Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Institute for AI and Autonomous Systems

Robot Learning

Introduction

Dr. Alex Mitrevski
Master of Autonomous Systems

Structure

Journal of Machine Learning Research 22 (2021) 1-82 Submitted 9/19; Revised 8/20; Published 1/21

A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms

Oliver Kroemer* Scott Niekum George Konidaris

Reinforcement learning in robotics: A survey

Jens Kober[1,2], J. Andrew Bagnell[3] and Jan Peters[4,5]

The International Journal of Robotics Research
32(11) 1238–1274

DOI: 10.1177/0278364913495721
ijr.sagepub.com
SAGE

Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning

Annual Review of Control, Robotics, and Autonomous Systems
Vol. 5:411-444 (Volume publication date May 2022)
First published as a Review in Advance on January 26, 2022
https://doi.org/10.1146/annurev-control-042920-020211

Lukas Brunke,[1,2,3,*] Melissa Greeff,[1,2,3,*] Adam W. Hall,[1,2,3,*] Zhaocong Yuan,[1,2,3,*] Siqi Zhou,[1,2,3,*] Jacopo Panerati,[1,2,3] and Angela P. Schoellig[1,2,3]

- Robot learning idea
- Robot learning techniques

Robot Learning Idea and Concepts

Why Robot Learning?

- Traditionally, robotics problems have been solved **using carefully crafted analytical models** (e.g. odometry models for estimating the motion of a mobile base)

Why Robot Learning?

- Traditionally, robotics problems have been solved **using carefully crafted analytical models** (e.g. odometry models for estimating the motion of a mobile base)
- Analytical models are indispensable when they can be defined, but **many problems of interest in robotics cannot be satisfactorily modelled, or models for those are often constrained to well-defined subproblems**

Why Robot Learning?

- Traditionally, robotics problems have been solved **using carefully crafted analytical models** (e.g. odometry models for estimating the motion of a mobile base)
- Analytical models are indispensable when they can be defined, but **many problems of interest in robotics cannot be satisfactorily modelled, or models for those are often constrained to well-defined subproblems**
- Robot learning offers the promise of **producing more flexible robots that can learn through their own interaction with the world** with minimal human involvement

Why Robot Learning?

- Traditionally, robotics problems have been solved **using carefully crafted analytical models** (e.g. odometry models for estimating the motion of a mobile base)
- Analytical models are indispensable when they can be defined, but **many problems of interest in robotics cannot be satisfactorily modelled, or models for those are often constrained to well-defined subproblems**
- Robot learning offers the promise of **producing more flexible robots that can learn through their own interaction with the world** with minimal human involvement
- Learning is also an **essential component when developing biologically-inspired robots** (e.g. in the context of cognitive robotics)

Learning Use Case 1: Social Robot Personalisation

- Let's suppose that you want to use a robot for robot-assisted therapy
- Therapy works best if it is personalised to the specific needs of the person under therapy — but **the needs of every possible person are impossible to foresee in advance**
- A more generalisable approach is to **define a set of therapy aspects that can be varied** (e.g. the manner in which the robot speaks) and **allow the robot's behaviour to be adapted based on direct feedback**
- In this case, learning can involve learning from the person under therapy (feedback-based) or learning from a therapist (guidance-based)

Learning Use Case 2: Lifelong Object Learning

- Consider a mobile manipulator that needs to pack items in a packing center
- The items that the robot needs to handle can change over time — **it is not possible to consider all possible objects that the robot needs to manipulate**
- The robot will be more useful in the packing center if it can **learn to recognise and appropriately manipulate objects continuously**
- Here, learning can be performed by trial-and-error interactions with an object (e.g. self-supervised) or by observing an expert (demonstration-based)

Robot Learning Problems

There is a large variety of robot problems where learning can be useful and which require different learning strategies to be applied. Some examples are:

Robot Learning Problems

There is a large variety of robot problems where learning can be useful and which require different learning strategies to be applied. Some examples are:

Predictive model learning

Acquiring models that enable a robot to predict how the world changes as a result of performing certain actions

Robot Learning Problems

There is a large variety of robot problems where learning can be useful and which require different learning strategies to be applied. Some examples are:

Predictive model learning

Acquiring models that enable a robot to predict how the world changes as a result of performing certain actions

Learning for scene understanding

Learning models for recognising parts of a scene (e.g. objects) or extracting relations between scene parts (e.g. spatial object relations)

Robot Learning Problems

There is a large variety of robot problems where learning can be useful and which require different learning strategies to be applied. Some examples are:

Predictive model learning

Acquiring models that enable a robot to predict how the world changes as a result of performing certain actions

Motor skill learning

Learning an execution policy for how a robot should move based on perceived environment information

Learning for scene understanding

Learning models for recognising parts of a scene (e.g. objects) or extracting relations between scene parts (e.g. spatial object relations)

Robot Learning Problems

There is a large variety of robot problems where learning can be useful and which require different learning strategies to be applied. Some examples are:

Predictive model learning

Acquiring models that enable a robot to predict how the world changes as a result of performing certain actions

Motor skill learning

Learning an execution policy for how a robot should move based on perceived environment information

Learning for scene understanding

Learning models for recognising parts of a scene (e.g. objects) or extracting relations between scene parts (e.g. spatial object relations)

Learning planning models

Acquiring models of action preconditions and/or effects that enable a robot to create action plans for solving complex tasks

State

- When making decisions about how to act in the environment, a robot needs to take into account **relevant information about itself and the environment**

State

- When making decisions about how to act in the environment, a robot needs to take into account **relevant information about itself and the environment**
- This information is summarised by **a state** $s_t, t \in [0, \infty)$

State

- When making decisions about how to act in the environment, a robot needs to take into account **relevant information about itself and the environment**
- This information is summarised by **a state $s_t, t \in [0, \infty)$**
- The state representation can **include prior environment- or task-relevant information** (e.g. there are objects in the environment, so the state represents their poses) or **can be unstructured** (e.g. a raw image)

State

- When making decisions about how to act in the environment, a robot needs to take into account **relevant information about itself and the environment**
- This information is summarised by **a state** $s_t, t \in [0, \infty)$
- The state representation can **include prior environment- or task-relevant information** (e.g. there are objects in the environment, so the state represents their poses) or **can be unstructured** (e.g. a raw image)
- The state space S is the **collection of all possible states** that the environment and a robot can be in (can be discrete or continuous)

State

- When making decisions about how to act in the environment, a robot needs to take into account **relevant information about itself and the environment**
- This information is summarised by **a state $\boldsymbol{s}_t, t \in [0, \infty)$**
- The state representation can **include prior environment- or task-relevant information** (e.g. there are objects in the environment, so the state represents their poses) or **can be unstructured** (e.g. a raw image)
- The state space S is the **collection of all possible states** that the environment and a robot can be in (can be discrete or continuous)

A state $\boldsymbol{s}_t, t \in [0, \infty]$ includes information about a robot and its environment. The state space S represents the set of all possible states a robot and its environment can be in.

Action

- Robot learning is typically performed so that a robot can **identify the best way to interact with its environment**

Action

- Robot learning is typically performed so that a robot can **identify the best way to interact with its environment**
- A robot interacts with its environment by **performing actions** $a_t, t \in [0, \infty)$

Action

- Robot learning is typically performed so that a robot can **identify the best way to interact with its environment**
- A robot interacts with its environment by **performing actions** $\boldsymbol{a}_t, t \in [0, \infty)$
- Actions $\boldsymbol{a}$ can be **defined at different level of abstractions** — they can be low-level actions (e.g. joint velocity commands) or high-level actions (e.g. picking an object)

Action

- Robot learning is typically performed so that a robot can **identify the best way to interact with its environment**
- A robot interacts with its environment by **performing actions** $a_t, t \in [0, \infty)$
- Actions $\boldsymbol{a}$ can be **defined at different level of abstractions** — they can be low-level actions (e.g. joint velocity commands) or high-level actions (e.g. picking an object)
- **The collection of actions that can be performed by a robot** is called an action space A (can also be discrete or continuous)

Action

- Robot learning is typically performed so that a robot can **identify the best way to interact with its environment**
- A robot interacts with its environment by **performing actions** $\boldsymbol{a}_t, t \in [0, \infty)$
- Actions $\boldsymbol{a}$ can be **defined at different level of abstractions** — they can be low-level actions (e.g. joint velocity commands) or high-level actions (e.g. picking an object)
- **The collection of actions that can be performed by a robot** is called an action space A (can also be discrete or continuous)

An action $\boldsymbol{a}_t, t \in [0, \infty)$ is a process that allows a robot to interact with its environment. The action space A is the set of all actions that the robot can perform.

Policy

- During its operation, a robot needs to **select actions a_t that will be useful for achieving its goals** (whatever those goals are)

Policy

- During its operation, a robot needs to **select actions a_t that will be useful for achieving its goals** (whatever those goals are)
- The selection of actions is done by taking the **current state $\mathbf{s}_t$** into account, potentially together with **other information $\mathbf{c}^T$ about the ongoing task**

Policy

- During its operation, a robot needs to **select actions a_t that will be useful for achieving its goals** (whatever those goals are)
- The selection of actions is done by taking the **current state $\mathbf{s}_t$** into account, potentially together with **other information $\mathbf{c}^T$ about the ongoing task**
- **The selection of actions is done by a policy π**
  - In the simplest case, this is a mapping $\pi : S \rightarrow A$
  - In a more general case, task-relevant information is also considered, so $\pi : (S, C) \rightarrow A$

Policy

- During its operation, a robot needs to **select actions a_t that will be useful for achieving its goals** (whatever those goals are)
- The selection of actions is done by taking the **current state $\mathbf{s}_t$** into account, potentially together with **other information $\mathbf{c}^T$ about the ongoing task**
- **The selection of actions is done by a policy π**
  - In the simplest case, this is a mapping $\pi : S \to A$
  - In a more general case, task-relevant information is also considered, so $\pi : (S, C) \to A$

A policy is a mapping π that maps a given state, and potentially other task-relevant information, to an action that a robot can perform

Visuomotor Policy

- In many cases in robotics, it can be impractical to define a task-relevant state space S; instead, it is more practical to **select actions by processing visual information directly**

Visuomotor Policy

- In many cases in robotics, it can be impractical to define a task-relevant state space S; instead, it is more practical to **select actions by processing visual information directly**
- A policy π that is defined to **perform direct selection of robot motor actions from visual input is called a visuomotor policy**

Visuomotor Policy

- In many cases in robotics, it can be impractical to define a task-relevant state space S; instead, it is more practical to **select actions by processing visual information directly**
- A policy π that is defined to **perform direct selection of robot motor actions from visual input is called a visuomotor policy**
- Note that the state space in a visuomotor policy can, and often does, include **additional, non-visual, information as well** (e.g. information about the robot's internal state)

Visuomotor Policy

- In many cases in robotics, it can be impractical to define a task-relevant state space S; instead, it is more practical to **select actions by processing visual information directly**
- A policy π that is defined to **perform direct selection of robot motor actions from visual input is called a visuomotor policy**
- Note that the state space in a visuomotor policy can, and often does, include **additional, non-visual, information as well** (e.g. information about the robot's internal state)
- Visuomotor policies are directly inspired by biological systems

Visuomotor Policy

- In many cases in robotics, it can be impractical to define a task-relevant state space S; instead, it is more practical to **select actions by processing visual information directly**
- A policy π that is defined to **perform direct selection of robot motor actions from visual input is called a visuomotor policy**
- Note that the state space in a visuomotor policy can, and often does, include **additional, non-visual, information as well** (e.g. information about the robot's internal state)
- Visuomotor policies are directly inspired by biological systems

A policy π for which the state space S is partially or completely defined by visual data and the action space A is defined by robot motion commands is called a visuomotor policy

Skill

- Typically, a robot will need to **use a collection of different policies for performing tasks**; such policies may have the same action space (e.g. joint torques), but are applicable in different cases

Skill

- Typically, a robot will need to **use a collection of different policies for performing tasks**; such policies may have the same action space (e.g. joint torques), but are applicable in different cases
- An execution policy together with conditions under which the policy is applicable and conditions under which the policy terminates is called a **robot skill** $\mathcal{S} = (\mathcal{S}_{\mathcal{I}}, \mathcal{S}_{\mathcal{T}}, \pi)$

Skill

- Typically, a robot will need to **use a collection of different policies for performing tasks**; such policies may have the same action space (e.g. joint torques), but are applicable in different cases
- An execution policy together with conditions under which the policy is applicable and conditions under which the policy terminates is called a **robot skill** $\mathcal{S} = (\mathcal{S}_\mathcal{I}, \mathcal{S}_\mathcal{T}, \pi)$
- This formulation enables **skill composition** (a complete task can be decomposed into a sequence of skills) as well as **hierarchical skill definitions** (each skill can be composed of multiple sub-skills)

Skill

- Typically, a robot will need to **use a collection of different policies for performing tasks**; such policies may have the same action space (e.g. joint torques), but are applicable in different cases
- An execution policy together with conditions under which the policy is applicable and conditions under which the policy terminates is called a **robot skill** $\mathcal{S} = (\mathcal{S}_\mathcal{I}, \mathcal{S}_\mathcal{T}, \pi)$
- This formulation enables **skill composition** (a complete task can be decomposed into a sequence of skills) as well as **hierarchical skill definitions** (each skill can be composed of multiple sub-skills)

A skill $\mathcal{S} = (\mathcal{S}_\mathcal{I}, \mathcal{S}_\mathcal{T}, \pi)$ specifies an execution policy π together conditions $\mathcal{S}_\mathcal{I}$ under which the skill can be applied and conditions $\mathcal{S}_\mathcal{T}$ under which the skill execution terminates

Prior Knowledge (Inductive Bias)

- As robots are physical agents, they are **embedded — and constrained by — the physical world**, which comes with many constraints (e.g. physical facts such as that two objects cannot occupy the same space at the same time)

Prior Knowledge (Inductive Bias)

- As robots are physical agents, they are **embedded — and constrained by — the physical world**, which comes with many constraints (e.g. physical facts such as that two objects cannot occupy the same space at the same time)
- Most robot learning problems are **too complex to perform from scratch**, but **become tractable if restrictions are put on the learning problem**

Prior Knowledge (Inductive Bias)

- As robots are physical agents, they are **embedded — and constrained by — the physical world**, which comes with many constraints (e.g. physical facts such as that two objects cannot occupy the same space at the same time)
- Most robot learning problems are **too complex to perform from scratch**, but **become tractable if restrictions are put on the learning problem**
- Prior knowledge (also called inductive bias) can **simplify a learning problem by restricting the type or amount of information that needs to be learned**

Prior Knowledge (Inductive Bias)

- As robots are physical agents, they are **embedded — and constrained by — the physical world**, which comes with many constraints (e.g. physical facts such as that two objects cannot occupy the same space at the same time)
- Most robot learning problems are **too complex to perform from scratch**, but **become tractable if restrictions are put on the learning problem**
- Prior knowledge (also called inductive bias) can **simplify a learning problem by restricting the type or amount of information that needs to be learned**
- Inducing prior knowledge into a learning problem is also a strategy for ensuring the safety of the learning problem or the learned behaviour

Prior Knowledge (Inductive Bias)

- As robots are physical agents, they are **embedded — and constrained by — the physical world**, which comes with many constraints (e.g. physical facts such as that two objects cannot occupy the same space at the same time)
- Most robot learning problems are **too complex to perform from scratch**, but **become tractable if restrictions are put on the learning problem**
- Prior knowledge (also called inductive bias) can **simplify a learning problem by restricting the type or amount of information that needs to be learned**
- Inducing prior knowledge into a learning problem is also a strategy for ensuring the safety of the learning problem or the learned behaviour

Prior knowledge is any type of knowledge (about the environment or the concrete tasks) that is embedded into a learning algorithm and which simplifies a learning problem

Sim-to-Real Transfer

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**

Sim-to-Real Transfer

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**
- For this reason, data collection for learning is often done in a **simulated environment that models the target real-world environment** as closely as possible

Sim-to-Real Transfer

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**
- For this reason, data collection for learning is often done in a **simulated environment that models the target real-world environment** as closely as possible
- A simulation is, however, never a completely faithful model, so **a model learned in simulation is unlikely to be directly usable in the real environment**

Sim-to-Real Transfer

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**
- For this reason, data collection for learning is often done in a **simulated environment that models the target real-world environment** as closely as possible
- A simulation is, however, never a completely faithful model, so **a model learned in simulation is unlikely to be directly usable in the real environment**
- Sim-to-real transfer is thus the problem of **adapting a simulation-based model to the real world**

Sim-to-Real Transfer

- In many learning scenarios in robotics, **collecting real-world for learning data can be impractical or dangerous**
- For this reason, data collection for learning is often done in a **simulated environment that models the target real-world environment** as closely as possible
- A simulation is, however, never a completely faithful model, so **a model learned in simulation is unlikely to be directly usable in the real environment**
- Sim-to-real transfer is thus the problem of **adapting a simulation-based model to the real world**

Sim-to-real transfer is the process of making a model that was trained in a simulated environment suitable for use in the real, target environment

Robot Learning Techniques

What Kinds of Learning are Useful in Robotics?

- Throughout the course, we will encounter a variety of learning concepts and techniques that can used in different contexts
  - Often, the same learning problem can be solved with different techniques
- On the following few slides, we will briefly introduce different learning techniques that we will encounter during the course

General Learning Paradigms

Supervised learning

Given: inputs $\boldsymbol{x}$ and labels/outputs $\boldsymbol{y}$
Learn: a mapping $f(\boldsymbol{x}) = \boldsymbol{y}$

General Learning Paradigms

Supervised learning

Given: inputs $\boldsymbol{x}$ and labels/outputs $\boldsymbol{y}$

Learn: a mapping $f(\boldsymbol{x}) = \boldsymbol{y}$

Unsupervised learning

Given: inputs $\boldsymbol{x}$

Learn: some structure about the data

General Learning Paradigms

Supervised learning

Given: inputs $\boldsymbol{x}$ and labels/outputs $\boldsymbol{y}$
Learn: a mapping $f(\boldsymbol{x}) = \boldsymbol{y}$

Unsupervised learning

Given: inputs $\boldsymbol{x}$
Learn: some structure about the data

Reinforcement learning

Given: a set of experiences (observed states, applied actions, and obtained rewards along the way)
Learn: how to behave so that the reward is maximised (i.e. an optimal mapping from observed states to actions)

Learning from Demonstration (LfD)

- In most applications, it is impractical to preprogram robot behaviours that will cover all relevant cases of a robot's use; instead, it is desirable that **the end users can adapt their robot to their own needs**

Learning from Demonstration (LfD)

- In most applications, it is impractical to preprogram robot behaviours that will cover all relevant cases of a robot's use; instead, it is desirable that **the end users can adapt their robot to their own needs**
- LfD is a technique that makes it possible to **acquire behaviours by observing experts** rather than by explicit programming

Learning from Demonstration (LfD)

- In most applications, it is impractical to preprogram robot behaviours that will cover all relevant cases of a robot's use; instead, it is desirable that **the end users can adapt their robot to their own needs**
- LfD is a technique that makes it possible to **acquire behaviours by observing experts** rather than by explicit programming
- There are **different ways in which demonstrations can be performed**, such as by moving a robot directly, by performing demonstrations in a virtual environment, or by letting a robot observe using exteroceptive sensors

Learning from Demonstration (LfD)

- In most applications, it is impractical to preprogram robot behaviours that will cover all relevant cases of a robot's use; instead, it is desirable that **the end users can adapt their robot to their own needs**
- LfD is a technique that makes it possible to **acquire behaviours by observing experts** rather than by explicit programming
- There are **different ways in which demonstrations can be performed**, such as by moving a robot directly, by performing demonstrations in a virtual environment, or by letting a robot observe using exteroceptive sensors
- Regardless of the demonstration mode, demonstrations result in **data that needs to be processed according to the demonstration learning objectives**

Learning from Demonstration (LfD)

- In most applications, it is impractical to preprogram robot behaviours that will cover all relevant cases of a robot's use; instead, it is desirable that **the end users can adapt their robot to their own needs**
- LfD is a technique that makes it possible to **acquire behaviours by observing experts** rather than by explicit programming
- There are **different ways in which demonstrations can be performed**, such as by moving a robot directly, by performing demonstrations in a virtual environment, or by letting a robot observe using exteroceptive sensors
- Regardless of the demonstration mode, demonstrations result in **data that needs to be processed according to the demonstration learning objectives**

Learning from demonstration is a robot programming technique in which models of motions or complete tasks are acquired from user demonstrations

Reinforcement Learning (RL)

- To enable the practicality of a robot, it is desirable to enable the robot to figure out its own execution policy based on **trial-and-error interactions with the environment in order to maximise a given performance objective**

Reinforcement Learning (RL)

- To enable the practicality of a robot, it is desirable to enable the robot to figure out its own execution policy based on **trial-and-error interactions with the environment in order to maximise a given performance objective**
- In RL, a robot **receives rewards** $r(s_t, a_t)$ as it performs actions in different environment states

Reinforcement Learning (RL)

- To enable the practicality of a robot, it is desirable to enable the robot to figure out its own execution policy based on **trial-and-error interactions with the environment in order to maximise a given performance objective**
- In RL, a robot **receives rewards** $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ as it performs actions in different environment states
- The RL learning objective is to **find a policy** π^* **that maximises the (discounted) expected return** $E\left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$ (i.e. the expected sum of rewards over the complete interaction)

Reinforcement Learning (RL)

- To enable the practicality of a robot, it is desirable to enable the robot to figure out its own execution policy based on **trial-and-error interactions with the environment in order to maximise a given performance objective**
- In RL, a robot **receives rewards** $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ as it performs actions in different environment states
- The RL learning objective is to **find a policy** π^* **that maximises the (discounted) expected return** $E\left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$ (i.e. the expected sum of rewards over the complete interaction)
- The reward used for learning can be **sparse** (received only in certain key states of the interaction) or **shaped** (received continuously throughout the interaction)

Reinforcement Learning (RL)

- To enable the practicality of a robot, it is desirable to enable the robot to figure out its own execution policy based on **trial-and-error interactions with the environment in order to maximise a given performance objective**
- In RL, a robot **receives rewards** $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ as it performs actions in different environment states
- The RL learning objective is to **find a policy** π^* **that maximises the (discounted) expected return** $E\left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$ (i.e. the expected sum of rewards over the complete interaction)
- The reward used for learning can be **sparse** (received only in certain key states of the interaction) or **shaped** (received continuously throughout the interaction)

Reinforcement learning is a learning technique based on which a robot can learn an optimal policy π^* that maximises an expected return over the robot's operation

Semi-Supervised Learning

- In certain learning scenarios, there is a large amount of data available, but only a small part is labelled (labelling is usually a time-consuming process)

Semi-Supervised Learning

- In certain learning scenarios, there is a large amount of data available, but only a small part is labelled (labelling is usually a time-consuming process)
- In such cases, **supervised learning on its own is unsuitable** (the unlabelled data would be discarded), **but so is unsupervised learning** (the valuable label information would be ignored)

Semi-Supervised Learning

- In certain learning scenarios, there is a large amount of data available, but only a small part is labelled (labelling is usually a time-consuming process)
- In such cases, **supervised learning on its own is unsuitable** (the unlabelled data would be discarded), **but so is unsupervised learning** (the valuable label information would be ignored)
- Semi-supervised learning can be a compromise solution that uses the labelled data to **learn an initial classifier** and to then **generate labels for the unlabelled data**

Semi-Supervised Learning

- In certain learning scenarios, there is a large amount of data available, but only a small part is labelled (labelling is usually a time-consuming process)
- In such cases, **supervised learning on its own is unsuitable** (the unlabelled data would be discarded), **but so is unsupervised learning** (the valuable label information would be ignored)
- Semi-supervised learning can be a compromise solution that uses the labelled data to **learn an initial classifier** and to then **generate labels for the unlabelled data**
- **This process is repeated iteratively** (with the most confident labels of the unlabelled data used as ground-truth labels in subsequent iterations) so that a classifier for the whole dataset can be learned (whether the process results in a good classifier depends on various properties of the data)

Semi-Supervised Learning

- In certain learning scenarios, there is a large amount of data available, but only a small part is labelled (labelling is usually a time-consuming process)
- In such cases, **supervised learning on its own is unsuitable** (the unlabelled data would be discarded), **but so is unsupervised learning** (the valuable label information would be ignored)
- Semi-supervised learning can be a compromise solution that uses the labelled data to **learn an initial classifier** and to then **generate labels for the unlabelled data**
- **This process is repeated iteratively** (with the most confident labels of the unlabelled data used as ground-truth labels in subsequent iterations) so that a classifier for the whole dataset can be learned (whether the process results in a good classifier depends on various properties of the data)

Semi-supervised learning is a technique that initially uses a small amount of labelled data to learn a classifier; the classifier is used to classify a large set of unlabelled data, which is used to iteratively train a better classifier over time

Self-Supervised Learning

- Learning scenarios in which only unlabelled data are available are also common — but in which labels may be given later, when domain-specific data are available

Self-Supervised Learning

- Learning scenarios in which only unlabelled data are available are also common — but in which labels may be given later, when domain-specific data are available
- While unsupervised learning can be used to learn the structure of the data, it has been shown that **generating labels automatically and learning from those can be beneficial for pre-training models** that can then be fine-tuned for concrete tasks (e.g. in natural language processing)

Self-Supervised Learning

- Learning scenarios in which only unlabelled data are available are also common — but in which labels may be given later, when domain-specific data are available
- While unsupervised learning can be used to learn the structure of the data, it has been shown that **generating labels automatically and learning from those can be beneficial for pre-training models** that can then be fine-tuned for concrete tasks (e.g. in natural language processing)
- Self-supervision can be performed by **solving a prediction task** (by masking parts of the input) or **using contrastive learning** (where a similarity model between inputs is learned)

Self-Supervised Learning

- Learning scenarios in which only unlabelled data are available are also common — but in which labels may be given later, when domain-specific data are available
- While unsupervised learning can be used to learn the structure of the data, it has been shown that **generating labels automatically and learning from those can be beneficial for pre-training models** that can then be fine-tuned for concrete tasks (e.g. in natural language processing)
- Self-supervision can be performed by **solving a prediction task** (by masking parts of the input) or **using contrastive learning** (where a similarity model between inputs is learned)
- In robotics, self-supervision is attractive because it can enable a robot to learn skills without requiring human involvement

Self-Supervised Learning

- Learning scenarios in which only unlabelled data are available are also common — but in which labels may be given later, when domain-specific data are available
- While unsupervised learning can be used to learn the structure of the data, it has been shown that **generating labels automatically and learning from those can be beneficial for pre-training models** that can then be fine-tuned for concrete tasks (e.g. in natural language processing)
- Self-supervision can be performed by **solving a prediction task** (by masking parts of the input) or **using contrastive learning** (where a similarity model between inputs is learned)
- In robotics, self-supervision is attractive because it can enable a robot to learn skills without requiring human involvement

Given only unlabelled data, self-supervised learning is a technique that acquires a predictive model trained with self-generated data labels

Curriculum Learning

- In some cases, learning the task of interest may be too complex, but **learning multiple tasks with increasing complexity may simplify the learning problem** (like learning math in school)

Curriculum Learning

- In some cases, learning the task of interest may be too complex, but **learning multiple tasks with increasing complexity may simplify the learning problem** (like learning math in school)
- **Curriculum learning makes use of a curriculum to direct the learning of a robot** so that it first focuses on learning simpler tasks and then progressively moves towards more complex tasks, **until the task of interest is finally learned**

Curriculum Learning

- In some cases, learning the task of interest may be too complex, but **learning multiple tasks with increasing complexity may simplify the learning problem** (like learning math in school)
- **Curriculum learning makes use of a curriculum to direct the learning of a robot** so that it first focuses on learning simpler tasks and then progressively moves towards more complex tasks, **until the task of interest is finally learned**
- Curriculum learning can be performed with a **manually specified curriculum**, but can also be done so that **the curriculum itself is adapted dynamically**

Curriculum Learning

- In some cases, learning the task of interest may be too complex, but **learning multiple tasks with increasing complexity may simplify the learning problem** (like learning math in school)
- **Curriculum learning makes use of a curriculum to direct the learning of a robot** so that it first focuses on learning simpler tasks and then progressively moves towards more complex tasks, **until the task of interest is finally learned**
- Curriculum learning can be performed with a **manually specified curriculum**, but can also be done so that **the curriculum itself is adapted dynamically**

Curriculum learning is a strategy based on which simpler tasks are learned before complex tasks — but can also include learning the curriculum itself

Safe Learning

- In robot learning, it is both desirable and often unavoidable to collect data from a physical robot, but **this can potentially lead to dangerous situations** (that might damage the robot or the environment)

Safe Learning

- In robot learning, it is both desirable and often unavoidable to collect data from a physical robot, but **this can potentially lead to dangerous situations** (that might damage the robot or the environment)
- **A desirable property of learning algorithms is thus having safety guarantees**, namely the selection of robot actions should be done to prevent such situations

Safe Learning

- In robot learning, it is both desirable and often unavoidable to collect data from a physical robot, but **this can potentially lead to dangerous situations** (that might damage the robot or the environment)
- **A desirable property of learning algorithms is thus having safety guarantees**, namely the selection of robot actions should be done to prevent such situations
- In safe learning, the focus is on **developing learning algorithms and execution policies that can provably, or with a high probability, avoid unsafe behaviours**

Safe Learning

- In robot learning, it is both desirable and often unavoidable to collect data from a physical robot, but **this can potentially lead to dangerous situations** (that might damage the robot or the environment)
- **A desirable property of learning algorithms is thus having safety guarantees**, namely the selection of robot actions should be done to prevent such situations
- In safe learning, the focus is on **developing learning algorithms and execution policies that can provably, or with a high probability, avoid unsafe behaviours**
- Safety is defined by **safety constraints**, which can be specified on the system's state, the actions, or the system's stability

Safe Learning

- In robot learning, it is both desirable and often unavoidable to collect data from a physical robot, but **this can potentially lead to dangerous situations** (that might damage the robot or the environment)
- **A desirable property of learning algorithms is thus having safety guarantees**, namely the selection of robot actions should be done to prevent such situations
- In safe learning, the focus is on **developing learning algorithms and execution policies that can provably, or with a high probability, avoid unsafe behaviours**
- Safety is defined by **safety constraints**, which can be specified on the system's state, the actions, or the system's stability

The objective of safe learning is to enable selection of learning experiences that are provably safe to be executed on a real system