Machine Learning 1.01: Introduction

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

UNIVERSITY OF BATH

Before we get started I...

- Per week:
  - 2 hour lab session.
  - 2 lectures.
- Lectures:
  - 16 normal.
  - 1 to introduce optional modules.
  - 1 for recap.
- Note: Computer science doesn't lecture in week 6!

UNIVERSITY OF BATH

Before we get started II...

- Marks:
  - 60% from 4 lab writeups (15% each, get harder).
  - 40% from final report (maximum 3000 words, easier than final lab).

UNIVERSITY OF BATH

Before we get started II...

- Marks:
  - 60% from 4 lab writeups (15% each, get harder).
  - 40% from final report (maximum 3000 words, easier than final lab).
- New lab every 2 weeks, hand in day before next.
- You get Christmas for report.
- Don't leave until last minute!
- Ask questions during labs, on Moodle forum.
- Using Jupyter workbooks...

Jupyter I

- Workbook via a web interface.
- Python 3 + machine learning libraries.

- Use Anaconda 3, not 2! (corresponds to Python version)
- Be careful of the Windows search bar – can run 2 or 3!

- Being taught in **Software Technologies for Data Science**...
  ...which not all of you are taking!

Jupyter II

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

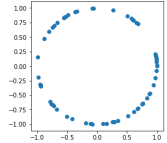
Run | Code

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

To learn

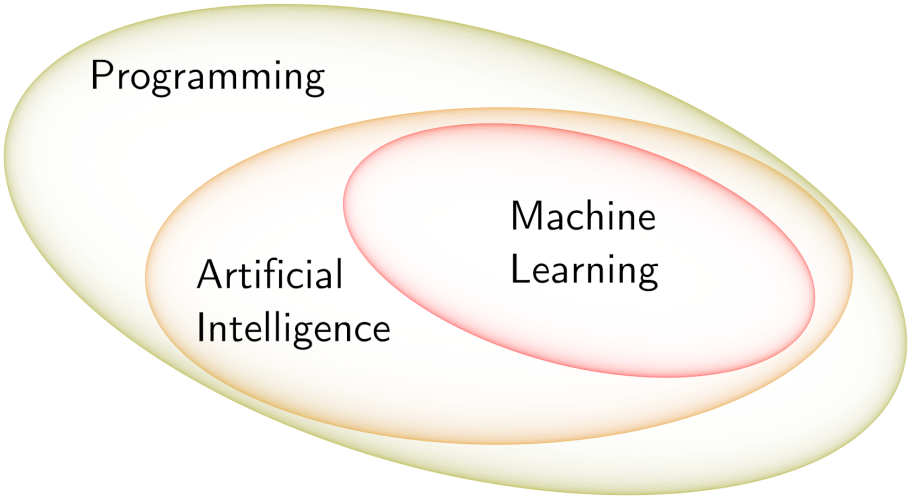
- Implementing machine learning algorithms
  (no black boxes!)
- Design machine learning algorithms
- Verifying/testing
- Optimisation

UNIVERSITY OF BATH

What is ML?

UNIVERSITY OF BATH

What is ML?



UNIVERSITY OF BATH

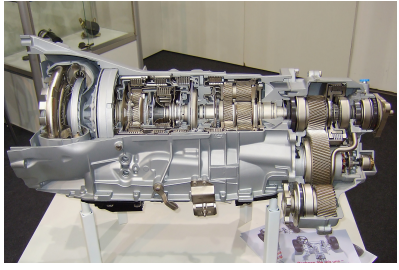
Imagine a car. . .

- 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()
    transmisison.gear += 1
    clutch.engage()
  elif engine.revs < 1000 and
       transmission.gear > 1:
    clutch.disengage()
    transmisison.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:

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

UNIVERSITY OF BATH

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?

UNIVERSITY OF BATH

What is ML?

- **Learning from data**.

What is ML?

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

UNIVERSITY OF BATH

What is ML?

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

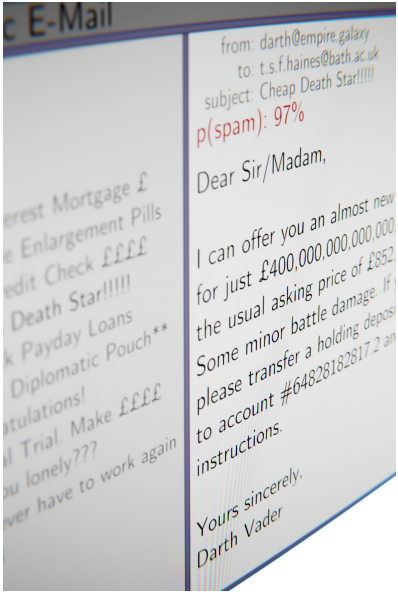
- Warning 1: Not everyone would agree with this definition.
  - overlaps with statistical models, data mining

What is ML?

- **Learning from data**.
- Built on (you also need to understand):
  - Maths, especially probability.
  - Optimisation.
  - Programming.
- Warning 1: Not everyone would agree with this definition.
  - overlaps with statistical models, data mining
- Warning 2: ML and AI often used interchangeably due to fashion/journalists.

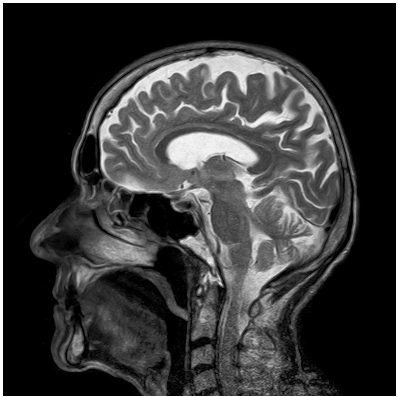
What can you do with it?

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: Medical



- Identifying a disease.
- Identifying an abnormality.
- Drug development
  (e.g. predicting which chemical to test).

Motivation: Human-computer interaction

- Handwriting or speech recognition.
- Personal assistants (e.g. Siri).
- Recommendation systems
  (e.g. product suggestions on Amazon).
- Automated design
  (e.g. designing a machine part,
  A/B testing a website design).

UNIVERSITY OF BATH

Motivation: People

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

Motivation: Financial

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

Motivation: Robots

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

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. . .)

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 – the output is a discrete label.
- It is one of the two main problem types in machine learning.

UNIVERSITY OF BATH

3. Choose or design a model

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

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

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?
- Does the algorithm really ask *"Is it an invertebrate?"*?

UNIVERSITY OF BATH

What was your algorithm?

What was your algorithm?

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

UNIVERSITY OF BATH

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.

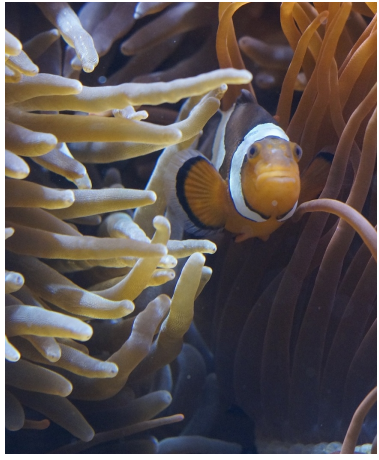
UNIVERSITY OF BATH

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.

What was your algorithm?

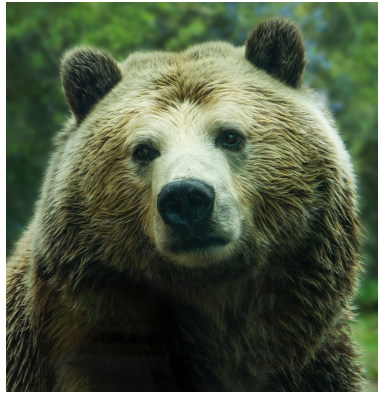
1. Has tail $\implies$ fish.
   - This is true for the training set, but violated by scorpions.
2. Has backbone $\implies$ fish.
   - Official biological definition.
   - Not always obvious! e.g. caterpillars.
3. Has teeth $\implies$ fish.
   - Defined to be true, and much more visible.
   - Invertebrates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



UNIVERSITY OF BATH

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.
4. Any others?



If you get the right answer, does how really matter?

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.

- You could program step 2, e.g.

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

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.

- You could 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

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

- Supplementary definition: A Machine Learning algorithm outputs code!

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!

- But parameters are more practical than code, e.g.

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

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

Rule search

- Can anyone describe their algorithm?

UNIVERSITY OF BATH

Rule search

- Can anyone describe their algorithm?

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

Rule search

- Can anyone describe their algorithm?

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

- 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 with yourselves instead of computers.
- An absurdly simple ML algorithm.

UNIVERSITY OF BATH

Further reading & sources

- "Information Theory, Inference and Learning Algorithms" by **David J. C. MacKay**.
- "Pattern Recognition and Machine Learning" by **Christopher M. Bishop**.
- "Computer Vision: Models, Learning, and Inference" by **Simon J. D. Prince**.

- 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)
  (Model originally proposed by psychologists to explain human behaviour in 1966!)

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.