Machine Learning Crash Course: 2. Regularization

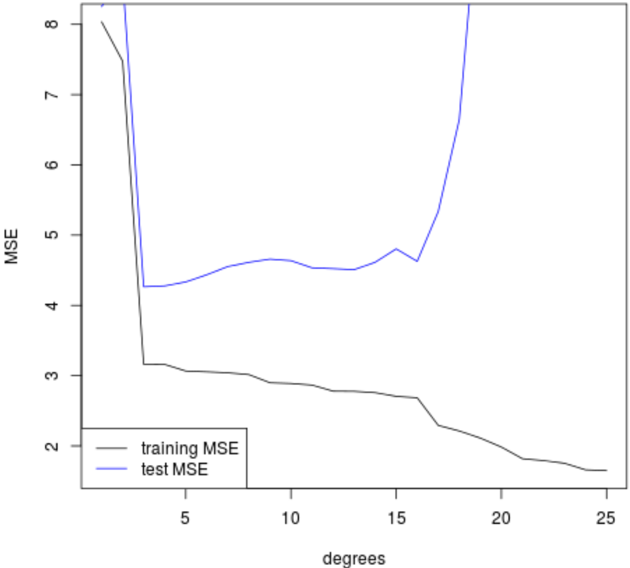
Carlo Zanella

September 11, 2018

This module

We have already encountered the problem of overfitting where more flexible models lead to worse out-of-sample performance. This module introduces the technique of regularization to help deal with that problem.

Overfitting



Overfitting

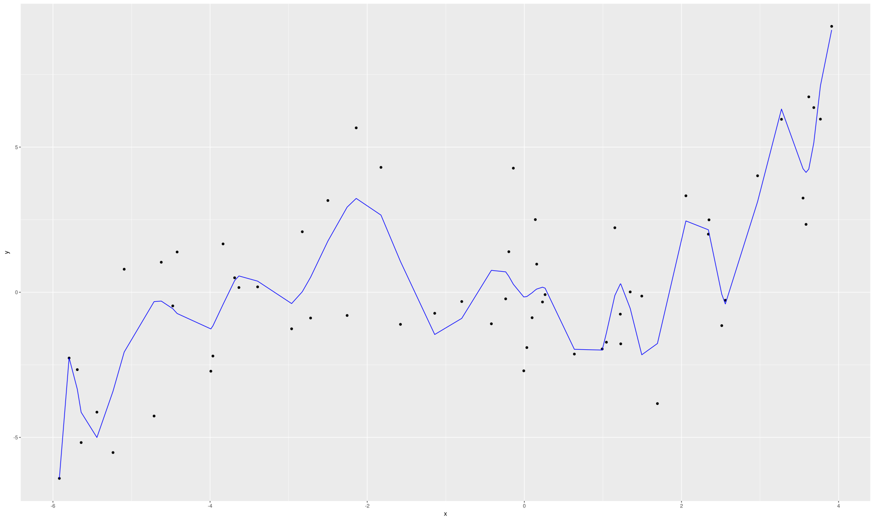


Figure: Overfitting

Overfitting

- Highly flexible models tend to try too much to fit the noise / error term rather than the true relationship.
- This is called overfitting.

Bias-Variance decomposition

- Let us try to better understand the issue of overfitting.
- Suppose we use training data to estimate a $\tilde{f}$ and we want to know the expected MSE on new test data (y_0, x_0)
- Now our estimate of $\tilde{f}$ is stochastic since it is based on a random draw of data.
- Hence we can decompose the test MSE as follows

  $$testMSE = E(Y - \hat{Y})^2 = \mathrm{Var}(\tilde{f}(x)) + [\mathrm{Bias}(\tilde{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

- We want both low bias and low variance

Bias-Variance decomposition

- We want both low bias and low variance
- More flexibility usually means lower bias: we can more closely capture the true relationship
- However, more flexibility usually requires more data: therefore we can estimate it less precisely $\Rightarrow$ higher variance
- In practice: we are willing to incur some bias to decrease variance
- Overfitting: noisy estimation $\Rightarrow$ high variance $\Rightarrow$ high test error

What can we do about it?

Regularization

- Regularization is a very general approach to reduce overfitting.
- Take Linear regression. We choose $\beta_0, \beta_1, \ldots$ to minimize the cost function

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots))^2$$

- Regularization adds a penalty term or regularization term to the cost function that penalizes more complex models:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots))^2 + \lambda \sum_{j=1}^{k} \beta_k^2$$

- Parameter λ determines how much we punish complexity.
- This is also known as Ridge regression.
- ML practitioners just call it regularized or penalized linear regression.

Regularization cont'd

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots))^2 + \lambda \sum_{j=1}^{k} \beta_k^2$$

- In statistics this is called a shrinkage method.
- Coefficients $\tilde{\beta}$ are shrunken to 0.
- This corresponds to simpler less flexible models.

The LASSO

- A very popular kind of regularization is called LASSO.
- LASSO = Least Absolute Shrinkage and Selection Operator
- The cost function is

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots))^2 + \lambda \sum_{j=1}^{k} |\beta_k|$$

- Notice that the objective function is not differentiable, hence there will be corner solutions at $\beta_k = 0$.
- In contrast to Ridge, LASSO likes to set coefficients of relatively unimportant factors exactly to zero.
- Excellent tool for variable selection!

LASSO vs Ridge

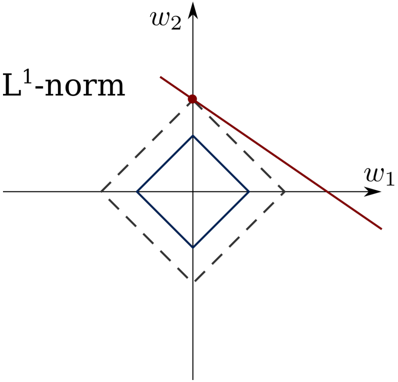


Figure: LASSO vs Ridge constraint regions (Source: Wikipedia)

Elastic Net

- A generalized cost function is

$$\sum_{i=1}^{n} \frac{1}{n} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \ldots))^2 +$$
$$\lambda \left(\alpha \sum_{j=1}^{k} |\beta_k| + (1-\alpha) \sum_{j=1}^{k} \beta_k^2 \right)$$

- Mixes LASSO ($\alpha = 1$) and Ridge ($\alpha = 0$)
- Popular in practice because LASSO alone sometimes "over-regularizes".

Hyperparameters

- What about λ? And α?
- These parameters of LASSO / Ridge / Elastic Net are called hyperparameters or tuning parameters.
- Most methods used in practice have these hyperparameters.
- In order to use these methods we have to specify a value for λ or α.
- How could we do that?
- This is almost always done by using Cross Validation.
- Idea is simple: estimate the model for many different values of λ
- Then choose the λ which yields the lowest CV test error.

(Notebook “03 LASSO, Ridge, and Hyperparameter tuning”)

Putting together Hyperparameter Tuning and Model Assessment

- We use Cross Validation both for Model Assessment and Hyperparameter tuning.
- It is important to be careful when one combines them.
- Hyperparameter tuning itself may cause overfitting
- By the usual logic evaluation of tuned models should be done on a dataset not used for tuning.

Putting together Hyperparameter Tuning and Model Assessment (2)

- Simple way to do it: "Cross-Validation Squared"
- For instance, split data into training set and testing set
- Use CV again on training set to find the best hyperparameter.
- Do that with all your models.
- Then use the test data to compare your models.
- This gives an "outer" and an "inner" cross-validation.
- Of course, you can also use 10-fold CV rather than the validation set approach for the outer CV
- Important: proper and careful model evaluation is often essential to successful prediction