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) - Tuesdays, Thursdays, Fridays
- Tutorials (starting in week 2)
- Drop-in labs for Learning (Fridays 14:10-15:00)
- 2 assessed assignments (with drop-in labs)

  CW2 (Learning): 13/Mar. – 04/Apr.

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
Drop-in labs for Learning

- Fridays 14:10-15:00 in AT-5.05 (West Lab)
  Starting in Week 1 (on Friday 19th January)
- Worksheets available on the course webpage
- Purposes of lab sessions
  - Assistance in understanding basic algorithms and
techniques of machine learning and data analysis
  - Assistance in programming with Matlab
  - Assistance in working on the 2nd assignment (CW2)
- Practice on machine learning using Matlab
  - Work on toy problems for the topics taught in the course
- Demonstrator: Andreas Kapourain
Face detection

How would you detect a face?

How does album software tag your friends?

http://demo.pittpatt.com/

(R. Vaillant, C. Monrocq and Y. LeCun, 1994)
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. [another e.g.]
- **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, Echo), 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
Private study

- ~2 hours private study per lecture in addition to tutorials & assignments
- No required textbook for Inf2b There are notes and slides. See those for recommended books.
- Importance of maths skills (especially 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] (attend the drop-in labs!)
Warning: Inf2b is NOT an easy course

Inf2b requires a solid maths background:
- Linear Algebra
- Calculus
- Probability

Independent learning (self-directed learning) is essential.

See the following page regarding differences between secondary-school and university in terms of learning style and what is expected from you as a student.

https://www.birmingham.ac.uk/accessibility/transcripts/school-uni-differences.aspx
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**
Maths skills (cont.)

- 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

- **Learn Matlab** try the lab sheets for the 1st lab this week.

- **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
Have a look at the lecture note and slides in advance to the lecture.

Have questions prepared to ask.

Laptops, tablets, phones are not allowed to use during lectures unless permitted.

NB: lectures will be recorded, and videos will be published in a few days after the lecture.
Classification of oranges and lemons
A two-dimensional space

Represent each sample as a point \((w, h)\) in a 2D space.
Pixel image to a feature vector

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

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

http://alex.seewald.at/digits/
Image data as a point in a vector space

\[ \mathbf{x} = (2, 6, 5)^T \]
Euclidean distance

Distance between 2D vectors: \( u = (u_1, u_2)^T \) and \( v = (v_1, v_2)^T \)

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

Distance between \( d \)-dimensional vectors: \( u = (u_1, \ldots, u_d)^T \) and \( v = (v_1, \ldots, v_d)^T \)

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

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

How many pixels should be sample?

What are pros and cons of:

$2 \times 2, \ 4 \times 4, \ 16 \times 16, \ or \ 100 \times 100$?
Accountability

Example of image resolutions
Try the exercises in the lecture note 1.

No solutions will be published.

In case you’re not sure if your answers are correct.

- Discuss them with your classmates
- Use the Inf2b-Learning discussion board on Piazza
Self-study both ADS and Learning everyday.

Drop-in for Learning starts on Friday, 19th Jan at 14:10. Try the worksheets before the lab.

Tutorial starts in Week 2.

Discussion forum for “Learning” in Piazza

Office hours: Wednesdays at 14:00-15:00 (TBC) in IF-3.04