Introduction to Machine Learning

Extending Linear Regression

Varun Chandola

March 8, 2021

Outline

Contents

1 Shortcomings of Linear Models

1. Susceptible to outliers
2. *Too simplistic* - Underfitting
3. No way to control overfitting
4. Unstable in presence of correlated input attributes
5. Gets “confused” by unnecessary attributes

Biggest Issue with Linear Models

- They are linear!!
- Real-world is usually non-linear
- How do learn non-linear fits or non-linear decision boundaries?
  - Basis function expansion
  - Kernel methods (*will discuss this later*)

2 Handling Non-linear Relationships

- Replace $\mathbf{x}$ with non-linear functions $\boldsymbol{\phi}(\mathbf{x})$

$$y = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})$$

- Model is still linear in $\mathbf{w}$
- Also known as **basis function expansion**

Example 1.

$$\boldsymbol{\phi}(x) = [1, x, x^2, \ldots, x^p]$$

- Increasing p results in more complex fits

2.1 Handling Overfitting via Regularization

- Always choose the simpler explanation
- Keep things simple
- *Pluralitas non est ponenda sine neccesitate*
- A general problem-solving philosophy

There are many ways to describe the Occam’s Razor principle. In simple words, if there are two possible explanations for a certain phenomenon, Occam’s Razor advocates choosing the “simpler” explanation.

How to Control Overfitting?

- Use simpler models (linear instead of polynomial)
  - Might have poor results (underfitting)
- Use regularized complex models

$$\widehat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} J(\mathbf{\Theta}) + \lambda R(\mathbf{\Theta})$$

- $R()$ corresponds to the penalty paid for complexity of the model

l_2 Regularization

Ridge Regression

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \frac{1}{2}\lambda\|\mathbf{w}\|_2^2$$

- Helps in reducing impact of correlated inputs

- $\|\mathbf{w}\|_2^2$ is the square of the l_2 norm of the vector $\mathbf{w}$:

$$\|\mathbf{w}\|_2^2 = \sum_{i=1}^{D} w_i^2$$

Exact Loss Function

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{1}{2}\lambda\|\mathbf{w}\|_2^2 \\ &= \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2}\lambda\|\mathbf{w}\|_2^2 \end{aligned}$$

Ridge Estimate of w

$$\widehat{\mathbf{w}}_{Ridge} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_D)^{-1}\mathbf{X}^\top\mathbf{y}$$

- I_D is a $(D \times D)$ identity matrix.

The above derivation can be easily done by reusing the result from linear regression, where we calculated the gradient of the un-regularized loss function, which was the above term without the regularization parameter. Using the result that:

$$\frac{d}{d\mathbf{w}}\|\mathbf{w}\|_2^2 = 2\mathbf{w}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y} + \lambda\mathbf{w}$$

Setting above to 0 and solving for $\mathbf{w}$ gives us the above result.

Using Gradient Descent with Ridge Regression

- Very similar to OLE
- Minimize the squared loss using *Gradient Descent*

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2}\lambda\|\mathbf{w}\|_2^2$$

$$\begin{aligned} \nabla J(\mathbf{w}) = \frac{d}{d\mathbf{w}}J(\mathbf{w}) &= \frac{1}{2}\frac{d}{d\mathbf{w}}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2}\lambda\frac{d}{d\mathbf{w}}\|\mathbf{w}\|_2^2 \\ &= \mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y} + \lambda\mathbf{w} \end{aligned}$$

Using the above result, one can perform repeated updates of the weights:

$$\mathbf{w} := \mathbf{w} - \eta\nabla J(\mathbf{w})$$

l_1 Regularization

Least Absolute Shrinkage and Selection Operator - LASSO

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \lambda|\mathbf{w}|$$

- Helps in feature selection – favors sparse solutions
- Optimization is not as straightforward as in Ridge regression
  - Gradient not defined for $w_i = 0, \forall i$

2.2 Elastic Net Regularization

LASSO vs. Ridge

- Both control overfitting
- Ridge helps reduce impact of correlated inputs, LASSO helps in feature selection
- Rule of thumb
  - If data has many features but only few are potentially useful, use LASSO
  - If data has potentially many correlated features, use Ridge

Elastic Net Regularization

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \lambda_1 |\mathbf{w}| + \lambda_2 \|\mathbf{w}\|_2^2$$

- The best of both worlds
- Again, optimizing for $\mathbf{w}$ is not straightforward

3 Handling Outliers in Regression

- Linear regression training gets impacted by the presence of outliers
- The square term in loss function is the culprit
- How to handle this (*Robust Regression*)?
  - *Least absolute deviations* instead of least squares

$$J(\mathbf{w}) = \sum_{i=1}^{N} |y_i - \mathbf{w}^\top \mathbf{x}|$$

References