Machine Learning: Pattern Recognition

Lecture 1: Introduction

University of Amsterdam

5 September 2011

Introduction

What's it about?

Machine Learning	Make machines learn from examples
Pattern Recognition	Find patterns in data

Objective:

- Learn advanced, state-of-the art methods for pattern recognition and data modeling
- When possible, to refer back to human learning
- Today's lecture: Introduction to the field, overview of the course.
- Today's Exercise session: Some basic math

Outline

Practical matters

Organisation:

- The course consists of lectures, labs and exercise sessions
- The final grade is weighed as:
  50% exam, 50% lab + exercises.
- You must pass for the exam (the exercises are in groups)
- Labs and homeworks not handed in on time get a zero score

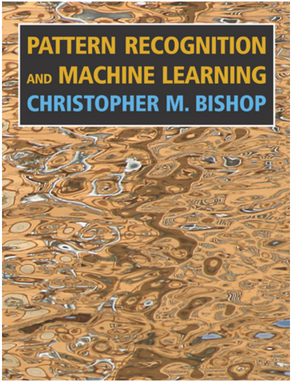
Schedule:

Lectures:	Monday, 4pm – 6pm
Exercise session:	Monday, 6pm – 7pm
Computer Labs:	Wednesday, 2pm – 4pm (Before mid-term break)
	Friday, 12pm – 2pm (After mid-term break)
Lecturer:	Gwenn Englebienne [G.Englebienne@uva.nl]
Assistant:	Martijn Liem [mliem@science.uva.nl]

Practical matters

Book:

- **Pattern Recognition and Machine Learning**, Christopher M. Bishop, Springer (2006)



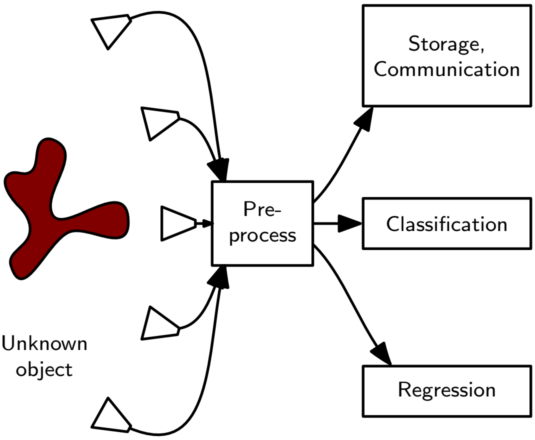
- Everything else will be available from Blackboard

Schedule

UNIVERSITY OF AMSTERDAM

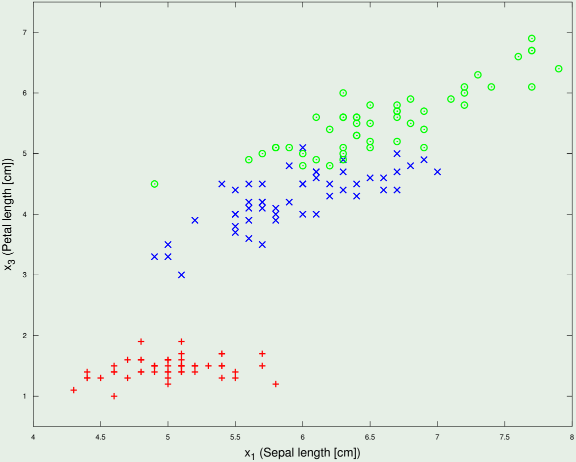
Wk	Date	Lecture	Exercise	Deadline
36	05 Sept.	Introduction & Mathematics		
	07 Sept.	Evaluation and issues	Classification lab	14 Sept.
37	12 Sept.	Bayesian decision theory		
	14 Sept.	Linear classification	Logistic regression	21 Sept.
38	19 Sept.	Graphical models		
	21 Sept.	Generative vs. Discriminative	Spam Filtering	28 Sept.
39	26 Sept.	Gaussian Mixtures and E.M.		
	28 Sept.	Unsupervised learning	EM for GMM	05 Oct.
40	03 Oct.	Guest Lecture		
	05 Oct.	Guest Lecture	Pedestrian classification	17 Oct.
41	10 Oct.	Dimensionality reduction		
	12 Oct.	Non-parametric models	Gaussian Processes	17 Oct
42	17 Oct.	Approximate inference		
	19 Oct.	Questions		
43	25 Oct.	Exam		

Basic Framework

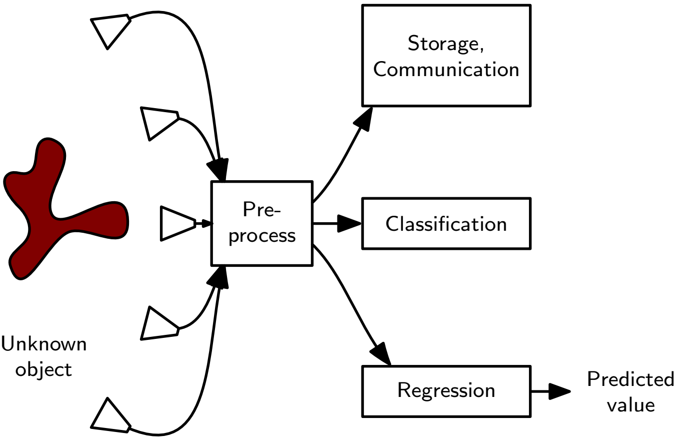


An example of classification

Example: Iris classification



Basic Framework

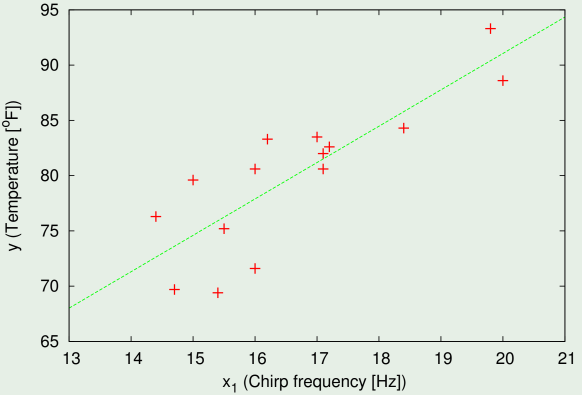


University of Amsterdam

Regression example

University of Amsterdam

Example: Evaluating temperature from cricket activity



Classification vs. Regression

- **Classification**: Predict a discrete label from features

Example

- Medicine: classify X-rays as "cancer" or "healthy"
- SPAM detection: classify emails as spam or not
- Face recognition, speech recognition, . . .

- **Regression**: Predict a continuous value

Example

- Weather forecasting (wind speed, mm rainfall, . . .)
- In financial markets: predict tomorrow's stock price from past evolution and external factors
- A robot learning its location in an environment

Classification vs. Regression

- **Classification**: Predict a discrete label from features

Example

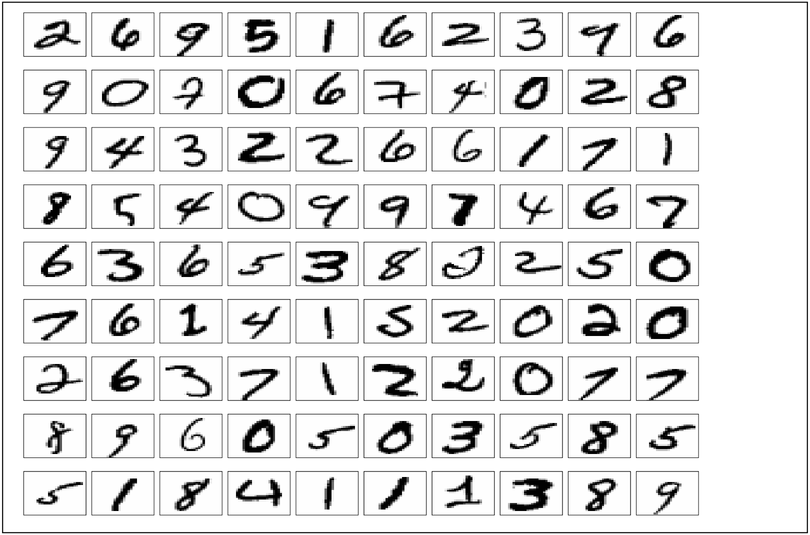
- Medicine: classify X-rays as "cancer" or "healthy"
- SPAM detection: classify emails as spam or not
- Face recognition, speech recognition, ...

- **Regression**: Predict a continuous value

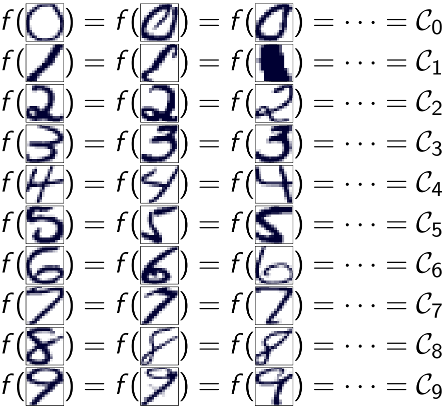
Example

- Weather forecasting (wind speed, mm rainfall, ...)
- In financial markets: predict tomorrow's stock price from past evolution and external factors
- A robot learning its location in an environment

Another classification example



In functional form



A little bit of context

Artificial Intelligence has been trying to solve such problems for a long time.
One approach was to give the computer a set of hard-coded rules.
In the 1980's, **expert systems**, were quite popular.

if A **then** X
if B **then** Y
...

There are of course problems with those:

- It may not be possible to account for all possibilities
- It is very hard to avoid inconsistent rules

Search-based systems

Another approach was to view classification/regression as a search problem:

Example

Games: assign a value to possible moves

This has been extremely successful when the world is known (Deep Blue beat Gary Kasparov in 1997 with such a technique)

Logic

A similar approach is used in deductive logic:

All men are mortal	$\forall x : \text{man}(x) \rightarrow \text{mortal}(x)$
Socrates is a man	$\text{man}(\text{socrates})$
Socrates is mortal	$\text{mortal}(\text{socrates})$

The discipline that deals with programs that can make such inferences is called **theorem proving**.
It was conjectured that this could be used for common sense reasoning:

- Code up common sense knowledge as logical axioms and let a theorem prover do the rest
- This is now out of fashion: logic is too rigid to accommodate many aspects of common sense reasoning

Drawbacks

There are two major problems with these approaches:

1. They cannot allow for uncertainty: theorem proving cannot handle it at all, and expert systems can easily become incoherent)
2. There are many problems we're all experts in, but we cannot transmit our knowledge

Example

Understanding speech — What are the properties of an "AH" phoneme, and how is this different from an "EH"? How are these affected by surrounding phonemes?

Machine Learning

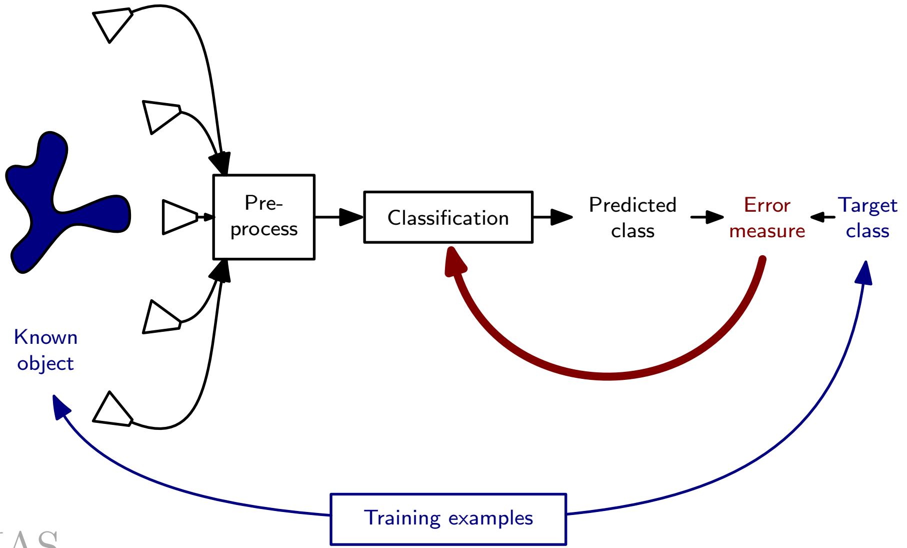
We therefore need systems that can

1. deal with uncertainty
2. learn from examples

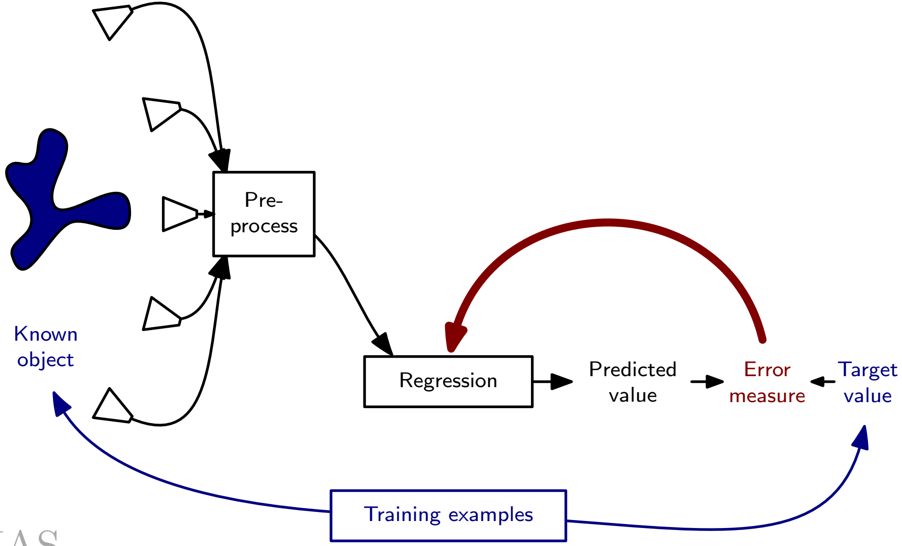
In this course we focus on systems where:

- We learn from some training data, use this to modify our machine's knowledge, and then use it. The machine does not adapt during use (off-line learning).
- Our system does not affect the world, and does not get feedback from it (cf. Reinforcement learning).

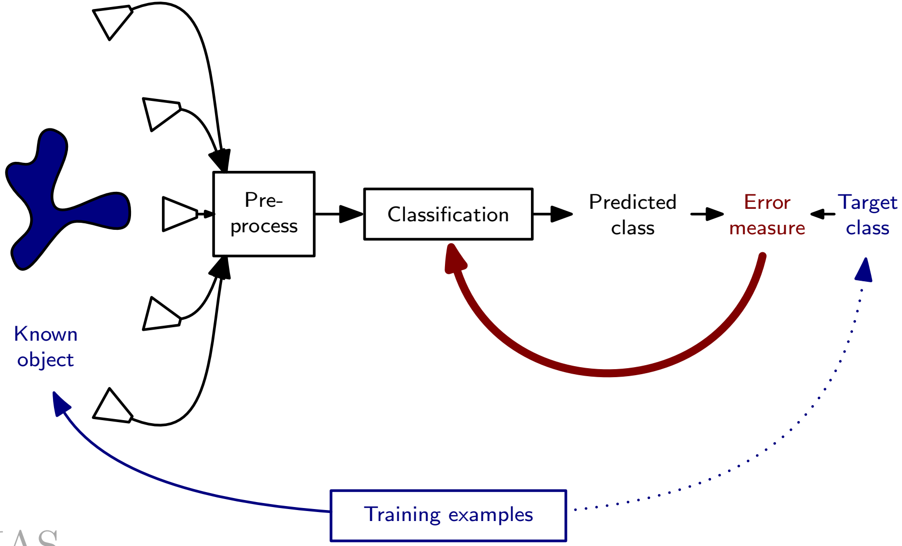
Supervised Training — Classification



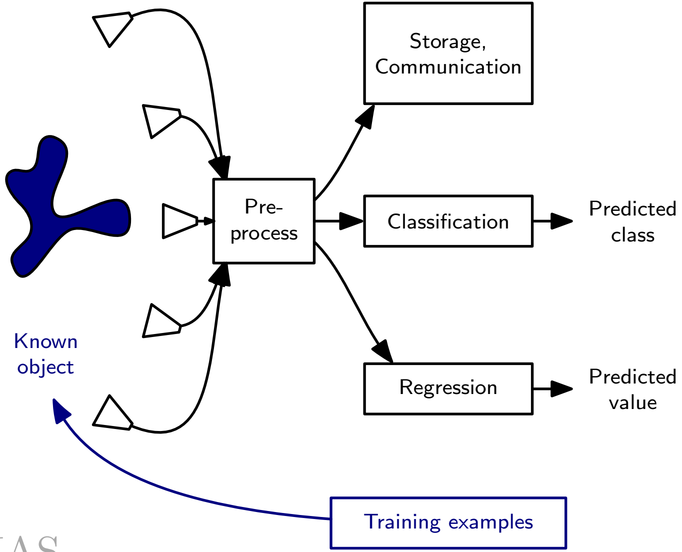
Supervised Training — Regression



Semi-supervised Training



Unsupervised Training

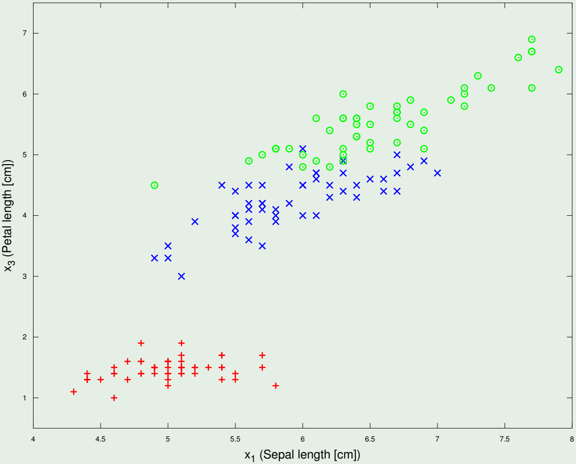


Learning

University of Amsterdam

What is it about the data that makes learning possible?

Example: Iris classification revisited



Structure

- Classification is based on this simple assumption:
  - *Similar things are likely to belong to the same class*
- More generally, all of machine learning is based on some assumption of *smoothness*
- So what does "similar" mean?
  - Based on the information we have — the features
  - Based on some measure of similarity — some distance metric
- A lot of effort in Machine Learning is put in selecting the right features and finding the right distance measure

A tale of learning

A young father wants to teach his son about sport cars. He attempts to describe them, but finds that quite challenging and instead takes his son to the nearest bridge over a highway and points out "*That's a sports car*" for each passing such car.

After a while, he asks his son whether he understands what sports cars are like?

"*Sure, it's easy,*" replies the son: "*That's a sports car!*" he exclaims, pointing out an old Trabant, whose red paint was full or rusty patches.

Dejected, the father asks why he thinks so.

"*Sport cars are red cars,*" his son replies.

— David Barber: ML, a probabilistic approach

Moral of the story: Don't expect miracles.

A tale of learning

A young father wants to teach his son about sport cars. He attempts to describe them, but finds that quite challenging and instead takes his son to the nearest bridge over a highway and points out "*That's a sports car*" for each passing such car.

After a while, he asks his son whether he understands what sports cars are like?

"*Sure, it's easy,*" replies the son: "*That's a sports car!*" he exclaims, pointing out an old Trabant, whose red paint was full or rusty patches.

Dejected, the father asks why he thinks so.

"*Sport cars are red cars,*" his son replies.

— David Barber: ML, a probabilistic approach

Moral of the story: Don't expect miracles.

Noise

The features are noisy

- Because the sensors are not perfect
- Because the process itself has a stochastic component

Example

Estimating the position of a satellite from radar measurements:

- Sensor noise: due to the imperfection of radar receiver, random deflections of the radar waves by atmospheric turbulence, ...
- Process noise: occasionally the satellite will hit debris, sustain atmospheric drag, ...

- It is therefore important to have some way of dealing with the noise

Uncertainty

How can we deal with the uncertainty of the sensors?
Probability theory:

- Provides a principled way of dealing with uncertainty
- Functional mapping from propositional logic to $[0, 1]$
- Based on two axioms:
  - if $\models \phi$, then $p(\phi) = 1$
  - if $\models \neg(\phi \wedge \psi)$, then $p(\phi \vee \psi) = p(\phi) + p(\psi)$

  All the rules of probability are derived from these axioms.
- Arguably the only principled model of reasoning (We'll come back to this)

Not all techniques and methods we'll see in this class are probabilistic. But when we'll want to prove that they're sensible, we'll resort to probabilistic reasoning.

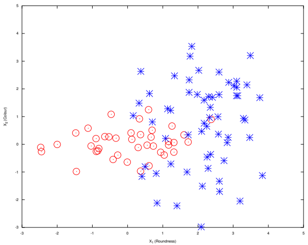
This course

UNIVERSITY OF AMSTERDAM

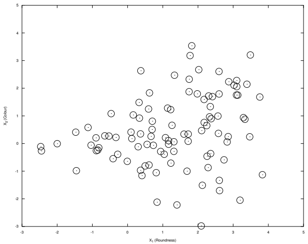
- We focus mainly on *classification* with *supervised* training
- We occasionally discuss regression
- Towards the end of the course we discuss
    - Unsupervised techniques
    - Dimensionality reduction

Supervised vs. Unsupervised

- In *supervised* methods, a classifier is trained on a set of *labeled* samples. The aim of the system is to predict the class of a previously unseen data element.



- In *unsupervised* methods, *no* class labels are given. It is up to the system to discover (hopefully meaningful) *structure* in the data, and to discover what classes exist in the data. Similar techniques are used for dimensionality reduction.



Supervised learning

Basic issues of classification:

- Given:
  - Classes, $\mathcal{C} \in \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$
  - Data elements / Feature values: $\mathbf{x} = (x_1, \ldots, x_d)^\top$
- What are the best features / Should we use all features?
- How do we *learn* to classify unseen data from a set of training examples $\{(\mathbf{x}^{(i)}, \mathcal{C}^{(i)}), i = 1, \ldots, n\}$

For regression, we predict a continuous value rather than a discrete label

Approaches to supervised learning

- Non-parametric methods (Sample/prototype based)
  - Store all training examples (or a selection of prototypes)
  - Classify based on similarities
- Discriminant Functions
  - Choose a decision function h, so that $\hat{\mathcal{C}} = h(\mathbf{x})$
  - Estimate the parameters of this function from training data
  - Maximum likelihood/à posteriori or worst case analysis to estimate the model
  - Classify new patterns based on the estimated decision rule
- Model-based (Bayesian Decision)
  - Assume / find probability density functions that can represent the distribution of the data
  - Estimate the parameters of those distributions by Maximum Likelihood/Maximum à posteriori estimation
  - Use this density estimate to classify new patterns

Clustering

University of Amsterdam

Goal: divide the data in groups, such that:

- Items in each group are similar
- Dissimilar items are in different groups

Example

Customer/product clustering

- Identify groups of customers with similar buying patterns for targeted marketing campaigns: send mailings only to likely buyers
- Identify groups of products that are often bought together, offer packages of products for reduced price
- Recommender systems: Jointly cluster users of movies, books, CD's,... (e.g. Amazon, Netflix, ...)

Clustering

Goal: divide the data in groups, such that:

- Items in each group are similar
- Dissimilar items are in different groups

Example

Customer/product clustering

- Identify groups of customers with similar buying patterns for targeted marketing campaigns: send mailings only to likely buyers
- Identify groups of products that are often bought together, offer packages of products for reduced price
- Recommender systems: Jointly cluster users of movies, books, CD's,. . . (e.g. Amazon, Netflix, . . .)

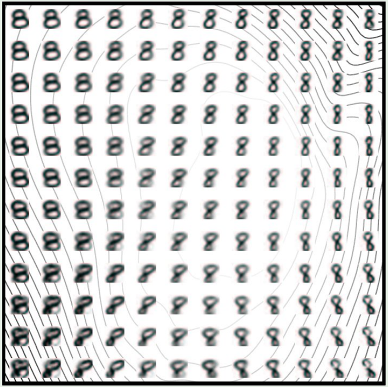
Dimensionality reduction

- Digit example: 16×16 pixels, 256 intensities
  - $256^{256} \approx 10^{616}$ possible images
  - If you tried to list all such images, and generated them at the rate of one per second, you'd need (a lot) more time than the lifespan of the universe ($\approx 10^{157} s$) to list them all.
  - Notice that doing it faster does not help much: a supercomputer generating 10 billion billion billion images per second would still need 10^{589} seconds, or 10^{432} universes ...
- However most of these possible images are not meaningful
  - In this 256D space, only limited locations are used
- It is therefore possible to reduce the size of the description, without losing information

Dimensionality reduction

- Used for data compressing and reconstruction
- Used as a pre-processing step, to reduce classifier complexity

Example



Summary

- We introduced Machine Learning
- Learning from data can be broadly divided as follows:
  - Supervised
    - Classification
    - Regression
  - Unsupervised
    - Clustering
    - Dimensionality reduction
- We need ways to find structure in data...
- ...while at the same time disregarding noise
- Example class: Maths and Probabilities
- Lab: Introduction to Matlab
- Next week: a more in-depth analysis of the issues