Batch Mode Active Learning for Individual Treatment Effect Estimation

Zoltan Puha, Maurits Kaptein, Aurelie Lemmens

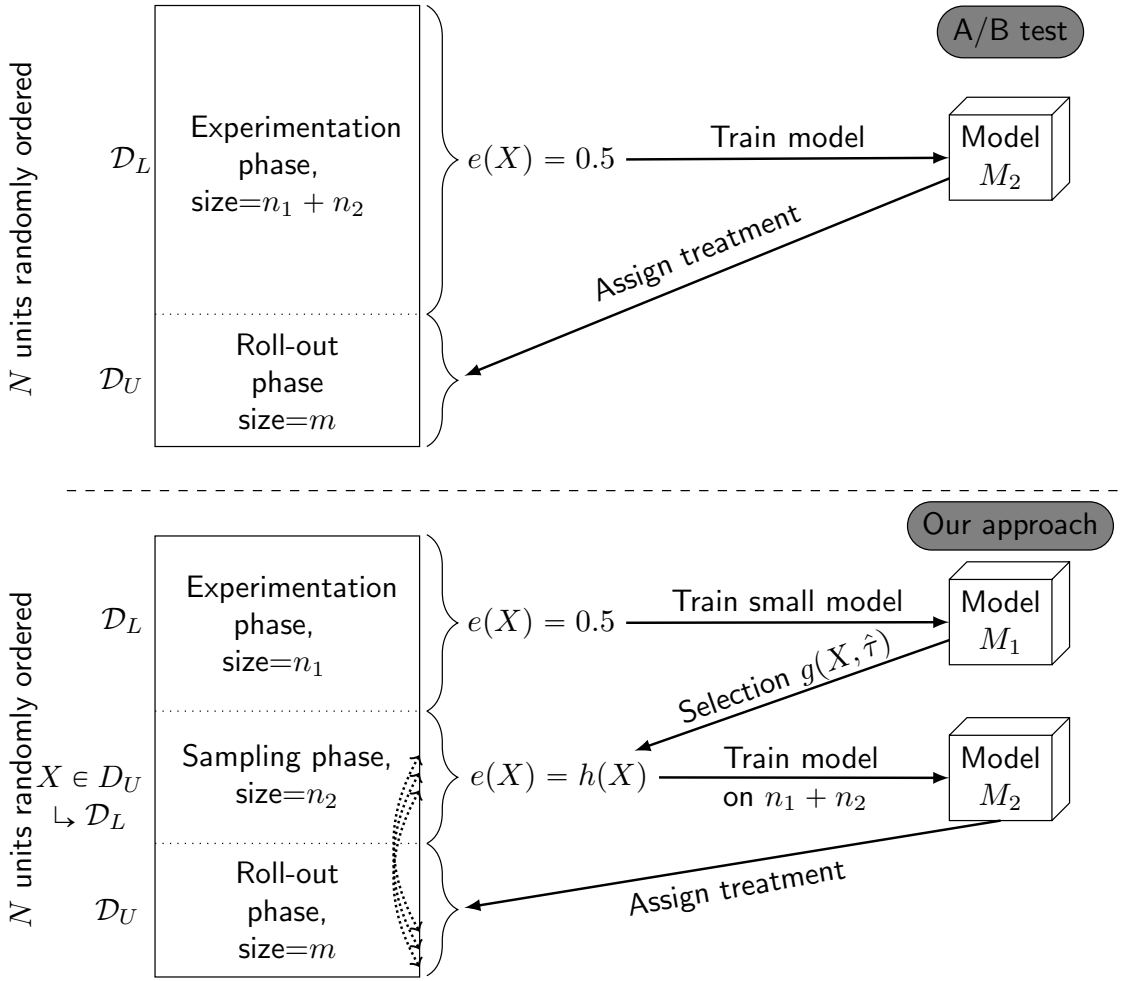
Jheronimus Academy of Data Science, Tilburg University & Erasmus University

IncrLearn @ ICDM

Combining Active Learning with Treatment Effect Estimation

- Field experimentation and uplift modeling is widespread in industry
- Several new methods have been proposed to tackle the uplift modeling and estimate treatment effects
- Surprisingly small focus on data collection
  - Current focus is on sequential data collection
  - We are motivated to develop method for estimating uplift accurately based on a small number of observations that arrive in a batch

Proposed framework



Uplift modeling

- Precise estimation of a treatment
  - In medicine: who should get the *drug*?
  - In marketing: who should get the *e-mail*?
  - In online settings: who should see the *ad*?
- We have data about the observed outcome:

$$Y = TY(1) + (1 - T)Y(0)$$

  - X are the covariates
  - T is the binary treatment
  - $Y(T) = \mathbb{E}[Y|X = x_i, T]$ is the potential outcome for treatment and control

Uplift modeling

- Uplift modeling aimes to estimate the difference between the potential outcomes

$$\tau_i = Y(1) - Y(0) \tag{1}$$

- Estimating τ_i is equivalent to finding the correct ordering of treatment effects
- We use **Bayesian Additive Regression Trees** (BART) as this model provides uncertainty around τ_i (needed for AL)

Active Learning (AL)

- AL lowers the data cost for supervised learning by finding *informative* units
- Research is about the definition of *informative*
  - Uncertainty based

Active Learning (AL)

- AL lowers the data cost for supervised learning by finding *informative* units
- Research is about the definition of *informative*
  - Uncertainty based
  - Query-by-comittee

Active Learning (AL)

- AL lowers the data cost for supervised learning by finding *informative* units
- Research is about the definition of *informative*
  - Uncertainty based
  - Query-by-comittee
  - Expected Error reduction

Active Learning (AL)

- AL lowers the data cost for supervised learning by finding *informative* units
- Research is about the definition of *informative*
  - Uncertainty based
  - Query-by-comittee
  - Expected Error reduction
  - Expected Model Change Maximization

Active Learning (AL)

- AL lowers the data cost for supervised learning by finding *informative* units
- Research is about the definition of *informative*
  - Uncertainty based
  - Query-by-comittee
  - Expected Error reduction
  - Expected Model Change Maximization
  - Many more
- Currently, limited usage for uplift modeling

Expected Model Change Maximization

- Measures the change of parameters at every potential new unit
- Consider squared loss:

$$L = \sum_{i=1}^{n_1} (y_i - f_R(\boldsymbol{x_i}, \theta))^2. \tag{2}$$

- Then, the optimal next x is:

$$\boldsymbol{x_i}^* = \underset{\boldsymbol{x_i}' \in \mathcal{D}_U}{\operatorname{argmax}} ||\Delta\theta||, \tag{3}$$

- where $\Delta\theta$ is the size of the coefficients after adding x_i'
- $\theta \not\perp y_i'$ and y_i' is not known $\rightarrow$ estimated outcomes are used

Active Learning and uplift modeling

Research in this direction picking up in recent years

- Healthcare and medical applications
- Type-S sampling (Sundin et. al, ICML'19)
- bounds on AL on observational data (Yan et. al, NeurIPS '19)

All of the above are for sequential designs $\rightarrow$

Why is batch-mode sampling important?

Sequential designs are not always fitting

- Delayed feedback
- Cost of setting up and maintaining online experimentation platform

Our algorithm considers experiment planning, where selection is in waves (in simulations we use one wave, one-shot AL)

Compared to A/B testing

We work with two functions, one for selecting people (*acquisition function*, $g(\cdot)$) and an other for allocating treatment (*assignment function*, $h(\cdot)$)

- In A/B testing the acquisition is random ($g(\cdot) = 1/n_2$)
- And the assignment is also random ($h(\cdot) = 1/2$)
- We change the functions for better data collection

Acquisition Function $g(X, \hat{\tau})$

- We extend EMCM to uplift modeling by using the treatment effects
- τ is never observed - we use the predictive posterior of BART to sample $\hat{\tau}_i$

$$L = \sum_{i=1}^{n_1} (1 + \zeta\gamma) \left(\hat{\tau}_i - f_R(\boldsymbol{x}_i, \theta)\right)^2 \tag{4}$$

- γ predicted Type-S error ($[0, 0.5]$)
- ζ weight given to Type-S error

Select x that maximizes $\frac{\partial L_{\boldsymbol{x}'}}{\partial \theta} \rightarrow$ do this iteratively until n_2

Assignment function - $h(X)$

- Working with treatment effects means treatment or control needs to be assigned
- One option is to do it randomly - $e(X) = 0.5$
- We use an *assignment function*:

$$e(\boldsymbol{x}_i) = \frac{V(\widehat{y}_i(1))}{V(\widehat{y}_i(0)) + V(\widehat{y}_i(1))} \tag{5}$$

- by setting the propensity score proportional to the variance of the potential outcomes
- We can then expect the selected potential outcome to decrease variance faster and lead to better estimate

Putting it together - algorithm

In short (full algorithm in paper):

- Train 1st model and predict τ on unlabeled
- While $i < n_2$:
  - calculate model change for D_U
  - add to training set highest gradient and remove from pool
  - $i+ = 1$
- Train 2nd model and predict on test set

Algorithm in details - speed-up, better performance

1. Speed: Checking EMCM only on a subsample of units

Algorithm in details - speed-up, better performance

1. Speed: Checking EMCM only on a subsample of units
2. Speed: Number of draws from the posterior predictive

Algorithm in details - speed-up, better performance

1. Speed: Checking EMCM only on a subsample of units
2. Speed: Number of draws from the posterior predictive
3. Performance: Subsampling in proportional to Type-S error

Algorithm in details - speed-up, better performance

1. Speed: Checking EMCM only on a subsample of units
2. Speed: Number of draws from the posterior predictive
3. Performance: Subsampling in proportional to Type-S error

Competing AL methods

Simulations:

We checked our acquisition function against 3 others

1. Random sampling (A/B tests)
2. Uncertainty based sampling (top-n_2 ordered by $Var(\hat{\tau})$)
3. Type-S based sampling (top-n_2 ordered by predicted Type-S error)

Evaluation - metrics and simulation

- We evaluated on different simulated datasets (see github) +
- IHDP data

Metrics used:

- normalized PEHE (squared error) on roll-out sample (people not selected)
- Effective Sample Size (approximated number of people to achieve same level of accuracy as with random sampling)

Simulation

- varying n_1, n_2 and hyperparameters

Results

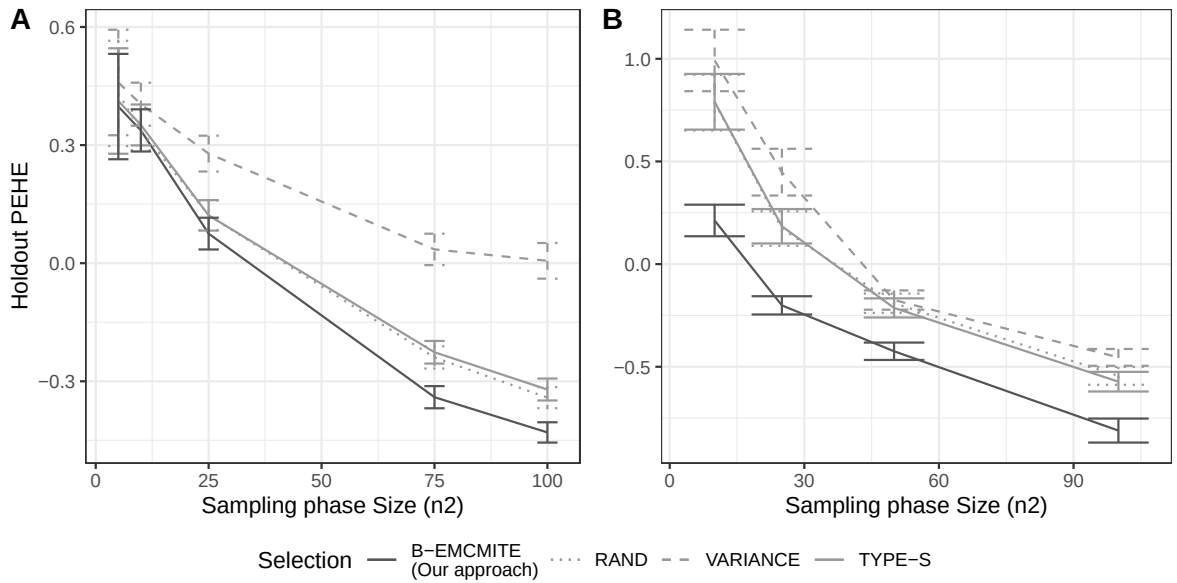


Figure: Performance - standardized PEHE scores

A are simulated data, **B** are IHDP datasets

Results II.

Simulated Data		*ESS* vs. **RAND** **B-EMCMITE** vs. **RAND** (in %)
Simulated Data		
DGP for Y(0):	*DGP for ITE:*	
Linear	Linear	**69.8**
Linear	Square	72.8
Sundin	Linear	74.5
Sundin	Square	74.7
Linear	Square,p=10	83.9
Lu	Lu	90.8
Zaidi	Athey	94.6
Linear	Sin	105.8
Zaidi Lower	Athey	109.5
B. Semi-Synthetic Data		
Response Surface 3		54.8
Response Surface 2		58.0
Response Surface 4		61.6
Response Surface 7		63.7
Response Surface 1		63.9
Response Surface 8		64.4
Response Surface 6		67.5
Response Surface 5		72.8
Response Surface 10		74.4
Response Surface 9		**77.2**

Table: Effective sample size (in %) for the simulated data, and IHDP semi-synthetic data (Panel B), ordered from smallest to largest.

Open questions

- Speed of the algorithm
- Reliance on BART - slow to perform consecutive retraining
- Approximation of τ with polynomial regression - kernel methods could be tested
- Use of real-world data - ongoing research

Summary

- We introduced a novel framework for batch-mode AL for uplift modeling
- We showed through extensive simulations its performance
- Our results indicate its potential benefits for companies in real-life settings