Lecture 1: Introduction to Machine Learning

4F13: Machine Learning

Zoubin Ghahramani and Carl Edward Rasmussen

Department of Engineering
University of Cambridge

http://mlg.eng.cam.ac.uk/teaching/4f13/

What is machine learning?

- *Machine learning is an interdisciplinary field focusing on both the mathematical foundations and practical applications of systems that learn, reason and act.*
- Other related terms: Pattern Recognition, Neural Networks, Data Mining, Statistical Modelling ...
- Using ideas from: Statistics, Computer Science, Engineering, Applied Mathematics, Cognitive Science, Psychology, Computational Neuroscience, Economics
- The goal of these lectures: to introduce important concepts, models and algorithms in machine learning.
- For more: Go to `talks.cam.ac.uk`, search for "Machine Learning" for various reading groups, lectures, and seminars. Open to anyone interested. Or go to `videolectures.net` for videos and slides of relevant talks.
- MSc and PhDs: MSc programmes at Edinburgh and UCL. Good places to do PhD: Cambridge, UCL, Edinburgh, Oxford, Sheffield, KCL... Also: Berkeley, CMU, Stanford, MIT, Toronto ...

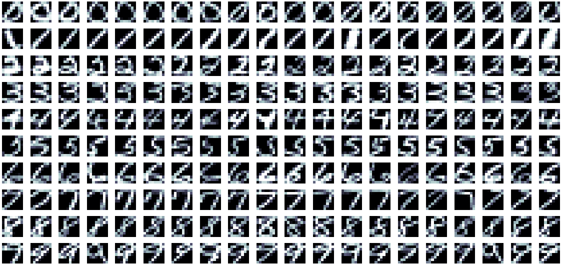
Warning!

Lecture 1 will overlap somewhat with my lectures in 3F3: Pattern Processing—but don't despair, a lot of new material later!

What is machine learning useful for?

Automatic speech recognition

Computer vision: e.g. object, face and handwriting recognition



(NORB image from Yann LeCun)

Information retrieval

Google Search: Unsupervised Learning http://www.google.com/search?q=Unsupervised+Learning&sourceid=fir...

Google

Web Images Groups News Froogle more »

Unsupervised Learning Search Advanced Search Preferences

Web Results **1** - **10** of about **150,000** for **Unsupervised Learning**. (**0.27** seconds)

Mixture modelling, Clustering, Intrinsic classification ...
Mixture Modelling page. Welcome to David Dowe's clustering, mixture modelling and **unsupervised learning** page. Mixture modelling (or ...
www.csse.monash.edu.au/~dld/mixture.modelling.page.html - 26k - 4 Oct 2004 - Cached - Similar pages

ACL'99 Workshop -- **Unsupervised Learning** in Natural Language ...
PROGRAM. ACL'99 Workshop **Unsupervised Learning** in Natural Language Processing. University of Maryland June 21, 1999. Endorsed by SIGNLL ...
www.ai.sri.com/~kehler/unsup-acl-99.html - 5k - Cached - Similar pages

Unsupervised learning and Clustering
cgm.cs.mcgill.ca/~soss/cs644/projects/wijhe/ - 1k - Cached - Similar pages

NIPS*98 Workshop - Integrating Supervised and **Unsupervised** ...
NIPS*98 Workshop "Integrating Supervised and **Unsupervised Learning**" Friday, December 4, 1998. ... 4:45-5:30, Theories of **Unsupervised Learning** and Missing Values. ...
www-2.cs.cmu.edu/~mccallum/supunsup/ - 7k - Cached - Similar pages

NIPS Tutorial 1999
Probabilistic Models for **Unsupervised Learning** Tutorial presented at the 1999 NIPS Conference by Zoubin Ghahramani and Sam Roweis. ...
www.gatsby.ucl.ac.uk/~zoubin/NIPStutorial.html - 4k - Cached - Similar pages

Gatsby Course: **Unsupervised Learning** : Homepage
Unsupervised Learning (Fall 2000). ... Syllabus (resources page): 10/10 1 - Introduction to **Unsupervised Learning** Geoff project: (ps, pdf). ...
www.gatsby.ucl.ac.uk/~quaid/course/ - 15k - Cached - Similar pages
[More results from www.gatsby.ucl.ac.uk]

[PDF] **Unsupervised Learning** of the Morphology of a Natural Language
File Format: PDF/Adobe Acrobat - View as HTML
Page 1. Page 2. Page 3. Page 4. Page 5. Page 6. Page 7. Page 8. Page 9. Page 10. Page 11. Page 12. Page 13. Page 14. Page 15. Page 16. Page 17. Page 18. Page 19 ...
acl.ldc.upenn.edu/J/J01/J01-2001.pdf - Similar pages

Unsupervised Learning - The MIT Press
... From Bradford Books: **Unsupervised Learning** Foundations of Neural Computation Edited by Geoffrey Hinton and Terrence J. Sejnowski Since its founding in 1989 by ...
mitpress.mit.edu/book-home.tcl?isbn=026258168X - 13k - Cached - Similar pages

[PS] **Unsupervised Learning** of Disambiguation Rules for Part of
File Format: Adobe PostScript - View as Text
Unsupervised Learning of Disambiguation Rules for Part of. Speech Tagging. Eric Brill. 1. ... It is possible to use **unsupervised learning** to train stochastic. ...
www.cs.jhu.edu/~brill/acl-wkshp.ps - Similar pages

The **Unsupervised Learning** Group (ULG) at UT Austin
The **Unsupervised Learning** Group (ULG). What ? The **Unsupervised Learning** Group (ULG) is a group of graduate students from the Computer ...
www.lans.ece.utexas.edu/ulg/ - 14k - Cached - Similar pages

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 Next

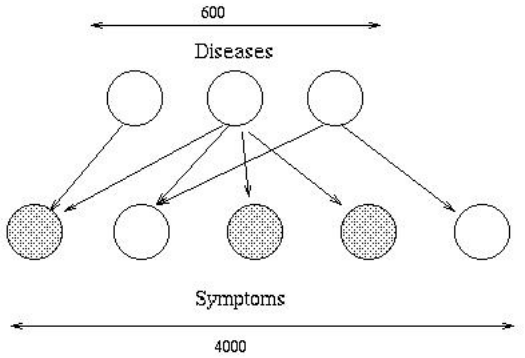
1 of 2 06/10/04 15:44

Web Pages

Retrieval
Categorisation
Clustering
Relations between pages

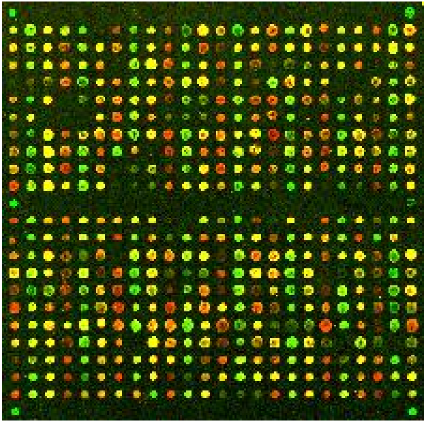
Financial prediction

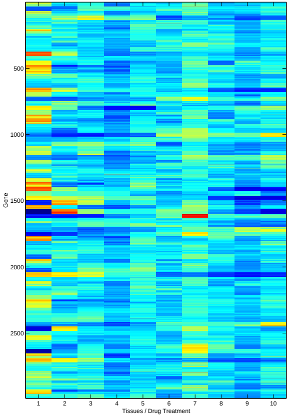
Medical diagnosis



(image from Kevin Murphy)

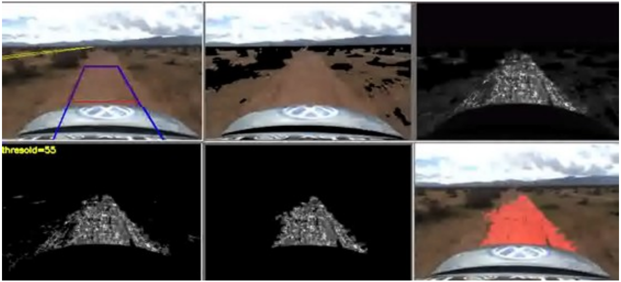
Bioinformatics





e.g. modelling gene microarray data, protein structure prediction

Robotics



DARPA $2m Grand Challenge

Movie recommendation systems

Challenge: to improve the accuracy of movie preference predictions

Netflix $1m Prize; competition 2006-2009. Netflix 2 contest coming up!

Three Types of Learning

Imagine an organism or machine which experiences a series of sensory inputs:

$$x_1, x_2, x_3, x_4, \ldots$$

Supervised learning: The machine is also given desired outputs $y_1, y_2, \ldots$, and its goal is to learn to produce the correct output given a new input.

Unsupervised learning: The goal of the machine is to build a model of x that can be used for reasoning, decision making, predicting things, communicating etc.

Reinforcement learning: The machine can also produce actions $a_1, a_2, \ldots$ which affect the state of the world, and receives rewards (or punishments) $r_1, r_2, \ldots$. Its goal is to learn to act in a way that maximises rewards in the long term.

(In this course we'll focus mostly on unsupervised learning and reinforcement learning.)

Key Ingredients

Data

We will represent data by vectors in some vector space[1]

Let $\mathbf{x}$ denote a data point with elements $\mathbf{x} = (x_1, x_2, \ldots, x_D)$

The elements of $\mathbf{x}$, e.g. x_d, represent measured (observed) features of the data point; D denotes the number of measured features of each point.

The data set $\mathcal{D}$ consists of N data points:

$$\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \ldots, \mathbf{x}^{(N)}\}$$

[1] This assumption can be relaxed.

Key Ingredients

Data

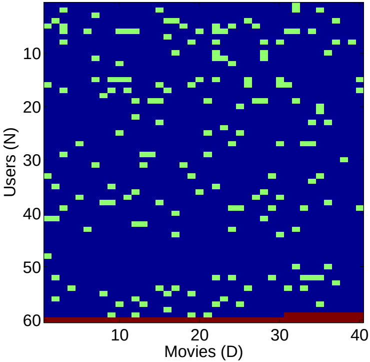
Let $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ denote a data point, and $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, a data set

Predictions

We are generally interested in predicting something based on the observed data.

Given $\mathcal{D}$ what can we say about $\mathbf{x}^{(N+1)}$?

Given $\mathcal{D}$ and $x_1^{(N+1)}, x_2^{(N+1)}, \ldots, x_{D-1}^{(N+1)}$, what can we say about $x_D^{(N+1)}$?



Key Ingredients

Data

Let $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ be a data point, and $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \ldots, \mathbf{x}^{(N)}\}$, a data set

Predictions

We are interested in predicting something based on the observed data set.
Given $\mathcal{D}$ what can we say about $\mathbf{x}^{(N+1)}$?
Given $\mathcal{D}$ and $x_1^{(N+1)}, x_2^{(N+1)}, \ldots, x_{D-1}^{(N+1)}$, what can we say about $x_D^{(N+1)}$?

Model

To make predictions, we need to make some *assumptions*. We can often express these assumptions in the form of a model, with some parameters, θ.
Given data $\mathcal{D}$, we learn the model parameters θ, from which we can predict new data points.
The model can often be expressed as a *probability distribution over data points*

Basic Rules of Probability

Let X be a random variable taking values x in some set $\mathcal{X}$.
Probabilities are non-negative $P(X = x) \geqslant 0 \ \forall x$.
Probabilities normalise: $\sum_{x \in \mathcal{X}} P(X = x) = 1$ for distributions if x is a discrete variable and $\int_{-\infty}^{+\infty} p(x)dx = 1$ for probability densities over continuous variables

The joint probability of $X = x$ and $Y = y$ is: $P(X = x, Y = y)$.
The marginal probability of $X = x$ is: $P(X = x) = \sum_y P(X = x, y)$, assuming y is discrete. I will generally write $P(x)$ to mean $P(X = x)$.
The conditional probability of x given y is: $P(x|y) = P(x, y)/P(y)$

Bayes Rule:

$$P(x, y) = P(x)P(y|x) = P(y)P(x|y) \quad \Rightarrow \quad \boxed{P(y|x) = \frac{P(x|y)P(y)}{P(x)}}$$

Information, Probability and Entropy

Information is the reduction of uncertainty. How do we measure uncertainty? Some axioms (informally):

- if something is certain, its uncertainty = 0
- uncertainty should be maximum if all choices are equally probable
- uncertainty (information) should add for independent sources

This leads to a discrete random variable X having uncertainty equal to the entropy function:

$$H(X) = -\sum_{x \in \mathcal{X}} P(X = x) \log P(X = x)$$

measured in *bits* (**bi**nary digi**ts**) if the base 2 logarithm is used or *nats* (**na**tural digi**ts**) if the natural (base e) logarithm is used.

Some Definitions Relating to Information Theory

- Surprise (for event $X = x$): $-\log P(X = x)$
- Entropy = average surprise: $H(X) = -\sum_{x \in \mathcal{X}} P(X = x) \, \log P(X = x)$
- Conditional entropy

$$H(X|Y) = -\sum_{x} \sum_{y} P(x, y) \log P(x|y)$$

- Mutual information

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X, Y)$$

- Independent random variables: $P(x, y) = P(x)P(y) \, \forall x \, \forall y$

How do we relate information theory and probabilistic modelling?

The source coding problem

Imagine we have a set of symbols $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}, \mathtt{h}\}$.
We want to transmit these symbols over some binary communication channel, i.e. using a sequence of bits to represent the symbols.
Since we have 8 symbols, we could use 3 bits per symbol ($2^3 = 8$). For example:
$\mathtt{a} = 000$, $\mathtt{b} = 001$, $\mathtt{c} = 010$, $\ldots$, $\mathtt{h} = 111$

Is this optimal?

What if some symbol, a, is much more probable than other symbols, e.g. f?
Shouldn't we use fewer bits to transmit the more probable symbols?

Think of a discrete variable X taking on values in $\mathcal{X}$, having probability distribution $P(X)$.

How does the probability distribution $P(X)$ relate to the number of bits we need for each symbol to optimally and losslessly transmit symbols from $\mathcal{X}$?

Shannon's Source Coding Theorem

A discrete random variable X, distributed according to $P(X)$ has entropy:

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log_2 P(x)$$

Shannon's source coding theorem: Consider a random variable X, with entropy $H(X)$. A sequence of n independent draws from X can be losslessly compressed into a minimum expected code of length $n\mathcal{L}$ bits, where $H(X) \leqslant \mathcal{L} < H(X) + \frac{1}{n}$.

If each symbol is given a code length $l(x) = -\log_2 Q(x)$ then the expected per-symbol length $\mathcal{L}_Q$ of the code is

$$\mathcal{L}_Q = \sum_x P(x) l(x) = -\sum_x P(x) \log_2 Q(x) = H(X) + \mathrm{KL}(P\|Q),$$

where the relative-entropy or Kullback-Leibler divergence is

$$\mathrm{KL}(P\|Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)} \geqslant 0$$

Take home message: better probabilistic models $\equiv$ more efficient codes

Some distributions

Univariate Gaussian density ($x \in \mathbb{R}$):

$$p(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

Multivariate Gaussian density ($\mathbf{x} \in \mathbb{R}^D$):

$$p(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Bernoulli distribution ($x \in \{0, 1\}$):

$$p(x|\theta) = \theta^x (1-\theta)^{1-x}$$

Discrete distribution ($x \in \{1, \ldots L\}$):

$$p(x|\theta) = \prod_{\ell=1}^{L} \theta_\ell^{\delta(x,\ell)}$$

where $\delta(a, b) = 1$ iff $a = b$, and $\sum_{\ell=1}^{L} \theta_\ell = 1$ and $\theta_\ell \geqslant 0 \ \forall \ell$.

Some distributions (cont)

Uniform ($x \in [a, b]$):

$$p(x|a,b) = \begin{cases} \frac{1}{b-a} & \text{if } a \leqslant x \leqslant b \\ 0 & \text{otherwise} \end{cases}$$

Gamma ($x \geqslant 0$):

$$p(x|a,b) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp\{-bx\}$$

Beta ($x \in [0, 1]$):

$$p(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

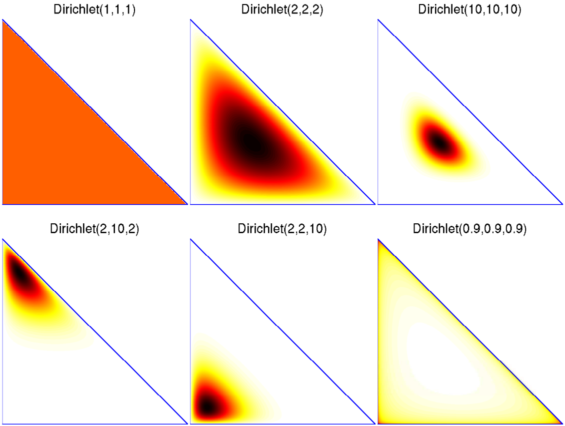
where $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is the gamma function, a generalisation of the factorial: $\Gamma(n) = (n-1)!$.

Dirichlet ($\mathbf{p} \in \mathbb{R}^D$, $p_d \geqslant 0$, $\sum_{d=1}^{D} p_d = 1$):

$$p(\mathbf{p}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \prod_{d=1}^{D} p_d^{\alpha_d - 1}$$

Dirichlet Distributions

Examples of Dirichlet distributions over $\mathbf{p} = (p_1, p_2, p_3)$ which can be plotted in 2D since $p_3 = 1 - p_1 - p_2$:



Other distributions you should know about...

Exponential family of distributions:

$$P(\mathbf{x}|\theta) = f(\mathbf{x})\; g(\theta) \exp\left\{ \boldsymbol{\phi}(\theta)^\top \mathbf{u}(\mathbf{x}) \right\}$$

where $\boldsymbol{\phi}(\theta)$ is the vector of *natural parameters*, $\mathbf{u}$ are *sufficient statistics*

- Binomial
- Multinomial
- Poisson
- ...

End Notes

It is very important that you *understand* all the material in the following cribsheet: `http://learning.eng.cam.ac.uk/zoubin/ml06/cribsheet.pdf`

Here is a useful statistics / pattern recognition glossary:
`http://alumni.media.mit.edu/~tpminka/statlearn/glossary/glossary.html`