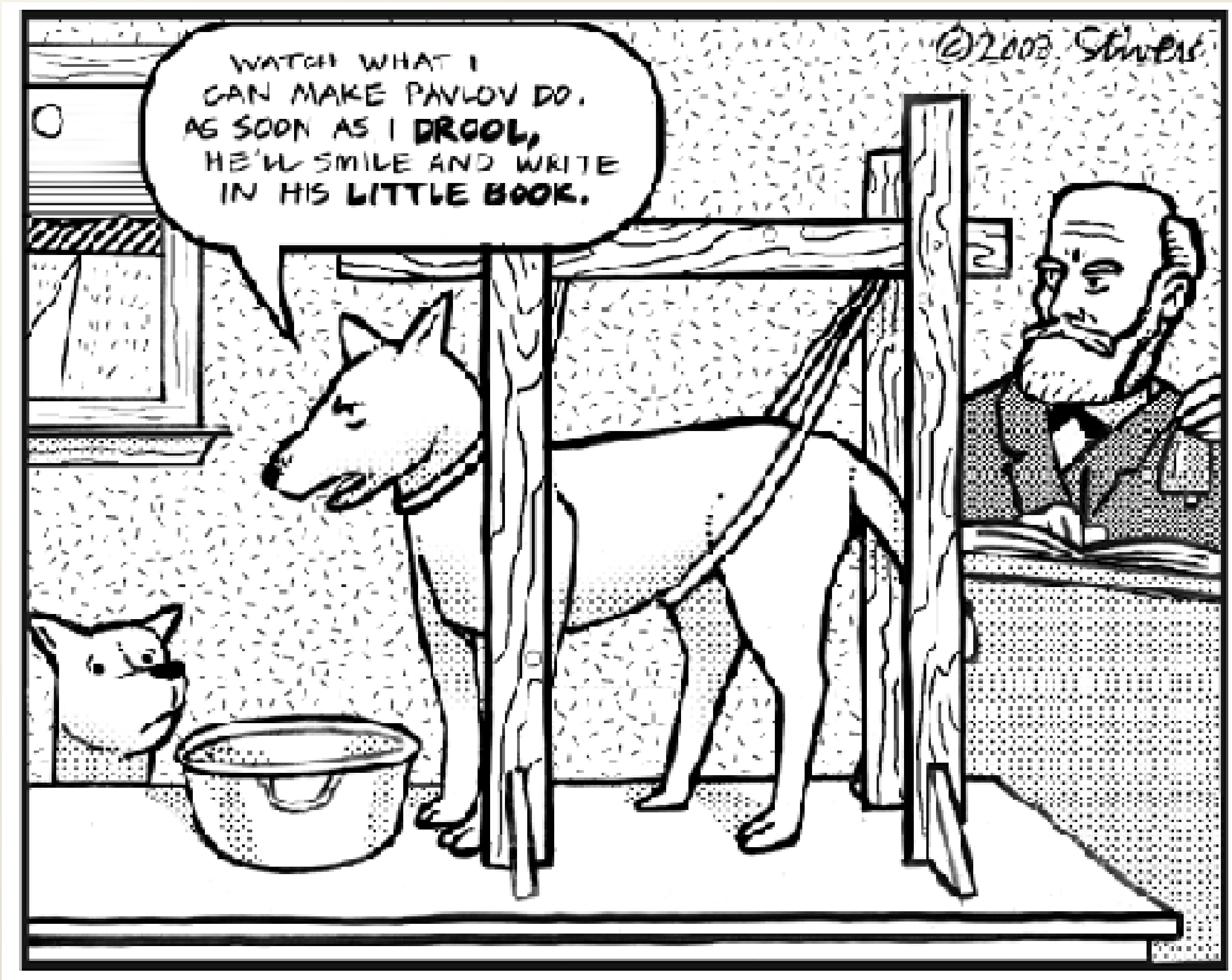
MARL: Multiagent Reinforcement Learning

Minwoo Jake Lee
@INRIA

Agenda

- Reinforcement Learning
  - Brief introduction
  - Function approximation
- Multiagent Learning
  - Multiagent Reinforcement Learning
  - Issues
- Current Work
- Future Work

What is Reinforcement Learning?

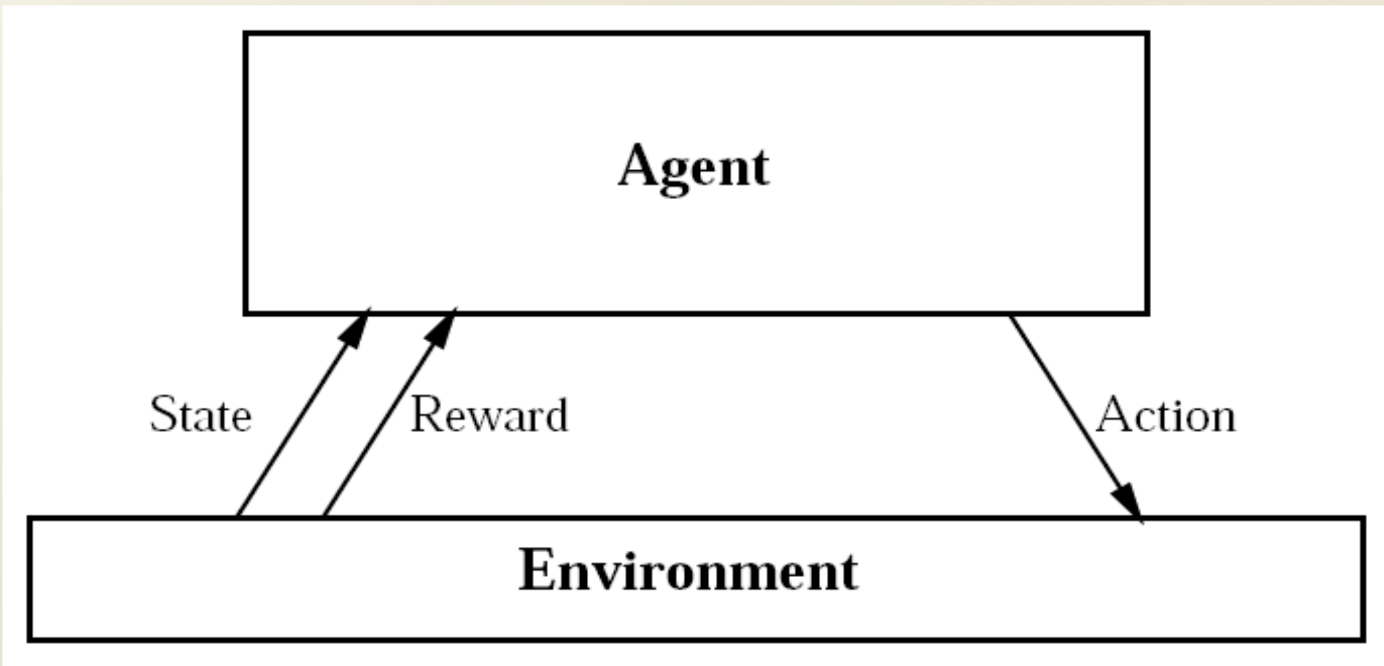


Reinforcement Learning

- Control Learning (Tom Mitchell)
  - Consider learning to choose actions
    - Learning to play Backgammon
    - Robot learning to dock on battery charger
    - Learning to control wind turbines for max power generation
- Learning Algorithms
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning?

Reinforcement Learning

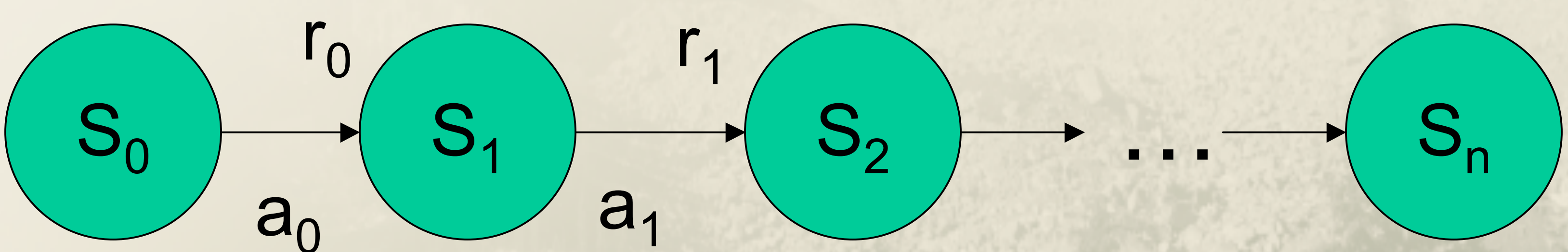
- No label but rewards info.
  - Delayed Rewards
  - Partially observable or noisy state reading



Goal: Reinforcement Learning

Goal: Learn to choose actions that maximizes

$$R = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$$



Values of States/Actions

- Not available
- Dynamic Programming
  - Require knowledge of transition probabilities
- Learning from Samples
  - Monte Carlo
  - Temporal Difference (TD)

Value Function

- Value Function

$$V^{\pi}(s) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$$
$$\equiv \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

$$\pi^* \equiv \operatorname*{argmax}_{\pi} V^{\pi}(s), (\forall s)$$

- Q Function

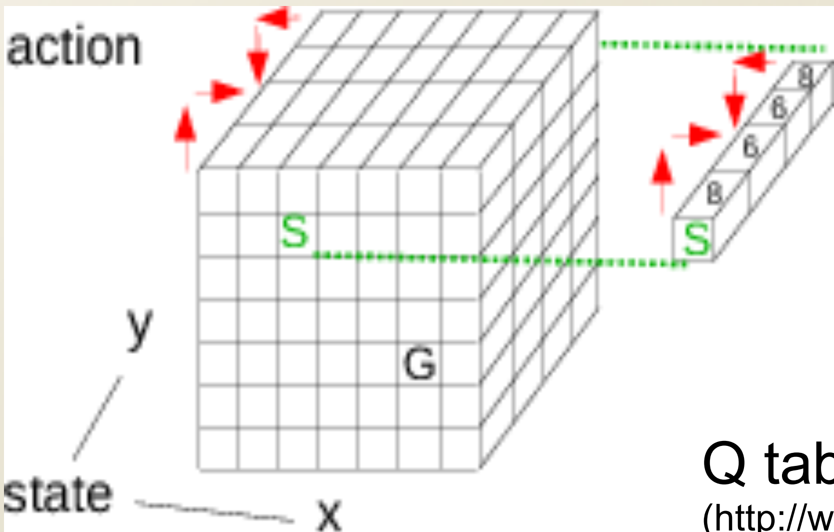
$$V^*(s) = \max_{a'} Q(s, a')$$

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma V^*(\delta(s_t, a_t)))$$
$$= r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')$$

$$\pi^*(s) = \operatorname*{argmax}_{a} Q(s, a)$$

Tabular Q

- Q table for low dimensional discrete state/action problems
- Impossible to apply in practical use



Q table for Maze problem
(http://www.cs.colostate.edu/cs545)

Function Approximation

- Approximate Q function as accurate as possible

$$Q(s_t, a_t) \approx \sum_{k=0}^{\infty} r_{t+k+1}$$

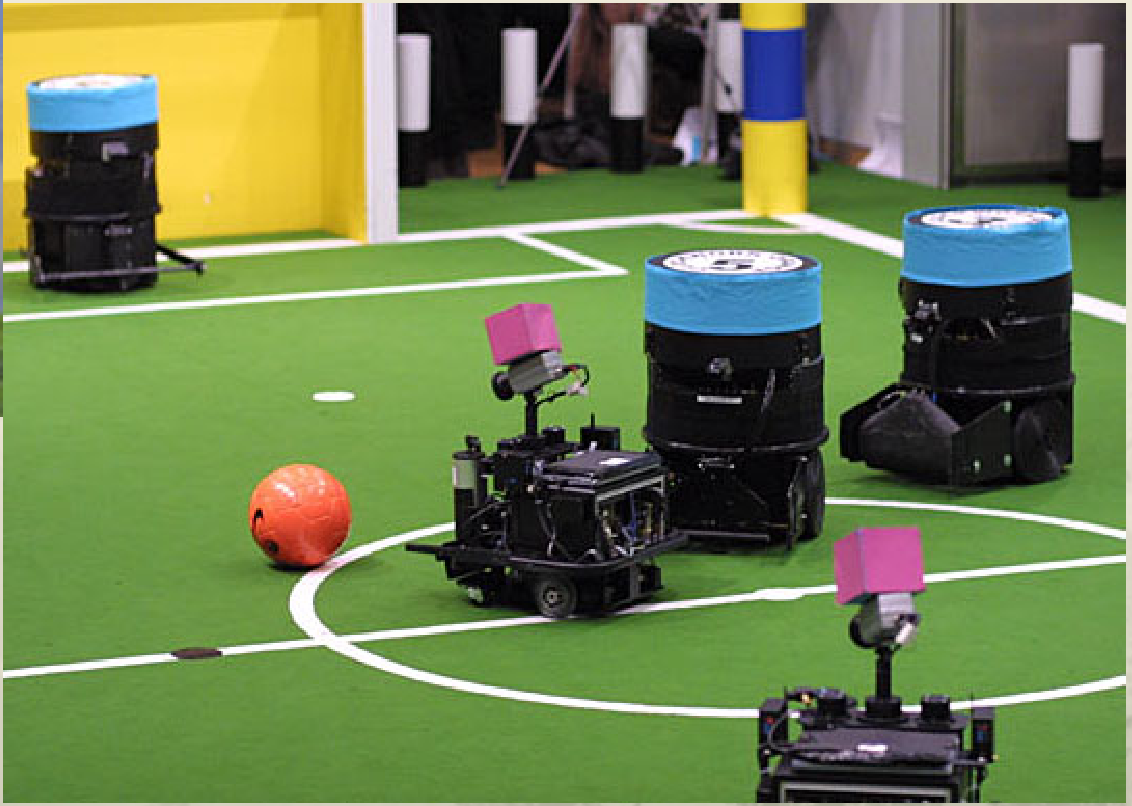
- Equivalently,

$$\text{Minimize } \mathrm{E}\left(\sum_{k=0}^{\infty} r_{t+k+1} - Q(s_t, a_t)\right)^2$$

- Regression
  - Neural Network / Radial Basis Function / SVR
  - Deep Architecture / Distance Weighted Discrimination

Multiagent Systems

http://www.mlplatform.nl



http://www.cs.cmu.edu/~sross1

Multiagent Reinforcement Learning

- Multiagent Systems
  - Decentralized way to solve complex problems
- Multiagent Learning
  - Automate the inductive process
  - Discover solutions on its own
- MARL
  - Simplicity: Model-free learning
  - Learn efficiently by trial-and-error

MARL: Issues

- Exponential growth of search space
  - Complexity of environment (problem)
  - The number of agents
  - Dynamic environment/agents
- In literature,
  - Function approximation
  - Decentralization

Recent Work

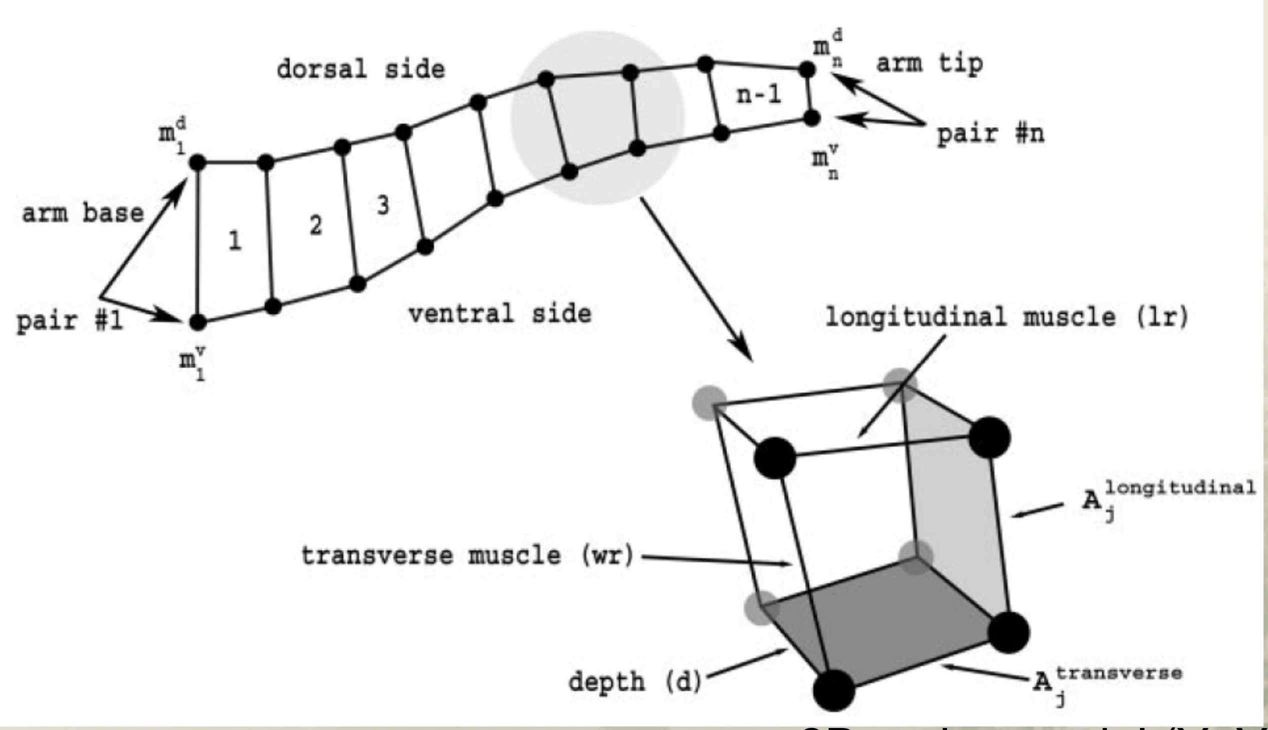
- Adopting heuristic functions to reduce the dimensionality

$$
\begin{aligned}
Q^j\left(s_t^j, a_t^j, \mathcal{H}^j(\vec{s}_t)\right) \leftarrow & Q^j\left(s_t^j, a_t^j, \mathcal{H}^j(\vec{s}_t)\right) + \\
& \alpha\Big[r_{t+1}^j + \gamma Q^j\left(s_{t+1}^j, a_{t+1}^j, \mathcal{H}^j(\vec{s}_{t+1})\right) \\
& - Q^j\left(s_t^j, a_t^j, \mathcal{H}^j(\vec{s}_t)\right)\Big]
\end{aligned}
$$

- Heterogeneous learning
  - Decomposition approach
  - Various algorithmic combination to improve adaptability

Octopus Arm

- Muscular hydrostat structure
  - consists of packed array of muscle bers in 3 main directions
  - maintains constant volume
  - forces are transferred between the longitudinal and the transverse directions



2D spring model (Y. Yekutieli et al.)

Future Work

- Currently, heterogeneous cooperative learning
  - Hierarchical Reinforcement Learning Framework
- Robust High Dimensional Regression
  - Directly apply on the function approximation
- Robust High Dimensional Clustering
  - To reduce dimensionality of states and actions
  - Reinforcement learning over abstract clusters of states and actions