

Lab 10

Daniel Tshiani

2025-06-08

a

```
library(MASS)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.4      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
data("Boston")
cor(Boston)
```

```
##           crim      zn      indus      chas      nox
## crim      1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
## zn        -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
## indus      0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
## chas      -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
## nox       0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
## rm        -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
## age       0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
## dis       -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
## rad       0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
## tax       0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
## ptratio   0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
## black     -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064
## lstat     0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
## medv     -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
##           rm      age      dis      rad      tax      ptratio
## crim     -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431  0.28994556
## zn        0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332 -0.39167855
## indus     -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018  0.38324756
## chas      0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652 -0.12151512
## nox     -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320  0.1889327
## rm        1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783 -0.3555015
## age     -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559  0.2615150
```

```
## dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158 -0.2324705
## rad      -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819  0.4647412
## tax      -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000  0.4608530
## ptratio -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304  1.0000000
## black    0.12806864 -0.27353398  0.29151167 -0.444412816 -0.44180801 -0.1773833
## lstat    -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341  0.3740443
## medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593 -0.5077867
##          black      lstat      medv
## crim    -0.38506394  0.4556215 -0.3883046
## zn       0.17552032 -0.4129946  0.3604453
## indus    -0.35697654  0.6037997 -0.4837252
## chas     0.04878848 -0.0539293  0.1752602
## nox      -0.38005064  0.5908789 -0.4273208
## rm       0.12806864 -0.6138083  0.6953599
## age      -0.27353398  0.6023385 -0.3769546
## dis      0.29151167 -0.4969958  0.2499287
## rad      -0.44441282  0.4886763 -0.3816262
## tax      -0.44180801  0.5439934 -0.4685359
## ptratio -0.17738330  0.3740443 -0.5077867
## black    1.00000000 -0.3660869  0.3334608
## lstat    -0.36608690  1.0000000 -0.7376627
## medv     0.33346082 -0.7376627  1.0000000
```

b

```
lm.ridge(medv ~., data = Boston)$coef |> round(2)
```

```
##      crim      zn      indus      chas      nox      rm      age      dis      rad      tax
##    -0.93     1.08     0.14     0.68    -2.06     2.67     0.02    -3.10     2.66    -2.08
## ptratio  black  lstat
##    -2.06     0.85    -3.74
```

c

```
lm.ridge(medv ~., lambda = 3, data = Boston)$coef |> round(2)
```

```
##      crim      zn      indus      chas      nox      rm      age      dis      rad      tax
##    -0.90     1.04     0.07     0.69    -1.98     2.70     0.00    -3.03     2.46    -1.89
## ptratio  black  lstat
##    -2.04     0.85    -3.71
```

```
lm.ridge(medv ~., lambda = 10, data = Boston)$coef |> round(2)
```

```
##      crim      zn      indus      chas      nox      rm      age      dis      rad      tax
##    -0.86     0.95    -0.04     0.71    -1.81     2.74    -0.03    -2.86     2.10    -1.57
## ptratio  black  lstat
##    -1.99     0.84    -3.62
```

```
lm.ridge(medv ~., lambda = 100, data = Boston)$coef |> round(2)
```

```
##      crim      zn      indus      chas      nox      rm      age      dis      rad      tax
##    -0.65     0.58    -0.40     0.74    -0.93     2.78    -0.17    -1.69     0.70    -0.61
## ptratio  black  lstat
##    -1.66     0.78    -2.96
```

as lambda increases, the slopes gravitate towards zero

d

```
rr <- lm.ridge(medv ~., lambda = seq(0, 100, 1), data = Boston)
MASS::select(rr)

## modified HKB estimator is 4.594163
## modified L-W estimator is 3.961575
## smallest value of GCV at 4
the best lambda to choose is 4
```

e

```
library(glmnet)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack
## Loaded glmnet 4.1-9
reg <- lm(medv ~ ., data = Boston)
x <- model.matrix(reg)
dim(x)

## [1] 506 14
x <- x[, -1]
dim(x)

## [1] 506 13
y <- Boston$medv
rr <- glmnet(x,y, alpha=0)
```

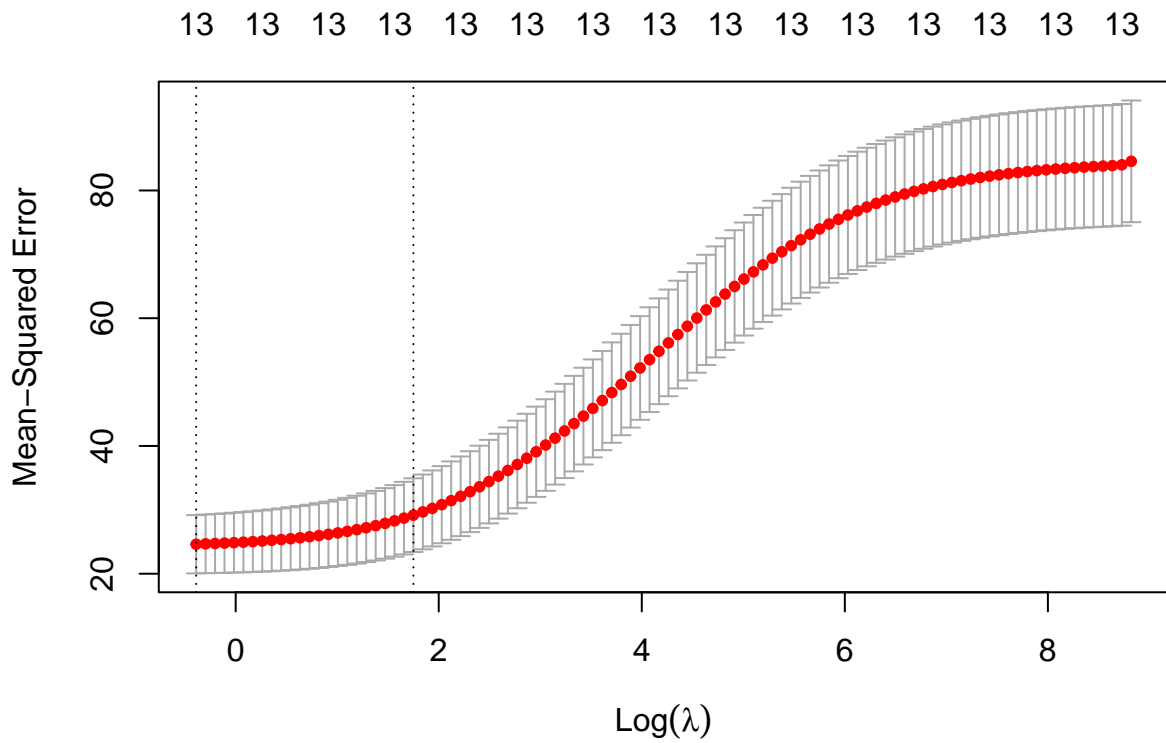
f

```
set.seed(123)
rr_cv <- cv.glmnet(x,y, alpha =0)
rr_cv

##
## Call: cv.glmnet(x = x, y = y, alpha = 0)
##
## Measure: Mean-Squared Error
##
##   Lambda Index Measure      SE Nonzero
```

```
## min 0.678 100 24.61 4.561 13
## 1se 5.759 77 29.17 5.767 13
```

```
plot(rr_cv)
```



lambda min is 0.678 so that is the best lambda to choose.

```
gg
```

```
coef(rr_cv, s = "lambda.min") |> round(2)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##           s0
## (Intercept) 28.00
## crim       -0.09
## zn          0.03
## indus      -0.04
## chas        2.90
## nox       -11.91
## rm          4.01
## age         0.00
## dis        -1.12
## rad         0.15
## tax        -0.01
## ptratio    -0.85
## black       0.01
## lstat      -0.47
```

the lambda that min MSE is 0.678

h

```
lr <- glmnet(x, y, alpha = 1)
lr
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1)
##
##      Df  %Dev Lambda
## 1     0   0.00 6.7780
## 2     1   9.24 6.1760
## 3     2  17.38 5.6270
## 4     2  25.27 5.1270
## 5     2  31.82 4.6720
## 6     2  37.26 4.2570
## 7     2  41.78 3.8780
## 8     2  45.52 3.5340
## 9     2  48.64 3.2200
## 10    3  51.56 2.9340
## 11    3  54.33 2.6730
## 12    3  56.62 2.4360
## 13    3  58.53 2.2190
## 14    3  60.12 2.0220
## 15    3  61.43 1.8430
## 16    3  62.52 1.6790
## 17    3  63.43 1.5300
## 18    3  64.18 1.3940
## 19    3  64.81 1.2700
## 20    4  65.43 1.1570
## 21    4  66.00 1.0540
## 22    5  66.54 0.9607
## 23    5  67.06 0.8754
## 24    5  67.49 0.7976
## 25    5  67.85 0.7267
## 26    6  68.15 0.6622
## 27    6  68.41 0.6034
## 28    7  68.75 0.5498
## 29    7  69.13 0.5009
## 30    8  69.56 0.4564
## 31    8  70.03 0.4159
## 32    8  70.42 0.3789
## 33    8  70.75 0.3453
## 34    9  71.06 0.3146
## 35    9  71.37 0.2866
## 36    9  71.63 0.2612
## 37    9  71.84 0.2380
## 38    9  72.02 0.2168
## 39   11  72.19 0.1976
## 40   11  72.42 0.1800
## 41   12  72.62 0.1640
## 42   12  72.86 0.1495
## 43   12  73.07 0.1362
## 44   12  73.24 0.1241
## 45   12  73.38 0.1131
```

```
## 46 12 73.49 0.1030
## 47 11 73.59 0.0939
## 48 11 73.67 0.0855
## 49 11 73.73 0.0779
## 50 11 73.79 0.0710
## 51 11 73.83 0.0647
## 52 11 73.87 0.0590
## 53 11 73.90 0.0537
## 54 11 73.93 0.0489
## 55 11 73.95 0.0446
## 56 11 73.97 0.0406
## 57 11 73.98 0.0370
## 58 11 74.00 0.0337
## 59 11 74.01 0.0307
## 60 11 74.01 0.0280
## 61 11 74.02 0.0255
## 62 11 74.03 0.0232
## 63 11 74.03 0.0212
## 64 11 74.04 0.0193
## 65 11 74.04 0.0176
## 66 11 74.04 0.0160
## 67 12 74.05 0.0146
## 68 12 74.05 0.0133
## 69 12 74.05 0.0121
## 70 12 74.05 0.0111
## 71 12 74.06 0.0101
## 72 12 74.06 0.0092
## 73 12 74.06 0.0084
## 74 12 74.06 0.0076
## 75 12 74.06 0.0069
## 76 12 74.06 0.0063
```

i

```
set.seed(123)
lr_cv <- cv.glmnet(x, y, nfolds = 10)
lr_cv

##
## Call:  cv.glmnet(x = x, y = y, nfolds = 10)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.0255     61   24.28 4.245         11
## 1se 0.5009     29   28.38 5.002          7

coef(lr_cv, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"
##
##              s0
## (Intercept) 34.594713536
## crim      -0.099226869
## zn         0.041830020
```

```
## indus      .
## chas      2.688250324
## nox      -16.401122005
## rm       3.861229964
## age      .
## dis     -1.404571750
## rad      0.256788020
## tax     -0.009997514
## ptratio  -0.931437290
## black    0.009049252
## lstat    -0.522505968
```

j

```
set.seed(123)
n <- nrow(x)
train_index <- sample(1:n, n * 0.5)
test_index <- setdiff(1:n, train_index)

x_train <- x[train_index, ]
y_train <- y[train_index]
x_test <- x[test_index, ]
y_test <- y[test_index]

lasso_cv <- cv.glmnet(x_train, y_train, alpha = 1)
best_lambda <- lasso_cv$lambda.min

y_pred <- predict(lasso_cv, s = best_lambda, newx = x_test)

test_mse <- mean((y_test - y_pred)^2)
test_mse

## [1] 24.5819
```