"Sapienza" University of Rome
Faculty of Information Engineering,
Informatics and Statistics
Department of Computer Science

Machine Learning

Lecture notes integrated with the book TODO

Author
Alessio Bandiera

October 1, 2024

Contents

Information and Contacts

Personal notes and summaries collected as part of the *Machine Learning* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link:
https://github.com/aflaag-notes. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: **alessio.bandiera02@gmail.com**
- LinkedIn: **Alessio Bandiera**

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

TODO

Licence:

1

TODO

1.1 Learning problems

1.1.1 TODO

A **machine learning problem** is defined by the following *three components*.

Definition 1.1.1.1: Learning

Learning is defined as *improving*, through *experience* E, at some *task* T, with respect to a *performance measure* P.

Example 1.1.1.1 (Machine Learning problem)**.** Consider the problem of learning how to play Checkers; in this example, the *task* T is to be able to play the game itself, the *performance measure* P could be the percentage of games won in a tournament, but *experience* E is more complex.

In general, *experience* can be acquired in several ways:

- in this example, a human expert may suggest optimal moves for each configuration of the board; however, this approach may not generalize for any problem, as human experts may not exists for certain tasks;
- alternatively, the computer may play against a human, and automatically detect win, draw and loss configurations;
- lastly, the computer may play against itself, learning from its own successes and failures.

For this particular game, a possible **target function** (the function that would be useful to learn in order to solve the learning problem) could be the following

$$\text{ChooseMove} : \text{Board} \to \text{Move}$$

which, given a board state, returns the best move to perform, but also

$$V : \text{Board} \to \mathbb{R}$$

which assigns a *score* to a given board.

For instance, consider the following target function:

$$\hat{V}(b) = w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b)$$

where b is a given *board state*, and

- $bp(b)$ is the number of *black pieces*
- $rp(b)$ is the number of *red pieces*
- $bk(b)$ is the number of *black kings*
- $rk(b)$ is the number of *red kings*
- $bt(b)$ is the number of *red pieces threatened by black pieces*
- $rt(b)$ is the number of *black pieces threatened by red pieces*

In this formulation, $\hat{V}$ is a *linear compbination* of multiple coefficients w_i, which are unknown. Therefore, in this example **goal** of the *learning problem* is to **learn** $\hat{V}$, or equivalently, to **estimate each coefficient** w_i. Note that this function *can be computed*.

Definition 1.1.1.2: Dataset

Let $V(b)$ be the *true target function* (always *unknown*), $\hat{V}(b)$ be the *learned function* — an approximation of $V(b)$ computed by the *learning algorithm* — and $V_t(b)$ the *training value* of b in the *training data*. A **dataset** is a set of samples, denoted as

$$D = \{(b_i, V_t(b_i)) \mid i \in [1, n]\}$$