Revisiting pitch framing with Bayesian Additive Regression Trees

Sameer K. Deshpande

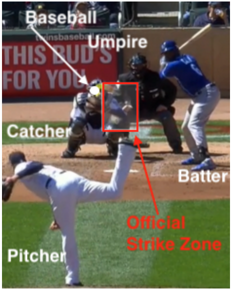
University of Wisconsin–Madison

EURO 2022

6 July 2022

Pitch framing

- Umpires' ball/strike decisions not deterministic
- Some catchers can influence $\mathbb{P}(\texttt{strike})$
  - Good framers "make balls look like strikes"
  - Good framers "steal strikes"
- Seemingly outsized value relative to # of frameable pitches
  - Top framers save 15–25 runs on avg. per season
  - Estimated \$14M/yr in contract value
  - Only a few frameable pitches per game
- Motivating questions:
  - How certain are we about the value of framing?
  - Is estimated framing value stable season-to-season?



Estimating the impact of pitch framing

- Catcher expected to save $\rho \times (p - p_0)$ runs over baseline
  - ρ: expected value of a called strike relative to ball
  - p & p_0: called strike prob. & league-average baseline prob.
  - ☺ Avoids awarding credit for obvious calls & accounts for context
  - ☹ We don't know the exact values ρ, p or p_0 for each pitch

Estimating the impact of pitch framing

- Catcher expected to save $\rho \times (p - p_0)$ runs over baseline
  - ρ: expected value of a called strike relative to ball
  - p & p_0: called strike prob. & league-average baseline prob.
  - ☺ Avoids awarding credit for obvious calls & accounts for context
  - ☹ We don't know the exact values ρ, p or p_0 for each pitch
- Value of a called strike estimated using historic data
  - ρ: diff. in avg. # runs in scored in half-inning after called ball & strike
  - Assumption: ρ depends only on count & outs
  - E.g. value of a strike on 1-1 pitch w/ 1 out is 0.14

Estimating the impact of pitch framing

- Catcher expected to save $\rho \times (p - p_0)$ runs over baseline
  - ρ: expected value of a called strike relative to ball
  - p & p_0: called strike prob. & league-average baseline prob.
  - ☺ Avoids awarding credit for obvious calls & accounts for context
  - ☹ We don't know the exact values ρ, p or p_0 for each pitch
- Value of a called strike estimated using historic data
  - ρ: diff. in avg. # runs in scored in half-inning after called ball & strike
  - Assumption: ρ depends only on count & outs
  - E.g. value of a strike on 1-1 pitch w/ 1 out is 0.14
- Estimated called strike probabilities
  - $\mathbb{P}(\texttt{called strike}) = F(\texttt{location}, \texttt{personnel}, \texttt{context})$
  - `context`: count, outs, baserunner configuration, ...
  - Baseline p_0: average of "counterfactual" catcher probs.
  - Need to estimate F and get reliable uncertainty estimates

Remaining plan

- D & Wyner (2017)'s hierarchical model for $\mathbb{P}(\texttt{called strike})$
  - ☺ "Shares statistical strength" between umpires
  - ☹ Heavily over-parametrized & indirect parametrization of `location`
  - ☹ Limited to two-way interactions (e.g. umpire $\times$ count)

Remaining plan

- D & Wyner (2017)'s hierarchical model for $\mathbb{P}(\texttt{called strike})$
  - ☺ "Shares statistical strength" between umpires
  - ☹ Heavily over-parametrized & indirect parametrization of `location`
  - ☹ Limited to two-way interactions (e.g. umpire $\times$ count)

- Towards a fully nonparametric model
  - Uses Bayesian Additive Regression Trees
  - ☺ Avoid specifying functional form of F or potential interactions *a priori*
  - Implementation challenge: handling all of the categorical inputs

Model from D. & Wyner (2017)

- Let $y = 1$ for called strike and $y = 0$ for ball
- For umpire u, we model

$$\log\left(\frac{\mathbb{P}(y=1)}{\mathbb{P}(y=0)}\right) = \theta^u_0 + \theta^u_b + \theta^u_c + \theta^u_p + \theta^u_{count} + \theta^u_{\texttt{loc}} \times \log\left(\frac{\hat{p}}{1-\hat{p}}\right)$$

- Umpire-specific effects of batters, catchers, pitchers, & count
- For each catcher c: θ^u_c's are i.i.d $\mathcal{N}(\overline{\theta}_c, \sigma^2_c)$
  - Shrink effects on each umpire to common mean $\overline{\theta}_c$
  - Common prior on $\overline{\theta}_c$'s shrinks all avg. catcher effects to 0
- $\hat{p}$ is historical estimate of called strike prob. based on location

Model from D. & Wyner (2017)

- Let $y = 1$ for called strike and $y = 0$ for ball
- For umpire u, we model

$$\log\left(\frac{\mathbb{P}(y=1)}{\mathbb{P}(y=0)}\right) = \theta_0^u + \theta_b^u + \theta_c^u + \theta_p^u + \theta_{count}^u + \theta_{\mathtt{loc}}^u \times \log\left(\frac{\hat{p}}{1-\hat{p}}\right)$$

- Umpire-specific effects of batters, catchers, pitchers, & count
- For each catcher c: θ_c^u's are i.i.d $\mathcal{N}(\overline{\theta}_c, \sigma_c^2)$
  - Shrink effects on each umpire to common mean $\overline{\theta}_c$
  - Common prior on $\overline{\theta}_c$'s shrinks all avg. catcher effects to 0
- $\hat{p}$ is historical estimate of called strike prob. based on location

☹ 50+ hours to get 6,000 samples (fit in Stan in an HPC environment)

☹ Heavily over-parameterized ($\geq 170,000$ parameters)

☹ Only supported 2-way interaction (e.g. umpire $\times$ catcher)

Bayesian Additive Regression Trees

- Chipman et al. (2010)'s probit model: $\mathbb{P}(y = 1) = \Phi(f(\boldsymbol{x}))$ where

$$f(\boldsymbol{x}) = \text{sum of } M \text{ binary regression trees}$$

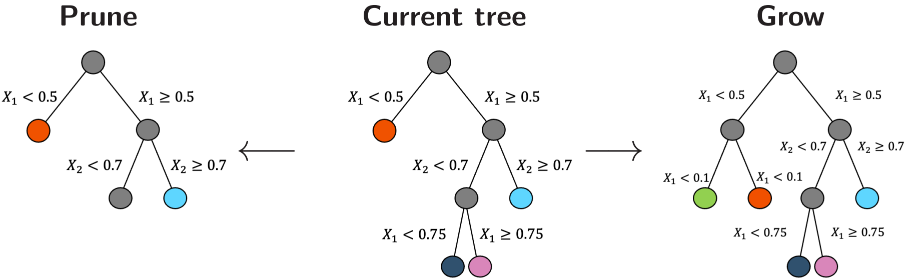
☺ No need to specify functional form of f *a priori*

☺ BART has proven tremendously successful in applications

- Sums of regression trees have great representational flexibility

Gibbs Sampling

- Introduce latent $z \sim \mathcal{N}(f(\boldsymbol{x}), 1)$ s.t. $y = \mathbb{1}(z \geq 0)$
- Given sum of trees $f(\boldsymbol{x})$, draw z from truncated normal
- Given z, sequentially grow or prune each regression tree at random

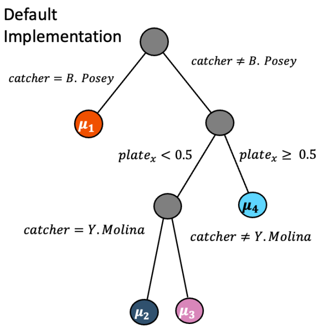


Handling categorical predictors

- We have several categorical variables with many levels
  - Each season: 900+ batters, 100+ catchers, ~ 100 umpires

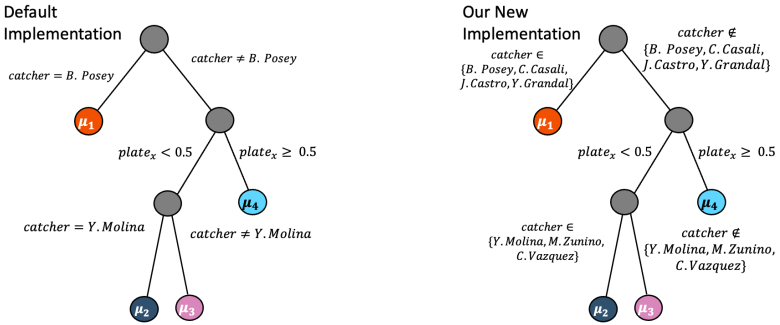
Handling categorical predictors

- We have several categorical variables with many levels
  - Each season: 900+ batters, 100+ catchers, ~ 100 umpires
- Default implementation of BART: create binary dummy variables



Handling categorical predictors

- We have several categorical variables with many levels
  - Each season: 900+ batters, 100+ catchers, ~ 100 umpires
- Default implementation of BART: create binary dummy variables
- Our implementation: randomly partition set of available levels

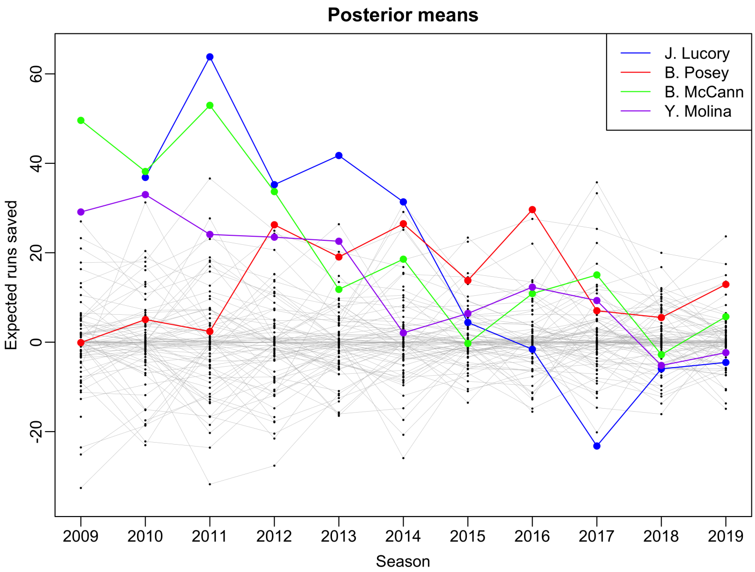


Framing the problem

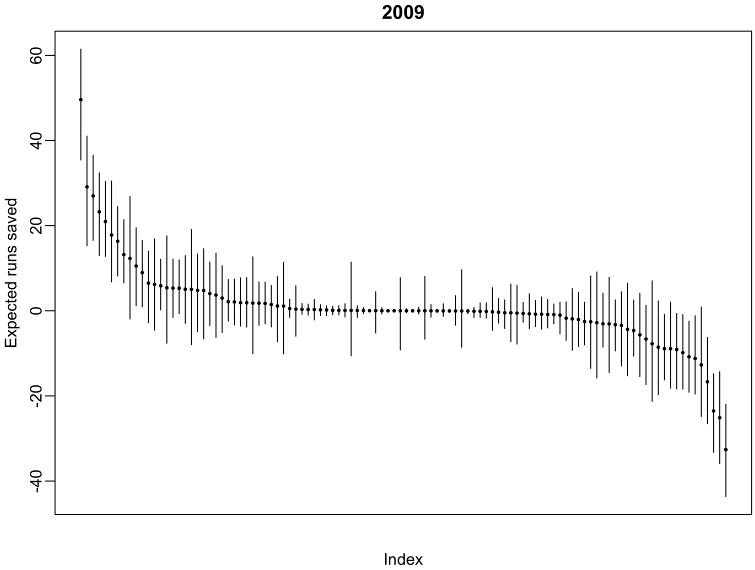
Estimating strike probabilities

Results & next steps

Expected runs saved



Uncertainty in expected runs saved



New directions

- Causal estimates?
  - We summed $\hat{\rho} \times (\hat{p} - \hat{p}_0)$ for each catcher
  - Expected runs saved is confounded by pitcher, batter, location, ...
  - Partial salvage: average out location (see D. & Wyner (2017))

New directions

- Causal estimates?
  - We summed $\hat{\rho} \times (\hat{p} - \hat{p}_0)$ for each catcher
  - Expected runs saved is confounded by pitcher, batter, location, ...
  - Partial salvage: average out location (see D. & Wyner (2017))
- Smoother called strike estimates
  - For each (`personnel`, `context`) output a smooth heat map
  - Sum of treed GPs or sum of treed low rank thin plate splines?

New directions

- Causal estimates?
  - We summed $\hat{\rho} \times (\hat{p} - \hat{p}_0)$ for each catcher
  - Expected runs saved is confounded by pitcher, batter, location, ...
  - Partial salvage: average out location (see D. & Wyner (2017))
- Smoother called strike estimates
  - For each $(\texttt{personnel}, \texttt{context})$ output a smooth heat map
  - Sum of treed GPs or sum of treed low rank thin plate splines?
- Leveraging full pitch trajectory?
  - Effect of pitch movement & break on umpire's decision?
  - *Functional* inputs for treed models?

New directions

- Causal estimates?
  - We summed $\hat{\rho} \times (\hat{p} - \hat{p}_0)$ for each catcher
  - Expected runs saved is confounded by pitcher, batter, location, ...
  - Partial salvage: average out location (see D. & Wyner (2017))
- Smoother called strike estimates
  - For each $(\texttt{personnel}, \texttt{context})$ output a smooth heat map
  - Sum of treed GPs or sum of treed low rank thin plate splines?
- Leveraging full pitch trajectory?
  - Effect of pitch movement & break on umpire's decision?
  - *Functional* inputs for treed models?

Thanks, y'all!

Email: `sameer.deshpande@wisc.edu`
D. & Wyner (2017): arXiv:1704.00823
Website: `https://skdeshpande91.github.io`