Machine Learning vs. Mathematical Models

- Machine Learning/statistical models are fit from historical data
- Mathematical Models are equations written on paper

Models fit from historical data

- what data do we need?
- get data
- empirical grid
- logistic regression — No
- XGBoost grid

Mathematical Models

- Poisson(λ) model
- Poisson(λ_x), Poisson(λ_y) model
- overdispersion hyperparameter K

See Code

A Estimating f using a mathematical, not a statistical, model

In this Section, we detail our modeling process for estimating the grid function $f = f(I,R)$ which, assuming both teams have randomly drawn offenses, computes the probability a team wins a game after giving up R runs through I complete innings. In particular, we compare statistical models fit from observational data to mathematical probability models, which are superior.

To account for different run environments accross different seasons and leagues (NL vs. AL), we estimate a different grid for each league-season. We begin by estimating f from our observational dataset of half-innings from 2010 to 2019. The response variable is a binary indicator denoting whether the pitcher's team won the game, and the features are the inning number I, the runs allowed through that half-inning R, the league, and the season. Note that if a home team leads after the top of the 9^{th} inning, then the bottom of the 9^{th} is not played. Therefore, to avoid selection bias, we exclude all 9^{th} inning instances in which a pitcher pitches at home.

With enough data, the empirical grid (e.g., binning and averaging over all combinations of I and R within a league-season) is a great estimator of f. In Figure 13a we visualize the empirical grid fit on a dataset of all half-innings from 2019 in which the home team is in the National League. The function f should be monotonic decreasing in R. In particular, as a pitcher allows more runs through a fixed number of innings, his team is less likely to win the game. It should also be monotonic increasing in I because giving up R runs through I innings is worse than giving up R runs through $I+i$ innings for $i > 0$, since giving up R runs through $I+i$ innings implies a pitcher gave up no more than R runs through I innings. The empirical grid, however, is not monotonic in either R or I because each league-season dataset is not large enough. Moreover, even when we use our entire dataset of all half-innings from 2010 to 2019, the empirical grid is still not monotonic in R or I.

To force our fitted f to be monotonic, we use XGBoost with monotonic constraints, tuned using cross validation (Chen and Guestrin, 2016). We visualize our 2019 NL XGBoost fit in Figure 13b. We indeed see that the fitted f is decreasing in R and increasing in I. Additionally, $R \mapsto f(I,R)$ is mostly convex: if a pitcher has already allowed a high number of runs, there is a lesser relative impact of throwing an additional run on winning the game. Nonetheless, XGBoost overfits, especially towards the tails (e.g., for R large). For instance, the 2019 NL XGBoost model indicates that the probability of winning a game after allowing 10 runs through 9 innings is about 0.11, which is too large.

As there is not enough data to use machine learning to fit a separate grid for each league-season

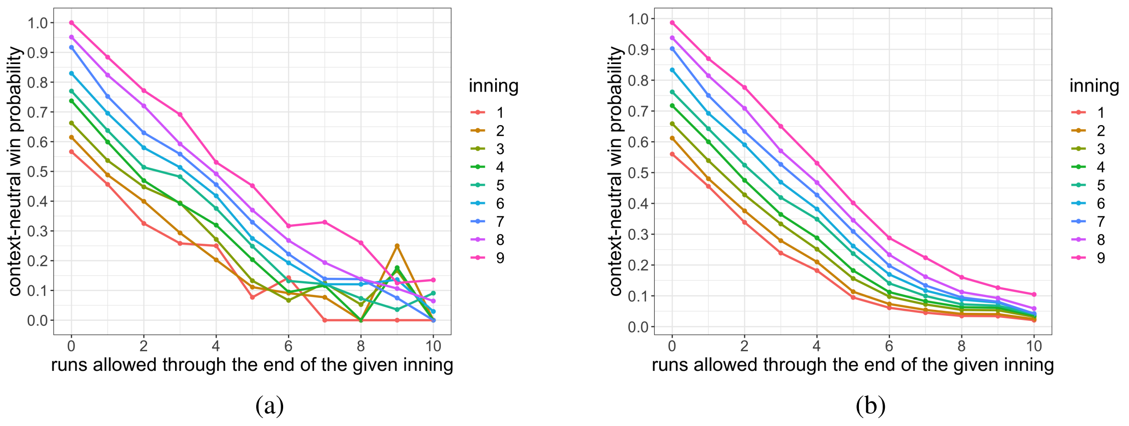


Figure 13: Estimates of the 2019 National League function $R \mapsto f(I,R)$ using the empirical grid (left) and XGBoost with monotonic constraints (right).

without overfitting, we turn to a parametric mathematical model. Indeed, the power of parameterization is that it distills the information of a dataset into a concise form (e.g., into a few parameters), allowing us create a strong model from limited data. Because the runs allowed in a half-inning is a natural number, we begin our parametric quest by supposing that the runs allowed in a half-inning is a Poisson(λ) random variable. In particular, denoting the runs allowed by the pitcher's team's batters in inning i by X_i and the runs allowed by the opposing team in inning i (for innings i after the pitcher exits the game), we assume

$$X_i, Y_i \overset{i.i.d.}{\sim} \text{Poisson}(\lambda). \tag{A.1}$$

Then the probability that a pitcher wins the game after allowing R runs through I innings, assuming win probability in overtime is $1/2$, is

$$f(I,R|\lambda) := \mathbb{P}\left(\sum_{i=1}^{9} X_i > R + \sum_{i=I+1}^{9} Y_i\right) + \frac{1}{2} \cdot \mathbb{P}\left(\sum_{i=1}^{9} X_i = R + \sum_{i=I+1}^{9} Y_i\right). \tag{A.2}$$

If $I = 9$, this is equal to

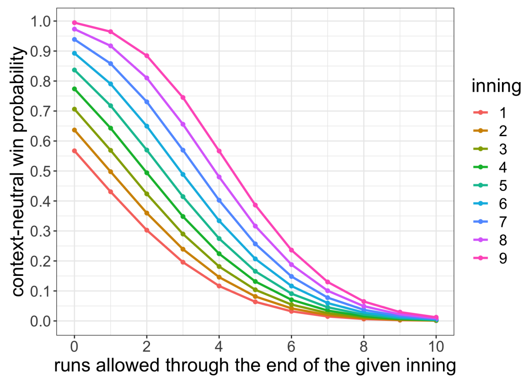
$$\mathbb{P}\big(\text{Poisson}(9\lambda) > R\big) + \frac{1}{2} \cdot \mathbb{P}\big(\text{Poisson}(\lambda) = R\big). \tag{A.3}$$

If $I < 9$, it is equal to

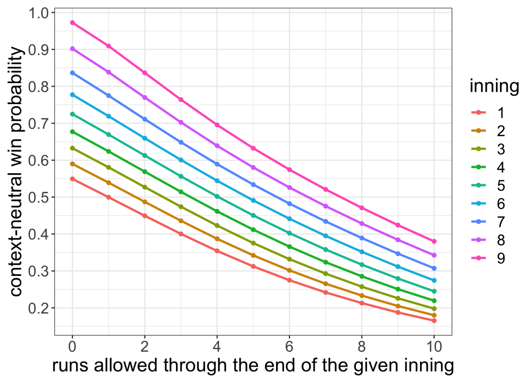
$$\mathbb{P}\big(\text{Skellam}(9\lambda, (9-I-1)\lambda) > R\big) + \frac{1}{2} \cdot \mathbb{P}\big(\text{Skellam}(9\lambda, (9-I-1)\lambda) = R\big), \tag{A.4}$$

noting that the Skellam distribution arises as a difference of 2 independent Poisson distributed random variables. Then, we estimate λ separately for each league-season by computing each team's mean runs allowed in each half inning, and then averaging over all teams.

In Figure 14a we visualize the estimated f according to our Poisson model (A.2) using the 2019 NL λ. We see that f is decreasing in R, increasing in I, convex in the tails of R, and is smooth. Nonetheless, some of the win probability values from this model are unrealistic. For instance, it implies the probability of winning the game after shutting out the opposing team through 9 innings is about 99%, which is too high, and the probability of winning the game after allowing 10 runs through 9 innings is about 1%, which is too low.



(a)



(b)

Figure 14: Estimates of the 2019 National League function $R \mapsto f(I,R)$ using our Poisson model (A.2) with constant λ (left) and our Poisson model (A.8) with a truncated normal prior (A.7) on 2 team strength parameters λ_X and λ_Y (right).

The win probability values at both tails of R are too extreme in our original Poisson model (A.6) because we assume both teams have the same mean runs per inning λ. This is an unrealistic assumption: in real life, a baseball season involves teams of varying strength playing against each other. When teams of differing batting strength play each other, win probabilities differ. For instance, when a great hitting team allows 7 runs to a terrible hitting team, the great hitting team has a larger probability of coming back to win the game than a worse hitting team would. Thus, accounting for random differences in team strength across games should flatten the $f(I,R)$ grid.

On this view, it is more realistic to assume the pitcher's team and the opposing team have their own runs scored per inning parameters,

$$X_i \overset{i.i.d.}{\sim} \text{Poisson}(\lambda_X) \quad \text{and} \quad Y_i \overset{i.i.d.}{\sim} \text{Poisson}(\lambda_Y), \tag{A.5}$$

and

$$f(I,R|\lambda_X,\lambda_Y) := \mathbb{P}\left(\sum_{i=1}^{9} X_i > R + \sum_{i=I+1}^{9} Y_i\right) + \frac{1}{2}\cdot\mathbb{P}\left(\sum_{i=1}^{9} X_i = R + \sum_{i=I+1}^{9} Y_i\right). \tag{A.6}$$

Moreover, to capture the variability in team strength across each of the 30 MLB teams, we impose a positive normal prior,

$$\lambda_X, \lambda_Y \sim \mathcal{N}_+(\lambda, \sigma_\lambda^2). \tag{A.7}$$

We estimate the prior hyperparameters λ and σ_λ separately for each league-season by computing each team's mean and s.d. of the runs allowed in each half inning, respectively, and then averaging over all teams.

Given λ_X and λ_Y, we compute Formula (A.6) similarly as before using the Poisson and Skellam distributions. We use Monte Carlo integration with $B = 100$ samples to estimate the posterior mean grid,

$$f(I,R|\lambda,\sigma_\lambda^2) \approx \frac{1}{B}\sum_{b=1}^{B} f(I,R|\lambda_X^{(b)},\lambda_Y^{(b)}), \tag{A.8}$$

where $\lambda_X^{(b)}$ and $\lambda_Y^{(b)}$ are i.i.d. samples from the prior distribution (A.7).

In Figure 14b we visualize the estimated f according to this Poisson model (A.8), with prior (A.7), using the 2019 NL λ and σ_λ^2. We see that f is mostly linear in R, rather than convex, and the values of f when R is large are highly unrealistic. For instance, this model indicates that the probability of winning the game after allowing 10 runs through 9 innings is about 38%, which is way too high. This is because our model is overdispersed, i.e. the estimated prior variance σ_λ^2 is too large. For example, too large of a σ_λ^2 allows λ_X and λ_Y to be very far apart, so if a pitcher allows 10 runs through 9 innings and λ_X is much larger than λ_Y, then his team will have a significant chance of coming back to win.

To resolve the overdispersion issue, we introduce a tuning parameter k designed to tune the dispersion across team strengths to match observed data,

$$\lambda_X, \lambda_Y \sim \mathcal{N}_+(\lambda, k\cdot\sigma_\lambda^2). \tag{A.9}$$

In particular, we use $k = 0.28$, which minimizes the log-loss between the observed win/loss column and predictions from the induced grid $f(I,R|\lambda,\sigma_\lambda^2,k)$. In Figure 15 we visualize the estimated f

according to our Poisson model (A.8), with tuned dispersion prior (A.9), using the 2019 NL λ and σ_λ^2. We see that f is decreasing in R, increasing in I, and convex when R is large. In particular, it looks like a smoothed version of the XGBoost grid from Figure 13b. Additionally, the values of the grid at both tails of R seem reasonable. For instance, the model indicates that allowing 0 runs through 9 innings has about a 97% win probability, which is more reasonable than before. For all of these reasons, we use this model for the grid f to compute Grid WAR for starting pitchers.

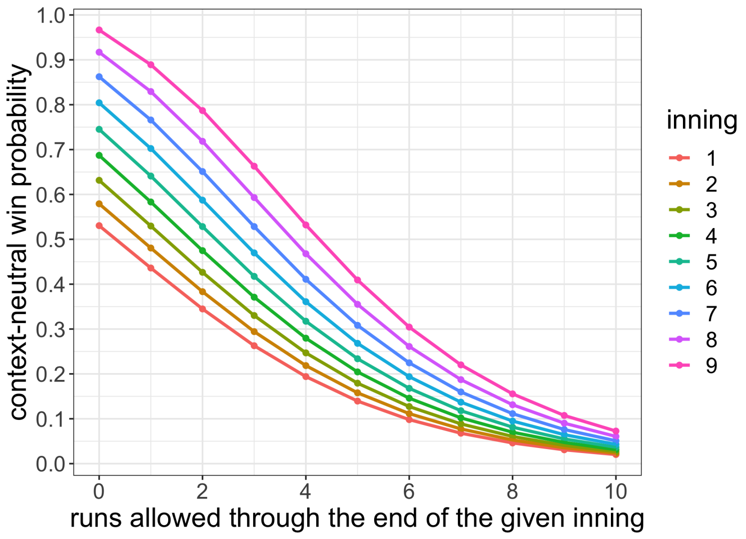


Figure 15: Estimates of the 2019 National League function $R \mapsto f(I,R)$ using our Poisson model (A.8) with tuned dispersion prior (A.9).

B Estimating pitcher quality using Empirical Bayes

In this Section, we describe our parametric Empirical Bayes estimators $\widehat{\mu}_p^{\text{GWAR}}$ and $\widehat{\mu}_p^{\text{FWAR}}$ of pitcher p's quality.

B.1 Empirical Bayes estimators of pitcher quality using Grid WAR

We begin with $\widehat{\mu}_p^{\text{GWAR}}$ which estimates pitcher quality using pitcher p's previous games' Grid WAR and number of games played. Specifically, index each starting pitcher by $p \in \{1, ..., \mathscr{P}\}$ and index pitcher p's games by $g \in \{1, ..., N_p\}$. Let X_{pg} denote pitcher p's observed Grid WAR in game

* We have found a solid model of $f = f(I, R)$!

Even though Runs are NOT poisson, the power of parameterization is that it stores information concisely!

Don't let the perfect be the enemy of the good

Remainder of the Research Process

* Grid WAR when starting pitcher exits in the middle of an inning

* WREP

* Adjustments:
  - park adjustment
  - opposing team's batting adjustment
  - fielding adjustment

will cover in a later lecture

* Results:
  - compare Grid WAR to previous WAR

HW options

1. Review today's lecture
   - take a look at the codefile and data, try to code some of these f grids yourself, review the methodology
2. Brainstorm Research Project Ideas
   - take a sports metric you've always been curious about and Read about how it works!
   - start reading articles/blogs to get some ideas
3. Create roadmaps for executing potential ideas
   - what's the problem in plain English?
     What incremental (or, major) improvements will you make?
     What kind of data do you need?
     What's the simplest version of this problem?