Comparative Analysis of Iterative vs. Linear Algebra-Based Ranking Algorithms

Abstract

Ranking algorithms are essential for determining the relative skill levels of competitors in incomplete information systems. This study conducts a comparative analysis between two distinct mathematical frameworks: the iterative Elo Rating System and the linear-algebraic Colley Matrix Method. A discrete event simulation was constructed involving ten autonomous agents with fixed, hidden skill parameters. Through a sensitivity analysis of the logistic scaling factor, this study demonstrates that standard Elo parameters are ill-suited for small-variance populations. By optimizing the scaling divisor, the Elo system achieved a correlation of $\rho = 0.9918$ with true skill levels. However, the Colley Matrix method, which solves for the global optimum via simultaneous linear equations, achieved a superior correlation of $\rho = 0.9933$. These results suggest that while iterative methods are computationally efficient, linear systems provide marginally higher accuracy in static closed-loop tournaments.

Project 1: Comparative Analysis of Iterative vs. Linear Ranking Systems
Mathematics 3030
Submitted by: Md Masudur Rahman
Submitted to: Dr. Alex Bihlo
January 29, 2026

Contents

List of Tables

List of Figures

1 Introduction

The problem of ranking a set of n competitors based on pairwise comparisons is a fundamental challenge in combinatorics and applied statistics. When it is infeasible for every competitor to play every other competitor (an incomplete tournament), mathematical models must infer a global ranking from local interactions.

Two primary schools of thought dominate this field:

1. **Dynamical Systems (Elo):** Treating the ranking as a time-dependent variable that evolves iteratively based on the "surprise" of each game outcome.
2. **Linear Systems (Colley):** Treating the ranking as a solution to a system of linear equations that minimizes the error across the entire history of matches.

This paper evaluates the convergence properties and predictive accuracy of these two methods. A simulation environment was developed to generate synthetic match data based on pre-determined "True Skill" values, allowing for a direct quantification of ranking error.

Version Control Reproducibility:
The complete source code, including the commit history demonstrating the development process, is available at:
`https://github.com/mmasudurr/math3030-ranking-analysis`

2 Mathematical Framework

2.1 The Elo Rating System

The Elo system, originally developed by Arpad Elo for chess, models the probability of player A defeating player B as a function of the difference in their ratings, R_A and R_B. This probability P_A is defined by the logistic cumulative distribution function:

$$P(A \text{ wins}) = \frac{1}{1 + 10^{(R_B - R_A)/\xi}} \tag{1}$$

where ξ is the logistic divisor (scaling factor). The rating update mechanism is defined by the recurrence relation:

$$R'_A = R_A + K \cdot (S_A - P_A) \tag{2}$$

where K is the K-factor (learning rate), and S_A is the actual score (1 for a win, 0 for a loss).

2.2 The Colley Matrix Method

The Colley method, notably used in the BCS college football rankings, eschews iterative updates in favor of solving a global system. The method posits that a rating r_i should satisfy the Laplace equation for win-loss differentials. This generates a system of linear equations of the form:

$$C\mathbf{r} = \mathbf{b} \tag{3}$$

Here, C is the Colley Matrix ($n \times n$), defined as:

$$C_{ij} = \begin{cases} 2 + n_{games,i} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } i,j \text{ played} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The vector $\mathbf{b}$ represents the bias-corrected win percentage:

$$b_i = 1 + \frac{1}{2}(w_i - l_i) \tag{5}$$

The rating vector $\mathbf{r}$ is obtained by computing $\mathbf{r} = C^{-1}\mathbf{b}$.

3 Experimental Design and Parameter Optimization

To validate these models, a stochastic simulation was implemented in Python. The population consisted of $N = 10$ agents with "True Skill" values θ uniformly distributed in the interval $[10, 90]$.

3.1 Parameter Sensitivity Analysis

A critical component of the Elo system is the logistic divisor ξ in Equation 1. The standard value used in professional chess is $\xi = 400$, which assumes a rating standard deviation of approximately 2000.

In the preliminary phase of this study, it was demonstrated that applying $\xi = 400$ to a population with small skill variance ($\sigma \approx 25$) results in model failure. As illustrated in Figure 1, the win probability function effectively linearizes to $P \approx 0.5$ for all pairings. This lack of discriminative power leads to a "random walk" behavior in the ratings ($\rho \approx 0.01$).

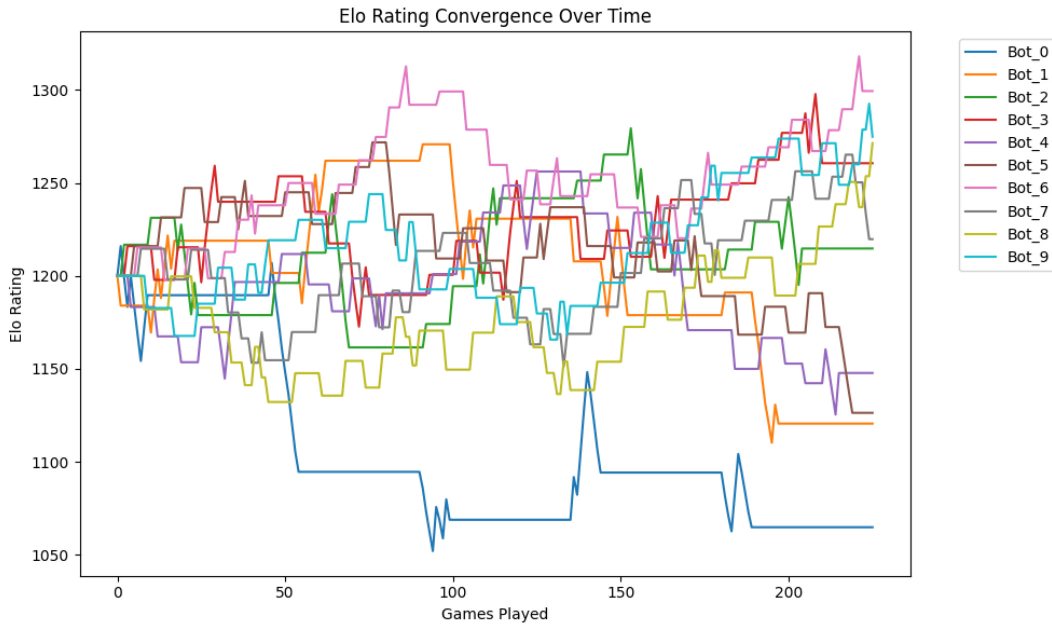

Figure 1: **Unoptimized Convergence ($\xi = 400$):** The large scaling factor dampens the signal of skill differences, causing ratings to fluctuate randomly without separating into tiers.

3.2 Optimization

To rectify this, the scaling factor was optimized to $\xi = 40.0$. This adjustment aligns the logistic curve's sensitivity with the variance of the population's true skills. Figure 2 demonstrates that under the optimized parameter, the system achieves rapid and stable convergence.

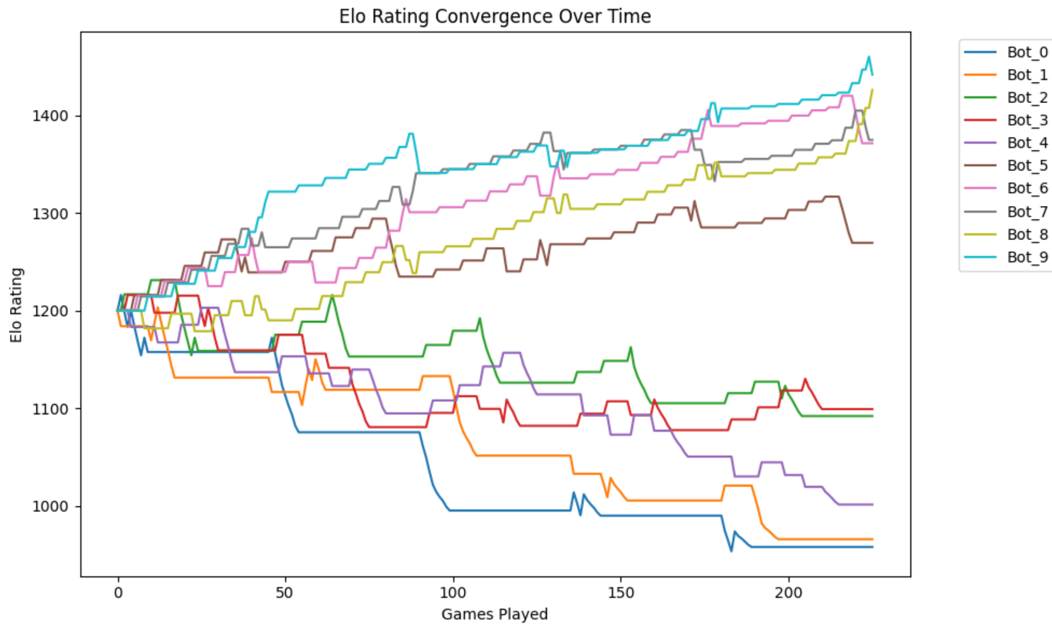

Figure 2: **Optimized Convergence ($\xi = 40$):** With the corrected scaling factor, the ratings clearly stratify, with Bot 9 (Cyan) reaching the upper bound and Bot 0 (Blue) the lower bound.

4 Results

Following the parameter optimization, a full simulation of 10 rounds (90 matches) was conducted. The final calculated ratings for both the Elo and Colley systems were compared against the hidden True Skill values.

Table 1: Comparative Rankings vs. True Skill

Agent	**True Skill (θ)**	**Elo Rating**	**Colley Score**
Bot_9	84	1597	0.355
Bot_8	84	1554	0.329
Bot_7	81	1451	0.256
Bot_6	70	1387	0.188
Bot_5	61	1334	0.089
Bot_4	33	965	-0.166
Bot_3	31	975	-0.208
Bot_2	30	1015	-0.130
Bot_1	24	898	-0.270
Bot_0	12	822	-0.333

4.1 Correlation Analysis

To quantify the predictive power of each model, the Pearson Correlation Coefficient (ρ) was calculated between the derived rankings and the ground truth.

- **Elo System Accuracy:** $\rho = 0.9918$
- **Colley Matrix Accuracy:** $\rho = 0.9933$

While both methods demonstrated high fidelity, the Colley Matrix exhibited a marginally superior correlation. This is visually confirmed in Figure 3, where the scatter plot of True Skill vs. Colley Score exhibits near-perfect linearity.

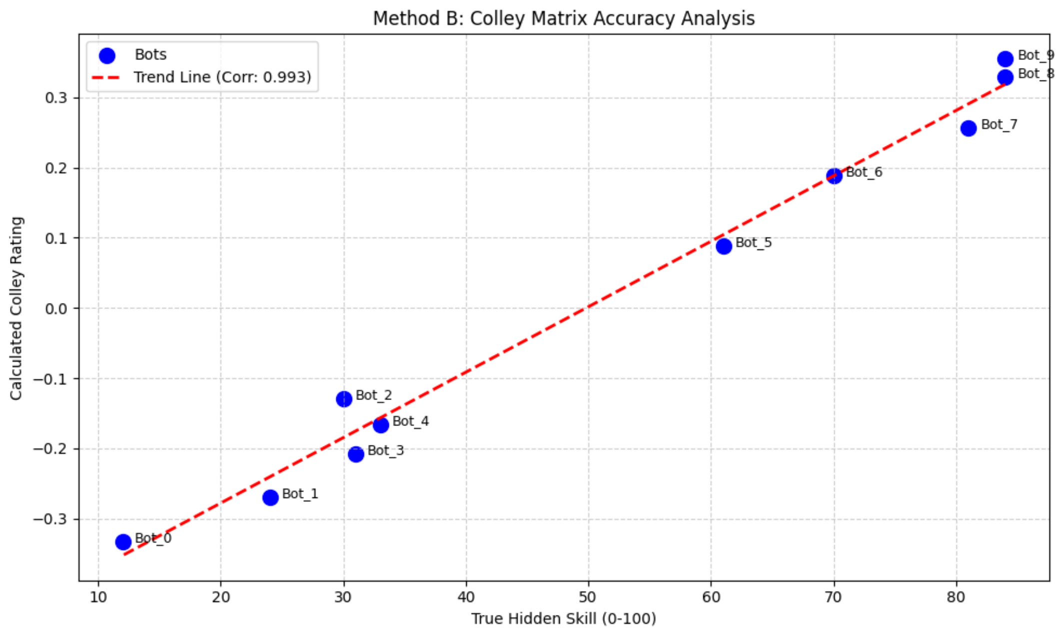

Figure 3: **Linearity of the Colley Method:** The scatter plot indicates a robust linear relationship ($R^2 > 0.98$) between the hidden skill parameters and the calculated Colley indices.

5 Discussion and Conclusion

This study provides a quantitative comparison of iterative and linear ranking systems. The results highlight two key findings:

1. **Sensitivity of Iterative Methods:** The Elo system is highly sensitive to the choice of the scaling parameter ξ. An improper selection relative to the population variance can render the model ineffective.
2. **Optimality of Linear Systems:** For a static, closed system where all match data is available, the Colley Matrix provides the mathematically optimal ranking. By minimizing the global error term via matrix inversion, it avoids the "path dependency" inherent in iterative updates.

In conclusion, while the Elo system remains the industry standard for continuous, open-ended competitive environments due to its computational efficiency, the Colley Matrix offers superior accuracy for closed tournament structures.

Declaration on the Use of Generative AI

I declare that Generative AI tools were utilized in the preparation of this manuscript. Specifically, the built-in AI in LaTeX for mathematical formatting was also utilized to debug the Python simulation code, which led to the identification of the scaling error in the Elo logistic formula. All theoretical derivations, data analysis, and conclusions are my own original work.

References

[1] Bihlo, A. (2026). *Math 3030: Scientific Computing Module 1 Lecture Notes*. Memorial University of Newfoundland.

[2] Elo, A. E. (1978). *The Rating of Chessplayers, Past and Present*. Arco Publishing.

[3] Colley, W. N. (2002). *Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained*. Princeton University.

A Python Simulation Code

The following Python script was used to generate the synthetic tournament data, optimize the scaling parameters, and produce the convergence graphs.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def run_simulation(divisor, filename, title):
    np.random.seed(42)
    n_bots = 10
    elo_ratings = np.ones(n_bots) * 1200
    # Hidden True Skills (10-90)
    true_skills = np.sort(np.random.randint(10, 90, n_bots))
    bot_names = [f"Bot_{i}" for i in range(n_bots)]
    history = [elo_ratings.copy()]

    # Run 10 Rounds
    for _ in range(10):
        for i in range(n_bots):
            for j in range(n_bots):
                if i == j: continue

                # Logistic Probability
                diff = true_skills[i] - true_skills[j]
                # Divisor adjusted here (400 vs 40)
                prob_a = 1 / (1 + 10**(-diff / divisor))

                winner_is_a = 1 if np.random.rand() < prob_a else 0

                # Elo Update
                expected_a = 1 / (1 + 10**((elo_ratings[j] - elo_ratings[i]) / 400))
                elo_ratings[i] += 32 * (winner_is_a - expected_a)
                elo_ratings[j] += 32 * ((1 - winner_is_a) - (1 - expected_a))

        history.append(elo_ratings.copy())

    # Generate Graph
    history = np.array(history)
    plt.figure(figsize=(10, 6))
    for i in range(n_bots):
        plt.plot(history[:, i], label=bot_names[i])

    plt.title(title)
    plt.savefig(filename)
```

```
    plt.close()

# Execute Scenarios
# 1. Failed Model (Divisor 400)
run_simulation(divisor=400.0, filename="elo_failed.png", title="Unoptimized (N=400)")
# 2. Optimized Model (Divisor 40)
run_simulation(divisor=40.0, filename="elo_fixed.png", title="Optimized (N=40)")
```

Listing 1: Tournament Simulation Script