M5MS10
Machine Learning

Spring 2018

Dr Ben Calderhead
b.calderhead@imperial.ac.uk

M5MS10
Machine Learning

Spring 2018

Dr Ben Calderhead
b.calderhead@imperial.ac.uk

Dr Ben Calderhead
Department of Mathematics
Statistics Section

Huxley Building Room 523
b.calderhead@imperial.ac.uk

Structure

Lectures:

Mondays 12pm - 1pm (Huxley 130)

Thursday 9am - 10am (Huxley 140)

Computer Lab:

Thursday 10am - 11am (Huxley 215)

Resources on Imperial BB:

Lecture slides available online

Suggested reading for each topic online

Computational tutorials in R/Matlab

Course Objectives

1) to provide an overview of machine learning

2) to offer insight into the theory behind a wide variety of commonly used computational approaches and algorithms

3) to study the application of supervised and unsupervised learning techniques and apply them

4) to illustrate selected methods and concepts in real applications

Approach

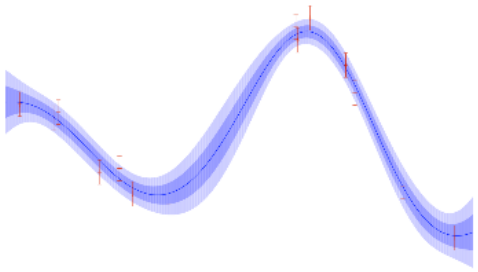
- Students are expected to code up their own versions of the algorithms we cover from scratch. Blindly using off-the-shelf Machine Learning Packages will be graded with 0 marks.

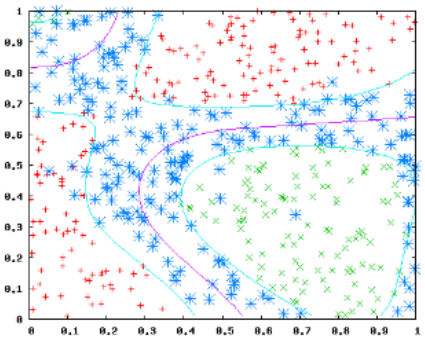
- Focus on basic principles underlying many Machine Learning methods

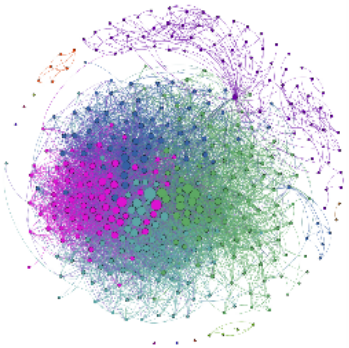
- Students should be capable of making intelligent decisions about which methods would be appropriate for specific problems

Students strongly advised to attend all lectures & laboratories and to keep up with material as delivered - both lecture material and recommended reading

Syllabus







- Regression and linear models
- Evaluating learning methods
- Discriminative classification approaches
- Generative classification approaches
- Dimensionality reduction
- Clustering
- Kernel methods
- Gaussian process regression
- Gaussian process classification

Machine Learning with R or Matlab

A very important part of the course is the implementation and application of machine learning methods to a variety of data sets, which *greatly* aids understanding of the uses and limitations of the methods.

High level languages such as R, Matlab or Python offer many advantages:

- rapid implementation of algorithms
- more flexibility and customisable methods
- can be called from other languages and embedded into larger programs

In this course we will use R and Matlab (although you might like to try coding the same algorithm in different languages...)

Further Reading

"Bayesian Reasoning and Machine Learning"
David Barber
Cambridge University Press

"The Elements of Statistical Learning"
Hastie, Tibshirani and Friedman
Springer

"Machine Learning: A Probabilistic Perspective"
Kevin Murphy
MIT Press

In addition, recommended reading from free available sources on the internet will be posted on BB.

Resources on the Internet

The Comprehensive R Archive Network

http://cran.r-project.org

Machine Learning Task View

http://cran.r-project.org/web/views/MachineLearning.html

Multivariate Statistics Task View

http://cran.r-project.org/web/views/Multivariate.html

Cluster Analysis Task View

http://cran.r-project.org/web/views/Cluster.html

Assessment (coursework only)

First coursework

- Hand-out date: Thursday 25th January
- Hand-in date: Thursday 8th February
- 20% of the total mark

Second coursework

- Hand-out date: Tuesday 15th February
- Hand-in date: Tuesday 15th March
- 80% of the total mark

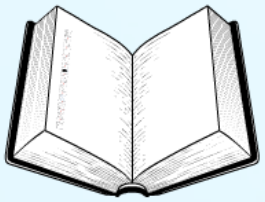
Enjoy...!

IS10
Learning

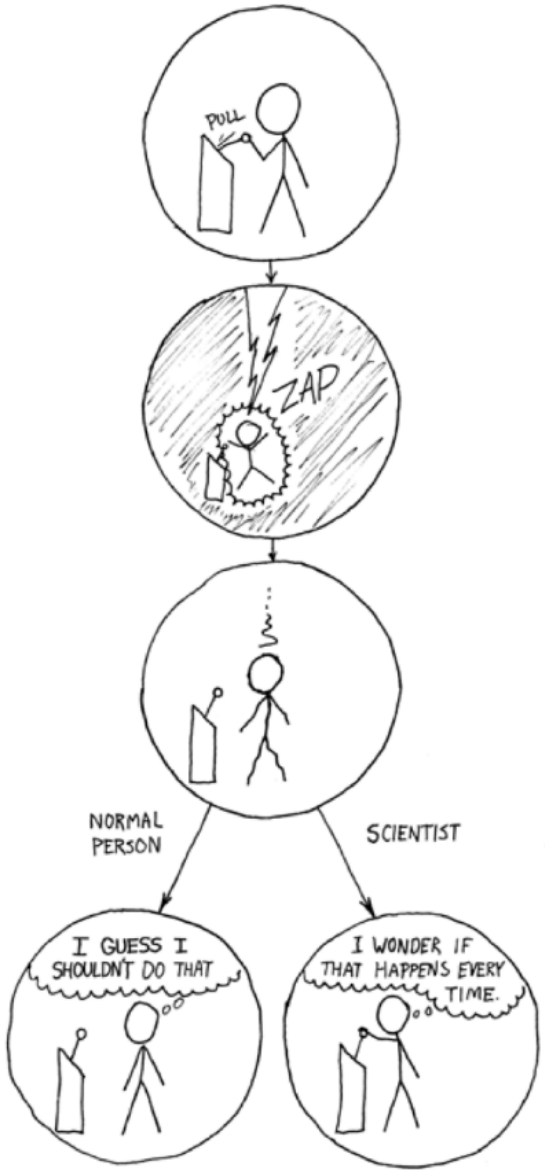
2018

lderhead
mperial.ac.uk

Introduction



What is Machine Learning?



Learning is an essential human property:

The acquisition of knowledge of skills through experience, practice, study, or by being taught.

As with humans, machine learning algorithms "learn" from examples (i.e. training data) how to carry out specialised tasks.

Learning is not learning by heart - the difficulty lies in generalising the behaviour to novel situations.

Applications of Learning

Learning to predict real-valued measurements

- House or stock prices given some economic variables
- Medical diagnosis given some clinical/genetic variables

Learning to assign objects to one of many predefined classes

- Classify webpages given their text and image content
- Recognise faces in photos and videos
- Speech and handwriting recognition

Learning to discover novel or unusual patterns

- Discover previously unknown tumour subtypes
- Detecting credit card fraud

Who does Machine Learning?

Machine learning overlaps heavily with statistics, since both fields study the analysis of data, but unlike statistics, machine learning is also concerned with the algorithmic complexity of computational implementations.

Researchers & practitioners from diverse backgrounds contributing to development of the discipline:

Computing Science - Artificial Intelligence, Neural Computing, Logic Programming, Algorithmics

Statistics - Multivariate Statistics, Bayesian Statistics, Statistical Pattern Recognition

Physics - Monte Carlo methods, Mean-Field approximations

Engineering - Control Theory and Adaptive methods

Psychology - Cognitive science and theories of learning

Some further motivation...

SENIOR DATA SCIENTIST – Machine Learning Startup-London (Ruby,Python)

The company is a growing machine learning startup who are rapidly expanding their engineering and data science team. You will be working in the heart of London and play a key role working closely with clients to develop their work in big data. As the... machine learning jobs... From £50,000 to £80,000 per annum
from: cwjobs.co.uk - 3 days ago
Similar: Learning Development Manager, Machine Mechanic

Machine Learning Analyst - Predictive Modelling - MATLAB

Good Degree in Computer Science, Mathematics, Operational Research or other highly quantitative field (2:1 or above). A relevant post-graduate qualification would be highly advantageous, ideally with a focus on Machine Learning. You will be an exceptional... Up to £50,000 per annum plus excellent Benefits
from: cwjobs.co.uk - 9 days ago

Applied Machine Learning Microsoft

Join the excitement of Machine Learning in the Cloud at Microsoft! We are a fast paced data science team in the Microsoft Cloud + Enterprise organization building machine learning powered intelligent web services and end to end solutions for scenarios in diverse enterprise and consumer verticals.

We are looking for applied scientists who are passionate about applying machine learning and data mining techniques to a variety of exciting applications for enterprises and consumers. You will apply a range of machine learning techniques including predictive modeling, text and image mining, recommendations, clustering, anomaly detection, forecasting methods, deep learning and other advanced statistical techniques.

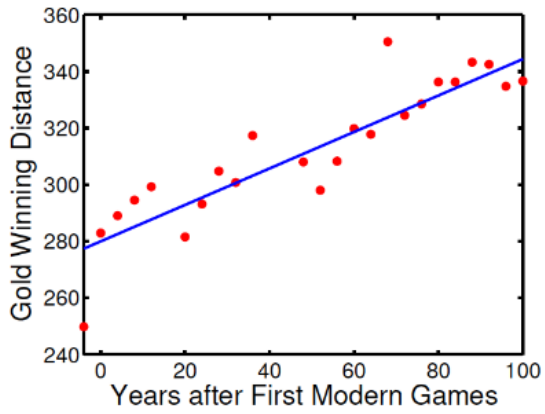
Jobs in large firms: Google amazon

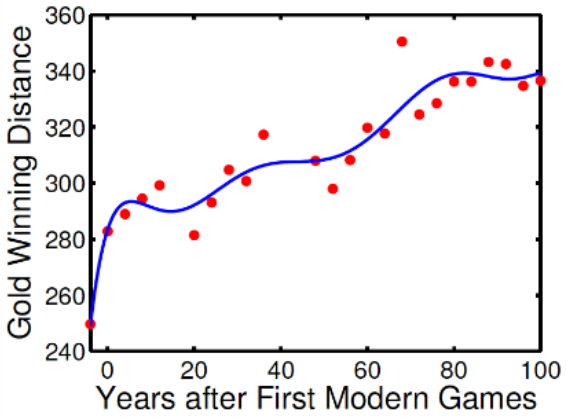
Search for Machine Learning jobs in start-ups at http://angel.co

Why is Learning Difficult?

Given a *finite* amount of data we must often derive a relation to an *infinite* domain.

There are an infinite number of such functional relationships, so how should we decide which is most appropriate for the data and chosen variables?





Occam's Principle

William of Occam (a wise monk from the 14th century) once said,

"Pluralitas non est ponenda sine necessitate"

"Plurality is not to be posited without necessity"

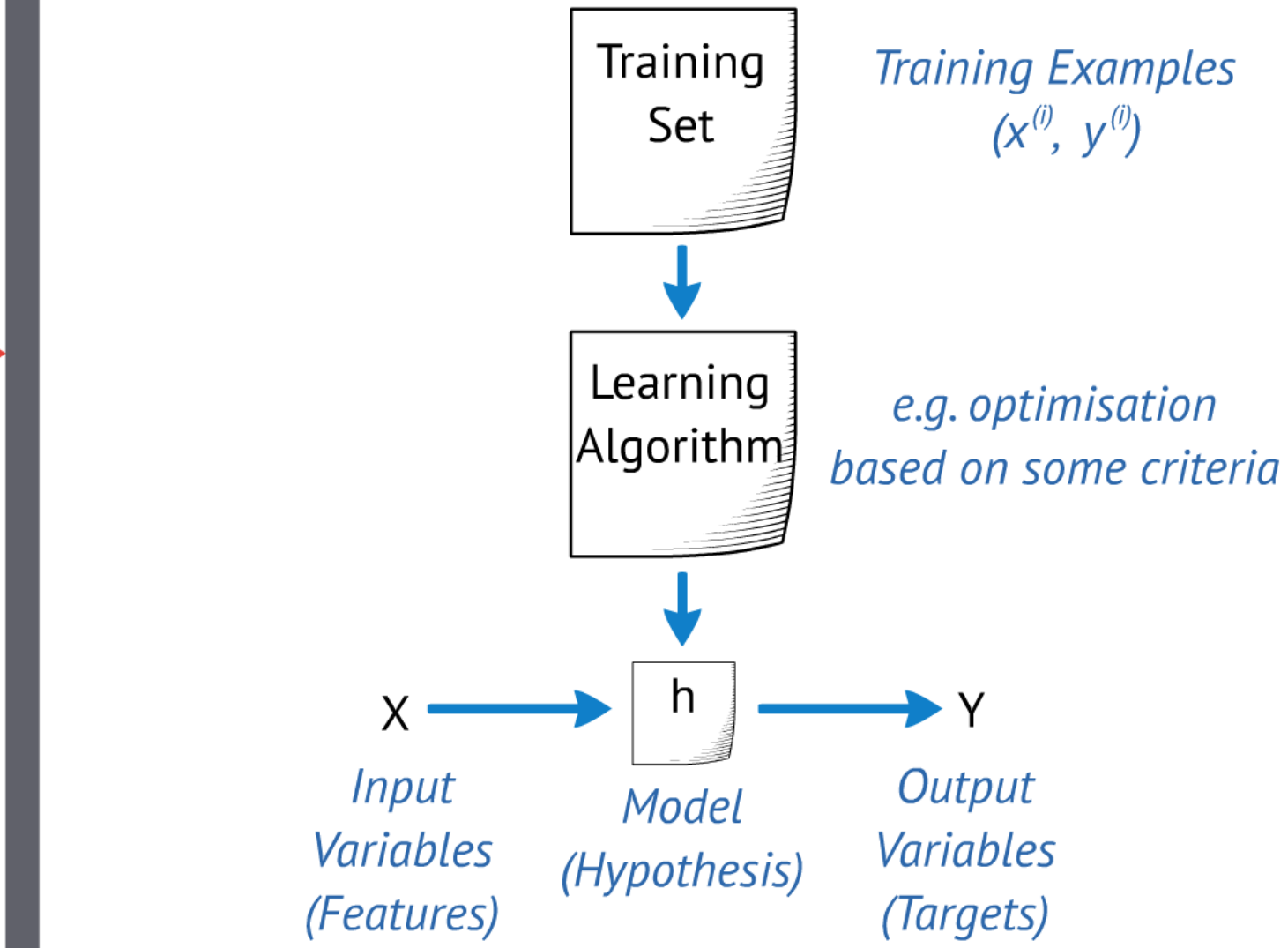
"Don't make things more complicated than necessary!"

When there are many solutions available for a given problem, we should select the simplest one.

We can define what we mean by "simple" in many ways:

...smoothness of the solutions, low dimensionality of the solution, sparsity of the solution...

The General Approach for Supervised Learning



Different Types of Learning

Supervised Learning:

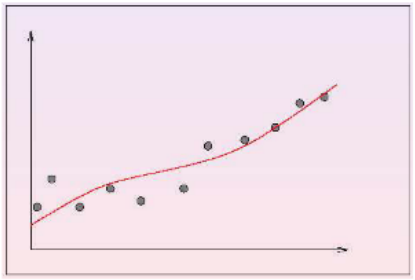
The data contains *labels*. In other words, we observe both the input variables and the corresponding output variables for each datapoint.

Unsupervised Learning:

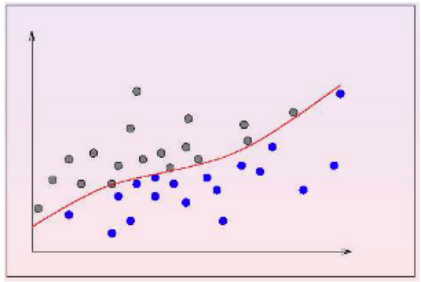
We observe only datapoints consisting of input variables and the aim is to find structure in it.

Types of Problems

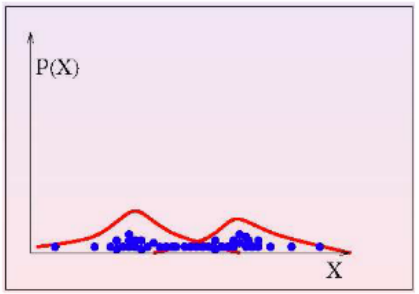
Regression - supervised learning in which the labels are continuous values.



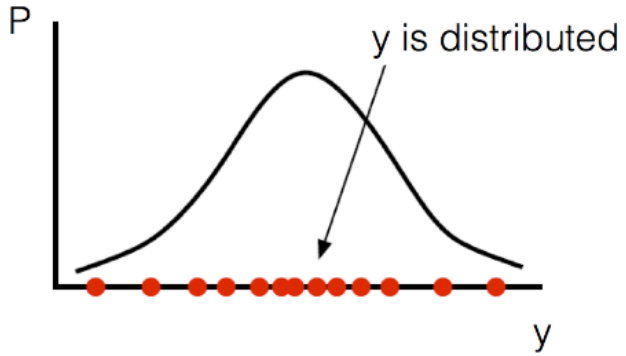
Classification - supervised learning in which the labels are discrete sets.



Density estimation - we are interesting in discovering the underlying distribution from which the data might have been generated.

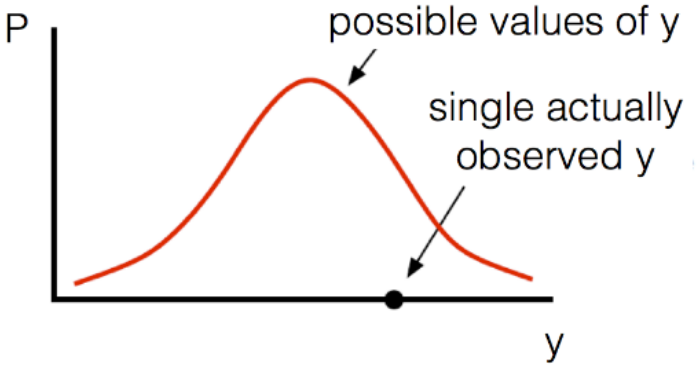


Bayesian or Frequentist?

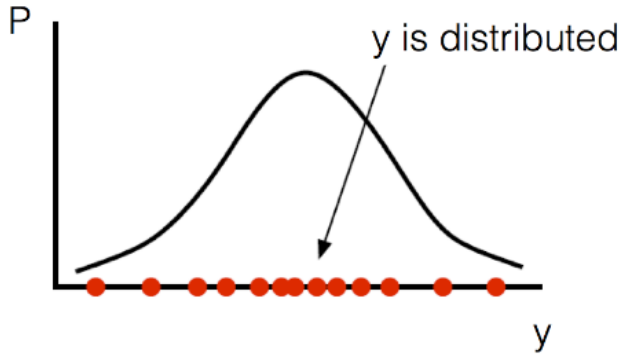


In a frequentist setting, probabilities are interpreted as the **limiting proportions or frequencies** of an collection of data points.

In a Bayesian setting, we interpret probability as a measure of **belief of the possible values** of the data, even if it's just a single data point.

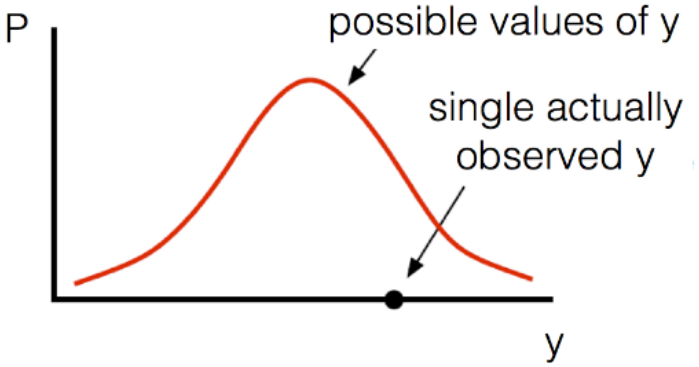


Bayesian or Frequentist?



In a frequentist setting, probabilities are interpreted as the **limiting proportions or frequencies** of an collection of data points.

In a Bayesian setting, we interpret probability as a measure of **belief of the possible values** of the data, even if it's just a single data point.



Learning Machine Learning

Machine learning is a huge and incredibly useful topic.

Fortunately, there are basic underlying issues and principles that arise again and again, many of them statistical!

We will focus on simpler models that illustrate the main principles, that are also pertinent for more complex models.

In each area I will point to further reading that develops the ideas we have covered

- increasing complexity of models
- improving the scalability of the approach
- improving robustness to deal with changing data

Please read: "**A Few Useful Things About Machine Learning**"

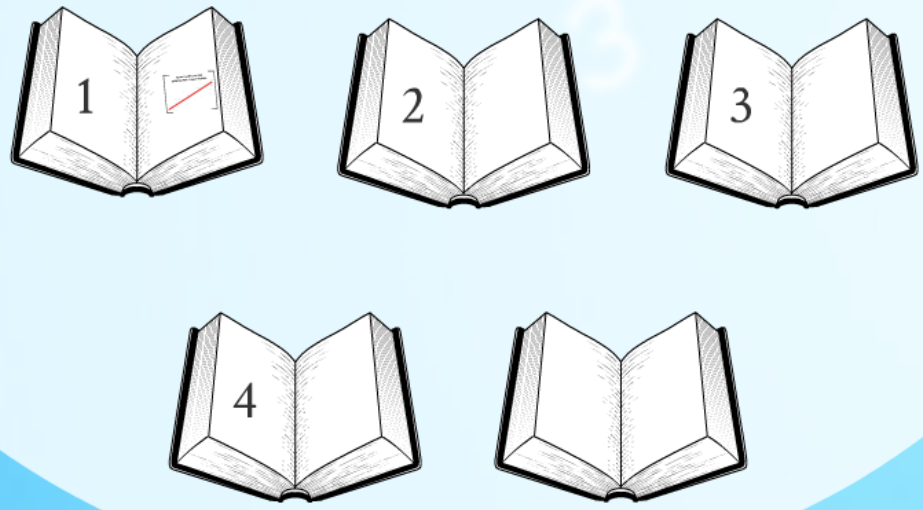
Lots to cover - so let's begin!

IS10
Learning

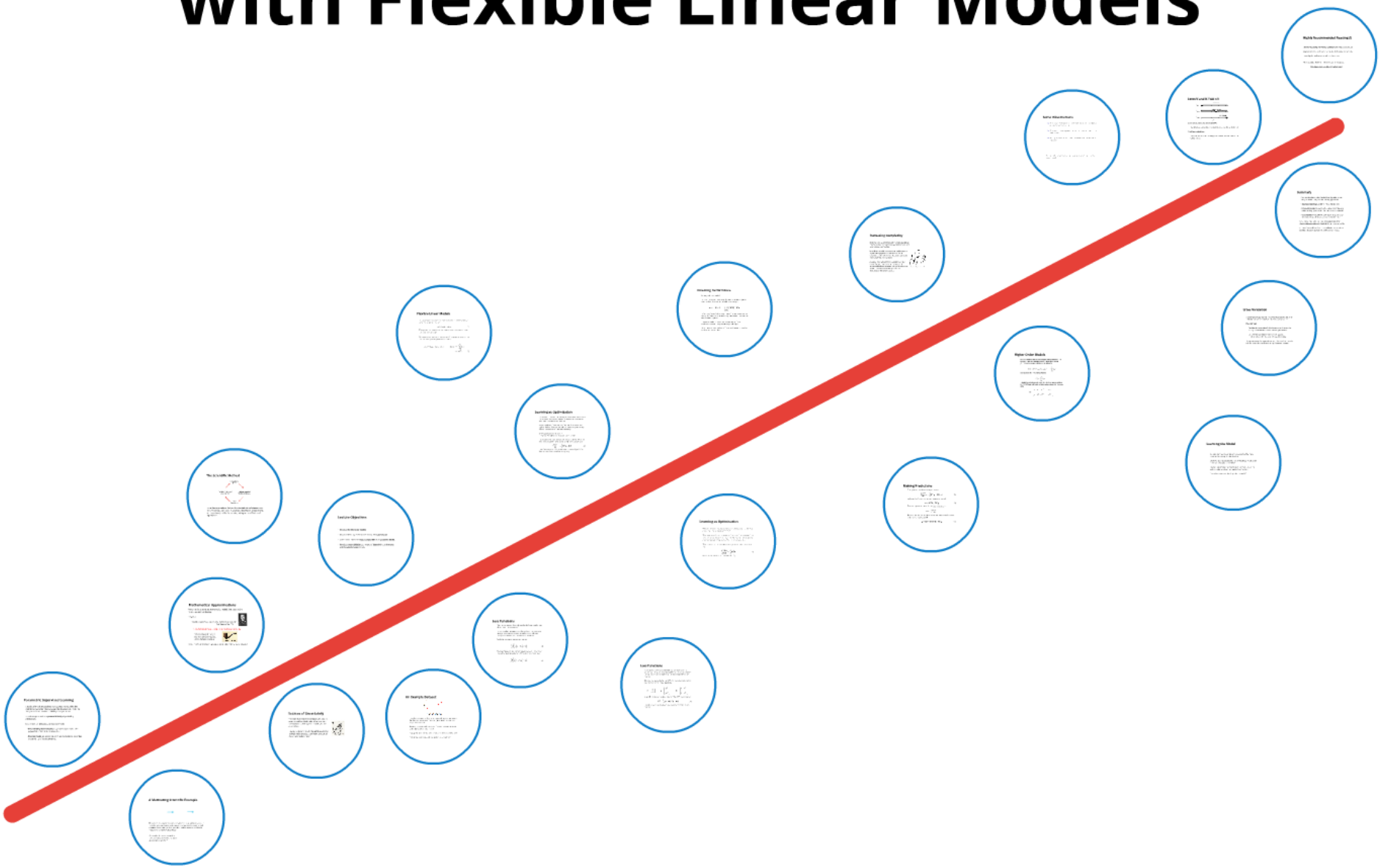
2018

lderhead
mperial.ac.uk

Parametric Methods



Supervised Learning with Flexible Linear Models



Parametric Supervised Learning

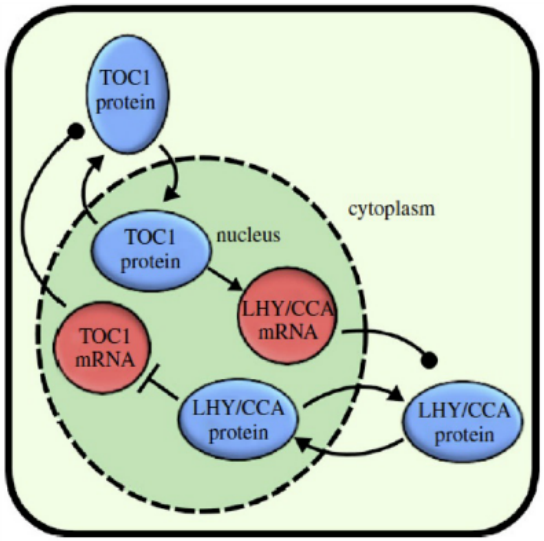
A statistical model encapsulates our assumptions regarding the underlying mechanism that generates the observed data including the systematic and random variability we expect to see.

It is often represented as a **parametric family** of probability distributions.

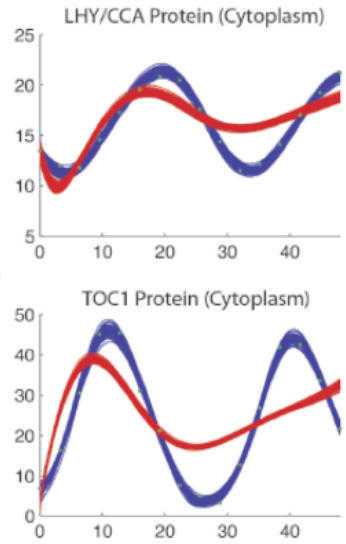
We can consider two parts to a statistical model

- **the underlying "true" behaviour**, e.g. a mean represented by a polynomial, differential equation, etc...
- **the error model,** e.g. a distribution or loss function that describes the variability of the observations.

A Motivating Scientific Example

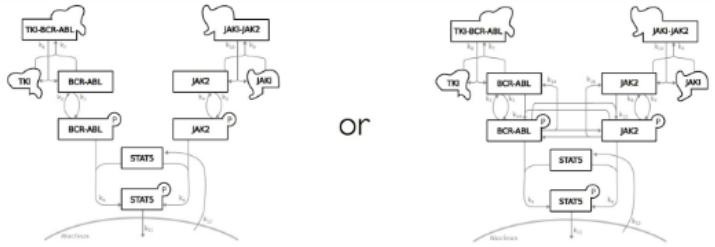


$$\frac{d[LHY]_m}{dt} = \frac{n_1[TOC1]_n^a}{g_1^a + [TOC1]_n^a} - \frac{m_1[LHY]_m}{k_1 + [LHY]_m},$$
$$\frac{d[LHY]_c}{dt} = p_1[LHY]_m - r_1[LHY]_c + r_2[LHY]_n - \frac{m_2[LHY]_c}{k_2 + [LHY]_c},$$
$$\frac{d[LHY]_n}{dt} = r_1[LHY]_c - r_2[LHY]_n - \frac{m_3[LHY]_n}{k_3 + [LHY]_n},$$
$$\frac{d[TOC1]_m}{dt} = \frac{n_2\, g_2^b}{g_2^b + [LHY]_n^b} - \frac{m_4[TOC1]_m}{k_4 + [TOC1]_m},$$
$$\frac{d[TOC1]_c}{dt} = p_2[TOC1]_m - r_3[TOC1]_c + r_4[TOC1]_n - \frac{m_5[TOC1]_c}{k_5 + [TOC1]_c}$$
and $$\frac{d[TOC1]_n}{dt} = r_3[TOC1]_c - r_4[TOC1]_n - \frac{m_6[TOC1]_n}{k_6 + [TOC1]_n}.$$



We can use this approach to reverse engineer cell regulatory networks. In particular we can use mechanistic models to represent the hypothesised structure of the system, and compare the model output to the biological measurements that have been made.

Given multiple parametric models, how do we decide which is the most appropriate description?




Ceci n'est pas une pipe.

Mathematical Approximations

We can strive to construct mathematical models that describe the natural systems we observe.

However

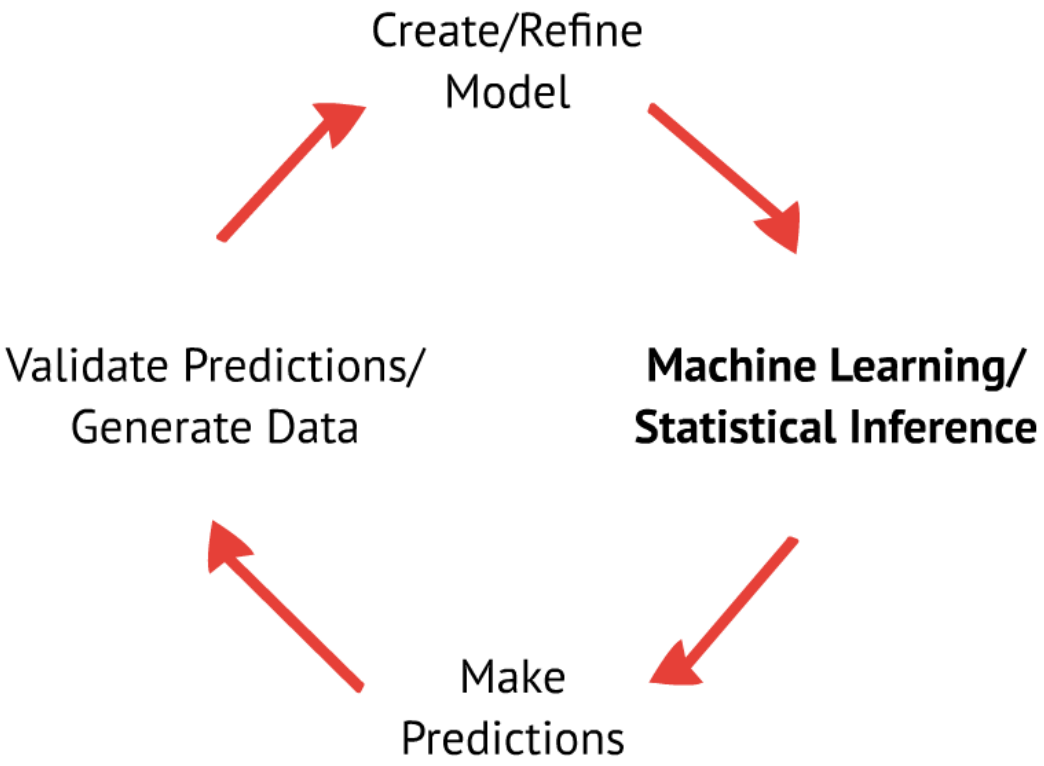
> “essentially, all models are wrong, but some are useful”
>
> Prof. George Box FRS

A mathematical model is only an approximation of reality.

"The Treachery of Images"
("La trahison des images"),
1928–29, Rene Magritte

One aim of machine learning is to automatically find the *useful* models!

The Scientific Method

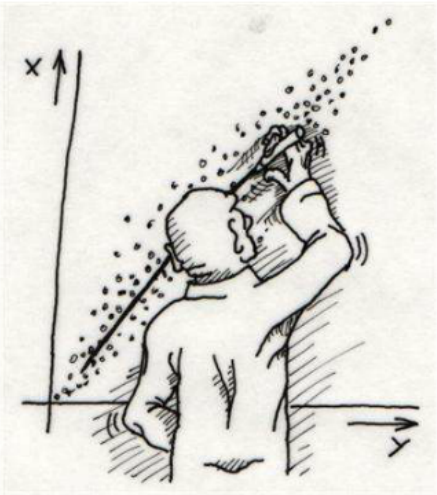


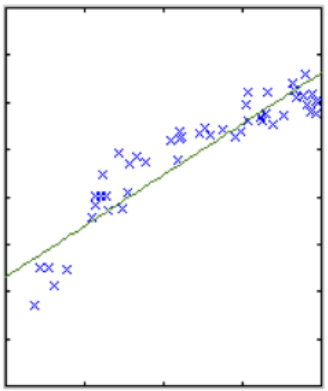
"a method or procedure that has characterised natural science since the 17th century, consisting in systematic observation, measurement, and experiment, and the formulation, testing, and modification of hypotheses."

Sources of Uncertainty

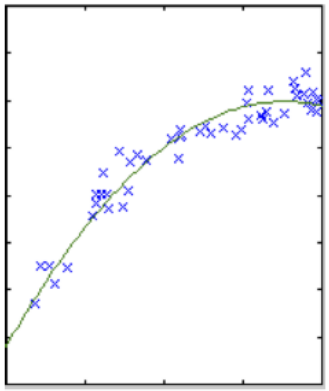
The main challenge with trying to construct an accurate mathematical model of some natural phenomenon is dealing with variability in the observations.

How do we determine whether what we observe is simply noise or actually an important part of the underlying dynamics?

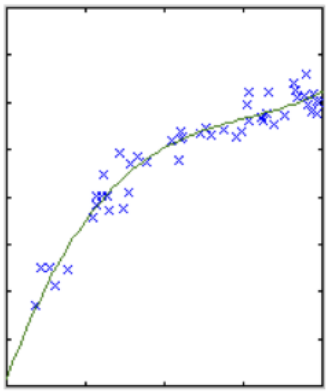




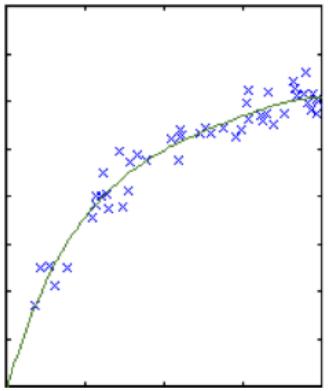
1st order polynomial



2nd order polynomial



3rd order polynomial

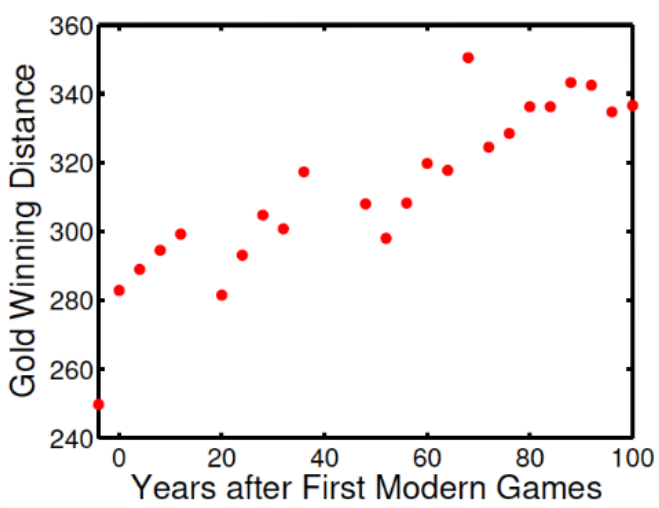


4th order polynomial

Lecture Objectives

- Introduce flexible **linear models**.
- Introduce learning based on the concept of a **loss function**.
- Consider the effect that **model complexity** has on **predictive ability**.
- Introduce **cross validation** as a means of determining performance and choosing between models.

An Example Dataset



In order to analyse this data using supervised learning, we have to decide upon the choice of functions (hypotheses) that we think might be appropriate.

We want to make predictions about future or unknown responses given new values of the attributes.

How do we learn the relationship from a finite set of observations?

How do we assess how good the model is as a predictor?

Flexible Linear Models

Let us consider first a simple linear relationship between y and x, using the following function,

$$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 \tag{1}$$

The θ_i's are the parameters that parameterise the space of linear functions from $\mathcal{X}$ to $\mathcal{Y}$.

For convenience, we can introduce $x_0 \equiv 1$ to allow us to write this function more generally as a vector product,

$$\begin{aligned} h(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_D x_D &= \sum_{i=0}^{D} \theta_i x_i \\ &= \boldsymbol{\theta}^T x \end{aligned} \tag{2}$$

Loss Functions

Now that we have our data and have decided upon a model, how do we "learn" the parameters?

Let us introduce the concept of a *loss function*. This function will measure the mismatch between the model output and each input/output example pair, for each set of parameters.

We define the sample average loss function,

$$\frac{1}{N}\sum_{i=1}^{N} J\left[y^{(i)}, h_{\boldsymbol{\theta}}\left(x^{(i)}\right)\right] \tag{3}$$

We may choose our loss function however we want. In particular, the sample mean squared error (MSE) loss is particularly useful,

$$\frac{1}{N}\sum_{i=1}^{N} \left(y^{(i)} - h_{\boldsymbol{\theta}}\left(x^{(i)}\right)\right)^2 \tag{4}$$

Loss Functions

Other types of losses can be defined that are more robust to outliers (e.g. using an L1 norm), however for illustration purposes the main points can be made using the mean squared error loss function.

We note that we can define the MSE loss in an even more succinct matrix form. For $D = 2$ we may define,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & x^{(1)} \\ \vdots & \vdots \\ 1 & x^{(N)} \end{bmatrix},$$

where $\mathbf{X}$ is known as the design matrix. The MSE then follows as,

$$\text{MSE} = \frac{1}{N}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \tag{5}$$

In order to learn the parameters, we therefore *minimise* this loss function.

Learning as Optimisation

In this case, it turns out that we can obtain an analytic solution to this minimisation problem, however for many other problems we must resort to numerical optimisation.

There already exist freely available functions for minimisation - gradient descent, Newton's method etc. These are generally very efficient, so make use of them when necessary!

A very good book on the topic is:
"Practical Methods of Optimization" by R. Fletcher

For an analytic solution, we can take the partial derivatives of the MSE function and set them to zero to find the stationary point.

$$\frac{\partial \text{MSE}}{\partial \boldsymbol{\theta}} = -\frac{2}{N}\mathbf{x}^T(\mathbf{y} - \mathbf{x}\boldsymbol{\theta}) \tag{6}$$

(The freely available "Matrix Cookbook" is *extremely* useful for manipulation of matrix and vector algebra.)

Learning as Optimisation

We can therefore find a stationary point analytically, but we must check that this is indeed a minimum!

This is satisfied for multi-parameter functions if the *Hessian* of the function is positive definite, i.e. if the matrix of all partial second order derivatives, H, satisfies $a^T Ha > 0$ for all vectors a.

This turns out to be the case, since high school calculus tells us that

$$\frac{\partial^2 \text{MSE}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}} = \frac{2}{N} \mathbf{x}^T \mathbf{x} \tag{7}$$

which is positive definite (provided $N > D$).

Making Predictions

Finally, we set the derivative equal to zero,

$$\frac{\partial \text{MSE}}{\partial \boldsymbol{\theta}} = -\frac{2}{N}\mathbf{x}^T(\mathbf{y} - \mathbf{x}\boldsymbol{\theta}) = 0 \tag{8}$$

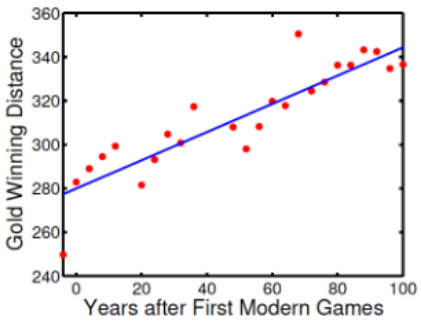
and solve for θ to obtain the least squares estimate $\hat{\theta}$

$$\hat{\boldsymbol{\theta}} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{9}$$

The least squares estimate for the long jump data is:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} 276.78 \\ 0.748 \end{bmatrix}$$

We can then use the parameter values we have learned to make predictions at new values $\bar{\mathbf{x}}$,



$$\hat{\mathbf{y}} = \bar{\mathbf{x}}\hat{\boldsymbol{\theta}} = \bar{\mathbf{x}}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{10}$$

Assessing Performance

So how good is our model?

Let's test this against the winning distance in the 2012 Olympics, since this data point was not included in our analysis.

$$\begin{aligned} \hat{\mathbf{y}}_{2012} = \bar{\mathbf{x}}_{2012}\hat{\boldsymbol{\theta}} &= [1, 112](\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \\ &= 360.5 \end{aligned}$$

In fact, Greg Rutherford's winning jump of just 327 inches was the shortest distance to win the men's long jump competition since the 1972 Summer Olympics!

It seems our model is too optimistic when making future predictions, which will only become longer and longer...

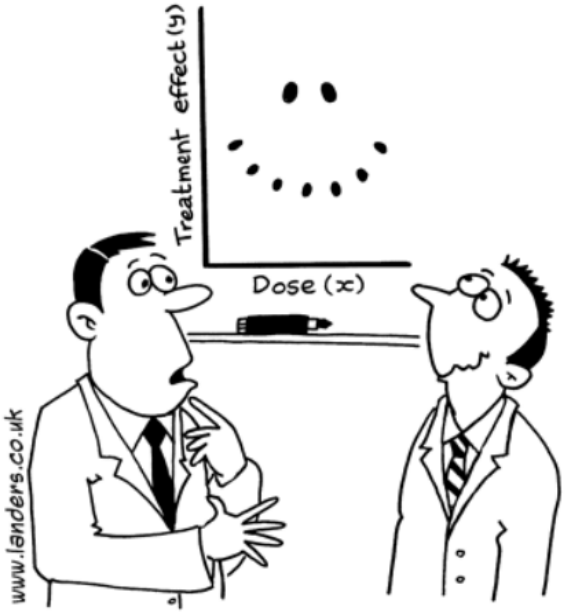
Using time as a linear predictor of future performance, is therefore probably not the best idea.

Increasing Complexity

So far we have seen how to learn the most appropriate parameters for our chosen linear parametric model and mean squared loss function.

We will now consider more complex models based on higher order polynomials, and consider how to determine which model offers the greatest predictive power given the data available.

(You may have noticed that the equation we have derived for our linear model is the same as the maximum likelihood estimator using a Gaussian error model… We will consider this probabilistic interpretation in the next lecture.)



"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

Higher Order Models

We can consider models based on higher order polynomials. For example a cubic relationship between a single input variable ($D = 1$) and an output variable may be defined as

$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \quad = \quad \sum_{i=0}^{3} \theta_i x^i$$

or more generally a kth order polynomial,

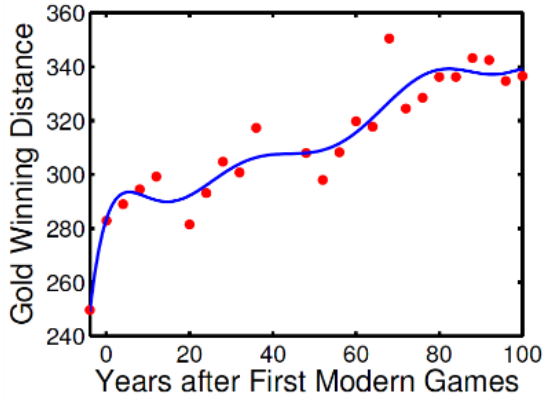
$$h(x) = \sum_{i=0}^{k} \theta_i x^i$$

It should be straightforward to see that the least squares solution we saw previously still holds for higher order polynomials, if we now define

$$\mathbf{x} = \begin{bmatrix} 1 & x^{(1)} & x^{(1)^2} & \dots & x^{(1)^k} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x^{(N)} & x^{(N)^2} & \dots & x^{(N)^k} \end{bmatrix},$$

Some Observations

- The model provides a nonlinear response, since it is based on a higher order polynomial.
- This is still linear regression, since the model is linear in the parameters.
- For higher order polynomials the answer is still analytically tractable.



But is the 9th order polynomial model any better than the 1st order model?

Learning the Model

We have seen how we can learn the *parameters* of the model based on the concept of a loss function.

But if we have multiple models, how do we choose which *model* is the best description of the data?

Clearly if we increase the complexity of the model, we will be able to continually lower the empirical loss function.

Does a lower empirical loss imply a better model?

Cross Validation

A very important approach for improving the predictive ability of these and many other classes of models is *cross validation*.

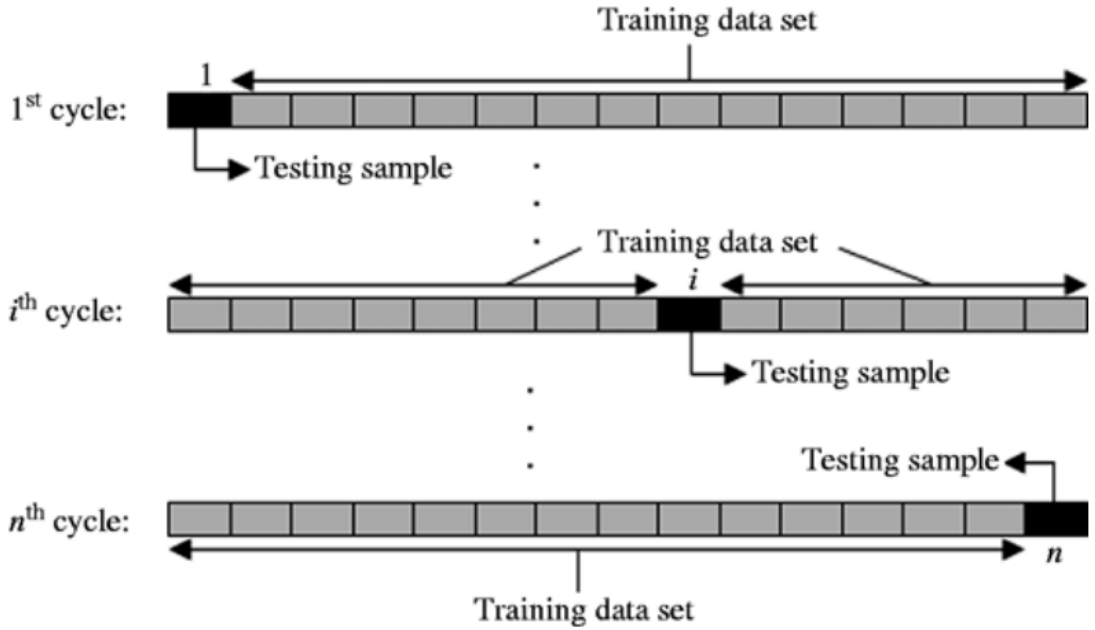
The main idea:

Partition the dataset into 2 disjoint subsets and use one for *training* the model, the other for *validating* the model.

i.e. Learn the parameters from the training data,
then calculate the loss using the validation data.

This procedure can be repeated over multiple subsets of the data and the *average loss* can be used as a performance measure.

LOOCV and K-Fold CV



Leave-one-out cross validation (LOOCV):

Use all subsets consisting of a single data pair as the validation set.

K-fold cross validation:

Partition the data into K equally sized subsets and use these as the validation sets.

Summary

- We have developed a feel for the kinds of problems that may be tackled using machine learning approaches.
- **Supervised learning** algorithms involve labeled data.
- **Parametric models** are very flexible but we must take care when deciding upon which is the most appropriate model.
- **Cross validation** is a powerful technique that allows us to determine the predictive power of a parametric model.

We will now investigate the use of **cross validation** when **choosing between polynomial models** during the computer lab.

In lecture 2 we will investigate cross validation more, and also consider a Bayesian approach to learning linear models.

Highly Recommended Reading (!)

"***Bayesian Reasoning and Machine Learning***" book: Chapters 1, 8, 9, 13

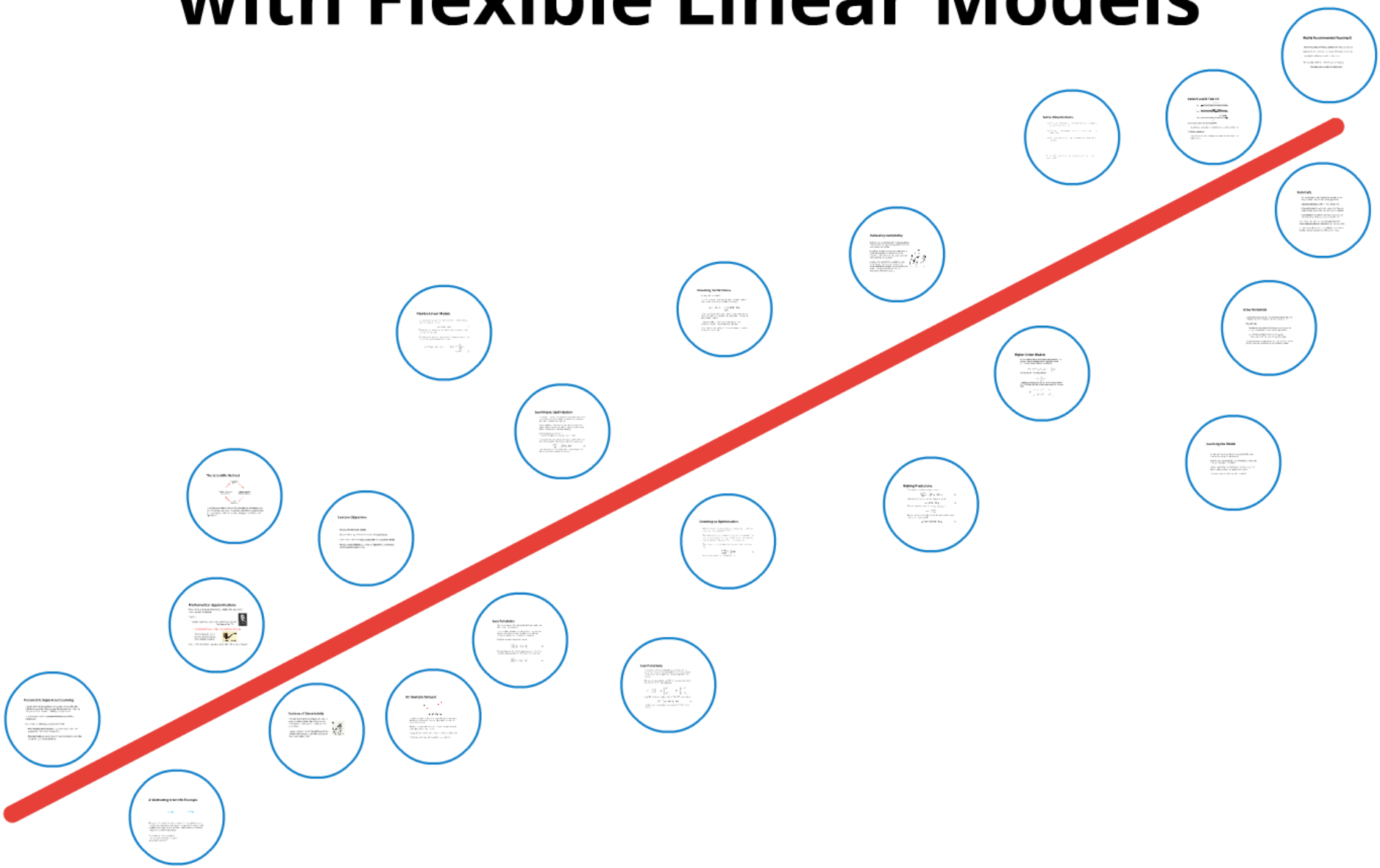
Appendix A in this book covers the required background mathematics:

Linear algebra, multivariate calculus, optimisation.

This is available freely online from the author's webpage:

http://www.cs.ucl.ac.uk/staff/d.barber/brml/

Supervised Learning with Flexible Linear Models



M5MS10
Machine Learning

Spring 2018

Dr Ben Calderhead
b.calderhead@imperial.ac.uk