Joint Action Learning

JOSE CHAVEZ

Joint Action Learners (JAL) learn joint-action values and employ empirical models of the other agents’ strategies.

Agent i learns models for all the other agents $j \neq i$, using

$$\hat{\sigma}_j^i(u_j) = \frac{C_j^i(u_j)}{\sum_{\tilde{u}_j \in U_j} C_j^i(\tilde{u}_j)}$$

where $\hat{\sigma}_j^i$ is agent i’s model of agent $j's$ strategy and C_j^i (u_j) counts the number of times agent i observed agent j taking action u_j .

The two forms of multiagent RL

INDEPENDENT LEARNERS

- Apply Q-learning by ignoring the existence of other agents.
- Use one shared policy network for all agents.
- Good results for noncooperative tasks.

JOINT ACTION LEARNERS

- Learn the value of their own actions in conjunction with those of other agents.
- Has the significant drawback that the action space in which the agents must learn scales exponentially in the number of agent.
- Better performance can be achieved in many scenarios like cooperative games.

Why use JALs?

Even though JALs have much more information at their disposal, **they do not perform much differently from ILs** in the straightforward application of Q-learning to MASs.

- **Conditional Joint Action Learning**

Reaching Pareto Optimality by marginal probability → conditional probability

Using a limited exploration technique these agents can actually learn to converge to the Pareto optimal solution that dominates the Nash Equilibrium.

- **Local Joint Action Learning**

LJALs do not coordinate over the joint actions of all agents, but rather coordinate with a specific subset of all agents.

Agents optimize their local joint actions without extensive communication, using global reward.

Conditional Joint Action Learning

Primary obstacle to JAL’s performance improvement **is their assumption that actions of different agents are uncorrelated**, which is not the case in general.

This new learner which understands and tries to use the fact that its own actions affect the action of other agents.

A CJAL tries to learn the correlation between its actions and the other agents' actions and uses conditional probability instead of marginal probability to calculate the expected utility of an action.

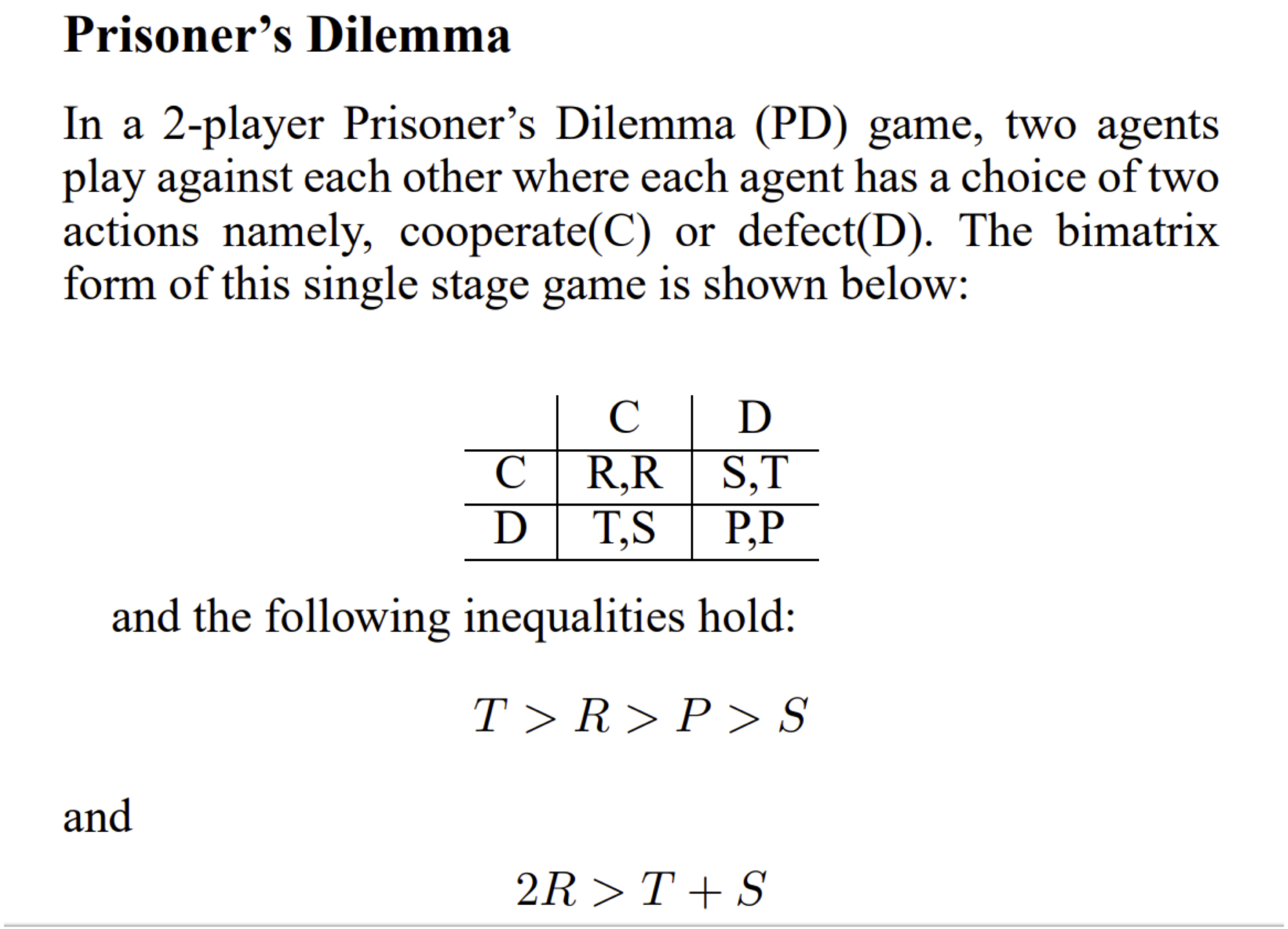
- **Marginal probability** is the probability of an event irrespective of the outcome of another variable.
- **Conditional probability** is the probability of one event occurring in the presence of a second event.

Reaching Pareto Optimality in Prisoner's Dilemma

In this paper they concentrate on two-player games where the agents play with one another repeatedly and tries to learn the optimal action choice which maximize their expected utility.

They are unaware about the duration for which the game will be played.

Therefore, no future discounted rewards while computing their expected utility .



Prisoner's Dilemma

In a 2-player Prisoner's Dilemma (PD) game, two agents play against each other where each agent has a choice of two actions namely, cooperate(C) or defect(D). The bimatrix form of this single stage game is shown below:

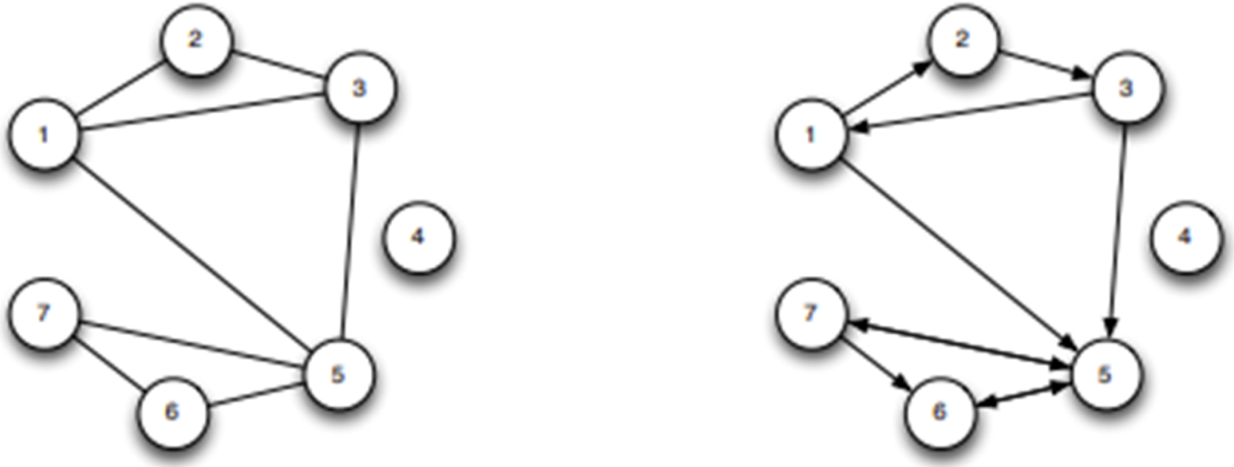
	C	D
C	R,R	S,T
D	T,S	P,P

and the following inequalities hold:

$$T > R > P > S$$

and

$$2R > T + S$$



(a) Undirected Coordination Graph

(b) Directed Coordination Graph

Local Joint Action Learning

THE LOCAL JOINT ACTION LEARNER APPROACH RELIES ON THE CONCEPT OF A COORDINATION GRAPH.

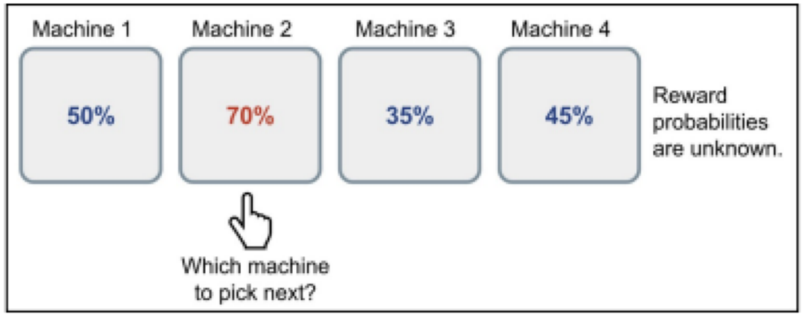
Multi-armed Bandits

- Repeated choice among k actions
  - – Reward from an action-dependent distribution
- k slot machines
  - – Each action is a play on one of the levers
  - – Rewards for hitting one of the jackpots
  - – Through action selections maximize winnings by concentrating actions on the best levers

Every agent must individually decide which of k actions to execute and the reward depends on the combination of all chosen actions.

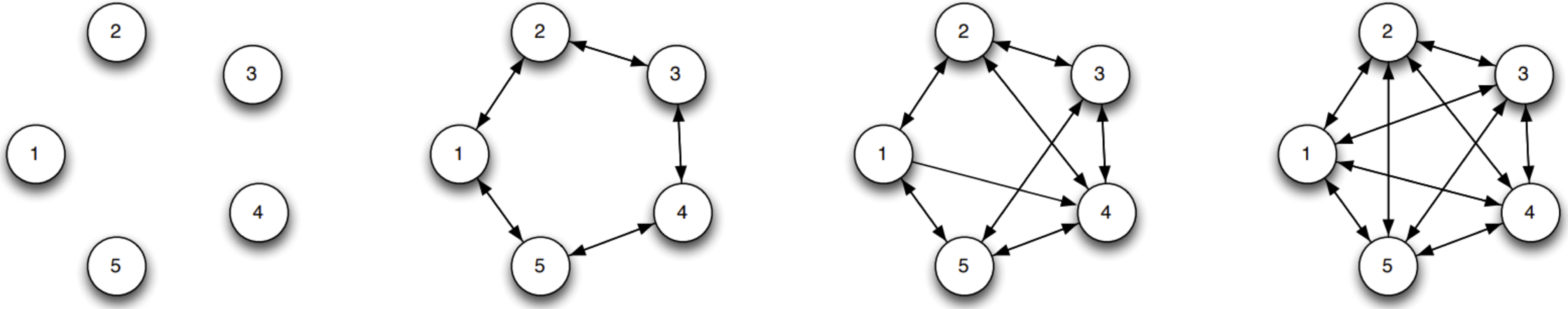
Multi-armed bandit Bernoulli reward

- Each machine provides a random reward
  - Machine-specific distribution unknown a-priori
- Binary case
  - Bernoulli distributions
  - Reward of 1 with probability p, otherwise 0


Machine 1
Machine 2
Machine 3
Machine 4
50%
70%
35%
45%
Reward probabilities are unknown.
Which machine to pick next?

- Maximize expected total reward
  - e.g., over 1000 action selections, or time steps

Local Joint Action Learning



IL

LJAL-2

LJAL-3

JAL

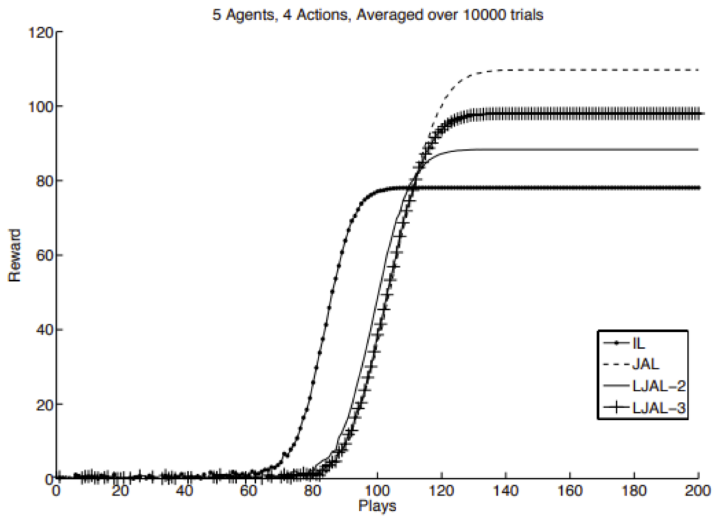
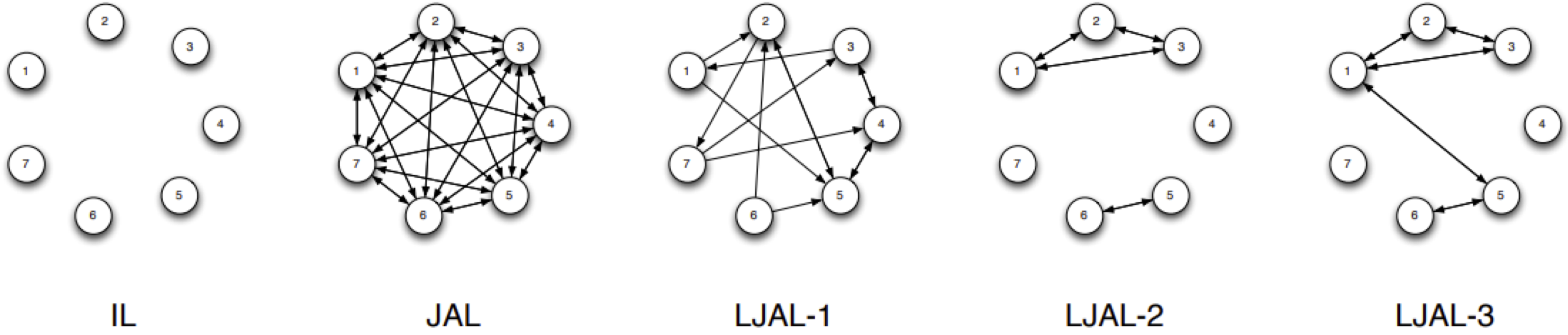


Fig. 4. Comparison of independent learners, joint action learners and local joint action learners on a typical distributed bandit problem.

Learner	Avg # partners	Speed	Solution Quality
IL	0	×31.5	71.1%
LJAL-2	2	×12.1	80.5%
LJAL-3	3	×4.4	89.3%
JAL	4	×1	100%

Table 1. Comparison of speed and solution quality for independent learners, joint action learners and local joint action learners solving a typical distributed bandit problem. All differences are significant, $p < 0.05$.



This approach which is an alternative to Independent Learning (IL) and Joint Action Learning (JAL) based on CGs, where agents optimize their local joint actions without extensive communication, using global reward.

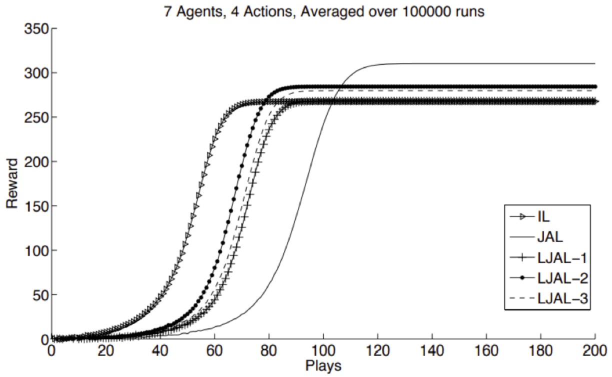


Fig. 7. Comparison of independent learners, joint action learners and local joint action learners on a distributed constraint optimization problem.

Learner	Avg # partners	Speed	Solution Quality
IL	0	×442	86.2%
LJAL-1	2	×172	86.4%
LJAL-2	1.14	×254	91.6%
LJAL-3	1.43	×172	90.2%
JAL	6	×1	100%

Table 2. Comparison of speed and solution quality for independent learners, joint action learners and local joint action learners solving a distributed constraint optimization problem. All differences are significant $p < 0.05$.

Papers related to JALs

- Reaching Pareto Optimality in Prisoner’s Dilemma Using Conditional Joint Action Learning
- Local Coordination in Online Distributed Constraint Optimization Problems
- The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems
- Joint Action Learning for Multi-Agent Cooperation using Recurrent Reinforcement Learning
- A Comprehensive Survey of Multiagent Reinforcement Learning

Next time:

I will be discussing the changes in the Q-function when applying CJAL and LJAL.