Cox with lasso

STK4080 - 22/10/2018

jonasfsc@math.uio.no

Theory

Cox with lasso

Data

$$(t_i, \delta_i, x_i)$$

Standard Cox

Maximize log partial likelihood

$$l(\beta) = \sum_{i=1}^{n} \delta_i \left(\beta^T x_i - \log\{\sum_{j \in R_i} \exp(\beta^T x_j)\} \right)$$

Cox with lasso (**l**east **a**bsolute **s**hrinkage and **s**election **o**perator) - Tibshirani (1997)

Maximize log partial likelihood subject to restriction

$$l(\beta) \quad \text{subject to} \qquad \sum_j |\beta_j| \leq c$$

Cox on a budget!

Why would we penalize coefficients?

Allows us to work with data that have p>n

Shrinkage

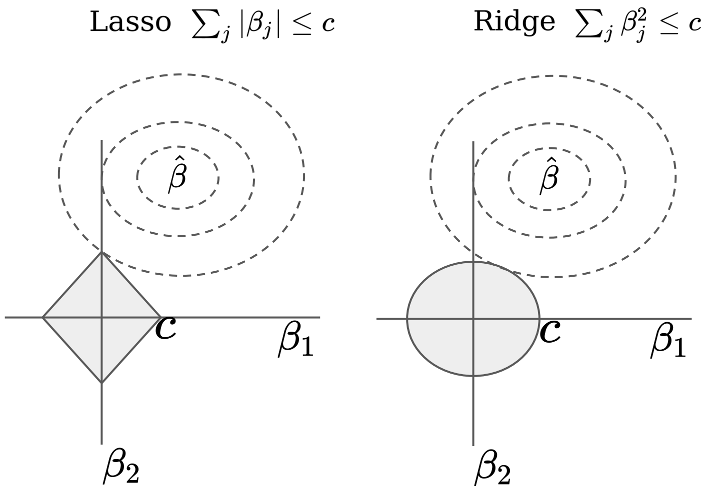
Lower MSE, trade bias for lower variance

Variable selection

Puts coefficients equal to zero

Good for interpretation that few covariates explain the effects

$$\max_{\beta} l(\beta) \quad \text{subject to}$$



Rewrite problem

Constrained optimization problem

$$\max_\beta \left\{\sum_{i=1}^n \delta_i \left(\beta^T x_i - \log S_n(t_i,\beta)\right)\right\}$$
$$\text{subject to} \quad \sum_j |\beta_j| \leq c$$

Constraint -> Lagrange multiplier method (KKT conditions)

$$\max_\beta \left\{\sum_{i=1}^n \delta_i \left(\beta^T x_i - \log S_n(t_i,\beta)\right) - \lambda \sum_j |\beta_j|\right\}$$

One-to-one correspondence between constrained form and Lagrangian form

or $$\min_\beta \left\{-l(\beta) + \lambda \sum_j |\beta_j|\right\}$$

Lambda

Tuning parameter

Cross validation

Simple and standard

Split data into k folds. Use all but fold j to estimate coefficients $\hat{\beta}^{-j}(\lambda)$

Compute deviance for fold j

$$\begin{aligned}\widehat{Dev}_\lambda^j &= Dev\left(\hat{\beta}^{-j}(\lambda)\right) - Dev^{-j}\left(\hat{\beta}^{-j}(\lambda)\right)\\ &= -2\left[l\left(\hat{\beta}^{-j}(\lambda)\right) - l^{-j}\left(\hat{\beta}^{-j}(\lambda)\right)\right]\end{aligned}$$

Choose lambda that minimizes

$$Dev(\lambda) = \sum_{j=1}^{k} \widehat{Dev}_\lambda^j$$

Examples

R

Tibshirani, Hastie and Friedman's package *glmnet* is good for penalized regression models. Minimizes

$$-\frac{l_n(\beta)}{n} + \lambda \sum_j |\beta_j|$$

"Scaled" lambda

Often standardize X such that columns are centered and have unit variance

$$\sum_i x_{ij}/n = 0$$
$$\sum_i x_{ij}^2/n = 1$$

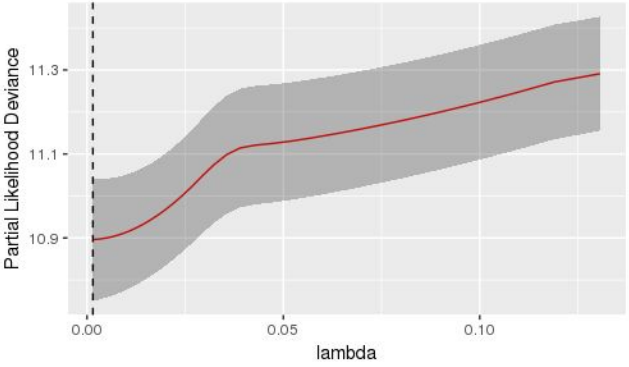
Australian data

n = 238

p = 3, methadone dosage, prison and clinic

Cox gives best model by AIC as p=3

Cox with lasso **without standardizing**



Comparison

Lambda_scaled = 0.0015 = 0.357 / 238

Coefficients

	lasso	cox
	<dbl>	<dbl>
1	-3.39	-3.54
2	0.311	0.327
3	-0.991	-1.01

	term	step	lambda
	<chr>	<dbl>	<dbl>
1	clinic	95	0.131
2	methadon	1017	0.0384
3	prison	1025	0.0376

Lymphoma

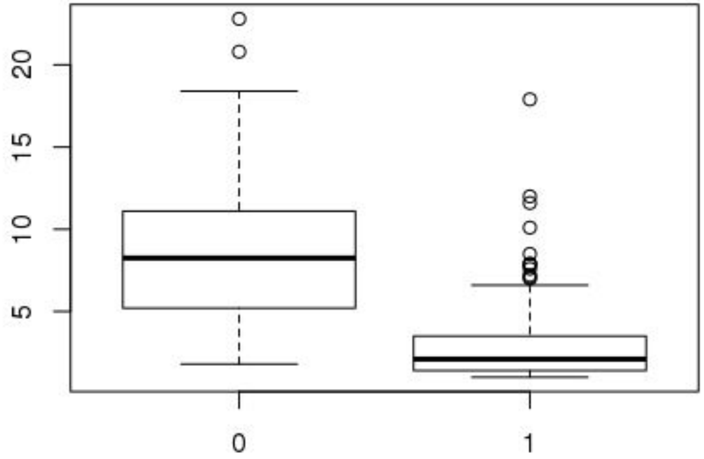
Data taken from Statistical Learning with Sparsity's website

n = 240

p = 7399

79 with death, 161 with censoring

boxplot, time ~delta

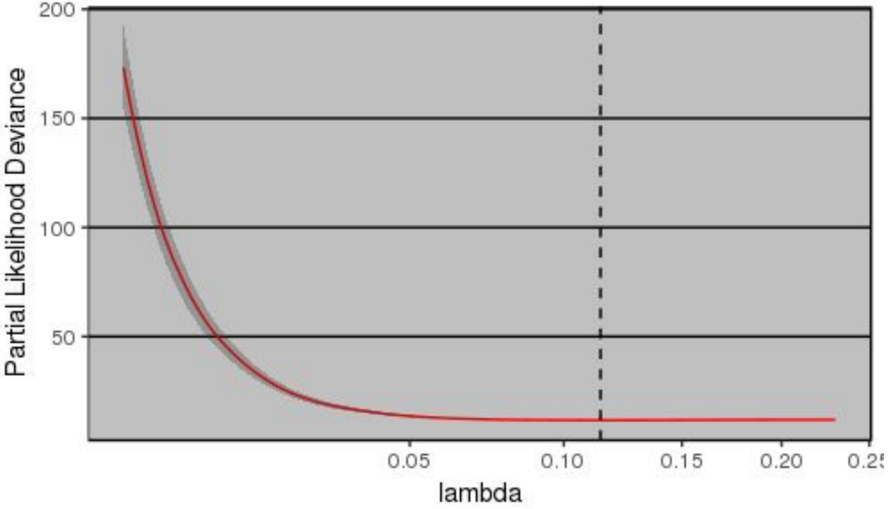


Optimal lambda

$$Dev(\lambda) = \sum_{j=1}^{k} \widehat{Dev}_{\lambda}^{j}$$
$$= -2 \sum_{j} \left[l\left(\hat{\beta}^{-j}(\lambda)\right) - l^{-j}\left(\hat{\beta}^{-j}(\lambda)\right) \right]$$

10-fold cross validation

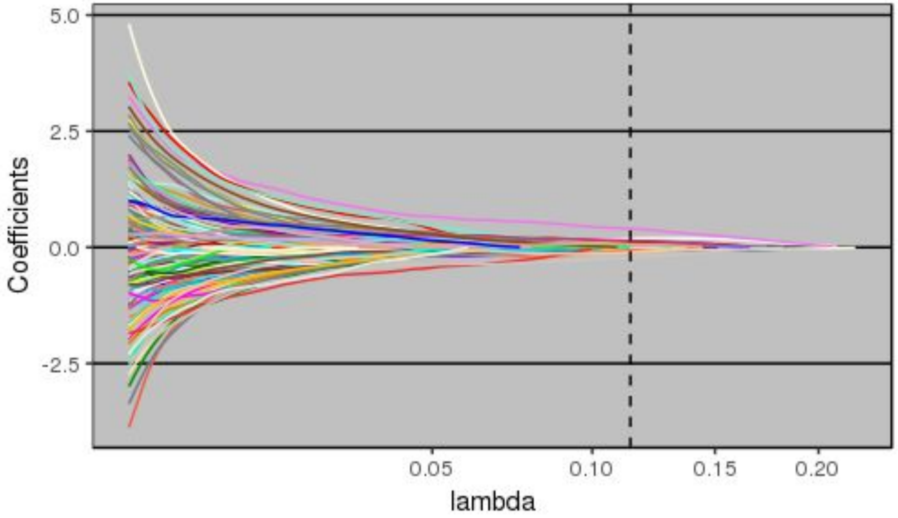
Standardized X



Coefficients

Only left with 28 non-zero coefficients!

Note trumpet shape of coefficient path



References

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

Tibshirani, R. (1997). The lasso method for variable selection in the Cox model. *Statistics in medicine*, *16*(4), 385-395.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

Script for reproducing blood cancer results in R

```
library(survival)
library(glmnet)

time = scan("https://web.stanford.edu/~hastie/StatLearnSparsity_files/DATA/lymphtim.txt")
delta = scan("https://web.stanford.edu/~hastie/StatLearnSparsity_files/DATA/lymphstatus.txt")
X = matrix(scan("https://web.stanford.edu/~hastie/StatLearnSparsity_files/DATA/lymphx.txt"),
           byrow=T,nrow=length(time))

cv.fit = cv.glmnet(X, Surv(time, delta),
                   family="cox",
                   standardize=T,
                   nfolds=10)

plot(cv.fit)
plot(cv.fit$glmnet.fit, xvar="lambda")
coef(cv.fit, s=cv.fit$lambda.min)[coef(cv.fit, , s=cv.fit$lambda.min)!=0]
```

```
# for uglier and less informative plots
library(ggplot2)
library(ggthemes)
library(broom)

tidied_cv = tidy(cv.fit)

ggplot(tidied_cv, aes(lambda, estimate)) +
  geom_line(color = "red") +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  scale_x_sqrt() +
  geom_vline(xintercept = cv.fit$lambda.min, lty=2) +
  ylab("Partial Likelihood Deviance")
  ylim(0, 200) +
  theme_excel()

tidy_covariates = tidy(cv.fit$glmnet.fit)
cols = colors()[!grepl("grey", colors()) & ! grepl("gray", colors())]
col_sam = sample(cols, length(unique(tidy_covariates$term)))

ggplot(tidy_covariates, aes(x=lambda, y=estimate, color=as.factor(term)))+
  geom_line() +
  guides(color=FALSE) +
  geom_vline(xintercept = cv.fit$lambda.min, lty=2) +
  scale_x_sqrt()+
  theme_excel()+
  scale_color_manual(values=col_sam)+
  ylab("Coefficients")
```