CS 217: Artificial Intelligence and Machine Learning **Jan-Apr 2024**

Lecture 6: Bias and Variance

Lecturer: Swaprava Nath *Scribe(s): SG11,SG12*

Disclaimer: *These notes aggregate content from several texts and have not been subjected to the usual scrutiny deserved by formal publications. If you find errors, please bring to the notice of the Instructor.*

6.1 Recap of MAP Estimate

$$w^*_{MAP} \in \arg\min_w (\frac{\|Xw - y\|^2}{2\sigma^2} + \frac{\lambda}{2}\|w\|^2)$$
$$= \frac{1}{\sigma^2}(\frac{1}{2\sigma^2}X^TX + \frac{\lambda}{2}I)^{-1}X^Ty$$

$A_{d\times d}$ is a Positive definite if $\forall x \in R^d \setminus \{0\}$, $x^TAx > 0$

Here $A = \frac{1}{2\sigma^2}X^TX + \frac{\lambda}{2}I$

$v^TAv = \frac{1}{2\sigma^2}v^TX^TXv + \frac{\lambda}{2}\|v\|^2 = \frac{1}{2\sigma^2}\|Xv\|^2 + \frac{\lambda}{2}\|v\|^2 > 0$, for all $\lambda > 0$ and $v \in R^d \setminus \{0\}$

Therefore, matrix A is positive definite. Any positive definite matrix is invertible, hence the matrix $\frac{1}{2\sigma^2}X^TX + \frac{\lambda}{2}I$ is invertible.

6.2 Bias and Variance

Goal: Estimate $\hat{y}$ which was not seen in our training example.

We have ($\hat{x}$, $\hat{y}$) as our test data points.We also have $(x_i, y_i) \in D_{train}$ as our training data points.

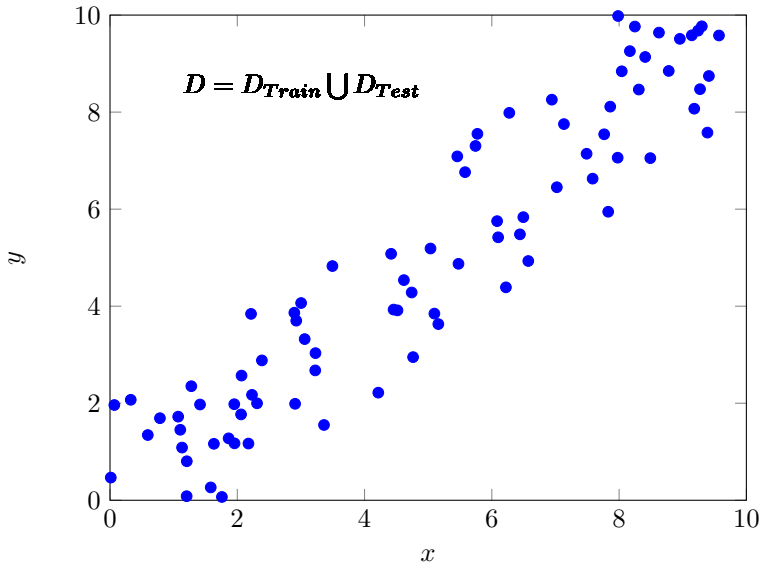
We want to find $f_D(\hat{x})$ to be as close to $\hat{y}$, $f_D(\hat{x}) \sim \hat{y}$ trained on dataset D.

- To measure how good the fit was, we have different measure of goodness which are listed below

1. **Training Error:** This error is given by the sum of the loss function applied to each training example:

$$\sum_{i\in D} \ell(g_D(x_i), y_i)$$

Here ℓ is the loss function, $\{x_i, y_i\} \in$ training data set and g_D represents the prediction function.

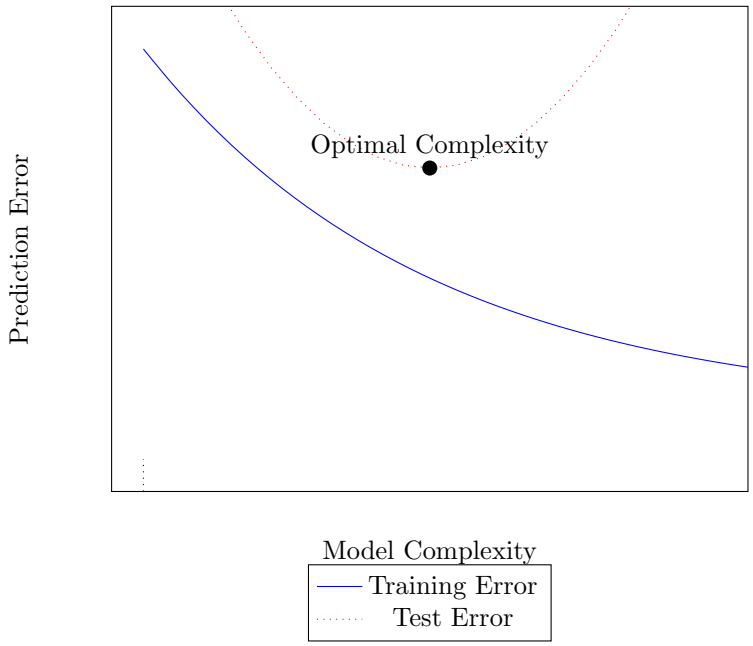


2. **Test Error:**

- We split the data set into D_{Train} and D_{Test}. Now, we hold out the set D_{Test} and train on the set D_{Train}. The test error is then calculated as:

$$\sum_{j \in D_{\text{Test}}} \ell(g_D(x_j), y_j).$$

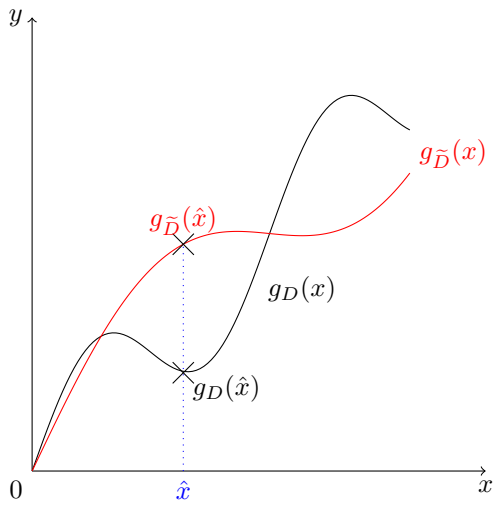
- If all data points are one dimensional and model is polynomial then degree of polynomial represents model complexity. When we increment the degree of polynomial, it might completely fit on higher degree but test data might not be fitting to the polynomial.



- **Primary contributors of Test Error:**

$\underbrace{a)Bias \qquad b)Variance}_{\text{model dependent}} \qquad\qquad \underbrace{c)Noise}_{\text{model independent as it arises due to data set}}$

6.3 What are Bias & Variance ?



- Our dataset that we sample is also an random variable.
  For example if the model is linear then $g_{\widetilde{D}}(x) = w^T x$
- If we sample a particular dataset D we will get an estimate to our curve as $g_D(x)$. If we would have sampled some other dataset say $\widetilde{D}$ we would get an estimate curve $g_{\widetilde{D}}(x)$. Thus our estimate curve depends on the sampled Dataset which makes it a random variable.
- Here y is also a random variable due to the noise present

$$y = f(x) + \epsilon, \quad \text{where } \epsilon \text{ is from the distribution } \mathcal{N}(\mu, \sigma^2)$$

$$f(x) \text{ is the function we want to estimate}$$

- Let $(\hat{x}, \hat{y})$ be our test data.
- It is important to note that we assume $\hat{y}$ to be independent of our Dataset D keeping $\hat{x}$ fixed.

Test Error:

$$err = g_D(\hat{x}) - \hat{y} \tag{6.1}$$

$$= \underbrace{(g_D(\hat{x}) - \mathbb{E}(g_D(\hat{x})))}_{\text{A}} + \underbrace{(\mathbb{E}(g_D(\hat{x})) - \mathbb{E}(\hat{y}))}_{\text{B}} + \underbrace{(\mathbb{E}(\hat{y}) - \hat{y})}_{\text{C}} \tag{6.2}$$

The D distribution is independent of $\hat{y}$ distribution. This is known as independence assumption.
Here we note that the term B is just a constant as the expectation of any variable is a constant. Also $\mathbb{E}(A)$ and $\mathbb{E}(C)$ are zero because

$$\mathbb{E}(A) = \mathbb{E}(g_D(\hat{x})) - \mathbb{E}(\mathbb{E}(g_D(\hat{x}))) = 0 \tag{6.3}$$

$$\mathbb{E}(C) = \mathbb{E}(\mathbb{E}(\hat{y})) - \mathbb{E}(\hat{y}) = 0 \tag{6.4}$$

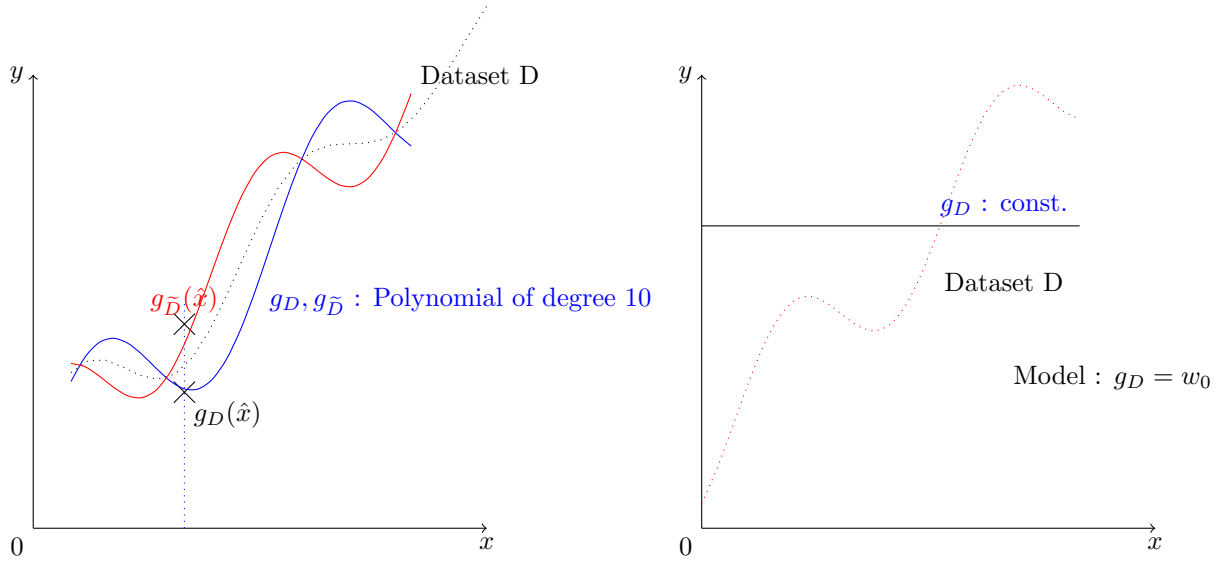
$$\therefore \mathbb{E}(err^2) = \mathbb{E}(A^2) + \underbrace{\mathbb{E}(B^2)}_{} + \mathbb{E}(C^2) + 2[\underbrace{\mathbb{E}(AB) + \mathbb{E}(BC) + \mathbb{E}(AC)}_{0}] \tag{6.5}$$

$$\therefore \mathbb{E}(err^2) = \underbrace{\mathbb{E}(A^2)}_{\text{Variance}} + \underbrace{B^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}(C^2)}_{\text{Noise}} \tag{6.6}$$

Thus we get the following expressions:

- **Variance of the model** : $\mathbb{E}[(g_D(\hat{x}) - \mathbb{E}(g_D(\hat{x})))^2]$
- **Bias of the model** : $\mathbb{E}[g_D(\hat{x}] - \mathbb{E}(\hat{y})$
- **Noise of the model** : $\mathbb{E}[(\hat{y} - \mathbb{E}(\hat{y}))^2]$

6.4 Behaviour of Variance and Bias



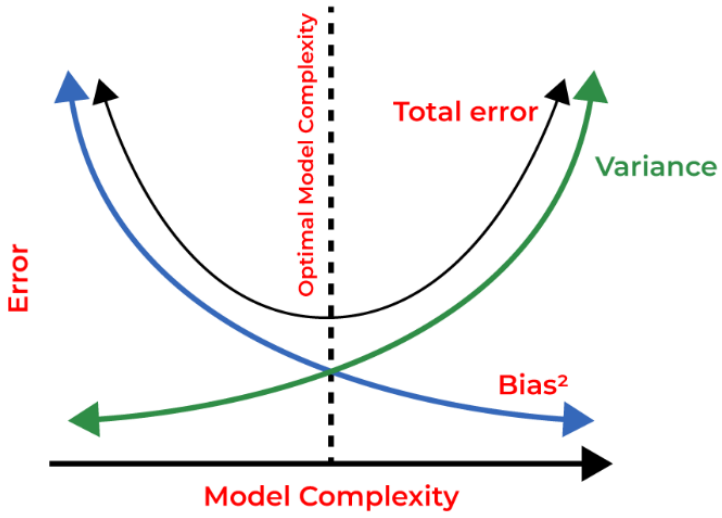
- As the model is highly complex it tries to fit each and every point in the Dataset.
- This leads the estimate curve to change drastically after changing the Dataset slightly. Leading to **High Variance**.
- As it almost fits the Dataset perfectly it has a **Low Bias**.
- These kind of fittings are called **Overfitting**.

- As the estimate does not change by a significant amount by change in Dataset, these models will have **Low Variance**

$$\begin{aligned} Variance &= \mathbb{E}(g_D(\hat{x}) - E_D(g_D(\hat{x})))^2 \\ &= \mathbb{E}((w_0 - w_0)^2) = 0 \end{aligned}$$

- It will have a **High Bias** as it is not trying hard enough to fit the data.
- These kind of fittings are called **Underfitting**.

A simple model (less model complexity) will have high train errors and a higher λ which makes the model underfit. Whereas a complex model has a high dependence on training data points leading to overfitting and has a smaller λ



6.5 Regularization for Linear Regression

The optimisation problem representing the linear regression model is given as follows:

- $w_{\text{MLE}} \in \arg\min \left\{ \frac{1}{2\sigma^2} \|\mathbf{X}w - y\|_2^2 \right\}$
- $w_{\text{MAP}} \in \arg\min \left\{ \frac{1}{2\sigma^2} \|\mathbf{X}w - y\|_2^2 + \frac{\lambda}{2} \|w\|_2^2 \right\}$

For a data set of a fixed given size, the general representation of a regularized model is :

$$\boxed{\text{Loss}(w) = \text{Loss}_D(w) + \lambda \text{Reg}(w)}$$

Here

- λ is a **hyperparameter** ie. is chosen beforehand by trial and error from a grid of possible values.
- $\lambda \text{Reg}(w)$ is the **regularizer** term which is a function of parameter w
- $\text{Loss}_D(w)$ represents the loss on data set ie. data error which depends on both data and parameters.

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.
There are two main types of regularization techniques:
Ridge Regularization and **Lasso Regularization**.
The regression model is classified depending on what the regulariser function is.

6.5.1 Ridge regression model:

Also known as Ridge Regularization, it modifies the over-fitted or under-fitted models by adding the penalty equivalent to the sum of the squares of the magnitude of coefficients (L2 norm square).

In this model, the regulariser function is :

$$\text{Reg}(w) = \|w\|_2^2 = \sqrt{\sum_{i=1}^{d} {w_i}^2}$$

The following formulation represents it as an optimisation problem:

$$w^* \in \arg\min_w \{ \ \|\mathbf{\Phi} w - y\|_2^2 \ + \lambda \|w\|_2^2 \ \}$$

It has a closed-form expression and is differentiable.
Here,

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & . & . & . & \phi_m(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & . & . & . & \phi_m(x_2) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \phi_0(x_n) & \phi_1(x_n) & . & . & . & \phi_m(x_n) \end{bmatrix}$$

This optimisation problem can be equivalently represented as :

$$\arg\min_w \ (\Phi w - y)^T (\Phi w - y)$$

$$\text{subject to constraint } \|w\|_2 \leq C_1$$

where C_1 is a constant which depends on the λ we have chosen.

6.5.2 LASSO regression model:

It modifies the over-fitted or under-fitted models by adding the penalty equivalent to the sum of the absolute values of coefficients (L1 norm).
LASSO stands for **Least absolute shrinkage and selection operator**.
In this model, the regulariser function is :

$$\text{Reg}(w) = \|w\|_1 = \sum_{i=1}^{d} |w_i|$$

The following formulation represents it as an optimisation problem:

$$w^* \in \arg\min_w \{ \ \|\Phi w - y\|_2^2 \ + \lambda \|w\|_1 \ \}$$

It doesn't have a closed-form expression and is non-differentiable.
This optimisation problem can be equivalently represented as :

$$\arg\min_w \ (\Phi w - y)^T (\Phi w - y)$$

$$\text{subject to constraint } \|w\|_1 \leq C_2$$

where C_2 is a constant which depends on the λ we have chosen.