Unsupervised Learning

Week 1: Introduction, Statistical Basics, and a bit of Information Theory

Zoubin Ghahramani
zoubin@gatsby.ucl.ac.uk

**Gatsby Computational Neuroscience Unit, and
MSc in Intelligent Systems, Dept Computer Science
University College London**

Autumn 2003

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 representations 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.

Goals of Supervised Learning

Classification: The desired outputs y_i are discrete class labels.
The goal is to classify new inputs correctly (i.e. to generalize).

Regression: The desired outputs y_i are continuous valued.
The goal is to predict the output accurately for new inputs.

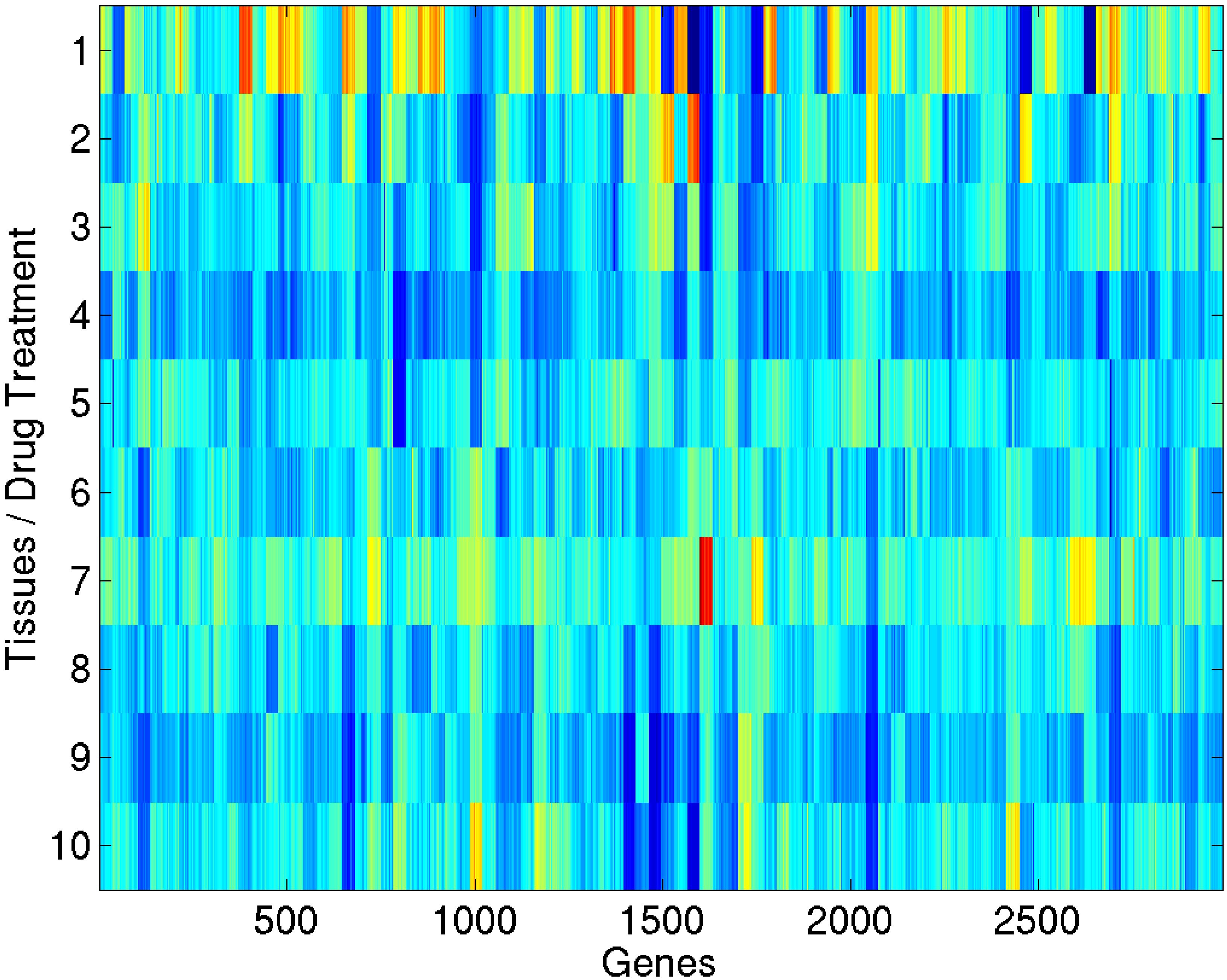
Goals of Unsupervised Learning

To find useful representations of the data, for example:

- finding clusters
- dimensionality reduction
- finding the hidden causes or sources of the data
- modeling the data density

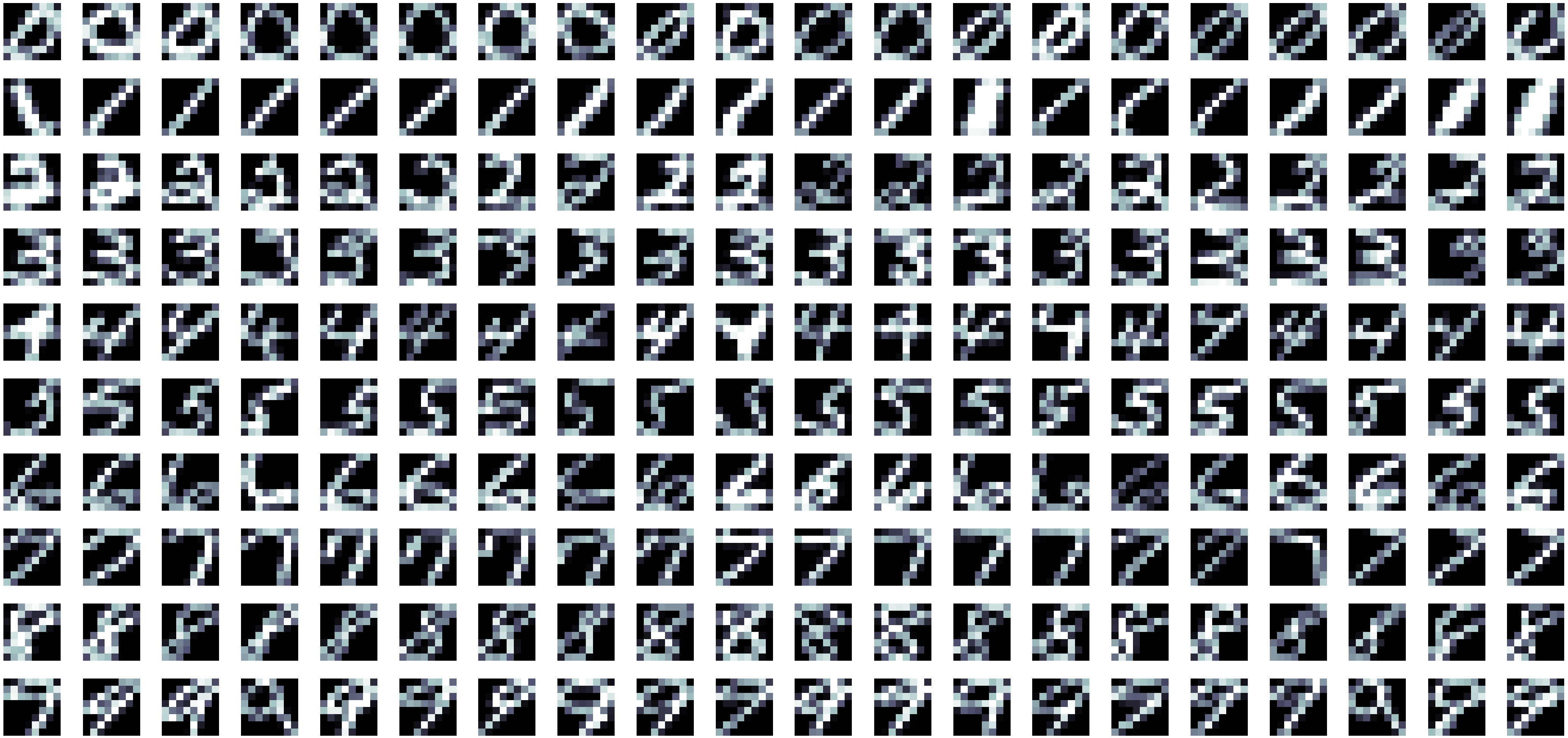
Uses of Unsupervised Learning

- data compression
- outlier detection
- classification
- make other learning tasks easier
- a theory of human learning and perception



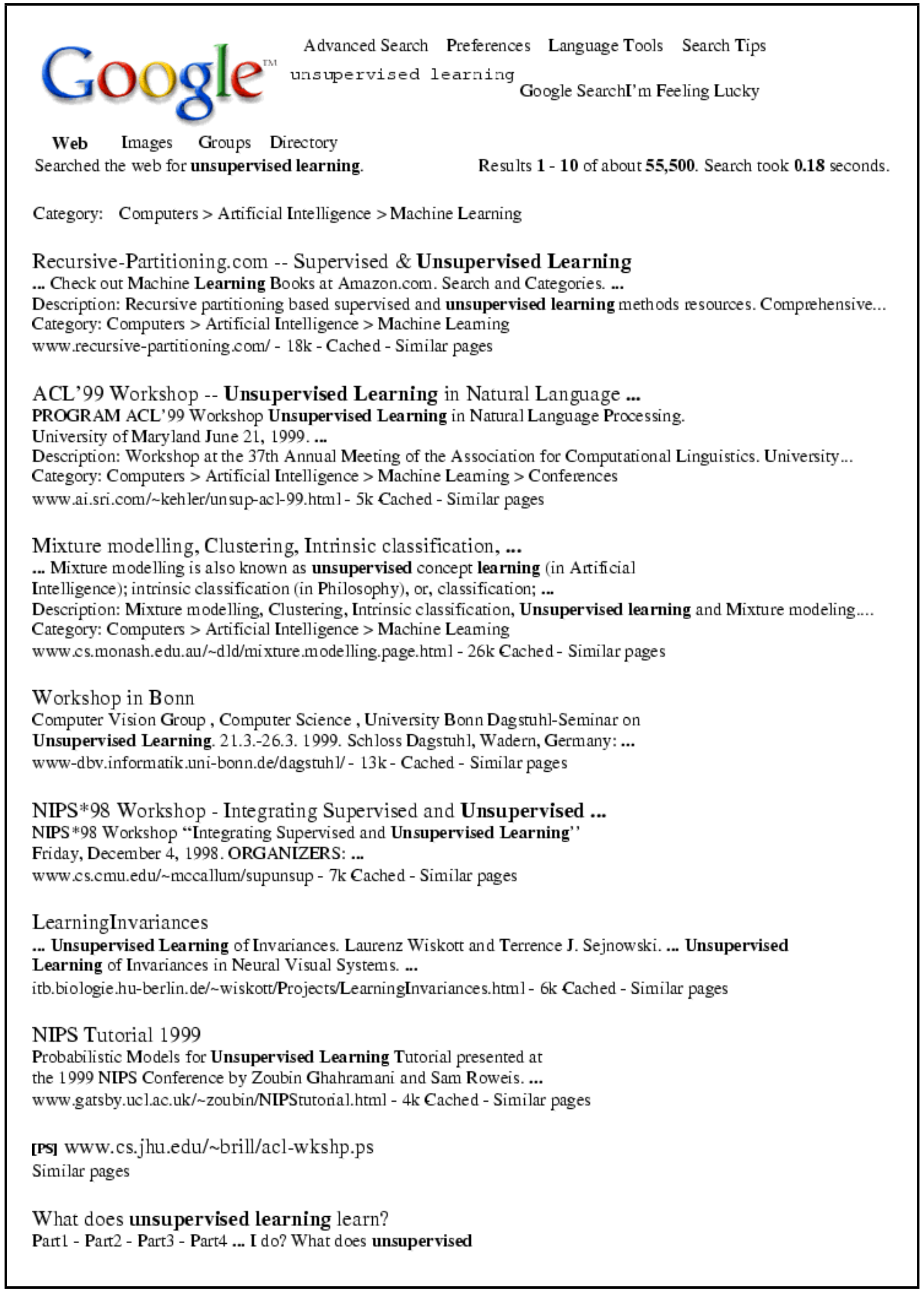
Tissues / Drug Treatment
1
2
3
4
5
6
7
8
9
10
500
1000
1500
2000
2500
Genes

Handwritten Digits



Web Pages

Categorisation
Clustering
Relations between pages



Advanced Search Preferences Language Tools Search Tips

unsupervised learning

Google SearchI'm Feeling Lucky

Web Images Groups Directory

Searched the web for **unsupervised learning**. Results **1** - **10** of about **55,500**. Search took **0.18** seconds.

Category: Computers > Artificial Intelligence > Machine Learning

Recursive-Partitioning.com -- Supervised & **Unsupervised Learning**
... Check out Machine **Learning** Books at Amazon.com. Search and Categories. ...
Description: Recursive partitioning based supervised and **unsupervised learning** methods resources. Comprehensive...
Category: Computers > Artificial Intelligence > Machine Learning
www.recursive-partitioning.com/ - 18k - 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. ...
Description: Workshop at the 37th Annual Meeting of the Association for Computational Linguistics. University...
Category: Computers > Artificial Intelligence > Machine Learning > Conferences
www.ai.sri.com/~kehler/unsup-acl-99.html - 5k Cached - Similar pages

Mixture modelling, Clustering, Intrinsic classification, ...
... Mixture modelling is also known as **unsupervised** concept **learning** (in Artificial
Intelligence); intrinsic classification (in Philosophy), or, classification; ...
Description: Mixture modelling, Clustering, Intrinsic classification, **Unsupervised learning** and Mixture modeling....
Category: Computers > Artificial Intelligence > Machine Learning
www.cs.monash.edu.au/~dld/mixture.modelling.page.html - 26k Cached - Similar pages

Workshop in Bonn
Computer Vision Group , Computer Science , University Bonn Dagstuhl-Seminar on
Unsupervised Learning. 21.3.-26.3. 1999. Schloss Dagstuhl, Wadern, Germany: ...
www-dbv.informatik.uni-bonn.de/dagstuhl/ - 13k - Cached - Similar pages

NIPS*98 Workshop - Integrating Supervised and **Unsupervised** ...
NIPS*98 Workshop "Integrating Supervised and **Unsupervised Learning**"
Friday, December 4, 1998. ORGANIZERS: ...
www.cs.cmu.edu/~mccallum/supunsup - 7k Cached - Similar pages

LearningInvariances
... **Unsupervised Learning** of Invariances. Laurenz Wiskott and Terrence J. Sejnowski. ... **Unsupervised**
Learning of Invariances in Neural Visual Systems. ...
itb.biologie.hu-berlin.de/~wiskott/Projects/LearningInvariances.html - 6k 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

[PS] www.cs.jhu.edu/~brill/acl-wkshp.ps
Similar pages

What does **unsupervised learning** learn?
Part1 - Part2 - Part3 - Part4 ... I do? What does **unsupervised**

Why a statistical approach?

- A probabilistic model of the data can be used to
  - make inferences about missing inputs
  - generate predictions/fantasies/imagery
  - make decisions which minimise expected loss
  - communicate the data in an efficient way
- Statistical modelling is equivalent to other views of learning:
  - information theoretic: finding compact representations of the data
  - physical analogies: minimising free energy of a corresponding statistical mechanical system

Information, Probability and Entropy

Information is the reduction of uncertainty. How do we measure uncertainty?

Some axioms (informal):

- 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:

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

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 and Intuitions

- Surprise (for event $X = x$): $-\log P(X = x)$
- Entropy = average surpise: $H(X) = -\sum_{X=x} P(X = x)\ \log_2 P(X = x)$
- Conditional entropy

$$H(X|Y) = -\sum_x \sum_y P(x, y) \log_2 P(x|y)$$

- Mutual information

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

- Kullback-Leibler divergence (relative entropy)

$$KL(P(X)\|Q(X)) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

- Relation between Mutual information and KL: $I(X;Y) = KL(P(X,Y)\|P(X)P(Y))$
- Independent random variables: $P(X,Y) = P(X)P(Y)$
- Conditional independence $X \perp\!\!\!\perp Y|Z$ (X conditionally independent of Y given Z) means $P(X,Y|Z) = P(X|Z)P(Y|Z)$ and $P(X|Y,Z) = P(X|Z)$

Shannon's Source Coding Theorem

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

$$H(X) = -\sum_x P(x) \log P(x)$$

Shannon's source coding theorem: n independent samples of the random variable X, with entropy $H(X)$, can be compressed into minimum expected code of length $n\mathcal{L}$, where

$$H(X) \leq \mathcal{L} < H(X) + \frac{1}{n}$$

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

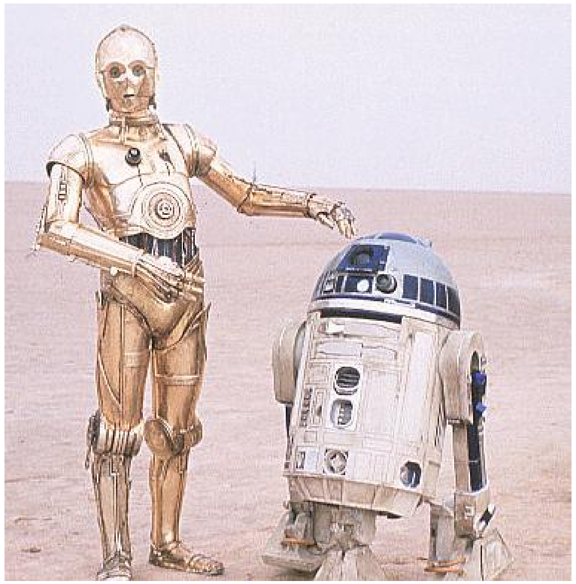
$$H(X) + KL(P\|Q) \leq \mathcal{L}_Q < H(X) + KL(P\|Q) + \frac{1}{n},$$

where the relative-entropy or Kullback-Leibler divergence is

$$KL(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \geq 0$$

Learning: A Statistcal Approach II

- Goal: to represent the beliefs of learning agents.
- Cox Axioms lead to the following:
  If plausibilities/beliefs are represented by real numbers, then the only reasonable and consistent way to manipulate them is Bayes rule.
- Frequency vs belief interpretation of probabilities
- The Dutch Book Theorem:
  If you are willing to bet on your beliefs, then unless they satisfy Bayes rule there will always be a set of bets ("Dutch book") that you would accept which is guaranteed to lose you money, no matter what outcome!



Desiderata (or Axioms) for Computing Plausibilities / Degrees of Belief

Paraphrased from E. T. Jaynes, using the notation $p(A|B)$ is the plausibility of statement A given that you know that statement B is true.

- Degrees of plausibility are represented by real numbers
- Qualitative correspondence with common sense, e.g.
  - If $p(A|C') > p(A|C)$ but $p(B|A\&C') = p(B|A\&C)$ then $p(A\&B|C') \geq p(A\&B|C)$
- Consistency:
  - If a conclusion can be reasoned in more than one way, then every possible way must lead to the same result.
  - All available evidence should be taken into account when inferring a plausibility.
  - Equivalent states of knowledge should be represented with equivalent plausibility statements.

Accepting these desiderata leads to Bayes Rule being the only way to manipulate plausibilities.

Bayes Rule

Probabilities are non-negative $P(x) \geq 0 \; \forall x$.

Probabilities normalise: $\sum_x P(x) = 1$ for discrete distributions and $\int p(x)dx = 1$ for probability densities.

The joint probability of x and y is: $P(x, y)$.

The marginal probability of x is: $P(x) = \sum_y P(x, y)$.

The conditional probability of x given y is: $P(x|y) = P(x, y)/P(y)$

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

Bayesian Learning

Data $\mathcal{D}$, model class $\mathcal{M}$, model parameters θ. The likelihood and parameter priors are combined into the posterior for a particular model, batch and online:

$$p(\theta|\mathcal{D},\mathcal{M}) = \frac{p(\mathcal{D}|\theta,\mathcal{M})p(\theta|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})} \qquad p(\theta|\mathcal{D},x,\mathcal{M}) = \frac{p(x|\theta,\mathcal{D},\mathcal{M})p(\theta|\mathcal{D},\mathcal{M})}{p(x|\mathcal{D},\mathcal{M})}$$

Predictions are made by integrating over the posterior:

$$p(x|\mathcal{D},\mathcal{M}) = \int d\theta \; p(x|\theta,\mathcal{M}) \; p(\theta|\mathcal{D},\mathcal{M}).$$

To compare models, we again use Bayes' rule and the prior on models

$$p(\mathcal{M}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}) \; p(\mathcal{M})$$

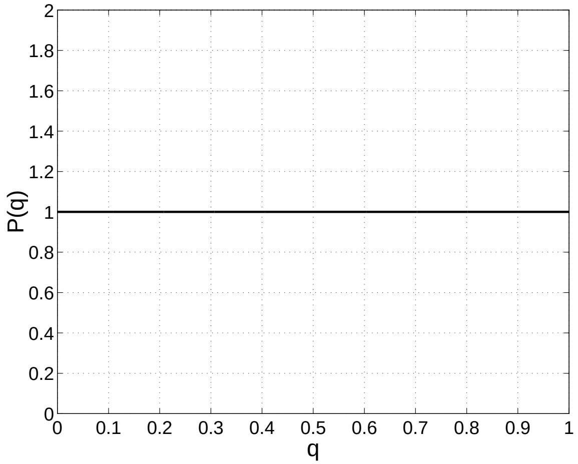
This also requires an integral over θ:

$$p(\mathcal{D}|\mathcal{M}) = \int d\theta \; p(\mathcal{D}|\theta,\mathcal{M}) \; p(\theta|\mathcal{M})$$

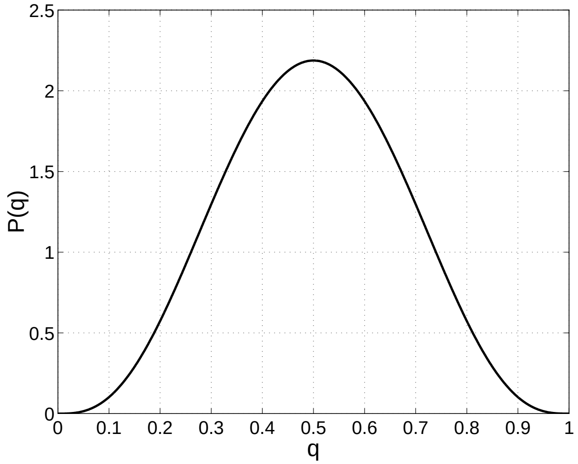
For interesting models, these integrals may be difficult to compute.

Bayesian Learning: A coin toss example

Coin toss: One parameter q — the odds of obtaining *heads*
So our space of models is the set $q \in [0, 1]$.
Learner A believes all values of q are equally plausible;
Learner B believes that it is more plausible that the coin is "fair" $(q \approx 0.5)$ than "biased".



A



B

These priors beliefs can be described by the Beta distribution:

$$p(q|\alpha_1, \alpha_2) = \frac{q^{(\alpha_1 - 1)}(1-q)^{(\alpha_2 - 1)}}{B(\alpha_1, \alpha_2)} = \mathrm{Beta}(q|\alpha_1, \alpha_2)$$

for A: $\alpha_1 = \alpha_2 = 1.0$ and B: $\alpha_1 = \alpha_2 = 4.0$.

Bayesian Learning: The coin toss (cont)

Two possible outcomes:

$$p(\text{heads}|q) = q \qquad p(\text{tails}|q) = 1 - q \tag{1}$$

Imagine we observe a single coin toss and it comes out *heads*
The probability of the observed data (likelihood) is:

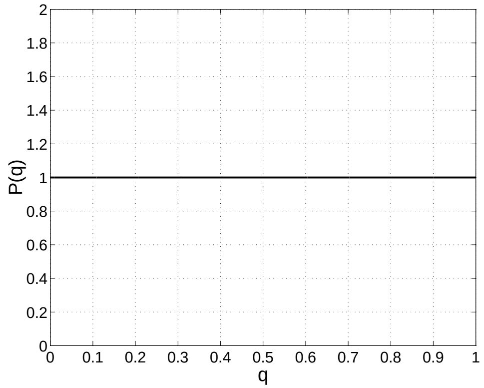
$$p(\text{heads}|q) = q \tag{2}$$

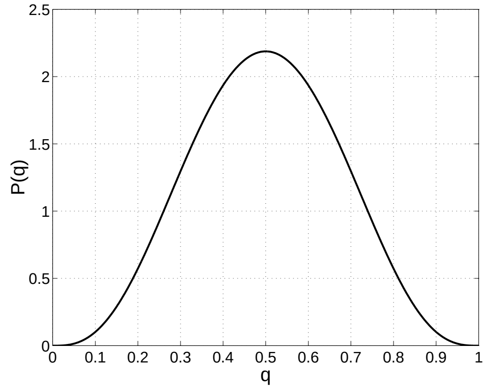
Using Bayes Rule, we multiply the prior, $p(q)$ by the likelihood and renormalise to get the posterior probability:

$$\begin{aligned} p(q|\text{heads}) &= \frac{p(q)p(\text{heads}|q)}{p(\text{heads})} \propto q \, \mathrm{Beta}(q|\alpha_1, \alpha_2) \\ &\propto q \, q^{(\alpha_1 - 1)}(1-q)^{(\alpha_2 - 1)} = \mathrm{Beta}(q|\alpha_1 + 1, \alpha_2) \end{aligned}$$

Bayesian Learning: The coin toss (cont)

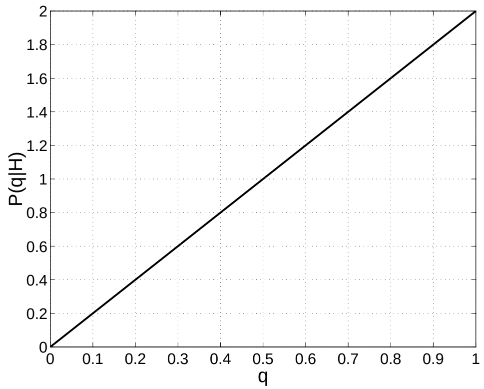
Prior

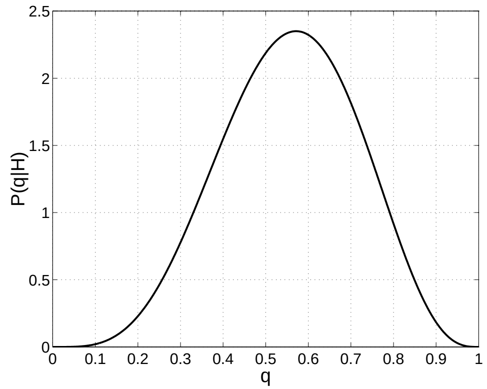




A

B





Posterior

Some Terminology

Maximum Likelihood (ML) Learning: Does not assume a prior over the model parameters. Finds a parameter setting that maximises the likelihood of the data: $P(\mathcal{D}|\theta)$.

Maximum a Posteriori (MAP) Learning: Assumes a prior over the model parameters $P(\theta)$. Finds a parameter setting that maximises the posterior: $P(\theta|\mathcal{D}) \propto P(\theta)P(\mathcal{D}|\theta)$.

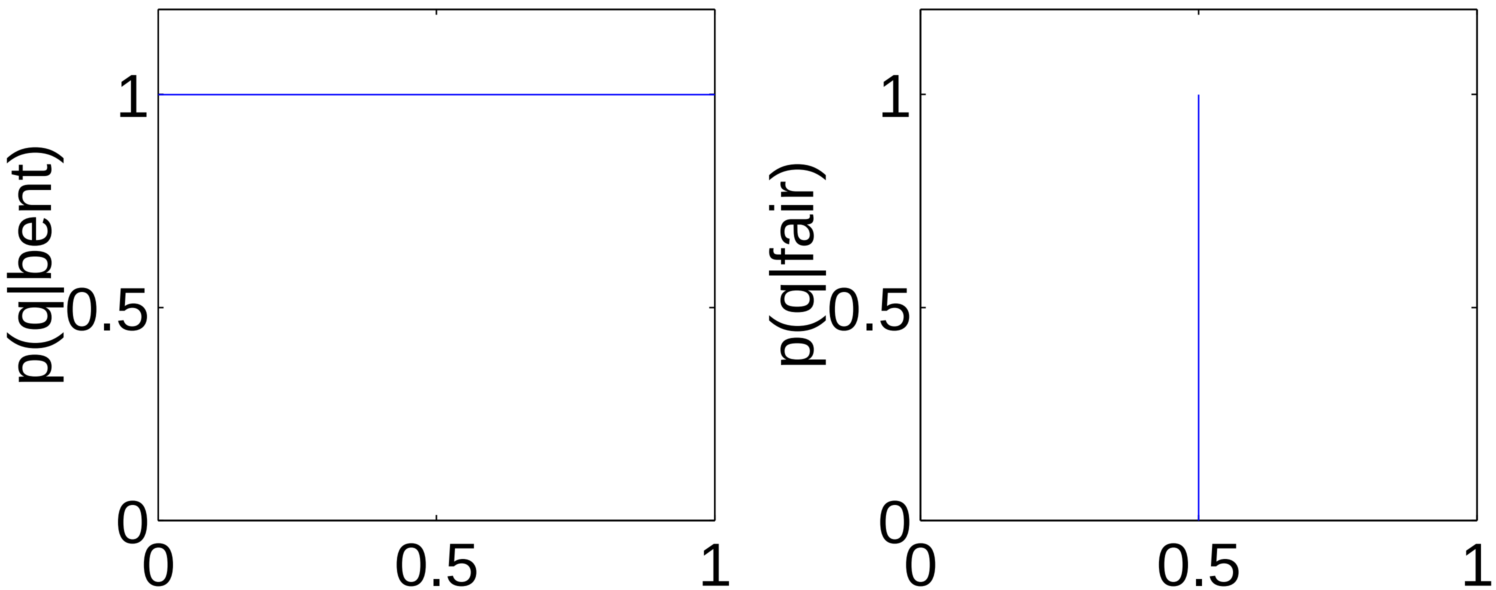
Bayesian Learning: Assumes a prior over the model parameters. Computes the posterior distribution of the parameters: $P(\theta|\mathcal{D})$.

Learning about a coin II

Consider two alternative models of a coin, "fair" and "bent". A priori, we may think that "fair" is more probable, eg:

$$p(\text{fair}) = 0.8, \qquad p(\text{bent}) = 0.2$$

For the bent coin, (a little unrealistically) all parameter values could be equally likely, where the fair coin has a fixed probability:



We make 10 tosses, and get: T H T H T T T T T T

Learning about a coin. . .

The evidence for the fair model is: $p(\mathcal{D}|\text{fair}) = (1/2)^{10} \simeq 0.001$
and for the bent model:

$$p(\mathcal{D}|\text{bent}) = \int dq\, p(\mathcal{D}|q, \text{bent}) p(q|\text{bent}) = \int dq\, q^2(1-q)^8 = \text{B}(3, 9) \simeq 0.002$$

The posterior for the models, by Bayes rule:

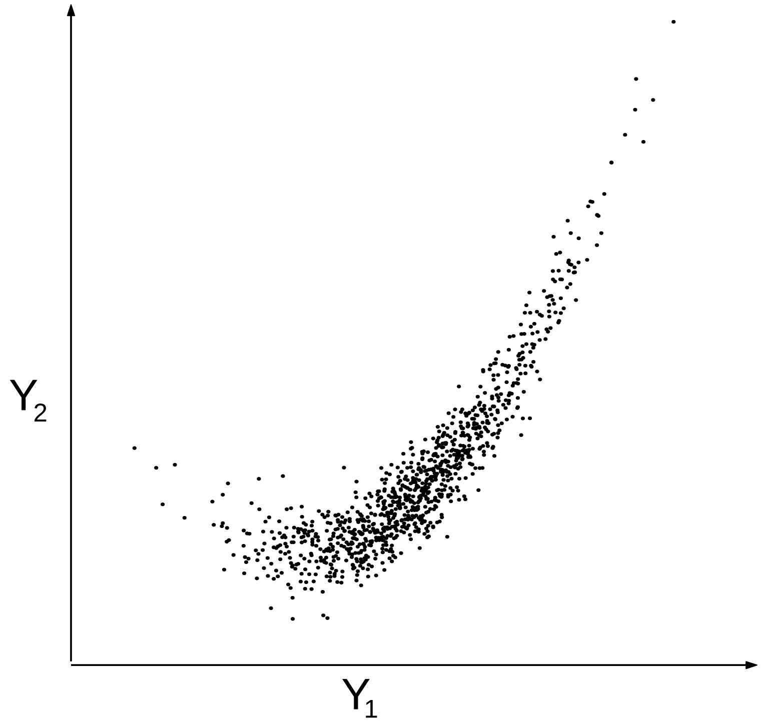
$$p(\text{fair}|\mathcal{D}) \propto 0.0008, \qquad p(\text{bent}|\mathcal{D}) \propto 0.0004,$$

ie, two thirds probability that the coin is fair.
How do we make predictions? By weighting the predictions from each model by their probability. Probability of Head at next toss is:

$$\frac{2}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{3}{12} = \frac{5}{12}.$$

[In contrast, the usual frequentist analysis might look something like this: Look at the observed data under the sampling distribution given the null hypothesis (fair) – the probability of the observed data, or something more extreme is $7/64$; this is larger than 0.1 so we do not reject the null hypothesis, and our prediction for future tosses is simply 0.5.]

Simple Statistical Modelling: modelling correlations



Assume:

- we have a data set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$
- each data point is a vector of D features: $\mathbf{y}_i = [y_{i1} \ldots y_{iD}]$
- the data points are i.i.d. (independent and identically distributed).

One of the simplest forms of unsupervised learning: model the **mean** of the data and the **correlations** between the D features in the data

We can use a multi-variate Gaussian model:

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

ML Estimation of a Gaussian

Data set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, likelihood: $p(Y|\mu, \Sigma) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mu, \Sigma)$

Maximize likelihood ⇔ maximize log likelihood

Goal: find μ and Σ that maximise log likelihood:

$$\begin{aligned}\mathcal{L} &= \log \prod_{n=1}^{N} p(\mathbf{y}_n|\mu, \Sigma) = \sum_n \log p(\mathbf{y}_n|\mu, \Sigma) \\ &= -\frac{N}{2} \log |2\pi\Sigma| - \frac{1}{2} \sum_n (\mathbf{y}_n - \mu)^\top \Sigma^{-1} (\mathbf{y}_n - \mu)\end{aligned} \tag{3}$$

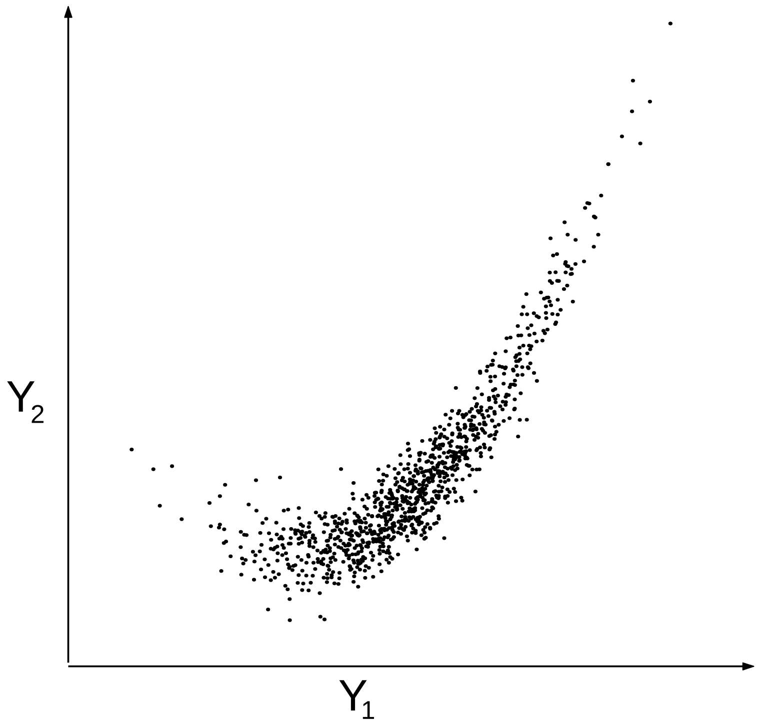
Note: equivalently, minimise $-\mathcal{L}$, which is *quadratic* in μ

Procedure: take derivatives and set to zero:

$$\frac{\partial \mathcal{L}}{\partial \mu} = 0 \quad \Rightarrow \quad \hat{\mu} = \frac{1}{N} \sum_n \mathbf{y}_n \quad \text{(sample mean)}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma} = 0 \quad \Rightarrow \quad \hat{\Sigma} = \frac{1}{N} \sum_n (\mathbf{y}_n - \hat{\mu})(\mathbf{y}_n - \hat{\mu})^\top \quad \text{(sample covariance)}$$

Note



modelling correlations

$\Updownarrow$

maximising likelihood of a Gaussian model

$\Updownarrow$

minimising a squared error cost function

$\Updownarrow$

minimizing data coding cost in bits (assuming Gaussian distributed)

Error functions, noise models, and likelihoods

- **Squared error:** $(y-\mu)^2$
  Gaussian noise assumption, y is real-valued

- **Absolute error:** $|y-\mu|$
  Exponential noise assumption, y real or positive

- **Binary cross entropy error:**
  $-y\log p-(1-y)\log(1-p)$
  Binomial noise assumption, y binary

- **Cross entropy error:** $\sum_i y_i \log p_i$
  Multinomial noise assumption, y is discrete (binary unit vector)

Three Limitations

- What about higher order statistical structure in the data? $\Rightarrow$ nonlinear and hierarchical models
- What happens if there are outliers? $\Rightarrow$ other noise models
- There are $D(D+1)/2$ parameters in the multi-variate Gaussian model. What if D is very large?

  $\Rightarrow$ dimensionality reduction

End Notes

For some matrix identities and matrix derivatives see:
`www.gatsby.ucl.ac.uk/∼roweis/notes/matrixid.pdf`

Also, see Tom Minka's notes on matrix algebra at CMU.
`http://lib.stat.cmu.edu/∼minka/papers/matrix.html`