Foundational Machine Learning
Lecture 1 - Introduction

30th September 2025

UNIVERSITY OF BATH

Logistics

- Taught by **Rohit Babbar** (rb2608@bath.ac.uk) and Wenbin Li (wl281@bath.ac.uk)
- Per week:
  - 2 lectures (some of the lectures will involve self-study with referenced materials such as video lectures)
  - 2 hour lab session (no labs in week 1)
- NOTE : Timetable `https://mytimetable.bath.ac.uk/schedule`:
  - There is no lecture this Friday (self-study material will be provided)
  - From next week onwards, the first of the two lectures will be on Thursdays in 5W 2.01
- Asking questions:
  - Live: Lectures and labs
  - Online: Moodle forum (`https://moodle.bath.ac.uk/`)
- Note: CS doesn't lecture in week 6! (consolidation/revision week)

UNIVERSITY OF BATH

Environment

- Everything on Moodle (or linked from Moodle)
- Lectures will be recorded, link will be added on Moodle within 24 hours
- Own laptop easier! (but keep backups)
- Labs:
  - Location : Library Level 4 GTA PC area
  - Time : Fridays 15:15 - 17:05
  - 2 tutors (our PhD students) will be there to help
  - NOTE : There is no Lab in Week 1

UNIVERSITY OF BATH

Assessment

- 3 labs in this semester – check moodle for schedule
- First (lab 0) is formative and optional

- Marks:
  - Coursework
    - Labs 1 & 2 with a weightage of 10% each are auto marked Jupyter notebooks
    - There will one more coursework (sometime in weeks 22-24 i.e., early March) in semester 2 with a weightage of 10%, exact format yet to decided
  - Exam
    - Pen & paper exam - 70%
    - Practice papers and past papers will be made available on Moodle

UNIVERSITY OF BATH

Auto-Marker

- **Auto marker**
  - The above linked is where labs are released and submitted (not Moodle)
  - You will be notified (via Moodle) when the labs are released
  - Instant marking!
  - Marks reduced with many submissions
- Can go wrong: ask if you have issues
- Has (invisible) plagiarism detection

UNIVERSITY OF BATH

Tool chain

- Using Python 3
- Within Jupyter or JupyterLab: Workbook interface in web browser
- Installation:
  - Easy option: `https://www.anaconda.com/`
    - Use Anaconda 3, not 2! (corresponds to Python version)
  - Linux: Can use `pip3` instead of Anaconda
  - Add machine learning libraries!
    - Good list: `https://hex.cs.bath.ac.uk/wiki/Libraries`

Jupyter

Jupyter Example - Mozilla Firefox

Home | Jupyter Example

localhost:8888/notebooks/Jupyter Example.ipynb

jupyter Jupyter Example Last Checkpoint: 8 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

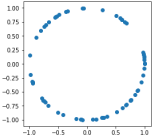
Run Code

Jupyter Example

This is inline documentation

In [1]:
```
%matplotlib inline

import numpy
import matplotlib.pyplot as plt
```

In [2]:
```
# This is a code block...

# Random points on circle...
points = numpy.random.standard_normal(size=(64,2))
points /= numpy.sqrt(numpy.square(points).sum(axis=1))[:,None]

# Plot...
plt.figure(figsize=(4,4))
plt.scatter(points[:,0], points[:,1])
plt.show()

# Output from a code block appears immediately below...
```



UNIVERSITY OF BATH

The primary goal if this unit is to ...

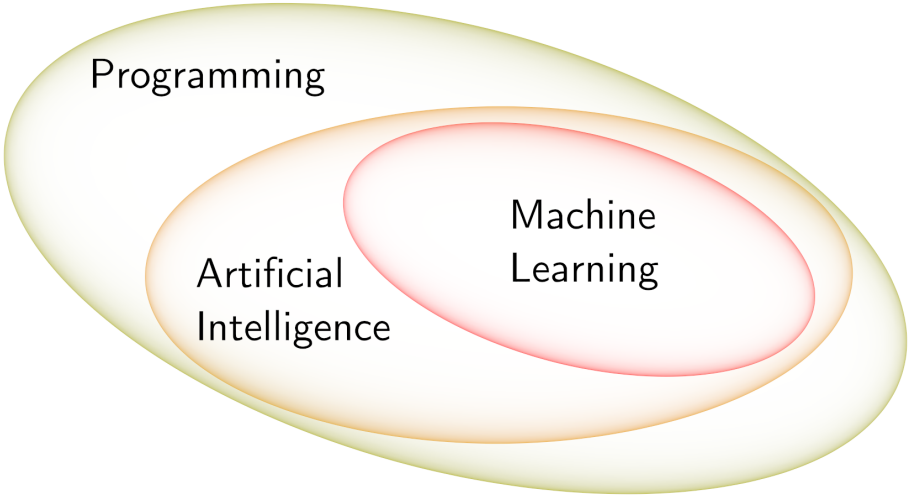
To learn

- Implementing machine learning algorithms
  (investigate what's going on under the hood. i.e. not just use off-the-shelf libraries such as Scikit Learn!)
- Designing machine learning algorithms
- Verifying/testing
- Optimisation

UNIVERSITY OF BATH

What is ML?

What is ML?



UNIVERSITY OF BATH

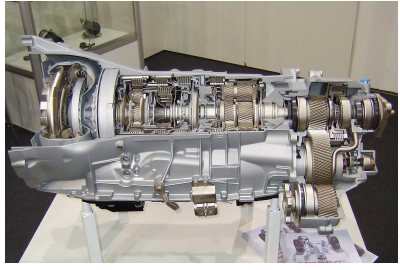
Imagine a car. . .

- Programming:

  Computer is an idiot –
  does exactly what you tell it to
  and nothing else!

- e.g. automatic gear box:

```
while True:
  if engine.revs > 5000 and
     transmission.gear < 5:
    clutch.disengage()
    transmission.gear += 1
    clutch.engage()
  elif engine.revs < 1000 and
       transmission.gear > 1:
    clutch.disengage()
    transmission.gear -= 1
    clutch.engage()
```



UNIVERSITY OF BATH

Imagine a car. . .

- Artificial Intelligence:

  Computer uses optimisation to find the best solution to a well defined problem.

- e.g. gps navigation (A* search algorithm [a] from 1960's):

```
graph.load_map('uk.h5')
graph.set_start('bath')
graph.set_end('bletchley')
route = graph.shortest_route()
```

[a] https://en.wikipedia.org/wiki/A*_search_algorithm

Imagine a car...

- Machine Learning:
  Computer learns from examples (data) and generalises to all inputs.
- e.g. recognising road signs:

```
model = Recogniser('15mph_sign.h5')
while True:
  if model.search(camera.image()):
    engine.target = 6.7 # m/s
```

- How can this go wrong?

What is ML?

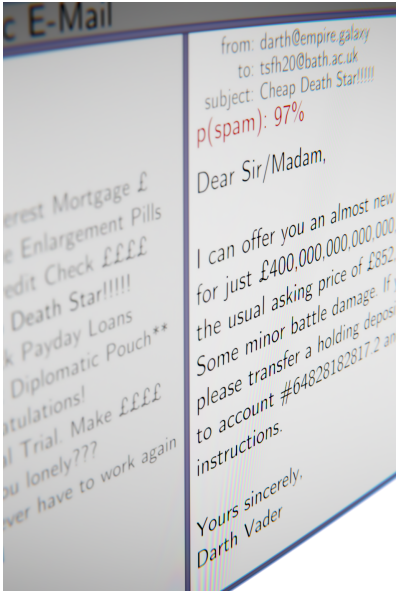
- **Learning from data**

What is ML?

- **Learning from data**
- Built on (you also need to understand):
  - Maths, especially probability
  - Optimisation
  - Programming

What can you do with it?

Motivation: Text



- Spam filtering
- Organising content (e.g. classifying news by area)
- Search engines (e.g. classifying if a web page is accurate)
- Monitoring social media
  (e.g. what do consumers think of your product)
- Automatic essay marking

Motivation: Financial

- Detecting fraud (e.g. spotting unusual spending patterns)
- Mortgage applications
  (i.e. calculating the probability of an applicant defaulting)
- Algorithmic trading [a]
  (e.g. predicting the future of the stock market)

[a] Salary of 200,000£ for the right candidate with a good blend of ML + programming + Maths/Stats
`https://www.levels.fyi/en-gb/companies/the-voleon-group/salaries/data-scientist?country=254`

Motivation: People

- Advertising [a]
  (e.g. choosing which advert to display on a website)
- Polling (e.g. estimating how a country will vote in an election)
- Sentiment analysis on social media
- Forensic accounting
- Detecting terrorism (e.g. spotting left bags at an airport)

[a] https://amazon.jobs/content/en-gb/teams/advertising

Motivation: Robots

- Walking robots
- Autonomous vehicles [a]
- Playing games (e.g. Deep Mind's Alpha Go)

[a] https://wayve.firststage.co/jobs?location=London%2C+United+Kingdom

UNIVERSITY OF BATH

The process

1. Choose a **problem**
2. Obtain required **data**
3. Choose or design a **model**
4. Fit model to data using **optimisation**
5. **Measure** performance

(there are variants. . .)

Process example: Mortgage

1. Choose a **problem**.

Identify if an applicant will pay back their mortgage based on their application.

2. Obtain required **data**.

Examples of past mortgage applications, plus if they paid the mortgage back or not.

3. Choose or design a **model**.

Output is **yes** or **no** – this requires a **classification** model.

UNIVERSITY OF BATH

Process example: Algorithmic Trading

1. Choose a **problem**.

Deciding which stocks to buy/sell on the stock market.

2. Obtain required **data**.

History of stock prices.

3. Choose or design a **model**.

Output is the future value of a stock – this is called **regression**.

Temporally varying data suggests a **time series** model.

Process example: Recommendation

1. Choose a **problem**.

Identify which products to suggest to a user on a commercial website.

2. Obtain required **data**.

Lots of users and everything they have bought.

3. Choose or design a **model**.

Output is the probability of buying every product – this requires a **probabilistic classification** model.

UNIVERSITY OF BATH

1. Choose a problem

e.g. this toy problem:

Given something in the ocean identify if it is a **fish** or an **invertebrate**

UNIVERSITY OF BATH

1. Choose a problem

e.g. this toy problem:

Given something in the ocean identify if it is a **fish** or an **invertebrate**

Input: Yes/no answers to questions such as:
Does it have teeth?

Output: Fish or invertebrate

UNIVERSITY OF BATH

2. Obtain required data

Animal name	bass	clam	carp	crab	catfish	crayfish	chub	lobster
Does it have teeth?	1	0	1	0	1	0	1	0
Does it breathe?	0	0	0	0	0	0	0	0
Does it have a backbone?	1	0	1	0	1	0	1	0
Is it aquatic?	1	0	1	1	1	1	1	1
Does it have a tail?	1	0	1	0	1	0	1	0
Is it a predator?	1	1	0	1	1	1	1	1
Is it an invertebrate?	0	1	0	1	0	1	0	1

- Use of 1 for "yes" and 0 for "no" is typical
- This module will be ignoring real data collection

3. Choose or design a model

- This is a **classification** problem – output is a discrete label
- One of the two main problem types in machine learning

UNIVERSITY OF BATH

3. Choose or design a model

- This is a **classification** problem – output is a discrete label
- One of the two main problem types in machine learning

- Hundreds of models for solving it
- Lets use another "model": A rule (algorithm) created by a human!

UNIVERSITY OF BATH

4. Fit model to data using optimisation

You have three minutes to come up with an algorithm:

Animal name	bass	clam	carp	crab	catfish	crayfish	chub	lobster
Does it have teeth?	1	0	1	0	1	0	1	0
Does it breathe?	0	0	0	0	0	0	0	0
Does it have a backbone?	1	0	1	0	1	0	1	0
Is it aquatic?	1	0	1	1	1	1	1	1
Does it have a tail?	1	0	1	0	1	0	1	0
Is it a predator?	1	1	0	1	1	1	1	1
Is it an invertebrate?	0	1	0	1	0	1	0	1

Write your algorithm down!

UNIVERSITY OF BATH

5. Measure performance I

- Previous slide was a **training set**
- Below is a **testing set**:

Animal letter Animal name	A	B	C	D	E	F	G
Does it have teeth?	1	0	0	1	1	0	1
Does it breathe?	0	0	1	0	0	0	1
Does it have a backbone?	1	0	0	1	1	0	1
Is it aquatic?	1	1	0	1	1	1	0
Does it have a tail?	1	0	1	1	1	0	1
Is it a predator?	1	1	1	0	1	1	1

- Apply algorithm and record results

UNIVERSITY OF BATH

5. Measure performance II

Animal letter	A	B	C	D	E	F	G
Animal name	dogfish	octopus	scorpion	haddock	pike	seawasp	bear
Does it have teeth?	1	0	0	1	1	0	1
Does it breathe?	0	0	1	0	0	0	1
Does it have a backbone?	1	0	0	1	1	0	1
Is it aquatic?	1	1	0	1	1	1	0
Does it have a tail?	1	0	1	1	1	0	1
Is it a predator?	1	1	1	0	1	1	1
Is it an invertebrate?	0	1	1	0	0	1	mammal

- How well did your algorithm do? (ignore the bear!)

UNIVERSITY OF BATH

5. Measure performance II

Animal letter	A	B	C	D	E	F	G
Animal name	dogfish	octopus	scorpion	haddock	pike	seawasp	bear
Does it have teeth?	1	0	0	1	1	0	1
Does it breathe?	0	0	1	0	0	0	1
Does it have a backbone?	1	0	0	1	1	0	1
Is it aquatic?	1	1	0	1	1	1	0
Does it have a tail?	1	0	1	1	1	0	1
Is it a predator?	1	1	1	0	1	1	1
Is it an invertebrate?	0	1	1	0	0	1	mammal

- How well did your algorithm do? (ignore the bear!)
- Is the bear unreasonable?

UNIVERSITY OF BATH

What was your algorithm?

UNIVERSITY OF BATH

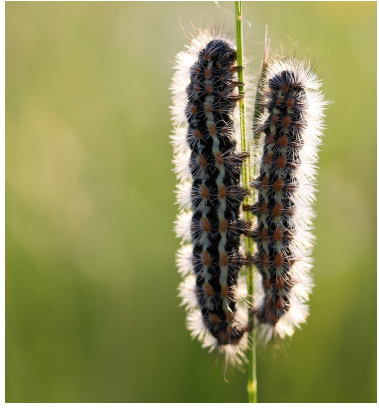
What was your algorithm?

1. Has tail $\implies$ fish
   - This is true for the training set, but violated by scorpions

What was your algorithm?

1. Has tail $\implies$ fish
   - This is true for the training set, but violated by scorpions

2. Has backbone $\implies$ fish
   - Official biological definition
   - Not always obvious! e.g. caterpillars



What was your algorithm?

1. Has tail $\implies$ fish
   - This is true for the training set, but violated by scorpions
2. Has backbone $\implies$ fish
   - Official biological definition
   - Not always obvious! e.g. caterpillars
3. Has teeth $\implies$ fish
   - Defined to be true, and much more visible
   - Invertebrates can have teeth-equivalent structures, e.g. a snail

UNIVERSITY OF BATH

What happened?

1. You found a rule that solved the problem for (training) data
2. You applied the rule to (testing) data

UNIVERSITY OF BATH

What happened?

1. You found a rule that solved the problem for (training) data
2. You applied the rule to (testing) data

- Can program step 2, e.g.

```
def invertebrate(fv):
  return fv['teeth'] == False
```

- But step 1 is less clear...

UNIVERSITY OF BATH

Supplementary definition

- Machine Learning is discovering the rule (step 1)
- Using the rule is just programming (step 2)

- Supplementary definition: A Machine Learning algorithm outputs code!

- Parameters more practical than code, e.g.

```
# Learn these:
feature = 'teeth'
match = False

# Code of model:
def evaluate(fv):
  return fv[feature] == match
```

Rule search

- How would one possible rule search process look like?

```
best = 0.0
for f in features:
  for m in [False, True]:
    accuracy = performance(f, m, train)
    if accuracy > best:
      feature = f
      match = m
```

- Search = optimisation
- This is the **decision stump** or **1 rule** algorithm
  (only works on really easy problems!)

UNIVERSITY OF BATH

Summary

- What is ML?
- Use cases
- The typical process
- Walk through / pretending to be computers
- Absurdly simple ML algorithm

Further reading & sources

Books for self-study, if you want go beyond the course material :

- "Pattern Recognition and Machine Learning" [1] by **Christopher M. Bishop**.
- "Understanding Machine Learning: From Theory to Algorithms" by **Shai Shalev-Shwartz and Shai Ben-David**. [2]

- Sample dataset for Zoo animal classification: `https://archive.ics.uci.edu/ml/datasets/Zoo`.
- A paper analysing the theoretical performance of decision stumps: "Induction of One-Level Decision Trees", by Iba & Langley (1992)

[1] `https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf`

[2] `https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf`

UNIVERSITY OF BATH

Legal

Image of automatic transmission: CC BY-ND 2.0 Kecko
https://www.flickr.com/photos/70981241@N00/1876479840

Image of speed sign: CC BY-SA 3.0 Jayron32 https://en.wikipedia.org/wiki/File:Antique_New_Hampshire_speed_limit_sign.jpg

HAL: Used without permission under fair use. From film 2001: A Space Odyssey.

Google car: Stolen from internet, used without permission under fair use.