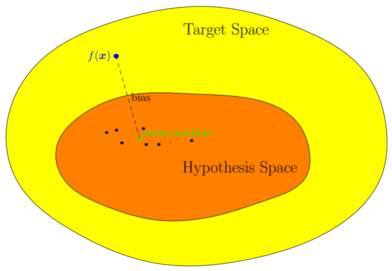
Advanced Machine Learning

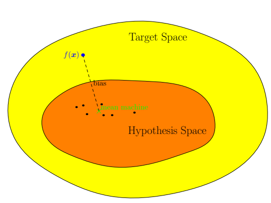
When Machine Learning Works



When ML Works, Bias Variance

Outline

1. What Makes a Good Learning Machine?
2. Bias-Variance Dilemma



What Makes a Good Learning Machine?

- We want to understand why some machine learning techniques work well and other don't
- To understand why these works we need to understand what makes a good learning machine
- For this we have to get conceptual and think about **generalisation** performance

  generalisation*: how well do we do on unseen data as opposed to the training data*

What Makes Machine Learning Hard?

- Typically we work in high dimensions (i.e. have many features)
- The problem can be over-constrained (i.e. we have conflicting data to deal with)—solve by minimising an error function
- The problem can be under-constrained (i.e. there are many possible solutions that are consistent with the data)—need to choose a plausible solution
- Typically in machine learning the data will be over-constrained in some dimensions and under-constrained in others
- We can't visualise the data to know what is going on

Least Squared Errors

- Suppose we want to learn some output y for a feature vector $\boldsymbol{x}$
- We construct a learning machine that makes a prediction $\hat{f}(\boldsymbol{x}|\boldsymbol{\theta})$
- We typically choose the machine to minimise a *training loss*

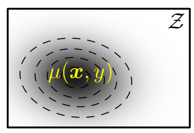
$$L_T(\mathcal{D}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}) - y\right)^2 = \sum_{i=1}^{m} \left(\hat{f}(\boldsymbol{x}_i|\boldsymbol{\theta}) - y_i\right)^2$$

  where $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ is a set of size m, sampled from a probability distribution $\mu(\boldsymbol{x}, y)$
- We call this machine $\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D})$

Generalisation Error

- We want to minimise the *generalisation loss* which in this case is

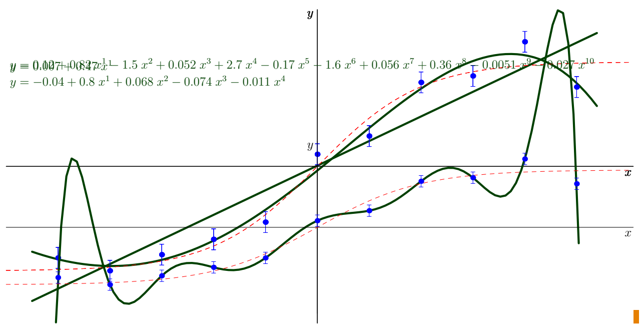
$$L_G(\mathcal{D}) = \sum_{(\boldsymbol{x},y)\in\mathcal{Z}} \mu(\boldsymbol{x}, y) \left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - y\right)^2$$



  (we can estimate this if we have some labelled examples $(\boldsymbol{x}_i, y_i)$ which we have not trained on)
- We want to minimise $L_G(\mathcal{D})$ but in practice we are minimising $L_T(\mathcal{D})$, *what could possibly go wrong?*

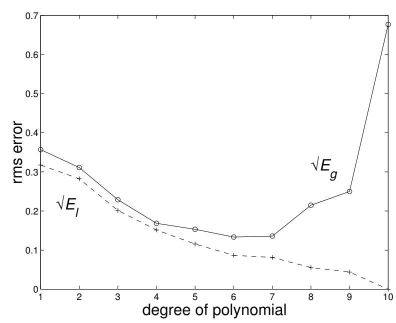
Too Simple or Too Complex?

- Fit $\hat{f}(x|\boldsymbol{\theta}_\mathcal{D})$ to data

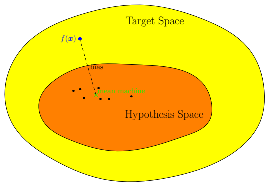


Measuring Generalisation Error for Regression

- Consider the regression example. The root mean squared error is



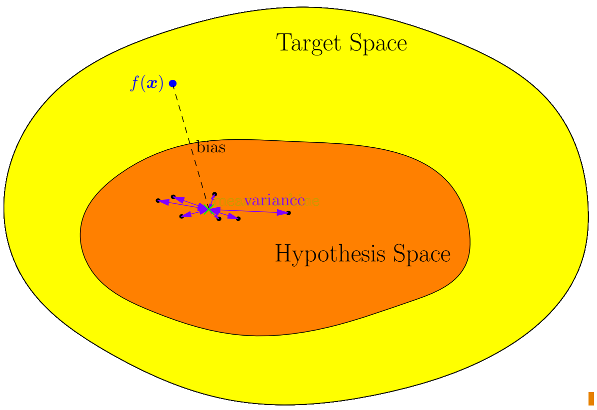
Outline



Expected Generalisation Performance

- Our generalisation performance will depend on our training set, $\mathcal{D}$
- To reason about generalisation we can ask what is the *expected generalisation loss*, when we average over all different data sets of size m drawn independently from $\mu(\boldsymbol{x},y)$
- For each data set, $\mathcal{D}$, we would learn a different approximator $\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D})$
- Note that in practice we only get one data set. We might be lucky and do better than the expected generalisation or we might be unlucky and do worse

Approximation and Estimation Errors



Mean Machine

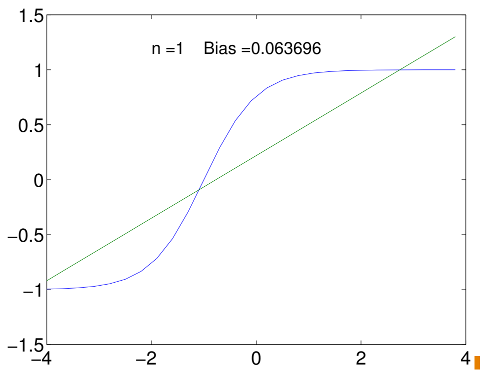
- To help understand generalisation we can consider the mean prediction with respect to machines trained with all data sets of size m

$$\hat{f}_m(\boldsymbol{x}) = \mathbb{E}_\mathcal{D}\Big[\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D})\Big]$$

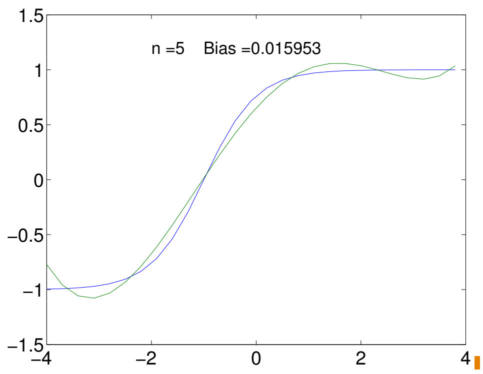
- We can define the **bias** to be generalisation performance of the mean machine

$$B = \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\left(\hat{f}_m(\boldsymbol{x}) - y\right)^2$$

Regression Example $n = 1$



Regression Example $n = 5$



Bias and Variance

Consider the expected generalisation for data sets of size $|\mathcal{D}| = m$

$$\begin{aligned}
\bar{L}_G = \mathbb{E}_\mathcal{D}[L_G(\mathcal{D})] &= \mathbb{E}_\mathcal{D}\left[\sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - y\right)^2\right] \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\,\mathbb{E}_\mathcal{D}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - y\right)^2\right] \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\,\mathbb{E}_\mathcal{D}\left[\left(\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right) + \left(\hat{f}_m(\boldsymbol{x}) - y\right)\right)^2\right] \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\Bigg(\mathbb{E}_\mathcal{D}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)^2 + \left(\hat{f}_m(\boldsymbol{x}) - y\right)^2\right] \\
&\qquad + 2\,\mathbb{E}_\mathcal{D}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)\left(\hat{f}_m(\boldsymbol{x}) - y\right)\right]\Bigg)
\end{aligned}$$

Cross Term

- The cross term vanishes

$$\begin{aligned}
C &= \mathbb{E}_\mathcal{D}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)\left(\hat{f}_m(\boldsymbol{x}) - y\right)\right] \\
&= \mathbb{E}_\mathcal{D}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)\right]\left(\hat{f}_m(\boldsymbol{x}) - y\right) \\
&= \left(\mathbb{E}_\mathcal{D}\left[\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D})\right] - \hat{f}_m(\boldsymbol{x})\right)\left(\hat{f}_m(\boldsymbol{x}) - y\right) \\
&= \left(\hat{f}_m(\boldsymbol{x}) - \hat{f}_m(\boldsymbol{x})\right)\left(\hat{f}_m(\boldsymbol{x}) - y\right) = 0
\end{aligned}$$

- Thus

$$\bar{L}_G = \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\,\mathbb{E}_\mathcal{D}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)^2 + \left(\hat{f}_m(\boldsymbol{x}) - y\right)^2\right]$$

Bias and Variance

- We can write the expected generalisation loss as

$$\mathbb{E}_{\mathcal{D}}[L_G(\mathcal{D})] = \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{D}}) - \hat{f}_m(\boldsymbol{x})\right)^2\right] + \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\left(\hat{f}_m(\boldsymbol{x}) - y\right)^2 = V + B$$

- Where B is the bias and V is the variance defined by

$$V = \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{D}}) - \hat{f}_m(\boldsymbol{x})\right)^2\right]$$

Bias-Variance Dilemma

- The bias measure the generalisation performance of the *mean machine* and is large if the machine is too simple to capture the changes in the function we want to learn
- The variance measures the variation in the prediction of the machines as we change the data set we train on

$$V = \sum_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x},y)\mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{D}}) - \hat{f}_m(\boldsymbol{x})\right)^2\right]$$

- The variance is usually large if we have a complex machine
- Striking the right balance is often the key to getting good results

Balancing Bias and Variance

- We want to choose a learning machine that is complex enough to capture the underlying function we are trying to learn, but otherwise as simple as possible
- There are a number of tricks to achieve this balance
- Some require us to preprocess the data to reduce the number of inputs
- Some machines cleverly adjust their own complexity
- This course looks at machines that achieve this balance

Lessons

- This course is about understanding machine learning techniques that work well
- Which one to use will depend on the data set
- One of the most useful intuitions about what works is the bias-variance framework
- The bias is high for simple machines that can't capture the data
- The variance is high for complex machines that are sensitive to the training set
- Good machines are powerful enough to capture complex data sets, but they can control their own capacity (ability to (over-)fit the data)