Classification and Regression Trees for Symbolic Data

by

Wanxue Zou

(Under the Direction of Lynne Billard)

Abstract

Compared to classical data which take a single value, there is another type of data, symbolic data, which can be a list, an interval, and even a distribution into consideration. Symbolic data are very common in our daily life; however, the analysis methods for symbolic data are very limited. For instance, a famous and useful method for supervised learning such as regression or classification is the decision tree. There are many useful algorithms based on the decision tree. However, the decision tree is only useful to classic data taking a single value, either numerical or categorical. In this dissertation, we will extend the classification and regression tree method (CART) to symbolic data.

Index words: Symbolic data, Interval-valued data, Modal multi-valued data, Decision Tree, Classification and Regression Tree (CART)

Classification and Regression Trees for Symbolic Data

by

Wanxue Zou

B.S., University of Science and Technology of China, 2016

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

2021

Classification and Regression Trees for Symbolic Data

by

Wanxue Zou

Major Professor: Lynne Billard

Committee: Abhyuday Mandal
Pengsheng Ji
Liang Liu

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and Dean of the Graduate School
The University of Georgia
August 2021

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my advisor, Dr. Lynne Billard, for her patient guidance, tremendous help, and cares through my research. Without the support of Dr. Billard, I would not be such lucky to devote myself to the research topics in which I am interested. No matter how busy, she is always willing to provide guidance and advice to students. She not only guides my academic research but also encourages and supports me in my life.

I would also like to thank my committee members, Dr. Abhyuday Mandal, Dr. Pengsheng Ji, and Dr. Liang Liu, who offered me priceless advice and challenged me to think more about my research. I would also like to thank the faculty, staff, and friends in the Department of Statistics for their help, knowledge, and guidance on my career path during my program.

Finally, I have big thanks to my parents for their unconditional love and support for all my decisions. I would also like to thank my parent-in-law for their understanding and encouragement. Lastly, I wish to especially thank the most important person in my life, my husband, Hanquan Su, who always trusts and encourages me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For supervised learning including regression or classification methodologies, a famous and useful method is a decision tree. There are a lot of useful algorithms based on the decision tree such as XGBoost proposed by T. Chen and Guestrin, 2016, and LightGBM introduced by Ke et al., 2017. Those methods have been widely used in distributed computing in industry. The tree model has excellent interpretability and is relatively accurate for classification prediction problems, which makes it attain prominence in data science competitions and real-life data analyses. However, classification and regression tree methods are only useful for classical data taking a single value, either numerical or categorical. The focus of this research is to propose an approach to tree methods for symbolic data, specifically for modal multi-valued data and interval-valued data.

Symbolic data were first introduced by Diday, 1987. Most of the data we collect and analyze are classical data, which can be analyzed by standard classical analysis methods. However, when the sample size is very large, for instance, 100 million, some traditional methods for analyzing such data sets will be problematic. One method to deal with the large data set is by aggregating the individuals into classes or categories so that the aggregated data set is greatly reduced in its sample size. In this case, the random variable does not take values at a single point anymore, no matter if it is quantitative or qualitative. In addition,

there are many situations where the raw data cannot be collected. For example, if we want to investigate the employees of companies, they often are reluctant to provide the information of individual employees because of privacy issues. In this case, to analyze the company's employees, we can consider a certain type of employee rather than separate individuals. Then independent variables and responses will take values in the form of an interval, a list, or even a distribution. What is more, symbolic data can also be used for time-series data to unified the dimension. For instance, if we are considering the classification of the charging process for shared bicycles, a traditional classification model is not applicable because the data matrix dimensions are inconsistent due to the fact that the charging time is not fixed. In these cases, converting the data into symbolic data to make the data dimension consistent with each charging process is a good choice to handle the analysis of the data. Billard and Diday, 2006, also provides many examples of natural and aggregated symbolic data that need to be analyzed. Hence, it is of vital importance for us to find methods for symbolic data analysis.

There are many existing works for symbolic data analysis, for instance, properties of the sample mean and variance, principal component analysis (PCA), and clustering methods. As for supervised learning, the simple linear regression for interval-valued data is first published by Billard and Diday, 2000. In addition, Billard and Diday, 2002, introduced linear regression for the other types of symbolic data such as histogram-valued data. Xu, 2010, proposed another approach for linear regression for interval-valued data. Due to the reason that symbolic data are very common in our daily life but the methods to analyze symbolic data are very limited, the analysis methods that contain symbolic variables need to be explored, especially for the classification and regression parts. Classification and Regression Tree (CART) is not only widely used for both classification and regression but also it is very easy to explain the tree structure. In addition, the forecasting of CART is very impressive. That is the main reason why it is necessary to explore this tree-based model for symbolic data.

Based on the previous work such as descriptive statistics and clustering methods for symbolic data, and classification and regression tree method (CART) for classical data, we propose a new method to analyze symbolic data called Classification and Regression Tree for Symbolic data (CART for SD).

In Chapter 2, Section 2.1 introduces what are symbolic data and what is the difference between symbolic data and classical data. In addition, some statistical concepts, properties, and basic analyzing methods are also introduced in this section. Section 2.2 describes the commonly used decision tree method, the classification and regression tree method which is originally developed by Breiman et al., 1984. In Chapter 3, we first consider one type of symbolic data, multi-valued data as the explanatory or the response variables in Section 3.1. Section 3.2 and 3.3 introduce the situations where we consider several other symbolic data types as explanatory and response variables. Chapter 4 gives some simulations and real data examples using the algorithms. In Chapter 5, some conclusions and future work are discussed.

Chapter 2

Literature Review

To establish a foundation for the classification and regression tree for symbolic data, we give a brief definition of symbolic data with some examples in Section 2.1. For the classification or regression tree method (CART), we always hope that with as few groups as possible, the higher the similarity in the same group, the better. There are many ways to divide the data set, among which clustering is a good step to pre-process the attributes. Some existing cluster methods for symbolic data are reviewed in Section 2.1.4. Then we summarize the main idea of a decision tree in Section 2.2.

2.1 Symbolic Data

Compared with classical observations which are points either qualitative or quantitative, symbolic data takes values as lists, intervals, histograms, and so on. All of the definitions and notations in this subsection are from Chapter 2 and Chapter 3 of Billard and Diday, 2006. To understand the definitions and properties better, we can use the Bank-marketing data set from https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#. Here, there are $n = 41188$ observations and $p = 20$ explanatory variables. We just choose five variables to obtain a brief understanding of symbolic data. The descriptions of the variables in this Bank-marketing

data set are shown in Table 2.1.

Table 2.1: Description of Bank-marketing data set

No.	Variable	Type	Description
V1	Age	numerical	(in years): ≥ 0
V2	Job	categorical	Type: administration, blue-collar(BC), entrepreneur, housemaid, management, retired, self-employed(SE), services, student, technician, unemployed, unknown
V3	Marital	categorical	Type: married, divorced, single, unknown
V4	Duration	numerical	Last contact duration, in seconds
V5	Housing	categorical	Housing loan (Yes, No, unknown)

We just choose ten of the $n = 4188$ observations to obtain a brief understanding of symbolic data, shown in Table 2.2. The explanatory variables "Age" and "Duration" are numerical and the other three explanatory variables are categorical.

Table 2.2: Part of the Bank-marketing data set (Classical)

No.	Age	Job	Marital	Duration	Housing
1	56	housemaid	married	261	no
2	57	services	married	149	no
3	37	services	married	226	no
4	40	administration	married	151	no
5	56	services	married	307	yes
6	45	services	married	198	no
7	59	administration	married	139	no
8	41	blue-collar	married	217	no
9	24	technician	single	380	no
10	25	services	single	50	no

Since there are only $n = 41188$ observations and $p = 20$ explanatory variables, these data can be analyzed using classical techniques at this size. However, if the sample size is very large, such as 100 million, some traditional methods do not work when analyzing such data sets. Therefore, reducing the sample size is critical.

There is another situation that due to privacy restrictions, the bank cannot provide the specific value of each individual for some special variables, such as the number of deposits. If we divide clients into several groups and only record the variable values of the groups, we can solve this problem well while retaining key information. Another important issue that we need to consider is how to reconfigure the data set to a size that allows the analysis to proceed. At first, we need to consider what we want to learn from the data to "reduce" the sample size in a meaningful way. For example, it may be of greater interest to the researcher to determine whether a certain category, such as married management denoted by 'married-management', or a technician who is between 25 and 35 years old denoted by 'technician between 25-35 years old' rather than considering all the individuals. Since each of these categories may consist of more than one individual, the observed value becomes a list or an interval rather than a single point. For example, if we consider the category 'technician between 25-35 years old', the number of contacts performed during this campaign for the whole group will take values in the interval $[1, 21]$ and the marital status will be the list {married, divorced, single}.

2.1.1 Notations and Definitions

Before we study the existing analytical methods of symbolic data, we first need to know some basic notations and definitions.

Notation 2.1.1.1 Let $E = \{\omega_u, u = 1, \ldots, m\}$ be a set of m symbolic **concepts/categories**.

Notation 2.1.1.2 For the random variables $X_j, j = 1, ..., p$, let the notation x_{ij} represent a classical value or realization on the individual $i = 1, ..., n$ and ξ_{ij} be a symbolic realization. Therefore, for a classical realization of the variable X_i, we have $X_j(i) = x_u$, while for a symbolic one, $X_j(i) = \xi_u$.

If the random variables are measured on a category $\omega_u \in E$ rather than an individual, it could be written as $X_j(\omega_u) = \xi_u$. Let $\mathcal{X}_j$ be the domain of X_j and $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_p$ be the domain of $\mathbf{X} = (X_1, ..., X_p)$. We have further definitions.

Definition 2.1.1 Every point $\boldsymbol{x} = (x_1, ..., x_p)$ in $\mathcal{X}$ is called a **description vector**.

Definition 2.1.2 The p-dimensional subspace $\mathbf{D} = (D_1, ..., D_p) \subseteq \mathcal{X}$ is called a **description set**, where $D_j \subseteq \mathcal{X}_j, j = 1, ..., p$. If $\mathbf{D} = \Pi_{j=1}^{p} D_j$ is the Cartesian product of the sets D_j, D is called a **Cartesian description set**.

2.1.2 Several Types of Symbolic Data

Compared with classical data which consist of qualitative and quantitative single point values, symbolic data contain several different types: multi-valued, interval-valued, modal multi-valued, and histogram interval-valued. In this subsection, we will introduce the definitions of these four types with examples. The examples are still drawn from the Bank-marketing data set.

Definition 2.1.3 A **multi-valued** symbolic random variable X is one whose possible value takes one or more values from the list of values in its domain $\mathcal{X}$.

Example 2.1.1 Suppose we are interested in observations of different ages and we want to know the difference in "Job" (X_1) between categories, the domain={administration, unknown, unemployed,..., (list of jobs), technician, services}. Then for the group of 22-year-old:

$$X_1(\omega_1) = X_1(\text{22-year-old}) = \xi_{11} = \{\text{administrate}, \text{student}, \text{technician}\};$$

for the group of 75-year-old:

$$X_1(\omega_2) = X_1(\text{75-year-old}) = \xi_{12} = \{\text{housemaid}, \text{retired}\};$$

and so on.

Another example in life is the multiple-choice questionnaire. There are multiple options in one question. We can select all the options that apply. Therefore, the value for each problem is a list that contains unfixed elements.

Definition 2.1.4 An **interval** symbolic random variable takes values in an interval, which can be closed or open at either end.

After aggregating the individuals, some continuous values emerge as an interval. The two sides of the interval can be calculated by

$$a_{uj} = \min_{i \in \Omega_u} x_u, \quad b_{uj} = \max_{i \in \Omega_u} x_u$$

where Ω_u is the set of $i \in \Omega$ values which make up the category ω_u.

Example 2.1.2 Suppose we are interested in observations of different ages and we want to know the difference in "Duration"(X_2) between categories. The domain is $\{x \geq 0\}$. Then, for the group of 20-year-old:

$$X_2(\omega_3) = X_2(\text{20-year-old}) = \xi_{23} = [172, 274];$$

for the group of 80-year-old:

$$X_2(\omega_4) = X_2(\text{80-year-old}) = [123, 712];$$

and so on.

In addition, a common question when applying for a job is, what is your expected salary. In this case, we should provide a range of values rather than a single value. As a result,

the realization will be an interval. Here, we have a special case that if we only have one observation a in a group or all the observations take just one numerical value a, the interval value becomes $[a, a]$. Based on the assumption that the values within intervals are uniformly distributed over the interval $[a, b]$, we have

$$P(z \leq \xi) = \begin{cases} 0, & \xi < a, \\ \dfrac{\xi - a}{b - a}, & a \leq \xi \leq b, \\ 1, & \xi \geq b. \end{cases} \tag{2.1}$$

Definition 2.1.5 Let $\mathcal{X} = \{\eta_1, \eta_2, ...\}$ be the domain of a multi-valued random variable X. A **modal multi-valued** variable is one whose observed outcomes take values that are a subset of $\mathcal{X}$ with a nonnegative measure.

That is, for the category ω_u, the realization takes the form

$$X(\omega_u) = \{\eta_{u1}, p_{u1}; ...; \eta_{us_u}, p_{us_u}\}$$

where $\eta_1, \eta_2, ... \subseteq \mathcal{X}$ and where the outcome η_{uk} occurs with weight p_{uk},$k = 1, ..., s_u$, and with $\sum_{k=1}^{s_u} p_{uk} = 1$.

Example 2.1.3 Suppose we are interested in observations of different ages and we want to know the difference in "Marital" (X_3) between categories, the domain is {married, divorced, single}. Then, for the group of 21-year-old:

$$X_3(\omega_5) = X_3(\text{21-year-old}) = \xi_{35} = \{\text{married}, 0.22;\ \text{divorced}, 0.02;\ \text{single}, 0.76\};$$

for the group of 62-year-old:

$$X_3(\omega_6) = X_3(\text{62-year-old}) = \xi_{36} = \{\text{married}, 0.67;\ \text{divorced}, 0.33;\ \text{single}, 0.00\};$$

and so on.

If we consider a multi-valued type here rather than the modal multi-valued, the value of Marital for both ω_5 and ω_6 is {married, divorced, single}, which makes it difficult for us to find the difference of the variable between groups. Adding weights to the values makes it better to distinguish different groups and to capture more information than the multi-valued type does. There are plenty of examples in life. For instance, suppose we want to study the daily behavior of one animal. If we only use a multi-valued variable to represent the daily behavior of the animal, it is difficult for us to distinguish animals of different species through living habits.

Definition 2.1.6 Let X be a quantitative random variable that can take values on a finite number of non-overlapping intervals $\{[a_k, b_k),\ k = 1, 2, ...\}$ with $a_k \leq b_k$. Then, a **histogram** random variable takes the form

$$X(\omega_u) = \{[a_{uk}, b_{uk}), p_{uk}; k = 1, ..., s_u\}$$

where s_u is the finite number of intervals forming the support for the outcome $X(\omega_u)$ for observation ω_u with weight p_{uk}, $k = 1, ..., s_u$, and $\sum_{k=1}^{s_u} p_{uk} = 1$.

Example 2.1.4 Suppose we are interested in observations of different ages and we want to know the difference in "Campaign" (X_4) between categories, the domain is $\{0 \leq x \leq 50\}$. Then, for the group of 35-year-old:

$$X_4(\omega_7) = X_4(\text{35-year-old}) = \xi_{47} = \{[0, 2], 0.62;\ (2, 4], 0.25;\ (4, 6], 0.07;\ (6, 50], 0.06\};$$

for the group of 75-year-old:

$$X_4(\omega_8) = X_4(\text{75-year-old}) = \xi_{47} = \{[0, 2], 0.83;\ (2, 4], 0.00;\ (4, 6], 0.00;\ (6, 50], 0.17\};$$

and so on.

Take the daily rainfall in a city as an example. Because there are multiple regions in one city, the rainfall situation in different regions is not the same. If we only use the average or median of the rainfall in all regions to represent the rainfall situation in the entire city, we will lose part of the information. Therefore, using histogram-valued type to record data will make the recorded data more complete. There are more examples in Billard and Diday, 2006.

2.1.3 Descriptive Statistics For Symbolic Data

After giving the basic definition of different types of symbolic data, some descriptive statistics, such as sample means, sample variance, covariance, and histograms are described by Billard and Diday, 2006.

Multi-valued Random Variable

The sample mean and variance for multi-valued random variable is defined in Bertrand and Goupil, 2000.

Definition 2.1.7 Let X be a multi-valued random variable taking values in $\mathcal{X} = \{X_1, \ldots, X_s\}$, and let $X_u = \{X_{uk}, p_{uk}; k = 1, \ldots, s\}, u = 1, \ldots, n$, be a random sample of size n. Then, the **sample mean** for **modal multi-valued** observations is given by:

$$\bar{X} = \{X_k, \bar{p}_k; k = 1, \ldots, s\}, \quad \bar{p}_k = \frac{1}{n} \sum_{u=1}^{n} p_{uk}. \tag{2.2}$$

Definition 2.1.8 Let X be a multi-valued random variable taking values in $\mathcal{X} = \{X_1, \ldots, X_s\}$, and $X_u = \{X_{uk}, p_{uk}; k = 1, \ldots, s\}, u = 1, \ldots, n$, be a random sample of size n. Then, the

sample variance for **modal multi-valued** observations is given by:

$$S^2 = \left\{ X_k, S_k^2; k = 1, \ldots, s \right\}, \quad S_k^2 = \frac{1}{n-1} \sum_{u=1}^{n} \left(p_{uk} - \bar{p}_k \right)^2 \tag{2.3}$$

where $\bar{p}_k$ is given by Equation 2.2.

Multi-valued attributes can be transferred to a similar form with modal multi-valued variable, that is, $\{x_k, p_k; k = 1, \ldots, s\}$ with $p_k = 1/n_k$ if x_k occurs in the sample, $p_k = 0$ otherwise, where n_k is the number of values from the domain $\mathcal{X}$ which occur in the observation. Thus, the sample mean and variance for multi-valued observations also can be calculated by Equation 2.2 and Equation 2.3.

Interval-valued Random Variable

The sample mean and variance for interval-valued random variable is first introduced by Bertrand and Goupil, 2000.

Definition 2.1.9 Let $X_u = [a_u, b_u], u = 1, \ldots, n$, be a random interval sample of size n. Then, the **sample mean** for **interval** observations is given by:

$$\bar{X} = \frac{1}{2n} \sum_{u=1}^{n} \left(a_u + b_u \right). \tag{2.4}$$

Definition 2.1.10 Let $X_u = [a_u, b_u], u = 1, \ldots, n$, be a random interval sample of size n. Then, the **sample variance** for **interval** observations is given by:

$$S^2 = \frac{1}{3n} \sum_{u=1}^{n} \left(a_u^2 + a_u b_u + b_u^2 \right)^2 - \frac{1}{4n^2} \left[\sum_{i=1}^{n} \left(a_u + b_u \right) \right]^2 \tag{2.5}$$

where $\bar{X}$ is given by Equation 2.4.

Billard, 2007, and Billard, 2008, show that the sample variance in Equation 2.5 is a function of the total sum of squares (TSS), which can be divided into two terms: within sum of

squares (WSS) represents the internal variation and between sum of squares (BSS) represents external variation. This means

$$nS^2 = \text{TSS} = \text{BSS} + \text{WSS}, \tag{2.6}$$

where

$$\text{BSS} = \sum_{u=1}^{n} \left[(a_u + b_u)/2 - \bar{X} \right]^2, \tag{2.7}$$

and

$$\text{WSS} = \sum_{u=1}^{n} \left[\left(a_u - \bar{X}_u\right)^2 + \left(a_u - \bar{X}_u\right)\left(b_u - \bar{X}_u\right) + \left(b_u - \bar{X}_u\right)^2 \right] /3, \tag{2.8}$$

where $\bar{X}_u = \frac{1}{2}(a_u + b_u)$ is the sample mean of the observation X_u and $\bar{X}$ is given by Equation 2.4.

Definition 2.1.11 Let $X_u = \{[a_{uk}, b_{uk}), p_{uk}; k = 1, \ldots, s_u\}, u = 1, \ldots, n$, be a random histogram sample of size n. Then, the **sample mean** for **histogram** observations is given by:

$$\bar{X} = \frac{1}{2n} \sum_{u=1}^{n} \sum_{k=1}^{s_u} (a_{uk} + b_{uk})\, p_{uk}. \tag{2.9}$$

Definition 2.1.12 Let $X_u = \{[a_{uk}, b_{uk}), p_{uk}; k = 1, \ldots, s_u\}, u = 1, \ldots, n$, be a random histogram sample of size n. Then, the **sample variance** for **histogram** observations is given by:

$$S^2 = \frac{1}{3n} \sum_{u=1}^{n} \sum_{k=1}^{s_u} \left\{ \left[\left(a_{uk} - \bar{X}\right)^2 + \left(a_{uk} - \bar{X}\right)\left(b_{uk} - \bar{X}\right) + \left(b_{uk} - \bar{X}\right)^2 \right] p_{uk} \right\} \tag{2.10}$$

where $\bar{X}$ is given by Equation 2.9.

In addition, for histogram realizations we have that

$$nS^2 = \text{TSS} = \text{BSS} + \text{WSS}, \tag{2.11}$$

where

$$\mathrm{BSS} = \sum_{u=1}^{n} \sum_{k=1}^{s_u} \left\{ \left[(a_{uk} + b_{uk})/2 - \bar{X} \right]^2 p_{uk} \right\}, \tag{2.12}$$

and

$$\mathrm{WSS} = \frac{1}{3} \sum_{u=1}^{n} \sum_{k=1}^{s_u} \left\{ \left[\left(a_{uk} - \bar{X}_u\right)^2 + \left(a_{uk} - \bar{X}_u\right)\left(b_{uk} - \bar{X}_u\right) + \left(b_{uk} - \bar{X}_u\right)^2 \right] p_{uk} \right\}, \tag{2.13}$$

where $\bar{X}_u = \sum_{k=1}^{s} (a_{uk} + b_{uk}) p_{uk}/2$ is the sample mean of the observation X_u and $\bar{X}$ is defined in Equation 2.9.

2.1.4 Cluster Analysis For Symbolic Data

In this section, some clustering methods for symbolic data are introduced. As the first step of the classical data clustering method, dissimilarity and distance measurement are very basic in cluster analysis. Gowda and Diday, 1991, proposed the Gowda-Diday dissimilarity of interval value data. Ichino and Yaguchi, 1994, proposed another distance of interval value data, called Ichino-Yaguchi distance. They also extended the Minkowski distance to the generalized (weighted) Minkowski distance. de Carvalho, 1994, and de Carvalho, 1998, proposed two extensions of the Ichino-Yaguchi distance. Kim, 2009, extended the Gowda-Diday distance and the Ichino-Yaguchi distance from interval value data to histogram value data. The definitions of Gowda-Diday dissimilarity measure are given below, which is used as our main tools to split the data and evaluate the performance of the models. In addition, there are some extensions of the Hausdorff distance, such as the Euclidean Hausdorff distance and the normalized Euclidean Hausdorff distance. Y. Chen, 2014, compared these Hausdorff distances.

Dissimilarity and Distance Measures

Suppose we have a one-dimensional multi-valued random variable Y taking values in $\mathcal{X} = \{X_1, \ldots, X_s\}$, The observation ω_u can be rewritten in the following form:

$$\xi(\omega_u) = \{X_{uk}, p_{uk}; k = 1, \ldots, s\} \tag{2.14}$$

for $u = 1, \ldots, m$, where p_{uk} is the relative frequency of X_{uk}. Then the multi-valued random variable X has a similar form with that for a modal multi-valued variable. Now, $p_{uk} = 0$ if Y_{uk} does not occur in the observation, $p_{uk} = 1/n_u$ if Y_{uk} occurs where n_u is the number of values from $\mathcal{Y}$ which do occur. In this case, the realization of multi-valued variable is the same with that of modal multi-valued one. As a result, only modal multi-valued typed will be considered.

Definition 2.1.13 gives a simple modal multi-valued distance, introduced by Chavent, 2000.

Definition 2.1.13 For a multi-valued variable of the form of Equation 2.14 or a modal multi-valued data, a **modal multi-valued distance measure** between any two observations ω_{u_1} and ω_{u_2} is $d(\omega_{u_1}, \omega_{u_2})$ which can be calculated by

$$d^2(\omega_{u_1}, \omega_{u_2}) = \sum_{k=1}^{s} \left(\sum_{u=1}^{m} p_{uk} \right)^{-1} (p_{u_1 k} - p_{u_2 k})^2 . \tag{2.15}$$

In addition, Kim and Billard, 2012, extended the measures in Gowda and Diday, 1991, and Gowda and Diday, 1992, to the modal multi-valued random variable shown in Definition 2.1.16. At first step, one should define the union and intersection operators for modal multi-valued data.

Definition 2.1.14 For a multi-valued data of the form of Equation 2.14 or a modal multi-valued data, the realization of the **union** of ω_{u_1} and ω_{u_2}, $\omega_{u_1} \cup \omega_{u_2}$ is described by

$$\xi(\omega_{u_1} \cup \omega_{u_2}) = \{X_k, p_{(u_1 \cup u_2)k}; k_{=}1, \ldots, s\}, \tag{2.16}$$

where $p_{(u_1 \cup u_2)k} = \max(p_{u_1 k}, p_{u_2 k})$.

Definition 2.1.15 For a multi-valued data of the form of Equation 2.14 or a modal multi-valued data, the realization of the **intersection** of ω_{u_1} and ω_{u_2}, $\omega_{u_1} \cap \omega_{u_2}$ is described by

$$\xi(\omega_{u_1} \cap \omega_{u_2}) = \{X_k, p_{(u_1 \cap u_2)k}; k_{=}1, \ldots, s\}, \tag{2.17}$$

where $p_{(u_1 \cap u_2)k} = \min(p_{u_1 k}, p_{u_2 k})$.

After knowing the definition of the union and intersection operators, Kim and Billard, 2012, proposed the extended multi-valued Gowda-Diday dissimilarity.

Definition 2.1.16 For a multi-valued variable of the form of Equation 2.14 or a modal multi-valued data, **the extended multi-valued Gowda-Diday dissimilarity** between any two observations ω_{u_1} and ω_{u_2} is $d(\omega_{u_1}, \omega_{u_2})$ which can be calculated by

$$d^2(\omega_{u_1}, \omega_{u_2}) = d_1(w_{u_1}, w_{u_2}) + d_2(w_{u_1}, w_{u_2}), \tag{2.18}$$

where

$$d_1(w_{u_1}, w_{u_2}) = \sum_{k=1}^{s} |p_{u_1 k} - p_{u_2 k}| \,/\, \sum_{k=1}^{s} p_{(u_1 \cup u_2)k}, \tag{2.19}$$

and

$$d_2(w_{u_1}, w_{u_2}) = \sum_{k=1}^{s} \left(p_{u_1 k} + p_{u_2 k} - 2p_{(u_1 \cap u_2)k}\right) / \sum_{k=1}^{s} \left(p_{u_1 k} + p_{u_2 k}\right), \tag{2.20}$$

with the union and intersection operators as introduced in Definition 2.1.14 and 2.1.15.

What is more, Kim and Billard, 2012, extended the Ichino and Yaguchi, 1994 distances.

Definition 2.1.17 For a multi-valued variable of the form of Equation 2.14 or a modal multi-valued data, **the extended multi-valued Ichino-Yaguchi dissimilarity measure** between any two observations ω_{u_1} and ω_{u_2} is $d(\omega_{u_1}, \omega_{u_2})$ which can be calculated by

$$d(w_{u_1}, w_{u_2}) = \sum_{k=1}^{s} \left[p_{(u_1 \cup u_2)k} - p_{(u_1 \cap u_2)k} + \gamma\left(2p_{(u_1 \cap u_2)k} - p_{u_1 k} - p_{u_2 k}\right)\right], \tag{2.21}$$

with the union and intersection operators as introduced in Definition 2.1.14 and 2.1.15, and $0 \leq \gamma \leq 0.5$ is a pre-specified constant.

As for interval-valued data, only the Gowda-Diday dissimilarity measure will be defined here; we can find the other distance/dissimilarity measures in Billard and Diday, 2006. Let us denote a one-dimensional interval-valued random variable X with the following form:

$$\xi_u = [a_u, b_u], u = 1, \ldots, m. \tag{2.22}$$

Definition 2.1.18 The **Gowda-Diday dissimilarity measure** between two interval-valued observations w_{u_1} and w_{u_2} of the form of Equation 2.22 is given by:

$$D(w_{u_1}, w_{u_2}) = \sum_{k=1}^{3} D_k(w_{u_1}, w_{u_2}) \tag{2.23}$$

with

$$D_1(w_{u_1}, w_{u_2}) = \| b_{u_1} - a_{u_1} | - | b_{u_2} - a_{u_2} \| / k \tag{2.24}$$

where

$$k = |\text{Max}(b_{u_1}, b_{u_2}) - \text{Min}(a_{u_1}, a_{u_2})| \tag{2.25}$$

is the length of the entire distance spanned by w_{u_1} and w_{u_2} , with

$$D_2(w_{u_1}, w_{u_2}) = (|b_{u_1} - a_{u_1}| + |b_{u_2} - a_{u_2}| - 2I) / k \tag{2.26}$$

where

$$I = \text{Max}(\text{Min}(b_{u_1}, b_{u_2}) - \text{Max}(a_{u_1}, a_{u_2}), 0) \tag{2.27}$$

is the length of the intersections of the intervals $[a_{u_1}, b_{u_1}]$ and $[a_{u_2}, b_{u_2}]$ if the intervals overlap, and 0 otherwise with

$$D_3(w_{u_1}, w_{u_2}) = |a_{u_1} - a_{u_2}| / |x| \tag{2.28}$$

where x is the total length covered by the observed values of X, i.e.,

$$|x| = \max_u\{b_u\} - \min_u\{a_u\}. \tag{2.29}$$

As for the histogram type data, Kim, 2009, and Kim and Billard, 2013, not only introduced the extended multi-valued Gowda-Diday dissimilarity and the extended multi-valued Ichino-Yaguchi dissimilarity measure for histogram-valued data, but also proposed a cumulative density function dissimilarity for histogram data.

Clustering Methods For Symbolic Data

After defining the distance between symbolic variables, we can obtain some clustering methods for symbolic data, such as hierarchical-divisive clustering and hierarchical-pyramid clustering. There are many clustering methods for symbolic data. Gowda and Diday, 1991, proposed a symbol clustering method using minimum similarity. In addition, Chavent, 1998, also developed a single partition hierarchical clustering algorithm for symbolic objects. The algorithm assigns the initial sequence to the time interval through the midpoint value of the time interval of each clustering stage. In addition, Chavent, 2000, published a clustering method based on hierarchical methodology. Guru et al., 2004, and Guru and Kiranagi, 2005, proposed clustering method for interval-valued data. Irpino and Verde, 2006, proposed an agglomerative hierarchical clustering of histogram data based on the Ward criterion. Kim and Billard, 2011, developed quality indices for histogram observations in a hierarchical clustering environment based on Dunn, 1974, and Davies and Bouldin, 1979, which can be used to identify the best value of the number of clusters. Brito and Chavent, 2012, proposed a monothetic divisive clustering algorithm for both interval-valued and histogram-valued variables. Kim and Billard, 2018, introduced a method called histogram clustering to process the data of histogram values. In contrast to these techniques based on the hierarchical methodology, El-Sonbaty and Ismail, 1998, applied the concept of fuzziness on a data set

of symbolic objects to their clustering method. What is more, Zhu, 2019, proposed three monothetic divisive clustering algorithms for interval-valued data. We can find more clustering methods in Billard and Diday, 2019.

2.2 Decision Trees

A decision tree is a popular machine learning method, including regression trees and classification trees. The most outstanding advantage of decision trees is that they will be easier for people to understand and explain than other machine learning methods. In addition, many decision tree-based integration algorithms are widely used in big data mining algorithms. For example, Ho, 1995, proposed a bagging method called Random Forest, which constructs a multitude of decision trees at training time to make a stable prediction. Friedman, 2001, and Friedman, 2002, subsequently developed a stage-wise model as an ensemble of weak prediction models. Typically decision trees are based on the idea of gradient boosting proposed by Breiman, 1997. In addition, there are many other publications such as Mason et al., 1999, and Ridgeway, 2007, for gradient boosting, the base of which is a decision tree.

2.2.1 Introduction of Decision Tree

In this Section, an example about the hepatitis data set available from https://archive.ics.uci.edu/ml/datasets/Hepatitis will be used to introduce the definition of the decision tree. The response variable of this hepatitis data set is categorical with two categories, 1 for death and 2 for alive. The descriptions of the explanatory variables in this data set are shown in Table 2.3.

To understand the data better, we choose the first ten observations as shown in Table 2.4. All of the explanatory variables except “Age” are binary.

Table 2.3: Description of hepatitis data set

No.	Variable	Type	Description
V1	Age	numerical	(in years): ≥ 0
V2	Sex	categorical	1=male, 2=female
V3	Steroid	categorical	1=no, 2=yes
V4	Malaise	categorical	1=no, 2=yes
V5	Anorexia	categorical	1=no, 2=yes
V6	Histology	categorical	1=no, 2=yes

Table 2.4: Part of the hepatitis data set

No.	Class	Age	Sex	Steroid	Malaise	Anorexia	Histology
1	2	30	2	1	2	2	1
2	2	50	1	1	2	2	1
3	2	78	1	2	2	2	1
5	2	34	1	2	2	2	1
6	2	34	1	2	2	2	1
7	1	51	1	1	2	1	1
8	2	23	1	2	2	2	1
9	2	39	1	2	2	2	1
10	2	30	1	2	2	2	1

It is of interest to know what kind of hepatitis patients will die or survive. This is a binary classification case, and is used as an example to illustrate how decision trees work. If we just consider all the discrete variables, we can divide the data set into two groups by one feature. For instance, the data set can be divided into two groups based on gender and there will be four groups if we keep dividing two subsets by the steroid situation.

Decision trees contain a root node, several internal nodes, and leaf nodes. For a decision tree, the root node is the set of all samples that construct the tree. According to one of the features, the root node is divided into several child nodes. The child nodes are recursively constructed through other features, thereby generating new branches. Let us now think about when we should stop dividing the data set. In the basic algorithm of the decision tree, several situations cause the set to be undivided. When the node cannot be divided anymore, the node is a leaf node. One case is when all samples of one subset belong to one category

and do not need to be divided (for classification), or the Mean Square Error (MSE) is small enough to be ignored (for regression). Another situation is when the child set is empty or the samples have the same value of that attribute. In addition, the set cannot be divided if all the features have been used to divide the sets. In other words, there is no improvement on the Gini index defined in Definition 2.2.4 or MSE defined in Definition 2.2.6 by splitting the data.

In the process of constructing a decision tree, the most critical step is how to choose the partitioning attribute. Several splitting rules and how those rules help us find a classification tree will be stated in Section 2.2.2. What is the order of explanatory variables we should use? How to measure one splitting result? These two questions can be answered in Section 2.2.2.

2.2.2 Classification Tree

Information entropy is one of the most commonly used indicators for measuring samples and purity, which was first introduced by Breiman et al., 1984.

Definition 2.2.1 Assume the proportion of the k^{th} group in the sample set D is $p_k,\ k = 1, ..., |\mathcal{Y}|$. Then the **information Entropy** of D is defined as:

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k. \tag{2.30}$$

The smaller the value of $\mathrm{Ent}(D)$ is, the higher the purity of D. Suppose we divide the set based on a discrete attribute X with V possible values $\{x_1, \ldots, x_V\}$ and this gives V child nodes $\{D_1, \ldots, D_V\}$. To consider the fact that the sample sizes of those child nodes are different, a weight $|D^v| / |D|$ is added to calculate the new information entropy since the more samples we have, the greater the influence of branch nodes on purity.

Definition 2.2.2 Information Gain (IG) measures how much "information" a feature gives us about the class. The information gain of D is defined as:

$$\mathrm{IG}(D, X) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}\left(D^v\right). \tag{2.31}$$

The well-known ID3 decision tree algorithm was introduced by Quinlan, 1986, to divide the data set by information gain. However, there is a huge disadvantage in using information gain as a splitting rule. For instance, if we consider an attribute that has only one sample per branch, such as the ID number, the purity of the branch node can be maximized. However, the decision tree obtained in this way cannot be generated to predict new samples effectively, so it does not make any sense. Quinlan, 1993, introduced another algorithm named C4.5, which used the combination of information gain and gain ratio to choose the attribute with the best division rather than the information gain alone.

Definition 2.2.3 The **Gain ratio** of set D is defined as:

$$\text{Gain ratio } (D, X) = \frac{\mathrm{IG}(D, X)}{\mathrm{IV}(X)} \tag{2.32}$$

where

$$\mathrm{IV}(X) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|} \tag{2.33}$$

is called the intrinsic value. Generally speaking, the intrinsic value will increase as the number of the attribute (V) increased. Thus, the gain ratio has a preference for attributes with a small number of values. The C4.5 algorithm by Quinlan, 1993, first chooses the attributes with higher information gain than average and then obtains the attribute by calculating the gain ratio.

Another well-known algorithm based on the decision tree is the classification and regression tree (CART), which was first introduced by Breiman et al., 1984. The CART algorithm used the Gini index to measure the impurity of the data set.

Definition 2.2.4 The **Gini index** of one data set D for a categorical attribute X is defined as:

$$\text{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2, \tag{2.34}$$

$$\text{Gini index } (D, X) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Gini } (D^v) . \tag{2.35}$$

Figure 2.1 shows a decision tree for the same Hepatitis data set with only categorical attributes using the Gini index. At first, we have the whole data set with 153 samples in it, 121 of which belong to class 2. After splitting the data set by the explanatory variable called "Malaise", there will be two subgroups. For these two sub-groups, we can keep splitting the subgroups until all the explanatory variables are used already or the subgroup is pure enough. Here, pure means the proportion of one class is approximately 1.

We just consider the discrete variables and discuss how a data set can be divided into several groups by those features. Data in our daily life tend to contain continuous attributes. How to divide continuous attributes becomes a problem. Compared to discrete variables, continuous attributes cannot divide nodes by possible values. Therefore, discretization is needed to deal with the problem. The simplest strategy is the bi-partition method, which is used by Quinlan, 1993.

Suppose we want to divide the set based on a continuous attribute X with n different values with sorted values $\{x_1, \ldots, x_n\}$ on the data set D. For any arbitrarily dividing point t, the set D can be divided into two subsets, D_t^- and D_t^+, where D_t^- contains the value not greater than t on attribute X and D_t^- contains the value greater than t on attribute X. Since any value in the interval $[a_i, a_{i+1}) , (i = 1, \ldots, n-1)$ is divided into the same result,

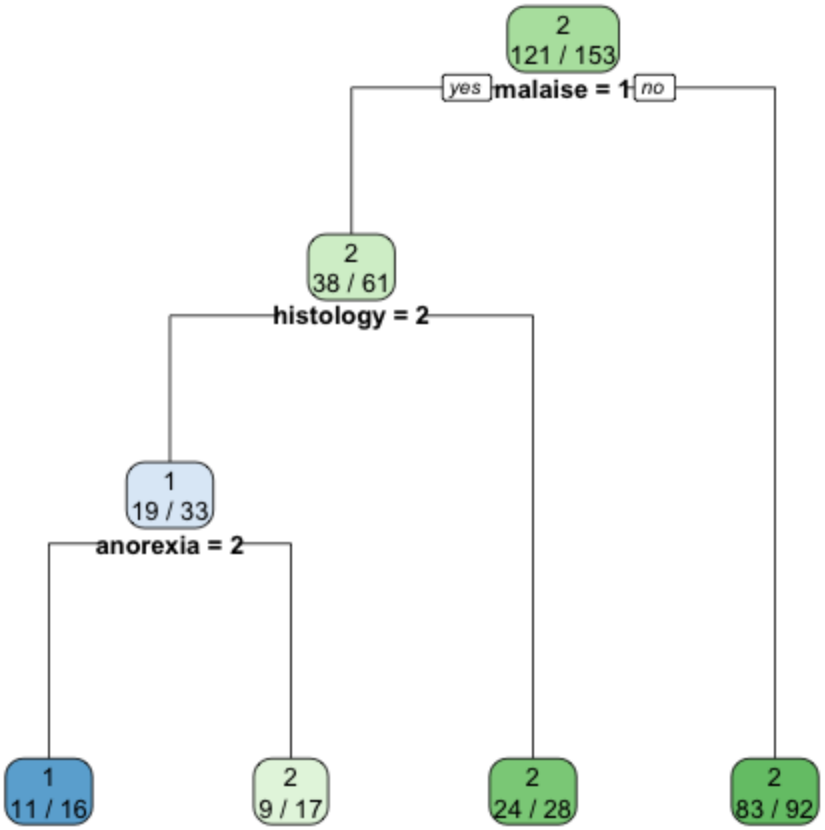

Figure 2.1: An example of decision tree with categorical variable by Gini index.

the partition point set of $n-1$ elements can be considered for the continuous attribute X, which is shown as Equation 2.36, i.e.,

$$T = \left\{ \frac{x^i + x^{i+1}}{2} | 1 \leq i \leq n-1 \right\}. \tag{2.36}$$

Then, as in the discrete case, the optimal segmentation point is selected to divide the sample set. The optimazation rule is to maximize the information gain by Equation 2.31.

Definition 2.2.5 The **Gini index** of D for a continuous attribute X is defined as:

$$\begin{aligned} \mathrm{IG}(D, X) &= \max_{t \in T} \mathrm{IG}(D, X, t) \\ &= \max_{t \in T} (\mathrm{Ent}(D) - \sum_{\lambda \in \{-,+\}} \frac{\left|D_t^\lambda\right|}{|D|} \mathrm{Ent}\left(D_t^\lambda\right)), \end{aligned} \tag{2.37}$$

where a weight $\left|D_t^\lambda\right| / |D|$ is added to resolve the uneven number of subsets.

If the current partition attribute is continuous, it can also be used as the partition attribute of its descendant node. Figure 2.2 shows a decision tree based on the same Hepatitis data set with categorical attributes and one continuous attribute, "Age". It is easy to see that the attribute "Age" is used to divide the data set more than once.

2.2.3 Regression Tree

There are two splitting rules introduced in Section 2.2.2, Information Gain and Gini Index, the main idea of which is to obtain several subsets by measuring the purity. However, if the response is continuous, it is meaningless for us to measure a set of samples by purity. Thus, Breiman et al., 1984, introduced the regression tree with the least-squares error criterion to choose the splitting attribute for a continuous response.

Suppose a learning sample D consists of n cases $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$. Our aim is to predict a continuous response y by a predictor $d(\boldsymbol{x}, \boldsymbol{\beta})$ where $d(\boldsymbol{x}, \boldsymbol{\beta})$ is a function of $\boldsymbol{x}$ with parameters $\boldsymbol{\beta}$. For instance, $d(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{x}'\boldsymbol{\beta}$ if we choose a linear model. Before we describe the principles of the regression tree, we need to know the definition of the cost function.

The cost function (loss function) returns to a non-negative value to measure the quality of the predictor. In statistics, a loss function is used for parameter estimation.

Notation 2.2.3.1 Let $\mathrm{R}^{\mathrm{D}}(d(\mathbf{x}, \boldsymbol{\beta}))$ denote the value of the cost function based on the model $d(\mathbf{x}, \boldsymbol{\beta})$ on the sample set D, where $\boldsymbol{\beta}$ is the parameter.

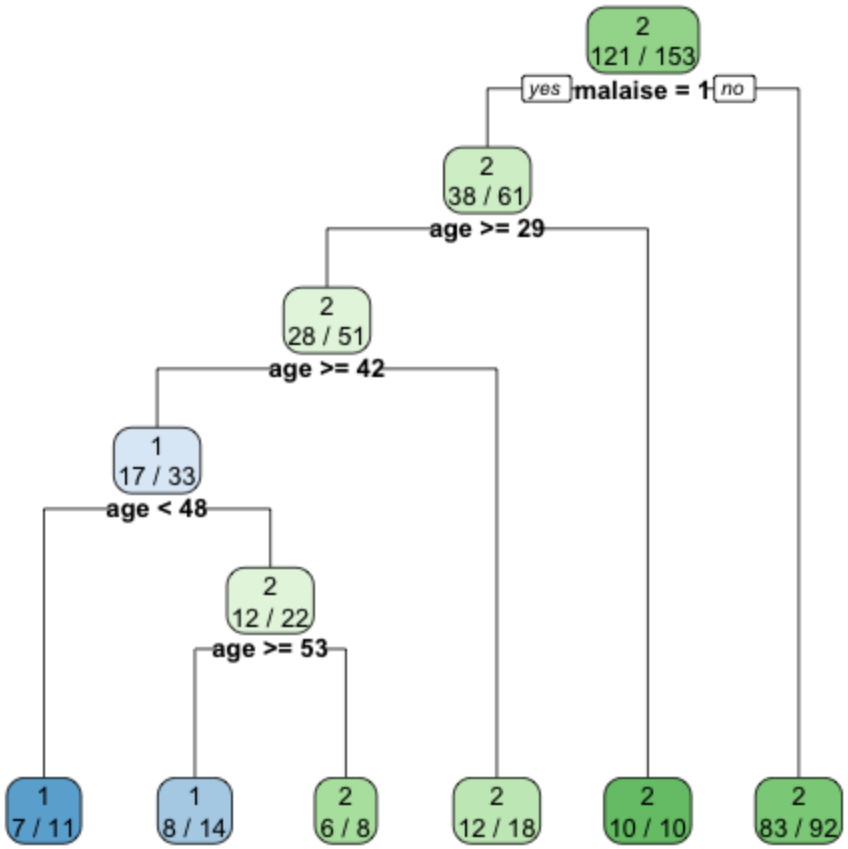

Figure 2.2: An example of decision tree with continuous variable.

For easier computation, the accuracy classically used in regression is the average squared error. Breiman et al., 1984, used the least-squares error criterion to choose the splitting attribute for a continuous response.

Definition 2.2.6 The **Mean Squared Error(MSE)** is defined as:

$$\text{MSE}(D) = \frac{1}{n}\sum_{i=1}^{n}(y_i - d(\mathbf{x}_i, \boldsymbol{\beta}))^2. \tag{2.38}$$

To have a good prediction of y based on the value of $\mathbf{x}$, we want the cost function to be as small as possible. Then, the parameter $\boldsymbol{\beta}$ is estimated by $\hat{\boldsymbol{\beta}}$ which minimizes the $\mathrm{R^D}(d(\mathbf{x}, \boldsymbol{\beta}))$. That is,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, \mathrm{R^D}(\mathrm{d}(\mathbf{x}, \boldsymbol{\beta})) \, . \tag{2.39}$$

Breiman et al., 1984, mentioned the value that minimizes the Equation 2.38 is the average of y_i. That is, for a group of data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, the estimate that minimizes the average squared error is

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i. \tag{2.40}$$

For a set of data D with a continuous response, the best division for this data set is the one that maximizes the difference between the Mean Square Error (MSE) before division and after:

$$\Delta\mathrm{MSE} = \mathrm{MSE}(D) - \mathrm{MSE}(D_1) - \mathrm{MSE}(D_2). \tag{2.41}$$

Thus, a regression tree is formed by iteratively splitting the nodes to maximize the decrease in the MSE of Equation 2.41.

2.2.4 Evaluation of Classification and Regression

Once we stop dividing the entire data set, we can obtain a complete tree model similar to what we have in Figure 2.1 and Figure 2.2. For data that need to be classified or regressed in the future, we will obtain a corresponding child node after entering all the inputs into the tree model. For the classification tree, we use the mode of the response of all the training data in the child node to predict the label corresponding to the new data. As for the regression tree, we can use the mean value of the response of all the training data in the child node to predict the numerical value corresponding to the new data.

For classification problems, we generally use four values specifically, true positive, true negative, false positive, and false negative to measure the quality of the classification. The definitions of these four measurements are in Definition 2.2.7.

Definition 2.2.7 A **false positive (FP)** is an error in binary classification in which a test result incorrectly predicts a negative case. A **false negative (FN)** is an opposite error where the test result incorrectly predicts a positive case. Contrary to the two incorrect results, there are two correct results, called **true positive (TP)** and **true negative (TN)**.

One simple measurement to evaluate one classification model is accuracy, which is the number of cases with correct prediction over the total number of cases for prediction. The method to calculate accuracy is in Definition 2.2.8.

Definition 2.2.8 Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2.42}$$

where TP, FP, TN, FN are defined in Definition 2.2.7.

Accuracy only works well when the number of samples belonging to each category is equal. For example, suppose 98% of samples are in class 1, and only 2% of samples are in class 2. Then, by simply predicting each training sample belonging to class 1, our model can easily obtain 98% training accuracy. However, the prediction is meaningless. It may also cause a dangerous result in real life. For instance, suppose we are fitting a model to predict whether a patient has a kind of cancer or not, class 1 for no and class 2 for yes. Ignoring the 2% error is fatal to those patients with cancer. As a result, there are many possible measurements proposed for classification, such as recall, precision, and F-score. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that

were retrieved. The F-score is calculated from the precision and recall of the test, which is the harmonic mean of the precision and recall. Generally speaking, recall and precision trade each other, so that an F-score considers these two metrics at the same time, making it better to evaluate the performance of the model. The definition of precision, recall, and F-score is in Definition 2.2.9.

Definition 2.2.9 Precision and **recall** are defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.43}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.44}$$

where TP, FP, TN, FN are defined in Definition 2.2.7. After calculating the value of precision and recall, the **F-score** is defined by:

$$\text{F} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.45}$$

As for the regression problems, Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R- Square (R^2) are widely used to evaluate the performance of the model. The MSE is defined in Definition 2.2.6. The RMSE is the square root of MSE, making the metric more sensitive to the scale of response. The method to calculate the RMSE is in Definition 2.2.10.

Definition 2.2.10 Suppose the model $d(\mathbf{x}, \boldsymbol{\beta})$ on the sample set D is used to predict the response y, where $\boldsymbol{\beta}$ is the parameter. The Root Mean Squared Error(RMSE) is defined as:

$$\text{RMSE}(D) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - d(\mathbf{x}_i, \boldsymbol{\beta}))^2}. \tag{2.46}$$

Generally speaking, when we have a set of data and want to use this data to fit a regression or classification model, our first step is to divide the training set, the validation set, and the testing set. The training set is used to fit the model. The validation set is used to adjust high-dimensional parameters. We use the testing set to simulate future unknown data to test the predictive effect of the model on future data. Therefore, the predictions of the model on the testing set are often not as good as on the training set, because in the process of fitting the model, we make full use of the information in the training set, while the testing set does not contribute any information to the model. What we hope is that we can achieve an ideal performance on the testing set. Therefore, there are two bad situations when evaluating the model. The first one, called under-fitting, is when the model is not very good even on the training set. This means that the information we extracted is not enough, and often we need to increase the number of explanatory variables or collect more samples. Another situation is over-fitting. Under this situation, the model is useful only to the training set, and not to any other data sets.

In Chapter 3, we will not only propose how to build a tree model for symbolic data but also how to come up with some evaluation methods for symbolic trees. In the next chapter, we will also introduce some metrics for evaluation for symbolic responses.

Chapter 3

Methodology

In this chapter, we classify the methods to be developed according to different types of variables. Although there are four different types of symbolic variables, we will only consider modal multi-valued, interval-valued, and histogram-valued variables in this chapter. There are multiple reasons why it is unnecessary to consider the multi-valued data no matter they be as a response variable or as explanatory variables. Section 3.1, Section 3.2 and Section 3.3, respectively, describe the methods to deal with different kinds of response and explanatory variables. Section 3.4 considers all the scenarios that use the new methodologies related to symbolic data.

3.1 Multi-valued Variable

To develop our methodology, we need a splitting variable. Suppose we have a multi-valued variable X as our splitting attribute. One problem that cannot be ignored is the size of the domain of X. Definition 2.1.3 defines a multi-valued variable showing that it takes one or more values from the list of values in its domain $\mathcal{X}$. In this case, the number of possible values of X should be $2^n - 1$, where n is the number of values in $\mathcal{X}$. In other words, the number would be huge even if we only have a slightly large number of n. Take $n = 5$ as an

instance; this size is analyzable in traditional decision tree methods. However, $2^n - 1 = 31$ is an unmanageable size if we consider all possible scenarios since the gain ratio tends to avoid the attributes with a large number of possible values. As a result, if we consider a modal multi-valued variable with 31 possible values, it is likely to remove the attribute by information gain, resulting in a loss of information. One possible way to avoid a huge number of values is to have a much smaller number of possible values by clustering. There are many clustering methods for symbolic data such as the hierarchy-pyramid clusters in Billard and Diday, 2019. Whatever clustering method we use, a multi-valued variable first needs to be changed to a similar form as for a modal multi-valued realization for calculating the dissimilarity measure. As a result, the approach for a multi-valued explanatory variable is the same as that for a model multi-valued attribute, which will be explained in detail in Section 3.2.1.

On the other hand, suppose there is a multi-valued response variable Y with domain $\mathcal{Y}$ requiring analysis. When the number n of values in $\mathcal{Y}$ is large, there are too many categories to obtain an accurate result. If we still use clustering methods to find smaller sizes of category, we can only know that each response variable on the testing set belongs to one cluster and cannot know the specific value. As a result, we cannot achieve good prediction results since we usually do not know what the practical meaning of each cluster is. Even if we can predict to which cluster one testing set belongs, it is hard to interpret the result. However, the data still cannot be analyzed even though n is small enough. For instance, we may have many different types of employees and want to know the turnover of each type of employee. In this situation, the data we collect are in the symbolic type because we are not focusing on individuals but a group of people. If we use a multi-valued response variable, it is unreasonable because each type of employee has either resigned or not resigned for example. Then the response variables of all groups are the same, $\{\text{yes}, \text{no}\}$. In this case, it is meaningless to analyze data in this

situation. Generally speaking, it would be more meaningful to use the modal multi-valued type when n is small because we can then distinguish the groups by weights.

3.2 Splitting Measures for Symbolic Response Variable

For classic decision trees, one of the most important steps is to find a splitting rule for each kind of response variable. In other words, how can we compare different divisions. Section 2.2 introduced the Information Gain and Gini index for classification problems with a categorical response variable and Mean Square Error (MSE) for regression problems with numerical response variables. In this section, we will consider the splitting rule for four different types of response variables.

3.2.1 Modal Multi-valued Response Variable

In this section, we assume that the response variable is modal multi-valued with different types of explanatory variables. Suppose we have a random sample of size n, $Y_u, u = 1, \ldots, n$, with Y_u taking modal multi-valued values from the domain $\mathcal{Y} = \{Y_1, \ldots, Y_s\}$. The observation ω_u can be rewritten as the following form:

$$\xi(\omega_u) = \{Y_k, p_{uk}; k = 1, \ldots, s\}. \tag{3.1}$$

Since the domain $\mathcal{Y}$ is fixed, the possible values of the response variable are the same for different categories. The only way to distinguish the response variable of observations is by comparing the probability set $\{p_{uk}; k = 1, \ldots, s\}$ for $u = 1, \ldots, n$, where n is the number of aggregated groups in the data. What we want to predict are the probabilities rather than the possible values. Therefore, the decision tree for a modal multi-valued response variable is more like a regression tree than a classification tree.

There are two approaches to deal with the modal multi-valued response variable. The first one is to use the difference in MSE to evaluate the splitting as in a classical decision tree since the output is a numerical probability. Here, we are analogous to the situation of a regression tree for numerical outputs; our goal is to divide the data set into several subsets. The closer the output in each subset, the better the splitting is. In other words, the smaller the MSE, the smaller is the variation in the subsets. The second approach is using the similarity or the distance between multi-valued values introduced in Section 2.1.4. In this method, we use the similarity or the distance to measure the purity or variation within each subset. The details of these two methods are listed below.

Mean Square Error (MSE)

To simplify the question, we first consider the binary case. That means there are only two possible values in $\mathcal{Y} = \{Y_1, Y_2\}$. Then, the observation ω_u for $u = \{1, \ldots, n\}$ becomes:

$$\xi(\omega_u) = \{Y_1, p_u; Y_2, 1 - p_u\}. \tag{3.2}$$

In this binary case, the only thing we need to predict is the probability of the value (Y_1, p) for any one observation. The output is one single numerical variable p with $0 \leq p \leq 1$. Suppose we have a set of symbolic data D with n categories and a modal multi-valued response variable taking values as in Equation 3.2 and the index set for D is $I = \{1, \ldots, n\}$. The estimate of p based on the whole set D is the sample mean of p_u, $u \in I$, which can be calculated by $\hat{p} = \frac{1}{n} \sum_{u \in I} p_u$. Then, the MSE for the set is

$$\text{MSE}(D) = \frac{1}{n} \sum_{u \in I} (p_u - \hat{p})^2. \tag{3.3}$$

Suppose the set is divided into two groups D_1 and D_2 by an attribute. The MSE of these two subsets can be calculated by:

$$\text{MSE}(D_1) = \frac{1}{n_1} \sum_{u \in I_1} (p_u - \hat{p}_1)^2, \tag{3.4}$$

$$\text{MSE}(D_2) = \frac{1}{n_2} \sum_{u \in I_2} (p_u - \hat{p}_2)^2, \tag{3.5}$$

where I_1 and I_2 are the index sets, n_1 and n_2 are the sizes of two groups D_1 and D_2, respectively, and $\hat{p}_1$ and $\hat{p}_2$ are the respective sample means.

Definition 3.2.1 For a set of symbolic data D with a modal multi-valued response variable taking binary values as in Equation 3.2, a best division for this data set is the one which maximizes the following equation:

$$\Delta\text{MSE} = \text{MSE}(D) - \text{MSE}(D_1) - \text{MSE}(D_2), \tag{3.6}$$

where MSE(D), MSE(D_1), and MSE(D_2) are given in Equation 3.3, Equation 3.4, and Equation 3.5, respectively.

For a modal multi-valued response variable with multiple possible values in the domain $\mathcal{Y} = \{Y_1, \ldots, Y_k\}$, the observation ω_u becomes:

$$\xi(\omega_u) = \{Y_1, p_{u1}; Y_2, p_{u2}; \ldots; Y_k, 1 - \sum_{i=1}^{k-1} p_{ui}\}. \tag{3.7}$$

In this multiple case with k possible values, the only thing we need to predict is the probability of all the values except the last value for any observation. The output is the $(k-1)$ dimensional numerical variable $\mathbf{p} = (p_1, \ldots, p_{k-1})'$ with $0 \leq p_i \leq 1$ for $i = 1, \ldots, k-1$. Suppose there is a set of symbolic data D with n categories and a modal multi-valued response variable taking values as in Equation 3.7 and the index set is $I = \{1, \ldots, n\}$. There are n groups in

D with response variable $\mathbf{p}_1 = (p_{11}, \ldots, p_{1,k-1})'$, $\ldots$, $\mathbf{p}_n = (p_{n1}, \ldots, p_{n,k-1})'$. The estimate of $\mathbf{p}$ based on the whole set D is the sample mean of $\mathbf{p}_u$, $u \in I$, which can be calculated by

$$\hat{\mathbf{p}} = \frac{1}{n}\sum_{u\in I}\mathbf{p}_u = (\frac{1}{n}\sum_{u\in I}p_{u1}, \ldots, \frac{1}{n}\sum_{u\in I}p_{u,k-1})'.$$

Then, the MSE for the set is

$$\text{MSE}(D) = \frac{1}{n}\sum_{u\in I}(\mathbf{p}_u - \hat{\mathbf{p}})'(\mathbf{p}_u - \hat{\mathbf{p}}) = \frac{1}{n}\sum_{u\in I}\sum_{j=1}^{k-1}(p_{uj} - \bar{p}_{\cdot j})^2 \tag{3.8}$$

where $\bar{p}_{\cdot j} = \frac{1}{n}\sum_{u\in I}p_{uj}$. Suppose the set is divided into two groups D_1 and D_2 by an attribute. The MSE of these two subsets can be calculated by:

$$\text{MSE}(D_1) = \frac{1}{n_1}\sum_{u\in I_1}\sum_{j=1}^{k-1}(p_{uj} - \bar{p}^{(1)}_{\cdot j})^2, \tag{3.9}$$

$$\text{MSE}(D_2) = \frac{1}{n_2}\sum_{u\in I_2}\sum_{j=1}^{k-1}(p_{uj} - \bar{p}^{(2)}_{\cdot j})^2, \tag{3.10}$$

where I_1 and I_2 are the index sets, n_1 and n_2 are the sizes of two groups D_1 and D_2, respectively, and $\bar{p}^{(1)}$ and $\bar{p}^{(2)}$ are the sample means calculated by $\bar{p}^{(1)}_{\cdot j} = \frac{1}{n_1}\sum_{u\in I_1}p_{uj}$ and $\bar{p}^{(2)}_{\cdot j} = \frac{1}{n_2}\sum_{u\in I_2}p_{uj}$, respectively.

Definition 3.2.2 For a set of symbolic data D with a modal multi-valued response variable taking multiple values as in Equation 3.7, a best division for this data set is the one which maximizes the following equation:

$$\Delta\text{MSE} = \text{MSE}(D) - \text{MSE}(D_1) - \text{MSE}(D_2), \tag{3.11}$$

where MSE(D), MSE(D_1), and MSE(D_2) are given in Equation 3.8, Equation 3.9, and Equation 3.10, respectively.

Similarity or Distance

Suppose there is a set of symbolic data D with a modal multi-valued response variable taking values as in Equation 3.7 and the index set is $I = \{1, \ldots, n\}$. The output is the $(k-1)$ dimensional numerical variable $\mathbf{p} = (p_1, \ldots, p_{k-1})'$ with $0 \leq p_i \leq 1$ for $i = 1, \ldots, k-1$. The i^{th} output in the symbolic set is: $\xi(\omega_i) = \{Y_k, p_{ik}; k = 1, \ldots, s\}$.

In Section 2.1.4, several similarity measures for multi-valued data are introduced. Here we take the simple distance introduced by Definition 2.1.13 as an example. As a result, the distance between for any two observations i and j in the symbolic set D can be calculated by:

$$d^2(\omega_i, \omega_j) = \sum_{k=1}^{s} \left(\sum_{u=1}^{n} p_{uk} \right)^{-1} (p_{ik} - p_{jk})^2 . \tag{3.12}$$

Definition 3.2.3 For a set of symbolic data D with a modal multi-valued response variable taking multiple values as in Equation 3.7 and n observations, the average distance within the set can be calculated by

$$d(\mathrm{D}) = \frac{1}{n} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d(\omega_i, \omega_j) = \frac{1}{n} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sqrt{\sum_{k=1}^{s} \left(\sum_{u=1}^{n} p_{uk} \right)^{-1} (p_{ik} - p_{jk})^2}. \tag{3.13}$$

Suppose the set is divided into two groups D_1 and D_2 by an attribute. The average distance of these two subsets can be calculated by:

$$d(\mathrm{D}_1) = \frac{1}{n_1} \sum_{i \in I_1} \sum_{j>i} d(\omega_i, \omega_j) = \frac{1}{n_1} \sum_{i \in I_1} \sum_{j>i} \sqrt{\sum_{k=1}^{s} \left(\sum_{u \in I_1} p_{uk} \right)^{-1} (p_{ik} - p_{jk})^2}, \tag{3.14}$$

and

$$d(\mathrm{D}_2) = \frac{1}{n_2} \sum_{i \in I_2} \sum_{j>i} d(\omega_i, \omega_j) = \frac{1}{n_2} \sum_{i \in I_2} \sum_{j>i} \sqrt{\sum_{k=1}^{s} \left(\sum_{u \in I_2} p_{uk} \right)^{-1} (p_{ik} - p_{jk})^2}, \tag{3.15}$$

where I_1 and I_2 are the index sets, and n_1 and n_2 are the sizes of two groups D_1 and D_2, respectively.

Definition 3.2.4 For a set of symbolic data D with a modal multi-valued response variable taking multiple values as in Equation 3.7, a best division for this data set is the one which maximizes the following equation:

$$\Delta d = d(\mathrm{D}) - d(\mathrm{D}_1) - d(\mathrm{D}_2), \tag{3.16}$$

where $d(\mathrm{D})$, $d(\mathrm{D}_1)$, and $d(\mathrm{D}_2)$ are given in Equation 3.13, Equation 3.14, and Equation 3.15, respectively.

Since there are several possible dissimilarity measurements introduced in Section 2.1.4, only one of them is selected as an example here. We can replace the distance in Definition 3.2.3 and use the difference in other kinds of distances after division according to the splitting rules.

3.2.2 Interval-valued Response Variable

In this section, we assume that the response variable is interval-valued. Suppose we have a random sample of size n, $D = \{Y_u, u \in I\}$ and the index set is $I = \{1, \ldots, n\}$, with Y_u taking values as in the following form:

$$\xi(\omega_u) = [a_u, b_u]. \tag{3.17}$$

For the set D, we use the corresponding mean to estimate the upper and lower bounds, respectively. That is,

$$\hat{a} = \bar{a} = \frac{1}{n}\sum_{u \in I} a_u, \quad \hat{b} = \bar{b} = \frac{1}{n}\sum_{u \in I} b_u. \tag{3.18}$$

We want to make the intervals in the set as similar as possible. Here "similar" is measured in terms of the overlap; thus the more overlap there is between two intervals, the more they are similar to each other. Considering that the sample size cannot have a significant impact on the measurement, rather than using the total sum of squares (SST) introduced in Equation 2.6, we use the sample variance which can be calculated by $\mathrm{SST}(D)/n$ to measure the purity of the set D, i.e.,

$$\begin{aligned}\mathrm{MSE}(D) = \mathrm{SST}(D)/n = &\sum_{u\in I}\left[\left(a_u+b_u\right)/2-\bar{Y}\right]^2/n \\ &+\sum_{u\in I}\left[\left(a_u-\bar{Y}_u\right)^2+\left(a_u-\bar{Y}_u\right)\left(b_u-\bar{Y}_u\right)+\left(b_u-\bar{Y}_u\right)^2\right]/3n,\end{aligned} \tag{3.19}$$

where $\bar{Y}=\frac{1}{2n}\sum_{u\in I}\left(a_u+b_u\right)$ and $\bar{Y}_u=\frac{1}{2}\left(a_u+b_u\right)$.

Suppose the set D is divided to two groups D_1 and D_2 by an attribute with the index set I_1 and I_2. The sample variance of these two subsets can be calculated by:

$$\begin{aligned}\mathrm{MSE}(D_1) = &\sum_{u\in I_1}\left[\left(a_u+b_u\right)/2-\bar{Y}_1\right]^2/n_1 \\ &+\sum_{u\in I_1}\left[\left(a_u-\bar{Y}_u\right)^2+\left(a_u-\bar{Y}_u\right)\left(b_u-\bar{Y}_u\right)+\left(b_u-\bar{Y}_u\right)^2\right]/3n_1,\end{aligned} \tag{3.20}$$

$$\begin{aligned}\mathrm{MSE}(D_2) = &\sum_{u\in I_2}\left[\left(a_u+b_u\right)/2-\bar{Y}_2\right]^2/n_2 \\ &+\sum_{u\in I_2}\left[\left(a_u-\bar{Y}_u\right)^2+\left(a_u-\bar{Y}_u\right)\left(b_u-\bar{Y}_u\right)+\left(b_u-\bar{Y}_u\right)^2\right]/3n_2,\end{aligned} \tag{3.21}$$

where n_1 and n_2 are the sizes of two groups D_1 and D_2, respectively, and $\bar{Y}_1$ and $\bar{Y}_2$ are the sample means of the two groups, respectively.

Definition 3.2.5 For a set of symbolic data D with an interval response variable taking values as in Equation 3.17, a best division for this data set is the one which maximizes the following equation:

$$\Delta\mathrm{MSE} = \mathrm{MSE}(D) - \mathrm{MSE}(D_1) - \mathrm{MSE}(D_2), \tag{3.22}$$

where MSE(D), MSE(D_1), and MSE(D_2) are given in Equation 3.19, Equation 3.20, and Equation 3.21, respectively.

Similarity or Distance

Suppose there is a set of symbolic data D with an interval-valued response variable taking values as in Equation 3.17 and the index set is $I = \{1, \ldots, n\}$. The output is an interval variable $[a_u, b_u]$ for the u^{th} observation.

In Section 2.1.4, the similarity for interval-valued data is introduced by Definition 2.1.18. As a result, the distance between any two interval observations i and j in the symbolic set D can be calculated by:

$$d\left(w_i, w_j\right) = \sum_{k=1}^{3} d_k\left(w_i, w_j\right), \tag{3.23}$$

with

$$d_1\left(w_i, w_j\right) = \left\| b_i - a_i \right| - \left| b_j - a_j \right\| / k, \tag{3.24}$$

where

$$k = \left| \text{Max}\left(b_i, b_j\right) - \text{Min}\left(a_i, a_j\right) \right| \tag{3.25}$$

is the length of the entire distance spanned by w_i and w_{u_2} , with

$$d_2\left(w_i, w_j\right) = \left(\left| b_i - a_i \right| + \left| b_j - a_j \right| - 2I \right) / k, \tag{3.26}$$

where

$$I = \text{Max}\left(\text{Min}\left(b_i, b_j\right) - \text{Max}\left(a_i, a_j\right), 0\right) \tag{3.27}$$

is the length of the intersection of intervals $[a_i, b_i]$ and $[a_j, b_j]$ if the intervals overlap, and 0 otherwise with

$$d_3\left(w_i, w_j\right) = \left| a_i - a_j \right| / \left| x \right|, \tag{3.28}$$

where x is the total length covered by the observed values of X, i.e.,

$$|x| = \max_u\{b_u\} - \min_u\{a_u\}. \tag{3.29}$$

Definition 3.2.6 For a set of symbolic data D with a interval-valued response variable and n elements, the average distance within the set can be calculated by

$$d(\mathrm{D}) = \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d\left(\omega_i, \omega_j\right), \tag{3.30}$$

where $d\left(\omega_i, \omega_j\right)$ is defined in Equation 3.23.

Suppose the set is divided into two groups D_1 and D_2 by an attribute. The average distance of these two subsets can be calculated by:

$$d(\mathrm{D}_1) = \frac{1}{n_1}\sum_{i\in I_1}\sum_{j>i} d\left(\omega_i, \omega_j\right), \tag{3.31}$$

and

$$d(\mathrm{D}_2) = \frac{1}{n_2}\sum_{i\in I_2}\sum_{j>i} d\left(\omega_i, \omega_j\right), \tag{3.32}$$

where I_1 and I_2 are the index sets, and n_1 and n_2 are the sizes of two groups D_1 and D_2, respectively, and $d\left(\omega_i, \omega_j\right)$ is defined in Equation 3.23.

Definition 3.2.7

For a set of symbolic data D with an interval-valued response variable taking multiple values as in Equation 3.17, a best division for this data set is the one which maximizes the following equation:

$$\Delta d = d(\mathrm{D}) - d(\mathrm{D}_1) - d(\mathrm{D}_2), \tag{3.33}$$

where $d(\mathrm{D})$, $d(\mathrm{D}_1)$, and $d(\mathrm{D}_2)$ are given in Equation 3.30, Equation 3.31, and Equation 3.32, respectively.

Similarly, we can replace the measurement of distance in Definition 3.2.6 to capture other kinds of splitting measures. There are many existing works for interval-valued regression. In addition to the above two methods similar to modal multi-valued, we can also refer to the measurement of interval-valued regression.

Measurements from Interval-valued Regression

Since Billard and Diday, 2000, proposed the first method, different methods have been introduced to perform linear regression analysis on symbolic data, especially interval-valued data. By fitting a linear regression model to the center point and the range of the interval, their model can predict the upper and lower boundaries, respectively. de Carvalho et al., 2004, converted the interval variables into the center point and range variables, and performed classical regression analysis on the central point and the range variables. Billard and Diday, 2006, introduced the bi-variate center and range method (BCRM method) based on the center and range method (CRM method) in de Carvalho et al., 2004. Therefore, it is obvious that the key point of these linear regressions for interval-valued data is to consider the center and range of intervals.

There are two ways to fit the regression, either to use the upper and lower boundaries, respectively, or to use the center point and range. Generally speaking, we choose range and center instead of the lower bound and upper bound when fitting a linear regression model. The main reason is that there is an assumption that explanatory variables are independent of each other in a linear model. In the tree-based model, we do not have any assumptions about the relationship between explanatory variables. Here we choose the upper and lower boundaries to fit two regression trees. The main reason is we can obtain the value of the center and range once we collect the two boundaries of the intervals because there is a one-to-one correspondence between the two endpoints and the center-range values. In addition, we do not have to calculate the range and center for the interval on the training set or calculate

the estimated bounds by the estimated center and range on the testing set. We can directly use the traditional regression tree to predict the estimated lower bound and estimated upper bound of the testing set.

Here are the steps to obtain the prediction of intervals on the testing set:

Step 1: For the samples in the training set, we can fit a regression tree called the lower-bound tree, T_L, where T_L uses all the explanatory variables to predict the lower-bound of all the intervals. Here the response variable is a numerical variable representing the lower bound, and MSE will be used to construct the tree.

Step 2: For the samples in the training set, we can fit another regression tree called the upper-bound tree, T_U, where T_U uses all the explanatory variables to predict the upper-bound of all the intervals.

Step 3: Given the values of explanatory variables on the testing set, we can use T_L to estimate the lower bound for all the samples. For the u^{th} sample from the testing set, the estimated lower-bound is assumed to be $\hat{a}_u$, which is defined in Equation 3.18.

Step 4: Given the values of explanatory variables on the testing set, we can use T_U to estimate the upper bound for all the samples. For the u^{th} sample from the testing set, the estimated upper-bound is assumed to be $\hat{b}_u$, which is also defined in Equation 3.18.

Step 5: For the u^{th} sample from the testing set, the estimated interval is $[\hat{a}_u, \hat{b}_u]$.

3.2.3 Histogram-valued Response Variable

In this section, we assume that the response variable is histogram-valued. Suppose we have a random sample of size n, $D = \{Y_u, u \in I\}$ and the index set is $I = \{1, \ldots, n\}$, with Y_u taking values in the following form:

$$\xi(\omega_u) = \{[a_k, b_k), p_{uk}; k = 1, ..., s\}. \tag{3.34}$$

Here, we assume $s_u = s$ is the same for all categories; otherwise, we can reorganize the sub-intervals $[a_k, b_k)$. As in the modal multi-valued case, what we need to estimate are the probabilities of each interval, $\mathbf{p}_u = (p_{u1}, \ldots, p_{u,k-1})'$. There are n groups in D with response variable $\mathbf{p}_1 = (p_{11}, \ldots, p_{1,k-1})'$, $\ldots$, $\mathbf{p}_n = (p_{n1}, \ldots, p_{n,k-1})'$. We use the sample mean of $\mathbf{p}_u$, $u \in I$, to estimate $\mathbf{p}$ based on the whole set D, which can be calculated by

$$\hat{\mathbf{p}} = \frac{1}{n}\sum_{u\in I}\mathbf{p}_u = (\frac{1}{n}\sum_{u\in I}p_{u1}, \ldots, \frac{1}{n}\sum_{u\in I}p_{u,k-1})'.$$

We want to make the histograms within one set as close as possible to each other, and the histograms between different sets as heterogeneous as possible. Here "close" is measured in terms of the overlap; thus the more overlap there is between two histograms, the closer they are to each other. In other words, we would like to have the overlap between those histograms as large as possible. Therefore, we use S^2 which can be calculated by SST$/n$ where SST is defined in Equation 2.6 to measure the purity of set D, i.e.,

$$\begin{aligned}\mathrm{MSE}(D) = &\sum_{u\in I}\sum_{k=1}^{s}\left\{\left[(a_k+b_k)/2-\bar{Y}\right]^2 p_{uk}\right\}/n \\ &+\sum_{u\in I}\sum_{k=1}^{s}\left\{\left[\left(a_k-\bar{Y}_u\right)^2+\left(a_k-\bar{Y}_u\right)\left(b_k-\bar{Y}_u\right)+\left(b_k-\bar{Y}_u\right)^2\right]p_{uk}\right\}/3n,\end{aligned} \tag{3.35}$$

where $\bar{Y} = \frac{1}{2n}\sum_{u\in I}\sum_{k=1}^{s}(a_k+b_k)\,p_{uk}$ and $\bar{Y}_u = \sum_{k=1}^{s}(a_k+b_k)\,p_{uk}/2$.

Suppose the set is divided into two groups D_1 and D_2 by an attribute with the index sets I_1 and I_2, respectively. The S^2 of these two subsets can be calculated by:

$$\begin{aligned}\mathrm{MSE}(D_1) = &\sum_{u\in I_1}\sum_{k=1}^{s}\left\{\left[(a_k+b_k)/2-\bar{Y}_1\right]^2 p_{uk}\right\}/n_1 \\ &+\sum_{u\in I_1}\sum_{k=1}^{s}\left\{\left[\left(a_k-\bar{Y}_u\right)^2+\left(a_k-\bar{Y}_u\right)\left(b_k-\bar{Y}_u\right)+\left(b_k-\bar{Y}_u\right)^2\right]p_{uk}\right\}/3n_1,\end{aligned} \tag{3.36}$$

and

$$\begin{aligned}\mathrm{MSE}(D_2) = &\sum_{u\in I_2}\sum_{k=1}^{s}\left\{\left[(a_k+b_k)/2-\bar{Y}_2\right]^2 p_{uk}\right\}/n_2 \\ &+\sum_{u\in I_2}\sum_{k=1}^{s}\left\{\left[\left(a_k-\bar{Y}_u\right)^2+\left(a_k-\bar{Y}_u\right)\left(b_k-\bar{Y}_u\right)+\left(b_k-\bar{Y}_u\right)^2\right]p_{uk}\right\}/3n_2,\end{aligned} \tag{3.37}$$

where n_1 and n_2 are the sizes of two groups D_1 and D_2, respectively, and where $\bar{Y}_1$ and $\bar{Y}_2$ are the sample means of the two groups, respectively.

Definition 3.2.8 For a set of symbolic data D with a histogram response variable taking values as in Equation 3.34, the best splitting measure for the histogram-valued response variable is the one which maximizes the change of SST, which can be calculated by the following equation:

$$\Delta\mathrm{MSE} = \mathrm{MSE}(D) - \mathrm{MSE}(D_1) - \mathrm{MSE}(D_2), \tag{3.38}$$

where $\mathrm{MSE}(D)$, $\mathrm{MSE}(D_1)$, and $\mathrm{MSE}(D_2)$ are given in Equation 3.35, Equation 3.36, and Equation 3.37, respectively.

Since there are also many distance/dissimilarity measurements defined for histogram-valued data, we can use those measurements to define some splitting measurements based on distance/dissimilarity like what we have for interval-valued data and modal multi-valued data.

3.3 Splitting Rules for Symbolic Dividing Attribute

3.3.1 Modal Multi-valued Explanatory Variable

Suppose we want to divide the set based on a model multi-valued explanatory variable $X = \{x_k, p_k; k = 1, \ldots, s\}$ taking values in $\mathcal{X} = \{x_1, \ldots, x_s\}$, with n different values $\{x_k, p_{uk}; k = 1, \ldots, s\}, u = 1, \ldots, n$, on the symbolic data set D.

For a binary situation, there are only two possible values in $\mathcal{X} = \{x_1, x_2\}$ with n different values $\{x_1, p_u^x; x_2, 1-p_u^x\}, u = 1, \ldots, n$, on the symbolic data set D, where p_u^x is the probability of x_1 in the u^{th} observation. In this case, the only value we can use to distinguish different observations is one single numerical variable p^x with $0 \leq p^x \leq 1$. Thus, we can regard p^x as a numerical variable. However, there are too many variables to consider if the number of values in $\mathcal{X}$ is large. We need to change one single modal multi-valued variable to $s-1$ numerical variables where s is the number of values in $\mathcal{X} = \{x_1, \ldots, x_s\}$. Thus the number of variables will be greatly increased by this method when there are many variables to consider, making the analysis more complicated. In this case, we are dividing the sample set by a $(s-1)$-dimensional vector rather than a single value. It is complex and time-consuming to find a threshold that can divide a set of vectors well and in a meaningful way. Thus, neither using the vectors as categories nor comparing a vector with one single value or a fixed vector is applicable in this case.

One way to deal with the multiple-cases modal multi-valued variable is to use cluster methods. Although it is impossible to exhaust all possible values for the vectors and regard them as categories, we can convert the modal multi-valued data into several clusters according to cluster methods and treat the clusters as divided category attributes. There are some definitions of dissimilarity and cluster methods in Section 2.1.4.

As we mentioned in Section 2.1.3, a multi-valued attribute can be transferred to a similar form with a modal multi-valued variable. Therefore, the methods in this section are also useful for a multi-valued variable.

3.3.2 Interval-valued Explanatory Variable

In Section 2.2.2 we talked about how to divide data based on continuous attributes. Compared to the continuous variable, the interval-valued realization is hard to be divided when using the bi-partition strategy introduced by Quinlan, 1993, since given a critical value, it is difficult

to compare an interval with a single value. However, it is possible to compare the lower and upper bound of an interval with a specific value. Therefore, a method to deal with the partition based on the interval-valued variable can be described as one that treats the lower and upper bounds to be two continuous variables.

Suppose we want to divide the set based on an interval attribute $V = [a, b]$ with N different sorted values $\{[a_1, b_1], \ldots, [a_N, b_N]\}$ on the symbolic data set D. Rather than consider the interval attribute V, we take two sub-variables V_{min} and V_{max} into consideration. Also, since the range and mean have a one-to-one correspondence with the lower and upper bound, this is equivalent to using the range and the midpoint as two numerical variables.

In addition, based on the idea of bi-partition, we can have three different situations when comparing an interval with a specific value. We still have to consider how to divide the set based on an interval attribute $V = [a, b]$ with N samples $\{[a_1, b_1], \ldots, [a_N, b_N]\}$ on the symbolic data set D. For arbitrary dividing point t, a set D can be divided into three subsets, D_t^-, D_t^0, D_t^+, where D_t^- contains intervals with the upper bound not greater than t on attribute V, D_t^0 consists of intervals which contain t on attribute V and D_t^- contains intervals with the lower bound greater than t on attribute V.

To achieve the best performance of division, we need to find the critical value which maximizes the splitting measures we proposed in Section 3.2. At first, we should combine the bounds of the intervals and sort them. Suppose the sorted non-decreasing list of $\{a_1, b_1, \ldots, a_N, b_N\}$ is $\{c_1, \ldots, c_{2N}\}$. Since any value in the interval $[c_i, c_{i+1})$ $(i = 1, \ldots, 2n-1)$ divides the whole data set into the same result, we can consider a partition point set T_V which contains all the possible splitting values. There are $2N - 1$ elements in the partition point set T_V for the interval-valued attribute V, which is shown in Equation 3.39:

$$T_V = \left\{ \frac{c^i + c^{i+1}}{2} | 1 \leq i \leq 2N - 1 \right\}. \tag{3.39}$$

Then, as in the classical case, the optimal segmentation point is selected to divide the sample set. The optimization rule is to maximize the information gain given by Equation 2.31. If the current partition attribute is interval-valued, then it can also be used as the partition attribute of its descendant node as in the classical case.

3.3.3 Histogram Explanatory Variable

Suppose we have an explanatory variable, X_u taking values in the following form:

$$\xi(\omega_u) = \{[a_k, b_k), p_{uk}; k = 1, \ldots, s\}, \quad u = 1, \ldots, n. \tag{3.40}$$

Here, we assume $s_u = s$ is the same for all categories; otherwise, we can reorganize the sub-intervals $[a_k, b_k)$. Since the sub-intervals for each realization can be the same and fixed by reorganizing the sub-intervals, the most important information from the realizations is the probability set $\{p_{uk}; k = 1, \ldots, s\}$ for $u = 1, \ldots, n$, where n is the number of aggregated groups in the data. Therefore, a straightforward method is to treat the probability p_{uk} corresponding to each sub-interval $[a_k, b_k)$ as a numerical variable with a value between 0 and 1. In this way, we can convert a histogram-valued variable into several numerical variables.

It is not easy to divide the data set into several groups by a histogram explanatory variable since the histogram variable contains lots of information, which makes it hard to find a rule to split the data set. Another way to deal with the histogram variable is to use cluster methods. Although it is impossible to exhaust all possible values for the histograms and regard them as categories, we can convert the histogram data into several clusters according to cluster methodology and treat the clusters as divided category attributes. There are some definitions of dissimilarity and cluster methods in Section 2.1.4.

3.4 Methodologies for Different Scenarios

In this section, several scenarios will be introduced with details and examples. First, all possible single types of explanatory variables will be stated and expanded to the situation with mixed explanatory variables.

Scenario 1 (Multi-valued/modal multi-valued explanatory variable and categorical response variable):

There are many questionnaires in life, for which most questions are often multiple-choice questions. For example, a game company will investigate which type of game users like. Each user may like more than one type of game. Therefore, the result collected for this problem is a list of game types rather than one single option. The first scenario is that a game company designs a set of questionnaires for a new advertisement of a certain game. The company would like to fit a classification model based on each user's feedback on the game (whether or not a user is willing to download the game), to predict the main audience of this game to target advertising better. For this study, the input (explanatory variables) are the answers to each user's questionnaire, and the output (response variable) is a binary categorical variable, which is or is not downloaded. To analyze the data in this situation, the steps are:

Step 1: Transfer all the multi-valued explanatory variables to modal multi-valued type as described in Section 3.1.

Step 2: The sample group can be divided into several subsets as described in Section 3.3.1. Consider all explanatory variables and select an explanatory variable that maximizes the difference of gain ratio of Equation 2.32.

Step 3: Repeat Step 2 until the group cannot be divided anymore.

What is more, if the feedback of the users is a rating score that is numerical, then Step 2 will be changed to the difference of the MSE defined in Equation 2.41.

Scenario 2 (Multi-valued/modal multi-valued explanatory variable and multi-valued/modal multi-valued response variable):

The game company can be also used as an example for this scenario. The game company designed a set of questionnaires for recommending suitable games to new users, to predict the games that each new user might like and recommend to the user accordingly. In this study, the input (explanatory variable) is the answer to each user's questionnaire, and the output (response variable) is the appropriate game name. Since multiple games can be recommended here, the output is also a multi-valued type. To analyze the data, in this case, the following the steps below are:

Step 1: Transfer all the multi-valued explanatory variable to modal multi-valued type as described in Section 3.1.

Step 2: The sample group can be divided into several subsets as described in Section 3.3.1. Consider all explanatory variables and select an explanatory variable that maximizes the difference of gain ratio in Equation 2.32.

Step 3: Repeat Step 2 until the group cannot be divided anymore.

Scenario 3 (Interval-valued explanatory and categorical response variable):

When applying for a job, a common question in the questionnaire is what is your expected salary. At this time, we are asked to provide a range of values instead of a fixed value. A website that helps job seekers find a job may need to collect the information, such as expected salary, expected working hours per week, etc. Based on the corresponding information of a user, we can decide whether or not to recommend a certain job to the user. The company would like to fit a classification model based on each user's feedback on the job (whether or not he/she is willing to apply for the job). For this study, the input (explanatory variables) are the answers to each user's questionnaire (expected salary, expected working hours per week, etc), and the output (response variable) is a binary categorical variable, corresponding to whether or not they are willing to apply. To analyze the data in this situation, we can use

the following steps:

Step 1: The sample group can be divided into several subsets as described in Section 3.3.2.

Step 2: Consider all explanatory variables and select an explanatory variable that maximizes the difference of gain ratio in Equation 2.32.

Step 3: Repeat Step 2 until the group cannot be divided anymore.

What is more, if the feedback of the users is a numerical rating score, then Step 2 will be changed to the difference of the MSE defined in Equation 2.41. If we collect the response variable with other types of data, the splitting measures can be changed accordingly.

Scenario 4 (Histogram-valued explanatory and categorical response variable):

When studying the relevant information of a certain area, such as rainfall, the rainfall of that day is not very representative if only a certain day is considered due to randomness. However, when the time window is set at one week or one month, using the mean or median to estimate the overall rainfall will lose a lot of information; so histogram-valued data should be considered. If we would like to classify areas by the rainfall situation during one month, the input becomes histogram-valued data and the output is a binary categorical variable. To analyze the data in this situation, the following steps are needed:

Step 1: The sample group can be divided into several subsets as described in Section 3.2.3.

Step 2: Consider all explanatory variables and select an explanatory variable that maximizes the difference of gain ratio in Equation 2.32.

Step 3: Repeat Step 2 until the group cannot be divided anymore.

Similarly, if the output is numerical or other types of symbolic data, the splitting measures can be changed accordingly.

Scenario 5 (Mixed-type explanatory variables and categorical response variable):

After discussing some of the above special cases, we note that only one type of explanatory variable is considered. Below, we discuss how to analyze mixed types of explanatory variables. In this case, the steps for mixed-type variables are:

Step 1: The sample group can be divided into several subsets as described in Section 3.3. If the explanatory variable is multi-valued or modal multi-valued type, then we can use methods discussed in Section 3.3.1. If interval-valued type is considered, then methods of Section 3.3.2 should be used. The sample group can be divided into several subsets as described in Section 3.2.3 if a histogram-valued explanatory variable is considered.

Step 2: Consider all explanatory variables and select an explanatory variable that maximizes the difference of gain ratio in Equation 2.32.

Step 3: Repeat Step 2 until the group cannot be divided anymore.

Similarly, if the output is numerical or other types of symbolic data, the splitting measures can be changed accordingly.

Chapter 4

Real Data Examples and Simulations

In this chapter, two real-data examples and several simulations will be used to check the performance of the methodologies developed in Chapter 3. Section 4.1 used a real-life data set to verify the application of our CART methodology for histogram-valued explanatory variables and a binary response variable. Section 4.2 used another real-life data set to verify the application of our CART methodology for interval-valued explanatory variables and a binary response variable. Section 4.3 generated some symbolic data and gave the results of simulations. Here only histogram-valued and interval-valued type of data is collected in real life. As for other types of symbolic data, simulation methods will be used to generate data.

4.1 Charging Data Example

Nowadays, electric transportation is becoming more and more popular. For example, many cities such as Boston have lots of shared bicycles to be used anywhere. Shared bikes provide convenience for people in life. In addition, electric cars are also popular because of their sustainability and environmental protection. However, for this kind of electric vehicle, one

problem is that the battery is consumable and needs to be repaired by professionals after use. In addition, different devices have different usage times and driving conditions, making it difficult to determine an exact inspection period. What is more, every time the battery is taken out of the device to check the battery condition, it is usually very labor-intensive and time-consuming. Energy researchers hope to classify the situation based on the data obtained during the charging process to predict the battery states, either normal or abnormal. This method not only reduces the number of times to take out the battery for quality inspection but also provides an early warning of an abnormal battery immediately. After all, the frequency of charging is much higher than taking the battery out for maintenance for these electric vehicles. Therefore, it is easier to find battery abnormalities as soon as possible by the charging data, avoiding many dangerous situations. This method reduces the possibility of danger and saves material resources and time.

4.1.1 Data Description

The data are collected from multiple bicycles. For each bike, multiple charging processes are recorded. The output (or the response variable) is the situation of the battery, either normal or abnormal. As for the input (or the explanatory variables), there are five features recorded during charging. The five features are explained in Table 4.1.

Table 4.1: Description of bicycle charging data set (Classical)

No.	Variable	Type	Description
V1	Voltage	numeric	Voltage when charging
V2	Current	numeric	Current when charging
V3	State of charge (SOC)	numeric	State of charge (0-100%)
V4	max_temp	numeric	The maximum cell temperature at the moment
V5	min_temp	numeric	The minimum cell temperature at the moment

Table 4.2 shows how the observations look. Assume there are N observations in total. Suppose that the n^{th} charging process $n = 1, \ldots, N$, is from time 0 to t_{q_n}; then, the time

length is t_{q_n}, where q_n is the number of records collected for the n^{th} charging process. As a result, the n^{th} charging process is a $q_n \times J$ matrix, where $J = 5$ is the number of explanatory variables described in Table 4.1. Generally speaking, most of the data we collect in life is in the form of a table, where each row represents a sample, and each column represents an explanatory variable. For each sample, the features we collected are the same. Therefore, the feature dimensions of this type of data are consistent. In other words, for general classical data, each sample is a row vector, and each number or category represents the value of the corresponding feature. However, each charging process is a matrix, and the dimensions are not consistent in this charging data.

Table 4.2: The n^{th} sample of the bicycle charging data set (Classical)

Record.	Voltage	Current	max_temp	min_temp	SOC
t_1	v_1	c_1	$\max_1$	$\min_1$	soc_1
t_2	v_2	c_2	$\max_2$	$\min_2$	soc_2
t_3	v_3	c_3	$\max_3$	$\min_3$	soc_3
...	...	...	...	...	...
t_{q_n}	v_{q_n}	c_{q_n}	$\max_{q_n}$	$\min_{q_n}$	soc_{q_n}

Figure 4.1 shows the data collected for one charging process. Each charging data value does not only record the current, voltage, and other variables at a fixed moment but rather values are recorded across time, i.e., for several moments. Suppose the n^{th} observation records all multivariate input during a period from time t_1 to t_{q_n}; here, we call the period as the time series length. Suppose we have J input variables ($J = 5$ in this case), and the time series length is the $t_{q_n} - t_1$ with q_n as the number of records. For any n, $n = 1, \ldots, N$, the curve of X^j consists of all the values for the j^{th}, $j = 1, \ldots, J$, explanatory variable during the charging process. In other words, the observations for this example have the form of a matrix with dimension $q_n \times J$ rather than just a row vector. What makes the data even harder to analyze is that the dimension of the matrices for different observations is not fixed.

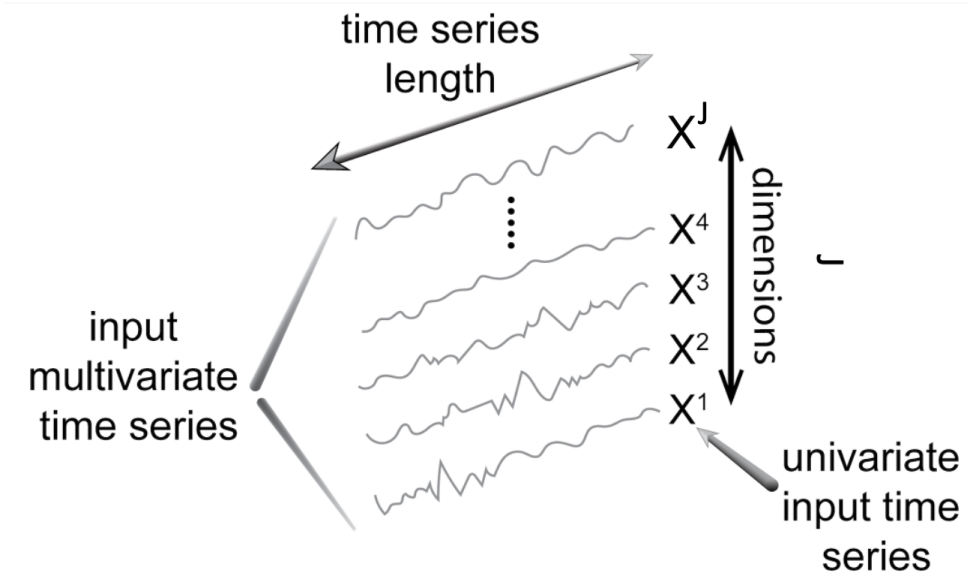

Figure 4.1: Description of one charging process of shared bicycles.

A matrix-type of data is not rare; we have lots of data in our daily life with a matrix type of observations. Take an image recognition problem as an instance. The inputs in the image recognition problem are matrices recording the information of the pixels. To analyze this image recognition data, we will first transfer all the images to matrices with the same size, $(N \times J)$-dimensional matrix, where J is the number of explanatory variables and N is the number of classical observations. Then we will convert the matrices into a $1 \times (N \times J)$ row vector when processing the data with each sample as a matrix. However, compared to the image detection example, the charging process example is much more complex. The fact that different charging processes have different charging times makes it difficult to analyze the data by traditional methods because of the inconsistency of the data matrix dimension. That is, the row number q_n of each charging process is not fixed. For example, suppose there are two charging process observations collected, and that they are called charging A and charging B. Suppose the charging time of charging A is long and the charging is complete, with the state of charge (SOC) from 20% to 100%, including 200 records of current and voltage data; and suppose charging B changes from 50% to 60%, containing 50 records. Therefore, charging A

contains far more information than charging B. Such a non-fixed data dimension makes the traditional classification model not suitable for bicycle charging data. In addition, another reason why traditional classification methods are not suitable for this data set is that even though there are a lot of data recorded, the number of basic features is only five. This small number of features tends to cause an underfitting problem.

4.1.2 Methods

There are several possible solutions to deal with an inconsistent dimension for the data. In this section, three possible methods with figures are listed in detail.

Method 1: Basic Statistics

Suppose we have J input variables ($J = 5$ in this case) and N observations, the time series length is q_n records for the $n^{th}, n = 1, \ldots, N$, charging process; then, the n^{th} charging process observation is an $q_n \times J$ matrix. Although the time length q_n is not fixed for different $n, n = 1, \ldots, N$, several basic statistics can be used to represent the whole process. For instance, three basic statistics, mean, median, and variance are chosen to summarize the process. Then each sample can be transformed to a $S \times M$ matrix, where S is the number of statistics used in the method. Since the three basic statistics, mean, median, and variance are chosen to summarize the process, $S = 3$ in this example. Figure 4.2 intuitively illustrates the way of data transformation by basic statistics for one charging process, where J is the number of input variables.

Table 4.3 shows how the new sample looks by Method 1 using basic statistics. Take the explanatory variable "Current" as an example. Since the time series length is q_n records for the n^{th} charging process, we can collect q_n values of "Current" during charging. Let us denote the values as $c_1, \ldots, c_{q_n}$. We can calculate the mean of current by $\bar{c}_n = \frac{1}{q_n} \sum_{m=1}^{q_n} c_k$ for the n^{th} observation. Similarly, we can find the median$(c)_n$ and var$(c)_n$ for the n^{th} observation.

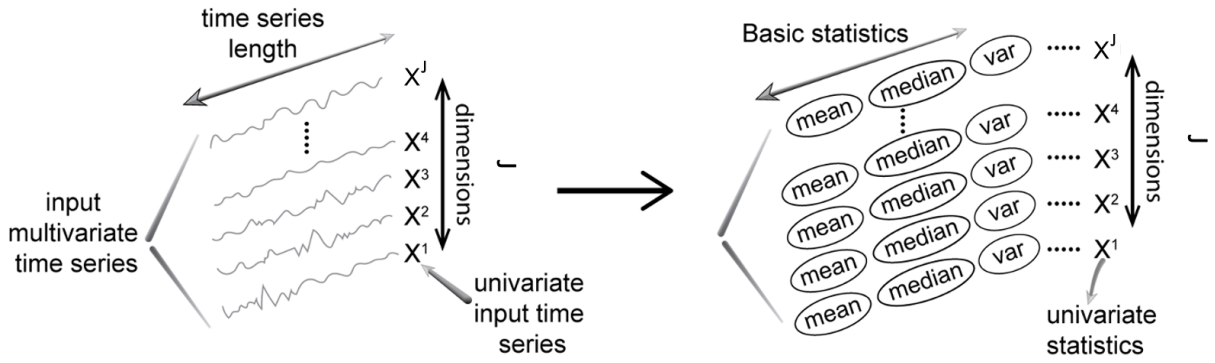

Figure 4.2: Method 1 - Reducing the dimension by statistics.

Compared with the original with its $q_n \times 5$ dimension, the new format has a fixed dimension 3×5, making it possible to transform the matrices to vectors with the same dimension, 1×15.

Table 4.3: The n^{th} sample of the bicycle charging data set by Basic Statistics Method

Statistics.	Voltage	Current	max_temp	min_temp	SOC
mean	$\bar{v}_n$	$\bar{c}_n$	$\bar{\text{max}}$	$\bar{\text{min}}$	$\bar{\text{soc}}$
median	$\text{median}(v)_n$	$\text{median}(c)_n$	$\text{median}(\text{max})_n$	$\text{median}(\text{min})_n$	$\text{median}(\text{soc})_n$
variance	$\text{var}(v)_n$	$\text{var}(c)_n$	$\text{var}(\text{max})_n$	$\text{var}(\text{min})_n$	$\text{var}(\text{soc})_n$

However, one drawback of this method is that we delete too many records which contain a lot of information. In this way, only a few statistics are selected to summarize the whole process very crudely, losing a lot of information. As a result, Method 1 will greatly increase the predicted bias.

Method 2: Filling by Time Series Model

Another method is to use time series models to predict the rest of the charging data, making all the observations have the same data dimension. Assume that the time length required to charge the battery from zero to full is T. For all of the observations where the charging process has a length smaller than T, we can predict the rest of the values to time T from

the previous time-series data based on the information collected. Time series models such as Auto Regressive Integrated Moving Average Model (ARIMA) and Recurrent Neural Networks (RNN) can be fitted to the actual values recorded. For example, if one charging process collected is from 20% to 80%, we can predict the charging data from 0% to 20% and 80% to 100% by the charging information collected (all the values from 20% to 80%). Figure 4.3 illustrates this way of data transformation by completing data where the filled-in data are shown in red curves.

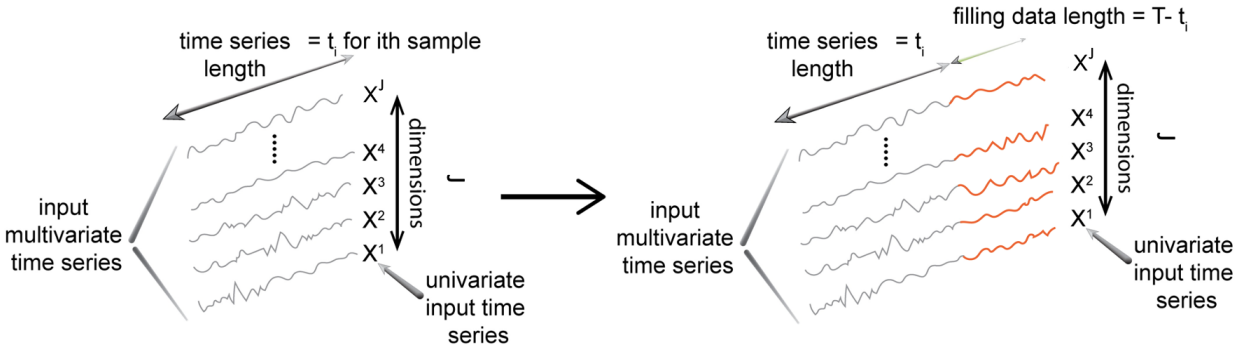

Figure 4.3: Method 2 - Filling data by a time series model.

However, in this case, we have to fill in all the variables for all the observations. This action will increase the number of calculations, and there will be an extra error when filling the data. Especially when the period of charging process we collected is not large enough, the increased calculation and errors can be very problematic.

Method 3: Histogram-valued Explanatory Variables

To analyze data with inconsistent feature dimensions, Method 3 first converts the data matrix into a symbolic type. Since all the six explanatory variables are numeric, we can obtain either interval or histogram type symbolic data. The interval data only contain the maximum and minimum values without considering the distribution of the data within the interval. For these data with such a small number of features, a lot of information will be missing if an

interval type is considered. As a result, histogram data will be used for the following analysis. This method is very similar to Method 1 which considers basic statistics, while Method 3 uses the histogram-valued variable to summarize the time series data, capturing more information from the original data.

There are two ways to convert the explanatory variables to histogram-valued. Take variable "Current" as an example; for each sample, we can collect lots of values. The first method is to convert directly. Suppose we have n charging processes in total. Then the n^{th} histogram-valued realization has the following formula:

$$I_n^{(1)} = \{[I_{ns}^{(1)}, I_{ns}^{(2)}), p_{ns}; s = 1, \ldots, S_n\}, \quad n = 1, \ldots, N, \tag{4.1}$$

where S_n is the finite number of sub-intervals forming the support for the outcome q_n for observation n with weight p_{ns}, $s = 1, ..., S_i$, and $\sum_{s=1}^{s_n} p_{ns} = 1$. The $[I_{ns}^{(1)}, I_{ns}^{(2)})$ are the sub-intervals we defined. In addition, the notation (1) in $I_n^{(1)}$ refers to the first way to convert to the histogram-valued data. However, one disadvantage of this method is that we directly capture all the recorded current values together and ignore the current fluctuation process. That means we fail to consider the order of recording. For the same set of values, different orders will result in different results.

The second method is to use the state of charge (SOC) to split the charging processes into several sub-charging processes. For example, we can consider ten sub-charging processes, $0\% - 10\%, 10\% - 20\%, \cdots, 90\% - 100\%$ and one explanatory variable "Current". For each sub-process, we use the mean of values for the explanatory variables to represent the whole sub-process. Suppose we have N charging processes in total. Then the n^{th} realization has the following formula:

$$I_n^{(2)} = \{[soc_{nl}^{(1)}, soc_{nl}^{(2)}), \bar{I}_{nl}; l = 1, ..., L\}, \quad n = 1, \ldots, N, \tag{4.2}$$

where L is the finite number of sub-processes for all the observations, and $\bar{I}_{nl}$ is the mean value of current in the l^{th} sub-process for the n^{th} observation. In addition, the notation (2) in $I_n^{(2)}$ refers to the second way to convert to the histogram-valued data. The symbolic realization obtained by this conversion method is somewhat different from the previous definition. For the histogram-valued data defined in Definition 2.1.6, we add the frequency of the corresponding sub-interval to each sub-interval as a weight. As a result, all the weights should sum up to 1. Compared with splitting the intervals into several sub-intervals, we cut the whole charging process into several sub-processes. Here, we use the mean of the current to summarize the values within the sub-process. Similarly with the splitting methods for histogram-valued data introduced in Section 3.3.3, we use a set of continuous values $\{\bar{I}_{n1}, \ldots, \bar{I}_{nL}\}$ to represent the n^{th} histogram-valued data.

Figure 4.4 shows the way of data transformation by histogram-valued method. Suppose we have J explanatory variables and N observations. For each $j, j = 1, \ldots, J$, and $n, n = 1, \ldots, N$, the n^{th} observation has a series of values X_n^j for the j^{th} explanatory variable. For any j, $j = 1, \ldots, J$, $X^{(j)}$ is the histogram-valued variable transformed by the original series X^j. By using symbolic type of data, each charging data set becomes one row with a histogram-valued element for each explanatory variable. Different ranges of SOC are indicated by different colors in the figure.

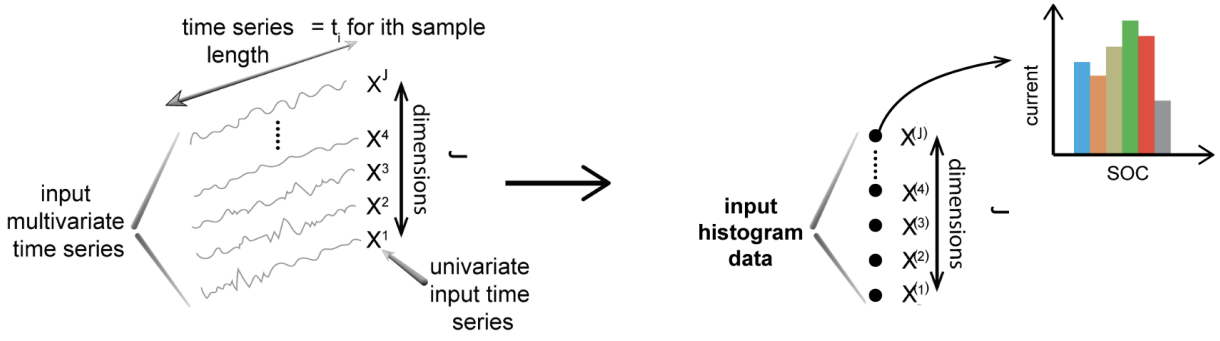

Figure 4.4: Method 3 - Histogram-valued explanatory variables.

4.1.3 Results

Since the output of this data set is a binary categorical variable, it belongs to a binary classification problem. From Section 2.2.4, there are two metrics to measure the performance of each model, recall, and precision. The metric accuracy is not appropriate because of the imbalanced classes. Only 10% of observations we collected are abnormal. Here, we are more focused on the abnormal batteries. As a result, we assume the abnormal charging process as a positive case. After defining the positive case, we can calculate the precision and recall by Equation 2.43 and Equation 2.44 in Definition 2.2.9. Type I error is defined as wrongly predicting a normal battery as abnormal, and Type II error is erroneously predicted an abnormal battery as normal. Both of the errors will cause great losses to us. Type I error will increase our subsequent workload and consume a lot of time and material resources to repair the normal battery. However, it will be very dangerous if we have a Type II error. The result of large precision is to reduce Type I error, and the result of large recall is to reduce Type II error. Recall and precision are both trade-offs; we give priority to the model with high recall because the consequences of the two errors are quite different. Table 4.4 shows the result of classification for the charging data set by the three methods.

Table 4.4: Comparison of different methods for charging data set

Method.	Recall (training)	Precision (training)	Recall (testing)	Precision (testing)
1	1	0.77	0.69	0.31
2	1	0.92	0.2	0.5
3	1	0.86	1	0.45

As can be seen from Table 4.4, the prediction of the three methods on the training set is good enough, especially Method 2. To compare the prediction of the model for future data without a known output, either normal or abnormal, we should mainly compare the performance on the testing set. Here, collecting a low recall rate for Method 1 means that we have mistakenly predicted too much abnormal charging, which has the consequence that

some dangerous vehicles are not discovered. A low precision rate represents that we predict many normal charging processes incorrectly. The result will increase the subsequent series of further inspections and the time to check the vehicles. In addition, it will consume more material resources. In contrast, the harm of a low recall rate is greater. As a result, we should choose a model with a larger recall when attaining a proper value of precision. From Table 4.4, we can see that performance on the testing set by Method 3 is significantly better than the other two methods. Method 3 can capture a high precision value while keeping recall at 1. It shows that there might be some normal charging process to be predicted as abnormal incorrectly based on the precision. However, we can give an early warning of all the abnormal charging processes. As a result, the prediction of Method 3 for future data without a known output is much better than the other two methods.

4.2 Iris Data Example

4.2.1 Data Description

We apply our proposed CART for symbolic data method to the Iris data set. The Iris data were first collected by Anderson, 1935, to determine the geographic variation of Iris flowers. Fisher, 1936, used the data set as an application of discriminant analysis, making the Iris data well-known and popular in data analysis. The Iris data set contains measurements of 50 Iris flowers each from three species: setosa, versicolor, and virginica. For each observation, four numerical explanatory variables, V_1 = sepal length, V_2 = sepal width, V_3 = petal length and V_4 = petal width were measured. All of the four features and the categorical response variable are explained in Table 4.5.

To analyze a classical data set like this, one important step is to visualize the data to obtain some intuitive conclusion. Figure 4.5 shows a scatter plot of all attribute pairs with the points of different species taking different colors (blue if the iris is setosa, pink if the

Table 4.5: Description of Iris data set (Classical)

No.	Variable	Type	Description
V_1	sepal length	numerical	in cm
V_2	sepal width	numerical	in cm
V_3	petal length	numerical	in cm
V_4	petal width	numerical	in cm
Y	Species	categorical	setosa, versicolor, virginica

iris is versicolor, and green if the iris is virginica). In addition, since the scatter plot shows that the points of each species are well-separated, ellipses are drawn around them. We can see that there are some clear relationships between input attributes (trends) and between attributes and species of the iris.

There are 150 classical observations in the original iris data set. Billard and Diday, 2006, assume that every five consecutive flowers listed in the original data set come from the same location, e.g., a nursery. In addition, suppose we are interested in the characteristics of species by location rather than the characteristics of individual flowers. Then, each group of five classical observations can be summarized into an interval value observation. As a result, the interval-valued data set after aggregation includes ten observations for setosa species, ten observations for variegated species, and ten observations for virginica species. After aggregating the data, the four numerical variables V_1, V_2, V_3, and V_4 become interval-valued type. The four interval-valued explanatory variables are $V_{(1)}$ = sepal length, $V_{(2)}$ = sepal width, $V_{(3)}$ = petal length and $V_{(4)}$ = petal width, where $V_{(i)}$ is the interval-valued variable for the numerical variable V_i for $i = 1, 2, 3, 4$. Table 4.6 shows the interval-valued Iris data.

Similar to the steps of analyzing classical data, we can obtain some preliminary intuitive conclusions through some visualization methods. Figure 4.6 shows the scatter plot of interval type iris data set of Table 4.6. Since there are four interval-valued type of explanatory variables in the data set, in order to see the distribution of the data intuitively, we select

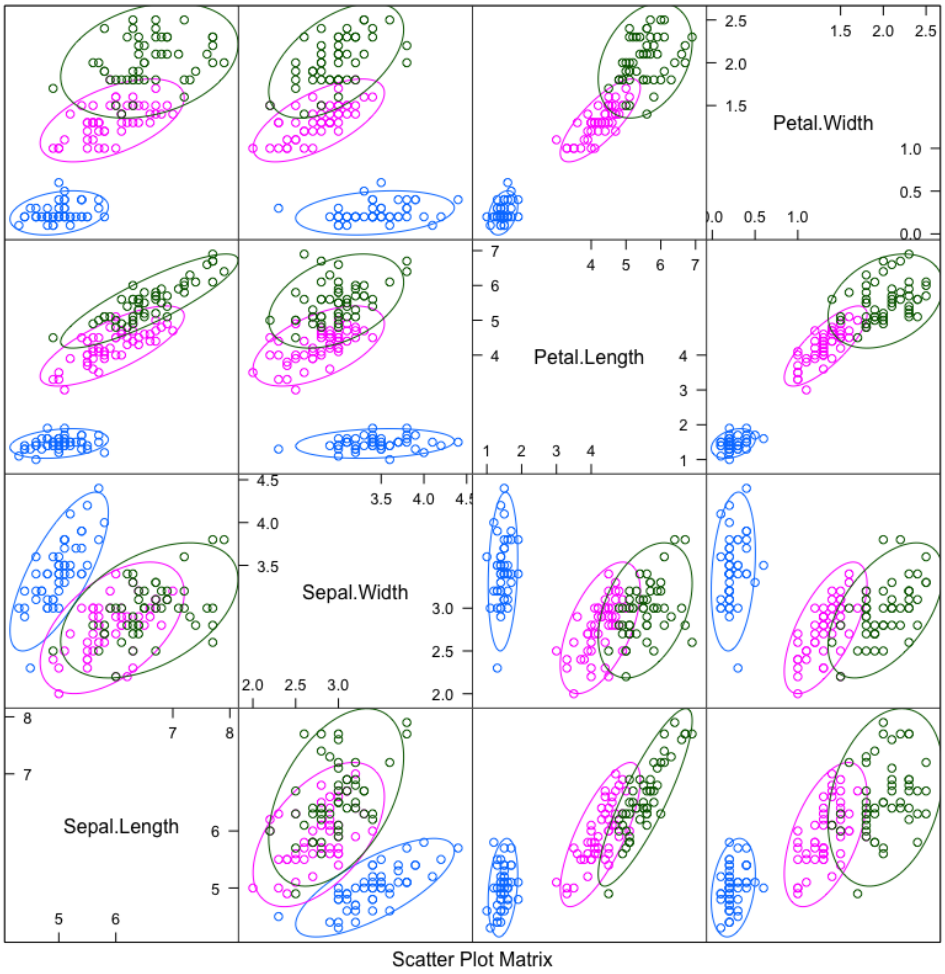

Figure 4.5: The scatter plot of original Iris data set.

two variables as the axis each time to capture six scatter plots. In addition, we still use the different colors to distinguish the different species. From the Figure 4.6, we can see that through these four interval-valued variables, we can distinguish irises with different species, especially from the scatter plot with petal length and petal width in Figure 4.6. This conclusion is consistent with the finding from Figure 4.5. It is easy to show that although we have lost some information when aggregating the data, interval-valued data greatly reduce the number of observations while retaining the original distribution characteristics.

Table 4.6: Interval-valued Iris data set

No.	Species	Sepal Length	Sepal Width	Petal Length	Petal Width
1	S1	[4.6, 5.1]	[3.0, 3.6]	[1.3, 1.5]	[0.2, 0.2]
2	S2	[4.4, 5.4]	[2.9, 3.9]	[1.4, 1.7]	[0.1, 0.4]
3	S3	[4.3, 5.8]	[3.0, 4.0]	[1.1, 1.6]	[0.1, 0.2]
4	S4	[5.1, 5.7]	[3.5, 4.4]	[1.3, 1.7]	[0.3, 0.4]
5	S5	[4.6, 5.4]	[3.3, 3.7]	[1.0, 1.9]	[0.2, 0.5]
6	S6	[4.7, 5.2]	[3.0, 3.5]	[1.4, 1.6]	[0.2, 0.4]
7	S7	[4.8, 5.5]	[3.1, 4.2]	[1.4, 1.6]	[0.1, 0.4]
8	S8	[4.4, 5.5]	[3.0, 3.5]	[1.2, 1.5]	[0.1, 0.2]
9	S9	[4.4, 5.1]	[2.3, 3.8]	[1.3, 1.9]	[0.2, 0.6]
10	S10	[4.6, 5.3]	[3.0, 3.8]	[1.4, 1.6]	[0.2, 0.3]
11	Ve1	[5.5, 7.0]	[2.3, 3.2]	[4.0, 4.9]	[1.3, 1.5]
12	Ve2	[4.9, 6.6]	[2.4, 3.3]	[3.3, 4.7]	[1.0, 1.6]
13	Ve3	[5.0, 6.1]	[2.0, 3.0]	[3.5, 4.7]	[1.0, 1.5]
14	Ve4	[5.6, 6.7]	[2.2, 3.1]	[3.9, 4.5]	[1.0, 1.5]
15	Ve5	[5.9, 6.4]	[2.5, 3.2]	[4.0, 4.9]	[1.2, 1.8]
16	Ve6	[5.7, 6.8]	[2.6, 3.0]	[3.5, 5.0]	[1.0, 1.7]
17	Ve7	[5.4, 6.0]	[2.4, 3.0]	[3.7, 5.1]	[1.0, 1.6]
18	Ve8	[5.5, 6.7]	[2.3, 3.4]	[4.0, 4.7]	[1.3, 1.6]
19	Ve9	[5.0, 6.1]	[2.3, 3.0]	[3.3, 4.6]	[1.0, 1.4]
20	Ve10	[5.1, 6.2]	[2.5, 3.0]	[3.0, 4.3]	[1.1, 1.3]
21	Vi1	[5.8, 7.1]	[2.7, 3.3]	[5.1, 6.0]	[1.8, 2.5]
22	Vi2	[4.9, 7.6]	[2.5, 3.6]	[4.5, 6.6]	[1.7, 2.5]
23	Vi3	[5.7, 6.8]	[2.5, 3.2]	[5.0, 5.5]	[1.9, 2.4]
24	Vi4	[6.0, 7.7]	[2.2, 3.8]	[5.0, 6.9]	[1.5, 2.3]
25	Vi5	[5.6, 7.7]	[2.7, 3.3]	[4.9, 6.7]	[1.8, 2.3]
26	Vi6	[6.1, 7.2]	[2.8, 3.2]	[4.8, 6.0]	[1.6, 2.1]
27	Vi7	[6.1, 7.9]	[2.6, 3.8]	[5.1, 6.4]	[1.4, 2.2]
28	Vi8	[6.0, 7.7]	[3.0, 3.4]	[4.8, 6.1]	[1.8, 2.4]
29	Vi9	[5.8, 6.9]	[2.7, 3.3]	[5.1, 5.9]	[1.9, 2.5]
30	Vi10	[5.0, 6.7]	[2.5, 3.4]	[5.0, 5.4]	[1.8, 2.3]

4.2.2 Results

There are four interval-valued explanatory variables $V_{(1)}$, $V_{(2)}$, $V_{(3)}$ and $V_{(4)}$, and one categorical response variable which is species in this data set. Therefore, two splitting rules can be used to divide the data set, as explained in Section 3.3.2. The first method is to regard the maximum and minimum values of each independent interval-valued variable as two correlated numerical

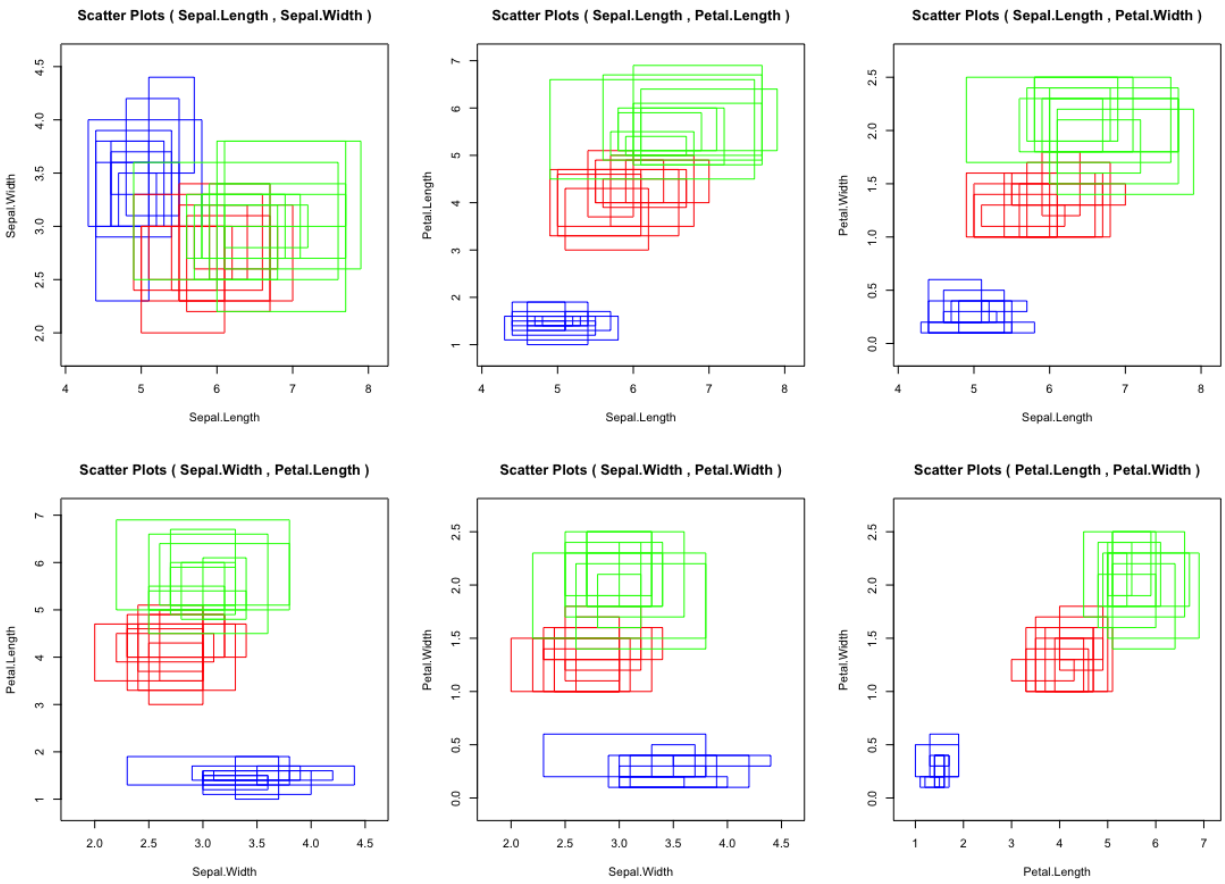

Figure 4.6: The scatter plot of interval-valued Iris data set.

variables. The other method is to divide the data set by interval-valued variables with the rule of three partitions in Section 3.3.2. Both methods will be used for this data set and the results will be compared. The 30 symbolic observations will be divided into two subsets, a training set with 24 observations to build the tree model, and the remaining 6 observations will be used as the testing set to check the prediction of the models. Figure 4.7 shows the decision trees on the training set by these two methods.

Both of these methods provide a good prediction on the testing set. All observations are correctly predicted by these two methods. The key point in these two methods is to find the "best" feature and the corresponding value compared with the two edges of the interval. However, the two sides of the interval are considered separately in the first method, while the

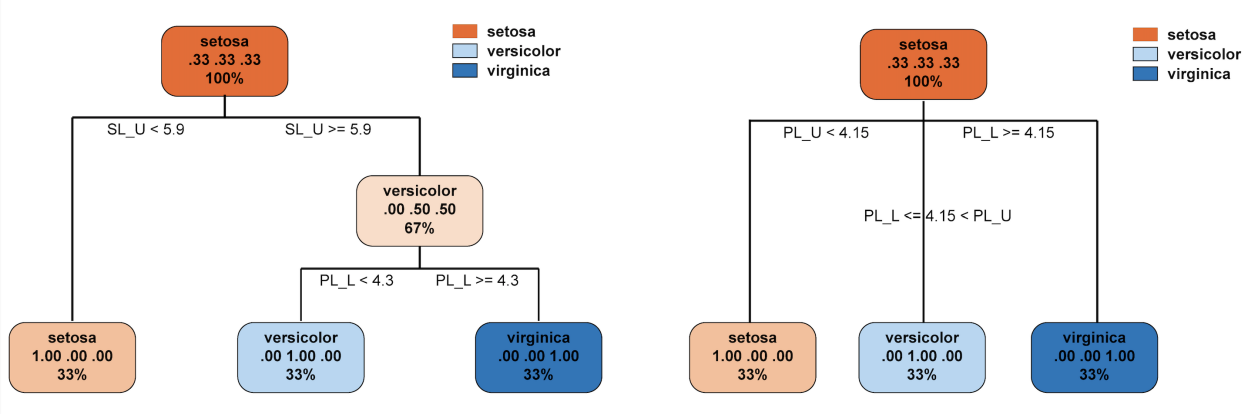

Figure 4.7: The CART for interval-typed Iris data: the left plot shows the tree structure by the bi-partition for interval-valued explanatory variables, and the right one shows the tree structure by the triple-partition method.

two sides are compared at the same time when using the second method. Even though results of these two methods show our CART for symbolic data method has an ideal performance when predicting binary classification problems, the sample size in the data set is too small to conclude the overall situation. Therefore, several raw data will be simulated to compare these methods with traditional methods in the rest of this chapter. In addition, different situations will be simulated to draw convincing conclusions.

4.3 Simulation Methods

In this section, we will introduce the method to generate the classical data with I groups and J explanatory variables, where K_i classical observations will be in the i^{th} group for $i = 1, \ldots, I$. After aggregating the classical observations in one group, we can obtain a symbolic data with I symbolic observations and J symbolic explanatory variables.

4.3.1 Multi-valued Data Generation

To obtain a data set with modal multi-valued or interval-valued variables, we should know how to generate categorical or numerical variables first. As a result, we should clarify how to generate single multi-valued and interval-valued data first.

To simplify the process, only multi-valued variables with binary possible values are considered as explanatory variables during the simulation. At first, it is necessary to clarify the method to simulate one binary variable. Suppose we would like to generate one modal multi-valued variable with I groups. For the i^{th} group, $i = 1, \ldots, I$, we will generate K_i classical values to obtain one group. The process to generate one modal multi-valued variable is listed in Process 1.

Process 1 : Generate one modal multi-valued variable

1: Generate I values from a uniform distribution U(0,1) as $p_1, \ldots, p_I$.

2: For each $i = 1, \ldots, I$, generate K_i data points x_{ik}, $k = 1, \ldots, K_i$ from a Bernoulli distribution with $p = p_i$.

3: For each $i = 1, \ldots, I$, aggregate the data points x_{ik}, $k = 1, \ldots, K_i$ to one group with modal multi-valued realization $\xi_i = \{1, q_i; 0, (1 - q_i)\}$, where $q_i = \frac{1}{K_i} \sum_{k=1}^{K_i} x_{ik}$.

From the first two steps, there will be I groups of values with K_i values in the i^{th} group. In addition, the values in each group are generated from the same distribution. Values from different groups are generated from the Bernoulli distribution but with different parameters. In this way, we can distinguish observations from different groups. As a result, there are $\sum_{i=1}^{I} K_i$ realizations in total to be generated in these two steps. As the last step in Process 1, we can obtain I multi-valued realizations by aggregating the data points in the different groups.

After introducing the method to simulate one modal multi-valued variable, we can construct a data frame with several multi-valued explanatory variables, as shown in Process 2.

Suppose we would like to generate J modal multi-valued variable with I groups. The K_i in Step 2 of Process 2 is the same for different variables since we assume the aggregating method is the same for different variables.

Process 2 : Generate several modal multi-valued variables

1: For the $j^{th}, j = 1, \ldots, J$, variable, generate I values from a uniform distribution U(0,1) as $p_{1j}, \ldots, p_{Ij}$.

2: For each $i = 1, \ldots, I$, generate K_i data points x_{ijk}, $k = 1, \ldots, K_i$, from a Bernoulli distribution with $p = p_{ij}$.

3: Repeat Step 1 and Step 2 J times to generate J binary explanatory variables.

4: For each $i = 1, \ldots, I$, groups and each $j = 1, \ldots, J$, explanatory variables, aggregate the data points x_{ijk}, $k = 1, \ldots, K_i$, to one group with J-dimensional modal multi-valued realization $\xi_{\mathbf{i}} = (\xi_{i1}, \ldots, \xi_{iJ})'$ with $\xi_{ij} = \{1, q_{ij}; 0, (1 - q_{ij})\}$, where $q_{ij} = \frac{1}{K_i} \sum_{k=1}^{K_i} x_{ijk}$.

For a better understanding of Process 2, consider an example with $K_i = 5, i = 1, \ldots, I$. Table 4.7 shows part of the original data, where $x_{ijk}, i = 1, \ldots, I, j = 1, \ldots, J, k = 1, \ldots, K_i$, refers to the k^{th} realization in the i^{th} group for the j^{th} variable. Each x_{ijk} is Bernoulli distributed with parameter $p = p_{ij}$. In other words, x_{ijk} will be either 0 or 1. Since $K_i = 5, i = 1, \ldots, I$, in this case, there will be 5 classical observations in each group with I groups in total. There are $5 \times I$ rows and J columns in the table, each row represents a classical observation, and each column represents one categorical explanatory variable with two possible outputs.

There are I groups in total with $K = 5$ classical observations in each group. For each observation, there are J categorical variables collected. After aggregating the observations in the same group, there will be I groups of classical observations. For each group, there are J explanatory variables with 5 realizations for each variable. Table 4.8 shows the aggregated data.

To analyze the data with more than one realization for each observation, we can transfer the data type to a symbolic one. That means we can summarize all the information of one

Table 4.7: One example of a data frame with binary explanatory variables

Group	Variable 1	Variable 2	...	Variable J
Group 1	x_{111}	x_{121}	$\cdots$	x_{1J1}
Group 1	x_{112}	x_{122}	$\cdots$	x_{1J2}
Group 1	x_{113}	x_{123}	$\cdots$	x_{1J3}
Group 1	x_{114}	x_{124}	$\cdots$	x_{1J4}
Group 1	x_{115}	x_{125}	$\cdots$	x_{1J5}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Group I	x_{I11}	x_{I21}	$\cdots$	x_{IJ1}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Group I	x_{I15}	x_{I25}	$\cdots$	x_{IJ5}

Table 4.8: The aggregated data frame with binary explanatory variables

Group	Variable 1	Variable 2	...	Variable J
Group 1	x_{111},x_{121},x_{131},x_{141},x_{151}	x_{112},x_{122},x_{132},x_{142},x_{152}	$\cdots$	x_{11J},x_{12J},x_{13J},x_{14J},x_{15J}
Group 2	x_{211},x_{221},x_{231},x_{241},x_{251}	x_{212},x_{222},x_{232},x_{242},x_{252}	$\cdots$	x_{21J},x_{22J},x_{23J},x_{24J},x_{25J}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Group I	x_{I11},x_{I21},x_{I31},x_{I41},x_{I51}	x_{I12},x_{I22},x_{I32},x_{I42},x_{I52}	$\cdots$	x_{I1J},x_{I2J},x_{I3J},x_{I4J},x_{I5J}

cell to obtain the modal multi-valued data. The aggregated multi-valued data are shown in Table 4.9. There are J modal multi-valued explanatory variables in the table. The symbolic data frame is a $I \times J$ matrix with modal multi-valued type cells.

Table 4.9: Data frame with modal multi-valued explanatory variables

Group	Variable 1	Variable 2	...	Variable J
Group 1	$\xi_{11} = \{1, q_{11}; 0, (1 - q_{11})\}$	ξ_{12}	$\cdots$	$\xi_{1J} = \{1, q_{1J}; 0, (1 - q_{1J})\}$
Group 2	$\xi_{21} = \{1, q_{21}; 0, (1 - q_{21})\}$	ξ_{22}	$\cdots$	$\xi_{2J} = \{1, q_{2J}; 0, (1 - q_{2J})\}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Group I	$\xi_{I1} = \{1, q_{I1}; 0, (1 - q_{I1})\}$	ξ_{I2}	$\cdots$	$\xi_{IJ} = \{1, q_{IJ}; 0, (1 - q_{IJ})\}$

To simplify the process, we only consider multi-valued variables with binary possible values as explanatory variables during the simulation processes. If we want to generate multi-valued variables with more than two possible values, we can change the Bernoulli distribution to a Multinomial distribution in Process 1 and Process 2. In addition, we can also use some

other discrete distributions such as truncated Poisson and binomial distributions to simulate other situations.

4.3.2 Interval-valued Data Generation

To obtain a data set with an interval-valued explanatory variable and interval-valued response variable, we should generate the numerical variables first. Therefore, we first need to clarify the method to simulate one numerical variable. Suppose we would like to generate one interval-valued variable with I groups. For the i^{th} group, $i = 1, \ldots, I$, we will generate K_i classical values to obtain one group. The process to generate one interval-valued variable is described in Process 3.

Process 3 Generate one interval-valued variable

1: Generate I values from a normal distribution with mean $= \eta_1$, and variance $= \eta_2$; denote them as $\mu_1, \ldots, \mu_I$. The μ_i is generated for the mean value for the i^{th} group, $i = 1, \ldots, I$.

2: For each $i, i = 1, \ldots, I$, generate one value from a chi-square distribution with $df = K_i - 1$ as v_i, where K_i is the number of classical observations in the i^{th} group, and v_i is generated for the variance for the i^{th} group.

3: For each pair of (μ_i, v_i), $i = 1, \ldots, I$, generate K_i data points x_{ik}, $k = 1, \ldots, K_i$, from a normal distribution with mean $= \mu_i$, and variance $= v_i$.

4: For each $i = 1, \ldots, I$, aggregate K_i data points $x_{ik}, k = 1, \ldots, K_i$, to one group with interval-valued realization $\xi_i = [a_i, b_i]$ where $a_i = \min_k\{x_{ik}\}$ and $b_i = \max_k\{x_{ik}\}$.

From these four steps, we will have I groups of values with K_i in the i^{th} group. In addition, the values in each group are generated from the exact same normal distribution. Values from different groups are generated from a normal distribution but with different parameters. In this way, we can distinguish the observations from different groups. Similarly to the modal

multi-valued case, there will be $\sum_{i=1}^{I} K_i$ realizations in total which are generated as classical numeric variables. The values in one group are aggregated together in the last step.

After obtaining the method to simulate one interval-valued variable as in Process 3, several interval-valued variables can be generated as the explanatory variables for the simulated data. Suppose we would like to generate J interval-valued variable with I groups and K_i classical observations in the i^{th} group, $i = 1, \ldots, I$. Process 4 describes the steps to generate multiple interval-valued explanatory variables.

Process 4 : Generate multiple interval-valued variables

1: For the j^{th} variable, generate I values from a from a normal distribution with mean = η_{1j}, and variance = η_{2j}; denote them as $\mu_{1j}, \ldots, \mu_{Ij}$. Let μ_{ij} be the mean value for the i^{th} group and j^{th} explanatory variable.

2: For each $i, i = 1, \ldots, I$, and $j, j = 1, \ldots, J$, generate one value from a chi-square distribution with $df = K_i - 1$ as v_{ij}, where K_i is the number of classical observations in the i^{th} group. Let v_{ij} be the variance for the i^{th} group and j^{th} explanatory variable.

3: For each pair of (μ_{ij}, v_{ij}), $i = 1, \ldots, I$, generate K_i data points x_{ijk}, $k = 1, \ldots, K_i$ from a normal distribution with mean = μ_{ik}, and variance = v_{ik}.

4: Repeat step 1 - step 3 J times to generate J continuous explanatory variables.

5: For each $i = 1, \ldots, I$ groups and each $j = 1, \ldots, J$ explanatory variables, aggregate K_i classical data points x_{ijk}, $k = 1, \ldots, K_i$, to one group with J-dimensional interval-valued realizations $\xi_{\mathbf{i}} = (\xi_{i1}, \cdots, \xi_{iJ})'$ with $\xi_{ij} = [a_{ij}, b_{ij}]$, where $a_{ij} = \min_k\{x_{ijk}\}$ and $b_{ij} = \max_k\{x_{ijk}\}$.

Similarly to Process 2, we can obtain J continuous variables with $I \times K$ classical observations. By aggregating all the classical observations based on the groups and calculating the minimal and maximal of each group, we can capture a data frame with interval-valued explanatory variables, as shown in Table 4.10. Here for continuous variables, normal distributions are assumed because the normal distribution is the most common distribution

encountered in real-life applications. We can also assume other continuous distributions such as a chi-square distribution, or a Beta distribution to simulate different situations.

Table 4.10: One example of data frame with interval-valued explanatory variables

Group	Variable 1	Variable 2	...	Variable J
Group 1	$\xi_{11} = [a_{11}, b_{11}]$	$\xi_{12} = [a_{12}, b_{12}]$	$\cdots$	$\xi_{1p} = [a_{1J}, b_{1J}]$
Group 2	$\xi_{21} = [a_{21}, b_{21}]$	$\xi_{22} = [a_{22}, b_{22}]$	$\cdots$	$\xi_{2p} = [a_{2J}, b_{2J}]$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Group I	$\xi_{I1} = [a_{I1}, b_{I1}]$	$\xi_{I2} = [a_{I2}, b_{I2}]$	$\cdots$	$\xi_{IJ} = [a_{IJ}, b_{IJ}]$

4.4 Simulations and Results

In this section, several symbolic variables will be generated as explanatory variables, including modal multi-valued types and interval-valued types. For the response variables, categorical variables, multi-valued variables, and interval-valued variables will be considered. For all the situations in all the scenarios, we will first generate classical data with I groups and J explanatory variables, where K classical observations will be in each group. After aggregating the classical observations in one group, we can obtain a symbolic data set with I symbolic observations and J symbolic explanatory variables. For different situations, we will choose different values of I and J, while K is fixed as 10 for all the situations. We considered four different scenarios, which are listed in Table 4.11.

Table 4.11: Different scenarios for simulation.

	Explanatory variables	Response variable
Scenario 1	Multi-valued	Categorical
Scenario 2	Multi-valued	Multi-valued
Scenario 3	Interval-valued	Categorical
Scenario 4	Interval-valued	Interval-valued

For the first two scenarios, the only type of explanatory variable is multi-valued. Simulations with different numbers of explanatory variables, different numbers of groups, and

multiple situations of the distance between groups will be used to compare the performances. The situations are listed in Table 4.12.

Table 4.12: Different situations for data set with modal multi-valued explanatory variables.

	J	I	The distance between groups
Situation 1	4	100	Small
Situation 2	4	1000	Small
Situation 3	20	100	Small
Situation 4	4	100	Large

The J in Table 4.12 represents the number of explanatory variables, and I is the number of groups. Suppose the output is a categorical variable with two possible values, 1 for the first class and 2 for the second class. In addition, we assume all the explanatory variables are modal multi-valued variables with only two possible values, 0 and 1. That means the symbolic realization of the j^{th} explanatory variable for the i^{th} group is $\{1, p_{ij}; 0, (1 - p_{ij})\}$. Since only a binary multi-valued explanatory variable is considered here, the distance is the absolute value of the difference between the probability corresponding to the value 1, which is represented by p_{ij}. For instance, suppose we have a symbolic realization with the first explanatory variable as $s = \{1, 0.2; 0, 0.8\}$, and there are two other observations to be compared with the symbolic realization, with the first explanatory variable $s_1 = \{1, 0.3; 0, 0.7\}$ and $s_2 = \{1, 0.8; 0, 0.2\}$ for the two observations, respectively. It is obvious that the distance between s and s_1 calculated by $|0.2 - 0.3| = 0.1$ is much smaller than the distance between s and s_2 calculated by $|0.2 - 0.8| = 0.6$. Generally speaking, the larger the distance between different groups, the easier it is to distinguish groups and classify them.

By comparing Situation 1 and Situation 2, we can find how the performance of the CART produced for symbolic data with modal multi-valued explanatory variables will change when the number of groups changes. By comparing Situation 1 and Situation 3, the influence of the number of explanatory variables on the performance can be discovered. Based on the results

of Situation 1 and Situation 4, we can obtain a measure of the effect of distance between groups on the prediction of the classes.

For the last two scenarios of Table 4.11, the explanatory is interval-valued. Not only simulations with different numbers of explanatory variables and groups will be compared, but also simulations with different scales of explanatory variables will be generated. In addition, the Signal-to-Noise Ratio (SNR) and the correlation between explanatory variables should also be considered. For Situation 3, we assume the noise is equal to the signal (SNR = 1). In this case, the noise is very large compared to our signal. For Situation 1, which is also the base situation, the noise is a little bit larger than the signal (SNR = 1.5). Here, the SNR value of 1.5 is chosen because the assumption is that there exists an overlap between intervals under this situation. To ensure that there is an overlap between the intervals of two classes, we should not choose a value that is too large. For Situation 4, the noise is much smaller than the signal (SNR = 3). Why is SNR = 3 is a good value for this case? To ensure that the intervals in different classes can be well-separated, we should choose a value that is large enough. Another reason for choosing the SNR value as 3 is that this value is not large enough to completely separate the data of the two classes, and there is still a small amount of overlap. This also makes our simulated data closer to real-world data, and it is more necessary to fit the model. Otherwise, if the data from different classes have no intersection at all, then a very simple classification rule can be used to separate the two classes perfectly, and it is not so necessary to fit a complex model.

All of the seven different situations to be studied for the last two scenarios are presented in Table 4.13. The meanings of J and I in Table 4.13 are exactly the same as the definitions in Table 4.12, i.e., J represents the number of explanatory variables and I is the number of groups. By comparing Situation 1 and Situation 2, we can find out how the performance of CART for symbolic data with modal multi-value explanatory variables will change when the number of groups changes. By comparing Situation 1, Situation 3 and Situation 4, it can

be found how the SNR will affect performance. According to the results of Situation 1 and Situation 5, data with different numbers of explanatory variables are compared. Based on the results of Situation 1 and Situation 6, different scales are compared. Based on the results of Situation 1 and Situation 7, we can obtain the influence by whether or not there exists a correlation between explanatory variables.

Table 4.13: Different situations for data set with interval-valued explanatory variables.

	J	I	SNR	Scale of explanatory variables	Correlation
Situation 1	4	100	1.5	Same	Independent
Situation 2	4	1000	1.5	Same	Independent
Situation 3	4	100	1	Same	Independent
Situation 4	4	100	3	Same	Independent
Situation 5	20	100	1.5	Same	Independent
Situation 6	4	100	1.5	Different	Independent
Situation 7	4	100	1.5	Same	Correlated

4.4.1 Scenario 1

Suppose we would like to generate a data frame with J modal multi-valued explanatory variables and I groups of data with K classical observations in each group. To simulate this data set, $I \times K$ classical observations are randomly drawn from the same distribution. For the k^{th} observation in the i^{th} group, the j^{th} explanatory variable is distributed with a Bernoulli distribution which is given in Equation 4.3. Every K classical observations are aggregated into one single modal multi-valued object and thus we obtain a data set that consists of several modal multi-valued explanatory variables.

$$X_{ijk} \sim \text{bernoulli}(p_{ij}), \qquad i = 1, \ldots, I, \quad j = 1, \ldots, J, \quad k = 1, \ldots, K, \tag{4.3}$$

where p_{ij} is decided by the different situations and different classes. Table 4.12 shows the four different situations to be considered. In the rest of this section, we will consider these four situations one by one.

Situation 1

Let us consider the values of the response variable first. Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 4$ modal multi-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $N = I \times K = 1000$ classical observations are randomly drawn from the same distribution. Since there are 1000 observations in total, we can assume that the first 500 observations are in class 1, and the remaining 500 are in class 2. Since the response variable is a binary categorical variable with two possible classes, class 1 and class 2, we assume different variables for p_{ij} for class 1 and class 2 to make sure the values of the explanatory variables are good enough to distinguish the data from different classes well. We treat Situation 1 as the baseline; here we take $p^{(1)} = (p_1^{(1)}, p_2^{(1)}, p_3^{(1)}, p_4^{(1)}) = (0.1, 0.2, 0.3, 0.4)$ for class 1 and $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for class 2. Here, we set different probabilities for the variables in the same group because we want each explanatory variable to contain different information. For instance, we choose $p_1^{(1)} = 0.1$ as the probability for the first explanatory variable in class 1 and $p_2^{(1)} = 0.2$ as the probability for the second one in class 1. If we choose the same probability for different explanatory variables, then each explanatory variable is independent and identically distributed. In addition, because we assume that the distance between the data from different classes is small in this case, 0.3 is chosen here as the distance. As a result, we set $p_1^{(2)} = p_1^{(1)} + 0.3 = 0.4$ as the probability for the first explanatory variable in class 2 and $p_2^{(2)} = p_2^{(1)} + 0.3 = 0.5$ as the probability for the second one in class 2. We assume all the explanatory variables are modal multi-valued variables with only two possible values, 0

and 1. Therefore, for each explanatory variable, the probability of taking value 1 in class 2 is 0.3 greater than the probability in class 1.

Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into a modal multi-valued object and thus we can obtain a data set which consists of $J = 4$ modal multi-valued explanatory variables. There will be $I = 100$ groups after aggregation. The first two and the last groups are listed in Table 4.14. From the first row, the probability vector of value 1, $(0.1, 0.2, 0.3, 0.3)$ is good estimate of $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ for class 1; from the last row, the probability vector of value 1 is also very close to $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for the class 2.

Table 4.14: Part of modal multi-valued data set in Scenario 1 under Situation 1.

Group	Variable 1	Variable 2	Variable 3	Variable 4	Class
1	$\{1, 0.1; 0, 0.9\}$	$\{1, 0.2; 0, 0.8\}$	$\{1, 0.3; 0, 0.7\}$	$\{1, 0.3; 0, 0.7\}$	1
2	$\{1, 0.1; 0, 0.9\}$	$\{1, 0.4; 0, 0.6\}$	$\{1, 0.4; 0, 0.6\}$	$\{1, 0.5; 0, 0.5\}$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$\{1, 0.6; 0, 0.4\}$	$\{1, 0.4; 0, 0.6\}$	$\{1, 0.8; 0, 0.2\}$	$\{1, 0.5; 0, 0.5\}$	2

Assume the case is positive if the observation is in class 1. The true positive means the number of observations in class 1 has a correct prediction. Similarly, we can find the value of true negative, false positive, and false negative. Then, the accuracy for the model can be calculated by Equation 2.42 in Definition 2.2.8. After defining the positive case, we can calculate the precision, recall, and F-score by Equation 2.43, Equation 2.44, and Equation 2.45, respectively, in Definition 2.2.9. Table 4.15 summarizes the results for all the comparisons of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables. We split the symbolic data set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set. Then, we can use the mean of the four measurements to evaluate the CART for SD model, and use the variance of the four measurements to compare the stability of the CART for classical data and the CART for symbolic data. Take the accuracy for classical

CART as an example, the value 0.90 is the mean of all the $B = 100$ accuracies, and the value 0.003 is the variance.

Table 4.15: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 1 under Situation 1.

	accuracy	recall	precision	F-score	running time
CART	0.74 (0.0004)	0.75 (0.001)	0.72 (0.001)	0.74 (0.0005)	0.008 (<1e-5)
CART for SD	0.86 (0.005)	0.93 (0.008)	0.78 (0.011)	0.85 (0.006)	0.004 (0.001)

By comparing CART for classical data and CART for symbolic data, all the means of the prediction metrics are improved a lot for the symbolic data analysis, while the variances are slightly increased by aggregating the classical data to a symbolic type. One possible reason is that the randomness of the sample is reduced by aggregating the classical data. As a result, we can obtain a better prediction. However, the number of the symbolic observations is much less than the number of classical observations, making the variance higher. In addition, the running time of CART for symbolic data is much less after aggregation. The sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly. In addition, the variance of the running time increased after aggregating the classical data since the sample size is decreased.

Situation 2

Suppose we would like to generate a data frame with $J = 4$ modal multi-valued explanatory variables and $I = 1000$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 10000$ classical observations are randomly drawn from the same distribution. Compared with Situation 1 which had only 1000 observations, there will be 10000 observations in Situation 2. Suppose the first 5000 observations are in class 1, and the remaining 5000 are in class 2. We assume different p_{ij} for class 1 and class 2. Here, we take $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ for the class 1 and $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for the class 2 the

same as we assumed for Situation 1. In this way, we can control the other conditions to be the same and so can find the influence of sample size.

Suppose every $K = 10$ consecutive classical observations are in one group and are aggregated into a modal multi-valued object. Thus, we can obtain a data set that consists of $J = 4$ modal multi-valued explanatory variables. There will be $I = 1000$ groups after aggregation.

Table 4.16 summarizes all the comparisons of the performances of CART and CART for symbolic data with modal multi-valued explanatory variables for Situation 2. The method to obtain the values is the same as for Situation 1. We split the symbolic data set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set. Then, we can use the mean of the four measurements to evaluate the CART for SD model, and use the variance of the four measurements to compare the stability of the CART for classical data and the CART for symbolic data. All values are the mean of the corresponding column, the values in brackets are the variances.

Table 4.16: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 1 under Situation 2.

	accuracy	recall	precision	F-score	running time
CART	0.75 (<1e-4)	0.77 (0.0002)	0.71 (0.0001)	0.74 (<1e-4)	0.045 (<1e-5)
CART for SD	0.94 (0.0001)	0.95 (0.0006)	0.94 (0.0005)	0.94 (0.0001)	0.010 (0.003)

From Table 4.16, all the means of the prediction metrics are highly improved, while the variances are slightly increased by aggregating the classical data to a symbolic type. One possible reason is that the randomness of the sample is reduced by aggregating the classical data. As a result, we can obtain a better prediction. However, the number of symbolic observations is much less than the number of classical observations, making the variance higher. In addition, the running time of CART for symbolic data is much less after aggregation. The sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly. In addition, the variance of the running time increased after aggregating the classical data since the sample size is decreased.

Situation 3

Suppose we would like to generate a data frame with $J = 20$ modal multi-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the same distribution. Assume the first 500 observations are in class 1, and the remaining 500 observations are in class 2, which is the same as for Situation 1. Compared with Situation 1 which consists of 4 explanatory variables, we consider 20 explanatory variables in Situation 3. Similarly, we assume different p_{ij} for class 1 and class 2. Here we take $p^{(1)} = (p_1^{(1)}, \ldots, p_{20}^{(1)})$ for the class 1 and $p^{(2)} = (p_1^{(2)}, \ldots, p_{20}^{(2)})$ for the class 2. For this situation, instead of assuming the specific value of each probability as in Situation 1, we randomly generate a value from 0 to 0.7. The reason for choosing 0.7 here is to ensure that our assumptions about distance are consistent with Situation 1. In Situation 1, the probability to obtain value 1 for each explanatory variable in class 2 is 0.3 greater than the corresponding probability in class 1. In other words, we have

$$p^{(2)} = (p_1^{(2)}, \cdots, p_{20}^{(2)}) = p^{(1)} + (0.3, \ldots, 0.3) = (p_1^{(1)} + 0.3, \cdots, p_{20}^{(1)} + 0.3). \tag{4.4}$$

Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into a modal multi-valued object. Thus, we can obtain a data set that consists of $J = 20$ modal multi-valued explanatory variables. There will be $I = 100$ groups after aggregation. Table 4.17 lists all the comparisons of the performances of CART and CART for symbolic data with modal multi-valued explanatory variables. We split the symbolic data set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set. Then, we can use the mean of the four measurements to evaluate the CART for SD model, and use the variance of the four measurements to compare the stability of the CART for classical data and the CART for symbolic data.

Table 4.17: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 1 under Situation 3.

	accuracy	recall	precision	F-score	running time
CART	0.83 (0.0004)	0.84 (0.001)	0.82 (0.002)	0.83 (0.0005)	0.032 (0.011)
CART for SD	0.95 (0.001)	0.96 (0.005)	0.95 (0.003)	0.95 (0.001)	0.008 (0.001)

From Table 4.17, all the means of the prediction metrics are highly improved when using CART for SD, while the variances are slightly increased by aggregating the classical data to a symbolic type. One possible reason is that the randomness of the sample is reduced by aggregating the classical data. As a result, we can obtain a better prediction. However, the number of symbolic observations is much less than the number of classical observations, making the variance higher. In addition, the running time of CART for symbolic data is much less after aggregation. The sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly. In addition, the variance of the running time decreased after aggregating the classical data since the sample size is decreased. Compared with Situation 1 and Situation 2, the conclusion about the variance of the running time is the opposite of the previous conclusions. The sample size is reduced after aggregating, reducing the stability of the prediction. While the randomness is also reduced when grouping the data, improving the stability of the prediction. As a result, the impact on stability may be positive or negative after these two effects offset each other.

Situation 4

Suppose we would like to generate a data frame with $J = 4$ modal multi-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the same distribution. Assume that the first 500 observations are in class 1, and the remaining 500 observations are in class 2. Under this situation, a large distance between the data within the two classes is assumed. In this case, not only should we choose different p_{ij} for

the class 1 and class 2, but also the distance should be large. Under Situation 1, we take $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ for the class 1 and $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for the class 2. The distance between the two classes is 0.3 for all the explanatory variables. Here we still take $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ as the base line for the class 1, while the value of $p^{(2)}$ will be changed to $p^{(2)} = (0.6, 0.7, 0.8, 0.9)$ for the class 2. We have assumed that there are two possible values, 0 and 1, for each explanatory variable. Therefore, the probability for each explanatory variable to obtain value 1 in class 2 is 0.5 greater than the corresponding probability of class 1.

Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into a modal multi-valued object. Thus, we can obtain a data set that consists of $J = 4$ modal multi-valued explanatory variables. There will be $I = 100$ groups after aggregation. Table 4.18 shows all the comparisons of the performances of CART and CART for symbolic data with modal multi-valued explanatory variables. We split the symbolic data set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set. Then, we can use the mean of the four measurements to evaluate the CART for SD model, and use the variance of the four measurements to compare the stability of the CART for classical data and the CART for symbolic data.

Table 4.18: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 1 under Situation 4.

	accuracy	recall	precision	F-score	running time
CART	0.84 (0.0003)	0.82 (0.001)	0.86 (0.002)	0.84 (0.0004)	0.011 (0.002)
CART for SD	0.97 (0.001)	0.96 (0.003)	0.99 (0.001)	0.97 (0.001)	0.006 (0.001)

From Table 4.18, all the means of the prediction metrics are highly improved, while the variances are slightly increased by aggregating the classical data to a symbolic type when using CART for SD. One possible reason is that the randomness of the sample is reduced by aggregating the classical data. As a result, we can obtain a better prediction. However, the number of symbolic observations is much less than the number of classical observations,

making the variance higher. In addition, the running time of CART for symbolic data is much less after aggregation. The sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly. In addition, the variance of the running time decreased after aggregating the classical data since the sample size is decreased.

Comparison and Conclusions

Based on the previous tables (Table 4.15, Table 4.16, Table 4.17, and Table 4.18), the CART for symbolic data greatly reduces the running time because of the reduced number of observations for the four different situations according to the previous comparison between CART and CART for symbolic data. Therefore, CART for symbolic data can not only greatly improve the accuracy of prediction for all the situations in Scenario 1, but also reduces the running time. The only problem is that the sample size will be reduced after grouping the classical data, making the prediction less stable. However, the randomness is also reduced when grouping the data, improving the stability of the prediction. As a result, the impact on stability may be positive or negative after these two effects offset each other.

After comparing the CART for symbolic data and the CART for classical data for each situation, we can conclude that the CART for SD reduces the running time with a high accuracy. In addition, we can compare the results of different situations. All the results for the four situations are summarized in Table 4.19, including the mean and variance of the $B = 100$ values of accuracy, recall, precision, and F-score. The running time is also listed in Table 4.19.

By comparing Situation 1 and Situation 2, we can conclude that the performance of CART for symbolic data is improved when the sample size increases while the running time is much larger. According to the results of Situation 1 and Situation 3, the performance of CART for symbolic data is much better when the numbers of explanatory variables increase from

Table 4.19: Comparison of different situations for Scenario 1 using CART for SD.

	accuracy	recall	precision	F-score	running time
Situation 1	0.86 (0.005)	0.93 (0.008)	0.78 (0.011)	0.85 (0.006)	0.008 (0.001)
Situation 2	0.94 (0.0001)	0.95 (0.0006)	0.94 (0.0005)	0.94 (0.0001)	0.010 (0.003)
Situation 3	0.95 (0.001)	0.96 (0.005)	0.95 (0.003)	0.95 (0.001)	0.008 (0.001)
Situation 4	0.97 (0.001)	0.96 (0.003)	0.99 (0.001)	0.97 (0.001)	0.006 (0.001)

4 to 20. By checking the performance of Situation 1 and Situation 4, we can conclude that if the distance between the data from different classes is larger, the model is more accurate in predicting to which class the new observation belongs. This conclusion is very consistent with what we expected; that is, the larger the distance between the two classes of data, the easier it is to divide them correctly. In addition, the running time is increased if the number of input variables is increasing.

4.4.2 Scenario 2

In this scenario, we use the same method as for Scenario 1 to generate the response variable and explanatory variables. Therefore, we will not explain how to generate the variables in detail. We generate a data set with J explanatory variables and I groups with K classical observations in each group. The response variable will be categorical after aggregating since all the K classical observations in one group are in the same class as we assumed in Scenario 1. However, in Scenario 2, we would like to obtain a modal multi-valued response variable after aggregating. One way to simulate the modal multi-valued response variable is to randomly select several classical observations and change the value of the class. Take the first group as an example, there are K classical observations and all of them belong to class 1. If we group these K classical observations together, the aggregated response variable is categorical since all the classical observations are in class 1. If we change the response variable of 20% classical observations to construct class 2, then the aggregated response variable will be a modal multi-valued type realization, with $\xi_i = \{1, 0.8; 2, 0.2\}$ for the i^{th} group. Here, we

would like to simulate the scenario that there is one binary categorical response variable with two different classes and I groups of classical observations. In each group, there will be a major class in which most of the observations in the group are in that class and there will be several outliers in the group. As a result, there will be two kinds of group, the first kind of group with most of the observations in class 1 and the second kind of group with most of the observations in class 2.

Compared with Scenario 1 which assumes that the first half of the observations are in class 1 and the remaining observations are in class 2, in Scenario 2, the response variable of the first half of the observations are generated from a Bernoulli distribution with probability 0.8, and the remaining observations are from a Bernoulli distribution with probability 0.2. Here, the probability means the probability that the observation is from class 1. By this method, we can obtain observations in two different kinds of group, one with most of the observations in class 1 and the other one with most of the observations in class 2.

Situation 1

Suppose we would like to generate a data frame with $J = 4$ modal multi-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times J = 1000$ classical observations are randomly drawn from the same distribution. Assume there will be two kinds of group, the first kind of group with most of the observations in class 1 and the second kind of group with most of the observations in class 2. Suppose the first 50 groups belong to the first kind of group with the response variable generated from a Bernoulli distribution with probability 0.8, and the remaining 50 groups belong to the second kind of group with the response variable generated from a Bernoulli distribution with probability 0.2. To make sure the values of the explanatory variables are good enough to distinguish the data from different kinds of group well, we assume different variables for p_{ij} for two kinds of group. We treat Situation 1 as the baseline; here we take

$p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ for the first kind of group and $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for the second kind of group.

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into modal multi-valued objects and thus we can obtain a data set which consists of $J = 4$ modal multi-valued explanatory variables and one modal multi-valued response variable. There will be $I = 100$ groups after aggregation. The first two and the last groups are listed in Table 4.20. Compared with Table 4.7 for Scenario 1 with the categorical response variable, we have a modal multi-valued response variable in Scenario 2.

Table 4.20: Part of modal multi-valued data set in Scenario 2 under Situation 1.

Group	Variable 1	Variable 2	Variable 3	Variable 4	Response variable
1	$\{1, 0.2; 0, 0.8\}$	$\{1, 0.3; 0, 0.7\}$	$\{1, 0.3; 0, 0.7\}$	$\{1, 0.4; 0, 0.6\}$	$\{1, 0.9; 2, 0.1\}$
2	$\{1, 0.1; 0, 0.9\}$	$\{1, 0.5; 0, 0.5\}$	$\{1, 0.5; 0, 0.5\}$	$\{1, 0.3; 0, 0.7\}$	$\{1, 0.9; 2, 0.1\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$\{1, 0.3; 0, 0.7\}$	$\{1, 0.5; 0, 0.5\}$	$\{1, 0.7; 0, 0.3\}$	$\{1, 0.9; 0, 0.9\}$	$\{1, 0.2; 0, 0.8\}$

For the original data set, we have a categorical response variable and it belongs to a classification problem. However, the response variable for the symbolic data set is the modal multi-valued type and the prediction is more like a regression problem, making the performance metrics different and therefore cannot be compared directly. One way to compare these two methods is to use the predicted probability for a modal multi-valued response variable to predict the original class. Take the group 1 in Table 4.20 as an example, where the true value of the response variable is $\{1, 0.9; 2, 0.1\}$. After simulating the data, we can see there are 10 observations in group 1 and only one observation belongs to class 2. We can obtain the predicted response variable for group 1 by the CART for SD model. After fitting the model, the predicted response variable is $\{1, 0.89; 2, 0.11\}$. To predict the original class, the probability of class 1 is 0.89 for each observation in group 1. As a result, we can use the CART for SD model to predict the original class for classical data.

Assume the observation is positive if it is in class 1. The true positive means the number of classical observations in class 1 which we predict as being in class 1. Similarly we can find the value of true negative, false positive, and false negative. Then, the accuracy for the model can be calculated by Equation 2.42 in Definition 2.2.8. After defining the positive case, we can calculate the precision, recall, and F-score by Equation 2.43, Equation 2.44, and Equation 2.45 in Definition 2.2.9, respectively. Table 4.21 lists all the comparisons of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables to predict the class of the original classical data. We split the symbolic data set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set. Then, we can use the mean of the four measurements to evaluate the CART for SD model, and use the variance of the four measurements to compare the stability of the CART for classical data and the CART for symbolic data. Take the accuracy for classical CART as an example, the value 0.65 is the mean of all the $B = 100$ values of accuracy, and the value 0.0005 is the variance.

Table 4.21: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 2 under Situation 1.

	accuracy	recall	precision	F-score	running time
CART	0.65 (0.0005)	0.63 (0.002)	0.67 (0.006)	0.65 (0.001)	0.013 (0.003)
CART for SD	0.63 (0.0009)	0.61 (0.003)	0.62 (0.004)	0.61 (0.002)	0.007 (0.001)

From Table 4.21, all the means of the prediction metrics are similar in these two methods, while the variances are slightly increased by aggregating the classical data to a symbolic type. One possible reason is that the randomness of the sample is reduced by aggregating the classical data, reducing the random error. In addition, new errors are introduced when using the predicted probability in modal multi-valued response to simulate the original class. As a result, these two types of error offset each other, the predictions are almost the same after aggregating the classical data to a symbolic type. However, the number of symbolic observations is much less than the number of classical observations, making the

variance higher. In addition, the running time of CART for symbolic data is much less after aggregation. The sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly. In addition, the variance of the running time decreased after aggregating the classical data since the sample size is decreased.

Situation 2

Suppose we would like to generate a data frame with $J = 4$ modal multi-valued explanatory variables and $I = 1000$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 10000$ classical observations are randomly drawn from the same distribution. Compared with Situation 1 with only 1000 observations, now there will be 10000 observations in Situation 2. Assume there will be two kinds of group, the first kind of group with most of the observations in class 1 and the second kind of group with most of the observations in class 2. Suppose the first 500 groups belong to one kind of group with the response variable generated from a Bernoulli distribution with probability 0.8, and the remaining 500 groups belonging to another kind of group with the response variable generated from a Bernoulli distribution with probability 0.2. Here, we take $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ for the first kind of group and $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for the second kind as what we assumed for Situation 1. In this way, we can control the other conditions to be the same and so can study the influence of sample size.

Suppose every $K = 10$ consecutive classical observations are in one group and are aggregated into a modal multi-valued object. Thus, we can obtain a data set that consists of $J = 4$ modal multi-valued explanatory variables and one modal multi-valued response variable. There will be $I = 1000$ groups after aggregation. The method to obtain the explanatory variables is the same as for Situation 1. Table 4.22 lists all the comparisons of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables to predict the class of the original classical data. We split the symbolic data

set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set. Then, we can use the mean of the four measurements to evaluate the CART for SD model, and use the variance of the four measurements to compare the stability of the CART for classical data and the CART for symbolic data. All the means and variances of the prediction metrics are similar in these two methods, and the running time of CART for symbolic data is much less after aggregation.

Table 4.22: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 2 under Situation 2.

	accuracy	recall	precision	F-score	running time
CART	0.64 (<1e-4)	0.65 (0.0003)	0.64 (0.001)	0.64 (0.0003)	0.049 (0.006)
CART for SD	0.65 (<1e-4)	0.66 (0.0003)	0.66 (0.0003)	0.66 (0.0002)	0.010 (0.004)

Situation 3

Suppose we would like to generate a data frame with $J = 20$ modal multi-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the same distribution. Assume there will be two kinds of group, the first kind of group with most of the observations in class 1 and the second kind of group with most of the observations in class 2. Suppose the first 50 groups belong to one kind of group with the response variable generated from a Bernoulli distribution with probability 0.8, and the remaining 50 groups belong to another kind of group with the response variable generated from a Bernoulli distribution with probability 0.2. Compared with Situation 1 which consists of $J = 4$ explanatory variables, now we consider $J = 20$ explanatory variables for Situation 3. Similarly, we assume different p_{ij} for the two kinds of groups. Here, we take $p^{(1)} = (p_1^{(1)}, \ldots, p_{20}^{(1)})$ for the first kind of group and $p^{(2)} = (p_1^{(2)}, \ldots, p_{20}^{(2)})$ for the second one. For this situation, instead of assuming the specific value of each probability as in Situation 1, we randomly generate a value from 0 to 0.7. The reason for choosing 0.7 is exactly the same as for Situation 3 in Scenario 1 which

consists of $J = 4$ explanatory variables and for each explanatory variable, the distance of the probability between the two classes is 0.3. As a result, the distance between $p^{(1)}$ and $p^{(2)}$ is 0.3, i.e.,

$$p^{(2)} = (p_1^{(2)}, \ldots, p_{20}^{(2)}) = (p_1^{(1)} + 0.3, \ldots, p_{20}^{(1)} + 0.3) = p^{(1)} + (0.3, \ldots, 03). \tag{4.5}$$

Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into a modal multi-valued object. Thus, we can obtain a data set that consists of $J = 20$ modal multi-valued explanatory variables and one modal multi-valued response variable. There will be $I = 100$ groups after aggregation. Table 4.23 lists all the comparisons of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables to predict the class of the original classical data. The method to obtain the explanatory variables is the same as for Situation 1, except we generate $J = 20$ explanatory variables here rather than $J = 4$ explanatory variables as in Situation 1. Table 4.23 lists all the comparisons of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables to predict the class of the original classical data. We split the symbolic data set into a training set and a testing set, $B = 100$ times, and record the $B = 100$ values of the prediction metrics on the testing set.

Table 4.23: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 1 under Situation 3.

	accuracy	recall	precision	F-score	running time
CART	0.70 (0.0006)	0.71 (0.002)	0.68 (0.003)	0.69 (0.0009)	0.037 (0.006)
CART for SD	0.61 (0.001)	0.62 (0.003)	0.48 (0.002)	0.54 (0.001)	0.010 (0.002)

From Table 4.23, all the means of the prediction metrics are worse and the variances are slightly increased by aggregating the classical data to a symbolic type. One possible reason is that new errors are introduced when using the predicted probability in modal multi-valued

response to simulate the original class. The sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly.

Situation 4

Suppose we would like to generate a data frame with $J = 4$ modal multi-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the same distribution. Assume there will be two kinds of group, the first kind of group with most of the observations in class 1 and the second kind of group with most of the observations in class 2. Suppose the first 50 groups belong to one kind of group with the response variable generated from a Bernoulli distribution with probability 0.8, and the remaining 50 groups belong to the second kind of group with the response variable generated from a Bernoulli distribution with probability 0.2. Under this situation, a large distance between data with the two classes is assumed. In this case, not only should we choose different p_{ij} for the two kinds of groups, but also the distance should be large. Under Situation 1, we take $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ for the first kind of group with class 1 as the major class and $p^{(2)} = (0.4, 0.5, 0.6, 0.7)$ for the second kind of group with class 2 as the majority. The distance between two classes is 0.3 for all the explanatory variables. Here we still take $p^{(1)} = (0.1, 0.2, 0.3, 0.4)$ as the baseline, while the value of $p^{(2)}$ will be changed to $p^{(2)} = (0.6, 0.7, 0.8, 0.9)$.

Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into a modal multi-valued object. Thus, we can obtain a data set that consists of $J = 4$ modal multi-valued explanatory variables. There will be $I = 100$ groups after aggregation. Table 4.24 lists all the comparisons of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables to predict the class of original classical data. The method to obtain the values is the same as for Situation 1. From Table 4.24, we see that all the means of the prediction metrics are similar in these two

methods, while the variances are slightly increased by aggregating the classical data to a symbolic type. In addition, the sample size is greatly reduced by aggregating the classical data to a symbolic type, thus reducing the running time accordingly.

Table 4.24: Comparison of the performances of CART and CART for symbolic data (SD) with modal multi-valued explanatory variables in Scenario 1 under Situation 4.

	accuracy	recall	precision	F-score	running time
CART	0.70 (0.0006)	0.72 (0.001)	0.70 (0.001)	0.71 (0.0006)	0.012 (0.003)
CART for SD	0.66 (0.0008)	0.68 (0.002)	0.68 (0.002)	0.68 (0.002)	0.006 (0.001)

Comparison and Conclusion

Based on the Table 4.21, Table 4.22, Table 4.23, and Table 4.24 for Scenario 2, we conclude that the CART for symbolic data greatly reduces the running time because of the reduced number of observations for all four different situations. Therefore, CART for symbolic data can maintain a similar prediction accuracy while reducing running time for all four situations in Scenario 2. It is obvious that the CART for symbolic data can also be used successfully for classification on the classical data rather than just for symbolic data.

After comparing the CART for symbolic data and the CART for classical data for each situation, we can conclude that the CART for SD reduces the running time with a high accuracy. In addition, we can compare the results of different situations. Table 4.25 summarizes the results for all four situations, with the root Mean Square Error (RMSE) calculated by Equation 2.46 in Definition 2.2.10, and the running time for symbolic data. We split the symbolic data set into a training set and a testing set, $B = 100$ times and record the $B = 100$ values of RMSE on the testing set. Then, we can use the mean of RMSE to evaluate the performance of the CART for symbolic data model, and use the variance of RMSE to compare the stability under different situations.

The difference between Situation 1 and Situation 2 is that the number of groups in Situation 1 is 100 and the number of groups in Situation 2 is 1000. By comparing Situation 1

Table 4.25: Comparison of different situations in Scenario 2 using CART for SD.

	RMSE (mean)	RMSE (var)	running time (mean)	running time (var)
Situation 1	0.21	0.001	0.007	0.001
Situation 2	0.17	0.0001	0.010	0.006
Situation 3	0.12	0.0003	0.010	0.002
Situation 4	0.15	0.0008	0.006	0.001

and Situation 2, we can conclude that the prediction of CART for symbolic data is even worse when the sample size increases since the Root Mean Square Error (RMSE) for Situation 2 is 0.17, which is smaller than the RMSE of 0.21 for Situation 1. This finding is consistent with our assumptions. Generally speaking, the larger the number of observations, the more convincing and stable is the prediction of the model. The mean value of RMSE will be reduced to 0.12 from 0.21 if there are 20 explanatory variables, rather than the 4 explanatory variables in Situation 1. By checking the performance of Situation 1 and Situation 3, the performance of CART for symbolic data will be improved when the numbers of explanatory variables increases. According to the results of Situation 1 and Situation 4, the performance of CART for symbolic data will be better if the data with different outputs have a larger distance from each other.

4.4.3 Scenario 3

Suppose we would like to generate a data frame with J interval-valued explanatory variables and I groups of data with K classical observations in each group. To simulate this data set, $N = I \times K$ classical observations are randomly drawn from the normal distributions, i.e., $\{X_{i1k}, \ldots, X_{iJk}\} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), i = 1, \ldots, I, k = 1, \ldots, K$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are decided by different situations. Suppose every K classical observations are aggregated into one single interval-valued object and thus we obtain a data set that consists of J interval-valued explanatory variables. Tables 4.13 shows thee seven different situations to be considered. In the rest of this section, we will consider these seven situations one by one. Based on

the conclusion for Scenario 1 and Scenario 2, the variance of all the prediction metrics will increase if we aggregate the classical data to a symbolic type. Therefore, for the Scenario 3, we just split the training set and testing set once to save the running time.

Situation 1

Let us consider the values of the response variable first. Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.6. Here, we use a diagonal matrix with elements equal to one since the scales for the explanatory variables are the same for this situation. In addition, we assume no correlation between all the explanatory variables in this situation. The case for correlated explanatory variables is considered in Situation 7. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables, i.e.,

$$\{X_{i1k}, \ldots, X_{i4k}\} \sim N_4 \left(\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \\ \mu_{i3} \\ \mu_{i4} \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right), i = 1, \ldots, I, k = 1, \ldots, K. \quad (4.6)$$

The mean value $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ in Equation 4.6 is to be defined. Let us consider the mean values of the response variable first. Since there are 1000 classical observations in total, we can assume that the first 500 observations are in class 1, and the remaining 500 observations are in class 2. In order to make sure the values of the explanatory variables are good enough to distinguish the data from different classes well, we assume the mean value

$\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \mu_{i2}^{(1)}, \mu_{i3}^{(1)}, \mu_{i4}^{(1)})' = (0, 0, 0, 0)'$ for the class 1 and $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \mu_{i2}^{(2)}, \mu_{i3}^{(2)}, \mu_{i4}^{(2)})' = (1.5, 1.5, 1.5, 1.5)'$ for the class 2. After generating the classical data with $I \times K = 1000$ observations, we can visualize the data set as shown in Figure 4.8. From Figure 4.8, we can see that there is a large amount of overlap in the points in the two different classes, which is consistent with our assumption. In addition, the points with different colors are not completely overlapped, which means that different groups can be distinguished from the information contained in the explanatory variables.

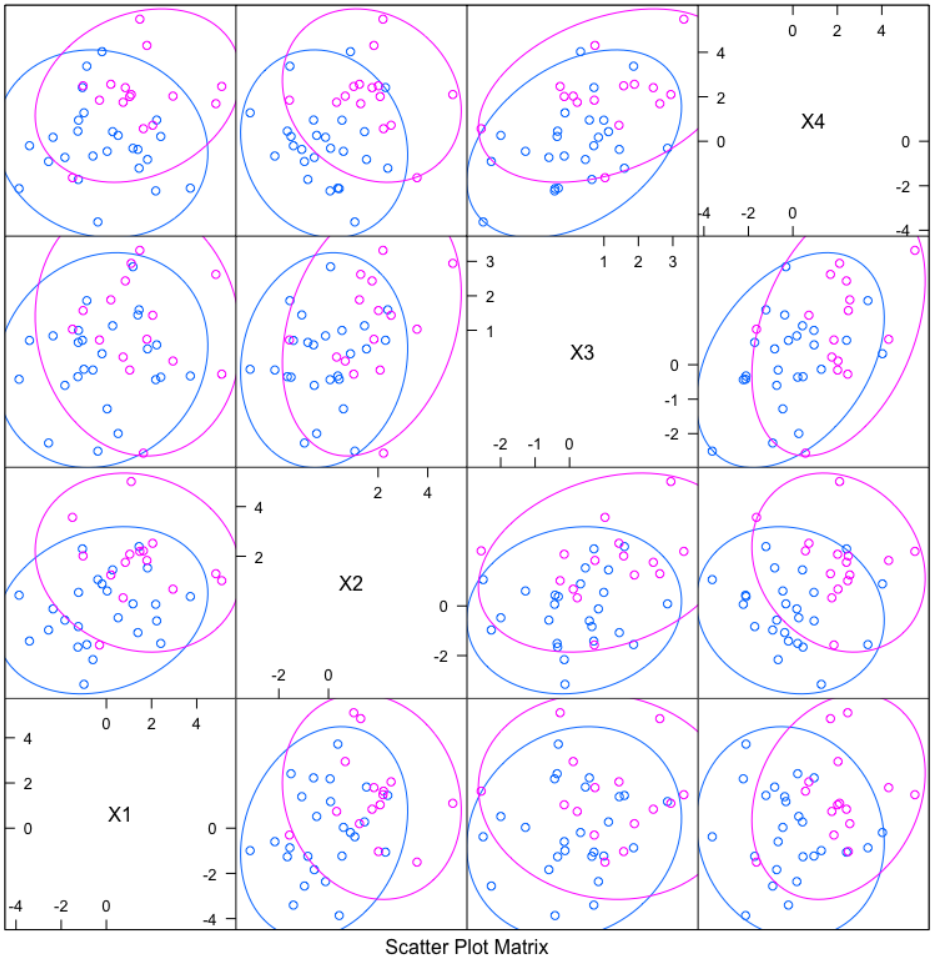

Figure 4.8: The scatter plot of classical data set in Scenario 3 under Situation 1.

Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation. The first two and the last two groups are listed in Table 4.26.

Table 4.26: Part of interval-valued data set in Scenario 3 under Situation 1.

Group	Variable 1	Variable 2	Variable 3	Variable 4	Class
Group 1	$[-1.78, 1.06]$	$[-0.10, 2.66]$	$[-1.80, 1.66]$	$[-2.18, -0.60]$	1
Group 2	$[-1.21, 2.25]$	$[-1.99, -0.11]$	$[-1.72, 0.29]$	$[-1.19, 1.32]$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Group 99	$[0.068, 3.39]$	$[-0.34, 2.78]$	$[0.85, 4.37]$	$[-0.58, 2.63]$	2
Group 100	$[-2.81, 2.36]$	$[0.03, 2.77]$	$[-0.65, 2.77]$	$[-0.24, 3.21]$	2

Figure 4.9 shows the scatter plot of these interval-valued data. From Figure 4.9, we can see that there is a large amount of overlap between the intervals in the two different classes, which is consistent with the assumption and the conclusion drawn in classical data.

Table 4.27 lists all the comparisons of the performance metrics of CART and CART for symbolic data for these interval-valued explanatory variables. Two different splitting methods, a bi-partition for two bounds of intervals and a triple partition based on the intervals, are considered. Five metrics, accuracy, recall, precision, F-score, and the code running time as defined in Definition 2.2.8 and Definition 2.2.9 are used to measure the performance of the classical CART model and CART for SD. From Table 4.27 we can see that all the metrics except running time are the same for the two different CART methods for symbolic data. The reason is that these two methods follow the same principle, which is to select the best explanatory variable with the corresponding "best" value and compare the "best" value with the lower bound and upper bound of the interval-valued explanatory variable. The reason for the difference in running time between the two CART for symbolic data methods is that the functions we used are different. The functions in the CART for symbolic data using binary partition are in the "rpart" package in R while the functions in the CART for symbolic data

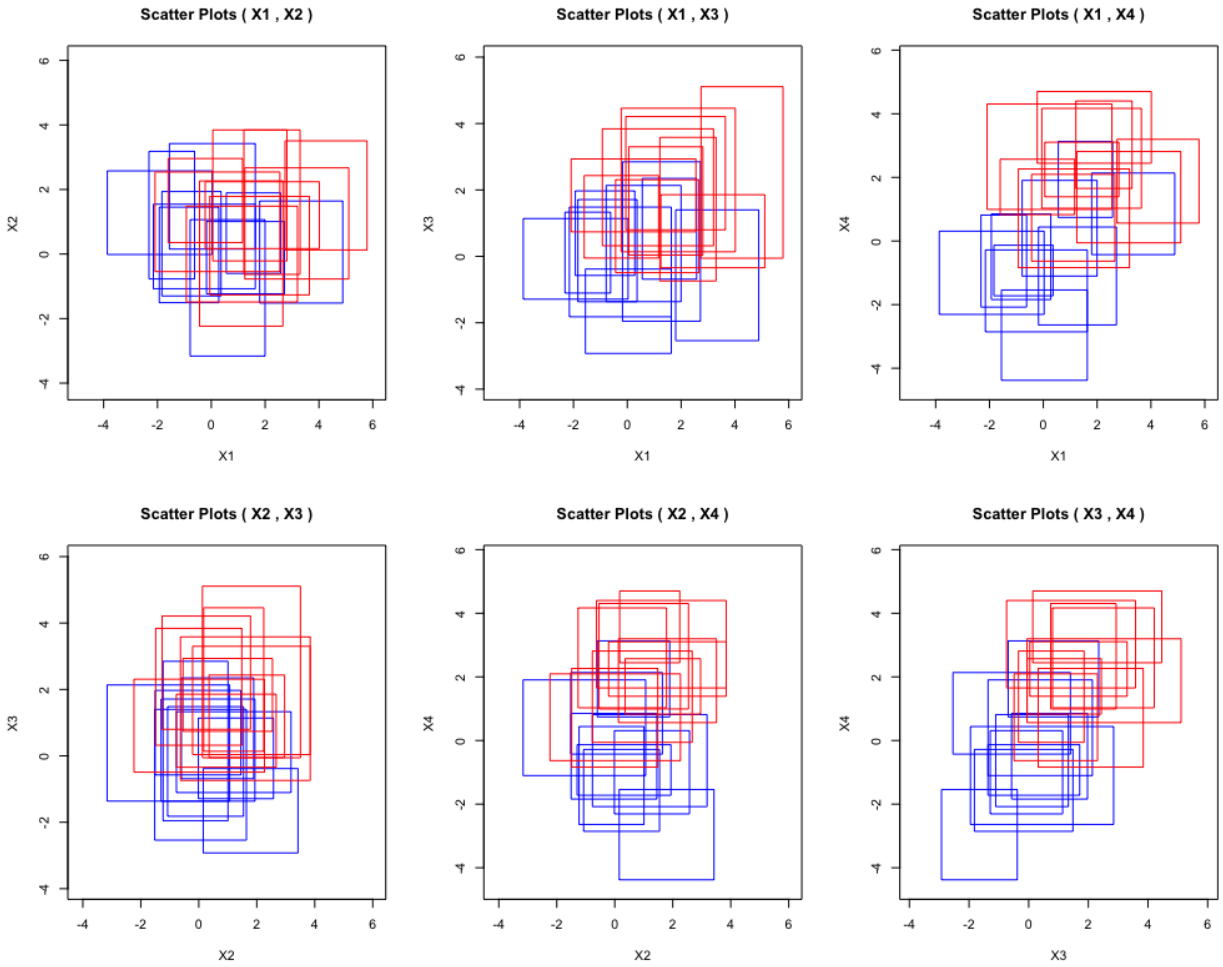

Figure 4.9: The scatter plot of interval-valued data set in Scenario 3 under Situation 1.

using the triple partition are written by myself. As a result, one future work is to write more effective functions. For the other scenarios with interval-valued explanatory variables, only the CART for symbolic data using the binary partition will be considered since it has the same performance but much less running time. By comparing the CART for classical data and the CART for symbolic data using the bi-partition method, all the metrics are very similar and the running time of CART for symbolic data is much less. The sample size is reduced by aggregating data into symbolic form, reducing the running time accordingly.

Therefore, the CART for symbolic data maintain a high accuracy, and greatly reduces the running time of the model at the same time.

Table 4.27: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 3 under Situation 1.

	accuracy	recall	precision	F-score	running time
CART	0.84	0.84	0.85	0.85	0.016
CART for SD (bi-partition)	0.83	0.82	0.88	0.85	0.006
CART for SD (tri-partition)	0.83	0.82	0.88	0.85	1.334

Situation 2

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 1000$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 10000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.6, which is the same as for Situation 1 except that the sample size has increased ten-fold. Then, we obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables.

Let us consider the values of the response variable first. Since there are 10000 classical observations in total, we can assume that the first 5000 observations are in class 1, and the remaining 5000 observations are in class 2. Assume the mean value $\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \mu_{i2}^{(1)}, \mu_{i3}^{(1)}, \mu_{i4}^{(1)})' = (0, 0, 0, 0)'$ and $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \mu_{i2}^{(2)}, \mu_{i3}^{(2)}, \mu_{i4}^{(2)})' = (1.5, 1.5, 1.5, 1.5)'$ for the class 1, and the class 2, respectively. Suppose every $K = 10$ consecutive classical observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 1000$ groups after aggregation.

Table 4.28 lists all the comparisons of the performances including accuracy, recall, precision, F-score, and running time of CART and CART for symbolic data using the bi-partition method with interval-valued explanatory variables. By comparing the CART for classical

data and CART for symbolic data, we can conclude that all the metrics of CART for symbolic data are slightly higher than the metrics of CART for classical data, and the running time is much less by aggregating the classical data to a symbolic type. The reason for better performance and less time is that grouping the data reduces the sample size and the possibility of over-fitting.

Table 4.28: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 3 under Situation 2.

	accuracy	recall	precision	F-score	running time
CART	0.77	0.78	0.77	0.78	0.119
CART for SD (bi-partition)	0.8	0.8	0.82	0.81	0.019

Situation 3

Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.6. Here we use the same diagonal matrix as in Situation 1. The only difference is that different $\boldsymbol{\mu}_i$ will be chosen. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables.

We still assume that the first 500 classical observations are in class 1, and the remaining 500 observations are in class 2. The only difference between Situation 1 and Situation 3 is the Signal-to-Noise Ratio (SNR) which is calculated by the ratio of signal power to the noise power. The larger the value of SNR, the easier it to distinguish two different classes. Compared to Situation 1 where SNR = 1.5, we now assume that SNR = 1. The mean value is still $\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \mu_{i2}^{(1)}, \mu_{i3}^{(1)}, \mu_{i4}^{(1)})' = (0, 0, 0, 0)'$ for the class 1, and the mean value becomes to $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \mu_{i2}^{(2)}, \mu_{i3}^{(2)}, \mu_{i4}^{(2)})' = (1, 1, 1, 1)'$ for the class 2. After generating the classical

data with $I \times K = 1000$ observations, we can visualize the data set as shown in Figure 4.10. From Figure 4.10, we can see that the points with different colors are much closer to each other than in Figure 4.8 under Situation 1, which is consistent with our assumption that the observations in different classes are closer under Situation 3 than in Situation 1. Furthermore, observations from different classes can still be distinguished by explanatory variables.

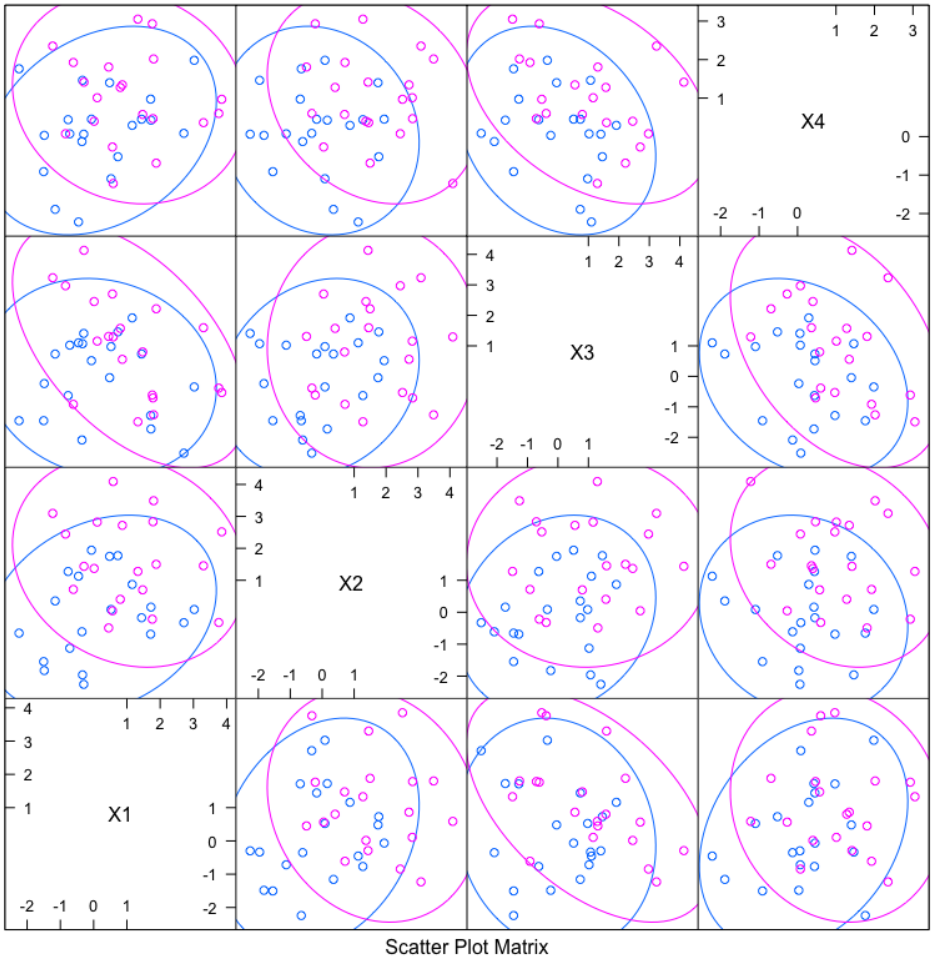

Figure 4.10: The scatter plot of classical data set in Scenario 3 under Situation 3.

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation. The first two

and the last groups are shown in Table 4.29. The intersection of the interval value of the same variable in the first two rows and the last row is very small, while the intersection of the interval value of each variable in the two rows of the same category is large. The finding is consistent with the assumption that the observations in different classes are closer under Situation 3 than in Situation 1.

Table 4.29: Part of interval-valued data set in Scenario 3 under Situation 3.

Group	Variable 1	Variable 2	Variable 3	Variable 4	Class
1	$[1.56, 2.85]$	$[-1.83, 2.66]$	$[-2.22, 0.07]$	$[-3.01, -1.37]$	1
2	$[-0.92, 1.90]$	$[-2.87, 0.28]$	$[-1.88, 1.03]$	$[-3.42, -1.08]$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$[1.90, 4.98]$	$[0.55, 5.49]$	$[-0.35, 2.67]$	$[-0.04, 2.82]$	2

Figure 4.11 shows the scatter plot of these interval-valued data. From Figure 4.11, we can see that the intervals are highly overlapped. The intervals in the different classes cannot be separated very well, which is consistent with the assumption and the conclusion drawn in the classical data.

Table 4.30 shows all the comparisons of the performances of CART and CART for symbolic data using the bi-partition method with interval-valued explanatory variables. The metrics, accuracy, recall, precision, and F-score, can be calculated by Equation 2.42 in Definition 2.2.8, Equation 2.43, Equation 2.44, and Equation 2.45 in Definition 2.2.9, respectively. For all of these metrics, the larger they are, the better. By comparing the CART for classical data and CART for symbolic data, all the prediction metrics are slightly improved by aggregating the data into the symbolic form, and the running time is the same. We can conclude that the results are as good as the classical CART model using CART for symbolic data.

Situation 4

Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 4$ interval-valued

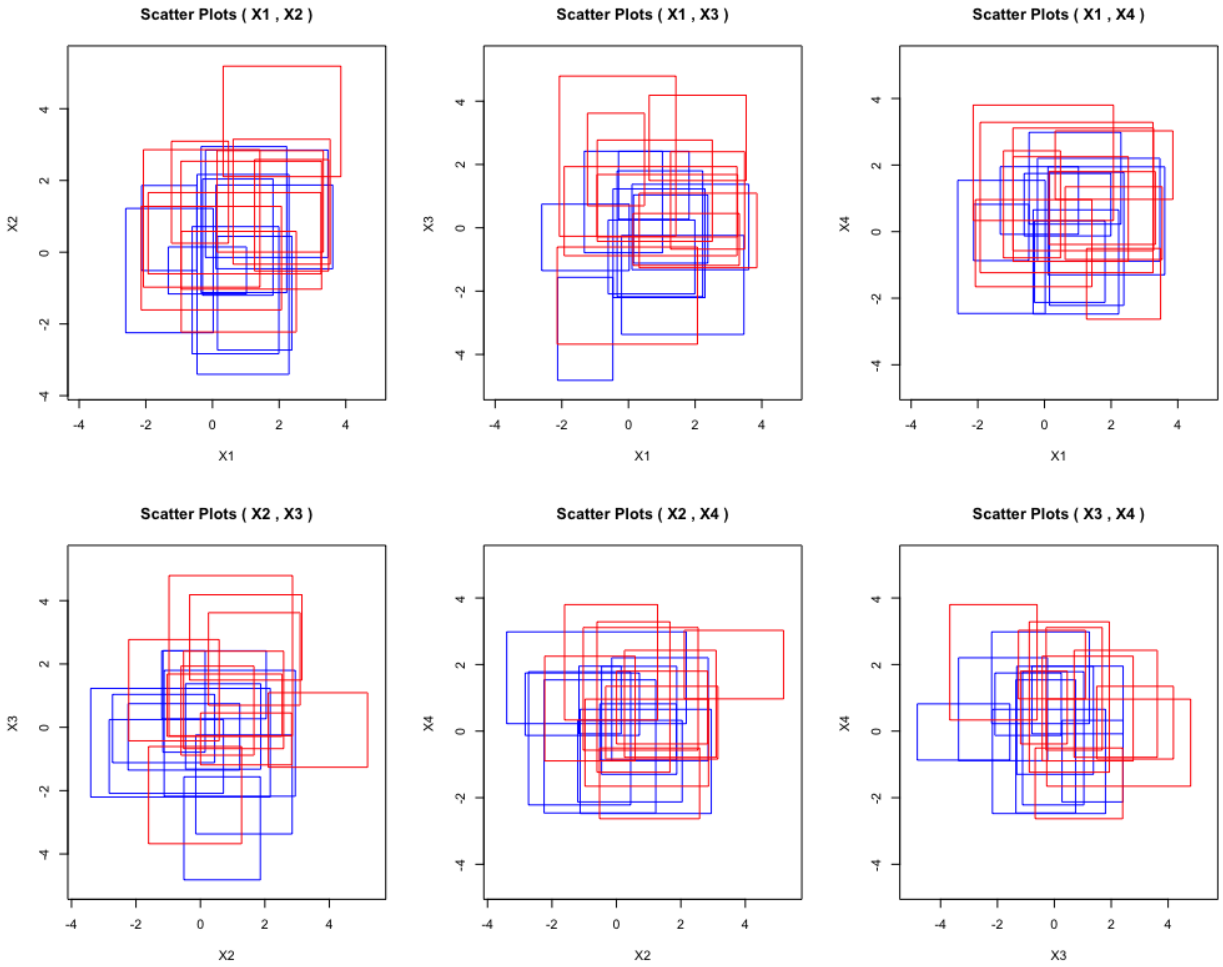

Figure 4.11: The scatter plot of interval-valued data set in Scenario 3 under Situation 3.

Table 4.30: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 3 under Situation 3.

	accuracy	recall	precision	F-score	running time
CART	0.68	0.74	0.55	0.63	0.004
CART for SD (bi-partition)	0.7	0.75	0.6	0.67	0.004

explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.6. Here we use the same diagonal matrix as in Situation 1. The different values of mean $\boldsymbol{\mu}_i$ will be chosen compared

to Situation 1 and Situation 3 where the SNR in Situation 1 is 1.5 and the SNR in Situation 3 is 1. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables.

We still assume that the first 500 observations are in class 1, and the remaining 500 are in class 2. The only difference between Situation 1 and Situation 3 is the Signal-to-Noise Ratio (SNR) which is the ratio of signal power to the noise power. The larger the value of SNR, the easier it to distinguish two different classes. Compared to Situation 1 where SNR = 1.5 and Situation 3 where SNR = 1, we now assume that SNR = 3. Situation 1, Situation 3, and Situation 4 are used to compare the influence of SNR. In addition, we can also find the prediction by CART for symbolic data under different situations. The mean value is still $\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \mu_{i2}^{(1)}, \mu_{i3}^{(1)}, \mu_{i4}^{(1)})' = (0, 0, 0, 0)'$ for the class 1, and the mean value becomes to $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \mu_{i2}^{(2)}, \mu_{i3}^{(2)}, \mu_{i4}^{(2)})' = (3, 3, 3, 3)'$ for the class 2. After generating the classical data with $I \times K = 1000$ observations, we can visualize the data set as shown in Figure 4.12. From Figure 4.12, we can see that the points with two different colors are well-separated by the explanatory variables, which is consistent with our assumption that the observations in different classes are farther apart under Situation 4 than in Situation 1.

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation. The first two and the last groups are shown in Table 4.31. The intersection of the interval value of one variable in any two rows in different classes is large. The finding is consistent with the assumption that the SNR is much larger under Situation 4 than in Situation 1.

Figure 4.13 shows the scatter plot of the interval-valued data. From Figure 4.13, we can see that there is only a very small amount of overlap in the intervals in these two different classes. The intervals in the different classes are separated very well, which is consistent with the assumption and the conclusion drawn for the classical data.

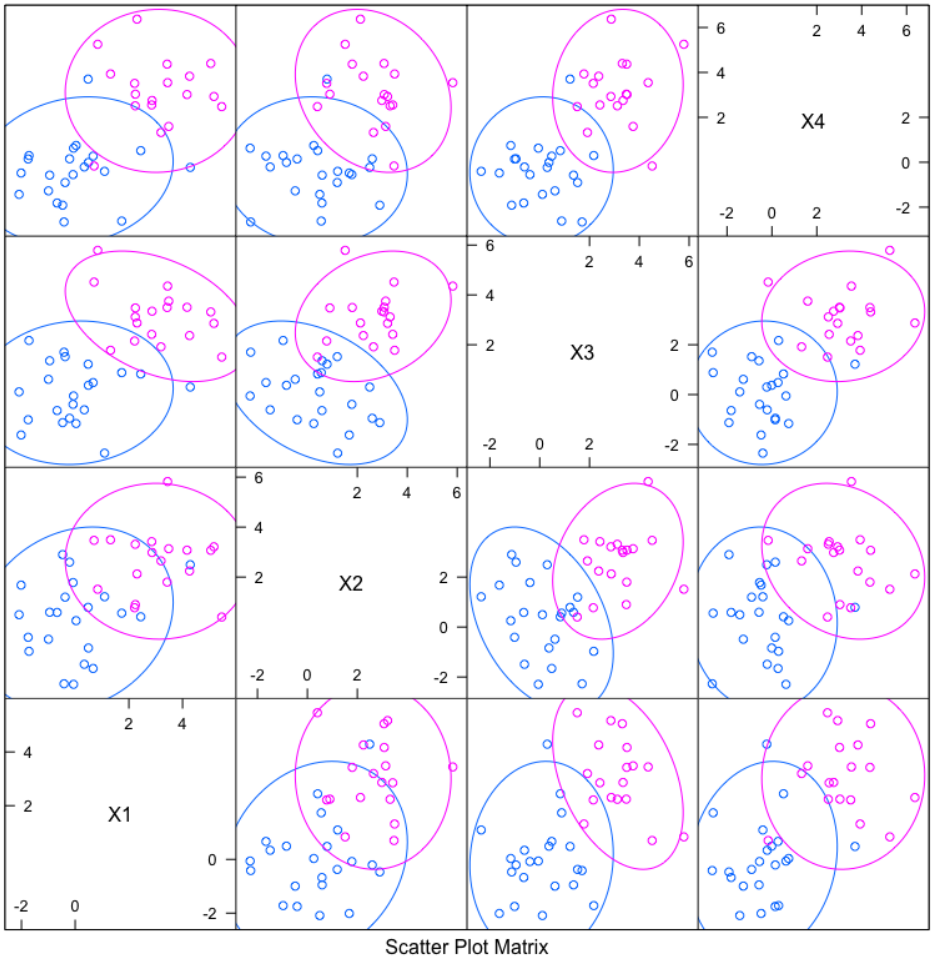

Figure 4.12: The scatter plot of classical data set in Scenario 3 under Situation 4.

Table 4.31: Part of interval-valued data set in Scenario 3 under Situation 4.

Group	Variable 1	Variable 2	Variable 3	Variable 4	Class
1	$[-2.01, 2.27]$	$[-2.18, 2.23]$	$[-2.02, 0.12]$	$[0.55, 3.51]$	1
2	$[-2.14, 1.01]$	$[-1.02, 2.32]$	$[-1.72, 1.61]$	$[0.54, 3.90]$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$[3.59, 5.81]$	$[2.04, 5.36]$	$[0.70, 2.37]$	$[1.98, 5.78]$	2

Table 4.32 shows all the comparisons of the performances of CART and CART for symbolic data using the bi-partition method with interval-valued explanatory variables. By comparing the CART for classical data and CART for symbolic data, all the prediction metrics are

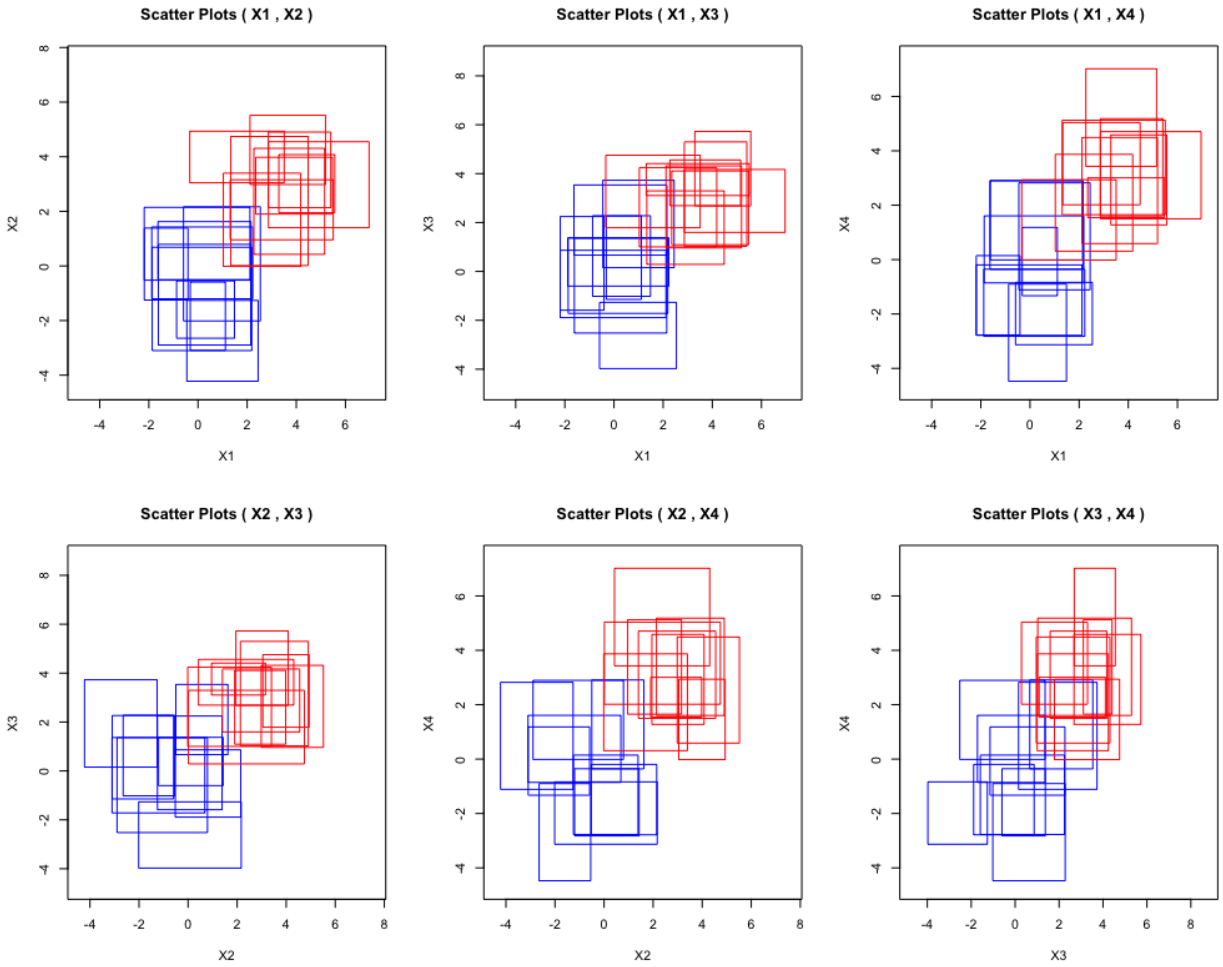

Figure 4.13: The scatter plot of interval-valued data set in Scenario 3 under Situation 4.

improved by aggregating the data. What is more, the running time is greatly reduced to half the time for the classical CART.

Table 4.32: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 3 under Situation 4.

	accuracy	recall	precision	F-score	running time
CART	0.85	0.82	0.9	0.86	0.008
CART for SD (bi-partition)	0.93	0.94	0.92	0.93	0.004

Situation 5

Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 20$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.7. A diagonal matrix with one on the diagonal is used because of the same scale and independent assumptions. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 20$ continuous explanatory variables, i.e.,

$$\{X_{i1}, \ldots, X_{i,20}\} \sim N_{20}\left(\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \\ \vdots \\ \mu_{i,20} \end{pmatrix}, \begin{pmatrix} 1 & 0 & \vdots & 0 \\ 0 & 1 & \vdots & 0 \\ \cdots & \cdots & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\right), i = 1, \ldots, I, k = 1, \ldots, K. \tag{4.7}$$

The mean value $\boldsymbol{\mu}_i = (\mu_{i1}, \ldots, \mu_{i,20})'$ in Equation 4.7 is to be defined. Let us consider the response variable first. Since there are 1000 classical observations in total, we can assume that the first 500 observations are in class 1, and the remaining 500 are in class 2. In order to make sure the values of the explanatory variables are good enough to distinguish the data from different classes well, we assume the mean value $\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \ldots, \mu_{i,20}^{(1)})' = (0, \ldots, 0)'$ for class 1 and $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \ldots, \mu_{i,20}^{(2)})' = (1.5, 1.5, \ldots, 1.5)'$ for class 2.

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 20$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation.

Table 4.33 shows all the comparisons of the performances of CART and CART for symbolic data using bi-partition with interval-valued explanatory variables. By comparing the CART

for classical data and CART for symbolic data, all the prediction metrics are similar by aggregating the data but the running time is greatly reduced.

Table 4.33: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 3 under Situation 5.

	accuracy	recall	precision	F-score	running time
CART	0.79	0.75	0.82	0.78	0.042
CART for SD (bi-partition)	0.73	0.83	0.62	0.71	0.009

Situation 6

Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.8. Here we use a diagonal matrix with different diagonal elements since the scales for explanatory variables are assumed to be different for this situation and also we assume no correlation between all the explanatory variables. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables, i.e.,

$$\{X_{i1}, \ldots, X_{i4}\} \sim N_4 \left(\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \\ \mu_{i3} \\ \mu_{i4} \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix} \right), i = 1, \ldots, I, k = 1, \ldots, K. \quad (4.8)$$

The mean value $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ in Equation 4.8 is to be defined. Let us consider the response variable first. Since there are 1000 classical observations in total, we can assume that the first 500 observations are in class 1, and the remaining 500 observations are in class 2.

In order to make sure the values of the explanatory variables are good enough to distinguish the data from different classes well, we assume the mean value $\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \mu_{i2}^{(1)}, \mu_{i3}^{(1)}, \mu_{i4}^{(1)})' = (0, 0, 0, 0)'$ for class 1 and $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \mu_{i2}^{(2)}, \mu_{i3}^{(2)}, \mu_{i4}^{(2)})' = (1.5, 1.5, 1.5, 1.5)'$ for class 2 similar to what we assumed in Situation 1. After generating the classical data with $I \times K = 1000$ observations, we can visualize the data set as shown in Figure 4.14. From Figure 4.14, we can see that there is a large amount of overlap in the points in the two different classes, which is consistent with our assumptions. In addition, the distribution for the points of each class presents an obvious ellipse shape, which is different from the approximate circle of Situation 1. This is because, in the present case, we assume that the scale of each explanatory variable is different so that the multivariate distribution appears as an ellipse instead of a circle. In addition, the overlap of the two colors in Figure 4.14 is more than that in Figure 4.8. The reason is that we assume a larger scale of variance while the difference in the means does not change. Therefore, the overlap will increase within the range of the explanatory variable.

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation. The first two and the last groups are listed in Table 4.34. There are $J = 4$ interval-valued explanatory variables in the data, the range of intervals is becoming larger from Variable 1 to Variable 4, which is to be expected because of the assumption that the variance gradually increases.

Table 4.34: Part of interval-valued data set in Scenario 3 under Situation 6.

Group	Variable 1	Variable 2	Variable 3	Variable 4	Class
1	$[-1.15, 1.38]$	$[-1.02, 4.53]$	$[-9.29, 0.88]$	$[-29.75, 19.68]$	1
2	$[-0.63, 1.57]$	$[-7.98, 5.60]$	$[-6.83, 4.91]$	$[-22.57, 12.27]$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$[0.53, 3.29]$	$[-3.22, 3.80]$	$[-3.03, 7.11]$	$[-18.23, 13.00]$	2

Figure 4.15 shows the scatter plot of the interval-valued data. From Figure 4.15, we can see that there is a large amount of overlap in the intervals in the two different classes and the

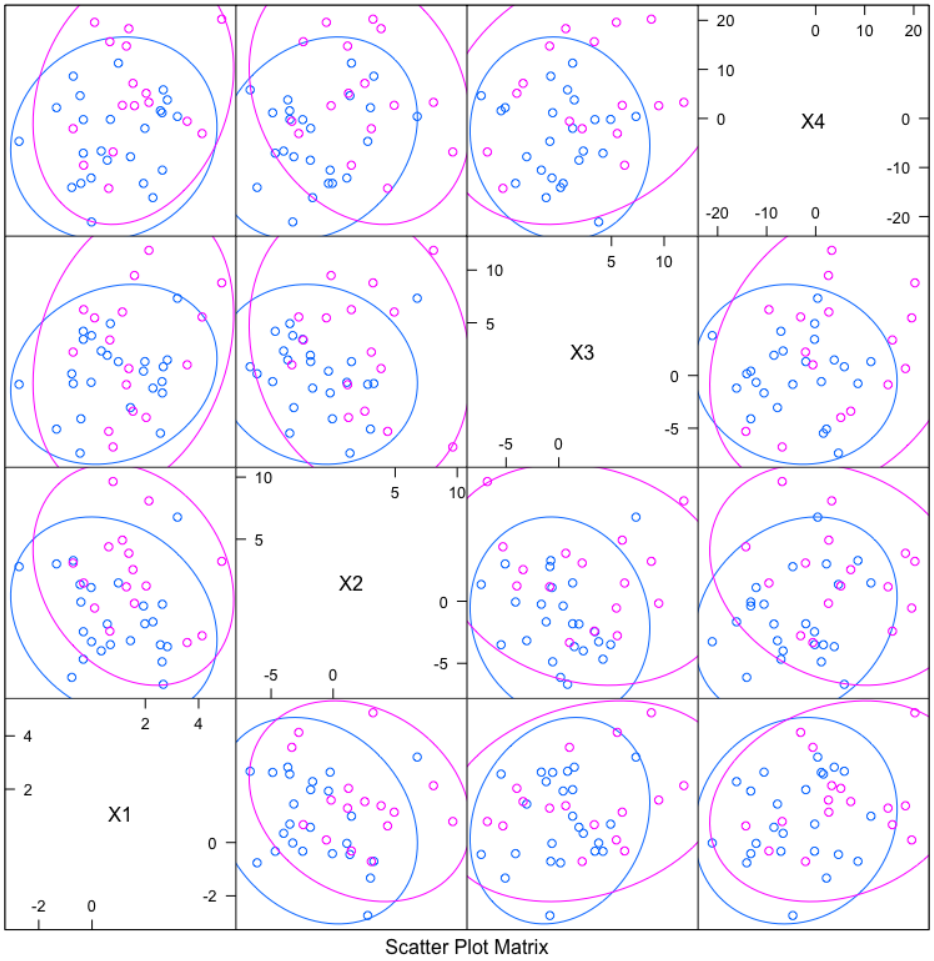

Figure 4.14: The scatter plot of classical data set in Scenario 3 under Situation 6.

shape between intervals is more like a rectangle than a square that presented in Situation 1, which is consistent with the assumptions and the conclusion drawn in the classical data case.

All the comparisons of the performances of CART and CART for symbolic data with interval-valued explanatory variables are shown in Table 4.35. By comparing the CART for classical data and CART for symbolic data, all the metrics are similar between the two models, but the running time of CART for symbolic data is much less with a two-thirds reduction. As a result, we see that the scale of the explanatory variable does not affect the

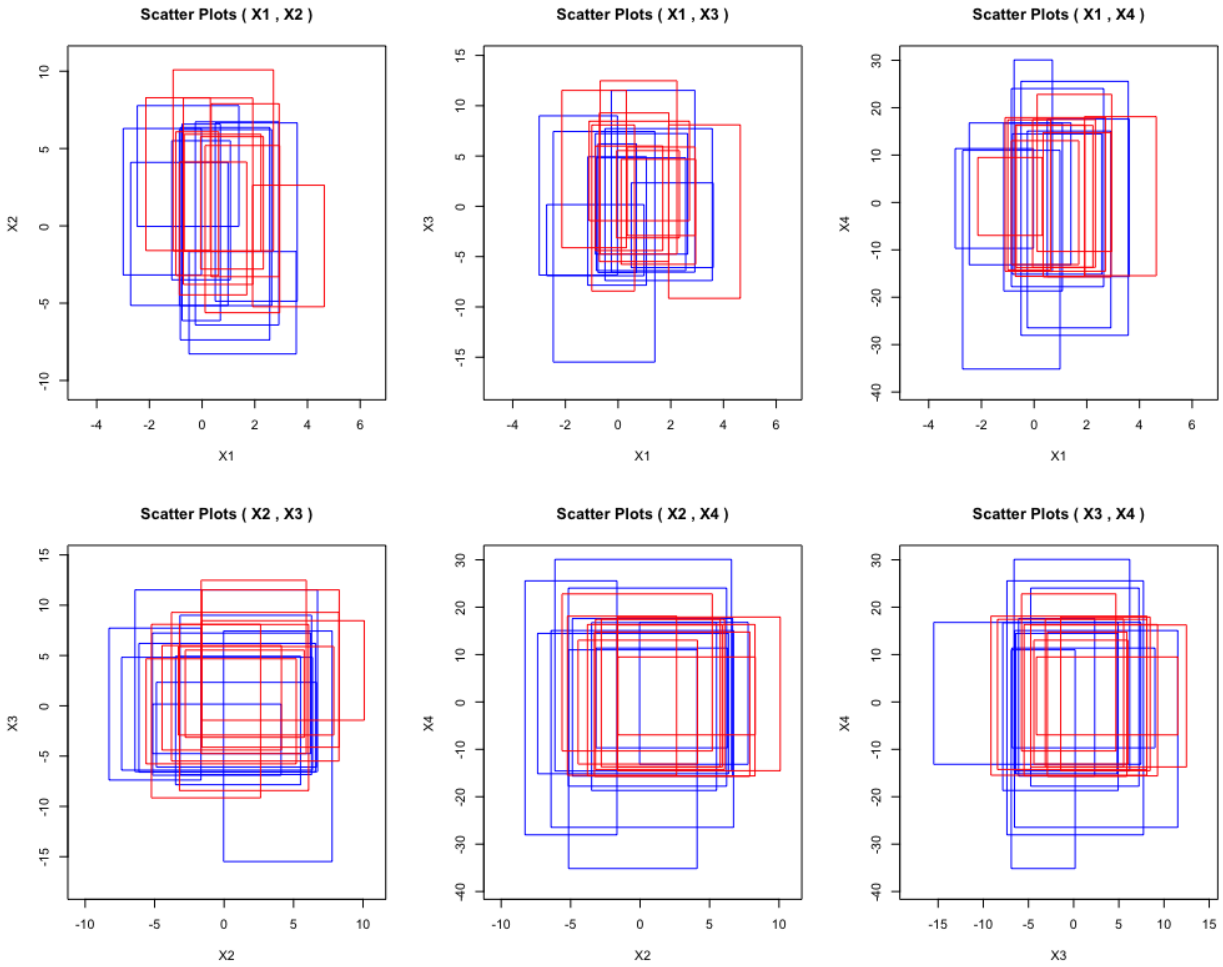

Figure 4.15: The scatter plot of interval-valued type data set in Scenario 3 under Situation 6.

results. The reason is that the tree-based model is not affected by the scale of explanatory variables.

Table 4.35: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 3 under Situation 6.

	accuracy	recall	precision	F-score	running time
CART	0.71	0.71	0.7	0.7	0.015
CART for SD (bi-partition)	0.7	0.79	0.65	0.71	0.005

Situation 7

Suppose the response variable is a binary categorical variable with two possible classes, class 1 and class 2. Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.9. Here we use a non-diagonal matrix with diagonals value one since the scales for explanatory variables are assumed to be the same for this situation and also we assume the explanatory variables are positively correlated. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables, i.e.,

$$\{X_{i1}, \ldots, X_{i4}\} \sim N_4 \left(\begin{pmatrix} \mu_{i1k} \\ \mu_{i2k} \\ \mu_{i3k} \\ \mu_{i4k} \end{pmatrix}, \begin{pmatrix} 1 & 0.8 & 0.5 & 0.2 \\ 0.8 & 1 & 0.8 & 0.5 \\ 0.5 & 0.8 & 1 & 0.8 \\ 0.2 & 0.5 & 0.8 & 1 \end{pmatrix} \right), i = 1, \ldots, I, k = 1, \ldots, K. \tag{4.9}$$

The mean value $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ in Equation 4.9 is to be defined. Let us consider the response variable first. Since there are 1000 classical observations in total, we can assume that the first 500 observations are in class 1, and the remaining 500 observations are in class 2. In order to make sure the values of the explanatory variables are good enough to distinguish the data from different classes well, we assume the mean value $\boldsymbol{\mu}_i^{(1)} = (\mu_{i1}^{(1)}, \mu_{i2}^{(1)}, \mu_{i3}^{(1)}, \mu_{i4}^{(1)})' = (0, 0, 0, 0)'$ for the class 1 and $\boldsymbol{\mu}_i^{(2)} = (\mu_{i1}^{(2)}, \mu_{i2}^{(2)}, \mu_{i3}^{(2)}, \mu_{i4}^{(2)})' = (1.5, 1.5, 1.5, 1.5)'$ for the class 2. After generating the classical data with $I \times K = 1000$ observations, we can visualize the data set as shown in Figure 4.16. From Figure 4.16, we can see that there is a large amount of overlap in the points in the two different classes. In addition, the distribution of the data

is elliptical and positively distributed. The reason for this shape is that we assume all the explanatory variables are positively correlated to each other.

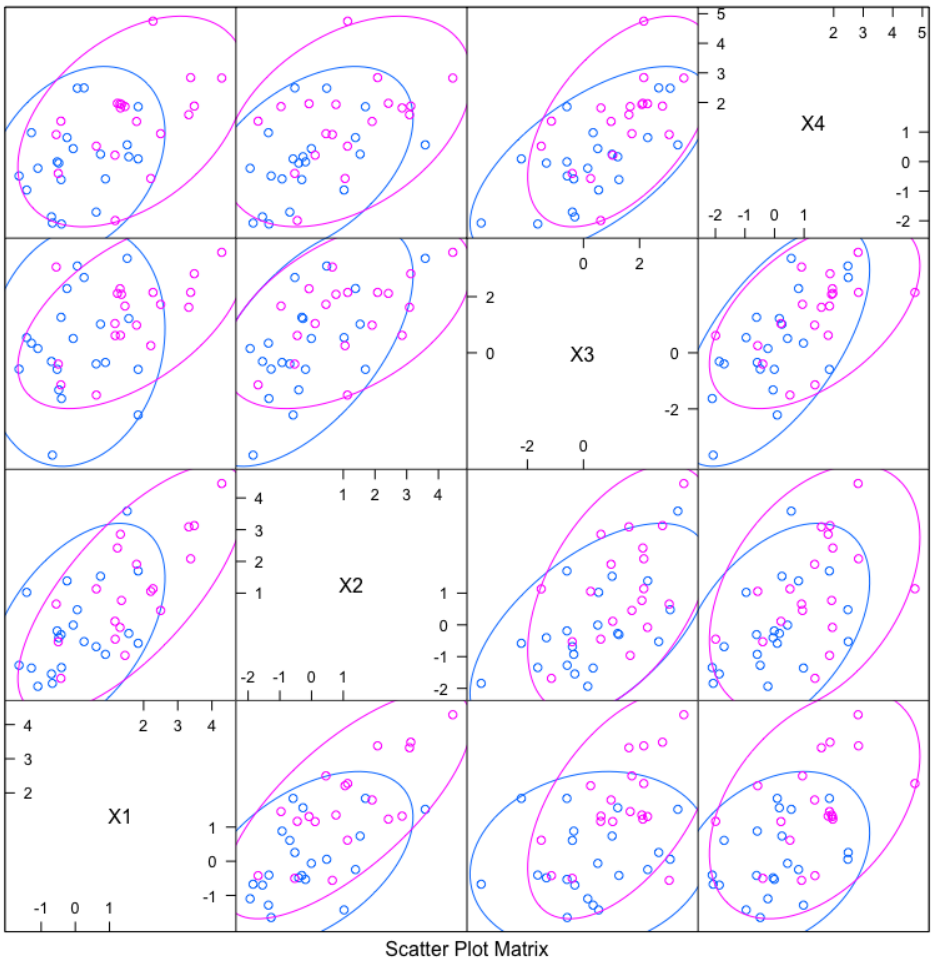

Figure 4.16: The scatter plot of classical data set in Scenario 4 under Situation 7.

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation. Figure 4.17 shows the scatter plot of the interval-valued data. From Figure 4.17, we can see that there is a large amount of overlap in the intervals in these two different classes, which is consistent with the assumption and the conclusion drawn in classical data.

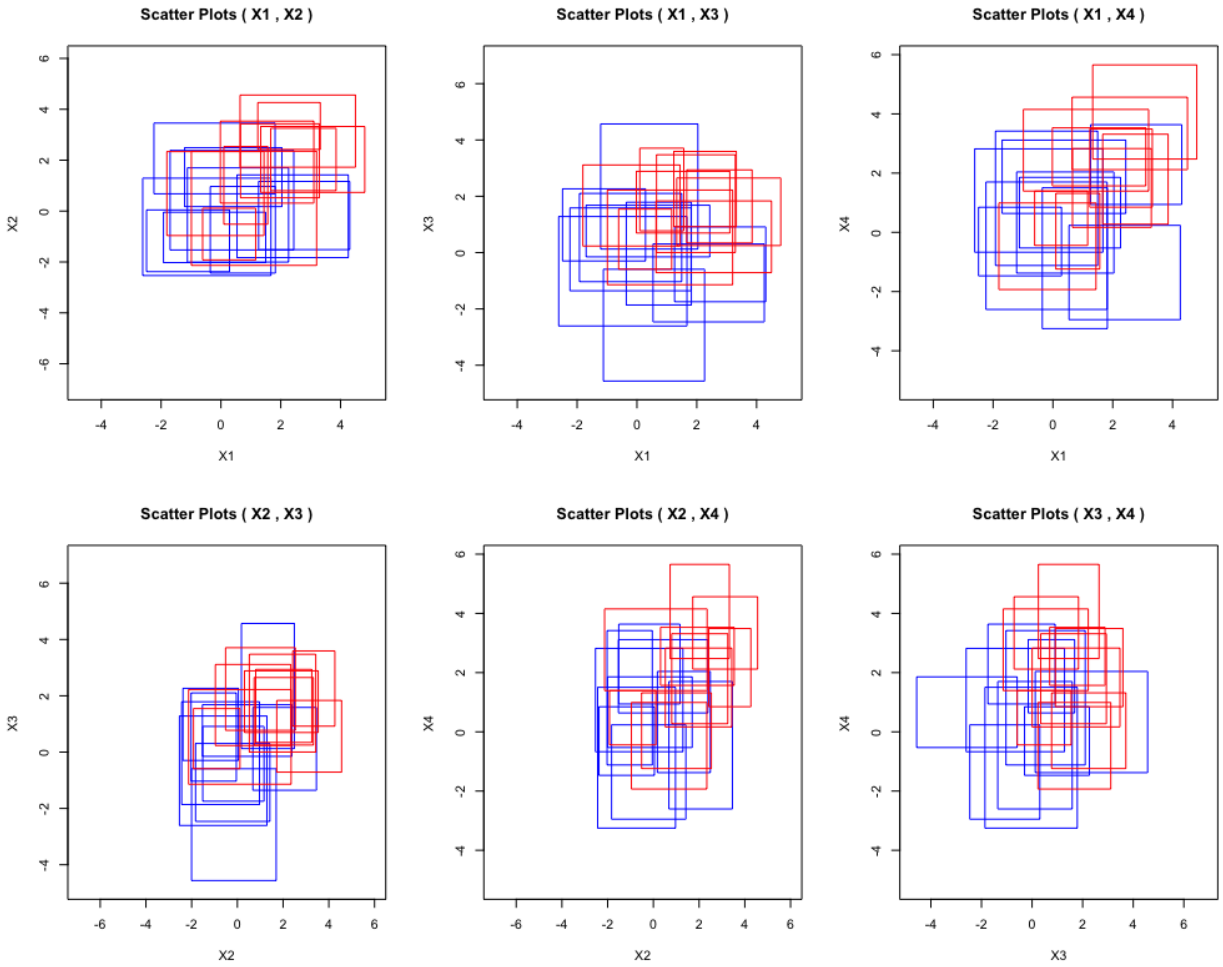

Figure 4.17: The scatter plot of interval-valued data set in Scenario 4 under Situation 7.

All the comparisons of the performances of CART and CART for symbolic data with interval-valued explanatory variables are summarized in Table 4.36. From Table 4.36, we can conclude that all the metrics are very similar overall while the running time of CART for symbolic data is much less.

Table 4.36: Comparison of the performances of CART and CART for symbolic data (SD) with interval-valued explanatory variables in Scenario 4 under Situation 7.

	accuracy	recall	precision	F-score	running time
CART	0.73	0.73	0.71	0.72	0.015
CART for SD (bi-partition)	0.77	0.61	1	0.76	0.005

Comparison and Conclusions

According to the previous comparisons between CART and CART for symbolic data, the CART for symbolic data greatly reduces the running time because of the reduced number of observations for all seven different situations. Therefore, CART for symbolic data can reduce the running time while maintaining the good performance metrics as for classical CART.

After comparing the CART for symbolic data and the CART for classical data for each situation, we can conclude that the CART for SD reduces the running time with a high value of accuracy. In addition, we can compare the results of different situations. Table 4.37 summarizes the results for all seven situations, including the accuracy, recall, precision, F-score, and running time.

Table 4.37: Comparison of different situations for data set with interval-valued explanatory variables for Scenario 3.

	accuracy	recall	precision	F-score	running time
Situation 1	0.83	0.82	0.88	0.85	0.006
Situation 2	0.8	0.8	0.82	0.81	0.019
Situation 3	0.7	0.75	0.6	0.67	0.004
Situation 4	0.93	0.94	0.92	0.93	0.004
Situation 5	0.73	0.83	0.62	0.71	0.009
Situation 6	0.7	0.79	0.65	0.71	0.005
Situation 7	0.77	0.61	1	0.76	0.005

By comparing Situation 1 and Situation 2, we can conclude that the performance of CART for the symbolic data model is comparable when the number of groups increases while the running time is improved accordingly. Generally speaking, the accuracy of classification will increase as the training sample size increases. However, the accuracy of the classical data is similar to Situation 1. The main reason may be that we randomly take observations from the same distribution. Even if the number of groups is greatly increased, the information gain of the data is not very large. By checking the performance of Situation 1, Situation 3, and Situation 4, we can conclude that if the data of different classes are originally more scattered,

the CART for the symbolic data model is more accurate in predicting the classes. The only difference between Situation 1, Situation 3, and Situation 4 is the value of SNR. As shown in Table 4.13, the SNR in Situation 1 is 1.5, the SNR in Situation 3 is 1, and the SNR in Situation 4 is 3. When the SNR is increasing, the difference between the two classes becomes larger, and it will be much easier to classify the data. This conclusion is very consistent with what we expected, that the larger the distance between the two classes of data, the easier it is to divide them correctly. According to the results of Situation 1 and Situation 5, the performance of CART for symbolic data with the different number of explanatory variables is similar. This conclusion is inconsistent with our expectation that the larger the number of explanatory variables is, the better the fitting model. The reason here may be that the simulation data are randomly generated from a multivariate normal distribution, so increasing the number of explanatory variables will not bring much extra information. In addition, if the number of input variables increases, the running time increases. From the performance of Situation 1 and Situation 6, we can conclude that whether the scales of explanatory variables are the same or not makes a difference in the prediction. The variation between intervals will also increase when the variance of the explanatory variable increases. Finally, based on the performance of Situation 1 and Situation 7, the data with correlated explanatory variables will have a worse result than the data with independent explanatory variables. This finding is also an intuitive conclusion. When the number of explanatory variables is fixed, the information contained in the correlated explanatory variables is smaller than for independent explanatory variables.

4.4.4 Scenario 4

In this scenario, the response variable is interval-valued. Suppose we would like to generate a data frame with J interval-valued explanatory variables and I groups of data with K observations in each group. To simulate this data set, $I \times K$ classical observations are

randomly drawn from the multivariate normal distribution of Equation 4.10. Then, we can obtain a classical data set with $I \times K$ observations and J continuous explanatory variables, i.e.,

$$\{X_{i1k}, \ldots, X_{iJk}\} \sim N_J \left(\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \\ \vdots \\ \mu_{iJ} \end{pmatrix}, \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1J} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2J} \\ \vdots & \vdots & & \vdots \\ \sigma_{J1} & \sigma_{J2} & \cdots & \sigma_{JJ} \end{pmatrix} \right), i = 1, \ldots, I, k = 1, \ldots, K, \tag{4.10}$$

where all the parameters are to be decided. Every K classical observations are aggregated into an interval-valued object and thus we obtain a data set that consists of J interval-valued explanatory variables. Based on the conclusion for Scenario 1 and Scenario 2, the variance of all the prediction metrics will increase if we aggregate the classical data to a symbolic type. Therefore, for the Scenario 4, we still split the training set and testing set $B = 100$ times to obtain the mean of Root Mean Square Error (RMSE).

Situation 1

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.11. Here, we use a diagonal matrix with elements equal to one since the scales for the explanatory variables are the same for this situation. In addition, we assume no correlation between all the explanatory variables in this situation. The case for correlated data is considered in Situation 7. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables, i.e.,

$$\{X_{i1k},\ldots,X_{i4k}\} \sim N_4\left(\left(\begin{array}{c}\mu_{i1}\\ \mu_{i2}\\ \mu_{i3}\\ \mu_{i4}\end{array}\right),\left(\begin{array}{cccc}1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\end{array}\right)\right), i=1,\ldots,I, k=1,\ldots,K, \quad (4.11)$$

where $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ is generated from normal distribution with mean 0 and variance 1. The values of the means are the same for the observations in one group and are different for observations from different groups.

After generating the values for four explanatory variables, we can assume the numerical response variable is from the linear regression:

$$y_{ik} = \beta_0 + \beta_1 X_{i1k} + \cdots + \beta_4 X_{i4k} + \epsilon_{ik}, \quad i = 1,\ldots,I, k = 1,\ldots,K, \quad (4.12)$$

where β_0 is assumed to be 1, $\beta_j = j$ for the j^{th} explanatory variable. Here we choose $\beta_j = j$ because we would like to simulate the situation that the effects on the response variable of the different explanatory variables are not the same. We can also choose other values. The ϵ_{ik} is the random error for the k^{th} classical observation in the i^{th} group, which is normally distributed with mean 0 and variance 1. For example, if we want to investigate the purchase of a product in one state to predict how many products should be shipped to the state. If the product is invested too much, it will cause a product backlog and cause great losses. If the product input is too small, it will lead to short supply and reduce sales. There are many cities in a state, and each city has many selling points. We regard a city as a group and multiple sales points in the city as multiple observations of the group. The classical data we collected contains I cities (group), and each city has K observations. For each selling point, we record the total number of purchases as the response variable and record some information about the selling point as the explanatory variable. Suppose we have collected $J = 4$ numerical

information, including the area of the selling point, the number of nearby communities, the distance from the nearest subway station or bus station to the selling point, and the number of products of the same type sold at the selling point. We hope to use these four explanatory variables to predict the range of sales in each city. Therefore, we aggregate the sales points in the same city into a group and obtain an interval-valued response variable and $J = 4$ interval-valued explanatory variables. Here, we assume the four explanatory variables, the area of the selling point, the number of nearby communities, the distance from the nearest subway station or bus station to the selling point, and the number of products of the same type sold at the selling point, have an increasing influence on the sales volume. Therefore, we choose different $\beta_j = j$ for the four explanatory variables. Table 4.38 shows part of the original data set generated by R.

Table 4.38: Part of original data set in Scenario 4 under Situation 1.

	Variable 1	Variable 2	Variable 3	Variable 4	response variable
1	2.70	0.90	-3.06	-0.92	-7.06
2	2.32	-1.03	-2.34	-1.68	-12.58
⋮	⋮	⋮	⋮	⋮	⋮
1000	2.02	-0.28	-0.60	0.46	2.52

Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables along with an interval-valued response. There will be $I = 100$ groups after aggregation. Table 4.39 lists part of the symbolic data set after aggregation.

Table 4.39: Part of symbolic data set in Scenario 4 under Situation 1.

Group	Variable 1	Variable 2	Variable 3	Variable 4	response variable
1	$[-0.11, 3.32]$	$[-1.77, 1.37]$	$[-3.42, 0.20]$	$[-1.68, 0.80]$	$[-13.84, 6.35]$
2	$[-2.82, 0.45]$	$[-3.05, 0.38]$	$[-1.56, 0.81]$	$[-0.18, 3.14]$	$[-6.62, 5.13]$
⋮	⋮	⋮	⋮	⋮	⋮
100	$[-0.33, 3.15]$	$[-1.52, 1.04]$	$[-2.73, 0.63]$	$[-2.28, 0.46]$	$[-12.39, 2.52]$

Table 4.40 lists all the comparisons of the performance metrics of CART and CART for symbolic data with interval-valued explanatory variables and an interval-valued response variable. Two different splitting measurements are considered. The first one is to use the lower and upper bounds method which uses the CART to predict the bounds of the intervals separately. The second one is to use the Mean Square Error (MSE) to measure the prediction of the response variable. Two metrics, Root Mean Square Error (RMSE) and the code running time, will be used to measure the performance of the models. Since RMSE is only calculated for numerical values, we record the RMSE for the lower bound and upper bound separately when the response variable is interval-valued. As a result, there are two RMSE values for CART for symbolic data methods. The first value represents the RMSE of the lower bound and the second one means the RMSE for the upper bound. It is obvious that the RMSE of the numerical response variable in CART and the RMSE of the two bounds of the interval-valued response variable in CART for symbolic data are similar. The performance of CART for symbolic data using the lower-upper method is slightly worse than that of CART for symbolic data using MSE. However, the running time of CART for symbolic data using MSE is much larger than the CART for symbolic data using the lower-upper method. The main reason is still the effectiveness of the R function. Therefore, another future work will be how to write a more effective function to solve the CART for symbolic data in R. In addition, the RMSE of the two bounds obtained by the two CART for symbolic data methods are very close to the RMSE of CART for classical data.

Table 4.40: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 1.

	RMSE	running time
CART	4.52	0.011
CART for SD (lower-upper)	4.77, 5.08	0.019
CART for SD (MSE)	3.87, 4.70	1.00

Situation 2

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $J = 1000$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 10000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.11, which is exactly the same as for Situation 1. The mean values are the same for the observations in one group and are different for observations from different groups. After generating the values for the four explanatory variables, we can assume the numerical output is from the same linear regression as in Equation 4.12 in Situation 1. Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 1000$ groups after aggregation.

Table 4.41 lists the RMSE and the running time metrics of CART and CART for symbolic data with interval-valued explanatory variables and an interval-valued response variable. Two different splitting measurements are considered, the lower and upper bounds method and the MSE method. We will use RMSE and the code running time to evaluate the models. We record the RMSE for the lower bound and upper bound separately when the response variable is interval-valued. Since RMSE is only calculated for numerical values, we record the RMSE for the lower bound and upper bound separately when the response variable is interval-valued. The first value represents the RMSE of the lower bound and the second one corresponds to the RMSE for the upper bound. We can conclude that the RMSE of the numerical response variable in CART and the RMSE of two bounds of the interval-valued response variable in CART for symbolic data are similar. The performance of CART for symbolic data using the lower-upper method is slightly better than that of CART for symbolic data using MSE. What is more, the running time of CART for symbolic data using MSE is much larger than the CART for symbolic data using the lower-upper method. The main reason may be that using

different metrics to measure the division results will construct two kinds of tree structures, causing differences in predictions. Based on the conclusions of Situation 1 and Situation 2, we will only use CART for symbolic data using the lower-upper method for the remaining simulations due to its effectiveness.

Table 4.41: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 2.

	RMSE	running time
CART	4.16	0.049
CART for SD (lower-upper)	4.14, 4.14	0.032
CART for SD (MSE)	5.36, 5.54	12.59

Situation 3

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.11 where $\boldsymbol{\mu} = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ is generated from the normal distribution with mean 0 and variance 0.5. Compared with Situation 1 where the variance of each parameter in $\boldsymbol{\mu}$ is 1, here 0.5 will be used as the variance to make the observations from different groups closer to each other. The mean values are the same for the observations in one group and are different for observations from different groups. After generating the values for $J = 4$ explanatory variables, we can assume the numerical output is from the same linear regression as in Equation 4.12 as in Situation 1. Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation.

Table 4.42 lists the RMSE and the running time metrics of CART for the classical data with numerical response variable and CART for the symbolic data with interval-valued

explanatory variables and an interval-valued response variable. Since RMSE is only calculated for numerical values, we record the RMSE for the lower bound and upper bound separately when the response variable is interval-valued. The first value represents the RMSE of the lower bound and the second one represents the RMSE for the upper bound. It is obvious that the RMSE of the numerical response variable in CART and the RMSE of two bounds of the interval-valued response variable in CART for symbolic data are very similar.

Table 4.42: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 3.

	RMSE	running time
CART	3.31	0.01
CART for SD (lower-upper)	4.08, 4.41	0.01

Situation 4

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ classical observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.11 where $\boldsymbol{\mu} = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ is generated from the normal distribution with mean 0 and variance 5. Compared with Situation 1 where the variance of each parameter in $\boldsymbol{\mu}$ is 1, here, 5 will be used as the variance to make the observations from different groups easier to distinguish from each other. The mean values are the same for the observations in one group and are different for observations from different groups. After generating the values for $J = 4$ explanatory variables, we can assume the numerical output is from the same linear regression as in Equation 4.12 as in Situation 1. Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation.

Table 4.43 lists the RMSE and the running time metrics of CART and CART for symbolic data with interval-valued explanatory variables and interval-valued response variable. It is obvious that the RMSE of the numerical response variable in CART is smaller than the RMSE of the two bounds of the interval-valued response variable in CART for symbolic data. The main reason is that there is some information lost when aggregating the data, especially when the variance of the value of the explanatory variables increases.

Table 4.43: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 4.

	RMSE	running time
CART	14.26	0.008
CART for SD (lower-upper)	19.3, 19.78	0.009

Situation 5

Suppose we would like to generate a data frame with $J = 20$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.13. Here, we use a diagonal matrix with elements equal to one since the scales for the explanatory variables are the same for this situation. In addition, we assume no correlation between all the explanatory variables in this situation. All the assumptions with the variance are the same except for the dimension. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 20$ continuous explanatory variables, i.e.,

$$\{X_{i1k}, \ldots, X_{i,20,k}\} \sim N_{20}\left(\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \\ \vdots \\ \mu_{i,20} \end{pmatrix}, \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix}\right), i = 1, \ldots, I, k = 1, \ldots, K, \tag{4.13}$$

where $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \ldots, \mu_{i,20})'$ is generated from the normal distribution with mean 0 and variance 1. The mean values are the same for the observations in one group and are different for observations from different groups. After generating the values for $J = 20$ explanatory variables, we can assume the numerical output is from the linear regression

$$y_{ik} = \beta_0 + \beta_1 X_{i1k} + \cdots + \beta_{20} X_{i,20,k} + \epsilon_{ik}, \quad i = 1, \ldots, I, \quad k = 1, \ldots, K, \tag{4.14}$$

where β_0 is assumed to be 1, $\beta_j = j$ for the j^{th} explanatory variable. Here, we choose $\beta_j = j$ because we would like to simulate the situation where different explanatory variables have different influences on the response variable. We can also choose other values. The ϵ_{ik} is the random error for the k^{th} in the i^{th} group, which is normally distributed with mean 0 and variance 1.

Table 4.44 lists the RMSE and the running time metrics of CART and CART for symbolic data with interval-valued explanatory variables and interval-valued response variable. The first value for RMSE in the CART for SD (lower-upper) represents the RMSE of the lower bound and the second one corresponds to the RMSE for the upper bound. We can conclude that the RMSE of the numerical response variable in CART and the RMSE of the upper bound of the interval-valued response variable in CART for symbolic data are very close, while the RMSE of the lower bound of the interval-valued response variable in CART for

symbolic data is larger than the other two RMSE values. The main reason is that there is some information lost when aggregating the data.

Table 4.44: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 5.

	RMSE	running time
CART	65.79	0.072
CART for SD (lower-upper)	75.79, 64.64	0.081

Situation 6

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.15. Here we use a diagonal matrix with different diagonal elements since the scales for explanatory variables are assumed to be different for this situation and also we assume no correlation between all the explanatory variables. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables, i.e.,

$$\{X_{i1k}, \ldots, X_{i4k}\} \sim N_4 \left(\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \\ \mu_{i3} \\ \mu_{i4} \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix} \right), i = 1, \ldots, I, k = 1, \ldots, K, \tag{4.15}$$

where $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ is generated from the normal distribution with mean 0 and variance 1. The mean values are the same for the observations in one group and are different for observations from different groups. After generating the values for $J = 4$ explanatory

variables, we can assume the numerical output is from the same linear regression as in Equation 4.12 as in Situation 1. Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation.

Table 4.45 lists the RMSE and the running time metrics of CART and CART for symbolic data with interval-valued explanatory variables and interval-valued response variable. The first value for RMSE in the CART for SD (lower-upper) represents the RMSE of the lower bound and the second one corresponds to the RMSE for the upper bound. It is obvious that the RMSE of the numerical response variable in CART and the RMSE of the two bounds of the interval-valued response variable in CART for symbolic data are similar.

Table 4.45: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 6.

	RMSE	running time
CART	15.47	0.009
CART for SD (lower-upper)	16.87, 15.14	0.019

Situation 7

Suppose we would like to generate a data frame with $J = 4$ interval-valued explanatory variables and $I = 100$ groups of data with $K = 10$ observations in each group. To simulate this data set, $I \times K = 1000$ classical observations are randomly drawn from the multivariate normal distribution of Equation 4.16. Here, we use a non-diagonal matrix with diagonals value equal to one since the scales for explanatory variables are assumed to be the same for this situation and also we assume the explanatory variables are positively correlated. Then, we can obtain a classical data set with $I \times K = 1000$ observations and $J = 4$ continuous explanatory variables, i.e.,

$$\{X_{i1k},\ldots,X_{i4k}\} \sim N_4\left(\begin{pmatrix}\mu_{i1}\\ \mu_{i2}\\ \mu_{i3}\\ \mu_{i4}\end{pmatrix}, \begin{pmatrix}1 & 0.8 & 0.5 & 0.2\\ 0.8 & 1 & 0.8 & 0.5\\ 0.5 & 0.8 & 1 & 0.8\\ 0.2 & 0.5 & 0.8 & 1\end{pmatrix}\right), i = 1,\ldots,I, k = 1,\ldots,K, \tag{4.16}$$

where $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4})'$ is generated from the normal distribution with mean 0 and variance 1. The mean values are the same for the observations in one group and are different for observations from different groups. After generating the values for $J = 4$ explanatory variables, we can assume the numerical output is from the same linear regression as in Equation 4.12 as in Situation 1. Suppose every $K = 10$ consecutive observations are in one group and can be aggregated into an interval-valued object. Thus, we can obtain a data set that consists of $J = 4$ interval-valued explanatory variables. There will be $I = 100$ groups after aggregation.

Table 4.46 lists the RMSE and the running time metrics of CART and CART for symbolic data with interval-valued explanatory variables and interval-valued response variable. The first value for RMSE in the CART for SD (lower-upper) represents the RMSE of the lower bound and the second one means the RMSE for the upper bound. It is obvious that the RMSE of the numerical response variable in CART and the RMSE of two bounds of the interval-valued response variable in CART for symbolic data are similar.

Table 4.46: Comparison of the performances of CART and CART for symbolic data (SD) with modal-valued explanatory variables in Scenario 4 under Situation 7.

	RMSE	running time
CART	4.90	0.01
CART for SD (lower-upper)	5.78, 5.37	0.01

Comparison

The scale of the response variable will be changed if we have different assumptions on the parameters. For example, when adjusting the variance of the mean $\boldsymbol{\mu}_i$ in Situation 3 and Situation 4 to control the distance between different groups, the scale of the response variable will be larger if we choose a larger variance, given the mean is fixed. Another example is the number of explanatory variables. Since we assume the same linear regression for all situations, the response variable will be larger if we obtain a regression with more explanatory variables. What is more, the different scales of the explanatory variables as in Situation 6 will also greatly influence the scale of the response variable. As a result, it is not appropriate to compare these situations together. Based on the conclusion for each situation, we find that the prediction for intervals is very similar to the prediction for numerical values. Although the sample size will be reduced by grouping the data, the running time for symbolic data is increasing because we need to predict two values in the CART for symbolic data.

Chapter 5

Conclusion and Future Work

Symbolic data are very common in our daily life, while the analytic methods for symbolic data are very limited. For example, a famous and useful method called decision tree is widely used. However, decision tree methods are only useful for classic data taking a single value. As a result, we cannot apply the algorithms on data sets with symbolic explanatory variables or a symbolic response variable. In this dissertation, we propose a method to build a tree-based model when analyzing a data set with symbolic variables. For different types of symbolic data, especially modal-valued and interval-valued, we have discussed the situation of these symbolic data as explanatory variables and response variable, respectively. To build a tree-based model, we need to consider two important parts. The first one is how to divide the data set according to the explanatory variable. The other part is how to evaluate the result of the division according to the response variable. Therefore, we have proposed a variety of methods for dividing data and evaluating the division.

Furthermore, we consider different scenarios containing symbolic data and use two real-life data sets and several simulated data sets to illustrate our methods. According to the real-life data analysis, we found that CART for symbolic data greatly improved the efficiency of the algorithm and reduced the running time for the categorical response variable. This new method solved how to handle a large amount of data in big data analysis. What

is more, we find that the classification and regression tree (CART) for symbolic data is very useful for a data set with inconsistent dimensions as in the electric charging example. Generally speaking, CART for symbolic data can reduce the running time by reducing the sample size and keeps the prediction effect similar to the original data as adjudged by some performance measurements such as recall, precision, and accuracy. In Section 4.4.2, we conducted several simulations to show that it is also possible to use CART for symbolic data to predict classification problems with classical data with less running time. For an interval-valued response variable, because we need to predict two values, either lower bound and upper bound or range and center, the running time will be relatively larger. The only problem of CART for symbolic data is that the sample size is reduced after aggregating the data, making the prediction less stable.

Therefore, the CART for symbolic data method is very suitable for the following situations in life. The first is when the data we collect contain symbolic data such as questionnaires. In this case, traditional statistical models and machine learning methods cannot solve such problems. Secondly, when the number of our data samples is too large and the response variable is categorical, we can greatly reduce the running time by aggregating the data.

In this dissertation, we proposed that the clustering method for symbolic data can be used to divide the data set, especially for histogram data. Modal multi-valued and interval-valued are relatively simple and straightforward to divide; and histogram-valued data are more difficult to divide because of its more complex structure. In addition, there are some clustering methods published for symbolic data. As a result, an extended work can be developed using clustering methods to check and compare their performance. What is more, we can use the similarity and distance functions to measure the split as mentioned in Section 3.1. However, only Root Mean Square Error (RMSE) was used for a modal multi-valued response variable with two possible values and interval-valued response variable. One future work is to simulate and compare the prediction using similarity or distance. Lastly, modal

multi-valued data and interval-valued data are simulated to evaluate the methods for the simulation study. More simulations for histogram-valued data are under discovery. Moreover, we only use the Bernoulli distribution for categorical generation and the Normal distribution for numerical generation. To simulate more real-life situations, we can simulate classical data by other kinds of distribution such as a Poisson distribution and an Exponential distribution.

Another potential problem is that using Mean Square Error (MSE) as the standard function for evaluating splitting is inefficient, so the algorithm needs to be improved. In addition, functions based on similarities and distances are under discovery for splitting measurement. What is more, we can compare the methods with different splitting measurements on the same data set.

Furthermore, there are many extended works for classical CART, e.g., tree pruning, and ensemble methods based on the tree model. A disadvantage of decision trees is overfitting, especially when the tree structure is large. The reason is that we will end up with very few instances on each leaf node of the tree if the tree becomes too large. As a result, the estimated average value on each child node will be inaccurate. Therefore, it is essential to prune the tree for a good model. For classical CART, we have post-pruning and pre-pruning. One future work is to develop pruning CART for symbolic data. What is more, many ensemble methods such as random forest and XGBoost based on a simple decision tree structure are proposed. In the future, we can also investigate some ensemble methods based on the CART for symbolic data method. Finally, we only use really data and simulations to check and compare the performance of the CART for symbolic data. In the future, we can consider some discussion for mathematical justification for the performance.

Bibliography

Anderson, E. (1935). The irises of the gaspe peninsula. *Bull. Am. Iris Soc.*, *59*, 2–5.

Bertrand, P., & Goupil, F. (2000). Descriptive statistics for symbolic data. *Analysis of Symbolic Data*, 106–124.

Billard, L., & Diday, E. (2006). Symbolic data analysis: Conceptual statistics and data mining.

Billard, L. (2007). Dependencies and variation components of symbolic interval-valued data. *Selected Contributions in Data Analysis and Classification*, 3–12.

Billard, L. (2008). Sample covariance functions for complex quantitative data. *Proceedings of World IASC Conference, Yokohama, Japan*, 157–163.

Billard, L., & Diday, E. (2000). Regression analysis for interval-valued data. *Data Analysis, Classification, and Related Methods*, 369–374.

Billard, L., & Diday, E. (2002). Symbolic regression analysis. *Classification, Clustering, and Data Analysis*, 281–288.

Billard, L., & Diday, E. (2019). *Clustering methodology for symbolic data*. John Wiley & Sons.

Breiman, L. (1997). *Arcing the edge* (tech. rep.). Technical Report 486, Statistics Department, University of California at Berkeley.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Brito, P., & Chavent, M. (2012). Divisive monothetic clustering for interval and histogram-valued data. *ICPRAM 2012-1st International Conference on Pattern Recognition Applications and Methods*, 229–234.

Chavent, M. (1998). A monothetic clustering method. *Pattern Recognition Letters*, *19*(11), 989–996.

Chavent, M. (2000). Criterion-based divisive clustering for symbolic data. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*, 299–311.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chen, Y. (2014). *Symbolic data regression and clustering methods* (Doctoral dissertation). University of Georgia.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2), 224–227.

de Carvalho. (1994). Proximity coefficients between boolean symbolic objects. *New Approaches in Classification and Data Analysis*, 387–394.

de Carvalho. (1998). Extension based proximities between constrained boolean symbolic objects. *Data Science, Classification, and Related Methods*, 370–378.

de Carvalho, T., F. A., Neto, E. D. A. L., & Tenorio, C. P. (2004). A new method to fit a linear regression model for interval-valued data. *Annual Conference on Artificial Intelligence*, 295–306.

Diday, E. (1987). *Introduction à l'approche symbolique de l'analyse des données*. CEREMADE, Université de Paris, Dauphine, 21-56.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, *4*(1), 95–104.

El-Sonbaty, Y., & Ismail, M. A. (1998). Fuzzy clustering for symbolic data. *IEEE Transactions on Fuzzy Systems*, *6*(2), 195–204.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Gowda, K. C., & Diday, E. (1991). Symbolic clustering using a new dissimilarity measure. *Pattern Recognition*, *24*(6), 567–578.

Gowda, K. C., & Diday, E. (1992). Symbolic clustering using a new similarity measure. *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(2), 368–378.

Guru, D., & Kiranagi, B. B. (2005). Multi-valued type dissimilarity measure and concept of mutual dissimilarity value for clustering symbolic patterns. *Pattern Recognition*, *38*(1), 151–156.

Guru, D., Kiranagi, B. B., & Nagabhushan, P. (2004). Multivalued type proximity measure and concept of mutual similarity value useful for clustering symbolic patterns. *Pattern Recognition Letters*, *25*(10), 1203–1213.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282.

Ichino, M., & Yaguchi, H. (1994). Generalized minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*(4), 698–708.

Irpino, A., & Verde, R. (2006). A new wasserstein based distance for the hierarchical clustering of histogram symbolic data. *Data Science and Classification*, 185–192.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*, 3146–3154.

Kim, J. (2009). *Dissimilarity measures for histogram-valued data and divisive clustering of symbolic objects* (Doctoral dissertation). University of Georgia.

Kim, J., & Billard, L. (2013). Dissimilarity measures for histogram-valued observations. *Communications in Statistics-Theory and Methods*, *42*(2), 283–303.

Kim, J., & Billard, L. (2018). Double monothetic clustering for histogram-valued data. *Communications for Statistical Applications and Methods*, *25*(3), 263–274.

Kim, J., & Billard, L. (2011). A polythetic clustering process and cluster validity indexes for histogram-valued objects. *Computational Statistics & Data Analysis*, *55*(7), 2250–2262.

Kim, J., & Billard, L. (2012). Dissimilarity measures and divisive clustering for symbolic multi modal-valued data. *Computational Statistics & Data Analysis*, *56*(9), 2795–2808.

Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). Boosting algorithms as gradient descent in function space.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc.

Ridgeway, G. (2007). Generalized boosted models: A guide to the gbm package. *Update*, *1*(1), 2007.

Xu, W. (2010). *Symbolic data analysis: Interval-valued data regression* (Doctoral dissertation). University of Georgia.

Zhu, J. (2019). *Divisive hierarchical clustering for interval-valued data* (Doctoral dissertation). University of Georgia.

Appendix A

Code

A.1 Python code

```
## Packages
import pandas as pd
import numpy as np
from numpy import mean
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectKBest
from sklearn.model_selection import learning_curve, GridSearchCV
from sklearn import tree
from sklearn.metrics import classification_report
from sklearn import metrics
```

```
from imblearn.pipeline import Pipeline
import seaborn as sns
from pmdarima.arima import auto_arima
from collections import Counter
from pandas.core.frame import DataFrame
import glob
import warnings
warnings.filterwarnings("ignore")

## Method 1: basic statistics
def stat_charging(data):
    e1 = data[['v','c','c_soc','max_temp','min_temp']]
    e1['max_temp'] = e1['max_temp'].apply(lambda x: x % 50)
    e1['min_temp'] = e1['min_temp'].apply(lambda x: x % 50)
    result = []
    result += e1.mean().tolist()
    result += e1.std().tolist()
    result += e1.median().tolist()
    result.append(0)
    return result

# grab excel files only
pattern_fault = '/Users/zouwanxue/Downloads/fault data/*.csv'
csv_files_fault = glob.glob(pattern_fault)
l = []
for file in csv_files_fault:
    #  Read xlsx into a DataFrame
    df = pd.read_csv(file)
```

```
    # Append df to frames
    l.append(df.shape[0])
print(min(l),max(l))


pattern_normal = '/Users/zouwanxue/Downloads/normal data/*/*.csv'
csv_files_normal = glob.glob(pattern_normal)
# Check the range of charging length
l = []
for file in csv_files_normal:
    df = pd.read_csv(file)
    l.append(df.shape[0])
print(min(l),max(l))


frames = []
for file in csv_files_normal:
    #  Read xlsx into a DataFrame
    df = pd.read_csv(file)
    # Append df to frames
    if df.shape[0] > 30:
        frames.append(stat_charging(df) + [0])
for file in csv_files_fault:
    df = pd.read_csv(file)
    frames.append(stat_charging(df) + [1])


df = pd.DataFrame(frames)
df.head()
df.info()
# Check whether it is a balanced data set or not.
```

```
counter = Counter(df.loc[:,24])
print(counter) # ratio (0:0.86,1:0.14)

# Transform the data
x = df.iloc[:,:-1].values
x = StandardScaler().fit_transform(x)
y = df.loc[:,16].values
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33,
    random_state=42)

# Find the best parameters.
w = {0:0.14,1:0.86}
pipe=Pipeline([('select',SelectKBest(k=10)),
              ('classify', tree.DecisionTreeClassifier(random_state = 10,
                  max_features = 'sqrt'))])
param_test = {'classify__max_depth':list(range(1,10,2)),
             'classify__min_samples_split':list(range(1,10,2)),
            'classify__class_weight':[w]}
gsearch = GridSearchCV(estimator = pipe, param_grid = param_test,
    scoring='f1_macro', cv=5)
gsearch.fit(X_train,y_train)
print(gsearch.best_params_, gsearch.best_score_)

# Fit the model with best parameters.
DT = tree.DecisionTreeClassifier(class_weight = w, max_depth = 9,
    min_samples_split = 5)
print(DT.fit(X_train, y_train))
print("accuracy on testing: ", DT.score(X_test, y_test))
```

```
X_train_pred = DT.predict(X_train)
X_test_pred = DT.predict(X_test)

# Get the prediction results
print('training:')
print(classification_report(y_train, X_train_pred))
print('confusion matrix on traning:')
print(metrics.confusion_matrix(y_train, X_train_pred))
print('testing:')
print(classification_report(y_test, X_test_pred))
print('confusion matrix on testing:')
print(metrics.confusion_matrix(y_test, X_test_pred))

## Method 2: filling data by ARIMA
def arima_charging(data):
    e1 = data[['v','c','c_soc','max_temp','min_temp']]
    e1['max_temp'] = e1['max_temp'].apply(lambda x: x % 50)
    e1['min_temp'] = e1['min_temp'].apply(lambda x: x % 50)
    n = e1.shape[0]
    e1 = e1.loc[1:(n-1),:]
    soc = e1.c_soc
    col_list = ['v','c','max_temp','min_temp']
    res = []
    for col in col_list:
        l = e1[col]
        start = 0
        end = 100
        tmp = {'l' : l, 'soc' : soc}
```

```
data = DataFrame(tmp)
df_temp = data[['soc']]
df_temp.drop_duplicates(inplace=True)
data = data.iloc[df_temp.index.tolist(), :].dropna()
data.set_index(['soc'], drop=False, inplace=True)
curve = data.iloc[:, 0]
train1, train2 = curve, curve[::-1]
train1, train2 = np.transpose(np.array([train1])),
    np.transpose(np.array([train2]))
soc_list = data.index.tolist()
length_1, length_2 = end - soc_list[-1], soc_list[0] - start
model1 = auto_arima(train1, start_p=1, start_q=1, max_p=5, max_q=5, m=12,
                    start_P=0, seasonal=True, d=1, D=1,
                    trace=True, error_action='ignore',
                        suppress_warnings=True)
model1.fit(train1)
pre1 = model1.predict(n_periods=int(length_1))
model2 = auto_arima(train2, start_p=1, start_q=1, max_p=5, max_q=5,
                    m=12, start_P=0, seasonal=True, d=1, D=1,
                    trace=True, error_action='ignore',
                        suppress_warnings=True)
model2.fit(train2)
pre2 = model2.predict(n_periods=int(length_2))
curve = (pre2.tolist())[::-1] + train1.T[0].tolist() + pre1.tolist()
if len(curve) > 100:
    curve = curve[:100]
elif len(curve) < 100:
    curve = curve + [0]*(100-len(curve))
```

```
        res += curve
    return res

frames = []
for file in csv_files_normal:
    #  Read xlsx into a DataFrame
    df = pd.read_csv(file)
    # Append df to frames
    try:
        frames.append(arima_charging(df) + [0])
    except:
        pass
for file in csv_files_fault:
    df = pd.read_csv(file)
    try:
        frames.append(arima_charging(df) + [1])
    except:
        pass

df = pd.DataFrame(frames)
df.head()
df.info()
counter = Counter(df.loc[:,24])
print(counter)

# Transform the data
x = df.iloc[:,:-1].values
x = StandardScaler().fit_transform(x)
```

```
y = df.loc[:,500].values
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33,
    random_state=42)

# Find the best parameters.
w = {0:0.14,1:0.86}
pipe=Pipeline([('select',SelectKBest(k=10)),
               ('classify', tree.DecisionTreeClassifier(random_state = 10,
                   max_features = 'sqrt'))])
param_test = {'classify__max_depth':list(range(1,10,2)),
              'classify__min_samples_split':list(range(1,10,2)),
             'classify__class_weight':[w]}
gsearch = GridSearchCV(estimator = pipe, param_grid = param_test,
    scoring='f1_macro', cv=5)
gsearch.fit(X_train,y_train)
print(gsearch.best_params_, gsearch.best_score_)

# Fit the model with best parameters.
DT = tree.DecisionTreeClassifier(class_weight = w, max_depth = 9,
    min_samples_split = 5)
print(DT.fit(X_train, y_train))
print("accuracy on testing: ", DT.score(X_test, y_test))
X_train_pred = DT.predict(X_train)
X_test_pred = DT.predict(X_test)

# Get the prediction results
print('training:')
print(classification_report(y_train, X_train_pred))
```

```
print('confusion matrix on traning:')
print(metrics.confusion_matrix(y_train, X_train_pred))
print('testing:')
print(classification_report(y_test, X_test_pred))
print('confusion matrix on testing:')
print(metrics.confusion_matrix(y_test, X_test_pred))

## Method 3: Histogram data
def hist_charging(data):
    e1 =data[['v','c','c_soc','max_temp','min_temp']]
    if max(e1.c_soc) > 100:
        e1['c_soc'] = e1['c_soc']/10
    e1['max_temp'] = e1['max_temp'].apply(lambda x: x % 50)
    e1['min_temp'] = e1['min_temp'].apply(lambda x: x % 50)
    e1['group'] = e1['c_soc'].apply(lambda x:split(x,10))
    for i in range(21):
        if i not in set(e1.group):
            e1 = e1.append({'v':0,'c':0,'c_soc':0,
                  'max_temp':0,'min_temp':0,'group':i}, ignore_index=True)
    e2 = e1.groupby('group').agg({ 'v':['mean'], 'c':['mean'],
        'max_temp':['mean'],'min_temp':['mean']})
    d = []
    M = e2.to_numpy()
    for i in M:
        for j in i:
            d.append(j)
    d.append(0)
    return d
```

```
frames = []
for file in csv_files_normal:
    #  Read xlsx into a DataFrame
    df = pd.read_csv(file)
    # Append df to frames
    if df.shape[0] > 30:
        frames.append(hist_charging(df) + [0])
for file in csv_files_fault:
    df = pd.read_csv(file)
    frames.append(hist_charging(df) + [1])

df = pd.DataFrame(frames)
df.head()
df.info()
counter = Counter(df.loc[:,24])
print(counter)

# Transform the data
x = df.iloc[:,:-1].values
x = StandardScaler().fit_transform(x)
y = df.loc[:,16].values
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33,
    random_state=42)

# Find the best parameters
w = {0:0.14,1:0.86}
pipe=Pipeline([('select',SelectKBest(k=10)),
```

```
            ('classify', tree.DecisionTreeClassifier(random_state = 10,
                max_features = 'sqrt'))])
param_test = {'classify__max_depth':list(range(1,10,2)),
            'classify__min_samples_split':list(range(1,10,2)),
           'classify__class_weight':[w]}
gsearch = GridSearchCV(estimator = pipe, param_grid = param_test,
   scoring='f1_macro', cv=5)
gsearch.fit(X_train,y_train)
print(gsearch.best_params_, gsearch.best_score_)

# Fit the model with best parameters.
DT = tree.DecisionTreeClassifier(class_weight = w, max_depth = 9,
   min_samples_split = 5)
print(DT.fit(X_train, y_train))
print("accuracy on testing: ", DT.score(X_test, y_test))
X_train_pred = DT.predict(X_train)
X_test_pred = DT.predict(X_test)

# Get the prediction results.
print('training:')
print(classification_report(y_train, X_train_pred))
print('confusion matrix on traning:')
print(metrics.confusion_matrix(y_train, X_train_pred))
print('testing:')
print(classification_report(y_test, X_test_pred))
print('confusion matrix on testing:')
print(metrics.confusion_matrix(y_test, X_test_pred))
```

A.2 R code

```
require(ggplot2)
require(rpart)
require(rpart.plot)
require(caret)
require(MASS)
require(purrr)
require(tree)

### 0. Functions
# Function to generate binary data
generatebern <- function(n,m,P){
  # n means the number of groups
  # m means the number of samples in each group
  # P is p-dimension vector for probabilies in the class
  res = NULL
  p = length(P)
  for (j in 1:p){
    tmp = rbernoulli(n*m,p = P[j])
    res = cbind(res,tmp)
  }
  return (res)
}

# Function to generate numerical data
generatenormal <- function(n,m,mu,v,M){
  # n means the number of groups
```

```
  # m means the number of samples in each group
  # M is the covariance matrix
  # The output should be a dataframe with (mn)*p where p is the dimension of M.
  res = NULL
  p = dim(M)[1]
  for (i in 1:n){
    tmp_u = rnorm(p,mu,v)
    tmp = mvrnorm(n = m, mu = tmp_u, Sigma = M)
    res = rbind(res,tmp)
  }
  return (res)
}

# Function to aggregate the categorical explanatory variables
# with aggregated categorical response
aggregate_modal <- function(classic,m){
  # m means the number of samples in one group
  n = nrow(classic)
  p = ncol(classic)-1
  group = ceiling(n/m)
  df_modal = NULL
  for (i in 1:group){
    tmp = (m*i-m+1):(m*i)
    df_tmp = classic[tmp,]
    row_tmp = NULL
    for (j in 1:p){
      row_tmp = c(row_tmp, sum(df_tmp[,j])/m)
    }
```

```
    df_modal = rbind(df_modal,row_tmp)
  }
  df = data.frame(df_modal)
  return (df)
}

# Function to aggregate the categorical explanatory variables
# with aggregated modal multi-valued or interval-valued response
aggregate_modal2 <- function(classic,m,type){
  # m means the number of samples in one group
  n = nrow(classic)
  p = ncol(classic)-1
  group = ceiling(n/m)
  df_modal = NULL
  for (i in 1:group){
    tmp = (m*i-m+1):(m*i)
    df_tmp = classic[tmp,]
    row_tmp = NULL
    for (j in 1:p){
      row_tmp = c(row_tmp, sum(df_tmp[,j])/m)
    }
    if (type == 'modal'){
      row_tmp = c(row_tmp, sum(df_tmp[,(p+1)])/m)
    }
    else{
      row_tmp = c(row_tmp, min(df_tmp[,(p+1)]), max(df_tmp[,(p+1)]))
    }
    df_modal = rbind(df_modal,row_tmp)
```

```
  }
  df = data.frame(df_modal)
  return (df)
}

# Function to aggregate the numerical explanatory variables
# with aggregated categorical response
aggregate_interval <- function(classic,m){
  # m means the number of samples in one group
  n = nrow(classic)
  p = ncol(classic)-1
  group = ceiling(n/m)
  df_interval = NULL
  for (i in 1:group){
    tmp = (m*i-m+1):(m*i)
    df_tmp = classic[tmp,]
    row_tmp = NULL
    for (j in 1:p){
      row_tmp = c(row_tmp, min(df_tmp[,j]))
      row_tmp = c(row_tmp, max(df_tmp[,j]))
    }
    df_interval = rbind(df_interval,row_tmp)
  }
  df = data.frame(df_interval)
  return (df)
}

# Function to aggregate the numerical explanatory variables
```

```
# with aggregated modal multi-valued or interval-valued response
aggregate_interval2 <- function(classic,m,type){
  # m means the number of samples in one group
  n = nrow(classic)
  p = ncol(classic)-1
  group = ceiling(n/m)
  df_interval = NULL
  for (i in 1:group){
    tmp = (m*i-m+1):(m*i)
    df_tmp = classic[tmp,]
    row_tmp = NULL
    for (j in 1:p){
      row_tmp = c(row_tmp, min(df_tmp[,j]))
      row_tmp = c(row_tmp, max(df_tmp[,j]))
    }
    if (type == 'modal'){
      row_tmp = c(row_tmp, sum(df_tmp[,(p+1)])/m)
    }
    else{
      row_tmp = c(row_tmp, min(df_tmp[,(p+1)]), max(df_tmp[,(p+1)]))
    }
    df_interval = rbind(df_interval,row_tmp)
  }
  df = data.frame(df_interval)
  return (df)
}

# Function to calculate MSE
```

```
MSE <- function(intervals){
  # intervals is a matrix with n*2 dimention
  # each row represent an interval
  n = nrow(intervals)
  a = intervals[,1]
  b = intervals[,2]
  y = (a+b)/2
  Y = mean(y)
  Yvec = rep(Y,n)
  SST = sum((y - Yvec)^2) + sum((y - a)^2)/3 +
    sum((y - b)^2)/3 + sum((y - a) * (y - b))/3
  return (SST/n)
}

# Function to calculate the gini
gini <-function(v){
  if (length(v) == 0){
    return (0)
  }
  category <- unique(v)
  n <- length(v)
  res <- 1
  for (i in 1:length(category)){
    p <- length(v[v==category[i]])/n
    res <- res - p^2
  }
  return (res)
}
```

```
# Functions to split the data for interval explanatory variables
# and categorical response
tripart_best <- function(data){
  if (nrow(data) == 0){
    return (data)
  }
  p <- (ncol(data) -1)/2
  var <- 1:p
  delta_gini = NULL
  col_index = NULL
  mid_val = NULL
  for (i in var){
    vals <- unique(sort(c(data[,(2*i-1)],data[,2*i])))
    tmp_n <- length(vals)
    mid <- (vals[1:(tmp_n-1)]+vals[2:tmp_n])/2
    for (val in mid){
      D1 = data[data[,(2*i-1)] > val,]
      D2 = data[(data[,(2*i-1)] <= val) & (data[,2*i] > val),]
      D3 = data[data[,2*i] < val,]
      ginichange = gini(data[,ncol(data)]) -
        nrow(D1)/nrow(data)*gini(D1[,ncol(D1)]) -
        nrow(D2)/nrow(data)*gini(D2[,ncol(D2)]) -
        nrow(D3)/nrow(data)*gini(D3[,ncol(D3)])
      delta_gini <- c(delta_gini,ginichange)
      col_index <- c(col_index,i)
      mid_val <- c(mid_val, val)
    }
```

```
  }
  id = which.max(delta_gini)
  return (list(delta = delta_gini[id],
               new_delta = gini(data[,ncol(data)]) - delta_gini[id],
               col_index = col_index[id],
               mid_val = mid_val[id]))
}

# Functions to split the data for interval explanatory variables
# and interval-valued response
tripart_best_MSE <- function(data){
  if (nrow(data) == 0){
    return (data)
  }
  p <- ncol(data) -2
  var <- 1:p
  delta_MSE = NULL
  col_index = NULL
  mid_val = NULL
  for (i in var){
    vals <- unique(data[,i])
    tmp_n <- length(vals)
    mid <- (vals[1:(tmp_n-1)]+vals[2:tmp_n])/2
    for (val in mid){
      D1 = data[data[,i] >= val,]
      D2 = data[data[,i] < val,]
      MSEchange = MSE(data[,((p+1):(p+2))]) -
        nrow(D1)/nrow(data)*MSE(D1[,((p+1):(p+2))]) -
```

```
        nrow(D2)/nrow(data)*MSE(D2[,((p+1):(p+2))])
      delta_MSE <- c(delta_MSE,MSEchange)
      col_index <- c(col_index,i)
      mid_val <- c(mid_val, val)
    }
  }
  id = which.max(delta_MSE)
  return (list(delta = delta_MSE[id],
               new_delta = MSE(data[,((p+1):(p+2))]) - delta_MSE[id],
               col_index = col_index[id],
               mid_val = mid_val[id]))
}

# Functions to evaluate models
evaluation <- function(model, data, atype) {
  # This function is for classification with input models
  prediction = predict(model, data, type=atype)
  xtab = table(prediction, data$label)
  accuracy = sum(prediction == data$label)/length(data$label)
  precision = xtab[1,1]/sum(xtab[,1])
  recall = xtab[1,1]/sum(xtab[1,])
  f = 2 * (precision * recall) / (precision + recall)
  return (list(acc = accuracy,recall = recall, pre = precision, f = f))
}

compare <- function(l1,l2){
  # This function is for classification with input lists
  xtab = table(l1, l2)
```

```
  accuracy = sum(l1 == l2)/length(l1)
  precision = xtab[1,1]/sum(xtab[,1])
  recall = xtab[1,1]/sum(xtab[1,])
  f = 2 * (precision * recall) / (precision + recall)
  return (list(acc = accuracy,recall = recall, pre = precision, f = f))
}



### 1. Iris dataset
## 1.1 The original iris dataset
data("iris")
head(iris)
x <- iris[,1:4]
y <- iris[,5]
par(mfrow=c(2,2))
for(i in 1:4) {
  boxplot(x[,i], main=names(iris)[i])
}
# scatterplot matrix for classical data
featurePlot(x=x, y=y, plot="ellipse")

tree.model = rpart(Species ~ Sepal.Width + Petal.Width, data = iris,
    method='class')
plot(iris$Petal.Width, iris$Sepal.Width, pch=19, col=as.numeric(iris$Species))
partition.tree(tree.model, label="Species", add=TRUE)
legend("topright",legend=unique(iris$Species),
    col=unique(as.numeric(iris$Species)), pch=19)
```

```
## 1.2 Interval-valued iris dataset
iris_interval = aggregate_interval(iris,m=5)
iris_interval$Species = rep(c('setosa', 'versicolor',
                              'virginica'),each = 10)
colnames(iris_interval)<-c('SL_L','SL_U','SW_L','SW_U',
                           'PL_L','PL_U','PW_L','PW_U','Species')
rownames(iris_interval)<-paste('group',1:nrow(iris_interval))
head(iris_interval)

# Visualize the data
par(mfrow = c(2,3))
for (i in 1:3){
  for (j in (i+1):4){
    X_L = iris_interval[,(2*i-1)]
    X_U = iris_interval[,2*i]
    Y_L = iris_interval[,(2*j-1)]
    Y_U = iris_interval[,2*j]
    plot(c(min(X_L)-0.2,max(X_U)+0.2), c(min(Y_L)-0.2, max(Y_U)+0.2),
         type = "n", xlab = colnames(iris)[i], ylab = colnames(iris)[j],
         main = paste('Scatter Plots (', colnames(iris)[i],',',
                      colnames(iris)[j],')' ))
    rect(X_L[1:10], Y_L[1:10], X_U[1:10], Y_U[1:10], border = "blue")
    rect(X_L[11:20], Y_L[11:20], X_U[11:20], Y_U[11:20], border = "red")
    rect(X_L[21:30], Y_L[21:30], X_U[21:30], Y_U[21:30], border = "green")
  }
}

s <- sample(1:30,24,replace = FALSE)
```

```
train = iris_interval[s,]
test = iris_interval[-s,]
rtree <- rpart(Species ~ ., data = train, method='class')
printcp(rtree)
summary(rtree)
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
tripart_best(train)

### 2. Simulations
## For each scenario, only the codes for situation 1 are shown here.
## 2.1 Scenario 1
# Data generation
X1 = generatebern(50,10,c(0.1,0.2,0.3,0.4))
X2 = generatebern(50,10,c(0.4,0.5,0.6,0.7))
X = rbind(X1,X2)
sim11 = data.frame(X)
dim(sim11)
sim11$label = as.factor(rep(0:1,each = 500))

# CART for classical data
s <- sample(1:1000,700,replace = FALSE)
train = sim11[s,]
test = sim11[-s,]
rtree <- rpart(label ~ .,data = train, method='class')
system.time(rpart(label ~ .,data = train, method='class'))
printcp(rtree)
summary(rtree)
```

```
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
evaluation(rtree, test, "class")

# Run multiple times to get a stable estimator.
acc = NULL
recall = NULL
pre = NULL
f = NULL
for (i in 1:100){
  s <- sample(1:1000,700,replace = FALSE)
  train = sim11[s,]
  test = sim11[-s,]
  rtree <- rpart(label ~ .,data = train, method='class')
  acc <- c(acc,evaluation(rtree, test, "class")$acc)
  recall <- c(recall,evaluation(rtree, test, "class")$recall)
  pre <- c(pre,evaluation(rtree, test, "class")$pre)
  f <- c(f,evaluation(rtree, test, "class")$f)
}
c(mean(acc),var(acc))
c(mean(recall),var(recall))
c(mean(pre),var(pre))
c(mean(f),var(f))

# Convert the data to symbolic type
sim11_modal = aggregate_modal(sim11,m=10)
dim(sim11_modal)
sim11_modal$label = as.factor(rep(0:1,each = 50))
```

```
rownames(sim11_modal)<-paste('group',1:nrow(sim11_modal))
head(sim11_modal)
tail(sim11_modal)



# CART for symbolic data
s <- sample(1:100,70,replace = FALSE)
train = sim11_modal[s,]
test = sim11_modal[-s,]
rtree <- rpart(label ~ .,data = train, method='class')
system.time(rpart(label ~ .,data = train, method='class'))
printcp(rtree)
summary(rtree)
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
evaluation(rtree, test, "class")

# Run multiple times to get a stable estimator.
acc = NULL
recall = NULL
pre = NULL
f = NULL
for (i in 1:100){
  s <- sample(1:100,70,replace = FALSE)
  train = sim11_modal[s,]
  test = sim11_modal[-s,]
  rtree <- rpart(label ~ .,data = train, method='class')
  acc <- c(acc,evaluation(rtree, test, "class")$acc)
```

```
  recall <- c(recall,evaluation(rtree, test, "class")$recall)
  pre <- c(pre,evaluation(rtree, test, "class")$pre)
  f <- c(f,evaluation(rtree, test, "class")$f)
}
c(mean(acc),var(acc))
c(mean(recall),var(recall))
c(mean(pre),var(pre))
c(mean(f),var(f))

## 2.2 Scenario 2
# Data generation
X1 = generatebern(50,10,c(0.1,0.2,0.3,0.4))
X2 = generatebern(50,10,c(0.4,0.5,0.6,0.7))
X = rbind(X1,X2)
sim21 = data.frame(X)
dim(sim21)
l1 = sample(c(0,1),500,replace = TRUE, prob = c(0.8,0.2))
l2 = sample(c(0,1),500,replace = TRUE, prob = c(0.2,0.8))
sim21$label = c(l1,l2)

# CART for classical data
s <- sample(1:1000,700,replace = FALSE)
train = sim21[s,]
test = sim21[-s,]
rtree <- rpart(label ~ .,data = train, method='class')
system.time(rpart(label ~ .,data = train, method='class'))
printcp(rtree)
summary(rtree)
```

```
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
evaluation(rtree, test, "class")

# Run multiple times to get a stable estimator.
acc = NULL
recall = NULL
pre = NULL
f = NULL
for (i in 1:100){
  s <- sample(1:1000,700,replace = FALSE)
  train = sim21[s,]
  test = sim21[-s,]
  rtree <- rpart(label ~ .,data = train, method='class')
  acc <- c(acc,evaluation(rtree, test, "class")$acc)
  recall <- c(recall,evaluation(rtree, test, "class")$recall)
  pre <- c(pre,evaluation(rtree, test, "class")$pre)
  f <- c(f,evaluation(rtree, test, "class")$f)
}
c(mean(acc),var(acc))
c(mean(recall),var(recall))
c(mean(pre),var(pre))
c(mean(f),var(f))

# Convert the data to symbolic type
sim21_modal = aggregate_modal2(sim21,m=10,type = 'modal')
dim(sim21_modal)
rownames(sim21_modal)<-paste('group',1:nrow(sim11_modal))
```

```
head(sim21_modal)
tail(sim21_modal)

# Use the CART for SD to estimate the original class
s <- sample(1:100,70,replace = FALSE)
train = sim21_modal[s,]
test = sim21_modal[-s,]
so = NULL
for (k in s){
  so <- c(so,(10*k-9):(10*k))
}
test_original = sim21[-so,]
rtree <- rpart(X5 ~ .,data = train, method='anova')
system.time(rpart(X5 ~ .,data = train, method='anova'))
printcp(rtree)
summary(rtree)
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
pred <- predict(rtree,test)
res = NULL
for (i in pred){
  res <- c(res,sample(c(0,1),10,replace = TRUE, prob = c(1-i,i)))
}
compare(res,test_original$label)

# Run multiple times to get a stable estimator.
acc = NULL
recall = NULL
```

```
pre = NULL
f = NULL
for (i in 1:100){
  s <- sample(1:100,70,replace = FALSE)
  train = sim21_modal[s,]
  test = sim21_modal[-s,]
  so = NULL
  for (k in s){
    so <- c(so,(10*k-9):(10*k))
  }
  test_original = sim21[-so,]
  rtree <- rpart(X5 ~ .,data = train, method='anova')
  pred <- predict(rtree,test)
  res = NULL
  for (i in pred){
    res <- c(res,sample(c(0,1),10,replace = TRUE, prob = c(1-i,i)))
  }
  acc <- c(acc,compare(res,test_original$label)$acc)
  recall <- c(recall,compare(res,test_original$label)$recall)
  pre <- c(pre,compare(res,test_original$label)$pre)
  f <- c(f,compare(res,test_original$label)$f)
}
c(mean(acc),var(acc))
c(mean(recall),var(recall))
c(mean(pre),var(pre))
c(mean(f),var(f))

# Run multiple times to get a stable RMSE.
```

```
l = NULL
for (i in 1:100){
  s <- sample(1:100,70,replace = FALSE)
  train = sim21_modal[s,]
  test = sim21_modal[-s,]
  rtree <- rpart(X5 ~ .,data = train, method='anova')
  pred <- predict(rtree, test)
  l = c(l,RMSE(pred, test$X5))
}
mean(l)
var(l)

## 2.3 Scenario 3
# Data generation
X1 = generatenormal(50,10,0,1,diag(4))
X2 = generatenormal(50,10,1.5,1,diag(4))
X = rbind(X1,X2)
sim31 = data.frame(X)
dim(sim31)
sim31$label = as.factor(rep(0:1,each = 500))

# CART for classical data
s <- sample(1:1000,40,replace = FALSE)
featurePlot(x=sim31[s,1:4], y=sim31[s,5], plot="ellipse")
s <- sample(1:1000,700,replace = FALSE)
train = sim31[s,]
test = sim31[-s,]
rtree <- rpart(label ~ .,data = train, method='class')
```

```
system.time(rpart(label ~ .,data = train, method='class'))
printcp(rtree)
summary(rtree)

#Plots for numerical data
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
evaluation(rtree, test, "class")

# Convert the data to symbolic type
sim31_interval = aggregate_interval(sim31,m=10)
dim(sim31_interval)
sim31_interval$label = as.factor(rep(0:1,each = 50))
colnames(sim31_interval)<-c('X1_L','X1_U','X2_L','X2_U',
                        'X3_L','X3_U','X4_L','X4_U','label')
rownames(sim31_interval)<-paste('group',1:nrow(sim31_interval))
head(sim31_interval)
tail(sim31_interval)

#Plots for interval-valued data
group1 = sample(which(sim31_interval$label == 0),10,replace = FALSE)
group2 = sample(which(sim31_interval$label == 1),10,replace = FALSE)
par(mfrow = c(2,3))
for (i in 1:3){
  for (j in (i+1):4){
    X_L = sim31_interval[,(2*i-1)]
    X_U = sim31_interval[,2*i]
    Y_L = sim31_interval[,(2*j-1)]
```

```
    Y_U = sim31_interval[,2*j]
    plot(c(min(X_L)-0.2,max(X_U)+0.2), c(min(Y_L)-0.2, max(Y_U)+0.2),
         type = "n", xlab = colnames(sim31)[i], ylab = colnames(sim31)[j],
         main = paste('Scatter Plots (', colnames(sim31)[i],',',
                      colnames(sim31)[j],')' ))
    rect(X_L[group1], Y_L[group1], X_U[group1], Y_U[group1], border = "blue")
    rect(X_L[group2], Y_L[group2], X_U[group2], Y_U[group2], border = "red")
  }
}

# CART for symbolic data
s <- sample(1:100,70,replace = FALSE)
train = sim31_interval[s,]
test = sim31_interval[-s,]
rtree <- rpart(label ~ .,data = train, method='class')
system.time(rpart(label ~ .,data = train, method='class'))
printcp(rtree)
summary(rtree)
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
evaluation(rtree, test, "class")

## 2.4 Scenario 4
# Data generation
X = generatenormal(100,10,0,1,diag(4))
beta = c(1,1,2,3,4)
eta = rnorm(nrow(X))
y = cbind(rep(1,nrow(X)),X)%*%beta + eta
```

```
sim41 = data.frame(cbind(X,y))
dim(sim41)
head(sim41)
tail(sim41)

# CART for classical data
s <- sample(1:1000,700,replace = FALSE)
train = sim41[s,]
test = sim41[-s,]
rtree <- rpart(X5 ~ .,data = train, method='anova')
system.time(rpart(X5 ~ .,data = train, method='anova'))
printcp(rtree)
summary(rtree)
par(mfrow = c(1,1))
rpart.plot(rtree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
pred <- predict(rtree, test)
RMSE(pred, test$X5)

# Convert the data to symbolic type
sim41_interval = aggregate_interval2(sim41,m=10,type = 'interval')
dim(sim41_interval)
rownames(sim41_interval)<-paste('group',1:nrow(sim41_interval))
head(sim41_interval)
tail(sim41_interval)

# Run multiple times to get a stable estimator.
ll = NULL
lu = NULL
```

```
for (i in 1:100){
  s <- sample(1:100,70,replace = FALSE)
  train = sim41_interval[s,]
  test = sim41_interval[-s,]
  train1 = train[,1:9]
  train2 = train[,c(1:8,10)]
  test1 = test[,1:9]
  test2 = test[,c(1:8,10)]
  rtree1 <- rpart(X9 ~ .,data = train1, method='anova')
  pred1 <- predict(rtree1, test1)
  ll = c(ll,RMSE(pred1, test1$X9))
  rtree2 <- rpart(X10 ~ .,data = train2, method='anova')
  pred2 <- predict(rtree2, test2)
  lu = c(lu,RMSE(pred2, test2$X10))
}
mean(ll)
var(ll)
mean(lu)
var(lu)
```