

On Statistical Analysis of Survival Data

(生存データの統計解析について)

Asanao Shimokawa

(下川 朝有)

Department of Mathematical Information Science
Tokyo University of Science
1-3, Kagurazaka, Shinjuku-ku Tokyo, Japan

January 30, 2015

Acknowledgements

First, I would like to express my gratitude to my supervisor Professor Etsuo Miyaoka. Thanks to him, I could learn a variety of things related to mathematical statistics, especially analysis of medical data. Whenever I faced any difficulty with statistics, he helped me with his invaluable advice. Moreover, he provided me with opportunities to present our work in both domestic and international conferences. Without Professor Miyaoka's guidance and continuing help, it would not have been possible for me to complete this thesis.

Second, I am deeply grateful to Drs. Yoshitaka Narita (M.D.) and Soichiro Shibui (M.D.). They gave me a variety of advice on brain tumors from a medical perspective. They also gave me the opportunity to analyze actual data from brain cancer patients.

I am also thankful to Drs. Shuichi Midorikawa and Yohei Kawasaki for their insightful comments and suggestions about our researches.

In addition, I would like to thank all my friends, and senior and junior colleagues at the laboratory, for their support and encouragement with my studies.

Last but not least, I would like to express my gratitude to my family for all their understanding and support throughout my studies.

Asanao Shimokawa

Contents

1	Introduction	4
1.1	Supervised learning methods for prediction	5
1.2	Statistical methods for survival data modeling	9
1.3	Statistical methods based on interval-valued variables	12
1.4	Medical image processing	15
1.5	Organization of the thesis	19
2	Application of Genetic Algorithm Based on Texture Features Obtained Through CT Images of Lung Cancer Patients	20
2.1	Introduction	20
2.2	Patients and Preprocessing	22
2.2.1	Patients	22
2.2.2	Extraction of the region of interest	22
2.3	Feature extraction	25
2.3.1	Gray-level co-occurrence matrix	25
2.3.2	Wavelet transformation	28
2.4	Feature selection and classification	31
2.4.1	Genetic algorithm	31
2.4.2	Nearest neighbor method	33
2.4.3	Evaluation function	34
2.4.4	Weighting methods	35
2.5	Results	37
2.6	Conclusion	38
3	Tree-structured Prediction Model	40
3.1	Introduction	40
3.2	Regression Tree	42
3.2.1	Splitting	42
3.2.2	Pruning	44
3.2.3	Selection	46
3.3	Survival Tree	47
3.3.1	Splitting	47
3.3.2	Pruning	50

3.3.3	Selection	51
3.4	Conclusion	52
4	Application of Survival Tree Based on Texture Features Ob- tained Through MRI Images of Brain Cancer Patients	53
4.1	Introduction	53
4.2	Patients and Methods	54
4.2.1	Patients	54
4.2.2	Image analysis	55
4.2.3	Statistical analysis	55
4.3	Results	57
4.4	Conclusion	59
5	Survival Tree Based on Combination of Covariates	61
5.1	Introduction	61
5.2	Splitting method based on combination of covariates	62
5.3	Simulation	63
5.3.1	Model and Setting	63
5.3.2	Evaluation methods	64
5.3.3	Results	65
5.4	Example	66
5.5	Conclusion	67
6	Regression Tree on Interval-valued Symbolic Data	70
6.1	Introduction	70
6.2	Methods	71
6.2.1	Predictive model in each node	72
6.2.2	Splitting criteria	74
6.2.3	Optimal splitting point	75
6.2.4	Prediction of a new data	78
6.2.5	Algorithm	78
6.3	Simulation	79
6.3.1	Model and Setting	79
6.3.2	Results	80
6.4	Example	82
6.5	Conclusion	87
	Appendix.	87
7	Concluding Remarks	90

Chapter 1

Introduction

In the field of medical research, a wide variety of factors must be taken into account for each individual patient. For example, when one examines a brain cancer patient one must consider the patient's age, gender, the number of tumors from which they suffer, the size of their largest tumor, performance status, the laterality of the tumors, their surgeries, radiotherapy, chemotherapy, expected survival time, medical images, the cancer's progression, and so on. The determination of both the patient's prognosis and treatment rely heavily on this information.

Unfortunately, a large quantity of information does not necessarily generate a useful model, that is, there are many nuisances in the data, which are difficult to account for. If we were to construct a model with this data, it could yield incorrect predictions or suggest improper courses of action. To address this problem, we must distinguish between the pertinent and unimportant information in order to construct a useful research model. While these variable selection and model construction problems have been addressed in a variety of ways, there is still ample room for improvement, as many issues remain unresolved. Consequently, we propose a new method for constructing classification and prediction models that addresses several of these problems, and compare it with a number of other approaches by using simulations and real data analysis. In particular, we focus on the genetic algorithm (GA) and the classification and regression tree (CART) algorithm, which are typical supervised learning methods.

One data source that is considered in this study is medical images, which are thought to provide a great deal of information on each patient. Therefore, another goal of this study is to evaluate the variables obtained from patients' medical images and construct a more suitable model based upon them.

1.1 Supervised learning methods for prediction

We begin with a learning set of data that is composed of outcome measurements and covariates (sometimes called features), which are usually either quantitative or categorical. Using this data, we construct a model that will enable us to predict a new patient's outcome with a process called supervised learning. On the other hand, if a learning set has no outcome measurements, the process is termed unsupervised learning.

A wide variety of statistical supervised learning methods has been proposed. If the outcome is a continuous variable, then linear regression is generally used. In this approach, we let X be the continuous outcome and $\mathbf{Z} = (Z_1, Z_2, \dots, Z_p)$ be the p -dimensional covariate vector. Then, we are able to predict the outcome with the following model

$$X = \beta_0 + \sum_{j=1}^p \beta_j Z_j,$$

where β_0 is the intercept, and β_j is the coefficient parameter. The least squares method is generally used to fit the linear model to the learning dataset. As such, we let $\mathcal{L} = \{(x_i, \mathbf{z}_i); i = 1, 2, \dots, N\}$ be an observed learning set, where $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{pi})$. If we describe $\boldsymbol{\beta}$, \mathbf{x} , and \mathbf{Z} as

$$\boldsymbol{\beta} \equiv \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \mathbf{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \mathbf{Z} \equiv \begin{bmatrix} 1 & z_{11} & \cdots & z_{p1} \\ 1 & z_{12} & \cdots & z_{p2} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & z_{1N} & \cdots & z_{pN} \end{bmatrix},$$

then, the least squares estimates of $\boldsymbol{\beta}$ are given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{x}.$$

By using to this estimate, the prediction of the outcome for a new input, \mathbf{z}_0 , can be given by $\hat{X} = \hat{\beta}_0 + \sum_{j=1}^p \hat{\beta}_j z_{j0}$. This simple method can be an effective predictor in situations where the learning dataset is small (Hastie et al., 2009).

If the outcome is a binary variable, though, the logistic regression approach is commonly employed. For this method, we let X be a binary variable with possible values of either 0 or 1. The model then takes the form:

$$\begin{aligned} \Pr(X = 1) &= \frac{\exp(\gamma_0 + \sum_{j=1}^p \gamma_j Z_j)}{1 + \exp(\gamma_0 + \sum_{j=1}^p \gamma_j Z_j)}, \\ \Pr(X = 0) &= 1 - \Pr(X = 1). \end{aligned}$$

In this case, the maximum likelihood method is used to fit the model to the

learning dataset. Here, we define $\boldsymbol{\gamma}$ and $\mathbf{Z}_{(i)}$ as

$$\boldsymbol{\gamma} \equiv \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_p \end{bmatrix}, \mathbf{Z}_{(i)} \equiv \begin{bmatrix} 1 \\ z_{1i} \\ \vdots \\ z_{pi} \end{bmatrix}.$$

By differentiating the log likelihood with respect to $\boldsymbol{\gamma}$ and then setting it equal to $\mathbf{0}$, we obtain the estimating equations

$$\sum_{i=1}^N y_i \mathbf{Z}_{(i)} = \sum_{i=1}^N \frac{\exp(\mathbf{Z}_{(i)}' \hat{\boldsymbol{\gamma}})}{1 + \exp(\mathbf{Z}_{(i)}' \hat{\boldsymbol{\gamma}})} \mathbf{Z}_{(i)},$$

where $\hat{\boldsymbol{\gamma}}$ is the maximum likelihood estimator of the coefficients. These estimating equations are nonlinear and, therefore, calculated by using an iterative method (Agresti, 2002; Fox, 2008). For more details on the logistic regression approach (including cases where the responses have multiple categories), see Agresti (2002).

A common method for predicting categorical variables is the k -nearest neighbor (k -NN) method. This classifier is memory-based, and does not require a model for fitting. To classify a new input, \mathbf{z}_0 , this method finds the k learning points, $\mathbf{z}_{(r)}$, $r = 1, 2, \dots, k$, closest in distance to \mathbf{z}_0 . Despite the fact that this method has only one parameter, it has been successful in a large number of classification problems, including those related to satellite images and handwritten digits (Hastie et al., 2009).

One weakness of this simple classifier is that it does take into account equally all of the covariates. If nuisance variables are included among the covariates, the performance of the classification degrades. Moreover, when this method is carried out in a high-dimensional covariate space (as in the case of medical research) a point's nearest neighbors are quite far away, which biases the classification. In order to control for this problem, Hastie and Tibshirani (1996) propose using a local linear discriminant analysis to estimate an effective metric for computing neighborhoods; then, by using simulation studies, they show that this method can improve classification performance. Cunningham and Delany (2007) propose the filter and wrapper approaches to reduce the dimension of the covariate space. Another approach selects the covariates (sometimes called feature selection) by combining the other supervised learning techniques. For example, Punch et al. (1993) combine the GA with a k -NN method in order to optimize their classification.

Many other supervised learning techniques have been proposed, including discriminant analysis, kernel smoothing methods, and non-linear models (like the additive model) (Hastie et al., 2009). Moreover, recent advances in computing technology have led to the development of more complex supervised

learning techniques that require numerous iterative calculations. The support vector machine (SVM) technique, for example, which can learn linear and nonlinear optimization problems by using kernel functions, is one of the most popular methods for regression and classification problems (Burges, 1998; Bishop, 2006). Another such example is the artificial neural network (ANN) approach, which optimizes linear combination models of nonlinear basis functions $\phi_m(\cdot)$, such as

$$X = g \left(\sum_{m=1}^M w_m \phi_m(\mathbf{Z}) \right),$$

where $g(\cdot)$ is a nonlinear function identified in the case of problems. The basis functions, $\phi_m(\cdot)$, are constructed to depend on the parameters, then those parameters and coefficients (w_m) are adjusted by means of network training (Bishop, 2006; Hastie et al., 2009). These techniques are often called machine learning.

Although SVM and ANN are widely used, they are not without their flaws. For example, ANN is restricted to the prediction model form by a nonlinear transformation of the sum of the basis functions. This makes the model less flexible than the other nonparametric approaches, like the k -NN. Moreover, the interpretation of these models' results is challenging, as it is difficult to understand how each covariate value contributes to the outcome. This poses a particular problem for the analysis of medical data because health care professionals rely on these results in their treatment decisions. For example, if medicine A works well in patients under 20 years of age, then older patients would be better served by taking medicine B.

Holland (1975) introduced yet another supervised learning approach called the genetic algorithm (GA), which is a computational model inspired by evolution. This algorithm encodes a potential solution to a specific problem in a chromosome-like data structure and applies recombination operators to said structure in order to preserve critical information. Although the GA is often viewed as a function optimizer, the range of problems to which it has been applied is quite broad (Whitley, 1994); for example, it can be used in classification, feature selection, ranking, and scaling. As such, this approach has been utilized in a variety of research fields (Goldberg, 1989), including biology and medicine, business, computer science, engineering and operations, machine learning, the social sciences, and so on.

The GA is implemented as follows (Whitley, 1994): First, an initial population is generated. Each member of this population is represented by a binary string that corresponds to the encoded problem. For example, suppose that our current problem is selecting features from p -dimensional covariates (\mathbf{Z}), then the binary string is encoded with p elements, each of which takes on a value of either 1 or 0 depending on whether the feature has been selected or not. Thus,

if our current problem is selecting features from Z_1, \dots, Z_{10} , then this string can be represented as

0101101000

indicating a feature space comprised of the covariates Z_2, Z_4, Z_5 , and Z_7 . Each string is referred to as a "chromosome." The execution of the GA is an iterative approach; it recursively creates a set of new chromosomes to evaluate. In Chapter 2, we use the GA to select features by which to classify lung cancer patients' data.

Binary tree-structured models may be the easiest to interpret. Tree-based models were introduced by Morgan and Sonquist (1963), and further developed by Breiman et al. (1984), who proposed the commonly used CART algorithm. The main advantage of this model is that the relationship between covariates and outcomes can be easily determined from its hierarchical structure. The goal of this model is to split the data into several groups with differing outcomes by using covariates. If the outcome of our problem is a categorical variable, then the tree-structured model is called a classification tree, and it is constructed to minimize the misclassification rate in the new data. On the other hand, if the outcome is a continuous variable, then the model is called as regression tree, and it minimizes the squared errors in the new data. A survival prediction model is called a survival tree, and various criteria for constructing such models have been proposed (Ciampi et al., 1988; Segal, 1988; Davis and Anderson, 1989; Therneau et al., 1990; Leblanc and Crowley, 1992; and Zhang, 1995). A detailed flow of the tree construction algorithm is presented in Chapter 3.

Recently, the ensemble methods that require more computational power have been studied in greater detail. It is thought that a single, effective prediction model could be built by combining the weaker prediction models. In fact, an ensemble model is obtained by evaluating the performances of several individual tree-structured models and merging them. Of course, there is no reason to restrict this methodology to tree-structured models, we can apply it to regression models, classifiers, ANN, and many others. In many studies, ensemble methods are shown to be more accurate than any of the individual models that go into them. Summaries and comparative analyses on ensemble models can be found in Opiz and Maclin (1999), Polikar (2006), and Rokach (2010). Based on this ensemble method and an iteration of the GA (presented in Chapter 2), we propose an approach for weighting features. We conduct comparative research with data on lung cancer patients to demonstrate the effectiveness of this approach.

1.2 Statistical methods for survival data modeling

Survival time analysis is an important topic in the field of medical research. Survival time is defined as the amount of time until the occurrence a certain event, such as death, disease development, or recurrence. In other words, survival time is defined as the time until death, length of remission, or tumor-free time. We let Y denote a survival time that extends from the time of origin to the occurrence of the event (sometimes it called as failure time). Then, we define the survival function as the probability that an individual survives longer than y :

$$S(y) = \Pr(Y > y).$$

In this thesis, we consider the case where Y is a continuous variable and $S(y)$ is a continuous and strictly monotonically decreasing function.

In survival data analysis, the instantaneous failure rate is given by the hazard function. The hazard function is defined as

$$\lambda(y) = \lim_{\Delta y \rightarrow 0} \frac{\Pr(y \leq Y < y + \Delta y | Y \geq y)}{\Delta y}.$$

Furthermore, the cumulative hazard function is defined as

$$\Lambda(y) = \int_0^y \lambda(u) du.$$

The hazard function, $\lambda(y)$, is a non-negative function and it can be thought of as the instantaneous probability that an event occurs at time y , given that the individual survives until then.

The survival function may be estimated by analyzing time-to-event data. However, the presence of censored data complicates this estimation. Censored data does not have the correct interval between the start point (e.g., illness detection date or surgery date) and the end point (e.g., date of death or date of recurrence). In this thesis, we deal with right censored cases as they are frequently encountered in medical survival data. The survival times of right censored patients are not actually known, but are instead recorded as being at least the length of the observation period.

In order to deal with right censored data, we denote the censoring time as C . If right censored patients are included, the data is represented as a pair of random variables, (X, δ) , where $\delta = I(Y \leq C)$ represents the censoring indicator, which is 1 if the observation is an event and 0 if the observation is censored. X represents the completion of the event/censoring time interval, and it is given by $X = \min(Y, C)$. We assume that the survival time Y and a censoring

time C are independent. Then, the likelihood of the random samples, (X_i, δ_i) ($i = 1, 2, \dots, N$), can be represented as follows (Klein and Moeschberger, 2003):

$$L = \prod_{i=1}^N [\lambda(x_i)]^{\delta_i} S(x_i). \quad (1.1)$$

In order to estimate a survival function, $S(x)$, with a nonparametric framework we use the Kaplan–Meier estimator (Kaplan and Meier, 1958), which is given by

$$\hat{S}(x) = \begin{cases} 1 & (x < y_{(1)}) \\ \prod_{i \in A} \left(1 - \frac{d_i}{n_i}\right) & (y_{(1)} \leq x) \end{cases}, \quad (1.2)$$

where $y_{(1)}$ represents the earliest event occurrence time in the data, and $A = \{i : x_i \leq x\}$ is the set of observation labels. d_i represents the number of events at time x_i , and $n_i = \sum_{j=1}^N I(x_j \geq x_i)$ is the number of patients at risk at time x_i . This estimator can be interpreted as follows: The event times, $y_{(1)}, y_{(2)}, \dots$ are observed discretely; therefore, the survival function, $S(x)$, is estimated by a discrete distribution that has jumps at each event time. The probability of $\Pr(X > y_{(i)} | X \geq y_{(i)})$ can be estimated by $(n_i - d_i)/n_i$. Since $S(0) = 1$ and $S(y_{(i-1)}) = \Pr(X > y_{(i-1)}) = \Pr(X \geq y_{(i)})$ under a discrete distribution, $S(y_{(i)})$ can be estimated as follows (Klein and Meshberger, 2003):

$$\begin{aligned} S(y_{(i)}) &= \frac{S(y_{(i)})}{S(y_{(i-1)})} \frac{S(y_{(i-1)})}{S(y_{(i-2)})} \dots \frac{S(y_{(1)})}{S(0)} S(0) \\ &= \Pr(X > y_{(i)} | X \geq y_{(i)}) \Pr(X > y_{(i-1)} | X \geq y_{(i-1)}) \dots 1. \end{aligned}$$

This estimator represents the Kaplan–Meier estimator (1.2).

In Chapters 4 and 5, we try to construct survival time prediction models for several subgroups of patients. In order to do so, we use a binary tree-structured model that divides the patients into two groups by repeatedly using covariates. At each step in the model, we compare the survival functions of the two groups by using Mantel’s (1966) log-rank test as one way.

Now, we shall discuss the two groups, which we label group 1 and group 2. Suppose that there are r distinct event times, $y_{(1)} < y_{(2)} < \dots < y_{(r)}$, across the two groups. Let d_{1k} and d_{2k} be the number of events at $y_{(k)}$ for group 1 and group 2, respectively ($k = 1, 2, \dots, r$). In addition, let $R(y_{(k)}) = \{i; x_i \geq y_{(k)}, i = 1, 2, \dots, N\}$ be the set of patients who are alive and uncensored at a time just prior to $y_{(k)}$ (sometimes called the risk set). Furthermore, we suppose that there are n_{1k} patients in the risk set at $y_{(k)}$ for group 1 and n_{2k} in the risk set for group 2. Consequently, there are $d_k = d_{1k} + d_{2k}$ events and $n_k = n_{1k} + n_{2k}$ risks in total at time $y_{(k)}$. If we regard d_{1k} as a random variable which can take any value between 0 and $\min(d_k, n_{1k})$, then d_{1k} has a hypergeometric distribution with

parameters n_k , d_k , and n_{1k} under the null hypothesis that there is no difference in the survival of the two groups' patients. Therefore, the probability that the number of events in group 1 takes the value d_{1k} can be given by

$$f(d_{1k}) = \begin{cases} \frac{\binom{d_k}{d_{1k}} \binom{n_k - d_k}{n_{1k} - d_{1k}}}{\binom{n_k}{n_{1k}}} & (d_{1k} = 0, 1, \dots, \min(d_k, n_{1k})) \\ 0 & (\text{otherwise}) \end{cases}, \quad (1.3)$$

where $\binom{a}{b}$ represents a binomial coefficient. Based on (1.3), the expectation and variance of d_{1k} can be given as

$$e_{1k} = \frac{n_{1k}d_k}{n_k}, \quad v_{1k} = \frac{n_{1k}n_{2k}d_k(n_k - d_k)}{n_k^2(n_k - 1)}. \quad (1.4)$$

In order to combine the patients' information for each event time and obtain an overall measure of the deviation between the observed and expected values of d_{1k} , we use Mantel and Haenszel's (1959) test statistic

$$\frac{\sum_{k=1}^r (d_{1k} - e_{1k})}{\sqrt{\sum_{k=1}^r v_{1k}}}. \quad (1.5)$$

If the number of event times is not too small, then this statistic has an approximate standard normal distribution (Collett, 2003).

We can consider several approaches for identifying the significant prognostic factors necessary to construct a prediction model. If know the survival data's underlying distribution, then we can use parametric methods based on the proportional model, the accelerated failure time model, or other such models (Lawless, 2002; Collett, 2003; Klein and Moeschberger, 2003; Lee, 2003). However, in many situations we cannot know the exact form of the distribution. Consequently, Cox's (1972) proportional hazard model is commonly used in medical data analysis because it does not require a parametric survival distribution. This hazard model can be written as

$$\lambda(x|\mathbf{Z}) = \lambda_0(x) \exp(\boldsymbol{\beta}'\mathbf{Z}),$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)$ are the coefficients on the parameters. $\lambda_0(x)$ represents the baseline hazard function, which is dependent on time, and the exponential term is dependent upon the covariates. In this model, the form of $\lambda_0(x)$ is not specified, and therefore, $\lambda(x|\mathbf{Z})$ is called a semiparametric model.

To estimate the parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)$, Cox (1972) proposes maximizing a partial likelihood. For simplicity, we assume that there are no tie events in the data (for a discussion of cases that include tie events, see Klein and Moeshberger (2003)). Suppose that there are r distinct event times $y_{(1)} <$

$y_{(2)} < \dots < y_{(r)}$ with corresponding covariates $\mathbf{z}_{(1)}, \mathbf{z}_{(2)}, \dots, \mathbf{z}_{(r)}$, where $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{pi})$. We let $R(y_{(i)}) = \{j; x_j \geq y_{(i)}, j = 1, 2, \dots, N\}$ be the risk set of patients whose observed times are at least $y_{(i)}$. Then, the conditional probability that a patient experiences an event at $y_{(i)}$, with covariates $\mathbf{z}_{(i)}$, under the condition that the patient is in $R(y_{(i)})$, is given by

$$\frac{\exp(\boldsymbol{\beta}' \mathbf{z}_{(i)})}{\sum_{j \in R(y_{(i)})} \exp(\boldsymbol{\beta}' \mathbf{z}_j)}.$$

By multiplying these conditional probabilities across all events, we obtain the partial likelihood function

$$L(\boldsymbol{\beta}) = \prod_{i=1}^r \frac{\exp(\boldsymbol{\beta}' \mathbf{z}_{(i)})}{\sum_{j \in R(y_{(i)})} \exp(\boldsymbol{\beta}' \mathbf{z}_j)}.$$

Just as in the logistic regression, the parameters which maximize this partial likelihood can be obtained through iterative methods, like the Newton–Raphson approach (Klein and Moeschberger, 2003; Lee, 2003). In Chapter 4, we use the Cox proportional model to analyze data on brain cancer patients.

1.3 Statistical methods based on interval-valued variables

In classical analysis, the data on p random variables are represented as a single point in p -dimensional space. In Figure 1.1, we show an example of the classical representation of individuals with three random variables, Y_1 , Y_2 , and Y_3 . However, in some cases, it is difficult to represent data as a single point in variable space. For example, if we want to represent the length of a crow’s body based not on an individual but on the species, we cannot represent it as a single point because it is a range of 30 to 40 cm. Additionally, if we wanted to represent an individual’s blood pressure as measured over the span of one month, we would be unable to do so as blood pressure varies from day to day.

One possible method for handling such case is analysing data based on the mean value of the variables. However, this method ignores the variations within each data, and as such, a different approach is called for. Consequently, methods based on symbolic data have received a great deal of attention in recent years. Symbolic data on random variables are represented by interval-valued or multi-valued variables. In this thesis, we treat interval-valued symbolic data, which have p random variables, as p -dimensional hyperrectangles in \mathbb{R}^p . In the analysis of symbolic data, each unit of data (which is composed of a single individual or a number of individuals) is called a concept (this is terminology commonly used in symbolic data analysis). Figure 1.2 provides an example of the symbolic representation of concepts. Because analyses based on symbolic data account

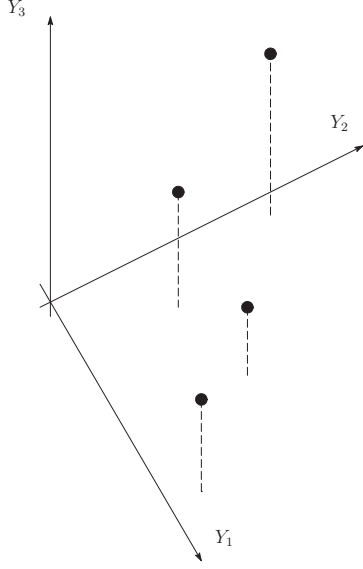


Figure 1.1: Example of the classical representation of individuals.

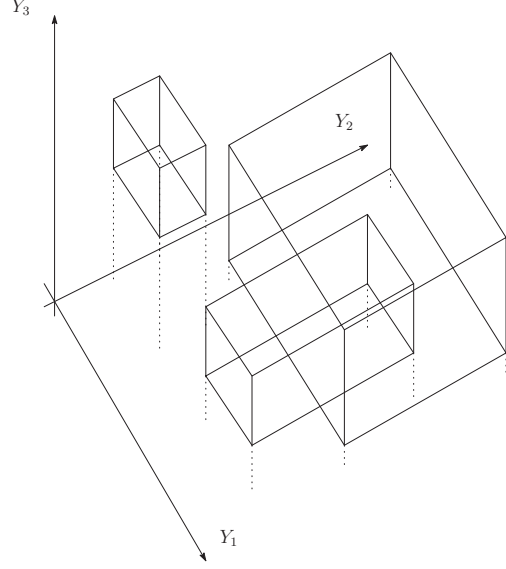


Figure 1.2: Example of the symbolic representation of concepts.

for the variation within the concepts, they are considered more suitable than those based merely on the mean value.

Let $E = \{\omega_1, \omega_2, \dots, \omega_m\}$ be a set of samples where ω_u ($u = 1, 2, \dots, m$) represents a concept of observation u . We assume that the concepts are described by p continuous quantitative variables, $\mathbf{Y} = (Y_1, Y_2, \dots, Y_p)$, with realizations $\mathbf{y}(\omega_u) = (y_1(\omega_u), y_2(\omega_u), \dots, y_p(\omega_u))$, where $y_j(\omega_u) = [a_{ju}, b_{ju}]$ ($j = 1, 2, \dots, p$). Each interval $[a_{ju}, b_{ju}]$ can be closed or open at either end.

We shall now provide some formal definitions for interval-valued symbolic-data analysis (Billard and Diday, 2006). Let the random variables, \mathbf{Y} , have domains of $\mathcal{X} = \times_{j=1}^p \mathcal{Y}_j$, where \mathcal{Y}_j represents the set of possible values of Y_j . Then, every point, $\mathbf{x} = (x_1, x_2, \dots, x_p)$, in \mathcal{X} is called a description vector. Furthermore, we let D_j be a subset of \mathcal{Y}_j ($D_j \subseteq \mathcal{Y}_j, j = 1, 2, \dots, p$). Then, the p -dimensional subspace $D = (D_1, D_2, \dots, D_p) \subseteq \mathcal{X}$ is called a description set.

Let $\mathbf{Y} = (\mathbf{Y}^1, \mathbf{Y}^2)$ where $\mathbf{Y}^1 = (Y_1^1, Y_2^1, \dots, Y_k^1)$ takes point values $\mathbf{x}^1 = (x_1^1, x_2^1, \dots, x_k^1) \in \mathcal{X}^1 = \times_{j=1}^k \mathcal{Y}_j^1$, and $\mathbf{Y}^2 = (Y_1^2, Y_2^2, \dots, Y_{p-k}^2)$ takes subset values $D^2 = (D_1^2, D_2^2, \dots, D_{p-k}^2) \subseteq \times_{j=1}^{p-k} \mathcal{Y}_j^2$. Then, $\mathbf{d} = (x_1^1, \dots, x_k^1, D_1^2, \dots, D_{p-k}^2)$ is called a description \mathbf{d} . Furthermore, the set of all possible descriptions is called the description space, \mathcal{D} . The symbolic description of a concept ω_u is given by the description vector $\mathbf{d}_u \in (D_1, D_2, \dots, D_p)$ in the space $\mathcal{D} = \times_{j=1}^p D_j$. If each D_j is a set of only one value, then the descriptions denoted by \mathbf{x} are called individual descriptions. That is, $\mathbf{x} = (x_1, x_2, \dots, x_p) \equiv \mathbf{d} = (\{x_1\}, \{x_2\}, \dots, \{x_p\})$, $\mathbf{x} \in \mathcal{X} = \times_{j=1}^p \mathcal{Y}_j$.

Bertrand and Goupil (2000) put forth the basic statistics for interval-valued

variables. For simplicity, we focus on a particular random variable, $Y \in \mathbf{Y}$, with realization $y(\omega_u) = [a_u, b_u]$, where $\omega_u \in E$. If we assume that the individual descriptions, x , are uniformly distributed over the interval $y(\omega_u)$, it follows that,

$$\Pr(x \leq \xi) = \begin{cases} 0 & (\xi < a_u) \\ \frac{\xi - a_u}{b_u - a_u} & (a_u \leq \xi < b_u) \\ 1 & (b_u \leq \xi) \end{cases}, \quad (1.6)$$

for each ξ . From (1.6), we obtain the empirical density function of the random variable Y

$$g(\xi) = \frac{1}{m} \sum_{u \in A_\xi} \frac{1}{b_u - a_u}, \quad (1.7)$$

where A_ξ is the set of labels for the concepts, $\{u : \xi \in y(\omega_u)\}$. Moreover, the observed frequency of concept ω_u over interval $I_h = [\xi_1, \xi_2]$ is defined as

$$f_h(u) = \frac{\|y(\omega_u) \cap I_h\|}{\|y(\omega_u)\|}, \quad (1.8)$$

where $\|\cdot\|$ represents the length of that interval, and the total observed frequency is defined as

$$f_h = \sum_{u=1}^m f_h(u).$$

Using the empirical density function (1.7), the symbolic sample mean and variance are given by

$$\bar{Y} = \frac{1}{m} \sum_{u=1}^m \frac{(b_u + a_u)}{2},$$

and

$$V_Y = \frac{1}{3m} \sum_{u=1}^m [(a_u - \bar{Y})^2 + (a_u - \bar{Y})(b_u - \bar{Y}) + (b_u - \bar{Y})^2],$$

respectively.

Billard (2007) states that the total sum of squares for Y (called the symbolic total sum of squares) can be written as

$$\text{TSS}(Y) = \text{WSS}(Y) + \text{BSS}(Y) = mV_Y, \quad (1.9)$$

where

$$\begin{aligned} \text{WSS}(Y) &= \frac{1}{3} \sum_{u=1}^m [(a_u - \bar{Y}_u)^2 + (a_u - \bar{Y}_u)(b_u - \bar{Y}_u) + (b_u - \bar{Y}_u)^2], \\ \text{BSS}(Y) &= \sum_{u=1}^m [\bar{Y}_u - \bar{Y}]^2, \end{aligned}$$

and $\bar{Y}_u = (a_u + b_u)/2$ is the mean of the random variable Y for concept ω_u . More detailed descriptions of symbolic data are given in Billard and Diday (2006), Bock and Diday (2000), and Diday and Noirhomme-Fraiture (2008).

Although methods involving interval-valued variables are not as well studied as classical variable methods, several regression models have been proposed. Billard and Diday (2000) present an approach for fitting a linear regression model to interval-valued symbolic data. First, they fit a linear regression model to the midpoint of the sample's interval values. Next, the estimated model is applied to the lower and upper bounds of the interval values in order to predict the lower and upper bounds of the response variables. De Carvalho et al. (2004) propose another approach that fits two linear regression models to the midpoint and the range of the interval values. De Carvalho et al. (2004) and Lima Neto et al. (2004) show that this method is superior to that of Billard and Diday with a series of Mont Carlo simulations. Lima Neto et al. (2005) improve upon Billard and Diday's and De Carvalho et al.'s methods by guaranteeing mathematical coherence between the upper and lower predicted response variables. In Chapter 6, we propose a new model based on interval-valued variables and use it to analyze data on HIV-1-infected patients.

1.4 Medical image processing

In recent years, medical images have come to play an increasingly crucial role in the early detection, diagnosis, and treatment of cancer patients. In response to this increased demand, a wide variety of medical imaging techniques have been developed, such as X-rays, computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound. Through the processing and classification of these images, computer-aided diagnosis systems are able to assist medical professionals in detecting and identifying structures such as tumors or lesions as well as in predicting patients' prognoses, among other things. Figures 1.3 and 1.4 provide examples of cancer patients' CT and MRI images, respectively.

A digital, monochrome image is obtained by sampling the corresponding continuous image function, $I(m, n)$, and storing the discretized values in the form of a two-dimensional image matrix, $f(m, n)$, with $m = 0, 1, \dots, N_m - 1$ and $n = 0, 1, \dots, N_n - 1$, where N_m and N_n are the sizes of the image. Every element, (m, n) , is called a pixel, and each pixel has a gray level, (y_1, y_2, \dots, y_B) , that is determined by the image matrix $f(m, n)$. y_B is given by quantization level b and set to y_{2^b} . Generally, the gray level takes an integer value between zero and the number of gray levels.

Medical image processing and classification involves multiple processes; it consists of a preprocessing step, a feature extraction step, a feature selection step, and a classification step. In the preprocessing step, unnecessary information is removed from the image and important information is emphasized by

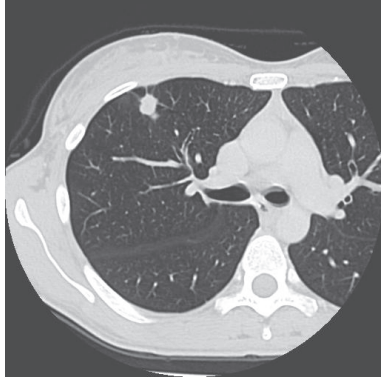


Figure 1.3: Example of a lung cancer patient's CT image.

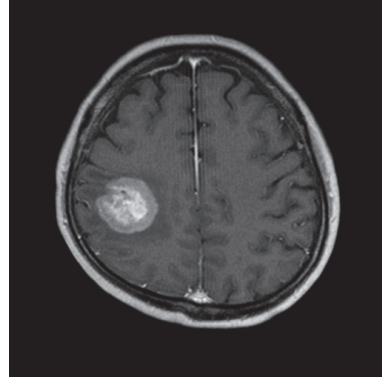


Figure 1.4: Example of a brain tumor patient's MRI.

means of denoising, deblurring, edge detection, etc. There are many methods by which to do this (Bankman, 2009), and we use several of them in Chapters 2 and 4 to perform our analysis. As shown in Figure 1.3, CT images of lung cancer patients display areas surrounding the lung, such as their fat and heart. In order to consider only the lung, we attempt to extract the region of interest (ROI) through binary image processing (Otsu, 1979) and edge detection (Marr and Hildreth, 1980). In Chapter 4, we use MRI images to conduct survival analyses on patients with brain metastases from breast cancer. In that analysis, we use the watershed transform technique (Horst and Heinz, 2000) to focus on the skull and cerebral fluid images.

In the subsequent step, features are extracted from the preprocessed images. A 512×512 image has 262,144 pixels, and as such, we cannot use this raw information for classification due to the curse of dimensionality. Therefore, we have to generate new features from the available image matrix, $f(m, n)$, and the generated features must contain all of the relevant information from the original image. These features are divided into four categories, nontransformed structural characteristics, transformed structural characteristics, structural descriptions, and graph descriptors (Ciaccio et al., 1993). In this thesis, we focus on texture features, which are obtained by transforming the preprocessed image matrix. Texture describes the spatial distribution of the pixels' gray levels, and it does not depend on the object's size, shape, orientation, or brightness. More exactly, texture statistically represents a quantitative measure of an images' arrangement of intensities (Mayer-Bäse, 2004). The examples of texture images are shown in Figure 1.5.

Gray-level co-occurrence matrix (GLCM) based statistics have been widely used to represent the global aspects of texture in images. A GLCM is a $y_B \times y_B$ matrix whose elements represent the relationship between two points' gray levels

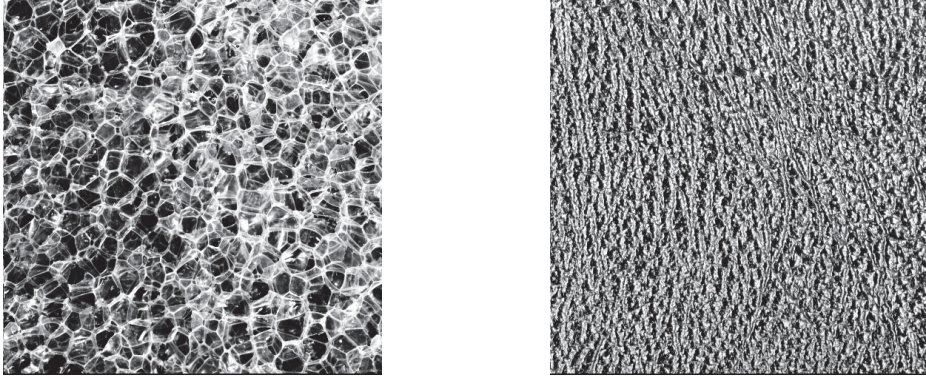


Figure 1.5: Examples of texture images.

for a fixed distance vector

$$\mathbf{d} \equiv (\Delta_m, \Delta_n),$$

where Δ_m and Δ_n are fixed values in the ranges $0, 1, \dots, N_m-1$ and $0, 1, \dots, N_n-1$, respectively. In other words, the GLCM represents the frequency of an arbitrary gray level combination over a fixed distance in the image matrix. Each element of the GLCM is defined as

$$G(q, r, \mathbf{d}) = \sum_{m=0}^{N_m-1} \sum_{n=0}^{N_n-1} I \{f(m, n) = y_q \cap f(m + \Delta_m, n + \Delta_n) = y_r\}, \quad (1.10)$$

where y_q and y_r indicate the arbitrary gray level combination and assumed values of y_1, y_2, \dots, y_B . Statistics (like the mean or the deviation) obtained from the GLCM are used to analyze textures in medical images (Haralick et al., 1973; Dhawan et al., 1996). In Chapters 2 and 4, we describe and utilize 10 texture features based on the GLCM.

Although the GLCM provides information on the texture of the entire image, it does not completely describe the texture, particularly in regards to the local aspects. In order to address this problem, we use Dhawan et al.'s (1996) wavelet-transformation-based method. Wavelet theory has been widely used in a number of signal processing applications, such as multiresolution signal processing, subband coding, and image and speech compression (Mallat, 1989). Laine and Fan (1993) state that by using wavelet decomposition, the texture can be represented in full detail. They successfully classify 25 natural textures without any error by using features acquired with Daubechies wavelets.

In the analysis of an image given in a spatial domain, the location of each pixel and the magnitude of each gray level are emphasized. In frequency domain analysis, on the other hand, it is possible to focus on the periodicity of each pixel in the image. We use the wavelet transformation for this analysis. Using

the one-staged two-dimensional wavelet transformation, the image matrix is decomposed into an approximation coefficient and three wavelet coefficients (Mallat, 1989; Addison, 2002)

$$\begin{aligned}
S_{1,(q_1,q_2)} &= \frac{1}{2} \sum_{k_1} \sum_{k_2} c_{k_1} c_{k_2} f(2q_1 + k_1, 2q_2 + k_2), \\
T_{1,(q_1,q_2)}^h &= \frac{1}{2} \sum_{k_1} \sum_{k_2} b_{k_1} c_{k_2} f(2q_1 + k_1, 2q_2 + k_2), \\
T_{1,(q_1,q_2)}^v &= \frac{1}{2} \sum_{k_1} \sum_{k_2} c_{k_1} b_{k_2} f(2q_1 + k_1, 2q_2 + k_2), \\
T_{1,(q_1,q_2)}^d &= \frac{1}{2} \sum_{k_1} \sum_{k_2} b_{k_1} b_{k_2} f(2q_1 + k_1, 2q_2 + k_2),
\end{aligned}$$

where $q_1 = 0, 1, \dots, \lfloor \frac{N_m-1}{2} \rfloor$ and $q_2 = 0, 1, \dots, \lfloor \frac{N_n-1}{2} \rfloor$ are dilation parameters, and where $\lfloor x \rfloor = \max\{n \in \mathbb{Z} | n \leq x\}$. The scaling coefficients, c_k , and wavelet coefficients, b_k , are given in advance for the various wavelets. We use Daubechies wavelets, D_6 and D_{20} , to analyze medical images (Daubechies, 1988). The coefficients c_k and b_k act as low- and high-pass filters for the signal, respectively. Therefore, $S_{1,(q_1,q_2)}$ can be obtained through horizontal and vertical low-pass filtering, $T_{1,(q_1,q_2)}^h$ through horizontal low-pass filtering and vertical high-pass filtering, $T_{1,(q_1,q_2)}^v$ through high-pass filtering and vertical low-pass filtering, and $T_{1,(q_1,q_2)}^d$ through horizontal and vertical high-pass filtering. Based on the approximation coefficient and three wavelet coefficients, we are able to calculate the energy and entropy by using Laine and Fan (1993) and Dhawan et al.'s (1996) method; we then use these values as features.

In the feature selection step, less discriminatory features are removed. Furthermore, as the number of features used in classification is reduced, faster and more accurate classification is possible. Exhaustive search, branch-and-bound, and sequential forward selection algorithms are also used (Mayer-Bäse, 2004). In Chapter 2, we use the GA in conjunction with the k -NN method to select an optimal combination of features.

Finally, in the classification step, the features obtained in the feature selection step are used to classify the images. In this step, supervised techniques (as discussed in Section 1.1) or unsupervised techniques, like clustering, can be used. In many supervised learning approaches, the feature selection and classification steps are performed simultaneously. For example, in the CART algorithm, a single feature is selected recursively to construct a splitting rule for dividing the data.

1.5 Organization of the thesis

This thesis is organized as follows: In Chapter 2, we apply the GA to the classification problem. We propose a new approach for feature weighting based on the GA, and then compare its performance to the existing methods by actually analyzing data. In the subsequent chapters, we study tree-structured prediction models for patients who have continuous or survival time responses. In Chapter 3, we present the basic method for constructing regression trees and survival trees. We present an example of survival tree construction in Chapter 4. In Chapter 5, we propose a method for constructing survival trees that breaks away from the traditional model and deals with the "exclusive OR" problem. We then evaluate this method's performance with simulation studies. A new approach for constructing regression trees based on interval-valued variables is described in Chapter 6. Finally, we present our conclusions in Chapter 7.

Because we focus on medical data analysis in this study, we, in fact, use several series of actual medical data. In Chapter 2, we examine CT images taken from lung cancer patients. In Chapter 4, we study MRI data on patients with breast cancer brain metastases. In Chapter 5, data on leukemia patients' bone marrow transplants is used to construct a tree-structured model. Furthermore, we refer to studies on HIV-1-infected patients' data for our proposed approach in Chapter 6.

Chapter 2

Application of Genetic Algorithm Based on Texture Features Obtained Through CT Images of Lung Cancer Patients

2.1 Introduction

In this chapter, we examine the availability of medical image classification techniques that involve the processing and pattern recognition for gray images by CT scan of lung cancer patients with different grades of the disease. As mentioned in previous chapter, medical image processing and classification involves multiple processes that can be classified into the following general steps: pre-processing step, feature extraction step, feature selection step, and classification step.

In preprocessing step, we attempt to remove the lung region from the whole image. First, binary image processing is used as a means to execute it. In binary image processing, pixels with a gray level higher than a threshold value y_s are set to 1 and the others are set to 0. The method used in our study for determining y_s is the Otsu method. This method uses the between-class variance as the criterion for determining y_s (Otsu (1979)). However, in this study, there are several images from which the region of interest (ROI) could not be removed by using only binary image processing. In such images, we attempted to remove the ROI by adding the edge-detected image to the binary image. An edge is a point such that the gray level of adjacent pixels changes rapidly. We used a Laplacian-of-Gaussian (LoG) filter and detected the zero-crossing point for edge detection. By convolving with the LoG filter, the noise is removed from the image and the sign of the edge is changed from plus to minus or minus to plus (Marr and Hildreth (1980)).

In feature extraction step, we focus on texture features. Based on gray-level

co-occurrence matrix (GLCM) and wavelet transformation of the image matrix $f(m, n)$, we extract 36 features for each patient. Three types of texture features are extracted in this research: statistical texture features extracted by using the GLCM of the entire image matrix, statistical texture features extracted by using the GLCM of the ROI in the image matrix, and texture features extracted by wavelet transform of the entire image matrix.

A genetic algorithm (GA) is used for feature selection. Siedlecki and Sklansky (1989) showed that the GA is a powerful tool for feature selection when the number of initial feature sets is large. If the initial number of features is p , the size of the entire search space is 2^p . This search space increases with p , and exploring the entire search space requires a lot of time, and occasionally, it is impossible. In the study of Siedlecki and Sklansky (1989), when the initial number of features exceeds 20, the feature selection problem is treated as a large-scale problem, and comparative studies of the exhaustive search method, branch-and-bound method, and GA method were performed. The study showed that the GA method is a powerful tool for large-scale feature selection.

In this study, for the purpose of image classification, we use k -nearest neighbor (k -NN) classification or classification by nearest neighbor of the center of each class. There are several reasons for using nearest neighbor methods. One is their simplicity, which makes them easy to implement, and another is the goodness of classification performance for a wide range of real world data sets (Raymer et al. (2000)). In the study of Punch et al. (1993), an GA-based approach that combines feature selection and data classification was proposed. They used a GA combined with a k -NN algorithm to optimize classification; the optimization was achieved by determining an optimal feature weighting. The weights were chosen to have a value from 256 values between 0 and 10, and each weight was represented by using 8 bits/feature. Further research was carried out by Raymer et al. (2000). By using a GA, they proposed the addition of a step to the feature selection step to select the number of optimal nearest neighbors, k .

Here, we propose an approach for searching the weights of features and examine its effectiveness. In this approach, the images given for learning are divided into two groups at random. The images in one group are used for training nearest neighbor classifier, while those in the other group are used for testing. By using the nearest neighbor method and a GA, an optimal feature set for classifying the test images is determined. Next, the images given for learning are again divided into two groups at random, and an optimal feature set is selected. By repeating this operation, the probability that each feature is chosen at the end of the feature selection is decided, and we use this probability directly as the weight in the feature space. The motivation to propose this iteration approach is that it is expected that the available images can be utilized more effectively, like ensemble methods. By performing cross-validation, we verified the classification accuracy of the proposed approach.

This chapter is organized as follows. The details of patients, and methods used for preprocessing of CT images are described in the next section. In Section 2.3, we describe the extraction methods of texture features from the images. In Section 2.4, we describe the methods of feature selection and classification of patients. Further, the feature weighting method proposed in this study are presented in the same section. The experimental results obtained by cross-validation are shown in Section 2.5, and the conclusion is presented in Section 2.6.

2.2 Patients and Preprocessing

2.2.1 Patients

In our analysis, we used a data set containing 39 CT images taken from different lung cancer patients. These included 19 images taken from patients with low-grade malignancy and 20 images taken from patients with high-grade malignancy. By performing medical images processing, we determined the percentage of images in which the malignant grade could be correctly identified. Each image had 512×512 pixels and the gray-level range was transformed into 8 bits (0–255). That is, $N_m = 512$, $N_n = 512$, $b = 8$, and $(y_1, y_2, \dots, y_B = y_{256}) = (0, 1, \dots, 255)$.

2.2.2 Extraction of the region of interest

In order to extract the ROI from the whole image, first, binary processing is performed by using only the Otsu method. By performing binary processing on the images, we aim to distinguish parts with high gray levels (like fat-containing parts and the heart) from those with low gray levels (like the lungs and objects outside the body). The ROI was extracted by labelling the lung portion of the binary-processed images.

The Otsu method determines the threshold for binary processing as follows (Otsu (1979)). Let m_1, m_2, \dots, m_B denote the number of pixels with gray levels y_1, y_2, \dots, y_B in the image matrix, respectively. Total number of pixels is given by $M = m_1 + m_2 + \dots + m_B$. Then, the probability distribution of each gray level is given by

$$p_i = \frac{m_i}{M} \quad \text{for } i = 1, 2, \dots, B.$$

Suppose that we dichotomize the pixels into two classes C_0 and C_1 by considering the threshold to correspond to gray level y_s . C_0 has pixels with gray levels y_1, y_2, \dots, y_s , and C_1 has pixels with gray levels $y_{s+1}, y_{s+2}, \dots, y_B$. Let σ_W^2 , σ_B^2 , and σ_T^2 denote the within-class variance, the between-class variance, and total

variance of gray levels, respectively:

$$\begin{aligned}\sigma_W^2 &= w_0\sigma_0^2 + w_1\sigma_1^2, \\ \sigma_B^2 &= w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2, \\ \sigma_T^2 &= \sum_{i=1}^B (y_i - \mu_T)^2 p_i,\end{aligned}$$

where $\mu_T = \sum_{i=1}^B y_i p_i$, $w_0 = \sum_{i=1}^s p_i$, $w_1 = 1 - w_0$, $\mu_0 = \sum_{i=1}^s y_i p_i / w_0$, $\mu_1 = \sum_{i=s+1}^B y_i p_i / w_1$, $\sigma_0^2 = \sum_{i=1}^s (y_i - \mu_0)^2 p_i / w_0$, $\sigma_1^2 = \sum_{i=s+1}^B (y_i - \mu_1)^2 p_i / w_1$. Then, the following basic relation always holds:

$$\sigma_W^2 + \sigma_B^2 = \sigma_T^2.$$

The within-class variance and the between-class variance depend on the threshold y_s , but the total variance does not depend on it. Therefore, there is a trade-off between the within-class variance and the between-class variance in terms of y_s . Moreover the within-class variance is based on the second-order statistics, while the between-class variance is based on the first-order statistics. Therefore, it is simpler to use the between-class variance for the criterion measure in order to evaluate the goodness of the threshold y_s than the within-class variance. Accordingly, Otsu used the following discriminant criterion measure:

$$\eta(y_s) = \frac{\sigma_B^2(s)}{\sigma_T^2}. \quad (2.1)$$

The optimal threshold y_{s^*} is the value that maximizes this equation (2.1), i.e., the between-class variance σ_B^2 . In order to find the optimal threshold, we calculate the following equation for all the gray levels in the image matrix:

$$\sigma_B^2(s) = \frac{[\mu_T \nu(s) - \mu(s)]^2}{\nu(s)(1 - \nu(s))},$$

where $\nu(s) = w_0$ and $\mu(s) = \sum_{i=1}^s y_i p_i$. The optimal threshold y_{s^*} satisfies the

$$\sigma_B^2(s^*) = \max_{1 \leq s \leq B} \sigma_B^2(s).$$

However, there are several cases in which it is difficult to extract the ROI solely through binary processing. An example is shown in Figure 1.3. One reason for the difficulty is that if a tumor is near the lung wall, then there is no significant difference between gray levels of fat and the tumor. To overcome this problem, we use the edge-detection process involving on LoG filter. An edge in the image is a place where the gray level changes rapidly. By combining a binary image and the edge-detected image, we try to extract the ROI.

An LoG filter performs smoothing with a Gaussian filter and quadratic differential processing with a Laplacian filter simultaneously when applied to an image. This filter is obtained as follows (Marr and Hildreth (1980)). As mentioned at the previous chapter, an image matrix $f(m, n)$ is obtained by sampling the corresponding continuous image function $I(m, n)$. The Laplacian of $I(m, n)$ is defined by

$$\nabla^2 I(m, n) = \frac{\partial^2}{\partial m^2} I(m, n) + \frac{\partial^2}{\partial n^2} I(m, n).$$

By taking the two-dimensional Taylor expansion of $I(m+1, n)$, $I(m-1, n)$, $I(m, n+1)$, and $I(m, n-1)$ around (m, n) and adding the four resulting equations, the following approximate expression is obtained:

$$\nabla^2 I(m, n) \approx I(m+1, n) + I(m-1, n) + I(m, n+1) + I(m, n-1) - 4I(m, n).$$

By expressing this approximate equation in a filter form for using an image matrix $f(m, n)$, the Laplacian filter is obtained as

$$\text{Laplacian filter} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The output image matrix obtained by the convolution of an input image matrix with the Laplacian filter is the approximation of the Laplacian of the input image matrix. Because the convolution involves quadratic differential processing of the image matrix, the sign of the output image changes from positive to negative (or negative to positive) at the edge, which is a position where the slope of a plane increases or decreases rapidly. Therefore, edge detection can be done by detecting the zero-crossing points of the output image. A zero crossing point is a position where the signs of two adjoining pixels are opposite to each other.

However only the Laplacian filter is used and when noise is present in the input image, the result will emphasize the noise. In order to overcome this problem, the input image is smoothed using a Gaussian filter before applying the Laplacian filter. The Gaussian filter is given by a two-dimensional Gaussian function $G(m, n)$ with mean 0 and variance σ^2 . The Laplacian of a smoothed image $F(m, n)$ can be obtained by calculating the convolution of the Laplacian of this two-dimensional Gaussian function with the continuous image $I(m, n)$:

$$\nabla^2 F(m, n) = \nabla^2 G(m, n) * I(m, n).$$

The Laplacian of the two-dimensional gaussian function, $\nabla^2 G(x, y)$, is given by

$$\nabla^2 G(m, n) = \frac{1}{2\pi\sigma^6} \left\{ (m^2 + n^2 - 2\sigma^2) \exp\left(-\frac{m^2 + n^2}{2\sigma^2}\right) \right\}, \quad (2.2)$$

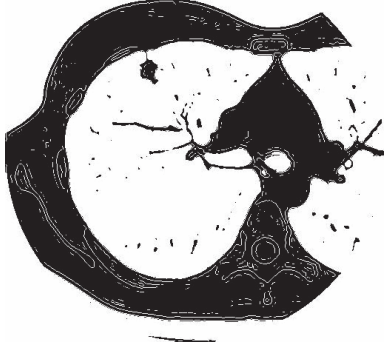


Figure 2.1: Combination of a binary image and the edge-detected image.

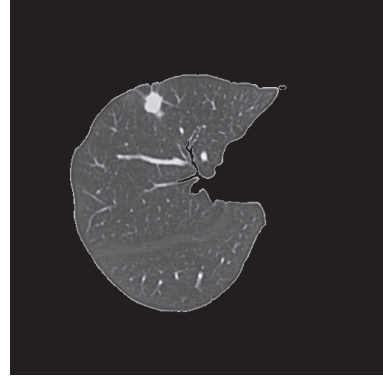


Figure 2.2: Extracted ROI the combination of the binary image and edge-detected image.

where σ^2 represents the degree of smoothing performed by the Gaussian function. If the value of σ^2 is large, then the output image $\nabla^2 F(m, n)$ represents the Laplacian of a blurred image that has been smoothed considerably. On the other hand, if the value of σ^2 is small, then output image represents the Laplacian of an input image that is almost identical to the output image. In this study, a filter of size 13×13 was constructed using the Laplacian of the two-dimensional gaussian function (2.2), and the edge was detected from the zero-crossing points of the output image, which obtained by convolving the filter and $f(m, n)$. The value of σ^2 was set to 2 in this study.

An image obtained by combining a binary image and the edge-detected image is shown in Figure 2.1. The ROI extracted by labelling the lung portion in Figure 2.1 is shown in Figure 2.2. It can be seen that the ROI has been extracted well in this case.

However, there are a few images in which the ROI could not be extracted by using this combination method. One reason for this is that the area of contact between the lung wall and tumor is very wide. That is, the lung boundaries could not be distinguished by using the edge-detection process. There are five such images out of the 39 images used in this study. For these five images, interpolation is carried out by using the cursor, and ROI images are obtained.

2.3 Feature extraction

2.3.1 Gray-level co-occurrence matrix

It is assumed that we obtain the GLCM for a patient's CT image by using (1.10). Let $H(q, r, \mathbf{d})$ be an element of the matrix which obtained by dividing the $G(q, r, \mathbf{d})$ by the sum of the all elements of the GLCM. In this study, we

extract 10 features by using $H(q, r, \mathbf{d})$ as a reference (Dhawan et al. (1996)). The definition of these features is as follows:

1. The contrast of $H(q, r, \mathbf{d})$ is defined as

$$\sum_{q=1}^B \sum_{r=1}^B \delta(q, r) H(q, r, \mathbf{d}),$$

where $\delta(q, r) = (y_q - y_r)^2$.

2. The correlation of $H(q, r, \mathbf{d})$ is defined as

$$\frac{\sum_{q=1}^B \sum_{r=1}^B y_q y_r H(q, r, \mathbf{d}) - \mu_{H_m(q, \mathbf{d})} \mu_{H_m(r, \mathbf{d})}}{\sigma_{H_m(q, \mathbf{d})} \sigma_{H_m(r, \mathbf{d})}},$$

where

$$H_m(q, \mathbf{d}) = \sum_{r=1}^B H(q, r, \mathbf{d})$$

and

$$H_m(r, \mathbf{d}) = \sum_{q=1}^B H(q, r, \mathbf{d})$$

are one-dimensional marginal distributions of $H(q, r, \mathbf{d})$. μ and σ represent the mean value and the standard deviation, respectively.

3. The energy of $H(q, r, \mathbf{d})$ is defined as

$$\sum_{q=1}^B \sum_{r=1}^B [H(q, r, \mathbf{d})]^2.$$

4. The homogeneity of $H(q, r, \mathbf{d})$ is defined as

$$\sum_{q=1}^B \sum_{r' \in A_q} \left[\frac{H(q, r', \mathbf{d})}{1 + \delta(q, r')} \right],$$

where $A_q = \{r; y_r \neq y_q, r = 1, 2, \dots, B\}$.

5. The entropy of $H(q, r, \mathbf{d})$ is defined as

$$-\sum_{q=1}^B \sum_{r=1}^B [H(q, r, \mathbf{d})] \log[H(q, r, \mathbf{d})].$$

6. The mean of $H_m(q, \mathbf{d})$ is defined as

$$\sum_{q=1}^B y_q H_m(q, \mathbf{d}).$$

7. The deviation of $H_m(q, \mathbf{d})$ is defined as

$$\sqrt{\sum_{q=1}^B \left[y_q - \sum_{p=1}^B y_p H_m(p, \mathbf{d}) \right]^2} H_m(q, \mathbf{d}).$$

The following three features are computed from the difference of $H(q, r, \mathbf{d})$ statistics. The difference of $H(q, r, \mathbf{d})$ represents the probability of occurrence of differences, $|y_q - y_r| = c_s$, in the gray level values of two pixels separated by a specific distance \mathbf{d} ($c_s = y_B - y_{B-s+1}$, $s = 1, 2, \dots, B$). It is defined as

$$H_{diff}(c_s, \mathbf{d}) = \sum_{q=1}^B \sum_{r' \in A_{c_s}} H(q, r', \mathbf{d}),$$

where $A_{c_s} = \{r; |y_q - y_r| = c_s, r = 1, 2, \dots, B\}$

8. The entropy of $H_{diff}(c_s, \mathbf{d})$ is defined as

$$-\sum_{s=1}^B H_{diff}(c_s, \mathbf{d}) \log H_{diff}(c_s, \mathbf{d}).$$

9. The energy of $H_{diff}(c_s, \mathbf{d})$ is defined as

$$\sum_{s=1}^B [H_{diff}(c_s, \mathbf{d})]^2.$$

10. The mean of $H_{diff}(c_s, \mathbf{d})$ is defined as

$$\sum_{s=1}^B c_s H_{diff}(c_s, \mathbf{d}).$$

To determine \mathbf{d} for calculating the GLCM, we perform the calculations shown below, following Dhawan et al. (1996). In one direction, we computed the correlation coefficients of the 10 texture features over all combinations of the 10 distances by taking 2 distances at a time ($\mathbf{d} = (1, -1), (2, -2), \dots, (10, -10)$). Thus, for each feature, a total of 45 different correlation coefficients were computed. Table 2.1 shows the averages and standard deviations of the correlation

Table 2.1: Averages and standard deviations of the correlation coefficients over all different combinations of the 10 distances.

Features	Ave. of Corr.	Std. of Corr.
1. Contrast	0.9123	0.0952
2. Correlation	0.9094	0.0986
3. Energy	0.9976	0.0022
4. Homogeneity	0.9881	0.0141
5. Entropy	0.9923	0.0107
6. Mean	0.9999	0.0001
7. Deviation	0.9977	0.0025
8. Entropy (diff)	0.9785	0.0243
9. Angular second moment	0.9860	0.0166
10. Mean	0.9583	0.0450

Ave.: average; Corr.: correlation; Std.: standard deviation.

coefficients over all different combinations of the 10 distances. Similar calculations were carried out for the other directions. For one distance, we computed the correlation coefficients of 10 features over all combinations of the four directions by taking two directions at a time ($\mathbf{d} = (5, 0), (0, 5), (5, 5), (5, -5)$). In Table 2.2, we show the results of this computation.

As seen from the results, for each feature, a high average value and a low standard deviation value of the correlation coefficients are observed in the case of both distance and direction. This observation implies that the GLCM is almost independent of the \mathbf{d} value used for its determination. Therefore, we decided to use only one distance and one direction ($\mathbf{d} = (5, -5)$). By using $\mathbf{d} = (5, -5)$, we calculated the 10 texture features described above.

2.3.2 Wavelet transformation

In the following, we describe the wavelet transformation discussed by Mallat (1989) and Addison (2002). For simplicity, we first describe the wavelet transformation for a one-dimensional signal vector $f(m)$, where $m = 0, 1, \dots, N_m - 1$, and then extend the description to a two-dimensional signal matrix (image matrix) $f(m, n)$.

The wavelet transformation of a signal $f(m)$ is its decomposition with a family of real orthonormal bases $\psi_{p,q}(m)$ obtained through the translation and dilation of a function known as the mother wavelet $\psi(m)$:

$$\psi_{p,q}(m) = \frac{1}{\sqrt{2^p}} \psi \left(\frac{m - q2^p}{2^p} \right), \quad (2.3)$$

where p and q are the translation and dilation parameters respectively. Assume

Table 2.2: Averages and standard deviations of the correlation coefficients over all different combinations of the four directions.

Features	Ave. of Corr.	Std. of Corr.
1. Contrast	0.9558	0.0160
2. Correlation	0.9526	0.0176
3. Energy	0.9983	0.0006
4. Homogeneity	0.9960	0.0017
5. Entropy	0.9979	0.0010
6. Mean	1.0000	0.0000
7. Deviation	0.9992	0.0006
8. Entropy (diff)	0.9890	0.0046
9. Angular second moment	0.9951	0.0023
10. Mean	0.9809	0.0072

See Table 2.1.

that the size of the signal vector N_m is sampled at a multiple of 2, $N_m = 2^M$, then the translation parameter is $p = 0, 1, \dots, M$ and the dilation parameter is $q = 0, 1, \dots, \lfloor \frac{N_m-1}{2^p} \rfloor$. The translation parameter represents the degree of decomposition by wavelet transformation. By convolution with the wavelet (2.3), the signal vector is decomposed, and the wavelet coefficients $T_{p,q}$ are obtained as

$$T_{p,q} = \sum_{m=0}^{N_m-1} f(m)\psi_{p,q}(m).$$

The wavelet transformation is associated with a scaling function that has the same form as that of the wavelet function. The scaling function $\phi_{p,q}(m)$ is obtained by using a function known as the father scaling function $\phi(m)$:

$$\phi_{p,q}(m) = \frac{1}{\sqrt{2^p}}\phi\left(\frac{m - q2^p}{2^p}\right). \quad (2.4)$$

The father scaling function $\phi(m)$ is the function, which satisfies the two-scale equation represented as

$$\phi(m) = \sum_{k=1}^{N_k} c_k\phi(2m - k),$$

where the coefficient c_k is called scaling coefficient. By the convolution of the signal with the scaling function (2.4), approximation coefficients $S_{p,q}$ are obtained:

$$S_{p,q} = \sum_{m=0}^{N_m-1} f(m)\phi_{p,q}(m).$$

The decomposition of the signal vector $f(m)$ to a certain level p_0 is described as

$$f(m) = \sum_q S_{p_0,q} \phi_{p_0,q}(m) + \sum_p^{p_0} \sum_q T_{p,q} \psi_{p,q}(m).$$

The mother wavelet can be constructed as follows:

$$\psi(m) = \sum_{k=1}^{N_k} b_k \phi(2m - k).$$

Coefficients c_k and b_k are related as

$$b_k = (-1)^k c_{N_k-1-k},$$

where N_k is the number of coefficients c_k . The coefficients c_k and N_k are given in advance for various wavelets. Using the scaling coefficients, the wavelet coefficients $T_{p+1,q}$ and the approximation coefficients $S_{p+1,q}$ are given by

$$T_{p+1,q} = \frac{1}{\sqrt{2}} \sum_k b_k S_{p,2q+k} \quad (2.5)$$

and

$$S_{p+1,q} = \frac{1}{\sqrt{2}} \sum_k c_k S_{p,2q+k}. \quad (2.6)$$

These relational expressions are known as the expressions of decomposition algorithm, and they indicate that if an approximation coefficient $S_{0,q}$ is given, then wavelet coefficients and approximation coefficients at each level can be obtained sequentially. In general, the sampled signal vector $f(m)$ is treated as the approximation coefficients $S_{p,q}$ at $p = 0$ and $q = 0, 1, \dots, N_m - 1$, and the signal vector is decomposed sequentially by the expressions (2.5) and (2.6).

The wavelet transformation of the two-dimensional signal matrix $f(m, n)$ is obtained by extending the one-dimensional decomposition. The two-dimensional decomposition algorithm can be written as

$$\begin{aligned} S_{p+1,(q_1,q_2)} &= \frac{1}{2} \sum_{k_1} \sum_{k_2} c_{k_1} c_{k_2} S_{p,(2q_1+k_1,2q_2+k_2)}, \\ T_{p+1,(q_1,q_2)}^h &= \frac{1}{2} \sum_{k_1} \sum_{k_2} b_{k_1} c_{k_2} S_{p,(2q_1+k_1,2q_2+k_2)}, \\ T_{p+1,(q_1,q_2)}^v &= \frac{1}{2} \sum_{k_1} \sum_{k_2} c_{k_1} b_{k_2} S_{p,(2q_1+k_1,2q_2+k_2)}, \\ T_{p+1,(q_1,q_2)}^d &= \frac{1}{2} \sum_{k_1} \sum_{k_2} b_{k_1} b_{k_2} S_{p,(2q_1+k_1,2q_2+k_2)}, \end{aligned}$$

where k_1 and k_2 are indices of scaling coefficients and q_1 and q_2 are dilation parameters.

In this study, we have obtained one signal approximation coefficient and three wavelet coefficients from the image matrix by two-dimensional wavelet transformation (that is $p = 1$). From each of these coefficients we obtained the energy and entropy by using the method of Laine and Fan (1993) and Dhawan et al. (1996):

1. The energy of coefficient is defined as

$$\frac{\sum_{q_1} \sum_{q_2} x_{p,q_1 q_2}^2}{\text{length} * \text{breadth}},$$

where $x_{p,q_1 q_2}$ is the computed signal approximation coefficient or signal wavelet coefficient value of the q_1 th row and q_2 th column and length and breadth are the dimensions of each coefficient.

2. The entropy of coefficient is defined as

$$-\sum_{q_1} \sum_{q_2} \left[\frac{x_{p,q_1 q_2}^2}{\text{norm}} \right] \log_{10} \left[\frac{x_{p,q_1 q_2}^2}{\text{norm}} \right],$$

where $\text{norm} = \sum_{q_1} \sum_{q_2} x_{p,q_1 q_2}^2$.

We use Daubechies wavelets D_6 and D_{20} , following Laine and Fan (1993). As described in the previous chapter, these authors successfully classified 25 natural textures without any error by using these wavelets. For each of these wavelets, the signal approximation coefficients and horizontal, vertical, and diagonal signal wavelet coefficients of the decomposition level $p = 1$ are calculated. From each of these coefficients, we obtained the energy and entropy, and we used a total of 16 features for classification.

A total of 36 features are used in this study, and they were standardized to the range $[0, 1]$. Then, the observed learning set is defined as $\mathcal{L} = \{(g_i, \mathbf{z}_i); i = 1, 2, \dots, N = 39\}$, where $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{36i})$. and g_i is 1 if the patient i has low-grade malignancy and is 2 if the patient has high-grade malignancy.

2.4 Feature selection and classification

2.4.1 Genetic algorithm

The first step in the implementation of the GA is to generate an initial population of chromosomes. Each chromosome will be represented as a binary string of length L , which corresponds to the problem encoded. In most cases, the initial population is generated randomly, and each chromosome is then evaluated.

The execution of the GA can be expressed as a two-stage process. In the first stage, it starts with the current population, and selection is applied to create an intermediate population. The selection is carried out on the basis of the evaluation of each chromosome a_u . In the second stage, crossover and mutation are applied to the intermediate population to create the next population. The process of proceeding from the current population to the next population is one generation in the GA. In the first generation, the current population is the initial population, which is generated randomly.

Crossover is applied by randomly selecting a pair of chromosomes from the intermediate population. In the case of crossover with a probability P_c , these chromosomes recombine to form two new chromosomes. The position of recombination (crossover point) is randomly chosen. For example, assume that the two chromosomes 110110011 and 010110100 are chosen from the intermediate population and that the crossover point is chosen to be between the fifth and sixth points in the chromosomes. Then, crossover is performed as follows:

$$\begin{array}{cc} 11011 & \times & 0011 \\ 01011 & \times & 0100 \end{array} .$$

Swapping the chromosomes between the two parents produces the offsprings 110110100 and 010110011. Generally, the probability of crossover, P_c , is high near bits with a value of 1. After the crossover operation, we apply a mutation operator. The mutation operation flips each bit in the population with some low probability P_m . For example, if the third and fifth points in the chromosome 110110100 are chosen, then mutation is carried out as

$$110110100 \longrightarrow 111100100.$$

Typically, the mutation rate is applied with a probability less than 0.01.

After the process of selection, crossover, and mutation is complete, the next population will be evaluated. The process of evaluation, selection, recombination, and mutation corresponds to one generation in the execution of a GA. An algorithmic description of the GA used in this study is given below:

Algorithm2.1.

1. Generate the initial population randomly for the chromosomes a_u .
2. The current population: $\Pi = \{a_u\}, u = 1, 2, \dots, U$.
3. **For** $v \leftarrow 1$ **to** the number of generations V **do**
4. Initialize the intermediate population I and the offspring population O .
5. **For** $u \leftarrow 1$ **to** the number of chromosomes U **do**
6. Evaluate the chromosome a_u in the population Π .
7. Copy to the intermediate population I from Π on the basis of the evaluation of chromosomes
 $(I = \{a'_u\}, u = 1, 2, \dots, U)$.

8. **For** $k \leftarrow 1$ **to** $n/2$ **do**
9. Choose the two parents a'_q and a'_r at random from I , and apply performed with the probability P_c
 $(a''_q, a''_r = (a'_q, a'_r) \cup \text{crossover}(a'_q, a'_r))$.
10. Add a''_q and a''_r to the offspring population O .
11. The offspring population: $O = \{a''_u\}, u = 1, 2, \dots, U$.
12. **For** $u \leftarrow 1$ **to** the number of chromosomes U **do**
13. **For** $w \leftarrow 1$ **to** the number of bits W **do**
14. Apply with mutation probability P_m to the w th bit from the chromosome a''_u in O .
15. Replace population Π by the offspring population O .
16. Evaluate each chromosome in population Π of the last generation and get the best chromosome.

In the GA, U is 100, V is 200, P_c is 0.8, and P_m is 0.01.

In step 7 of Algorithm 2.1, we use the remainder stochastic selection method. Let f_u is the evaluation of the chromosome a_u . Then the value of f_u/\bar{f} , where \bar{f} is the average evaluation of all chromosomes in the population of the generation v , is calculated in this method. For each chromosome a_u , the integer portion of the value of f_u/\bar{f} indicates how many copies of this chromosome are directly placed in the intermediate population. In addition to this, with a probability corresponding to the fractional portion of f_u/\bar{f} , an additional copy is made in the intermediate population. For example, a chromosome with $f_u/\bar{f} = 1.56$ places 1 copy in the intermediate population and has a 0.56 chance of placing another one chromosome in the intermediate population. This operation is repeated until the number of chromosomes in the intermediate population is U .

2.4.2 Nearest neighbor method

The k -nn begins by putting the learning data points in the feature space represented by a chromosome a_u . Let N denote the number of learning images, and let $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{pi})$ represents the coordinate of image i in the feature space ($i = 1, 2, \dots, N$). Furthermore, the chromosome a_u for each generation v ($v = 1, 2, \dots, V$) is represented as $\{f_{u1}, f_{u2}, \dots, f_{up}\}$, where f_{uj} is defined as

$$f_{uj} = \begin{cases} 1 & \text{when feature } Z_j \text{ is included} \\ 0 & \text{when feature } Z_j \text{ is not included} \end{cases} ,$$

($j = 1, 2, \dots, p$). Then, the classification rate for the chromosome a_u by k -nn is defined as

$$CR_u = \frac{1}{N} \sum_{i=1}^N I_{ui},$$

where I_{ui} represents whether or not the classification has been successful at patient i , and it is given as follows:

$$I_{ui} = \begin{cases} 1 & \sum_{a=1}^k \alpha_{uia} > \frac{k}{2} \\ 0 & \sum_{a=1}^k \alpha_{uia} < \frac{k}{2} \end{cases},$$

where k is an odd number fixed in advance. α_{uia} is defined to be 1 when in the feature space represented by the chromosome a_u , the a th neighbor data point m of i is in the same class as the i ; otherwise, is defined to be 0. The a th neighbor data point m is the point where the value of

$$\sqrt{\sum_{j=1}^p (f_{uj} \cdot z_{ji} - f_{uj} \cdot z_{jm})^2}$$

is the a th smaller for $m = 1, 2, \dots, N$ ($m \neq i$). We use $k = 1$ and 3 in this study.

Apart from this, the nearest neighbor method that involves the use of the center of each class is applied. In this method, we begin by the finding the center of each class for the learning images other than image i . That is to say, for a certain class c , each element z_{jc} of the center $z_c = (z_{1c}, z_{2c}, \dots, z_{pc})$ in the feature space represented by the chromosome a_u is defined as

$$z_{jc} = \frac{1}{N_c} \sum_{i=1}^{N_c} f_{uj} z_{ji},$$

where $i = 1, 2, \dots, N_c$ represents the patient of the member of c ($c = 1, 2$). Then, patient i is classified in the same class as the nearest neighbor center point in the feature space at the Euclidean distance.

2.4.3 Evaluation function

We use the evaluation function that was used by Siedlecki et al. (1989), to obtain the evaluation value for each chromosome in step 7 of Algorithm 2.1. This evaluation function uses a penalty function that increases exponentially when a chromosome has a error rate higher than a fixed threshold. This penalty function is defined as

$$p(e_u) = \frac{\exp((e_u - \theta_t)/\theta_m) - 1}{\exp(1) - 1},$$

where $e_u = 1 - CR_u$ is the error rate of a chromosome a_u , θ_t is the threshold, and θ_m is the scale factor. We note that $p(\theta_t) = 0$ and $p(\theta_t + \theta_m) = 1$ and that for higher values of the error rate, the penalty value quickly increases toward

infinity. By adding the number of features used in a_u to the penalty value $p(e_u)$, the score $J(a_u)$ can be obtained:

$$J(a_u) = p(e_u) + \sum_{j=1}^p f_{uj}.$$

Using this score, the evaluation function is defined as

$$f_u = (1 + \epsilon) \max_{a_u \in \Pi} J(a_u) - J(a_u),$$

where ϵ is a small positive constant. This ϵ assures that $\min(f_u) > 0$. In other words, even the chromosome with the least value of the evaluation function gets a chance to survive in the intermediate population. We set the parameters as $\theta_t = 0.01$, $\theta_m = 0.1$, and $\epsilon = 0.05$.

2.4.4 Weighting methods

In feature selection, we focus on deciding whether to include each feature, and the classification is carried out in the selected feature space. However, a higher classification rate is expected by suitable weighting each feature and classifying in the feature space.

In Punch et al. (1993), weights were chosen to have a value from among 256 values between 0 and 10, using 8 bits/feature to represent each weight. The string length in each chromosome was represented as $8 \times p$, and an optimal chromosome was searched for by using a GA. That is, the elements of the chromosome a_u were represented as $\{f_{u11}, f_{u12}, \dots, f_{u18}, f_{u21}, f_{u22}, \dots, f_{u28}, \dots, f_{up1}, f_{up2}, \dots, f_{up8}\}$, where each element f_{ujc} was 0 or 1. For feature Z_j , the weight was defined as

$$10 \times \frac{y_{uj}}{256},$$

where y_{uj} is the value obtained by translation from the binary representation $\{f_{uj1}, f_{uj2}, \dots, f_{uj8}\}$ corresponding to feature Z_j in the chromosome a_u to decimal representation. Using this weight, the a th neighbor data point is the point where the value of

$$\sqrt{\sum_{j=1}^p \left(10 \times \frac{y_{uj}}{256} \cdot z_{ji} - 10 \times \frac{y_{uj}}{256} \cdot z_{jm} \right)^2}$$

is smaller in the a th for $m = 1, 2, \dots, N$ ($m \neq i$). They evaluated the each chromosome using k -nn and searched an optimal weighting by GA.

The proposed weighting method begins by randomly dividing the learning images into two sets: for training and testing. Let $i_1 = 1, 2, \dots, N_1$ and $i_2 =$

$1, 2, \dots, N_2$ denote the training images and test images, respectively ($N_1 + N_2 = N$). Then, an optimal combination of features for classifying the test images by k -nn, which involves the use of the training images, is obtained by applying the GA. As discussed in Section 2.4.2, the classification rate for the chromosome a_u when k -nn is used is defined as

$$CR_u = \frac{1}{N_2} \sum_{i_2=1}^{N_2} I_{ui_2},$$

where I_{ui_2} is given by

$$I_{ui_2} = \begin{cases} 1 & \sum_{a=1}^k \alpha_{ui_2 a} > \frac{k}{2} \\ 0 & \sum_{a=1}^k \alpha_{ui_2 a} < \frac{k}{2} \end{cases}.$$

$\alpha_{ui_2 a}$ is defined to be 1 when in the feature space represented by the chromosome a_u , the a th neighbor training data point m of the testing data point i_2 is in the same class as i_2 ; otherwise, it is defined to be 0. The a th neighbor training data point is the point where the value of

$$\sqrt{\sum_{j=1}^p (f_{uj} \cdot z_{ji_2} - f_{uj} \cdot z_{ji_1})^2}$$

is the a th smaller for $i_1 = 1, 2, \dots, N_1$.

After an optimal combination of features is selected, again by the iterative process of randomly dividing the images into two groups and obtaining an optimal combination, the proportion of features selected from each group at the end by the GA is calculated. That is, if $\{f_{g1}, f_{g2}, \dots, f_{gp}\}$ denote the elements of the chromosome corresponding to the maximum evaluation in the last generation $v = V$ at iteration g ($g = 1, 2, \dots, G$), the proportion of each feature that are selected at the end by the GA is defined as

$$\left\{ \frac{1}{G} \sum_{g=1}^G f_{g1}, \frac{1}{G} \sum_{g=1}^G f_{g2}, \dots, \frac{1}{G} \sum_{g=1}^G f_{gp} \right\}.$$

We directly used this proportion as the weight of each feature. The aim of using this iteration method is that a more robust and more reliable conclusion could be arrived at owing to the randomness of the method. An algorithm of proposed method is described below:

Algorithm2.2.

1. **For** $g \leftarrow 1$ **to** the number of iterations G **do**
2. The N_1 training images are selected from N learning images at random ($i_1 = 1, 2, \dots, N_1$).

3. The remaining learning images are treated as test images ($i_2 = 1, 2, \dots, N_2$).
4. p features are extracted from each image matrix during training and testing.
5. An optimal feature set $\{f_{g1}, f_{g2}, \dots, f_{gp}\}$ for classifying the test images is selected by using the GA and k -nn.
6. The mean probability of each feature selected at the end by the GA is calculated $\left(\left\{\frac{1}{G} \sum_{g=1}^G f_{g1}, \frac{1}{G} \sum_{g=1}^G f_{g2}, \dots, \frac{1}{G} \sum_{g=1}^G f_{gp}\right\}\right)$.

In this method, G is 500, the number of generations in the GA is 150, and the values of the other parameters like the probability of crossover and mutation are the same as those specified in Section 2.4.1.

2.5 Results

The classification accuracy of the malignancy is examined for the no-feature selection method (that is, when all features are used for classification), the method using only feature selection by GA, the with-weighting method used by Punch et al., and the with-weighting method proposed in this study. The accuracy is calculated by using the leave-one-out cross-validation method. Namely, the available 39 images are divided into 38 learning images and 1 verifying image. Then, an optimal feature set (or an optimal feature weight) for classification is searched for through feature extraction and feature selection by using the learning images. After that, the verifying image is classified using the resulting optimal feature set. This operation is performed on all 39 images, and the classification accuracy is calculated.

In Table 2.3, we show the results of the cross-validation when all features are used (that is 36 features). The results show the correctness of CT image classification for malignant grade performed through feature extraction and selection in this study. The abbreviations used in Table 2.3 are as follows: NFS refers to the no-feature selection method, OFS refers to the method using only feature selection by a GA, WWPU refers to the with-weighting method used by Punch et al., and WWPR refers to the with-weighting method proposed in this study. In the $k = 1$ nearest neighbor method, the with-weighting methods have a higher classification rate than no-weighting methods. Additionally, there is no difference between the method used by Punch et al. and the method proposed in this study with respect to the classification rate. In the $k = 3$ nearest neighbor method, on the other hand, the methods of no-weighting showed slightly better results compared to the methods of with-weighting. In the nearest neighbor method, which uses the center of each class, the no-feature selection method shows a lower classification rate, about 72%. In contrast, the method proposed in this study shows a higher classification rate, about 90%, and this is the highest classification rate in this study.

Table 2.3: Results of cross-validation for about 36 features.

	$k = 1$	$k = 3$	Center
NFS	82.05%	89.74%	71.79%
OFS	84.62%	89.74%	87.18%
WWPU	87.18%	87.18%	87.18%
WWPR	87.18%	87.18%	89.74%

NFS: the no-feature selection method; OFS: the method using only feature selection by a GA; WWPU: the with-weighting method used by Punch et al.; WWPR: the with-weighting method proposed in this study.

Furthermore, we focus on the number of features that are finally selected in the feature selection process by the GA. The average number of features that are finally selected for $k = 1$, $k = 3$, and the center of nearest neighbor method were 1.74, 1.35, and 1.13 respectively. From this results, it is clear that most of the 36 features extracted from each image are rarely selected by GA as a suitable. Moreover, the percentage that the correlation of $H(q, r, \mathbf{d})$ in the ROI is selected at the end of the feature selection process is 100% in the case of each of the nearest neighbor methods. Thus, in the case of this classification, the correlation of $H(q, r, \mathbf{d})$ in the ROI is a very good feature. On the other hand, considering a case where a such a good feature could not be obtained by the feature extraction step, we performed the cross-validation by using the 35 features left by ignoring the correlation of $H(q, r, \mathbf{d})$ in the ROI by $k = 1$ and the center of nearest neighbor method. The results are shown in Table 2.4. Predictably, for all methods, the classification rate decreased compared to the case where all features are considered. In the $k = 1$ nearest neighbor method, the classification rate of the no-feature selection is reduced by about 13%. On the other hand, for the with-weighting methods, the decrease is only about 5%. That is, in the case of the $k = 1$ nearest neighbor method, the with-weighting methods are more effective and more robust for the classification. In the case of the center of nearest neighbor method, however, the percentage of reduction is higher for the with-weighting methods. Therefore with regard to this nearest neighbor method, the feature weighting show be performed more carefully. We think that the reason for the reduction in the classification rate is that the parameter setting of the GA for searching for the optimal weighting is not suitable. So we must continue to looking for an optimal method for appropriate parameter setting.

2.6 Conclusion

In this chapter, we verified the correctness of CT image classification for malignant grade tumors through feature extraction and selection. In the feature

Table 2.4: Results of cross-validation for about 35 features (except the correlation of $H(q, r, \mathbf{d})$ in the ROI).

	$k = 1$	Center
NFS	69.23%	71.79%
OFS	74.36%	82.05%
WWPU	82.05%	79.49%
WWPR	82.05%	76.92%

See Table 2.3.

extraction step, we focused on the texture features and extracted 36 features from each image by using a GLCM, wavelet transformation, binary processing, and edge detection. In particular, in those features, it was found that the correlation of the elements of the GLCM in the ROI is very effective for the classification. In the feature selection step, we used a genetic algorithm to search for a feature set or a weighting that was effective for the classification of the images. In this process, we proposed a new method for feature weighting using the iteration of GA. We performed a comparison of our method with the method of no weighting and the method of weighting using GA, which is proposed in another paper.

The correctness of CT image classification for malignant grade tumors in this study indicate that the classification rate can be expected to reach nearly 90% when the optimal method is used. For the proposed method, some cases showed good results. However, there was also a case where the performance of our method was worse than that of the method of no weighting. In future studies, there is a need to examine whether the proposed method is good for all situations. We believe that the iteration method involving a GA has the potential to be improved for achieving better feature weighting.

Chapter 3

Tree-structured Prediction Model

3.1 Introduction

In this chapter, we present the basic method for constructing tree-structured prediction models with the classification and regression tree (CART) algorithm (Breiman et al., 1984). In Chapters 4 and 5, we demonstrate the analysis of survival data with a tree-structured model (survival tree) and its expansion. In Chapter 6, a new approach for constructing a regression tree based on interval-valued variables is proposed. As mentioned in Chapter 1, the goal of tree-structured prediction models is to split the data into groups with differing outcomes by using covariates. An example of the tree-structured model is shown in Figure 3.1.

The basic description of the tree-structured model is as follows: Let T denote a tree model, which includes the set of splits and the order in which they are used, and subsets of covariate space. The p -dimensional covariate space R^p contains all possible covariate vectors. Each subset of the covariate space in the model, T , is called a "node" and we denote it as t , where $t \subseteq R^p$. Let \tilde{T} be a set of "terminal nodes" in T that represent the nodes at the bottom layer of the tree. The other nodes (those not at the bottom layer) are called "internal nodes". We define the set of internal nodes as U . The circles and squares in the figure represent the internal and terminal nodes, respectively. The size of the tree is defined by the number of terminal nodes, $|\tilde{T}|$ (the size of the tree in the figure is 7). The node at the top layer of the tree is called the "root node" (t_1 in the figure). The two nodes obtained by splitting an internal node are called "child nodes"; for example, the child nodes of t_6 are t_8 and t_9 . A tree T_t is called a "branch" of T (with root node $t \notin \tilde{T}$), if it contains the root node t and all descendants of t in T ; for example, the branch T_{t_3} in the figure consists of the root node t_3 and the terminal nodes $t_9 - t_{13}$. Moreover, a tree, like T_l , is called a "subtree" of T , if it is a subset of the tree T and has the same root

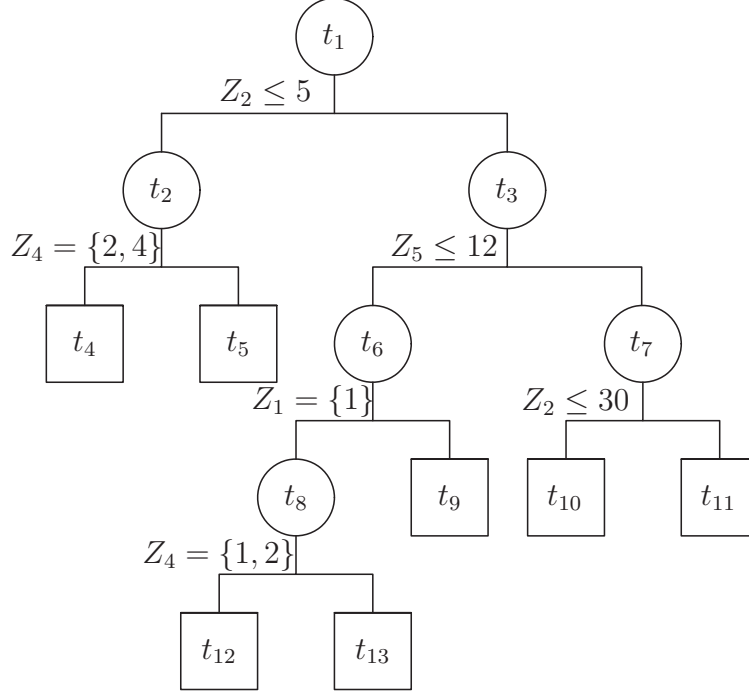


Figure 3.1: Example of a tree-structured model.

node.

Define a binary split s_t at node t , which divides t into child nodes t_L and t_R . The s_t is generated by the splitting rule, which is the form of the question " $\mathbf{Z} \in A$ ", where $A \subset \mathbb{R}^p$. Then, the split s_t sends all \mathbf{z}_i in node t that answer "yes" to t_L , and all \mathbf{z}_i in node t that answer "no" to t_R . The splitting rule generally consists of a single covariate, Z_j . If Z_j is a continuous variable, then the rule takes the form " $Z_j \leq s$ ", where s is the threshold value for splitting (see, for example, the splits of the nodes t_1 , t_3 , and t_7 in the figure). If Z_j is a categorical variable with possible values $F = \{f_{j1}, f_{j2}, \dots, f_{jC_j}\}$ (where f_j is a category value) then the rule is " $Z_j \in R_{tj}$ " where $R_{tj} \subset F$. t_2 , t_6 , and t_8 in Figure 3.1 are examples of this kind of split.

Under a non-parametric framework, the CART algorithm for constructing a tree-structured model with an observed learning set, \mathcal{L} , consists of three steps, splitting, pruning, and selection. First, all of the learning samples are divided into two groups according to a covariate, Z_j . The method for selecting the covariate and the threshold, s (or combination of categories, R_{tj}), is described later in this chapter. By dividing the data at each terminal node, a maximum size tree, T_0 , is constructed. Although T_0 is the optimal tree-structured model for the learning sample, \mathcal{L} , it may not always yield the best predictions (i.e., there may be overfitting). Therefore, we must prune T_0 and select the optimal tree size in order to improve the model.

In the pruning step, a set of subtrees, T_1, T_2, \dots, T_M , can be constructed by using Breiman et al.'s (1984) cost-complexity measure or Leblanc and Crowley's (1993) split-complexity measure. The cost-complexity measure combines the sum of the data's heterogeneity in each terminal node with the size of the tree. The split-complexity measure, on the other hand, is based on the degree of separation between the nodes and the size of the tree. In the selection step, the optimal subtree is chosen from among those created in the pruning process. If a large number of samples were included in \mathcal{L} , we could use the test sample method for this step. However, in many cases, we are not able to use such a large data set, and then we must employ either the V -fold cross-validation method or the bootstrap method (Breiman et al., 1984; Leblanc and Crowley, 1993).

This rest of this chapter is organized as follows: In Section 3.2, we describe the construction of the regression tree. In Section 3.3, we describe the construction of the survival tree. The conclusion is presented in Section 3.4.

3.2 Regression Tree

We denote the observed learning samples as $\mathcal{L} = \{(x_i, \mathbf{z}_i); i = 1, 2, \dots, N\}$, where x_i denotes a continuous response and $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{pi})$ denotes a p -dimensional covariate vector for the i th patient. Using this \mathcal{L} , the goal is to construct a tree-structured model, T , that minimizes the mean squared error, $R^*(T)$, about a new datum

$$R^*(T) = E[(X - T(\mathbf{z}))^2], \quad (3.1)$$

where $T(\mathbf{z})$ is the predicted value of the continuous response variable, X . In order to do this, we follow three steps to construct a tree-structured model.

3.2.1 Splitting

In the splitting step, a sequence of binary splits are used to divide the covariate space, R^p , into terminal nodes. In each node $t \in T$, the predicted value $\hat{x}(t)$ is given by a constant. Let $\mathcal{L}_t = \{(x_i, \mathbf{z}_i); i = 1, \dots, N_t\}$ be the learning samples in node t , where N_t is the number of patients included in \mathcal{L}_t . To establish a rule for assigning a value, $\hat{x}(t)$, to each node, we consider the resubstitution estimate for $R^*(T)$ from (3.1)

$$\begin{aligned} R(T) &= \frac{1}{N} \sum_{i=1}^N (x_i - T(\mathbf{z}_i))^2 \\ &= \frac{1}{N} \sum_{t \in \tilde{T}} \sum_{i \in A_{\mathcal{L}_t}} (x_i - \hat{x}(t))^2, \end{aligned} \quad (3.2)$$

where $A_{\mathcal{L}_t}$ is the set of labels, $\{i : \mathbf{z}_i \in t\}$, for the observations in node t . Then, the predicted value, $\hat{x}(t)$, that minimizes (3.2) can be given by the mean of the observations in node t :

$$\bar{x}_t = \frac{1}{N_t} \sum_{i \in A_{\mathcal{L}_t}} x_i. \quad (3.3)$$

By substituting \bar{x}_t into (3.2), $R(T)$ can be given by

$$R(T) = \sum_{t \in \bar{T}} R(t), \quad (3.4)$$

where $R(t)$ is the mean of the residual sum of squares in t .

$$R(t) = \frac{1}{N} \sum_{i \in A_{\mathcal{L}_t}} (x_i - \bar{x}_t)^2. \quad (3.5)$$

Let S_t be the set of all possible splits s_t for node t into its two child nodes t_L and t_R . As such, each element of S_t corresponds to the question of " $\mathbf{Z} \in A$?" where $A \subset R^p$. If we define the best split of t (s_t^*) as that which most decreases $R(T)$, then it can be written as follows

$$\max_{s_t \in S_t} \{R(t) - (R(t_L) + R(t_R))\}. \quad (3.6)$$

By the recursive application of this criterion to each terminal node, a maximum-sized tree, T_0 , can be constructed. The set of all possible splits (or rather questions), S_t , can be obtained as follows:

Rule 1. Each member of S_t is given by the value of only a single variable, Z_j , $j = 1, 2, \dots, p$.

Rule 2. If Z_j is a ordered variable, then S_t includes all " $Z_j \leq s$?" questions. There are at most N_t distinct values $(z_{j1}, z_{j2}, \dots, z_{jN_t})$ included in \mathcal{L}_t . The threshold value, s , is taken halfway between the consecutive, distinct values of $z_{j(i)}$ and $z_{j(i+1)}$, where $i = 1, 2, \dots, N_t - 1$ and $z_{j(1)} \leq z_{j(2)} \leq \dots \leq z_{j(N_t)}$. Therefore, there are at most $N_t - 1$ distinct splits, " $Z_j \leq s$?"

Rule 3. If Z_j is a categorical variable with possible values $F = \{f_{j1}, \dots, f_{jC_j}\}$, then S_t includes all " $Z_j \in R_{tj}$?" questions where $R_{tj} \subset F$. In this case, the number of all possible subset combinations is at most $2^{C_j-1} - 1$.

Let $\mathcal{L}_L = \{(x_i, \mathbf{z}_i); i = 1, \dots, N_L\}$ and $\mathcal{L}_R = \{(x_i, \mathbf{z}_i); i = 1, \dots, N_R\}$ be the learning samples in the child nodes t_L and t_R , respectively. Then, the

following relationship holds

$$R(t) - (R(t_L) + R(t_R)) = \frac{1}{N} \left\{ \sum_{i \in A_{\mathcal{L}_L}} (x_i - \bar{x}_t)^2 + \sum_{i \in A_{\mathcal{L}_R}} (x_i - \bar{x}_t)^2 \right\} - \left\{ \frac{1}{N} \sum_{i \in A_{\mathcal{L}_L}} (x_i - \bar{x}_{t_L})^2 + \frac{1}{N} \sum_{i \in A_{\mathcal{L}_R}} (x_i - \bar{x}_{t_R})^2 \right\},$$

where $A_{\mathcal{L}_L}$ and $A_{\mathcal{L}_R}$ are the set of labels for the samples in \mathcal{L}_L and \mathcal{L}_R , respectively. Since \bar{x}_{t_L} and \bar{x}_{t_R} minimize (3.5) for \mathcal{L}_L and \mathcal{L}_R , $R(t) - (R(t_L) + R(t_R))$ is always non-negative. Therefore, for any split of t into t_L and t_R , the error measure, $R(t)$, has the property

$$R(t) \geq R(t_L) + R(t_R). \quad (3.7)$$

Once the number of patients in the node is equal to or lower than the setting number (which is set to a sufficiently small number, like 1), it is not divided any further. The resulting tree, T_0 , is large and overfits the data. To deal with this problem, we prune the tree to an optimal size.

3.2.2 Pruning

In the pruning step, nested subtrees are obtained by recursively removing branches from T_0 . The nested subtrees are defined as T_1, T_2, \dots, T_M , where T_{m+1} is a subtree of T_m , $m = 1, 2, \dots, M - 1$. In this step, we use the cost-complexity measure

$$R_\alpha(T) = \sum_{t \in \tilde{T}} R(t) + \alpha |\tilde{T}|, \quad (3.8)$$

where $|\tilde{T}|$ is the number of terminal nodes in T . α represents its penalty and controls the optimal complexity of the tree.

For each value of α , we define an optimal subtree, $T(\alpha)$, which satisfies the following condition:

$$R_\alpha(T(\alpha)) = \min_{T \preceq T_0} R_\alpha(T), \quad (3.9)$$

where $T' \preceq T$ indicates " T' is a subtree of T ." Moreover, if $T(\alpha) \preceq T'(\alpha)$ for all optimal subtrees, $T'(\alpha)$, then $T(\alpha)$ is the smallest optimal subtree. When α has a high value, a small tree will have the minimum number of $R_\alpha(T)$ and will be considered optimal. On the other hand, a large subtree will be considered optimal when α has a low value. For instance, if α is 0 then an optimal subtree is T_0 .

In order to find the smallest optimal subtree for an arbitrary α , we employ the weakest-link cutting method. Let T_1 be the smallest optimal subtree of T_0 for $\alpha = 0$, which satisfies

$$R(T_1) = R(T_0).$$

Then, let T_t be any branch of T_1 with root node $t \in U_1$, where U_1 is the set of internal node of T_1 . Then, the following relationship (from (3.7)) holds

$$R(t) > R(T_t).$$

Based on the definition in (3.8), the cost-complexity measure of a tree that only has node t is given by

$$R_\alpha(t) = R(t) + \alpha. \quad (3.10)$$

Whereas, the cost-complexity measure of a branch, T_t , is given by

$$R_\alpha(T_t) = R(T_t) + \alpha|\tilde{T}_t|. \quad (3.11)$$

If the following relationship holds, then branch T_t has a smaller cost-complexity measure than node t

$$R_\alpha(T_t) < R_\alpha(t).$$

At some critical value of α , these two measures become equal. To find this critical value, we calculate the equation $R_\alpha(t) = R_\alpha(T_t)$ with (3.10) and (3.11), and obtain the following equation

$$\alpha = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}.$$

Based on this equation, the weakest link node t_1^* in T_1 is defined as that which satisfies the following condition

$$\alpha_2 = \min_{t \in U_1} \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1},$$

where U_1 is the set of internal nodes in T_1 .

A new tree, $T_2 \prec T_1$ (where $T' \prec T$ indicates that " T' is a pruned subtree of T "), can be obtained by pruning $T_{t_1^*}$ away from T_1 . In the same manner and by using T_2 instead of T_1 , we can find the weakest link node, t_2^* , for T_2 and the value of α_3 . Continuing in this way, we can get the sequence of the smallest optimal subtrees:

$$T_1 \succ T_2 \succ \cdots \succ T_M, \quad (3.12)$$

where T_M is the root node of T_0 . The obtained sequence α_m ($m = 1, 2, \dots, M$, where $\alpha_1 = 0$) is increased; that is, for $m \geq 1$, $\alpha_m < \alpha_{m+1}$ holds. For $\alpha_m \leq \alpha < \alpha_{m+1}$, T_m is the smallest subtree of T_0 and minimizes $R_\alpha(T)$ in (3.8). The proof of this can be found in Breiman et al. (1984).

3.2.3 Selection

The estimates of $R^*(T_m)$ in (3.1) are needed to select the optimal sized tree from (3.12). If we have a sufficiently large number of observations, we can use the test sample estimator to do this. That is, \mathcal{L} is randomly divided into the subsets $\mathcal{L}_1 = \{(x_i, \mathbf{z}_i); i = 1, \dots, N_1\}$ and $\mathcal{L}_2 = \{(x_i, \mathbf{z}_i); i = 1, \dots, N_2\}$ for training and testing, respectively, where $N_1 + N_2 = N$. Using the training set, \mathcal{L}_1 , a maximum size tree, T_0 , is constructed using the process presented in the splitting step, and a set of subtrees, T_1, T_2, \dots, T_M , is obtained through the pruning process. Then, the test sample estimator for T_m is defined as

$$R^{ts}(T_m) = \frac{1}{N_2} \sum_{i \in A_{\mathcal{L}_2}} (x_i - T_m(\mathbf{z}_i))^2, \quad (3.13)$$

where the predictor, $T_m(\mathbf{z}_i)$, is given by the sample mean of the training data in the terminal node t in T_m . $A_{\mathcal{L}_2}$ is the set of labels for the samples in \mathcal{L}_2 .

Unfortunately, in many cases, the data is insufficient to use this process. In response to this problem, we generally use the V -fold cross-validation method. The learning set \mathcal{L} is randomly divided into $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_V$, where $\mathcal{L}_v = \{(x_i, \mathbf{z}_i); i = 1, 2, \dots, N_v\}$, $v = 1, 2, \dots, V$. Each subsample has approximately the same number of patients ($N_1 \cong N_2 \cong \dots \cong N_V$). Using the training set, $\mathcal{L}^{(v)} = \mathcal{L} - \mathcal{L}_v$, a maximum size tree, $T_0^{(v)}$, is constructed and a set of subtrees $T_1^{(v)}, T_2^{(v)}, \dots, T_{M_v}^{(v)}$ is obtained. For all $v = 1, 2, \dots, V$, we repeat this process and create V sets of subtrees.

Next, we can calculate the sequences of subtrees T_1, T_2, \dots, T_M and corresponding complexity parameters $\alpha_1, \alpha_2, \dots, \alpha_M$ by using all of the learning samples \mathcal{L} . The value that satisfies $T(\alpha) = T_m$ is

$$\alpha'_m = \sqrt{\alpha_m \alpha_{m+1}}.$$

For each α'_m , we obtain the minimal cost-complexity subtrees, $T^{(v)}(\alpha'_m)$ ($v = 1, 2, \dots, V$). Then, the cross-validation estimate of T_m can be defined as

$$R^{cv}(T_m) = \frac{1}{N} \sum_{v=1}^V \sum_{i \in A_{\mathcal{L}_v}} (x_i - T_m^{(v)}(\mathbf{z}_i))^2, \quad (3.14)$$

where $T_m^{(v)}(\mathbf{z}_i)$ is the prediction for data i , which is obtained from the tree $T^{(v)}(\alpha'_m)$. $A_{\mathcal{L}_v}$ is the set of labels for the samples in \mathcal{L}_v .

Using either the test sample or the cross-validation estimator, the best subtree is defined as that which satisfies the following condition

$$\min_m \hat{R}(T_m),$$

where $\hat{R}(T_m)$ is given by (3.13) or (3.14).

3.3 Survival Tree

We denote the observed learning samples as $\mathcal{L} = \{(x_i, \delta_i, \mathbf{z}_i); i = 1, 2, \dots, N\}$, where x_i denotes the observed time until the event or censoring, δ_i is the censoring indicator, and $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{pi})$ is a p -dimensional covariate vector for the i th patient. Our next goal is to split the data, \mathcal{L} , into groups with differing survival times. Several splitting algorithms have been suggested to accomplish this (Ciampi et al., 1988; Segal, 1988; Davis and Anderson, 1989; Therneau et al., 1990; Leblanc and Crowley, 1992; Zhang, 1995). In this section, we detail the three splitting criteria. Additionally, we present Leblanc and Crowley's (1993) pruning and selection methods, which are based on log-rank test statistics as shown in (1.5).

3.3.1 Splitting

Let $\mathcal{L}_t = \{(x_i, \delta_i, \mathbf{z}_i); i = 1, 2, \dots, N_t\}$ be the subset of learning samples that are included in node t . As described in the previous section, a tree is formed by recursively dividing \mathcal{L}_t into $\mathcal{L}_L = \{(x_i, \delta_i, \mathbf{z}_i); i = 1, 2, \dots, N_L\}$ and $\mathcal{L}_R = \{(x_i, \delta_i, \mathbf{z}_i); i = 1, 2, \dots, N_R\}$ at each terminal node. The set of all possible splits, S_t , is determined by Rules 1–3 in Section 3.2.1. There are several different definitions of the best split of t ($s_t^* \in S_t$), though.

Davis and Anderson (1989) choose the split that minimizes loss based on an exponential log-likelihood loss criterion. The simplest and most widely used survival distribution is given by the exponential model, which has a constant hazard function. Let the hazard function of patients included in node t be

$$\lambda(x|\mathbf{z} \in t) = \lambda_t,$$

where λ_t represents a constant parameter ($\lambda_t > 0$). Then, the likelihood of \mathcal{L}_t can be obtained from (1.1) and written as

$$L_t(\lambda_t) = \prod_{i \in A_{\mathcal{L}_t}} \lambda_t^{\delta_i} \exp(-\lambda_t x_i),$$

where $A_{\mathcal{L}_t}$ is the set of labels, $\{i; \mathbf{z}_i \in t\}$ for the observations in node t . The maximum likelihood estimator of λ_t is given by

$$\hat{\lambda}_t = \frac{\sum_{i \in A_{\mathcal{L}_t}} \delta_i}{\sum_{i \in A_{\mathcal{L}_t}} x_i}.$$

By using L_t and $\hat{\lambda}_t$, the exponential log-likelihood loss of node t can be given by

$$\begin{aligned} R(t) &= -\log L_t(\hat{\lambda}_t) \\ &= \sum_{i \in A_{\mathcal{L}_t}} \delta_i - \sum_{i \in A_{\mathcal{L}_t}} \delta_i \log(\hat{\lambda}_t). \end{aligned} \tag{3.15}$$

As per (3.6), the optimal split is that which maximizes the improvement in this exponential log-likelihood loss.

Leblanc and Crowley (1992) use the node deviance measure in conjunction with the proportional hazard model, the full likelihood of which is approximated by replacing the cumulative baseline hazard function with the Nelson–Aalen estimator. They propose this approach because it is analogous to the splitting process presented in (3.6). Let the hazard function for node t be

$$\lambda(x|\mathbf{z} \in t) = \lambda_0(x)\theta_t,$$

where $\lambda_0(x)$ is the baseline hazard and θ_t is a nonnegative parameter. Then, we use (1.1) to obtain patients' likelihood, which is

$$L = \prod_{t \in \tilde{T}} \prod_{i \in A_{\mathcal{L}_t}} (\lambda_0(x_i)\theta_t)^{\delta_i} \exp(-\Lambda_0(x_i)\theta_t),$$

where $\Lambda_0(x)$ is the baseline cumulative hazard function. If we assume that $\Lambda_0(x)$ is given, then the maximum likelihood estimates of θ_t can be given by

$$\hat{\theta}_t = \frac{\sum_{i \in A_{\mathcal{L}_t}} \delta_i}{\sum_{i \in A_{\mathcal{L}_t}} \Lambda_0(x_i)}. \quad (3.16)$$

Moreover, the Breslow estimator of $\Lambda_0(x)$, given $\hat{\theta}_t$, is

$$\hat{\Lambda}_0(x) = \sum_{i \in A} \frac{\delta_i}{\sum_{t \in \tilde{T}} \sum_{j \in R_t(x_i)} \hat{\theta}_t}. \quad (3.17)$$

where $A = \{i; x_i \leq x, i = 1, 2, \dots, N\}$ and $R_t(x_i) = \{j; x_j \geq x_i, j \in A_{\mathcal{L}_t}\}$ are the sets of observation labels. By using an alternating estimation of (3.16) and (3.17), we can estimate θ_t and $\Lambda_0(x)$.

Leblanc and Crowley propose using the estimates obtained by only the first iteration of the CART algorithm equations. The Breslow estimator evaluated at $\hat{\theta}_t = 1$ is given as the Nelson–Aalen cumulative hazard estimator

$$\hat{\Lambda}_0^1(x) = \sum_{i \in A} \frac{d_i}{n_i},$$

where d_i and n_i represent the number of events and risk at x_i , respectively. Then, the one-step estimator of θ_t can be given by

$$\hat{\theta}_t^1 = \frac{\sum_{i \in A_{\mathcal{L}_t}} \delta_i}{\sum_{i \in A_{\mathcal{L}_t}} \hat{\Lambda}_0^1(x_i)}.$$

Using these estimates, the deviance residual (Therneau et al., 1990) of node t can be defined as

$$\begin{aligned}
R(t) &= \sum_{i \in A_{\mathcal{L}_t}} \left[\operatorname{sgn} \left(\delta_i - \hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1 \right) \sqrt{-2 \left\{ (\delta_i - \hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1) + \delta_i \log(\hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1) \right\}} \right]^2 \\
&= \sum_{i \in A_{\mathcal{L}_t}} 2 \left[\delta_i \log \delta_i - \delta_i \log(\hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1) - (\delta_i - \hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1) \right] \\
&= \sum_{i \in A_{\mathcal{L}_t}} 2 \left[\delta_i \log \left(\frac{\delta_i}{\hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1} \right) - (\delta_i - \hat{\Lambda}_0^1(x_i) \hat{\theta}_t^1) \right]. \tag{3.18}
\end{aligned}$$

As in (3.6), the optimal split is that which maximizes the improvement in the deviance.

Because the algorithm for constructing a survival tree splits the data into groups with differing survival times, the log-rank test given in (1.5) naturally presents itself as a suitable splitting criterion. Many authors have studied this criterion (Ciampi et al., 1988; Segal, 1988; Leblanc and Crowley, 1993). The method of using the degree of separation between the child nodes' estimated survival functions or hazard functions (i.e., log-rank test statistics) differs greatly from those based on exponential log-likelihood loss or the deviance residual (as described above). That is, in the aforementioned methods, the best split of t , s_t^* , is defined as that which minimizes the heterogeneity measure, $R(t)$, and it is given by (3.6). On the other hand, in the method that uses the degree of separation between estimated survival or hazard functions, $G(t)$, the best split, s_t^* , is defined such that

$$\max_{s_t \in S_t} \{G(t)\}.$$

In our case, $G(t)$ is the log-rank test statistic for child nodes t_L and t_R :

$$G(t) = \frac{\sum_{i \in A_{\mathcal{L}_t}} (d_{Li} - e_{Li})}{\sqrt{\sum_{i \in A_{\mathcal{L}_t}} v_{Li}}}, \tag{3.19}$$

where d_{Li} is the number of events in one child node (t_L or t_R) at x_i . e_{Li} and v_{Li} are the expectation and variance of d_{Li} , respectively (these are defined in (1.4)).

Just as in the construction of a regression tree, a maximum-size tree T_0 is constructed by recursively splitting the data in each terminal node. Subsequently, we must prune T_0 and select the optimal size tree to deal with the problem of overfitting.

3.3.2 Pruning

If the criterion that minimizes a node's heterogeneity measure, $R(t)$ (discussed in (3.15) and (3.18)), is used in the splitting step, then the cost-complexity measure, (3.8), should be used in the pruning step. On the other hand, if the criterion that maximizes the degree of separation between nodes, $G(t)$ (discussed in (3.19)), is used, then Leblanc and Crowley's (1993) split-complexity measure must be used. This measure is defined as follows

$$G_\alpha(T) = \sum_{t \in U} G(t) - \alpha|U|,$$

where U is the set of internal nodes in T , and $|U|$ is the number of elements, which is the tree's complexity measure.

As in (3.9), we define the optimal subtree, $T(\alpha)$, for each value of α so that it satisfies the following condition

$$G_\alpha(T(\alpha)) = \max_{T \preceq T_0} G_\alpha(T),$$

and if $T(\alpha) \preceq T'(\alpha)$ for all optimal subtrees, $T'(\alpha)$, then $T(\alpha)$ is the smallest optimal subtree. If α is small, then the optimal subtree is large and has a maximum number of $G_\alpha(T)$. As α increases, the optimal subtree decreases in size until it consists of only the root node.

To find the smallest optimal subtree for an arbitrary α , we utilize the weakest-link cutting methods (as discussed in Section 3.2.2). Let T_1 be the smallest optimal subtree of T_0 for $\alpha = 0$. Furthermore, $G(T)$ represents the sum of the degrees of separation between nodes in T :

$$G(T) \equiv \sum_{t \in U} G(t).$$

If we define any branch of T_1 with root node $t \notin U_1$ as T_t , then the critical value of α (i.e., the value at which the split-complexity measure, $G_\alpha(T_t)$, becomes less than 0) is given by the following equation

$$\alpha = \frac{G(T_t)}{|U_t|},$$

where U_t is the set of internal nodes in T_t , and $|U_t|$ is the number of internal nodes. The weakest link node in T_1 , t_1^* , is defined as that for which

$$\alpha_2 = \min_{t \in U_1} \frac{G(T_t)}{|U_t|},$$

where U_1 is the set of internal nodes in T_1 .

By pruning $T_{t_1^*}$ away from T_1 , we can obtain a new tree, $T_2 \prec T_1$. By repeating this operation, we can obtain a sequence of the smallest optimal subtrees, $T_1 \succ T_2 \succ \dots \succ T_M$, and the sequence $\alpha_1 < \alpha_2 < \dots < \alpha_M$. The proof that T_m is the smallest optimal subtree of T_0 for $\alpha_m \leq \alpha < \alpha_{m+1}$, $m = 1, 2, \dots, M$, can be found in Leblanc and Crowley (1993).

3.3.3 Selection

As explained in Section 3.2.3, if we construct a survival tree based on the heterogeneity measure within a node $R(t)$, then the test-sample estimator or the V -fold cross-validation estimator can be used to select an optimal sized tree from the sequence of nested subtrees. If we construct a survival tree based on the degree of separation between nodes $G(t)$, then the test-sample estimator or the bootstrap estimator may be used. The test sample estimator may be obtained as follows: First, we assume that a large sample is available. Then, the sample is divided into a training set, \mathcal{L}_1 , which is used to construct a sequence of subtrees (T_1, T_2, \dots, T_M) , and a test set, \mathcal{L}_2 , which is used to evaluate the tree's performance. For each subtree, T_m , $m = 1, 2, \dots, M$, the test samples, \mathcal{L}_2 , are sent down, and the splitting statistics $\hat{G}(t)$ are calculated for each internal node. The optimal tree is defined that which satisfies the following condition

$$\max_m \left\{ \sum_{t \in U_m} \hat{G}(t) - \alpha_c |U_m| \right\}, \quad (3.20)$$

where U_m is the set of internal nodes in T_m , and $|U_m|$ is the number of elements. Leblanc and Crowley (1993) recommend selecting the penalty for each internal node α_c such that $2 \leq \alpha_c \leq 4$.

If we cannot use a large sample set, bias correction techniques are employed in the subtree selection (Leblanc and Crowley, 1993). In order to do this, we first obtain a sequence of subtrees, T_m , and corresponding parameters, α_m , $m = 1, 2, \dots, M$, by using all of the learning samples, \mathcal{L} . We use the value $\alpha'_m = \sqrt{\alpha_m \alpha_{m+1}}$ that satisfies $T(\alpha) = T_m$. Next, we draw B bootstrap samples from \mathcal{L} ($\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_B$). In the following notation

$$G(\mathcal{L}_2; \mathcal{L}_1, T) \equiv \sum_{t \in U} G(t),$$

where \mathcal{L}_1 is the set of samples that shall be used to build the tree, T , and \mathcal{L}_2 is the set of samples that shall be used to evaluate $G(t)$ in the tree. For each bootstrap sample, \mathcal{L}_b , $b = 1, 2, \dots, B$, we construct a maximum size tree and find the smallest optimal subtree, $T_b(\alpha'_m)$, for each α'_m . Then, we calculate the following value

$$o_{b_m} = G(\mathcal{L}; \mathcal{L}_b, T_b(\alpha'_m)) - G(\mathcal{L}_b; \mathcal{L}_b, T_b(\alpha'_m)),$$

for $b = 1, 2, \dots, B$ and $m = 1, 2, \dots, M$. For each subtree, T_m , the mean value of the o_{b_m} bootstrap samples is calculated as

$$\bar{o}_m = \frac{1}{B} \sum_{b=1}^B o_{b_m}.$$

Using \bar{o}_m , we calculate the sum of $G(t)$ over the internal nodes and then use the bootstrap technique to correct for bias as follows

$$\sum_{t \in U_m} \hat{G}(t) = G(\mathcal{L}; \mathcal{L}, T_m) + \bar{o}_m. \quad (3.21)$$

Finally, we are able to build a tree that satisfies the condition in (3.20) and substitute in the bootstrap estimator, (3.21). Through simulation studies and real data analysis, Leblanc and Crowley show that $B \geq 25$ is a sufficient number of the bootstrap samples for the selection step.

3.4 Conclusion

In this chapter, we show the processes for constructing tree-structured prediction models based on the CART algorithm. In particular, we focus on regression and survival trees. In the splitting and pruning steps, heterogeneity measures and cost-complexity measures are taken into account. There are two main methods for constructing survival trees. One method uses the node's heterogeneity measure and cost-complexity measure, while the other uses the degree of separation between nodes and the split-complexity measure. We present three methods for splitting, the test-sample estimator, the V -fold cross-validation estimator, and the bootstrap estimator. In the following chapters, we examine the tree-structured prediction models that rely on the methods discussed in this chapter.

Chapter 4

Application of Survival Tree Based on Texture Features Obtained Through MRI Images of Brain Cancer Patients

4.1 Introduction

In this chapter, we examine the possibility of MRI for providing the prognosis of cancer patients. In 10% of cancer patients, the probability of brain metastases occurring within a year is high and the survival time of these patients is approximately 1 year (Narita and Shibui (2009)). The typical prognostic factors that were determined by recursive partitioning analysis (RPA) are as follows: the Karnofsky Performance Scale (KPS), primary lesion (controlled vs. uncontrolled), age, and extracranial systemic metastases (present or absent) (Gaspar et al. (1997)). According to this classification, patients with a KPS score lower than 70 (RPA III) have the worst survival prognosis. The patients with the best survival prognosis have a KPS of at least 70, are under 65 years of age, and have a controlled primary tumor (RPA I). This RPA classification is commonly used in clinical situations involving brain metastases. In a more recent research study, the effectiveness of the factors postoperative systemic therapy, controlled or uncontrolled extra-cranial malignancy, and postoperative KPS was shown from the aspect of early death (Arita et al. (2014)). Recently, the Graded Prognostic Assessment (GPA), which is a prognostic index for patients with brain metastases, was published (Sperduto et al. (2012)). This index, which comprises diagnosis-specific prognostic indices based on age, KPS, presence of extracranial systemic metastases, the number of brain metastases, and primary cancer site, can be used for the assessment of patients. Although these factors are used to estimate the survival time of cancer patients, there are many covariates that can potentially be used for this purpose. Recently, several studies on

estimating the prognosis of patients with gliomas that used covariates obtained through MRI have been reported (Li et al. (2004), Sanz-Requena et al. (2013)). MRI is necessary for the diagnosis of and treatment decision for patients with brain metastases.

The aim of this research study is to evaluate the covariates of texture patterns obtained from preoperative T1-weighted MRI scans of patients with metastatic brain tumor from breast cancer. Texture features are obtained by using the gray-level co-occurrence matrix (GLCM) and wavelet transformation, which are shown in Section 2.3. Same as in Chapter 2, ten statistical features, such as mean and variance, extracted from the GLCM are used as covariates. Further, sixteen features, such as energy and entropy, are used as covariates by wavelet transformation using the Daubechies D_6 and D_{20} wavelets. Moreover, age, KPS score, and the indicator of whether solitary or multiple metastatic are present are used as the covariates.

To identify novel prognostic factors, we use the survival tree method which discussed in previous chapter. By constructing the tree-structured model, we want to construct a model which classifies the patients into groups with similar covariates and prognosis. Although some authors used CART algorithm to construct tree-structured models for brain tumor patients (Gaspar et al. (1997), Sperduto et al. (2012), Arita et al. (2014), Li et al. (2004)), there are some differences in the splitting criterion used in the algorithm. Because there is no gold standard for this criterion, we used three criteria proposed in Section 3.3.1.

This chapter is organized as follows. The detail of the patients, the methods of image analysis, and the methods of statistical analysis are described in the next section. In Section 4.3, we show the experimental results. The conclusion is described in Section 4.4.

4.2 Patients and Methods

4.2.1 Patients

The data of forty patients with brain metastases from breast cancer were analyzed in this study. All patients had at least a single metastasis, whose diameter was greater than 3 cm. The patients underwent surgery as an initial treatment. The analysis in this study was approved by the local institutional review board of the national cancer center. The patients underwent MRI examinations prior to surgery. The data were collected from the years 2000 to 2011. All patients were female, and their cancer onset age ranged from 37 to 79 years. They underwent surgery and standard radiation therapy. The observation period for analysis was defined as the time between surgery and death, when the date of death could not be confirmed, it was defined as the time until the final confirmation of survival. MRI data were acquired approximately 1 week before surgery.

All images were obtained using T1, T2, FLAIR (fluid-attenuated inversion recovery) and T1+Gd-DTPA (gadolinium with diethylenetriamine-pentaacetic acid) imaging. Among these images, we focused on T1 images, which were obtained using the following setting: axial plane, acquisition matrix $(N_m \times N_n) = (256 - 512) \times (256 - 512)$, 19 – 27 slices (in the case of one patient, only 10 slices), gray-level range was transformed into $b = 8$ bits, $(y_1, y_2, \dots, y_B = y_{256}) = (0, 1, \dots, 255)$.

4.2.2 Image analysis

All images are skull stripped images and were processed by using the watershed transform technique, semiautomatically (Horst and Heinz (2000)). Two types of texture features are extracted from the skull stripped images: statistical texture features extracted from the GLCM of the image matrix and texture features extracted by using the wavelet transform of the image matrix. To calculate the GLCM for each image, we use $\mathbf{d} = (5, -5)$ as the distance and direction parameter. Ten features which are described in Section 2.3.1 are obtained from the GLCM: contrast, correlation, energy, homogeneity, entropy, mean, deviation, entropy from difference of GLCMs, energy from difference of GLCMs, and mean from difference of GLCMs. The Daubechies D_6 and D_{20} wavelets are selected for use in wavelet transformation, following the method used in Section 2.3.2. Sixteen features were obtained by wavelet transformation: energy and entropy from one signal approximation coefficient and three wavelet coefficients. All features were extracted from each slice, and the mean values were defined as covariates.

4.2.3 Statistical analysis

All patients are classified according to the covariates, which were age, KPS score, solitary or multiple brain metastases indicator, and the 26 texture features described in the previous section. The summary statistics of these covariates are shown in Table 4.1. To identify effective prognostic factors and seek their splitting point for grouping, we use the survival tree method based on three splitting criteria: the exponential log-likelihood loss (EL) given by (3.15), the deviance residual obtained from proportional hazard model (PD) given by (3.18), the log-rank test statistics (LR) given by (3.19).

In this study, each setting for constructing survival trees was defined as follows. The minimum number of events in nodes as the stop condition of the splitting step is set to 5 or less. In selection step, the optimal size of the trees is determined based on the 10-fold cross-validation results for the analyses based on EL or PD splitting criterion. On the other hand, 25-bootstrap technique is used about LR splitting criterion. Kaplan-Meier curves were plotted for each group of patients, categorized by the terminal nodes in the tree-structured

Table 4.1: Summary statistics of the covariates

Covariates			Mean	Standard Deviation	Cox p -value
Commonly used covariates					
	age		52.375	9.981	0.432
	KPS	60	4	7.232	0.154
		70	22		
		80	12		
		90	2		
	S/M metastatic	single	29	0.452	0.573
		multiple	11		
GLCM					
	contrast		539.892	243.121	0.815
	correlation		0.429	0.099	0.158
	energy		0.001	0.001	0.400
	homogeneity		0.200	0.031	0.717
	entropy		7.516	0.390	0.186
	mean		85.904	16.827	0.923
	deviation		21.089	4.552	0.382
	entropy of dif.		3.555	0.225	0.845
	mean of dif.		0.046	0.011	0.982
	energy of dif.		13.981	3.328	0.778
Wavelet transform					
	energy of app. (D6)		8252.027	3481.806	0.643
	energy of hor. (D6)		14.940	9.863	0.026
	energy of ver. (D6)		16.968	9.674	0.492
	energy of dia. (D6)		4.775	3.812	0.100
	energy of app. (D20)		7489.638	3134.975	0.600
	energy of hor. (D20)		11.813	8.512	0.015
	energy of ver. (D20)		13.294	8.181	0.521
	energy of dia. (D20)		3.761	3.154	0.087
	entropy of app. (D6)		3.583	0.181	0.113
	entropy of hor. (D6)		2.793	0.221	0.046
	entropy of ver. (D6)		2.789	0.215	0.051
	entropy of dia. (D6)		2.644	0.171	0.051
	entropy of app. (D20)		3.584	0.181	0.112
	entropy of hor. (D20)		2.901	0.205	0.044
	entropy of ver. (D20)		2.890	0.183	0.042
	entropy of dia. (D20)		2.789	0.151	0.062

In the KPS and S/M metastatic items, the value of the mean is the number of patients. app.: approximation coefficient; hor.: horizontal wavelet coefficient; ver.: vertical wavelet coefficient; dia.: diagonal wavelet coefficient.

model. Let $\mathcal{L}_t = \{(x_i, \delta_i, \mathbf{z}_i) : i = 1, 2, \dots, N_t\}$ be the set of learning samples in the node t . Then, the Kaplan-Meier estimation of t is defined by using \mathcal{L}_t as (1.2):

$$\hat{S}_t(x) = \begin{cases} 1 & (x < y_{(1)}) \\ \prod_{i \in A_{\mathcal{L}_t}} \left(1 - \frac{d_i}{n_i}\right) & (y_{(1)} \leq x) \end{cases}, \quad (4.1)$$

where $A_{\mathcal{L}_t} = \{i; x_i \leq x, i = 1, 2, \dots, N_t\}$ is the set of observation labels included in node t , and $y_{(1)}$ represents the earliest event occurrence time in \mathcal{L}_t . d_i and

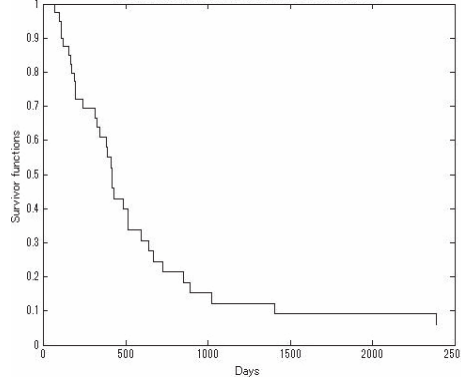


Figure 4.1: Kaplan-Meier survival curve for all patients.

n_i represent the number of events and risk at time x_i , $i = 1, 2, \dots, N_t$, in \mathcal{L}_t , respectively. Moreover, logrank test are performed to compare the relationship between the groups.

To determine which covariates showed significant predictive values, individual Cox regression analyses are also performed separately. Moreover, the covariates that are considered to be useful for estimating a prognosis are included in a second Cox analysis.

4.3 Results

The overall Kaplan-Meier curve for all patients is displayed in Figure 4.1. The median survival time is 414 days, while the five-year survival rate was only 9.2%. The results of the Cox regression analysis of individual covariates are shown in Table 4.1. According to the p -values, which are calculated by a Wald test, the covariates commonly used in medical research are not statistically significant for prognosis nor are the covariates extracted from the GLCM. However, the five covariates obtained by wavelet transformation show p -values lower than 0.05. By stepwise selection with an entry parameter of 0.1 and stay parameter of 0.05, the energy of the horizontal and vertical wavelet coefficients (D_{20}) are included in the Cox regression model, with a p -value of 0.013 and 0.002 for the horizontal and vertical coefficients, respectively.

Survival trees are constructed using the 29 covariates described in Table 4.1. The EL, PD, and LR criteria yielded the different results. For the EL criterion, the obtained tree structure is shown in Figure 4.2. The circle and squares in the figure represent the internal nodes and terminal nodes, respectively. The values in the shapes represent the number of patients in the node, and the values in parentheses represent the number of events. The tree has only one split and two terminal nodes; the covariate used in the tree is the energy extracted from the

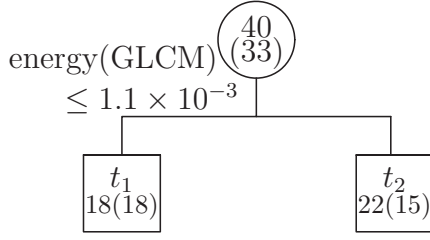


Figure 4.2: Survival tree by EL criterion.

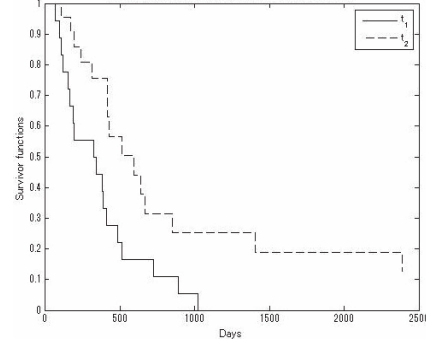


Figure 4.3: Kaplan-Meier survival curves for the terminal nodes in Figure 4.2.

GLCM. The Kaplan-Meier survival functions for each terminal node (t_1, t_2) are shown in Figure 4.3. The patients in node t_1 , which includes patients having a lower GLCM energy, have a higher risk of death than those in node t_2 . The median survival of the patients in node t_1 is 323 days, and their five-year survival rate is 0%. However, the median survival of patients in node t_2 is 593 days, and their five-year survival rate is 12.6%. The p -value of the logrank test for nodes t_1 and t_2 is 0.0091, showing that a statistically significant difference in prognosis existed between the two groups of patients.

For the PD criterion, the obtained tree structure is given in Figure 4.4. The tree has two splits and three terminal nodes. The covariates used in the tree are the energy of the horizontal wavelet coefficient (D_{20}) and the energy of the diagonal wavelet coefficient (D_6). The Kaplan-Meier survival functions for each terminal node ($t_1 - t_3$) are shown in Figure 4.5. The patients in node t_3 , which includes patients whose energy of horizontal wavelet coefficient (D_{20}) is higher, had the highest risk of death; those in node t_2 , which includes patients whose energy of horizontal wavelet coefficient (D_{20}) is lower and energy of diagonal wavelet coefficient (D_6) is higher, had the lowest risk of death. The median survival time of the patients included in nodes t_1 , t_2 , and t_3 is 323 days, 672 days, and 114 days, respectively. The five-year survival rate of patients included in these nodes is 0%, 12.89%, and 0%, respectively. The p -values of the logrank tests for each combination of nodes show that statistically significant differences in prognosis exist between the groups: p -value ($t_1 - t_2$)=0.0003; p -value ($t_2 - t_3$) < .0001; p -value ($t_3 - t_1$)=0.0037.

The tree structure obtained by the LR criterion is given in Figure 4.6. The tree has one split and two terminal nodes; the covariate used in the tree is the energy of the horizontal wavelet coefficient (D_6). The Kaplan-Meier survival functions for each terminal node (t_1, t_2) are shown in Figure 4.7. The patients in node t_1 , which includes the patients whose energy of horizontal wavelet co-

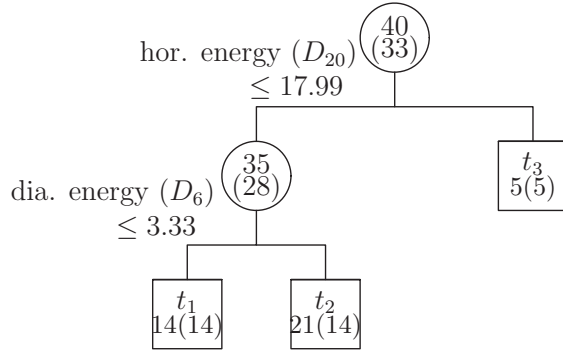


Figure 4.4: Survival tree by PD criterion.

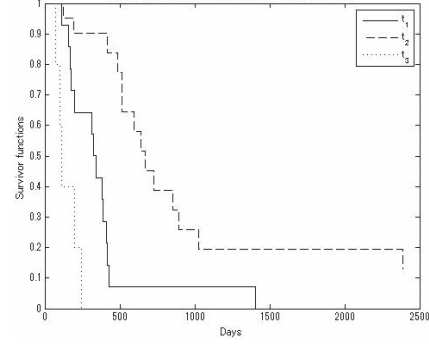


Figure 4.5: Kaplan-Meier survival curves for the terminal nodes in Figure 4.4.

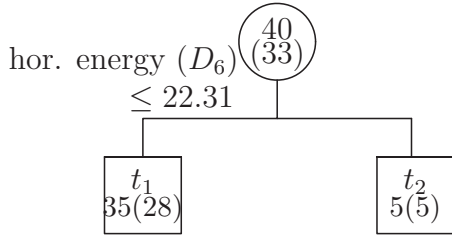


Figure 4.6: Survival tree by LR criterion.

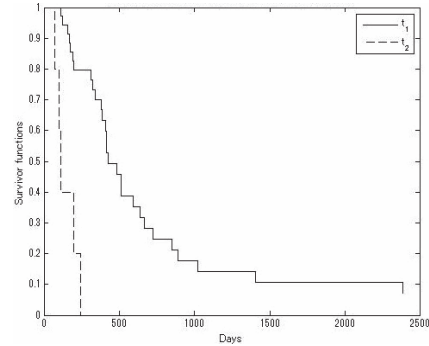


Figure 4.7: Kaplan-Meier survival curves for the terminal nodes in Figure 4.6.

efficient (D_6) is lower, had a lower risk of death than those in node t_2 . The median survival time of the patients in node t_1 is 431 days, and their five-year survival rate was 7.02%. In contrast, the median survival time of the patients in node t_2 is 114 days, and their five-year survival rate is 0%. The p -value of the log-rank test is $< .0001$, showing that a statistically significant difference in prognosis existed between the two groups.

4.4 Conclusion

The prognosis of patients with brain metastases depends on various factors. We focus on the more objective features obtained from MRI and searched the factors that affect survival. The texture patterns of images, which are obtained by GLCM and wavelet transformation, are proven to be useful in this study.

The Cox proportional hazard model and the survival tree method are used to evaluate the covariates. The results showed that the objective features obtained from simple medical images can be used effectively to obtain a prognosis.

To identify which covariates show significant predictive values, 26 covariates relevant to texture pattern and three covariates that are commonly used in medical research for the purpose of prognosis are investigated using Cox regression analyses. Three commonly used covariates, age, KPS, and solitary or multiple brain metastases indicator, do not yield statistically significant results for prognosis in this study. As regards the texture patterns, 10 features obtained from GLCM also do not yield statistically significant results. In contrast, some of the 16 features obtained by wavelet transformation yield significant results. Moreover, the model that includes two features obtained by wavelet transformation is selected as the optimal regression model by the stepwise selection method.

Next, we performed an additional analysis using the survival tree method, which is assumed to be a nonparametric model. The survival tree method based on the classic CART algorithm segregates patients into groups with similar covariates and prognosis. Several criterion functions for the splitting step in the algorithm have been proposed, we used three criteria. The trees obtained based on each criterion each show different results. The covariates selected in the final trees are texture features obtained from MRI images: energy extracted from GLCM, energy of horizontal wavelet coefficient (D_{20}), energy of diagonal wavelet coefficient (D_6), and energy of horizontal wavelet coefficient (D_6). The Kaplan-Meier survival curves for each group are well separated, and the p -values of the log-rank tests show statistically significant differences between the groups.

Although age and KPS, in particular, are considered effective prognostic factors for brain metastases patients according to a previous large scale survey, we could not find evidence that these covariates are effective for estimating the survival time of patients with brain metastases from breast cancer. In contrast, texture features obtained from MRI were determined to be effective prognosis factors in Cox analysis and survival tree analysis. In particular, the energy of GLCM and wavelet coefficients showed high-level performance in tree structure analysis.

Finally, the Cox model is an effective, powerful, and widely used tool in the analysis of survival data. However, this model requires several assumptions and its interpretation is not easy when this model includes many covariates. The survival tree method, which recursively partitions the data into groups with similar covariates and prognosis, can determine the relationship between covariates and hazards easily. Moreover, a new patient can easily be incorporated into the model. In this chapter, we demonstrate the application of the survival tree method and the effectiveness of using this method for survival analysis. According to the results obtained using the Cox model and the survival tree method, texture patterns obtained by simple preoperative MRI can be considered valid prognosis factors for patients with brain metastases from breast cancer.

Chapter 5

Survival Tree Based on Combination of Covariates

5.1 Introduction

In this chapter, we conduct a study on the splitting rule based on two or more combinations of covariates for survival tree analysis. As described in Section 3.2.1, the splitting rule of a node in traditional classification and regression tree (CART) algorithm is restricted by only one covariate. That is, if Z_j is a continuous variable, then the rule is the form of " $Z_j \leq s$ ". If Z_j is a categorical variable with possible values $F = \{f_{j1}, f_{j2}, \dots, f_{jC_j}\}$, then the rule is the form of " $Z_j \in R_{tj}$ " where $R_{tj} \subset F$. Although constructing the tree-structured model under this restriction has several merits like short learning time and inhibition of over learning, there are some difficult problems like exclusive OR (XOR) problem. That is, if the true model is linearly-inseparable, then the tree becomes prohibitively large and sometimes it becomes cause to lower prediction accuracy of the obtained model. As other example, consider the case that the true model has high probability of event in a small partial space in p -dimensional covariates space. Then, if we construct a tree-structured model based on traditional CART algorithm, $2 \times p$ splits are needed in the model. Since the data in each node are divided repeatedly, the number of patients which are used to evaluate a split in a node becomes small by splitting. Therefore, the estimation of an optimal splitting point becomes bad by splitting in generally.

To handle these situations, we can consider the approach that two or more combinations of covariates are considered for dividing the data in each node. For example, if $p = 2$, and Z_1 and Z_2 are categorical variables, a splitting rule of a node can be represented as " $Z_1 \in R_{t1} \cap Z_2 \in R_{t2}$ " or not. Although this approach is considered to need a long learning time, there is possibility to construct a more suitable model in some situations like XOR problem because the size of model is constricted. We show it through the comparative research by simulations. As the simulation-based evaluation method, we use the integrated

Brier score (Graf et al. (1999)) , which is widely used for evaluation of the predict model in survival analysis. Moreover, we present the result of an actual analysis based on proposed approach in the end of this chapter.

The remainder of this chapter is organized as follows. In the next section, the detail of approach that is studied in this chapter is described. The comparative simulation methods and results are given in Section 5.3. The result of an actual analysis using leukemia patients bone marrow transplantation data is shown in Section 5.4. Finally, the conclusion of this chapter is given in Section 5.5.

5.2 Splitting method based on combination of covariates

In traditional CART algorithm for constructing a tree-structured model, which is described in Chapter 3, the splitting rule of the node is restricted by only one covariate (Rule 1. in Section 3.2.1). As described in previous section, this restriction has the possibility of overlooking a more suitable model in some situations. To handle these situations, we consider the splitting rule that two or more combinations of covariates are considered for dividing the data in each node. For example, suppose a case that the combination of two covariates is considered in splitting step. If the covariate vector is defined as $\mathbf{Z} = (Z_1, Z_2, Z_3)$, where Z_j is the numerical variable ($j = 1, 2, 3$), then the splitting rule in each node for traditional approach is one of the following structures:

$$\{Z_1 \leq s_1\}, \{Z_2 \leq s_2\}, \{Z_3 \leq s_3\}. \quad (5.1)$$

On the other hand in our approach, the splitting rule in each node is one of (5.1) or following structures:

$$\begin{aligned} &\{Z_1 \leq s_1 \cap Z_2 \leq s_2\}, \{Z_1 \leq s_1 \cap Z_2 > s_2\}, \{Z_1 > s_1 \cap Z_2 \leq s_2\}, \\ &\{Z_1 > s_1 \cap Z_2 > s_2\}, \{Z_2 \leq s_2 \cap Z_3 \leq s_3\}, \{Z_2 \leq s_2 \cap Z_3 > s_3\}, \\ &\{Z_2 > s_2 \cap Z_3 \leq s_3\}, \{Z_2 > s_2 \cap Z_3 > s_3\}, \{Z_1 \leq s_1 \cap Z_3 \leq s_3\}, \\ &\{Z_1 \leq s_1 \cap Z_3 > s_3\}, \{Z_1 > s_1 \cap Z_3 \leq s_3\}, \{Z_1 > s_1 \cap Z_3 > s_3\}. \end{aligned}$$

In this example we consider the case that the variables included in covariate vector are numeric only, but if categorical variables are included in the vector we can consider as the same.

In traditional approach, as can be seen from this example, each splitting rule is practiced along single covariate axis. On the other hand, if we consider the combination of covariates for splitting, then it is practiced by whether data are included in a hyper-rectangle or not in some covariate spaces. Therefore, we can address several problems such as proposed in previous section.

As disadvantage of this approach, it is required a long learning time because it need to evaluate a large number of combinations of covariates. Specifically,

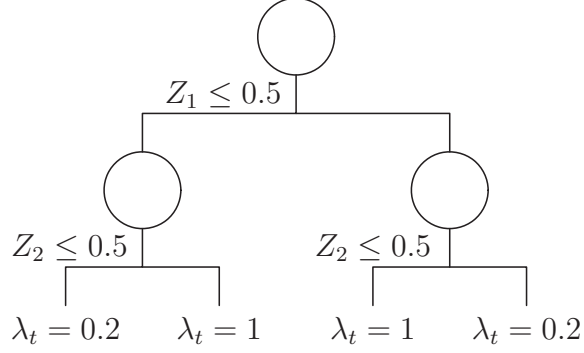


Figure 5.1: True tree structure used in simulation

the number of structures of splitting rule which are considered in this approach is $\binom{p}{k}2^k$ when the case of k combination of covariates is considered, while the number of it is p in traditional approach. However, computing technology in recent years may be able to resolves this disadvantage. As one merit of this approach, the pruning and selection steps in traditional CART algorithm can be used as traditional approach. Moreover, the processing time of these steps is not increased compared to the traditional approach.

5.3 Simulation

5.3.1 Model and Setting

To compare the traditional approach and the approach which takes into account the combination of covariates through simulations, we use the true tree model which are shown in Figure 5.1. The circles in the figures represent internal nodes. This model shows a typical XOR problem. The covariates used in this simulations are Z_1, Z_2 and Z_3 . These are random values created from two patterns: one is a discrete uniform distribution with $\{1/50, \dots, 1/50\}$, and the other is a Bernoulli distribution with parameter 0.5. This model assumes that the variables Z_1 and Z_2 are used in the true tree model and Z_3 is a nuisance.

We suppose exponential survival model, which has constant hazard to the change of time, for the simulations. This survival function is given by

$$S(y; \lambda_t) = \Pr(Y > y; \lambda_t) = \exp(-\lambda_t y),$$

where the λ_t represents the constant hazard of node t . Based on the trees which are shown in Figure 5.1, we suppose λ_t as follows:

$$\lambda_t = \begin{cases} 0.2 & (Z_1 \leq 0.5 \cap Z_2 \leq 0.5) \\ 1 & (Z_1 \leq 0.5 \cap Z_2 > 0.5) \\ 1 & (Z_1 > 0.5 \cap Z_2 \leq 0.5) \\ 0.2 & (Z_1 > 0.5 \cap Z_2 > 0.5) \end{cases}.$$

This model is difficult to detect in traditional approach, because the evaluations obtained by splitting in first node are nearly the same in each case that the covariate Z_1 , Z_2 , or Z_3 is used as splitting criterion.

By using uniform random numbers, the censoring rates are set as 0% and approximately 25% and 50%. The number of learning samples N are set to 200. The 5-fold cross validation is used in the selection step. We set 30 minimum number of events in nodes as the stop condition of splitting. Simulations are repeated 100 times in every setting. The number of covariates which are considered to construct a combination for splitting in proposed approach is restricted to two.

5.3.2 Evaluation methods

The integrated Brier score for censored data, which are proposed by Graf et al. (1999), is used to evaluate the proposed approach. This score is calculated based on the test samples $\mathcal{L}_{\text{test}} = \{(x_i, \delta_i, \mathbf{z}_i); i = 1, 2, \dots, N_{\text{test}}\}$. For each simulations, this $\mathcal{L}_{\text{test}}$ is obtained from the simulated population of the same setting. Let $\hat{S}(x|\mathbf{z}, T)$ be the estimated survival function of T . If censoring data are not included in $\mathcal{L}_{\text{test}}$, the expected Brier score of T at fixed time point x is defined as the mean square error:

$$\begin{aligned} \text{BS}_T(x) &= E[(I(Y > x) - \hat{S}(x|\mathbf{z}, T))^2] \\ &= \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (I(x_i > x) - \hat{S}(x|\mathbf{z}_i, T))^2. \end{aligned}$$

To consider the expected Brier score for all x about $0 \leq x \leq \max(x_i)$, we can consider the integrated Brier score for $\hat{S}(x|\mathbf{z}, T)$, which is defined as

$$\text{IBS}_T = \frac{1}{\max(x_i)} \int_0^{\max(x_i)} \text{BS}_T(x) dx.$$

To consider the situation that possibility of censoring data are included in $\mathcal{L}_{\text{test}}$, we need to consider the individual contributions of the test data to the Brier score. For a fixed time point x , the contribution of a data i is categorized to one of three cases: data with $I(x_i > x) = 0$ and uncensored $\delta_i = 1$, data with $I(x_i > x) = 1$ and censored/uncensored $\delta_i = 0$ or 1, data with $I(x_i > x) = 0$ and censored $\delta_i = 0$. For the last case of the data which are $I(x_i > x) = 0$ and censored $\delta_i = 0$, the information about contribution to the Brier score can not be calculated. For the first and second cases of the data, the loss of information due to censoring are weighted by using $\hat{G}(x)$ which is the Kaplan-Meier estimate of the censoring distribution of C . That is, $\hat{G}(x)$ is obtained by using (1.2) based on $\{(x_i, 1 - \delta_i); i = 1, 2, \dots, N_{\text{test}}\}$. Then, the weighted empirical Brier score

under censorship is defined as

$$\begin{aligned} \text{BS}_T(x) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \{ & (0 - \hat{S}(x|\mathbf{z}_i, T))^2 I(x_i \leq x, \delta_i = 1)(1/\hat{G}(x_i)) \\ & + (1 - \hat{S}(x|\mathbf{z}_i, T))^2 I(x_i > x)(1/\hat{G}(x)) \}. \end{aligned}$$

We use the up to median time in learning sample to evaluate this score. The number of test samples N_{test} is set to 200 for every setting.

Moreover, we use the following criterion as the measure of the explained residual variation:

$$R^2 = 1 - \frac{\text{IBS}_T}{\text{IBS}_{T_M}},$$

where IBS_{T_M} is the integrated Brier score evaluated from the T_M , which has a root node only. Based on the IBS_T , R^2 , number of terminal nodes, and the proportion of covariates which are used in the final tree model, we evaluate the each approaches.

5.3.3 Results

The simulation results about two approaches are shown in Table 5.1. The values in table are average of each measure obtained through simulations. In each case of simulations, the combination approach shows the good results than the traditional approach about the values of IBS_T , R^2 and the proportion of covariates which are used in the final tree model. Especially, about the covariates selected in the final model, the approach considering the combination of covariates does not choose the nuisance covariate in almost cases. When censoring rate is increased, the traditional approach make it difficult to detect the Z_1 and Z_2 which are critical variables in the true model. This is thought to be the cause of leading the negative results of explained residual variation when the censoring rate is 50%.

As a natural outcome, the combination approach of covariates tends to construct the small size tree compared to the traditional approach, since more than one covariates are considered in each node at a time. In our simulation case, the optimal sizes of tree are 4 and 3 in the traditional approach and the combination approach, respectively. If censoring is not occurred, both approaches have almost correct number of terminal nodes. However, if censoring rate be increased, the traditional approach rapidly decreases the number of nodes. As a result, we conclude that the proposed approach gives a more suitable model than the traditional approach when the true model includes the XOR problem.

Table 5.1: The results obtained by simulations.

covariate	sensor rate	approach	IBS_T	R^2	$ \tilde{T} $	prop. Z_1	prop. Z_2	prop. Z_3
quantitative	0%	tra.	17.3	3.1	4.3	1.4	1.3	0.6
		com.	15.2	14.8	3.2	2.2	2.1	0.1
	25%	tra.	16.1	0.6	3.3	0.9	0.9	0.4
		com.	14.3	11.8	3.0	2.0	2.0	0.0
	50%	tra.	13.8	-0.8	2.5	0.5	0.6	0.4
		com.	12.9	6.0	2.6	1.6	1.6	0.0
binary	0%	tra.	17.1	9.6	3.7	1.2	1.2	0.3
		com.	16.1	14.8	3.1	2.1	2.0	0.0
	25%	tra.	17.5	0.3	3.0	0.7	0.5	0.7
		com.	15.3	12.7	3.0	2.0	2.0	0.0
	50%	tra.	14.8	-0.9	2.0	0.4	0.2	0.4
		com.	14.1	3.6	2.1	1.1	1.1	0.0

IBS_T : the integrated brier score evaluated from the selected tree $\times 100$, R^2 : the explained residual variation $\times 100$, $|\tilde{T}|$: the number of terminal nodes about selected tree, prop.: the proportion of the covariates which are used in the final tree model. tra.: traditional approach. com.: combination approach of covariates.

5.4 Example

We show the application of the combination approach using leukemia patients bone marrow transplantation data. This data was collected from 1984 to 1989 at one of four hospitals. The data are composed of 137 patients. The observation period is defined from the data of transplant surgery to the date of relapse, death or last survival verified. The observation is considered as an event occurred if it was determined by death or relapse. The 54 patients are censored (the censoring rate is about 39%). We used five covariates: Z_1 is the indicator of Acute Myelocytic Leukemia (AML) low risk group or not, Z_2 is the indicator of AML high risk group or not, Z_3 is the indicator of French-American-British Classification (FAB) grade is 4 or 5 and the disease group is AML or not, Z_4 is the donor age, and Z_5 is the patient age. The patients which are not included in AML group are included in Acute Lymphoblastic Leukemia (ALL) group. The details of this research are given in Copelan et al. (1991). The data used in this example are available from the Web site offered by Klein and Moeschberger (2003).

We set 10 minimum number of events in nodes as the stop condition of splitting step. The number of cross validation is set to 10. The tree obtained by proposed approach is shown in Figure 5.2. The circle and square shapes in the figure represent the internal and terminal nodes, respectively. The values in the shapes represent the number of samples included in the node, and the values in parentheses represent the number of events. The tree has two splitting points and three terminal nodes ($t_1 - t_3$). As the result, the patients are divided

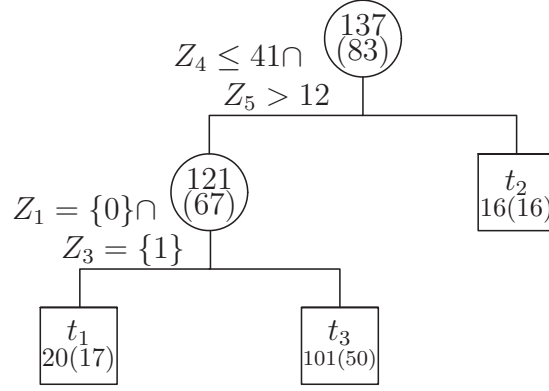


Figure 5.2: Tree-structured model constructed from bone marrow transplant patients data.

to three groups:

- group t_1 : $\{ 41 \text{ or younger donor } \cap 12 \text{ or older patient}$
 $\cap \text{ AML high risk or ALL } \cap \text{ FAB grade is 4 or 5} \}$,
- group t_2 : $\{ 41 \text{ or older donor } \cup 12 \text{ or younger patient } \}$,
- group t_3 : $\{ 41 \text{ or younger donor } \cap 12 \text{ or older patient}$
 $\cap \{ \text{ AML low risk } \cup \text{ FAB grade is 1, 2 or 3 } \} \}$.

The graphical representation of the covariate space separated by the obtained tree-structured model is shown in Figure 5.3. Although the model obtained by the combination approach seems difficult to understand than the classical approach, each terminal node is disjoint in covariate space. The Kaplan-Meier survival curves for each group are shown in Figure 5.4. From the survival curves, we can understand that the group t_1 has the highest risk of death or recurrence and the group t_3 has the lowest risk of it. The group t_2 has the risk of death or recurrence between t_1 and t_3 . The Kaplan-Meier survival curves are well separated from each other, and we conclude that the combination approach gives the reasonable result.

5.5 Conclusion

In this chapter, we consider the approach, which considers two or more combinations of covariates for dividing a node, for constructing a survival tree based on CART algorithm. As the motivation of this approach, we have considered that it would be more suitable than the traditional approach under some situations like XOR problem.

Through the simulation study, we have been shown the performance of these approaches. As the result, the combination approach is considered that has a

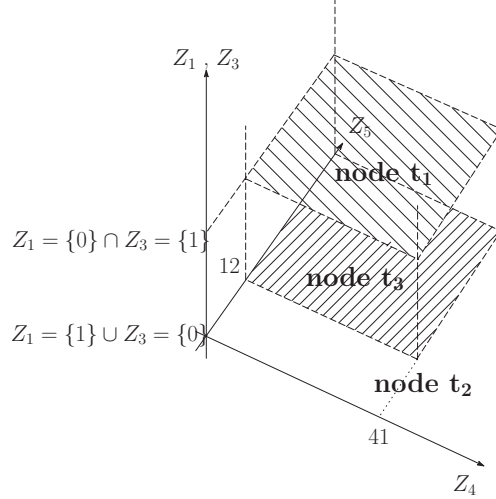


Figure 5.3: The covariate space represented by tree-structured model in Figure 5.2.

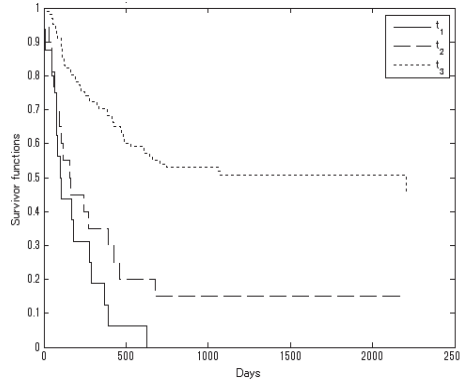


Figure 5.4: The Kaplan-Meier survival curves for the terminal nodes in Figure 5.2.

potential to construct a more suitable tree-structured model than the traditional approach in some situations. The utility of the proposed method has been shown by using an actual medical data. The tree constructed from the data divided the patients to three groups. From the Kaplan-Meier survival curves of each group, we conclude that the obtained results are considered to be reasonable.

Tree-structured model has an advantage that the relationship between the covariates and hazards is easy to show. Moreover, there is another advantage that is easy to insert a new data to the model. The studied approach has the possibility to construct an optimal model, while holding this advantages. However, from the aspect of "fragmentation" (Friedman and Fisher (1999)), the proposed approach has a risk of building a low performance model than

the model obtained by traditional approach. That is, in a covering algorithm the low number of splitting step becomes a cause of a high value of bias and variance in the model. Therefore, we consider that further studies are needed in order to address this problem. As an another disadvantage, it requires a long learning time, but we consider that computing technology in recent years may be able to resolve this problem.

Chapter 6

Regression Tree on Interval-valued Symbolic Data

6.1 Introduction

In this chapter, we propose a new approach for constructing a regression tree based on interval-valued variables, which are introduced in Chapter 1. As discussed in the chapter, the analysis based on symbolic data is considered appropriate in some scenarios. In the tree-structured model for regression analysis, a classification tree based on symbolic data was proposed by Mballo and Diday (2005). In their research, response and explanatory variables were assumed as classical categorical and interval-valued variables, respectively. To split concepts in an arbitrary node, they used the Kolmogorov-Smirnov criterion based on the concepts that are ordered based on either upper, lower, mean, or span length of the interval. By using example data, their proposed criterion was compared with the classical criterion of Gini. In Quantin et al. (2011), the classification tree was constructed based on modal multi-valued variables to determine whether hospital pathways were considered in the selection of prognostic factors of one-year survival after acute myocardial infarction. In their research, the response variable was treated as a modal binary variable, and the Gini index was used in the splitting and pruning step in the symbolic classification and regression tree (CART) algorithm.

In our proposed model, both response and explanatory variables can be assumed as interval-valued variables. This is assumed the case that we want to additional analysis based on the data which are grouped by already some variables. For example, consider the case that we have patients that are grouped by medical rationale as age, sex and some drug is used or not. If we are interested in identifying the other variables like weight and blood glucose level that can explain the cholesterol level of patients who followed the same group, then the response and explanatory variables are represented as interval-valued variables.

Our proposed model is different from the other models, because in our model,

a concept can be included in several terminal nodes in a tree. That is, when splitting an arbitrary parent node, each concept can be divided and included in two child nodes. For constructing this model, several problems need to be considered. These problems include the representation method of predictive models in each node, the evaluation criteria for splitting concepts, the method for searching an optimal splitting point in interval values, and the method to predict the response value of a new datum based on the tree-structured model. In this study, we address these problems and present an application of this model in refer to the study of HIV-1-infected patients data.

The remainder of this chapter is organized as follows. The proposed model for constructing regression trees on interval-valued variables and the method to overcome the problems in constructing the model are shown in Section 6.2. The results of simple simulation studies and example of the application of the proposed model are described in Section 6.3 and 6.4, respectively. Finally, we conclude this chapter in Section 6.5.

6.2 Methods

Based on p interval-valued explanatory variables Z_1, Z_2, \dots, Z_p and one class response variable, Mballo and Diday (2005) constructed decision trees using CART as follows. In the splitting step, the concepts included in a terminal node are ordered based on either upper, lower, mean, or span length of the interval at first. This will enable us to treat interval values of concepts in the node as classical values. Next, the optimal splitting rule " $Z_j \leq s$?" are defined using the criterion of Gini or Kolmogorov-Smirnov. Since the interval-valued explanatory variables can be treated as classical variables by ordering, all the other steps (pruning and selection steps) in order to build the decision tree are same as classical case. Although their proposed method is constructing a classification tree, regression trees can be constructed as same.

Here, the observed learning sample is assumed as $\mathcal{L} = \{(x(\omega_u), \mathbf{z}(\omega_u)); u = 1, 2, \dots, m\}$, where $x(\omega_u) = [a_u, b_u]$ and $\mathbf{z}(\omega_u) = \{(z_1(\omega_u), z_2(\omega_u), \dots, z_p(\omega_u)); z_j(\omega_u) = [c_{ju}, d_{ju}], j = 1, \dots, p\}$ is a realization of interval-valued variables. To construct the proposed model, several problems have to be considered. First, we have to determine the predictive model for each node in the tree. Next, the calculation method for evaluating each split is required. Further, when an interval-valued variable is used as a covariate, we need to determine the searching points for dividing the data included in terminal node for the splitting step. Finally, it is necessary to determine how to predict the response value of new data based on the tree-structured model. In the following sections, we propose the methods that are used to address these problems.

6.2.1 Predictive model in each node

When a tree is constructed on the data that has classical variables only, as described in the Chapter 3, the prediction in each node is given by only one value which is estimated by sample mean of the data in the node as (3.3). When $x(\omega_u)$ is a realization of interval-valued variables, there are several possibilities of representing the predictive model in each node. In this section, we propose two models.

1. Model using the midpoint and range of $x(\omega_u)$

In order to contain the interval information of $x(\omega_u) = [a_u, b_u]$, we can fit two models on the midpoint $x^c(\omega_u) = (a_u + b_u)/2$ and the range $x^r(\omega_u) = (b_u - a_u)$ of the interval. When these variables are used, we can consider two methods for constructing a regression tree. One method is to construct two regression trees, T^c and T^r , for $x^c(\omega_u)$ and $x^r(\omega_u)$, respectively. In this case, we consider the response values as the classical data and construct each regression tree as described in Section 3.2.1. Let the interval included in node t on Z_j be $I_{Z_j t}$ ($j = 1, \dots, p$). Then, the observed frequency of concept ω_u included in t is given by

$$\begin{aligned} f_t(u) &= \frac{\| \mathbf{z}(\omega_u) \cap t \|}{\| \mathbf{z}(\omega_u) \|}, \\ &= \frac{\| z_1(\omega_u) \cap I_{Z_1 t} \|}{\| z_1(\omega_u) \|} \times \dots \times \frac{\| z_p(\omega_u) \cap I_{Z_p t} \|}{\| z_p(\omega_u) \|}, \end{aligned}$$

from (1.8). The total observed frequency included in node t is given by

$$f_t = \sum_{u=1}^m f_t(u).$$

As (3.2), we define the mean of the residual sum of squares of the tree for $x^c(\omega_u)$ as

$$\begin{aligned} R(T^c) &= \frac{1}{m} \sum_{u=1}^m (x^c(\omega_u) - T^c(\mathbf{z}_u))^2 \\ &= \frac{1}{m} \sum_{t \in \tilde{T}} \sum_{u=1}^m f_t(u) (x^c(\omega_u) - x^c(t))^2, \end{aligned} \quad (6.1)$$

where $x^c(t)$ is the constant predictor of x^c for node t . As the same, the $R(T^r)$ for $x^r(\omega_u)$ is defined as

$$\begin{aligned} R(T^r) &= \frac{1}{m} \sum_{u=1}^m (x^r(\omega_u) - T^r(\mathbf{z}_u))^2 \\ &= \frac{1}{m} \sum_{t \in \tilde{T}} \sum_{u=1}^m f_t(u) (x^r(\omega_u) - x^r(t))^2, \end{aligned} \quad (6.2)$$

where $x^r(t)$ is the constant predictor of x^r for node t . The conclusion can be considered based on both trees, and it is possible to predict the lower and upper bound of the interval value of the response variable. For example, consider a case that a new datum reaches the terminal node t^c and t^r of trees T^c and T^r , respectively. When the estimate value of response in t^c and t^r are 1 and 4, respectively, we can predict that the lower and upper bound of the data are -1 and 3, respectively.

The other method to construct a regression tree that minimizes the sum of the residual sum of squares for $x^c(\omega_u)$ and $x^r(\omega_u)$, individually. That is, $R(T)$ is defined as

$$\begin{aligned} R(T) &= \frac{1}{m} \sum_{u=1}^m \{w_c(x^c(\omega_u) - T_1(\mathbf{z}_u))^2 + w_r(x^r(\omega_u) - T_2(\mathbf{z}_u))^2\} \\ &= \frac{1}{m} \sum_{t \in \tilde{T}} \sum_{u=1}^m f_t(u) \{w_c(x^c(\omega_u) - x^c(t))^2 \\ &\quad + w_r(x^r(\omega_u) - x^r(t))^2\}, \end{aligned} \quad (6.3)$$

where $x^c(t)$ and $x^r(t)$ are the constant predictors of $x(t)$ about the midpoint and the range for the response variable. Although T^c and T^r , which are defined above have different tree structures, T_1 and T_2 have the same tree structure. The variables w_c and w_r are weight parameters, which are determined previously. Since the former method constructs two tree-structured models and the latter method constructs only one tree-structured model, it is considered that the former method is more effective. However, the former method has a disadvantage that the resulting model becomes complex and the interpretation is difficult.

2. Methods using the minimum and maximum value of $x(\omega_u)$

Instead of using the midpoint and range of $x(\omega_u)$, we can consider using its minimum value $x^l(\omega_u) = a_u$ and its maximum value $x^u(\omega_u) = b_u$. In this case, the tree is constructed in the same manner as the above method, except for the response values, $x^c(\omega_u)$ and $x^r(\omega_u)$, which are changed to $x^l(\omega_u)$ and $x^u(\omega_u)$, respectively. The advantage of using $x^l(\omega_u)$ and $x^u(\omega_u)$ instead of $x^c(\omega_u)$ and $x^r(\omega_u)$ is as follows. When we construct a tree-structured model based on $x^c(\omega_u)$ and $x^r(\omega_u)$ using equation (6.3), we have to determine the w_c and w_r previously. On the other hand, if we construct a model based on $x^l(\omega_u)$ and $x^u(\omega_u)$, we can consider them to

be equality as natural:

$$\begin{aligned}
R(T) &= \frac{1}{m} \sum_{u=1}^m \{(x^l(\omega_u) - T_1(\mathbf{z}_u))^2 + (x^u(\omega_u) - T_2(\mathbf{z}_u))^2\} \\
&= \frac{1}{m} \sum_{t \in \tilde{T}} \sum_{u=1}^m f_t(u) \{(x^l(\omega_u) - x^l(t))^2 \\
&\quad + (x^u(\omega_u) - x^u(t))^2\}, \tag{6.4}
\end{aligned}$$

where $x^l(t)$ and $x^u(t)$ are the constant predictors of $x(t)$ about the minimum and the maximum values for the response variable.

6.2.2 Splitting criteria

When we construct a tree that minimizes $R(T)$ defined as (6.1), (6.2), (6.3), or (6.4), the estimator of $x(t)$ is given by the sample mean of the data in node t as

$$\bar{x}_t^* = \frac{1}{f_t} \sum_{u=1}^m f_t(u) x^*(\omega_u), \tag{6.5}$$

where $x^*(\omega_u)$ is defined as $x^c(\omega_u)$, $x^r(\omega_u)$, $x^l(\omega_u)$ or $x^u(\omega_u)$ in response to $R(T)$. Then, the minimum mean of the residual sum of squares of t for (6.1) or (6.2) is given by

$$R(t) = \frac{1}{m} \sum_{u=1}^m f_t(u) (x^*(\omega_u) - \bar{x}_t^*)^2. \tag{6.6}$$

Similarly, the minimum mean of the residual sum of squares of t for (6.3) or (6.4) is given by

$$R(t) = \frac{1}{m} \sum_{u=1}^m w_c f_t(u) (x^c(\omega_u) - \bar{x}_t^c)^2 + \frac{1}{m} \sum_{u=1}^m w_r f_t(u) (x^r(\omega_u) - \bar{x}_t^r)^2 \tag{6.7}$$

or

$$R(t) = \frac{1}{m} \sum_{u=1}^m f_t(u) (x^l(\omega_u) - \bar{x}_t^l)^2 + \frac{1}{m} \sum_{u=1}^m f_t(u) (x^u(\omega_u) - \bar{x}_t^u)^2, \tag{6.8}$$

respectively. Using (6.6), (6.7), or (6.8), we seek the best split in all possible splits of t , which satisfies (3.6).

When we construct a regression tree based on $R(T)$ which is given by using (6.6), (6.7), or (6.8), the variation of within variable is not considered. As another possibility of constructing regression trees based on symbolic data, we can

construct a tree that minimizes the symbolic total sum of squares. As described in (1.7), we define the empirical density function of the random variable X for node t by using the observed frequencies as

$$g_t(\xi) = \frac{1}{f_t} \sum_{u \in A_\xi} f_t(u) \frac{1}{b_u - a_u},$$

where A_ξ is the set of labels of the concepts $\{u : \xi \in x(\omega_u)\}$. Then the symbolic sample mean and variance is given by

$$\bar{x}_t = \frac{1}{f_t} \sum_{u=1}^m f_t(u) \frac{(b_u + a_u)}{2}, \quad (6.9)$$

and

$$V_t = \frac{1}{3f_t} \sum_{u=1}^m f_t(u) [(a_u - \bar{x}_t)^2 + (a_u - \bar{x}_t)(b_u - \bar{x}_t) + (b_u - \bar{x}_t)^2],$$

respectively.

By using (1.9), the symbolic total sum of squares of t is defined as

$$\text{TSS}(t) = f_t V_t. \quad (6.10)$$

Same as with the case of $R(t)$, the best split s_t^* is defined as a split that satisfies the following.

$$\max_{s_t \in S_t} \{\text{TSS}(t) - (\text{TSS}(t_L) + \text{TSS}(t_R))\}. \quad (6.11)$$

Since the symbolic total sum of squares includes both the within and between sum of squares as described in (1.9), the regression tree constructed using $\text{TSS}(t)$ can include the information of variation within variables.

6.2.3 Optimal splitting point

In the splitting step, all possible splits S_t are evaluated in each terminal node and the data are divided into two child nodes. Let $z_{j(1)}, z_{j(2)}, \dots, z_{j(m_t)}$ be covariates that are arranged in the ascending order about Z_j in an arbitrary node t . If Z_j is a classical variable, the value of \bar{x}_t , which is defined by (3.3), does not change in the interval between $z_{j(u)}$ and $z_{j(u+1)}$, and therefore, the value of $R(t)$ defined by (3.5) does not change in that interval ($u = 1, \dots, m_t - 1$). Thus, the obvious choice of splitting point s for the interval between $z_{j(u)}$ and $z_{j(u+1)}$ is at the midpoint of their. On the other hand, when Z_j is an interval-valued variable, the value of \bar{x}_t that is defined by (6.5) or (6.9) is changed in the interval in which the concept (or concepts) is segmentalized by s , and therefore, the value of $R(t)$ or $\text{TSS}(t)$, which is defined by (6.6), (6.7), or (6.10) is changed in the interval.

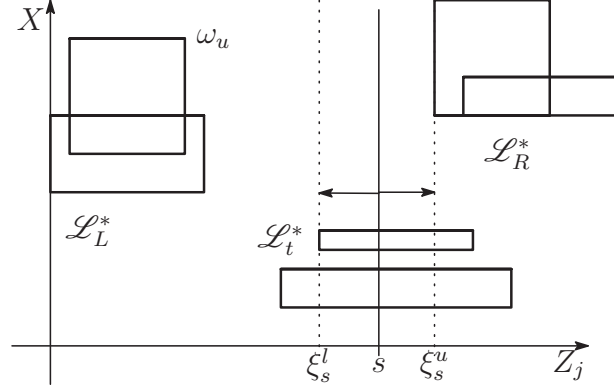


Figure 6.1: An example of the interval I_s which the member of \mathcal{L}_t^* is not changed by s .

Of course, we cannot search all splitting points in that interval. Therefore, we need some criterion for searching the points that may satisfy (3.6) or (6.11). In this section, we propose the method that determine the points required to evaluate $R(t)$ or $\text{TSS}(t)$ as follows.

Now, we consider the division of the node t . Let $\mathcal{L}_t = \{(x(\omega_u), \mathbf{z}(\omega_u)); u = 1, 2, \dots, m_t\}$, where $x(\omega_u) = [a_u, b_u]$ and $\mathbf{z}(\omega_u) = (z_1(\omega_u), \dots, z_p(\omega_u))$, where $z_j(\omega_u) = [c_{ju}^*, d_{ju}^*]$, $j = 1, 2, \dots, p$, be the learning samples included in node t . The variables c_{ju}^* and d_{ju}^* denote the minimum and maximum values of concept ω_u , which is included in the node t about Z_j , respectively. Let $\mathcal{L}_t^* = \{(x(\omega_u), \mathbf{z}(\omega_u)); u = 1, 2, \dots, m_t^* (m_t^* \leq m_t)\}$ be the set of samples in node t , which are segmentalized by splitting point s . Let $I_s = [\xi_s^l, \xi_s^u]$ be the interval such that the member of \mathcal{L}_t^* is not changed by s . An graphical example of that interval I_s is shown in Figure 6.1.

The points required to evaluate $R(t)$ defined by (6.6) or $\text{TSS}(t)$ defined by (6.10) for I_s are

$$\begin{cases} \frac{(\xi_s^l + \xi_s^u)}{2} & \text{in the interval that } \mathcal{L}_t^* = \emptyset, \\ \xi_s^l, \xi_s^u, s_{\text{stat}} & \text{in the interval that } \mathcal{L}_t^* \neq \emptyset, \end{cases} \quad (6.12)$$

where s_{stat} will be evaluated only if it is included in I_s , and given as

$$s_{\text{stat}} = \frac{-cf - de + \frac{2}{a}bcd}{a(f - e) + b(c - d)}, \quad (6.13)$$

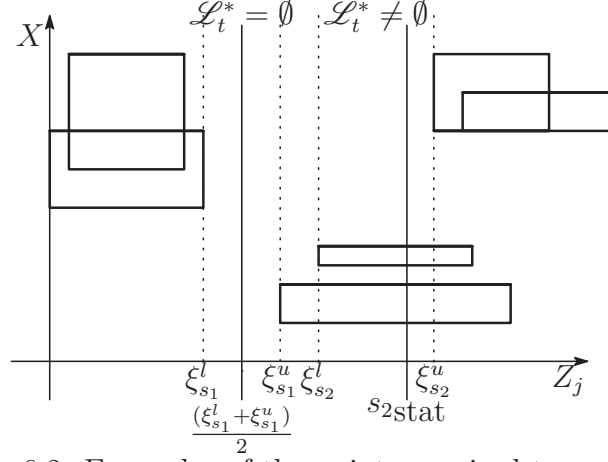


Figure 6.2: Examples of the points required to evaluate.

where

$$\begin{aligned}
a &\equiv \sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*}, \\
b &\equiv \sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} x^*(\omega_u), \\
c &\equiv m_L - \sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} c_{ju}^*, \\
d &\equiv m_R + \sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} d_{ju}^*, \\
e &\equiv \sum_{u=1}^{m_L} f_t(u) x^*(\omega_u) - \sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} c_{ju}^* x^*(\omega_u), \\
f &\equiv \sum_{u=1}^{m_R} f_t(u) x^*(\omega_u) + \sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} d_{ju}^* x^*(\omega_u),
\end{aligned}$$

where $\mathcal{L}_L^* = \{(x(\omega_u), z(\omega_u)); u = 1, \dots, m_L\}$ and $\mathcal{L}_R^* = \{(x(\omega_u), z(\omega_u)); u = 1, \dots, m_R\}$ are the set of concepts in node t , which are not segmentalized by splitting point s and lower or upper than s , respectively. The graphical examples of the points which required to evaluate by this definition are shown in Figure 6.2. $x^*(\omega_u)$ is defined as $x^c(\omega_u)$, $x^r(\omega_u)$, $x^l(\omega_u)$, or $x^u(\omega_u)$ with respect to $R(T)$. When TSS(t) is considered, $x^*(\omega_u)$ is defined as $a_u + b_u$.

See appendix for the derivation of the stationary point s_{stat} . When $R(t)$ is defined by (6.7), search points can be calculated as the simple expansion of (6.12). We evaluate the points indicated by (6.12) for all intervals for which the

members of \mathcal{L}_t^* , \mathcal{L}_L^* , and \mathcal{L}_R^* , are different, and we determine the best splitting point for node t .

6.2.4 Prediction of a new data

When new data that have symbolic covariate variables are obtained, and we want to predict their response using the regression tree constructed by learning samples; then, we need to determine the method used for allocating data to the tree. If the new data have only classical values, the data will reach only one terminal node in the tree. Therefore, the allocation of the predicted value to the new data is the estimated value in that node. However, if the new data have symbolic covariate variables, they have the possibility to reach some terminal nodes in our model. For example, if the new data ω_u^{new} have the range of age between 20 to 30 as a covariate and if a regression tree has only one split that has the criterion of age ≤ 23 , the data will reach the left and right terminal node with observed frequency $f_{t_L}(u) = 0.3$ and $f_{t_R}(u) = 0.7$, respectively.

The natural allocation method for new data in our model is based on the weighted mean of the estimated values in the terminal nodes:

$$\sum_{t \in T} f_t(u^{\text{new}}) \bar{x}_t^*, \quad (6.14)$$

where $f_t(u^{\text{new}})$ is the observed frequency of concept ω_u^{new} for a node t and \bar{x}_t^* is defined as (6.5) or (6.9).

6.2.5 Algorithm

The algorithm for constructing a symbolic regression tree by the proposed method is described as follows:

Algorithm6.1.

1. Determine the representation of the predictive model in each node as described in Section 6.2.1.
2. Determine whether to use RSS or TSS in splitting step.
3. All observed learning sample \mathcal{L} is set to the current node.
4. The current node t : $\mathcal{L}_t = \{(y(\omega_u), \mathbf{z}(\omega_u)); u = 1, \dots, m_t\}$.
5. **For** $j \leftarrow 1$ **to** the number of covariates p **do**
6. **For** $i \leftarrow 1$ **to** the number of possible splits defined by (6.13) **do**
7. Devide the \mathcal{L}_t into two child nodes based on splitting point $Z_j \leq s_i$.
8. Calculate RSS using (6.6) or (6.7) (or TSS using (6.10)) for two child nodes.

9. Determine the optimal splitting point $Z_j^* \leq s_i^*$ which satisfies (3.6) (or (6.11)), and divide the \mathcal{L}_t into two child nodes t_L and t_R .
10. **If** there is a node or nodes that do not satisfy the stop condition, **then** define one of those as the current node t and go to step 4.
11. Obtain the nested subtrees by using the cost-complexity measure (3.8) as in the classical case.
12. Determine the optimal tree size using test sample or V -fold cross validation method (validate data will be allocated by (6.14)).

6.3 Simulation

We present simple simulation studies to compare the predictive model and splitting criteria described in previous section. Our purpose in these simulations is to study the properties of the each predictive model and splitting criteria.

6.3.1 Model and Setting

We used the mixed data generated from the models of both child nodes and divided them by each method. The covariates $Z_j = [c_j, d_j]$ are generated as follows ($j = 1, 2$):

$$\begin{aligned} c_j &= m_j - \frac{r_j}{2}, \\ d_j &= m_j + \frac{r_j}{2}, \end{aligned}$$

where m_j and r_j are random values created from a discrete uniform distribution with $\{0/50, 1/50, \dots, 50/50\}$. Then, the response value $X = [a, b]$ is generated as

$$\begin{aligned} a &= m - \frac{r}{2}, \\ b &= m + \frac{r}{2}, \end{aligned}$$

where the midpoint m and the range r of interval are generated from the following models:

$$\begin{aligned} m - \text{Model 1} &: n(1, 1), \\ m - \text{Model 2} &: f_L(u)n(1, 1) + f_R(u)n(2, 1), \end{aligned}$$

and

$$\begin{aligned} r - \text{Model 1} &: u[0, 1], \\ r - \text{Model 2} &: f_L(u)u[0, 1] + f_R(u)u[0, 2]. \end{aligned}$$

$n(\mu, \sigma^2)$ and $u[\alpha, \beta]$ are random values created from a normal distribution with parameters μ and σ^2 , and uniform distribution with parameters α and β , respectively. The true observed frequency of concept ω_u in the left child node is defined as follows:

$$f_L(u) = \begin{cases} 1 & (d_{1u} \leq 0.5) \\ 0 & (0.5 < c_{1u}) \\ \frac{0.5 - c_{1u}}{d_{1u} - c_{1u}} & (\text{others}) \end{cases}.$$

That is, all models assume that the covariate z_1 is used in the true node model and its true threshold is set as 0.5.

We compare the six methods of predictive models and splitting criteria. The first method splits the data based on the residual $R(T^c)$ defined as (6.1). The second method uses the residual $R(T^r)$ defined as (6.2). The third and fourth methods split the data based on the residual $R(T)$ defined as (6.3) where $w_c = w_r = 1$ and $w_c = 2, w_r = 1$, respectively. The fifth method uses the residual $R(T)$ defined as (6.4). The final method uses the symbolic total sum of squares $\text{TSS}(t)$ defined as (6.10). The number of samples is set to 100, and simulations are repeated 300 times.

6.3.2 Results

In Tables 6.1, 6.2, 6.3, and 6.4, we show the results of six methods for four patterns of models. In Table 6.1, we show the results in the case that m and r -Models are 1. In this case, the true tree-structured model has no splitting points in the model. Not surprisingly, each methods select the covariates Z_1 and Z_2 with approximately half and half probabilities. As the same, the obtained predictions in each child nodes take close values in each other.

In Table 6.2, the data are generated from the m -Model 2 and r -Model 1. Both child nodes in the true tree structured model have different midpoints and the same range of the response. The method using $R(T^r)$ only shows a poor result as expected. The other methods show approximately the same results. The predicted values of midpoints (or minimum and maximum) in each child node come closer to each other than the true values. This reason is considered to arise from the partial data overlap beyond the true threshold.

In Table 6.3, the data are generated from the m -Model 1 and r -Model 2. The child nodes in the true tree-structured model have the same midpoints and different ranges of the response. It should be understood that the method using $R(T^c)$ only shows a poor result, and the method using $R(T^r)$ only shows the best result. The method using $\text{TSS}(T)$ shows a similar result to $R(T^c)$ only. Moreover, the method using x^l and x^u shows a poor result. The reason of this is considered that the method using minimum and maximum values of x can not reflect the interval information of x like the method using x^r .

Table 6.1: Comparison of six methods for splitting in the case of m -Model 1 and r -Model 1. The true values of midpoint, range, minimum and maximum of x in each child node are as follows: $x_L^c = x_R^c = 1$, $x_L^r = x_R^r = 0.5$, $x_L^l = x_R^l = 0.75$, $x_L^u = x_R^u = 1.25$.

Methods	pro. Z_1	mea. split	var. split	\bar{x}_L^c	\bar{x}_L^r	\bar{x}_R^c	\bar{x}_R^r
$R(T^c)$	0.51	0.52	0.06	0.97	0.50	1.01	0.50
$R(T^r)$	0.50	0.48	0.05	0.99	0.50	1.02	0.50
$R(T)$	0.53	0.52	0.05	0.97	0.50	1.01	0.50
$(w_c = w_r = 1)$							
$R(T)$	0.52	0.53	0.05	0.98	0.50	1.01	0.50
$(w_c = 2, w_r = 1)$							
				\bar{x}_L^l	\bar{x}_L^u	\bar{x}_R^l	\bar{x}_R^u
$R(T)$	0.52	0.53	0.05	0.73	1.22	0.77	1.26
(using x^l and x^u)							
				\bar{x}_L^c		\bar{x}_R^c	
TSS(T)	0.51	0.52	0.06	0.97		1.01	

pro. Z_1 : the probability of selecting the true covariate z_1 for splitting, mea. split: the mean of threshold value that is estimated for splitting the covariate when the true covariate Z_1 is selected, var. split: the variance of threshold value that is estimated for splitting the covariate when the true covariate Z_1 is selected. \bar{x}_L^c and \bar{x}_R^c : the sample mean of the midpoint of x in two child nodes. \bar{x}_L^r and \bar{x}_R^r : the sample mean of the range of x in two child nodes. \bar{x}_L^l and \bar{x}_R^l : the sample mean of the minimum value of x in two child nodes. \bar{x}_L^u and \bar{x}_R^u : the sample mean of the maximum value of x in two child nodes.

In Table 6.4, we show the results in the case that m and r -Models are 2. In this case, the both child nodes in the true tree structured model have different midpoints and ranges of the response. All methods show good results, and especially the method using $R(T)$ with $w_c = w_r = 1$ shows the best result. As the case in Table 6.2, the predicted values of midpoints (or minimum and maximum) in each child node come closer to each other than the true values.

Throughout the simulation studies, the result using TSS(T) shows approximately the same result as $R(T^c)$ only. Moreover, when the method using minimum and maximum values of the response symbolic data is used, its interval information has only a small impact on the result. The predicted values of each node are affected worse by data that are over the true threshold as same as in the classical data case.

6.4 Example

We present a simple application of the regression tree on interval-valued variables in refer to the study of HIV-1-infected patients (Nishijima et al. (2014)).

Table 6.2: Comparison of six methods for splitting in the case of m -Model 2 and r -Model 1. The true values of midpoint, range, minimum and maximum of x in each child node are as follows: $x_L^c = 1$, $x_R^c = 2$, $x_L^r = x_R^r = 0.5$, $x_L^l = 0.75$, $x_L^u = 1.25$, $x_R^l = 1.75$, $x_R^u = 2.25$.

Methods	pro. Z_1	mea. split	var. split	\bar{x}_L^c	\bar{x}_L^r	\bar{x}_R^c	\bar{x}_R^r
$R(T^c)$	1.00	0.50	0.01	1.14	0.50	1.87	0.50
$R(T^r)$	0.46	0.53	0.06	1.23	0.50	1.83	0.51
$R(T)$	1.00	0.50	0.01	1.14	0.50	1.87	0.50
$(w_c = w_r = 1)$							
$R(T)$	1.00	0.50	0.01	1.14	0.50	1.87	0.50
$(w_c = 2, w_r = 1)$							
				\bar{x}_L^l	\bar{x}_L^u	\bar{x}_R^l	\bar{x}_R^u
$R(T)$	1.00	0.50	0.01	0.89	1.39	1.61	2.12
(using x^l and x^u)							
				\bar{x}_L^c	\bar{x}_R^c		
TSS(T)	1.00	0.50	0.01	1.14	1.87		

See Table 6.1.

These data were collected to study the effect of an active drug use on renal function. As the indicator of the effect of the active drug, the decrement in estimated glomerular filtration rate (eGER) was used in their study, and we use this indicator as response variable X . The data we used consist of 619 individual patients. Each individual has several potential risk factors for renal dysfunction, and six binary variables were used in Nishijima and colleagues' research: active or reference drug, ritonavir-boosted protease inhibitors, hypertension, dyslipidemia, diabetes mellitus, concurrent use of nephrotoxic drugs. By aggregating the patients which have the same binary variables, we constructed 28 concepts (although there are $2^6 = 64$ combinations of covariates, 36 combinations of the data do not include the patients). As other baseline characteristics of patients, each patient has 6 continuous variables: age, CD4 count, HIV RNA viral load (VL), weight, BMI, serum creatinine (SCR). These variables are treated as interval-valued variables $Z_1 - Z_6$, and used to construct a tree structured model. A part of the concepts is listed in Table 6.5.

Two regression trees are constructed based on the following settings. One tree is constructed using midpoint and range values of $x(\omega_u)$ as the predictive model in each node, and the mean of the residual sum of squares $R(T)$ is defined in the form of (6.3), where $w_c = w_r = 1$. The other tree is constructed using the minimum and maximum values of $x(\omega_u)$ as the predictive model. As the stop condition for splitting, the minimum number of observed frequencies in nodes is set to three. In the selection step, we use 10-fold cross validation.

Table 6.3: Comparison of six methods for splitting in the case of m -Model 1 and r -Model 2. The true values of midpoint, range, minimum and maximum of x in each child node are as follows: $x_L^c = x_R^c = 1$, $x_L^r = 0.5$, $x_R^r = 1$, $x_L^l = 0.75$, $x_L^u = 1.25$, $x_R^l = 0.5$, $x_R^u = 1.5$.

Methods	pro. Z_1	mea. split	var. split	\bar{x}_L^c	\bar{x}_L^r	\bar{x}_R^c	\bar{x}_R^r
$R(T^c)$	0.45	0.52	0.06	0.99	0.61	1.03	0.90
$R(T^r)$	1.00	0.52	0.01	0.99	0.58	1.01	0.94
$R(T)$	0.96	0.53	0.01	0.99	0.59	1.02	0.94
$(w_c = w_r = 1)$							
$R(T)$	0.82	0.51	0.02	0.99	0.59	1.02	0.93
$(w_c = 2, w_r = 1)$							
				\bar{x}_L^l	\bar{x}_L^u	\bar{x}_R^l	\bar{x}_R^u
$R(T)$	0.67	0.51	0.03	0.69	1.28	0.58	1.50
(using x^l and x^u)							
				\bar{x}_L^c	\bar{x}_R^c		
TSS(T)	0.45	0.52	0.06	0.99	1.03		

See Table 6.1.

The tree structure obtained by the method using the midpoint and range values of $x(\omega_u)$ is shown in Figure 6.3. The circles and squares in the figure represent the internal and terminal nodes, respectively. The values in the shapes represent the number of observed frequencies in the node. The covariate used in the tree was Z_3 , which is the HIV RNA viral load. The estimated midpoint and range values of the decrement in eGER using this tree structure were

$$\hat{X}^c = \begin{cases} -11.1 & (Z_3 \leq 4.6 \times 10^5) \\ -33.8 & (Z_3 > 4.6 \times 10^5) \end{cases},$$

and

$$\hat{X}^r = \begin{cases} 34.5 & (Z_3 \leq 4.6 \times 10^5) \\ 112.0 & (Z_3 > 4.6 \times 10^5) \end{cases},$$

respectively.

The tree constructed using the minimum and maximum values of $x(\omega_u)$ is shown in Figure 6.4. The splitting points of root node in the tree are same as the tree in Figure 6.3. The tree has three splitting points and four terminal nodes ($t_1 - t_4$). The estimated minimum and maximum values of the decrement in eGER using this tree structure were

$$[\hat{X}^l, \hat{X}^u] = \begin{cases} [-54.6, 9.2] & (Z_3 \leq 4.6 \times 10^5 \cap Z_6 \leq 0.66) \\ [-18.0, 4.9] & (Z_3 \leq 1.3 \times 10^5 \cap Z_6 > 0.66) \\ [-40.5, 7.8] & (1.3 \times 10^5 < Z_3 \leq 4.6 \times 10^5 \cap Z_6 > 0.66) \\ [-90.0, 22.4] & (Z_3 > 4.6 \times 10^5) \end{cases}.$$

Table 6.4: Comparison of six methods for splitting in the case of m -Model 2 and r -Model 2. The true values of midpoint, range, minimum and maximum of x in each child node are as follows: $x_L^c = 1$, $x_R^c = 2$, $x_L^r = 0.5$, $x_R^r = 1$, $x_L^l = 0.75$, $x_L^u = 1.25$, $x_R^l = 1.5$, $x_R^u = 2.5$.

Methods	pro. Z_1	mea. split	var. split	\bar{x}_L^c	\bar{x}_L^r	\bar{x}_R^c	\bar{x}_R^r
$R(T^c)$	1.00	0.49	0.01	1.15	0.59	1.86	0.92
$R(T^r)$	1.00	0.52	0.01	1.19	0.59	1.85	0.94
$R(T)$	1.00	0.50	0.01	1.15	0.59	1.86	0.92
$(w_c = w_r = 1)$							
$R(T)$	1.00	0.49	0.01	1.15	0.59	1.85	0.92
$(w_c = 2, w_r = 1)$							
				\bar{x}_L^l	\bar{x}_L^u	\bar{x}_R^l	\bar{x}_R^u
$R(T)$	1.00	0.49	0.01	0.85	1.44	1.40	2.32
(using x^l and x^u)							
				\bar{x}_L^c		\bar{x}_R^c	
TSS(T)	1.00	0.49	0.01	1.15		1.86	

See Table 6.1.

Table 6.5: The concepts constructed from the patients' data.

Patient	eGER	age	weight	...	SCR
ω_u	X	Z_1	Z_2	...	Z_6
ω_1	$[-91.4, 17.4]$	$[23, 74]$	$[7, 308]$...	$[0.4, 1.02]$
ω_2	$[-150, 22]$	$[21, 63]$	$[2, 415]$...	$[0.29, 1.06]$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
ω_{28}	$[-19.5, 8.7]$	$[26, 33]$	$[43, 143]$...	$[0.57, 0.68]$

From these results, if we are interested in analysis based on concepts which are grouped by basic risk factors for renal dysfunctions rather than each patient, the most important factor is considered to be the HIV RNA viral load. The low HIV RNA viral load groups have low risk of the average decrement in eGER. On the other hand, high HIV RNA viral load groups have high risk of the average decrement and variability in eGER. Moreover, in the low HIV RNA viral load groups, the concepts which have high value of the SCR have lower risk of the average decrement in eGER than the other groups.

For comparison, we show the result of the model obtained by the classical CART approach in Figure 6.5. The covariate used in the tree is Z_6 only, and the model has four splitting points and five terminal nodes ($t_1 - t_5$). The estimated

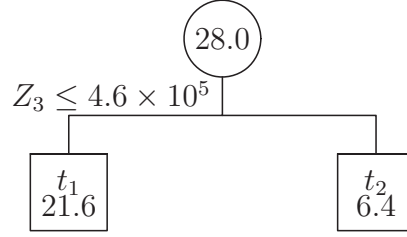


Figure 6.3: Tree constructed from the patients' data using the midpoint and range values of interval $x(\omega_u)$.

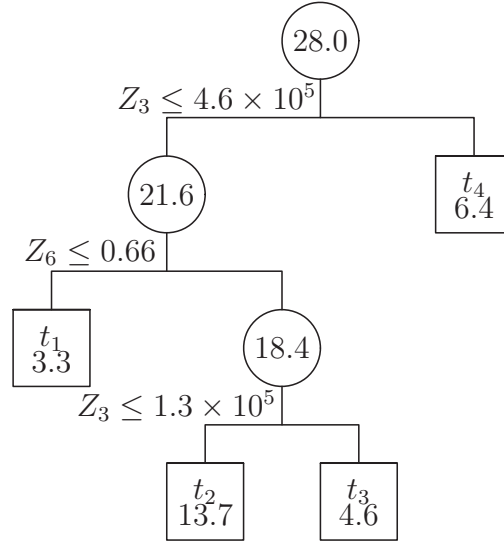


Figure 6.4: Tree constructed from the patients' data using the minimum and maximum values of interval $x(\omega_u)$.

mean values of the decrement in eGER were

$$\bar{X} = \begin{cases} -98.5 & (Z_6 < 0.47) \\ -25.8 & (0.47 \leq Z_6 < 0.58) \\ -12.4 & (0.58 \leq Z_6 < 0.71) \\ -5.8 & (0.71 \leq Z_6 < 0.86) \\ 1.4 & (0.86 \leq Z_6) \end{cases} .$$

Intriguingly, if we analyse the data based on each patient, the most important factor is considered to be the SCR. Especially, the patients who have lower value of SCR have high risk of the decrement in eGER.

The difference between the results obtained by the classical and proposed approaches is considered as follows. In the classical regression tree, we determined which covariates could discriminate patients according to their baseline

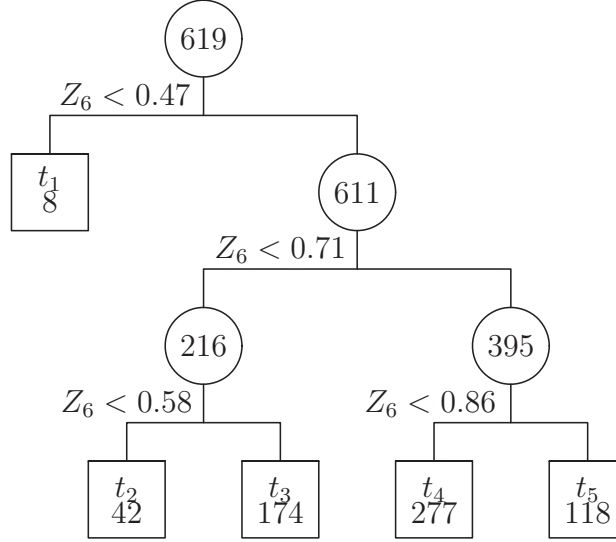


Figure 6.5: Tree constructed from the patients' data using classical approach.

characteristics whatever their groups that have been classified by the medical rationale. In the symbolic tree, we are interested in identifying the variables that can explain the decrement in eGER of patients who fall in the same group which is determined by the medical rationale. Then, the symbolic analysis has considered that group itself becomes an important variable. Therefore, it is considered that this difference of assumptions has led to difference in the results. Although the interpretation of the effect of covariates on decrementation is not straightforward like the classical approach, symbolic analysis enables us to do an analysis based on groups of interest.

The interpretation of the model obtained by the proposed approach is slightly different from a tree structured model obtained by the classical approach. In the classical approach, each observation is assigned to one terminal node, and the prediction of a new datum is given from a terminal node which the datum is included. Therefore, the splitting rule in each internal node ($Z_j < c$) exists to assign a predicted value to the observation. On the other hand, in the proposed approach, an observation (concept) can be included in several terminal nodes, and the predicted value of the observation is constructed by a weighted sum of all predicted values of terminal nodes with observed frequencies. In this case, the splitting rule in each internal node is used to assign a weight to the observation.

6.5 Conclusion

In this chapter, we proposed a new method to construct a regression tree based on interval-valued variables. In some scenarios, using symbolic data representation is considered more suitable than using classical variables. For example, when each sample is constructed based on species instead of individuals, it is obvious to consider symbolic data.

Our proposed method is considerably different from the existing methods on classical variables, because in our method individual data can be included in several terminal nodes in a tree. Therefore, we need to address several problems for constructing such a tree-structured model. These problems include the representation of predictive models in each node, and the decision rule of searching points for splitting the node data. In Section 6.2, we proposed several methods to address these problems.

As a simple example of the application of regression trees on interval-valued variables, we constructed regression trees of 36 concepts aggregated from 619 patients in the refer to the study of HIV-1-infected patients. From the comparison to the analysis based on the classical regression tree approach, we can obtain another aspect of the data by the symbolic approach. That is, if we consider the analysis based on the group which is determined by the medical rationale rather than the individual, the variable that can explain the response is not the same as the variable that is determined by the classical approach. Although these analyses need to consider more complex aspects than the classical variable case, we expect that these analyses give us a opportunity to understand a new interpretation of data.

Appendix

We use the notation as defined in Section 6.2.3 and earlier. If we want to search for the minimum point of $R(t_L) + R(t_R)$, where $R(t)$ is defined as (6.6), in interval $I_s = [\xi_s^l, \xi_s^u]$, we have to search ξ_s^l and ξ_s^u , and the stationary point s_{stat} if it is included in I_s . From (6.5) and (6.6), $R(t)$ is represented as

$$R(t) = \frac{1}{m} \left\{ \sum_{u=1}^{m_t} f_t(u) x^*(\omega_u)^2 - \frac{(\sum_{u=1}^{m_t} f_t(u) x^*(\omega_u))^2}{f_t} \right\}. \quad (6.15)$$

Using (6.15), $R(t_L) + R(t_R)$ is given by

$$\begin{aligned} R(t_L) + R(t_R) = & \frac{1}{m} \left\{ \sum_{u=1}^{m_t} f_t(u) x^*(\omega_u)^2 \right. \\ & \left. - \frac{(\sum_{u=1}^{m_t} f_{t_L}(u) x^*(\omega_u))^2}{f_{t_L}} - \frac{(\sum_{u=1}^{m_t} f_{t_R}(u) x^*(\omega_u))^2}{f_{t_R}} \right\}. \end{aligned}$$

In interval I_s , $R(t_L) + R(t_R)$ can be rewritten as

$$\begin{aligned}
R(t_L) + R(t_R) = & \frac{1}{m} \left\{ \sum_{u=1}^{m_t} f_t(u) x^*(\omega_u)^2 \right. \\
& - \frac{(\sum_{u=1}^{m_L} f_{t_L}(u) x^*(\omega_u) + \sum_{u=1}^{m_t^*} f_{t_L}(u) x^*(\omega_u))^2}{m_L + \sum_{u=1}^{m_t^*} f_{t_L}(u)} \\
& \left. - \frac{(\sum_{u=1}^{m_R} f_{t_R}(u) x^*(\omega_u) + \sum_{u=1}^{m_t^*} f_{t_R}(u) x^*(\omega_u))^2}{m_R + \sum_{u=1}^{m_t^*} f_{t_R}(u)} \right\}. \quad (6.16)
\end{aligned}$$

For concepts ω_u in \mathcal{L}_t^* , $f_{t_L}(u)$ and $f_{t_R}(u)$ can be represented as

$$f_{t_L}(u) = \frac{s - c_{ju}^*}{d_{ju}^* - c_{ju}^*} f_t(u) \quad (6.17)$$

and

$$f_{t_R}(u) = \frac{d_{ju}^* - s}{d_{ju}^* - c_{ju}^*} f_t(u), \quad (6.18)$$

respectively. Now, the region R_t , the value of c_{ju}^* , d_{ju}^* , and $f_t(u)$ is fixed. Moreover, the value of m_L and m_R are fixed in the interval I_s . By substituting (6.17) and (6.18) in (6.16) and differentiating it with respect to s , we get

$$\begin{aligned}
\frac{\partial}{\partial s} \{R(t_L) + R(t_R)\} = & \frac{1}{m} \left\{ -2\bar{x}_L^*(s) \left(\sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} x^*(\omega_u) \right) + \bar{x}_L^{*2}(s) \left(\sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} \right) \right. \\
& \left. + 2\bar{x}_R^*(s) \left(\sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} x^*(\omega_u) \right) - \bar{x}_R^{*2}(s) \left(\sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} \right) \right\},
\end{aligned}$$

where

$$\bar{x}_L^*(s) = \frac{\sum_{u=1}^{m_L} f_{t_L}(u) x^*(\omega_u) + \sum_{u=1}^{m_t^*} \frac{s - c_{ju}^*}{d_{ju}^* - c_{ju}^*} f_t(u) x^*(\omega_u)}{m_L + \sum_{u=1}^{m_t^*} \frac{s - c_{ju}^*}{d_{ju}^* - c_{ju}^*} f_t(u)}$$

and

$$\bar{x}_R^*(s) = \frac{\sum_{u=1}^{m_R} f_{t_R}(u) x^*(\omega_u) + \sum_{u=1}^{m_t^*} \frac{d_{ju}^* - s}{d_{ju}^* - c_{ju}^*} f_t(u) x^*(\omega_u)}{m_R + \sum_{u=1}^{m_t^*} \frac{d_{ju}^* - s}{d_{ju}^* - c_{ju}^*} f_t(u)}.$$

Setting this partial derivative equal to 0, we get

$$\bar{x}_L^*(s) + \bar{x}_R^*(s) = 2 \frac{\left(\sum_{u=1}^{m_t^*} \frac{f_t(u) x^*(\omega_u)}{d_{ju}^* - c_{ju}^*} \right)}{\left(\sum_{u=1}^{m_t^*} \frac{f_t(u)}{d_{ju}^* - c_{ju}^*} \right)}.$$

By solving this equation, we get the s_{stat} as described in (6.13).

If we want to search the minimum point of $\text{TSS}(t_L) + \text{TSS}(t_R)$, we can start with

$$\begin{aligned} \text{TSS}(t) = \sum_{u=1}^{m_t} f_t(u) & \left\{ \frac{1}{f_t^2} \left(\sum_{u=1}^{m_t} f_t(u) \frac{a_u + b_u}{2} \right)^2 \right. \\ & \left. - (a_u + b_u) \frac{1}{f_t} \sum_{u=1}^{m_t} f_t(u) \frac{a_u + b_u}{2} + \frac{1}{3} (a_u^2 + a_u b_u + b_u^2) \right\} \end{aligned}$$

instead of (6.15), and the following calculation is the same as that in the case of $R(t)$.

Chapter 7

Concluding Remarks

In this thesis, we took a statistical approach to analyzing medical data. In particular, we focused on data that had a large quantity of covariates, like medical diagnostic images. To construct a model that could predict a new patient's disease malignancy, prognosis, or survival time, we studied supervised learning methods that were based on the genetic algorithm (GA) and the classification and regression tree (CART) algorithm.

In Chapter 2, we introduced the GA and several processing methods for medical images. In the preprocessing step, we used filtering and binary methods to extract the region of interest (ROI) from the images. In the feature extraction step, we used gray-level co-occurrence matrix (GLCM) and wavelet transformations to account for texture features. We proposed a new approach for the feature selection and classification steps that was inspired by ensemble methods and combined the GA and k -nearest neighbor (k -nn) methods. From the comparative analysis of lung cancer patients' CT images, we have concluded that the proposed approach yields more stable results than the traditional one.

In Chapter 3, we introduced the basic methods for constructing tree-structured models with CART algorithms; in particular, we focused on regression and survival trees. A regression tree can be effectively constructed using the cost-complexity measure proposed by Breiman et al. If we wish to construct a survival tree based on the degree of separation between nodes, though, then Leblanc and Crowley's split-complexity measure should be used.

In Chapter 4, we provided an example of a survival tree based on medical images. We constructed the model with data on patients with brain metastases from breast cancer. We employed covariates that are commonly used in medical research, like age and Karnofsky Performance Scale (KPS), as well as the texture features, as covariates in the model. Based on the results of this model, we conclude that the texture features obtained from MRIs may be highly useful in determining cancer patients' prognoses.

In chapter 5, we explored a splitting rule that was not restricted to a single covariate in order to deal with some of the issues posed by tree-structured

models. As a representative example of the tree-structured model's inherent problems, we examined the exclusive OR (XOR) problem. Based on simulation studies, we have concluded that survival trees perform more effectively when the combination of covariates is accounted for in the splitting rule. We demonstrated the utility of this approach by applying it to leukemia patients' bone marrow transplantation data.

In Chapter 6, we proposed a new approach for constructing a regression tree that was based on interval-valued variables. A key element of this approach is that each concept can be included in several terminal nodes of the tree. Through simulation studies and the application of referenced data from HIV-1-infected patients, we showed that this approach may highlight new aspects of the data that were largely ignored in the classical approach.

References

- Addison, P. S. (2002). *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance*. Institute of Physics Publishing, London.
- Agresti, A. (2002). *Categorical Data Analysis: Second Edition*. Wiley, New York.
- Arita, H., Narita, Y., Miyakita, Y., Ohono, M., Sumi, M. and Shibui, S. (2014). Risk factors for early death after surgery in patients with brain metastases: reevaluation of the indications for and role of surgery. *Journal of Neuro-Oncology*, **116**, 145-152.
- Bertrand, P. and Goupil, F. (2000). Descriptive statistics for symbolic data. In: *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag, Berlin, 106-124.
- Billard, L. and Diday, E. (2000). Regression analysis for interval-valued data. *Proceedings of the Seventh Conference of the International Federation of Classification Societies: Data Analysis, Classification, and Related Methods*, Springer, 369-374.
- Billard, L. and Diday, E. (2002). Symbolic regression analysis. *Proceedings of the Eighth Conference of the International Federation of Classification Societies: Classification, Clustering, and Data Analysis*, Springer, 281-288.
- Billard, L. and Diday, E. (2006). *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. Wiley, Chichester.
- Billard, L. (2007). Dependencies and variation components of symbolic interval-valued data. In: *Selected Contributions in Data Analysis and Classification*, Springer-Verlag, Berlin, 3-12.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag, New York.
- Bock, H. H. and Diday, E. (2000). *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag, Berlin.
- Bou-Hamad, I., Larocque, D. and Ben-Ameur, H. (2011). A review of survival trees. *Statistics Surveys*, **5**, 44-71.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC, London.

Ciaccio, E. J., Dunn, S. M. and Akay, M. (1993). Biosignal pattern recognition and interpretation systems: part 1: fundamental concepts. *IEEE Engineering in Medicine and Biology*, **12**, 89-95.

Ciaccio, E. J., Dunn, S. M. and Akay, M. (1993). Biosignal pattern recognition and interpretation systems: part 2: methods for feature extraction and selection. *IEEE Engineering in Medicine and Biology*, **12**, 106-113.

Ciampi, A., Chang, C. H., Hogg, S. and McKinney, S. (1987). Recursive partition: a versatile method for exploratory data analysis in biostatistics. *Biostatistics*, **38**, 23-50.

Ciampi, A., Hogg, S. A., Mckinney, S. and Thiffault, J. (1988). RECPAM : A computer program for recursive partition and amalgamation for censored survival data and other situations frequently occurring in biostatistics: I. Methods and Program Features. *Computer Methods and Programs in Biomedicine*, **26**, 239-256.

Ciampi, A., Thiffault, J. and Sagman, U. (1989). RECPAM : A Computer program for recursive partition and amalgamation for censored survival data and other situations frequently occurring in biostatistics. II. applications to data on small cell carcinoma of the lung (SCCL). *Computer Methods and Programs in Biomedicine*, **30**, 283-296.

Collett, D. (2003). *Modelling Survival Data in Medical Research: Second Edition*. Chapman & Hall/CRC, London.

Copelan, E. A., Biggs, J. C., Thompson, J. M., Crilley, P., Szer, J., Klein, J. P., Kapoor, N., Avalos, B. R., Cunningham, I., Atkinson, K., Downs, K., Harmon, G. S., Daly, M. B., Brodsky, I., Bulova, S. I. and Tutschka, P. J. (1991). Treatment for acute myelocytic leukemia with allogeneic bone marrow transplantation following preparation with BuCy2. *Blood*, **78**, 838-843.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B*, **34**, 187-220.

Cunningham, P. and Delany, S. J. (2007). k-Nearest neighbour classifiers. *Technical Report*, UCD-CSI-2007-4.

- Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, **41**, 909-996.
- Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Davis, R. B. and Anderson, J. R. (1989). Exponential survival trees. *Statistics in Medicine*, **8**, 947-961.
- De Carvalho, F. A. T., Lima Neto, E. A., and Tenorio, C. P. (2004). A new method to fit a linear regression model for interval-valued data. *Proceedings of the Twenty Seventh Germany Conference on Artificial Intelligence: Advances in Artificial Intelligence*, Springer-Verlag, 295-306.
- Dhawan, A. P., Chitre, Y., Kaiser-Bonasso C. and Moskowitz M. (1996). Analysis of mammographic microcalcifications using gray-level image structure features. *IEEE Transactions on Medical Imaging*, **15**, 246-259.
- Diday, E. and Noirhomme-Fraiture, M. (2008). *Symbolic Data Analysis and the SODAS software*. Wiley, Chichester.
- Efron, B. and Tibhirani R. J. (1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC, New York.
- Fox, J. (2008). *Applied Regression Analysis and Generalized Linear Models: Second Edition*. Sage, California.
- Friedman, J. H. and Fisher, N. I. (1999). Bump hunting in high-dimensional data. *Statistics and Computing*, **9**, 123-143.
- Gaspar, L., Scott, C., Rotman, M., Asbell, S., Phillips, T., Wasserman, T., McKenna, W. G. and Byhardt, R. (1997). Recursive partitioning analysis (RPA) of prognostic factors in three Radiation Therapy Oncology Group (RTOG) brain metastases trials. *International Journal of Radiation Oncology, Biology, Physics*, **37**, 745-751.
- Goldberg, D. E. (1989). *Genetic Algorithms: in Search, Optimization, and Machine Learning*. addison-wesley, Boston.
- Gordon, L. and Olshen, R. A. (1985). Tree-structured survival analysis. *Cancer Treatment Reports*, **69**, 1065-1069.
- Graf E., Schmoor, C., Sauerbrei, W. and Schumacher, M. (1999). Assessment

and comparisons of prognostic classification schemes for survival data. *Statistics in Medicine*, **18**, 2529-2545.

Haralick, R. M., Shanmugam, K. and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, **3**, 610-621.

Hastie, T. and Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 607-616.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data mining, Inference, and Prediction, Second Edition*. Springer, New York.

Hess, K. R., Abbruzzese, M. C., Lenzi, R., Raber, M. N. and Abbruzzese, J. L. (1999). Classification and regression tree analysis of 1000 consecutive patients with unknown primary carcinoma. *Clinical Cancer Research*, **5**, 3403-3410.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Horst, K. H. and Heinz, O. P. (2000). The skull stripping problem in MRI solved by a single 3D watershed transform. *Lecture Notes in Computer Science*, **1935**, 134-143.

Hosmer, D. W. and Lemeshow, S. (1999). *Applied Survival Analysis: Regression Modeling of Time to Event Data*. Wiley, New York.

Keles, S. and Segal, M. R. (2002). Residual-based tree-structured survival analysis. *Statistics in Medicine*. **21**, 313-326.

Klein, J. P. and Moeschberger, M. L. (2003). *Survival Analysis: Techniques for Censored and Truncated Data: Second Edition*. Springer, New York.

Lachin, J. M. (1981). Introduction to sample size determination and power analysis for clinical trials. *Controlled Clinical Trials*, **2**, 93-113.

Laine, A. and Fan J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1186-1191.

Lawless, F. J. (2003). *Statistical Models and Methods for Lifetime Data: Second*

Edition. Wiley, New York.

Leblanc, M. and Crowley, J. (1992). Relative risk trees for censored survival data. *Biometrics*, **48**, 411-425.

Leblanc, M. and Crowley, J. (1993). Survival trees by goodness of split. *Journal of the American Statistical Association*, **88**, 457-467.

Leblanc, M. and Crowley, J. (1995). A review of tree-based prognostic models. *Journal of Cancer Treatment and Research*, **75**, 113-124.

Lee, E. T. and Wang, J. W. (2003). *Statistical Methods for Survival Data Analysis: Third Edition*. Wiley, New York.

Le-Rademacher, J. and Billard, L. (2012). Symbolic covariance principal component analysis and visualization for interval-valued data. *Journal of Computational and Graphical Statistics*, **21**, 413-432.

Li, X., Jin, H., Lu, Y., Oh, J., Chang, S. and Nelson, S. J. (2004). Identification of MRI and ¹H MRSI parameters that may predict survival for patients with malignant gliomas. *NMR in Biomedicine*, **17**, 10-20.

Lima Neto, E. A., De Carvalho, F. A. T. and Tenorio, C. P. (2004). Univariate and multivariate linear regression methods to predict interval-valued features. *Proceedings of the Seventeenth Australian joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, Springer-Verlag, 526-537.

Lima Neto, E. A., De Carvalho, F. A. T. and Freire, E. S. (2005). Applying constrained linear regression models to predict interval-valued data. *Proceedings of the Twenty eighth Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence*, Springer-Verlag, 92-106.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 674-693.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London: Series B: Biological Sciences*, **207**, 187-217.

Mballo, C. and Diday, E. (2005). Decision trees on interval valued variables. *The Electronic Journal of Symbolic Data Analysis*, **3**, 8-18.

Mayer-Bäse, A. (2004). *Pattern Recognition for Medical Imaging*. Elsevier

Academic Press, San Diego.

Miller, M. T., Jerebko, A. K., Malley, J. D. and Summers, R. M. (2003). Feature selection for computer-aided polyp detection using genetic algorithms: medical Imaging: physiology and function: methods systems, and applications. *Proceedings of SPIE*, **5031**, 102-110.

Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, **58**, 415-434.

Narita, Y. and Shibui, S. (2009). Strategy of surgery and radiation therapy for brain metastases. *International Journal of Clinical Oncology*, **14**, 275-280.

Nishijima, T., Kawasaki, Y., Tanaka, N., Mizushima, D., Aoki, T., Watanabe, K., Kinai, E., Honda, H., Yazaki, H., Tanuma, J., Tsukada, K., Teruya, K., Kikuchi, Y., Gatanaga, H. and Oka, S. (2014). Long-term exposure to tenofovir continuously decrease renal function in HIV-1-infected patients with low body weight: results from 10 years of follow-up. *AIDS*, **28**, 1903-1910.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, **11**, 169-198.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**, 62-66.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, **6**, 21-45.

Punch, W. F., Goodman, E. D., Pei, M., Chia-Shun, L., Hovland, P. and Enbody, R. (1993). Further research on feature selection and classification using genetic algorithms. *Proceedings of Fifth International Conference on Genetic Algorithms*, 557-564.

Quantin, C., Billard, L., Touati, M., Andreu, N., Cottin, Y., Zeller, M., Afonso, F., Battaglia, G., Seck, D., Le, Teuff G. and Diday, E. (2011). Classification and regression trees on aggregate data modeling: an application in acute myocardial infarction. *Journal of Probability and Statistics*, **2011**, 1-19.

Radespiel-Tröger, M., Rabenstein, T., Schneider, H. T. and Lausen, B. (2003). Comparison of tree-based methods for prognostic stratification of survival data. *Artificial Intelligence in Medicine*, **28**, 323-341.

- Radespiel-Tröger, M., Gefeller, O., Rabenstein, T. and Hothorn, T. (2006). Association between split selection instability and predictive error in survival trees. *Methods of information in medicine*, **45**, 548-556.
- Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A. and Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, **4**, 164-171.
- Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, **33**, 1-39.
- Sanz-Requena, R., Revert-Ventura, A., Martí-Bonmatí, L., Alverich-Bayarri, Á. and García-Martí, G. (2013). Quantitative MR perfusion parameters related to survival time in high-grade gliomas. *European Radiology*, **23**, 3456-3465.
- Schoenfeld, D. A. and Richter, J. R. (1982). Nomograms for calculating the number of patients needed for a clinical trial with survival as an endpoint. *Biometrics*, **38**, 163-170.
- Segal, M. R. (1988). Regression trees for censored data. *Biometrics*, **44**, 35-47.
- Shimokawa, A., and Miyaoka, E. (2012). Application of genetic algorithm for classification of medical images. *Advances and Applications in Statistics*, **29**, 1-31.
- Shimokawa, A., Kawasaki, Y. and Miyaoka, E. (2014). An approach for constructing survival tree based on combination of covariates. *SUT Journal of Mathematics*, to appear.
- Shimokawa, A., Kawasaki, Y. and Miyaoka, E. (2014). Construction of regression trees on interval-valued symbolic variables. *Journal of the Japanese Society of Computational Statistics*, **27**, 61-79.
- Shimokawa, A., Narita, Y., Shibui, S. and Miyaoka, E. (2014). Application of survival tree based on texture features obtained through MRI of patients with brain metastases from breast cancer. *International Journal of Statistics in Medical Research*, **3**, 340-347.
- Siedlecki, W. and Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recogn. Lett*, **10**, 335-347.
- Sperduto, P. W., Kased, N., Roberge, D., Xu, Z., Shanley, R., Luo, X., Sneed, P. K., Chao, S. T., Weil, R. J., Suh, J., Bhatt, A., Jensen, A. W., Brown, P. D.,

- Shih, H. A., Kirkpatrick, J., Gaspar, L. E., Fiveash, J. B., Chiang, V., Knisely, J. P. S., Sperduto, C. M., Lin, N. and Mehta, M. (2012). Summary report on the graded prognostic assessment: an accurate and facile diagnosis-specific tool to estimate survival for patients with brain metastases. *Journal of Clinical Oncology*, **30**, 419-425.
- Therneau, T. M., Grambsch, P. M. and Fleming, T. R. (1990). Martingale-based residual for survival models, *Biometrika*, **77**, 147-160.
- Vasanth, M., Bharathi, V. S. and Dhamodharan, R. (2010). Medical image feature, extraction, selection and classification. *International Journal of Engineering Science Technology*, **2**, 2071-2076.
- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, **4**, 65-85.
- Walter, G. G. and Shen, X. (2001). *Wavelets and Other Orthogonal Systems: Second Edition*. Chapman & Hall/CRC, Boca Raton.
- Wu, C. C., Lee, W. L., Chen, Y. and Hsieh, K. S. (2009). A GA-based multiresolution feature selection for ultrasonic liver tissue characterization. *Proceedings of Fourth International Conference on Innovative Computing, Information and Control*, 1542-1545.
- Zhang, H. P. (1995). Splitting criteria in survival trees. *In Statistical Modelling: Proceedings of the 10th International Workshop on Statistical Modeling*, 305-314.