AIFS ML Lecture 6: Machine Learning Basics

Suraj Narayanan Sasikumar

Hessian AI Labs

May 06, 2020

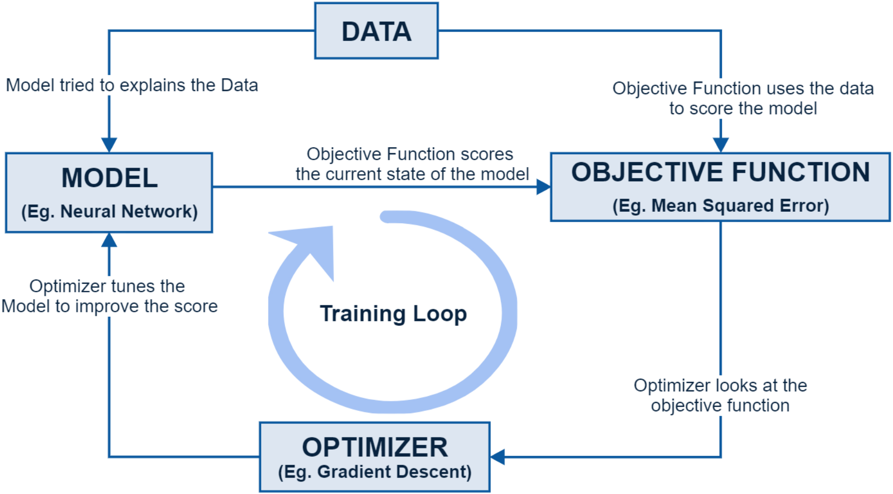
Overview

1. Recap
2. Observations from Polynomial Regression
3. Norm of a Vector
4. LASSO Regression
5. Ridge Regression

Table of Contents

1. Recap
2. Observations from Polynomial Regression
3. Norm of a Vector
4. LASSO Regression
5. Ridge Regression

Recap - Learning Algorithm



Recap - Polynomial Regression

- Polynomial Regression is Linear Regression with polynomial features.
- The goal is to find the best polynomial that explains the data, and has good prediction capability.
- Components of the algorithm
  - **Data:** $\boldsymbol{X}, \mathbf{y}$ ($\boldsymbol{X}$ has only one feature, i.e. $(D = 1)$)
  - **Model:** $\hat{y} = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M$
  - **Loss Function:** Minimize Mean Squared Error

$$\text{MSE}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

  - **Optimizer:** Closed Form Solution obtained by setting gradient of MSE to 0, $\nabla_{\mathbf{w}} \text{MSE}(\mathbf{w}) = 0$

$$\mathbf{w}^\star = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \mathbf{y}$$

Recap

- **Hyperparameters:** Configurable aspects of a learning algorithm that is set before training, and whose values has an impact on the performance, speed, and generalization capability of the learning algorithm.
- **Model Capacity:** The range of functions that can be learned by the model. Ability to learn wide range of functions $\Rightarrow$ high capacity. Can learn only a limited set of functions $\Rightarrow$ low capacity.
- **Underfitting:** Occurs when the model does not have enough capacity to explain the data, or there is not enough data for the model to fit properly.
- **Overfitting:** Occurs when the capacity of a model is much more than what is required to model the data-generating process. The Model tries to explain the noise as well.

Table of Contents

1 Recap

2 Observations from Polynomial Regression

3 Norm of a Vector

4 LASSO Regression

5 Ridge Regression

Observation from Polynomial Regression

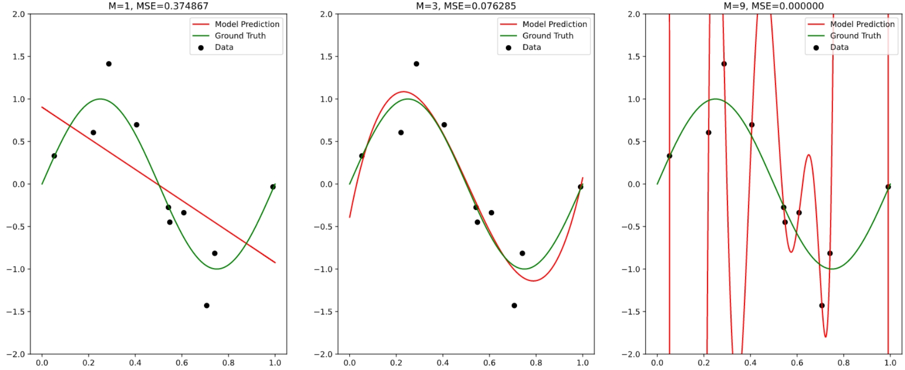
- In polynomial regression, we saw that for larger values of M, the model suffered from poor generalization.
- We observed that $\mathbf{w}^\star$ had the following values when $M = 1, 3, 9$

$\mathbf{w}^\star$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.90	-0.39	-52.4
$w_1^\star$	-1.82	14.13	1918.62
$w_2^\star$		-39.49	-23962.53
$w_3^\star$		25.82	148868.43
$w_4^\star$			-524354.65
$w_5^\star$			1103142.7
$w_6^\star$			-1393114.38
$w_7^\star$			1011368.03
$w_8^\star$			-372616.57
$w_9^\star$			48786.23

- We can see that, as M increases, $\mathbf{w}^\star$ takes on huge values that alternate positive and negative.

Observations from Polynomial Regression

- As the model over-fits, i.e. tries to explain the data including the noise, the resulting polynomial pass through each data-point in a quest to obtain zero loss.



- In $M = 9$, the large alternating positive and negative values of $\mathbf{w}^\star$ can be attributed to the curve having steep slopes and changing direction frequently to "explain" all data-points.

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.
- How can we leverage this intuition to control over-fitting?

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.
- How can we leverage this intuition to control over-fitting?
- Increase the loss (score) of those models with large parameter values by adding a penalty term to the loss function. Eg:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \text{penalty}$$

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.
- How can we leverage this intuition to control over-fitting?
- Increase the loss (score) of those models with large parameter values by adding a penalty term to the loss function. Eg:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \text{penalty}$$

- What is a good penalty term?

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.
- How can we leverage this intuition to control over-fitting?
- Increase the loss (score) of those models with large parameter values by adding a penalty term to the loss function. Eg:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \text{penalty}$$

- What is a good penalty term?
- Since the objective is to ensure that the parameter value's are small, the length of the parameter vector (or some function of its length) is a good fit.

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.
- How can we leverage this intuition to control over-fitting?
- Increase the loss (score) of those models with large parameter values by adding a penalty term to the loss function. Eg:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \text{penalty}$$

- What is a good penalty term?
- Since the objective is to ensure that the parameter value's are small, the length of the parameter vector (or some function of its length) is a good fit.
- But, how do we measure the length of a vector?

Observations from Polynomial Regression

- Can we draw a relationship between overfitting and our observation about $\mathbf{w}\star$?
- Intuitively, we can draw the conclusion that when the model overfits, the model parameter explodes.
- How can we leverage this intuition to control over-fitting?
- Increase the loss (score) of those models with large parameter values by adding a penalty term to the loss function. Eg:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \text{penalty}$$

- What is a good penalty term?
- Since the objective is to ensure that the parameter value's are small, the length of the parameter vector (or some function of its length) is a good fit.
- But, how do we measure the length of a vector?
- The norm of a vector gives its length.

Table of Contents

1. Recap
2. Observations from Polynomial Regression
3. Norm of a Vector
4. LASSO Regression
5. Ridge Regression

Norm of a Vector

- A *Norm* is a function that gives the length (magnitude/size) of a vector. The notation $||\cdot||$ is often used for a norm.
- Since a vector is representative of a point in space, the length of a vector is always relative to the origin (the zero vector: $\mathbf{0}$)
- For a function, $f(\cdot) = ||\cdot||$, to be a norm, it has to satisfy the following properties.
- *Triangle Inequality*: $||\mathbf{u}+\mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$
- *Absolute Homogeneity*: $||a\mathbf{u}|| = a||\mathbf{u}||$
- *Positive Definite*: If $||\mathbf{u}|| = 0 \Rightarrow \mathbf{u} = 0$

Examples of Norms

- $\mathbb{R}^n$ with ℓ_1-*norm* (Taxicab norm):

$$||\mathbf{u}|| = ||\mathbf{u}||_1 = \sum_{i=1}^{n} |u_i|$$

- $\mathbb{R}^n$ with ℓ^2-*norm* (Euclidean norm):

$$||\mathbf{u}|| = ||\mathbf{u}||_2 = \sqrt{u_1^2 + u_2^2 + \ldots + u_n^2}$$

Table of Contents

1 Recap

2 Observations from Polynomial Regression

3 Norm of a Vector

4 LASSO Regression

5 Ridge Regression

LASSO Regression

- When the ℓ_1-norm of the parameter vector ($\mathbf{w}$) is added to the loss function of linear regression, the resulting algorithm is called *LASSO Regression*.
- Components of the algorithm
  - **Data:** $\boldsymbol{X}, \mathbf{y}$
  - **Model:** $\hat{y} = \mathbf{w}^T\mathbf{x}$ (y-intercept included in the dot-product, $x_0 = 1$)
  - **Loss Function:** Minimize Mean Squared Error with ℓ_1 penalty

$$\text{MSE}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}^T\mathbf{x} - y_i)^2 + \lambda||\mathbf{w}||_1$$

  - **Optimizer:** LASSO Regression does not have a closed form solution. For LASSO, the following two iterative optimization algorithms are used:
    1. LARS: least-angle regression.
    2. Coordinate Descent. Minimize over one dimension (coordinate) at a time.

Table of Contents

1. Recap
2. Observations from Polynomial Regression
3. Norm of a Vector
4. LASSO Regression
5. Ridge Regression

Ridge Regression

- When the squared ℓ_2-norm of the parameter vector ($\mathbf{w}$) is added to the loss function of linear regression, the resulting algorithm is called *Ridge Regression*.
- Components of the algorithm
  - **Data:** $\boldsymbol{X}, \mathbf{y}$
  - **Model:** $\hat{y} = \mathbf{w}^T\mathbf{x}$ (y-intercept included in the dot-product, $x_0 = 1$)
  - **Loss Function:** Minimize Mean Squared Error with ℓ_2 penalty

$$\text{MSE}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}^T\mathbf{x} - y_i)^2 + \lambda||\mathbf{w}||_2^2$$

  - **Optimizer:** Closed Form Solution obtained by setting gradient of MSE to 0, $\nabla_{\mathbf{w}}\text{MSE}(\mathbf{w}) = 0$

$$\mathbf{w}^\star = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\mathbf{y}$$