



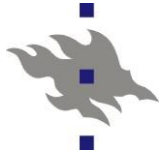
HELSINGIN YLIOPISTO  
HELSINGFORS UNIVERSITET  
UNIVERSITY OF HELSINKI

# Game Theory and Reinforcement Learning with Applications to Routing in Ad Hoc Networks

Petteri Nurmi

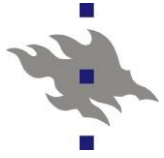
Helsinki Institute for Information Technology HIIT

[petteri.nurmi@cs.helsinki.fi](mailto:petteri.nurmi@cs.helsinki.fi)



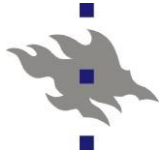
## Outline

- n Part 0: (Very short) Introduction to Game Theory
- n Part I: Learning in Games
  - n Pavlovian Reinforcement Models
  - n Myopic Learning
  - n Fictitious play
  - n Connections to Reinforcement Learning
  - n Example: Routing in Ad Hoc Networks
- n Part II: Equilibrium Selection
  - n Independent Action Learners
  - n Joint Action Learners
  - n WoLF



## What is a game?

- n A game models interactions between multiple decision makers, called agents
  - n The decisions made by the agents define an outcome for the game
  - n Different outcomes are valued differently by the players, i.e., some are more desirable than others
- n Assuming that players are rational and want to maximize their own utility, the best prediction for the outcome of a game is the equilibrium of the game
  - n A point where none of the agents has a motivation to choose its strategy differently
- n To illustrate the concepts, we consider a famous example, the prisoners' dilemma



## Prisoners' dilemma

Strategy

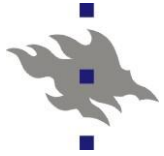
Socially optimal outcome

Payoff

I/II	C	D
C	(1,1)	(-1,2)
D	(2,-1)	(0,0)

Nash Equilibrium

Detailed description: A 2x2 payoff matrix for the Prisoners' Dilemma. The rows and columns are labeled 'C' and 'D'. The payoffs are (1,1) for (C,C), (-1,2) for (C,D), (2,-1) for (D,C), and (0,0) for (D,D). The (1,1) payoff is highlighted in pink, and the (0,0) payoff is highlighted in red. Labels with arrows point to the matrix: 'Strategy' points to the column headers, 'Socially optimal outcome' points to the (1,1) cell, 'Payoff' points to the (-1,2) cell, and 'Nash Equilibrium' points to the (0,0) cell.



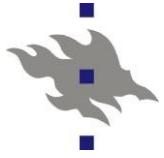
## Further Examples

n Matching pennies

D/S	H	T
H	(-1,1)	(1,-1)
T	(1,-1)	(-1,1)

n The penalty game ( $k \ll 0$ )

I/II	a	b	c
a	10	0	k
b	0	2	0
c	k	0	10



## Questions

- n How can players learn to behave optimally over the course of time?
- n How can socially optimal behavior emerge?
- n How can we learn stochastic policies?
- n How can we ensure that learning can discriminate between alternative equilibriums?

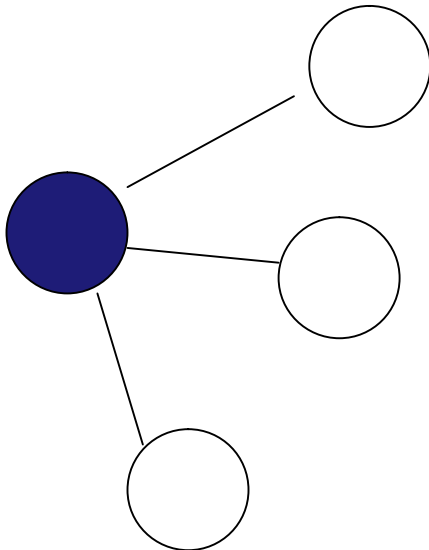


## Example problem: Routing in Ad Hoc Networks

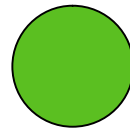
n The following problem will be used as an example throughout the presentation:

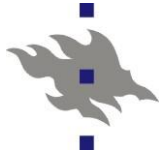
n Some node, called the *source* (blue), wants to send packets to a *destination* or *sink* (green) and it wants to optimize the *route* that the packets take

- We focus on the case where the node optimizes only the first hop
- And we assume that the decisions (potentially) depend on factors such as selfishness, energy etc.

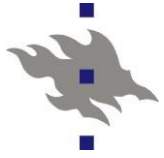


...





# Part I: Learning in Games



## Pavlovian Reinforcement

- n The simplest possible learning model, players adjust their strategies mechanically based on external stimuli
  - n The stimuli is determined by the payoff received from executing a particular action
  - n Example:
    - Let  $\theta_{ik}$  denote the *propensity* with which player  $i$  adopts strategy  $k$ .
    - Assume payoffs are strictly positive, i.e.,  $\pi_i(s_{ik}, s_{jl}) > 0$
    - The propensities induce the following behavior strategy:

$$\sigma_{ik} = \theta_{ik} / \sum_k \theta_{ik}$$



## Pavlovian Reinforcement

n Assume players adjust their strategies using the rule:

$$\theta_{ik}(t+1) = \theta_{ik}(t) + \psi_{ik}$$

- where  $\psi_{ik}$  is the payoff to strategy  $\theta_{ik}$  (or zero if the strategy was not adopted)

n It turns out that the *expected motion* of the strategies can be approximated using the so-called continuous time *replicator dynamics*.

n Furthermore, using stochastic approximation theory it can be shown that the strategies converge

n to a strict equilibrium point

n or to a cycling path



## Pavlovian Reinforcement

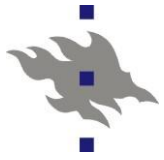
- n Many alternative formulations have been suggested
  - n E.g., players can track the accumulated average payoff
    - Strategies are adjusted based on how well they fare against the average payoff
    - In this case, the average payoff forms the *aspiration level* of a player
- n A problem with the Pavlovian model is that although the strategies of the individuals converge, nothing can be said about the *joint distribution* of the strategies
  - n This is not guaranteed to converge and indeed there are games where the Pavlovian model does not converge to an equilibrium



## Example of Pavlovian Models

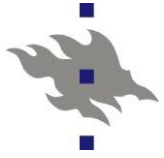
- n A typical example of Pavlovian model is the Independent Action Learn (IAL) framework in multi-agent reinforcement learning
  - n Multi-agent settings where agents
    - have no knowledge about the other player or the payoffs
    - attempt to optimize their behavior over time
  - n Example: prisoners' dilemma with non-observable actions
    - At the end of each iteration, the players are informed only of the payoff they receive
    - Used as a (hard) example of cases when cooperation is unlikely to emerge

I/II	C	D
C	(1,1)	(-1,2)
D	(2,-1)	(0,0)



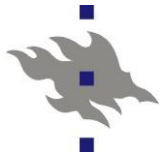
## Myopic Learning 1/3

- n A more sophisticated form of learning occurs when
  - n The agents are aware of the existence of other agents
  - n The agents model the decisions of other agents and act optimally according to their beliefs
  - n Agents are able to experiment with alternative choices
  
- n Example:
  - n Let  $\sigma_j$  be the behavior strategy of player  $j$
  - n Let  $\phi_i^j(t)$  denote player  $i$ 's estimate of the behavior strategy of player  $j$  at time  $t$
  - n At time  $t+1$ , player  $i$  acts as if player  $j$  would act according to  $\phi_i^j(t)$  and optimizes its own play according to this choice
  - n At the end of the game, players update their estimates



## Myopic Learning 2/3

- n If the players merely select the (seemingly) optimal action at all stages of game, the process is not ensured to converge to an equilibrium
  - n Intuitively, the process does not guarantee that the behavior strategy that yields the equilibrium strategy is ever tried  $\Rightarrow$  the process can not converge to it
- n To overcome this deficiency, the players are supposed to experiment
  - n At any given time  $t$ , there is a (strictly) positive probability that the players select a suboptimal action
  - n The probability for selecting a suboptimal action is required to converge to zero over time



## Myopic Learning 3/3

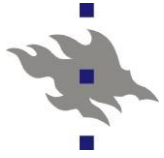
### n Convergence?

- n If players experimentation is "patient" enough, the process converges to some equilibrium point
- n But what is then patient enough?
  - Stochastic approximation theory, all actions should be explored infinitely often
  - I.e., let  $\varepsilon_t$  be the probability of (some agent) selecting a suboptimal action at time  $t$
  - Then stochastic approximation theory works if  $\sum \varepsilon_t = \infty$  and  $\sum \varepsilon_t^2 < \infty$
- n However, this is **not** sufficient for multi-agent settings!!!
  - Instead, we must require that these conditions are satisfied for the **joint distribution**



## Example of Myopic Learning: Routing

- n We return now to the routing example
  - n Assume the decisions of the intermediate nodes are parameterized and depend, e.g., on their selfishness
  - n The goal of the source node then is to
    - Estimate the parameters (e.g., selfishness) over time
    - Optimize its behavior based on its current estimates
  - n A simple exploration rule, such as  $1/\epsilon_t$  is sufficient for this example, if we assume the decisions of the intermediate nodes to have **full support**
    - I.e., all actions have a strictly positive probability of being taken
  - n In this case, we can show that the behavior of the nodes converges to a (sequential) equilibrium point



## Fictitious Play 1/2

- n The myopic framework is more sophisticated than the Pavlovian framework, but it suffers from one major limitation
  - n Namely, the process requires that the decisions of the agents are statistically independent
  - n In this case, it can be ensured that at some point of time, the equilibrium strategy is played, after which the learning dynamics converge towards that point
    - ...since the equilibrium point is a stable rest point of the learning dynamics
- n Fictitious play attempts to overcome this problem by modeling instead directly the **joint** distribution of actions



## Fictitious Play 2/2

n As an example, we return to the prisoners' dilemma

n In the Pavlovian model, we simply attempt to learn the *payoff* from each individual action and act accordingly

n In the myopic model, we attempt to learn the strategy of player II also, but we don't associate it with our own actions

I/II	C	D
C	(1,1)	(-1,2)
D	(2,-1)	(0,0)

- Thus, the behaviors are modeled independently of our choice
- E.g., player II chooses C with probability 0.5 and D with probability 0.5

n In Fictitious play, our beliefs are conditioned

- E.g., player II chooses C when player I chooses C with probability 0.2



## Connections to Reinforcement Learning

- Before we consider the relationships between the different learning models and reinforcement learning, we consider the "anatomy" of a reinforcement learning algorithm
- Typical example, Q-learning:

$$\Delta Q(s,a) = \alpha[r + \gamma \max_b Q(s',b) - Q(s,a)]$$

$Q(s,a)$  is the value of taking action  $a$  in state  $s$

$r$  is the reward

$\alpha$  is the learning rate

$\gamma$  is the discount factor

$\max_b Q(s',b)$  is the value of the action that maximizes the Q-value in the next state  $s'$



## Connections to Reinforcement Learning

- Using Q-learning requires that we know the state dynamics, i.e., to which state we get from the current state
- On the other hand, if we set the discount term  $\gamma$  to zero, the equation reduces to

$$\Delta Q(s,a) = \alpha[r - Q(s,a)]$$

which is equivalent to the (Robbins-Munro) stochastic approximation algorithm

Then the convergence of the estimates follows from

Martingale equalities (in the theory of stochastic process)

- Thus, many reinforcement learning algorithms can be considered of consisting of
  - a stochastic approximation component
  - a dynamic programming component



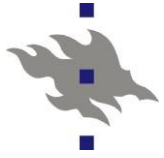
## Connections to Reinforcement Learning

- n The Pavlovian and Myopic Learning models are simple:
  - n The dynamics are assumed unknown  $\Rightarrow$  no dynamic programming component
  - n Thus corresponds to stochastic approximation
  - n But, contrary to the standard application domain, the environment where the algorithms are applied is **non-stationary**
  - n The analysis thus requires that the joint distribution of the beliefs satisfies conditions that ensure convergence
- n The Fictitious play model
  - n performs a one step approximation to the dynamic programming component



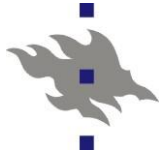
## Routing in Ad Hoc Networks

- n The models are also based on the assumption that parameters of interest remain fixed
  - è dynamic programming could be used in full generality, but the predictions could be wrong as the parameters change
  
- n We return now to the problem of routing in ad hoc networks and assume
  - n That the behavior of intermediate nodes depends on a set of parameters  $\{\theta_{jk}\}$
  - n That the parameters change over time according to some dynamics that we assume satisfy the Markov assumption

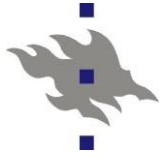


## Routing in Ad Hoc Networks

- n In this case, we can
  - n Try to learn a model of the environment dynamics
    - E.g., using sequential Bayesian modeling
  - n And attempt to optimize the performance of the nodes considering both discounting and future actions
    - The next states are fully determined by the environment dynamics
    - The assumption that the behavior of the nodes depends on a set of parameters ensures that the behavior of the intermediate nodes is characterized by the real values of the estimated parameters

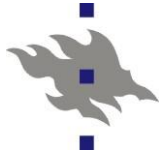


# Part II: Equilibrium Selection



## Problem setting

- n A problem with the learning approaches is that they can only guarantee convergence to **some** equilibrium
  - n But no guarantees can be given that it converges to a specific equilibrium point
  - n For example, in economically motivated applications we are usually interested in the **Pareto optimal** equilibrium
    - I.e., the equilibrium that allocated all resources in the "economy" optimally
    - In other words, the globally (and socially) optimal solution



## Independent action learners

- n We begin by considering the framework of independent action learners
  - n I.e., agents cannot observe the actions of the others
  - n The only feedback thus consists of the reward
  
- n In this kind of setting, the only hope of the agents is to use some form of exploration that
  - n avoids unnecessary risks
  - n but explores "enough"
  - n ...thus has to balance both in some suitable way



## Independent Action Learners

From theoretical perspective, IALs are difficult:

- Consider the so-called *penalty-game* given below ( $k \ll 0$ )
- Many learning algorithms are able to learn the policy (a,a) or (c,c), but

- They usually gather a large regret before learning the (Pareto-)optimal policy
- Thus, when the game is repeated a finite (but the exact number of repetitions is unknown to the agents) amount of time, the greedy policy (b,b) almost always yields a better overall result

		Agent 1		
		a	b	c
Agent 2	a	10	0	k
	b	0	2	0
	c	k	0	10

- We say that the action b *risk-dominates* the choice of action a (or c)



## Independent action learners: FMQ (Frequency Maximum Q Value)

- As an example, we consider the FMQ rule
  - In Q-learning, actions are selected according to some criterion that reflects their desirability
  - One of the standard choices is the softmax action rule:  
$$P(\text{action}) = \frac{e^{(Q(\text{action}) / T)}}{\sum e^{(Q(a') / T)}}$$
  - FMQ modifies this by using an alternative value EV instead of the Q-values in the softmax rule
  - The EV value is defined as:

$$EV(\alpha) = Q(\alpha) + c * \text{freq}(\max R(\alpha)) * \max R(\alpha)$$

$$P(\text{action}) = \frac{e^{\frac{EV(\text{action})}{T}}}{\sum_{\text{action}' \in A_i} e^{\frac{EV(\text{action}')}{T}}}$$



## Joint action learners

- n Alternative is the joint action learning (JAL) framework
  - n Corresponds to Fictitious play
  - n Agents observe a payoff and information about what the agents performed in the previous round
  
- n Example: A Bayesian approach
  - n Attempt to model the underlying MDP of the decision problem using Bayesian methodology
  - n Perform myopic expected value of information calculations to determine the optimal action
    - I.e., estimate the most likely state of the MDP
    - And perform value iterations using the estimated model and the given belief state



## WoLF(-PHC)

- n As a special case that can be applied either to IAL or to JAL setting, we consider the WoLF algorithm
  - n WoLF = Win or Lose fast
  - n A stochastic approximation algorithm, where the speed of learning depends on whether we are "winning" or "losing"
    - In the terminology of the Part I, we define winning to take place, when the reward exceeds our aspiration level
    - Respectively, when this is not the case, we are losing
  - n In WoLF, the policy of a player is adjusted as in stochastic approximation, but with the exception that the we have two stepsize parameters  $\alpha_w$  and  $\alpha_l$ 
    - Require that  $\alpha_w < \alpha_l$
    - If winning, use  $\alpha_w$  otherwise  $\alpha_l$
    - Motivation, when players are winning, both are adjusting their strategies  $\Rightarrow$  ensures careful enough adjustments