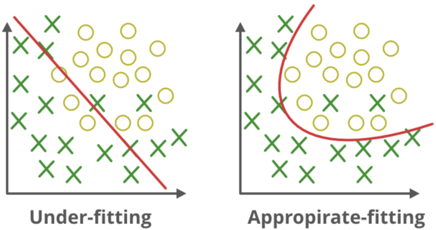
Overfitting: Classification Example

- Assume:
  - You want to separate 2 classes using 2 features x_1, x_2
  - The true class boundary has a parabola shape
- You can use logistic regression and a decision boundary equal to:
  - A line $\text{logit}(w_0 + w_1 x + w_2 y)$
    - Underfit
    - High bias, low variance
  - A parabola $\text{logit}(w_0 + w_1 x + w_2 x^2 + w_3 xy + w_4 y^2)$
    - Right fit
  - A wiggly decision boundary $\text{logit}(w_0 + \text{high powers of } x_1, x_2)$
    - Overfit
    - Low bias, high variance



- ***Bias Variance Analysis***

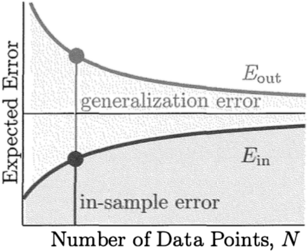
VC Analysis vs Bias-Variance Analysis

- Both VC analysis and bias-variance analysis are concerned with the hypothesis set $\mathcal{H}$
  - **VC analysis**:

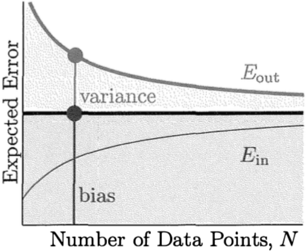
$$E_{out} \leq E_{in} + \Omega(\mathcal{H})$$

  - **Bias-variance analysis**

$$E_{out} = \text{bias} + \text{variance}$$



VC Analysis



Bias-Variance Analysis

Hypothesis Set and Bias-Variance Analysis

- **Learning** consists in finding $g \in \mathcal{H}$ such that $g \approx f$ where f is an unknown function
- The **tradeoff in learning** is between:
  - Bias vs variance
  - Overfitting vs underfitting
  - More complex vs less complex $\mathcal{H}$ / h
  - Approximation (in-sample) vs generalization (out-of-sample)

Decomposing Error in Bias-Variance (1/4)

- **Problem**
  - Regression set-up: target is a real-valued function
  - Hypothesis set $\mathcal{H} = \{h_1(\underline{\boldsymbol{x}}), h_2(\underline{\boldsymbol{x}}), ...h_n(\underline{\boldsymbol{x}})\}$
  - Training data $\mathcal{D}$ with N examples
  - Squared error $E_out = \mathbb{E}[(g(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2]$
  - Choose the best function $g \in \mathcal{H}$ that approximates unknown f
- **Question**
  - What is the out-of-sample error $E_{out}(g)$ as function of $\mathcal{H}$ for a training set of N examples?

SCIENCE ACADEMY

Decomposing Error in Bias-Variance (2/4)

- The final hypothesis g depends on training set D, so make the dependency explicit $g^{(D)}$:

$$E_{out}(g^{(D)}) \triangleq \mathbb{E}_{\underline{\boldsymbol{x}}}[(g^{(D)}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2]$$

- Interested in:
  - Hypothesis set $\mathcal{H}$ rather than specific h
  - Training set D of N examples, rather than a specific D
- Remove dependency from D by averaging over all possible training sets D with N examples:

$$E_{out}(\mathcal{H}) \triangleq \mathbb{E}_D[E_{out}(g^{(D)})] = \mathbb{E}_D[\mathbb{E}_{\underline{\boldsymbol{x}}}[(g^{(D)}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2]]$$

Decomposing Error in Bias-Variance (3/4)

- Switch the order of the expectations since the quantity is non-negative:

$$E_{out}(\mathcal{H}) = \mathbb{E}_{\underline{\boldsymbol{x}}}[\mathbb{E}_D[(g^{(D)}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2]$$

- Focus on $\mathbb{E}_D[(g^{(D)}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2]$ which is a function of $\underline{\boldsymbol{x}}$
- Define the *average hypothesis* over all training sets as:

$$\overline{g}(\underline{\boldsymbol{x}}) \triangleq \mathbb{E}_D[g^{(D)}(\underline{\boldsymbol{x}})]$$

- Add and subtract it inside the $\mathbb{E}_D$ expression:

$$\begin{aligned}
E_{out}(\mathcal{H}) =& \mathbb{E}_{\underline{\boldsymbol{x}}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}})\right)^2\right]\right] \\
=& \mathbb{E}_{\underline{\boldsymbol{x}}}\mathbb{E}_D[(g^{(D)} - \overline{g} + \overline{g} - f)^2] \\
=& \mathbb{E}_{\underline{\boldsymbol{x}}}\mathbb{E}_D[(g^{(D)} - \overline{g})^2 + (\overline{g} - f)^2 + 2(g^{(D)} - \overline{g})(\overline{g} - f)] \\
& (\mathbb{E}_D \text{ is linear and } (\overline{g} - f) \text{ doesn't depend on } D) \\
=& \mathbb{E}_{\underline{\boldsymbol{x}}}\left[\mathbb{E}_D[(g^{(D)} - \overline{g})^2] + (\overline{g} - f)^2 + 2\mathbb{E}_D[(g^{(D)} - \overline{g})](\overline{g} - f)\right]
\end{aligned}$$

Decomposing Error in Bias-Variance (4/4)

- The cross term:

$$\mathbb{E}_D[(g^{(D)} - \overline{g})](\overline{g} - f)$$

disappears since applying the expectation on D, it is equal to:

$$(g^{(D)} - \mathbb{E}_D[\overline{g}])(\overline{g} - f) = 0 \cdot (\overline{g} - f) = 0 \cdot \text{constant}$$

- Finally:

$$\begin{aligned} E_{out}(\mathcal{H}) &= \mathbb{E}_{\underline{\boldsymbol{x}}}[\mathbb{E}_D[(g^{(D)} - \overline{g})^2] + (\overline{g}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2] \\ &= \mathbb{E}_{\underline{\boldsymbol{x}}}[\mathbb{E}_D[(g^{(D)} - \overline{g})^2]] + \mathbb{E}_{\underline{\boldsymbol{x}}}[(\overline{g} - f)^2] \quad (\mathbb{E}_{\underline{\boldsymbol{x}}} \text{ is linear}) \\ &= \mathbb{E}_{\underline{\boldsymbol{x}}}[\text{var}(\underline{\boldsymbol{x}})] + \mathbb{E}_{\underline{\boldsymbol{x}}}[\text{bias}(\underline{\boldsymbol{x}})^2] \\ &= \text{variance} + \text{bias} \end{aligned}$$

Interpretation of Average Hypothesis

- The **average hypothesis** over all training sets

$$\overline{g}(\underline{\boldsymbol{x}}) \triangleq \mathbb{E}_D[g^{(D)}(\underline{\boldsymbol{x}})]$$

  can be interpreted as the “best” hypothesis from $\mathcal{H}$ training on N samples
  - Note: $\overline{g}$ is not necessarily $\in \mathcal{H}$
- In fact it's like **ensemble learning**:
  - Consider all the possible data sets D with N samples
  - Learn g from each D
  - Average all the hypotheses

Interpretation of Variance and Bias Terms

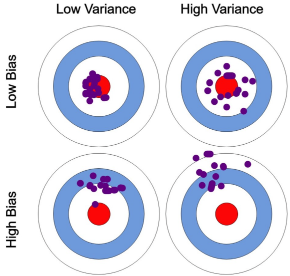
- The out-of-sample error can be decomposed as:

$$E_{out}(\mathcal{H}) = \text{bias}^2 + \text{variance}$$

- **Bias term**

$$\text{bias}^2 = \mathbb{E}_{\underline{\boldsymbol{x}}}[(\overline{g}(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}}))^2]$$

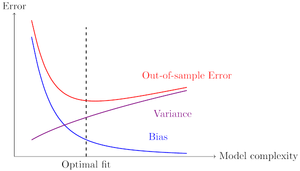
  - Does not depend on learning as it is not a function of the data set D
  - Measures how limited $\mathcal{H}$ is
    - I.e., the ability of $\mathcal{H}$ to approximate the target with infinite training sets



- **Variance term**

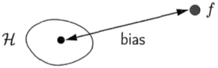
$$\text{variance} = \mathbb{E}_{\underline{\boldsymbol{x}}}\mathbb{E}_D[(g^{(D)}(\underline{\boldsymbol{x}}) - \overline{g}(\underline{\boldsymbol{x}}))^2]$$

  - Measures variability of the learned hypothesis from D for any $\underline{\boldsymbol{x}}$
    - With infinite training sets, we could focus on the "best" g, which is $\overline{g}$
    - But we have only one data set D at a time, incurring a cost

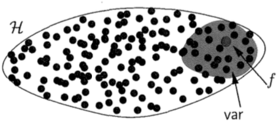


Variance and Bias Term Varying Cardinality of $\mathcal{H}$

- If hypothesis set **has a single function**:
  $\mathcal{H} = \{h \neq f\}$
  - Large bias
    - h might be far from f
  - Variance $= 0$
    - No cost in choosing hypothesis



- If hypothesis set **has many functions**:
  $\mathcal{H} = \{\text{many hypotheses } h\}$
  - Bias can be 0
    - E.g., if $f \in \mathcal{H}$
  - Large variance
    - Depending on data set D, end up far from f
    - Larger $\mathcal{H}$, farther g from f



Bias-Variance Trade-Off: Numerical Example

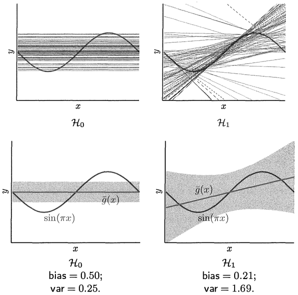
- **Machine learning problem**:
  - Target function $f(x) = \sin(\pi x), x \in [-1, 1]$
  - Noiseless target
  - You have $f(\underline{\boldsymbol{x}})$ for $N = 2$ points
- **Two hypotheses sets $\mathcal{H}$**:
  - Constant model: $\mathcal{H}_0 : h(x) = b$
  - Linear model: $\mathcal{H}_1 : h(x) = ax + b$
- **Which model is best**?
  - Depends on the perspective!
  - Best for *approximation*: minimal error approximating the sinusoid
  - Best for *learning*: learn the unknown function with minimal error from 2 points

Bias-Variance Trade-Off: Numerical Example

- **Approximation**
  - $E_{out}(g_0) = 0.5$
    - g_0 is a constant and approximates the sinusoid poorly (higher bias)
  - $E_{out}(g_1) = 0.2$
    - g_1 is a line and has more degrees of freedom (lower bias)
  - The line model *approximates better* than the constant model
- **Learning**
  - Algorithm:
    - Pick 2 points as training set D
    - Learn g from D
    - Different D gives different g
    - Compute $\mathbb{E}_D[E_{out}(g)]$
  - Average over all data sets D:

$$E_{out} = \text{bias}^2 + \text{variance}$$

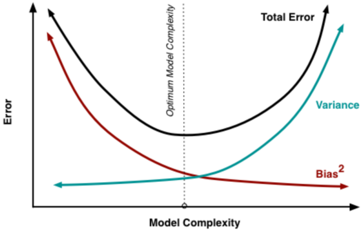
  - $E_{out}(g_0) = 0.5 + 0.25 = 0.75$
    - g_0 is more stable from the data set (lower variance)
  - $E_{out}(g_1) = 0.2 + 1.69 = 1.9$
    - g_1 heavily depends on the training set (higher variance)
  - The constant model *learns better* than



SCIENCE ACADEMY

Bias-Variance Curves

- Bias-variance curve are plots of E_{out} increasing the complexity of the model
  - Can diagnose bias-variance problem
- Typical form of bias-variance curves
- E_{in} and E_{out} start from the same point
- E_{in}
  - Is decreasing with increasing model complexity
  - Can even go to 0
  - Is shaped like an hyperbole
- E_{out}
  - Is always larger than E_{in}
  - Is the sum of bias and variance
  - Has a bowl shape
  - Reaches a minimum for optimal fit
  - Before the minimum there is a "high bias / underfitting" regime



How to Measure the Model Complexity

- Number of features
- Parameters for model form / degrees of freedom, e.g.,
  - VC dimension d_{VC}
  - Degree of polynomials
  - k in KNN
  - ν in NuSVM
- Regularization param λ
- Training epochs for neural network

Bias-Variance Curves and Regularization

- We can use a complex model together with regularization to learn at the same time:
  - The model coefficients $\underline{\boldsymbol{w}}$
  - The model "complexity" (e.g., VC dimension), which is related to the regularization parameter λ
- For each different values of $\lambda = \{10^{-1}, 1.0, 10\}$ we optimize:

$$\underline{\boldsymbol{w}}\lambda = \text{argmin}_{\underline{\boldsymbol{w}}} E_{aug}(\underline{\boldsymbol{w}}) = E_{in}(\underline{\boldsymbol{w}}) + \Omega(\lambda)$$

  - $\underline{\boldsymbol{w}}(\lambda)$ is the optimal model as function of λ
- Then estimate E_{out} using $\underline{\boldsymbol{w}}(\lambda)$ and λ
  - Small λ means
    - Complex model (with respect to data)
    - Low bias
    - High variance
  - Large λ means
    - Simple model
    - High bias
    - Low variance
- There will be an intermediate value of λ that optimizes the trade-off between bias and variance

Bias-Variance Decomposition with a Noisy Target

- We can extend the bias-variance decomposition to the noisy target

$$y = \underline{\boldsymbol{w}}^T \underline{\boldsymbol{x}} + \varepsilon$$

- With similar hypothesis and a similar analysis we conclude that:

$$\begin{aligned} E_{out}(\mathcal{H}) &= \mathbb{E}_{D,\underline{\boldsymbol{x}}}\left[(g^{(D)} - \overline{g})^2\right] + \mathbb{E}_{\underline{\boldsymbol{x}}}\left[(\overline{g} - f)^2\right] + \mathbb{E}_{\varepsilon,\underline{\boldsymbol{x}}}\left[(f - y)^2\right] \\ &= \text{variance} + \text{bias } (= \text{deterministic noise}) + \text{stochastic noise} \end{aligned}$$

- **Interpretation**:
  - The error is the sum of 3 contributions
    1. Variance: from the set of hypotheses to the centroid of the hypothesis set
    2. Bias: from the centroid of the hypothesis set to the noiseless function
    3. Noise: from the noiseless function to the real function

Bias as Deterministic Noise

- The bias term can be interpreted as "deterministic noise"
  - Bias is the part of the target function that our hypothesis set cannot capture:
    $$h^*(\underline{\boldsymbol{x}}) - f(\underline{\boldsymbol{x}})$$
    where
    - $h^*()$ is the best approximation of $f(\underline{\boldsymbol{x}})$ in the hypothesis set $\mathcal{H}$
    - E.g., $\overline{g}(x)$
- The hypothesis set $\mathcal{H}$ cannot learn the deterministic noise since it is outside of its ability, and thus it behaves like noise

Deterministic vs Stochastic Noise in Practice

- In bias-variance analysis, the error for a noisy target is decomposed into:
  - Bias (deterministic noise)
  - Variance
  - Stochastic noise
- **Deterministic noise:**
  - Fixed for a particular $\underline{\boldsymbol{x}}$
  - Depends on $\mathcal{H}$
  - Independent of ε or D
- **Stochastic noise:**
  - Not fixed for $\underline{\boldsymbol{x}}$
  - Independent of D or $\mathcal{H}$
- In an actual machine learning problem, there's no difference between stochastic and deterministic noise, since $\mathcal{H}$ and D are fixed
  - E.g., from the training set alone, we cannot tell if the data is from a *noiseless complex* target or a *noisy simple* target

Deterministic vs Stochastic Noise Example

- 2 targets:
  - Noisy low-order target (5-th order polynomial)
  - Noiseless high-order target (50-th order polynomial)
  - Generate $N = 15$ data points from them
- 2 models:
  - $\mathcal{H}_2$ low-order hypothesis (2nd order polynomial)
  - $\mathcal{H}_{10}$ high-order hypothesis (10-th order polynomial)
- When learning a model there is no difference between deterministic and stochastic noise
- In fact the learning algorithm only sees the samples in the training set and one cannot distinguish the two different sources
- For noisy low-order target: going from fitting the 2nd order to the 10-th order polynomial we see that $\downarrow E_{in}$ (we have more degrees of freedoms) and $\uparrow\uparrow E_{out}$ (since the 10-th polynomial fits the noise)
- For noiseless high-order target: exactly the same phenomenon!
- Knowing that the target is a 10-th order polynomial, one can think that

Amount of Data and Model Complexity

- The lesson learned from bias-variance analysis is that one must match the *model complexity*:
  - To the *data resources*
  - To the *signal to noise ratio*
  - **Not** to the *target complexity*
- The rule of thumb is:

$$d_{VC}(\text{degrees of freedom of the model}) = N(\text{number of data points})/10$$

  - In other words, 10 data points needed to fit a degree of freedom
  - If the data is noisy, you need even more data

SCIENCE ACADEMY