Multi-Agent Learning

Overview of Algorithms.

Fictitious Play

Fictitious Play

- Model-based learning with beliefs about the opponent's strategy
  - Mixed strategy according to empirical distribution of previous actions

$$P(a) = \frac{w(a)}{\sum_{a' \in A} w(a')}$$

- Properties
  - Sensitive to initial beliefs
  - Steady state action profiles (action profile played in all future rounds)
    - If a pure-strategy profile is a strict Nash equilibrium of a stage game, then it is a steady state of fictitious play in the repeated game
    - If a pure-strategy profile is a steady state of fictitious play in the repeated game, then it is a (possibly weak) Nash equilibrium in the stage game

Fictitious Play

- Properties
  - If the empirical distribution of each player's strategies converges in fictitious play, then it converges to a Nash equilibrium
  - Sufficient (independent) conditions for empirical frequencies of play to converge in fictitious play
    - The game is zero sum
    - The game is solvable by iterated elimination of strictly dominated strategies
    - The game is a potential game
    - The game is 2 x n and has generic payoffs

Fictitious Play - Example

- Anti-Coordination Game
  - Empirical action probabilities according to Nash equilibrium
  - No payoff, since the learning algorithm makes the agents coordinated

	A	B
A	0,0	1,1
B	1,1	0,0

Round	1's action	2's action	1's beliefs	2's beliefs
0			(1,0.5)	(1,0.5)
1	B	B	(1,1.5)	(1,1.5)
2	A	A	(2,1.5)	(2,1.5)
3	B	B	(2,2.5)	(2,2.5)
4	A	A	(3,2.5)	(3,2.5)
…	…	…	…	…

Rational Learning

Rational Learning

- Beliefs of each player about his opponent's strategies may be expressed by any probability distribution over the set of all possible strategies
- Start with prior beliefs about the opponent's strategy, as well as possible strategies
  - S_{-i}^{i}: The set of strategies that player i considers possible for the opponent -i
  - H: The set of possible histories of the game
- Use Bayesian updating to update beliefs about opponent strategies

$$P_i(s_{-i}|h) = \frac{P_i(h|s_{-i})P_i(s_{-i})}{\sum_{s'_{-i} \in S_{-i}^{i}} P_i(h|s'_{-i})P_i(s'_{-i})}$$

Rational Learning

- Properties
  - In self-play, under ”some conditions”
    - Agents get close to correct beliefs about their opponent’s strategy
    - Agents converge towards a Nash equilibrium with high probability
  - Rough summary of conditions
    - Plays best response strategy
    - Positive probability histories are assigned positive probability beliefs

Rational Learning - Example

- Prisoners' Dilemma
- Assumption regarding strategies
  - Belief (S^i_{-i}): Opponent plays one of $g_0, g_1, \ldots, g_\infty$
  - g_∞ is the trigger strategy g_∞
  - g_T coincides with g_∞ for t < T, then defects
  - Player selects best response from $g_0, g_1, \ldots, g_\infty$

	C	D
C	3,3	0,4
D	4,0	1,1

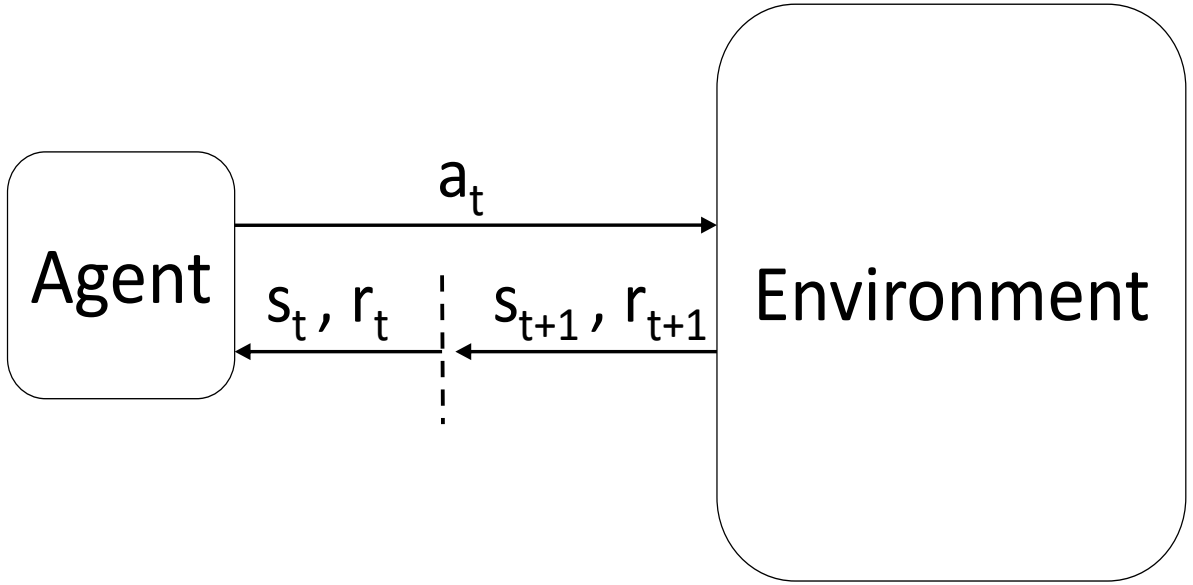
- Update of player i after seeing opponent cooperate in every step (depending on value of $P_i(h_t|g_T)$)

$$P_i(g_T|h_t) = \begin{cases} 0, & T \leq t \\ \dfrac{P_i(g_T)}{\sum_{k=t+1}^{\infty} P_i(g_k)}, & T > t \end{cases}$$

Reinforcement Learning

Reinforcement Learning

- Trial-and-error learning
- Typically modelled as a Markov Decision Process (MDP), (S, A, T, R)
  - S: Set of states
  - A: Set of actions
  - T: Transition dynamics
  - R: Reward function
- Goal: Maximize expected future return



Reinforcement Learning

- Value functions
  - $V_\pi(s)$: The value of being in state s and then following policy π

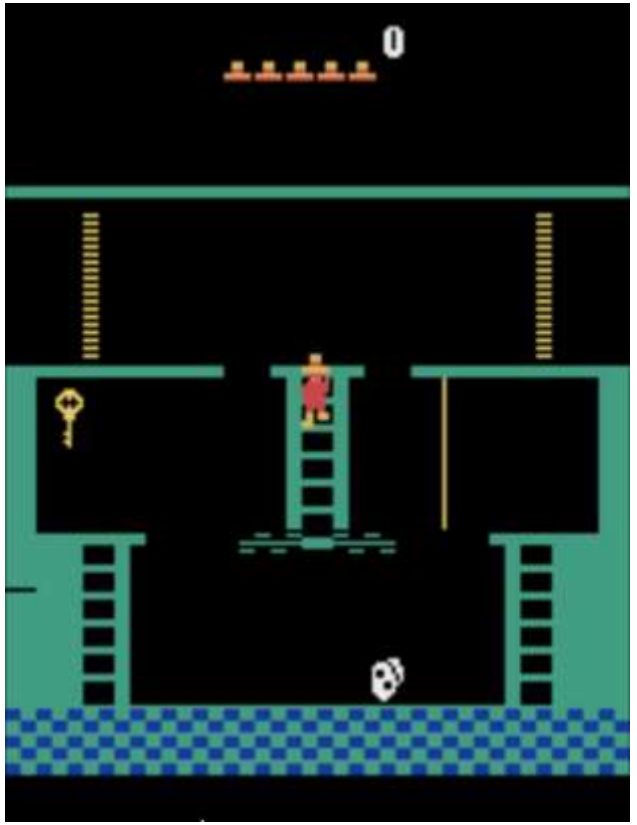
$$V_\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\right]$$

  - $Q_\pi(s)$: The value of being in state s, taking action a, and then following policy π

$$Q_\pi(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a\right]$$

Reinforcement Learning

- The trade-off between exploration and exploitation
  - To learn the agent must explore
  - To accumulate reward the agent must exploit
- Challenges
  - Difficult to balance the two (e.g., when to stop exploring)
  - Difficult to explore & learn in environments with sparse rewards



Q-learning

- Q-learning: Learning the state-action value function

  Initialize the Q-function arbitrarily

  repeat *until convergence*

  1. Observe the current state s_t
  2. Select action a_t (e.g., through ε-greedy selection) and take it
  3. Observe the next state and reward s_{t+1}, r_{t+1}
  4. Perform the following update step (with the learning rate $\alpha \in (0,1)$)

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right)$$

Q-learning Convergence Properties

- **Theorem 7.4.2**: *Q-learning guarantees that the Q values converge to those of the optimal policy, provided that each state-action pair is sampled an infinite number of times, and that the time-dependent learning rate α_t obeys*

$$0 \leq \alpha_t < 1$$

$$\sum_0^\infty \alpha_t = \infty$$

$$\sum_0^\infty \alpha_t^\infty < \infty$$

Extensions to Zero-Sum Stochastic Games

- Approaches
  - Ignore the existence of the other agent
    - Works well against opponents with stationary strategies
    - Otherwise no guarantees
  - Find Q-function for combined actions of agents $A = A_1 \times A_2$
    - Requires assumptions about opponent
      - Assume best response
      - Keep track of action frequency
      - Estimate probability of strategy

Minimax-Q

- Agent uses policy π to select actions, update step according to maxmin strategy

$$Q_i(s_t, a_t, o_t) = Q_i(s_t, a_t, o_t) + \alpha\left(r_{t+1} + \gamma max_\pi min_o Q_i(s_{t+1}, \pi(s_{t+1}, a_{t+1}), o) - Q_i(s_t, a_t, o_t)\right)$$

- Policy π is updated in each step based on the current Q function

$$\pi(s,\cdot) = argmax_{\pi'(s,\cdot)}(min_{o'} \sum_{a'} (\pi(s, a') * Q(s, a', o')))$$

- Converges to the value of zero-sum games in self play (under conditions of Q-learning)

Belief-Based Reinforcement Learning

- Extension of Q-learning
  - Model other agent (e.g., as in Fictitious Play or Rational Learning)
- Update step based on other agent's action probabilities

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma max_{a_i} \sum_{a_{-i} \subset A_{-i}} Q\left(s_{t+1}, (a_i, a_{-i})\right) Pr_i(a_{-i}) - Q(s_t, a_t) \right)$$

No-Regret Learning

No-Regret Learning

- Select actions to minimize regret for achieved reward α compared to reward given by strategy s

$$R^t(s) = \alpha^t(s) - \alpha^t$$

- Learning rule exhibits no regret if for pure strategy s

$$\Pr([\lim inf R^t(s)] \leq 0) = 1$$

- Does not require an opponent model
- Does not take into account that opponents may change strategy over time

Example

- Regret Matching
  - Let action probability be proportional to its regret (for positive regrets)

$$\sigma_i^{t+1}(s) = \frac{R^t(s)}{\sum_{s' \in S_i} R^t(s')}$$

Evolutionary Learning

Evolutionary Learning

- Simulates populations of agents
- Individual agents are evaluated by a fitness function
- Uses a reproduction mechanisms to produce a new population
  - Individuals with high fitness values affect the next population the most
  - Random mutations to, e.g., add qualities missing in the population

The Replicator Dynamic

- Models a population of frequently interacting agents
  - For example symmetric 2-player normal form stage game, e.g., Prisoner's Dilemma
  - Each agent plays a pure strategy at each point in time
  - Proportion of population that plays a certain strategy changes over time, based on its payoff

	A	**B**
A	*x,x*	*u,v*
B	*v,u*	*y,y*

The Replicator Dynamic

- NFG $G = (\{1,2\}, A, u)$, $\varphi_t(a)$ is the number of players playing action a at time t

$$\theta_t(a) = \frac{\varphi_t(a)}{\sum_{a' \in A} \varphi_t(a')} \ , \ u_t(a) = \sum_{a'} \theta_t(a') u(a, a') \ , \ \dot{\varphi}_t(a) = \varphi_t(a) u_t(a)$$

- Average expected payoff of the whole population

$$u_t^* = \sum_a \theta_t(a) u_t(a)$$

- Change in fraction of agents playing action a at time t

$$\dot{\theta}_t(a) = \frac{\left[\dot{\varphi}_t(a) \sum_{a' \in A} \varphi_t(a')\right] - \left[\varphi_t(a) \sum_{a' \in A} \dot{\varphi}_t(a')\right]}{\left[\sum_{a' \in A} \varphi_t(a')\right]^2} = \theta_t(a)[u_t(a) - u_t^*]$$

Properties of the Replicator Dynamic

- Definitions
  - Steady-state: Action fractions do not change
  - Stable steady-state: A system that starts close to the steady-state remains nearby
  - Asymptotically stable state: A system that starts close to the steady-state approaches the steady-state over time
- A symmetric mixed strategy Nash equilibrium of G is a steady state
- A stable steady-state for the mixed strategy s is a Nash equilibrium of G
- An asymptotically stable steady-state for the mixed strategy s is a Nash equilibrium of G that is trembling-hand perfect and isolated

www.liu.se

LINKÖPING UNIVERSITY