LASSO

Prof Wells

STA 295: Stat Learning

March 14th, 2024

Outline

- Discuss LASSO as a method of penalized regression AND variable selection
- Implement LASSO in R

Section 1

The LASSO

Metrics on R^p

How can we measure the distance of a point $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ from the origin?

Metrics on R^p

How can we measure the distance of a point $x = (x_1, \dots, x_p) \in \mathbb{R}^p$ from the origin?

- A natural measurement is the Euclidean distance, which is the square root of the sum of squared coordinates. This is called the ℓ_2 norm:

$$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_p^2} = \sqrt{\sum_{i=1}^{p} x_i^2}$$

Metrics on R^p

How can we measure the distance of a point $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ from the origin?

- A natural measurement is the Euclidean distance, which is the square root of the sum of squared coordinates. This is called the ℓ_2 norm:

$$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_p^2} = \sqrt{\sum_{i=1}^{p} x_i^2}$$

- An alternative measurement is the Manhattan distance, which is the sum of absolute values of coordinates. This is called the ℓ_1 norm:

$$\|x\|_1 = |x_1| + \cdots + |x_p| = \sum_{i=1}^{p} |x_i|$$

Metrics on R^p

How can we measure the distance of a point $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ from the origin?

- A natural measurement is the Euclidean distance, which is the square root of the sum of squared coordinates. This is called the ℓ_2 norm:

$$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_p^2} = \sqrt{\sum_{i=1}^{p} x_i^2}$$

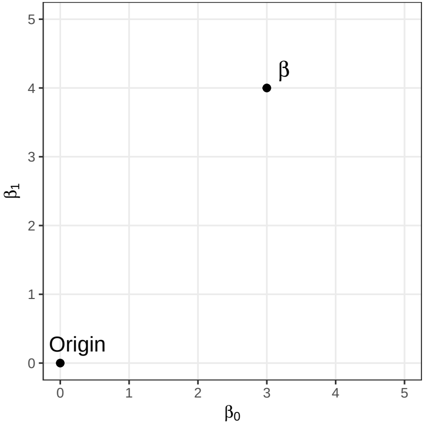
- An alternative measurement is the Manhattan distance, which is the sum of absolute values of coordinates. This is called the ℓ_1 norm:

$$\|x\|_1 = |x_1| + \cdots + |x_p| = \sum_{i=1}^{p} |x_i|$$

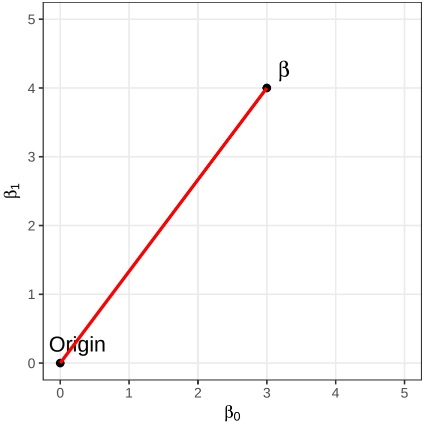
- Occasionally, it might be useful to consider the ℓ_0 "norm" and ℓ_∞ norm

$$\|x\|_0 = \#(x_i \neq 0) \qquad \|x\|_\infty = \max |x_i|$$

Geometric Perspective

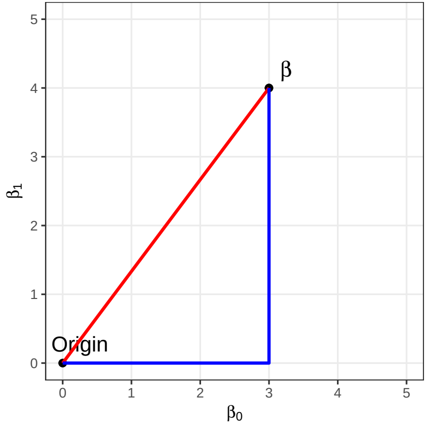


Geometric Perspective



Euclidean Distance
$\|\beta\|_2 = \sqrt{3^2 + 4^2} = 5$

Geometric Perspective



Euclidean Distance
$\|\beta\|_2 = \sqrt{3^2 + 4^2} = 5$

Manhattan Distance
$\|\beta\|_1 = 3 + 4 = 7$

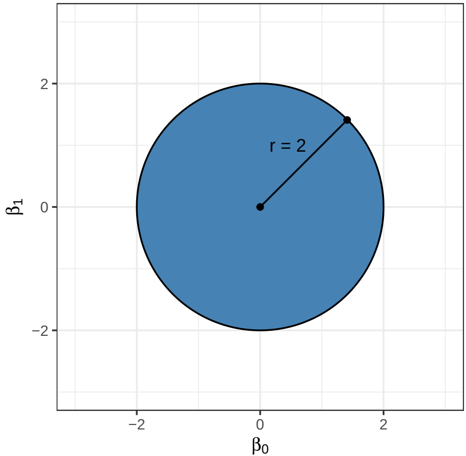
Geometric Perspective II

- What does a circle of radius r look like in the ℓ_2 norm?

Geometric Perspective II

- What does a circle of radius r look like in the ℓ_2 norm?

$$\|\beta\|_2 = \sqrt{\beta_0^2 + \beta_1^2} \leq r$$



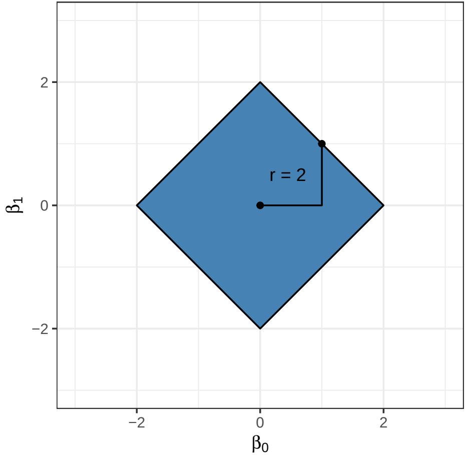
Geometric Perspective II

- What does a "circle" of radius r look like in the ℓ_1 norm?

Geometric Perspective II

- What does a "circle" of radius r look like in the ℓ_1 norm?

$$\|\beta\|_1 = |\beta_0| + |\beta_1| \leq r$$



LASSO

In ridge regression, we seek parameters β that minimize RSS plus the ℓ_2 norm of β:

$$\mathrm{RSS} + \lambda \sum_{i=1}^{p} \beta_i^2 = \mathrm{RSS} + \lambda \|\beta\|_2$$

LASSO

In ridge regression, we seek parameters β that minimize RSS plus the ℓ_2 norm of β:

$$\mathrm{RSS} + \lambda \sum_{i=1}^{p} \beta_i^2 = \mathrm{RSS} + \lambda \|\beta\|_2$$

Alternatively, we could seek parameters β that minimize RSS plus the ℓ_1 norm of β:

$$\mathrm{RSS} + \lambda \sum_{i=1}^{p} |\beta_i| = \mathrm{RSS} + \lambda \|\beta\|_1$$

This latter method is called the LASSO (least absolute shrinkage and selection operator)

LASSO

In ridge regression, we seek parameters β that minimize RSS plus the ℓ_2 norm of β:

$$\mathrm{RSS} + \lambda \sum_{i=1}^{p} \beta_i^2 = \mathrm{RSS} + \lambda \|\beta\|_2$$

Alternatively, we could seek parameters β that minimize RSS plus the ℓ_1 norm of β:

$$\mathrm{RSS} + \lambda \sum_{i=1}^{p} |\beta_i| = \mathrm{RSS} + \lambda \|\beta\|_1$$

This latter method is called the LASSO (least absolute shrinkage and selection operator)

- In addition to shrinking coefficients, it also happens to perform variable selection!

Alternative Formulations

Instead of thinking of Ridge Regression and LASSO as minimizing the sum of RSS and the shrinkage penalty, we can think of them as solving a restricted optimization problem:

Alternative Formulations

Instead of thinking of Ridge Regression and LASSO as minimizing the sum of RSS and the shrinkage penalty, we can think of them as solving a restricted optimization problem:

- For each $s \geq 0$, Ridge Regression seeks to minimize RSS subject to $\|\beta\|_2 \leq s$
- For each $s \geq 0$, LASSO seeks to minimize RSS subject to $\|\beta\|_1 \leq s$

Alternative Formulations

Instead of thinking of Ridge Regression and LASSO as minimizing the sum of RSS and the shrinkage penalty, we can think of them as solving a restricted optimization problem:

- For each $s \geq 0$, Ridge Regression seeks to minimize RSS subject to $\|\beta\|_2 \leq s$
- For each $s \geq 0$, LASSO seeks to minimize RSS subject to $\|\beta\|_1 \leq s$

The best subset algorithm also fits in this paradigm:

- For each $s \geq 0$, best s-subset seeks to minimize RSS subject to $\|\beta\|_0 \leq s$

Alternative Formulations

Instead of thinking of Ridge Regression and LASSO as minimizing the sum of RSS and the shrinkage penalty, we can think of them as solving a restricted optimization problem:

- For each $s \geq 0$, Ridge Regression seeks to minimize RSS subject to $\|\beta\|_2 \leq s$
- For each $s \geq 0$, LASSO seeks to minimize RSS subject to $\|\beta\|_1 \leq s$

The best subset algorithm also fits in this paradigm:

- For each $s \geq 0$, best s-subset seeks to minimize RSS subject to $\|\beta\|_0 \leq s$

Suppose q is 0, 1, or 2. For each $\lambda \geq 0$, there is exactly one $s \geq 0$ so that if β minimizes

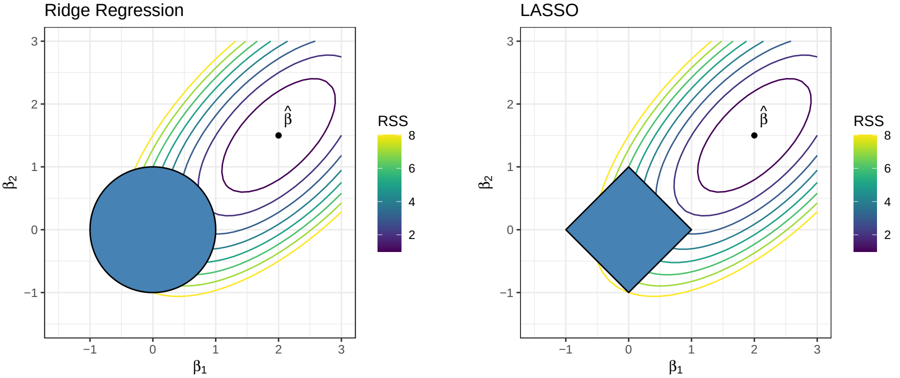
$$\mathrm{RSS} + \lambda\|\beta\|_q$$

then β minimizes

$$\mathrm{RSS} \qquad \text{subject to } \|\beta\|_q \leq s$$

Variable Selection with LASSO

For LASSO, the solution to the optimization problem often lies on a vertex of the domain, which corresponds to a subspace where one or more parameters are 0.



Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)

Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)
  - Can be fit in about the same amount of time as ordinary least squares

Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)
  - Can be fit in about the same amount of time as ordinary least squares
  - Trade slightly increased bias for greatly reduced variance, compared to the full model.

Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)
  - Can be fit in about the same amount of time as ordinary least squares
  - Trade slightly increased bias for greatly reduced variance, compared to the full model.
- **Differences**
  - LASSO performs variable selection in addition to coefficient shrinkage

Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)
  - Can be fit in about the same amount of time as ordinary least squares
  - Trade slightly increased bias for greatly reduced variance, compared to the full model.
- **Differences**
  - LASSO performs variable selection in addition to coefficient shrinkage
  - In Ridge Regression, correlated predictors tend to have similar coefficients. The same is not true of LASSO.

Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)
  - Can be fit in about the same amount of time as ordinary least squares
  - Trade slightly increased bias for greatly reduced variance, compared to the full model.
- **Differences**
  - LASSO performs variable selection in addition to coefficient shrinkage
  - In Ridge Regression, correlated predictors tend to have similar coefficients. The same is not true of LASSO.
  - In general, LASSO tends to outperform Ridge Regression in cases where some of the coefficients are nearly or truly 0.

Comparison of Penalized Regression Models

- **Similarities**
  - Can be implemented in R using `glmnet`. (Ridge regression uses `alpha = 0`, while LASSO uses `alpha = 1`)
  - Can be fit in about the same amount of time as ordinary least squares
  - Trade slightly increased bias for greatly reduced variance, compared to the full model.
- **Differences**
  - LASSO performs variable selection in addition to coefficient shrinkage
  - In Ridge Regression, correlated predictors tend to have similar coefficients. The same is not true of LASSO.
  - In general, LASSO tends to outperform Ridge Regression in cases where some of the coefficients are nearly or truly 0.
  - Ridge Regression outperforms LASSO when all coefficients are significant (but variance is still a liability for MSE)

Section 2

LASSO in R

Solubility

The `solubility` data set from the `AppliedPredictiveModeling` package contains solubility and chemical structure for a sample of 1,267 different compounds.

Solubility

The `solubility` data set from the `AppliedPredictiveModeling` package contains solubility and chemical structure for a sample of 1,267 different compounds.

- For this demonstration, we'll work with just a subset of 30% of the available observations.

Solubility

The `solubility` data set from the `AppliedPredictiveModeling` package contains solubility and chemical structure for a sample of 1,267 different compounds.

- For this demonstration, we'll work with just a subset of 30% of the available observations.
- This subsetted data has been split into a training set `solTrain` and a testing set `solTest`.

Solubility

The `solubility` data set from the `AppliedPredictiveModeling` package contains solubility and chemical structure for a sample of 1,267 different compounds.

- For this demonstration, we'll work with just a subset of 30% of the available observations.
- This subsetted data has been split into a training set `solTrain` and a testing set `solTest`.

Solubility

The `solubility` data set from the `AppliedPredictiveModeling` package contains solubility and chemical structure for a sample of 1,267 different compounds.

- For this demonstration, we'll work with just a subset of 30% of the available observations.
- This subsetted data has been split into a training set `solTrain` and a testing set `solTest`.

```
nrow(solTrain)
```

```
## [1] 285
```

```
ncol(solTrain)
```

```
## [1] 21
```

```
nrow(solTest)
```

```
## [1] 95
```

```
ncol(solTest)
```

```
## [1] 21
```

LASSO in R

- We build LASSO models using identical code to Ridge Regression:

LASSO in R

- We build LASSO models using identical code to Ridge Regression:

```
library(glmnet)
grid = 10^(seq( -5, 5, length = 100))
x<-model.matrix(Solubility ~., data = solTrain)[,-1]
y<-solTrain$Solubility
lasso_mod <- glmnet(x, y, alpha = 1, lambda = grid)
```

LASSO in R

- We build LASSO models using identical code to Ridge Regression:

```
library(glmnet)
grid = 10^(seq( -5, 5, length = 100))
x<-model.matrix(Solubility ~., data = solTrain)[,-1]
y<-solTrain$Solubility
lasso_mod <- glmnet(x, y, alpha = 1, lambda = grid)
```

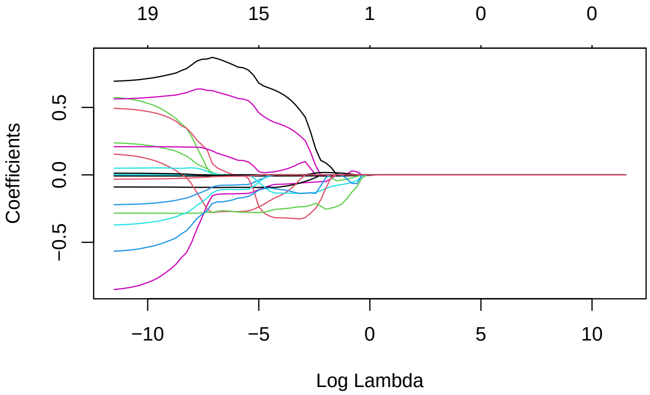
- But note what happens to coefficients:

```
coef(lasso_mod)[1:5,c(1:3,98:100)]
```

```
## 5 x 6 sparse Matrix of class "dgCMatrix"
##                    s0        s1        s2           s97           s98
## (Intercept)  -2.775404 -2.775404 -2.775404  6.393845e-01  6.413927e-01
## MolWeight     .         .         .        -8.100227e-03 -8.100687e-03
## NumAtoms      .         .         .        -5.785492e-04 -6.844627e-04
## NumNonHAtoms  .         .         .         2.340836e-01  2.358484e-01
## NumBonds      .         .         .        -1.342641e-05 -1.501692e-05
##                       s99
## (Intercept)   6.430179e-01
## MolWeight    -8.101076e-03
## NumAtoms     -7.733290e-04
## NumNonHAtoms  2.372857e-01
## NumBonds     -2.094374e-05
```

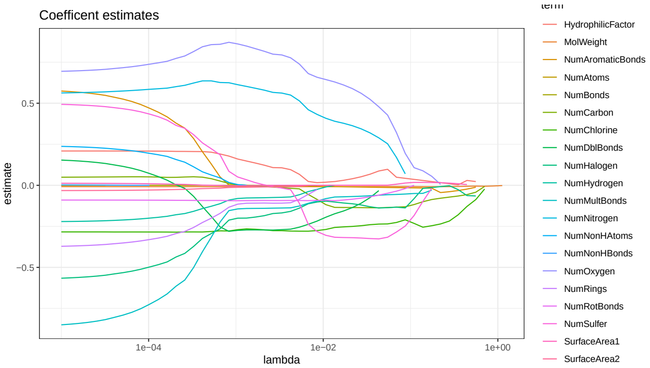
Coefficient Paths

```
plot(lasso_mod, xvar = "lambda")
```



Coefficient Paths

```r
library(broom)
tidied <- tidy(lasso_mod) %>% filter(term != "(Intercept)")
ggplot(tidied, aes(lambda, estimate, group = term, color = term)) +
    geom_line() + scale_x_log10()+ theme_bw()+labs(title = "Coefficent estimates")
```



Cross-Validation

- To find the optimal penalty, we use `cv.glmnet`:

Cross-Validation

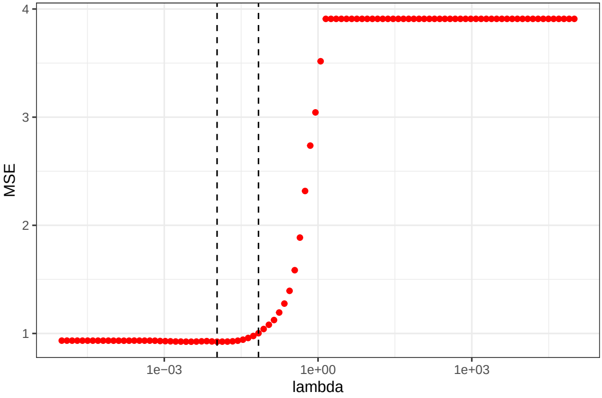
- To find the optimal penalty, we use cv.glmnet:

```
set.seed(1010)
my_cv<-cv.glmnet(x, y, alpha = 1, lambda = grid, nfolds = 10)

best_L <- my_cv$lambda.min
reg_L <- my_cv$lambda.1se

data.frame(best_L, reg_L)
```

```
##   best_L  reg_L
## 1 0.0107 0.0689
```

Cross-validation plot

```
tidied <- tidy(my_cv)
ggplot(tidied, aes(x = lambda, y = estimate))+geom_point( color = "red")+
  scale_x_log10()+theme_bw()+labs(y = "MSE")+
  geom_vline(xintercept = best_L, linetype = "dashed" )+
  geom_vline(xintercept = reg_L, linetype = "dashed")
```



Feature Selection

- What features did the best λ select?

Feature Selection

- What features did the best λ select?

```
s <- which(lasso_mod$lambda==best_L)
s
```

```
## [1] 70
```

```
coef(lasso_mod)[,s]
```

```
##       (Intercept)         MolWeight          NumAtoms       NumNonHAtoms
##           0.03232          -0.00806           0.00000            0.00000
##          NumBonds       NumNonHBonds      NumMultBonds       NumRotBonds
##           0.00000           0.00000          -0.08122           -0.09071
##       NumDblBonds  NumAromaticBonds       NumHydrogen         NumCarbon
##          -0.19428           0.00000          -0.01010           -0.11791
##       NumNitrogen         NumOxygen         NumSulfer       NumChlorine
##           0.40824           0.64413          -0.30461           -0.26894
##       NumHalogen          NumRings HydrophilicFactor      SurfaceArea1
##          -0.09626           0.00000           0.01904            0.00000
##      SurfaceArea2
##           0.00000
```

```
sum(coef(lasso_mod)[,s] !=0 )
```

```
## [1] 13
```

Overall Performance

- Recall that glmnet already fits a model, so we just need to use predict to get predictions:

```
x_tst <- model.matrix(Solubility ~., data = solTest)[,-1]
lasso_preds <- predict(lasso_mod, s = best_L, newx = x_tst)
rmse <- sqrt(mean( (solTest$Solubility - lasso_preds)^2))
rmse
```

```
## [1] 0.852
```

Overall Performance

- Recall that glmnet already fits a model, so we just need to use predict to get predictions:

```
x_tst <- model.matrix(Solubility ~., data = solTest)[,-1]
lasso_preds <- predict(lasso_mod, s = best_L, newx = x_tst)
rmse <- sqrt(mean( (solTest$Solubility - lasso_preds)^2))
rmse
```

```
## [1] 0.852
```

- Let's compare performance for: the full model, ridge regression, LASSO with $\lambda = 0.011$, and LASSO with $\lambda = 0.069$.

Overall Performance

- Recall that glmnet already fits a model, so we just need to use predict to get predictions:

```
x_tst <- model.matrix(Solubility ~., data = solTest)[,-1]
lasso_preds <- predict(lasso_mod, s = best_L, newx = x_tst)
rmse <- sqrt(mean( (solTest$Solubility - lasso_preds)^2))
rmse
```

```
## [1] 0.852
```

- Let's compare performance for: the full model, ridge regression, LASSO with $\lambda = 0.011$, and LASSO with $\lambda = 0.069$.

```
##    full rr_min lasso_min lasso_1se
## 1 0.868  0.859     0.852     0.857
```

- **LASSO wins!**