Regularized Regression

Dave Miller

2/14/2019

Why Regularize?

- Models are much more efficient when we have fewer inputs (m).
- If we have collinear features, it is better for our models if we select a few of them.
- It is not obvious which features to select, so we will do so with **regularization**.

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:

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

Lasso regression

- **Lasso regression** (or L1 regression) is similar to ridge regression, except it uses the L1 penalty:

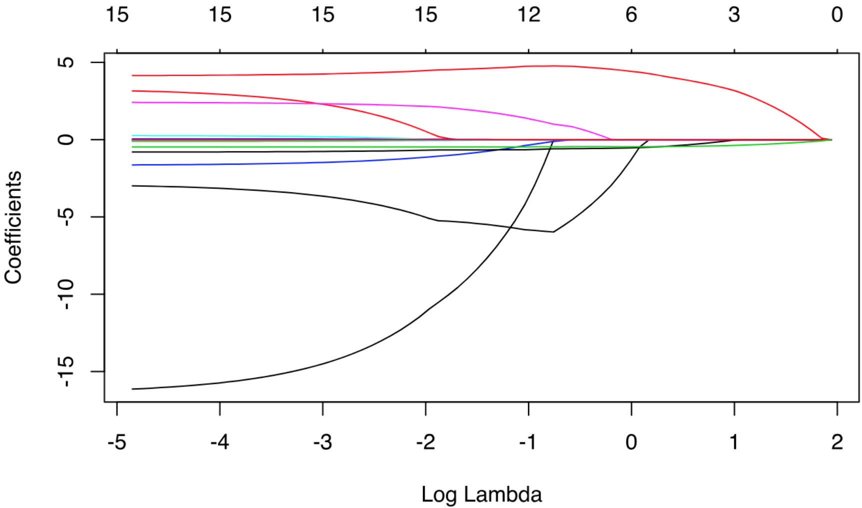
$$
\begin{aligned}
\text{Cost} &= \text{MSE} + \lambda ||\mathbf{b}||_1 \\
&= \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{m} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{m} |\beta_j|
\end{aligned}
$$

Lasso regression

- Whereas the ridge regression approach pushes variables to approximately but not equal to zero, the lasso penalty will actually push coefficients to zero.

- The lasso model not only improves the model with regularization but it also conducts **automated feature selection.**

Coefficient plot



Ridge and Lasso in R using glmnet

To apply a ridge model we can use the `glmnet::glmnet` function. The `alpha` parameter tells `glmnet` to perform a ridge (`alpha = 0`), lasso (`alpha = 1`), or elastic net ($0 \leq alpha \leq 1$) model. Behind the scenes, `glmnet` is doing two things that you should be aware of:

1. It is essential that predictor variables are standardized when performing regularized regression. `glmnet` performs this for you. If you standardize your predictors prior to `glmnet` you can turn this argument off with `standardize = FALSE` .
2. `glmnet` will perform ridge models across a wide range of λ parameters, which are illustrated in the figure below.

Figure 2:

Ridge and Lasso in R using glmnet

Implementing lasso follows the same logic as implementing the ridge model, we just need to switch `alpha = 1` within `glmnet`.

Figure 3:

Ridge and Lasso in R using glmnet

Predicting

Once you have identified your preferred model, you can simply use `predict` to predict the same model on a new data set. The only caveat is you need to supply `predict` an `s` parameter with the preferred models λ value. For example, here we create a lasso model, which provides me a minimum MSE of 0.022. I use the minimum λ value to predict on the unseen test set and obtain a slightly lower MSE of 0.015.

Figure 4: