Computer Vision Group
Prof. Daniel Cremers

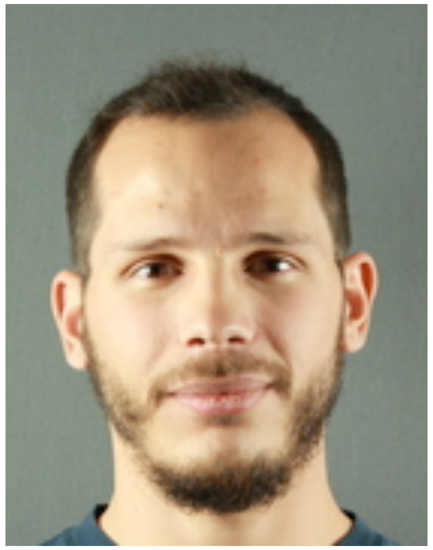
Technische Universität München

Machine Learning for Computer Vision

PD Dr. Rudolph Triebel

Lecturers

- PD Dr. Rudolph Triebel
- rudolph.triebel@in.tum.de
- Room number 02.09.058 (Fridays)
- Main lecture



- MSc. Ioannis “John” Chiotellis
- ioannis.chiotellis@in.tum.de
- Room number 02.09.058
- Assistance and exercises

- MSc. Maximilian Denninger
- maximilian.denninger@dlr.de
- Assistance and exercises

Lecturers

- PD Dr. Rudolph Triebel
- rudolph.triebel@in.tum.de
- Room number 02.09.058 (Fridays)
- Main lecture

Main affiliation (Mo - Thur):

Head of department for Perception and Cognition

Institute of Robotics and Mechatronics, DLR

rudolph.triebel@dlr.de

Class Webpage

https://vision.in.tum.de/teaching/ss2018/ml4cv

- Contains the slides and assignments for download
- Also used for communication, in addition to email list
- Some further material will be developed in class
- Material from earlier semesters also available
- Video lectures from an earlier semester on YouTube

Aim of this Class

- Give a major **overview** of the most important machine learning methods
- Present relations to **current research** applications for most learning methods
- Explain some of the more **basic** techniques in more detail, others in less detail
- Provide a **complement** to other machine learning classes

Prerequisites

Main background needed:

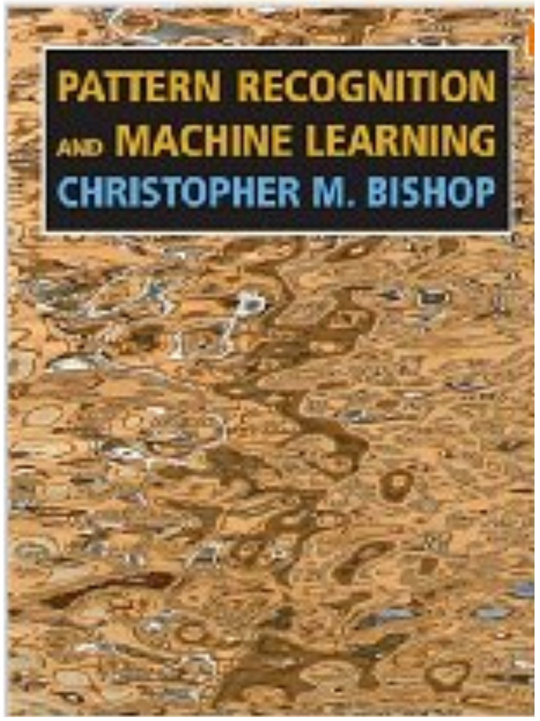
- Linear Algebra
- Calculus
- Probability Theory

There is a “Linear Algebra Refresher” on the web page!

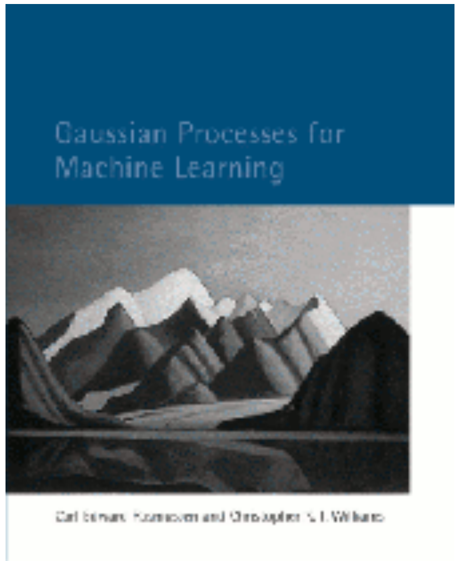
Topics Covered

- Introduction (today)
- Regression
- Graphical Models (directed and undirected)
- Clustering
- Boosting and Bagging
- Metric Learning
- Convolutional Neural Networks and Deep Learning
- Kernel Methods
- Gaussian Processes
- Learning of Sequential Data
- Sampling Methods
- Variational Inference
- Online Learning

Literature

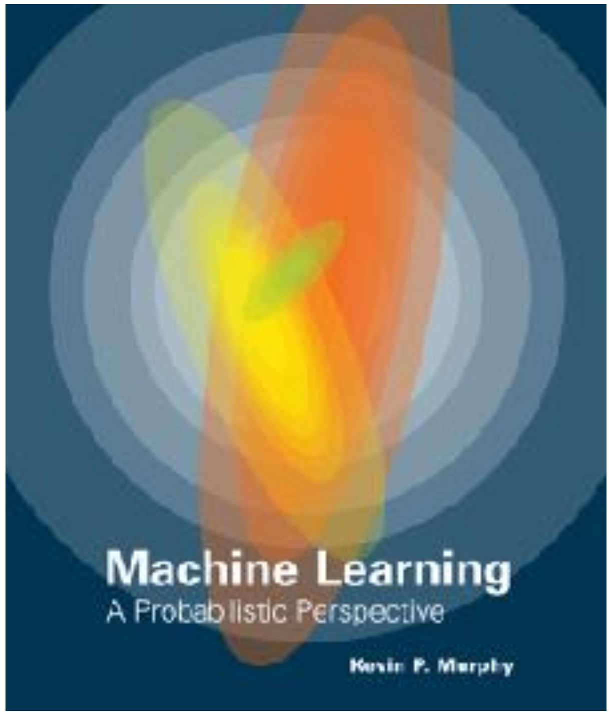


Recommended textbook for the lecture: Christopher M. Bishop: “Pattern Recognition and Machine Learning”



More detailed:

- “Gaussian Processes for Machine Learning” Rasmussen/Williams
- “Machine Learning - A Probabilistic Perspective” Murphy



The Tutorials

- **Weekly** tutorial classes
- Lecturers are alternating (John and Max)
- Participation in tutorial classes and submission of solved assignment sheets is **free**
- In class, you have the opportunity to present your solution
- Assignments will be theoretical and practical problems (in Python)
- Software library: https://github.com/johny-c/mlcv-tutorial
- First tutorial class: April 19

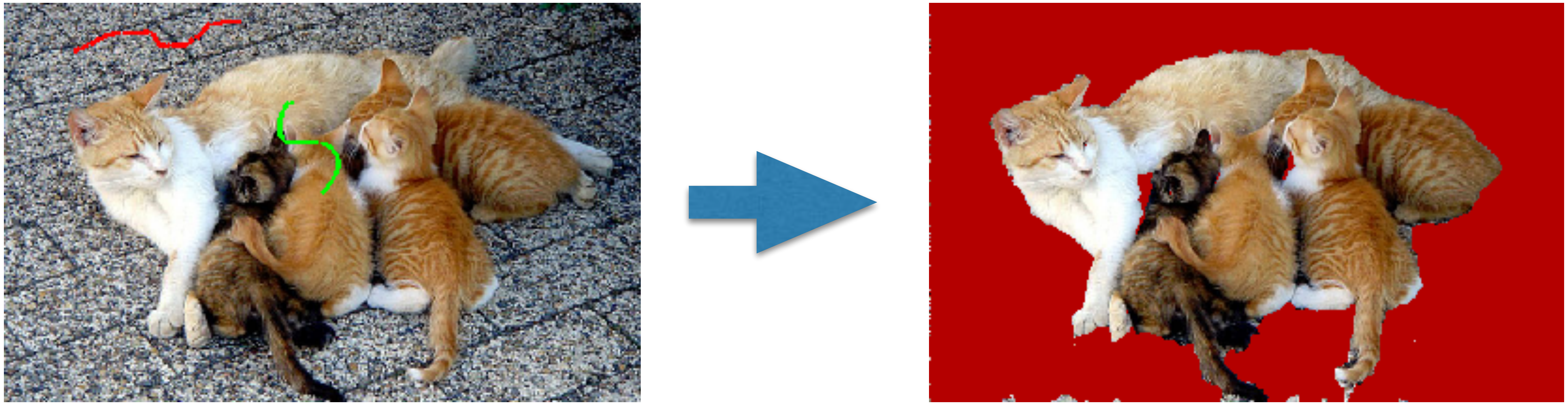
The Exam

- No “qualification” necessary for the final exam
- It will be a **written** exam
- So far, the date is not fixed yet, it will be announced within the next weeks
- In the exam, there will be more assignments than needed to reach the highest grade

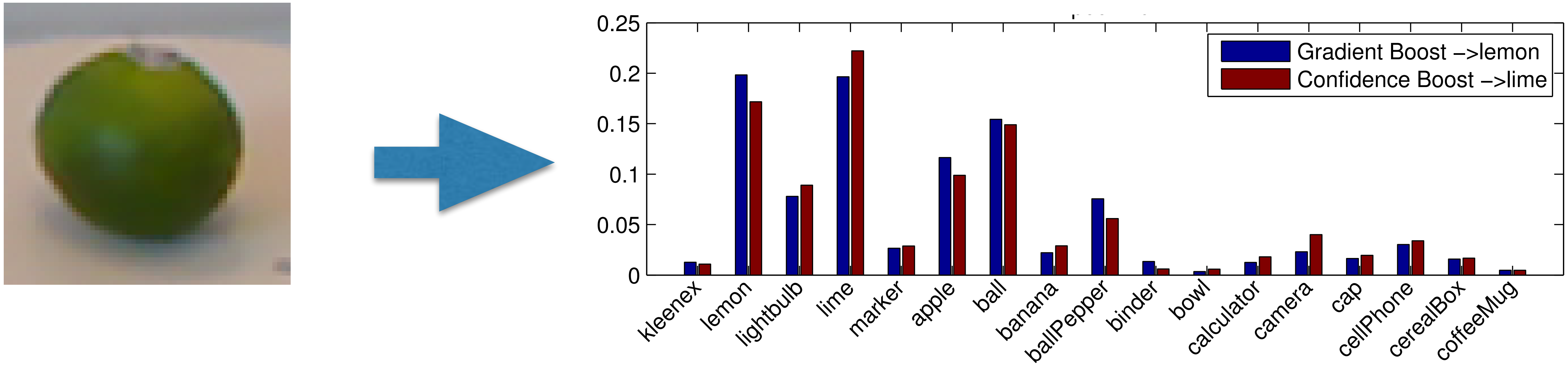
Why Machine Learning?

Typical Problems in Computer Vision

Image Segmentation

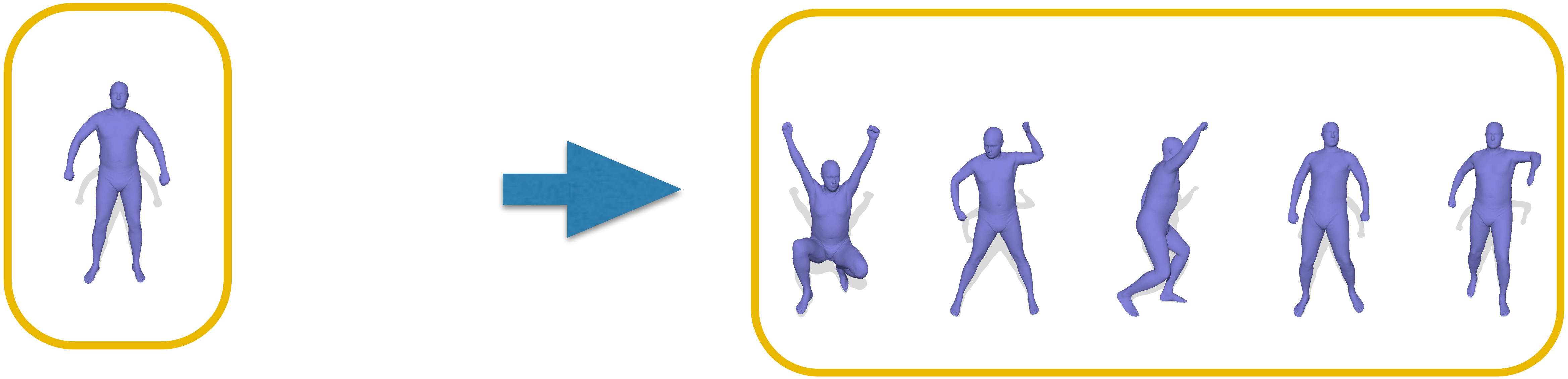


Object Classification

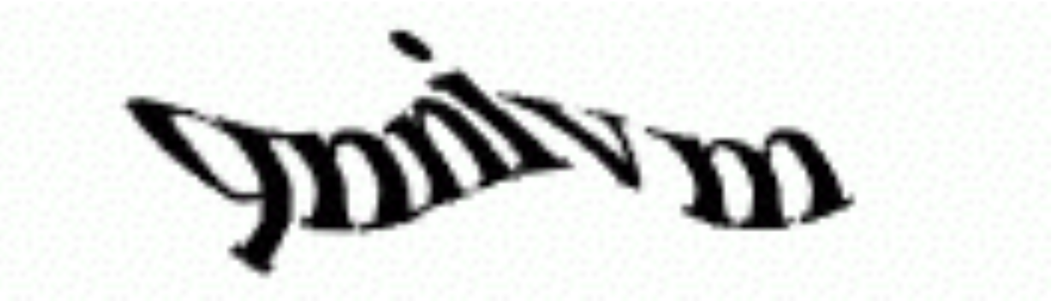


Typical Problems in Computer Vision

3D Shape Analysis, e.g. Shape Retrieval



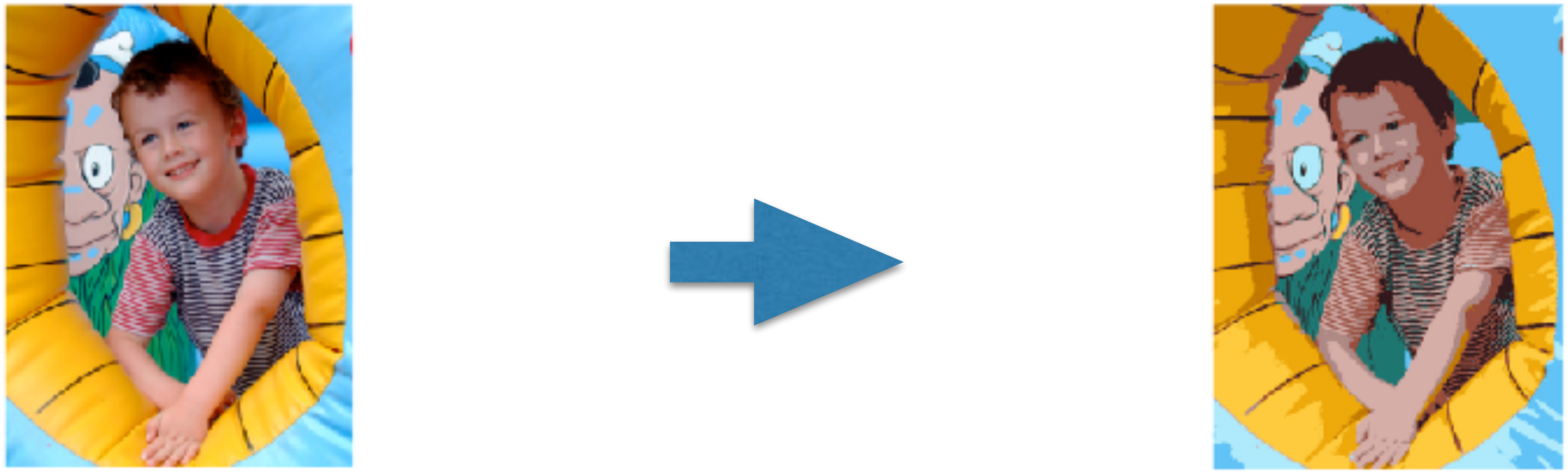
Optical Character Recognition



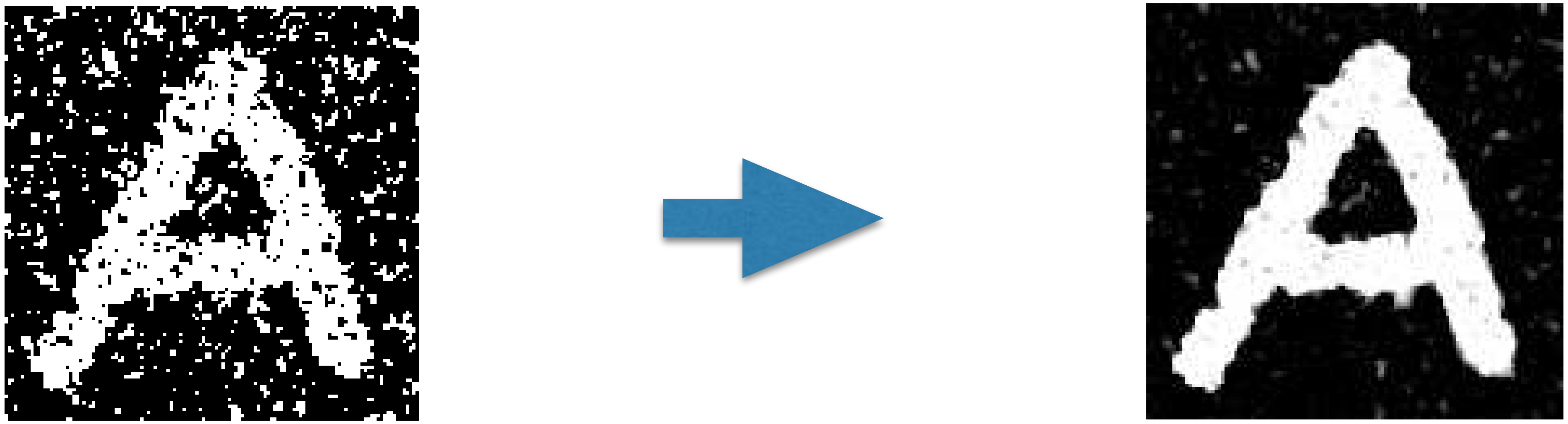
“qnnivm”

Typical Problems in Computer Vision

Image compression



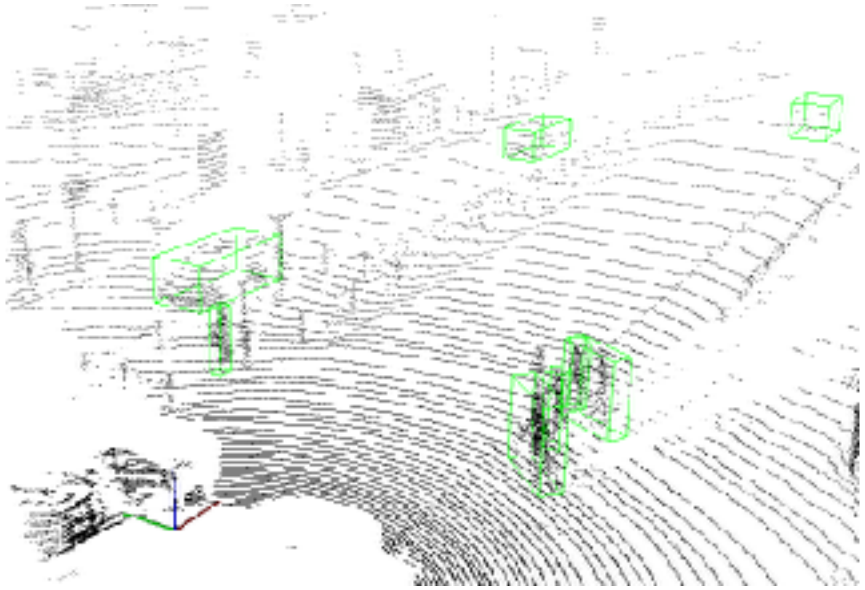
Noise reduction



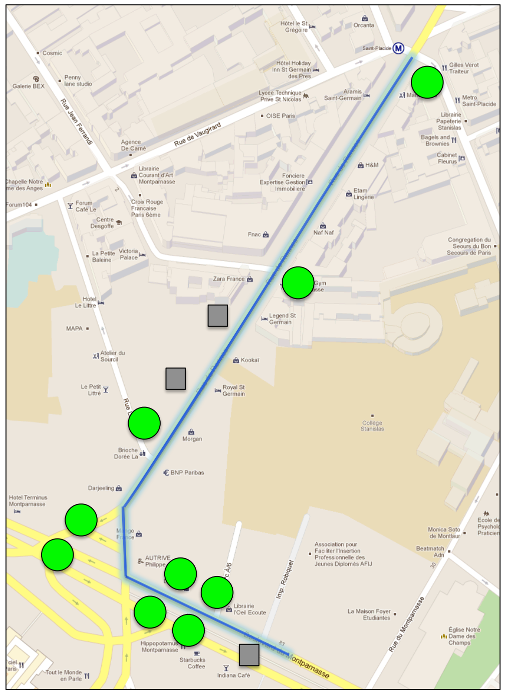
… and many others, e.g.: optical flow, scene flow, 3D reconstruction, stereo matching, …

Some Applications in Robotics

Detection of cars and pedestrians for autonomous cars



Semantic Mapping



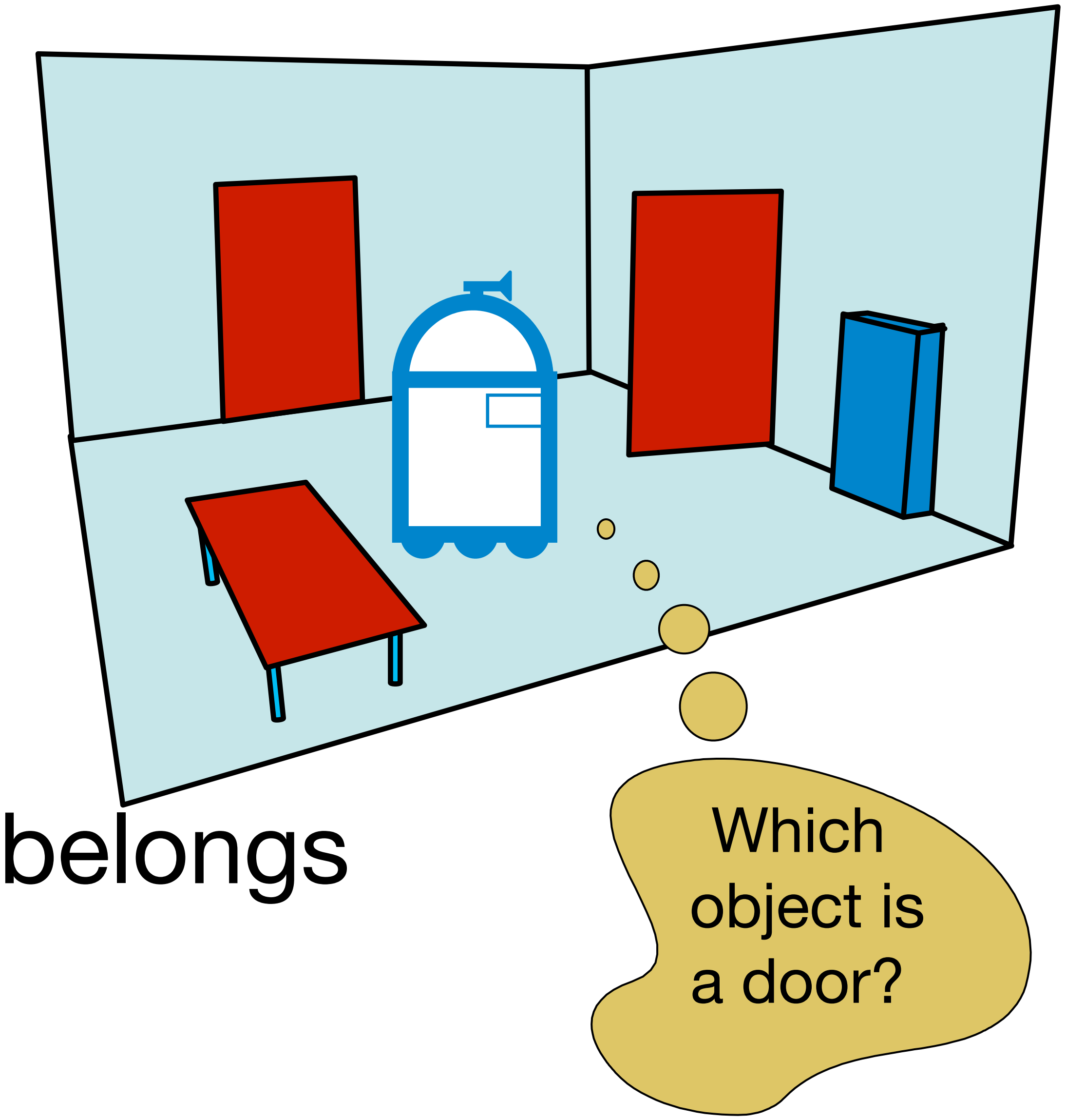
What Makes These Problems Hard?

- It is very hard to express the relation from input to output with a mathematical model.
- Even if there was such a model, how should the parameters be set?
- A hand-crafted model is **not general** enough, it can not be used again in similar applications
- There is often no one-to-one mapping from input to output

Idea: extract the needed information from a data set of input - output pairs by optimizing an objective function

Example Application of Learning in Robotics

- Most objects in the environment can be classified, e.g. with respect to their size, functionality, dynamic properties, etc.
- Robots need to ***interact*** with the objects (move around, manipulate, inspect, etc.) and with humans
- For all these tasks it is necessary that the robot knows to which ***class*** an object belongs



Learning = Optimization

- A natural way to do object classification is to first find a mapping from input data to object labels (“**learning**”) and then **infer** from the learned data a possible class for a new object.
- The area of **machine learning** deals with the formulation and investigates methods to do the learning automatically.
- It is essentially based on **optimization** methods
- Machine learning algorithms are widely used in robotics and computer vision

Mathematical Formulation

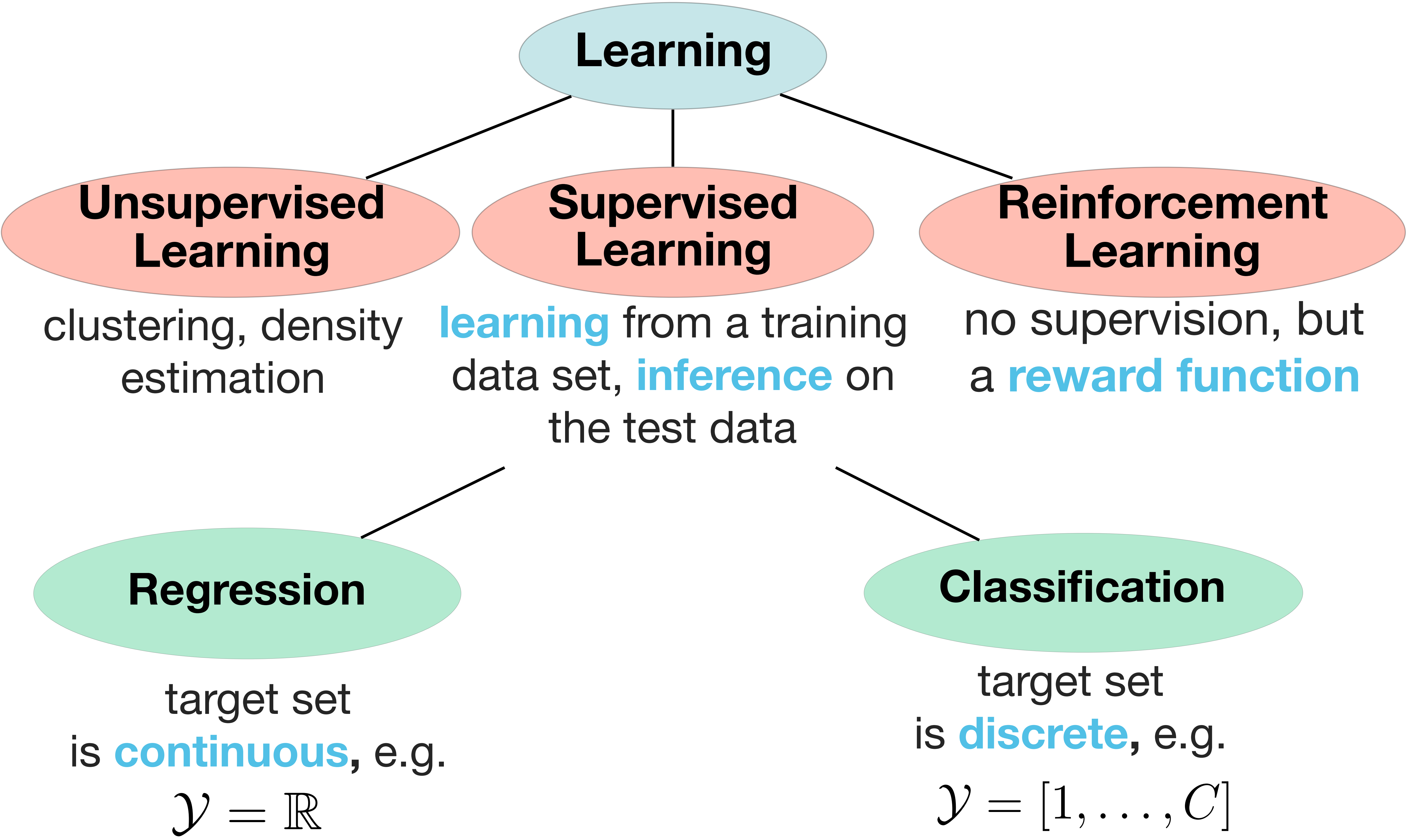
Suppose we are given a set $\mathcal{X}$ of objects and a set $\mathcal{Y}$ of object categories (classes). In the learning task we search for a mapping $\varphi : \mathcal{X} \rightarrow \mathcal{Y}$ such that **similar** elements in $\mathcal{X}$ are mapped to **similar** elements in $\mathcal{Y}$.

Examples:

- Object classification: chairs, tables, etc.
- Optical character recognition
- Speech recognition

Important problem: Measure of similarity!

Categories of Learning



Learning

Unsupervised Learning

clustering, density estimation

Supervised Learning

learning from a training data set, **inference** on the test data

Reinforcement Learning

no supervision, but a **reward function**

Regression

target set is **continuous**, e.g.

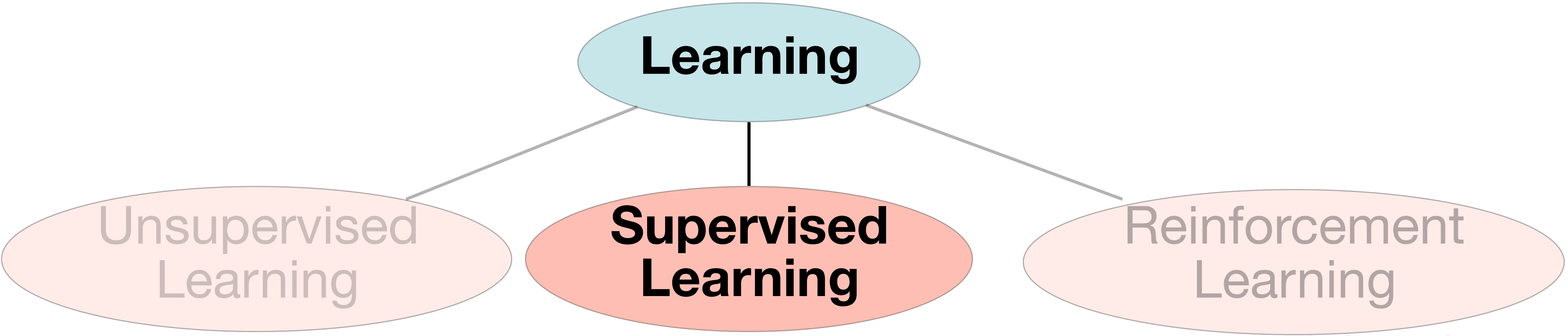
$$\mathcal{Y} = \mathbb{R}$$

Classification

target set is **discrete**, e.g.

$$\mathcal{Y} = [1, \ldots, C]$$

Categories of Learning



Learning

Unsupervised Learning

Supervised Learning

Reinforcement Learning

clustering, density estimation

learning from a training data set, inference on the test data

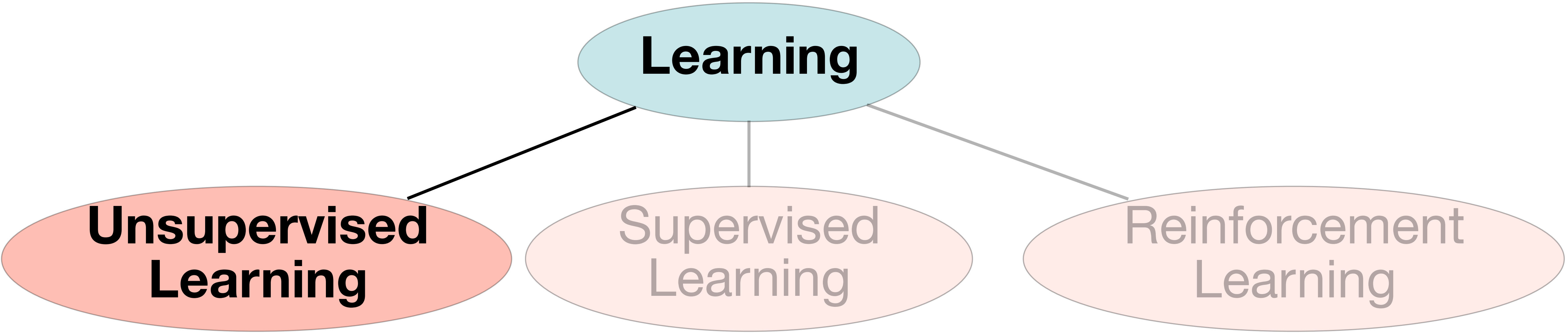
no supervision, but a reward function

Supervised Learning is the main topic of this lecture!

Methods used in Computer Vision include:

- Regression
- Conditional Random Fields
- Boosting
- Deep Neural Networks
- Gaussian Processes
- Hidden Markov Models

Categories of Learning



Learning

Unsupervised Learning

clustering, density estimation

Supervised Learning

learning from a training data set, inference on the test data

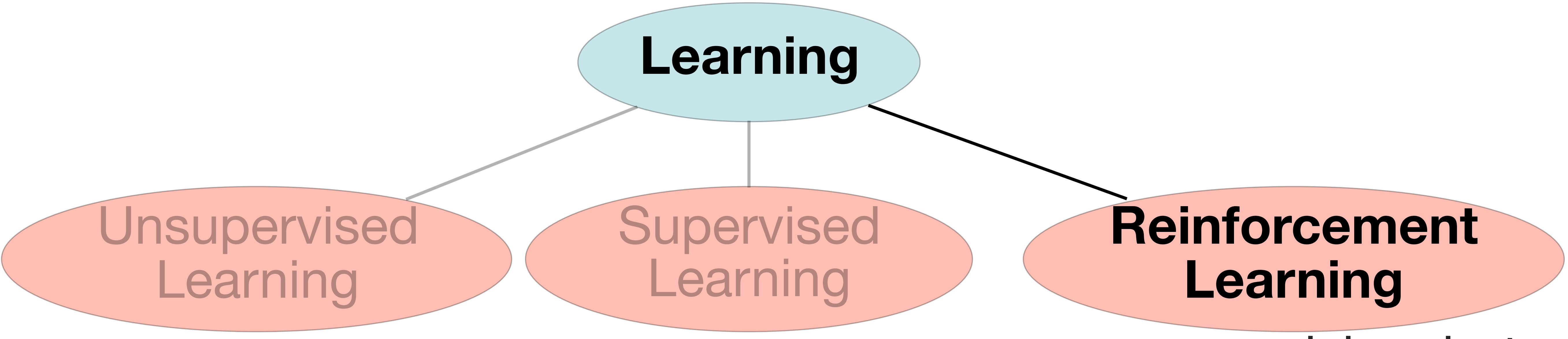
Reinforcement Learning

no supervision, but a reward function

In unsupervised learning, there is no **ground truth** information given.

Most Unsupervised Learning methods are based on **Clustering**.

Categories of Learning



Learning

Unsupervised Learning

clustering, density estimation

Supervised Learning

learning from a training data set, inference on the test data

Reinforcement Learning

no supervision, but a **reward function**

Reinforcement Learning requires an ***action***

- the reward defines the quality of an action
- mostly used in robotics (e.g. manipulation)
- can be dangerous, actions need to be “tried out”
- not handled in this course

Categories of Learning

Further distinctions are:

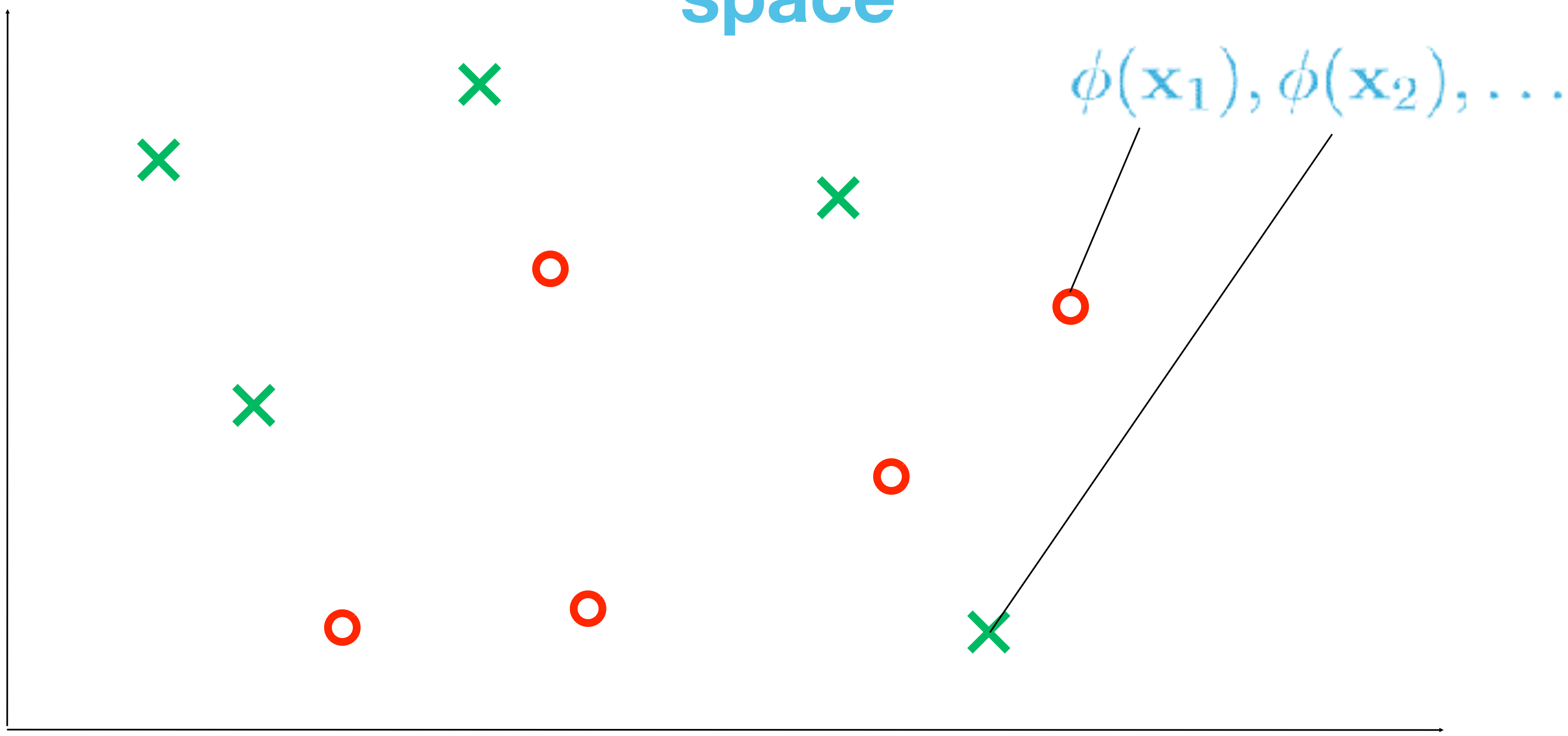
- **online** vs **offline** learning (both for supervised and unsupervised methods)
- **semi-supervised** learning (a combination of supervised and unsupervised learning)
- multiple instance / single instance learning
- multi-task / single-task learning
- …

Generative Model: Example

Nearest-neighbor classification:

- Given: data points $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots$
- Rule: Each new data point is assigned to the class of its nearest neighbor in feature space

1. **Training instances in feature space**

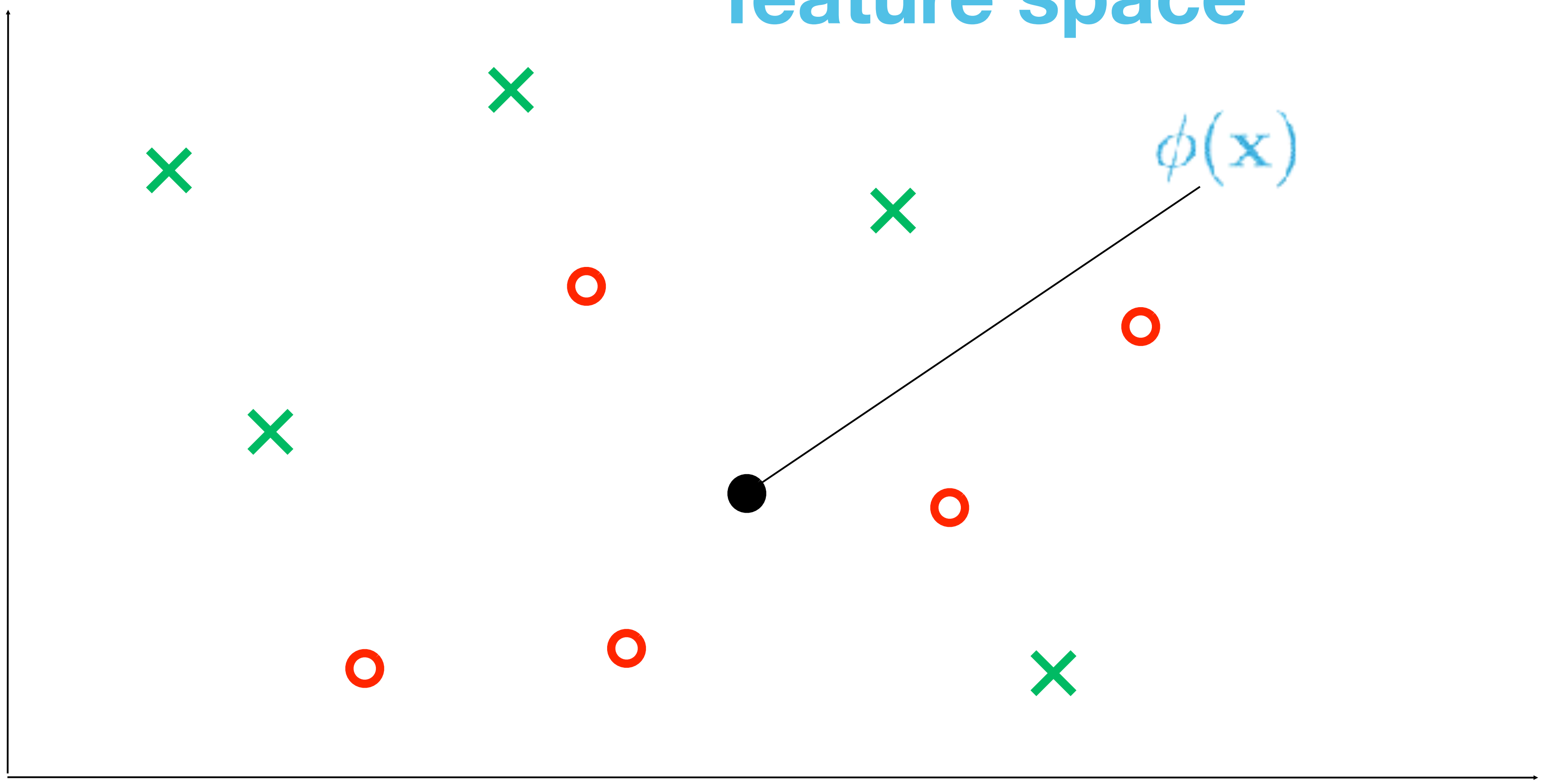


Generative Model: Example

Nearest-neighbor classification:

- Given: data points $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots$
- Rule: Each new data point is assigned to the class of its nearest neighbor in feature space

2. Map new data point into feature space

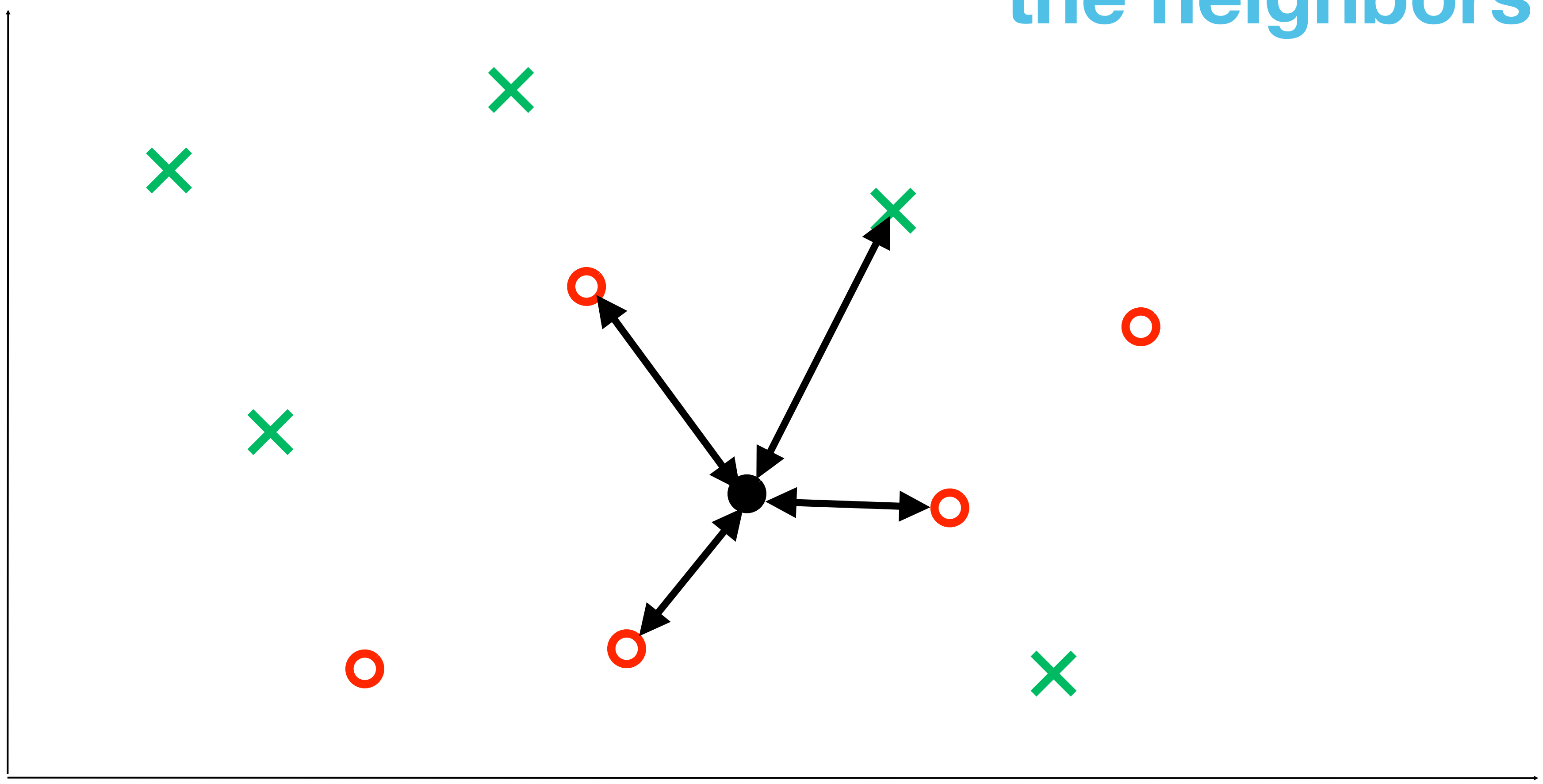


Generative Model: Example

Nearest-neighbor classification:

- Given: data points $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots$
- Rule: Each new data point is assigned to the class of its nearest neighbor in feature space

3. Compute the distances to the neighbors

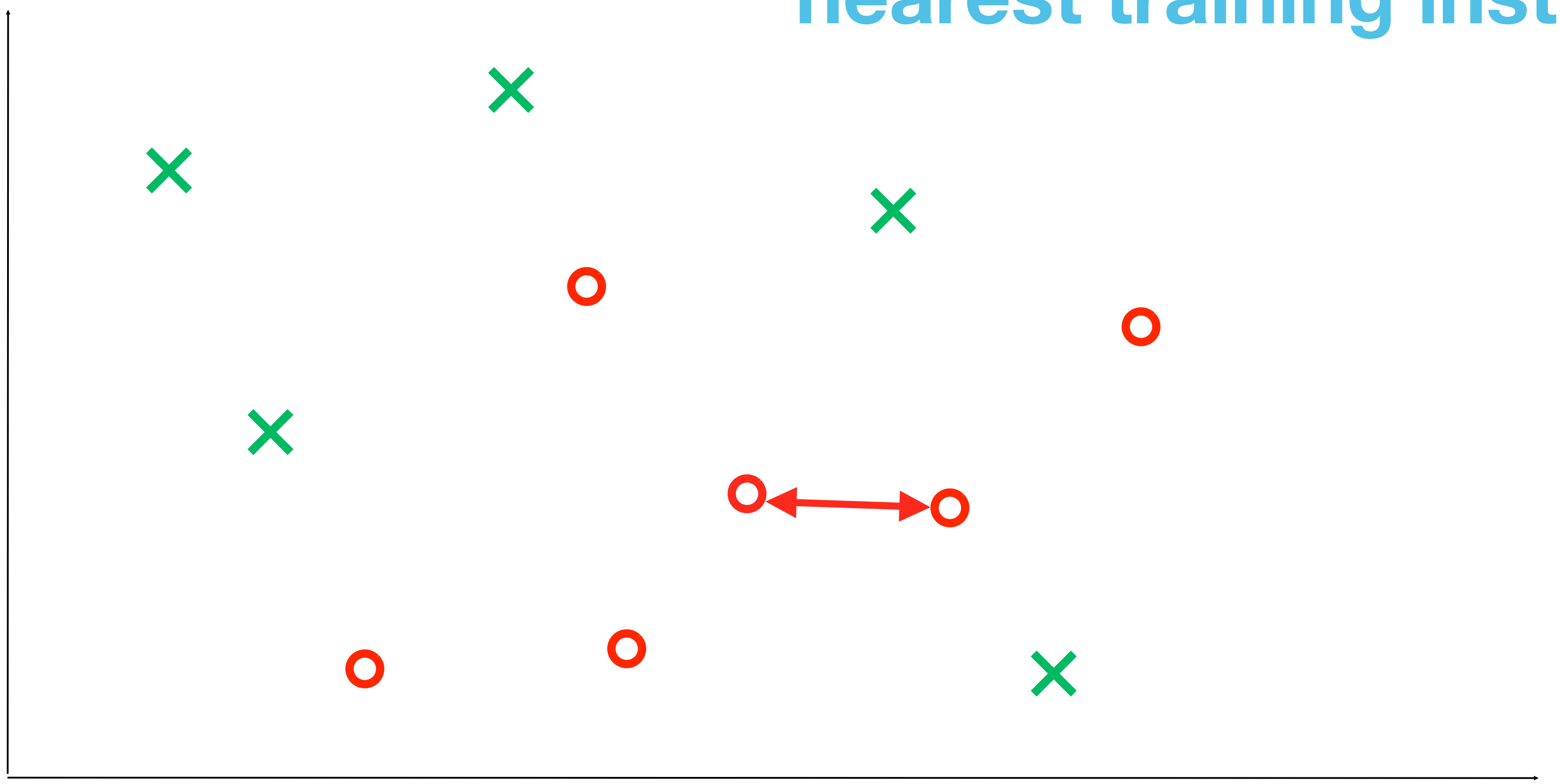


Generative Model: Example

Nearest-neighbor classification:

- Given: data points $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots$
- Rule: Each new data point is assigned to the class of its nearest neighbor in feature space

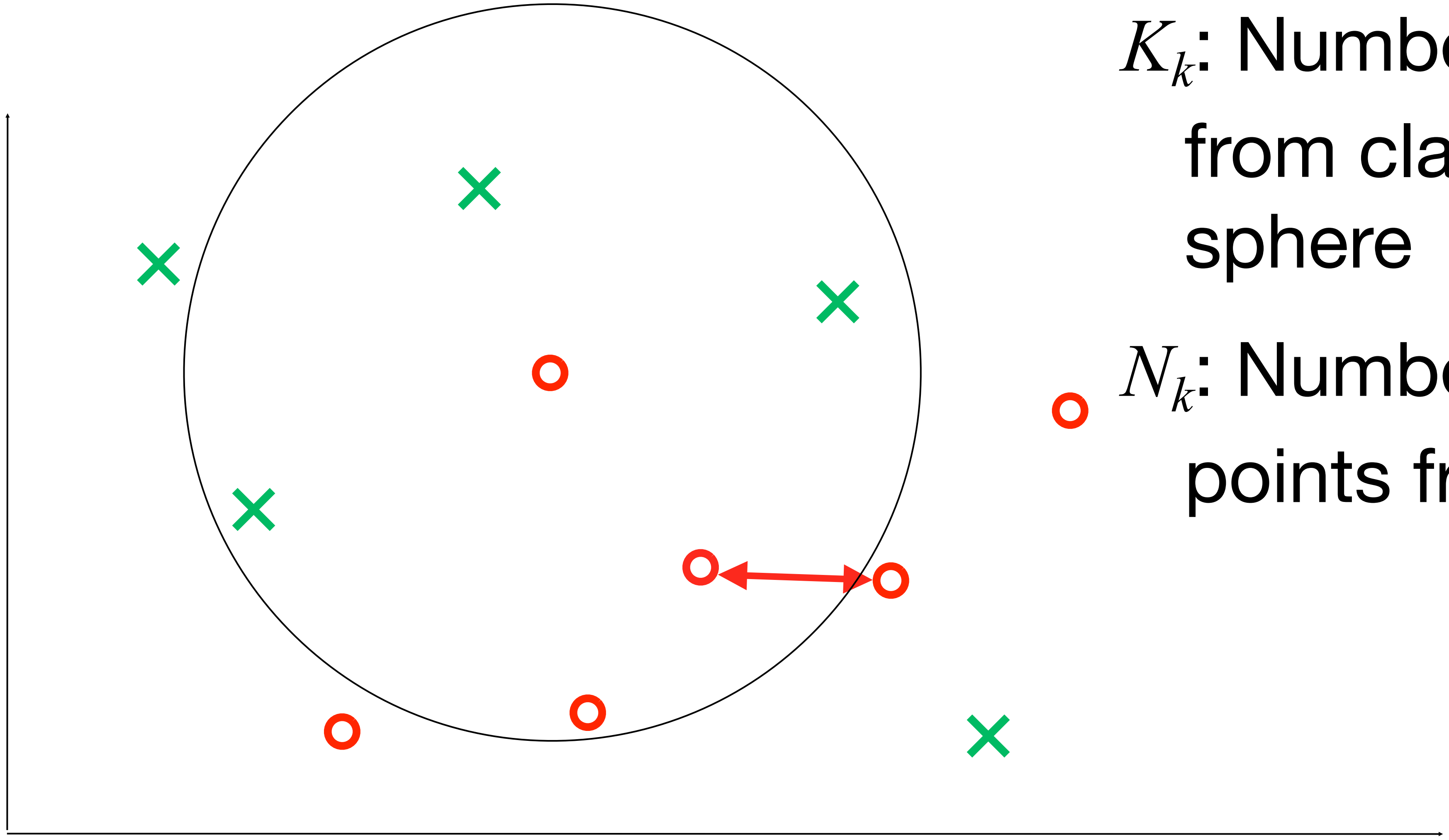
4. Assign the label of the nearest training instance



Generative Model: Example

Nearest-neighbor classification:

- General case: K nearest neighbors
- We consider a sphere around each training instance that has a fixed volume V.



K_k: Number of points from class k inside sphere

N_k: Number of all points from class k

Generative Model: Example

Nearest-neighbor classification:

- General case: K nearest neighbors
- We consider a sphere around a training / test sample that has a fixed volume V.
- With this we can estimate:

$$p(\mathbf{x} \mid y = k) = \frac{K_k}{N_k V}$$

“likelihood”

- and likewise:

$$p(\mathbf{x}) = \frac{\overset{\text{\# points in sphere}}{K}}{\underset{\text{\# all points}}{NV}}$$

“uncond. prob.”

- using Bayes rule:

$$p(y = k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = k)p(y = k)}{p(\mathbf{x})} = \frac{K_k}{K}$$

“posterior”

Generative Model: Example

Nearest-neighbor classification:

- General case: K nearest neighbors

$$p(y = k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = k)p(y = k)}{p(\mathbf{x})} = \frac{K_k}{K}$$

- To classify the new data point $\mathbf{x}$ we compute the posterior for each class k = 1,2,… and assign the label that **maximizes the posterior (MAP)**.

$$t := \arg\max_k p(y = k \mid \mathbf{x})$$

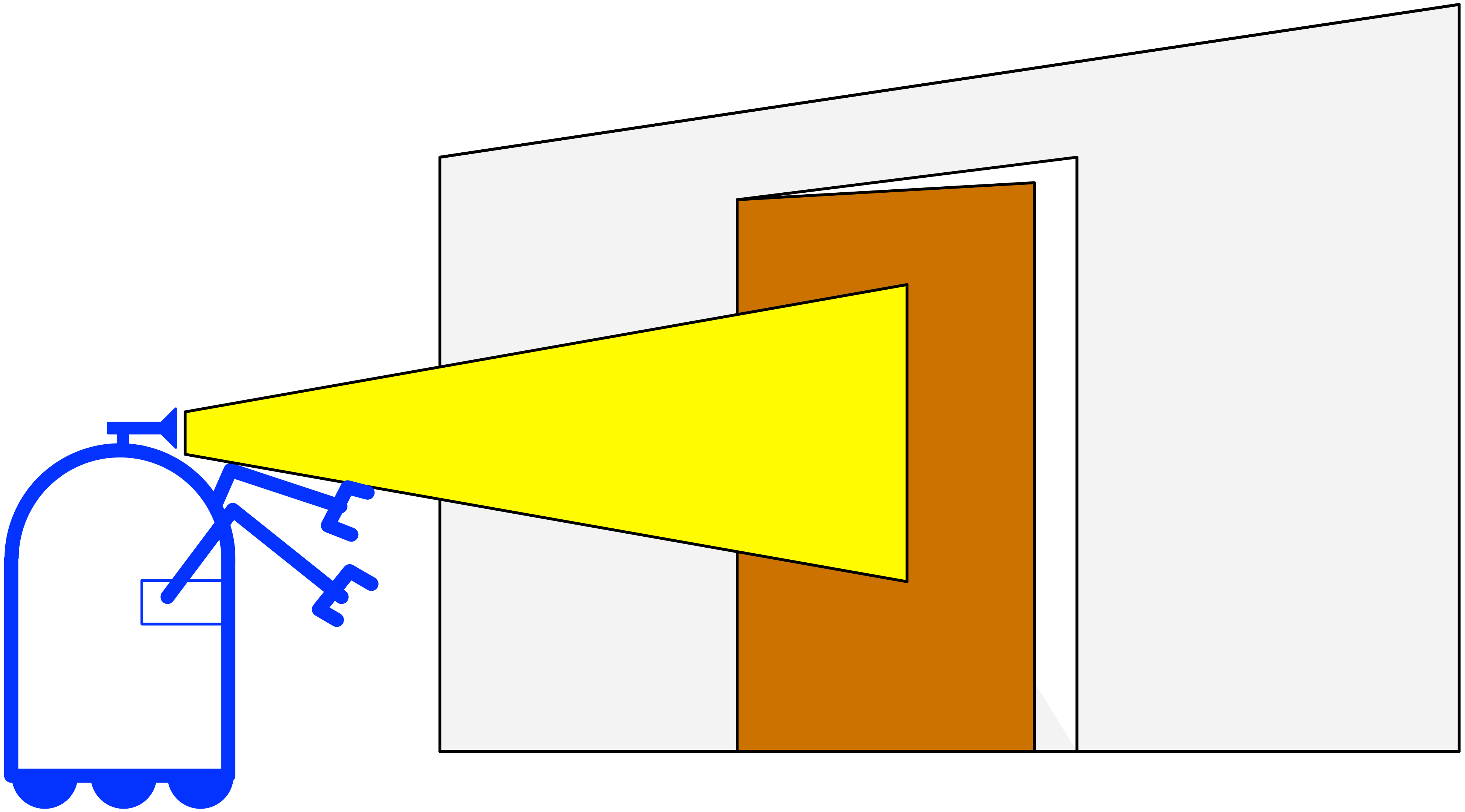
Summary

- Learning is usually a two-step process consisting in a ***training*** and an ***inference*** step
- Learning is useful to extract ***semantic*** information, e.g. about the objects in an environment
- There are three main categories of learning: ***unsupervised***, ***supervised*** and ***reinforcement*** learning
- Supervised learning can be split into ***regression***, and ***classification***
- An example for a generative model is ***nearest neighbor classification***

Introduction to Probabilistic Reasoning

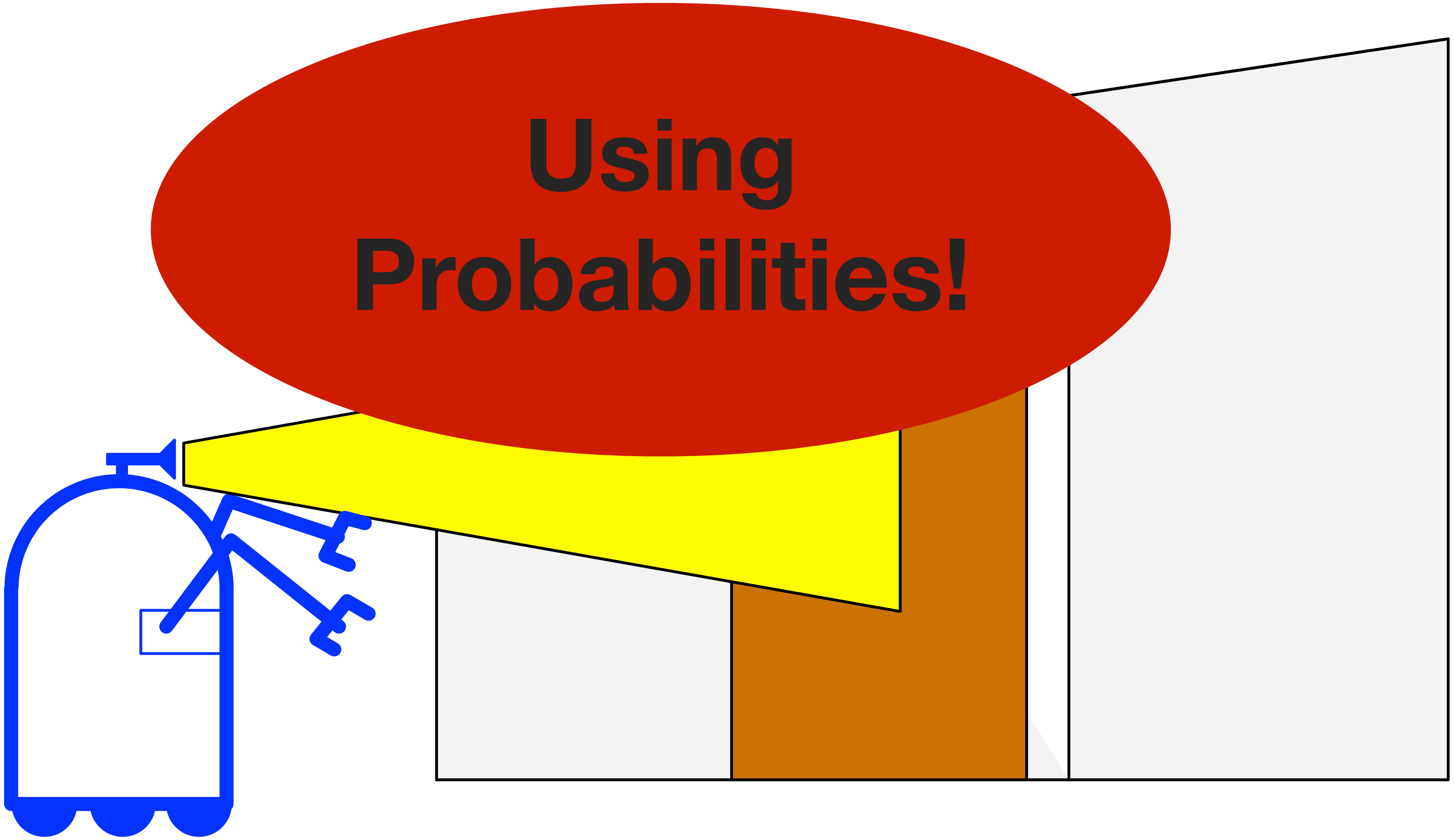
Motivation

Suppose a robot stops in front of a door. It has a sensor (e.g. a camera) to measure the state of the door (open or closed). **Problem**: the sensor may fail.



Motivation

Question: How can we obtain knowledge about the environment from sensors that may return incorrect results?



Using Probabilities!

Basics of Probability Theory

Definition 1.1: A *sample space* $\mathcal{S}$ is a set of outcomes of a given experiment.

Examples:

a) Coin toss experiment: $\mathcal{S} = \{H, T\}$

b) Distance measurement: $\mathcal{S} = \mathbb{R}_0^+$

Definition 1.2: A *random variable* X is a function that assigns a real number to each element of $\mathcal{S}$.

Example: Coin toss experiment: $H = 1, T = 0$

Values of random variables are denoted with small letters, e.g.: $X = x$

Discrete and Continuous

If $\mathcal{S}$ is countable then X is a ***discrete*** random variable, else it is a ***continuous*** random variable.

The probability that X takes on a certain value x is a real number between 0 and 1. It holds:

$$\sum_x p(X = x) = 1$$

Discrete case

$$\int p(X = x) dx = 1$$

Continuous case

A Discrete Random Variable

Suppose a robot knows that it is in a room, but it does not know in ***which*** room. There are 4 possibilities:

Kitchen, Office, Bathroom, Living room

Then the random variable *Room* is discrete, because it can take on one of four values. The probabilities are, for example:

$$P(Room = \text{kitchen}) = 0.7$$

$$P(Room = \text{office}) = 0.2$$

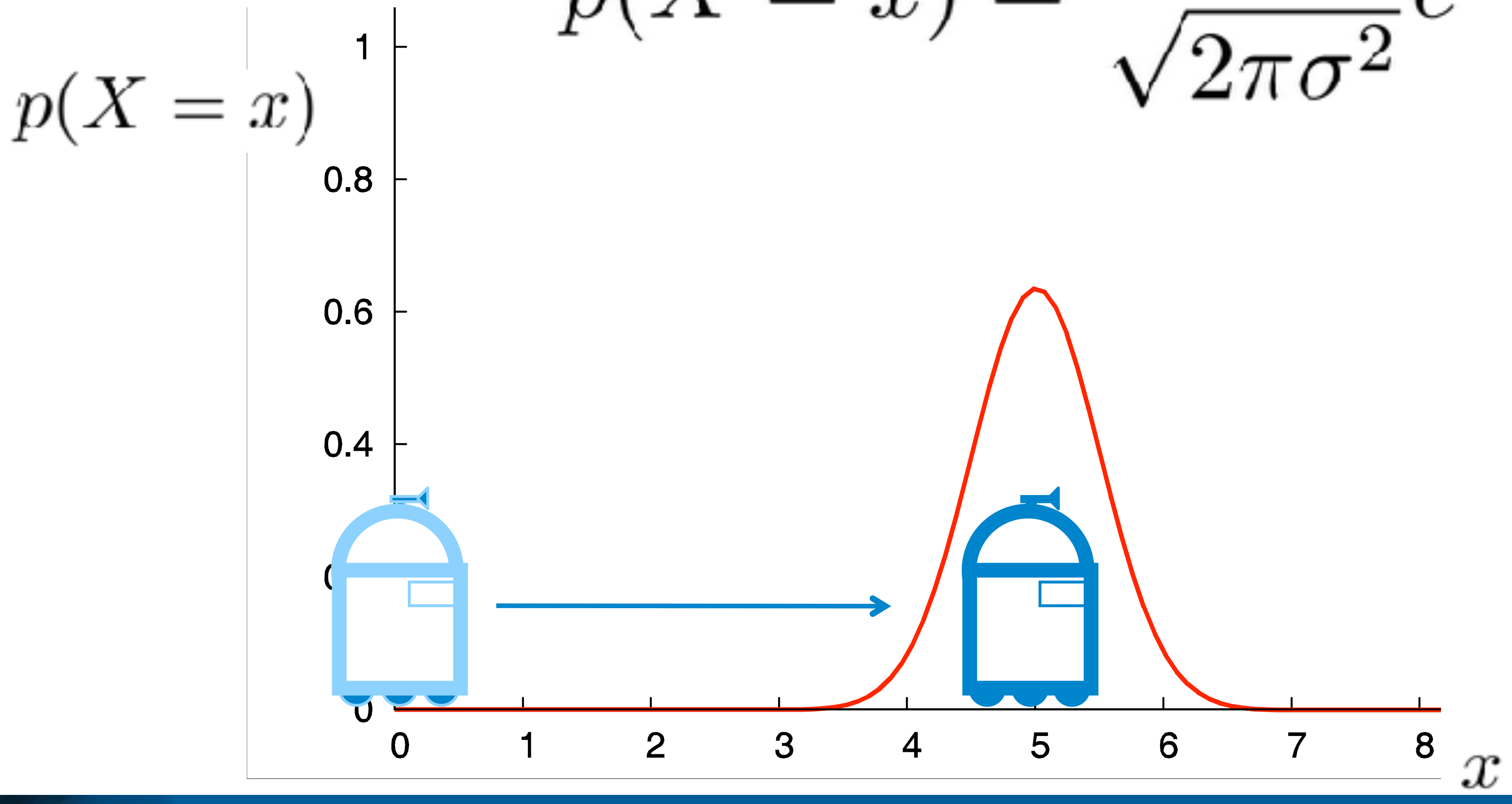
$$P(Room = \text{bathroom}) = 0.08$$

$$P(Room = \text{living room}) = 0.02$$

A Continuous Random Variable

Suppose a robot travels 5 meters forward from a given start point. Its position X is a continuous random variable with a ***Normal distribution***:

$$p(X=x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}\frac{(x-5)^2}{\sigma^2}}$$



Shorthand:

$$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

$$\downarrow$$

$$\mathcal{N}(x;\mu,\sigma^2)$$

Joint and Conditional Probability

The ***joint probability*** of two random variables X and Y is the probability that the events $X = x$ and $Y = y$ occur at the same time:

$$p(X = x \text{ and } Y = y)$$

Shorthand:

$$p(X = x) \longrightarrow p(x)$$

$$p(X = x \text{ and } Y = y) \longrightarrow p(x, y)$$

Definition 1.3: The ***conditional probability*** of X given Y is defined as:

$$p(X = x \mid Y = y) = p(x \mid y) := \frac{p(x, y)}{p(y)}$$

Independency, Sum and Product Rule

Definition 1.4: Two random variables X and Y are *independent* iff:

$$p(x, y) = p(x)p(y)$$

For independent random variables Xnd Ye have:

$$p(x \mid y) = \frac{p(x, y)}{p(y)} = \frac{p(x)p(y)}{p(y)} = p(x)$$

Furthermore, it holds:

$$p(x) = \sum_y p(x, y) \qquad p(x, y) = p(y \mid x)p(x)$$

“Sum Rule” **“Product Rule”**

Law of Total Probability

Theorem 1.1: For two random variables X and Y it holds:

$$p(x) = \sum_y p(x \mid y) p(y) \qquad p(x) = \int p(x \mid y) p(y) dy$$

Discrete case

Continuous case

The process of obtaining $p(x)$ from $p(x, y)$ by summing or integrating over all values of y is called

Marginalisation

Bayes Rule

Theorem 1.2: For two random variables X and Y it holds:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}$$

“Bayes Rule”

Proof:

I. $$p(x \mid y) = \frac{p(x, y)}{p(y)}$$ *(definition)*

II. $$p(y \mid x) = \frac{p(x, y)}{p(x)}$$ *(definition)*

III. $$p(x, y) = p(y \mid x)p(x)$$ *(from II.)*

Bayes Rule: Background Knowledge

For $p(y \mid z) \neq 0$ it holds:

Background knowledge

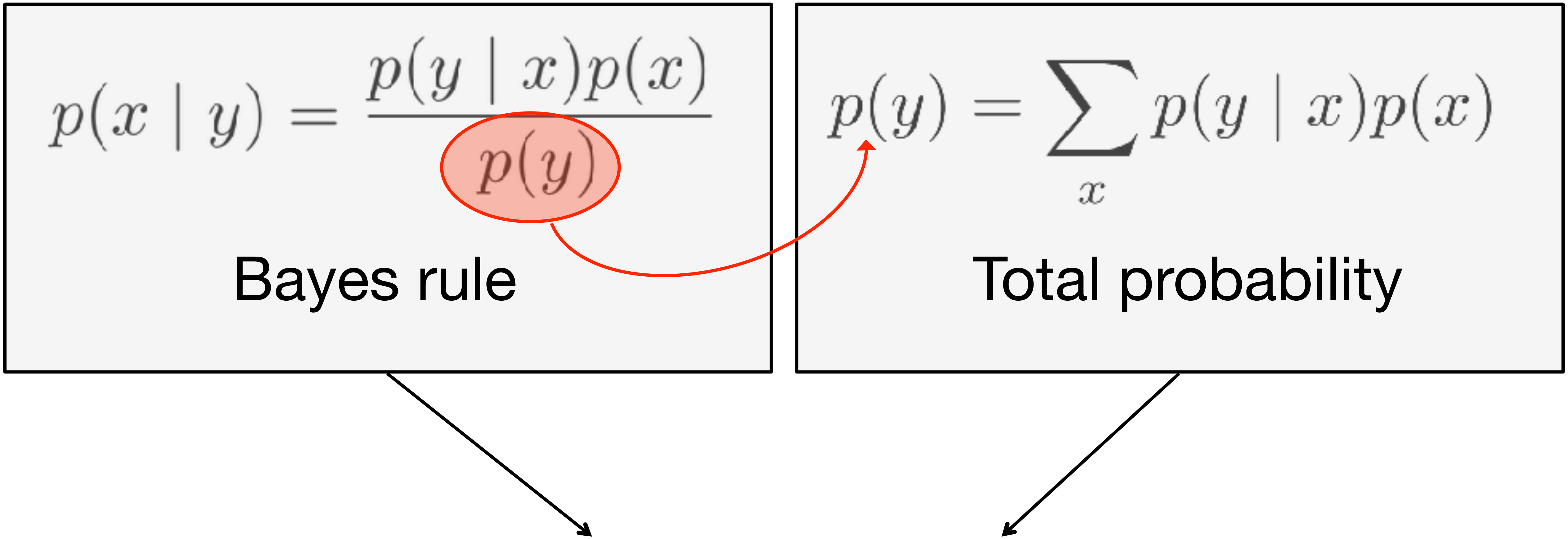
$$p(x \mid y, z) = \frac{p(y \mid x, z)p(x \mid z)}{p(y \mid z)}$$

Shorthand: $p(y \mid z)^{-1} \longrightarrow \eta$

“Normalizer”

$$p(x \mid y, z) = \eta \, p(y \mid x, z)p(x \mid z)$$

Computing the Normalizer



$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}$$

Bayes rule

$$p(y) = \sum_x p(y \mid x)p(x)$$

Total probability

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\sum_{x'} p(y \mid x')p(x')}$$

$p(x \mid y)$ can be computed without knowing $p(y)$

Conditional Independence

Definition 1.5: Two random variables X and Y are *conditional independent* given a third random variable Z iff:

$$p(x, y \mid z) = p(x \mid z)p(y \mid z)$$

This is equivalent to:

$$p(x \mid z) = p(x \mid y, z)$$ and

$$p(y \mid z) = p(y \mid x, z)$$

Expectation and Covariance

Definition 1.6: The ***expectation*** of a random variable X is defined as:

$$E[X] = \sum_x x\, p(x)$$ (discrete case)

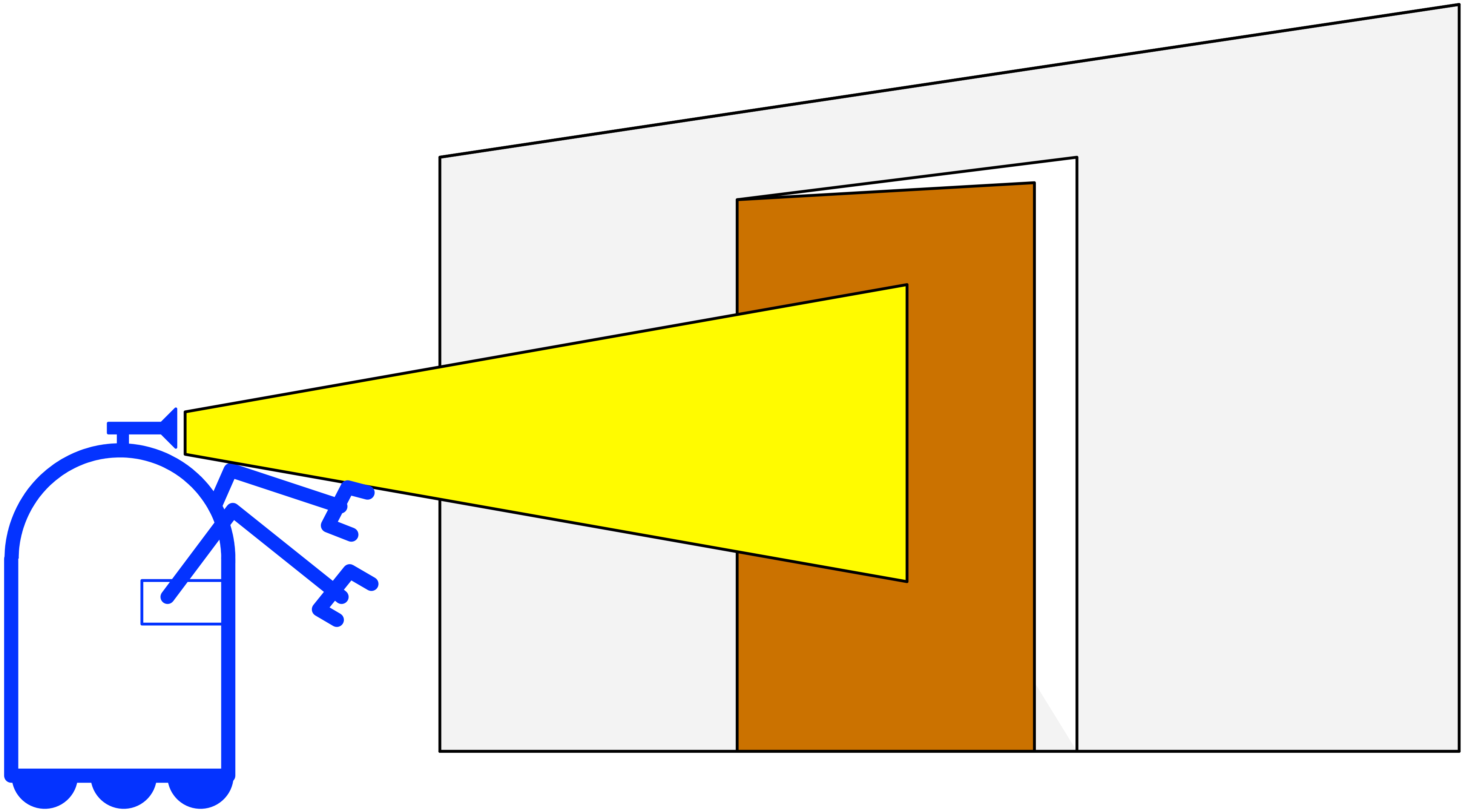
$$E[X] = \int x\, p(x) dx$$ (continuous case)

Definition 1.7: The ***covariance*** of a random variable X is defined as:

$$Cov[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

Mathematical Formulation of Our Example

We define two binary random variables: z and open, where z is “light on” or “light off”. Our question is: What is $p(\mathrm{open} \mid z)$?



Causal vs. Diagnostic Reasoning

- Searching for $p(\text{open} \mid z)$ is called ***diagnostic reasoning***
- Searching for $p(z \mid \text{open})$ is called ***causal reasoning***
- Often causal knowledge is easier to obtain
- Bayes rule allows us to use causal knowledge:

$$
\begin{aligned}
p(\text{open} \mid z) &= \frac{p(z \mid \text{open})p(\text{open})}{p(z)} \\
&= \frac{p(z \mid \text{open})p(\text{open})}{p(z \mid \text{open})p(\text{open}) + p(z \mid \neg\text{open})p(\neg\text{open})}
\end{aligned}
$$

Example with Numbers

Assume we have this ***sensor model:***

$$p(z \mid \text{open}) = 0.6 \qquad p(z \mid \neg\text{open}) = 0.3$$

and: $p(\text{open}) = p(\neg\text{open}) = 0.5$ ***"Prior prob."***

then:

$$\begin{aligned} p(\text{open} \mid z) &= \frac{p(z \mid \text{open})p(\text{open})}{p(z \mid \text{open})p(\text{open}) + p(z \mid \neg\text{open})p(\neg\text{open})} \\ &= \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67 \end{aligned}$$

"z raises the probability that the door is open"

Combining Evidence

Suppose our robot obtains another observation z_2 , where the index is the point in time.

Question: How can we integrate this new information?

Formally, we want to estimate $p(\text{open} \mid z_1, z_2)$. Using Bayes formula with background knowledge:

$$p(\text{open} \mid z_1, z_2) = \frac{\overset{?\qquad\qquad ?}{p(z_2 \mid \text{open}, z_1)}\, p(\text{open} \mid z_1)}{p(z_2 \mid z_1)}$$

Markov Assumption

“If we know the state of the door at time $t = 1$ then the measurement z_1 does not give any further information about z_2.”

Formally: “z_1 and z_2 are conditional independent given open.“ This means:

$$p(z_2 \mid \mathrm{open}, z_1) = p(z_2 \mid \mathrm{open})$$

This is called the ***Markov Assumption.***

Example with Numbers

Assume we have a second sensor:

$$p(z_2 \mid \text{open}) = 0.5 \qquad p(z_2 \mid \neg\text{open}) = 0.6$$

$$p(\text{open} \mid z_1) = \frac{2}{3}$$ (from above)

Then: $p(\text{open} \mid z_1, z_2) =$

$$\frac{p(z_2 \mid \text{open})p(\text{open} \mid z_1)}{p(z_2 \mid \text{open})p(\text{open} \mid z_1) + p(z_2 \mid \neg\text{open})p(\neg\text{open} \mid z_1)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

“z_2 lowers the probability that the door is open”