Introduction to Machine Learning

Christos Dimitrakakis

August 23, 2024

Outline

Topic

The problems of Machine Learning (1 week)

Introduction

Machine Learning And Data Mining

The nuts and bolts

- Models
- Algorithms
- Theory
- Practice

Workflow

- Scientific question
- Formalisation of the problem
- Data collection
- Analysis and model selection

Types of statistics / machine learning problems

- Classification
- Regression
- Density estimation
- Reinforcement learning

The nuts and bolts

- Models
- Algorithms
- Theory
- Practice

Machine learning

Data Collection

- Downloading a clean dataset from a repository
- Performing a survey
- Scraping data from the web
- Deploying sensors, performing experiments, and obtaining measurements.

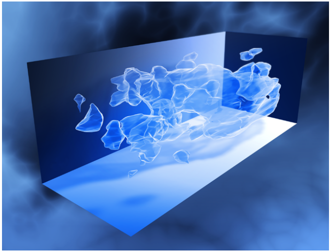
Modelling (what we focus on this course)

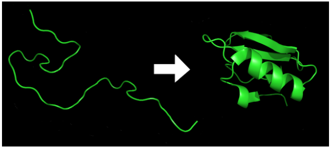
- Simple: the bias of a coin
- Complex: a language model.
- The model depends on the data and the problem

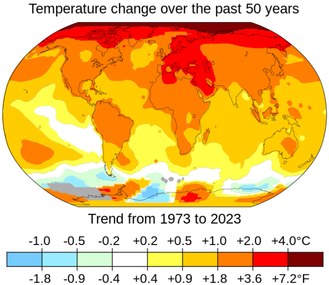
Algorithms and Decision Making

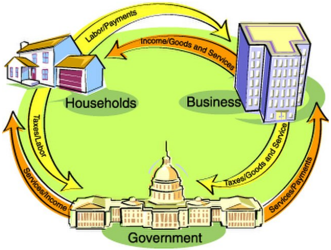
- We want to use models to make decisions.
- Decisions are made every step of the way.
- Decisions are automated algorithmically.

The main problems in machine learning and statistics

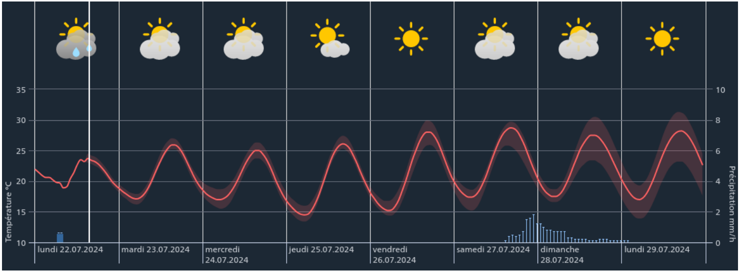






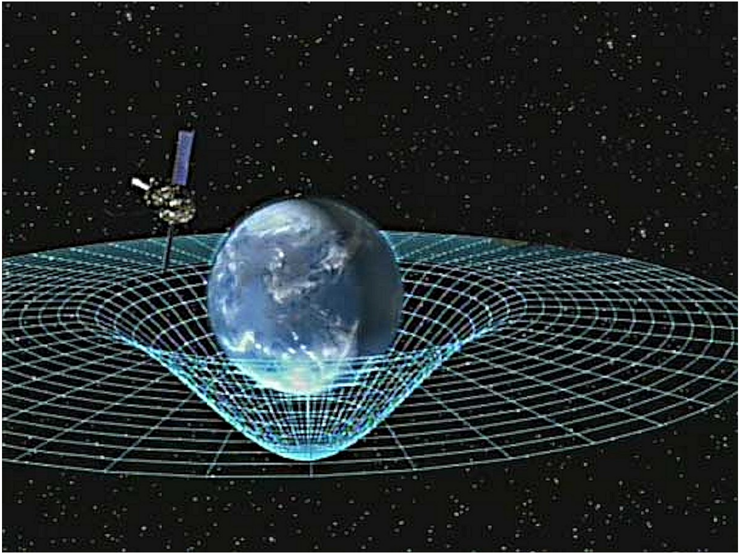


Prediction



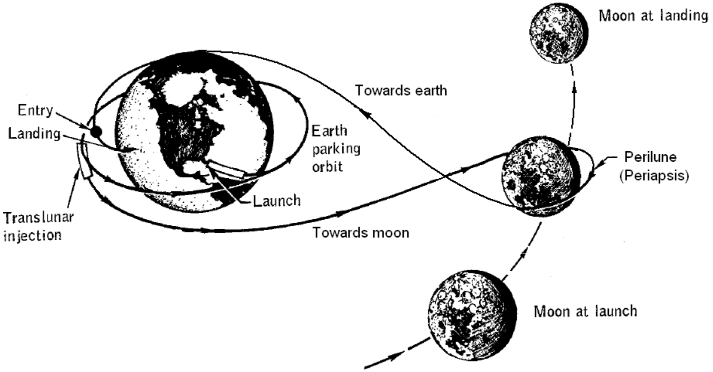
- Will it rain tomorrow?
- How much will bitcoin be worth next year?
- When is the next solar eclipse?

Inference



- Does my poker opponent have two aces?
- What is the law of gravitation?

Decision Making



./fig/artemis.gif

- What data should I collect?
- Which model should I use?
- Should I fold, call, or raise in my poker game?
- How can I get a spaceship to the moon and back?

The need to learn from data

Problem definition

- What problem do we need to solve?
- How can we formalise it?
- What properties of the problem can we learn from data?

Data collection

- Why do we need data?
- What data do we need?
- How much data do we want?
- How will we collect the data?

Modelling and decision making

- How will we compute something useful?
- How can we use the model to make decisions?

Course Material

Moodle

- Assignments and proejct
- Additional reading material
- Asking questions

Course Github https://github.com/olethrosdc/machine-learning-neuch/tree/main/BSc

- .org files for notes, PDF for slides
- source code for examples

Course literature

- An Introduction to Statistical Learning with Python https://hastie.su.domains/ISLP/ISLP_website.pdf.download.html

book chapters will be mentioned in the course

Assignment, teaching and questions

Assignments and project

- Indidivual Weekly assignments in the first half
- Group project in the second half
- Project presentation
- No exam.

Other questions

- Use Moodle for technical/administrative questions: That way everybody gets the same information.
- Use email for personal problems or extra help, if the moodle is not enough.
- Complicated questions can be answered at the next lecture

Office hours

- Fridays 13:00-14:00: book with an email to avoid clashes.
- Email me for an appointment outside those hours.

Topic

Problem definition

- Example: Health, weight and height

Example (Health questions regarding height and weight)

- What is a normal height and weight?
- How are they related to health?
- What variables affect height and weight?

Data collection

Think about which variables we need to collect to answer our research question.

Necessary variables

The variables we need to know about

- Weight
- Height
- Dependent: (health/vote/opinion/salary)

Auxiliary variables

Measurable factors related to the variables of interest

Possible confounders

Hidden factors that might affect variables

Class data and variables

- The class enters their data into the excel file.

- Pay attention to the variables we wish to measure

Privacy

- Is the use of a pseudonym sufficient to hide your identity?

Types of learning problems

Unsupervised learning (unconditional estimation)

- Predict the gender of an unknown individual.
- Predict the height.
- Predict the height and weight?

Supervised learning problems (conditional estimation)

- Classification: Can we predict gender from height/weight?
- Regression: Can we predict weight from height and gender?
- In both cases we predict output variables from input variables

Variables

- Input variables: aka features, predictors, independent variables
- Output variables: aka response, dependent variables, labels, or targets.
- The input/output dichotomy only exists in some prediction problems.

Variables

The class data looks like this

First Name	Gender	Height	Weight	Age	Nationality	Smoking
Lee	M	170	80	20	Chinese	10
Fatemeh	F	150	65	25	Turkey	0
Ali	Male	174	82	19	Turkish	0
Joan	N	5'11	180	21	Brtish	4

- $\boldsymbol{X}$: Everybody's data
- x_t: The t-th person's data
- $x_{t,k}$: The k-th feature of the t-th person.
- $\boldsymbol{x}_k$: Everybody's k-th feature

Raw versus neat data

- Neat data: $x_t \in \mathbb{R}^n$
- Raw data: text, graphs, missing values, etc

Python pandas for data wrangling

Reading class data

```
import pandas as pd
X = pd.read_excel("data/class.xlsx")
X["First Name"]
```

- Array columns correspond to features
- Columns can be accessed through namesx

Summarising class data

```
X.hist()
import matplotlib.pyplot as plt
plt.show()
```

Pandas and DataFrames

- Data in pandas is stored in a DataFrame
- DataFrame is not the same as a numpy array.

Core libraries

```
import pandas as pd
import numpy as np
```

Series: A sequence of values

```
# From numpy array:
s = pd.Series(np.random.randn(3),  index=["a", "b", "c"])
# From dict:
d = {"a": 1, "b": 0, "c": 2}
s = pd.Series(d)
# accessing elemets
s.iloc[2] #element 2
s.iloc[1:2] #elements 1,2
s.array # gets the array object
s.to_numpy() # gets the underlying numpy array
```

DataFrames

Constructing from a numpy array

```
data = np.random.uniform(size = [3,2])
df = pd.DataFrame(data, index=["John", "Ali", "Sumi"],
 columns=["X1", "X2"])
```

Constructing from a dictionary

```
d = {  "one": pd.Series([1, 2], index=["a", "b"]),
       "two": pd.Series([1, 2, 3], index=["a", "b", "c"])}
df = pd.DataFrame(d)
```

Access

```
X["First Name"] # get a column
X.loc[2] # get a row
X.at[2, "First Name"] # row 2, column 'first name'
X.loc[2].at["First Name"] # row 2, element 'first name' of the s
X.iat[2,0] # row 2, column 0
```

Modelling variables

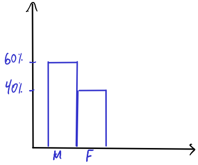


Figure: $x \in \mathbb{N}$

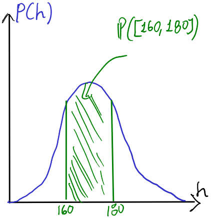


Figure: $x \in \mathbb{R}$

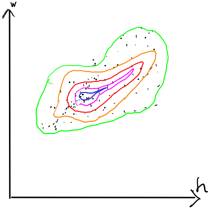


Figure: $x \in \mathbb{R}^2$

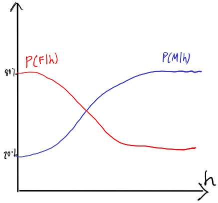


Figure:
$x \in \mathbb{R} \to y \in \mathbb{N}$

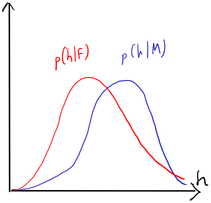


Figure:
$x \in \mathbb{N} \to y \in \mathbb{R}$

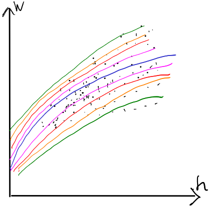


Figure:
$x \in \mathbb{R} \to y \in \mathbb{R}$

Means using python

Example (Calculating the mean of our class data)

```
X.mean() # gives the mean of all the variables through pandas.co
X["Height"].mean()
np.mean(X["Weight"])
```

- The mean here is fixed because we calculate it on the same data.
- If we were to collect new data then the answer would be different.

Example (Calculating the mean of a random variable)

```
import numpy as np
X = np.random.gamma(170, 1, size=20)
X.mean()
np.mean(X)
```

- The mean is random, so we get a different answer everytime.

One variable: expectations and distributions

Definition (The expected value)

Assume $x : \Omega \to \mathbb{R}$, and $\omega_t \sim P$

- $x_1, \ldots, x_t, \ldots, x_T$: random i.i.d. variables with $x_t = x(\omega_t)$
- Ω: random outcome space
- P: distribution of outcomes $\omega \in \Omega$
- $\mathbb{E}_p[x]$: expectation of x under P

$$\mathbb{E}_P[x_t] = \sum_{\omega \in \Omega} x_t(\omega) P(\omega)$$

One variable: expectations and distributions

Definition (The expected value)

Assume $x : \Omega \to \mathbb{R}$, and $\omega_t \sim P$

- $x_1, \ldots, x_t, \ldots, x_T$: random i.i.d. variables with $x_t = x(\omega_t)$
- Ω: random outcome space
- P: distribution of outcomes $\omega \in \Omega$
- $\mathbb{E}_p[x]$: expectation of x under P

$$\mathbb{E}_P[x_t] = \sum_{\omega \in \Omega} x_t(\omega) P(\omega)$$

Definition (The sample mean)

The sample mean of $x_1, \ldots, x_T$ is

$$\frac{1}{T} \sum_{t=1}^{T} x_t$$

Under P, the sample mean is $O(1/\sqrt{T})$-close to the expected value $\mathbb{E}_P[x_t]$.

Reminder: expectations of random variables

A gambling game

What are the expected winnings if you play this game?

- [a] With probability 1%, you win 100 CHF
- [b] With probability 40%, you win 20 CHF.
- [c] Otherwise, you win nothing

Solution

Reminder: expectations of random variables

A gambling game

What are the expected winnings if you play this game?

- [a] With probability 1%, you win 100 CHF
- [b] With probability 40%, you win 20 CHF.
- [c] Otherwise, you win nothing

Solution

- Let x be the amount won, then $x(a) = 100, x(b) = 20, x(c) = 0$.
- We need to calculate

$$\mathbb{E}_P(x) = \sum_{\omega \in \{a,b,c\}} x(\omega) P(\omega) = x(a)P(a) + x(b)P(b) + x(c)P(c)$$

- $P(c) = 59\%$, as $P(\Omega) = 1$. Substituting,

$$\mathbb{E}_P(x) = 1 + 8 + 0 = 9.$$

Models

Models as summaries

- They summarise what we can see in the data
- The ultimate model of the data is the data

Models as predictors

- They make predictions about things beyond the data
- This requires some assumptions about the data-generating process.

Example models

- A numerical mean
- A linear classifier
- A linear regressor
- A deep neural network
- A Gaussian process
- A large language model

The simplest model: A mean

Modelling variables

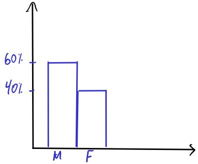


Figure: $x \in \mathbb{N}$

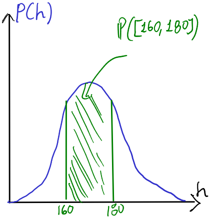


Figure: $x \in \mathbb{R}$

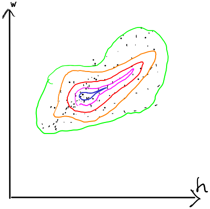


Figure: $x \in \mathbb{R}^2$

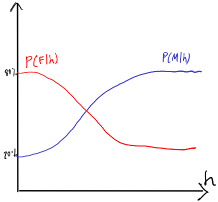


Figure:
$x \in \mathbb{R} \rightarrow y \in \mathbb{N}$

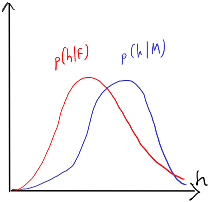


Figure:
$x \in \mathbb{N} \rightarrow y \in \mathbb{R}$

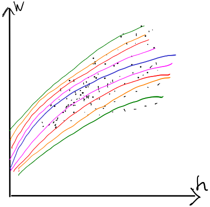


Figure:
$x \in \mathbb{R} \rightarrow y \in \mathbb{R}$

The Bernoulli distribution

Definition (Bernoulli distribution)

We say that $x \in \{0, 1\}$ has Bernoulli distribution with parameter θ and write

$$x \sim \text{Bernoulli}(\theta),$$

when

$$\mathbb{P}(x) = \begin{cases} \theta & x = 1 \\ 1 - \theta & x = 0. \end{cases}$$

Example (Applications of the Bernoulli distribution)

- A biased coin flip.
- Classification errors.

Predicting y from x, discrete case.

Consider two variables, x, y. We can either care about

- $\mathbb{E}[y|x]$ the expectation of y for all x.
- $\mathbb{P}[y|x]$ the distribution of y for all x.

Probability table for $P(x, y)$

$P(x, y)$	y = 0	y = 1
x = 0	54%	6%
x = 1	16%	24%

- What is $P(x)$?

Conditional probability table for $P(y|x)$

$P(y \mid x)$	y = 0	y = 1
x = 0	90%	10%
x = 1	40%	60%

- What is $\mathbb{E}[y \mid x]$?

Distributions of two variables

In this setting, both x and y have a Bernoulli distribution. If we use a model whereby x is sampled first, and then y, then we can define two Bernoulli distributions. The first, for x is unconditional, while the second, for y, depends on the value of x:

$$x \sim \mathrm{Bernoulli}(\theta)$$
$$y \mid x \sim \mathrm{Bernoulli}(\phi_x).$$

In our example, $\phi_0 = 0.1$ and $\phi_1 = 0.6$.

Homework

Probability table for $P(x, y)$

$P(x, y)$	y = -1	y = 0	y = 1
x = 0	10%	20%	10%
x = 1	30%	20%	10%

Given the above table, calculate

- $P(x)$
- The conditional probability table for $P(y|x)$.
- $\mathbb{E}[y|x]$ for all values of x.

Two variables: conditional expectation

The height of different genders

The conditional expected height

$$\mathbb{E}[h \mid g = 1] = \sum_{\omega \in \Omega} h(\omega) P[\omega \mid g(\omega) = 1]$$

The empirical conditional expectation

$$\mathbb{E}[h \mid g = 1] \approx \frac{\sum_{t : g(\omega_t) = 1} h(\omega_t)}{|\{t : g(\omega_t) = 1\}|}$$

Python implementation

Two variables: conditional expectation

The height of different genders

The conditional expected height

$$\mathbb{E}[h \mid g = 1] = \sum_{\omega \in \Omega} h(\omega) P[\omega \mid g(\omega) = 1]$$

The empirical conditional expectation

$$\mathbb{E}[h \mid g = 1] \approx \frac{\sum_{t : g(\omega_t) = 1} h(\omega_t)}{|\{t : g(\omega_t) = 1\}|}$$

Python implementation

```
h[g==1] / sum(g==1)
## alternative
import numpy as np
np.mean(h[g==1])
```

Topic

Populations, samples, and distributions

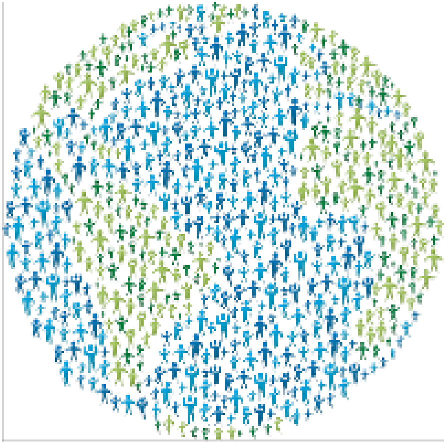


Figure: The world population

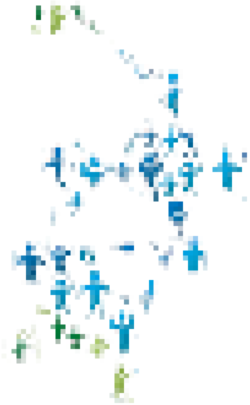


Figure: A sample

Statistical assumptions

Independent, Identically Distributed data

- $\omega_t \sim P$: individuals $\omega_t \in \Omega$ are drawn from some distribution P
- $\boldsymbol{x}_t \triangleq \boldsymbol{x}(\omega_t)$ are some features of the t-th individual
- Here we are interested in properties of the unknown distribution P.

Representative sample from a fixed population

- Finite population $\Omega = \{\omega_1, \omega_2, \ldots, \omega_N\}$
- A subset $S \subset \Omega$ of size $T < N$ is selected with a uniform distribution, i.e. so that

$$P(S) = T/N, \qquad \forall S \subset \Omega.$$

- Here we are interested in statistics of the unknown population Ω.
- We assume an underlying distribution P for convenience.
- We can tried both cases essentially the same.

Learning from data

Unsupervised learning

- Given data $x_1, \ldots, x_T$.
- Learn about the data-generating process.
- Example: Estimation, compression, text/image generation

Supervised learning

- Given data $(x_1, y_1), \ldots, (x_T, y_T)$
- Learn about the relationship between x_t and y_t.
- Example: Classification, Regression

Online learning

- Sequence prediction: At each step t, predict x_{t+1} from $x_1, \ldots, x_t$.
- Conditional prediction: At each step t, predict y_{t+1} from $x_1, y_1 \ldots, x_t, y_t, x_{t+1}$

Reinforcement learning

Learn to act in an unknown world through interaction and rewards

Robust models of the mean

Validating models

Training data

- Calculations, optimisation
- Data exploration

Validation data

- Fine-tuning
- Model selection

Test data

- Performance comparison

Simulation

- Interactive performance comparison
- White box testing

Real-world testing

- Actual performance measurement

Model selection

- Train/Test/Validate
- Cross-validation
- Simulation

Topic

Course Contents

Models

- k-Nearest Neighbours.
- Linear models and perceptrons.
- Multi-layer perceptrons (aka deep neural networks).
- Bayesian Networks

Algorithms

- (Stochastic) Gradient Descent.
- Bayesian inference.

Supervised learning

The general goal is learning a function $f : X \to Y$.

Classification

- Input data $x_t \in \mathbb{R}$, $y_t \in [m] = \{1, 2, \ldots, m\}$
- Learn a mapping f so that $f(x_t) = y_t$ for unseen data

Regression

- Input data x_t, y_t
- Learn a mapping f so that $f(x_t) = \mathbb{E}[y_t]$ for unseen data
- Can be mapped into classification by binning.

Unsupervised learning

The general goal is learning the data distribution.

Density estimation

- Input data $x_1, \ldots, x_T$ from distribution with density p
- Problem: Estimate p.

Special case: Compression

- Learn two mappings c, d
- $c(x)$ compresses an image x to a small representation z.
- $d(z)$ decompresses to an approximate datapoint $\hat{x}$.

Special case: Clustering

- Input data $x_1, \ldots, x_T$.
- Estimate latent cluster labels c_t to model the distribution of x as

a mix over densities p_c.

$$p(x_t) = \sum_c P(c_t = c) p_c(x_t)$$

Supervised learning objectives

- Data (x_t, y_t), $x_t \in X$, $y_t \in Y$, $t \in [T]$.
- i.i.d assumption: $(x_t, y_t) \sim P$ for all t.
- Supervised decision rule $\pi(a_t | x_t)$

Classification

- Predict the labels correctly, i.e. $a_t = y_t$.
- Have an appropriate confidence level

Regression

- Predict the mean correctly
- Have an appropriate variance around the mean

Unsupervised learning objectives

- Reconstruct the data well
- Be able to generate data

Reinforcement learning objectives

- Maximise total reward

Pitfalls

Reproducibility

- Modelling assumptions
- Distribution shift
- Interactions and feedback

Fairness

- Implicit biases in training data
- Fair decision rules and meritocracy

Privacy

- Accidental data disclosure
- Re-identification risk

Topic

Reading for this week

ISLP Chapter 1