Outline

- Multi-agent Reinforcement Learning (MARL)
- Stochastic Games
- Opponent Modelling
  - Fictitious Play
  - Solving (Unique) Equilibrium
- Cooperative Stochastic Games
  - Joint Q learning
  - Convergence properties
- Competitive Stochastic Games (Zero-sum games)
  - Minimax Q learning
  - Convergence properties
- Mixed Cooperative-Competitive Stochastic Games (General-sum games)
  - Nash Q learning
  - Convergence properties
Multi-agent Reinforcement Learning

Multi-agent Games + Sequential decision making

Newer field with unique challenges and opportunities
Stochastic Games

- (Simultaneously moving) Stochastic Game ($N$-agent MDP)
  - Tuple $⟨N, S, A^1, \ldots, A^N, R^1, \ldots, R^N, T, γ⟩$
  - $N$: Number of agents
  - $S$: Shared state space $s ∈ S$
  - $A^j$: Action space of agent $j$
    $$⟨a^1, a^2, \ldots, a^N⟩ ∈ A^1 × A^2 × \ldots × A^N$$
  - $R^j$: Reward function for agent $j$ - $R^j(s, a^1, \ldots, a^N) = Pr(r^j | s, a^1, \ldots, a^N)$
  - $T$: Transition function - $Pr(s' | s, a^1, \ldots, a^N)$
  - $γ$: Discount factor: $0 ≤ γ ≤ 1$
    - Discounted: $γ < 1$
    - Undiscounted: $γ = 1$
  - Horizon (i.e., # of time steps): $h$
    - Finite horizon: $h ∈ \mathbb{N}$
    - Infinite horizon: $h = \infty$
  - Policy (strategy) for agent $i$ - $π^i : S → Ω(A^i)$
    - Goal: Find optimal policy such that $π^* = \{π^*_1, \ldots, π^*_N\}$, where
      $$π^*_i = \arg\max_{π^i} \sum_{t=0}^{h} γ^t E[π[r^j(s, a)]]$$
      where $a ≡ \{a^1, \ldots, a^N\}$ and $π ≡ \{π^1, \ldots, π^N\}$
Playing a stochastic game

- Players choose their actions at the same time
  - No communication with other agents
  - No observation of other player’s actions

- Each player chooses a strategy $\pi^i$ which is a mapping from states to actions and can be either
  - Mixed strategy: Distribution over actions for at least one state
  - Pure strategy: One action with prob 100% for all states

- At each state, all agents face a stage game (normal form game) with the Q values of the current state and joint action of each player being the utility for that player

- The stochastic game can be thought of as a repeated normal form game with a state representation
Optimal Policy

- In MARL, the optimal policy should correspond to some equilibrium of the stochastic game.

- The most common solution concept is the Nash equilibrium.

- Let us define a value function for the multi-agent setting:
  \[
  v^j_\pi(s) \triangleq \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[r^j_t | s_o = s, \pi]
  \]

- Nash equilibrium under the stochastic game satisfies:
  \[
  v^j_{(\pi^*, \pi^*_w)}(s) \geq v^j_{(\pi^j, \pi^*_{-j})}(s)
  \]
  \[
  \forall s \in S; \forall j; \forall \pi^j \neq \pi^*_j
  \]
Independent learning

- Naive approach: Apply the single agent Q-learning directly
- Each agent would update its Q-values using the Bellman update:

\[
Q^j(s, a^j) \leftarrow Q^j(s, a^j) + \alpha \left( r^j + \gamma \max_{a'^j} Q^j(s', a'^j) - Q^j(s, a^j) \right)
\]

- Each agent assumes that the other agent(s) are part of the environment

- Merit: Simple approach, easy to apply
- Demerit: Might not work well against opponents playing complex strategies
- Demerit: Non-stationary transition and reward models
- Demerit: No convergence guarantees
Cooperative Stochastic Games

- (Simultaneously moving) Stochastic Game ($N$-agent MDP)
  - Tuple $\langle N, S, A^1, ..., A^N, R^1, ..., R^N, T, \gamma \rangle$
  - $N$: Number of agents
  - $S$: Shared state space $s \in S$
  - $A^j$: Action space of agent $j$
    - $\langle a^1, a^2, ..., a^N \rangle \in A^1 \times A^2 \times ... \times A^N$
  - $R^j$: Reward function for agent $j$ - $R(s, a^1, ..., a^N) = Pr(r|s, a^1, ..., a^N)$, $\forall j$
  - $T$: Transition function - $Pr(s'|s, a^1, ..., a^N)$
  - $\gamma$: Discount factor: $0 \leq \gamma \leq 1$
    - Discounted: $\gamma < 1$
    - Undiscounted: $\gamma = 1$
  - Horizon (i.e., # of time steps): $h$
    - Finite horizon: $h \in \mathbb{N}$
    - Infinite horizon: $h = \infty$
  - Policy (strategy) for agent $i$ - $\pi^i: S \rightarrow \Omega(A^i)$
- Goal: Find optimal policy such that $\pi^* = \{\pi^*_1, ..., \pi^*_N\}$, where

$$\pi^*_i = \arg\max_{\pi^i} \sum_{t=0}^{h} \gamma^t \mathbb{E}_{\pi}[r^i(s, a)],$$

where $a \triangleq \{a^1, ..., a^N\}$ and $\pi \triangleq \{\pi^1, ..., \pi^N\}$
Optimal Policy

• The equilibrium in the case of cooperative stochastic games is the Pareto dominating (Nash) equilibrium
• Each stage game of this stochastic game faces a coordination game
• There exists a unique Pareto dominating (Nash) equilibrium in utilities

<table>
<thead>
<tr>
<th></th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseball</td>
</tr>
<tr>
<td>Alice</td>
<td>2,2</td>
</tr>
<tr>
<td></td>
<td>0,0</td>
</tr>
</tbody>
</table>

Baseball
Soccer
Opponent Modelling

- Note that an agent’s response requires knowledge of other agent’s actions
- This is a simultaneously move game where each agent does not know what the other agents will do
- So each agent should maintain a belief over other agents actions at current state
- This process of maintaining and updating a belief over the next actions of other agents is called opponent modelling

Types of Opponent Modelling:
- Fictitious Play
- Gradient Based Methods
- Solving Unique Equilibrium (for each stage game)
- Bayesian Approaches
Fictitious Play

- Each agent assumes that all opponents are playing a **stationary mixed strategy**
- Agents maintain a count of number of times another agent performs an action
  \[ n^i_t(s, a_j) \leftarrow 1 + n^i_{t-1}(s, a_j), \forall j, \forall i \]
- Agents **update and sample an action from their belief** about this strategy at each state according to
  \[ \mu^i_t(s, a_j) \sim \frac{n^i_t(s, a_j)}{\sum_{a_j'} n^i_t(s, a_j')} \]
  - The fictitious action $\mu^i_t(s, a_j)$ is sampled from an empirical distribution of past actions of other agent (mixed strategy)
- Agents calculate best responses according to this belief
Learning in cooperative stochastic games

- Algorithm: Joint action learner (JAL) or Joint Q learning (JQL)
- Challenge: Respond to environment as well as opponent(s)
- Same as Q learning but agents also include the opponent action in Q-updates
- Each agent would update its Q-values using the Bellman update:

\[ Q_i^j(s, a^j, a_i^{-j}) \leftarrow Q_i^j(s, a^j, a_i^{-j}) + \alpha \left( r_i^j + \gamma \max_{a_i^{-j}} Q_i^j(s', a^j, a_i^{-j}) - Q_i^j(s, a^j, a_i^{-j}) \right) \]

- Need to balance exploration exploitation tradeoff
- Objective for agent: Find the optimal policy for best response
- Objective for system: Find the NE of the stochastic game (or Nash Q function for the game)

- Nash Q function: Agent’s immediate reward and discounted future rewards when all agents follow the NE policy

\[ Q_i^*(s, a) = r_i^*(s, a) + \gamma \sum_{s' \in S} P(s' \mid s, a) v_i'(s', \pi_1^*, \ldots, \pi_n^*) \]
Joint Q learning

\[ \text{JointQlearning}(s, Q) \]

Repeat

Repeat for each agent \( i \)

Select and execute \( a^i \)

Observe \( s', r^i \) and \( a^{-i} \), where \( a^{-i} = \{a^1, \ldots, a^{i-1}, a^{i+1}, \ldots, a^N\} \)

Update counts: \( n(s, a) \leftarrow n(s, a) + 1 \)

Update counts: \( n^i_j(s, a) \leftarrow 1 + n^i_{-1}(s, a), \forall j \)

Learning rate: \( \alpha \leftarrow \frac{1}{n(s, a)} \)

Update Q-value:

\[ Q^i(s, a^i, a^{-i}) \leftarrow Q^i(s, a^i, a^{-i}) + \alpha \left( r^i + \gamma \max_{a'} Q^i(s', a^i, \mu^i(s', a_1), \ldots, \mu^i(s', a_N)) - Q^i(s, a^i, a^{-i}) \right) \]

\( s \leftarrow s' \)

Until convergence of \( Q^i \)
Convergence of joint Q learning

- If the game is finite (finite agents and finite number of strategies for each agent), then fictitious play will converge to true response of opponent(s) in the time limit in self-play.
- Self-play: All agents learn using the same algorithm.
- Joint Q-learning converges to Nash Q-values in a cooperative stochastic game if:
  - Every state is visited infinitely often (due to exploration).
  - The learning rate $\alpha$ is decreased fast enough, but not too fast (sufficient conditions for $\alpha$):
    \[
    (1) \sum_n \alpha_n \to \infty \quad (2) \sum_n (\alpha_n)^2 < \infty
    \]
- In cooperative stochastic games, the Nash Q-values are unique (guaranteed unique equilibrium point in utilities).
Joint Q learning

\textbf{JointQlearning}(s, Q)

\begin{itemize}
  \item Repeat
  \begin{itemize}
    \item Repeat for each agent \( i \)
    \begin{itemize}
      \item Select and execute \( a^i \)
      \item Observe \( s', r^i \) and \( a^{-i} \), where \( a^{-i} = \{ a^1, \ldots, a^{i-1}, a^{i+1}, \ldots, a^N \} \)
      \item Update counts: \( n(s, a) \leftarrow n(s, a) + 1 \)
      \item Update counts: \( n^i_t(s, a_j) \leftarrow 1 + n^i_{t-1}(s, a_j), \forall j \)
      \item Learning rate: \( \alpha \leftarrow \frac{1}{n(s, a)} \)
    \end{itemize}
    \item Update Q-value:
    \[ Q^i(s, a^i, a^{-i}) \leftarrow Q^i(s, a^i, a^{-i}) + \alpha \left( r^i + \gamma \max_{a^i} Q^i(s', a^i, a^{-i}) - Q^i(s, a^i, a^{-i}) \right) \]
  \end{itemize}
\end{itemize}

\( s \leftarrow s' \)

Until convergence of \( Q^i \)
Common exploration methods

- $\epsilon$-greedy:
  - With probability $\epsilon$, execute random action
  - Otherwise execute best action
  \[
  a_i^* = \arg \max_{a^i} Q^i(s, a^i, \mu^i(s, a_1), \ldots, \mu^i(s, a_N))
  \]

- Boltzmann exploration
  - Increasing temperature $T$ increases stochasticity
  \[
  Pr(a) = \frac{e^{Q_i^i(s, a^i, \mu^i(s, a_1), \ldots, \mu^i(s, a_N))}}{\sum_a e^{Q_i^i(s, a^i, \mu^i(s, a_1), \ldots, \mu^i(s, a_N))}}
  \]
Competitive Stochastic Games

- (Simultaneously moving) Stochastic Game ($N$-agent MDP)
  - Tuple $\langle N, S, A^1, A^2, R^1, R^2, T, \gamma \rangle$
  - $N$: Number of agents
  - $S$: Shared state space $s \in S$
  - $A^j$: Action space of agent $j$
    $$\langle a^1, a^2 \rangle \in A^1 \times A^2$$
  - $R^j$: Reward function for agent $j$ - $R^j(s, a^1, a^2) = Pr(r^j_t | s_t, a^1_t, a^2_t)$, $\forall j$
  - Condition on Reward function: $r^1_t + r^2_t = 0$, $\forall t$
  - $T$: Transition function - $Pr(s' | s, a^1, a^2)$
  - $\gamma$: Discount factor: $0 \leq \gamma \leq 1$
    - Discounted: $\gamma < 1$
    - Undiscounted: $\gamma = 1$
  - Horizon (i.e., # of time steps): $h$
    - Finite horizon: $h \in \mathbb{N}$
    - Infinite horizon: $h = \infty$
  - Policy (strategy) for agent $i$ - $\pi^i : S \rightarrow \Omega(A^i)$

- Goal: Find optimal policy such that $\pi^* = \{ \pi^*_1, \ldots, \pi^*_N \}$, where
  $$\pi^*_i = \arg \max_{\pi^i} \sum_{t=0}^{h} \gamma^t \mathbb{E}[r^i_t(s, a)], \text{where } a \triangleq \{a^1, a^2\} \text{ and } \pi \triangleq \{\pi^1, \pi^2\}$$
Optimal Policy

• The equilibrium in the case of competitive stochastic games is the min-max Nash equilibrium
• Each stage game of this stochastic game faces a zero-sum game
• There exists a unique min-max (Nash) equilibrium in utilities
• Optimal min-max value function

\[
V^*_j(s) = \max_{a^j} \min_{a^{-j}} [r^j(s, a^j, a^{-j}) + \gamma \sum_{s'} Pr(s' | s, a^j, a^{-j}) V^*_j(s')] 
\]

• For a competitive stochastic game there exists a unique min-max value function and hence a unique min-max Q-function
Learning in competitive stochastic games

- Algorithm: Minimax Q-Learning
- Q-values for each agent $j$ are over joint actions: $Q^j(s, a^j, a^{-j})$
  - $s =$ state
  - $a^j =$ action
  - $a^{-j} =$ opponent action
- Instead of playing the best $Q^j(s, a^j, a^{-j})$ play min-max Q

\[
Q^j(s, a^j, a^{-j}) \leftarrow (1 - \alpha)Q^j(s, a^j, a^{-j}) + \alpha(r^j + \gamma V^j(s'))
\]

\[
V^j(s') \leftarrow \max_{a^j} \min_{a^{-j}} Q^j(s', a^j, a^{-j})
\]
Minimax Q learning

Minimax Q\text{learning}(s, a, Q^*)

Repeat
  Repeat for each agent
    Select and execute action $a^j$
    Observe $s', a^{-j}$ and $r$
    Update counts: $n(s, a) \leftarrow n(s, a) + 1$
    Learning rate: $\alpha \leftarrow \frac{1}{n(s, a)}$
    Update Q-value:
    \[
    Q^j(s, a^j, a^{-j}) \leftarrow (1 - \alpha)Q^j(s, a^j, a^{-j}) + \alpha(r^j + \gamma \max_{a^j^{-j}} Q^j(s', a^j, a'^{-j}))
    \]
    $s \leftarrow s'$
  Until convergence of $Q^*$
Return $Q^*$
Opponent Modelling

- In a competitive game rational agents always take a min-max action

- There is no requirement for a separate opponent modelling strategy in self-play

- However:
  - Other agents could use different algorithms
  - Computing the min-max action can be time consuming

- Alternative: Fictitious play
  - Theorem: Fictitious play also converges in competitive zero-sum games
  - Theorem: Fictitious play converges to the min-max action in self-play
Convergence of Minimax Q learning

- Convergence in self-play
- Minimax Q-learning converges to min-max equilibrium in a competitive stochastic game if:
  - Every state is visited infinitely often (due to exploration)
  - The learning rate $\alpha$ is decreased fast enough, but not too fast

(sufficient conditions for $\alpha$):

\[
(1) \sum_{n} \alpha_n \rightarrow \infty \quad (2) \sum_{n} (\alpha_n)^2 < \infty
\]

- In a competitive stochastic games, the Nash Q-values are unique (guaranteed unique min-max equilibrium point in utilities)
Exploration vs Exploitation Tradeoff

- Same as Q-learning and Joint Q learning
- \( \varepsilon \)-greedy
  - Play random action with probability \( \varepsilon \)
  - Play min-max action with probability \( 1 - \varepsilon \)
  (or)
  - Play max action based on fictitious belief
(Mixed) Stochastic Games/ General-sum Stochastic Game

- (Simultaneously moving) Stochastic Game (N-agent MDP)
  - Tuple $\langle N, S, A^1, \ldots, A^N, R^1, \ldots, R^N, T, \gamma \rangle$
  - $N$: Number of agents
  - $S$: Shared state space $s \in S$
  - $A^i$: Action space of agent $i$
    $$\langle a^1, a^2, \ldots, a^N \rangle \in A^1 \times A^2 \times \ldots \times A^N$$
  - $R^i$: Reward function for agent $i$ - $R^i(s, a^1, \ldots, a^N) = Pr(r^i | s, a^1, \ldots, a^N)$
  - Rewards of all agents can be related arbitrarily
  - $T$: Transition function - $Pr(s' | s, a^1, \ldots, a^N)$
  - $\gamma$: Discount factor: $0 \leq \gamma \leq 1$
    - Discounted: $\gamma < 1$ Undiscounted: $\gamma = 1$
    - Horizon (i.e., # of time steps): $h$
      - Finite horizon: $h \in \mathbb{N}$ Infinite horizon: $h = \infty$
  - Policy (strategy) for agent $i$ - $\pi^i: S \rightarrow \Omega(A^i)$
- Goal: Find optimal policy such that $\pi^* = \{\pi^*_1, \ldots, \pi^*_N\}$, where
  $$\pi^*_i = \arg \max_{\pi^i} \sum_{t=0}^{h} \gamma^t \mathbb{E}_s [r^i_t(s, a)], \text{ where } a \triangleq \{a^1, \ldots, a^N\} \text{ and } \pi \triangleq \{\pi^1, \ldots, \pi^N\}$$
Optimal Policy

• The equilibrium in the case of competitive stochastic games is the *(mixed strategy) Nash equilibrium* for the stochastic game

• Nash theorem guarantees *at-least one* mixed strategy NE exists

• There could be *multiple* Nash equilibria
  - Objective for agent: Find the *optimal policy for best response*
  - Objective for system: Find the NE of the stochastic game (or *Nash Q function* for the game)
  - Nash Q function: Agent’s immediate reward and discounted future rewards when all agents follow the NE policy

\[
Q^i_*(s, a) = r^i(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) v^i(s', \pi^1_*, \ldots, \pi^n_*)
\]

• Problem: Which NE should we converge to?
Learning in General-sum stochastic games

- Algorithm: Nash Q-learning
- Assumption: Self-play
- Every agent maintains the Q values of all other agents
- At each state, every agent face a stage (normal form) game
- Utilities of the normal form game are the Q values for each action for each agent
- Need to calculate NE of the normal form game $\pi^1(s') \cdots \pi^n(s')$
- Each agent would update its Q-values using the Bellman update:

$$Q^j(s, a^j, a^{-j}) \leftarrow Q^j(s, a^j, a^{-j}) + \alpha \left( r^j + \gamma \text{Nash}Q^j(s') \right)$$

where

$$\text{Nash}Q^j(s') = \pi^1(s') \cdots \pi^n(s') \cdot Q^j(s')$$

- Here, $\pi^j(s')$ is a vector containing distribution of probability of each action (mixed strategy)
- Here, $Q^j(s')$ is a vector containing Q values for all actions of the agent $j$
Nash Q learning

NashQ learning($s, a, Q^*$)

Repeat
  Repeat for each agent
    Select and execute action $a^j$
    Observe $s', a^{-j}$ and $r \equiv r^1, \ldots, r^N$
    Update counts: $n(s, a) \leftarrow n(s, a) + 1$
    Learning rate: $\alpha \leftarrow \frac{1}{n(s, a)}$
    Update Q-value for every $j = 1, \ldots, n$: 
    $$Q^j(s, a) \leftarrow (1 - \alpha)Q^j(s, a) + \alpha(r^j + \gamma NashQ^j(s'))$$
    $s \leftarrow s'$
  Until convergence of $Q^*$
Return $Q^*$
Opponent Modelling

- **Note:** Each agent is maintaining Q-values of all agents
- **Solution 1:** Agents can take equilibrium action if unique
  - Problem: Non-unique equilibria in practice
  - Problem: Equilibrium computation can take a long time
  - Problem: Convergence only under strong assumptions (unique equilibrium)
- **Solution 2:** Fictitious play
  - Problem: Convergence only under strong assumptions (unique equilibrium)
- **Solution 3:** Assume every agent is playing independent learning
  - Problem: No convergence guarantees
Convergence of Nash Q-learning

- Convergence in self-play (under strong assumptions)
- Nash Q-learning converges to the NE in a general sum stochastic game if
  - Every state is visited infinitely often (due to exploration)
  - The learning rate $\alpha$ is decreased fast enough, but not too fast
    (sufficient conditions for $\alpha$):
      \[
      (1) \sum_{n} \alpha_n \to \infty \quad (2) \sum_{n} (\alpha_n)^2 < \infty
      \]
  - The NE can be considered as a global optimum or a saddle point in each stage game of the stochastic game
    - (Important qualification) Can only be one of global optimum or saddle point (cannot alternate)
    - Extremely rare to hold in practice
    - Convergence observed even when the condition is violated
    - Guarantees unique convergence point in utilities and hence unique Nash Q function
Exploration vs Exploitation Tradeoff

- In practice, same as JAL, Minimax Q-learning and Q-learning
- $\epsilon$-greedy
  - Play random action with probability $\epsilon$
  - Play max action based on fictitious belief with probability $1 - \epsilon$
  (Or)
  - Play equilibrium action with probability $1 - \epsilon$
Alternative approaches

- A NE is **not always the best solution**
- NE is attractive because it is **unrestrictive** (all agents can be independent) and **Nash theorem guarantees existence**
- Can consider **other equilibria** as well:
  - Pareto-optimality
  - Regret
  - Correlated equilibrium
  - Dominant strategy equilibrium

- **Function approximation** techniques
- **Model-based** techniques
Summary

- Multi-agent Reinforcement Learning (MARL)
- Stochastic Games
- Opponent Modelling
  - Fictitious Play
  - Solving (Unique) Equilibrium
- Cooperative Stochastic Games
  - Joint Q learning
  - Convergence properties
- Competitive Stochastic Games (Zero-sum games)
  - Min-max Q learning
  - Convergence properties
- Mixed Cooperative-Competitive Stochastic Games (General-sum games)
  - Nash Q learning
  - Convergence properties