Introduction

Machine Learning

Philosophical Background I

- induction
  - Involves logically unsound reasoning
  - Used to obtain empirical generalisations
  - Used to obtain predictions (need not involve a generalisation step)
- Deduction
  - Involves logically sound reasoning
  - Used to obtain specific details (theorems) from premises (axioms)
  - Used to identify necessary truths/inconsistencies/fallacies
- Bacon (1561–1626)
  - Begin by making a large number of observations. Laws will follow from this data, by a mechanical process of inference from the facts
- Hume (1711–1776)
  - Problem with this form of reasoning:

Philosophical Background II

1. All F's observed so far are G's
2. f is an F
3. Therefore f is a G

- This relies on a "principle of uniformity" which cannot be justified logically

- Peirce (1839–1914):

$$\textit{Induction} = \textit{Abduction} + \textit{Justification}$$

  - *Abduction*: the step of inventing/discovering *hypotheses*
  - *Justification*: the step of ordering/selecting amongst hypotheses, based on data

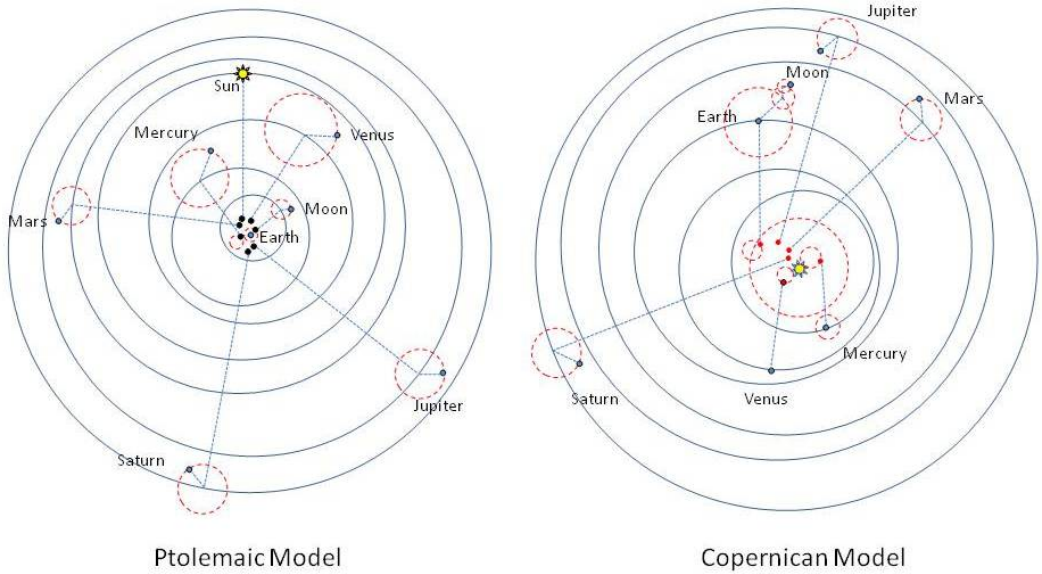
- Carnap (1891–1970)
  - Logic for justification based on probability theory
  - Purely syntax-based theory of confirmation of hypotheses ("inductive logic")

Philosophical Background III

- Popper (1902–1994)
  - Induction in the Baconian sense is not possible, in theory or in practice
  - Abduction in the Peirce-ian sense is essentially a creative process with no logical formulation
  - An inductive logic in the Carnap-ian sense is not possible, since and falsification based on deductive logic is the only option
- Bayesian Confirmation Theory
  - Based on a simple principle of conditioning: assuming the probability of new evidence e is non-zero, the probability of a proposition s after observing e is $P(s|e)$
  - For a hypothesis H and evidence E, the principle says the probability of H is $P(H|E)$, which is given by Bayes Rule as $P(E|H)P(H)/P(E)$
  - Evidence E *confirms* H if $P(H|E) > P(H)$.
  - The prior $P(H)$ is no longer purely syntactic (as required by Carnap)

Scientic Background I

- Background: Ptolemaic (100–170) and Copernican (1473–1543) models



1. Motion of all planets is circular
2. Velocity of all planets is uniform about a fixed centre
3. Orbits of all planets are in the same plane

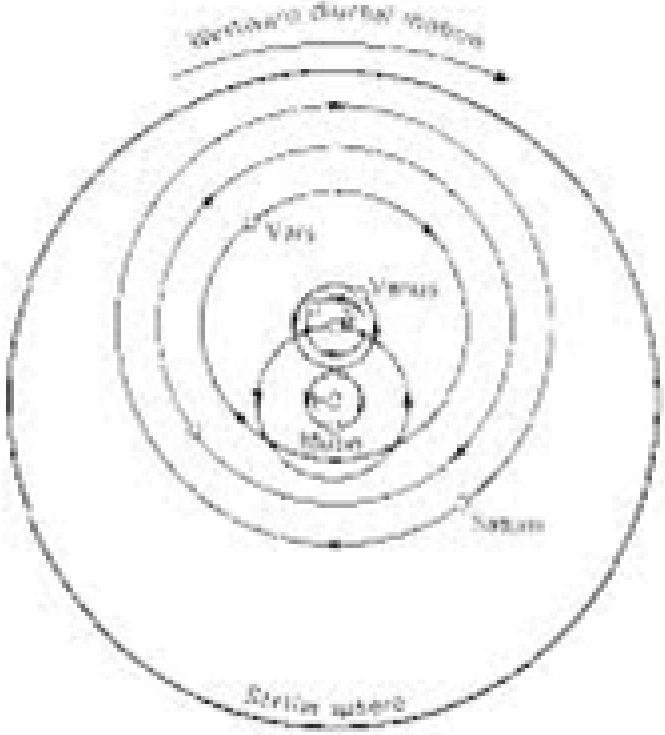
Scientic Background II

All astronomical models
were only to be taken as empirical theories, not necessarily true.

- Tycho Brahe (1546–1601)
  - Constructed an observatory with the most precise scientific equipment for the time
  - Approximate modern cost of $5 billion
  - Compiled the most accurate database of astronomical observations in human history upto the 16th century

Scientic Background III

- Constructed a model for the solar system in which: (a) the Earth was at rest; (b) the Sun and the Moon orbit the Earth; and (c) all other planets orbit the Sun



- Johannes Kepler (1571–1630)
  - Model for the orbit of Mars, using Tycho Brahe's data

Scientic Background IV

- First model: uniform circular motion. Did not match Brahe's data
- Second model: non-uniform circular motion. Matched Brahe's data almost perfectly (error of 4/1000).
- Third model: non-uniform egg-shaped orbit
- Fourth model: non-uniform oval shape to fit Tycho's data (but doesn't recognise it as the equation of an ellipse)
- Fifth model: non-uniform ellipse (!)

- Isaac Newton (1643–1727)
  - Concept of inertia and position as a mathematical function of time
  - Concept of infinitessimals with application to velocity and acceleration
  - Concept of force acting on bodies
  - Universal laws of motion using concept of force
  - Law of gravitation as a special case of laws of motion
  - Kepler's laws as a special case of law of gravitation

Scientic Background V

Which of these contributions can be made by Machine Learning today?

Computational Background I

Why Should Machines Learn?

Scientific Reason. Understanding of brains and behaviour; computational theory of Mind

Engineering Reason. Develop tools for Intelligent Modelling; (data analysis tools); Intelligent Assistants (tools for diagnostics and progrnosis); and Intelligent Infrastructure (tools for monitoring and control)

Computational Reason. Adaptive programs; autonomous agents

Business Reason. Improve business processes (increasing productivity, reducing wastage *etc.*

Philosophical Reason. Practical epistemology

Computational Background II

Can Machines Learn?

- What do we mean by *Can*?
- What do we mean by *Machine*?
- What do we mean by *Learn*?
- *Can*
  - What can machines learn?
    - Correlation: how wou;ld observing X change Y?
    - Intervention: how would changing X change Y?
    - Causation: Did X cause Y?
  - What can they learn now?
  - What can they ever learn?

Computational Background III

- *Machines*
  - Church-Turing Hypothesis: Any computable function can be computed by a (Universal) Turing Machine
  - Turing-completeness: A programming language is Turing-complete if we can write a program to simulate the computation of any Turing Machine
  - By "machine" we will really mean a sufficiently powerful computer program
  - So, by "machine" learning, we will mean a program that can learn

Computational Background IV

- *Learn*

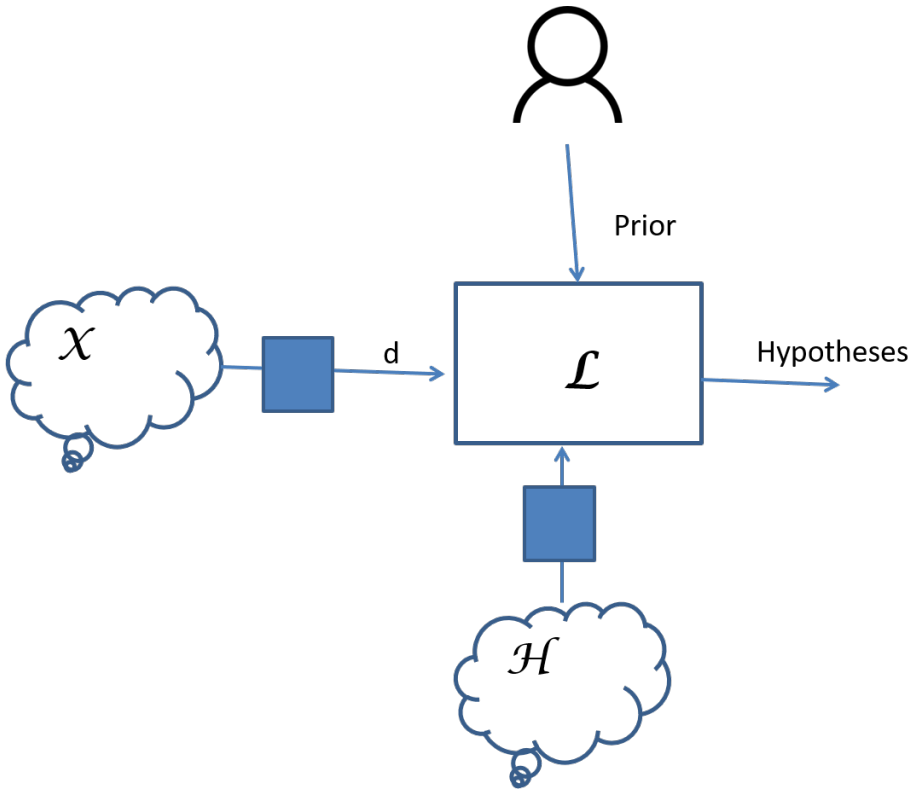
  Given: A data sample S drawn from set of instances $\mathcal{S}$; and a program $\mathcal{L}$

  Find: A program $h = \mathcal{L}(S)$ s.t. h computes answers to questions of correlation or intervention or causation

Computational Background V

So, Machine Learning is about programs that construct programs
Most of modern machine learning is about constructing programs that answer correlation-type questions ("how does observing X change Y?")

ML: Where Do We Start I



1. Data samples are drawn from $\mathcal{X}$. Each sample is called a "training set". For example, instances in the sample may be (real-valued) vectors of instances each with a label (say $+$ or $-$)

ML: Where Do We Start II

- This does not mean that we know the distribution that is generating the training data (if we did, then what is the point?)
- This does not mean we have LOTS of data. We may have very small amounts of training data
- This does not mean that the data are sampled uniformly from $\mathcal{X}$

2. We want a program that can answer questions, given data instances. Specifically, we will want the program to be a function that can map data instances to a value (a label, or a probability, or a number, or a group)
   - Functions will have a STRUCTURE and PARAMETERS. For example, the linear discriminant function $2x_1 + 3x_2 > 5$ has a linear structure, and parameters $2, 3$ and 5
   - In general, there may be many structures and associated parameters that can be candidate choices. This is called the MODEL SELECTION problem

ML: Where Do We Start III

3. A learning program $\mathcal{L}$ takes a data sample, and (perhaps) some prior information about the function to be constructed, and draws models (including structure and parameters) from a space of possible models $\mathcal{H}$
   - This does not tell us how to draw models from $\mathcal{H}$
   - This does not tell us how to select amongst models, given the data and prior preferences
4. $\mathcal{L}$ constructs H using two processes: inference (deduction) and estimation (induction)
   - Structure estimation is often done by a systematic or random search procedure; parameter estimation is usually done using sampling theory
   - The probability of the resulting program (or function) being "correct" is obtained using probabilistic inference

Statistical Learning I

- Given a sample $d = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$
  - Often $\mathcal{X} = \Re^d$
  - $\mathcal{Y} = \{0, 1\}$ (or $\{0, 1, 2, \ldots, k\}$ or $\mathcal{Y} = \Re$
  - For any x, we have y is a function of x. That is, $y = y(x)$. In general, x is taken as the value of a r.v. X (sometimes also written as X) and $y(x)$ is the value of the r.v. $Y = y(X)$ (sometimes $y(X)$ is called the *target* function)
  - We want to identify, or approximaely identify the target function, using one or more *hypotheses* or *models* from $\mathcal{H}$
- Since Y is a r.v. dependent on a r.v. X, there will be a conditional distribution $P(Y|X)$ (that is, $y(X)$ is a probabilistic function)

Statistical Learning II

- The "Bayes Classifier" for $\mathcal{Y} = \{0, 1\}$ and $\mathcal{X} = \Re^d$

$$\begin{aligned} h_B(X) &= 0 \quad \text{if } P(Y = 0|X) > P(Y = 1|X) \\ &= 1 \quad \text{otherwise} \end{aligned}$$

That is, the class with the maximum posterior probability is selected. From Bayes, we know:

$$P(Y|X) \propto f(X|Y)P(Y)$$

- If $P(Y)$ and $f(X|Y)$ are known, then no other hypothesis can do better than the Bayes Classifier (we will see in a moment what "better" means).

Statistical Learning III

- If we do not know $P(Y)$ and $f(X|Y)$, then we do not know $P(Y|X)$ and we cannot use the Bayes Classifier. But will attempt to estimate these quantities using and some set of hypotheses $\mathcal{H} = \{h|h : \mathcal{X} \mapsto \mathcal{Y}\}$.
- This is usually called *supervised* learning or *discriminative* learning.
- We will look at estimating $P(Y|X)$ in a moment. For the present, let us assume we know $P(Y|X)$ and consider the *conditional loss* or *conditional risk* of a hypothesis h $L(h|\mathrm{x})$ given x
- Suppose we are told (beforehand) the *cost* of the decision $Y = \hat{y}$ when the true value of $Y = y$ is is $\lambda(\hat{y}|y)$

Statistical Learning IV

- BUT: what do we mean by the "true" value of Y? We only have a distribution for this, given by $P(Y|X = \mathrm{x})$. So, we can only meaningfully obtain the (conditional) *expected* loss of h, given an instance x:

$$\mathrm{E}[L(h|\mathrm{x})] = \sum_{Y} \lambda(h(\mathrm{x})|y)P(Y = y|X = \mathrm{x})$$

- As an example, suppose we have the following loss function for a discrete r.v. Y:

$$\begin{aligned} \lambda(\hat{y}|y) &= 1 \quad \text{if} y = \hat{y} \\ &= 0 \quad \text{otherwise} \end{aligned}$$

This is called a $0-1$ loss function. For this function, suppose we have a hypothesis $h(\mathrm{x}) = \hat{y}$:

Statistical Learning V

$$\begin{aligned}\mathrm{E}[L(h|\mathrm{x})] &= \sum_{y \neq \hat{y}} P(Y = y|\mathrm{x}) \\ &= 1 - P(\hat{y}|\mathrm{x})\end{aligned}$$

- So, to minimises conditional expected loss we should use a hypothesis h .t. $P(\hat{y}|\mathrm{x})$ is a maximum. For $\mathcal{Y} = \{0, 1\}$, this is just what the Bayes Classifier h_B does
- So, all we need is $\mathcal{H} = \{h_B\}$ IF: we know $P(Y|\mathrm{x})$. BUT: we don't know $P(Y|\mathrm{x})$
  - We could try estimating them given sample data d
  - BUT THEN: the maximum posterior probability hypothesis may not have minimal error (WHY?)

Statistical Learning VI

- Since $P(Y|X) = P(X|Y)P(Y)$, if all $P(Y)$'s were the same, then maximising $P(Y|X)$ is the same as maximising $P(X|Y)$ (i.e. maximising the likeliho
- More generally, for any hypothesis h, we want to estimate the expected value of *unconditional loss* or simply expected *loss*, over values of (X, Y)

$$\mathrm{E}[L(h)] = \sum_{X,Y} \lambda(i|j) P(Y = j, h(X) = i)$$

This means we will need to estimate $P(X, Y)$.

Statistical Learning VII

- ASIDE: h_B is an example of a *discriminant function*

$$\begin{aligned} h(X) &= 0 \quad \text{if } g(X) > 0 \\ &= 1 \quad \text{otherwise} \end{aligned}$$

  For h_B, $g(X) = P(Y = 0|X) - P(Y = 1|X)$

- If $g(X)$ is a linear function of the form $w_0 + \sum i = 1^d w_i x_i$ then it is called a *linear discriminant* function.

Statistical Learning VIII

- So, how do we go about estimating $P(Y|\mathrm{x})$? We will assume we are also given some data d, and set of possible hypotheses $\mathcal{H}$. By conditional marginalisation:

$$\begin{aligned} P(Y|\mathrm{x}, d) &\propto \sum_{h \in \mathcal{H}} P(Y, h|\mathrm{x}, d) \\ &\propto \sum_{h \in \mathcal{H}} P(Y|\mathrm{x}, d, h) P(h|\mathrm{x}, d) \\ &\propto \sum_{h \in \mathcal{H}} P(Y|\mathrm{x}, d, h) P(h|d) \end{aligned}$$

- We will be concerned with hypotheses $h_i = (\pi_i, \theta_i)$, where π is a structure, and θ are parameters

Statistical Learning IX

- So, estimating $P(Y|\mathrm{x})$ requires us to estimate $P(h|d)$. By Bayes:

$$P(h|d) \propto P(d|h)P(h)$$

So, to estimate $P(h|d)$, we need to estimate the likelihood $P(d|h)$ and the prior $P(h)$. ‘ Any specific value $H = h$ is actually composed of 2 parts, the structure π and its parameters θ. These are in turn values of r.v's Π and Θ. So:

$$\begin{aligned} P(h|d) &\propto P(d|h)(h) \\ &\propto P(d|(\pi,\theta))P(\pi,\theta) \\ &\propto P(d|\pi,\theta)P(\pi)P(\theta|\pi) \end{aligned}$$

Statistical Learning X

- So, we need to estimate $P(d|\theta,\pi)$, $P(\pi)$ and $P(\theta|\pi)$. The first is a likelihood computation: let us assume that this can be done as a product of likelihoods of the individual instances given the hypothesis (recall: what is likelihood of a head, given a coin with $\Theta = \theta$)
- What about $P(\Pi = \pi)$? This is a prior over structures. We will have to estimate this, to capture something we know about the problem. For example, perhaps we know that all hypotheses are linear models and models with smaller number of variables are more likely. A prior p.m.f. for a linear model h with n_h variables is:

$$P(\Pi = \pi) \;=\; 2^{-n_h}$$

- What about $P(\theta|\pi)$. For the present, we will assume that his is a constant (for example, $\mathrm{E}[\Theta|\pi]$)

Statistical Learning XI

- ASIDE (AGAIN): If $P(h)$ is the same for all $h \in \mathcal{H}$, then $P(h|d) \propto P(d|h)$. That is the posterior probability of a hypothesis given data proportional to the likelihood of the data.

Statistical Learning XII

- ASIDE: What happens if elements of $\mathcal{H}$ is are (uncountably) infinite? (For example, $\mathcal{H}$ is all 1-variable linear models of the form $mX + c$, where m, c are numbers)
  - $P(h|d)$ will be $f(h|d)dh$
- MORAL: You only need the 4 combinations we looked at earlier (Discrete-Discrete; Discrete-Continuous; Continuous-Discrete; Conitinous-Continuous). For continuous r.v. use p.d.f's, and for discrete r.v. use p.m.f's. Replace σ with $\int$ when needed.

Problems I

- What if $\mathcal{H}$ is very large? Even if we could estimate $P(h|d)$, for any one h, the estimation of $P(Y|\mathrm{x})$ requires us to do this for all h's in $\mathcal{H}$:

$$P(Y|\mathrm{x}) \propto \sum_{h \in \mathcal{H}} P(Y|\mathrm{x}, d, h) P(h|d)$$

- APPROXIMATION:
  - Only sample hypotheses from $\mathcal{H}$. We will see examples of this when we look at ensemble methods.
  - Only pick one hypothesis from $\mathcal{H}$ (for example, the one with maximum $P(h|d)$, since this will have the highest weight). BUT: how do we get this?
    - Approximate $P(h|d)$ by a coinuous function (preferably convex) and optain an "optimal" hypothesis, using continuous oprimisation. We will see examples of this when we look at *regularisation* based model selection

Problems II

- Use combinaton of discrete combinatorial optimisation (branch-and-bound search) to search over structures and continuous optimisation to estimate parameters. We will see examples of this when we do structure-learning over graphical models.

- Even then, it may not be easy to compute $P(d|h)$ given an arbirtrary $P(h)$. SO: Assume some specific functional form for the likelihood $P(d|h)$ and prior $P(h)$, that allows easy calculation of $P(h|d)$

Summary

- Machine Learning has its roots in Philosophy, Empirical Science, Computation and Mathematics
- In this course, we will be looking at mainly at some of the main ideas underlying Statistical Machine Learning, and how those ideas apply to some broad categories of Machine Learning
- Bayes rule forms the basis for several inference tasks in Statistical Machine Learning
- 3 such inference tasks are: parameter estimation, structure estimation, and prediction
- Each of these can be formulated as the problem of inferring the posterior probability of a hypothesis, given data
- Hypotheses and data are both seen as values taken by random variables. The r.v. could be discrete or continuous, but the basic approach is the same (differences arise only in the use of p.m.f's or p.d.f's)