CS 341 Lecture 2

Supervised Learning
- problem setup
- examples

Linear regression in one variable
- setup
- cost function
- calc review
- minimization of cost function

Supervised Learning

- Learn an unknown function f from example input/output pairs

- Given: training examples

  $(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N),$

  $y_i = f(x_i)$

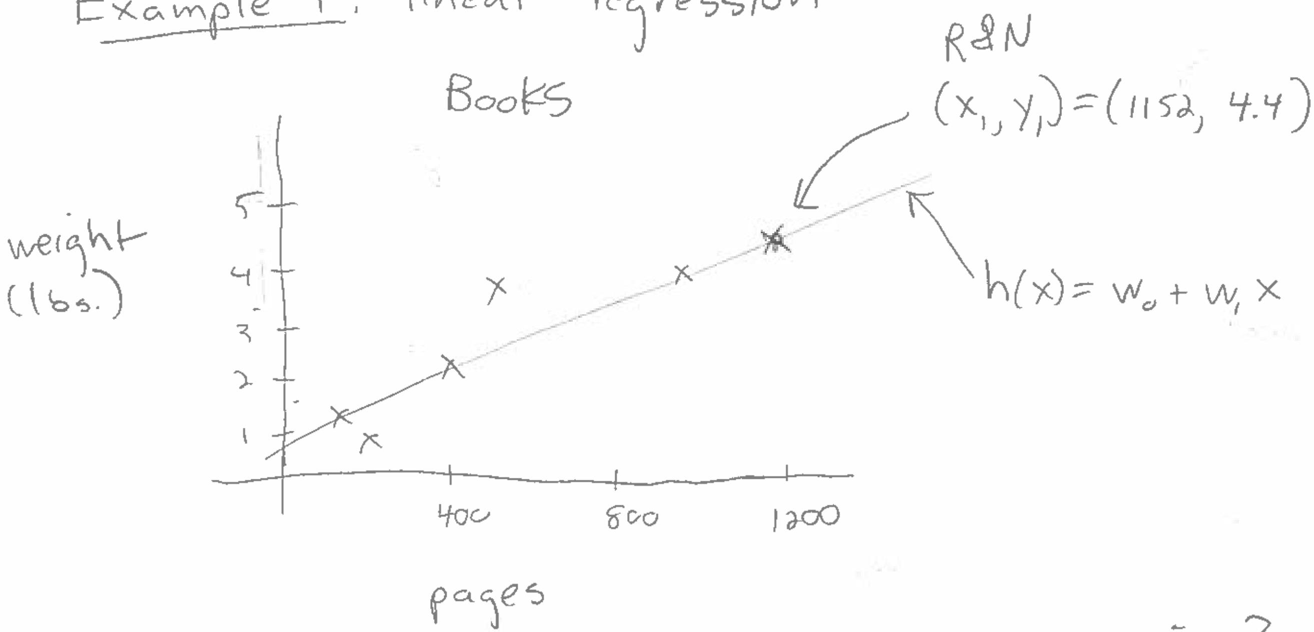
- Find: function $h \approx f$

  y_{N+1}

  (prediction $h(x_{N+1}) \approx f(x_{N+1})$ for unseen x_{N+1})

- Variations:
  - Input (x)
  - Output (y)
  - Function (h)

Example 1: linear regression

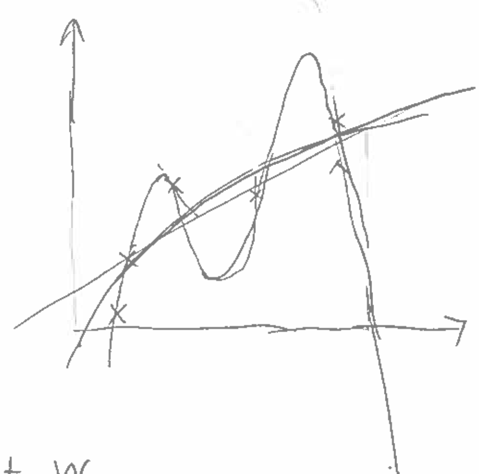


How much does an 800 page book weigh?

- $x \in \mathbb{R}$ (one real number) pages
- $y \in \mathbb{R}$ (weight in lbs.)

Options:

1a) h is linear $\left(h(x) = w_1 x + w_0 \right)$



1b) h is quadratic

$$h(x) = w_2 x^2 + w_1 x + w_0$$

1c) h is n^{th}-degree polynomial

$$h(x) = w_n x^n + w_{n-1} x^{n-1} + \ldots + w_1 x + w_0$$

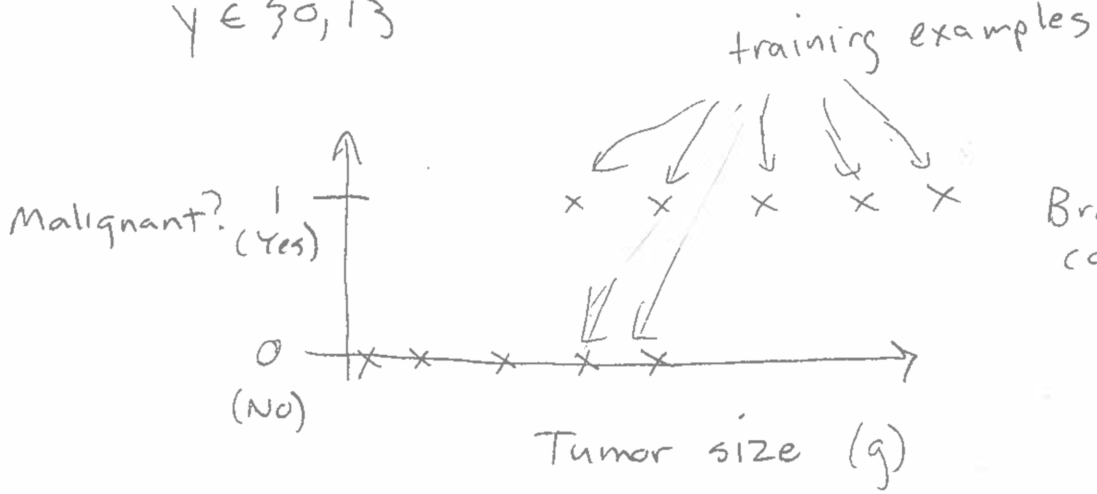
- Family of functions is called hypothesis space $\mathcal{H}$.
- A particular function $h \in \mathcal{H}$ is a hypothesis
- Learning algorithm: find "best" hypothesis $h \in \mathcal{H}$ according to some measure of quality

- Goals:
  - good fit to training data
  - predict well on new examples

Example 2: classification

2a) $x \in \mathbb{R}$

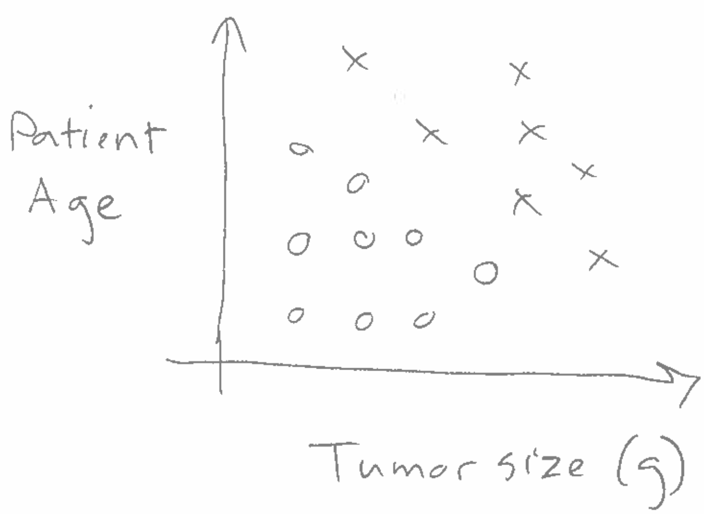
$y \in \{0, 1\}$



Breast cancer

2b) $x \in \mathbb{R}^2$ ← two input "features"

$y \in \{0, 1\}$



x_1: Tumor size

x_2: Age

$y_i = h(x_1, x_2)$

2c) $x \in \mathbb{R}^d$ ← d features

$y \in \{0, 1\}$

x_1: Tumor size

x_2: Age

x_3: Uniformity of cell size

⋮

x_d: Uniformity of cell shape

2d) $x \in \mathbb{R}^d$

$y \in \{0, 1, 2, \ldots, k\}$

benign   type 1   type k

"Multiclass classification"

3

Example 3: Obama recognition

$x \in \{100 \times 100 \text{ pixel grayscale photos}\} = \mathbb{R}^{10000}$

In practice, would not use this representation

$y \in \{0, 1\}$

0: not obama

1: Obama

Note:

- Feature design: create measurements $x_1, x_2, \dots, x_d$ from actual input (picture of Obama, cell biopsy)
- Features should capture important aspects of learning task

Linear Regression

$x \in \mathbb{R}$

$y \in \mathbb{R}$

h linear $\Rightarrow h_w(x) = w_1 x + w_0$

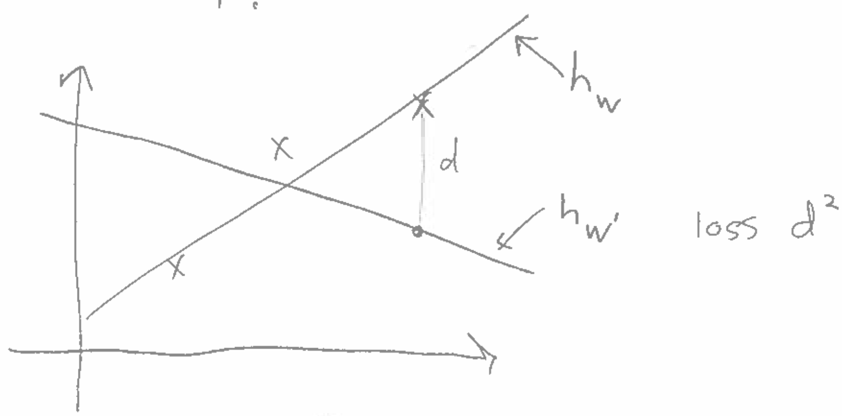
Input: $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N) \quad x_i, y_i \in \mathbb{R}$

Goal: Find "best" w_0 + w_1 so

$$y_i \approx h_w(x_i) \text{ for all } i$$

Cost function (aka Loss function)

- How can we decide between two hypotheses w_0 + w_1?



- Cost function $\Rightarrow$ numerical measure of error on training data
- Gauss (1777-1865) "squared error" or "L_2"
  - Cost on i^{th} training example $(y_i - h_w(x_i))^2$

truth   prediction

  - Overall cost

$$J(w_0, w_1) = \sum_{i=1}^{N} (y_i - h_w(x_i))^2$$

$$= \sum_{i=1}^{N} (y_i - (w_1 x_i + w_0))^2$$

Running
Example $(N=2)$

x_i	y	$w_1=0, w_0=0$ $h(x)$	loss	$w_1=1, w_0=0$ $h(x)$	loss	$w_1=2, w_0=0$ $h(x)$	loss
1	2	$0\cdot1+0=0$	$(0-2)^2=4$	$1\cdot1+0=1$	$(1-2)^2=1$	$2\cdot1+0=2$	$(2-2)^2=0$
2	3	$0\cdot2+0=0$	$(0-3)^2=9$	$1\cdot2+0=2$	$(2-3)^2=1$	$2\cdot2+0=4$	$(4-3)^2=1$
			13		2		1

But how can we minimize cost? optimization/ calculus

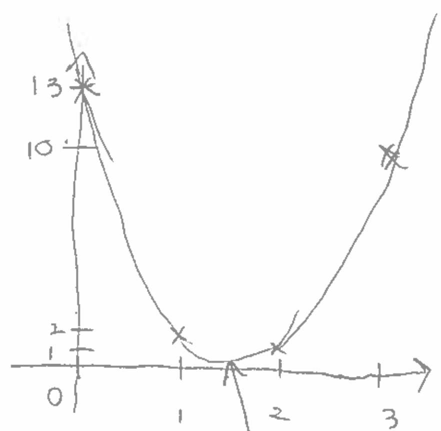
Simplification: assume $w_0=0$

$$h_w(x) = w_1 x$$

$$J(w_0,w_1) = J(w_1) = \sum_{i=1}^{N} (y_i - w_1 x_i)^2$$

Ex:

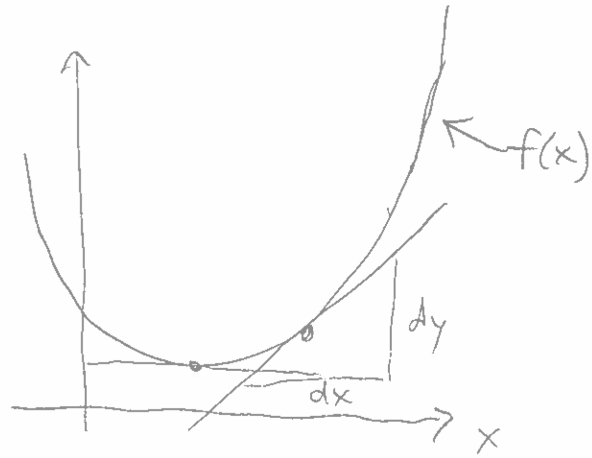
$$J(w_1) = (2-w_1\cdot 1)^2 + (3-w_1\cdot 2)^2 \leftarrow \text{quadratic in } w_1$$



$J(3) = 1+9=10$

minimum: find using calculus tools

Review of derivatives:



- Function $f(x)$
- Derivative $\frac{d}{dx} f(x)$ is slope of tangent line at x
- $\frac{d}{dx} f(x) = 0$ at (local) optima (minima/maxima)
- f convex $\Rightarrow$ one global minimum ($J(w_1)$ is convex)

Facts:

1) $\frac{d}{dx} x^k = k x^{k-1}$

2) $\frac{d}{dx}\left(af(x) + bg(x)\right) = a \frac{d}{dx} f(x) + b \frac{d}{dx} g(x)$

3) $\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot \frac{d}{dx} g(x)$ (chain rule)
$(f(g(x)))' = f'(g(x)) \cdot g'(x)$

Examples:

$\frac{d}{dx} 4x^3$

$\frac{d}{dx} (3x - 4)^2$

$\frac{d}{dx} (y_i - w_1 x_i)^2$

Example (cont!)

$$\min_{w_1} J(w_1) = (2-w_1)^2 + (3-2w_2)^2$$

$$\text{Set } 0 = \frac{d}{dw_1} J(w_1) = 2(2-w_1) + 2(3-2w_2)\cdot 2$$
$$= (2-w_1) + (6-4w_1)$$

$$\Rightarrow 5w_1 = 8$$
$$\Rightarrow w_1 = 8/5 = 1.6 \qquad \text{(YAY!)}$$

General case ($w_0 = 0$)

$$J(w_1) = \sum_{i=1}^{N} (y_i - w_1 x_i)^2$$

$$0 = \frac{d}{dw_1} J(w_1) = 2(y_1 - w_1 x_1)(-x_1) + 2(y_2 - w_1 x_2)(-x_2) + \dots + 2(y_N - w_1 x_N)(-x_N)$$

$$\Rightarrow 0 = (-x_1 y_1 + w_1 x_1^2) + (-x_2 y_2 + w_1 x_2^2) + \dots + (-x_N y_N + w_N x_N^2)$$

$$\Rightarrow x_1 y_1 + x_2 y_2 + \dots + x_N y_N = w_1 (x_1^2 + \dots + x_N^2)$$

$$\Rightarrow w_1 = \frac{x_1 y_1 + \dots + x_N y_N}{x_1^2 + \dots + x_N^2}$$

Check with example;

$$w_1 = \frac{2+6}{1+4} = \frac{8}{5} = 1.6 \qquad \text{(YAY!)}$$

Next time:

- minimize $J(w_0, w_1)$
- gradient descent