An Introduction to Counterfactual Regret Minimization

Todd W. Neller

Gettysburg College

History

- 1974: Robert Aumann introduces correlated equilibrium solution concept
- 2000: Hart and Mas-Colell introduced regret matching algorithm for computing correlated equilibria
- 2007: Zinkevich et al. introduced counterfactual regret minimization (CFR), dominant in computer poker competitions
- 2015: Bowling et al. solve heads-up limit hold’em poker (CFR+)
- CFR is the current state of the art for strategic sequential game play

Outline

- Regret
- Counterfactual Regret
- Counterfactual Regret Minimization

Rock-Paper-Scissors (RPS)

- Rock-Paper-Scissors (RPS)
  - 2 players, 3 possible simultaneous actions: rock (R), paper (P), scissors (S)
  - R, P, S beats S, R, P, respectively. Equal actions tie.
  - Win, tie, loss score +1, 0, -1, respectively

Regret

- Suppose you choose rock and your opponent chooses paper. Relative to your choice, how much do you regret not having chosen
  - paper?
  - scissors?
- Regret is the difference in utility between an action and your chosen action.
- Regrets: R→0 P→1 S→2

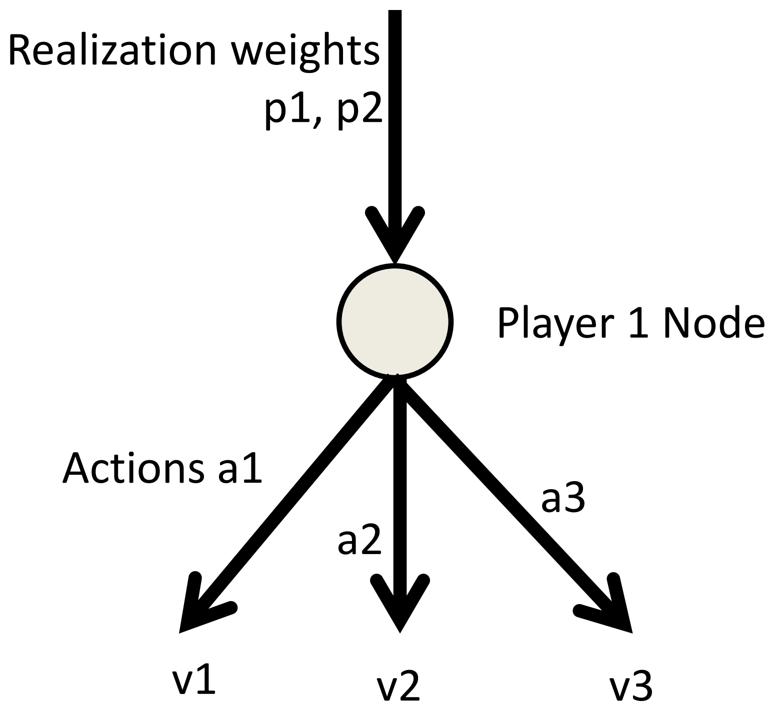
Regret Matching

- Choose an action with probability proportional to positive regrets.
- Regrets (0, 1, 2) normalized to probabilities: (0, 1/3, 2/3)
- Suppose we now choose S while our opponent chooses R.
  - Regrets: (1, 2, 0)
  - Cumulative regrets: (1, 3, 2)
  - Normalized cumulative regrets: (1/6, 3/6, 2/6)

Regret Minimization

- Regret Matching alone will not minimize regrets in the long run.
- However, the average strategy used over all iterations converges to a *correlated equilibrium.*
- In this example, average the strategies (1/3, 1/3, 1/3), (0, 1/3, 2/3), (1/6, 3/6, 2/6), etc.

Counterfactual Regret Example



- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.5	0.25	
Player 1 Node:			
Actions:	a1	a2	a3
Cumulative Regret	20	-10	30
Positive Regret	20	0	30
Strategy	0.4	0	0.6
To Cum. Strategy add	0.2	0	0.3
Recursive CFR Action Evals			
p1’ Realization Weights	0.2	0	0.3
p2’ Realization Weights	0.25	0.25	0.25
Action Values for Player 1	40	-8	20
Node Value for Player 1	28		
Action Regrets	12	-36	-8
Counterfactual Regrets	3	-9	-2
Old Cumulative Regret	20	-10	30
New Cumulative Regret	23	-19	28

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret			
Strategy			
To Cum. Strategy add			
Recursive CFR Action Evals			
p1' Realization Weights			
p2' Realization Weights			
Action Values for Player 2	-250	350	300
Node Value for Player 2			
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy			
To Cum. Strategy add			
Recursive CFR Action Evals			
p1’ Realization Weights			
p2’ Realization Weights			
Action Values for Player 2	-250	350	300
Node Value for Player 2			
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add			
Recursive CFR Action Evals			
p1' Realization Weights			
p2' Realization Weights			
Action Values for Player 2	-250	350	300
Node Value for Player 2			
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1’ Realization Weights			
p2’ Realization Weights			
Action Values for Player 2	-250	350	300
Node Value for Player 2			
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1’ Realization Weights	.2	.2	.2
p2’ Realization Weights			
Action Values for Player 2	-250	350	300
Node Value for Player 2			
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1’ Realization Weights	.2	.2	.2
p2’ Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2			
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1' Realization Weights	.2	.2	.2
p2' Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2	0(-250)	+.8(350)	+.2(300)
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1’ Realization Weights	.2	.2	.2
p2’ Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2	0	+280	+60
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1’ Realization Weights	.2	.2	.2
p2’ Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2	340		
Action Regrets			
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1' Realization Weights	.2	.2	.2
p2' Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2	340		
Action Regrets	-590	10	-40
Counterfactual Regrets			
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1’ Realization Weights	.2	.2	.2
p2’ Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2	340		
Action Regrets	-590	10	-40
Counterfactual Regrets	-118	2	-8
Old Cumulative Regret	-900	800	200
New Cumulative Regret			

- Input: realization weights
- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.
- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Counterfactual Regret Example

	p1	p2	
Realization Weights	0.2	0.5	
Player 2 Node:			
Actions:	a1	a2	a3
Cumulative Regret	-900	800	200
Positive Regret	0	800	200
Strategy	0	.8	.2
To Cum. Strategy add	0	.4	.1
Recursive CFR Action Evals			
p1' Realization Weights	.2	.2	.2
p2' Realization Weights	0	.4	.1
Action Values for Player 2	-250	350	300
Node Value for Player 2	340		
Action Regrets	-590	10	-40
Counterfactual Regrets	-118	2	-8
Old Cumulative Regret	-900	800	200
New Cumulative Regret	-1018	802	192

- Input: realization weights

- Compute node strategy from normalized positive cumulative regret.
- Update avg. output strategy weighted by player realization weight.
- Recursively evaluate strategy to compute action values and node value.

- Compute counterfactual regret.
- Update cumulative regret weighted by opponent realization weight.

Traversal of Game Tree

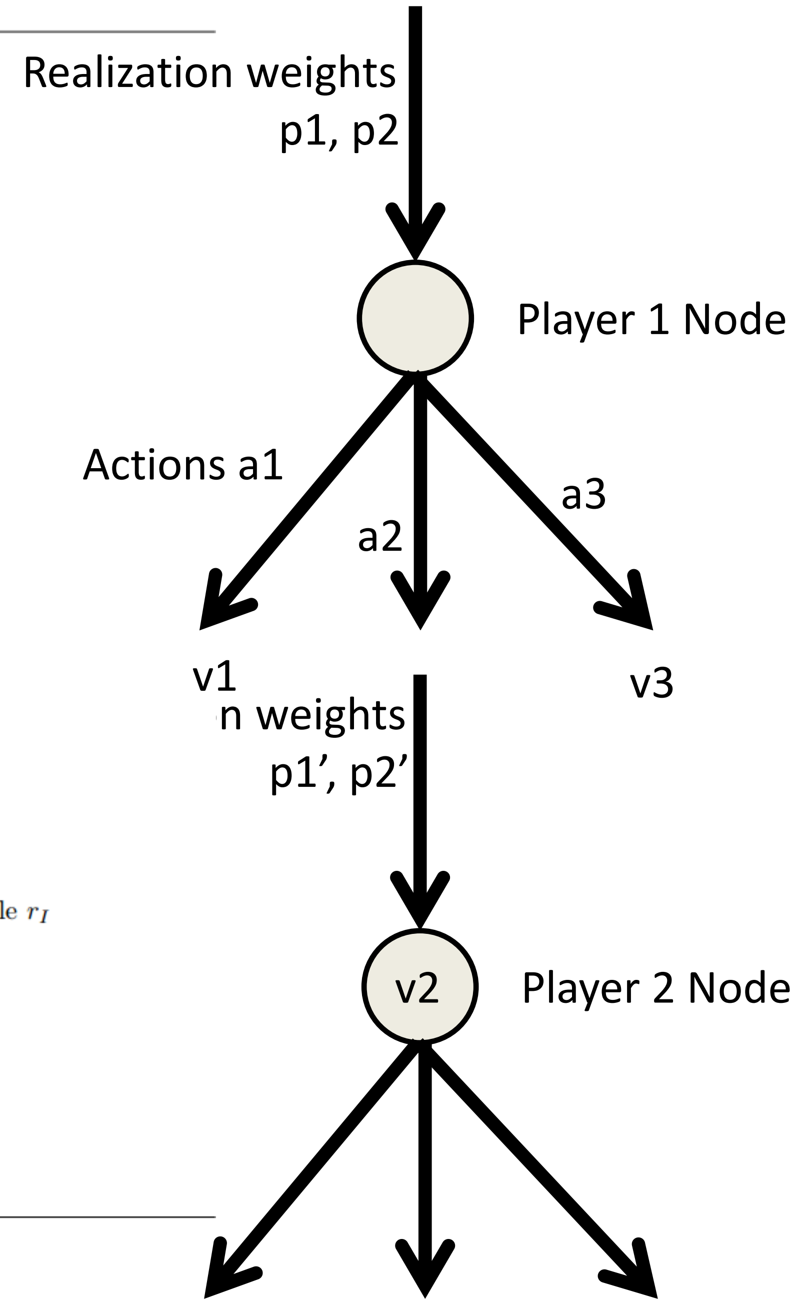
- This process begins at the root node and recursively traverses the tree.
  - At each chance node, one may average all possible chance events as part of play history, or simply chance sample 1 or more chance events.
  - If it's a player node, first determine the information set for that state (imperfect information), and then proceed as described to
    - Compute strategy from +ve cum. regret, update cum. strategy, and recursively evaluate all actions
    - When node values return from the recursive evaluations, compute and accumulate counterfactual regrets *as if the current player meant to play to this info. set.*

Algorithm 1 Counterfactual Regret Minimization (with chance sampling)

Initialize cumulative regret tables: $\forall I, r_I[a] \leftarrow 0$.
Initialize cumulative strategy tables: $\forall I, s_I[a] \leftarrow 0$.
Initialize initial profile: $\sigma^1(I,a) \leftarrow 1/|A(I)|$

function CFR(h, i, t, π_1, π_2):
if h is terminal **then**
    return $u_i(h)$
else if h is a chance node **then**
    Sample a single outcome $a \sim \sigma_c(h,a)$
    return CFR(ha, i, t, π_1, π_2)
end if
Let I be the information set containing h.
$v_\sigma \leftarrow 0$
$v_{\sigma_{I \to a}}[a] \leftarrow 0$ for all $a \in A(I)$
for $a \in A(I)$ **do**
    if $P(h) = 1$ **then**
        $v_{\sigma_{I \to a}}[a] \leftarrow$ CFR(ha, i, t, $\sigma^t(I,a) \cdot \pi_1$, π_2)
    else if $P(h) = 2$ **then**
        $v_{\sigma_{I \to a}}[a] \leftarrow$ CFR(ha, i, t, π_1, $\sigma^t(I,a) \cdot \pi_2$)
    end if
    $v_\sigma \leftarrow v_\sigma + \sigma^t(I,a) \cdot v_{\sigma_{I \to a}}[a]$
end for
if $P(h) = i$ **then**
    for $a \in A(I)$ **do**
        $r_I[a] \leftarrow r_I[a] + \pi_{-i} \cdot (v_{\sigma_{I \to a}}[a] - v_\sigma)$
        $s_I[a] \leftarrow s_I[a] + \pi_i \cdot \sigma^t(I,a)$
    end for
    $\sigma^{t+1}(I) \leftarrow$ regret-matching values computed using Equation 5 and regret table r_I
end if
return v_σ

function Solve():
for $t = \{1, 2, 3, \ldots, T\}$ **do**
    for $i \in \{1, 2\}$ **do**
        CFR($\emptyset$, i, t, 1, 1)
    end for
end for



Conclusion

- Regret minimization algorithms are an important part of the modern game theory landscape.
- Counterfactual regret minimization (CFR) is a complex algorithm, but can be understood one node at a time.
- CFR+ doesn’t allow negative regret sums.
- This brings you closer to understanding the state of the art.