Introduction to Machine Learning

Introduction: Supervised Learning

compstat-lmu.github.io/lecture_i2ml

FUNCTIONS - FROM OBJECT TO OUTCOME

- In **supervised ML** we try to find a mapping $f()$ that connects (the description of) an object to a certain aspect of that object.

$$f(\text{object}) = \text{target}$$

- So we would like to compute / predict / describe the target outcome, given an object from a certain set of admissible objects.
- This mapping f must be computable by a machine.

COMPUTABLE FUNCTIONS

- This seems like a very general and powerful concept. Computable function are also called *algorithms*.
- For many kinds of problems, the mapping f can be constructed explicitly, either as an actual mathematical mapping or in terms of a finite number of well-defined computable steps, i.e., as an algorithm that computes f, for example:
  - $f(\text{list of elements}) = \text{count of distinct elements}$
  - $f(\text{polynomial function}) = \text{locations of extrem points}$
  - $f(\text{map of roads}) = \text{shortest route from A to B}$
- Computer scientists study which kinds of functions are computable, how algorithms to compute a given f scale in terms of input sizes, and how to properly construct and implement them.
- Obviously, such manual or explicit construction of f varies from entirely simple and obvious to nearly or fully impossible.

LEARNING FUNCTIONS ON A MACHINE

- For many important problems, defining and implementing *f* "manually" is difficult to impossible, e.g.:

$$f(\text{person}) = \text{probability of developing breast cancer}$$

- **Supervised ML aims at automatically constructing *f* from a set of example objects where we already know the target outcome.** This mapping *f* is then called a **model**.
- Humans are pretty good at this, at least for some domains. We learn that for objects exhibiting certain patterns or properties, certain outcomes are much more likely.
  This pattern recognition occurs implicitly and gradually through examples over time, without an explicit construction of an "algorithm". Hence, *learning*.

LEARNING FUNCTIONS ON A MACHINE

- *Supervised* learning because the target is known for all the example objects we use for learning, so we can “supervise” how well a model f works by comparing its outputs (i.e, its **predictions**) to the target outcomes we have actually observed.

FEATURE VECTORS

- In order to simplify this problem we (usually) assume that the input for the model f is not the (often complex) object itself, but rather a list of **features** that describe it.
- This list of features must be available for all objects, i.e., measurable or computable.
- E.g.,

$$f(\text{age, sex, gene expression, ...}) = \text{prob. of developing breast cancer}$$

PARAMETERS, STATISTICS AND SUPERVISED MACHINE LEARNING

- Supervised ML additionally assumes that f is of a certain "form" or comes from a certain *class of functions*.
  This is necessary to make the problem of automatically finding a "good" model feasible at all.
- The specific behavior of a mapping from this class can then be described by **parameters** which defines its shape.
- Statistics also studies how to learn such functions (or, rather: their parameters) from example data and how to perform inference on them and interpret the results.
- For historical reasons, statistics is mostly focused on fairly simple classes of mappings, like (generalized) linear models.
- Supervised ML also includes more complex kinds of mappings that can often deal with more complicated and high-dimensional inputs.

SUMMARY

Supervised machine learning is concerned with learning a function that predicts a certain **target** from an object's **features** from a set of examples for which both the features and the target are known.
The function to be learned is restricted to come from a certain class of functions and its precise shape is defined in terms of a set of **parameters**.