Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations Graduate School

2008

Variable Selection in Competing Risks Using the L1-Penalized Cox Model

XiangRong Kong
Virginia Commonwealth University

School of Medicine
Virginia Commonwealth University

This is to certify that the dissertation prepared by Xiangrong Kong entitled “Variable Selection in Competing Risks Using The L1 Penalized Cox Model” has been approved by her committee as satisfactory completion of the dissertation requirement for the degree of Doctor of Philosophy.

Kellie J. Archer, Ph.D., Director of Dissertation

Robert A. Fisher, M.D., School of Medicine

Chris Gennings, Ph.D., School of Medicine

Charles Kish, Ph.D., Wyeth Consumer Health Care

Viswanathan Ramakrishnan, Ph.D., School of Medicine

Shumei S. Sun, Ph.D., Department Chair

Jerome F. Strauss, M.D., Ph.D., Dean, School of Medicine

F. Douglas Boudinot, Ph.D, Dean of the School of Graduate Studies

September 22, 2008

Variable Selection in Competing Risks Using The L1 Penalized Cox Model

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by Xiangrong Kong

Bachelor of Science in Computational Mathematics, 1997
Beijing University of Technology, Beijing, China

Director: Kellie J. Archer, Ph.D.
Assistant Professor
Department of Biostatistics

Virginia Commonwealth University
Richmond, Virginia
December 2008

Acknowledgements

I would like to thank all my committee members: Dr. Archer, Dr. Fisher, Dr. Gennings, Dr. Kish and Dr. Ramesh. I always feel very honored for their agreeing to serve on my committee. All of their insightful comments and questions during the committee meetings made the meeting time also a great learning time for me. My special thanks to my advisor Dr. Archer, who has been very supportive during all my training. I just feel short of words to express how many thanks I owe her. Her way of conducting the academic work will definitely be influential (p-value=0) on my future career path.

I also appreciate my family's support and my husband's tolerance. Their company makes life much easier in a place far away from my home town.

Table of Contents

List of Tables

List of Figures

Abstract

VARIABLE SELECTION IN COMPETING RISKS USING THE L1 PENALIZED COX MODEL

By Xiangrong Kong, Ph.D.

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2008

Major Director: Kellie J. Archer, Ph.D.
Assistant Professor
Department of Biostatistics

One situation in survival analysis is that the failure of an individual can happen because of one of multiple distinct causes. Survival data generated in this scenario are commonly referred to as competing risks data. One of the major tasks, when examining survival data, is to assess the dependence of survival time on explanatory variables. In competing risks, as with ordinary univariate survival data, there may be explanatory variables associated with the risks raised from the different causes being studied. The same variable might have different degrees of influence on the risks due to different causes. Given a set of explanatory variables, it is of interest to identify the subset of variables that are significantly associated with the risk corresponding to each failure cause. In this project, we develop a statistical methodology to achieve this purpose, that is, to perform variable selection in the presence of competing risks survival data. Asymptotic properties of the model and empirical simulation results for evaluation of the model performance are provided. One important feature of our method, which is based on the idea of the L_1 penalized Cox model, is the ability to perform variable selection in situations where we have high-dimensional explanatory variables, i.e. the number of explanatory variables is larger than the number of observations. The method was applied on a real dataset originated from the National Institutes of Health funded project "Genes related to hepatocellular carcinoma progression in living donor and deceased donor liver transplant" to identify genes that might be relevant to tumor progression in hepatitis C virus (HCV) infected patients diagnosed with hepatocellular carcinoma (HCC). The gene expression was measured on Affymetrix GeneChip microarrays. Based on the current available 46 samples, 42 genes show very strong association with tumor progression and deserve to be further investigated for their clinical implications in prognosis of progression on patients diagnosed with HCV and HCC.

Chapter 1

Introduction to Survival Analysis

Survival analysis is a field in statistics that specifically deals with the modeling and analysis of survival data: time from a well-defined time origin until the occurrence of some event or end point of interest. In medical research, the time origin may be the date of diagnosis of a disease of interest, or the date of an individual being recruited to take certain treatment regimen, or among others. The end point of interest can thus be the date of death of a patient, and the resulting data are known as failure time data. The end point of interest may correspond to situations other than death, such as the date of progression or recurrence of a disease, or the date of recovery of the patient (Collett, 2003). The methodologies presented in survival analysis can also be applied in modeling data generated in other fields of science. In economics, the "survival" time can be the time of unemployment of an unemployed person. In industrial applications, the "survival" time can be the lifetime of a unit or some component in a unit, and here survival analysis is termed reliability analysis (Hougaard, 2000).

The distinguishing feature of survival data is that survival times are frequently censored, and therefore special methods are required when analyzing survival data. Often in survival studies, in addition to observed survival time, some explanatory variables may be observed. Such explanatory variables typically describe pre-existing heterogeneity in the samples under study (Kalbfleisch and Prentice, 2002), and one of the major tasks when modeling survival data, which is the focus of this project, is to assess the dependence of survival time on explanatory variables.

In this chapter, we start with Section 1.1 by introducing the mathematical notations and statistical framework that are commonly used to describe survival problems. Different modeling techniques, including non-parametric, parametric, and semi-parametric models, have been presented in the literature on survival analysis. They are briefly reviewed in Section 1.2 to 1.4, with emphasis on the semi-parametric Cox proportional hazards model in Section 1.4. These models are generally used in the analysis of univarariate survival data where independence between survival times is assumed. In Section 1.5, we introduce multivariate survival data, including the competing risks scenario, which is the main topic of this work.

1.1 Survival Data

Let T be a nonnegative random variable representing the survival time of an individual from a population. T can be either discrete or continuous. In this project, we focus on the more common situation where T is continuous.

1.1.1 Survival Time Distribution

The probabilistic aspect of T can be described in many ways, three of which are particularly used in survival analysis: the survivor function, the probability density function, and the hazard function (Kalbfleisch and Prentice, 2002).

Survivor Function

The survivor function $S(t)$ is defined by the probability that T exceeds a value t, that is,

$$S(t) = P(T > t), \ 0 < t < \infty \, .$$

Thus

$$F(t) = 1 - S(t) \, , \tag{1.1}$$

is the commonly seen cumulative distribution function (CDF) of T.

Probability Density Function

We know that the probability density function (PDF) $f(t)$ for T is defined as:

$$f(t) = \frac{dF(t)}{dt} = \frac{-dS(t)}{dt} \ .$$

Hazard Function

The hazard function $h(t)$ is defined as:

$$h(t) = \lim_{\delta t \to 0} \frac{P(t \leq T \leq t + \delta t | T > t)}{\delta t} \ . \tag{1.2}$$

It is the instantaneous rate at which failures occur for individuals who have survived up to time t. As the survivor function and the probability density function, the hazard function itself fully specifies the distribution of T.

The following relationships between the survivor function, the PDF and the hazard function can be derived using their definitions (Kalbfleisch and Prentice, 2002). From Equation 1.2,

$$\begin{aligned} h(t) &= \frac{f(t)}{S(t)} \\ &= \frac{-d \log F(t)}{dt} \ . \end{aligned}$$

Integrating with respect to t and using $F(0) = 1$, we obtain

$$\begin{aligned} S(t) &= \exp\left[-\int_0^t h(s)ds\right] \\ &= \exp\left[-H(t)\right] , \end{aligned} \tag{1.3}$$

where $H(t) = \int_0^t h(s)ds$ is called the **cumulative hazard function**. The PDF of T can be obtained by differentiating Equation 1.3:

$$f(t) = h(t) \exp\left[-H(t)\right] .$$

1.1.2 Time Origins and Censoring

In considering survival time data, it is important to have a clear and unambiguous definition of the time origin from which survival is measured. For instance, if the time of interest represents age, the time origin thus is date of birth of the individual. In other instances, the natural time origin may be the date of occurrence of some event, such as the date of diagnosis of a particular disease. Similarly, the event or end point of interest should be clearly defined as well. For example, in a trial comparing treatments of heart disease, we might take previous documented occurrence of heart attack as providing eligibility for study. The time origin can be the date of admission and randomization to the study, and the event may correspond to the date of recurrence of a heart attack. The clinical medical conditions corresponding to the event should be carefully specified. Clear identification of an origin and an end point are crucial in survival analysis (Kalbfleisch and Prentice, 2002).

The survival time of an individual is said to be censored when the end point of interest is not observed for that individual. This may occur when the data from a study are analyzed at a point in time when some individuals are still alive. Alternatively, the survival status of an individual may be lost to follow-up. For example, suppose that after being recruited to a clinical trial, a patient moves to another place, and can no longer be traced. The only information available on the survival experience of that patient is the last date on which he or she was known to be alive. This date may well be the last time that the patient reported to a clinic for a regular check-up. In each situation, the observed time of an individual is less than the actual, but unobserved, survival time. This kind of censoring occurs after the individual has been entered into a study, that is, the true survival time is to the right of (i.e. greater than) the observed survival time, and is therefore known as *right censoring* (Collett, 2003).

Another form of censoring is *left censoring*, which is encountered when the actual survival time of an individual is less than that observed. For illustration, consider a study in which interest centers on the time to recurrence of a particular cancer following surgical removal of the primary tumor. Three months after their operation, the patients are examined to determine if the cancer has recurred. At this time, some of the patients may be found to have a recurrence. For such patients, the actual time to recurrence is less than three months, and the actual recurrence times of these patients are not observed and thus considered left censored (Collett, 2003).

Another type of censoring is *interval censoring*. In this situation, individuals are known to have experienced an event within an interval of time. For illustration, consider again the above example concerning the time to recurrence after a tumor removal surgery. If a patient is observed to be free of the disease at three months, but is found to have had a recurrence when examined six months after surgery, the actual recurrence time of the patient is thus known to be between three months and six months (Collett, 2003).

The application dataset used in this thesis includes right censored data, so **the emphasis of this project will be on the analysis of right censored data.**

An important assumption that will be made in the analysis of censored survival data is that the actual survival time of an individual is independent of any mechanism that may cause that individual's survival time to be censored before the actual survival time. This means that if we consider a group of individuals, all of whom have the same values of relevant prognostic variables, an individual whose survival time is censored at a time point must be representative of all other individuals in that group who have survived to that time. "A patient whose survival time is censored will be representative of those at risk at the censoring time if the censoring process operates randomly"(Collett, 2003). Similarly, when survival data are to be analyzed at a predetermined point in calendar time, or at a fixed interval of time after the time origin for each patient, that is, left censoring or interval censoring may be observed, the prognosis for individuals who are still alive can be taken to be independent of the censoring, as long as the time of analysis is specified before the data are observed. However, the independence assumption cannot be made if, for example, the survival time of an individual is censored because treatment was withdrawn as a result of deterioration in their physical condition. This type of censoring is known as *informative censoring* (Collett, 2003). The non-informative censoring is not uncommon in medical studies. **In this thesis, non-informative censoring is assumed for all application datasets.**

1.2 Exploring Univariate Survival Data - Non-parametric Procedures

The initial step in analyzing survival data can be some exploratory analyses of the survival times for individuals in a particular group. Such summaries may be of immediate interest, or as a precursor to a more detailed analysis of the data, through which the dependence of the survival on some predictor variables can be studied (Collett, 2003). Either the survivor function or the hazard function fully specifies the distribution of the survival time variable, and they are often of interest to be estimated for summarizing the survival data.

1.2.1 Estimating the Survivor Function

We know that the cumulative density function (CDF) of a random variable can be estimated by the empirical distribution function, therefore, with Equation 1.1, the survivor function of T can be estimated by the empirical survivor function, given by

$$\widehat{S}(t) = \frac{\text{Number of individuals with survival times} \geq t}{\text{Number of individuals in the data set}} .$$

The empirical survivor function is constant between any two adjacent observed event times, so it is a step-function. The fact that the survival times of some individuals are censored, however, makes it necessary to take this information into consideration when estimating the survivor function. One of the most frequently used estimates for censored survival data is the *Kaplan-Meier estimate*. Assume the survival data are recorded on n individuals drawn from the population, and let t_i denote the survival time observed for the ith individual. Further assume that the failures are observed on m individuals out of the total n individuals, while the survival times of the other individuals are censored. Let $t_{(1)} \leq t_{(2)} \leq \cdots \leq t_{(m)}$ denote the ordered m failure times. The set of individuals who are alive just prior of time $t_{(l)}$ and thus are subject to the risk of failure is denoted by $R(t_{(l)})$, and the number of individuals in this set is denoted by n_l. Let d_l denote the number who die at this time. Then the Kaplan-Meier estimate of the survivor function is given by:

$$\widehat{S}(t) = \prod_{t_{(l)}<t} \exp\left(\frac{n_l - d_l}{n_l}\right), \tag{1.4}$$

with $\widehat{S}(t) = 1$ for $t < t_{(1)}$, and where $t_{(m+1)}$ is taken to be ∞. If the largest observed time $t_{(m)}$ is an uncensored failure time, then $\widehat{S}(t) = 0$ for $t \geq t_{(m)}$. On the other hand, if the largest observation is a censored survival time, say t^*, then the Kaplan-Meier estimate $\widehat{S}(t)$ is undefined for $t > t^*$. Other methods proposed for estimating the tail of $\widehat{S}(t)$ when the last observation is censored can be used, such as the Brown-Hollander-Korwar tail estimate (Brown et al., 1974), in which they suggest completing the tail by an exponential curve.

If there are no censored observations, then the Kaplan-Meier estimate in Equation 1.4 is reduced to be the ordinary empirical survivor function.

The standard error of the Kaplan-Meier estimate is given by *Greenwood's formula*:

$$\mathrm{se}\left\{\widehat{S}(t)\right\} \approx \widehat{S}(t) \left\{\sum_{t_{(l)}<t} \frac{d_l}{n_l(n_l - d_l)}\right\}^{\frac{1}{2}} .$$

For more details about the Kaplan-Meier estimate and Greenwood's formula, and other estimates of the survivor function, such as the life-table and Nelson-Aalen estimates, the reader can refer to Chapter 2 in Collett 2003.

1.2.2 Estimating the Hazard Function

The survival distribution may also be summarized through the hazard function, which is the instantaneous risk of failure. The Kaplan-Meier estimate of the hazard function is:

$$\widehat{h}(t) = \frac{d_l}{n_l \tau_l}$$

for $t_{(l)} \leq t < t_{(l+1)}$, where $\tau_l = t_{(l+1)} - t_{(l)}$.

The standard error of $\widehat{h}(t)$ is given by

$$\text{se}\left\{\widehat{h}(t)\right\} = \widehat{h}(t)\left\{\frac{n_l - d_l}{n_l d_l}\right\}^{\frac{1}{2}} .$$

For other ways of estimating the hazard function, refer to Chapter 2 in Collett 2003.

1.3 Modeling Univariate Survival Data - Parametric Procedures

The non-parametric methods described in Section 1.2 can be useful for an initial exploration of the survival times observed; however, more often in medical research we desire to study the association between survival and some explanatory variables, and therefore regression techniques that model survival time as a function of the explanatory variables are needed. One popular approach is to fully specify the parametric form of the distribution of survival time, and the two most commonly used distributions are the exponential distribution and the Weibull distribution. Other commonly used distributions include the

Log-Normal distribution, the (Generalized) Gamma distribution, the Log-Logistic Distribution, and the generalized F distribution. All these aforementioned parametric regression models lead to a unified form: a log-linear model. That is, the predictor variables have linear effect on the logarithm of T, where the distribution of the random error determines the parametric form of T.

The log-linear model can be generalized into two classes of models: *relative risk* (also called *proportional hazards*) *model*, and *accelaratedfailure time model*. In the proportional hazards model, the effect of the predictor variables is to act multiplicatively on the hazard function; while in the accelerated failure time model, the predictor variables act multiplicatively on the survival time T directly. The exponential and Weibull regression models are the only two log-linear models that belong to both the class of proportional hazards model and the class of accelerated failure time model (Kalbfleisch and Prentice, 2002). Either class of models can be parametric or semi-parametric models. The paramteric models correspond to the aforementioned distributions and are described in this section, and the semi-parametric models will be discussed in the next section.

1.3.1 Parametric Proportional Hazard Models: the Exponential Model and Weibull Model

Exponential Model

Assume the survival time T is exponentially distributed with parameter λ, i.e., $f(t) = h\exp(-ht)$. The parameter h here is essentially the hazard, and is constant with respect to t. This property is known as the *memoryless property* of the exponential distribution. Further assume we also have measurements on k predictor variables denoted as $\underline{\mathrm{x}}$, where $\underline{\mathrm{x}}$ is vector of length k. We model the hazard at time t to be a function of the predictor variables $\underline{\mathrm{x}}$. The form of the function is not unique, and the most common one to consider is

$$h(t; \underline{x}) = h\exp\left\{\underline{x}'\underline{\beta}\right\}, \tag{1.5}$$

where $\underline{\beta}$ is the vector of coefficients corresponding to the explanatory variables. Thus the conditional PDF of T given $\underline{x}$ is

$$f(t;\underline{x}) = \left[h \exp(\underline{x}'\underline{\beta})\right] \exp\left\{-\left[h \exp(\underline{x}'\underline{\beta})\right] t\right\} . \tag{1.6}$$

Therefore the survival time T is still exponentially distributed, and its dependence on $\underline{\mathrm{x}}$ is directly modeled through the hazard. The inference on the parameters $\underline{\beta}$ can be obtained through the maximum likelihood approach. For more details, the reader can refer to Chapter 3 of Kalbfleisch and Prentice (2002).

Model (1.6) specifies that the log survival time is a linear function of the predictor variables $\underline{x}$. If we let $Y = \log T$, and let

$$Y = \alpha - \underline{x}'\underline{\beta} + W , \tag{1.7}$$

where $\alpha = -\log\lambda$ and W follows the extreme value distribution with PDF $\exp(w - e^w)$, it is easy to show that model (1.6) is equivalent to model (1.7). Model (1.7) is a log-linear model with the error variable W having a specified distribution (Kalbfleisch and Prentice, 2002).

Weibull Model

Let the survival time T follow the Weibull distribution with hazard function

$$h(t) = h\gamma(ht)^{\gamma-1} ,$$

for $h, \gamma > 0$. This hazard is monotone decreasing for $\gamma < 1$, and increasing for $\gamma > 1$, and reduces to the constant exponential hazard if $\gamma = 1$. Therefore, the Weibull model has the flexibility to model monotonically changing hazard. Now consider when we have predictor variables $\underline{x}$, we again can model the hazard to be a function of $\underline{x}$. Specifically, we can model the hazard function as

$$h(t;\underline{x}) = \gamma(ht)^{\gamma-1}\exp(\underline{x}'\underline{\beta})\,. \tag{1.8}$$

Thus the conditional PDF of T given $\underline{x}$ is

$$f(t;\underline{x}) = h\gamma(ht)^{\gamma-1}\exp(\underline{x}'\underline{\beta})\exp\left\{-(ht)^{\gamma}\exp(\underline{x}'\underline{\beta})\right\}. \tag{1.9}$$

The inference of $\underline{\beta}$ can be obtained through the maximum likelihood approach and the reader can refer to Chapter 3 of Kalbfleisch and Prentice (2002).

Equation 1.8 specifies that the predictors act multiplicatively on the Weibull hazard. Model (1.9) can also be expressed as a log-linear model. Let

$$Y = \alpha + \underline{x}'\underline{\beta^{*}} + \sigma W\,, \tag{1.10}$$

where $Y = \log(T)$, $\alpha = -\log h$, $\sigma = \gamma^{-1}$, $\underline{\beta^{*}} = -\sigma\underline{\beta}$, and W follows the extreme value distribution with PDF $\exp(w - e^{w})$. It is easy to see that the Weibull model has an extra scale parameter γ compared to the exponential model, and if $\gamma = 1$, the Weibull model reduces to the exponential model (Kalbfleisch and Prentice, 2002).

In model (1.5) or model (1.8), if we let the predictor variables be 0, we get the so-called baseline hazard, denoted as $h_0(t)$. From (1.5), the baseline hazard in the exponential regression model is h; and from (1.8) in the Weibull model it is $\gamma(ht)^{\gamma-1}$. If we further assume an unspecified form of $h_0(t)$, then we generalize the parametric proportional hazards model to the semi-parametric proportional hazards model (or the famous *Cox proportional hazards model*), and this will be discussed in Section 1.4.

1.3.2 Parametric Accelerated Failure Time Models

The log-linear form of the exponential and Weibull regression models suggests that the predictor variables also act additively on the logarithm of the survival time. From this, we obtain a general class of log-linear models: the accelerated failure time model (Kalbfleisch and Prentice, 2002). Let

$$Y = \underline{x}'\underline{\beta} + \sigma W , \tag{1.11}$$

where $\sigma > 0$, W is an error variable with density $f_w(w)$ and assumed to be independent of $\underline{\beta}$. The interpretation of the above model in terms of $\log T$ is straightforward, it is equivalently saying that the predictor variables have a multiplicative effect directly on T (rather than the hazard function), so the role of $\underline{x}$ is to accelerate (or decelarate) the time to failure (Kalbfleisch and Prentice, 2002).

From (1.11), the PDF of the survival time T is $f(t) = (1/\sigma t)f_w\left((\log t - \underline{x}'\underline{\beta})/\sigma\right)$. If $f_w(w)$ is fully specified, then model (1.11) is a parametric accelerated failure time model. If $W \sim$ extreme value distribution, where $f_w(w) = \exp(w - e^w)$, model (1.11) is the Weibull regression model (including the exponential model), which is also a proportional hazards model.

Log-Normal Model

If $W \sim N(0, 1)$, that is $f_w(w) = (2\pi)^{-1/2}\exp(-w^2/2)$, then T follows the log-normal distribution, and model 1.11 yields the log-normal regression model. The PDF of T can be written as

$$f(t) = (2\pi)^{-1/2}(\sigma t)^{-1}\exp\left[\frac{-\left(\log(t\exp(-\underline{x}'\underline{\beta}))\right)^2}{2\sigma^2}\right].$$

The survivor and hazard functions involve the normal distribution function $\Phi(w) = \int_{-\infty}^{w}\phi(u)du$, where $\phi(u)$ is the PDF of the standard normal distribution. The survivor function is $S(t) =$

$1 - \Phi(\left[\log\left(t\exp(-\underline{x}'\underline{\beta})\right)\right]/\sigma)$, and the hazard function is $f(t)/S(t)$. The hazard function has value 0 at $t = 0$, increases to a maximum and then decreases, approaching zero as t becomes large (Kalbfleisch and Prentice, 2002).

(Generalized) Gamma Model

If $W \sim$ extreme value distribution with one parameter a, that is $f_w(w) = \exp(aw - e^w)/\Gamma(a)$, then model (1.11) yields the generalized gamma regression model. T has PDF

$$f(t) = \frac{\exp(-\underline{x}'\underline{\beta})\left[t\exp-(\underline{x}'\underline{\beta})\right]^{(\frac{a}{\sigma}-1)}\exp\left\{-\left[t\exp-(\underline{x}'\underline{\beta})\right]^{\frac{1}{\sigma}}\right\}}{\sigma\Gamma(a)}.$$

When $\sigma = 1$, $T \sim \Gamma(a, \exp-(\underline{x}'\underline{\beta}))$, and this corresponds to the gamma regression model. The generalized gamma model also includes the exponential model when $\sigma = a = 1$, and the Weibull model when $a = 1$. The log-normal model is also a limiting special case as $a \to \infty$. The hazard function of the generalized gamma distribution incorporates a variety of shapes, as indicated by the special cases. However, the distribution of survival time is most easily visualized in terms of the log survival time Y, that is, through the log-linear model (1.11) (Kalbfleisch and Prentice, 2002).

Log-Logistic Model

If the error variable $W \sim$ logistic distribution, that is $f_w(w) = e^w/(1+e^w)^2$, then the log-linear model yields the log-logistic model. The PDF of T is

$$f(t) = \frac{\exp(-\underline{x}'\underline{\beta})(t\exp(-\underline{x}'\underline{\beta}))^{\frac{1}{\sigma}-1}}{\sigma\left[1+\left(t\exp(-\underline{x}'\underline{\beta})\right)^{\frac{1}{\sigma}}\right]^2}.$$

This model has the advantage of having simple algebraic expressions for the survivor and hazard functions. The survivor and hazard functions are, respectively,

$$S(t) = \frac{1}{1 + \left(t \exp(-\underline{x}'\underline{\beta})\right)^{\frac{1}{\sigma}}} ,$$

and

$$h(t) = \frac{\exp(-\underline{x}'\underline{\beta}) \left(t \exp(-\underline{x}'\underline{\beta})\right)^{\frac{1}{\sigma}-1}}{\sigma(1 + \left(t \exp(-\underline{x}'\underline{\beta})\right)^{\frac{1}{\sigma}})} .$$

The log-logistic model thus is more convenient than the log-normal distribution in handling censored data, "while providing a good approximation to it except in the extreme tails" (Kalbfleisch and Prentice, 2002).

Generalized F Model

If the error variable $W \sim$ the logarithm of an F-distribution with $2m_1$ and $2m_2$ degrees of freedom, then the resulting model from T through model (1.11) is the generalized F distribution. This model incorporates all the foregoing distributions as special cases, and has the advantage that it can adapt to a wide variety of distributional shapes. For more details, the reader can refer to Chapter 2 of Kalbfleisch and Prentice (2002).

There are, of course, other distributions that can be used to model survival data, such as the Gompertz hazard model. All these models are specified through the form of the log-linear model $Y = \underline{x}'\underline{\beta} + \sigma W$, which means that the regression variables have multiplicative effect on the survival time T directly. The likelihood approach is often used for inference on the coefficients. Rank based tests are another alternative to the likelihood approach and the reader can refer to Chapter 7 of Kalbfleisch and Prentice (2002).

The accelerated failure time model with unspecified error distribution can be considered as a semi-parametric model, similar to the semi-parametric Cox proportional hazards model. "Although rank tests for censored data for testing $\underline{\beta} = \underline{\beta_0}$ under the semi-parametric accelerated failure model are readily carried out, the corresponding estimation problem is generally more challenging" (Kalbfleisch and Prentice, 2002). The method for this project is based on the semi-parametric Cox proportional hazards model, therefore we will have a more thorough introduction of the Cox model in the next section.

1.4 Modeling Univariate Survival Data - Semi-parametric Procedure

We have shown in Section 1.3 two major modeling techniques to explore the dependence of survival on explanatory variables: modeling the survival time directly (through the log survival time), or modeling the hazard function directly. Each modeling approach can be of interest for use in its own right, and it is not the purpose of this project to compare them with each other. We will focus on the latter approach, however, because of the availablility of efficient computational algorithms for the proportional hazards models.

1.4.1 The Cox Proportional Hazards Model

Assume the hazard of failure at a particular time depends on the values $x_1, x_2, \cdots, x_k$ of k explanatory variables $X_1, X_2, \cdots, X_k$. The values of these variables will be assumed to have been recorded at the time origin of the study. Let $\underline{x}_i$ be the vector of explanatory variables observed on the ith individual. As mentioned at the end of Section 1.3.1, let $h_0(t)$ be the *baseline hazard function*, which is the hazard for an individual for whom the values of all the explanatory varaibles are zero. Then the general proportional hazard model is

$$h_i(t) = \psi(\underline{x}_i) h_0(t) \, , \tag{1.12}$$

where $\psi(\underline{x}_i)$ is a function of the values of the vector of explanatory variables for the ith individual. The dependence of survival on the explanatory variables is modeled through the $\psi(\cdot)$ function. The $\psi(\cdot)$ function can be interpreted as the hazard at time t for an individual whose vector of explanatory variables is $\underline{x}_i$, relative to the hazard for an individual for whom $\underline{x} = \underline{0}$ where $\underline{0}$ denotes the vector of zeros (Collett, 2003). This model is also known as the Cox proportional hazards model, or Cox regression model (Cox, 1972). Although the model is based on the assumption of proportional hazards, no particular form of distribution is specified for $h_0(t)$, and therefore it is referred to as a semi-parametric model.

The $\psi(\cdot)$ function, as the relative hazard, cannot be negative. There are different forms for $\psi(\cdot)$, but the most commonly used is to take $\psi(\underline{x}) = \exp(\underline{x}'\underline{\beta})$. Thus model (1.12) leads to

$$h_i(t) = \exp(\beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki}) h_0(t) = \exp(\underline{x}_i'\underline{\beta}) h_0(t) \,, \tag{1.13}$$

where β_j is the coefficient correponding to the jth explanatory variable ($j = 1, 2, \cdots, k$), and its magnitude determines the significance of this variable on the hazard. Notice this model can be re-expressed in the form

$$\log\left\{\frac{h_i(t)}{h_0(t)}\right\} = \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} = \underline{x}_i'\underline{\beta} \,.$$

Therefore the proportional hazards model may also be considered as a linear model for the logarithm of the hazard ratio (Collett, 2003). Although it is not needed to specify the distribution of T, the proportional hazards assumption for any pair of values of the explanatory variables is relatively strong. However, the Cox model actually encompasses a wide class of models, as “further extensions of the model to allow stochastic time-dependent explanatory variables are possible and important” (Kalbfleisch and Prentice, 2002).

1.4.2 Estimation of The Coefficients

The Partial Likelihood

The primary method of estimation for the Cox proportional hazards model is called *partial likelihood*. Suppose that m failures are observed on the n individuals. Let $t_{(1)} <$

$t_{(2)} < \cdots < t_{(m)}$ be the ordered m failure times, assuming for now that only one failure can happen at each failure time, that is, there are no ties in the data. Let $R(t_{(l)})$ denote the risk set which is the set of individuals who are alive and uncensored, and thus subject to failure at the time a little prior to $t_{(l)}$. Consider the probability of an individual with associated explanatory variables $\underline{x}_i$ fails at $t_{(l)}$, conditional on $t_{(l)}$ being one of the failure times:

$$\mathrm{P}\left(\text{individual with variables } \underline{x}_{(l)} \text{ fails at } t_{(l)} | \text{ one failure at } t_{(l)}\right) \tag{1.14}$$

$$= \frac{\mathrm{P}\left(\text{individual with variables } \underline{x}_{(l)} \text{ fails at } t_{(l)}\right)}{\mathrm{P}\left(\text{one failure at } t_{(l)}\right)}$$

$$= \frac{\mathrm{P}\left(\text{individual with variables } \underline{x}_{(l)}\text{fails at } t_{(l)}\right)}{\sum_{l\in R(t_{(l)})} \mathrm{P}\left(\text{individual } l \text{ fails at } t_{(l)}\right)}$$

$$= \frac{\text{Hazard at } t_{(l)} \text{ for the individual with } \underline{x}_{(l)}}{\sum_{a\in R(t_{(l)})} \left(\text{Hazard at } t_{(l)} \text{ for individual } a\right)}$$

$$= \frac{h_i(t_{(l)})}{\sum_{a\in R(t_{(l)})} h_a(t_{(l)})},$$

where i indexes the individual who has variables $\underline{x}_{(l)}$ and fails at $t_{(l)}$. It follows that

$$\frac{h_i(t_{(l)})}{\sum_{a\in R(t_{(l)})} h_a(t_{(l)})}$$

$$= \frac{h_0(t_{(l)}) \exp(\underline{x}'_{(l)}\underline{\beta})}{\sum_{a\in R(t_{(l)})} h_0(t_{(l)}) \exp(\underline{x}'_a\underline{\beta})}$$

$$= \frac{\exp(\underline{x}'_{(l)}\underline{\beta})}{\sum_{a\in R(t_{(l)})} \exp(\underline{x}'_a\underline{\beta})}.$$

Therefore, the baseline hazard $h_0(t)$ cancels out, and the inference of $\underline{\beta}$ is not dependent on $h_0(t)$. The partial likelihood is the product of the conditional probabilities over all the m

failure times:

$$L(\underline{\beta}) = \prod_{l=1}^{m} \frac{\exp(\underline{x}'_{(l)}\underline{\beta})}{\sum_{a\in R(t_{(l)})} \exp(\underline{x}'_a\underline{\beta})} \ . \tag{1.15}$$

Individuals for whom the survival times are censored do not contribute to the numerator of the partial likelihood function, but they are included in the summation over the risk sets at failure times that occur before a censored time (Collett, 2003).

Now consider the data of the n individuals, and let $t_1, t_2, \cdots, t_n$ be their observed times. Let δ_i be a binary indicator for the ith individual ($i = 1, 2, \cdots, n$), which is 0 if this individual is right-censored at t_i, and 1 if failure happens at t_i. Thus the partial likelihood function in (1.15) can be expressed as

$$L(\underline{\beta}) = \prod_{i=1}^{n} \left\{ \frac{\exp(\underline{x}'_i\underline{\beta})}{\sum_{l\in R(t_i)} \exp(\underline{x}'_l\underline{\beta})} \right\}^{\delta_i} \ ,$$

where $R(t_i)$ is the risk set at time t_i. The corresponding log-likelihood function is given by

$$\log L(\underline{\beta}) = \sum_{i=1}^{n} \delta_i \left\{ \underline{x}'_i\underline{\beta} - \log \sum_{l\in R(t_i)} \exp(\underline{x}'_l\underline{\beta}) \right\} \ . \tag{1.16}$$

The estimates of the β-estimators in the Cox model shown in (1.13) can be obtained by maximizing the above log-likelihood function; or equivalently, the estimates are solutions to the vector equation

$$U(\underline{\beta}) = \frac{\partial \log L(\underline{\beta})}{\partial \underline{\beta}} = \sum_{l=1}^{m} \left\{ \underline{x}_{(l)} - \frac{\sum_{a\in R(t_{(l)})} \underline{x}_a \exp \underline{x}'_a\underline{\beta}}{\sum_{a\in R(t_{(l)})} \exp \underline{x}'_a\underline{\beta}} \right\} = 0 \ . \tag{1.17}$$

The $U(\underline{\beta})$ is the so-called *score vector*. The equation can be solved using numeric procedures, such as the Newton-Raphson method (Collett, 2003).

Treatment of Ties

The construction of the partial likelihood is based on the assumption of no ties among the survival times, however, it is not uncommon to observe tied survival times. In addition, there might also be one or more censored observations at a failure time. When both censored survival times and failures occur at a given time, the censoring is often assumed to occur after all the deaths, and therefore there is no ambiguity concerning which individuals should be included in the risk set at this time. When there are tied failures observed, the exact partial likelihood at a tied failure time is constructed by breaking the ties in all possible ways and taking the average (Kalbfleisch and Prentice, 2002). Maximizing this partial likelihood, however, is normally computationally intensive, particularly when the number of ties is large at any failure time.

Some approximations to the exact partial likelihood have been proposed and are widely used. One is by Breslow and Crowley (1974), and the approximate partial likelihood over the *m* observed failure times of the *n* individuals is

$$L(\underline{\beta}) \approx \prod_{l=1}^{m} \left\{ \frac{\exp\left(\underline{s}_l'\underline{\beta}\right)}{\left[\sum_{a\in R(t_{(l)})} \exp\left(\underline{x}_a'\underline{\beta}\right)\right]^{d_l}} \right\}, \tag{1.18}$$

where d_l is the number of tied failures at $t_{(l)}$, $l = 1, 2, \cdots, m$, and $\underline{s}_l$ is the vector of sums of each of the k covariates for these d_l individuals. That is, $\underline{s}_l = \sum_{h=1}^{d_l} \underline{x}_{(l)h}$, where $\underline{x}_{(l)h}$ is the vector of covariates for the hth of the d_l individuals who fail at $t_{(l)}$. Here the d_l failures at time $t_{(l)}$ are considered to be distinct and to occur sequentially. The probalilities of all possible sequences of failures are then summed to give the above approximation of the partial likelihood. (1.18) is quite straighforward to compute, and is an adequate approximation when the number of tied observations at any one failure time is not too large (Collett, 2003).

Another approximation method by Efron (1977) is to approximate the partial likelihood as

$$L(\underline{\beta}) \approx \prod_{l=1}^{m} \frac{\exp\left(\underline{s}_l'\underline{\beta}\right)}{\prod_{h=1}^{d_l} \left[\sum_{a\in R(t_{(l)})} \exp\left(\underline{x}_a'\underline{\beta}\right) - (h-1)d_l^{-1} \sum_{a\in D(t_{(l)})} \exp\left(\underline{x}_a'\underline{\beta}\right)\right]}, \tag{1.19}$$

where $D(t_{(l)})$ is the set of all individuals who fail at time $t_{(l)}$. This is a closer approximation to the appropriate partial likelihood function than Breslow's approximation method, although in practice, both methods often give similar results (Collett, 2003). Most statistical packages for survival analysis provide the options of these two approximation methods for the situation with tied failures.

1.4.3 Definition and Some Theory for Partial Likelihood

Cox (1972) introduced the concept of partial likelihood and Cox (1975) provided more formal justification for the likelihood-like properties of the partial likelihood function (Crowder, 2001). The partial likelihood is not the ordinary likelihood function, however, the estimates from maximizing the partial likelihood have asymptoic properties similar to the traditional maximum likelihood estimators. Andersen and Gill (1982) used multivariate counting process set-up to derive the asymptotic properties of the coefficients estimated from the Cox partial likelihood. Gill (1984) provided "a readable illustration" of the techniques behind the Cox model (Crowder, 2001). Here we introduce some basic theory about the partial likelihood following Section 4.4.5 of Crowder (2001).

The generic set-up is as follows. Suppose the parameter set is denoted as $\underline{\theta}$, and let $\underline{\theta} = (\underline{\beta}, \phi)$, where $\underline{\beta}$ is the vectors of parameter of interest and ϕ is a vector of nuisance parameters that is often of very high or infinite dimension. In some applications, ϕ can be a nuisance function, such as the baseline hazard function $h_0(t)$ in the Cox regression model (1.13), which is of inifinite dimension. Suppose further that the accumulating data sequence can be formulated as $d_l = ((A_1, B_1), (A_2, B_2), \cdots, (A_l, B_l))$ for $l = 1, 2, \cdots m$. Then the likelihood function can be written as

$$
\begin{aligned}
L(\underline{\theta}) = f_{\underline{\theta}}(D_m) &= \prod_{l=1}^{m} f_{\underline{\theta}}(A_l, B_l|D_{l-1}) && (1.20)\\
&= \prod_{l=1}^{m} f_{\underline{\theta}}(A_l|D_{l-1}, B_l) f_{\underline{\theta}}(B_l|D_{l-1})\\
&= \prod_{l=1}^{m} f_{\underline{\theta}}(A_l|D_{l-1}, B_l) \times \prod_{l=1}^{m} f_{\underline{\theta}}(B_l|D_{l-1})\\
&= P(\underline{\beta}) \times Q(\underline{\theta})\ ,
\end{aligned}
$$

where $f_{\underline{\theta}}$ is a generic notation for a probability density or mass function. $P(\underline{\beta})$ is called the *partial likelihood* of $\underline{\beta}$ based on A_l in the sequence $\{A_l, B_l\}$. Normally $Q(\underline{\theta})$ depends to some extent on $\underline{\beta}$ and therefore contains some residual information about $\underline{\beta}$. Cox's justification for ignoring $Q(\underline{\theta})$, in the case of the proportional hazards model, is that this residual information is unavailable because $\underline{\beta}$ and ϕ are inextricably entangled in $Q(\underline{\theta})$ (Crowder, 2001). The partial likelihood $P(\underline{\beta}) = \prod_{l=1}^{m} f_{\underline{\theta}}(A_l|D_{l-1}, B_l)$ arises as the product of conditional probability statements, but is not directly interpretable as a likelihood in the ordinary sense. In general, "it cannot be given any direct probability interpretation as either the conditional or the marginal probability of any event. Nonetheless, in many instances it can be used like an ordinary likelihood for purpose of large-sample estimation in that the usual asymptotic properties formulas and the properties associated with the likelihood function and likelihood estimation apply" (Kalbfleisch et al.2002).

In the proportional hazards set-up, the distinct observed failure times are $t_{(1)} < t_{(2)} < \cdots < t_{(m)}$ and the censoring times during the interval $[t_{(l)}, t_{(l+1)})$ are t_{ls} $(s = 1, 2, \cdots, s_l)$, where s_l is the number of censored observations during this interval. Assuming no ties, the individual who fails at time $t_{(l)}$ has index i_l, and those censored during $[t_{(l)}, t_{(l+1)})$ have indices i_{ls} $(s = 1, 2, \cdots, s_l)$. Let $C_l = \{t_{ls}, i_{ls} : s = 1, 2, \cdots, s_l\}$ comprise the full record of censored cases during the interval $[t_{(l)}, t_{(l+1)})$. Now take $A_l = \{i_l\}$ and $B_l = \{C_{l-1}, t_l\}$ in the above factorization of $L(\underline{\theta})$. Then $f_{\underline{\theta}}(A_l|D_{l-1}, B_l)$ is just the conditional probability in (1.14), and so $P(\underline{\beta})$ is the $L(\underline{\beta})$ in (1.15).

The score function of the partial likelihood is

$$U(\underline{\beta}) = \frac{\partial \log P(\underline{\beta})}{\partial \underline{\beta}} = \sum_{l=1}^{m} \frac{\partial \log f_{\underline{\beta}}(A_l|D_{l-1}, B_l)}{\partial \underline{\beta}} = \sum_{l=1}^{m} U_l(\underline{\beta}) \, .$$

Under usual regularity conditions, and conditional on $\{D_{l-1}, B_l\}$, $f_{\underline{\beta}}(A_l|D_{l-1}, B_l)$ is a density function (Kalbfleisch et al.2002), so it can be shown that

$$U(\underline{\beta}) \sim N(0, I(\underline{\beta})) \text{ as } m \to \infty \, ,$$

where $I(\underline{\beta})$ is the expected information matrix:

$$I(\underline{\beta}) = -E\left[\frac{\partial^2 \log f_{\underline{\beta}}(A_l|D_{l-1}, B_l)}{\partial \underline{\beta} \partial \underline{\beta}'}\right] \, . \tag{1.21}$$

Let $\hat{\underline{\beta}}$ be the maximum partial likelihood estimator obtained from solving $U(\underline{\beta}) = \underline{0}$, and let $I_{obs}(\underline{\beta}) = -\partial^2 \log L(\underline{\beta})/\partial \underline{\beta} \partial \underline{\beta}'$ be the observed information matrix from the partial likelihood. Under certain conditions (Crowder, 2001, Page 75), $\hat{\underline{\beta}}$ is consistent for the true parameter $\underline{\beta}_0$; and the asymptotic distribution of $\hat{\underline{\beta}}$ is

$$\hat{\underline{\beta}} \sim N\left(\underline{\beta}_0, I(\underline{\beta}_0)\right) \text{ as } m \to \infty \, ,$$

where $I(\underline{\beta}_0)$ is the information matrix in (1.21) evaluated at $\underline{\beta}_0$. In practice, $I(\underline{\beta}_0)$ can be estimated by $I_{obs}(\underline{\beta})$. In the Cox proportional hazards model (1.12), with $\psi(\underline{x}) = \exp(\underline{x}'\underline{\beta})$, the conditions for the asymptotic properties are usually met.

For more discussions about the large sample properties for partial likelihood estimators, the reader can refer to Chapter 4 of Crowder (2001) and Chapter 4 of Kalbfleisch et al.(2002). Specifically, for the asymptotic properties of the partial likelihood estimators in the Cox model, the reader can refer to Andersen and Gill (1982) and Gill (1984), in which counting process theory was used to rigorously derive the large sample properties.

1.5 Some Topics about Multivariate Survival Data

So far the methods for survival analysis we have discussed are applicable to the situation when there is one single (possibly censored) failure time of the same type on each study subject, and the different subjects are assumed to be independent. This is what we call *univariate survival data*. In some applications, the survival data observed might be of more complex structures, such as the so-called *multivariate survival data* (Hougaard, 2000). The fundamental characteristic of multivariate survival data is that independence between survival times cannot be assumed, which adds complexity when modeling and analyzing the data.

One structure of multivariate survival data involves *recurrent events*. That is, a single individual might experience the same event multiple times during the study period. The failure times observed on the same individual apparently cannot be assumed to be independent. One simple example arises for times to tumor recurrence among patients of a certain type of cancer. The shared frailty model presented in Chapter 9 of Hougaard (2000) is exclusively for analyzing recurrent events data. Chapter 9 of Kalbfleisch and Prentice (2002) also provides methods which directly model the intensity process corresponding to the recurrent events. The rationale of these methods is quite intuitive with some knowledge about counting process theory.

Another situation when multivariate survival data arise is *multi-state data*. Similar to the recurrent events data, multiple failure times are observed on a single individual, however, these times correspond to the occurrence of events of distinct types. Generally the life history of an individual under study may involve multiple types of failures that happen longitudinally. For example, a patient in a study might be first observed to have a certain disease, and the patient is followed until death. The patient therefore experiences two events: the disease and the death. He or she is transitioned from the state of being free of the disease, to the state of being diseased, and then to the state of being dead. Chapter 5 and 6 of Hougaard (2000) elaborates different scenarios of multi-state data and the corresponding modeling techniques. Kalbfleisch and Prentice (2002) also provides models based on Markov process in Chapter 8.

In some studies, although each individual is only observed with one event, the survival times of different individuals may not be independent, such as the times recorded on members from the same family, or the data generated from multi-center studies. The frailty model has the flexibility for modeling this kind of correlated (or clustered) data and is presented in Chapter 7 and 8 of Hougaard (2000). The model of jointly modeling the correlated survival times within a cluster is described in Chapter 10 of Kalbfleisch and Prentice (2002).

One more type of multivariate survival data is *competing risks*. Each individual is observed with one failure time, however, the failure may be one of several distinct failure types; or the failure happens because of one of multiple causes, that is, the different causes are competing to be the final reason for the failure to happen on the patient. With competing risks data, three problems might be of interest (Kalbfleisch and Prentice, 2002): 1. To estimate the relationship between some explanatory variables and the rate of occurrence of failures of specific types (or causes). 2. To study the interrelation between failure types. 3. To estimate failure rates for certain types of failure given the removal of some or all other failure types. “Strictly speaking, however, competing risks data is not multivariate survival data, as only one time is observed on each subject, and thus it is likely impossible to study the dependence between failure types” (Hougaard, 2000). This fact determines that only under some specific study conditions, the three problems of interest, especially the last two problems, can be answered. The reader can refer to page 249 of Chapter 8 in Kalbfleisch and Prentice (2002) for a more detailed discussion.

This project focuses on analyzing competing risks data, and the main purpose is to study the association between explanatory variables and the failures caused by different reasons. In the next chapter, we will introduce more about competing risks, including the mathematical notation and some modeling techniques.

Chapter 2

More About Competing Risks

In Chapter 1, we introduced the topic of survival analysis and reviewed some popular modeling techniques for univariate survival data. Some topics where multivariate survival data arise were also introduced at the end of Chapter 1. In this chapter, we will discuss in detail the specific topic of competing risks. We start with Section 2.1 to introduce the probabilistic framework for the description of competing risks survival data. The hazard functions for competing risks are defined in Section 2.2. The modeling techniques are introduced in Section 2.3 and Section 2.4, including the traditional latent time variable approach and its limitation, and the hazard based approach - the proportional hazards model for competing risks. Two examples, one derived from a prostatic cancer clinical trial and the other from an ongoing NIH (National Institutes of Health) funded project studying hepatitis C virus (HCV) infected patients diagnosed with hepatocellular carcinoma (HCC), are presented in Section 2.5 to illustrate the situation where competing risks can arise in real-world research.

2.1 Introduction of Competing Risks

The earliest recorded attempt at modeling competing risks involved analyzing cause-specific mortality from smallpox, among other diseases, in order to estimate the effectiveness of smallpox vaccination by Bernoulli in 1760. Seal (1977) and David and Moeschberger (1978) provide more work in the field. Even though the field of statistics has made extensive progress in modeling time-to-event data, specifically, modeling time-to-event in the presence of competing risks is relatively new (Crowder, 2001).

In classical competing risks, the observed outcome is denoted as (T, C), where T represents the time to failure and C represents the cause of failure. Similar to traditional univariate survival modeling approaches, T is a continuous random variable. However, in traditional survival analytic models, C often is a dichotomous variable indicating that the individual either experienced the event or was censored. In competing risks models, C represents the cause of failure and hence is a discrete random variable taking one of a fixed (normally small) number of values, labeled as $1, 2, \cdots, p$. Therefore, for our competing risks models, the survival outcome is from a bivariate distribution with one continuous component and one discrete component. It is a key feature of competing risks that to every failure one and only one cause can be assigned from the given set of p causes, that is, the different causes (or failure types) compete to be the final cause of failure. For example, in the field of renal transplantation, C can be the cause of graft failure and T is the time from transplant until graft failure. Competing risks modeling is also useful for applications in other fields, such as the field of engineering where C can indicate the failing component of an electronic system.

The identifiable probabilistic framework for competing risks is the joint distribution of C and T, which can be specified through the so-called *sub-distribution function* $F(j, t) = P(C = j, T \leq t)$, or equivalently by the *sub-survivor function* $S(j, t) = P(C = j, T > t)$ Crowder (2001). Note that the sum of $F(j, t)$ and $S(j, t)$ is not unity, but $F(j, t)+S(j, t) = p_j$, where $p_j = P(C = j) = F(j, \infty) = S(j, 0)$ is the marginal distribution of C and is the probability of cause j to "win". Thus $F(j, t)$ is not a proper distribution function. It is assumed that $p_j > 0$ and $\sum_{j=1}^{p} p_j = 1$. The *sub-density function* for T corresponding to cause j is $f(j, t) = -dS(j, t)/dt$.

The marginal survivor function and marginal density function of T can be calculated from

$$S(t) = \sum_{j=1}^{p} S(j,t)\ ,$$

and

$$f(t) = \frac{-dS(t)}{dt} = \sum_{j=1}^{p} f(j,t)\ .$$

Some related conditional probabilities may be of interests in real-world contexts. The conditional probability $P(\text{failure at } t|\text{cause } j) = f(j,t)/p_j$, for instance, provides the distribution of time to graft failure from cause j in the aforementioned renal transplant example; the conditional probability $P(\text{cause } j|\text{failure at } t) = f(j,t)/f(t)$, provides the probability of graft failure from cause j at a specified time; or $P(C = j|T > t) = S(j,t)/S(t)$, for example, gives the probability of ultimate graft failure from cause j for a patient observed at time t post-transplant.

2.2 Hazard Functions in Competing Risks

The hazard function corresponds to the conditional instantaneous rate of failure and is defined in subsection 1.1.1 for univariate survival data. There are various hazard functions describing probabilities of imminent failure associated with the competing risks set up: the *overall hazard function* from all failure types (or causes) and the failure type specific (or cause specific) *sub-hazard function*.

2.2.1 Sub-Hazard and Overall Hazard

The Overall Hazard

The overall hazard function is defined as:

$$\begin{aligned} h(t) &= \lim_{\delta t \to 0} \frac{P(T \leq t + \delta t | T > t)}{\delta t} \\ &= \lim_{\delta t \to 0} \frac{S(t) - S(t + \delta t)}{S(t)\delta t} \\ &= \frac{f(t)}{S(t)} \\ &= \frac{-d \log S(t)}{dt} \quad , \end{aligned} \tag{2.1}$$

where T is the random variable representing time to failure, and failure can be due to one of the p failure types.

The sub-hazard

The hazard function for failure due to cause j, in the presence of all p risks, is defined as:

$$\begin{aligned} h(j,t) &= \lim_{\delta t \to 0} \frac{P(C = j, T \leq t + \delta t | T > t)}{\delta t} \\ &= \lim_{\delta t \to 0} \frac{S(j,t) - S(j,t + \delta t)}{S(t)\delta t} \\ &= \frac{f(j,t)}{S(j,t)} , \end{aligned} \tag{2.2}$$

So $h(t) = \sum_{j=1}^{p} h(j,t)$.

As mentioned in Section 1.5, sometimes in applications it is of interest to assess the consequence of changes in certain risks. For example, in oncology, neo-adjuvant chemotherapy in combination with surgery may improve survival for patients with breast cancer compared to surgery alone. Suppose during a time period $I = (a, b)$, the sub-hazard $h(j,t)$ of cause j is increased whereas the sub-hazards of other risks are not changed, one would intuitively expect that the overall probability of failure in period I is increased, while the relative probability of failure from a cause other than j in I is decreased (Crowder, 2001). The following theorem (Kimball, 1969) states this formally. And the opposite conclusions would hold if $h(j,t)$ were decreased over I.

THEOREM 2.1 (Kimball (1969))**.** *Suppose that for the interval $I = (a, b)$, $\int_a^b h(C, t)$ is increased for $C = j$ only. Then,*

(i) P_I, the probability of failure in $I = (a, b)$, conditional on survival to enter I, is increased;

(ii) $P_{Ij'}$, the probability of failure in I from cause j', conditional on entry to I, is decreased for $j' \neq j$.

The proofs follow from Crowder (2001) with some additional details. We have

$$\begin{aligned} P_I &= P(a < T \leq b | T > a) \\ &= \frac{(S(a) - S(b))}{S(a)} \\ &= 1 - \exp\left\{-\int_a^b h(t)dt\right\} \quad , \end{aligned}$$

using Equation (1.3). Since obviously $h(t) = \sum_{j=1}^{p} h(j, t)$ is increased with the increase in the specified sub-hazard, so the exponential term is decreased. Thus *(i)* is verified.

For *(ii)*, we have

$$\begin{aligned} P_{Ij'} &= P(C = j', a < T \leq b | T > a) \\ &= \frac{(S(j', a) - S(j', b))}{S(a)} \\ &= \frac{\int_a^b f(j', t)dt}{S(a)} \\ &= \int_a^b \frac{h(j', t)S(t)}{S(a)} dt \quad . \end{aligned}$$

Since $h(j', t)$ is not changed, and from (i), $S(t)/S(a)$ is increased, hence we get the result in *(ii)*. □

As for univariate survival data, modeling of competing risks survival data can be specified directly in terms of the sub-hazards, other than the sub-survivor function or sub-density function. Next we will particularly introduce the concept of proportional hazards in the context of competing risks.

2.2.2 Proportional Hazards in Competing Risks

If the relative risk of failure from cause j at time t, $h(j,t)/h(t)$, is independent of t for each $j(j = 1, 2, \cdots, p)$, then *proportional hazards* are said to be obtained. This means that as time goes on, "the relative risks of the various causes of failure stay the same, none increasing its share of the overall risk, though the overall risk might change along time" (Crowder, 2001).

The following theorem states some conditions where proportional hazards is obtained.

THEOREM 2.2 (Elandt-Johnson (1976); David and Moeschberger (1978); Kochar and Proschan (1991))**.** *The following conditions are equivalent:*

(i) proportional hazards is obtained;

(ii) the time and cause of failure are independent;

(iii) $h(j,t)/h(j',t)$ *is independent of* t *for all* j *and* j', $j, j' = 1, 2, \cdots, p$*;*

If either condition holds, then $h(j,t) = p_j h(t)$*, or equivalently,* $f(j,t) = p_j f(t)$*, or* $F(j,t) = p_j F(t)$

The proof follows what is outlined on Page 13 of Crowder (2001).

Part (ii) of the theorem indicates that failure during some particular period does not make it more or less likely to be from cause j than failure in another period. In many areas of application, however, this probably would be an exceptional situation. For example, in public health sciences, the relative risks of cot death, and senile dementia might be expected to differ with age (Crowder, 2001). But the concept of proportional hazards is still attractive, as the proportional hazards assumption often is reasonable at least in a piecewise fashion along the time scale (Chiang (1961), David (1970), Seal (1977)).

2.3 Modeling Continuous Competing Risks Data - The Traditional Latent Lifetimes Approach and Its Limitation

The specification of competing risks survival data that we presented in Section 2.1 and 2.2 is based on the joint modeling of the failure time and the failure cause (or failure type). The traditional route is based on assigning a set of latent lifetime variables corresponding to the multiple failure causes. Although this approach is intuitive at first sight, it is either accompanied by the additional assumption of independence between these latent lifetimes, which may not be applicable to real applications; or without the independence assumption, there is a certain issue of model identifiability presented in detail in Chapter 7 of Crowder (2001).

2.3.1 Introduction of Latent Lifetimes

In the traditional approach for describing competing risks survival data, it is assumed that there is a potential failure time associated with each of the p risks to which an individual is exposed. Let T_j denotes the time to failure from cause j ($j = 1, 2, \cdots, p$). Then the smallest T_j ($j = 1, 2, \cdots, p$) determines the time T to overall failure, and we use its index C to denote the cause of failure, i.e., $T = \min\{T_1, T_2, \cdots, T_p\} = T_C$. Once the individual has failed, the remaining lifetimes corresponding to the individual causes are lost to observation (Crowder, 2001).

The joint distribution of the random vector $\underline{T} = (T_1, T_2, \cdots, T_p)$ is used to statistically describe the competing risks data. The *joint distribution function* is defined as $G(\underline{t}) = P(\underline{T} \leq \underline{t}) = P(T_1 \leq t_1, T_2 \leq t_2, \cdots, T_p \leq t_p)$; and similarly, the *joint survivor function* is $\overline{G}(\underline{t}) = P(\underline{T} > \underline{t})$. If the T_j ($j = 1, 2, \cdots, p$) are jointly continuous, the *joint density* is $\partial^p G(\underline{t})/\partial t_1 \partial t_2 \cdots \partial t_p$, or equivalently, $(-1)^p \partial^p \overline{G}(\underline{t})/\partial t_1 \partial t_2 \cdots \partial t_p$.

Independence of the T_js is defined by

$$G(\underline{t}) = \prod_{j=1}^{p} G_j(t_j) \quad ; \tag{2.3}$$

or equivalently,

$$\overline{G}(\underline{t}) = \prod_{j=1}^{p} \overline{G}_j(t_j) \quad , \tag{2.4}$$

where $G_j(t_j) = P(T_j \leq t_j)$ is the *marginal distribution function* of T_j and $\overline{G}_j(t_j) = P(T_j > t_j)$ is the *marginal survivor function* of T_j, where T_j corresponds to the time to failure due to cause j specifically.

It will be assumed that T_j $(j = 1, 2, \cdots, p)$ are continuous and that ties cannot happen, that is, $P(T_j = T_{j'}) = 0$ for all $j \neq j'$, otherwise C is not easily defined (refer to Section 7.2 of Crowder (2001)).

2.3.2 Marginal Distribution of the Latent Failure Times and the Sub-distribution

In the older terminology, the sub-survivor function $S(j, t)$, defined in Section 2.1, was called the *crude survivor function*; and the marginal distribution function $\overline{G}_j(t)$ was called the *net survivor function*. These two ways of modeling competing risks survival data are not irrelevant. If the joint survivor function $\overline{G}(\underline{t})$ of $\underline{T}$ has known form, then the overall survivor function $S(t)$ can be obtained as $\overline{G}(t\underline{1}_p)$, where $\underline{1}_p = (1, 1, \cdots, 1)$ is of length p. Moreover, the following theorem shows the relation between the sub-density function and the joint survivor function of the latent failure times.

THEOREM 2.3 (Tsiatis (1975))**.** *The sub-density function can be calculated directly from* *the joint suvrivor function of the latent failure times as*

$$f(j,t) = \frac{\partial \overline{G}(\underline{t})}{\partial t_j} \,|_{t\underline{1}_p} \quad . \tag{2.5}$$

The proof of the theorem can be found on Page 38 of Crowder (2001) and is omitted here.

It follows from the theorem that the sub-hazard function can also be calculated directly from the joint survivor function of the latent time variables $\overline{G}(\underline{t})$ as

$$h(j,t) = \frac{f(j,t)}{S(t)} = \frac{-\partial \log \overline{G}(\underline{t})}{\partial t_j} \,|_{t\underline{1}_p} \quad .$$

2.3.3 The Identifiability Crisis of Latent Failure Times Approach

Describing competing risks from the point of view of latent lifetimes seems very natural, and statistically, one can specify a parametric model for the joint multivariate distribution of the latent time variables $\overline{G}(\underline{t})$, fit it to the data, and do the inference, etc. However, one problem of the approach of latent time variables is that the $\overline{G}_j(t)$ $(j = 1, 2, \cdots, p)$ do not describe events that physically occur - "they only describe failures from isolated causes in situations where all the other risks have been removed somehow" (Crowder, 2001). It is the $S(j,t)$, the sub-distributions, not the $\overline{G}_j(t)$, the marginal distribution of latent variables, that are truly observed in the real situation. Moreover, it often happens that the whole mechanism is changed after the removal of the other causes, and therefore it is not valid to assume that when T_j is observed without other competing risks, its distribution is the same as its marginal distribution derived from the joint distribution (Crowder, 2001). Even more so, without the assumption of independence between the latent times, which is often the reality, the modeling of latent times can have a serious identifiability problem brought out by Cox (1959). Cox (1959) studied various parametric models with two causes, and Tsiatis (1975) extended to the general case of p risks. They showed that, "given any joint survivor function with arbitrary dependence between the component variates (i.e., the latent times), there exists a different joint survivor function in which the variates are independent and which reproduces the sub-densities $f(j,t)$ precisely" (Crowder, 2001). This implies that from the same set of observations on (C,T) alone, we can have two different models that fit the data equally well; that is, model identifiability problem arises with the latent time variables approach. The following theorem theoretically describes this problem. The proof follows that in Chapter 7 of Crowder (2001) with additional details and correction of a mistake, and can be found in the appendix.

THEOREM 2.4 (Tsiatis (1975))**.** *Suppose that the set of $S(j,t)$ is given for some model with dependent risks. Then there exists a unique proxy model with independent risks yielding identical $S(j,t)$. It is defined by $\overline{G}(\underline{t}) = \prod_{j=1}^{p} \overline{G}_j^*(t_j)$, where $\overline{G}_j^*(t_j) = \exp\left\{-\int_0^t h(j,s)ds\right\}$ and the sub-hazard functiion $h(j,s)$ derives from the given $S(j,t)$.*

The theorem establishes only that to each dependent-risks model there corresponds a unique independent-risks proxy model with the same sub-survivor functions. Moreover, it has been shown (Crowder, 1991) that each independent-risks model actually has a whole class of satellite dependent-risks models and that this class can be further partitioned into sets with the same marginal functions. Therefore, "it is not possible to obtain from the observations of (C,T) unique information about the distributions of cause-specific failure times or on the dependence structure between them" (Crowder, 2001). An exception is the case of regression model where there are explanatory variables in the model, identification is possible within a certain framework (Heckman and Honor, 1989), though Kalbfleisch and Prentice (2002, page 261) pointed out that it is still more straightforward to consider only specifying the cause-specific hazards since in fact only the cause-specific hazards (i.e. sub-hazards) enter the likelihood function and they are all that is needed to specify the joint distribution of the failure time and the cause (T,C).

For more discussions and some examples about the problem of non-identifiability, the reader can refer to Chapter 7 of Crowder (2001).

2.4 Modeling Continuous Competing Risks Data - The Hazard Based Approach

The traditional latent failure time approach has been heavily criticized by Prentice et al. (1978) and Kalbfleisch and Prentice (2002). The main criticism has been that the joint survivor function suffers from the aforementioned identifiability problem. The fact that strong untestable assumptions are needed about the nature of the failure mechanism and the effect of cause removal is another weakness of the latent failure time approach. In addition, the existence of hypothetical latent failure times is highly questionable.

In Section 6.1 of Crowder (2001), the author partially defended for the traditional latent time approach over the aforementioned strong arguments by Prentice et al. (1978) and Kalbfleisch and Prentice (2002). A doctrine brought out by the author, however, is that “one should set up models only for observable phenomena”, i.e., “a kind of what you see is what you set” doctrine; and his main recommendation lies on the hazard-based (specifically, sub-hazard) approach as “models for processes evolving over time can be developed much more naturally in terms of hazards than multivariate survivor functions. Thus one can deal with quite complex situations that would be difficult, even intractable, from the traditional point of view”. Section 6.2 and 6.3 of Crowder (2001) provided some nice examples of parametric modeling of the hazard function, and also some non-parametric methods. Here we will review in detail the semi-parametric proportional hazards regression model for competing risks.

2.4.1 Proportional Hazards Model for Competing Risks

The ordinary Cox proportional hazards model for univariate survival data is $h(t, \underline{x}) = \psi(\underline{x})h_0(t)$ (Equation (1.12)), where h_0 is an unspecified baseline hazard function and $\psi(\underline{x})$ is a positive function of the vector $\underline{x}$ of explanatory variables, such as the commonly used

$\psi(\underline{x}) = \exp \underline{x}'\underline{\beta}$. There is a difference between what is meant by "proportional hazards" in traditional Cox proportional hazards model compared to proportional hazards as defined in the subsection 2.2.2 for competing risks survival data. However, the cause-specific sub-hazard from the competing risks model can be rewritten to mimic the more familiar Cox proportional hazards model.

Following the notation used in Section 1.4, for the ith individual, the sub-hazard functions for the $j = 1, 2, \cdots, p$ causes are specified as

$$h(j, t; \underline{x}_i) = \psi_{j,i} h_0(j, t) \quad , \tag{2.6}$$

where the $h_0(j, t)$, $j = 1, 2, \cdots, p$, form a set of baseline sub-hazards that not necessarily need be explictly specified, and $\psi_{j,i} = \psi_j(\underline{x}_i; \underline{\beta}^j)$ is some positive function of both $\underline{x}_i$, the vector of explanatory variables for the ith individual, and $\underline{\beta}^j$, the associated vector of regression coefficients corresponding to cause j. One popular choice of the ψ function is $\psi_{j,i} = \exp \underline{x}_i'\underline{\beta}^j$. The parameter vector, i.e., the vector of coefficients, in the full model thus is

$$\underline{\beta} = \left(\beta_1^1, \beta_2^1, \cdots, \beta_k^1 \; ; \; \beta_1^2, \beta_2^2, \cdots, \beta_k^2 \; ; \cdots \; ; \; \beta_1^p, \beta_2^p, \cdots, \beta_k^p\right)'$$

of length $k \times p$, where k is the number of explanatory variables. Since the same explanatory variable may have different effects on the different risks, it is reasonable to assume that the $\underline{\beta}^j$, $j = 1, 2, \ldots, p$, vectors are independent of each other. In practical applications, one may want to seek for a parsimonious model by testing for restrictions on the $\underline{\beta}^j$s, such as $\underline{\beta}^1 = \underline{\beta}^2$, or some particular components of $\underline{\beta}^j$ are zero.

The construction of the partial likelihood is similar to that for univariate proportional hazards model described in subsection 1.4.2. Suppose that m failures are observed on the n individuals. Let $t_{(1)} < t_{(2)} < \cdots < t_{(m)}$ be the ordered m failure times, with $t_0 = 0$ and $t_{m+1} = \infty$, assuming for now that only one failure can happen at each failure time, that is, there are no tied failure times in the data, and let $R(t_{(l)})$ be the risk set at time $t_{(l)}$.

The probability that individual $i \in R_{(l)}$ fails from cause j in the time interval $(t_{(l)}, t_{(l)} + dt]$ is $h(j, t_{(l)}; \underline{x}_i)dt$. Suppose that $t_{(l)}$ is the failure time of the individual i_l, and the observed cause being c_l. Given the events up to time $t_{(l)}^-$, and given that there is a failure of type c_l at time t_l, the conditional probability that, among all the individuals in $R(t_{(l)})$, it is individual i_l who fails, is

$$\begin{aligned}\frac{h(c_l, t_l; \underline{x}_{i_l})dt}{\sum_{a\in R(t_{(l)})} h(c_l, t_l; \underline{x}_a)} &= \frac{\psi_{c_l,i_l}\, h_0(c_l, t)}{\left(\sum_{a\in R(t_{(l)})} \psi_{c_l,a}\right) h_0(c_l, t)} \\ &= \frac{\psi_{c_l,i_l}}{\sum_{a\in R(t_{(l)})} \psi_{c_l,a}} \quad ,\end{aligned} \tag{2.7}$$

where $\sum_{a\in R(t_{(l)})}$ is the summation over individuals in the risk set at $t_{(l)}$, and baseline subhazard function, as in the partial likelihood of univariate proportional hazards model in the subsection 1.4.2, have canceled out. The corresponding partial likelihood function thus is,

$$P(\underline{\beta}) = \prod_{l=1}^{m} \left(\frac{\psi_{c_l,i_l}}{\sum_{a\in R(t_{(l)})} \psi_{c_l,a}} \right)$$

The maximum partial likelihood estimator of $\underline{\beta}$ can be obtained by maximizing $P(\underline{\beta})$ over $\underline{\beta}$, or equivalently, by solving the equation that the score function equals zero (i.e. $U(\underline{\beta}) = \partial \log P(\underline{\beta})/\partial \underline{\beta} = \underline{0}$). Large-sample inference can be conducted by treating $\log P(\underline{\beta})$ as a log-likelihood function in the usual way. Under usual regularity conditions, the inverse of the observed information matrix $I(\underline{\beta}) = -\partial^2 \log P(\underline{\beta})/\partial\underline{\beta}\partial\underline{\beta}^{'}$ provides an estimate for the variance-covariance matrix of $\underline{\beta}$.

The set-up of the partial likelihood of the proportional hazards model for competing risks also conforms to the generic theory about the partial likelihood described in the subsection 1.4.3. Here we take $A_l = \{i_l\}$, and $B_l = \{C_{l-1}, t_{(l)}, c_l\}$, then the $f_{\underline{\theta}}(A_l | D_{l-1}, B_l)$ in Equation (1.20) is just the conditional probability in Equation (2.7).

For more details about this topic, including the proportional hazards model for com-

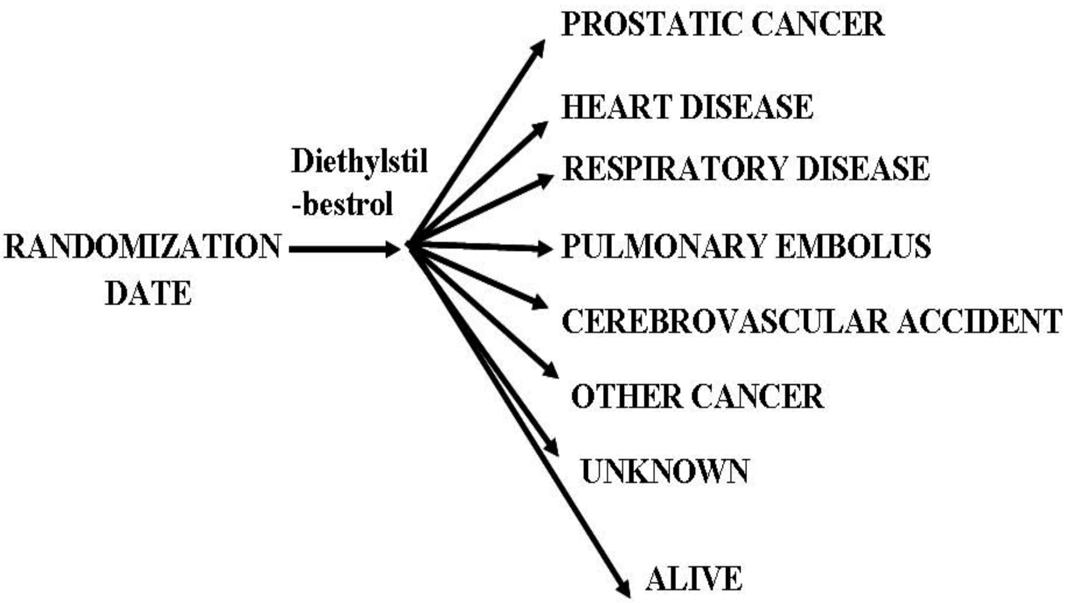


Figure 2.1: Survival Outcomes of the Prostate Cancer Study

peting risks using the counting process framework, the reader can refer to Section 6.4 and Section 8.10 of Crowder (2001), and Section 8.2 of Kalbfleisch and Prentice (2002).

2.5 Examples of Competing Risks Problems

2.5.1 Prostatic Cancer Data

The prostatic cancer data were obtained from a randomized clinical trial conducted in the late 1960's comparing four treatments for patients with advance-staged prostatic cancer (Stage 3 and 4). The trial was double-blinded and the treatments were placebo pill, 0.2 mg diethylstilbestrol (DES), 1.0 mg of DES, or 5.0 mg of DES, all drugs administered daily by mouth. The patients were followed at 6 month intervals according to a standard protocol or more frequently if required. The survival outcomes of 506 patients were collected during the trial, and the outcomes were categorized into different categories of diseases. Figure 2.1 summarizes the categorization of the survival outcomes.

It can be seen from Figure 2.1 that each patient was subject to the risks of multiple kinds of diseases, such as prostate cancer, cardiovascular disease, and others; while only one kind of disease was ascribed to be the reason for the patient to die. The figure graphically shows the competing risks structure of the data.

Also recorded during the trial were some pretreatment covariates, including age, weight, and variables regarding to the health situation of the patients. This dataset has been collected in Andrews and Herzberg (1985) and has been a classical example about competing risks survival problem arising from real-world research. Multiple groups of researchers have analyzed these data (Byar and Corle, 1977), or used the data to illustrate their proposed methods for analysis of competing risks problem (Kay (1986), Lunn and McNeil (1995), Ng and McLachlan (2003), and others).

2.5.2 Hepatitis C Virus (HCV) Infected Patients Diagnosed with Hepatocellular Carcinoma (HCC) Data

These data originate from Dr. Robert A. Fisher's National Institutes of Health/National Institute of Diabetes and Digestive and Kidney Diseases funded project "Genes related to Hepatocellular (HCC) progression in living donor and diseased donor liver transplant" (R01DK069859). HCC is a worldwide prevalent malignancy, with more than 500,000 fatalities annually (El-Serag and Mason (1999), Davila et al. (2003), and El-Serag (2002)). The major risk factor for the development of HCC is hepatitis B virus (HBV) infection (Block et al., 2003), followed by hepatitis C virus (HCV) infection. HCV has high incidence rate in the United States, with about 3 million Americans estimated to be chronically infected. Even though causative factors are known, the molecular mechanism that leads to malignant transformation of hepatocytes is not understood. In oncology, it is recognized that tumor development and progression involve multi-level genetic changes. Multiple molecular studies have shown that genomic changes accumulate during the development and progression of HCC (Marsh and Dvorchik (2003), Gross-Goupil et al. (2003), Tseng et al. (2003), and Guan et al. (2003)). Because patients with HCC arising from chronic cirrhosis

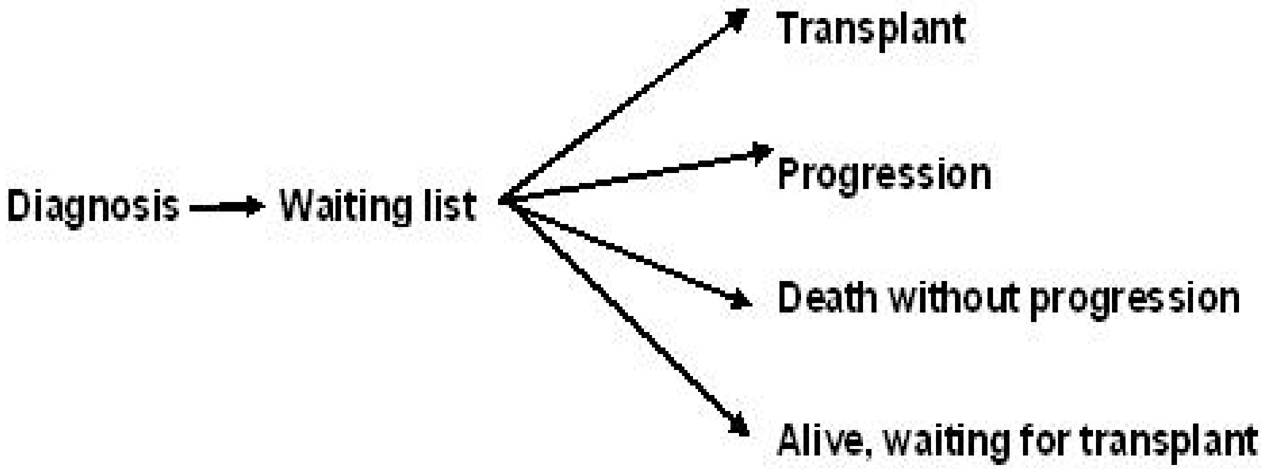


Figure 2.2: Survival Outcomes of the HCC+HCV Study

due to HCV infection essentially have a non-functioning liver, liver transplantation is the only viable treatment option. Unfortunately, there is a shortage of donor livers available compared to the number of patients on the liver transplant waitlist, so that 30% of patients will progress and be removed from the waitlist prior to an organ becoming available (Gores, 2003). Until organ availability improves, transplantation for HCC can only be offered to patients whose survival is predicted to be similar to that in patients transplanted for benign diseases. One specific aim of the funded project is to examine the genes that are implicated in tumor progression in patients with HCV and HCC while waiting for liver transplantation. After a patient is diagnosed with HCV+HCC and waitlisted for liver transplantation, the tumor may progress while the patient is on the waitlist; or liver transplantation may be performed if an appropriate donor is available before progression is observed; or the patients may die without progression, or still be waiting for transplantation. Therefore, progression, transplantation and death are competing events for the patient, and the problem can thus be described as competing risks survival data. The survival outcomes are summarized in Figure 2.2, which graphically presents the competing risks structure of the data.

Due to the shortage of donor organ supply for liver transplantations, it is of interest to explore the hypothesis that establishment of a molecular-based method for the classification of HCV+HCC patients at diagnosis may permit the differentiation between patients who will and will not have tumor progression, and thus allow a better accuracy in selecting patients for treatment cure with liver transplantation. The platform for gene expression measurement is Affymetrix HG-U133A or HG-U133A2 GeneChip microarray. The tumor tissue was biopsied from each patient after diagnosis of HCC, and hybridized to the microarray following the relevant protocol to obtain the gene expression measurements. To date the data of 46 patients have been collected, and the study is continuing with a target enrollment of 150 HCV infected patients with HCC. The anticipated progression rate among the patients is about 40%. It is of interest to identify the subset of genes that might be relevant to tumor progression and thus may be potential markers for prognosis.

Chapter 3

Review of Penalized Regression Model for Variable Selection

In this chapter, we formally introduce penalized regression models which have been found to be useful in improving prediction accuracy of model parameter estimates. With an appropriate choice of the penalty, this approach can effectively shrink the parameter estimates such that some estimates are shrunken to be exactly zero. Therefore this method can be used for variable selection without undertaking a forward, backward, or best subset variable selection procedure. In Section 3.1, the definition of penalized linear regression model is introduced. In Section 3.2, a specific penalized regression model based on the L_1 norm of the coefficients, also known as the "lasso" method is introduced. In Section 3.3, the lasso method applied to Cox proportional hazards model for survival data analysis is introduced. Finally, an algorithm specifically proposed for estimation in lasso models is reviewed in Section 3.4.

3.1 Introduction of Penalized Regression Model

Consider the linear regression model

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i = \beta_0 + \underline{x}_i^{'}\underline{\beta} + \epsilon_i \quad , \tag{3.1}$$

where $i = 1, 2, \cdots, n$ indexes the observations, y_i is a continous response for the ith observation, $\underline{x}_i = (x_{1i}, x_{2i}, \cdots, x_{ki})^{'}$ is the vector of k explanatory variables for the ith observation, β_0 is the intercept term, $\underline{\beta} = (\beta_1, \beta_2, \cdots, \beta_k)^{'}$ is the vector of coefficients corresponding to the k explanatory variables, and $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$ are i.i.d random variables with mean 0 and variance σ^2. Equation (3.1) can be more compactly expressed using matrix notation, where the model is written as

$$\underline{y} = X\underline{\beta} + \underline{\epsilon} \quad ,$$

where $\underline{y}$ is the vector of response, X is the $n \times (k+1)$ design matrix whose first column are all 1's (corresponding to the intercept term) and the remaining k columns are the k observed explanatory variables for the n observations, $\underline{\beta}$ is the vector of coefficients, and $\underline{\epsilon}$ is the vector of random errors. Without loss of generality, we assume the explanatory variables are standardized so that the mean and variance of each variable are 0 and 1, respectively. The ordinary least squares (OLS) estimates of $\underline{\beta}$ are solutions of

$$\min \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} x_{i,j}\beta_j)^2 \quad , \tag{3.2}$$

which is equivalent to seeking the solutions to the normal equation $X^{'}X\underline{\beta} = X^{'}\underline{y}$. The OLS estimator of the coefficients are also the maximum likelihood estimator (MLE) if in model (3.1) we assume the random errors ϵ_i $(i = 1, 2, \cdots, n)$ are normally distributed. The OLS estimates are unbiased estimators but may suffer from the problem of large variance, especially when the explanatory variables are correlated.

The penalized regression model is

$$\begin{aligned} \hat{\underline{\beta}} \quad &= \quad \arg\min_{\underline{\beta}} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} x_{j,i}\beta_j)^2 \qquad (3.3) \\ &\text{such that} \quad \sum_{j=1}^{k} |\beta_j|^{\gamma} \leq s \quad , \end{aligned}$$

that is, a constraint on the L_γ norm of the coefficients is applied on the OLS model to shrink the coefficient estimates. The constraining parameter (or tuning parameter), s, if allowed to tend to infinity, results in the OLS model. For all s, the solution for β_0 is $\hat{\beta}_0 = \sum_{i=1}^{n} y_i/n = \bar{y}$, so without loss of generality, we can center the outcome so that hereafter we assume that $\underline{y} = 0$ and hence omit the intercept term β_0.

Equivalently, with Lagrange multiplier theory, Model (3.3) can be expressed as

$$\hat{\underline{\beta}} = \arg\min_{\underline{\beta}} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{k} x_{j,i}\beta_j)^2 + \lambda \sum_{j=1}^{k} |\beta_j|^{\gamma} \quad , \qquad (3.4)$$

where $\gamma > 0$ and λ is the tuning parameter (corresponding to s in Model (3.3) though not equal to s) whose value determines the magnitude of penalty on the sum of squared errors. The penalty term $\sum_{j=1}^{k} |\beta_j|^{\gamma}$, which is the γ-norm of the coefficients, can be generalized to other forms of functions of the coefficients to achieve certain purposes. The estimators from the penalized regression model (3.4) were called Bridge estimators in Frank and Friedman (1993) where they were introduced as a generalization of the well known ridge regression (Knight and Fu, 2000). Ridge regression is a popular procedure for "combating multicollinearity in linear regression models" (Myers, 1990). It is often known to be the procedure that introduces a little bias into the diagonal of $X'X$ in order to reduce the large variance of the parameter estimates. That is, ridge estimators are the solution of

$$(X'X + dI)\underline{\beta} = X'\underline{y} \qquad (3.5)$$

where d is like the tuning paramter λ in Model (3.4). The choice of d often is taken to be the value where stability of the coefficient estimates shows up on the plot of ridge trace (Hoerl and Kennard, 1970).

Another aspect of ridge regression is that it, in fact, is the penalized regression model based on L_2 norm of the coefficients, that is, Model (3.3) or Model (3.4) with $\gamma = 2$. This can be shown if we keep in mind that the normal equation is obtained by differentiation of the equation $(\underline{y} - X\underline{\beta})'(\underline{y} - X\underline{\beta})$, which is matrix notation for the function in Equation (3.2). If we reversely integrate both sides of Equation (3.5) with respect of $\underline{\beta}$, then we can obtain the equivalent L_2 penalized regression model having the form in (3.4). Although the L_2 penalized estimators (i.e. the ridge estimators) are biased, compared to OLS estimators, they may have smaller variance and thus may be better estimates in terms of the criterion of prediction accuracy.

Knight and Fu (2000) established the asymptotic properties of the penalized linear regression estimators under different situations of the penalty term ($0 < \gamma \leq 1$, or $\gamma > 1$). When $\gamma \leq 1$, "the limiting distribution of the penalized estimators suggests that the estimates of truly 0 coefficients are shrunken to be exactly 0 with positive probability"(Knight and Fu, 2000). With this property, the model with $\gamma = 1$ is especially attractive as the optimization problem in Model (3.4) remains to be a convex problem. In fact, Tibshirani (1996) proposed the "lasso" model (Least Absolute Shrinkage and Selection Operator) which essentially is the L_1 penalized linear regression model, and showed that the lasso model, as ridge regression, can yield better prediction accuracy compared to OLS estimators. Another attractive advantage of the lasso model is that it shrinks the coefficient estimates and some are shrunken to be exactly 0, which indicates its usefulness as a method for identifying the subset of variables that are significantly associated with the response. In the following sections, we will present additional details about the lasso model and its function as a variable selection method.

3.2 L1 Penalized Regression Model - Lasso

3.2.1 Estimation Shrinkage by Lasso

The lasso model by Tibshirani (1996) is

$$\begin{aligned} \underline{\hat{\beta}} &= \arg\min_{\underline{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2 \right\} \qquad (3.6) \\ \text{st.} & \sum_{j=1}^{k} |\beta_j| \leq s \end{aligned}$$

The tuning parameter $s \geq 0$ controls the amount of shrinkage that is applied to the estimates. If we let $\hat{\beta}_j^0$ represent the OLS estimate of β_j $(j = 1, 2, \cdots, k)$, then the tuning parameter that will result in no shringkage is given by $s_0 = \sum_{j=1}^{k} |\hat{\beta}_j^0|$. Model (3.6) with $s < s_0$ will cause shrinkage of the coefficient estimates towards 0, and some coefficient estimates may be exactly equal to 0. For example, if we let $s = s_0/2$, approximately half of the coefficient estimates will be shrunken to 0, which is a convenient way of identifying the best subset of variables of size $k/2$.

Ridge regression which uses the penalty $\sum_{j=1}^{k} \beta_j^2 \leq s$ also shrinks the coefficient estimates, however, the coefficient estimates often are not shrunken to be exactly 0. The rationale behind the different effect between the L_1 penalty and the L_2 penalty was geometrically explained by Tibshirani (1996) for a scenario that contains two explanatory variables, using the plot shown in Figure 3.1. The objective function $\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2$ in the lasso model (3.6) equals the quadratic function (in matrix notation)

$$(\underline{\beta} - \underline{\hat{\beta}}^0)' X' X (\underline{\beta} - \underline{\hat{\beta}}^0)$$

plus a constant, where $\underline{\hat{\beta}}^0$ is the OLS estimates of $\underline{\beta}$. The elliptical contours of this quadratic function are shown in Figure 3.1 (a) and (b) by the full curves and they are centered at the

OLS estimates $\underline{\hat{\beta}}^0$. The constraint region in lasso, determined by $\sum_{j=1}^{k} |\beta_j| \leq s$, is the rotated square in Figure 3.1 (a). Therefore, the lasso solution is the first place where the contours hit the square, and this will sometimes occur at a corner, yielding a zero coefficient estimate. The constraint region in ridge regression, determined by $\sum_{j=1}^{k} \beta_j^2 \leq s$, as shown in Figure 3.1 (b), is a circle with no corners for the contours to touch, and therefore zero solutions will rarely occur.

Another way to understand the shrinkage effect of the penalty is from the Bayes point of view. As we mentioned earlier, Model (3.6) is equivalent to (Murray et al. 1981)

$$\underline{\hat{\beta}} = \arg\min \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2 \right\} + \lambda \sum_{j=1}^{k} |\beta_j| \quad . \tag{3.7}$$

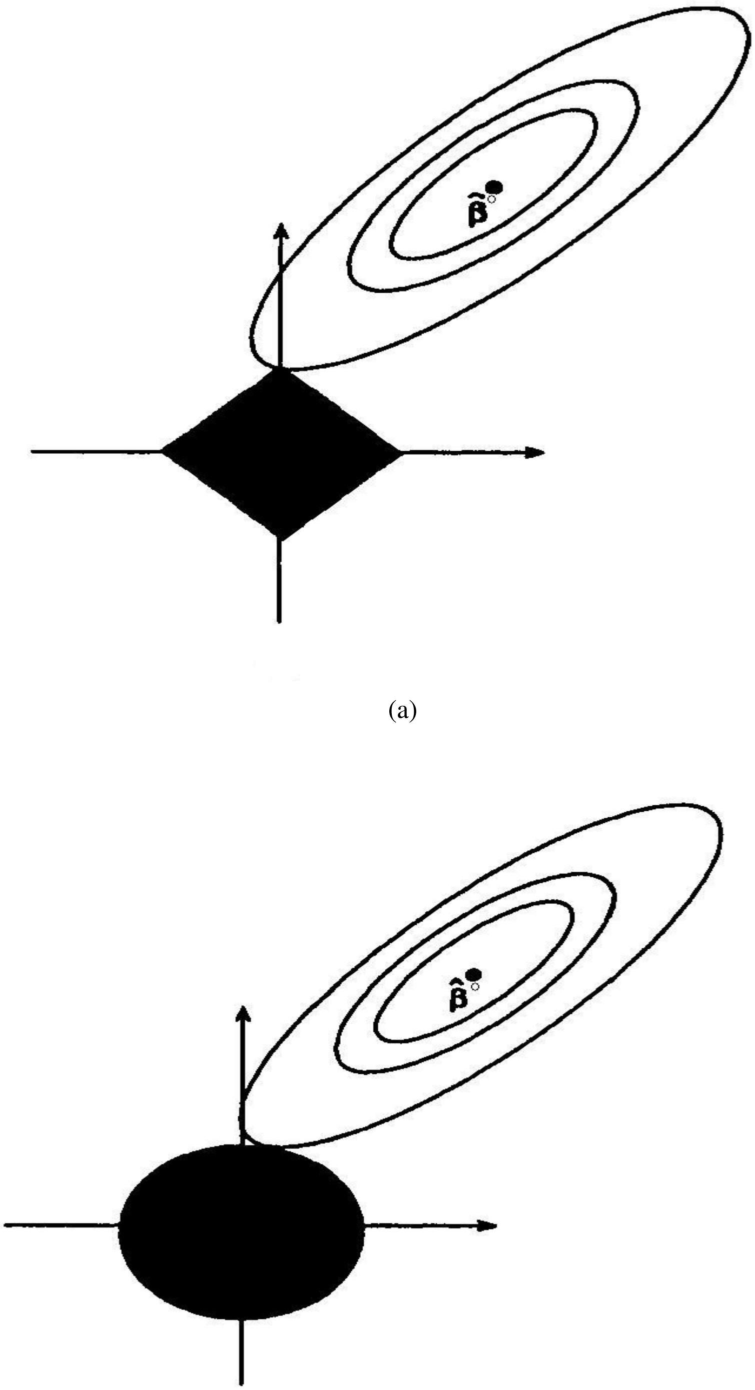
$|\beta_j|$ is actually proportional to the minus log-density of the double exponential distribution. As a result, the lasso estimate can be derived as the Bayes posterior mode under independent double-exponential priors for the β_js, where the density is

$$f(\beta_j) = \frac{1}{2\tau} \exp\left(-\frac{|\beta_j|}{\tau} \right) \quad ,$$

where $\tau = 1/\lambda$. Note that ridge regression corresponds to independent $N(0, 1/\lambda)$ priors for the β_js (refer to section 7.4.3 of Wang and Chow (1994)). The comparison between the double-exponential density and the normal density (with the same scale parameter) explains the difference in shrinkage effect between the lasso and ridge regression. Figure 3.2 shows the density curves of these two distributions. It can be seen that "the double-exponential density puts much more mass near 0 and in the tails, and this reflects the greater tendency of the lasso to produce estimates that are either large or exactly 0" (Tibshirani, 1996).

3.2.2 Asymptotics of Lasso Linear Regression Model

Knight and Fu (2000) studied the asymptotic properties for lasso type estimators. Their



(a)

(b)

Figure 3.1: Geometric explanation of estimation for (a) lasso linear model (b) ridge regression (Reproduced from Figure 2 in Tibshirani (1996))

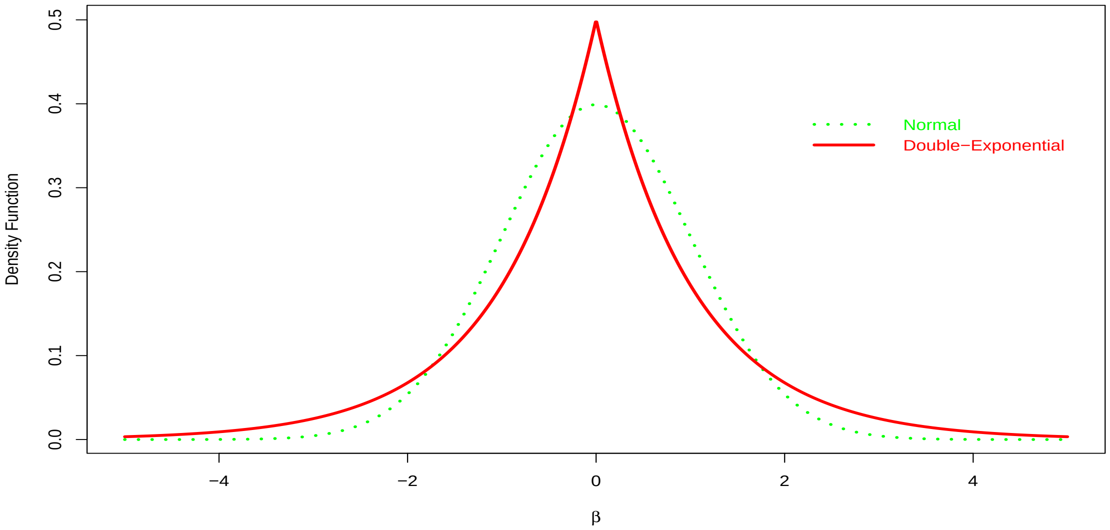


Figure 3.2: Density curves of double-exponential and normal distribution (scale parameter is 1) (Reproduced from Figure 7 in Tibshirani (1996))

discussions and conclusions are established for the general penalized least squares model (Model (3.4)) covering different kinds of penalty terms (in terms of the value of γ). A distinguishing feature of Model (3.4) when $\gamma \leq 1$ is the possibility of obtaining exact zero parameter estimates. When $\gamma < 1$, however, the objective function in (3.4) is not convex and thus it is computationally challenging to obtain the estimates, especially when the number of coefficients k is large. "There may be multiple local minima of the objective function where it is nondifferentiable" (Knight and Fu, 2000). Therefore, in this subsection, we explicitly focus on the properties for the lasso linear regression model (Model (3.7)) discussed in the earlier subsection.

Following the notation in Model (3.7), we denote the tuning parameter λ as λ_n as it is of interest to discuss the asymptotic property of the model. Further we assume the following regularity conditions,

$$D_n = \frac{1}{n}\sum_{i=1}^{n} \underline{x}_i \underline{x}_i^{'} \rightarrow D \quad , \tag{3.8}$$

where $\underline{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ik})'$ is the vector of explanatory variables observed on the ith sample. D is a nonnegative definite matrix and

$$\frac{1}{n} \max_{1 \le i \le n} \underline{x}_i' \underline{x}_i \to 0 \quad . \tag{3.9}$$

Normally in practice, the explanatory variables can be standardized so that the diagonal elements of D_n (and hence the diagonal elements of D) are all 1. The conditions in (3.8) and (3.9) ensure that D_n stabilizes at a constant when $n \to \infty$. We will also assume D is nonsingular, that is, the design matrix X is of full column rank. The following two theorems state the consistency and the limiting distribution of the lasso estimator $\hat{\underline{\beta}}_n$. The proofs follow the proofs in Section 2 of Knight and Fu (2000) with additional details.

THEOREM 3.1. *If $\lambda_n/n \to \lambda_0 \ge 0$, then $\hat{\underline{\beta}}_n \xrightarrow{P} \underline{\beta}_0$, where $\underline{\beta}_0$ denotes the vector of true parameters.*

Proof: Consider

$$Z_n(\underline{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \underline{x}_i' \underline{\beta}\right)^2 + \frac{\lambda_n}{n} \sum_{j=1}^{k} |\beta_j|$$

Also let

$$Z(\underline{\beta}) = (\underline{\beta} - \underline{\beta}_0)' D (\underline{\beta} - \underline{\beta}_0) + \lambda_0 \sum_{j=1}^{k} |\beta_j|$$

Note that the unique minimizer of $Z(\underline{\beta})$ is $\underline{\beta}_0$, the vector of true parameters.

For the first part in $Z_n(\underline{\beta})$, we have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \underline{x}_i{}'\underline{\beta}\right)^2 &= \frac{1}{n}\sum_{i=1}^{n}\left[(y_i - \underline{x}_i'\underline{\beta}_0) + (\underline{x}_i'\underline{\beta}_0 - \underline{x}_i'\underline{\beta})\right]^2 \\
&= \frac{1}{n}\left\{\sum_{i=1}^{n}(y_i - \underline{x}_i'\underline{\beta}_0)^2 + (\underline{\beta}_0 - \underline{\beta})'\sum_{i=1}^{n}\underline{x}_i\underline{x}_i'(\underline{\beta}_0 - \underline{\beta}) + 2\sum_{i=1}^{n}(y_i - \underline{x}_i'\underline{\beta}_0)\underline{x}_i'(\underline{\beta}_0 - \underline{\beta})\right\} \\
&= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 + (\underline{\beta}_0 - \underline{\beta})'D_n(\underline{\beta}_0 - \underline{\beta}) + 2\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\underline{x}_i'(\underline{\beta}_0 - \underline{\beta})
\end{aligned} \tag{3.10}$$

where ϵ_i $(i = 1, 2, \cdots, n)$ are the i.i.d random errors assigned in Model (3.1), with mean 0 and variance σ^2.

By law of large numbers, $\frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 \xrightarrow{P} \sigma^2$, and $\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\underline{x}_i'(\underline{\beta}_0 - \underline{\beta}) \xrightarrow{P} 0$. Therefore, if $\lambda_n/n \to \lambda_0 \geq 0$, it follows that

$$\sup_{\underline{\beta}\in K}|Z_n(\underline{\beta}) - Z(\underline{\beta}) - \sigma^2| \xrightarrow{P} 0 \quad ,$$

where K is any compact set of the parameter space $\mathbf{R}^k$.

Then applying Corollary II.2 in Andersen and Gill (1982), the minimizer of $Z_n(\underline{\beta})$, which is our lasso estimate $\hat{\underline{\beta}}_n$ by definition, converges to the minimizer of $(Z(\underline{\beta}) - \sigma^2)$ (hence $Z(\underline{\beta})$). That is,

$$\hat{\underline{\beta}}_n \xrightarrow{P} \underline{\beta}_0 \quad .$$

□

THEOREM 3.2. *If $\lambda_n/\sqrt{n} \to 0$, then*

$$\sqrt{n}(\hat{\underline{\beta}}_n - \underline{\beta}_0) \xrightarrow{D} N(\underline{0}, \sigma^2 D^{-1}) \quad .$$

Proof: Following the steps in (3.10), we let

$$\begin{aligned} Q_n(\underline{\beta}) &= \sum_{i=1}^{n}\left(y_i - \underline{x}_i'\underline{\beta}\right)^2 + \lambda_n \sum_{j=1}^{k} |\beta_j| \\ &= \sum_{i=1}^{n} \epsilon_i^2 + n(\underline{\beta}_0 - \underline{\beta})' D_n (\underline{\beta}_0 - \underline{\beta}) + 2\sum_{i=1}^{n} \epsilon_i \underline{x}_i'(\underline{\beta}_0 - \underline{\beta}) + \lambda_n \sum_{j=1}^{k} |\beta_j| \quad . \end{aligned}$$

Let $\underline{u} = \sqrt{n}(\underline{\beta} - \underline{\beta}_0)$, then

$$Q_n(\underline{\beta}) = \sum_{i=1}^{n} \epsilon_i^2 + \underline{u}' D_n \underline{u} - 2\left(\sum_{i=1}^{n} \epsilon_i \underline{u}' \underline{x}_i\right) / \sqrt{n} + \lambda_n \sum_{j=1}^{k} |\beta_j| \quad .$$

Note that $\underline{u}' D_n \underline{u} \rightarrow \underline{u}' D \underline{u}$, and by central limit theorem (Lindeberg-Feller) (Lin et al., 1999),

$$-2\left(\sum_{i=1}^{n} \epsilon_i \underline{u}' \underline{x}_i\right) / \sqrt{n} \overset{D}{\rightarrow} -2\underline{u}'\underline{w} \quad , \tag{3.11}$$

where $\underline{w}$ is a random vector of length k and $\underline{w} \sim N(\underline{0}, \sigma^2 D)$. Also we have

$$\lambda_n \sum_{j=1}^{k} |\beta_j| = \frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^{k} |\sqrt{n}\beta_j + u_j| - \sqrt{n}|\beta_j| \quad . \tag{3.12}$$

If $\lambda_n / \sqrt{n} \rightarrow 0$, then (3.12)$\rightarrow 0$.

If we let

$$Q(\underline{\beta}) = \sum_{i=1}^{n} \epsilon_i^2 + \underline{u}' D \underline{u} + -2\underline{u}'\underline{w} \quad ,$$

then $Q_n(\underline{\beta}) \overset{D}{\rightarrow} Q(\underline{\beta})$.

The minimizer of $Q(\underline{\beta})$ is $\underline{u} = D\underline{w} \sim N(0, \sigma^2 D^{-1})$, and the minimizer of $Q_n(\underline{\beta})$ is $\underline{\hat{u}} = \sqrt{(n)}(\underline{\hat{\beta}} - \underline{\beta}_0)$ (by definition). Since $Q_n(\underline{\beta})$ is convex and $Q(\underline{\beta})$ has a unique minimum, it follows Geyer (1996) that

$$\arg\min(Q_n(\underline{\beta})) \overset{D}{\rightarrow} \arg\min Q(\underline{\beta}) \quad .$$

That is, $\sqrt{n}(\hat{\underline{\beta}} - \underline{\beta}_0) \overset{D}{\rightarrow} N(0, \sigma^2 D^{-1})$. □

For more discussion when the design matrix is singular and nearly singular, the reader can refer to Knight and Fu (2000).

3.2.3 Some Discussion

The OLS estimator is also the maximum likelihood estimator (MLE) if in Model (3.2) the random errors are normally distributed. Penalizing the residual sum of squares with the L_1 norm, as in the lasso model (3.7), is equivalent to penalizing the likelihood function (or log-likelihood) with the same kind of penalty. Thus the lasso linear regression model can be easily generalized to a generalized linear model with an L_1 penalty on the likelihood function to perform coefficient estimation and variable selection. For example, applying the L_1 penalty in the logistic regression model, the penalized estimates, compared to the ordinary MLE, will be shrunken and some coefficient estimates may be shrunken to be exactly 0. Interested readers can refer to Section 8 of Tibshirani (1996) for lasso in generalized regression models, where an example of L_1 penalized logistic regression was also provided.

The tuning parameter in lasso controls the magnitude of the penalty on the residual sum of squares (or the likelihood function), and in effect determines the number of coefficient estimates that will be shrunken to be exactly 0. The limiting conditions about the magnitude of λ (Model (3.7)) in the afore-presented theorems are required for the consistency and limiting distribution of the lasso estimates. However, in practice, we have to decide an appropriate value of λ (or tuning parameter) to use for calculating the coefficient estimates. In Tibshirani (1996), the author proposed to use the tuning parameter value that minimizes the prediction error as estimated by cross-validation or through linear approximation of the lasso estimate. Another method the author proposed is based on Stein's unbiased estimate of risk (Stein, 1981), where the value of tuning parameter is chosen to minimize the approximate unbiased estimate of the risks of mean-square error of the β estimates. Other criterion for evaluation of model fitness can be used to choose the λ value, such as the Akaike information criterion (AIC) (Akaike, 1973) or the Bayesian information criterion (BIC) (Schwarz, 1978). Most of these criteria are based on prediction accuracy. Leng et al. (2006) shows that prediction-accuracy-based criteria alone are not sufficient for the purpose of variable selection using lasso in linear regression problems. When the analysis purpose is variable selection, selection of an optimal tuning parameter value for variable selection procedure has yet to be examined.

Efron et al. (2004) proposed a new model selection algorithm for linear regression model called "least angle regression" (LARS). Their work also establishes the connection between the lasso linear model and LARS, which is that "a simple modification of the LARS algorithm implements the Lasso. The LARS algorithm requires only the same order of magnitude of computational effort as OLS estimation, and thus can be an efficient method for estimating the coefficient estimates in lasso linear regression model".

3.3 Lasso in Cox Model

The nature and shrinking effect in parameter estimation of the L_1 penalty (in fact $L_\gamma, \gamma \leq 1$) make it appealing as a variable selection method for other statistical settings. One important application is the lasso method for variable selection in the Cox model for survival data analysis (Tibshirani 1997). The L_1 penalty is applied on the maximization of the partial likelihood for the Cox proportional hazards model, and the some of the parameter estimates will be estimated exactly as 0, and thus realize the function of variable selection.

Following the set up for survival data in Section 1.4, we consider the commonly-used exponential form of the proportional hazards model shown in Equation (1.13), that is, the hazard is modeled as

$$h_i(t) = \exp(\beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki}) h_0(t) = \exp(\underline{x}_i^{'} \underline{\beta}) h_0(t) \,, \tag{3.13}$$

where $h_0(t)$ is the baseline hazard and not necessarily be specified.

Usually the coefficients $\underline{\beta} = (\beta_1, \beta_2, \cdots, \beta_k)^{'}$ corresponding to the k explanatory variables can be estimated through maximization of the log partial likelihood (or minimization of the minus log partial likelihood function) (refer to subsection 1.4.2)

$$l(\underline{\beta}) = \log L(\underline{\beta}) = \sum_{i=1}^{n} \delta_i \left\{ \underline{x}_i^{'} \underline{\beta} - \log \sum_{l \in R(t_i)} \exp(\underline{x}_l^{'} \underline{\beta}) \right\} \,. \tag{3.14}$$

Without loss of generality, we assume that the explanatory variables are standardized so that the mean and standard deviation of each variable are 0 and 1, respectively.

The L_1 penalized Cox model is (Tibshirani 1997)

$$\begin{aligned} \hat{\underline{\beta}} &= \arg\min \left\{ -l(\underline{\beta}) \right\} \\ \text{st.} & \sum_{j=1}^{k} |\beta_j| \leq s \quad , \end{aligned} \tag{3.15}$$

where $s > 0$ is a user-specified tuning parameter. Suppose $\hat{\underline{\beta}}^0 = (\hat{\beta}_1^0, \hat{\beta}_2^0, \cdots, \hat{\beta}_k^0)$ denote the maximizers of the log partial likelihood (3.14). Then if $s \geq \sum_{j=1}^{k} |\hat{\beta}_j^0|$, the solutions to (3.15) are the usual maximum partial likelihood estimates. If $s < \sum_{j=1}^{k} |\hat{\beta}_j^0|$, however, the solutions to (3.15) are shrunken toward zero. As in the linear regression model setting, an attractive feature of the constraint based on L_1 norm of the coefficients is that often some of the estimated coefficients are exactly zero, and therefore Model (3.15) can be used for variable selection. In addition, compared to stepwise and best subset selection procedures where variables enter (or leave) the model discretely, the L_1 penalty is constraining all the variables simultaneously, and thus variable selection is performed in a more smooth manner.

Note that the adoption of the exponential form of the relative risk in (3.13) guarantees that the objective function in Model (3.15) is still convex, which makes easier the computation to seeking for solutions of the optimization problem.

Equivalently, using Lagrange multiplier method for the constrained optimization problem in (3.15), the L_1 penalized Cox model can be written as

$$\hat{\underline{\beta}} = \arg\min\left\{-l(\underline{\beta})\right\} + \lambda \sum_{j=1}^{k} |\beta_j| \quad , \tag{3.16}$$

where $\lambda \geq 0$ is the tuning parameter (as s in (3.15) but not equal to s) that controls the magnitude of penalty and thus determines the number of nonzero coefficient estimates. When $\lambda = 0$, the solutions to (3.15) are just the ordinary maximum partial likelihood estimates.

Tibshirani (1997) performed a simulation study and confirmed that the lasso Cox model can better pick up those truly 0 coefficients (i.e., their lasso estimates are 0) in comparison to a stepwise selection procedure. Fan and Li (2002), using counting process theory, established the asymptotic properties of penalized Cox proportional hazards models which encompass the lasso model based on L_1 penalty. Interested Readers can refer to Tibshirani (1997) for more details about the lasso Cox model and two real applications of the method. Another example of applying this model for variable selection can be found in Gui and Li (2005), where the authors performed a penalized Cox regression analysis on the high-dimensional gene expression microarray data for selecting the subset of genes associated with survival phenotype.

3.4 L1 Regularization Path Algorithm for the Calculation of the Lasso Estimates

So far we have not thoroughly discussed how to compute the lasso estimates (either in linear regression or Cox survival model). We briefly mentioned the LARS algorithm (Efron et al. 2003) that can be used for estimation in lasso linear regression model. Tibshirani (1996, 1997) proposed some algorithms for estimation, however, these algorithms are only applicable when $X^{'}X$ (X is the design matrix) is non-singular. When the number of observations n is smaller than the number of explanatory variables k, the design matrix is not of full column rank, and thus those algorithms cannot be applied. Here we will specifically review the L_1 regularization path algorithm for generalized linear models (including the lasso Cox model) by Park and Hastie (2007), which can be used for high-dimensional problems where $n \leq k$.

The optimization problem in Park and Hastie (2007) is

$$\hat{\underline{\beta}}(\lambda) = \arg\min_{\underline{\beta}} \left\{ - \log L(\underline{\beta}) + \lambda \|\underline{\beta}\|_1 \right\} \quad , \tag{3.17}$$

where $L(\underline{\beta})$ can be the likelihood function for the linear model in (3.1), or the likelihood function for a generalized linear model, or the partial likelihood for the Cox proportional hazards model in (3.13); and $\|\underline{\beta}\|_1 = \sum_{j=1}^{k} |\beta_j|$ is the L_1 norm of the coefficient vector.

We illustrate the algorithm using the generalized linear model set-up. Suppose the response variable y follows a distribution in the exponential family with mean μ =E(Y) and variance V =Var(Y). The generalized linear model is modeling the mean of Y linearly dependent on the explanatory variables through a link function $g(\cdot)$:

$$\eta = g(\mu) = \beta_0 + \underline{x}^{'}\underline{\beta} \quad .$$

Then the likelihood of Y can be expressed through the natural parameter $\theta = \theta(\underline{\beta})$ as (McCullagh and Nelder, 1999):

$$L(\underline{y};\underline{\theta},\phi) = \exp\left\{([y\theta - b(\theta)]/a(\phi) + c(\underline{y},\phi)\right\} \quad ,$$

where ϕ is the dispersion parameter of the distribution of Y and is assumed to be known, and the functions $a(\cdot)$, $b(\cdot)$, and $c(\cdot)$ are dependent on the distribution of Y. Then Model (3.17) is

$$l(\underline{\beta},\lambda) = -\sum_{i=1}^{n}\left\{y_i\theta(\underline{\beta})_i - b(\theta(\underline{\beta}))_i\right\} + \lambda\|\underline{\beta}\|_1 \tag{3.18}$$

For a given λ, (3.18) is a convex function of $\underline{\beta}$, and $\hat{\underline{\beta}}(\lambda)$ which minimizes $l(\underline{\beta},\lambda)$ should be unique. To seek for the solutions $\hat{\underline{\beta}}(\lambda)$, we can differentiate $l(\underline{\beta},\lambda)$ with respect of $\underline{\beta}$ and let the derivative equal 0, that is, let

$$H(\underline{\beta},\lambda) = \underline{0} \quad , \tag{3.19}$$

where

$$H(\underline{\beta},\lambda) = \frac{\partial l(\underline{\beta},\lambda)}{\partial \underline{\beta}} = -X^{'}W(\underline{y}-\mu)\frac{\partial \eta}{\partial \mu} + \lambda \mathrm{sgn}(0,\beta_1,\beta_2,\cdots,\beta_k)^{'} \quad ,$$

where X is the design matrix of dimension $n\times(k+1)$ (including the first column of 1's), W is a diagonal matrix of dimension $n\times n$ and the ith diagonal element is $V_i^{-1}\left(\frac{\partial\mu}{\partial\eta}\right)_i^2$, and $(\underline{y}-\mu)\frac{\partial\eta}{\partial\mu}$ is a vector of length n with the ith ($i = 1, 2, \cdots, n$) element being $(y_i - \mu)\left(\frac{\partial\eta}{\partial\mu}\right)_i$. Solutions to Equation (3.19) can be calculated using iterative procedures, such as the Newton Raphson algorithm (Agresti, 2003).

The goal of the L_1 regularization algorithm (Park and Hastie, 2007) is to "compute the entire solution path for the coefficients $\underline{\beta}$ with λ varying from ∞ to 0". The authors showed that when λ exceeds a threshold (refer to their Lemma 2.1), i.e. when the penalty is too large, all of the explanatory variables' coefficient estimates will be zero and only the estimate of the intercept term β_0 is non-zero. This threshold thus is used to initialize the computation in their algorithm. When λ decreases from the starting value, some variables will join the active set (the set of variables whose coefficient estimates are non-zero). The estimation of coefficients at each value of λ is performed with a "predictor" step and a "corrector" step. And the values of λ at which the estimations are conducted are the values where changes, in comparison to the active set estimated at the immediate prior value of λ, of the active set happen. The authors summarized the computation at each iteration (say the mth) into four steps:

1. Step length: determine the step length for decrement of λ. Given the current value of λ_m, the approximate step length to λ_{m+1} at which the active set of variables would change is calculated using the weighted LARS algorithm (Efron et al. 2003). Briefly, the correlation between each variable and the residuals at the current iteration can be expressed as a function of λ; thus the change of λ will change the correlations of the variables. The minimum change of λ, such that there is at least one variable from the non-active set whose absolute correlation is changed to be as those in the active set (and thus this variable would enter the active set), hence can be obtained and serves as the step length for the next iteration.

2. Predictor step: with the updated λ value λ_{m+1}, linearly approximate the corresponding change in the coefficient estimates $\hat{\underline{\beta}}^{m}$ for the variables in the active set, and thus obtain the updated estimates corresponding to λ_{m+1}, denote them as $\hat{\underline{\beta}}^{m+}$. Note the estimation of $\underline{\beta}$ is dependent on λ, meaning $\underline{\beta}$ is a function of λ. Using Taylor expansion to linearly approximate this function, the change of λ from λ_m to λ_{m+1} will yield the updated coefficient estimates $\hat{\underline{\beta}}^{m+}$.

3. Corrector step: using $\hat{\underline{\beta}}^{m+}$ as the starting value, find the exact solution of coefficients

at $\lambda = \lambda_{m+1}$ for the variables in the active set, and denote it as $\hat{\underline{\beta}}^{m+1}$. This step is essentially solving Equation (3.19) with λ_{m+1} using the variables in the current active set. As mentioned earlier, any iterative procedure for solving non-linear equations can be applied. The $\hat{\underline{\beta}}^{m+}$ from the corrector step normally is close to the exact solution and thus provides an efficient start for the calculation of $\hat{\underline{\beta}}^{m+1}$.

4. Active set: check to see whether the active set of variables has been changed with the updated estimates $\hat{\underline{\beta}}^{m+1}$. For each variable not in the active set, use condition (13) of Park and Hastie (2007) to judge whether this variable should be added into the active set. If the active set is modified, repeat the corrector step with the variables in the newly updated active set and obtain their estimates. If the new estimate of a variable in the updated active set becomes 0, then this variable will be removed from the active set.

The algorithm iterates with the above four steps at each iteration until λ reaches 0. From the description of the steps, we can see that the algorithm does not perform parameter estimation with all the variables simultaneously in the model, rather it only attempts to estimate the variables joined in the active set (which means the coefficient estimates for the variables not in the active set are all zero). Therefore, the algorithm can be applied to problems where $n \leq k$.

When the explanatory variables are highly correlated, the diagonals of $(X'X)^{-1}$ will be large, and hence the coefficient estimates from the L_1 penalized model (3.17) will be highly unstable. Park and Hastie (2007) proposed to add a small fixed penalty based on the L_2 norm of the coefficients, that is, they actually seek the solution of

$$\hat{\underline{\beta}}(\lambda) = \arg\min_{\underline{\beta}} \left\{ -\log L(\underline{\beta}) + \lambda ||\underline{\beta}||_1 + \frac{s_0}{2} ||\underline{\beta}||_2^2 \right\} \quad ,$$

where s_0 is a small positive constant (default in their algorithm is 1e-5). The effect of this fixed L_2 penalty is as that in ridge regression, to reduce the variability of parameter estimates when the variables are correlated. When the correlations are small, the effect of the very small quadratic penalty is trivial.

The rationale of the algorithm for L_1 penalized Cox model is the same as that for the generalized linear model, and reader can refer to the appendix of Park and Hastie (2007) for more details.

The authors implemented their algorithm as an R package “glmpath” and made it publicly available for downloading. The availability of this software makes the algorithm especially appealing to use, and is also the main reason that we adopted this algorithm in this thesis work.

Chapter 4

Model Development for Variable Selection in Competing Risks

We have reviewed in Chapter 1 the background material pertaining to survival analysis and introduced some special situations where multivariate survival data arise. In Chapter 2, we exclusively discussed the *competing risks* multivariate survival data, including the probabilistic description of the data and the modeling approaches for studying the association between explanatory variables and the survivals of different failure types (or risks). The two examples, the prostatic cancer data and the HCC data, exemplify the competing risks problems that arise in real research. From these two examples, especially the HCC study, we see the need for selecting the subset of explanatory variables (the genes in the HCC study) that are influential on the survival times corresponding to specific failure types. Chapter 3 provides a review of penalized regression techniques, expecially the L_1 penalized model which can be used for variable selection in different settings, such as linear regression and Cox proportional hazards models. In this chapter, we start Section 4.1 by explicitly stating the problem of interest in this thesis. In Section 4.2, we propose our model for solving the problem, followed by the estimation algorithm and asymptotic properties of the model. Numerical simulations are used to evaluate the proposed model, and the simulation mechanism and results are described in Section 4.3.

4.1 The Problem of Interest

Assume there are p failure types (or risks). Each individual is subject to failure from these p failure types, though only one type of failure is observed for each individual. That is, we have competing risks survival data. Also measured are k explanatory variables at the time origin on each individual. The problem of interest is **to identify the subset of explanatory variables that are significantly associated with each failure type or some specific failure types**.

4.2 The Proposed Model

Let T denote the variable for the observed time, which is the length of time from the pre-defined time of origin until failure or censoring; and let C be the discrete random variable for the failure types. Suppose there are n individuals, and $T = t_i$ is the observed time on the ith individual. The failure can be of p types, labeled as $1, 2, \cdots, p$; and let $C = c_i$ denote the failure type observed on the ith individual. Let d_i be the censoring indicator for the ith individual where $d_i = 0$ if the individual is censored, and $d_i = 1$ if the failure is observed. Further suppose there are k explanatory variables, and let $\underline{x}$ be the vector of length k for the explanatory variables. Then $\underline{x}_i$ is the vector of explanatory variables observed on the ith individual.

4.2.1 The Proposed Model

The probabilistic aspect in modeling the competing risks is the joint distribution of T and C (Crowder, 2001). It is specified through the sub-survivor function $S(j, T)$ or the sub-hazard function $h(j, t)$ for $j = 1, 2, \cdots, p$, as defined in Section 2.2. This approach of modeling competing risks data does not require the assumption of independence between the different failure types. Our variable selection method is based on the proportional hazards regression model for competing risks presented in Section 2.4. For the completeness of presentation of our methodological development for variable selection, we write down the model here:

$$h(j,t;\underline{x}) = \psi_{j,\underline{x}} h_0(j,t) \text{ for } j = 1,2,\cdots,p\ ,$$

where $h_0(j,t)$ is an unspecified baseline sub-hazard corresponding to failure type j, $\psi_{j,\underline{x}} = \exp(\underline{x}'\underline{\beta}^j)$ is the function through which the effect of the explanatory variables on the sub-hazards is specified, $\underline{\beta}^j = (\beta^j_1,\beta^j_2,\cdots,\beta^j_k)'$ is the $k \times 1$ vector of coefficients of the explanatory variables corresponding to failure type j. Thus

$$\underline{\beta} = \left(\beta^1_1,\beta^1_2,\cdots,\beta^1_k\ ;\ \beta^2_1,\beta^2_2,\cdots,\beta^2_k\ ;\cdots\ ;\ \beta^p_1,\beta^p_2,\cdots,\beta^p_k\right)$$

is the overall vector of coefficients in the model based on data of all p types of failure and needs to be estimated. The baseline sub-hazard $h_0(j,t)$ for cause j is not required to be proportional to the baseline sub-hazard $h_0(j',t)$ for another cause j' where $j \neq j'$. Since the same explanatory variable may have different effects on the different types of failures, it is reasonable to assume that the $\underline{\beta}^j$, $j = 1,2,\cdots,p$, vectors are independent of each other.

Let $t_{(1)} < t_{(2)} < \cdots < t_{(m)}$ be the ordered m observed failure times on the n individuals. From the subsection 2.4.1, the overall partial likelihood function for all individuals is

$$\begin{aligned} P(\underline{\beta}^1,\underline{\beta}^2,\cdots,\underline{\beta}^p) &= \prod_{l=1}^{m} \frac{\psi_{c_l,i_l}}{\sum_{a\in R(t_{(l)})} \psi_{c_l,a}} \\ &= \prod_{l=1}^{m} \frac{\exp(\underline{x}_{i_l}'\underline{\beta}^{c_l})}{\sum_{a\in R(t_{(l)})} \exp(\underline{x}_a'\underline{\beta}^{c_l})}\ , \end{aligned} \tag{4.1}$$

where i_l is the index for the individual who fails at $t_{(l)}$, and the corresponding failure type is c_l; $R(t_{(l)})$ is the set of individuals at risk of failure type c_l at the time just prior to $t_{(l)}$. Note as in the partial likelihood of Cox model for univariate survival data, the baseline sub-hazards $h_0(j,t)$, $j = 1,2,\cdots,p$, cancel out. Since they are independent of the explanatory variables and the purpose is to evaluate the influence of the explanatory variables on the survival, the

$h_0(j,t)$, $j = 1,2,\cdots,p$, are not of interest to be estimated.

The log partial likelihood then is

$$l(\underline{\beta}^1,\underline{\beta}^2,\cdots,\underline{\beta}^p) = \log P(\underline{\beta}^1,\underline{\beta}^2,\cdots,\underline{\beta}^p) = \sum_{l=1}^{m}\left\{\underline{x}_{i_l}'\underline{\beta}^{c_l} - \log\left[\sum_{a\in R(t_{(l)})}\exp(\underline{x}_a'\underline{\beta}^{c_l})\right]\right\}.$$

Adopting the idea of penalized likelihood, specifically, the penalty based on L_1 norm of the coefficients as in the method of "least absolute shrinkage and selection operator" (lasso) by Tibshirani (1997), we propose using the following model for variable selection:

$$\begin{aligned}
(\hat{\underline{\beta}}^1,\hat{\underline{\beta}}^2,\cdots,\hat{\underline{\beta}}^p) &= \arg\max_{\underline{\beta}^j, j=1,2,\cdots,p} l(\underline{\beta}^1,\underline{\beta}^2,\cdots,\underline{\beta}^p) \qquad (4.2)\\
\text{st: } \|\underline{\beta}^1\|_1 &\leq s_1\\
\|\underline{\beta}^2\|_1 &\leq s_2\\
&\vdots\\
\|\underline{\beta}^p\|_1 &\leq s_p \quad ,
\end{aligned}$$

where $\|\underline{\beta}^j\|_1 = |\beta_1^j|+|\beta_2^j|+\cdots+|\beta_k^j|$ is the L_1 norm of the coefficients vector $\underline{\beta}^j$ that corresponds to failure type j; s_j, $j = 1,2,\cdots,p$, are tuning parameters that quantify the magnitude of the constraints on the L_1 norms of the coefficients vectors and determine the number of coefficients estimated as zero in the model. Rather than constraining all the coefficients simultaneously with one single tuning parameter, here we propose to use a different tuning parameter for each vector of coefficients corresponding to each type of failure. This is more intuitive since the influence on different failure types of the same explanatory variable can be different and we are not interested in studying the effect of explanatory variables on the overall survival without differentiating the failure types.

Model (4.2) is essentially an optimization problem with multiple constraints. Applying the Lagrange multipliers method, we obtain a model equivalent to (4.2):

$$\begin{aligned}(\underline{\hat{\beta}}^1, \underline{\hat{\beta}}^2, \cdots, \underline{\hat{\beta}}^p) &= \arg \min_{\underline{\beta}^j, j=1,2,\cdots,p} \left[-l(\underline{\beta}^1, \underline{\beta}^2, \cdots, \underline{\beta}^p) + \sum_{j=1}^{p} \lambda_j \|\underline{\beta}^j\| \right] \\ &= \arg \min_{\underline{\beta}^j, j=1,2,\cdots,p} \left[-\sum_{l=1}^{m} \left\{ \underline{x}_{i_l}{}' \underline{\beta}^{c_l} - \log \left[\sum_{a \in R(t_{(l)})} \exp\,(\underline{x}_a{}' \underline{\beta}^{c_l}) \right] \right\} + \sum_{j=1}^{p} \lambda_j \|\underline{\beta}^j\| \right] \end{aligned} \tag{4.3}$$

where λ_j, $j = 1, 2, \cdots, p$, are tuning parameters which determine the magnitude of penalty on the log partial likelihood.

When $p = 1$, i.e. there is only one failure type, as is the case in univariate survival analysis, model (4.2) reduces to the lasso model (3.15) for Cox regression as described by Tibshirani (1997). Then the L_1 regularization path algorithm for Cox model (Park and Hastie, 2007) reviewed in the subsection 3.4 can be applied directly to compute the coefficient estimates. With competing risks, $p > 1$, and we have multiple constraints in the variable selection model, which make it more complicated to estimate the model parameters. Next we will show that although the model is built upon the overall likelihood encompassing the data of all failure types, with the choice of the form of the constraints, model (4.2) can be expressed in terms of p components corresponding to the p types of failure.

4.2.2 Estimation of the Model Parameters

Recall that we have m distinct failure times over the n individuals, denoted as $t_{(1)}, t_{(2)}, \cdots, t_{(m)}$. Among the ordered m times, let $t^j_{(1)}, t^j_{(2)}, \cdots, t^j_{(m_j)}$ be the ordered failure times due to cause j, $j = 1, 2, \cdots, p$, where m_j is the number of observed failures due to cause j, and $\sum_{j=1}^{p} m_j = m$. The contribution to the overall partial likelihood (4.1) from the failures of type j is $P_j(\underline{\beta}^j) = \prod_{l=1}^{m_j} m_j \left(\psi_{j,i^j_l} / \sum_{a \in R(t^j_{(l)})} \psi_{j,a} \right)$, where i^j_l is the index of the individual who fails at $t^j_{(l)}$ (of failure type j). It is a function only of $\underline{\beta}^j$ and is independent of the coefficients corresponding to other failure types. The overall partial likelihood (4.1) then can be expressed as the product of p components corresponding to the p failure types:

$$
\begin{aligned}
P(\underline{\beta}^1, \underline{\beta}^2, \cdots, \underline{\beta}^p) &= \prod_{l=1}^{m} \frac{\psi_{c_l, i_l}}{\sum_{a \in R(t_{(l)})} \psi_{c_l, a}} \\
&= \prod_{l=1}^{m_1} \left[\frac{\psi_{1, i_l^1}}{\sum_{a \in R(t^1_{(l)})} \psi_{1,a}} \right] \times \prod_{l=1}^{m_2} \left[\frac{\psi_{2, i_l^2}}{\sum_{a \in R(t^2_{(l)})} \psi_{2,a}} \right] \times \cdots \times \prod_{l=1}^{m_p} \left[\frac{\psi_{1, i_l^p}}{\sum_{a \in R(t^p_{(l)})} \psi_{p,a}} \right] \\
&= \prod_{j=1}^{p} P_j(\underline{\beta}^j) \quad .
\end{aligned}
$$

The log partial likelihood thus is

$$
l(\underline{\beta}^1, \underline{\beta}^2, \cdots, \underline{\beta}^p) = \sum_{j=1}^{p} \log P_j(\underline{\beta}^j) \quad .
$$

Therefore, solving model (4.3) is equivalent to solving:

$$
\begin{aligned}
(\hat{\underline{\beta}}^1, \hat{\underline{\beta}}^2, \cdots, \hat{\underline{\beta}}^p) &= \arg \min_{\underline{\beta}^j, j=1,2,\cdots,p} \left[-\left(\sum_{j=1}^{p} \log P_j(\underline{\beta}^j) \right) + \sum_{j=1}^{p} \lambda_j \|\underline{\beta}^j\|_1 \right] \\
&= \arg \min_{\underline{\beta}^j, j=1,2,\cdots,p} \sum_{j=1}^{p} \left[-\log P_j(\underline{\beta}^j) + \lambda_j \|\underline{\beta}^j\|_1 \right] \quad . \qquad (4.4)
\end{aligned}
$$

Because of the assumption that $\underline{\beta}^j$, $j = 1, 2, \cdots, p$ are independent of each other, the inference from the component corresponding to failure type j, $j = 1, 2, \cdots, p$, within the summation in (4.4) does not depend on the inference from any other component within that summation. This is equivalent to the problem that simultaneously minimizes p functions $Q_j(\underline{\beta}^j)$, $j = 1, 2, \cdots, p$, where

$$
\begin{aligned}
Q_j(\underline{\beta}^j) &= -\log P_j(\underline{\beta}^j) + \lambda_j \|\underline{\beta}^j\|_1 \\
&= -\sum_{l=1}^{m_j} \left[\underline{x}_{i_l^j}{}' \underline{\beta}^j - \log \left(\sum_{a \in R(t^j_{(l)})} \exp\left(\underline{x}_a{}' \underline{\beta}^j\right) \right) \right] + \lambda_j \|\underline{\beta}^j\|_1 \quad .
\end{aligned}
$$

(4.5)

Therefore, we can seek the solutions for the p functions $Q_j(\underline{\beta}^j)$, $j = 1, 2, \cdots, p$, individually by applying any existing optimization algorithm that is available for computing the lasso estimates in the Cox model for univariate survival data, such as the L_1 regularization path algorithm for Cox model by Park and Hastie (2007) reviewed in subsection 3.4.

However, special care is needed before applying the algorithm for the Cox model for univariate survival data, since the risk set at any time point has to be clearly defined corresponding to each failure type. Recall that in univariate survival analysis, the risk set at any time t contains individuals who have not failed by t, and it is composed of two groups of individuals: those who will fail at or after t, and those who will be censored at or after t. Here in competing risks survival data, $R(t^j_{(l)})$ in (4.5) is the risk set with respect to failure type j at a time just prior to $t^j_{(l)}$ (the lth failure time due to cause j), and it is composed of three groups of individuals: the group of individuals who fail due to failure type j at or after $t^j_{(l)}$; the group of individuals who are censored at or after $t^j_{(l)}$; and the group of individuals who fail after $t^j_{(l)}$ due to failure types other than j. For failure type j specifically, the last two groups of individuals can both be considered as “censored” since their failure times of type j are not observed. Therefore, before solving solutions for $Q_j(\underline{\beta}^j)$ using an algorithm for univariate survival data, we need to manually create a binary status indicator specific to each failure type for each individual, indicating whether the individual is observed to fail of type j, or the individual is “censored” (i.e. truly censored or fail of other failure types). Then the L_1 regularization path algorithm (Park and Hastie, 2007) for Cox model with univariate survival data can be applied for solving $Q_j(\underline{\beta}^j)$ in (4.5) to obtain coefficient estimates corresponding to failure type j. For each failure type, we have to create its specific binary status indicator, though this is not difficult to do in most programming environments. Because the nature of the path-following algorithm makes it possible to solve problems with more explanatory variables than number of observations, we can also deal with competing risks data with high-dimensional explanatory variables by using this algorithm.

It has been assumed so far that one and only one failure can happen at any observed failure time. When there are ties at the observed failure times, the scenario can be differentiated into two situations. One situation is that only ties of the same type of failure can happen at an observed failure time. Then any method of approximation to the exact likelihood, such as methods by Breslow and Crowley (1974) and Efron (1977) described in subsection 1.4.2, can be used. The other situation is that ties may happen of different types. For example, two failures occur at time t, one of type j and the other of type j'. When studying the effect of the explanatory variables on hazard for failure type j, we minimize $Q_j(\underline{\beta}^j)$ in Equation 4.5, and the failure of type j' is viewed as "censored". Censored observations are often assumed to occur after all the failures, and thus there is no ambiguity when minimizing $Q_j(\underline{\beta}^j)$ (Kalbfleisch and Prentice, 2002).

4.2.3 Aymptotic Properties of the Estimators

Since the estimation of the coefficients corresponding to one failure type is independent of that for other types, and the number of failure types of interest is always fixed (and usually small), we only need to discuss the asymptotic properties of the estimates corresponding to one single failure type, which are equivalently the lasso estimates in the Cox model with univariate survival data. Therefore, without loss of generality, we are concerned with estimating the model when the number of failure types $p = 1$. The coefficient estimates are obtained by minimizing $Q(\underline{\beta})$ as the $Q_j(\underline{\beta}^j)$ in (4.5), i.e.

$$\hat{\underline{\beta}} = \arg\min_{\underline{\beta}} \left(-l(\underline{\beta}) + \lambda\|\underline{\beta}\|_1\right) = \arg\min_{\underline{\beta}} Q(\underline{\beta}) \quad , \tag{4.6}$$

where the j's indexing the failure type in the superscript and subscript are omitted. $l(\underline{\beta}) = \log P(\underline{\beta}) = \log\left[\prod_{l=1}^{m}\left(\psi_{i_l}/\sum_{a\in R(t_{(l)})}\psi_a\right)\right]$ is the log partial likelihood, where i_l is the index for the individual who fails at $t_{(l)}$, and $\psi_i = \exp(\underline{x}_i'\underline{\beta})$, and $R(t_{(l)})$ is the set of individuals at risk of failure type c_i at the time just prior to $t_{(l)}$

It will be shown in theorems that when the rate of the tuning parameter λ going to 0 satisfies certain conditions, the L_1 penalized partial likelihood estimators are consistent, and they are asymptotically normally distributed. We will denote λ as λ_n since we are interested in its convergence speed as $n \to \infty$, where n still denotes the sample size. It is implicitly assumed that the number of observed failures $m \to \infty$ when $n \to \infty$. Prior to stating the theorems, let us review the intuition about the asymptotic properties of partial likelihood estimators in general. Recall from subsection 1.4.3 that the data sequence can be formulated as $D_l = ((A_1, B_1), (A_2, B_2), \cdots, (A_l, B_l))$ for $l = 1, 2, \cdots m$. Then the likelihood function can be written as

$$\begin{aligned} L(\underline{\theta}) = f_{\underline{\theta}}(D_m) &= \prod_{l=1}^{m} f_{\underline{\theta}}(A_l, B_j | D_{l-1}) \\ &= \prod_{l=1}^{m} f_{\underline{\theta}}(A_l | D_{l-1}, B_l) \times \prod_{l=1}^{m} f_{\underline{\theta}}(B_l | D_{l-1}) \\ &= P(\underline{\beta}) \times Q(\underline{\theta}) \quad . \end{aligned}$$

Here, $P(\underline{\beta}) = \prod_{l=1}^{m} f_{\underline{\theta}}(A_l | D_{l-1}, B_l)$ is the partial likelihood. Consider the score components

$$U_l = \frac{\partial \log f(A_l | H_l; \underline{\beta})}{\partial \underline{\beta}}, \; l = 1, 2, \cdots, m \quad , \tag{4.7}$$

where $H_l = (D_{l-1}, B_l)$ is used to specify the conditioning variables for the lth term in (4.7). So the total score arising from the overall partial likelihood $P(\underline{\beta})$ is

$$U = \frac{\partial \log P(\underline{\beta})}{\partial \underline{\beta}} = \sum_{l=1}^{m} U_l \quad .$$

Conditionally on $H_l = h_l$, $f(A_l | h_l; \underline{\beta})$ is a density function. Thus, under the usual regularity conditions, we have $\mathrm{E}(U_l | H_l = h_l) = 0$. It follows that

$$\mathrm{E}(U_l) = \mathrm{E}\mathrm{E}(U_l | H_l) = 0 \quad .$$

Further, let $l < l'$, the condition $H_l = h_l$ implies that U_l is fixed. Hence, for $l < l'$,

$$E(U_l U_{l'}{}') = EE(U_l U_{l'}{}'|H_{l'}) = E[U_l E(U_{l'}|H_{l'})] = 0 \quad .$$

Therefore the score contributions $U_1, U_2, \cdots$, have mean zero and are uncorrelated (Chapter 4 of Kalbfleisch et al. 2002).

Specifically to the case of the Cox proportional hazards model, $f(A_l|h_l;\underline{\beta})$ is the conditional probability $\mathrm{P}\left(\text{individual with variables } \underline{x}_{(l)} \text{ fails at } t_{(l)}| \text{ one failure at } t_{(l)}\right) = \psi_{i_l}/\sum_{a\in R(t_{(l)})}\psi_a$. The partial likelihood is $P(\underline{\beta}) = \prod_{l=1}^{m}\left(\psi_{i_l}/\sum_{a\in R(t_{(l)})}\psi_a\right)$. Recall the notation $l(\underline{\beta}) = \log P(\underline{\beta})$, and let $l_l(\underline{\beta}) = \log\left(\psi_{i_l}/\sum_{a\in R(t_{(l)})}\psi_a\right)$ be the lth term in $l(\underline{\beta})$. We now verify that the following regularity conditions hold with the partial likelihood $P(\underline{\beta})$ from the Cox model.

ASSUMPTION 1. *$P(\underline{\beta})$ is identifiable with respect to $\underline{\beta}$, that is, $\forall \underline{\beta} \neq \underline{\beta}'$, $P(\underline{\beta}) \neq P(\underline{\beta}')$.*

ASSUMPTION 2. *Let $\underline{\beta}_0$ be the vector of true values of $\underline{\beta}$. In the neighborhood of $\underline{\beta}_0$, the first order derivative, second order derivative, and the third order derivative of the log partial likelihood $l(\underline{\beta})$ exist for all $\underline{x}$.*

In the Cox model, with $\psi_i = \exp(\underline{x}'\underline{\beta})$, it is not difficult to see that $P(\underline{\beta})$ is identifiable, and $l(\underline{\beta})$ is a differentiable convex function of $\underline{\beta}$ (Chapter 4 of Crowder (2001)) , and the first order, second order, and the third order derivatives all exist.

Let

$$\underline{u}(\underline{\beta}) = \frac{\partial l(\underline{\beta})}{\partial \underline{\beta}} \quad ,$$

which is the score statistic (a vector of length k (the number of explanatory variables)). Let $u_w(\underline{\beta}) = \partial l(\underline{\beta})/\partial \beta_w$ be the wth element of $\underline{u}(\underline{\beta})$, $w = 1, 2, \cdots, k$. Let

$$A_w = \frac{\partial^2 u_w(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}'}$$

which is a $k \times k$ matrix of the second order derivative of the score statistic $\underline{u}(\underline{\beta})$, and its (w_1, w_2) component is $\partial^2 u_w(\underline{\beta})/\partial\beta_{w_1}\partial\beta_{w_2}$, $w_1, w_2 = 1, 2, \cdots, k$.

ASSUMPTION 3. *In the neighborhood of* $\underline{\beta}_0$*,* $|A_w| \leq Z(\underline{x})$*, where* $\mathrm{E}Z(\underline{x}) < \infty$*,* $w = 1, 2, \cdots, k$*.*

This assumption requires that the values of the explanatory variables are bounded. In reality, the measurements of the explanatory variables are normally bounded, and thus this assumption holds in the Cox model with $\psi_i = \exp(\underline{x}'\underline{\beta})$.

ASSUMPTION 4. *In the neighborhood of* $\underline{\beta}_0$*,*

$$\begin{aligned} E_{\underline{\beta}_0}\left[\frac{\partial P(\underline{\beta})/\partial\beta_w}{P(\underline{\beta})}\right] &= 0 \\ E_{\underline{\beta}_0}\left[\frac{\partial^2 P(\underline{\beta})/\partial\beta_w\partial\beta_{w'}}{P(\underline{\beta})}\right] &= 0 \quad . \end{aligned}$$

This assumption basically requires that the operations of integration and differentiation are exchangeable. This is not a problem for the partial likelihood $P(\underline{\beta})$ in the Cox model with $\psi_i = \exp(\underline{x}'\underline{\beta})$, since $P(\underline{\beta})$ and its first order derivative are continuous with respect to both $\underline{x}$ and $\underline{\beta}$. Therefore,

$$\mathrm{E}_{\underline{\beta}_0}\left[\frac{\partial P(\underline{\beta})/\partial\beta_w}{P(\underline{\beta})}\right] = \frac{\partial \mathrm{E}_{\underline{\beta}_0}\left(P(\underline{\beta})/P(\underline{\beta})\right)}{\partial\beta_w} = \frac{\partial 1}{\partial\beta_w} = 0 \quad .$$

Recall that $l_l(\underline{\beta}) = \log\left(\psi_{i_l}/\sum_{a\in R(t_{(l)})}\psi_a\right)$ denote the lth term in the log partial likelihood, that is, $l(\underline{\beta}) = \sum_{l=1}^{m} l_l(\underline{\beta})$.

ASSUMPTION 5. *The information matrix is positive-definite in the neighborhood of* $\underline{\beta}_0$*. That is,*

$$I_l(\underline{\beta}_0) = E_{\underline{\beta}_0} - \left[\left(\frac{\partial l_l(\underline{\beta})}{\partial\underline{\beta}}\right)\left(\frac{\partial l_l(\underline{\beta})}{\partial\underline{\beta}}\right)'\right] > 0 \quad .$$

And the total information matrix $I(\underline{\beta}_0) = \sum_{l=1}^{m} I_l(\underline{\beta}_0)$ approaches infinity at the rate of $O(m)$ when $m \rightarrow \infty$.

THEOREM 4.1. *Assume that $(\underline{x}_1, t_1, d_1), (\underline{x}_2, t_2, d_2), \cdots, (\underline{x}_n, t_n, d_n)$ are independent and identically distributed according to the population $(\underline{x}, T, D)$, where $\underline{x}$ is the vector of explanatory variables, T is the survival time, D is the variable for censoring status, and T and D are conditionally independent given $\underline{x}$. Assume that the regularity conditions in Assumption 1-5 hold, then for model (4.6), if when $n \rightarrow \infty$, the number of failures $m \rightarrow \infty$, and the tuning parameter $\lambda_n/n \rightarrow 0$, then there exists a local minimizer of $Q(\underline{\beta})$, that is, $\hat{\underline{\beta}}$ exists and it is consistent for $\underline{\beta}_0$.*

Proof: We have

$$Q(\underline{\beta}) = -l(\underline{\beta}) + \lambda_n \|\underline{\beta}\|_1 \quad ,$$

where $\|\underline{\beta}\|_1 = \sum_{w=1}^{k} \beta_w$ is the L_1 norm of the vector of coefficients.

We want to show that in a close neighborhood $\underline{\delta}$ of $\underline{\beta}_0$, where $\|\underline{\delta}\|_1 \leq \delta_0$ a small constant, we have,

$$\frac{1}{n}\left(-Q(\underline{\beta}_0 + \underline{\delta}) - (-Q(\underline{\beta}_0))\right) \rightarrow 0 \quad , \tag{4.8}$$

then there exists a local maximizer of $-Q(\underline{\beta})$ near $\underline{\beta}_0$, which is the minimizer of $Q(\underline{\beta})$, denoted by $\hat{\underline{\beta}}$, and $\hat{\underline{\beta}} \rightarrow \underline{\beta}_0$.

$$\begin{aligned}
& \frac{1}{n}\left(-Q(\underline{\beta}_0 + \underline{\delta}) + Q(\underline{\beta}_0)\right) \\
= \; & \frac{1}{n}\left[l(\underline{\beta}_0 + \underline{\delta}) - \lambda_n \|\underline{\beta}_0 + \underline{\delta}\|_1 - \left(l(\underline{\beta}_0) - \lambda_n \|\underline{\beta}_0\|_1\right)\right] \\
= \; & \frac{1}{n}\left[\left(l(\underline{\beta}_0 + \underline{\delta}) - l(\underline{\beta}_0)\right) + \left(-\lambda_n \|\underline{\beta}_0 + \underline{\delta}\|_1 + \lambda_n \|\underline{\beta}_0\|_1\right)\right] \\
\leq \; & \frac{1}{n}\left(l(\underline{\beta}_0 + \underline{\delta}) - l(\underline{\beta}_0)\right) + \frac{1}{n}\lambda_n \left| \|\underline{\beta}_0 + \underline{\delta}\|_1 - \|\underline{\beta}_0\|_1 \right| \quad .
\end{aligned}$$

Consider the Taylor approximation of $l(\underline{\beta}_0 + \underline{\delta})$ at $\underline{\beta}_0$,

$$l(\underline{\beta}_0 + \underline{\delta}) = l(\underline{\beta}_0) + \frac{\partial l(\underline{\beta})}{\partial \underline{\beta}'} |_{\underline{\beta}_0} \underline{\delta} + \underline{\delta}^{*\prime} \frac{\partial l^2(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}'} |_{\underline{\beta}_0} \underline{\delta}^* \quad ,$$

where $||\underline{\delta}^*||_1 < ||\underline{\delta}||_1$, that is, it is within the $\underline{\delta}$ neighborhood of $\underline{\beta}_0$.

So

$$\frac{1}{n}\left[l(\underline{\beta}_0 + \underline{\delta}) - l(\underline{\beta}_0)\right] = \frac{1}{n}\left[\underline{u}'(\underline{\beta}_0)\underline{\delta}\right] - \frac{1}{n}\left[\underline{\delta}^{*\prime} I_{obs}(\underline{\beta}_0)\underline{\delta}^*\right] \quad , \tag{4.9}$$

where $\underline{u}'(\underline{\beta}_0)$ is the score statistic evaluated at $\underline{\beta}_0$, and

$$I_{obs}(\underline{\beta}_0) = -\frac{\partial^2 l(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}'} |_{\underline{\beta}_0}$$

is the observed information matrix at $\underline{\beta}_0$.

The second part in the right-hand side of Equation (4.9) goes to zero with Assumption 5. The first part

$$\frac{1}{n}\left[\underline{u}'(\underline{\beta}_0)\underline{\delta}\right] = \frac{1}{n}\left[\sum_{w=1}^{k} u_w(\underline{\beta}_0)\delta_w\right] \quad , \tag{4.10}$$

where $u_w(\underline{\beta}_0)$ is the wth element of the score vector $\underline{u}(\underline{\beta}_0)$, and δ_w is the wth element of $\underline{\delta}$, $w = 1, 2, \cdots, k$.

With the justification on Page 72, $\underline{u}(\underline{\beta})$ is the sum of m uncorrelated components, by the law of large numbers, $u_w(\underline{\beta}_0)/n \to 0$ when $n \to \infty$ and $m \to \infty$, $w = 1, 2, \cdots, k$. Thus Equation (4.10)$\to 0$, and it follows that Equation (4.9)$\to 0$.

Also we have

$$\left| ||\underline{\beta}_0 + \underline{\delta}||_1 - ||\underline{\beta}_0||_1 \right| \leq ||\underline{\delta}||_1 \leq \underline{\delta}_0 \quad .$$

Therefore, if when $n \to \infty$, $\lambda_n/n \to 0$, then (4.8) holds. □

THEOREM 4.2. *With the regularity conditions in the assumptions,*

$$(\hat{\underline{\beta}} - \underline{\beta}_0) \xrightarrow{L} N(\underline{0}, I(\underline{\beta}_0)^{-1})$$

Proof: For $Q(\underline{\beta}) = -l(\underline{\beta}) + \lambda_n \|\underline{\beta}\|_1$, let's consider its score statistic, denoted as $\underline{v}(\underline{\beta})$:

$$\underline{v}(\underline{\beta}) = \frac{\partial Q(\underline{\beta})}{\partial \underline{\beta}} = \frac{\partial(-l(\underline{\beta}))}{\partial \underline{\beta}} + \lambda_n \mathrm{sgn}(\underline{\beta}) = -\underline{u}(\underline{\beta}) + \lambda_n \mathrm{sgn}(\underline{\beta}) \quad ,$$

where $\mathrm{sgn}(\underline{\beta}) = \mathrm{sgn}\left((\beta_1, \beta_2, \cdots, \beta_k)\right)$. $\underline{v}(\underline{\beta})$ is a vector of length k, where k is the number of explanatory variables.

Now we expand $v_w(\underline{\beta})$, the wth element of $\underline{v}(\underline{\beta})$, at $\underline{\beta}_0$,

$$\begin{aligned} v_w(\underline{\beta}) &= \left[\frac{\partial(-l(\underline{\beta}))}{\partial \beta_w} + \lambda_n \mathrm{sgn}(\beta_w)\right]_{\underline{\beta}_0} + \left\{\sum_{i=1}^{k}\left[\frac{\partial^2(-l(\underline{\beta}))}{\partial\beta_w \partial\beta_i}\right]_{\underline{\beta}_0} \cdot (\beta_i - \beta_{i0}) + 0\right\} \\ &+ \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\left[\frac{\partial^3(-l(\underline{\beta}))}{\partial\beta_w\partial\beta_i\partial\beta_j}\right]_{\underline{\beta}*} \cdot (\beta_i^* - \beta_{i0}) \cdot (\beta_j^* - \beta_{j0}) \quad , \end{aligned} \tag{4.11}$$

where $w = 1, 2, \cdots, k, \underline{\beta}^*$ is between $\underline{\beta}$ and $\underline{\beta}_0$. Since $\hat{\underline{\beta}}$ is the local minimizer from Theorem 4.1, the left-hand side of (4.11) is 0. Thus we have

$$\underline{0} = \underline{u}(\underline{\beta}_0) - \lambda_n \mathrm{sgn}(\underline{\beta}_0) - \left(I_{obs}(\underline{\beta}_0) - A\right) \cdot (\hat{\underline{\beta}} - \underline{\beta}_0) \quad ,$$

where

$$I_{obs}(\underline{\beta}_0) = -\frac{\partial^2(l(\underline{\beta}))}{\partial\underline{\beta}\partial\underline{\beta}'}|_{\underline{\beta}_0}$$

is the observed information matrix evaluated at $\underline{\beta}_0$. and

$$A = \frac{1}{2}\begin{pmatrix} (\hat{\underline{\beta}} - \underline{\beta}_0)' \cdot A_1 \\ (\hat{\underline{\beta}} - \underline{\beta}_0)' \cdot A_2 \\ \vdots \\ (\hat{\underline{\beta}} - \underline{\beta}_0)' \cdot A_k \end{pmatrix}$$

where

$$A_w = \left[\frac{\partial^2 u_w(\underline{\beta})}{\partial\underline{\beta}\partial\underline{\beta}'}\right]_{\underline{\beta}_0}$$

is the second derivative matrix of the wth element of the score statistic $\underline{u}(\underline{\beta})$, and is of dimension $k \times k$.

So we have

$$(\hat{\underline{\beta}} - \underline{\beta}_0) = \left(I_{obs}(\underline{\beta}_0) - A\right)^{-1}\left[\underline{u}(\underline{\beta}_0) - \lambda_n \mathrm{sgn}(\underline{\beta}_0)\right] \quad . \tag{4.12}$$

From Theorem 4.1, if $\lambda_n/n \to 0$ when $n \to \infty$, $(\hat{\underline{\beta}} - \underline{\beta}_0) \xrightarrow{P} \underline{0}$, and with Assumption 3, $|A_w|$ is bounded, thus the rows of A go to $\underline{0}$ when $n \to \infty$.

Now assume that the observed information matrix stabilizes at its expected value, that is, $I_{obs}(\underline{\beta}_0) \to \mathrm{E}\left[I_{obs}(\underline{\beta}_0)\right] = I(\underline{\beta}_0)$.

In partial likelihood, the score vector $\underline{u}(\underline{\beta}_0)$ is the sum of m uncorrelated terms as discussed on Page 72. With Assumption 5, using central limit theorem, we can get

$$\underline{u}(\underline{\beta}_0) \to N(\underline{0}, I(\underline{\beta}_0)) \quad .$$

So from (4.12), if when $n \to \infty$, not only $\lambda_n/n \to 0$, but further $\lambda_n \to 0$, then by Slutsky theorem,

$$(\hat{\underline{\beta}} - \underline{\beta}_0) \xrightarrow{L} N(\underline{0}, I(\underline{\beta}_0)^{-1})$$

□

Based on the proven theorems 4.1 and 4.2, we conclude that the penalized maximum partial likelihood estimator has the same asymptotic properties as the regular maximum partial likelihood estimator, as long as the penalty is small. More rigorous proofs require knowledge of counting process theory (Andersen et al., 1993), and the reader can refer to Fan and Li (2002).

4.2.4 Choice of the tuning parameter value

In the proposed model (4.3), the λ_j, $j = 1, 2, \cdots, p$ are tuning parameters. The magnitude of λ_j determines the number of variables whose coefficients will be estimated as zero for failure type j. The larger the value of λ_j, the larger the penalty is on the partial likelihood in Equation (4.3) , and the more variables will have zero coefficient estimates. The asymptotic results require that when sample size is large, the tuning parameter should approach 0. However, it does not explicitly state what value of the tuning parameter to use. When applying the model to do variable selection, we thus need to decide what specific values to use for estimating the coefficients.

One may choose the tuning parameter values that minimize some criterion, such as the Akaike information criterion (AIC) (Akaike, 1973), the Bayesian information criterion (BIC) (Schwarz, 1978), or a criterion based on cross-validation as used in Tibshirani (1997). All of these criteria are based on prediction accuracy. Leng et al. (2006) shows that prediction-accuracy-based criteria alone are not sufficient for the purpose of variable selection using lasso in linear regression problems. A similar conclusion is drawn in Meinshausen and Buhlmann (2006) where they studied neighborhood selection in high-dimensional graph with the lasso. They showed that the probability of including noise variables with the prediction-optimal λ value is in fact asymptotically 1. However, they demonstrated that consistent neighborhood selection is possible if the penalty is chosen larger than the prediction-optimal value, and they proposed to use the tuning parameter value that controls the probability of falsely joining some distinct components of the graph. This criterion cannot be applied to this project since the response is not multivariate normal data. The asymptotic results in subsection 4.2.3 imply that λ should go to 0 when $n \to \infty$ and $m \to \infty$. We used either $\lambda^1 = n^{0.3}$ or $\lambda^2 = n^{0.1}$ in model (4.3) when estimating the coefficients, where n is the total sample size.

4.3 Numerical Simulations to Evaluate the Proposed Model

The asymptotic properties state the consistency and normality of the L_1 penalized partial likelihood estimates when the sample size and the number of failures approach infinity. In medical applications, however, most often we have a small or moderate sample size. Therefore, we conducted numerical simulations to empirically evaluate the performance of the proposed model in identifying true important variables.

4.3.1 Simulation Parameters

To simulate the competing risks data, the following simulation parameters need be specified in order to generate the data:

- p: the number of failure types.

  In this simulation, p was fixed at 2. That is, we simulated the situation where there are two failure types.

- n: the total sample size.

  Two levels of n were simulated, specifically, n=100; and $n = 200$.

- k: the number of explanatory variables.

  One important application of the proposed method is for variable selection when the sample size is smaller than the number of explanatory variable, therefore, we allowed k to vary from smaller to larger than n. That is, we let $k = 10, 50, 100, 200, 500$, and 1000.

- k_{n0}: the number of truly important explanatory variables.

  Among the k explanatory variables, only k_{n0} of them are truly related to the survival. That is, their corresponding coefficients are non-zero whereas the remaining $k - k_{n0}$ variables have zero coefficients. To mimic the situation in gene expression study where only a small number of genes influence a phenotype, we assume that only a small number of variables are truly important to the survival. When $k = 10$, $k_{n0} = 4$; when $k > 10$, $k_{n0} = 20$.

- β_{n0}: the coefficient of the important explanatory variables.

  For the k_{n0} truly important variables, their effects on the survival are specified through their corresponding coefficients. The coefficients are fixed at $\beta_{n0} = \pm 2$.

Other parameters included:

- the ratio between the number of observations of Failure type I and the number of observations of Failure type II was fixed at 6:4.
- the percentage of overall censoring was fixed at 15%.
- the number of simulation runs for each combination of parameters was 20.
- the maximum correlation between the truly important explanatory variables and noise variables (variables not relevant to survival) was 0 (not correlated), or 0.80.

4.3.2 Data Generation

In each simulation run, for each combination of the simulation parameters, we generated the data as follows:

1. Generate the design matrix $X_{n \times k} = (\underline{x}_{(1)}', \underline{x}_{(2)}', \cdots, \underline{x}_{(k)}')$ following the outlined procedure, where $\underline{x}_{(w)}$, $w = 1, 2, \cdots, k$, is the vector of the n observations for the wth variable.

   (a) Generate $\underline{x}_w \sim N(0, 1)$ for $w = 1, 2, \cdots, k_{n0}$. Denote $X_a = (\underline{x}_{(1)}', \underline{x}_{(2)}', \cdots, \underline{x}_{(k_{n0})}')$. So X_a, of dimension $n \times k_{n0}$, is the submatrix of X, and corresponds to the observations of the k_{n0} truly important variables.

   (b) Denote as X_b the submatrix of X which corresponds to the observations of the $(k - k_{n0})$ noise variables. The noise variables can be correlated to the truly important variables, however, it is very important that the noise variables are not correlated to the survival times which are generated based on the truly important variables. Similar to the simulation mechanism in Gui and Li (2005), we generated X as follows:

      i. If we assume the noise variables are not correlated to the truly important variables, let Ω denote the linear space expanded by the vectors of $\underline{x}_{(1)}, \underline{x}_{(2)}, \cdots,$ and $\underline{x}_{(k_{n0})}$. We can obtain a normal-orthogonal base of the orthogonal complement space of Ω using the QR decomposition of X_a. Let $X_a = QR$ be the decomposition of X_a, then Q is an orthogonal matrix of dimension $n \times n$. Let B denote the submatrix of Q which excludes the first k_{n0} columns of Q, then $B^{'} X_a$ $=\mathbf{0},$ where $\mathbf{0}$ is the matrix only of 0s. B, of dimension $n \times (n - k_{n0})$, thus is a normal-orthogonal base of the orthogonal complement space of Ω. Any linear transformation of B is orthogonal complement of X_a. We generate a matrix C of dimension $(n-k_{n0})\times(k-k_{n0})$

(each column of $C \sim N(0,1)$), then $X_b = BC$, of dimension $n \times (k - k_{n0})$, is orthogonal complement of X_a, and therefore not correlated to the survival times. So X_b is the design matrix for the noise variables. Let $X = (X_a \vdots X_b)$, then X is the total design matrix whose first k_{n0} columns are observations for the k_{n0} truly important variables, and the remaining $(k - k_{n0})$ columns correspond to the observations of the noise variables.

ii. If we allow the noise variables to be correlated to the truly important variables, we first use Gram-Schmidt orthonormalization to obtain an orthogonal base of X_a (which is also an orthogonal base of the linear space Ω expanded by X_a), and denote this matrix as $X_{a\ m}$. Following the procedure previously described, using QR decomposition of $X_{a\ m}$ to get B, the normal-orthogonal base of the orthogonal complement space of Ω. Let $A = B + X_{a\ m} C_m$, where C_m is of dimension $k_{n0} \times (n - k_{n0})$ whose eigenvalues are not all zero, then A spans a linear space Ψ. Ψ is not orthogonal complement of Ω. The correlation between an arbitrary vector on Ψ and an arbitrary vector on Ω is bounded by

$$e_\lambda / \sqrt{(1 + e_\lambda^2)} \quad , \tag{4.13}$$

where e_λ^2 is the largest eigenvalue of $C^{'}_m C_m$ (the proof is sketched in the Appendix). So we select C_m with an appropriate choice of maximum eigenvalue of $C^{'}_m C_m$, then we can generate the observations of the noise variables on the space Ψ by taking linear transformations (of appropriate dimensions) of A. That is, let $X_b = AC$, where C is generated to be a matrix of dimension $(n - k_{n0}) \times (k - k_{n0})$ (each column of $C \sim N(0,1)$). Let $X = (X_a \vdots X_b)$, then X is the total design matrix whose first k_{n0} columns are observations for the k_{n0} truly important variables, and the remaining

$(k - k_{n0})$ columns correspond to the observations of the noise variables. The correlation between the important variables and the noise variables is bounded by Equation (4.13) (Gui and Li, 2005).

Here we consider the maximum correlation to be 0.8. From Equation (4.13), the largest eigenvalue of $C_m^{'}C_m$ is $e_\lambda^2 = 1.778$. We can simply let $C_m^{'}C_m$ be a diagonal matrix with 1.778 being the largest diagonal value. Then C_m can take the first k_{n0} rows of $C_m^{'}C_m$, with the original diagonal values being replaced by their square roots. Notice that the upper bound of the correlation between vectors of the spaces Ψ and Ω given in Equation (4.13) is a very loose bound. The observed maximum correlation normally is much smaller than this bound.

2. The vector of coefficients for Failure type I is $\underline{\beta}^1 = (\underbrace{\beta_{n0}, \cdots, \beta_{n0}}_{k_{n0}}, 0, \cdots, 0)$, that is, the first k_{n0} variables are those truly important variables which have non-zero effects on the survival. The vector of coefficients for Failure type II is $\underline{\beta}^2 = -\underline{\beta}^1$.

3. Generate the survival times corresponding to each failure type respectively. The survival time of each failure type T^j, $j = 1, 2$, is assumed to be exponentially distributed, where the effect of the explanatory variables is specified through the hazard function. To generate exponentially distributed survival data, we first generate $U \sim \text{Uniform}(0, 1)$, then

$$T = -\frac{\log(U)}{h_0 \exp(\underline{x}'\underline{\beta})} \sim \text{Exponential distribution} \quad ,$$

and the hazard function is $h(t|\underline{x}) = h_0 \exp(\underline{x}'\underline{\beta})$ (Bender et al. (2005), Leemis (1987)), where $\underline{x}$ is the vector of explanatory variables and h_0 is the baseline hazard, which is generated from the Weibull distribution with shape parameter 5 and scale parameter 2.

4. Generate a binary variable b to indicate the failure type of each individual. The probability of an observation with Failure type I is 60%, that is $b \sim$ Bernoulli(0.6), individual i is of Failure type I if $b_i = 1$, and of Failure type II if otherwise, $i = 1, 2, \cdots, n$.

5. The survival data for each individual then is if $b_i = 1$, then $T_i = T_i^1$, otherwise $T_i = T_i^2$, $i = 1, 2, \cdots, n$.

6. Generate a binary variable to indicate the censoring status of each observation. The probability of an observation to be censored is 15%. The censored times are generated from Uniform (2,10)

4.3.3 Simulation Results with Some Discussion

With the simulated dataset in each simulation run, we created the failure type specific "censoring" variable, that is, for the model for Failure type I, all observations that were either censored or failed due to Failure type II were considered censored. Likewise, for the model for Failure type II, all observations that were either censored or failed due to Failure type I were considered censored. Thereafter, we applied the proposed variable selection model using the L_1 regularization path algorithm (Park and Hastie, 2007) to estimate the coefficients. The performance of the proposed method was evaluated by the average sensitivity and specificity for identifying the truly important variables over the 20 simulations. The sensitivity in each simulation run was defined as the percent of the truly important variables being included in the estimated model, calculated as the number of truly important variables having a non-zero coefficient estimate in the final model out of k_{n0}. The specificity in each simulation run was defined as the percent of true noise variables not appearing in the estimated model, calculated as the number of true noise variables having a zero coefficient estimate out of $(k - k_{n0})$.

The choice of the tuning parameter values for identifying important variables used was described in Section 4.2.4. Again, two values $\lambda^1 = n^{0.3}$ and $\lambda^2 = n^{0.1}$ were examined.

When The Explanatory Variables Are Uncorrelated

When the truly important variables and noise variables were uncorrelated, the average sensitivity and specificity over the 20 simulation runs for each scenario are summarized in Table 4.1. For each value of λ and n, the sensitivity is plotted against k for both failure types in the left panel of Figure 4.1, and the specificity is plotted against k in the right panel of Figure 4.1.

From Table 4.1 and Figure 4.1, we see that:

- **as** $n \nearrow$

  Considering either failure type, fixing the number of variables (k), and considering the same tuning parameter λ, when the sample size increases, more variables tend to have coefficients estimated to be non-zero, resulting in higher sensitivity, with specificity being decreased slightly. For larger k, however, with the assumption that only a small number of variables are truly important, the specificity does not diminish as much as the gain in sensitivity.

- **as** $k \nearrow$

  Considering either failure type and the same tuning parameter λ, when the number of variables (k) increases, the specificity increases dramatically with a larger number of variables being noise variables; the sensitivity is quite stable when the sample size is $n = 200$, and only drops slightly when $n = 100$.

- **as** $\lambda \searrow$

  Considering either failure type, and fixing the number of variables (k) and the sample size (n), a smaller tuning parameter value imposes less penalty in Model (4.5) and more coefficients will be estimated to be non-zero. This means that using $\lambda^2 = n^{0.1} < \lambda^1 = n^{0.3}$, so that, we will expect a higher sensitivity and lower specificity with smaller λ. This conforms to what is observed from Table 4.1 and Figure 4.1. However, when the number of variables is large and only a small number of variables are truly important, such as $k = 500$ or $k = 1000$, the specificity using λ^2 is not much less (1%) than when using λ^1, whereas the sensitivity is about 5% higher.

- **result for Failure Type I and Failure Type II**

  The main difference between Failure type I and Failure type II in the simulations is that more events of Failure type I were generated (recall that the ratio between the number of observations of Failure type I and the number of observations of Failure type II is 6:4). With the availability of more events, the sensitivity for Failure type I was always better than that for Failure type II, while the specificity was lower than the specificity for Failure type II. When k is large, the difference in specificity is relatively small compared to the gain in sensitivity for Failure type I.

The simulation results demonstrate that a higher sensitivity is always coupled with a lower specificity, however, when the number of variables is large, and the assumption holds that only a small number of variables are truly important to survival, the decrease of specificity is much less than the gain of sensitivity. When $k \geq 500$, the false discovery rate (FDR) (FDR=1-positive predicted value) can be controlled to be less than 15%. This is a common threshold used for gene identification in high-throughput genomic experiments, so that the L_1 penalized Cox proportional hazards model for competing risks should be useful in identifying genes that are truly related to specific risks of events.

Further, the simulation study suggests that in general, a large sample size is desirable (as is the case for most statistical analyses). The variable selection method is also improved for failure types with more observed events. Moreover, when the sample size is relatively large compared to the number of explanatory variables, in order to guarantee an acceptable specificity, it may be preferable to apply a relatively large penalty (i.e. use larger tuning parameter value); on the other hand, when the number of variables is relatively large compared to the sample size, as is in gene expression studies, it is recommended that a smaller penalty be applied (i.e. use smaller tuning parameter value).

			k											
	n		10		50		100		200		500		1000	
			λ^1	λ^2	λ^1	λ^2	λ^1	λ^2	λ^1	λ^2	λ^1	λ^2	λ^1	λ^2
Failure Type I	100	sensitivity%	56.3	76.3	67.3	85.0	63.8	79.3	66.8	77.5	58.5	68.0	57.3	63.3
		specificity%	47.5	24.2	46.7	22.5	57.6	39.2	74.6	69.3	89.5	88.3	94.5	93.9
	200	sensitivity%	75.0	90.0	68.5	89.3	72.5	91.3	75.3	87.0	73.3	78.5	76.3	79.3
		specificity%	49.2	19.2	37.2	14.7	42.1	15.3	54.5	35.3	77.5	74.8	87.7	86.6
Failure Type II	100	sensitivity%	50.0	82.5	47.0	74.3	43.5	59.8	43.0	53.0	43.5	52.0	36.8	42.5
		specificity%	58.3	27.5	56.3	24.7	68.6	52.1	80.2	75.1	91.1	89.8	95.5	95.1
	200	sensitivity%	56.3	76.3	52.3	82.8	56.8	84.8	56.3	75.3	52.3	61.5	46.3	54.0
		specificity%	49.2	17.5	51.0	19.0	51.6	20.1	62.0	45.8	80.4	77.5	89.7	88.9

Table 4.1: Average sensitivity and specificity over the 20 simulation runs (variables are uncorrelated)

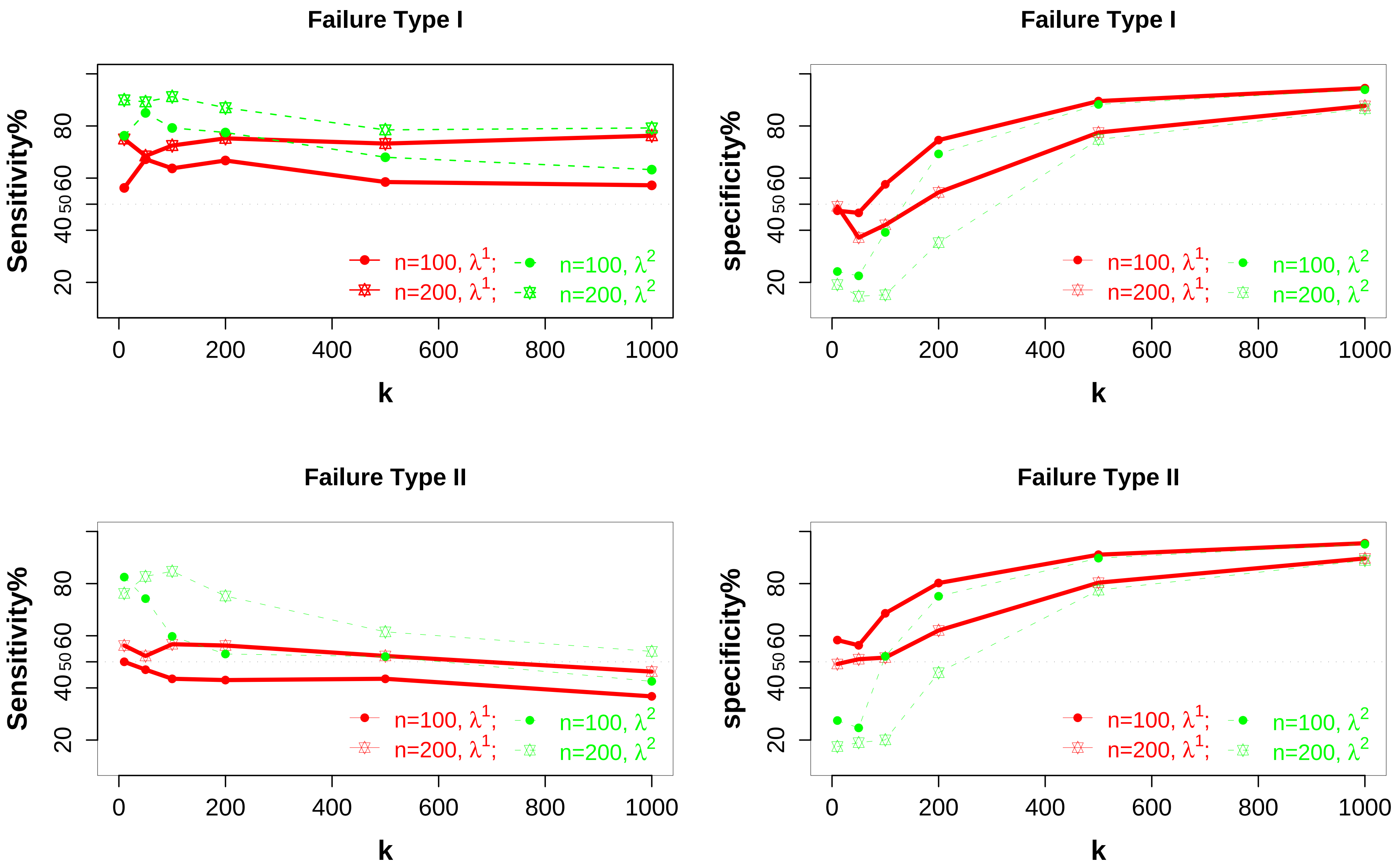


Figure 4.1: Average sensitity and specificity over 20 simulation runs when variables are uncorrelated. *k*: No. of variables.

When The Important Variables And The Noise Variables Are Correlated

We examined the situation where $n = 200$ and $k = 1000$ when the truly important and noise variables are correlated. As previously described, the maximum correlation allowed is bounded by 0.8. The data generation followed the procedures described in subsection 4.3.2. The maximum observed correlation coefficient between the important variables and noise variables from the generated data is about 0.2, which is in fact much smaller than the upper bound 0.8. The average sensitivity and specificity over the 20 simulation runs are summarized in Table 4.2.

Maximum correlation 0.8 n=200, k=1000			
		λ^1	λ^2
Failure Type I	sensitivity%	52.3	62.3
	specificity%	87.7	86.8
Failure Type II	sensitivity%	28.0	35.8
	specificity%	89.4	88.5

Table 4.2: Average sensitivity and specificity over the 20 simulation runs when the maximum correlation between the truly important variables and noise variables is 0.8, and $n = 200, k = 1000$.

To explore the influence of the correlation between the variables, we can compare the results to the corresponding columns in Table 4.1 where the variables are uncorrelated. For each failure type, the specificity was plotted against the sensitivity for both λ^1 and λ^2 for the correlated and uncorrelated scenario when $n = 200$ and $k = 1000$ (Figure 4.2). With the presence of correlation between the important variables and the noise variables, the average sensitivity diminishes. Using either choice of λ, the average sensitivity decreases more than 15%; whereas the average specificity remains to be larger than 85%. In addition, similar to the observation from Table 4.1, since there are more events of Failure type I, it always has a better sensitivity than Failure type II. A smaller tuning parameter value λ ($\lambda^2 < \lambda^1$) imposes less penalty and makes more variables have coefficients estimated to be non-zero, and therefore yields a better sensitivity with the cost of a slightly diminished specificity.

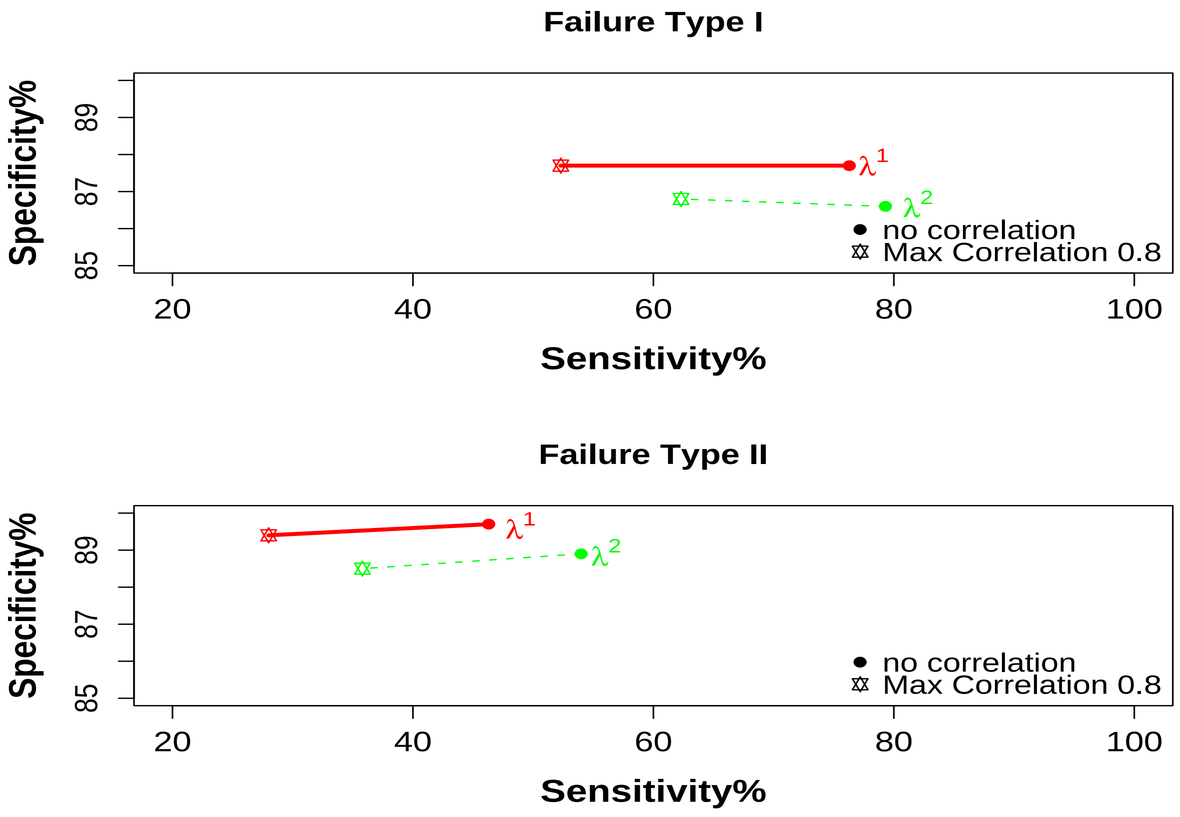


Figure 4.2: Comparing the average sensitivity and specificity over 20 simulation runs when the maximum correlation between the truly important and noise variables is 0.8 and when there is no correlation between the variables. n=200, k=1000.
(The maximum observed correlation between variables is about 0.2, smaller than the upper bound 0.8.)

Comparing to variable selection using the univariate Cox model approach

One intuitive method for variable selection is to fit a model using only one variable at a time, and then use the resulting p-value from testing the significance of the variable's coefficient estimate to quantify the relevance of the variable to the response. This univariable approach is especially widely used when the number of variables is large relative to the number of observations. We obtained the sensitivity and specificity from using the univariable approach for the situation where $n = 200$ and $k = 1000$, and compared them to the results from using our proposed approach based on the L_1 penalized Cox model.

With the same datasets generated for the situation where the variables were uncorrelated and $n = 200$ and $k = 1000$, for each failure type, we fit a univariable Cox model for each variable, with the variable being the single predictor. The p-value from the likelihood ratio test to test the significance of the model (that is, the significance of the variable) was compared to a pre-determined threshold α. If the p-value was less than α, the variable was selected as an important variable to survival of the failure type. We explored the result when choosing $\alpha = 0.05$, $\alpha = 0.25$ and $\alpha = 0.50$. The average sensitivity and specificity over the 20 simulations are listed in Table 4.3.

		α		
		0.05	0.25	0.50
Failure Type I	sensitivity%	7.5	32.0	58.3
	specificity%	95.2	75.6	50.6
Failure Type II	sensitivity%	4.0	25.0	52.5
	specificity%	94.8	74.8	50.3

Table 4.3: Average sensitivity and specificity over the 20 simulation runs using the univariable Cox model approach for the situation where the variables are uncorrelated and $n = 200$ and $k = 1000$

$\alpha = 0.05$ is a commonly used p-value threshold, however, from Table 4.3, using this criterion for variable selection does not yield satisfactory sensitivity. To ensure the sensitivity to be at least 50%, one should use a threshold as large as 0.50, with the cost of a dramatically decreased specificity.

Comparing to the results in Table 4.1 for the situation where $n = 200$ and $k = 1000$, our proposed model outperforms the univariable Cox model approach in terms of both sensitivity and specificity. The fact that the proposed L_1 penalized Cox model approach models all the variables simultaneously, may contribute to the better performance.

Chapter 5

Applications of the Proposed Model

In Chapter 4, we have developed the model for variable selection in competing risks survival data, and established some asymptotic properties of the model. The numerical simulations were useful in assessing the performance of the method under different scenarios. In this chapter, we illustrate the use of the proposed model by applying it to the two real competing risks problems introduced in Chapter 2, the prostatic cancer study and the HCV+HCC study. Specifically, Section 5.1 describes the results from the prostatic cancer analysis which was performed to study the effect of the treatment and identify important covariates. The conclusion from this analysis is discussed in reference to literature about earlier analyses performed using this dataset. Section 5.2 describes the results from the HCV+HCC analysis which was performed to identify genes significantly associated with tumor progression. When data from more patients become available, the updated results from the statistical analysis may be informative for clinical researchers' understanding of the roles of genes that are involved in the development and rapid progression of HCC.

5.1 Application to the Prostatic Cancer Study

The background of this study has been introduced in Section 2.5, with the "competing risks"' structure of the data shown in Figure 2.1. The survival time of a patient is defined as the time from the date of randomization or study entry to the date of death (or the end of the clinical trial or date of last follow-up if death was not observed during the trial). The explanatory variables in this prostatic cancer study included the treatment, either placebo or different doses of diethylstilbestrol (placebo, 0.2mg, 1.0mg, or 5.0mg), and the following 11 pretreatment covariates recorded at the beginning of the clinical trial (Andrews and Herzberg, 1985):

- Age in years (Age);
- Weight in kg (Wgt);
- Performance rating (PF): 0, normal activity; 1, in bed less than 50% of daytime; 2, in bed more than 50% of daytime; and 3, confined to bed;
- History of cardiovascular disease (CH): 0, no; 1, yes;
- Systolic blood pressure (SBP);
- Diastolic blood pressure (DBP);
- Serum haemoglobin in g/100ml (HG);
- Size of primary tumor estimated in cm^2 from rectal examination (TS): 00=no palpable tumor;
- Combined index of tumor stage and histologic grade (CI);
- Serum prostatic acid phosphatase in King-Armstrong units (AP);
- and Bone metastases (BM): 0, no; 1, yes.

There were 506 patients who participated in this trial and a subset of 483 patients had complete information for all covariates. The goal of this trial was to compare the effect due to treatment on survival of the patients with prostatic cancer. In addition, it is of interest to identify pretreatment covariates that are of prognostic importance. The different types of failure recorded in the original data are detailed in Figure 2.1. For the purpose of comparing our results to those previously published, we followed Cheng et al. (1998) and Ng and McLachlan (2003) categorization and coded failures into three types: death due to prostate cancer; death due to cardiovascular disease; and death due to other diseases.

5.1.1 Statistical Analysis and Result

To illustrate the use of our proposed model for variable selection in competing risks, we applied the L_1 penalized Cox model to the subset of the 483 patients having complete information for the covariates, to explore the effect of the treatment and identify important covariates for the risks of death due to prostate cancer and cardiovascular disease, respectively. Among these 483 patients, 125 patients (25.9%) died of prostatic cancer, 94 patients (19.5%) died of cardiovascular disease, 125 patients (25.9%) died of other diseases, and the remaining 139 patients (28.8%) were alive at the end of the trial.

All categorical variables were either ordinal in nature, such as treatment which was increasing doses of diethylstilbestro, or performance rating which was increasing with the severity of patient illness; the binary variables were coded 1 indicating the presence of the condition and 0 otherwise. Therefore, all variables were treated as continuous in the model. Before applying the L_1 regularization path algorithm, the measurements of the explanatory variables were standardized (each variable centered by subtracting its sample mean and scaled by dividing by its square root of variance) to avoid any impact of the original units of the variables on the downstream statistical analysis.

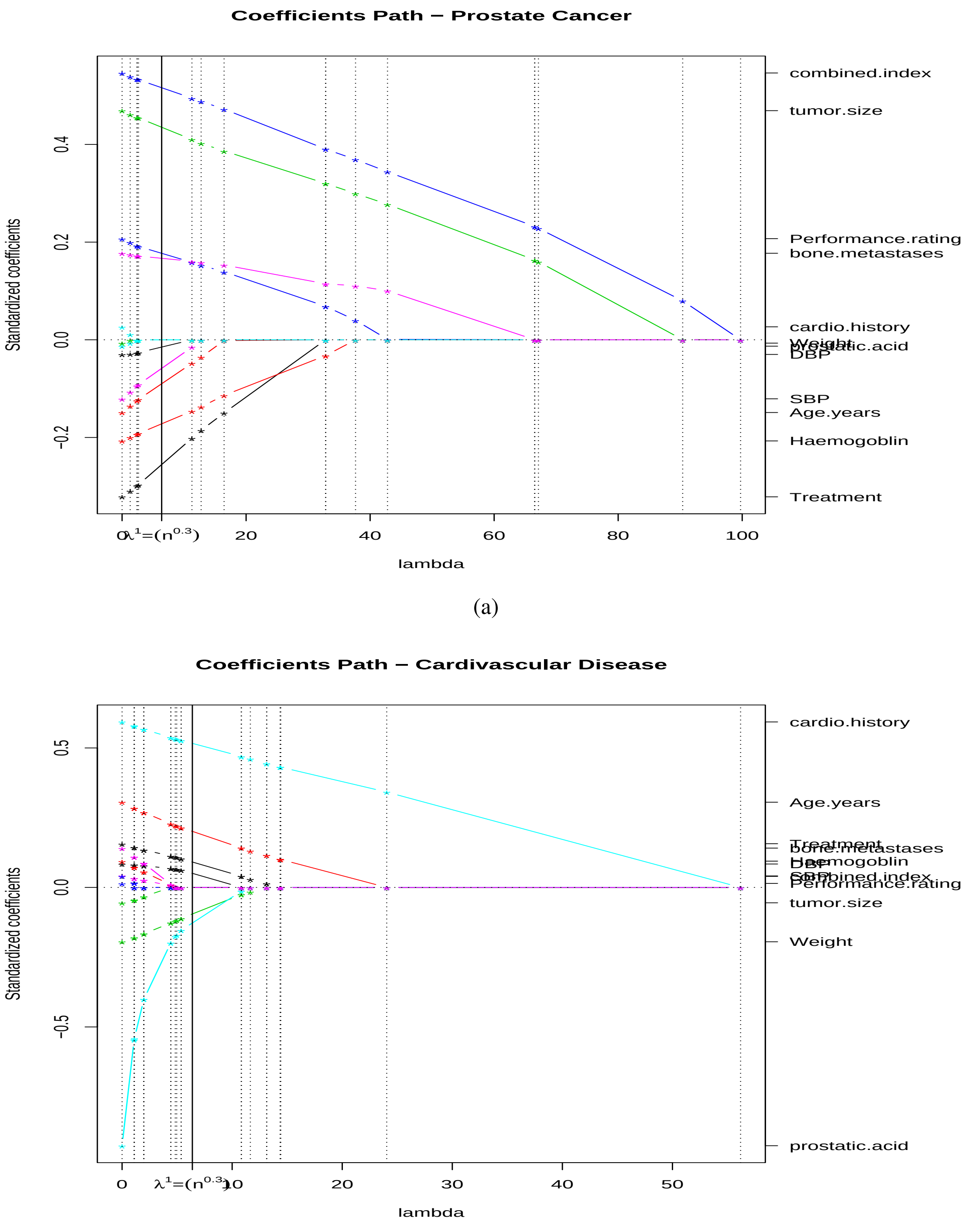
Model 4.3 was used to model the data. To estimate the model coefficients, following the procedures described in the subsection 4.2.2, we first created the "censoring" variable corresponding to each failure type that is of interest, and then the L_1 regularization path algorithm (Park and Hastie, 2007) was run. For this dataset, the number of variables is small relative to the total sample size. The simulation study in Section 4.3 suggests that when the sample size and the number of events for each failure type are relatively large compared to the number of variables, a smaller penalty (i.e., the smaller λ is chosen to be) will lead to decreased specificity. Therefore, it might be preferable to use a relatively large tuning parameter value. To ensure a reasonable level of specificity, we used $\lambda = n^{0.3}$. In this specific application, $\lambda^1 = n^{0.3} = 483^{0.3} = 6.385$. Figure 5.1 displays the traces of the coefficient estimates along the change of the tuning parameter λ (i.e. the magnitude of penalty) for death due to prostate cancer (top panel) and for death due to cardiovascular disease (bottom panel).

The coefficient estimates from the model using $\lambda^1 = n^{0.3}$, and the model using $\lambda = 0$, in which case the estimates are the ordinary maximum likelihood estimators of the coefficients are listed in Table 5.1. The standard errors, which appear in parentheses in Table 5.1, were estimated using the bootstrap method with $B = 100$ bootstrap resamplings (Efron and Tibshirani, 1993). In each bootstrap resampling, a random sample of size of n ($n = 483$) observations was drawn with replacement from the original dataset, and then the proposed L_1 penalized Cox model (Model (4.3)) with $\lambda = n^{0.3}$ was applied on the bootstrap sample to estimate the coefficients. If a variable is truly important to survival of a specific failure type, its corresponding coefficient estimate is expected to be non-zero when using a bootstrap resample. Figure 5.2 shows the boxplots of the coefficient estimates from the 100 bootstrap resamples for all the variables, respectively.

To quantify the significance of difference from 0 for each variable's estimated coefficient, we used a Wald test to test the null hypothesis that the coefficient estimate is equal to 0 versus the alternative that it is not equal to 0. The p-values for all variables are summarized in Table 5.2.

5.1.2 Conclusion and Discussion

From Figure 5.2 and Table 5.2, we can see that for the failure type death due to prostate cancer, the covariates tumor size (TS) and combined index of tumor stage and histologic grade (CI) have significant impact on the hazard of death from prostate cancer. The larger



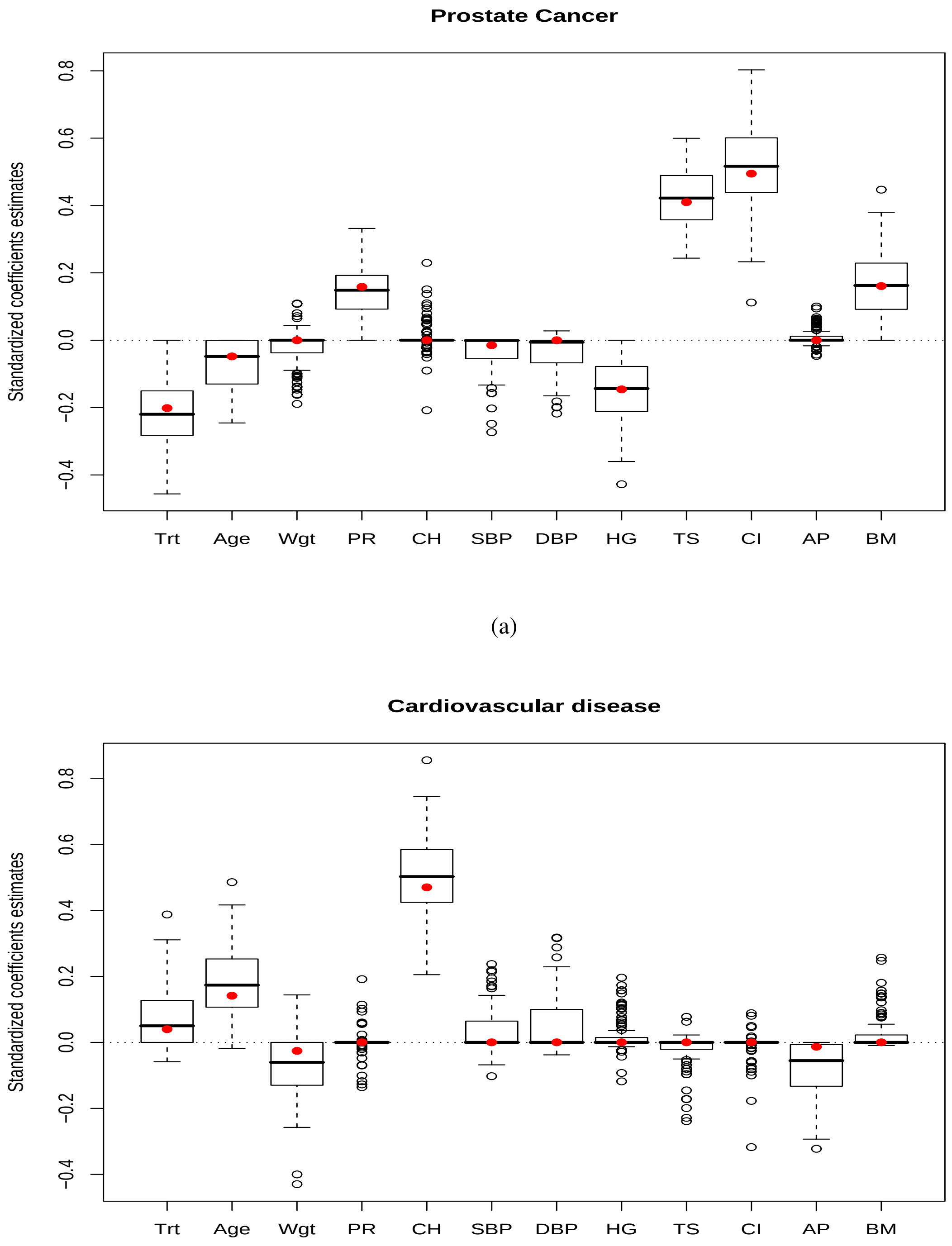
(a)

(b)

Figure 5.1: Traces of the coefficient estimates along the change of λ for death due to prostate cancer (top panel) and death due to cardiovascular disease (bottom panel). The dotted vertical lines indicate the values of λ at which the coefficient estimates change. Each $*$ is the estimated coefficient at the corresponding λ value for a variable. The real vertical line indicates the values of λ^1. The standardized coefficients refer to the coefficient estimates with the standardized measurements of the variables.

	Prostate Cancer				Cardiovascular Disease			
Coefficient	λ^1		$\lambda = 0$		λ^1		$\lambda = 0$	
Estimates	$n^{0.3}$		MLE		$n^{0.3}$		MLE	
Treatment	0.20	(0.10)	-0.32	(0.10)	0.04	(0.09)	0.16	(0.11)
Age	-0.05	(0.07)	-0.15	(0.09)	0.14	(0.11)	0.31	(0.13)
Wgt	0.00	(0.05)	-0.01	(0.10)	-0.03	(0.10)	-0.20	(0.12)
PR	0.16	(0.08)	0.21	(0.08)	0.00	(0.04)	0.01	(0.13)
CH	0.00	(0.05)	0.03	(0.10)	0.47	(0.11)	0.59	(0.11)
SBP	-0.02	(0.06)	-0.12	(0.12)	0.00	(0.06)	0.04	(0.13)
DBP	0.00	(0.06)	-0.03	(0.13)	0.00	(0.08)	0.08	(0.13)
HG	-0.15	(0.09)	-0.21	(0.10)	0.00	(0.05)	0.09	(0.12)
TS	0.41	(0.09)	0.47	(0.08)	0.00	(0.05)	-0.06	(0.13)
CI	0.49	(0.12)	0.55	(0.10)	0.00	(0.04)	0.04	(0.12)
AP	0.00	(0.03)	-0.01	(0.06)	-0.01	(0.08)	-0.93	(0.73)
BM	0.16	(0.10)	0.18	(0.09)	0.00	(0.05)	0.14	(0.14)

Table 5.1: Coefficient estimates for each variable and for death due to prostate cancer (left panel), and death due to cardiovascular disease (right panel) under two choices of tuning parameter value. The coefficient estimates correspond to standardized measurements of the variables. The standard errors in brackets were obtained using the bootstrap method for λ^1.



(a)

(b)

Figure 5.2: Boxplots of the coefficient estimates from the bootstrap resamples using the proposed L_1 penalized Cox model for death due to prostate cancer (top panel) and death due to cardiovascular disease (bottom panel). The red · in each box indicates the coefficient estimate from the original dataset.

the tumor is, or the more advanced the tumor is, the higher the risk of death from prostate cancer. Another factor that increases the hazard of death due to prostate cancer is the ability to perform normal activity (performance rating PF, p-value=0.021). Having bone metastases, which is an aspect considered in the combined index, also increases the hazard of death due to prostate cancer (p-value=0.061). On the other hand, a higher level of the covariate Haemoglobin (HG) indicates a decreased risk of death due to prostate cancer. Decrease of haemoglobin often leads to symptoms of anemia, and “anemia associated with advanced prostate cancer is a common occurrence” (Nalesnik et al., 2004). Moreover, researchers have studied the “diagnostic value of anemia in newly diagnosed metastatic prostate cancer” (Beer et al., 2004). The main interest of this clinical trial, the treatment diethylstilbestrol, decreases the risk of death from prostate cancer among the prostatic cancer patients. A higher dose of the treatment is statistically significantly associated with a decreased risk (p-value=0.027).

For the failure type death due to cardiovascular disease, the presence cardiovascular disease history (CH) significantly impacts the hazard of death from cardiovascular disease (p-value< 0.001). Patients’ age marginally increases the risk with p-value=0.098. An increased dose of treatment corresponds to an increased risk of death from cardiovascular disease, though the influence is not significant (p-value=0.331).

These findings are in accordance to what was concluded in Ng and McLachlan (2003). For further investigation of the dataset, the interaction terms between the treatment and covariates can be included in the model, by which we can study the effect of the treatment on different groups of patients. The reader can refer to Cheng et al. (1998), Kay (1986), Lunn and McNeil (1995), and Lunn and McNeil (1992) for other methods that were applied on this study and the results that were reported in these works.

5.1.3 Model Diagnostic

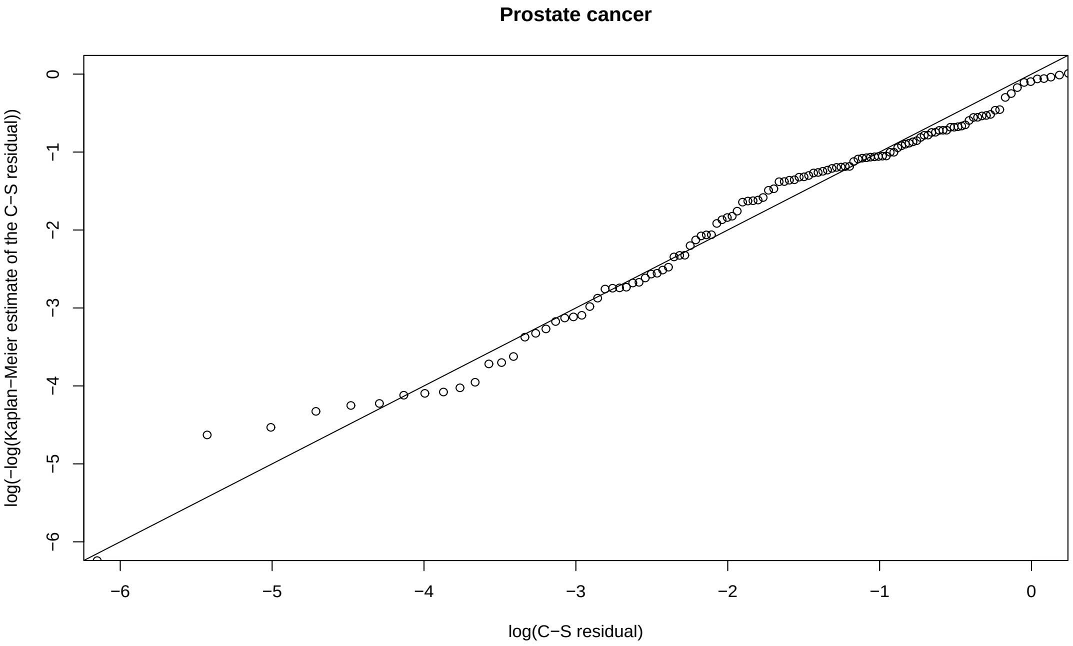
Since our model for variable selection is based on the Cox regression model which requires the assumption of proportional hazards, we examined the diagnostic plots of the Cox-Snell residuals (Chapter 4 of Collett (2003)) for each failure type (death due to prostate cancer and death due to cardiovascular disease), to check the validity of this assumption.

For death due to prostate cancer, all individuals with other types of failure were treated as censored. A Cox proportional hazards model was fit with the treatment and all the covariates as predictors. If the proportional hazards assumption is satisfied, the Cox-Snell residuals from this model for the individuals who died of prostate cancer are expected to follow an exponential distribution with parameter 1. Therefore, the plot of $\log r_i$ vs. $\log - \log \hat{S}(r_i)$ should be the diagonal line with unit slope and zero intercept, where r_i denotes the Cox-Snell residual for the ith individual who died of prostate cancer, and $\hat{S}(r_i)$ is the Kaplan-Meier estimate at r_i (refer to Section 1.2). For the failure type Cardiovascular disease, we obtained the same kind of plot of Cox-Snell residuals. Both plots are shown in Figure 5.3.

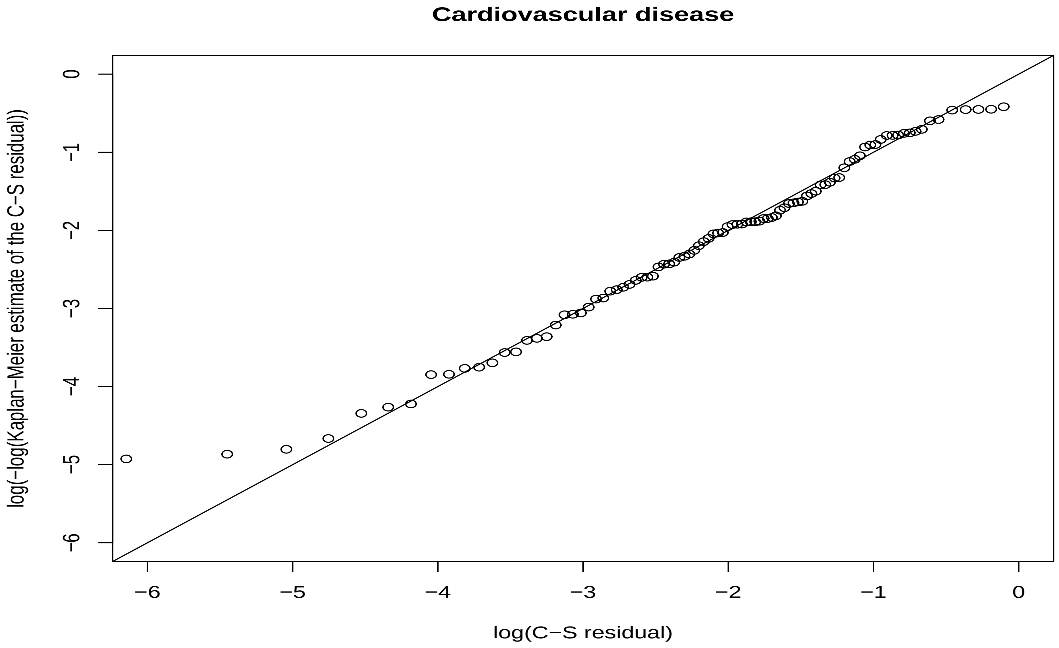
It can be seen that for both failure types, the assumption of proportional hazards is not severely violated. Therefore it is reasonable to use the proposed model which is based on Cox proportional hazards model for variable selection.

5.2 Application to the HCV+HCC study - Finding the Genes Related to Tumor Progression

The background of the HCV+HCC study was introduced in Section 2.5, with the “competing risks” structure of the data shown in Figure 2.2. The “survival time” for each patient is defined to be the time from the date of diagnosis of hepatocellular carcinoma (HCC) until the date of an event. Here the event can be either tumor progression, transplantation, death,



(a)



(b)

Figure 5.3: Plot of Cox-Snell residuals to check the validity of the proportional hazards assumption in the model for death due to prostate cancer (top panel) and the model for death due to cardiovascular disease (bottom panel).

or censored among those who had not experienced one of the previous three events. The explanatory variables are the expression level of 22,215 genes that were measured using Affymetrix GeneChip microarrays.

The tissue sample from 46 patients diagnosed with HCV+HCC underwent RNA extraction, cDNA synthesis, and biotin labeling. Among the 46 samples, 9 were hybridized to HG-U133A arrays and the remaining were hybridized to HG-U133A 2.0 arrays. The microarray data from the two different versions of arrays were first merged by probe sequence, and then the normalized probe set expression summaries were obtained using the Robust Multichip Average (RMA) method ((Irizarry et al., 2003). The quality of the microarrays was assessed by examining the 3':5' ratios of the control genes *ISGF*, *GAPDH* and *β-ACTIN*, which did not reveal quality concerns. The Affymetrix control probe sets were then removed from downstream survival analysis, leaving 22,215 probe sets to be analyzed.

5.2.1 Statistical Analysis and Results

Among the 46 patients, 14 were observed to have tumor progression, 25 had liver transplants, 2 died without progression and transplantation, and 5 were alive and on the waitlist as of the date the analysis was performed. Because there were only 2 deaths, we treated them as censored. Thus there are two main "failure causes": progression and transplantation. With transplantation as a competing event, it was of interest to identify a subset of genes that are significantly associated with tumor progression. We applied our proposed L_1 penalized Cox model for variable selection in competing risks (Model 4.3) to estimate the coefficients of the genes corresponding to the failure type "tumor progression".

Prior to fitting the model for time to tumor progression, we first created a "censoring" variable wherein all events other than progression were considered censored. Subsequently the measures of gene expression were standardized, and then the L_1 regularization path algorithm (Park and Hastie, 2007) was invoked. For this dataset, the number of variables is much larger than the total sample size. The simulation study in Section 4.3 suggests that when the number of variables is very large and a small portion of the variables is expected to be associated with the type of failure of interest, as is commonly assumed to be the case in gene expression studies, a smaller penalty (or smaller λ value) can yield better sensitivity without loss of much specificity. Therefore, for this dataset, we used $\lambda = n^{0.1} = 46^{0.1} = 1.466$.

Figure 5.4 shows the traces of the coefficient estimates as the tuning parameter λ (i.e. the magnitude of penalty) varied. The coefficient estimates of 19 probe sets were non-zero in the final model.

To quantify the significance of difference from 0 for each probe set's coefficient estimate, we used the bootstrap method to estimate the variability of the estimates with B =100 bootstrap resamplings (Efron and Tibshirani, 1993). In each bootstrap resampling, a random sample of size of n ($n = 46$) observations was drawn with replacement from the original dataset, and then the proposed L_1 penalized Cox model (Model 4.3) with $\lambda = n^{0.1}$ was applied on the bootstrap resample to estimate the coefficients of the probe sets. If a variable is truly important to survival of a specific failure type, its corresponding coefficient estimate is expected to be non-zero. However, for the majority of the 22,215 probe sets, the distribution of coefficients estimated using the bootstrap resamples are centered about 0 without much variability. This fact implies that a p-value from the Wald type test is not appropriate for evaluating the significance of difference from 0 for each probe set. We therefore used a Wilcoxon signed rank test (Hollander and Wolfe, 1999, Chapter 3) for each probe set to test the null hypothesis that the median coefficient estimate over the 100 bootstrap resamples is 0, versus the two-sided alternative hypothesis that the median coefficient estimate was not 0.

If we use $\alpha = 0.05$ as a threshold for the p-values from the individual Wilcoxon signed rank tests for all the probe sets, 42 probe sets are significantly associated with the hazard of tumor progression. Figure 5.5 shows the boxplots of the coefficient estimates from the

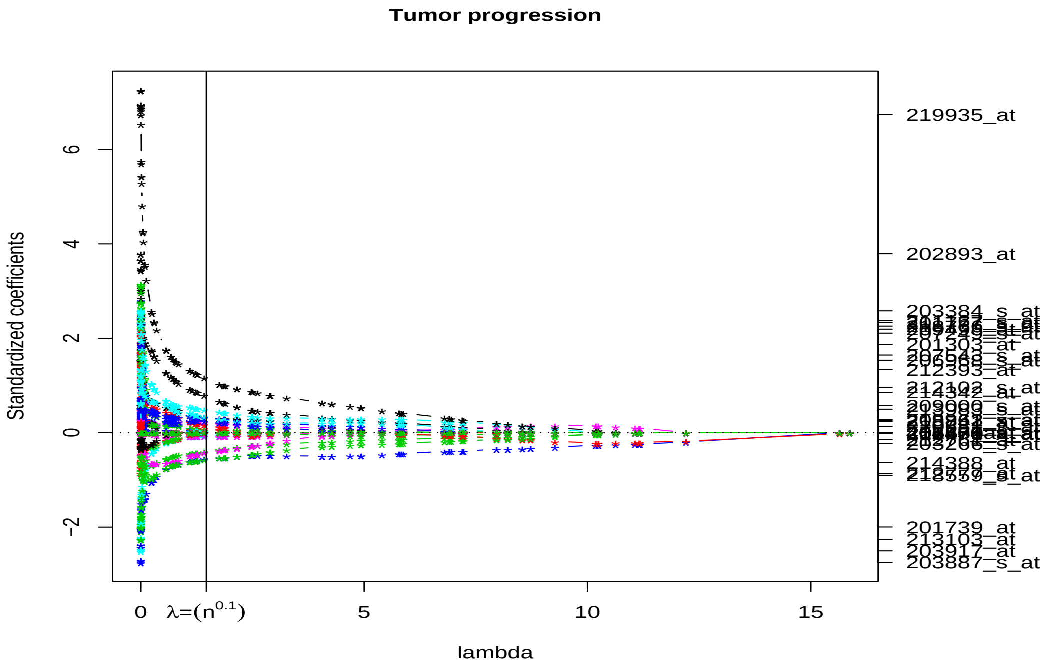


Figure 5.4: Traces of the coefficient estimates as the tuning parameter λ varied in the model for the time to tumor progression. Each $*$ indicates the estimated coefficient at the corresponding λ value for a variable. The real vertical line indicates the values of $\lambda = n^{0.1}$. The standardized coefficients refer to the coefficient estimates with the standardized measurements of the variables.

bootstrap resamples for these 42 probe sets. Among these 42 probe sets, the coefficients of 10 probe sets were estimated non-zero using the original data. The annotation data, the standardized coefficient estimates using the orginal data, the counts of non-zero bootstrap estimates out of the 100 bootstrap resamplings, and the p-values from the Wilcoxon signed rank tests for these probe sets are presented in Table 5.3.

	Prostate Cancer			Cardiovascular Disease		
	Estimate	(SE)	p-value	Estimate	(SE)	p-value
Treatment	-0.202	(0.104)	0.027	0.040	(0.091)	0.331
Age	-0.048	(0.069)	0.242	0.141	(0.109)	0.098
Wgt	0.000	(0.054)	0.500	-0.026	(0.097)	0.396
PR	0.159	(0.078)	0.021	0.000	(0.039)	0.500
CH	0.000	(0.046)	0.500	0.470	(0.112)	< 0.001
SBP	-0.015	(0.056)	0.394	0.000	(0.063)	0.500
DBP	0.000	(0.550)	0.500	0.000	(0.080)	0.500
HG	-0.146	(0.092)	0.056	0.000	(0.051)	0.500
TS	0.410	(0.087)	< 0.001	0.000	(0.052)	0.500
CI	0.495	(0.116)	< 0.001	0.000	(0.043)	0.500
AP	0.000	(0.027)	0.500	-0.013	(0.084)	0.436
BM	0.161	(0.104)	0.061	0.000	(0.055)	0.500

Table 5.2: Coefficient estimates for each variable and its corresponding standard error and p-value obtained using the Wald test, based on the choice of $\lambda^1 = n^{0.3}$. The coefficient estimates are based on standardized measurements of the variables. The standard errors in brackets were obtained using the bootstrap method with $B = 100$ resamplings.

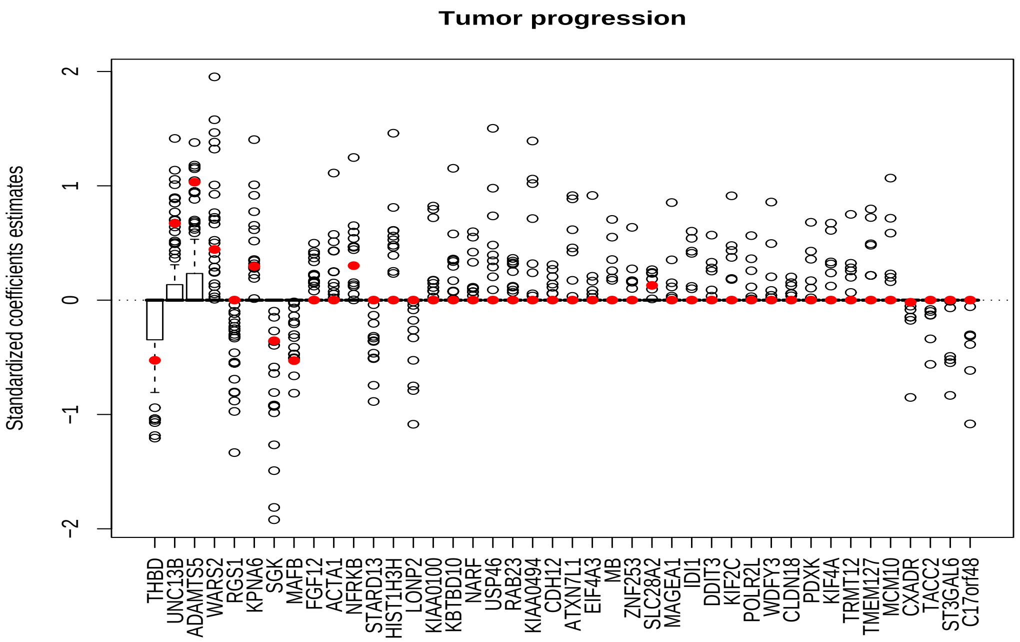


Figure 5.5: Boxplots of the coefficient estimates from the bootstrap resamples for the 42 identified probe sets. The red · in each box indicates the coefficient estimate from the original dataset.

Table 5.3: Probe sets significantly associated with time to tumor prossion. The probe sets are listed in increasing order of p-values.

LocusLink	UnigeneID	AffyID	Gene Symbol	Gene Name	Chromo -some	Map	Coeffcient Estimate	Count	p-value
7056	Hs.2030	203887_s_at	THBD	thrombomodulin	20	20p11.2	-0.53	37	< 0.01
10497	Hs.493791	202893_at	UNC13B	unc-13 homolog B (C. elegans)	9	9p12-p11	0.67	33	< 0.01
11096	Hs.58324	219935_at	ADAMTS5	ADAM metallopeptidase with thrombospondin type 1 motif, 5 (aggrecanase-2)	21	21q21.3	1.03	33	< 0.01
10352	Hs.523506	218766_s_at	WARS2	tryptophanyl tRNA synthetase 2, mitochondrial	1	1p13.3-p13.1	0.44	24	< 0.01
5996	Hs.75256	216834_at	RGS1	regulator of G-protein signaling 1	1	1q31	0.00	21	< 0.01
23633	Hs.470588	212102_s_at	KPNA6	karyopherin alpha 6 (importin alpha 7)	1	1p35.1-p34.3	0.29	16	< 0.01
6446	Hs.510078	201739_at	SGK	serum/glucocorticoid regulated kinase	6	6q23	-0.35	16	< 0.01
9935	Hs.651210	218559_s_at	MAFB	v-maf musculoaponeurotic	20	20q11.2-q13.1	-0.53	15	< 0.01

Continued on next page

Table 5.3: Probe sets significantly associated with time to tumor prossion. The probe sets are listed in increasing order of p-values.

LocusLink	UnigeneID	AffyID	Gene Symbol	Gene Name	Chromo -some	Map	Coeffcient Estimate	Count	p-value
				fibrosarcoma oncogene homolog B (avian)					
2257	Hs.584758	207501_s_at	FGF12	fibroblast growth factor 12	3	3q28	0.00	13	< 0.01
58	Hs.1288	203872_at	ACTA1	actin, alpha 1, skeletal muscle	1	1q42.13-q42.2	0.00	12	< 0.01
4798	Hs.530539	206968_s_at	NFRKB	nuclear factor related to kappaB binding protein	11	11q24-q25	0.30	12	< 0.01
90627	Hs.507704	213103_at	STARD13	StAR-related lipid transfer (START) domain containing 13	13	13q12-q13	0.00	12	< 0.01
8357	Hs.591778	206110_at	HIST1H3H	histone cluster 1, H3h	6	6p22-p21.3	0.00	11	< 0.01
83752	Hs.694785	221834_at	LONP2	lon peptidase 2, peroxisomal	16	16q12.1	0.00	10	0.01
9703	Hs.591189	201729_s_at	KIAA0100	KIAA0100	17	17q11.2	0.00	10	0.01
10324	Hs.50550	219106_s_at	KBTBD10	kelch repeat and BTB (POZ) domain containing 10	2	2q31.1	0.00	10	0.01

Continued on next page

Table 5.3: Probe sets significantly associated with time to tumor prossion. The probe sets are listed in increasing order of p-values.

LocusLink	UnigeneID	AffyID	Gene Symbol	Gene Name	Chromo -some	Map	Coeffcient Estimate	Count	p-value
26502	Hs.256526 Hs.600304	219862_s_at	NARF	nuclear prelamin A recognition factor	17	17q25.3	0.00	10	0.01
64854	Hs.331478	203870_at	USP46	ubiquitin specific peptidase 46	4	4q12	0.00	9	0.01
51715	Hs.555016	220955_x_at	RAB23	RAB23, member RAS oncogene family	6	6p11	0.00	9	0.01
9813	Hs.100874	201777_s_at	KIAA0494	KIAA0494	1	1pter-p22.1	0.00	8	0.01
1010	Hs.113684	207149_at	CDH12	cadherin 12, type 2 (N-cadherin 2)	5	5p14-p13	0.00	7	0.02
222255	Hs.489603	214342_at	ATXN7L1	ataxin 7-like 1	7	7q22.2	0.00	7	0.02
9775	Hs.389649	201303_at	EIF4A3	eukaryotic translation initiation factor 4A, isoform 3	17	17q25.3	0.00	6	0.04
4151	Hs.517586	204179_at	MB	myoglobin	22	22q13.1	0.00	6	0.04
56242	Hs.659321	206900_x_at	ZNF253	zinc finger protein 253	19	19p13.11	0.00	6	0.04
9153	Hs.367833	207249_s_at	SLC28A2	solute carrier family 28 (sodium-coupled nucleoside transporter),	15	15q15	0.13	6	0.04

Continued on next page

Table 5.3: Probe sets significantly associated with time to tumor prossion. The probe sets are listed in increasing order of p-values.

LocusLink	UnigeneID	AffyID	Gene	Gene	Chromo	Map	Coeffcient	Count	p-value
			Symbol	Name	-some		Estimate		
				member 2					
4100	Hs.72879	207325_x_at	MAGEA1	melanoma antigen family A, 1 (directs expression of antigen MZ2-E)	X	Xq28	0.00	6	0.04
3422	Hs.283652	208881_x_at	IDI1	isopentenyl-diphosphate delta isomerase 1	10	10p15.3	0.00	6	0.04
1649	Hs.505777 Hs.690217	209383_at	DDIT3	DNA-damage-inducible transcript 3	12	12q13.1-q13.2	0.00	6	0.04
11004	Hs.69360	209408_at	KIF2C	kinesin family member 2C	1	1p34.1	0.00	6	0.04
5441	Hs.441072	211730_s_at	POLR2L	polymerase (RNA) II (DNA directed) polypeptide L, 7.6kDa	11	11p15	0.00	6	0.04
23001	Hs.480116	212602_at	WDFY3	WD repeat and FYVE domain containing 3	4	4q21.23	0.00	6	0.04
51208	Hs.655324	214135_at	CLDN18	claudin 18	3	3q22.3	0.00	6	0.04
8566	Hs.284491	218019_s_at	PDXK	pyridoxal (pyridoxine,	21	21q22.3	0.00	6	0.04

Continued on next page

Table 5.3: Probe sets significantly associated with time to tumor prossion. The probe sets are listed in increasing order of p-values.

LocusLink	UnigeneID	AffyID	Gene Symbol	Gene Name	Chromo -some	Map	Coeffcient Estimate	Count	p-value
				vitamin B6) kinase					
24137	Hs.648326	218355_at	KIF4A	kinesin family member 4A	X	Xq13.1	0.00	6	0.04
55039	Hs.9925	219299_at	TRMT12	tRNA methyltransferase 12 homolog (S. cerevisiae)	8	8q24.13	0.00	6	0.04
55654	Hs.355708	219460_s_at	TMEM127	transmembrane protein 127	2	2q11.2	0.00	6	0.04
55388	Hs.198363	220651_s_at	MCM10	minichromosome maintenance complex component 10	10	10p13	0.00	6	0.04
1525	Hs.634837	203917_at	CXADR	coxsackie virus and adenovirus receptor	21	21q21.1	-0.02	6	0.04
10579	Hs.501252 Hs.695119	211382_s_at	TACC2	transforming, acidic coiled-coil containing protein 2	10	10q26	0.00	6	0.04
10402	Hs.148716	213355_at	ST3GAL6	ST3 beta-galactoside alpha-2,3-	3	3q12.1	0.00	6	0.04

Continued on next page

Table 5.3: Probe sets significantly associated with time to tumor prossion. The probe sets are listed in increasing order of p-values.

LocusLink	UnigeneID	AffyID	Gene Symbol	Gene Name	Chromo-some	Map	Coeffcient Estimate	Count	p-value
				sialyltransferase 6					
56985	Hs.661424 Hs.47668	220606_s_at	C17orf48	chromosome 17 open reading frame 48	17	17p13.1	0.00	6	0.04

5.2.2 Conclusion and Discussion

Clinical researchers may be interested in further investigating the functions of the genes in Table (5.3) on the progression of hepatocellular carcinoma. In fact, several genes have been reported in liver disease research. The gene *THBD* (thrombomodulin) has a statistically significant coefficient -0.526 (p-value< 0.001), meaning that an increased level of *THBD* expression reduces the risk of tumor progression. It is known that *THBD* "converts thrombin from procoagulant into anticoagulant protein to activate protein C. Thrombin also plays an important role in the metastatic process of cancer cells"(Suehiro et al., 1995). The authors performed an immunohistochemical and clinicopathological study of *THBD* in 141 patients with resected hepatocellular carcinoma (HCC) measuring less than 6 cm in diameter. They found that the recurrence freedom rate was significantly higher in patients whose tissue stained positive for *THBD* than patients whose tissue stained negative for *THBD* . And thus *THBD* -producing HCC showed a slow intrahepatic spread. They concluded that these findings "suggested that *THBD* may inhibit the adhesion of tumor cells to the portal vein because of anticoagulant activity and thus prevent the spread of intrahepatic metastasis" (Suehiro et al., 1995). Expression of *THBD* was also compared between cirrhotic non-HCC patiens and HCC patients in a separate study by Biguzzi et al. (2007). The authors found that *THBD* had elevated levels among patients with HCC in comparison to those without HCC, and concluded that *THBD* may be an important marker of HCC development among patients with liver cirrhosis.

Although this analysis included 46 HCV+HCC patients, the study is continuing with a target enrollment of 150 hepatitis C virus infected patients with HCC, and the anticipated progression rate among these patients is 40%. With the availability of more data, the variable selection model for competing risks proposed in this thesis may yield improved results because the genes identified from the analysis of the full study data will be of greater sensitivity and specificity. The results may help the researchers better understand the progression of HCC at the molecular level, and may be used as markers for prognosis or drug target in the future.

5.2.3 Model Diagnostics

Since there are many more variables than the sample size, it is not possible to build a Cox model with all the probe sets as predictors. To evaluate the validity of the proportional hazards assumption, we obtained plots of the Cox-Snell residuals from the univariate Cox models (i.e., each gene as the single predictor in the model) for a few genes. Specifically, Figure 5.6 shows that plot for the probe set “202893_at”, which is an identified probe set in Table 5.3. It can be seen that the assumption of proportional hazards is reasonable.

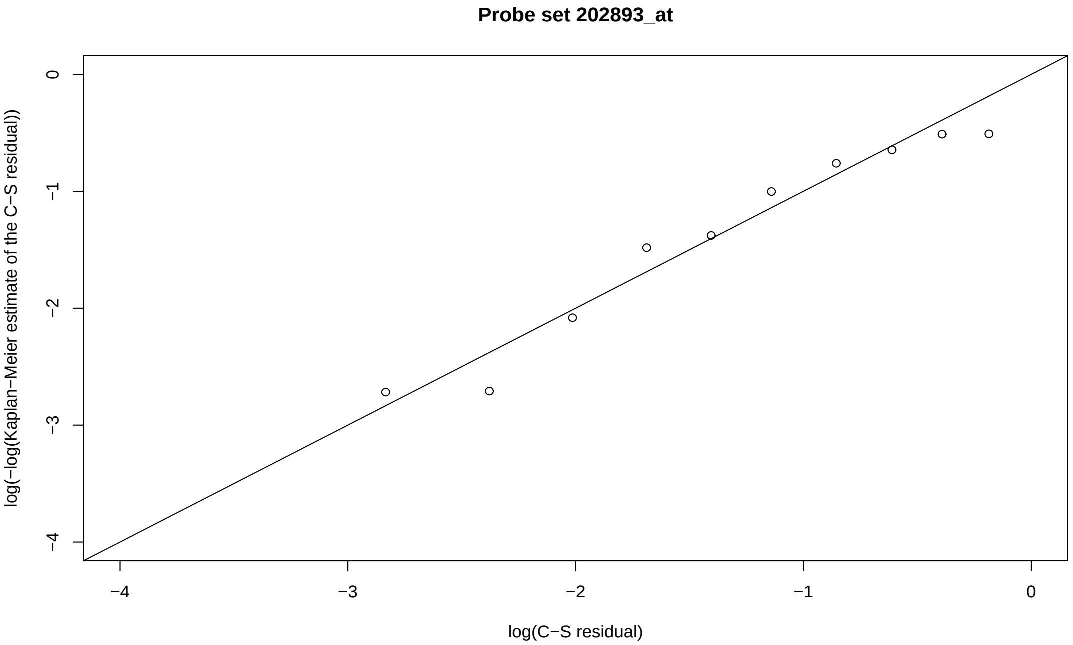


Figure 5.6: Plot of Cox-Snell residuals to check the validity of the proportional hazards assumption for probe set 203893_at.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

Survival analysis is a field in statistics that deals with the modeling and analysis of survival data: time from a well-defined time origin until the occurrence of some event or end points of interest. In Chapter 1, we reviewed the features of survival data and the probabilistic functions used for describing survival data. As one of the major tasks when examining survival data is to assess the dependence of survival time on explanatory variables, we reviewed in detail some popular modeling techniques used in univariate survival data analysis. Under certain scenarios, problems with multivariate survival data can arise, and some topics about multivariate survival data were introduced at the end of Chapter 1. Specifically, the topic of competing risks, which is the focus of this thesis, was thoroughly reviewed in Chapter 2, including the probabilistic functions used for describing competing risks data, the modeling approaches, and two examples from real-world problems. One example pertained to a prostatic cancer clinical trial, which has been a classical example for illustrating competing risks survival data. The other example originated from the ongoing NIH funded project "Genes related to HCC (hepatocellular carcinoma) progression in living donor and deceased donor liver transplant". One specific aim of this study is to identify genes that are associated with tumor progression in hepatitis C virus (HCV) infected patients diagnosed with HCC. In Chapter 3, we reviewed the penalized regression model, with emphasis on the lasso linear model (i.e. L_1 penalized model) and lasso Cox model, which can be used for variable selection. The L_1 regularization path algorithm by Park and Hastie (2007) was also introdued in Chapter 3.

The goal of this thesis, which was explicitly stated in Chapter 4, is to perform variable selection with competing risks survival outcome. That is, each individual is subject to failure from multiple failure types, though only one type of failure is observed for the individual. Also measured are some explanatory variables at the time origin on each individual. It is of interest to identify the subset of explanatory variables that are significantly associated with each failure type or some specific failure types. We proposed a model based on L_1 penalized Cox proportional hazards model to achieve this purpose. The algorithm that can be used to estimate the model parameters was also provided. Some asymptotic properties of the model, together with the proofs, were presented. Moreover, numerical simulations were conducted to empirically evaluate the performance of the proposed model in selecting the correct important variables to survival due to each failure type. One important feature of the proposed model is that all the explanatory variables are modeled simultaneously, and the algorithm presented can be used for estimation when the number of variables is larger than the number of sample size. In Chapter 5, the proposed model was applied to two real-world problems that had been previously described in Chapter 2. The result from the statistical analysis on the prostatic cancer dataset conforms to what has been concluded in earlier publications by other researchers. For the HCV+HCC study, with the currently available 46 samples, the application of the model identified 42 genes that were significantly associated with tumor progression in HCV infected patients diagnosed with HCC. When more samples become available, the result of identifying the subset of genes using the proposed model will be improved. Further investigation and validation of the identified genes may lead to better understanding of tumor progression at the molecular level, and thus improve prognosis among HCV+HCC patients waitlisted for liver transplantation.

6.2 Future Work

6.2.1 Variable Selection in Competing Risks Taking into Account the Correlations between Genes

The proposed model for variable selection in competing risks was applied on the gene expression study about HCV+HCC patients to identify the subset of genes that are associated with tumor progression. However, it is known that genes do not work independently, and the interrelation structure among the genes was neglected when applying the proposed model.

The Gene Ontology (GO) project provides databases of structured controlled vocabularies (ontologies) which describe gene and gene products in terms of their associated biological processes, cellular components and molecular functions for any organism. The Kyoto Encyclopedia of Genes and Genomes (KEGG) project is another resource for information about genomes, enzymatic pathways, and biological chemicals. The KEGG Pathway database collects known knowledge about molecular interaction and reaction networks. Using information from these databases, we can mathematically describe the interrelationships between genes, and then incorporate the interrelation structure into the variable selection analysis.

Graphs are a common way of depicting the gene networks. An example is shown in Figure 6.1 that graphically illustrates genes involved in the Prion disease pathway.

One intuitive method to mathematically describe a graph is its adjacency matrix (Wang, 1997). Let G represent a graph with k vertices, and $V(G) = \{v_1, v_2, \cdots, v_k\}$ represent the set of vertices of G. Then $A(G)$, of dimension $k \times k$, is the adjacent matrix of G, where the i, jth element of $A(G)$ is

$$a_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are connected,} \\ 0, & \text{if } v_i \text{ and } v_j \text{ are not connected or } i = j. \end{cases}$$

Therefore, for the graph in Figure 6.1, if neglecting the directions in the graph, its adjacent matrix is

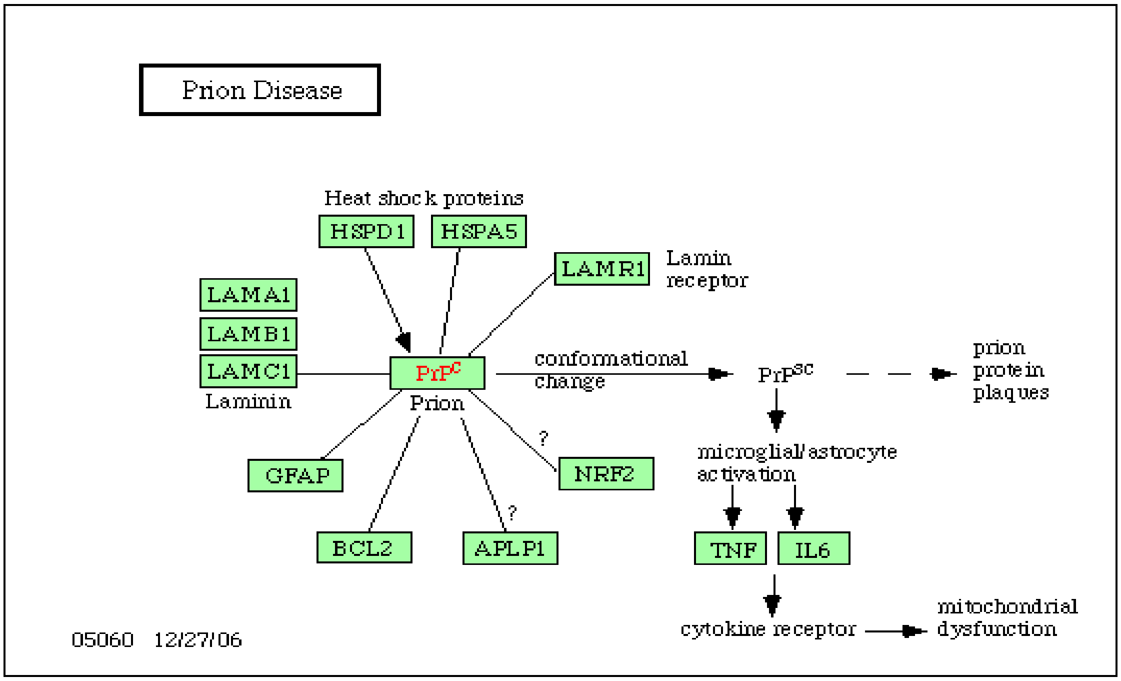


Figure 6.1: Prion disease pathway. http://www.genome.ad.jp/kegg/pathway/hsa/hsa05040.html.

	$LAMA1$	$LAMB1$	$LAMC1$	$HSPD1$	$HSPA5$	$LAMR1$	PrP^c	$GFAP$	$BCL2$	$APLP1$	$NRF2$	TNF	$IL6$
$LAMA1$	0	0	0	0	0	0	1	0	0	0	0	0	0
$LAMB1$	0	0	0	0	0	0	1	0	0	0	0	0	0
$LAMC1$	0	0	0	0	0	0	1	0	0	0	0	0	0
$HSPD1$	0	0	0	0	0	0	1	0	0	0	0	0	0
$LAMR1$	0	0	0	0	0	0	1	0	0	0	0	0	0
PrP^c	0	0	0	0	0	0	0	0	0	0	0	0	0
$GFAP$	0	0	0	0	0	0	1	0	0	0	0	0	0
$BCL2$	0	0	0	0	0	0	1	0	0	0	0	0	0
$APLP1$	0	0	0	0	0	0	1	0	0	0	0	0	0
$NRF2$	0	0	0	0	0	0	1	0	0	0	0	0	0
TNF	0	0	0	0	0	0	1	0	0	0	0	0	0
$IL6$	0	0	0	0	0	0	1	0	0	0	0	0	0

Li and Li (2008) in fact have recently used the normalized Laplacian matrix for a graph G to account for the known information of the gene networks. They imposed both L_1 penalty and a quadratic penalty of the coefficients based on the Laplacian matrix in a linear regression model to identify the subset of genes that were associated with the response. They did not consider the directions known in the gene networks, and theories about the directed graph may be further utilized to better describe the networks of genes. Another extension of their work is to model survival responses using models for survival analysis.

6.2.2 Variable Selection in Models for Categorical Data Analysis

The penalized likelihood approach can be easily extended to models for categorical data analysis, such as logistic regression model, logit model, or loglinear model. Penalty based on L_r ($r \leq 1$) norm of the coefficients in these models normally have a similar effect on the estimation of the coefficients as in a linear regression model and Cox proportional hazards survival model. That is, the penalty yields sparseness of the coefficient estimates, and thus can be used for the purpose of variable selection.

One application of the penalized likelihood approach in logistic regression model is to analyze SNP (Single Nucleotide Polymorphism) data for comparing regions of the genome between cohorts, such as matched cohorts with and without a certain phenotype. A SNP is a DNA sequence variation occurring when a single nucleotide - A, T, C, or G - in the genome differs between individuals of a species. For a variation to be considered a SNP, it must occur in at least 1% of the population. Many SNPs have no effect on cell function, but it is believed that some variations in the human DNA sequences can affect how humans develop diseases and respond to pathogens, chemicals, drugs, and other agents (interested readers can refer to Human Genome Project - SNP Fact Sheet: http://www.ornl.gov/sci/techresources/Human_Genome/faq/snps.shtml#snps). The variable selection function of the penalized likelihood approach can be utilized to find the SNPs that might be associated with the phenotype of interest. The L_1 regularization path algorithm for generalized linears model by Park and Hastie (2007) can again be used for estimation, which accommodates the situation where there are more variables (SNPs) than number of observations. It is also potentially possible to account for gene-gene interactions and gene-environment interactions when using penalized likelihood in a logistic regression model.

It may be interesting to further investigate the identified genes from the HCV+HCC study by exploring the SNPs on these gene sequences with consideration of the sample's population infomation in the study. A hypothesis is that if that the expression of the identified genes is associated with tumor progression is due to variations on the gene sequences (including copy number variation), then these variations might be used for prognosis or as a drug target for the studied population.

6.2.3 Choice of Tuning Parameter for Best Variable Selection

The coefficient estimation using the penalized likelihood approach as in Model 4.3 is dependent on the choice of the tuning parameter value. Often the choice of the tuning parameter is chosen to be the value that minimizes prediction error which can be estimated by cross-validation (Tibshirani, 1996, 1997). For model selection (i.e. selection of the subset of important variables) in ordinary linear regression setting, Shao (1993) considered the problem of using leave-one-out cross-validation method to evaluate the predictive ability of a model. The author showed that the leave-one-out cross-validation method, "which is asymptotically equivalent to other methods such as Akaike information criterion (AIC) (Akaike, 1973), the C_p (Mallows, 1973) and bootstrap (Efron, 1983, 1986)", is asymptoically inconsistent in the sense that the probability of selecting the model with the best predictive ability does not converge to 1 as the sample size $n \rightarrow \infty$. It was concluded that using a leave-n_v-out cross-validation (n_v being the number of observations reserved for validation), where n_v satisfying $n_v/n \rightarrow 1$ as $n \rightarrow \infty$, can solve the inconsistency problem. Zaman (1984) also discussed that model selection procedures, including those based on popular criteria such as predictive loss and information, lead to inadmissible procedures.

Particularly, when using the L_1 penalized likelihood approach in a linear regression model for variable selection, Leng et.al. (2006) pointed out that when the prediction accuracy is used as the criterion to choose the tuning parameter, the L_1 penalized likelihood procedure does not yield consistent variable selection. Similar to the consistency definition in Shao (1993), a variable selection procedure is consistent if the probability that the procedure correctly identifies the set of important explanatory variables approaches one when the sample size goes to infinity. They showed that "when there are superfluous variables in the linear regression model and the design matrix is orthogonal, the probability of correctly identifying the true set of important variables using prediction-accuracy-based criteria is less than a constant not depending on the sample size". Meinshausen and Buhlmann (2006) used L_1 penalized likelihood approach to perform neighborhood selection in high-dimensional graphs, which is equivalent to variable selection in Gaussian linear models. They also concluded that the optimal tuning parameter value for prediction does not lead to a consistent neighborhood estimate. Instead, they proposed to control the probalility of falsely joining some distinct connectivity components of the graph (i.e., the probability of falsely selecting a variable).

All of the above findings are in the context of linear regression model. For variable selection in survival analysis models using L_1 penalized likelihood procedure, it has not been studied how to choose the tuning parameter value to obtain a consistent result. If the sensitivity, which can be evaluated by the asympotic probability of selecting the true important variables, or the specificity, which can be evaluated by the asympotic probability of falsely selecting a noise variable, can be established, then the optimal tuning parameter value might be further studied.

Bibliography

A Agresti. *Categorical data analysis.* John Wiley & Sons, New York; Chichester, 2003.

H Akaike. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973.

O. Andersen, P. K.and Borgan, R. D. Gill, and N Keiding. *Statistical models based on counting processes.* Springer-Verlag Inc, Berlin; New York, 1993.

P. K. Andersen and R. D. Gill. Cox's regression model for counting processes: a large sample study. *Ann. Statist.*, 10(4):1100–1120, 1982.

D. F. Andrews and A. M. Herzberg. *Data: a collection of problems from many fields for the student and research worker.* Springer-Verlag Inc, Berlin; New York, 1985.

T. M. Beer, C. M. Tangen, L. B. Bland, I. M. Thompson, and E. D. Crawford. Prognostic value of anemia in newly diagnosed metastatic prostate cancer: a multivariate analysis of southwest oncology group study 8894. *J Urol*, 172(6):2213–7, 2004.

R Bender, T Augustin, and M Blettner. Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine*, 24:1713–1723, 2005.

E Biguzzi, F Franchi, P Bucciarelli, M Colombo, and R. Romeo. Endothelial protein C receptor plasma levels increase in chronic liver disease, while thrombomodulin plasma levels increase only in hepatocellular carcinoma. *Thromb Res.*, 120(2):289–93, 2007.

T. M. Block, A. S. Mehta, C. J. Fimmel, and R Jordan. Molecular viral oncology of hepatocellular carcinoma.. *Oncogene.*, 22(33):5093–107, 2003.

N Breslow and J Crowley. A large sample study of the life table and product limit estimates under random censorship. *Ann. Statist.*, 2:437–453, 1974.

B. W. Brown, M. Hollander, and R. M. Korwar. Nonparametric tests of independence for censored data, with applications to heart transplant studies. *Reliability and Biometry*, pages 327–354, 1974.

D. P. Byar and D. K. Corle. Selecting optimal treatment in clinical trials using covariate information. *Journal of Chronic Diseases*, 30:445–459, 1977.

S. C. Cheng, J. P. Fine, and L. J. Wei. Prediction of cumulative incidence function under the proportional hazards model. *Biometrics*, 54:219–228, 1998.

C Chiang. On the probability of death from specific causes in the presence of competing risks. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 4:169–180, 1961.

D Collett. *Modelling survival data in medical research.* Chapman & Hall Ltd, London; New York, 2nd edition, 2003.

D. R. Cox. The analysis of exponentially distributed lifetimes with two types of failure. *J. R. Statist. Soc. B*, 21:411–421, 1959.

D. R. Cox. Regression models and life-tables. *J. Roy. Statist. Soc. Ser. B*, 34:187–220, 1972.

D. R. Cox. Partial likelihood. *Biometrika*, 2:269–276, 1975.

M Crowder. *Classical competing risks.* Chapman & Hall/CRC, London, New York, 2001.

M Crowder. On the identifiability crisis in competing risks analysis. *Scandinavian Journal of Statistics*, 18:223–233, 1991.

H. A. David. On Chiang's proportionality assumption in the theory of competing risks. *Biometrics*, 26:336–339, 1970.

H. A. David and M. L. Moeschberger. *The theory of competing risks.* acmillan Publishing Co Inc, New York; London, 1978.

J. A. Davila, N. J. Petersena, H. A. Nelson, and H. B. El-Serag. Geographic variation within the United States in the incidence of hepatocellular carcinoma. *J Clin Epidemiol.*, 56(5):487–93, 2003.

B Efron. The efficiency of Cox's likelihood function for censored data. *J. Amer. Statist. Assoc.*, pages 557–565, 1977.

B Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78:316–331, 1983.

B Efron. Comments on "jackknife, bootstrap and other resampling methods in regression analysis". *The Annals of Statistics*, 14:1301–1304, 1986.

B Efron and R Tibshirani. *An introduction to the bootstrap.* Chapman & Hall Ltd, London, New York, 1993.

B Efron, T Hastie, I Johnstone, and R Tibshirani. Least angle regression. *The Annals of Statistics*, 32:407–499, 2004.

H. B. El-Serag. Hepatocellular carcinoma: recent trends in the United States. *Gastroenterology*, 5 Suppl 1:S27–34, 2002.

H. B. El-Serag and A. C. Mason. Rising incidence of hepatocellular carcinoma in the United States. *N Engl J Med.*, 340(10):745–50, 1999.

R. C. Elandt-Johnson. Some models in competing risk theory: Multiple causes of single deaths. *Proceedings of the International Biometric Conference, The Biometric Society (Washington)*, 9/1:391–407, 1976.

J Fan and R Li. Variable selection for Cox's proportional hazards model and frailty model. *The Annals of Statistics*, 30:74–99, 2002.

I. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35:109–135, 1993.

C Geyer. On the asymptotics of convex stochastic optimization. *Technical Report . University of Minnesota, Minneapolis*, 1996.

R. D. Gill. Understanding Cox's regression model: a martingale approach. *J. Amer. Statist. Assoc.*, 79(386):441–447, 1984.

G. J. Gores. Hepatocellular carcinoma: gardening strategies and bridges to transplantation. *Liver Transpl.*, 9(2):199–200, 2003.

M Gross-Goupil, R Saffroy, D Azoulay, S Precetti, JF Emile, V Delvart, F Tindilire, A Laurent, MF Bellin, H Bismuth, B Debuire, and A Lemoine. Real-time quantification of AFP mRNA to assess hematogenous dissemination after transarterial chemoembolization of hepatocellular carcinoma. *Ann Surg.*, 238(2):241–8, 2003.

X. Y. Guan, J. S. Sham, L. S. Tai, Y Fang, H Li, and Q Liang. Evidence for another tumor suppressor gene at 17p13.3 distal to tp53 in hepatocellular carcinoma. *Cancer Genet Cytogenet.*, 140(1):45–8, 2003.

J Gui and H. Li. Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. *Bioinformatics*, 21 (13):3001–8, 2005.

B. E. Heckman, J. J.and Honore and BE. Honor. The identifiability of the competing risks model. *Biometrika*, 76:325–330, 1989.

A. E. Hoerl and R. W Kennard. Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

M Hollander and D. A. Wolfe. *Nonparametric statistical methods.* John Wiley & Sons, New York; Chichester, 1999.

P Hougaard. *Analysis of multivariate survival data.* Springer-Verlag, Springer-Verlag, New York, 2000.

R. A. Irizarry, B Hobbs, F Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U Scherf, and T Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data . *Biostatistics (Oxford)*, 4:249–264, 2003.

J. D. Kalbfleisch and R. L. Prentice. *The statistical analysis of failure time data.* Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2nd edition, 2002.

R Kay. Treatment effects in competing-risks analysis of prostate cancer data. *Biometrics*, 42:203–211, 1986.

A. W. Kimball. Models for the estimation of competing risks from grouped data. *Biometrics*, 25:329–337, 1969.

K Knight and W Fu. Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28: 1356–1378, 2000.

S. C Kochar and F Proschan. Independence of time and cause of failure in the multiple dependent competing risks model. *Statistica Sinica*, 1:295–299, 1991.

L. M. Leemis. Variate generation for accelerated life and proportional hazards models. *Operations Research*, 35(6):892–894, 1987.

C Leng, Y Lin, and G Wahba. A note on the lasso and related procedures in model selection. *Statistica Sinica*, 16:1273–1284, 2006.

C Li and H Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–82, 2008.

Z Lin, C Lu, and Z Su. *Fundamental of probability limit theory (Chinese).* Advanced Education Press, Beijing, China, 1999.

M Lunn and D McNeil. A semiparametric mixture model for the analysis of competing risks data. *The Australian Journal of Statistics*, 34:169–180, 1992.

M Lunn and D McNeil. Applying Cox regression to competing risks. *Biometrics*, 51: 524–532, 1995.

C. L. Mallows. Some comments on c_p. *Technometrics*, 15:661–675, 1973.

J. W. Marsh and I Dvorchik. Liver organ allocation for hepatocellular carcinoma: are we sure? *Liver Transpl.*, 9(7):693–6, 2003.

P McCullagh and J. A Nelder. *Generalized linear models.* Chapman & Hall Ltd, London; New York, 1999.

N Meinshausen and P Buhlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34:1436–1462, 2006.

R. H. Myers. *Classical and modern regression with applications.* Duxbury Press, North Scituate, MA, 1990.

J. G. Nalesnik, A. G. Mysliwiec, and E Canby-Hagino. Anemia in men with advanced prostate cancer: incidence, etiology, and treatment. *Rev Urol*, 6(1):1–4, 2004.

S. K. Ng and G. J. McLachlan. An EM-based semi-parametric mixture model approach to the regression analysis of competing-risks data. *Statistics in Medicine*, 22:1097–1111, 2003.

M.Y. Park and T Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 69:659–677, 2007.

R. L. Prentice, J. D. Kalbfleisch, A. V. Peterson, N. Jr Flournoy, V. T. Farewell, and N. E. Breslow. The analysis of failure times in the presence of competing risks. *Biometrics*, 34:541–554, 1978.

G Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 1978.

H. L. Seal. Studies in the history of probability and statistics. xxxv: Multiple decrements or competing risks. *Biometrika*, 64:429–440, 1977.

J Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88:486–494, 1993.

C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9:1135–1151, 1981.

T Suehiro, M Shimada, T Matsumata, A Taketomi, K Yamamoto, and K Sugimachi. Thrombomodulin inhibits intrahepatic spread in human hepatocellular carcinoma . *Hepatology*, 21:1285–90, 1995.

R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B: Methodological*, 58:267–288, 1996.

R Tibshirani. The lasso method for variable selection in the Cox model. *Statistics in Medicine*, 16:385–395, 1997.

T. L. Tseng, Y. P. Shih, Y. C. Huang, C. K. Wang, P. H. Chen, J. G. Chang, K. T. Yeh, Y. M. Chen, and K. H. Buetow. Genotypic and phenotypic characterization of a putative tumor susceptibility gene, GNMT, in liver cancer. *Cancer Res.*, 63(3):647–54, 2003.

A. Tsiatis. A nonidentifiability aspect of the problem of competing risks. *Proc. Natl. Acad. Sci. U.S.A.*, 72:20–22, 1975.

C Wang. *Graph theory (Chinese).* Beijing Institute of Technology Press, Beijing, China, 2nd edition, 1997.

S Wang and S Chow. *Advanced linear models: theory and applications.* Marcel Dekker Inc, New York, 1994.

A. Zaman. Avoiding model selection by the use of shrinkage techniques. *Journal of Econometrics*, 25:73–85, 1984.

Appendix A

Proof of Theorem 2.4

The proof follows that in Chapter 7 of Crowder (2001) with additional details and correction of a mistake.

THEOREM A.1 (Tsiatis (1975))**.** *Suppose that the set of $S(j,t)$ is given for some model with dependent risks. Then there exists a unique proxy model with independent risks yielding identical $S(j,t)$. It is defined by $\overline{G}(\underline{t}) = \prod_{j=1}^{p} \overline{G}^*_j(t_j)$, where $\overline{G}^*_j(t_j) = \exp\left\{-\int_0^t h(j,s)ds\right\}$ and the sub-hazard functiion $h(j,s)$ derives from the given $S(j,t)$.*

Proof: We want to find a function $\overline{G}^*(\underline{t}) = \prod_{j=1}^{p} \overline{G}^*_j(t_j)$ such that $\overline{G}^*(\underline{t})$ is the joint survivor function corresponding to the given $S(j,t)$, that is, from Equation (2.5), we should have $f(j,t) = -\partial\overline{G}^*(\underline{t})/\partial t_j|_{t\underline{1}_p}$ for $j = 1, 2, \cdots, p$. Hence, we need that

$$
\begin{aligned}
f(j,t) &= \frac{\partial\overline{G}^*_j(t_j)\,\overline{G}^*(\underline{t})}{\partial t_j\,\overline{G}^*_j(t_j)}\,|_{t\underline{1}_p} \qquad \text{(A.1)}\\
&= \frac{-d\log\overline{G}^*_j(t)}{dt}\,\overline{G}^*(t\underline{1}_p) \quad .
\end{aligned}
$$

Summing Equation (A.1) over j yields

$$
f(t) = \frac{-d\log\overline{G}^*(t\underline{1}_p)}{dt}\,\overline{G}*(t\underline{1}_p) = \frac{-d\overline{G}^*(t\underline{1}_p)}{dt} \quad . \qquad \text{(A.2)}
$$

Integrating Equation (A.2), we have $S(t) = \overline{G}^*(t\underline{1}_p)$ and from Equation (A.1) , we have

$$\frac{-d\log\overline{G}_j^*(t)}{dt} = \frac{f(j,t)}{S(t)} = h(j,t) \quad . \tag{A.3}$$

We have shown that if the independent-risks proxy model exists, then it has relationship with the given dependent-risks model as above. Now we need to show that the proxy model does exist, which means that we need to show that $\overline{G}_j^*(t)$ $(j = 1, 2, \cdots, p)$ are valid survivor functions and the sub-densities from the proxy model mimic the sub-densities from the given dependent-risks model. It is obvious that as $t \to \infty$, $\overline{G}_j^*(t) \to 0$, and therefore $\overline{G}_j^*(t)$ $(j = 1, 2, \cdots, p)$ are valid survivor functions. To show the latter part, keep in mind Equation (2.5) and that the proxy model has independent risks, so we have

$$\begin{aligned}
g^*(j,t) &= g^*(t)\prod_{j'\neq j}\overline{G}_{j'}(t) \\
&= \frac{-d\overline{G}_j^*(t)}{dt}\,\frac{\prod_{j'}^{p}\overline{G}_{j'}(t)}{\overline{G}_j^*(t)} \\
&= \frac{-d\log\overline{G}_j^*(t)}{dt}\prod_{j'=1}^{p}\overline{G}_{j'}^*(t) \\
&\text{from Equation (A.3)} \\
&= h(j,t)\prod_{j'=1}^{p}\exp\left\{-\int_0^t h(j',s)ds\right\} \\
&= h(j,t)\exp\left\{-\int_0^t h(s)ds\right\} \\
&= h(j,t)S(t) \\
&= f(j,t) \quad .
\end{aligned}$$

□

Appendix B

Proof of Formula (4.13)

Proof: We want to prove that the correlation between a vector on the space Ψ and a vector on the space Ω is as shown in (4.13), where the column vectors of $X_{a\ m}$ are a set of orthonormal base of Ω, and the column vectors of $A = B + X_{a\ m}C_m$ are a set of orthonormal base of Ψ, where B is a orthonormal matrix whose columns span the orthogonal complement space of Ω.

Let $\underline{u}$ be a vector of length k_{n0}, then $X_{a\ m}\underline{u}$ is a vector on the space Ω. Let $\underline{v}$ be a vector of length $(n - k_{n0})$, then $A\underline{v} = (B + X_{a\ m}C_m)\underline{v}$ is a vector on the space of Ψ. The correlation between the two vectors is:

$$r = \frac{\underline{u}' X'_{a\ m} \cdot (B + X_{a\ m}C_m)\underline{v}}{\sqrt{\underline{u}' X'_{a\ m} X_{a\ m}\underline{u}} \cdot \sqrt{\underline{v}'(B + X_{a\ m}C_m)'(B + X_{a\ m}C_m)\underline{v}}} \quad .$$

Since $X_{a\ m}$ is the orthonormal base of Ω, so $X'_{a\ m}X_{a\ m} = I$, where I is the identity matrix of approapriate dimension; similarly, $B'B = I$. Also, sine B is orthogonal to $X_{a\ m}$, so we have $X'_{a\ m}B = \mathbf{0}$, where $\mathbf{0}$ is a matrix of all 0 elements. Therefore,

$$\begin{aligned} r &= \frac{\underline{u}' C_m \underline{u}}{\sqrt{\underline{u}'\underline{u}}\sqrt{(\underline{v}'\underline{v} + \underline{v}'C'_m C_m \underline{v})}} \\ &\le \frac{\sqrt{\underline{u}'\underline{u}}\sqrt{\underline{v}'C'_m C_m \underline{v}}}{\sqrt{\underline{u}'\underline{u}}\sqrt{(\underline{v}'\underline{v} + \underline{v}'C'_m C_m \underline{v})}} \\ &= \sqrt{\frac{\underline{v}'C'_m C_m \underline{v}}{\underline{v}'\underline{v} + \underline{v}'C'_m C_m \underline{v}}} \quad . \end{aligned} \tag{B.1}$$

Note the first inequality in (B.1) holds because of Cauchy's inequality. Hence

$$r^2 \leq \frac{1}{\frac{\underline{v}'\underline{v}}{\underline{v}'C_m'C_m\underline{v}} + 1} \quad . \tag{B.2}$$

Let E be a diagonal matrix whose diagonals are the eigenvalues of $C_m'C_m$ (some eigenvalues of $C_m'C_m$ are 0 since C_m is not of full rank), and let P be the orthonormal matrix (of dimension $(n - k_{n0}) \times (n - k_{n0})$) whose columns are the eigenvectors for matrix $C_m'C_m$. Therefore, by eigen decomposition theorem, we have $C_m'C_m = PEP^{-1}$, and the quadratic form $\underline{v}'C_m'C_m\underline{v}$ in (B.2) is

$$\begin{aligned} \underline{v}'C_m'C_m\underline{v} &= \underline{v}'PEP^{-1}\underline{v} \\ &\leq \underline{v}'PE_\lambda P^{-1}\underline{v} \\ &= e_\lambda^2\underline{v}'PIP^{-1}\underline{v} \\ &= e_\lambda^2\underline{v}'\underline{v} \quad , \end{aligned}$$

where E_λ is the diagonal matrix as E, but the nonzero diagonals are replaced by e_λ^2, the largest eigenvalue of $C_m'C_m$.

So in (B.2),

$$r^2 \leq \frac{1}{\frac{1}{e_\lambda^2} + 1} \quad ,$$

and hence (4.13) holds, that is, the correlation between $\underline{u}$ and $\underline{v}$

$$r \leq \frac{e_\lambda}{\sqrt{1 + e_\lambda^2}} \quad .$$

□

Appendix C

Source Code for the Simulation Study in Chapter 4

C.1 Simulation When the Truly Important Variables Are Independent of the Noise Variables - Using the Proposed Method

```
#########################################################
#########################################################
#PURPOSE: SIMULATION IN DISSERTAION: L1 PENALIZED MAXIMUM
#LIKELIHOOD APPROACH IN COMPETING RISKS
#Note:GENERATION OF X USING DECOMPOSITION: TRULY
#IMPORTANT VARIABLES ARE INDEPENDENT OF NOISE VARIABLES
#BY:      XIANGRONG KONG
#LAST MODIFIED DATE:    July 2nd, 2008
########################################################
########################################################
setwd("C:/talaci/research/thesis/data")
#setwd("C:/Kong/simulation/data/May17")


library(survival)
library(glmpath)
library(MASS)

memory.limit(size=4000)
#######################################################
run<-50                #the no. of simulation runs
p.censor<-0.15         #proportion of censoring
B<-100
########################################################
```

```
alpha<-0.05

sim.func<-function(n=50, k=100,p.risk1=0.5,p.censor=0.15)
        ###The correlation coefficient matrix for
        ### x - Identity matrix now
        rho<-diag(1,nzero,nzero)


        ###Design Matrix
        x.a<-matrix(rep(0,n*nzero),ncol=nzero)
        x.a<-mvrnorm(n = n, mu=rep(0,nzero), Sigma=rho)
        colnames(x.a)<-paste("x",1:nzero,sep="")


        B<-qr.Q(qr(x.a),complete=TRUE)[,-c(1:nzero)]
        C<-mvrnorm(n = (n-nzero), mu=rep(0,(k-nzero)),
                Sigma=diag(1,(k-nzero),(k-nzero)))

        x.b<-B%*%C
        colnames(x.b)<-paste("x",(nzero+1):k,sep="")

        #max(t(x.b)%*%x.a)

        x<-cbind(x.a,x.b)

        lambda1<-exp(x%*%beta1)
        set.seed(123*sample(1:1000,1)+i*10)
        u<-runif(n)
        base.t<-rweibull(n, shape=5, scale = 2)

        t1<--log(u)/(lambda1*base.t)


        lambda2<-exp(x%*%beta2)
        set.seed(2345*sample(1000:2000,1)+i*10)
        u<-runif(n)
        t2<--log(u)/(lambda2*base.t)

        ###Generate group indicator indicating fail
        ###due to cause 1 or cause 2; and true censoring
        ### indicator
        group.ind<-rbinom(n, 1, p.risk1)
                #1 if group1 and 0 if group2
        censor.ind<-rbinom(n,1,1-p.censor)
                #1 if event and 0 if censored
        censor.t<-runif(n,2,10)


        ###Get the observed days
        days<-ifelse(group.ind==1, t1, t2)
        days<-ifelse(censor.ind==0,censor.t,days)
                #censored times are from a uniform
                #distribution

        #cbind(days, group.ind, censor.ind)
        #days<-(days-mean(days))/sqrt(days)

        ###Preparing for estimation: creat cause
        ###specific censoring indicators
        censor.ind.1<-ifelse(group.ind==1
                & censor.ind!=0, 1, 0)
        censor.ind.2<-ifelse(group.ind!=1
                & censor.ind!=0, 1, 0)

        ###Estimation using coxpath for cause 1
        data.c1<-list(x=x, time=days
                , status=censor.ind.1)
        coxpath.c1<-coxpath(data=data.c1
                ,standardize = TRUE,trace = F)

        ###Estimation using coxpath for cause 2

        data.c2<-list(x=x, time=days,
                 status=censor.ind.2)
        coxpath.c2<-coxpath(data=data.c2,
                standardize = TRUE,trace = F)

        ###Report
        result<-list(coxpath.c1,coxpath.c2)
        return(result)
}

#########################################################
#SITUATION1: N=100,K=10,P.RISK1=0.6
#########################################################
n<-100                      #total number of observations
k<-10                       #number of covariates
p.risk1<-0.6                #proportion of observations fail
                            #due to risk 1
```

```
nzero.effect1<-2        #the non-zero coefficient effect
                        # of beta
nzero.effect2<-2
nzero<-4
run<-20

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")


###
###
result.c1.r1<-list()
result.c2.r1<-list()

coef.c1.r1<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r1<-matrix(rep(NA,run*k),ncol=k)

check.c1.r1<-matrix(rep(NA,run*k),ncol=k)
check.c2.r1<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r1<-c()
spec.c1.r1<-c()

sens.c2.r1<-c()
spec.c2.r1<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1

        result.c2.r1[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                #length(result.c1.r1[[i]]$lambda)-1
                ind<-length(result.c1.r1[[i]]$lambda
                [result.c1.r1[[i]]$lambda>threshold[j]])
                coef.c1.r1[i,]<-result.c1.r1[[i]]$
                b.corrector[ind,]

                #length(result.c2.r1[[i]]$lambda)-1
                ind<-length(result.c2.r1[[i]]$lambda
                [result.c2.r1[[i]]$lambda>threshold[j]])
                coef.c2.r1[i,]<-result.c2.r1[[i]]$
                b.corrector[ind,]

                check.c1.r1[i,]<-ifelse(coef.c1.r1[i,]
                !=0,1,0)
                check.c2.r1[i,]<-ifelse(coef.c2.r1[i,]
                !=0,1,0)

        }
        sens.c1.r1[j]<-sum(as.vector(check.c1.r1
                [,c(1:nzero)]))/(nzero*run)

        spec.c1.r1[j]<-1-sum(as.vector(check.c1.r1
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r1[j]<-sum(as.vector(check.c2.r1
                [,c(1:nzero)]))/(nzero*run)

        spec.c2.r1[j]<-1-sum(as.vector(check.c2.r
                [,c((nzero+1):k)]))/((k-nzero)*run)

}



#########################################################
#SITUATION2: N=100,K=50,P.RISK1=0.6             #
#########################################################
n<-100                  #total number of observations
k<-50                   #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
```

```
run<-20

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")


###
###
result.c1.r2<-list()
result.c2.r2<-list()

coef.c1.r2<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r2<-matrix(rep(NA,run*k),ncol=k)

check.c1.r2<-matrix(rep(NA,run*k),ncol=k)
check.c2.r2<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r2<-c()
spec.c1.r2<-c()

sens.c2.r2<-c()
spec.c2.r2<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k
        ,p.risk1=p.risk1,p.censor=p.censor
        , beta1=beta1, beta2=beta2,B=B)
        result.c1.r2[[i]]<-temp[[1]]
        result.c2.r2[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r2[[i]]
        $lambda[result.c1.r2[[i]]$lambda>threshold[j]])
                coef.c1.r2[i,]<-result.c1.r2[[i]]
        $b.corrector[ind,]
```

```
                ind<-length(result.c2.r2[[i]]
        $lambda[result.c2.r2[[i]]$lambda>threshold[j]])
                coef.c2.r2[i,]<-result.c2.r2[[i]]
        $b.corrector[ind,]

                check.c1.r2[i,]<-ifelse(coef.c1.r2[i,]
        !=0,1,0)
                check.c2.r2[i,]<-ifelse(coef.c2.r2[i,]
        !=0,1,0)

        }
        sens.c1.r2[j]<-sum(as.vector(check.c1.r2[,
        c(1:nzero)]))/(nzero*run)

        spec.c1.r2[j]<-1-sum(as.vector(check.c1.r2[,
        c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r2[j]<-sum(as.vector(check.c2.r2[,
        c(1:nzero)]))/(nzero*run)

        spec.c2.r2[j]<-1-sum(as.vector(check.c2.r2[,
        c((nzero+1):k)]))/((k-nzero)*run)

}




#########################################################
#SITUATION3: N=100,K=100,P.RISK1=0.6,EFFECT1=2, EFFECT2=2               #
#########################################################
n<-100                  #total number of observations
k<-100                  #number of covariates
p.risk1<-0.6            #proportion of observations
                        # fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        # effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2
```

```
###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r3<-list()
result.c2.r3<-list()

coef.c1.r3<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r3<-matrix(rep(NA,run*k),ncol=k)

check.c1.r3<-matrix(rep(NA,run*k),ncol=k)
check.c2.r3<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r3<-c()
spec.c1.r3<-c()

sens.c2.r3<-c()
spec.c2.r3<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
        ,p.censor=p.censor, beta1=beta1
        , beta2=beta2,B=B)
        result.c1.r3[[i]]<-temp[[1]]
        result.c2.r3[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r3[[i]]
                $lambda[result.c1.r3[[i]]$lambda
                >threshold[j]])
                coef.c1.r3[i,]<-result.c1.r3[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r3[[i]]
                $lambda[result.c2.r3[[i]]$lambda
                >threshold[j]])
                coef.c2.r3[i,]<-result.c2.r3[[i]]
                $b.corrector[ind,]

                check.c1.r3[i,]<-ifelse(coef.c1.r3
                [i,]!=0,1,0)
                check.c2.r3[i,]<-ifelse(coef.c2.r3
                [i,]!=0,1,0)

        }
        sens.c1.r3[j]<-sum(as.vector(check.c1.r3
                [,c(1:nzero)]))/(nzero*run)

        spec.c1.r3[j]<-1-sum(as.vector(check.c1.r3
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r3[j]<-sum(as.vector(check.c2.r3
                [,c(1:nzero)]))/(nzero*run)

        spec.c2.r3[j]<-1-sum(as.vector(check.c2.r3
                [,c((nzero+1):k)]))/((k-nzero)*run)

}


#########################################################
#SITUATION4: N=100,K=200,P.RISK1=0.6,EFFECT1=2, EFFECT2=2
#########################################################
n<-100                  #total number of observations
k<-200                  #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")
```

```
beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")


###
###
result.c1.r4<-list()
result.c2.r4<-list()

coef.c1.r4<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r4<-matrix(rep(NA,run*k),ncol=k)

check.c1.r4<-matrix(rep(NA,run*k),ncol=k)
check.c2.r4<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r4<-c()
spec.c1.r4<-c()

sens.c2.r4<-c()
spec.c2.r4<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r4[[i]]<-temp[[1]]
        result.c2.r4[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r4[[i]]
                $lambda[result.c1.r4[[i]]
                $lambda>threshold[j]])
                coef.c1.r4[i,]<-result.c1.r4[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r4[[i]]
                $lambda[result.c2.r4[[i]]$lambda
                >threshold[j]])
                coef.c2.r4[i,]<-result.c2.r4[[i]]
                $b.corrector[ind,]

                check.c1.r4[i,]<-ifelse(coef.c1.r4
                [i,]!=0,1,0)
                check.c2.r4[i,]<-ifelse(coef.c2.r4
                [i,]!=0,1,0)

        }
        sens.c1.r4[j]<-sum(as.vector(check.c1.r4[,
                c(1:nzero)]))/(nzero*run)

        spec.c1.r4[j]<-1-sum(as.vector(check.c1.r4[,
                c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r4[j]<-sum(as.vector(check.c2.r4[,
                c(1:nzero)]))/(nzero*run)

        spec.c2.r4[j]<-1-sum(as.vector(check.c2.r4[,
                c((nzero+1):k)]))/((k-nzero)*run)

}


save.image("SITUATION␣1␣to␣4-composition␣x.RData")

#########################################################
#SITUATION5: N=200,K=10,P.RISK1=0.6,EFFECT1=2, EFFECT2=2
#########################################################
n<-200                  #total number of observations
k<-10                   #number of covariates
p.risk1<-0.6            #proportion of observations fail
                        #due to risk 1
nzero.effect1<-2        #the non-zero coefficient effect
                        # of beta
nzero.effect2<-2
nzero<-4
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")
```

```
beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")


###
###
result.c1.r5<-list()
result.c2.r5<-list()

coef.c1.r5<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r5<-matrix(rep(NA,run*k),ncol=k)

check.c1.r5<-matrix(rep(NA,run*k),ncol=k)
check.c2.r5<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r5<-c()
spec.c1.r5<-c()

sens.c2.r5<-c()
spec.c2.r5<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta
                1, beta2=beta2,B=B)
        result.c1.r5[[i]]<-temp[[1]]
        result.c2.r5[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r5[[i]]
                $lambda[result.c1.r5[[i]]$lambda
                >threshold[j]])
                coef.c1.r5[i,]<-result.c1.r5[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r5[[i]]$
                lambda[result.c2.r5[[i]]$lambda
                >threshold[j]])
                coef.c2.r5[i,]<-result.c2.r5[[i]]
                $b.corrector[ind,]

                check.c1.r5[i,]<-ifelse(coef.c1.r5[i,
                ]!=0,1,0)
                check.c2.r5[i,]<-ifelse(coef.c2.r5[i,
                ]!=0,1,0)

        }
        sens.c1.r5[j]<-sum(as.vector(check.c1.r5[,c
                (1:nzero)]))/(nzero*run)

        spec.c1.r5[j]<-1-sum(as.vector(check.c1.r5[,c
                ((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r5[j]<-sum(as.vector(check.c2.r5[,c(
                1:nzero)]))/(nzero*run)

        spec.c2.r5[j]<-1-sum(as.vector(check.c2.r5
                [,c((nzero+1):k)]))/((k-nzero)*run)

}


#########################################################
#SITUATION6: N=200,K=50,P.RISK1=0.6             #
#########################################################
n<-200                  #total number of observations
k<-50                   #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
```

```
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r6<-list()
result.c2.r6<-list()

coef.c1.r6<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r6<-matrix(rep(NA,run*k),ncol=k)

check.c1.r6<-matrix(rep(NA,run*k),ncol=k)
check.c2.r6<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r6<-c()
spec.c1.r6<-c()

sens.c2.r6<-c()
spec.c2.r6<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r6[[i]]<-temp[[1]]
        result.c2.r6[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r6[[i]]
                $lambda[result.c1.r6[[i]]$lambda>threshold[j]])
                coef.c1.r6[i,]<-result.c1.r6[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r6[[i]]
                $lambda[result.c2.r6[[i]]$
                        lambda>threshold[j]])
                coef.c2.r6[i,]<-result.c2.r6[
                [i]]$b.corrector[ind,]

                check.c1.r6[i,]<-ifelse(coef.c1.r6[i,]
                !=0,1,0)
                check.c2.r6[i,]<-ifelse(coef.c2.r6[i,]
                !=0,1,0)

        }
        sens.c1.r6[j]<-sum(as.vector(check.c1.r6[,
                c(1:nzero)]))/(nzero*run)

        spec.c1.r6[j]<-1-sum(as.vector(check.c1.r6[,c((nzero+1):k)]))/((k


        sens.c2.r6[j]<-sum(as.vector(check.c2.r6[
                ,c(1:nzero)]))/(nzero*run)

        spec.c2.r6[j]<-1-sum(as.vector(check.c2.r6[,
                c((nzero+1):k)]))/((k-nzero)*run)

}


#########################################################
#SITUATION7: N=200,K=100,P.RISK1=0.6            #
#########################################################
n<-200                  #total number of observations
k<-100                  #number of covariates
p.risk1<-0.6            #proportion of observations
                        # fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r7<-list()
result.c2.r7<-list()

coef.c1.r7<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r7<-matrix(rep(NA,run*k),ncol=k)

check.c1.r7<-matrix(rep(NA,run*k),ncol=k)
check.c2.r7<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r7<-c()
spec.c1.r7<-c()

sens.c2.r7<-c()
spec.c2.r7<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1,
                 beta2=beta2,B=B)
        result.c1.r7[[i]]<-temp[[1]]
        result.c2.r7[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r7[[i]]
                $lambda[result.c1.r7[[i]]$lambda
                >threshold[j]])
                coef.c1.r7[i,]<-result.c1.r7[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r7[[i]]$
                lambda[result.c2.r7[[i]]$lambda>threshold[j]])
                coef.c2.r7[i,]<-result.c2.r7[[i]]
                $b.corrector[ind,]

                check.c1.r7[i,]<-ifelse(coef.c1.r7
                [i,]!=0,1,0)
                check.c2.r7[i,]<-ifelse(coef.c2.r7
                [i,]!=0,1,0)

        }
        sens.c1.r7[j]<-sum(as.vector(check.c1.r7
                [,c(1:nzero)]))/(nzero*run)

        spec.c1.r7[j]<-1-sum(as.vector(check.c1.r7
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r7[j]<-sum(as.vector(check.c2.r7
                [,c(1:nzero)]))/(nzero*run)

        spec.c2.r7[j]<-1-sum(as.vector(check.c2.r7
                [,c((nzero+1):k)]))/((k-nzero)*run)

}


#########################################################
#SITUATION8: N=200,K=200,P.RISK1=0.6            #
#########################################################
n<-200                  #total number of observations
k<-200                  #number of covariates
p.risk1<-0.6            #proportion of observations
                        # fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r8<-list()
result.c2.r8<-list()

coef.c1.r8<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r8<-matrix(rep(NA,run*k),ncol=k)
```

```
check.c1.r8<-matrix(rep(NA,run*k),ncol=k)
check.c2.r8<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r8<-c()
spec.c1.r8<-c()

sens.c2.r8<-c()
spec.c2.r8<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r8[[i]]<-temp[[1]]
        result.c2.r8[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r8[[i]]
                $lambda[result.c1.r8[[i]]$lambda
                >threshold[j]])
                coef.c1.r8[i,]<-result.c1.r8[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r8[[i]]$
                lambda[result.c2.r8[[i]]$lambd
                        a>threshold[j]])
                coef.c2.r8[i,]<-result.c2.r8[[i]]
                $b.corrector[ind,]

                check.c1.r8[i,]<-ifelse(coef.c1.r8
                [i,]!=0,1,0)
                check.c2.r8[i,]<-ifelse(coef.c2.r8
                [i,]!=0,1,0)

        }
        sens.c1.r8[j]<-sum(as.vector(check.c1.r8
                [,c(1:nzero)]))/(nzero*run)
        spec.c1.r8[j]<-1-sum(as.vector(check.c1.r8
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r8[j]<-sum(as.vector(check.c2.r8[,
                c(1:nzero)]))/(nzero*run)

        spec.c2.r8[j]<-1-sum(as.vector(check.c2.r8[,
                c((nzero+1):k)]))/((k-nzero)*run)

}

save.image("SITUATION␣5␣to␣8-composition␣x.RData")


#########################################################
#SITUATION9: N=200,K=500,P.RISK1=0.6             #
#########################################################
n<-200                  #total number of observations
k<-500                  #number of covariates
p.risk1<-0.6            #proportion of observations
                        # fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r9<-list()
result.c2.r9<-list()

coef.c1.r9<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r9<-matrix(rep(NA,run*k),ncol=k)
```

```
check.c1.r9<-matrix(rep(NA,run*k),ncol=k)
check.c2.r9<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r9<-c()
spec.c1.r9<-c()

sens.c2.r9<-c()
spec.c2.r9<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r9[[i]]<-temp[[1]]
        result.c2.r9[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r9[[i]]$lambda
                [result.c1.r9[[i]]$lambda
                >threshold[j]])
                coef.c1.r9[i,]<-result.c1.r9[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r9[[i]]
                $lambda[result.c2.r9[[i]]$lambda
                >threshold[j]])
                coef.c2.r9[i,]<-result.c2.r9[[i]]
                $b.corrector[ind,]

                check.c1.r9[i,]<-ifelse(coef.c1.r9
                [i,]!=0,1,0)
                check.c2.r9[i,]<-ifelse(coef.c2.r9
                [i,]!=0,1,0)

        }
        sens.c1.r9[j]<-sum(as.vector(check.c1.r9[,
                c(1:nzero)]))/(nzero*run)

        spec.c1.r9[j]<-1-sum(as.vector(check.c1.r9[,
                c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r9[j]<-sum(as.vector(check.c2.r9[,
                c(1:nzero)]))/(nzero*run)

        spec.c2.r9[j]<-1-sum(as.vector(check.c2.r9[,
                c((nzero+1):k)]))/((k-nzero)*run)

}

#########################################################
#SITUATION10: N=200,K=1000,P.RISK1=0.6          #
#########################################################
n<-200                  #total number of observations
k<-1000                 #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient effect
                        # of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r10<-list()
result.c2.r10<-list()

coef.c1.r10<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r10<-matrix(rep(NA,run*k),ncol=k)

check.c1.r10<-matrix(rep(NA,run*k),ncol=k)
check.c2.r10<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r10<-c()
```

```
spec.c1.r10<-c()

sens.c2.r10<-c()
spec.c2.r10<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r10[[i]]<-temp[[1]]
        result.c2.r10[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r10[[i]]
                $lambda[result.c1.r10[[i]]$lambda
                >threshold[j]])
                coef.c1.r10[i,]<-result.c1.r10[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r10[[i]]
                $lambda[result.c2.r10[[i]]$lambda
                >threshold[j]])
                coef.c2.r10[i,]<-result.c2.r10
                [[i]]$b.corrector[ind,]

                check.c1.r10[i,]<-ifelse(coef.c1.r10
                [i,]!=0,1,0)
                check.c2.r10[i,]<-ifelse(coef.c2.r10
                [i,]!=0,1,0)

        }
        sens.c1.r10[j]<-sum(as.vector(check.c1.r10
                [,c(1:nzero)]))/(nzero*run)

        spec.c1.r10[j]<-1-sum(as.vector(check.c1.r10
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r10[j]<-sum(as.vector(check.c2.r10
                [,c(1:nzero)]))/(nzero*run)

        spec.c2.r10[j]<-1-sum(as.vector(check.c2.r10
                [,c((nzero+1):k)]))/((k-nzero)*run)

}

save.image("SITUATION␣9␣to␣10-composition␣x.RData")


###################################################
###################################################
#ADD: SITUATION5A N=100,K=500 AND SITUATION6A
# N=100, K=1000
###################################################
####################################################


########################################################
#SITUATION5A: N=200,K=500,P.RISK1=0.6                  #
########################################################
n<-100                 #total number of observations
k<-500                 #number of covariates
p.risk1<-0.6           #proportion of observations
                       # fail due to risk 1
nzero.effect1<-2       #the non-zero coefficient
                       #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r5a<-list()
result.c2.r5a<-list()
```

```
coef.c1.r5a<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r5a<-matrix(rep(NA,run*k),ncol=k)

check.c1.r5a<-matrix(rep(NA,run*k),ncol=k)
check.c2.r5a<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r5a<-c()
spec.c1.r5a<-c()

sens.c2.r5a<-c()
spec.c2.r5a<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r5a[[i]]<-temp[[1]]
        result.c2.r5a[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r5a[[i]]
                $lambda[result.c1.r5a[[i]]$lambda
                >threshold[j]])
                coef.c1.r5a[i,]<-result.c1.r5a[[i]]
                $b.corrector[ind,]

                ind<-length(result.c2.r5a[[i]]
                $lambda[result.c2.r5a[[i]]$lambda
                >threshold[j]])
                coef.c2.r5a[i,]<-result.c2.r5a[[i]]
                $b.corrector[ind,]

                check.c1.r5a[i,]<-ifelse(coef.c1.r5a
                [i,]!=0,1,0)
                check.c2.r5a[i,]<-ifelse(coef.c2.r5a
                [i,]!=0,1,0)

        }
        sens.c1.r5a[j]<-sum(as.vector(check.c1.r5a[,
                c(1:nzero)]))/(nzero*run)

        spec.c1.r5a[j]<-1-sum(as.vector(check.c1.r5a[,
                c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r5a[j]<-sum(as.vector(check.c2.r5a[,
                c(1:nzero)]))/(nzero*run)

        spec.c2.r5a[j]<-1-sum(as.vector(check.c2.r5a[,
                c((nzero+1):k)]))/((k-nzero)*run)

}

#########################################################
#SITUATION6A: N=200,K=1000,P.RISK1=0.6          #
#########################################################
n<-100                  #total number of observations
k<-1000                 #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r6a<-list()
result.c2.r6a<-list()

coef.c1.r6a<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r6a<-matrix(rep(NA,run*k),ncol=k)

check.c1.r6a<-matrix(rep(NA,run*k),ncol=k)
check.c2.r6a<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r6a<-c()
spec.c1.r6a<-c()

sens.c2.r6a<-c()
spec.c2.r6a<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1,
                p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B)
        result.c1.r6a[[i]]<-temp[[1]]
        result.c2.r6a[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r6a[[i]]$
                lambda[result.c1.r6a[[i]]$lambda
                >threshold[j]])
                coef.c1.r6a[i,]<-result.c1.r6a[[i]]
                        $b.corrector[ind,]

                ind<-length(result.c2.r6a[[i]]
                $lambda[result.c2.r6a[[i]]$lambda
                >threshold[j]])
                coef.c2.r6a[i,]<-result.c2.r6a[[i]]
                $b.corrector[ind,]

                check.c1.r6a[i,]<-ifelse(coef.c1.r6a
                [i,]!=0,1,0)
                check.c2.r6a[i,]<-ifelse(coef.c2.r6a
                [i,]!=0,1,0)

        }
        sens.c1.r6a[j]<-sum(as.vector(check.c1.r6a[,
                c(1:nzero)]))/(nzero*run)

        spec.c1.r6a[j]<-1-sum(as.vector(check.c1.r6a[,
                c((nzero+1):k)]))/((k-nzero)*run)
        sens.c2.r6a[j]<-sum(as.vector(check.c2.r6a[,
                c(1:nzero)]))/(nzero*run)

        spec.c2.r6a[j]<-1-sum(as.vector(check.c2.r6a[,
                c((nzero+1):k)]))/((k-nzero)*run)

}

save.image("SITUATION␣5A␣to␣6A-composition␣x.RData")
```

C.2 Simulation When the Truly Important Variables Are Independent Correlated (maximum correlation 0.8) - Using the Proposed Method

```
###################################################################
###################################################################
#PURPOSE: SIMULATION IN DISSERTAION: L1 PENALIZED MAXIMUM
#        LIKELIHOOD APPROACH IN COMPETING RISKS
#NOTE: GENERATION OF X MATRIX USING DECOMPOSITION, TRULY
#        IMPORTANT
#        VARIABLES ARE CORRELATED WITH NOISE
#      VARIABLES WITH MAXIMUM CORRELATION BEING 0.8
#BY:       XIANGRONG KONG
#LAST MODIFIED DATE:    July 4TH, 2008
#########################################################
##########################################################
setwd("C:/talaci/research/thesis/data")
#setwd("C:/Kong/simulation/data/May17")

library(survival)
library(glmpath)
library(MASS)

memory.limit(size=4000)

########################################################
run<-50                 #the no. of simulation runs
p.censor<-0.15          #proportion of censoring
B<-100
max.corr<-0.8
max.eigen<-max.corr^2/(1-max.corr^2)

#########################################################
alpha<-0.05

sim.func<-function(n=50, k=100,p.risk1=0.5,p.censor=0.15
                , beta1, beta2,B, max.eigen){  #this is
                #the function for simmulation

        ###The correlation coefficient matrix for x
                # - Identity matrix now
        rho<-diag(1,nzero,nzero)
```

```
###Design Matrix
x.a<-matrix(rep(0,n*nzero),ncol=nzero)
x.a<-mvrnorm(n = n, mu=rep(0,nzero), Sigma=rho)
colnames(x.a)<-paste("x",1:nzero,sep="")

x.am<-qr.Q(qr(x.a))
        #This is the ortho-normal basis of the
        #space Omega expanded by x.a

B<-qr.Q(qr(x.a),complete=TRUE)[,-c(1:nzero)]
        #This is an ortho-normal basis of
        # the orthogonal complement space of
        #Omega


Cm.Cm<-matrix(rep(0, (n-nzero)*(n-nzero))
, ncol=n-nzero)
diag(Cm.Cm)<-seq(from=max.eigen, to=0,
length.out=n-nzero)
                #Then the largest eigenvalue
                #of (Cm’Cm) is 1/3
Cm<-((Cm.Cm)^(1/2))[1:nzero,]

A<-B+x.am%*%Cm
        #This is a basis for the space Psi this
        #is not orthogonal complement
        #of Omega. Use A to generate the
        #design matrix
        # X.b for noise variables.

C<-mvrnorm(n = (n-nzero), mu=rep(0,(k-nzero))
        , Sigma=diag(1,(k-nzero),(k-nzero)))


x.b<-A%*%C
#This is the design matrix for the noise
#variables
colnames(x.b)<-paste("x",(nzero+1):k,sep="")

x<-cbind(x.a, x.b)


lambda1<-exp(x%*%beta1)
```

```
set.seed(123*sample(1:1000,1)+i*10)
u<-runif(n)
base.t<-rweibull(n, shape=5, scale = 2)

t1<--log(u)/(lambda1*base.t)


lambda2<-exp(x%*%beta2)
set.seed(2345*sample(1000:2000,1)+i*10)
u<-runif(n)
t2<--log(u)/(lambda2*base.t)

###Generate group indicator indicating
### fail due to cause 1 or cause 2;
###and true censoring indicator
group.ind<-rbinom(n, 1, p.risk1)
        #1 if group1 and 0 if group2
censor.ind<-rbinom(n,1,1-p.censor)
        #1 if event and 0 if censored

censor.t<-runif(n,2,10)


###Get the observed days
days<-ifelse(group.ind==1, t1, t2)
days<-ifelse(censor.ind==0,censor.t,days)
        #censored times are from a uniform
        # distribution

#cbind(days, group.ind, censor.ind)
#days<-(days-mean(days))/sqrt(days)

###Preparing for estimation: creat cause
###specific
###censoring indicators
censor.ind.1<-ifelse(group.ind==1 &
        censor.ind!=0, 1, 0)
censor.ind.2<-ifelse(group.ind!=1 &
        censor.ind!=0, 1, 0)

###Estimation using coxpath for cause 1
data.c1<-list(x=x, time=days, status=
                censor.ind.1)
coxpath.c1<-coxpath(data=data.c1
                ,standardize = TRUE
        ,trace = F)

        ###Estimation using coxpath for cause 2

        data.c2<-list(x=x, time=days,
                        status=censor.ind.2)
        coxpath.c2<-coxpath(data=data.c2
                        ,standardize = TRUE
        ,trace = F)

        ###Report
        result<-list(coxpath.c1,coxpath.c2)
        return(result)
}


#########################################################
#SITUATION10: N=200,K=1000,P.RISK1=0.6          #
#########################################################
n<-200                  #total number of observations
k<-1000                 #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r10<-list()
result.c2.r10<-list()

coef.c1.r10<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r10<-matrix(rep(NA,run*k),ncol=k)
```

```
check.c1.r10<-matrix(rep(NA,run*k),ncol=k)
check.c2.r10<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r10<-c()
spec.c1.r10<-c()

sens.c2.r10<-c()
spec.c2.r10<-c()

threshold<-c(n^(0.3),n^(0.1))

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                ,p.censor=p.censor, beta1=beta1
                , beta2=beta2,B=B, max.eigen=max.eigen)
        result.c1.r10[[i]]<-temp[[1]]
        result.c2.r10[[i]]<-temp[[2]]

}

for(j in 1:length(threshold)){
        for(i in 1:run){

                ind<-length(result.c1.r10[[i]]$
                lambda[result.c1.r10[[i]]$lambda
                >threshold[j]])
                coef.c1.r10[i,]<-result.c1.r10[[i]
                ]$b.corrector[ind,]

                ind<-length(result.c2.r10[[i]]$
                lambda[result.c2.r10[[i]]$lambda
                >threshold[j]])
                coef.c2.r10[i,]<-result.c2.r10[[i]]
                $b.corrector[ind,]


                check.c1.r10[i,]<-ifelse(coef.c1.r10
                [i,]!=0,1,0)
                check.c2.r10[i,]<-ifelse(coef.c2.r10
                [i,]!=0,1,0)

        }
        sens.c1.r10[j]<-sum(as.vector(check.c1.r10
                [,c(1:nzero)]))/(nzero*run)
        spec.c1.r10[j]<-1-sum(as.vector(check.c1.r10
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r10[j]<-sum(as.vector(check.c2.r10
                [,c(1:nzero)]))/(nzero*run)

        spec.c2.r10[j]<-1-sum(as.vector(check.c2.r10
        [,c((nzero+1):k)]))/((k-nzero)*run)

}

save.image("SITUATION␣9␣to␣10-composition␣x.RData")
```

C.3 Simulation When the Truly Important Variables Are Independent of the Noise Variables - Using Univariable Apporach

```
#######################################################
#######################################################
#PURPOSE: SIMULATION IN DISSERTAION: UNIVARIABLE COX
# MODEL APPROACH
#NOTE: GENERATION OF X USING DECOMPOSITION
#BY:      XIANGRONG KONG
#LAST MODIFIED DATE:    July 2nd, 2008
#######################################################
#######################################################
setwd("C:/talaci/research/thesis/data")
#setwd("C:/Kong/simulation/data/May17")


library(survival)
library(glmpath)
library(MASS)

memory.limit(size=4000)

#######################################################
run<-20                 #the no. of simulation runs
p.censor<-0.15          #proportion of censoring
B<-100
########################################################
alpha<-c(0.01,0.05)

sim.func<-function(n=50, k=100,p.risk1=0.5,p.censor=0.15,
         beta1, beta2){#this is the simmulation function

        ###The correlation coefficient matrix for x
        ###- Identity matrix now
        rho<-diag(1,nzero,nzero)


        ###Design Matrix
        x.a<-matrix(rep(0,n*nzero),ncol=nzero)
        x.a<-mvrnorm(n = n, mu=rep(0,nzero), Sigma=rho)
        colnames(x.a)<-paste("x",1:nzero,sep="")
```

```
B<-qr.Q(qr(x.a),complete=TRUE)[,-c(1:nzero)]
C<-mvrnorm(n = (n-nzero), mu=rep(0,(k-nzero))
        , Sigma=diag(1,(k-nzero),(k-nzero)))

x.b<-B%*%C
colnames(x.b)<-paste("x",(nzero+1):k,sep="")

#max(t(x.b)%*%x.a)

x<-cbind(x.a,x.b)

lambda1<-exp(x%*%beta1)
set.seed(123*sample(1:1000,1)+i*10)
u<-runif(n)
base.t<-rweibull(n, shape=5, scale = 2)

t1<--log(u)/(lambda1*base.t)


lambda2<-exp(x%*%beta2)
set.seed(2345*sample(1000:2000,1)+i*10)
u<-runif(n)
t2<--log(u)/(lambda2*base.t)

###Generate group indicator indicating fail due
### to cause 1 or cause 2; and true censoring
###indicator
group.ind<-rbinom(n, 1, p.risk1)
        #1 if group1 and 0 if group2
censor.ind<-rbinom(n,1,1-p.censor)
        #1 if event and 0 if censored

censor.t<-runif(n,2,10)


###Get the observed days
days<-ifelse(group.ind==1, t1, t2)
days<-ifelse(censor.ind==0,censor.t,days)
        #censored times are from a uniform
        #distribution

#cbind(days, group.ind, censor.ind)
#days<-(days-mean(days))/sqrt(days)
```

```
###Preparing for estimation: creat cause
###specific censoring indicators
censor.ind.1<-ifelse(group.ind==1 & censor.ind!=0
                , 1, 0)
censor.ind.2<-ifelse(group.ind!=1 & censor.ind!=0
                , 1, 0)

###Estimation using coxpath for cause 1
data.c1<-list(x=x, time=days
        , status=censor.ind.1)
coxpath.c1<-coxpath(data=data.c1
        ,standardize = TRUE,trace = F)


###Estimation using coxpath for cause 2

data.c2<-list(x=x, time=days
        , status=censor.ind.2)
coxpath.c2<-coxpath(data=data.c2
        ,standardize = TRUE,trace = F)

###############################################
###Pvalues from likelihood ratio test using
###univariate Cox Ph model for cause 1
uni.pval.c1<-c()
for (l in 1:k){
        data.c1<-list(xx=x[,l], time=days
        , status=censor.ind.1)
        uni.pval.c1[l]<-summary(coxph(
        Surv(time, status) ~ xx, data.c1))
        $logtest["pvalue"]
}
names(uni.pval.c1)<-colnames(x)

###Pvalues from likelihood ratio test using
### univariate Cox Ph model for cause 2
uni.pval.c2<-c()
for (l in 1:k){
        data.c2<-list(xx=x[,l], time=days
        , status=censor.ind.2)
        uni.pval.c2[l]<-summary(coxph( Surv
        (time, status) ~ xx, data.c2))$
        logtest["pvalue"]
}
```

```
        names(uni.pval.c2)<-colnames(x)

        ###Report
        result<-list(coxpath.c1,coxpath.c2,
                uni.pval.c1, uni.pval.c2)
        return(result)
}

#######################################################
#Calculating the sensitivity and specificity for
#Univariate Cox model approach
#Jun 2nd, 2008
#######################################################
s.s.func<-function(alpha=alpha, p.val.mat,nzero=nzero)
        {
        k<-ncol(p.val.mat)
        run<-nrow(p.val.mat)

        check.mat<-matrix(rep(NA, run*k), ncol=k)
        for (i in 1:run){
                check.mat[i,]<-ifelse(p.val.mat[i,]
                <=alpha, 1, 0)
                }

        sens<-sum(as.vector(check.mat[,c(1:(nzero/2)
                ,(k-nzero/2+1):k)]))/(nzero*run)

        spec<-1-sum(as.vector(check.mat[,c((nzero/2+1)
                :(k-nzero/2))]))/((k-nzero)*run)
        result<-c(sens, spec)
        names(result)<-c("sens","spec")
        return(result)

}



####################################################
#########################################################
#SITUATION10: N=200,K=1000,P.RISK1=0.6
#########################################################
n<-200                  #total number of observations
k<-1000                 #number of covariates
p.risk1<-0.6            #proportion of observations
                        #fail due to risk 1
```

```
nzero.effect1<-2        #the non-zero coefficient
                        #effect of beta
nzero.effect2<-2
nzero<-20
run<-20

###Non-zero Coefficients of for cause 1 and cause 2

###Non-zero Coefficients of for cause 1 and cause 2
beta1<-c(rep(nzero.effect1,nzero),rep(0,k-nzero))
names(beta1)<-paste("x",1:k,sep="")

beta2<--beta1
names(beta2)<-paste("x",1:k,sep="")

###
###
result.c1.r10<-list()
result.c2.r10<-list()

coef.c1.r10<-matrix(rep(NA,run*k),ncol=k)
coef.c2.r10<-matrix(rep(NA,run*k),ncol=k)

check.c1.r10<-matrix(rep(NA,run*k),ncol=k)
check.c2.r10<-matrix(rep(NA,run*k),ncol=k)

sens.c1.r10<-c()
spec.c1.r10<-c()

sens.c2.r10<-c()
spec.c2.r10<-c()

threshold<-c(n^(0.3),n^(0.1))

pval.c1.r10<-matrix(rep(NA,run*k),ncol=k)
pval.c2.r10<-matrix(rep(NA,run*k),ncol=k)

for (i in 1:run){
        temp<-sim.func(n=n, k=k,p.risk1=p.risk1
                        ,p.censor=p.censor
                        , beta1=beta1, beta2=beta2)
        result.c1.r10[[i]]<-temp[[1]]
        result.c2.r10[[i]]<-temp[[2]]
        pval.c1.r10[i,]<-temp[[3]]
        pval.c2.r10[i,]<-temp[[4]]
```

```
}

for(j in 1:length(threshold)){
        for(i in 1:run){

                #length(result.c1.r10[[i]]$lambda)-1
                ind<-length(result.c1.r10[[i]]$
                lambda[result.c1.r10[[i]]$lambda
                >threshold[j]])
                coef.c1.r10[i,]<-result.c1.r10[[i]]
                $b.corrector[ind,]

                #length(result.c2.r10[[i]]$lambda)-1
                ind<-length(result.c2.r10[[i]]$
                lambda[result.c2.r10[[i]]$lambda
                >threshold[j]])
                coef.c2.r10[i,]<-result.c2.r10[[i]]
                $b.corrector[ind,]

                check.c1.r10[i,]<-ifelse(coef.c1.r10
                [i,]!=0,1,0)
                check.c2.r10[i,]<-ifelse(coef.c2.r10
                [i,]!=0,1,0)

        }
        sens.c1.r10[j]<-sum(as.vector(check.c1.r10
                [,c(1:nzero)]))/(nzero*run)

        spec.c1.r10[j]<-1-sum(as.vector(check.c1.r10
                [,c((nzero+1):k)]))/((k-nzero)*run)


        sens.c2.r10[j]<-sum(as.vector(check.c2.r10
                [,c(1:nzero)]))/(nzero*run)

        spec.c2.r10[j]<-1-sum(as.vector(check.c2.r10
                [,c((nzero+1):k)]))/((k-nzero)*run)

}

###Univariate approach result
uni.c1.r10<-matrix(rep(NA, length(alpha)*2), ncol=2)
colnames(uni.c1.r10)<-c("sens","spec")
rownames(uni.c1.r10)<-c("alpha.01", "alph.05")
uni.c1.r10[1,]<-s.s.func(alpha=alpha[1], p.val.mat
                =pval.c1.r10, nzero=nzero)
uni.c1.r10[2,]<-s.s.func(alpha=alpha[2], p.val.mat
                =pval.c1.r10, nzero=nzero)

save.image("SITUATION␣9␣to␣10-composition␣and
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣univariate␣x.RData")
```

C.4 Plotting for Table 4.1

```
########################################################
#########################################################
#BY:      XIANGRONG KONG
#LAST MODIFIED DATE:    JULY 2ND, 2008
########################################################
########################################################
setwd("C:/talaci/research/thesis/data/simulatioN
␣␣␣␣␣␣␣␣/simulation␣results/July22")

memory.limit(size=4000)

#######################################Failure Type

sim.result<-read.csv(file="New␣simulation␣result␣for
␣␣␣␣␣␣␣␣ploting␣xa␣indepen␣xb-␣CAUSE1.csv"
        ,colClasses = c(rep("numeric",6),"character"))

sim.result[,3:6]<-sim.result[,3:6]*100

attach(sim.result)

par(mfrow=c(2,2))
par(ps=18)

###k vs. sensitivity, Failure Type I

plot(k[n==100],sensitivity.lambda1[n==100], type="n",
        xlim=c(0,1000), ylim=c(10,100), xlab="k"
        ,ylab="Sensitivity%",font.lab=2,ps=100,
        main="Failure␣Type␣I",cex.main=0.9,cex.axis=0.9)

points(k[n==100],sensitivity.lambda1[n==100],col =
        "red", pch=19)
lines(k[n==100],sensitivity.lambda1[n==100],col =
        "red", pch=19,lwd=3)

points(k[n==200],sensitivity.lambda1[n==200],col =
        "red", pch=11)
lines(k[n==200],sensitivity.lambda1[n==200],col =
        "red", pch=11,lwd=3)
```

```
points(k[n==100],sensitivity.lambda2[n==100],col =
        "green",pch=19)
lines(k[n==100],sensitivity.lambda2[n==100],col =
        "green",pch=19,lty=2)

points(k[n==200],sensitivity.lambda2[n==200],col =
        "green",pch=11)
lines(k[n==200],sensitivity.lambda2[n==200],col =
        "green",pch=11,lty=2)

axis(2,at=50,labels=T, cex.axis=0.6)

par(ps=15)
legend(x=425,y=40, legend=c(expression(paste("n=100,
␣␣␣␣␣␣␣␣",lambda^1,";")),expression(paste("n=200,␣",
        lambda^1,";"))),pch=c(19,11),lty=c(1),
         text.col=c("red","red")
        ,col=c("red","red"), bty="n")


legend(x=735,y=39, legend=c(expression(paste("n=100,␣"
        ,lambda^2)),expression(paste("n=200,␣",
        lambda^2))),pch=c(19,11),lty=c(2), text.col=
        c("green","green")
        ,col=c("green","green"),bty="n")

par(lwd=0.2)
abline(h=50,lty=3)


###k vs. specificity, Failure type I
par(ps=18)

plot(k[n==100],specificity.lambda1[n==100], type="n",
        xlim=c(0,1000), ylim=c(10,100), xlab="k"
        ,ylab="specificity%",font.lab=2,ps=100,
        main="Failure␣Type␣I",cex.main=0.9,cex.axis=0.9)

points(k[n==100],specificity.lambda1[n==100],col =
         "red", pch=19)
lines(k[n==100],specificity.lambda1[n==100],col =
        "red", pch=19, lwd=3)

points(k[n==200],specificity.lambda1[n==200],col =
        "red", pch=11)
```

```
lines(k[n==200],specificity.lambda1[n==200],col =
        "red", pch=11,lwd=3)

points(k[n==100],specificity.lambda2[n==100],col =
        "green",pch=19)
lines(k[n==100],specificity.lambda2[n==100],col =
        "green",pch=19,lty=2)

points(k[n==200],specificity.lambda2[n==200],col =
        "green",pch=11)
lines(k[n==200],specificity.lambda2[n==200],col =
        "green",pch=11,lty=2)
axis(2,at=50,labels=T, cex.axis=0.6)

par(ps=15)
legend(x=425,y=40, legend=c(expression(paste("n=100,␣"
        ,lambda^1,";")),expression(paste("n=200,␣",
         lambda^1,";"))),pch=c(19,11),lty=c(1),
         text.col=c("red","red")
        ,col=c("red","red"), bty="n")


legend(x=735,y=39, legend=c(expression(paste("n=100,
␣␣␣␣␣␣␣␣",lambda^2)),expression(paste("n=200,␣",
        lambda^2))),pch=c(19,11),lty=c(2),
        text.col=c("green","green")
        ,col=c("green","green"),bty="n")

par(lwd=0.2)
abline(h=50,lty=3)

########################################Failure Type II

sim.result.2<-read.csv(file="New␣simulation␣result␣for
␣␣␣␣␣␣␣␣ploting␣xa␣indepen␣xb-␣CAUSE2.csv"
               ,colClasses = c(rep("numeric",6),
        "character"))


sim.result.2[,3:6]<-sim.result.2[,3:6]*100


###k vs. sensitivity, Failure Type II
par(ps=18)
```

```
plot(k[n==100],sim.result.2$sensitivity.lambda1[n==100]
        , type="n", xlim=c(0,1000), ylim=c(10,100),
        xlab="k"
        ,ylab="Sensitivity%",font.lab=2,ps=100,
        main="Failure␣Type␣II",cex.main=0.9,cex.axis=0.9)


points(k[n==100],sim.result.2$sensitivity.lambda1[n==100]
        ,col = "red", pch=19)
lines(k[n==100],sim.result.2$sensitivity.lambda1[n==100]
        ,col = "red", pch=19,lwd=3)

points(k[n==200],sim.result.2$sensitivity.lambda1[n==200]
        ,col = "red", pch=11)
lines(k[n==200],sim.result.2$sensitivity.lambda1[n==200]
        ,col = "red", pch=11,lwd=3)

points(k[n==100],sim.result.2$sensitivity.lambda2[n==100]
        ,col = "green",pch=19)
lines(k[n==100],sim.result.2$sensitivity.lambda2[n==100]
        ,col = "green",pch=19,lty=2)

points(k[n==200],sim.result.2$sensitivity.lambda2[n==200]
        ,col = "green",pch=11)
lines(k[n==200],sim.result.2$sensitivity.lambda2[n==200]
        ,col = "green",pch=11,lty=2)

axis(2,at=50,labels=T, cex.axis=0.6)

par(ps=15)
legend(x=425,y=40, legend=c(expression(paste("n=100,␣"
        ,lambda^1,";")),expression(paste("n=200,␣",
        lambda^1,";"))),pch=c(19,11),lty=c(1)
        , text.col=c("red","red")
        ,col=c("red","red"), bty="n")


legend(x=735,y=39, legend=c(expression(paste("n=100,␣"
        ,lambda^2)),expression(paste("n=200,␣",
        lambda^2))),pch=c(19,11),lty=c(2),
         text.col=c("green","green")
        ,col=c("green","green"),bty="n")

par(lwd=0.2)
abline(h=50,lty=3)
```

```
###k vs. specificity, Failure type II
par(ps=18)

plot(k[n==100],sim.result.2$specificity.lambda1[n==100]
        , type="n", xlim=c(0,1000), ylim=c(10,100),
         xlab="k"
        ,ylab="specificity%",font.lab=2,ps=100,
        main="Failure␣Type␣II",cex.main=0.9,cex.axis=0.9)

points(k[n==100],sim.result.2$specificity.lambda1
        [n==100],col = "red", pch=19)
lines(k[n==100],sim.result.2$specificity.lambda1[n==100]
        ,col = "red", pch=19, lwd=3)

points(k[n==200],sim.result.2$specificity.lambda1[n==200]
        ,col = "red", pch=11)
lines(k[n==200],sim.result.2$specificity.lambda1[n==200]
        ,col = "red", pch=11,lwd=3)

points(k[n==100],sim.result.2$specificity.lambda2[n==100]
        ,col = "green",pch=19)
lines(k[n==100],sim.result.2$specificity.lambda2[n==100]
        ,col = "green",pch=19,lty=2)

points(k[n==200],sim.result.2$specificity.lambda2[n==200]
        ,col = "green",pch=11)
lines(k[n==200],sim.result.2$specificity.lambda2[n==200]
        ,col = "green",pch=11,lty=2)

axis(2,at=50,labels=T, cex.axis=0.6)

par(ps=15)
legend(x=425,y=40, legend=c(expression(paste("n=100,␣"
        ,lambda^1,";")),expression(paste("n=200,␣",
        lambda^1,";"))),pch=c(19,11),lty=c(1),
         text.col=c("red","red")
        ,col=c("red","red"), bty="n")


legend(x=735,y=39, legend=c(expression(paste("n=100,
␣␣␣␣␣␣␣␣",lambda^2)),expression(paste("n=200,␣", lambda^2)))
        ,pch=c(19,11),lty=c(2), text.col=c("green","green")
        ,col=c("green","green"),bty="n")
```

```
par(lwd=0.2)
abline(h=50,lty=3)
```

C.5 Plotting for Table 4.2

```
#########################################################
#PURPOSE: PLOTTING THE SIMULATION RESULTS FIGURE 4.2
#BY:      XIANGRONG KONG
#LAST MODIFIED DATE:    JULY 2ND, 2008
#######################################################
setwd("C:/talaci/research/thesis/data/simulatioN
␣␣␣␣␣␣␣␣/simulation␣results\July22")

memory.limit(size=4000)

#######################################Failure Type I

sim.result<-read.csv(file="New␣simulation␣result␣for
␣␣␣␣␣␣␣␣ploting␣comparing␣ind␣and␣corr.csv"
        ,colClasses = "numeric")

attach(sim.result)

par(mfrow=c(2,1))
par(ps=18)

###Sensitivity vs. Specificity, Failure Type I

plot(sensitivity.lambda1[Type==1],specificity.lambda1
        [Type==1], type="n", xlim=c(20,100),
        ylim=c(85,90), xlab="Sensitivity%"
        ,ylab="Specificity%",font.lab=2, main=
        "Failure␣Type␣I",cex.axis=0.9,yaxt="n"
        ,cex.main=0.9)

points(sensitivity.lambda1[Type==1&Max.Correlation==0]
        ,specificity.lambda1[Type==1&Max.Correlation==0],
        col = "red", pch=19)

points(sensitivity.lambda1[Type==1&Max.Correlation==0.8]
        ,specificity.lambda1[Type==1&Max.Correlation==0.8]
        , ,col = "red", pch=11)

lines(sensitivity.lambda1[Type==1],specificity.lambda1
        [Type==1], col = "red", pch=19,lwd=3)
```

```
points(sensitivity.lambda2[Type==1&Max.Correlation==0.0]
        ,specificity.lambda2[Type==1&Max.Correlation==0.0
        ], ,col = "green", pch=19)

points(sensitivity.lambda2[Type==1&Max.Correlation==0.8]
        ,specificity.lambda2[Type==1&Max.Correlation==0.
        8], ,col = "green", pch=11)

lines(sensitivity.lambda2[Type==1],specificity.lambda2
        [Type==1], col = "green", pch=19, lty=2)

par(ps=15)

legend(locator(1), legend=expression(lambda^1),pch=c(),
        lty=c(), text.col=c("red")
        ,col=c("red"), bty="n")

legend(locator(1), legend=expression(lambda^2),pch=c(),
        lty=c(), text.col=c("green")
        ,col=c("green"), bty="n")

legend(x= 81.96931-6,y= 86.57141, legend=c("no␣correla
␣␣␣␣␣␣␣␣tion","Max␣Correlation␣0.8"),pch=c(19,11),bty="n"
        )

axis(side=2,at=c(85,86,87,88,89,90),label=T)

#######################################Failure Type II
par(ps=18)

plot(sensitivity.lambda1[Type==2],specificity.lambda1
        [Type==2], type="n", xlim=c(20,100), ylim=c
        (85,90), xlab="Sensitivity%"
        ,ylab="Specificity%",font.lab=2, main="Failure
␣␣␣␣␣␣␣␣␣Type␣II",cex.axis=0.9,yaxt="n",cex.main=0.9)

points(sensitivity.lambda1[Type==2&Max.Correlation==0]
        ,specificity.lambda1[Type==2&Max.Correlation==0]
        , ,col = "red", pch=19)

points(sensitivity.lambda1[Type==2&Max.Correlation==0.8]
        ,specificity.lambda1[Type==2&Max.Correlation==
        0.8], ,col = "red", pch=11)
```

```
lines(sensitivity.lambda1[Type==2],specificity.lambda1
        [Type==2], col = "red", pch=19,lwd=3)


points(sensitivity.lambda2[Type==2&Max.Correlation==0.
        0],specificity.lambda2[Type==2&Max.Correlation
        ==0.0], ,col = "green", pch=19)

points(sensitivity.lambda2[Type==2&Max.Correlation==
        0.8],specificity.lambda2[Type==2&Max.Correlation
        ==0.8], ,col = "green", pch=11)

lines(sensitivity.lambda2[Type==2],specificity.lambda2
        [Type==2], col = "green", pch=19, lty=2)

par(ps=15)
legend(locator(1), legend=expression(lambda^1),pch=c()
        ,lty=c(), text.col=c("red")
        ,col=c("red"), bty="n")
legend(locator(1), legend=expression(lambda^2),pch=c()
        ,lty=c(), text.col=c("green")
        ,col=c("green"), bty="n")


legend(x= 81.96931-6,y= 86.57141, legend=c("no␣correlation
␣␣␣␣␣␣␣␣","Max␣Correlation␣0.8"),pch=c(19,11),bty="n")

axis(side=2,at=c(85,86,87,88,89,90),label=T)
```

Appendix D
Source Code for the Two Applications of the Proposed Method in Chapter 5

D.1 Code for The Prostate Cancer Example

```
#########################################################
#########################################################
#PURPOSE: EXAMPLE IN DISSERTAION: L1 PENALIZED MAXIMUM
#LIKELIHOOD APPROACH IN COMPETING RISKS Prostatic Cancer
# DATA APPLICATION
#LAST MODIFIED DATE:    May. 07, 2008
#########################################################
#########################################################

setwd("C:/talaci/research/thesis/data/Prostate␣cancer")

library(survival)
library(glmpath)

memory.limit(size=4000)

########################################################
#REVISION OF BOOTSTRAP.PATH FUNCTION
#PURPOSE: the original bootstrap function in (glmpath)
#can only work for criteria "aic" and "bic";
#         to use lambda.1=n^0.3, the function is revised
########################################################

bootstrap.path.1<-function (x, y, data, B, index = NULL,
                path = c("glmpath", "coxpath"),
    method = c("aic", "bic"), trace = FALSE, ...)
                #When choosing
                #"aic", it actually means #using
                # lambda.1=n^0.3
{
    path <- match.arg(path)
    method <- match.arg(method)
    if (!missing(data))
        x <- data$x
    n <- nrow(x)
    p <- ncol(x)
    if (!is.null(index))
        B <- nrow(index)
    else index <- matrix(sample(c(1:n), n * B
                , replace = T),
        nrow = B)
    beta <- matrix(0, B, p)
    lambda.1<-n^0.3

    if (path == "glmpath") {
        if (!missing(data))
            y <- data$y
        fit <- glmpath(x, y, ...)
        s <- switch(method, aic = which.min(fit$aic)
                , bic = which.min(fit$bic))
        beta0 <- fit$b.corrector[s, -1] * fit$sdx
        for (b in 1:B) {
            bx <- x[index[b, ], ]
            by <- y[index[b, ]]
            fit <- glmpath(bx, by, ...)
            s <- switch(method, aic = which.min(fit$aic)
                , bic = which.min(fit$bic))
            beta[b, ] <- fit$b.corrector[s, -1] * fit$
                sdx
            if (trace)
                cat(b)
        }
    }
    else {
        time <- data$time
        status <- data$status
        fit <- coxpath(data, trace = FALSE, ...)
        s <- switch(method, aic = which.min(fit$lambda
                [fit$lambda>=lambda.1]), bic = which.min
                (fit$bic))
        beta0 <- fit$b.corrector[s, ] * fit$sdx
        for (b in 1:B) {
            bx <- x[index[b, ], ]
            btime <- time[index[b, ]]
            bstatus <- status[index[b, ]]
            fit <- coxpath(list(x = bx, time = btime,
                 status = bstatus),
                ...)
            s <- switch(method, aic = which.min(fit$
                lambda[fit$lambda>=lambda.1]), bic =
                which.min(fit$bic))
            beta[b, ] <- fit$b.corrector[s, ] * fit$sdx
            if (trace)
                cat(b)
        }
    }
    dimnames(beta) <- list(seq(B), dimnames(x)[[2]])
    attr(beta, "coefficients") <- beta0
    class(beta) <- "bootpath"
    beta
}



########################################################
raw.data<-read.csv(file="C:/talaci/research/thesis/data
␣␣␣␣␣␣␣␣/Prostate␣cancer/Prostate␣data_XK.csv"
        ,colClasses = "numeric")

status.char<-ifelse(raw.data$Survival.Status>=3,
        "other",raw.data$Survival.Status)

status.char<-ifelse(status.char=="0", "alive",
        status.char)

status.char<-ifelse(status.char=="1" , "pros.cancer"
        ,status.char)

status.char<-ifelse(status.char=="2", "heart",
         status.char)

survival.time<-raw.data$Follow.Up.months

x.data<-raw.data[,c(3,6:11,13:17)]

###Status indicator for "Prostatic Cancer"
cancer.status<-ifelse(status.char=="pros.cancer",1,0)
```

```
#######################################################

###Cox Path- Cancer group
cancer.data.list<-list(x=as.matrix(x.data)

cancer.cox<-coxpath(data=cancer.data.list
                , standardize = TRUE,trace = TRUE)
        #Note, the b.corrector saves the estimates



lambda.1<-483^0.3

min.aic<-which(min(cancer.cox$aic)==cancer.cox$aic
                ,cancer.cox$aic)
lambda.min<-cancer.cox$lambda[min.aic]
        #This is the lambda corresponding to the
        # least AIC

s.lambda.1<-which.min(cancer.cox$lambda[cancer.cox
                $lambda>=lambda.1])


#Standardized coefficients estimates from the original
        #data using lambda.1
cancer.est<-cancer.cox$b.corrector[c(s.lambda.1),]
                *cancer.cox$sdx


plot.coxpath(x=cancer.cox, xvar = "lambda",type =
                 c("coefficients", "aic", "bic")
                , xlimit = NULL,
               predictor = FALSE, omit.zero = TRUE
                , breaks =FALSE,mar = c(5, 4, 4, 8.5)
                , main="Coefficients␣Path␣-␣Prostate
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Cancer")
        #plot.coxpath plots the standardized
        # coefficients!
abline(v=lambda.1)
abline(v=lambda.min)
abline(v=cancer.cox$lambda, lty=3)
axis(side=1, at=c(lambda.1, lambda.min),
                labels=expression(lambda^1,lambda^2))
```

```
###Use Bootstrap to get the estimates of standard
###errors
#bootstrap.path.1
#The exisiting bootstrap.path function can only
#work for "method=aic or bic". So to use n^{0.3}
#, the function is revised to accomadate the
#choic of lambda=n^{0.3}. Refer to the top of
#this file for the new function!
#Note: bootstrap.path returns statndardized coefficients

cancer.boot.1<-bootstrap.path.1(data=cancer.data.list
        , B=100, path="coxpath", method=c("aic"),
        trace=FALSE)

cancer.boot.2<-bootstrap.path(data=cancer.data.list
        , B=100, path="coxpath", method=c("aic")
        , trace=FALSE)

###Plot the boxplot of the bootstrap estimates
par(mar=c(5.1, 4.1, 4.1, 2.5))
colnames(cancer.boot.1)<-c("Trt","Age","Wgt","PR",
        "CH","SBP","DBP","HG","TS","CI","AP","BM")
class(cancer.boot.1)<-"matrix"
boxplot(as.data.frame(cancer.boot.1), main="Prostate
␣␣␣␣␣␣␣␣␣Cancer" , ylab="Standardized␣coefficients
␣␣␣␣␣␣␣␣␣estimates")
abline(h=0, lty=3)

points(x=1:12, y=cancer.est, pch=19,col="red")


###Calculate p-values
cancer.std.err<-sqrt(apply(cancer.boot.1,2, var))
                #std error from bootstrap
cancer.p<-round(pnorm(abs(cancer.est/cancer.std.err),
        lower.tail=FALSE),3)
round(cancer.p,3)

###Save the standardized estimates
#write.csv(x= t(scale(cancer.cox$b.corrector[c(12:13,17)
        ,],center=F, 1/cancer.cox$sdx))
        , file="standardized␣estimates.csv")

###Ordinary Cox PH model to get the MLE and the std
```

```
        errors for the estimates
x.data.ph<-scale(x.data,center=T, scale=T)
cancer.data.list.ph<-list(x=as.matrix(x.data.ph)

coxph(Surv(time,status)~x, cancer.data.list.ph)

###Cox-Snell residuals
cox.csresid<-(cancer.status-coxph(Surv(time,status)
        ~x, cancer.data.list.ph)$residuals)

plot(log(-log(summary(survfit(Surv(cox.csresid
        , cancer.status)~1))$surv))
                ,log(summary(survfit(Surv(cox.csresid
        , cancer.status)~1))$time), xlim=c(-6, 0)
        , ylim=c(-6,0)
                , main="Prostate␣cancer", xlab=
        "log(C-S␣residual)"
                , ylab="log(-log(Kaplan-Meier
␣␣␣␣␣␣␣␣estimate␣of␣the␣C-S␣residual))")

abline(0,1)


########################################################
#######################################################

###Cox Path- Heart group
heart.status<-ifelse(status.char=="heart",1,0)

heart.data.list<-list(x=as.matrix(x.data)\
        , time=survival.time,status=heart.status)

heart.cox<-coxpath(data=heart.data.list
        , standardize = TRUE,trace = TRUE)

min.aic<-which(min(heart.cox$aic)==heart.cox$aic
        ,heart.cox$aic)
lambda.min<-heart.cox$lambda[min.aic]
#This is the lambda corresponding to the least AIC

s.lambda.1<-which.min(heart.cox$lambda[heart.cox
        $lambda>=lambda.1])
s.lambda.2<-which.min(heart.cox$aic)
s.lambda.0<-length(heart.cox$lambda)
```

```
#Standardized coefficients estimates on original
# data using lambda.1
heart.est<-heart.cox$b.corrector[c(s.lambda.1),]
        *heart.cox$sdx


#Plot of paths
x11()
plot.coxpath(x=heart.cox, xvar = "lambda",type =
        c("coefficients", "aic", "bic"), xlimit = NULL,
               predictor = FALSE, omit.zero =
        TRUE, breaks =FALSE,mar = c(5, 4, 4, 8.5)
                , main="Coefficients␣Path␣-
␣␣␣␣␣␣␣␣Cardivascular␣Disease")

abline(v=lambda.1)
abline(v=lambda.min)
abline(v=heart.cox$lambda, lty=3)
axis(side=1, at=c(lambda.1, lambda.min),
        labels=expression(lambda^1,lambda^2))


###Use Bootstrap to get the estimates of standard errors
#bootstrap.path.1
#The exisiting bootstrap.path function can only work for
#"method=aic or bic". So to use n^{0.3}, the function is
#revised to accomadate the choic of lambda=n^{0.3}. Refer
#to the top of this file for the new function!
#Note: bootstrap.path returns statndardized coefficients

heart.boot.1<-bootstrap.path.1(data=heart.data.list
        , B=100, path="coxpath", method=c("aic"),
        trace=FALSE)

heart.boot.2<-bootstrap.path(data=heart.data.list,
        B=100, path="coxpath", method=c("aic"),
        trace=FALSE)

###Plot the boxplot of the bootstrap estimates using
###lambda.1
x11()
par(mar=c(5.1, 4.1, 4.1, 2.5))
colnames(heart.boot.1)<-c("Trt","Age","Wgt","PR","CH"
        ,"SBP","DBP","HG","TS","CI","AP","BM")
class(heart.boot.1)<-"matrix"
```

```
boxplot(as.data.frame(heart.boot.1), main=
        "Cardiovascular␣disease" , ylab="Standardized
␣␣␣␣␣␣␣␣coefficients␣estimates")
abline(h=0, lty=3)

points(x=1:12, y=heart.est, pch=19,col="red")

###Calculate p-value

heart.std.err<-sqrt(apply(heart.boot.1,2, var))
        #std error from bootstrap
heart.p<-round(pnorm(abs(heart.est/heart.std.err),
         lower.tail=FALSE),3)
round(heart.p,3)


###Save the standardized estimates
#write.csv(x= t(scale(heart.cox$b.corrector[c(s
        .lambda.1,s.lambda.2,s.lambda.0),],center=F,
         1/heart.cox$sdx))
# ,append=TRUE, file="standardized estimates.csv")

###Ordinary Cox PH model to get the MLE and the
### std errors for the estimates
x.data.ph<-scale(x.data,center=T, scale=T)
heart.data.list.ph<-list(x=as.matrix(x.data.ph),
         time=survival.time,status=heart.status)

coxph(Surv(time,status)˜x, heart.data.list.ph)

###Cox-Snell residuals
cox.csresid<-(heart.status-coxph(Surv(time,status)˜x,
        heart.data.list.ph)$residuals)

plot(log(-log(summary(survfit(Surv(cox.csresid,
         heart.status)˜1))$surv))
                ,log(summary(survfit(Surv(cox.csresid,
         heart.status)˜1))$time), xlim=c(-6, 0),
         ylim=c(-6,0)
                , main="Cardiovascular␣disease",
        xlab="log(C-S␣residual)"
                , ylab="log(-log(Kaplan-Meier
␣␣␣␣␣␣␣␣estimate␣of␣the␣C-S␣residual))")

abline(0,1)
```

D.2 Code for The HCV+HCC Example

```
#######################################################
#######################################################
#PURPOSE: EXAMPLE IN DISSERTAION: L1 PENALIZED MAXIMUM
#LIKELIHOOD APPROACH IN COMPETING RISKS
# HCV-HCC PATIENTS PROGRESSION DATA APPLICATION
#BY:      XIANGRONG KONG
#LAST MODIFIED DATE:    Jul6. 15, 2008
######################################################
######################################################

setwd("C:/talaci/research/thesis/data/HCC")

library(annotate)
library(hgu133a)
library(hgu133a2)
library(affy)
library(matchprobes)
library(hgu133aprobe)
library(hgu133a2probe)

library(survival)
library(glmpath)

memory.limit(size=4000)
###################################################
######################################################

#REVISION OF BOOTSTRAP.PATH FUNCTION
#PURPOSE: the original bootstrap function in (glmpath)
# can only
#work for criteria "aic" and "bic"; to use lambda.1
#=n^0.1, the function is revised
#######################################################
bootstrap.path.1<-function (x, y, data, B, index =
                 NULL, path = c("glmpath", "coxpath"),
    method = c("aic", "bic"), trace = FALSE, ...)
                #When choosing "aic", it actually means

                #using lambda.1=n^0.1
{
    path <- match.arg(path)
```

```
method <- match.arg(method)
if (!missing(data))
    x <- data$x
n <- nrow(x)
p <- ncol(x)
if (!is.null(index))
    B <- nrow(index)
else index <- matrix(sample(c(1:n), n * B, replace
            = T),   nrow = B)
beta <- matrix(0, B, p)
lambda.1<-n^0.1

if (path == "glmpath") {
    if (!missing(data))
        y <- data$y
    fit <- glmpath(x, y, ...)
    s <- switch(method, aic = which.min(fit$aic)
            , bic = which.min(fit$bic))
    beta0 <- fit$b.corrector[s, -1] * fit$sdx
    for (b in 1:B) {
        bx <- x[index[b, ], ]
        by <- y[index[b, ]]
        fit <- glmpath(bx, by, ...)
        s <- switch(method, aic = which.min(fit$aic
                    ), bic = which.min(fit$bic))
        beta[b, ] <- fit$b.corrector[s, -1] * fit
            $sdx
        if (trace)
            cat(b)
    }
}
else {
    time <- data$time
    status <- data$status
    fit <- coxpath(data, trace = FALSE, ...)
    s <- switch(method, aic = which.min(fit$lambda
    [fit$lambda
                    >=lambda.1]), bic = which.min
    (fit$bic))
    beta0 <- fit$b.corrector[s, ] * fit$sdx
    for (b in 1:B) {
        bx <- x[index[b, ], ]
        btime <- time[index[b, ]]
        bstatus <- status[index[b, ]]
        fit <- coxpath(list(x = bx, time = btime
                                , status = bstatus),
                ...)
            s <- switch(method, aic = which.min(fit$
        lambda
                [fit$lambda>=lambda.1]), bic = which.
        min(fit$bic))
            beta[b, ] <- fit$b.corrector[s, ] * fit$
        sdx
            if (trace)
                cat(b)
        }
    }
    dimnames(beta) <- list(seq(B), dimnames(x)[[2]])
    attr(beta, "coefficients") <- beta0
    class(beta) <- "bootpath"
    beta
}

######################################################
raw.pheno<-read.csv(file="C:/talaci/research/thesis
␣␣␣␣␣␣␣␣/data
/HCC␣data␣2007_Xk.csv",colClasses = "character")

cbind(1:dim(raw.pheno)[2],colnames(raw.pheno))

raw.pheno.2<-raw.pheno[raw.pheno$Exclude=="",]
raw.pheno.2<-raw.pheno.2[raw.pheno.2$celfile!="D-451B"
        ,]
        #This is the replicate chip for patient
        #"4243570"
        #it's removed after consulting with Dr. Mas

raw.pheno.3<-raw.pheno.2[,c(1,3,4,6,7,8,11,12,15,16
        ,17,18,19)]

pheno<-raw.pheno.3
#write.csv(raw.pheno.3, "Reduced Pheno data_XK.csv")
rm(list=c("raw.pheno","raw.pheno.2","raw.pheno.3"))

diag.date<-as.Date(pheno$Diagnostic.date, "%m/%d/%Y")
names(diag.date)<-pheno$celfile

tran.date<-as.Date(pheno$TransDate,"%m/%d/%Y")
tran.leng.temp<-as.numeric(difftime(tran.date,diag.
        date ,units="days"))
```

```
names(tran.leng.temp)<-pheno$celfile

prog.date<-as.Date(pheno$DateProgress,"%m/%d/%Y")
prog.leng.temp<-as.numeric(difftime(prog.date, diag.
        date, units="days"))
names(prog.leng.temp)<-pheno$celfile

surv.date<-as.Date(pheno$SurvDate, "%m/%d/%Y")
surv.leng.temp<-as.numeric(difftime(surv.date,
        diag.date, units="days"))
        #died and alive are both treated as censored,
names(surv.leng.temp)<-pheno$celfile
                # due to lack of sample

cbind(tran.leng.temp, prog.leng.temp, surv.leng.temp)

temp<-ifelse(tran.leng.temp<prog.leng.temp,1,0)
tran.leng<-ifelse(is.na(temp) | temp==1, tran.leng.temp
        ,NA)

temp<-ifelse(prog.leng.temp<tran.leng.temp, 1,0)
prog.leng<-ifelse(is.na(temp) | temp==1, prog.leng.temp

        , NA)

cbind(tran.leng, tran.leng.temp)
cbind(prog.leng, prog.leng.temp)

cbind(tran.leng, prog.leng)
rm(list=c("tran.leng.temp", "prog.leng.temp", "temp"))

surv.leng<-ifelse(is.na(tran.leng) & is.na(prog.leng)
        , surv.leng.temp,NA)
cbind(tran.leng, prog.leng, surv.leng)
rm(list=c("surv.leng.temp"))

tran.censor.ind<-ifelse(is.na(tran.leng),0,1)
                #this is the censoring indicator
                #specifically for cause transplantation

prog.censor.ind<-ifelse(is.na(prog.leng),0,1)
                #this is the censoring indicator
                $specifically for cause progression

cbind(tran.leng, tran.censor.ind)
cbind(prog.leng, prog.censor.ind)

cmbd.length<-apply(cbind(tran.leng,prog.leng,surv.leng),1
        ,min,na.rm=TRUE)




#####################################################
#####################################################
#Microarry data processing: RMA
#####################################################
#####################################################
#cmbd.cdf<-cmbd.affy$cdf
#gn<-ls(cmbd.cdf)
### The following control genes are on the old but not
### the new GeneChips
### To avoid problems, just remove them from the
###datasets/cdf
#rm(list=grep("AFFX-r2-H",gn,value=TRUE)
        #,envir=cmbd.cdf)
#comb.rma<-rma(cmbd.affy$dat)
#comb.calls<-mas5calls(cmbd.affy$dat)
#save.image("HCCSurvival.RData")
#q()

#rma.data.temp<-exprs(comb.rma)
#rma.data<-rma.data.temp[-grep("AFFX",rownames
#(rma.data.temp)),]
#dim(rma.data)
#rm(cmbd.affy)
#rm(rma.data.temp)

#Compare the replicate chips of the same patient

#plot((rma.data[,grep("10-D-422",colnames(rma.data))]
               +rma.data
               [,grep("D-451B",colnames(rma.data))])
               /2
               ,log(rma.data[,grep("10-D-422",colnames
               (rma.data))]
               -rma.data[,grep("D-451B",colnames
               (rma.data))]
               ,base=2))
abline(h=0)    #Does not look like high
```

```
                # reproducibility


########################Progression
sum(ifelse(substr(dimnames(rma.data)[[2]],10,21)
        !=paste(names(cmbd.length),".CEL",sep=""),1,0))

###Penalized likelihood approach
rma.prog.data<-list(x=t(rma.data),time=cmbd.length
        , status=prog.censor.ind)
rma.prog.result<-coxpath(data=rma.prog.data
                , standardize    = TRUE,trace = TRUE)


lambda.1<-length(cmbd.length)^0.1

s.lambda.1<-which.min(rma.prog.result$lambda
                [rma.prog.result $lambda>=lambda.1])



#Standardized coefficients estimates on original data
#using lambda.1
coef.est<-rma.prog.result$b.corrector[c(s.lambda.1),]
        *rma.prog.result$sdx

#sig.coef<-coef.est[coef.est!=0]
#sig.names<-names(sig.coef)

#Plot of paths
par(cex.main=1, cex.lab=1)
plot.coxpath(x=rma.prog.result, xvar = "lambda",type
        = c("coefficients", "aic", "bic"), xlimit
                = NULL,
              predictor = FALSE, omit.zero = TRUE
        , breaks =FALSE,mar = c(5, 6, 4, 6.5)
        , main="Tumor␣progression")

abline(v=lambda.1,lwd=1)
#abline(v=rma.prog.result$lambda, lty=3)
axis(side=1, at=c(lambda.1), labels=expression(paste
        (lambda,"=",(n^0.1))))
```

```
###Use Bootstrap to quantify the estimates difference
### from 0
#bootstrap.path.1
#The exisiting bootstrap.path function can only work
#for"method=aic or bic". So to use n^{0.3},
#the function is revised to accomadate the choic of
# lambda=n^{0.1}. Refer to the top of this file for the
#new function!
#Note: bootstrap.path returns statndardized
# coefficients

prog.boot.1<-bootstrap.path.1(data=rma.prog.data,
                 B=100
                , path="coxpath", method=c("aic"),
                trace=FALSE)


#sig.boot<-prog.boot.1[, sig.names]

#std.err<-sqrt(apply(sig.boot, 2, var))

wilc.p<-c()
for(j in 1:dim(prog.boot.1)[2]){
wilc.p[j]<-wilcox.test(prog.boot.1[,j], exact=FALSE)
                $p.value
}

names(wilc.p)<-colnames(prog.boot.1)

all.count<-c()
for(j in 1:dim(prog.boot.1)[2]){
all.count[j]<-length(prog.boot.1[,j][prog.boot.1[,j]
                !=0])
}
names(all.count)<-colnames(prog.boot.1)

wilc.p[!is.na(wilc.p) & wilc.p<0.05]
        #These are the probesets with p-values<0.05
sig.names<-names(wilc.p[!is.na(wilc.p) & wilc.p<0.05])

coef.sig<-coef.est[sig.names]
        #The standardized coef estimates from the
        # orginal data
        #for the probesets with p-value<0.05
```

```
#write.csv(x=cbind(round(sig.coef,3), sig.count,
                round(wilc.p,3))
        , file="HCC␣result.csv")



###Report the significant genes. Individual alpha=0.05
sig.gene.p<-wilc.p[sig.names]

sig.gene.p<-sort(sig.gene.p)
sig.gene<-names(sig.gene.p)
sig.ll<-getLL(sig.gene, data="hgu133a2")
sig.sym<-getSYMBOL(sig.gene,"hgu133a2")
sig.uni<-mget(sig.gene,env=hgu133a2UNIGENE)
sig.chr<-mget(sig.gene,env=hgu133a2CHR)
sig.gn<-mget(sig.gene,env=hgu133a2GENENAME)
sig.map<-mget(sig.gene,env=hgu133a2MAP)
genelist<-list(sig.ll,sig.uni)

htmlpage(genelist=list(sig.ll),filename="Important
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣genes
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣for␣tumor␣progression.html"
        , title="Important␣genes␣for␣tumor␣progression"
                , table.head=c('LocusLink','UnigeneID'
                ,'AffyID','Gene␣Symbol','Gene␣Name'
                ,'Chromosome','Map',"Coef␣Estimate",
        "p-value"),
         othernames=list(sig.uni, sig.gene,sig.sym,
                 sig.gn , sig.chr,sig.map, round
                (coef.sig[sig.gene],3),
         round(wilc.p[sig.gene],3))
        ,table.center = TRUE
        , repository=list("ll"))

###Plot the boxplot of the bootstrap estimates
par(mar=c(6.5, 5.1, 4.1, 2.5))

boot.sig<-prog.boot.1[, sig.gene]
colnames(boot.sig)<-sig.sym
boxplot(as.data.frame(boot.sig),ylab=
        "Standardized
␣␣␣␣␣␣␣␣coefficients␣estimates",
         main="Tumor␣progression", las=3)
abline(h=0, lty=3)
```

```
coef.sig<-coef.est[sig.gene]
points(x=1:dim(boot.sig)[2], y=coef.sig,
         pch=19,col="red")

########################################################
########################################################

###Univaraite Cox PH model approach
log.p.value<-c()
for (j in 1:10){
        log.p.value[j]<-summary(coxph(Surv(cmbd.length
        ,prog.censor
                .ind)~t(rma.data)[,j]
                                , method=c("breslow"
        )))$logtest["pvalue"]
}

names(log.p.value)<-dimnames(rma.data)[[1]]
log.p.value[names(rma.prog.gene)]


########################
#Residual plot
#########################
x11()
sig.2.cox.residual<-coxph(Surv(cmbd.length,prog.censor
        .ind)~std.sig.rma[,2], method=c("breslow")
        )$residuals

cox.csresid<-(prog.censor.ind-sig.2.cox.residual)
plot(log(-log(summary(survfit(Surv(cox.csresid,
         prog.censor.ind)~1))$surv))
        ,log(summary(survfit(Surv(cox.csresid
        , prog.censor.ind)~1))$time), xlim=c(-4, 0)
        , ylim=c(-4,0)
        , main="Probe␣set␣202893_at"
        , xlab="log(C-S␣residual)"
        , ylab="log(-log(Kaplan-Meier
␣␣␣␣␣␣␣␣␣estimate␣of␣the␣C-S␣residual))")

abline(0,1)
```