The Cram Method for Efficient Simultaneous Learning and Evaluation

Kosuke Imai

Harvard University

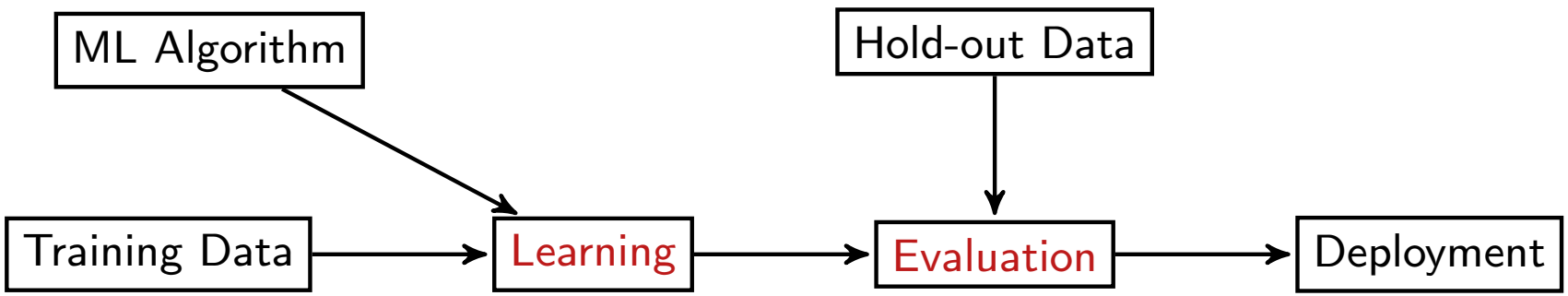
Joint Statistical Meetings
August 5, 2024

Joint work with Zeyang Jia and Michael Lingzhi Li (HBS)

Motivation

- Widespread use of data-driven algorithms for decisions and predictions
- In practice, we wish to use the same data to:
  1. learn a decision/prediction rule
  2. evaluate the learned rule

- Sample splitting achieves this goal but use the data inefficiently:



- Cross-validation is statistically efficient but
  - does not evaluate the learned rule
  - instead, it evaluates the average performance of ML algorithm
  - this leads to underestimation of uncertainty
  - in adition, it is computationally inefficient

The Cram Method

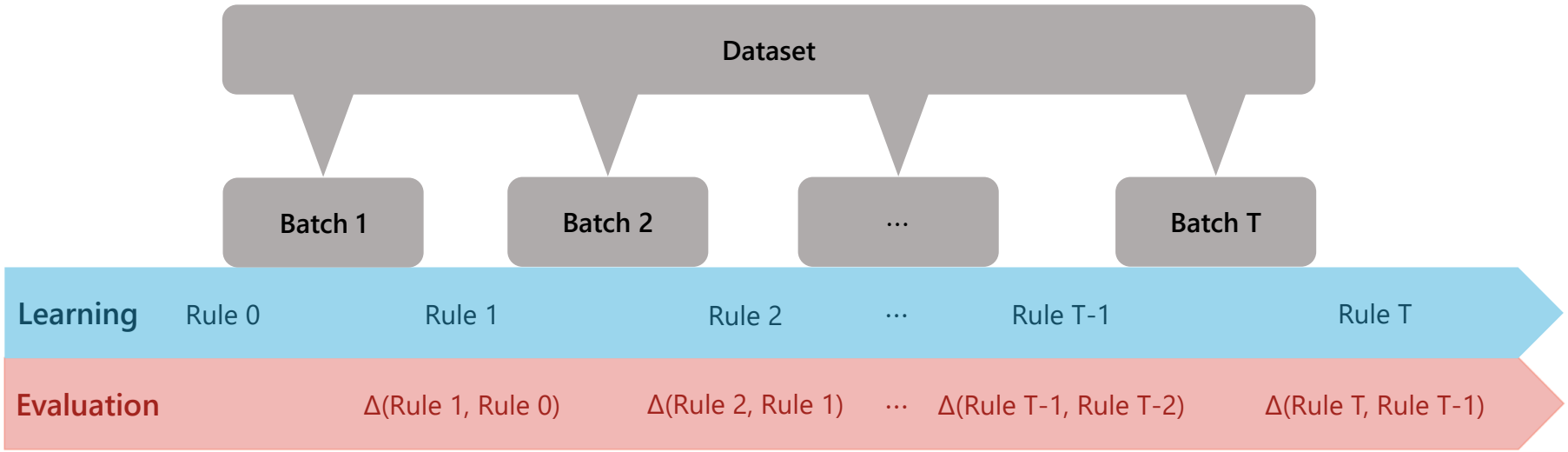
- General methodology for simultaneous learning and evaluation
  - process of repeated training and testing
  - yields a single learned rule and its statistical performance evaluation
  - incorporates both learning and evaluation uncertainties

- Cramming is statistically efficient:
  1. the entire sample is used to learn a decision/prediction rule
  2. the entire sample is used to evaluate the learned rule

- Cramming is computationally efficient:
  1. learning and evaluation occur through a single pass of the sample
  2. online fitting algorithms can be used

- Cram is a general methodology — various extensions are possible

Cramming at Grance



1. Divide the data into T batches
2. Start with Rule 0
3. Use Batch 1 to learn Rule 1
4. Use Batches 2–T to evaluate the performance difference between Rules 0 and 1, i.e., $\Delta(\text{Rule 1}, \text{Rule 0})$
5. Use Batches 1–2 to learn Rule 2
6. Use Batches 3–T to evaluate $\Delta(\text{Rule 2}, \text{Rule 1})$
7. Repeat

Cramming for Policy Learning and Evaluation

- Data (i.i.d.): $\mathcal{D}_n = \{\mathsf{X}_i, D_i, Y_i\}_{i=1}^n$
  - treatment: $D_i \in \{0, 1\}$
  - outcome: $Y_i = Y_i(D_i) \in \mathcal{Y} \subset \mathbb{R}$
  - pre-treatment covariates: $\mathsf{X}_i \in \mathcal{X} \subset \mathbb{R}^p$
- Assumption (Strong Ignorability):
  1. unconfoundedness: $\{Y(1), Y(0)\} \perp\!\!\!\perp D \mid \mathsf{X}$
  2. overlap: $c \leq e(\mathsf{x}) := \mathbb{P}(D = 1 \mid \mathsf{X} = \mathsf{x}) \leq 1 - c$ where $c > 0$
- Policy (either stochastic or deterministic):

$$\pi(\mathsf{x}) = \mathbb{P}(D = 1 \mid \mathsf{X} = \mathsf{x}) \in [0, 1]$$

- Value of policy π:

$$V(\pi) \;:=\; \mathbb{E}_{D\sim\pi}[Y(D)] \;=\; \mathbb{E}[Y(1)\pi(\mathsf{X}) + Y(0)(1 - \pi(\mathsf{X}))]$$

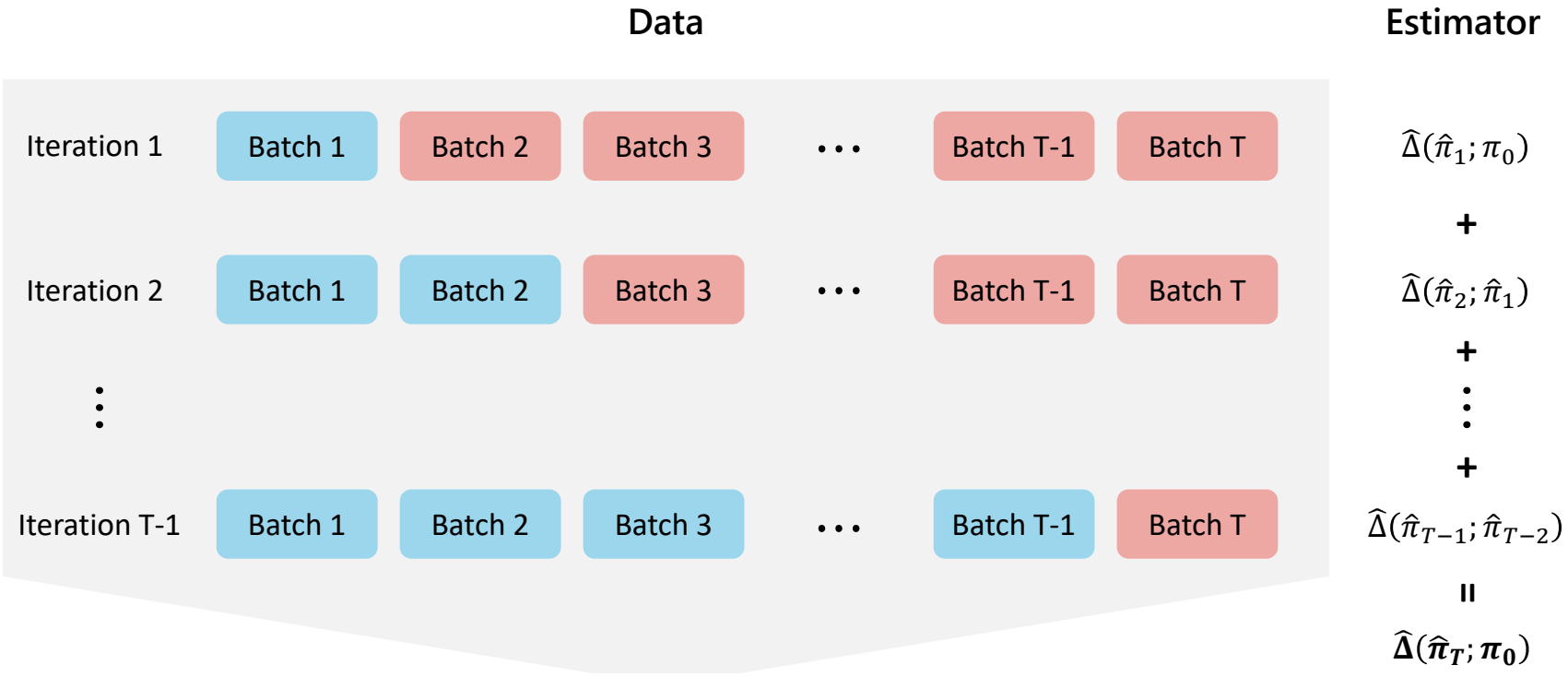
- Policy value difference:

$$\Delta(\pi; \pi') \;:=\; V(\pi) - V(\pi') = \mathbb{E}[(Y(1) - Y(0))(\pi(X) - \pi'(X))]$$

- Policy learning:

$$\hat{\pi} \;=\; \underset{\pi\in\Pi}{\operatorname{argmax}} \; V(\pi)$$

Cramming by Picture



- Use blue batches to learn and red batches to evaluate
- Key decomposition:

$$\Delta(\hat{\pi}_T; \pi_0) := V(\hat{\pi}_T) - V(\pi_0)$$
$$= \sum_{t=1}^{T} \Delta(\hat{\pi}_t; \hat{\pi}_{t-1}) \approx \sum_{t=1}^{T-1} \Delta(\hat{\pi}_t; \hat{\pi}_{t-1})$$

- Cram can also be used to evaluate $V(\hat{\pi}_T)$

The Cram Method for Policy Learning and Evaluation

Algorithm: Cramming

Data: $\mathcal{D}_n = \{X_i, D_i, Y_i\}_{i=1}^n$

Input: learning algorithm $\mathcal{A}$, baseline policy π_0, number of batches T

Output: estimated value difference between the learned and baseline policies $\widehat{\Delta}(\mathcal{A}(\mathcal{D}); \pi_0)$

1 Randomly partition the dataset $\mathcal{D}_n$ into T batches $\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_T$;

2 Set $\hat{\pi}_0 = \pi_0$;

3 **for** $t = 1$ **to** $T - 1$ **do**

   1. Learn a policy using the first t batches $\hat{\pi}_t := \mathcal{A}(\bigcup_{j=1}^t \mathcal{B}_j)$;
   2. Evaluate the policy value difference between $\hat{\pi}_t$ and $\hat{\pi}_{t-1}$ using the remaining batches $\bigcup_{j=t+1}^T \mathcal{B}_j$ and store the resulting estimate as $\widehat{\Delta}(\hat{\pi}_t; \hat{\pi}_{t-1})$;

4 Evaluate the value difference between the final learned policy $\hat{\pi}_T := \mathcal{A}(\mathcal{D}_n)$ and the baseline policy π_0 as:

$$\widehat{\Delta}(\hat{\pi}_T; \pi_0) := \sum_{t=1}^{T-1} \widehat{\Delta}(\hat{\pi}_t; \hat{\pi}_{t-1}).$$

Crammed Policy Evaluation Estimator

- Proposed estimator:

$$\widehat{\Delta}(\hat{\pi}_T; \pi_0) \;:=\; \sum_{t=1}^{T-1} \widehat{\Delta}(\hat{\pi}_t; \hat{\pi}_{t-1})$$

where

$$\widehat{\Delta}(\hat{\pi}_t; \hat{\pi}_{t-1}) \;:=\; \frac{1}{T-t} \sum_{j=t+1}^{T} \widehat{\Gamma}_{tj}, \text{ (avg. of } T-t \text{ unbiased estimates)}$$

$$\widehat{\Gamma}_{tj} \;:=\; \frac{1}{B} \sum_{i \in \mathcal{B}_j} \underbrace{\left\{ \frac{Y_i D_i}{e(X_i)} - \frac{Y_i(1-D_i)}{1-e(X_i)} \right\}}_{\text{inverse probability weighting}} \cdot \underbrace{\left(\hat{\pi}_t(X_i) - \hat{\pi}_{t-1}(X_i)\right)}_{\text{policy difference}}.$$

- Could use other unbiased estimator (e.g., doubly-robust estimator)
- Difficult to analyze because of complex correlations across $\widehat{\Delta}(\hat{\pi}_t; \hat{\pi}_{t-1})$

Exploiting the Sequential Structure of Cramming

- Alternaive expression of the same crammed estimator:

$$\widehat{\Delta}(\hat{\pi}_T; \pi_0) \;=\; \sum_{j=2}^{T} \widehat{\Gamma}_j(T) \quad \text{where} \quad \widehat{\Gamma}_j(T) \;:=\; \sum_{t=1}^{j-1} \frac{1}{T-t} \widehat{\Gamma}_{tj}.$$

- $\widehat{\Gamma}_{tj}$ is an unbiased estimator of $\Delta(\hat{\pi}_t; \hat{\pi}_{t-1})$ using batch j
- Using batch j, $\widehat{\Gamma}_j(T)$ estimates $\sum_{t=1}^{j-1} \Delta(\hat{\pi}_t; \hat{\pi}_{t-1})/(T-t)$, which depends only on prior batches

Policy difference	1	2	3	$\cdots$	$T-1$	T
$\Delta(\hat{\pi}_1; \pi_0)$	✓	$\frac{\widehat{\Gamma}_{1,2}}{T-1}$	$\frac{\widehat{\Gamma}_{1,3}}{T-1}$	$\cdots$	$\frac{\widehat{\Gamma}_{1,T-1}}{T-1}$	$\frac{\widehat{\Gamma}_{1,T}}{T-1}$
$\Delta(\hat{\pi}_2; \hat{\pi}_1)$	✓	✓	$\frac{\widehat{\Gamma}_{2,3}}{T-2}$	$\cdots$	$\frac{\widehat{\Gamma}_{2,T-1}}{T-2}$	$\frac{\widehat{\Gamma}_{2,T}}{T-2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\Delta(\hat{\pi}_{T-2}; \hat{\pi}_{T-3})$	✓	✓	✓	$\cdots$	$\frac{\widehat{\Gamma}_{T-2,T-1}}{2}$	$\frac{\widehat{\Gamma}_{T-2,T}}{2}$
$\Delta(\hat{\pi}_{T-1}; \hat{\pi}_{T-2})$	✓	✓	✓	$\cdots$	✓	$\frac{\widehat{\Gamma}_{T-1,T}}{1}$
$\Delta(\hat{\pi}_T; \hat{\pi}_{T-1})$	✓	✓	✓	$\cdots$	✓	✓

Stability Condition

- The amount of evaluation data decreases at the rate of $1/T$
- The policy value difference must stabilize at least at the same rate

Assumption 2 (Stability Condition)

The learning algorithm satisfies the following stabilization rate condition; $\exists \delta > 0, R_1 > 0, K_0 > 0$, *such that for all* $t \geq R_1$,

$$t^{1+\delta} Q_t \leq K_0 \text{ holds almost surely}$$

where $Q_t := \mathbb{E}_X [|\hat{\pi}_t(X) - \hat{\pi}_{t-1}(X)|] = \int_{x \in \mathcal{X}} |\hat{\pi}_t(x) - \hat{\pi}_{t-1}(x)| dF_X(x)$

- Q_t is a random variable that depends on the data $\mathcal{D}_n$
- We can relax this uniform condition to

$$\limsup_{t \to \infty} t^{1+\delta} Q_t \leq K_0 \text{ almost surely}$$

- Currently, exploring additional relaxation

A Simple Algorithm to Stabilize any Learning Algorithm

Algorithm: Stabilizer

Data: a sequence of batches from cramming, $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_t, \ldots$

Input: a policy learning algorithm $\mathcal{A}$, a baseline policy π_0, constants $\delta > 0$ and $C > 0$

Result: a sequence of learned policies $\hat{\pi}_1, \hat{\pi}_2, \ldots$ satisfying the stability condition of Assumption 2

Set $\hat{\pi}_0 = \pi_0$;

for $t \geq 1$ **do**

  Obtain a candidate policy $\tilde{\pi}_t := \mathcal{A}(\bigcup_{j=1}^{t} \mathcal{B}_j)$ by applying the algorithm $\mathcal{A}$ to the first t batches

  Compute the acceptance probability $p_t := \min\{Ct^{-1-\delta}, 1\}$

  Generate a learned policy as $\hat{\pi}_t(x) := p_t \tilde{\pi}_t(x) + (1 - p_t)\hat{\pi}_{t-1}(x)$

- Choose sufficiently large C and sufficiently small δ
- In practice, choose these values such that the algorithm can learn without modification for at least 80% of the data

Other Conditions on Learning Algorithms

Proposition 1 (Limit policy)

Under Assumption 2, for any learned policy sequence $\{\hat{\pi}_t\}_{t=1}^{\infty}$*, there exists a unique limit policy* $\hat{\pi}_\infty : \mathcal{X} \to [0,1]$ *such that with probability 1 the following equality holds,*

$$\lim_{t\to\infty} \mathbb{E}_X[|\hat{\pi}_\infty(X) - \hat{\pi}_t(X)|] \;=\; 0.$$

Assumption 3 (Limit policy differs from the baseline policy)

The limit policy $\hat{\pi}_\infty$ *of a learned policy sequence* $\{\hat{\pi}_t\}_{t=1}^{\infty}$ *differs from the baseline policy* π_0 *in the* L_1 *distance almost surely. That is, there exists* $M_1 > 0$ *such that,*

$$\mathbb{E}_X\left[|\pi_0(X) - \hat{\pi}_\infty(X)|\right] > M_1 \quad \textit{almost surely.}$$

Regularity Conditions

Assumption 4 (Bounded conditional moments)

Both the conditional expectation and conditional variance of the potential outcome, i.e., $\mu_d(\mathrm{x}) := \mathbb{E}[Y(d) \mid \mathrm{X} = \mathrm{x}]$ *and* $\sigma_d^2(\mathrm{x}) := \mathbb{V}(Y(d) \mid \mathrm{X} = \mathrm{x})$ *for* $d = 0, 1$*, respectively, are uniformly bounded on the covariate space* $\mathcal{X}$*:*

$$\sup_{\mathrm{x} \in \mathcal{X}} \mu_d(\mathrm{x}) < \infty, \quad 0 < \inf_{\mathrm{x} \in \mathcal{X}} \sigma_d^2(\mathrm{x}) \leq \sup_{\mathrm{x} \in \mathcal{X}} \sigma_d^2(\mathrm{x}) < \infty, \quad \text{for } d = 0, 1.$$

Assumption 5 (Moment condition)

The potential outcomes have finite fourth moments:

$$\exists K_4 > 0, s.t. \;\; \mathbb{E}[Y(d)^4] \leq K_4, \quad \text{for } d = 0, 1.$$

Consistency

Theorem 1 (L_1 consistency)

Suppose that a sequence of learned policies $\{\hat{\pi}_t\}_{t=1}^T$ satisfies Assumption 2. Then, under Assumptions 1, 4, and 5, we have,

$$\mathbb{E}\left[\left|\widehat{\Delta}(\hat{\pi}_T; \pi_0) - \Delta(\hat{\pi}_T; \pi_0)\right|\right] \to 0 \quad \textit{as} \quad T \to \infty.$$

Asymptotic Normality

Theorem 2 (Asymptotic normality)

Suppose that a sequence of learned policies $\{\hat{\pi}_t\}_{t=1}^T$ *satisfies Assumptions 2 and 3. Then, under Assumptions 1, 4, and 5, we have,*

$$\sqrt{T} \cdot \frac{\widehat{\Delta}(\hat{\pi}_T; \pi_0) - \Delta(\hat{\pi}_T; \pi_0)}{v_T} \xrightarrow{d} N(0, 1).$$

The asymptotic variance is given by,

$$v_T^2 := T \sum_{j=2}^{T} \mathbb{V}(\widehat{\Gamma}_j(T) \mid \mathcal{H}_{j-1}).$$

- Unlike the standard CLT, $\Delta(\hat{\pi}_T; \pi_0)$ depends on the data $\mathcal{D}_n$
- Leverage the fact that $\widehat{\Gamma}_j(T)$ is i.i.d. conditional on $\mathcal{H}_{j-1}$

Discussion

- Why does cramming work?
  - evaluation starts early if policy stabilizes fast
  - more samples used for evaluation early when policy is changing
  - plot estimated policy value differences to check stabilization

- Choice of batch size
  - smaller batch size leads to more efficient use of data
  - larger batch size leads to more stable policy learning
  - noisy data $\longrightarrow$ smaller batch size
  - we do not yet know optimal batch size / batch size can vary too
  - in practice, we recommend a batch size of about 5%

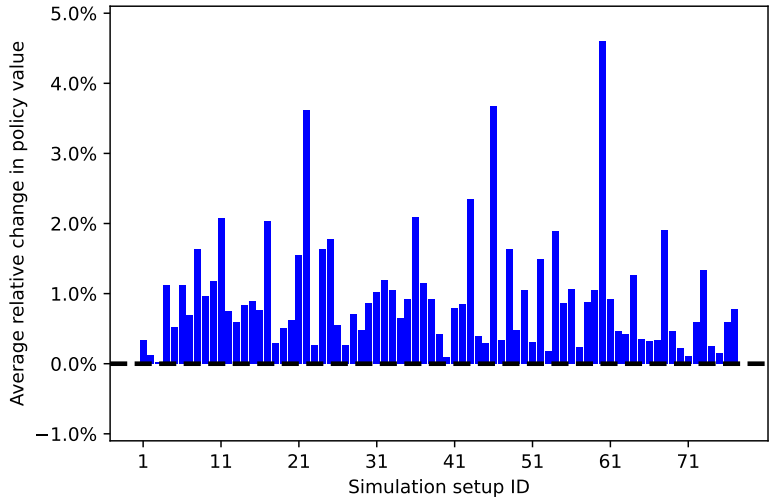
Simulation Studies

- ACIC 2016 Data set (Dorie et al. 2019)
  - 77 different DGPs
  - conditional average treatment effect (CATE) estimation
  - various nonlinearities and signal-noise ratios

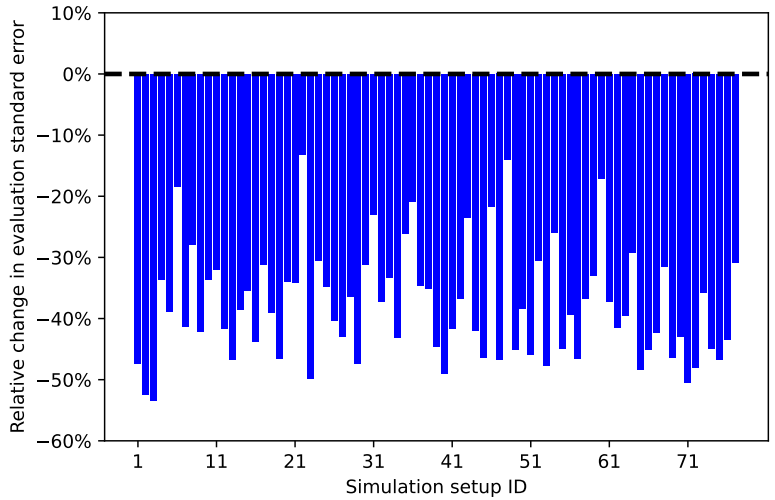
- Learning algorithms:
  - S-learner: outcome models with treatment / covariate interactions
  - M-learner: modified outcome model $YD/e(X) - Y(1-D)/(1-e(X))$
  - Causal Forest (Wager and Athey 2018)
  - For S and M-learners, we use ridge regression and neural networks

- Sample splitting: 80-20% (main text), 60-40% splits (appendix)
- Cramming: 5% batch size (results not sensitive to the batch size)

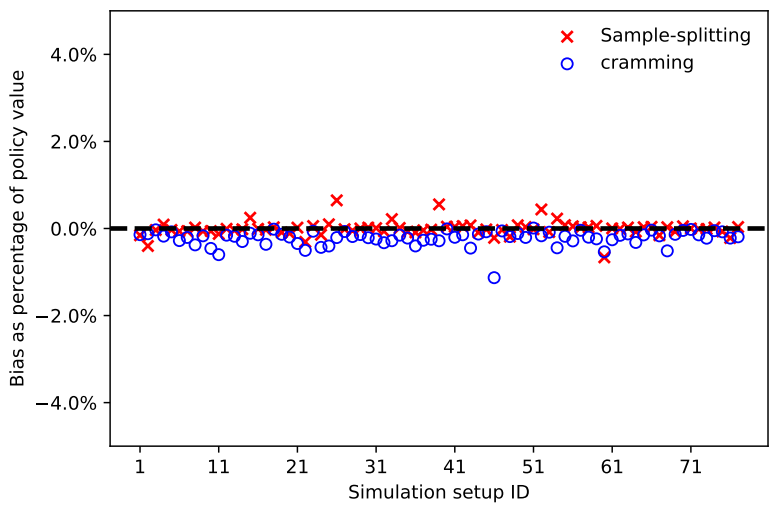
Cramming vs. Sample Splitting



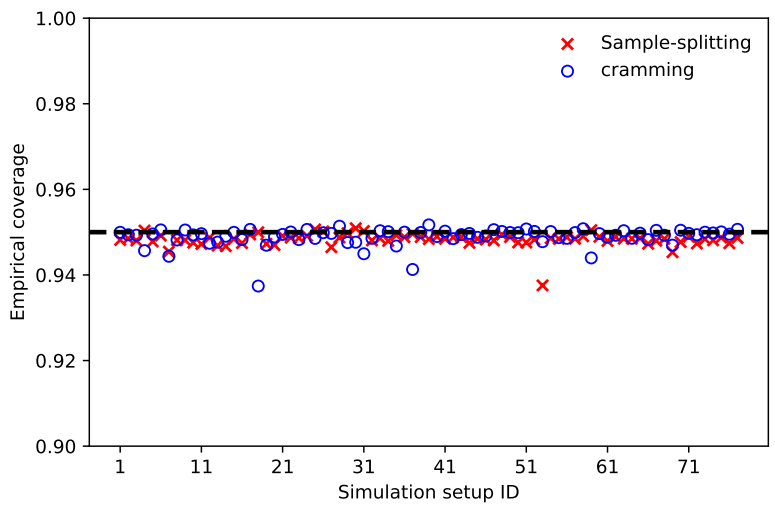
(a) Improvement in the policy value



(b) Improvement in standard error



(c) Standardized bias



(d) Coverage of 95% confidence intervals

Empirical Application

- Clinical trial: synthetic estrogen for late-stage prostate cancer
- Binary Treatment: 5.0mg/2.0mg/1.0mg estrogen vs control
- No statistically significant average treatment effect on total survival
- But, subsequent analyses found heterogeneous effects

- Trial data:
  - Covariates X: Patient characteristics at initial visit
  - Treatment D: Control ($D = 0$) and Estrogen ($D = 1$)
  - Outcome Y: Length of total survival
- Baseline: No treatment
- Cramming: 5% batch size
- Sample-splitting: 80/20 split

Cramming is More Effective than Sample Splitting

	cramming	**sample-splitting**
Estimated proportion treated	57.76%	56.94%
Estimated value	7.77	3.90
Estimated standard error	4.42	6.65
90% confidence interval	[0.50, 15.04]	[−7.03, 14.84]

- 99% increase in estimated policy value: $3.90 \longrightarrow 7.77$
- 33% reduction of standard error: $6.65 \longrightarrow 4.42$

Concluding Remarks

- The cram method for simultaneous policy learning and evaluation
  - statistically efficient
  - computationally efficient
  - more efficient alternative to sample splitting
  - application to policy learning and evaluation
  - evaluates a learned policy rather than an algorithm using the same data

- Future extensions:
  - onpolicy evaluation of bandit (coming soon!)
  - machine learning prediction and classification (work in progress)
  - cramming cross-validation
  - active learning

- Paper available at https://arxiv.org/pdf/2403.07031.pdf