Bias and Variance Decomposition
What we are doing next

- In this and the following few lectures, we will look into some theoretical analysis of learning.
- Such analysis will help us build stronger intuition and develop rule of thumb about how to best apply learning algorithms in different settings.
- First, we will look at the linear regression problem.
Analyzing the Expected Loss

- We assume that we observe \((x, y)\) pairs that are generated from a fixed distribution \(p(x, y)\).
- Let \(\hat{y}(x)\) be our estimation of the true value \(y\).
- Expected loss is then written as:

\[
E[L] = \int \int L(y, \hat{y}(x)) p(x, y) dx dy = \int \int (y - \hat{y}(x))^2 p(x, y) dx dy
\]

\[
= \int \int (y - E[y|x] + E[y|x] - \hat{y}(x))^2 p(x, y) dx dy
\]

\[
= \int \int (y - E[y|x])^2 p(x, y) dx dy + \int \int (E[y|x] - \hat{y}(x))^2 p(x, y) dx dy
\]

\[
+ 2 \int \int (y - E[y|x])(E[y|x] - \hat{y}(x)) p(x, y) dy dx
\]

\[
= \int \left[ \int (y - E[y|x]) c(x)p(x, y) dy \right] dx
\]
The Bias-Variance Decomposition (1)

- Consider the expected squared loss,

\[ E[L] = \iint (E[y|x] - \hat{y}(x))^2 p(x, y) dx dy + \iint (y - E[y|x])^2 p(x, y) dx dy \]

- The second term of E[L] corresponds to the noise inherent in the target variable y.

- This is part of the data, we have no control over it.

- What about the first term?
The Bias-Variance Decomposition (2)

- The first term in $E[L]$ depends on our choice for $\hat{y}$, which is to be minimized
- If we have infinite training data, we could in principle find the regression function $\hat{y}$ that approximate $E(y|x)$ to arbitrary accuracy
- But for a finite training set, we cannot achieve that
- Given a training set size $N$, the learned function $y$ will depend on the specific training set $D$ we use – thus we can denote it as $\hat{y}(x; D)$
- We are then interested in how well we can do in expectation over $D$
  \[ E_D[(E[y|x] - \hat{y}(x; D))^2] \]
- Below we will use $t(x)$ to denote $E(y|x)$ for simplicity
The Bias-Variance Decomposition (3)

\[(t(x) - \hat{y}(x; D))^2\]

\[= (t(x) - E_D[\hat{y}(x; D)] + E_D[\hat{y}(x; D)] - \hat{y}(x; D))^2\]

\[= (t(x) - E_D[\hat{y}(x; D)])^2 + (E_D[\hat{y}(x; D)] - \hat{y}(x; D))^2\]

\[+ 2(t(x) - E_D[\hat{y}(x; D)])(E_D[\hat{y}(x; D)] - \hat{y}(x; D))\]

Taking expectation over \(D\), the last term goes to 0:

\[E_D \left[ (t(x) - \hat{y}(x; D))^2 \right] \]

\[= (t(x) - E_D[\hat{y}(x; D)])^2 + E_D[(E_D[\hat{y}(x; D)] - \hat{y}(x; D))^2]\]

\[(\text{bias})^2 + \text{variance}\]
The Bias-Variance Decomposition (4)

• Thus we can write

\[ E(L) = \int (t(x) - E_D[\hat{y}(x; D)])^2 p(x)dx \]  
  Bias

\[ + \int E_D[(E_D[\hat{y}(x; D)] - \hat{y}(x; D))^2]p(x)dx \]  
  Variance

\[ + \iint (y - E[y|x])^2 p(x, y)dxdy \]  
  noise

expected loss = (bias)^2 + variance + noise

Bias: how well on average can our learning algorithm capture the target function

Variance: how significant does our learning algorithm fluctuate depending on the training set

Noise: inherent to the data generation process itself, not controlled by the choice of learning algorithm
The Bias-Variance Decomposition (5)

- Example: 25 training data sets from the sinusoidal, varying the degree of regularization, $\lambda$.
- The model uses 24 Gaussian Basis functions
The Bias-Variance Decomposition (6)

$$\ln \lambda = -0.31$$

Variance ↑

Bias ↓

model complexity ↑
The Bias-Variance Decomposition (7)

model complexity ↑↑

Variance ↑↑

Bias ↓↓
The Bias-Variance Trade-off

From these plots, we note that an over-regularized model (large $\lambda$) will have a high bias, while an under-regularized model (small $\lambda$) will have a high variance.
Summary

• Expected error can be decomposed into three components: squared bias, variance and noise
• Complex models lead to lower bias but higher variance
• Simple models lead to high bias but low variance
• Model selection is needed to trade-off the bias and variance