Regularization Methods

Business Analytics
Stefan Feuerriegel

Outline

Setup

- Predicting salaries of U. S. baseball players based on game statistics
- Loading data `Hitters`

```r
library(ISLR) # Hitters is located inside ISLR
data(Hitters)
Hitters <- na.omit(Hitters) # salary can be missing
```

Outline

Ridge Regression in R

- Loading package `glmnet` which implements ridge regression

```r
library(glmnet)
```

- Main function `glmnet(x, y, alpha=0)` requires dependent variable `y` and regressors `x`
- Function only processes numerical input, whereas categorical variables needs to be transformed via `model.matrix(...)`

Ridge Regression in R

- Prepare variables

```r
set.seed(0)

# drop 1st column with intercept (glmnet has already one)
x <- model.matrix(Salary ~ ., Hitters)[, -1]
y <- Hitters$Salary
train_idx <- sample(nrow(x), size=0.9*nrow(x))

x.train <- x[train_idx, ]
x.test <- x[-train_idx, ]
y.train <- y[train_idx]
y.test <- y[-train_idx]
```

- Call ridge regression and automatically test a sequence of λ

```r
lm.ridge <- glmnet(x.train, y.train, alpha=0)
```

Ridge Regression in R

- `coef(...)` retrieves coefficients belonging to each λ

```
dim(coef(lm.ridge))

## [1]  20 100
```

$\rightarrow$ here: 100 models with different λ and each with 20 coefficients

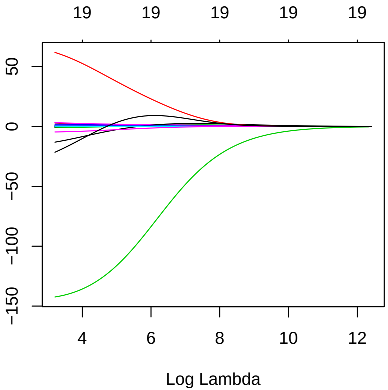
- For example, the 50th model is as follows

```
lm.ridge$lambda[50]       # tested lambda value

## [1] 2581.857

head(coef(lm.ridge)[,50]) # estimated coefficients

##  (Intercept)        AtBat         Hits        HmRun         Runs
## 211.76123020   0.08903326   0.37913073   1.21041548   0.64115228
##          RBI
##   0.59834311
```

Ridge Regression in R

- `plot(model, xvar="lambda")` investigates the influence of λ on the estimated coefficients for all variables

```
plot(lm.ridge, xvar="lambda")
```



- Bottom axis gives $\ln \lambda$, top the number of non-zero coefficients

Parameter Tuning

- Optimal λ is determined via cross validation by minimizing the mean squared error from a prediction
- Usage is `cv.glmnet(x, y, alpha=0)`

```
cv.ridge <- cv.glmnet(x.train, y.train, alpha=0)
```

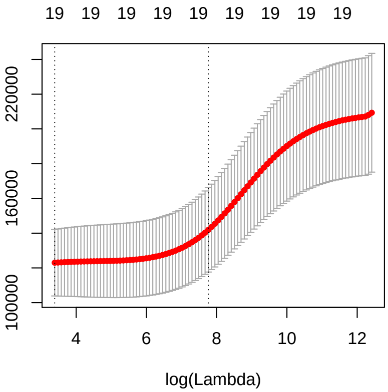
- Optimal λ and corresponding coefficients

```
cv.ridge$lambda.min

## [1] 29.68508

head(coef(cv.ridge, s="lambda.min"))

## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 109.4192279
## AtBat        -0.6764771
## Hits          2.5974777
## HmRun        -0.7058689
## Runs          1.8565943
## RBI           0.3434801
```

Parameter Tuning

- `plot(cv.model)` compares the means squared error across λ

```
plot(cv.ridge)
```



- Mean squared error first remains fairly constant and then rises sharply

Ridge Regression in R

- `predict(model, newx=x, s=lambda)` makes predictions for new data `x` and a specific λ

```
pred.ridge <- predict(cv.ridge, newx=x.test, s="lambda.min")
head(cbind(pred.ridge, y.test))
```

```
##                           1   y.test
## -Alan Ashby        390.1766  475.000
## -Andre Dawson     1094.5741  500.000
## -Andre Thornton    798.5886 1100.000
## -Alan Trammell     893.8298  517.143
## -Barry Bonds       518.9105  100.000
## -Bob Dernier       353.4100  708.333
```

- Mean absolute percentage error (MAPE)

```
mean(abs((y.test - pred.ridge)/y.test))
```

```
## [1] 0.6811053
```

Outline

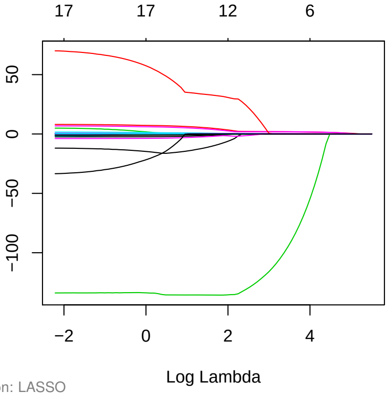
LASSO in R

- Implemented in `glmnet(x, y, alpha=1)` as part of the `glmnet` package

```
lm.lasso <- glmnet(x.train, y.train, alpha=1)
```

  Note: different value for `alpha`
- `plot(...)` shows how λ changes the estimated coefficients

```
plot(lm.lasso, xvar="lambda")
```



Parameter Tuning

- `cv.glmnet(x, y, alpha=1)` determines optimal λ via cross validation by minimizing the mean squared error from a prediction

```
set.seed(0)
cv.lasso <- cv.glmnet(x.train, y.train, alpha=1)
```

- Optimal λ and corresponding coefficients ("." are removed variables)

```
cv.lasso$lambda.min

## [1] 2.143503

head(coef(cv.lasso, s="lambda.min"))

## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 189.7212235
## AtBat        -1.9921887
## Hits          6.6124279
## HmRun         0.6674432
## Runs          .
## RBI           .
```

Parameter Tuning

- Total variables

```
nrow(coef(cv.lasso))

## [1] 20
```

- Omitted variables

```
dimnames(coef(cv.lasso, s="lambda.min"))[[1]][which(
  coef(cv.lasso, s="lambda.min") == 0)]

## [1] "Runs"   "RBI"    "CAtBat" "CHits"
```

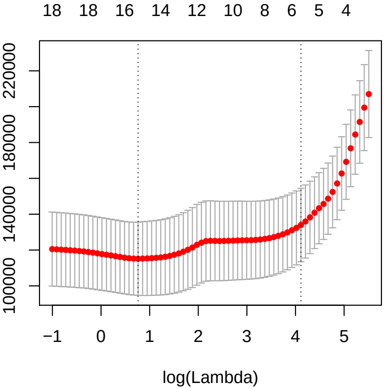
- Included variables

```
dimnames(coef(cv.lasso, s="lambda.min"))[[1]][which(
  coef(cv.lasso, s="lambda.min") != 0)]

##  [1] "(Intercept)" "AtBat"       "Hits"        "HmRun"
##  [6] "Years"       "CHmRun"      "CRuns"       "CRBI"
## [11] "LeagueN"     "DivisionW"   "PutOuts"     "Assists"
## [16] "NewLeagueN"
```

Parameter Tuning

- `plot(cv.model)` compares the means squared error across λ

```r
plot(cv.lasso)
```



- Mean squared error first remains fairly constant and then rises sharply
- Top axis denotes the number of included model variables

LASSO in R

- `predict(model, newx=x, s=lambda)` makes predictions for new data `x` and a specific λ

```
pred.lasso <- predict(cv.lasso, newx=x.test, s="lambda.min")
```

- Mean absolute percentage error (MAPE) of LASSO

```
mean(abs((y.test - pred.lasso)/y.test))

## [1] 0.6328225
```

- For comparison, error of ridge regression

```
## [1] 0.6811053
```

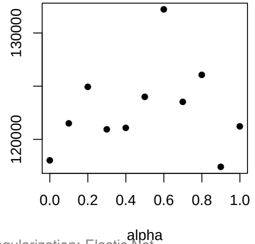
Outline

Elastic Net in R

Example

- Test the elastic net with a sequence of values for α
- Report in-sample mean squared error

```
set.seed(0)
alpha <- seq(from=0, to=1, by=0.1)
en <- lapply(alpha, function(a)
        cv.glmnet(x.train, y.train, alpha=a))
en.mse <- unlist(lapply(en, function(i)
            i$cvm[which(i$lambda==i$lambda.min)]))
plot(alpha, en.mse, ylab="Mean Squared Error", pch=16)
```

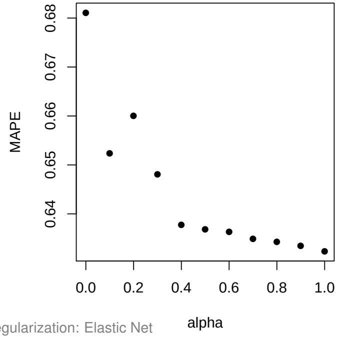


Elastic Net in R

Example (continued)

- Report out-of-sample mean absolute prediction error

```
en.mape <- unlist(lapply(en, function(i) {
             pred <- predict(i, newx=x.test,
                                 s="lambda.min")
             mean(abs((y.test - pred)/y.test))
           }))
plot(alpha, en.mape, ylab="MAPE", pch=16)
```



Outline

Further Readings

Package glmnet

- glmnet tutorial: `http://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html`
- glmnet webinar: `http://web.stanford.edu/~hastie/TALKS/glmnet_webinar.pdf`
  $\rightarrow$ see Hastie's website for data and scripts