

Imperial College
London

Introduction to Machine Learning

Olivier Dubrule and Lukas Mosser

Imperial College
London

Objectives of the Day

- Understand the main objectives of Machine Learning
- Introduce Supervised vs Unsupervised Learning
- Use Linear Regression as a starting point
- Introduce basic definitions and notations
- Logistic Regression as basic non-linear approach for Supervised Learning
- k-Means and Principal Component Analysis (PCA) as basic Unsupervised Learning approaches

Imperial College
London

Introduction to Machine Learning

1. What is Machine Learning
2. Unsupervised vs Supervised Learning
3. Linear Regression
4. Logistic Regression
5. K-Means and PCA

Imperial College
London

Machine Learning Examples

- **Finance:** Identify patterns preceding a significant financial event
- **Health:** Make a diagnostic from an X-ray or another kind of image
- **Retail:** Offer personal recommendations based on recent choices of products
- **Entertainment:** Offer program proposals based on past viewing or listening choices
- **Marketing:** Organize customers into classes based on their past behaviour

What is common between all these examples?

Availability of large datasets used to calibrate predictions

Imperial College
London

The Data Explosion behind Machine Learning

- 90% of the world's data was created in the last five years.
- 10^{18} bytes of data are created on the Internet every day.
- In 2 days, as many data are created as from the beginning of history to 2003.

Imperial College
London

Machine Learning Definition

- *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,*

(Toni Mitchell, 1997)

Imperial College
London

Classical Machine Learning Example

6	5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5	6	7	8	9	6	4	2	6	4	7	5	5
4	7	8	9	2	9	3	9	3	8	2	0	9	8	0	5	6	0	1	0
4	2	6	5	5	5	4	3	4	1	5	3	0	8	3	0	6	2	7	1
1	8	1	7	1	3	8	5	4	2	0	9	7	6	7	4	1	6	8	4
7	5	1	2	6	7	1	9	8	0	6	9	4	9	9	6	2	3	7	1
9	2	2	5	3	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3
4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	2	1	2	1	3
9	9	8	5	3	7	0	7	7	5	7	9	9	4	7	0	3	4	1	4
4	7	5	8	1	4	8	4	1	8	6	6	4	6	3	5	7	2	5	9

Extract from the MNIST dataset

Machine Learning Problem:

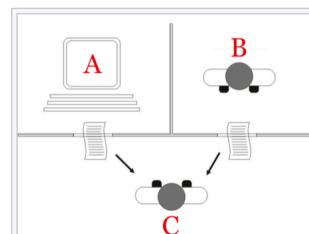
Based on a Training Set of 70,000 labelled digit images (**the experience E**), learn an algorithm that, given a new pixelized input image, automatically recognizes which digit it represents (**the Task T**). **Performance P** measure will be some kind of loss or objective function measured on a Test Set.

MNIST: Modified National Institute of Standards and Technology database

Imperial College
London

Some Definitions of Artificial Intelligence

- Machines that can perform tasks that are characteristics of human intelligence (Chess/Go games are an example)
- Turing test for human intelligence



The "standard interpretation" of the Turing test, in which player C, the interrogator, is given the task of trying to determine which player – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination.^[1] (Wikipedia)

Imperial College
London

Introduction to Machine Learning

1. What is Machine Learning
2. Unsupervised vs Supervised Learning
3. Linear Regression
4. Logistic Regression
5. K-Means and PCA

Imperial College
London

Supervised vs Unsupervised Learning

- **Supervised Learning:** the task T is specified by the user. T is clearly defined because the data are labelled and the computer has to predict the labels of new data.
- **Unsupervised Learning:** the computer is given some unlabeled data and the task T is to organize them based on the recognition of similar patterns in these data. Let the numbers speak for themselves!

Imperial College London

Supervised Learning: Classification

Data : (x, y) where x is image and y is label **Task:** Learn a function to find y when only x is known.

airplane		?	?	?
automobile		?	?	?
bird		?	?	?
cat		?	?	?
deer		?	?	?
dog		?	?	?
frog		?	?	?
horse		?	?	?
ship		?	?	?
truck		?	?	?

Excerpt of the CIFAR-10 dataset, Source: <https://www.cs.toronto.edu/~kriz/cifar.html>

The CIFAR-10 dataset consists of 60000 32x32 colour images labelled in 10 classes, with 6000 images per class.

Imperial College London

Supervised Learning Example: Regression

Task T: from the Training Set of pairs (Temperature, Yoghurt Sold) (**the Experience**) establish a formula for predicting Yoghurt Sold when only the Temperature is known (**the Task**) such that the averaged squared error is minimized (**the Performance**).

<https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithim-choice>

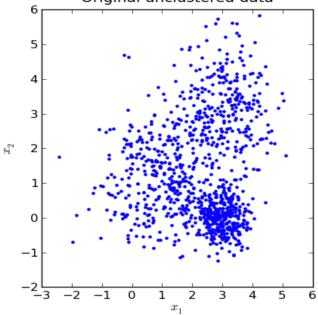
Imperial College London

Unsupervised Learning: Clustering

Data : x only, no label y

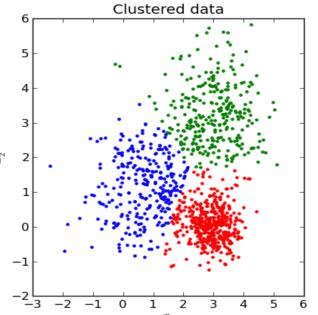
Task : Organize the x 's into classes in which a new x can be classified

Original unclustered data



A scatter plot showing a single, dense cloud of blue data points. The x-axis is labeled x_1 and ranges from -3 to 6. The y-axis is labeled x_2 and ranges from -2 to 6. The points are randomly distributed without any visible structure.

Clustered data



A scatter plot showing the same data points as above, but now organized into three distinct clusters. One cluster is colored red, another green, and the third blue. The x-axis is labeled x_1 and ranges from -3 to 6. The y-axis is labeled x_2 and ranges from -2 to 6. The points are now clearly separated into three groups.

Let the data speak for themselves!

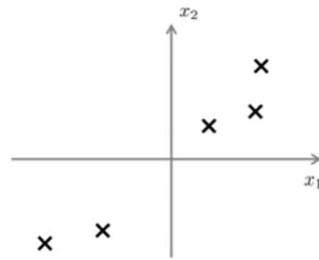
Imperial College London

Clustering Examples

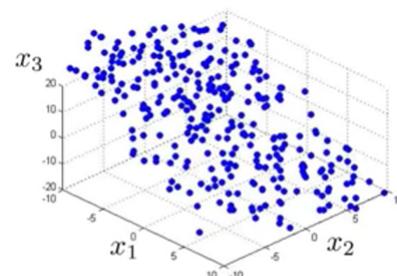
- Sort customers into different categories in order to target them by marketing
- Among many press articles, identify those which deal with the same story
- From a satellite image, identify classes of regions where the data seem to behave in a “similar” fashion

Imperial College
London

Unsupervised Learning: the PCA approach



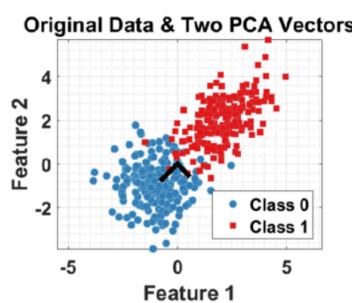
Opportunity to Project 2-D Data
into 1-D space



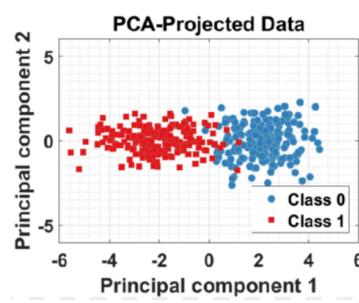
Opportunity to Project 3-D Data
into 2-D space

Imperial College
London

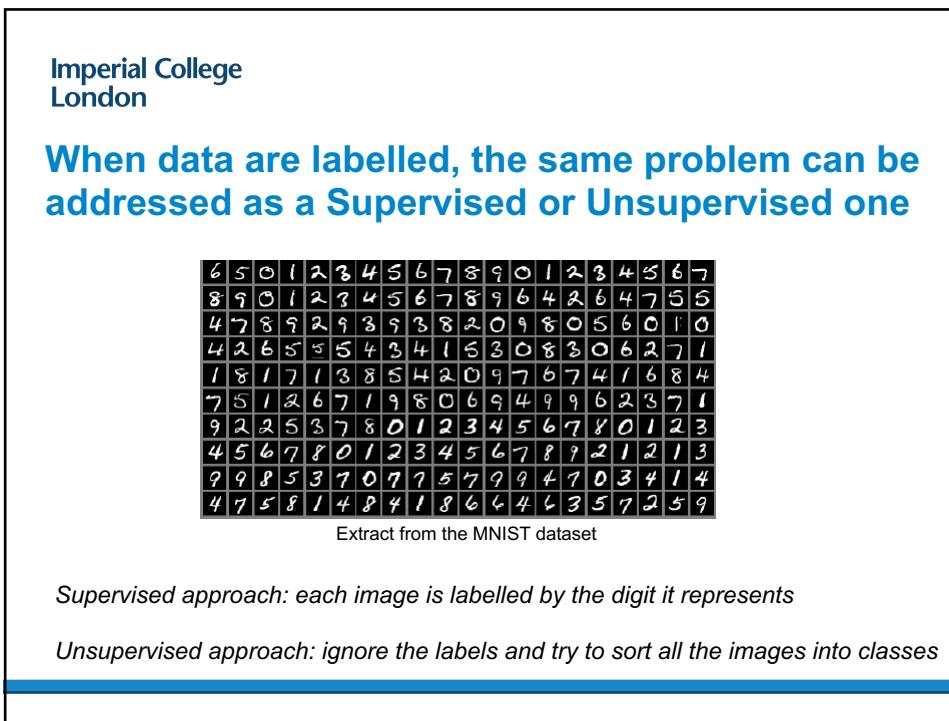
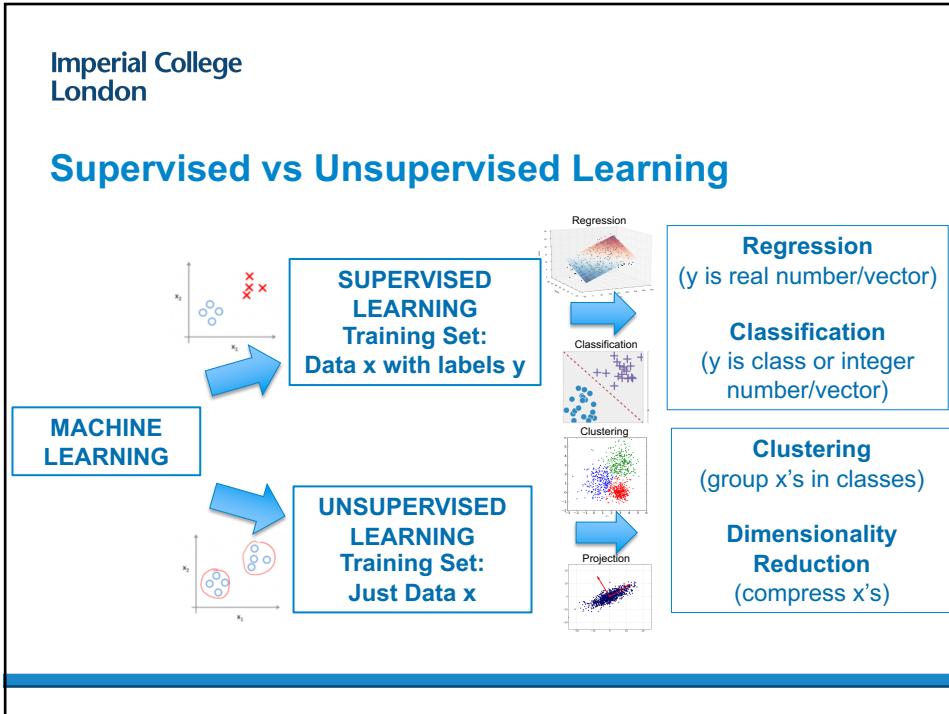
Unsupervised Learning with PCA



Task: identify lower dimension spaces where data can be easily classified



"Unsupervised" because the labels are not used in the PCA calculations



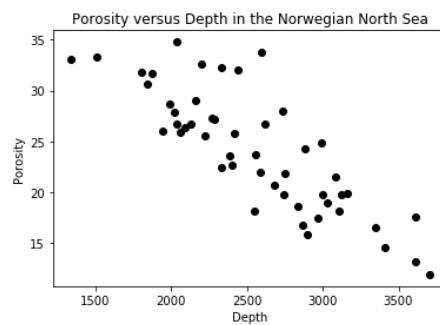
Imperial College
London

Introduction to Machine Learning

1. What is Machine Learning
2. Unsupervised vs Supervised Learning
3. Linear Regression
4. Logistic Regression
5. K-Means and PCA

Imperial College
London

Standard Machine Learning Terminology and Notation



m = number of training examples (here we have $m=50$ pairs of data)

x 's= input variables/**features** (here x is just depth)

y 's= output variables/**target** variables (here y is porosity)

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

Data derived from Ramm and Bjorlykke, 1994

Imperial College
London

What does Linear Regression do?

Fit a linear function to the data. In Machine Learning terminology, this function is called the « **Hypothesis** » $h_{\theta}(x)$

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

θ_0 and θ_1 are called the **parameters** (or the weights) of the model.

For each value of the feature $x^{(i)}$ in the Training Set, we want $h_{\theta}(x^{(i)})$ to be close to the target $y^{(i)}$. In other words, we wish to minimize the **Cost Function** $J(\theta_0, \theta_1)$ (also called **Loss Function**).

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

Imperial College
London

Minimizing the Cost Function for Linear Regression

To simplify, assume that the « **Bias** » term θ_0 in the hypothesis is zero

$$h_{\theta}(x) = \theta_1 x$$

The cost function $J(\theta_0, \theta_1)$ becomes

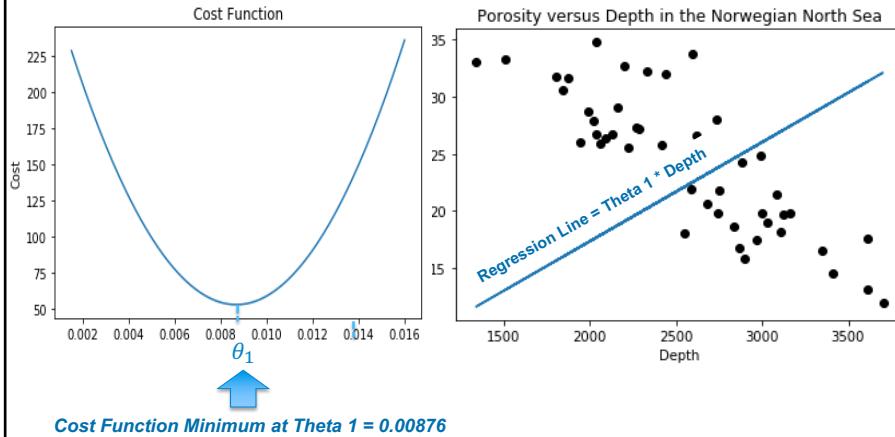
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m (y^{(i)} - \theta_1 x^{(i)})^2 = \theta_1^2 \left(\frac{1}{2m} \sum_{i=1}^m x^{(i)2} \right) - 2\theta_1 \left(\frac{1}{2m} \sum_{i=1}^m x^{(i)} y^{(i)} \right) + \left(\frac{1}{2m} \sum_{i=1}^m y^{(i)2} \right)$$

The cost function is simply a **second degree polynomial in θ_1** , minimum for

$$\theta_1 = \frac{\sum_{i=1}^m x^{(i)} y^{(i)}}{\sum_{i=1}^m x^{(i)2}} \quad (\text{this is called the normal equation})$$

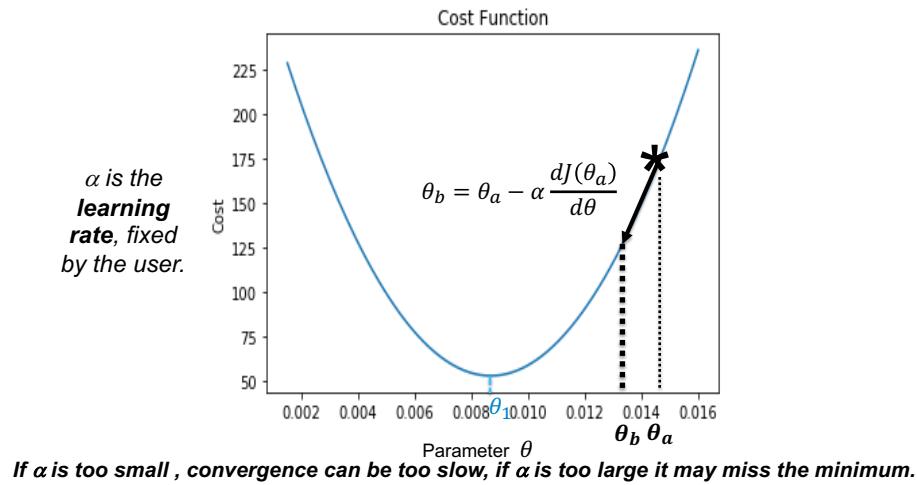
Imperial College
London

Cost Function is Convex for Regression



Imperial College
London

Gradient Descent on a Convex Function



Imperial College
London

Result with a Bias Term in the Linear Regression

The cost function $J(\theta_0, \theta_1)$ can be easily developed

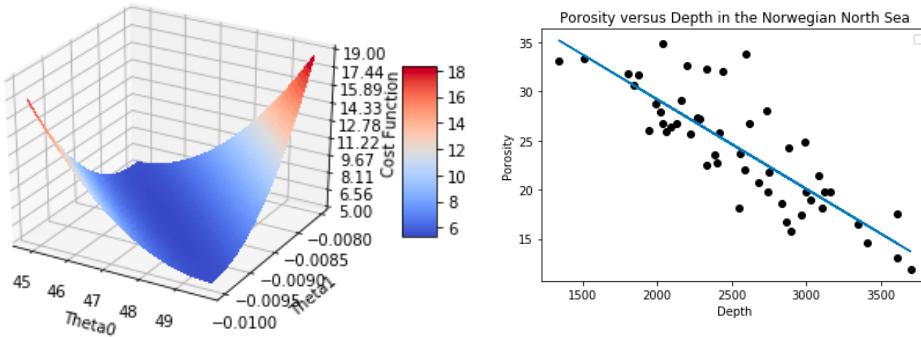
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (y^{(i)} - \theta_0 - \theta_1 x^{(i)})^2 = \frac{1}{2m} \sum_{i=1}^m y^{(i)2} + \frac{\theta_0^2}{2} + \theta_1^2 \left(\frac{1}{2m} \sum_{i=1}^m x^{(i)2} \right) - 2\theta_0 \left(\frac{1}{2m} \sum_{i=1}^m y_i \right) + 2\theta_0\theta_1 \left(\frac{1}{2m} \sum_{i=1}^m x_i \right) - 2\theta_1 \left(\frac{1}{2m} \sum_{i=1}^m x^{(i)} y^{(i)} \right)$$

The cost function is simply a second degree polynomial in θ_0 and θ_1 . It reaches its minimum for the values of θ_0 and θ_1 given by the so-called normal equations

$$\theta_0 = \frac{\left(\frac{1}{m} \sum_{i=1}^m x^{(i)2} \right) \left(\frac{1}{m} \sum_{i=1}^m y^{(i)} \right) - \left(\frac{1}{m} \sum_{i=1}^m x^{(i)} \right) \left(\frac{1}{m} \sum_{i=1}^m x^{(i)} y^{(i)} \right)}{\left(\frac{1}{m} \sum_{i=1}^m x^{(i)2} \right) - \left(\frac{1}{m} \sum_{i=1}^m x^{(i)} \right)^2} \quad \theta_1 = \frac{\frac{1}{m} \sum_{i=1}^m x^{(i)} y^{(i)} - \left(\frac{1}{m} \sum_{i=1}^m x^{(i)} \right) \left(\frac{1}{m} \sum_{i=1}^m y^{(i)} \right)}{\left(\frac{1}{m} \sum_{i=1}^m x^{(i)2} \right) - \left(\frac{1}{m} \sum_{i=1}^m x^{(i)} \right)^2}$$

Imperial College
London

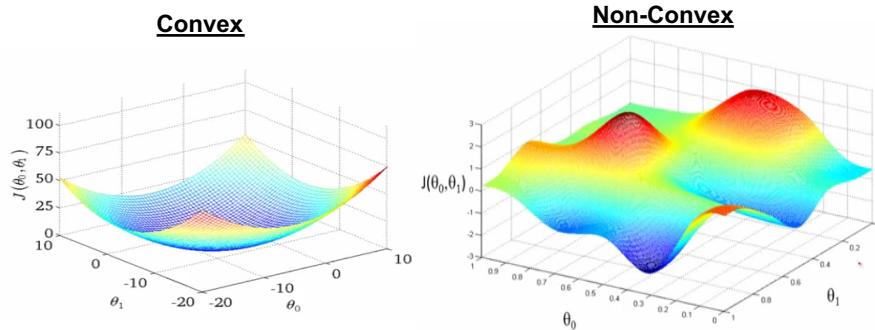
Cost Function for θ_0 and θ_1



A nice property of a convex cost function is that it has no local minima. If you cannot decrease the objective function anymore, it means you have reached the global minimum.

Imperial College
London

Convex vs Non-Convex Objective Function



Unfortunately, many cost functions in non-linear problems are non-convex

Imperial College
London

Multiple Linear Regression

From: $h_{\theta}(x) = \theta_0 + \theta_1 x$

To: $h_{\theta}(x_1, x_2, \dots, x_n) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

$y^{(i)}$ is the value of the target in the example i of the training set

$x_j^{(i)}$ is the value of the feature j in the example i of the training set

m is the number of examples in the training set

n is the number of features in each example of the training set

Imperial College
London

Examples of Multiple Linear Regression

Predict House Price from its squared footage, number of bedrooms, number of bathrooms and location.

Predict child height from both parents sizes, nutrition, environment factor.

Imperial College
London

Vector Notation for Multiple Linear Regression

The hypothesis function is, if we have n input features x_i :

$$y = h_{\theta}(x_1, x_2, \dots, x_n) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

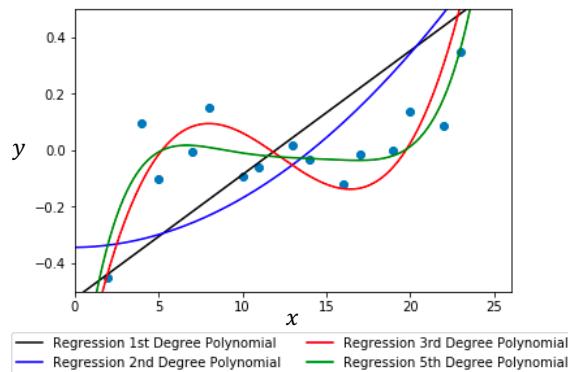
With the vector notation $x = \begin{pmatrix} 1 \\ x_1 \\ \dots \\ x_n \end{pmatrix}$ $\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{pmatrix}$ We have $h_{\theta}(x) = \theta^T x$

We write the m data vectors as: $(x^{(i)}, y^{(i)})$ $x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ \dots \\ x_n^{(i)} \end{pmatrix}$ and $y^{(i)}$ real for $i = 1 \dots m$

Imperial College
London

Example of Non-Linear Regression

$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_n x^n$$



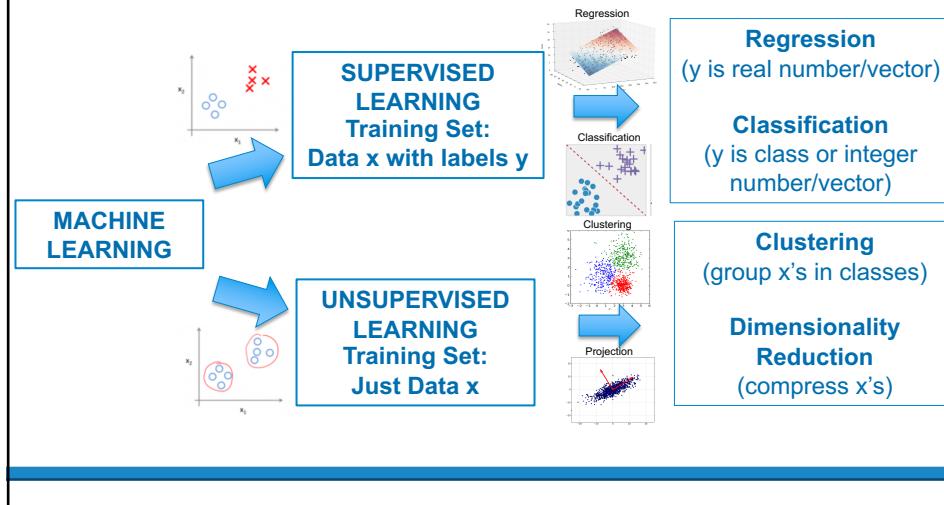
Imperial College
London

Introduction to Machine Learning

1. What is Machine Learning
2. Unsupervised vs Supervised Learning
3. Linear Regression
4. Logistic Regression
5. K-Means and PCA

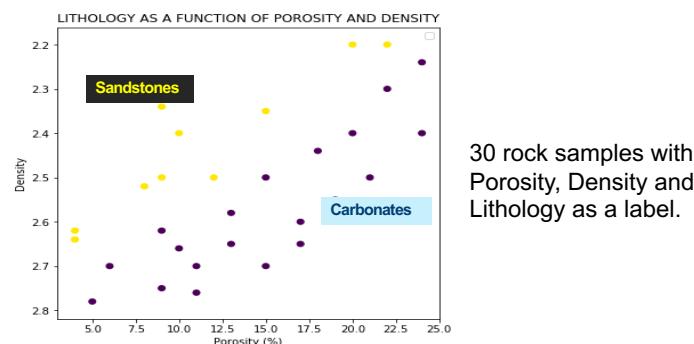
Imperial College
London

Reminder: Supervised vs Unsupervised Learning



Imperial College
London

Logistic Regression Problem

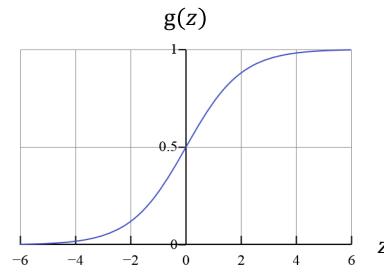


We wish to predict Lithology from Density and Porosity: this is a Supervised Classification problem .
Cannot use Linear Regression : need a transform from the domain of real values to the 0 or 1 indicator

Imperial College
London

Sigmoid Function for transformation to [0,1] domain.

- It is also called the **Logistic** Function
- It takes any real value z and transforms it into a value between 0 and 1



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

It is easy to prove that

$$g'(z) = (1 - g(z))g(z)$$

Imperial College
London

Interpreting the output of the Logistic function

- Say that y is the outcome of a regression equation for an input x

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

- In vector form:

$$y = \theta^T x$$

- y is a real number which can take any positive or negative value.

- If we apply the logistic (or sigmoid) function $g(y)$ we obtain a value between 0 and 1, which **we interpret as the probability for the class to be 1**

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

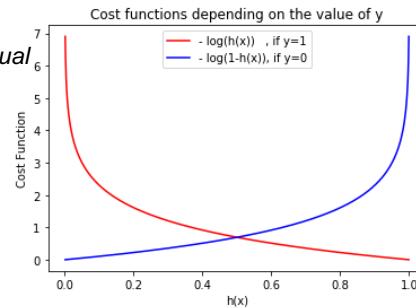
Imperial College
London

Cost Function for Logistic Regression (one data point)

For one data point , how to measure the discrepancy between the actual data value y (equal to 0 or 1) and the estimated probability $h_\theta(x)$?

$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \quad \text{if } y = 1$$

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \quad \text{if } y = 0$$



Combining the two possibilities above into one single line:

$$\text{Cost}(h_\theta(x), y) = -y\log(h_\theta(x)) - (1 - y)\log(1 - h_\theta(x))$$

Imperial College
London

Cost function for a Training whole Set: $\{(x^{(i)}, y^{(i)})\}_{i=1}^m$

$$\begin{aligned} J(\theta) &= \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_\theta(x^{(i)}), y^{(i)}) \\ &= -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))] \end{aligned}$$

= Cross-Entropy

To minimize, just calculate the derivatives $\frac{\partial J(\theta)}{\partial \theta_j}$ for $j = 1 \dots n$ and apply Gradient Descent

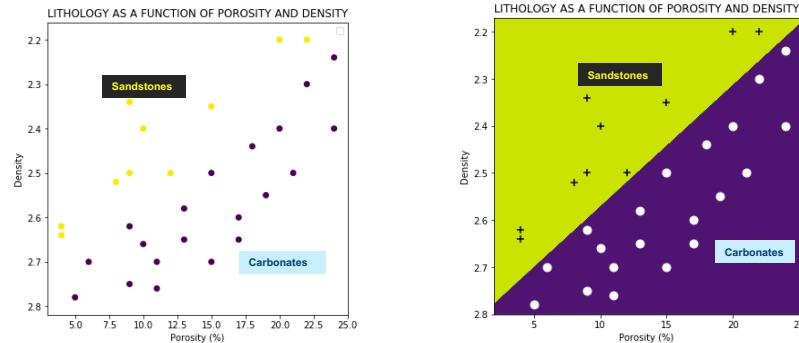
$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m [h_\theta(x^{(i)}) - y^{(i)}] x_j^{(i)}$$

Exercise: prove this relationship!

In spite of its name, Logistic Regression ifor Classification rather than Regression!

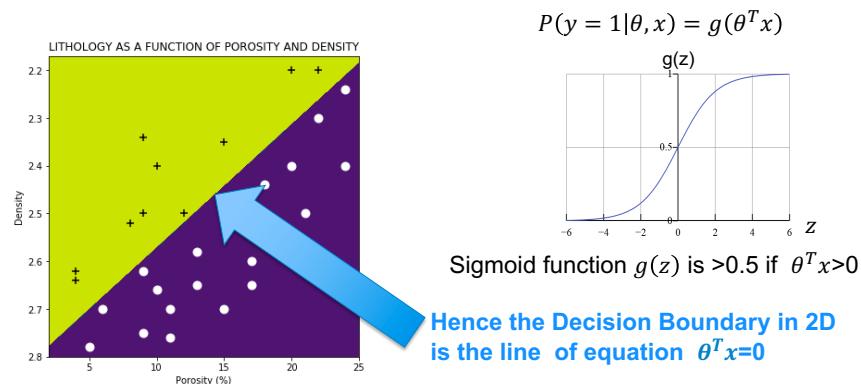
Imperial College
London

Example of Binary Logistic Regression Result



Imperial College
London

Definition of the Decision Boundary



Note that if the data are linearly separable in the feature space (as here), there is an infinity of "optimal" logistic regressions or of weights θ , or Decision Boundaries.

Imperial College
London

Multi-Class Classification with Logistic Regression The “One vs Rest” Approach. Example of 3 Classes.

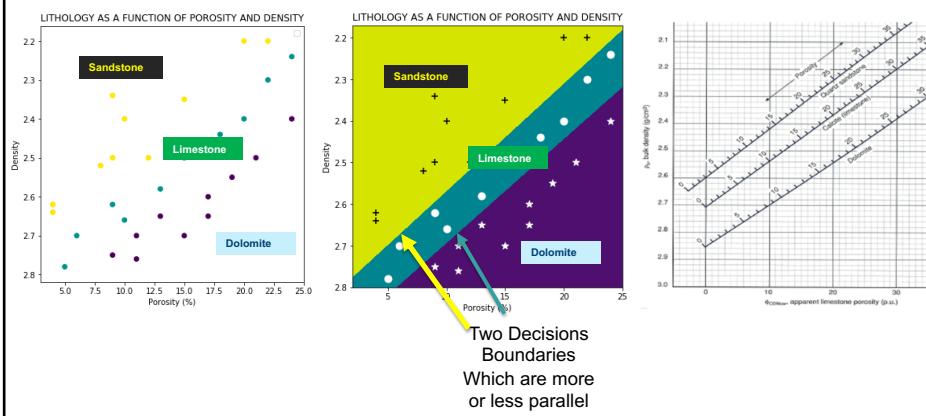
- For each of the three classes k , calculate the logistic function classifier for three binary classification problems:

$$\begin{aligned} h_{\theta_1}^1(x) &= P(y = 1|x, \theta_1) \\ h_{\theta_2}^2(x) &= P(y = 2|x, \theta_2) \\ h_{\theta_3}^3(x) &= P(y = 3|x, \theta_3) \end{aligned}$$

- Then pick as the predicted class k the one that maximizes the probability above

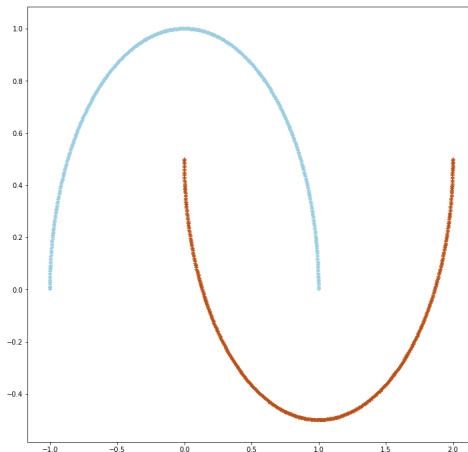
Imperial College
London

Example of MultiClass Logistic Regression Result



Imperial College
London

Simple Example of Non-Linear Logistic Regression(1)



The $m = 1000$ data points $(x_1^i, x_2^i)_{i=1,1000}$ are in the plane.
A group of data are one color, the other group another color.

Question: predict the color at each location of the plane.

Imperial College
London

Simple Example of Non-Linear Logistic Regression (2)

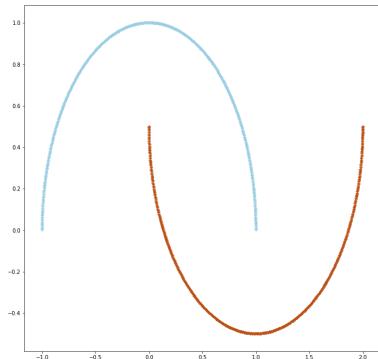
Model used for Logistic Regression:

$$\begin{aligned} h_\theta(x_1, x_2) \\ = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \dots + \theta_{16} x_1^5 + \theta_{17} x_2^5 \\ + \theta_{18} x_1^4 x_2 + \theta_{19} x_1 x_2^4 + \theta_{20} x_1^3 x_2^2 + \theta_{21} x_1^2 x_2^3) \end{aligned}$$

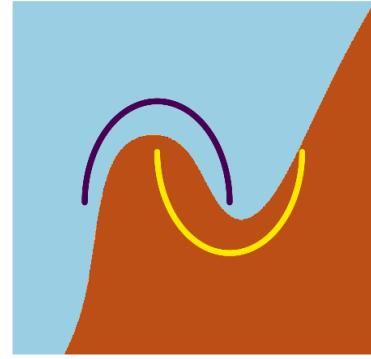
Decision Boundary will be a polynomial of degree 5.

Imperial College
London

Simple Example of Non-Linear Logistic Regression (3)



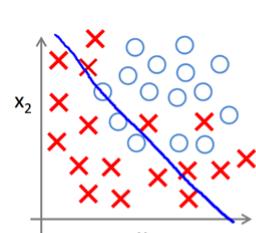
Input: 1000 Points of coordinates (x_1, x_2) taking two different colours



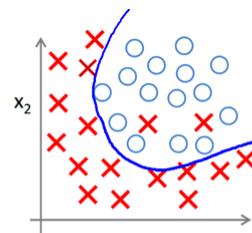
Output: Decision Boundary if a degree 5 polynomial in (x_1, x_2) is used

Imperial College
London

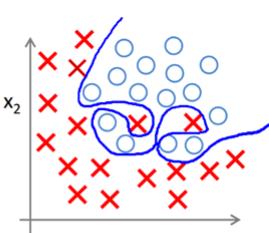
Configurations for Non-Linear Logistic Regression



$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2) \\ (g = \text{sigmoid function})$$



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



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

Will have to decide which polynomial degree is statistically reasonable and which one is too high and causes over-fitting (as seen with the third example).

Source: Machine Learning Course, Andrew Ng

Imperial College
London

Applying Logistic Regression to MNIST

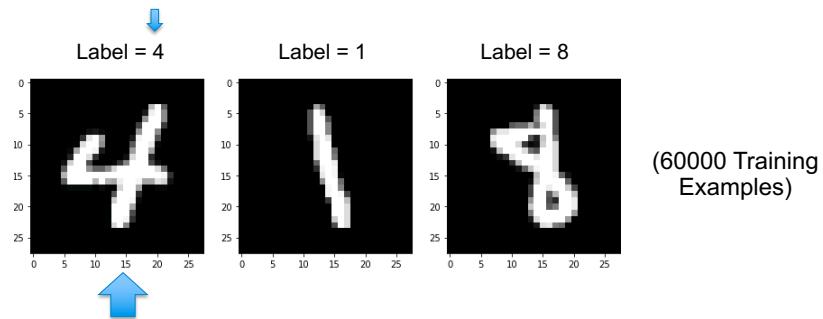
6	5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5	6	7	8	9	6	4	2	6	4	7	5	5
4	7	8	9	2	9	3	9	3	8	2	0	9	8	0	5	6	0	1	0
4	2	6	5	5	5	4	3	4	1	6	3	0	8	3	0	6	2	7	1
1	8	1	7	1	3	8	5	4	2	0	9	7	6	7	4	1	6	8	4
7	5	1	2	6	7	1	9	8	0	6	9	4	9	9	6	2	3	7	1
9	2	2	5	3	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3
4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	2	1	2	1	3
9	9	8	5	3	7	0	7	7	5	7	9	9	4	7	0	3	4	1	4
4	7	5	8	1	4	8	4	1	8	6	4	4	6	3	5	7	2	5	9

Extract from the MNIST Dataset

Imperial College
London

Input to Binary Logistic Regression on MNIST

Each label is the digit (between 0 and 9) associated with the image

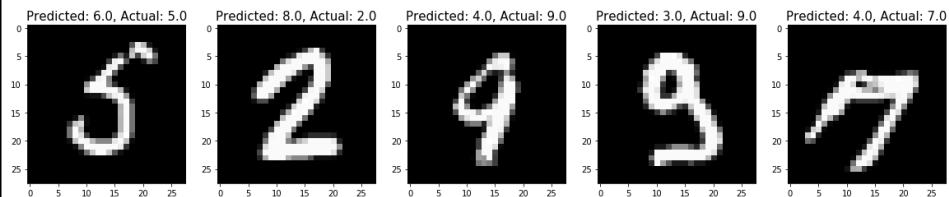


Each Image is coded as a 28x28 array of grey level pixels. Each pixel value varies between 0 and 255

Imperial College
London

Results of Logistic Regression on MNIST Dataset

Examples of Misclassified Images



Imperial College
London

Binary Logistic Regression on MNIST

The Results:

On the 60000 Training Images

Mean Accuracy: 0.93

Misclassified Images: 4329 (7.2%)

On the 10000 Test Images

Mean Accuracy: 0.91

Misclassified Images: 869 (8.7%)

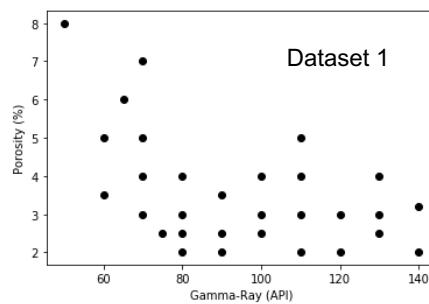
Imperial College
London

Introduction to Machine Learning

1. What is Machine Learning
2. Unsupervised vs Supervised Learning
3. Linear Regression
4. Logistic Regression
5. K-Means and PCA

Imperial College
London

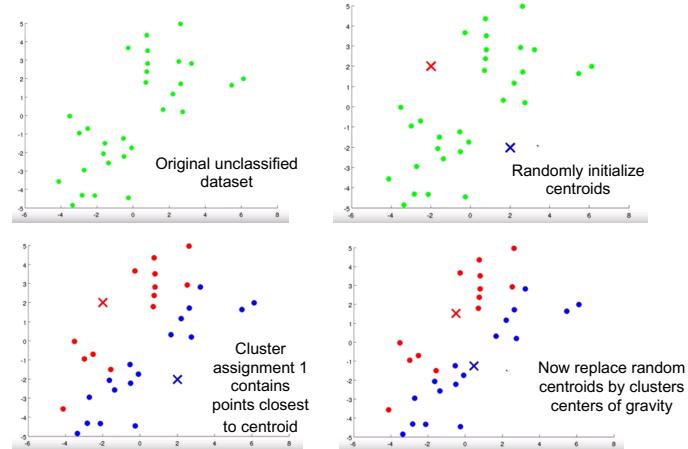
Clustering is an Unsupervised Learning Approach



The Training set has no labels $y^{(i)}$, it only has features $x^{(i)}$
Clustering automatically groups the input training examples into a small number of clusters of « similar » examples.

Imperial College
London

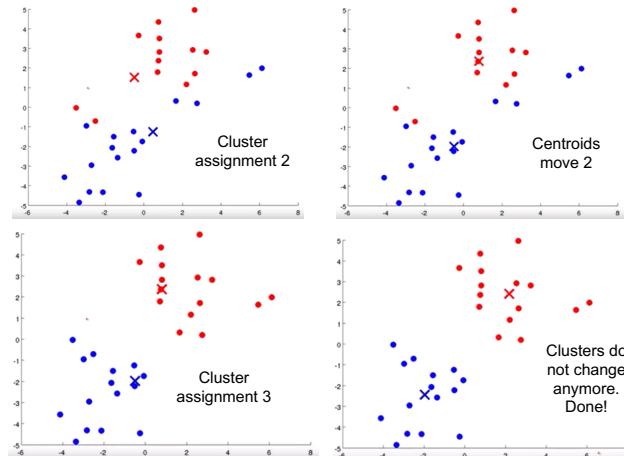
K-Means: The Algorithm (1)



Modified From Andrew Ng

Imperial College
London

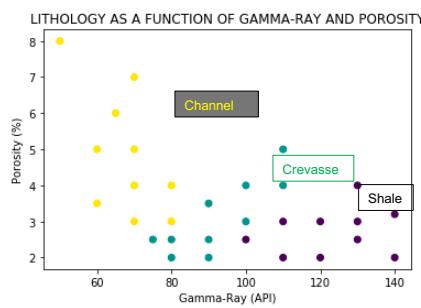
K-Means: The Algorithm (2)



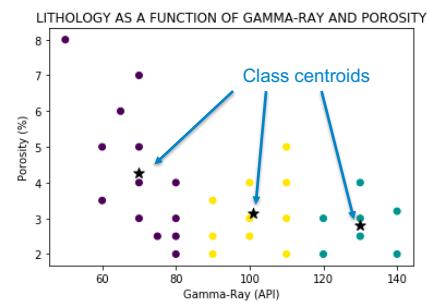
Imperial College
London

K-Means on Lithology Dataset 1

ACTUAL CLASSES



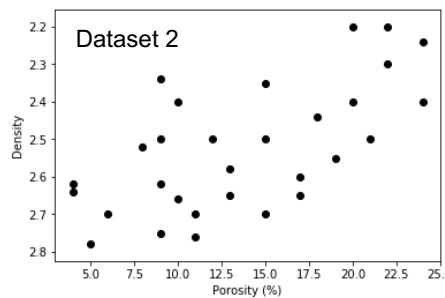
CLASSES FROM K-MEANS



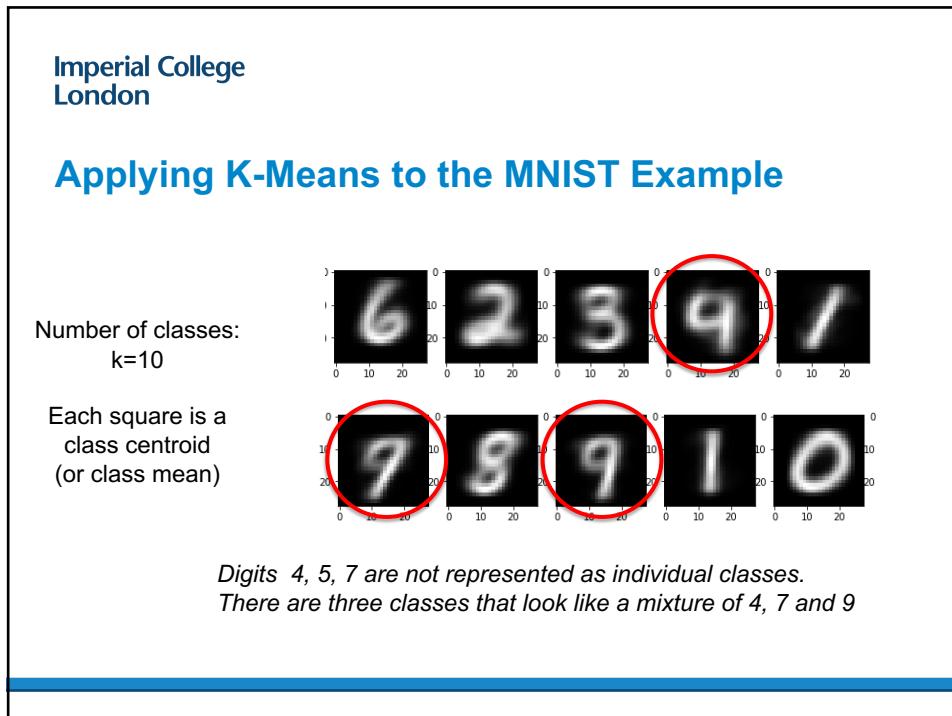
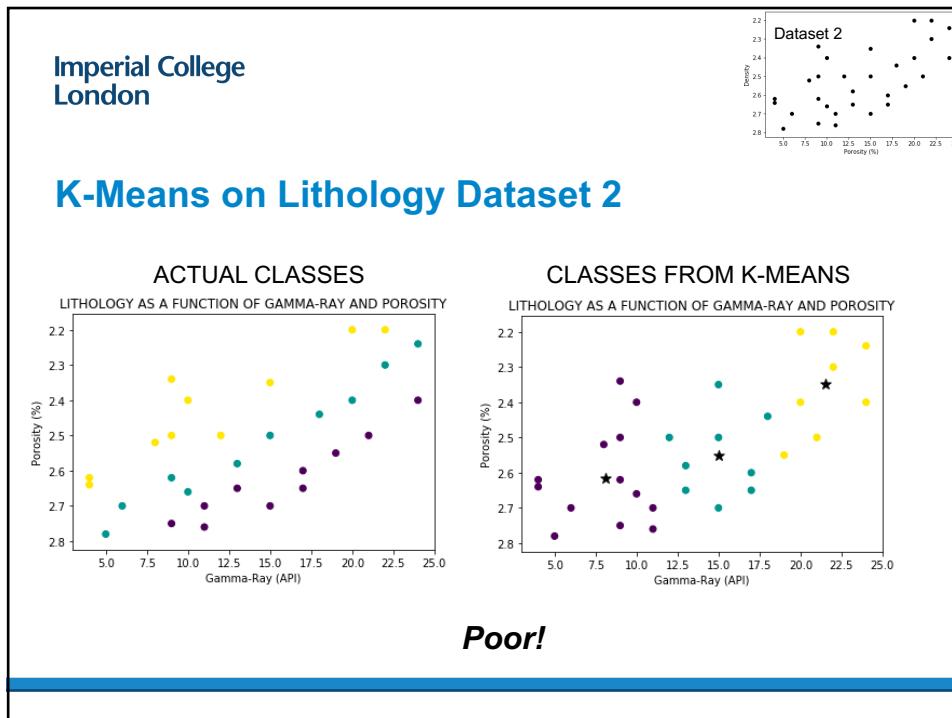
Good!

Imperial College
London

Clustering is an Unsupervised Learning Approach



The Training set has no labels $y^{(i)}$, it only has features $x^{(i)}$
Clustering automatically groups the input training examples into a small number of clusters of « similar » examples.

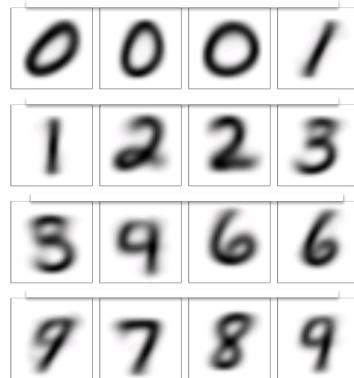


Imperial College
London

Applying K-Means with 16 Classes to MNIST

Number of classes:
 $k=16$

Each square is a
class centroid
(or class mean)

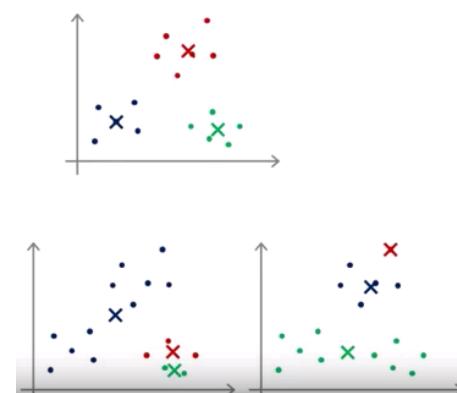


Source: <http://johnloeber.com/docs/kmeans.html>

Imperial College
London

Initial Centroids are Random, Results are non-unique!

Three examples of
possible clusters for
the same dataset



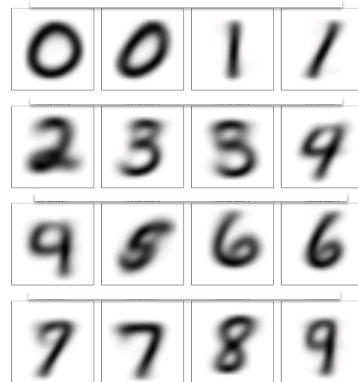
Solution: Try many and pick the best classification for your purpose!

Imperial College
London

Applying K-Means with 16 Classes to MNIST: Impact of Using Another Random Centroid

Number of classes:
 $k=16$

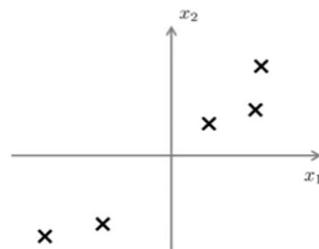
Each square is a
class centroid
(or class mean)



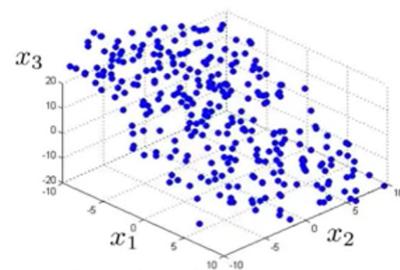
Source: <http://johnloeber.com/docs/kmeans.html>

Imperial College
London

Dimensionality Reduction: the PCA approach



Opportunity to Project 2-D Data
into 1-D Space



Opportunity to Project 3-D Data
into 2-D Space

Imperial College
London

The PCA Approach: Preliminary Data Normalization

Training Set vectors: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$. (no label)

Calculate mean of coordinates of the training vectors: $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$

Calculate standard deviation of coordinates of the training vectors: $s_j = \sqrt{\frac{1}{m} \sum_{i=1}^m x_j^{(i)2} - \mu_j^2}$

Replace each input vector by normalized value: $x_j^{(i)} := \frac{x_j^{(i)} - \mu_j}{s_j}$

Imperial College
London

Dimensionality Reduction: the PCA algorithm

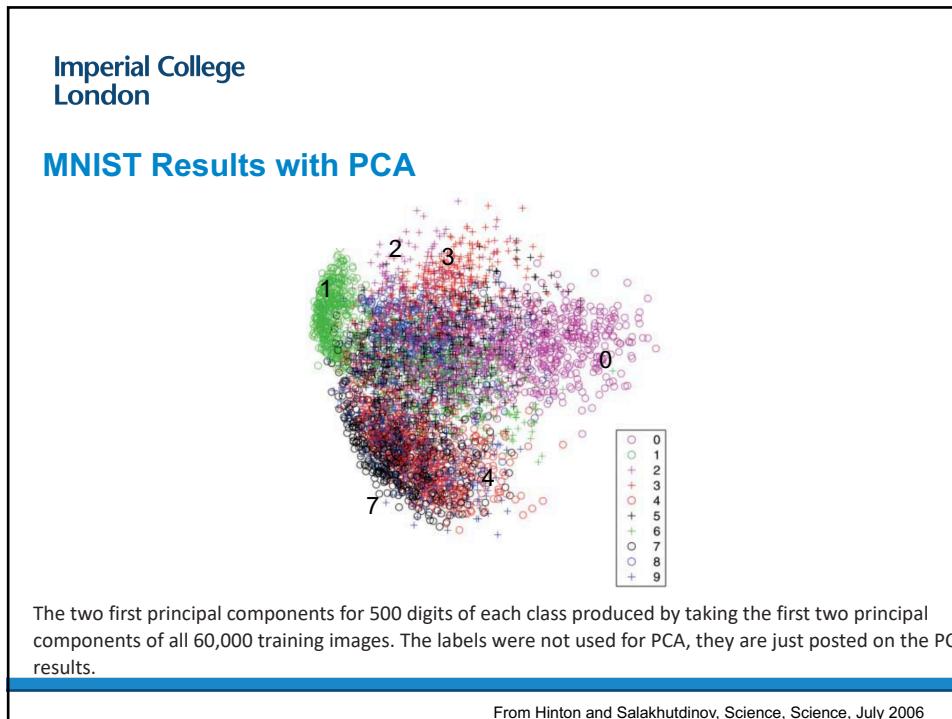
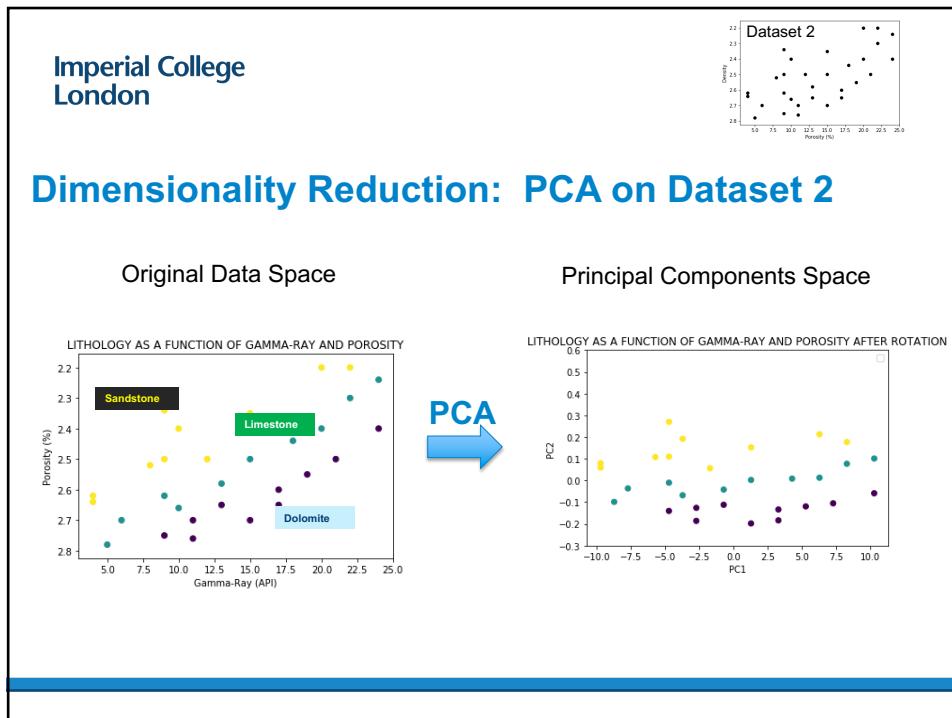
Training Set vectors: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$ (no label).

To project data from n-dimensional to k-dimensional space, calculate covariance matrix:

$$\Sigma = \frac{1}{m} \sum_{i=1}^m (x^{(i)}) (x^{(i)})^T$$

Then compute eigenvalues $(\lambda_i)_{i=1 \dots m}$ of matrix Σ

Keep the p largest eigenvalues $(\lambda_i)_{i=1 \dots p}$ and project on the space of dimension p defined by the associated p eigenvectors, also called principal components.



Imperial College
London

First Session Conclusion

- Supervised vs Unsupervised Learning
- Regression: The Elementary Machine Learning Approach
- Logistic Regression: The Elementary Supervised Classification Approach
- K-Means and PCA: The Elementary Unsupervised Classification Approaches
- Mathematical Notations are Important and will not Change during Module 8.

Neural Nets and Deep Learning are going to be a generalization of the above to much more complex (non-linear) approaches applied to huge datasets, possibly with hundreds of thousands of input features.