Simulation: Win Probability in Simplified Football

* XGBoost → win probability estimates

$$\widehat{WP}_{Go} = .63, \quad \widehat{WP}_{FG} = .60, \quad \widehat{WP}_{Punt} = .57$$

People say: therefore you should Go.

But what if these values are not right?
What if the model sucks?

* People saw a dataset of ≈ 200,000 1st down plays in the last 20 years and ≥ 500,000 plays altogether

so they think Big Data ⟹ Good Model.

* i index of i^{th} play

$$y_i = \begin{cases} 1 & \text{if team with possession on play } i \text{ wins that game} \\ 0 & \text{if loses} \end{cases}$$

What do you notice about this outcome variable?

- noisy
- Extreme Autocorrelation

* Every play i from the same game shares the exact same value of y_i or ($1-y_i$ if other team)

→ $\boxed{\text{there is only one independent draw of the outcome win/loss for each game}}$

* there are about ≈ 4000 games in the last 15 years, so the effective sample size of our model is $\approx 4{,}000$ not $\approx 500{,}000$.

→ SMALL data Regime.

yardline	point spread	...	score diff	outcome win/loss	
70	3		0	1	200 plays game 1
60	3		0	1	
52	3		0	1	
		⋮			
40	−3		−7	0	
					200 plays game 2

* XGBoost → win probability estimates

$$\widehat{WP}_{Go} = .63, \quad \widehat{WP}_{FG} = .60, \quad \widehat{WP}_{Punt} = .57$$

* If we have a good WP model, which would arise from XGBoost fit on a dataset with large number of Rows, then

prediction interval

confidence interval $\hat{I}(x) = [\widehat{WP}_L, \widehat{WP}_u]$

$$\widehat{WP}_{go} = .63 \qquad \hat{I}_{go} = [.625, .635]$$

What is a confidence interval?

Imagine we had 100 different training datasets (X, y) each with the same X, but y each time is a new draw from our model $y_i \sim \text{Bernoulli}(WP(x_i))$.

Then a 95% CI on $\hat{y}_i = \widehat{WP}_i(x_i)$ is an interval which contains 95% of the time the WP estimate fit on the dataset.

$$\begin{array}{cccc} (X, y^{(1)}) & (X, y^{(2)}) & \cdots & (X, y^{(100)}) \\ \downarrow & \downarrow & & \downarrow \\ \widehat{WP}^{(1)} & \widehat{WP}^{(2)} & & \widehat{WP}^{(100)} \end{array}$$

at game-state x, 95 of these 100 $\widehat{WP}(x)$ estimates must lie in the 95% confidence interval.

if $\widehat{WP}^{(1)}(x) \geq \widehat{WP}^{(2)}(x) \geq \dots \geq \widehat{WP}^{(100)}(x)$

then 95% CI would be $\left[\widehat{WP}^{(97)}(x), \widehat{WP}^{(3)}(x)\right]$.

What is a prediction interval?

A 95% PI says the true (unseen) outcome y^* lies in the PI 95% of the time.

Out-of-sample dataset $\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ $\begin{pmatrix} y_1^* \\ \vdots \\ y_m^* \end{pmatrix}$

prediction intervals $\begin{pmatrix} PI_1 \\ \vdots \\ PI_m \end{pmatrix}$

then $y_i^* \in PI_i$ in .95 of the rows.

* If we have a good WP model, which would arise from XGBoost fit on a dataset with large number of datapoints, then

confidence interval $\hat{I}(x) = [\widehat{WP}_L, \widehat{WP}_u]$

$\widehat{WP}_{go} = .63$ $\hat{I}_{go} = [.625, .635]$

* If we have a bad WP model, which would arise from XGBoost fit on a dataset with a small number of datapoints, then

$\widehat{WP}_{go} = .63$ $\hat{I}_{go} = [.54, .76]$

yardline
game sec, Rem
point spread
score diff

How can we get confidence intervals on our win probability estimates?

→ Obtaining confidence or prediction intervals for general blackbox machine learning models like XGBoost, Random Forests, or Neural Nets is a fundamental open problem in machine learning today...

→ for some special cases you can do ok.

Process

- Conjectured a way to get CI
- Created a simplified version of football in which the win probability is known and can be explicitly calculated.
- Then we generated a fake historical dataset of football plays which has the same autocorrelated win/loss outcome vector as our real dataset.
- Then we fit XGBoost on this fake historical dataset to estimate WP and get CI
- Then, because this is Simplified Football in which true WP is known, we can simply check if our CI worked.

Simplified Football

- begins at midfield
- each play, the ball moves left or right by 1 yardline with equal probability
- if ball reaches left endzone, team 1 scores TD +1 point
- if ball reaches right endzone, team 2 scores TD −1 point
- ball resets to midfield after each ~~TD~~ TD
- after N plays, game ends
- if tied after N plays, flip coin to determine winner.

How do we generate a fake historical dataset of simplified football plays?

index: n^{th} play of g^{th} game

Outcome play: $\varepsilon_{gn} \overset{iid}{\sim} \pm 1$

game starts at midfield: $X_{g0} = \frac{L}{2}$, L = length of field

game starts tied: $S_{g0} = 0$

field position at start of play n+1 :

$$X_{g,n+1} = \begin{cases} X_{gn} + \xi_{gn} & \text{if prev play not TD} \\ L/2 & \text{if prev play was TD} \end{cases}$$

not a TD $\rightarrow$ $0 < X_{gn} + \xi_{gn} < L$

Score differential at start of play n+1 :

$$S_{g,n+1} = \begin{cases} S_{gn} + 1 & \text{if } X_{gn} + \xi_{gn} = 0 \\ S_{gn} - 1 & \text{if } X_{gn} + \xi_{gn} = L \\ S_{gn} & \text{else} \end{cases}$$

Response column:

identically equal to

$$y_{gn} \equiv y_{g,N+1} = \begin{cases} 1 & \text{if } S_{g,N+1} > 0 \\ 0 & \text{if } S_{g,N+1} < 0 \\ \text{Bernoulli}(\tfrac{1}{2}) & \text{if } S_{g,N+1} = 0 \end{cases}$$

autocorrelation

Fake Historical Dataset

$$\mathcal{D} = \left\{ (n,\ X_{gn},\ S_{gn},\ y_{gn}) : \begin{matrix} g = 1, \dots, G \\ n = 1, \dots, N \end{matrix} \right\}$$

time (game sec. Rem)

field position (yardline)

score diff

win/loss outcome

autocorrelation

$G \approx 4000$

How do we explicity calculate WP?

true win probability

$$WP(n, x, s) = \mathbb{P}\left(S_{g,N+1} > 0 \,\middle|\, X_{gn} = x,\ S_{gn} = s\right)$$

time

field pos

score diff

How to evaluate $WP(n,x,s)$?

$$WP(N+1, x, s) = \begin{cases} 1 & \text{if } s > 0 \\ 1/2 & \text{if } s = 0 \\ 0 & \text{if } s < 0 \end{cases}.$$

dynamic programming

Recursion

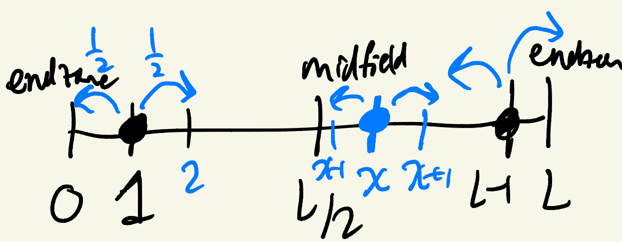
write $WP(n-1, x, s)$ in terms of $WP(n, x, s)$.

$$WP(n-1, x, s) = \begin{cases} \frac{1}{2} \cdot WP(n, x=2, s) + \frac{1}{2} \cdot WP(n, \frac{L}{2}, s+1) & \text{if } x = 1. \\ \frac{1}{2} \cdot WP(n, x = \frac{L}{2}, s-1) + \frac{1}{2} \cdot WP(n, L-2, s) & \text{if } x = L-1. \\ \frac{1}{2} \cdot WP(n, x+1, s) + \frac{1}{2} \cdot WP(n, x-1, s) & \text{else}. \end{cases}$$

- WP of last play (known)
- WP of 2nd to last play known in terms of WP of last play

...

Repeat this logic all the way to the 1st play.



$WP(n,x,s)$ is known.

created simplified footballs

Fake historical dataset $\mathcal{D} = \{(n, X_{gn}, S_{gn}, y_{gn}) : \substack{g=1,\dots,G \\ n=1,\dots,N}\}$

$\widehat{WP} = \mathrm{XGBoost}(\mathcal{D})$

$\widehat{CI} = \mathrm{some\_method}(\mathcal{D})$

We can evaluate how good $\widehat{WP}$ and $\widehat{CI}$ are because $\mathbf{WP}(n,x,s)$ is known!

$\widehat{CI}$ for $\widehat{WP}$

Standard Bootstrap → Resample m Rows of the original observed $T = (X, y)$ with Replacement

B = # bootstrapped outcome vectors

$$
\begin{array}{ccccc}
(X^{(1)}, y^{(1)}) & & (X^{(2)}, y^{(2)}) & \cdots & (X^{(B)}, y^{(B)}) \\
\downarrow & & \downarrow & & \downarrow \\
\widehat{WP}^{(1)} & & \widehat{WP}^{(2)} & & \widehat{WP}^{(100)}
\end{array}
$$

* Intuitively, bootstrapping CI works because

$T = (X, y)$ training dataset

one observed $T \sim$ true underlying WP model process

$b = 1, \ldots, B$ bootstrapped datasets $T^{(b)} \sim T \sim$ true underlying WP model process

Hope: therefore $T^{(b)} \approx$ true underlying WP model process

ex std bootstrap

yardline	point spread	...	score diff	outcome win/loss
70	3		0	1
60	3		0	1
52	3		0	1
		⋮		
40	−3		−7	0

200 plays game 1

200 plays game 2

$$T = (X, y) = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

for i in (1:m) {

sample a row (x_j, y_j) from T, where each row from T has prob $\frac{1}{n}$

}

size $T^{(b)}$

$$T^{(1)} = (X^{(1)}, y^{(1)}) = \begin{bmatrix} x_2 & y_2 \\ x_2 & y_2 \\ x_2 & y_2 \\ x_4 & y_4 \\ x_4 & y_4 \\ x_5 & y_5 \\ x_6 & y_6 \\ x_8 & y_8 \\ x_8 & y_8 \\ & \vdots \end{bmatrix}$$

at game-state x, 95 of these 100 $\widehat{WP}(x)$ estimates must lie in the 95% confidence interval.

if $\widehat{WP}^{(1)}(x) \geq \widehat{WP}^{(2)}(x) \geq \dots \geq \widehat{WP}^{(100)}(x)$

then 95% CI would be $\left[\widehat{WP}^{(97)}(x), \widehat{WP}^{(3)}(x)\right]$.

* the standard bootstrap treats the data generating process as having independent Rows.
By autocorrelation, this is False in our football dataset.
Each outcome y_i = win/loss is the same draw for each play i within the same game.
In other words, the win/loss data is generated once in each game, not separately on each play.

the groups of Rows which are dependent

* Cluster Bootstrap:
Resample clusters (games) rather than just Rows (plays).

* Randomized Cluster Bootstrap:

Resample clusters (games) with replacement and then within each cluster resample rows (plays) with replacement.

↓

best imitates the true data generating process

Fake historical dataset (from simplified football) $\mathcal{D} = \{(n, X_{gn}, S_{gn}, y_{gn}) : \begin{smallmatrix} g=1,\dots,G \\ n=1,\dots,N \end{smallmatrix}\}$

$\widehat{WP} = \text{XGBoost}(\mathcal{D})$

$$\widehat{CI} = \begin{cases} \text{Standard Bootstrap}(\mathcal{D}) & B, m \\ \text{Cluster Bootstrap}(\mathcal{D}) & B, G \\ \text{Randomized Cluster Bootstrap}(\mathcal{D}) & B, G, m' \end{cases}$$

Bootstrap Hyperparameters

- B = # bootstrapped datasets
- m = # rows resampled in each bootstrapped dataset
- G = # games resampled in each bootstrapped dataset
- m' = # plays resampled within each resampled game

Evaluate each bootstrapped CI using

Coverage = what proportion of the time does $\underline{WP} \in CI$.

Evaluate estimates $\widehat{WP}$ using

MAE = mean absolute error

$$= \frac{1}{n}\sum_{i=1}^{n} |WP_i - \widehat{WP}_i|$$

comes from simplified football

games in fake original historical dataset

plays in each game

autocorrelation: # autocorrelated plays per game

standard bootstrap

cluster bootstrap

randomized cluster boot

G	N	K	MAE bt WP and $\widehat{WP}$	~~CI covg. SB~~	~~CI covg. CB~~	CI covg. RCB	~~CI length SB~~	~~CI length CB~~	CI length RCB
4101	53	53	0.0179	~~0.73~~	~~0.85~~	**0.90**	~~[illegible]~~	~~[illegible]~~	**0.079**
[redacted]									

Table 3: Simulation study results. SB means standard bootstrap, CB means cluster bootstrap, and RCB means randomized cluster bootstrap.

$MAE = \frac{1}{n}\sum_{i=1}^{n} |WP_i - \widehat{WP}_i|$

On average, our WP predictions are about 2% off

→ ≈ unbiased

Covg = proportion of time that $WP \in CI$

90% of the time, true WP lies in CI.

$CI = [L, U]$

$length(CI) = U - L$

avg length = 8%

Hyperparameter RCB: B, G, m'

B → 25

G → o.g. # games/2

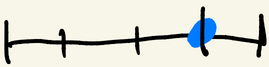
m' → ≈ o.g.

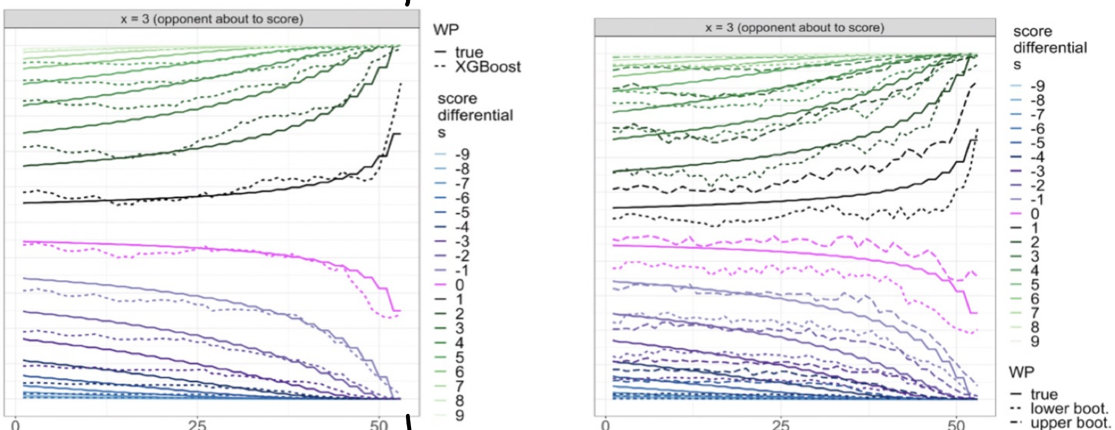
$\widehat{WP} = .63$

CI = [.59, .67]

in order to acheive 90% Coverage with our CI, our CI need to be 8% wide, even in Simplified football !!

We don't <u>really</u> have a sharp idea of WP.



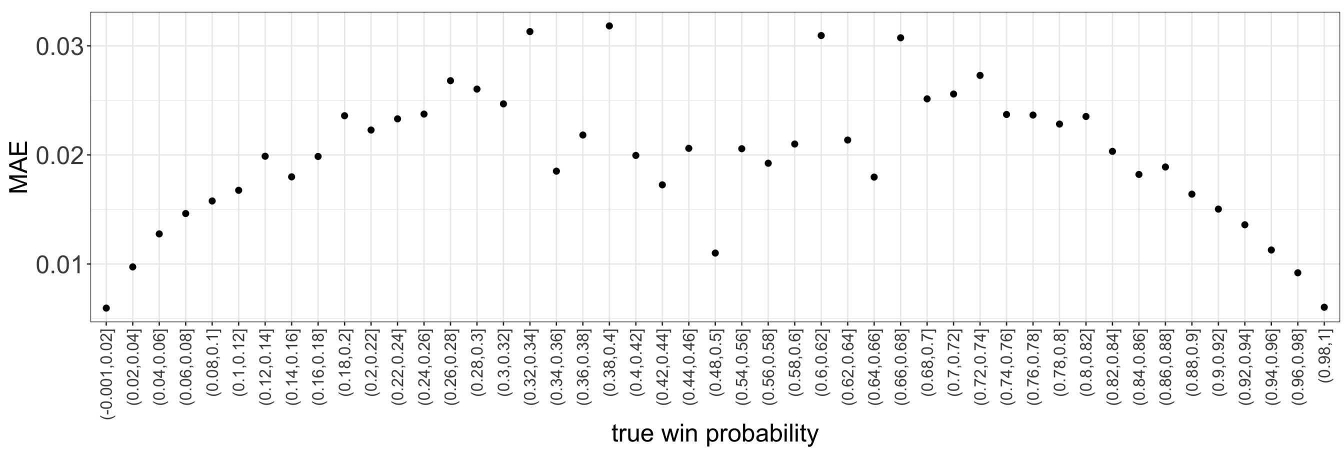


WP

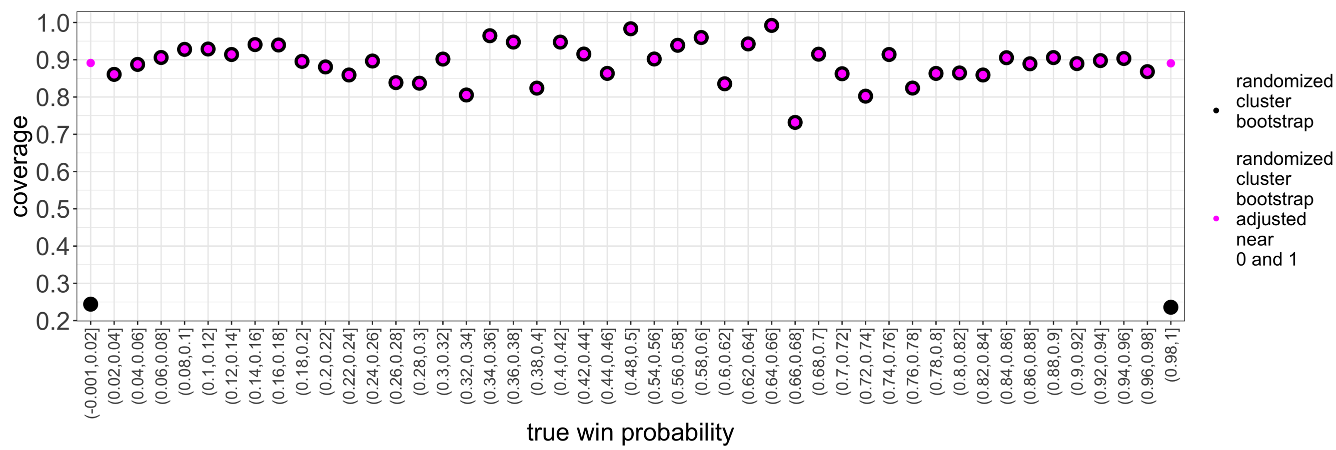
begining time end 53

(a) (b)

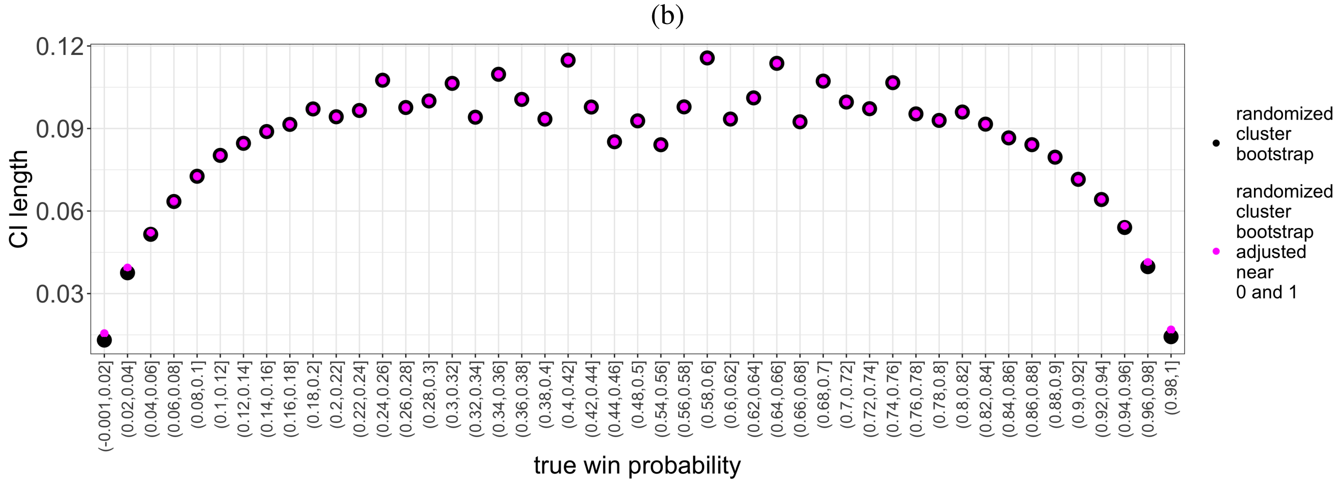
Figure S17: On the left, we visualize the error between estimated WP (dotted line) and true WP (solid line). On the right, we visualize the WP confidence intervals (dotted line) produced by the randomized cluster bootstrap and the true WP (solid line). Both figures display the results from one simulation and at yardline $x = 3$.



(a)



(b)



(c)

Figure S16: As a function of true WP, MAE of true and estimated WP (Figure (a)), coverage of true WP by randomized cluster bootstrap (Figure (b)), and confidence interval length of randomized cluster bootstrap (Figure (c)).

* Use
Randomized Cluster Bootstrap to obtain WP CI with adequate coverage.

* Verified the Kosher-ness of this method using a simplified football simulation in which true WP is known.

* Next time:

1. Apply RCB to real football data
2. Propagate the uncertainty to the 4th down decision itself
3. Use 4th app with uncertainty quantification