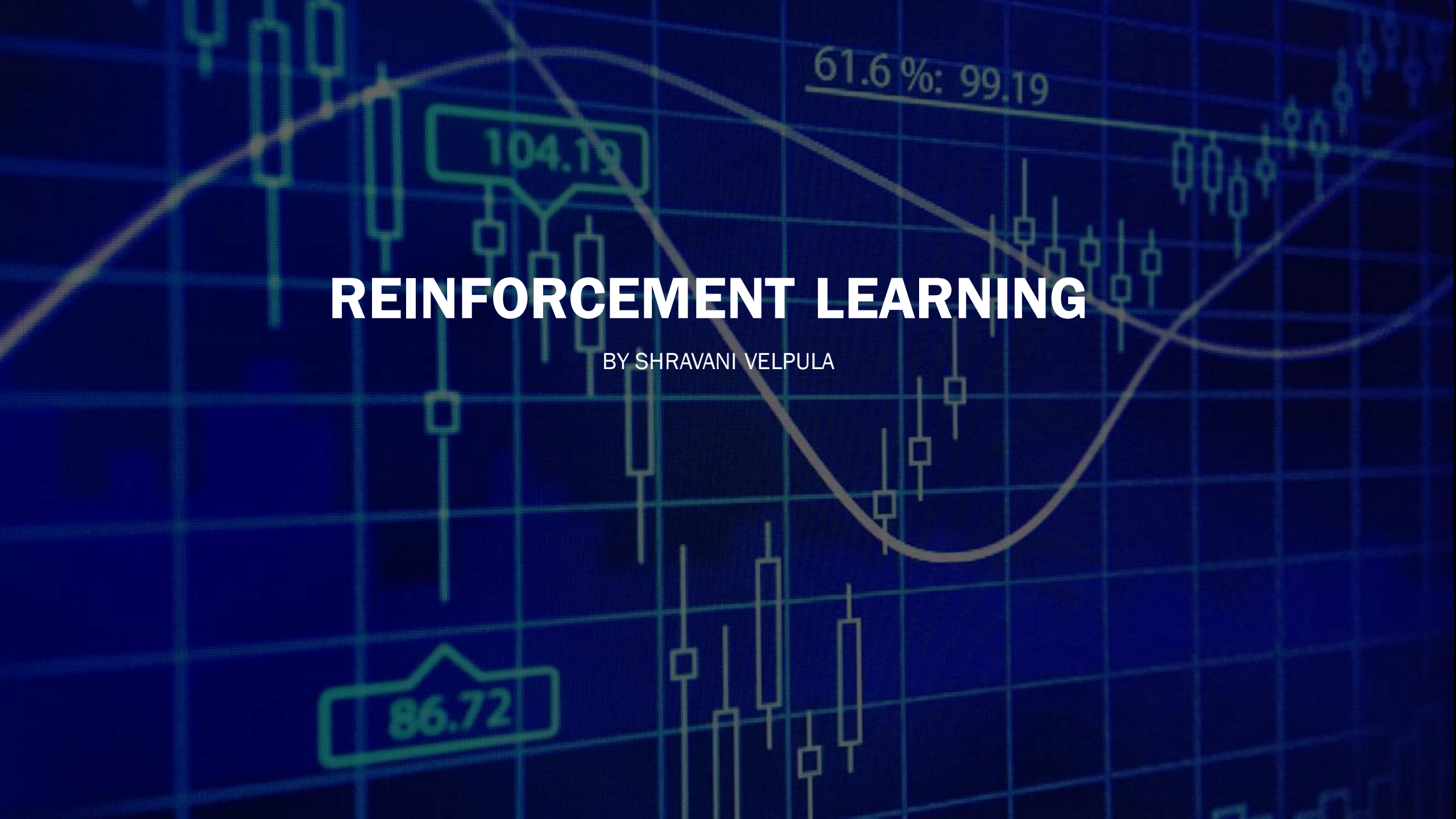


REINFORCEMENT LEARNING

BY SHRAVANI VELPULA

WHAT IS REINFORCEMENT LEARNING

- Reinforcement learning is learning what to do
- how to map situations to actions
- so as to maximize a numerical reward signal

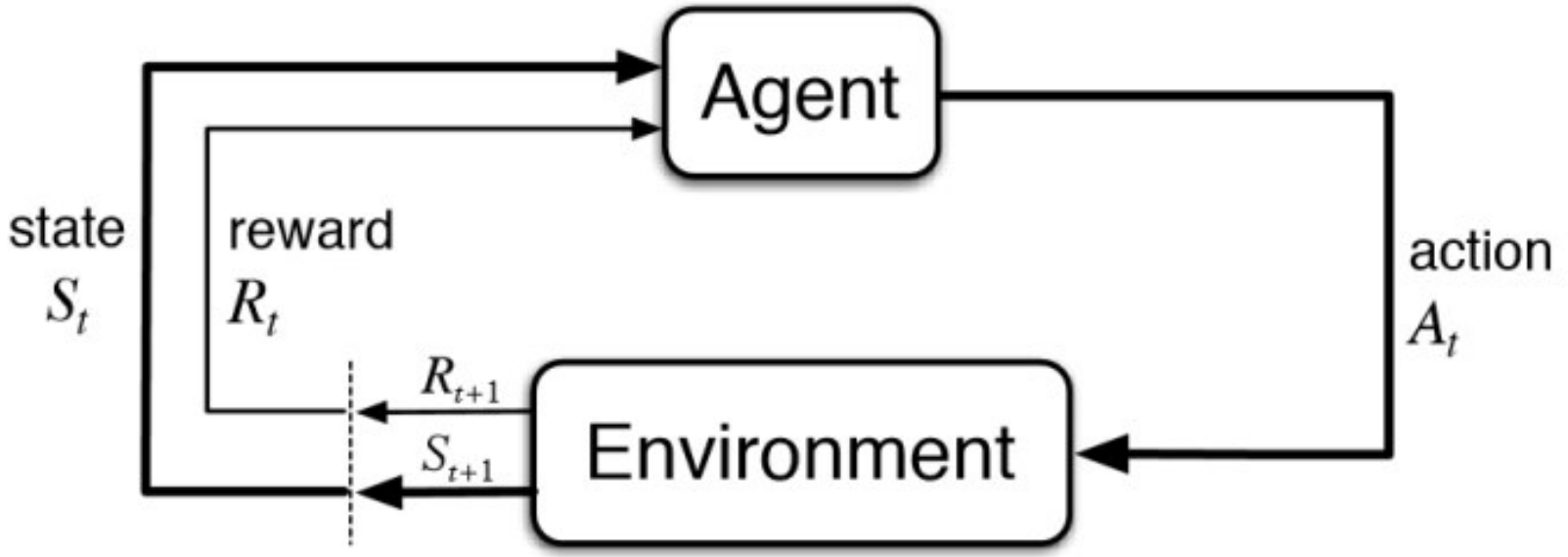
HOW IS IT DIFFERENT FROM OTHER MACHINE LEARNING PARADIGMS

- There is no supervisor, only reward signal
- Feedback is delayed, not instantaneous
- Time really matters(sequential)

RL PROBLEM

- Optimal control of incompletely known Markov Descision Process

TYPICAL RL SCENARIO



REWARDS

- A reward Rt is a scalar feedback signal
- Indicates how well agent is doing at step t
- The agent's job is to maximize cumulative reward

Reinforcement Learning is based on the reward hypothesis
Example of reward:

Play many different Atari games better than humans

+/−ve reward for increasing/decreasing score

AGENT AND ENVIRONMENT

- At each step t the agent:
  Executes action At
  Receives observation Ot
  Receives scalar reward Rt
- The environment:
  Receives action At
  Emits observation Ot+1
  Emits scalar reward Rt+1

HISTORY AND STATE

- The history is the sequence of observations, actions, rewards

$$H_t = O1, R1, A1, \ldots, A_{t-1}, O_t, R_t$$

- State is the information used to determine what happens next Formally, state is a function of the history:

$$S_t = f(H_t)$$

AGENT STATE

ENVIRONMENT STATE

INFORMATION STATE(MARKOV STATE)

AN INFORMATION STATE (A.K.A. MARKOV STATE) CONTAINS ALL USEFUL INFORMATION FROM THE HISTORY

A STATE ST IS MARKOV IF AND ONLY IF $P[S_{T+1} \mid S_T] = P[S_{T+1} \mid S_1, \ldots, S_T]$

SEQUENTIAL DECISION MAKING

- Goal: select actions to maximize total future reward
- Actions may have long term consequences
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward

IMPORTANT COMPONENTS OF A REINFORCEMENT LEARNING AGENT

- Policy: agent's behavior function
- Value function: how good is each state and/or action
- Model: agent's representation of the environment

POLICY

- Policy defines the behavior of the agent.

  It is a map from state to action

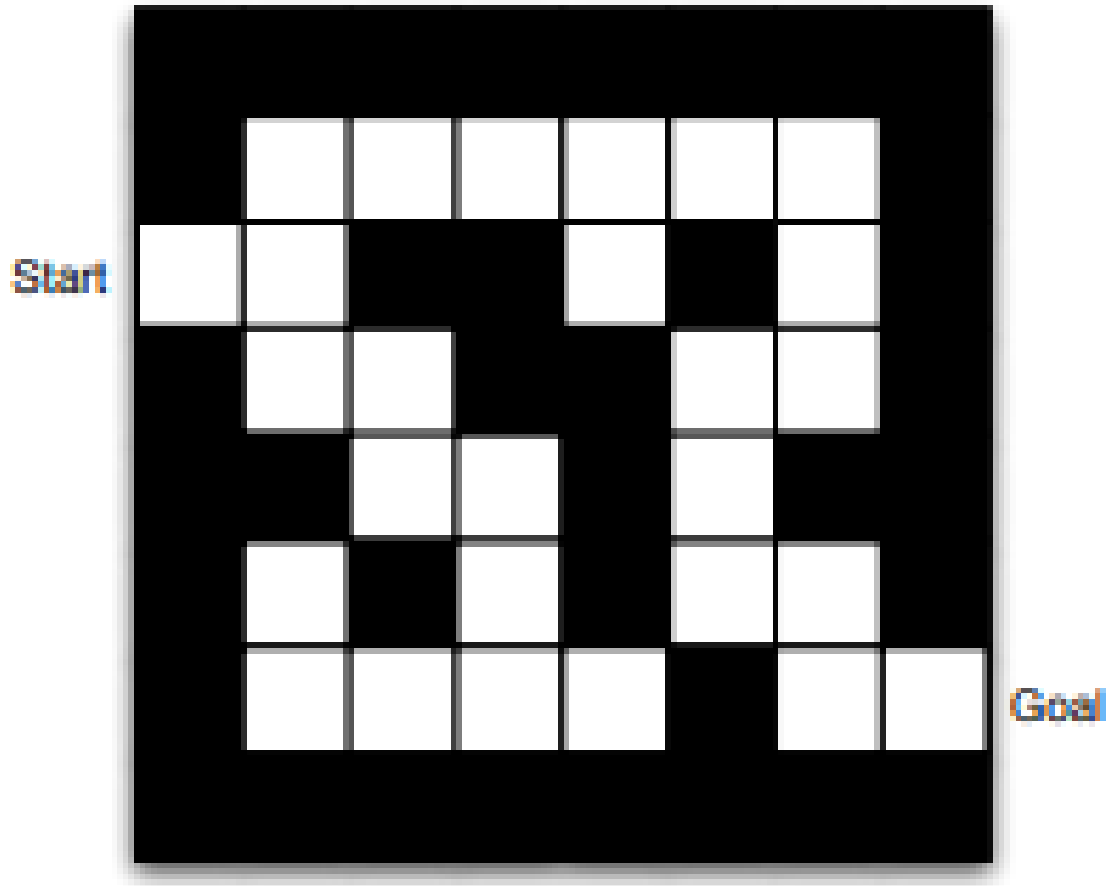
VALUE FUNCTION

- Value Function defines how good is each state and/or action
- Used to evaluate the goodness/badness of states
- And therefore to select between actions

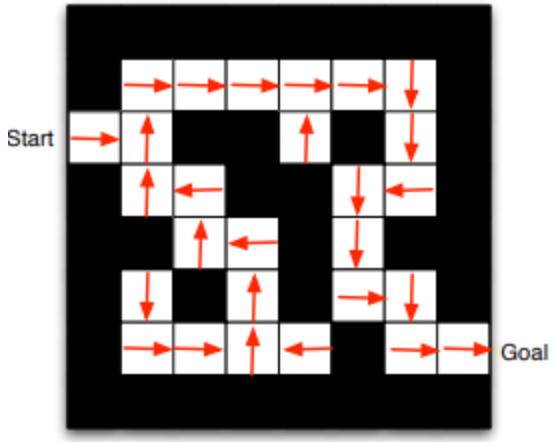
MODEL

- Model is the agent's representation of the environment.
- P predicts the next state
- R predicts the next (immediate) reward

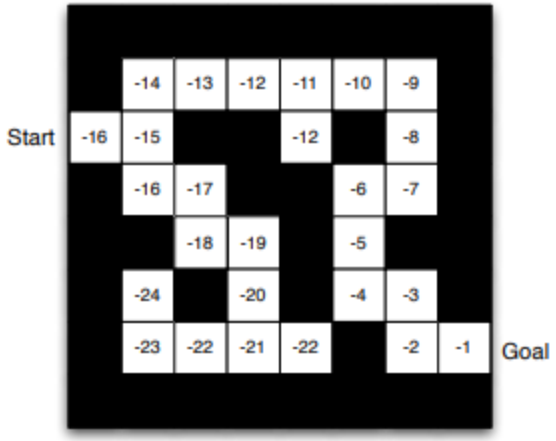
MAZE EXAMPLE



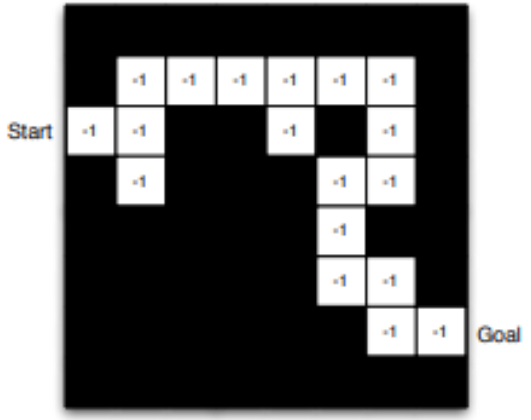
Policy



Value function



Model

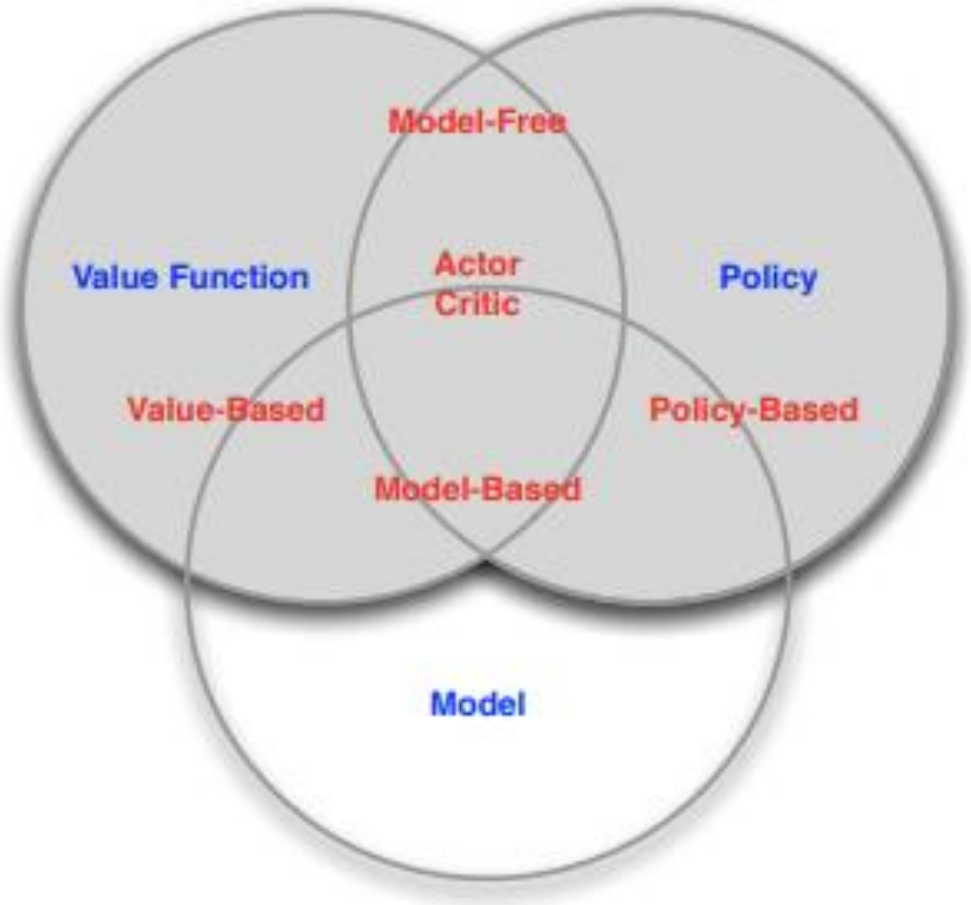


CATEGORIZING RL AGENTS

- Value Based
  No Policy (Implicit)
  Value Function
- Policy Based
  Policy
  No Value Function
- Actor Critic
  Policy
  Value Function

- Model Free
  Policy and/or Value Function
  No Model
- Model Based
  Policy and/or Value Function
  Model

RL Agent Classification



EXPLORATION AND EXPLOITATION

- Exploration finds more information about the environment Exploitation exploits known information to maximize reward

- Example:

  Game Playing Exploitation Play the move you believe is best Exploration Play an experimental move

APPLICATIONS:

- Traffic light control
- Robotics
- Chemistry
- Personalized recommendations
- Games: Atari games
- Deep learning

WHEN NOT TO USE REINFORCEMENT LEARNING?

- When you have enough data to solve the problem with a supervised learning method
- When the action space is large because Reinforcement Learning is computing-heavy and time-consuming.

THANK YOU