MCTS Variant Performance Prediction

Feature Engineering & Selection Analysis

林芃芃 | 郑炼鑫

Introduction

Goal

Predict the relative performance of MCTS variants (Agent 1 vs Agent 2) based on algorithm configuration and game features.

Competition: Game-Playing Strength of MCTS Variants

Methodology

"Structure Analysis —> Statistical Aggregation —> Interaction Modeling"

1. **Parse Components**: Extract Selection, Exploration, Playout, Score Bounds.
2. **Aggregate Stats**: Use Out-of-Fold (OOF) target encoding for robust estimation.
3. **Model Interactions**: Explore how Game Stochasticity regulates Agent differences.

Feature 1: ExplorationGap

Exploration Constant Difference

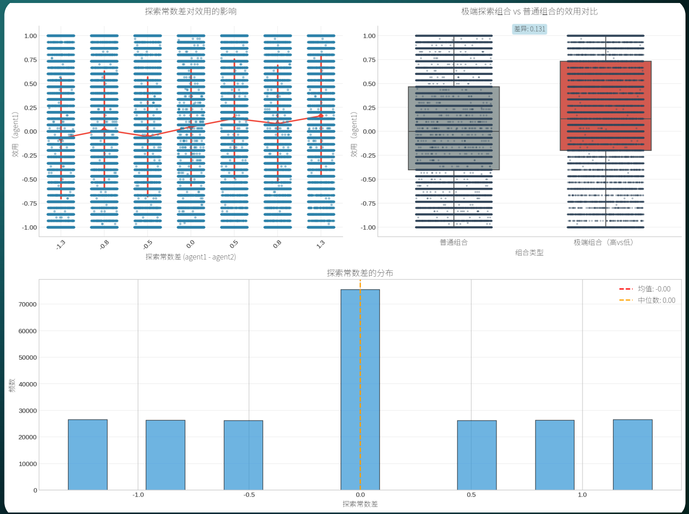
Definition

Measure the relative "exploration" tendency between two agents.

$$\text{ExplorationGap} = \epsilon_1 - \epsilon_2$$

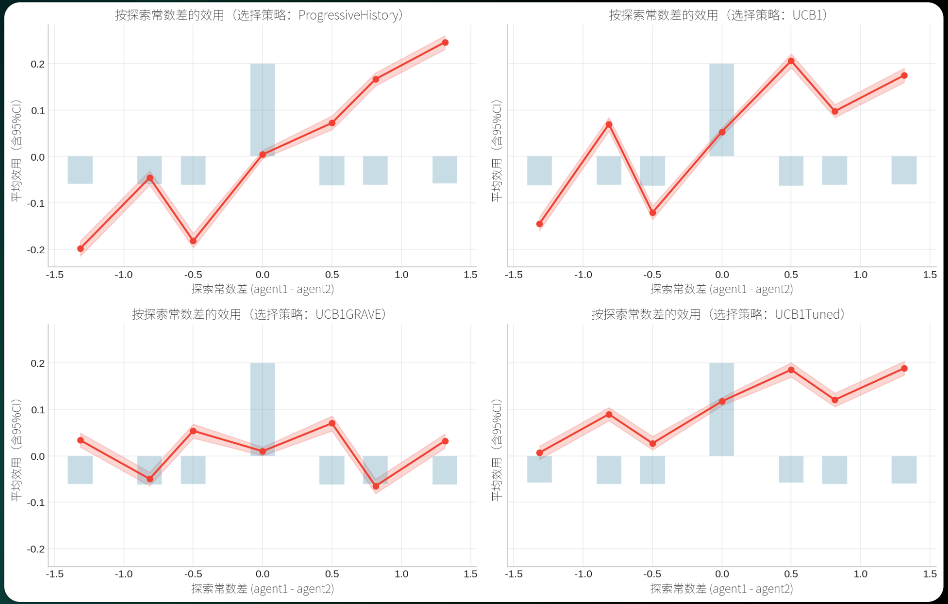
Key Insight

- **Positive Correlation**: $\rho = 0.107$ ($p \approx 0$)
- When Agent 1 is more "explorative", it generally achieves higher utility.
- **Extreme Case**: High vs Low exploration yields a stable mean utility advantage of **+0.131**.



Feature 1: ExplorationGap

Exploration Constant Difference



Robust across different Selection strategies (ProgressiveHistory, UCB1).

F1: Implementation

```
# 1. Construct Feature
df['ExplorationGap'] = (
    df['exploration1'] - df['exploration2']
)

# 2. Statistical Validation
rho, p = spearmanr(
    df['ExplorationGap'],
    df['utility_agent1']
)

# 3. Bootstrap Analysis
# (See Appendix for full code)
```

Findings

- **Robustness**: Validated via Stratified Bootstrap ($N = 2000$).
- **Conclusion**: A directionally clear, interpretable, and robust main effect feature.

> Higher exploration constant generally benefits Agent 1 in this dataset.

Feature 2: Playout Strategy (Part 1)

Initial Hypothesis

Match vs Mismatch

- **Idea**: Do they use the same playout strategy?
- **Result**: Failed
  - $U \approx 5.94e9, p \approx 0.706$
  - "Same" vs "Different" has no significant impact.

Refinement

Intensity Gap Treat as **Ordered Pairs** (Agent 1 vs Agent 2):

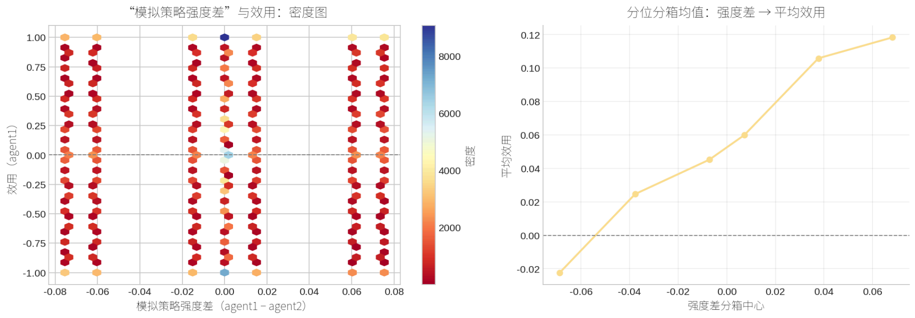
- **MAST vs NST**: Highest Utility **(+0.118)**
- **NST vs MAST**: Lowest Utility **(-0.030)**

Feature 2: Playout Strategy (Part 2)

PlayoutStrengthGap

$$\Delta S_{playout} = S(P_1) - S(P_2)$$

- **Correlation**: $\rho = 0.073$ ($p < 1e^{-200}$)
- A clear, monotonically increasing trend in utility.
- Although the effect is smaller than Selection Strategy, it is statistically significant.



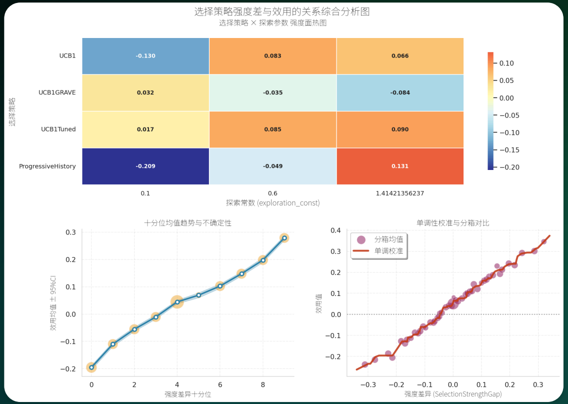
Feature 3: SelectionStrengthGap

The Strongest Single Feature

Concept

Compute "Strength Score" for each (Selection + Exploration) configuration using **OOF Smoothing.**

$$\text{SelectionStrengthGap} = \text{Strength}(A_1) - \text{Strength}(A_2)$$



- **Correlation:** ρ = 0.220
- **Range:** [-0.34, 0.34]
- **Monotonicity:** Highly stable monotonic relationship with Utility.

F3: Code Snippet

```
def oof_target_mean_smooth(df, col_name, target_col, n_splits=5, alpha=50):
    """
    Out-Of-Fold Target Encoding with Bayesian Smoothing
    """
    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
    oof = np.zeros(len(df))
    global_mean = df[target_col].mean()

    for train_idx, val_idx in kf.split(df):
        X_train, X_val = df.iloc[train_idx], df.iloc[val_idx]

        # Aggregation
        stats = X_train.groupby(col_name)[target_col].agg(['sum', 'count'])

        # Smoothing
        smoothed = (stats['sum'] + alpha * global_mean) / (stats['count'] + alpha)

        # Map to validation set
        oof[val_idx] = X_val[col_name].map(smoothed).fillna(global_mean)

    return oof
```

Feature 4: Interaction Effect

SelectionStrengthGap × Game Stochasticity

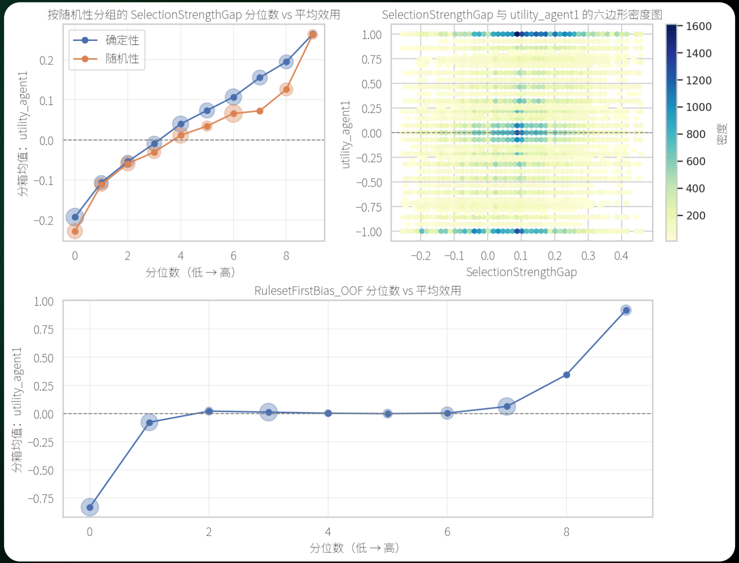
Does luck (Stochasticity) dilute skill (Selection Strength)?

Findings

- **Deterministic**: $\rho \approx 0.210$
- **Stochastic**: $\rho \approx 0.317$ **(+51%)**
- **Result**: **Counter-intuitive!**

In stochastic games, the advantage of a stronger selection strategy is **amplified**, not diminished.

> "Noise amplifies skill" — Stronger agents are more robust to randomness (or "noise") in the environment.



Negative Findings

We investigated features based purely on **Game Attributes**, but they showed weak signals.

Feature	Description	Result
BoardSizeSweetSpot	Is board area in "sweet spot" 2, 25?	$\rho = -0.030$. Only local, weak lift.
StabilityPace	Game duration vs Turns variance.	$\rho \approx 0$. Long-tail noise.
ProofFriendliness	Interaction of ScoreBounds & Game provability.	$\rho \approx 0.005$. Mechanism valid but effect negligible.

Takeaway: Static game attributes are poor predictors on their own. They work best as **regulators** for Agent-based features.

Conclusion

1. **ExplorationGap** ($\rho = 0.107$): More exploration $\rightarrow$ Higher utility.
2. **PlayoutStrengthGap** ($\rho = 0.073$): Modeling strategy Strength > Simple Matching.
3. **SelectionStrengthGap** ($\rho = 0.220$): Strongest main effect.
4. **Interaction** ($\rho = 0.215$): Stochasticity **amplifies** the effect of Selection Strength ($+51\%$).

Final Thought: In complex game systems, **Agent Mechanism Differences** are the primary source of performance variance, while **Environment Characteristics** act as regulators.

Thank You!