8-2023

An Interval-Valued Random Forests

Paul Gaona Partida
Utah State University

AN INTERVAL-VALUED RANDOM FORESTS

by

Paul Gaona Partida

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Statistics

Approved:

Yan Sun, Ph.D. Major Professor	Brennan Bean, Ph.D. Committee Member
D. Richard Cutler, Ph.D. Committee Member	Janis L. Boettinger Senior Vice Provost

UTAH STATE UNIVERSITY
Logan, Utah

2023

ABSTRACT

An interval-valued random forests

by

Paul Gaona Partida, Master of Science

Utah State University, 2023

Major Professor: Yan Sun, Ph.D.
Department: Mathematics and Statistics

Analyzing soft interval data provides increased flexibility for analyzing data with varying degrees of quality and precision. Within this context, regression methods for interval data have been extensively studied. As most existing works focus on linear models, it is important to note that many problems in practice are nonlinear in nature, and the development of nonlinear regression tools for interval data is crucial.

We propose the Interval-Valued Random Forests (IRF) model, which introduces a novel splitting criterion based on variance reduction and an L_2-type metric in the Banach space of compact intervals. The IRF model takes into account both the centers and ranges of the interval, as well as their interactions. Unlike linear models that require additional constraints for mathematical coherence, our model estimates the regression function in a nonparametric way, naturally ensuring nonnegative interval lengths without constraints.

Simulation studies show that our method outperforms typical existing methods for various linear, semi-linear, and nonlinear data archetypes under different error measures. A real data example is presented to demonstrate the applicability, where the price intervals of the Dow Jones Industrial Average index and its component stocks are analyzed.

(72 pages)

PUBLIC ABSTRACT

An interval-valued random forests

Paul Gaona Partida

There is a growing demand for the development of new statistical models and the refinement of established methods to accommodate different data structures. This need arises from the recognition that traditional statistics often assume the value of each observation to be precise, which may not hold true in many real-world scenarios. Factors such as the collection process and technological advancements can introduce imprecision and uncertainty into the data.

For example, consider data collected over a long period of time, where newer measurement tools may offer greater accuracy and provide more information than previous methods. In such cases, it becomes crucial to restructure the data to account for imprecision and incorporate uncertainty into the analysis.

Furthermore, the increasing availability of large datasets has introduced computational challenges in analyzing and processing the data. Representing the data in terms of intervals can help address this uncertainty by reducing the data size or accommodating imprecision. Traditional methods have already embraced this concept, but given the rising popularity of machine learning, it is essential to develop models for interval-valued data within the machine learning framework.

Tree-based methods, in particular, are well-suited for handling interval-valued data due to their robustness to outliers and their nonparametric nature. Therefore, we propose a new model that takes into account the natural structure of the interval-valued data.. These tree-based methods offer improvements over existing models for interval-valued data, providing a framework capable of effectively handling data with uncertainty arising from imprecision or the need for size management.

To my beautiful family; Mom, Dad, Lesly, Julian, Angel, Cristina, and Andrea.

ACKNOWLEDGMENTS

Thank you, Dr. Sun; through your continuous support, mentorship, and patience, I have been able to develop a passion for research, and I cannot express my indebtedness enough. My appreciation is extended to my committee members, Drs. Brennan Bean and Richard Cutler, for going above and beyond the requirements. I greatly appreciate the time and attention you have dedicated to getting to know me on a personal level.

I want to extend sincere thanks to Drs. Luis Gordillo and Stephen Walsh. Your mentorship has been pivotal in my decision to pursue a Ph.D. I would also like to express my gratitude to Kenneth, KD, and ANSC 207. The laughs, conversations, and discussions we shared will not be forgotten, and I am proud to call you my friends.

Lastly, I would like to express my appreciation and gratitude to the Utah State University Department of Mathematics and Statistics for providing an enriching program, financial support, and an incredible staff. I could not have asked for a better experience.

Paul

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ACRONYMS

BA	Boeing Co
CCRM	Constrained Center and Range Method
CART	Classification And Regression Tree
CM	Center Method
C-R^2	Comprehensive Coefficient of Determination
CRM	Center and Range Method
GE	General Electric Company
IRF	Interval-valued Random Forests
IRT	Interval-valued Regression Tree
JPM	JPMorgan Chase & Co
MAE	Mean Absolute Error
MSE	Mean Square Error
MSFT	Microsoft Corp
PG	Procter & Gamble Co
RF	Random Forests
R^2	Coefficient of Determination
SDA	Symbolic Data Analysis
SSD	Sum of Squared Difference
TS-MAE	Total Standardized Mean Absolute Error
TS-MSE	Total Standardized Mean Square Error

CHAPTER 1

INTRODUCTION

The development of traditional statistical and machine learning methods has primarily assumed that data is collected in the form of single numbers or points. However, this assumption has been observed to be increasingly insufficient or unrealistic in many real-world situations. In today's world, it is not uncommon to encounter data that is measured or observed with imprecision or consists of a massive number of observations [1]. The lack of precision in data can be attributed to various limitations in measurement tools or observer biases [2]. Moreover, the prevalence of massive datasets leading to ultra computational complexity is a consequence of the reduced cost of data creation and storage [3].

To address the issue of imprecision and to summarize data into a more manageable size, alternative methods need to be developed and considered. These methods aim to account for imprecision in measurements or provide meaningful summaries of the data. By doing so, researchers and practitioners can obtain valuable insights and make reliable inferences from the data.

An example using daily weather temperature suggests that, instead of registering temperature solely with minimum, maximum, or expected values, using an interval comprised of the minimum and maximum temperature provides more information and interpretability. Additionally, it reduces the size of the temperature data to a single interval-valued observation for each day, rather than a collection of individual temperatures associated with each day.

Over the past few decades, interval-valued data analysis has garnered increasing interest among researchers. Various models have been built upon set arithmetic [4–10], as well as within the framework of Symbolic Data Analysis (SDA) [11–14]. In the former domain, interval-valued data is typically regarded as realizations of one-dimensional random sets, and statistical inferences are made based on underlying probability theory. SDA [15], on

the other hand, focuses on extending classical statistical methods to handle complex forms of data, such as histograms, hypercubes, and intervals.

Models are being developed in more modern branches of statistics, including spatial statistics [16], Bayesian statistics [17], and statistical learning [18]. From an applied perspective, interval-valued data analysis has been extensively used in various disciplines, including biology, socio-demographic surveys, and forecasting [19].

Over time, the Classification and Regression Tree (CART) [20] and Random Forests (RF) [21] models have become quintessential tools in the machine learning toolbox as nonparametric approaches. Their ability to interpret regression and classification tasks through ensembles of decision trees makes them popular models among data scientists and statisticians. These black box supervised learning algorithms enable users to obtain uncorrelated outcomes that often outperform comparable parametric regression and classification models.

The widespread use of RF and the increasing interest in SDA offer an opportunity for an extension that bridges these two areas. This extension aims to provide a nonparametric approach that can compete with previously proposed regression models [13, 14, 22–25]. Additionally, it takes into consideration the correlation between the center and range of the intervals, which is advantageous as previous methods suggest modeling the centers and ranges individually [25].

The proposed method in this paper synchronizes the RF framework for regression with interval-valued data by introducing a new impurity measure that is based on the Bertoluzza metric d_W [26] for the centers (mids) and range (spread, radius, half-range). The advantages of the proposed method are empirically demonstrated using synthetic and real datasets.

The rest of the paper proceeds as follows: The next chapter introduces the preliminaries of random sets and provides a recap of the frameworks for the various distance measures. Chapter 3 reviews classical tree-based regression and the literature on interval-valued regression, respectively. Our proposed interval-valued RF model is described in Chapter 4. Chapters 5 and 6 present the results from the simulation and real-world data studies, respectively, and Chapter 7 concludes with final remarks.

CHAPTER 2

PRELIMINARIES

2.1 Random Sets

Denote by $\mathcal{K}\left(\mathbb{R}^d\right)$ or $\mathcal{K}$ the collection of all nonempty compact subsets of $\mathbb{R}^d$. The Hausdorff metric

$$\rho_H(A,B) = \max\Big(\sup_{a\in A}\rho\left(a,B\right), \sup_{b\in B}\rho\left(b,A\right)\Big), \forall A,B\in\mathcal{K}, \tag{2.1}$$

where $\mathcal{K}$ is the collection of nonempty closed and bounded (compact) subsets, defines a metric in $\mathcal{K}\left(\mathbb{R}^d\right)$. In $\mathbb{R}^d$, ρ denotes the Euclidean metric. Therefore $\mathcal{K}(\mathbb{R}^d)$ is defined as a metric space, that is complete and separable [27], such that the space $\mathcal{K}$, can be defined as a linear structure by Minkowski addition and scalar multiplication

$$A+B=\{a+b : a\in A, b\in B\} \qquad \text{and} \qquad \lambda A=\{\lambda a : a\in A\}.$$

$\forall A,B\in\mathcal{K}(\mathbb{R})$ and $\lambda\in\mathbb{R}$. Note that $\mathcal{K}(\mathbb{R})$ cannot be considered to be a vector space, as no inverse element of addition exists [28]. Generally, $A+B+(-1)B\neq A$. For example, letting $A=[2,7]$ and $B=[3,5]$, we have

$$A+B+(-1)B=[2,7]+[3,5]+[-5,-3]=[0,9]\neq A.$$

The collection of nonempty compact convex subsets of $\mathbb{R}^d$ is denoted by $\mathcal{K}_{\mathcal{C}}(\mathbb{R})$. When $d=1$, an interval can be represented as belonging to the class $\mathcal{K}_{\mathcal{C}}(\mathbb{R})=[a,b] : a,b\in\mathbb{R}, a\leq b$. All compact intervals, $A\in\mathcal{K}_{\mathcal{C}}(\mathbb{R})$ can be represented by their bounds or their infimum and supremum, i.e., $A=[\underline{a},\overline{a}]=[a^L,a^R]$, where $a^L\leq a^R$. An alternate expression is using the midpoint and the range (spread, radius, half-range), $[a^C\pm a^R]$, where $a^C=(a^R+a^L)/2$

and $a^R = (a^R - a^L)/2$.

Let $(\Omega, \mathcal{L}, P)$ be a probability space. Since $\mathcal{K}$ is equipped with the Borel σ-algebra induced by the Hausdorff Metric, a random compact set is a Borel measurable function $A : \Omega \to \mathcal{K}$. If $A(\omega)$ is convex almost surely, then A is known as a random compact convex set [29]. The expectation of A is defined by the Aumann integral of set-valued functions [30]

$$\mathbb{E}A = \mathbb{E}\xi : \xi \in A \text{ almost surely.} \tag{2.2}$$

For $d = 1$, a measurable function $[X], \Omega \to \mathcal{K}_{\mathcal{C}}(\mathbb{R})$ is called a random interval. The Aumann expectation of $[X]$ is given by

$$\mathbb{E}[X] = [\mathbb{E}X^L, \mathbb{E}X^U]. \tag{2.3}$$

2.2 Distance Measures

For the analysis of set- and interval-valued data, the measure of distance is a critical issue. Several distances were considered and studied in the early stages of SDA. Among those, for instance, the Hausdorff metric ρ_H is a natural choice but was determined to be less preferred for statistical inferences for several reasons. One important reason is that $E_{\rho_H^2}(X, h(X))$ is not minimized at $h(X) = E(X)$, which contradicts the fundamental principle of classical statistics.

For intervals (i.e., one-dimensional sets) in particular, Masuo Hukuhara [31] introduced the Hukuhara difference. Given intervals $A, B \in \mathcal{K}(\mathbb{R})$, where $A - HA = 0$ and $(A + B) - HB = A$, there exists some interval $C \in \mathcal{K}(\mathbb{R})$ [32] such that

$$A -_H B = C \Longleftrightarrow A = B + C, \tag{2.4}$$

$\forall A, B \in \mathbb{R}^d$, and $b^R \leq a^R$. This can be shown by letting $A = [2, 7]$ and $B = [3, 5]$, we then have:

$$A -_H B = [2, 7] -_H [3, 5] = [-1, 2] = C.$$

In general, $A - B \neq A -_H B$.

2.2.1 L_2-distance

According to the embedding theorems [33] and [34], $\mathcal{KC}(\mathbb{R}^d)$ can be isometrically embedded into the Banach space $\mathcal{C}(S)$ of continuous functions on the d-dimensional unit sphere $\mathcal{S}^{d-1}$, which are realized by the support function of $X \in \mathcal{KC}$. As a result, a compact convex random set $X \in \mathcal{KC}(\mathbb{R}^d)$ can be represented by its support function sX with an imposed normalized Lebesgue measure μ.

For the special case $d = 1$, a closed bounded interval $X \in \mathcal{K_C}(\mathbb{R})$ has its support function defined as:

$$S_X(u) = \sup\langle u, x\rangle,\ x \in X,\ u \in \mathcal{S}^0 = -1, 1.$$

Thus, it can be easily seen that S_X can be expressed as

$$S_X(u) = \begin{cases} x^U & \text{if } \mu = 1, \\ -x^L & \text{if } \mu = -1. \end{cases} \tag{2.5}$$

The L_p metric for X can be expressed by the L_p-norm of the support function

$$\|S_X\|_p = \left(\int_{\mathcal{S}^{d-1}} d|S_X(\mu)|^p \mu(du) \right)^{\frac{1}{p}}. \tag{2.6}$$

To obtain the L_2-distance (δ_2) in $\mathbb{R}$ between two intervals $A = [A^L, A^U]$ and $B = [B^L, B^U]$, a similar approach is taken as in (2.6), using the measures from (2.5)

$$\begin{aligned}\delta_2(S_A, S_B)^2 &= \int_{\mathcal{S}^0} (S_A(\mu) - S_B(\mu))^2 \mu(du) \\ &= \frac{1}{2}[(a^U + b^U)^2 - (b^L - a^L)^2],\end{aligned} \tag{2.7}$$

or equivalently, in terms of center and range

$$\delta_2(S_A, S_B)^2 = (a^C - b^C)^2 + (a^R - b^R)^2. \tag{2.8}$$

2.2.2 Weighted L_2-distance

The weighted L_2 distance, referred to as the W-distance, was proposed by [35] based on the Bertoluzza metric [26]. It is considered to be more general than the aforementioned standard L_2 distance because it not only involves the use of the distances between the extreme points (infima and suprema), but also the distances between the inner points in the intervals. Being an L_2-type metric, the W-metric has similar properties that are useful in relation with the least squares method applied in statistical problems.

The Bertoluzza metric [36] d_W on $\mathcal{K}_{\mathcal{C}}(\mathbb{R})$ is defined for every pair of intervals $A = [a^L, a^U]$, $B = [b^L, b^U]$ as the average distance between a point in A and the point in B with the same relative position. Precisely,

$$d_W^2(A, B) = \int_{[0,1]} (f_A(t) - f_B(t))^2 dW(t), \tag{2.9}$$

where W is any nondegenerate symmetric measure on $[0, 1]$ and $f_A(t) = t(a^L) + (1 - t)a^U$, $t \in [0, 1]$, represents the point in the interval A with relative position t. It is easily seen that d_W^2 is computed as

$$d_W^2(A, B) = \int_{[0,1]} [t(a^L - b^L) + (1 - t)(a^U - b^U)]^2 dW(t). \tag{2.10}$$

Considering the relationships

$$\begin{aligned} a^L &= a^C - a^R & b^L &= b^C - b^R \\ a^U &= a^C + a^R & b^U &= b^C + b^R, \end{aligned} \tag{2.11}$$

and letting

$$\begin{aligned} Z_1 &= \int_{[0,1]} t^2 dW(t) & Z_2 &= \int_{[0,1]} (1-t)^2 dW(t), \\ Z_3 &= \int_{[0,1]} t(1-t) dW(t). \end{aligned}$$

We obtain the following

$$\begin{aligned} d_W^2(A,B) &= (a^C - b^C)^2(Z_1 + Z_2 - 2Z_3) + (a^R - b^R)^2(Z_1 + Z_2 + 2Z_3) \\ &\quad + (a^C - b^C)(a^R - b^R)(-2Z_1 + 2Z_2), \end{aligned} \tag{2.12}$$

where

$$\begin{aligned} Z_1 + Z_2 - 2Z_3 &= 1 & Z_1 + Z_2 + 2Z_3 &= \int_{[0,1]} (2t-1)^2 dW(t) \\ -2Z_1 + 2Z_2 &= \int_{[0,1]} (2t-1) dW(t). \end{aligned}$$

Requiring $d_W(-A,-B) = d_W(A,B)\ \forall A, B \in \mathcal{K}_{\mathcal{C}}(\mathbb{R})$, we derive

$$4(a^C - b^C)(a^R - b^R)\left(\int_{[0,1]} (2t-1) dW(t)\right) = 0,$$

which implies

$$\int_{[0,1]} (2t-1) dW(t) = 0,$$

or equivalently

$$\mathbb{E}_W[t] = \int_{[0,1]} t dW(t) = \frac{1}{2}.$$

The alternative center-range formula is obtained for the d_W distance as

$$d_W^2(A,B) = (a^C - b^C)^2 + (a^R - b^R)^2 \left(\int_{[0,1]} (2t-1)^2 dW(t) \right). \tag{2.13}$$

Specifically, $\int_{[0,1]}(2t-1)dW(t) \in [0,1]$ is some constant determined by W, and as such, when calculating the L_2-distance, the W-distance can also be interpreted as a weight for $(a^R - b^R)^2$. Therefore (2.9) and (2.13), present the advantage of the flexibility of the W-distance in assigning weights to the points in the interval.

2.2.3 D_K-distance

In a separate context, Körner and Näther [37] proposed another L_2 metric, which when restricted to $\mathcal{K}_{\mathcal{C}}(\mathbb{R})$ is

$$D_K^2(A,B) = \sum_{(u,v)\in S^0 \times S^0} (s_A(u) - s_B(u))\,(s_A(v) - s_B(v))\,K(u,v), \tag{2.14}$$

where K is a symmetric positive definite kernel and $(u,v) \in \mathcal{S}^0 \times \mathcal{S}^0$. Recall that $S^0 = \{-1,1\}$ and

$$S_A(u) = \begin{cases} a^U, & u = 1 \\ -a^L, & u = -1, \end{cases} \qquad S_B(u) = \begin{cases} b^U, & u = 1 \\ -b^L, & u = -1. \end{cases}$$

It follows that

$$D_K^2(A,B) = K(1,1)(a^U - b^U)^2 + K(-1,-1)(b^L - a^L)^2 + 2K(1,-1)(a^U - b^U)(b^L - a^L). \tag{2.15}$$

Considering the relationships from (2.11), and the expressions of $D_K^2(A,B)$, each expression can be obtained separately as

$$\begin{aligned}
K(1,1)(a^U - b^U)^2 &= K(1,1)((a^C + a^R) - (b^C + b^R))^2 \\
K(-1,-1)(b^L - a^L)^2 &= K(-1,-1)((b^C - b^R) - (a^C - a^R))^2 \\
2K(1,-1)(a^U - b^U)(b^L - a^L) &= 2K(1,-1)((a^C + a^R) - (b^C + b^R))((b^C - b^R) - (a^C - a^R)).
\end{aligned}$$

After expanding and combining like terms, we form the following linear combinations of the kernel, K

$$\begin{aligned}
A_{11} &= K(1,1) + K(-1,-1) - [K(1,-1) + K(-1,1)] \\
A_{22} &= K(1,1) + K(-1,-1) + [K(1,-1) + K(-1,1)] \\
A_{12} &= A_{21} = K(1,1) - K(-1,-1).
\end{aligned}$$

Thus, the D_K distance can be defined alternatively in the center-range form as

$$D_K^2(A,B) = A_{11}(a^C - b^C)^2 + A_{22}(a^R - b^R)^2 + 2A_{12}(a^C - b^C)(a^R - b^R). \tag{2.16}$$

It is seen that, when K is a symmetric positive definite, so is A. Therefore, we see that D_K is a more generalized L_2 metric than d_W that takes into account the interaction between the center and the range.

These distance measures play a fundamental role in defining an appropriate splitting criterion (eq. 4.9) for interval-valued data when using a tree-based regression model. This development extends the regression framework for interval-valued data in a nonparametric manner. Linear regression for intervals typically treats an interval as a bivariate vector and fits separate point-valued models to the center and range (or lower and upper bounds) of the interval. However, our proposed model considers the relationship between the center and range, which previous linear regression models fail to capture.

CHAPTER 3
REGRESSION REVIEW

3.1 Review of Interval-Valued Regression Models

Linear regression for interval-valued data has been extensively studied over the past decades. Existing models have been developed mainly in the domains of random sets and SDA. In the framework of random sets, an interval is viewed as a single entity, and the linear relationships between intervals are modeled using set arithmetic. Models developed in this framework are generally restrictive to achieve certain mathematical properties and will not be further discussed in this thesis.

On the other hand, the aim of SDA is to extend classical data analysis techniques to nontraditional data formats, such as lists, intervals, histograms, and distributions. As a result, SDA models usually offer improved flexibility and are preferred in many practical situations. In the following, we will review the major SDA models for interval-valued regression.

Consider predictor intervals $[X_i], \ldots, [X_p]$ where $[\mathbf{X}]$ is a $n \times p$ matrix of intervals. Each row is a vector $x_i = (x_{ij}, \ldots, x_{ip})$, with $x_{ij} = [x_{ij}^L, x_{ij}^U]$. Let the response interval to be predicted $[\mathbf{Y}]$ and $y_i = [y_i^L, y_i^U]$.

3.1.1 Center Method (CM)

The CM [22] uses the the center as the parameter estimation for the β coefficients. It was the initial approach to fitting a linear regression model to interval-valued data.

Let $x_{ij}^C = (x_{ij}^L + x_{ij}^U)/2$ and $y_{ij}^C = (y_i^L + y_i^U)/2$, and the linear regression relationship is as follows

$$\mathbf{y}^C = \mathbf{X}^C \boldsymbol{\beta} + \varepsilon^C.$$

Where, $\mathbf{y}^C = (y_1^C, \ldots, y_i^C)$, $\mathbf{X}^C = ((x_1^c)^T, \ldots, (x_n^c)^T)$, $(x_1^c)^T = (1, x_{i1}^c, ..., x_{1p}^c)(i = 1, \ldots, n)$, $\beta = (\beta_0, \ldots, \beta_p)$, $\varepsilon_i^C = (\varepsilon_1^C, \ldots, \varepsilon_n^C)^T$, The $\boldsymbol{\beta}$ coefficients can be estimated if $\mathbf{X}^C$ has a full rank $p+1 \leq n$, and so the least squares estimates for β are given as

$$\hat{\boldsymbol{\beta}} = ((\mathbf{X}^C)^T \mathbf{X}^C)^{-1} (\mathbf{X}^C)^T \mathbf{y}^C.$$

Therefore the estimates for predictions of y are

$$\hat{\mathbf{y}}^L = (\mathbf{x}^L)^T \hat{\boldsymbol{\beta}} \qquad \hat{\mathbf{y}}^U = (\mathbf{x}^U)^T \hat{\boldsymbol{\beta}}. \tag{3.1}$$

Notice that the CM uses only the center point of the interval in estimating the $\boldsymbol{\beta}$ parameters, while it may be more suitable to consider both the centers and ranges of an interval for parameter estimation and improvement of model prediction performance.

3.1.2 Center and Range Method (CRM)

The CRM was introduced by [13], as a new linear regression method using the midpoints and range of the interval-valued data. Consider the formalities from the CM and let $x_{ij}^R = (x_{ij}^U - x_{ij}^L)/2$ and $y_{ij}^R = (y_{ij}^U - y_{ij}^L)/2$, the linear regression relationship is as follows

$$\mathbf{y}^C = \mathbf{X}^C \boldsymbol{\beta}^C + \varepsilon^C$$
$$\mathbf{y}^R = \mathbf{X}^R \boldsymbol{\beta}^R + \varepsilon^R.$$

Where the conditions for the center and range are

$$\begin{aligned}
\mathbf{y}^C &= (y_1^C, \ldots, y_i^C)^T & \mathbf{y}^R &= (y_1^R, \ldots, y_i^R) \\
\mathbf{X}^C &= ((x_1^C)^T \ldots, (x_n^C)^T) & \mathbf{X}^R &= ((x_1^R)^T, \ldots, (x_n^R)^T) \\
\boldsymbol{\beta}^C &= (\beta_0^C, ..., \beta_p^C)^T & \boldsymbol{\beta}^R &= (\beta_0^R, \ldots, \beta_p^R)^T \\
\varepsilon^C &= (\varepsilon_1^C, \ldots, \varepsilon_n^C)^T & \varepsilon^R &= (\varepsilon_1^R, \ldots, \varepsilon_n^R)^T \\
(x_i^C)^T &= (1, x_{i1}^C, \ldots, x_{1p}^C)(i = 1, \ldots, n) & (x_i^R)^T &= (1, x_{i1}^R, \ldots, x_{1p}^R)(i = 1, \ldots, n).
\end{aligned} \tag{3.2}$$

The least squares estimates for $\boldsymbol{\beta}^C$ and $\boldsymbol{\beta}^R$ are given, if both $\mathbf{X}^C$ and $\mathbf{X}^R$ are full rank $p+1 \leq n$, as

$$\hat{\boldsymbol{\beta}}^C = ((\mathbf{X}^C)^T\mathbf{X}^C)^{-1}(\mathbf{X}^C)^T\mathbf{y}^C$$
$$\hat{\boldsymbol{\beta}}^R = ((\mathbf{X}^R)^T\mathbf{X}^R)^{-1}(\mathbf{X}^R)^T\mathbf{y}^R.$$

Therefore the estimates of predictions $[\hat{\mathbf{y}}]$ are

$$\hat{\mathbf{y}}^C = (\mathbf{x}^C)^T\hat{\boldsymbol{\beta}}^C \qquad\qquad \hat{\mathbf{y}}^R = (\mathbf{x}^R)^T\hat{\boldsymbol{\beta}}^R. \tag{3.3}$$

Seeing as the CRM builds upon the CM by considering the range of an interval in estimating the coefficients. A concern is that the $\hat{\boldsymbol{\beta}}^R$ must be greater than or equal to zero. Which is not guaranteed unless one considers an inequality constraint over $\hat{\boldsymbol{\beta}}^R$.

3.1.3 Constrained Center and Range Method (CCRM)

The CCRM [14] considers one important feature that is not accounted for in the CRM. The CCRM mathematically ensures the inequality of $y_i^L \leq y_i^U$ is true by constraining $\beta_j^R \geq 0$. The linear regression relationship is as follows

$$\mathbf{y}^C = \mathbf{X}^C\boldsymbol{\beta}^C + \varepsilon^C$$
$$\mathbf{y}^R = \mathbf{X}^R\boldsymbol{\beta}^R + \varepsilon^R.$$

While $\beta_j^R \geq 0, j = 0, \ldots, p$ is the constraint required for the CCRM. One important item to note is the parameters: $\beta_j^C, (j = 0, \ldots, p)$ do not have any restrictions. Additionally,

$$\begin{aligned}
\mathbf{y}^C &= (y_1^C, \ldots, y_i^C)^T, & \mathbf{y}^R &= (y_1^R, \ldots, y_i^R), \\
\mathbf{X}^C &= ((x_1^C)^T \ldots, (x_n^C)^T), & \mathbf{X}^R &= ((x_1^R)^T, \ldots, (x_n^R)^T), \\
\boldsymbol{\beta}^C &= (\beta_0^C, ..., \beta_p^C)^T, & \boldsymbol{\beta}^R &= (\beta_0^R, \ldots, \beta_p^R)^T, \\
\varepsilon^C &= (\varepsilon_1^C, \ldots, \varepsilon_n^C)^T, & \varepsilon^R &= (\varepsilon_1^R, \ldots, \varepsilon_n^R)^T, \\
(x_i^C)^T &= (1, x_{i1}^C, \ldots, x_{1p}^C)(i = 1, \ldots, n), & (x_i^R)^T &= (1, x_{i1}^R, \ldots, x_{1p}^R)(i = 1, \ldots, n).
\end{aligned} \tag{3.4}$$

The least squares estimates for $\boldsymbol{\beta}^C$ and $\boldsymbol{\beta}^R$ are given as

$$\hat{\boldsymbol{\beta}}^C = ((\mathbf{X}^C)^T \mathbf{X}^C)^{-1} (\mathbf{X}^C)^T \mathbf{y}^C$$
$$\hat{\boldsymbol{\beta}}^R = ((\mathbf{X}^R)^T \mathbf{X}^R)^{-1} (\mathbf{X}^R)^T \mathbf{y}^R.$$

The predictions for $[\hat{\mathbf{y}}]$ are

$$\hat{\mathbf{y}}^C = (\mathbf{x}^C)^T \hat{\boldsymbol{\beta}}^C \qquad \hat{\mathbf{y}}^R = (\mathbf{x}^R)^T \hat{\boldsymbol{\beta}}^R. \tag{3.5}$$

Constraining the $\boldsymbol{\beta}^R$ coefficients mathematically ensures the nonnegativity required for the CCRM.

3.1.4 Limitation of Existing Models

There is an intrinsic difficulty in performing linear regression with interval-valued data because $\mathcal{K}_{\mathcal{C}}(\mathbb{R})$ does not have an inverse operation of addition, and therefore, it is not a linear space (see [38] for a detailed discussion). This fundamental issue necessitates the imposition of nonnegative constraints in most models. However, these constraints introduce biased estimators as they typically penalize underestimation more heavily than overestimation. Additionally, they significantly complicate the computational aspects, making it challenging to draw inferences.

Hence, there is a need to extend the analysis beyond these linear methods and introduce tree-based methods as an appropriate alternative.

3.2 Regression Trees

Regression trees are a powerful and popular tool for analyzing data in machine learning through a nonparametric framework. They first originated in the 1960s by Morgan and Sonquist [39] and were developed in the 1980s as CART by statisticians Leo Breiman and colleagues [20]. Regression trees are a type of decision tree that is grown when the data has a nominal response variable y, by dividing the predictor space $X = X_1, X_2, ..., X_p$ in $\mathcal{R}^n$ into J distinct nonoverlapping regions $R_1, ..., R_J$ referred to as nodes.

The formulation of the nodes can be framed as an optimization problem with the task of partitioning X into R_j boxes that minimize the RSS for each R_j.

$$RSS_{R_j} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{Rj})^2. \tag{3.6}$$

Here, $\hat{y}_{Rj}$ denotes the mean response for the training observations within the j-th region of X.

Unfortunately, considering every possible partition of X is computationally infeasible. Therefore, regression trees adopt a top-down greedy approach known as recursive binary partitioning.

The algorithm for growing a regression tree is as follows :

1. Begin with a single node containing X.
2. Stop if the stopping criterion is met. Otherwise, search for the greatest reduction in RSS_{R_j} by considering all binary splits s of X_j that partition X into binary child nodes $X_j < s$ and $X_j \geq s$.
3. Partition the data into two new child nodes based on the reduction in RSS_{R_j} found in step 2.
4. Repeat step 2 for each new node.

Once the algorithm concludes, each terminal (leaf) node c will have a sample mean computed from the response values of the data, representing the predicted value for that leaf node, denoted as $\hat{y}_c$. Resulting in the formulation of a regression tree, T.

Regression trees are particularly useful in scenarios where the underlying data is complex or nonlinear, or when the dataset contains numerous variables. They can also help identify interactions between predictor variables, facilitating the identification of the most important predictors in a dataset.

3.3 Random Forests (RF)

The RF algorithm is an ensemble method developed by Leo Breiman [21]. RF consist of a collection of trees. Prior to the introduction of RF, ensembles had already attracted significant attention due to their ability to achieve more accurate predictions than individual trees. Various techniques for constructing ensembles were proposed by different authors, including popular examples such as Bagging [40] and Adaboost [41]. In Bagging, each tree is constructed from a bootstrap sample drawn with replacement from the training data. The original version of Adaboost resamples observations with weights that are successively adjusted to give higher weight to "difficult" observations. Randomization is used to create an ensemble by randomizing the interval decisions made by the base algorithm.

After constructing the ensembles, classification is performed by taking the majority vote of the individual classifiers, while regression involves taking the average. It is well understood that averaging results leads to variance reduction and reduces the correlation between individual classifiers, thereby further enhancing the variance gains. Motivated by this principle, RF introduce another layer of randomness by changing the structure of each tree. Instead of optimizing the response by evaluating all the predictors, as is done with single-tree methods or Bagging, RF employ a subset of predictors that is randomly drawn independently for each node in each tree. This strategy has been shown to perform exceptionally well and exhibits robustness against overfitting.

One of the main advantages of RF over other estimation methods is their full non-parametric nature, which includes the effects of predictors and response variables. This

allows RF to effectively handle issues such as nonlinearity and mathematical coherence. Specifically, the prediction of the range is based on an average of the terminal nodes, which contain all positive elements (i.e., range of the observed intervals), resulting in automatic positive predictions without any constraints. Traditional methods may struggle with situations where the number of predictor variables is equal to or greater than the number of observations, but RF automatically handle the issue of dimensionality, largely due to the use of decision trees as base learners in the ensemble process. Like all tree-based methods, RF naturally capture interactions between predictors without the need for specifying them in advance.

Importantly, the RF algorithm has only three tuning parameters: the size of the subset at each node, the number of trees in the forest, and the depth of the trees. This makes it convenient and practical to apply. In fact, the number of trees can be chosen arbitrarily large without risking loss of accuracy, and for classification tasks, the trees can be grown to their maximum depth. The results are not particularly sensitive even to the size of the subset at each node, making tuning relatively straightforward.

The algorithm for growing a RF for regression is as follows:

1. For each b from 1 to B:
   (a) Take the training data and draw a bootstrap sample Z of size N.
   (b) For an individual regression tree T_b grown from Z, recursively repeat the following steps for each new node until a stopping criterion is met:
      i. Randomly sample p variables from the P variables of the data.
      ii. Follow the procedure for growing a regression tree as described in Section 3.2.
2. Output the ensemble of trees $T_{1:B}$.

After growing the forest, predictions can be made for a new point x by averaging the results from all the trees.

CHAPTER 4

INTERVAL-VALUED REGRESSION TREE (IRT) AND RANDOM FORESTS (IRF)

While linear regression is often preferred for its simplicity, there are many practical situations where nonlinearity exists and linear methods alone are insufficient to address those problems. Therefore, it is important to develop nonlinear regression methods for interval-valued data. However, compared to linear regression, nonlinear regression for interval-valued data has received much less attention in research.

Among these, we are particularly interested in the RF regression proposed by [25]. This model separates the centers and ranges of the intervals, formulating the regression task by fitting separate point-valued models for the center and range.

One concern is that fitting separate models does not consider the correlation between the centers and ranges. In the multivariate setting of RF, ignoring the correlation among response variables (treating each response variable separately) can potentially have a substantial impact on performance when the response variables are highly correlated [42] and [43]. Therefore, for interval-valued data, we will focus on proposing a new method that considers the potential impact of the correlation between the centers and ranges of the data.

The extension of tree-based regression methods to interval-valued data from univariate tree-based regression methods is achieved by replacing the univariate variables X and Y with an interval response variable $[X]$ and $[Y]$. In order to accommodate interval-valued data, it becomes necessary to redefine the splitting criterion from the univariate sum of squares to an interval-valued version, which involves computing the sum of squares of the interval-valued means. This is done by calculating the sum of squared differences (SSD) between the centers and ranges of the intervals. The objective at each node is to minimize the sum of squared distances of the intervals, thereby reducing the variance within each interval for the individual leaf nodes.

Consider $[X]$ and $[Y]$ to be interval-valued variables, where

$$[\mathbf{x_{ij}}] = [x^L ij, x^U ij] \in \mathcal{K_C}(\mathbb{R}) \tag{4.1}$$

$$= [x^L, x^U] : x^L, x^U \in \mathbb{R}, x^L \leq x^U \tag{4.2}$$

$$= [x^C \pm x^R] \tag{4.3}$$

and

$$[\mathbf{y_i}] = [y_i^L, y_i^U] \in \mathcal{K_C}(\mathbb{R}) = \qquad [y_i^C \pm y_i^R] \tag{4.4}$$

are the observed values of $[X]$ and $[Y]$.

Therefore interval-valued regression trees, $[T]$, can be grown by dividing the interval-valued predictor space $[X] = [X_1], [X_2], \ldots, [X_n]$ into K distinct nodes $R_1, \ldots, R_K$. Recall the Aumann Expectation (2.3) of $[X]$

$$\mathbb{E}[X] = [\mathbb{E}X^L, \mathbb{E}X^U],$$

the δ^2 of two intervals from Section 2.2.1

$$\delta^2([a], [b]) = (a^C - b^C)^2 + (a^R - b^R)^2,$$

and the δ_ω^2 of two intervals from Section 2.2.2

$$\delta_\omega^2([a], [b]) = (a^C - b^C)^2 + \omega(a^R - b^R)^2.$$

Therefore the variance of $[X]$ is

$$Var([X]) = \mathbb{E}(\delta^2([X], \mathbb{E}[X]))$$

$$= Var(X^C) + Var(X^R) \tag{4.5}$$

or

$$Var_{\omega}([X]) = Var(X^C) + \omega Var(X^R) \tag{4.6}$$

Note that these variance equations are when X^C and X^R are independent of each other. If we wanted to consider a more generalized version than we can use Sections 2.2.2 and 2.2.3, such that

$$Var([X]) = Var(X^C) + Var(X^R) + 2Cov(X^C, X^R) \tag{4.7}$$

and

$$Var_{\omega}([X]) = Var(X^C) + \omega Var(X^R) + 2\omega Cov(X^C, X^R). \tag{4.8}$$

Minimizing the SSD of the centers and range of $[y_i]$ and $[\hat{y}_i]$ requires the partitioning of $[X]$ into R_K boxes that minimize the the SSD of R_K.

$$SSD_{R_K} = \sum_{k=1}^{K} \sum_{i \in R_k} \delta^2([y_i], [\hat{y}_{R_k}]) \tag{4.9}$$

For the RF, by building B regression trees on bootstrap sets of the center and range data, denoted as $[T] \in \{T_j^C$ and $T_j^R\}$, we obtain the following predictions for the centers and ranges:

$$\hat{y}^C = \frac{1}{B} \sum_{j=1}^{B} T_j^C(x^C) \qquad \hat{y}^R = \frac{1}{B} \sum_{j=1}^{B} T_j^R(x^R), \tag{4.10}$$

The formulation of the proposed method of an Interval-valued regression tree follows the traditional regression tree algorithm 1, and the approach from the traditional RF algorithm 2 will be used for the proposed method of IRF. Both require a new stopping criterion derived from Equation 4.9. This new method will be tested in Chapter 5 and Chapter 6, with simulated and real data.

CHAPTER 5

SIMULATION STUDY

This section demonstrates the computational feasibility of the IRF model in comparison to CCRM, traditional RF, and IRT models. Monte Carlo experiments were conducted using various settings, considering different characteristics of the centers and ranges of the dependent and independent interval variables, as well as their respective error terms.

Each setting independently considered three sample sizes: $n = 100, 250, 500$. For each simulated dataset, 80% of the data was used as the training set, and the remaining 20% as the testing set. The analysis was performed using the datasets in R, with CCRM and RF models formulated using the `iRegression` [44] and `randomForest` [45] packages in R [46]. The models for IRT were generated using the `mvpart` [47] package.

One of our major contributions in this research is the manual programming necessary to implement IRF. Unfortunately, `mvpart` has been removed from CRAN and is no longer actively maintained to align with the current updates of R (4.2.3).

As a workaround, we wrapped `mvpart` within a Bagging architecture for simulation. The simulations and real data example exhibit a relatively small number of total predictor intervals, which can potentially minimize selection bias and allow for an exhaustive search of all possible data [40]. Each tree in the RF will be trained using all the available predictor intervals without any randomness introduced during variable selection. Therefore, the random selection of predictor intervals is effectively disabled, and the RF algorithm behaves similarly to Bagging [21]

In future research, we aim to extend the framework by incorporating random subsampling of predictor intervals at the individual node level to more closely resemble the exact structure of a RF. For programming details, please refer to the following source:

https://github.com/PaulGaona/IntRF.

The empirical evaluations and assessment of model performances between the IRF, IRT, RF, and CCRM, using various error metrics, are shown in Section 5.1.

5.1 Accuracy Measures

The assessment of model performance relies on three commonly used error metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{5.1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid, \tag{5.2}$$

and

$$R^2 = 1 - \frac{\text{MSE}}{\sigma_{Train}} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{5.3}$$

where $\mathbf{y} = (y_1, \cdots, y_n)$ and $\hat{\mathbf{y}} = (\hat{y}1, \cdots, \hat{y}n)$ represent the predicted and observed values from the testing dataset $[Y_{\text{Test}}]$, with $i = 1, \cdots, n$. Similarly, $\mathbf{y} = (y_1, \cdots, y_m)$ and $\hat{\mathbf{y}} = (\hat{y}1, \cdots, \hat{y}m)$ denote the observed values from the training dataset $[\mathbf{Y}_{\text{Train}}]$, with $j = 1, \cdots, m$.

These equations will be adapted for interval-valued data by combining errors from both the centers and ranges of the predicted intervals, ensuring consistency with the d_W-distance defined as the splitting criteria. They will be referred to as the Total Standardized Mean Squared Error (TS-MSE), Total Standardized Mean Absolute Error (TS-MAE), and Composite Coefficient of Determination (C-R^2). Their formulas are as follows:

$$TS\text{-}MSE = MSE^C + \omega MSE^R = \frac{1}{n}\sum_{i=1}^{n}(y_i^C - \hat{y}_i^C)^2 + \frac{\omega}{n}\sum_{i=1}^{n}(y_i^R - \hat{y}_i^R)^2, \tag{5.4}$$

$$TS\text{-}MAE = MSE^C + \omega MSE^R = \frac{1}{n}\sum_{i=1}^{n} \mid y_i^C - \hat{y}_i^C \mid + \frac{\omega}{n}\sum_{i=1}^{n} \mid y_i^R - \hat{y}_i^R \mid, \tag{5.5}$$

and

$$C\text{-}R^2 = 1 - \frac{\text{MSE}_{Test}^C + \text{MSE}_{Test}^R}{\sigma_{Train}^C + \sigma_{Train}^R} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i^C - \hat{y}_i^C)^2 + \frac{1}{n}\sum_{i=1}^{n}(y_i^R - \hat{y}_i^R)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i^C - \bar{y}^C)^2 + \frac{1}{n}\sum_{i=1}^{n}(y_i^R - \bar{y}^R)^2}, \tag{5.6}$$

where $\mathbf{y^C} = (y^C 1, \cdots, y^C n)$, $\hat{\mathbf{y}}^\mathbf{C} = (\hat{y}^C 1, \cdots, \hat{y}^C n)$, $\mathbf{y^R} = (y^R 1, \cdots, y^R n)$, $\hat{\mathbf{y}}^\mathbf{R} = (\hat{y}^R 1, \cdots, \hat{y}^R n)$ represent the predicted and observed values of the centers and ranges from the testing dataset [$\mathbf{Y}_{\text{Test}}$], where $i = 1, \cdots, n$. Similarly, $\mathbf{y^C} = (y^C 1, \cdots, y^C m)$ and $\mathbf{y^R} = (y^R 1, \cdots, y^R m)$ represent the observed values of the centers and ranges from the training dataset [$\mathbf{Y}_{\text{Train}}$], where $j = 1, \cdots, m$. σ^C is the standard deviation of $\mathbf{y^C}$, and σ^R is the standard deviation of $\mathbf{y^R}$.

Each setting was carefully selected to highlight the competitiveness of IRF against the other models. Setting 1 and 2 are linear settings. In Setting 1, we anticipate that CCRM will be the most competitive model due to the linear nature of the data and model. Additionally, we expect IRF and RF to perform well and be competitive with CCRM. In Setting 2, we maintain a linear dataset but introduce a negative trend for the centers and range. Here, we expect IRF and RF to outperform CCRM because the constraint of a nonnegative range in CCRM may lead to challenges compared to the tree-based methods.

Setting 3 and Setting 4 are nonlinear settings, but when introduced with errors, they exhibit a close-to-linear appearance. This creates a simulation scenario where CCRM may excel in predicting the data, despite its inherent nonlinearity. Thus, in Setting 4, we anticipate that CCRM will be competitive against IRF. Similar expectations to Setting 2 are applicable to Setting 3, as it also includes a negative trend in the range.

Setting 5 and Setting 6 are nonlinear settings. In Setting 5, both the centers and range follow a trigonometric relationship. However, similar to Settings 3 and 4, the range exhibits a close-to-linear appearance. Consequently, we expect the tree-based methods to outperform CCRM. Setting 6 features a parabolic relationship in the centers and a close-to-linear relationship in the range, while incorporating an interaction between the centers

and range for predicting the range. In this case, we anticipate that IRF will outperform RF and CCRM, taking advantage of the correlation among the outcome variables [42].

Lastly, Setting 7 adopts a multiple interval approach, considering nonlinearities and interactions among the centers and range with five predictor intervals. This enables the tree-based methods to construct splits based on multiple variables. Due to the nonlinear nature and correlations among some variables, we expect IRF to outperform RF and CCRM.

The following distributions for X^C, X^R, ε^C, and ε^R are independently generated according to the following specifications for each setting 1-6:

5.2 Settings

- Setting 1:

$$\begin{array}{ll} X^C \sim \mathcal{N}(12, 3^2) & X^R \sim \mathcal{U}(1, 3) \\ \varepsilon^C \sim \mathcal{N}(0, 0.75^2) & \varepsilon^R \sim \mathcal{N}(0, 0.05^2), \end{array}$$

  and the center and range of the response variable

$$\begin{aligned} Y^C &= 2X^C + 15 + \varepsilon^C \\ Y^R &= .25X^R + \varepsilon^R. \end{aligned} \tag{5.7}$$

- Setting 2:

$$\begin{array}{ll} X^C \sim \mathcal{N}(-5, 10^2) & X^R \sim \mathcal{U}(1, 2) \\ \varepsilon^C \sim \mathcal{N}(0, 2.5^2) & \varepsilon^R \sim \mathcal{N}(0, 0.1^2), \end{array}$$

  and where the center and range of the response variable as

$$\begin{aligned} Y^C &= -X^C + 50 + \varepsilon^C \\ Y^R &= -2X^R + 5 + \varepsilon^R. \end{aligned} \tag{5.8}$$

- Setting 3:

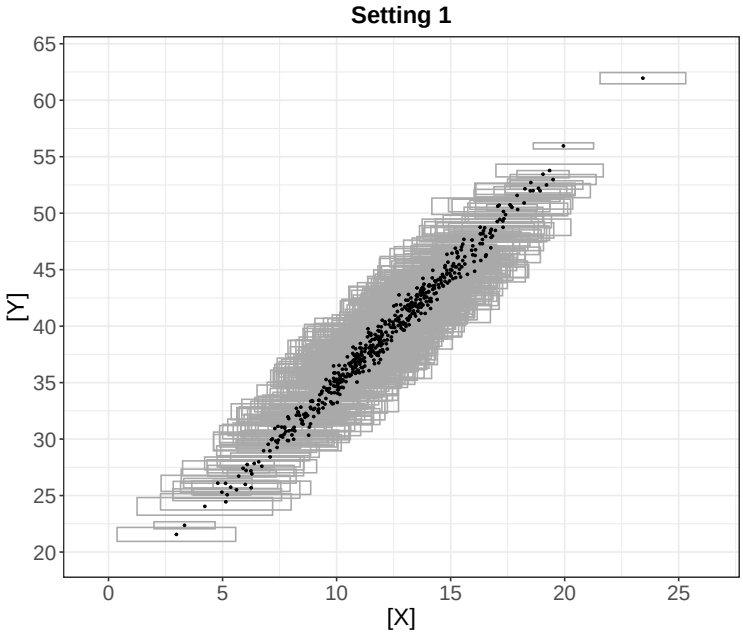

Fig. 5.1: Scatterplot of centers and range for the linear Setting 1 with 100 observations.

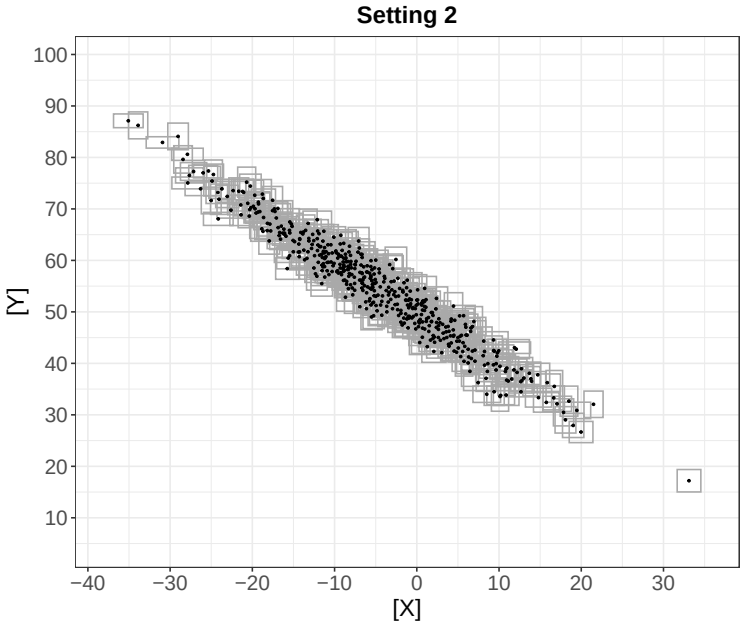

Fig. 5.2: Scatterplot of centers and range for the linear Setting 2 with 250 observations.

$$\begin{array}{ll} X^C \sim \mathcal{N}(5, 1^2) & X^R \sim \mathcal{U}(0.1, 0.25) \\ \varepsilon^C \sim \mathcal{N}(0, 1^2) & \varepsilon^R \sim \mathcal{N}(0, 0.25^2), \end{array}$$

therefore the equations for the center and range of the response variables are

$$\begin{aligned} Y^C &= .5(X^C)^2 + 20 + \varepsilon^C \\ Y^R &= 0.05(X^R)^{-2} + 1 + \varepsilon^R. \end{aligned} \tag{5.9}$$

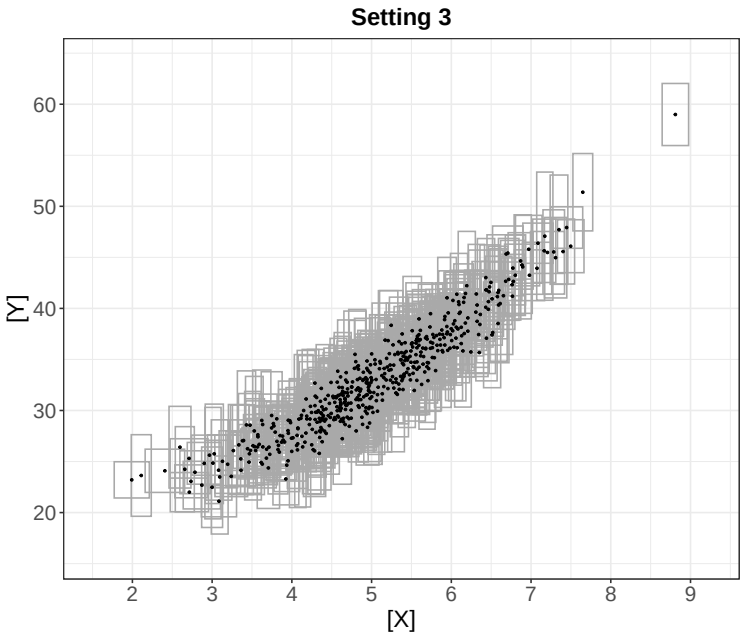

Fig. 5.3: Scatterplot of centers and range for the close-to-linear Setting 3 with 500 observations.

- Setting 4:

$$\begin{array}{ll} X^C \sim \mathcal{N}(12, 4^2) & X^R \sim \mathcal{U}(0.5, 2) \\ \varepsilon^C \sim \mathcal{N}(0, 1^2) & \varepsilon^R \sim \mathcal{N}(0, 0.125^2), \end{array}$$

the equations for the center and range of the response variables follow

$$Y^C = 10\ln(X^C) + 10 + \varepsilon^C$$
$$Y^R = 2.5\sqrt{X^R} + \varepsilon^R. \tag{5.10}$$

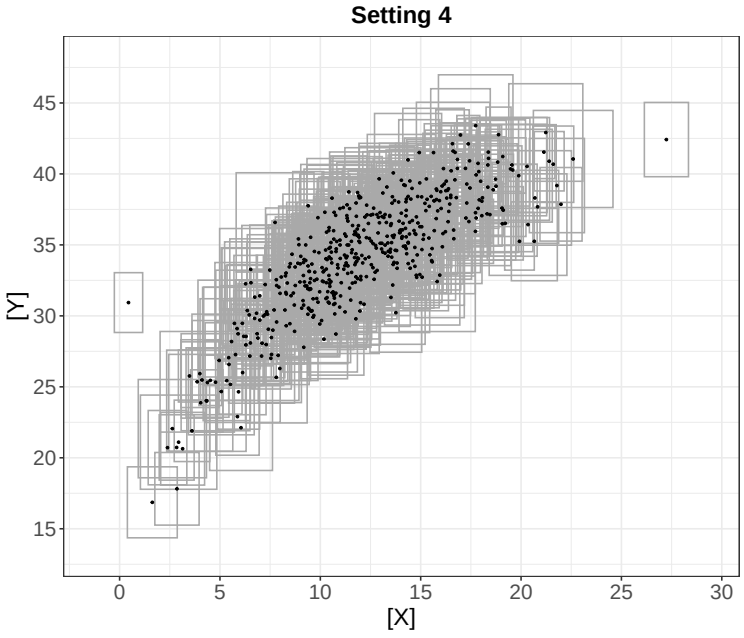

Fig. 5.4: Scatterplot of centers and range for the close-to-linear Setting 4 with 100 observations.

- Setting 5:

$$\begin{array}{ll} X^C \sim \mathcal{N}(3, 2^2) & X^R \sim \mathcal{U}(0.5, 1) \\ \varepsilon^C \sim \mathcal{N}(0, 1.5^2) & \varepsilon^R \sim \mathcal{N}(0, 0.0625^2), \end{array}$$

the equations for the center and range of the response variables follow

$$\begin{aligned} Y^C &= 10\sin(0.15\pi X^C) + 10 + \varepsilon^C \\ Y^R &= X^R + 0.5 + \varepsilon^R. \end{aligned} \tag{5.11}$$

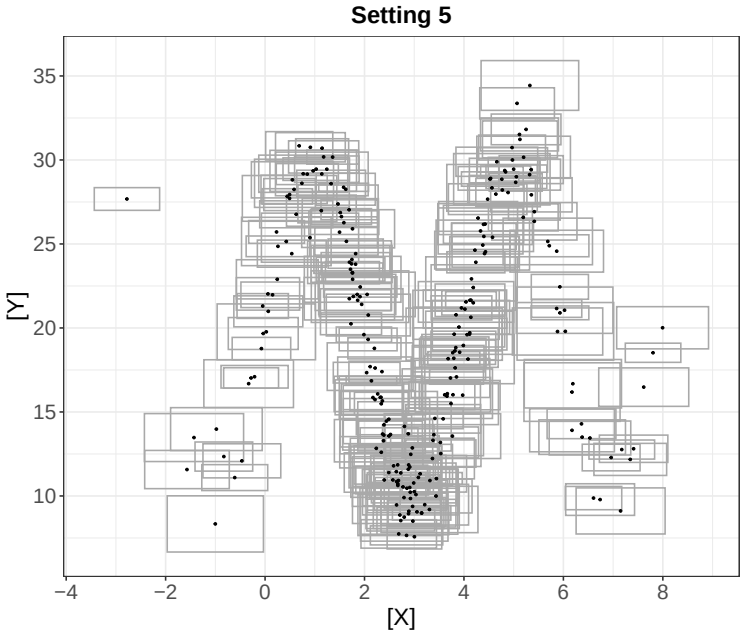

Fig. 5.5: Scatterplot of centers and range for the nonlinear Setting 5 with 250 observations.

- Setting 6:

$$\begin{array}{ll} X^C \sim \mathcal{N}(8, 2^2) & X^R \sim \mathcal{U}(0,1) \\ \varepsilon^C \sim \mathcal{N}(0, 1^2) & \varepsilon^R \sim \mathcal{N}(0, 0.25^2), \end{array}$$

the equations for the center and range of the response variables follow

$$\begin{aligned} Y^C &= -(X^C - 8)^2 + 32 + \varepsilon^C \\ Y^R &= 0.0625 e^{X^R}\sqrt{X^C} + \varepsilon^R. \end{aligned} \tag{5.12}$$

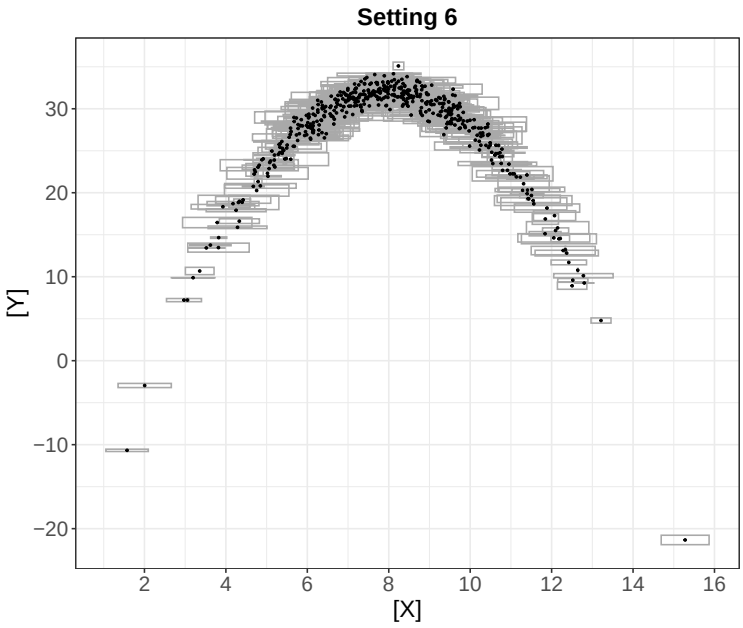

Fig. 5.6: Scatterplot of centers and range for the nonlinear Setting 6 with 500 observations.

- Setting 7: Their respective centers have the following distributions: $X_1^C \sim \mathcal{N}(5, 3^2)$, $X_2^C \sim \beta(0.5, -.5)$, $X_3^C \sim \mathcal{N}(10, 3.5^2)$, $X_4^C \sim \mathcal{U}(0.5, 1.5)$, and $X_5^C \sim \mathcal{N}(8, 3.5^2)$ Now define,

$$V_1 = u_1 e^{-0.5\gamma(3,2)+\tau_1},$$
$$V_2 = u_2 e^{-0.5\beta(1,2)+\tau_2}, \tag{5.13}$$

  where $\tau_1, \tau_2 \sim \mathcal{N}(0, 0.2^2)$ and $u_1, u_2 \sim \mathcal{U}(0, 0.5)$ are generated independently. The distribution for the predictor interval range are range for the predictor intervals are generated by $X_1^R = \dfrac{2V_1}{1+V1}$, $X_2^R = \dfrac{3V_2}{1+V2}$, $X_3^R \sim \mathcal{N}(10, 3^2)$, $X_4^R \sim \mathcal{U}(2.5, 3.5)$, $X_5^R \sim \beta(2, 5)$. The center and range of the response interval are determined by the

following equations

$$
\begin{aligned}
Y_i^C &= (X_{1i}^C + (X_{1i}^C)^2)(X_{2i}^C + (X_{2i}^C)^2) - (X_{3i}^C + (X_{3i}^C)^2)(X_{4i}^C + (X_{4i}^C)^2) - X_{5i}^C + \varepsilon_i^C \\
Y_i^R &= \frac{(X_{2i}^R)^2}{5} + 0.1X_{3i}^R - 5(X_{1i}^R X_{4i}^R + X_{5i}^R) + 4 + \varepsilon_i^R,
\end{aligned} \tag{5.14}
$$

where $\varepsilon_i^C \sim \mathcal{N}(0, 1^2)$ and $\varepsilon_i^R \sim \mathcal{N}(-3, 0.15^2)$, $i = 1, \ldots, n$.

5.3 Results

We observed the competitiveness of IRF against CCRM and RF in Settings 1 and 4. Specifically, in Setting 1 where both the center and range exhibit positive linearity, CCRM achieved the best metrics, while IRF and RF were close behind. In Setting 4, which demonstrates a close-to-linear relationship, IRF outperformed the other models, as evident in Table 5.1. The results from Settings 2 and 3 highlight the limitations of nonnegativity in CCRM, while IRF and RF competed closely with each other. Notably, RF's performance improved with increasing sample size.

An interesting phenomenon we have observed in the simulation is that the IRF seems to make more efficient use of the data than the separate RF. We can see for most of the settings that IRF achieves the optimal performance much faster than RF. Especially, for the scenarios where they are supposed to perform competitively, RF always starts inferior and catches up later for large sample size (≥ 500). In Table 5.2, Setting 5 is a good example that well demonstrated such a phenomenon. While RF is noncompetitive to IRF for small sample sizes, it surpasses IRT and becomes somewhat competitive with IRF when the sample size grows to 500. It is expected that if we simulated even more observations, we could see RF being more competitive against IRF.

The inclusion of an interaction in Setting 6 revealed the limitations of separate RF models. While CCRM's poor performance was expected, RF also struggled in this setting. On the other hand, both interval-valued tree-based methods performed well, with IRF outperforming IRT. In the multivariable Setting 7, IRF once again outperformed all other models. CCRM performed extremely poorly, while RF remained competitive against IRT.

	$TS\text{-}MSE_\omega$				$TS\text{-}MAE_\omega$				$C\text{-}R^2$			
N	IRF	IRT	RF	CCRM	IRF	IRT	RF	CCRM	IRF	IRT	RF	CCRM
Setting 1												
100	0.18	0.39	0.27	**0.13**	0.39	0.69	0.54	**0.38**	0.91	0.80	0.87	**0.93**
250	0.15	0.30	0.19	**0.13**	**0.36**	0.62	0.45	0.37	0.92	0.85	0.90	**0.94**
500	0.15	0.27	0.17	**0.12**	0.38	0.58	0.42	**0.36**	0.92	0.86	0.92	**0.94**
Setting 2												
100	**0.16**	0.35	0.21	1.08	**0.38**	0.66	0.48	1.06	**0.92**	0.82	0.89	0.46
250	**0.14**	0.28	**0.14**	1.05	**0.36**	0.59	0.40	1.05	**0.93**	0.86	**0.93**	0.48
500	0.14	0.25	**0.12**	1.07	**0.38**	0.58	**0.38**	1.06	0.93	0.87	**0.94**	0.47
Setting 3												
100	**0.18**	0.38	0.24	1.15	**0.40**	0.66	0.50	1.03	**0.91**	0.81	0.88	0.43
250	**0.15**	0.30	0.16	1.08	**0.37**	0.59	0.42	1.01	**0.92**	0.85	**0.92**	0.46
500	0.15	0.26	**0.13**	1.04	**0.39**	0.57	**0.39**	1.00	0.92	0.87	**0.94**	0.48
Setting 4												
100	**0.24**	0.46	0.33	**0.24**	**0.42**	0.69	0.57	0.50	**0.88**	0.77	0.83	**0.88**
250	**0.18**	0.33	0.24	0.25	**0.35**	0.60	0.48	0.48	**0.91**	0.83	0.88	0.88
500	**0.15**	0.28	0.20	0.24	**0.37**	0.56	0.46	0.48	**0.92**	0.86	0.90	0.88

Table 5.1: Averaged accuracy metrics from Monte Carlo simulations of Settings 1-4.

	$TS\text{-}MSE_\omega$				$TS\text{-}MAE_\omega$				C-R^2			
N	IRF	IRT	RF	CCRM	IRF	IRT	RF	CCRM	IRF	IRT	RF	CCRM
Setting 5												
100	**0.14**	0.30	0.41	1.20	**0.39**	0.61	0.68	1.23	**0.93**	0.85	0.79	0.40
250	**0.12**	0.24	0.23	1.19	**0.38**	0.56	0.51	1.23	**0.94**	0.88	0.88	0.41
500	**0.13**	0.22	0.17	1.18	**0.40**	0.54	0.45	1.23	**0.93**	0.89	0.91	0.41
Setting 6												
100	**0.33**	0.52	1.44	2.11	**0.41**	0.64	1.14	1.49	**0.84**	0.74	0.28	-0.06
250	**0.19**	0.37	1.22	2.01	**0.37**	0.60	1.06	1.48	**0.91**	0.81	0.39	-0.01
500	**0.16**	0.30	1.14	1.92	**0.38**	0.57	1.01	1.46	**0.92**	0.85	0.43	0.04
Setting 7												
100	**0.31**	0.56	0.68	11.11	**0.43**	0.74	0.81	3.41	**0.84**	0.72	0.66	-4.56
250	**0.20**	0.38	0.40	10.64	**0.35**	0.63	0.62	3.36	**0.90**	0.81	0.80	-4.32
500	**0.18**	0.33	0.28	11.50	**0.38**	0.60	0.52	3.34	**0.91**	0.84	0.86	-4.25

Table 5.2: Averaged accuracy metrics from Monte Carlo simulations of Settings 5-7

CHAPTER 6

REAL DATA STUDY

A real dataset is analyzed using the IRF regression model to demonstrate its applicability. The dataset comprises daily [min, max] stock price ranges for five companies: Boeing Aircraft Manufacturing Company (BA), General Electric (GE), JPMorgan Chase (JPM), Procter and Gamble (PG), and Microsoft (MSFT), as well as the Dow Jones Industrial Average index (DJIA). There were a total of 1509 price intervals for each asset, spanning from January 3rd, 2012, to December 30th, 2017. The data was divided into a training set of 1207 intervals (80%) and a test set of 302 intervals (20%).

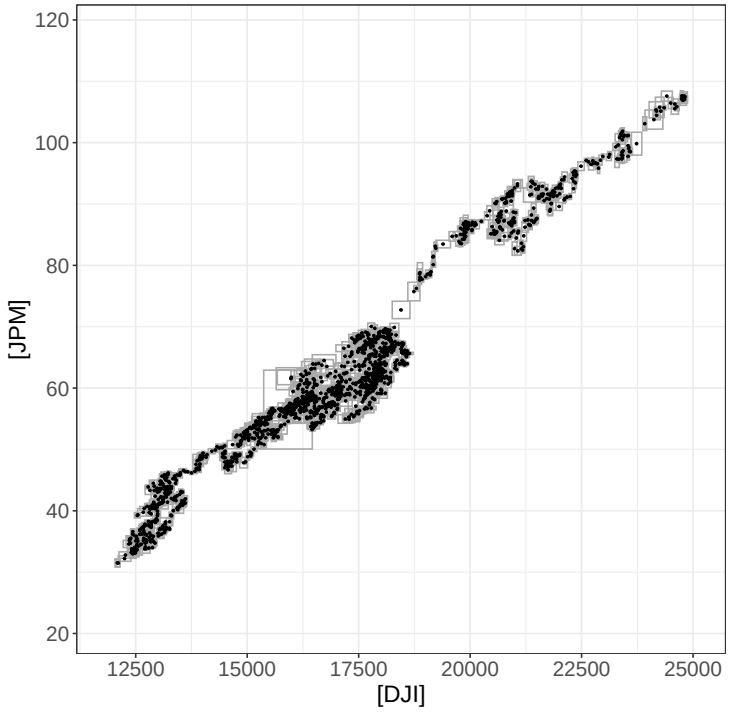

Fig. 6.1: Scatterplot of centers and range of JPM vs DJIA, where a linear relationship is shown

DJIA is a stock market index created by Charles Dow, the editor of the Wall Street

Journal, and co-founder of Dow Jones & Company. It aims to demonstrate the trading activities of 30 large, publicly owned companies based in the United States during a standard trading session in the stock market [48]. In our analysis, the DJIA is initially used as the sole variable to predict each of the three individual stocks. As shown in Figures 6.1, 6.2, and 6.3, JPM (along with other stocks) exhibits a fairly linear relationship with the DJIA index. Therefore, CCRM serves as the baseline model. To compare the results of the IRF model with those of CCRM, RF, and IRT models, the same accuracy measures used in the simulation study of Section 5.1 will be employed. Finally, a multiple-variable model using interval-valued data for BA, GE, JPM, PG, and MSFT will be used to predict the DJIA.

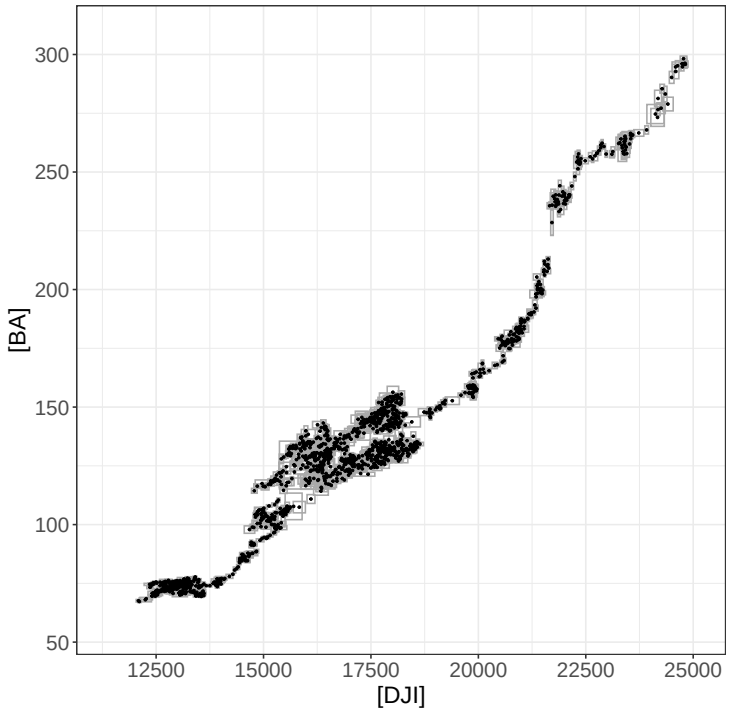

Fig. 6.2: Scatterplot of centers and range of BA vs DJIA, where a close-to-linear relationship is shown

BA, GE, JPM, PG, and MSFT are five leading companies in their respective industries. Therefore, these stocks have the potential to explain a significant portion of the variability in the DJIA index. The predictive model for the DJIA is formulated using the five stock

price intervals. The multivariate analysis is conducted similarly to the previous analysis, with both the centers and ranges of the stocks utilized as predictors for the center and range of the DJIA index.

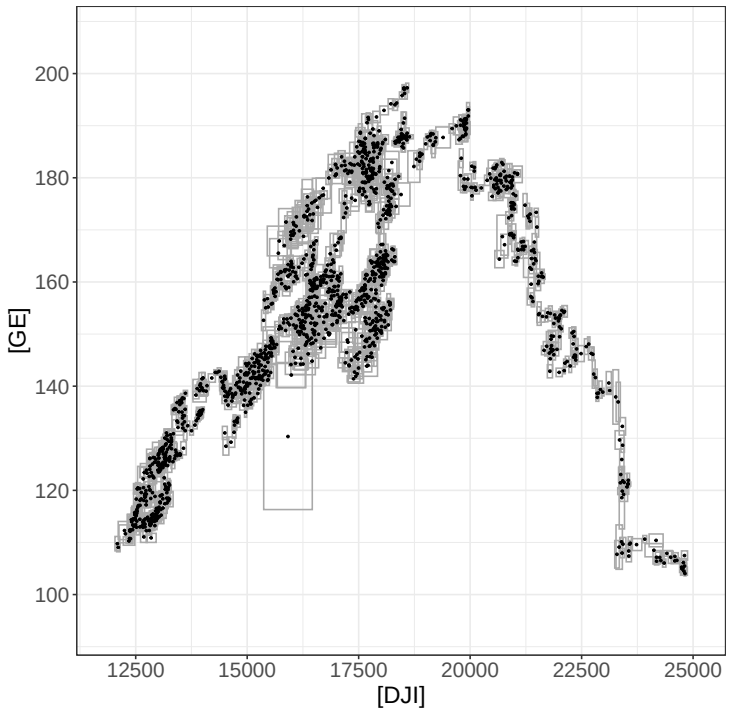

Fig. 6.3: Scatterplot of centers and range of GE vs DJIA, where a non-linear relationship is shown

6.1 Results

Individual predictive results demonstrate that our IRT and IRF models consistently outperform or compete with traditional RF and CCRM, despite the linear or close-to-linear trends observed in Figs. 6.1 and 6.2. Specifically, when predicting JPM, IRT competes well with RF and CCRM, while IRF outperforms all models. In the case of predicting BA, the interval-valued tree-based methods significantly outperform CCRM and RF. Similarly, the accuracy metrics for interval-valued tree-based methods surpass other models in predicting GE. In the multiple-variable setting for predicting DJIA, IRF and IRT outperform RF and

CCRM. However, RF performs better than IRT in terms of $TS\text{-}MSE_\omega$. Despite CCRM's efforts to capture as much information in the linear-appearing and multiple-variable setting, it struggles to compete with RF. The IRF model consistently achieves better performance than CCRM, traditional RF, and IRT. This can be attributed to RF' ability to account for nonlinearity, correlations between predictor variables, and, more importantly, correlations between outcome variables. This trend persists across all analyses, with the IRF model consistently yielding the best results.

	$TS\text{-}MSE_\omega$				$TS\text{-}MAE_\omega$				C-R^2			
Real Data	IRF	IRT	RF	CCRM	IRF	IRT	RF	CCRM	IRF	IRT	RF	CCRM
Stock												
JPM	**0.13**	0.41	0.56	0.67	**0.39**	0.49	0.62	0.74	**0.93**	0.79	0.72	0.66
BA	**0.15**	0.32	0.70	1.09	**0.33**	0.51	0.61	0.84	**0.92**	0.84	0.65	0.45
GE	**0.12**	0.36	0.60	1.27	**0.40**	0.50	0.75	1.13	**0.94**	0.82	0.70	0.37
DJIA	**0.08**	0.23	0.33	0.56	**0.32**	0.55	0.47	0.63	**0.96**	0.88	0.83	0.72

Table 6.1: Real data: Predicting index to stock price and multivariate stock prices predicting index.

CHAPTER 7

CONCLUSION

We proposed a RF regression model for interval-valued data, by considering the observed intervals as realizations of a random interval and utilizing the random sets theory. This set us to find the Aumann expectation and variance of an interval that allowed us to formulate a new splitting criterion for the RF model to jointly consider the centers and ranges, rather than creating separate models.

The proposed IRF model demonstrates robustness against overfitting, outliers, and high-dimensionality, making it more flexible than previous interval-valued regression models. It also retains the user-friendliness of the traditional RF model. The rigidity imposed by previous linear regression models for interval-valued data is eliminated due to the non-parametric nature of our method, which automatically ensures mathematical coherence.

The empirical results from both simulation and real stock market data highlight the effectiveness of the proposed IRF method in modeling and predicting interval-valued data. It consistently outperforms or competes with other traditional models for interval-valued data.

One notable result worth noting is that our proposed model demonstrates greater data efficiency compared to the separate RF model. This is demonstrated by the settings in which both models are expected to be competitive against each other. IRF achieves optimal performance with less data than the separate RF (Tables 5.1 and 5.2, specifically Setting 5), which requires a larger sample size to achieve competitive results. A deeper investigation for a more in-depth conclusion may be suitable.

Further development of the model, including accounting for the correlation between the centers and range, as well as evaluating variable importance, will enhance the advantages of the IRF model and establish it as an important tool in the analysis of interval-valued data.

REFERENCES

[1] R. H. Hariri, E. M. Fredericks, and K. M. Bowers, "Uncertainty in big data analytics: survey, opportunities, and challenges," *Journal of Big Data*, vol. 6, no. 1, p. 44, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0206-3

[2] T. Brakenhoff, M. van Smeden, F. Visseren, and R. Groenwold, "Random measurement error: Why worry? an example of cardiovascular risk factors," *PLOS ONE*, vol. 13, p. e0192298, 02 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0192298

[3] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015. [Online]. Available: https://doi.org/10.1016/j.ijinfomgt.2014.10.007

[4] P. Diamond, "Least squares fitting of compact set-valued data," *Journal of Mathematical Analysis and Applications*, vol. 147, no. 2, pp. 351–362, 1990. [Online]. Available: https://doi.org/10.1016/0022-247X(90)90353-H

[5] R. Körner and W. Näther, "Linear regression with random fuzzy variables: extended classical estimates, best linear estimates, least squares estimates," *Information Sciences*, vol. 109, no. 1, pp. 95–118, 1998. [Online]. Available: https://doi.org/10.1016/S0020-0255(98)00010-3

[6] M. A. Gil, M. A. Lubiano, M. Montenegro, and M. T. López, "Least squares fitting of an affine function and strength of association for interval-valued data," *Metrika*, vol. 56, no. 2, pp. 97–111, 2002. [Online]. Available: https://doi.org/10.1007/s001840100160

[7] M. A. Gil, G. González-Rodríguez, A. Colubi, and M. Montenegro, "Testing linear independence in linear models with interval-valued data," *Computational Statistics & Data Analysis*, vol. 51, pp. 3002–3015, 02 2007. [Online]. Available: https://doi.org/10.1016/j.csda.2006.01.015

[8] G. González-Rodríguez, Á. Blanco, N. Corral, and A. Colubi, "Least squares estimation of linear regression models for convex compact random sets," *Advances in Data Analysis and Classification*, vol. 1, no. 1, pp. 67–81, 2007. [Online]. Available: https://doi.org/10.1007/s11634-006-0003-7

[9] A. Blanco, N. Corral, and G. González-Rodríguez, "Estimation of a flexible simple linear model for interval data based on set arithmetic," *Computational Statistics & Data Analysis*, vol. 55, pp. 2568–2578, 09 2011. [Online]. Available: https://doi.org/10.1016/j.csda.2011.03.005

[10] M. E. Cattaneo and A. Wiencierz, "Likelihood-based imprecise regression," *International Journal of Approximate Reasoning*, vol. 53, no. 8, pp. 1137–1154, 2012, imprecise Probability: Theories and Applications (ISIPTA'11). [Online]. Available: https://doi.org/10.1016/j.ijar.2012.06.010

[11] F. Carvalho, E. Lima Neto, and C. Tenorio, "A new method to fit a linear regression model for interval-valued data," in *KI 2004: Advances in Artificial Intelligence*, vol. 3238, 09 2004, pp. 295–306. [Online]. Available: https://doi.org/10.1007/978-3-540-30221-6_23

[12] L. Billard, *Dependencies and Variation Components of Symbolic Interval-Valued Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 3–12. [Online]. Available: https://doi.org/10.1007/978-3-540-73560-1_1

[13] E. de A. Lima Neto and F. de A.T. de Carvalho, "Centre and range method for fitting a linear regression model to symbolic interval data," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1500–1515, 2008. [Online]. Available: https://doi.org/10.1016/j.csda.2007.04.014

[14] ——, "Constrained linear regression models for symbolic interval-valued variables," *Computational Statistics & Data Analysis*, vol. 54, no. 2, pp. 333–347, 2010. [Online]. Available: https://doi.org/10.1016/j.csda.2009.08.010

[15] P. Cazes, A. Douzal, E. Diday, and Y. Schektman, "Extensions de l'analyse en composantes principales à des données de type intervalle," *Revue de Statistique Appliquée*, vol. XIV, pp. 5–24, 01 1997. [Online]. Available: http://www.numdam.org/article/RSA_1997__45_3_5_0.pdf

[16] B. Bean, Y. Sun, and M. Maguire, "Interval-valued kriging for geostatistical mapping with imprecise inputs," *International Journal of Approximate Reasoning*, vol. 140, pp. 31–51, 2022. [Online]. Available: https://doi.org/10.1016/j.ijar.2021.10.003

[17] M. Xu and Z. Qin, "Bayesian framework for interval-valued data using jeffreys' prior and posterior predictive checking methods," *Communications in Statistics - Simulation and Computation*, vol. 0, no. 0, pp. 1–19, 2022. [Online]. Available: https://doi.org/10.1080/03610918.2022.2076869

[18] S. Dawn and S. Bandyopadhyay, "Iv-gnn : interval valued data handling using graph neural network," *Applied Intelligence*, 2022. [Online]. Available: https://doi.org/10.1007/s10489-022-03780-1

[19] J. Liu, P. Wang, H. Chen, and J. Zhu, "A combination forecasting model based on hybrid interval multi-scale decomposition: Application to interval-valued carbon price forecasting," *Expert Systems with Applications*, vol. 191, p. 116267, 2022. [Online]. Available: https://doi.org/10.1016/j.eswa.2021.116267

[20] E. a. Leo Breiman, Jerome Friedman, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984. [Online]. Available: https://doi.org/10.1201/9781315139470

[21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[22] L. Billard and E. Diday, "Regression analysis for interval-valued data," *Symbolic Data Analysis: Conceptual Statistics and Data Mining*, pp. 369–374, 01 2000. [Online]. Available: https://doi.org/10.1007/978-3-642-59789-3_58

[23] R. A. Fagundes, R. M. de Souza, and F. J. A. Cysneiros, "Interval kernel regression," *Neurocomputing*, vol. 128, pp. 371–388, 2014. [Online]. Available: https://doi.org/10.1016/j.neucom.2013.08.029

[24] L. Kong, X. Song, and X. Wang, "Nonparametric regression for interval-valued data based on local linear smoothing approach," *Neurocomputing*, vol. 501, pp. 834–843, 2022. [Online]. Available: https://doi.org/10.1016/j.neucom.2022.06.073

[25] J. E. Chacón and O. Rodríguez, "Regression models for symbolic interval-valued variables," *Entropy*, vol. 23, no. 4, 2021. [Online]. Available: https://doi.org/10.3390/e23040429

[26] C. Bertoluzza, N. Corral, and A. Salas, "On a new class of distances between fuzzy numbers," *Mathware & Soft Computing*, vol. 2, pp. 71–84, 01 1995. [Online]. Available: http://dmle.icmat.es/revistas/detalle.php?numero=1812

[27] G. Debreu *et al.*, "Integration of correspondences," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 2, no. part 1, pp. 351–372, 1967. [Online]. Available: https://digitalassets.lib.berkeley.edu/math/ucb/text/math_s5_v2_p1_article-21.pdf

[28] T. Mesikepp, "M-addition," *Journal of Mathematical Analysis and Applications*, vol. 443, no. 1, pp. 146–177, 2016. [Online]. Available: https://doi.org/10.1016/j.jmaa.2016.05.011

[29] I. Molchanov, "Theory of random sets," *Probability and Its Applications*, 01 2005. [Online]. Available: https://doi.org/10.1007/1-84628-150-4

[30] R. J. Aumann, "Integrals of set-valued functions," *Journal of Mathematical Analysis and Applications*, vol. 12, no. 1, pp. 1–12, 1965. [Online]. Available: https://doi.org/10.1016/0022-247X(65)90049-1

[31] M. Hukuhara, "Integration des applications mesurables dont la valeur est un compact convexe," *Funkcialaj Ekvacioj*, vol. 10, no. 3, pp. 205–223, 1967. [Online]. Available: http://fe.math.kobe-u.ac.jp/FE/FE_pdf_with_bookmark/fr/fe10-205-223/fe10-205-223.pdf

[32] L. Stefanini, "A generalization of hukuhara difference and division for interval and fuzzy arithmetic," *Fuzzy Sets and Systems*, vol. 161, pp. 1564–1584, 06 2010. [Online]. Available: https://doi.org/10.1016/j.fss.2009.06.009

[33] H. Rådström, "An embedding theorem for spaces of convex sets," *Proc. Amer. Math. Soc.*, vol. 3, pp. 165–169, 1952. [Online]. Available: https://doi.org/10.1090/S0002-9939-1952-0045938-2

[34] H. Hörmander, "Sur la fonction d'appui des ensembles convexes dans un espace localement convexe," *Arkiv för Mat*, vol. 3, pp. 181–186, 1954. [Online]. Available: https://doi.org/10.1007/BF02589354

[35] M. Á. Gil, M. T. López-García, M. A. Lubiano, and M. Montenegro, "Regression and correlation analyses of a linear relation between random intervals," *Test*, vol. 10, no. 1, pp. 183–201, 2001. [Online]. Available: https://doi.org/10.1007/BF02595831

[36] W. Trutschnig, G. González-Rodríguez, A. Colubi, and M. Á. Gil, "A new family of metrics for compact, convex (fuzzy) sets based on a generalized concept of mid and spread," *Information Sciences*, vol. 179, no. 23, pp. 3964–3972, 2009. [Online]. Available: https://doi.org/10.1016/j.ins.2009.06.023

[37] R. Körner and W. Näther, *On the variance of random fuzzy variables*, C. Bertoluzza, M.-Á. Gil, and D. A. Ralescu, Eds. Heidelberg: Physica-Verlag HD, 2002. [Online]. Available: https://doi.org/10.1007/978-3-7908-1800-0_2

[38] Y. Sun, "Linear regression with interval-valued data," *WIREs Computational Statistics*, vol. 8, no. 1, pp. 54–60, 2016. [Online]. Available: https://doi.org/10.1002/wics.1373

[39] J. N. Morgan and J. A. Sonquist, "Problems in the analysis of survey data, and a proposal," *Journal of the American Statistical Association*, vol. 58, no. 302, pp. 415–434, 1963. [Online]. Available: https://doi.org/10.1080/01621459.1963.10500855

[40] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. [Online]. Available: https://doi.org/10.1007/BF00058655

[41] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997. [Online]. Available: https://doi.org/10.1006/jcss.1997.1504

[42] M. Segal and Y. Xiao, "Multivariate random forests," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 80–87, 2011. [Online]. Available: https://doi.org/10.1002/widm.12

[43] R. Rahman, J. Otridge, and R. Pal, "Integratedmrf: Random forest-based framework for integrating prediction from different data types," *Bioinformatics (Oxford, England)*, vol. 33, 02 2017. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw765

[44] C. A. V. d. S. F. Eufrasio de A. Lima Neto and P. R. D. Marinho, *iRegression: Regression Methods for Interval-Valued Variables*, 2016. [Online]. Available: https://CRAN.R-project.org/package=iRegression

[45] A. Liaw and M. Wiener, *Classification and Regression by randomForest*, 2002. [Online]. Available: https://cran.r-project.org/package=randomForest

[46] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022. [Online]. Available: https://www.R-project.org/

[47] rpart by Terry M Therneau, B. A. R. port of rpart by Brian Ripley. Some routines from vegan by Jari Oksanen. Extensions, and adaptations of rpart to mvpart by Glenn De'ath., *mvpart: Multivariate partitioning*, 2014. [Online]. Available: https://CRAN.R-project.org/package=mvpart

[48] Research guides: This month in business history: Dow jones industrial average first published. [Online]. Available: https://guides.loc.gov/this-month-in-business-history/may/djia-first-published

APPENDICES

APPENDIX A

R Implementation

In this appendix, the three main R scripts for obtaining the accuracy metrics, as well as the code for plotting simulated data, are provided. The functionality for the IRT can be accessed at the following link:

https://github.com/PaulGaona/IntRF.

All additional files can be found at:

https://github.com/PaulGaona/IntRf_Application.

A.1 Settings 1 -7

These files are designed to conduct a simulation study with the objective of comparing the performance of different models and visualizing the relationships among variables in various settings. The simulation is conducted for different sample sizes, and it is repeated for each combination of setting and sample size. There are seven different settings defined by specific parameters passed to the functions `set1` to `set7`. Each set function returns a list of data frames, with each data frame containing the simulated data for one run of the simulation.

The performance of each model is assessed using the testing data. The average results across all simulation runs for each combination of setting and sample size are obtained. The results can be accessed by indexing the variable *all_res* with the appropriate sample size and setting name. For example, *all_res[[1]]$Setting1* contains the results for the first sample size (n = 100) and the first setting (Setting1).

Simulation: ("*all_setting_code.R*")

The data is then split into training and testing sets. The training data is fitted into the following models:

- IRF (Interval-Valued Random Forests)
- IRT (Interval-Valued Regression Tree)
- RF (Random Forests)
- and CCRM (Constrained Center and Range Model).

```
# Load necessary functions and libraries
source("./Analysis/Simulation/Settings.R")
source("./Functions/Auto_Models.R")
source("./Functions/Dat_Split.R")
source("./Functions/Res_Avg_Set.R")
source("./Functions/CCRM_Pred.R")
library("tidyverse")
library("IntRF")

# Set up simulation parameters
n_vec <- c(100, 250, 500) # Sample sizes to simulate
mc_sim <- 100 # Number of Monte Carlo simulations to run

# Initialize empty list to store results
all_res <- vector("list", length(n_vec))
names(all_res) <- paste("n=", n_vec)

# Loop over all elements of the n_vec vector
for (i in seq(all_res)) {
  set.seed(1)
  # Generate a list of simulated data sets for each
  # of the 7 different settings
  list_sims <- list(
    Setting1 = replicate(n = mc_sim, expr = set1(
      n = n_vec[i],
```

```
  Xc_a = 12, Xc_b = 3,
  ec_a = 0, ec_b = 3 / 4,
  Xr_a = 1, Xr_b = 3,
  er_a = 0, er_b = 1 / 20
)),
Setting2 = replicate(n = mc_sim, expr = set2(
  n = n_vec[i],
  Xc_a = -5, Xc_b = 10,
  ec_a = 0, ec_b = 10 / 4,
  Xr_a = 1, Xr_b = 2,
  er_a = 0, er_b = 1 / 10
)),
Setting3 = replicate(n = mc_sim, expr = set3(
  n = n_vec[i],
  Xc_a = 5, Xc_b = 1,
  ec_a = 0, ec_b = 1,
  Xr_a = .1, Xr_b = .25,
  er_a = 0, er_b = 0.25
)),
Setting4 = replicate(n = mc_sim, expr = set4(
  n = n_vec[i],
  Xc_a = 12, Xc_b = 4,
  ec_a = 0, ec_b = 1,
  Xr_a = .5, Xr_b = 2,
  er_a = 0, er_b = .5 / 4
)),
Setting5 = replicate(n = mc_sim, expr = set5(
  n = n_vec[i],
  Xc_a = 3, Xc_b = 2,
  ec_a = 0, ec_b = 1.5,
  Xr_a = 0.5, Xr_b = 1,
  er_a = 0, er_b = 1 / 16
```

```
  )),
  Setting6 = replicate(n = mc_sim, expr = set6(
    n = n_vec[i],
    Xc_a = 8, Xc_b = 2,
    ec_a = 0, ec_b = 1,
    Xr_a = 0, Xr_b = 1,
    er_a = 0, er_b = 0.25
  )),
  Setting7 = replicate(n = mc_sim, expr = set7(
    n = n_vec[i],
    X1c_a = 5, X1c_b = 3,
    X1r_a = NA, X1r_b = NA,
    X2c_a = .5, X2c_b = .5,
    X2r_a = NA, X2r_b = NA,
    X3c_a = 10, X3c_b = 3.5,
    X3r_a = 10, X3r_b = 3,
    X4c_a = .5, X4c_b = 1.5,
    X4r_a = 2.5, X4r_b = 3.5,
    X5c_a = 8, X5c_b = 3.5,
    X5r_a = 2, X5r_b = 5,
    ec_a = 0, ec_b = 1,
    er_a = -3, er_b = 15,
    tau1_a = 0, tau1_b = .2,
    tau2_a = 0, tau2_b = .2,
    un1_a = 0, un1_b = .5,
    un2_a = 0, un2_b = .5,
    v1_a = 3, v1_b = 2,
    v2_a = 1, v2_b = 3
  ))
)

# Convert the list of simulations into a list
```

```
  # of data frames
  list_sims_df <- lapply(list_sims, data.frame)

  # Convert each element of the list of data frames
  # into a nested
  # list of data frames
  list_sims_df2 <- lapply(list_sims_df, function(x) {
    lapply(x, data.frame)
  })

  # Call the auto_models function on each nested list
  # of data frames and store the results in a list
  set.seed(1)
  list_models <- lapply(list_sims_df2, function(x) {
    lapply(x, auto_models)
  })

  # Store the results for the i-th simulation in the i-th
  # element of the all_res list
  all_res[[i]] <- list_res(list_models)
}
all_res
```

Figures (“*Settings123_Plot.R*”)

This code demonstrates how to create a figure based on the simulated data, using the setting and parameters described in Section 5.2. The file generates figures for settings 1, 2, and 3, while `Settings456_Plot.R` generates figures for settings 4, 5, and 6. The example provided includes only the essential code to generate Figure 5.1.

```
# Load necessary packages
library("ggplot2")
```

```r
library("dplyr")
library("IntRF")
library("ggpubr")
library("scales")
library("gridExtra")
source("./Analysis/Simulation/Settings.R")
set.seed(1)
# Setting 1 data simulating
df_set1 <- set1(
  n = 500,
  Xc_a = 12, Xc_b = 3,
  ec_a = 0, ec_b = 3 / 4,
  Xr_a = 1, Xr_b = 3,
  er_a = 0, er_b = 1 / 20
)
# Plot Y vs X
set1_p <- IntRF::int_plot(
  # Select relevant columns from prices dataframe
  int_data = df_set1 %>%
    dplyr::select(Yc, Yr, Xc, Xr),
  title = "Setting1",
  xlabel = "[X]",
  ylabel = "[Y]"
) +
  ggplot2::scale_x_continuous(
    limits = c(
      # Set lower limit of x-axis
      min(df_set1$Xc - df_set1$Xr) - 1,
      # Set upper limit of x-axis
      max(df_set1$Xc + df_set1$Xr) + 1
    ),
    # Set number of x-axis breaks to 8
```

```
    n.breaks = 8,
    # Wrap x-axis labels to fit within 6 lines
    labels = scales::label_wrap(6)
  ) +
  ggplot2::scale_y_continuous(
    limits = c(
      # Set lower limit of y-axis
      min(df_set1$Yc - df_set1$Yr) - 1,
      # Set upper limit of y-axis
      max(df_set1$Yc + df_set1$Yr) + 1
    ),
    # Set number of y-axis breaks to 8
    n.breaks = 8
  ) +
  # Set title and text size of x and y axis
  theme(
    axis.title.x = element_text(size = 16),
    axis.text.x = element_text(size = 14),
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14)
  )

set1_p
```

A.2 Real Data: Individual Stocks

A similar process described in Appendix A.1 can be applied to predict the stock prices of various companies using the DJIA (Dow Jones Industrial Average) as the predictor variable. The same process is then repeated for predicting the stock prices of Boeing Co. (BA) and General Electric (GE). Additionally, in Appendix A.3, the prediction of DJIA using JPM, BA, GE, PG, and MSFT as predictors is also discussed.

Plotting Data ("*Stock_Plot.R*")

Specifically, this example, is training and evaluating models to predict the stock price of JPMorgan Chase & Co. (JPM).

```
# Load necessary packages
library("ggplot2")
library("dplyr")
library("IntRF")
library("ggpubr")
library("scales")
library("gridExtra")

# Plot JPM vs DJI
jpm_p <- IntRF::int_plot(
  # Select relevant columns from prices dataframe
  int_data = prices %>%
    dplyr::select(c.JPM, r.JPM, c.DJI, r.DJI),
  title = "",
  xlabel = "[DJI]",
  ylabel = "[JPM]"
) +
  ggplot2::scale_x_continuous(
    limits = c(
      # Set lower limit of x-axis
      min(prices$c.DJI) - 250,
      # Set upper limit of x-axis
      max(prices$c.DJI) + 250
    ),
    # Set number of x-axis breaks to 10
    n.breaks = 6,
    # Wrap x-axis labels to fit within 6 lines
    labels = scales::label_wrap(6)
```

```
  ) +
  ggplot2::scale_y_continuous(
    limits = c(
      # Set lower limit of y-axis
      min(prices$c.JPM) - 10,
      # Set upper limit of y-axis
      max(prices$c.JPM) + 10
    ),
    # Set number of y-axis breaks to 8
    n.breaks = 8
  ) +
  # Set title and text size of x and y axis
  theme(
    axis.title.x = element_text(size = 16),
    axis.text.x = element_text(size = 14),
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14)
  )
jpm_p

# export jpm
pdf(file = "./Analysis/Real/figs/jpm_fig.pdf")
jpm_p +
  theme(aspect.ratio=1)
dev.off()
```

Building the Models ("*Predicting_Stocks.R*")

Specifically, this example, is training and evaluating models to predict the stock price of JPMorgan Chase & Co. (JPM).

```
# Sourcing functions required
```

```
source("./Functions/CCRM_Pred.R")

# stock locations
# DJI : 1,7
# JPM : 4, 10
# BA : 5, 11
# GE : 2, 8

# Define the stock locations for each stock in the dataset

# JPM ~ DJI

# Data from Stock_Code.R
# train data

# Load in the training and testing data for the selected
# stock
price_train_jpm <- price_train_stand[c(4, 10, 1, 7)]
price_test_jpm <- price_test_stand[c(4, 10, 1, 7)]

# Split the training data into the response variable (price)
# and predictor variables (center and range values)
yprice_train_jpm <- price_train_jpm[c(1, 2)]
xcprice_train_jpm <- price_train_jpm[3]
xrprice_train_jpm <- price_train_jpm[4]

# Split the testing data into the response variable (price)
# and predictor variables (center and range values)
yprice_test_jpm <- price_test_jpm[c(1, 2)]

# Int RF
# Package
```

```
# Use the IntRF package to train a RF model for
# interval regression
set.seed(1)
int_price_rf_jpm <- IntRF::intrf(
  int_resp = yprice_train_jpm,
  cent_pred = xcprice_train_jpm,
  ran_pred = xrprice_train_jpm,
  train = price_train_jpm,
  test = price_test_jpm,
  mtry_int = ncol(xcprice_train_jpm)
)

# Extract the results from the model
res_jpm <- int_price_rf_jpm$Results

# Calculate accuracy metrics for the model
met_irf_jpm <- IntRF::acc_met(
  cent_pred = res_jpm$center_pred,
  cent_act = res_jpm$center_actual,
  ran_pred = res_jpm$range_pred,
  ran_act = res_jpm$range_actual,
  yprice_train_jpm
)

# Output the accuracy metrics
met_irf_jpm

# Int Tree
# IRT

# Use the mvpart function to train an interval regression
```

```
# tree model
set.seed(1)
ydat_jpm <- price_train_stand[names(yprice_train_jpm)]
irt_jpm <- IntRF::mvpart(data.matrix(ydat_jpm) ~ .,
  data = price_train_jpm,
  plot.add = FALSE,
  xv = "none"
)

# Make predictions using the model on the testing data
ctpred_jpm <- predict(irt_jpm,
  newdata = price_test_jpm,
  type = "matrix"
)[, 1]
rtpred_jpm <- predict(irt_jpm,
  newdata = price_test_jpm,
  type = "matrix"
)[, 2]

# Calculate accuracy metrics for the model
met_tree_jpm <- IntRF::acc_met(
  ctpred_jpm,
  t(yprice_test_jpm[1]),
  rtpred_jpm,
  t(yprice_test_jpm[2]),
  yprice_train_jpm
)

# Output the accuracy metrics
met_tree_jpm

# RF model
```

```
set.seed(1)
# create RF model for JPM stock price using
# all other variables
# except actual stock price
crf_jpm <- randomForest::randomForest(c.JPM ~ .,
  data = dplyr::select(price_train_jpm, -c(r.JPM))
)

set.seed(1)
# create RF model for JPM stock price using all
# other variables
# except actual stock price
rrf_jpm <- randomForest::randomForest(r.JPM ~ .,
  data = dplyr::select(price_train_jpm, -c(c.JPM))
)

# Make predictions using the model on the testing data
pcrf_jpm <- predict(crf_jpm, price_test_jpm)
prrf_jpm <- predict(rrf_jpm, price_test_jpm)

# calculate accuracy metrics for the RF models
met_rf_jpm <- IntRF::acc_met(
  pcrf_jpm,
  t(yprice_test_jpm[1]),
  prrf_jpm,
  t(yprice_test_jpm[2]),
  yprice_train_jpm
)

# output accuracy metrics for RF models
met_rf_jpm
```

```
# ccrm model
set.seed(1)

# create CCRM model for the JPM stock price as a function
# of the DJI stock price
simccrm_jpm <- iRegression::ccrm("c.JPM~c.DJI",
  "r.JPM~r.DJI",
  data = price_train_jpm
)

# predict  JPM stock price using the CCRM model
pred_ccrm_jpm <- ccrm_pred(
  cent_coef = simccrm_jpm[[1]],
  cent_pred = as.matrix(price_test_jpm[3]),
  ran_coef = simccrm_jpm[[5]],
  ran_pred = as.matrix(price_test_jpm[4])
)

# calculate accuracy metrics for the CCRM model
met_ccrm_jpm <- IntRF::acc_met(
  t(pred_ccrm_jpm$center_pred),
  t(yprice_test_jpm[1]),
  t(pred_ccrm_jpm$range_pred),
  t(yprice_test_jpm[2]),
  yprice_train_jpm
)

# combine accuracy metrics from all models into a single
# data frame for comparison
combined_res_jpm <- data.frame(
  IRF = t(met_irf_jpm),
```

```
  IRT = t(met_tree_jpm),
  RF = t(met_rf_jpm),
  CCRM = t(met_ccrm_jpm)
)

combined_res_jpm
```

A.3 Real Data: DJIA

This process is described in Appendices A.1 and A.2.

Model Metrics ("*Predicting_DJI.R*")

```
# Data from Stock_Code.R

# Load required packages and set seed value
library(IntRF)
library(randomForest)
library(iRegression)

# Perform intrf model for Int RF and obtain results
set.seed(1)

int_price_rf <- IntRF::intrf(
  int_resp = yprice_train,
  cent_pred = xcprice_train,
  ran_pred = xrprice_train,
  train = price_train_stand,
  test = price_test_stand,
  mtry_int = ncol(xcprice_train)
)

res <- int_price_rf$Results
```

```
# Obtain accuracy metrics for Int RF
met_irf_dji <- IntRF::acc_met(
  cent_pred = res$center_pred,
  cent_act = res$center_actual,
  ran_pred = res$range_pred,
  ran_act = res$range_actual,
  yprice_train
)

# Print output for Int RF accuracy metrics
met_irf_dji

# Perform mvpart model for Int Tree and obtain
# predictions
set.seed(1)

ydat <- price_train_stand[names(yprice_train)]
irt <- IntRF::mvpart(data.matrix(ydat) ~ .,
  data = price_train_stand,
  plot.add = FALSE,
  xv = "none"
)
ctpred <- predict(irt,
  newdata = price_test_stand,
  type = "matrix"
)[, 1]
rtpred <- predict(irt,
  newdata = price_test_stand,
  type = "matrix"
)[, 2]
```

```
# Obtain accuracy metrics for Int Tree
met_tree_dji <- IntRF::acc_met(
  ctpred,
  t(yprice_test[1]),
  rtpred,
  t(yprice_test[2]),
  yprice_train
)

# Print output for Int Tree accuracy metrics
met_tree_dji

# Perform randomForests model for RF and obtain
# predictions
set.seed(1)
crf <- randomForest(c.DJI ~ .,
  data = dplyr::select(price_train_stand, -c(r.DJI))
)
rrf <- randomForest(r.DJI ~ .,
  data = dplyr::select(price_train_stand, -c(c.DJI))
)
pcrf <- predict(crf, price_test_stand)
prrf <- predict(rrf, price_test_stand)

# Obtain accuracy metrics for RF
met_rf_dji <- acc_met(
  pcrf,
  t(yprice_test[1]),
  prrf,
  t(yprice_test[2]),
  yprice_train
)
```

```
# Print output for RF accuracy metrics
met_rf_dji

# Perform ccrm model and obtain predictions
set.seed(1)

simccrm <- ccrm("c.DJI~c.GE+c.PG+c.JPM+c.BA+c.MSFT",
  "r.DJI~r.GE+r.PG+r.JPM+r.BA+r.MSFT",
  data = price_train_stand
)

pred_ccrm <- ccrm_pred(
  cent_coef = simccrm[[1]],
  cent_pred = as.matrix(price_test_stand[2:6]),
  ran_coef = simccrm[[5]],
  ran_pred = as.matrix(price_test_stand[8:12])
)

# Obtain accuracy metrics for ccrm
met_ccrm_dji <- acc_met(
  pred_ccrm$center_pred,
  t(yprice_test[1]),
  pred_ccrm$range_pred,
  t(yprice_test[2]),
  yprice_train
)

# Combine accuracy metrics from all models into
# a single dataframe
combined_res_dji <- data.frame(
  IRF = t(met_irf_dji),
```

```
  IRT = t(met_tree_dji),
  RF = t(met_rf_dji),
  CCRM = t(met_ccrm_dji)
)

# output results
round(combined_res_dji, 3)
```