Week 13: Hazard of Overfitting

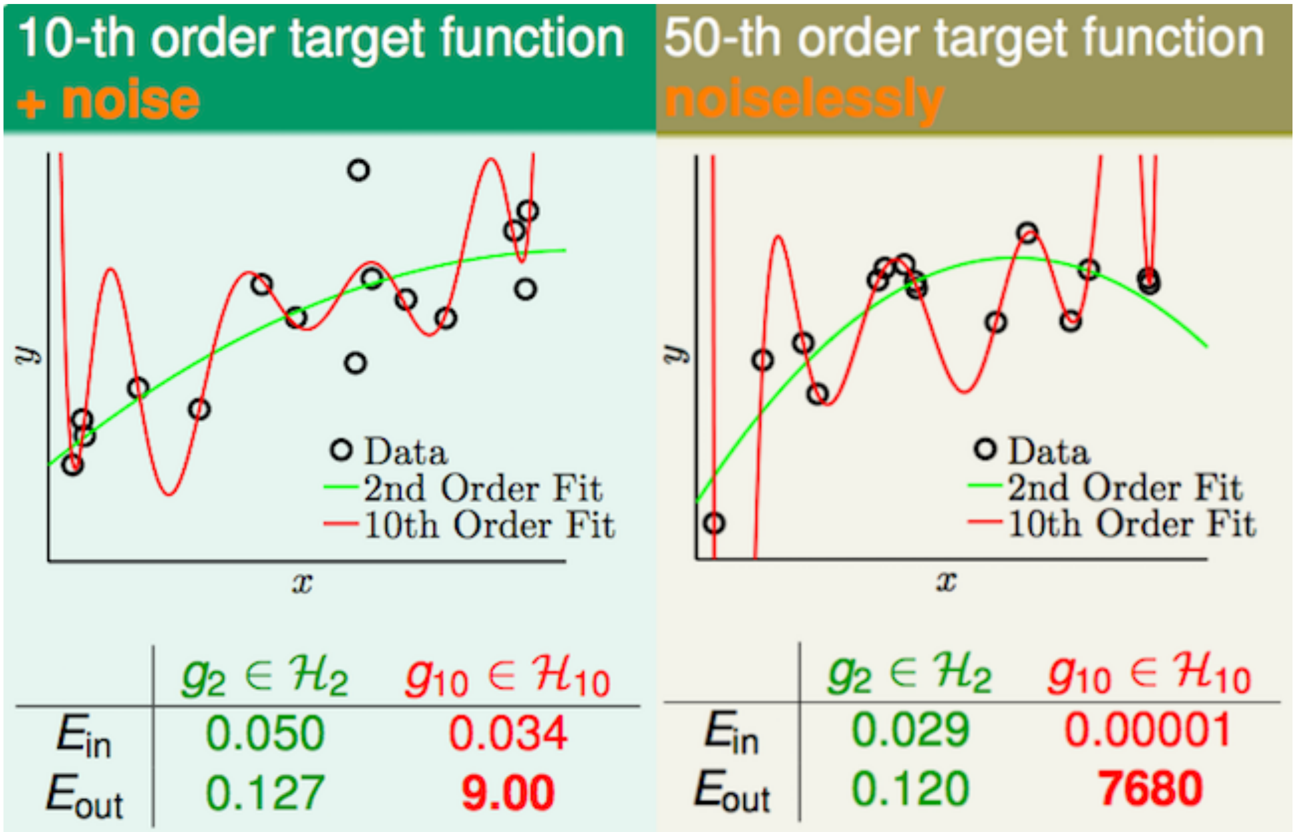
Table of Contents

What is Overfitting?

1. Overfitting
    - Bad generalization (large $E_{out} - E_{in}$)
    - $E_{in} \downarrow, E_{out} \uparrow$
        - Compare to *underfitting*, $E_{in} \uparrow, E_{out} \downarrow$
2. Cause of overfitting
    - Using excessive VC dimension d_{vc} (higher model complexity than needed)
    - Noise in training data
    - Limited data size N

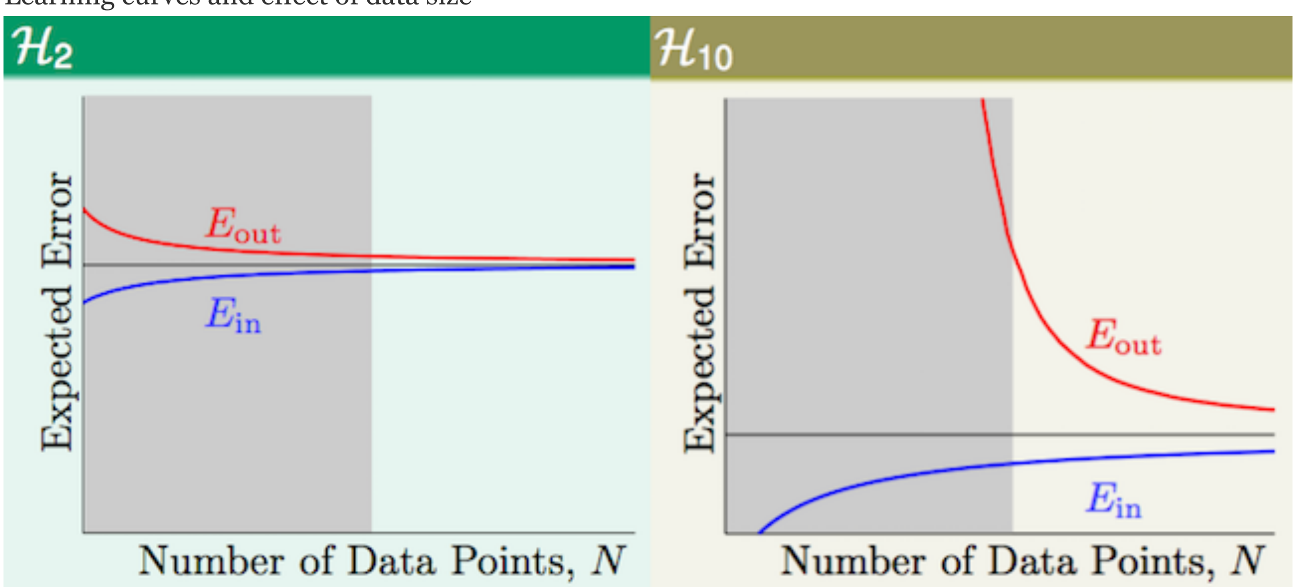
The Role of Noise and Data Size in Overfitting

1. Comparing performance of two models



- Given two hypotheses, one second-order $\mathcal{H}_2$, another 10th-order $\mathcal{H}_{10}$
  - $\mathcal{H}_{10}$ theoretically has more learning power than $\mathcal{H}_2$ and capable of learning more complex models
  - Given two target functions, one 10th-order, another 50th-order
    - $\mathcal{H}_2$ "gives up" ability to fit on both
    - $\mathcal{H}_{10}$ capable of fitting the first target function at full capacity. Also "gives up" ability to fit the second
  - Models learned from $\mathcal{H}_2$ have lower E_{out} for both target functions!

2. Learning curves and effect of data size



- When noise is present in data set, while average out-of-sample error $\bar{E_{out}}$ for $\mathcal{H}_{10}$ decreases as $N \to \infty$, generalization error is much larger for small N
  - $\mathcal{H}_{10}$ always overfits in gray area: $\bar{E_{in}} \downarrow, \bar{E_{out}} \uparrow$
  - Be cautious about using complex model when training data is limited
- In the case of very complex target function (e.g. 50th-order), the additional complexity (unable to be accomodated by hypothesis) acts as noise **even if the training data is noise-free**

Deterministic and Stochastic Noise

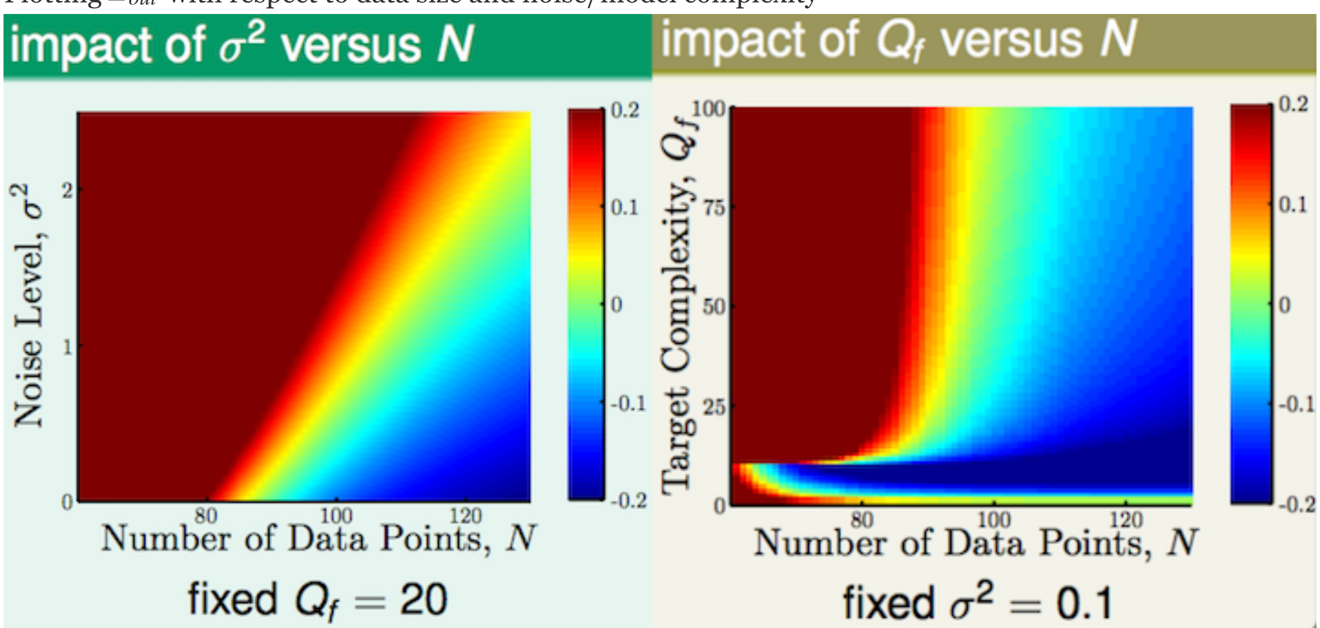
Measure Overfit

1. Given training labels y, target function $f(x)$ and noise ϵ as follows:

$$y = f(x) + \epsilon$$
$$\sim Gaussian(\underbrace{\sum_{q=0}^{Q_f} \alpha_q x^q}_{f(x)}, \sigma^2)$$

- Gaussian iid noise ϵ with level σ^2, on top of target function $f(x)$. The resulting training label y is an *uniform* distribution around $f(x)$ with complexity level Q_f (Q_t-th order target function)
- Data size N

2. Plotting E_{out} with respect to data size and noise/model complexity

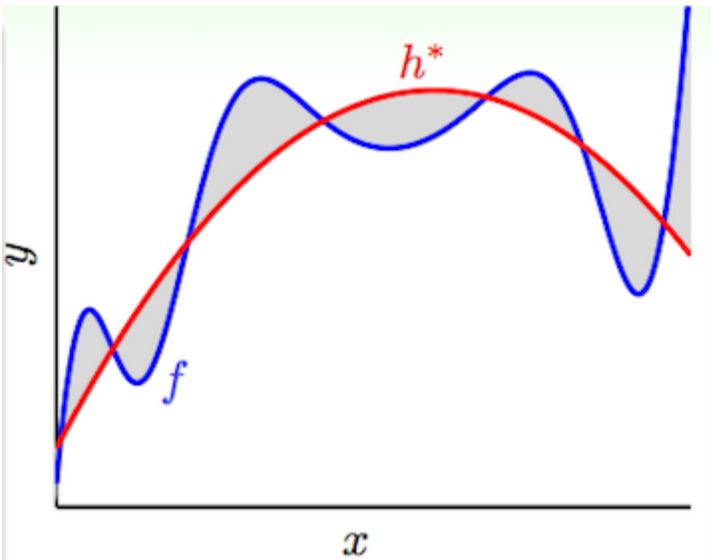


- Color measures extent of overfit: Red (extensive overfitting) → Blue (little overfit)
- **Stochastic noise**: σ^2 as a function of data size N
  - Part of training data. Randomly occurs
- **Deterministic noise**: Q_f as a function of data size N
  - Part of target function. Can be calculated.
  - Recall that *complexity of target function* has similar effect as noise in training data, when the

hypothesis does not possess enough modeling power

3. Four reasons of serious overfitting
    - Small data size (N ↓, overfit ↑)
    - Large stochastic noise (σ^2 ↑, overfit ↑)
    - Large deterministic noise (Q_f ↑, overfit ↑)
    - Excessive fitting power (Order of $\mathcal{H}$ ↑, overfit ↑)
        - Long tail at bottom of Q_f vs N graph
        - When hypothesis has tendency of learning more complicated models than required for target function

More on Deterministic Noise



1. Given hypothesis set $\mathcal{H}$ and target function f, deterministic noise arise when something of f **cannot be captured** by $\mathcal{H}$ ($f \notin \mathcal{H}$)
2. Deterministic noise represents the difference between **best** $h^* \in \mathcal{H}$ and f
3. Deterministic noise acts like 'stochastic noise' when comes to effect on overfitting
4. However, unlike stochastic noise, deterministic noise is **not random**, but
    - Related to the hypothesis set used. Given a target function, more powerful hypothesis set (within order of target function), the smaller deterministic noise is
        - Still subject to overfitting if there's stochastic noise in training data
        - Deterministic noise increases if hypothesis set used is more complex than target function
    - Fixed for a given x
        - Because the hypothesis set and target function remain the same
        - Stochastic noise, on the other hand, is random and might be different when sampling multiple times against the same x

Dealing with Overfitting

1. Ways to account for/couteract overfitting
    - Start from simple model
    - Data cleaning/pruning

        - Correct/remove outliers
    - Data hinting
        - Add *virtual examples* by slightly altering existing samples in training set, based on domain knowledge about the use case or expected learning outcome
        - Added examples are **no longer iid** with respect to $P(x, y)$
            - Otherwise the virtual examples will act as noise and distort learning outcome
    - Regularization
    - Validation