Basic Machine Learning

3 September 2018

Teachers/Instructors

Gwenn Englebienne

Office: ZI-2098
Email: g.englebienne@utwente.nl

Mannes Poel

Office: ZI-4035
Email: m.poel@utwente.nl

Recent Advances in ML

- Reading house numbers
- Interpreting pictures
- Playing Go at master level
- Google DeepMind
- Kaggle competitions
- Deep learning

VOCABULARY

Niiiiiiiiice: How Tweets Reveal Your Age

As they say, you are what you tweet

By Katy Steinmetz @katysteinmetz | July 17, 2013

Share | Like 240 | Tweet | G+1 9 | in Share 34 | Pin it | Read Later

Michael DeLeon / Getty Images

Writing about her new study on Twitter and age, researcher Dong Nguyen starts off with a little quiz. How old do you think the people are who sent these respective tweets?

AS LONG AS YOU LOVE ME❤

Interesting article about usability design on mobile search [LINK]

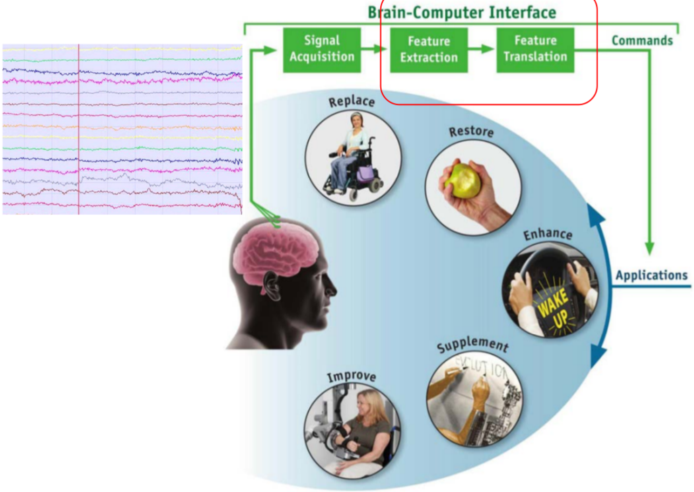
That might seem like a test for advanced Twitterati, but in a paper published this month—titled "'How Old Do You Think I Am?': A Study of Language and Age in Twitter"—four Dutch researchers reveal stylistic tics associated with younger and older tweeters. Nguyen's team also discovered that, based on such tweet-tics, an automated program can better predict your age than a fellow human can.

RELATED

The Edward Snowden Name Game: Whistle-Blower, Traitor, Leaker

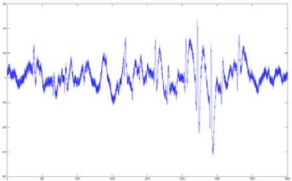
Brain-Computer interfaces

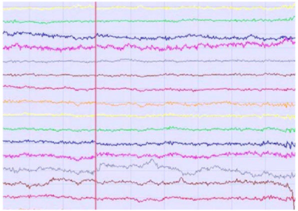
ML part



Goal of BCI

- Challenge: translate brain signals to intentions or mental state

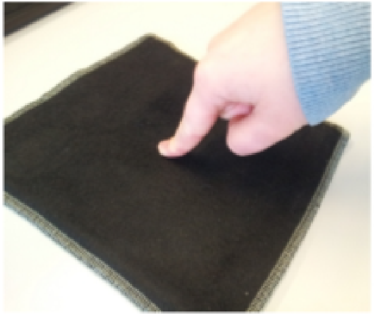


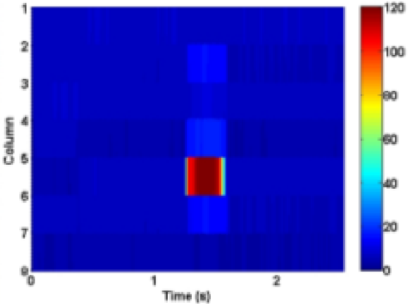


- Issues: Ground truth, signal-to-noise ratio

Touch interaction: integrating touch in HCI

Can we detect, recognise, interpret touch gestures





Results

Predicted class		Actual class 1	2	3	4	5	6	7	8	9	10	11	12	13	14
Grab	1	121	0	1	0	0	0	18	2	0	0	73	0	0	0
Hit	2	0	116	0	26	0	6	1	0	0	53	0	0	31	0
Massage	3	7	0	127	0	2	1	0	18	12	0	7	1	0	8
Pat	4	0	17	3	43	0	5	0	4	4	18	0	14	26	5
Pinch	5	2	2	12	4	125	18	15	5	5	2	26	5	1	3
Poke	6	0	7	0	7	16	131	15	0	1	2	1	0	18	3
Press	7	8	3	0	6	25	8	115	11	5	2	14	4	3	0
Rub	8	0	2	17	4	0	0	2	76	16	0	0	33	4	10
Scratch	9	0	1	3	4	0	1	1	18	96	0	0	8	0	40
Slap	10	0	25	0	20	0	0	0	0	1	93	0	1	17	0
Squeeze	11	46	1	7	2	17	1	15	0	1	1	65	2	1	0
Stroke	12	0	1	9	4	0	1	2	37	11	1	0	109	3	3
Tap	13	0	11	0	61	1	13	2	0	0	12	0	2	78	5
Tickle	14	2	0	7	5	0	1	0	14	34	2	0	6	4	109
	sum	186	186	186	186	186	186	186	185	186	186	186	185	186	186

Machine Learning

What's it about?

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

Objective:

- Introduction to state-of-the art methods for machine learning and data modeling
- When possible, to refer back to human learning
- Today's lecture: Introduction to the field, overview of the course.
- This week's lab: familiarization with the Python libraries

Practical matters

Organisation:

- Basic (1st quarter) and Advanced (2nd quarter) courses
- The course consists of lectures, labs and exercise sessions
- Advanced course ends with a project
- Toolkit: Python 3.5 (We recommend Anaconda for new users)
- Exercises are in groups of 2 persons
- The final grade is weighed as:

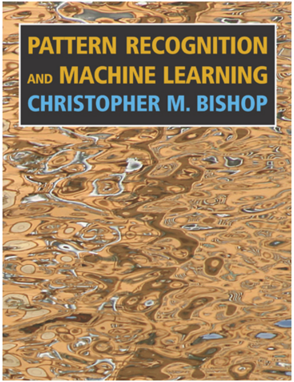
  $$50\%\ \text{exam},\ 50\%\ \text{lab} + \text{exercises}.$$

  Passing requires 5/10 for both parts

Practical matters

Book:

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



- Everything else will be available from Canvas

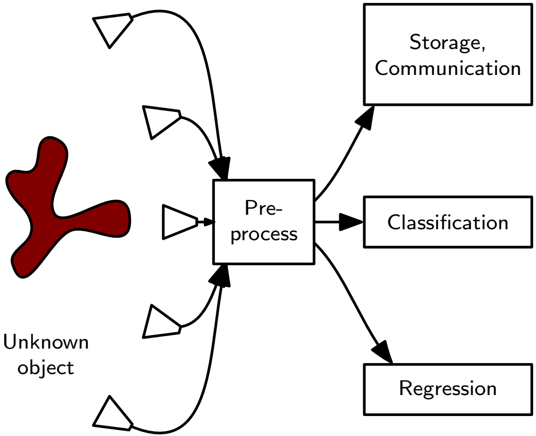
Schedule: Basic course

Wk.	Subject	Exercise/Lab
1	Introduction & Mathematics	Familiarising Python & libraries
2	Linear Discriminants	Linear discriminants, overfitting
3	Training/testing, validation, overfitting, regularisation...	Exercises on overfitting
4	Decision Trees	Decision Trees
5	Neural networks	Regression and classification with NN
6	Support Vector Machines	SVM
7	Bayesian modelling	Implement the E.M. algorithm
8	Dimensionality reduction	PCA, ...
9	Rehearsal, example exam	
10	Written exam	

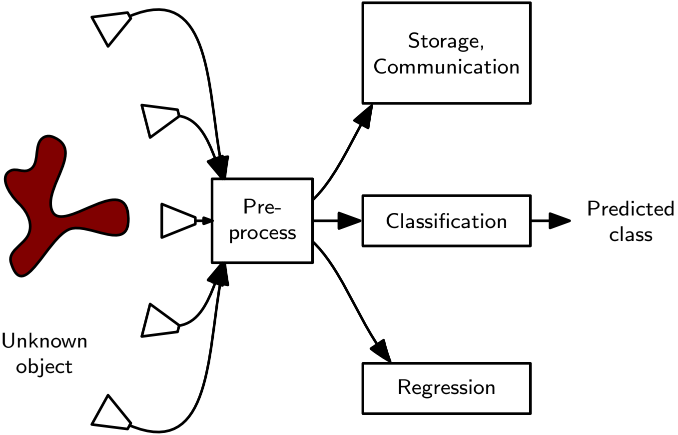
Schedule: Advanced course

Wk.	Subject	Exercise/Lab
1	Graphical models	Implementing GM
2	Dynamic models	Hidden Markov models
3	Sampling	Exercise on sampling
4	Deep learning	Exercise on deep learning
5	Combining models	Implement boosting
6-10	Project	

Basic Framework

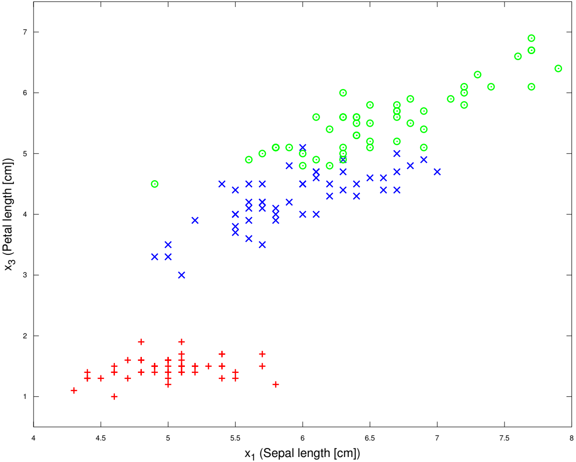


Basic Framework

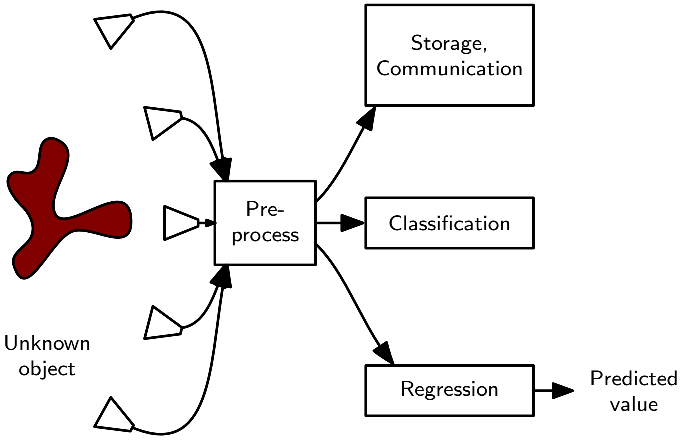


An example of classification

Example: Iris classification

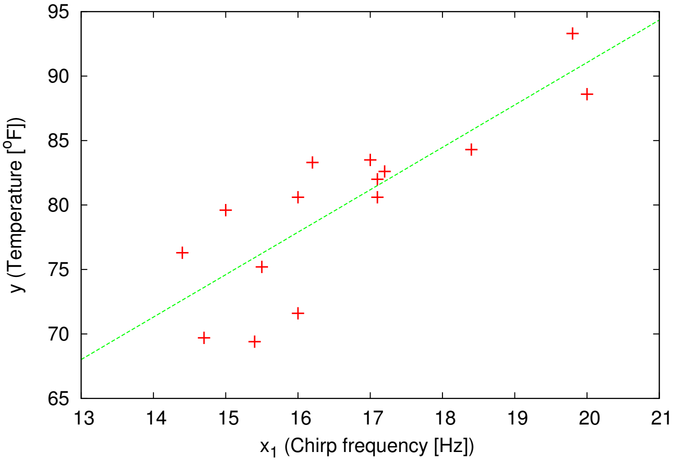


Basic Framework



Regression example

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, ...
  - Fall risk estimation

- **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

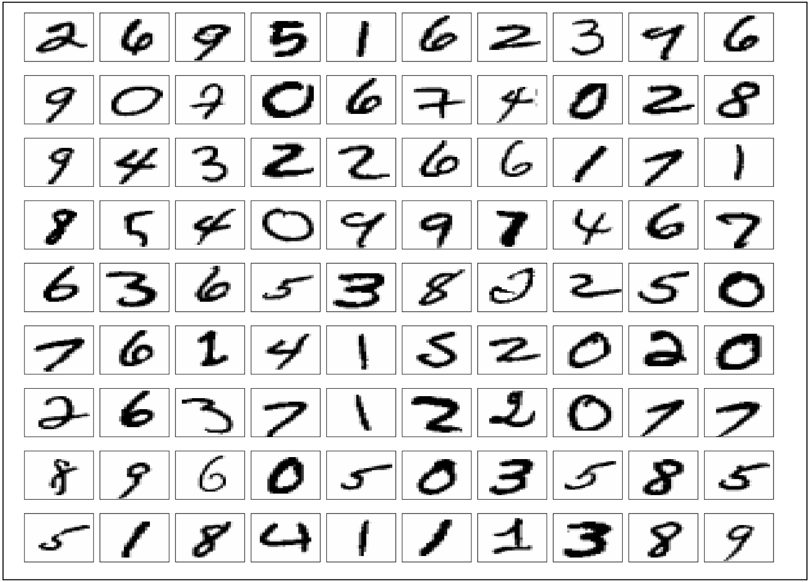
- **Classification**: Predict a discrete label from features

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

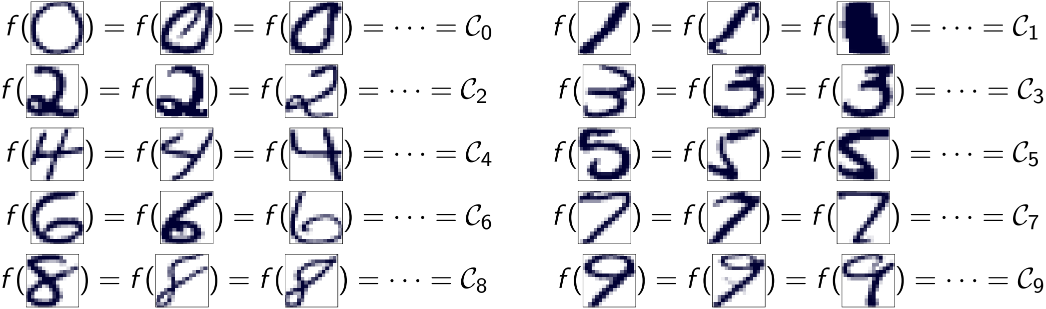
- **Regression**: Predict a continuous value

  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

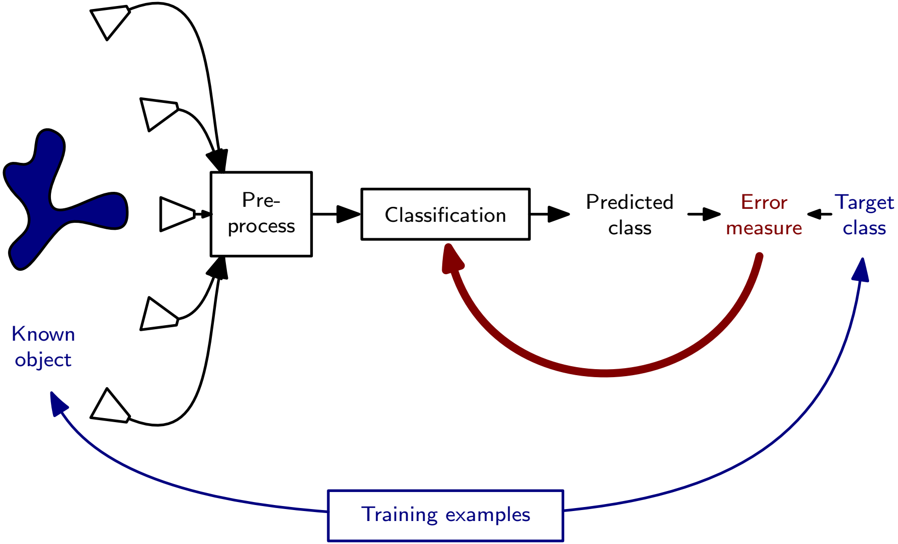
Another classification example



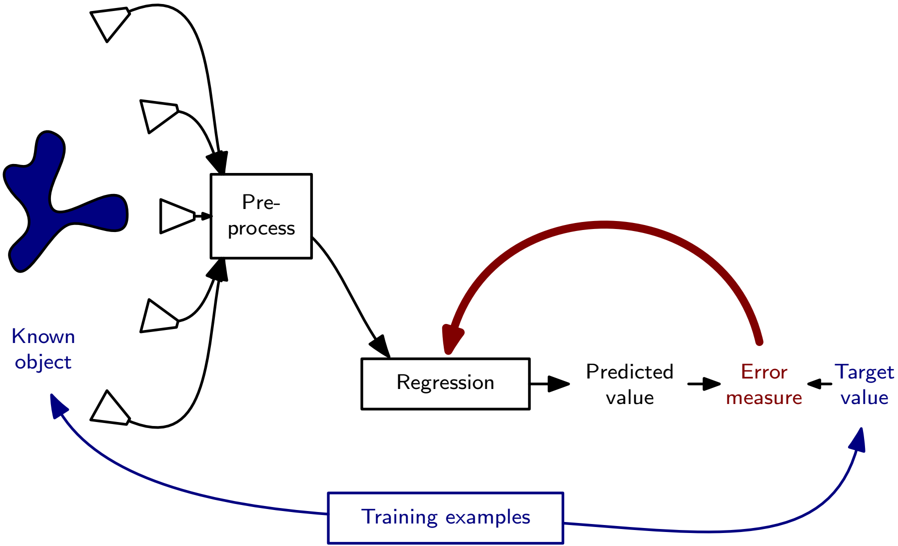
In functional form



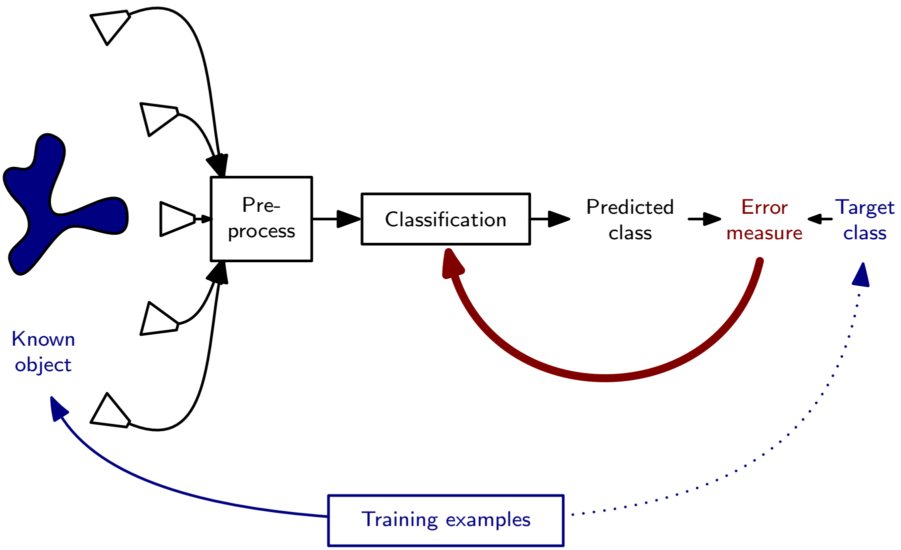
Supervised Training — Classification



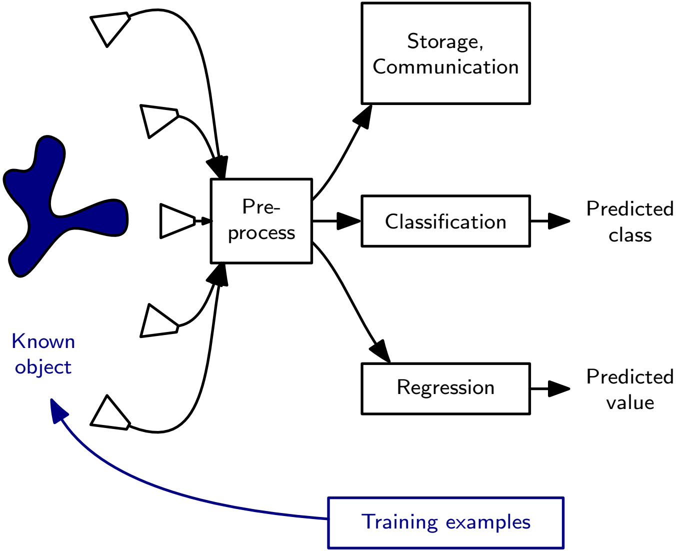
Supervised Training — Regression



Semi-supervised Training

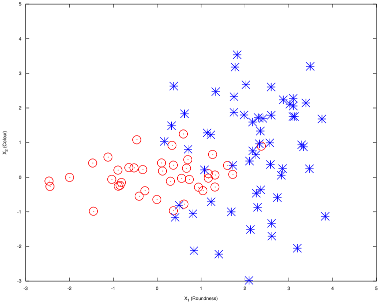


Unsupervised Training

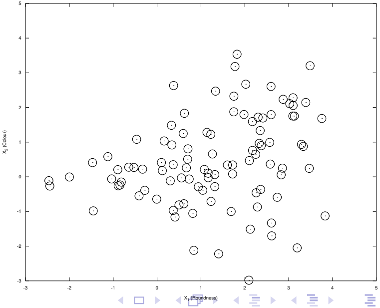


Supervised vs. Unsupervised

- 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:

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

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

Unsupervised: 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, ...)

Unsupervised: 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, . . .)

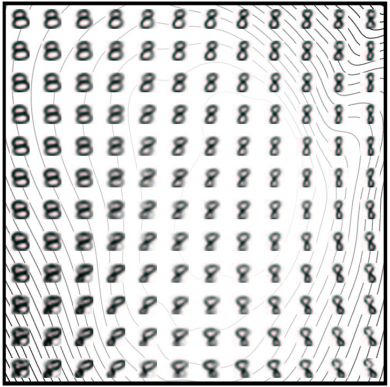
Unsupervised: Dimensionality reduction

- MNIST 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



Learning with delayed feedback

Sometimes the learning is part of a process

Example

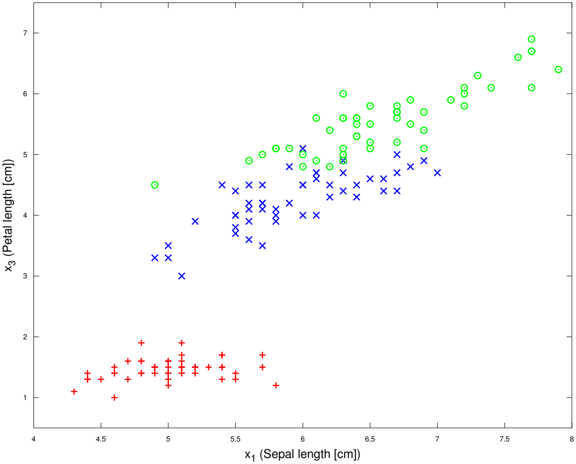
- Recommender systems, search engines: are the recommendations valuable?
- Game systems: what moves lead to a win?
- Robotics: What combinations of actions improve performance?

Reinforcement learning

- Such problems can be formalised as a sequence of steps (system states) that lead to a reward
- Reinforcement learning theory allows us to use that reward to learn good state transitions
- Complex states cannot be represented exactly $\Rightarrow$ dimensionality reduction, ...

Learning

What is it about the data that makes learning possible?



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

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?

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.

Occam's Razor

Inductive bias

- It is impossible to learn anything if we consider all possible hypotheses to explain the data:
- the best solution would simply memorise the data: it could not say anything about previously unseen examples

Occam's razor

Entia non sunt multiplicanda praeter necessitatem

"Entities should not be multiplied beyond necessity"

In other words: Keep it Simple

In practice: keep the *simplest* hypothesis that explains the data "well enough"

Occam's Razor

Inductive bias

- It is impossible to learn anything if we consider all possible hypotheses to explain the data:
- the best solution would simply memorise the data: it could not say anything about previously unseen examples

Occam's razor

Entia non sunt multiplicanda praeter necessitatem
"Entities should not be multiplied beyond necessity"

In other words: Keep it Simple

In practice: keep the *simplest* hypothesis that explains the data "well enough"

This course

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

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
- Lab: Introduction to the software environment
- Next week: linear discriminants


Are there any questions?