Northeastern University

Supervised Machine Learning

Roi Yehoshua

Agenda

Northeastern University

- Supervised learning formal definition
- Regression vs. classification
- Data-generating distribution
- Generative vs. discriminative models
- Bias-variance tradeoff
- Model capacity

Supervised Learning: Formal Definition

Northeastern University

- **Given**: a **training set** of n labeled examples $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$
  - Each $\mathbf{x}_i$ is a d-dimensional vector of feature values, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$
  - y_i is the target (output) variable we are trying to predict
  - y_i is generated by an unknown function $y = f(\mathbf{x})$
- In many cases, the training set is stored in a matrix known as the ***design matrix***

Feature 1 Feature d

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{id} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

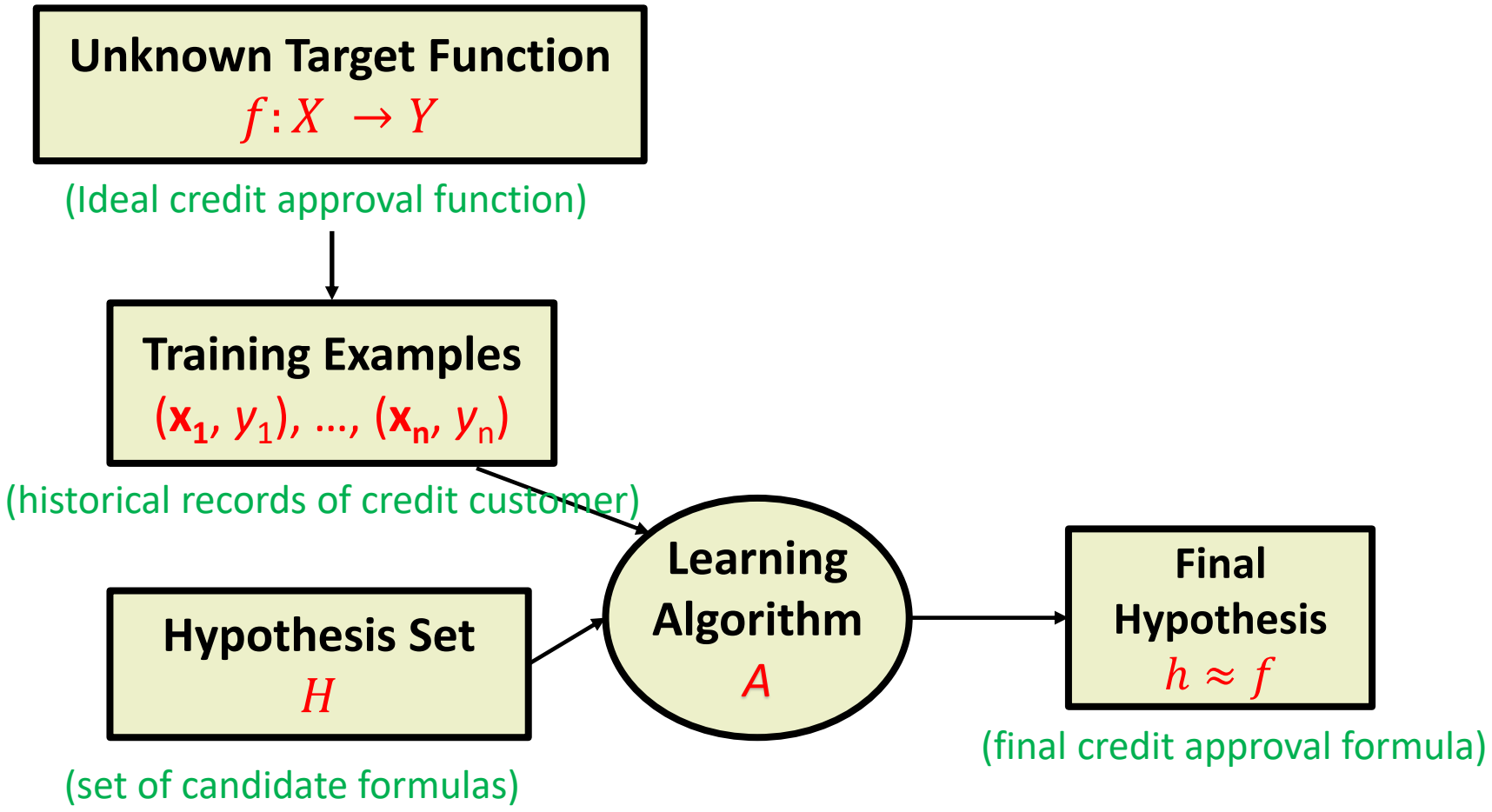
Training example i

Supervised Learning: Formal Definition

Northeastern University

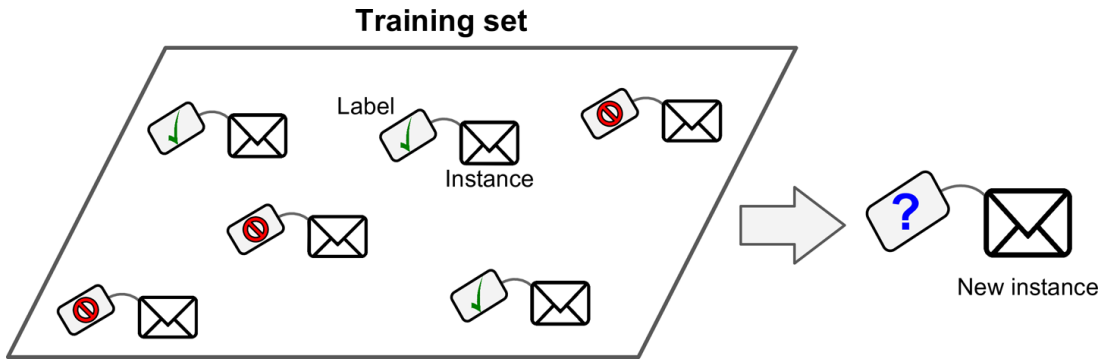
- **Goal**: find a function $h(\mathbf{x})$ (**hypothesis**) that approximates the true function f
- This function belongs to some **hypothesis space** H
  - The set of all functions that are learnable by our chosen algorithm
    - e.g., in linear regression, H contains all linear functions
- A hypothesis **generalizes** well if it correctly predicts the label y for novel examples
- **Learning** is a search through the space of possible hypotheses for one that generalizes well

Supervised Learning: Formal Definition



Classification vs. Regression

- When the label *y* is a discrete variable, the learning problem is called **classification**
  - *y* is the **class** that the sample belongs to
- When the label *y* is continuous, the learning problem is called **regression**



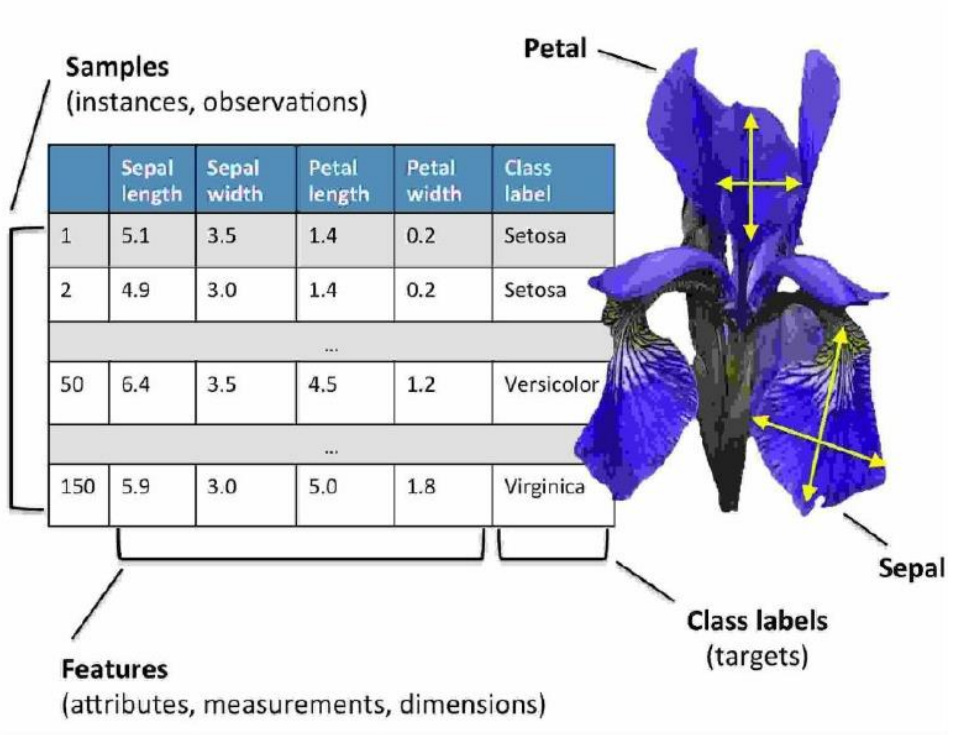
Classification

Regression

Example 1: Iris Flower Classification

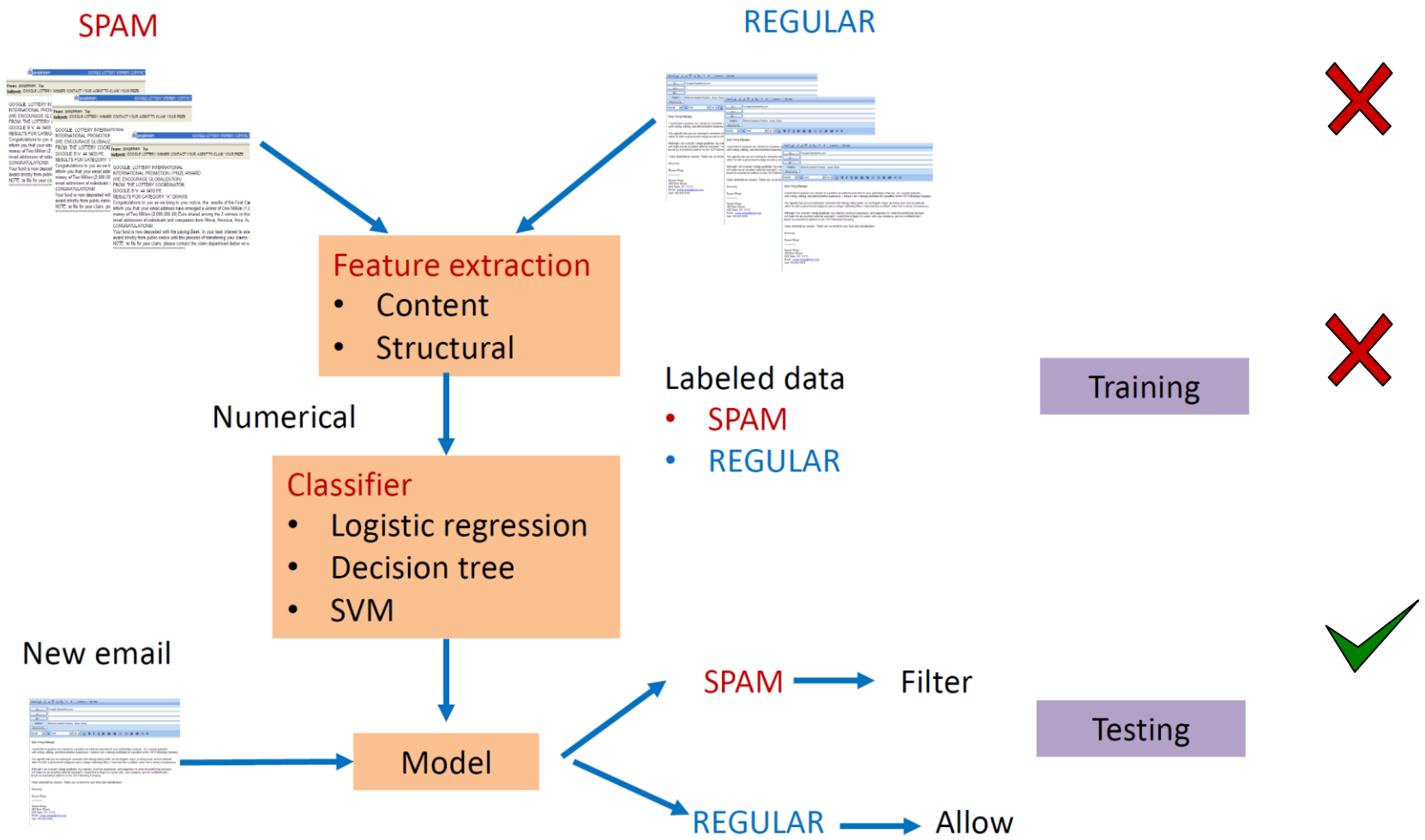
Northeastern University

- From the statistician Douglas Fisher (1936)
- Data set: 150 Iris flowers, 50 from each of the species: Setosa, Virginica, Versicolour
- Four attributes
  - Sepal width and length (in centimeters)
  - Petal width and length (in centimeters)



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Example 2: Spam Detection



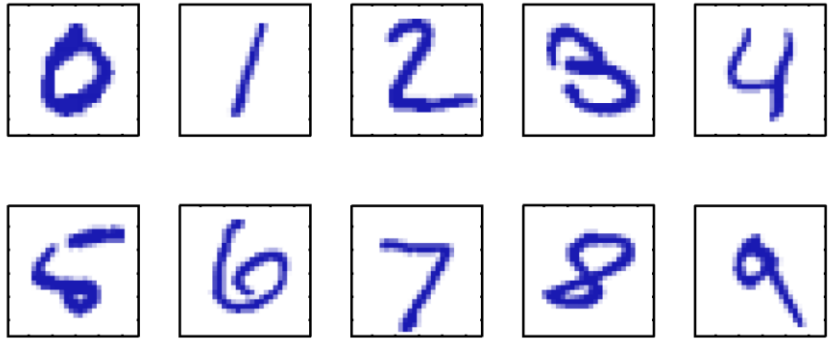
Dear Sir.
First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.
99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Example 3: Handwritten Digit Recognition

Northeastern University



- MNIST data set
- **Input**: images of 28×28 pixels
  - Pixel values range from 0 to 255 (grey level)
- **Output**: a digit 0-9
- Data set contains 60,000 training examples and 10,000 test examples
- **Setup**:
  - Represent each input image as a vector $\mathbf{x} \in R^{784}$
  - Learn a classifier $h(\mathbf{x})$ such that h: $\mathbf{x} \rightarrow \{0,1,2,3,4,5,6,7,8,9\}$
  - Feature extraction: NumComponents, AspectRatio, NumLoops
- Can achieve a testing error of 0.4%
- One of the first commercial and widely used ML systems (for zip codes and checks)

Example 4: Credit Approval

Northeastern University

- Input: applicant information

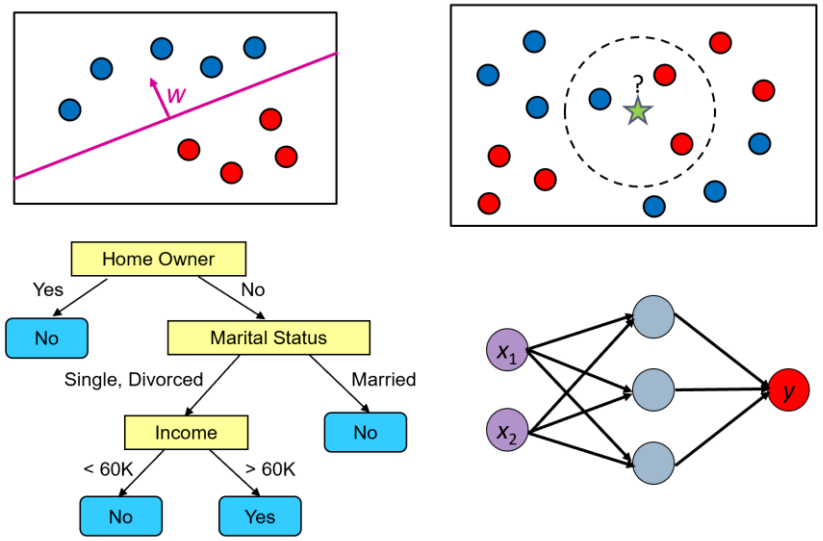
Relation: german_credit

No.	checking_status Nominal	duration Numeric	credit_history Nominal	purpose Nominal	credit_amount Numeric	savings_status Nominal	employment Nominal	installment_commitment Numeric	personal_status Nominal
1	(0	6.0	critical/other exi...	radio/tv	1169.0	no known savi...	)=7	4.0	male single
2	0(=X(200	48.0	existing paid	radio/tv	5951.0	(100	1(=X(4	2.0	female div/dep...
3	no checking	12.0	critical/other exi...	education	2096.0	(100	4(=X(7	2.0	male single
4	(0	42.0	existing paid	furnitu...	7882.0	(100	4(=X(7	2.0	male single
5	(0	24.0	delayed previously	new car	4870.0	(100	1(=X(4	3.0	male single
6	no checking	36.0	existing paid	education	9055.0	no known savi...	1(=X(4	2.0	male single

- Output:
  - approve credit? → classification
  - credit line (dollar amount) → regression

Learning Algorithms

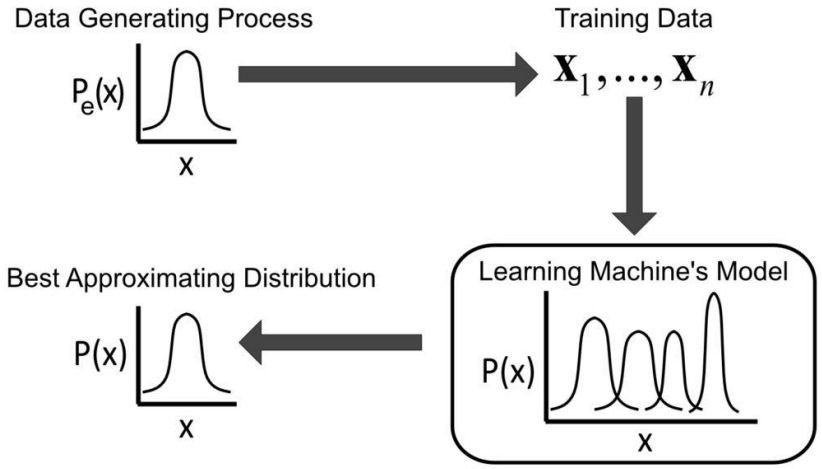
- There are many supervised learning algorithms
  - From simple decision trees to deep neural networks



- **No free lunch (NFL)** theorem:
  - No single learning algorithm performs best across all possible problems or datasets
- **Occam's Razor**:
  - Given multiple explanations (models) for a phenomenon, prefer the simplest one

Data-Generating Distribution

- The probability distribution that generates the observed data $P(\mathbf{x}, y)$
- Unknown in practice: we only observe samples from it
- Learning algorithms try to approximate this distribution from the training examples
  - Some algorithms assume specific form of the distribution (e.g., Gaussian)
- Allows to frame the learning problem within a probabilistic model

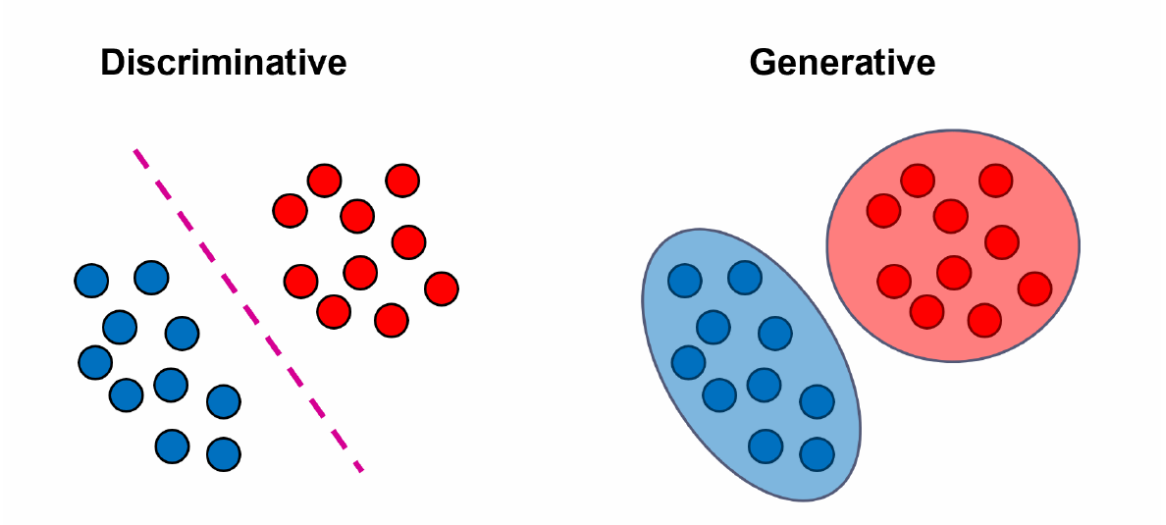


Discriminative vs. Generative Models

Northeastern University

- **Discriminative models** learn directly the posterior probability $P(y|\mathbf{x})$
- **Generative models** learn first the class-conditional input distribution $P(\mathbf{x}|y)$
  - Allows sampling from this distribution to generate new inputs
  - Using Bayes rule, we can infer the posterior distribution

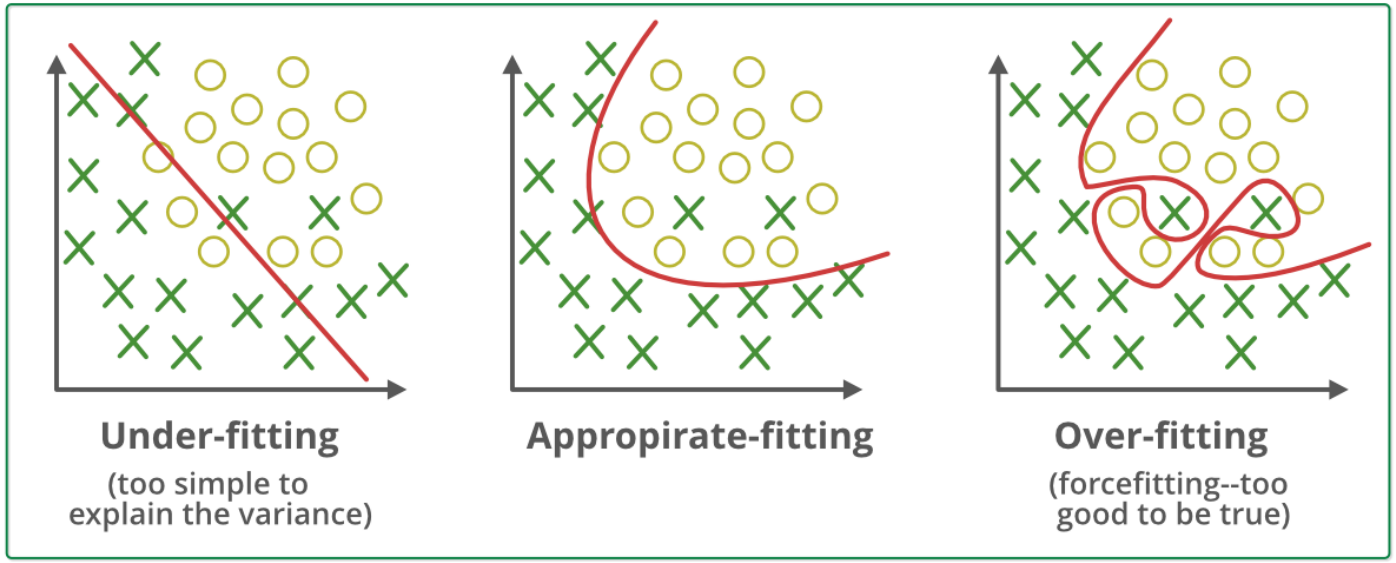
$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$



Bias-Variance Tradeoff

Northeastern University

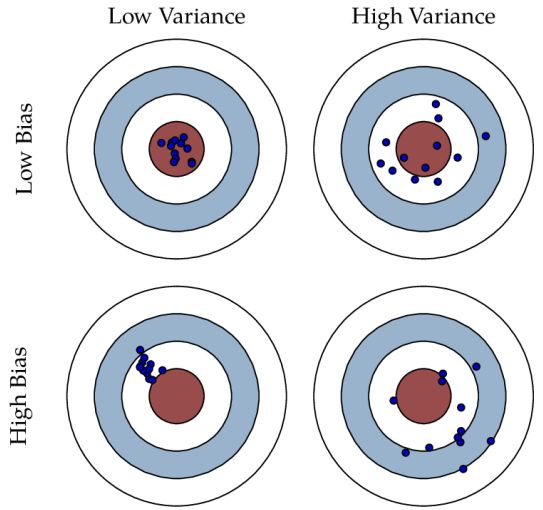
- **Underfitting**: the model is too simple to capture the pattern in the data
- **Overfitting**: the model learns the training data too well and fails to generalize
- **Bias-variance tradeoff**: find the right balance between underfitting and overfitting



The Bias-Variance Tradeoff

Northeastern University

- A key concept for analyzing tradeoffs between model complexity and generalization
- The generalization error of a model can be decomposed into 3 components:
  - **Bias**: A systematic error caused by incorrect or overly simplistic model
    - A model with **high bias** is **underfitting** the training data
  - **Variance**: The model's sensitivity to noise and fluctuations in the training set
    - Measured as the variance of the model's predictions when trained on different datasets
    - A model with **high variance** is **overfitting** the training data
  - **Irreducible error**
    - Caused by inherent noise in the data
    - Cannot be eliminated by any model regardless of its complexity
- Reducing bias often increases variance and vice versa



Formal Proof

Northeastern University

- We consider regression problems with the following setup:
  - The targets are generated by an unknown function $f(\mathbf{x})$, corrupted by zero-mean noise:

$$y = f(\mathbf{x}) + \epsilon, \quad \text{where } \mathbb{E}[\epsilon] = 0,\ \mathrm{Var}(\epsilon) = \sigma^2$$

  - The model is trained on data sampled from the data-generating distribution

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \quad \text{where } (\mathbf{x}_i, y_i) \overset{\text{i.i.d.}}{\sim} P(\mathbf{x}, y)$$

  - Generalization error is defined as the expected mean squared error on a new sample (**x**, y)
    - averaged over the randomness in the training set *D*

$$\mathrm{MSE}(\mathbf{x}, y) = \mathbb{E}_D\left[(y - h_D(\mathbf{x}))^2\right]$$

    - h_D is the prediction function of the model trained on the training set *D*

Formal Proof

Northeastern University

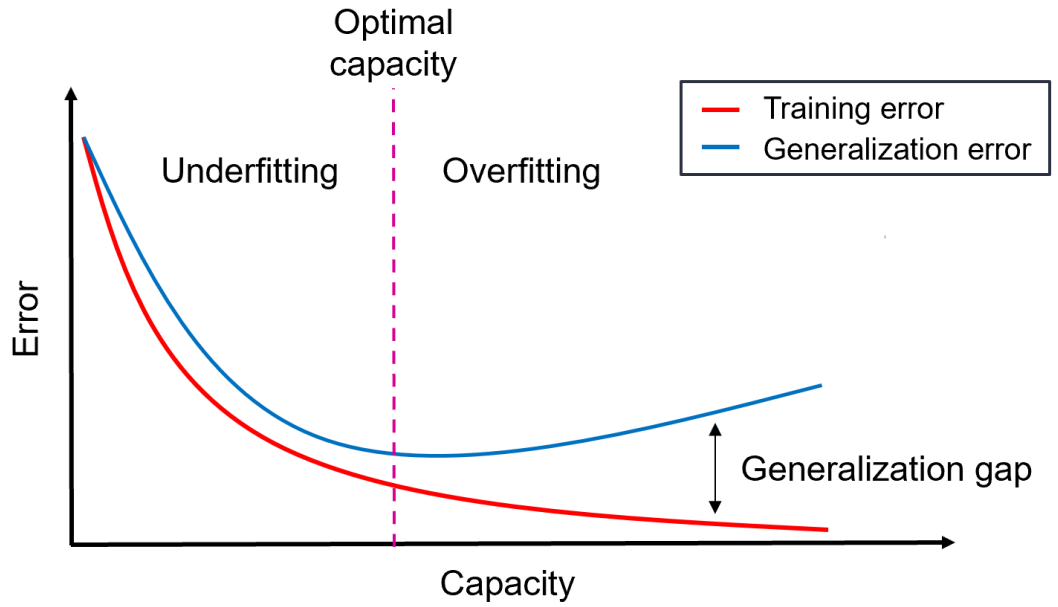
- We can now decompose the MSE into bias and variance terms:

$$
\begin{aligned}
\mathrm{MSE}(\mathbf{x}, y) &= \mathbb{E}_D\left[(y - h_D(\mathbf{x})^2)\right] && \text{(definition of MSE)} \\
&= \mathbb{E}_D\left[(f(\mathbf{x}) + \epsilon - h_D(\mathbf{x}))^2\right] && \text{(definition of } y\text{)} \\
&= \mathbb{E}_D\left[[(f(\mathbf{x}) - h_D(\mathbf{x})) + \epsilon]^2\right] && \text{(rearranging terms)} \\
&= \mathbb{E}_D\left[(f(\mathbf{x}) - h_D(\mathbf{x}))^2\right] + \mathbb{E}_D[\epsilon^2] && \text{(if } A \perp\!\!\!\perp B,\ \mathbb{E}[AB] = \mathbb{E}[A]\,\mathbb{E}[B]\text{)} \\
&= \mathbb{E}_D\left[(f(\mathbf{x}) - h_D(\mathbf{x}))^2\right] + \sigma^2 && (\mathbb{E}_D[\epsilon^2] = \mathrm{Var}(\epsilon) = \sigma^2) \\
&= \left(\mathbb{E}_D\left[f(\mathbf{x}) - h_D(\mathbf{x})\right]\right)^2 + \mathrm{Var}_D\left(f(\mathbf{x}) - h_D(\mathbf{x})\right) + \sigma^2 && (\mathbb{E}[X^2] = \mathbb{E}[X]^2 + \mathrm{Var}(X)) \\
&= \underbrace{\left(\mathbb{E}_D\left[f(\mathbf{x}) - h_D(\mathbf{x})\right]\right)^2}_{\text{Bias}^2} + \underbrace{\mathrm{Var}_D(h_D(\mathbf{x}))}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Noise}} && (\mathrm{Var}(a + bX) = b^2\,\mathrm{Var}(X))
\end{aligned}
$$

MSE = (bias)^2 + variance + noise

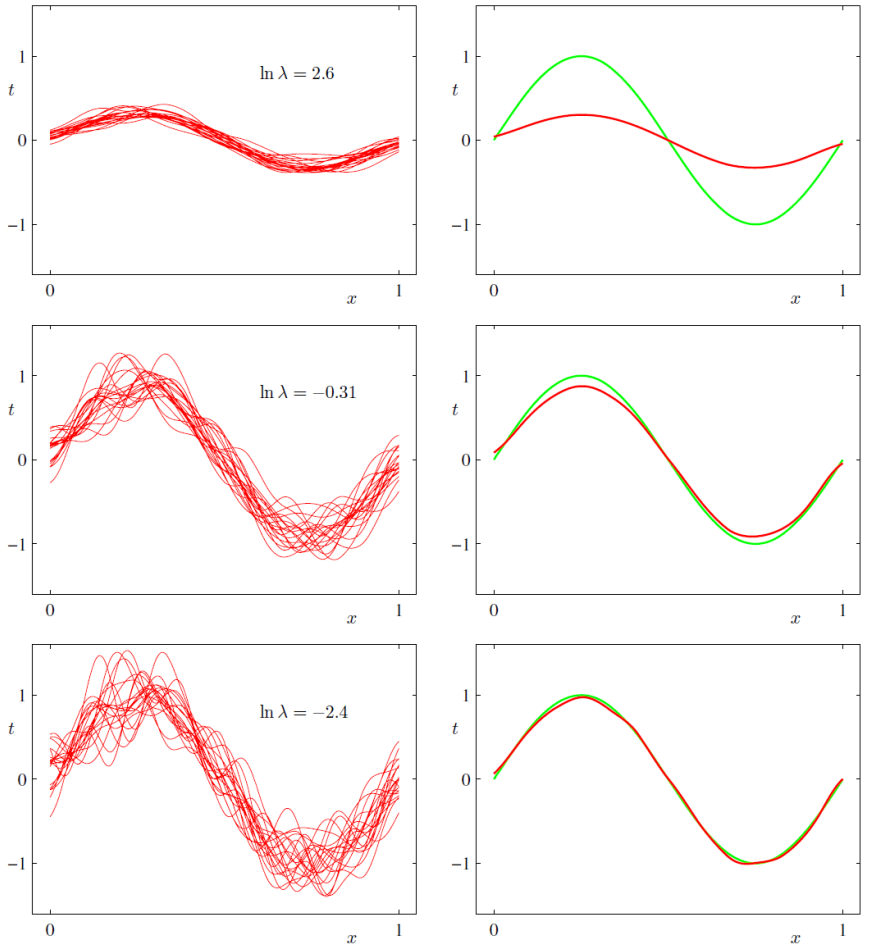
Model Capacity

- We can control the bias-variance tradeoff by adjusting the model's capacity
  - Complex (flexible) models typically have low bias and high variance
  - Simple (constrained) models typically have high bias and low variance
- The optimal model achieves the best tradeoff to minimize the generalization error



Bias-Variance Example

- The target function is $f(x) = \sin 2\pi x$
- 100 training sets are sampled from $f(x)$
- Each training set contains 25 points
- A separate regression model is trained on each set
- Each model uses 24 Gaussian basis functions
- Experiment repeated with 3 different regularization strengths



- Green: the true function
- Red: average prediction across the 100 models

Example adopted from Bishop “Pattern Recognition and Machine Learning,” Figure 3.14, p. 153.