INTRODUCTION TO MACHINE LEARNING

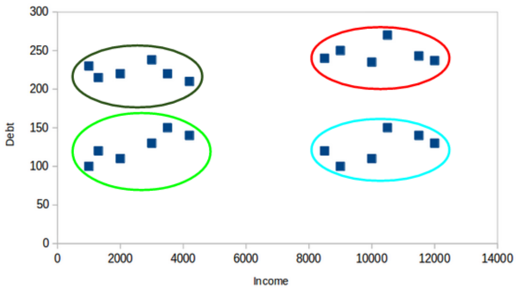
RBT 350

Roberto Martin-Martin

Assistant Professor of Computer Science.

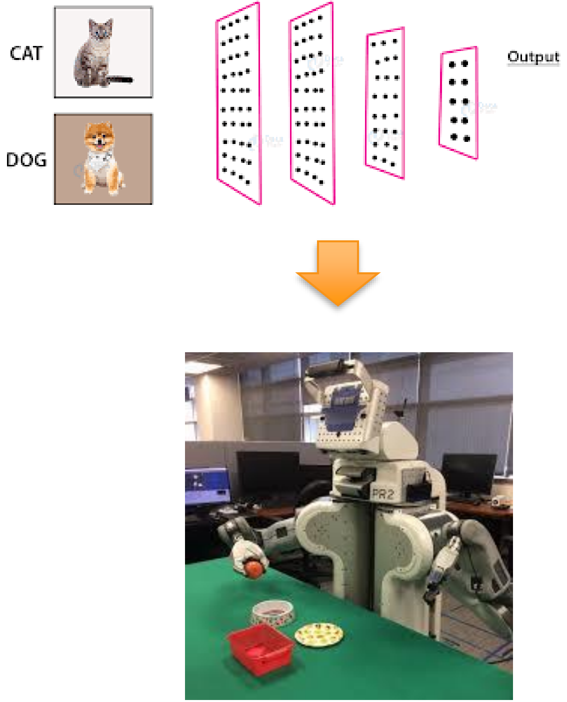
What will you learn today?

- Introduction to Machine Learning
  - Imitation Learning in Robotics
  - Clustering (K-Means)
  - Reinforcement Learning



Imitation Learning in Robotics

- Supervised learning in ML:
  - Given a large set of annotated data pairs (x, y) train a model that given new data x’ predicts the right y’
- Applying this to robotics:
  - Given a large set of data pairs (SensorSignal, Action) train a policy that given new SensorSignals x’ predicts the right action y’



Imitation Learning in Robotics

collect data

train model

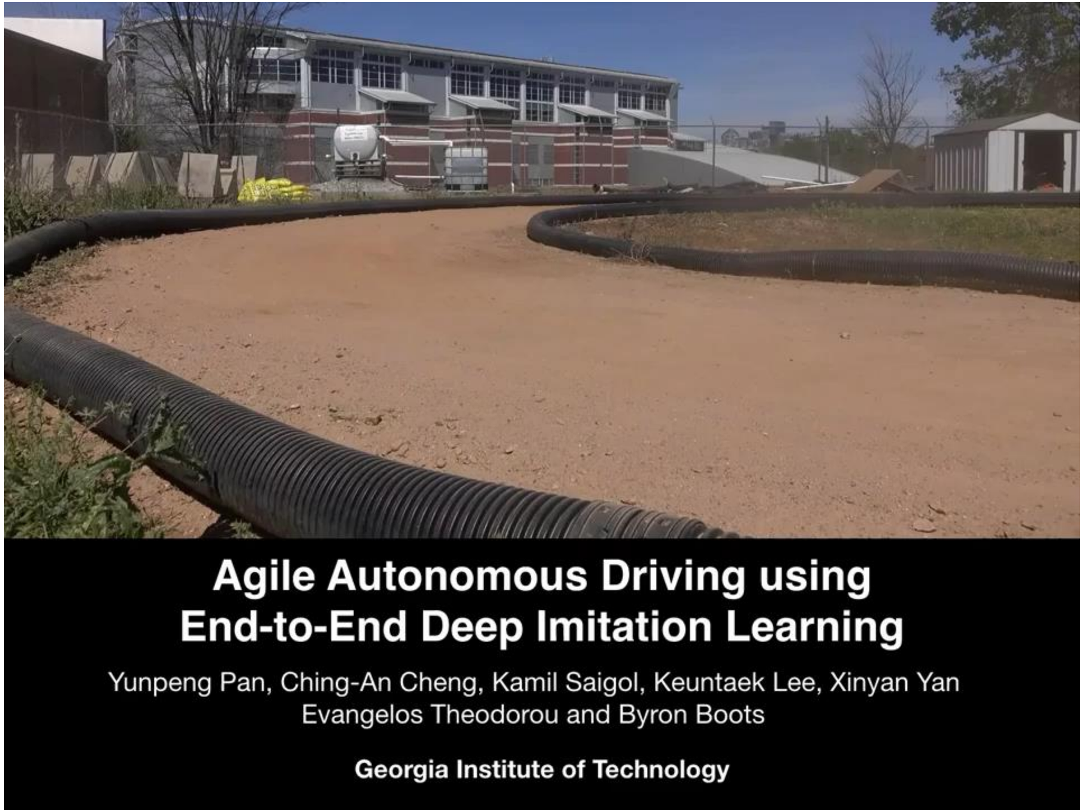
$$\pi(image) = action$$

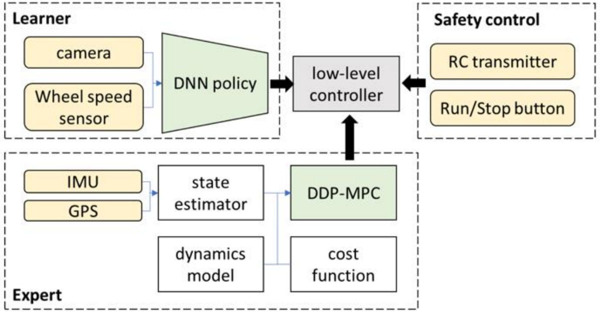
[(image, action)]

image → π → action

human teleoperation (demonstration)

real-time





"Agile Autonomous Driving using End-to-End Deep Imitation Learning" Pan, Cheng, Saigol, Lee, Yan, Theodorou, Boots. RSS 2018

Types of Machine Learning Problems

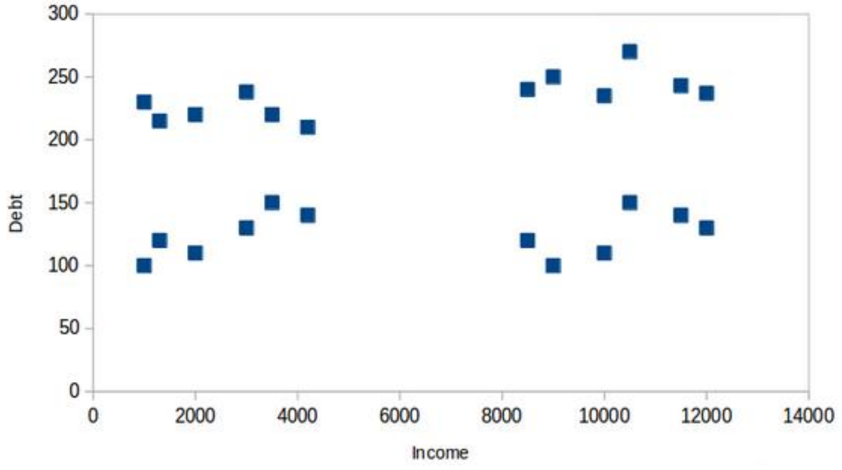
- Supervised Learning
  - Algorithm learns from a dataset of annotated cases (,  ”)
  - Regression
  - Classification

- **Unsupervised Learning**
  - **No annotated data is provided**
  - **Clustering**

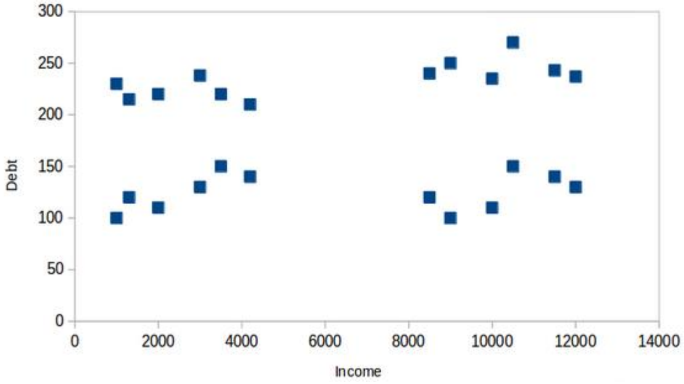
- Reinforcement Learning
  - Agent finds its own data → learns from successes and failures
  - Decision making in Markov Decision Processes (MDP)

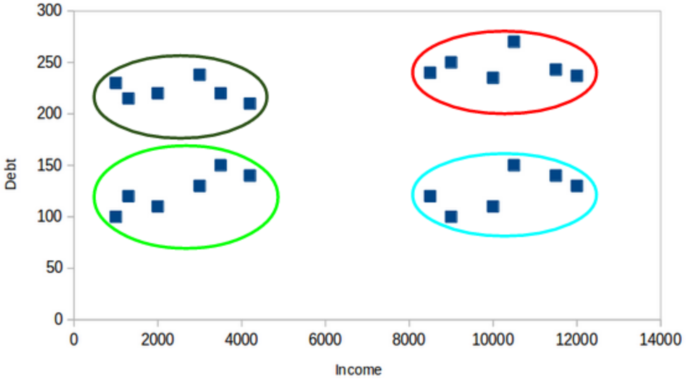
k-Means

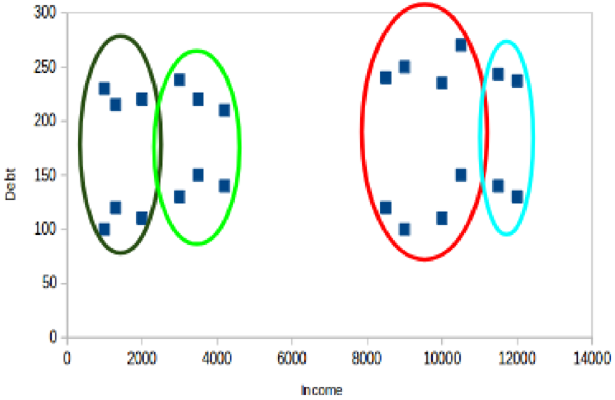
- Unsupervised ML technique → No labels!
- It clusters the data
  - Creates groups based on patterns in the data
    - Groups together items in the data that are *similar* to each other
    - Groups separately items in the data that are *different* to each other



Clustering







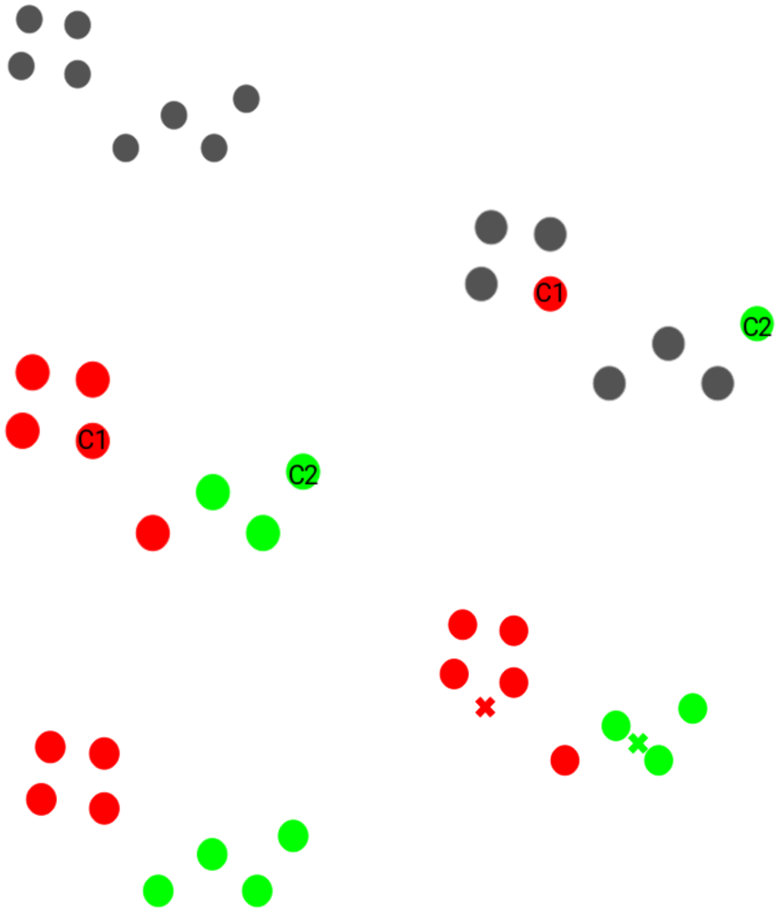
- Groups together items in the data that are *similar* to each other
- Groups separately items in the data that are *different* to each other

Applications of Clustering

- Recommendations systems!
- Image segmentation (not so much nowadays)

k-Means

- Iterative process (repeats, improving at each step)
- Works by minimizing intra-cluster distance (inertia)
- Process
  1. Choose number of clusters k
  2. Select k random data points as initial cluster centroids
  3. Assign the other points to clusters based on distance
  4. Recompute cluster centroids
  5. Repeat 3 and 4 until X



k-Means Stopping Criteria

- To be defined by the user
- Some options:
  - Centroids of newly formed clusters do not change
  - Points remain in the same cluster
  - Maximum number of iterations is reached

https://pollev.com/robertomartinmartin739

Exercise

- Imagine we have a dataset with values:
  - p1=(2,3)
  - p2=(7,9)
  - p3=(-1,0)
  - p4=(9,9)
- Assuming k=2
- Assuming centroid1=p1=(2,3), centroid2=p3=(-1,0)
- Assuming Euclidean distance

Types of Machine Learning Problems

- Supervised Learning
  - Algorithm learns from a dataset of annotated cases (,  ”)
  - Regression
  - Classification

- **Unsupervised Learning**
  - **No annotated data is provided**
  - **Clustering**

- Reinforcement Learning
  - Agent finds its own data → learns from successes and failures
  - Decision making in Markov Decision Processes (MDP)

Mathematical Framework: Markov Decision Processes

A **Markov Decision Process** is defined by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

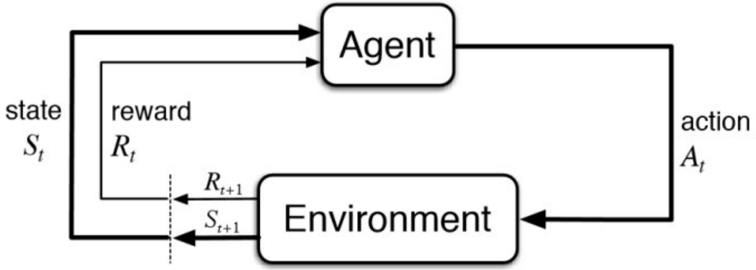
$\mathcal{S}$: state space $(s_t \in \mathcal{S})$

$\mathcal{A}$: action space $(a_t \in \mathcal{A})$

$\mathcal{P}$: transition probability $\mathcal{P}^a_{ss'} = \Pr[s_{t+1} \mid s_t, a_t]$

$\mathcal{R}$: reward function $r(s,a) = \mathbb{E}[r_{t+1} | s = s_t, a = a_t]$

γ : a discount factor $\gamma \in [0,1]$



Mathematical Framework: Markov Decision Processes

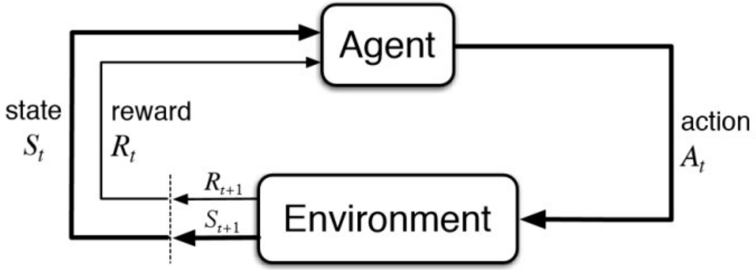
A **policy** maps states to actions $\pi : \mathcal{S} \rightarrow \mathcal{A}$

Goal of (robot) decision making

Choose policy that maximizes cumulative reward

$$\pi^* = \arg\max_{\pi} \mathbb{E}[\sum_{t \geq 0} \gamma^t r(s_t, \pi(s_t))]$$

Problem: the outcome of an action is
1) not known a priori
2) a bit stochastic

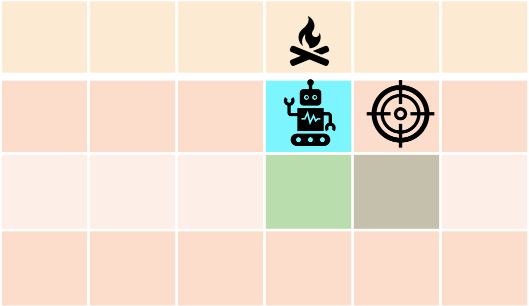


Example

$$\mathcal{P}_{ss'}^{a} = \Pr[s_{t+1} \mid s_t, a_t]$$

r()= 100

r()= −1000

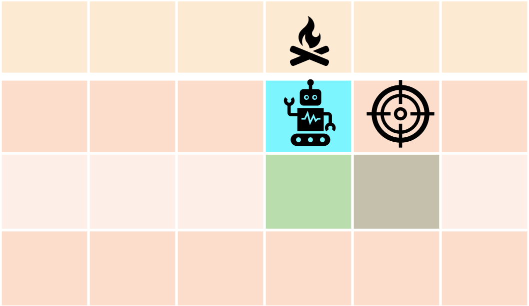


Example

$$\mathcal{P}_{ss'}^{a} = \Pr[s_{t+1} \mid s_t, a_t]$$

r()$= 100$

r()$= -1000$



$$P(s' = right | s, a = move_{right}) = 0.1$$

$$P(s' = up | s, a = move_{right}) = 0.9$$

$$P(s' = down | s, a = move_{down}) = 1$$

$$P(s' = up | s, a = move_{up}) = 1$$

$\pi^*($ $) = move_{down}$

$\pi^*($ $) = move_{right}$

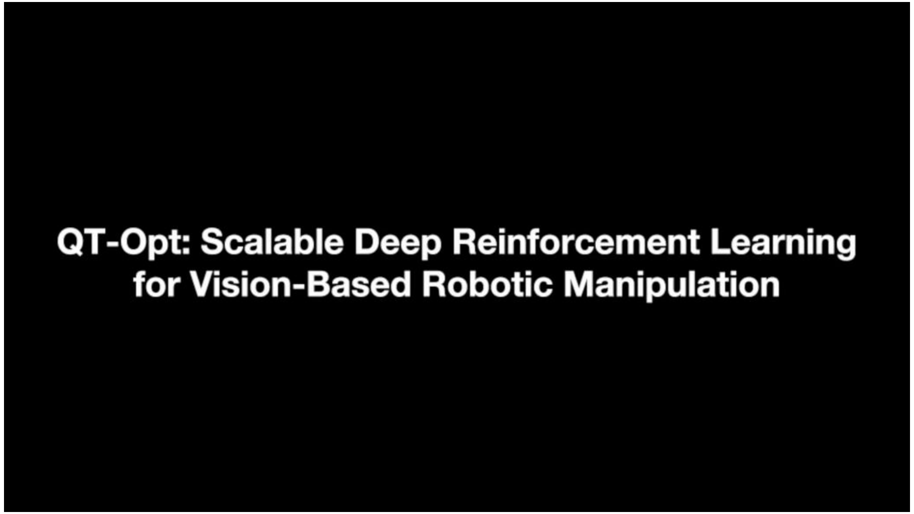
$\pi^*($  $) = move_{up}$

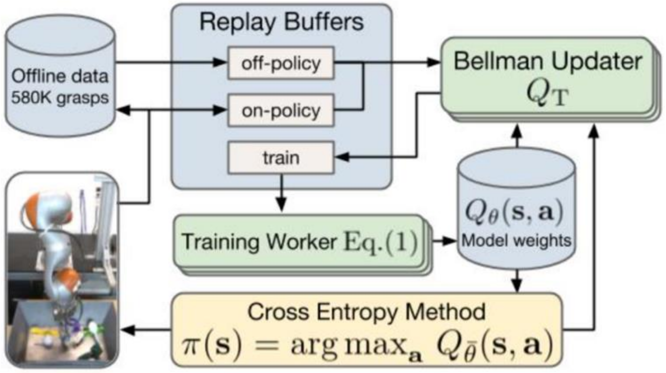
Recursion: the Bellman Equation

Bellman equation:

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma V^\pi(s')]$$

Examples of **Model-Free Reinforcement Learning**





"QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation." Kalashnikov et al. CoRL 2018

Recap

- What is Machine Learning?
- Types of Machine Learning problems
- Elements of a Machine Learning problem
- Supervised learning
  - Regression
    - Least squares
  - Classification
    - KNN, Decision Trees, DNNs (for more than classification!)
- Unsupervised learning → K-Means
- Imitation Learning in robotics
- Reinforcement Learning