Bayesian Hierarchical Model for Predicting Baseball Game Outcomes

Introduction

This document outlines the methodology for predicting baseball game outcomes using Bayesian hierarchical modeling and simulation techniques. The approach leverages event-level data from batter-pitcher matchups to generate prior distributions, update them with observed data, and simulate game outcomes with uncertainty estimates.

Step 1: Data Preparation

We begin by preparing a dataset containing play-by-play baseball event data. The dataset is aggregated by batter-pitcher matchup, with counts of various event types (e.g., strikeouts, walks, hits, etc.) and the corresponding matchup characteristics (e.g., batter hand, pitcher hand, runner positions, home/away).

Step 2: Negative Binomial & Multinomial Hyperparameters via MCMC

1. **Matchup-Level Totals:** For each batter-pitcher matchup, let T_m be the total number of observed events (summing across categories) for matchup m. We assume

$$T_m \sim \mathrm{NB}(r, p) \quad \text{(truncated at } T_m \geq 1),$$

   where (r, p) are global hyperparameters. Truncation ensures $T_m \neq 0$.

2. **Multinomial Allocation:** Conditioned on T_m, the counts among event categories are modeled by

$$(\alpha_{m,1}, \ldots, \alpha_{m,K}) \sim \mathrm{Multinomial}\big(T_m, \mathbf{q}\big),$$

   where $\mathbf{q} = (q_1, \ldots, q_K)$ is a probability vector over K event categories.

3. **MCMC Sampling:** We fit this hierarchical model via Markov Chain Monte Carlo, obtaining posterior draws of (r, p) for the negative binomial and the multinomial probability vector $\mathbf{q}$. Convergence is assessed via diagnostics such as $\hat{R}$ and effective sample sizes.

Step 3: Generating Matchup Priors (Alpha Vectors)

1. **Drawing Row Sums:** From each MCMC iteration, we draw T_m from the truncated zero negative binomial distribution:

$$T_m \sim \mathrm{NB}(\hat{r}, \hat{p}) \quad (T_m \geq 1).$$

2. **Allocating Counts via Multinomial:** We then draw $\alpha_m = (\alpha_{m,1}, \ldots, \alpha_{m,K}) \sim \mathrm{Multinomial}(T_m, \mathbf{q})$. Since $\sum_k \alpha_{m,k} = T_m \neq 0$, we avoid any all-zero rows.

3. **Event Probabilities:** We treat these alpha parameters as *Dirichlet parameters* rather than simply normalizing them. Concretely, we draw

$$\mathbf{p}_m = (p_{m,1}, \ldots, p_{m,K}) \sim \text{Dirichlet}\big(\alpha_{m,1}, \ldots, \alpha_{m,K}\big).$$

These $p_{m,k}$ serve as priors for subsequent updating or direct simulation of at-bats.

Step 4: Bayesian Updating for Posterior Inference

1. **Likelihood Specification:** Observed event counts y_m follow a multinomial distribution:

$$y_m \sim \text{Multinomial}(N_m, p_m)$$

2. **Bayesian Updating:** Update the Dirichlet prior with the observed counts:

$$\alpha_{m,i}^{posterior} = \alpha_{m,i}^{prior} + y_{m,i}$$

3. **Posterior Distribution:** The resulting posterior distribution reflects the updated beliefs about event probabilities given the observed data. We draw from the Dirichlet in the same way we did for prior probabilities to generate our posterior probabilities.

Step 5: Simulating Game Outcomes

1. **Sequential Event Simulation:** Simulate each plate appearance by sampling an event from the posterior event probability distribution. Based on the sampled event, adjust the game state (e.g., runner advancement, score changes).
2. **Game-Level Simulation:** Repeat the event-level simulation for the entire game.
3. **Monte Carlo Replication:** Run multiple simulations to estimate the distribution of possible game outcomes.

Step 6: Deriving Outcome Probabilities

1. **Probability Estimation:** Calculate the probability of a team winning as the proportion of simulations in which that team has a higher final score.
2. **Uncertainty Quantification:** Generate credible intervals for the estimated win probabilities to capture model uncertainty.

Conclusion

This methodology provides a structured Bayesian framework for predicting baseball game outcomes. By leveraging hierarchical modeling, Bayesian updating, and simulation techniques, we can generate probabilistic estimates of event outcomes with quantified uncertainty, enhancing our understanding of in-game dynamics and player performance.