MACHINE LEARNING

Introduction

Last Update: 6th October 2022

Prof. Dr. Shadi Albarqouni

Director of Computational Imaging Research Lab. (Albarqouni Lab.)
University Hopsital Bonn | University of Bonn | Helmholtz Munich

STRUCTURE

INTRODUCTION

WHAT IS MACHINE LEARNING (ML)?

Definition (Tom Mitchell)

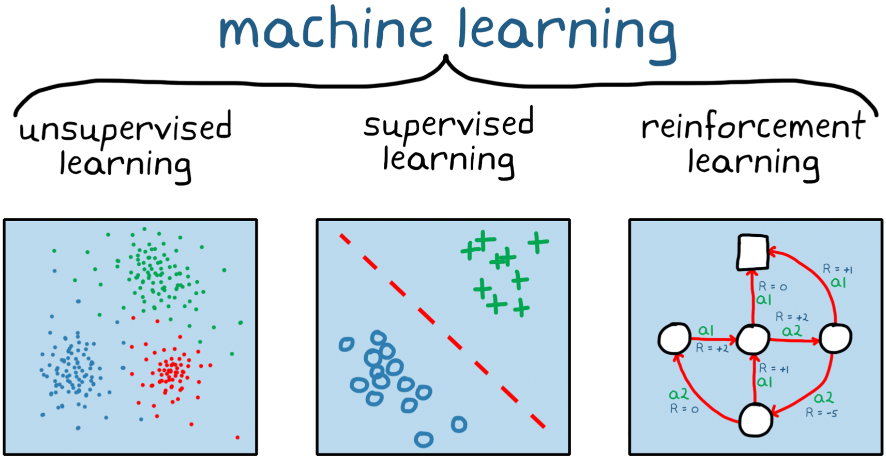
A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

PROBABILISTIC PERSPECTIVE

We will cover most types of ML, however, from a probabilistic perspective for two reasons:

- it is the optimal approach to decision making under uncertainty
- probabilistic modeling is the language used by most other areas of science and engineering, and thus provides a unifying framework between these fields.

TYPES OF MACHINE LEARNING



Source: https://www.mathworks.com/discovery/reinforcement-learning.html

SUPERVISED LEARNING

SUPERVISED LEARNING

The most common form of ML is supervised learning.

Definition

The task T is to learn a mapping $f(\cdot)$ from inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$.

The inputs x are also called the features, covariates, or predictors; this is often a fixed-dimensional vector of numbers, such as the height and weight of a person, or the pixels in an image. In this case, $\mathcal{X} = \mathbb{R}^D$, where D is the dimensionality of the vector (i.e., the number of input features).

The output y is also known as the label, target, or response.

The experience E is given in the form of a set of N input-output pairs $\mathcal{D} = \{(x_n; y_n)\}_{n=1}^{N}$, known as the training set. (N is called the sample size.)

The performance measure P depends on the type of output we are predicting.

CLASSIFICATION

Definition

The problem of predicting the class label y given an input x is often called classification or pattern recognition

The class label y belongs to a set of C unordered and mutually exclusive labels known as classes, $\mathcal{Y} = \{1, 2, \ldots, C\}$.

Special case: If there are just two classes, often denoted by $y \in \{0, 1\}$ or $y \in \{-1, +1\}$, it is often called binary classification.

CLASSIFICATION

Example: classifying Iris flowers

High dimensionality | domain experts | Images vs. tabular data

Given a 150 pictures of Iris flowers with the following features; sepal length, sepal width, petal length, and petal width along with their class labels; (a) Setosa, (b) Versicolor and (c) Virginica, train a classification model to recognize different types of Iris flowers.

$$D = \ldots, N = \ldots, C = \ldots$$

$$x \in \{\ldots\}$$

$$y \in \{\ldots\}$$

(a) (b) (c)

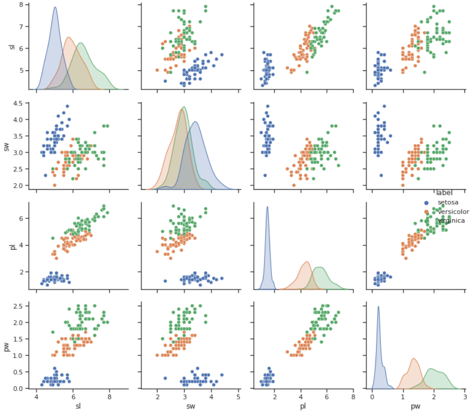
index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

CLASSIFICATION

Exploratory data analysis (Code)

Data exploration | pair plot | dimensionality reduction

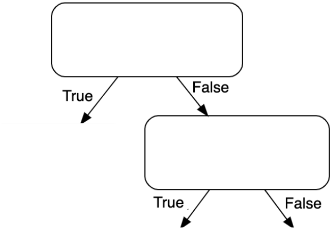
```python
import numpy as np
import matplotlib.pyplot as plt
import os
try:
    import probml_utils as pml
except ModuleNotFoundError:
    %pip install -qq git+https://github.com/probml/probml-utils.git
    import probml_utils as pml
import seaborn as sns;
sns.set(style="ticks", color_codes=True)

try:
    import pandas as pd
except ModuleNotFoundError:
    %pip install -qq pandas
    import pandas as pd
pd.set_option('display.precision', 2) # 2 decimal places
pd.set_option('display.max_rows', 20)
pd.set_option('display.max_columns', 30)
pd.set_option('display.width', 100) # wide windows

try:
    import sklearn
except ModuleNotFoundError:
    %pip install -qq scikit-learn
    import sklearn
from sklearn.datasets import load_iris
iris = load_iris()

# Extract numpy arrays
X = iris.data
y = iris.target
```

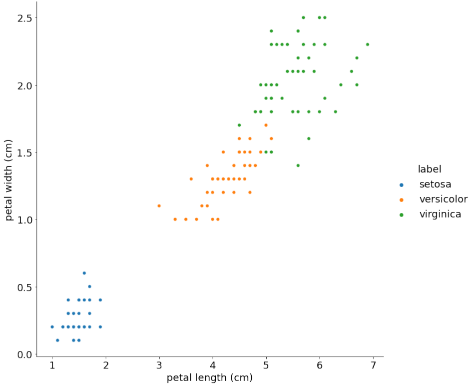


CLASSIFICATION

Learning a classifier (Code)

decision rule | decision boundary | decision tree | parameters

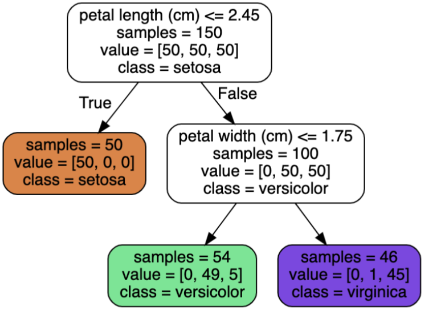


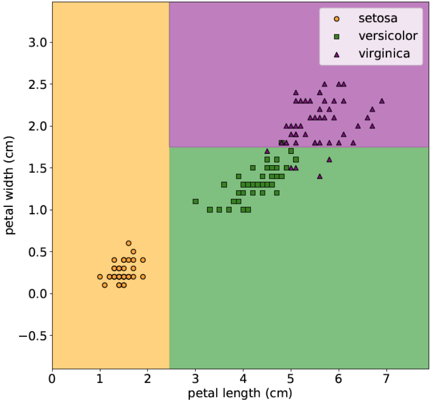


CLASSIFICATION

Learning a classifier (Code)

decision rule | decision boundary | decision tree | parameters





CLASSIFICATION

Empirical risk minimization

miclassification rate | loss function | uncertainty

Empirical risk

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N} \ell(y_n, f(\mathbf{x}_n; \theta))$$

where θ is the model parameters, $\ell(y, \hat{y}) = \mathbb{I}(y \neq \hat{y})$ and $\mathbb{I}(e) = \begin{cases} 1 & \text{if} \quad e \quad \text{is true} \\ 0 & \text{otherwise} \end{cases}$.

Compute the empirical risk for the previous Iris flowers example.
Have a look at different loss functions, e.g., negative log liklihood (Sec. 1.2.1.6).

CLASSIFICATION

Empirical risk minimization

model fitting | loss function | uncertainty

Empirical risk minimization

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, f(\mathbf{x}_n; \theta))$$

Our goal is to minimize the expected loss on the training data (model fitting) as well as the unseen data (generalization), e.g., validation and testing sets.

CLASSIFICATION

Empirical risk minimization

model fitting | loss function | uncertainty

In many cases, we will not be able to perfectly predict the exact output $f(x;\theta)$ given the input x

- due to lack of knowledge of the input-output mapping $f : \mathcal{X} \rightarrow C$ (this is called epistemic uncertainty or model uncertainty) and/or
- due to intrinsic (irreducible) stochasticity in the mapping (this is called aleatoric uncertainty or data uncertainty).

Why is it so important to represent uncertainty in our predictions? how can we express and quantify such uncertainty?

REGRESSION

Definition

The problem of predicting a real-valued quantity $y \in \mathbb{R}$ given an input x is often called regression.

- examples: degree of toxicity if the flower is eaten, the average height of the plant, the life expectancy of the flower.
- common loss function is the quadratic loss, $\ell_2(y, \hat{y}) = (y - \hat{y})^2$
- Have a look at different loss functions, e.g., negative log liklihood (Sec. 1.2.1.6), and huber loss (Sec. 5.1.5)

REGRESSION

Example: linear regression

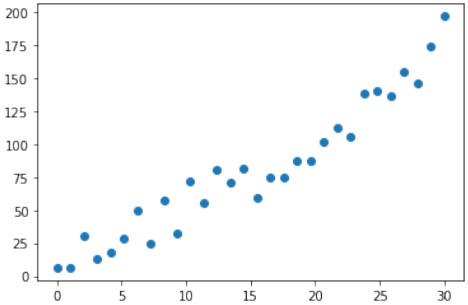
(Code)

Given the income of a freelancer for the last 30 weeks, train a regression model to predcit the income for the next 4 weeks.

$$D = \ldots, N = \ldots, C = \ldots$$

$$x \in \{\ldots\}$$

$$y \in \{\ldots\}$$



REGRESSION

Example: linear regression

simple linear regression | loss function

To model the given data in the previous example, you need to make an assumption that you data follows a linear function, e.g., $f(\mathbf{x};\theta) = b + \mathbf{w}^T\mathbf{x}$, and your main objective is find the model parameters $\theta = (b, \mathbf{w})$ where b as an offset or bias, and $\mathbf{w}$ as weights or regression coefficients.

Least squares solution

$$\widehat{\theta} = \arg\min_{\theta} MSE(\theta) \triangleq \frac{1}{N}\sum_{n=1}^{N} \ell_2(y_n, f(\mathbf{x}_n;\theta))$$

REGRESSION

Example: linear regression

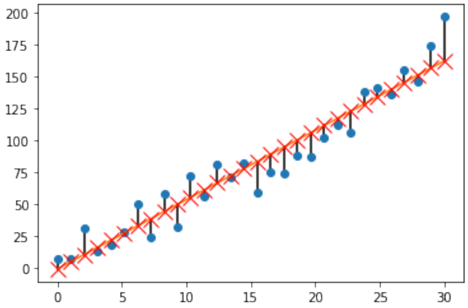
High dimensionality | feature engineering | polynomial regression

What happens if you have multiple features; e.g., internet connectivity (bandwidth), no. of assignments/homework per week, public/seasonal holidays, exchange rate ... etc.?

feature engineering $\rightarrow f(\mathbf{x};\omega) = \omega^T\phi(\mathbf{x}) \triangleq [1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2]$

deep neural networks

$\rightarrow f(\mathbf{x};\theta) = f_L(f_{L-1}(\ldots(f_1(\mathbf{x})\ldots)))$



WHICH MODEL IS THE BEST?

No Free Lunch Theorem

UNSUPERVISED LEARNING

UNSUPERVISED LEARNING

Definition

The task T is to fit an unconditional model of the form $p(\mathbf{x})$ given observed inpputs $\mathbf{x}$ without any corresponding outputs y.

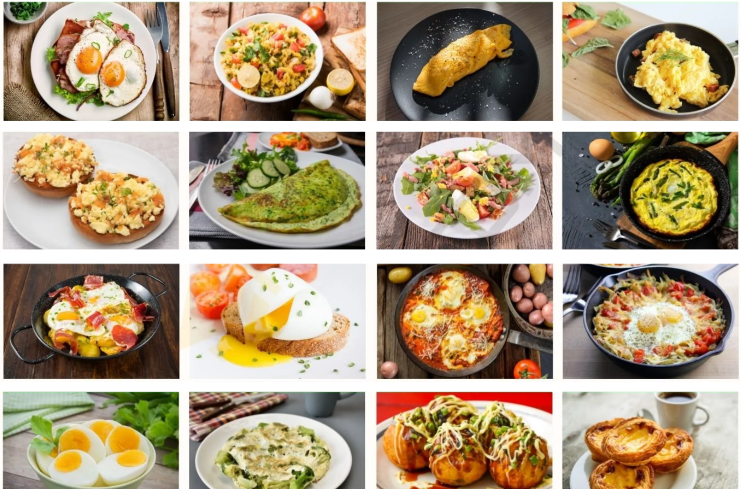
When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information. You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^{9} seconds. So it's no use learning one bit per second. You need more like 10^{5} bits per second. And there's only one place you can get that much information: from the input itself. — Geoffrey Hinton, 1996

UNSUPERVISED LEARNING

This is an Egg

UNSUPERVISED LEARNING

This is an Egg



UNSUPERVISED LEARNING

UNSUPERVISED LEARNING

Example: clustering

The goal is to partition the input into regions that contain "similar" points.

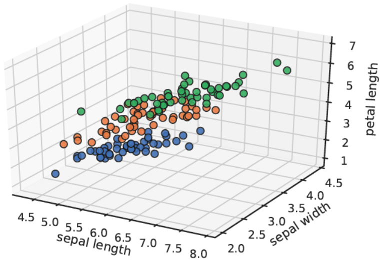
taken on 10.09.21, Gaza

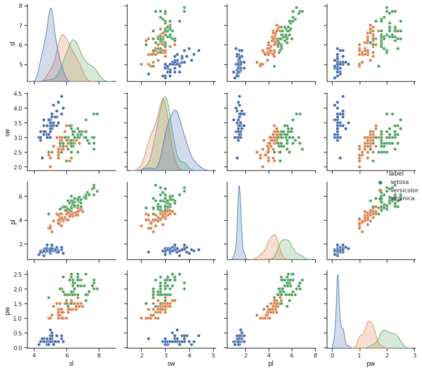
UNSUPERVISED LEARNING

Example: factors of variations

The process of projecting the high-dimensional data to a lower-dimensional supspace while capturing the “essence” of the data.

principal component analysis (PCA)



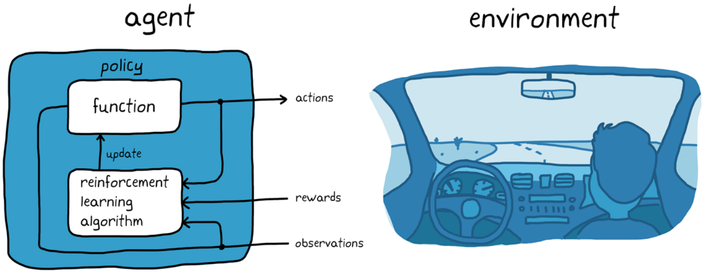


REINFORCEMENT LEARNING

REINFORCEMENT LEARNING

Definition

The system or agent has to learn how to interact with its environment. This can be encoded by means of a policy $a = \pi(x)$, which specifies which action to take in response to each possible input x.



Source: https://www.mathworks.com/discovery/reinforcement-learning.html

REINFORCEMENT LEARNING

DATA

TALL AND SKINNY VS. SHORT AND FAT

tall and skinny refers to the design matrix where $N >> D$, i.e., you have more examples than features.

short and fat refers to the design matrix where $D >> N$, i.e., you have more features than examples.

What about big data vs. wide data?

Questions