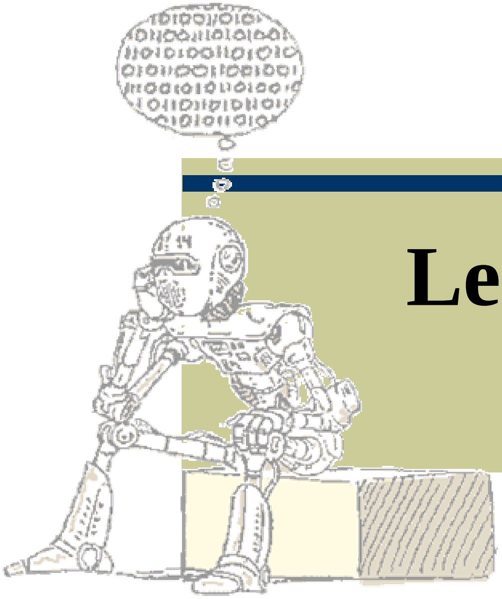


Lecture IV: Learning for Control

- Inference with Data

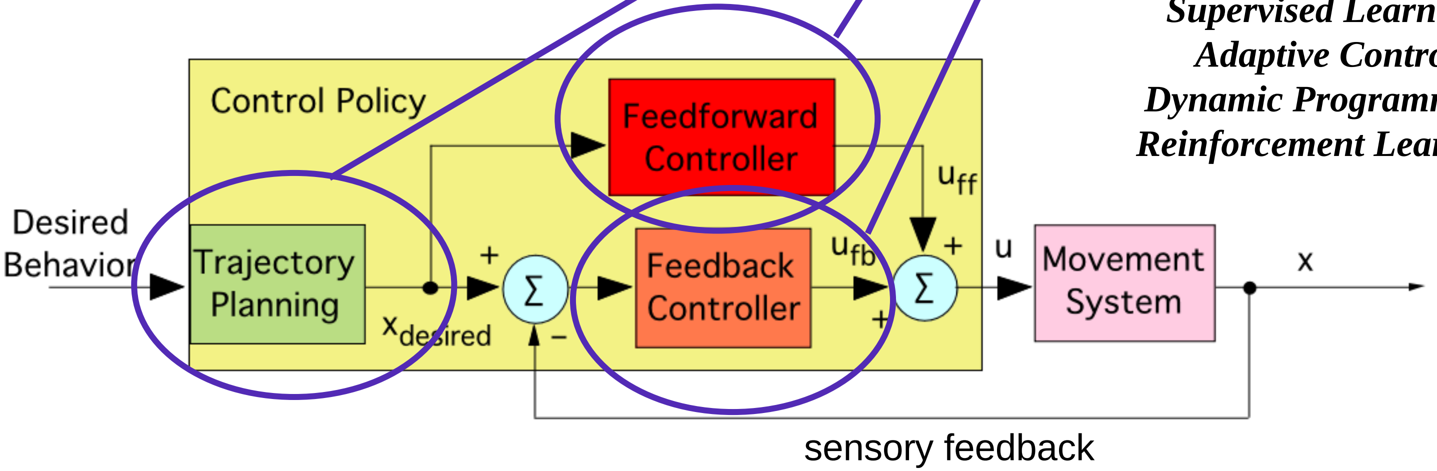
Overview

- Internal models and function approximation
- Cost Function & Optimization
- Generalization, Overfitting & Bias-Variance Dilemma

Function Approximation for control

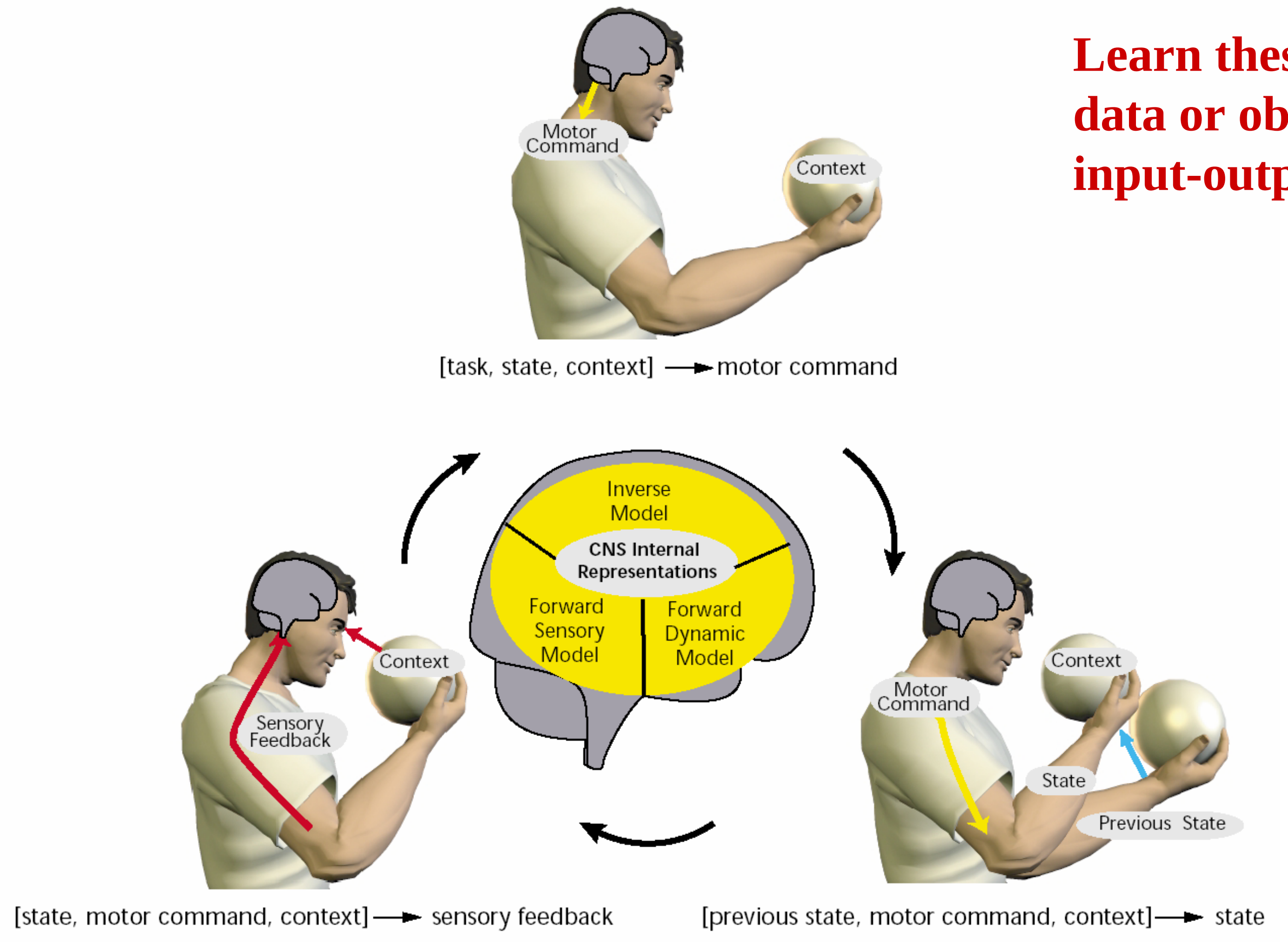
$$\tau = f(\theta, \dot{\theta}, \ddot{\theta})$$

Learning Internal Models or Control Policies is essentially performing function approximation



Supervised Learning
Adaptive Control
Dynamic Programming
Reinforcement Learning

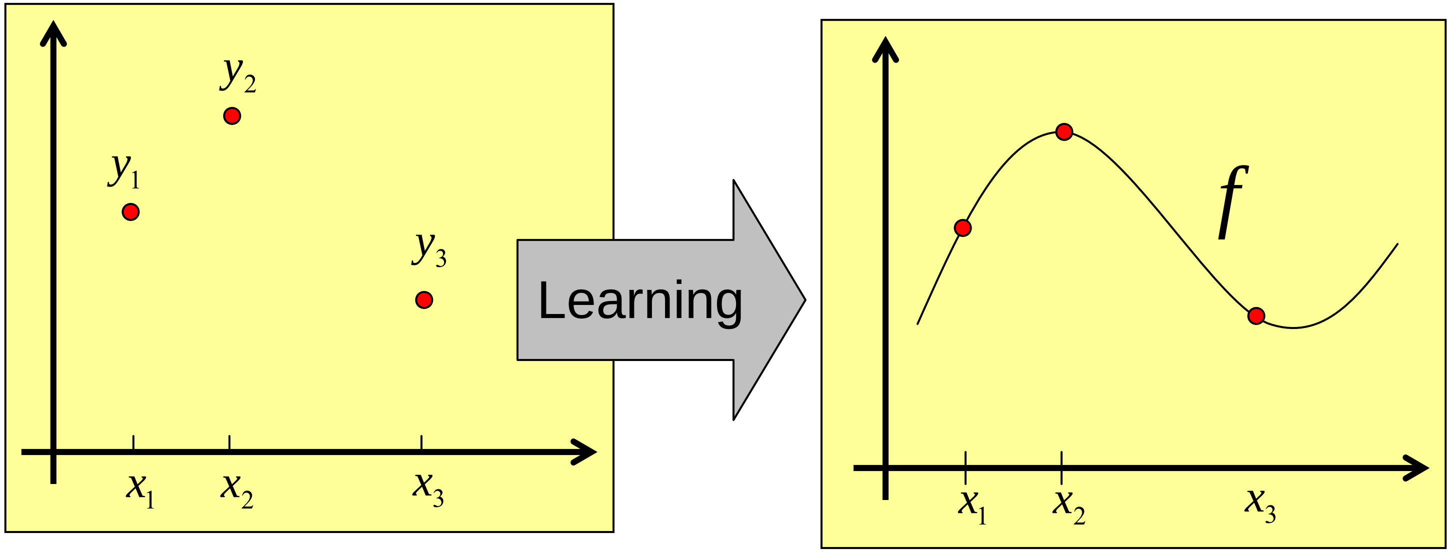
Types of internal models


Motor Command
Context
[task, state, context] → motor command
Inverse Model
CNS Internal Representations
Forward Sensory Model
Forward Dynamic Model
Context
Sensory Feedback
Context
Motor Command
State
Previous State
[state, motor command, context] → sensory feedback
[previous state, motor command, context] → state

Learn these models from data or observations of input-output pairs …

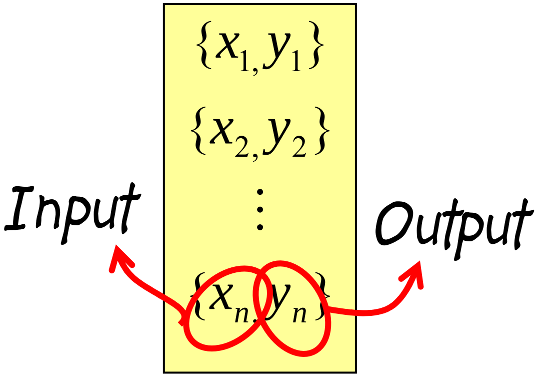
[Figure reproduced from Wolpert & Ghahramani, Nature Neuroscience(2000)]

Learning as a function approx. problem



Learning

Data



$\{x_1, y_1\}$

$\{x_2, y_2\}$

$\vdots$

$\{x_n, y_n\}$

Function Approx.

Regression

$$f : R^N \rightarrow R^M$$

Data and Inference

- **Training Data :** $\mathbf{D} = \{\mathbf{X}, \mathbf{t}\} = \{\mathbf{x}_i, t_i\}_{i=1}^{N}$

  The outputs t_i (targets) can be :

  true/false (Concept Learning),

  class labels (Classification) or

  real numbers (Regression).

- **Data Generating Process :** $y_i = f(\mathbf{x}_i, \mathbf{z}_i)$ *where* $\mathbf{z}_i$ *is the hidden variable.*

  Variables which cannot be directly measured

- **Observed data contaminated by noise:** $t_i = y_i + \varepsilon$

  $\quad$ *where* ε *is the noise*

- **Modeling the data:** $\hat{y}_i = g(\mathbf{x}_i \mid \theta)$

  $\quad$ *where* $\theta =$ *parameters.*

Machine Learning Design Issues

1. **Choosing the Training Data (Active learning)**

   - Decide on type of data (reward, class or real values)

   - Sampling

   To get a representative distribution

   To select informative data

2. **Model Selection or Target Representation**

   How to choose the right model $g(\mathbf{x} \mid \theta)$?

3. **Measure of Distance (Error/Loss function)-> d(.)**

   $$L(\theta \mid D) = \sum_i d(y_i, \hat{y}_i) = \sum_i d(y_i, g(\mathbf{x}_i \mid \theta))$$

4. **Optimization Procedure**

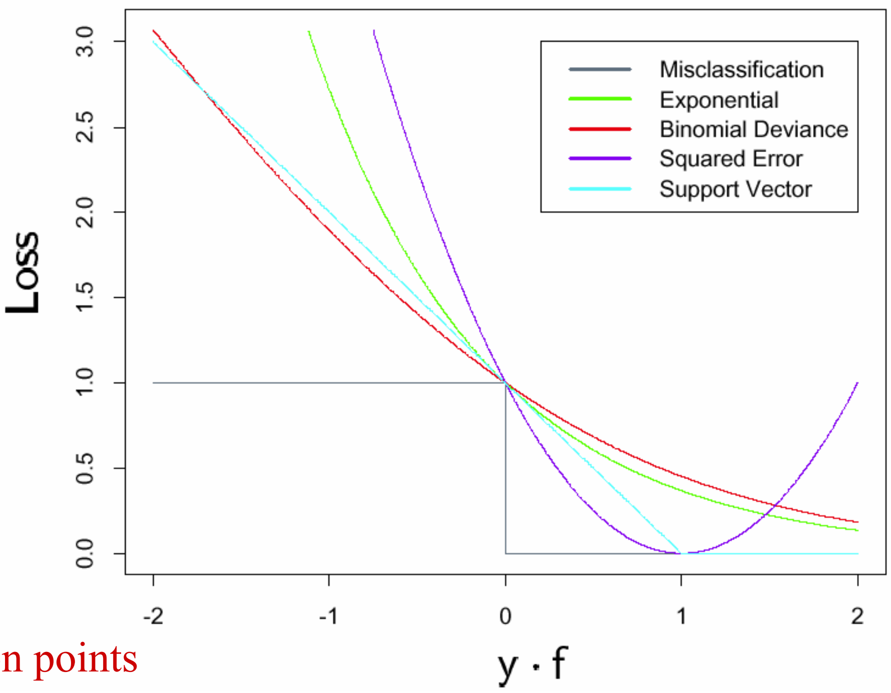
   $$\theta^* = \arg\min_{\theta} L(\theta \mid \mathbf{D})$$

Cost/Loss Functions(I)

For Classification $y = \pm 1,\ prediction = \hat{f}(x),\ Class\ prediction = \mathrm{sgn}(\hat{f}(x))$

- Misclassification : $I(\mathrm{sgn}(\hat{f}) \neq y)$
- Exponential : $\exp(-y\hat{f})$
- Binomial Deviance: $\log(1+\exp(-2y\hat{f}))$
- Squared Error : $(y-\hat{f}(x))^2$
- Support Vector : $(1-y\hat{f})\cdot I(y\hat{f} > 1)$

$$Here,\ I(x) = 1\ if\ \ x = TRUE$$
$$= 0\ otherwise.$$



Exponential error loss concentrates much more on points with large negative margins while **Binomial deviance** spreads the influence over all data. Hence, Binomial deviance is more robust in noise prone situations.

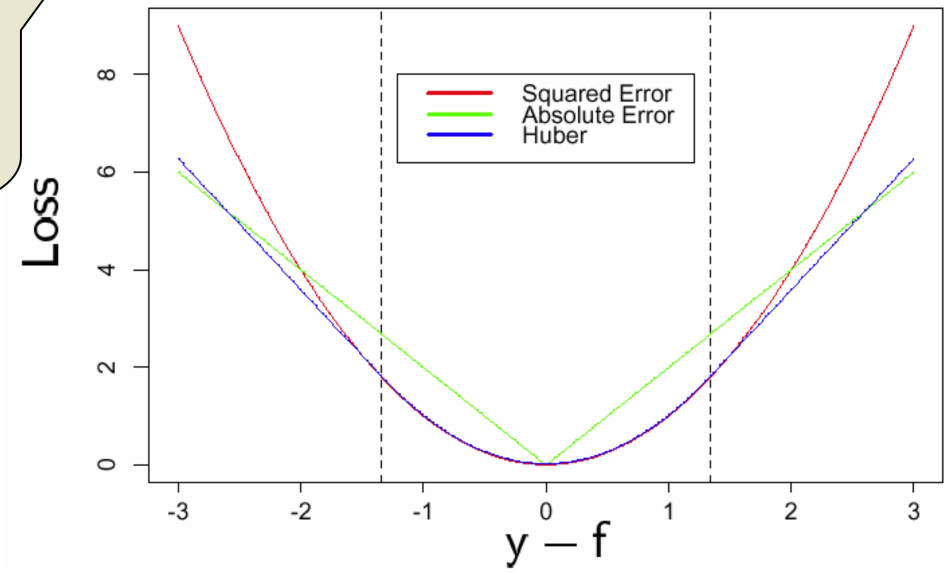
Cost/Loss Functions(II)

For Regression

- **Squared Error Loss :** $[y - \hat{f}(x)]^2$
- **Absolute Error Loss :** $| y - \hat{f}(x) |$
- **Huber Loss :** $\begin{matrix} [y - \hat{f}(x)]^2 & \textit{for}\ | y - \hat{f}(x) | \le \delta \\ 2\delta(| y - \hat{f}(x) | - \delta / 2), & \textit{otherwise} \end{matrix}$

Cost Functions

Huber Loss combines the good properties of squared error loss near zero and absolute error loss when the error is large.



Empirical Error vs. Generalization Error

Generalization

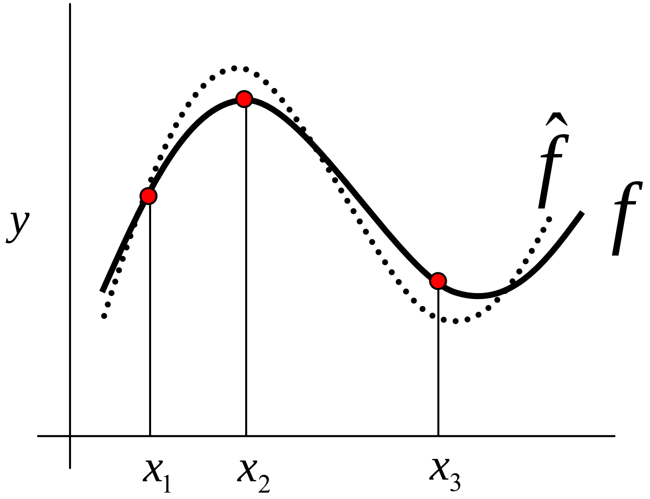
The ability of a learning system to not only memorize training data but also to predict reasonably well for novel inputs based on the training examples.

$$L(\alpha) = \int \frac{1}{2}\left| f(x) - \hat{f}(x,\alpha) \right| dx$$: True Error/Generalization Error

$$L_{emp}(\alpha) = \frac{1}{2N}\sum_{i=1}^{N}\left| f(x_i) - \hat{f}(x_i,\alpha) \right| = \frac{1}{2N}\sum_{i=1}^{N}\left| y_i - \hat{y}_i \right|$$: Empirical (training) Error

Usually, we only have access to the empirical (training) error since we do not know the true generating function.

However, performing optimization only based on empirical error does not ensure good generalization… we will see an example of **overfitting** soon !!



Overfitting

Overfitting

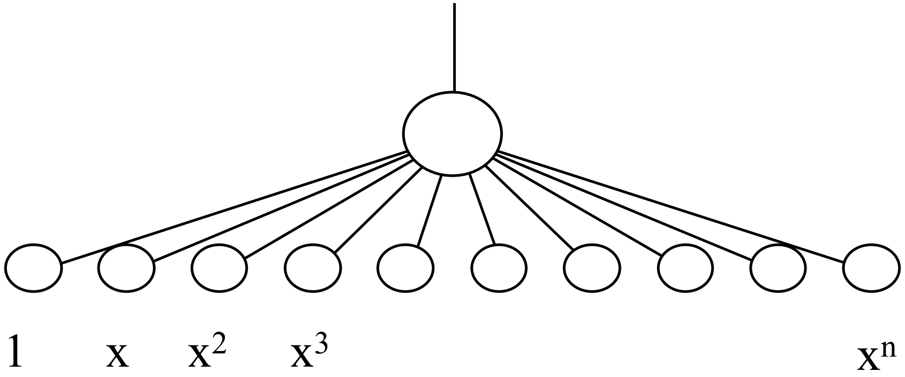
The tendency of the learning system (typically with too many open parameters) to concentrate on the idiosyncrasies of the training data and noise rather than capturing the essential features of the data generating mechanism.

- **Example from regression: Polynomial fitting**

$$y = f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_n x^n = \mathbf{w}^T \mathbf{x}$$

$$\textit{where} \quad \mathbf{x} = \begin{bmatrix} 1 \\ x \\ \ldots \\ x^n \end{bmatrix}$$

How many inputs (i.e., degree of polynomial) should be used to fit the data?



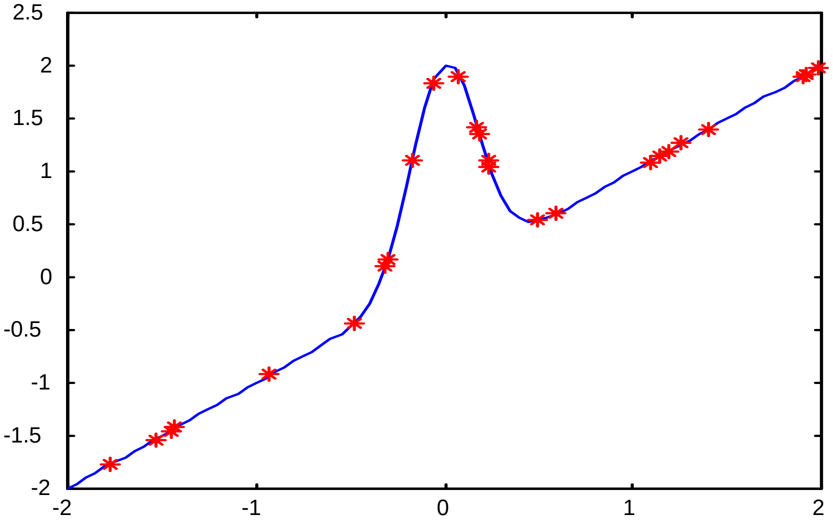
Overfitting (cont'd)

- **A popular error criterion is the Mean Squared Error Criterion**

$$J = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- **or the Normalized Mean-Squared Error**
  - the normalized MSE is a measure of how much variance in the output data was explained

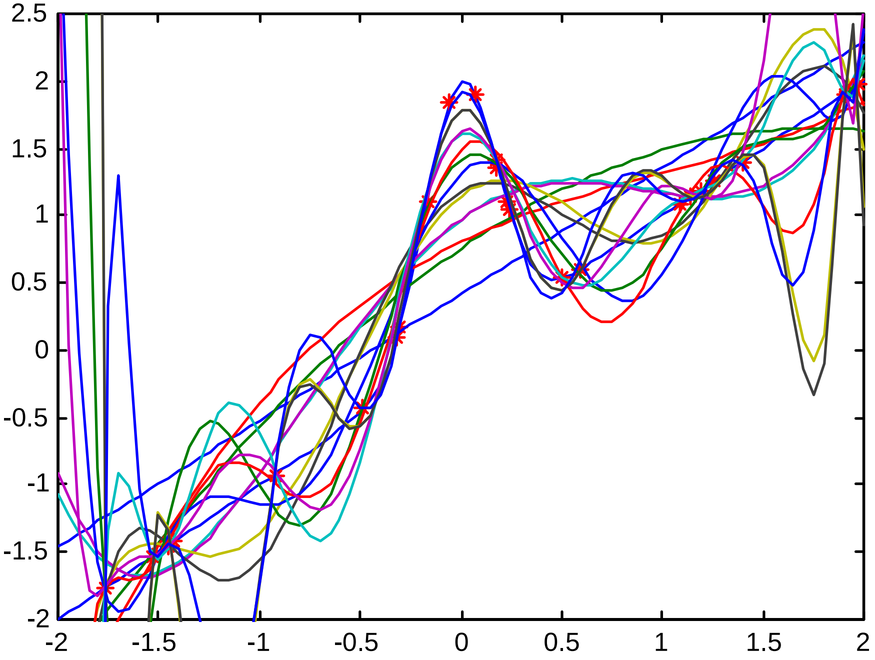
$$J = \frac{1}{N\sigma_y^2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

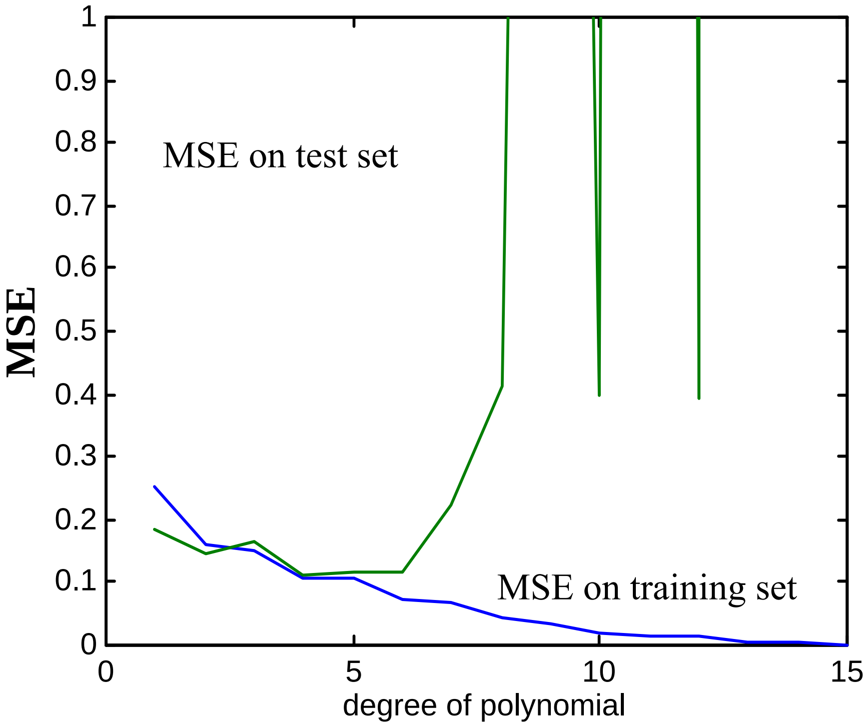


Example target function :

$$y = x + 2\exp(-16x^2)$$

Overfitting with Polynomials





***Observation:** Just concentrating on reducing training error results in worse generalization with novel data…due to **overfitting**.*

Bias-Variance Dilemma

- Too few features are bad, too many are bad, thus, there should be an optimum
- A closer look at the MSE criterion $$J=\frac{1}{N}\sum_{i=1}^{N}\left(t_i-\hat{y}_i\right)^2=\frac{1}{N}\sum_{i=1}^{N}\left(t_i-\hat{f}(\mathbf{x}_i)\right)^2$$
- What we actually want to minimize is the generalization/true error …but we do not have the original function and hence, all we have access to is the training /empirical error !!
- The next best thing to do: **minimize J in expectation, i.e., over infinitely many data sets ->**

$$\min\left(E\{J\}\right)$$

Bias-Variance Dilemma (II)

$$E\{J\} = E\left\{\frac{1}{N}\sum_{i=1}^{N}(t_i - \hat{y}_i)^2\right\} = \frac{1}{N}\sum_{i=1}^{N} E\left\{(t_i - \hat{y}_i)^2\right\} = \frac{1}{N}\sum_{i=1}^{N} E\{J_i\}$$

Bias –Variance Decomposition of Expected Error

$$\boxed{\begin{aligned} E\{J_i\} &= \sigma_\varepsilon^2 + \left(E\{\hat{y}_i\} - f(\mathbf{x}_i)\right)^2 + E\left\{(\hat{y}_i - E\{\hat{y}_i\})^2\right\} \\ &= \mathrm{var}(noise) + bias^2 + \mathrm{var}(estimate) \end{aligned}}$$

Note: For derivation of the decomposition, refer to class handout.

Adobe Acrobat Document

Usually, if we try to reduce bias in a model, it increases the variance and vice-versa, resulting in the dilemma for optimal choice.