

Regularized Regression

In a regression framework the goal is to model the relationship between a variable of interest, y , and a set of covariates \mathcal{X} . Using the normal distributional assumptions then the joint distribution of the observed data, given the data x_1, \dots, x_n along with β and σ^2 can be written as:

$$p(y_1, \dots, y_n | \tilde{x}_1, \dots, \tilde{x}_n, \tilde{\beta}, \sigma^2) = \prod_{i=1}^n p(y_i | \tilde{x}_i, \tilde{\beta}, \sigma^2) \quad (1)$$

$$= (2\pi\sigma^2)^{-n/2} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \tilde{\beta}^T \tilde{x}_i)^2 \right]. \quad (2)$$

Note this is the same as the sampling, or generative model, that we have seen earlier in class.

The model is often formulated using matrix expressions and a multivariate normal distribution. Let

In a classical setting, typically least squares methods are used to compute the values of the covariates in a regression setting. Specifically, we seek to minimize the sum of squared residuals (SSR), where $SSR(\tilde{\beta}) = (\tilde{y} - X\tilde{\beta})^T (\tilde{y} - X\tilde{\beta})$.

Thus we will take the derivative of this function with respect to β to minimize this expression.

$$\frac{d}{d\tilde{\beta}} SSR(\tilde{\beta}) = \frac{d}{d\tilde{\beta}} (\tilde{y} - X\tilde{\beta})^T (\tilde{y} - X\tilde{\beta}) \quad (3)$$

$$= \frac{d}{d\tilde{\beta}} (\tilde{y}^T \tilde{y} - 2\tilde{\beta}^T X^T \tilde{y} + \tilde{\beta}^T X^T X \tilde{\beta}) \quad (4)$$

$$= -2X^T \tilde{y} + 2X^T X \tilde{\beta} \quad (5)$$

$$\text{then set } = 0 \quad \text{which implies} \quad X^T X \beta = X^T \tilde{y} \quad (6)$$

$$\text{and} \quad \tilde{\beta} = (X^T X)^{-1} X^T \tilde{y}. \quad (7)$$

This value is the OLS estimate of $\tilde{\beta}_{OLS} = (X^T X)^{-1} X^T \tilde{y}$.

Bayesian Modeling and Regularization Ordinary Least Squares (OLS) regression can be written as:

$$\hat{\beta}_{OLS} = \arg \min_{\hat{\beta}} \|\tilde{y} - X\hat{\beta}\|_2^2 \rightarrow \hat{\beta} = (X^T X)^{-1} X^T \tilde{y},$$

where $\|\tilde{x}\|_p = (|x_1|^p + \dots + |x_m|^p)^{1/p}$ is an LP norm. So the L2 norm is $\|\tilde{x}\|_2 = \sqrt{x_1^2 + \dots + x_m^2}$.

An alternative form of penalized regression is known as Least Absolute Shrinkage and Selection Operator (LASSO). The LASSO uses an L1 penalty such that:

Consider the following prior $p(\tilde{\beta}) = N(0, I_p \tau^2)$. How does this relate to ridge regression? First compute the posterior distribution for $\tilde{\beta}$.

$$\begin{aligned} p(\tilde{\beta}|-) &\propto \exp \left[-\frac{1}{2} \left(\frac{1}{\sigma^2} \tilde{\beta}^T X^T X \tilde{\beta} - \frac{1}{\sigma^2} \tilde{\beta}^T X^T \tilde{y} + \tilde{\beta}^T \frac{I_p}{\tau^2} \tilde{\beta} \right) \right] \\ &\propto \exp \left[-\frac{1}{2} \left(\frac{1}{\sigma^2} \tilde{\beta}^T \left(X^T X + \frac{\sigma^2}{\tau^2} I_p \right) \tilde{\beta} - \frac{1}{\sigma^2} \tilde{\beta}^T X^T \tilde{y} \right) \right] \end{aligned}$$

Simulate Regression Data

```
n <- 50
p <- 10

X <- cbind(matrix(runif(n*p,min = -1, max = 1), n, p))
beta <- rep(0, p )
beta[c(1,2)] <- c(3, 3)

y <- rnorm(n, mean = X %*% beta, sd = 3)
y_center <- y - mean(y)
summary(lm(y_center~X))

##
## Call:
## lm(formula = y_center ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.4111 -1.9258  0.0543  1.6718  9.0089
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3999     0.5130  -0.780  0.440360
## X1             2.3536     0.8893   2.646  0.011669 *
## X2             3.8748     0.9215   4.205  0.000148 ***
## X3            -0.3988     1.0088  -0.395  0.694780
## X4             0.3629     0.8899   0.408  0.685608
## X5            -0.2413     0.8819  -0.274  0.785876
## X6            -1.2707     0.9899  -1.284  0.206873
## X7             0.8479     0.8304   1.021  0.313485
## X8            -0.7527     1.0737  -0.701  0.487448
## X9            -0.5595     1.0293  -0.544  0.589854
## X10           -0.6095     0.9993  -0.610  0.545485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.401 on 39 degrees of freedom
## Multiple R-squared:  0.4936, Adjusted R-squared:  0.3638
## F-statistic: 3.801 on 10 and 39 DF,  p-value: 0.001227
```

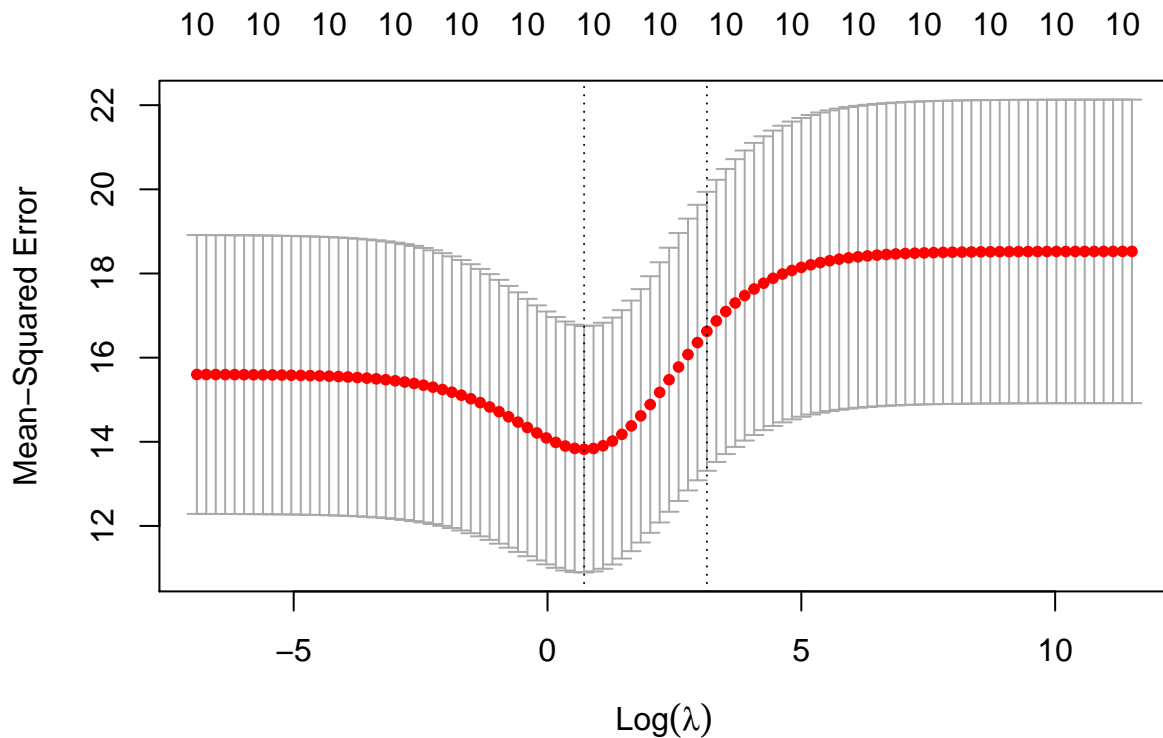
Ridge Regression

```
ridge_fit <- glmnet(X, y_center, alpha = 0, lambda = 0)
t(coef(ridge_fit))

## 1 x 11 sparse Matrix of class "dgCMatrix"
##      [[ suppressing 11 column names '(Intercept)', 'V1', 'V2' ... ]]
##
## s0 -0.3998859 2.354099 3.875362 -0.3979093 0.3623919 -0.2414221 -1.271342
##
## s0 0.8477595 -0.7520009 -0.5592078 -0.6091662
```

```
t(coef(glmnet(X, y_center, alpha = 0, lambda = 1e6)))

## 1 x 11 sparse Matrix of class "dgCMatrix"
##      [[ suppressing 11 column names '(Intercept)', 'V1', 'V2' ... ]]
##
## s0 -1.510178e-06 1.101924e-05 1.722992e-05 -7.840096e-06 2.256134e-06
##
## s0 -2.294676e-06 -4.885313e-06 1.179919e-06 -3.950416e-06 -1.836847e-06
##
## s0 -3.01518e-06
ridge_fit_cv <- cv.glmnet(X, y_center, alpha = 0)
lambdas_seq <- 10^seq(-3, 5, length.out = 100)
ridge_cv <- cv.glmnet(X, y_center, alpha = 0, lambda = lambdas_seq)
plot(ridge_cv)
```



```
print(ridge_cv)

##
## Call:  cv.glmnet(x = X, y = y_center, lambda = lambdas_seq, alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min  2.057    59   13.82  2.928        10
## 1se 23.101    46   16.63  3.315        10
```

```

ridge_fit <- glmnet(X, y_center, alpha = 0, lambda = ridge_cv$lambda.min)
t(coef(ridge_fit))

## 1 x 11 sparse Matrix of class "dgCMatrix"
##      [[ suppressing 11 column names '(Intercept)', 'V1', 'V2' ... ]]
##
## s0 -0.2631086 1.562949 2.567943 -0.6287719 0.3019515 -0.1498433 -0.7759721
##
## s0 0.462363 -0.6471148 -0.3728943 -0.4438042

```

Lasso

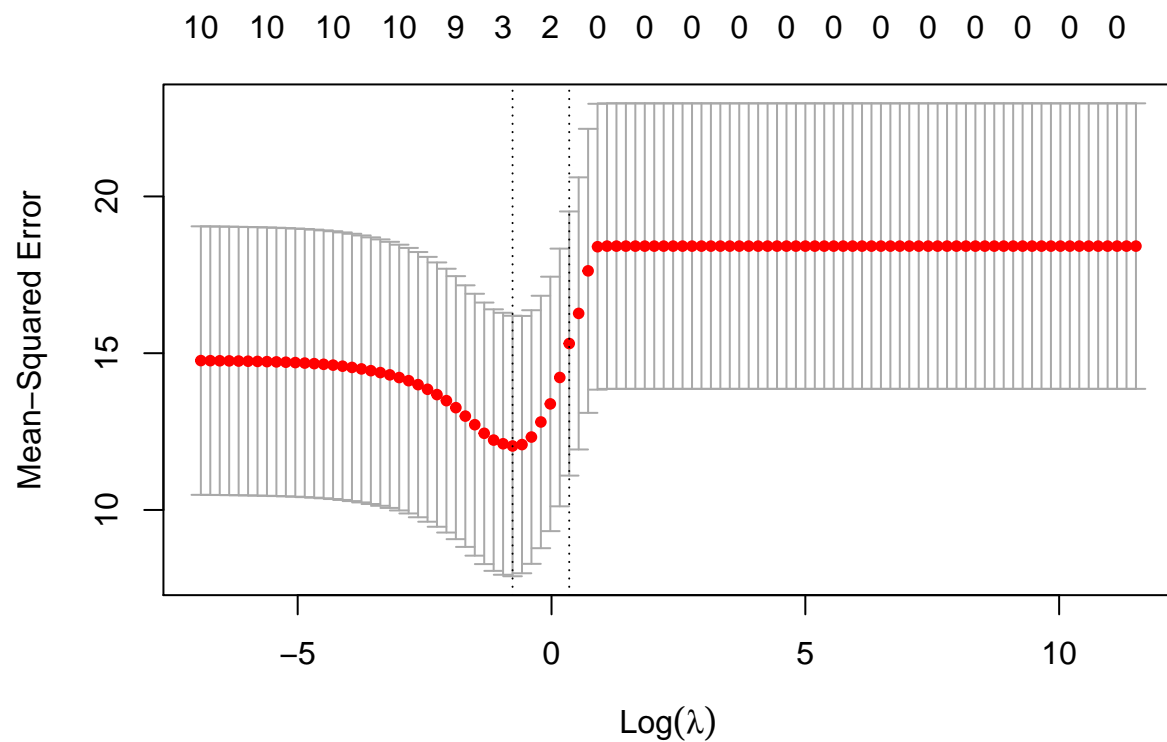
```

t(coef(glmnet(X, y_center, alpha = 0, lambda = 0)))

## 1 x 11 sparse Matrix of class "dgCMatrix"
##      [[ suppressing 11 column names '(Intercept)', 'V1', 'V2' ... ]]
##
## s0 -0.3998859 2.354099 3.875362 -0.3979093 0.3623919 -0.2414221 -1.271342
##
## s0 0.8477595 -0.7520009 -0.5592078 -0.6091662
t(coef(glmnet(X, y_center, alpha = 1, lambda = 1e6)))

## 1 x 11 sparse Matrix of class "dgCMatrix"
##      [[ suppressing 11 column names '(Intercept)', 'V1', 'V2' ... ]]
##
## s0 . 0 . . . . . . . . . .
lasso_cv <- cv.glmnet(X, y_center, alpha = 1, lambda = lambdas_seq)
plot(lasso_cv)

```



```
lasso_fit <- glmnet(X, y_center, alpha = 1, lambda = lasso_cv$lambda.min)
coef(lasso_fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##              s0
## (Intercept) -0.1580537
## V1          1.6410982
## V2          3.1995546
## V3          .
## V4          .
## V5          .
## V6         -0.8606924
## V7          .
## V8          .
## V9          .
## V10         .
```