Imperial College London

Schedule for O.Dubrule's Lectures and Coursework

Day 1 (April 26th): Introduction to Machine Learning

Day 2 (April 28th): Feed-Forward Neural Networks

Day 3 (April 30th): Regularization, Bias, Variance, etc…

Day 4 (May 6th) : Convolutional Neural Networks

May 7th Morning: Coursework

1

1

Imperial College London

Each of the four days:

Morning: Topic Presentation

Afternoon: Exercises

2

2

Imperial College London

Interaction Rules

You are welcome, at any time, to:

- **Raise your hand and ask an oral question**
- **Use the chat**

3

3

Imperial College London

April 26th, 2021

Introduction to Machine Learning

Olivier Dubrule

4

Imperial College London

Objectives of the Day

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

5

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

6

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 medical image*
- ***Retail:*** *Offer personal recommendations based on recent choices of products*
- ***Marketing:*** *Organize customers into classes based on their past behavior*

But also…

- ***Automotive :*** *Interpret traffic images for automated vehicles*
- ***Media/Avertising:*** *Generate images of non-existing people for synthetic scenes*
- ***Art:*** *Generate synthetic music, text or paintings in the "style" of an existing artist*

What is common between all these examples?

Availability of large Datasets used to "Train" the Machine Learning algorithm!

7

Imperial College London

The Data Explosion behind Machine Learning

- In 2018, 33 Zettabytes (or 33×10^{21}) of data had been created worldwide (this is the estimated number of sand grains on earth)!
- This *Global Data Sphere* volume is multiplied by 2 every two years or by 1000 every 20 years.
- Storage in Data Centres will not be sustainable beyond 2040 (theoretically the 33 Zetabytes of data could be stored in 100 grams of DNA, a promising area of research).

8

8

Imperial College London

Machine Learning Definition

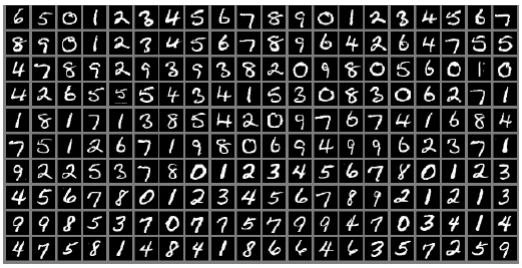
- *A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T, as measured by P, improves with experience E.*

(Toni Mitchell, 1997)

9

Imperial College London

Classical Machine Learning Example



Extract from the MNIST dataset

Machine Learning Problem:
*Based on a Training Set of 60,000 labelled digit images (**the experience E**), learn an algorithm that, given a new pixellized input image, automatically recognizes which digit it represents (**the Task T**). **Performance P** quantifies the accuracy of the algorithm on a Test Set of 10,000 images.*

MNIST: Modified National Institute of Standards and Technology database

10

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

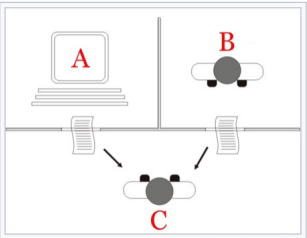
Turing, A.M. (1950). Computing machinery and intelligence. Mind, 59, 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The



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)

11

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**

12

Imperial College London

Supervised vs Unsupervised Learning

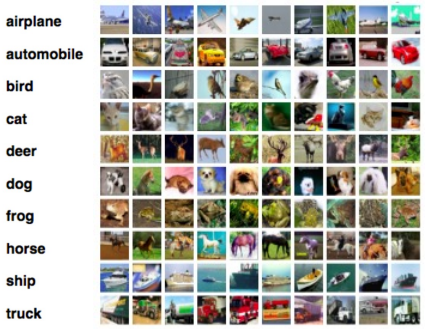
- **Supervised Learning:** the task T is clearly defined because the data are labeled and the computer has to predict the labels of new data. Performance P is quantified by the mismatch between predicted and actual labels of new data.
- **Unsupervised Learning:** the computer is given some unlabeled data and the task T is to organize them based on the recognition of similar patterns in some groups of data. The numbers speak for themselves, as there is no preliminary human intervention to produce the labels. The quantification of the Performance P is not as straightforward as for Supervised Learning.

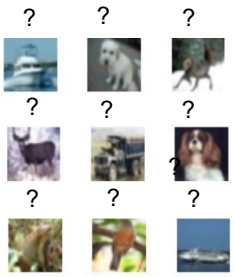
13

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.***





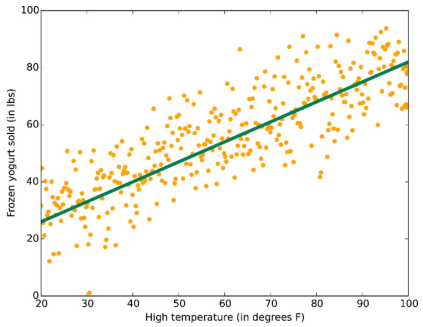
Excerpt of the CIFAR-10 dataset, Source: https://www.cs.toronto.edu/~kriz/cifar.html

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

14

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).**

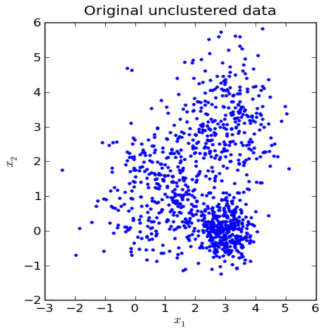
https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice

15

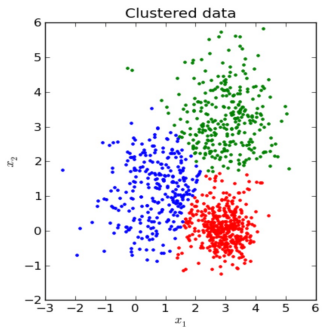
Imperial College London

Unsupervised Learning: Clustering

Data : x only, no label y



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



Let the data speak for themselves!

16

Imperial College London

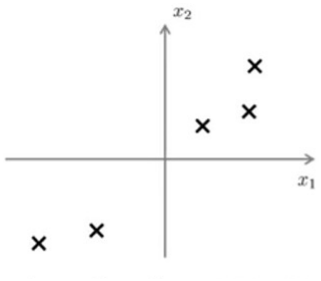
Clustering Examples

- Sort costumers 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

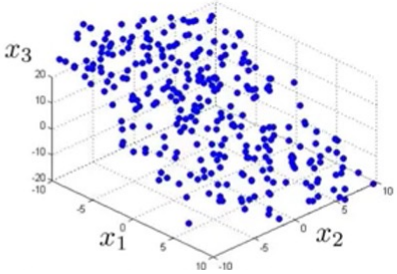
17

Imperial College London

Unsupervised Learning: the PCA approach



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

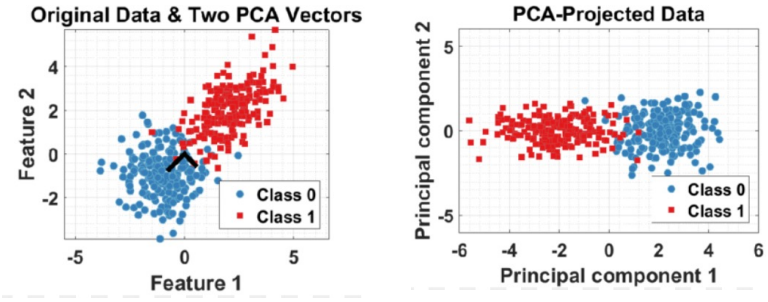


Opportunity to Project 3-D Data into 2-D space

18

Imperial College London

Unsupervised Learning with PCA (1)



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

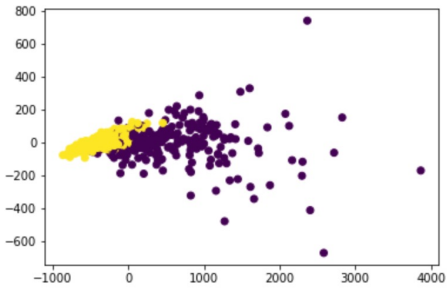
"Unsupervised" because the labels are not used in the PCA calculations

19

Imperial College London

Unsupervised Learning with PCA (2)

Breast Cancer Data (original dataset is in 30 dimensions)



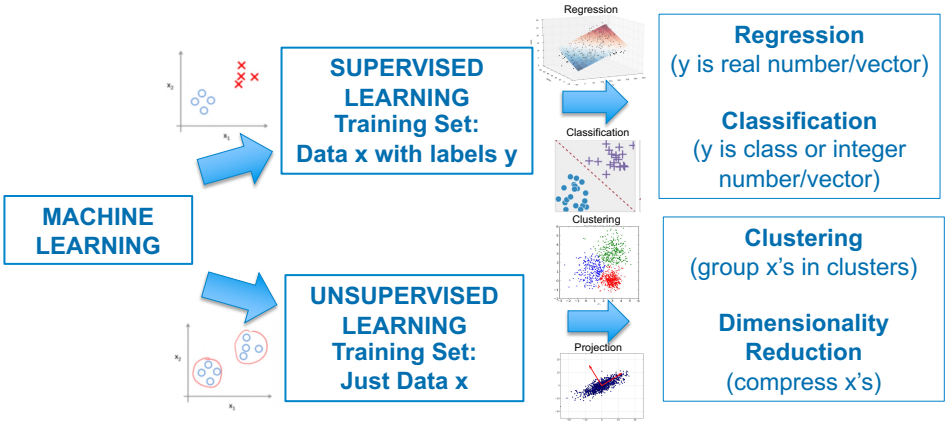
Task: identify lower dimension spaces where data can be easily classified

Dimensionality Reduction often used as a First Step before Clustering!

20

Imperial College London

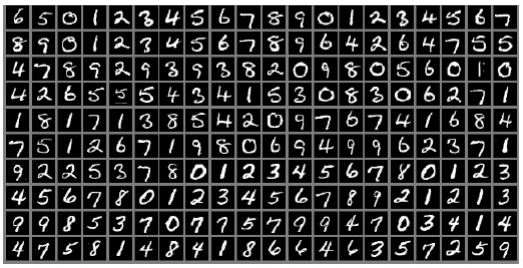
Supervised vs Unsupervised Learning



21

Imperial College London

When data are labelled, the same problem can be addressed as a Supervised or an Unsupervised one



Extract from the MNIST dataset

Supervised approach: each image is labelled by the digit it represents

Unsupervised approach: ignore the labels and try to sort all the images into classes

22

Imperial College London

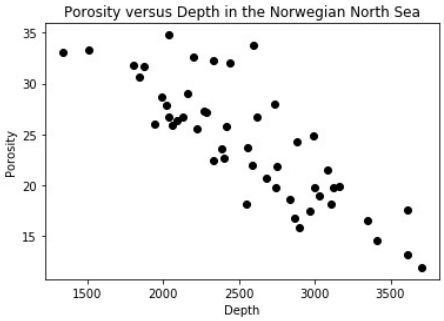
Introduction to Machine Learning

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

23

Imperial College London

Simple Regression: Standard ML Terminology & Notations



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

24

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** » $\boldsymbol{h_\theta(x)}$

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

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 **Loss Function** $\boldsymbol{J(\theta_0,\theta_1)}$ (also called **Cost Function**).

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

25

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 loss function $J(\theta_0,\theta_1)$ becomes

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

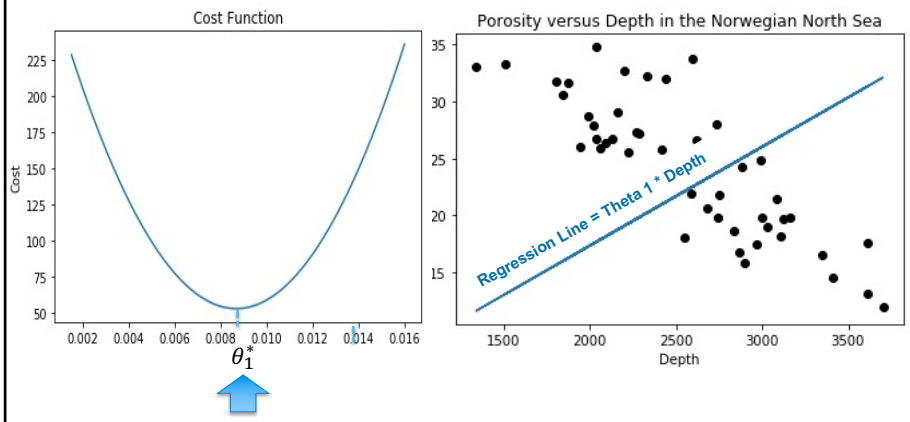
The loss function is simply a **second degree polynomial in $\boldsymbol{\theta_1}$**, minimum for

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

26

Imperial College London

Loss Function is Convex for Regression



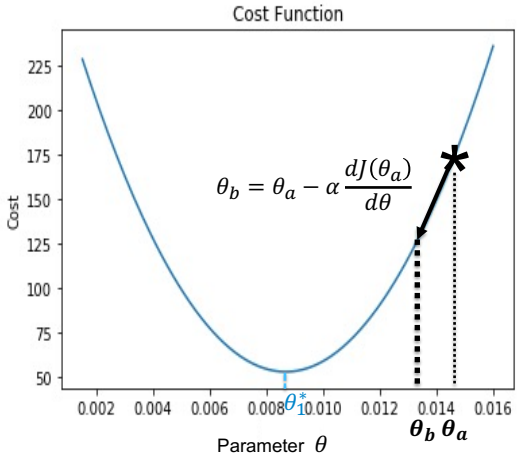
Loss Function Minimum at θ_1^* = 0.00876

27

Imperial College London

Gradient Descent on a Convex Function

*α is the **learning rate**, fixed by the user.*



If α is too small , convergence can be too slow, if α is too large it may miss the minimum.

28

Imperial College London

Result with a Bias Term in the Linear Regression

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

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

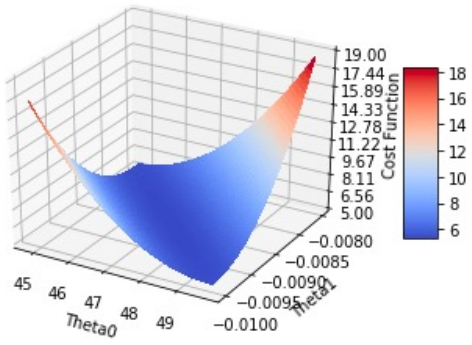
The cost function is simply a second degree polynomial in θ_0 and θ_1. It reaches its minimum for the values of $\theta_0$and θ_1 given by the normal equations

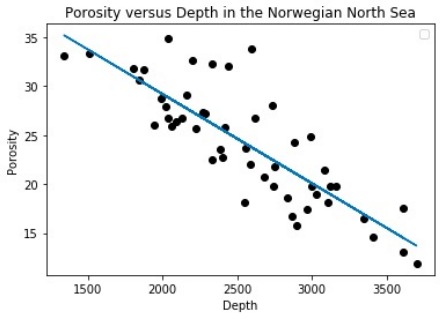
$$\theta_0^* = \frac{\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)2}\right)\left(\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\right) - \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right)\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}y^{(i)}\right)}{\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)2}\right) - \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right)^2}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

29

Imperial College London

Cost Function for θ_0 and θ_1





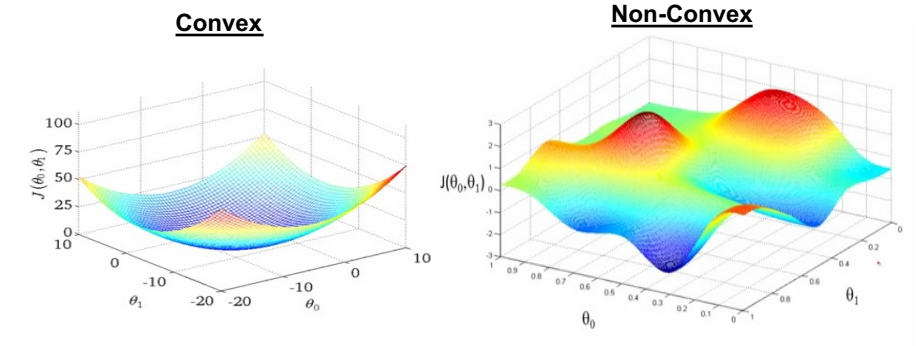
A nice property of a convex cost function is that it has no local minima. If you cannot decrease the loss function anymore, it means you have reached the global minimum.

30

Imperial College London

Convex vs Non-Convex Objective Function

Convex **Non-Convex**



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

31

Imperial College London

To see examples of Loss Landscapes

https://losslandscape.com

32

32

Imperial College London

Increasing the input dimension: 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

33

Imperial College London

Examples of Multiple Linear Regression

Predict House Price from its squared footage, number of bedrooms, number of bathrooms, …

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

34

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

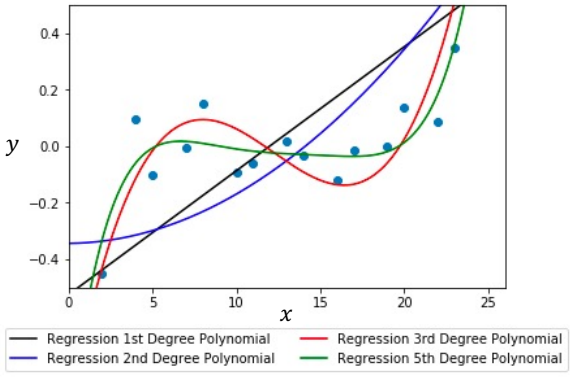
We write the m data vectors as: $(x^{(i)}, y^{(i)})$ $x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ \dots \\ x_n^{(i)} \end{pmatrix}$ and $y^{(i)}$ real for $i = 1 \dots m$

35

Imperial College London

Example of Non-Linear Regression

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



36

Imperial College London

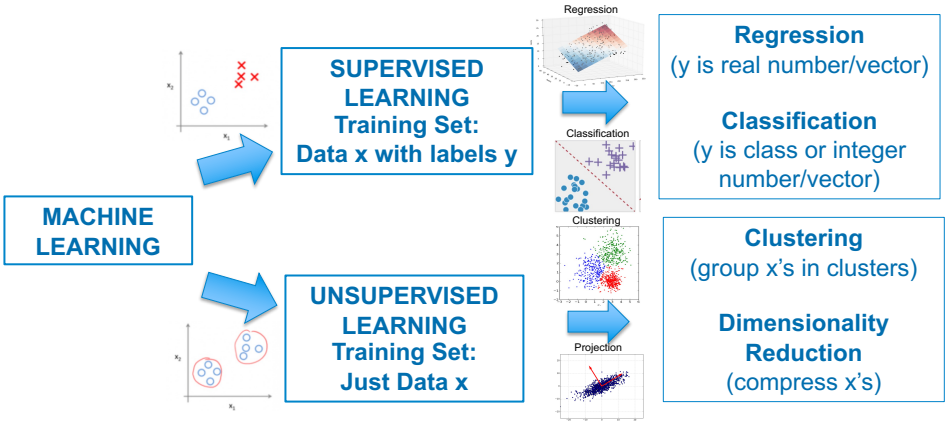
Introduction to Machine Learning

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

37

Imperial College London

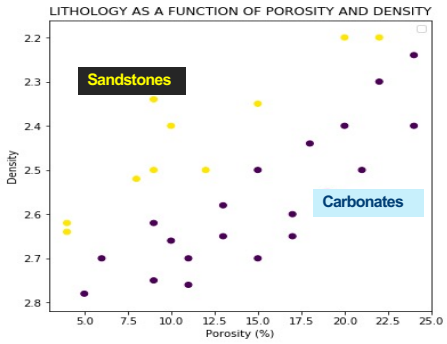
Reminder: Supervised vs Unsupervised Learning



38

Imperial College London

Logistic Regression Problem



30 rock samples with Porosity, Density and Lithology as a label.

We wish to predict Lithology from Density and Porosity: this is a Supervised Classification problem .

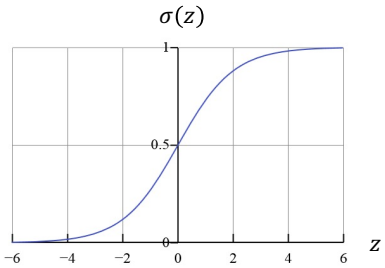
Cannot use Linear Regression : need a transform from the domain of real values to the 0 or 1 indicator

39

Imperial College London

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

- The **Sigmoid** function $\sigma(z)$ is also called the **Logistic** Function
- It transforms a real value z into a value between 0 and 1



If $z \geq 0$ then $\sigma(z) \geq 0.5$

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

A useful equation is:

$$\sigma'(z) = \big(1-\sigma(z)\big)\sigma(z)$$

40

Imperial College London

Interpreting the output of the Logistic function

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

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

- In vector form:

$$y = \theta^T x$$

- y is a real number which can take any positive or negative value.
- If we apply the sigmoid (or logistic) function $\sigma(y)$ we obtain a value between 0 and 1, which **we interpret as the probability for the class to be 1**

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

41

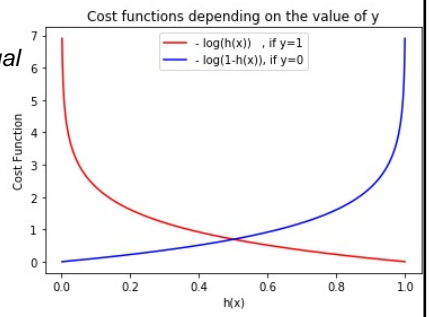
Imperial College London

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 $h_\theta(x)$?

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

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



Combining the two possibilities above into one single equation:

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

42

Imperial College London

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

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

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

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

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)}$$ *Exercise: prove this relationship!*

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

43

Imperial College London

Logistic Regression Output once Trained

- Once Logistic Regression has been trained on Training Set, for any input vector x we now know the function:

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

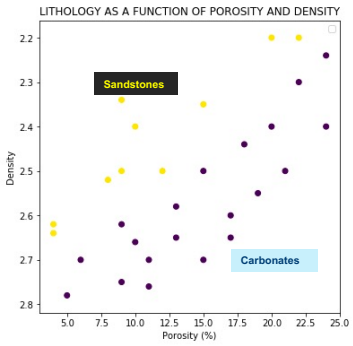
- $h_\theta(x)$is interpreted as a probability between 0 and 1
- Hence the predicted class for a new feature vector x will be:

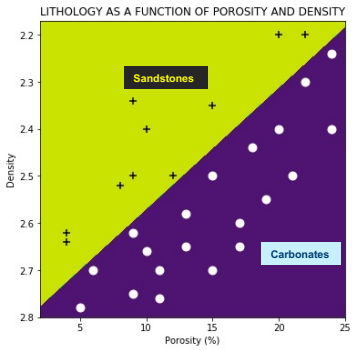
$$0\ if\ h_\theta(x) \leq 0.5$$
$$1\ if\ h_\theta(x) \geq 0.5$$

44

Imperial College London

Example of Binary Logistic Regression Result

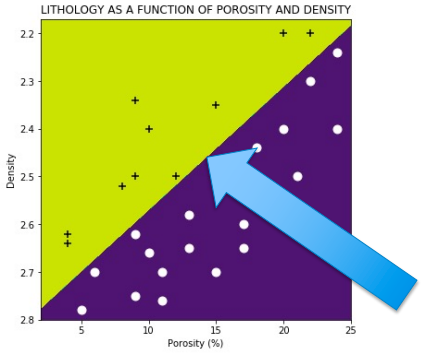




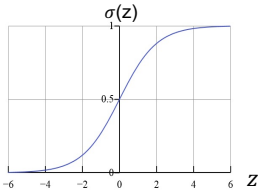
45

Imperial College London

Definition of the Decision Boundary



$$P(y = 1|\theta, x) = \sigma(\theta^T x)$$



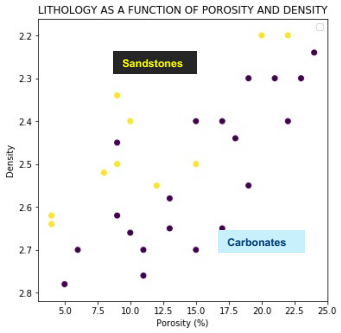
Sigmoid function $\sigma(z)$ is ≥ 0.5 if $\theta^T x \geq 0$

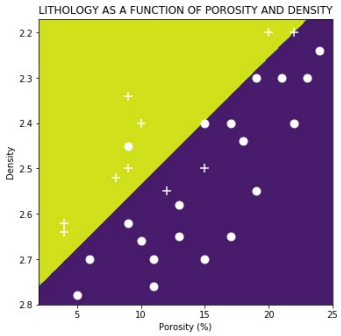
Hence the Decision Boundary in 2D is the line of equation $\theta^T x = 0$

46

Imperial College London

Example where the Data are not Linearly Separated

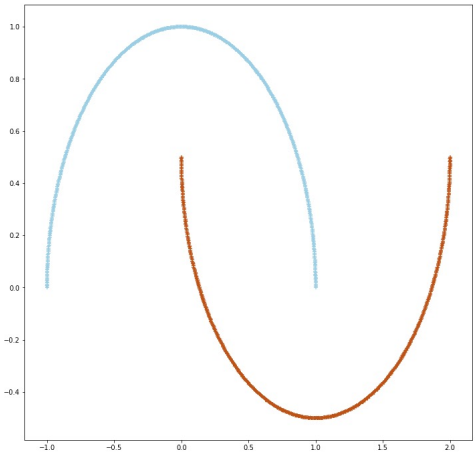




47

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.

(can also be seen as an interpolation problem in the plane)

48

Imperial College London

Simple Example of Non-Linear Logistic Regression (2)

Model used for Logistic Regression:

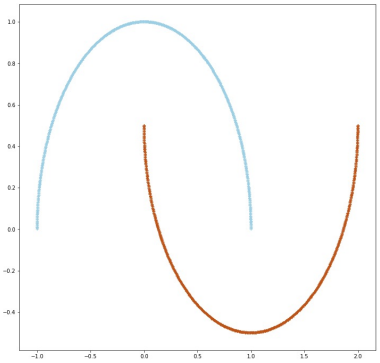
$$h_\theta(x_1, x_2) = \sigma\left(\begin{matrix}\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \ldots. + \theta_{16} x_1^5 + \theta_{17} x_2^5 + \\ \theta_{18} x_1^4 x_2 + \theta_{19} x_1 x_2^4 + \theta_{20} x_1^3 x_2^2 + \theta_{21} x_1^2 x_2^3\end{matrix}\right)$$

Decision Boundary will be a polynomial of degree 5.

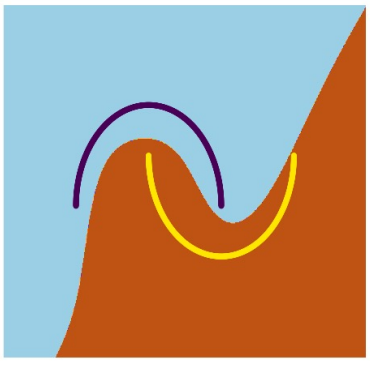
49

Imperial College London

Simple Example of Non-Linear Logistic Regression (3)



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

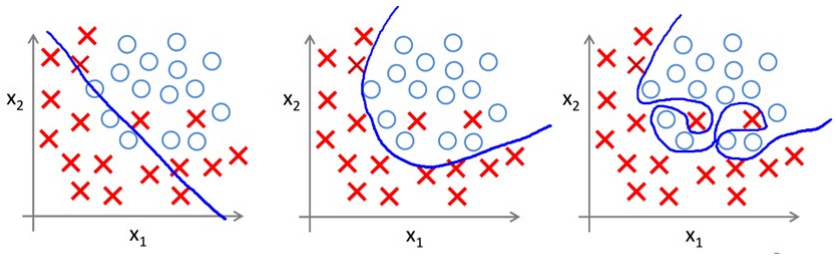


Output: Decision Boundary if a degree 5 polynomial in (x_1, x_2) is used

50

Imperial College London

Configurations for Non-Linear Logistic Regression



First Degree Polynomial | Second Degree Polynomial | n^{th} Degree Polynomial

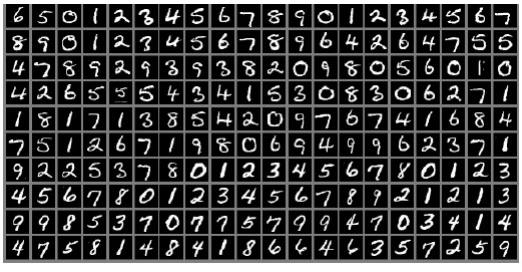
Will have to decide which polynomial degree is statistically reasonable and which one is too high and causes "overfitting" (as seen with the third example).

Source: Machne Learning Course, Andrew Ng

51

Imperial College London

Applying Logistic Regression to MNIST



Extract from the MNIST Dataset

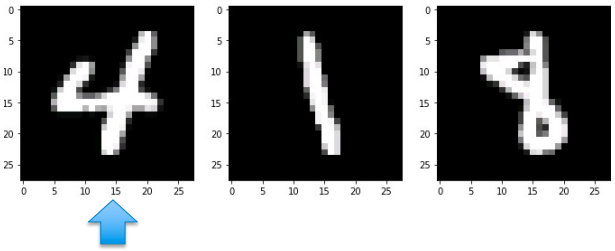
52

Imperial College London

Input to Logistic Regression on MNIST

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

Label = 4 Label = 1 Label = 8



(60000 Training Examples)

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

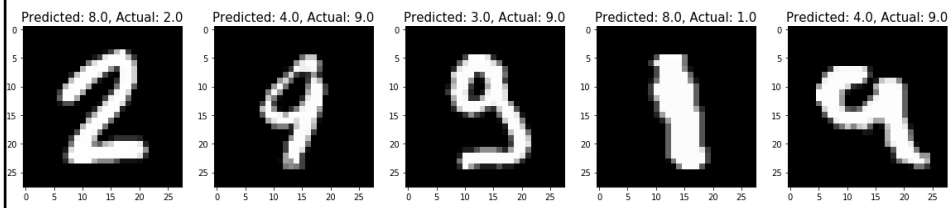
53

Imperial College London

Results of Softmax Regression on MNIST Dataset

(Softmax, a generalization of Logistic Regression >2 classes, to be discussed in next session)

Examples of Misclassified Images



https://www.kdnuggets.com/2016/07/softmax-regression-related-logistic-regression.html

54

Imperial College London

Softmax Regression on MNIST

The Results:

On the 60000 Training Images	On the 10000 Test Images
Mean Accuracy: 0.934	Mean Accuracy: 0.918
Misclassified Images: 3939 (6.57%)	Misclassified Images: 818 (8.18%)

55

Imperial College London

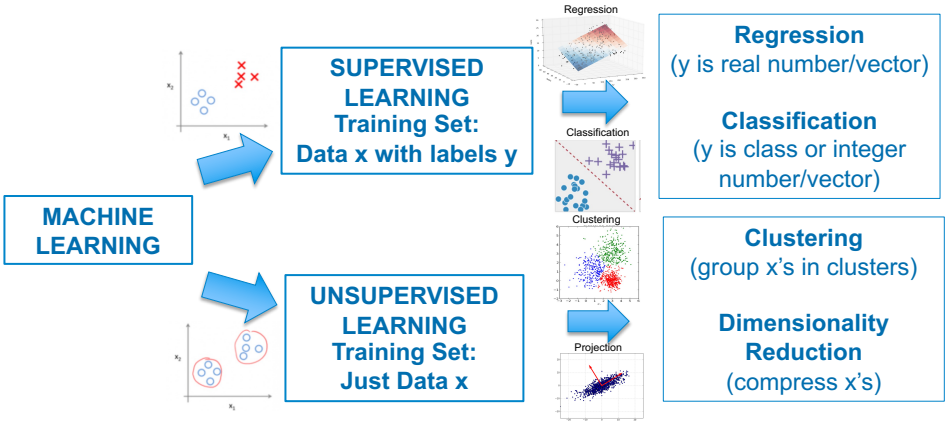
Introduction to Machine Learning

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

56

Imperial College London

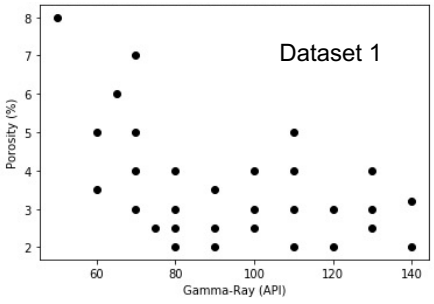
Reminder: Supervised vs Unsupervised Learning



57

Imperial College London

Clustering is an Unsupervised Learning Approach

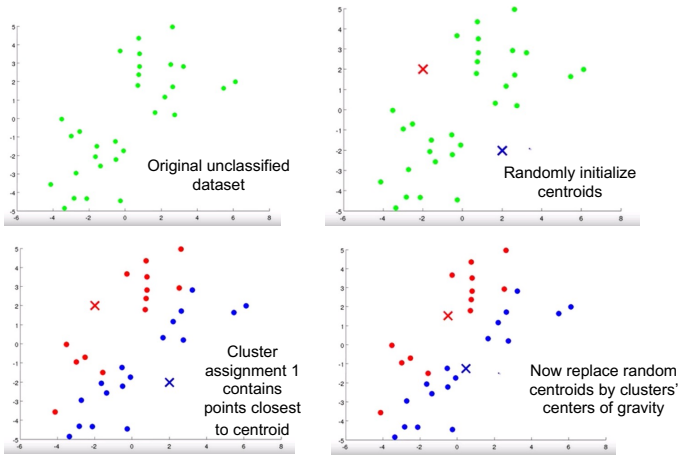


Here the Training Set has no labels $y^{(i)}$, it only has two features $x^{(i)}$, gamma-ray and porosity. Clustering automatically groups the input training examples into a small number of clusters of « similar » examples.

58

Imperial College London

K-Means: The Algorithm (1)

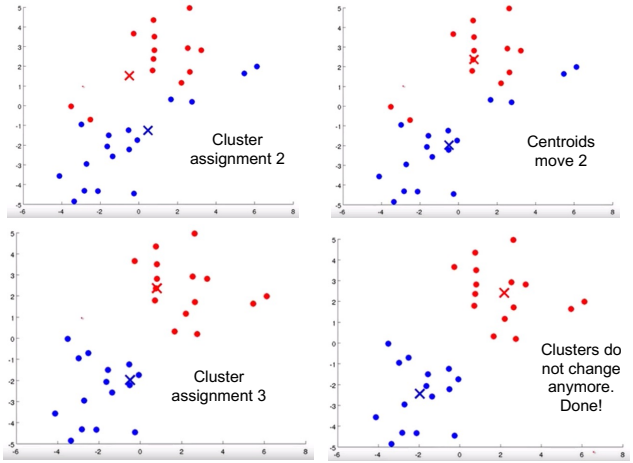


Modified From Andrew Ng

59

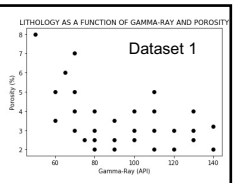
Imperial College London

K-Means: The Algorithm (2)



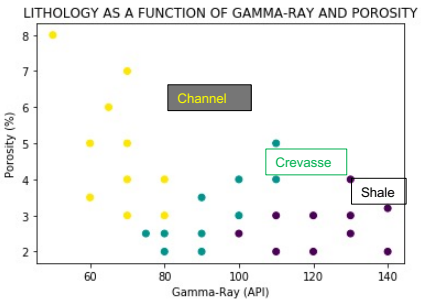
60

Imperial College London

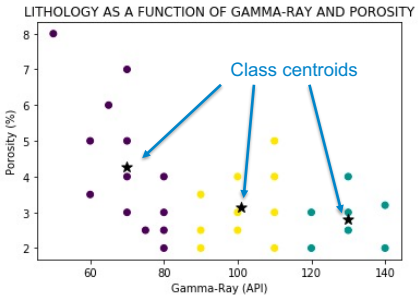


K-Means on Lithology Dataset 1

ACTUAL CLASSES



CLASSES FROM K-MEANS

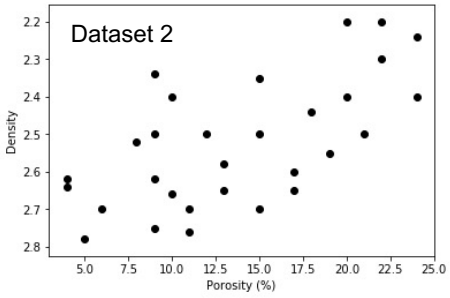


Not bad!

61

Imperial College London

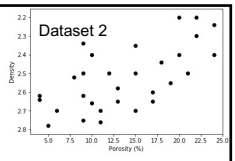
Clustering is an Unsupervised Learning Approach



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

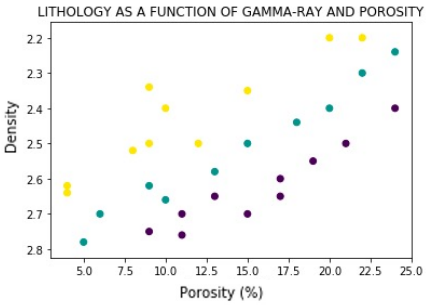
62

Imperial College London

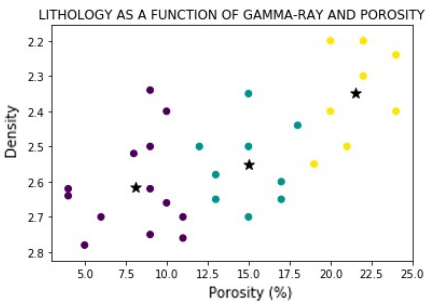


K-Means on Lithology Dataset 2

ACTUAL CLASSES



CLASSES FROM K-MEANS



Poor!

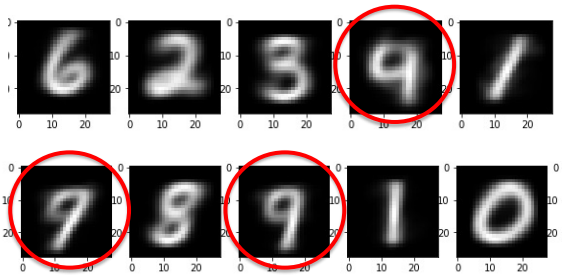
63

Imperial College London

Applying K-Means to the MNIST Example

Number of classes: k=10

Each square is a class centroid (or class mean)



Digits 4, 5, 7 are not represented as individual classes.
There are three classes that look like a mixture of 4, 7 and 9

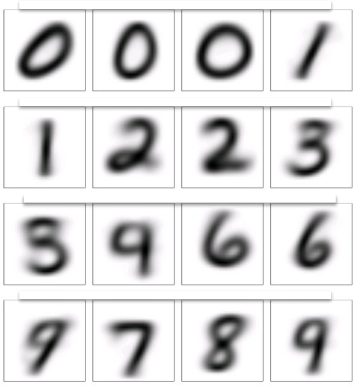
64

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)



Still some confusion about classes 4 and 5, and some multiple classes for the same digit (0, 1, 2, 6, 9)

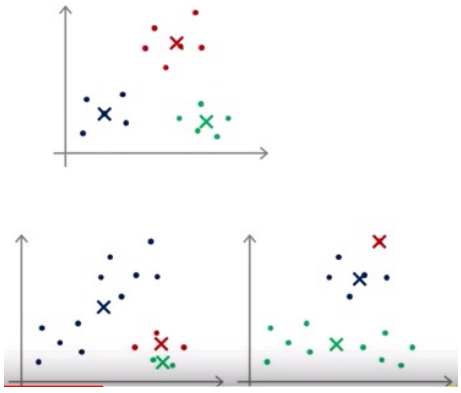
Source: http://johnloeber.com/docs/kmeans.html

65

Imperial College London

Initial Centroids are Random, Results are non-unique!

Three examples of possible clusters for the same dataset, one corresponds to global minimum, the two others to local minima.



66

See Andrew Ng's lessons 13.3 and 13.4 on YouTube

66

Imperial College London

But what is the The k-Means Implicit Loss Function?

The k-Means Loss Function is, for k classes and m data points (with $k \ll m$):

$$J\left(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots., \mu_k\right) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2$$

- $c^{(1)}, \dots, c^{(m)}$: Classes associated with each data point
- $\mu_1, \dots., \mu_k$: Class centroids
- $x^{(i)}$: Data Points
- $\mu_{c^{(i)}}$: Centroid of $x^{(i)}$'s class

Best approach: run say 100 k-Means with different random initializations of centroid, calculate Loss Function J for each of them, and pick run associated with lowest value of J.

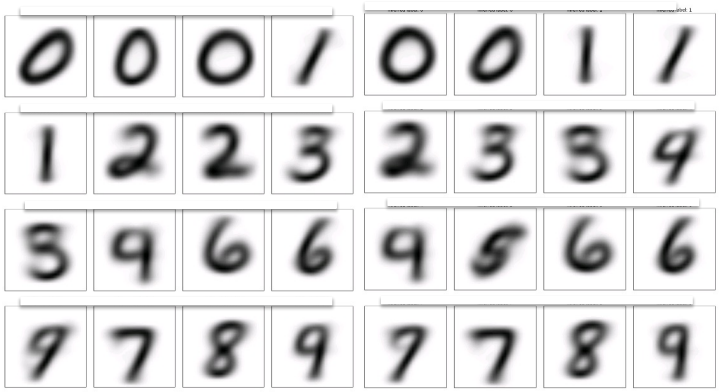
See Andrew Ng's lessons 13.3 and 13.4 on YouTube

Imperial College London

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

Number of classes: k=16

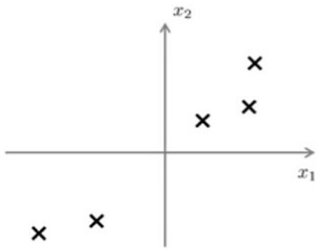
Each square is a class centroid (or class mean)



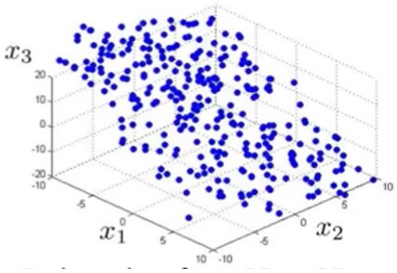
Source: http://johnloeber.com/docs/kmeans.html

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

69

Imperial College London

The PCA Approach: Preliminary Data Normalization

Training Set vectors: $x^{(1)}, x^{(2)}, \ldots\ldots, x^{(m)}$. (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 s_j of coordinates of the training vectors: $s_j^2 = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)2} - \mu_j^2$

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

70

Imperial College London

Dimensionality Reduction: the PCA algorithm

Training Set vectors: $x^{(1)}, x^{(2)}, \ldots\ldots, x^{(m)}$ (no label), each of dimension n

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

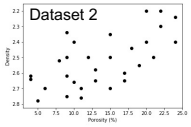
$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}\right)\left(x^{(i)}\right)^{T}$$

Σ is of dimension n

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

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

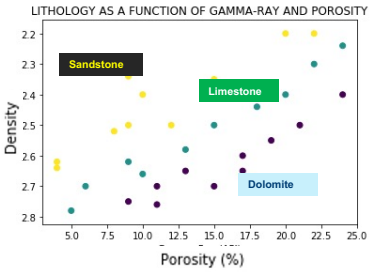
Imperial College London



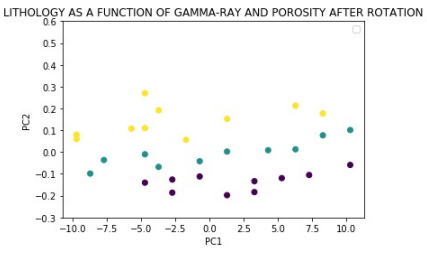
Dimensionality Reduction: PCA on Dataset 2

Original Data Space

Principal Components Space

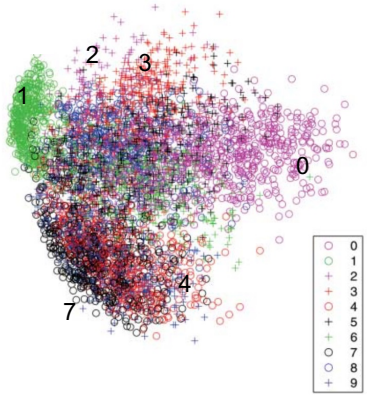


PCA



Imperial College London

MNIST Results with PCA



The two first principal components for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. The labels were not used for PCA, they are just posted on the PCA results.

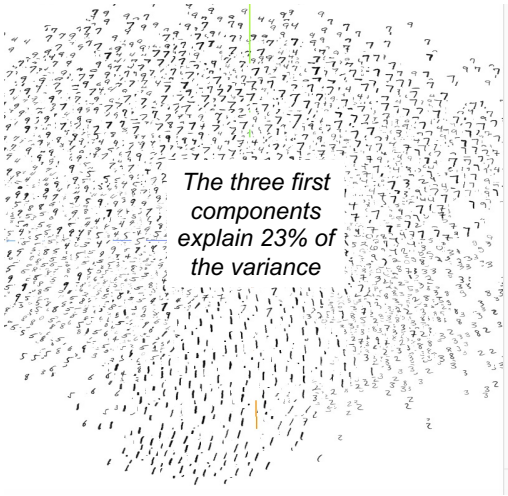
From Hinton and Salakhutdinov, Science, Science, July 2006

73

Imperial College London

3-D PCA on MNIST

https://projector.tensorflow.org/



74

74

Imperial College London

Mathematics of the PCA algorithm

PCA transforms the data to a new coordinate system such that the greatest variance after projection of the data lies on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on….

If X is the $m \times n$ data matrix (m data points, n features), each row vector $(x_{ij})_{j=1,\dots,n}$ of dimension n is mapped into a new vector $(t_{il})_{l=1,\dots,p}$ of dimension p which is a linear combination of the n coordinates of $(x_{ij})_{j=1,\dots,n}$:

$$t_{il} = \sum_{j=1}^{n} x_{ij} w_{jl}$$

We want the vector $(t_{i1})_{i=1,\dots,n}$ to maximize its norm, or the first weight vector to satisfy:

$$w_1 = argmax\left(\sum_{i=1}^{m} t_{i1}^2\right) = argmax(w_1^T X^T X w_1)$$

Rayleigh Theorem says that the maximum value of the above norm is the largest eigenvalue of $X^T X$, which occurs when w is the corresponding eigenvector.

https://en.wikipedia.org/wiki/Principal_component_analysis

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.**

Neural Nets and Deep Learning are going to be a generalization of the above to more complex (non-linear) approaches applied to huge datasets.

Imperial College London

Exercise 1: Calculate Logistic Regression "by Hand":

We have four points $x^{(i)}$ in the plane, each with a label $y^{(i)}$ equal to 0 or 1. The coordinates of the four points and their labels are as follows:

$x^{(1)} = (2, 4),\ y^{(1)} = 1$
$x^{(2)} = (1, 3),\ y^{(2)} = 1$
$x^{(3)} = (4, 2),\ y^{(3)} = 0$
$x^{(4)} = (2, 2),\ y^{(4)} = 0$

Exercise 2: Compare Regression and PCA in 2-D with associated Python code.