UNIVERSITY OF LIVERPOOL

COMP 532
Machine Learning and BioInspired Optimization

Lecture 18: Multi-Agent Learning

Dr. Shan Luo

Department of Computer Science

shan.luo@liverpool.ac.uk

Outline (two lectures)

- **Problem revisited**
- **Game Theory**
  - Repeated Games
  - Stochastic or Markov Games
- **"Naïve" approaches to multi-agent learning**
  - Fictitious play
- **More on stochastic games**
- **Spectrum of approaches**
- **Basic/state-of-the-art approaches**
  - minimax-Q, Nash-Q
  - tinkering with learning rates: WoLF
- **Challenges and Opportunities**

How do we model games that evolve over time?

- Stochastic (Markov) Games! (Games with states)
- Current Game = State
- State transitions are functions of joint actions

- Ingredients:
  - n Agents
  - States (S)
  - Payoffs (R)
  - Transition Probabilities (P)
  - Discount Factor (γ)

Example of a Stochastic Game

$\gamma = 0.9$

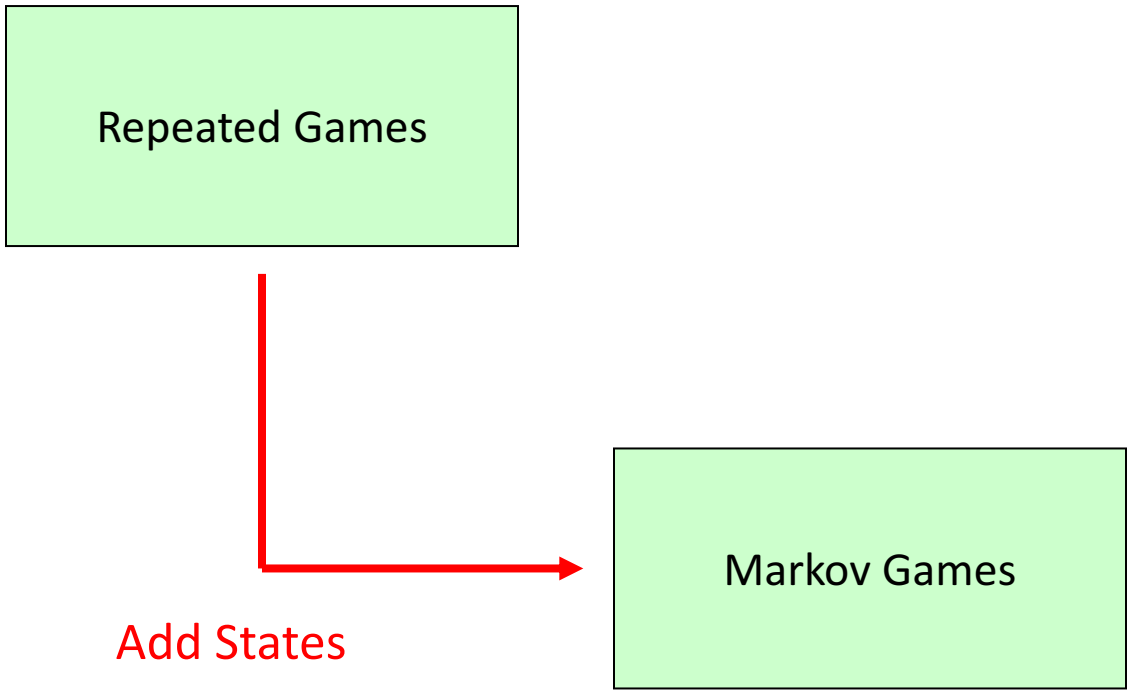
	C	D
A	1,2	3,4
B	5,6	7,8

Move with 50% probability when (A,C) or (A,D)

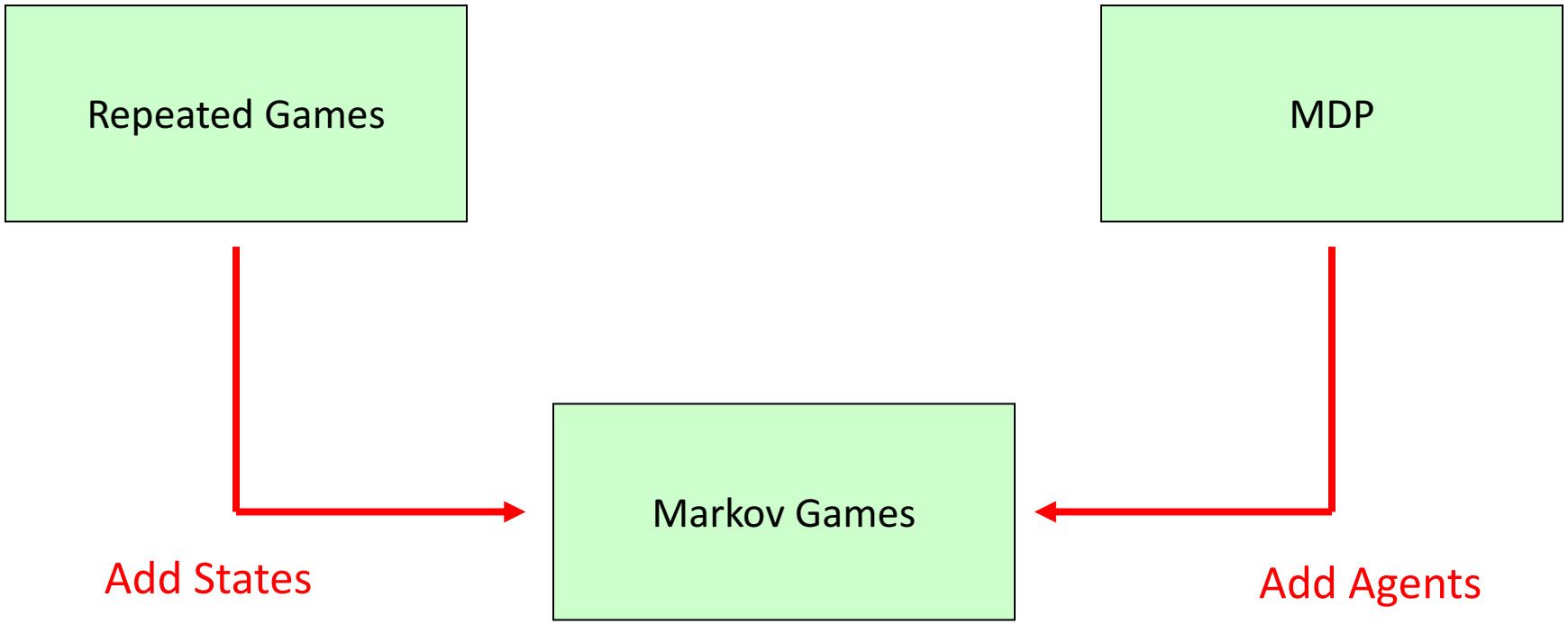
Move with 30% probability when (B,D)

	C	D	E
A	-1,2	-3,4	0,0
B	-5,6	-7,8	-10,10

Markov Game is a Generalization of…



Markov Game is a Generalization of…



Learning in Stochastic Games

- Learning is especially important in Markov Games because NE are hard to compute.

- Do we know:
  - Our own payoffs?
  - Others' rewards?
  - Transition probabilities?
  - Others' strategies?

Learning in Stochastic Games

- Adapted from Single-Agent RL:
  - **Independent Learning** (ignore others)
  - **Joint-Action Learning** (model others)
  - **Minimax Q-learning** (zero-sum games)
  - **Nash Q-learning** (based on Game Theory)
  - **CE Q-learning** (based on Game Theory)

Zero-Sum Stochastic Games

- Players' payoffs always sum up to the same value
  - Losses for one equal to winnings for others

- Nice properties:
  - All equilibria have the same value
  - It has a Bellman's-type equation
  - Value iteration using minimax => Nash equilibrium

Spectrum of MAL

From Independent Learners

TO

Joint Action Space Approaches

State-of-the-art:
usually in between these two extremes

Independent Learners

- Independent learners mutually ignore each other
- Implicitly perceive interaction with other agents as noise in a stochastic environment
- **Advantage**:
  - Straightforward application of single-agent techniques
  - Scales easily with number of agents
- **Disadvantage**:
  - Convergence guarantees from single-agent setting are lost
  - No explicit means for coordination

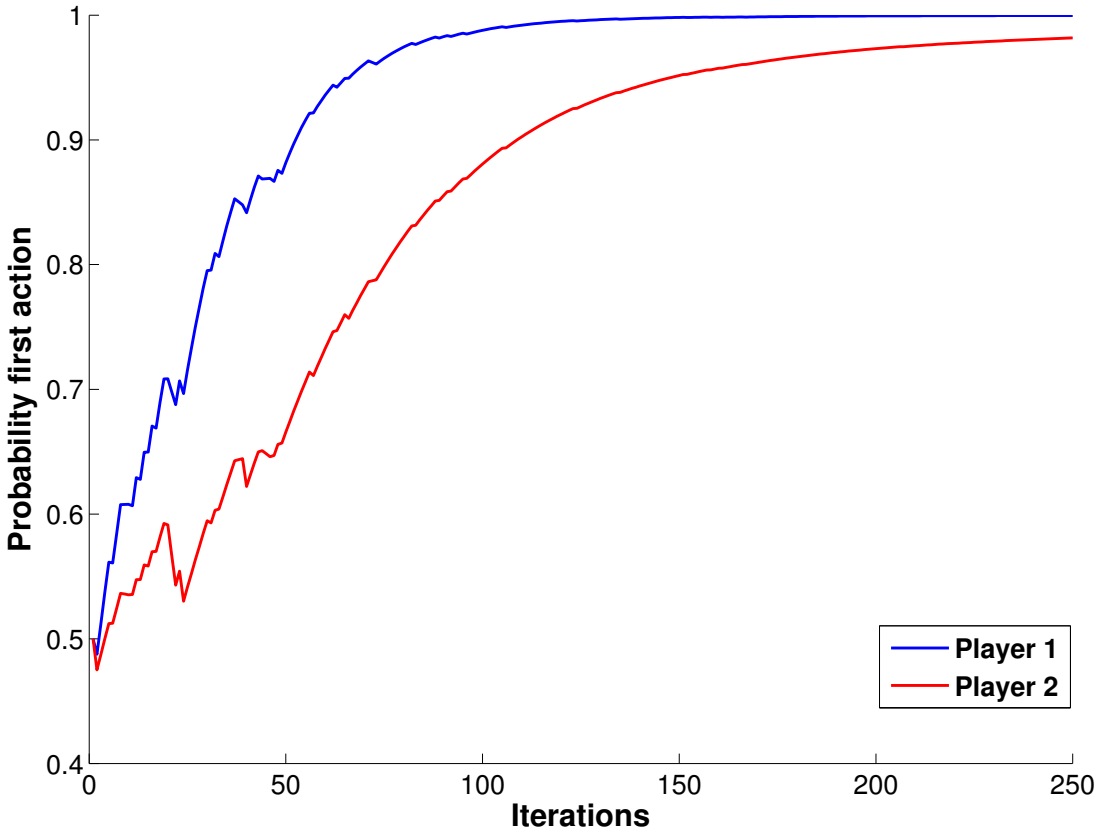
Independent Q-learners in normal-form games

- Two Q-learners interact in Battle of the Sexes
  - Learning rate $\alpha = 0.01$
  - Boltzmann with $\tau = 0.2$

	B	F
B	2, 1	0, 0
F	0, 0	1, 2

- They only observe their immediate reward
- Policy is gradually improved

Independent Q-learners in normal-form games

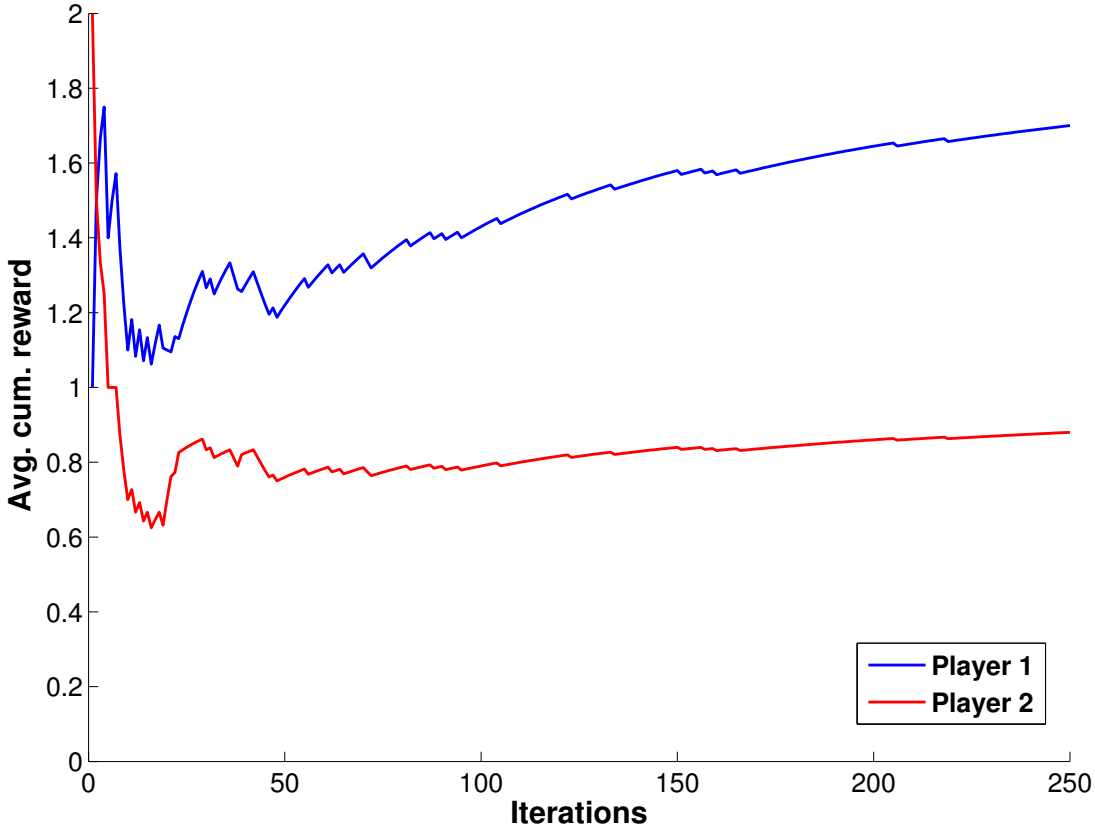


	B	F
B	2, 1	0, 0
F	0, 0	1, 2

This graph shows the progress of learning, which converges to a Nash equilibrium strategy.

The probability of the first actions: the probability with which either player chooses to play "B" in this game.

Independent Q-learners in normal-form games



The corresponding average reward received during the learning process

	B	F
B	2, 1	0, 0
F	0, 0	1, 2

The two Nash equilibria in this game are unbalanced!
Based on initial randomisation, the learning process could end up in either pure equilibrium (B,F) or (F,B) with equal probability.

Joint Action Learners

- Joint Action Learners observe the actions of other agents

- Similar to fictitious play, they assume a stationary policy (which they estimate) and play optimally against that estimate

- **Advantage**:
  - Better means of coordination by explicitly taking other agents into account

- **Disadvantage**:
  - Need to observe other agents' actions
  - Complexity grows exponentially with the number of agents

Minimax Q-learning
for two-player zero-sum stochastic games

- Zero sum: payoffs balance out, only need to observe own payoff

- Update rule based on joint action $< a_1, a_2 >$:

$$Q(s, a_1, a_2) \leftarrow (1 - \alpha) Q(s, a_1, a_2) + \alpha [r + \gamma V(s')]$$

$$V(s') = \max_{\pi_s} \min_{a_2'} \sum_{a_1'} Q(s', a_1', a_2') \, \pi_s(a_1')$$

probability of playing action a_1' when following policy π_s

Minimax Q-learning

- Does it work?
  - Performs better than naïve *independent* Q-learning
  - Converges to Nash
    (under similar conditions as single-agent Q-learning)
  - Limited to zero-sum games…

Q-learning in General-Sum Games

- Can we extend the algorithm to general-sum stochastic games?
  - Yes & No
  - Nash-Q Learning is such an extension
  - However, it has much worse computational and theoretical properties

Nash-Q Learning Algorithm

- Must learn $Q^i(s, a_1, \ldots, a_n)$ for all states s, joint actions $< a_1, \ldots, a_n >$, and for every agent i

- Update rule:

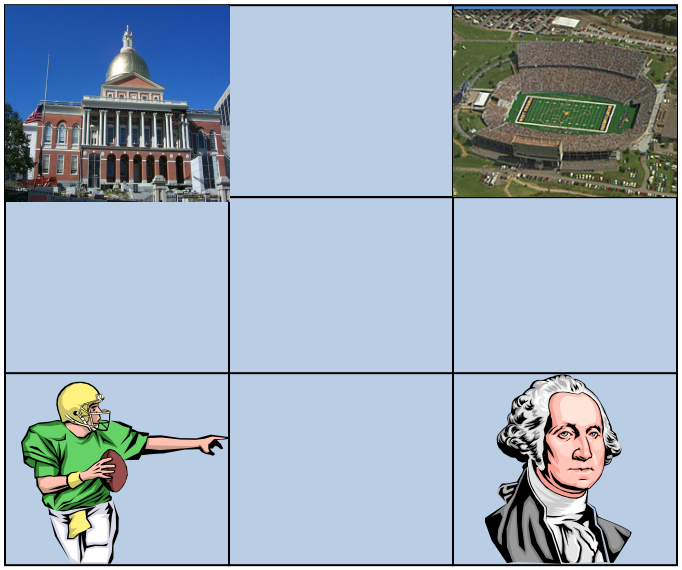
$$Q^i(s, a_1, \ldots, a_n) \leftarrow (1-\alpha)Q^i(s, a_1, \ldots, a_n) + \alpha\left[r + \gamma NashV^i(s')\right]$$

- $NashV^i(s')$ is the payoff in Nash equilibrium
  - needs to be computed!
- Assumes all players play the same Nash equilibrium
  - selection problem!

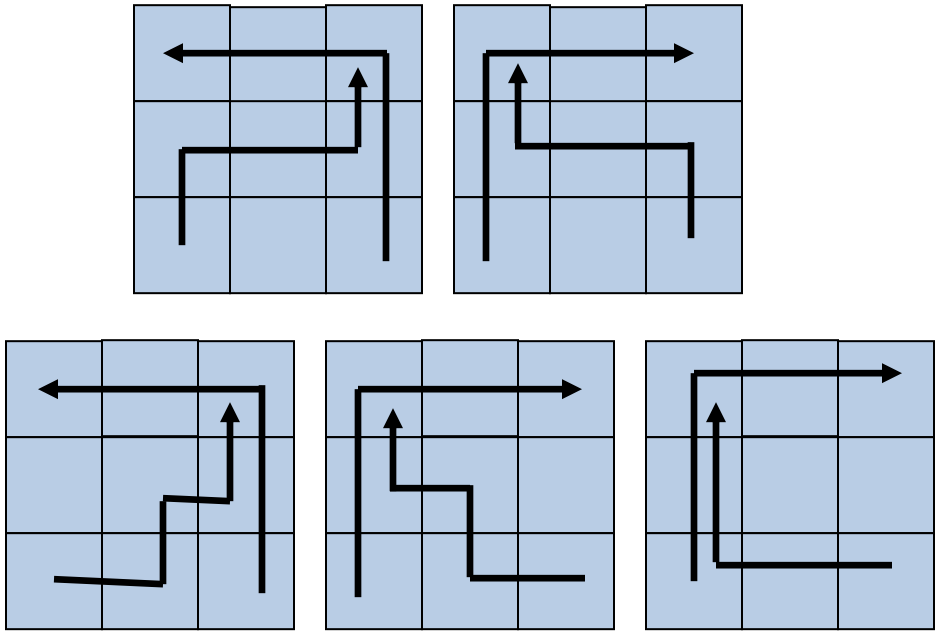
Convergence of Nash Q-learning

- **Theoretical guarantees:**
  - If every stage game encountered during learning has a global optimum, NashQ converges.
  - If every stage game encountered during learning has a saddle point, NashQ converges.
- Both of these are VERY strong assumptions.
- **However..**
  - can converge in practice without these assumptions
  - performs better than independent Q-learning.

Empirical Testing: The Grid-world



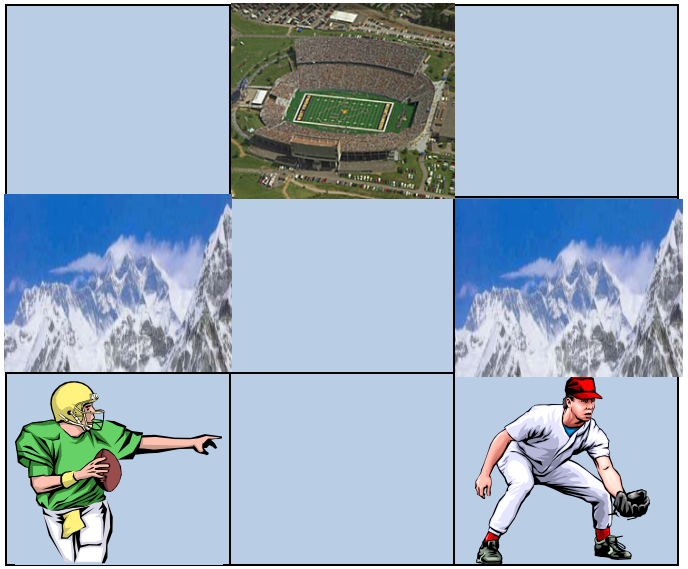
WORLD 1



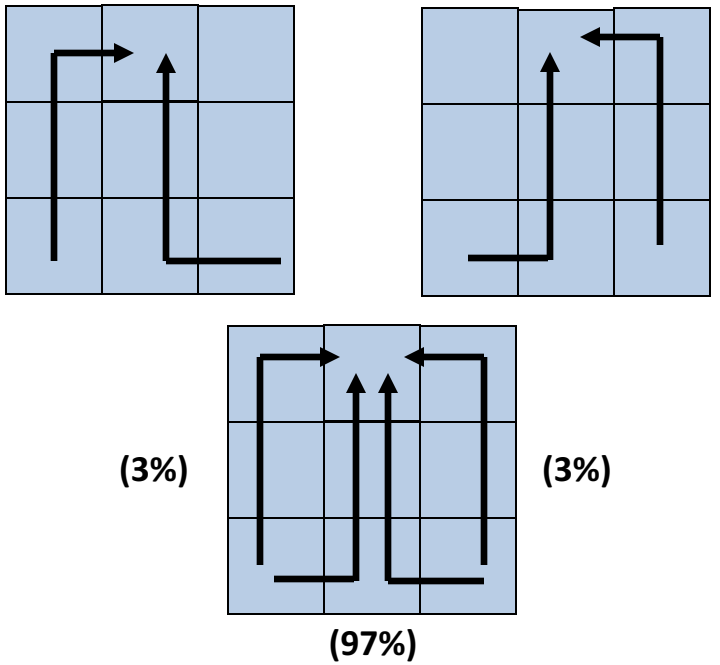
Some Nash Equilibria

See more detailed examples in Hu & Wellman (on VITAL)

Empirical Testing: Nash Equilibria



WORLD 2



All Nash Equilibria

Empirical Performance

- In very small and simple games, NashQ often converged even though theory did not predict so.
- In particular, if all Nash Equilibria have the same value NashQ did better than expected.
- Even if only one agent uses NashQ, convergence is improved.

Gradient Ascent Based Approaches

- Gradient Ascent: update policy directly in the direction of the gradient of its value function
  - Examples: Infinitesimal Gradient Ascent (IGA), Generalised IGA (GIGA)
  - Main idea:

$$\Delta x_i \leftarrow \alpha \frac{\partial V(x)}{\partial x_i}$$

$$x \leftarrow projection(x + \Delta x)$$

- Only suited for (2-player) matrix games

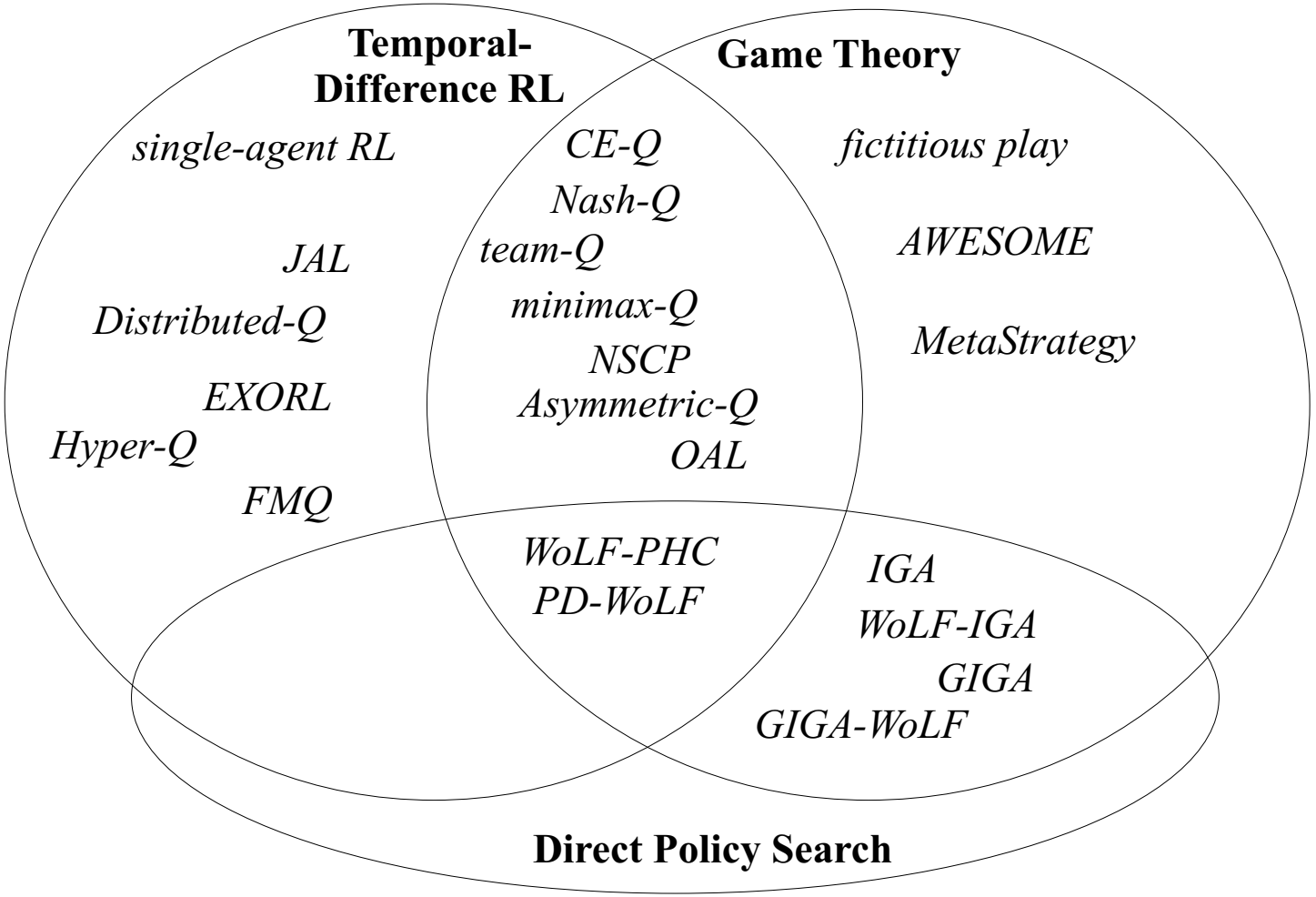
Gradient Ascent Based Approaches

- Can improve convergence using a variable learning rate

- **WoLF**: "Win or Learn Fast"
  - agent reduces its learning rate when performing well, and increases when doing badly
    (compared to an "average" policy)
  - improves convergence of (G)IGA and Policy Hill-Climbing (a variant of Q-learning)

- **Weighted Policy Learner**
  - decrease learning rate unless gradient direction changes
  - this means: learn fast when something unexpected happens

Taxonomy of Multi-Agent Learning Algorithms



See survey paper by Busoniu *et al*. (on VITAL)

Real-World Opportunities

Multi-agent systems where it's hard to do game theory

- Electronic marketplaces
- Mobile networks
- Self-managing computer systems
- Teams of robots
- Video games
- Military/counter-terrorism applications

What's next?

- Understanding convergence, and behaviour, in multi-agent learning is difficult.

- Next, we will study learning dynamics more formally by building a bridge

Multi-Agent Learning ↔ Evolutionary Game Theory