Debiasing Evaluations That Are Biased by Evaluations

Jingyan Wang
Carnegie Mellon/Georgia Tech

“Debiasing Evaluations That Are Biased by Evaluations”
Jingyan Wang, Ivan Stelmakh, Yuting Wei, Nihar B. Shah
AAAI 2021
Motivation 1: teaching evaluation

- Students are asked to rate instructors’ teaching effectiveness
- Correlation between ratings vs. teaching quality can be negative
  [Carrell & West, 2008; Braga et al., 2014; Boring et al., 2016]
- Highly biased by grading leniency:

  “...the effects of grades on teacher–course evaluations are both substantively and statistically important...”

  [Johnson, 2003]
Motivation 2: peer review

• Authors are asked to rate the reviews they receive
• Highly biased by positiveness of reviews: [Weber et al., 2002; Papagiannaki, 2007; Khosla, 2013]

“Satisfaction [of the author with the review] had a strong, positive association with acceptance of the manuscript for publication... Quality of the review of the manuscript was not associated with author satisfaction.”

[Weber et al., 2002]
High-level problem

Unfair for rigorous and strict instructors

This work: correct experience-induced bias
Introduce incentives for inflating grades, reducing content, "teaching to test" etc.

"... instructors can often double their odds of receiving high evaluations from students simply by awarding A’s rather than B’s or C’s." [Johnson, 2003]

This work: Correcting experience-induced bias reduces such incentives.
Problem formulation

- $n$ courses to evaluate: unknown true quality $x_i^*$ for $i \in [n]$  
- $d$ students per course  
- Student $j \in [d]$ in course $i \in [n]$ gives ratings:  
\[
y_{ij} = x_i^* + \text{bias} + \text{noise}
\]

- **Noise**: iid zero-mean normal  
- **Bias**: marginally distributed as normal

The observed experience gives structural information about the bias  
- Higher grades $\rightarrow$ better ratings
Problem formulation

Example 1: total ordering of grades

\[ n = 2, \; d = 3 \]

Course 1 \( (x_1^*) \)

\[
\begin{array}{c}
\text{Score} \\
\begin{array}{ccc}
90 & 85 & 60 \\
\end{array}
\end{array}
\]

Course 2 \( (x_2^*) \)

\[
\begin{array}{c}
\text{Score} \\
\begin{array}{ccc}
95 & 80 & 70 \\
\end{array}
\end{array}
\]

Bias: \( b_{95} \geq b_{90} \geq b_{85} \geq b_{80} \geq b_{70} \geq b_{60} \)
Problem formulation

Example 2: partial ordering of grades

\[ n = 2, \quad d = 6 \]

Course 1 (\( x_1^* \))

Course 2 (\( x_2^* \))

Bias:

\[ b_A \geq b_B \geq b_C \]

Ratings: \[ Y = x_1^T + B + \text{noise} \]

Goal: estimate \( x^* \) (given \( Y \) and ordering)
Proposed estimator

\[ \hat{x}^{(\lambda)} \in \arg\min_{x \in \mathbb{R}^n} \min_{B \text{ obeys ordering}} \| Y - x1^T - B \|_F^2 + \lambda \|B\|_F^2 \]

- Difference between raw ratings \( y \) vs. experience-corrected ratings \( x + b \)
- Regularization on magnitude of \( b \)

- Analyze two extremal cases: \( \lambda = 0 \) and \( \lambda = \infty \)
- Choose \( \lambda \) based on the data
Extremal case 1: $\lambda = 0$

$$\hat{x}^{(\lambda)} \in \operatorname{argmin}_{x \in \mathbb{R}^n} \min_{B \text{ obeys ordering}} \|Y - x1^T - B\|_F^2 + \lambda \|B\|_F^2$$

- No regularization means we “explain” the ratings as much as possible by $B$
- Closed-form solution
Extremal case 1: $\lambda = 0$

$$\hat{x}^{(\lambda)} \in \arg\min_{x \in \mathbb{R}^n} \min_{B \text{ obeys ordering}} \|Y - x1^T - B\|_F^2 + \lambda \|B\|_F^2$$

- No regularization means we “explain” the ratings as much as possible by $B$
- Closed-form solution
- Works well when there is no/little noise

Theorem 1 (informal). Our estimator (with $\lambda = 0$) is consistent when there is no noise.

- Sample mean is not consistent
Extremal case 2: $\lambda \to \infty$

$$\hat{x}(\lambda) \in \text{argmin}_{x \in \mathbb{R}^n} \min_{B \text{ obeys ordering}} \|Y - x1^T - B\|_F^2 + \lambda\|B\|_F^2$$

- $B \approx 0$
- $\hat{x}^{(\infty)} \approx \text{argmin}_{x \in \mathbb{R}^n} \|Y - x1^T\|_F^2 = \text{taking sample mean}$
- Formally, define $\hat{x}^{(\infty)} = \lim_{\lambda \to \infty} \hat{x}(\lambda)$

**Theorem 2.** $\hat{x}^{(\infty)}$ is equivalent to taking the sample mean.

- Our class of estimators includes one of the most commonly-used methods
- **Minimax optimal when there is no bias.** [Wainwright 2019]
Choosing $\lambda$

- $\lambda = 0$ and $\lambda = \infty$ work well respectively when there is no noise and no bias.

**Challenge:** don’t know the amount of bias vs. noise

**Idea:** carefully design a cross-validation algorithm to choose $\lambda$
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way

\[
\begin{align*}
Y_{\text{train}} & \quad \quad \quad Y_{\text{val}}
\end{align*}
\]
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)

\[
\|Y_{\text{val}} - \hat{x}_{\text{train}}^T - \hat{B}_{\text{train}}\|_{\text{val}}^2
\]

**Challenge:** different bias on different individuals 😞
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}},Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)

\[
\text{estimator} (\lambda) \quad \rightarrow \quad (\hat{x}_{\text{train}}, \hat{B}_{\text{train}}) \quad \rightarrow \quad \hat{B}_{\text{val}}
\]

Interpolate \(\hat{B}_{\text{val}}\) using \((\hat{B}_{\text{train}},\text{ordering})\)

\[
\text{error} \quad \|Y_{\text{val}} - \hat{x}_{\text{train}}1^T - \hat{B}_{\text{val}}\|^2_{\text{val}}
\]

**Challenge:** different bias on different individuals 😞

**Idea:** interpolate train bias \(\rightarrow\) val bias 😊
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)

\[
\begin{align*}
\text{estimator} \ (\lambda) & \quad (\hat{x}_{\text{train}}, \hat{B}_\text{train}) \\
& \quad \text{ordering} \quad \text{error} \\
& \quad \|Y_{\text{val}} - \hat{x}_{\text{train}}{1^T} - \hat{B}_\text{val}\|^2_{\text{val}} \\
& \quad \text{Interpolate} \ \hat{B}_\text{val} \ \text{using} \ (\hat{B}_\text{train}, \text{ordering}) \\
\end{align*}
\]
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)

\[
\text{estimator } (\lambda) \rightarrow (\hat{x}_{\text{train}}, \hat{B}_{\text{train}}) \rightarrow \hat{B}_{\text{train}} \leftarrow \text{ordering} \rightarrow \hat{B}_{\text{val}} \rightarrow \text{error} \rightarrow ||Y_{\text{val}} - \hat{x}_{\text{train}}^T - \hat{B}_{\text{val}}||^2_{\text{val}}
\]

Interpolate \(\hat{B}_{\text{val}}\) using \((\hat{B}_{\text{train}}, \text{ordering})\)

\[
\hat{B}_{\text{train}} \quad -1 \quad 0 \quad 2 \quad 3 \quad \ldots \quad \text{increasing}
\]

\[
\hat{B}_{\text{val}} \quad -1 \quad ? \quad ? \quad \ldots
\]
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)

Interpolate \(\hat{B}_{\text{val}}\) using \((\hat{B}_{\text{train}}, \text{ordering})\)

\[ \|Y_{\text{val}} - \hat{x}_{\text{train}}^T - \hat{B}_{\text{val}}\|_{\text{val}}^2 \]

- \(\hat{B}_{\text{train}}\) -1 → 0 → 2 → 3 → … increasing
- \(\hat{B}_{\text{val}}\) -1 0 ? → …
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)

\[
\hat{B}_{\text{val}} - \hat{Y}_{\text{val}} = (Y_{\text{val}} - \hat{Y}_{\text{train}}) + (\hat{B}_{\text{val}} - \hat{B}_{\text{train}})
\]

Interpolate \(\hat{B}_{\text{val}}\) using \((\hat{B}_{\text{train}}, \text{ordering})\)

---

\[
\begin{align*}
\hat{B}_{\text{train}} &\quad -1 & 0 & 2 & 3 \\
\hat{B}_{\text{val}} &\quad -1 & 0 & \frac{0 + 2}{2} = 1
\end{align*}
\]
Algorithm (sketch)

1. **Split** data to \((Y_{\text{train}}, Y_{\text{val}})\) in a “balanced” way
2. **Compute** validation error for each \(\lambda\)
3. **Choose** \(\lambda\) that minimizes the validation error

---

Interpolate \(\hat{B}_{\text{val}}\) using \((\hat{B}_{\text{train}}, \text{ordering})\)
Theoretical guarantees

**Theorem 3 (informal).** In cases of common partial orderings,

- when there is no noise, we have
  \[ \hat{x}_{CV} \rightarrow \hat{x}^{(0)}; \]
- when there is no bias, we have
  \[ \hat{x}_{CV} \rightarrow \hat{x}^{(\infty)}. \]

Our cross-validation successfully recovers the two extremal cases.
Experiment

- Indiana University Bloomington
- 10 sessions of a course
- Simulate bias and noise using real grading statistics

![Graph showing MSE (Mean Squared Error) against bias/(bias + noise)]

- no bias: all estimators work well
- lots of bias: our estimator significantly better than \{mean, median\}
Experiment

- Indiana University Bloomington
- 10 sessions of a course
- Simulate bias and noise using real grading statistics
Take-aways

• Use an ordering constraint to model experience-induced bias, without making restrictive assumptions
• Design a novel CV algorithm to tease out bias vs noise

Future work

• Sharp statistical bounds on error rates / sample complexity + when there is both bias and noise
• Combining with a game-theoretic approach to design mechanisms

Thanks :)
jingyanw@cmu.edu