Imperial College London

15 January 2024

The Magic of Machine Learning

Imperial College London

What is this course about?

1. Getting to know some (but not all) popular machine learning methods.

2. Understanding the mathematics of how they work.

3. Being able to apply machine learning algorithms in a lab setting to a range of problems.

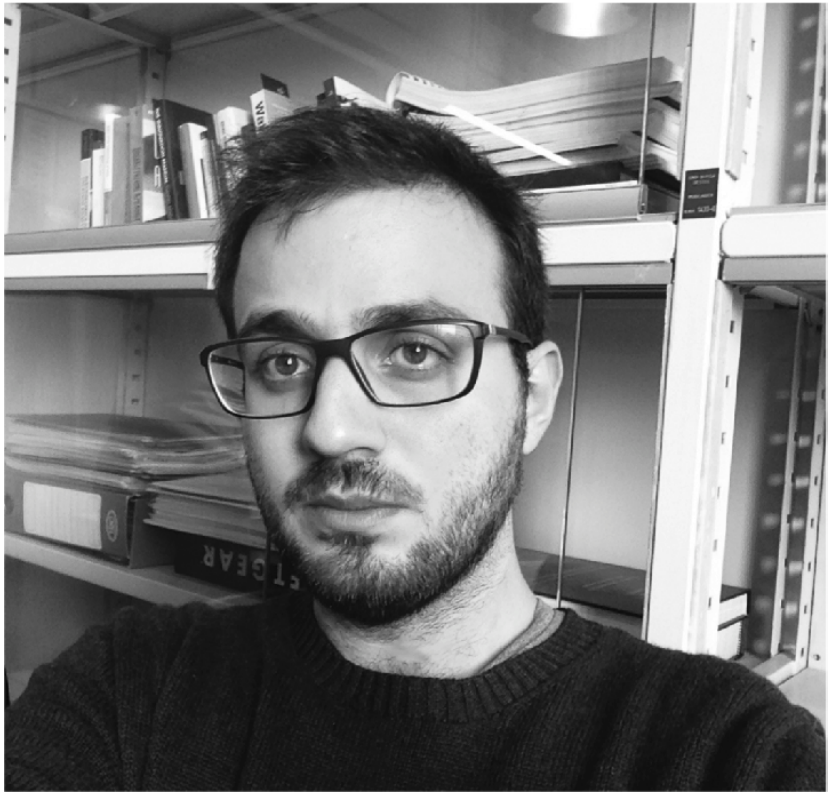
4. Being able to explain the operation and structure of a range of machine learning techniques

Imperial College London

Who are we?



Robert Shorten
r.shorten@imperial.ac.uk



Pietro Ferraro
p.ferraro@imperial.ac.uk

Imperial College London

Who are we?

- Aida Manzano Kharmen
- Barry Mulvey
- Abigail Langbridge
- Anthony Quinn
- Homayoun Rafati

Imperial College London

Logistics of course

1. **Lectures** on Tuesday 11am-1pm.[1]
- delivered by Bob, Pietro and guests.

2. **Tutorial and lab clinics** on Wednesday 9-11am.[2].
- delivered *online* to groups of two by Pietro, Bob, Barry, Aida, Abigail, Anthony, Homoqoun

3. **Draft lectures** available on Blackboard in advance of lectures.
- please check for updates after lectures.

More on lectures and labs in a few mins.

---

[1] Lecture content designed by Bob, Pietro.

[2] Lab content designed Pietro

Imperial College London

15 January 2024

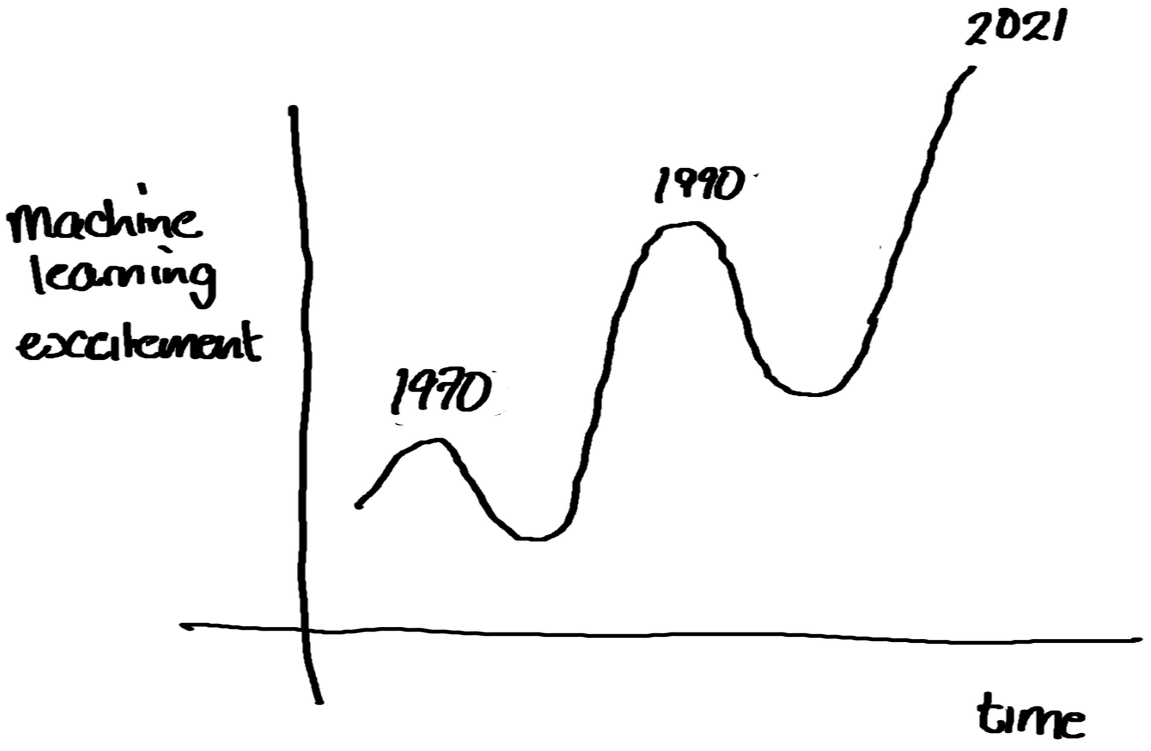
1. Machine learning's dirty little secrets

Secret 1: Machines don't learn?

- What is machine learning?
- ML produces algorithms that produce outputs given an input belonging to a class of data points known as the *training data*.
- The algorithm also gives "correct" outputs for other inputs provided that the belong to the same class as the training data. The machine is not really learning. It's more that the algorithm is produced that is able to generalise in some sense.
- Why is it called ML? - Blame IBM (Authur Samuel to be precise).

Imperial College London

Secret 2: Machine learning excitement



Secret 2: Neural NOTWORKS

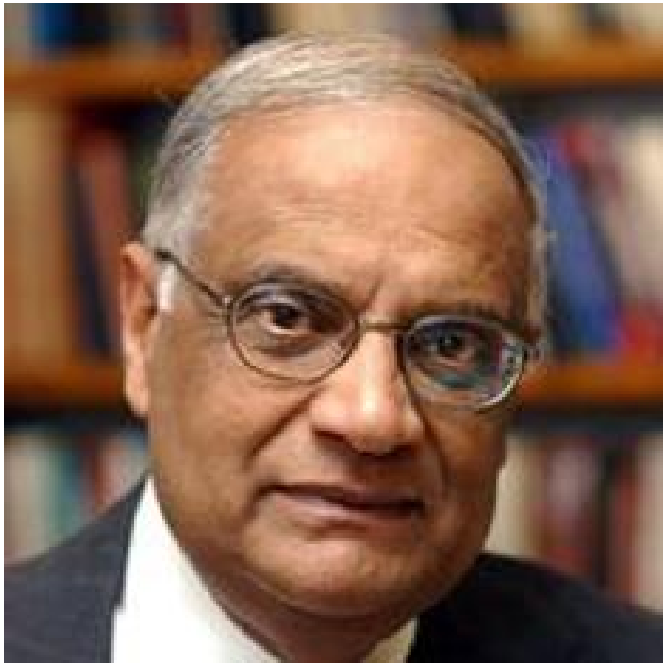


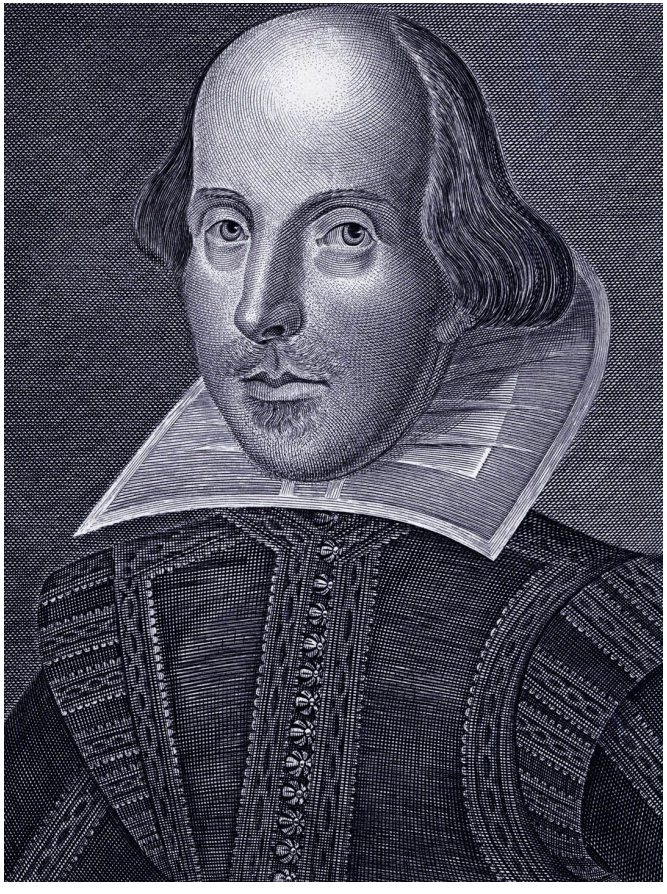
Figure 1: Professor K. S. Narendra

ML (Neural networks) was, until very recently, a dirty word in some areas of science.

- Classification problems
- Regression problems
- Dynamic systems

Imperial College London

Secret 2: Neural NOTWORKS



Classification problems

- Works of Goethe.
- Works of Shakesphere

- Can a machine distinguish between the two?

Imperial College London

Secret 2: Neural NOTWORKS



Regression problems

- Given historical GDP data
- Give previous external shocks

- Can we predict effect of COVID on economy?

Imperial College London

Secret 2: Neural NOTWORKS



Dynamics and control

- Really complicated physical systems
- Robots, Autonomous vehicles,...

- Can we make them behave as we want?

Secret 3: The astroid event



Figure 2: Geoff Hinton

ImageNet 2012

- Alex Kriznetski
- Order of magnitude improvement in image calssification

- Things that seemed impossible now possible

Imperial College London

HOT OFF THE PRESS: Another astroid event



Figure 3: Another astroid event

ImageNet 2012

- Open AI
- Change to social order

- Things that seemed impossible now possible

Secret 3: The astroid event

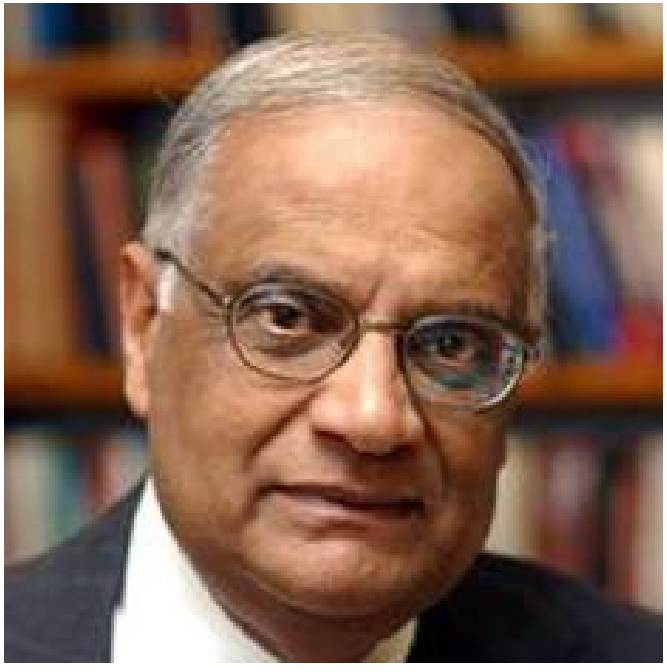


Figure 4: Professor K. S. Narendra

ML (Neural networks) was, until very recently, a dirty word in some areas of science.

- Classification problems
- Regression problems
- Dynamic systems (sort of)

Imperial College London

15 January 2024

2. What is Machine learning good at?

Problems where ML makes a difference

Machine learning is good at:

- Regression problems
- Pattern recognition problems
- Problems that involve sequential decision making
- Finding structure in graphs and networks.

Imperial College London

Regression problems



- Making predictions about economic growth

Imperial College London

Pattern recognition



- Identify *plug-in vehicles* from images

Imperial College London

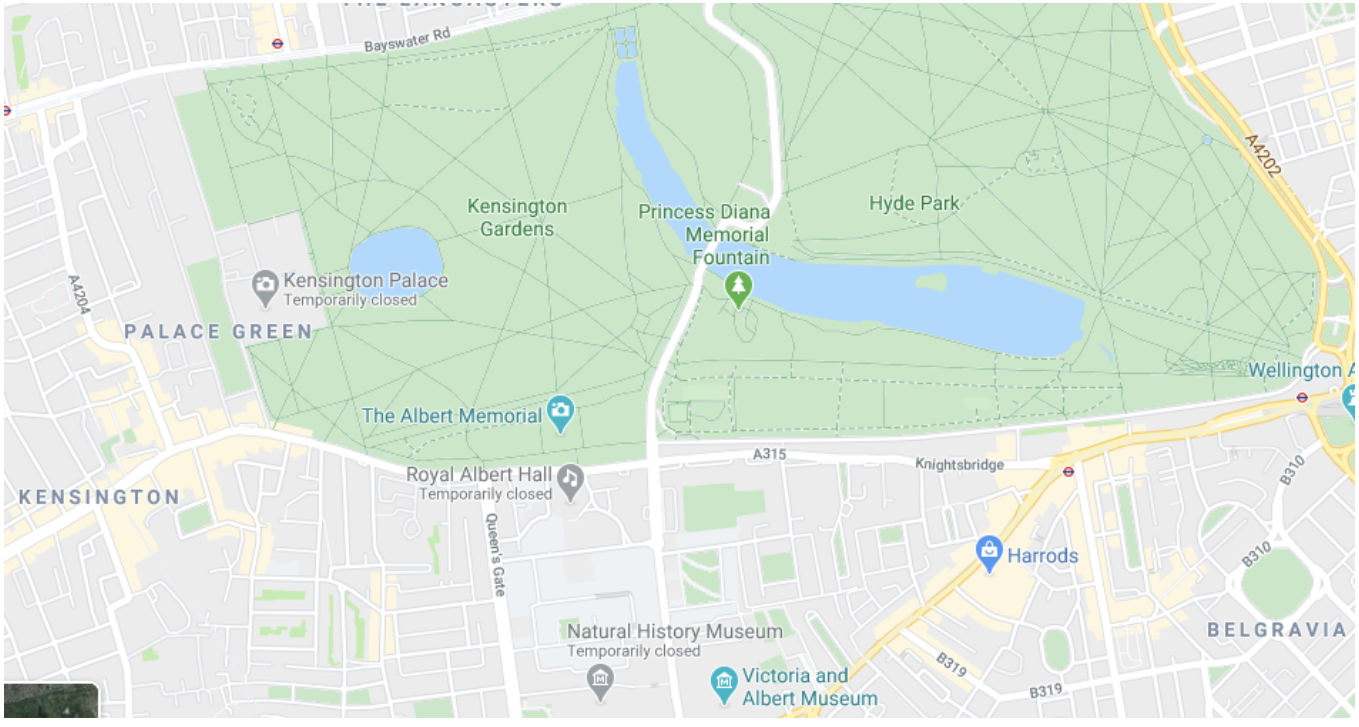
Sequential decision making



- Learn strategy to help machine win at Chess

Imperial College London

Machine learning on graphs



- Identify important aspects of graphs.

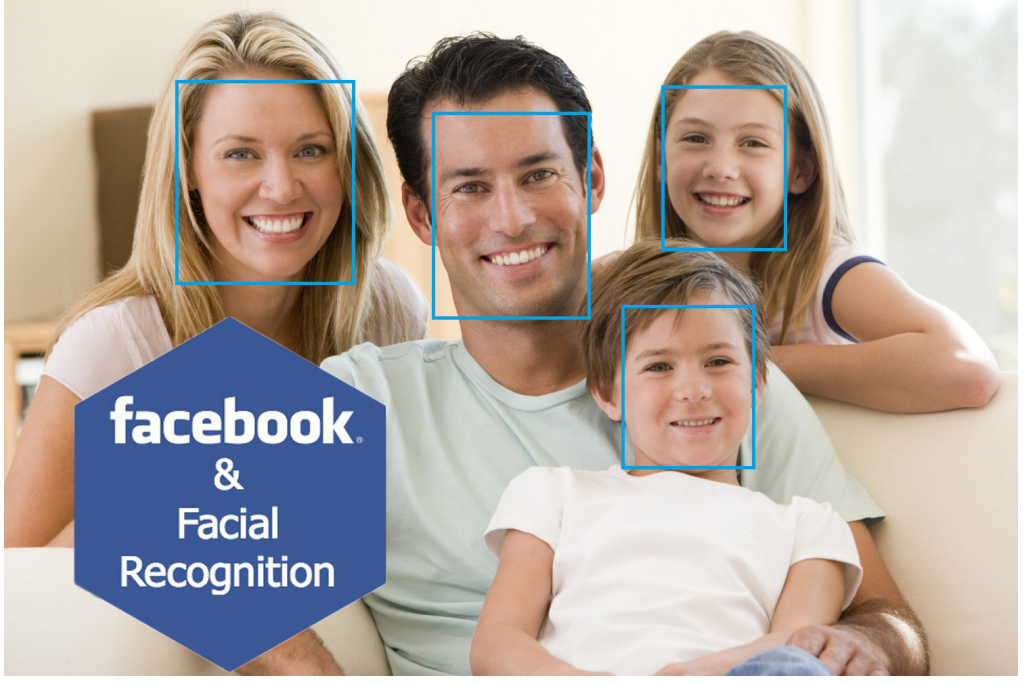
Imperial College London

15 January 2024

3. Some magic and a game (regression, classify, SDM/control, graphs)

Imperial College London

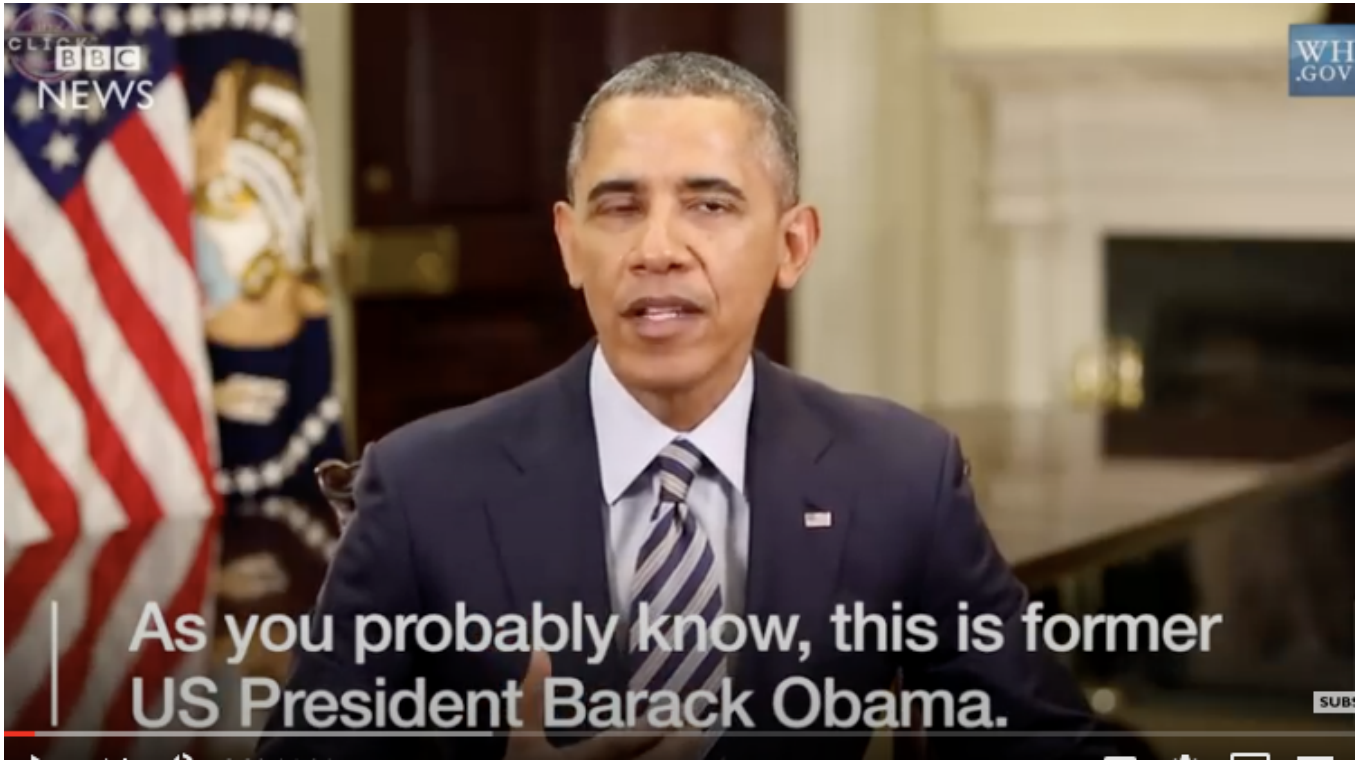
Face recognition



http://2.bp.blogspot.com/Facebook-face_recognition.jpg

Imperial College London

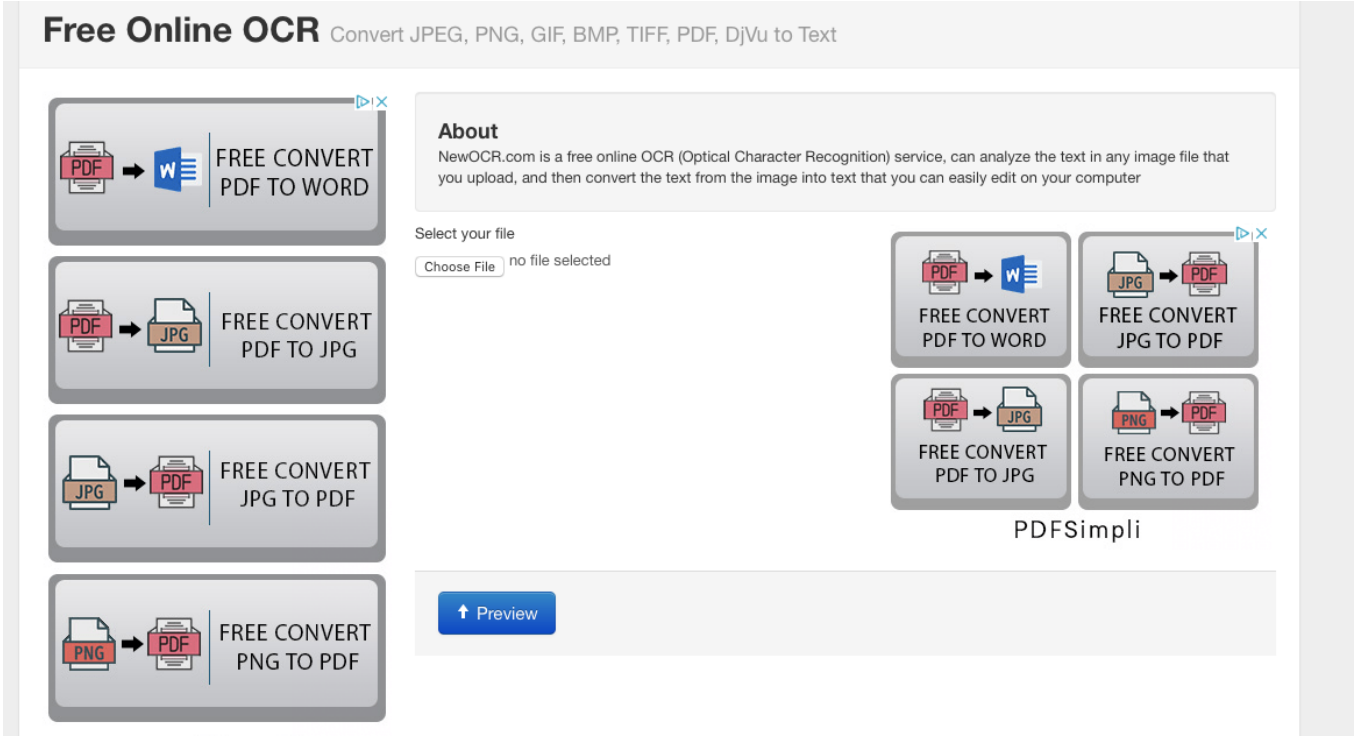
Fake video generation



Check this out:
https://www.youtube.com/watch?v=AmUC4m6w1wo

Imperial College London

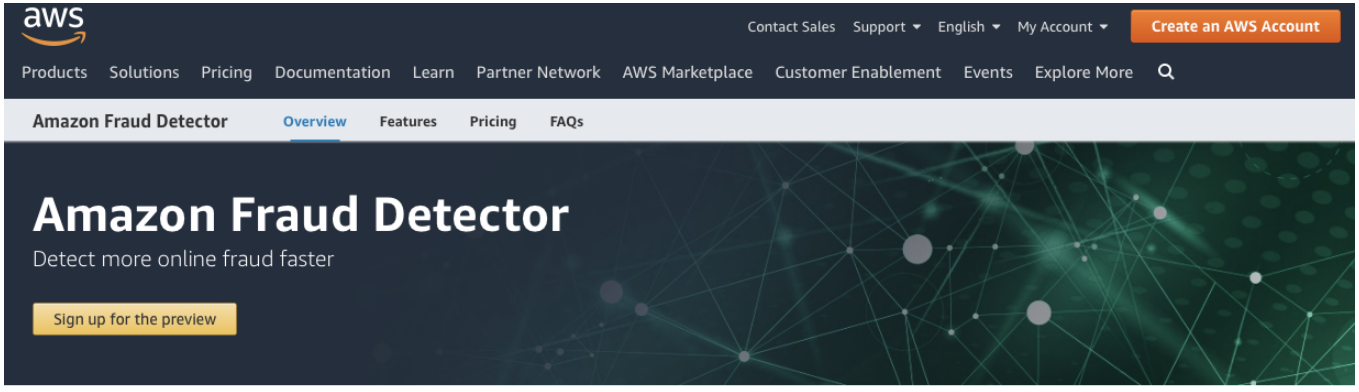
Optical character recognition



https://www.newocr.com

Imperial College London

Fraud detection



Amazon Fraud Detector is a fully managed service that makes it easy to identify potentially fraudulent online activities such as online payment fraud and the creation of fake accounts.

Globally each year, tens of billions of dollars are lost to online fraud. Companies conducting business online are especially prone to attacks from bad actors who often exploit different tactics such as creating fake accounts and making payments with stolen credit cards. Companies typically use fraud detection applications to identify fraudsters and stop them before they cause costly business disruptions. However, these applications often rely on business rules that don't keep up with the changing behaviors of fraudsters. More recent fraud detection applications have tried to use machine learning. But, they frequently use a one-size-fits-all

https://aws.amazon.com/fraud-detector/

Imperial College London

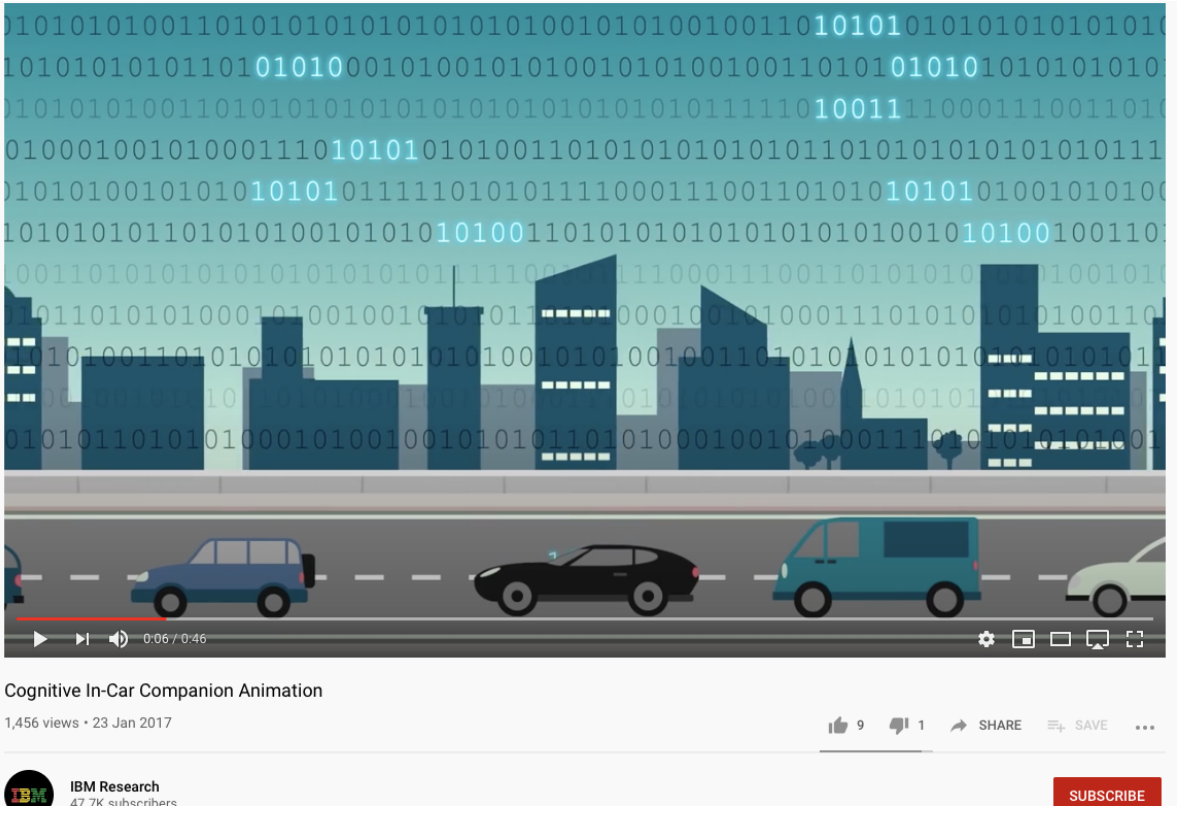
Game playing



https://deepmind.com/research/case-studies/alphago-the-story-so-far

Imperial College London

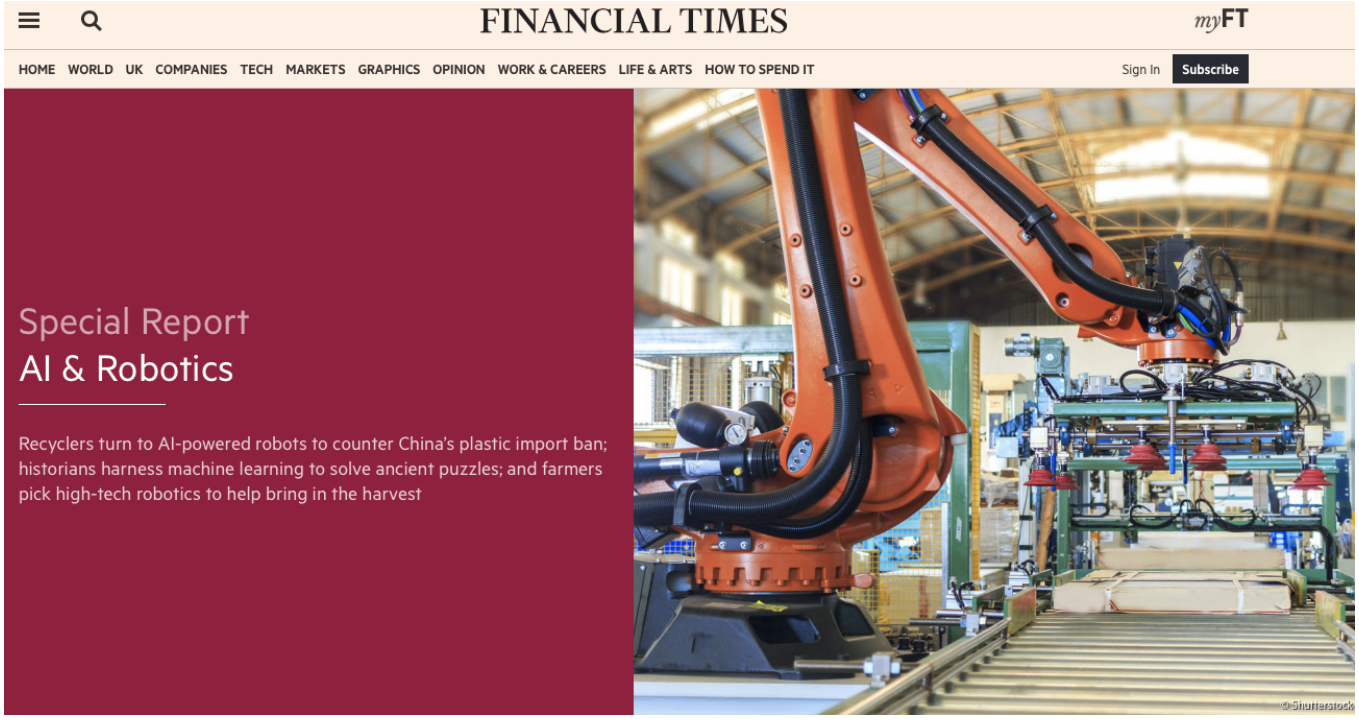
In car companions



https://www.youtube.com/watch?v=L6VyHN4PVcY&feature=youtu.be

Imperial College London

Robots



https://www.ft.com/reports/ai-robotics

Imperial College London

Chatbots

3. Facebook – Chatbot Army

Although Facebook's Messenger service is still a little...contentious (people have *very* strong feelings about messaging apps, it seems), it's one of the most exciting aspects of the world's largest social media platform. **That's because Messenger has become something of an experimental testing laboratory for chatbots**.



Some chatbots are virtually indistinguishable from humans when conversing via text

Any developer can create and submit a chatbot for inclusion in Facebook Messenger. This means that companies with a strong emphasis on customer

https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

Imperial College London

Personal companions

1. Virtual Personal Assistants

Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. All you need to do is activate them and ask “What is my schedule for today?”, “What are the flights from Germany to London”, or similar questions. For answering, your personal assistant looks out for the information, recalls your related queries, or send a command to other resources (like phone apps) to collect info. You can even instruct assistants for certain tasks like “Set an alarm for 6 AM next morning”, “Remind me to visit Visa Office day after tomorrow”.

Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences.

Virtual Assistants are integrated to a variety of platforms. For example:

- Smart Speakers: Amazon Echo and Google Home
- Smartphones: Samsung Bixby on Samsung S8
- Mobile Apps: Google Allo

https://medium.com/app-affairs/9-applications-of-machine-learning-from-day-to-day-life-112a47a429d0

Imperial College London

Cameras

CAMERAS AND PHOTOGRAPHY | TECH | ARTIFICIAL INTELLIGENCE

How AI is changing photography

Cameras' biggest recent advancements have come from AI, not sensors and lenses

By Sam Byford | @345triangle | Jan 31, 2019, 8:00am EST

SHARE

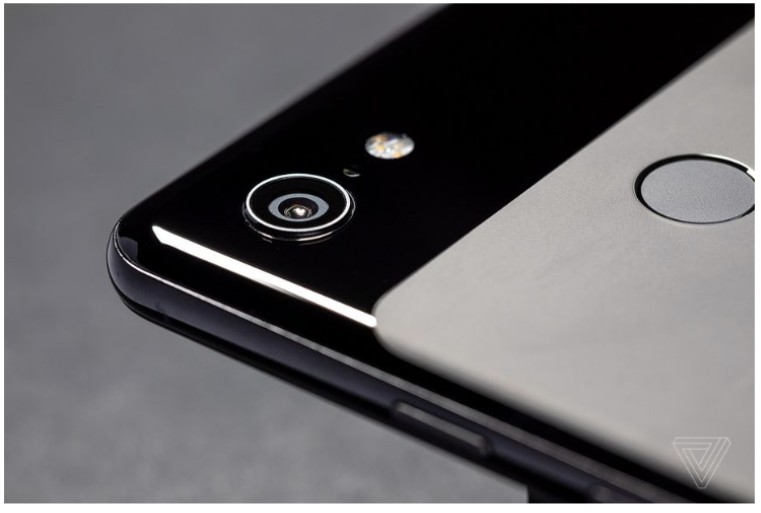


Photo by James Bareham / The Verge

https://www.theverge.com/2019/1/31/18203363/ai-artificial-intelligence-photography-google-photos-apple-huawei

Imperial College London

Additive manufacturing

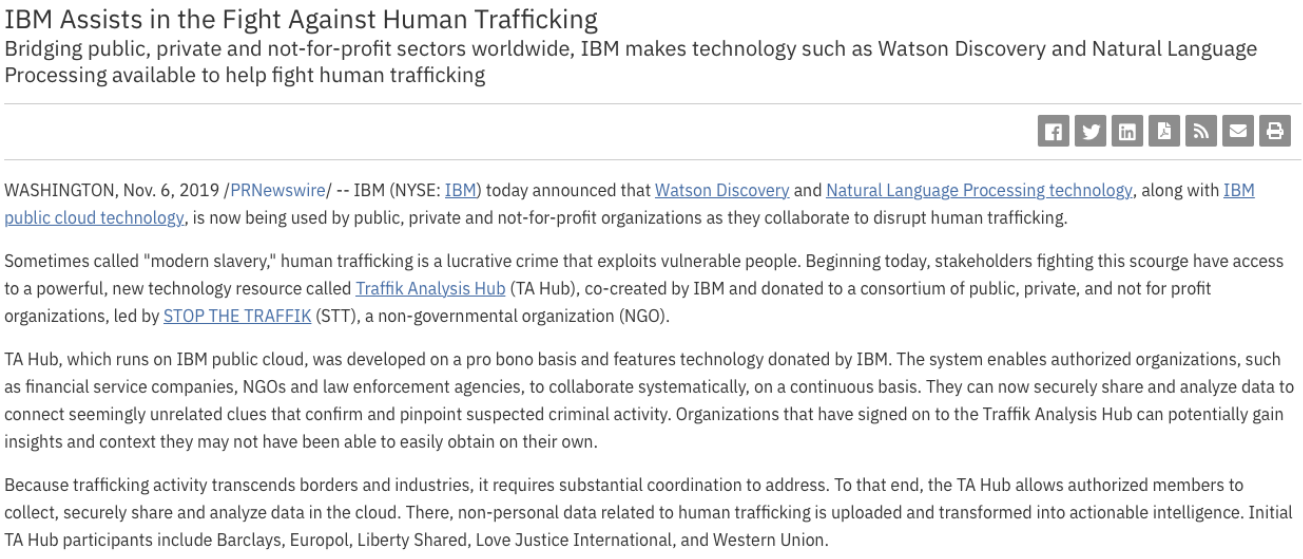


Check this out:

https://github.com/ShichenLiu/SoftRas

Imperial College London

Human trafficing



IBM Assists in the Fight Against Human Trafficking

Bridging public, private and not-for-profit sectors worldwide, IBM makes technology such as Watson Discovery and Natural Language Processing available to help fight human trafficking

WASHINGTON, Nov. 6, 2019 /PRNewswire/ -- IBM (NYSE: IBM) today announced that Watson Discovery and Natural Language Processing technology, along with IBM public cloud technology, is now being used by public, private and not-for-profit organizations as they collaborate to disrupt human trafficking.

Sometimes called "modern slavery," human trafficking is a lucrative crime that exploits vulnerable people. Beginning today, stakeholders fighting this scourge have access to a powerful, new technology resource called Traffik Analysis Hub (TA Hub), co-created by IBM and donated to a consortium of public, private, and not for profit organizations, led by STOP THE TRAFFIK (STT), a non-governmental organization (NGO).

TA Hub, which runs on IBM public cloud, was developed on a pro bono basis and features technology donated by IBM. The system enables authorized organizations, such as financial service companies, NGOs and law enforcement agencies, to collaborate systematically, on a continuous basis. They can now securely share and analyze data to connect seemingly unrelated clues that confirm and pinpoint suspected criminal activity. Organizations that have signed on to the Traffik Analysis Hub can potentially gain insights and context they may not have been able to easily obtain on their own.

Because trafficking activity transcends borders and industries, it requires substantial coordination to address. To that end, the TA Hub allows authorized members to collect, securely share and analyze data in the cloud. There, non-personal data related to human trafficking is uploaded and transformed into actionable intelligence. Initial TA Hub participants include Barclays, Europol, Liberty Shared, Love Justice International, and Western Union.

https://newsroom.ibm.com/2019-11-06-IBM-Assists-in-the-Fight-Against-Human-Trafficking

Imperial College London

Video based security

3. Videos Surveillance

Imagine a single person monitoring multiple video cameras! Certainly, a difficult job to do and boring as well. This is why the idea of training computers to do this job makes sense.

The video surveillance system nowadays are powered by AI that makes it possible to detect crime before they happen. They track unusual behaviour of people like standing motionless for a long time, stumbling, or napping on benches etc. The system can thus give an alert to human attendants, which can ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they help to improve the surveillance services. This happens with machine learning doing its job at the backend.

https://medium.com/app-affairs/9-applications-of-machine-learning-from-day-to-day-life-112a47a429d0

Imperial College London

Traffic predictions

2. Predictions while Commuting

Traffic Predictions: We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences.

Online Transportation Networks: When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning. Jeff Schneider, the engineering lead at Uber ATC reveals in a an interview that they use ML to define price surge hours by predicting the rider demand. In the entire cycle of the services, ML is playing a major role.

https://medium.com/app-affairs/9-applications-of-machine-learning-from-day-to-day-life-112a47a429d0

Imperial College London

Spam detection



https://towardsdatascience.com/detecting-a-simple-neural-network-architecture-using-nlp-for-email-classification-f8e9e98742a7

Imperial College London

Medical diagnosis

How do we use AI in medicine

AI-enabled tools can identify meaningful relationships in raw data and has the potential to be applied in almost every field of medicine, including drug development, treatment decisions, patient care, and financial and operational decisions.

With AI, healthcare professionals could can tackle complex problems that would be difficult, time-intensive, or inefficient to address alone.AI could be a valuable resource for medical professionals, allowing them to better use their expertise and provide value across the health ecosystem.



https://www.ibm.com/watson-health/learn/artificial-intelligence-medicine

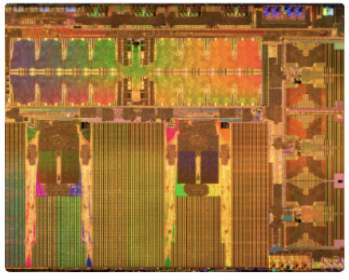
Imperial College London

Autonomous driving

TESLA MODEL S MODEL 3 MODEL X MODEL Y CYBERTRUCK ROADSTER ENERGY SIGN IN

Autopilot

We develop and deploy autonomy at scale. We believe that an approach based on advanced AI for vision and planning, supported by efficient use of inference hardware is the only way to achieve a general solution to full self-driving.



Hardware

Build silicon chips that power our full self-driving software from the ground up, taking every small architectural and micro-architectural improvement into account while pushing hard to squeeze maximum silicon performance-per-watt. Perform floor-planning, timing and power analyses on the design. Write robust, randomized tests and scoreboards to verify functionality and performance. Implement compilers and drivers to program and communicate with the chip, with a strong focus on performance optimization and power savings. Finally, validate the silicon chip and bring it to mass production.



Neural Networks

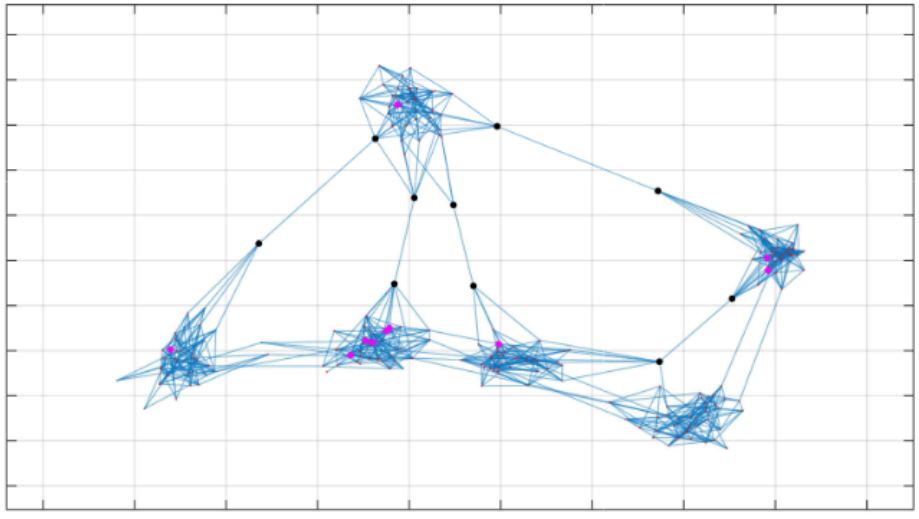
Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the road layout, static infrastructure and

https://www.tesla.com/en_GB/autopilotAI

Imperial College London

Covid testing

A different type of machine learning problem is when we try to learn properties of graphs. For example in the context of contact tracing in Covid-19, we would like to learn who to test.



We will have some fun with these algorithms.

Imperial College London

15 January 2024

4. Machine learning's and design engineering

Imperial College London

ML and design engineering

- Using ML to design products
- Using ML as part of a designed system
- Designing ML algorithms

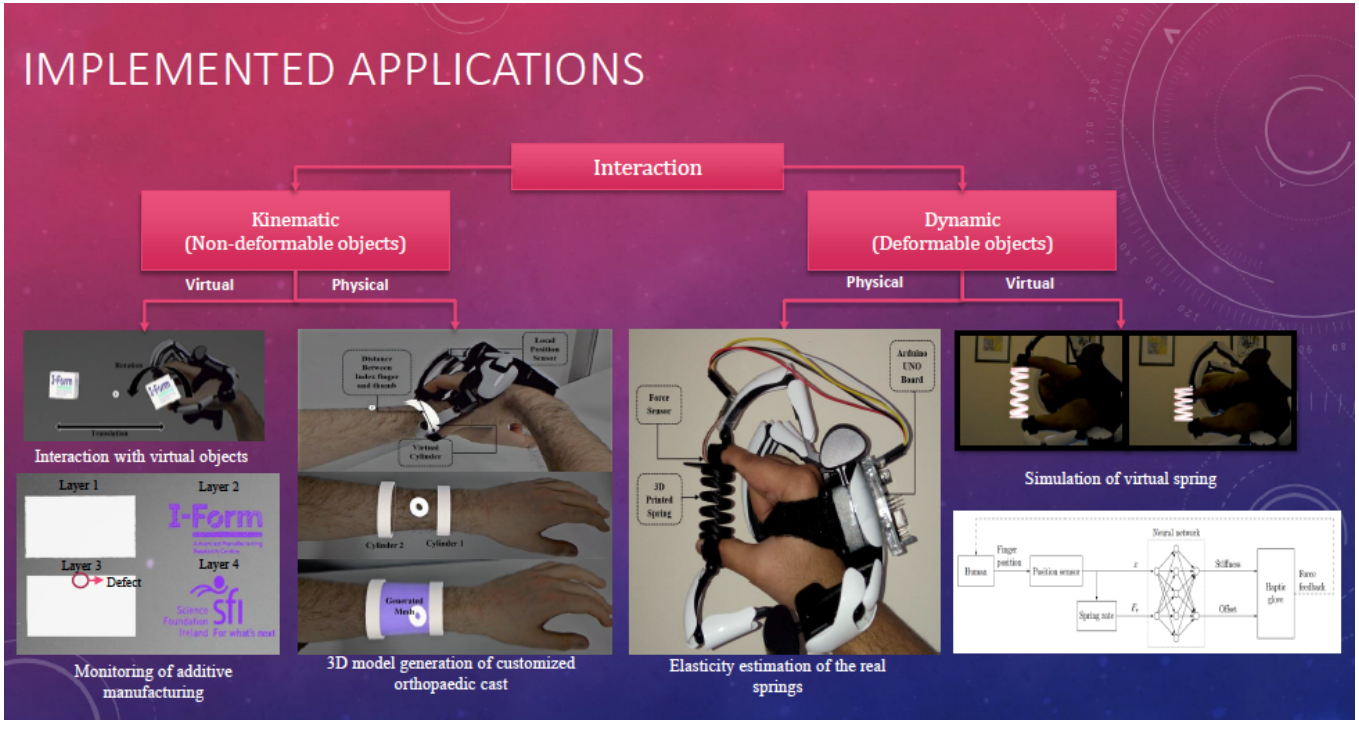
Imperial College London

Product design using ML



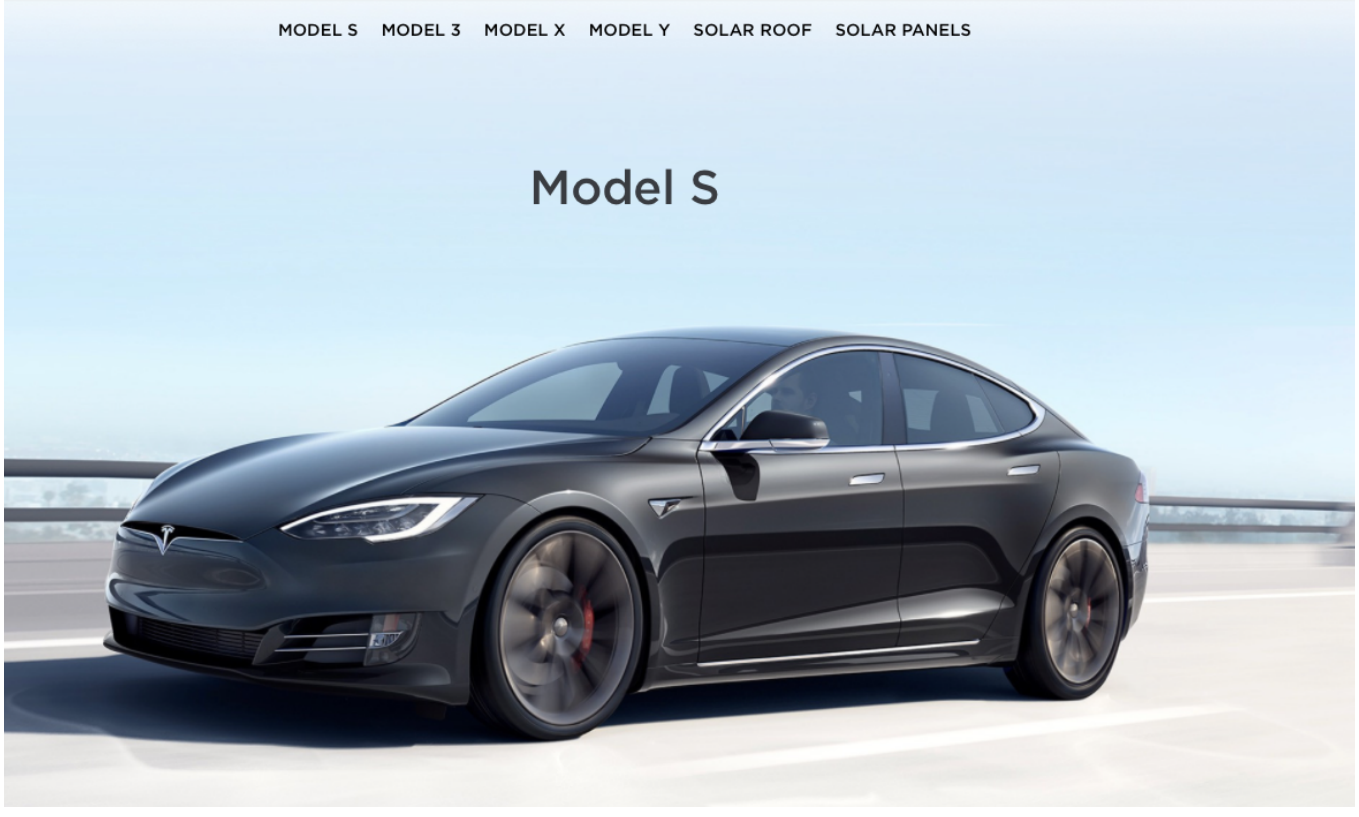
Imperial College London

Product design using ML



Imperial College London

Using ML in products



Imperial College London

Designing ML Algorithms: Bias

BBC | Your account | News | Sport | Weather | iPlayer | Sounds

NEWS

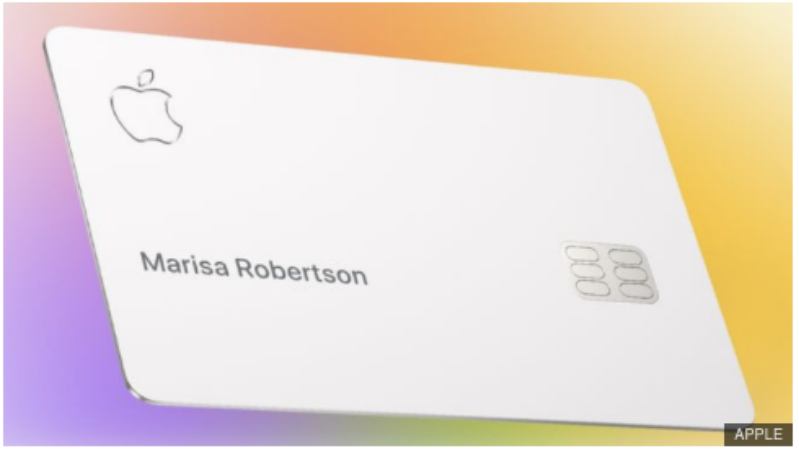
Home | UK | World | Business | Politics | Tech | Science | Health | Family & Education

Business | Your Money | Market Data | Companies | Economy

Sexist and biased? How credit firms make decisions

By Kevin Peachey
Personal finance reporter

18 November 2019 | Share



Jamie Heinemeier Hansson had a better credit score than her husband, tech entrepreneur David. They have equal shares in their property and file joint tax returns.

Imperial College London

Designing ML Algorithms: Noise



BBC | Your account | News | Sport | Weather | iPlayer | Sounds

NEWS

Home | UK | World | Business | Politics | Tech | Science | Health | Family & Education

Science & Environment

AAAS: Machine learning 'causing science crisis'

By Pallab Ghosh
Science correspondent, BBC News, Washington

16 February 2019 Share

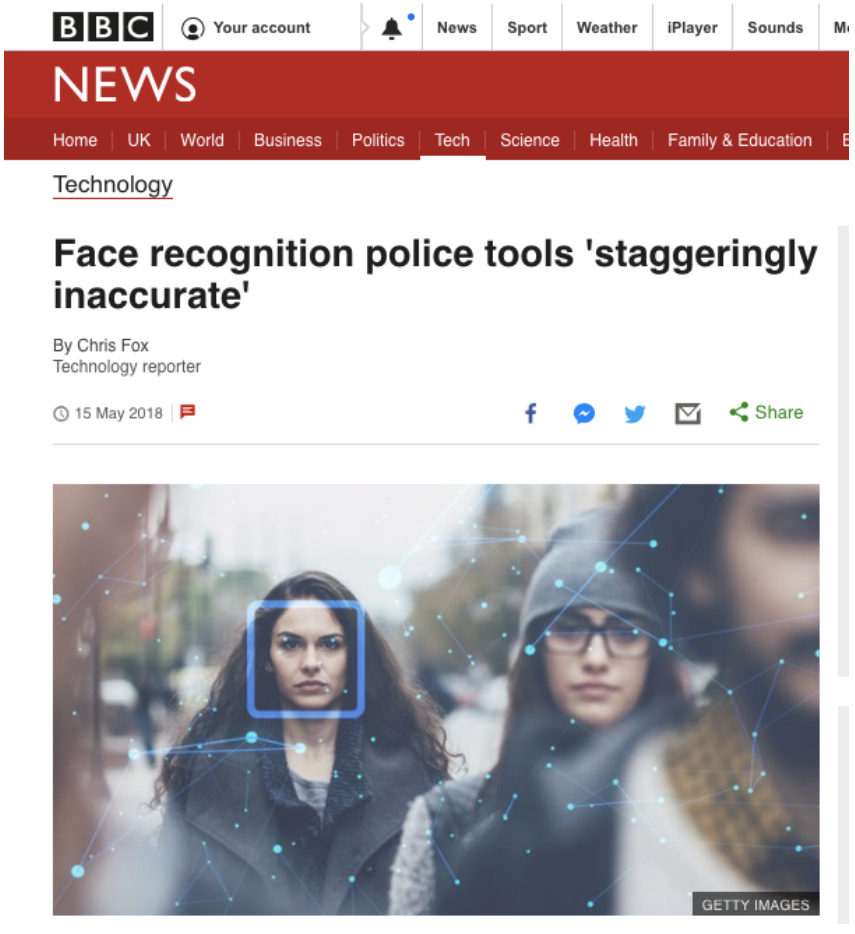
AAAS meeting

REUTERS

Astronomy is one of the many areas of science in which machine learning is used to make discoveries

Imperial College London

Designing ML Algorithms: Ethics



Imperial College London

15 January 2024

The Course

What we will look at?

Examples of:

- regression problems;
- pattern recognition problems;
- problems that involve sequential decision making;
- finding structure in graphs and networks.

Types of learning

We will encounter

- supervised learning (data with labels);
- unsupervised learning (data without labels);
- learning based on exploration (finding optimal policies);
- machine learning on graphs.

Imperial College London

Supervised learning



$x(i) =$ Image; $y(i) =$ car.

Imperial College London

Unsupervised learning



$x(i) =$ Wine classified by sugar content;

Imperial College London

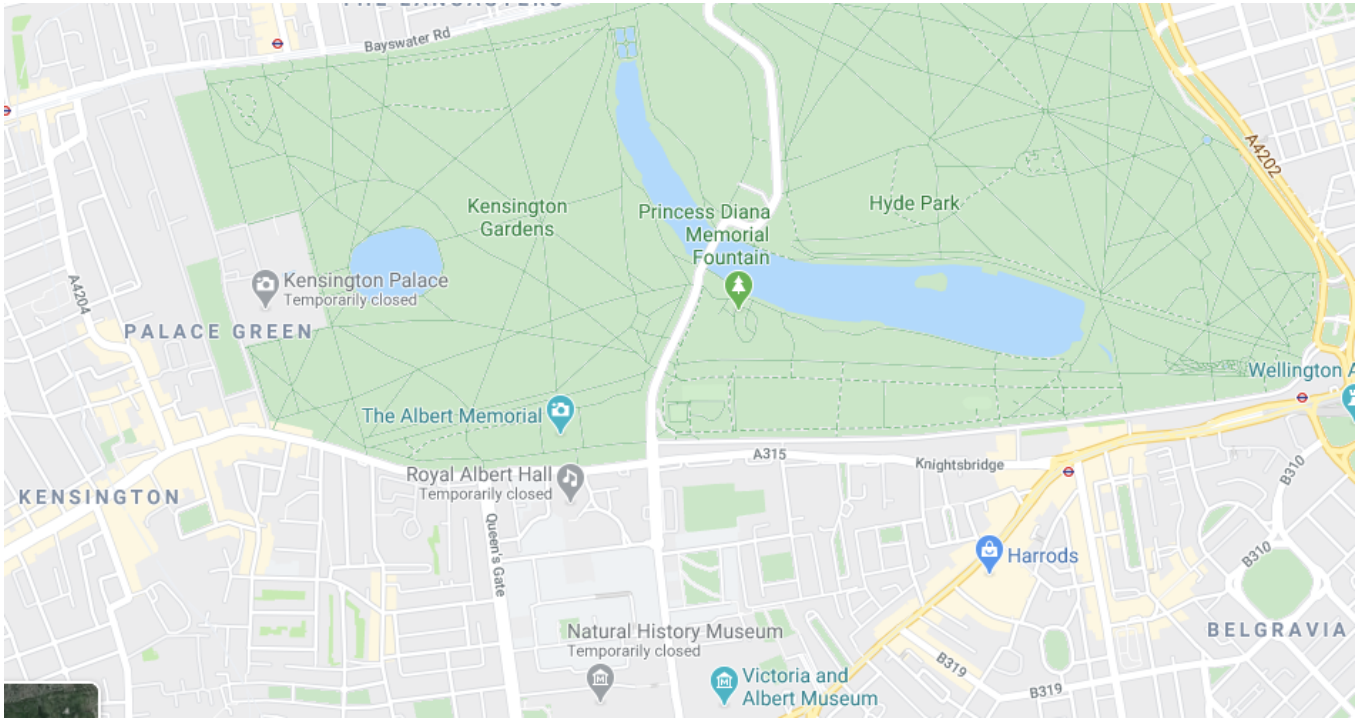
Exploration based learning (reinforcement learning)



$x(1), x(2), ...$ = Sequence of moves to win game;

Imperial College London

Machine learning on graphs



$x(i) =$ most important street;

Imperial College London

Lecture content

5. Overview of lecture content		
Lecture title	**Instructor**	**Content**
1. The magic of machine learning	Robert Shorten	Overview of course Introduction to team and assessment Some background material Reading list Background material for laboratories
2. The machine learning portfolio	Robert Shorten	What is Machine learning? Types of machine learning? Basic mathematical background Sprint on Markov chains
3. Key ideas in one lecture	Robert Shorten	The basics explained in an elementary manner.
4. Practicalities of real-life machine learning.	Robert Shorten	Some of the practical issues in dealing with data.
5. ANNs	Robert Shorten	An introduction to artificial neural networks.
6. ANNs	Robert Shorten	Learning algorithms for artificial neural networks.
7. CNNS and RNNS	Pietro Ferraro	An overview of convolutional neural networks
8. SVM	Robert Shorten	An overview of support vector machines and Kernel methods
9. Supervised learning	Robert Shorten	An introduction to K-means clustering
10. Reinforcement learning	P. Ferraro	What are Markov Decision processes?
11. Reinforcement learning	P. Ferraro	What is Q-learning?
12. SPtoken	Robert Shorten	An example of reinforcement learning
13. Spectral graph theory.	Robert Shorten	An introduction to machine learning over graphs.
14. Spectral graph theory	Robert Shorten	An introduction to machine learning over graphs.
15. Fairness in ML	Robert Shorten	Advanced topic
16. Optimal Transport	Robert Shorten	Advanced topic

Imperial College London

Labs

Four labs have been designed to support key aspects of the course.

Labs are designed to be self-contained - i.e. can be completed offline.

Clinics during tutorial sessions to support students with difficulties.

Pre-requisite - installation of Python. Any trouble - reach out to Pietro or myself.

Labs start week of Jan 22nd.

Imperial College London

Labs and tutorials - the Clinics

Lab 0: Installation of Anaconda or Python. *(Jan 17th)*	
Clinic	The class will be divided into smaller groups and instructors will assist with anaconda installation.

Lab 1: The objective of the first lab is to program and experiment with a first classifier. The specific objectives will be to *understand and code a simple neuron, and to develop an understanding of its limitations. (Jan 24th)*	
Clinic	The class will be divided into smaller groups and instructors will assist with the lab exercise.

Lab 2: The objective of the second lab is to program a neural network with one or more hidden layers, and to understand how to build a general classifier. *(Jan 31st)*	
Clinic	The class will be divided into smaller groups and instructors will assist with the lab exercise.

Assignment deadline (Feb 7th)	
Presentations	No clinic this week. Instead, presentations based on reading list.

Lab 3: The objective of the third lab is to program a neural network using Pytorch. (Feb 21st)	
Clinic	The class will be divided into smaller groups and instructors will assist with the lab exercise.

Lab 4: The objective of the fourth lab is to design a convolutional neural network for the MNIST database. (Feb 28th)	
Clinic	The class will be divided into smaller groups and instructors will assist with the lab exercise.

Lab 5: LAB EXAM (March 6th)	
Lab Exam	In class test based on labs completed. This is a viva exam – 15% of final mark.

Tutorials (March 13th)	
Clinics	Revision of class material and paper presentations

Imperial College London

Labs and tutorials - the Clinics

Six tutorial rooms. Class split into groups (2 students per group).

We will rotate each week - meet each group for 30 mins. Full list on Blackboard.

Labs start in week 2 (next week - week of Jan 22nd).

Imperial College London

Assignments

Assignment Organised around popular and quirky applications.

Imperial College London

Assessment

Assessment will have **three components**.

- An **end of term exam** which will constitute 70% of the final grade.

- A **short oral examination** based on the completed labs. This will constitute 15% of the final grade.

- **Written assignment**. This will constitute 15% of the final grade.

Imperial College London

Software and labs

- Software: Python 3.8 and above.
- You can use any Python distribution you prefer. The one we recommend is visual studio code *https://code.visualstudio.com*
- Required libraries: Numpy, Torch, Torchvision. For Torch and Torchvision, please use the following link `https://pytorch.org/` (select Conda or Pip, depending on which one you are using and for "compute platform" select CPU.)
- **Please, make sure that all the required software is installed before Labs start**

Imperial College London

Software and labs

- Coding a single Neuron
- Coding a Neural Network and a classifier
- Using Pytorch to create a complex Neural Network
- Using Pytorch to create a Convolutional Neural Network



Imperial College London

Books

1. The hundred-page machine learning book, A. Burkov

2. Hands-on machine learning with SciKit-learn, Keras and TensorFlow, A. Geron

3. Decision making under uncertainty, M. Kovhendorfer

4. Google's page rank and beyond, A. Langville, C. Meyer

5. Mathematics for Machine Learning, M. Diesenroth *et. al.*

Imperial College London

15 January 2023

The machine learning portfolio

Imperial College London

Lecture overview

- What is machine learning?
- Types of learning
- Mathematics of machine learning
- Machine learning buzzwords

Imperial College London

15 January 2023

1. What is machine learning?

Imperial College London

What is machine learning?

Machine learning is about model building. Models are just compact mathematical representations of data. *All models are wrong, but some are useful*.

In machine learning we build a model to solve a problem:

1 - by gathering some real data;

2 - and then build a mathematical model of some phenomena based on the gathered data.

ML algorithms are classified according to the manner in which the model is built (supervised learning, unsupervised learning, reinforcement learning, graph based algorithms)

Imperial College London

15 January 2023

2. Types of machine learning

Supervised learning

In supervised learning, the dataset is a collection of labelled examples $(x(i), y(i))$, $i \in 1, .., N$. Each element $x(i)$ is called a feature vector, and each $y(i)$ is a label. Typically, the $x(i)$, $y(i)$ could be vectors.

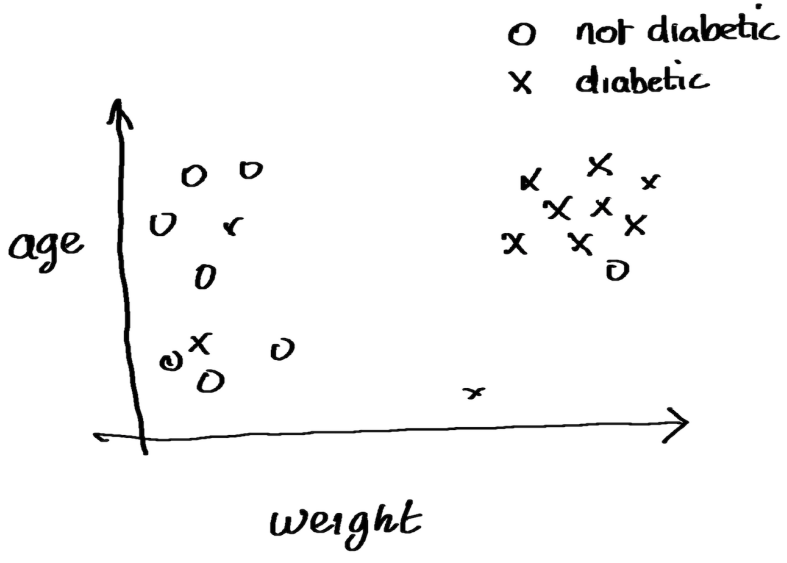
The goal of a supervised learning algorithm is to produce an algorithm such that given a feature vector x, we can produce a label y. Typically, we have two types of problems in supervised learning:

- regression *and* classification.

Imperial College London

Supervised learning: classification

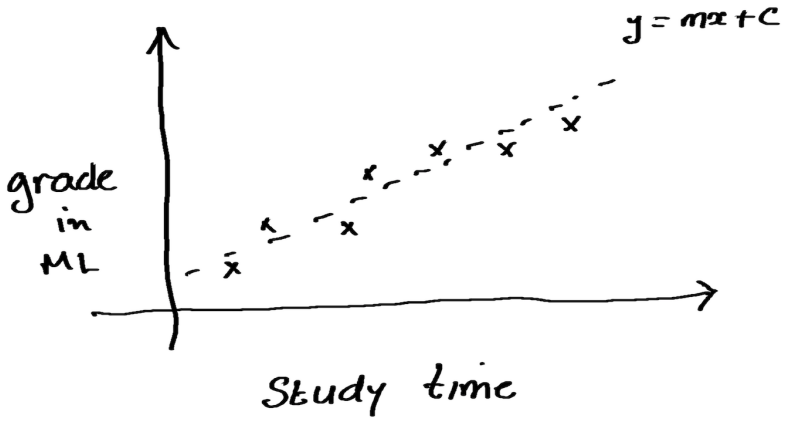
For example, suppose that $x(i)$ was a vector that recorded age and weight of an individual, and $y(i)$ recorded whether this individual was diabetic. Could we predict *diabetes* given an $x(i)$?



Imperial College London

Supervised learning: regression

Suppose now that $x(i)$ was a vector that recorded time an individual studied, and $y(i)$ recorded the final grade in ML. Could we predict success in ML class given an $x(i)$?



Un-supervised learning

In unsupervised learning the training data is not labeled. Here the system tries to learn without a teacher.

Adding labels to data is very expensive. Usually humans have to do it. Think of taking pictures to monitor a machine.

Unsupervised data-sets are by far the most common in practice.

Imperial College London

Un-supervised learning

Used in clustering applications; anomaly detection; density estimation applications.



Semi-supervised learning

In semi-supervised learning, the dataset contains both labeled and un-labeled examples. The goal of the semi-supervised learning is the same as in supervised learning.

An example of un-supervised learning is services such as Google photo.

Images are clustered according to faces appearing in the image. *This is the unsupervised part of the algorithm.*

Some images are labeled - this is Bob Shorten. Any cluster with Bob Shorten then has an identifier.

Reinforcement learning

A completely different ML system is called *reinforcement learning*.

Reinforcement learning is really useful in Robotic applications, or in traffic applications.

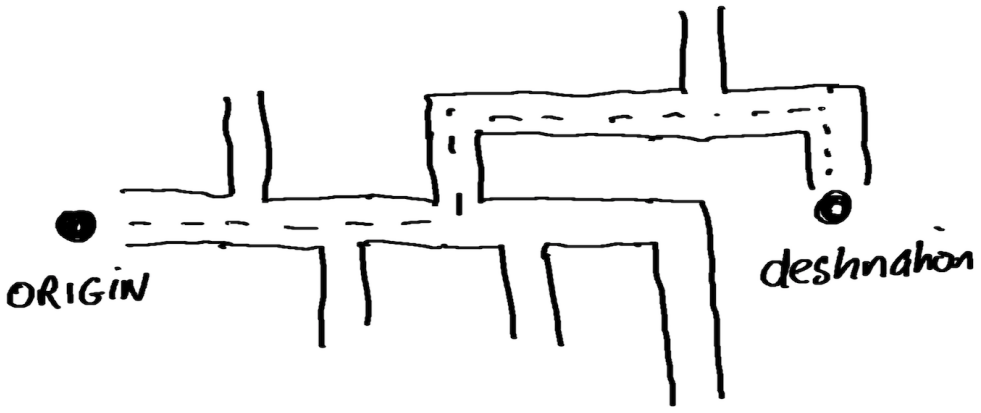
Here the learning system actively probes the environment, and is rewarded, or punished, based on the outcome.

The objective is then to find strategy, or *policy*, that maximizes the reward.

Imperial College London

Reinforcement learning

Imagine an automomous vehicle, trying to find the best route to a destination. The route with the shortest path maximizes the reward.



Reinforcement learning

Reinforcement learning has been amazingly successful in many applications.

- Robotics
- Games - Chess, Go.
- Control of complex dynamical systems.

Imperial College London

Machine learning on graphs

A different type of machine learning problem is when we try to learn properties of graphs. For example in the context of contact tracing in Covid-19, we would like to learn:

- Who are the most connected people in the population?
- Do sub-communities exist in the graph?
- Which individuals link sub-communities together?

Imperial College London

Machine learning on graphs

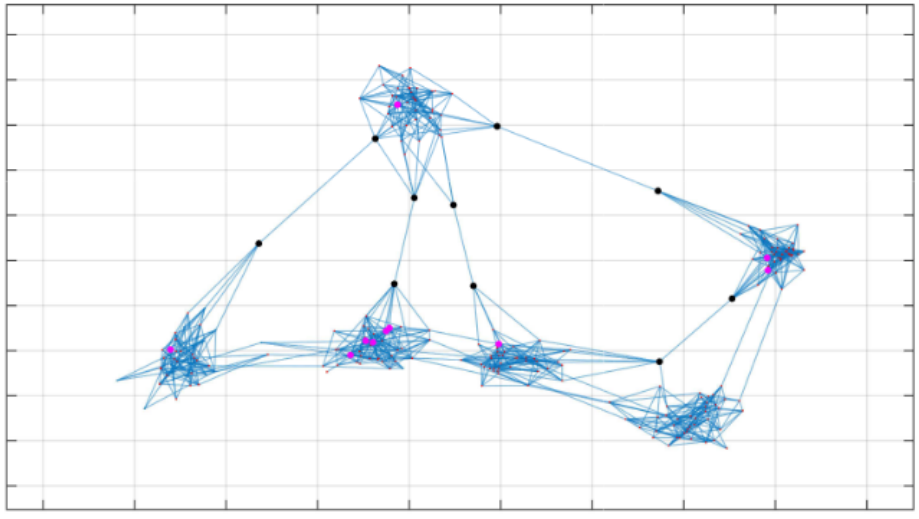
A different type of machine learning problem is when we try to learn properties of graphs. For example in the context of contact tracing in Covid-19, we would like to learn:

- Spreaders?
- Nursing homes, Satmar community?
- Stragegic spreaders?

Imperial College London

Machine learning on graphs

A different type of machine learning problem is when we try to learn properties of graphs. For example in the context of contact tracing in Covid-19, we would like to learn:



We will have some fun with these algorithms.

Imperial College London

15 January 2023

4. The mathematics of machine learning

Imperial College London

Mathematical background

So whats needed to do some ML? ML is built on four areas of mathematics.

1. Optimization
2. Probabilty
3. Linear algebra
4. Markov chains

1-3 is revision. We'll spend a little bit more time learning about Markov chains.

Notation

First some notation. We will try to keep to the following convention.

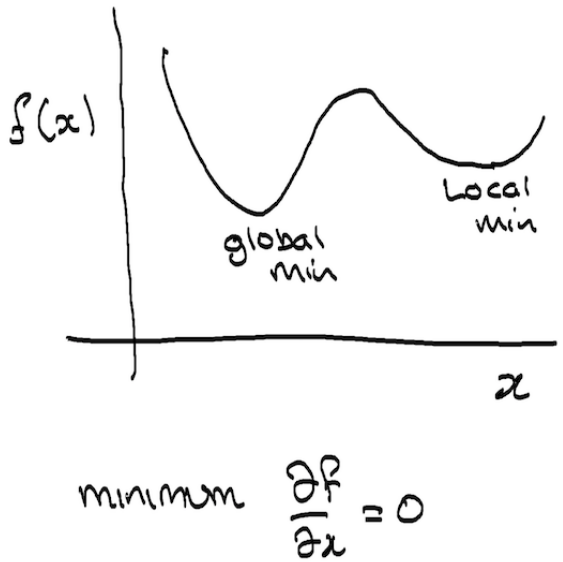
- Matrices variables are denoted in capital letters - for example, the matrix P.
- Vectors are denoted in lower case, with a subscript denoting the element of a matrix - for example $w^T = (w_1, ..., w_n)$.
- Sometimes vectors are indexed, using for example, time. We denote this by: $w(i)^T = (w_1(i), ..., w_n(i))$ meaning the vector w at time index i.

In out context, one can think of subscripts denoting variables, and parentheses denoting measured data points.

Imperial College London

Optimization

I am not going to go over what David Boyle did. I just want to make a few points.



The notion of a **gradient** is central to many aspects of machine learning.

Imperial College London

Lagrange multipliers

Some more sophisticated machine learning problems give rise to constrained optimization problems.

A standard method for dealing with constrained optimization problems is to use the concept of Lagrange multipliers.

Do you remember these? If you don't, please revise.

Lagrange multipliers

Lagrange multipliers are useful to solve problems

$$min_x f(x) \tag{1}$$

with additional constraints $g_1(x) < 0, ..., g_m(x) < 0$. To solve such problems we form the Lagrangian:

$$L(x, \lambda_1, ..., \lambda_m) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \tag{2}$$

Solving the constrained optimization is the same as solving (here λ denotes the vectors of all λ's).

$$\min_x \max_\lambda L(x, \lambda_1, ..., \lambda_m) \tag{3}$$

Imperial College London

Probability theory

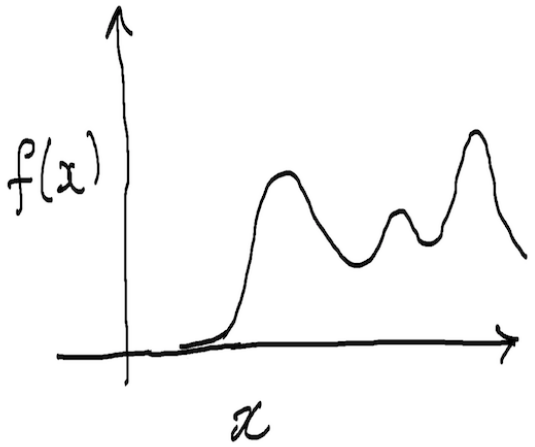
Random variable: A variable whose outcome is determined by a random event. There are two types of random variable - *discrete* and *continuous*.

A discrete random variable takes on a countable set of values such as *red*, *blue*, ...

The probability distribution of a discrete random variable is described by a list of probabilities, all of which sum to 1. Sometimes this function is called a **probability mass function**.

Probability theory

A continuous random variable (CRV) takes on an infinite number of values in some interval. The probability density function of a CRV is described by a probability density function.



$$\int_{-\infty}^{\infty} f(x)\,dx = 1$$

Imperial College London

Some basic concepts

For the discrete random variable X:

Expectation: $\mathbb{E}(X) = \sum_{i=1}^{k} x(i)p(x(i))$. $\mathbb{E}(X)$ is often denoted μ.
Variance: $var(x) = \mathbb{E}\big((X-\mu)^2\big)$. $var(x)$ is often denoted by σ^2.

For the continuous random variable X:

Expectation: $\mathbb{E}(X) = \int_{-\infty}^{\infty} xf(x)dx$. $\mathbb{E}(X)$ is often denoted μ.
Variance: As before $var(x) = \mathbb{E}\big((X-\mu)^2\big)$. $var(x)$ is often denoted by σ^2.

Some basic concepts

Most of the time we do not know $f(x)$ but can observe some values of X. The collection of these samples is called a data-set.

Often we are interested in estimating some statistic associated with $f(x)$ from the observed samples. If we denote the observed set of samples by $S_X = \{x_1, ..., x_N\}$, then as estimate of some statistic $\hat{\theta}(S_X)$ is said to be unbiased if:

$$\mathbb{E}(\hat{\theta}(S_X)) = 0. \tag{4}$$

Bayes theorem

The conditional probability $p(X = x|Y = y)$ is the probability that one random variable takes a certain value, given that another random variable takes a specific value. Bayes theorem states:

$$p(X = x|Y = y) = \frac{p(Y = y|X = x)p(X = x)}{p(Y = y)} \tag{5}$$

Bayes theorem is very useful for estimating unknown parameters of distributions.

Imperial College London

15 January 2023

5. Markov Chains and Machine Learning

A few comments

I now want to talk a little bit about Markov chains. I will do this for two reasons.

- They are genuinely useful for learning the structure of graphs.
- They underpin the theory of *Reinforcement Learning*.

Imperial College London

What is a Markov Chain

A Markov chain is a special type of dynamic system that evolves probabilistically.

The defining characteristic of Markov chains is that they have no memory.

They are really useful in modelling things like the evolution of the weather from day to day.

They are also finding new application is modelling and characterising network behaviour. We are interested in this latter application of Markov chains.

Background

The operation of a Markov chain describes a process where a system can be in a finite number of states.

For example, the weather can be "hot", "cold", "rainy", "icy", and so on.

The transition from one state to another is probabilistic, rather than deterministic. For example, if its "rainy" today, we can say with probability p_R that it will be rainy tomorrow.

Tomorrows weather cannot be determined with certainty, but we can assign probabilities to the various possible weather conditions conditioned on todays weather.

Example

The weather in a certain city is "sunny", "cloudy" or "rainy". If it is sunny one day then sun or clouds are equally likely the next day.

If it is cloudy there is a fifty percent chance of sun, a twenty five percent chance of continued clouds, and a twenty five percent change of rain.

If it is raining, it will not be sunny, but continued rain or clouds are equally likely.

Imperial College London

Graphs and Markov Chains

What have graphs got to do with Markov chains? Well, think of fleas?

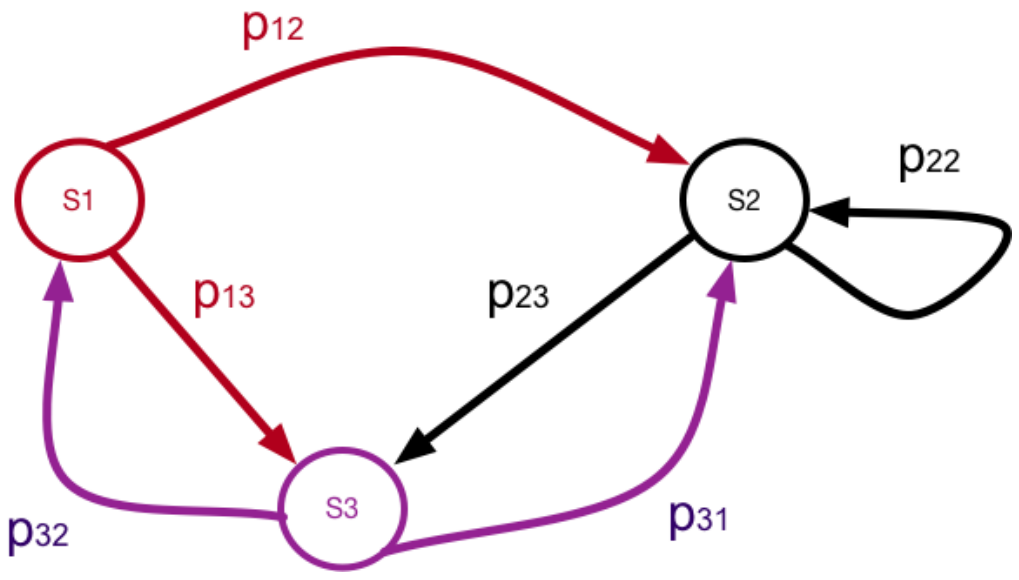


Figure 1: A flea hopping between positions with different probabilities

Imperial College London

Graphs and Markov chains

Situation is really unbiquitous:

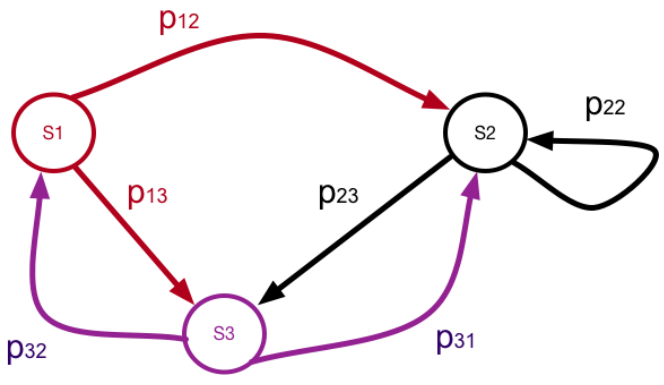


Figure 2: A directed graph

- A person browsing the web.
- Aggregated traffic in a road network.
- Virus infecting agents in a community.
- Many others?

Imperial College London

Graphs and Markov chains

Also really useful for determining:

(a) The existence of communities.

(b) The importance of nodes.

(c) The importance of links.

Can you think how?

One of the most important applications of this idea underpins the company Google - the *PageRank* algorithm. Before discussing this - a little revision on Markov chains!

Imperial College London

So how does Google rank pages?

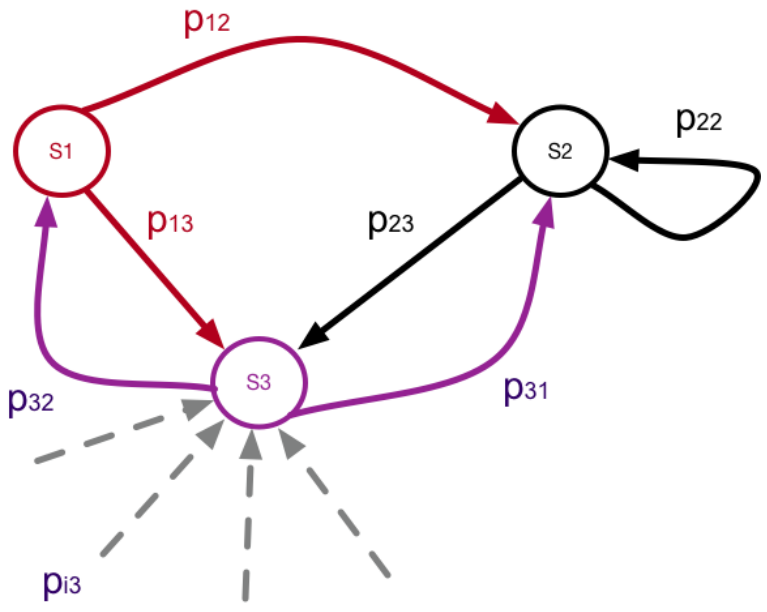


Figure 3: Which pages are more important than others?

S3 looks like it might be the most important since it has the highest *in-degree*. But this not need be the case. **Why?**

The Google model

Consider the following scenario. Three webpages $\{Bob, Alice, Mary\}$. Each web page links the others.

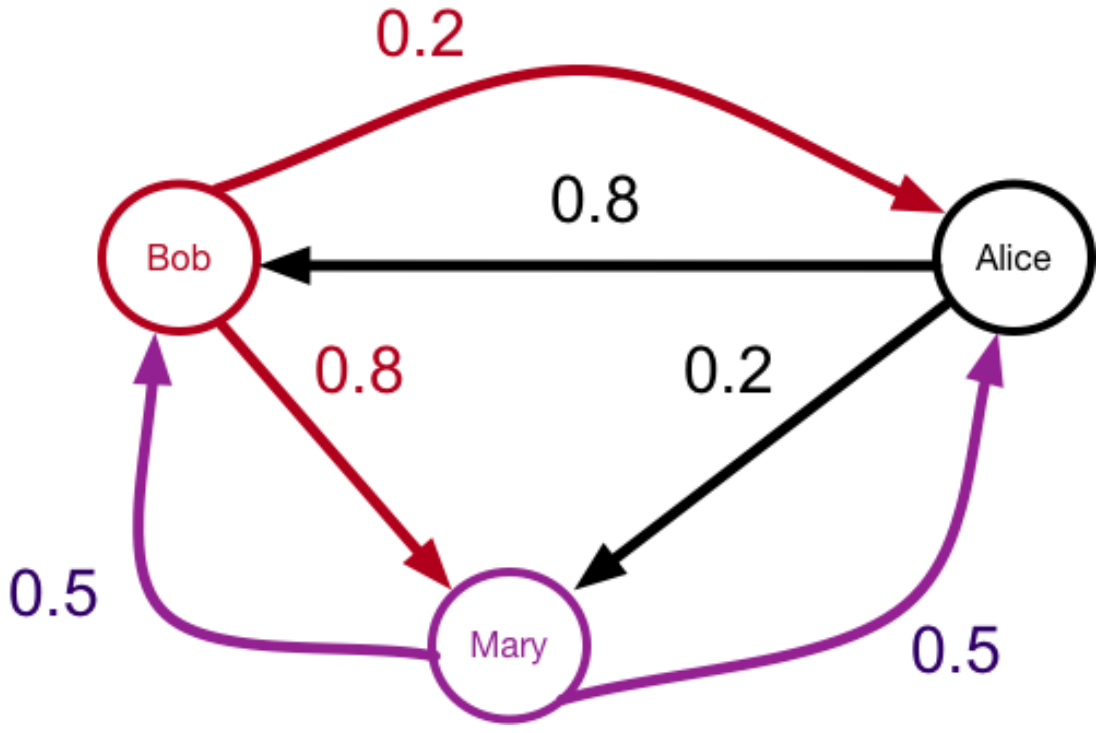
On average someone viewing Bob's page will visit Alice's web-page next with probability 0.2, and Mary's web page with probability 0.8.

On average someone viewing Alice's web-page will visit Bob's page next with probability 0.8 and Mary with 0.2.

On average someone viewing Mary's web-page will visit Bob's page next with probability 0.5 and that of Alice with 0.5.

Imperial College London

The Google model



One can think of a random walker moving from page to page. One way to measure the importance of each page is to see how often a walker lands on a particular page in the *long run*.

Imperial College London

15 January 2023

6. Machine learning buzz words

Imperial College London

Jargon busting

Regression: Predicting a real value given some information

Classification: Attaching labels to things.

Models: A compact description of a data set.

Parametric models: A models with class parameters - $y = mx + c$.

Non-parametric models: Models with no parameters - $y = mean(x_1 + x_2 + ...)$.

Imperial College London

Jargon busting

Bias: Models that favour a group (don't generalise well)

Bias - variance trade-off: Trade off between generalisation and representing training set.

Training and test set: Data used to train and test the ML tool.

Regularisation: Constraining a model with some prior information.

Over-fitting: Learning the data-set too well. (Question - can you guess why this is a problem?)

Imperial College London

Jargon busting

Supervised learning: Using data-labels as part of learning scheme

Unsupervised learning: Learning without data-labels.

Semi-supervised learning: Some data-points have labels. Others do not.

Reinforcement learning: Learning how to make sequences of decisions.

Imperial College London

Jargon busting - unbiased estimators

Most of the time we do not know some function $f(x)$, but can observe some values of X. The collection of these samples is called a data-set.

Often we are interested in estimating some statistic associated with some random variable from the observed samples. If we denote the observed set of samples by $S_X = \{x_1, ..., x_N\}$, then as estimate of some statistic $\hat{\theta}(S_X)$ is said to be unbiased if:

$$\mathbb{E}(\hat{\theta}(S_X)) = 0. \tag{6}$$

Imperial College London

Jargon busting - Max and Arg Max

Very often in Machine Learning problems we are given a set of values: $\mathcal{A} = \{a_1, ..., a_n\}$.

Given some function $f(\cdot)$ we are interested in the element that gives rise to the largest value when this function is evaluated for all elements in the set.

We denote this element that gives rise to this maximum by:

$$\arg\max_{a \in \mathcal{A}} f(a)$$

Imperial College London

22 January 2024

The key ideas in machine learning in one lecture

Imperial College London

Lecture overview

- Linear regression
- What's gradient descent?
- Building a simple predictor
- Building a simple classifier

Imperial College London

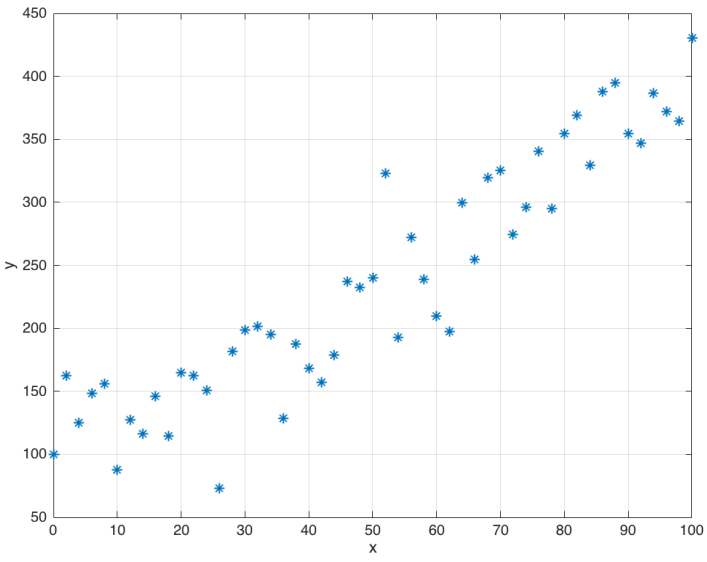
22 January 2024

Linear regression

Imperial College London

Lets start with some data

Every machine learning problem starts with some data.



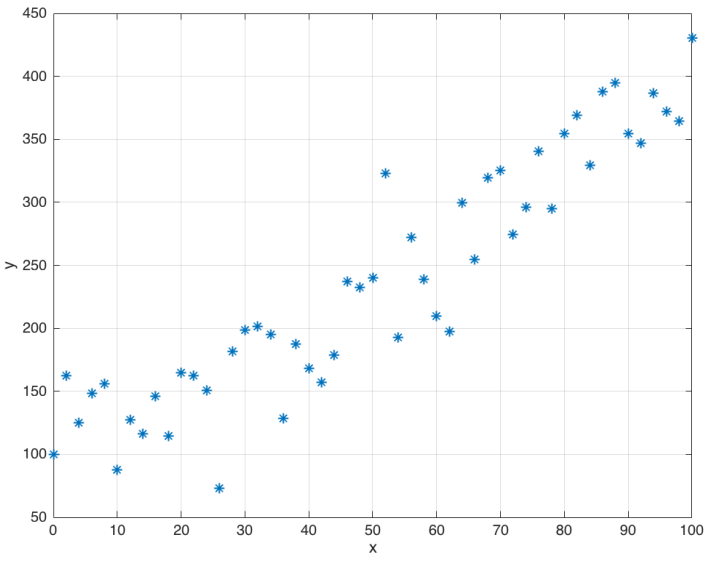
For example, each $x(i)$ could be an amount spent on advertising, and $y(i)$ is yearly revenue.

Say we now want to predict the return on investment if we increase $x(i)$.

Imperial College London

Lets start with some data

Every machine learning problem starts with some data.



Eyeballing the data, it looks like there is a linear relationship between each $x(i)$ and $y(i)$. That is:

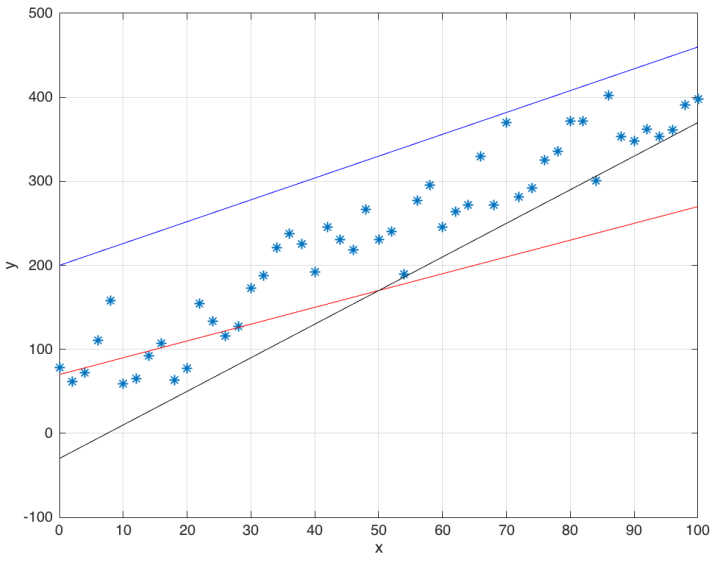
$$y(i) = mx(i) + c \qquad (1)$$

might be a good representation of the data.

Imperial College London

Lets start with some data

Every machine learning problem starts with some data.



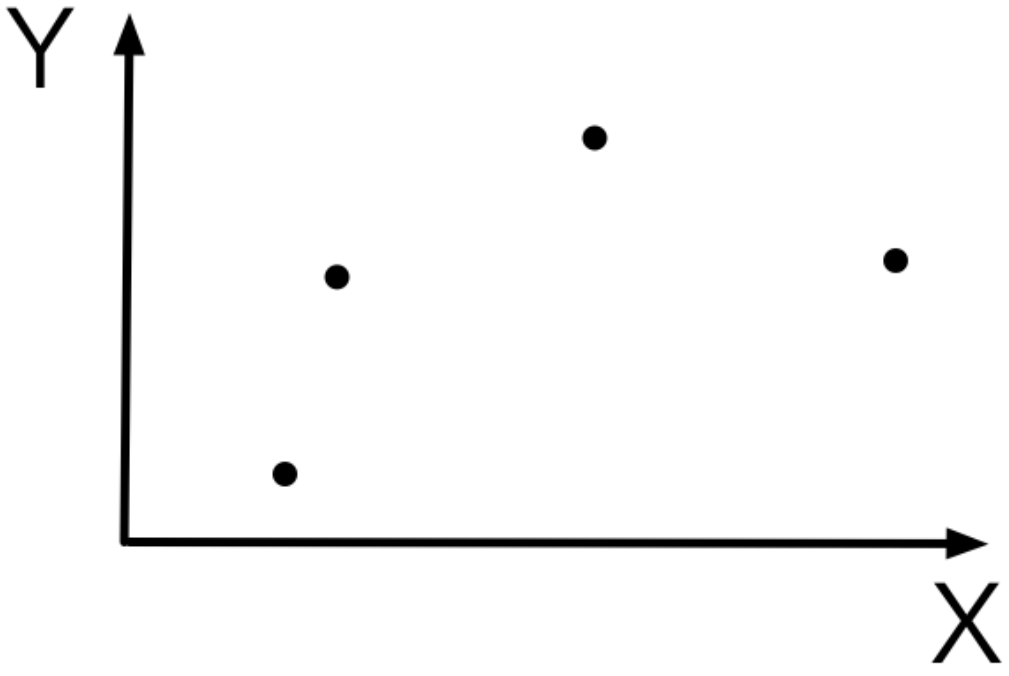
OK. That seems like a good idea:

$$y(i) = mx(i) + c \tag{2}$$

How do we find m and c?

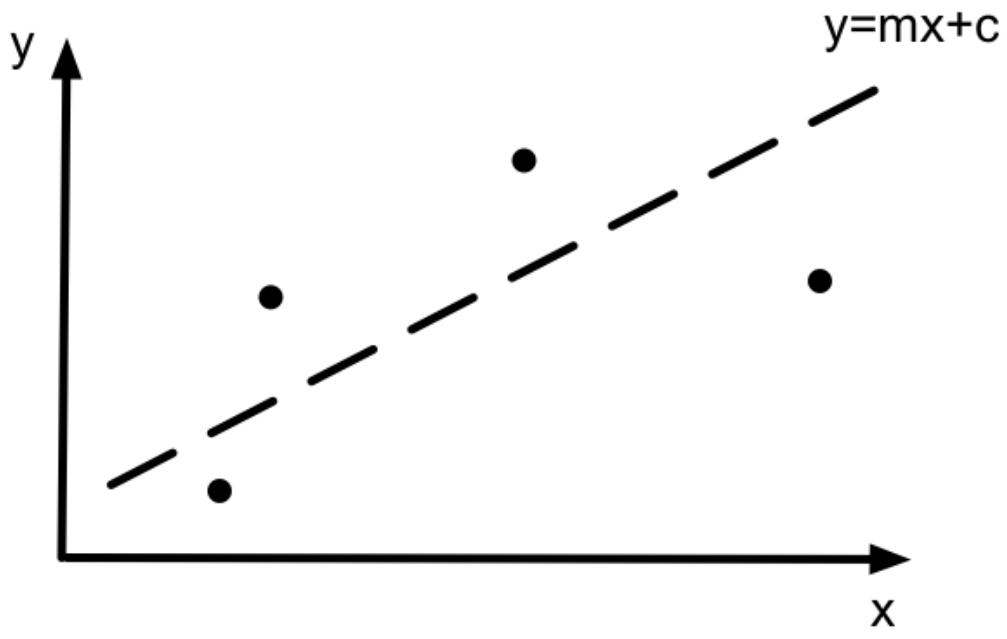
Imperial College London

Optimization enters the picture



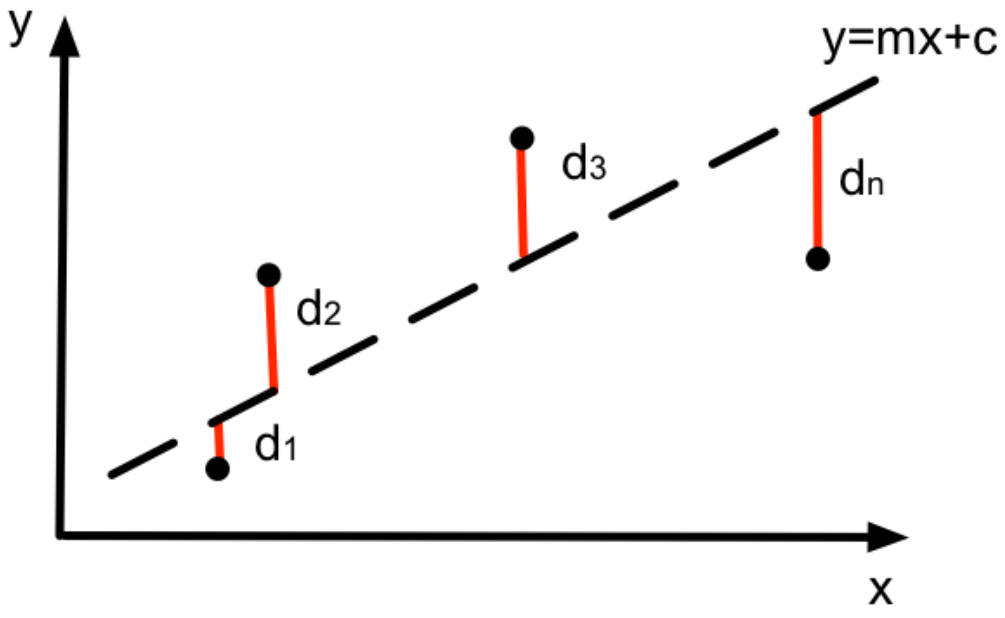
Imperial College London

Optimization enters the picture



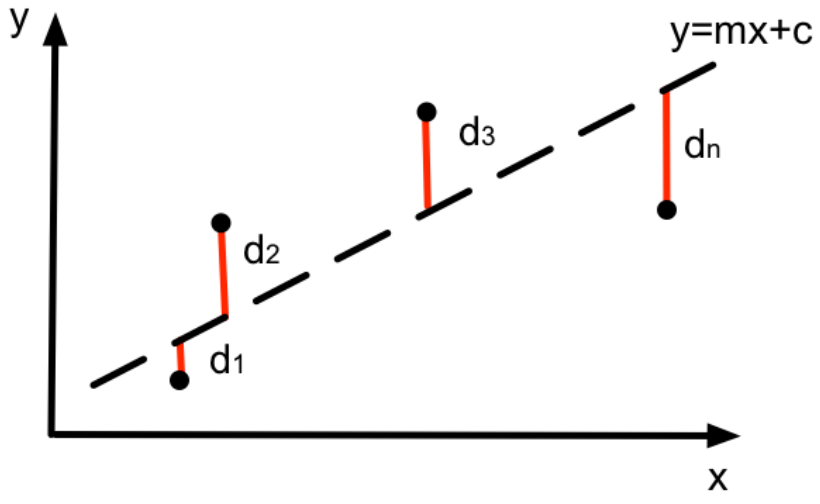
Imperial College London

Optimization enters the picture



Imperial College London

Optimization enters the picture



Lets formulate an optimization by minimizing the sum of the squares.

$$J(m, c) = d_1^2 + d_2^2 + ... + d_n^2 \quad (3)$$

Seems reasonable, right?

Optimization enters the picture

More formally, given n data points, we form the *cost function*

$$J(m, c) = \sum_{i=1}^{n} (mx(i) + c - y(i))^2 \tag{4}$$

We seek to minimize $J(m, c)$ to find the line that best represents the given data. This is called a **least squares problem.**

Question : Who said minimizing the sum of the squares is a good idea? Well, it isn't always (Explain). But sometimes it is.

Imperial College London

Back to our data

What does the cost function look like for our data set?

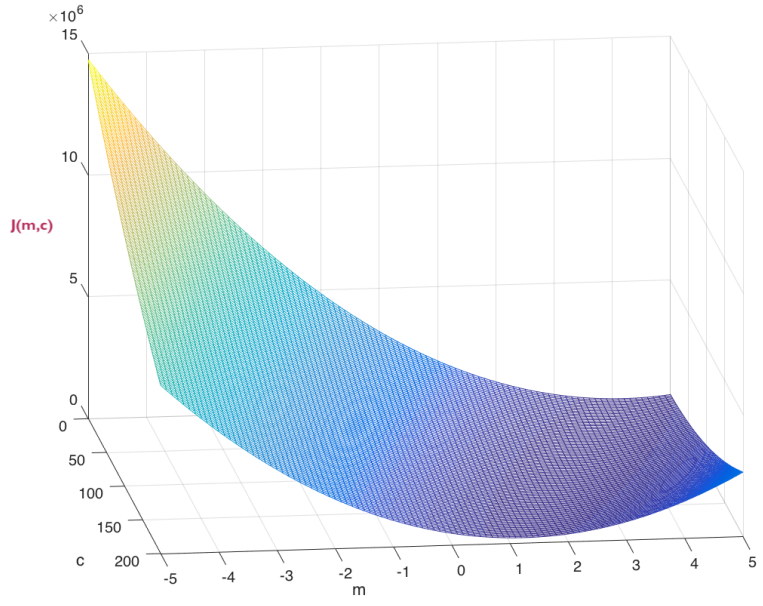


Figure 1: $J(m, c)$ as a function of m and c.

Well, it's a convex function so it should be easy to find the minimum?

Imperial College London

Back to our data

There are many methods to find the minimum of this function.

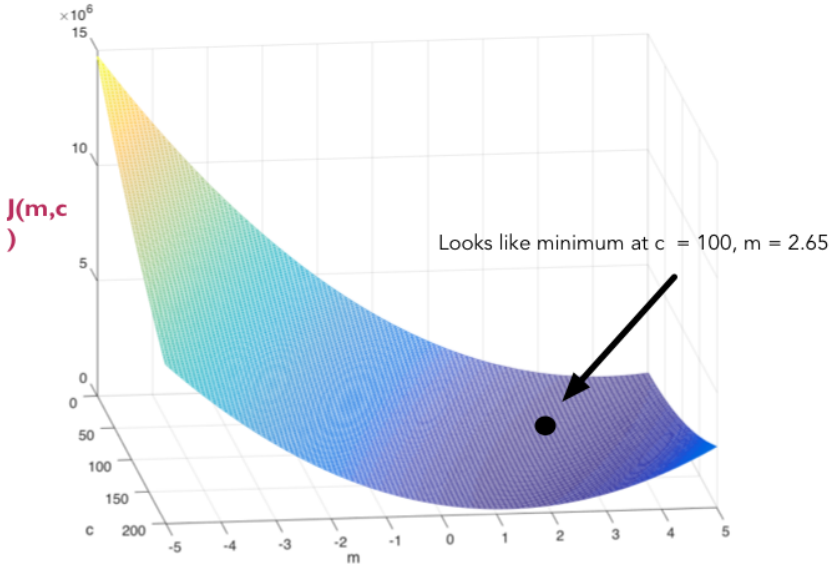


Figure 2: $J(m, c)$ as a function of m and c.

For the case of 2 parameters, we can just plot $J(c, m)$ and pick out the minimum. But for higher dimensional problems, we need other

Finding the optimal point

Our cost function is very simple. We can just write down $J(m, c)$ and use the results from our optimization course to find the optimal m, c (lets call them m^*, c^*).

It turns out solving linear regression problems using least squares is very simple. The easiest way to solve these problems is to gather everything in vector form.

Let's start by defining the vector of parameters as:

$$\theta^T = (m, c). \tag{5}$$

Finding the optimal point

Now lets gather together the observations:

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(n) \end{bmatrix}, \quad U = \begin{bmatrix} x(1), 1 \\ x(2), 1 \\ \vdots \\ x(n), 1 \end{bmatrix} \tag{6}$$

If our guess of a linear relationship between y and x is correct then we have:

$$Y = U\theta \tag{7}$$

where U is sometimes called the *regressor matrix*.

Finding the optimal point

Assuming

$$Y = U\theta \tag{8}$$

,

If we multiply both sides of the equation by U^T, we get: follows that

$$U^T Y = U^T U\theta. \tag{9}$$

If in addition we assume that UU^T is invertible, we can write

$$(U^T U)^{-1} U^T Y = \theta \tag{10}$$

Imperial College London

Time for reflection

It turns out that

$$(U^T U)^{-1} U^T Y \quad = \quad \theta \tag{11}$$

yields the optimum c^*, m^*. However, we have made some VERY BIG assumptions.

1. A linear relationship between the $x(i)$ and the $y(i)$.
2. We squared the sum of the distances between the line and the $y(i)$. Can you think of when this makes sense?
3. We assumed $U^T U$ is invertible? What does this mean?

Back again to our data

Let try out our formula on the data from the example:

$$U^T U = \begin{bmatrix} 171700 & 2550 \\ 2550 & 51 \end{bmatrix}$$

$$U^T Y = \begin{bmatrix} 72246 \\ 1196 \end{bmatrix}$$

and $\theta = (U^T U)^{-1} U^T Y$. Hence $\theta^T = (2.74, 100.3)$.

Interlude - What just happened?

You might ask - where did that all come from? Well, the cost can be written in matrix form as:

$$J(m,c) \quad = \quad (Y - U\theta)^T (Y - U\theta) \tag{12}$$

Taking derivatives and setting to zero we have:

$$\frac{\partial J}{\partial \theta} \quad = \quad -2U'Y + 2U'U\theta = 0 \tag{13}$$

which yields $U'U\theta = UY$ or

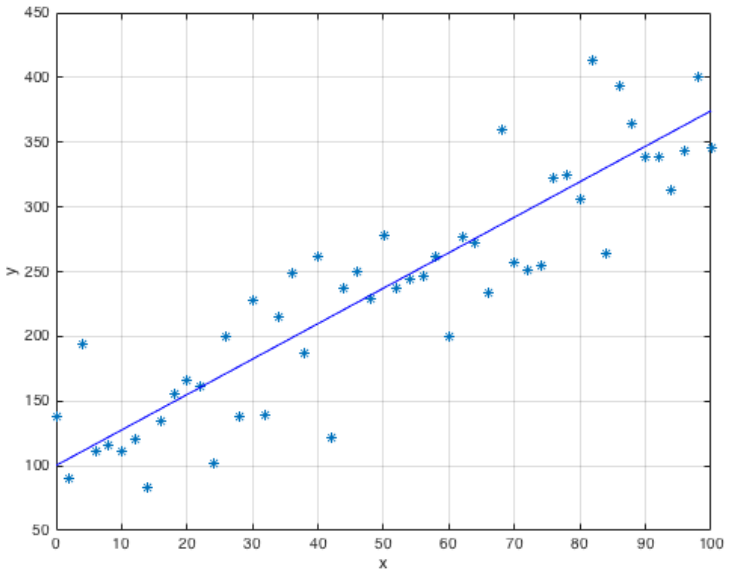
$$\theta = (U^T U)^{-1} U^T Y \tag{14}$$

when the inverse exists.

Imperial College London

Back to our data

Here is the line we just found:



An important remark

Linear regression is really powerful. It can also be used to build non-linear models. Suppose we postulated a model of the form:

$$y(i) = mx(i)^2 + c. \tag{15}$$

We can still use linear regression to find m and c because the unknown parameters appear in a linear way in the equation. In this case we have as before:

$$Y = \begin{bmatrix} y(1) \\ \vdots \\ y(n) \end{bmatrix}, \quad U = \begin{bmatrix} x(1)^2, 1 \\ \vdots \\ x(n)^2, 1 \end{bmatrix}, \quad Y = U\theta \tag{16}$$

Everything else is the same.

Gradient descent

Linear regression is really useful when unknown parameters enter into an equation in a linear manner.

If this does not happen, then we need to do something else.

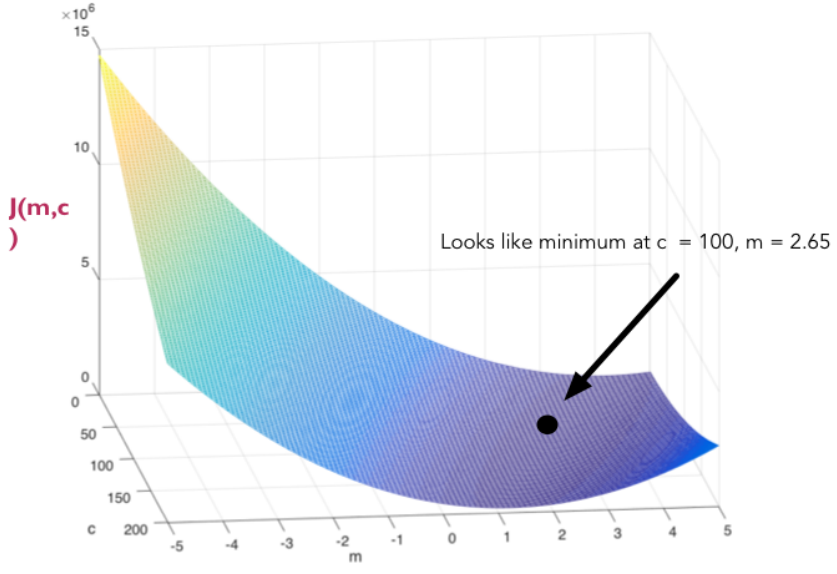
A very common technique is to use gradient descent. Again the idea is very simple.

Lets look again at the cost function for our problem.

Imperial College London

Gradient descent

What does the cost function look like for our data set?



Well, it's a convex function so it should be easy to find the minimum?

Gradient descent

One approach would be to simply *guess* a value of m and c, called them m_0, c_0, and calculate the gradient of $J(m, c)$ at that point. Recall, the gradient is the vector:

$$(\frac{\partial J(m,c)}{\partial m}, \frac{\partial J(m,c)}{\partial c}) \tag{17}$$

evaluated at m_0, c_0. We can then use these to update our estimates of the parameters.

Gradient descent

Recall, the gradient is the vector:

$$(\frac{\partial J(m,c)}{\partial m}, \frac{\partial J(m,c)}{\partial c}) \tag{18}$$

evaluated at m_0, c_0. We can then use these to update our estimates of the parameters using the rule:

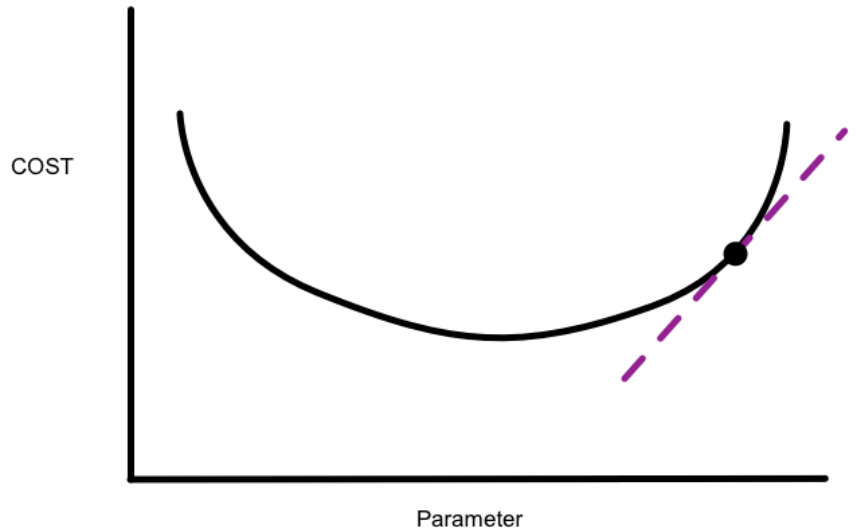
$$m_1 = m_0 - \alpha_m \frac{\partial J(m,c)}{\partial m} \tag{19}$$

$$c_1 = c_0 - \alpha_c \frac{\partial J(m,c)}{\partial c} \tag{20}$$

where α_c, α_m are small constants. We can repeat this over and over again to find our parameters.

Imperial College London

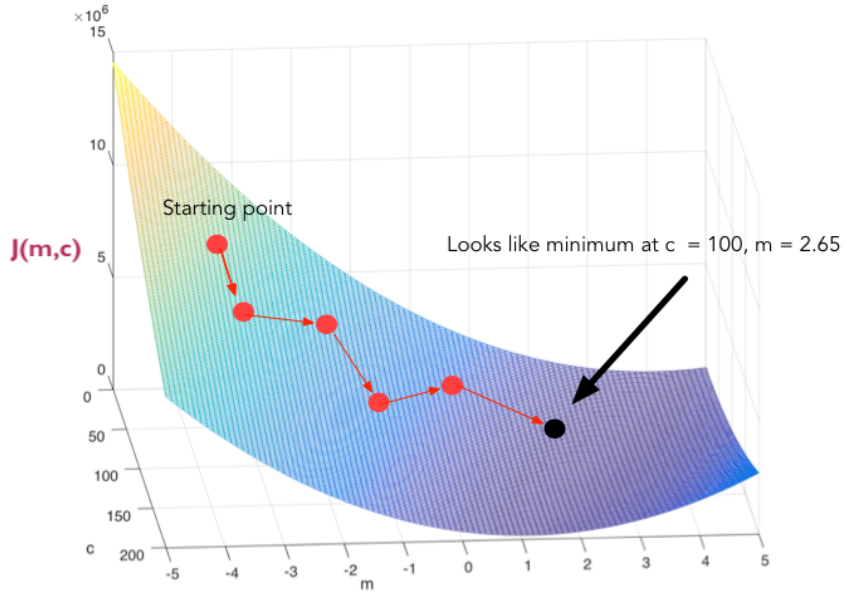
Gradient descent



Imperial College London

Gradient descent

We can then repeat this strategy and walk along the cost function to the minimum as shown below.



Well, it's a convex function so it should be easy to find the minimum?

Gradient descent

Gradient descent is widely used in the machine learning community.

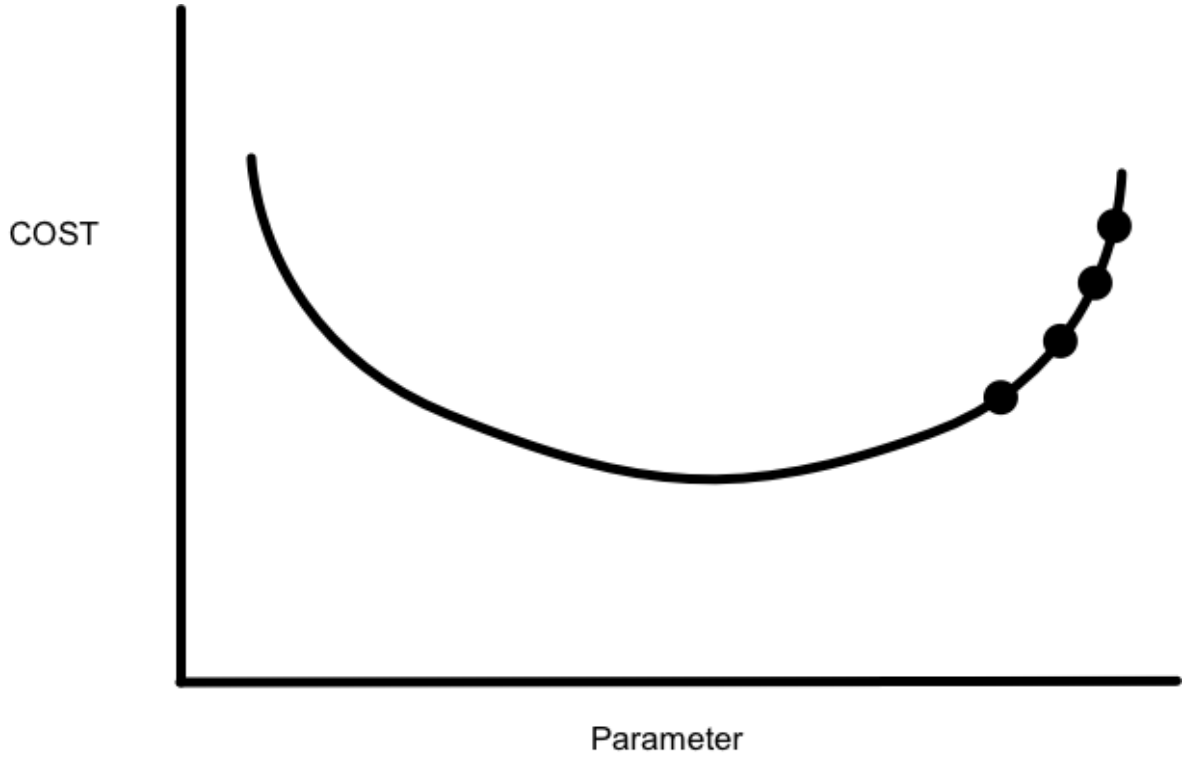
There are many issues? How does one cope with non-convexity, local minima. How does one select the parameters of the walk (α_m, α_c). When does one stop.

That said, the idea is really simple. And its all captured in the previous few slides.

Note also that all the ideas presented carry over to models with more parameters.

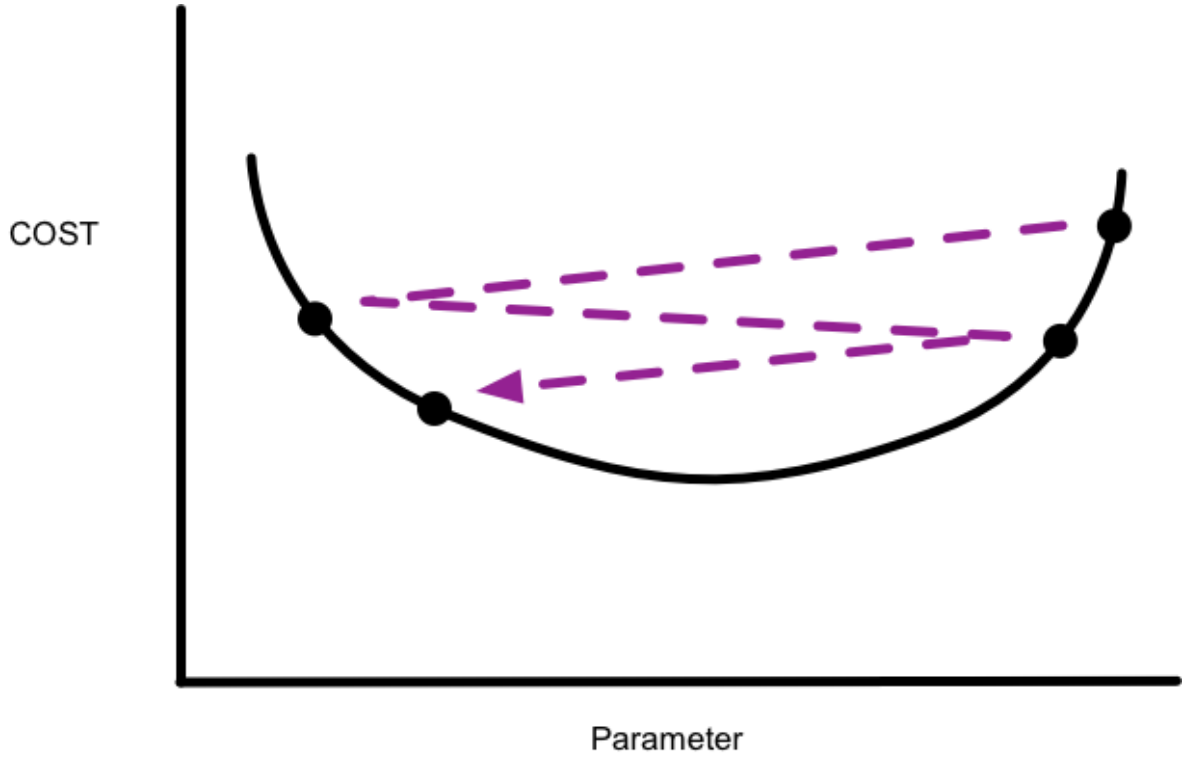
Imperial College London

Gradient descent in practice



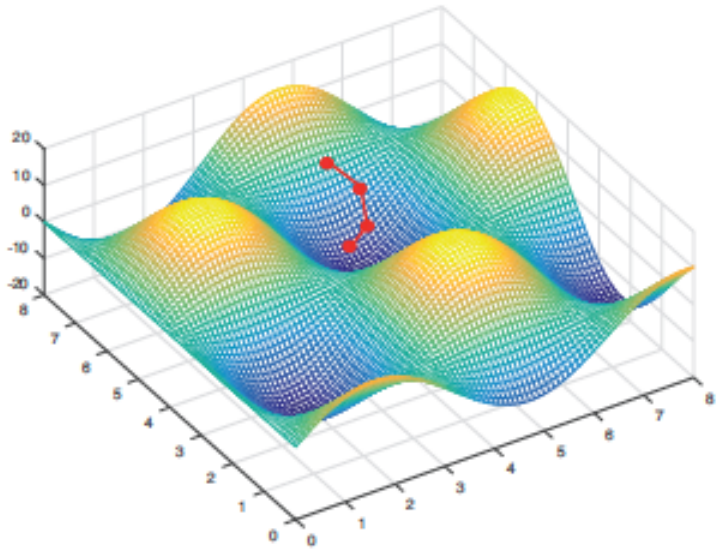
Imperial College London

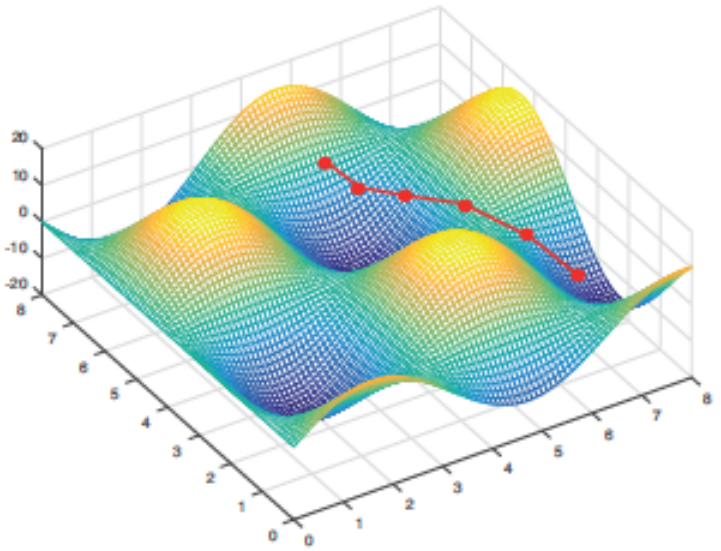
Gradient descent in practice



Imperial College London

Gradient descent in practice





Imperial College London

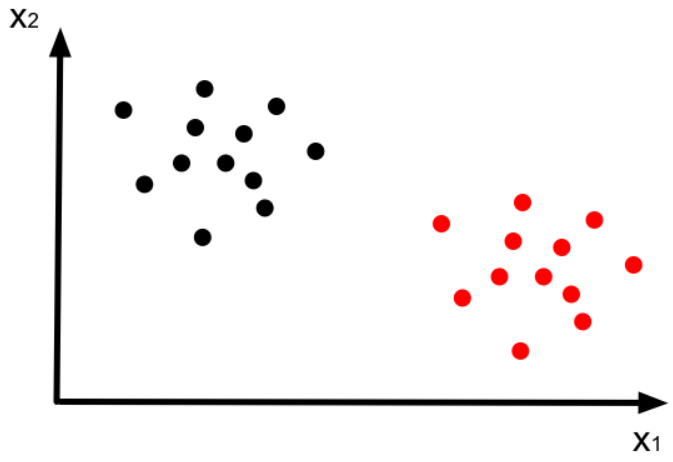
22 January 2024

A simple classifier

Imperial College London

Classification

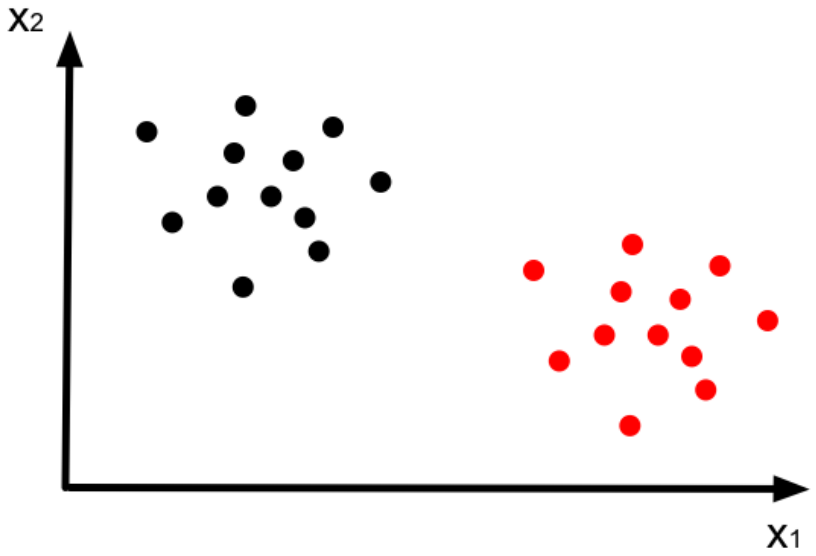
Suppose now we wish to build a simple classifier.



As in the regression case, the story starts with data. In this case we have features, (x_1, x_2) and labels y that indicate which features belong to which class.

Imperial College London

Lets start with some data

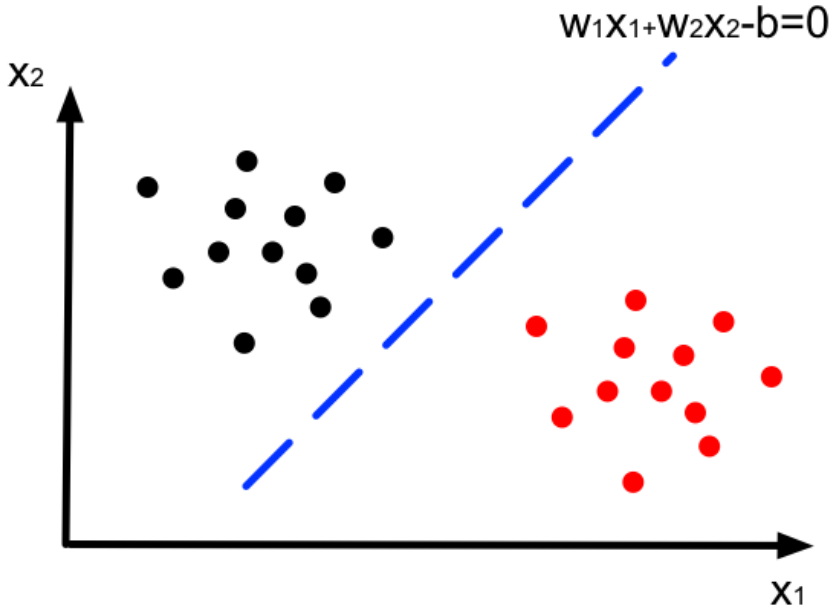


For example, x_1 might be the age of a driver, x_2 the time since passing a driving test, and the label might be the likelihood of speeding.

Our objective is to build a classifier that can predict whether someone might speed in the near future.

Imperial College London

Classification



As before, eyeballing the data seems to indicate that the *classes* are linearly separable. That is, we can fit a line (or hyperplane) between them, and use the fact that a feature is either side of the line to predict class membership.

Classification

The goal, as in linear regression, is to find the hyperplane:

$$w_1 x_1 + w_2 x_2 - b \tag{21}$$

that separates the two classes. How do we do that? Well, suppose we can find such a hyperplane, then for all black points

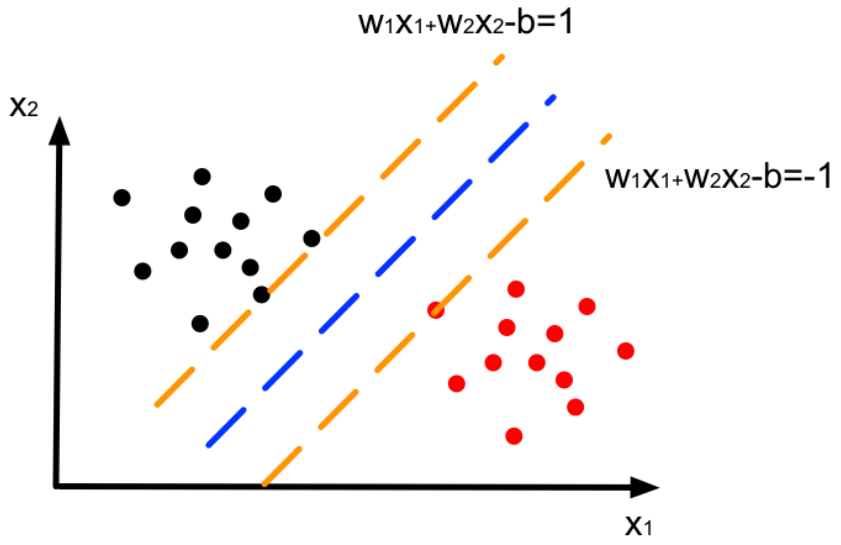
$$w_1 x_1 + w_2 x_2 - b > 0, \tag{22}$$

with all red points negative.

Imperial College London

Classification

The optimization is usually set up as follows.



We try to find a function $f(x) = sign(w_1^* x_1 + w_2^* x_2 - b^*)$ that correctly labels old and new samples.

Classification

The optimization attempts to minimize $w_1^2 + w_2^2$ subject to the constraint that

$$w_1 x_1(i) + w_2 x_2(i) - b \geq 1 \text{ if } y(i) = 1 \quad (23)$$

$$w_1 x_1(i) + w_2 x_2(i) - b \leq 1 \text{ if } y(i) = 1 \quad (24)$$

The minimization of $w_1^2 + w_2^2$ maximizes the distance between the two hyperplanes in the previous diagram. Can you explain why?

Imperial College London

27 December 2021

Getting our hands dirty - real life machine learning

Imperial College London

Lecture overview

- Anatomy of a machine learning algorithm
- Coding our first learning algorithm
- Some practical tips on machine learning

Imperial College London

27 December 2021

Calculating gradients

Anatomy of a machine learning algorithm

Roughly speaking, every machine learning algorithm has the following steps

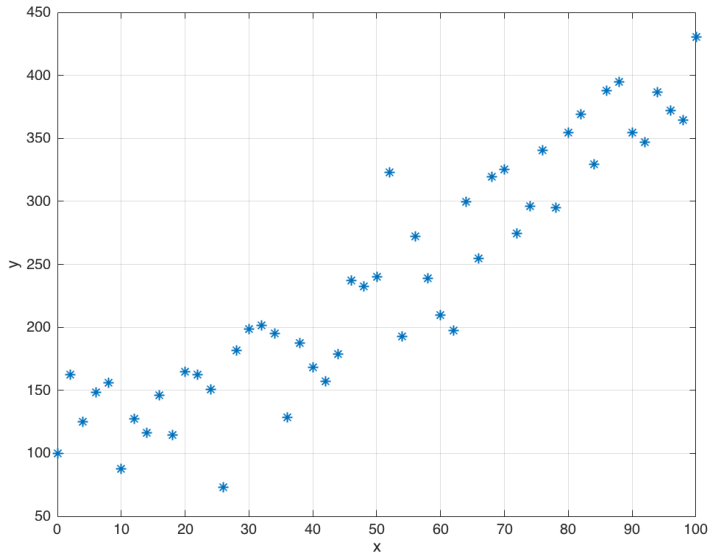
- A cost function
- Some optimization criterion based on the loss function
- An algorithm to implement the optimization

These are the building blocks of pretty much any machine learning algorithm.

Imperial College London

Gradient descent

Often, gradient descent is the method of choice to implement a specific optimization. Gradient descent is an iterative optimization algorithm for finding the minimum of a function.

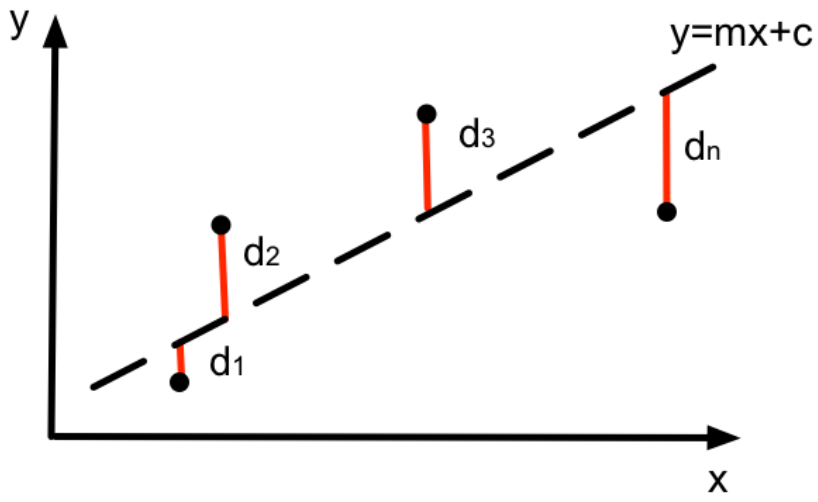


Lets go back to our data from the previous lecture. Recall we wish to find representation of the data in the form of a linear model.

$$y(i) = mx(i) + c \qquad (1)$$

Imperial College London

Revision from the last lecture



Lets formulate an optimization by minimizing the sum of the squares.

$$J(m,c) = \frac{1}{n}\left(d_1^2 + d_2^2 + ... + d_n^2\right) \quad (2)$$

Note here that I have divided the sum by n to stop the numbers from becoming very large.

Revision from the last lecture

More formally, given n data points, we form the *cost function*

$$J(m, c) = \frac{1}{n}\Big(\sum_{i=1}^{n}(mx(i) + c - y(i))^2\Big) \tag{3}$$

We seek to minimize $J(m, c)$ to find the line that best represents the given data.

Gradient descent

Gradient descent starts by calculating the partial derivative of every parameter.

$$\frac{\partial J(m,c)}{\partial m} = \frac{1}{n}\sum_{i=1}^{n} -2x(i)(y(i) - (mx(i) + c)) \tag{4}$$

$$\frac{\partial J(m,c)}{\partial c} = \frac{1}{n}\sum_{i=1}^{n} -2(y(i) - (mx(i) + c)) \tag{5}$$

Gradient descent then proceeds in *epochs*.

Gradient descent

An epoch consists of using the entire training set to update each parameter. At each epoch we update b and m using the partial derivatives.

$$m \leftarrow m - \alpha_m \frac{\partial J(m, c)}{\partial m} \tag{6}$$

$$c \leftarrow c - \alpha_c \frac{\partial J(m, c)}{\partial c} \tag{7}$$

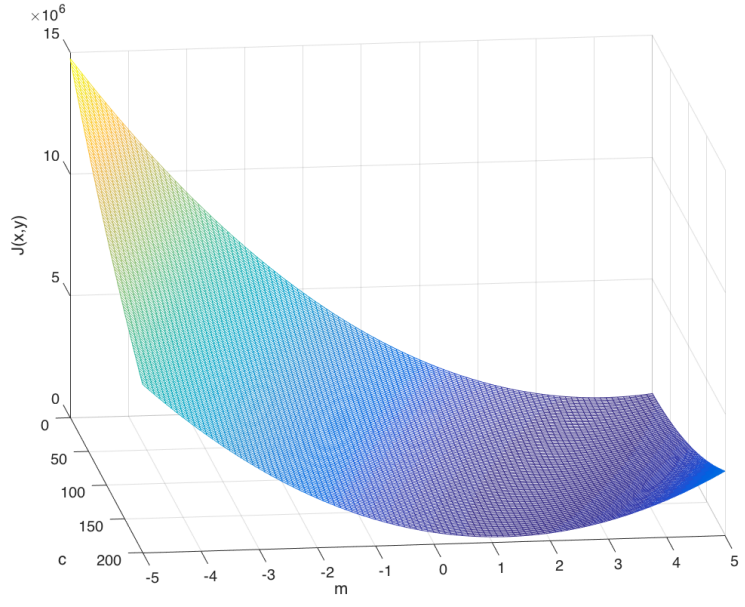
where α_c, α_m are small constants that control the rate of learning.

At the next update we recalculate the partial derivatives with the updates values of m and c, and we repeat until the parameters converge.

Imperial College London

Back to our data

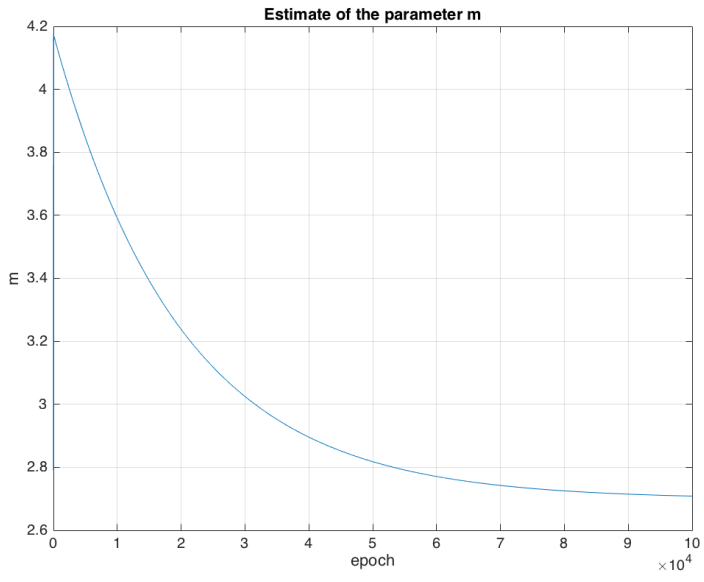
Our cost function looks something like?

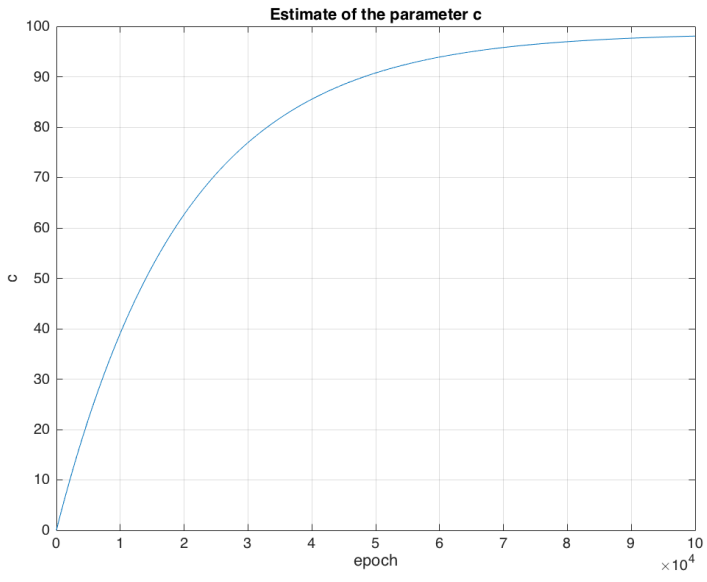


Its convex so we should be able to find the minumum.

Imperial College London

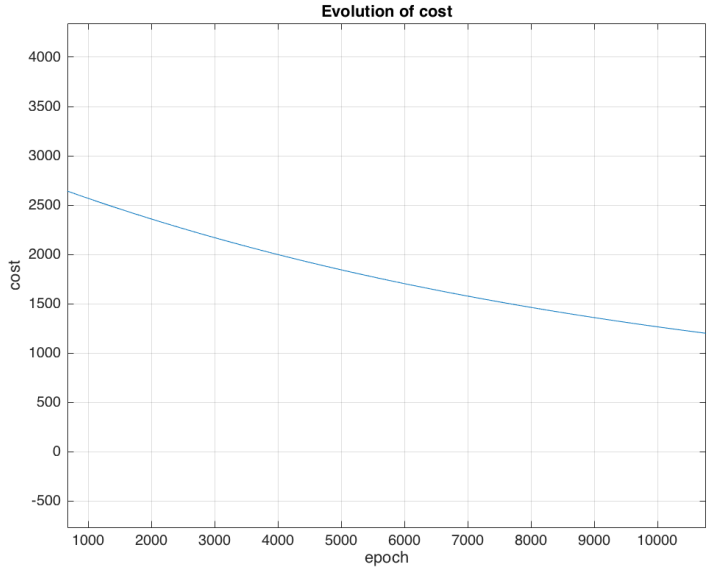
Evolution of parameters

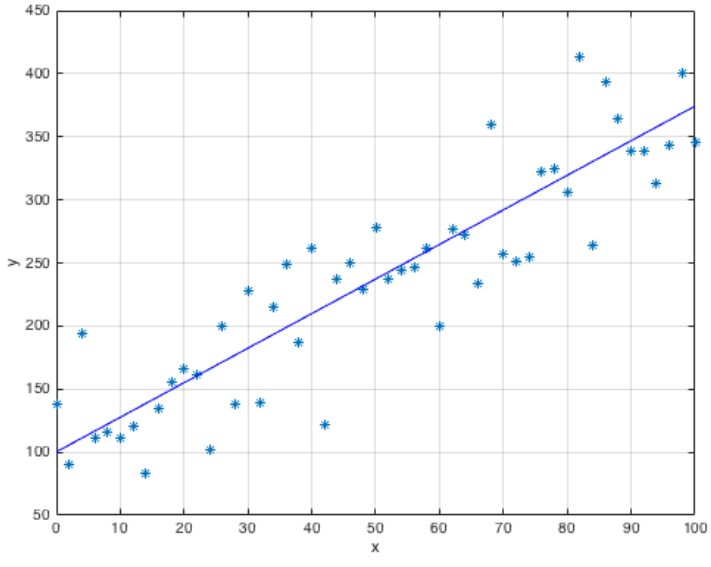




Imperial College London

Evolution of parameters





Imperial College London

27 December 2021

Some practical comments

Imperial College London

Practical comments

The truth about machine learning is that it is both an art and a science. In the past couple of lectures we have focussed on some of the "science".

But the reality is that we must take a lot of care with several aspects of the "problem" in order to get to the point of being able to effectively learn.

I want to take a few minutes to talk about some of these issues now.

Practical comments

Roughly speaking we need to consider three things.

- Getting the data ready for learning.
- Selecting features and the model.
- Choosing an appropriate learning algorithm.

Imperial College London

Getting the data ready

There are many steps in making a data set suitable for a machine learning algorithm.

As we have already discussed, at a very basic level, one must label a data set to make it suitable for machine learning algorithms. Often, this can be a very labour intensive process - think of the task of labelling images.

Sometimes, problems arise in which we must convert non-numeric labels in to numeric ones and vice-versa. Again, this is a task that must be undertake with great care.

Getting the data ready

A more common data curation technique is called *data normalisation*. Normalisation is necessary to avoid having different scales in the training data. For example, suppose we postulate a model of the form:

$$y = ax_1 + bx_2 + c; \tag{8}$$

namely, the output depends on two input features x_1, x_2. suppose further that x_1 is in the interval $[0.1, 0.7]$ and x_2 is in the interval $[-10000, 20000]$.

Situations like this almost always cause problems for machine learning algorithms. Often the lead to numeric problems, or just make the choice of learning parameters difficult.

Getting the data ready

Here the basic idea it to convert the actual range which a numerical technique can take to be in a standard interval; say $[0, 1]$. That is:

$$\tilde{x}(j) \leftarrow \frac{x(j) - min_j x}{max_j x - min_j x} \tag{9}$$

or in the interval $[-1, 1]$ with zero mean,

$$\tilde{x}(j) \leftarrow \frac{x(j) - \mu}{max_j x - min_j x} \tag{10}$$

where μ is the average of the $x(i)$.

Getting the data ready

Another problem that often arises concerns data with missing features. This often happens when a data set is hand-crafted. Typical approaches to dealing with missing data include:

- removing examples with missing features;
- using special learning algorithms that can deal with missing features;
- using a **data imputation** technique (for example - replacing the missing feature by an average value of the feature in the training set).

Data and learning

Another issue concerns dividing the data into *test and training* sets. Remember, our objective is to learn. In fact, up to now in all our examples, we have not really considered *learning*. What we have done is to fit a sets of data-points to a parameterised model.

This is not *learning*. To learn, we wish our models to be able to generalise. That is:

make predictions on data that the model has not seen before.

Data and learning

Usually we divide our data into 3 parts.

- a training set;
- a test set;
- a validation set.

Sometimes, we use only a training set and a test set.

Data and learning

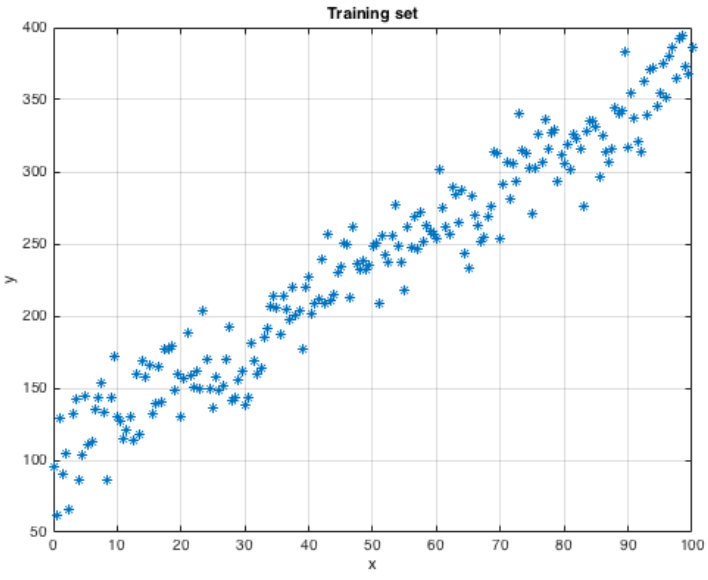
The meaning of these data sets is pretty self-explanatory.

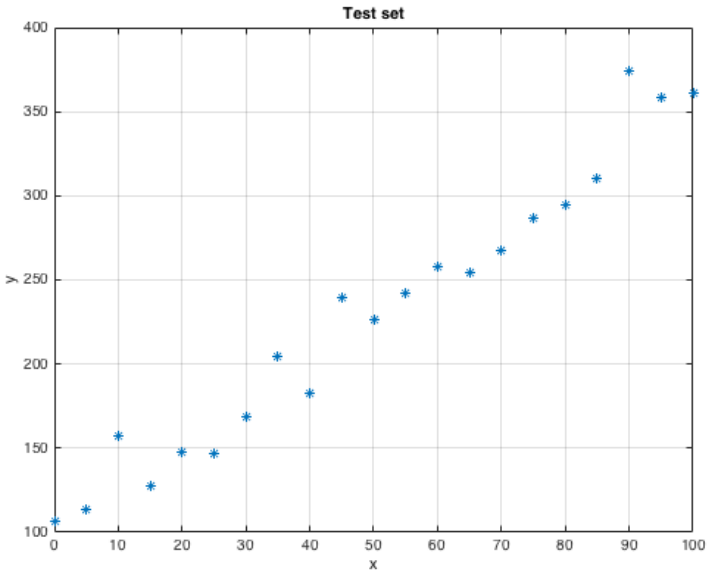
Roughly speaking, the training set is used to find the models parameters (using for example, gradient descent); the test set is to see how well the model has learned, and to detect over-fitting; and the validation set is to provide an independent verification of the *goodness* of the model.

How to split the original data set into these three sub-datasets, is a art rather than a science.

Imperial College London

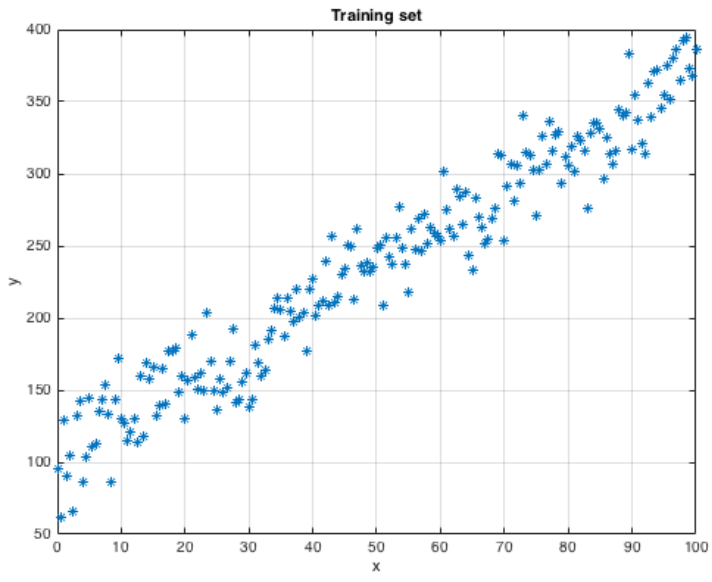
Data sets

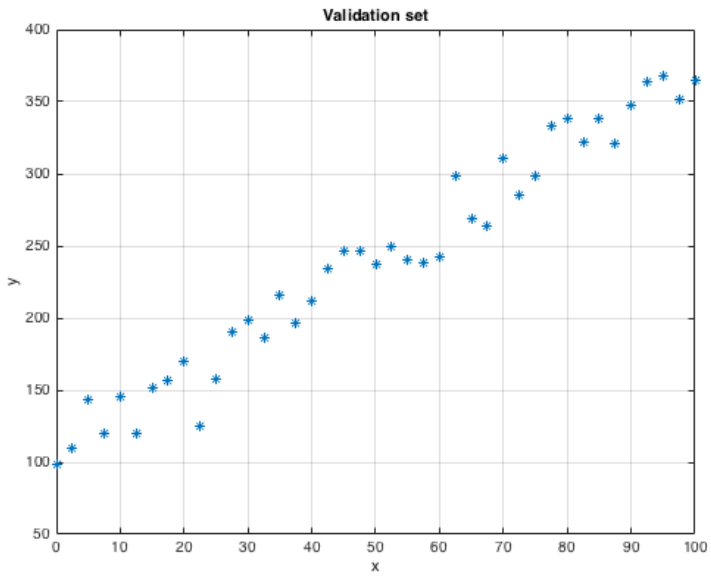




Imperial College London

Data sets





Model and feature selection

In the previous example we postulated a model to fit to the data of the form:

$$y = mx + c. \tag{11}$$

Why did we pick a model of this form? How do we know that this model is more appropriate than, for example:

$$y = ax^2 + bx + c. \tag{12}$$

Is'nt it a good idea to use richer model structures?

Model and feature selection

Is'nt it a good idea to use richer model structures? Well, surprisingly, not always.

- Too few parameters $\rightarrow$ Don't really capture what is going on.
- Too many parameters $\rightarrow$ We end up fitting noise.

In reality, both situations are bad, and leads to models with very poor ability to generalise.

Imperial College London

Model and feature selection

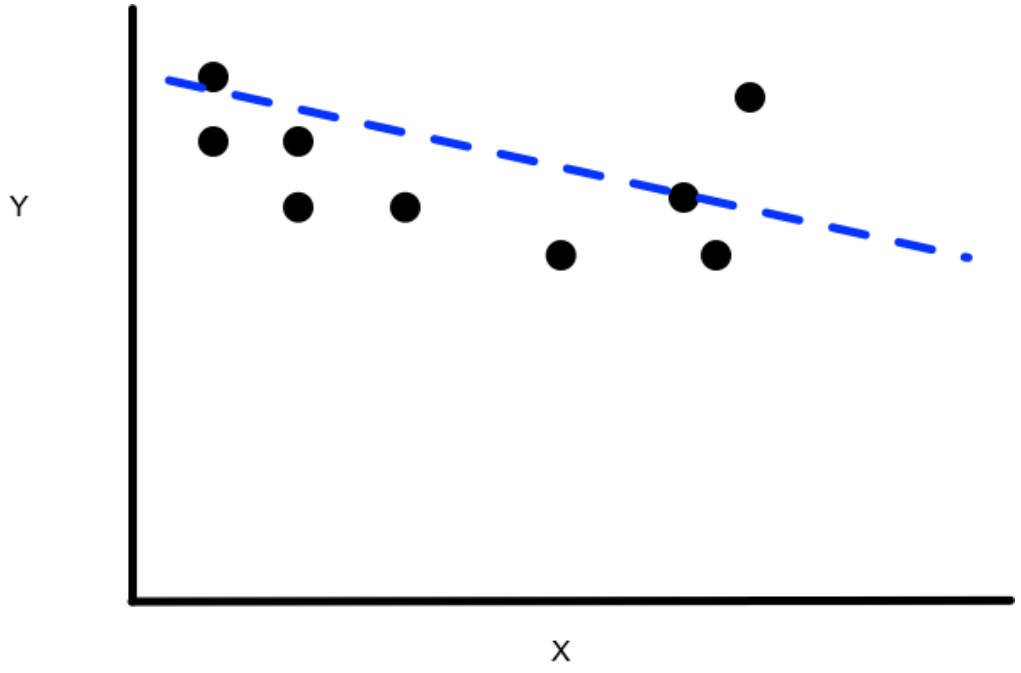


Figure 1: Underfitting

Imperial College London

Model and feature selection

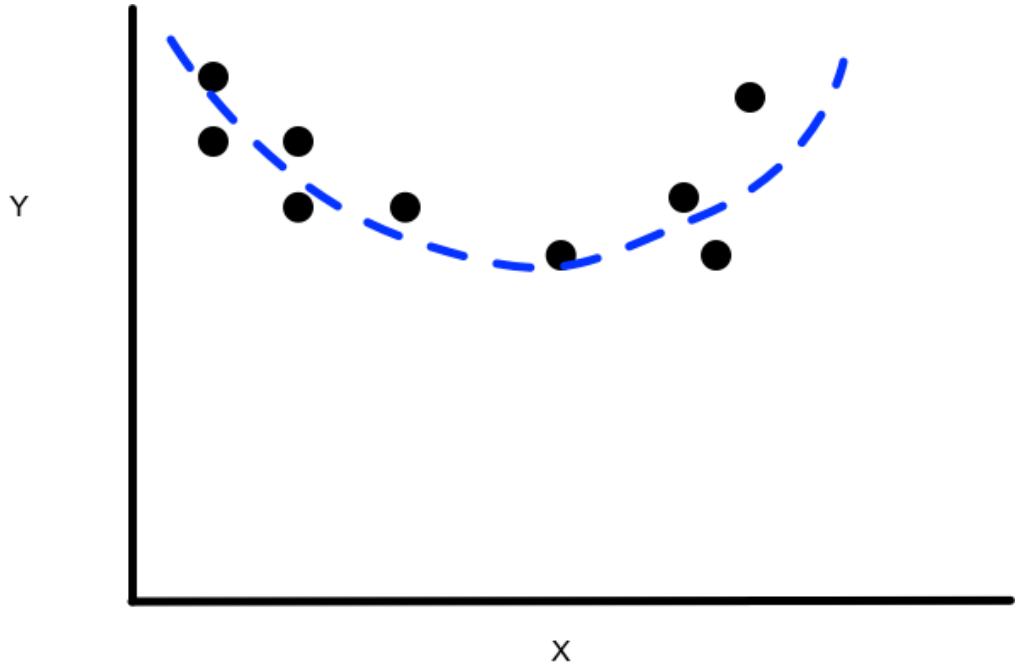


Figure 2: Good fit

Model and feature selection

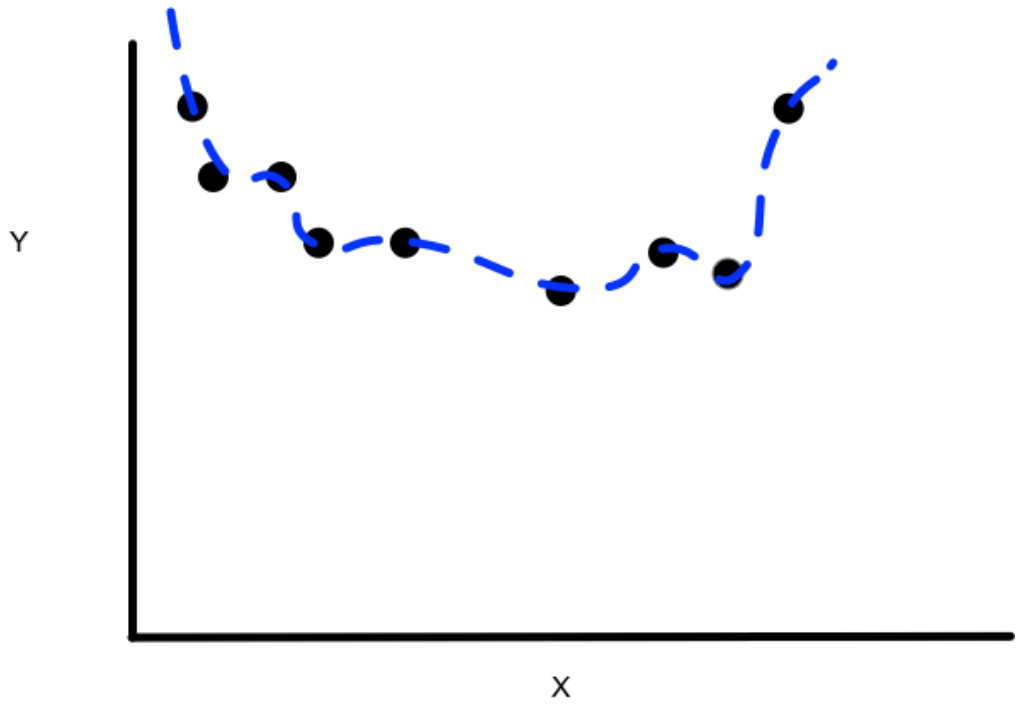


Figure 3: Overfitting

Overfitting and bias variance dilemma

In reality, selecting the structure for the model, and the features to predict labels, is usually based on prior information about the problem.

For example, in robotics, we often know the differential equation that describes the process.

Overfitted models ususally have *high variance*. Change the data, and the model parameters change a lot. Underfitted models usually have a *high bias*. That is, they make a lot of errors in the test set. Striking a balance between the two is called the **bias-variance** tradeoff.

Regularisation

There are many solutions to the problem of overfitting. Perhaps the one that is most widely used is known as *regularisation*.

Regularisation is an umbrella term that constrains the model parameters: in other words, it forces the learning algorithm to find a less complex model.

Suppose, given a data-set, we postulate a model of the form: $y = f(x, \theta)$, where θ denotes the parameters of the model.

Regularisation

Up to now, we find the 'optimal', θ by finding the parameters that solves the following optimization:

$$\min_\theta \Big(\frac{1}{n} \sum_{i=1}^{n} \big(f(x(i), \theta) - y(i) \big)^2 \tag{13}$$

Regularisation involves adding another term to the optimization to constrain the parameters.

$$\min_\theta \Big(g(\theta) + \frac{1}{n} \sum_{i=1}^{n} \big(f(x(i), \theta) - y(i) \big)^2 \Big). \tag{14}$$

Regularisation

Typical choices of $g(\theta)$ are $\| \theta \|$ and $|\theta|$.

The former is called L2 regularisation, or *ridge regularisation*. L2 regularisation looks for small parameter vectors.

The latter is called L1 regularisation, or *LASSO regularisation*. L1 regularisation looks for sparse parameter vectors.

The choice of regularisation term is often based on prior knowledge. For example, we sometimes know that the parameters are constrained to be positive or lie in a certain interval. Such constraints can also be included as part of a regularisation term.

Selecting a learning algorithm

Another important issue concerns the choice of learning algorithm. We have already seen examples of *iterative* algorithms - gradient descent. and batch algorithms where all the data is used in one shot to find the optimal parameters: recall:

$$(U^T U)^{-1} U^T Y = \theta \tag{15}$$

There are many flavours of learning algorithm (there are even many flavours of gradient descent).

Selecting a learning algorithm

Selecting a learning algorithm is a really important part of any machine learning problem.

Selecting the most appropriate algorithm depends on many factors:

- hardware constraints;
- training speed;
- type of data and problem;
- any nonlinearities present;
- memory constraints.

Imperial College London

Contacting us

We are available at any time. Contact us at:

r.shorten@imperial.ac.uk

p.ferraro@imperial.ac.uk

Very happy to arrange 1:1 tutorials with a little notice.