Reinforcement Learning for Cognitive Radio Networks

BWN Lab Workshop'09

Brandon F. Lo

Broadband Wireless Networking Lab
School of Electrical and Computer Engineering
Georgia Institute of Technology

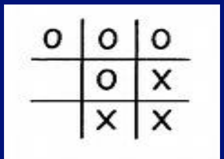
Outline

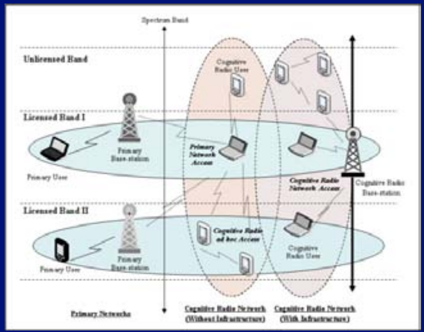
- **Reinforcement Learning (RL) Preliminaries**
  - Temporal-difference learning
  - Applications of RL to Cognitive Radio (CR) Networks

- **Multi-agent Reinforcement Learning (MARL)**
  - Fully cooperative tasks
  - Fully competitive tasks
  - Mixed tasks

What is Reinforcement Learning?

- **A Branch of Machine Learning**
  - Computational method for a decision-making learner (agent) to:
    - Sense and act in its environment
    - Learn to choose optimal actions to achieve its goal
  - Also known as:
    - Approximate dynamic programming
    - Neuro-dynamic programming
  - Applications



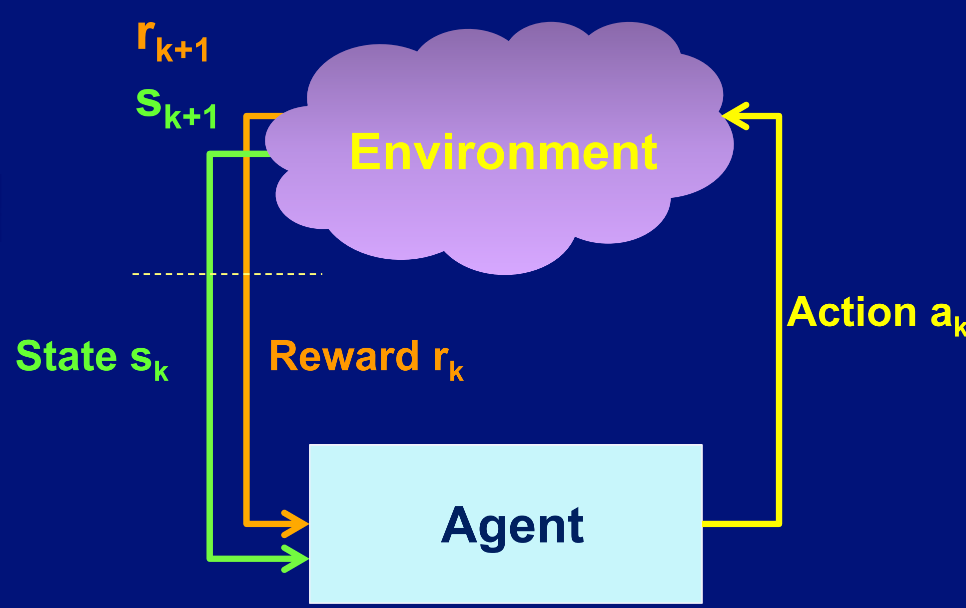


Anatomy of Reinforcement Learning

- **Markov Decision Process**
  - A quadruple: <*S*, *A*, *f*, *ρ*>
    - *S*: set of all states
    - *A*: set of all actions
    - *f*: transition probability function
      f: *S* x *A* x *S* → [0,1]
    - *ρ*: reward function
      ρ: *S* x *A* x *S* → R
- **Objectives**
  - Find optimal policy π*: S → A
  - Maximize discounted cumulative reward

$$R_k = \sum_{n=0}^{\infty} \gamma^n r_{k+n+1}$$

Markov Decision Process



Exploration vs. Exploitation

- Exploration
  - Explore the unknown states to achieve potentially higher cumulative reward
- Exploitation
  - Exploit the current knowledge of best actions to receive potentially highest immediate reward

Action Selection Strategy for Exploration vs. Exploitation

- **Softmax (Boltzmann) Selection Strategy**
  - Probability of selecting action a_i in state s with action-value function $Q(s,a_i)$:

$$p(s,a_i) = \frac{e^{Q(s,a_i)/T}}{\sum_j e^{Q(s,a_j)/T}}$$

- T: temperature: making tradeoff between exploration and exploitation
- Explore with large T: all actions are equally probable
- Exploit with small T: the action with maximum Q(s,a) value is favored

Temporal Difference Learning Methods

- **Q Learning**
  - Off-policy TD method
    - Policy for making decisions and policy to be improved are separate
- **Sarsa**
  - On-policy TD method
    - The policy to be improved is also used in determining actions
- **Actor-Critic**
  - Always on-policy
  - Agent consists of an actor and a critic
    - Actor: action selection and policy updates
    - Critic: state value function estimation and updates

Temporal-Difference (TD) Methods: Q-Learning

- Q-Learning
  - Off-policy TD method
    - Policy for making decisions and policy to be improved are separate
  - Algorithm:
    - Initialize Q(s,a) and repeat the following for each episode:
    - Repeat the following until s is terminal:
    - Choose and take action a_k, observe r_{k+1}, s_{k+1}
    - Action value update

$$
\begin{aligned}
Q(s_k, a_k) &\leftarrow Q(s_k, a_k) + \alpha[r_{k+1} + \gamma \max_{a \in A} Q(s_{k+1}, a) - Q(s_k, a_k)] \\
&= (1-\alpha) Q(s_k, a_k) + \alpha[r_{k+1} + \gamma \max_{a} Q(s_{k+1}, a)]
\end{aligned}
$$

    - State update: $s_k \leftarrow s_{k+1}$

Temporal-Difference (TD) Methods: Sarsa

- Sarsa $(s_k, a_k, r_{k+1}, s_{k+1}, a_{k+1})$
  - On-policy TD method
    - The policy to be improved is also used in making decisions
  - Algorithm:
    - Initialize Q(s,a) and repeat the following for each episode:
    - Choose action a_k and repeat until s is terminal:
    - Take action a_k and observe r_{k+1}, s_{k+1}
    - Choose a_{k+1} from s_{k+1} using action selection strategy
    - Action value update

$$Q(s_k, a_k) \leftarrow (1-\alpha)Q(s_k, a_k) + \alpha[r_{k+1} + \gamma Q(s_{k+1}, a_{k+1})]$$

    - State-action pair update: $(s_k, a_k) \leftarrow (s_{k+1}, a_{k+1})$

Temporal-Difference (TD) Methods: Actor-Critic Method

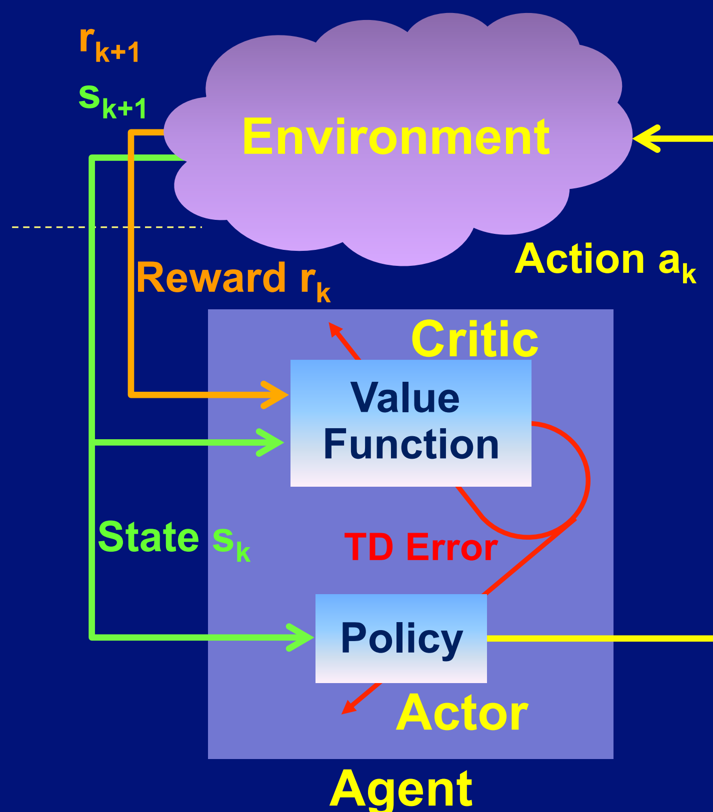
- **Actor-Critic Method**
  - Always on-policy
  - Critic: state value function estimation and update

$$V(s_k) \leftarrow V(s_k) + \beta\delta_k$$

- TD error: $$\delta_k = r_{k+1} + \gamma V(s_{k+1}) - V(s_k)$$

  - Actor: action selection and policy update

$$\pi(s,a_i) = \frac{e^{p(s,a_i)}}{\sum_j e^{p(s,a_j)}} \qquad p(s_k,a) \leftarrow p(s_k,a) + \beta\delta_k$$



Challenges of Reinforcement Learning

- Curse of Dynamic Programming
  - State and action spaces may grow exponentially
- Exploration-Exploitation Dilemma
  - Tradeoff between exploration and exploitation
- Convergence Problem
  - The algorithm should converge and converge fast
    - Related to memory, time, and energy costs

Properties for Convergence

- **Basic Conditions on Learning Rate**

$$(1)\ \alpha_n \geq 0, \qquad n = 0,1,\ldots$$

$$(2)\ \sum_{n=0}^{\infty} \alpha_n = \infty$$

$$(3)\ \sum_{n=0}^{\infty} (\alpha_n)^2 < \infty$$

  - (2) makes sure the algorithm does not stall prematurely
  - (3) guarantees the variance of the estimate of the optimal solution goes to zero in the limit
- **Requirement for Q-learning**
  - Each state-action pair must be visited infinitely often

Applications of RL to Cognitive Radio Networks

- Dynamic Channel Selection
- Spectral Resource Detection
- Cooperation
  - Cooperation reliability and security
  - Cooperative sensing

Multiagent Reinforcement Learning (MARL)

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 38, no.2, Mar. 2008.

- **Generalization of Markov Decision Process**
- **Stochastic Game:** $\langle S, A_1, \ldots, A_n, f, \rho_1, \ldots, \rho_n \rangle$
  - Joint action set: $A = A_1 \times \ldots \times A_n$
  - State transition probability function $f: S \times A \times S \rightarrow [0,1]$
  - Joint Policy $\Pi = \{\pi_i: S \times A_i \rightarrow [0,1]\}$
  - Q-function of each agent $Q_i^{\pi}: S \times A \rightarrow R$
    - Fully cooperative: agents have the same goal: $\rho_1 = \ldots = \rho_n$
    - Fully competitive: agents have opposite goals: $\rho_1 = -\rho_2$ for n=2

Goals of MARL

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Stability of Learning Process**
  - Convergence to an equilibrium (may not be Nash)
  - Prediction: agent's capability to learn nearly accurate models of other agents
- **Adaptation to Other Agents**
  - Rationality: the agent converges to a best response when other agents remain stationary
  - No-regret: the agent achieves a return that is at least as good as the return of any stationary strategy
    - This prevents an agent from "being exploited" by other agents

Benefits of MARL

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Experience Sharing**
  - Information exchange (cooperation)
  - Teacher for learners (training set)
  - Emulation
- **Inherent Robustness**
  - The remaining agents can take over the tasks when one or more agents fail
- **High Degree of Scalability**
  - New agent can be easily inserted into the system
- **Benefits can be challenges when some agents are malicious**

Challenges of MARL

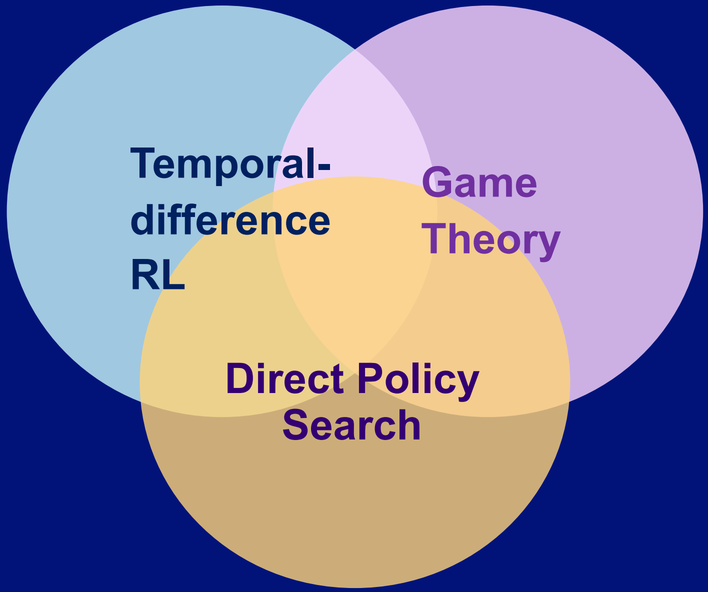
L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **All Challenges of Single-agent RL**
  - Curse of Dimensionality, exploration-exploitation dilemma, and Convergence
- **Nonstationarity**
  - Moving-target learning problem: the best policy changes as the other agents' policies change
  - Exploration strategy is crucial for stability and efficiency
- **The Need for Coordination**
  - Agents' choices must be mutually consistent
  - Coordination boils down to breaking ties between equally good strategies

Techniques in MARL Algorithms

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Techniques**
  - Temporal-difference reinforcement learning
  - Game theory
  - Direct policy search



Classification of MARL Algorithms

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Type of Tasks**
  - Fully Cooperative
  - Fully Competitive
  - Mixed

Task Type -> Agent Awareness	Cooperative	Competitive	Mixed
Independent	Coordination-free	Opponent-independent	Agent-independent
Tracking	Coordination-based	-	Agent-tracking
Aware	Indirect coordination	Opponent-aware	Agent-aware

Fully Cooperative Tasks

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

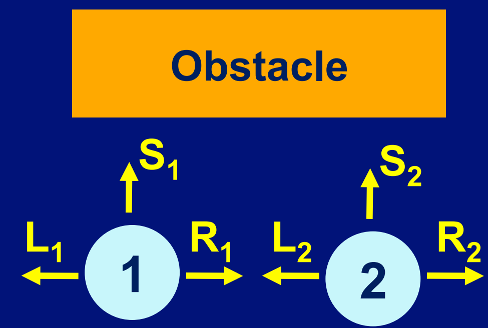
- **Fully Cooperative Stochastic Game**
  - Agents have the same reward function and learning goal
    - $\rho_1 = \ldots = \rho_n$
  - The goal is to maximize common discounted reward

- **The Need for Coordination**
  - Coordination-free methods are suboptimal

The Need for Coordination Example

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Two Mobile Agents**
  - Avoid the obstacle
  - Maintain their relative position
- **The tie between Optimal Joint Actions**
  - (L_1, L_2) and (R_1, R_2)
- **Suboptimal joint actions**
  - (L_1, R_2) and (R_1, L_2)

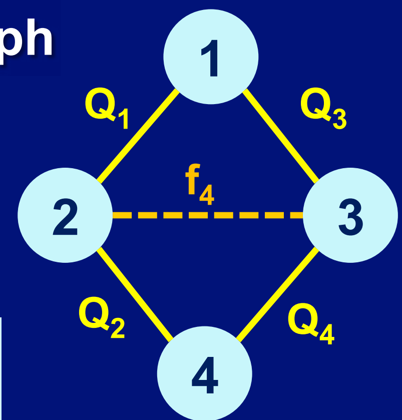


Q	L_2	S_2	R_2
L_1	10	-5	0
S_1	-5	-10	-5
R_1	-10	-5	10

Coordinated Reinforcement Learning

C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," in Proc. Int'l Conf. Machine Learning (ICML-02), Jul. 2002.

- **Cooperative Action Selection**
  - Exploit local structure thru coordination graph
  - Maximize over variables one at a time
  - Start with agent 4
    - Agent 4 communicates with agent 2 & 3



$$\max_{a_1,a_2,a_3,a_4} Q_1(a_1,a_2)+Q_2(a_2,a_4)+Q_3(a_1,a_3)+Q_4(a_3,a_4)$$

$$\rightarrow \max_{a_1,a_2,a_3} Q_1(a_1,a_2)+Q_3(a_1,a_3)+\max_{a_4}\left[Q_2(a_2,a_4)+Q_4(a_3,a_4)\right]$$

$$\rightarrow \max_{a_1,a_2,a_3} Q_1(a_1,a_2)+Q_3(a_1,a_3)+f_4(a_2,a_3)$$

Coordinated Reinforcement Learning

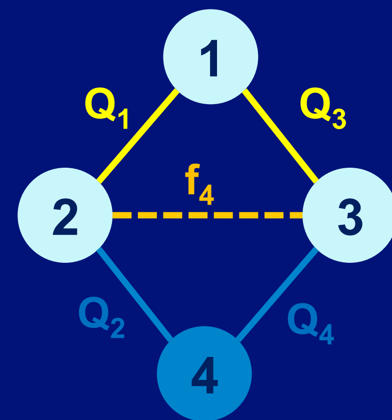
C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," *in Proc. Int'l Conf. Machine Learning (ICML-02),* Jul. 2002.

Cooperative Action Selection

- Agent 3:

$$\max_{a_1,a_2,a_3} Q_1(a_1,a_2)+Q_3(a_1,a_3)+f_4(a_2,a_3)$$
$$\rightarrow \max_{a_1,a_2} Q_1(a_1,a_2)+\max_{a_3}\left[Q_3(a_1,a_3)+f_4(a_2,a_3)\right]$$
$$\rightarrow \max_{a_1,a_2} Q_1(a_1,a_2)+f_3(a_1,a_2)$$

- Agent 1: $f_1(a_2)=\max_{a_1} Q_1(a_1,a_2)+f_3(a_1,a_2)$
- Agent 2: $f_2=\max_{a_2} f_1(a_2)$
- Recover maximizing set of actions in reverse
  - $f_2 \rightarrow a_2^* \rightarrow f_1 \rightarrow a_1^* \rightarrow f_3 \rightarrow a_3^* \rightarrow f_4 \rightarrow a_4^*$



Fully Competitive Tasks

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

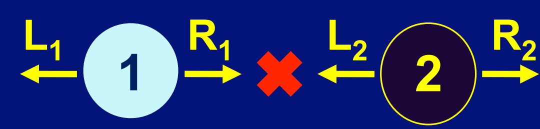
- **Fully Competitive Stochastic Game**
  - $\rho 1 = - \rho 2$ for two agents
  - Minimax principle can be applied

- **Minimax Principle**
  - Maximize one's benefit while the opponent endeavors to minimize it

Minimax Principle Example

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews, vol. 38, no.2, Mar. 2008.

- **Zero-Sum Static Game**
  - Agent 1
    - Reach the goal in the middle
    - Avoid capture by its opponent
  - Agent 2
    - Prevent agent 1 from reaching the goal
    - Prefer to capture agent 1
  - Opposite goal
    - Q function of agent 2 is the negative of agent 1



Q_1	L_2	R_2
L_1	0	1
R_1	-10	10

Q_2	L_2	R_2
L_1	0	-1
R_1	10	-10

Minimax Q-learning

L. M. Littman, "Markov games as a framework for multi-agent reinforcement learning," *in Proc. Int'l Conf. Machine Learning (ICML-94)*, Jul. 1994.

- **Opponent Independent Algorithm**
- **Algorithm**
  - Update rule for agent 1:

$$Q_{k+1}(s_k, a_{1,k}, a_{2,k}) = (1-\alpha)Q_k(s_k, a_{1,k}, a_{2,k}) + \alpha[r_{k+1} + \gamma \mathbf{m}_1(Q_k, s_{k+1})]$$

$$\mathbf{m}_1(Q, s) = \max_{\pi_1(s,\cdot)} \min_{a_2} \sum_{a_1} \pi_1(s, a_1) Q(s, a_1, a_2)$$

$$\pi_{1,k}(s_k, \cdot) = \arg \mathbf{m}_1(Q_k, s_k)$$

- $m_1(Q,s)$: minimax return of agent 1 (solved by linear programming)
- $\pi_{1,k}(s,\bullet)$: stochastic strategy of agent 1 in state s at time k

Mixed Tasks

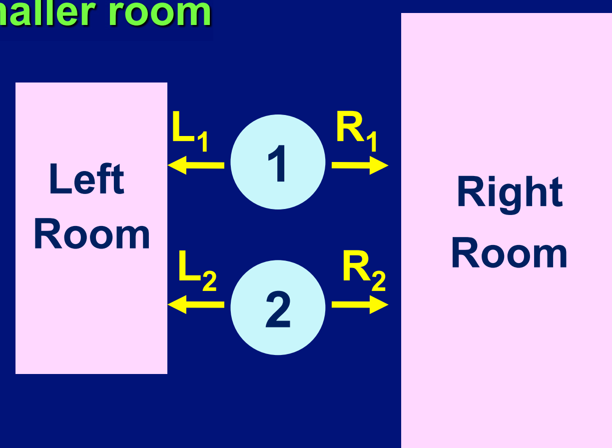
L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Mixed Stochastic Game**
  - No constraints imposed on the reward functions of the agents
  - Appropriate for immediate interests of agents in conflict
  - Multiple equilibriums may exist in a particular state
- **Equilibrium Selection**
  - Break the tie between multiple equilibriums
- **Agent Tracking**
  - Estimate models of other agents' strategies or policies
  - Act best response to these models

Equilibrium Selection Example

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **General-Sum Static Game**
  - Two Cleaning Robots
    - Each prefers to clean the smaller room
  - Two Nash equilibriums
    - (L_1, R_2) and (R_1, L_2)
  - Break the tie
    - Coordination
    - Social convention



Q_1	L_2	R_2
L_1	0	3
R_1	2	0

Q_2	L_2	R_2
L_1	0	2
R_1	3	0

Agent Tracking

L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.

- **Fictitious Play**
  - Agent i learns models for all other agent j ≠ i
  - Agent i's model of agent j's strategy

$$\pi_j^i(a_j) = \frac{C_j^i(a_j)}{\sum_{\tilde{a}_j \in A_j} C_j^i(\tilde{a}_j)}$$

    - $C_j(a_j)$ counts the number of times agent j taking action a_j
  - Multi-state version:

$$\hat{\pi}_j^i(s, a_j) = \frac{C_j^i(s, a_j)}{\sum_{\tilde{a}_j \in A_j} C_j^i(s, \tilde{a}_j)}$$

MARL for Cognitive Radio Networks

- Coordination for Cooperation
- Adaptation to behaviors of PUs and CR users
- Tracking of Malicious CR users

References

- R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
- W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality,* John Wiley & Sons, New York, NY, 2007.
- T. M. Mitchell, *Machine Learning,* McGraw-Hill, 1997.
- E. Alpaydin, *Introduction to Machine Learning,* MIT Press, Cambridge, MA, 2004.
- E. Hossain, D. Niyato, and Z. Han, *Dynamic Spectrum Access and Management in Cognitive Radio Networks,* Cambridge University Press, New York, NY, 2009.
- L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews,* vol. 38, no.2, Mar. 2008.
- C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," *in Proc. Int'l Conf. Machine Learning (ICML-02),* Jul. 2002.
- L. M. Littman, "Markov games as a framework for multi-agent reinforcement learning," *in Proc. Int'l Conf. Machine Learning (ICML-94)*, Jul. 1994.