(1)

MARC TRUTER
HEFIN LAMBLEY

Machine learning reading group, talk 1 (Introduction)

What is the reading group about?

A brief introduction to the core concepts in machine learning, assuming no prior knowledge of statistics or programming, but a strong (masters/phd) background in mathematics.

Goals: By end of reading group should be able to:
① understand main tasks ML can be used for
② understand selection of most common methods in ML
③ be able to implement simple algorithms in python and use prebuilt packages for more complex methods
④ have an awareness of the theoretical garuntees and possible failure of different methods

---

Structure of reading group

Each talk : ~30 mins theory + ~30 mins practical implentation in python

In first talks, the practical section will cover setting up python and basic use of relevant packages.

For rest of term : Part I (weeks 3-8) : Supervised learning
Part II (weeks 9-10) : Unsupervised learning

For the rest of the talk we will explain what supervised and unsupervised learning is, and why they are useful.

What is ML?

In a nutshell, using algorithms which learn from data to solve tasks.

Example (Supervised learning)

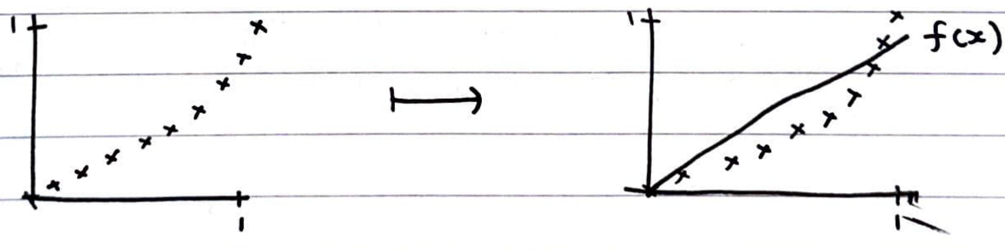
Let X and Y be metric spaces, $f^+ : X \to Y$ a function. Assume we want to approximate f^+, which we don't know using data

$$\{ (x_1, f^+(x_1) = y_1), \dots, (x_N, f^+(x_N) = y_N) \} \subset X \times Y$$

A supervised-learning algorithm takes this data as an input and returns a function $f : X \to Y$ for which $d_Y(f^+(x), f(x))$ is small, (at least for x that is close to training data).

Concrete example

$$X = Y = \mathbb{R}^2, \quad f^+(x) = x^2, \quad \text{data} : \{ (\overset{x_i}{\overset{\parallel}{\tfrac{i}{N}}}, \overset{(\frac{i}{N})^2}{\overset{\parallel}{f^+(\tfrac{i}{N})}}) \}_{i=0}^{N}$$



Least squares regression is a supervised ML algorithm that tries to find $f(x) = ax + b$ to minimise

$$\sum_{i=0}^{N} |(ax_i + b) - f^+(x_i)|^2$$

This makes the error between f^+ and f small for $x \in [0,1]$ ("interpolation") but the error will be big for $x >> 1$ ("extrapolation").

Practical example (see code session)

We'll see a dataset where $X = Y = \mathbb{R}^3$ where we have $N=20$ data pairs $\{(x_1, y_1), \dots, (x_N, y_N)\}$ with

$x_i = $ (# chin ups, # sit ups, # jumps)

$f^+(x_i) = y_i = $ (weight, waist, pulse)

of person i in a physiological experiment.

Goal: learn underlying mapping $f^+: X \to Y$, for at least "realistic" values of $x \in X$ similar to training data.

Rmk: In next talk we will consider a more sophisticated approach where we don't assume a one-to-one mapping $f^+: X \to Y$.

Practical example (classification of images)

The previous examples have all been an ~~regression~~, predicting a continuous quantity eg $y \in Y = \mathbb{R}^3$. A related problem, which is mathematically the same, is predicting a discrete quantity eg

$X = \{$ set of $d \times d$ pixelated images of either cats or dogs $\}$
$\subseteq [0,1]^{d \times d \times 3}$ ← red, green or blue (RGB)
$Y = \{$cat, dog$\}$

Goal: use data $\{(x_i, y_i)\}_{i=1}^N$ to learn the map $f^+: X \to Y$ that gives each image the correct label.

This is typically done by convolutional neural networks (week 7)

Part I will be dedicated to supervised learning. But there are lots of tasks other than supervised learning, this will be the focus of part II (§2.7 Bach)

The distinction is between <u>supervised</u> and <u>unsupervised</u> with semi-supervised learning in between (wait discuss).

Supervised learning (e.g classification, regression)	Semi-supervised learning	Unsupervised learning (e.g generative modelling, clustering, dimension reduction, ...)
all data $x \in X$ has a label $y \in Y$	some data have known labels, some don't	none of data $x \in X$ have labels

---

<u>Summary of unsupervised learning methods</u>

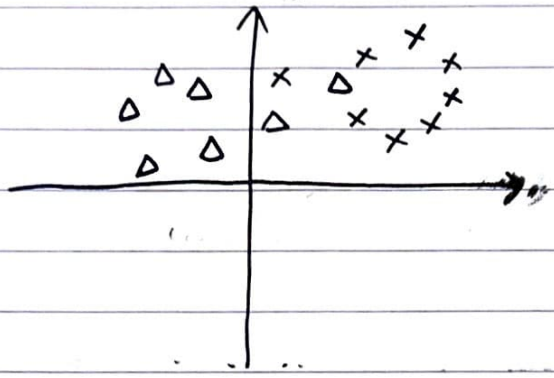
① <u>Generative modelling</u> : Assume $\{x_i\}_{i=1}^N$ are independent samples from some prob. distribution μ^+ on the metric space X. would like to learn a prob. dist. μ on X that is "close" to μ^+ in some <u>distance</u> on space of prob dist on X.

e.g Kullbach-Leibler, divergence-area, relative entropy or Wasserstein distance.

Practical ability: large language models (LLMs), let $\{x_i\}_{i=1}^N$ be sentences downloaded from internet, learning μ allows us to generate new sentences

Rmk: for a useful LLM, use <u>conditional generation</u>, don't just generate random sentences, generate something conditioned on the user's input

② <u>Clustering</u>: Given data $\{x_i\}_{i=1}^N$ which we postulate arise from K "clusters", come up with a decision rule to seperate the data.



Suppose there are two distinct groups $\times$ and Δ, we know there are 2 groups but not the labels $\times$ and Δ, just coordinates $x_i \in \mathbb{R}^2$.

want to learn which data gets which label. ....

---

③ <u>Dimension reduction</u>: Given data $\{x_i\}_{i=1}^N \sim \mu^+$, learn a map $f: X \to Z$ with $\dim Z \ll \dim X$ where $f(x) \in Z$ meaningfully preserves some features of $x \in X$.

<u>Example</u> (Autoencoders, week 10)

Given $\{x_i\}_{i=1}^N \sim \mu^+$, learn maps $f: X \to Z$ and $g: Z \to X$ such that $g \circ f(x) \approx x$ for $x \sim \mu^+$.