Regularized Regression

Dave Miller

2/14/2019

Why Regularize?

- The objective in least-squares regression is to minimize the MSE:

$$\text{Cost} = \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 = \frac{1}{n}(Y - X\mathbf{b})^T(Y - X\mathbf{b})$$

- Least squares performs quiet well when the data aligns with the key assumptions, mainly:
  - Linear relationship
  - There are more observations (n) than features (m) ($n > p$)
  - No or little multicollinearity

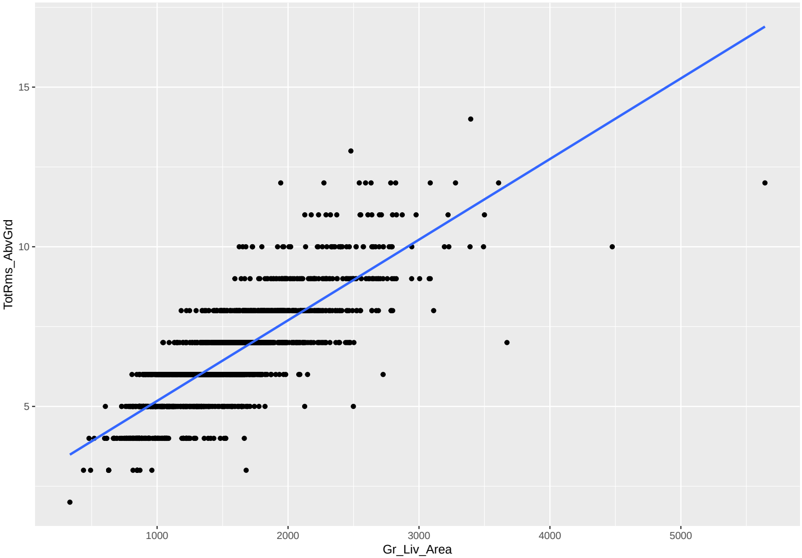
- One feature of overfitting is that the coefficients are large in magnitude. Regularization helps reduce the liklihood of overfitting.

Multicollinearity

- As we gain more features, we are more likely to capture multiple features that are **collinear**, or have a very high correlation to one-another.

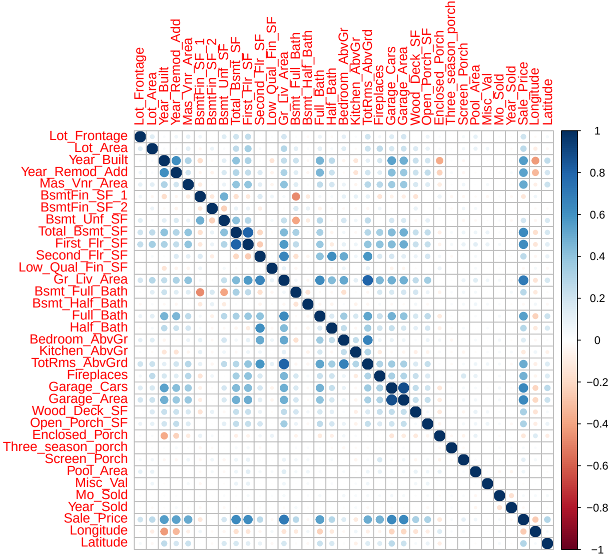
- For example, the size of a house (Gr_Liv_Area) and the number of rooms (TotRms_AbvGrd) are highly correlated at 0.801.

Plot



Correlation plot

- We can view the correlation of all the features in AmesHousing



Linear model example

```r
lm(Sale_Price ~ Gr_Liv_Area, data = ames_train)
```

```
##
## Call:
## lm(formula = Sale_Price ~ Gr_Liv_Area, data = ames_train)
##
## Coefficients:
## (Intercept)  Gr_Liv_Area
##      8659.0        115.4
```

Linear model example

```r
lm(Sale_Price ~ TotRms_AbvGrd, data = ames_train)
```

```
##
## Call:
## lm(formula = Sale_Price ~ TotRms_AbvGrd, data = ames_train)
##
## Coefficients:
##   (Intercept)  TotRms_AbvGrd
##         15194          25846
```

Multivariate linear model example

```r
lm(Sale_Price ~ Gr_Liv_Area + TotRms_AbvGrd, data = ames_trai
```

```
##
## Call:
## lm(formula = Sale_Price ~ Gr_Liv_Area + TotRms_AbvGrd, data
##
## Coefficients:
##   (Intercept)    Gr_Liv_Area  TotRms_AbvGrd
##       37243.8          142.7       -10806.2
```

Multicolinearity in modeling

- Coefficients for correlated features become over-inflated and can fluctuate significantly.
- One consequence of these large fluctuations in the coefficient terms is overfitting.
- Although an analyst can use tools such as variance inflaction factors to identify and remove those strongly correlated variables, it is not always clear which variable(s) to remove.

Regularized regression

- When we experience these concerns, one alternative to regression is to use **regularized regression** (also commonly referred to as *penalized models* or *shrinkage methods*) to control the parameter estimates.

- Regularized regression puts contraints on the magnitude of the coefficients and will progressively shrink them towards zero.

- This constraint helps to reduce the magnitude and fluctuations of the coefficients and will reduce the variance of our model.

Regularized regression

- The objective of regularized regression is given similar to regression, but we add a penalty term P:

  $$\text{Cost} = \text{MSE} + P$$

  where MSE is measuring the fit of the data, and the penalty term P is measuring the **magnitude of the coefficients** $\beta_0, \beta_1, \ldots, \beta_m$.

- L1 norm:

  $$||\mathbf{b}|| = |\beta_0| + |\beta_1| + \cdots + |\beta_m|$$

- L2 norm:

  $$||\mathbf{b}||_2^2 = (\beta_0)^2 + (\beta_1)^2 + \cdots + (\beta_m)^2$$

Standardize the predictor variables

- In regularized regression, it is important that the predictor variables X are on the same scale.

- To do so, we **standardize** or **normalize** each variable via the Z-score standardization:

$$x_{new} = \frac{x - \bar{x}}{\sigma(x)}$$

where $\bar{x}$ is the mean and $\sigma(x)$ is the standard deviation.

- Now, each variable will have mean 0 and sd 1.

Standardize the predictor variables

- Using standardized inputs will change the coefficients in the model:

```
##   (Intercept)   Gr_Liv_Area TotRms_AbvGrd
##    37243.8168      142.6897   -10806.2353
```

```
##           (Intercept)   Gr_Liv_Area_scaled TotRms_AbvGrd_sca
##             181117.72             70118.80            -1665
```

Standardize the predictor variables

- However, after scaling the test set using the training mean / sd, the predictions are unchanged:

```
## # A tibble: 879 x 2
##       pred pred_scaled
##      <dbl>       <dbl>
##  1 197894.     197894.
##  2 219012.     219012.
##  3 197752.     197752.
##  4 141779.     141779.
##  5 170641.     170641.
##  6 208339.     208339.
##  7 117303.     117303.
##  8 224720.     224720.
##  9 200463.     200463.
## 10 126473.     126473.
## # ... with 869 more rows
```

Ridge regression

- Ridge regression controls the coefficients by adding the L2 penalty to the cost function:

$$\text{Cost} = \text{MSE} + \lambda ||\mathbf{b}||_2^2$$

where λ is the **tuning parameter**.

- In vector notation, this is written:

$$\text{Cost} = \frac{1}{n}(Y - X\mathbf{b})^T(Y - X\mathbf{b}) + \lambda \mathbf{b}^{\mathbf{T}}\mathbf{b}$$

Ridge regression

$\text{Cost} = \text{MSE} + \lambda||\mathbf{b}||_2^2$

- If $\lambda = 0$:

  $$\text{Cost} = \text{MSE} + 0 * ||\mathbf{b}||_2^2$$

  is the cost function for ordinary least squares.

- If $\lambda = \infty$ and the coefficients are not 0:

  $$\text{Cost} = \text{MSE} + \infty * ||\mathbf{b}||_2^2 = \infty,$$

  forcing all coefficients to equal 0.

- So we are looking for $0 < \lambda < \infty$.

- **Note:** We don't include the intercept β_0 in ridge regression because a large intercept does not indicate overfitting.

Coefficient plot

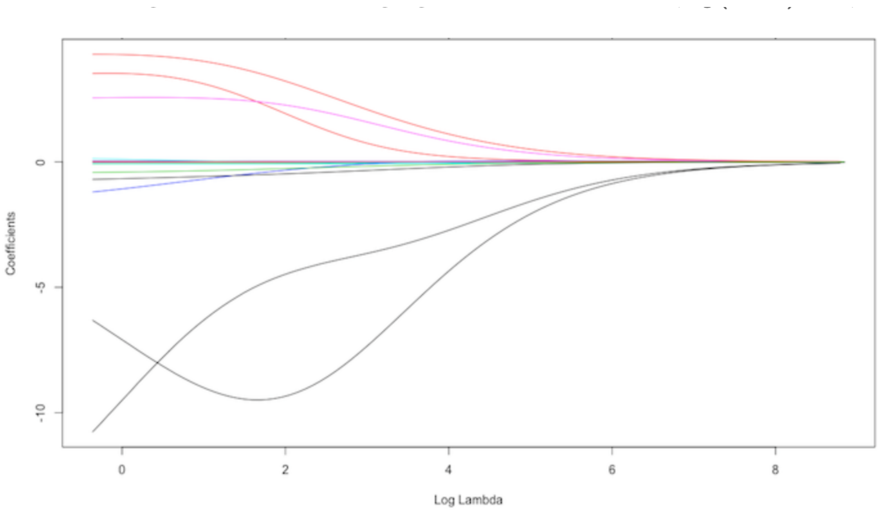


Figure 1:

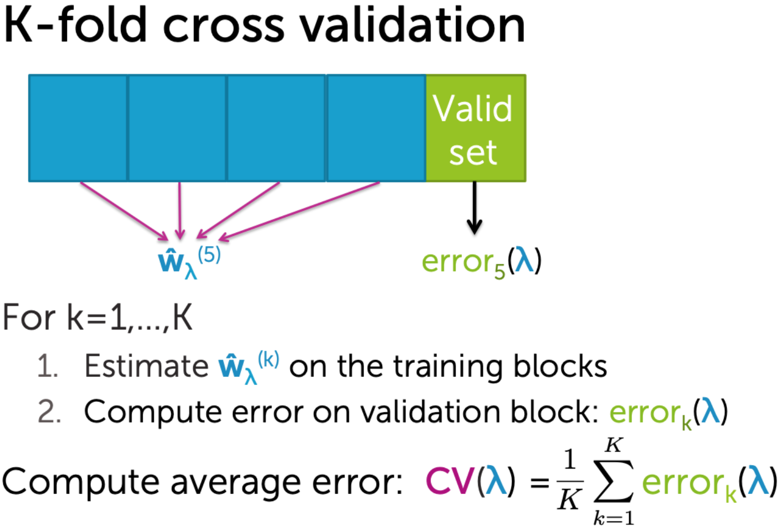
Ridge regression

- In essence, the ridge regression model has pushed many of the correlated features towards each other rather than allowing for one to be wildly positive and the other wildly negative.

- Furthermore, many of the non-important features have been pushed closer to zero. This means we have reduced the noise in our data, which provides us more clarity in identifying the true signals in our model.

How to choose λ

- We use **Cross Validation** to choose λ, using MSE as the cost function.



K-fold cross validation

For k=1,…,K

1. Estimate $\hat{\mathbf{w}}_\lambda^{(k)}$ on the training blocks
2. Compute error on validation block: $\text{error}_k(\lambda)$

Compute average error: $CV(\lambda) = \frac{1}{K}\sum_{k=1}^{K} \text{error}_k(\lambda)$