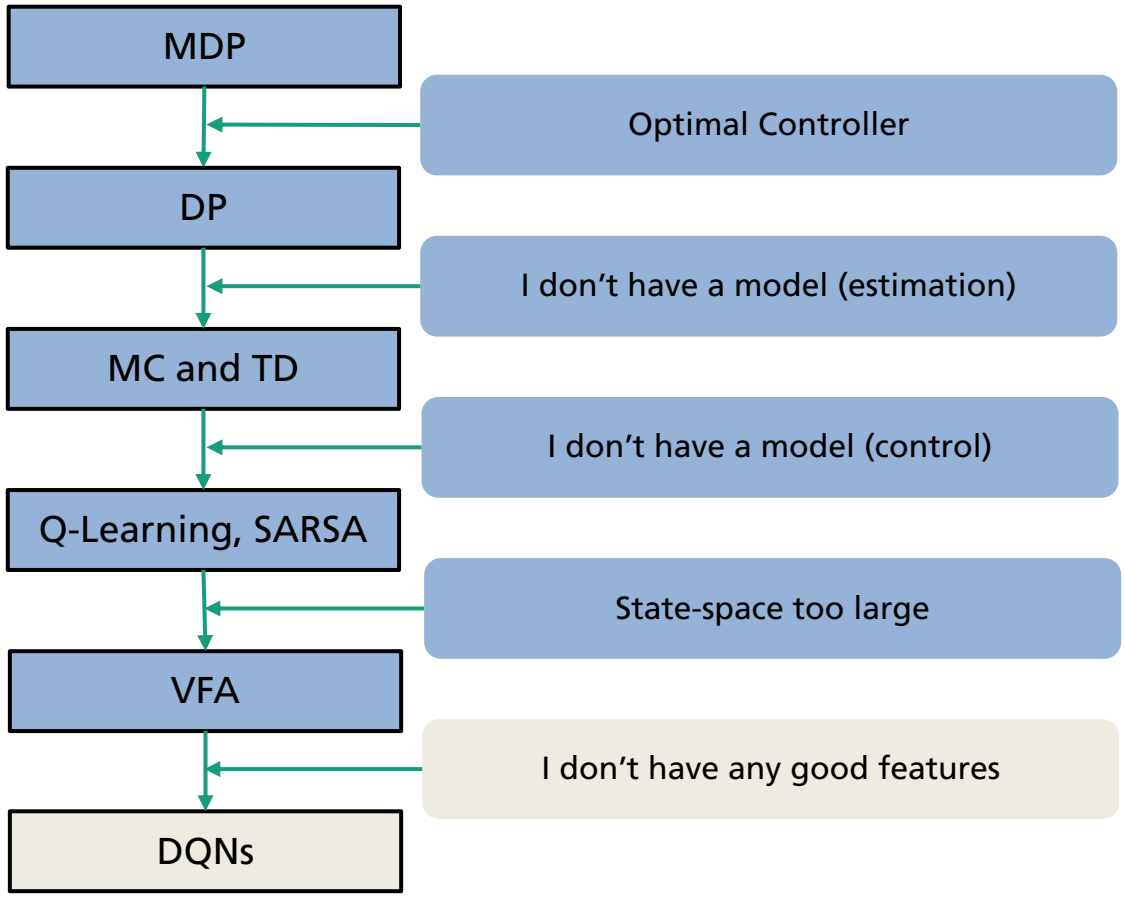
Fraunhofer IIS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Value Function Approximation

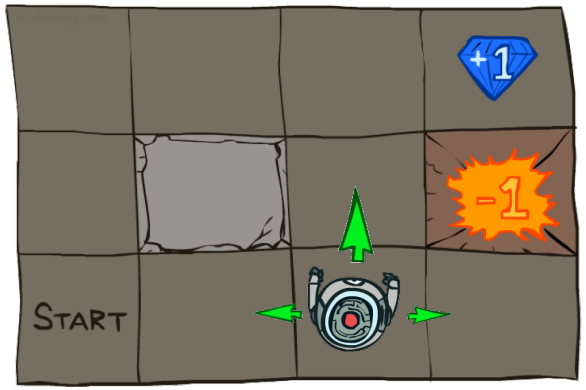
Christopher Mutschler

Overview

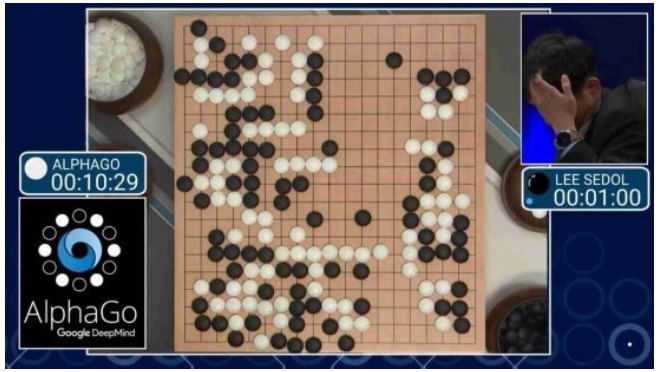

MDP
Optimal Controller
DP
I don’t have a model (estimation)
MC and TD
I don’t have a model (control)
Q-Learning, SARSA
State-space too large
VFA
I don’t have any good features
DQNs

Value Function Approximation

- Challenge #1: In real world problems, the state space can be large
  - Backgammon: 10^{20} states
  - Computer Go: 10^{170} states
  - Robot arm: **infinite** number of states! (continuous)
- Problems with large MDPs:
  - There are too many states and/or actions to store in memory
  - It is too slow to learn the value of each state individually



http://ai.berkeley.edu/lecture_slides.html

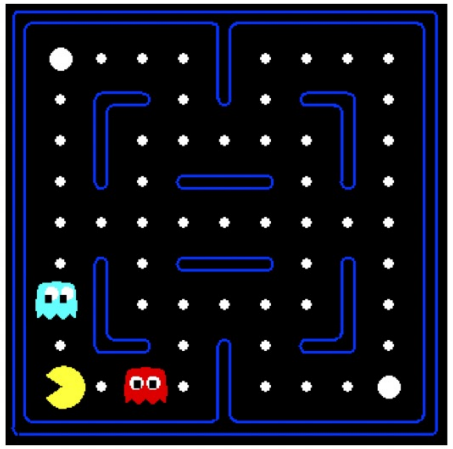


https://www.youtube.com/watch?v=HT-UZkiOLv8

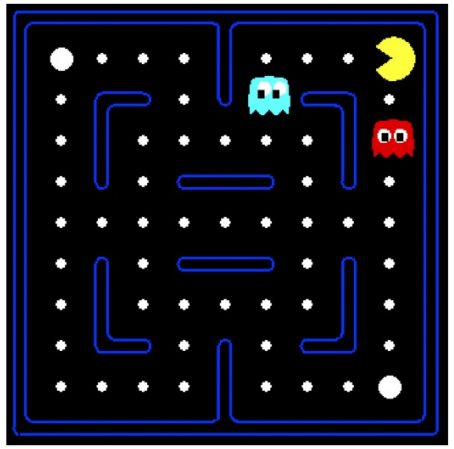
Value Function Approximation

- Challenge #2: Generalization across states

Let‘s say we discover through experience that this state is bad:



In naive Q-learning we know nothing about this state:



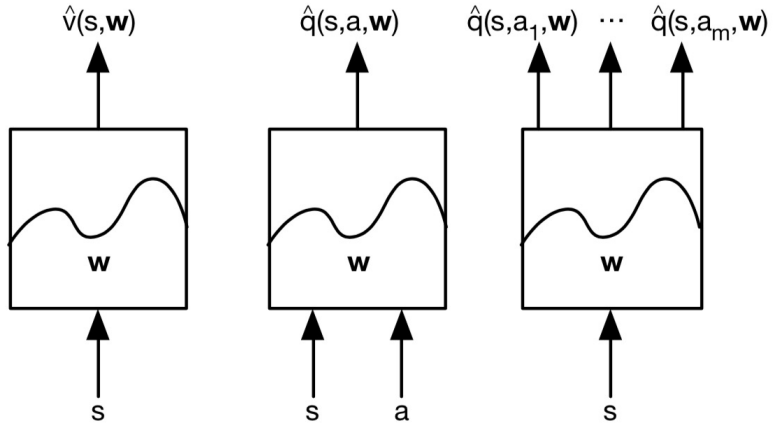
Or even this one:

Pieter Abbeel: CS 188 Introduction to Artificial Intelligence. Fall 2018

Value Function Approximation

- Value Function Representations
- Exact:
  - A table with a distinct value for each case
  - V: one entry per s
  - Q: one entry for each (s, a) pair
- Approximate:
  - Approximate V or Q with a function approximator
    (e.g., NN, polynomials, RBF, …)

$$\hat{v}(s, \boldsymbol{w}) \approx v_\pi(s)$$
$$\hat{q}(s, a, \boldsymbol{w}) \approx q_\pi(s, a)$$



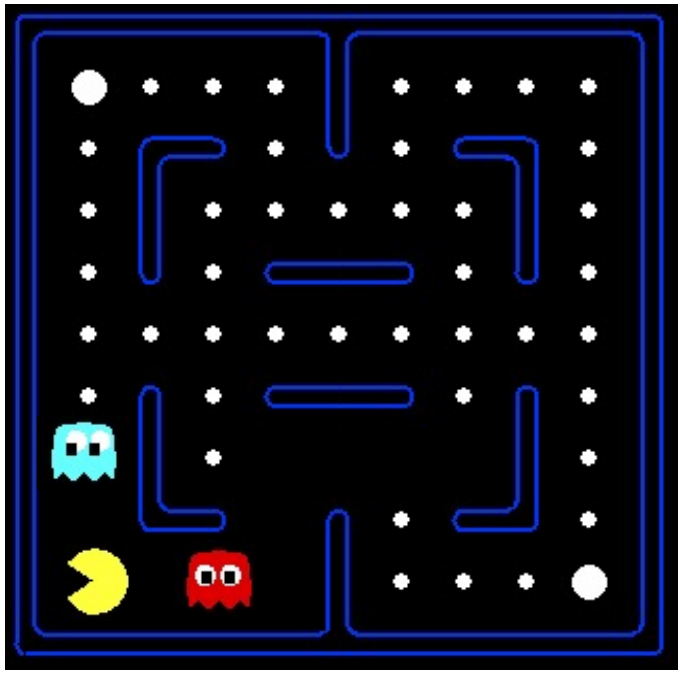
David Silver. 2016.

+ We need only to store the approximator parameters

- Convergence properties do not hold anymore

Value Function Approximation

- VFA: Describe a state using a vector of features
- Features are functions from states to real numbers that capture important properties of the state
- Example features for Pac Man:
  - Distance to closest ghost
  - Distance to closest dot
  - Number of ghosts
  - …



Value Function Approximation

- Our goal is to learn good parameters w that approximate the true value function well:

$$C = \left(Q^+(s,a) - \hat{Q}^\pi(s,a;w)\right)^2$$
$$= (Q^+(s,a) - \phi(s,a)^T w)^2$$

$$\rightarrow \quad \frac{\partial C}{\partial w} = -2\,\phi(s,a)(Q^+(s,a) - \phi(s,a)^T w)$$

$$\hookrightarrow \quad w \leftarrow w - \eta \frac{\partial C}{\partial w}$$

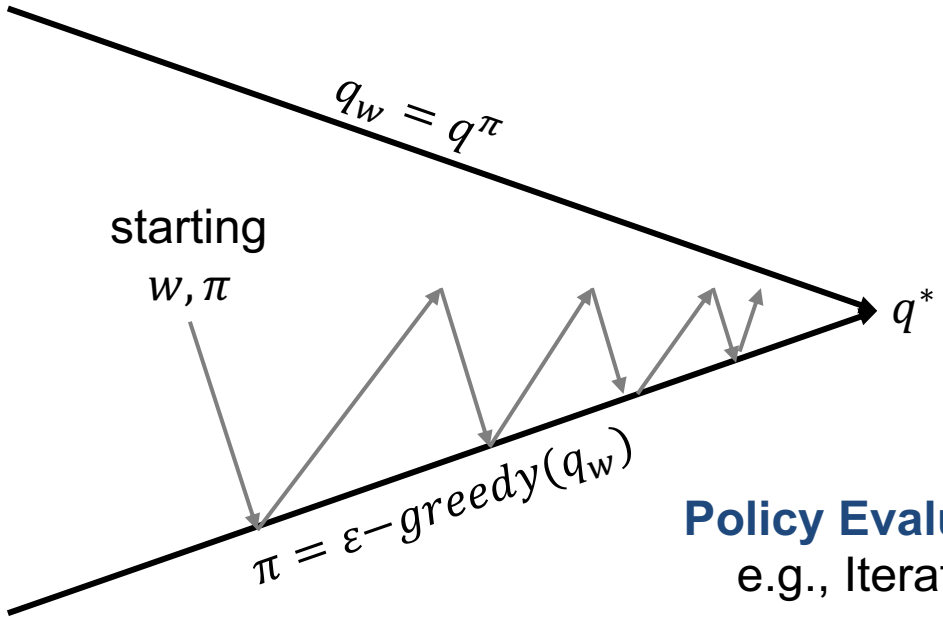
$$w \leftarrow w + 2\,\eta\,\phi(s,a)\left(Q^+(s,a) - \hat{Q}^\pi(s,a;w)\right)$$

$Q^+(s,a) =$

$r + \gamma\, max_{a'} Q(s',a')$	Q-Learning with Linear VFA
$r + \gamma\, Q(s',a')$	SARSA with Linear VFA
G_t	MC with Linear VFA

Value Function Approximation

- Our goal is to learn good parameters w that approximate the true value function well:

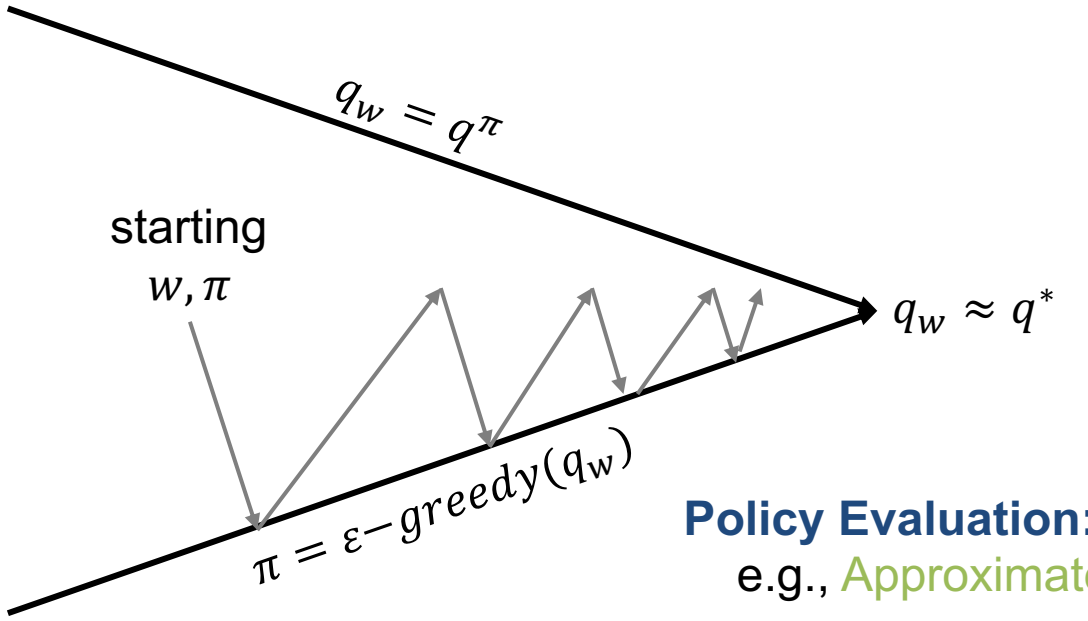


Policy Evaluation: Estimate q_π
e.g., Iterative Policy Evaluation

Policy Improvement: Generate $\pi' \geq \pi$
e.g., ϵ-greedy Policy Improvement

Value Function Approximation

- Our goal is to learn good parameters w that approximate the true value function well:



Policy Evaluation: Estimate $\hat{q}(\cdot,\cdot,w) \approx q_\pi$
e.g., Approximate Policy Evaluation

Policy Improvement: Generate $\pi' \geq \pi$
e.g., ϵ-greedy Policy Improvement