Correlated Q-Learning

&

No-Regret Q-Learning

Amy Greenwald

with

David Gondek

Keith Hall

City University of New York

Logic and Games Seminar

February 28, 2003

Part I

Multiagent Q-Learning

- Correlated-Q Learning
  - converges (empirically) to equilibrium policies
- Nash-Q [Hu and Wellman, 1998]
  - converges (empirically), but not necessarily to equilibrium policies
- Minimax-Q [Littman, 1994]
  - converges (analytically) to equilibrium policies in constant-sum games

AI Agenda Learn Q-Values

Part II

Approximate Q-Learning

- No-regret Q-learning
  - No external regret learning
    - converges to minimax strategies in constant-sum games
  - No internal regret learning
    - converges to correlated equilibrium in general-sum games

GT Agenda Learn Equilibria

Markov Decision Processes (MDPs)

Decision Process

- S is a set of states ($s \in S$)
- A is a set of actions ($a \in A$)
- $R : S \times A \to \mathbb{R}$ is a reward function
- $P[s_{t+1}|s_t, a_t, \ldots, s_0, a_0]$ is a probabilistic transition function that describes transitions between states, conditioned on past states and actions

MDP $=$ Decision Process $+$ Markov Property:

$$P[s_{t+1}|s_t, a_t, \ldots, s_0, a_0] = P[s_{t+1}|s_t, a_t]$$

$\forall t,\ \forall s_0, \ldots, s_t \in S,\ \forall a_0, \ldots, a_t \in A$

Bellman's Equations

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P[s'|s,a] V^*(s')$$

$$V^*(s) = \max_{a \in A(s)} Q^*(s,a)$$

Value Iteration

VALUE_ITERATION(MDP, γ)	
Inputs	discount factor γ
Output	optimal state-value function V^*
	optimal action-value function Q^*
Initialize	$V = Q = 0$

```
REPEAT
    for all s ∈ S
        for all a ∈ A
            Q(s,a) = R(s,a) + γ Σ_{s'} P[s'|s,a] V(s')
        V(s) = max_a Q(s,a)
FOREVER
```

Q-Learning

```
Q_LEARNING(MDP, γ, α, ε)
 Inputs       discount factor γ
              rate of averaging α
              rate of exploration ε
 Output       optimal state-value function V*
              optimal action-value function Q*
 Initialize   V = Q = 0
```

```
REPEAT
    initialize s, a
    WHILE s is nonabsorbing DO
        simulate action a in state s
        observe reward R and next state s'

        compute V(s') = max_{a∈A(s)} Q(s, a)
        update Q(s, a) = (1 − α)Q(s, a) + α[R + γV(s')]

        choose action a' (on- or off-policy)
        s = s', a = a'
        decay α
FOREVER
```

Theorem [Watkins, 1989]

Q-learning converges to V^* and Q^*

Markov Games

Stochastic Game

- I is a set of n players ($i \in I$)
- S is a set of states ($s \in S$)
- $A_i(s)$ is the ith player's set of actions at state s
  let $A(s) = A_1(s) \times \ldots \times A_n(s)$ ($\vec{a} \in A(s)$)
- $P[s_{t+1}|s_t, \vec{a}_t, \ldots, s_0, \vec{a}_0]$ is a probabilistic transition function that describes transitions between states, conditioned on past states and actions
- $R_i(s, \vec{a})$ is the ith player's reward at state s for action vector $\vec{a}$

Markov Game = Stochastic Game + Markov Property:

$$P[s_{t+1}|s_t, \vec{a}_t, \ldots, s_0, \vec{a}_0] = P[s_{t+1}|s_t, \vec{a}_t]$$

$\forall t$, $\forall s_0, \ldots, s_t \in S$, $\forall \vec{a}_0, \ldots, \vec{a}_t \in A$

Bellman's Analogue

$$Q_i^*(s,\vec{a}) = R_i(s,\vec{a}) + \gamma \sum_{s'} P[s'|s,\vec{a}] V_i^*(s')$$

Foe-Q

$V_1^*(s) = \max_{\sigma_1 \in \Sigma_1(s)} \min_{a_2 \in A_2(s)} Q_1^*(s,\sigma_1,a_2) = -V_2^*(s)$

Friend-Q $\quad V_i^*(s) = \max_{\vec{a} \in A(s)} Q_i^*(s,\vec{a})$

Nash-Q $\quad V_i^*(s) \in \mathrm{Nash}_i(Q_1^*(s),\ldots,Q_n^*(s))$

CE-Q $\quad V_i^*(s) \in \mathrm{CE}_i(Q_1^*(s),\ldots,Q_n^*(s))$

Multiagent Q-Learning

MULTIQ(MGame, γ, α, ϵ)

REPEAT
    initialize $s, a_1, \ldots, a_n$
    WHILE s is nonabsorbing DO
        simulate actions $a_1, \ldots, a_n$ in state s
        observe rewards $R_1, \ldots, R_n$ and next state s'
 	
        for all $i \in I$
            compute $V_i(s')$
            update $Q_i(s, a_1, \ldots, a_n)$
 	
        (simultaneously) choose actions $a'_1, \ldots, a'_n$
        $s = s'$, $a_1 = a'_1, \ldots, a_n = a'_n$
        decay α
FOREVER

Nash-Q converges (empirically)

not necessarily to equilibrium policies

FF-Q converges (analytically)

to equilibrium policies in restricted classes of games

CE-Q converges (empirically)

to equilibrium policies

Why CE?

- easily computable via linear programming, unlike Nash equilibrium
- players can achieve payoffs outside the convex hull of Nash payoffs [Aumann, 74]
- players learn correlated equilibrium via no-regret algorithms [Foster & Vohra, 99]
- consistent with the usual AI view of individually rational behavior

Why NOT (Nash or) CE?

- equilibrium selection problem

Correlated Equilibrium

Chicken

	L	R
T	6,6	2,7
B	7,2	0,0

CE

	L	R
T	1/2	1/4
B	1/4	0

$$\max 12\pi_{TL} + 9\pi_{TR} + 9\pi_{BL} + 0\pi_{BR}$$

subject to probability constraints

$$\pi_{TL} + \pi_{TR} + \pi_{BL} + \pi_{BR} = 1$$
$$\pi_{TL}, \pi_{TR}, \pi_{BL}, \pi_{BR} \geq 0$$

& individual rationality constraints

$$\begin{array}{rcl} 6\pi_{LT} + 2\pi_{RT} & \geq & 7\pi_{LT} + 0\pi_{RT} \\ 7\pi_{LB} + 0\pi_{RB} & \geq & 6\pi_{LB} + 2\pi_{RB} \\ 6\pi_{TL} + 2\pi_{BL} & \geq & 7\pi_{TL} + 0\pi_{BL} \\ 7\pi_{TR} + 0\pi_{BR} & \geq & 6\pi_{TR} + 2\pi_{BR} \end{array}$$

$$\mathsf{CE}_i(Q_1(s),\ldots,Q_n(s)) = \{\textstyle\sum_{\vec{a}\in A}\sigma^*(\vec{a})Q_i(s,\vec{a}) \mid \sigma^* \text{ satisfies Eq. 1, 2, 3, or 4}\}$$

- Utilitarian maximize the sum of rewards

$$\sigma^* \in \arg\max_{\sigma\in\mathsf{CE}} \sum_{i\in I}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{1}$$

- Egalitarian maximize the minimum reward

$$\sigma^* \in \arg\max_{\sigma\in\mathsf{CE}} \min_{i\in I}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{2}$$

- Republican maximize the maximum reward

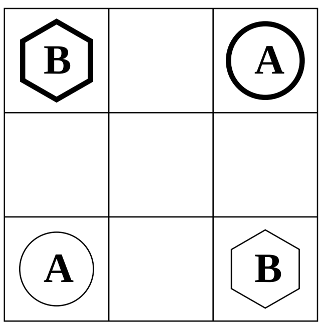
$$\sigma^* \in \arg\max_{\sigma\in\mathsf{CE}} \max_{i\in I}\sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{3}$$

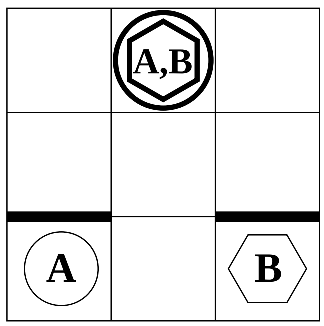
- Libertarian i maximizes only i's rewards
  let $\sigma^* = \prod_i \sigma^i$ where

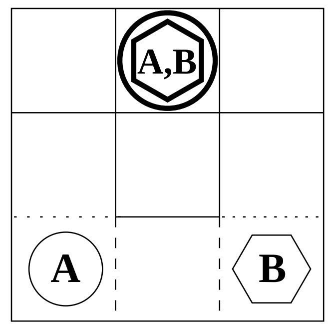
$$\sigma^i \in \arg\max_{\sigma\in\mathsf{CE}} \sum_{\vec{a}\in A}\sigma(\vec{a})Q_i(s,\vec{a}) \tag{4}$$

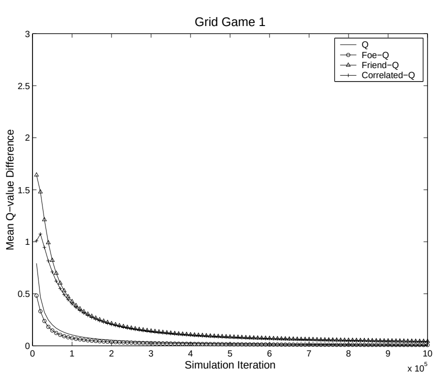
Grid Games

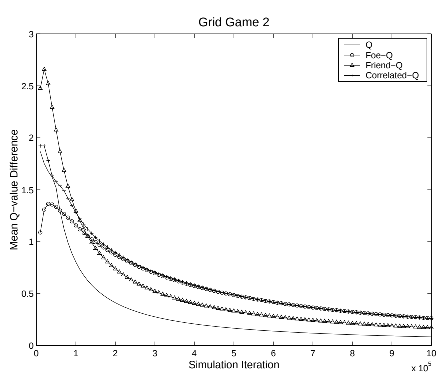
GG1 GG2 GG3

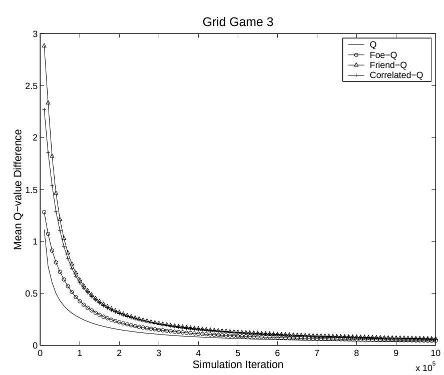












Equilibrium Policies

Grid Games	GG1		GG2		GG3	
Algorithm	Score	Games	Score	Games	Score	Games
Q	100,100	2500	49,100	3333	100,125	3333
Foe-Q	0,0	0	67,68	3003	120,120	3333
Friend-Q	$-10^4, -10^4$	0	$-10^4, -10^4$	0	$-10^4, -10^4$	0
uCE-Q	100,100	2500	50,100	3333	116,116	3333
eCE-Q	100,100	2500	51,100	3333	117,117	3333
rCE-Q	100,100	2500	100,49	3333	125,100	3333
lCE-Q	100,100	2500	100,51	3333	$-10^4, -10^4$	0

Part I

Correlated-Q Learning

- good news
  - converges (empirically) to an equilibrium policy
- bad news
  - equilibrium policy is path dependent
    i.e., dynamics are nonergodic

Part II

No-regret Q-Learning

- No-external-regret
  - converge to minimax strategies
    in constant-sum games
- No-internal-regret
  - converge to correlated equilibrium
    in general-sum games

Repeated Games

A game is a tuple $\Gamma = (I, (A_i, R_i)_{i \in I})$ where

- I is a set of players ($i \in I$)
- A_i is a set of pure actions ($a_i \in A_i$)
- $R_i : A \to \mathbb{R}$ is a reward function ($a \in A = \prod_i A_i$)

A repeated game is a sequence of tuples Γ^T or Γ^∞

No-Regret Definitions

Regret is the difference in rewards for playing action a_i' rather than a_i at time t:

$$\rho_i^t(a_i, a_i') = \pi_i^t(a_i') \left[R_i(a_i, a_{-i}^t) - R_i(a_i', a_{-i}^t) \right]$$

A learning algorithm exhibits no-external-regret iff it generates weights $\{\pi_i^t\}$ *s.t.* for all opposing policies, there exists T *s.t.* for all $T > T_0$,

$$\max_{a_i \in A_i} \frac{1}{T} \sum_{t=1}^{T} \sum_{a_i' \in A_i} \pi_i^t(a_i') \rho_i^t(a_i, a_i') \leq \mathrm{ERR}(T)$$

where $\mathrm{ERR}(T) \to 0$ as $T \to \infty$.

A learning algorithm exhibits no-internal-regret iff it generates weights $\{\pi_i^t\}$ *s.t.* for all opposing policies, there exists T *s.t.* for all $T > T_0$,

$$\max_{a_i \in A_i} \frac{1}{T} \sum_{a_i' \in A_i} \left(\sum_{t=1}^{T} \pi_i^t(a_i') \rho_i^t(a_i, a_i') \right)^+ \leq \mathrm{ERR}(T)$$

where $\mathrm{ERR}(T) \to 0$ as $T \to \infty$ and $X^+ = \max\{X, 0\}$.

No-Regret Algorithms

$$\mathcal{R}_i^t(a_i', a_i) = \frac{1}{t} \sum_{x=1}^{t} \rho_i^x(a_i', a_i)$$

$$\mathcal{X}_i^t(a_i) = \sum_{a_i' \in A_i} \mathcal{R}_i^t(a_i', a_i) \qquad \mathcal{W}_i^t(a_i) = \sum_{a_i' \in A_i} \pi_i^t(a_i') \mathcal{R}_i^t(a_i', a_i)$$

Hart and Mas-Colell (HMC)

$$\pi_i^{t+1}(a_i) = \frac{[\mathcal{X}_i^t(a_i)]^+}{\sum_{a_i \in A_i} [\mathcal{X}_i^t(a_i)]^+}$$

NER learning approximates minimax equilibria

[Freund and Schapire, 1996]

WAR Weighted Average Regret (WAR)

$$\pi_i^{t+1}(a_i) = \frac{[\mathcal{W}_i^t(a_i)]^+}{\sum_{a_i \in A_i} [\mathcal{W}_i^t(a_i)]^+}$$

NIR learning approximates correlated equilibria

[Foster and Vohra, 1997]

Normal Form Games

Matching Pennies

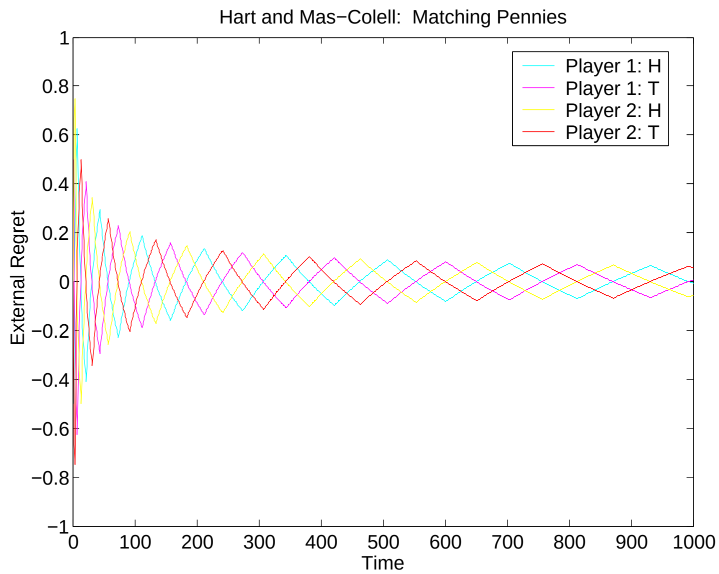
1 \ 2	H	T
H	1,0	0,1
T	0,1	1,0

Shapley Game

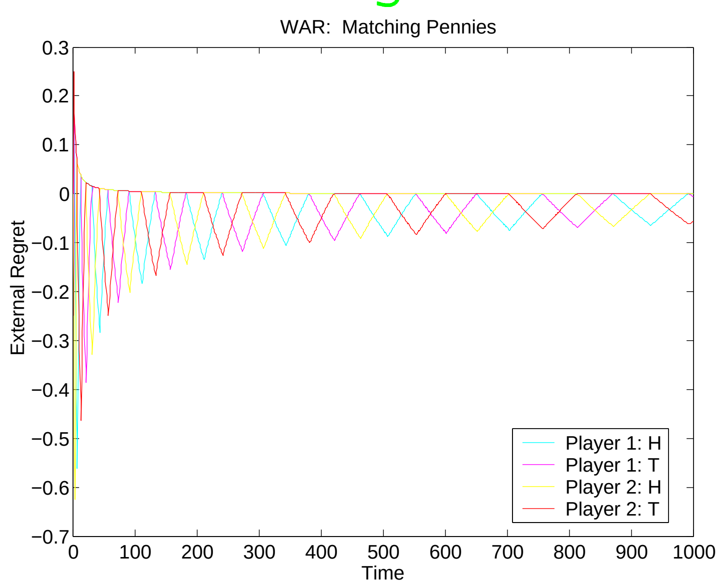
1 \ 2	L	C	R
T	1,0	0,1	0,0
M	0,0	1,0	0,1
B	0,1	0,0	1,0

Matching Pennies

External Regret: HMC

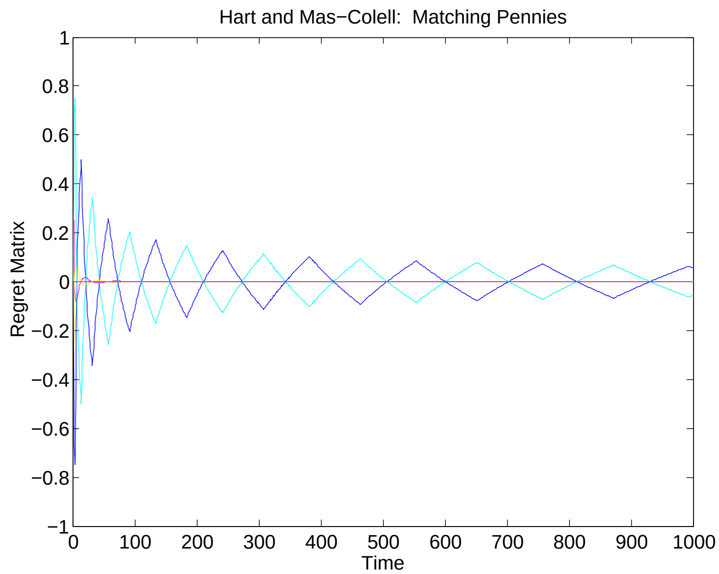


External Regret: WAR

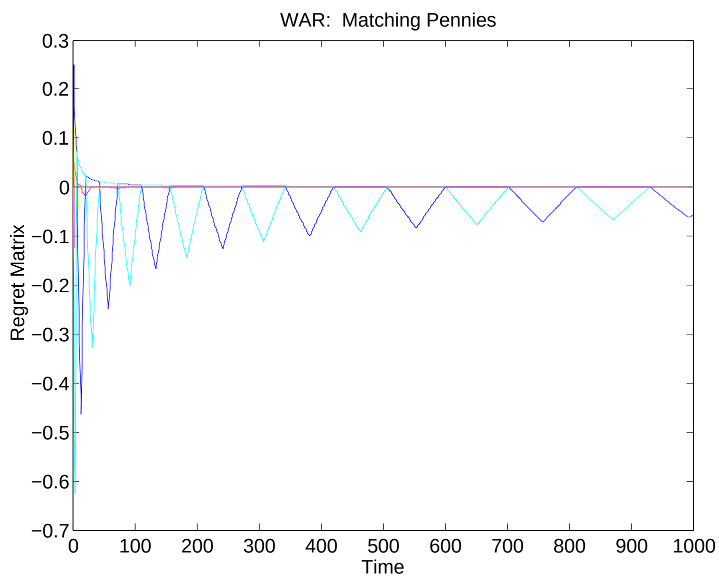


Matching Pennies

Internal Regret: HMC

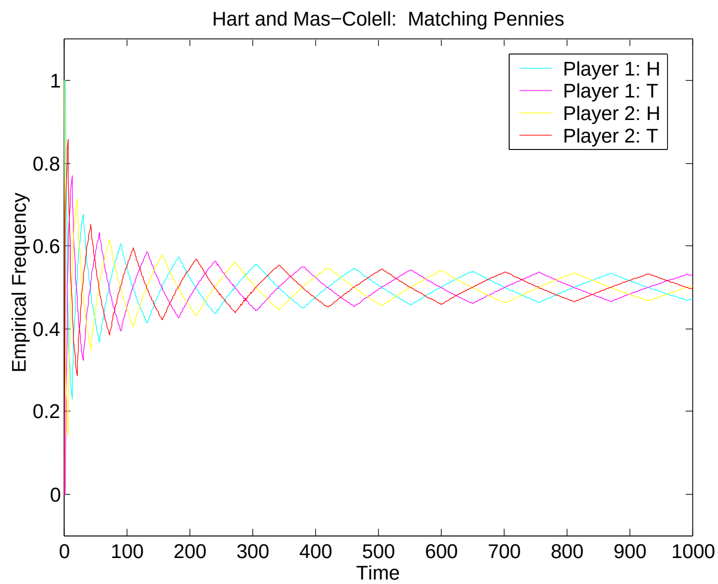


Internal Regret: WAR

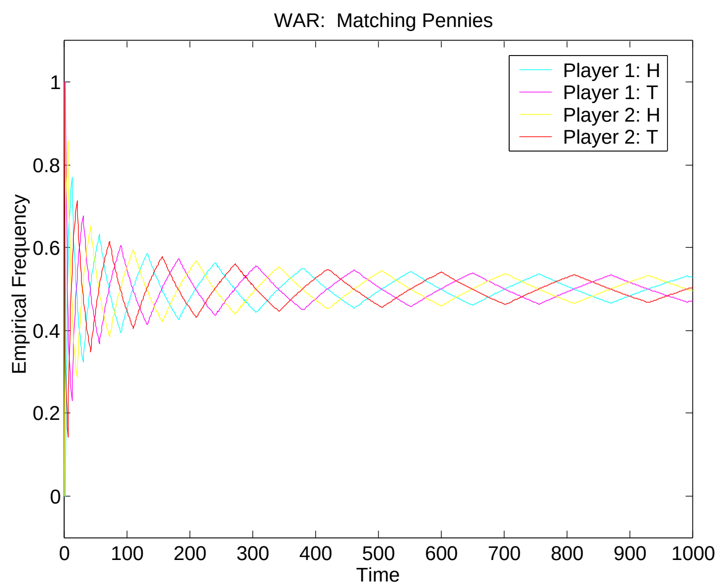


Matching Pennies

Frequencies: HMC

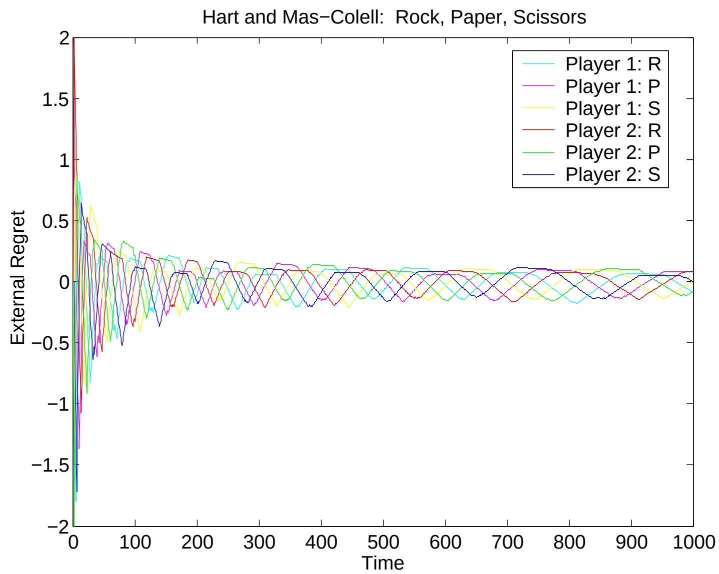


Frequencies: WAR

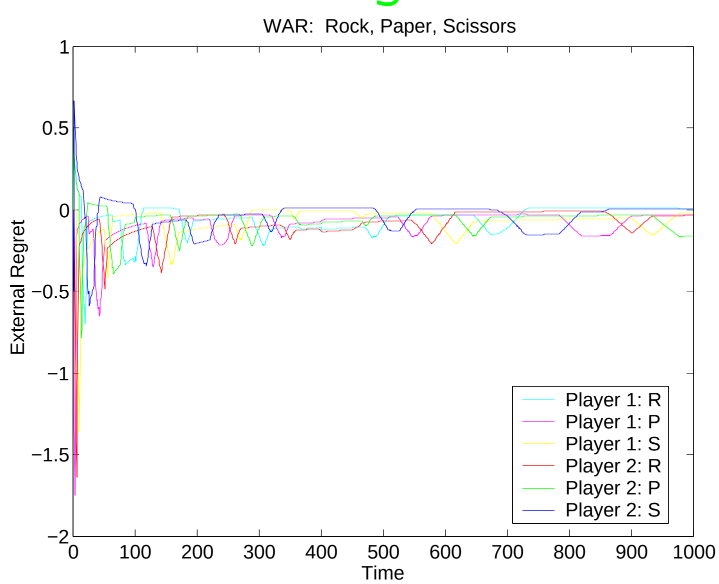


Rochambeau

External Regret: HMC

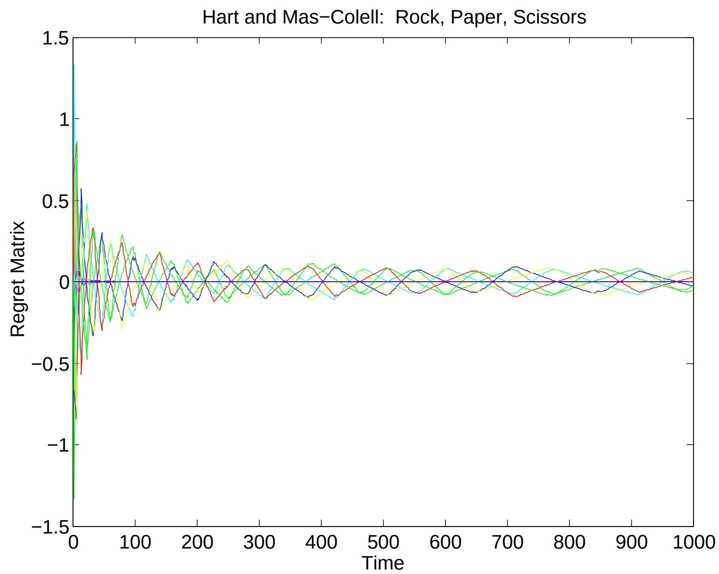


External Regret: WAR

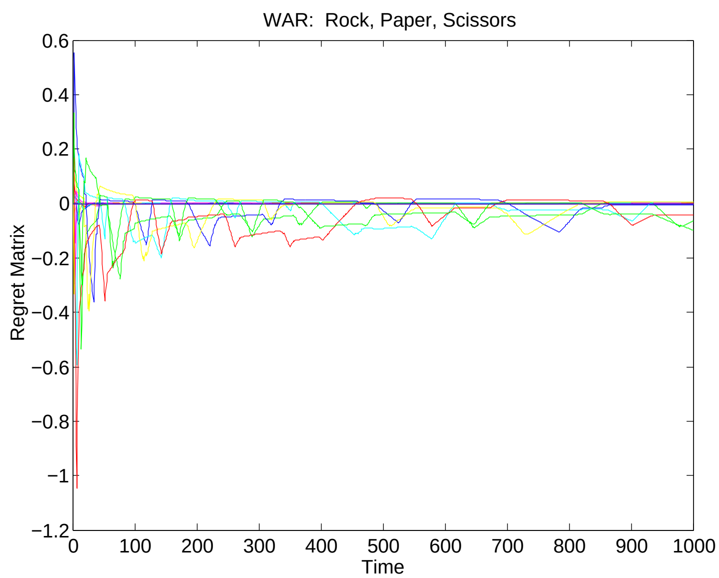


Rochambeau

Internal Regret: HMC

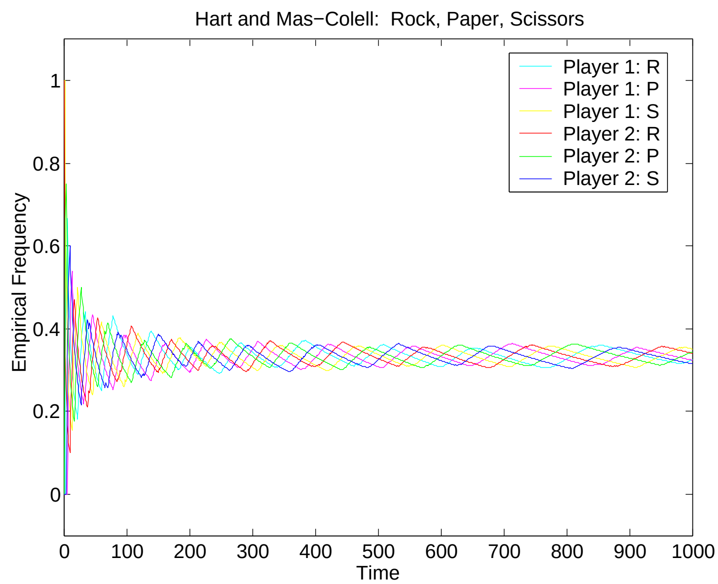


Internal Regret: WAR

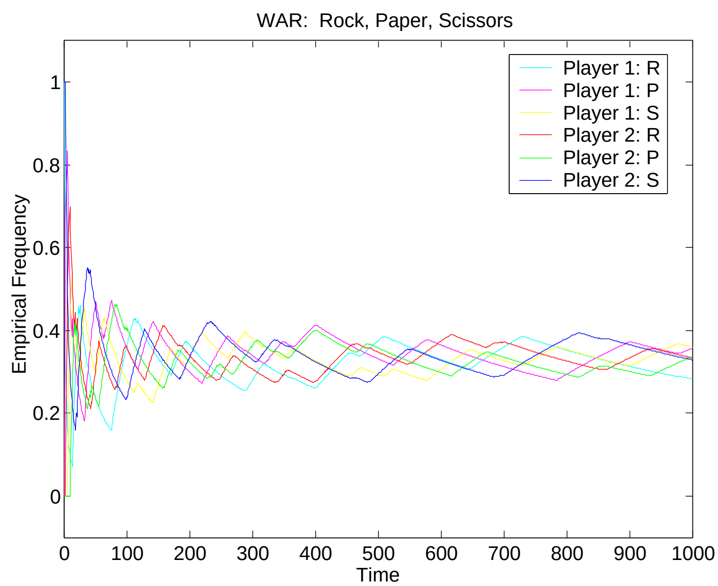


Rochambeau

Frequencies: HMC

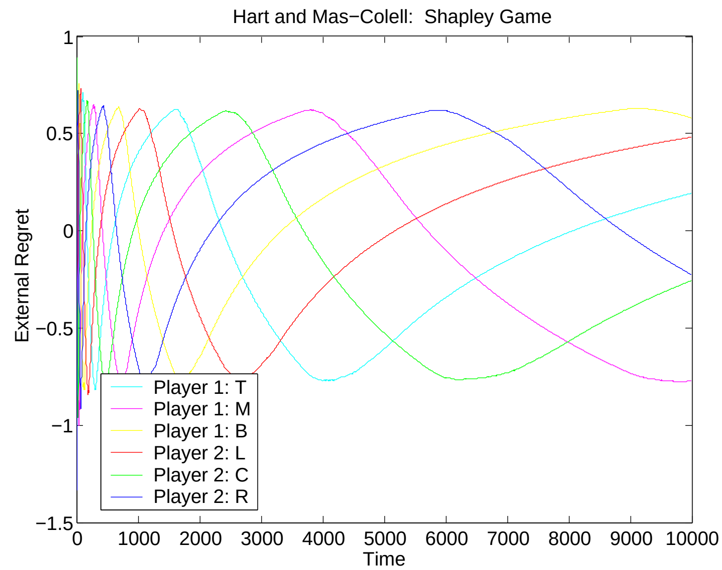


Frequencies: WAR

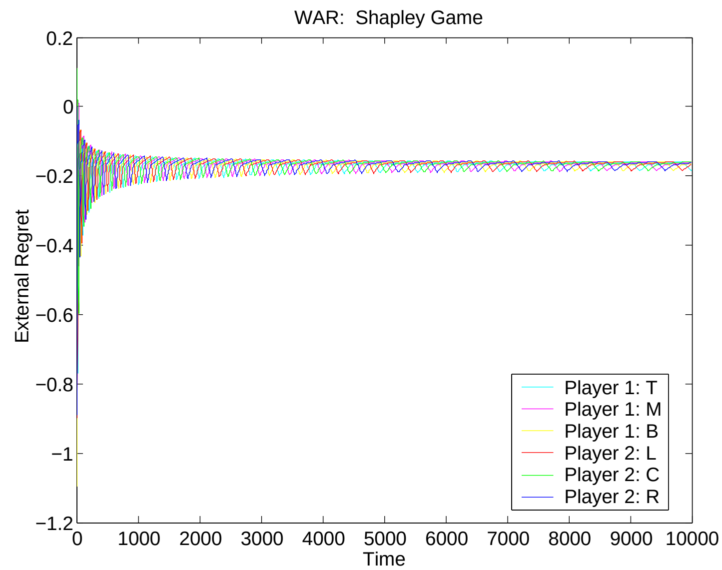


Shapley Game

External Regret: HMC

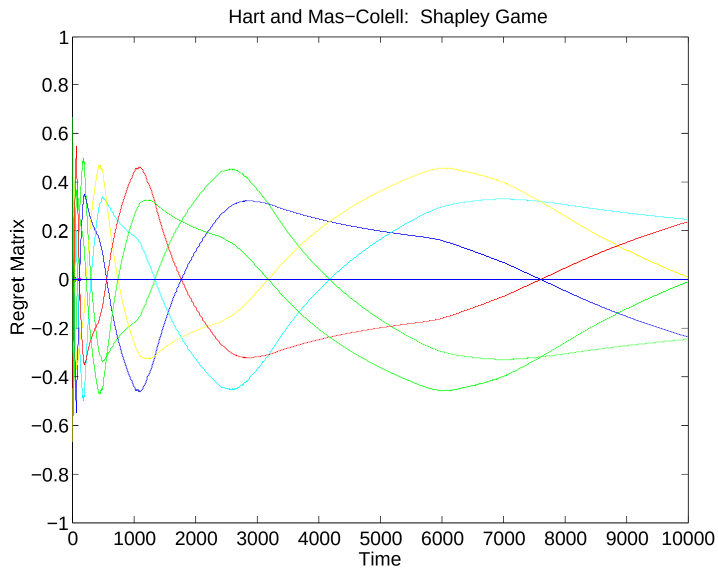


External Regret: WAR

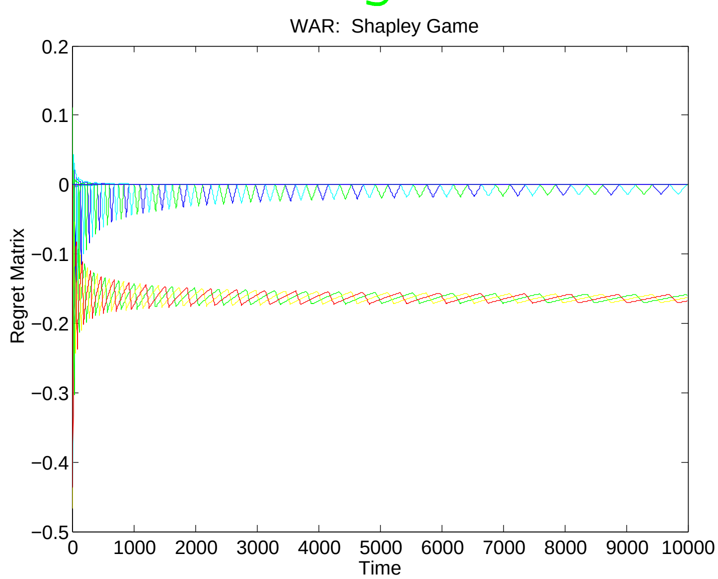


Shapley Game

Internal Regret: HMC

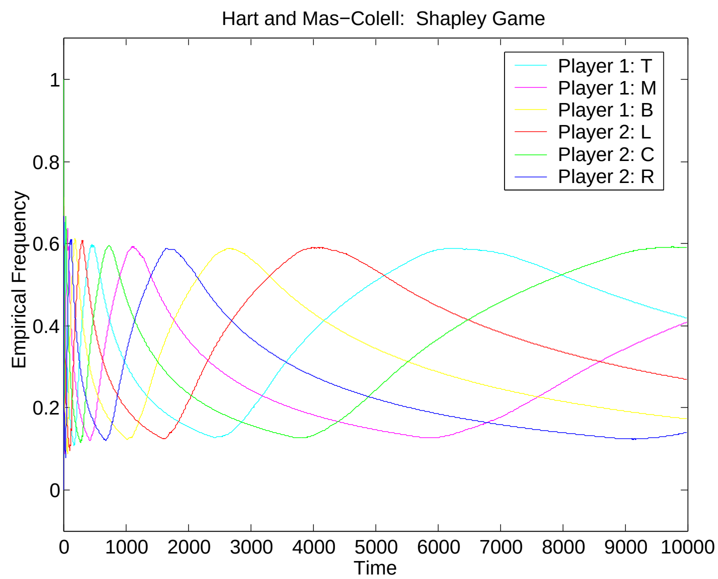


Internal Regret: WAR

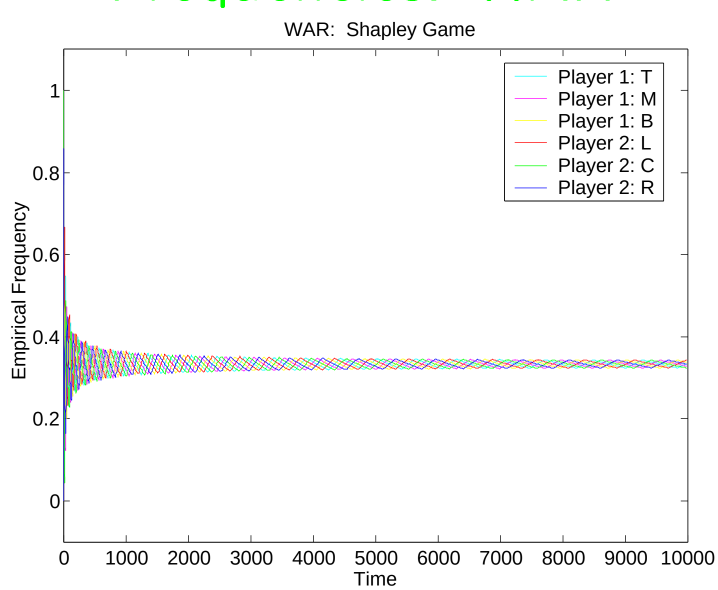


Shapley Game

Frequencies: HMC



Frequencies: WAR



Naive No-Regret Learning

$$\widehat{R}_i(a_i, a_{-i}^t) = \begin{cases} \frac{R_i(a_i, a_{-i}^t)}{\widehat{\pi}_i^t(a_i)} & \text{if } a_i^t = a_i \\ 0 & \text{otherwise} \end{cases}$$

$$\widehat{\pi}_i^t = (1-\epsilon)\pi_i^t + \frac{\epsilon}{|A_i|}$$

Theorem

If an informed learning algorithm $\mathcal{A}_i$ exhibits no-regret, then the naive learning algorithm $\widehat{\mathcal{A}}_i$ exhibits ϵ-no-regret.

No-Regret Q-Learning

NRQ(MGame, γ, α, ϵ)

Inputs	discount factor γ
	rate of averaging α
	rate of exploration ϵ
Output	equilibrium state-value function V^*
	equilibrium action-value function Q^*
Initialize	$V = Q = 0$

REPEAT
    initialize $s, a_1, \ldots, a_n$
    WHILE s is nonterminal DO
        simulate actions $a_1, \ldots, a_n$ in state s
        observe rewards $R_1, \ldots, R_n$ and next state s'

        for all $i \in I$
            let $\hat{\pi}(s', \vec{a}) = \Pi_i \hat{\pi}_i(s', a_i)$
            compute $V_i(s') = \sum_{\vec{a} \in A} \hat{\pi}(s', \vec{a}) Q_i(s', a)$
            update $Q_i(s, \vec{a}) = (1 - \alpha) Q_i(s, \vec{a}) + \alpha P_i(s, a_i)$
                with $P_i(s, a_i) = R_i + \gamma V_i(s')$
            update policies
                informed
                naive

        (simultaneously) choose actions $a'_1, \ldots, a'_n$
        $s = s'$, $a_1 = a'_1, \ldots, a_n = a'_n$
        decay α
FOREVER

Part I

Correlated-Q Learning

- good news
  - converges (empirically) to an equilibrium policy
- bad news
  - equilibrium policy is path dependent

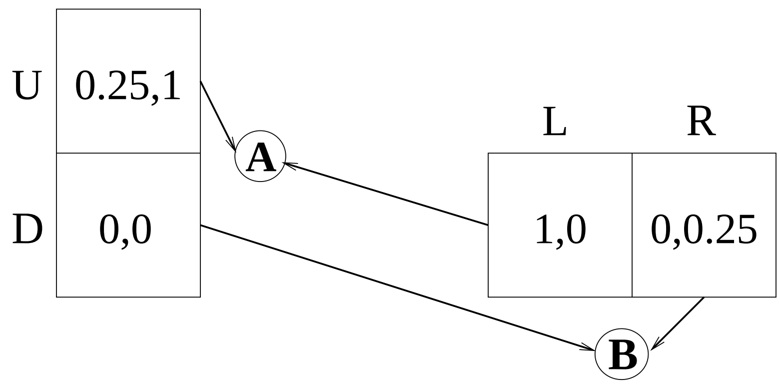
Part II

No-Regret-Q Learning

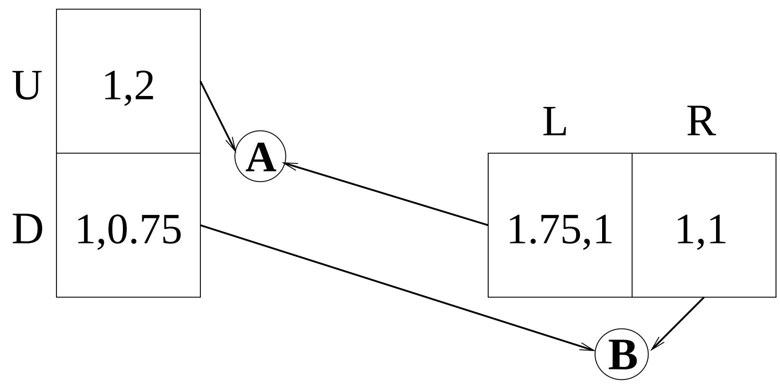
- conjectures
  - WAR and PEACE exhibit no-internal regret
  - NER Q-Learning converges to minimax strategies in constant-sum Markov games
  - NIR Q-Learning converges to correlated equilibrium in general-sum Markov games

Marty's Game

Rewards



Q-Values



Unique Mixed Strategy Equilibrium

$\pi_r(U) = 7/15$ and $\pi_c(L) = 4/9$