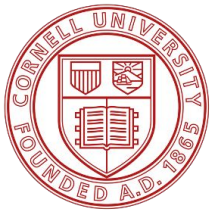
POTEC: Off-Policy Contextual Bandits for Large Action Spaces via Policy Decomposition

ICLR2025 Spotlight

Yuta Saito, Jihan Yao, Thorsten Joachims

Cornell University, University of Washington

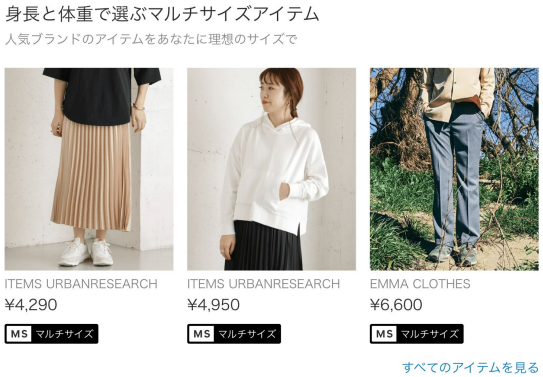


Data-driven decision making in online platforms

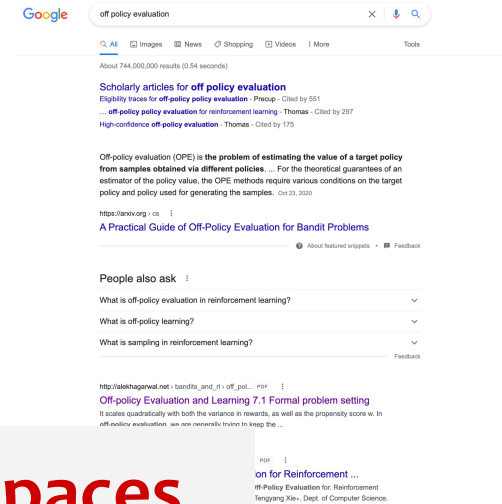
There are many **auto-decision-makings** in today's online platforms

next video suggestions

item recommendations



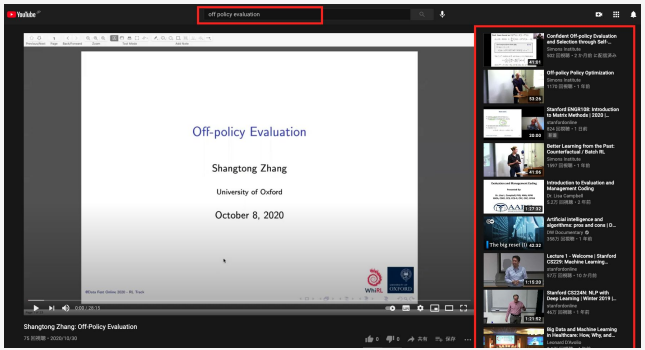
web search



These systems often have large action spaces

Off-Policy Learning (OPL) for Large Action Spaces

e.g., next video suggestion



$\sim \pi_0$

current (logging) system

Logged data collected by
the currently running system

$\mathcal{D}$

Our Goal

How can we learn a new, better policy using only the logged dataset?

π

Standard Off-Policy Learning Setup

OPL aims to maximize the expected reward

$$\max_{\theta} V(\pi_\theta) := \mathbb{E}_{p(x)\pi_\theta(a|x)}[q(x,a)]$$

expected reward function: $q(x,a) = \mathbb{E}[r \mid x, a]$

Using only the historical logged data collected by a logging policy

$$\mathcal{D} := \{(x_i, a_i, r_i)\} \sim \pi_0$$

Baseline: The Regression-based Approach

To deal with the unknown q function,
we often apply the **regression-based** approach

The Regression-based Approach

$$\pi_\theta(a \mid x) := \begin{cases} 1 & (a = \arg\max_{a'} \hat{q}_\theta(x, a')) \\ 0 & (\text{otherwise}) \end{cases}$$

estimated (feasible) reward function

where $\theta = \text{argmin}_\theta \sum_i (r_i - \hat{q}_\theta(x_i, a_i))^2$

However, this approach is sensitive to reward modeling bias

The Policy-based Approach

The **policy-based approach estimates the policy gradient from the logged data** and updates the policy parameters via gradient ascent

The Policy-based Approach

$$\theta_{t+1} \leftarrow \theta_t + \underbrace{\eta}_{\text{learning rate}} \underbrace{\nabla_\theta V\left(\pi_\theta\right)}_{\textbf{policy gradient}}$$

where **the policy gradient** is given as follows (via the log-derivative trick)

$$\nabla_\theta V\left(\pi_\theta\right) := \mathbb{E}_{p(x)\pi_\theta(a|x)}[q(x,a)\nabla_\theta \log \pi_\theta(a|x)]$$

Baselines: The policy-based Approach

True PG: $$\nabla_\theta V(\pi_\theta) := \mathbb{E}_{p(x)\pi_\theta(a|x)}[q(x,a)\nabla_\theta \log \pi_\theta(a|x)]$$

IPS Policy Gradient (IPS-PG)

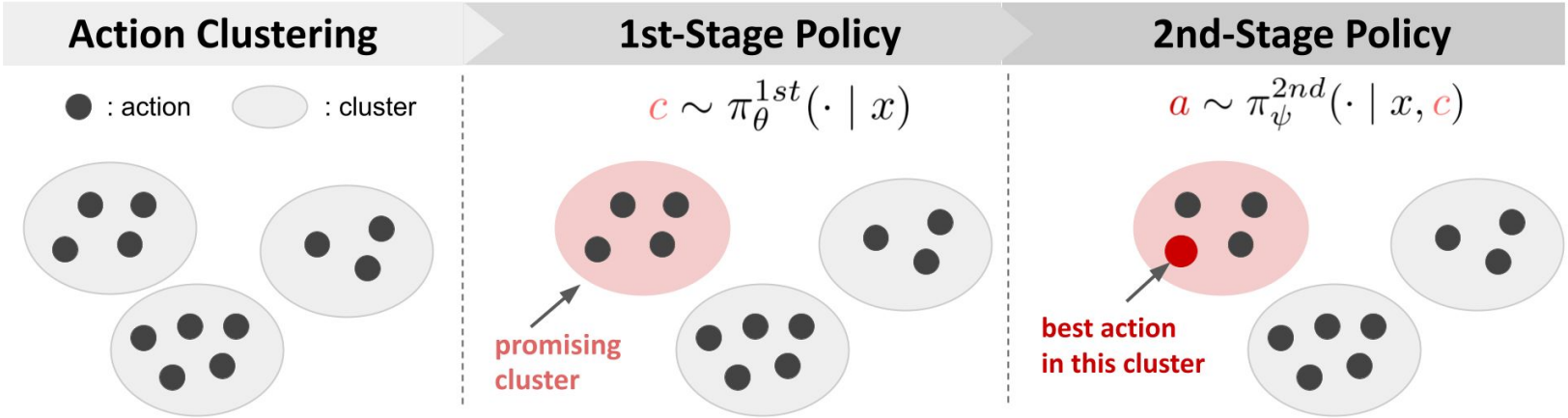
$$\widehat{\nabla_\theta V}_{IPS}(\pi_\theta; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n} \frac{\pi_\theta(a_i \mid x_i)}{\pi_0(a_i \mid x_i)} r_i \nabla_\theta \log \pi_\theta(a_i \mid x_i)$$

vanilla importance weighting

$$\mathcal{D} \sim \pi_0$$

However, this approach suffers from the variance issue particularly for large action spaces

POTEC: A New Two-Stage OPL Algorithm



- **1st-stage policy**: **to identify promising action clusters via policy-based** $\pi_\theta^{1st}(c \mid x)$ **which should have much lower variance than typical IPS**
- **2nd-stage policy**: **reg-based learning of promising actions within** $\pi_\phi^{2nd}(a \mid x, c)$ **each action cluster, easier than typical reg-based method**

POTEC: A New Two-Stage OPL Algorithm

POTEC still aims to solve the same OPL problem

OPL aims to maximize the expected reward

$$\max_{\theta,\phi} V\left(\pi_{\theta,\phi}^{overall}\right) := \mathbb{E}_{p(x)\pi_{\theta,\phi}^{overall}(a|x)}[q(x,a)]$$

However, we decompose the overall policy and train two policies differently

$$\pi_{\theta,\phi}^{overall}(a \mid x) = \sum_{c} \pi_{\theta}^{1st}(c \mid x)\pi_{\phi}^{2nd}(a \mid x, c)$$

policy-based reg-based

Policy-based Learning of the 1st-stage Policy

First, consider **how to train the 1st-stage policy given a 2nd-stage policy**

Policy-based training of the 1st-stage

$$\theta_{t+1} \leftarrow \theta_t + \eta \nabla_\theta V\left(\pi_{\theta,\phi}^{overall}\right)$$

we aim to improve the overall policy

the true policy gradient in this case is given as follows:

$$\nabla_\theta V\left(\pi_{\theta,\phi}^{overall}\right) := \mathbb{E}_{p(x)\pi_\theta^{1st}(c|x)}\left[q^{\pi_\phi^{2nd}}(x,c)\nabla_\theta \log \pi_\theta^{1st}(c \mid x)\right]$$

A policy gradient estimator for the 1st-stage policy

The POTEC gradient estimator

a pre-trained regression model

$$\widehat{\nabla_\theta V}_{POCEM}(\pi_{\theta,\phi}^{overall}) := \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{\pi_\theta^{1st}(c_{a_i} \mid x_i)}{\pi_0^{1st}(c_{a_i} \mid x_i)}\left(r_i - \hat{f}(x_i, a_i)\right)\nabla_\theta \log \pi_\theta^{1st}(c_i \mid x_i) + \mathbb{E}_{\pi_\theta^{1st}(c|x_i)}\left[\hat{f}^{\pi_\phi^{2nd}}(x_i, c)\nabla_\theta \log \pi_\theta^{1st}(c \mid x_i)\right]\right\}$$

- **substantially lower variance due to cluster importance weighting**
  - variance reduction will be larger with a smaller cluster space
- **can optimize the regression model to minimize the bias**

Pairwise Regression to Minimize the Bias

To minimize the bias, we suggest performing pairwise regression

$$\min_{\phi} \sum_{(x,a,b,r_a,r_b)\in\mathcal{D}_{pair}} \Big((r_a - r_b) - (\hat{f}_\phi(x,a) - \hat{f}_\phi(x,b)) \Big)^2$$

parameterized regression model

where the pairwise dataset is defined as:

$$\mathcal{D}_{pair} := \left\{ (x,a,b,r_a,r_b) \mid \begin{array}{l} (x_a,a,r_a),(x_b,b,r_b) \in \mathcal{D} \\ x = x_a = x_b, c(x,a) = c(x,b) \end{array} \right\}$$

(Pairwise) Reg-based Learning of the 2nd-stage Policy

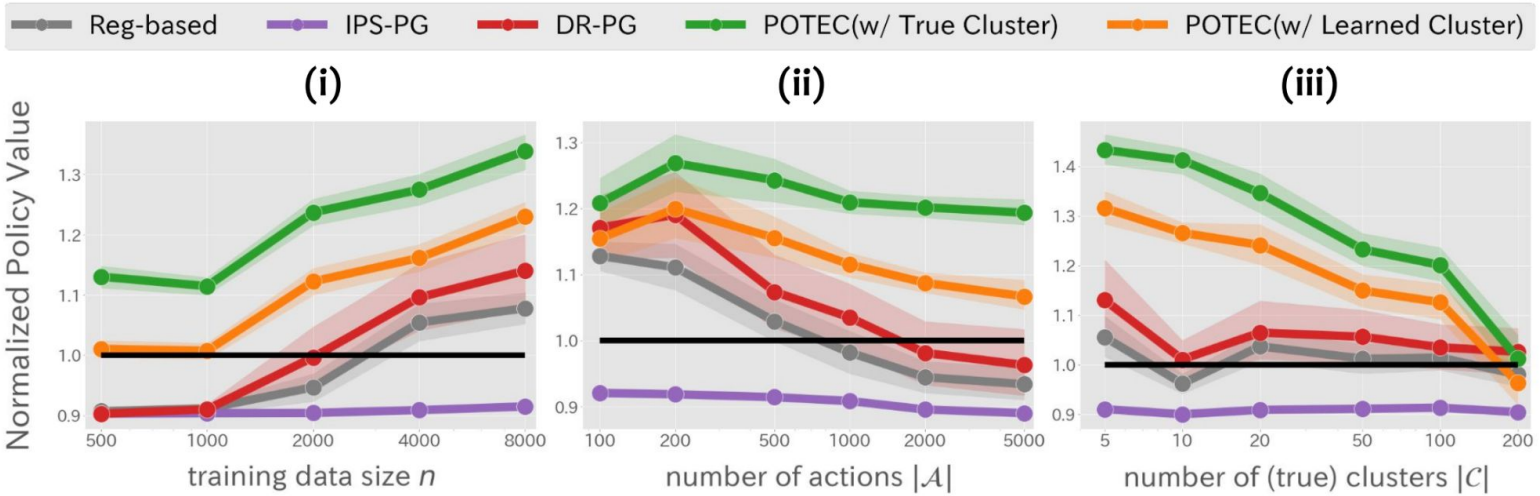
How should we then train the 2nd-stage policy?

Given that 2nd-stage policy aims to identify the best action given a cluster, we can simply use the pre-trained pairwise regressor here

$$\pi_{\phi}^{2nd}(a \mid x, c) = \begin{cases} 1 & (a = \arg\max_{a:c_a=c} \hat{f}_{\phi}(x, a)) \\ 0 & (\text{otherwise}) \end{cases}$$

- **This is more robust to bias than the typical reg-based approach**
- **If local correctness is true, this 2nd-stage policy will be optimal**

Synthetic Experiment Results



Our POTEC algorithm is more sample-efficient and robust to growing action spaces than existing methods

*we observed similar trends in the real-world experiment as well