Introduction to Machine Learning

Extending Linear Regression

Varun Chandola

February 22, 2019

Outline

Contents

1 Shortcomings of Linear Models

1. Not truly Bayesian
2. Susceptible to outliers
3. *Too simplistic* - Underfitting
4. No way to control overfitting
5. Unstable in presence of correlated input attributes
6. Gets "confused" by unnecessary attributes

2 Bayesian Linear Regression

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*)

3 Handling Non-linear Relationships

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

$$p(y|\mathbf{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}))$$

- 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

3.1 Handling Overfitting via Regularization

How to Control Overfitting?

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

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

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

l_2 **Regularization**

Ridge Regression

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

- Helps in reducing impact of correlated inputs

Exact Loss Function

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

Ridge Estimate of w

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

Using Gradient Descent with Ridge Regression

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

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

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

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

$$w_j := w_j - \eta\frac{\partial J(\mathbf{w})}{\partial w_j}$$

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 $w_i = 0, \forall i$

3.2 Elastic Net Regularization

LASSO vs. Ridge

- Both control overfitting
- Ridge helps reduce impact of correlated inputs, LASSO helps in feature selection

Elastic Net Regularization

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

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

4 Handling Outliers in Regression

- Linear regression training gets impacted by the presence of outliers
- The square term in the exponent of the Gaussian pdf is the culprit
  - Equivalent to the square term in the loss
- 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