Large-scale Open Dataset, Pipeline, and Benchmark for Off-Policy Evaluation

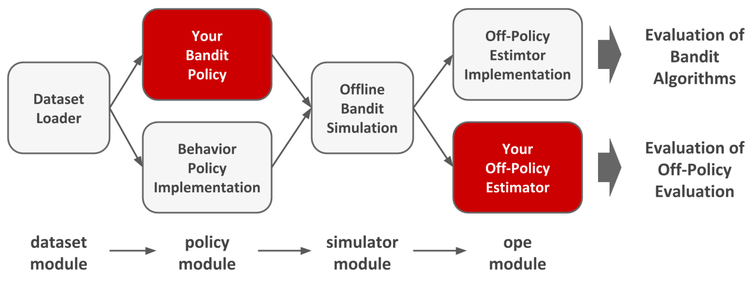
Yuta Saito[1,2], **Shunsuke Aihara**[3], **Megumi Matsutani**[3], **Yusuke Narita**[1,4]
Hanjuku-kaso Co.,Ltd.[1], Tokyo Institute of Technology[2]
ZOZO Technologies, Inc.[1], Yale University[2]

Abstract

- We build and publicize the *Open Bandit Dataset and Pipeline* to facilitate scalable and reproducible research on bandit algorithms. They are especially suitable for *off-policy evaluation* (OPE), which attempts to predict the performance of hypothetical algorithms using data generated by a different algorithm in use.

- We construct the dataset based on experiments and implementations on a large-scale fashion e-commerce platform, ZOZOTOWN. The data contain the ground-truth about the performance of several bandit policies and enable the fair comparisons of different OPE estimators. To streamline and standardize the analysis of the Open Bandit Dataset, we also provide the *Open Bandit Pipeline*, a series of implementations of dataset preprocessing, behavior bandit policy simulators, and OPE estimators. The figure below illustrates the structure of our pipeline package. Our open data and pipeline will allow researchers and practitioners to easily evaluate and compare their bandit algorithms and OPE estimators with others in a large, real-world setting.

- Our pipeline and sample data are available at https://github.com/st-tech/zr-obp



Note: **Structure of Open Bandit Pipeline**

[1]https://corp.zozo.com/en/about/profile/
[2]https://zr-obp.readthedocs.io/en/latest/index.html

Open Bandit Dataset and Pipeline

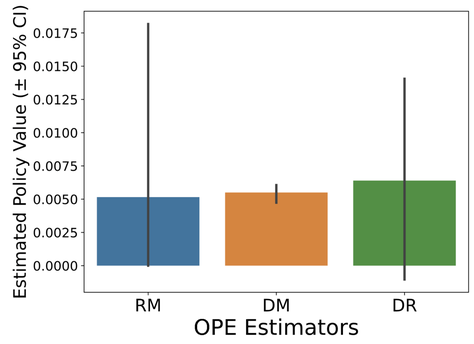
Our real-world open data is logged bandit feedback data we call the Open Bandit Dataset. The dataset is provided by ZOZO Inc.[1], the largest Japanese fashion e-commerce company with over 5 billion USD market capitalization (as of May 2020). The company recently started using context-free multi-armed bandit algorithms to recommend fashion items to users in their large-scale fashion e-commerce platform.

We collected the data in a 7-day experiment in late November 2019 on three "campaigns," corresponding to "all", "men's", and "women's" items, respectively. Each campaign randomly uses either the Random algorithm or the Bernoulli Thompson Sampling (Bernoulli TS) algorithm for each user impression. The data is large and contains many millions of recommendation instances. The number of actions is also sizable, so this setting is challenging for bandit algorithms and their OPE.

To facilitate the usage of the Open Bandit Dataset, we also build a toolkit called the Open Bandit Pipeline. Our pipeline contains implementations of dataset preprocessing, behavior policy simulators, and evaluation of OPE estimators. This pipeline allows researchers to focus on building their OPE estimator and easily compare it with other methods in realistic and reproducible ways. To our knowledge, our real-world dataset and pipeline are the first to include multiple behavior policies, their implementations used in production, and their ground-truth policy values. These features enable the evaluation of OPE for the first time. Please refer to the documentation[2] for the detailed descriptions.

```python
# a case for implementing OPE of the BernoulliTS policy using log data generated by the Random polic
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_polic

# (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'].mea
print(relative_policy_value_of_bernoulli_ts) # 1.120574...
```

(a) **Code snippet from Open Bandit Pipeline**



(b) **OPE Results**