Algorithms for learning in simple and complex games

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(Sep 24, 2018)

Algorithms for learning in simple and complex games

Brief Introduction to Game Theory

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(Sep 24, 2018)

Game Theory

Mathematical framework studying strategies of players in situations where the outcomes of their actions critically depend on the actions performed by the other players.










Computational Game Theory

Analytic approach

- Small model size
- Inputs in analytic form
- Analysis of system behavior
- Complete understanding

Computational approach

- Huge model size
- Real world data as inputs
- Computing optimal strategies
- Partial understanding

Matrix (normal form) games

Player 2
Column player
Minimizer

Player 1
Row player
Maximizer

	r	p	s
R	0	-1	1
P	1	0	-1
S	-1	1	0

Zero-sum game, pure strategy, mixed strategy

Best response $$BR_i(\sigma_{-i}) = \arg\max_{a_i \in A_i} U_i(a_i, \sigma_{-i})$$

Nash equilibrium, game value

Non-zero Sum Games

	b	f
B	2, 1	0, 0
F	0, 0	1, 2

What is the Nash equilibrium?

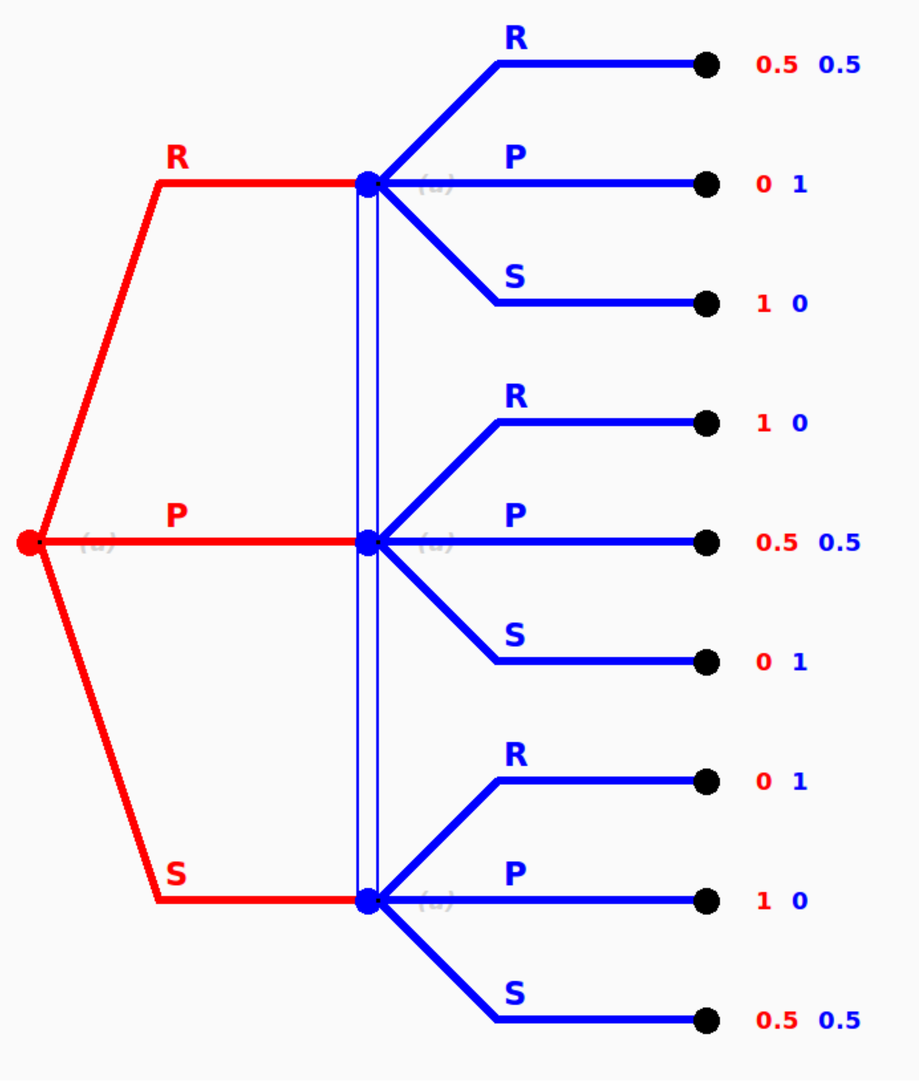
Equilibrium selection problem

Correlated equilibria, coarse correlated

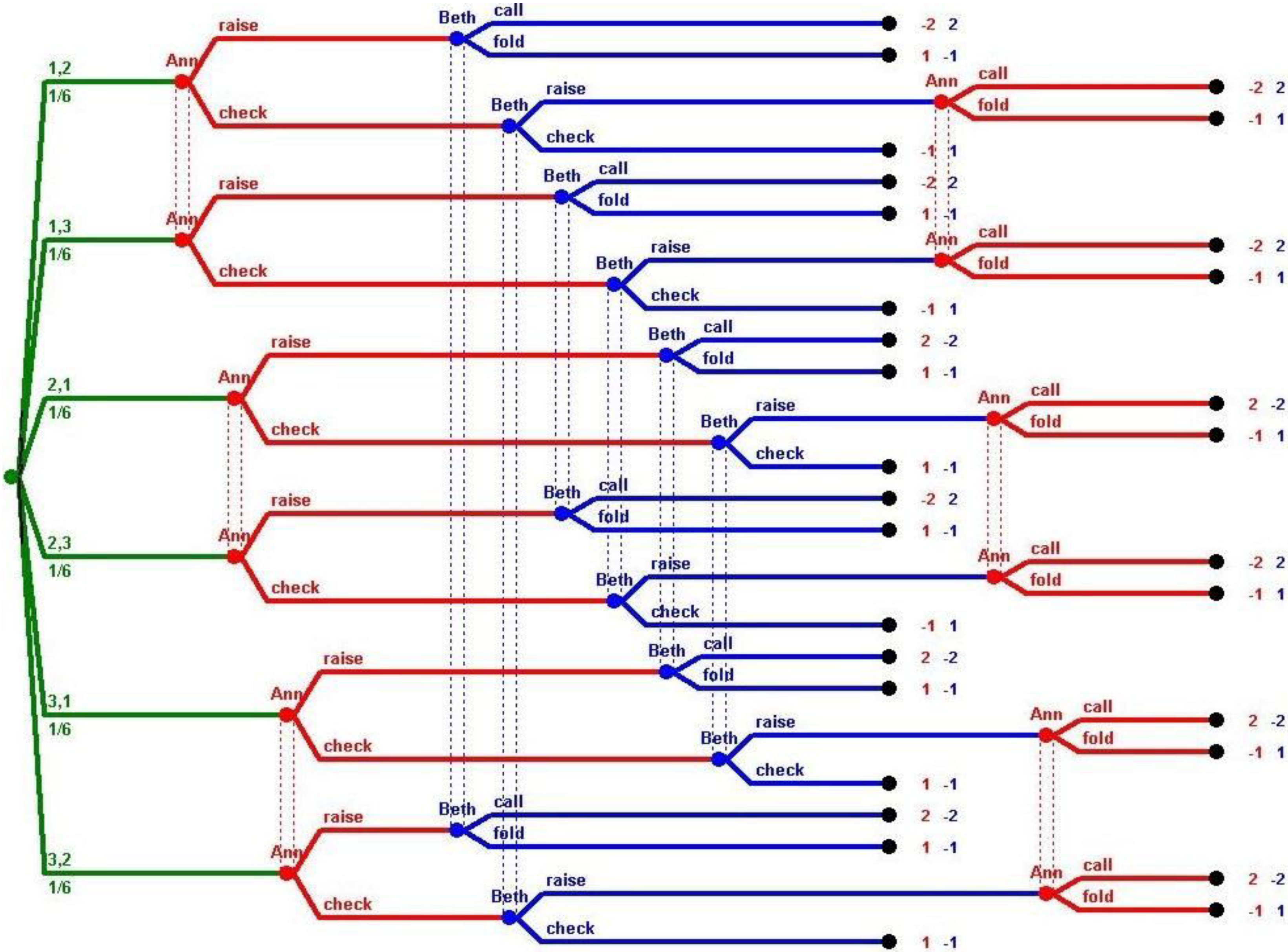
Stackelberg equilibrium

	c	d
C	-1, -1	-7, 0
D	0, -7	-5, -5

Extensive-form game



Extensive Form Games



Plan

Online learning and prediction

- single agent learns to select the best action

Learning in normal form games

- the same algorithms used by multiple agents

Learning in extensive form games

- generalizing these ideas to sequential games

Brief introduction to neural networks

DeepStack

Algorithms for learning in simple and complex games

Introduction to Online Learning and Prediction

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(Sep 24, 2018)

Introduction

Online learning and prediction

- learning from data that become available in sequence
- adapting prediction (behavior) after each data point
- optimizing overall precision (not only after all data arrive)

Applications

- investing in best fond
- web advertisements
- selecting the best (e.g., page replacement) algorithm

Introduction

Why do we care about online learning in games?

- repeated play against an unknown opponent
- (repeated) play of an unknown game
- understanding how equilibria may occur in real world
- computationally efficient equilibrum approximation algorithms

Prediction with expert advice

a_1

a_2

a_3

Problem definition

- Set of n actions (experts) $A = \{a_1, a_2, \ \dots, a_n\}$
- Set of time steps $t = \{1,2, \dots, T\}$
- In each step
  - Decision-maker selects a mixed strategy σ^t
  - An adversary selects rewards $u^t: A \to [0,1]$ (adaptive vs oblivious)
  - Action $\mathrm{a}^{\mathrm{t}} \in A$ is selected based on σ^t
  - The decision-maker receives reward $u^t(a^t)$ (learns the whole u^t)

External Regret

	σ^0	u^0	σ^1	u^1	σ^2	u^2		σ^T	u^T
a_1	0.2	0	0.1	1	0.3	0			
a_2	0.5	0.5	0.4	0.5	0.3	1	…		
a_3	0.3	1	0.5	0	0.4	0			
$\sigma^t \cdot u^t$	$x^0 = 0.55$		$x^1 = 0.3$		$x^2 = 0.3$				x^T

Goal: play as well as the best expert

Immediate regret at time t for not choosing action i

$r^t(i) = u^t(i) - x^t$

Cumulative external regret for playing $\sigma^0, \sigma^1 \dots \sigma^T$

$R^T = max_{i \in A} \sum_{t=0}^{T} r^t(i) = max_{i \in A} \sum_{t=0}^{T} u^t(i) - \sum_{t=0}^{T} x^t$

Average external regret for playing $\sigma^0, \sigma^1 \dots \sigma^T$

$\bar{r}^T = \frac{1}{T} R^T$

Swap Regret

	σ^0	u^0	σ^1	u^1	σ^2	u^2		σ^T	u^T
a_1	0.2	0	0.1	1	0.3	0			
a_2	0.5	0.5	0.4	0.5	0.3	1	…		
a_3	0.3	1	0.5	0	0.4	0			
$\sigma^t \cdot u^t$	$x^0 = 0.55$		$x^1 = 0.3$		$x^2 = 0.3$			x^T	

Goal: minimize regret for not playing a $\delta(a)$ instead of a for some $\delta: A \to A$

Cumulative swap regret for playing $\sigma^0, \sigma^1 \ldots \sigma^T$

$$R^T = max_\delta \sum_{t=0}^T \sum_{i \in A} \sigma^t(i)(u^t(\delta(i)) - u^t(i))$$

Internal regret

allows switching only all occurrences of a_i by a_j

External ⊂ Swap, Internal ⊂ Swap

No-regret algorithms

An algorithm has **no regret** if for any $u^0, u^1 \dots u^T$ produces $\sigma^0, \sigma^1 \dots \sigma^T$ such that $\bar{r}^T \rightarrow 0$ as $T \rightarrow \infty$.

Why not simply to maximize reward?

$$maximize \sum_{t=0}^{T} x^t$$

The adversary may choose $\forall i \in A$, $u^t(i) = 0$ and we have minimal reward regardless of the used algorithm.

Any algorithm has (optimal) 0 regret.

Regret towards best strategy in hindsight

$$R_{best}{}^{T} = \sum_{t=0}^{T} max_{i \in A}\, u^t(i) - \sum_{t=0}^{T} x^t$$

Proposition: There is no algorithm with no regret towards the best sequence of choices.

Proof: Let $A = \{U, D\}$. For an arbitrary sequence of strategies σ^t, choose a reward vector $u^t = (0{,}1)$ if $\sigma^t(U) \geq \frac{1}{2}$ and $u^t = (1{,}0)$ otherwise.

The cumulative reward of the algorithm $\sum_{t=0}^{T} x^t \leq \frac{T}{2}$, while the best strategy in hindsight has reward $\sum_{t=0}^{T} max_{i \in A}\, u^t(i) = T$. Therefore

$$R_{best}{}^{T} \geq \frac{T}{2} \text{ and } \bar{r}_{best}^{T} \to z \geq \frac{1}{2}$$

Regret of deterministic algorithms

Proposition: There is no deterministic no-external-regret algorithm.

Proof: We assume that the adversary selects rewards u^t knowing strategy σ^t. (For example, it can simulate the deterministic algorithm from the beginning.) Therefore, with $n = 2$, he can always give reward 0 for the selected action and 1 for the other action. One of the actions got reward 1 at least $T/2$ times, therefore $\bar{r}^t \geq \frac{1}{2}$.

Lower bound on external regret

Theorem: No (randomized) algorithm over n actions has expected external regret vanishing faster than $\Theta(\sqrt{\ln(n)/T})$.

Proof sketch: Assume n=2. Consider an adversary that, independently on each step t, chooses uniformly at random between the cost vectors (1, 0) and (0, 1) regardless of the decision-making algorithm. The cumulative expected reward is exactly $T/2$. In hindsight, however, with constant probability one of the two fixed actions has cumulative reward $\mathrm{T}/2 + \Theta(\sqrt{T})$. The reason is that T fair coin flips have standard deviation $\Theta(\sqrt{T})$.

Lower bound on external regret

Theorem: There exist no-regret algorithms with expected external regret $O(\sqrt{\ln(n)\,/T})$.

Proof: We will show Randomized Weighted Majority algorithm.

Corollary: There exists a decision-making algorithm that, for every $\epsilon > 0$, has expected regret less than ϵ after $O(\ln(n)/\epsilon^2)$ iterations.

Randomized Weighted Majority

Aka Hedge or multiplicative weights (MW) algorithm. It is easier to analyze in costs $c(i) = (1 - u(i))$. The algorithm maintains weights $w(i)$ for each action $i \in A$.

Initialize $w^1(i) = 1$ for every $i \in A$

For each time $t = 1,2, \dots, T$

- Let $W^t = \sum_{i \in A} w^t(i)$ and play $\sigma^t(i) = w^t(i)/W^t$
- Given costs c^t, set $w^{t+1}(i) = w^t(i)(1-\gamma)^{c^t(i)}$ for each $i \in A$
- (Equivalently $w^{t+1}(i) = w^t(i)e^{-\eta c^t(i)}$ for $\eta = -\ln(1-\gamma)$)

Hedge Regret Bound

Theorem: Expected external regret of Hedge is $\bar{r}^T < 2\sqrt{ln(n)/T}$

Proof: W.L.O.G. we assume oblivious adversary.

Let $OPT = \min_{i \in A} \sum_{t=1}^{T} c^t(i)$ be the cost for optimal action i^* and

$\nu^t = \sum_{i \in A} \sigma^t(i) c^t(i) = \sum_{i \in A} \frac{w^t(i)}{W^t} c^t(i)$ be the algorithms cost at t.

$$W^T \geq w^T(i^*) = w^1(i^*) \prod_{t=1}^{T} (1-\gamma)^{c^t(i^*)} = (1-\gamma)^{OPT}$$

$$W^{t+1} = \sum_{i \in A} w^{t+1}(i) = \sum_{i \in A} w^t(i)(1-\gamma)^{c^t(i)}$$

$$\leq \sum_{i \in A} w^t(i)\left(1 - \gamma c^t(i)\right) = W^t(1-\gamma \nu^t)$$

$$(1-\gamma)^{OPT} \leq W^T \leq W^1 \prod_{t=1}^{T} (1-\gamma\nu^t)$$

$$OPT \ln(1-\gamma) \leq \ln n + \sum_{t=1}^{T} \ln(1-\gamma\nu^t)$$

$$\ldots \sum_t^T \nu^t \leq OPT + \gamma T + \frac{\ln n}{\gamma} \quad \Rightarrow \frac{1}{\mathrm{T}} \sum_t^T \nu^t \leq \frac{OPT}{T} + 2\sqrt{\frac{\ln n}{T}}$$

Regret Matching

The algorithm maintains cummulative regrets $\mathrm{R}(i)$ for each action $i \in A$.

Initialize $R^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Let $S^t = \sum_{i \in A} \max(0, R^t(i))$ and play $\sigma^t(i) = \max(0, R^t(i))/S^t$

Given rewards u^t, for each $i \in A$ set

$$R^{t+1}(i) = R^t(i) + r^t(i) = R^t(i) + (u^t(i) - \sum_{j \in A} \sigma^t(j) u^t(j))$$

Regret Matching+

The algorithm maintains cumulative regrets-like values $\mathrm{Q}(i)$ for each action $i \in A$.

Initialize $Q^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Play $\sigma^t(i) = Q^t(i) / \sum_{j \in A} Q^t(j)$

Given rewards u^t, for each $i \in A$ set

$$Q^{t+1}(i) = \max(0, Q^t(i) + r^t(i)) = \max(0,\ u^t(i) - \sum_{j \in A} \sigma^t(j) u^t(j))$$

RM+ Regret Bound

Lemma: Regret-like values $Q^t(i)$ are an upper bound on $R^t(i)$.

Proof: $Q^{t+1}(i) - Q^t(i) = \max\left(0, Q^t(i) + r^t(i)\right) - Q^t(i)$
$$\geq Q^t(i) + r^t(i) - Q^t(i) = r^t(i)$$

Lemma: For any i and value functions $Q^T(i) \leq \sqrt{nT}$.

Proof: $\left(\max_{i \in A} Q^T(i)\right)^2 = \max_{i \in A} Q^T(i)^2 \leq \sum_{i \in A} Q^T(i)^2 =$

$$= \sum_{i \in A} \max(0, Q^{T-1}(i) + u^T(i) - \sum_{j \in A} \sigma^T(j) u^T(j))^2$$

$$\ldots \leq \sum_i Q^{T-1}(i)^2 + n$$

By induction $Q^T(i)^2 \leq nT$.

Summary

General setting of prediction with expert advice

Regret as a measure of distance from the optimal strategy

There are no-regret algorithms

- Hedge, Regret matching, Regret matching+

Plan

Online learning and prediction

- single agent learns to select the best action

Learning in normal form games

- the same algorithms used by multiple agents

Learning in extensive form games

- generalizing these ideas to sequential games

Brief introduction to neural networks

DeepStack

Algorithms for learning in simple and complex games

Learning in Normal Form Games

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(Sep 24, 2018)

Introduction

How may simple learning agents achieve equilibrium outcomes?

Best Response Dynamics (Fictitious play)

- best response to average empirical play
- needs to know the game

No-Regret Dynamics

- each player uses no-regret algorithm
- may now only their own actions and received rewards

Best response dynamics

Fictitious play

Players maintain empirical distribution of past opponent's actions

$$\bar{\sigma}_{-i}^{T} = \frac{1}{\mathrm{T}} \sum_{t=1}^{T} \sigma_{-i}^{t} \qquad \text{(often in form of frequencies } \eta_i^T \text{)}$$

In each round, each player plays BR to these distributions

$$\sigma_i^t = \arg\max_{a_i \in A_i} U_i(a_i, \bar{\sigma}_{-i}^{t})$$

Player 1 \ Player 2	heads	tails
heads	$(1, -1)$	$(-1, 1)$
tails	$(-1, 1)$	$(1, -1)$

Time	η_1^t	η_2^t	Play
0	$(0, 0)$	$(0, 2)$	(H, H)

Result of FP in case of convergence

Theorem: If the empirical action frequencies of fictitious play converge ($\bar{\sigma}^t \rightarrow \sigma^*$) they converge to the Nash equilibrium of the game.

Theorem: The empirical frequencies of FP converge to NE in

- constant-sum games
- two player games where each player has up to two actions
- games solvable by iterated strict dominance
- identical interest games
- potential games

Why it may not converge?

Shapley's example in a modified rock-paper-scissors:

	R	S	P
R	0, 0	1, 0	0, 1
S	0, 1	0, 0	1, 0
P	1, 0	0, 1	0, 0

Unique NE is the uniform strategy for both players.

Let $\eta_1^0 = (1,0,0)$ and $\eta_2^0 = (0,1,0)$.

Play may be (P,R),(P,R)… for k steps until column switches to S.

Then (P,S) follows until row switches to R (for βk steps, $\beta > 1$).

Then (R,S) follows until column switches to P (for $\beta^2 k$ steps).

The play cycles among all 6 non-diagonal profiles with periods of ever-increasing length, hence, the empirical frequencies cannot converge.

Convergence of FP

Theorem (Brandt, Fischer, Harrenstein, 2010): In symmetric two-player constant-sum games, FP may require exponentially many rounds (in the size of the representation of the game) before an equilibrium action is eventually played. This holds even for games solvable via iterated strict dominance.

Proof:

	a	b	c
a	0	-1	-ϵ
b	1	0	-ϵ
c	ϵ	ϵ	0

With $\epsilon = 2^{-k}$, FP may take 2^k rounds to play the equilibrium action c for the first time.

(a,a),(b,b),…,(b,b)

$2^k - 1$ times

No-Regret Learning Summary

Immediate regret at time t for not choosing action i

$$r^t(i) = u^t(i) - \sigma^t \cdot u^t$$

Cumulative external regret for playing $\sigma^0, \sigma^1 \ldots \sigma^T$

$$R^T = max_{i \in A} \sum_{t=0}^{T} r^t(i) = max_{i \in A} \sum_{t=0}^{T} u^t(i) - \sum_{t=0}^{T} \sigma^t \cdot u^t$$

Average external regret for playing $\sigma^0, \sigma^1 \ldots \sigma^T$

$$\bar{r}^T = \frac{1}{T} R^T$$

An algorithm has **no regret** if for any $u^0, u^1 \ldots u^T$ produces $\sigma^0, \sigma^1 \ldots \sigma^T$ such that $\bar{r}^T \to 0$ as $T \to \infty$.

From External to Swap Regret

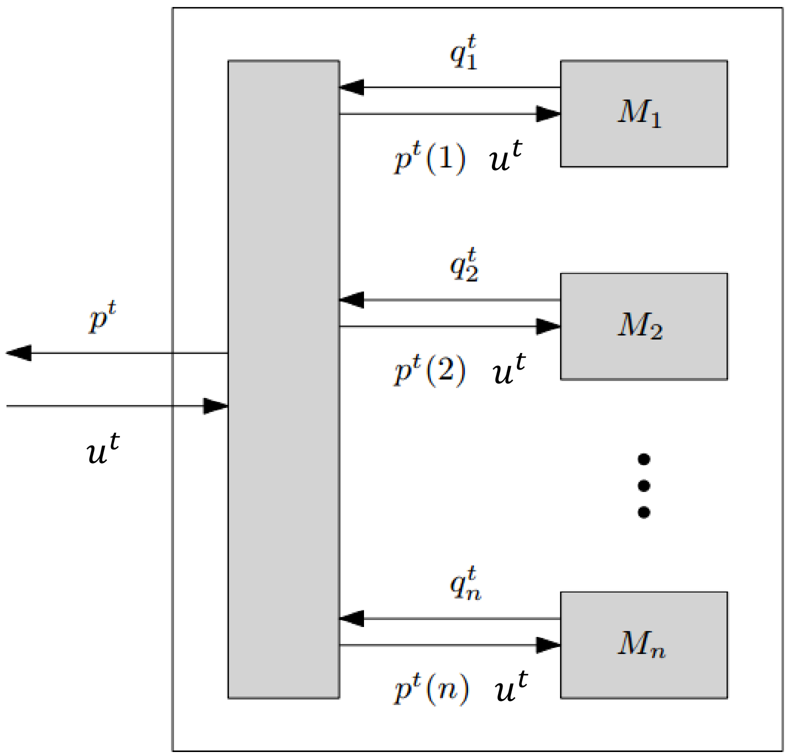
Cumulative swap regret for playing $\sigma^0, \sigma^1 \dots \sigma^T$

$$R^T = max_{\delta:A \to A} \sum_{t=0}^{T} \sum_{i \in A} \sigma^t(i)(u^t(\delta(i)) - u^t(i))$$

From External to Swap Regret

Theorem (Blum & Mansour 2007):If there is a no-external-regret algorithm for a setting, there is also a no-swap-regret algorithm.

Proof: Polynomial black-box reduction.



From External to Swap Regret

Proof: Average expected reward of the overall algorithm

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n} p^t(i)\, u^t(i)$$

No-regret algorithm M_j choses $q_j^1, \dots, q_j^T$, gets $p^1(j)u^1, \dots, p^T(j)u^T$. Thus

$$\forall k \in A\colon \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n} q_j^t(i)\, (p^t(j)\, u^t(i)) \geq \frac{1}{T}\sum_{t=1}^{T} p^t(j)u^t(k) - \bar{r}_{\mathrm{j}}$$

Fix an arbitrary $\delta\colon A \to A$ and sum over all $j \in A$:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n} q_j^t(i)(p^t(j)\, u^t(i)) \geq \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{n} p^t(j)u^t\big(\delta(j)\big) - \sum_{j=1}^{n} \bar{r}_j$$

From External to Swap Regret

We are done if we ensure

$$p^t(i) = \sum_{j=1}^{n} q_j^t(i) p^t(j)$$

This is true if p^t is the eigenvector of matrix given by q_j^t for $\lambda = 1$.

Equivalently, p^t are the stationary distribution of Markov chain.

Such vector p^t always exists and can be easily found!

From External to Swap Regret

Corollary: Let $\overline{r_M}(t) \rightarrow 0$ be the external regret convergence bound for a base algorithm used in the black-box reduction with $|A|$ actions. Than the swap regret of the overall algorithm is

$$\overline{r_{sw}}(T) \leq |A|\overline{r_M}(T).$$

Corollary: The black-box reduction with Hedge for all actions produces an algorithm with $\overline{r_{sw}}(T) \leq 2|A|\sqrt{\ln|A|/T}$.

No-Regret Dynamics – full information

Definition:

1) Each player i choses independently a mixed strategy σ_i^t using a no-regret algorithm.

2) Each player receives for all $a_i \in A_i$ rewards

$$u_i^t(a_i) = \mathbf{E}_{a_{-i}\sim\sigma_{-i}}[U(a_i, a_{-i})]$$

No-Regret Dynamics – full information

Theorem: If after T iterations of no-regret dynamics each player has external regret lower then ϵ than $\sigma = \frac{1}{T}\sum_t^T \sigma^t$, where $\sigma^t = \prod_{i=1}^k \sigma_i^t$, is an ϵ-coarse correlated equilibrium of the game. I.e., for any $a_i' \in A_i$

$$\mathbf{E}_{a\sim\sigma}[U_i(a)] \geq \mathbf{E}_{a\sim\sigma}[U_i(a_i', a_{-i})] - \epsilon$$

Corollary: If we run Hedge in a game with less than $|A|$ actions for each player for T iterations, the resulting average strategy is an ($\sqrt{ln(|A|)/T}$)-coarse correlated equilibrium of the game.

Corollary: If we run regret matching+ in a game with less than $|A|$ actions for each player for T iterations, the resulting average strategy is an ($\sqrt{|A|/T}$)-coarse correlated equilibrium of the game.

Minimax Theorem

Note: In zero-sum games, coarse correlated equilibria are Nash.

Theorem (Minimax Theorem): For any matrix game G

$$\max_x \min_y x^T G y = \min_y \max_x x^T G y$$

Proof: For contradiction assume that for some $\alpha > 0$

$$\max_x \min_y x^T G y < \min_y \max_x x^T G y - \alpha\,.$$

Set $\epsilon = \frac{\alpha}{2}$ and let both players run Hedge for time $\tau = 2 \ln n / \epsilon^2$.
Let $\hat{x}, \hat{y}$ be the empirical frequencies of their play and v the average reward of the maximizer.

$$\max_x \min_y x^T G y \geq \min_y \hat{x}^T G y \geq v - \epsilon \geq \max_x x^T G \hat{y} - 2\epsilon \geq \min_y \max_x x^T G y - \alpha$$

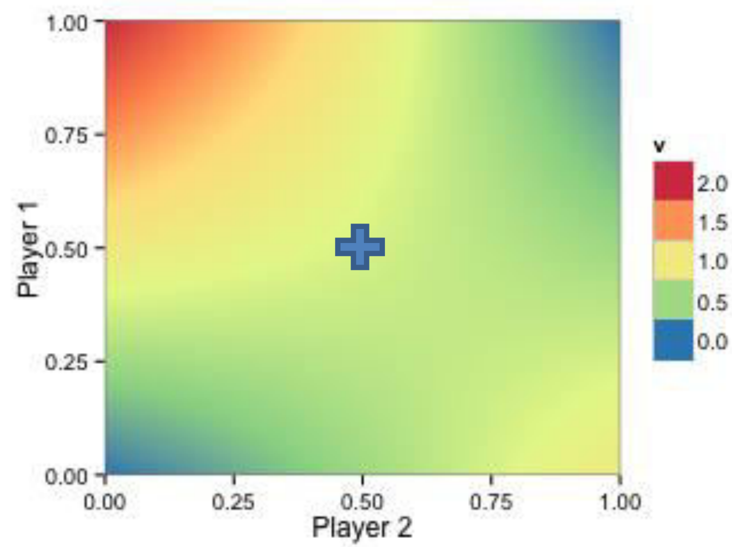
No-Regret Dynamics

Theorem: If after T iterations of no-regret dynamics each player has swap regret lower then ϵ than $\sigma = \frac{1}{T}\sum_t^T \sigma^t$, where $\sigma^t = \prod_{i=1}^k \sigma_i^t$, is an ϵ-correlated equilibrium of the game. I.e., for any player i and switching function $\delta: A \to A$

$$\mathbf{E}_{a\sim\sigma}[U_i(a)] \geq \mathbf{E}_{a\sim\sigma}[U_i(\delta(a_i), a_{-i})] - \epsilon$$

Regret matching+

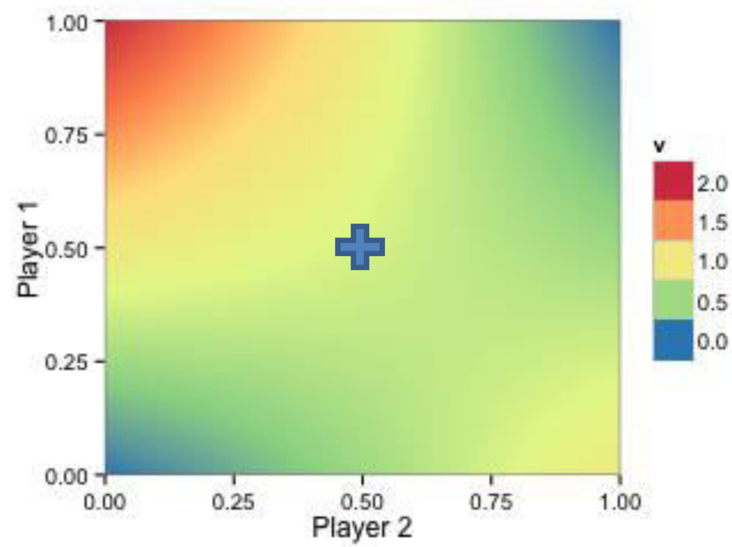
σ^t	0.5	0.5
0.5	2	0
0.5	0	1



Regret matching+

Iteration:
1

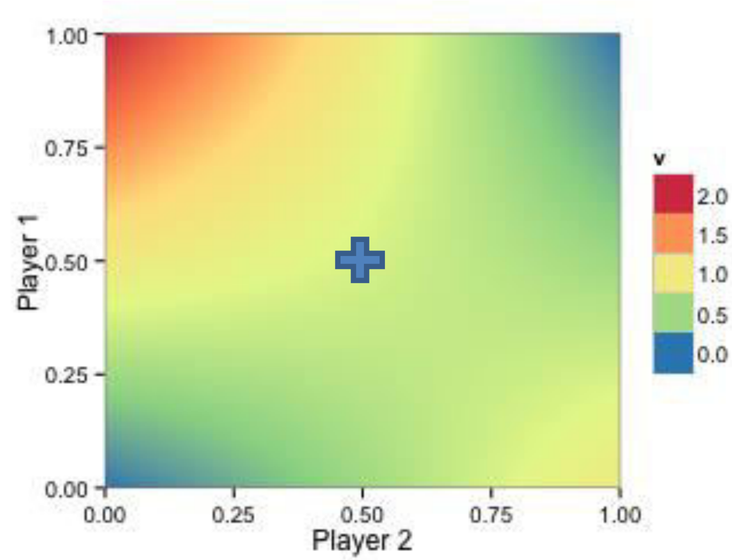
			$\overline{\sigma_2}$		
			R_2	0	0
			r_2		
$\overline{\sigma_1}$	R_1	r_1	σ^t	0.5	0.5
	0	0.25	0.5	2	0
	0	-0.25	0.5	0	1



Regret matching+

Iteration:
1

			$\overline{\sigma_2}$		
			R_2	0	0
			r_2		
$\overline{\sigma_1}$	R_1	r_1	σ^t	0.5	0.5
	0.25	0.25	0.5	2	0
	0	-0.25	0.5	0	1

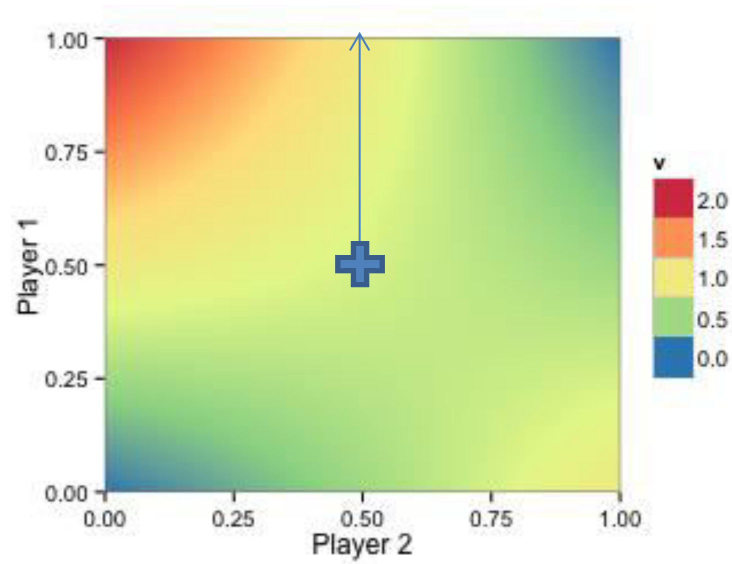


Regret matching+

Iteration:
1

$\overline{\sigma_2}$		
R_2	0	0
r_2		
σ^t	0.5	0.5

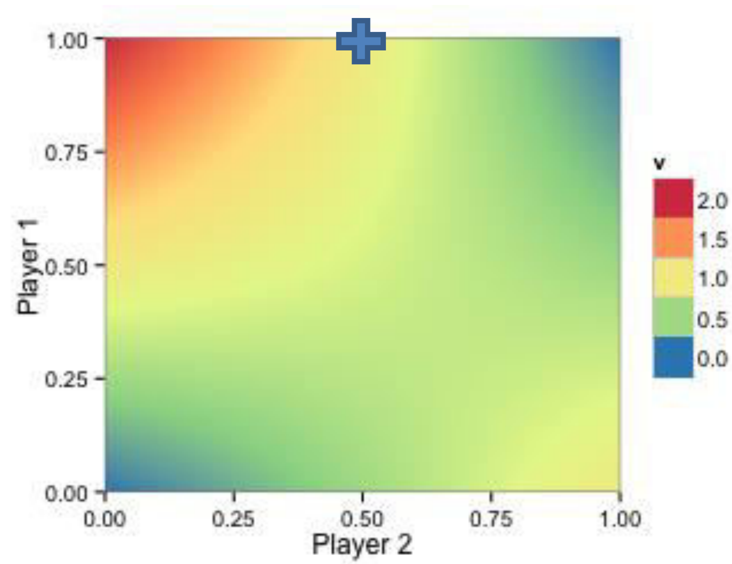
$\overline{\sigma_1}$	R_1	r_1		
	0.25	0.25	2	0
	0	-0.25	0	1



Regret matching+

Iteration:
1

			$\overline{\sigma_2}$		
			R_2	0	0
			r_2	-1	1
$\overline{\sigma_1}$	R_1	r_1	σ^t	0.5	0.5
1	0.25		1	2	0
0	0		0	0	1

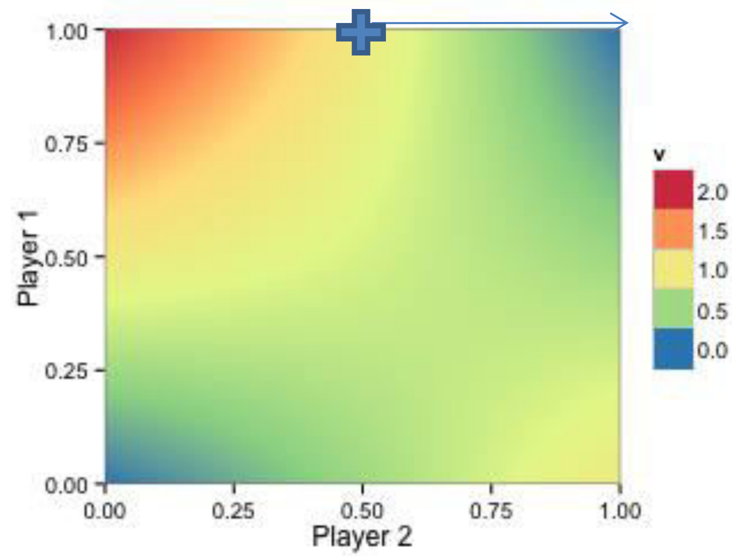


Regret matching+

Iteration:
1

$\overline{\sigma_2}$		
R_2	0	1
r_2	-1	1

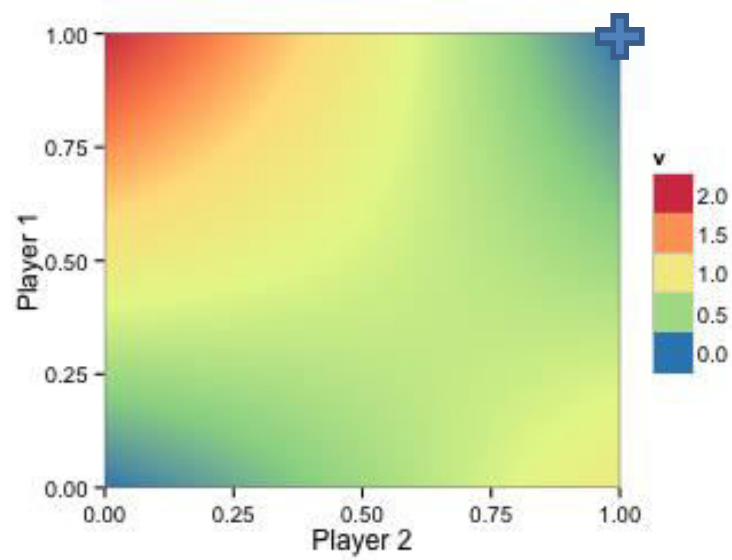
$\overline{\sigma_1}$	R_1	r_1	σ^t		
1	0.25		1	2	0
0	0		0	0	1



Regret matching+

Iteration:
2

			$\overline{\sigma_2}$	0	1
			R_2	0	1
			r_2		
$\overline{\sigma_1}$	R_1	r_1	σ^t	0	1
1	0.25	0	1	2	0
0	0	1	0	0	1



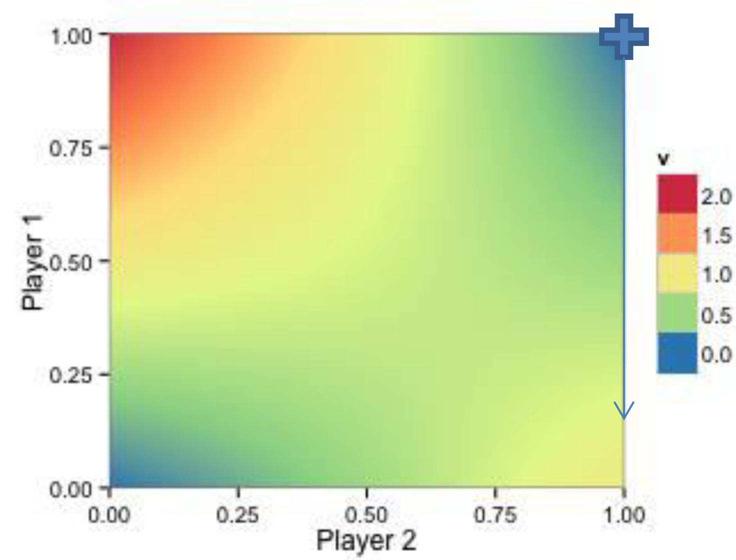
Regret matching+

Iteration:
2

$\overline{\sigma_2}$	0	1
R_2	0	1
r_2		
σ^t	0	1

$\overline{\sigma_1}$	R_1	r_1
1	0.25	0
0	1	1

2	0
0	1

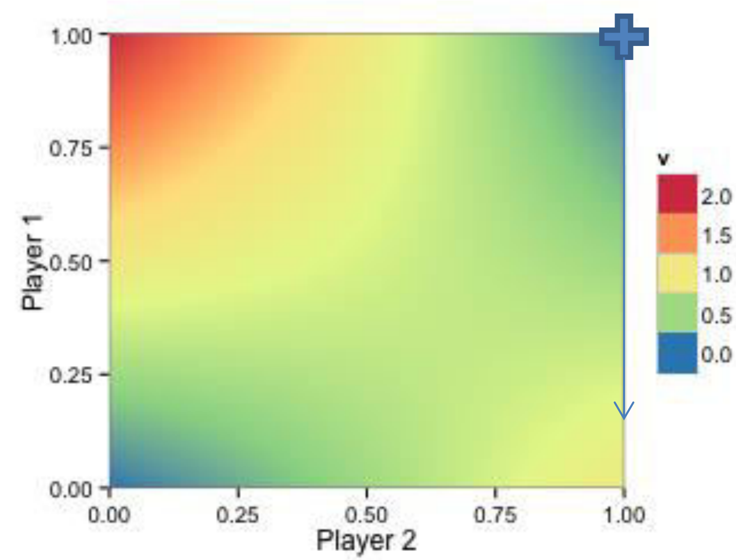


Regret matching+

Iteration:
2

$\overline{\sigma_2}$	0	1
R_2	0	1
r_2		
σ^t	0	1

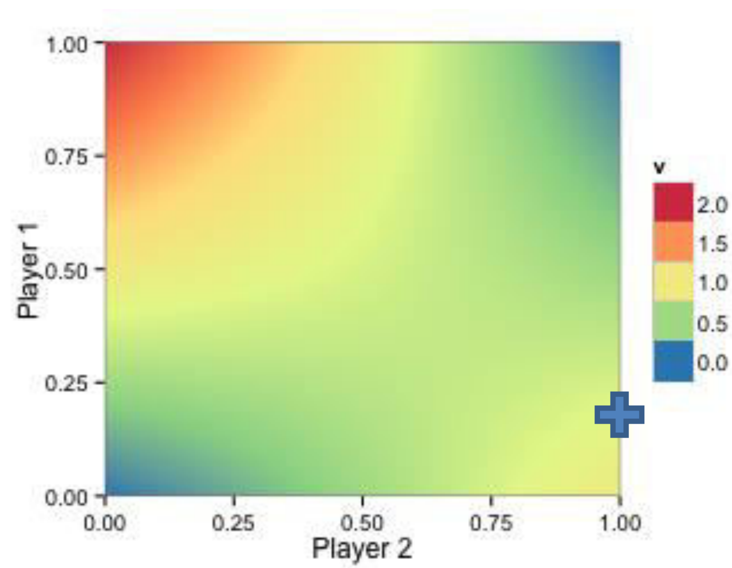
$\overline{\sigma_1}$	R_1	r_1	σ^t		
0.46	0.25	0	0.2	2	0
0.54	1	1	0.8	0	1



Regret matching+

Iteration:
2

			$\overline{\sigma_2}$	0	1
			R_2	0	1
			r_2	0.4	0
$\overline{\sigma_1}$	R_1	r_1	σ^t	0	1
0.46	0.25		0.2	2	0
0.54	1		0.8	0	1

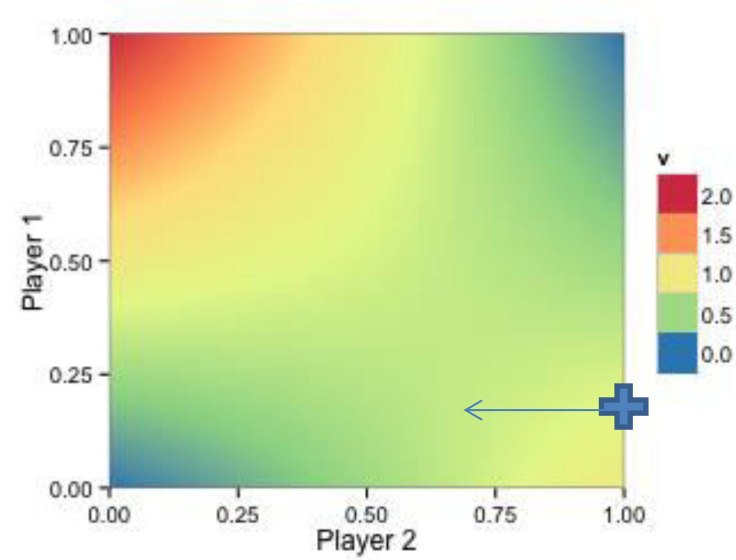


Regret matching+

Iteration:
2

$\overline{\sigma_2}$	0	1
R_2	0.4	1
r_2	0.4	0

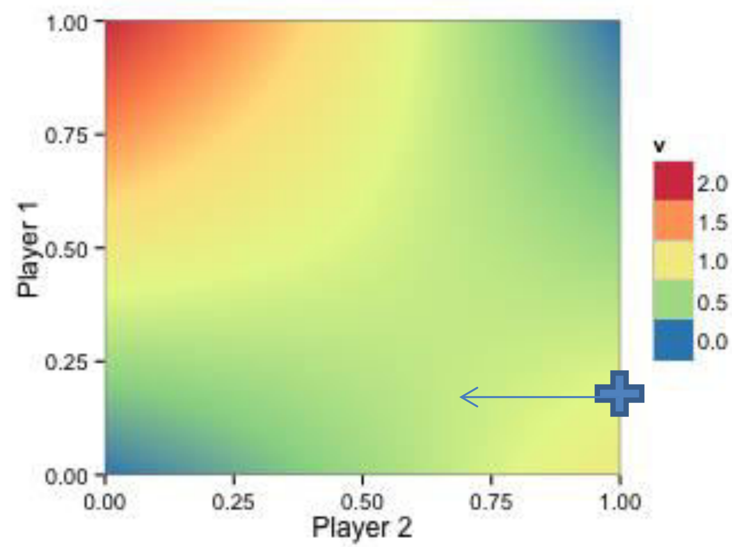
$\overline{\sigma_1}$	R_1	r_1	σ^t		
0.46	0.25		0.2	2	0
0.54	1		0.8	0	1



Regret matching+

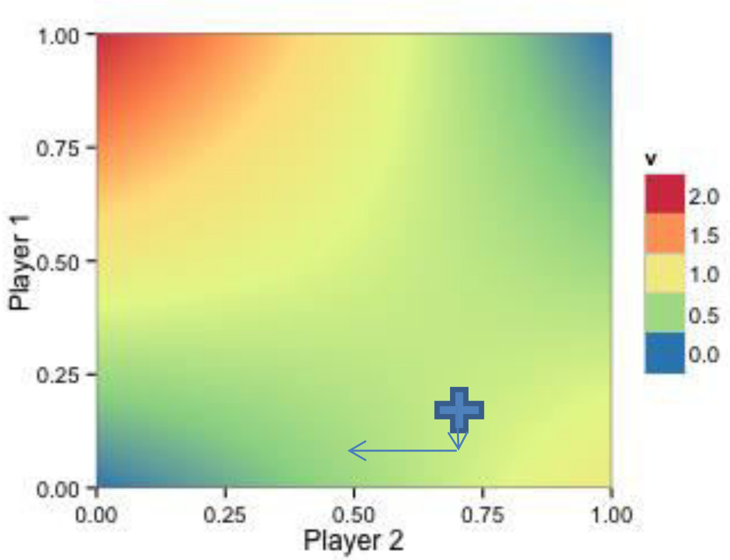
Iteration:
2

			$\overline{\sigma_2}$	0.19	0.81
			R_2	0.4	1
			r_2	0.4	0
$\overline{\sigma_1}$	R_1	r_1	σ^t	0.29	0.71
0.46	0.25		0.2	2	0
0.54	1		0.8	0	1



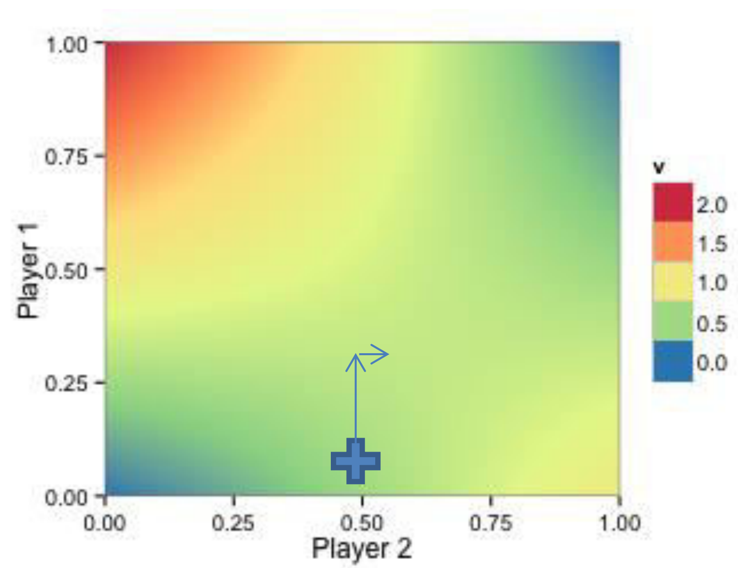
Regret matching+

Iteration:
3



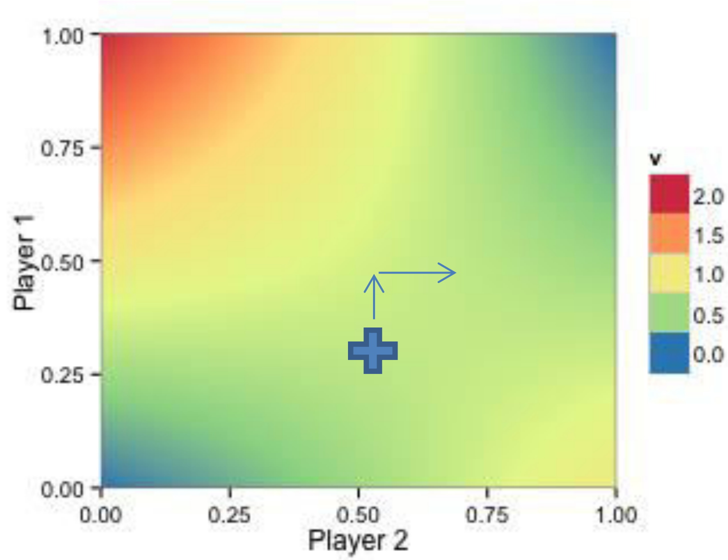
Regret matching+

Iteration:
4



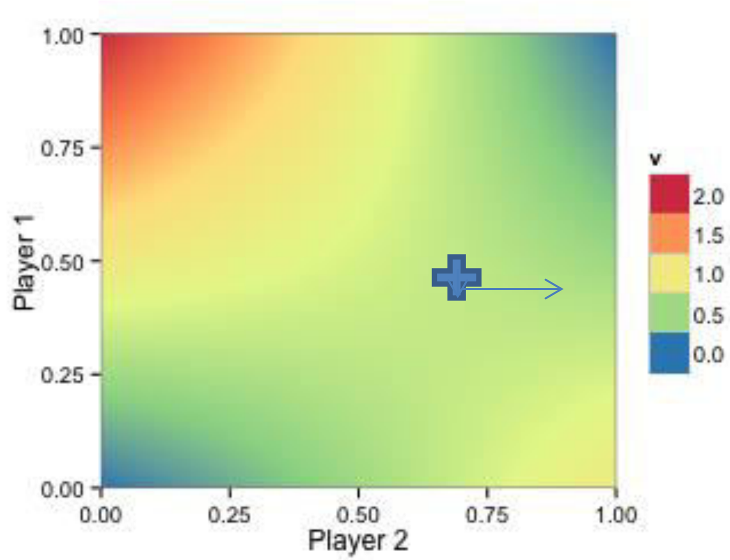
Regret matching+

Iteration:
5



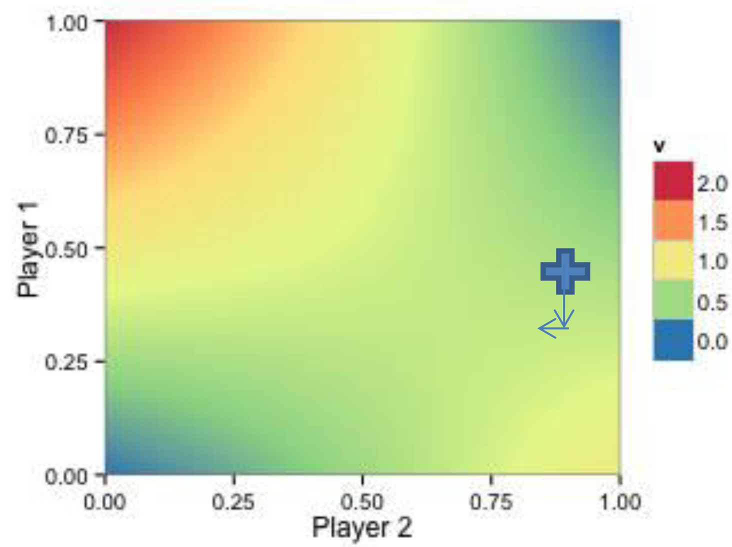
Regret matching+

Iteration:
6



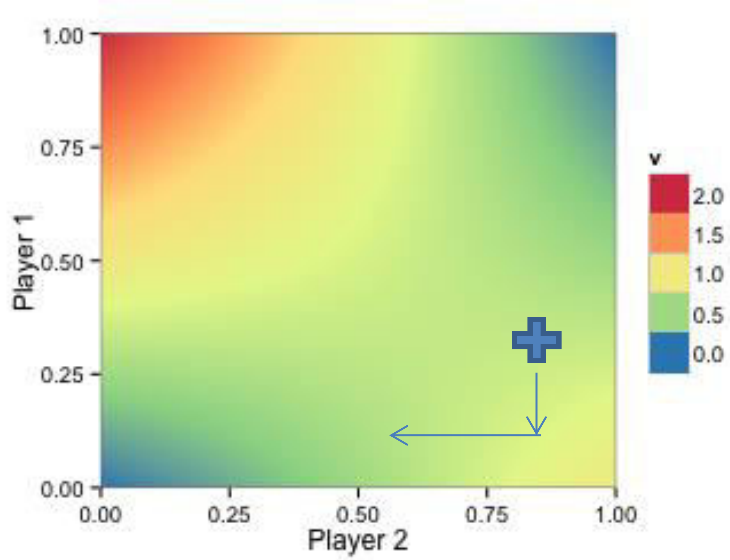
Regret matching+

Iteration:
7



Regret matching+

Iteration:
8

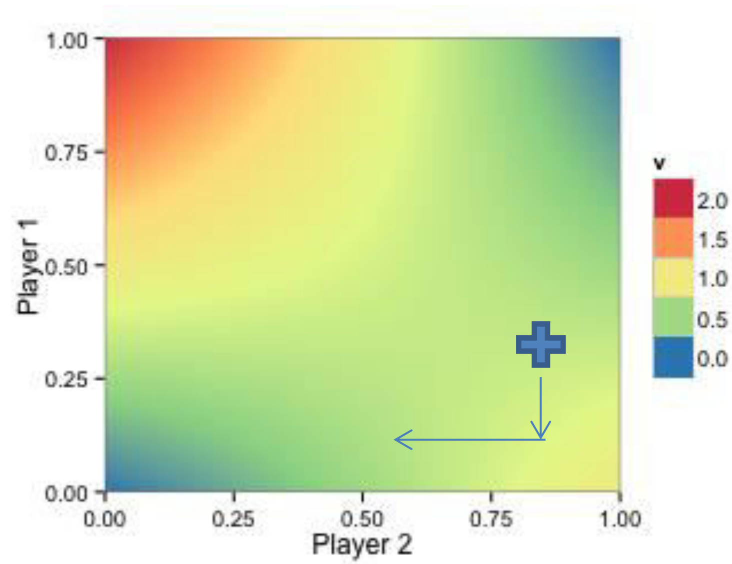


Regret matching+

Iteration:
8

		$\overline{\sigma_2}$	0.30	0.70
		R_2	0.83	1.15
		r_2		
		σ^t	0.42	0.58

$\overline{\sigma_1}$	R_1	r_1			
0.33	0.17		0.11	2	0
0.67	1.30		0.88	0	1



Plan

Online learning and prediction

- single agent learns to select the best action

Learning in normal form games

- the same algorithms used by multiple agents

Learning in extensive form games

- generalizing these ideas to sequential games

Brief introduction to neural networks

DeepStack

Algorithms for learning in simple and complex games

Refresh

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(Sep 25, 2018)

Prediction with expert advice

a_1

a_2

a_3

Problem definition

- Set of n actions (experts) $A = \{a_1, a_2, \ \dots, a_n\}$
- Set of time steps $t = \{1,2, \dots, T\}$
- In each step
  - Decision-maker selects a mixed strategy σ^t
  - An adversary selects rewards $u^t: A \rightarrow [0,1]$ (adaptive vs oblivious)
  - Action $\mathrm{a}^{\mathrm{t}} \in A$ is selected based on σ^t
  - The decision-maker receives reward $u^t(a^t)$ (learns the whole u^t)

Regret Matching+

The algorithm maintains cumulative regrets-like values $\mathrm{Q}(i)$ for each action $i \in A$.

Initialize $Q^1(i) = 0$ for every $i \in A$

For each time $t = 1,2, \dots, T$

Play $\sigma^t(i) = Q^t(i) / \sum_{j \in A} Q^t(j)$

Given rewards u^t, for each $i \in A$ set

$$Q^{t+1}(i) = \max(0, Q^t(i) + r^t(i)) = \max(0,\ u^t(i) - \sum_{j \in A} \sigma^t(j) u^t(j))$$

RM+ Regret Bound

Lemma: Regret-like values $Q^t(i)$ are an upper bound on $R^t(i)$.

Proof: $Q^{t+1}(i) - Q^t(i) = \max\big(0, Q^t(i) + r^t(i)\big) - Q^t(i)$
$\geq Q^t(i) + r^t(i) - Q^t(i) = r^t(i)$

Lemma: For any i and value functions $Q^T(i) \leq \sqrt{nT}$.

Proof: $\left(\max_{i \in \mathrm{A}} Q^T(i)\right)^2 = \max_{i \in \mathrm{A}} Q^T(i)^2 \leq \sum_{i \in A} Q^T(i)^2 =$

$= \sum_{i \in A} \max(0, Q^{T-1}(i) + u^T(i) - \sum_{j \in A} \sigma^T(j) u^T(j))^2$

$\ldots \leq \sum_i Q^{T-1}(i)^2 + n$

By induction $Q^T(i)^2 \leq nT$.

No-Regret Dynamics – full information

Theorem: If after T iterations of no-regret dynamics each player has external regret lower then ϵ than $\boldsymbol{\sigma} = \frac{\mathbf{1}}{\boldsymbol{T}}\sum_{\boldsymbol{t}}^{\boldsymbol{T}} \boldsymbol{\sigma}^{\boldsymbol{t}}$, where $\sigma^t = \prod_{i=1}^{k} \sigma_i^t$, is an ϵ-coarse correlated equilibrium of the game (ϵ-Nash equilibrium in zero-sum). I.e., for any $a_i' \in A_i$

$$\mathbf{E}_{a\sim\sigma}[U_i(a)] \geq \mathbf{E}_{a\sim\sigma}[U_i(a_i', a_{-i})] - \epsilon$$

Corollary: If we run regret matching+ in a game with less than $|A|$ actions for each player for T iterations, the resulting average strategy is an $(\sqrt{|A|/T})$-coarse correlated equilibrium of the game.

Algorithms for learning in simple and complex games

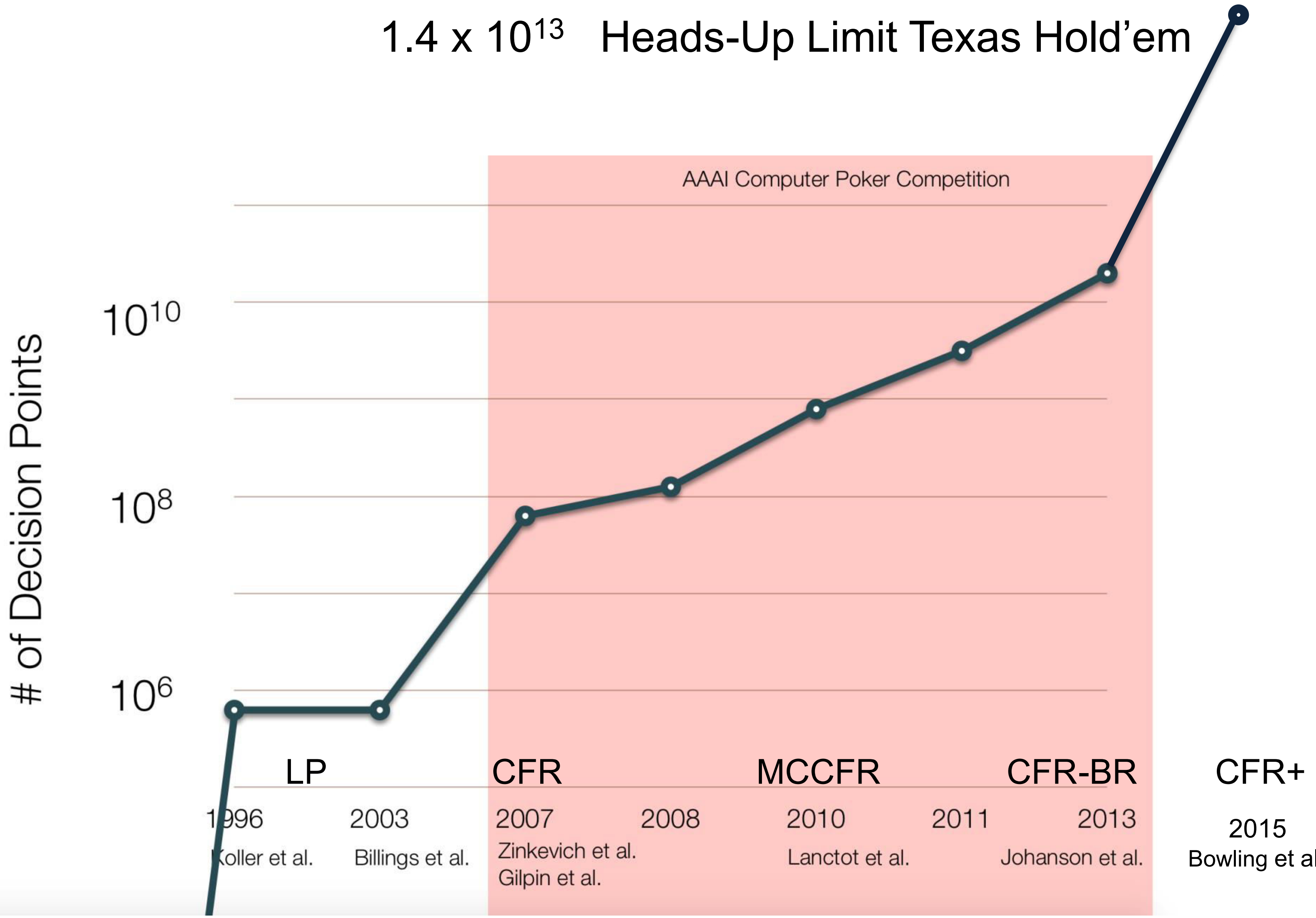
Learning in Extensive Form Games

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

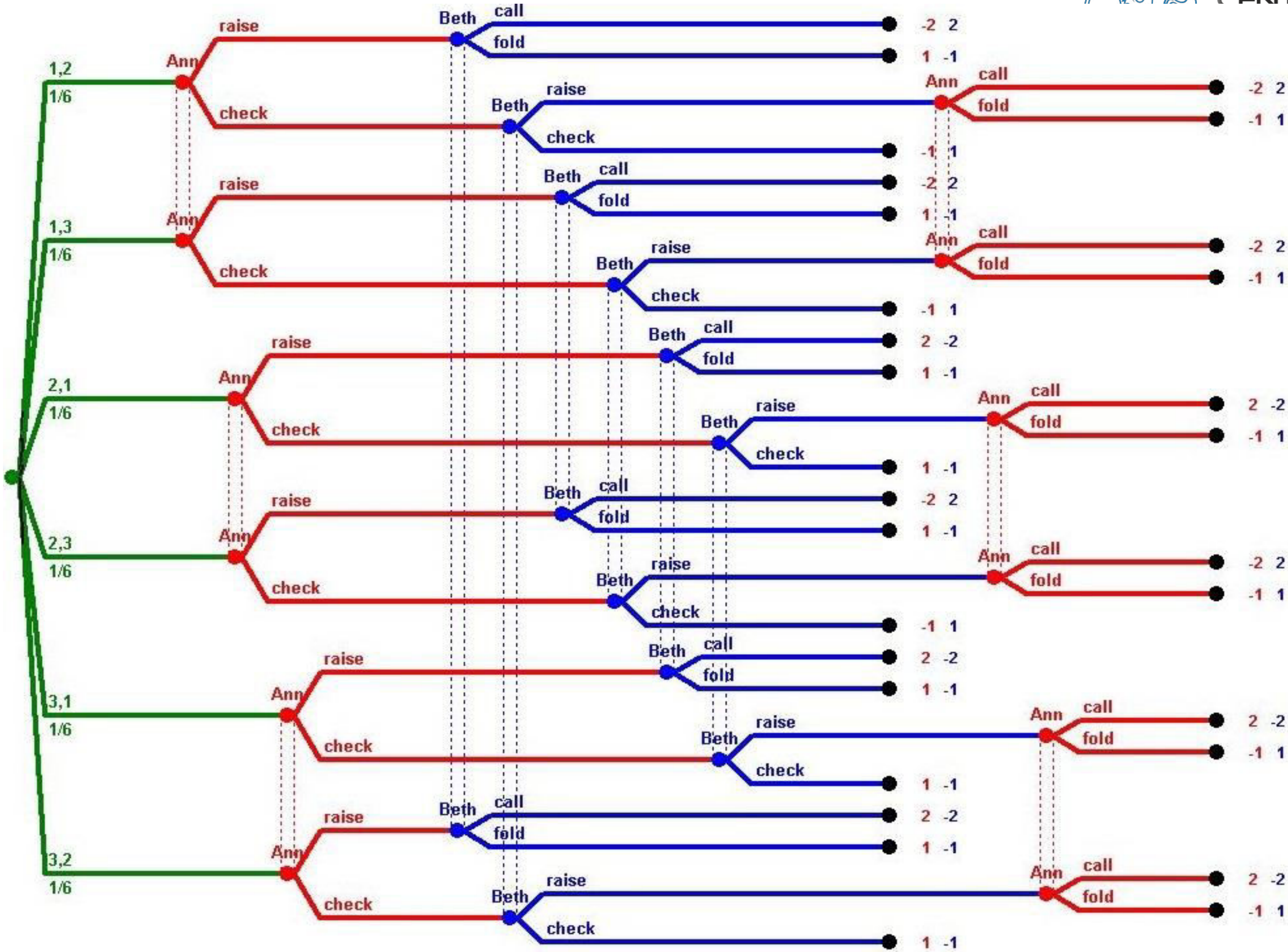
(Sep 25, 2018)

Impact on poker performance

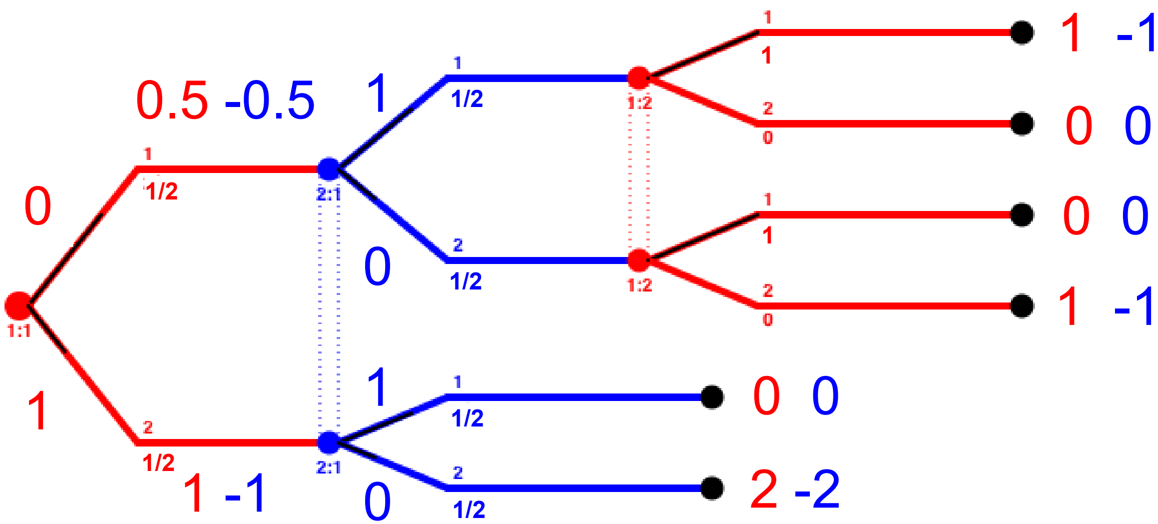


Based on M. Bowling's slide from AAAI 2015 keynote

Extensive form games



Counterfactual Regret - Motivation



1	0
0	1

0	2

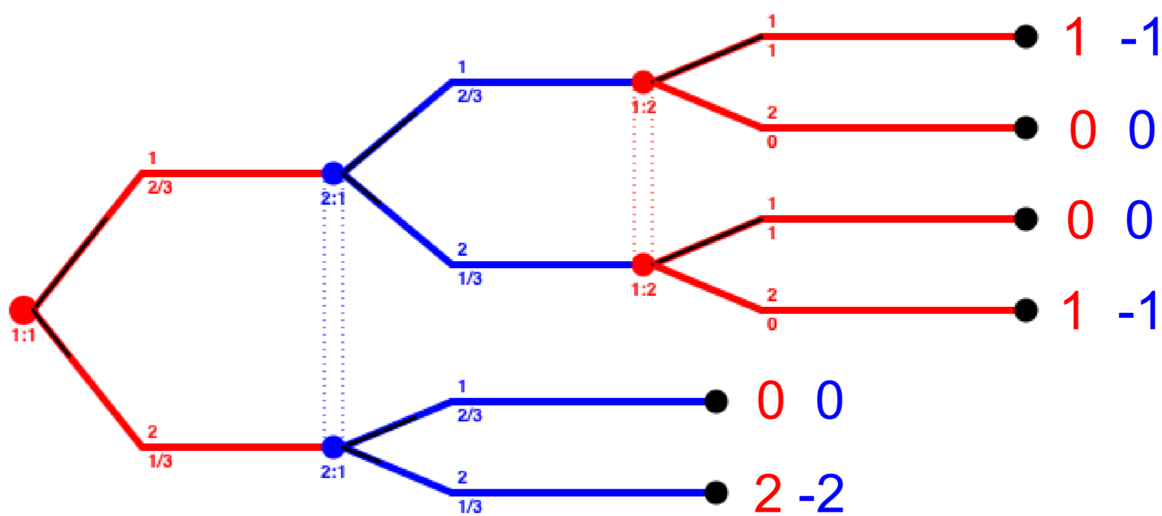
Take the current reach probabilities?

-> undefined belief

Take only opponent’s reach probability!

-> defined where necessary

Counterfactual Regret - Definition



Counterfactual value: $v_i^{\sigma}(I,a) = \sum_{(h,z)\in Z_I} \pi_{-i}^{\sigma}(h)\, \pi^{\sigma}(ha,z) u_i(z)$

Counterfactual regret: $r^t(I,a) = v_i^{\sigma^t}(I,a) - v_i^{\sigma^t}(I)$

Can be computed in **one tree walk**

Counterfactual Regret Minimization

1) Walk the tree to compute conterfactual values in all ISs

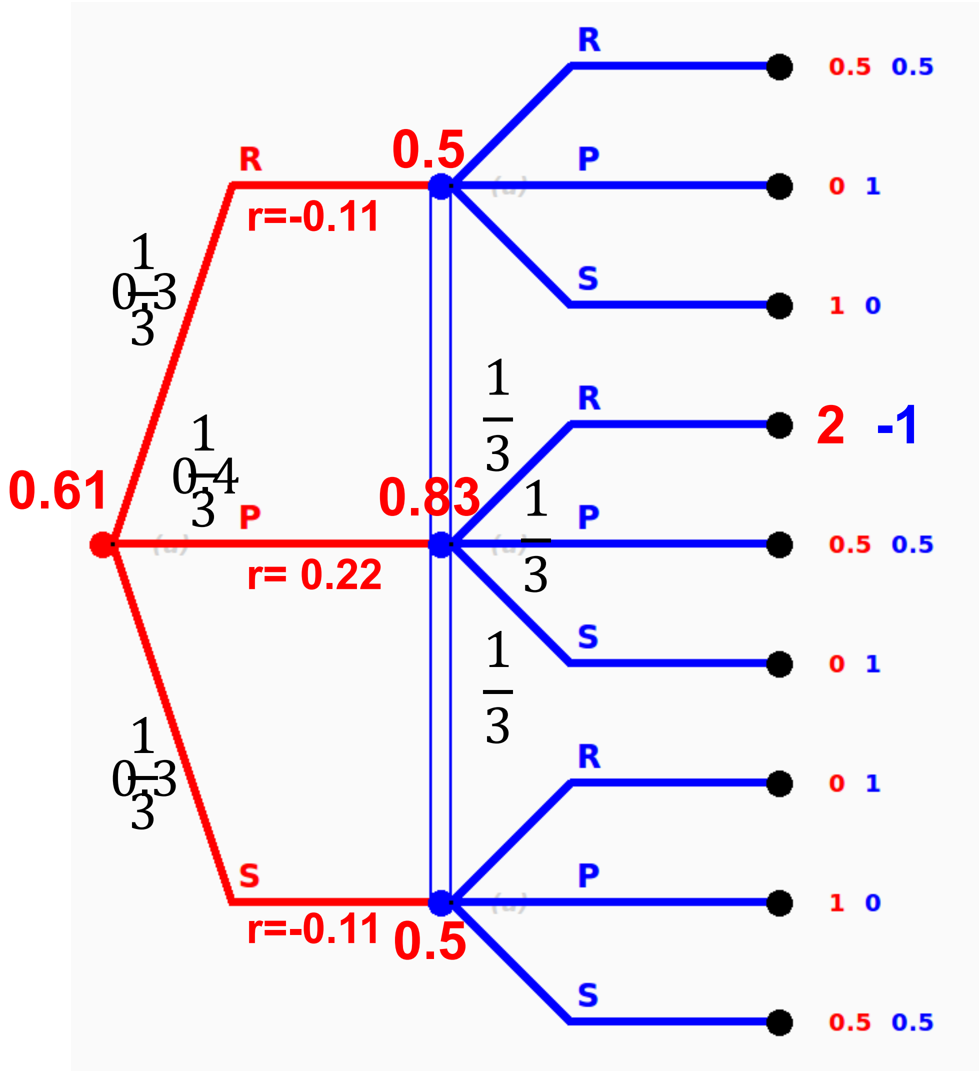
2) Use RM, RM+, Hedge,… to compute next strategy for each IS

3) Goto 1

4) Return **mean** of all used strategies

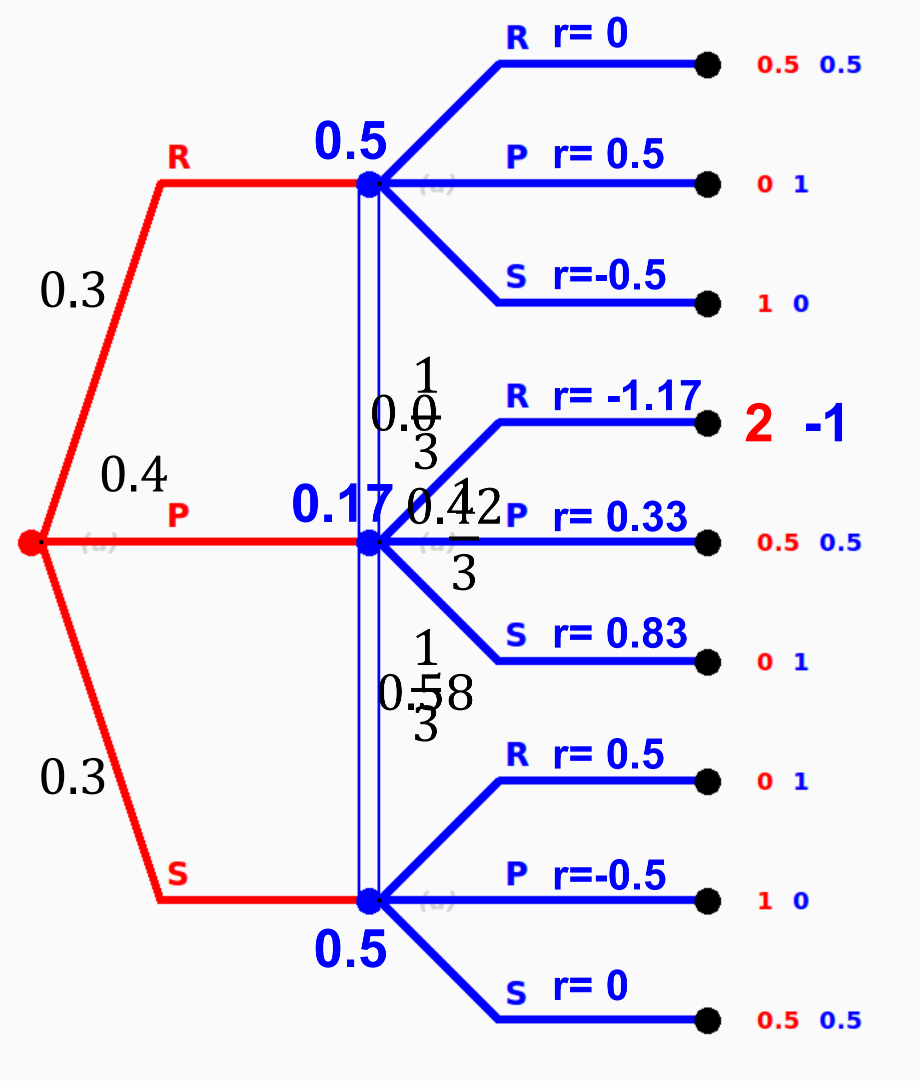
Counterfactual regret minimization

Player 1 iteration



Counterfactual regret minimization

Player 2 iteration



R: 0.3*0
+ 0.4*-1.17
+ 0.3*0.5
= -0.318

P: 0.3*0.5
+ 0.4*0.33
+ 0.3*-0.5
= 0.132

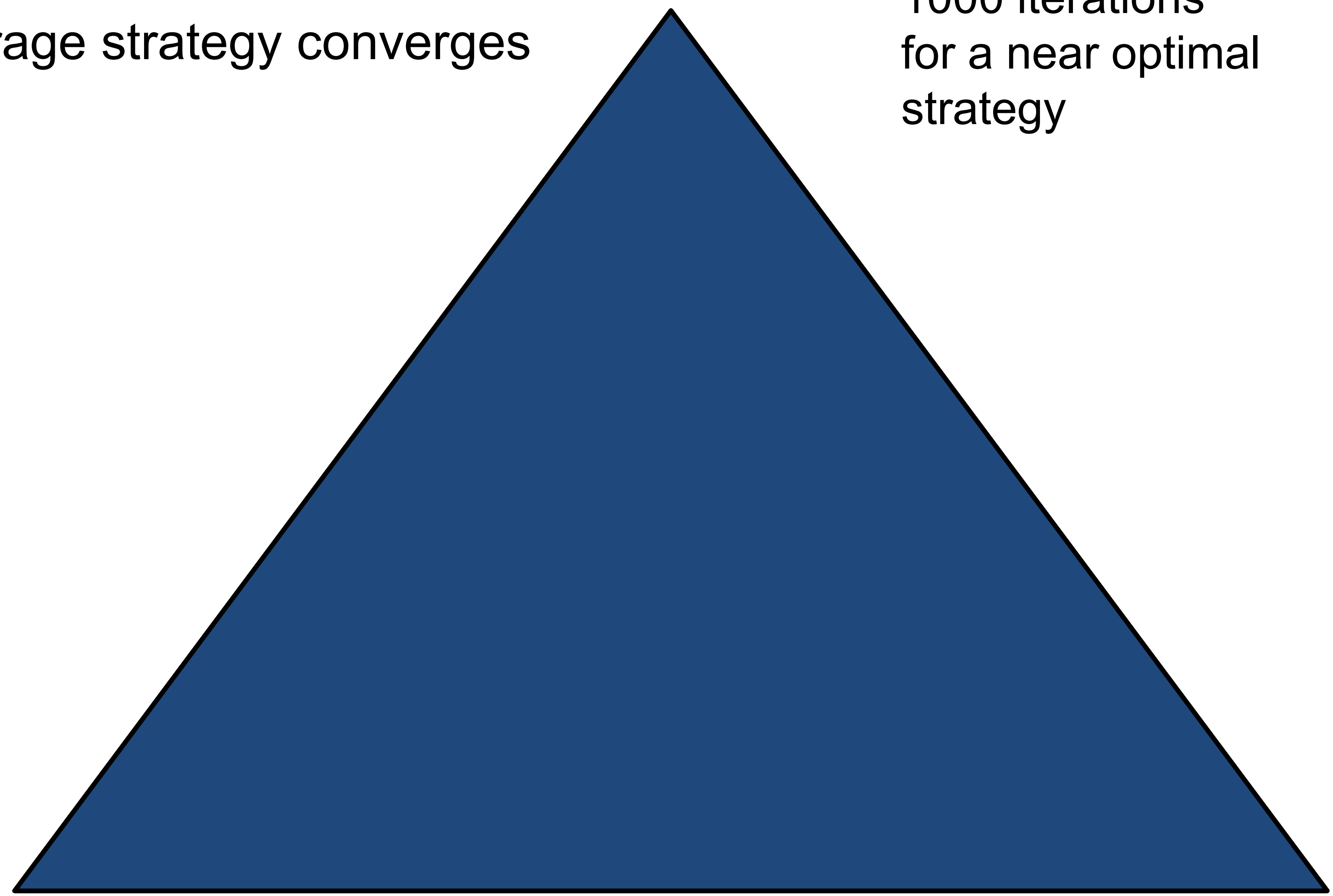
S: 0.3*-0.5
+ 0.4*0.83
+ 0.3*0
= 0.182

Counterfactual regret minimization

Each iteration requires full tree traversal

Average strategy converges

1000 iterations
for a near optimal
strategy



Counterfactual Regret Minimization

Theorem (Zinkevich et al. 2008): For a sequence of (mixed) strategies σ_i^t, let $R_{i,imm}^T(I) = \max_a \sum_{t\in 1..T} r^t(I,a)$ then

$$R_{i,full}^T \leq \sum_I R_{i,imm}^{T,+}(I)$$

Proof: Let $D(I)$ be the information sets reachable from I, $Succ_i(I,a)$ be the possible next information sets, $Succ_i(I) = \bigcup_{a\in A(I)} Succ_i(I,a)$.

$$R_{i,full}^T(I) = \max_{\sigma'\in\Sigma_i} \sum_{t\in 1..T} \left(v_i\left(\sigma^t\big|_{D(I)\to\sigma'}, I\right) - v_i(\sigma^t, I)\right)$$

$$v_i^\sigma(I,a) = \sum_{(h,z)\in Z_I} \pi_{-i}^\sigma(h)\,\pi^\sigma(ha,z) u_i(z); \quad r^t(I,a) = v_i^{\sigma^t}(I,a) - v_i^{\sigma^t}(I)$$

$$R_{i,imm}^T(I) = \max_{a\in A(I)} \sum_{t\in 1..T} (v_i(\sigma^t|_{I\to a}, I) - v_i(\sigma^t, I))$$

Lemma: $R^T_{i,full}(I) \leq R^T_{i,imm}(I) + \sum_{I' \in Succ_i(I)} R^{T,+}_{i,full}(I')$

$$R^T_{i,full}(I) = \max_{a \in A(I)} \max_{\sigma' \in \Sigma_i} \sum_{t \in 1..T}$$

$$(v_i(\sigma^t|_{I \to a}, I) - v_i(\sigma^t, I)$$

$$+ \sum_{I' \in Succ_i(I,a)} succ_i^{\sigma}(I'|I,a)\left(\frac{\pi_{-i}^{\sigma^t}(I)}{\pi_{-i}^{\sigma^t}(I')}\right)\left(v_i\left(\sigma^t|_{D(I) \to \sigma'}, I'\right) - v_i(\sigma^t, I')\right))$$

$$R^T_{i,full}(I) \leq \max_{a \in A(I)} \max_{\sigma' \in \Sigma_i} \sum_{t \in 1..T} (v_i(\sigma^t|_{I \to a}, I) - v_i(\sigma^t, I))$$

$$+ \max_{a \in A(I)} \max_{\sigma' \in \Sigma_i} \sum_{t \in 1..T} \sum_{I' \in Succ_i(I,a)} \left(v_i\left(\sigma^t|_{D(I') \to \sigma'}, I'\right) - v_i(\sigma^t, I')\right)$$

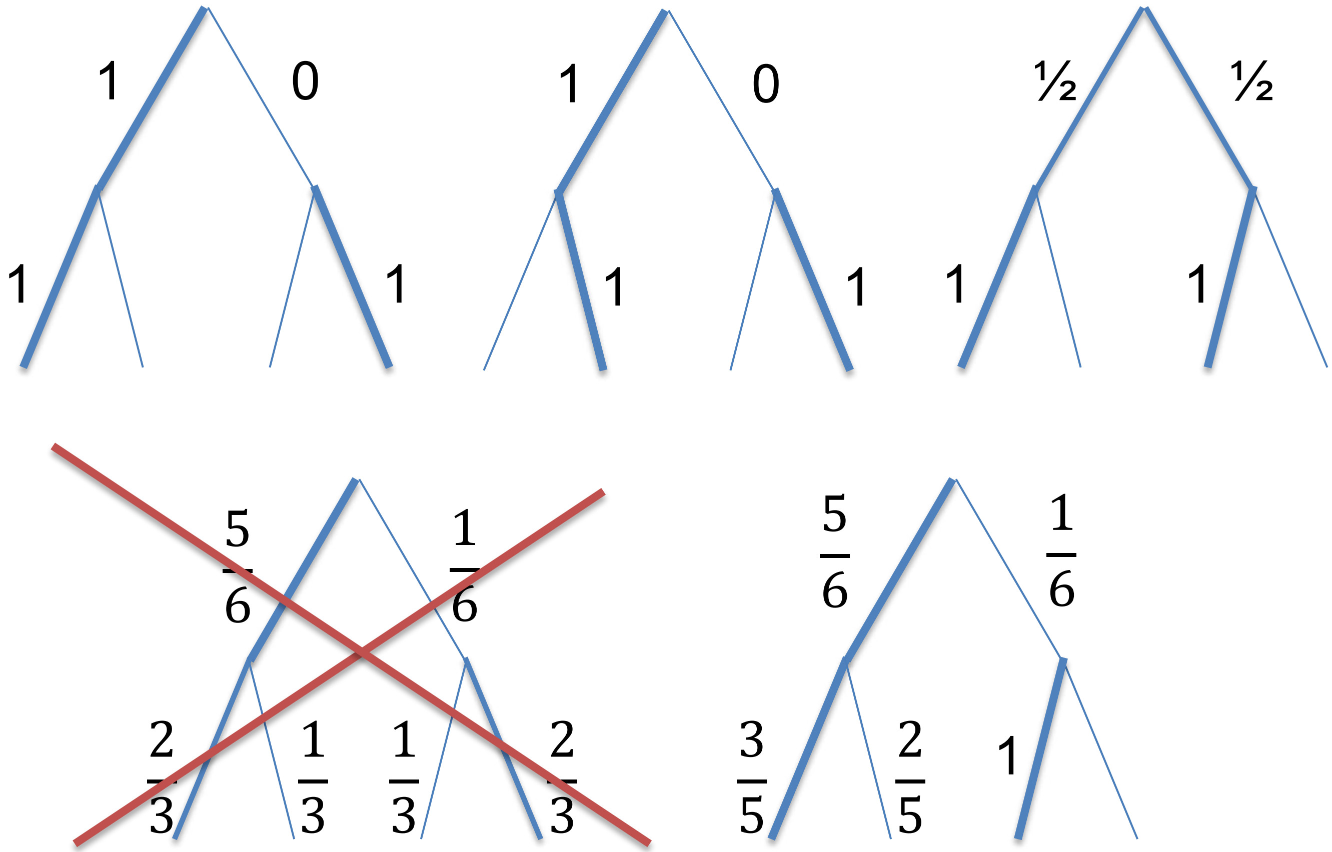
$$R^T_{i,full}(I) \leq R^T_{i,imm}(I) + \max_{a \in A(I)} \sum_{I' \in Succ_i(I,a)} R^T_{i,full}(I')$$

$$\leq R^T_{i,imm}(I) + \sum_{I' \in Succ_i(I)} R^{T,+}_{i,full}(I').$$

The proof of the theorem is completed by induction, using the Lemma above.

Average Strategy in CFR

$$\bar{\sigma}_i^T(I,a) = \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)\sigma^t(I,a)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)}$$



Weighted averaging!

CFR+ Convergence Speed

Theorem (Tammelin et al. 2015): The mean strategies form CFR+ in a game with payoff range Δ, $A = \max_{I} |A(I)|$, after T iterations form an $\frac{2(|I_1|+|I_2|)\Delta\sqrt{A}}{\sqrt{T}}$-Nash equilibrium.

CFR Variants – CFR-BR

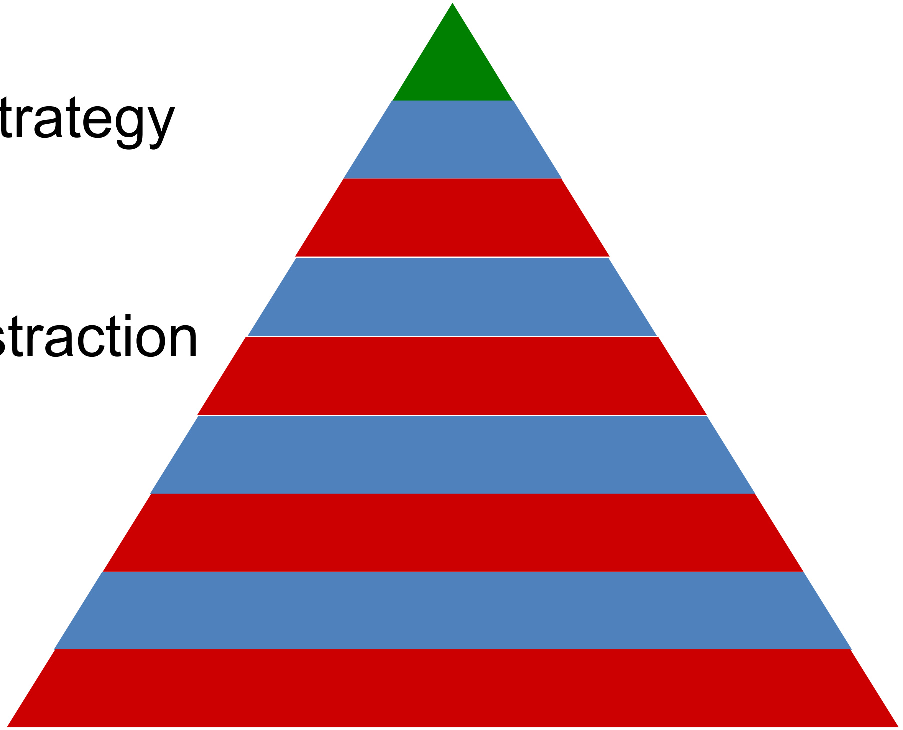
Opponent always plays best response (Johanson et al. 2012)

No storage for the opponent's strategy

No need for average strategy

Opponent can play in a finer abstraction

- Infinite strategy space
- Optimal abstract strategies



CFR Variants – CFR-BR

Theorem (Johanson et al. 2012):

After T iterations, the average strategy of CFR-BR converges to

$$\frac{\Delta|\mathrm{I}_1|\sqrt{|A_1|}}{\sqrt{T}}\text{-Nash equilibrium}$$

Proof sketch:

CFR player: $\sigma_i^0, \sigma_i^1, \ldots, \sigma_i^T$ - no regret sequence of strategies

BR player: $BR(\sigma_i^0), BR(\sigma_i^1), \ldots, BR(\sigma_i^T)$

Both players eventually have external regret $< \epsilon$

CFR Variants – CFR-BR

Theorem (Johanson et al. 2012):

After T iteration with probability (1-p) the **current strategy** of CFR-BR converges to

$\frac{\Delta|\mathrm{I}_1|\sqrt{|A_1|}}{p\sqrt{T}}$-Nash equilibrium

Proof sketch:

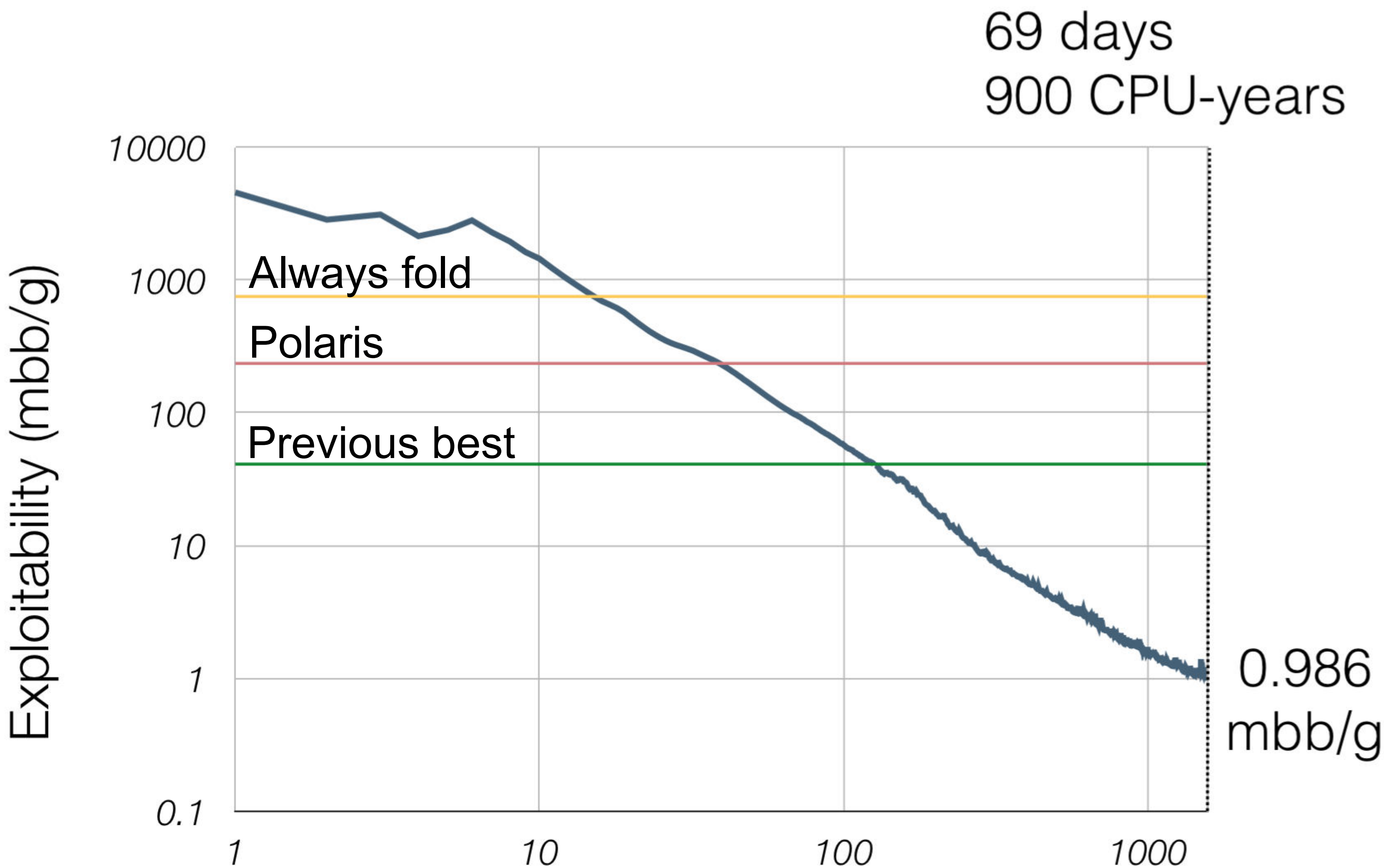
$$\bar{r}_{i,full}^{T} = \frac{1}{T}\max_{\sigma'}\sum_{t=1}^{T} u_i(\sigma', \sigma_{-i}^{t}) - \frac{1}{T}\sum_{t=1}^{T} u_i\left(\sigma_i^t, \sigma_{-i}^{t}\right) < \epsilon$$

$$\frac{1}{T}\sum_{t=1}^{T} u_i\left(\sigma_i^t, \sigma_{-i}^{t}\right) \geq \frac{1}{T}\max_{\sigma'}\sum_{t=1}^{T} u_i\left(\sigma', \sigma_{-i}^{t}\right) - \epsilon \geq \max_{\sigma'} u_i\left(\sigma', \bar{\sigma}_{-i}^{T}\right) - \epsilon$$

$\geq v_i^* - \epsilon$, but $u_i\left(\sigma_i^t, \sigma_{-i}^{t}\right) \leq v_i^*$, therefore $u_i\left(\sigma_i^t, \sigma_{-i}^{t}\right) > v_i^* - \frac{\epsilon}{p}$ often.

Solving Limit Texas Hold'em

(Bowling et al., Sience 2015)



Slide from M. Bowling's AAAI 2015 keynote

Plan

Online learning and prediction

- single agent learns to select the best action

Learning in normal form games

- the same algorithms used by multiple agents

Learning in extensive form games

- generalizing these ideas to sequential games

Brief introduction to neural networks

DeepStack

Algorithms for learning in simple and complex games

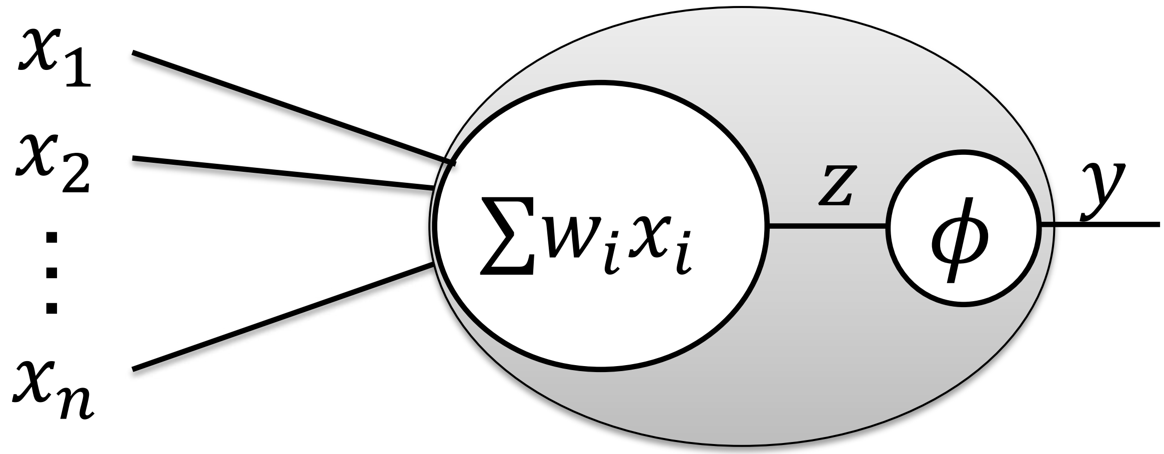
Brief Introduction to Neural Networks

Viliam Lisý

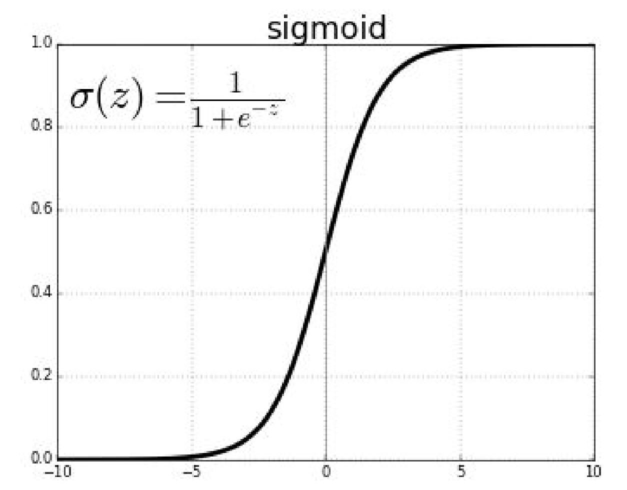
Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

(Sep 25, 2018)

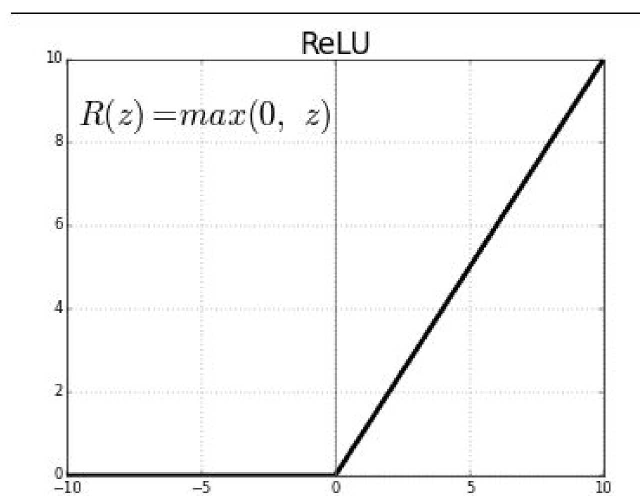
Neuron



x_1
x_2
x_n
$\sum w_i x_i$
z
ϕ
y

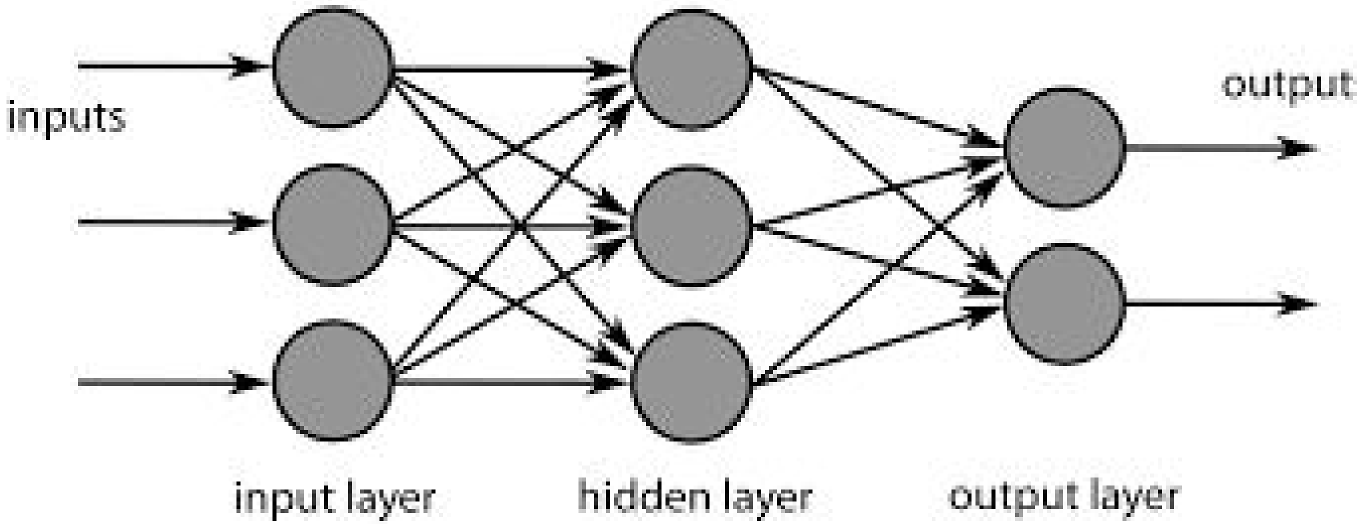


sigmoid
$\sigma(z) = \frac{1}{1+e^{-z}}$
1.0
0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10



ReLU
$R(z) = max(0, \ z)$
10
8
6
4
2
0
−10
−5
0
5
10

Neural Network

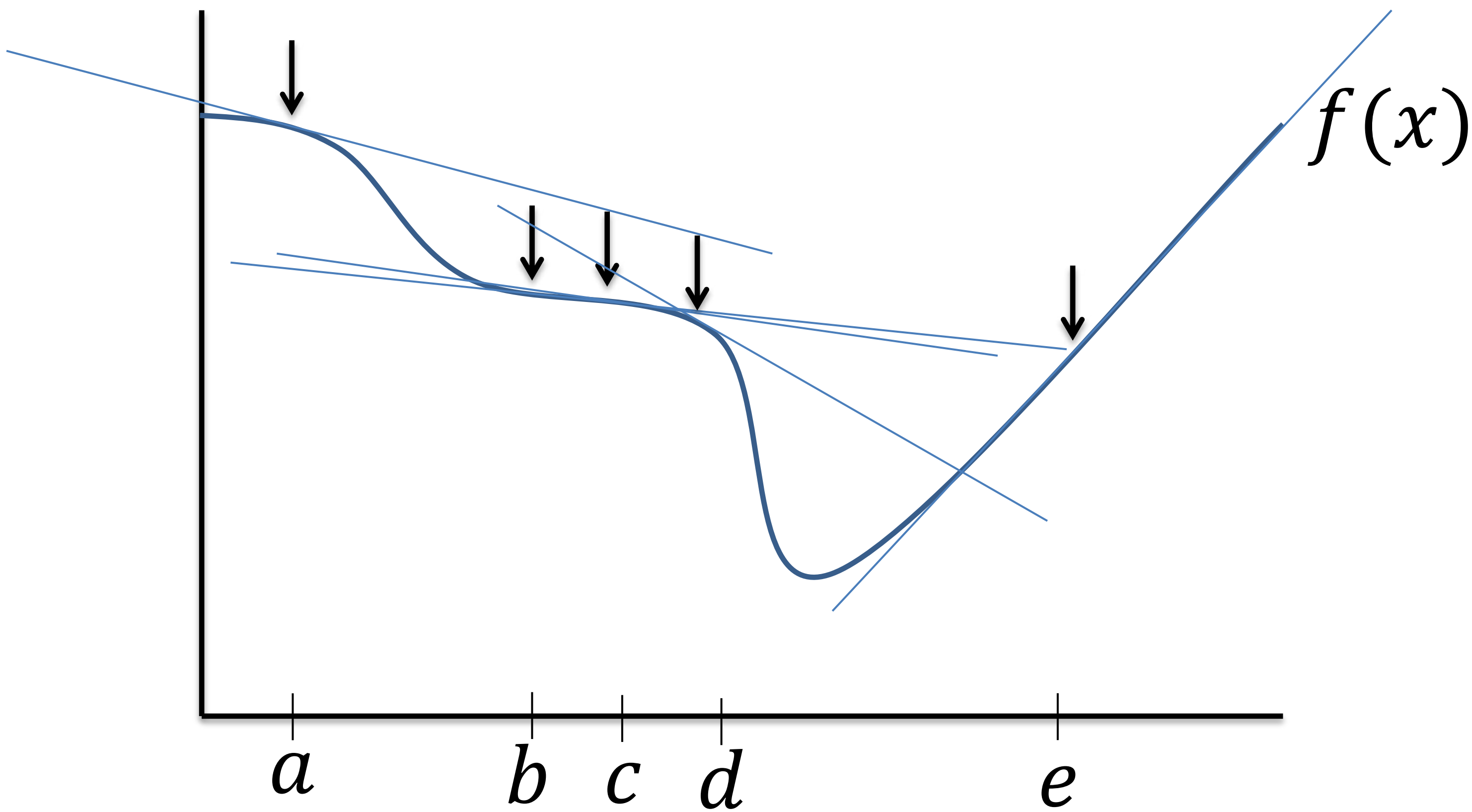


Universal approximation theorem (1989, etc.)

For any non-constant, monotonically increasing, bounded ϕ, a feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of R^n.

Gradient Descent

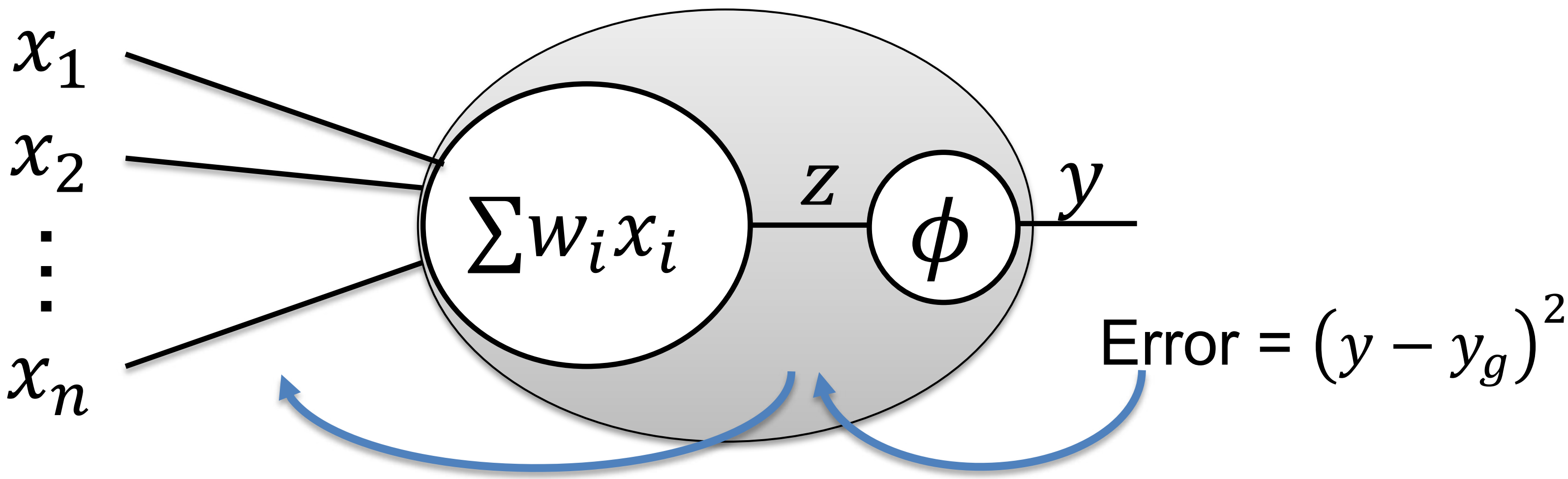
$$a_{n+1} = a_n - \gamma \nabla \mathrm{f}(\mathrm{a}_\mathrm{n})$$



$$= a = b - \frac{1}{3}() - \frac{1}{20})$$

Backpropagation

What is $\frac{\delta Error}{\delta w_i}$?



$$\frac{\delta Error}{\delta w_2} = \frac{\delta Error}{\delta y} * \frac{\delta y}{\delta z} * \frac{\delta z}{\delta w_2}$$

Stochastic gradient descent

$$a_{n+1} = a_n - \gamma \nabla \mathrm{f}(\mathrm{a_n})$$

$$\mathrm{f(x)} = \frac{1}{\mathrm{n}} \sum_i f(i, x)$$

$$a_{k+1} = a_k - \frac{\gamma}{n} \sum_i \nabla \mathrm{f(i, a_k)}$$

An unbiased estimate of the gradient is enough!

In practice, usually mini-batch and not a single sample.

Algorithms for learning in simple and complex games

DeepStack

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

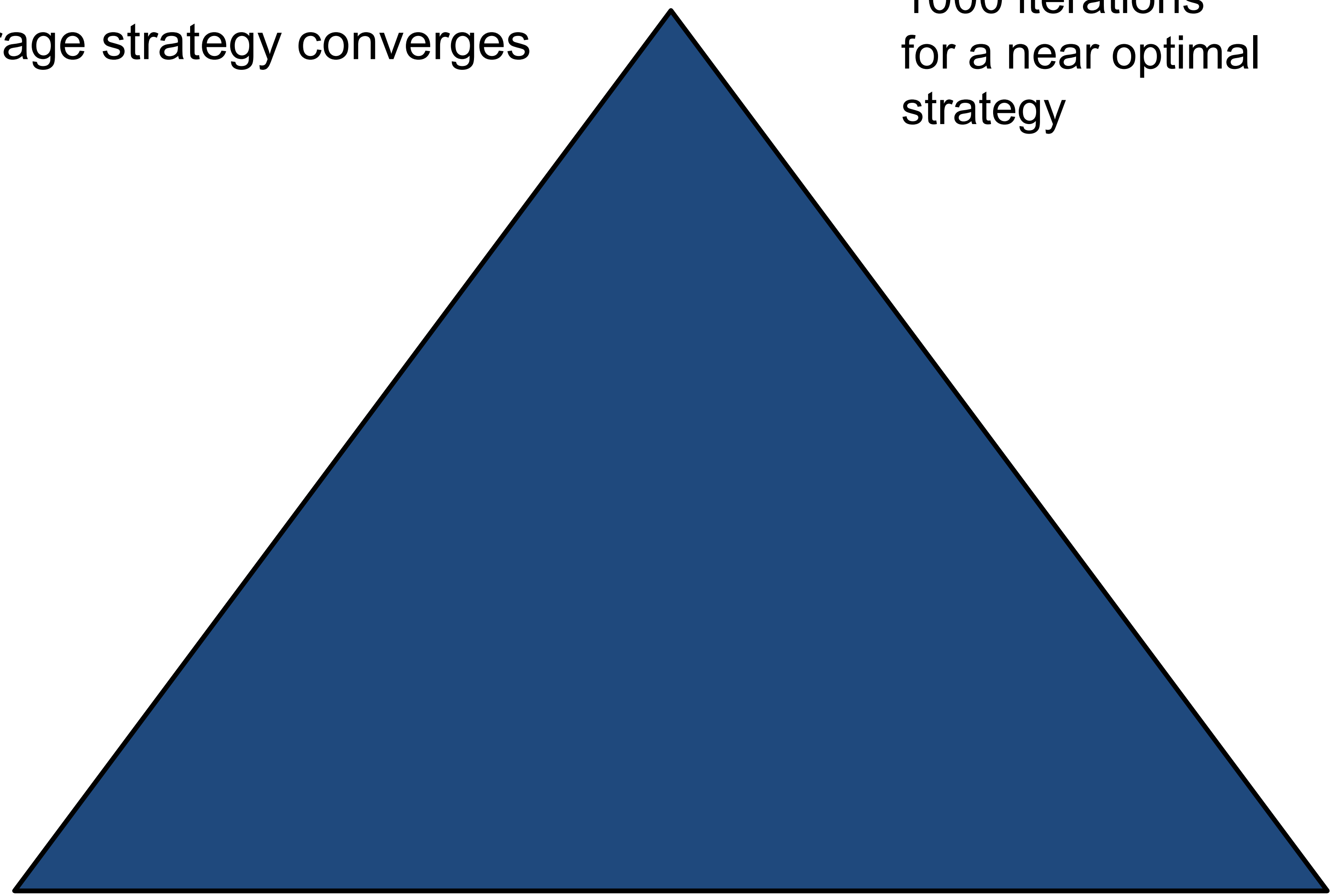
(Sep 25, 2018)

Counterfactual regret minimization

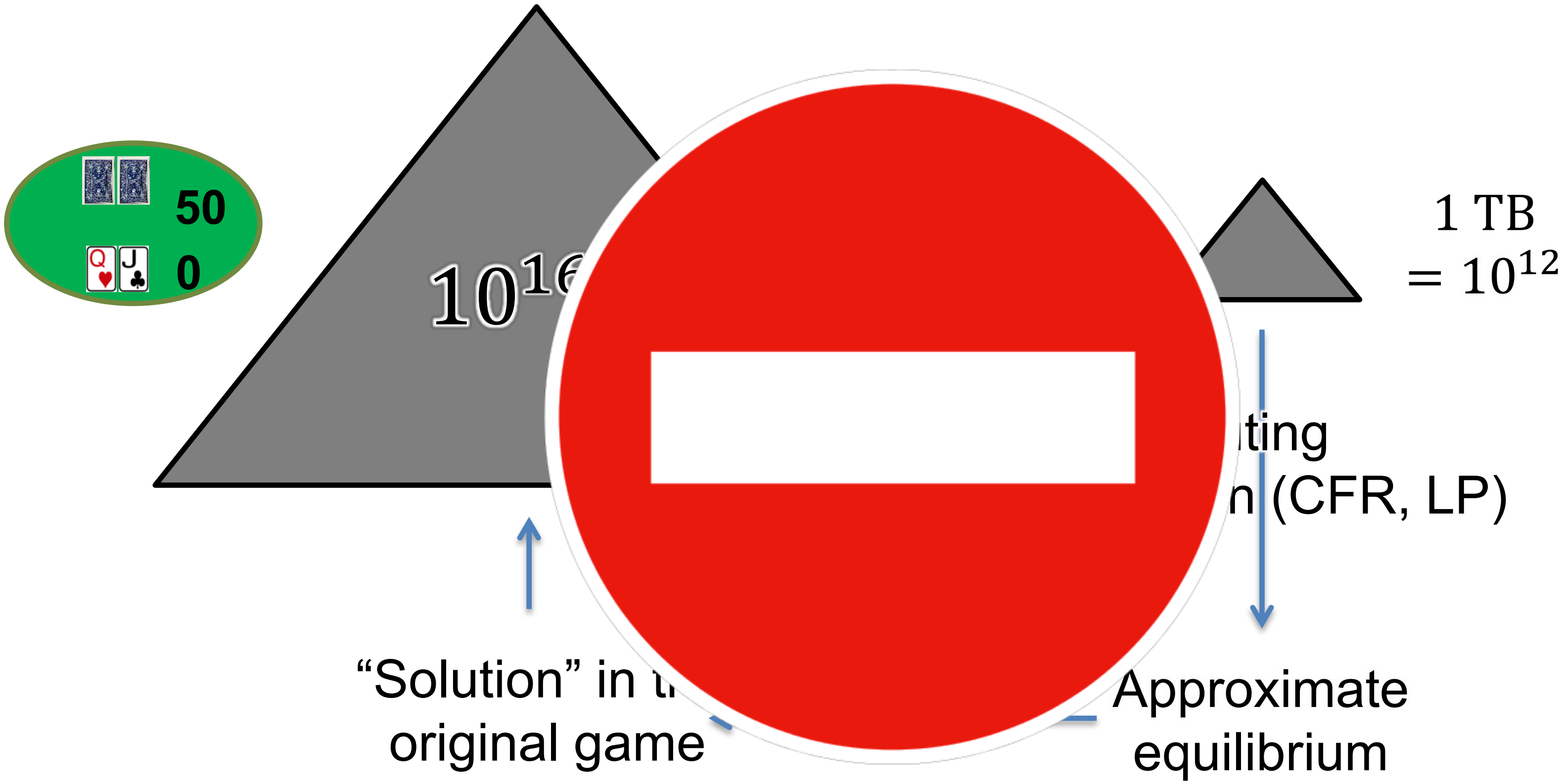
Each iteration requires full tree traversal

Average strategy converges

1000 iterations
for a near optimal
strategy

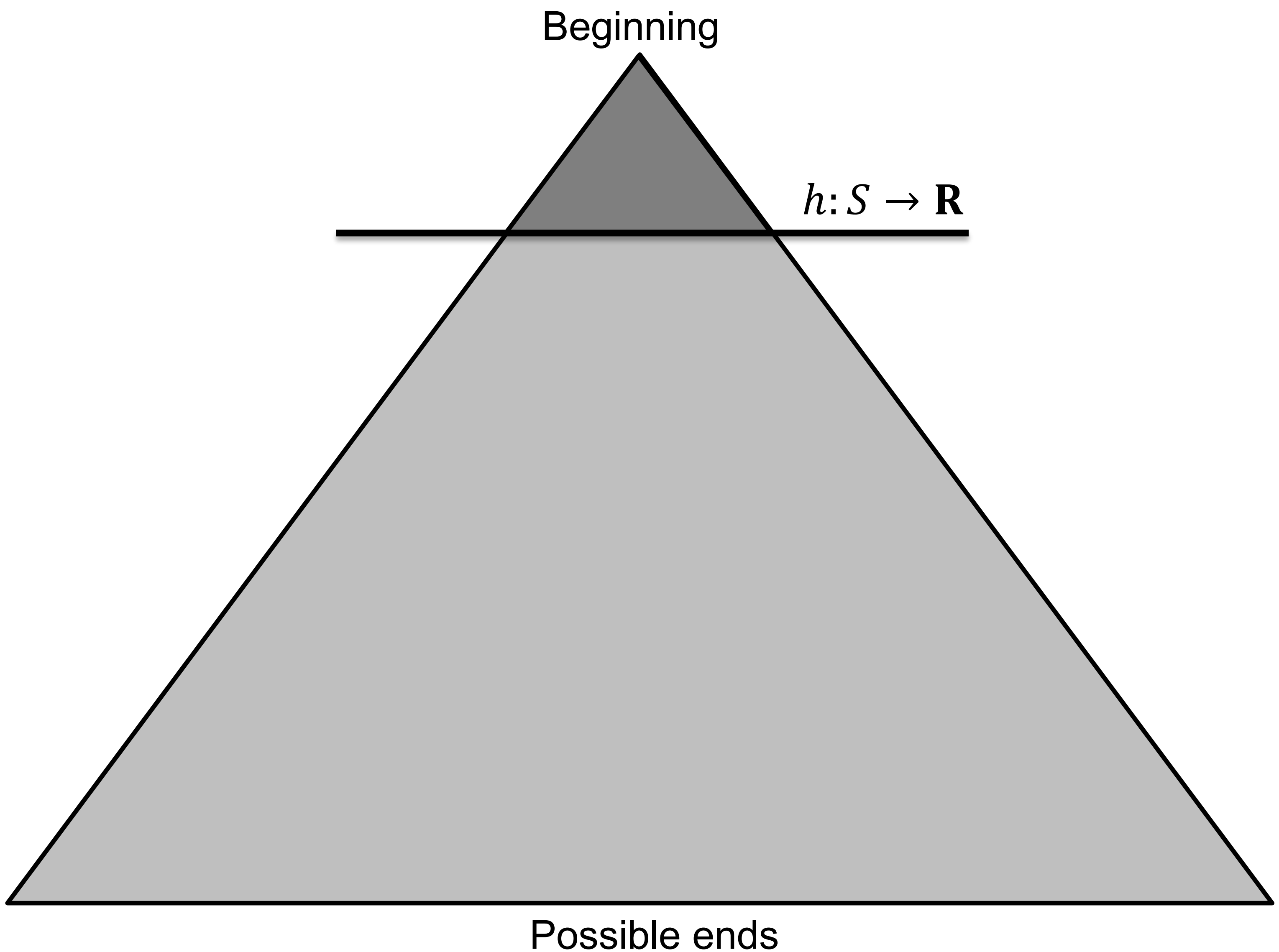


Computing strategies via abstraction

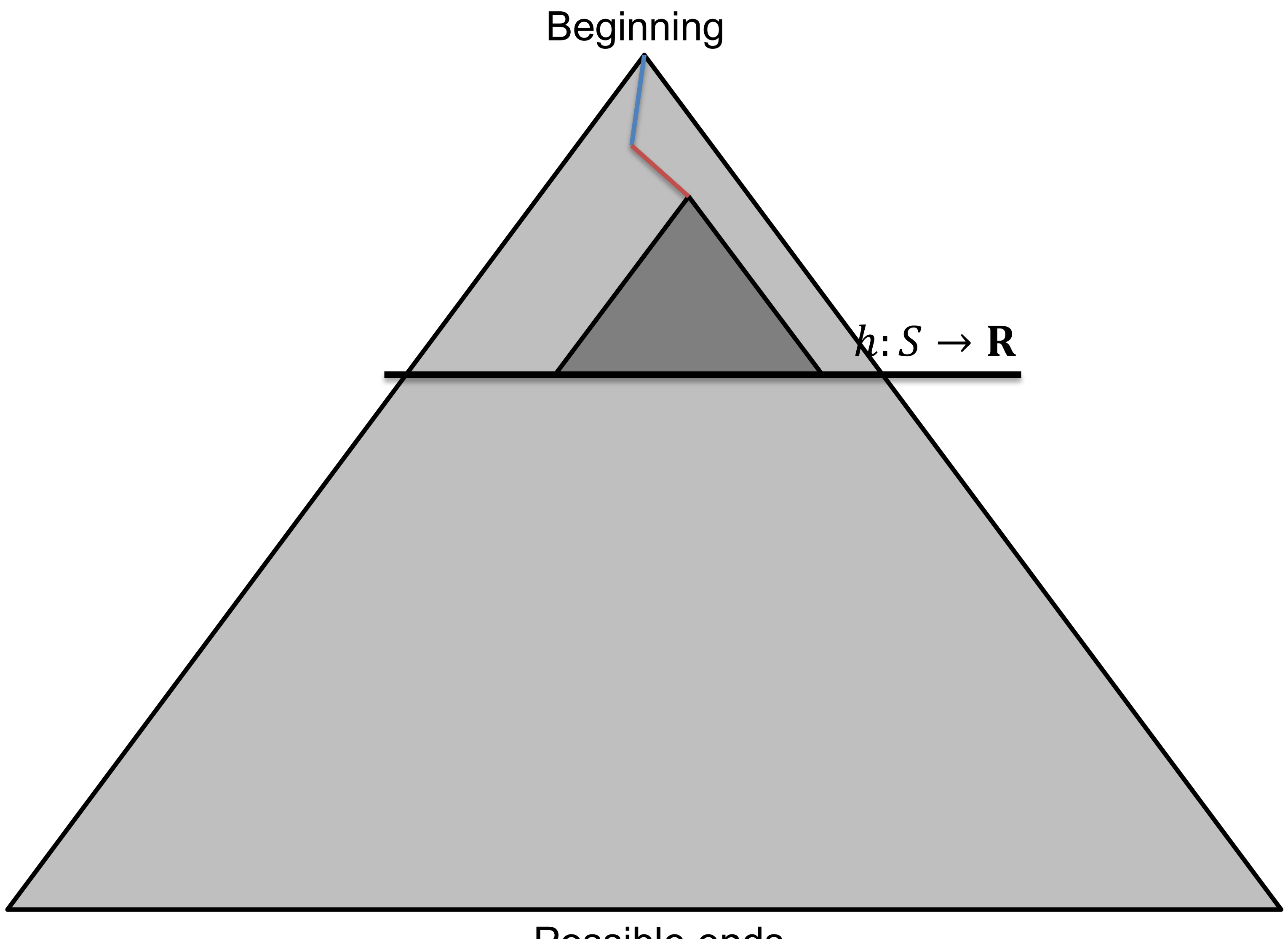


$$10^{148} : 1$$

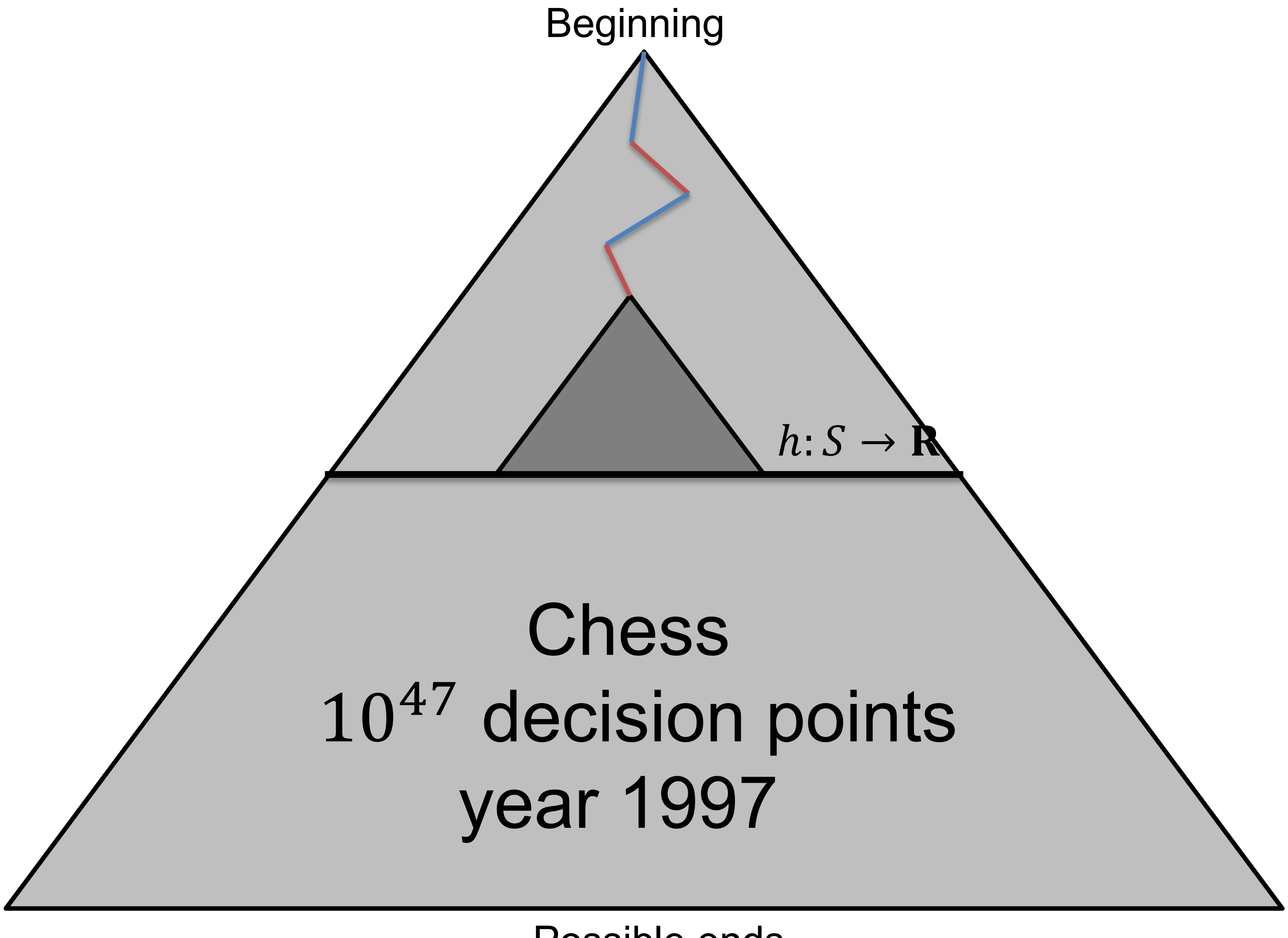
Depth limited look-ahead search



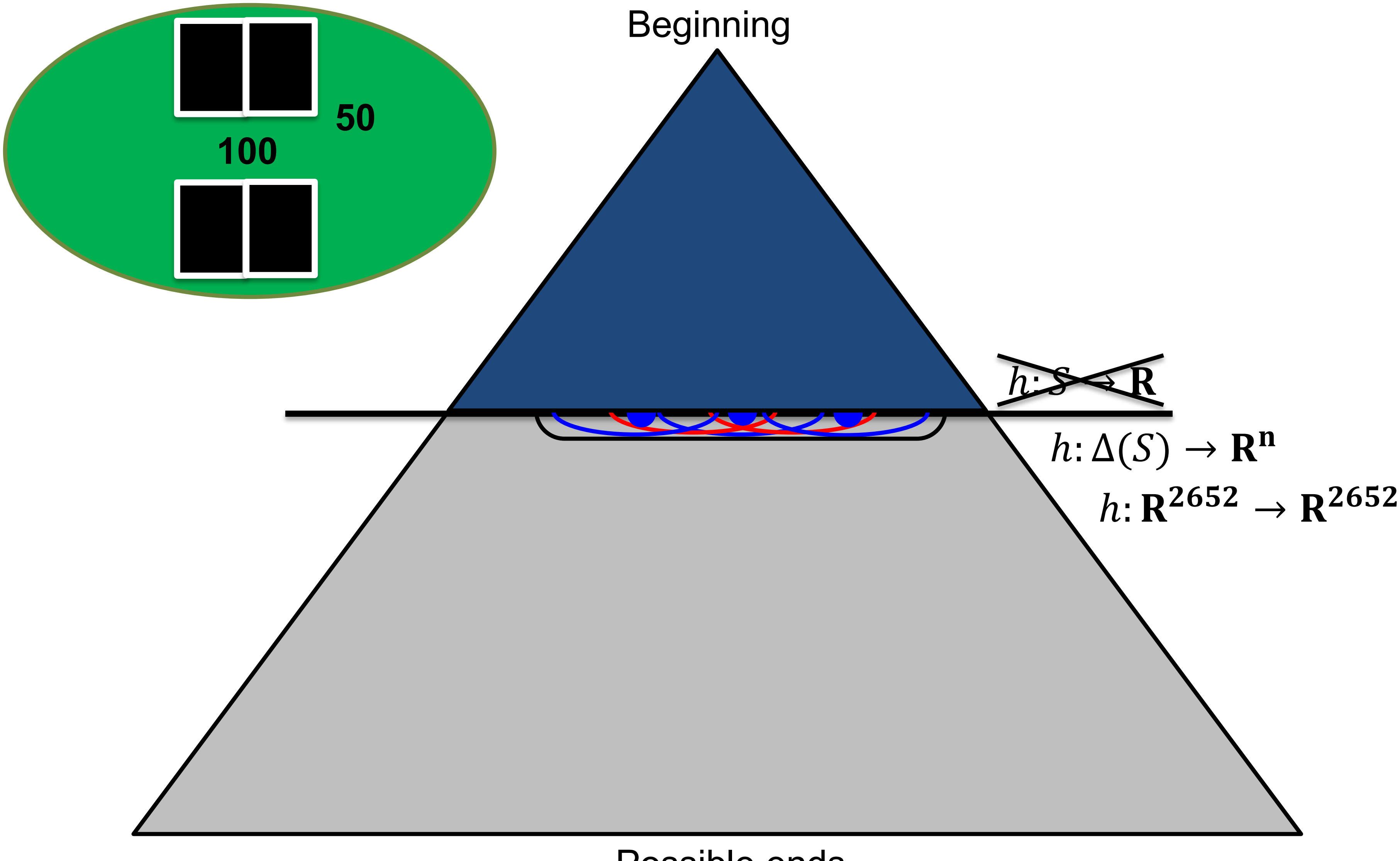
Depth limited look-ahead search



Depth limited look-ahead search



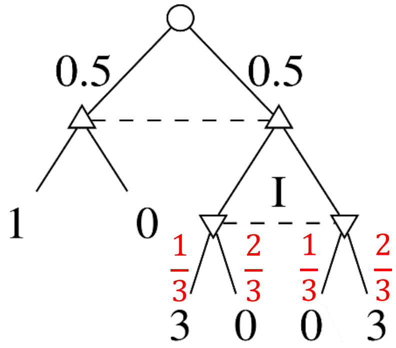
Depth limited look-ahead search



Game decomposition

Perfect information example

Imperfect information example



DeepStack team at University of Alberta

Photo: John Ulan for the University of Alberta

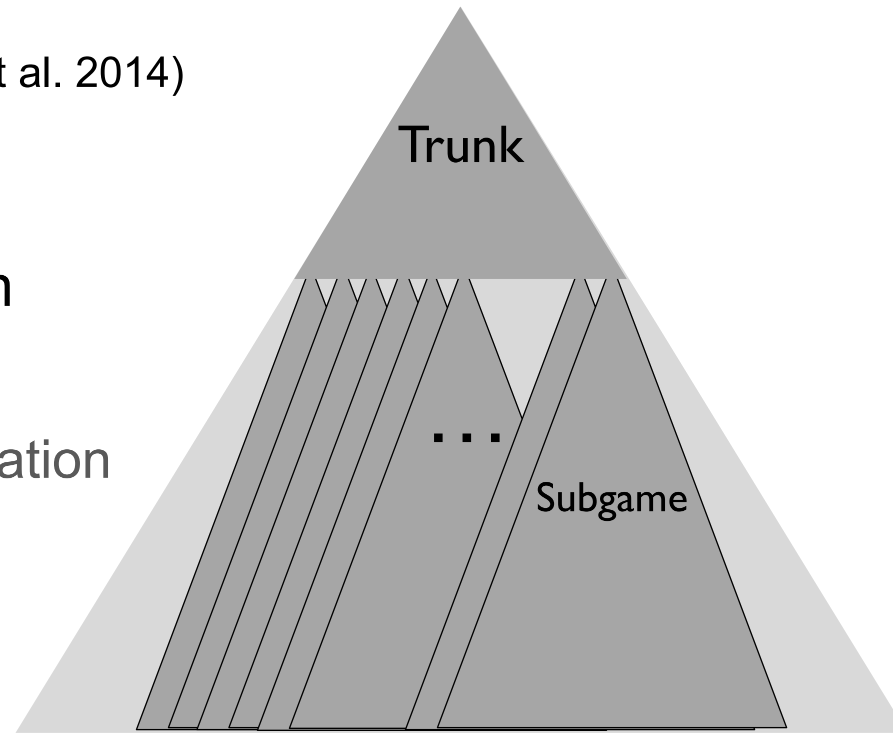
CFR-D

CFR with Decomposition (Burch et al. 2014)

Trades-of space for computation

- Store only the trunk
- Resolve subgames in each iteration

Resolve on demand in play



CDR-D

Augmented information set

Set on undistinguishable histories for any player, not just the deciding one

Subgame (denoted S)

forest of trees closed under descendance and belonging into augmented information sets

R(S)

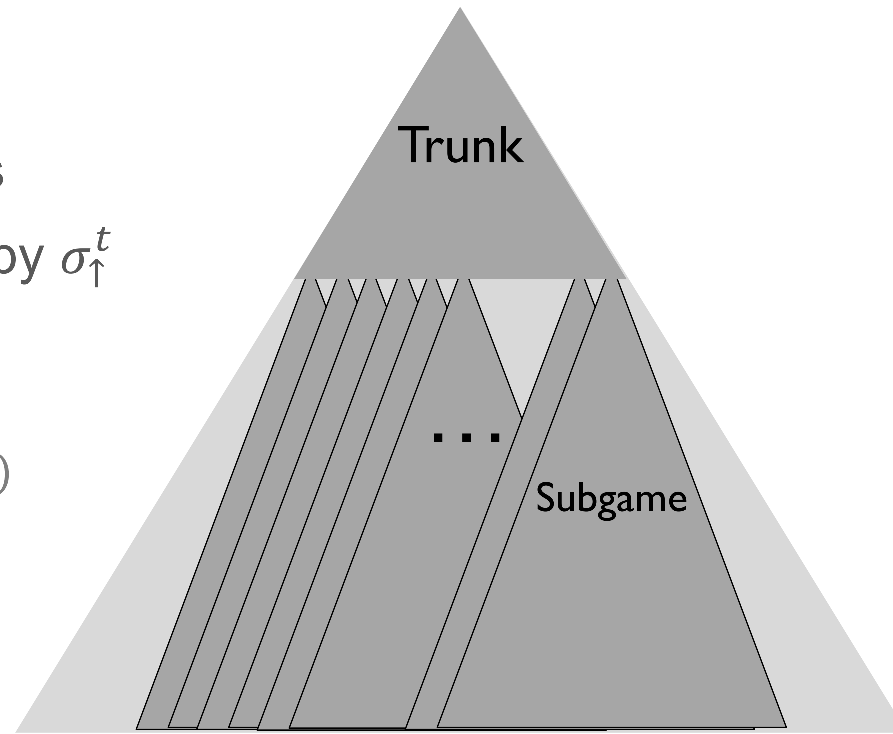
set of augmented information sets in the root of a subgame

CFR-D: Solving Trunk Strategy

Initialize regrets to 0

For iteration $t = 1, \ldots, T$

- compute $\sigma_{\uparrow}^{t}$ from stored regrets
- update trunk average strategy by $\sigma_{\uparrow}^{t}$
- For each subgame S
  - $\sigma_S^t \leftarrow$SOLVE(S, $\sigma_{\uparrow}^{t}$)
  - For each augmented $I_p \in R(S)$
    - Compute value v_{I_p}
    - Update average value $\mathrm{cf}v_{I_p}$
- Update trunk regrets using v_{I_p}



CFR-D: Computing Trunk Strategy

CFR-D: Resolving Subgame

Assume blue player played D and the game reached S1

Unsafe resolving

Save resolving

No incentive to change trunk!

CFR-D More Complicated Resolving

CFR-D Resolving Game

When resolving for player 1

- Create new chance node as the root
- Create new nodes for player 2 grouped by her “information sets”
- Connect the root to nodes in proportion to player 1 trunk strategy
- For each player 2 node, add follow action leading to subgame
- For each player 2 node, add terminate action with CFV of IS

We need

- Distribution in the root IS generated by player 1 trunk strategy
- Counterfactual value achievable by player 2 in his root ISs

CFR-D Convergence properties

CFR-D achieves no regret in the trunk

It the counterfactual regret at each information set I at the root of a subgame is bounded by ϵ_S, then than the average regret over the whole game is

$$R_{full}^{T} \leq \frac{N_{TR}\sqrt{A}}{\sqrt{T}} + N_S \epsilon_S$$

Proof sketch: $\sigma^0[S \leftarrow \sigma_S^{0,*}], \sigma^1[S \leftarrow \sigma_S^{1,*}], \ldots$

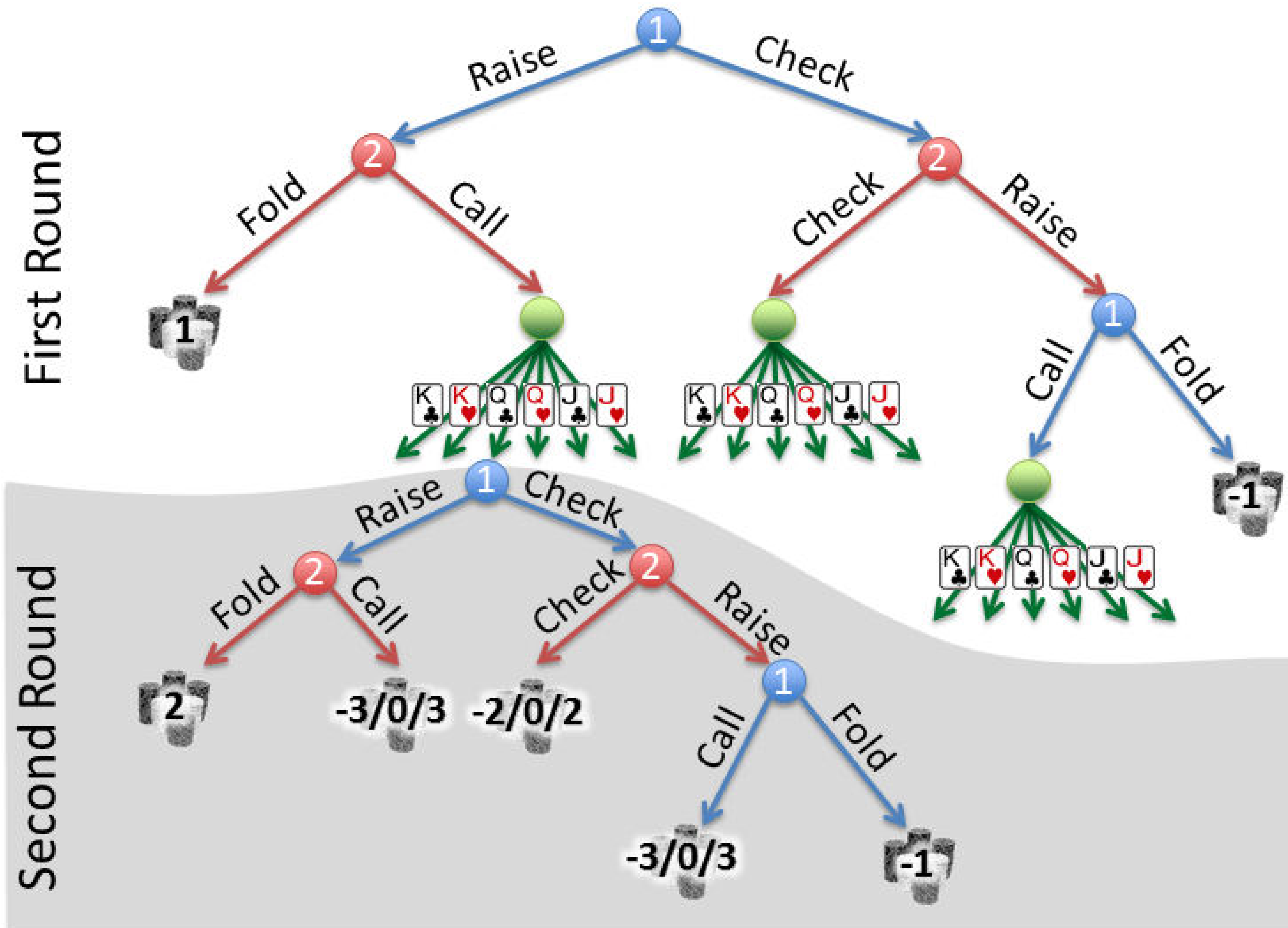
- CF regret in the trunk minimized by CFR
- CF regret in the subgame close to 0 for both players

CFR-D resolving forms a Nash equilibrium

If we run the recovery game for each player and each subgame until we reach regret below ϵ_R, the combined strategy has regret

$$R_{full}^{T} \leq \frac{N_{TR}\sqrt{A}}{\sqrt{T}} + N_S(3\epsilon_S + 2\epsilon_R)$$

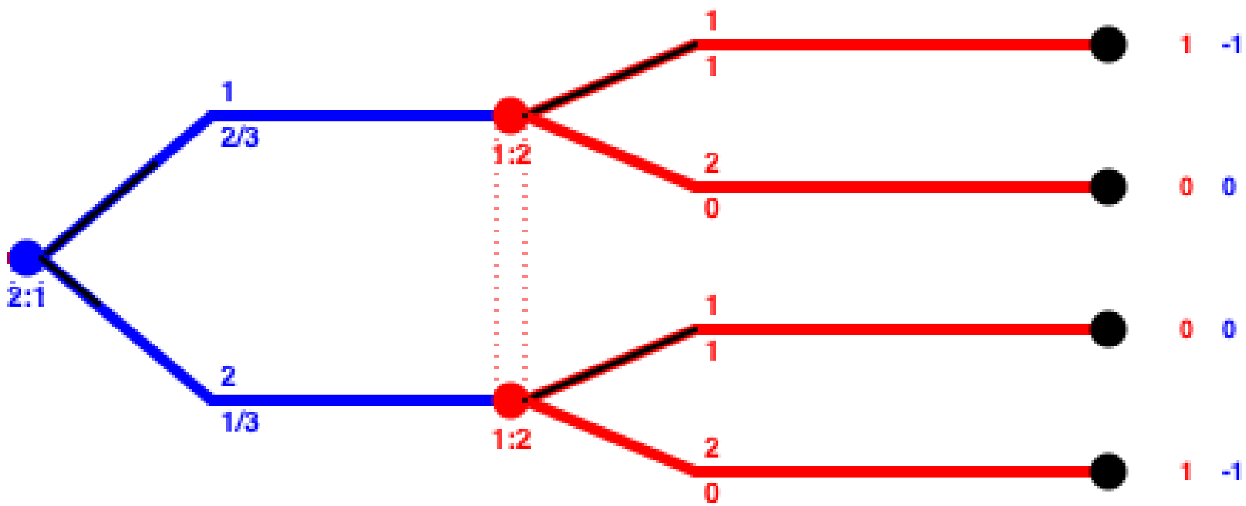
Public Tree



First Round
Second Round
1
2
Raise
Check
Fold
Call
1
-1
K
Q
J
Raise
Check
Fold
Call
2
-3/0/3
-2/0/2
Call
Fold
-3/0/3
-1

Public Tree

Matching pennies



Phantom Tic-Tac-Toe

Visibility-based pursuit-evasion games

Augmented IS in poker public node

Resolving poker subgame

To resolve, we need

$\forall I_1 \in R(S) \;\; \pi_1(I_1)$

$\forall I_2 \in R(S) \;\; cfv_2(I_2)$

In poker it means

$\pi_1(I_1)$ - probability that player 1 holds each hand = range

$cfv_2(I_2)$ - how much player 2 can win with each hand

In root (after chance reals hole cards)

$\pi_i(I_i)$ - uniform

$cfv_i(I_i)$ - pre-computed offline

DeepStack: updating maintained values

Assuming DeepStack is player 1

Own action

- replace player 2’s *cfv*s by the once computed in the resolve game
- update player 1’s range based on the played strategy

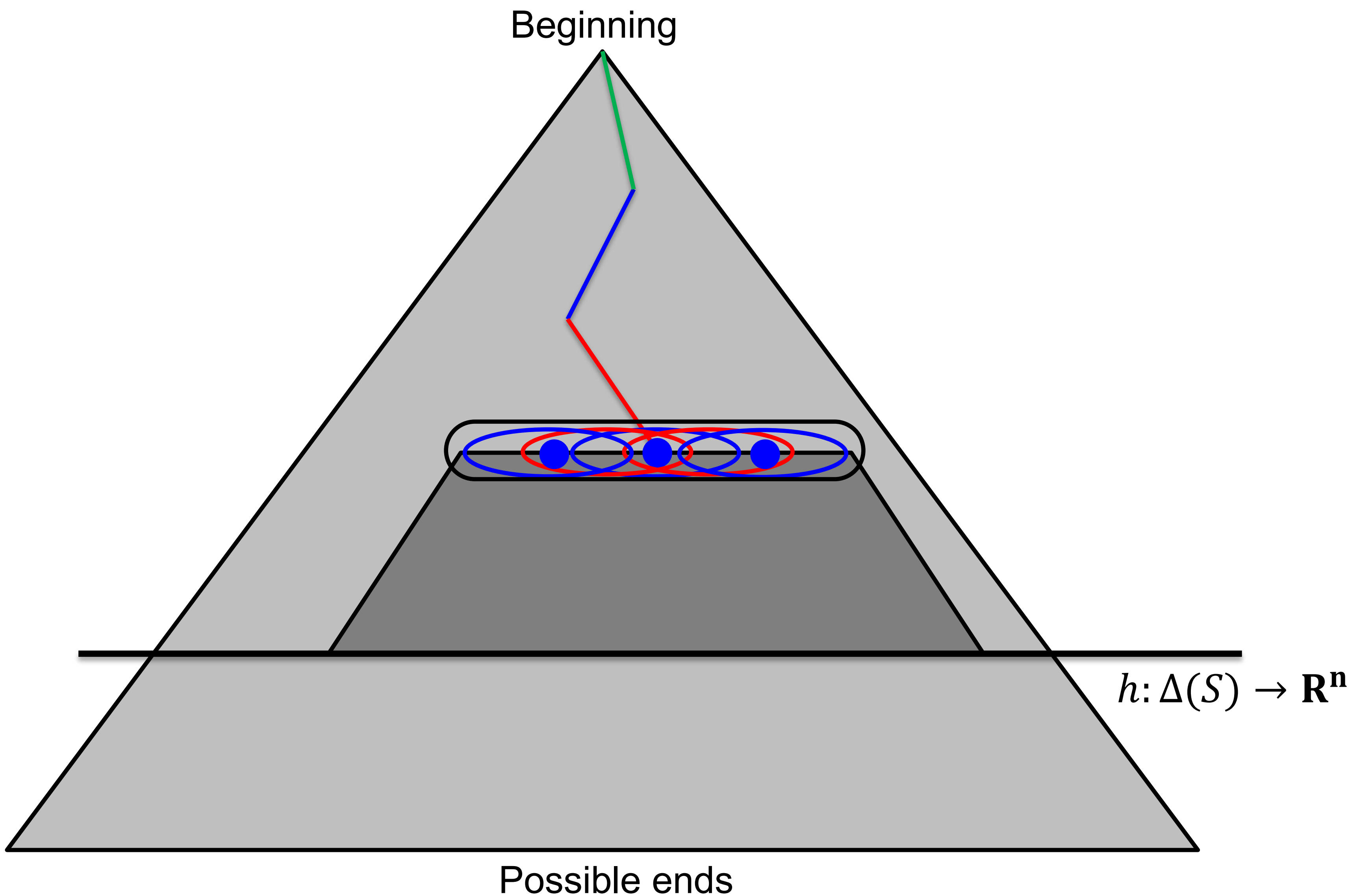
Chance action

- replace player 2’s *cfv*s from the last resolve above chance
- keep player 1’s range unchanged

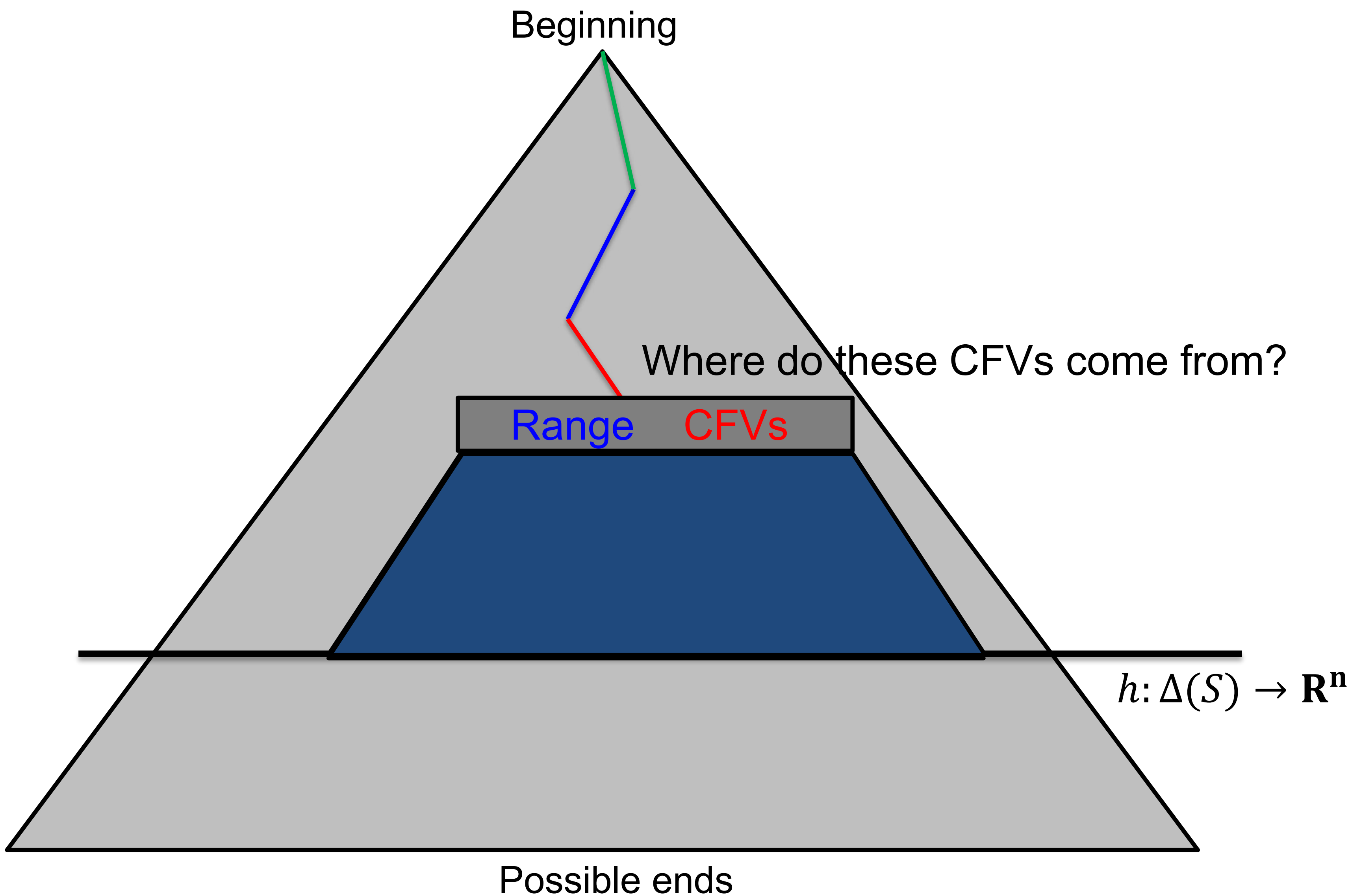
Opponent’s action

- no update required!

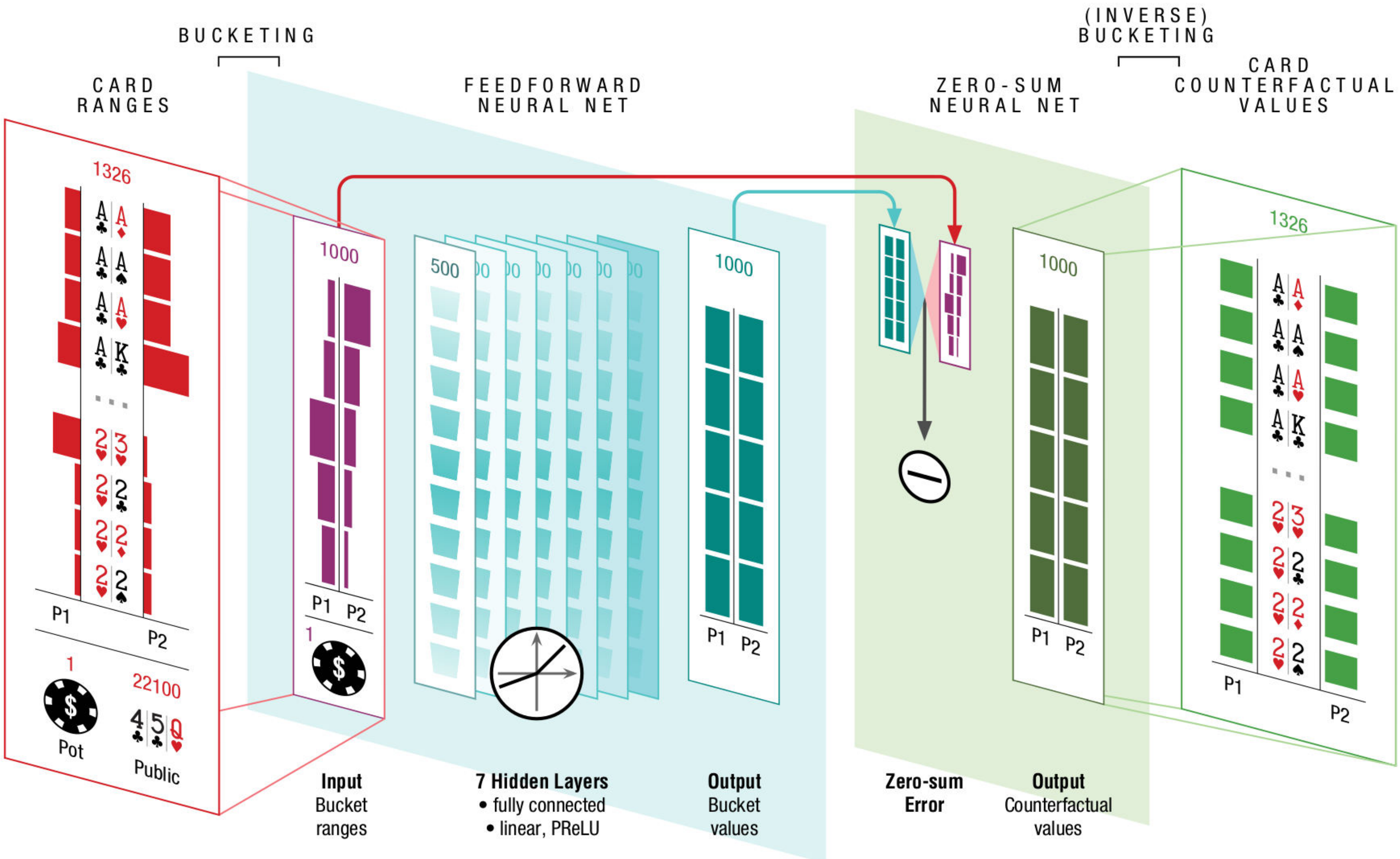
Depth limited look-ahead search



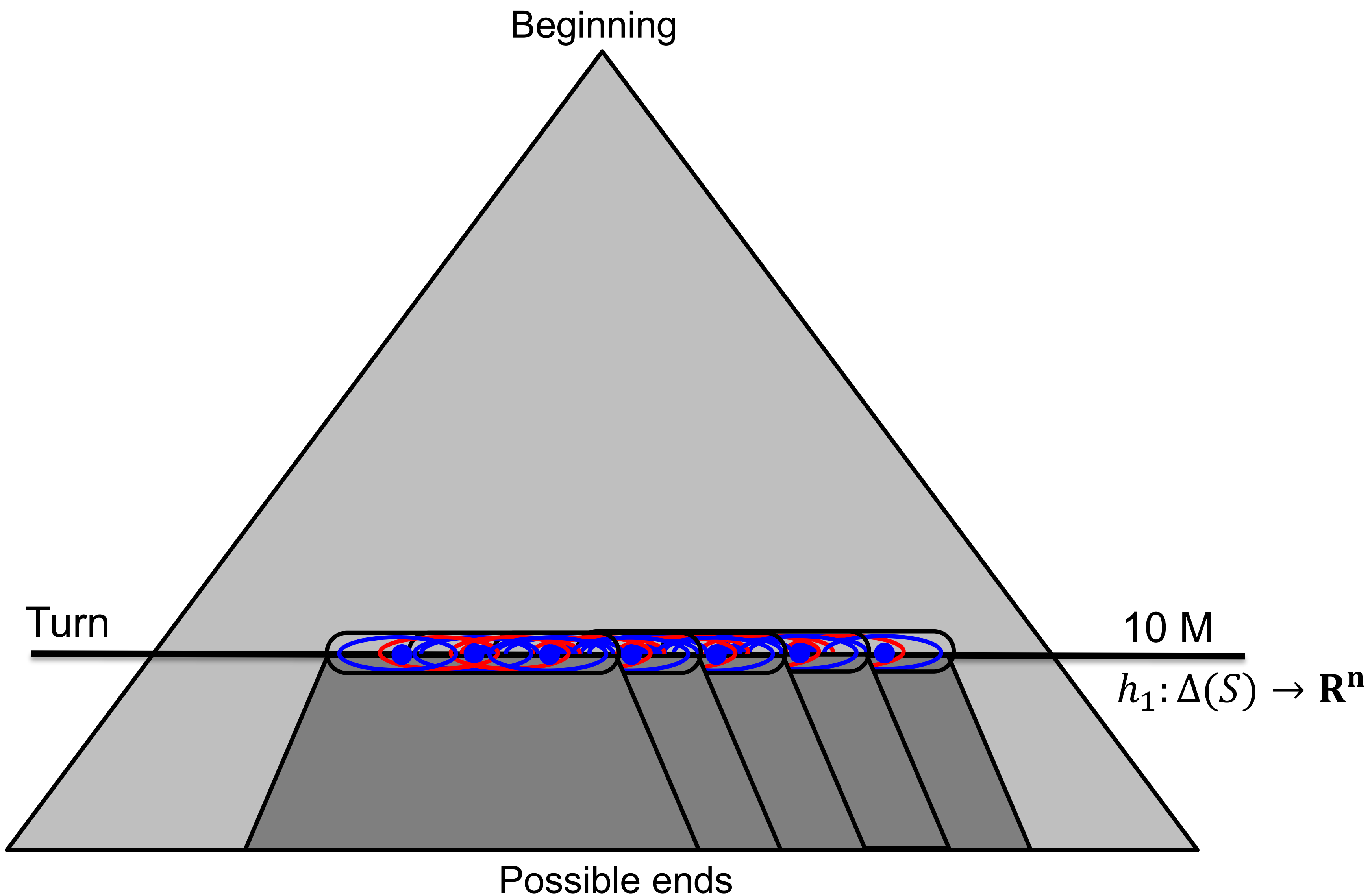
Depth limited look-ahead search



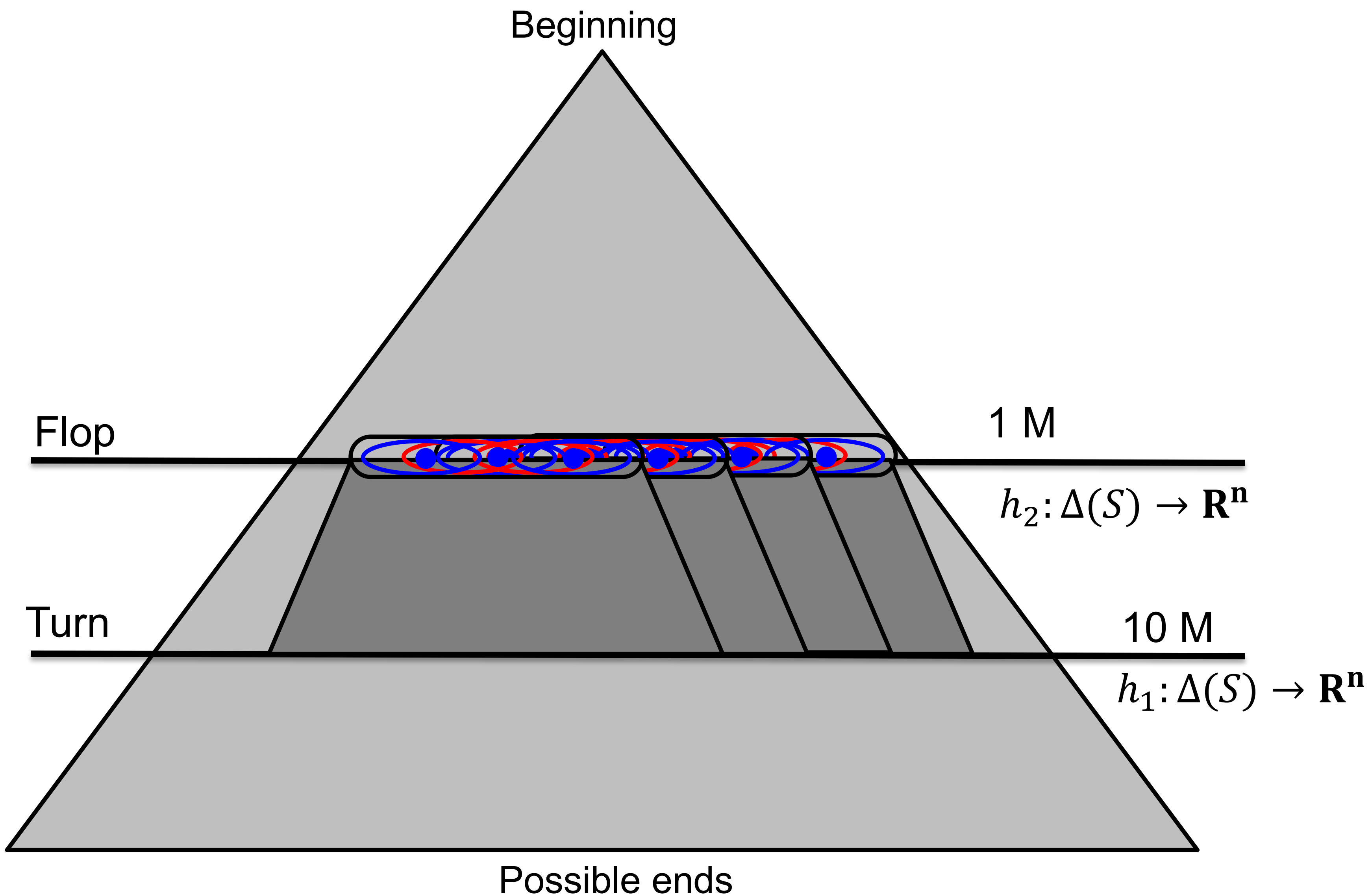
DeepStack: Neural Network



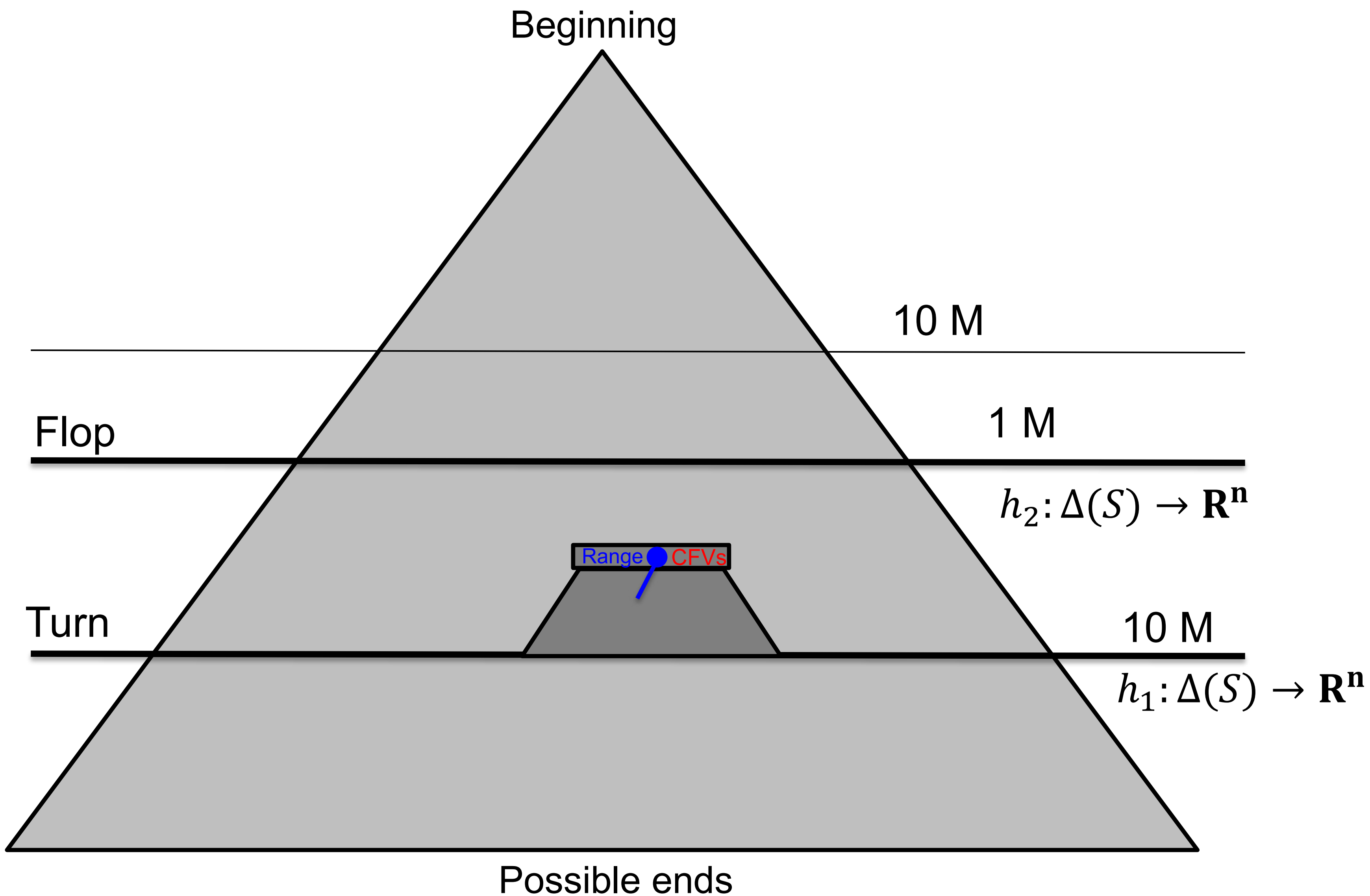
Where do we get training data?



Where do we get training data?



Where do we get training data?



DeepStack: Training

Turn Network (right after dealing turn card)

- 10M pseudo-random ranges, pots, random boards
- Solve by CFR^+ until the end of the game
- Extract CFVs for training, train Turn NN

Flop Network (right after dealing flop cards)

- 10M pseudo-random ranges, pots, random boards
- Solve by DeepStack (CFR-D) using the pre-trained Turn NN
- Extract CFVs for training, train Turn NN

Pre-flop Network

- 10M pseudo-random ranges, pots
- Enumerating 22100 possible flops and averaging

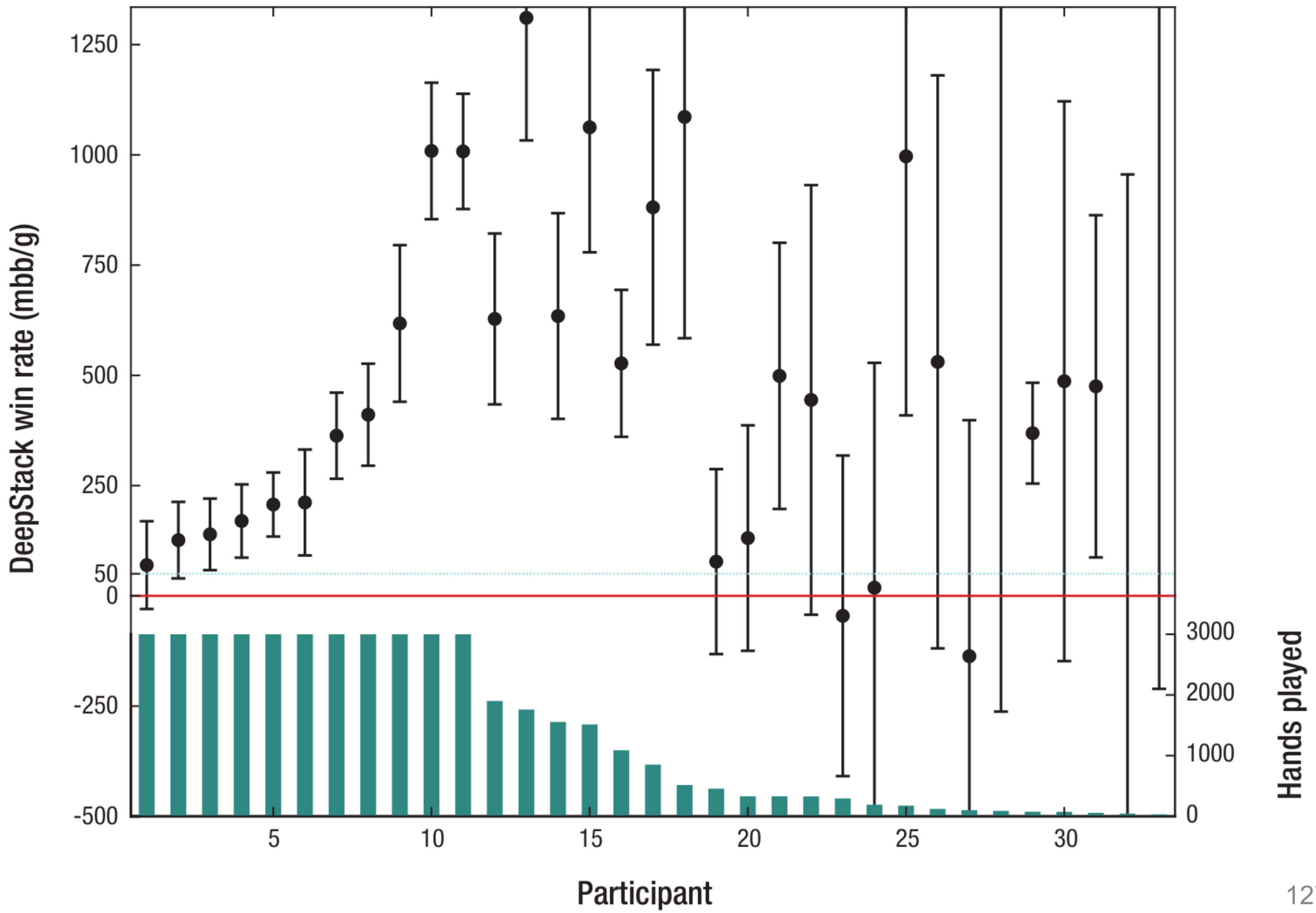
DeepStack: Convergence

Theorem: If the error of CFVs returned by the value function is less then ϵ and T iterations of resolving are used for each decision, than the exploitability of the player strategy is less than

$$k_1\epsilon + \frac{k_2}{\sqrt{T}}$$

where k_1, k_2 are game-specific constants.

DeepStack: Results



References

Burch, N., & Bowling, M. (2013). CFR-D: Solving Imperfect Information Games Using Decomposition. arXiv Preprint arXiv:1303.4441, 1–15. Retrieved from http://arxiv.org/abs/1303.4441

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis T., Waugh K., Johanson M., Bowling, M. (2017). DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. www.deepstack.ai

References

Asu Ozdaglar. 6.254 : Game Theory with Engineering Applications. Lecture 11: Learning in Games. March 11, 2010.

Brandt, Felix, Felix Fischer, and Paul Harrenstein. "On the rate of convergence of fictitious play." International Symposium on Algorithmic Game Theory. Springer Berlin Heidelberg, 2010.

T. Roughgarden, “Lecture Notes: Algorithmic Game Theory,” tech. rep., Stanford, 2013.

References

Blum, Avrim, and Yishay Mansour. "From external to internal regret." Journal of Machine Learning Research 8.Jun (2007): 1307-1324.

T. Roughgarden, “Lecture Notes: Algorithmic Game Theory,” tech. rep., Stanford, 2013.

Tammelin, Oskari, Neil Burch, Michael Johanson, and Michael Bowling. "Solving Heads-Up Limit Texas Hold'em." In Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

Bubeck, Sébastien, and Nicolo Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems." Foundations and Trends in Machine Learning 5.1 (2012): 1-122.