Inf2b Learning and Data
Lecture 1: Introduction to Learning and Data

Hiroshi Shimodaira
(Credit: Iain Murray and Steve Renals)

Centre for Speech Technology Research (CSTR)
School of Informatics
University of Edinburgh

Jan-Mar 2015
Welcome to Inf2b!

Inf2b: Algorithms, Data Structures (ADS), and Learning

Today’s Schedule:

1. Course structure
2. What is (machine) learning? (and why should you care?)
3. Administrative stuff
   - How to do well
4. Setting up a learning problem (time allowing)
Course structure

http://www.inf.ed.ac.uk/teaching/courses/inf2b/

Constituents:

- 30 lectures (including review)
- Tutorials starting in week 2
- 2 assessed assignments (with drop-in labs)

  CW1 (ADS): 12/02 – 06/03
  CW2 (Learning): 10/03 – 27/03

Equal split into two threads:

- Algorithms and Data Structures – KK (Kyriakos Kalorkoti)
  Design and analysis of correct, efficient and elegant data structures and algorithms.
- Learning – Hiroshi
  Building models that describe a data set and can make predictions about new data
1. Course structure

2. What is machine learning

3. Administrative stuff

4. Setting up a learning problem
Face detection

How would you detect a face?

(R. Vaillant, C. Monrocq and Y. LeCun, 1994)

http://demo.pittpatt.com/

How does album software tag your friends?
Viola–Jones Face detection (2001)

- Face detector consists of linear combination of 'weak' classifiers that utilise five types of primitive features.
- The detector is trained on a training data set of a large number of positive and negative samples.
- Scan the input image with a sub-window (24 x 24 pixels) to detect a face.

Taken from: http://ahprojects.com/projects/cv-dazzle
A nice demo: http://vimeo.com/12774628
Hiding from the machines (cameras)

The Viola-Jones face detector is fast, but has some drawbacks.

Taken from: http://ahprojects.com/projects/cv-dazzle
Applications of machine learning

Within informatics:

- **Vision**: as we’ve seen
- **Graphics**: increasingly data driven
- **AI & Natural Language Processing (NLP)**: text search/summarisation, speech recognition/synthesis, e.g. IBM Watson
- **Robotics**: vision, planning, control, . . .
- **Compilers**: learning how to optimise and beyond: data analysis across the sciences

Every day:

- Adverts / recommendations all over the web · · · Big Data
- Speech recognition and synthesis (e.g. Siri), Machine Translation, . . . with self-driving cars
Fit numbers in a program to data (i.e. train machines on data)
More robust than hand-fitted rules
Can’t approach humans at some tasks (e.g., vision)
Machines make better predictions in many other cases
Course structure

What is machine learning

Administrative stuff

Setting up a learning problem
Private study

- ~2 hours private study per lecture *in addition to tutorials & assignments*
- No required textbook for Inf2b There are notes. See those for recommended books.
- Required maths skills (especially linear algebra)
  Why should you remember and get familiar with maths formulas for machine learning?
    - Good understanding of the ideas
    - Guessing reasonable output of the model
    - Identifying/spotting the problems (bugs) with the system implemented
- Importance of programming practice [with Matlab or Python]
Maths skills

Useful webpage to check your maths:
http://www.mathsisfun.com/algebra

- Laws of exponents (Exponent rules)
  e.g. $x^m x^n = x^{m+n}$, $(x^m)^n = x^{mn}$

- Log and exponential
  e.g. $\log(x^n y^m) = n \log x + m \log y$, $e^{\ln x} = x$

- Quadratic Equations and their solutions
  e.g. $ax^2 + bx + c = 0$, $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

- Vectors $\mathbf{v} = (v_1, v_2, \ldots, v_d)^T$
  - Notation: column/row vectors, transpose
  - Addition and subtraction eg. $\mathbf{u} + \mathbf{v}$
  - Dot product (inner product) $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v}$

- Equation of a straight line, linear equations
Matrices \( A = (a_{ij}) \), \( A_{ij} = a_{ij} \)

- Addition, subtraction \( A + B, A - B \)
- Multiplication \((AB)_{ij} = \sum_{k=1}^{d} a_{ik}b_{kj}\)
- Transpose \((ABC)^T = C^T B^T A^T\)
- Determinant \(|A|\)
- Inverse \(A^{-1}A = AA^{-1} = I\)
- Eigen values and eigen vectors
- Vector spaces, subspaces, linear independence, basis and dimension, rank and nullity
- Linear transformations \(y = Ax\)

NB: See Section 4 of Learning Note No. 1 for the notation we use.
Two hours study this week?

Start to familiarise yourself with **Matlab** (or **Octave**)  
Introductory worksheet on the course website  
Many others at the end of a web search

**Love Python?** Learn **Numpy**+**SciPy**+**Matplotlib**  
(instead, or as well)

**Vital skills:**
- add, average, multiply vectors and matrices
- plot data stored in vectors
- save/read data to/from files
WANTED: Inf2b class reps (for ADS & Learning)

Email: h.shimodaira@ed.ac.uk
your name, degree, email address.
1 Course structure

2 What is machine learning

3 Administrative stuff

4 Setting up a learning problem
Classification of oranges and lemons
A two-dimensional space

Represent each sample as a point \((w, h)\) in a 2D space
Handwritten digits recognition

http://alex.seewald.at/digits/
A 64-dimensional space

Turn each cell into a number (somehow, see notes)
Unravel into a column vector, a **feature vector**
⇒ represented digit as point in 64D

\[ \mathbf{x} = (x_1, x_2, \ldots, x_{64})^T, \quad x_i \in [0, 127] \text{ or } x_i \in [0, 1] \]

http://alex.seewald.at/digits/
Euclidean distance

Distance between 2D vectors: \( \mathbf{u} = (u_1, u_2) \) and \( \mathbf{v} = (v_1, v_2) \)

\[
    r_2(\mathbf{u}, \mathbf{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2}
\]

Distance between \( d \)-dimensional vectors: \( \mathbf{u} \) and \( \mathbf{v} \)

\[
    r_2(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{k=1}^{d} (u_k - v_k)^2}
\]

Measures similarities between feature vectors
i.e., similarities between digits, movies, sounds, galaxies, ...
Have high-resolution scans of digits.

How many pixels should be sampled?

What are pros and cons of:

2×2, 4×4, 16×16, or 100×100?
Accountability