Coxnet: Regularized Cox Regression

Noah Simon
Jerome Friedman
Trevor Hastie
Rob Tibshirani

February 18, 2012

1 Introduction

We will give a short tutorial on using coxnet. Coxnet is a function which fits the Cox Model regularized by an elastic net penalty. It is used for underdetermined (or nearly underdetermined systems) and chooses a small number of covariates to include in the model. Because the Cox Model is rarely used for actual prediction, we will rather focus on finding and interpretating an appropriate model. We give a simple example of how to format data and run the Cox Model in glmnet with cross validation.

2 Example

We first load our data and set up the response. In this case x must be an n by p matrix of covariate values — each row corresponds to a patient and each column a covariate. y is an n length vector of failure/censoring times, and status is an n length vector with each entry, a 1 or a 0, indicating whether the corresponding entry in y is indicative of a failure time or right censoring time (1 for failure, 0 for censoring)

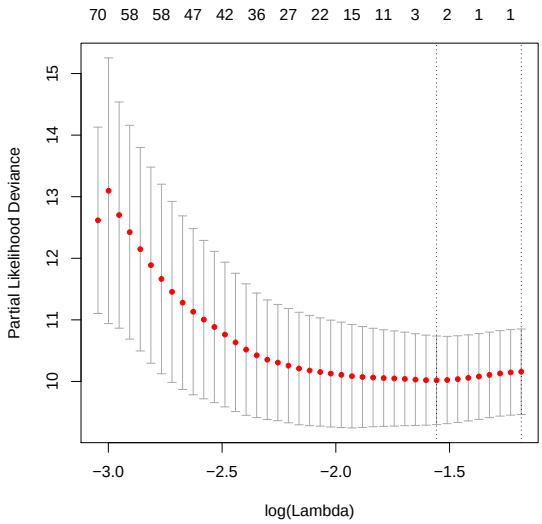
```
> library("glmnet")
> library("survival")
> load(system.file("doc","VignetteExample.rdata",package="glmnet"))
> attach(patient.data)
```

We then call our functions to fit with the lasso penalty ($\alpha = 1$), and cross validate. We set `maxit = 1000` (increasing the maximum number of iterations to 1000) because our data is relatively high dimensional, so more iterations are needed for convergence. In practice, the function will spit out an error if convergence isn't reached by the maximum number of iterations.

```
> cv.fit <- cv.glmnet(x, Surv(time, status), family="cox", maxit = 1000)
> fit <- glmnet(x, Surv(time,status), family =  "cox", maxit = 1000)
```

The `Surv` function packages the survival data into the form expected by glmnet. Once fit, we can view the optimal λ value and a cross validated error plot to help evaluate our model.

```
> plot(cv.fit)
```




```
> cv.fit$lambda.min

[1] 0.2107387
```

The left vertical line in our plot shows us where the CV-error curve hits its minimum. The right vertical line shows us the most regularized model with CV-error within 1 standard deviation of the minimum. In this case, we see that the minimum was achieved by a fairly regularized model, however the right line indicates that the null model (no coefficients included) is within 1 sd of the minimum. This might lead us to believe that in actuality the covariates are not explaining any variability. For the time being we will concern ourselves with the minimum CV-error model. We can check which covariates our model chose to be active, and see the coefficients of those covariates.

```
> Coefficients <- coef(fit, s = cv.fit$lambda.min)
> Active.Index <- which(Coefficients != 0)
> Active.Coefficients  <- Coefficients[Active.Index]
```

`coef(fit, s = cv.fit$lambda.min)` returns the p length coefficient vector of the solution corresponding to $\lambda =$`cv.fit$lambda.min`.

```
> Active.Index

[1]  80 394

> Active.Coefficients

[1] 0.4013208 0.1139639
```

We see that our optimal model chose 2 active covariates ($X80$ and $X394$) each with a small positive effect on hazard.