CS6220: Data mining techniques
Multiple regression

Olga Vitek

October 8, 2015

Outline

Subset regression

Variable selection by cross-validation

Another example: stepwise AIC

Ridge regression

Lasso regression

The bootstrap

Example: surgical unit

- Random sample of 54 patients undergoing a liver operation
- Response `surv` or `lsurv` post-operation survival (or log-survival) time
- Predictor variables
  - `blood` blood clotting score
  - `prog` prognostic index
  - `enz` enzyme function score
  - `liver` liver function score
  - `age` in years
  - `female` gender, 0=male, 1=female
  - `modAlc` and `heavyAlc` alcool use

Getting to know the data

```
> X <- read.table('/Users/ovitek/Dropbox/Olga/Teaching/CS6220/Fall15/Le
> dimnames(X)[[2]] <- c('blood', 'prog', 'enz', 'liver',
+  'age', 'female', 'modAlc', 'heavyAlc', 'surv', 'lsurv')
> dim(X)
```

```
[1] 54 10
```

```
> head(X)
```

```
  blood prog enz liver age female modAlc heavyAlc surv lsurv
1   6.7   62  81  2.59  50      0      1        0  695 6.544
2   5.1   59  66  1.70  39      0      0        0  403 5.999
3   7.4   57  83  2.16  55      0      0        0  710 6.565
4   6.5   73  41  2.01  48      0      0        0  349 5.854
5   7.8   65 115  4.30  45      0      0        1 2343 7.759
6   5.8   38  72  1.42  65      1      1        0  348 5.852
```

```
> sum(is.na(X))
```

```
[1] 0
```

Subset regression

Exhaustive search

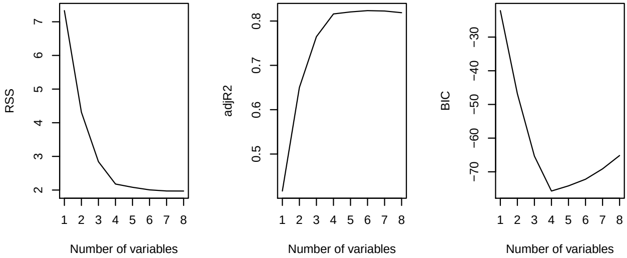
By default - exhaustive search

```
> library(leaps)
> regfit.full <- regsubsets(lsurv ~ ., data=X[,-9])
> reg.summary <- summary(regfit.full)
> names(reg.summary)

[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
> library(leaps)
> reg.summary

Subset selection object
Call: regsubsets.formula(lsurv ~ ., data = X[, -9])
8 Variables  (and intercept)
         Forced in Forced out
blood        FALSE      FALSE
prog         FALSE      FALSE
enz          FALSE      FALSE
liver        FALSE      FALSE
age          FALSE      FALSE
female       FALSE      FALSE
modAlc       FALSE      FALSE
heavyAlc     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         blood prog enz liver age female modAlc heavyAlc
1  ( 1 ) " "   " "  "*" " "   " " " "    " "    " "
2  ( 1 ) " "   "*"  "*" " "   " " " "    " "    " "
3  ( 1 ) " "   "*"  "*" " "   " " " "    " "    "*"
4  ( 1 ) "*"   "*"  "*" " "   " " " "    " "    "*"
5  ( 1 ) "*"   "*"  "*" " "   " " "*"    " "    "*"
6  ( 1 ) "*"   "*"  "*" " "   "*" "*"    " "    "*"
7  ( 1 ) "*"   "*"  "*" " "   "*" "*"    "*"    "*"
8  ( 1 ) "*"   "*"  "*" "*"   "*" "*"    "*"    "*"
```

Exhaustive search

```
> par(mfrow=c(2,3))
> plot(reg.summary$rss, xlab='Number of variables', ylab='RSS', type='l')
> plot(reg.summary$adjr2, xlab='Number of variables', ylab='adjR2', type='l')
> plot(reg.summary$bic, xlab='Number of variables', ylab='BIC', type='l')
> # Best model dimension
> which.min(reg.summary$bic)
[1] 4
> # Best model with 4 predictors
> coef(regfit.full, 4)
(Intercept)       blood        prog         enz    heavyAlc
 3.85241856  0.07332263  0.01418507  0.01545270  0.35296762
```



Forward selection

```
> regfit.full1 <- regsubsets(lsurv ~ ., method='forward', data=X[,-9])
> reg.summary1 <- summary(regfit.full1)
> which.min(reg.summary1$bic)

[1] 4

> coef(regfit.full1, 4)

(Intercept)       blood        prog         enz    heavyAlc
 3.85241856  0.07332263  0.01418507  0.01545270  0.35296762
```

```
> reg.summary1

Subset selection object
Call: regsubsets.formula(lsurv ~ ., method = "forward", data = X[,
    -9])
8 Variables  (and intercept)
         Forced in Forced out
blood        FALSE      FALSE
prog         FALSE      FALSE
enz          FALSE      FALSE
liver        FALSE      FALSE
age          FALSE      FALSE
female       FALSE      FALSE
modAlc       FALSE      FALSE
heavyAlc     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: forward
         blood prog enz liver age female modAlc heavyAlc
1  ( 1 ) " "   " "  "*" " "   " " " "    " "    " "
2  ( 1 ) " "   "*"  "*" " "   " " " "    " "    " "
3  ( 1 ) " "   "*"  "*" " "   " " " "    " "    "*"
4  ( 1 ) "*"   "*"  "*" " "   " " " "    " "    "*"
5  ( 1 ) "*"   "*"  "*" " "   " " "*"    " "    "*"
6  ( 1 ) "*"   "*"  "*" " "   "*" "*"    " "    "*"
7  ( 1 ) "*"   "*"  "*" " "   "*" "*"    "*"    "*"
8  ( 1 ) "*"   "*"  "*" "*"   "*" "*"    "*"    "*"
```

Variable selection by cross-validation

Cross-validation

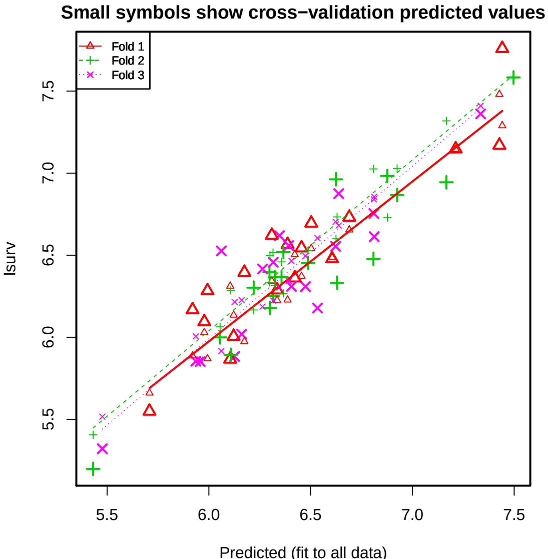
```
> # Fix all the predictors
> library(DAAG)
> lm.full <- lm(lsurv ~ ., data=X[,-9])
> summary(lm.full)
Call:
lm(formula = lsurv ~ ., data = X[, -9])

Residuals:
     Min       1Q   Median       3Q      Max
-0.35562 -0.13833 -0.05158  0.14949  0.46472

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.050515   0.251756  16.089  < 2e-16 ***
blood        0.068512   0.025422   2.695  0.00986 **
prog         0.013452   0.001947   6.909 1.39e-08 ***
enz          0.014954   0.001809   8.264 1.43e-10 ***
liver        0.008016   0.046708   0.172  0.86450
age         -0.003566   0.002752  -1.296  0.20163
female       0.084208   0.060750   1.386  0.17253
modAlc       0.057864   0.067483   0.857  0.39574
heavyAlc     0.388383   0.088380   4.394 6.69e-05 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ

Residual standard error: 0.2093 on 45 degrees of freedom
```

Cross-validation

```
> # Fix all the predictors
> CVlm(X[,-9], lm.full)
Analysis of Variance Table

Response: lsurv
          Df Sum Sq Mean Sq F value  Pr(>F)
blood      1   0.78    0.78   17.73 0.00012 ***
prog       1   2.59    2.59   59.11 9.8e-10 ***
enz        1   6.33    6.33  144.63 1.2e-15 ***
liver      1   0.02    0.02    0.56 0.45767
age        1   0.13    0.13    2.89 0.09615 .
female     1   0.05    0.05    1.19 0.28067
modAlc     1   0.09    0.09    2.03 0.16137
heavyAlc   1   0.85    0.85   19.31 6.7e-05 ***
Residuals 45   1.97    0.04
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ



fold 1
Observations in test set: 18
                1     3    5    12    16     20     22    23    26   29     31
Predicted   6.455 6.387 7.44 5.708 6.503 6.6898  6.105 6.174 6.309 5.92 5.9775
cvpred      6.372 6.227 7.29 5.660 6.541 6.6539  6.311 5.974 6.342 5.89 6.0293
lsurv       6.544 6.565 7.76 5.549 6.695 6.7310  5.866 6.395 6.621 6.17 6.0940
```

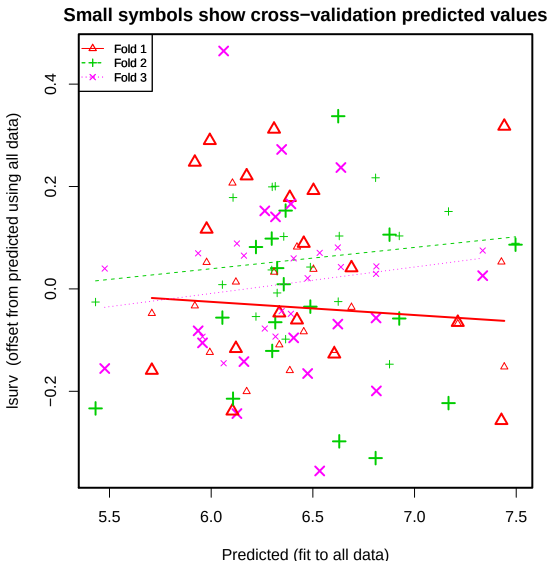
Cross-validation

```
> # Fix all the predictors
> CVlm(X[,-9], printit=FALSE, lm.full)
```



Cross-validation

```
> # Fix all the predictors
> CVlm(X[,-9], lm.full, printit=FALSE, plotit='Residual')
```



Cross-validation

Important!
The example above assumes that we are only interested in one model, which has all the predictors.
If we want to select a subset of predictors (e.g., using stepwize selection) we need to perform a separate step of subset selection within *each* fold of cross-validation.

Stepwise AIC

Stepwise AIC

```
> library(MASS)
> stepAIC(lm.full)
Start:  AIC=-161
lsurv ~ blood + prog + enz + liver + age + female + modAlc +
    heavyAlc

           Df Sum of Sq  RSS  AIC
- liver     1     0.001 1.97 -163
- modAlc    1     0.032 2.00 -162
- age       1     0.074 2.04 -161
<none>                  1.97 -161
- female    1     0.084 2.05 -160
- blood     1     0.318 2.29 -155
- heavyAlc  1     0.846 2.82 -144
- prog      1     2.090 4.06 -124
- enz       1     2.991 4.96 -113

Step:  AIC=-163
lsurv ~ blood + prog + enz + age + female + modAlc + heavyAlc

           Df Sum of Sq  RSS    AIC
- modAlc    1      0.03 2.01 -163.8
<none>                  1.97 -162.7
- age       1      0.09 2.06 -162.4
- female    1      0.10 2.07 -162.1
- blood     1      0.63 2.60 -149.8
```

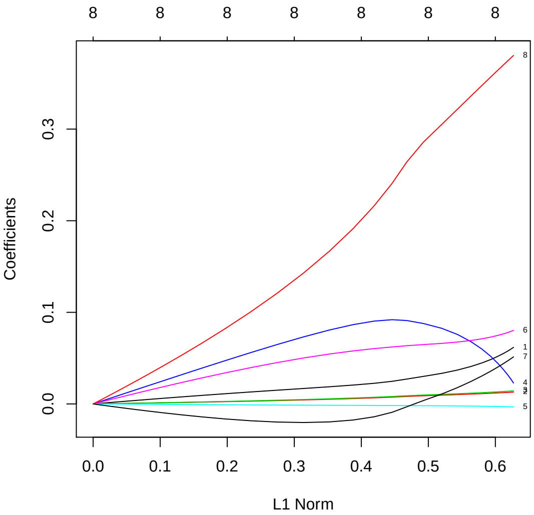
Ridge regression

Ridge regression

```
> library(glmnet)
> grid=10^seq(10,-2,length=100)
> ridge.mod <- glmnet(x=as.matrix(X[,-c(9:10)]),y=X[,10],alpha=0,lambda=grid)
> names(ridge.mod)

 [1] "a0"        "beta"      "df"        "dim"       "lambda"    "dev.ratio"
 [7] "nulldev"   "npasses"   "jerr"      "offset"    "call"      "nobs"

> ridge.mod$lambda[20]

[1] 49770236

> coef(ridge.mod)[,20]

(Intercept)       blood        prog         enz       liver         age
   6.43e+00    7.39e-10    1.34e-10    1.48e-10    2.92e-09   -6.27e-11
     female      modAlc    heavyAlc
   2.22e-09   -1.21e-09    4.57e-09

> ridge.mod$lambda[95]

[1] 0.0404

> coef(ridge.mod)[,95]

(Intercept)       blood        prog         enz       liver         age
    4.30462     0.04847     0.01166     0.01278     0.05191    -0.00268
     female      modAlc    heavyAlc
    0.07294     0.03577     0.35691
```

Ridge regression

```
> plot(ridge.mod, label=TRUE)
```



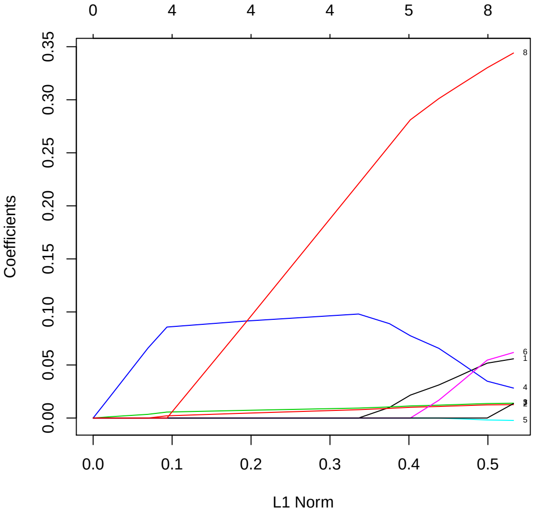
Lasso regression

Lasso regression

```
> lasso.mod  <- glmnet(x=as.matrix(X[,-c(9:10)]), y=X[,10], alpha=1,
+          lambda=grid)
> names(lasso.mod)

 [1] "a0"        "beta"      "df"        "dim"       "lambda"    "dev.ratio"
 [7] "nulldev"   "npasses"   "jerr"      "offset"    "call"      "nobs"

> lasso.mod$lambda[20]

[1] 49770236

> coef(lasso.mod)[,20]

(Intercept)       blood        prog         enz       liver         age
       6.43        0.00        0.00        0.00        0.00        0.00
     female      modAlc    heavyAlc
       0.00        0.00        0.00

> lasso.mod$lambda[95]

[1] 0.0404

> coef(lasso.mod)[,95]

(Intercept)       blood        prog         enz       liver         age
     4.5198      0.0216      0.0101      0.0114      0.0775      0.0000
     female      modAlc    heavyAlc
     0.0000      0.0000      0.2811
```

Lasso regression

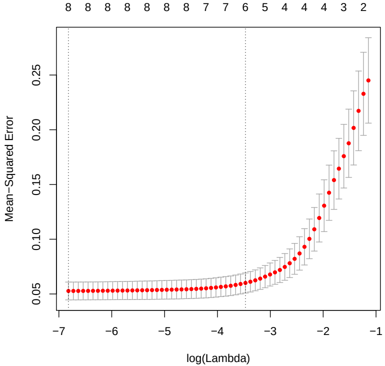
```
> plot(lasso.mod, label=TRUE)
```



Lasso regression

```
> cv.out <- cv.glmnet(x=as.matrix(X[,-c(9:10)]), y=X[,10], alpha=1)
> plot(cv.out)
> bestlam <- cv.out$lambda.min
> bestlam
```

```
[1] 0.00109
```



The bootstrap

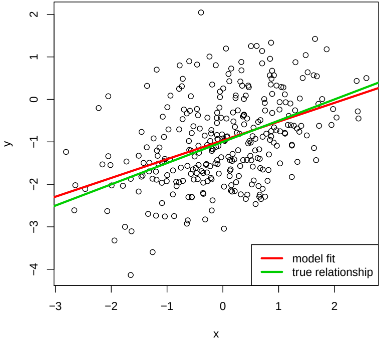
Simulate data with known answer

```
> set.seed(123)
> n <- 300
> eps1 = rnorm(n)
> x = rnorm(n)
> y = -1 + 0.5*x + eps1
> fit = lm(y~x)
> summary(fit)

Call:
lm(formula = y ~ x)

Residuals:
   Min     1Q Median     3Q    Max 
-2.440 -0.615 -0.102  0.580  3.183 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.9650     0.0546  -17.68  < 2e-16 ***
x             0.4420     0.0553    7.99  2.9e-14 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ

Residual standard error: 0.946 on 298 degrees of freedom
Multiple R-squared:  0.176,	Adjusted R-squared:  0.174 
F-statistic: 63.9 on 1 and 298 DF,  p-value: 2.95e-14
```

Simulate data with known answer

```
> plot(x, y)
> abline(fit, lwd=3, col=2)
> abline(-1, 0.5, lwd=3, col=3)
> legend('bottomright', legend = c("model fit", "true relationship"),
+        lwd=3, col=2:3)
> confint(fit)
```

```
             2.5 % 97.5 %
(Intercept) -1.072 -0.858
x            0.333  0.551
```

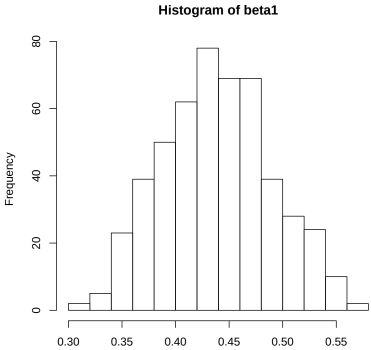


Bootstrap confidence interval

```
> B <- 500
> beta1 <- rep(NA, B)
> for (i in 1:B) {
+   selectObservations <- sample(1:n, size=n, replace=TRUE)
+   beta1[i] <- coef(lm(y[selectObservations] ~ x[selectObservations]))[2]
+ }
> quantile(beta1, c(0.05/2, 0.5, 1-0.05/2))
```

```
 2.5%   50% 97.5%
0.345 0.438 0.539
```

```
> hist(beta1)
```


Histogram of beta1
Frequency