An Elo-based Pitching vs. Batting Model for Baseball

Jack O'Connor, Computer Science, Professor James Glenn, Computer Science, Yale University

Version 1: Basic Elo

What is Elo?

- Purpose: calculate the relative skills of players in zero-sum games. Originally designed for chess, we will be utilizing a very similar model to predict outcomes for baseball.
- Algorithm: initialize all new players' ratings to a standard value. In all matchups of player A vs. B, calculate the expected value for player A in the matchup according to the following formula:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}$$

- Update the players' rankings according to the formula (where R denotes rating, E denotes expected value):

$$R'_A = R_A + K \cdot (S_A + E_A)$$

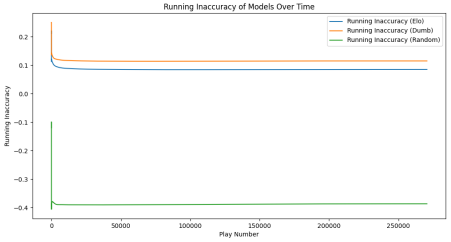
- The first attempt assigned each player, pitchers and batters a rating of 1500, and a K value of 32 was used to update players' values. The first attempt used a partial wins model, mapping a single to 0.25 wins, double 0.5 wins, triple 0.75 wins, home run 1.0 wins for the batter (Where S_b represents the batter, S_p the pitcher; S_p = 1-S_b). **Training:** the model was trained on 75000 at-bats from the 2021 and 2022 MLB seasons.
- **Testing:** Testing was performed over 25000 at-bats from the 2022 MLB season. Two major forms of testing were used. The first measured the long-term accuracy of the model by calculating the difference between the number of wins predicted by the model and the number of wins that actually occurred. The second measured the average error of each individual prediction; this was called the play-by-play accuracy.
- **Performance:** The basic model was highly inaccurate at predicting results when it came to predicting partial outcomes. To adjust, testing was changed to have a hit/no-hit model where a hit was scored as a full win and an out was a loss. This model was highly accurate long-term (3% inaccurate), but inaccurate play-by-play (41% inaccurate)

KEY FINDINGS

- Overall, the final model was inaccurate around 8.6% of the time when tested against the training data when tested long-term. Play-by play accuracy was better than the baseline and random models but marginally so

Version 2: Partial wins model & further attempts

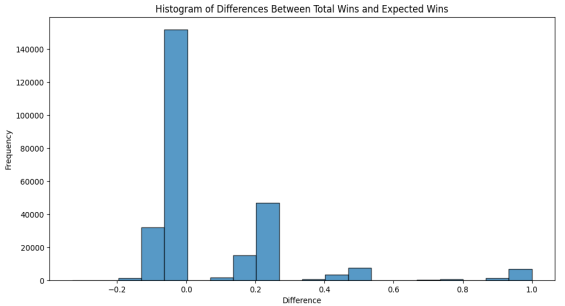
- In the previous attempts, the partial wins model did not prove adaptable to the Elo algorithm. To try and integrate the Elo prediction algorithm into a partial wins model, a new procedure was used: after predicting the probability of a hit with the Elo algorithm, we used each pitcher's historical outcomes to get the expected value of a hit
- For example, if a pitcher had a 50% chance of throwing a hit according to Elo, and when they were hit on, they historically threw 25% home runs and 75% singles, the expected value was 0.50 * (0.25 * 1.0 + 0.75 * 0.25) = 0.21875
- This model proved much more accurate than the Elo algorithm's previous attempt at predicting outcomes; it had 8.6% accuracy over the long-term, beating both the baseline and random models



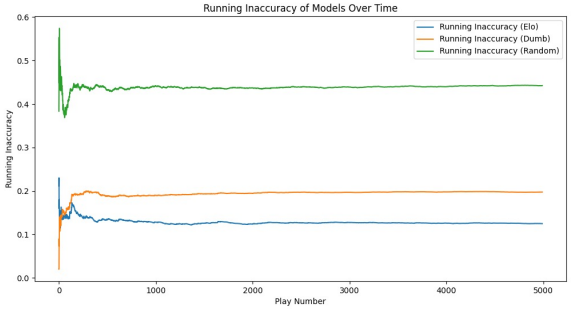
The running accuracy of the model over time shows that eventually, the Elo-based model converges the closest to 0 inaccuracy at 8%. While the "dumb" model which predicts an out every time converges to around 11% inaccuracy, and the random model is much farther off at about 39%

Other Factors/Attempts/Findings

- Attempted to account for experience, as other Elo models have, by decaying the K-factor as pitchers and batters gain more experience
- Original Elo model had an interesting imbalance since pitchers and batters play an imbalanced game: 70% of all at-bats ended in an out. Note that new MLB rules changes could have interesting impacts on the accuracy of this model
- The baseline model reflected an unfair informational advantage since it predicted an expected value equal to the average expected value over all of the *testing* data. Despite this, the Elo-based model remained more accurate (see graphs)



The model frequently predicts very close to the observed results. The model almost never predicts a home-run when there was actually an out (value of -1.0; an expected but still interesting finding)



Future Possibilities

- Could potentially account for factors like player handedness, number of batters on base, pitcher fatigue (n pitches thrown)
- Implement other tests: categorical cross-entropy, mean squared error, etc.
- Apply model to 2023 mlb season and possibly betting lines to see if the model performs well