Introduction to Statistical Learning
with applications in Python

Based on "Introduction to Statistical Learning, with applications in R" by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Introduction
Course Overview & Work Environment

course content & goals, a look at some data, mathematical notation, technicalities, getting ready

Kurt Rinnert

Physics Without Frontiers

ICTP The Abdus Salam International Centre for Theoretical Physics

UNIVERSITY OF LIVERPOOL

Abstract

“If you fail to prepare you are preparing to fail.”
– Anonymous

We’ll discuss the content and goals of the course and introduce some of the datasets we’ll use, some mathematical notation and the work environment.
We will need to spent some time to make sure everyone is technically ready to go.
We’d like to get an impression of what you expect, what you know and what you want to learn.

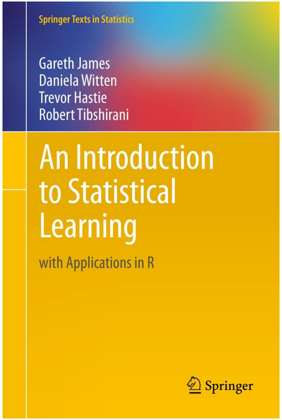
Overview

- Literature
- A first look at some data
- Course outline
- Premises of the course
- Some nomenclature & mathematical notation
- The work environment
- Making sure everything works…

This will get us ready to go.

Literature: The Backbone of the Course

- The course is mainly based on this book.
- It is freely available as a PDF.
- The book's website is linked from each title page.
- The book uses *R* but we will use *Python*.
- Most of the data sets we'll use are the ones from this book.



Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

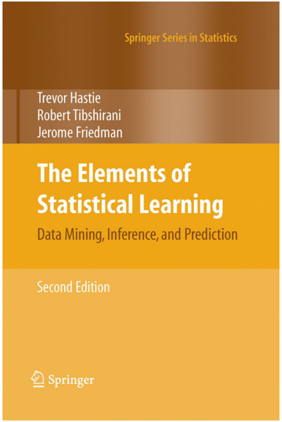
Springer

ISLR

We will add to the content and be slightly more mathematical.

Literature: The more in-depth Tome

- This is a well-known reference text by two of the co-authors of ISLR.
- It also is freely available as a PDF.
- This is the book's website.
- The book covers more topics than ISLR and provides a more formal background.
- We'll use some data sets from this book that are not used in ISLR.



Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

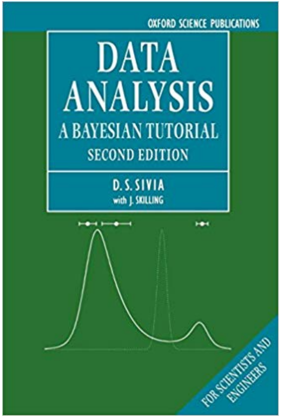
Springer

ESL

Consider this a reference for concepts in ISLR.

Literature: Foundations

- An excellent introduction to statistics, deriving everything from Bayesian probabilities.
- We won't cover this book, but refer to it to lay some foundations.
- Develops the theory around examples.
- Explains well why things are done the way they are.



ABT

Highly recommended!

What is Statistical Learning?

- Statistical learning refers to various tools and methods for *understanding* data.
- We usually distinguish between *supervised* and *unsupervised* methods.
- Supervised methods try to predict *outputs* from *inputs*.
- Unsupervised methods try to find *structures* in the *inputs*.
- We illustrate this briefly with some data sets.

The next lecture will cover this in more detail.

But what about Machine Learning?

> *“Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.”*
>
> *– Wikipedia page on Machine Learning*

> *“"Statistical learning" redirects here.”*
>
> *– Wikipedia page on Machine Learning*

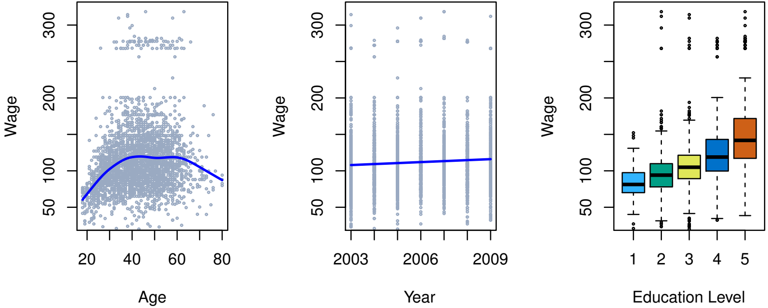
Machine Learning is just a fancy new name. It sure sounds cool!

Wage Data

- Here we refer to the `Wage` data set.
- The `Wage` data set contains data related to a group of males from the US.
- We are interested in how `age`, `education` and calendar `year` affect `wage`.
- We are going to *visualize* the data to get a first understanding of the relationships among the variables.

Note the notation for data sets and variables.

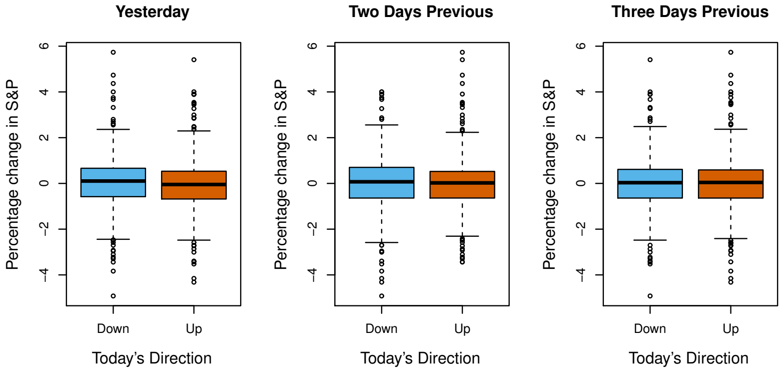
Wage Data



- There is a correlation between age and the average wage.
- There is a slow but steady increase of wage over time.
- The education clearly has an impact on the wage.

Visualization is extremely important. Always have a look first!

Stock Market Data

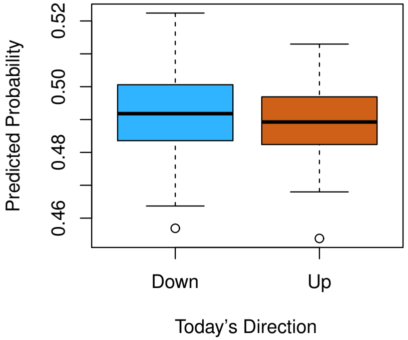


We would like to predict today's market direction from what happened in the previous days.

We can not stress enough the importance of visualization.

Stock Market Data

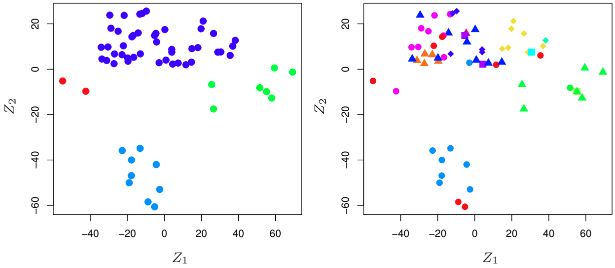
- The Wage data set involved *quantitative* data.
- This is referred to as a *regression* problem.
- Other problems involve predicting *qualitative* data.
- These are *classification* problems.
- For example, predicting whether the stock market goes up or down.
- The Smarket data set provides daily percentage returns for S&P 500 over five years.



Correct prediction 60% of the time using quadratic discriminant analysis.

Obviously, there is a lot of interest in this kind of problem.

Gene Expression Data



- The NCI60 dataset contains 6,830 gene expression measurements for 64 cell lines of 14 different cancer types.
- The figure shows scatter plots of the first two *principal components*, Z_1 and Z_2.
- Cell lines of the same cancer type tend to cluster in $\{Z_1, Z_2\}$ space.

This is an example of *unsupervised* learning.

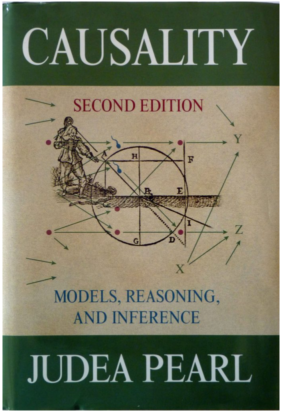
Some Nomenclature

- In most scenarios (certainly in all *supervised* scenarios) there are two types of data:
    - The data we consider the description of a situation.
    - The data we consider the outcome of a situation.
- The former are called *input variables*, *predictors*, *independent variables*, *features* or simply *variables*.
  For example, in the `Wage` data set these are `age`, `year`, `education` and so on.
- The latter are called *output variables*, *dependent variables* or *responses*.
  For example, in the `Wage` data set it would be `wage`.

We'll use the various terms interchangeably but stay true to the concepts.

Beware of Causality Claims

- The distinction between predictors and responses seems to imply a causal connection.
- This is in general *not* the case!
- Be *very* careful about this!
- It is possible, however, to formalize the establishment of causal connections.
- The book on the right is the seminal work on this subject.
- Formalizing causality is far beyond the scope of this course.



CAUS

Don't make formal claims of causality without reading (at least part of) this book.

Course Premises

- Many statistical learning methods are relevant in a wide range of disciplines.
- Statistical Learning should not be viewed as a series of black boxes.
- While it is important to understand the methods, their technical implementation is (mostly) not our concern.
- We presume you are interested in applying statistical learning to real world problems.

We will be more mathematically inclined than the ISLR book. You can take it.

Course Outline

1. Introduction
2. Statistical Learning
3. Linear Regression
4. Classification
5. Resampling Methods
6. Model Selection & Regularization
7. Beyond Linearity
8. Tree-Based Methods
9. Support Vector Machines
10. Unsupervised Learning
11. Neural Networks & Deep Learning
12. Reinforcement Learning

This is the ISLR book content plus neural networks and reinforcement learning.

Notation: Feature Matrix

- The predictors of a data set with p predictors and n observations are represented as a matrix like this:

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- Each *row* represents one observation of all variables.
- Each *column* represents all observations of one variable.

Notation is hard to get right and boring. Please bear with us.

Notation: Feature Rows & Columns

- We might be interested the rows of $\boldsymbol{X}$.
- We write the rows as $x_1, x_2, \ldots, x_n$.
- Each x_i is a *vector* of length p.
- For the `Wage` data set the components of the x_i would be `age`, `education`, `year` and so on.

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}$$

- Or we need to refer to the columns of $\boldsymbol{X}$.
- We write the columns as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p$.
- Each $\boldsymbol{x}_j$ is a *vector* of length n.

$$\boldsymbol{x}_j = \begin{pmatrix} \boldsymbol{x}_{1j} \\ \boldsymbol{x}_{2j} \\ \vdots \\ \boldsymbol{x}_{nj} \end{pmatrix}$$

Note that we represent vectors as *column vectors* by default.

Notation: Transposition

- Now $\boldsymbol{X}$ can be written like this:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_p \end{pmatrix}$$

- Or like this:

$$\boldsymbol{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}$$

- The ^{T} denotes the *transpose* of a matrix:

$$\boldsymbol{X}^T = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} & x_{2p} & \dots & x_{np} \end{pmatrix}$$

- Or of a vector:

$$x_i^T = \begin{pmatrix} x_{i1} & x_{i2} & \dots & x_{ip} \end{pmatrix}$$

Being consistent about transposition is important.

Notation: Response Vector

- We write the ith observation of a response, say wage, as y_i.
- The set of all n observations is then the vector

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

- In general, we write vectors of length n in **bold** face:

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

- However, vectors of different lengths, for example p, are written in normal face:

$$x_i$$

This convention is purely for notational clarity & convenience.

Notation: Summary

With n observations in a data set with p features:

object	space	notation
scalar	$\mathbb{R}$	a
column vector ($k = n$)	$\mathbb{R}^{n \times 1}$ or $\mathbb{R}^n$	$\boldsymbol{a}$
row vector ($k = n$)	$\mathbb{R}^{1 \times n}$ or $\mathbb{R}^n$	$\boldsymbol{a}^T$
column vector ($k \neq n$)	$\mathbb{R}^{k \times 1}$ or $\mathbb{R}^k$	a
row vector ($k \neq n$)	$\mathbb{R}^{1 \times k}$ or $\mathbb{R}^k$	a^T
matrix (r rows, d columns)	$\mathbb{R}^{r \times d}$	$\boldsymbol{A}$
feature matrix	$\mathbb{R}^{n \times p}$	$\boldsymbol{X}$
ith feature row	$\mathbb{R}^{1 \times p}$ or $\mathbb{R}^p$	x_i^T
jth feature column	$\mathbb{R}^{n \times 1}$ or $\mathbb{R}^n$	$\boldsymbol{x}_j$
response vector	$\mathbb{R}^{n \times 1}$ or $\mathbb{R}^n$	$\boldsymbol{y}$

Forgive us for occasionally relaxing some of this in hand writing.

Symbol Manipulation: Matrix Multiplication

- For two matrices $\boldsymbol{A} \in \mathbb{R}^{r \times d}$ and $\boldsymbol{B} \in \mathbb{R}^{d \times s}$ their *matrix product* $\boldsymbol{C} \in \mathbb{R}^{r \times s}$ is:

$$\boldsymbol{C} = \boldsymbol{AB}$$

- The *components* $c_{ij} = (\boldsymbol{C})_{ij} = (\boldsymbol{AB})_{ij}$ are computed as follows:

$$c_{ij} = \sum_{k=1}^{d} a_{ik} b_{kj}$$

- For example:

$$\boldsymbol{AB} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

The ISLR book is a bit shy about using this. We are not.

Symbol Manipulation: Useful Properties

- Matrix multiplication is *not* commutative:

$$\boldsymbol{AB} \neq \boldsymbol{BA}$$

- Matrix multiplication is associative:

$$\boldsymbol{CBA} = \boldsymbol{C}(\boldsymbol{BA}) = (\boldsymbol{CB})\boldsymbol{A}$$

- Matrix multiplication is distributive:

$$\boldsymbol{C}(\boldsymbol{B} + \boldsymbol{A}) = \boldsymbol{CB} + \boldsymbol{CA}$$

- The transpose of a matrix product is:

$$(\boldsymbol{BA})^T = \boldsymbol{A}^T\boldsymbol{B}^T$$

We left out a few "obvious" properties.

Symbol Manipulation: Matrix Inversion

- A *square matrix* $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is *invertible* if, and only if, $\exists \boldsymbol{A}^{-1}$ such that:

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

- Where the components of the *unit matrix* $\boldsymbol{I} \in \mathbb{R}^{d\times d}$ are:

$$(\boldsymbol{I})_{ij} = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$\implies \boldsymbol{I}\boldsymbol{M} = \boldsymbol{M}, \boldsymbol{M} \in \mathbb{R}^{d\times s}$$

- Furthermore:

$$\exists \boldsymbol{A}^{-1} \implies \exists (\boldsymbol{A}^{T})^{-1} = (\boldsymbol{A}^{-1})^{T}$$

Inverting matrices can be hard – we'll leave that to the computer.

Symbol Manipulation: Gradients

- The *gradient* of a *scalar function* $f(a) \in \mathbb{R}$ wrt. a vector $a \in \mathbb{R}^k$ is a *column vector* of dimension k with components $\partial f / \partial a_i$:

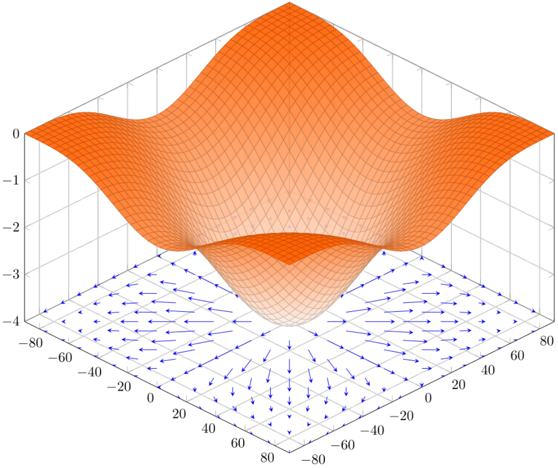
$$\frac{\partial f}{\partial a} = \nabla_a f = \begin{pmatrix} \partial f/\partial a_1 \\ \partial f/\partial a_2 \\ \vdots \\ \partial f/\partial a_k \end{pmatrix}$$

- In particular, the gradient wrt. the vector a of the *scalar product* $a^T b = b^T a$ is:

$$\nabla_a a^T b = \nabla_a b^T a = b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix}$$

This is by no means obvious, but we can't dive deeply into differential geometry here.

The Gradient Illustrated



$f(x, y) = -(\cos^2 x + \cos^2 y)^2$

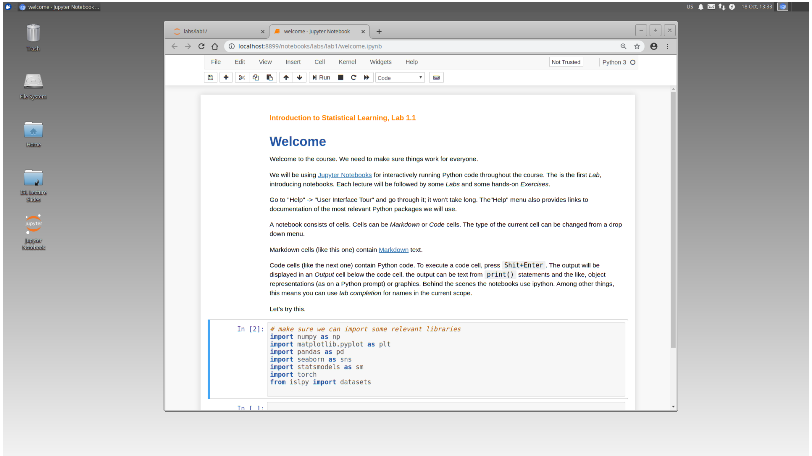
Think of the gradient as the direction and magnitude of the steepest ascent.

The Work Environment

- We'll use Python and a number of common libraries for the labs and exercises.
- For interactive work (most labs and exercises) we'll use jupyter notebooks.
- We have prepared a Python library (`islpwf`) that provides easy access to & documentation for all the data sets.
- We provide a virtual machine running Xubuntu 18.04 with everything pre-installed.

There is not enough room for the logos of all the libraries!

The Virtual Machine



Introduction to Statistical Learning, Lab 1.1

Welcome

Welcome to the course. We need to make sure things work for everyone.

We will be using Jupyter Notebooks for interactively running Python code throughout the course. The is the first *Lab*, introducing notebooks. Each lecture will be followed by some *Labs* and some hands-on *Exercises*.

Go to "Help" -> "User Interface Tour" and go through it; it won't take long. The"Help" menu also provides links to documentation of the most relevant Python packages we will use.

A notebook consists of cells. Cells can be *Markdown* or *Code* cells. The type of the current cell can be changed from a drop down menu.

Markdown cells (like this one) contain Markdown text.

Code cells (like the next one) contain Python code. To execute a code cell, press `Shit+Enter` . The output will be displayed in an *Output* cell below the code cell. the output can be text from `print()` statements and the like, object representations (as on a Python prompt) or graphics. Behind the scenes the notebooks use ipython. Among other things, this means you can use *tab completion* for names in the current scope.

Let's try this.

```
In [2]: # make sure we can import some relevant libraries
        import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        import seaborn as sns
        import statsmodels as sm
        import torch
        from islpy import datasets

In [ ]:
```

Let’s make sure this works for everyone.