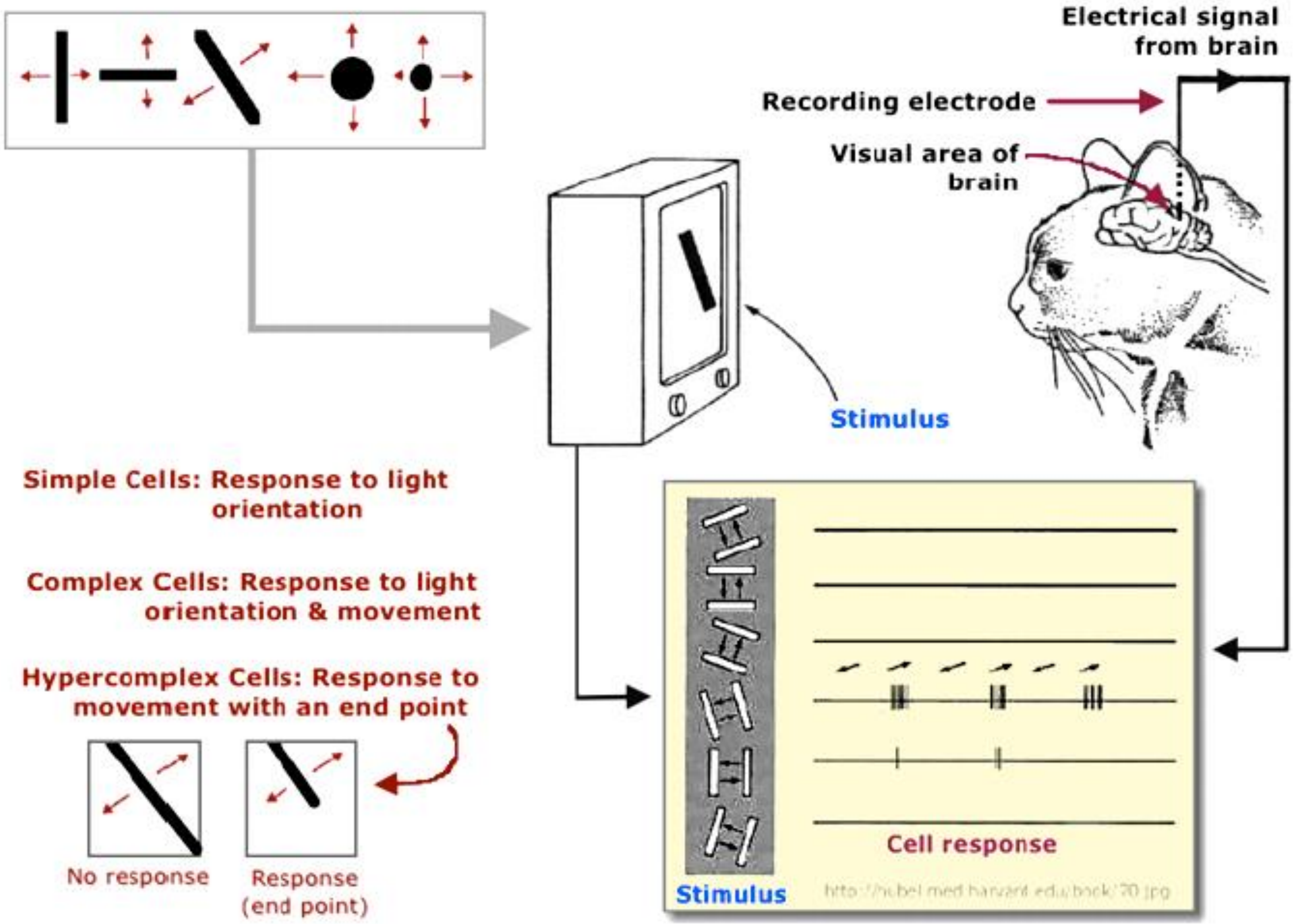
Introduction

Machine Learning Fundamentals



Hubel & Wiesel, 1959

Machine Learning

Field of study that gives computers the ability to learn without being explicitly programmed.

By: Arthur Samuel (1959)

Applications

- Database mining:
  - Web click data, medical records, biology, engineering etc.
- Application that can’t be programmed by hand
  - Autonomous helicopter, handwriting recognition, NLP, computer vision
- Self Customizing Programs Recommendation
  - Amazon, Netflix
- Understanding human learning
  - Brain, real AI

Learning Problem

- Well-posed learning problem:

  A computer program is said to *learn* from *experience* E with respect to some *task* T and some *performance measure* P, if its performance on T, as measured by P, improves with *experience* E.

- By Tome Mitchell (1998)

Example

Your Email program watches which emails you do or do not mark as a spam and based on that learns how to better filter spam.

- Task (T): Classifying the emails as spam or not
- Experience (E): Watching you label emails as spam or not spam
- Performance (P): The number of emails correctly classified as spam / not spam

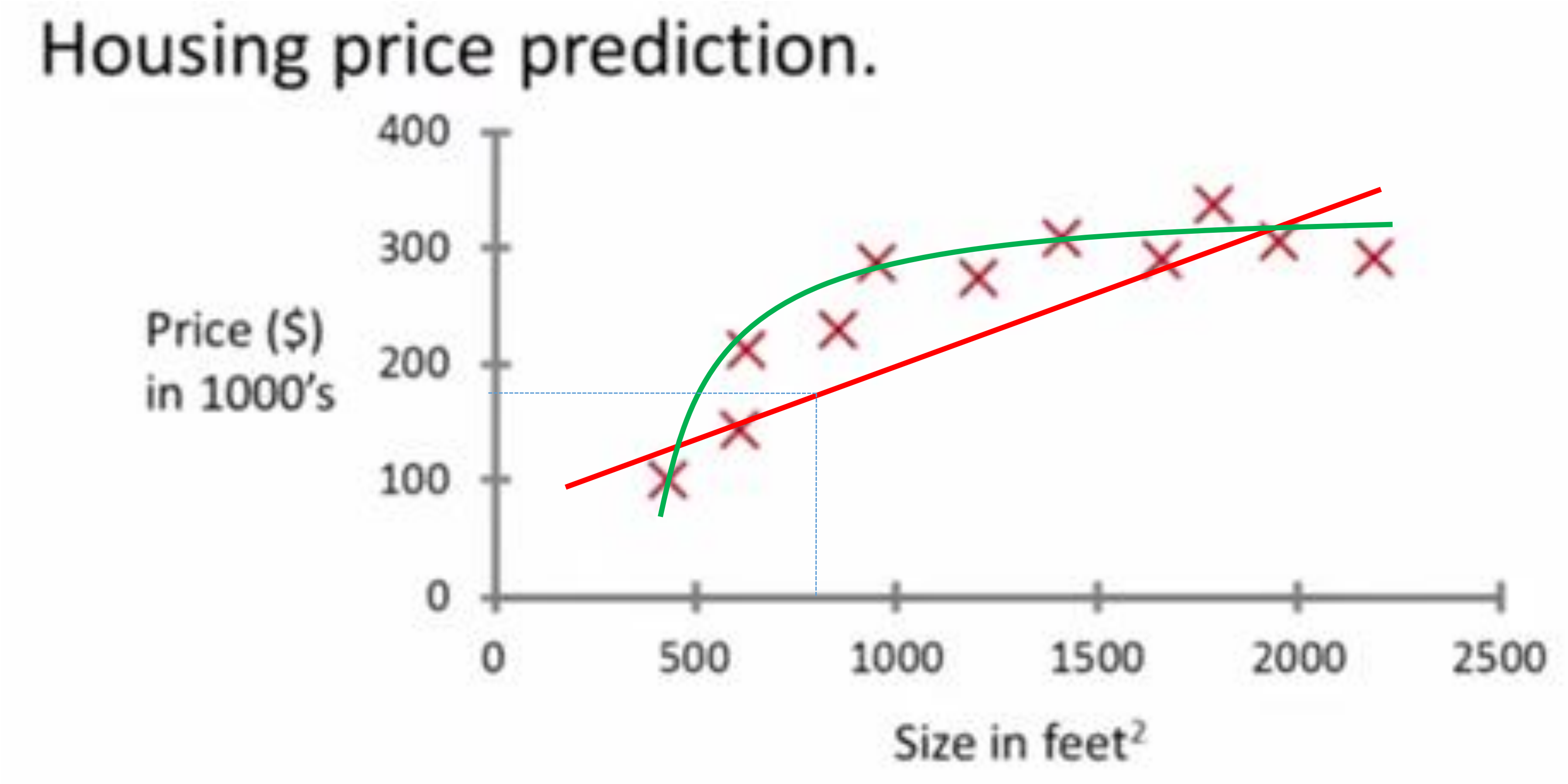
Machine Learning Algorithms

- Supervised Learning
- Un-supervised Learning

- Others: Reinforcement learning, recommender system

- Where to apply which algorithms

Supervised Learning

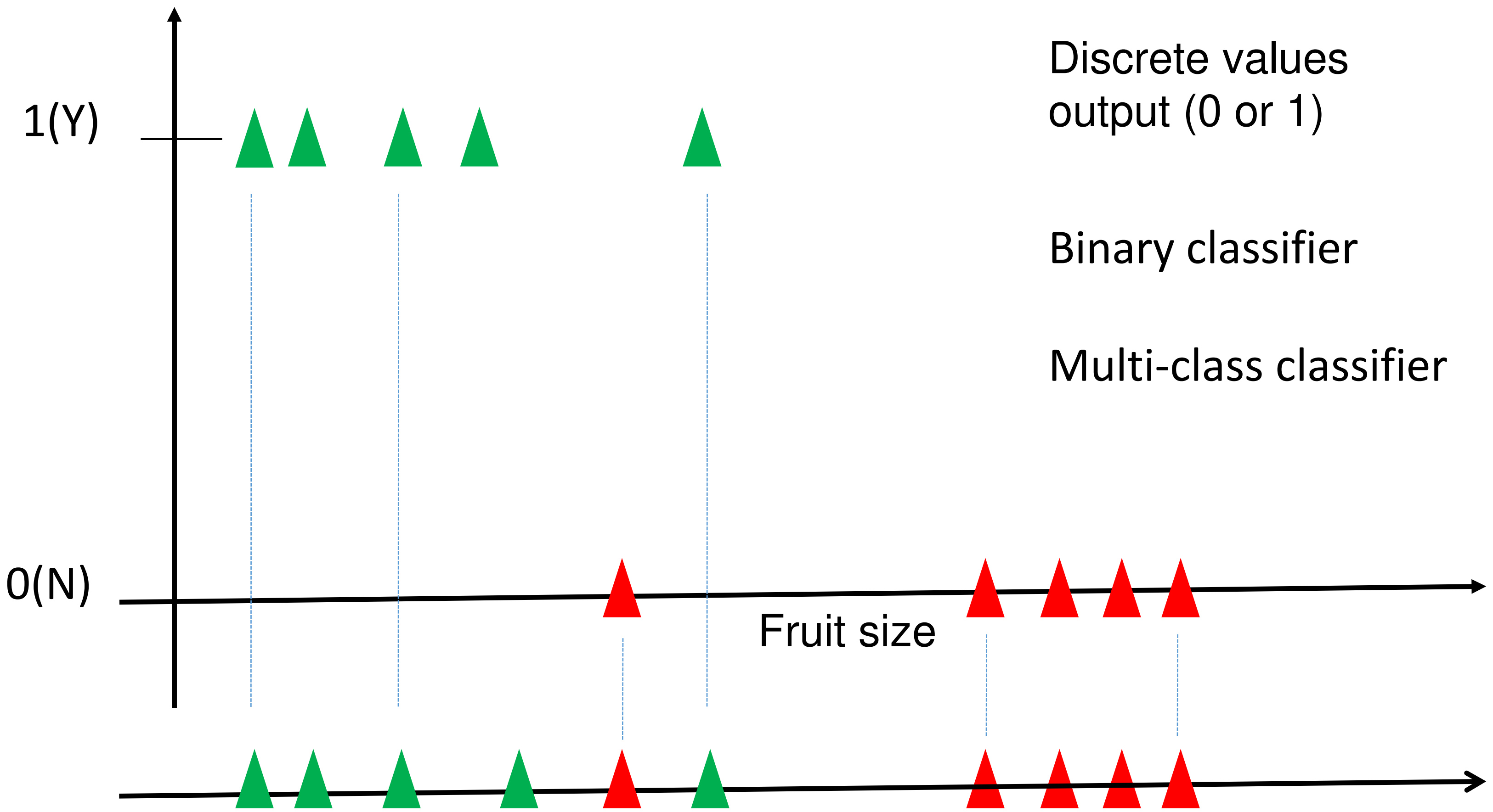


Supervised Learning:
Right answer given

Regression:
Predict continuous valued output

Supervised Learning

Fruit : Apple or not?



Classification

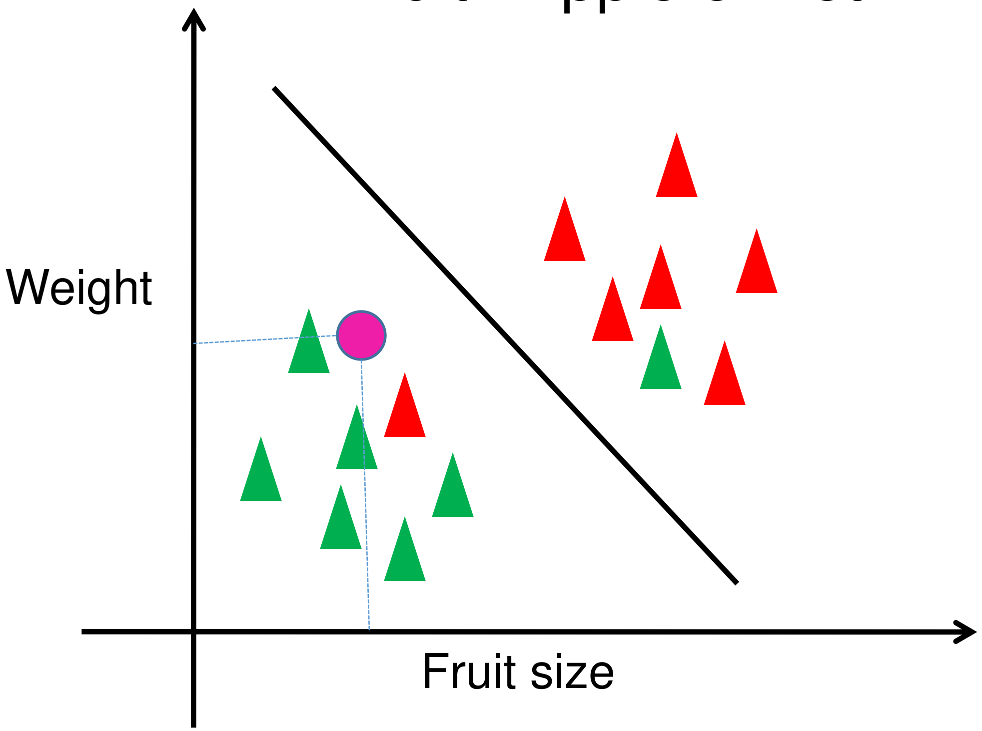
Discrete values output (0 or 1)

Binary classifier

Multi-class classifier

Supervised Learning

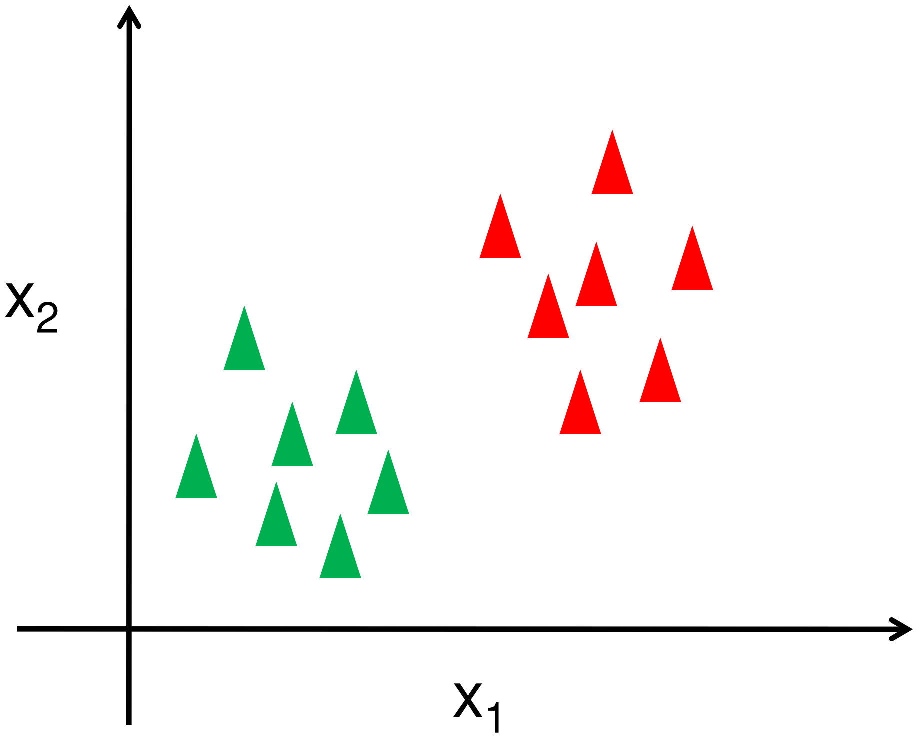
Fruit : Apple or not?



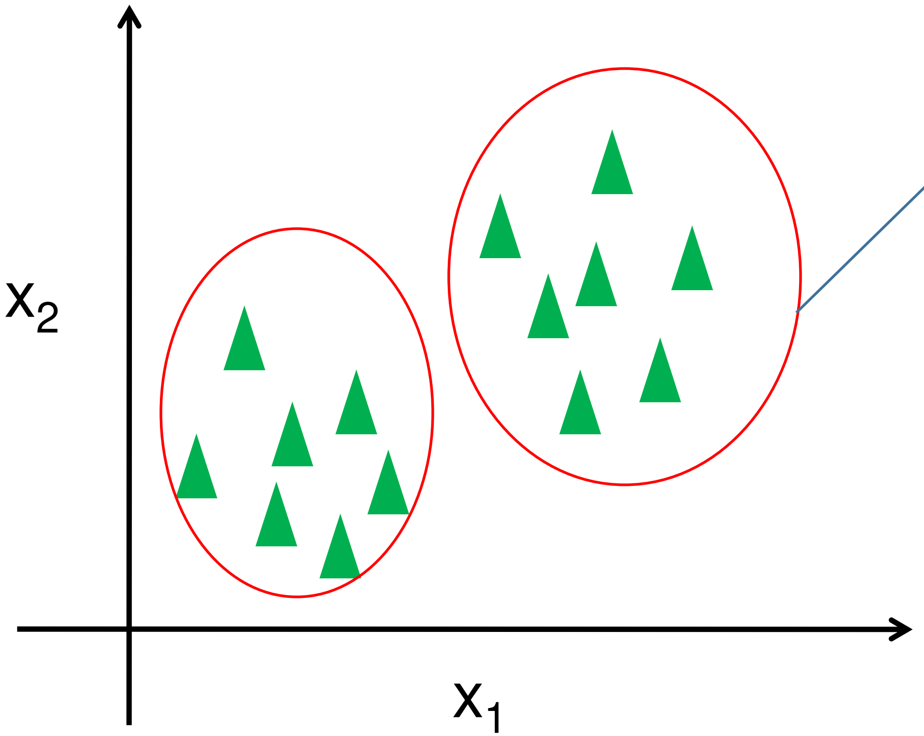
Two features examples

- Features:
  - color
  - texture
  - cost
  - …..

Supervised Learning



Un-supervised Learning



No label

Clustering Algorithm

In a supervised learning model, the algorithm learns on a labeled dataset, providing an answer key that the algorithm can use to evaluate its accuracy on training data. An unsupervised model, in contrast, provides unlabeled data that the algorithm tries to make sense of by extracting features and patterns on its own.


news.google.co.in
Top Stories
IPL Live Score RCB vs CSK: RCB struggle against CSK in Ranchi
The Indian Express - 35 minutes ago
Catch IPL LIVE score here. Latest IPL 8 news: IPL 2015, RCB vs CSK- IPL LIVE scorecard. RCB struggle against CSK. By: Web Desk | May 22, 2015 9:07 pm.
CSK vs RCB - IPL Qualifier 2 Live: Raina gets the big wicket of Gayle, Bangalore ... Hindustan Times
IPL 8 Qualifier 2 CSK VS RCB Live: CSK totally bog down RCB Firstpost
Opinion: Commentary: CSK vs. RCB The Hindu
Live Updating: IPL 2015 Qualifier 2: Chennai Super Kings vs Royal Challengers Bangalore ... Zee News
Related
Chennai Super Kings »
Indian Premier League »
Royal Challengers Bangalore »
See realtime coverage
Saudi-led coalition pounds Yemen rebels in three cities
Times of India - 37 minutes ago
SANAA: Warplanes from the Saudi-led coalition pounded Shiite rebels across three Yemeni cities on Friday, as Riyadh reported the death of a Saudi child from cross-border fire.
Tanu Weds Manu Returns 'Outstanding,' Tweets Bollywood After Film's Release
NDTV - 1 hour ago
With her 'swagger' in Tanu Weds Manu Returns, Kangana Ranaut has once again proved why she's called the 'Queen' of Bollywood.
India »
NDA government deaf to farmers' agony: Congress
Editors' Picks
The Economic Times
Nestle swings into damage control, says no further order to recall Maggi...
Major global banks admit guilt in forex probe, fined $6 bn
Everybody united against us, but we have public support: Arvind Kejriwal
Mukesh Ambani keeps salary capped for seventh year in a row
Big retailers file case against Centre demanding level playing field on...
Spotlight »
Karnataka 2nd PUC Results 2015 Declared at karresults.nic.in
Most popular

IPL Live Score RCB vs CSK: RCB struggle against CSK in Ranchi

The Indian Express - 35 minutes ago

Catch IPL LIVE score here. Latest IPL 8 news: IPL 2015, RCB vs CSK- IPL LIVE scorecard. RCB struggle against CSK. By: Web Desk | May 22, 2015 9:07 pm

CSK vs RCB - IPL Qualifier 2 Live: Raina gets the big wicket of Gayle, Bangalore ... Hindustan Times

IPL 8 Qualifier 2 CSK VS RCB Live: CSK totally bog down RCB Firstpost

Opinion: Commentary: CSK vs. RCB The Hindu

Live Updating: IPL 2015 Qualifier 2: Chennai Super Kings vs Royal Challengers Bangalore ... Zee News

Related

Chennai Super Kings »

Indian Premier League »

Royal Challengers Bangalore »

See realtime coverage

YouTube | Star Sports | Hindustan ... | IBNLive | Deccan He... | Sportskeeda | Internation...

follow us:

hindustantimes

india | world | cities | opinion | sports | entertainment | lifestyle | tech | photos | videos

ipl 2015 | throwback thursday | cannes 2015 | unclog noida | movie reviews

IPL Qualifier 2: 5 things to watch out for in Bangalore vs Chennai match

Manoj Bhagavatula and Abhimanyu Kulkarni, Hindustan Times, New Delhi | Updated: May 22, 2015 18:51 IST

Chennai Super Kings (CSK) players celebrate their win over Royal Challengers Bangalore (RCB) during their IPL 2015 match at MA Chidambaram Stadium in Chepauk, Chennai. (PTI Photo)

INTERNATIONAL BUSINESS TIMES

News | Business | Technology | Sport | Entertainment | Life & Style

BID2travel India's 1st Hotel Bid... Decide Your Own Hotel P...

Sports | Cricket | IPL 2015

IPL 2015: We Played one of our Most Perfect Games, says Mumbai Indians All-rounder Kieron Pollard

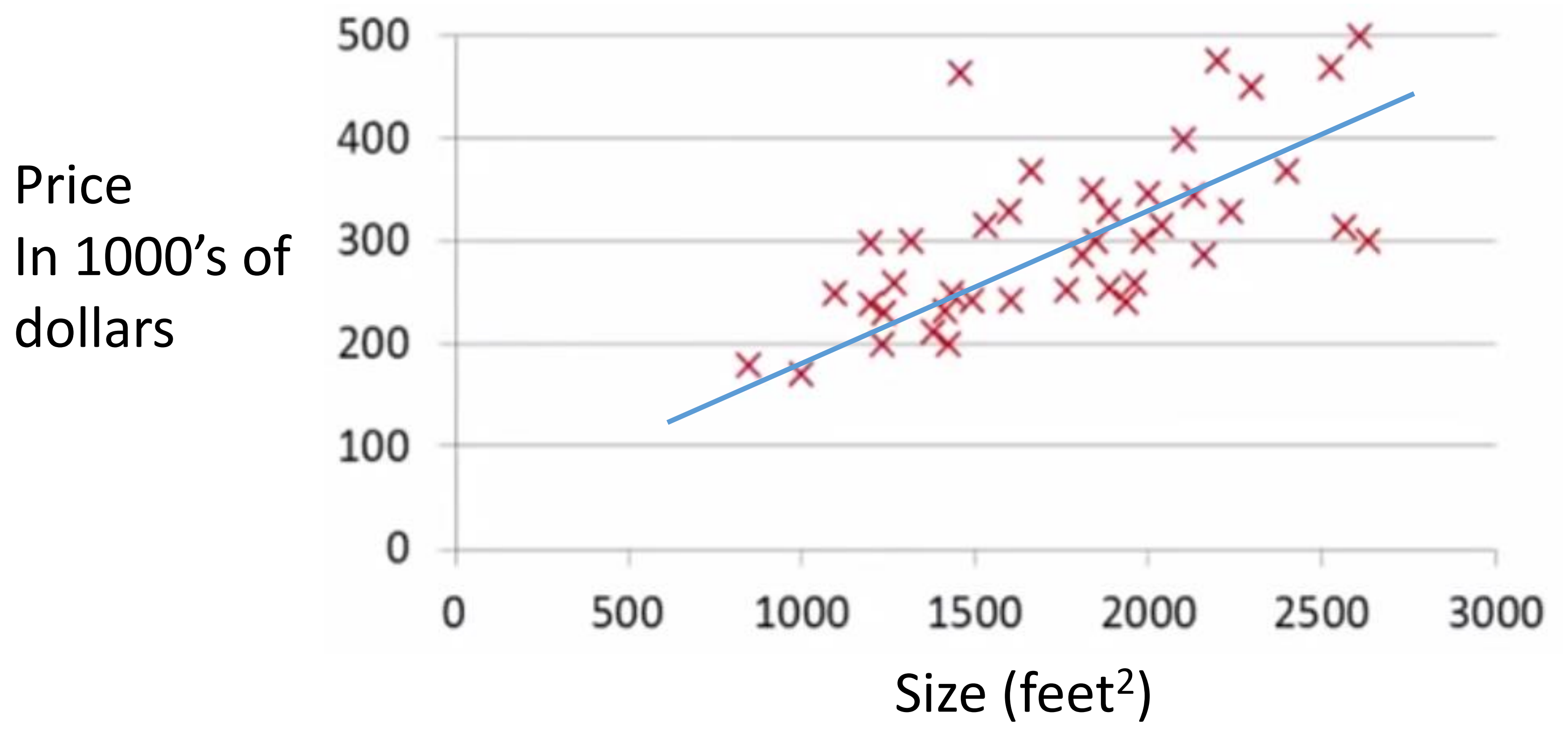
By Rajarshi Majumdar May 20, 2015 12:31 IST

Application of Clustering Algorithms

- Organizing computer clusters
- Social network analysis
- Market segmentation
- Astronomical image/data analysis
- Speaker recognition and many more…

Supervised Learning

Supervised Learning



- Given the right answer for each example of the data
  - Classification: discrete no. of outputs
  - Regression: Predict real valued data

Supervised Learning

Training set of housing prices

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

- **m** = Number of training examples
- **x**'s = "input" variable / features
- **y**'s = "output" variable / "target" variable

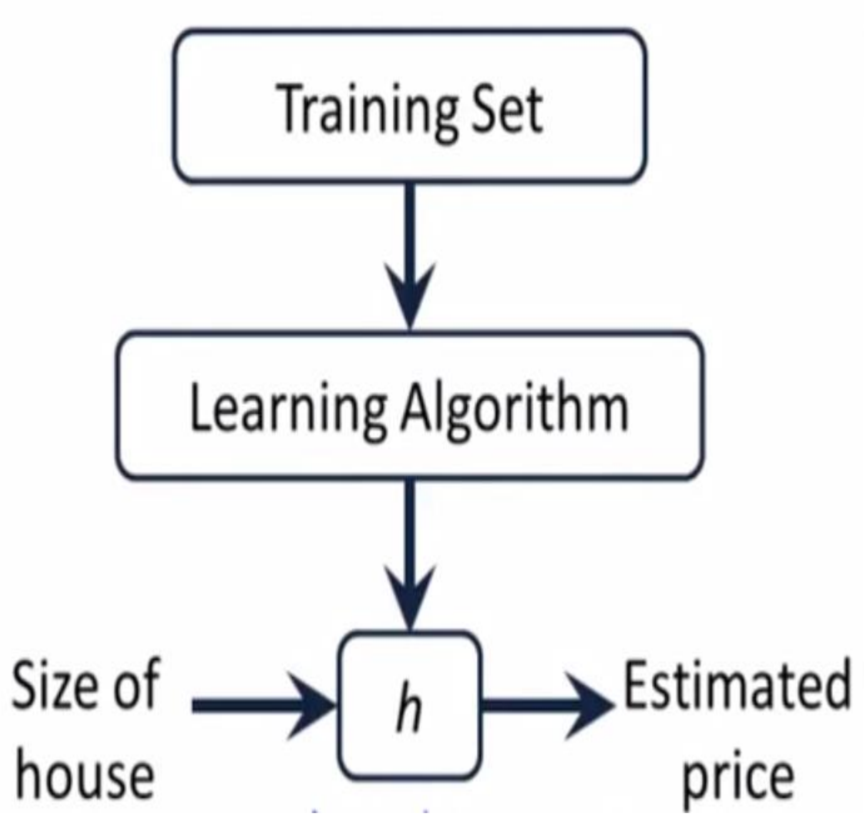
(x,y) → one training example

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

$x^{(i)} = 2104$

$y^{(i)} = 460$

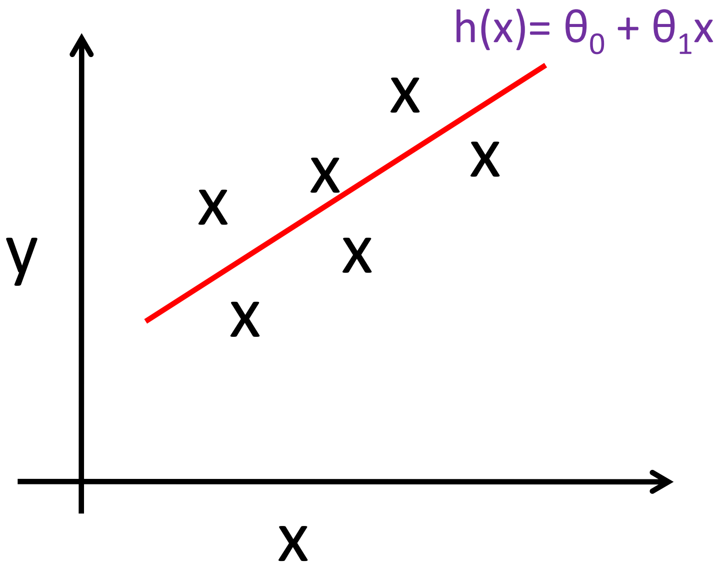
Supervised Learning



x hypothesis Estimated value of y

How do we represent h

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



Univariate linear regression: linear regression with one variable

Cost Function

Training Set

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

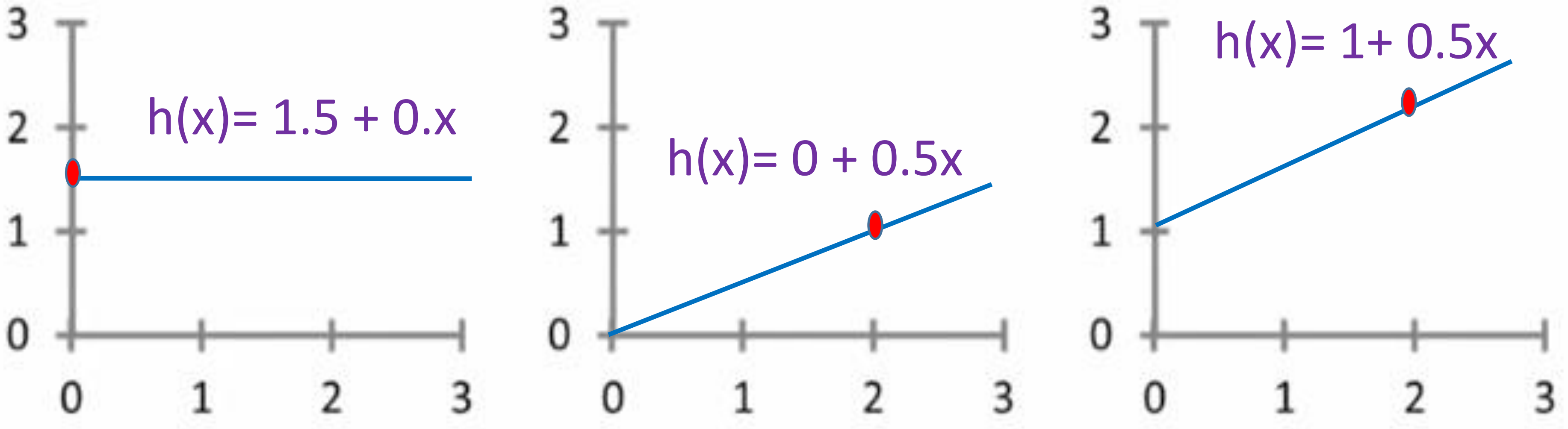
θ_i's → Parameters

How to choose θ_i's

Cost Function

Hypothesis Function:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

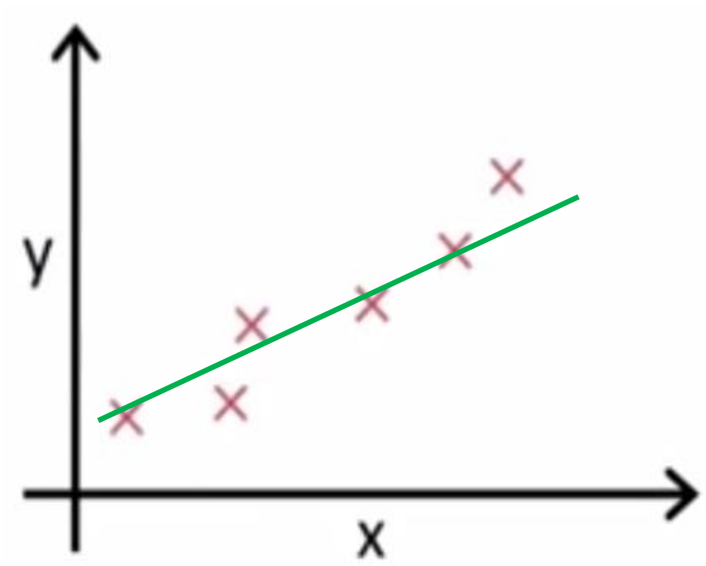


$\theta_0 = 1.5$
$\theta_1 = 0$

$\theta_0 = 0$
$\theta_1 = 0.5$

$\theta_0 = 1$
$\theta_1 = 0.5$

Cost Function



Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

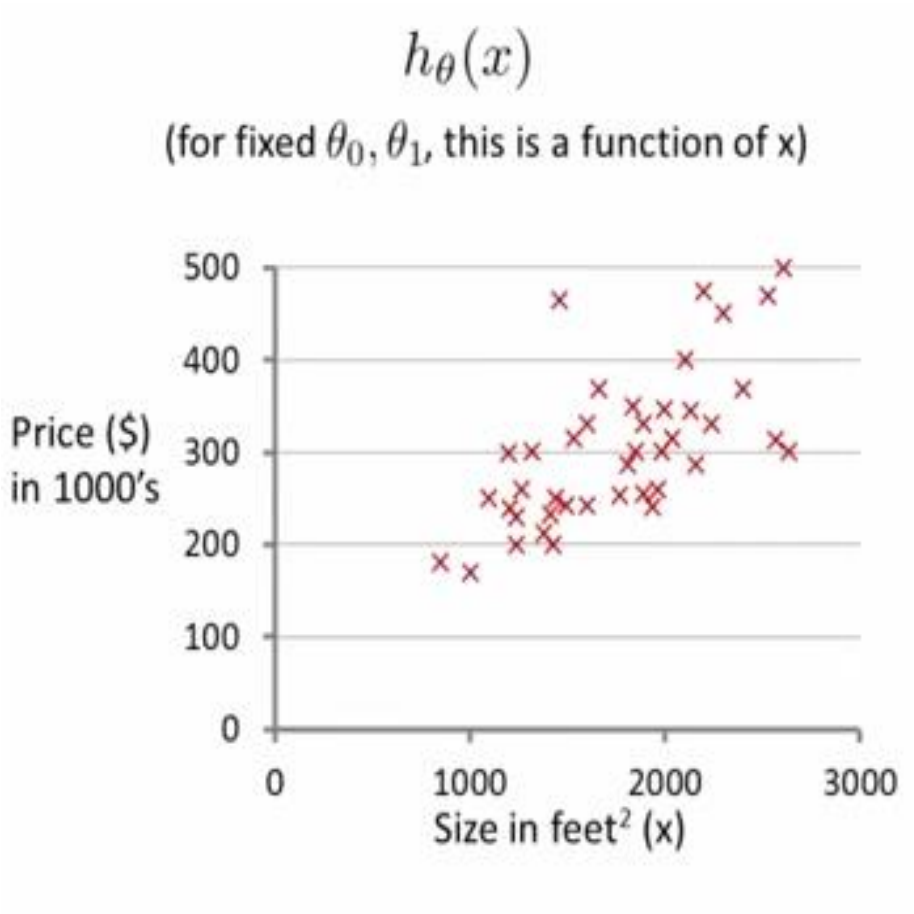
Squared error function

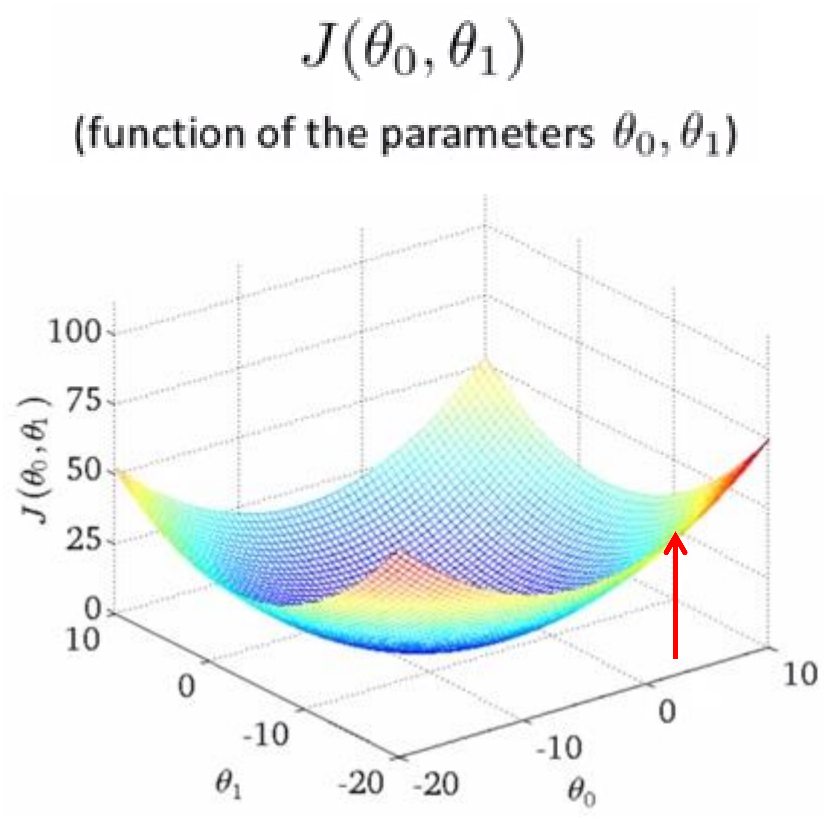
Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y)

m = No. of training samples

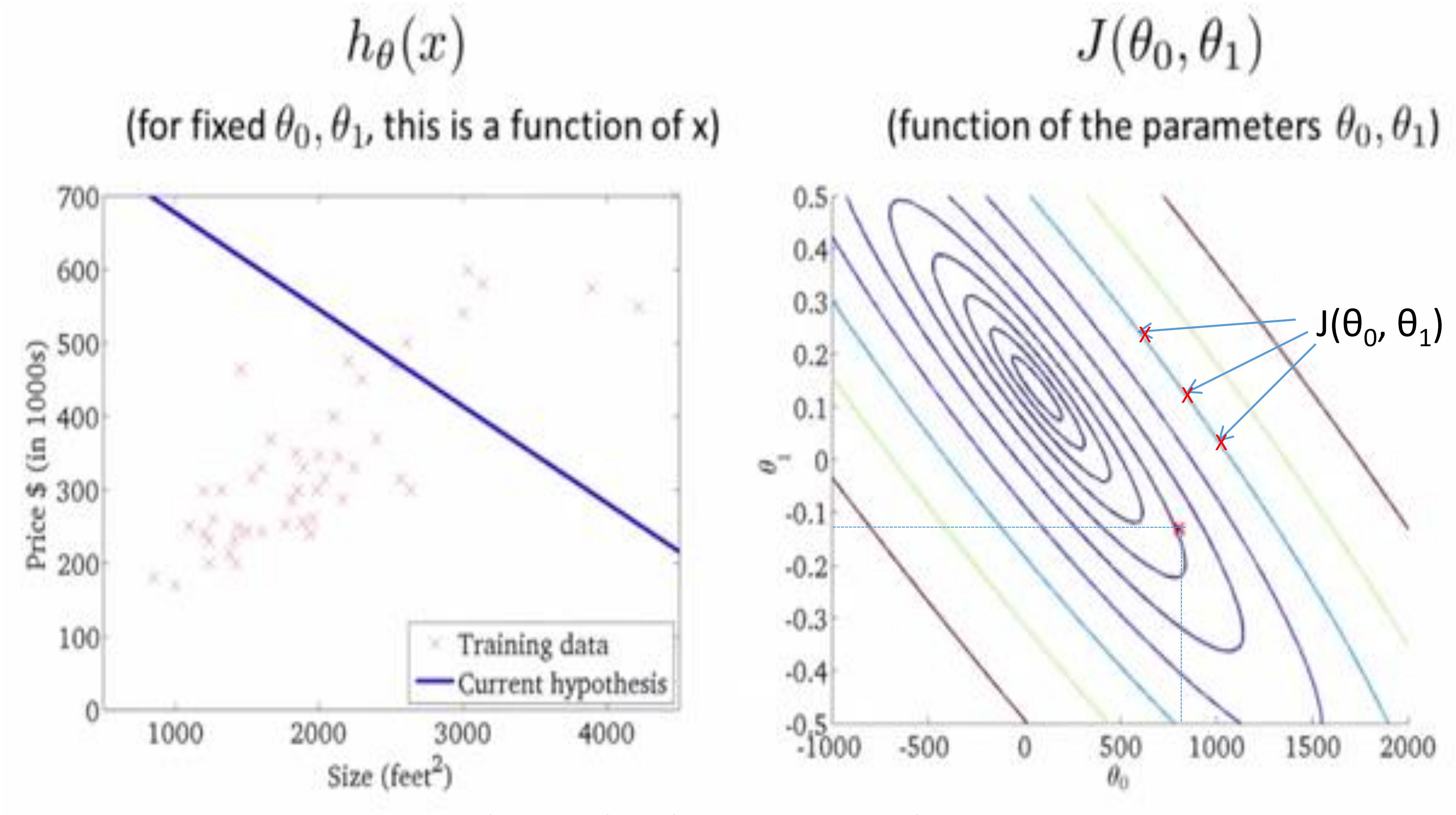
Cost Function





$J(\theta_0, \theta_1)$= value of the height of the surface

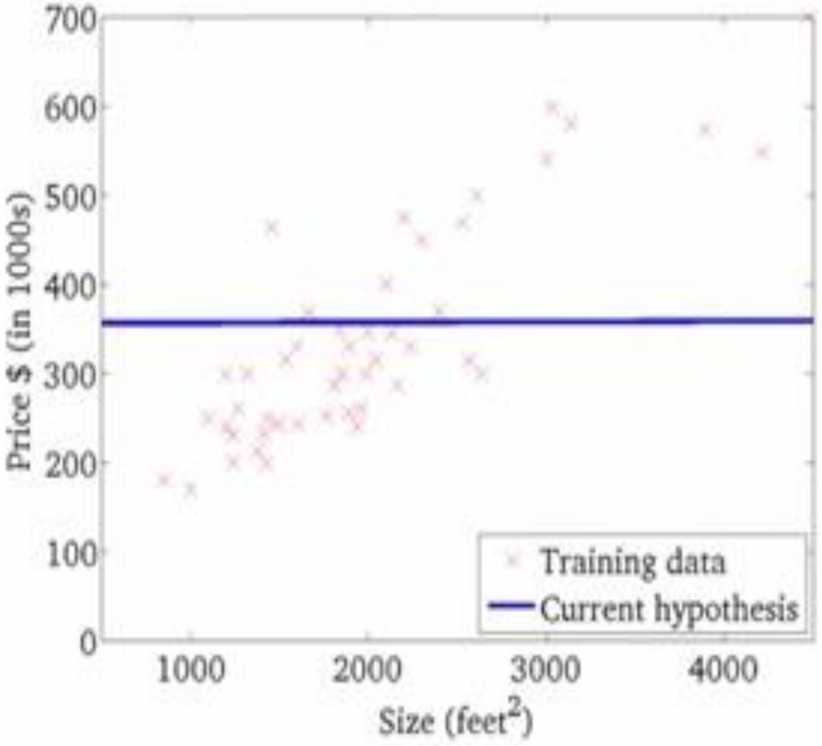
Contour Plots / Figures



$(\theta_0, \theta_1) = (800, -0.125)$

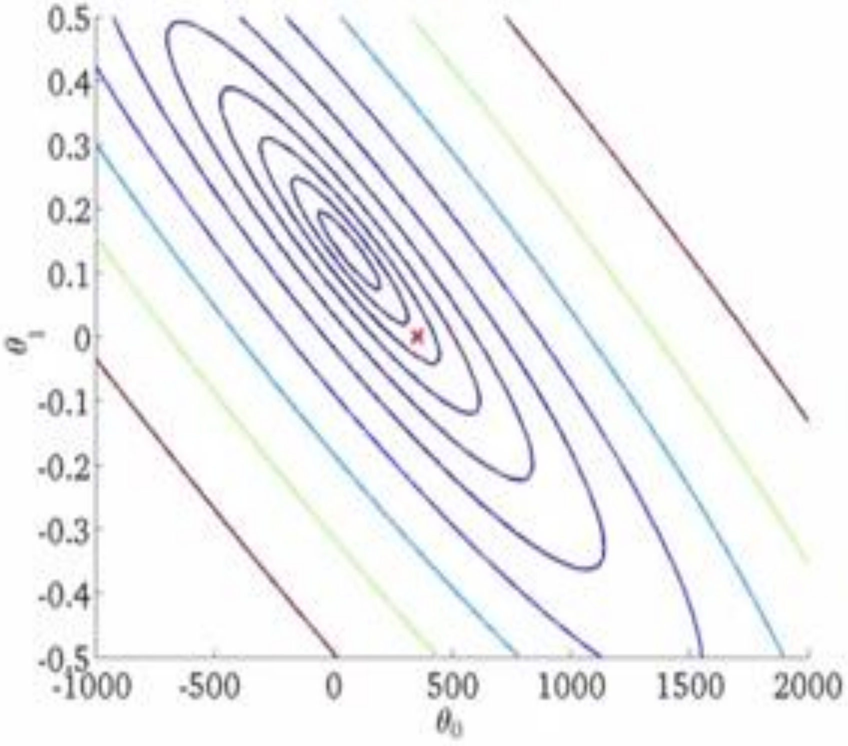
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

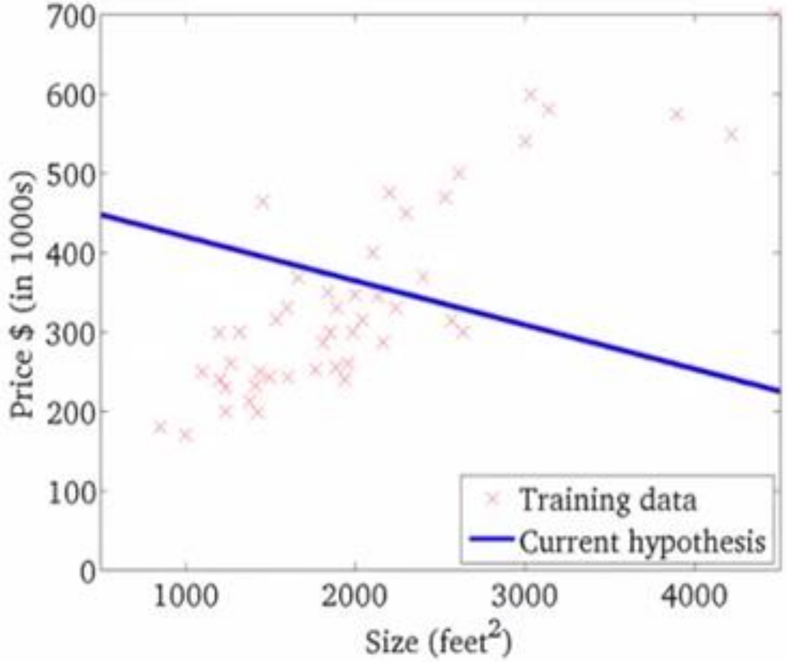
(function of the parameters θ_0, θ_1)



$(\theta_0, \theta_1) = (360, 0)$

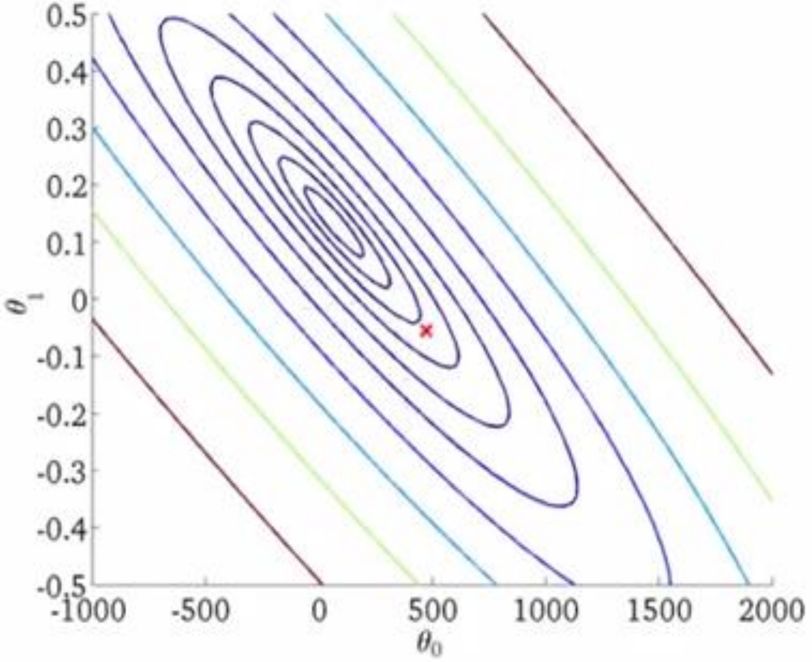
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



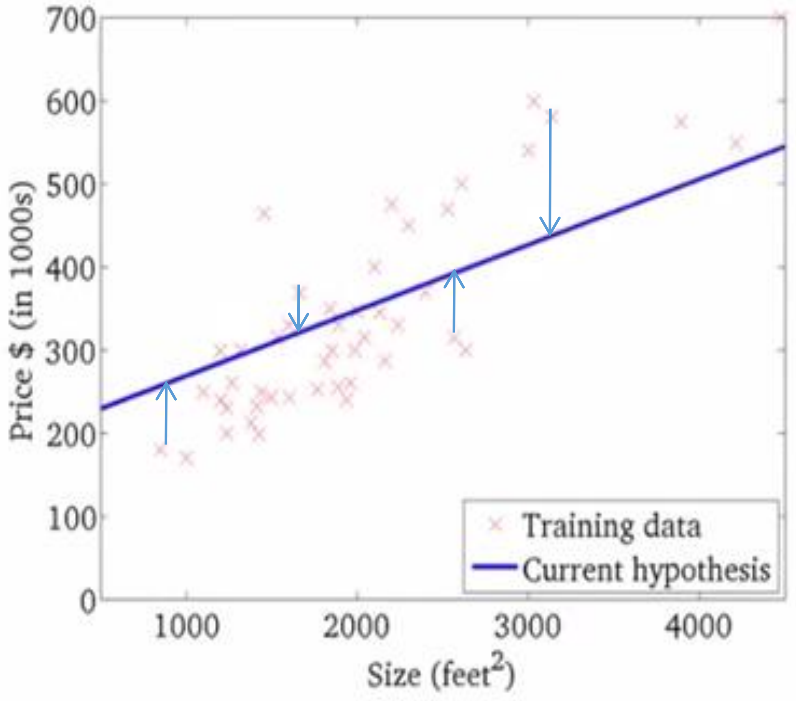
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



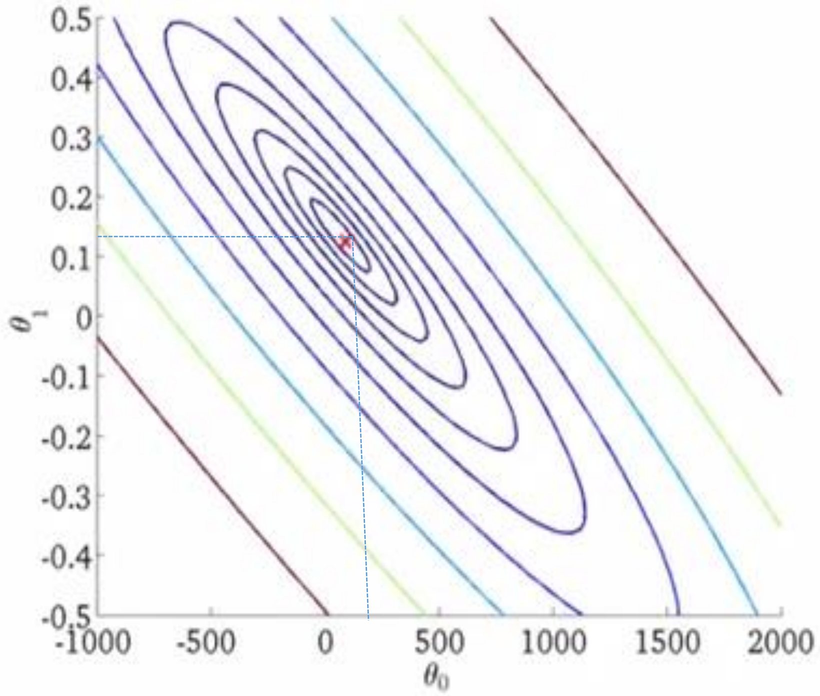
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

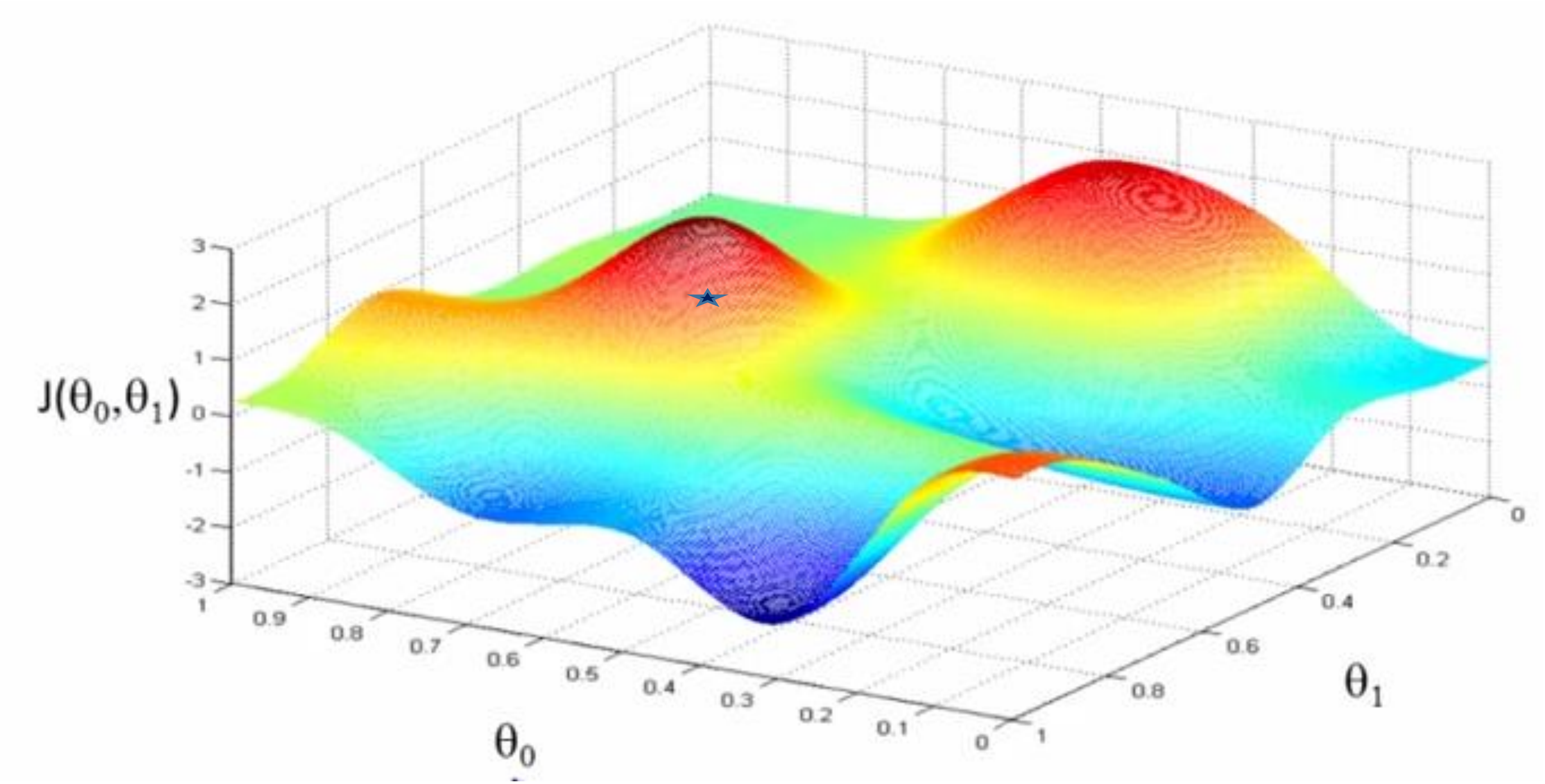
(function of the parameters θ_0, θ_1)



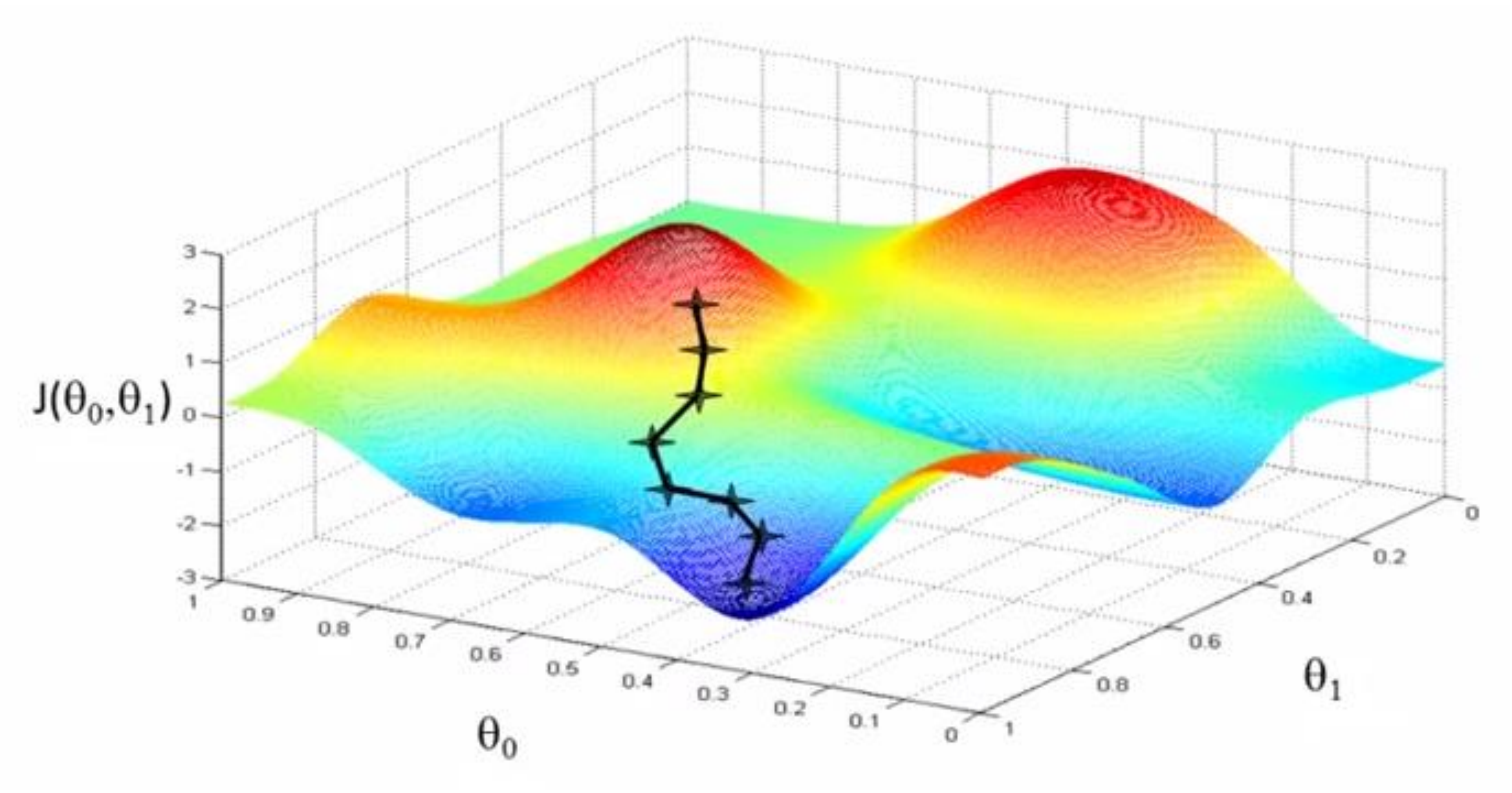
Gradient Descent

- Let some function $J(\theta_0, \theta_1)$
- We have to find $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$
- Start with some (θ_0, θ_1) (let say θ_0=0, θ_1=0)
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

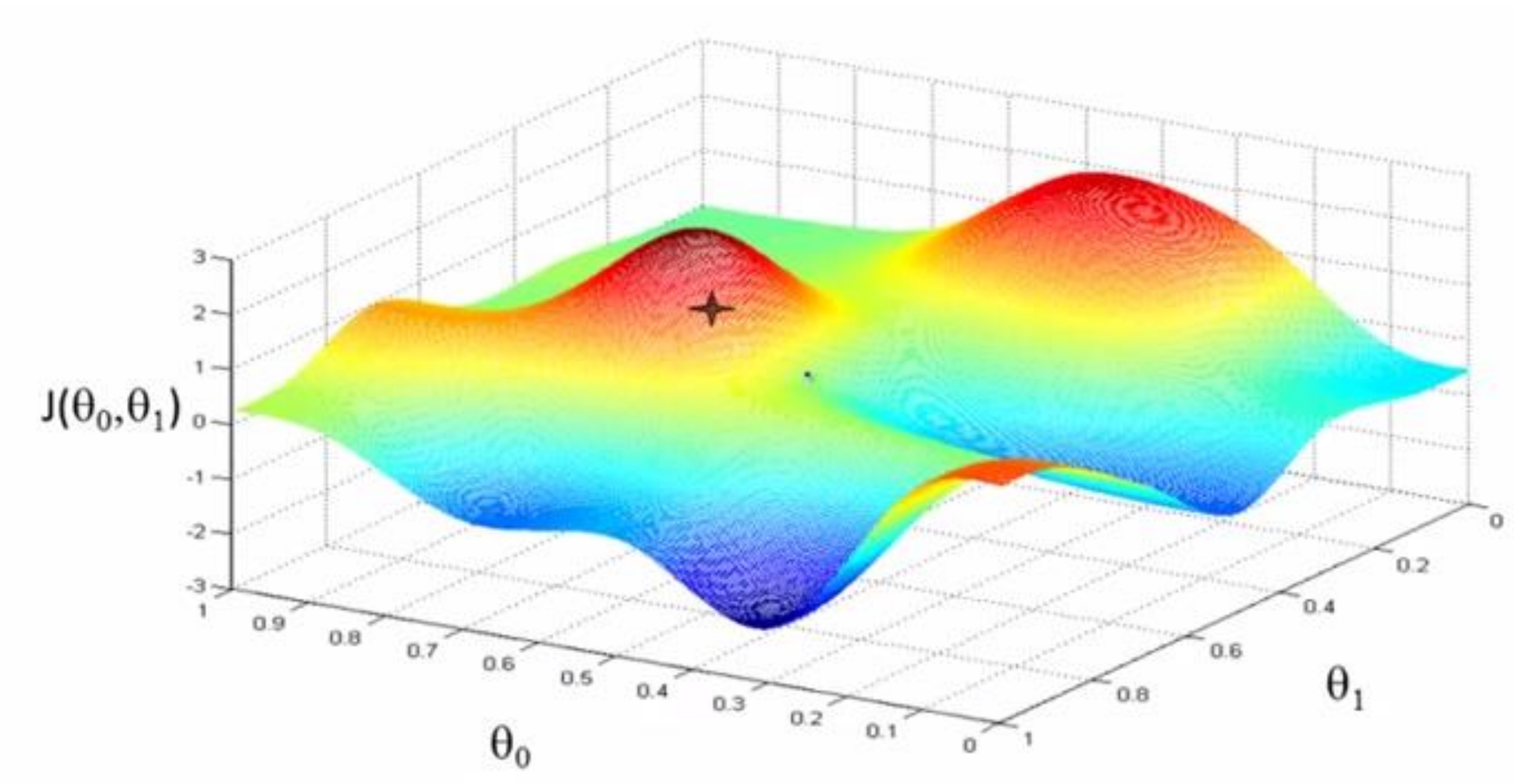
Gradient Descent



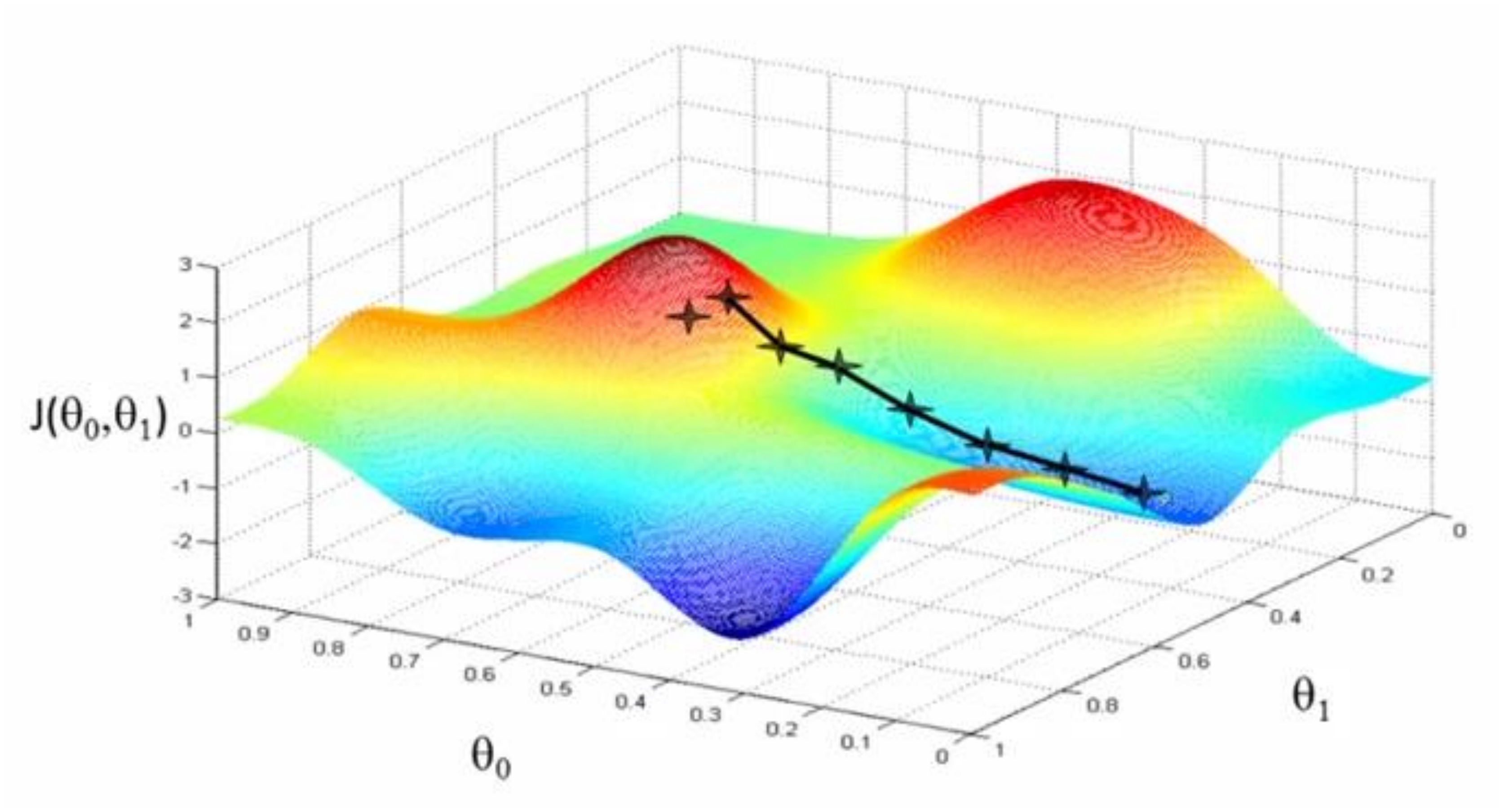
Gradient Descent



Gradient Descent



Gradient Descent



Gradient Descent Algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1)$$

}

α = learning rate

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad j \in [0,1]$$

Implication of α = it controls how bigger steps we are taking over gradient descent

✓ Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

✗ Incorrect:

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

Gradient Descent Algorithm

- Let take a single variable

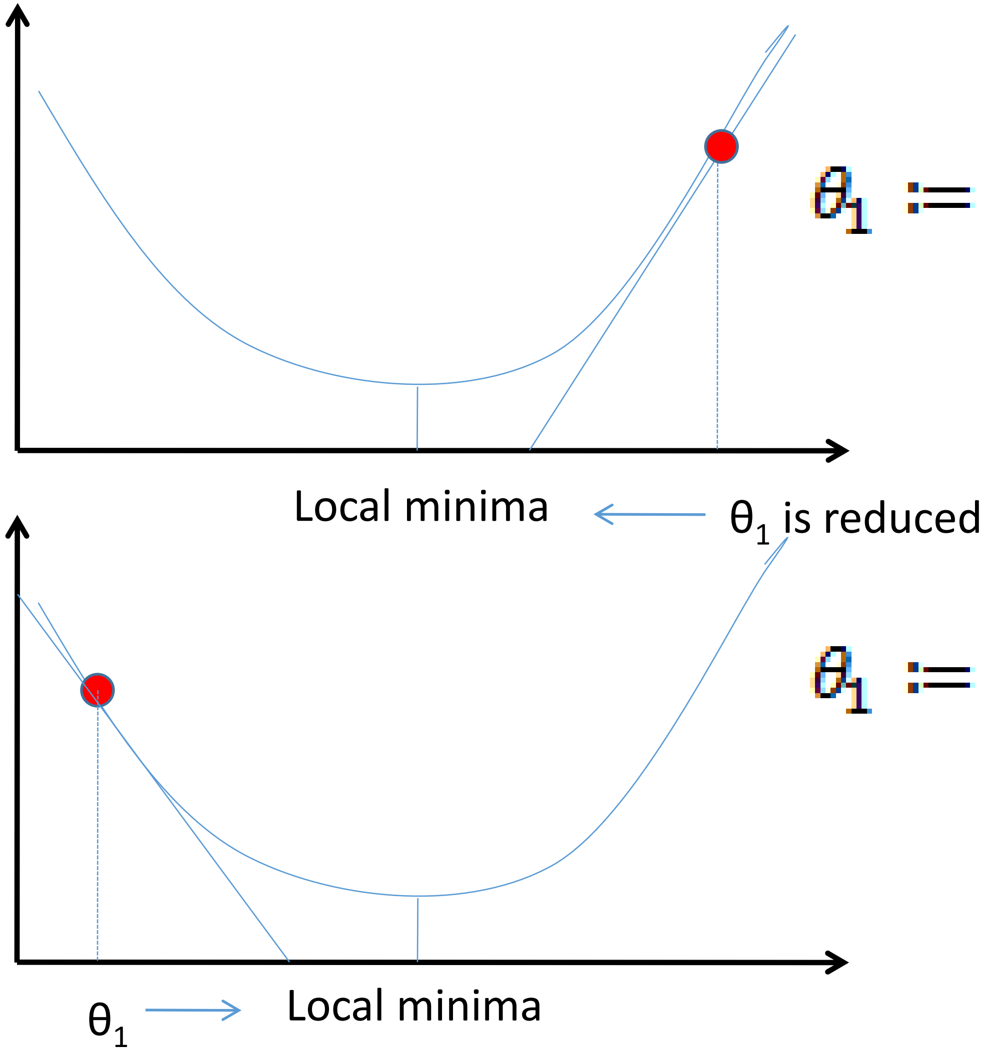
- we have to minimize $\min_{\theta_1} J(\theta_1)$

  where $\theta_1 \in R$

- So the GD algorithm becomes

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Gradient Descent Algorithm



$$\theta_1 := \theta_1 - \alpha \boxed{\frac{\partial}{\partial \theta_1} J(\theta_1)}$$

≥ 0

(slope is positive)

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{\partial}{\partial \theta_1} J(\theta_1)}$$

≤ 0

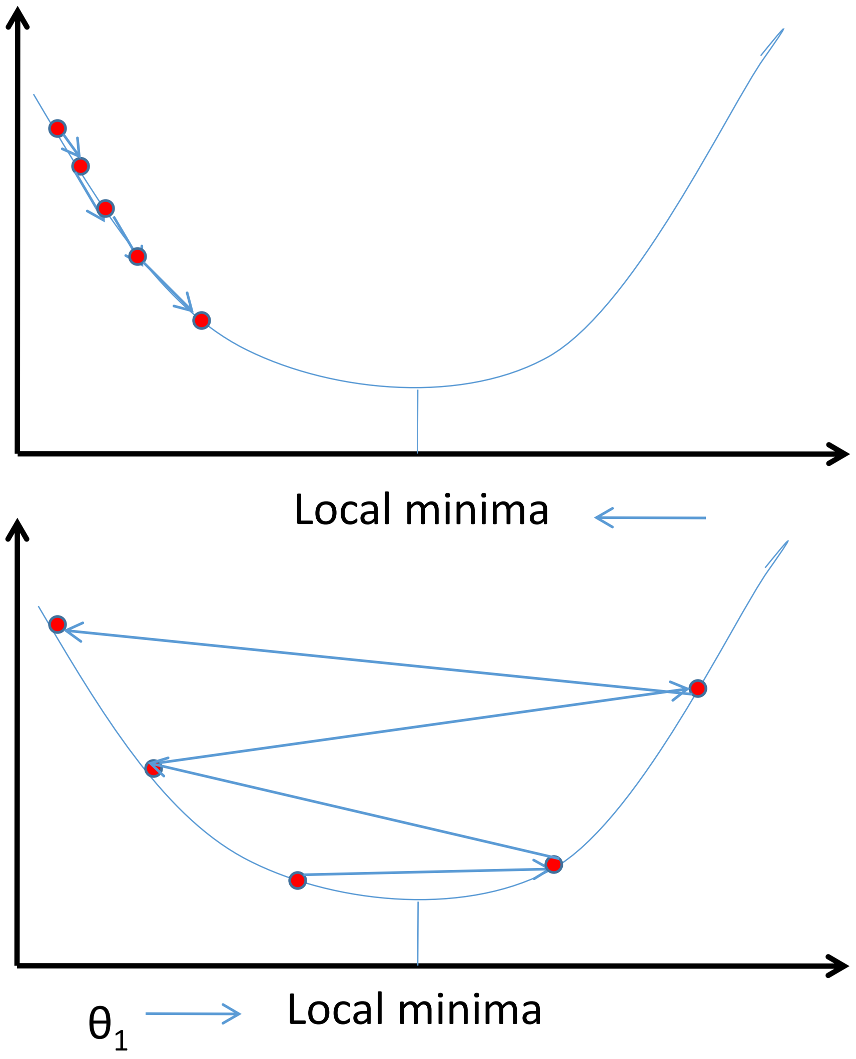
(slope is negative)

Gradient Descent Algorithm

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

If α is too small, gradient descent can be slow.

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



Multivariate Linear Regression

Univariate Hypothesis function:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Multivariate Hypothesis function:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$\mathrm{X} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$\Theta^T x = [\Theta_0 \quad \Theta_1 \quad \dots \quad \Theta_n] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

where $x_0 = 1$

$$\boxed{h_\Theta(\mathrm{x}) = \Theta^T x}$$

Multivariate Gradient Descent

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$ → **ϴ :** n+1 dimensional vector

Cost function:

$$\boxed{J(\theta_0, \theta_1, \ldots, \theta_n)} = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

J(ϴ)

Gradient descent:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \boxed{J(\theta_0, \ldots, \theta_n)}$$

} (simultaneously update for every $j = 0, \ldots, n$)

J(ϴ)

Multivariate Gradient Descent

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

Gradient Descent

Previously (n=1):

Repeat {

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

(simultaneously update θ_0, θ_1)

}

New algorithm $(n \geq 1)$:

$\frac{\partial}{\partial \Theta_j} J(\Theta)$

Repeat {

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

(simultaneously update θ_j for $j = 0, \ldots, n$)

}

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

… to continue

Feature Scaling

Feature Scaling

Idea: Make sure features are on a similar scale.

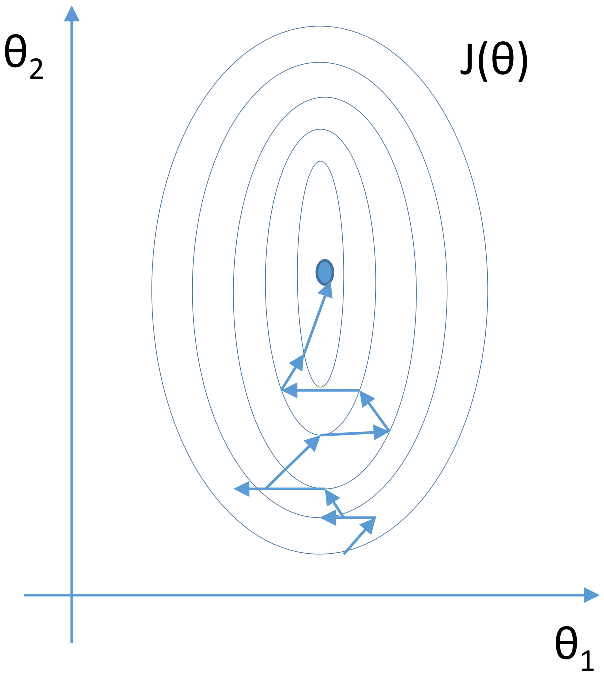
E.g. x_1 = size (0-2000 feet^2)

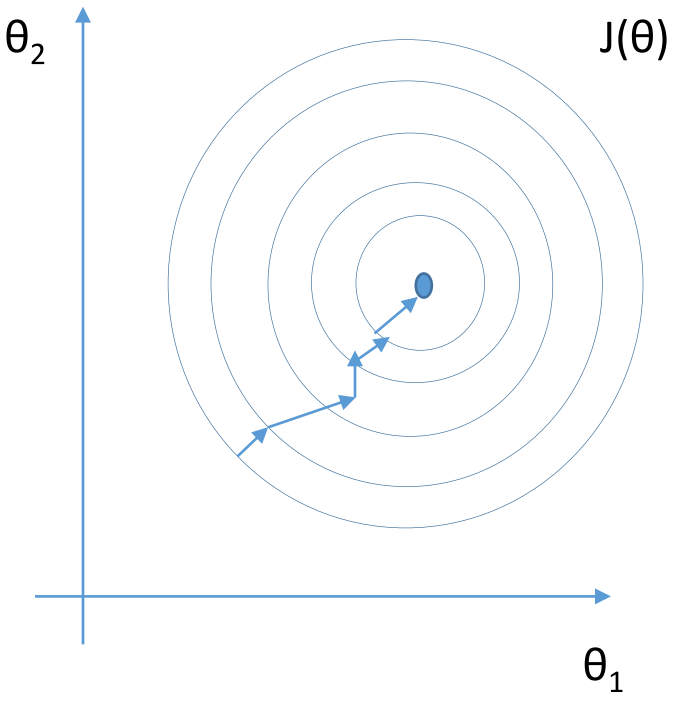
x_2 = number of bedrooms (1-5)

$$x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$

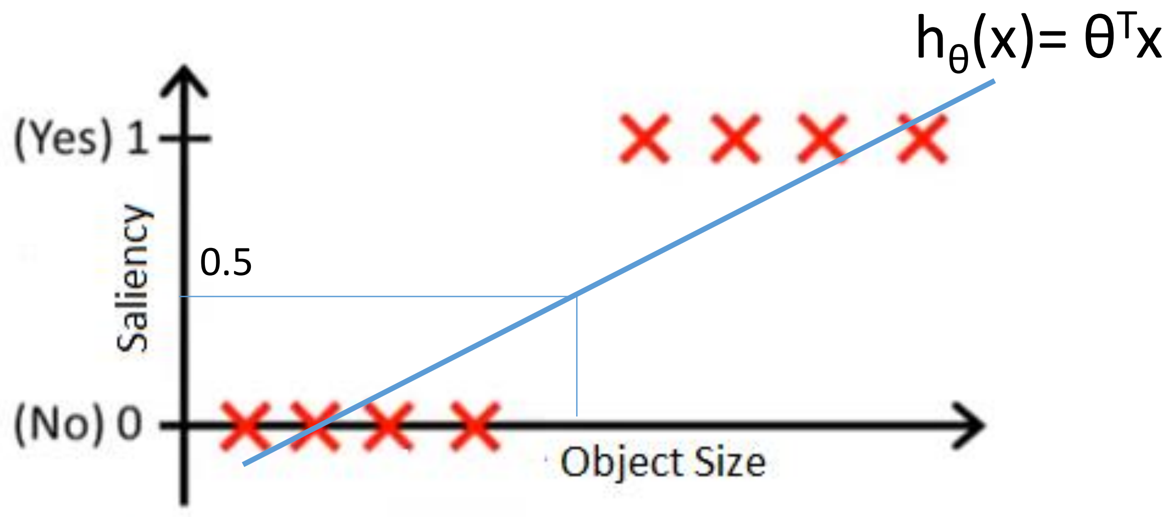
$0 \le x_1 \le 1$ $\quad$ $0 \le x_2 \le 1$





Get every feature into approximately a $-1 \le x_i \le 1$ range.

Logistic Regression: Classification



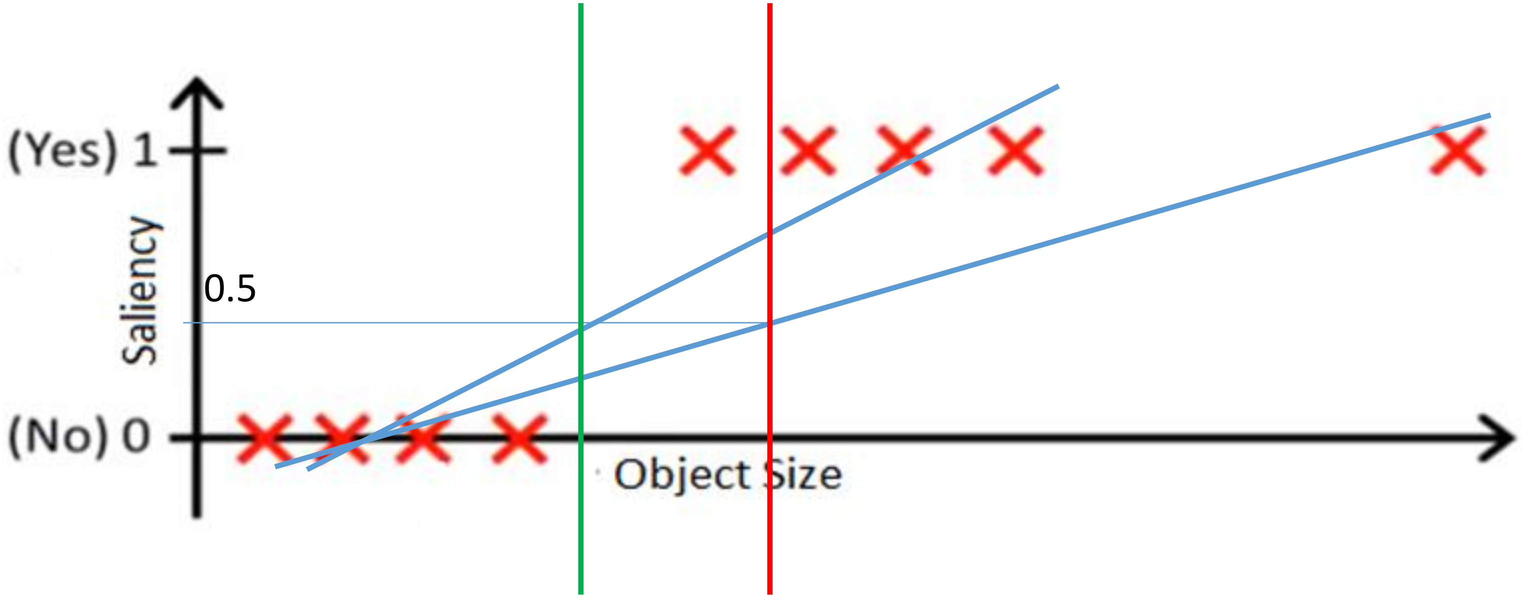
$h_\theta(x) = \theta^T x$

Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Logistic Regression



Linear regression for classification problem is not always good

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Logistic Regression Model

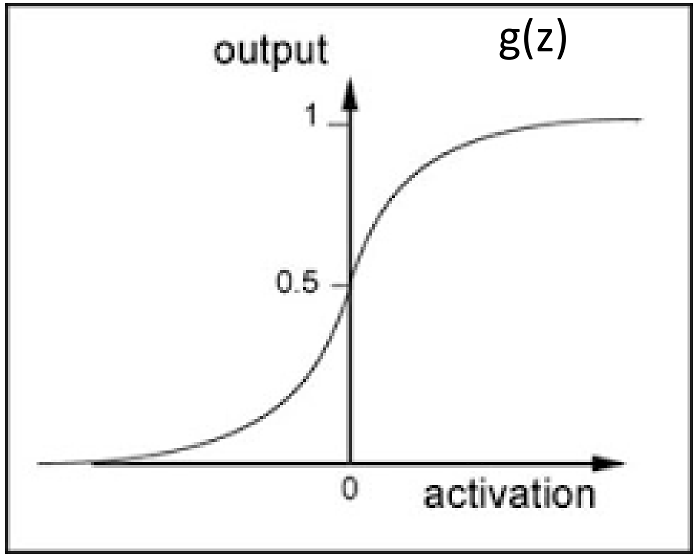
Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Linear Regression: $h_\theta(x) = \theta^T x$

Logistic Regression:

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

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



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

Sigmoid Function or Logistic function

Hypothesis Representation

$h_\Theta(x)$ = estimated probability that y=1 on input x

Example: if $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ Object\ Size \end{bmatrix}$

$h_\Theta(x) = 0.7$

There is 70% chance that the object is salient

$h_\Theta(x) = p(y=1|x, \Theta)$

i.e. "probability that y=1, given x, parameterized by ϴ"

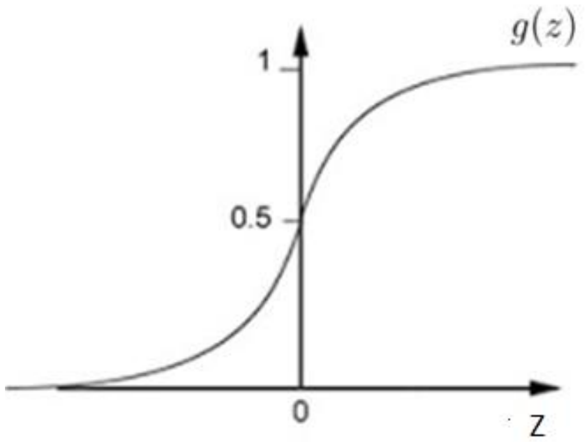
$p(y=0|x; \Theta) + p(y=1|x; \Theta) = 1$

$p(y=0|x; \Theta) = 1 - p(y=1|x; \Theta)$

Decision Boundary

Logistic regression

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



Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

$g(z) \geq 0.5 \quad when \quad z \geq 0$

i.e. $\Theta^T x \geq 0$

$h_\Theta(x) = g(\Theta^T x) \geq 0.5$

whenever $\Theta^T x \geq 0$

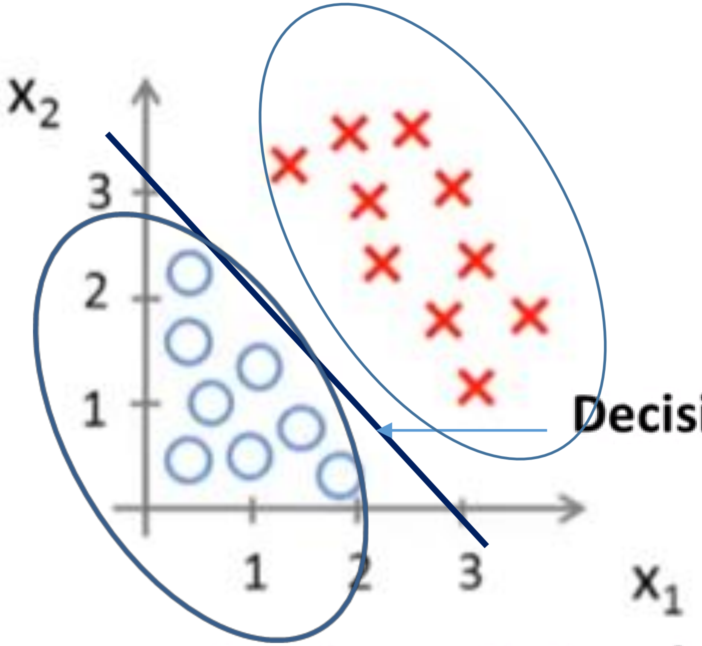
z

predict "$y = 0$" if $h_\theta(x) < 0.5$

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

i.e. $\Theta^T x < 0$

Decision Boundary



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

$$\Theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

$$x_1 + x_2 = 3$$

Decision Boundary

$$h_\Theta(x) = g(\Theta^T x) = 0.5$$

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

i.e. $\Theta^T x \geq 0$

$$x_1 + x_2 \geq 3$$

Predict "y = 0" if

$$x_1 + x_2 < 3$$

Decision boundary is a property of hypothesis function NOT of a data set

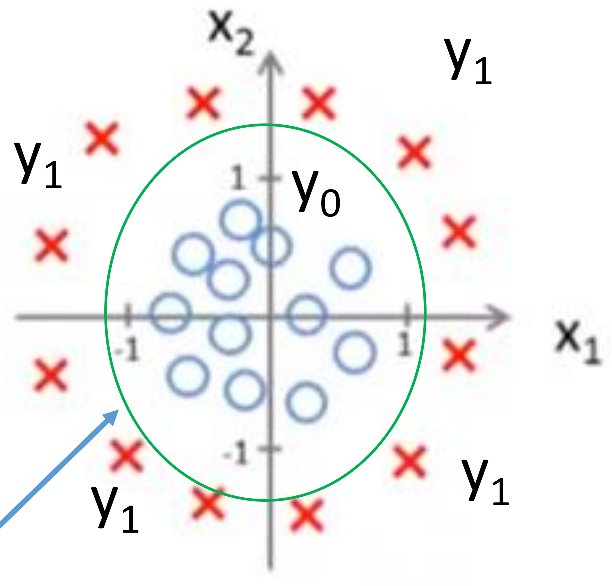
Non-Linear Decision Boundary

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Let $\Theta^T = [-1 \quad 0 \quad 0 \quad 1 \quad 1]$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

$$x_1^2 + x_2^2 \geq 1$$



$$x_1^2 + x_2^2 = 1$$

Decision Boundary

Again, decision boundary is a property of hypothesis function NOT of a data set

Cost Function

- Optimization objective of the cost function

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $\quad x \in \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad x_0 = 1, y \in \{0, 1\}$

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

How to choose parameters θ ?

Cost Function

Cost function

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

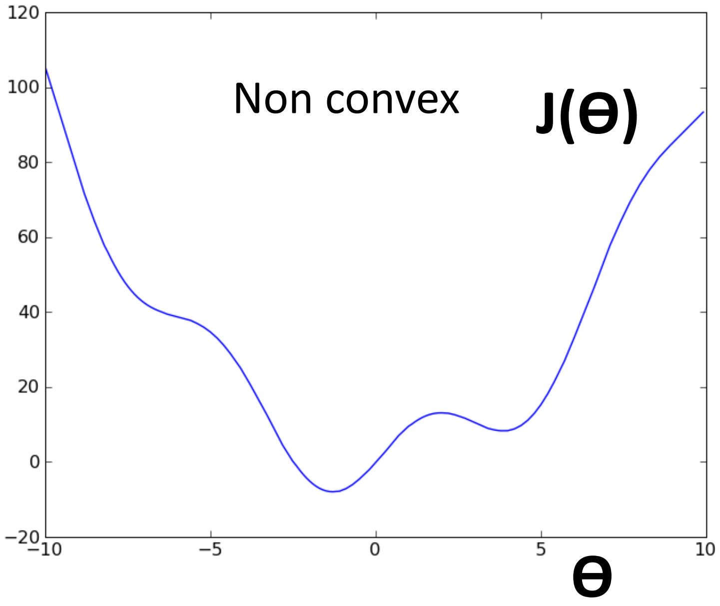
Let, $\text{Cost}\left(h_\Theta(x^{(i)}), y^{(i)}\right) = \frac{1}{2}\left(h_\Theta(x^{(i)}) - y^{(i)}\right)^2$

So, $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$

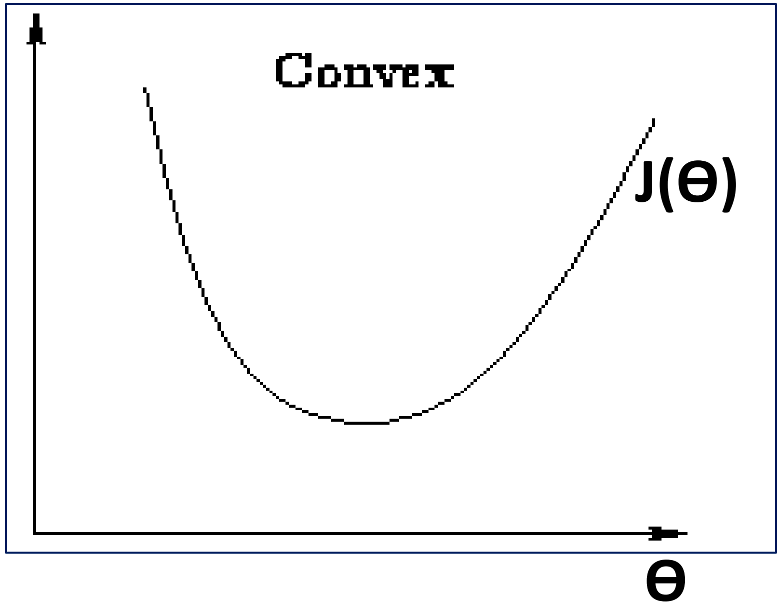
where, for logistic regression

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

Cost Function



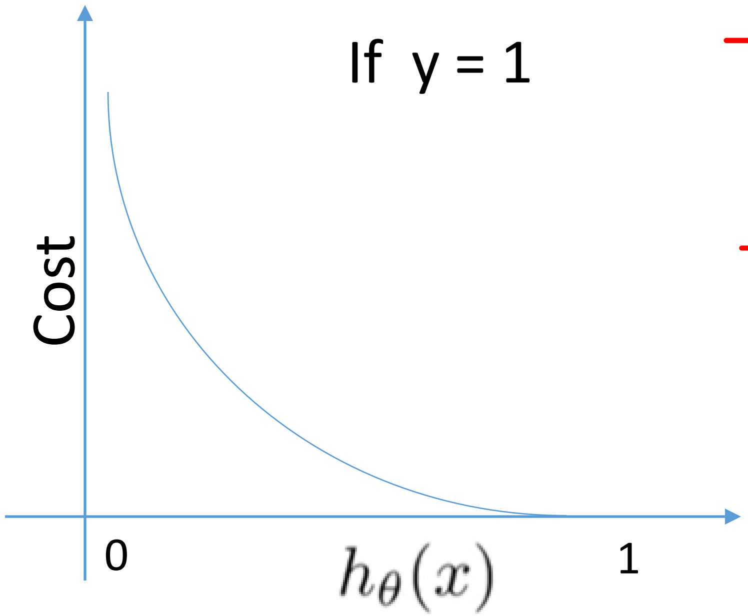
Logistic Regression



Linear Regression

Cost Function: Logistic Regression

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

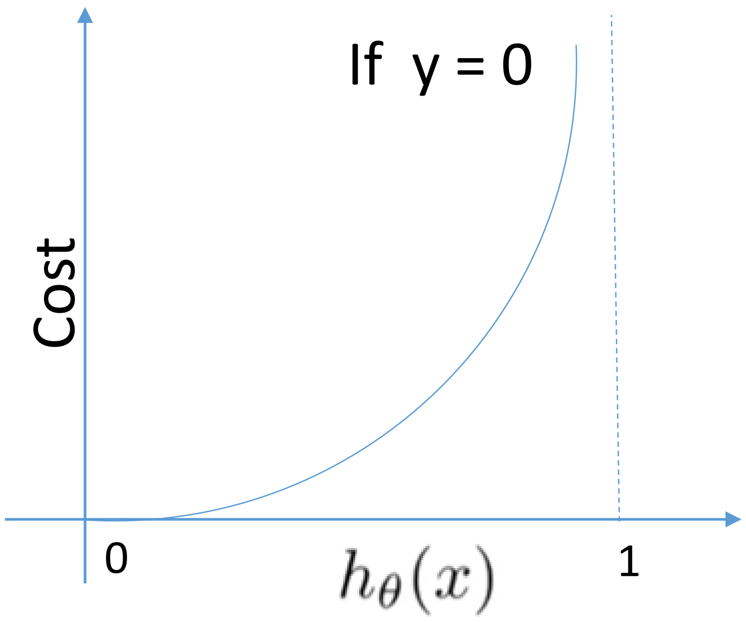


$\text{Cost} = 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

Cost Function: Logistic Regression

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



Cost =0 if y=0, $h_\Theta(x) = 0$
But as $h_\Theta(x) \rightarrow 1$
Cost $\rightarrow \infty$
Captures intuition that if $h_\Theta(x) = 1$,
(predict $P(y=0|x; \Theta) = 1$), but $y = 0$,
We will penalize learning algorithm
by a very large cost.

* It can be shown that the overall cost function is convex function and local optimum free. But details of such convexity analysis is beyond of the scope of this course.

Cost Function: Logistic Regression

Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Note: $y = 0$ or 1 always

$$\mathrm{Cost}\,(h_\Theta(x), y) = -y \log\left(h_\Theta(x)\right) - (1 - y)\log(1 - h_\Theta(x))$$

$$\text{If } \mathrm{y} = 1 : \ \mathrm{Cost}\,(h_\Theta(x), y) = -\log\left(h_\Theta(x)\right))$$

$$\text{If } \mathrm{y} = 0 : \ \mathrm{Cost}\,(h_\Theta(x), y) = -\log(1 - h_\Theta(x))$$

Cost Function: Logistic Regression

Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\rightarrow J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]$$

* Principle of Maximum Likelihood Estimation

To fit parameters θ:

Obtain $\min_\theta J(\theta)$

and get ϴ

To make a prediction given new x:

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

For p(y=1|x; ϴ)

How to minimize J(ϴ) ?

Cost Function and Gradient Descent

Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

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

$\}$ (simultaneously update all θ_j)

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

Cost Function and Gradient Descent

Gradient Descent

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

Want $\min_\theta J(\theta)$:

Repeat {

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

(simultaneously update all θ_j)

}

For Linear Regression:

$$h_\Theta(\mathrm{x}) = \Theta^T x$$

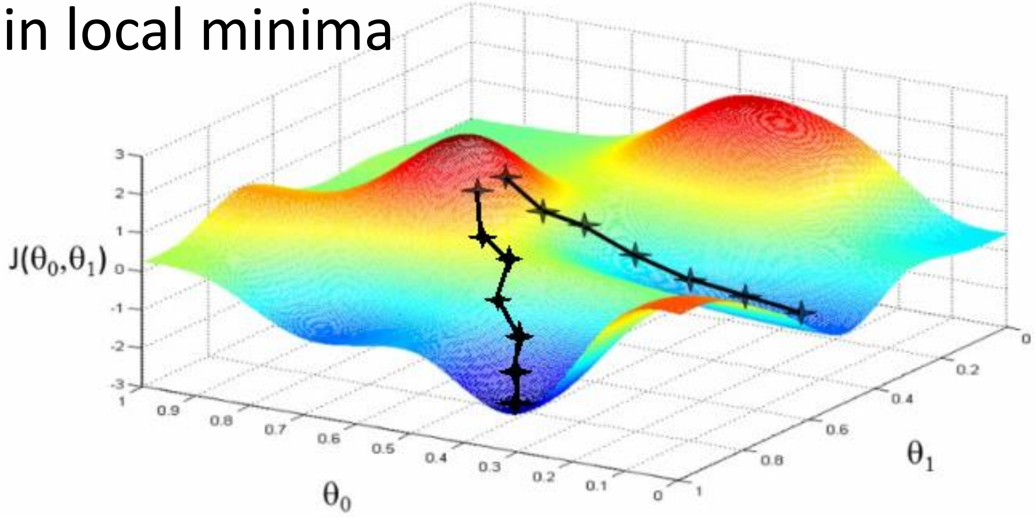
For Logistic Regression:

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

Algorithm looks identical to linear regression!

Gradient descent optimization

- Problems:
  - Choosing step size
    - too small $\rightarrow$ convergence is slow and inefficient
    - too large $\rightarrow$ may not converge
  - Can get stuck on "flat" areas of function
  - Easily trapped in local minima



Stochastic gradient descent

Stochastic (definition):

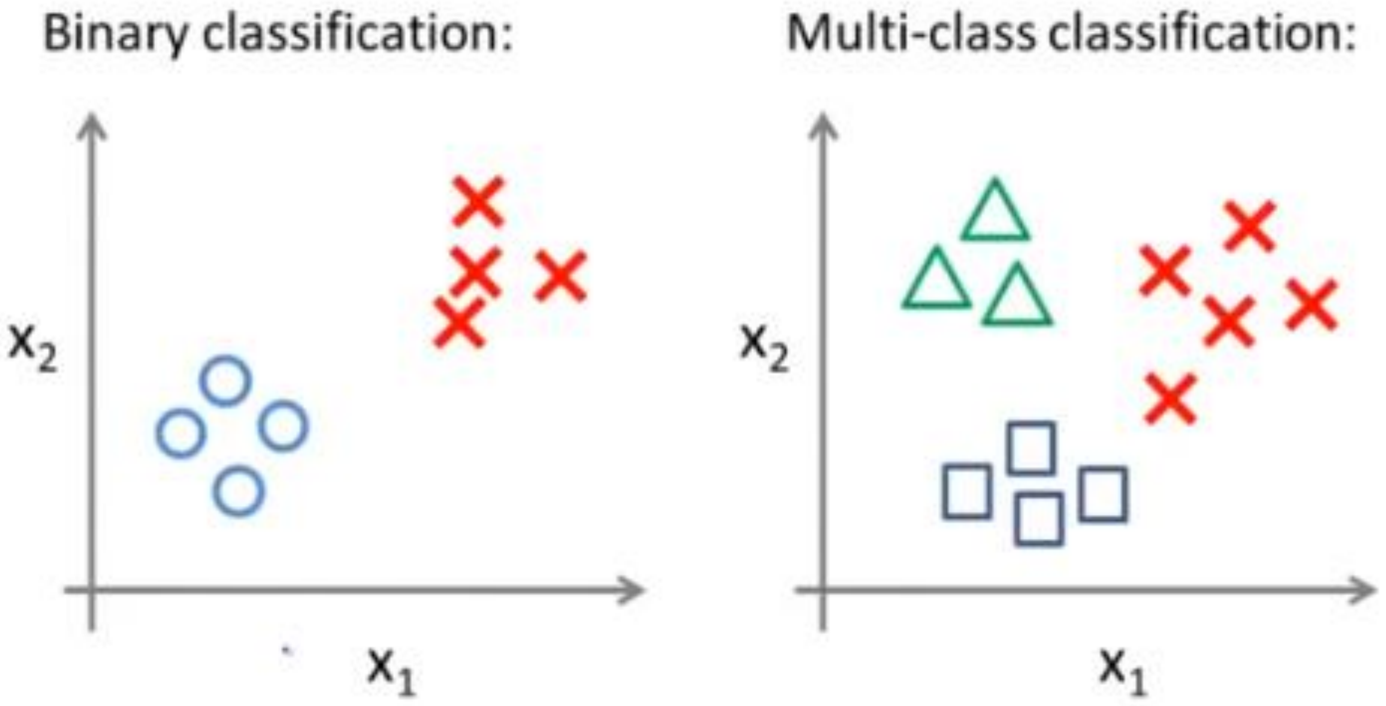
1. involving a random variable
2. involving chance or probability; probabilistic

Stochastic gradient descent

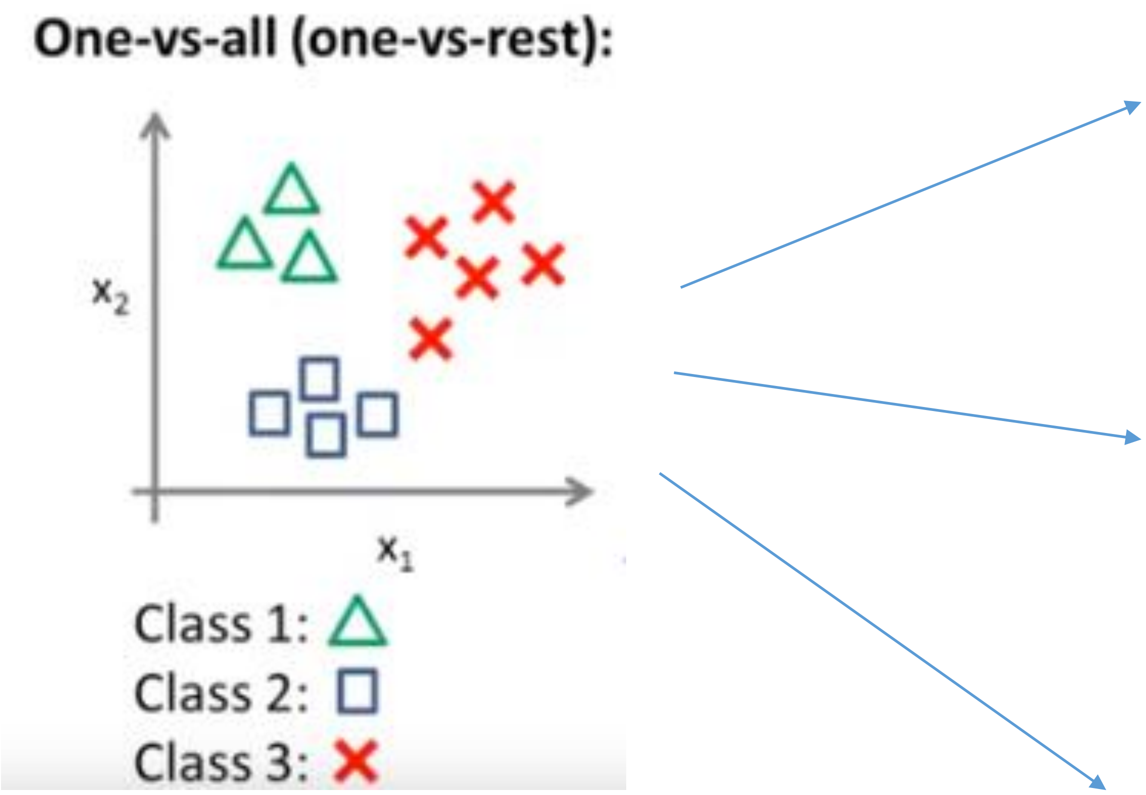
- Application to training a machine learning model:
    1. Choose one sample from training set
    2. Calculate loss function for that single sample
    3. Calculate gradient from loss function
    4. Update model parameters a single step based on gradient and learning rate
    5. Repeat from 1) until stopping criterion is satisfied

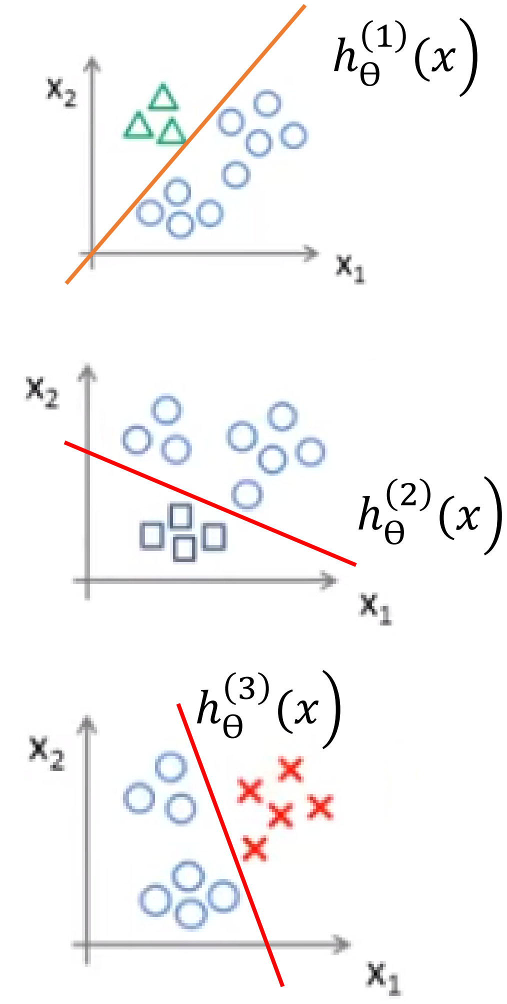
- Typically entire training set is processed multiple times before stopping.
- Order in which samples are processed can be fixed or random.

Multi Class Classification



One vs. All (One vs. Rest)





$$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$$

One vs. All (One vs. Rest)

- Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

- On a new input x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)}(x)$$

Overfitting

- A hypothesis function h is said to overfit the training data if there is another hypothesis h’ such that h’ has more error than h on training data but h’ has less error than h on testing data.

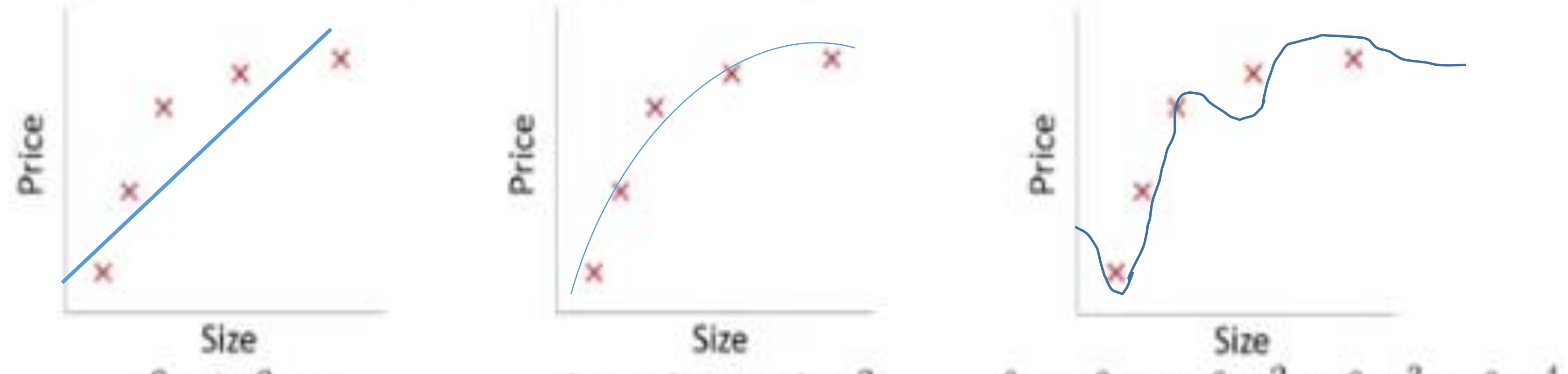
- Learning a classifier that classifies a training data perfectly may not lead to the classifier with best generalization performance
  - There may be noise in training data
  - Training data set is too small

- Simplistically, overfitting occurs when model is too complex whether underfitting occurs when model is too simple.

- Note: Training error is not a good predictor for the testing error.

The problem of overfitting

Example: Linear regression (housing prices)



$\theta_0 + \theta_1 x$ | $\theta_0 + \theta_1 x + \theta_2 x^2$ | $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Under fit or High bias

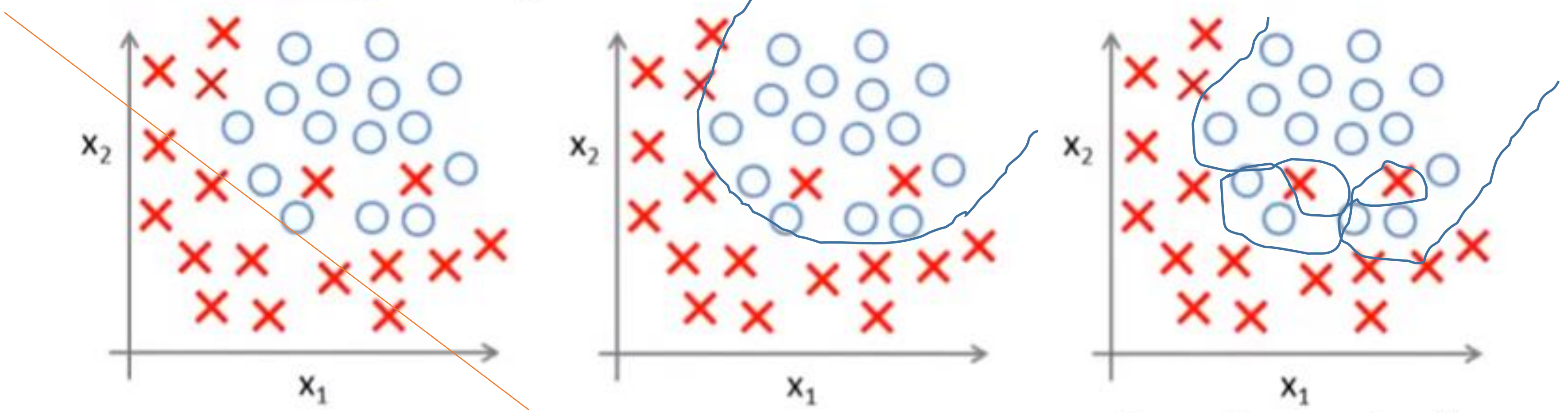
low variance

Over fit or High variance

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples).

The problem of overfitting

Example: Logistic regression



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

(g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$

Under fit or High bias

Over fit or High variance

The problem of overfitting

- **Let's consider D, the entire distribution of data, and T, the training set.**
- **Hypothesis h $\in$ H overfits D if**

  $\exists$ h'$\neq$ h $\in$ H such that

  (1) $error_T(h) < error_T(h')$ [i.e. doing well on training set] but

  (2) $error_D(h) > error_D(h')$

- **What do we care about most (1) or (2)?**
- ***Estimate error on full distribution by using test data set.***

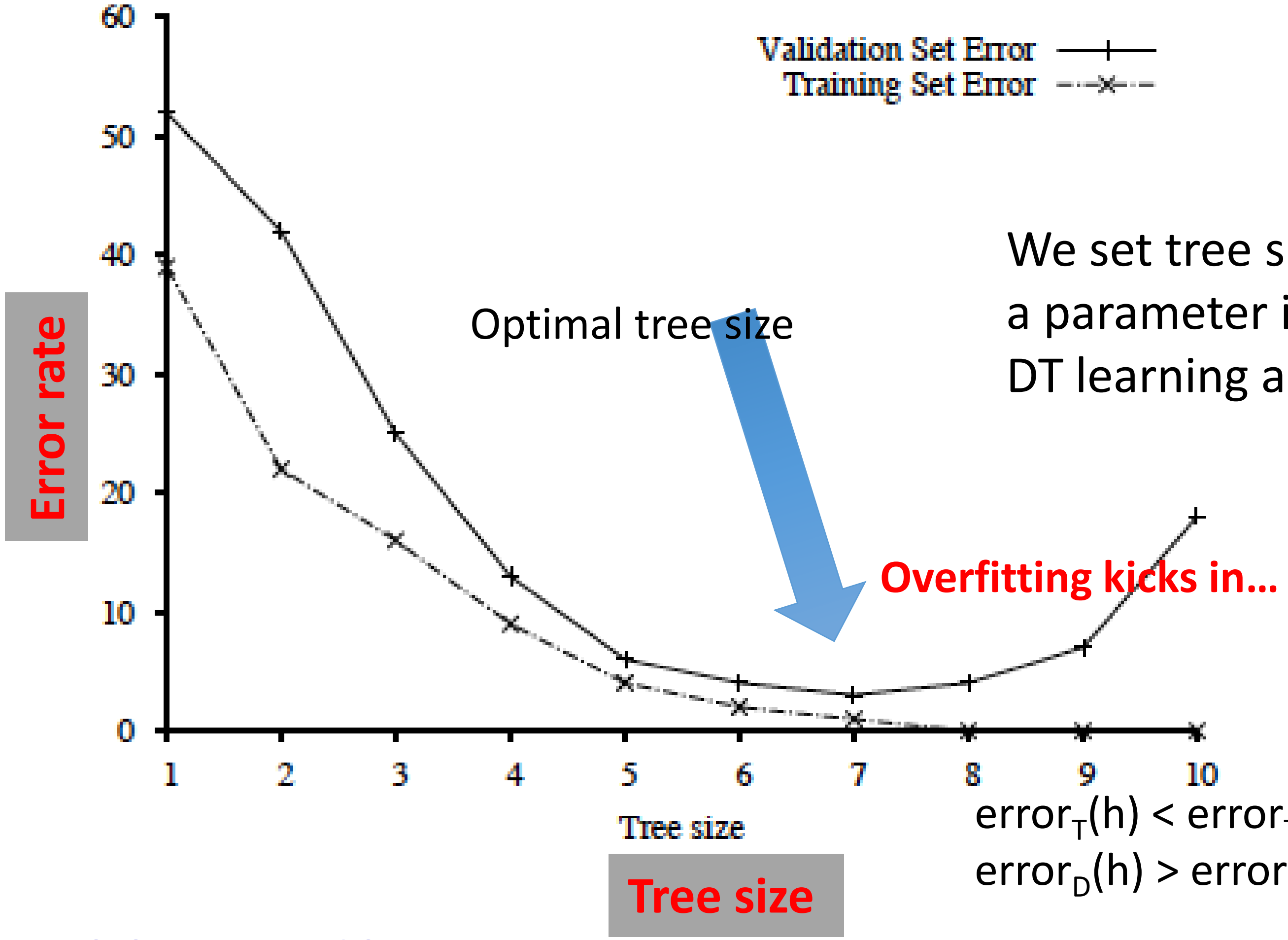
  Error on test data: Generalization error (want it low!!)

- ***Generalization to unseen examples/data is what we care about.***

The problem of overfitting

- Data overfitting is the arguably the most common pitfall in machine learning.
- **Why?**
- Temptation to use as much data as possible to train on. ("Ignore test till end." Test set too small.) Data "peeking" not noticed.
- *Temptation to fit very complex hypothesis* (e.g. large decision tree). In general, the larger the tree, the better the fit to the training data.
- It's hard to think of a better fit to the training data as a "worse" result. Often difficult to fit training data well, so it seems that "a good fit to the training data means a good result."

Key figure in machine learning



We set tree size as a parameter in our DT learning alg.

$\text{error}_T(h) < \text{error}_T(h')$ but $\text{error}_D(h) > \text{error}_D(h')$

Note: with larger and larger trees, we just do better and better on the training set!

But note the performance on the validation set degrades!

Note: Similar curves can happen when training too long in complex hypothesis space with lots of parameters to set.

Solutions for Overfitting

- K- fold Cross Validation
- Regularization
- Early stopping
- Drop-out
- Pre or post pruning for decision tree
- Minimum description length (MDL) principle

K- fold Cross Validation

Training Set		Testing Set
Training Set	Validation Set	Testing Set

S1	S2	S3	…	Sk

Training Set = S

Average test score = 1/k (∑Si)

Round	Training Set	Testing Set
1	S1	S – S1
2	S2	S – S2
i	Si	S-Si

- **Trade-off**:
- Complex hypothesis fit the data well → may tend to overfitting
- Simple hypothesis may generalize better → may tend to underfitting
- As the training data samples increase, generalization error decreases.

Regularization

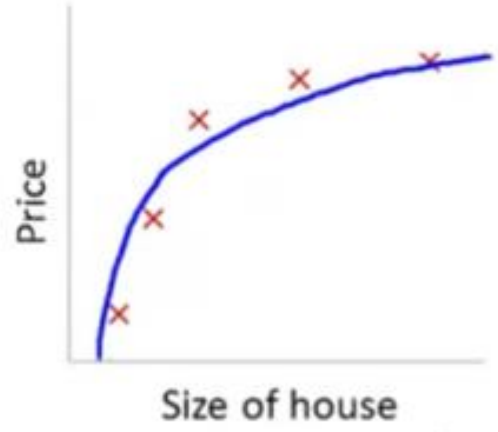
Options:

1. Reduce number of features.
   — Manually select which features to keep.
   — Model selection algorithm

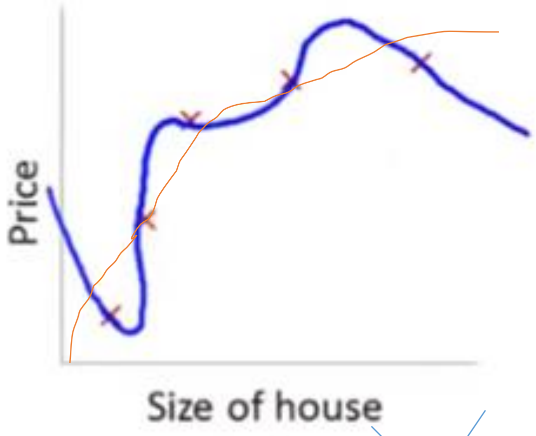
2. Regularization.
   — Keep all the features, but reduce magnitude/values of parameters θ_j.
   — Works well when we have a lot of features, each of which contributes a bit to predicting y.

Regularization

Intuition



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



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\,\theta_3^2 + 1000\,\theta_4^2$$

$$\theta_3 \cong 0 \qquad \theta_4 \cong 0$$

Regularization

Regularization.

Small values for parameters $\theta_0, \theta_1, \ldots, \theta_n$

- "Simpler" hypothesis
- Less prone to overfitting

Housing:

- Features: $x_1, x_2, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{i=1}^{n} \theta_j^2$$

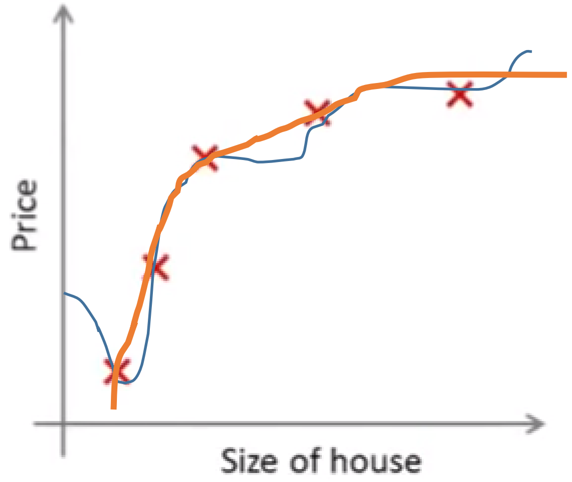
Regularization

Regularization.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\lambda}\sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

Regularization parameter



- Fitting the data points well
- Keeping the no. of parameters (Ɵs) small

Regularization

In regularized linear regression, we choose θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

$$\Theta_1 \cong 0;\ \Theta_2 \cong 0; \Theta_3 \cong 0;\ \Theta_4 \cong 0;$$

$$h_\Theta(x) = 0$$

$$h_\Theta(x) = \ \theta_0 + \cancel{\theta_1 x} + \cancel{\theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

Regularized Linear Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

Gradient descent

Repeat $\{$

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

$$(j = \cancel{0}, 1, 2, 3, \ldots, n)$$

$\}$

Regularized Linear Regression

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

$$(j = 0, 1, 2, 3, \ldots, n)$$

Shrinkage

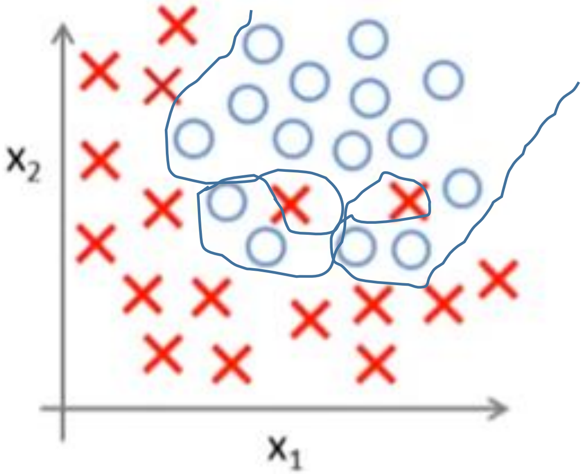
Parameter updation

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

$$(1 - \alpha \frac{\lambda}{m}) < 1$$

Shrinkage $\left(1 - \frac{\alpha\lambda}{m}\right)$

Regularized Logistic Regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

Cost function:

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

$$+ \frac{\lambda}{2m} \sum_{i=1}^{n} \theta_j^2$$

Regularized Logistic Regression

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

$$(j = 0, 1, 2, 3, \ldots, n)$$

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

For Logistic Regression:

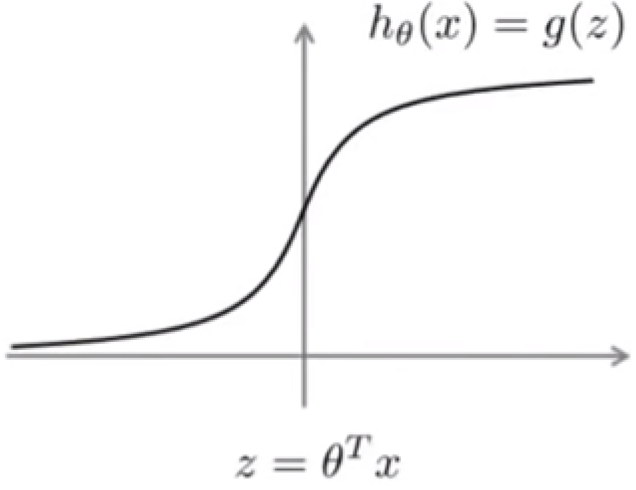
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Support Vector Machine

Optimization objective

Alternative view of logistic regression

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



If $y = 1$, we want $h_\theta(x) \approx 1$, $\theta^T x \gg 0$

If $y = 0$, we want $h_\theta(x) \approx 0$, $\theta^T x \ll 0$

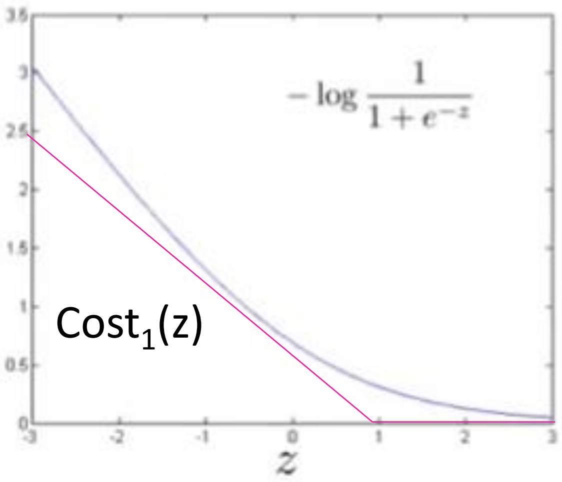
SVM

Alternative view of logistic regression

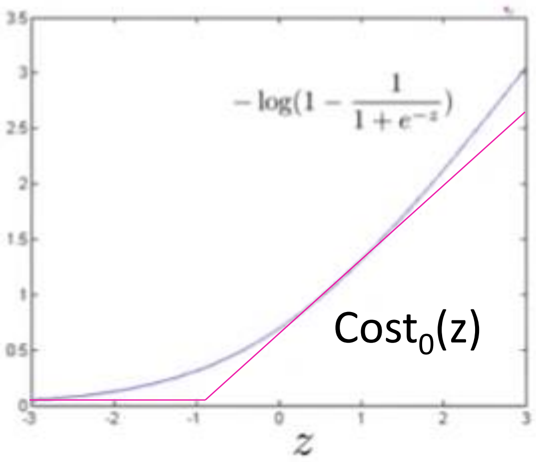
Cost of example: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

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

If $y = 1$ (want $\theta^T x \gg 0$):

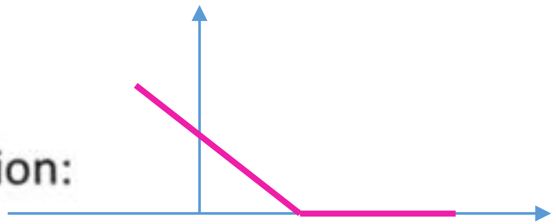


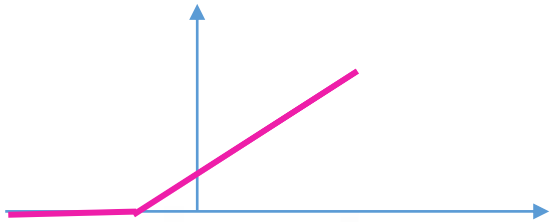
If $y = 0$ (want $\theta^T x \ll 0$):



SVM

Logistic regression:





$$\min_{\theta} \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(\underbrace{-\log h_\theta(x^{(i)})}_{Cost_1(\Theta^T x^{(i)})}\right) + (1-y^{(i)})\left(\underbrace{(-\log(1-h_\theta(x^{(i)})))}_{Cost_0(\Theta^T x^{(i)})}\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Support vector machine:

$$\min_{\Theta} \frac{1}{m}\sum_{1}^{m} y^{(i)} Cost_1(\Theta^T x^{(i)}) + (1-y^{(i)}) Cost_0(\Theta^T x^{(i)}) \ + \frac{\lambda}{2m}\sum_{i=1}^{n}\Theta_j^2$$

A + λ B = C A + B where C = 1/ λ

$$\min_{\theta} C \sum_{i=1}^{m}\left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

SVM

SVM hypothesis

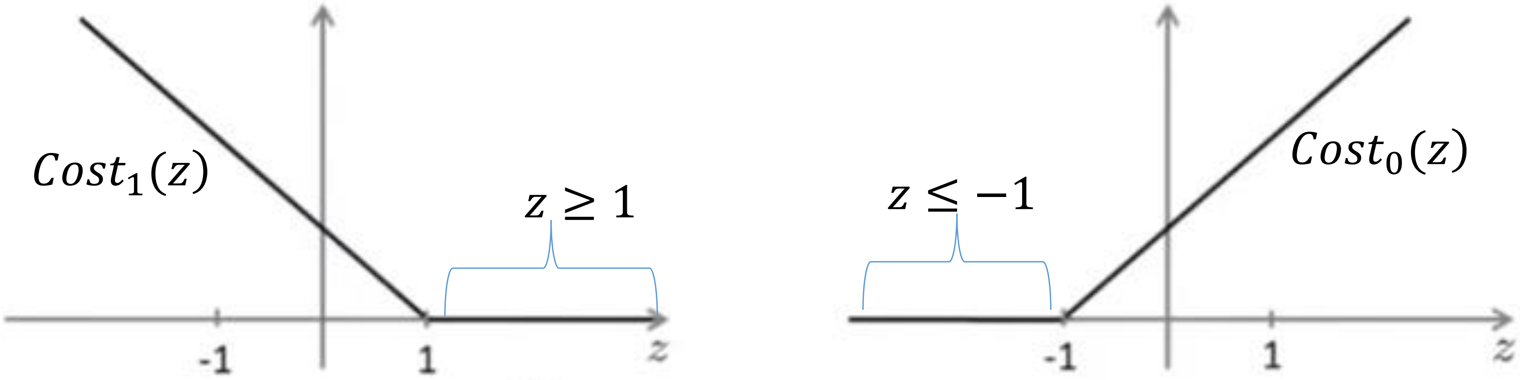
$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

Hypothesis:

$$h_\Theta(x) = \begin{cases} 1 & if \;\; \Theta^T x \geq 0 \\ 0 & otherwise \end{cases}$$

SVM: As Large Margin Classifier

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$



If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

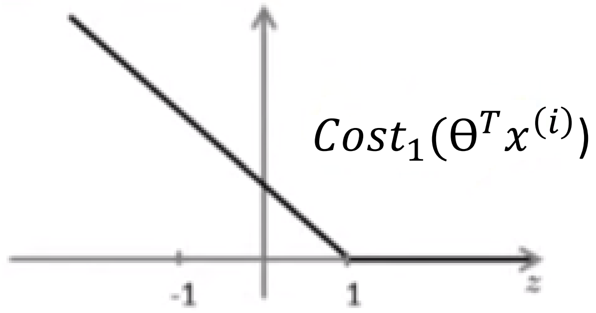
If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0)

SVM Decision Boundary

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

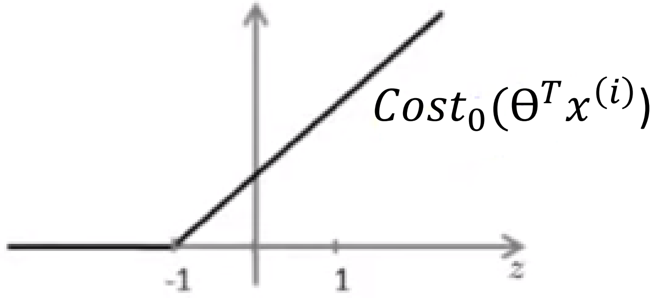
Whenever $y^{(i)} = 1$:

$$\Theta^T x^{(i)} \geq 1$$



Whenever $y^{(i)} = 0$:

$$\Theta^T x^{(i)} \leq -1$$



SVM Decision Boundary

$$\min_{\theta} C \sum_{i=1}^{m} \underbrace{\left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right]}_{= 0 \text{ as C is very big number}} + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

Whenever $y^{(i)} = 1$:

$$\theta^T x^{(i)} \geq 1$$

$$\min_{\theta} C \times 0 + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2 \quad \text{i.e.} \quad \min_{\theta} \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

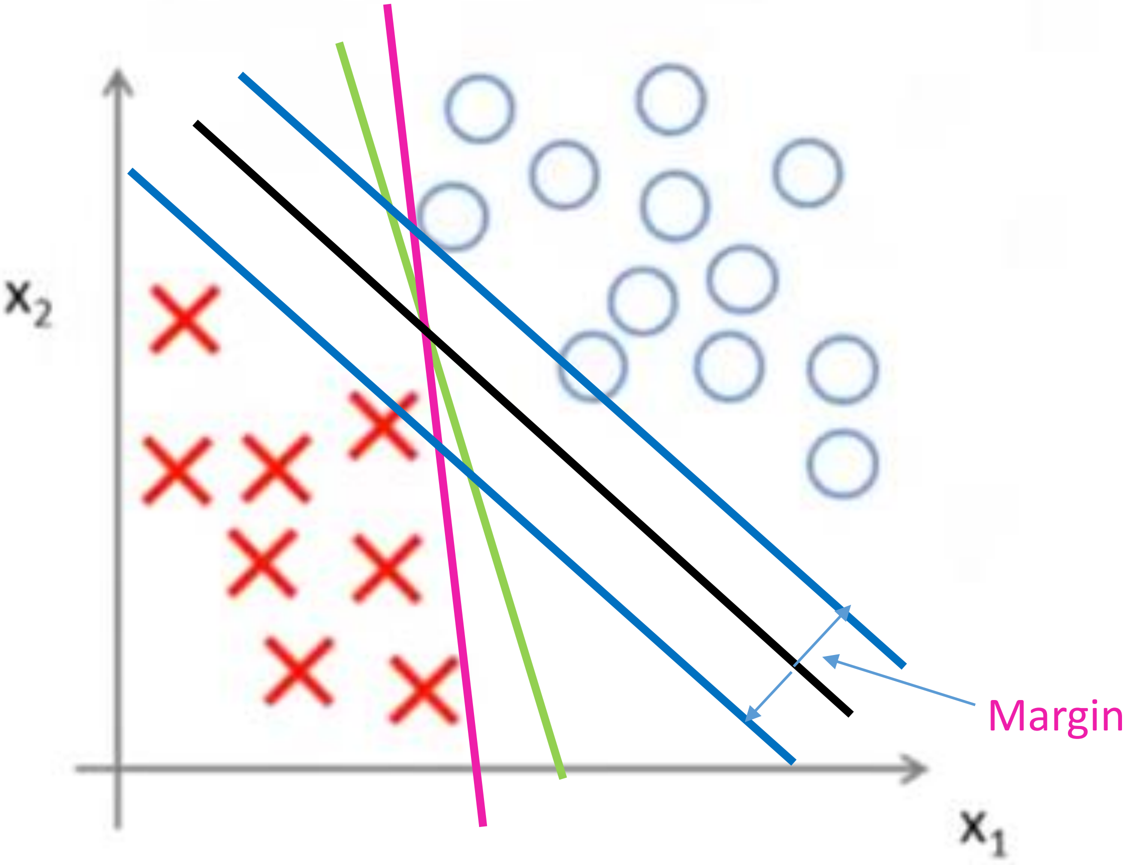
$$s.t. \quad \theta^T x^{(i)} \begin{cases} \geq 1 & if \quad y^{(i)} = 1 \\ \leq -1 & if \quad y^{(i)} = 0 \end{cases}$$

Whenever $y^{(i)} = 0$:

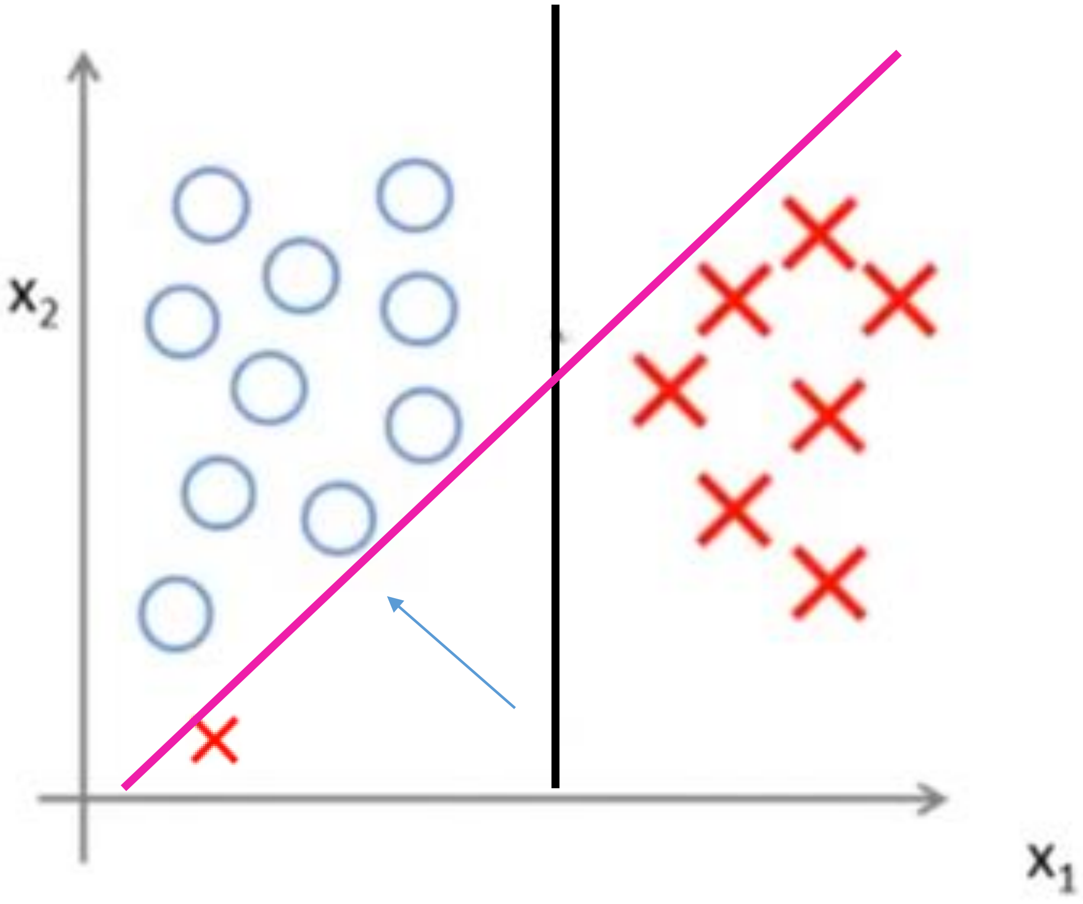
$$\theta^T x^{(i)} \leq -1$$

SVM Decision Boundary

Linearly separable case



Large Margin Classifier
In case of Outliers



C is very large

Sensitive to outliers

Thanks

… to continue