Bridging the Gap Between Theory and Practice in Off-Policy Evaluation

Open-Source Dataset/Software and Application in Fashion E-Commerce

Yuta Saito[1] , Shunsuke Aihara[2],

Megumi Matsutani[2], and Yusuke Narita[3]

[1]Tokyo Institute of Technology [2]ZOZO Technologies, Inc. [3]Yale University.

Outline

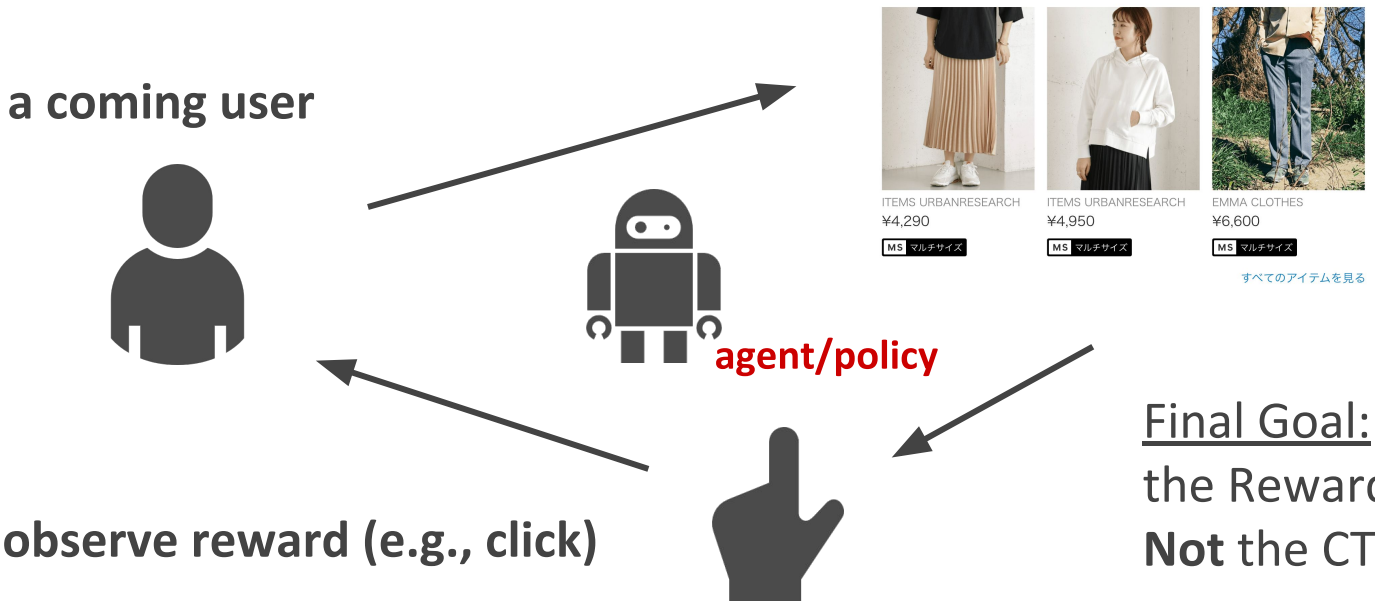
- Off-Policy Evaluation (OPE)
  - Basics and Current Issues
- Open Bandit Project
  - Open Source Dataset and Software for OPE
  - A Live Demonstration about OPE with the Open-Source
- Applications on the Large Fashion E-Commerce Platform
  - Off-Policy Estimator Selection / Counterfactual Policy Search

Off-Policy Evaluation:
Basics and Issues

Machine Learning for Decision Making (Bandit / RL)

We often use machine learning to make decisions, not predictions

Decision Making: item recommendation



Final Goal:
the Reward maximization
Not the CTR predction

Many Applications of "Machine Decision Making"

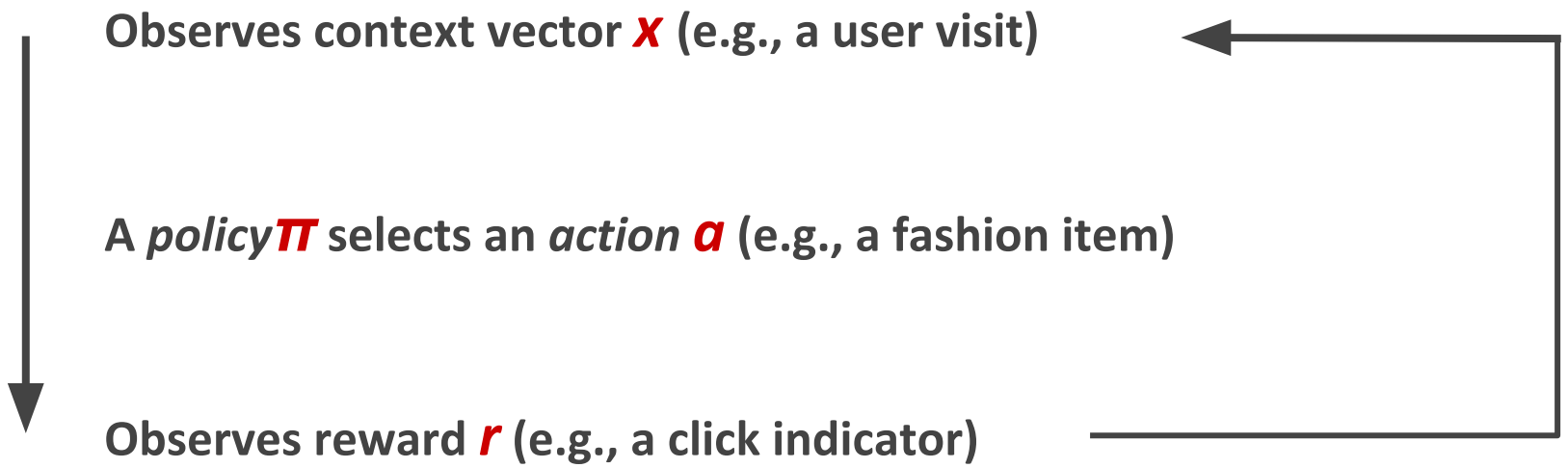
- news recommendation (by Yahoo)
- music/playlist/podcast recommendation (by Spotify)
- artwork personalization (by Netflix)
- ad allocation optimization (by Criteo)
- medicine
- education

etc...

Motivation of OPE

We want to evaluate the performance of a *new decision making policy* using data generated by a *behavior, past policy*

Data Generating Process (contextual bandit setting)



a *policy* interacts with the environment

and produces the *log data* valuable for *redesigning the system*

Logged Bandit Feedback

a behavior (or past) policy gives us *logged bandit feedback*

$$\mathcal{D} := \{(x_i, a_i, r_i)\}_{i=1}^{n}$$

$$a_i \sim \pi_b(a \mid x_i)$$

action choice by behavior policy

$$r_i \sim p(r \mid x_i, a_i)$$

observed reward

Estimation Target in Off-Policy Evaluation (OPE)

In OPE, we want to estimate the *policy value (policy performance)* of an *evaluation (or new)* policy

$$V(\pi_e) := \mathbb{E}_{p(x)\pi_e(a|x)p(r|x,a)}[r]$$

policy value

evaluation policy

expected reward obtained by running π_e on a real system

e.g., expected revenue by (hypothetically) deploying a new policy

Benefits of Off-Policy Evaluation

Goal: Accurate OPE of the policy value of an evaluation policy

$$V(\pi_e) \approx \hat{V}(\pi_e; \mathcal{D})$$

$$\mathcal{D} \sim p(x)\pi_b(a|x)p(r|x,a)$$

- avoid deploying poor performing policies without A/B tests
- identify promising new policies among many candidates

etc...

Growing interest in OPE!

Direct Method (DM)

DM first estimates the expected reward and uses it to estimate the policy value

$$\hat{V}_{\mathrm{DM}}\left(\pi_e; \mathcal{D}, \hat{q}\right) := \mathbb{E}_n\left[\sum_{a \in \mathcal{A}} \pi_e\left(a \mid x_i\right) \hat{q}\left(x_i, a\right)\right]$$

reward estimator

- **Large bias** when the model is mis-specified
- **Small variance**

$$\mathbb{E}[r \mid x, a] \approx \hat{q}(x, a)$$

$\mathbb{E}_n[\cdot]$: empirical expectation over D

Inverse Probability Weighting (IPW) Estimator

IPW re-weighs observed rewards by importance weights

$$\hat{V}_{\mathrm{IPW}}\left(\pi_e; \mathcal{D}\right) := \mathbb{E}_n\left[\underbrace{w\left(x_i, a_i\right)}_{\text{importance weight}} r_i\right]$$

$$w\left(x, a\right) := \pi_e(a \mid x) / \pi_b(a \mid x)$$

- **Consistent**
- **Large variance** when old and new policies are largely different

Doubly Robust (DR) Estimator

DR uses DM as a baseline and applies IPW to shifted rewards

$$\hat{V}_{\mathrm{DR}}\left(\pi_e; \mathcal{D}, \hat{q}\right)$$
$$:= \underbrace{\hat{V}_{\mathrm{DM}}\left(\pi_e; \mathcal{D}, \hat{q}\right)}_{\text{baseline}} + \underbrace{\mathbb{E}_n\left[w\left(x_i, a_i\right)\left(r_i - \hat{q}\left(x_i, a_i\right)\right)\right]}_{\text{weighted shifted reward}}$$

$$\mathbb{E}[r \mid x, a] \approx \hat{q}(x, a)$$

- **Consistent**
- **Locally Efficient**
  (desirable variance)

Theoretical/Methodological Advances in OPE

- Self-Normalized IPW [Swaminathan and Joachims 2015]
- Switch Doubly Robust Estimator [Wang+ 2017]
- More Robust Doubly Robust Estimator [Farajtabar+ 2018]
- Hirano-Imbence-Ridder Estimator [Narita+ 2019]
- Continuous Adaptive Blending [Su+ 2019]
- REG and EMP [Kallus & Uehara 2019]
- Doubly Robust with Shrinkage [Su+ 2020]

It seems the OPE community have made great progress over the years!

There are many other estimators in the reinforcement learning setting

Theoretical/Methodological Advances in OPE

- Self-Normalized IPW [Swaminathan and Joachims 2015]
- Switch Doubly Robust Estimator [Wang+ 2017]
- More Robust Doubly Robust Estimator [Farajtabar+ 2018]
- Hirano-Imbence-Ridder Estimator [Narita+ 2019]
- Continuous Adaptive Blending [Su+ 2019]
- REG and EMP [Kallus & Uehara 2019]
- Doubly Robust with Shrinkage [Su+ 2020]

Does this trend really expect the “era of OPE”?

There are many other estimators in the reinforcement learning setting

Issues with the current experimental procedures...

Experiments in ***every*** OPE paper rely on either

1. Synthetic or classification data (unrealistic)

or

2. (Real, but) Unpublished data (irreproduceble)

there is the ***critical gap between theory and practice in OPE***

Open Bandit Project:
Public Dataset and Software

The Goal of the “Open Bandit Project”

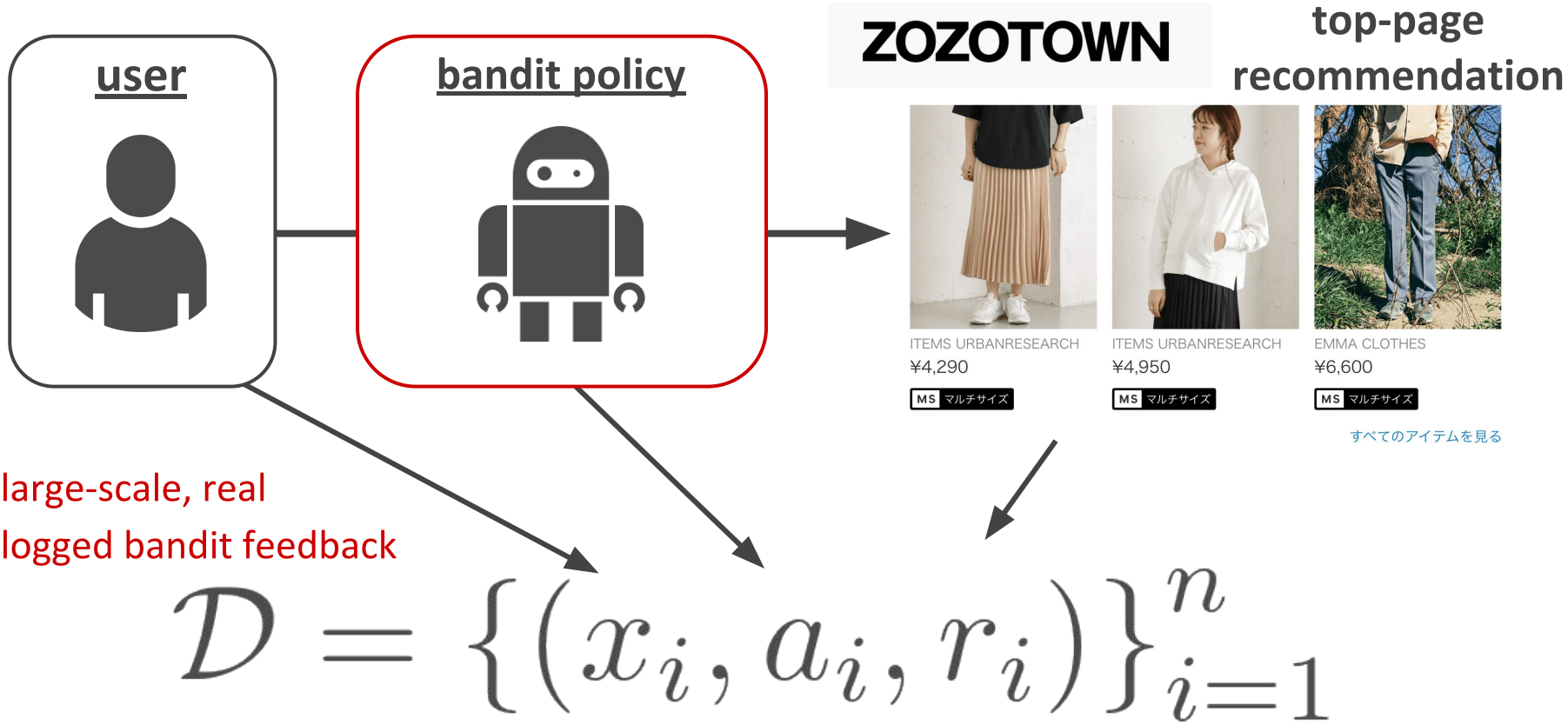
We enable *realistic* and *reproducible* experiments on

- **Off-Policy Evaluation (OPE)**
- (Bandit Algorithms)

“Open Bandit Dataset”

and ***“Open Bandit Pipeline”***

Open Bandit Dataset: Data Collection on ZOZOTOWN



Open Bandit Dataset: Release Page

https://research.zozo.com/data.html

Open Bandit Dataset

Open Bandit Dataset is a public real-world logged bandit feedback data. The dataset is provided by ZOZO, Inc., the largest Japanese fashion e-commerce company with over 5 billion USD market capitalization (as of May 2020). The company uses multi-armed bandit algorithms to recommend fashion items to users in a large-scale fashion e-commerce platform called ZOZOTOWN.

This dataset is released along with the paper:

Yuta Saito, Shunsuke Aihara, Megumi Matsutani, Yusuke Narita.
Large-scale Open Dataset, Pipeline, and Benchmark for Bandit Algorithms https://arxiv.org/abs/2008.07146

When using this dataset, please cite the paper with following bibtex:

```
@article{saito2020large,
  title={Large-scale Open Dataset, Pipeline, and Benchmark for Bandit Algorithms},
  author={Saito, Yuta, Shunsuke Aihara, Megumi Matsutani, Yusuke Narita},
  journal={arXiv preprint arXiv:2008.07146},
  year={2020}
}
```

Data description

Open Bandit Dataset is constructed in an A/B test of two multi-armed bandit policies in a large-scale fashion e-commerce platform, ZOZOTOWN. It currently consists of a total of 26M rows, each one representing a user impression with some feature values, selected items as actions, true propensity scores, and click indicators as an outcome. This is especially suitable for evaluating *off-policy evaluation* (OPE), which attempts to estimate the counterfactual performance of hypothetical algorithms using data generated by a different algorithm in use.

Fields

Here is a detailed description of the fields (they are comma-separated in the CSV files):
{behavior_policy}/{campaign}.csv (behavior_policy in (bts, random), campaign in (all, men, women))

- timestamp: timestamps of impressions.
- item_id: index of items as arms (index ranges from 0-80 in "All" campaign, 0-33 for "Men" campaign, and 0-46 "Women" campaign).
- position: the position of an item being recommended (1, 2, or 3 correspond to left, center, and right position of the ZOZOTOWN recommendation interface, respectively).
- click: target variable that indicates if an item was clicked (1) or not (0).
- propensity_score: the probability of an item being recommended at each position.
- user feature 0-4: user-related feature values.
- user-item affinity 0-: user-item affinity scores induced by the number of past clicks observed between each user-item pair.

item_context.csv

- item_id: index of items as arms (index ranges from 0-80 in "All" campaign, 0-33 for "Men" campaign, and 0-46 "Women" campaign).
- item feature 0-3: item related feature values

Please visit the examples to learn how to use the data.

Google Group

Open Bandit Dataset: Schema Image

	a		π_b	r	x	
timestamp	**item_id**	**position**	**action prob**	**click indicator**	**features**	**...**
2019-11-xx	25	1	0.0002	0	e2500f3f	...
2019-11-xx	32	2	0.043	1	7c414ef7	...
2019-11-xx	11	3	0.167	0	60bd4df9	...
2019-11-xx	40	1	0.0011	0	7c20d9b5	...
...	...	...	...	...	...	...

Open Bandit Dataset: Essential Features

- over 25M records collected by online experiments of bandit algorithms on a large-scale fashion e-commerce (*ZOZOTOWN*)

- *two* sets of logged bandit feedback collected by running *multiple* different bandit policies **most important**
  - *Uniform Random/Thompson Sampling* (fixed)

realistic and reproducible **"evaluation of OPE"** for the first time

Open Bandit Dataset: Protocol for the "Evaluation of OPE"

We can use our Open Bandit Dataset for the **"evaluation of OPE"**

evaluate the estimation accuracy of
an OPE estimator in a data-driven manner

1. Prepare *two* sets of logged bandit feedback
2. Evaluate the performance of an eval policy by an OPE estimator
3. Calculate the ground-truth performance of the eval policy
   by the *on-policy estimation*
4. Compare the ground-truth with the value estimated by OPE
   to evaluate the estimation accuracy of the estimator

Protocol for the Evaluation of OPE with Open Bandit Dataset

1. **Our data have logged bandit feedback collected by *two different policies***

$$\mathcal{D}_e \sim p(x)\pi_e(a \mid x)p(r \mid x, a)$$

e.g.) thompson sampling

$$\mathcal{D}_b \sim p(x)\pi_b(a \mid x)p(r \mid x, a)$$

e.g.) uniform random

Protocol for the Evaluation of OPE with Open Bandit Dataset

2. Regard one policy as an ***evaluation policy*** and the other as a ***behavior policy***. Then, estimate the performance of the evaluation policy by OPE

$$V(\pi_e) \approx \hat{V}(\pi_e; \mathcal{D}_b)$$

e.g.) DM/IPW/DR

π_e : **evaluation policy** π_b : **behavior policy**

- The task here is to evaluate the estimation accuracy of $\hat{V}$

Protocol for the Evaluation of OPE with Open Bandit Dataset

3. Regard the ***on-policy estimation*** of the policy value of the evaluation policy as the ground-truth policy value

$$V_{\mathrm{on}}\left(\pi_e; \mathcal{D}_e\right) := \mathbb{E}_{n_e}\left[r_i\right]$$

collected by running the **evaluation policy**

we can do this on-policy estimation, as we have $\mathcal{D}_e$ in our dataset

Protocol for the Evaluation of OPE with Open Bandit Dataset

4. Compare the estimated policy value with the ground-truth to evaluate the OPE estimator, for example, using the *squared error*

squared error of $\hat{V}$ **(performance metric of an OPE estimator; lower = accurate)**

$$:= \left(\underbrace{V_{\mathrm{on}}\left(\pi_e; \mathcal{D}_e\right)}_{\text{ground-truth policy value}} - \underbrace{\hat{V}\left(\pi_e; \mathcal{D}_b\right)}_{\text{value estimated by OPE}} \right)^2$$

By applying this procedure to several estimators,
we can evaluate and compare different OPE methods

Comparison with Existing Real-World Bandit Datasets

Table 2: Comparison of Currently Available Large-scale Bandit Datasets

	Criteo Data (Lefortier et al. 2016)	**Yahoo! Data** (Li et al. 2010)	**Open Bandit Dataset** (ours)
Domain	Display Advertising	News Recommendation	Fashion E-Commerce
#Data	>= 103M	>= 40M	>= 26M (will increase)
#Behavior Policies	1	1	2 (will increase)
Random A/B Test Data	✗	✓	✓
Behavior Policy Code	✗	✗	✓
Evaluation of Bandit Algorithms	✓	✓	✓
Evaluation of OPE	✗	✗	✓
Pipeline Implementation	✗	✗	✓

the existing datasets cannot compare different OPE estimators

Our Open Bandit Dataset

- contains ***multiple*** different policies
- enables the ***evaluation of OPE*** for the first time
- comes with the pipeline implementations (Open Bandit Pipeline)

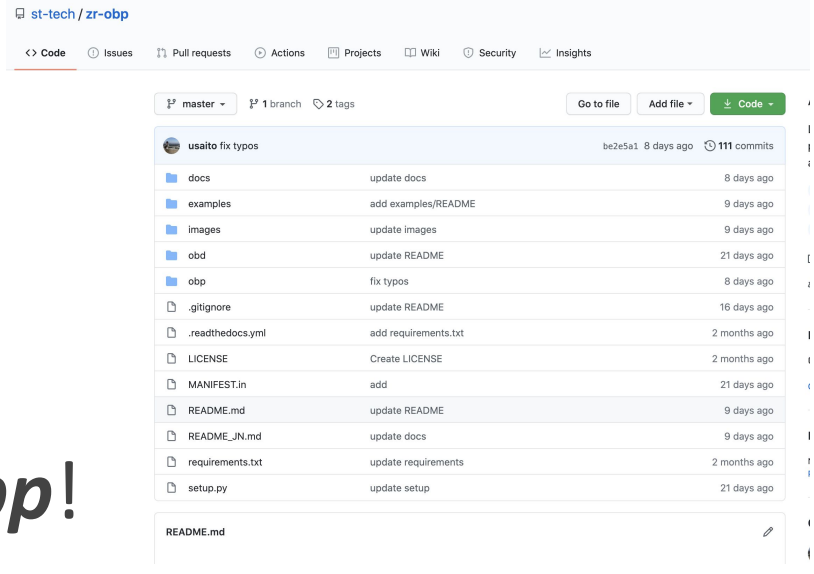
Open Bandit Pipeline (OBP)

We have implemented ***Open Bandit Pipeline (OBP)***

to streamline and standardize experiments on OPE

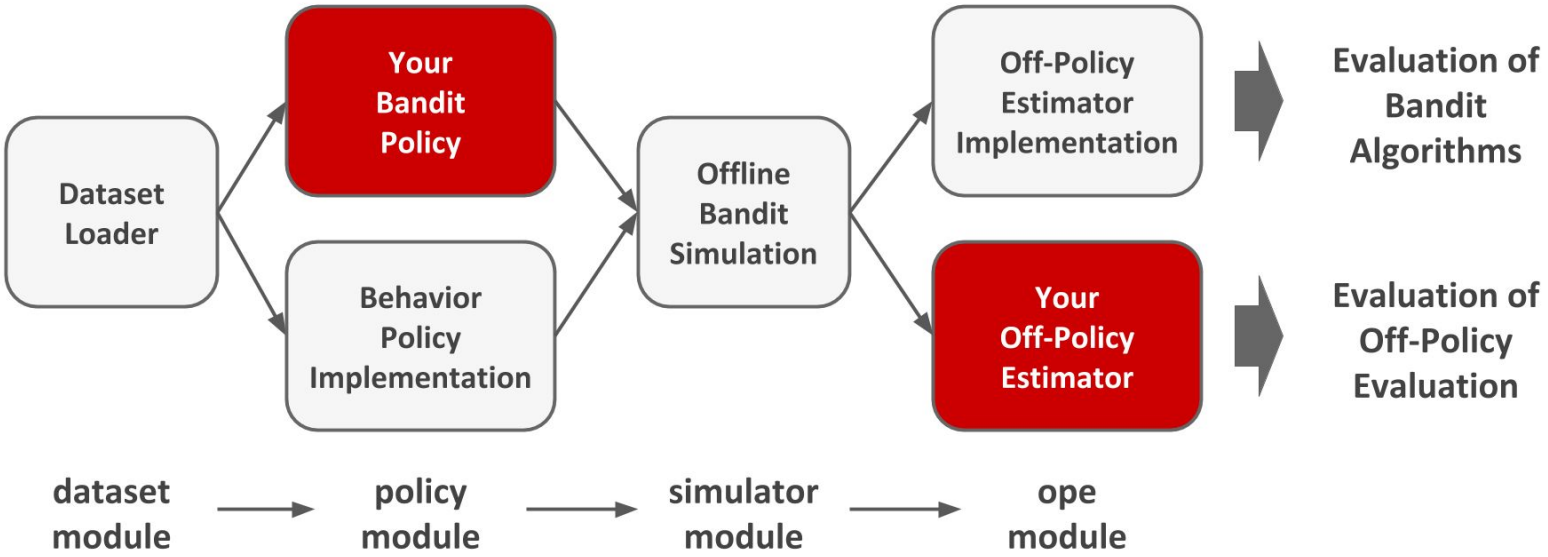
OPEN
BANDIT
PIPELINE™

find out ***zr-obp***!

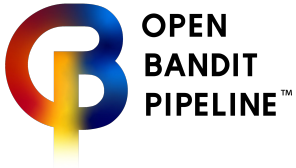


Open Bandit Pipeline: Main Modules

OBP consists of **four main modules** (dataset, policy, simulator, and ope)



Proof of Concept Analysis with Our Data and Pipeline

Live OPE demonstration with 

(our pipeline software)

1. Load and Preprocess Open Bandit Dataset (dataset module)
2. Train a counterfactual policy based on the logged bandit feedback (policy module)
3. Off-Policy Evaluation of the counterfactual policy (ope module)

Q. Should ZOZOTOWN use the counterfactual policy or stick to the current one?

Open Bandit Pipeline: Simple OPE Implementation

We can easily implement OPE itself and experiments on OPE

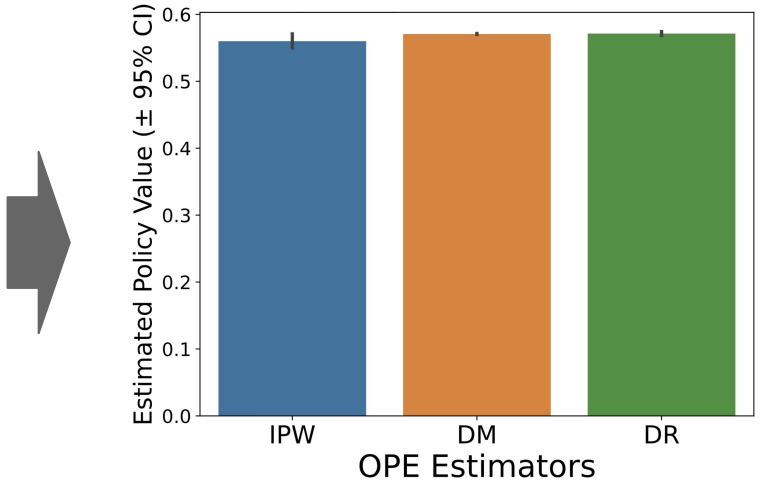
```python
# a case for implementing OPE of the BernoulliTS policy using log data generated by the Random policy
from obp.dataset import OpenBanditDataset
from obp.policy import BernoulliTS
from obp.simulator import run_bandit_simulation
from obp.ope import OffPolicyEvaluation, ReplayMethod

# (1) Data loading and preprocessing
dataset = OpenBanditDataset(behavior_policy='random', campaign='women')
bandit_feedback = dataset.obtain_batch_bandit_feedback()

# (2) Offline Bandit Simulation
counterfactual_policy = BernoulliTS(n_actions=dataset.n_actions, len_list=dataset.len_list)
selected_actions = run_bandit_simulation(bandit_feedback=bandit_feedback, policy=counterfactual_policy)

# (3) Off-Policy Evaluation
ope = OffPolicyEvaluation(bandit_feedback=bandit_feedback, ope_estimators=[ReplayMethod()])
estimated_policy_value = ope.estimate_policy_values(selected_actions=selected_actions)

# estimated performance of BernoulliTS relative to the ground-truth performance of Random
relative_policy_value_of_bernoulli_ts = estimated_policy_value['rm'] / bandit_feedback['reward'].mean()
print(relative_policy_value_of_bernoulli_ts) # 1.120574...
```



Democratizing and Standardizing Off-Policy Evaluation

Open Bandit Pipeline: Documentation

We built a detailed **documentation** of the Open Bandit Pipeline

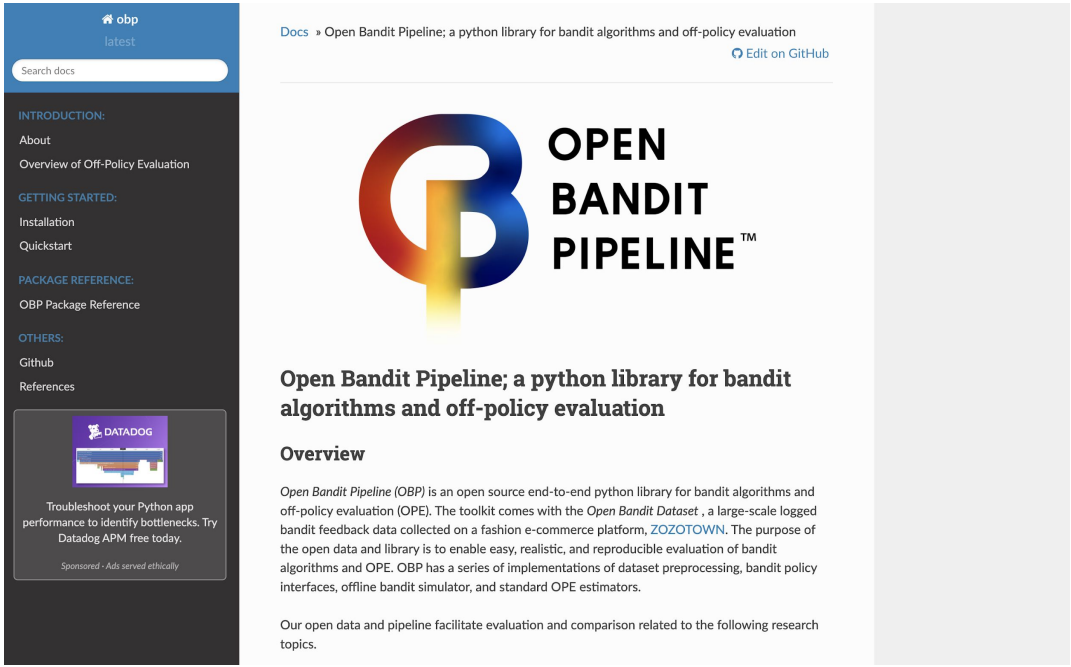


Table of Contents (ToC)

- Overview of OPE
- Evaluation of OPE
- Dataset Description
- Package References
- Related Work

Real-World Applications:
Estimator Selection and Counterfactual Policy Search

Application1: Off-Policy Estimator Selection

ZOZOTOWN

RQ: Which is the best estimator in DM/IPW/SNIPW/DR?

1. Apply the evaluation of OPE protocol on the platform by the software

Table 4: Comparing Relative-Estimation Errors of OPE Estimators (**Random** $\rightarrow$ **Bernoulli TS**)

	Campaigns		
Estimators	All	Men's	Women's
DM	0.23879 [0.22998, 0.24988]	0.24155 [0.22656, 0.25592]	0.22884 [0.22224, 0.23423]
IPW	0.03477 [0.01147, 0.06592]	0.09806 [0.07485, 0.12151]	0.03252 [0.01708, 0.04912]
SNIPW	0.03381 [0.01005, 0.06662]	0.08153 [0.05677, 0.10592]	0.03179 [0.01562, 0.04825]
DR	0.03487 [0.01094, 0.06784]	0.08528 [0.06186, 0.10876]	0.03224 [0.01605, 0.04843]

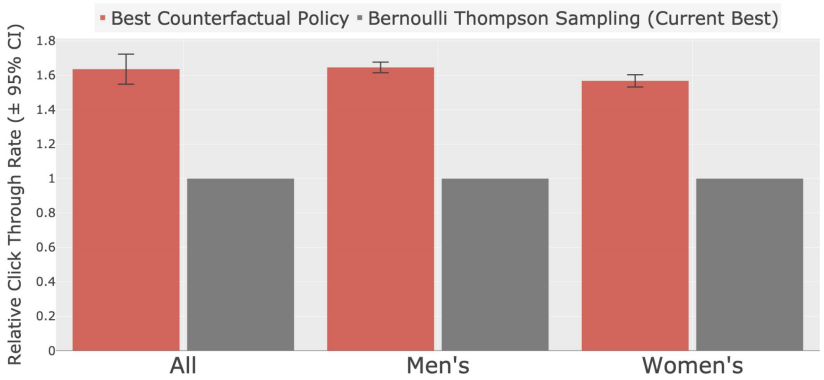
Result: SNIPW is the most accurate estimator on the ZOZOTOWN platform

Application2: Counterfactual Policy Search

ZOZOTOWN

RQ: Can we find the promising counterfactual policy by OPE?

1. Construct a set of candidate counterfactual policies
2. Apply the most accurate estimator (SNIPW) to the candidate policies
3. Compare the counterfactual policy selected by OPE and the current one



Result:

We can identify the counterfactual policy that improves the CTR about 40-60%

Figure 3: Comparing Counterfactual Policy and Bernoulli Thompson Sampling

Summary

- Accurate OPE enables the safe policy improvements in theory
- Experiments in OPE papers are *unrealistic* or *irreproducible*
  - which creates the critical gap in theory and practice
- We release the *open-source dataset/software*
  - enable the *realistic* and *reproducible* experiments on OPE
  - make it easier to use OPE in practice
  - their benefits are verified on our fashion EC platform

Thank you!

email: saito@hanjuku-kaso.com

paper: https://arxiv.org/abs/2008.07146

github: https://github.com/st-tech/zr-obp

docs: https://zr-obp.readthedocs.io/en/latest/

google group: https://groups.google.com/g/open-bandit-project

dataset: https://research.zozo.com/data.html