Counterfactual Regret Minimization

Solving Imperfect-Information Games

Intelligent Agents: Computational Game Solving

Today – Counterfactual Regret Minimization

Goal: Solve 2-player zero-sum imperfect-information games

Recall:

- MWU/FPL: regret minimization in normal-form $\rightarrow$ converge to CCE
- Extensive-form: exponentially many pure strategies $\rightarrow$ need structure

Today's plan:

1. Counterfactual reasoning: π^{σ}_{-i} weighting
2. Regret decomposition over information sets
3. Vanilla CFR algorithm + convergence
4. Worked example: Kuhn poker
5. Practical variants (CFR+, sampling)

Key insight: Regret decomposes across infosets; local RM at each $I \rightarrow$ global Nash

Setup Reminder

Extensive-form game:

- Histories $h \in \mathcal{H}$, terminal Z, utilities $u_i(z)$
- Player i information sets $\mathcal{I}_i$; actions $A(I)$ at infoset I
- **Behavioral strategy** $\sigma_i : \mathcal{I}_i \to \Delta(A(I))$
- Perfect recall: σ_i induces unique distribution over outcomes

What we want:

- Two-player zero-sum: Nash equilibrium $\Leftrightarrow$ minimax solution
- **Exploitability**: $\mathrm{expl}(\sigma_i) = \max_{\sigma'_{-i}} u_{-i}(\sigma_i, \sigma'_{-i})$
- Goal: drive exploitability $\to 0$

Reach Probabilities

For profile $\sigma = (\sigma_1, \sigma_2)$ and history h:

$$\pi^\sigma(h) = \prod_{(I,a)\in h} \sigma_{\rho(I)}(I, a)$$

Decomposition:

$$\pi^\sigma(h) = \pi_c^\sigma(h) \cdot \pi_1^\sigma(h) \cdot \pi_2^\sigma(h)$$

- $\pi_c^\sigma(h)$: chance contribution
- $\pi_i^\sigma(h)$: player i's actions along h
- $\pi_{-i}^\sigma(h)$: opponent's actions

Extend to infosets: $\pi^\sigma(I) = \sum_{h\in I} \pi^\sigma(h)$

Why π_{-i} matters: Counterfactual = "if I reach I deterministically, what's my value?"

Counterfactual Values

Definition (Counterfactual value at infoset):

$$v_i^\sigma(I) = \sum_{h \in I} \frac{\pi^\sigma(h)}{\pi^\sigma(I)} \sum_{z \in Z, h \sqsubset z} \pi^\sigma(z \mid h)\, u_i(z)$$

Simpler form (using π_{-i} weighting):

$$v_i^\sigma(I) = \frac{1}{\pi^\sigma(I)} \sum_{h \in I, z \sqsupseteq h} \pi_{-i}^\sigma(h)\, \pi^\sigma(h \to z)\, u_i(z)$$

Action-conditional:

$$v_i^\sigma(I, a) = \text{counterfactual value if we take action } a \text{ at } I$$

$$v_i^\sigma(I) = \sum_{a \in A(I)} \sigma_i(I, a)\, v_i^\sigma(I, a)$$

Key: $v_i^\sigma(I, a)$ weights opponent reach π_{-i}, treats own reach as 1

Reach Probability Decomposition

Recall the reach probability factorization:

$$\pi^{\sigma}(h) = \pi^{\sigma}_{c}(h) \cdot \pi^{\sigma}_{i}(h) \cdot \pi^{\sigma}_{-i}(h)$$

and for an information set:

$$\pi^{\sigma}(I) = \sum_{h' \in I} \pi^{\sigma}(h') = \sum_{h' \in I} \pi^{\sigma}_{c}(h')\, \pi^{\sigma}_{i}(h')\, \pi^{\sigma}_{-i}(h')$$

Within one infoset:

$$\pi^{\sigma}_{c}(h) = \pi^{\sigma}_{c}(h'), \quad \pi^{\sigma}_{i}(h) = \pi^{\sigma}_{i}(h') \quad \forall h, h' \in I$$

(chance and player i made identical decisions to reach I).

Canceling Own Reach Within an Infoset

Normalized probability of a history $h \in I$:

$$\frac{\pi^\sigma(h)}{\pi^\sigma(I)} = \frac{\pi^\sigma_c(h)\pi^\sigma_i(h)\pi^\sigma_{-i}(h)}{\pi^\sigma_c(I)\pi^\sigma_i(I)\sum_{h' \in I} \pi^\sigma_{-i}(h')}$$

Since $\pi^\sigma_c(h) = \pi^\sigma_c(I)$ and $\pi^\sigma_i(h) = \pi^\sigma_i(I)$:

$$\frac{\pi^\sigma(h)}{\pi^\sigma(I)} = \frac{\pi^\sigma_{-i}(h)}{\sum_{h' \in I} \pi^\sigma_{-i}(h')}.$$

Substitute into value expression:

$$v^\sigma_i(I) = \sum_{h \in I} \frac{\pi^\sigma_{-i}(h)}{\sum_{h' \in I} \pi^\sigma_{-i}(h')} \sum_{z \succeq h} \pi^\sigma(z \mid h)\, u_i(z)$$

Own reach probabilities cancel—only opponents' reach remains.

Why This Matters for CFR

Key consequences of the counterfactual form:

- **Opponent reach weights the regret:**
  Regret at each I is scaled by $\pi^{\sigma}_{-i}(I)$.
- **Clean regret decomposition:**

$$R_i^T = \sum_{I \in \mathcal{I}_i} \sum_{t=1}^{T} \pi^t_{-i}(I) \left[v_i^t(I, a^*) - v_i^t(I) \right]$$

Regret can be minimized locally at each infoset.

Instantaneous & Cumulative Regret

At iteration t, profile σ^t, infoset $I \in \mathcal{I}_i$:

Instantaneous regret:

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

Cumulative regret:

$$R^T(I, a) = \sum_{t=1}^{T} r^t(I, a)$$

Positive regret:

$$R^{T,+}(I, a) = \max\{R^T(I, a), 0\}$$

Why positive regret? Regret-matching only boosts actions we *wish* we'd played more

Regret Matching (RM)

Update rule at infoset I for iteration $t+1$:

$$\sigma^{t+1}(I,a) = \begin{cases} \dfrac{R^{t,+}(I,a)}{\sum_{a' \in A(I)} R^{t,+}(I,a')} & \text{if denominator} > 0, \\ \text{uniform over } A(I) & \text{otherwise.} \end{cases}$$

Properties:

- Proven: $\mathrm{Regret}_I^T \leq \mathcal{O}(\sqrt{T|A(I)|})$ per infoset
- Simple, no learning rate tuning
- Local: each infoset independent

Average Strategy

Reach-weighted average:

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

Why weight by $\pi_i(I)$?

- Rarely-reached infosets matter less
- Matches the regret decomposition

Theorem (informal): If each player runs RM at every infoset, then

$$\text{Exploitability}(\bar{\sigma}^T) = \mathcal{O}\left(\frac{\sqrt{|I||A|}}{\sqrt{T}}\right)$$

for two-player zero-sum games.

CFR Algorithm (Vanilla)

Initialize $R^0(I, a) \leftarrow 0$ for all I, a
for $t = 1 \dots T$ **do**
    for each player i **do**
        Compute σ_i^t via RM from R^{t-1}
    end for
    Walk game tree: compute $v_i^{\sigma^t}(I, a)$ for all I, a
    for each infoset I **do**
        $r^t(I, a) \leftarrow v_i^{\sigma^t}(I, a) - v_i^{\sigma^t}(I)$
        $R^t(I, a) \leftarrow R^{t-1}(I, a) + r^t(I, a)$
    end for
    Accumulate $\pi_i^{\sigma^t}(I)\sigma^t(I, a)$ for averaging
end for
return $\bar{\sigma}^T$

Complexity per iteration: $\mathcal{O}(|Z|)$ (one tree traversal)

Why Does CFR Work? (Proof Sketch)

Step 1: Regret decomposition

Player i's global regret against any fixed strategy σ_i^*:

$$\text{Regret}_i^T(\sigma_i^*) = \sum_{I \in \mathcal{I}_i} \sum_{t=1}^{T} \pi^{\sigma^t_{-i}}(I) \left[v_i^{\sigma^t}(I, a_I^*) - v_i^{\sigma^t}(I) \right]$$

where a_I^* is action prescribed by σ_i^* at I.

Step 2: Local RM guarantees

Each infoset's cumulative regret $\leq \mathcal{O}(\sqrt{T})$

Step 3: Sum over infosets

$$\text{Regret}_i^T \leq \sum_{I \in \mathcal{I}_i} \mathcal{O}(\sqrt{T|A(I)|}) \leq \mathcal{O}(|\mathcal{I}_i|\sqrt{T|A|})$$

Step 4: Two-player zero-sum

Nash $\Leftrightarrow$ both players' average regret $\to 0 \Rightarrow$ exploitability $\mathcal{O}(1/\sqrt{T})$

Convergence Statement

Theorem (Zinkevich et al. 2007):
In 2-player zero-sum games, if both players run CFR:

$$\text{Exploitability}(\bar{\sigma}_1^T) \leq \frac{\Delta_{\max}\sqrt{|\mathcal{I}_1||A|}}{\sqrt{T}}, \quad \text{Exploitability}(\bar{\sigma}_2^T) \leq \frac{\Delta_{\max}\sqrt{|\mathcal{I}_2||A|}}{\sqrt{T}}$$

where $\Delta_{\max} = \max_{z,z'} |u_i(z) - u_i(z')|$.

Corollary: $\bar{\sigma}^T$ is an ϵ-Nash for $T = \mathcal{O}(1/\epsilon^2)$

Practical: Often converges much faster; modern variants (CFR+) empirically linear in T

CFR vs. Other Methods

Method	**Regret per iter**	**Tree traversals**	**Memory**	**Notes**
Vanilla CFR	$\mathcal{O}(1/\sqrt{T})$	1 full	$\mathcal{O}(\lvert\mathcal{I}\rvert\lvert A\rvert)$	Simple, determin
CFR+	$\sim \mathcal{O}(1/T)$ empirical	1 full	Same	Alternating upda gret floor
Monte Carlo CFR	Unbiased estimates	Samples	Same	Scales to huge ga
Policy-gradient (e.g., REINFORCE)	Variance issues	Full/sampled	Depends	Needs careful ba less theory

Why CFR dominates in poker: Exploits game structure, proven convergence, easy parallelization

Practical Variants – CFR+

CFR+ improvements (Tammelin 2014):

1. **Regret floor:** $R^{t,+}(I,a) \leftarrow \max\{R^t(I,a), 0\}$ stored (discard negatives)
2. **Alternating updates:** only update one player per iteration
3. **Linear weighting:** weight recent iterations more in $\bar{\sigma}^T$

Result: Empirical $\mathcal{O}(1/T)$ convergence in many games

One-line change:

$$R^t(I,a) \leftarrow \max\left\{R^{t-1}(I,a) + r^t(I,a),\ 0\right\}$$

Sampling Variants

Problem: Full tree traversal $\mathcal{O}(|Z|)$ intractable for large games

Solution: Sample outcomes, maintain unbiased CFV estimates

Types:

- **Outcome sampling:** sample one $z \sim \sigma^t$, update along that path
- **External sampling:** sample opponent & chance, enumerate own actions
- **Chance sampling:** sample chance, enumerate both players

Trade-off: Lower cost per iteration $\leftrightarrow$ higher variance $\rightarrow$ more iterations

Practical: External sampling very popular (e.g., Pluribus poker AI)

Abstraction & Large Games

Challenge: Real poker has $\sim 10^{161}$ states (Texas Hold'em)

Abstraction pipeline:

1. **Action abstraction:** reduce bet sizes (e.g., fold/call/pot/2×pot)
2. **Information abstraction:** cluster similar hands (e.g., EHS bucketing)
3. Solve abstract game with CFR
4. Map real states $\rightarrow$ abstract infosets at runtime

Caveat: Abstraction introduces approximation error; recent work on end-to-end neural CFR

References: Johanson et al. (2013), Brown & Sandholm (2019 – Pluribus)

CFR vs. Policy Gradient in EFGs

Aspect	**CFR**	**Policy Gradient (REINFORCE, PPO)**
Convergence	Proven $\mathcal{O}(1/\sqrt{T})$ for 2p-zs	No guarantees; can cycle or diverge
Variance	Deterministic (full CFV)	High variance (Monte Carlo returns)
Memory	$\mathcal{O}(\|\mathcal{I}\|\|A\|)$ explicit	Parameterized policy (can be smaller)
Scalability	Needs abstraction for huge games	Function approx scales, but needs tricks
Theory-practice gap	Tight	Large (needs baselines, entropy reg, ...)

When to use CFR: Medium-large structured games (poker), want guarantees
When to use PG: Gigantic state spaces, can tolerate sample inefficiency, or need online learning
Hybrid: Neural CFR (DeepStack, ReBeL) – CFR-style updates with neural value nets

Worked Example: Kuhn Poker – Game Description

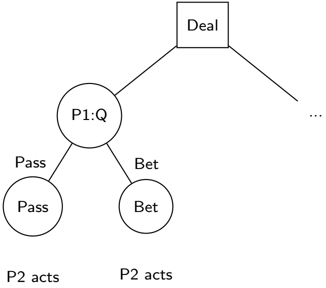
Kuhn Poker:

- Deck: {J, Q, K} (three cards)
- Two players, ante 1 chip each
- Each dealt one card (6 deal permutations, each prob 1/6)
- **Round 1 (P1):** Pass or Bet(1)
- **Round 2 (P2):** if P1 bet $\rightarrow$ Call(1) or Fold; if P1 passed $\rightarrow$ Pass or Bet(1)
- **Round 3 (P1, after P2 bet):** Call or Fold
- **Showdown:** higher card wins pot (or fold ends immediately)

Information sets (example for P1):

- `J-start`, `Q-start`, `K-start` (initial decision)
- `J-P2bet`, `Q-P2bet`, `K-P2bet` (after P1 passed, P2 bet)

Kuhn Poker – Game Tree (Simplified)



Focus: Infoset `Q-start` (P1 holds Queen at root)
Actions: Pass (p), Bet (b)

Kuhn Poker – Initial Strategy (Iteration 0)

Suppose uniform at all infosets:

- $\sigma_1^0(I, a) = 0.5$ for all $a \in \{\text{Pass}, \text{Bet}\}$
- $\sigma_2^0(I, a) = 0.5$ for all $a \in \{\text{Pass/Fold}, \text{Bet/Call}\}$

Goal: Compute counterfactual values at `Q-start`

Computing Counterfactual Values – `Q-start`

Infoset: P1 holds Q at root
Actions: Pass (p), Bet (b)

Calculation sketch (P2 strategies matter):
For action **Pass**:

- P2 can hold J or K (each prob 1/2 given P1 has Q)
- If P2-J: P2 passes w.p. 0.5 $\rightarrow$ P1 wins 2; P2 bets w.p. 0.5 $\rightarrow$ outcomes depend on P1's response
- If P2-K: similar branching...
- *(Full computation omitted; representative value:)*

$$v_1^{\sigma^0}(\texttt{Q-start}, \text{Pass}) \approx +0.25$$

For action **Bet**:

- P2-J folds w.p. 0.5 $\rightarrow$ P1 wins 2; calls w.p. 0.5 $\rightarrow$ P1 wins 3 total
- P2-K folds w.p. 0.5 $\rightarrow$ P1 wins 2; calls w.p. 0.5 $\rightarrow$ P1 loses 3

$$v_1^{\sigma^0}(\texttt{Q-start}, \text{Bet}) \approx +0.5$$

Instantaneous Regret at Q-start

Current value:

$$v_1^{\sigma^0}(\texttt{Q-start}) = 0.5 \times 0.25 + 0.5 \times 0.5 = 0.375$$

Instantaneous regret:

$$r^0(\texttt{Q-start}, \text{Pass}) = 0.25 - 0.375 = -0.125$$

$$r^0(\texttt{Q-start}, \text{Bet}) = 0.5 - 0.375 = +0.125$$

Interpretation: We regret not betting more with Q

Cumulative Positive Regret & Next Strategy

Cumulative positive regret (after iteration 0):

$$R^{0,+}(\texttt{Q-start}, \text{Pass}) = \max\{-0.125, 0\} = 0$$

$$R^{0,+}(\texttt{Q-start}, \text{Bet}) = \max\{+0.125, 0\} = 0.125$$

Regret Matching for iteration 1:

$$\sigma^1(\texttt{Q-start}, \text{Pass}) = \frac{0}{0 + 0.125} = 0$$

$$\sigma^1(\texttt{Q-start}, \text{Bet}) = \frac{0.125}{0 + 0.125} = 1.0$$

Interpretation: After one iteration, P1 with Q now always bets

Summary Table – One Iteration

Infoset	CFV		Regret		Next σ
	Pass	Bet	Pass	Bet	(Pass, Bet)
Q-start	0.25	0.50	-0.125	$+0.125$	(0.0, 1.0)

Key takeaway:

- CFR identifies that betting with Q is better than passing (under uniform opponent)
- Regret matching shifts all probability mass to the better action
- Over many iterations, strategies converge to Nash equilibrium

In-Class Exercise 1: Counterfactual vs. Expected Value

Setup: Consider infoset I for player 1. Current strategy profile σ has $\pi_1^\sigma(I) = 0.2$ and $\pi_2^\sigma(I) = 0.8$. The expected utility to P1 from I onward (given I is reached) is $+3$.

Questions:

1. What weight does CFR use when computing $v_1^\sigma(I)$?
2. If the *actual* reach probability $\pi^\sigma(I) = \pi_c \cdot \pi_1 \cdot \pi_2 = 0.05 \times 0.2 \times 0.8 = 0.008$, does this affect the counterfactual value?

Take 2 minutes to discuss with a neighbor.

Exercise 1 – Solution

Answers:

1. CFR weights by $\pi^{\sigma}_{-i}(I) = \pi^{\sigma}_{2}(I) = 0.8$ (ignores π_1).
2. No; counterfactual value normalizes by $\pi^{\sigma}(I)/\pi^{\sigma}_{i}(I)$, effectively treating $\pi_i = 1$. The value remains $+3$ (or weighted appropriately by π_{-i} and chance, but not by own actions).

Key insight: Counterfactual reasoning asks "what if I deterministically reached this infoset?" – removes the effect of our own past actions.

In-Class Exercise 2: Regret Matching Calculation

Setup: Infoset I with 3 actions $\{a_1, a_2, a_3\}$. After 5 iterations, cumulative regrets are:

- $R^5(I, a_1) = +2.0$
- $R^5(I, a_2) = -0.5$
- $R^5(I, a_3) = +1.5$

Question: What is $\sigma^6(I, \cdot)$ using regret matching?

Take 2 minutes to compute.

Exercise 2 – Solution

Solution:

Positive regrets:

- $R^{5,+}(I, a_1) = 2.0$
- $R^{5,+}(I, a_2) = 0$
- $R^{5,+}(I, a_3) = 1.5$

Sum $= 2.0 + 0 + 1.5 = 3.5$

$$\sigma^6(I, a_1) = \frac{2.0}{3.5} = \frac{4}{7} \approx 0.571$$

$$\sigma^6(I, a_2) = 0$$

$$\sigma^6(I, a_3) = \frac{1.5}{3.5} = \frac{3}{7} \approx 0.429$$

In-Class Exercise 3: Why π_{-i} and not π_i?

Thought experiment: Suppose we incorrectly computed counterfactual values weighting by $\pi_i^\sigma(I)$ instead of $\pi_{-i}^\sigma(I)$.

Question: What goes wrong with the regret decomposition?

Discuss for 2 minutes.

Exercise 3 – Solution

Answer:
The global regret decomposes as:

$$\text{Regret}(\sigma_i^*) = \sum_I \sum_t \pi_{-i}^{\sigma^t}(I)\Big[v_i(I, a^*) - v_i(I)\Big]$$

The opponent's strategy π_{-i} weights how often *they* lead us to I. If we used π_i, we'd:

- Double-count our own choices (they're already in the strategy σ_i)
- Break the telescoping property linking infoset regrets to global regret

Key intuition: Counterfactual = "my value if I force myself to reach I, given opponent's play."

Summary

Today we covered:

1. **Counterfactual values:** weight by π_{-i}, not π_i
2. **Regret decomposition:** global regret = sum of local regrets
3. **Vanilla CFR:** regret matching at each infoset $\rightarrow \mathcal{O}(1/\sqrt{T})$ exploitability
4. **Kuhn poker example:** one iteration of CFR in action
5. **Practical variants:** CFR+, sampling, abstraction

Key takeaway: CFR exploits extensive-form structure to scale regret minimization to large imperfect-information games.

Next lecture: Extensions (Monte Carlo CFR, Neural CFR)

References & Further Reading

- **Zinkevich et al. (2007):** "Regret Minimization in Games with Incomplete Information" (original CFR paper)
- **Tammelin (2014):** "Solving Large Imperfect Information Games Using CFR+" (CFR+ variant)
- **Lanctot et al. (2009):** "Monte Carlo Sampling for Regret Minimization in Extensive Games" (sampling variants)
- **Johanson et al. (2013):** "Finding Optimal Abstract Strategies in Extensive-Form Games" (abstraction)
- **Brown & Sandholm (2019):** "Superhuman AI for multiplayer poker" (Pluribus, Science paper)
- **Moravčík et al. (2017):** "DeepStack: Expert-level artificial intelligence in heads-up no-limit poker" (neural CFR)

Open-source implementations:

- OpenSpiel (DeepMind): `github.com/deepmind/open_spiel`
- PokerRL: `github.com/TinkeringCode/PokerRL`