 (0.033)	1.267 (0.034)

Table 4.4: Number of Times Each Variable Is Used For Cutting

# of Intervals	Variable	H	EH	GSNEH	GNEH	LSNEH	LNEH
4	X_1	0	508	508	508	508	508
	X_2	1000	492	492	492	492	492
10	X_1	0	955	954	954	954	954
	X_2	1000	45	46	46	46	46
20	X_1	0	998	998	998	998	998
	X_2	1000	2	2	2	2	2
40	X_1	0	1000	1000	1000	1000	1000
	X_2	1000	0	0	0	0	0
100	X_1	0	1000	1000	1000	1000	1000
	X_2	1000	0	0	0	0	0

In this example, except for the Hausdorff distance, all other distances are able to group interval observations into their underlying clusters perfectly. Since the Hausdorff distance utilizes just one variable when calculating distances, the calculation of the distance between two observations can easily be dominated by the variable with larger scale (variable X_2 in this example), and the distance is exaggerated. As a consequence, the within-cluster variation $I(C_u^r)$ (Equation (2.30)) will be calculated based on X_2, and Δ^{ru} and Δ^r in Step 3 and Step 4 of Section 4.1 will also be obtained by using X_2. Therefore, X_2 will always be the variable to be used for cutting, even if it is not the best choice. This can also be observed in Table 4.4 where X_2 is used for splitting 1000 out of 1000 times when the Hausdorff distance is used.

Figure 4.3 is the clustering result of one simulated data set when $k = 20$ using the Hausdorff distance. As discussed above, variable X_2 is dominating the clustering process all the time while the data should better be clustered by variable X_1. Although observations $\{2, 3, 4, 9\}$ are successfully separated from $\{11, ..., 20\}$ at stage 3, the clustering process should have stopped at stage 2 instead of turning the result into 3 clusters.

The accuracy of using the Hausdorff distance decreases when the number of intervals increases. As mentioned in Section 4.2, if more intervals are generated from a classical data set, that means there are fewer points needed to constitute one interval. Therefore, each interval contains less information about the underlying cluster to which it belongs; and for those intervals belonging to the same underlying cluster, their between-interval variations increase. This will cause the total within-cluster variation (Equation (2.32)) of each underlying cluster to increase. Therefore, it will be harder for the clustering process to discover the underlying cluster structure since the process is always trying to minimize the total within-cluster variation to find the best clustering structure. This gives us an idea that sometimes aggregating a classical data set into too many intervals may not be a good idea. Instead, aggregating the data set into fewer intervals may give better results.

Statistics of Table 4.3 computing time are similar to those in Table 4.1. Computing time

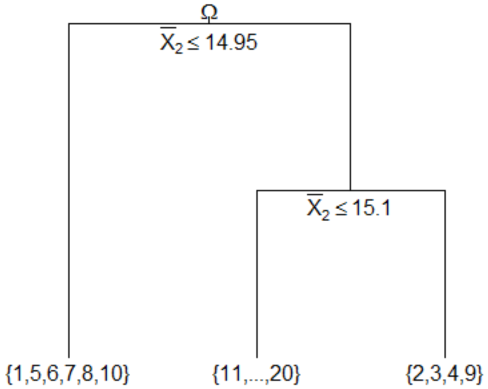


Figure 4.3: Clustering Result Using Hausdorff Distance for Simulation Case 2

increases quadratically when the number of intervals increases. The order of time needed for clustering is $\{H\} < \{EH,\ GSNEH,\ GNEH\} < \{LSNEH,\ LNEH\}$ (”<” represents ”less than”).

Case 3: A Bad Example For Hausdorff/Euclidean Hausdorff Distance

Although the Euclidean Hausdorff distance performs well in the Case 2 study, it can still meet with problems when the scale of one variable is extraordinary larger than other variables. Consider an example where the first 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix} 14 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 30 \end{pmatrix}\right) \tag{4.9}$$

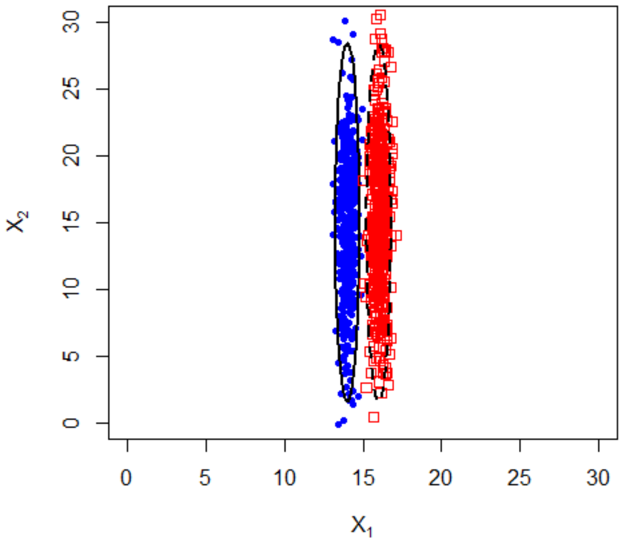


Figure 4.4: Simulated Data from Equations (4.9) and (4.10)

and the second 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix} 16 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 30 \end{pmatrix}\right). \tag{4.10}$$

See Figure 4.4 for their distributions. The interval valued observations aggregated from these data are

$$\{\mathbf{X}(1), ..., \mathbf{X}(\frac{k}{2})\} \sim N_2\left(\begin{pmatrix} 14 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 30 \end{pmatrix}\right)$$

and

$$\{\mathbf{X}(\frac{k}{2}+1), ..., \mathbf{X}(k)\} \sim N_2\left(\begin{pmatrix} 16 \\ 15 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 30 \end{pmatrix}\right)$$

where $k = 4, 10, 20, 40$ and 100. The rest of the simulation process is the same as those in Section 4.2. Table 4.5 lists the means and standard deviations of accuracy and computing time of doing clustering on 4, 10, 20, 40 and 100 intervals by using the six different distances for 1000 replications. Table 4.6 lists the number of times each variable is used for splitting.

Table 4.5: Mean and Standard Deviation of Accuracy and Computing Time

# of Intervals	H	EH	GSNEH	GNEH	LSNEH	LNEH
			Mean (SD) of Accuracy			
4	1.000 (0.000)	1.000 (0.011)	1.000 (0.000)	0.975 (0.076)	1.000 (0.000)	0.975 (0.076)
10	0.994 (0.051)	0.992 (0.050)	1.000 (0.000)	0.999 (0.009)	1.000 (0.000)	0.999 (0.009)
20	0.925 (0.163)	0.940 (0.141)	1.000 (0.000)	1.000 (0.003)	1.000 (0.000)	1.000 (0.003)
40	0.614 (0.148)	0.650 (0.164)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
100	0.540 (0.028)	0.545 (0.031)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
			Mean (SD) of Computing Time (s)			
4	0.009 (0.010)	0.009 (0.010)	0.009 (0.011)	0.009 (0.010)	0.011 (0.010)	0.011 (0.010)
10	0.012 (0.010)	0.014 (0.010)	0.014 (0.010)	0.014 (0.010)	0.023 (0.010)	0.023 (0.011)
20	0.020 (0.009)	0.030 (0.011)	0.030 (0.011)	0.030 (0.012)	0.063 (0.013)	0.063 (0.013)
40	0.052 (0.012)	0.087 (0.013)	0.088 (0.013)	0.089 (0.012)	0.207 (0.017)	0.206 (0.018)
100	0.309 (0.022)	0.539 (0.023)	0.547 (0.022)	0.546 (0.024)	1.270 (0.033)	1.278 (0.032)

Table 4.6: Number of Times Each Variable Is Used For Cutting

# of Intervals	Variable	H	EH	GSNEH	GNEH	LSNEH	LNEH
4	X_1	1000	993	1000	896	1000	896
	X_2	0	7	0	104	0	104
10	X_1	986	977	1000	997	1000	997
	X_2	14	23	0	3	0	3
20	X_1	821	843	1000	1000	1000	1000
	X_2	179	157	0	0	0	0
40	X_1	118	166	1000	1000	1000	1000
	X_2	882	834	0	0	0	0
100	X_1	0	0	1000	1000	1000	1000
	X_2	1000	1000	0	0	0	0

It can be seen that Hausdorff and Euclidean Hausdorff distances perform worse as the number of intervals increases, because they tend to use X_2 as the splitting variable when the number of intervals increases; when the number of intervals reaches 100, they use X_2 for cutting all the time. However, from Figure 4.4, it is obvious that X_1 should be chosen as the cutting variable. Otherwise, the result cannot be right since the two underlying clusters have the same mean and variance along variable X_2.

All other Normalized distances still perform well in this case since the impact of X_1's large scale is eliminated by doing normalization.

Case 4: A Bad Example For Global (Span) Normalized Euclidean Hausdorff Distance

In this simulation study, we want to compare the Global Normalization with the Local Normalization Hausdorff distance. As mentioned before, $|\mathcal{Y}_j|$ and H_j from Equations (4.3) and (4.4) can either rely on the the total number of observations in the data set, or they can rely on the number of observations within the cluster that the clustering process is going through. Chavent (1998) mentioned the different performances between these two choices by a real example with classical data. Advantages and disadvantages between the two choices have not yet been discussed in the literature, and the global normalized distances have not been applied to symbolic data yet.

The first 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix} 2 \\ 5 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right), \tag{4.11}$$

the next 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix} 7 \\ 5 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right), \tag{4.12}$$

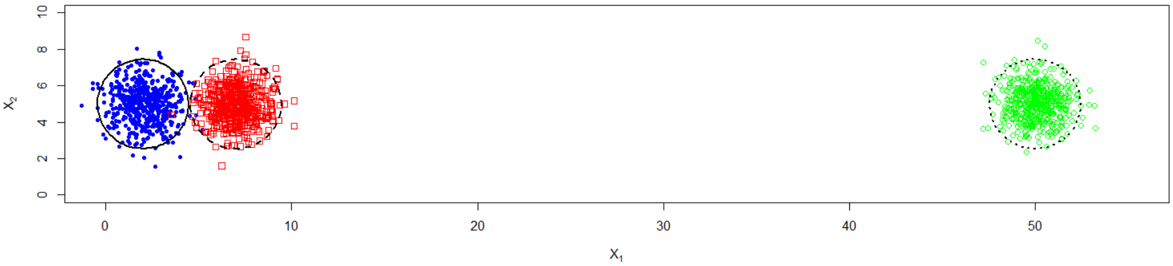


Figure 4.5: Simulated Data from Equations (4.11), (4.12) and (4.13)

and the last 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix}50\\5\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right). \tag{4.13}$$

See Figure 4.5 for their distributions. The interval valued observations aggregated from these data are

$$\{\mathbf{X}(1),...,\mathbf{X}(10)\}\sim N_2\left(\begin{pmatrix}2\\5\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right),$$

$$\{\mathbf{X}(11),...,\mathbf{X}(20)\}\sim N_2\left(\begin{pmatrix}7\\5\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right),$$

and

$$\{\mathbf{X}(21),...,\mathbf{X}(30)\}\sim N_2\left(\begin{pmatrix}50\\5\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$$

with 100 classical points to construct an interval. It is obvious that the data are overdispersed along variable X_1.

Figure 4.6 is the clustering result of one simulated data set using the six different distances. Except for the two Global Normalized distances (Figure 4.6 (C) and (D)), other distances are able to group the observations successfully into three underlying clusters. All the distances separate $\{\mathbf{X}(21), ..., \mathbf{X}(30)\}$ (hollow round dots on the right in Figure 4.5) from other observations using X_1 as the cutting variable at the 1^{st} stage of the clustering. Since the data are dispersed along X_1, $|\mathcal{Y}_1|$ and H_1 are very large at the 1^{st} stage. At the 2^{nd} stage, $\{\mathbf{X}(1), ..., \mathbf{X}(20)\}$ are supposed to be divided into two groups like $\{\mathbf{X}(1), ..., \mathbf{X}(10)\}$ (solid round dots on the left in Figure 4.5) and $\{\mathbf{X}(11), ..., \mathbf{X}(20)\}$ (hollow square dots on the left in Figure 4.5) along X_1. It is obvious that the variation along variable X_1 will be much smaller after removing $\{\mathbf{X}(21), ..., \mathbf{X}(30)\}$ (see Figure 4.5). Nevertheless, GSNEH and GNEH distances will still use the same $|\mathcal{Y}_1|$ and H_1 from the 1^{st} stage to calculate distances when separating $\{\mathbf{X}(1), ..., \mathbf{X}(20)\}$. As discussed in Section 4.2, the Global Normalized distances between two observations $\mathbf{X}(i_1)$ and $\mathbf{X}(i_2)$ at the 2^{nd} stage are

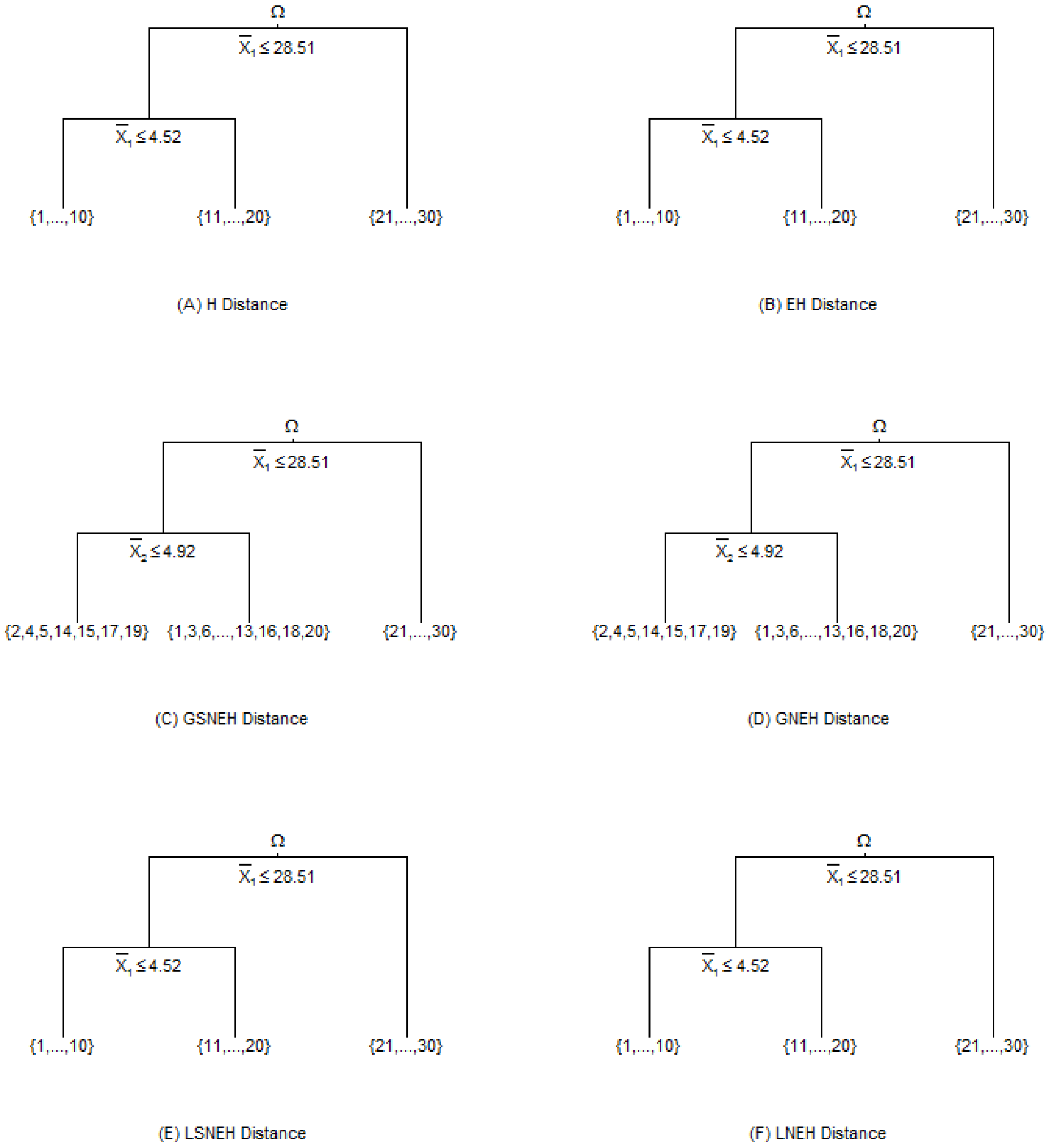
$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{ \sum_{j=1}^{2} \left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{|\mathcal{Y}_j|} \right]^2 \right\}^{1/2}$$

and

$$d(\mathbf{X}(i_1), \mathbf{X}(i_2)) = \left\{ \sum_{j=1}^{2} \left[\frac{d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))}{H_j} \right]^2 \right\}^{1/2}.$$

However, $|\mathcal{Y}_1|$ and H_1 are too large for the 2^{nd} stage. Therefore, $d(\mathbf{X}(i_1), \mathbf{X}(i_2))$ will be dominated by variable X_2 since $\frac{d_1(\mathbf{X}(i_1),\mathbf{X}(i_2))}{|\mathcal{Y}_1|}$ and $\frac{d_2(\mathbf{X}(i_1),\mathbf{X}(i_2))}{H_1}$ are too small due to the large $|\mathcal{Y}_1|$ and H_1 values. Hence, the clustering process will choose variable X_2 for splitting observations while $\{\mathbf{X}(1), ..., \mathbf{X}(20)\}$ are supposed to be divided into two groups along X_1.

By contrast, Local Normalized distances adjust the $|\mathcal{Y}_1|$ and H_1 at the 2^{nd} stage according to which cluster they are in. After separating $\{\mathbf{X}(21), ..., \mathbf{X}(30)\}$ from other observations at stage 1, $|\mathcal{Y}_1|$ and H_1 are much smaller at stage 2 if LSNEH and LNEH distances are used since the variation of the left two clusters (in Figure 4.5) along X_1 is much smaller. Thus,



(A) H Distance

(B) EH Distance

(C) GSNEH Distance

(D) GNEH Distance

(E) LSNEH Distance

(F) LNEH Distance

Figure 4.6: Clustering Result of One Data Set from Simulation Case 4

X_2 is not able to dominate the distances and the cutting process and hence X_1 is correctly chosen as the splitting variable.

Case 5: Outliers

In this section, a data set with outliers will be considered. The first 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix}12\\5\end{pmatrix},\begin{pmatrix}2&0\\0&2\end{pmatrix}\right), \tag{4.14}$$

the next 1000 classical random samples are simulated from

$$N_2\left(\begin{pmatrix}18\\5\end{pmatrix},\begin{pmatrix}2&0\\0&2\end{pmatrix}\right), \tag{4.15}$$

and the last 100 classical random samples are simulated from

$$N_2\left(\begin{pmatrix}10\\30\end{pmatrix},\begin{pmatrix}0.1&0\\0&0.1\end{pmatrix}\right). \tag{4.16}$$

The last 100 classical samples are considered outliers here. See Figure 4.7 for their distributions. The interval valued observations aggregated from these data are

$$\{\mathbf{X}(1),...,\mathbf{X}(10)\} \sim N_2\left(\begin{pmatrix}12\\5\end{pmatrix},\begin{pmatrix}2&0\\0&2\end{pmatrix}\right),$$

$$\{\mathbf{X}(11),...,\mathbf{X}(20)\} \sim N_2\left(\begin{pmatrix}18\\5\end{pmatrix},\begin{pmatrix}2&0\\0&2\end{pmatrix}\right),$$

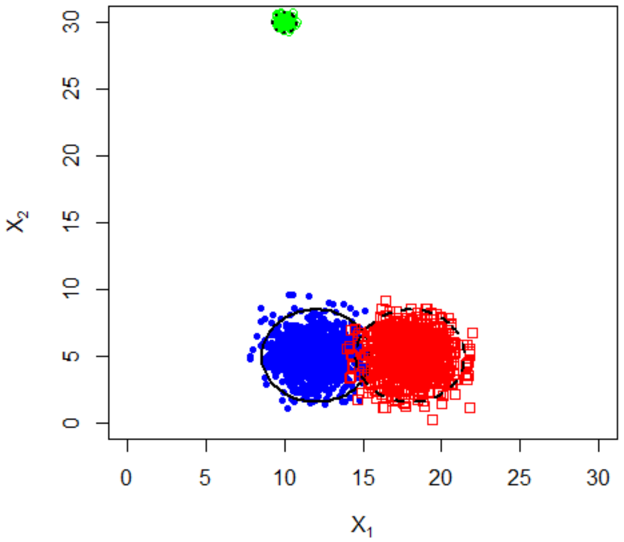


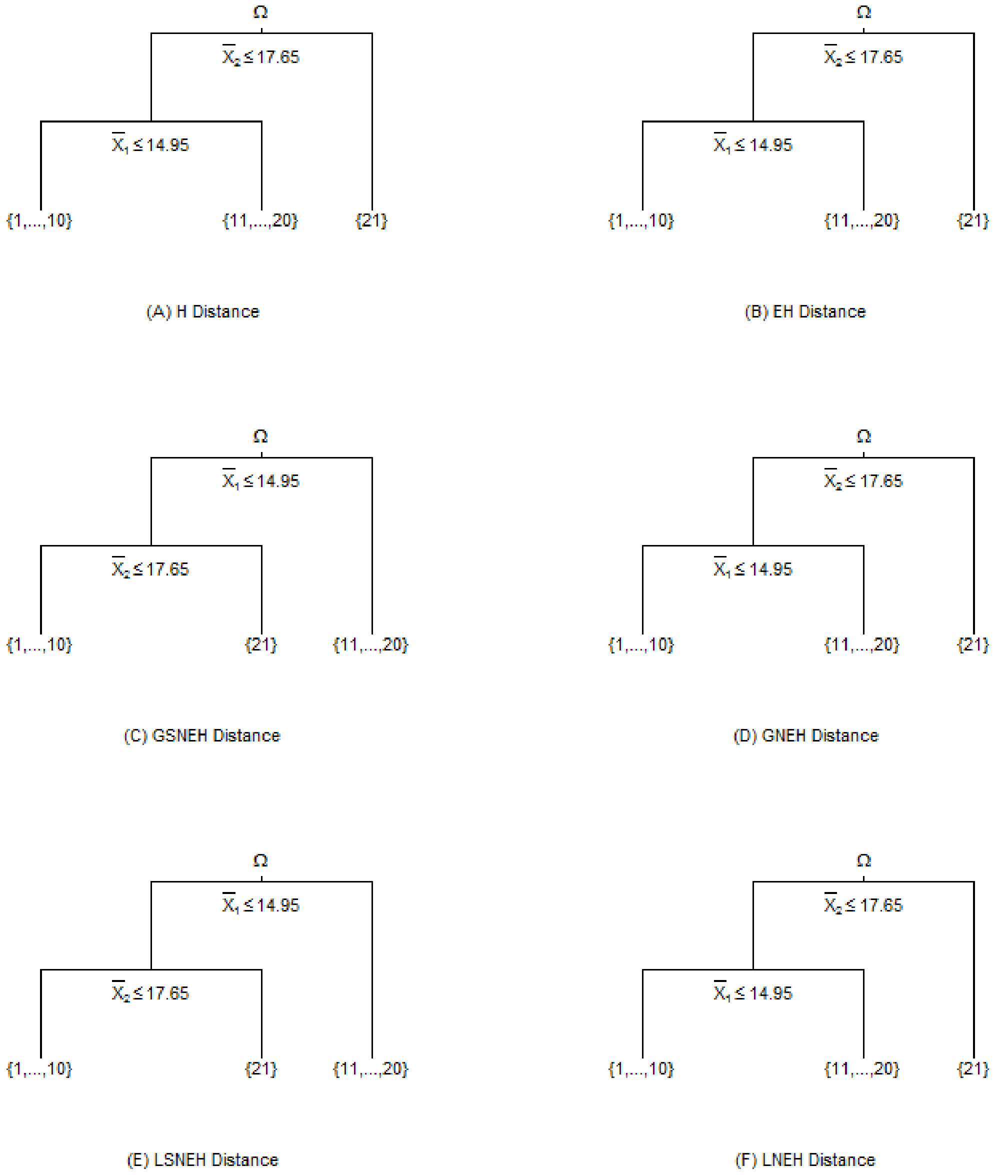
Figure 4.7: Simulated Data from Equations (4.14), (4.15) and (4.16)

and

$$\{\mathbf{X}(21)\} \sim N_2\left(\begin{pmatrix} 10 \\ 30 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}\right)$$

with 100 classical points to construct an interval. Observation $\mathbf{X}(21)$ is considered an outlier here.

Figure 4.8 is the clustering result of one simulated data set using the six different distances. Unlike GSNEH and LSNEH distances (Figure 4.8 (C) and (E)), other distances separate the outlier at the 1^{st} stage. The GSNEH and LSNEH distances do not exclude the outlier until the 2^{nd} stage. This is because the variation along variable X_2 is cancelled out by the span normalization factor $|\mathcal{Y}_2|$ at the 1^{st} stage (recall that $|\mathcal{Y}_2| = \max_i(b_{i2}) - \min_i(a_{i2})$) and hence X_1 dominates the clustering process at the 1^{st} stage. This is similar to the discussion in Section 4.2. This brings up the potential problem that if the clustering is



(A) H Distance

(B) EH Distance

(C) GSNEH Distance

(D) GNEH Distance

(E) LSNEH Distance

(F) LNEH Distance

Figure 4.8: Clustering Result of One Data Set from Simulation Case 5

stopped at stage 2, the outlier cannot be separated out and will be grouped with observations $\{\mathbf{X}(1), ..., \mathbf{X}(10)\}$. Other distances exclude the outlier at the first place, which is preferable. Therefore, it is always beneficial to continue the clustering process for more steps even if when the target structure has been achieved so that potential outliers can be eliminated from the main clusters.

Case 6: A Special Example For Symbolic Data

This simulation study considers a special case when the centers of the underlying clusters overlaps. The simulation process is the same as in Section 4.2, except that the bivariate normal distributions to generate random samples are

$$N_2\left(\begin{pmatrix} 15 \\ 15 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) \tag{4.17}$$

and

$$N_2\left(\begin{pmatrix} 15 \\ 15 \end{pmatrix}, \begin{pmatrix} 5 & 0 \\ 0 & 30 \end{pmatrix}\right). \tag{4.18}$$

Figure 4.9 is the scatter plot of the simulated data. Table 4.7 shows the means and standard deviations of accuracy and computing time of doing clustering on 4, 10, 20, 40 and 100 aggregated intervals by using the six different distances on 1000 sets of simulated samples. Table 4.8 shows the number of times each variable is used as the cutting variable.

All the distances perform similarly to each other. Obviously, it will be very hard for the divisive monothemic clustering method to do clustering on a data set like this if they are treated as classical points since the method divides the data into two separate groups at each step and uses just one variable at each step to do the splitting. The accuracy will most probably be around 0.5, not to mention the tremendous computing time needed to group

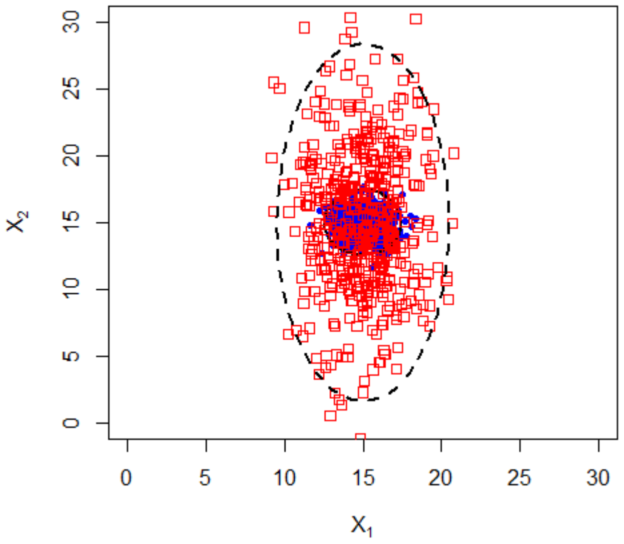


Figure 4.9: Simulated Data from Equations (4.17) and (4.18)

2000 observations since the method stops at every observation and calculates the distances and variations at each stop when performing the clustering algorithm. The time increases quadratically as the number of observations increases. However, by aggregating them into interval observations, the accuracy can even reach 0.913 when the number of intervals is 4. Still, accuracies decrease as the number of intervals increases, and arrive at 0.7 when there are 100 intervals. Although a result with an accuracy at 0.7 is not ideal, considering the improvement of accuracy and computing time compared with the classical data, it is worthwhile to aggregate classical points into intervals for the clustering analysis.

Notice that in Table 4.8, all distances prefer to use X_2 as the splitting variable. This is reasonable since by looking at Figure 4.9, variable X_2 is the direction where the big difference between the two clusters is located.

Table 4.7: Mean and Standard Deviation of Accuracy and Computing Time

# of Intervals	H	EH	GSNEH	GNEH	LSNEH	LNEH
			Mean (SD) of Accuracy			
4	0.861 (0.124)	0.913 (0.119)	0.913 (0.119)	0.913 (0.119)	0.913 (0.119)	0.913 (0.119)
10	0.798 (0.083)	0.822 (0.077)	0.822 (0.077)	0.823 (0.077)	0.822 (0.077)	0.823 (0.077)
20	0.760 (0.062)	0.770 (0.060)	0.772 (0.059)	0.771 (0.059)	0.772 (0.059)	0.771 (0.059)
40	0.727 (0.045)	0.731 (0.044)	0.733 (0.043)	0.732 (0.044)	0.733 (0.043)	0.732 (0.044)
100	0.698 (0.031)	0.699 (0.030)	0.700 (0.030)	0.698 (0.032)	0.700 (0.030)	0.698 (0.032)
			Mean (SD) of Computing Time (s)			
4	0.009 (0.010)	0.009 (0.010)	0.009 (0.010)	0.010 (0.010)	0.011 (0.010)	0.011 (0.010)
10	0.012 (0.010)	0.014 (0.010)	0.015 (0.010)	0.014 (0.010)	0.023 (0.010)	0.024 (0.011)
20	0.021 (0.011)	0.031 (0.012)	0.032 (0.012)	0.031 (0.011)	0.064 (0.014)	0.064 (0.014)
40	0.052 (0.012)	0.089 (0.013)	0.089 (0.014)	0.091 (0.014)	0.211 (0.020)	0.209 (0.019)
100	0.313 (0.025)	0.546 (0.035)	0.557 (0.034)	0.556 (0.035)	1.295 (0.074)	1.299 (0.073)

Summary

From the simulation studies, we can tell that no particular distance can outperform the other distances all the time. Each Hausdorff distance has its certain limitations. Their advantages and disadvantages can be summarized as follows:

- **Hausdorff Distance**
  - **Pros:** Fastest.
  - **Cons:** Sensitive to variables with large scale. The clustering process can easily be controlled by the large-scaled variables even if it is the wrong choice.
- **Euclidean Hausdorff Distance**
  - **Pros:** Fast.
  - **Cons:** Sensitive to variables with large scale. The clustering process can easily be controlled by the large-scaled variables even if it is the wrong choice.
- **Global Span Normalized/Normalized Euclidean Hausdorff Distance**

Table 4.8: Number of Times Each Variable Is Used For Cutting

# of Intervals	Variable	H	EH	GSNEH	GNEH	LSNEH	LNEH
4	X_1	0	243	252	264	252	264
	X_2	1000	757	748	736	748	736
10	X_1	0	314	343	360	343	360
	X_2	1000	686	657	640	657	640
20	X_1	0	158	210	246	210	246
	X_2	1000	842	790	754	790	754
40	X_1	0	55	118	177	118	177
	X_2	1000	945	882	823	882	823
100	X_1	0	1	43	123	43	123
	X_2	1000	999	957	877	957	877

  - **Pros:** Fast.
  - **Cons:** Can give the wrong result after the 1^{st} stage, especially when underlying clusters are dispersed along one variable.

    [Overdispersed Variable: When all variables are of the same magnitude, the standard deviation of the overdispersed variable is much larger (e.g., at least 5 times larger) than other variables' standard deviations.]

- **Local Span Normalized Euclidean Hausdorff Distance**
  - **Pros:** Not sensitive to variables with large scales; robust to large variance (variable with large variance will not be ignored during the clustering process).
  - **Cons:** Slow; variable with outliers may be ignored by the clustering process, hence not able to identify outliers as early as possible.

- **Local Normalized Euclidean Hausdorff Distance**
  - **Pros:** Not sensitive to variables with large scales; robust to outliers (variable

with outliers at along it will not be ignored during the clustering process).

- **Cons:** Slow; variable with large variance may be ignored by the clustering process.

As a summary, Global Normalized distances are not recommended since they cannot reveal the real variation within each cluster after the 1^{st} stage. If we are conscious of variables with significant larger scales, we can start with LSNEH and LNEH distances. Then, if large variance exists along some variable, the LSNEH distance should be considered. Or, if outliers are observed along some variable, the LNEH distance should be considered. Otherwise, these two distances perform similarly. If variables are of the same magnitude, H and EH distances should be preferred since they are much faster than LSNEH and LNEH distances and do not have the limitations that LSNEH and LNEH distances have.

4.3 Application

The Hausdorff distance and its related distances will be applied to practical data sets introduced below. Comparisons among the six distances will also be made.

China Temperatures

Table 4.9 shows temperature data from 15 cities of China in 1988. The data set consists of minimum and maximum temperatures for each month. Here, variables $X_1 - X_{12}$ represent twelve months, i.e., January - December, and X_{13} is the elevation. The unit of temperatures is Celsius degree. Each observation is of equal weight here. These data are extracted from a larger data set. The full data set contains observations for many more stations, more variables and more years, which can be found at <http://rda.ucar.edu/datasets/ds578.5/>.

Table 4.9: China Monthly Temperatures and Elevations

Station Name	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1. BoKeTu	[-23.4, -15.5]	[-24.0, -14.0]	[-17.6, -4.7]	[-4.5, 9.5]	[1.9, 15.2]	[9.3, 23.0]	[12.9, 23.0]
2. Hailaer	[-28.4, -19.1]	[-29.6, -18.1]	[-20.2, -7.9]	[-2.9, 10.0]	[3.8, 16.5]	[12.5, 24.2]	[14.7, 25.4]
3. LaSa	[-8.4, 9.0]	[-3.5, 11.2]	[-1.2, 13.7]	[1.7, 16.4]	[5.9, 21.5]	[9.7, 24.8]	[10.8, 23.2]
4. KunMing	[2.8, 16.6]	[4.0, 19.4]	[6.7, 21.4]	[10.4, 23.4]	[15.9, 25.3]	[16.4, 25.3]	[16.9, 24.4]
5. TengChong	[2.3, 16.9]	[4.0, 18.9]	[6.4, 21.3]	[9.9, 24.3]	[15.0, 22.6]	[17.2, 24.1]	[17.4, 22.8]
6. WuZhou	[10.0, 17.7]	[9.1, 14.7]	[10.4, 16.2]	[15.8, 23.9]	[22.2, 30.3]	[23.9, 34.3]	[24.2, 33.8]
7. GuangZhou	[12.2, 20.6]	[11.4, 17.7]	[12.7, 17.9]	[17.4, 24.3]	[24.2, 30.3]	[25.1, 32.8]	[25.3, 33.6]
8. NanNing	[11.5, 17.7]	[10.3, 14.3]	[12.1, 17.3]	[17.8, 24.2]	[23.8, 30.7]	[24.9, 33.2]	[25.8, 33.5]
9. ShanTou	[11.8, 19.2]	[12.2, 17.7]	[12.7, 17.8]	[16.4, 22.7]	[22.5, 28.1]	[24.9, 31.5]	[25.6, 32.6]
10. HaiKou	[17.3, 23.0]	[17.3, 23.0]	[17.1, 23.7]	[19.9, 25.9]	[25.3, 33.4]	[25.5, 33.2]	[25.8, 34.0]
11. ZhanJiang	[15.8, 21.5]	[14.6, 19.4]	[14.4, 18.8]	[19.2, 24.2]	[26.0, 31.7]	[26.2, 32.3]	[27.1, 33.0]
12. MuDanJiang	[-18.4, -7.5]	[-19.2, -7.3]	[-10.7, 2.4]	[-0.1, 13.2]	[6.8, 18.5]	[13.7, 26.1]	[17.0, 26.5]
13. HaErBin	[-20.0, -9.6]	[-22.3, -10.0]	[-10.8, 1.7]	[0.2, 11.9]	[7.5, 19.0]	[14.6, 27.3]	[17.8, 27.2]
14. QiQiHaEr	[-22.7, -11.5]	[-21.1, -9.6]	[-11.5, 0.7]	[0.6, 13.8]	[7.8, 18.9]	[16.7, 27.4]	[19.1, 26.5]
15. NenJiang	[-27.9, -16.0]	[-27.7, -12.9]	[-16.5, -3.4]	[-1.5, 12.0]	[4.5, 17.5]	[13.7, 26.7]	[16.1, 25.0]

Station Name	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}
1. BoKeTu	[10.8, 23.6]	[4.1, 17.0]	[-4.0, 8.9]	[-13.5, -4.2]	[-21.1, -13.1]	[14.78, 14.78]
2. Hailaer	[13.7, 24.8]	[5.3, 17.6]	[-3.2, 9.8]	[-13.8, -3.7]	[-26.0, -17.2]	[12.26, 12.26]
3. LaSa	[10.2, 20.8]	[7.7, 19.9]	[1.4, 18.7]	[-4.9, 11.4]	[-7.5, 8.8]	[73.16, 73.16]
4. KunMing	[16.6, 23.8]	[14.0, 21.7]	[12.4, 19.7]	[7.4, 16.3]	[3.8, 15.5]	[37.82, 37.82]
5. TengChong	[17.4, 22.5]	[16.3, 23.2]	[14.5, 23.5]	[9.4, 20.0]	[4.7, 17.8]	[32.96, 32.96]
6. WuZhou	[23.4, 32.1]	[22.3, 31.8]	[19.2, 27.6]	[12.4, 23.2]	[9.8, 21.6]	[2.38, 2.38]
7. GuangZhou	[25.0, 31.9]	[24.1, 31.4]	[21.0, 28.5]	[14.2, 23.4]	[10.9, 22.2]	[0.14, 0.14]
8. NanNing	[25.1, 31.9]	[24.2, 31.5]	[20.3, 26.9]	[14.8, 24.3]	[12.4, 22.4]	[1.44, 1.44]
9. ShanTou	[24.9, 30.8]	[23.8, 30.0]	[20.4, 27.3]	[13.9, 22.6]	[10.0, 19.7]	[0.02, 0.02]
10. HaiKou	[25.0, 32.6]	[25.1, 31.5]	[23.0, 26.7]	[18.6, 22.6]	[16.3, 21.6]	[0.28, 0.28]
11. ZhanJiang	[25.6, 31.1]	[25.7, 31.9]	[22.0, 27.2]	[16.1, 23.2]	[14.5, 21.8]	[0.50, 0.50]
12. MuDanJiang	[18.9, 28.4]	[10.3, 21.4]	[0.6, 13.1]	[-8.9, 1.5]	[-18.9, -7.8]	[4.82, 4.82]
13. HaErBin	[17.7, 26.3]	[8.9, 20.8]	[-0.2, 12.5]	[-9.6, 0.4]	[-21.6, -10.1]	[3.44, 3.44]
14. QiQiHaEr	[19.1, 26.9]	[10.5, 19.8]	[1.9, 12.2]	[-9.0, 0.2]	[-17.8, -9.1]	[2.92, 2.92]
15.NenJiang	[14.5, 26.0]	[6.9, 19.2]	[-2.6, 10.9]	[-14.4, -3.6]	[-26.1, -13.8]	[4.84, 4.84]

Each station's average temperatures of four seasons for 1988 are shown in Table 4.10, and each station's temperature variations $[max(temperature) - min(temperature)]$ of four seasons for 1988 are shown in Table 4.11.

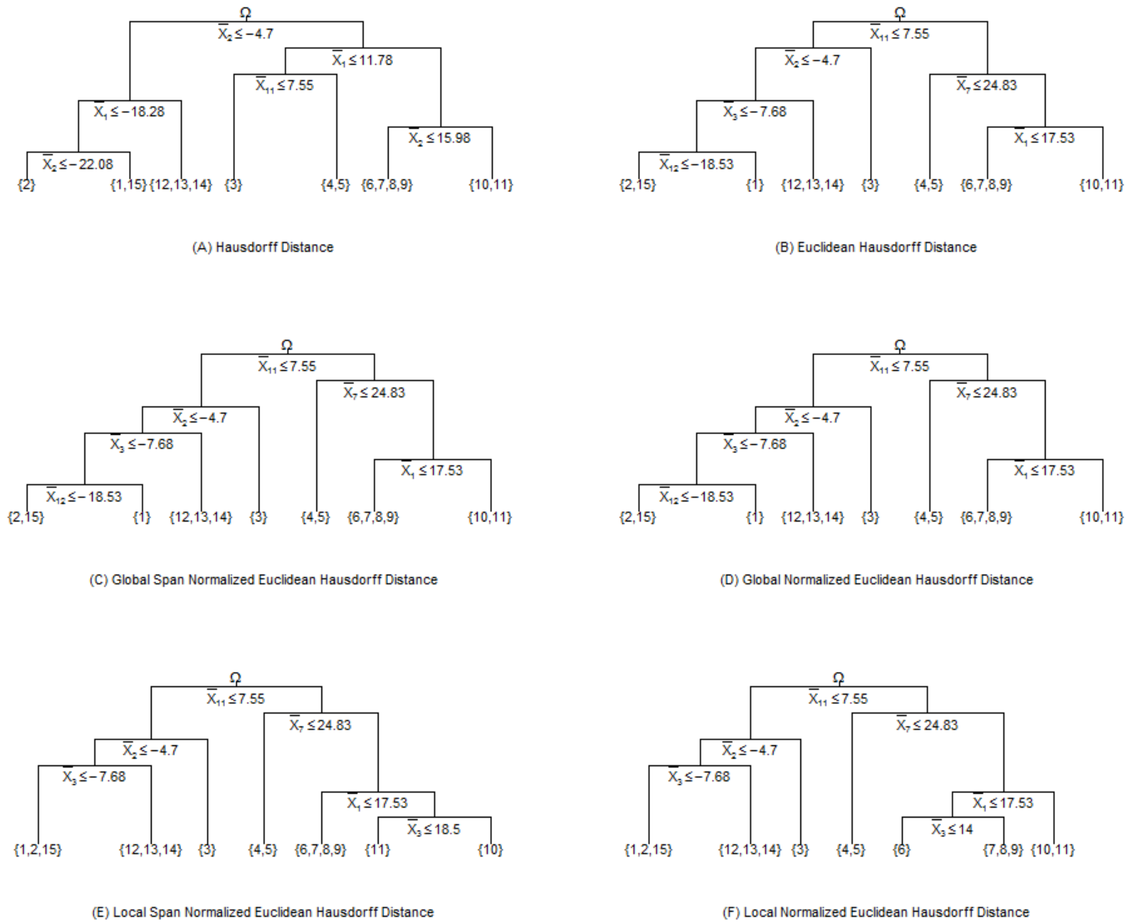
Table 4.10: China Seasonal and Annual Average Temperatures - 1988

Station	Winter	Spring	Summer	Autumn	Annual
Name	Average	Average	Average	Average	Average
1. BoKeTu	-18.5	0.0	17.1	1.4	0.0
2. Hailaer	-23.1	-0.1	19.2	2.0	-0.5
3. LaSa	1.6	9.7	16.6	9.0	9.2
4. KunMing	10.4	17.2	20.6	15.3	15.8
5. TengChong	10.8	16.6	20.2	17.8	16.4
6. WuZhou	13.8	19.8	28.6	22.8	21.2
7. GuangZhou	15.8	21.1	29.0	23.8	22.4
8. NanNing	14.8	21.0	29.1	23.7	22.1
9. ShanTou	15.1	20.0	28.4	23.0	21.6
10. HaiKou	19.8	24.2	29.4	24.6	24.5
11. ZhanJiang	17.9	22.4	29.2	24.4	23.5
12. MuDanJiang	-13.2	5.0	21.8	6.3	5.0
13. HaErBin	-15.6	4.9	21.8	5.5	4.2
14. QiQiHaEr	-15.3	5.1	22.6	5.9	4.6
15. NenJiang	-20.7	2.1	20.3	2.7	1.1

Figure 4.10 is the result of clustering the China temperature data of Table 4.9 using the Hausdorff distance, Euclidean Hausdorff distance, Global and Local Span Normalized Euclidean Hausdorff distance and Global and Local Normalized Euclidean Hausdorff distance based on the $p = 12$ temperature variables.

According to Tables 4.9, 4.10 and 4.11, it is evident that the twelve variables used here are of the same magnitude. Therefore, based on the conclusion from Section 4.2, all the distances except for GSNEH and GNEH distances can be used here and they should give similar results.

From Figure 4.10, it can be seen that the two partitions we obtained by using the Hausdorff distance [Figure 4.10 (A)] and the Euclidean Hausdorff distance [Figure 4.10 (B)] are


Ω
$\overline{X}_2 \le -4.7$
$\overline{X}_1 \le 11.78$
$\overline{X}_{11} \le 7.55$
$\overline{X}_1 \le -18.28$
$\overline{X}_2 \le 15.98$
$\overline{X}_2 \le -22.08$
{2}
{1,15}
{12,13,14}
{3}
{4,5}
{6,7,8,9}
{10,11}
Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_2 \le -4.7$
$\overline{X}_7 \le 24.83$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_{12} \le -18.53$
{2,15}
{1}
{12,13,14}
{3}
{4,5}
{6,7,8,9}
{10,11}
Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_7 \le 24.83$
$\overline{X}_2 \le -4.7$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_{12} \le -18.53$
{2,15}
{1}
{12,13,14}
{3}
{4,5}
{6,7,8,9}
{10,11}
Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_7 \le 24.83$
$\overline{X}_2 \le -4.7$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_{12} \le -18.53$
{2,15}
{1}
{12,13,14}
{3}
{4,5}
{6,7,8,9}
{10,11}
Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_7 \le 24.83$
$\overline{X}_2 \le -4.7$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_3 \le 18.5$
{1,2,15}
{12,13,14}
{3}
{4,5}
{6,7,8,9}
{11}
{10}
Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_7 \le 24.83$
$\overline{X}_2 \le -4.7$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_3 \le 14$
{1,2,15}
{12,13,14}
{3}
{4,5}
{6}
{7,8,9}
{10,11}

(A) Hausdorff Distance

(B) Euclidean Hausdorff Distance

(C) Global Span Normalized Euclidean Hausdorff Distance

(D) Global Normalized Euclidean Hausdorff Distance

(E) Local Span Normalized Euclidean Hausdorff Distance

(F) Local Normalized Euclidean Hausdorff Distance

Figure 4.10: China Temperature Clustering Results Based on $(X_1, ..., X_{12})$

Table 4.11: China Seasonal and Annual Temperature Variations - 1988

Station Name	Winter Range	Spring Range	Summer Range	Autumn Range	Annual Range
1.BoKeTu	10.9	32.8	14.3	30.5	47.6
2.Hailaer	12.4	36.7	12.9	31.4	55.0
3.LaSa	19.6	22.7	15.1	24.8	33.2
4.KunMing	16.6	18.6	8.9	14.3	22.5
5.TengChong	16.6	17.9	6.9	14.1	22.0
6.WuZhou	12.5	19.9	10.9	19.4	25.2
7.GuangZhou	11.3	17.6	8.6	17.2	22.7
8.NanNing	12.1	18.6	8.6	16.7	23.2
9.ShanTou	9.7	15.4	7.7	16.1	22.6
10.HaiKou	6.7	16.3	9.0	12.9	17.7
11.ZhanJiang	7.3	17.3	7.4	15.8	18.6
12.MuDanJiang	11.9	29.2	14.7	30.3	47.6
13.HaErBin	12.7	29.8	12.7	30.4	49.6
14.QiQiHaEr	13.6	30.4	10.7	28.8	50.1
15.NenJiang	15.0	34.0	13.0	33.6	54.6

almost the same, except for station 15 being clustered with station 1 in Figure 4.10 (A) and with station 2 in Figure 4.10 (B). These clusters are consistent with the annual and seasonal average temperatures shown in Table 4.10 and the corresponding variations shown in Table 4.11. We observe that stations 6 - 11 are successfully separated from the others since they are warmer than are the other stations across the year. Stations 10 and 11 are further separated from stations 6 - 9 because they are the hottest stations across the year and their annual temperature variations are the smallest. Stations 1, 2 and 15 are grouped together since they are the coldest stations according to their annual average and their temperature variations are similar to each other (large). Although the clustering result has 7 clusters with the last cutting on stations 1, 2 and 15, the clustering could have been stopped at the 5^{th} stage if the small differences among stations 1, 2 and 15 are not important to researchers. Stations

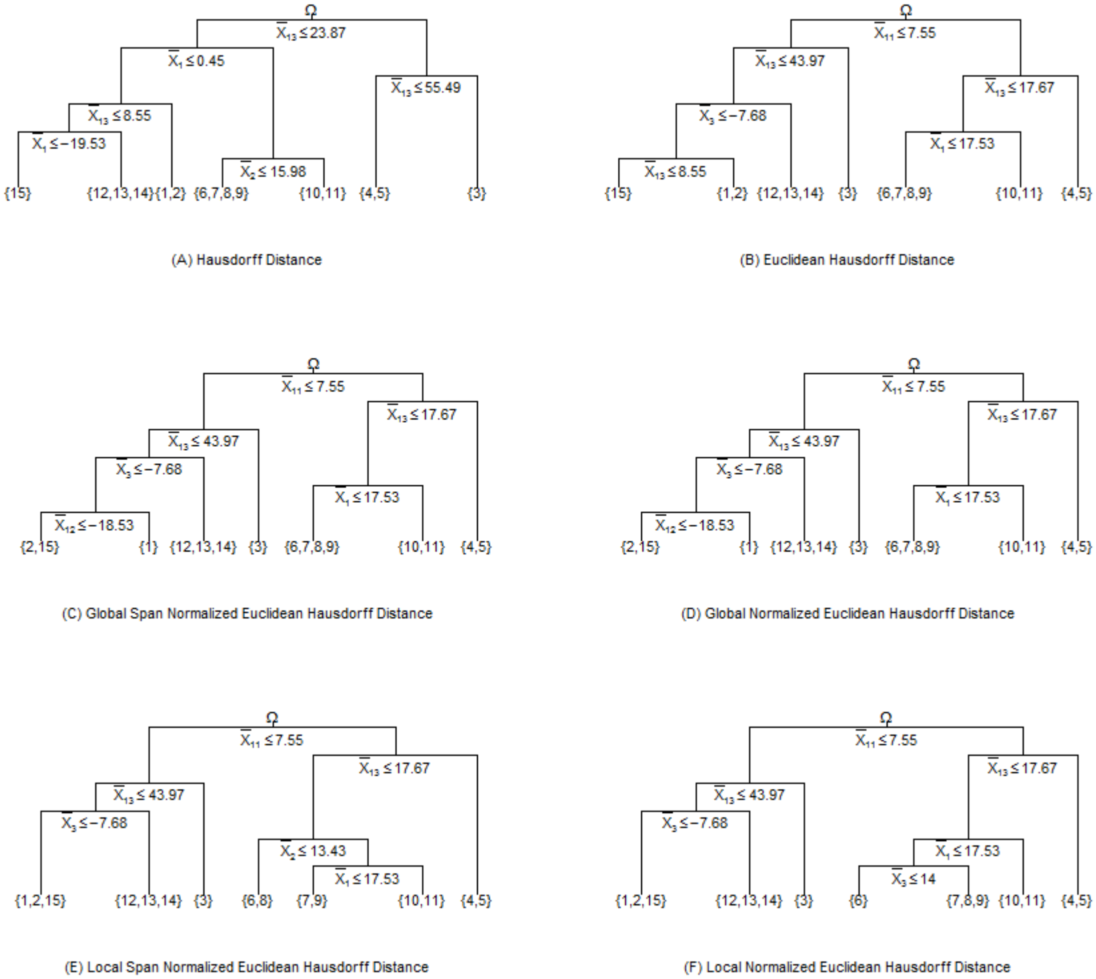
12, 13 and 14 are grouped together because they are the second coldest stations on annual temperature averages and their temperature variations are also similar (large). Stations 4 and 5 are clustered together with moderate temperatures across the year compared to other stations. Station 3, LaSa, is a special case with temperature being median among stations during the winter, spring and autumn and the lowest during the summer. Also, its temperature variation is the largest during the winter and summer while being moderate during the spring and autumn.

The partitions (Figure 4.10 (C) and (D)) using the Global Span Normalized Euclidean Hausdorff distance and the Global Normalized Euclidean Hausdorff distance are the same as using the Euclidean Hausdorff distance (Figure 4.10 (B)) here. By observing Table 4.9, we can see that the data are not overdispersed along any of the twelve variables. Hence the GSNEH and GNEH distances give the same result as the EH distance.

The partition [Figure 4.10 (E)] using the Local Span Normalized Euclidean Hausdorff distance is very similar to Figure 4.10 (A) and Figure 4.10 (B) except for stations 10 and 11 being separated at the 6^{th} stage. Also, the partition [Figure 4.10 (F)] using the Local Normalized Euclidean Hausdorff distance is very similar to Figure 4.10 (A) and Figure 4.10 (B) except for stations 6 and {7, 8, 9} being separated at the 6^{th} stage. Again, the clustering could have been stopped before the 6^{th} stage and thus the H, EH, LSNEH and LNEH distances would give the same clustering structures.

Clustering results of using the six different distances based on all thirteen variables (temperatures and elevation) are shown in Figure 4.11. Now, the variable elevation is also being considered during clustering.

Let us look at the result from using the Hausdorff distance [Figure 4.11 (A)] first. Comparing Figures 4.10 (A) and 4.11 (A), we see that the biggest difference occurs at the 1^{st} stage. By using thirteen variables [Figure 4.11 (A)], the clustering algorithm separates station 3, 4, and 5 from other stations in the first place since they have higher elevations. After


Ω
$\overline{X}_{13} \le 23.87$
$\overline{X}_1 \le 0.45$
$\overline{X}_{13} \le 55.49$
$\overline{X}_{13} \le 8.55$
$\overline{X}_1 \le -19.53$
$\overline{X}_2 \le 15.98$
{15}
{12,13,14}
{1,2}
{6,7,8,9}
{10,11}
{4,5}
{3}

(A) Hausdorff Distance

Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_{13} \le 43.97$
$\overline{X}_{13} \le 17.67$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_{13} \le 8.55$
{15}
{1,2}
{12,13,14}
{3}
{6,7,8,9}
{10,11}
{4,5}

(B) Euclidean Hausdorff Distance

Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_{13} \le 17.67$
$\overline{X}_{13} \le 43.97$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_{12} \le -18.53$
{2,15}
{1}
{12,13,14}
{3}
{6,7,8,9}
{10,11}
{4,5}

(C) Global Span Normalized Euclidean Hausdorff Distance

Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_{13} \le 17.67$
$\overline{X}_{13} \le 43.97$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_{12} \le -18.53$
{2,15}
{1}
{12,13,14}
{3}
{6,7,8,9}
{10,11}
{4,5}

(D) Global Normalized Euclidean Hausdorff Distance

Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_{13} \le 17.67$
$\overline{X}_{13} \le 43.97$
$\overline{X}_3 \le -7.68$
$\overline{X}_2 \le 13.43$
$\overline{X}_1 \le 17.53$
{1,2,15}
{12,13,14}
{3}
{6,8}
{7,9}
{10,11}
{4,5}

(E) Local Span Normalized Euclidean Hausdorff Distance

Ω
$\overline{X}_{11} \le 7.55$
$\overline{X}_{13} \le 17.67$
$\overline{X}_{13} \le 43.97$
$\overline{X}_3 \le -7.68$
$\overline{X}_1 \le 17.53$
$\overline{X}_3 \le 14$
{1,2,15}
{12,13,14}
{3}
{6}
{7,8,9}
{10,11}
{4,5}

(F) Local Normalized Euclidean Hausdorff Distance

Figure 4.11: China Temperature Clustering Results Based on $(X_1, ..., X_{13})$

the 1^{st} stage, variable elevation is used again for twice in [Figure 4.11 (A)]. Out of the six times when a variable needs to be chosen as the cutting variable, elevation is chosen for three times in total. As discussed in Section 4.2, this is due to the larger scale that variable elevation has compared to other variables. Other than that, most clusters from the clustering result by using thirteen variables [Figure 4.11 (A)] are the same as the result by using the twelve temperature variables only [Figure 4.10 (A)], except for station 15. Station 15 is more similar to stations 12, 13 and 14 if elevation is also considered [Figure 4.11 (A)], while it is more similar to stations 1 and 2 if elevation is not considered [Figure 4.10 (A)] (stations 1 and 2 do have higher elevation). In Figure 4.11 (A), station 15 is finally separated from station 12, 13 and 14 due to the lower temperature in winter. If the researcher's interest is mainly in the elevation's impact on the clustering result, the Hausdorff distance can be considered since elevation is used three out of six times here.

The result from using the Euclidean Hausdorff distance [Figure 4.11 (B)] is the same as the result in Figure 4.10 (B) by using the twelve temperature variables if the cutting was stopped at the 5^{th} stage. Since station 15 is very dissimilar to stations 1 and 2 from the aspect of elevation, it is separated at the last stage [Figure 4.11 (B)], which is reasonable. The differences between Figure 4.11 (A) and (B) are similar to the differences between Figure 4.10 (A) and Figures 4.11 (A). Figure 4.11 (B) uses a temperature variable for cutting at the 1^{st} stage while Figure 4.11 (A) uses the elevation for cutting. Station 15 is more similar to stations 1 and 2 in Figure 4.11 (B) while it is more similar to stations 12, 13 and 14 in Figure 4.11 (A). Here, by using the Euclidean Hausdorff distance, variable elevation is still used three out of six times. Since variable elevation has larger variation along it and hence its scale is also a little larger compared to other variables, both Hausdorff and Euclidean Hausdorff distances will prefer to use it as the cutting variable (the reason is explained in Section 4.2). If the researcher's interest is mainly in the elevation's impact on the clustering result, the Euclidean Hausdorff distance can also be considered and it will give an alternative result for reference.

The partitions [Figure 4.11 (C) and (D)] using the Global Span Normalized Euclidean Hausdorff distance and the Global Normalized Euclidean Hausdorff distance are the same as using the Euclidean Hausdorff distance [Figure 4.11 (B)] if the clustering is stopped at the 5^{th} stage. And they are the same as Figure 4.10 (C) and (D).

The partition (E) in Figure 4.11 is different from Figure 4.11 (B) when the clustering process goes beyond the 4^{th} stage. In Figure 4.11 (E), stations 6 and 8 are separated from stations 7, 9, 10 and 11 at the 5^{th} stage because they are a little warmer in winter and their elevations are higher. Stations 7 and 9 are further separated from stations 10 and 11 since they are cooler across the year and their elevations are lower. After the 3^{rd} stage in Figure 4.11 (E), the variable elevation is no longer used as the cutting variable. As explained in Section 4.2, when the Local Normalized distances are used, since the variation along variable elevation is larger (see Table 4.9), variable elevation can be ignored by the clustering process when a cutting variable needs to be chosen due to the normalization.

The result of using LNEH distance [Figure 4.11 (F)] is also different from Figure 4.11 (B) when the clustering process goes beyond the 5^{th} stage. Also, Figure 4.11 (F) is similar to the result of using LSNEH distance [Figure 4.11 (E)], while the only difference occurs at the 6^{th} stage. The clustering process chooses to separate station 6 from stations 7, 8 and 9 since it is cooler across the year and the elevation is higher. Also, variable elevation is no longer chosen at the cutting variable after the 3^{rd} stage and the reason is analogous to using the LSNEH distance.

It is clear that by using the Local Normalized distances, the result can be quite different from that obtained when using unnormalized distances, especially after several stages of clustering. Unlike using the Hausdorff distance and the Euclidean Hausdorff distance, results from using the Local Normalized distances will not be influenced by variable elevation much and treat each variable equally, whether it is temperature or elevation. Hence, by using the

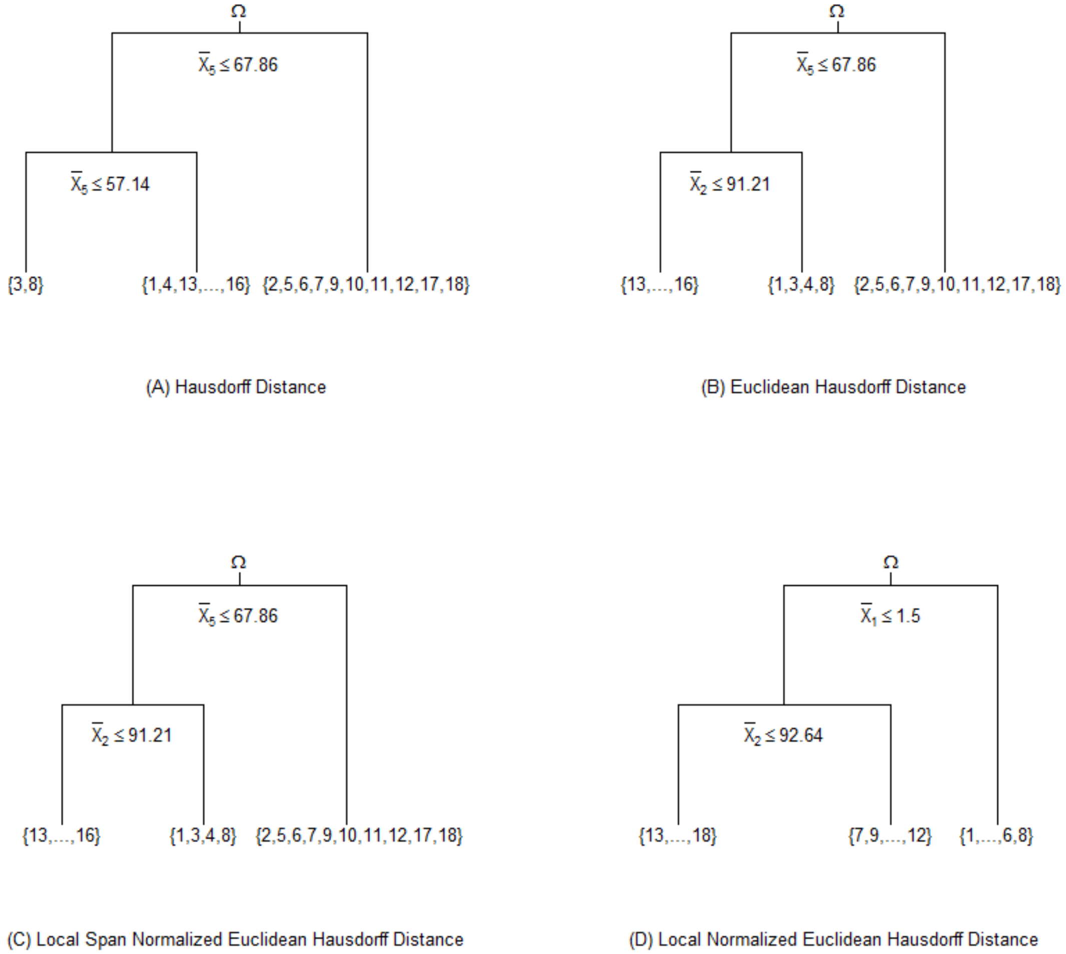
LSNEH and LNEH distances, results are given from an aspect that variables are ”equally weighted”. If the researcher’s interest is not particularly in the variable elevation, the LSNEH and LNEH distances can be considered.

The Chicken Data

Let us consider the chicken data from Section 3.3 again. The values from the original classical data set are aggregated into 6 intervals for each diet group. In the original data set, the row number of each chicken is also provided. For each diet, there are 6 rows with 8 chickens in each row. Those observations belonging to the same row and diet group are aggregated into an interval. And the interval’s lower bound is the 5^{th} percentile and upper bound is the 95^{th} percentile of the values. Table 4.12 is the interval-valued data set extracted from the original data. Table 4.13 lists the interval means for the chicken data [(interval lower limit + interval upper limit)/2] and Table 4.14 lists the interval ranges for the chicken data (interval upper limit - interval lower limit).

Figure 4.12 is the clustering result of using four different Hausdorff distances. As mentioned in Section 4.2, since the GSNEH and GNEH distances are not recommended most of the time, they are no longer used here for the chicken data.

From Table 4.12, observations 1 to 6 eat diet 1, observations 7 to 12 eat diet 2 and observations 13 to 18 eat diet 3. Potentially, the eighteen observations will be clustered into three groups with each group representing one diet. From Figure 4.12, it is obvious that only the LNEH distance is able to allocate the observations to their underlying clusters with an accuracy $= 17/18 = 0.94$. All other distances perform poorly in this example. Since X_2 and X_5 have larger scales compared to other variables, the Hausdorff distances uses X_5 for clustering and the EH and LSNEH distances use X_2 and X_5 for clustering all the time. As discussed in Section 4.2, this can be questionable when the clustering process is dominated by variables with large scales, which is why the standardized distances might be better here.



(A) Hausdorff Distance

(B) Euclidean Hausdorff Distance

(C) Local Span Normalized Euclidean Hausdorff Distance

(D) Local Normalized Euclidean Hausdorff Distance

Figure 4.12: Clustering Results for Chicken Data

Table 4.12: Chicken Data

Observation	Row	Diet	X_1	X_2	X_3	X_4	X_5
1	1	1	[1.22, 1.85]	[81.17, 118.69]	[51.38, 64.64]	[2, 7]	[28.57, 100]
2	5	1	[1.16, 1.97]	[80.89, 117.87]	[50.95, 65.07]	[4, 7]	[57.14, 100]
3	9	1	[1.16, 1.96]	[73.21, 118.03]	[49.03, 67.36]	[0, 7]	[0.00, 100]
4	16	1	[1.24, 1.90]	[87.84, 118.66]	[52.93, 69.12]	[2, 7]	[28.57, 100]
5	20	1	[1.18, 2.11]	[76.74, 120.19]	[48.73, 65.10]	[3, 7]	[42.86, 100]
6	24	1	[1.20, 2.00]	[80.17, 115.00]	[51.26, 69.44]	[4, 7]	[57.14, 100]
7	2	2	[1.16, 1.77]	[79.50, 116.87]	[50.84, 62.30]	[4, 7]	[57.14, 100]
8	6	2	[1.23, 1.90]	[68.11, 125.64]	[50.04, 63.64]	[0, 7]	[0.00, 100]
9	10	2	[1.14, 1.66]	[71.41, 114.99]	[48.29, 64.27]	[4, 7]	[42.86, 100]
10	15	2	[1.16, 1.62]	[73.26, 116.91]	[48.56, 63.80]	[4, 7]	[57.14, 100]
11	19	2	[1.17, 1.62]	[73.31, 114.83]	[48.56, 62.12]	[4, 7]	[57.14, 100]
12	23	2	[1.22, 1.64]	[77.94, 117.57]	[52.14, 62.56]	[5, 7]	[71.43, 100]
13	3	3	[1.04, 1.51]	[54.89, 111.39]	[48.00, 57.93]	[2, 7]	[28.57, 100]
14	7	3	[1.08, 1.61]	[62.61, 110.97]	[45.80, 57.70]	[2, 7]	[28.57, 100]
15	11	3	[0.97, 1.49]	[54.97, 106.36]	[47.00, 56.80]	[2, 7]	[28.57, 100]
16	14	3	[1.04, 1.60]	[57.81, 115.11]	[46.72, 62.45]	[2, 7]	[28.57, 100]
17	18	3	[1.12, 1.62]	[62.87, 114.31]	[45.96, 57.92]	[3, 7]	[42.86, 100]
18	22	3	[1.10, 1.65]	[65.27, 118.89]	[47.35, 58.26]	[3, 7]	[42.86, 100]

The result from using the LNEH distance is consistent with the result from the ANOVA in Chapter 3. Chickens fed with the same diet are more similar to each other while chickens fed with different diets are more dissimilar to each other. By observing Table 4.13 and Table 4.14, we know that observations $\{1, ..., 6, 8\}$ are grouped together since overall they have the largest layer body weights (X_1) , the highest average daily feed intakes (X_2) and the largest egg weights (X_3). Meanwhile, they have the biggest variation in layer body weight (X_1), which means their body weights increase most from week 18 to 75; they have lower variations in average daily feed intake (X_2) compared to observations {13,...,18}, which means their daily feed intakes increase more slowly from week 18 to 76; they have the largest variations in egg weight (X_3), which means their egg weights increase most from week 19 to 76. Except for observation 8, these may benefit from the diet they take, diet 1. Observation 8 is a special

Table 4.13: Chicken Data Interval Averages

Observation	Row	Diet	X_1	X_2	X_3	X_4	X_5
1	1	1	1.53	99.93	58.01	4.50	64.29
2	5	1	1.57	99.38	58.01	5.50	78.57
3	9	1	1.56	95.62	58.19	3.50	50.00
4	16	1	1.57	103.25	61.02	4.50	64.29
5	20	1	1.64	98.47	56.92	5.00	71.43
6	24	1	1.60	97.59	60.35	5.50	78.57
7	2	2	1.47	98.19	56.57	5.50	78.57
8	6	2	1.56	96.88	56.84	3.50	50.00
9	10	2	1.40	93.20	56.28	5.50	71.43
10	15	2	1.39	95.09	56.18	5.50	78.57
11	19	2	1.39	94.07	55.34	5.50	78.57
12	23	2	1.43	97.76	57.35	6.00	85.71
13	3	3	1.28	83.14	52.96	4.50	64.29
14	7	3	1.35	86.79	51.75	4.50	64.29
15	11	3	1.23	80.66	51.90	4.50	64.29
16	14	3	1.32	86.46	54.59	4.50	64.29
17	18	3	1.37	88.59	51.94	5.00	71.43
18	22	3	1.37	92.08	52.81	5.00	71.43

case here since it should not be clustered with $\{1, ..., 6\}$, and the producer may want to pay more attention to this observation for further interests.

Observations $\{7, 9, ..., 12\}$ are grouped together because they have smaller layer body weights (X_1), lower average daily feed intakes (X_2) and smaller egg weights (X_3) compared to observations $\{1, ..., 6, 8\}$. And, they have lower variation in layer body weight (X_1) compared to observations $\{1, ..., 6, 8\}$, which means their body weights increase less from week 18 to 75; they have lower variations in average daily feed intake (X_2) compared to observations {13,...,18}, which means their daily feed intakes increase more slowly from week 18 to 76; they have lower variations in egg weight (X_3) compared to observations $\{1, ..., 6, 8\}$, which means their egg weights increase more slowly from week 20 to 76. These may be due to the

Table 4.14: Chicken Data Interval Ranges

Observation	Row	Diet	X_1	X_2	X_3	X_4	X_5
1	1	1	0.62	37.51	13.26	5.00	71.43
2	5	1	0.80	36.99	14.12	3.00	42.86
3	9	1	0.80	44.81	18.33	7.00	100.00
4	16	1	0.65	30.81	16.18	5.00	71.43
5	20	1	0.94	43.45	16.37	4.00	57.14
6	24	1	0.80	34.83	18.19	3.00	42.86
7	2	2	0.61	37.37	11.46	3.00	42.86
8	6	2	0.67	57.53	13.60	7.00	100.00
9	10	2	0.51	43.57	15.98	3.00	57.14
10	15	2	0.46	43.66	15.24	3.00	42.86
11	19	2	0.46	41.51	13.56	3.00	42.86
12	23	2	0.42	39.63	10.42	2.00	28.57
13	3	3	0.47	56.50	9.93	5.00	71.43
14	7	3	0.54	48.36	11.90	5.00	71.43
15	11	3	0.52	51.39	9.80	5.00	71.43
16	14	3	0.56	57.30	15.73	5.00	71.43
17	18	3	0.49	51.44	11.96	4.00	57.14
18	22	3	0.56	53.61	10.91	4.00	57.14

diet they take, diet 2.

Observations $\{13, ..., 18\}$ are grouped together because they have the smallest layer body weights (X_1), the lowest average daily feed intakes (X_2) and the smallest egg weights (X_3). In addition, they have lower variation in layer body weight (X_1) compared to observations $\{1, ..., 6, 8\}$, which means their body weights increase less from week 18 to 75; they have the largest variations in average daily feed intake (X_2), which means their daily feed intakes increase more than the other diet groups from week 18 to 76; they have the smallest variations in egg weight (X_3), which means their egg weights increase most slowly from week 20 to 76. These may due to the diet they take, diet 3.

As to egg production (X_5), observations with diet 2 ($\{7, 9, ..., 12\}$) have the smallest variations in egg production while egg production averages of observations $\{7, 9, ..., 12\}$ are not smaller than the other two groups. This means the egg production of chickens with diet 2 reaches a higher value quickly at an early stage (week 21 to 31) but increases slowly as time goes by. This can be checked by taking means of egg productions for each diet at each week. Table 4.15 shows the averages of egg productions by diet and week.

4.4 Discussion

Notice that when Local Normalized distances are introduced, we only mention that the normalization factors $|\mathcal{Y}_j|$ and H_j need to be recalculated according to which cluster they are in, but do not show how they are recalculated. In this section, let us look at how the LNEH distance is calculated. The way to deal with the LSNEH distance is the same.

Recall that the normalization factor for LNEH distance in Equation (4.4) is

$$H_j^2 = \frac{1}{2n^2} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} [d_j(\mathbf{X}(i_1), \mathbf{X}(i_2))]^2, \quad j = 1, ..., p.$$

Here, n is the size of the cluster which H_j belongs to.

If the clustering process is working on the u^{th} cluster at the r^{th} stage, i.e., C_u^r, and the size of the cluster is n_{ru}. Then,

$$\Delta_{jq}^{ru} \triangleq I(C_u^r) - I(C_u^{r+1}) - I(C_{u+1}^{r+1})$$

from Step 2 in Section 4.1 will be calculated for all the $j = 1, ..., p$ and $q = 1, ..., n_{ru} - 1$ so that a maximum Δ_{jq}^{ru} can be found. And the corresponding variable j and cut-point q will be used for splitting C_u^r into C_u^{r+1} and C_{u+1}^{r+1}. Hence, for each combination of j and q, there are three within-cluster variations need to be calculated. They are $I(C_u^r)$, $I(C_u^{r+1})$ and $I(C_{u+1}^{r+1})$. As introduced in Section 2.4, in order to get $I(C_u^r)$, $I(C_u^{r+1})$ and $I(C_{u+1}^{r+1})$, the LNEH distances between each pair of observations within the cluster C_u^r, C_u^{r+1} and C_{u+1}^{r+1} need to be calculated. Therefore, H_j needs to be determined when calculating LNEH distances between observations. Here comes the options:

Option 1.

The normalization factor H_j uses n_{ru}, the size of the cluster C_u^r, all the time and hence is invariant in all the three clusters C_u^r, C_u^{r+1} and C_{u+1}^{r+1}.

Option 2.

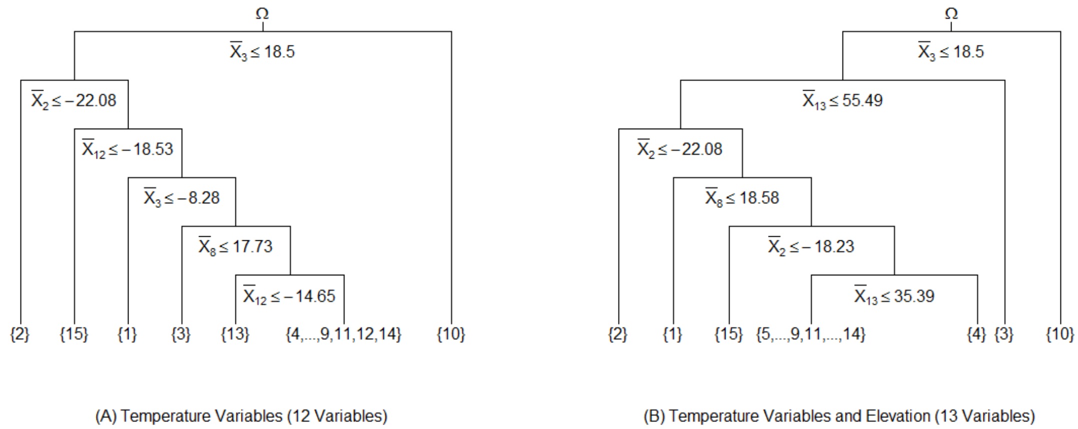
The normalization factor H_j uses n_{ru}, $n_{r+1,u}$ and $n_{r+1,u+1}$, in the clusters C_u^r, C_u^{r+1} and C_{u+1}^{r+1}, respectively and hence varies.

To sum up, the question is, when the clustering process is splitting a cluster into two sub-clusters and the LNEH distance is used, shall we use the same H_j from the parent cluster (Option 1) for both the parent cluster and the two successive clusters, or shall we use the three different H_j's from the parent cluster and the two successive clusters (Option 2) when calculating distances?

All the results using the LNEH distance in this dissertation have been using Option 1 so far. However, if Option 2 is selected, the results will be quite different. Let us consider the China temperature data from Section 4.3 again. Figure 4.13 shows the clustering result of using the LNEH distance for both twelve and thirteen variables by choosing Option 2.

Results from using the Local Normalized Euclidean Hausdorff distance in Figure 4.13 are peculiarly unique. At each stage, only one station was picked up and formed a new cluster. This is due to the invariant *total within-cluster variation*, which is defined in Equation (2.32).

If observations are evenly weighted, and the Local Normalized Euclidean Hausdorff dis-


Ω
$\overline{X}_3 \leq 18.5$
$\overline{X}_2 \leq -22.08$
$\overline{X}_{12} \leq -18.53$
$\overline{X}_3 \leq -8.28$
$\overline{X}_8 \leq 17.73$
$\overline{X}_{12} \leq -14.65$
{2} {15} {1} {3} {13} {4,...,9,11,12,14} {10}
Ω
$\overline{X}_3 \leq 18.5$
$\overline{X}_{13} \leq 55.49$
$\overline{X}_2 \leq -22.08$
$\overline{X}_8 \leq 18.58$
$\overline{X}_2 \leq -18.23$
$\overline{X}_{13} \leq 35.39$
{2} {1} {15} {5,...,9,11,...,14} {4} {3} {10}

(A) Temperature Variables (12 Variables)

(B) Temperature Variables and Elevation (13 Variables)

Figure 4.13: China Temperature Clustering Results Using Option 2

tance is used, it can be shown that

$$I(C_u^r) = \begin{cases} \frac{n_{ru}}{n}p, & \text{if } n_{ru} > 1, \\ 0, & \text{if } n_{ru} = 1, \end{cases} \tag{4.19}$$

where $I(C_u^r)$ is defined in Equation (2.31), n_{ru} is the number of observations in the u^{th} cluster at the r^{th} stage, n is the number of all observations and p is the number of variables being used. (The detailed proof is in Appendix 4.5.)

For example, without loss of generality, let us have a look at the first stage of clustering. At the first stage, we only have one cluster, C_1^1, with $n_{11} = n > 1$ observations. If we divide it into two clusters, C_1^2 and C_2^2, with n_{21} and n_{22} observations respectively, by Equations (2.32) and (4.19), we will have

$$W(P_1) = I(C_1^1) = \frac{n_{11}}{n}p = p;$$

and

$$W(P_2) = I(C_1^2) + I(C_2^2) = \begin{cases} \frac{n_{21}}{n}p + \frac{n_{22}}{n}p & = p, \qquad \text{if } n_{21} > 1 \text{ and } n_{22} > 1, \\ 0 + \frac{n_{22}}{n}p & = \frac{n-1}{n}p, \qquad \text{if } n_{21} = 1, \\ \frac{n_{21}}{n}p + 0 & = \frac{n-1}{n}p, \qquad \text{if } n_{22} = 1. \end{cases}$$

Therefore,

$$\triangle = W(P_1) - W(P_2) = \begin{cases} 0, & \text{if } n_{21} > 1 \text{ and } n_{22} > 1, \\ \frac{p}{n}, & \text{if } n_{21} = 1 \text{ or } n_{22} = 1. \end{cases}$$

It is obvious that that the cut-point, q, will always be $q = 1$ or $q = n-1$ so that $\triangle$ (discussed in Section 2.4) can be maximized as p/n. In other words, C_1^1 will always be divided into two groups with one of them having just one observation and the other having the rest of observations, as long as observations are evenly weighted and the LNEH distance is used. This clustering process cannot reveal the real underlying clusters.

Therefore, Option 2 should be avoided when the LNEH distance is used. This also applies to the LSNEH distance.

4.5 Appendix

Proof of Equation (4.19)

The proof is based on classical data. The proof for interval data can be achieved analogously.

Suppose we have n classical observations and p variables being used for clustering. Let X_j denote the j^{th} variable and $\mathbf{X}(i)$ denote the i^{th} observation, where $j = 1, ..., p$ and $i = 1, ..., n$. The realization of the i^{th} observation with the j^{th} variable is denoted as x_{ij}. By Equation

(2.30), we know

$$I(C_u^r) = \frac{1}{2\lambda}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}} w_{i_1}w_{i_2}d^2(i_1,i_2) \tag{4.20}$$

where n_{ru} is the number of observations in the u^{th} cluster from the r^{th} stage, $d^2(i_1,i_2)$ is a distance or dissimilarity measure between the observations $\mathbf{X}(i_1)^{ru}$ and $\mathbf{X}(i_2)^{ru}$ in C_u^r, $i_1, i_2 = 1, ..., n_{ru}$, w_i is the weight associated with the observation $\mathbf{X}(i)^{ru}$ and $\lambda = \sum_{i=1}^{n_{ru}} w_i$. If the Local Normalized Euclidean Hausdorff distance is used, then

$$d^2(i_1,i_2) = \sum_{j=1}^{p}\left[\frac{d_j(i_1,i_2)}{H_j}\right]^2 \tag{4.21}$$

where $d_j(i_1,i_2) = |x_{i_1j} - x_{i_2j}|$ and H_j is the standard deviation of X_j.

When $n_{ru} > 1$, we substitute Equation (4.21) into Equation (4.20), to obtain

$$\begin{aligned}
I(C_u^r) &= \frac{n}{2n_{ru}}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}} w_{i_1}w_{i_2}\sum_{j=1}^{p}\frac{(x_{i_1j}-x_{i_2j})^2}{H_j} \\
&= \frac{n}{2n_{ru}}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}} w_{i_1}w_{i_2}\sum_{j=1}^{p}\frac{(x_{i_1j}-x_{i_2j})^2}{\sum_{i=1}^{n_{ru}}\frac{(x_{ij}-\bar{x}_{\cdot j})^2}{n_{ru}}} \\
&= \frac{n}{2}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}} w_{i_1}w_{i_2}\sum_{j=1}^{p}\frac{(x_{i_1j}-x_{i_2j})^2}{\sum_{i=1}^{n_{ru}}(x_{ij}-\bar{x}_{\cdot j})^2} \\
&= \frac{n}{2}\sum_{j=1}^{p}\frac{1}{\sum_{i=1}^{n_{ru}}(x_{ij}-\bar{x}_{\cdot j})^2}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}} w_{i_1}w_{i_2}(x_{i_1j}-\bar{x}_{\cdot j}+\bar{x}_{\cdot j}-x_{i_2j})^2 \\
&= \frac{n}{2}\sum_{j=1}^{p}\frac{1}{\sum_{i=1}^{n_{ru}}(x_{ij}-\bar{x}_{\cdot j})^2}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}} w_{i_1}w_{i_2}[(x_{i_1j}-\bar{x}_{\cdot j})^2+ \\
&\quad (\bar{x}_{\cdot j}-x_{i_2j})^2 + 2(x_{i_1j}-\bar{x}_{\cdot j})(\bar{x}_{\cdot j}-x_{i_2j})].
\end{aligned}$$

Consider the situation where weights are even, i.e., $w_{i1} = w_{i2} = 1/n$. Hence,

$$\begin{aligned} I(C_u^r) &= \frac{1}{2n}\sum_{j=1}^{p}\frac{1}{\sum_{i=1}^{n_{ru}}(x_{ij}-\bar{x}_{\cdot j})^2}\sum_{i_1=1}^{n_{ru}}\sum_{i_2=1}^{n_{ru}}[(x_{i_1j}-\bar{x}_{\cdot j})^2+(\bar{x}_{\cdot j}-x_{i_2j})^2+ \\ &\quad 2(x_{i_1j}-\bar{x}_{\cdot j})(\bar{x}_{\cdot j}-x_{i_2j})] \\ &= \frac{1}{2n}\sum_{j=1}^{p}\frac{1}{\sum_{i=1}^{n_{ru}}(x_{ij}-\bar{x}_{\cdot j})^2}[n_{ru}\sum_{i_1=1}^{n_{ru}}(x_{i_1j}-\bar{x}_{\cdot j})^2+n_{ru}\sum_{i_2=1}^{n_{ru}}(x_{i_2j}-\bar{x}_{\cdot j})^2+0] \\ &= \frac{1}{2n}\sum_{j=1}^{p}\frac{2n_{ru}\sum_{i_1=1}^{n_{ru}}(x_{i_1j}-\bar{x}_{\cdot j})^2}{\sum_{i=1}^{n_{ru}}(x_{ij}-\bar{x}_{\cdot j})^2} \\ &= \frac{n_{ru}}{n}p. \end{aligned}$$

When $n_{ru} = 1$, it is clear that $I(C_u^r) = 0$.

■

Table 4.15: Egg Production Averages by Diet and Week

Week	Diet 1	Diet 2	Diet 3
19	1.8	1.2	3.6
20	27.1	25.9	28.6
21	60.1	71.1	65.5
22	83.9	87.8	88.7
23	91.1	93.8	90.2
24	92.6	95.7	92.3
25	97.3	97.0	90.8
26	92.0	96.0	88.1
27	92.0	95.1	87.2
28	89.9	92.7	87.5
29	91.7	94.2	82.1
30	92.3	94.2	81.8
31	95.2	96.4	80.7
32	96.4	93.0	74.1
33	94.0	94.2	74.1
34	94.9	91.8	71.1
35	95.5	89.7	65.8
36	92.0	88.1	69.0
37	93.2	84.2	64.0
38	96.4	91.5	70.8
39	93.8	90.0	69.0
40	93.5	90.0	67.6
41	96.4	90.0	72.9
42	97.0	88.8	72.6
43	94.3	86.9	72.3
44	94.3	88.8	71.1
45	94.6	90.0	69.9
46	94.3	90.6	69.9
47	94.9	88.8	64.9
48	94.6	91.2	68.5
49	94.8	91.8	68.5
50	97.0	88.4	67.0
51	96.0	86.6	62.8
52	94.8	90.0	72.0
53	89.7	88.8	73.5
54	93.6	87.5	64.6
55	96.0	91.6	72.6
56	94.8	87.3	75.9
57	94.5	88.8	80.7
58	92.5	89.4	78.0
59	93.5	93.0	77.4
60	91.5	91.8	77.7
61	95.7	90.7	77.1
62	93.0	88.5	75.3
63	92.9	91.4	80.1
64	88.8	91.9	82.1
65	91.9	88.8	78.6
66	92.9	93.9	76.9
67	91.0	90.0	72.6
68	92.2	90.9	68.1
69	91.6	90.0	69.0
70	90.6	88.4	69.9
71	91.2	89.7	66.6
72	90.9	88.8	69.8
73	90.9	87.8	61.1
74	87.8	90.3	69.7
75	89.8	87.0	67.3
76	88.8	88.5	66.7

R Code for Divisive Monothetic Clustering Method

```
HDMCM.Interval <- function(distance='H',no.c=4,index,weight,data)

  # Hierarchical Divisive Monothetic Clustering Method for Interval-valued Data

  # Distance Input: 'H' (Hausdorff), 'EH' (Euclidean Hausdorff),

  #                 'GSNEH' (Global Span Normalized EH),

  #                 'GNEH' (Global Normalized EH),

  #                 'SNEH' (Local Span Normalized EH),

  #                 'NEH' (Local Normalized EH)

  # Number of Clusters Wanted: no.c = integer

  # Indices for Observations: index = integer vector

  # Weights for Observations: weight = number vector

  # Interval-Valued Data: data = data.frame/matrix

{

  Hdistance <- function(...) # Calculate Hausdorff distance matrix

    # Input should be an interval variable (a,b)

  {

    var = as.matrix(...)

    n = length(var[,1])

    d = matrix(0,n,n)

    for (i in 1:n)

    {

      for (j in 1:n)

      {

        d[i,j] = max(abs(var[i,1]-var[j,1]),

                     abs(var[i,2]-var[j,2]))

      }

    }
    return(d)
  }

  EHdistance <- function(...)
    # Calculate Euclidean Hausdorff distance matrix
    # Input should be interval data (Can be multiple variables)
  {
    var = as.matrix(...)
    p = length(var[1,])/2
    n = length(var[,1])
    d = matrix(0,n,n)
    for (j in 1:p)
    {
      dj = matrix(0,n,n)
      for (a in 1:n)
      {
        for (b in 1:n)
        {
          dj[a,b] = max(abs(var[a,2*j-1]-var[b,2*j-1]),
                        abs(var[a,2*j]-var[b,2*j]))
        }
      }
      d=d+(dj)^2
    }
    d=sqrt(d)
    return(d)
  }
```

```
NEHdistance <- function(...)
  # Calculate Normalized Euclidean Hausdorff dist matrix
  # Input should be interval data (Can be multiple variables)
{
  var = as.matrix(...)
  p = length(var[1,])/2
  n = length(var[,1])
  d = matrix(0,n,n)
  for (j in 1:p)
  {
    dj = matrix(0,n,n)
    for (a in 1:n)
    {
      for (b in 1:n)
      {
        dj[a,b] = max(abs(var[a,2*j-1]-var[b,2*j-1]),
                      abs(var[a,2*j]-var[b,2*j]))
      }
    }
    Hj = sqrt((1/2/n/n)*sum(dj*dj))
    if (Hj==0)
    {
      d = d + 0
    }
    else
    {
      d=d+(dj/Hj)^2
    }
  }
  d=sqrt(d)
  return(d)
}

SNEHdistance <- function(...)
  # Calculate Span Normalized Euclidean Hausdorff dist matrix
{
  var = as.matrix(...)
  p = length(var[1,])/2
  n = length(var[,1])
  d = matrix(0,n,n)
  for (j in 1:p)
  {
    dj = matrix(0,n,n)
    for (a in 1:n)
    {
      for (b in 1:n)
      {
        dj[a,b] = max(abs(var[a,2*j-1]-var[b,2*j-1]),
                           abs(var[a,2*j]-var[b,2*j]))
      }
    }
    yj = max(var[,2*j]) - min(var[,2*j-1])
    if (yj==0)
    {
      d = d + 0
    }
    else
    {
      d=d+(dj/yj)^2
    }
  }
  d=sqrt(d)
  return(d)
}

Distance <- function(which.dist,X,j=0)
  # Get the distance matrix of X
  # (by variable j if using Hausdorff distance)
{
  if (which.dist=='H')
  {dist = as.matrix(Hdistance(X[,c(2*j-1,2*j)]))}
  if (which.dist=='EH')
  {dist = as.matrix(EHdistance(X))}
  if (which.dist=='NEH')
  {dist = as.matrix(NEHdistance(X))}
  if (which.dist=='SNEH')
  {dist = as.matrix(SNEHdistance(X))}
  return(dist)
}

P = length(data[1,])/2 # Number of variables
N = length(data[,1]) # Number of observations
D.matrix.array = array(dim=c(N,N,P)) # Distance matrices
  # for all observations by variable 1 to variable p
```

```
if (distance %in% c('H','EH'))
{
  for (i in 1:P)
  {
    D.matrix.array[,,i] = Distance(distance,data,j=i)
  }
}
if (distance=='GNEH') # Global Normalized Euclidean Hausdorff Distance
{
  for (i in 1:P)
  {
    D.matrix.array[,,i] = Distance('NEH',data)
  }
}
if (distance=='GSNEH')
  # Global Normalized Span Normalized Euclidean Hausdorff Distance
{
  for (i in 1:P)
  {
    D.matrix.array[,,i] = Distance('SNEH',data)
  }
}



WSS <- function (w, D)
  # Calculate within cluster variation using distance matrix D
  # Input should be weight and distance matrix
{
  D = as.matrix(D)
  n = length(D[,1])
  W1 = matrix(w, nrow=n, ncol=n, byrow = F) # creat weight matrix
  W2 = matrix(w, nrow=n, ncol=n, byrow = T)
  lamda = sum(w)
  wss = sum(D*D*W1*W2)/2/lamda # within cluster variation
  return(wss)
}


Delta1 <- function (I, w, X, j)
  # For one cluster, return the maximum Delta_q by variable j,
  # cut value and Hausdorff dist between two observations around the cut point
  # Input should be index, weight, interval variables
  # and which variable to be used
{
  X_bar = (X[,2*j-1]+X[,2*j])/2 # Mean of each observation by jth variable
  s_X = cbind(I,w,X_bar,rank(X_bar,ties.method=’first’))
    # statistics matrix: get rank according to mean
  colnames(s_X)[length(s_X[1,])] = "rank"
  n = max(s_X[,"rank"]) # Number of observations within the cluster
  delta = rep(0,times=n) # Vector containing delta

  if (distance %in% c(’H’,’EH’,’GNEH’,’GSNEH’))
    # Get within cluster variation before clustering
  {
    D = as.matrix(D.matrix.array[I,I,j])
      # Distance matrix with row and column number in I
    wss = WSS(w,D)
  }
  if (distance %in% c('NEH','SNEH'))
    # Have to recalculate distance matrix each step
  {
    D = Distance(distance,X)
    wss = WSS(w,D)
    D = cbind(D,s_X[,c('I','w','rank')])
    colnames(D)[1:n] = I
  }

  for (i in 1:(n-1)) # Get within cluster variation after clastering
  {
    if (distance %in% c('H','EH','GNEH','GSNEH'))
    {
      I1 = s_X[s_X[,"rank"]<=i,'I']
      w1 = s_X[s_X[,"rank"]<=i,'w']
      D1 = as.matrix(D.matrix.array[I1,I1,j])
      wss1 = WSS(w1,D1)
      I2 = s_X[s_X[,"rank"]>i,'I']
      w2 = s_X[s_X[,"rank"]>i,'w']
      D2 = as.matrix(D.matrix.array[I2,I2,j])
      wss2 = WSS(w2,D2)
    }
    if (distance %in% c('NEH','SNEH'))
    {
      I1 = s_X[s_X[,"rank"]<=i,'I']
      w1 = s_X[s_X[,"rank"]<=i,'w']
      D1 = as.matrix(D[D[,’I’] %in% I1,colnames(D) %in% I1])
      wss1 = WSS(w1,D1)
      I2 = s_X[s_X[,"rank"]>i,’I’]
      w2 = s_X[s_X[,"rank"]>i,’w’]
      D2 = as.matrix(D[D[,’I’] %in% I2,colnames(D) %in% I2])
      wss2 = WSS(w2,D2)
    }
   delta[i] = wss - wss1 - wss2
  }
  delta[n] = -Inf
  if (sum(delta^2)==0)
    # Situation when observations with variable j are the same
  {
    cut_value = -1/0
    m_delta=0
    Hdist=0
  }
  else
  {
    delta = cbind(seq(1:n),delta)
    colnames(delta) = c("rank","delta")
    XX = merge(s_X,delta,by="rank") # Matrix containing rank and delta
    m_delta = max(XX[,"delta"])
    q = XX[XX[,"delta"]==m_delta,"rank"]
      # Get the cut point q, according to max delta
    if (length(q)>1)
      # If more then one cut point, choose the one with surrounded
      # observations having largest Hausdorff dist
    {
     Hdist=rep(0,times=length(q))
     for (k in 1:length(q))
     {
      Hdist[k]=Hdistance(X[s_X[,"rank"]==q[k]
                 | s_X[,"rank"]==q[k]+1,(2*j-1):(2*j)])[1,2]
     }
     q = q[which(Hdist==max(Hdist),arr.ind=T)][1]
    }
    cut_value = (XX[XX[,"rank"]==q,"X_bar"] + XX[XX[,"rank"]==q+1,"X_bar"])/2
    Hdist = Hdistance(X[s_X[,"rank"]==q|s_X[,"rank"]==q+1,(2*j-1):(2*j)])[1,2]
     # Hausdiff distance between two observations around cut point
   }
   return(list(cut_value=cut_value, max_delta=m_delta, Hdist=Hdist))
  }

  Delta2 <- function(index,w,X)
   # For one cluster, return the maximum Delta_q over all
   # variables, cut value, variable used to cut and grouping.
   # And return Hausforff dist b/w two observations around cut point.
   # Input should be indices of observations, weight and
   # multiple interval variables
  {
   p = length(X[1,])/2 # Number of variables
   delta = rep(0,times=p) # Contain biggest delta using each variable
   c = rep(0,times=p) # Contain cut value using each variable
   dist = rep(0,times=p) # Contain distance between two observations around
    # cut point using each variable
```

```
for (j in 1:p)
{
  delta1 = Delta1(index,w,X,j)
  delta[j] = delta1$max_delta
  c[j] = delta1$cut_value
  dist[j] = delta1$Hdist
}
which.var = which(delta==max(delta),arr.ind=T)
 # Decide which variable is with max delta.
 # If there exists tie, pick up the one with larger Hausdorff distance
 # (as below)
which.var.one = which.var # which.var.one will contain just one variable
max.dist = dist[which.var]
if (length(which.var)>1)
{
  temp = cbind(delta,c,dist)
  max.dist = max(temp[temp[,"delta"]==max(temp[,"delta"]),"dist"])
   # Calculate maximum Hausdorff distance between two points around
   # the cut point from those with maximum delta using different
   #variables
  which.var.one =
    which(temp[,"delta"]==max(temp[,"delta"])&temp[,"dist"]==max.dist)[1]
   # Decide which variable with maximum delta and maximum distance.
   # If tie exists even after these two conditions, pick up the 1st var
}
cut = c[which.var.one]  # Decide the cut value
mean.which.var = (X[,(2*which.var.one-1)]+X[,(2*which.var.one)])/2
 # Mean of the variable used
```

```
  I = cbind(index,mean.which.var)
  cluster1 = I[which(mean.which.var<=cut,arr.ind=T),’index’]
   # Index of observations distributed to the 1st cluster
  cluster2 = I[which(mean.which.var>cut,arr.ind=T),’index’]
   # Index of observations distributed to the 2nd cluster
  return(list(variable=which.var.one, max.delta=max(delta),
              cut.value=cut, max.Hdist=max.dist, cluster1=sort(cluster1),
              cluster2=sort(cluster2)))
}

Chavent <- function(r,index,w,X) # Final function.
  # Input should be r=number of clusters wanted,
  # indices and weights for observations.
{
  n = length(X[,1])
  cluster = rep(1,times=n) # Cluster index
  XX <- cbind(index,w,cluster,X)
  for (i in 1:(r-1)) # Do clustering at the ith stage
  {
    delta = rep(0,times=i) # Contains maxiumum deltas from every cluster
    dist = rep(0,times=i)
     # Contain distances between two observations around cut point
     # from every cluster
    for (j in 1:i) # do clustering to all i clusters
    {
      if (length(XX[XX[,’cluster’]==j,1])==1)
        # Clusters with one obs’n in it will have delta=0
      {
```

```
    delta[j]=0
  }
  else
  {
    xx = XX[XX[,’cluster’]==j,]
    ind = xx[,’index’]
    weight = xx[,’w’]
    x = xx[,-(1:3)]
    delta2=Delta2(index=ind,w=weight,X=x)
    delta[j]=delta2$max.delta
     # Get maxiumum delta using different variables of the jth cluster
    dist[j]=delta2$max.Hdist
     # Get maxiumum Hausdorff dist. b/w two obs’ns around
     # the cut point of the jth cluster, using different variables
  }
}
which.cluster = which(delta==max(delta),arr.ind=T)
 # Decide which cluster with maximum delta.
 # If there exists tie, pick up the one with larger Hausforff dist.
which.cluster.one = which.cluster
if (length(which.cluster)>1)
{
  temp = cbind(delta,dist)
  max.dist = max(temp[temp[,"delta"]==max(temp[,"delta"]),"dist"])
   # Maximum distance from those with max delta
  which.cluster.one = which(temp[,"delta"]==max(temp[,"delta"])
              &temp[,"dist"]==max.dist)[1]
   # Which cluster with max delta and maximum distance.
```

```
     # If tie exists even after these two conditions, pick up the 1st
     # cluster.
   }
   Xi = XX[XX[,'cluster']==which.cluster.one,] # data used at ith stage
   result = Delta2(index=Xi[,'index'],w=Xi[,'w'],X=Xi[,-(1:3)])
   XX[is.element(XX[,'index'], result$cluster2),'cluster']=i+1
    # Divide observations into two clusters
   names(i) = '################ At the ith stage: #################'
   names(result) = c('Variable Used To Cluster',
            'The Maximum Delta','Cut Value',
     'Hausdorff Distance Between Two Observations Around The Cut Value',
            'The Smaller Cluster','The Larger Cluster')
   print(c(i,result))
  }
  cluster.result.1 = rbind(XX[,'index'],XX[,'cluster'])
  rownames(cluster.result.1)=c('Index','Cluster')
  cluster.result.2 =
   rbind(XX[,'index'],XX[,'cluster'])[,sort.list(XX[,'cluster'])]
  rownames(cluster.result.2)=c('Index','Cluster')
  list(ClusterResult1=cluster.result.1,ClusterResult2=cluster.result.2,
     variable=result[1])
 }
 Chavent(r=no.c, index=index, w=weight, X=data)
}
```

R Code for Simulation Studies

```
Data.Sim <- function(sim.n.cluster, cluster.size, points.in.obs,
                    sigma, mean,xlim=c(0,30),ylim=c(0,30),plot="F")
  # Function to simulate one interval-valued data set
  # Number of Clusters Want to Simulate: sim.n.cluster = integer
  # Cluster Size of Each Cluster: cluster.size = integer vector
  # (order from big to small numbers)
  # Number of Points to Construct An Interval: points.in.obs = integer
  # Covariance Matrix: sigma = numeric vector
  # Mean vector: mean = numeric vector
  # Parameters for Plotting: xlim, ylim = integer vector
  # Whether or Not Want the Plotting: plot = "T" or "F"
{
  Sigma <- matrix(ncol=2,nrow=2*sim.n.cluster)
  Mean <- matrix(ncol=2,nrow=sim.n.cluster)
  for (j in 1:sim.n.cluster)
  {
    Sigma[((j-1)*2+1):((j-1)*2+2),]=matrix(sigma[((j-1)*4+1):((j-1)*4+4)],2,2)
    Mean[j,] = mean[((j-1)*2+1):((j-1)*2+2)]
  }
  data = matrix(nrow=(sum(cluster.size)*points.in.obs),ncol=2)
  for (j in 1:sim.n.cluster)
  {
    data[(sum(cluster.size[0:(j-1)])*points.in.obs+1):
           (sum(cluster.size[0:j])*points.in.obs),] =
      mvrnorm(n=cluster.size[j]*points.in.obs, mu=Mean[j,],
              Sigma=Sigma[((j-1)*2+1):((j-1)*2+2),])
  }
  if (plot=="T")
  {
    plot(data[1:(cluster.size[1]*points.in.obs),],
         xlim=xlim,ylim=ylim,pch=20,col=’blue’,
         xlab=expression(paste(’X’[’1’])),ylab=expression(paste(’X’[’2’])))
    lines(ellipse(Sigma[1:2,],centre=Mean[1,]),lty=1,lwd=2)
    points(data[(cluster.size[1]*points.in.obs+1):
                  (sum(cluster.size[1:2])*points.in.obs),],pch=0,col=’red’)
    lines(ellipse(Sigma[3:4,],centre=Mean[2,]),lty=2,lwd=2)
    if (sim.n.cluster==3)
    {
      points(data[(sum(cluster.size[1:2])*points.in.obs+1):
                    (sum(cluster.size[1:3])*points.in.obs),],pch=1,col=’green’)
      lines(ellipse(Sigma[5:6,],centre=Mean[3,]),lty=3,lwd=2)
    }
  }
  int.data = matrix(nrow=sum(cluster.size),ncol=4)
  for (i in 1:sum(cluster.size))
  {
    int.data[i,1] = quantile(data[(points.in.obs*(i-1)+1):
                                    (points.in.obs*i),1],probs=0.05)
    int.data[i,2] = quantile(data[(points.in.obs*(i-1)+1):
                                    (points.in.obs*i),1],probs=0.95)
    int.data[i,3] = quantile(data[(points.in.obs*(i-1)+1):
                                    (points.in.obs*i),2],probs=0.05)
    int.data[i,4] = quantile(data[(points.in.obs*(i-1)+1):
                                    (points.in.obs*i),2],probs=0.95)
  }
  return(int.data)
}
```

```
Data.Sim.Multiple <- function(sim.n.cluster, cluster.size,
                              points.in.obs, sigma, mean)
  # Function to simulate 5 interval-valued data sets for each replication
  # Number of Clusters Want to Simulate: sim.n.cluster = integer
  # Cluster Size of Each Cluster: cluster.size = integer matrix
  # Number of Points to Construct An Interval: points.in.obs = matrix
  # Covariance Matrix: sigma = numeric vector
  # Mean vector: mean = numeric vector
  # Parameters for Plotting: xlim, ylim = integer vector
  # Whether or Not Want the Plotting: plot = "T" or "F"
{
  Sigma <- matrix(ncol=2,nrow=2*sim.n.cluster)
  Mean <- matrix(ncol=2,nrow=sim.n.cluster)
  for (j in 1:sim.n.cluster)
  {
    Sigma[((j-1)*2+1):((j-1)*2+2),] =
      matrix(sigma[((j-1)*4+1):((j-1)*4+4)],2,2)
    Mean[j,] = mean[((j-1)*2+1):((j-1)*2+2)]
  }

  data = matrix(nrow=(sum(cluster.size[1,])*points.in.obs[1]),ncol=2)
  for (j in 1:sim.n.cluster)
  {
    data[(sum(cluster.size[1,0:(j-1)])*points.in.obs[1]+1):
           (sum(cluster.size[1,0:j])*points.in.obs[1]),] =
      mvrnorm(n=cluster.size[1,j]*points.in.obs[1],
              mu=Mean[j,], Sigma=Sigma[((j-1)*2+1):((j-1)*2+2),])
  }
  for (i in 1:length(cluster.size[,1])) # Creat matrices for interval data
  {
    assign(paste("x",i,sep=""),matrix(nrow=sum(cluster.size[i,]), ncol=4))
  }
  for (i in 1:length(cluster.size[,1]))
  {
    temp = matrix(nrow=sum(cluster.size[i,]),ncol=4)
    for (j in 1:sum(cluster.size[i,]))
    {
      temp[j,1] = quantile(data[(points.in.obs[i]*(j-1)+1):
                                (points.in.obs[i]*j),1],probs=0.05)
      temp[j,2] = quantile(data[(points.in.obs[i]*(j-1)+1):
                                (points.in.obs[i]*j),1],probs=0.95)
      temp[j,3] = quantile(data[(points.in.obs[i]*(j-1)+1):
                                (points.in.obs[i]*j),2],probs=0.05)
      temp[j,4] = quantile(data[(points.in.obs[i]*(j-1)+1):
                                (points.in.obs[i]*j),2],probs=0.95)
    }
    assign(paste("x",i,sep=""), temp)
  }
  return(list(x1,x2,x3,x4,x5))
}


Accuracy <- function(c.result,cluster.size)
  # Function to calculate accuray; currently only works for 2 clusters
{
  table1 = table(c.result[2,1:cluster.size[1]])
```

```
table2 = table(c.result[2,(cluster.size[1]+1):sum(cluster.size[1:2])])
table = matrix(data=0,nrow=n.cluster,ncol=n.cluster)
# within each true cluster, how the clustered observations are distributed
rownames(table) = c("Underlying Cluster 1", "Underlying Cluster 2")
colnames(table) = c("# of clusters labeled as 1",
                    "# of clusters labeled as 2")
table[1,1] = table1["1"];  table[1,2] = table1["2"]
table[2,1] = table2["1"];  table[2,2] = table2["2"]
table[is.na(table)] = 0
percent.table = matrix(nrow=n.cluster,ncol=n.cluster)
for (i in 1:n.cluster)
{
  for (j in 1:n.cluster)
  {percent.table[i,j] = table[i,j]/sum(table[i,])}
}
index = matrix(nrow=n.cluster,ncol=2)
# find location of number of correctly clustered observations.
# According to which number, 1 or 2, dominates the percentage of that cluster
colnames(index)=c("row","col")
index[1,] = which(percent.table==max(percent.table), arr.ind=TRUE)[1,]
# which location has the biggest percent.
# If tie, use first (since cluster size bigger)
index[2,] = index[1,]+1
index[index[,]>2] = index[index[,]>2]%%2
correct.clustered = table[index[1,1],index[1,2]] +
  table[index[2,1],index[2,2]]
acc = correct.clustered/sum(table)
return(acc)
```

```
}


Sim <- function(n.ite, D.v = c('H','EH','GSNEH','GNEH','SNEH','NEH'),
                cluster.size=rbind(c(10,10),c(20,20),c(50,50),c(100,100)),
                points.in.obs=c(1000,500,200,100), accuracy="T",report.var="F")
  # Number of itterations wanted: n.ite = ingeter
  # Whether or not to report the number of times each variable is used:
  # report.var = "T" or "F"
  {
  if (accuracy=="T")
  {
    Acc = array(0,dim=c(n.ite,length(D.v),length(cluster.size[,1])))
    colnames(Acc) = D.v
  }
  if (report.var=="T")
  {
    Var = array(0,dim=c(n.ite,length(D.v),length(cluster.size[,1])))
    colnames(Var) = D.v
  }
  time = array(0,dim=c(n.ite,length(D.v),length(cluster.size[,1])))
  colnames(time) = D.v
  for (i in 1:n.ite)
  {
    sink("NUL")
    X <- Data.Sim.Multiple(sim.n.cluster, cluster.size, points.in.obs,
                           sigma, mean)
    for (j in 1:length(D.v))
    {
```

```
    for (k in 1:length(cluster.size[,1]))
    {

        time[i,j,k] =
          system.time(c.result <-
            HDMCM.Interval(distance=D.v[j],no.c=n.cluster,
                index=seq(1:sum(cluster.size[k,])),
                weight=rep(1/sum(cluster.size[k,]),sum(cluster.size[k,])),
                data=X[[k]]))[3]
      if (accuracy=="T")
      {
        Acc[i,j,k] <- Accuracy(c.result$ClusterResult1,
                               cluster.size=cluster.size[k,])
      }
      if (report.var=="T")
      {
        variable <- c.result[[3]]
        Var[i,j,k] <- as.numeric(variable)
      }
    }
  }
  sink()
  if (i%%100==0)
  {
    print(Sys.time())
    print(c(’Now at iteration:’,i))
  }
}
```

```
if (accuracy=="T")
{
  Accuracy.Mean = round(apply(Acc,c(3,2),mean),digits=3)
  colnames(Accuracy.Mean) = D.v
  rownames(Accuracy.Mean) = rowSums(cluster.size)
  Accuracy.SD = round(apply(Acc,c(3,2),sd),digits=4)
  colnames(Accuracy.SD) = D.v
  rownames(Accuracy.SD) = rowSums(cluster.size)
}
if (report.var=="T")
{
  Variable1.Cut = apply(-Var+2,c(3,2),sum)
  colnames(Variable1.Cut) = D.v
  rownames(Variable1.Cut) = rowSums(cluster.size)
  Variable2.Cut = apply(Var-1,c(3,2),sum)
  colnames(Variable2.Cut) = D.v
  rownames(Variable2.Cut) = rowSums(cluster.size)
}
Time.Mean = round(apply(time,c(3,2),mean),digits=4)
colnames(Time.Mean) = D.v
rownames(Time.Mean) = rowSums(cluster.size)
Time.SD = round(apply(time,c(3,2),sd),digits=5)
colnames(Time.SD) = D.v
rownames(Time.SD) = rowSums(cluster.size)
result = list(Accuracy.Mean=Accuracy.Mean, Accuracy.SD=Accuracy.SD,
              Time.Mean=Time.Mean, Time.SD=Time.SD,
              Variable1.Cut=Variable1.Cut, Variable2.Cut=Variable2.Cut)
write.table(result, file="SimulationResult.txt",append=TRUE)
```

```
  return(result)
}

################# Case 1: An Intuitive Example ####################

sigma = c(10,0,0,10,10,0,0,10)
mean=c(10,15,20,15)
sim.n.cluster=2
n.cluster = 2
cluster.size = c(10,10)
points.in.obs = 50

x = Data.Sim(sim.n.cluster, cluster.size, points.in.obs, sigma, mean, plot='T')

Sim(n.ite=1000, D.v = c('H','EH','GSNEH','GNEH','SNEH','NEH'),
    cluster.size=rbind(c(2,2),c(5,5),c(10,10),c(20,20),c(50,50)),
    points.in.obs=c(500,200,100,50,20), accuracy="T",report.var="T")

################# Case2: A bad example for Hausdorff #################

sigma = c(0.1,0,0,30,0.1,0,0,0.1)
mean=c(14,15,16,15)
sim.n.cluster=2
n.cluster = 3
cluster.size = c(10,10)
points.in.obs = 50

x = Data.Sim(sim.n.cluster, cluster.size, points.in.obs, sigma, mean, plot='T')
```

```
Sim(n.ite=1000, D.v = c(’H’,’EH’,’GSNEH’,’GNEH’,’SNEH’,’NEH’),
    cluster.size=rbind(c(2,2),c(5,5),c(10,10),c(20,20),c(50,50)),
    points.in.obs=c(500,200,100,50,20), accuracy="T",report.var="T")

################# Case3: A bad e.g. for Hausdorff/EH ##################

sigma = c(0.1,0,0,30,0.1,0,0,30)
mean=c(14,15,16,15)
sim.n.cluster=2
n.cluster = 2
cluster.size = rbind(c(10,10))
points.in.obs = c(50)

x = Data.Sim(sim.n.cluster, cluster.size, points.in.obs, sigma, mean, plot=’T’)

Sim(n.ite=1000, D.v = c(’H’,’EH’,’GSNEH’,’GNEH’,’SNEH’,’NEH’),
    cluster.size=rbind(c(2,2),c(5,5),c(10,10),c(20,20),c(50,50)),
    points.in.obs=c(500,200,100,50,20), accuracy="T",report.var="T")

################# Case4: Dispersion ################

sigma = c(1,0,0,1,1,0,0,1,1,0,0,1)
mean=c(2,5,7,5,50,5)
sim.n.cluster = 3
n.cluster = 3
cluster.size = c(10,10,10)
points.in.obs = 100
```

```
x = Data.Sim(sim.n.cluster, cluster.size, points.in.obs, sigma, mean,
             plot='T',xlim=c(0,55),ylim=c(0,10))


HDMCM.Interval(distance='H',no.c=n.cluster, index=seq(1:sum(cluster.size)),
              weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='EH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
              weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='GSNEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
              weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='GNEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
              weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='SNEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
              weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='NEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
              weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)


################ Case5: Outlier ###################


sigma = c(2,0,0,2,2,0,0,2,0.1,0,0,0.1)
mean=c(12,5,18,5,10,30)
sim.n.cluster = 3
n.cluster = 3
cluster.size = c(10,10,1)
points.in.obs = 100


x = Data.Sim(sim.n.cluster, cluster.size, points.in.obs, sigma, mean, plot='T')
```

```
HDMCM.Interval(distance='H',no.c=n.cluster, index=seq(1:sum(cluster.size)),
               weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='EH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
               weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='GSNEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
               weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='GNEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
               weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='SNEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
               weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)
HDMCM.Interval(distance='NEH',no.c=n.cluster, index=seq(1:sum(cluster.size)),
               weight=rep(1/sum(cluster.size),sum(cluster.size)), data=x)

################# Case6: One cluster within the other ###############

sigma = c(1,0,0,1,5,0,0,30)
mean=c(15,15,15,15)
sim.n.cluster = 2
n.cluster = 2
cluster.size = c(10,10)
points.in.obs = 50

x = Data.Sim(sim.n.cluster, cluster.size, points.in.obs, sigma, mean, plot='T')

Sim(n.ite=1000, D.v = c('H','EH','GSNEH','GNEH','SNEH','NEH'),
    cluster.size=rbind(c(2,2),c(5,5),c(10,10),c(20,20),c(50,50)),
    points.in.obs=c(500,200,100,50,20), accuracy="T",report.var="T")
```

References

Billard, L. and Diday, E. (2006). *Symbolic Data Analysis: Conceptual Statistics and Data Mining.* John Wiley & Sons, Chichester.

Chavent, M. (1998). A Monothetic Clustering Method. *Pattern Recognition Letters*, 19, 989-996.

Chavent, M. (2000). Criterion-Based Divisive Clustering for Symbolic Data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data.* (Eds. Bock, H.-H. and Diday, E.). Springer-Verlag, Berlin, 299-311.

Gowda, K.C. and Diday, E. (1991). Symbolic Clustering Using a New Dissimilarity Measure. *Pattern Recognition*, 24, 567-578.

Gowda, K.C. and Diday, E. (1992). Symbolic Clustering Using a New Similarity Measure. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 368-378.

Hausdorff, F. (1937). *Set Theory* (translated into English by Aumann, J. R. 1957). Chelsey, New York.

Ichino, M. and Yaguchi, H. (1994). Generalized Minkowski Metrics for Mixed Feature Type Data Analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 24, 698-708.

Chapter 5

Future Work

In Table 3.3 from Section 3.2, we consider mean square of treatments over mean square of error (MST/MSE) as a statistic following the F distribution. However, while this result is intuitively correct, the proof has not been formally shown in this dissertation and remains as future research by using the definition of ϵ_{ij} in Section 3.4. Also, results from Section 3.3 only tell whether the treatments are significant overall. In order to tell what pairs of treatments are different, other methods, e.g., the method of multiple comparisons, need to be developed for interval-valued data. Finally, the results from a one-way ANOVA can be extended to two-way and multi-way ANOVA.

Note that the chicken data used in Section 3.3 can be reorganized as a one-way layout table with 3 diet groups with repeated measures over 64 weeks. Methods for analyzing data with repeated measures such as using RM-ANOVA and linear mixed-effects models should be more appropriate for these data. Methods for analyzing symbolic data with repeated measures need to be developed in the future.

In this dissertation, we have shown and extended a divisive clustering method for interval data proposed by Chavent (1998, 2000) and compared six different Hausdorff distances, but the problem of when to stop the clustering process is still unsolved. Kim (2009, 2011) extended the Dunn (1974) index and the Davis and Bouldin (1979) index to histogram data. The Dunn index and the Davis-Bouldin index can also be extended to interval data. A pseudo F statistic developed by Calinski and Harabasz (1974) can also be considered to be extended to interval data to determine the optimal number of clusters.

Notice that the divisive clustering method utilizes one variable each time for separating the cluster and can continue until there is only one observation in any cluster. Usually, the variables it uses are a subset of all the variables of the data set. Therefore, the clustering method can also perform variable selection for interval-valued data. The reliability of the variables selected by the clustering method remains as future research.

Since interval data are easy to be aggregated from large data sets and the computing time for analyzing them is fast, it is always beneficial to start with interval data. If satisfactory results can be achieved by performing analysis on interval data, it may not be necessary to use more sophisticated symbolic data. However, interval data will often lose some information by just taking the lower and upper bounds of the data. Currently, we assume that the spread within each interval follows a uniform distribution. If the data have more complicated structures and the assumption can not hold, the results of using interval data can be poor. In these situations, more sophisticated symbolic data should be considered. Aggregating data into histograms is an example where the uniform assumption is broken. Performing clustering analysis on histogram data by using Hausdorff distances is for the future work.

References

Calinski, R.B. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3, 1-27.

Chavent, M. (1998). A Monothetic Clustering Method. *Pattern Recognition Letters*, 19, 989-996.

Chavent, M. (2000). Criterion-Based Divisive Clustering for Symbolic Data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data.* (Eds. Bock, H.-H. and Diday, E.). Springer-Verlag, Berlin, 299-311.

Davis, D.L. and Bouldin, D.W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1, 224-227.

Dunn, J.C. (1974). Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetica*, 4, 95-104.

Kim, J. (2009). *Dissimilarity Measures for Histogram-Valued Data and Divisive Clustering of Symbolic Objects.* Doctoral Dissertation, University of Georgia.

Kim, J. and Billard, L. (2011). A polythetic clustering process and cluster validity indexes for histogram-valued objects. *Computational Statistics and Data Analysis*, 55, 2250-2262.