

# Sequential Decision Making and Reinforcement Learning

(SDMRL)

Supervised Learning vs. Model Bases Planning for Long-Term Decision-Making

---

Sarah Keren

The Taub Faculty of Computer Science  
Technion - Israel Institute of Technology

- Model-based / model-free spectrum
- Supervised Learning for long-term decision-making
- Model-based planning for long-term decision-making

## Model-free vs. Model-based Decision-making

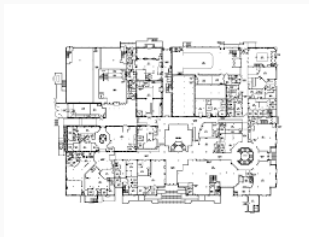
---

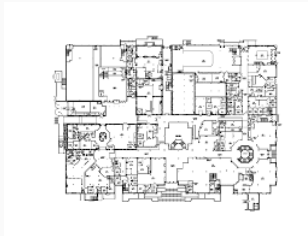
Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning





How to come up with 'good' policies?



How to come up with 'good' policies?



How to come up with 'good' policies?  
If we want to avoid collision?



How to come up with 'good' policies?

If we want to avoid collision?

If we want to turn left at the next intersection?





How to come up with 'good' policies?

If we want to avoid collision?

If we want to turn left at the next intersection?

If we want to buy apples and make it on time for dinner but have fuel for about 30 mins?

An agent typically maintains one or more of these components:

- **Model:** agent's representation of the environment
- **Value function:** how good is each state and/or action
- **Policy:** agent's behaviour function
  - **Deterministic Policy:** a mapping  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  from states/observations to actions.
  - **Stochastic Policy:** a mapping  $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$  from state and action pairs to the probability  $\pi(a|s)$  of taking action  $a$  when in state  $s$ .

In partially observable settings, mapping can be from beliefs  $\mathcal{B}$  instead of actions.

- Expected return for episodic tasks:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T = \sum_{k=0}^T R_{t+k+1}$$

- Expected return for continuing tasks:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

when  $\gamma$  is the discount factor.

- Returns at successive time steps are related to each other:

$$G_t = R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots) = R_{t+1} + \gamma G_{t+1}$$

# Value Functions

Value functions estimate how ‘good’ it is for the agent to be in a given state or how good it is to perform a given action in a given state.

“How good” is defined in terms of expected return.

Since the rewards the agent can expect to receive in the future depend on what actions it will take, value functions are defined with respect to particular policies.

**State-Value Function for Policy  $\pi$ :**

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi} [G_t \mid S_t = s]$$

**Action-Value Function for Policy  $\pi$ :**

$$q_{\pi}(s, a) \doteq \mathbb{E}_{\pi} [G_t \mid S_t = s, A_t = a]$$

What is  $G_t$  ?

Reinforcement  
Learning

(SDMRL)

Sarah Keren

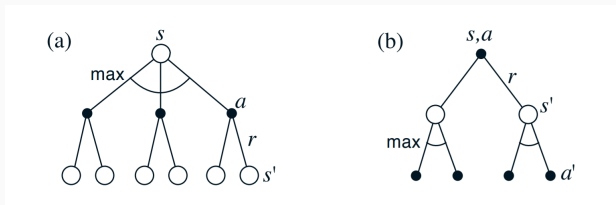
Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Value Functions



Backup diagrams for  $v_\pi(s)$  (left) and  $q_\pi(s, a)$  (right).

Since it may not be practical to keep separate averages for each state individually  $v_\pi$  and  $q_\pi$  are often represented as parameterized functions (with fewer parameters than states).

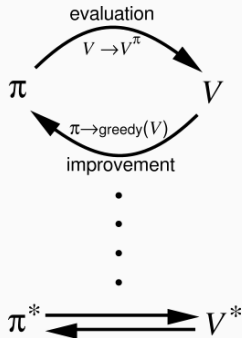
How to find an optimal policy ?

## Recipes for Control Approaches

---

# Policy Evaluation and Policy Update

- Prediction / Evaluation: evaluate the future given a policy
- Control/ Update / Improvement: optimize the policy.





# Control Approaches Skeleton

Reinforcement  
Learning

(SDMRL)

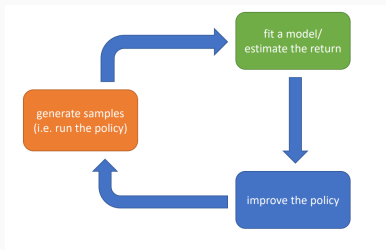
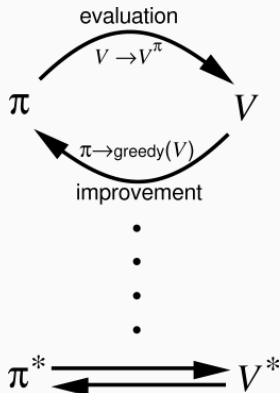
Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning



Left image by Sutton and Barto. Right image By Sergey Levine

# Model Based vs. Model Free

- For this distinction, a **model** typically refers to the transition function  $\mathcal{P}$  and the reward function  $\mathcal{R}$ .
- A model free approach only maintains a policy or value function, but no model.
- A model-based approach maintains a policy or value function and a model.



[https://www.davidsilver.uk/wp-content/uploads/2020/03/intro\\_RL.pdf](https://www.davidsilver.uk/wp-content/uploads/2020/03/intro_RL.pdf)

# Value vs. Policy Based

- **Value Based:**
  - No Policy (Implicit)
  - Value Function
- **Policy Based**
  - Policy
  - No Value Function
- **Combined approach (a.k.a Actor-Critic)**
  - Policy
  - Value Function

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# On Policy vs. Off Policy

According to Sutton and Barto 2018

- **On-policy** methods evaluate and improve the policy based on the policy that is used to make decisions.
- **Off-policy** methods evaluate or improve a policy different from that used to generate the data.
- We distinguish between the **behavior policy**, according to which an agent interacts with the environment and **target policy** the policy the agent is trying to learn that will optimize its utility.
- In on-policy methods behavior policy == target policy. In off-policy methods they are different.

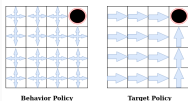


Image from <https://towardsdatascience.com/>

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

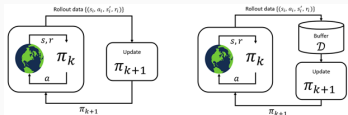
Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Online vs. Offline

- **Online** iteratively collecting data (a.k.a. experiences) by interacting with the environment.
- **Offline** utilize previously collected data, without additional online data collection.
  - resembles the standard supervised learning.
  - make it possible to turn large datasets into powerful decision making engines.
  - Batch Reinforcement Learning, behavioral cloning



From Or Rivlin <https://towardsdatascience.com/the-power-of-offline-reinforcement-learning-5e3d3942421c>

- **Planning:** the task of coming up with a sequence of actions that will achieve a goal.
  - **Sensorless (conformant) planning:** constructing sequential plans to be executed without perception
  - **Conditional (contingent) planning:** constructing a conditional plan with different branches for different contingencies that could happen.
  - **Continuous planning:** a planner designed to persist over a lifetime.
- **Execution monitoring and replanning:** constructing a plan, but monitoring its execution and generating a new plan when necessary.

from <https://www.cpp.edu/~ftang/courses/CS420/notes/planning.pdf>

# Solution Approaches

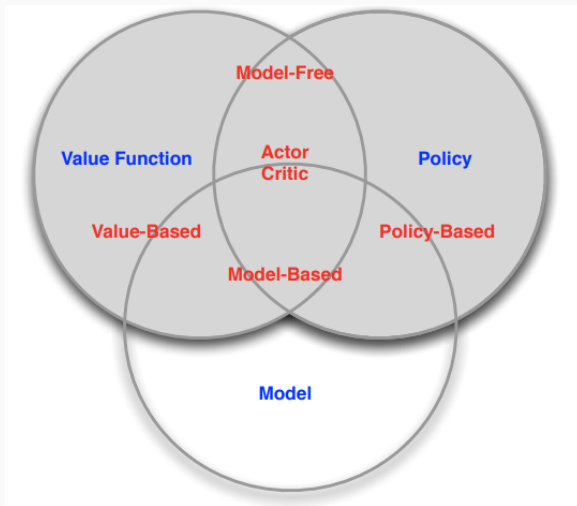


image by David Silver

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Approaches to Control

- Supervised learning
- Model-Based Planning
- Monte-Carlo methods
- Temporal-Difference methods
- Combined approaches

Model Free

Model Based

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning



# Decision-making as Supervised Learning

---

# Behavior Cloning: Example

Reinforcement  
Learning

(SDMRL)

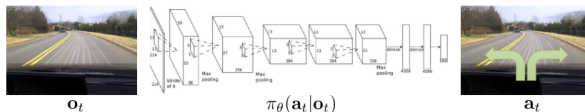
Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning



- The objective is to learn a policy by mimicking the behavior demonstrated by an expert.
- A model (policy)  $\pi_{\theta}(s)$  is trained to minimize the discrepancy between its predicted actions and the expert's actions over the collected dataset.

Can be broken down into the following steps:

- **Data Collection:** Gather a dataset of state-action pairs  $(s_i, a_i)$ , where  $s_i$  represents the state and  $a_i$  represents the action taken by the expert at that state.
- **Model Selection:** Choose a model to approximate the expert's policy. Typically, a neural network  $\pi_{\theta}(s)$  parameterized by  $\theta$ .

Ideas for managing the learning process?

- **Loss Function:** measures the difference between the expert's actions and the actions predicted by the model. For example, mean squared error (MSE) or cross-entropy loss for discrete actions.

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \|a_i - \pi_{\theta}(s_i)\|^2$$

where  $N$  is the number of state-action pairs in the dataset.


- **Training:** Optimize the model parameters  $\theta$  by minimizing the loss function (e.g., using gradient descent)

$$\theta^* = \arg \min_{\theta} L(\theta)$$

- **Policy Execution:** Use the trained model  $\pi_{\theta^*}(s)$  as the policy to predict actions for new states during execution.

## Limitations?

# DAgger (Dataset Aggregation)

- 
1. train  $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$  from human data  $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
  2. run  $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$  to get dataset  $\mathcal{D}_\pi = \{\mathbf{o}_1, \dots, \mathbf{o}_M\}$
  3. Ask human to label  $\mathcal{D}_\pi$  with actions  $\mathbf{a}_t$
  4. Aggregate:  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

<https://jonathan-hui.medium.com/rl-imitation-learning-ac28116c02fc>

<https://bair.berkeley.edu/blog/2017/10/26/dart/>

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Applications

Reinforcement  
Learning

(SDMRL)

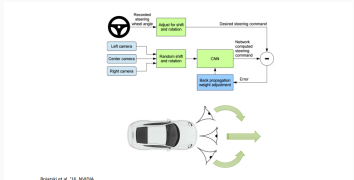
Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning



# Model-based Planning

---



# Problem-solving agents

Restricted form of general agent

```
def Simple-Problem-Solving-Agent (problem):  
    state, some description of the current world state  
    seq, an action sequence, initially empty  
    state  $\leftarrow$  UPDATE-STATE(state, percept)  
    if seq is empty then  
        seq  $\leftarrow$  SEARCHFOR SOLUTION(problem)  
    action  $\leftarrow$  SELECT ACTION(seq, state)  
    seq  $\leftarrow$  REMAINDER(seq, action)  
    return action
```

- Offline problem solving; solution executed “eyes closed.”
- Based on a model of the environment and its dynamics

Reinforcement  
Learning

(SDMRL)

Sarah Keren

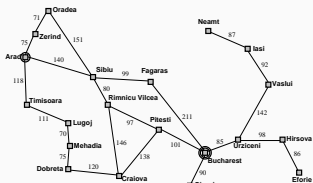
Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Examples



Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Model-Based Planning

Characterized by:

- A set of **states**  $\mathcal{S}$  a system can be in.
  - a state is a full assignment to the set of **variables** (features)  $\mathcal{X}$ .
- **Actions**  $\mathcal{A}$  change the values of certain variables.
- **Reward Function**  $\mathcal{R}$  sets a numeric signal passed from the environment (can represent cost)
  - used to signal the objective
  - some domains have **goal conditions** such that the agent should reach **goal states** that satisfy is(e.g., 'be at Austin').
- **Objective:** find a **policy** that drives the initial state into a goal state or that maximizes the expected accumulated reward.
- Language is **generic** and not domain specific.
- **Complexity:** Even in the simplest setting it is NP-hard; i.e., exponential in the number of variables in the worst case.

Reinforcement  
Learning

(SDMRL)

Sarah Keren

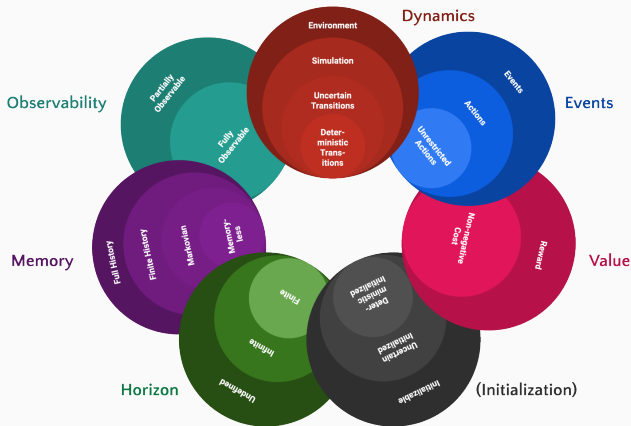
Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Many Forms of Planning



[https://github.com/fteicht/  
icaps24-skdecide-tutorial/tree/main](https://github.com/fteicht/icaps24-skdecide-tutorial/tree/main)

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Classification of search algorithms

Uninformed search vs. heuristic search:

- **uninformed search algorithms** only use the basic ingredients for general search algorithms
- **heuristic search algorithms** additionally use **heuristic functions** which estimate how close a node is to the goal

Systematic search vs. local search:

- **systematic algorithms** consider a large number of search nodes simultaneously
- **local search algorithms** work with one (or a few) candidate solutions (search nodes) at a time
- not a black-and-white distinction; there are **crossbreeds** (e.g., enforced hill-climbing)

**More and more learning-based approaches used for planning**

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Uninformed search algorithms

Popular uninformed systematic search algorithms:

- breadth-first search
- depth-first search
- iterated depth-first search

Popular uninformed local search algorithms:

- random walk

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Evaluating Algorithms

## Dimensions for evaluation

- **completeness**: always find a solution if one exists?
- **time complexity**: number of nodes generated/expanded
- **space complexity**: number of nodes in memory
- **optimality**: does it always find a least-cost solution?
- **anytime**: does the solution improve the more resources are used ?

## Time/space complexity measured in terms of

- b* maximum branching factor of the search tree
- d* depth of the least-cost solution
- m* maximum depth of the state space (may be  $\infty$ )

Reinforcement  
Learning

(SDMRL)

Sarah Keren

Model-free vs.  
Model-based  
Decision-making

Recipes for  
Control  
Approaches

Decision-making  
as Supervised  
Learning

Model-based  
Planning

# Recap

- Spectrum of approaches to control
- Various characteristics (model-free model-based, offlineonline, etc)
- Policy extraction as supervised learning
- Model-based planning in deterministic and full observable domains

