Pattern Analysis and Recognition

Lecture 2: polynomial regression, logistic regression, multi-class classification

Last time on Pattern Analysis and Recognition

RECAP

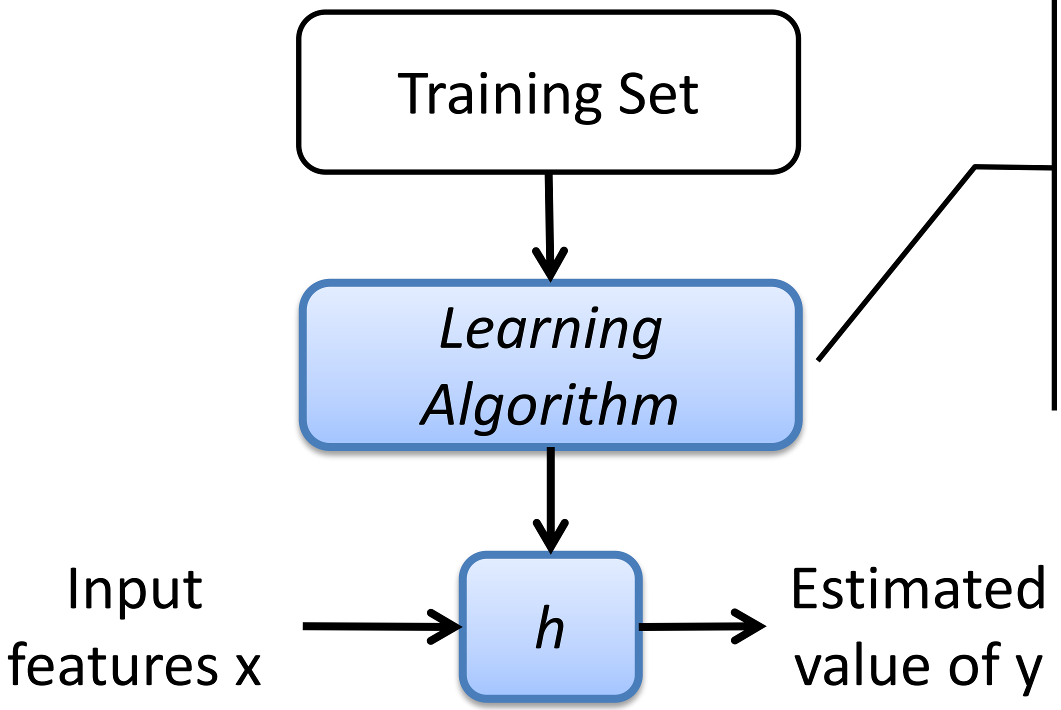
House price prediction

<u>Supervised Learning</u>

"right answers" given

<u>Regression:</u> Predict continuous valued output (price)

Gradient Descent for Linear Regression



Gradient descent algorithm:

Repeat until convergence

{

$$\vartheta_j := \vartheta_j - \alpha \frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1)$$

(for all j simultaneously)

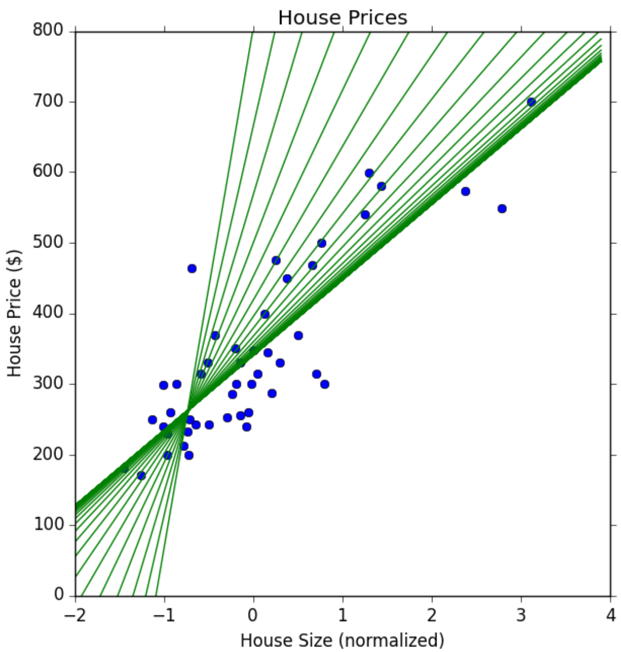
}

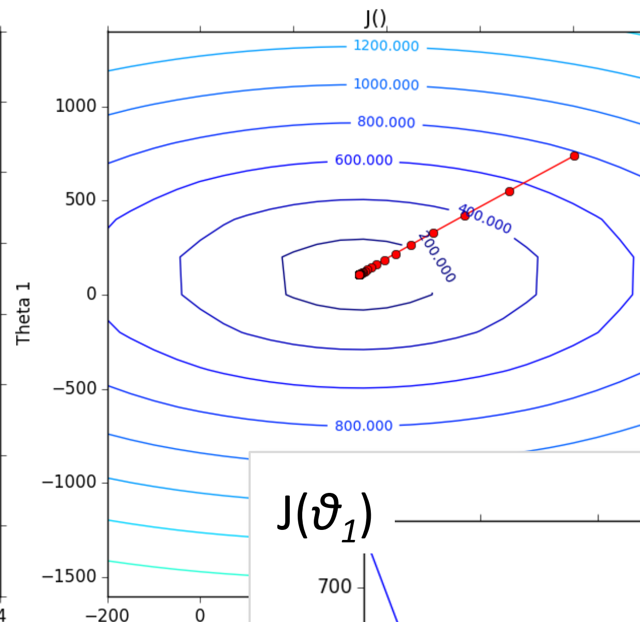
Linear regression model:

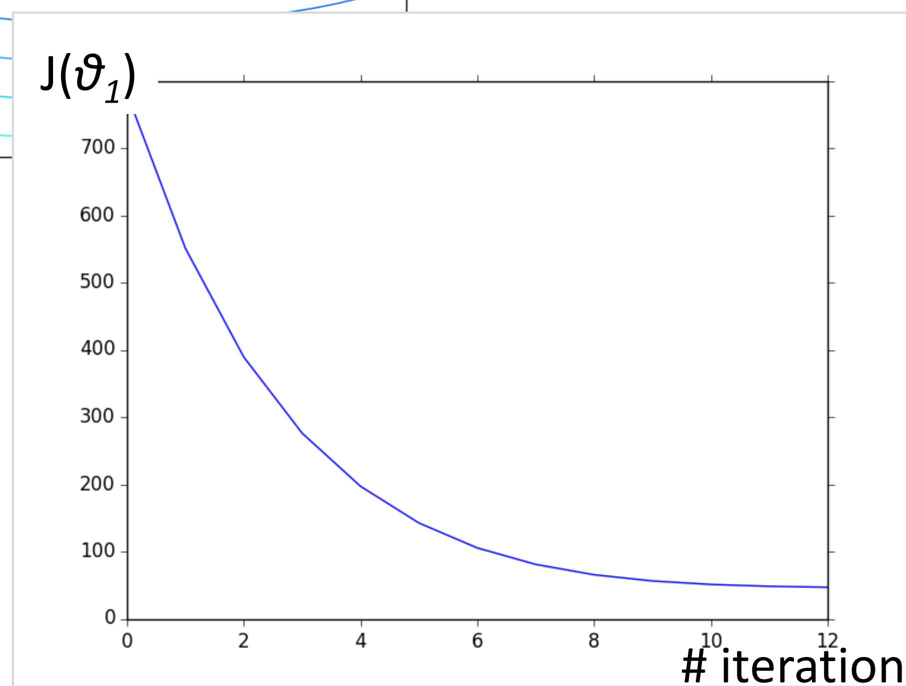
$$h_\vartheta(x) = \vartheta_0 + \vartheta_1 x$$

$$J(\vartheta_0, \vartheta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\vartheta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

Simple Linear Regression







MULTIPLE REGRESSION

One Feature Scenario

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

m

m	number of training examples
x	Input variable / features
y	Output variable / target variable
(x, y)	one training example
$(x^{(i)}, y^{(i)})$	i^{th} training example

Multiple Feature Scenario

Size (feet²) (x_1)	**Number of bedrooms (x_2)**	**Number of floors (x_3)**	**Age of home (years) (x_4)**	**Price ($1000) ($y$)**
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
…	…	…	…	…

m

m	number of training examples
n	number of features
$x_j^{(i)}$	Value of feature *j* in *i*th training example
$y^{(i)}$	Output variable / target variable

Hypothesis Representation

We represent *h* as a linear function of **multiple** variables:

$$h_\vartheta(x) = \vartheta_0 + \vartheta_1 x_1$$

For convenience of notation we introduce $\boldsymbol{x_0 = 1}$:

$$h_\vartheta(x) = \vartheta_0 x_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \ldots + \vartheta_n x_n$$

Multiple Variables Hypothesis

$$h_{\vartheta}(x) = \vartheta_0 x_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \ldots + \vartheta_n x_n = \Theta^T X$$

where:

$$\Theta = \left.\begin{bmatrix} \vartheta_0 \\ \vartheta_1 \\ \vartheta_2 \\ \ldots \\ \vartheta_n \end{bmatrix}\right\} n+1 \qquad X = \left.\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}\right\} n+1$$

Linear Regression Revisited

n=1	***n≥1***
Hypothesis:	
$h_\vartheta(x) = \vartheta_0 + \vartheta_1 x$	$h_\vartheta(x) = \vartheta_0 x_0 + \vartheta_1 x_1 + \ldots + \vartheta_n x_n$
Parameters:	
ϑ_0, ϑ_1	$\vartheta_0, \vartheta_1, \ldots, \theta_n$
Cost Function:	
$J(\vartheta_0, \vartheta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)^2$	$J(\vartheta_0, \vartheta_1, \ldots \vartheta_n) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)^2$
Goal:	
$\underset{\vartheta_0, \vartheta_1}{\text{minimise}}\left(J\left(\vartheta_0, \vartheta_1\right)\right)$	$\underset{\vartheta_0, \vartheta_1, \ldots \vartheta_n}{\text{minimise}}\left(J\left(\vartheta_0, \vartheta_1, \ldots \vartheta_n\right)\right)$

Linear Regression Revisited

n=1 ***n≥1***

Hypothesis:

$$h_\vartheta(x) = \vartheta_0 + \vartheta_1 x$$

$$h_\vartheta(x) = \Theta^T X$$

Parameters:

$$\vartheta_0, \vartheta_1$$

$$\Theta$$

Cost Function:

$$J(\vartheta_0, \vartheta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\vartheta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

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

Goal:

$$\underset{\vartheta_0, \vartheta_1}{\text{minimise}} \left(J\left(\vartheta_0, \vartheta_1\right)\right)$$

$$\underset{\Theta}{\text{minimise}} \left(J\left(\Theta\right)\right)$$

Gradient Descent Revisited

n=1

Repeat until convergence

{

$$\vartheta_j := \vartheta_j - \alpha \frac{\partial}{\partial \vartheta_j} J\left(\vartheta_0, \vartheta_1\right)$$

(simultaneously for j=0 and j=1)

}

n≥1

Repeat until convergence

{

$$\vartheta_j := \vartheta_j - \alpha \frac{\partial}{\partial \vartheta_j} J\left(\Theta\right)$$

(simultaneously for all j)

}

Gradient Descent Revisited

n=1

Repeat until convergence

{

$$\vartheta_0 := \vartheta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\vartheta_1 := \vartheta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

$$\boxed{x^{(i)} \equiv x_1^{(i)}}$$

(simultaneously for j=0 and j=1)

}

n≥1

Repeat until convergence

{

$$\boxed{x_0^{(i)} = 1}$$

$$\vartheta_0 := \vartheta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)x_0^{(i)}$$

$$\vartheta_1 := \vartheta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)x_1^{(i)}$$

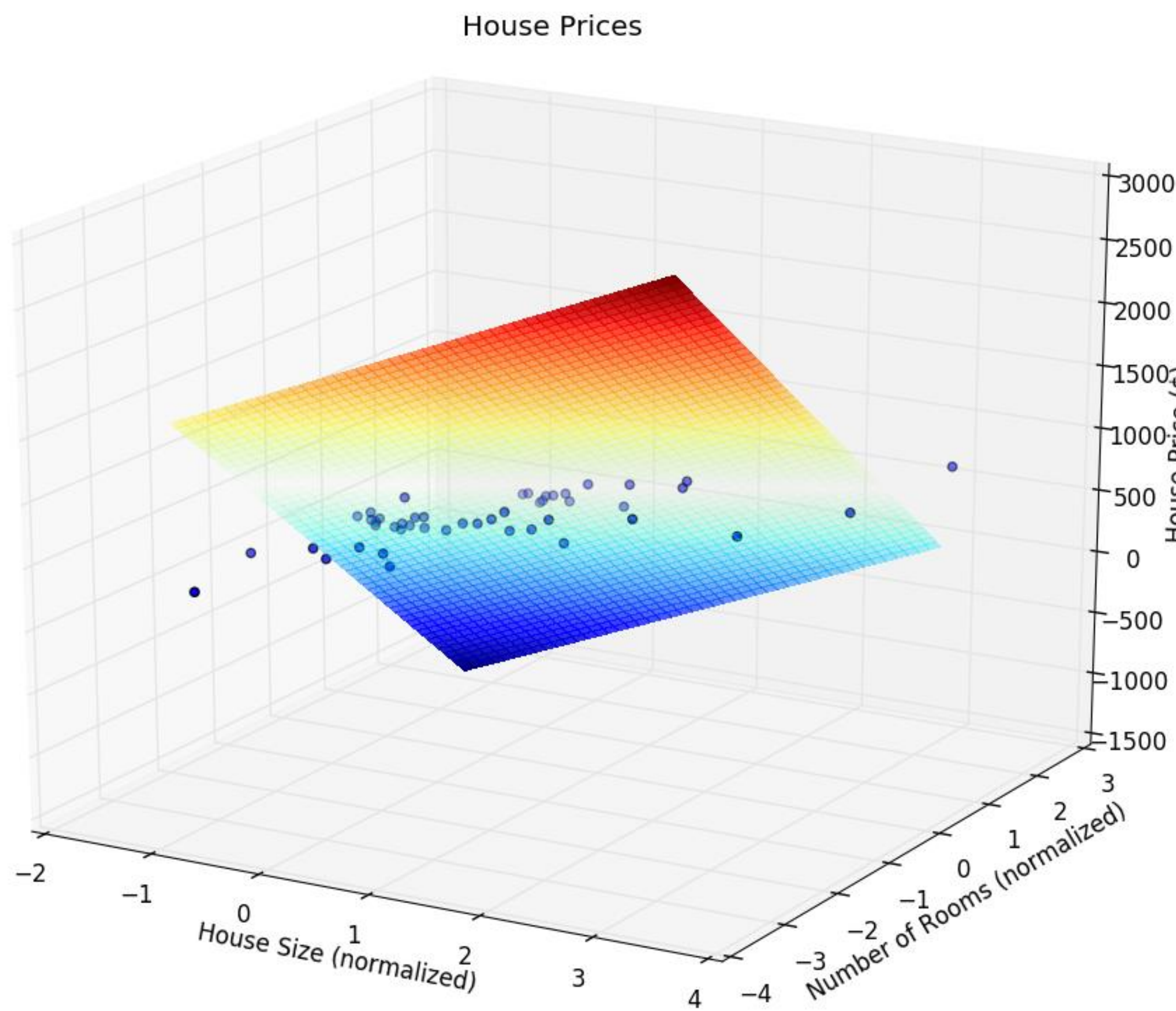
$$\vdots$$

$$\vartheta_n := \vartheta_n - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)x_n^{(i)}$$

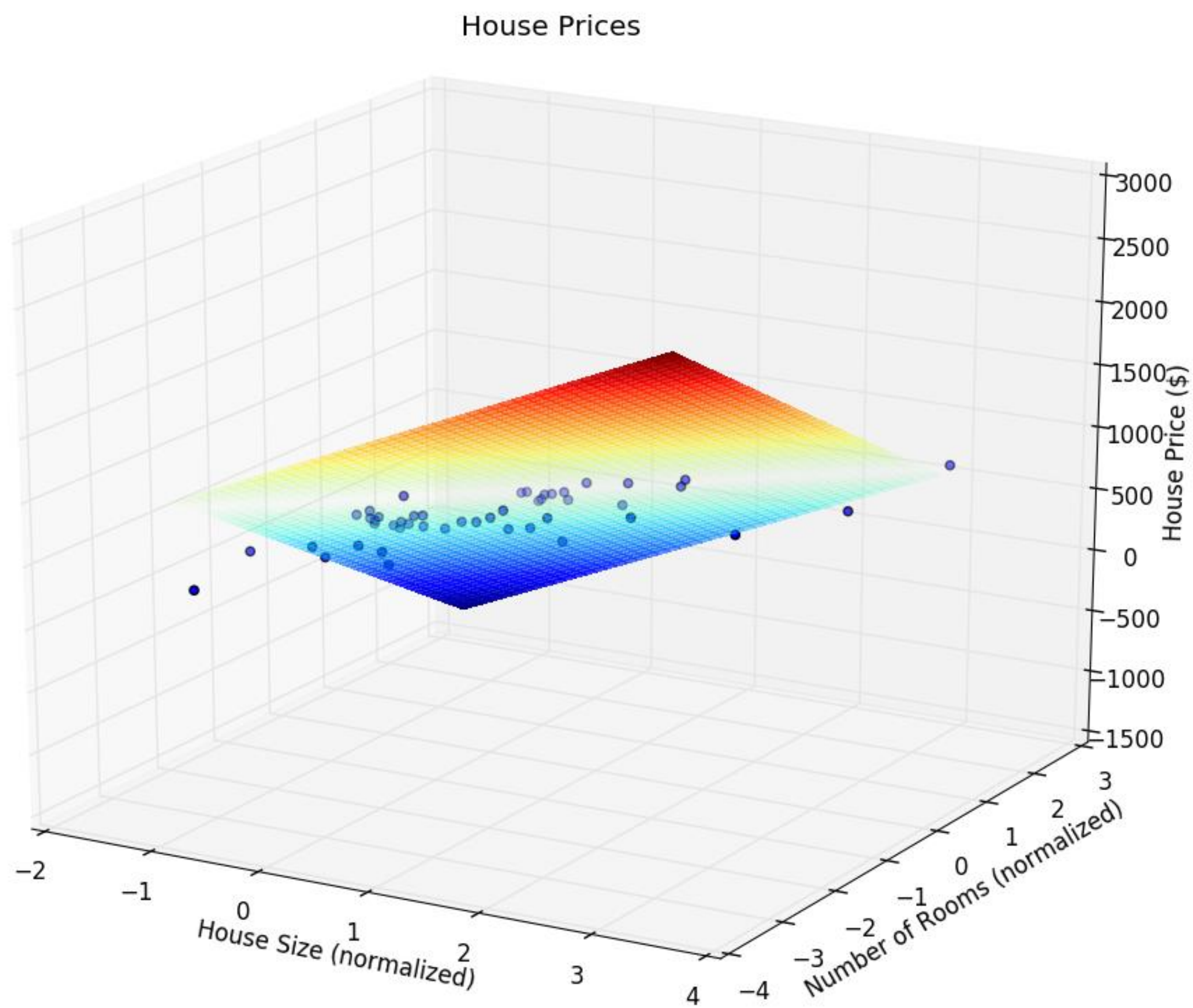
(simultaneously for all j)

}

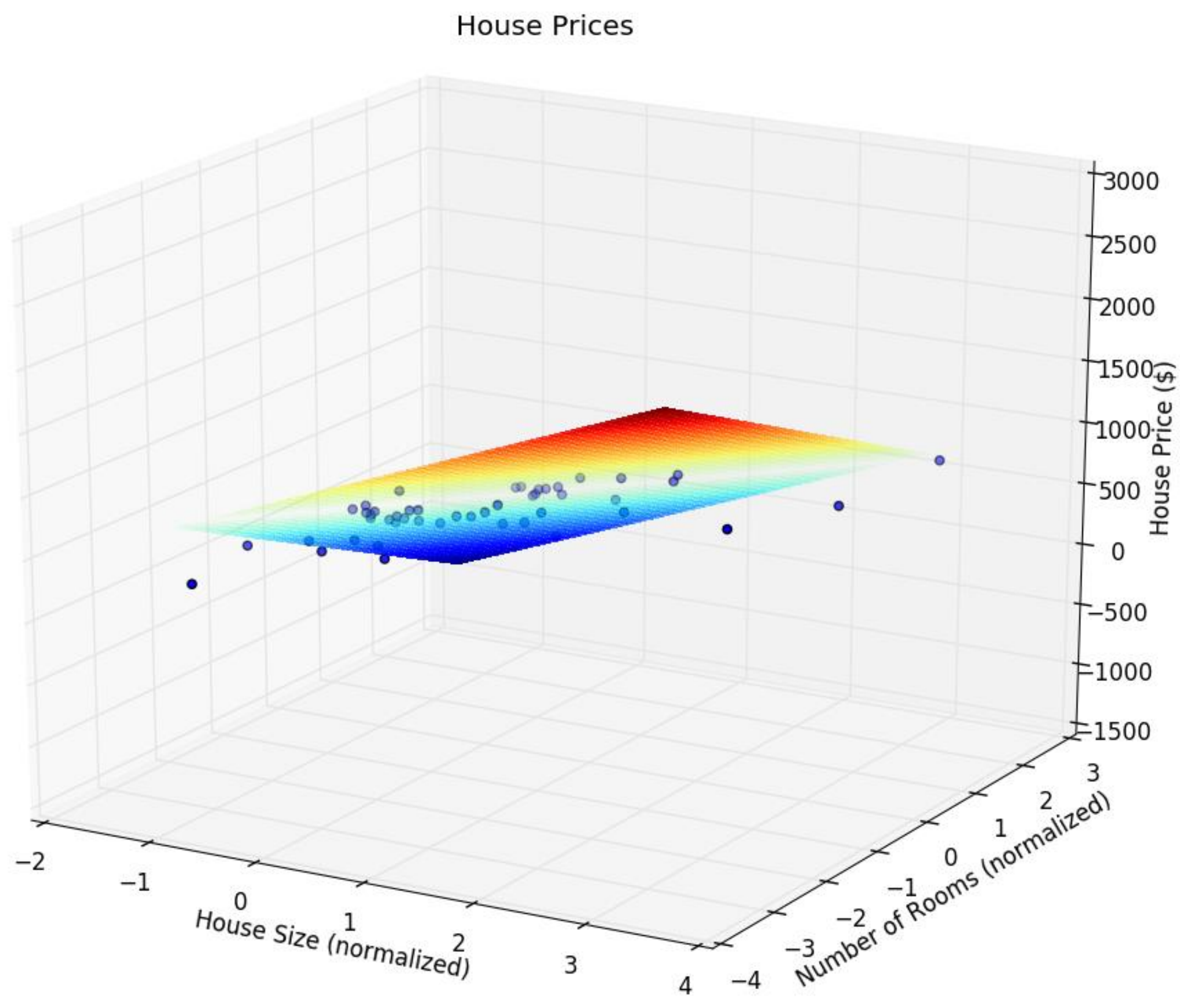
Multiple Regression Example



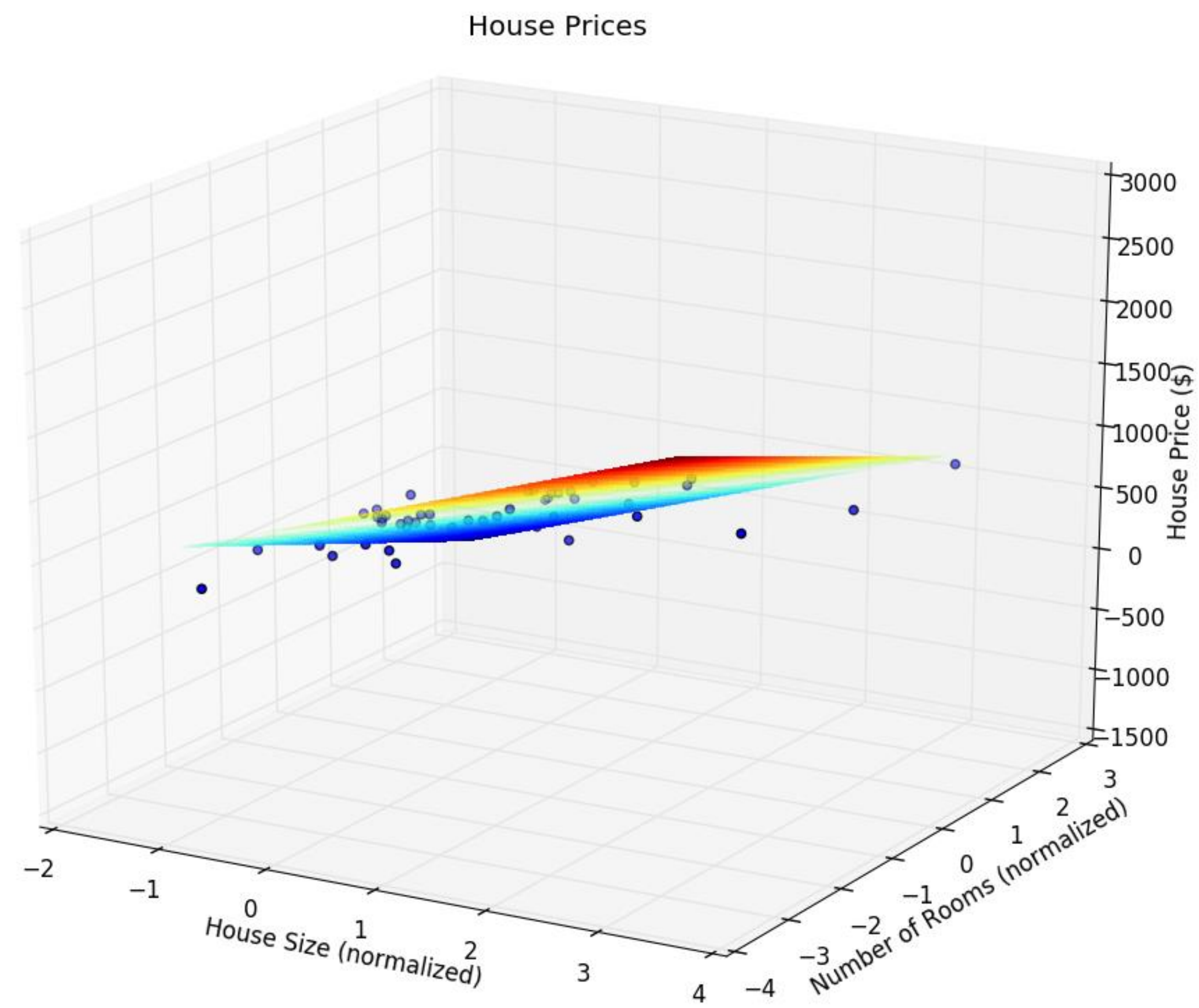
Multiple Regression Example



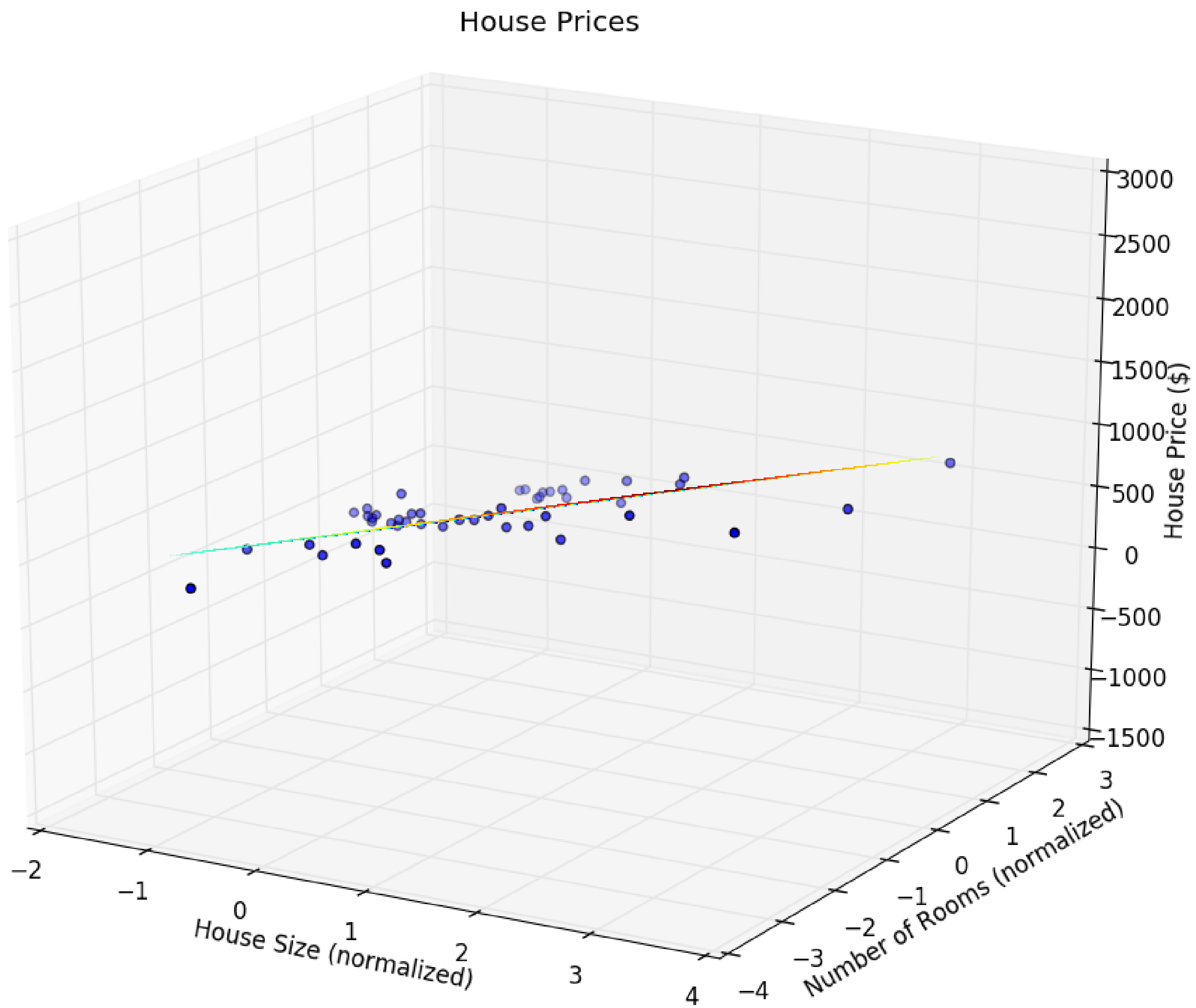
Multiple Regression Example



Multiple Regression Example



Multiple Regression Example



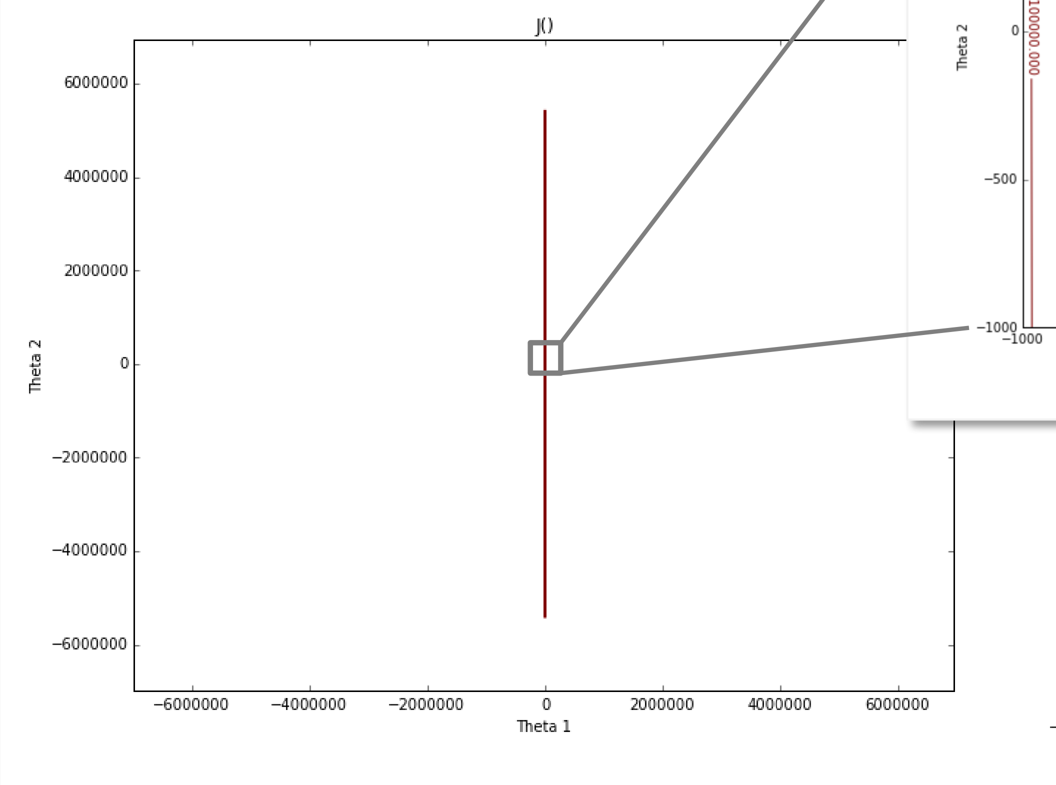
FEATURE NORMALISATION

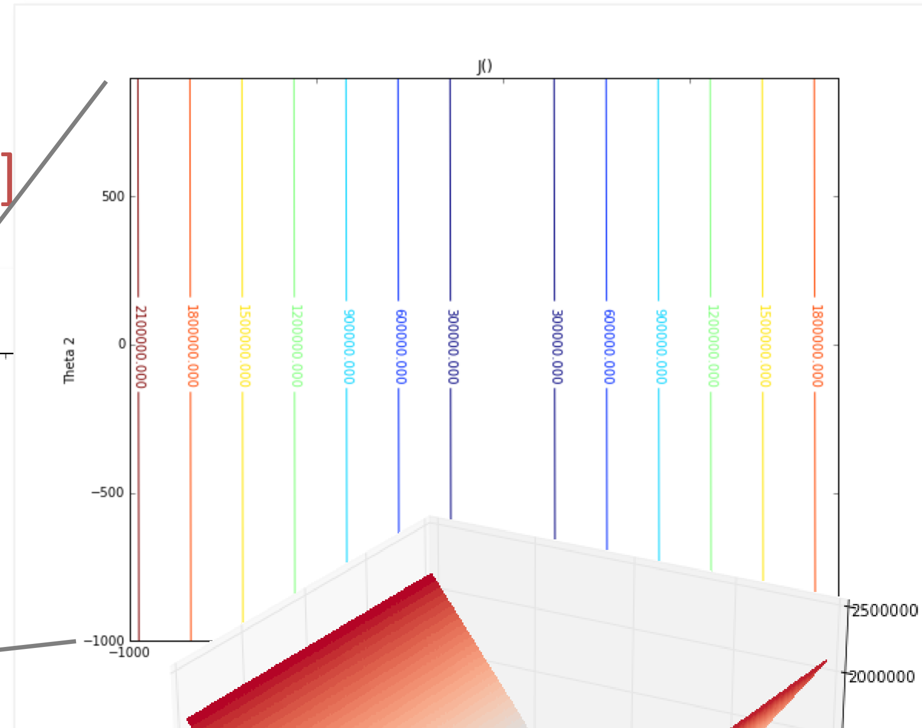
Feature ranges – the problem

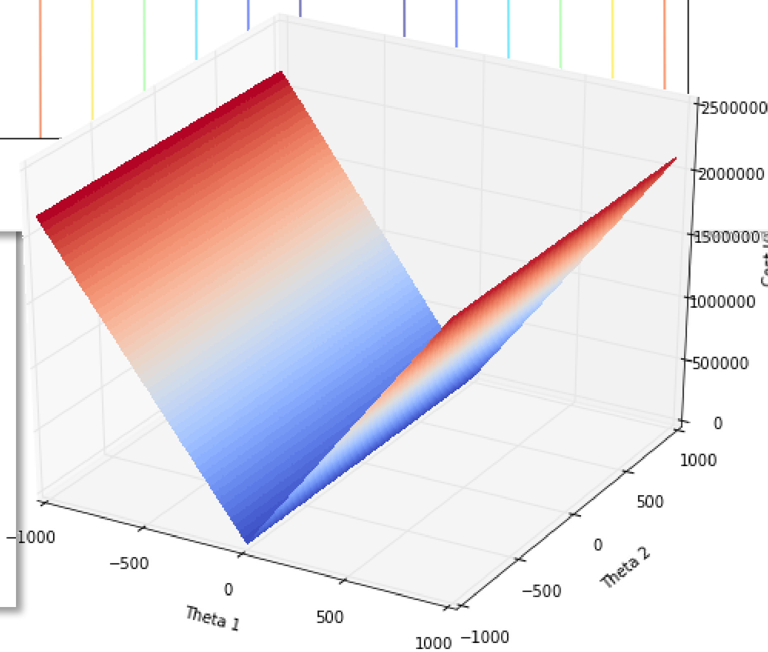
$$h_{\vartheta}(x) = \vartheta_0 x_0 + \vartheta_1 x_1 + \vartheta_2 x_2$$

x_1 = house size (feet^2) [800-5000]

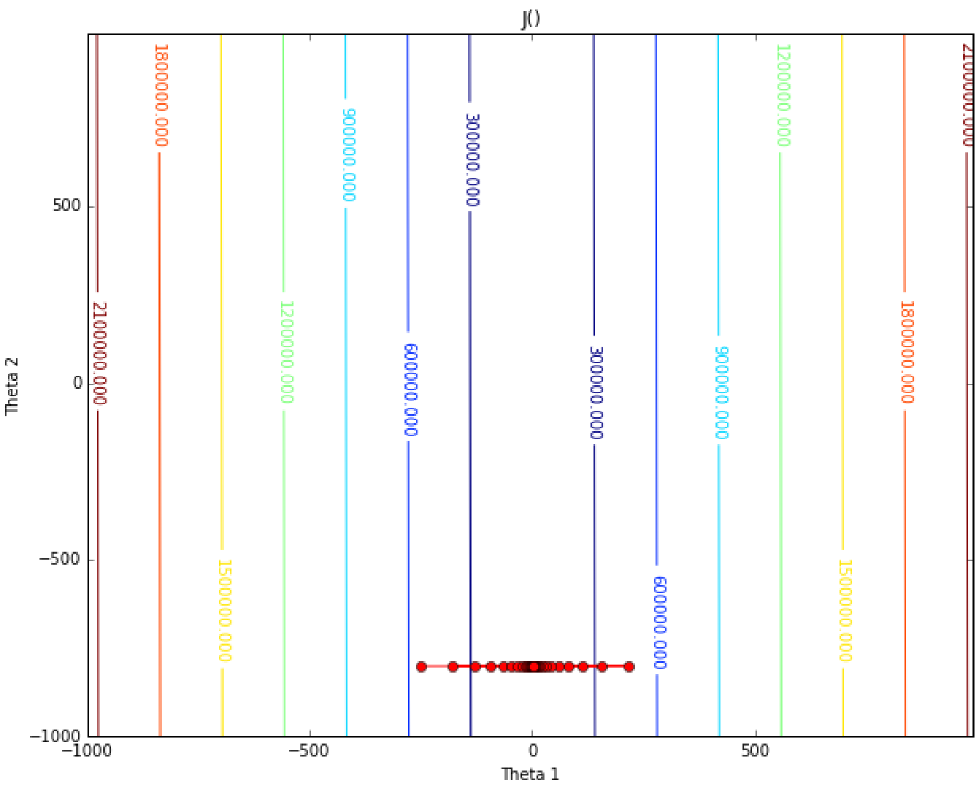
x_2 = # bedrooms [1-5]

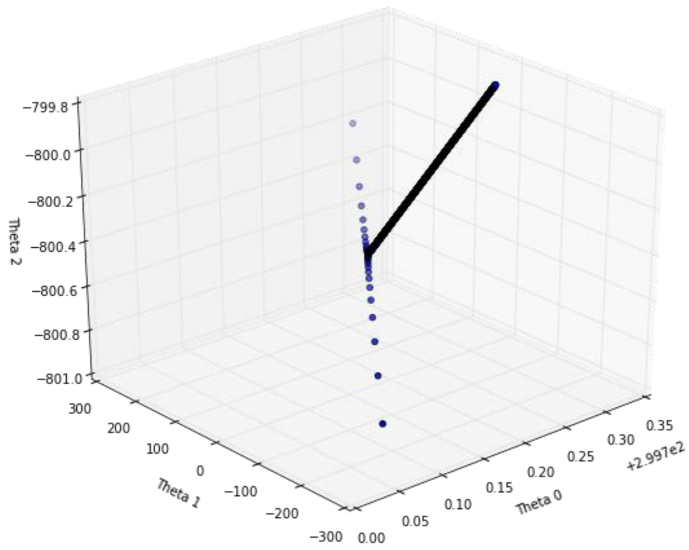


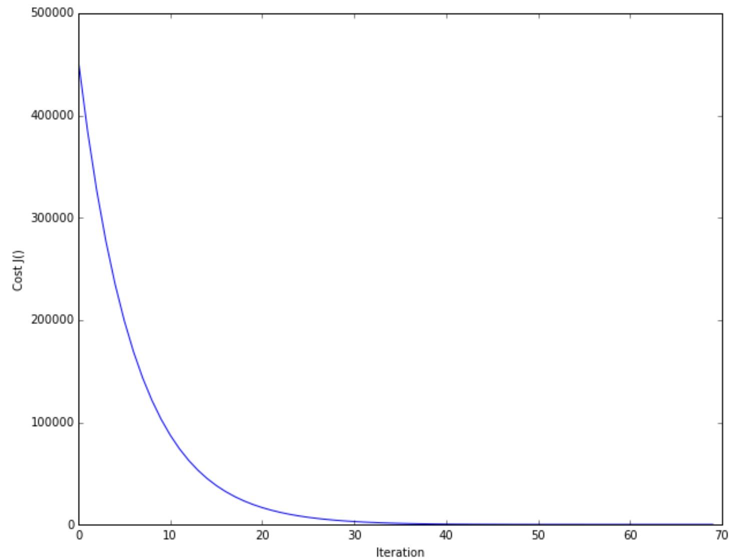




Feature ranges – the problem







Feature scaling – mean normalisation

Aim: get every feature x_j into approximately a $-1 \le x_j \le 1$ range

Mean normalization:

- Subtract from each feature x_j the feature mean (μ_j) to make features have approximately zero mean
- Divide by the feature range or the standard deviation (s_j)
- Do not apply to x_0 !!!

$$x_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{s_j}$$

$$\mu_j = \overline{x_j} = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$$

$$s_j = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(x_j^{(i)} - \mu_j \right)^2}$$

Feature scaling – mean normalisation

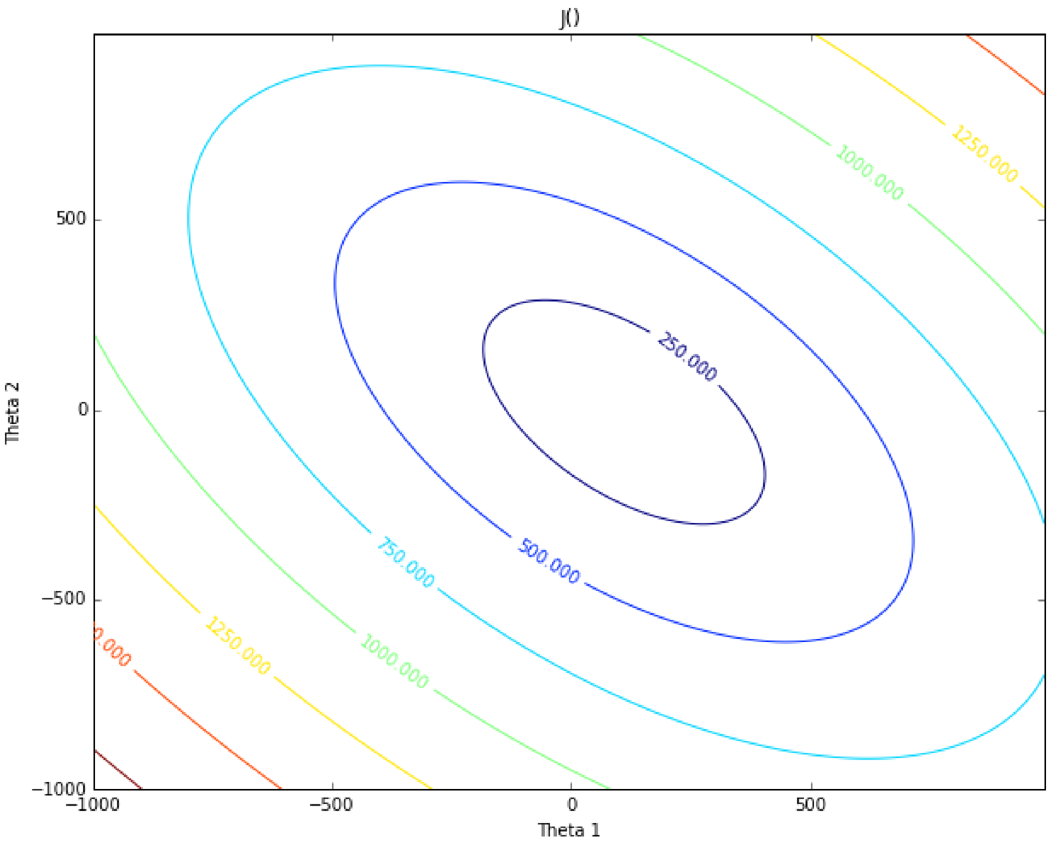
house size ($feet^2$) [800, 5000] → [-1.44, 3.12]

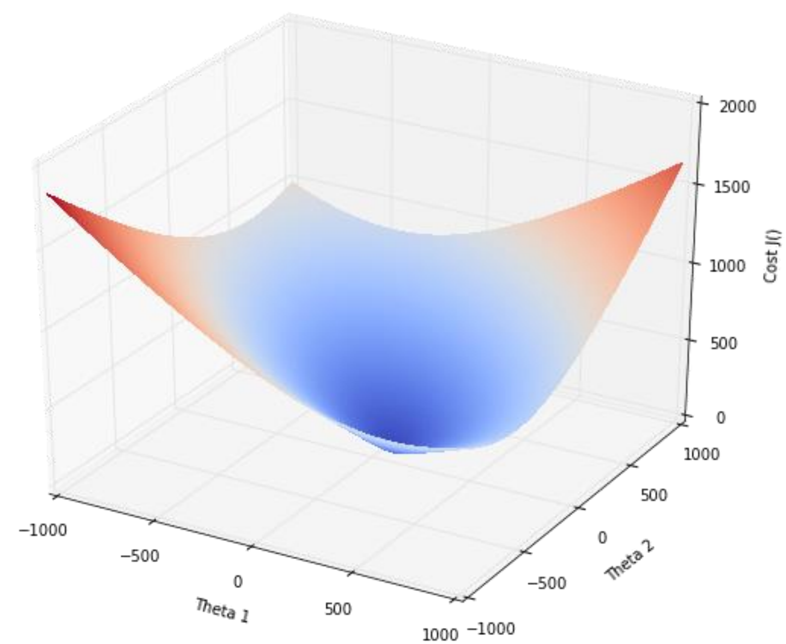
$$x_1^{(i)} = \frac{x_1^{(i)} - \mu_1}{s_1} = \frac{x_1^{(i)} - 2000.68}{794.7}$$

# of bedrooms [1-5] → [-2.85, 2.40]

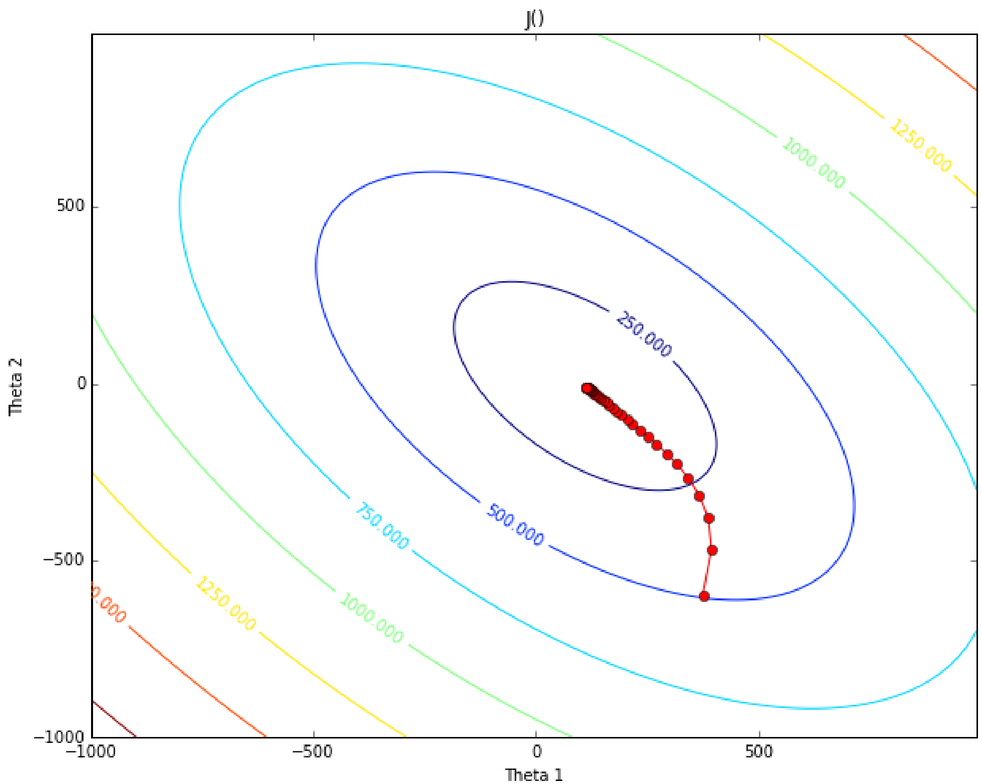
$$x_2^{(i)} = \frac{x_2^{(i)} - \mu_2}{s_2} = \frac{x_2^{(i)} - 3.17}{0.76}$$

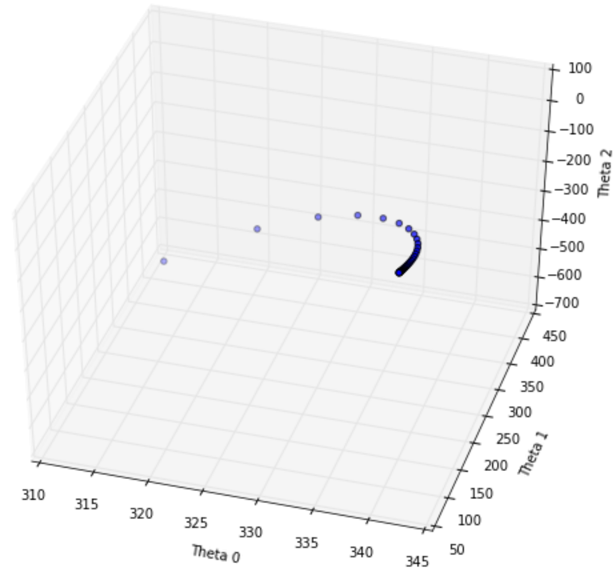
Feature scaling – mean normalisation

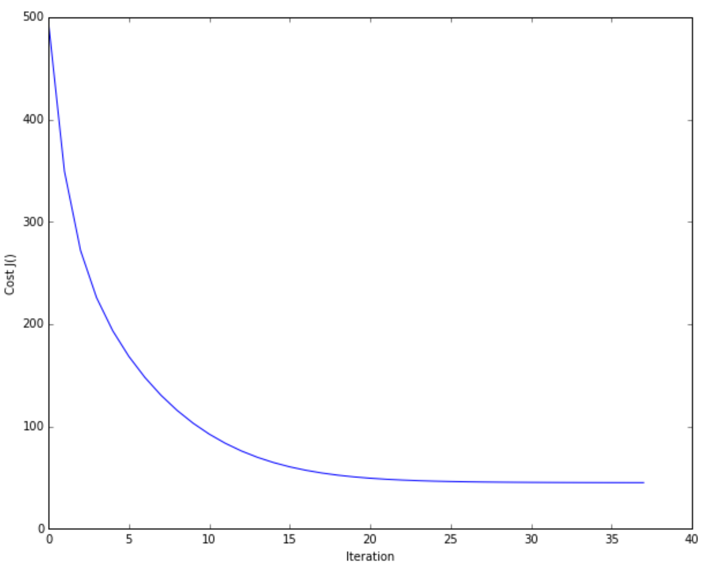




Feature scaling – mean normalisation

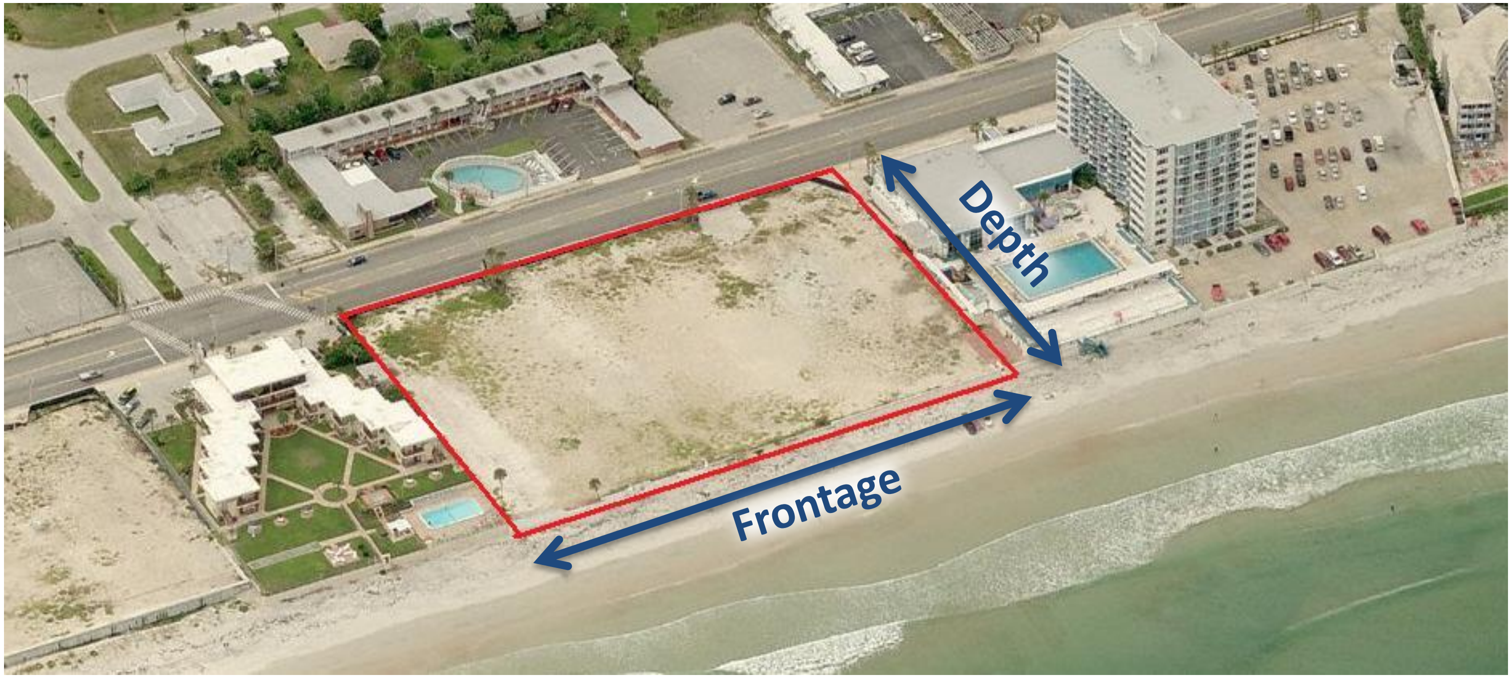






POLYNOMIAL REGRESSION

Creating new features

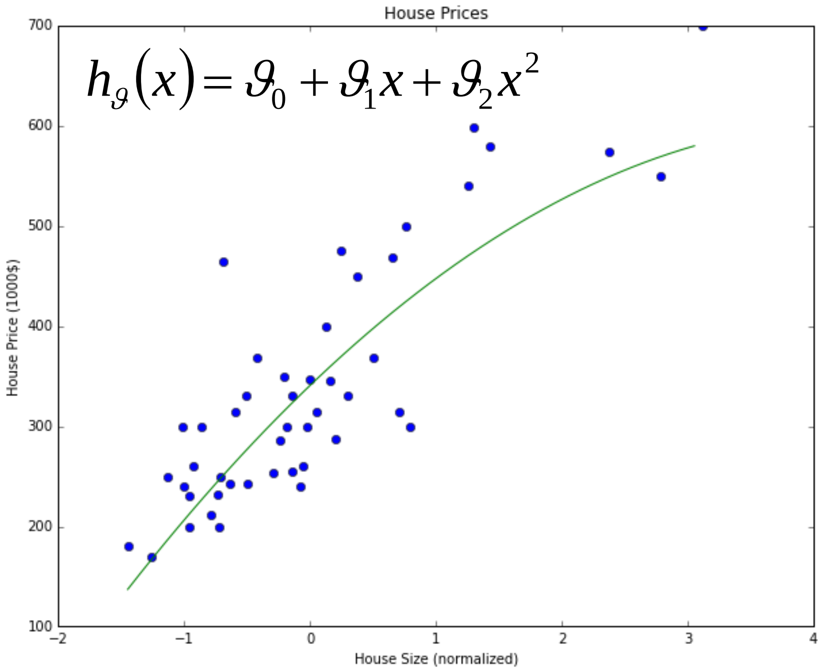


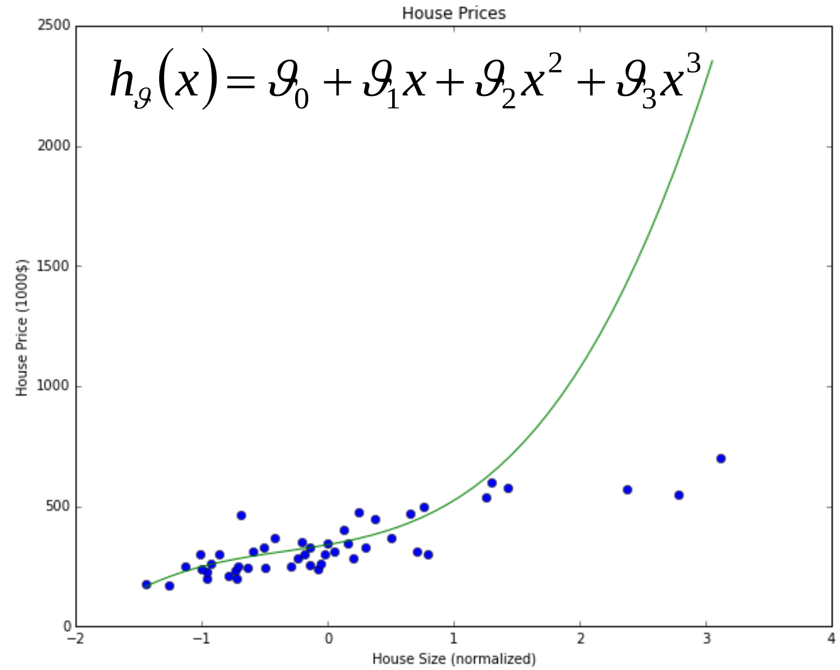
x_1 = *Frontage*

x_2 = *Depth*

x_3 = *Area* = *Frontage* x *Depth* = x_1x_2

Polynomial models for house prices





$$h_\vartheta(x) = \vartheta_0 + \vartheta_1(size) + \vartheta_2(size^2) + \vartheta_3(size^3)$$

$$h_\vartheta(x) = \vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \vartheta_3 x_3 \qquad , \quad \begin{aligned} x_1 &= size \\ x_2 &= size^2 \\ x_3 &= size^3 \end{aligned}$$

NORMAL EQUATION

Normal Equation

$$h_\vartheta(x) = \vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \ldots + \vartheta_n x_n$$

$$J(\vartheta_0, \vartheta_1, \vartheta_2, \ldots, \vartheta_n) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\vartheta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

It is possible to solve for the parameters ϑ_j analytically by setting:

$$\frac{\partial}{\partial \vartheta_j} J(\vartheta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\vartheta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)} = 0 \qquad \text{, for every } j$$

and solving for: $\vartheta_0, \vartheta_1, \ldots, \vartheta_n$

The Generic Case – multiple features

(x_0)	Size (feet^2) (x_1)	Number of bedrooms (x_2)	Number of floors (x_3)	Age of home (years) (x_4)	Price ($1000) (y)
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
…	…	…	…	…	…

$\underbrace{\hphantom{xxxxxxxxxx}}_{n+1}$ $\quad m$

Design matrix

$$X = \underbrace{\begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{n+1} \left.\vphantom{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}}\right\} m \qquad y = \underbrace{\begin{bmatrix} 460 \\ 232 \\ 315 \\ \vdots \end{bmatrix}}_{1} \left.\vphantom{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}}\right\} m$$

Normal Equation

Inverse of Matrix $X^T X$

$$\vartheta = \left(X^T X\right)^{-1} X^T y$$

What if $X^T X$ is non-invertible?

- Redundant features (linearly dependent).
- Too many features (e.g. $m \leq n$).

Solution: delete some features, or use regularization

You can still calculate the pseudo-inverse matrix $(X^T X)^+$

`numpy.linalg.pinv()`

When to use

Gradient Descent

- Need to choose α
- Needs many iterations
- Works well even when n is large
- Needs feature normalization

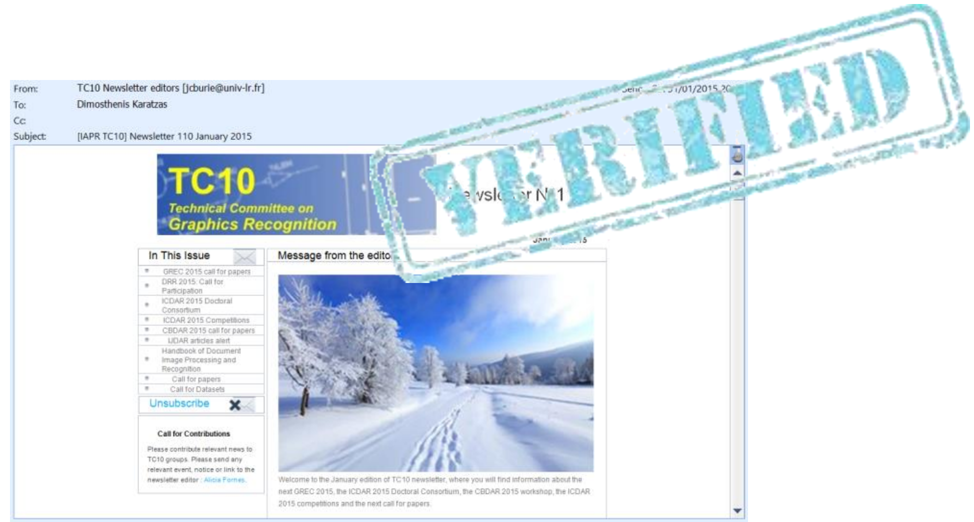
Normal Equation

- No need to choose α
- No need to iterate
- Need to compute $(X^T X)^{-1}$
- Slow if n is very large, complexity $O(n^3)$

LOGISTIC REGRESSION

Classification

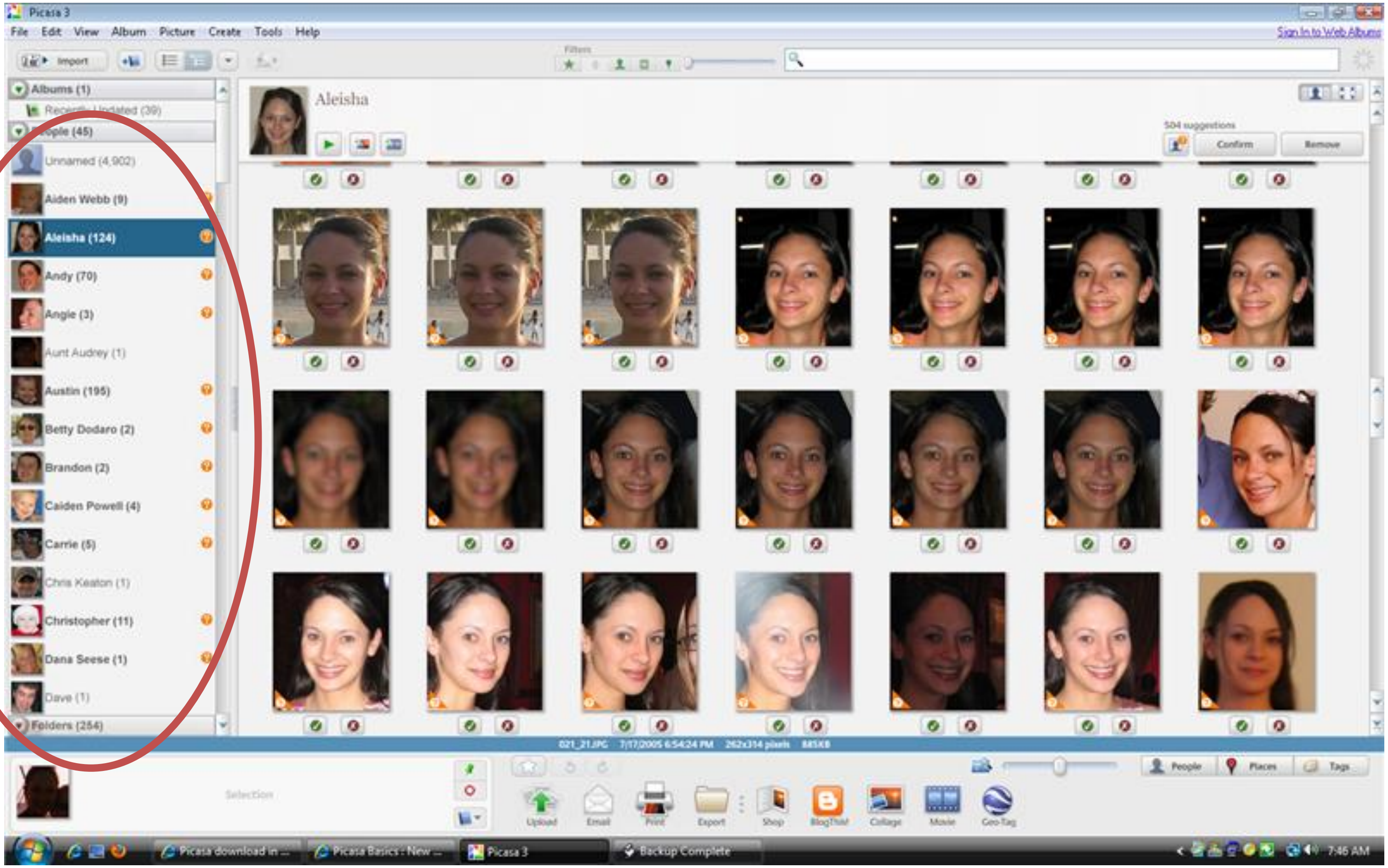
Class = 0



Class = 1

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

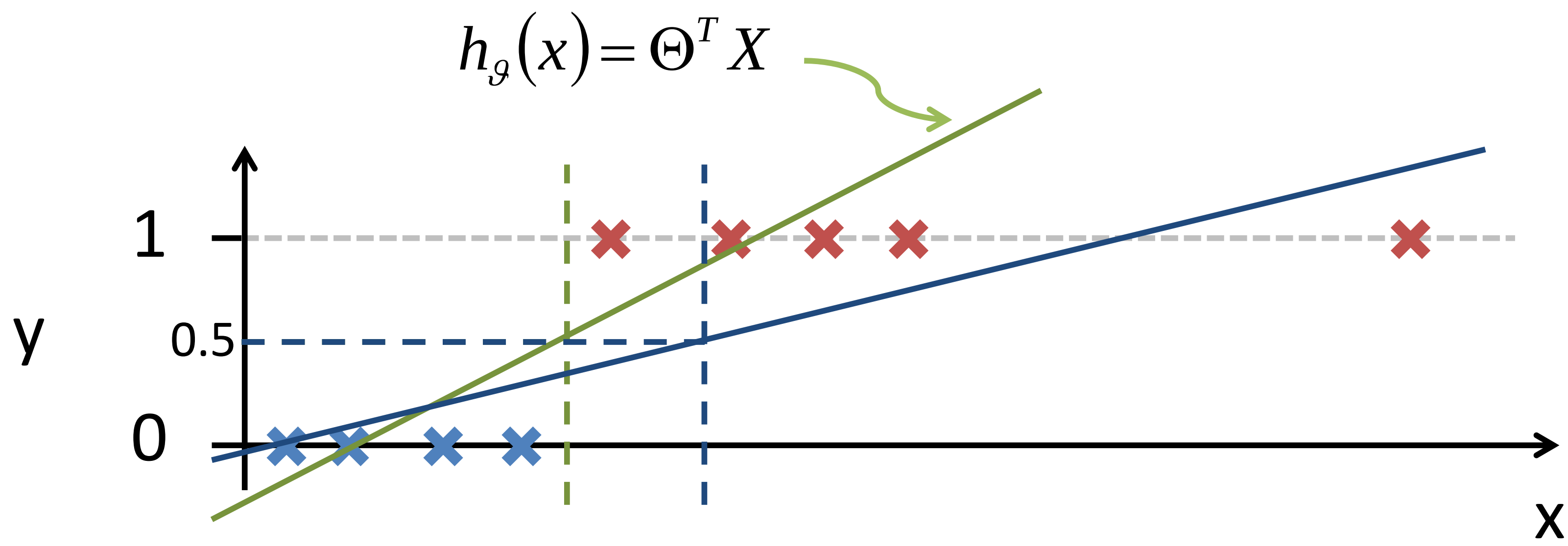
Classification



$$y \in \{0,1,2,3,\dots\}$$

Classification

<u>Classification:</u> Predict discrete valued output (e.g. 0 or 1)

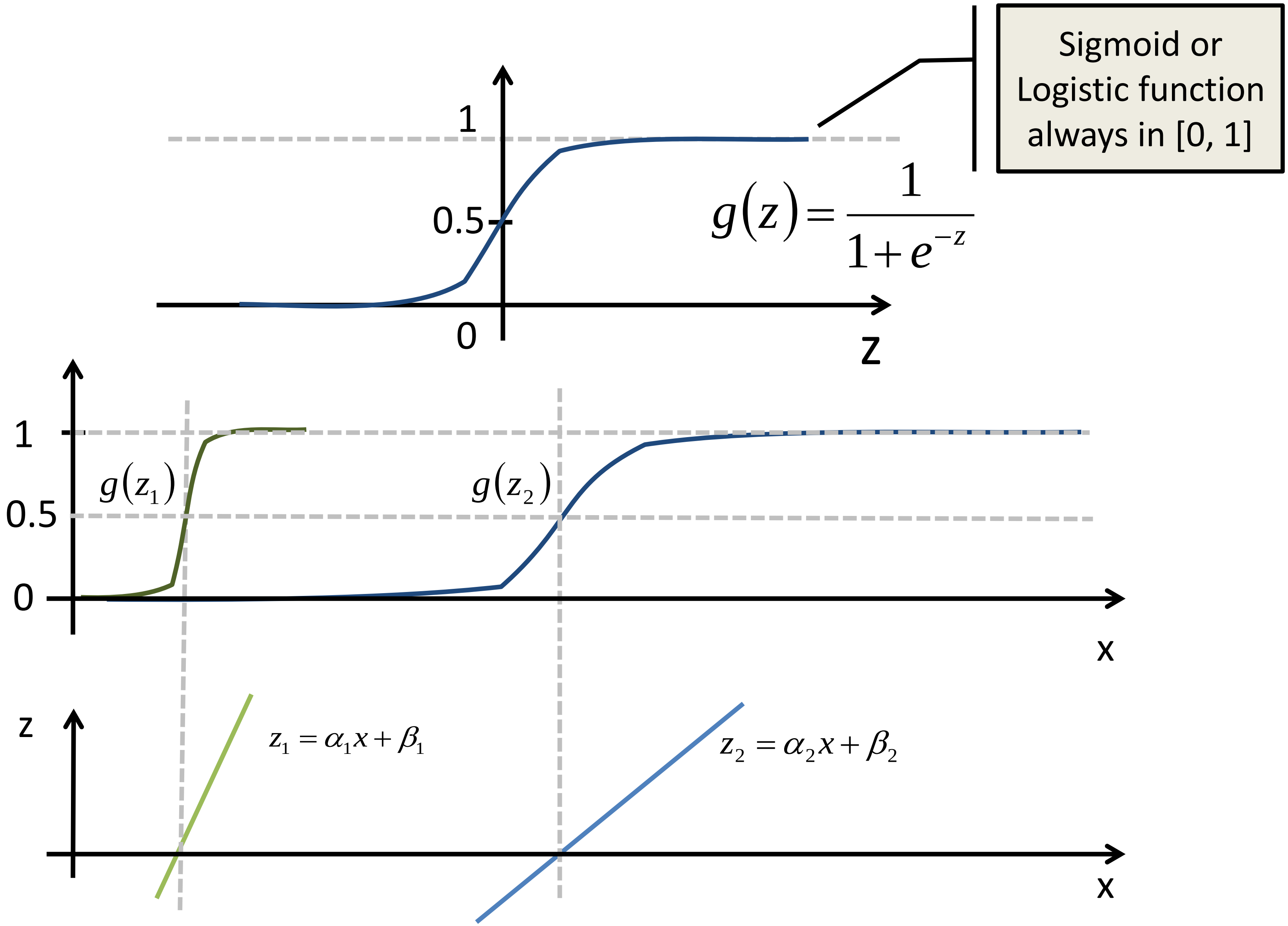


Idea: do linear regression and then threshold the prediction at 0.5 so that:

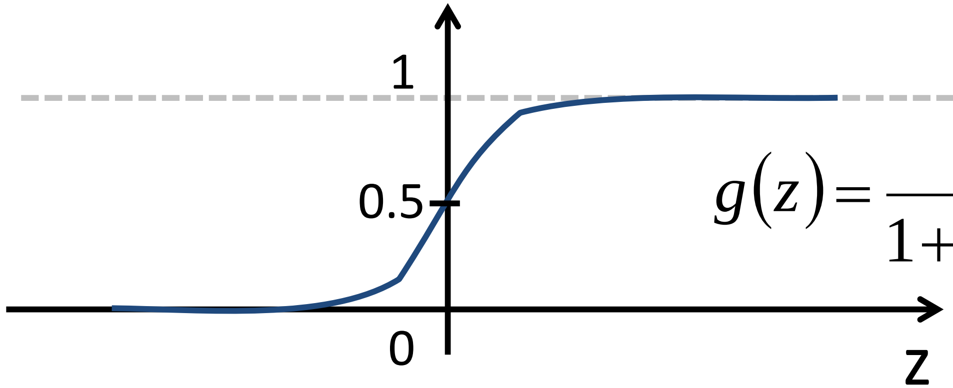
– if $h_\vartheta(x) \geq 0.5$, then class = 1

– if $h_\vartheta(x) < 0.5$, then class = 0

Logistic Regression Model

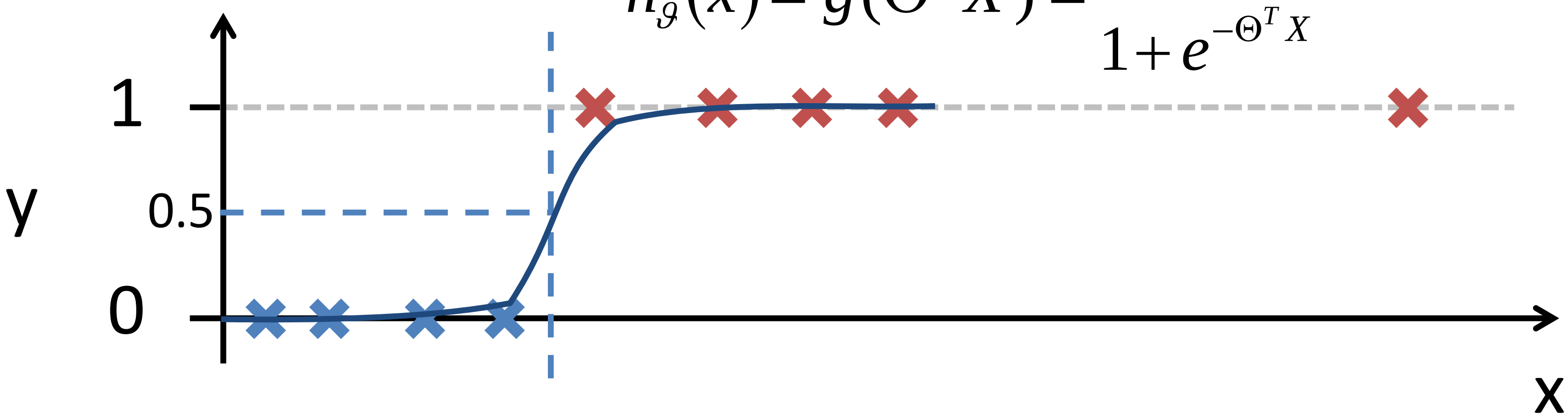


Logistic Regression Model



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

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



Interpretation of hypothesis output

$h_\vartheta(x)$ can be interpreted as the **estimated probability** that $y = 1$ given input x

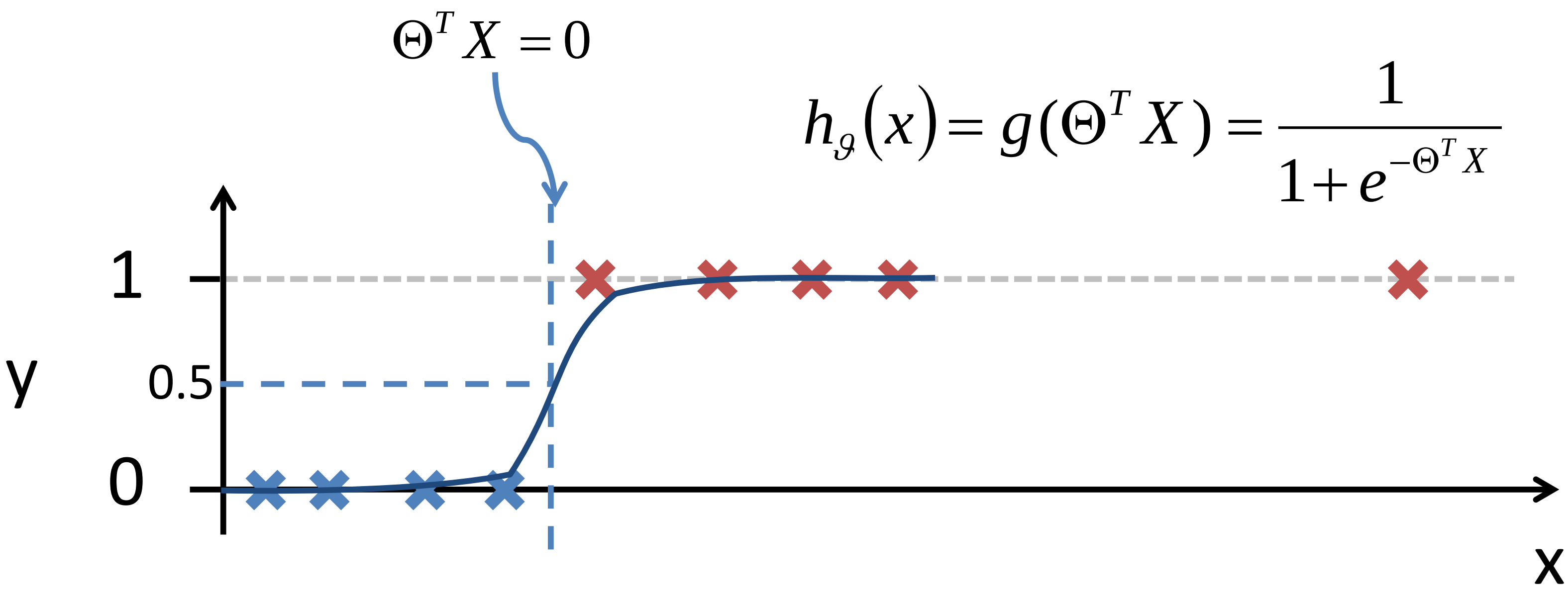
So in the tumor example $h_\vartheta(x) = 0.8$ would mean that the patient has 80% chance of tumor being malignant ($y = 1$)

$$P(y=0|x;\vartheta)+P(y=1|x;\vartheta)=1$$
$$P(y=0|x;\vartheta)=1-P(y=1|x;\vartheta)$$

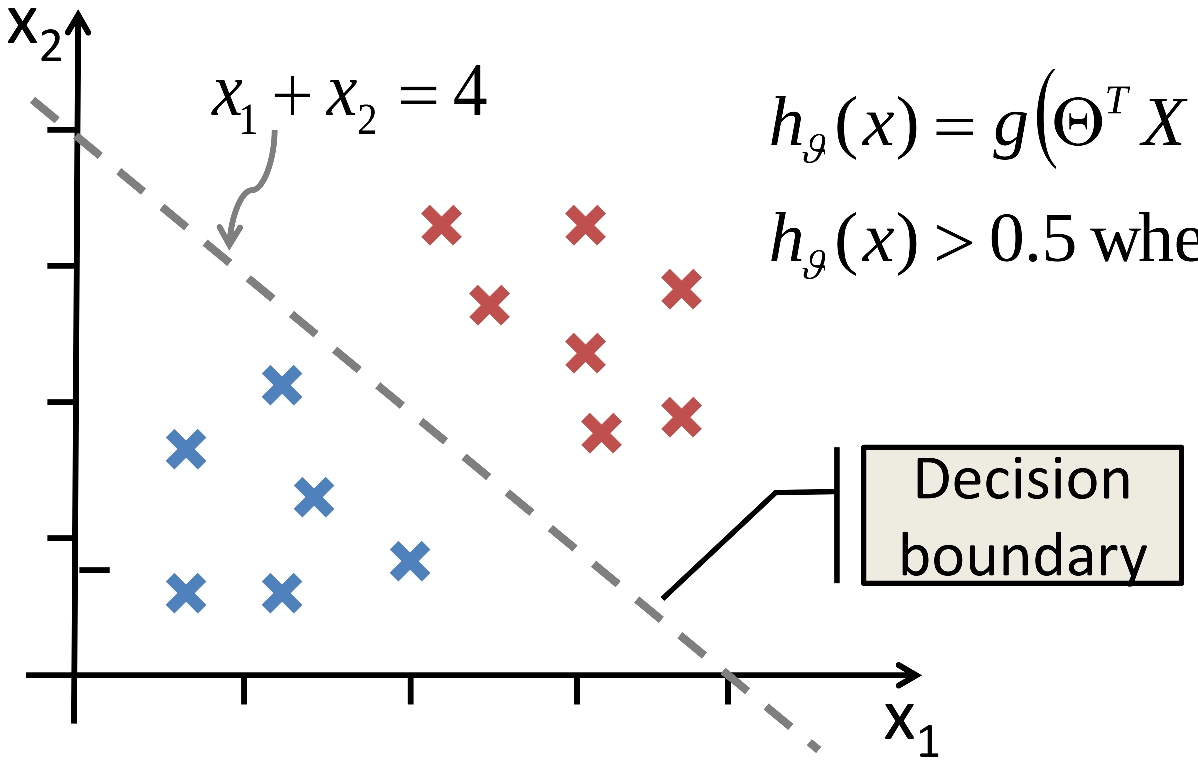
The Decision Boundary

Suppose we predict "*y=1*" if $h_\vartheta(x) \geq 0.5$

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



Decision Boundary



$$h_\vartheta(x) = g\left(\Theta^T X\right) = g\left(\vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2\right)$$

$$h_\vartheta(x) > 0.5 \text{ when } \Theta^T X > 0$$

$$e.g. \quad \Theta = \begin{bmatrix} -4 \\ 1 \\ 1 \end{bmatrix}$$

$$y = 1, \text{ when } \quad h_\vartheta\left(\Theta^T X\right) \geq 0.5 \Leftrightarrow \Theta^T X = -4 + x_1 + x_2 \geq 0$$

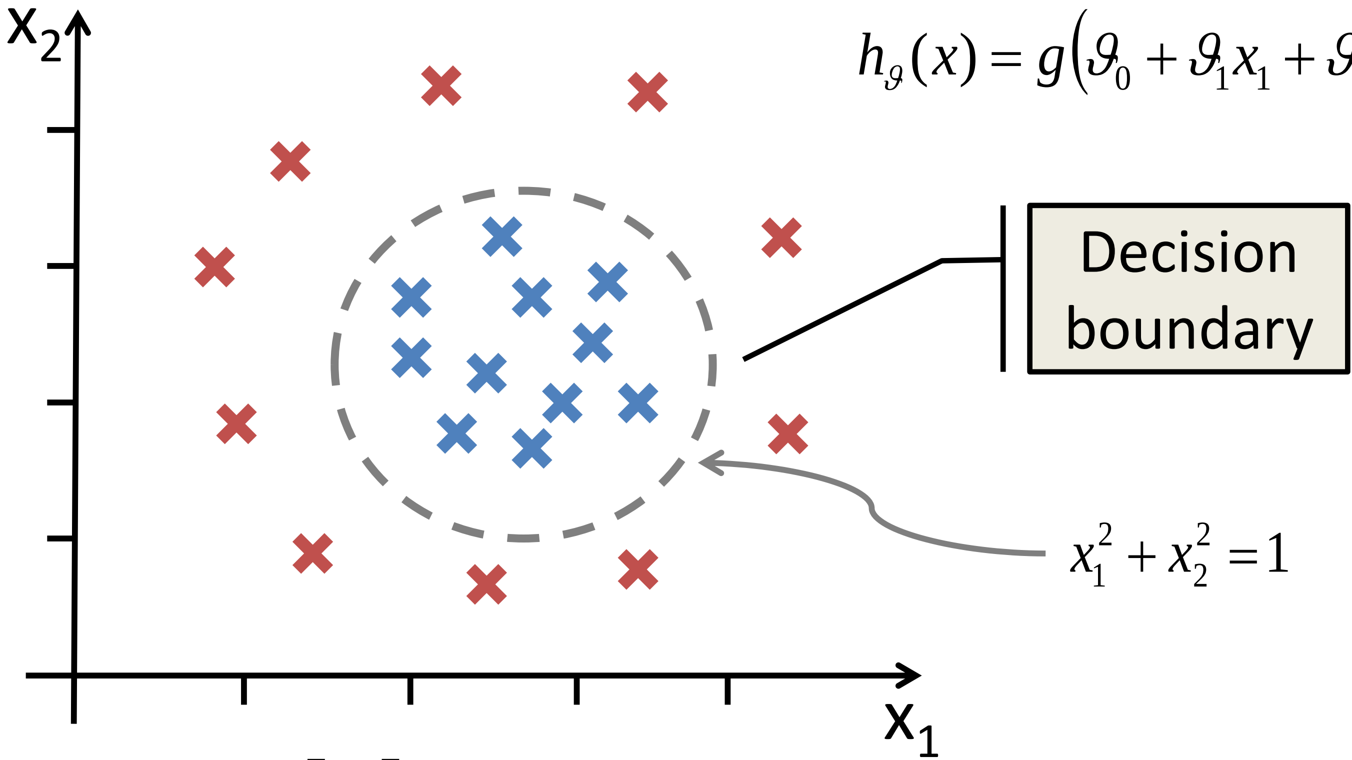
$$y = 1, \text{ when } \quad x_1 + x_2 \geq 4$$

$$y = 0, \text{ when } \quad h_\vartheta\left(\Theta^T X\right) < 0.5 \Leftrightarrow \Theta^T X = -4 + x_1 + x_2 < 0$$

$$y = 0, \text{ when } \quad x_1 + x_2 < 4$$

Non linear decision boundaries

$$h_\vartheta(x) = g\left(\vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \vartheta_3 x_1^2 + \vartheta_4 x_2^2\right)$$



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

$$e.g. \quad \Theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$y = 1, \text{ when } \quad h_\vartheta\left(\Theta^T X\right) \geq 0.5 \Leftrightarrow \Theta^T X = -1 + x_1^2 + x_2^2 \geq 0$$

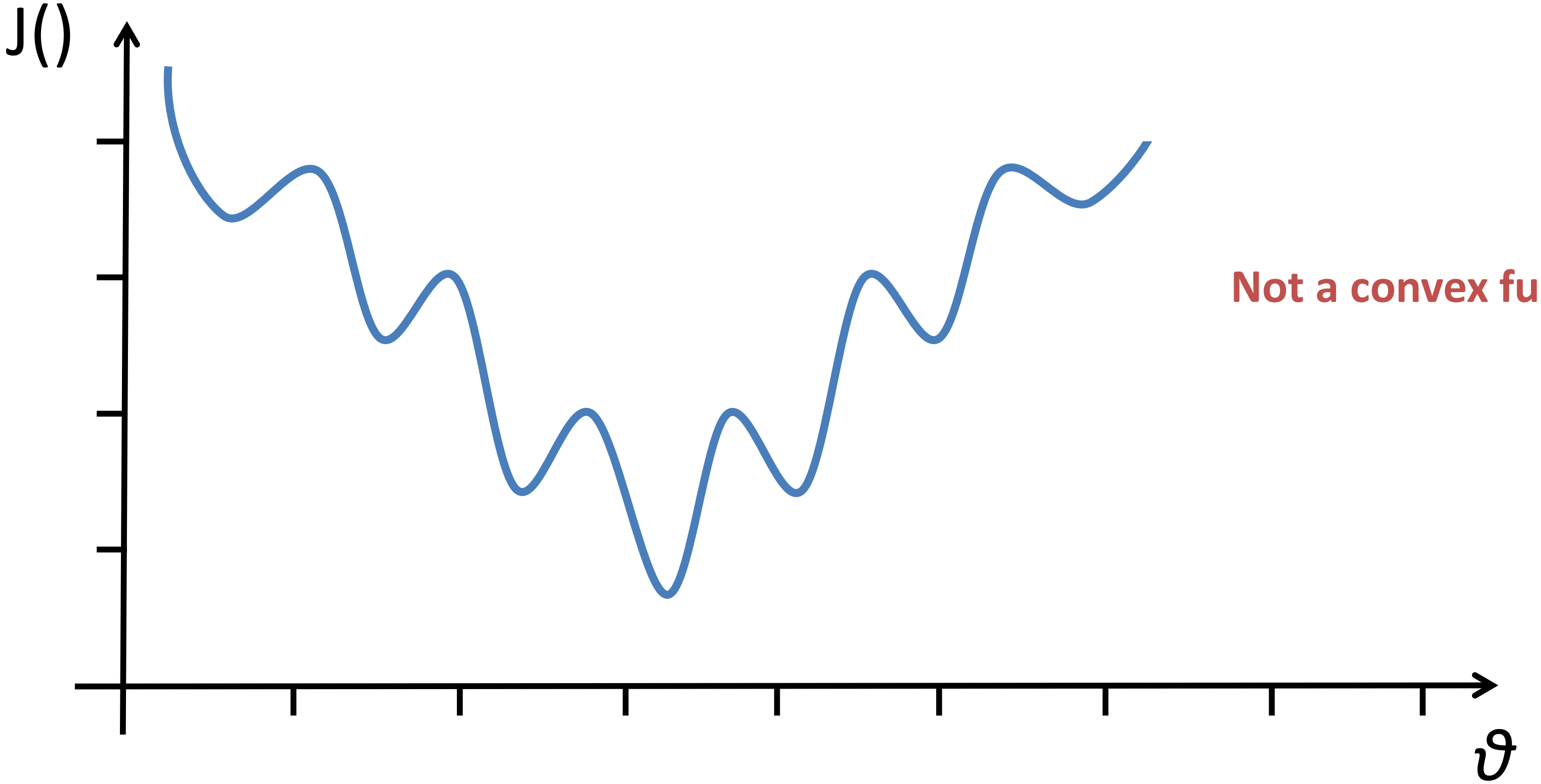
$$y = 1, \text{ when } \quad x_1^2 + x_2^2 \geq 1$$

$$y = 0, \text{ when } \quad h_\vartheta\left(\Theta^T X\right) < 0.5 \Leftrightarrow \Theta^T X = -1 + x_1^2 + x_2^2 < 0$$

$$y = 0, \text{ when } \quad x_1^2 + x_2^2 < 1$$

Cost Function

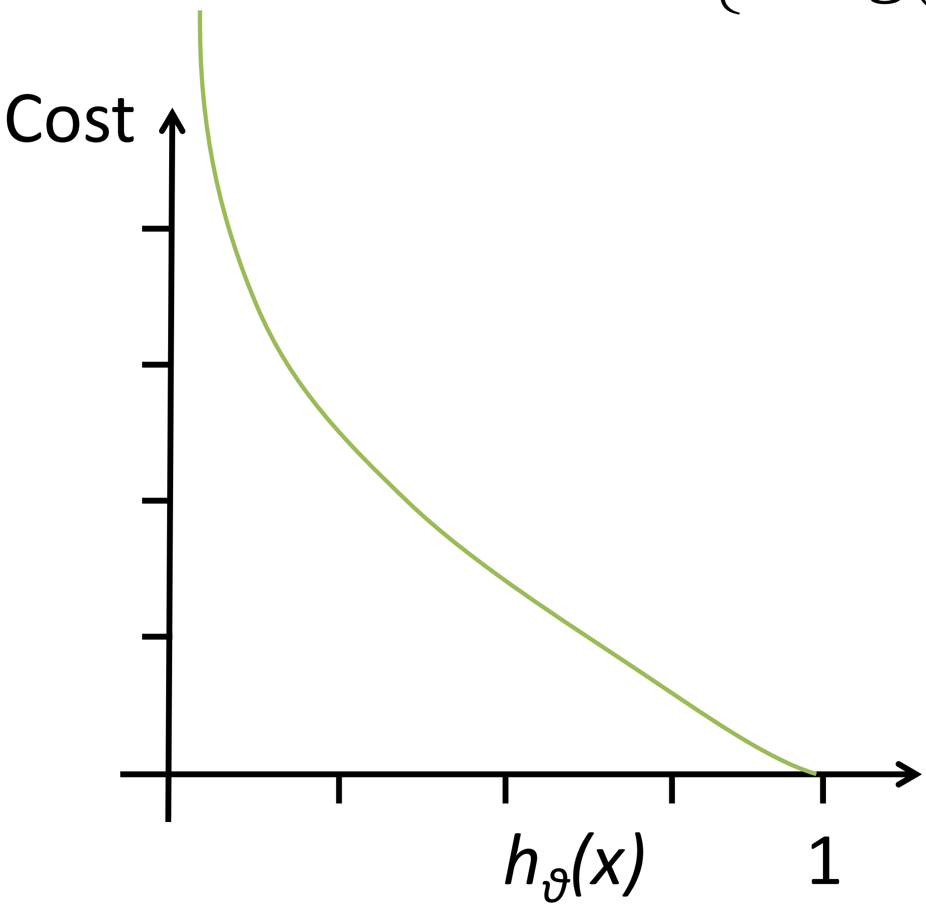
$$J(\vartheta_0, \vartheta_1) = \frac{1}{m} \sum_{i=1}^{m} Cost\left(h_\vartheta\left(x^{(i)}\right), y^{(i)}\right)$$



Not a convex function!

Cost function for logistic regression

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



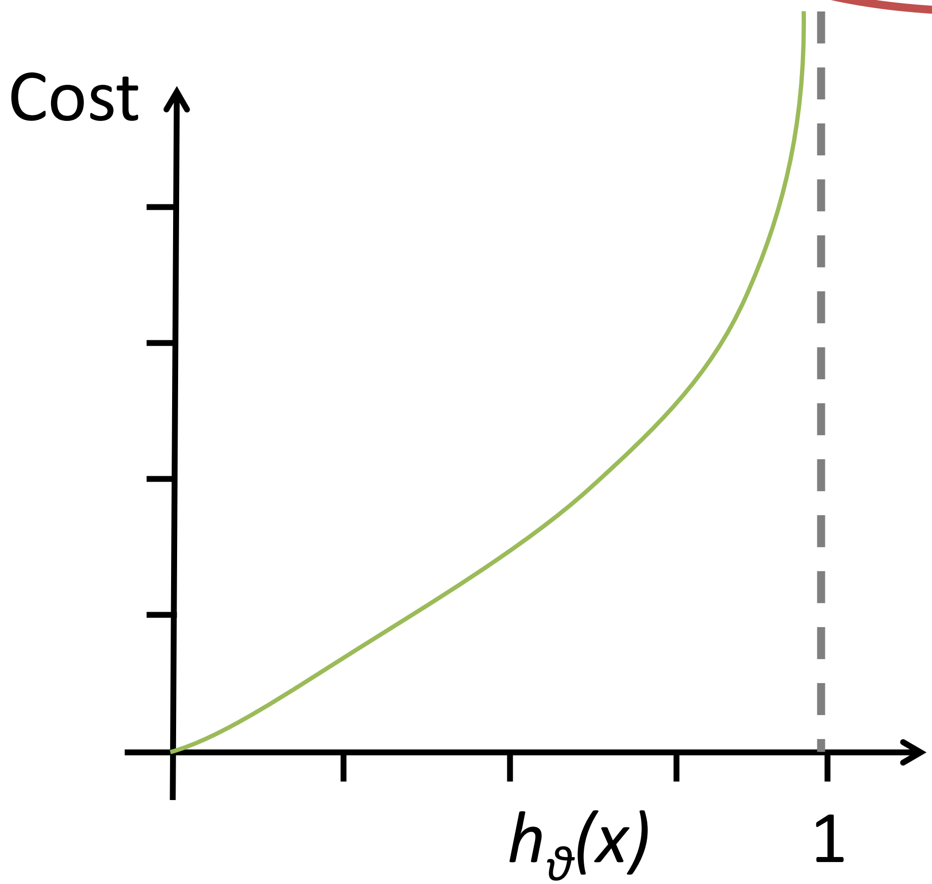
If ***y=1*** AND ***$h_\vartheta(x)$=1*** (real data and our prediction agree) then the ***Cost = 0***

If ***y=1*** BUT ***$h_\vartheta(x)\rightarrow 0$*** (real data and our prediction disagree) then the ***Cost*** **$\rightarrow \infty$**

This captures the intuition that if ***$h_\vartheta(x)$=0*** (our algorithm predicts that *$P(y=1|x,\vartheta)=0$*) but y=1 we will penalise the learning algorithm by a very large cost

Cost function for logistic regression

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



If ***y=0*** AND ***$h_\vartheta(x)$=0*** (real data and our prediction agree) then the ***Cost = 0***

If ***y=0*** BUT ***$h_\vartheta(x)\to 1$*** (real data and our prediction disagree) then the ***Cost*** $\to\infty$

This captures the intuition that if ***$h_\vartheta(x)$=1*** (our algorithm predicts that *P(y=1|x,ϑ)=1*) but y=0 we will penalise the learning algorithm by a very large cost

Simplified cost function

$$J(\vartheta) = \frac{1}{m}\sum_{i=1}^{m} Cost\left(h_\vartheta\left(x^{(i)}\right), y^{(i)}\right)$$

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

(Note that y is always either 0 or 1)

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

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

$$\text{If } y = 0 : Cost\left(h_\vartheta(x), y\right) = -\log\left(1-h_\vartheta(x)\right)$$

Gradient Descent

$$J(\vartheta)=\frac{1}{m}\sum_{i=1}^{m}Cost\left(h_{\vartheta}\left(x^{(i)}\right),y^{(i)}\right)$$

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

To fit parameters θ we want to minimise the cost function J(θ): $\arg\min_{\vartheta} J(\vartheta)$

Repeat

{

$$\vartheta_j := \vartheta_j - \alpha\frac{\partial}{\partial\vartheta_j}J\left(\vartheta\right)$$

(simultaneously for all ϑ_j)

}

Gradient Descent

$$J(\vartheta) = \frac{1}{m}\sum_{i=1}^{m} Cost\left(h_\vartheta\left(x^{(i)}\right), y^{(i)}\right)$$

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

To fit parameters θ we want to minimise the cost function J(θ): $\arg\min_{\vartheta} J(\vartheta)$

Repeat

{

$$\vartheta_j := \vartheta_j - \alpha\sum_{i=1}^{m}\left(h_\vartheta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

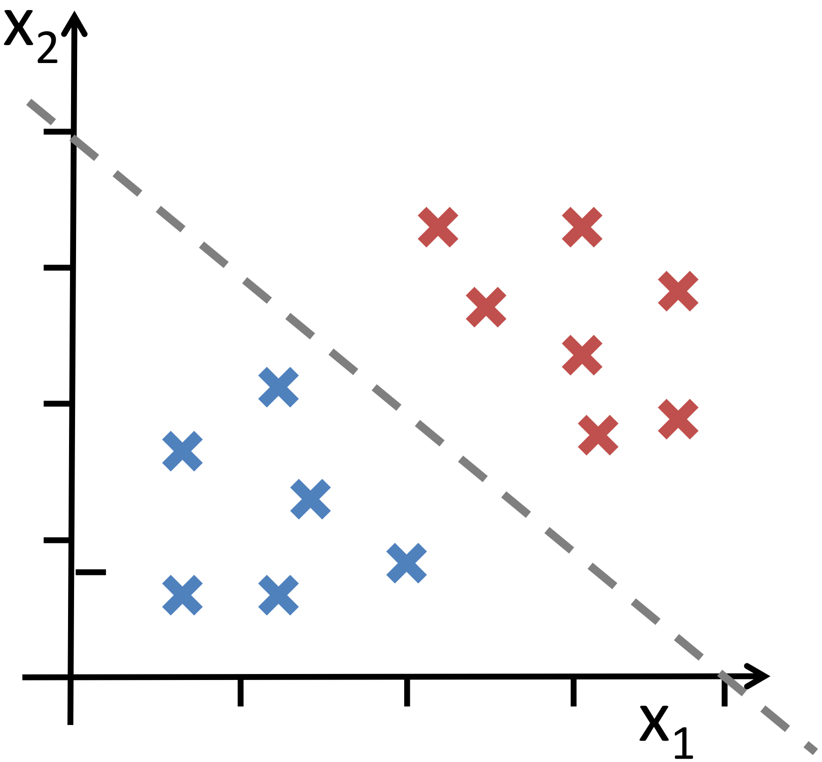
(simultaneously for all ϑ_j)

}

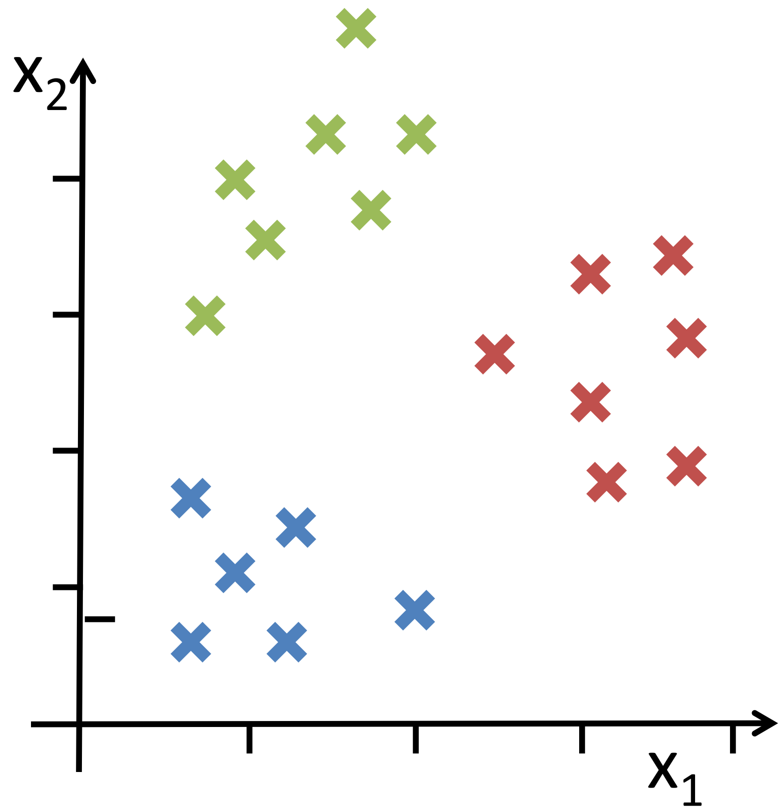
Algorithm looks identical to linear regression!

MULTICLASS CLASSIFICATION

Binary vs Multi-class classification

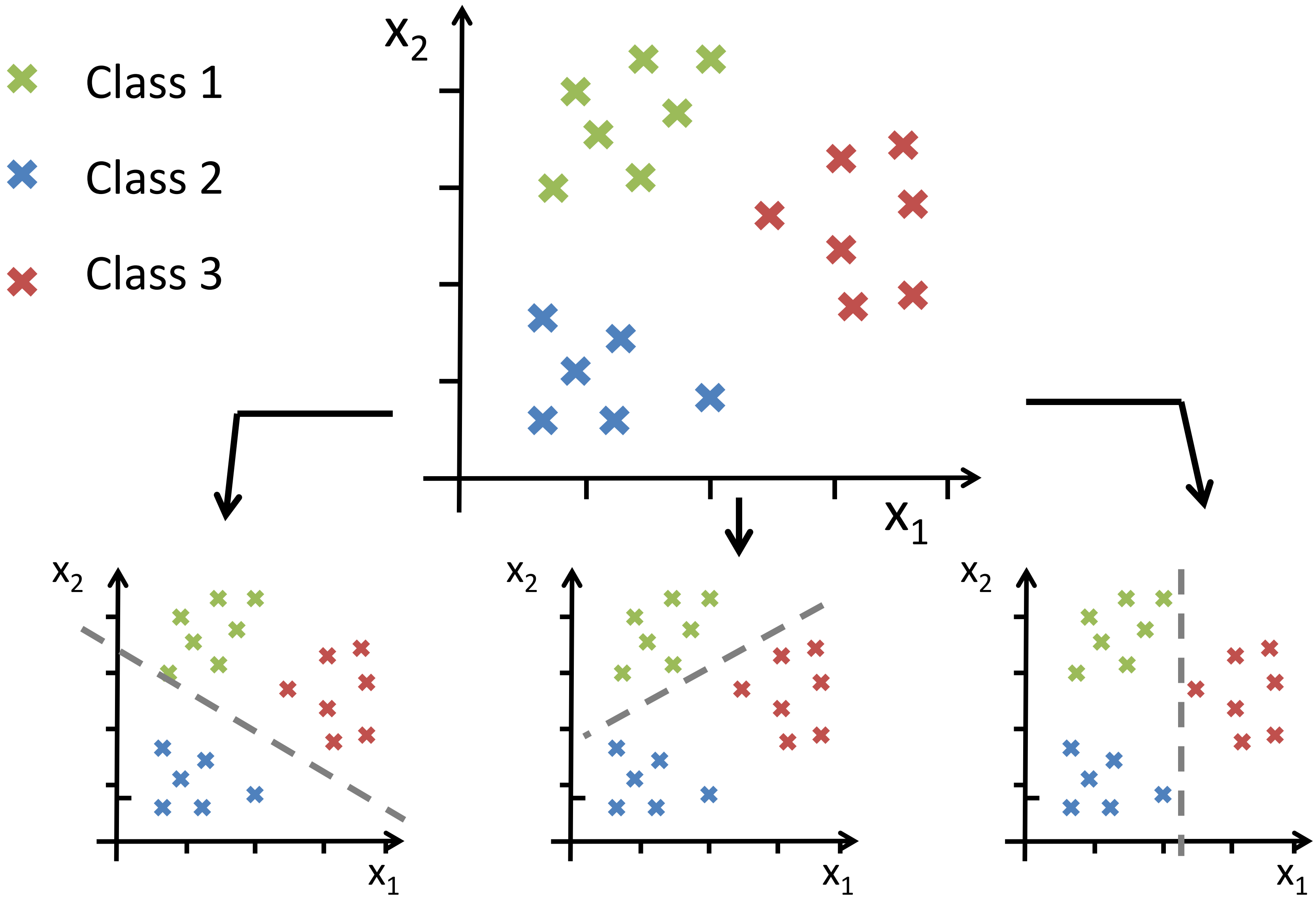


Binary: *y = {0,1}*



Multi-class: *y = {0,1, …, n}*

One vs all (one vs rest)



One vs All Classification

Train a (logistic regression) classifier $h_{\vartheta}^{(c)}(x)$ for each class *c* to predict the probability that *y = c*

On a new input *x*, to make a prediction, pick the class c that maximizes the probability $h_{\vartheta}^{(c)}(x)$

$$\arg\max_{c} h_{\vartheta}^{(c)}(x) \Leftrightarrow \arg\max_{c} P(y = c | x, \vartheta)$$

What's Next

Practical Sessions		M	T	W	T	F	Lectures
	Feb	8	9	10	11	12	Introduction and Linear Regression
P0. Introduction to Python, Linear Regression		15	16	17	18	19	Logistic Regression, Normalization
P1. Text non-text classification (Logistic Regression)		22	23	24	25	26	Regularization, Bias-variance decomposition
	Mar	29	1	2	3	4	Normalization and subspace methods (dimensionality reduction)
		7	8	9	10	11	Probabilities, Bayesian inference
Discussion of intermediate deliverables / project presentations		14	15	16	17	18	Parameter Estimation, Bayesian Classification
		21	22	23	24	25	Easter Week
	Apr	28	29	30	31	1	Clustering, Gausian Mixture Models, Expectation Maximisation
P2. Feature learning (k-means clustering, NN, bag of words)		4	5	6	7	8	Nearest Neighbour Classification
		11	12	13	14	15	
		18	19	20	21	22	Kernel methods
Discussion of intermediate deliverables / project presentations		25	26	27	28	29	Support Vector Machines, Support Vector Regression
P3. Text recognition (multi-class classification using SVMs)	**May**	2	3	4	5	6	Neural Networks
		9	10	11	12	13	Advanced Topics: Metric Learning, Preference Learning
		16	17	18	19	20	Advanced Topics: Deep Nets
Final Project Presentations		23	24	25	26	27	Advanced Topics: Structural Pattern Recognition
	Jun	30	31	1	2	3	Revision

LEGEND	
(blue)	Project Follow Up
(light purple)	Project presentations
(green)	Lectures
(dark purple)	Project Deliverable due date
(pink)	Vacation / No Class