Recalibration of Predicted Probabilities Using the "Logit Shift": Why does it work, and when can it be expected to work well?

Evan T. R. Rosenman,† Cory McCartan, & Santiago Olivella

†Department of Mathematical Sciences, Claremont McKenna College

February 21, 2024

My work

Research: Statistical methods for public policy and social science!

- **Political methodology**: elections
- **Causal inference**: data fusion for inference, design
- **Gender-based violence**: prevention science, methodology

Applied elections work

- Data Scientist, Biden for President (2020)
- Data Science Consultant, Civis Analytics + DSCC (2021–2)

Today's Talk

1. **Introduce** the "logit shift," a widely-used heuristic technique for recalibrating probabilities
2. **Characterize** the logit shift via connections to information theory and a Bayesian update procedure
3. **Highlight drawbacks** of the logit shift in practice

Outline

The Recalibration Problem

- **Problem:** calibrating probabilities to observed totals

The Recalibration Problem

- **Problem:** calibrating probabilities to observed totals
- Example: suppose we have scores $p_i \in (0,1)$, $i = 1, \ldots, N$ estimating prob. that each voter supports a Dem candidate
  - After election, observe total Dem votes D, cast by subset $\mathcal{V} \subset \{1, \ldots, N\}$ of registered voters who participated
  - Absent perfect prediction, we will find

$$\sum_{i \in \mathcal{V}} p_i \neq D$$

The Recalibration Problem

- **Problem:** calibrating probabilities to observed totals
- Example: suppose we have scores $p_i \in (0,1)$, $i = 1, \ldots, N$ estimating prob. that each voter supports a Dem candidate
  - After election, observe total Dem votes D, cast by subset $\mathcal{V} \subset \{1, \ldots, N\}$ of registered voters who participated
  - Absent perfect prediction, we will find

$$\sum_{i \in \mathcal{V}} p_i \neq D$$

- How to compute recalibrated scores, $\tilde{p}_i$, incorporating information about the realized electoral outcome?

Recalibration in Practice: Research Problems

Problem manifests in many electoral problems due to **individual-level predictions** but only **aggregated outcome data**

- Ghitza and Gelman (2020) recalibrate MRP estimates of voter support levels to match county-level totals
- Kuriwaki et al. (2022) use a multi-way recalibration to estimate magnitude of racially polarized voting

Recalibration in Practice: Electioneering

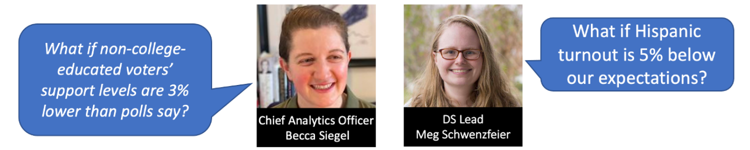
Biden campaign recalibrated *turnout* and *support* scores to simulate electoral scenarios to determine state-level investments.

Goal: robustness to uncertainty.

Recalibration in Practice: Electioneering

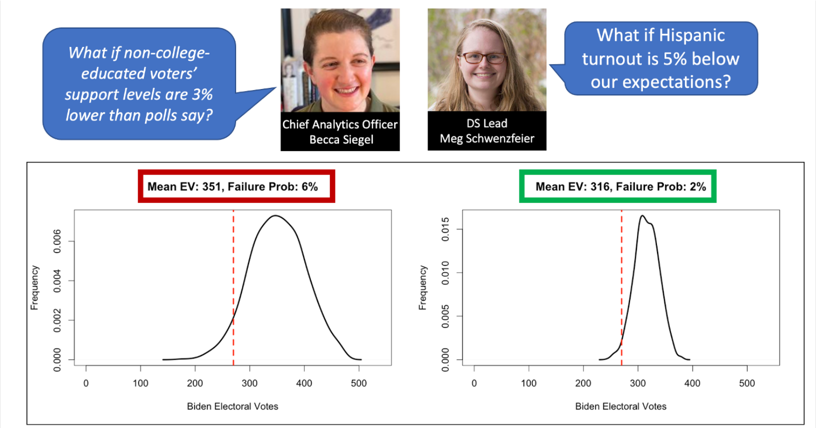
Biden campaign recalibrated *turnout* and *support* scores to simulate electoral scenarios to determine state-level investments.

Goal: robustness to uncertainty.



Recalibration in Practice: Electioneering

Biden campaign recalibrated *turnout* and *support* scores to simulate electoral scenarios to determine state-level investments.
Goal: robustness to uncertainty.



A Heuristic Solution: the Logit Shift!

Intuition: suppose p_i are generated by a logistic regression. Shift model *intercept* until recalibrated scores $\tilde{p}_i$ satisfy

$$\sum_{i \in \mathcal{V}} \tilde{p}_i = D \, .$$

A Heuristic Solution: the Logit Shift!

Intuition: suppose p_i are generated by a logistic regression. Shift model *intercept* until recalibrated scores $\tilde{p}_i$ satisfy

$$\sum_{i \in \mathcal{V}} \tilde{p}_i = D\,.$$

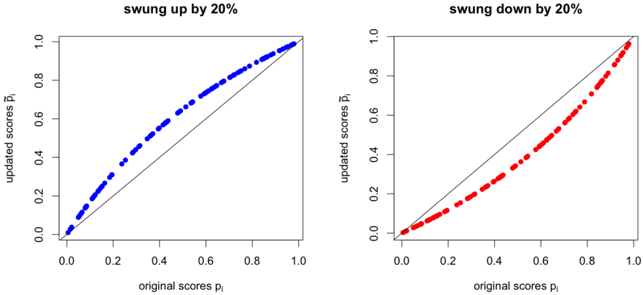
Can be used even if scores p_i not generated by logistic regression.

A Heuristic Solution: the Logit Shift!

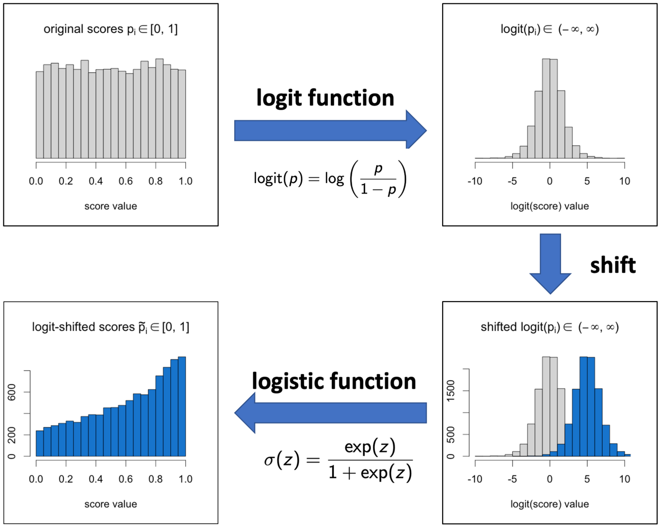
Intuition: suppose p_i are generated by a logistic regression. Shift model *intercept* until recalibrated scores $\tilde{p}_i$ satisfy

$$\sum_{i \in \mathcal{V}} \tilde{p}_i = D \,.$$

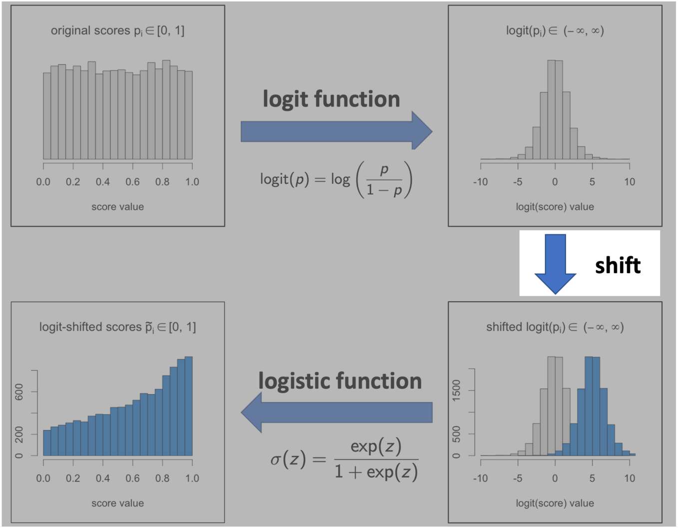
Can be used even if scores p_i not generated by logistic regression.



Computation in Practice (I)



Computation in Practice (II)



Computation in Practice (III)

How much to shift?

- Denote as $f(p, \alpha)$ the "shift" function:

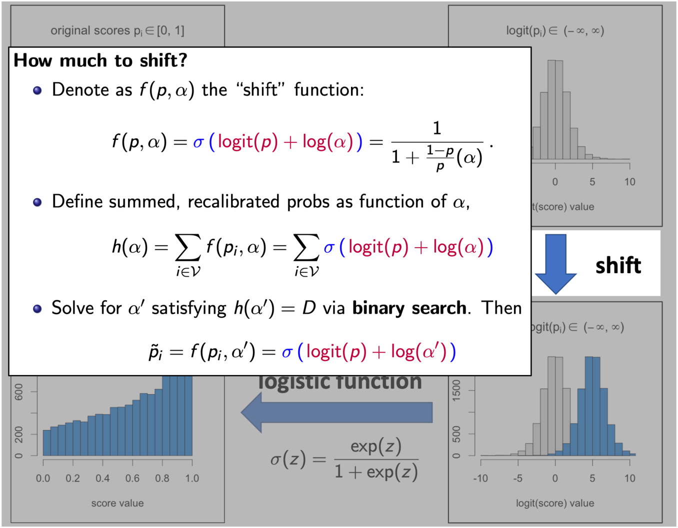
$$f(p,\alpha) = \sigma\left(\text{logit}(p) + \log(\alpha)\right) = \frac{1}{1 + \frac{1-p}{p}(\alpha)}.$$

- Define summed, recalibrated probs as function of α,

$$h(\alpha) = \sum_{i \in \mathcal{V}} f(p_i, \alpha) = \sum_{i \in \mathcal{V}} \sigma\left(\text{logit}(p) + \log(\alpha)\right)$$

- Solve for α' satisfying $h(\alpha') = D$ via **binary search**. Then

$$\tilde{p}_i = f(p_i, \alpha') = \sigma\left(\text{logit}(p) + \log(\alpha')\right)$$



Outline

A simple characterization from information theory

Can derive the logit shift from **information theory.**

A simple characterization from information theory

Can derive the logit shift from **information theory.**

Logit-shifted $\{\tilde{p}_i\}$ solve the problem

$$\begin{aligned} \text{minimize} \quad & \sum_i \mathcal{D}_{KL}\left(\tilde{p}_i \,||\, p_i\right) \\ \text{subject to} \quad & \sum_i \tilde{p}_i = D, \end{aligned}$$

i.e. minimize the summed **KL divergence** with the original $\{p_i\}$ among all sets of probabilities summing to D.

Posterior update procedure

Define $W_i \in \{0, 1\}$ as vote choice ($1 =$ Dem, $0 =$ Rep).

Model $W_i \sim \text{Bern}(p_i)$ where p_i are prior Dem support probability.

Posterior update procedure

Define $W_i \in \{0, 1\}$ as vote choice ($1 =$ Dem, $0 =$ Rep).

Model $W_i \sim \text{Bern}(p_i)$ where p_i are prior Dem support probability.
Posterior update conditional on observed votes D:

$$p_i^\star = \mathbb{P}\left(W_i = 1 \,\middle|\, \sum_{j \in \mathcal{V}} W_j = D\right) = \frac{\mathbb{P}\left(W_i = 1, \sum_{j \in \mathcal{V}} W_j = D\right)}{\mathbb{P}\left(\sum_{j \in \mathcal{V}} W_j = D\right)}$$

$$= p_i \times \frac{\mathbb{P}\left(\sum_{j \neq i} W_j = D - 1\right)}{\mathbb{P}\left(\sum_{j \in \mathcal{V}} W_j = D\right)}.$$

Posterior update procedure

Define $W_i \in \{0,1\}$ as vote choice ($1 = \text{Dem}$, $0 = \text{Rep}$).

Model $W_i \sim \text{Bern}(p_i)$ where p_i are prior Dem support probability. Posterior update conditional on observed votes D:

$$p_i^\star = \mathbb{P}\left(W_i = 1 \,\middle|\, \sum_{j\in\mathcal{V}} W_j = D\right) = \frac{\mathbb{P}\left(W_i = 1, \sum_{j\in\mathcal{V}} W_j = D\right)}{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D\right)}$$

$$= p_i \times \frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)}{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D\right)}.$$

Observe $\sum_{j\in\mathcal{V}} p_i^\star = D$ automatically, because

$$\sum_{j\in\mathcal{V}} p_i^\star = \sum_{j\in\mathcal{V}} \mathbb{E}\left(W_i \,\middle|\, \sum_{j\in\mathcal{V}} W_j = D\right) = D.$$

Posterior update procedure: the problem

Define $W_i \in \{0, 1\}$ as vote choice ($1 =$ Dem, $0 =$ Rep).

Model $W_i \sim \text{Bern}(p_i)$ where p_i are prior Dem support probability.
Posterior update conditional on observed votes D:

$$p_i^\star = \mathbb{P}\left(W_i = 1 \,\middle|\, \sum_{j \in \mathcal{V}} W_j = D\right) = \frac{\mathbb{P}\left(W_i = 1, \sum_{j \in \mathcal{V}} W_j = D\right)}{\mathbb{P}\left(\sum_{j \in \mathcal{V}} W_j = D\right)}$$

$$= p_i \times \underbrace{\frac{\mathbb{P}\left(\sum_{j \neq i} W_j = D - 1\right)}{\mathbb{P}\left(\sum_{j \in \mathcal{V}} W_j = D\right)}}_{\text{Poisson-Binomial probabilities} \Rightarrow \text{uh oh!}} .$$

Poisson-Binomial Distribution

A **Poisson-Binomial** random variable is the sum of independent *but not identically distributed* Bernoulli random variables.

Poisson-Binomial Distribution

A **Poisson-Binomial** random variable is the sum of independent *but not identically distributed* Bernoulli random variables.

PMF now involves **combinatoric sums** $\Rightarrow$
Despite recent advances (Olivella and Shiraito, 2017; Junge, 2020), still computationally demanding to compute.

Poisson-Binomial Distribution

A **Poisson-Binomial** random variable is the sum of independent *but not identically distributed* Bernoulli random variables.

PMF now involves **combinatoric sums** $\Rightarrow$
Despite recent advances (Olivella and Shiraito, 2017; Junge, 2020), still computationally demanding to compute.

Implication: not feasible to compute $p_i^\star$ at even modest sample sizes (e.g. tens of precincts).

Main Results

Posterior updated probabilities $\{p_i^\star\}$ not computable in practice $\Rightarrow$

But logit shifted scores $\{\tilde{p}_i\}$ are a good approximation!

Theorem (Error Bounds)

For large sample sizes, we obtain

$$\tilde{p}_i = p_i^\star + \mathcal{O}\left(\frac{1}{|\mathcal{V}|}\right).$$

Proof Sketch (I): Preliminaries

Recall the shift function $f(p, \alpha)$.
Adjusts a score p by $\log(\alpha)$ on the logit scale:

$$f(p, \alpha) = \sigma\left(\text{logit}(p) + \log(\alpha)\right) = \frac{1}{1 + \frac{1-p}{p}(\alpha)}.$$

Define also the *unit-specific* Poisson-Binomial ratio:

$$\phi_i = \frac{\mathbb{P}\left(\sum_{j \neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j \neq i} W_j = D - 1\right)}.$$

Proof Sketch (II): Taking p_i to $p_i^\star$ using a shift by ϕ_i

$$
\begin{aligned}
f(p_i,\phi_i) &= \frac{1}{1+\frac{1-p_i}{p_i}\phi_i} = \frac{1}{1+\frac{1-p_i}{p_i}\frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)}} \\
&= \frac{p_i \times \mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)}{p_i \times \mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right) + (1-p_i)\times \mathbb{P}\left(\sum_{j\neq i} W_j = D\right)} \\
&= \frac{\mathbb{P}\left(W_i = 1, \sum_{i\in\mathcal{V}} W_i = D\right)}{\mathbb{P}\left(\sum_{i\in\mathcal{V}} W_i = D\right)} = p_i^\star
\end{aligned}
$$

where on the last line we have used the recursion

$$
\mathbb{P}\left(\sum_j W_j = D\right) = p_i \times \mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right) + (1-p_i)\times \mathbb{P}\left(\sum_{j\neq i} W_j = D\right).
$$

Proof Sketch (III): Final Steps

We have shown: ϕ_i is the *unit-specific* adjustment that turns each p_i into the desired $p_i^\star$:

$$f(p_i, \phi_i) = p_i^\star.$$

Proof Sketch (III): Final Steps

We have shown: ϕ_i is the *unit-specific* adjustment that turns each p_i into the desired $p_i^\star$:

$$f(p_i, \phi_i) = p_i^\star.$$

Recall: the logit shift uses a *precinct-specific* adjustment α to update the probabilities via

$$f(p_i, \alpha) = \tilde{p}_i.$$

Proof Sketch (III): Final Steps

We have shown: ϕ_i is the *unit-specific* adjustment that turns each p_i into the desired $p_i^\star$:

$$f(p_i, \phi_i) = p_i^\star.$$

Recall: the logit shift uses a *precinct-specific* adjustment α to update the probabilities via

$$f(p_i, \alpha) = \tilde{p}_i.$$

Final proof step: show that the α used in the logit shift, found by solving $\sum_{i \in \mathcal{V}} f(p_i, \alpha) = D$, satisfies

$$\alpha \approx \phi_i \quad \text{for all values of } i.$$

Simulations: Probability Distributions

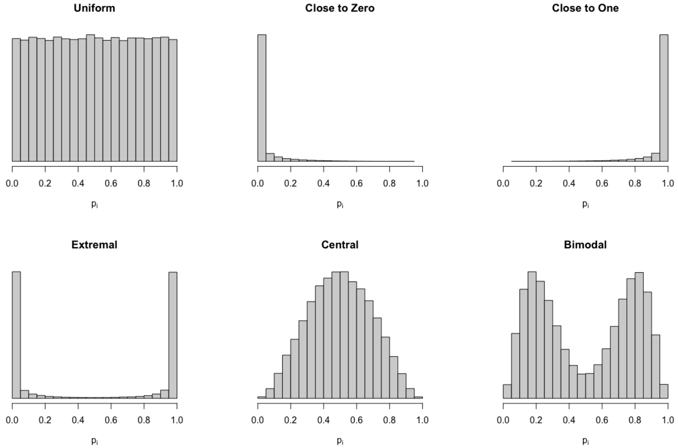
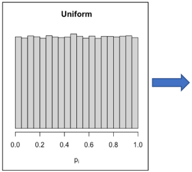


Figure 1: The distributions used for sampling p_i in simulations. Drawn from Biscarri et al. (2018).

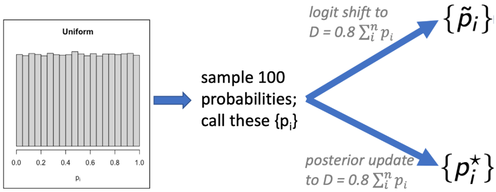
Simulations: Procedure



sample 100 probabilities; call these $\{p_i\}$

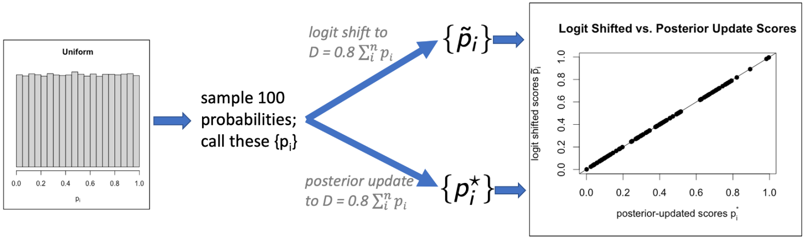
Sample the initial scores p_i from the distribution.

Simulations: Procedure



Compute both the logit shifted scores $\{\tilde{p}_i\}$ and the posterior updated scores $\{p_i^\star\}$, with $D = 0.8 \times \sum_i p_i$.

Simulations: Procedure



Compare the logit shifted scores $\{\tilde{p}_i\}$ vs. the posterior updated scores $\{p_i^\star\}$.

Simulation Results

p_i Setting	Sample Size	$1 - R^2$
Uniform	100	5.81×10^{-5}
Uniform	1000	5.51×10^{-7}
Close to 0	100	1.06×10^{-2}
Close to 0	1000	3.11×10^{-5}
Close to 1	100	1.12×10^{-4}
Close to 1	1000	1.12×10^{-6}
Extremal	100	1.16×10^{-6}
Extremal	1000	1.19×10^{-6}
Central	100	7.66×10^{-5}
Central	1000	7.16×10^{-7}
Bimodal	100	6.72×10^{-5}
Bimodal	1000	6.77×10^{-7}

Table 1: Discrepancy between between logit shift and exact Poisson-Binomial probabilities. Observed D is equal to $0.8 \times \sum_i p_i$.

Outline

Outline

Heterogeneous Groups

Logit shift is **rank-preserving** $\Rightarrow$

But what if errors vary by subgroup?

Heterogeneous Groups

Logit shift is **rank-preserving** $\Rightarrow$

But what if errors vary by subgroup?

Simple example:

- Suppose two groups of voters: Black and White

Heterogeneous Groups

Logit shift is **rank-preserving** $\Rightarrow$

But what if errors vary by subgroup?

Simple example:

- Suppose two groups of voters: Black and White
  - Black voters: $p_i = 0.7$ but $p_i^{\text{true}} = 0.8$
  - White voters: $p_i = 0.3$ but $p_i^{\text{true}} = 0.2$

Heterogeneous Groups

Logit shift is **rank-preserving** $\Rightarrow$

But what if errors vary by subgroup?

Simple example:

- Suppose two groups of voters: Black and White
  - Black voters: $p_i = 0.7$ but $p_i^{\text{true}} = 0.8$
  - White voters: $p_i = 0.3$ but $p_i^{\text{true}} = 0.2$
- Run logit shift; suppose population is heavily White

Heterogeneous Groups

Logit shift is **rank-preserving** $\Rightarrow$

But what if errors vary by subgroup?

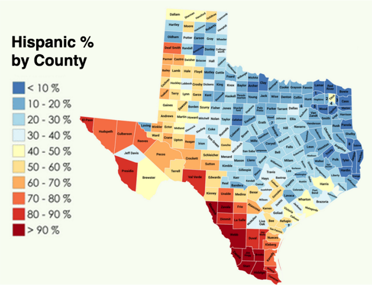
Simple example:

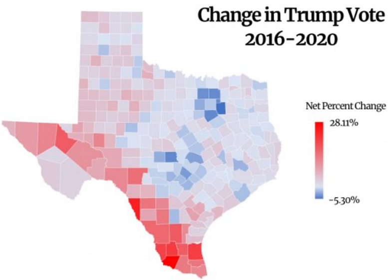
- Suppose two groups of voters: Black and White
  - Black voters: $p_i = 0.7$ but $p_i^{\text{true}} = 0.8$
  - White voters: $p_i = 0.3$ but $p_i^{\text{true}} = 0.2$
- Run logit shift; suppose population is heavily White
- This will yield a big downward adjustment to the scores $\Rightarrow$
  - $\tilde{p}_i$ more accurate for White voters, but...
  - less accurate for Black voters!

Implications in Practice (I)

- Logit shift is most performant if conducted in groupings with more homogeneous voters
- Recommend conducting logit shift at finest available level of aggregation (e.g. voting precincts)
    - Populations typically more homogenous at finer aggregations.
    - 2020 Census data on race/ethnicity: 39.8% of voting-age population was in the minority nationwide, but only 12.9% within Census block (U.S. Census Bureau, 2021).

Implications in Practice (II)





If subgroup-specific prediction errors are highly variable, may need a richer model than the logit shift!

Outline

Same Total, Different Distributions

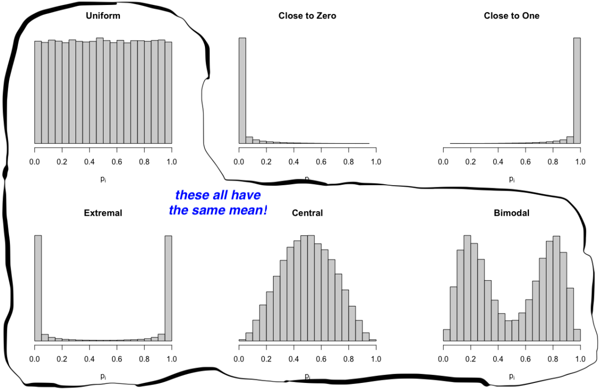
Recall: can only observe how the original predictions p_i differ from true p_i^{true} via discrepancy in their sum

But the distribution carries much more info than its mean!

Same Total, Different Distributions

Recall: can only observe how the original predictions p_i differ from true p_i^{true} via discrepancy in their sum

But the distribution carries much more info than its mean!



Simulation Results

- Sample $1,000$ voters such that p_i^{true} and p_i follow each possible pair of distributions among the 36 pairs.
- Sample the outcomes; conduct logit shift to obtain $\tilde{p}_i$. Report correlational improvement: $\frac{\text{cor}(\tilde{p}_i, p_i^{\text{true}}) - \text{cor}(p_i, p_i^{\text{true}})}{\text{cor}(p_i, p_i^{\text{true}})}$.

Simulation Results

- Sample $1,000$ voters such that p_i^{true} and p_i follow each possible pair of distributions among the 36 pairs.
- Sample the outcomes; conduct logit shift to obtain $\tilde{p}_i$. Report correlational improvement: $\frac{\text{cor}(\tilde{p}_i, p_i^{\text{true}}) - \text{cor}(p_i, p_i^{\text{true}})}{\text{cor}(p_i, p_i^{\text{true}})}$.

		Initial Prediction Distribution (p_i)					
		Unif.	**≈ 0**	**≈ 1**	**Extremal**	**Central**	**Bimodal**
True Distr. (p_i^{true})	**Unif.**	*0.00*	0.65	0.84	*0.00*	*0.00*	*0.00*
	≈ 0	0.65	*0.00*	4.61	1.68	0.46	0.89
	≈ 1	0.68	2.36	*0.00*	1.46	0.46	0.91
	Extremal	*0.00*	1.15	1.13	*0.00*	*0.00*	*0.00*
	Central	*0.00*	0.55	0.49	*0.00*	*0.00*	*0.00*
	Bimodal	*0.00*	0.72	0.84	*0.00*	*0.00*	*0.00*

Outline

Recap

- Logit shift is a simple algorithm for updating predicted probabilities p_i to match an aggregate total D
- Closely approximates a (computationally intractable) posterior probability that conditions on the total D
- Limitations include:
  - Inability to reorder predictions $\Rightarrow$ inappropriate for subgroup-specific errors under heterogeneity
  - Cannot correct incorrect shape of the distribution

Future Work

Natural extension is to consider a more expressive update model that includes covariates, e.g.

$$\tilde{p}_i = \frac{1}{1 + \frac{1-p_i}{p_i} \exp(\beta^T X_i)}.$$

Need to learn coefficient β from the data. Achievable via:

- Approximating Poisson-Binomial likelihood with a Gaussian (Siripraparat and Neammanee, 2021).
- Training model via gradient descent

Future Work

Natural extension is to consider a more expressive update model that includes covariates, e.g.

$$\tilde{p}_i = \frac{1}{1 + \frac{1-p_i}{p_i} \exp(\beta^T X_i)}.$$

Need to learn coefficient β from the data. Achievable via:

- Approximating Poisson-Binomial likelihood with a Gaussian (Siripraparat and Neammanee, 2021).
- Training model via gradient descent

Principled way to update the model to account for subgroup-specific shifts

Thanks!

Thanks to my co-authors, Cory and Santiago!

Full paper available now in **Political Analysis**.

References (I)

Biscarri, W., Zhao, S. D., and Brunner, R. J. (2018). A simple and fast method for computing the poisson binomial distribution function. *Computational Statistics & Data Analysis*, 122:92–100.

Ghitza, Y. and Gelman, A. (2020). Voter registration databases and mrp: Toward the use of large-scale databases in public opinion research. *Political Analysis*, 28(4):507–531.

Huang, Y. and Pepe, M. S. (2009). Semiparametric methods for evaluating risk prediction markers in case-control studies. *Biometrika*, 96(4):991–997.

Junge, F. (2020). Package 'poissonbinomial'. *Computational Statistics & Data Analysis*, 59:41–51.

Kuriwaki, S., Ansolabehere, S., Dagonel, A., and Yamauchi, S. (2022). The geography of racially polarized voting: Calibrating surveys at the district level. *OSF Preprints*.

Olivella, S. and Shiraito, Y. (2017). poisbinom: A faster implementation of the poisson–binomial distribution. r package version 1.0. 1.

Pencina, M. J., D'Agostino Sr, R. B., and Steyerberg, E. W. (2011). Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Statistics in medicine*, 30(1):11–21.

Siripraparat, T. and Neammanee, K. (2021). A local limit theorem for Poisson Binomial random variables. *Sci. Asia. https://doi. org/10.2306/scienceasia1513-1874.2021*, 6.

U.S. Census Bureau (2021). 2020 census. U.S. Department of Commerce.

Simulation Results

- Simulate $n = 1,000$ voters, 80% White and 20% Black
- Draw p_i from each distribution, but p_i^{true} are 10% higher for White voters and 10% lower for Black voters
- Sample the outcomes; conduct logit shift to obtain $\tilde{p}_i$. Report correlational improvement value, $\frac{\text{cor}(\tilde{p}_i, p_i^{\text{true}}) - \text{cor}(p_i, p_i^{\text{true}})}{\text{cor}(p_i, p_i^{\text{true}})}$.

Simulation Results

- Simulate $n = 1,000$ voters, 80% White and 20% Black
- Draw p_i from each distribution, but p_i^{true} are 10% higher for White voters and 10% lower for Black voters
- Sample the outcomes; conduct logit shift to obtain $\tilde{p}_i$. Report correlational improvement value, $\frac{\text{cor}(\tilde{p}_i, p_i^{\text{true}}) - \text{cor}(p_i, p_i^{\text{true}})}{\text{cor}(p_i, p_i^{\text{true}})}$.

Initial Score Dist	White	Black	Overall
Uniform	0.01	-0.02	0.01
Close to 0	0.13	-0.11	0.04
Close to 1	0.07	-0.18	0.04
Extremal	0.08	-0.12	0.04
Central	0.00	-0.01	0.00
Bimodal	0.00	-0.01	0.00

Logit Shift Computation

- Define $\alpha \in [0, \infty)$ s.t. its log is equal to the intercept shift,

$$\operatorname{logit}(\tilde{p}_i) = \operatorname{logit}(p_i) - \log(\alpha)\,,$$

where $\operatorname{logit}(z) = \log(z/(1-z))$.

Logit Shift Computation

- Define $\alpha \in [0, \infty)$ s.t. its log is equal to the intercept shift,

$$\text{logit}(\tilde{p}_i) = \text{logit}(p_i) - \log(\alpha)\,,$$

  where $\text{logit}(z) = \log(z/(1-z))$.
- Define summed, recalibrated probabilities as function of α,

$$h(\alpha) = \sum_{i \in \mathcal{V}} \tilde{p}_i = \sum_{i \in \mathcal{V}} \sigma\left(\text{logit}(p_i) - \log(\alpha)\right)$$

  where $\sigma(z) = \exp(z)/(1 + \exp(z))$

Logit Shift Computation

- Define $\alpha \in [0, \infty)$ s.t. its log is equal to the intercept shift,

$$\text{logit}(\tilde{p}_i) = \text{logit}(p_i) - \log(\alpha)\,,$$

  where $\text{logit}(z) = \log(z/(1-z))$.

- Define summed, recalibrated probabilities as function of α,

$$h(\alpha) = \sum_{i \in \mathcal{V}} \tilde{p}_i = \sum_{i \in \mathcal{V}} \sigma\left(\text{logit}(p_i) - \log(\alpha)\right)$$

  where $\sigma(z) = \exp(z)/(1 + \exp(z))$

- Solve for α' satisfying $h(\alpha') = D$ via binary search. Then

$$\tilde{p}_i = \sigma\left(\text{logit}(p_i) - \log(\alpha')\right)$$

Proof Sketch (I): Preliminaries

Recall that $f(p, \alpha)$ shifts a score p by $\log(\alpha)$ on the logit scale:

$$f(p, alpha) = \sigma\left(\text{logit}(p) + \log(\alpha)\right) = \frac{1}{1 + \frac{1-p}{p}(\alpha)}.$$

Define also the *unit-specific* quantity

$$\phi_i = \frac{\mathbb{P}\left(\sum_{j \neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j \neq i} W_j = D - 1\right)}.$$

Proof Sketch (II): Taking p_i to $p_i^\star$ using ϕ_i

$$f(p_i,\phi_i) = \frac{1}{1+\frac{1-p_i}{p_i}\phi_i} = \frac{1}{1+\frac{1-p_i}{p_i}\frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)}}$$

$$= \frac{p_i \times \mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)}{p_i \times \mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right) + (1-p_i)\times \mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}$$

$$= \frac{\mathbb{P}\left(W_i = 1, \sum_{i\in\mathcal{V}} W_i = D\right)}{\mathbb{P}\left(\sum_{i\in\mathcal{V}} W_i = D\right)} = p_i^\star .$$

Idea: ϕ_i is precisely the (unit-specific) adjustment that turns each p_i into the desired $p_i^\star$ using the function f.
The logit shift uses a constant α to approximate each entry in the vector of unit-specific adjustments $\{\phi_i\}_{i\in\mathcal{V}}$.

Proof Sketch (III): Helpful Poisson-Binomial Properties

TODO: Show that the single value of α used by the logit shift is a very good approximation of ϕ_i for all values of i.

Proof Sketch (III): Helpful Poisson-Binomial Properties

TODO: Show that the single value of α used by the logit shift is a very good approximation of ϕ_i for all values of i.

Theorem (Poisson-Binomial Properties)

The value of α used by the logit shift satisfies:

$$\min_i \frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)} \leq \alpha \leq \max_i \frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)}.$$

Moreover, for any choice of $i \in \mathcal{V}$, we have

$$\frac{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D+1\right)}{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D\right)} \leq \frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)} \leq \frac{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D\right)}{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D-1\right)}.$$

Proof Sketch (IV): Combining Bounds Approximating

We can combine the two prior results to observe

$$\frac{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D+1\right)}{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D\right)} \leq \min_i \frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)} \leq \alpha$$

$$\leq \max_i \frac{\mathbb{P}\left(\sum_{j\neq i} W_j = D\right)}{\mathbb{P}\left(\sum_{j\neq i} W_j = D-1\right)} \leq \frac{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D\right)}{\mathbb{P}\left(\sum_{j\in\mathcal{V}} W_j = D-1\right)}.$$

Lastly, apply normal approximation bounds to the outermost Poisson-Binomial expressions (Siripraparat and Neammanee, 2021) to obtain the result.

Simulation Results

p_i Setting	Sample Size	RMSE	$1 - R^2$	KLD
Uniform	100	0.00195	5.81×10^{-5}	1.21×10^{-3}
Uniform	1000	0.00021	5.51×10^{-7}	1.43×10^{-4}
Close to 0	100	0.00772	1.06×10^{-2}	1.68×10^{-2}
Close to 0	1000	0.00043	3.11×10^{-5}	5.24×10^{-4}
Close to 1	100	0.00369	1.12×10^{-4}	6.38×10^{-3}
Close to 1	1000	0.00034	1.12×10^{-6}	5.13×10^{-4}
Extremal	100	0.00496	1.16×10^{-6}	1.22×10^{-2}
Extremal	1000	0.00050	1.19×10^{-6}	1.05×10^{-3}
Central	100	0.00161	7.66×10^{-5}	7.04×10^{-4}
Central	1000	0.00016	7.16×10^{-7}	6.63×10^{-5}
Bimodal	100	0.00227	6.72×10^{-5}	1.74×10^{-3}
Bimodal	1000	0.00023	6.77×10^{-7}	1.93×10^{-4}

Table 2: Discrepancy between between logit shift and exact Poisson-Binomial probabilities. Observed D is equal to $0.8 \times \sum_i p_i$.

Logit Shift Computation (I)

- Recall the function inverses:

$$\mathrm{logit}(p) = \log\left(\frac{p}{1-p}\right) \qquad \text{“logit”}: \; p \in (0,1) \Rightarrow (-\infty, \infty)$$

$$\sigma(z) = \frac{\exp(z)}{1+\exp(z)} \qquad \text{“logistic”}: \; z \in (-\infty, \infty) \Rightarrow (0,1)$$

Logit Shift Computation (I)

- Recall the function inverses:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \qquad \text{“logit”}:\ p \in (0,1) \Rightarrow (-\infty, \infty)$$

$$\sigma(z) = \frac{\exp(z)}{1+\exp(z)} \qquad \text{“logistic”}:\ z \in (-\infty, \infty) \Rightarrow (0,1)$$

- Denote as $f(p, \alpha)$ the “shift” function:

$$f(p,\alpha) = \sigma\left(\text{logit}(p) + \log(\alpha)\right) = \frac{1}{1 + \frac{1-p}{p}(\alpha)}.$$

Takes in in a probability, returns“shifted” probability adjusted by by $\log(\alpha)$ on the logit scale.

Logit Shift Computation (II)

- Define summed, recalibrated probabilities as function of α,

$$h(\alpha) = \sum_{i \in \mathcal{V}} f(p_i, \alpha) = \sum_{i \in \mathcal{V}} \sigma\left(\text{logit}(p_i) - \log(\alpha)\right)$$

Logit Shift Computation (II)

- Define summed, recalibrated probabilities as function of α,

$$h(\alpha) = \sum_{i \in \mathcal{V}} f(p_i, \alpha) = \sum_{i \in \mathcal{V}} \sigma\left(\text{logit}(p_i) - \log(\alpha)\right)$$

- Solve for α' satisfying $h(\alpha') = D$ via binary search. Then

$$\tilde{p}_i = f(p_i, \alpha') = \sigma\left(\text{logit}(p_i) - \log(\alpha')\right).$$

Note: there is a **single value** for shift parameter α corresponding to all individuals in a precinct.

How much to shift?

- Denote as $f(p, \alpha)$ the "shift" function:

$$f(p,\alpha) = \sigma\left(\text{logit}(p) + \log(\alpha)\right) = \frac{1}{1 + \frac{1-p}{p}(\alpha)}.$$

- Define summed, recalibrated probs as function of α,

$$h(\alpha) = \sum_{i \in \mathcal{V}} f(p_i, \alpha) = \sum_{i \in \mathcal{V}} \sigma\left(\text{logit}(p) + \log(\alpha)\right)$$

- Solve for α' satisfying $h(\alpha') = D$ via **binary search**. Then

$$\tilde{p}_i = f(p_i, \alpha') = \sigma\left(\text{logit}(p) + \log(\alpha')\right)$$

Notation

Symbol	Definition
p_i	prior probability
$\tilde{p}_i$	logit-shifted probability
$p_i^\star$	posterior-updated probability
p_i^{true}	true probability

Notation

Symbol	Definition
p_i	prior probability
$\tilde{p}_i$	logit-shifted probability
$p_i^\star$	posterior-updated probability
p_i^{true}	true probability

Suppose true sampling model is $W_i \sim \text{Bern}(p_i^{\text{true}})$ independently.