**Multi-Agent Reinforcement Learning**

Algorithms:

* **QLearnerBoltzmann**: A standard Q-learning approach combined with Boltzmann (softmax) exploration. Balances exploration and exploitation using a temperature parameter.
* **Learning Automata**: Uses probability vectors updated based on reward/penalty feedback. Effective in converging rapidly when clear optimal solutions exist.
* **Joint Action Learner**: Considers joint action pairs, estimating opponent action distributions. Learns stable joint policies effectively.
* **WoLF (Win-or-Learn-Fast)**: Adjusts its learning rate based on current performance relative to historical averages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Game/Agent** | **QLearnerBoltzmann** | **Learning Automata** | **JointActionLearner** | **WoLFAgent** |
| Prisoner's Dilemma | Converged to Nash equilibrium with some smaller oscillations. | Fastest convergence to Nash equilibrium. | Convergence to Nash equilibrium (similar to QLearnerBoltzman). | Converged after oscillations. |
| Matching Pennies | Oscillaing around 50-50, didn't converge. | Persistent oscillation around 50-50 without convergence. | Less oscillation, limited convergence to 50-50. | Significant oscillations, did not stabilize. |
| Battle of Sexes | Rapidly found equilibrium. | After oscillating, converged to different optima from other agents. | Converged, slower than QLearnerBoltzman. | Fast and stable convergence. |

**Results:**

* **QLearnerBoltzmann**: Generally effective but struggles in competitive games like Matching Pennies, leading to oscillatory behavior.
* **Learning Automata**: Excellent in Prisoner's Dilemma game, but struggles significantly in competitive settings like Matching Pennies game.
* **JointActionLearner**: Robust in most scenarios thanks to considering opponent actions explicitly.
* **WoLFAgent**: Effective in Prisoner’s Dilemma and Battle of Sexes games; however, prone to oscillations in competitive scenarios (Matching Pennies).