Introduction to Machine Learning

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

UNIVERSITY OF BATH

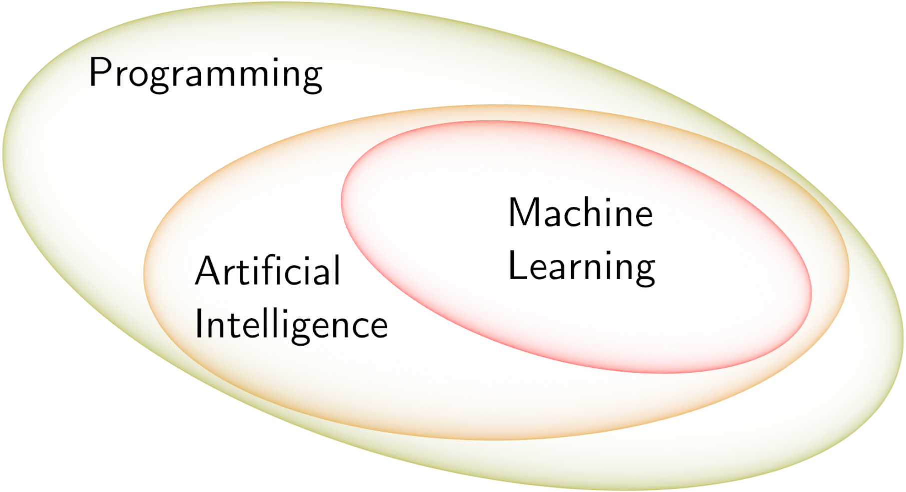
Covering

- What is ML?
- **Examples/glossary**
- Process walkthrough

(modified from intro of ML1 unit from Data Science MSc)

What is ML?

What is ML?



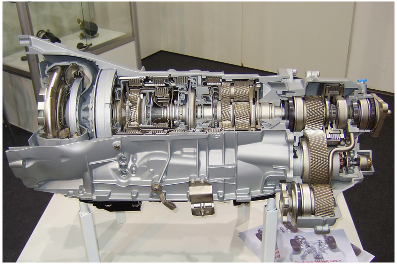
Imagine a car. . .

- 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()
```



Imagine a car. . .

- Artificial Intelligence:

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

- e.g. gps navigation:

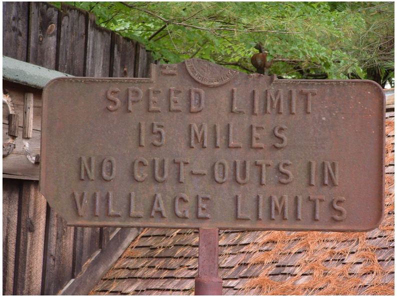
```
graph.load_map('uk.h5')
graph.set_start('bath')
graph.set_end('bletchley')
route = graph.shortest_route()
```

Imagine a car...

- Machine Learning:
  Computer learns from examples (data) and (tries to) generalise to all inputs

- e.g. recognising road signs:

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



What is ML?

- **Learning from data**

What is ML?

- **Learning from data**
- Built on:
  - Maths, especially probability
  - Optimisation
  - Programming

(this makes it quite challenging!)

What is ML?

- **Learning from data**
- Built on:
  - Maths, especially probability
  - Optimisation
  - Programming

  (this makes it quite challenging!)

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

What is ML?

- **Learning from data**
- Built on:
  - Maths, especially probability
  - Optimisation
  - Programming

  (this makes it quite challenging!)

- 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

Aside: What is data science?

Aside: What is data science?

- I don't know! (all the definitions suck)

Aside: What is data science?

- I don't know! (all the definitions suck)
- Wikipedia claims:
  "Data science [...] is an interdisciplinary field about scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured, similar to data mining"

  Isn't that just **science**?

Aside: What is data science?

- I don't know! (all the definitions suck)
- Wikipedia claims:
  "Data science [...] is an interdisciplinary field about scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured, similar to data mining"
  Isn't that just **science**?
- Wikipedia also says:
  "When Harvard Business Review called it "The Sexiest Job of the 21st Century" the term became a buzzword, and is now often applied to business analytics, or even arbitrary use of data, or used as a sexed-up term for statistics"

What can you do with it?

Supervised learning

- Learn a function: $y = f(\vec{x})$
- From (many) examples of input ($\vec{x}$) and output (y)

- Majority of ML:
  Classification or regression...

Supervised learning: Classification

- Learn a function: $y = f(\vec{x})$
- Classification: y is **discrete**

Supervised learning: Classification

- Learn a function: $y = f(\vec{x})$
- Classification: y is **discrete**

- Identifying camera trap animals
  - Input: Image
  - Output: Which animal

(peccary)

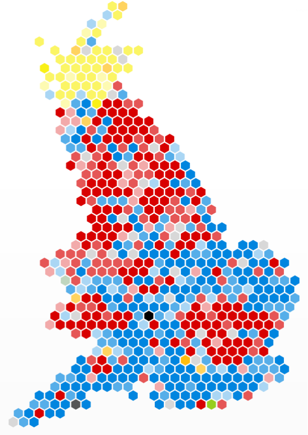
Supervised learning: Classification

- Learn a function: $y = f(\vec{x})$
- Classification: y is **discrete**

- Identifying camera trap animals
  - Input: Image
  - Output: Which animal

- Predicting voting intention
  - Input: Demographics
  - Output: Preferred candidate (probabilistic)
  - Run on entire country $\rightarrow$ Predict election winner

(peccary)



(YouGov, 2017-06-07)

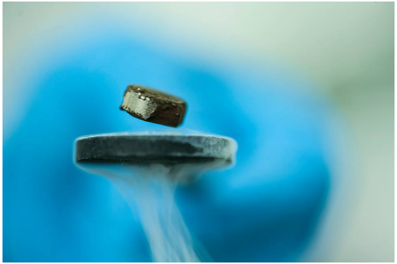
Supervised learning: Regression

- Learn a function: $y = f(\vec{x})$
- Regression: y is **continuous**

Supervised learning: Regression

- Learn a function: $y = f(\vec{x})$
- Regression: y is **continuous**

- Predicting critical temperature of a superconductor
  - Input: Material properties
  - Output: Temperature

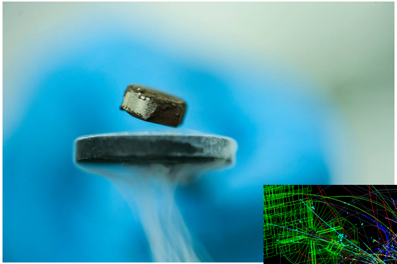


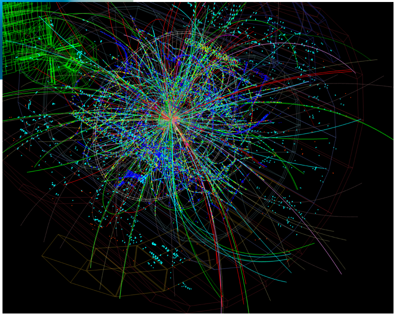
Supervised learning: Regression

- Learn a function: $y = f(\vec{x})$
- Regression: y is **continuous**

- Predicting critical temperature of a superconductor
  - Input: Material properties
  - Output: Temperature

- Inferring particle paths (LHC)
  - Input: Detector energy spikes
  - Output: Particle paths
  - Trained with simulation





Supervised learning: Further kinds

- Multi-label classification: y is a **set**
  - e.g. identifying objects in an image
  - e.g. text summarisation (reusing source sentences)

Supervised learning: Further kinds

- Multi-label classification: y is a **set**
  - e.g. identifying objects in an image
  - e.g. text summarisation (reusing source sentences)

- Structured prediction: y is anything else!
  - e.g. Sentence tagging: y is a sequence
    (such as part-of-speech tagging)
  - e.g. Automated design: y is a CAD model

Unsupervised learning

- No y!
- Finds *patterns* in data

- Examples:
  - Clustering
  - Density estimation
  - Dimensionality reduction

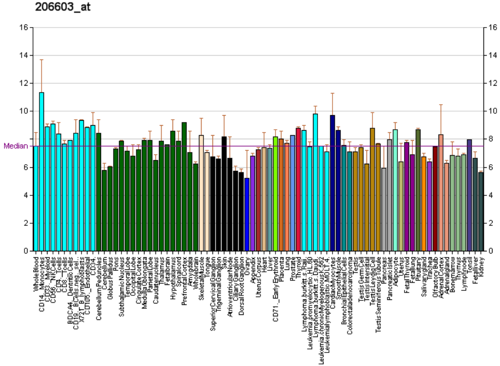
Unsupervised learning: Clustering

- Clustering:
  - Groups "similar" data points
  - Arbitrary similarity definition

Unsupervised learning: Clustering

- Clustering:
  - Groups “similar” data points
  - Arbitrary similarity definition
- Identifying *co-regulated genes*:
  - Input: Many expression level measurements
  - Output: Groups of genes that tend to express at same time

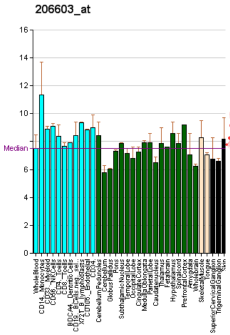
(Gene expression of SLC2A4)

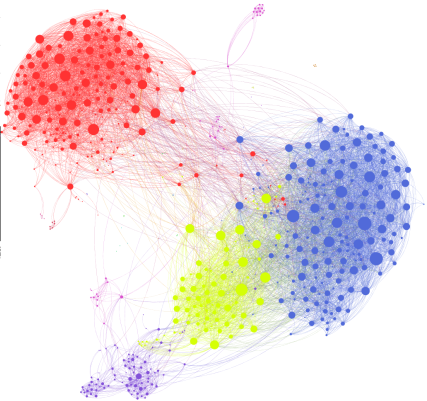


Unsupervised learning: Clustering

- Clustering:
  - Groups "similar" data points
  - Arbitrary similarity definition
- Identifying *co-regulated genes*:
  - Input: Many expression level measurements
  - Output: Groups of genes that tend to express at same time
- Discovering social groups
  - Input: Friend graph
  - Output: Social groups (individuals may belong to several)

(Gene expression of SLC2A4)





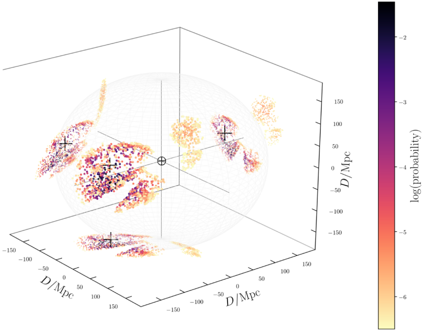
(A Facebook friend graph)

Unsupervised learning: Density estimation

- Density estimation:
  - Learns distribution of data
  - i.e. $x_i \sim P$

Unsupervised learning: Density estimation

- Density estimation:
  - Learns distribution of data
  - i.e. $x_i \sim P$
- Finding coalescing binary neutron stars with LIGO
  - Input: Locations that explain detection
  - Output: Search order for optical follow up



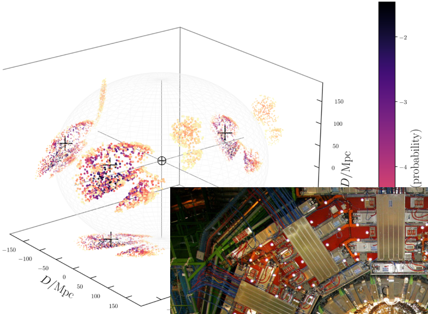
Unsupervised learning: Density estimation

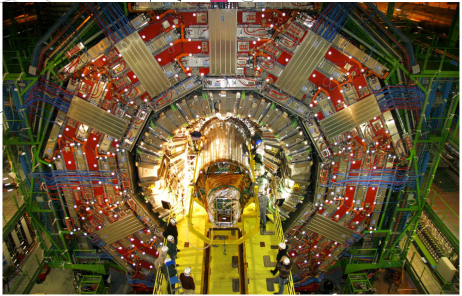
- Density estimation:
  - Learns distribution of data
  - i.e. $x_i \sim P$

- Finding coalescing binary neutron stars with LIGO
  - Input: Locations that explain detection
  - Output: Search order for optical follow up

- Detecting LHC problems
  - Input: Normal outputs
  - Output: Possible failures

  Example of *abnormality detection*





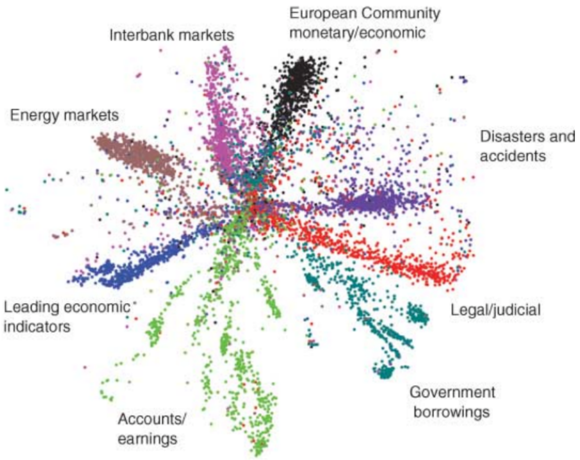
Unsupervised learning: Dimensionality reduction

- Dimensionality reduction / manifold learning:
    - Reduce dimensions while preserving information
    - Also used for visualisation (important for verification)

Unsupervised learning: Dimensionality reduction

- Dimensionality reduction / manifold learning:
  - Reduce dimensions while preserving information
  - Also used for visualisation (important for verification)

- Organising news
  - Input: Word vectors
  - Output: Position in layout



*-supervised

- Collecting data cheap
- Labelling data expensive

- **Semi-supervised**:
    - Some labelled data
    - Lots of unlabelled data

*-supervised

- Collecting data cheap
- Labelling data expensive

- **Semi-supervised**:
  - Some labelled data
  - Lots of unlabelled data

- Precise labels expensive,
  inaccurate labels cheap

- **Weakly-supervised**:
  - Learns from “weak” labels
  - Outputs “strong” labels

*-supervised

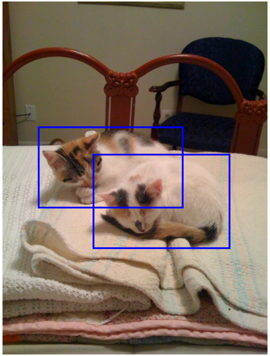
- Collecting data cheap
- Labelling data expensive

- **Semi-supervised**:
  - Some labelled data
  - Lots of unlabelled data

- Precise labels expensive, inaccurate labels cheap

- **Weakly-supervised**:
  - Learns from "weak" labels
  - Outputs "strong" labels

- e.g. finding cats
  - Image contains cat – fast
  - Box around cat – slow



Graphical models

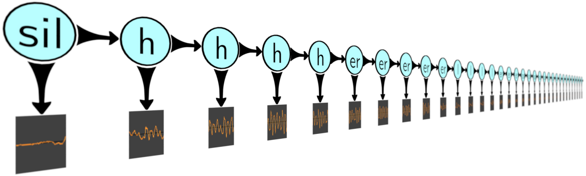
- **Represents structure** by drawing relationships. . .

Graphical models

- **Represents structure** by drawing relationships. . .

- Voice recognition
- Hidden Markov model, used twice:
  1. Align phonemes with audio (weakly-supervised)
  2. Recognition using language model (structured prediction)

“hello”



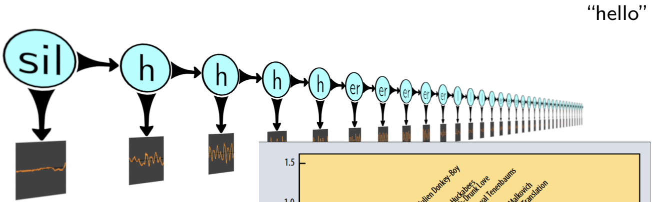
Graphical models

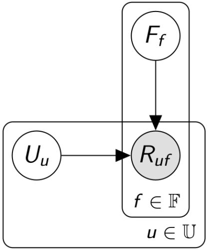
- **Represents structure** by drawing relationships. . .

- Voice recognition
- Hidden Markov model, used twice:
  1. Align phonemes with audio
     (weakly-supervised)
  2. Recognition using language model (structured prediction)

- Recommender systems
  e.g. for films
  - Input: User ratings (sparse)
  - Output: Omitted ratings

(estimating missing values)





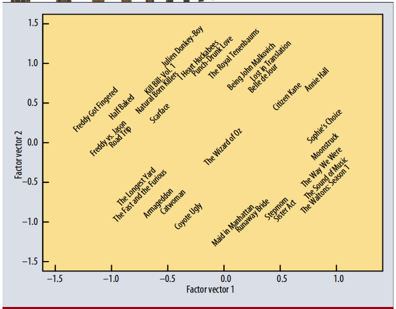
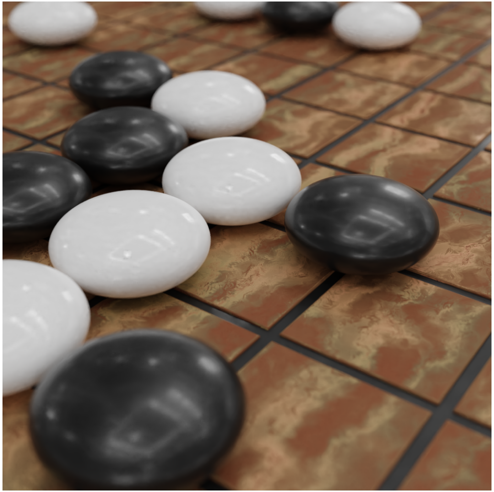


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

Reinforcement learning

- *Actions* in an *environment*
- *Delayed reward*

- Examples:
  - Games, e.g. *Alpha Go*
  - Agents (inc. working together)
  - Robots



Glossary

- **Supervised**
  - Classification
  - Regularisation
  - Multi-label classification
  - Structured prediction
- Semi-supervised
- Weakly-supervised
- Graphical models

(incomplete!)

- **Unsupervised**
  - Clustering
  - Density estimation / abnormality detection
  - Dimensionality reduction / manifold learning
- **Reinforcement learning**

Further categories

Can also classify ML algorithms by...

Further categories

Can also classify ML algorithms by...

- Answer quality:
  - Point estimate
    e.g. "*You have cancer*"
  - Probabilistic
    e.g. "*60% chance you have cancer*"
  - Bayesian
    e.g. "*5% chance you have cancer*"

Further categories

Can also classify ML algorithms by…

- Answer quality:
  - Point estimate
    e.g. "*You have cancer*"
  - Probabilistic
    e.g. "*60% chance you have cancer*"
  - Bayesian
    e.g. "*5% chance you have cancer*"

- Workflow:
  - Batch learning
    i.e. Collect data then learn
  - Incremental learning
    i.e. Learn as data arrives
  - Active learning
    i.e. Algorithm selects data to learn from!
    (leads to automating science…)

UNIVERSITY OF BATH

Further categories

Can also classify ML algorithms by...

- Answer quality:
  - Point estimate
    e.g. "*You have cancer*"
  - Probabilistic
    e.g. "*60% chance you have cancer*"
  - Bayesian
    e.g. "*5% chance you have cancer*"
- Area:
  - Traditional
  - Computer vision
  - Natural language processing (NLP)
  - Interactive
- Workflow:
  - Batch learning
    i.e. Collect data then learn
  - Incremental learning
    i.e. Learn as data arrives
  - Active learning
    i.e. Algorithm selects data to learn from!
    (leads to automating science...)

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. . .)

1. Choose a problem

e.g. this toy problem:

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

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
- Note: Data collection is usually the hardest bit!

UNIVERSITY OF BATH

3. Choose or design a model

- This is a **classification** problem – supervised, output discrete

3. Choose or design a model

- This is a **classification** problem – supervised, output discrete

- There are hundreds of models for solving it
- Lets use another “model”: A rule (algorithm) created by you!

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!

5. Measure performance I

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

Animal letter	A	B	C	D	E	F	G
Animal name							
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

5. Measure performance II

Animal letter Animal name	A dogfish	B octopus	C scorpion	D haddock	E pike	F seawasp	G 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!)

5. Measure performance II

Animal letter Animal name	A dogfish	B octopus	C scorpion	D haddock	E pike	F seawasp	G 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?"*?

What was your algorithm?

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?

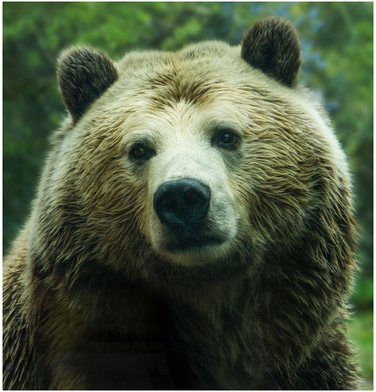
1. Has tail $\implies$ fish
   - This is true for the training set, but violated by scorpions
2. Has backbone $\implies$ fish
   - Official biological definition
   - Not always obvious! e.g. caterpillars
3. Has teeth $\implies$ fish
   - Defined to be true, and more visible
   - Invertebrates can have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

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 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?

What happened?

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

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.

```python
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...

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!

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.

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

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

Rule search

- Can anyone describe their step 1?

Rule search

- Can anyone describe their step 1?

```
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 step 1?

```
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!)

Summary

- What is ML?
- Glossary of scenarios
- Typical process

- Walk through
- Absurdly simple 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!)

Legal I

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`

Camera trap images: CC BY 2.0 J.N. Stuart
`https://www.flickr.com/photos/usfwshq/albums/72157628775623325/with/6659380637/`

Particle tracks: Copyright CERN, stolen without permission

Legal II

Gene expression: CC BY-SA 3.0 Genomics Institute of the Novartis Research Foundation
`https://commons.wikimedia.org/wiki/File:PBB_GE_SLC2A4_206603_at_fs.png`

Facebook social graph: Stolen, license unknown
`https://griffsgraphs.wordpress.com/2012/07/02/a-facebook-network`

News story analysis: Stolen, license unknown
`http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality/`

Kittens: CC BY 2.0 Kacie "Aurora" `https://www.flickr.com/photos/toodamnadorable/3059980811/`