ITCS 6156/8156 Fall 2023
Machine Learning

Linear Regression

Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Some content in the slides is based on Dr. Razvan’s lecture

Recap

- Spam email classification:
  - Binary classification of emails:
    Spam vs. Ham (Legitimate message)

- Expert Systems approach (Rule-based)
  - A group of experts write rules determining whether an email is spam or not.
  - A programmer implement the rules into computer code

- Machine Learning algorithms can automatically learn the weights to combine features.

Machine Learning

- Function is everywhere!
  - Function → f ; Input instance → x; Output Targe → y

$f($  $)$ → 

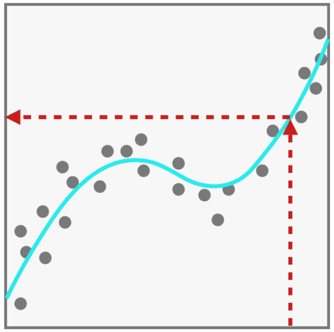
$f($  $)$ → 

- Machine Learning Task:
  - learn an (unknown) function $f: X \rightarrow Y$ that maps input instances x $\in X$ to output targets $f(x) \in Y$.

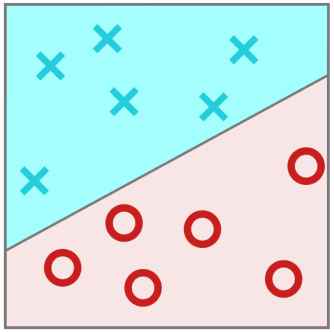
Classification vs. Regression

- Machine Learning Task:
  - learn an (unknown) function $f: X \rightarrow Y$ that maps input instances x $\in X$ to $\boxed{\text{output targets } f(x) \in Y}$.
- Linear Regression is a Regression algorithm.

- **Regression**:
  - The output targets $f(x) \in Y$ is continuous, or has a continuous component (e.g., stock price prediction).



- **Classification**:
  - The output targets $f(x) \in Y$ is one of a finite set of discrete categories (e.g., mail classification).



Supervised Learning

- Machine Learning Task:
  - learn an (unknown) function $f: \boxed{X} \rightarrow \boxed{Y}$ that maps input instances x $\in X$ to output targets $f(x) \in Y$.

- Linear Regression is a Supervised Learning algorithm.

- Supervised Learning: The output targets are known in the set of training examples:
  - $(x_1, y_1), (x_2, y_2) \ldots (x_i, y_i)$ [e.g., predict stock price]

- Training data vs. Test data

All Data

Training Data

Test Data

Supervised Learning

- Training data vs. Test data

All Data

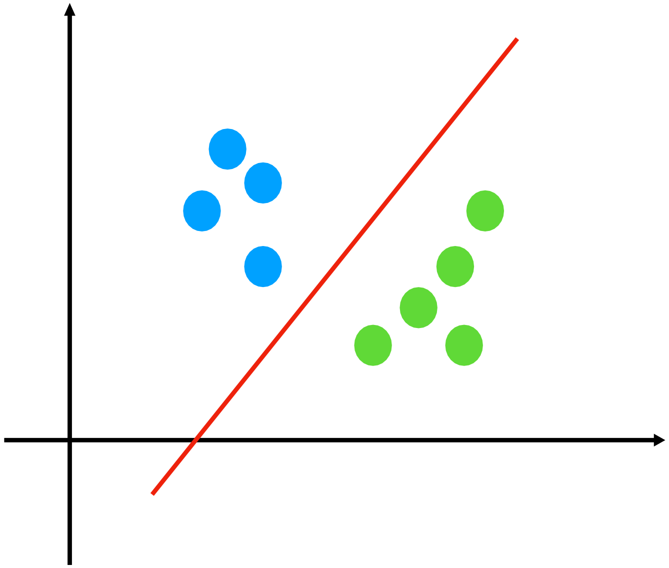
Training Data	Test Data

- The goal of machine learning algorithms is to build a function (hypothesis) $h(x)$ such that:
  - h matches f well on the training data → h is able to fit data that it has see
  - h also f well on the test data → h is able to generalize to unseen data
- To achieve the goal, we want to choose h from a "nice" class of functions that depends on a vector of parameters w:
  - $h(x) \equiv h_w(x) \equiv h(w, x)$

Do we need to make assumptions on the data?

- **No free lunch theorem:** we must make such assumptions.

- **Informal theorem:** for any machine learning algorithm $\mathcal{A}$, there must exist a task $\mathcal{P}$ on which it will fail

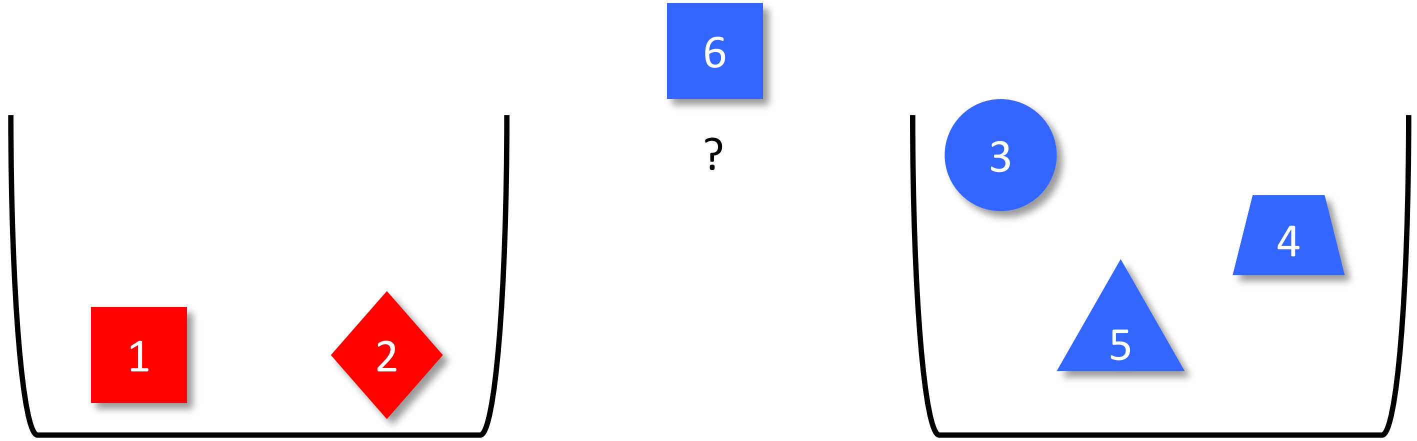


- We use prior knowledge (i.e., we believe linear function is enough) to design an ML algorithm here

Which Hypothesis?

- Hypothesis class: $\mathcal{H} = \{h\}$
  - E.g.: linear models, quadratic models, neural networks, etc.

- An example:
  - M_1: x is Red => $x \in C_1$
  - M_2: x is a Square or x is a Diamond => $x \in C_1$
  - M_3: x is Red and x is a Quadrilateral => $x \in C_1$



Container C_1 Container C_2

Occam’s Razor

William of Occam (1288 – 1348)

English Franciscan friar, theologian and philosopher.

“*Entia non sunt multiplicanda praeter necessitatem*”

- Entities must not be multiplied beyond necessity.

i.e. Do not make things needlessly complicated.

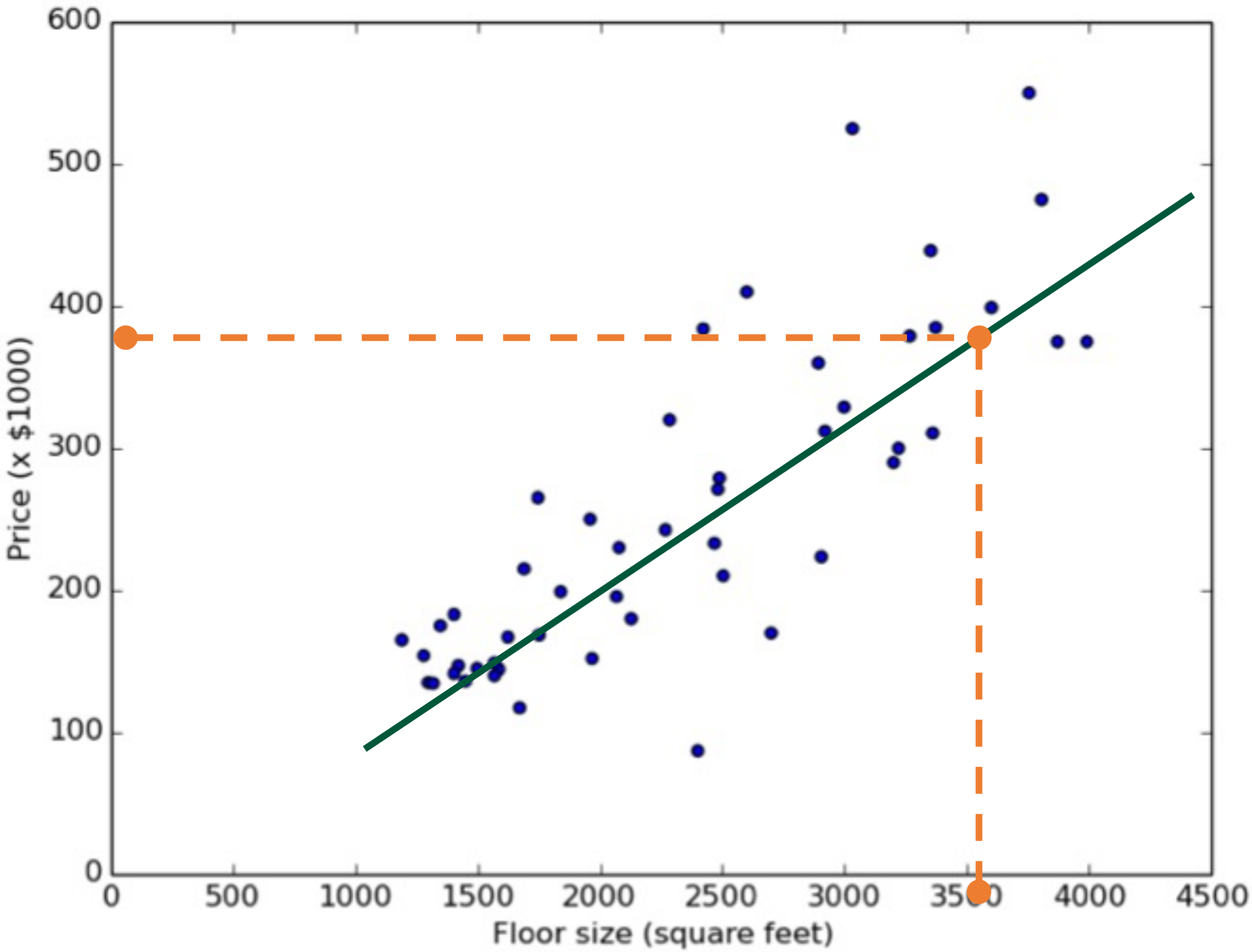
i.e. Prefer the simplest hypothesis that fits the data.

House Price Prediction

- Given the floor size in square feet, predict the selling price:
  - Input x: the floor size of the house
  - Output y: the selling price of the house
  - Need to learn a function (hypothesis) h such that $h(x) \approx f(x)$

- Is this a classification or regression task?
  - Regression, because the house price is real-valued.
  - Would a problem with only two labels $y_1 = 0.5$ and $y_2 = 1.0$ still be regression?

House Price Prediction



- **Another hypothesis?**
- **Pros and cons of the hypothesis?**

Linear Regression

- Use a linear function to approximate the real (unknown) function:
  - $h_{\mathbf{w}}(X) = \mathbf{w}^T X = [w_0, w_1]^T [1, x] = w_1 x + w_0$
  - In our case, $h_{\mathbf{w}}(X)$ is a straight line
    - w_0 is the intercept (or the bias term)
    - w_1 controls the slope
- Actually, the floor size of the house is not the only factor determining its sell price. There are many factors: $(x_1, \dots, x_d)$.
  - $h_{\mathbf{w}}(X) = \mathbf{w}^T X = [w_0, w_1, \dots, w_d]^T [1, x_1, \dots, x_d]$
    $= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$
- Assumption: the prediction results are the linear combination of input attributes (features).

Error Measurement

- Our linear approximation function:
  - $$h_{\mathbf{w}}(X) = \mathbf{w}^T X = [w_0, w_1, \dots, w_d]^T [1, x, \dots, x_d]$$
    $$= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$
- Error Measurement: Find **w** that obtains the best fit on the training data, i.e. find **w** that minimizes the sum of square errors:

$$J(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} (h_w(X_n) - y_n)^2$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} J(\mathbf{w})$$

  - N: Total number of samples in training set
  - $J(\mathbf{w})$: Error function
  - $\hat{\mathbf{w}}$: Optimimal $\boldsymbol{w}$ that minimize $J(\mathbf{w})$
- Why do we use the square errors?

Inductive Learning Hypothesis

- Learning = finding the "right" parameters $\mathbf{w}^T = [w_0, w_1, \dots, w_d]$
  - Find $\mathbf{w}$ that minimizes an error function $J(\mathbf{w})$ which measures the misfit between $h(\mathbf{x}_i, \mathbf{w})$ and t_i.
  - Expect that $h(\mathbf{x}, \mathbf{w})$ performing well on training examples $\mathbf{x}_i \Longrightarrow h(\mathbf{x}, \mathbf{w})$ will perform well on arbitrary test examples $\mathbf{x}_\mathrm{j} \in \mathbf{X}$.

Inductive Learning Hypothesis

Matrix Notation

- **Linear Regression Learning Task**
  - learn $\mathbf{w}$ given training examples $< \mathbf{X}, \boldsymbol{y} >$.
  - The training data is denoted as $< \mathbf{X}, \boldsymbol{y} >$, where $\mathbf{X}$ is a $N \times D$ data matrix consisting of N data examples such that each data example is a D dimensional vector. $\boldsymbol{y}$ is a $N \times 1$ vector consisting of corresponding target values for the examples in $\mathbf{X}$.

- The derivation of the least squares estimate can be done by first converting the expression of the squared loss into matrix notation, i.e.,

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 = \frac{1}{2}(\boldsymbol{y} - \mathbf{X}\mathbf{w})^T(\boldsymbol{y} - \mathbf{X}\mathbf{w})$$

Analytical Solution

- To minimize the error, we first compute its derivative with respect to **w**:

$$\frac{\partial LL(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{2}\frac{\partial}{\partial \mathbf{w}}(\boldsymbol{y} - \mathbf{Xw})^T(\boldsymbol{y} - \mathbf{Xw})$$

- Note that, we use the fact that $(\mathbf{Xw})^T\boldsymbol{y} = \boldsymbol{y}^{\boldsymbol{T}}\mathbf{Xw}$, since both quantities are scalars, and the transpose of a scalar is equal to itself. Continuing with the derivative:

$$\frac{\partial LL(\mathbf{w})}{\partial \mathbf{w}} = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{2w}^{\boldsymbol{T}}\mathbf{X}^{\boldsymbol{T}}\mathbf{X} - \mathbf{2}\boldsymbol{y}^{\boldsymbol{T}}\mathbf{X})$$

Analytical Solution

- Setting the derivation to 0, we get:

$$\begin{aligned}
2\mathbf{w}^\top\mathbf{X}^\top\mathbf{X} - 2\mathbf{y}^\top\mathbf{X} &= 0 \\
\mathbf{w}^\top\mathbf{X}^\top\mathbf{X} &= \mathbf{y}^\top\mathbf{X} \\
(\mathbf{X}^\top\mathbf{X})^\top\mathbf{w} &= \mathbf{X}^\top\mathbf{y} \text{ (Taking transpose both sides)} \\
(\mathbf{X}^\top\mathbf{X})\mathbf{w} &= \mathbf{X}^\top\mathbf{y} \\
\mathbf{w} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}
\end{aligned}$$

The Moore-Penrose pseudo-inverse

Summary

- Our linear approximation function:
  - $$h_{\mathbf{w}}(X) = \mathbf{w}^T X = [w_0, w_1, \dots, w_d]^T [1, x, \dots, x_d]$$
    $$= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$
- Error Measurement: Find **w** that minimizes the sum of square errors:

$$J(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} (h_w(X_n) - y_n)^2$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} J(\mathbf{w})$$

- Analytical Solution:

$$\widehat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Questions?

UNIVERSITY OF NORTH CAROLINA CHARLOTTE