

Coding manual for “Bayesian variable selection in the AFT model with an application to the SEER breast cancer data”

For each of the methods, we report the results including false negatives/positives based on $M = 500$ simulated data (different M replications for different methods) under the same simulation design.

Cox-LASSO

```
> rm(list=ls())
> library(glmnet)
> # simulation 1
> M <- 500
> mysBeta <- matrix(0,M,20)
> mydirs = './';
> for(k in 1:M){
+   cat(k, ' ')
+   if(!k%%20) cat('\n')
+   sdat <- readMat(paste(mydirs, 'simudat', k, '.mat', sep=' '))
+   x <- as.matrix(sdat$Z)
+   y <- cbind(time=as.numeric(sdat$V), status=as.integer(sdat$Delta))
+   # first do 10-fold cross-validation to select lambda
+   cvfit <- cv.glmnet(x,y, family="cox")
+   # plugin the optimal lambda
+   fit <- glmnet(x,y, family="cox", lambda=cvfit$lambda.min)
+   mysBeta[k,which(fit$beta!=0)] <- 1
+ }
```

Generalized F with forward/backward selection based on BIC

```
> rm(list = ls())
> library(R.matlab)
> library(flexsurv)
> library(survival)
> options(warn = -1)
> repl <- as.numeric(commandArgs(TRUE))
> myfilename <- paste("out", repl, sep = "")
> M <- 500
> mysBeta_gf <- matrix(0, M, 20)
> mydirs_gf = "./"
> forward <- FALSE
> ptm <- proc.time()[3]
> count <- 0
> for (k in repl) {
+   cat("\n\nreplication =", k, "\n")
+   print(repl)
+   if (!k%%20)
+     cat("\n")
+   sdat <- readMat(paste(mydirs_gf, "simudat", k, ".mat", sep = ""))
+   p <- ncol(tmp <- as.data.frame(sdat$Z))
+   names(tmp) <- paste("X", 1:p, sep = "")
+   tmp$time <- as.numeric(sdat$V)
+   tmp$status <- as.integer(sdat$Delta)
+   if (forward) {
+     fit0 <- flexsurvreg(Surv(time, status) ~ 1, data = tmp,
```

```

+         dist = "lnorm")
+     coefs <- as.numeric(coef(fit0))
+     init0 <- c(coefs[1], exp(coefs[2]), 0, 1e-06, coefs[-c(1:2)])
+     fit <- update(fit0, inits = init0, data = tmp, dist = "genf")
+     myfit <- fit
+     len <- length(indc <- 1:p)
+     q <- length(inds <- numeric(0))
+     aic0 <- BIC(myfit)
+     aics <- aic0 - 1
+     while (min(aics) < aic0 && q <= p) {
+         cat("q = ", q, " ", sep = "")
+         aic0 <- BIC(myfit)
+         aics <- numeric(len)
+         fits <- list(len)
+         for (i in 1:len) {
+             cat("X", indc[i], " ", sep = "")
+             inds0 <- c(inds, indc[i])
+             myform0 <- as.formula(paste("Surv(time, status) ~ ",
+                 paste(names(tmp)[inds0], collapse = "+")))
+             fit0 <- flexsurvreg(myform0, data = tmp, dist = "lnorm",
+                 method = "Nelder-Mead")
+             coefs <- as.numeric(coef(fit0))
+             init0 <- c(coefs[1], exp(coefs[2]), 0, 1e-06,
+                 coefs[-c(1:2)])
+             fits[[i]] <- update(fit0, inits = init0, dist = "genf",
+                 method = "Nelder-Mead")
+             aics[i] <- BIC(fits[[i]])
+             count <- count + 1
+         }
+         cat("\n")
+         optind <- which(aics == min(aics))
+         if (min(aics) < aic0) {
+             myfit <- fits[[optind]]
+             cat("X", indc[optind], " was chosen\n\n", sep = "")
+         }
+         q <- length(inds <- c(inds, indc[optind]))
+         len <- length(indc <- indc[optind])
+     }
+ }
+
+ if (!forward) {
+     myform0 <- as.formula(paste("Surv(time, status) ~ ",
+         paste(names(tmp)[1:p], collapse = "+")))
+     fit0 <- flexsurvreg(myform0, data = tmp, dist = "lnorm",
+         method = "Nelder-Mead")
+     coefs <- as.numeric(coef(fit0))
+     init0 <- c(coefs[1], exp(coefs[2]), 0, 1e-06, coefs[-c(1:2)])
+     fit <- update(fit0, inits = init0, data = tmp, dist = "genf")
+     myfit <- fit
+     q <- length(inds <- 1:p)
+     aic0 <- BIC(myfit)
+     aics <- aic0 - 1
+     while (min(aics) < aic0 && q > 0) {
+         cat("q = ", q, " ", sep = "")
+         aic0 <- BIC(myfit)

```

```

+         cat(round(aic0), ": ", sep = "")
+         aics <- numeric(q)
+         fits <- list(q)
+         for (i in 1:q) {
+             cat("X", inds[i], " ", sep = "")
+             inds0 <- inds[-i]
+             myform0 <- as.formula(paste("Surv(time, status) ~ ",
+                 paste(names(tmp)[inds0], collapse = "+")))
+             fit0 <- flexsurvreg(myform0, data = tmp, dist = "lnorm",
+                 method = "Nelder-Mead")
+             coefs <- as.numeric(coef(fit0))
+             init0 <- c(coefs[1], exp(coefs[2]), 0, 1e-06,
+                 coefs[-c(1:2)])
+             fits[[i]] <- update(fit0, inits = init0, dist = "genf")
+             aics[i] <- BIC(fits[[i]])
+             count <- count + 1
+             cat(round(aics[i]), " ", sep = "")
+         }
+         cat("\n")
+         optind <- which(aics == min(aics))
+         if (min(aics) < aic0) {
+             myfit <- fits[[optind]]
+             cat("X", inds[optind], " was excluded\n\n", sep = "")
+         }
+         q <- length(inds <- inds[-optind])
+     }
+ }
+     mysBeta_gf[k, inds] <- 1
+ }
> cputime <- as.numeric(proc.time())[3] - ptm)
> cputime <- cputime/60
> cat(count, "search steps done with CPUtime", round(cputime, 3),
+     "minutes: completed!", "\n")
> save(file = myfilename, "mysBeta_gf")

```