Advanced Machine Learning Subsidary Notes

Lecture 1: When Machine Learning Works

Adam Prügel-Bennett

February 7, 2021

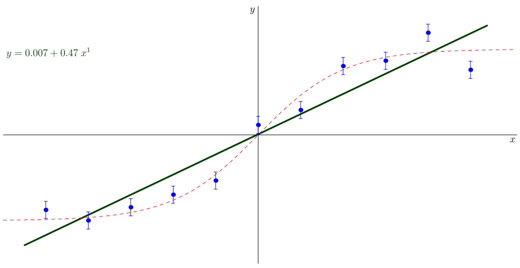
1 Keywords

- When ML Works, Bias Variance

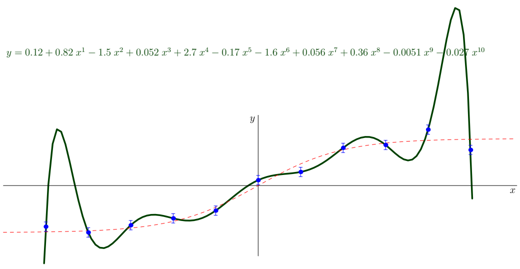
2 Main Points

2.1 Generalisation

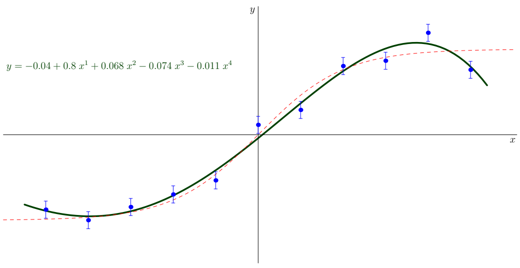
- We train our learning machines on a finite data set
- But we use our learning machines on unseen data
- If we have a too simple machine we might not be able to fit the training data and are unlikely to do well on unseen data



- If we have a too complicated machine we might be able to fit the training data almost perfectly, but we might have learnt a too complex rule that doesn't fit the test set



- Often there is a good compromise so that the learning machine learns a simple rule that fits the training data quite well but isn't too complicated



2.2 Bias-Variance Dilemms

- We assume that we are trying to learn some function $f(\boldsymbol{x})$ where $\boldsymbol{x}$ are feature vectors
- Our task is to learn a function $\hat{f}\,(\boldsymbol{x}|\mathcal{D})$ based on a training set $\mathcal{D}$
- We consider a scenario where we draw different training datasets $\mathcal{D}$ from a distribution of training examples $p(\boldsymbol{x})$
- Each training set contains m independent examples
- We start from the definition of the *mean machine*

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

- The mean machine makes a prediction by averaging the results of machines trained on all possible learning datasets (clearly this is a thought experiment and not something practical)
- Now the **bias** is equal to generalisation performance of mean machine

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

- We consider the expected generalisation loss for a randomly drawn dataset
  - For any particular dataset we might do better or worse than this expected generalisation loss

$$\begin{aligned}
\bar{L}_G &\overset{(1)}{=} \mathbb{E}_{\mathcal{D}}[L_G(\mathcal{D})] \overset{(2)}{=} \mathbb{E}_{\mathcal{D}}\left[\sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})\ \left(\hat{f}(\boldsymbol{x}|\mathcal{D}) - f(\boldsymbol{x})\right)^2\right] \\
&\overset{(3)}{=} \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})\,\mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}|\mathcal{D}) - f(\boldsymbol{x})\right)^2\right] \\
&\overset{(4)}{=} \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})\,\mathbb{E}_{\mathcal{D}}\left[\left(\left(\hat{f}(\boldsymbol{x}|\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right) + \left(\hat{f}_m(\boldsymbol{x}) - f(\boldsymbol{x})\right)\right)^2\right] \\
&\overset{(5)}{=} \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})\Bigg(\mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}|\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)^2 + \left(\hat{f}_m(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right] \\
&\qquad\qquad + 2\,\mathbb{E}_{\mathcal{D}}\left[\left(\hat{f}(\boldsymbol{x}|\mathcal{D}) - \hat{f}_m(\boldsymbol{x})\right)\left(\hat{f}_m(\boldsymbol{x}) - f(\boldsymbol{x})\right)\right]\Bigg)
\end{aligned}$$

    (1) This is the definition of the expected generalisation loss, $\bar{L}_G$
    (2) The generalisation loss is the squared difference between the prediction of the learning machine, $\hat{f}(\boldsymbol{x}|\mathcal{D})$, and the true function, $f(\boldsymbol{x})$, averaged over all possible input feature vectors, $\boldsymbol{x}$, weighted by the probability of the input, $p(\boldsymbol{x})$
    (3) We exchange the sum and expectation
    (4) We add and subtract the prediction of the mean machine
    (5) We expand out the sum

  - The cross term cancels

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

- Note we use the following properties of expectations
  (1) $\mathbb{E}[A+B] = \mathbb{E}[A] + \mathbb{E}[B]$
  (2) $\mathbb{E}[c\,A] = c\,\mathbb{E}[A]$ where c doesn't depend on the random variable you are averaging over
  (3) $\mathbb{E}[1] = 1$
- We are left with

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

- Where we used the fact that the last term doesn't depend on the dataset
- The last term is equal to the bias, defined earlier as the generalisation performance of the mean machine
- The first term is known as the **variance**

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

- It measure how a single learning machine differs from the mean machine
- We therefore have $\bar{L}_G = B + V$ or

$$\text{Expected Generalisation Loss} = \text{Bias} + \text{Variance}$$

- The **Bias-Variance Dilemma** is that
  - Simple machine are likely to have high bias
    * because any single machine can't represent the data well the mean machine won't be accurate
    * this is true of the curve fitting example, but it is not true of decision trees where the average of many decision trees can learn a far more complex division boundary than a single machine
  - Complex machines are likely to have high variance
    * Complex machine are likely to be sensitive to the training data whereas simpler machines (because of their lack of flexibility) aren't as sensitive
- A lot of this course will be looking at machines that cleverly resolve this dilemma

3 Experiments

Download the Jupyter Notebook

- This computes the training and generalisation loss as well as the bias and variance for arbitrary functions (at least approximately)
- We can do this because it is a 1-D function
- See if you can understand the code

3.1 Questions

- What is the effect of increasing the number of training points?
- What is the effect of using a more complex function, E.g. $\mathrm{e}^{-x}\sin(x)$?