Model Selection

Using Overfitting to evaluate different models

Presented by
Sai Harshinee Roopakula
19577

Table of Contents

Introduction - Underfitting

Underfitting:
A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data. *(It's just like trying to fit undersized pants!)* Underfitting destroys the accuracy of our machine learning model.

Its occurrence simply means that our model or the algorithm does not fit the data well enough.

It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data. In such cases the rules of the machine learning model are too easy and flexible to be applied on such minimal data and therefore the model will probably make a lot of wrong predictions.

In a nutshell, **Underfitting – High bias and low variance**
Techniques to reduce underfitting :
1. Increase model complexity
2. Increase number of features, performing feature engineering
3. Remove noise from the data.
4. Increase the number of epochs or increase the duration of training to get better results.

Introduction - Overfitting

Overfitting:
A statistical model is said to be overfitted, when we train it with a lot of data (*just like fitting ourselves in oversized pants!*).

When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too many details and noise.

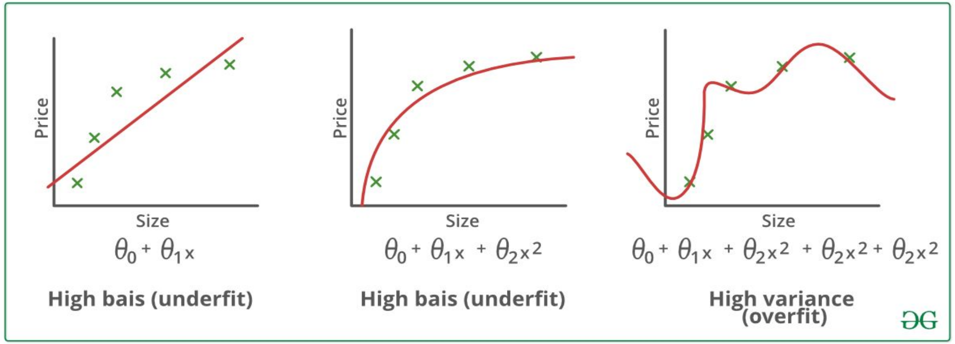
The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.

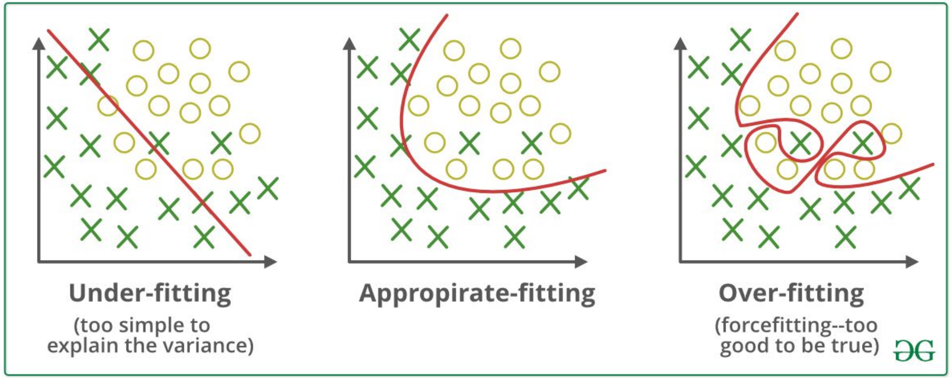
In a nutshell, **Overfitting – High variance and low bias**

Techniques to reduce overfitting :
1. Increase training data.
2. Reduce model complexity.
3. Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
4. Ridge Regularization and Lasso Regularization
5. Use dropout for neural networks to tackle overfitting.

Introduction - Underfitting & Overfitting

Examples of Underfitting and Overfitting





Design-Understanding the project

Suppose, we have collected a set of sample data and then distributed the sample data in the following way

- Training phase = 50%
- Validation phase = 25%
- Test phase = 25%

We are given two Regression models namely,

- Linear Regression - Model 1
- Non-Linear Regression - Model 2

Desired Output from the project

We have to compare the above 2 Regression models and see which one has more serious overfitting issue.

We have to select a better model depending on the analysis of overfitting and calculate $\hat{y}$ for the test phase data.

Design-Project Dataset

Training Data

Training phase (50% of the collected data)	
X	Y
1	1.8
2	2.4
3.3	2.3
4.3	3.8
5.3	5.3
1.4	1.5
2.5	2.2
2.8	3.8
4.1	4
5.1	5.4

Validation Data

Validation phase (25% of the collected data)	
X	Y
1.5	1.7
2.9	2.7
3.7	2.5
4.7	2.8
5.1	5.5

Test Data

Test phase (25% of the collected data)
X
1.4
2.5
3.6
4.5
5.4

Implementation - Finding Linear regression equation (Model1)

Regression Equation(y) = a + bx
Slope(b) = (NΣXY - (ΣX)(ΣY)) / (NΣX^2 - $(ΣX)^2$)
Intercept(a) = (ΣY - b(ΣX)) / N

To find the regression equation, we will first find the slope, intercept and use it to form a regression equation.

Step1:

Count the number of values. N=10

Step2:

Find X*Y and X^2

X	y	X*y	X*X
1	1.8	1.80	1.00
2	2.4	4.80	4.00
3.3	2.3	7.59	10.89
4.3	3.8	16.34	18.49
5.3	5.3	28.09	28.09
1.4	1.5	2.10	1.96
2.5	2.2	5.50	6.25
2.8	3.8	10.64	7.84
4.1	4	16.40	16.81
5.1	5.4	27.54	26.01

Implementation - Finding Linear regression equation (Model1)

Step3:

Find ΣX, ΣY, ΣXY, ΣX^2.

$\Sigma X = 31.80$
$\Sigma Y = 32.50$
$\Sigma XY = 120.80$
$\Sigma X^2 = 121.34$

Step4:

Substitute in the above slope formula given.

Slope(b) = $(N\Sigma XY - (\Sigma X)(\Sigma Y)) / (N\Sigma X^2 - (\Sigma X)^2)$
$= ((10)*(120.80)-(31.80)*(32.50))/((10)*(121.34)-(31.80)^2)$
= 0.863177681

	X	y	X*y	X*X
	1	1.8	1.80	1.00
	2	2.4	4.80	4.00
	3.3	2.3	7.59	10.89
	4.3	3.8	16.34	18.49
	5.3	5.3	28.09	28.09
	1.4	1.5	2.10	1.96
	2.5	2.2	5.50	6.25
	2.8	3.8	10.64	7.84
	4.1	4	16.40	16.81
	5.1	5.4	27.54	26.01
Sum	31.80	32.50	120.80	121.34

Implementation - Finding Linear regression equation (Model1)

Step5:

Now, again substitute in the above intercept formula given.

Intercept(a) = (ΣY - b(ΣX)) / N
= (32.50 - 0.863177681(31.80))/10
= 0.505094974

Step6:

Then substitute Intercept(a) and Slope(b) in regression equation formu

Regression Equation(y) = a + bx = 0.505094974 + 0.863177681x.

	X	y	X*y	X*X
	1	1.8	1.80	1.00
	2	2.4	4.80	4.00
	3.3	2.3	7.59	10.89
	4.3	3.8	16.34	18.49
	5.3	5.3	28.09	28.09
	1.4	1.5	2.10	1.96
	2.5	2.2	5.50	6.25
	2.8	3.8	10.64	7.84
	4.1	4	16.40	16.81
	5.1	5.4	27.54	26.01
Sum	31.80	32.50	120.80	121.34

Implementation - Finding Non- Linear regression (Model2)

Regression Equation(y) = $a + bx^2$

Slope(b) = $(N\Sigma\underline{P}Y - (\Sigma\underline{P})(\Sigma Y)) / (N\Sigma\underline{P}^2 - (\Sigma\underline{P})^2)$
Intercept(a) = $(\Sigma Y - b(\Sigma\underline{P})) / N$
where $\underline{P} = X * X$

X	$\underline{X}$	Y
1	1	1.8
2	4	2.4
3.3	10.9	2.3
4.3	18.5	3.8
5.3	28.1	5.3
1.4	1.96	1.5
2.5	6.25	2.2
2.8	7.84	3.8
4.1	16.8	4
5.1	26	5.4

Step 0:

We calculate $\underline{X}$ from X which is X*X

To find the regression equation, we will first find the slope, intercept and use it to form a regression equation.

Step1:

Count the number of values. N=10

Implementation - Finding Non- Linear regression (Model2)

Step 2:

Find $\underline{X} * Y$, $\underline{X}^2$

Step 3:

Find $\Sigma\underline{X}$, ΣY, $\Sigma\underline{X}Y$, $\Sigma\underline{X}^2$.

$\Sigma\underline{X} = 121.34$
$\Sigma Y = 32.5$
$\Sigma\underline{X}Y = 509.762$
$\Sigma\underline{X}^2 = 2329.9862$

Step4:

Substitute in the above slope formula given.

Slope(b) = $(N\Sigma\underline{X}Y - (\Sigma\underline{X})(\Sigma Y)) / (N\Sigma\underline{X}^2 - (\Sigma\underline{X})^2)$
$= ((10)*(509.762)-(121.34)*(32.5))/((10)*(2329.9862)-(121.34)^2)$
= 0.134562411

	X	Y	X*Y	X*X
	1	1.8	1.8	1
	4	2.4	9.6	16
	10.89	2.3	25.047	118.5921
	18.49	3.8	70.262	341.8801
	28.09	5.3	148.88	789.0481
	1.96	1.5	2.94	3.8416
	6.25	2.2	13.75	39.0625
	7.84	3.8	29.792	61.4656
	16.81	4	67.24	282.5761
	26.01	5.4	140.45	676.5201
Sum	121.34	32.5	509.762	2329.9862

Implementation - Finding Non- Linear regression (Model2)

Step5:

Now, again substitute in the above intercept formula given.

Intercept(a) = (ΣY - b(ΣX)) / N
= (32.5 - 0.134562411(121.34))/10
= 1.6172197

Step6:

Then substitute Intercept(a) and Slope(b) in regression equation

Regression Equation(y) = a + bx^2
= 1.6172197 + 0.134562411x^2

	X	Y	X*Y	X*X
	1	1.8	1.8	1
	4	2.4	9.6	16
	10.89	2.3	25.047	118.5921
	18.49	3.8	70.262	341.8801
	28.09	5.3	148.88	789.0481
	1.96	1.5	2.94	3.8416
	6.25	2.2	13.75	39.0625
	7.84	3.8	29.792	61.4656
	16.81	4	67.24	282.5761
	26.01	5.4	140.45	676.5201
Sum	121.34	32.5	509.762	2329.9862

Implementation - Calculating ŷ values

Now, we will calculate the approximate y value (ŷ value) for every x in the training phase and validation phase for both Model1 and Model2

For Example
Model 1 (a=0.505094974, b=0.863177681)

Suppose if we want to know the approximate y value for the variable x = 2. Then we can substitute the value in the below equation.

Regression Equation(y) = $a + bx$
= 0.505094974 + 0.863177681(2).
= 2.2315

Model 2 (a=1.6172197 , b=0.134562411)

Suppose if we want to know the approximate y value for the variable x = 2. Then we can substitute the value in the below equation.

Regression Equation(y) = $a + bx^2$
= $1.6172197 + 0.134562411(2)^2$
= 2.1555

Similarly, we find ŷ value for all x values in the training phase and validation phase for both Model 1 and Model 2

Implementation - Data Table

Training Phase				Validation Phase			
Real Data Set 1 50% of the collected data		Model 1: Linear Regression	Model 2: Non-Linear Regression	Real Data Set 2 25% of the collected data		Model 1: Linear Regression	Model 2: Non-Linear Regression
x	y	$\hat{y}=a1 + b1 * x$ $y = 0.505094974 + 0.863177681x$	$\hat{y}=a2 + b2 * x^2$ $y = 1.6172197 + 0.134562411x^2$	x	y	$\hat{y}=a1 + b1 * x$ $y = 0.505094974 + 0.863177681x$	$\hat{y}=a2 + b2 * x^2$ $y = 1.6172197 + 0.134562411x^2$
1	1.8	1.3683	1.7518	1.5	1.7	1.7999	1.9200
2	2.4	2.2315	2.1555	2.9	2.7	3.0083	2.7489
3.3	2.3	3.3536	3.0826	3.7	2.5	3.6989	3.4594
4.3	3.8	4.2168	4.1053	4.7	2.8	4.5620	4.5897
5.3	5.3	5.0799	5.3971	5.1	5.5	4.9073	5.1172
1.4	1.5	1.7135	1.8810	X	X	X	X
2.5	2.2	2.6630	2.4582	X	X	X	X
2.8	3.8	2.9220	2.6722	X	X	X	X
4.1	4	4.0441	3.8792	X	X	X	X
5.1	5.4	4.9073	5.1172	X	X	X	X

The above table shows the $\hat{y}$ values for the Training and Validation phase.

Implementation - Calculating MSE

The Mean Squared Error (MSE) is a measure of how close a fitted line is to data points.

- The smaller the MSE, the closer the fit is to the data.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2.$$

<u>For Training data</u>

Model 1

MSE =
((1.3683-1.8)^2+(2.2315-2.4)^2+(3.3536-2.3)^2+(4.2168-3.8)^2+(5.0799-5.3)^2+(1.7135-1.5)^2+(2.6630-2.2)^2+(2.9220-3.8)^2+(4.0441-4)^2+(4.9073-5.4)^2)/10 = 0.28225297

Model 2

MSE =
((1.7518-1.8)^2+(2.1555-2.4)^2+(3.0826-2.3)^2+(4.1053-3.8)^2+(5.3971-5.3)^2+(1.8810-1.5)^2+(2.4582-2.2)^2+(2.6722-3.8)^2+(3.8792-4)^2+(5.1172-5.4)^2)/10 = 0.235553231

Implementation - Calculating MSE

For Validation data

Model 1

MSE =
((1.7999-1.7)^2+(3.0083-2.7)^2+(3.6989-2.5)^2+(4.5620-2.8)^2+(4.9073-5.5)^2)/5 = 0.99966548

Model 2

MSE=
((1.9200-1.7)^2+(2.7489-2.7)^2+(3.4594-2.5)^2+(4.5897-2.8)^2+(5.1172-5.5)^2)/5 = 0.8641603

We can evaluate different models by using the formula:

max(Training_Set_MSE, Validation_Set_MSE) / min(Training_Set_MSE, Validation_Set_MSE)

Implementation - Comparing MSEs

<u>Compare Model 1 and Model 2</u>

Model 1

max(Training_Set_MSE, Validation_Set_MSE) / min(Training_Set_MSE, Validation_Set_MSE)
=0.99966548/0.28225297
= 3.5417

Model 1

max(Training_Set_MSE, Validation_Set_MSE) / min(Training_Set_MSE, Validation_Set_MSE)
=0.8641603/0.235553231
= 3.6686

Model 1 is better as it has less MSE(Mean Square Error)

Implementation - Calculating ŷ values for Test phase data

As we know the better model is Model 1, selected from both the training and Validation phase, we use Model 1 a, b values to find ŷ values of test phase data.

Test Phase	
Real Data Set 3 **25% of the collected data**	The better model (Model 1 or Model 2) selected from the Validation Phase based on the analysis of overfitting will be used to calculate ŷ
x	**ŷ=a1 + b1 * x** $y = 0.505094974 + 0.863177681x$
1.4	1.7135
2.5	2.6630
3.6	3.6125
4.5	4.3894
5.4	5.1663
X	X
X	X
X	X
X	X
X	X

Test Results - Complete data table

Training Phase				Validation Phase				Test Phase	
Real Data Set 1 50% of the collected data		Model 1: Linear Regression	Model 2: Non-Linear Regression	Real Data Set 2 25% of the collected data		Model 1: Linear Regression	Model 2: Non-Linear Regression	**Real Data Set 3** **25% of the collected data**	The better model (Model 1 or Model 2) selected from the Validation Phase based on the analysis of overfitting will be used to calculate $\hat{y}$
x	y	$\hat{y}=a1 + b1 * x$ $y = 0.505094974 + 0.863177681x$	$\hat{y}=a2 + b2 * x^2$ $y = 1.6172197 + 0.134562411x^2$	x	y	$\hat{y}=a1 + b1 * x$ $y = 0.505094974 + 0.863177681x$	$\hat{y}=a2 + b2 * x^2$ $y = 1.6172197 + 0.134562411x^2$	**x**	$\boldsymbol{\hat{y}=a1 + b1 * x}$ $y = 0.505094974 + 0.863177681x$
1	1.8	1.3683	1.7518	1.5	1.7	1.7999	1.9200	1.4	1.7135
2	2.4	2.2315	2.1555	2.9	2.7	3.0083	2.7489	2.5	2.6630
3.3	2.3	3.3536	3.0826	3.7	2.5	3.6989	3.4594	3.6	3.6125
4.3	3.8	4.2168	4.1053	4.7	2.8	4.5620	4.5897	4.5	4.3894
5.3	5.3	5.0799	5.3971	5.1	5.5	4.9073	5.1172	5.4	5.1663
1.4	1.5	1.7135	1.8810	X	X	X	X	X	X
2.5	2.2	2.6630	2.4582	X	X	X	X	X	X
2.8	3.8	2.9220	2.6722	X	X	X	X	X	X
4.1	4	4.0441	3.8792	X	X	X	X	X	X
5.1	5.4	4.9073	5.1172	X	X	X	X	X	X

Conclusion

Here, we have evaluated Linear Regression Model (Model 1) and Non-Linear Regression model (Model 2) using Overfitting.

We have found that Model 1 has less MSE (Mean Squared Error) and therefore better than Model 2.

Thus we have used Model 1's a, b values to predict y values of Test Phase data.

Bibliography

https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

https://npu85.npu.edu/~henry/npu/classes/data_science/algorithm/slide/linear_regression_example.html

https://npu85.npu.edu/~henry/npu/classes/data_science/algorithm/slide/non_linear_regression_example.html

https://npu85.npu.edu/~henry/npu/classes/data_science/algorithm/slide/overfit.html

Link to view the presentation
https://docs.google.com/presentation/d/1zO7a5CTktUDbzhb3iUy8v5vMAFjWxRhUXEo-l5Zf-kE/edit?usp=sharing