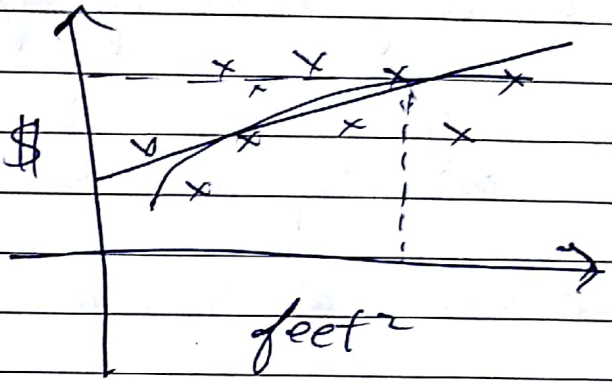
Machine Learning

#Vid I

Defn:- Arthur Samuel (1959)
Ability of computers to learn without being explicitly programmed

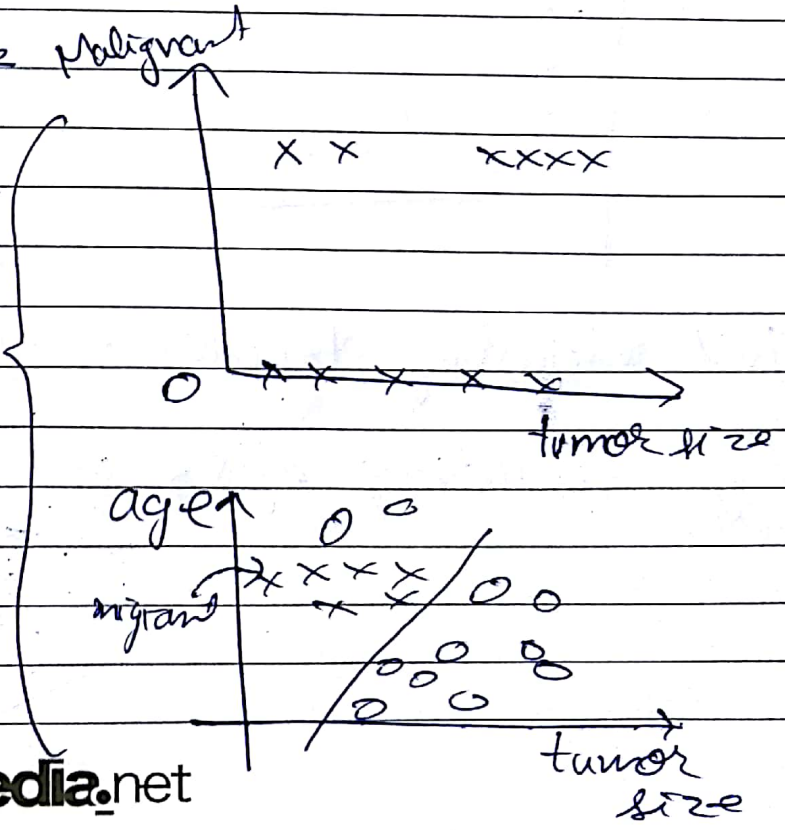
Tom Mitchell (1998) - Learn from exp E, Task T, Measure P, P increases for T with E.

Regression problem
↳ variable is continuous



Supervised learning
provided data with labels

Classification Problem
↳ variable is discrete



Supervised classification

Support Vector Machines

↳ infinite dimensional vector

data changed to ↑ ↳ of classification

Learning Theory

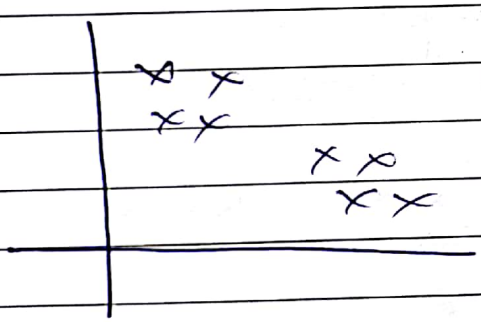
how & why learning algo. works?

E.g.

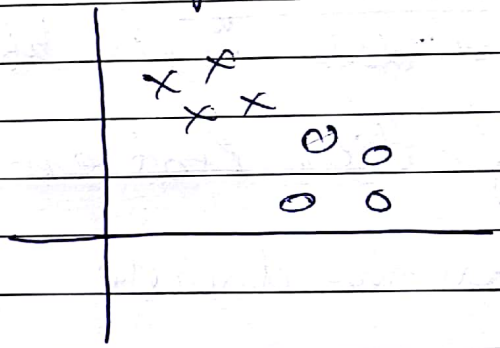
Theorem :- guaranteeing learning algo. with predict 99.9% correctly on reading of cells

Take tools of ML, and apply tools very craftfully.

Unsupervised learning



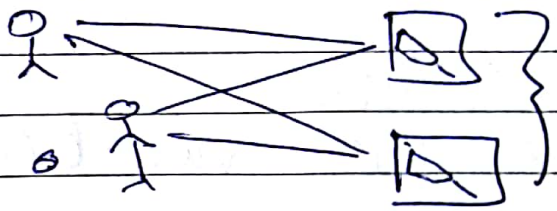
Supervised



Find interesting structure in the data

E.g. clustering algos in computing clusters
social network clusters

Cocktail party problem



Independent Component Analysis

~~#Vid 5~~ Reinforcement Learning

We have reward function. Give positive reward for desirable action else negative

#Vid 2 1994 - Autonomous driving

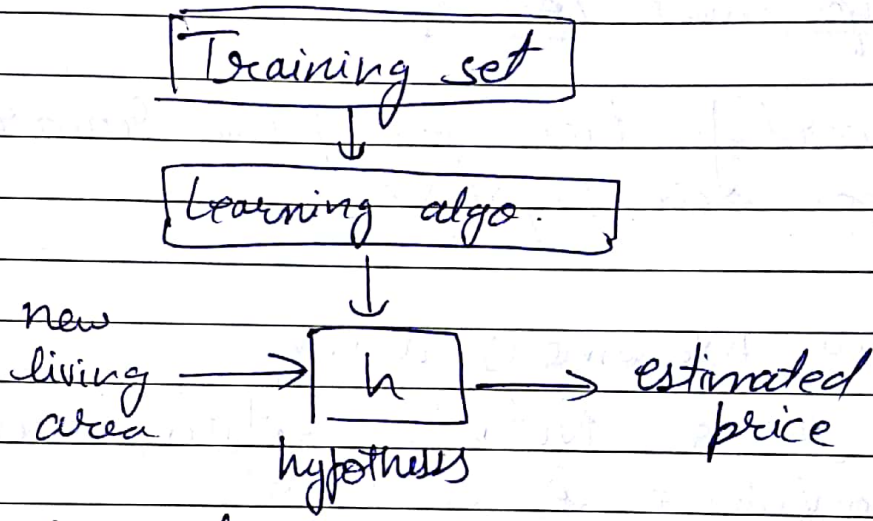
Alvin - Jeep driver - Network returns steering angle and confidence is returned.

1) ~~Regression~~ Problem

Living area (feet^2)	$ (1000s)
2104	400
1416	232
1534	315
852	178
1940	240

Notation

m = # training examples
x = "input" variables / features
y = "output" variable / "target" variable
(x, y) – training e.g.
i^{th} training example = $(x^{(i)}, y^{(i)})$



$h(x) = \cancel{h_\theta(x) = h} \; \theta_0 + \theta_1 x$

x_1 = size, x_2 = # bedrooms

$h(x) = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

For conciseness, $x_0 = 1$

$$h(x) = \sum_{i=0}^{2} \theta_i x_i = \theta^T x$$

n = # features

θ's are called parameters

Learning algo.s & find the value of θ.

Can θ be function of other variables? Yes

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$\min_{\theta}$ $J(\theta)$

Start with some θ. (Say $\theta = \vec{0}$)
Keep changing θ to reduce $J\theta$

Gradient Descent

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta)$$

For one training e.g.

$$\frac{\partial}{\partial \theta_i} J\theta = \frac{\partial}{\partial \theta_i} \frac{1}{2}\left(h_\theta(x) - y\right)^2$$

$$= 2 \cdot \frac{1}{2}\left(h_\theta(x) - y\right) \frac{\partial}{\partial \theta_i}\left(h_\theta(x) - y\right)$$

$$= \left(h_\theta(x) - y\right) \frac{\partial}{\partial \theta_i}\left(\theta_0 x_0 + \cdots \theta_n x_n - y\right)$$

$$= \left(h_\theta(x) - y\right) x_i$$

Repeat till convergence :-

$$\theta_i = \theta_i - \alpha \underbrace{\sum_{j=1}^{m} (h_\theta(x^{(j)}) - y^{(j)}) \cdot x_i^{(j)}}_{\frac{\partial}{\partial \theta_i} J(\theta)}$$

For sum of squares, $J(\theta)$ is a bowl shaped with one local minima

Batch Gradient Descent

Look at entire training set, entire m instances are used for every update.

Stochastic Gradient Descent

Repeat {
  for $j = 1$ to m {
    $\theta_i := \theta_i - \alpha (h_\theta(x^{(j)}) - y^{(j)}) \cdot x_i^{(j)}$
  }
}

→ Works faster than ~~entire~~ batch gradient descent

$$\nabla_\theta J = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix} \in \mathbb{R}^{n+1}$$

Gradient Descent:

$$\theta := \theta - \alpha \nabla_\theta J$$

$\theta \in \mathbb{R}^{n+1}$, $\nabla_\theta J \in \mathbb{R}^{n+1}$

$$f : \mathbb{R}^{m \times n} \longmapsto \mathbb{R} \qquad f(A) \quad A \in \mathbb{R}^{m \times n}$$

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

If $A \in \mathbb{R}^{n \times n}$

$$\text{tr} A = \sum_{i=1}^{n} A_{ii}$$

Fact:

$$\text{tr} AB = \text{tr} BA$$

$$\text{tr} ABC = \text{tr} CAB = \text{tr} BCA$$

If $f(A) = \text{tr}(AB)$ $\nabla_A \text{tr} AB = B^T$

If $a \in \mathbb{R}$, $\text{tr} a = a$

$$X\theta = \begin{bmatrix} \text{---} & (x^{(1)})^T & \text{---} \\ \text{---} & (x^{(2)})^T & \text{---} \\ & \vdots & \\ \text{---} & (x^{(m)})^T & \text{---} \end{bmatrix} \theta$$

$$X\theta = \begin{bmatrix} (x^{(1)})^T\theta \\ x^{(2)T}\theta \\ \vdots \\ x^{(m)T}\theta \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) \\ \vdots \\ h_\theta(x^{(m)}) \end{bmatrix}$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$$X\theta - y = \begin{bmatrix} h(x^{(1)}) - y^{(1)} \\ \vdots \\ h(x^{(m)}) - y^m \end{bmatrix}$$

$$\frac{1}{2}(X\theta - y)^T(X\theta - y) = \frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2 = J(\theta)$$

$$\nabla_\theta J(\theta) \overset{set}{=} \vec{0}$$

$$\nabla_\theta \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

$$= \frac{1}{2}\nabla_\theta tr\left(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y\right)$$

So trace ⇒ Real number is introduced

$y^T y$ → Independent of θ

$$= \frac{1}{2}\left[\nabla_\theta tr\,\theta\theta^T X^T X - \nabla_\theta tr\, y^T X\theta\right.$$

$\theta\theta^T X^T X$ → applied cyclic prop. of trace

$y^T X\theta$ → transpose of real is same real

$$\left. - \nabla_\theta y^T X\theta\right]$$

$$\nabla_\theta tr\, \underset{A}{\theta}\underset{B}{I}\underset{A^T}{\theta^T}\underset{C}{X^T X} = \underset{C}{X^T X}\underset{A}{\theta}\underset{B}{I} + \underset{C^T}{X^T X}\underset{A}{\theta}\underset{B^T}{I}$$

$$\nabla_\theta tr\, \underset{B}{y^T}\underset{A}{X\theta} = \underset{B^T}{X^T y}$$

$$\nabla_\theta J(\theta) = \frac{1}{2}\left[X^T X\theta + X^T X\theta - X^T y - X^T y\right]$$

$$= X^T X\theta - X^T y \overset{set}{=} 0$$

$$\Rightarrow X^T X\theta = X^T y$$

$$\Rightarrow \theta = (X^T X)^{-1} X^T y$$

~~# Vid 2~~ # Vid 3

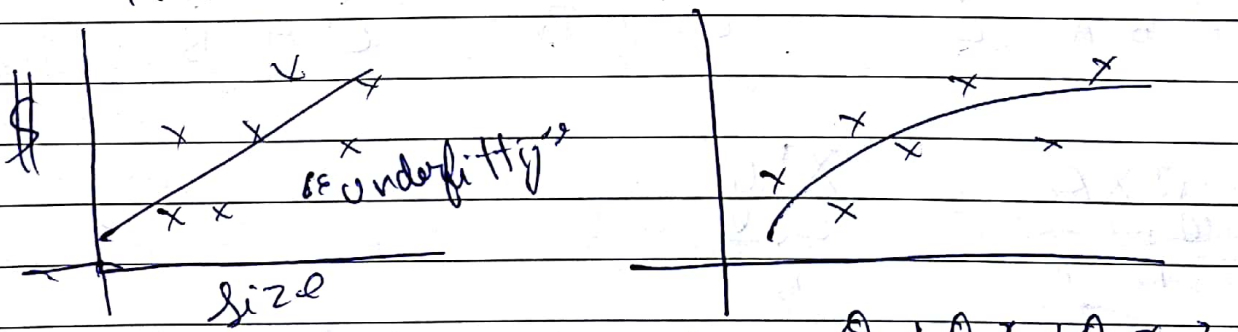
Outline

Linear regression
↓ → locally weighted regression
probabilistic interpretation
↓
logistic regression
↓ → Digression Perceptron
Perceptron

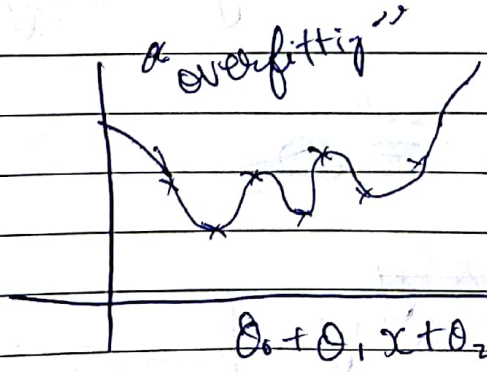
$(x^{(i)}, y^{(i)})$ – i^{th} training example

Choice of features – has large impact on learning algorithms

x_1 = size x_2 = (size)^2



$\theta_0 + \theta_1 x + \theta_2 x_2^2$



$\theta_0 + \theta_1 x + \theta_2 x_2 + \dots + \theta_6 x^6$

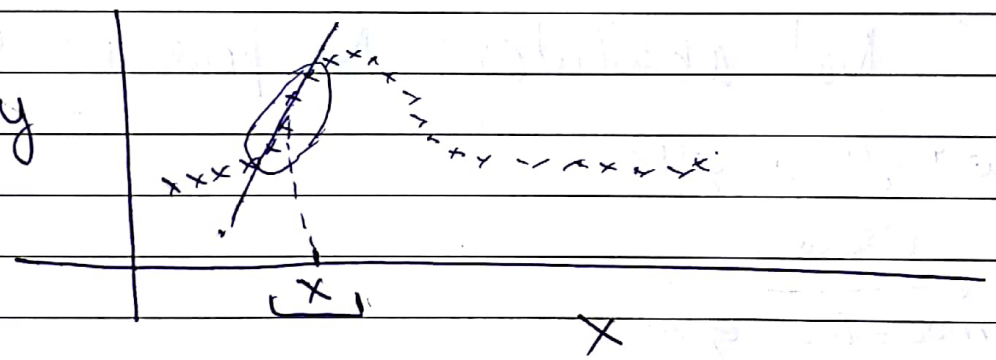
"Parametric" learning algorithm

number of parameters - fixed e.g. linear regression

"Non-parametric" learning algorithm

- no. of parameters grows with m

Locally weighted regression (Loess / Lowess)



To evaluate h at a certain x

For predicting at X, we only consider in vicinity of x

LWR: Fit θ to minimize

$$\sum_i w^{(i)} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

where $w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$

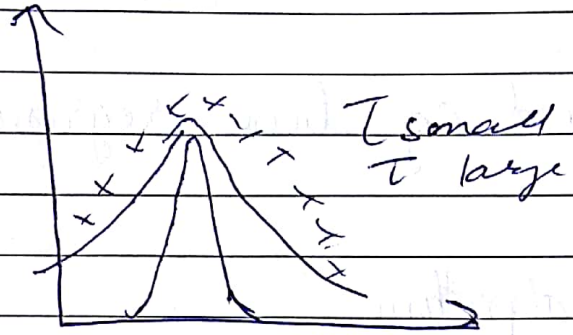
τ - bandwidth parameter

→ don't interpret it as gaussian function

→ don't associate semantic of gaussian

If $|x^{(i)} - x|$ small, then $w^{(i)} \simeq 1$

If $|x^{(i)} - x|$ large then $w^{(i)} \simeq 0$



Andrew More on KD Trees

Probabilistic interpretation

Why squares? Not absolutes? Not powers of 4?

Assume $y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$

$\varepsilon^{(i)}$ = error

unmodeled effects + random noise

$\varepsilon^i \sim \mathcal{N}(0, \sigma^2)$ → Normal / Gaussian dist.

↓ Distributed

$$P(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\varepsilon^{(i)})^2}{2\sigma^2}\right)$$

$$P\left(y^{(i)} | x^{(i)}; \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$y^{(i)} | x^{(i)}; \theta \sim \mathcal{N}(\theta^T x, \sigma^2)$$

Why gaussian?

1) Mathematically convenient
2) Central Limit Theorem
   → Sum of many RVs (independent) is apptox. gaussian

Assumptions we make are never absolutely true

$P(y^i | x^i; \theta)$ → Notation :- θ is not a RV.
∴ $P(y^i | x^i, \theta)$ is wrong

Frequentist viewpt.

$\varepsilon^{(i)}$s are IID – independently identically distributed.

$$L(\theta) = \boxed{P(\vec{y} | X; \theta)}$$

Likelihood emphasis the boxed term as function of θ
Likelihood of parameter ⇔ Probability of data

$$= \prod_{i=1}^{m} P(y^{(i)} | x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

Maximum likelihood:

Choose θ to maximize $L(\theta) = P(\vec{y} | X; \theta)$

$$\ell(\theta) = \log(L(\theta))$$

$$= \log \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right]$$

$$= \sum_{i=1}^{m} \log\left[\frac{1}{\sqrt{2\pi}\sigma} \exp(\cdots)\right]$$

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} -\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}$$

So maximizing $\ell(\theta)$ is the same as minimizing

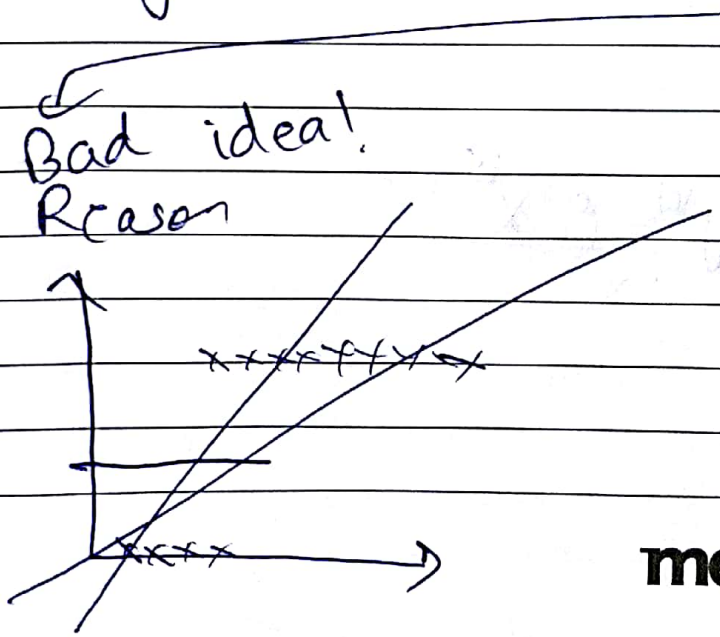
$$\sum_{i=1}^{m} \frac{(y^{(i)} - \theta^T x)^2}{2} = J(\theta)$$

Classification

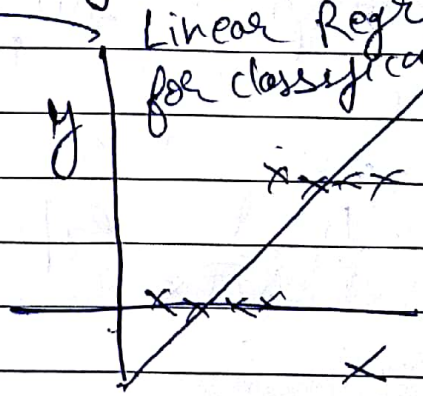
target variable is discrete $y \in \{0, 1\}$

Bad idea!

Reason



Linear Regression for classification



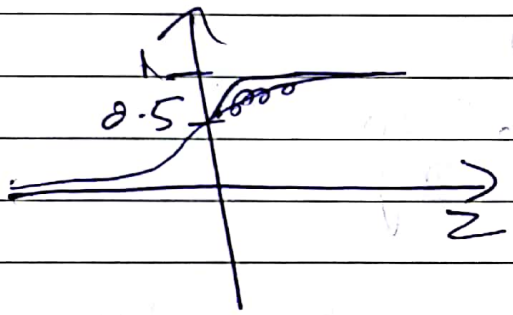
$y \in \{0, 1\}$

$h_\theta(x) \in [0, 1]$

Choose

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$g(z) = \frac{1}{1 + e^{-z}}$$



$g(z)$ – sigmoid function,
logistic function

From where?

1) ~~Generalized Linear Models~~

2)

$$P(y = 1 | x; \theta) = h_\theta(x)$$

$$P(y = 0 | x; \theta) = 1 - h_\theta(x)$$

$$P(y | x; \theta) = h_\theta(x)^y \left(1 - h_\theta(x)\right)^{1-y}$$

If $y = 1$, $h_\theta(x)$

$y = 0$, $1 - h_\theta(x)$

$$L(\theta) = P(\vec{y} \mid X; \theta) = \prod_i P(y^{(i)} \mid x^{(i)}; \theta)$$
$$= \prod_i h(x^{(i)})^{y^{(i)}} (1 - h(x^{(i)}))^{1 - y^{(i)}}$$

$$l(\theta) = \log L(\theta)$$
$$= \sum_{i=1}^{m} \left[y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log (1 - h(x^{(i)})) \right]$$

$$\theta := \theta + \alpha \nabla_\theta l(\theta)$$

$$\frac{\partial l(\theta)}{\partial \theta_j} = \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

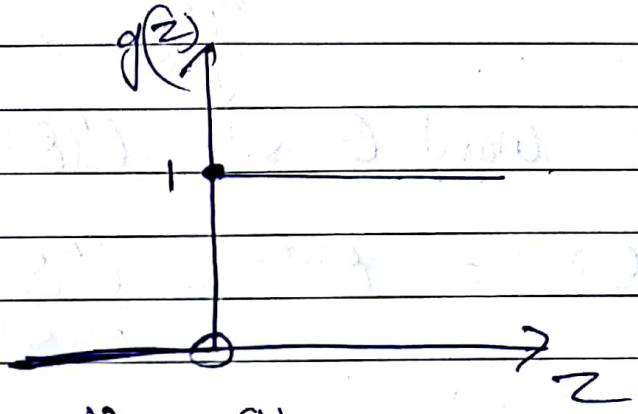
Diff. from that of linear regression

Generalized linear model will show that even if we are using entirely diff. model, update eqn. will look cosmetically similar

Cover up the derivations & prove them
— Andrew Ng

Digression : Perceptron

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



$$h_\theta(x) = g(\theta^T x)$$

$$\theta_j = \theta_j + \alpha\left(y^{(i)} - h(\theta^T x^{(i)})\right) x_j^{(i)}$$

Vid 4

Today
Logistic Regression
- Newton's Method
Exponential Family
Generalized Linear Models (GLMs)

Find θ s.t. $f(\theta) = 0$

$$f'(\theta^{(0)}) = \frac{f(\theta^{(0)})}{\Delta}$$

$$\Delta = \frac{f(\theta^{(0)})}{f'(\theta^{(0)})}$$