Information Abstraction in Poker

Features, Buckets, and Potential-Aware Methods

Intelligent Agents: Computational Game Solving

October 30, 2025

Recap: Why Abstraction?

Last lecture:

- Real poker games (Hold'em) have 10^{14}–10^{161} states
- Can't solve exactly with CFR $\rightarrow$ need abstraction
- Two main types: **information** abstraction (merge infosets) and **action** abstraction (discretize actions)

Today: Deep dive into information abstraction

- How do we decide which infosets to merge?
- What features capture strategic similarity?
- What algorithms cluster hands into buckets?
- How do we evaluate abstraction quality?

Running examples: Texas Hold'em features, Leduc Poker bucketing

The Information Abstraction Problem

Goal: Reduce information set count while preserving strategic distinctions

Challenge: Which infosets are "similar enough" to merge?

Example (Hold'em preflop):

- $\binom{52}{2} = 1,326$ possible 2-card hands
- Are A♠A♡ and A♣A♢ similar? **Yes** (same rank, both aces)
- Are A♠K♠ and A♡K♢ similar? **Mostly** (same ranks, but suited vs. offsuit)
- Are K♠K♡ and Q♠Q♡ similar? **Somewhat** (both premium pairs, different equity)
- Are A♠K♠ and 7♣2♢ similar? **No!** (very different strength)

Our approach:

1. Define **features** that capture strategic value
2. **Cluster** hands based on feature similarity
3. Map each hand to a **bucket** (abstract infoset)

Feature 1: Effective Hand Strength (EHS)

Definition: Probability your hand wins at showdown (if no more betting)

$$\text{EHS}(h, \text{board}) = \Pr[\text{win} \mid h, \text{board}]$$

Computation:

- Enumerate all possible opponent hands
- For each opponent hand, determine winner
- Average over uniform opponent distribution

Example (Hold'em flop):

- You hold: A♠K♠
- Board: K♡7♢2♣
- You have top pair, top kicker (pair of Kings with Ace kicker)
- EHS ≈ 0.78 (you beat most hands, lose to sets and two-pair)

Properties:

- EHS $\in [0, 1]$; higher is stronger
- Easy to compute (enumerate $\sim\binom{50}{2}$ opponent hands)
- **Static:** doesn't account for future cards

Feature 2: Hand Potential

Problem with EHS: Ignores draws and future improvement

Hand Potential: Expected change in hand strength on future cards

Two components:
1. Positive Potential (PPot):

$$\text{PPot} = \Pr[\text{behind now, ahead at showdown}]$$

Measures: "Can I improve to win?"
Example: Flush draw (4 spades)

- Currently losing to opponent's pair
- But PPot ≈ 0.35 (9 outs / 47 cards $\times$ 2 streets)

2. Negative Potential (NPot):

$$\text{NPot} = \Pr[\text{ahead now, behind at showdown}]$$

Measures: "Can opponent catch up?"
Example: Top pair (K♠Q♡ on K♣7♠2♢ flop)

- Currently ahead, but NPot > 0 if opponent has flush or straight draw

Feature 3: Expected Hand Strength (E[HS])

Idea: Average hand strength over all possible future cards

$$\text{E[HS]} = \sum_{\text{future cards}} \Pr[\text{future}] \cdot \text{EHS}(\text{hand}, \text{board} + \text{future})$$

Incorporates potential:

$$\text{E[HS]} \approx \text{EHS} + \text{PPot} - \text{NPot}$$

(Approximate; exact formula involves correlations)

Example:

- Flush draw on flop: EHS ≈ 0.15 (losing now), E[HS] ≈ 0.35 (35% to make flush)
- Top pair: EHS ≈ 0.78, E[HS] ≈ 0.75 (slightly vulnerable)

Why useful for bucketing?

- Captures *expected* value, not just current value
- Groups draws with made hands of similar expected strength

Feature 4: Hand Strength Variance ($E[HS^2]$)

Definition: Variance of hand strength across future runouts

$$\mathrm{Var}[HS] = E[HS^2] - (E[HS])^2$$

Intuition:

- **High variance:** Volatile hand (big draws, or vulnerable made hand)
- **Low variance:** Stable hand (locked-in strength)

Examples:

Hand	E[HS]	Var[HS]	Interpretation
Nut flush draw	0.35	High	Volatile (0% or 100%)
Top pair	0.75	Medium	Somewhat stable
Set (trips)	0.92	Low	Very stable
Made flush	0.95	Very low	Locked in

Strategic relevance:

- High variance → aggressive play (semi-bluff, raise for fold equity)
- Low variance → value betting (stable strength, extract value)

Feature 5: Draw Types and Outs

Explicit draw classification:

- **Flush draw:** 4 cards of same suit (9 outs)
- **Open-ended straight draw:** 4 in sequence, completes either end (8 outs)
  - Example: 9-8-7-6 $\rightarrow$ need 10 or 5
- **Gutshot (inside straight draw):** Missing interior card (4 outs)
  - Example: J-10-8-7 $\rightarrow$ need 9
- **Backdoor flush:** 2 cards of same suit (need both turn and river)
- **Combo draw:** Flush + straight draw (up to 15 outs)

Outs counting:

$$\text{Equity} \approx \frac{\text{outs}}{47} \times 2 \quad \text{(``Rule of 4 and 2'')}$$

Why useful?

- Explicit features for common strategic patterns
- Can weight by draw strength (flush draw stronger than gutshot)

Feature 6: Blockers (Advanced)

Definition: Cards you hold that reduce opponent's possible hand combinations

Example 1: Ace blocker

- You hold: A♠K♠
- Board: K♡7♢2♣
- You block top pair with better kicker (opponent can't have A-K)
- Reduces opponent's strong hands $\rightarrow$ increases your bluffing fold equity

Example 2: Flush blocker

- Board: A♠K♠Q♠7♢2♣
- You hold: 9♠8♣
- Your 9♠ blocks opponent flush combinations
- Makes opponent less likely to have flush $\rightarrow$ good bluff candidate

Use in abstraction:

- Blocker effects matter for bluffing and thin value betting
- Can be captured via opponent hand distribution analysis
- Usually omitted in coarse abstractions (second-order effect)

Public vs. Private Information

Key distinction in poker:

- **Public cards:** Board (flop, turn, river) — everyone sees
- **Private cards:** Hole cards — only you see

Implication for abstraction:
Naive approach: Bucket all hands globally

- Problem: Same hand has different value on different boards
- Example: 7♠7♡ is strong on 7♢2♣K♡ (set), weak on A♠K♠Q♠ (low pair)

Better approach: Public Belief States (PBS)

1. Partition by **public cards** (e.g., flop texture: paired, monotone, rainbow)
2. Within each PBS, bucket **private cards** based on features

Result: Hand buckets adapt to board context
Example: Flush draw bucketed as "strong draw" on flush-heavy board, "weak draw" on paired board

Bucketing Algorithm 1: K-Means Clustering

Goal: Group hands into k buckets based on feature similarity

Setup:

1. Represent each hand as feature vector: $\mathbf{x}_h = (\text{EHS}, \text{PPot}, \text{NPot}, \ldots)$
2. Choose number of buckets k (hyperparameter)

K-means algorithm:

1. Initialize k cluster centroids randomly
2. Repeat until convergence:
   1. **Assign:** Each hand h to nearest centroid

      $$\text{bucket}(h) = \arg\min_{j \in [k]} \|\mathbf{x}_h - \mathbf{c}_j\|_2$$

   2. **Update:** Recompute centroids as mean of assigned hands

      $$\mathbf{c}_j = \frac{1}{|C_j|} \sum_{h \in C_j} \mathbf{x}_h$$

Properties:

- Simple, fast ($\mathcal{O}(nk \cdot \text{iters})$ for n hands)
- Requires feature normalization (scale to [0,1])
- Euclidean distance may not match strategic similarity

Bucketing Algorithm 2: Quantile Bucketing

Idea: Sort hands by a single feature, divide into equal-sized buckets

Algorithm:

1. Choose a primary feature (e.g., EHS or E[HS])
2. Sort all hands by that feature value
3. Divide into k buckets of equal size (n/k hands per bucket)

Example (10 buckets on river):

- Bucket 1: Top 10% hands (EHS $\in [0.9, 1.0]$)
- Bucket 2: Next 10% (EHS $\in [0.8, 0.9]$)
- ... Bucket 10: Bottom 10% (EHS $\in [0.0, 0.1]$)

Pros:

- Very simple, no clustering algorithm needed
- Balanced bucket sizes; natural for river (pure hand strength)

Cons:

- Only uses one feature (ignores potential, draws)
- May merge strategically different hands near boundaries

Bucketing Algorithm 3: Earth Mover's Distance (EMD)

Motivation: Strategic similarity = similar outcome distributions

Approach:

1. For each hand h, compute distribution over final hand strengths:

$$P_h = \{\Pr[\text{EHS} = x \mid h, \text{future cards}]\}_{x \in [0,1]}$$

2. Define distance between hands as Earth Mover's Distance:

$$\text{EMD}(h_1, h_2) = \min_{\text{flow}} \sum_{i,j} \text{flow}_{ij} \cdot d(x_i, x_j)$$

   (Min "work" to transform distribution P_{h_1} into P_{h_2})

3. Cluster hands using EMD as distance metric

Intuition:

- Two hands similar if they have similar equity distributions
- Example: Flush and straight draws both bimodal (0% or 100%)

Pros: Game-theoretically motivated, captures uncertainty
Cons: Expensive ($\mathcal{O}(n^3)$ for EMD, $\mathcal{O}(n^2)$ for clustering)

Potential-Aware Abstraction

Problem: Early-street buckets should account for future cards

Example (Hold'em flop):

- Hand 1: Top pair (K♠Q♡ on K♣7♢2♠) — EHS ≈ 0.78
- Hand 2: Flush draw (A♠9♠ on K♣7♠2♠) — EHS ≈ 0.35
- Different current strength, but similar *potential*

Potential-aware features:

- E[HS], PPot, NPot (already discussed)
- Histogram of future hand strengths
- Cluster using EMD on histograms

Hierarchical bucketing:

- **Preflop:** Very coarse (5–10 buckets)
- **Flop:** Medium (50–100 buckets, include draw types)
- **Turn:** Fine (200 buckets, specific draws)
- **River:** Very fine (1000 buckets, pure hand strength)

Rationale: More info revealed later → need finer distinctions

Public Belief States (PBS)

Full framework:

1. **Partition by public cards:**
   - Group boards by texture (monotone, rainbow, paired, connected)
   - Or: use all possible boards as separate PBS (expensive)
   - Or: cluster boards by features
2. **Within each PBS, bucket private hands:**
   - Compute features (EHS, PPot, etc.) conditional on that PBS
   - Run k-means or quantile bucketing

Example (simplified):

- PBS 1: "Monotone flop" (3 cards same suit) $\rightarrow$ 20 private buckets
- PBS 2: "Paired flop" (e.g., K-K-7) $\rightarrow$ 15 private buckets
- PBS 3: "Rainbow flop" (3 different suits) $\rightarrow$ 25 private buckets

Result: Total infosets = (# PBS) $\times$ (avg buckets/PBS) $\times$ (betting seqs)

Example: Leduc Poker Abstraction

Leduc Poker recap:

- Deck: 6 cards (J, J, Q, Q, K, K)
- 2 players, 2 betting rounds (preflop, flop)
- 1 community card on flop; Goal: Pair or better

Full game size:

- Preflop: 6 hole $\times$ 5 opponent cards $=$ 30 deals
- Flop: 30 deals $\times$ 4 community cards $=$ 120 situations
- Betting sequences: $\sim$100s of histories
- Total infosets: $\sim$10,000

Abstraction design — Preflop (4 buckets):

1. Pair of Jacks (JJ)
2. Pair of Queens (QQ)
3. Pair of Kings (KK)
4. Unpaired (J, Q, or K)

Leduc Abstraction (continued)

Flop (8 buckets):

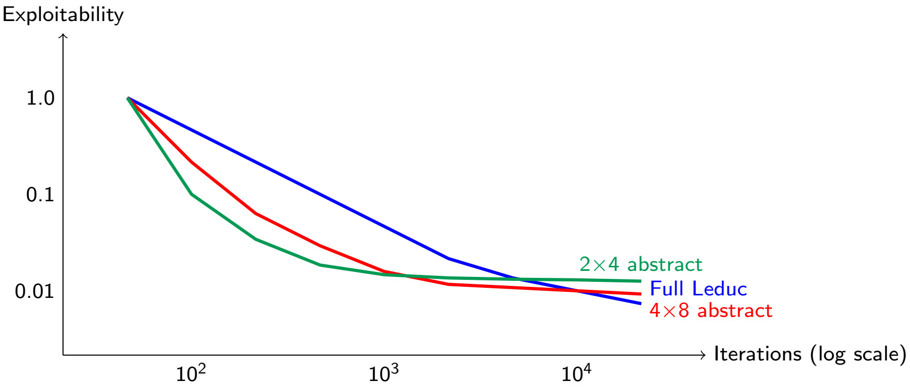
1. Three of a kind (trips) — very strong
2. Pair of Kings with higher hole card — strong
3. Pair of Queens with higher hole card — medium-strong
4. Pair of Jacks with higher hole card — medium
5. Pair with lower hole card — medium-weak
6. High card King (no pair) — weak
7. High card Queen (no pair) — weak
8. High card Jack (no pair) — very weak

Abstract game size:

- Flop: 4 preflop $\times$ 4 flop cards $\times$ 8 buckets $\approx$ 128 infosets
- With betting: $\sim$500 abstract infosets (vs. 10,000 full)
- **Reduction: 20$\times$**

CFR Convergence Comparison (Leduc)

Experiment: Solve Leduc with different abstractions, measure exploitability



Observations:

- Coarser abstraction converges faster (fewer infosets)
- But final exploitability higher (approximation error)
- 4×8: good balance (90% performance, 5× speedup)

Head-to-Head Evaluation

Alternative metric: Pit strategies against each other, measure win rate

Experiment: Play 10,000 hands between strategies

Matchup	Win Rate	Interpretation
Full vs. Full	0.0 mbb/hand	Nash (zero expected value)
Full vs. 4×8 abstract	+0.5 mbb/hand	Abstract slightly exploitable
Full vs. 2×4 abstract	+2.1 mbb/hand	Coarse abstraction weak
4×8 vs. 2×4	+1.8 mbb/hand	Finer dominates

Notes:

- mbb/hand = milli-big-blinds per hand (standard poker metric)
- Variance is high; need many hands for statistical significance
- 4×8 performs well (only 0.5 mbb/hand loss vs. full)

Evaluation Pitfall 1: Strategy Fusion

Problem: Merging dissimilar hands forces them to play identically

Example:

- Bucket "flush draw" and "weak pair" into "medium strength"
- Optimal: flush draw semi-bluffs (bet/raise), weak pair check/calls
- After merging: compromise (mixed bet/check)
- Result: Neither hand plays optimally

Mitigation:

- Use finer-grained features (separate draw type feature)
- Increase bucket count
- Accept some error (unavoidable in abstraction)

Theoretical note:

- Error bounded by infoset dissimilarity (Waugh et al., 2009)
- If merged infosets have similar counterfactual values, error small
- Key: minimize within-bucket variance

Evaluation Pitfall 2: Abstraction Mismatch

Problem: Training abstraction differs from deployment/opponent

Scenario 1: Different bucket counts

- You train with 50 buckets, opponent uses 100
- Opponent can exploit your coarser distinctions

Scenario 2: Different features

- You bucket by EHS, opponent by E[HS] + potential
- Strategies "talk past each other"
- May not reach Nash in combined abstract game

Mitigation:

- **Cross-abstraction testing:** Train with A, test vs. B
- Use real-time re-solving (next lecture) to adapt
- Design robust abstractions (not overfitted)

Evaluation Pitfall 3: Bucket Leakage

Problem: Opponent can infer your bucket from betting patterns

Example:

- Your abstraction: "strong" always bets 2×pot, "weak" checks
- Opponent observes: you bet 2×pot
- Opponent infers: you have "strong" bucket → folds more
- Your abstraction exploitable (too deterministic)

Why it happens:

- Abstraction reduces strategy space
- Patterns emerge: buckets → actions
- Opponent can reverse-engineer bucketing

Mitigation:

- Use mixed strategies (even within buckets)
- Add action diversity (multiple bet sizes per bucket)
- Real-time re-solving (new strategy each decision)
- Adversarial evaluation (vs. opponent trained to exploit you)

Evaluation Pitfall 4: Cross-Street Generalization

Problem: Buckets on one street may not align with next

Example (Hold'em):

- Flop: Bucket "strong draw" includes flush draw (9 outs)
- Turn: Draw hits → "made flush" bucket
- Or: Draw misses → "weak draw" bucket (3 outs)
- Bucket identity changes dramatically

Issue:

- CFR learns "play bucket X aggressively on flop"
- But X transitions to different buckets on turn
- Abstraction may not capture transitions well

Solution: Potential-aware abstraction

- Incorporate future card distributions into flop buckets
- Use E[HS] and histograms (EMD clustering)
- Result: Flop buckets "know" their likely turn buckets

Practical Design Considerations

1. **Bucket count trade-off:**
   - More buckets → better approximation, slower solve
   - Rule of thumb: 10–50 early streets, 100–1000 river
2. **Feature engineering:**
   - Start with EHS, E[HS], PPot, NPot
   - Add domain knowledge (flush/straight indicators)
   - Normalize features to [0,1] for clustering
3. **Algorithm choice:**
   - K-means: fast, high-dimensional features
   - Quantile: simple baseline, works well on river
   - EMD: expensive but principled; use for early streets
4. **Evaluation pipeline:**
   - Compute exploitability in abstract game (upper bound)
   - Play head-to-head vs. baselines (practical performance)
   - A/B test: vary bucket counts, compare
5. **Iteration:** Design → solve → evaluate → refine

Summary: Information Abstraction

Key ideas:

1. **Features:** EHS, E[HS], PPot, NPot, variance, draws, blockers
2. **Bucketing:** K-means, quantile, EMD; potential-aware
3. **PBS:** Partition by public cards, bucket private within
4. **Hierarchical:** Coarse early, fine late

Trade-offs:

- Coarser: faster solve, weaker strategy
- Finer: slower solve, better strategy
- Balance depends on budget and target performance

Evaluation challenges:

- Strategy fusion, mismatch, leakage, cross-street issues
- Must test empirically (exploitability + head-to-head)

Next: Action abstraction, real-time re-solving, endgame solving

References

- **Johanson et al. (2013):** "Finding Optimal Abstract Strategies in Extensive-Form Games" (AAAI) — Comprehensive framework
- **Ganzfried & Sandholm (2014):** "Potential-Aware Imperfect-Recall Abstraction with EMD" (AAAI) — EMD bucketing
- **Waugh et al. (2009):** "A Practical Use of Imperfect Recall" (AAMAS) — Theory, strategy fusion bounds
- **Shi & Littman (2001):** "Abstraction Methods for Game Theoretic Poker" (CG) — Early work on EHS and potential