ISyE 6404 CP.1: Proportional Hazards Regression

Yuan Gao, Kevin Lee, Akshay Govindaraj,
Yijun (Emma) Wan, Peter Williams, Ruixuan Zhang

Date: 2018-10-21

Contents

Workload Distribution

Below is a description of tasks and the distribution of work (%) by team member for this project:

Team Member	Task Description
Yuan Gao	TBD
Kevin Lee	TBD
Akshay Govindaraj	TBD
Yijun (Emma) Wan	TBD
Peter Williams	Code Compilation, R Debugging, Latex Formatting, Visualization
Ruixuan Zhang	TBD

Proportional Hazards Regression Tasks

1. Data Description & PH-regression

Task: Locate a data set in the field of your interest, e.g., eCommerce, medical study, drug development, supply-chain/logistics operations, to practice the PH-regression technique. Note that we need to predict both hazard-rate and the survival function at an input x0.

For this exercise, I located a dataset in the survival package, that describes the recurrence times to infection, at the point of insertion of the catheter, for kidney patients using portable dialysis equipment. Catheters may be removed for reasons other than infection, in which case the observation is censored. Each patient has exactly 2 observations. It has seven variables:

1) patient: id
2) time: time
3) status: event status
4) age: in years
5) sex: 1=male, 2=female
6) disease: disease type (0=GN, 1=AN, 2=PKD, 3=Other)
7) frail: frailty estimate from original paper

First, I compute the univariate Cox analyses for the four variables; then we'll fit multivariate cox analyses using two variables to describe how the factors jointly impact on survival.

Table 1: Model Results - Add a Better Title

covariate	beta	pvalue	rho
age	0.00	0.61	0.09
sex	-0.84	0.00	0.44
diseaseGN	0.35	1.46	-0.05
diseaseAN	0.38	0.99	0.09
diseasePKD	-0.26	0.61	-0.01
frail	1.01	0.00	0.08

The syntax above gives us the method to get the survival function of any given variables.The output above shows the regression beta coefficients, the effect sizes (given as hazard ratios) and statistical significance for each of the variables in relation to overall survival.

Following is the Cox regression results Analysis:

1) Statistical significance, z, gives the Wald statistic value,evaluates whether the beta coefficient of a given variable is statistically significantly different from 0. From the output above, we can conclude that the variable sex and frail have highly statistically significant coefficients, but age and disease don't. So, following we focus on the sex and frail.
2) The regression coefficients are positive signs meaning that the hazard (risk of death) is higher,

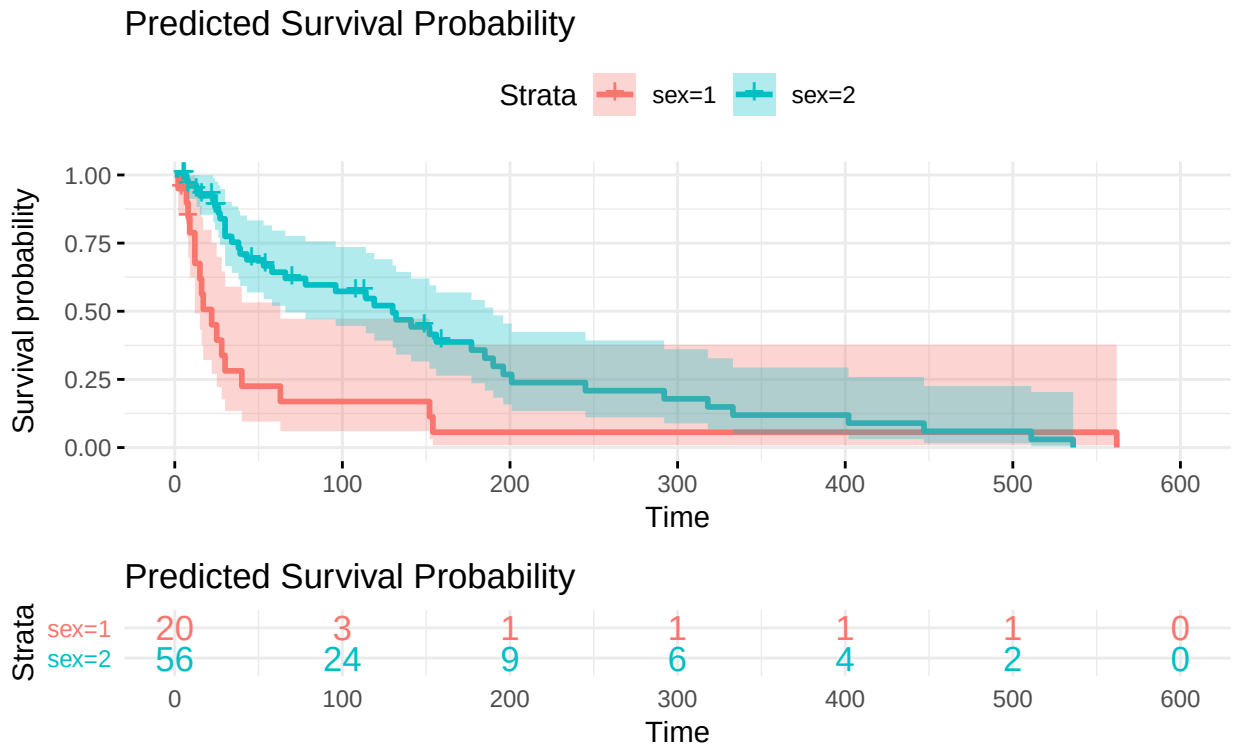
and thus the prognosis worse, for subjects with higher values of that variable.

3) Hazard ratios are the exponentiated coefficients (exp(coef)) which give the effect size of the variable age. So, it gives us the predict about the hazard ratio for any given variables. The variable sex is encoded as a numeric vector. 1: male, 2: female. The beta coefficient for sex = -0.838 indicates that females have lower risk of kidney (lower survival rates) than males, in these data.

4) Confidence intervals of the hazard ratios is shown by the upper and lower 95% confidence intervals for the hazard ratio (exp(coef)).

5) Global statistical significance p-values for three alternative tests for overall significance of the model: The likelihood-ratio test, Wald test, and score logrank statistics. These three methods are asymptotically equivalent.

6) cox.zph() function shows the data are sufficiently consistent with the assumption of proportional hazards with respect to each of the variables separately as well as globally.



The picture is to visualize the predicted survival proportion at any given point for a particular risk group.

Besides, the syntax above is to make predict about the survival interval of any given new data.

```
res.cox <- coxph(Surv(time, status) ~ sex + frail, data =  kidney)
new_dat <- data.frame(sex = c(1), frail = c(2))
pref_df <- survfit(res.cox, newdata = new_dat)
pref_df
```

```
## Call: survfit(formula = res.cox, newdata = new_dat)
##
```

```
##       n  events  median 0.95LCL 0.95UCL
##      76      58      15      12      30
```

2. Pointwise Confidence Interval - Bootstrap Method

Task: Follow #4 in EP-2 to construct a 90% pointwise confidence interval based on the bootstrap method.

3. Reference | Literature Review

Task: Skim through the CP-1 references given in Files> Projects> CP-1> Reference> directory to write a two-page report for summarizing the work there.

Ref-1.0 and Ref-1.1 introduces the semiparametric regression, which means the combination of parametric(may be mis specified and inconsistent) and nonparametric models. So, there are many assumptions should be satisfied or verified. Although some assumptions are difficult to verify, they are generally satisfied for well-behaved estimators. But, they make Andrews' MINPIN, the semiparametric estimator has the same asymptotic distribution as the idealized estimator obtained by replacing the nonparametric estimate with the true function, not easy to implement. Semi-parametric regression is typically used in cases where a nonparametric model may not perform well enough or where a parametric model error distribution is unknown. Semiparametric regression owns three popular methods: 1) partially linear, 2) index and 3) varying coefficient models. Reference 2.0 and 2.1 focus on introducing the process and characters of PH model. Firstly it describes the definition of survivor function S(t) , hazard function h(t) and cumulative hazard function H(t) which bring interpretability, analytic simplifications and modeling simplifications. Secondly, it introduces the situation with censored data which is classified into three categories: 1) clearly informative: Type I or II Censoring. 2) noninformative and 3) Less clear situation. Thirdly, after the introduction, they move to the process of PH model, which shows X are time independent. And PH model is popular due to robustness, non-negative hazards, easily compute the hazard ratio and can estimate h(t,X) and S(t,X). Fourthly, reference talks two types of likelihood in ML Estimation of the Cox PH Model. 1) full likelihood 2) partial likelihood which considers probabilities for subject who fail and does not consider probabilities for censored subjects explicitly. Full likelihood allows us to estimate the baseline hazard function, but the partial likelihood allows us to make estimates of parameter B while accounting for censored data. Once partial likelihood is maximized with respect to the regression parameter, it can then be used to eventually estimate the baseline hazard function. A similar partial likelihood process (penalized partial likelihood) can be utilized in addition to iterative sure independence screening to allow high-dimensional variable selection, which apparently has very small false selection while maintaining small mean squared error. This type of selection through penalization is an extension of classical selection techniques such a stepwise and bootstrap procedures to be used for PH regression, which means simulations can be used to demonstrate the viability of said selection methods Fifthly, reference also introduces the estimation of survival curves called adjusted survival curves. The last but not the least, as the content in reference 1, there are several assumptions in semiparametric regression, so does PH model. Reference 2.2 gives us a real world case in population-based cancer survival analysis using PH model. Patient survival rates provide such a measure to effectively diagnose and treat the cancers that arise and require a means of measuring progress in this specific area. There are several measures including cause-specific survival, which can estimate net survival,

and relative survival which is calculated by observed survival proportion divided by expected survival proportion. And, it uses flexible parametric models with restricted cubic splines to fit relative survival curve other than cox model that cannot be applied to model a difference in two rates. Reference 2.3 talks about variable selection for Cox's proportional hazards model in high dimensional space by extending the sure screening procedure to an iterative version. It extends the key idea of SIS and ISIS to handle Cox's proportional hazards model: 1) ranking by marginal utility 2) Conditional feature ranking and iterative feature selection3) new variants of SIS and ISIS for reducing FSR. Finally, numerical simulation studies have shown encouraging performance of the proposed method in comparison with other techniques such as LASSO. Reference 3.1 introduces the Bayesian variable selection for proportional hazards regression models with right censored data where a nonparametric prior is specified for the baseline hazard rate with the use of discrete gamma process and a semi-automatic parametric informative prior specification that focuses on the observables for the regression coefficients and the model space. In addition, it proposes a Markov chain Monte Carlo method to compute the posterior model probabilities. Reference 4.1 proposes a piecewise exponential representation of the original survival data to link hazard regression with estimation schemes, which is based on the Poisson likelihood. And it makes recent advances for model building in exponential family regression accessible and in the nonproportional hazard regression. The reason why coming the above new method is that recent statistical methods typically introduce additional difficulties, such as immediate appeal in terms of flexibility, when a subset of covariates and the corresponding modeling alternatives have to be chosen. The article implement a two-stage stepwise selection approach, an approach based on doubly penalized likelihood, and a component wise functional gradient descent approach will be adapted to the piecewise exponential regression problem. Reference 5.1 shows using machine learning to do survival analysis. Because there are always some censored data which can be effectively handled using survival analysis techniques. Although above references developed traditional statistical approaches to overcome this censoring issue. In addition, many machine learning algorithms are adapted to effectively handle survival data and tackle other challenging problems that arise in real world data. It provides a comprehensive and structured review of the representative statistical methods along with the machine learning techniques used in survival analysis and provide a detailed taxonomy of the existing methods. One can perform survival trees, neural networks, bayesian methods (as discussed prior), support vector machines, boosting, and other such advanced machine learning techniques.

4. Implementation: Reference Procedure

Task: Outline steps for implementing one of the studied procedure addressed in the reference. You DO NOT need to implement them, but describe how to do it.

A method of survival analysis provided in the references was the accelerated failure time (AFT) model. This is a parametric alternative to the Cox Proportional Hazards model, which is originally semi-parametric. As the AFT model is parametric, certain assumptions must first be met. The first thing to understand about the model is that the primary differences lies with these assumptions. Keep in mind that in an AFT model, the survival time will "contract" or "stretch" as a function of the model's predictor variables (slightly different interpretaion compared to Cox PH). This stems from the idea that in AFT models the predictor variables have a multiplicative event on the log of survival time. The big difference (and assumption) for an AFT model is that it assumes an underlying distribution for its survival times.

The general equation for an accelerated failure time model is

$$log(T) = B_0 + B_1X_1 + ... + B_pX_p + \epsilon\sigma$$

where T is the survival time, the B values are coefficients of each predictor's effect on the log(T) (since survival times are always positive). The sigma term in the model equation is a scale parameter, which depends upon the shape of the underlying distribution, and the error term is assumed to be independent and identically distributed (as well as independent of the X-values). As it acts as a parametric alternative to the Cox PH regression model, the AFT model also commonly employs the Weibull and Exponential disitributions, though the typical distribution used is the log-logistic distribution. As we can see from the simplied AFT model equation, AFT models (parametric as they are) are analaous to generalized linear regression models in the sense that the results of the model can be interpreted similarly to how a typical regression can be.

In R, it is simple to create an AFT model for survival analysis. Similarly to how we plot and interpret data as above in PH regression, we use the surv() function to establish that our model will be used to interpret survival data. The difference with AFT models lies in the usage of the survreg() function, which in R allows us to establish parameters (most notably distribution shape) for our model. This allows us to specify whether we want to use a Weibull, exponential, gamma, or log-logistic distribution. There are certain advantages to each type of distribution depending on circumstance, but log-logistic has the advantage of allowing non-monotonic hazard functions, which the Weibull distribution cannot replicate.

Once the regression is performed, one can even compare the effectiveness of models with differing underlying distributions through AIC comparisons. One must simply compare the AIC values of each different model to see which has the lower AIC value, though this is not the only method of comparison. As we would have a model with multiple predictor variables, one can also perform variable selection to try to cut variables that are not as explanatory in the context of survival time or that are simply insignificant based on given p-value (or simply removed through stepwise AIC).

In interpreting the coefficients, one must note that the equation is based on a logarithm of time, so once we get the values of the B coefficients, we must rearrange the equation.

$$\hat{\gamma} = exp(\hat{\alpha})$$

This allows us to calaculate our acceleration factor. For example, if we have a positive acceleration factor, for example 2, then we would be able to say that the variable increases our survival time by a factor of 2. If we have a negative value of 2 as our acceleration factor, the survival time would be "accelerated" or shortened by a factor of 2. Keep in mind that the predictor variables' effects are indeed multiplicative.

Note: As with most comparisons between parametric and non(or semi-)parametric methods, parametric models have the advantage in terms of ease of interpretability (the goal of a parametric model is to find a certain parameter that helps build a model representative of the data. This also leads to a disadvantage: parametric models require a foundational distribution of the data with which to base the model off of, which may not always be feasible. A violation of assumptions in a parametric model would then lead us to potentially sub-optimal results.

5. Result Review

Task: Discuss whether the results getting in (2) and (4) might be different (in what way).

Code Appendix

PH Regression Model Fitting Code

The following code demonstrates our approach to fit each potential co-variate independently to estimate its impact in context of the survival function, utilized in task 1:

```
#extract name of var, estimate, and p-value
#helper to extract model summaries in tabular format
get_cox_ph_results <- function(coxph_res){
  zph <- cox.zph(coxph_res)
  res_table <- data.frame(covariate = names(coxph_res$coefficients),
            beta = as.numeric(coxph_res$coefficients),
            pvalue =
             as.numeric(summary(coxph_res)$coefficients[colnames(summary(coxph_res)$coefficients
            rho = as.numeric(zph$table[,1])[!is.na(zph$table[,1])] )
  return(res_table)
}

variables <- c('age','sex','disease','frail')

results <- do.call('rbind',
          lapply(variables, function(x){
                  model_formula <- as.formula(paste0("Surv(time, status) ~ ",x))
          model_res <- coxph(formula = model_formula, data = kidney)
          return(get_cox_ph_results(model_res))
          }))
```

Bootstrap Procedure

The following outlines our bootstrap procedure for section 1.3, utilizing functionality from the *boot* package in R:

```
# must pass indices argument so that bootstrap can randomly choose - in this case row index
# in this case only using age as a covariate below
suppressPackageStartupMessages( library(boot))
#bootstrap function - age coefficient only here
bs_fun <- function(data, indices){
    bs_dat <- data[indices,]
    res.cox <- coxph(Surv(time, status) ~ age , data = bs_dat)
    return(as.numeric(res.cox$coefficients["age"]))
}

bs_res <- boot(kidney, bs_fun, R=2000) # R = number of replications

#plot(bs_res) #if symmetric - ok to use percentile approach
# type options include: "norm", "basic", "perc", "stud"

bs_ci <- boot.ci(bs_res, conf = 0.95, var.t0 = NULL, type = 'perc')
```

Questions?

Contact: ygao390, kylee20, ywan40, agovindaraj6, pwilliams60, rzhang438 | @gatech.edu