Machine Learning

Bias Variance Trade-off

Indian Institute of Information Technology
Sri City, Chittoor

Today’s Agenda

- Error of the Linear regression model
- Fitting Non-linear Data
- Bias Variance
- Underfitting
- Overfitting
- Bias Variance Trade-off

Error of Regression

- Let us assume that the target variables and the inputs are related via the equation

$$Y=f(X) + e \quad \Rightarrow \quad y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

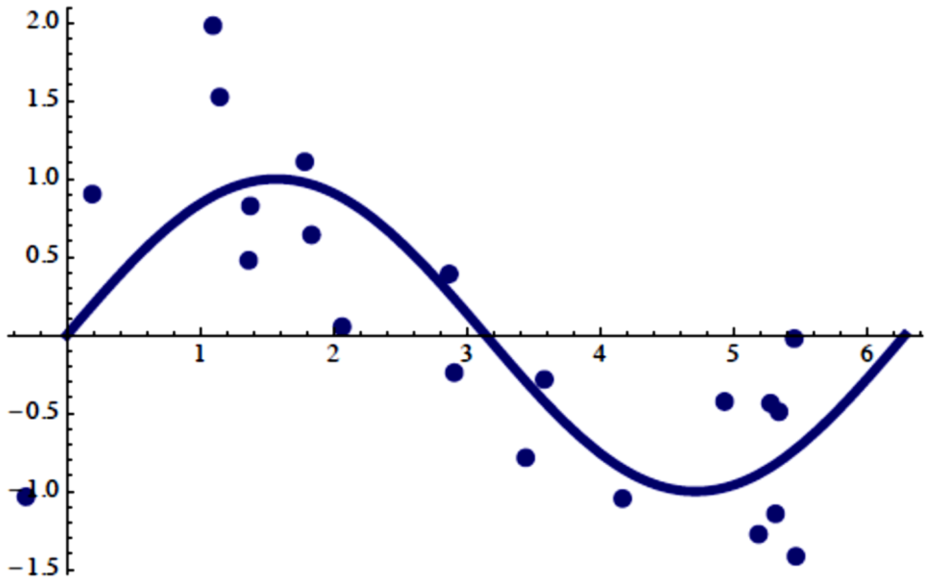
where $\epsilon^{(i)}$ is an error term that captures either unmodeled effects or random noise.

- Let us further assume that the $\epsilon^{(i)}$ are distributed IID (independently and identically distributed) according to a Gaussian distribution with mean zero and some variance σ^2.

- We can write this assumption as "$\epsilon(i) \sim N(0, \sigma^2)$."

Fitting Non-linear Data

- What if Y has a non-linear response?



- Can we still use a linear model?

Transforming the feature space

Transform features x_i

$$x_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,p})$$

By applying non-linear transformation ϕ:

$$\phi : \mathbb{R}^p \rightarrow \mathbb{R}^k$$

Example:

$$\phi(x) = \{1, x, x^2, \ldots, x^k\}$$

– others: splines, radial basis functions, …

– Expert engineered features (modeling)

Basis Function Choices

- **Polynomial**

$$\phi_j(x) = x^j$$

- **Gaussian**

$$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$

- **Sigmoidal**

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \text{ with } \sigma(a) = \frac{1}{1+e^{-a}}$$

- **splines**, **Fourier**, **wavelets**, etc.

Error of Regression

- So the expected squared error at a point x is

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

- The Err(x) can be further decomposed as expression for the expectation of the loss function:

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

- Err(x) is the sum of Bias^2, variance and the irreducible error.
- *Irreducible error is the error that can't be reduced by creating good models. It is a measure of the amount of noise in our data*.

Bias

- Given: dataset *D* with *m* samples.

- Learn: for different datasets D, you will get different functions f(x).
- Expected prediction (averaged over hypotheses): $E_D\left[f(x)\right]$

- Bias: difference between expected prediction and ground truth

– Measures how well you expect to represent true solution

– Decreases with more complex model

$$\text{Bias}^2 = \left(E[\hat{f}(x)] - f(x)\right)^2$$

Variance

- Given: dataset *D* with *m* samples.

- Learn: for different datasets D, you will get different functions f(x):
- Expected prediction (averaged over hypotheses): $E_D\,[f(x)]$

- Variance : difference between what you expect to learn and what you learn from a from a particular dataset.

– Measures how sensitive is learner is to the specific dataset

– Decreases with the simpler model.

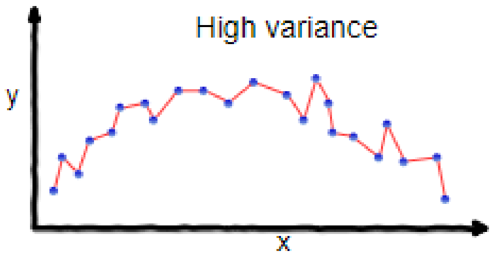
Variance= $E\left[\left(\hat{f}\,(x) - E[\hat{f}\,(x)]\right)^2\right]$

Underfitting

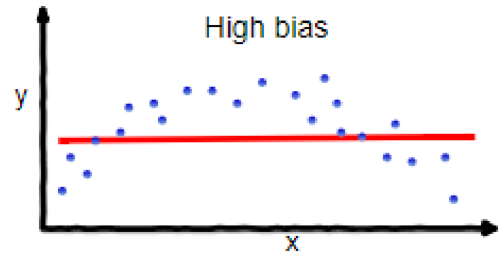
- In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data.
- These models usually have high bias and low variance.
- It happens when we have very less amount of data to build an accurate model or when we try to build a linear model with a nonlinear data.
- Also, these kind of models are very simple to capture the complex patterns in data like Linear and logistic regression.

Overfitting

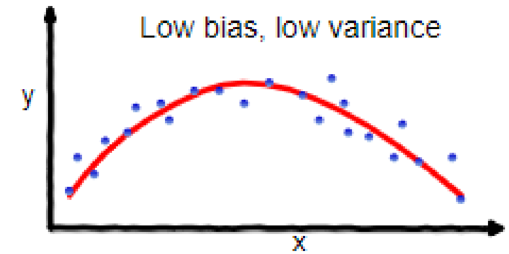
- In supervised learning, **overfitting** happens when our model captures the noise along with the underlying pattern in data.
- It happens when we train our model a lot over noisy dataset.
- These models have low bias and high variance.
- These models are very complex like Decision trees which are prone to overfitting.



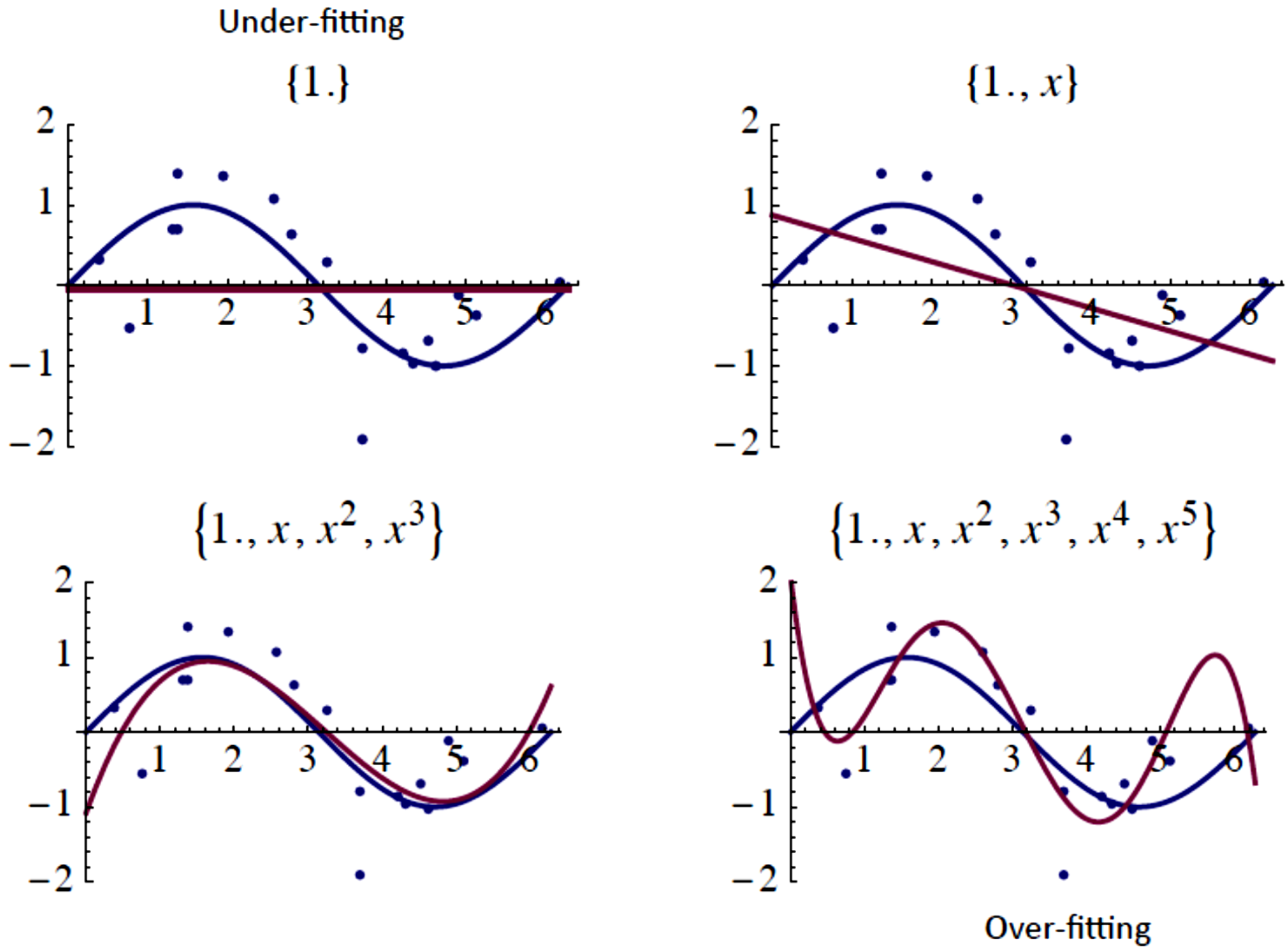
overfitting



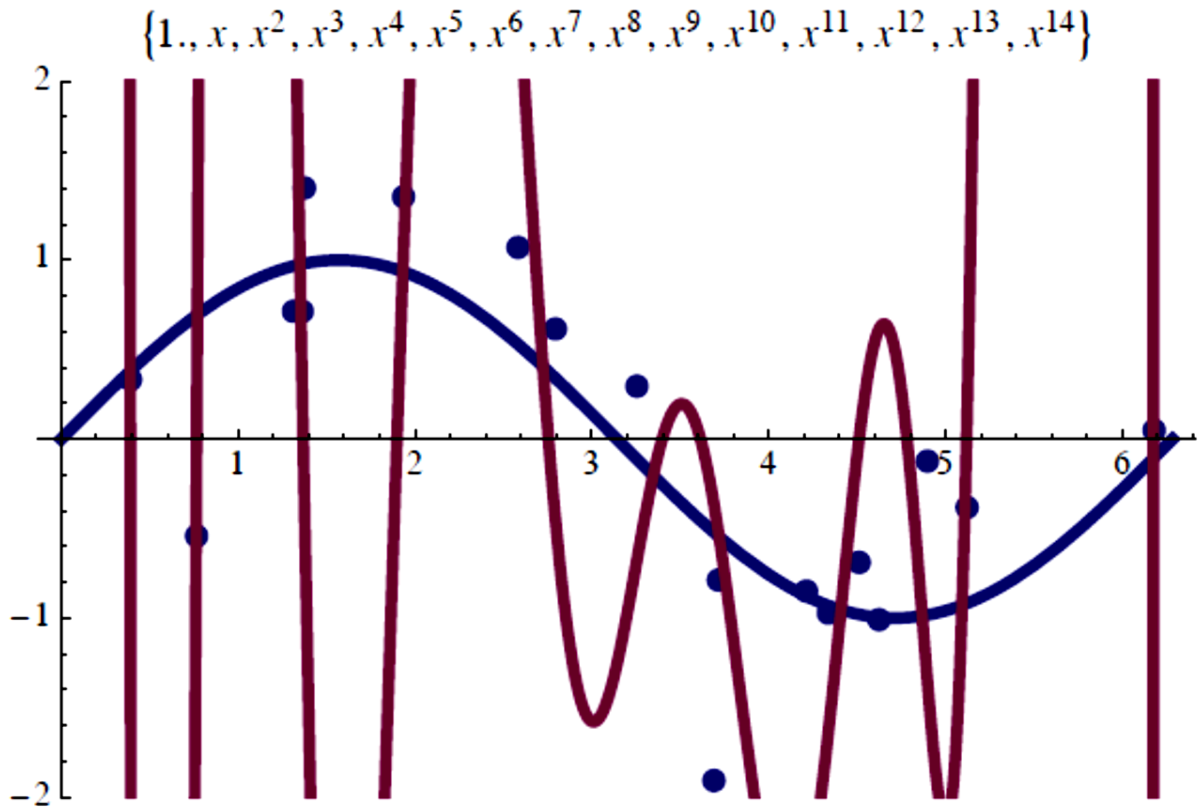
underfitting



Good balance


Under-fitting
{1.}
{1., x}
{1., x, x^2, x^3}
{1., x, x^2, x^3, x^4, x^5}
Over-fitting

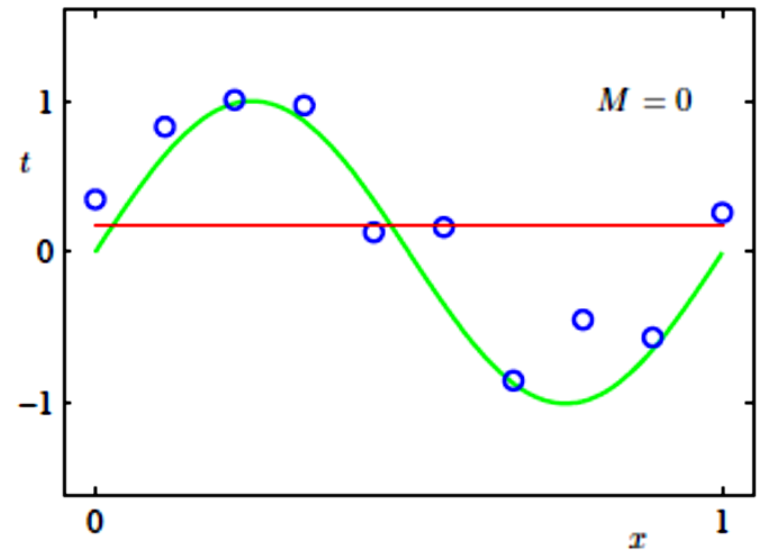
Really Over-fitting!



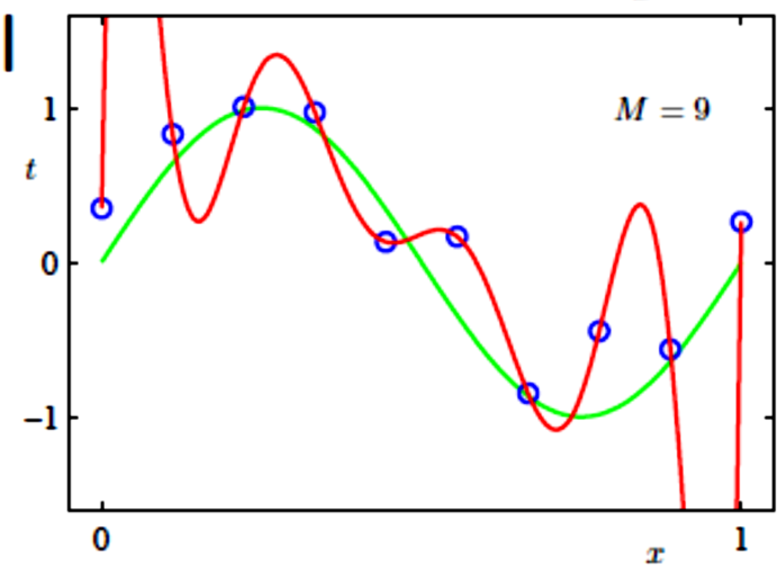
- Errors on training data are small
- But errors on new points are likely to be large

Bias Variance trade-off Intuition

- Model too simple: does not fit the data well
  - A *biased* solution



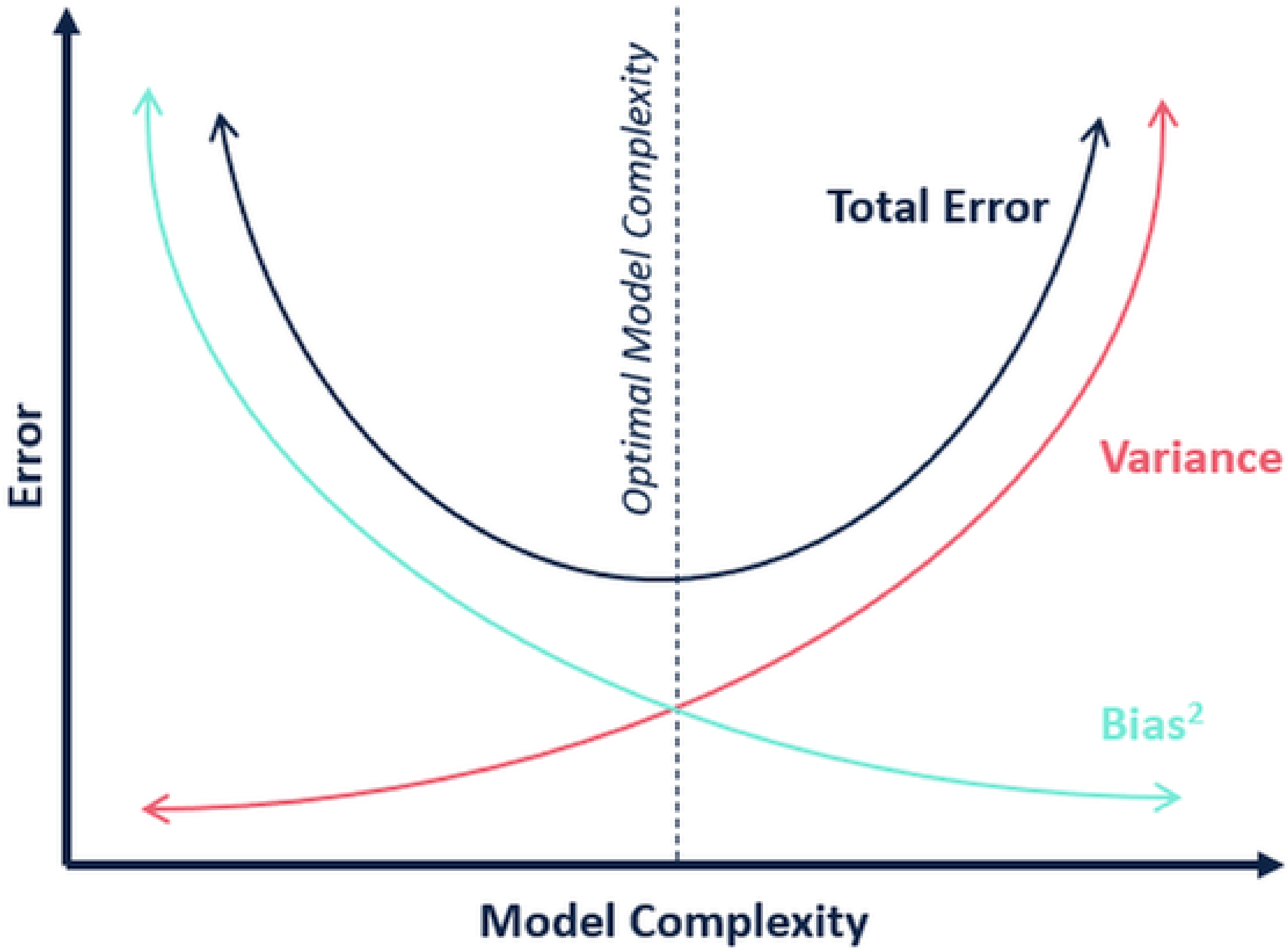
- Model too complex: small changes to the data, solution changes a lot
  - A *high-variance* solution



Bias Variance Trade-off

- If our model is too simple and has very few parameters then it may have high bias and low variance.
- On the other hand if our model has large number of parameters then it's going to have high variance and low bias.

So we need to find the right/good balance without overfitting and underfitting the data.



Additional Reading I found Helpful

http://www.stat.cmu.edu/~roeder/stat707/lectures.pdf

http://people.stern.nyu.edu/wgreene/MathStat/GreeneChapter4.pdf

http://www.seas.ucla.edu/~vandenbe/103/lectures/qr.pdf