Learning to Generalize

AW

Lecture Overview

1. **Bias-Variance Tradeoff Informally**

2. **Bias-Variance Tradeoff Formally**

Generalization (again)

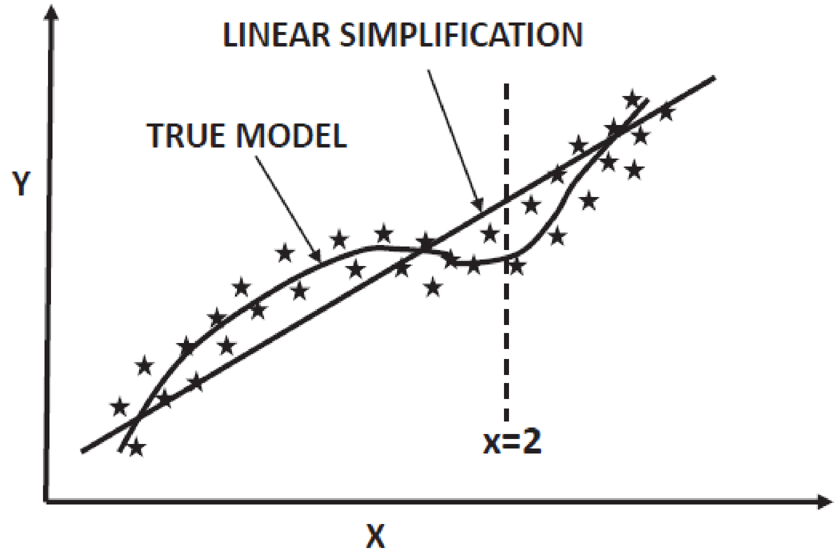
- In a data mining problem, we try to generalize the known dependent variable on seen instances to unseen instances.
  - Unseen $\Rightarrow$ The model did not see it during training.
- Generalization= given training images with seen labels, try to label an unseen image.
- The classification accuracy on instances used to train a model is usually higher than on unseen instances.
- We only care about the accuracy on unseen data.

Generalization – Reasons for Overfitting

- Why is the accuracy on seen data higher?
  - Trained model remembers some of the irrelevant nuances.
- When is the gap between seen and unseen accuracy likely to be high?
  - When the amount of data is limited.
  - When the model is complex (which has higher *capacity* to remember nuances).
  - The combination of the two is a deadly cocktail.
- A high accuracy gap between the predictions on seen and unseen data is referred to as *overfitting*.

Data and Models

Example: True data and 2 models- polynomial and linear



First impression: Polynomial model such as

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

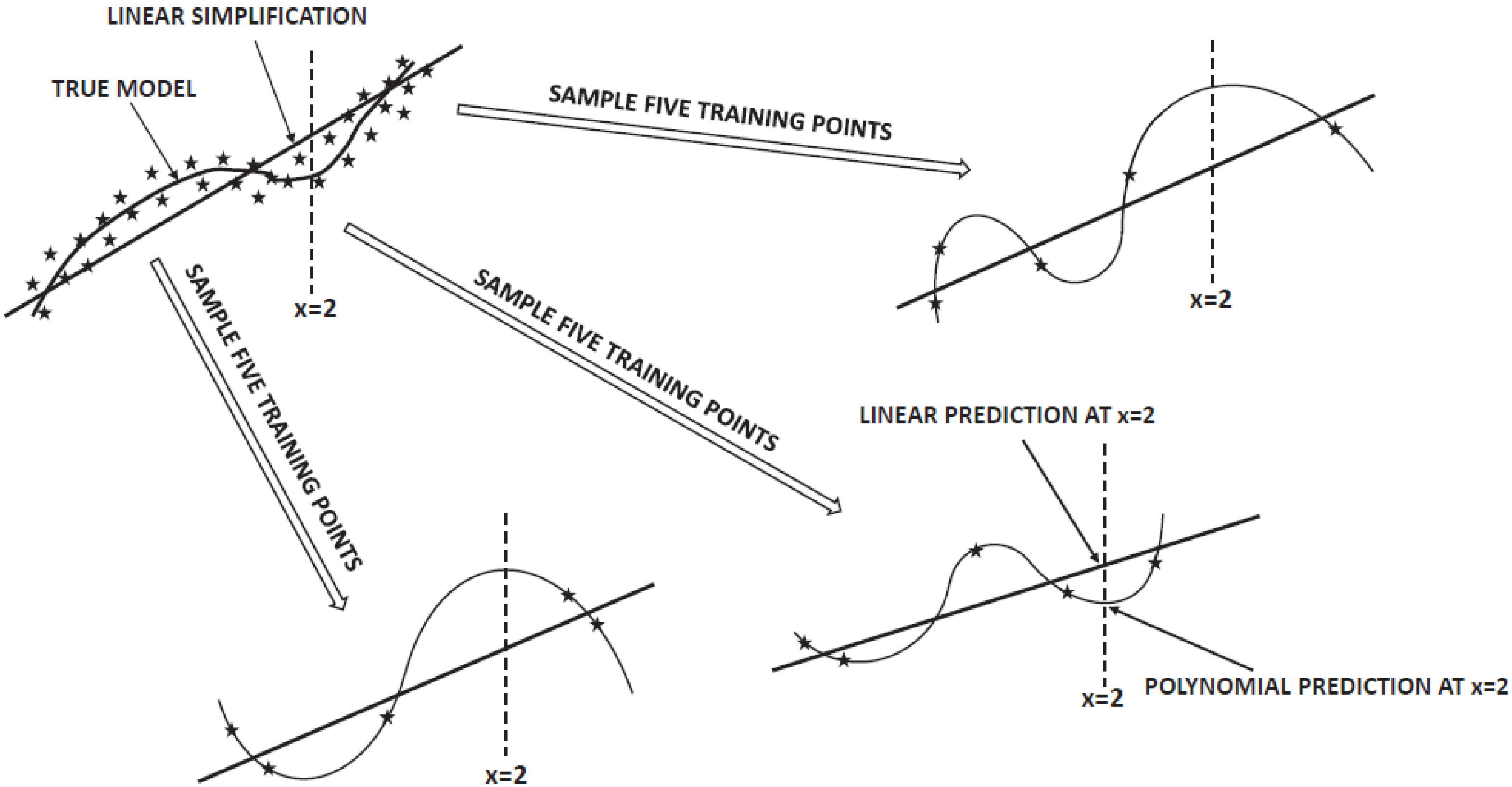
is “better” than linear model

$$y = w_0 + w_1 x$$

Bias-variance trade-off says: “Not necessarily! Are you looking at all data or just some sample?”

Data and Models (cont.)

Example: Polynomial model vs linear



Second look: Zero error on training data, but wildly varying predictions of $x = 2$

Bias and Variance -Intuition

Intuitively *bias* is an error caused by the simplifying assumptions built into the learning method

- The higher-order model is more complex than the linear model and has less bias.
  - But it has more parameters.
  - For a small training data set, the learned parameters will be more sensitive to the nuances of that data set.
  - Different training data sets will provide different predictions for y at a particular x.
  - This variation is referred to as model *variance*.

Intuitively variance how much the learning method will move around its mean

- Neural networks are inherently low-bias and high-variance learners $\Rightarrow$ Need ways of handling complexity.

Assumptions About Data

- The relationship between the dependent variable y and its feature representation $\vec{x}$ is given by some unknown function and additive noise ϵ
- The true model is $y = f(\vec{x})$ where $y \in Y, x \in X$ are range and domain of the function which needs to be learned. Noise refers to unexplained variations of data from true model so that data is given by $y_i = f(\vec{x}_i) + \epsilon_i$ for data points $(x_i, y_i) \in D \subseteq X \times Y,\ i \in \{1, \dots, N\}$. Noise is a property of the *data* rather than model.
- Noise is a random variable with 0 expectation, so the model $y = f(\vec{x})$ together with noise defined true distribution B on $X \times Y$
- Real-world noise examples:
  - Human mislabeling of test instance $\Rightarrow$ Ideal model will never predict it accurately.
  - Error during collection of temperature due to sensor malfunctioning.
- Cannot do anything about it even if seeded with knowledge about true model.

Bias-Variance Imaginable Experiment

- Imagine you are given the true distribution B of training data (including labels).
- You have a principled way of sampling data sets $B \sim \mathcal{D}$ from the training distribution.
- Imagine you create an infinite number of training data sets $\mathcal{D}_i$ (and trained models $A(\mathcal{D}_i)$) by repeated sampling.
- You have a *fixed* set $\mathcal{T}$ of unlabeled test instances.
  - The test set $\mathcal{T}$ does not change over different training data sets.
- Compute prediction $g(x_j, A(\mathcal{D}_i))$ of each instance x_j in $\mathcal{T}$ for each trained model $A(\mathcal{D}_i)$.

Informal Definition of Bias

- Compute averaged prediction of each test instance x over different training models $G_A(x)$(something like $\lim_{n\to\infty} \frac{1}{n} \sum_i g(x, A(\mathcal{D}_i)))$
- Averaged prediction of test instance will be different from true (unknown) model $f(x)$. Difference between (averaged) $G_A(x)$ and $f(x)$ caused by erroneous assumptions/simplifications in modeling $\Rightarrow$ Bias
- High bias = underfitting!

Example (cont.): Linear simplification to polynomial model causes bias.

- If the true (unknown) model $f(x)$ were an order-4 polynomial, and we used any polynomial of order-4 or greater in $G_A(x)$ bias would be 0.

Informal Definition of Variance

- The value $g(x, A(\mathcal{D}_i))$ will vary with $\mathcal{D}_i$ for fixed x.
  - The prediction of the same test instance will be different over different trained models.
- Variance of $g(x, A(\mathcal{D}_i))$ over different training data sets $\mathcal{D}_i \Rightarrow$ *Model Variance*
- All these predictions cannot be simultaneously correct $\Rightarrow$ *v*ariation contributes to error
- High variance = overfitting!

Example (cont.): Linear model will have low model variance.

- Higher-order model will have high variance.

Lecture Overview

1. Bias-Variance Tradeoff Informally

1. Bias-Variance Tradeoff Formally

Formal Bias-Variance Equation for MSE

$MSE(D, A(\mathcal{D})) = \frac{1}{t}\sum_i (\hat{y}_i - y_i)^2 = \frac{1}{t}\sum_i \left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i) - \epsilon_i\right)^2$ so

$$E_{\mathcal{D}\sim B}(MSE(D, A(\mathcal{D}))) = E\left(\frac{1}{t}\sum_i \left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i) - \epsilon_i\right)^2\right)$$

$$= \frac{1}{t}\sum_i E\left(\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i) - \epsilon_i\right)^2\right)$$

$$= \frac{1}{t}\sum_i E\left(\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i)\right)^2 - 2\epsilon_i\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i)\right) + \epsilon_i^2\right)$$

$$= \frac{1}{t}\sum_i E\left(\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i)\right)^2\right) - E\left(2\epsilon_i\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i)\right)\right) + E(\epsilon_i^2)$$

Since ϵ_i is independent of $g\left(\vec{x}_i, A(\mathcal{D})\right)$ which is determined by B we have $E\left(2\epsilon_i\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i)\right)\right) = 2E(\epsilon_i)E\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i)\right) = 0$ as $E(\epsilon_i) = 0$ which is our assumption, so

$$E_{D\sim B}\left(MSE\left(D, A(\mathcal{D})\right)\right) = \frac{1}{t}\sum_i E\left(\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f\left(\vec{x}_i\right)\right)^2\right) + \frac{1}{\mathrm{t}}\sum_i E(\epsilon_i^2)$$

Formal Bias-Variance Equation for MSE (cont.)

$$E_{D\sim B}\left(MSE\big(D, A(\mathcal{D})\big)\right) = E\left(\frac{1}{t}\sum_i(\hat{y}_i - y_i)^2\right) = E\left(\frac{1}{t}\sum_i\big(g\big(\vec{x}_i, A(\mathcal{D})\big) - f(\vec{x}_i) - \epsilon_i\big)^2\right)$$

$$= \frac{1}{t}\sum_i E\left(\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - f(\vec{x}_i)\right)^2\right) + \frac{1}{\mathrm{t}}\sum_i E(\epsilon_i^2)$$

$$= \frac{1}{t}\sum_i E\left(\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right) + E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right) - f(\vec{x}_i)\right)^2\right) + \frac{\sum_i E(\epsilon_i^2)}{\mathrm{t}}$$

$$= \frac{1}{t}\sum_i E\left(\left(f(\vec{x}_i) - E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right)\right)^2\right) + \frac{1}{t}\sum_i E\left(\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right)\right)^2\right)$$

$$-\frac{2}{t}\sum_i E\left[\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - f(\vec{x}_i)\right)\cdot\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right)\right)\right] + \frac{\sum_i E(\epsilon_i^2)}{\mathrm{t}}$$

Clearly terms of the product in the third term (prediction error and prediction difference from its expectation) are independent so

$$E\left[\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - f(\vec{x}_i)\right)\cdot\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right)\right)\right] =$$

$$E\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - f(\vec{x}_i)\right)\cdot E\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - E\left(g\big(\vec{x}_i, A(\mathcal{D})\big)\right)\right)$$

$$= E\left(g\big(\vec{x}_i, A(\mathcal{D})\big) - f(\vec{x}_i)\right)\cdot 0 = 0$$

Formal Bias-Variance Equation for MSE (cont.)

So $E\left(MSE\left(D, A(\mathcal{D})\right)\right) = \frac{1}{t}\sum_i(\hat{y}_i - y_i)^2 = \frac{1}{t}\sum_i\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - f(\vec{x}_i) - \epsilon_i\right)^2$

$$= \underbrace{\frac{1}{t}\sum_i\left(f(\vec{x}_i) - E\left(g\left(\vec{x}_i, A(\mathcal{D})\right)\right)\right)^2}_{bias^2}$$

$$+ \underbrace{\frac{1}{t}\sum_i E\left(\left(g\left(\vec{x}_i, A(\mathcal{D})\right) - E\left(g\left(\vec{x}_i, A(\mathcal{D})\right)\right)\right)^2\right)}_{variance}$$

$$+ \underbrace{\frac{\sum_i E(\epsilon_i^2)}{\mathrm{t}}}_{noise}$$

Bias-Variance Equation Interpretation

- $E[MSE]$ is the expected mean-squared error of the fixed set of test instances over different samples of training data sets.

  $E[MSE]$ = Bias^2 + Variance +Noise

  - In linear models, the bias component will contribute more to $E[MSE]$.
  - In polynomial models, the variance component will contribute more to $E[MSE]$.
- We have a trade-off, when it comes to choosing model complexity!

Main Lessons form Bias-Variance Analysis

- A model with greater complexity might be *theoretically* more accurate (i.e., low bias).
  - But you have less control on what it might predict on a small training data set.
  - Different training data sets will result in widely *varying* predictions of same test instance.
  - Some of these must be wrong $\Rightarrow$ Contribution of model variance.
- *A more accurate model for infinite data is not a more accurate model for finite data.*
  - Do not use a sledgehammer to swat a fly!

Reading

Sections 4.10.3

ZM -22.3 (prior to 22.3.1.)