Learning from Data.
Introduction

David Barber, with modifications by Amos Storkey

Course lecturer: Amos Storkey
a.storkey@ed.ac.uk
Course page : http://www.anc.ed.ac.uk/∼amos/lfd/

Why Learning from Data?

Don't specify everything : Just learn it!

There are many possible motivations as to why one might want to "learn" from data. The typical scenario that is envisaged is the existence of a large database on which we perform computations. We wish to distinguish the kinds of computational procedures in this course from those that might be performed with a simple spreadsheet like application, which usually perform pre-determined computations such as calculating means and simple functions of the data. I don't want to get bogged down by what I mean by "learn", but I have in mind applications that would be hard to program in a traditional manner, such as the task of face recognition. Formally specifying why you recognise a collection of images as examples of John's face may be extremely difficult. Indeed, why bother?! You may as well just give examples of John's face, and let a machine "learn" – based on the statistics of the data – what it is that differentiates John's face from other faces in the database. That is not to say that all information can be learned solely on the basis of large databases – prior information about the domain is often crucial to the successful application of machine learning. However, the basic strategy is to try to make weak, yet consistent modelling assumptions, and let the data specify the rest. Some examples may enlighten this...

Knowledge Discovery

We may have various questions that we would like to ask about the database. For example, if we have a database of records of customer buying patterns :

coffee	1	0	0	1	0	0	0	· ·
tea	0	0	1	0	0	0	0	· ·
milk	1	0	1	1	0	1	1	· ·
beer	0	0	0	1	1	0	1	· ·
diapers	0	0	1	0	1	0	1	· ·
aspirin	0	1	0	0	1	0	1	· ·

where each column represents the buying patterns of a single customer (only 7 customer records shown). Here a 1 indicates that the customer bought that item (it does not record if multiple purchases of the item were made).

We may wish to find common patterns in the data, such as if someone buys milk they are also likely to have bought either tea or coffee. Whilst we may be able to spot such intuitive relationships by simply eye-balling the data, with many products and many customers, we need automated approaches.

Prediction

Consider the following banking data :

age	26	22	19	27	45	39	68	· ·
marital status	M	S	S	M	S	S	M	· ·
number children	0	0	1	2	0	3	0	· ·
salary	25000	18000	N/A	29000	50000	N/A	0	· ·
loan amount	100000	10000	1	15000	300000	1000	1	· ·
profession	teacher	nurse	student	1	IT	unemployed	retired	· ·
defaulted?	N	N	N	N	Y	N	N	· ·

Based on the data we may wish to construct a classifier that can predict whether or not a potential customer, based on their marital status, salary, loan amount requested and profession is likely or not to default.

A difficult, yet important problem is related to the prediction of (macro) molecular structure, given only the sequence of bases. Databases of sequence-structure relations (obtained by physical measurements of atomic co ordinates) exist, such as the fictitious RNA sequence-structure database below.

sequence	A	C	G	G	U	..
3D co-ordinates	(0.1 1.3 0.5)	(0.2 1.4 0.8)	(0.4 1.7 1.0)	(0.3 1.8 1.1)	(0.2 1.7 0.9)	..
sequence	C	A	G	U	G	..
3D co-ordinates	(-0.2 1.6 -0.5)	(0.1 1.2 -0.3)	(0.0 0.8 0.1)	(0.3 1.8 1.1)	(0.2 1.7 0.9)	..
sequence	..	..	..	..	..	..
3D co-ordinates	..	..	..	..	..	..

There may be trouble ahead ...!

The course is designed I think in a pragmatic way, ultimately with the aim to provide you with a set of tools, and also an appropriate philosophy for understanding and modelling data. However, the tremendously varied background amongst the participants means that making a course that will remain interesting, challenging, yet achievable for all the time for all of you is (to be honest) slightly beyond me. I anticipate some potential difficulties...

I've done (next to) no mathematics – the course is impossible!

The course is not designed for maths geniuses. However, in order to remain principled, rather than just getting you to learn recipes of algorithms, I will make considerable use of elementary mathematics. I strongly recommend that you study carefully the mathematics presented in this course, and spend some considerable effort in learning the mathematical skills. The mathematics needed is useful for many other courses later, where it will be assumed that you already understand the notation and basics.

I've done maths at Cambridge – why are you wasting my time with this!

Congratulations to those of you with strong maths backgrounds – it will come in handy. However, the course only uses maths as a tool and aims to provide intuitions and skills in data analysis. Whilst some of the initial material may be rather elementary, I hope that you'll find the course challenging overall.

How do you expect to learn anything man, my philosophy professor says that .....

Well, this is a course designed by someone who thinks of himself as a principled pragmatist who wants to solve real problems. I have precious little enthusiasm for philosophical debates, and their won't be much discussion about this in this course.

I want to do natural language processing, and your methods won't be any good to me

I hope that you'll find the methods I discuss very useful. I had several students last year who, whilst doing their final MSc projects in NLP made heavy use of some software developed in LFD. Furthermore, I hope that many students found the principles in the course useful.

I studied physics – you call this science?

Well, like physicists, we will also make models of the way we think the world works. If they are wrong we will use some principles to determine if our model is acceptable, given the available data.

I find the course way too hard – what should I do?

I would recommend that you stay with the course, whilst studying also another, backup course. You can always decide later if you wish to take an exam in a different course. I think that LFD is vitally important for all the other courses, and you should try to stay with it, even just to observe.

I did computer science – Algorithms are the key!

You may make up the majority of the class. The traditional viewpoint is that computer science is driven by algorithms. For example, find the minimum weighted path between vertices on a directed graph. This course, however, is driven by *models*. Essentially, this course is more about science, where models are made and evaluated in light of the data. The computational application of a model will typically involve some algorithm. However, the algorithms are simply consequences of implementing the model.

The LDF1 Approach

Make a (mathematical) model of the data. Use the available training data to adjust free parameters of the model/tune the model. Everything else (algorithms, programming languages etc...) is subservient to this approach.

Syllabus

The following core topics will be included :

- Introduction and Mathematical Preliminaries
- Supervised Learning
  - perceptrons, neural networks, feature selection, nearest neighbour methods, decision trees

- Generalisation
  - assessment of performance, experimental methodology and design, model selection
- Unsupervised learning
  - clustering, PCA

Intellectual skills and development

This course involves some implementation work using MATLAB, and contains also significant theoretical work involving areas of mathematics other than logic. The course details the specific applications of various basic mathematical techniques to areas of pattern recognition and processing, the aim being that at the end of the course, participants should have a good understanding and ability to use these techniques in practice. The course aims to foster a systematic approach to experiments.

Activities and Assessment

The course comprises 20 lectures and 8 tutorials (one per week). There will be two assessed practicals. Together, these will carry 20% of the course marks. The remaining 80% will be carried by the exam.

Context

A reasonable level of familiarity with computational, logical, geometric and set-theoretic concepts is assumed. Knowledge of vectors and matrices, together with a basic grasp of probability and partial differentiation, will be very important. The course involves a small amount of programming work in MATLAB.

The module on Probabilistic Modelling and Reasoning (PMR) (available to MSc students) is a parallel module, dealing with issues of learning and inference in probabilistic systems. The PMR module will require a somewhat higher level of mathematics than LFD. The module Reinforcement Learning (RL) concerns the problem of learning how to act in an environment where one does not obtain immediate rewards.

MSc students can take the module on Neural Computation (NC) which provides, in the main, a computational neuroscience perspective on neural computation.

References

The lecture notes provided are intended to be largely self-contained. There is no single book available which covers all of the material. You don't need to buy any of the following books. However, you will find in them material related to this course which may provide an alternative explanation to mine.

The following book is written by a computer scientist. I don't really like it since it is an algorithms driven perspective. However, those of you that studied computer science might find a more kindred spirit here.

- *Machine Learning* T. Mitchell, McGraw-Hill, 1997

The following is a nice book which has some good chapters on basic data analysis. I don't believe strongly in "neural nets" but the book, despite the title, contains a great deal of non-neural nets stuff.

- *Neural Networks for Pattern Recognition.* C. M. Bishop. Oxford University Press, Oxford. 1995.