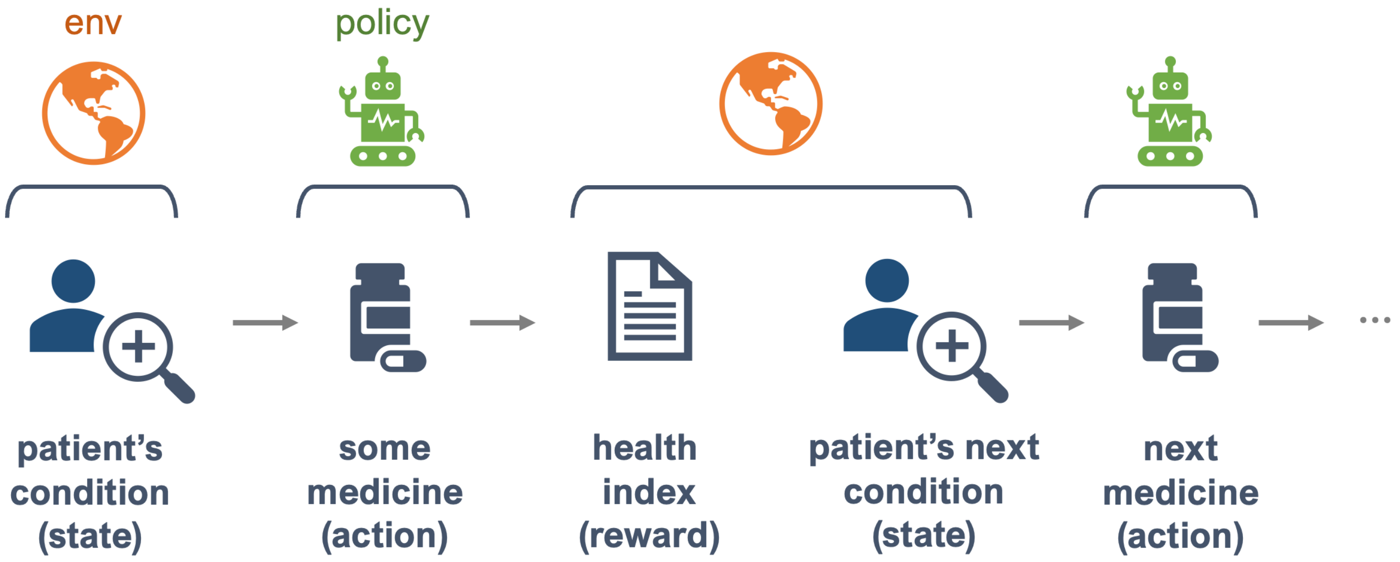
Towards Assessing and Benchmarking Risk-Return Tradeoff of Off-Policy Evaluation

Haruka Kiyohara, Ren Kishimoto, Kosuke Kawakami,
Ken Kobayashi, Kazuhide Nakata, Yuta Saito

Haruka Kiyohara
https://sites.google.com/view/harukakiyohara

Real-world sequential decision making

Example of sequential decision-making in healthcare



Other applications include..

- Robotics
- Education
- Recommender systems
- …

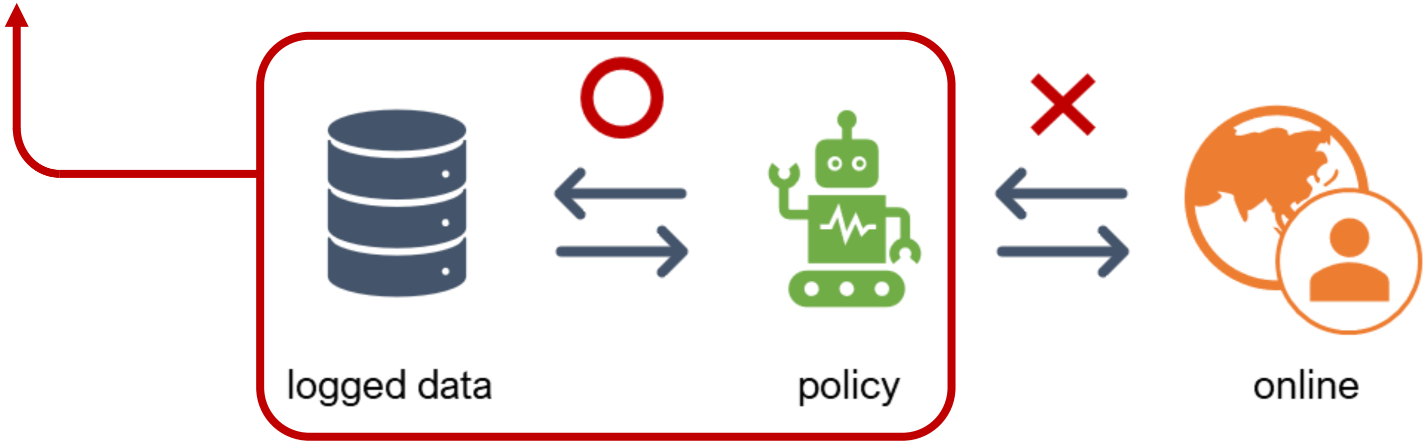
Sequential decision-making is everywhere!

We aim to optimize such decisions as a Reinforcement Learning (RL) problem.

Online and *Offline* Reinforcement Learning (RL)

- Online RL –
  - learns a policy through interaction
  - may harm the real system with bad action choices

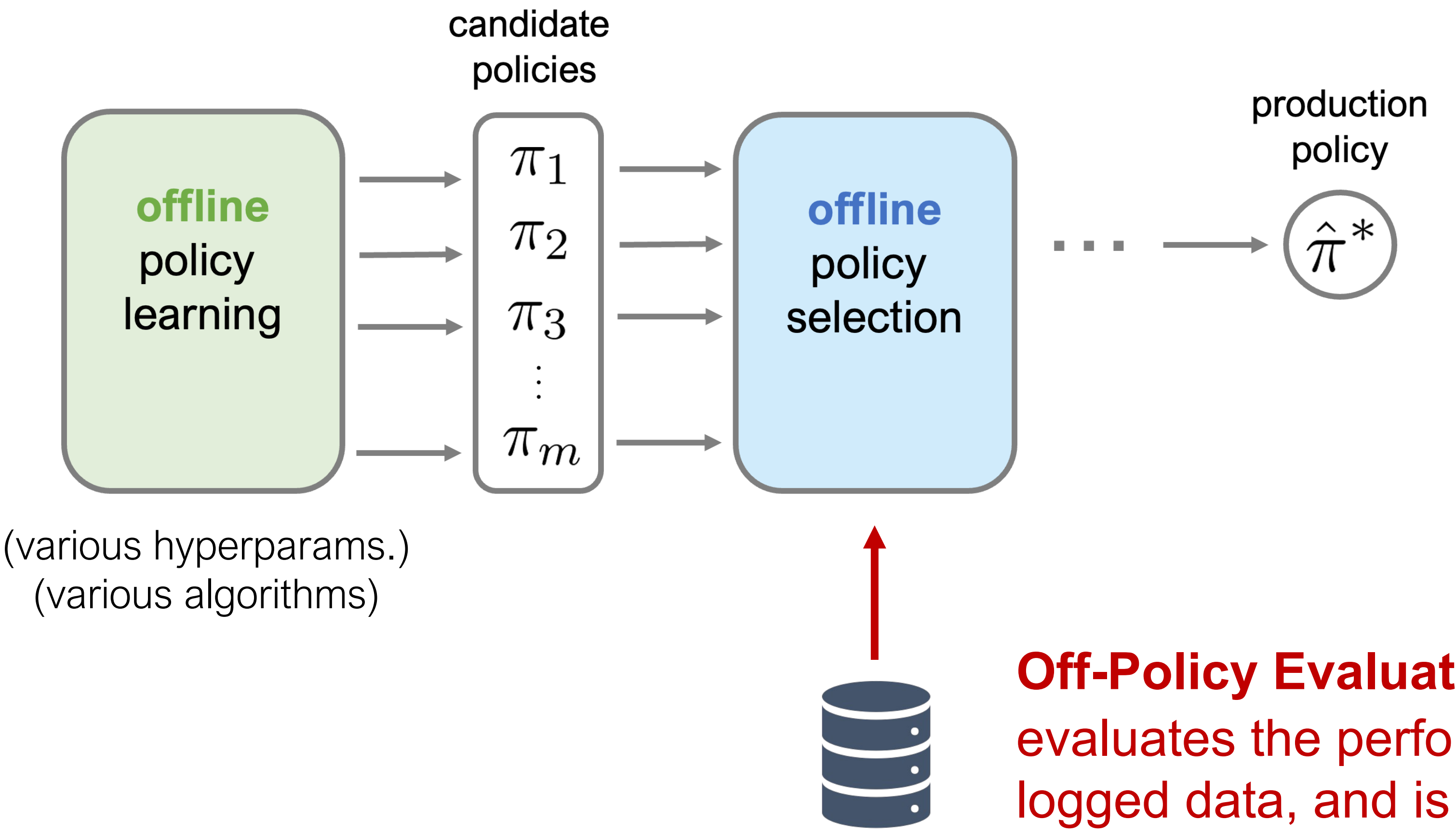
- Offline RL –
  - learns and evaluate a policy solely from offline data
  - can be a safe alternative for online RL



Particularly focusing on Off-Policy Evaluation (OPE)

Why is Off-Policy Evaluation (OPE) important?

The performance of production policy heavily depends on the *policy selection*.



Off-Policy Evaluation (OPE)
evaluates the performance of new policies using logged data, and is used for policy selection.

Content

- Introduction to Off-Policy Evaluation (OPE) of RL policies

- Issues of the existing metrics of OPE

- Our proposal: Evaluating the risk-return tradeoff of OPE via SharpeRatio@k

- Case Study: Why should we use SharpeRatio@k?

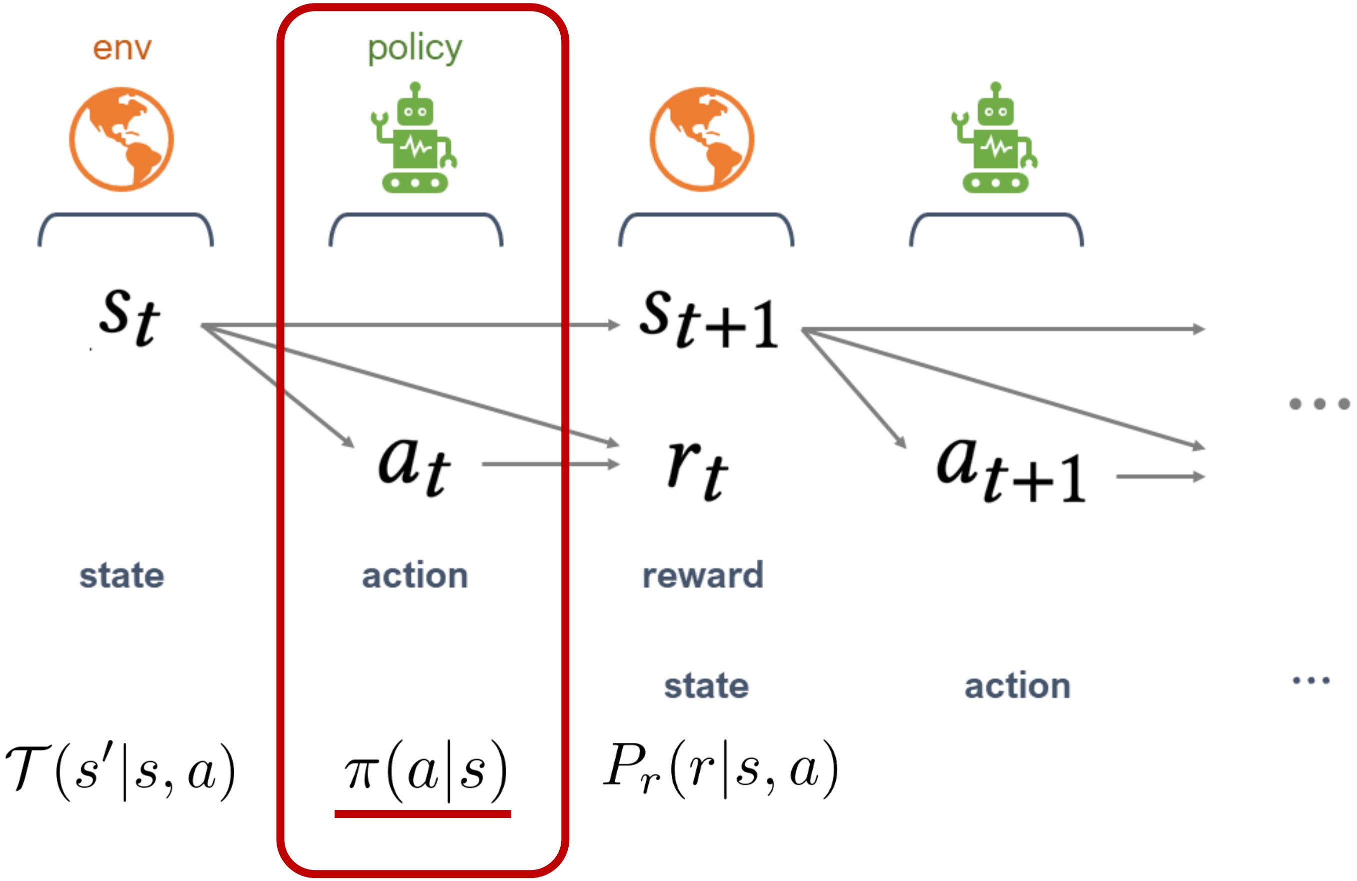
Introduction to Off-Policy Evaluation (OPE)

Preliminary: Markov Decision Process (MDP)

MDP is defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, P_r, \gamma \rangle$.

- $s \in \mathcal{S}$: state
- $a \in \mathcal{A}$: action
- $r \in \mathbb{R}$: reward
- $t = 0, 1, \dots, T-1$: timestep
- $\mathcal{T}(s'|s,a)$: state transition
- $P_r(r|s,a)$: reward function
- $\gamma \in (0, 1]$: discount

▼ our interest



Estimation Target of OPE

We aim to estimate the expected trajectory-wise reward (i.e., policy value):

$$J(\pi) := \mathbb{E}_{p_\pi(\tau)}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right]$$

$$\longrightarrow \underline{\hat{J}}(\pi; \underline{\mathcal{D}}) \approx J(\pi) \leftrightarrow \pi_b$$

OPE estimator

logged data collected by
a past (behavior) policy

*counterfactuals &
distribution shift*

Example of OPE Estimators

We will briefly review the following OPE estimators.

- Direct Method (DM)
- (Per-Decision) Importance Sampling (PDIS)
- Doubly Robust (DR)

- (State-action) Marginal Importance Sampling (MIS)
- (State-action) Marginal Doubly Robust (MDR)

Note: we describe DR and MDR in detail in Appendix.

Direct Method (DM) [Le+,19]

DM trains a value predictor and estimates the policy value from the prediction.

$$\hat{J}_{\mathrm{DM}}(\pi;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}} \pi(a|s_0^{(i)})\hat{Q}(s_0^{(i)}, a)$$

empirical average
(n is the data size and i is the index)

value prediction

$$\hat{Q}(s_t, a_t) \approx R(s_t, a_t) + [\sum_{t'=t+1}^{T-1} \gamma^{t'-t} r_t | s_t, a_t, \pi]$$

estimating expected reward at future timesteps

Pros: variance is small.

Cons: bias can be large when $\hat{Q}$ is inaccurate.

Per-Decision Importance Sampling (PDIS) [Precup+,00]

PDIS applies importance sampling to correct the distribution shift.

$$\hat{J}_{\mathrm{PDIS}}(\pi; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \underbrace{\prod_{t'=0}^{t} \frac{\pi(a_{t'}^{(i)} \mid s_{t'}^{(i)})}{\pi_b(a_{t'}^{(i)} \mid s_{t'}^{(i)})}}_{\text{importance weight}} r_t^{(i)}$$

importance weight

= product of step-wise importance weights

Pros: unbiased (under the common support assumption: $\prod_{t=0}^{T-1} \pi(a_t|s_t) > 0 \rightarrow \prod_{t=0}^{T-1} \pi_b(a_t|s_t) > 0$).

Cons: variance can be exponentially large as t grows.

State-action Marginal IS (MIS) [Uehara+,20]

To alleviate variance, MIS considers IS on the (state-action) marginal distribution.

$$\hat{J}_{\mathrm{SAM-IS}}(\pi; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \underbrace{\hat{\rho}(s_t^{(i)}, a_t^{(i)})}_{\text{(estimated) marginal importance weight}} r_t^{(i)}$$

$$\hat{\rho}(s, a) \approx \underbrace{d^{\pi}(s, a)}_{\text{state-action visitation probability}} / d^{\pi_b}(s, a)$$

Pros: unbiased when $\hat{\rho}$ is correct and reduces variance compared to PDIS.

Cons: accurate estimation of $\hat{\rho}$ is often challenging, resulting in some bias.

Summary of OPE

- Off-Policy Evaluation (OPE) aims to evaluate the expected performance of a policy using only offline logged data.

- However, counterfactual estimation and distribution shift between π and π_b causes either bias or variance issues.

In the following, we discuss..

"How to assess OPE estimators for a reliable policy selection in practice?"

Summary of OPE

- Off-Policy Evaluation (OPE) aims to evaluate the expected performance of a policy using only offline logged data.

- However, counterfactual estimation and distribution shift between π and π_b causes either bias or variance issues.

We discuss the RL settings, but the same idea is applicable to contextual bandits as well.

In the following, we discuss..

"How to assess OPE estimators for a reliable policy selection in practice?"

Issues of the existing metrics of OPE

Conventional metrics focus on “accuracy”

There are three metrics used to assess the accuracy of OPE and policy selection.

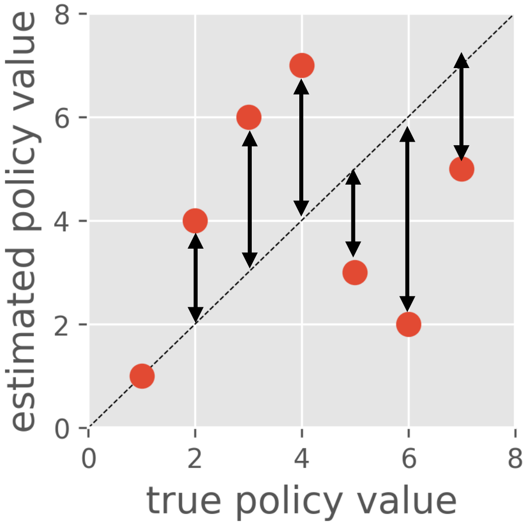
- Mean squared error (MSE) – “accuracy” of policy evaluation
- Rank correlation (RankCorr) – “accuracy” of policy alignment
- Regret – “accuracy” of policy selection

Conventional metrics focus on “accuracy”

There are three metrics used to assess the accuracy of OPE and policy selection.

- Mean squared error (MSE) – “accuracy” of policy evaluation [Voloshin+,21]

$$\frac{1}{|\Pi|}\sum_{\pi\in\Pi}\left(\underbrace{\hat{J}(\pi;\mathcal{D})}_{\text{estimation}} - \underbrace{J(\pi)}_{\text{true value}}\right)^2$$

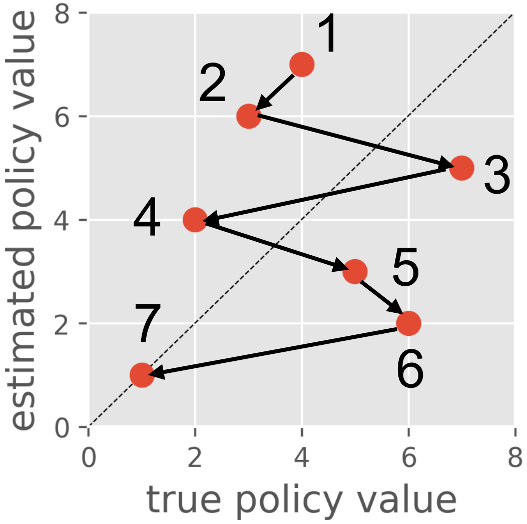


Conventional metrics focus on “accuracy”

There are three metrics used to assess the accuracy of OPE and policy selection.

- **Rank correlation (RankCorr)** – “accuracy” of policy alignment [Fu+,21]

$$\frac{\mathrm{cov}(R_{\hat{J}}(\Pi), R_J(\Pi))}{\underbrace{\mathrm{std}(R_{\hat{J}}(\Pi))}_{\text{estimation}}\,\underbrace{\mathrm{std}(R_J(\Pi))}_{\text{true ranking}}}$$

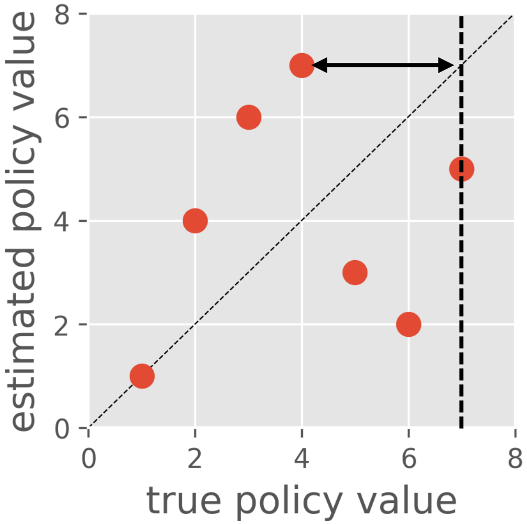


Conventional metrics focus on “accuracy”

There are three metrics used to assess the accuracy of OPE and policy selection.

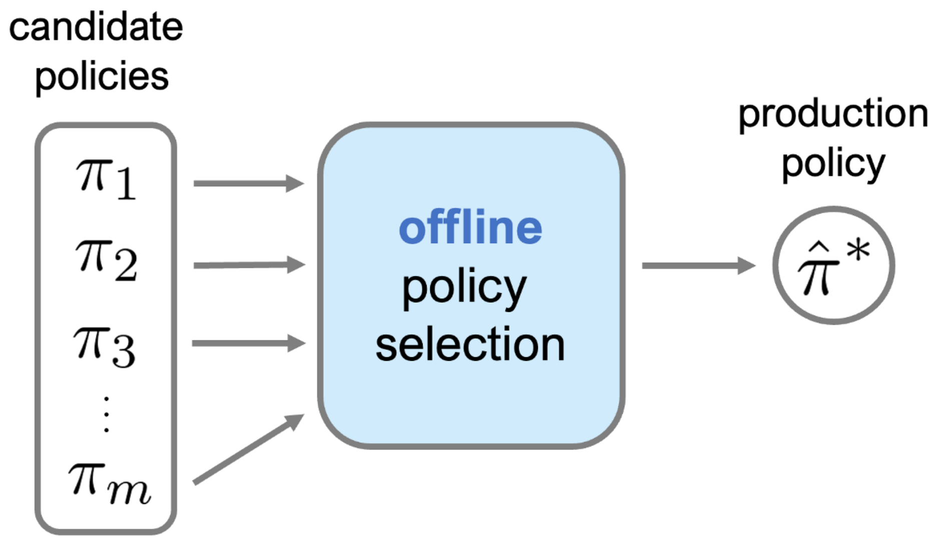
- **Regret** – “accuracy” of policy selection [Doroudi+,18]

$$\underbrace{\max_{\pi \in \Pi} J(\pi)}_{\text{performance of the true best policy}} - \underbrace{\max_{\pi \in \Pi_k(\hat{J})} J(\pi)}_{\text{performance of the estimated best policy}}$$



Existing metrics are suitable for the top-1 selection

Three metrics can assess how likely an OPE estimator chooses a near-best policy.



candidate policies

π_1

π_2

π_3

$\vdots$

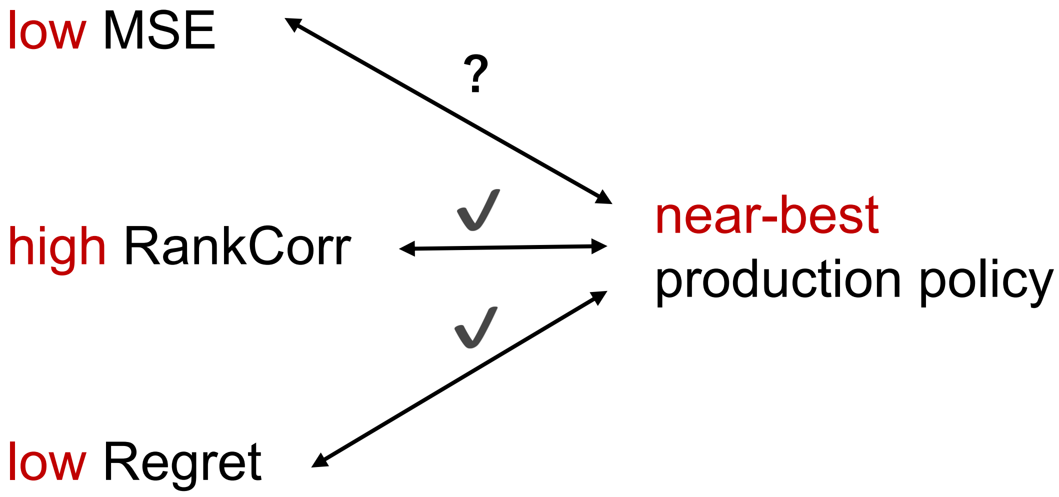
π_m

offline policy selection

production policy

$\hat{\pi}^*$

directly chooses the production policy via OPE



low MSE

?

high RankCorr

✓

✓

low Regret

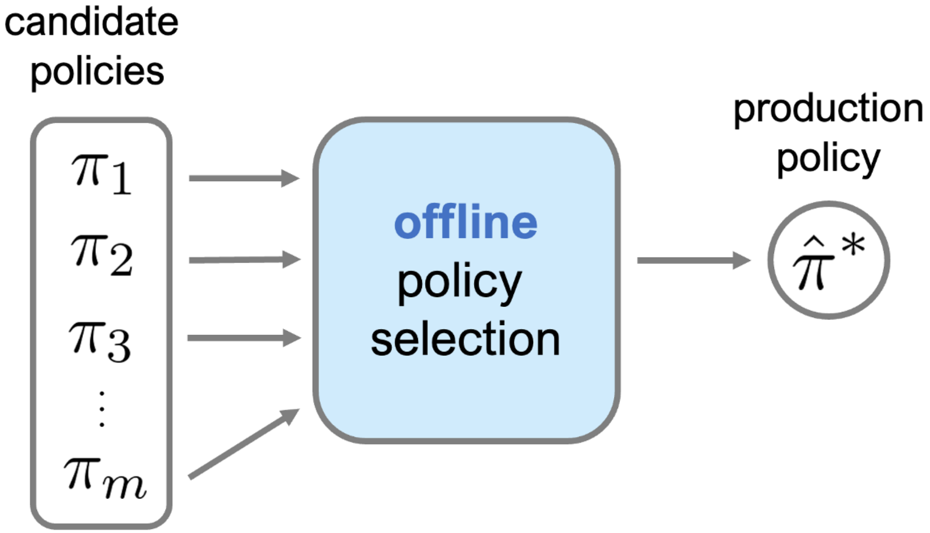
near-best production policy

assessment of OPE

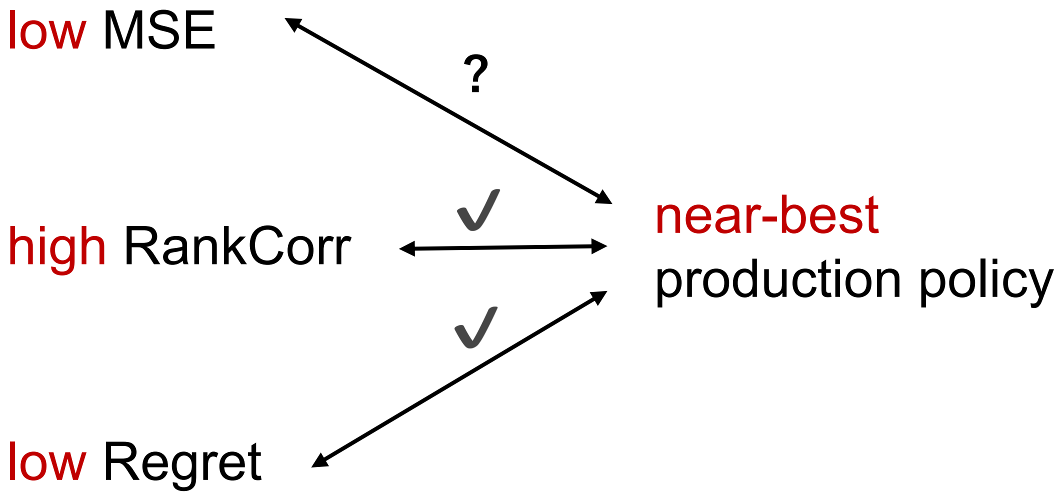
Existing metrics are suitable for the top-1 selection

Three metrics can assess how likely an OPE estimator chooses a near-best policy.

.. but in practice, we cannot sorely rely on the OPE result.



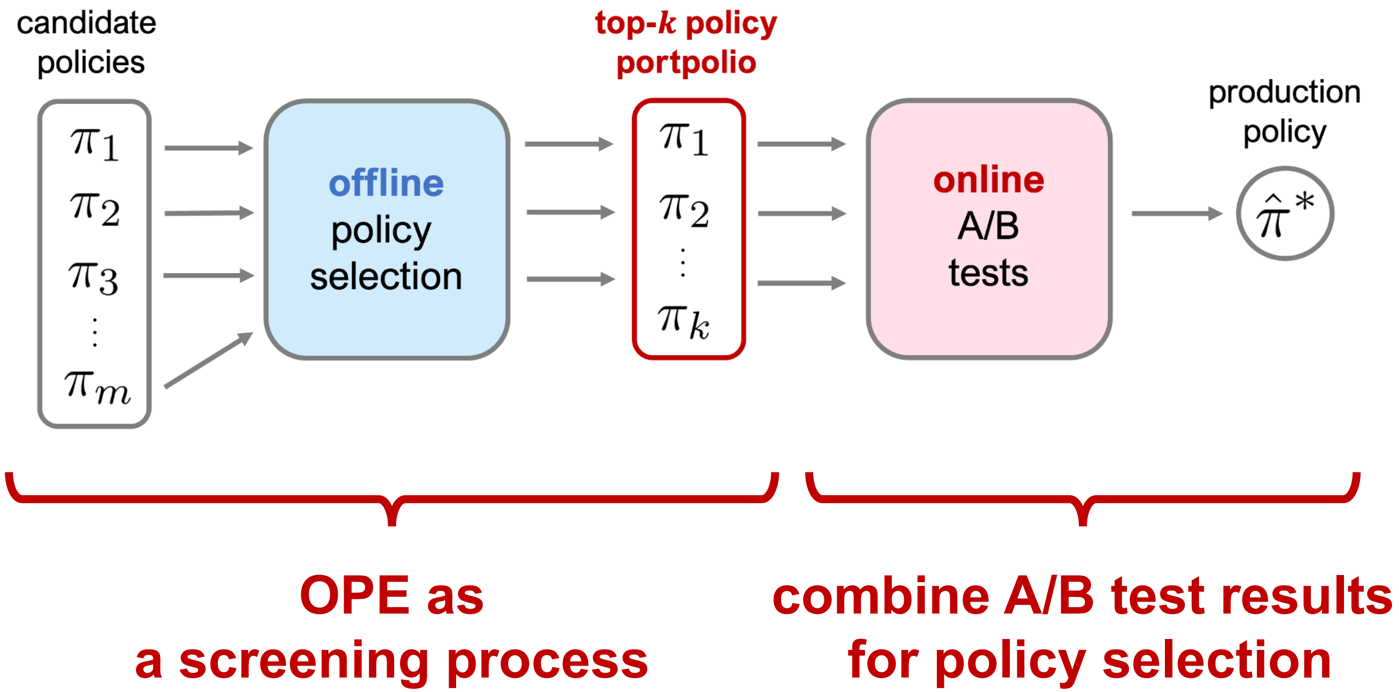
directly chooses the production policy via OPE



assessment of OPE

Research question: How to assess the top-k selection?

We consider the following two-stage policy selection for practical application:

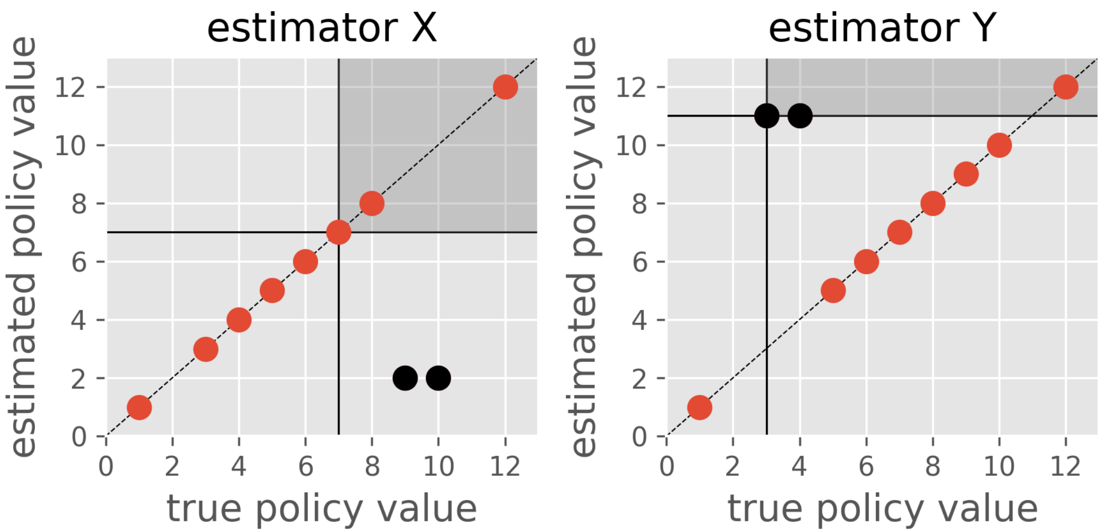


- Are existing metrics enough to assess the top-k policy selection?
- How should we assess OPE estimators accounting safety during A/B tests?

…

Existing metrics fail to distinguish two estimators (1/2)

Three existing metrics report almost the same values for the estimators X and Y.



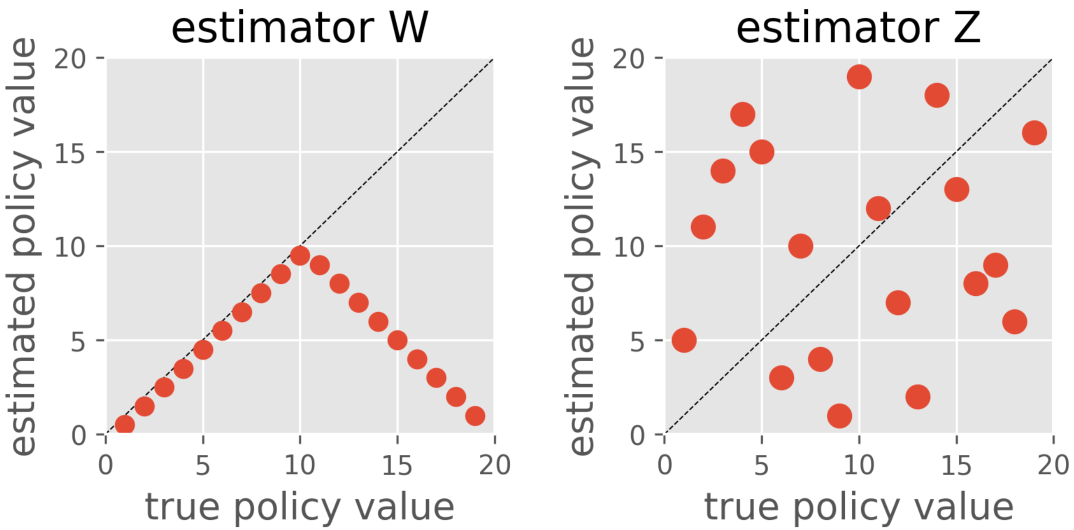
	estimator X	estimator Y
MSE	11.3	11.3
RankCorr	0.413	0.413
Regret	0.0	0.0

Top-3 policy portfolio is very different from each other.

Existing metrics fail to distinguish ***underestimation vs. overestimation.***

Existing metrics fail to distinguish two estimators (2/2)

Three existing metrics report almost the same values for the estimators W and Z.



estimator Z is uniform random and thus is riskier.

	estimator W	estimator Z
MSE	60.1	58.6
RankCorr	0.079	0.023
Regret	9.0	9.0

Existing metrics fail to distinguish ***conservative vs. high-stakes.***

Summary of the existing metrics

- Existing metrics focus on "accuracy" of OPE or the downstream policy selection.

- However, they are not quite suitable for the practical top-k policy selection.
  - Existing metrics cannot take the risk of deploying poor policies into account.
  - Existing metrics fail to distinguish very different OPE estimators:
    - (overestimation vs. underestimation) and (conservative vs. high-stakes)

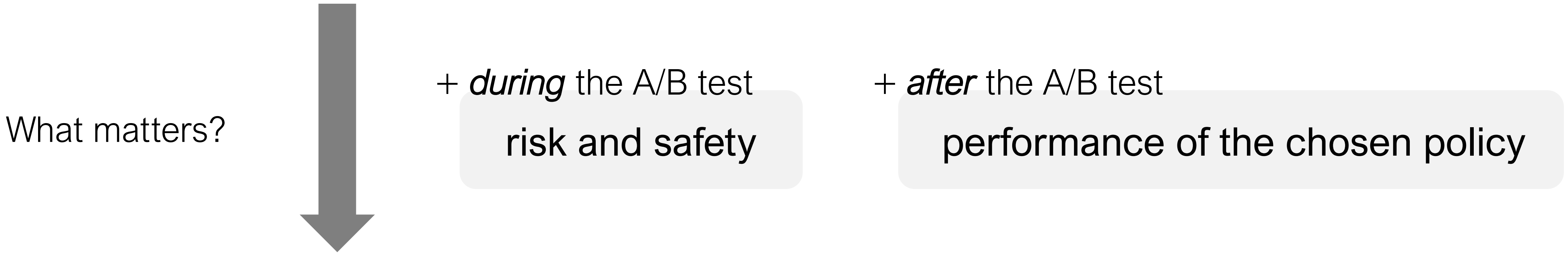
How to assess OPE estimators for the top-k policy selection?

Our proposal: Evaluating the risk-return tradeoff of OPE via SharpeRatio@k

What is the desirable property of the top-k metric?

Existing metrics did **not** consider:

the risk of deploying poor performing policies in online A/B tests



What matters?

+ *during* the A/B test

risk and safety

+ *after* the A/B test

performance of the chosen policy

A new metric should tell:

whether an OPE estimator is *efficient* wrt the risk-return tradeoff

Proposed metric: SharpeRatio@k

Inspired by the portfolio management in finance, we define SharpeRatio in OPE.

$$\textbf{SharpeRatio@k}(\hat{J}) := \frac{\mathrm{best@}k(\hat{J}) - J(\pi_b)}{\mathrm{std@}k(\hat{J})}$$

$$\mathrm{best@}k(\hat{J}) := \max_{\pi \in \Pi_k(\hat{J})} J(\pi)$$

The best policy performance among the top-k policies.

$$\mathrm{std@}k(\hat{J}) := \sqrt{\frac{1}{k} \sum_{\pi \in \Pi_k(\hat{J})} \left(J(\pi) - \left(\frac{1}{k} \sum_{\pi \in \Pi_k(\hat{J})} J(\pi) \right) \right)^2}$$

Standard deviation among the top-k policies.

Proposed metric: SharpeRatio@k

Inspired by the portfolio management in finance, we define SharpeRatio in OPE.

$$\textbf{SharpeRatio@k}(\hat{J}) := \frac{\mathrm{best@}k(\hat{J}) - J(\pi_b)}{\mathrm{std@}k(\hat{J})}$$

$\mathrm{best@}k(\hat{J}) - J(\pi_b)$ measures the **return** over the risk-free baseline.

$\mathrm{std@}k(\hat{J})$ measures the **risk** experienced during online A/B tests.

Example: Calculating SharpeRatio@3

Let’s consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	1.8	?
candidate 2	1.2	?
candidate 3	1.0	?
candidate 4	0.8	?
candidate 5	0.5	?

Example: Calculating SharpeRatio@3

Let’s consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	1.8	?
candidate 2	1.2	?
candidate 3	1.0	?
candidate 4	0.8	?
candidate 5	0.5	?

A/B test

Example: Calculating SharpeRatio@3

Let’s consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	1.8	**2.0**
candidate 2	1.2	**0.5**
candidate 3	1.0	**1.2**
candidate 4	0.8	?
candidate 5	0.5	?

denominator

$= \text{best@}k - J(\pi_b)$

$= 2.0 - 1.0 = 1.0$

Example: Calculating SharpeRatio@3

Let’s consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	1.8	**2.0**
candidate 2	1.2	**0.5**
candidate 3	1.0	**1.2**
candidate 4	0.8	?
candidate 5	0.5	?

denominator

$= \text{best}@k - J(\pi_b)$

$= 2.0 – 1.0 = 1.0$

numerator

$= \text{std}@k$

$= \sqrt{1/k \sum_{i=1}^{k} (J(\pi_i) - \text{mean}@k)^2}$

$= 0.75$

Example: Calculating SharpeRatio@3

Let's consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	1.8	**2.0**
candidate 2	1.2	**0.5**
candidate 3	1.0	**1.2**
candidate 4	0.8	?
candidate 5	0.5	?

denominator

$= \text{best@}k - J(\pi_b)$

$= 2.0 - 1.0 = 1.0$

numerator

$= \text{std@}k$

$= \sqrt{1/k \sum_{i=1}^{k} (J(\pi_i) - \text{mean@}k)^2}$

$= 0.75$

SharpeRatio = 1.0 / 0.75 = 1.33..

Example: Calculating SharpeRatio@3

Let's consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	**1.8**	**2.0**
candidate 2	**1.2**	**0.5**
candidate 3	**1.0**	**1.2**
candidate 4	0.8	?
candidate 5	0.5	?

SharpeRatio = 1.33..

value estimated by OPE	true value of the policy
-	1.0
1.8	**2.0**
0.8	?
1.0	**1.2**
1.2	**1.0**
0.5	?

SharpeRatio = 1.92..

Example: Calculating SharpeRatio@3

Let’s consider the case of performing top-3 policy selection.

policy	value estimated by OPE	true value of the policy
behavior π_b	-	1.0
candidate 1	**1.8**	**2.0**
candidate 2	**1.2**	**0.5** ←
candidate 3	**1.0**	**1.2**
candidate 4	0.8	?
candidate 5	0.5	?

value estimated by OPE	true value of the policy
-	1.0
1.8	**2.0**
0.8	?
1.0	**1.2**
1.2	**1.0** ←
0.5	?

SharpeRatio = 1.33..

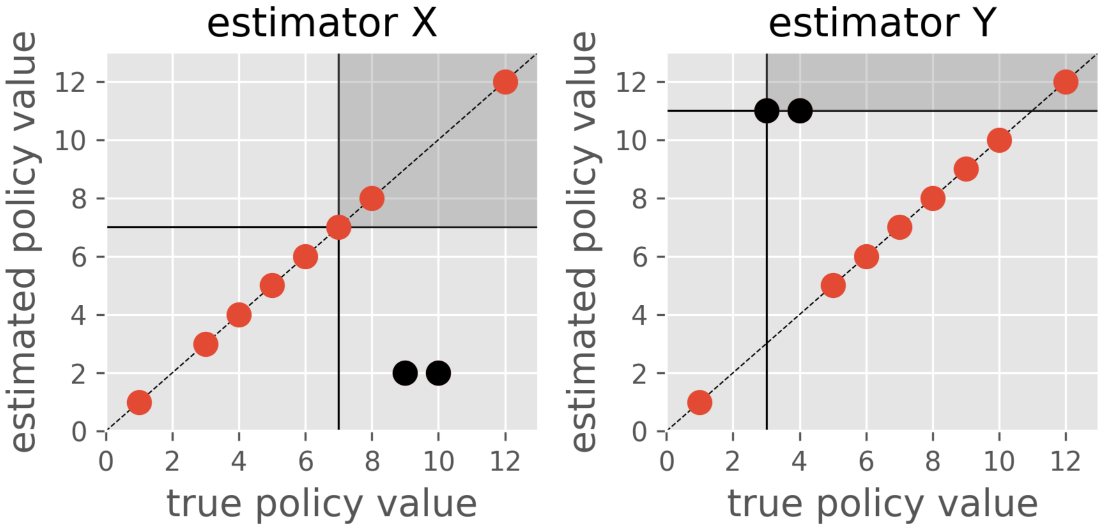
SharpeRatio = 1.92..

Lower risk of deploying detrimental policies!

Case study

SharpeRatio enables informative assessments (1/2)

Let’s compare the case where the existing metrics failed to distinguish the two.



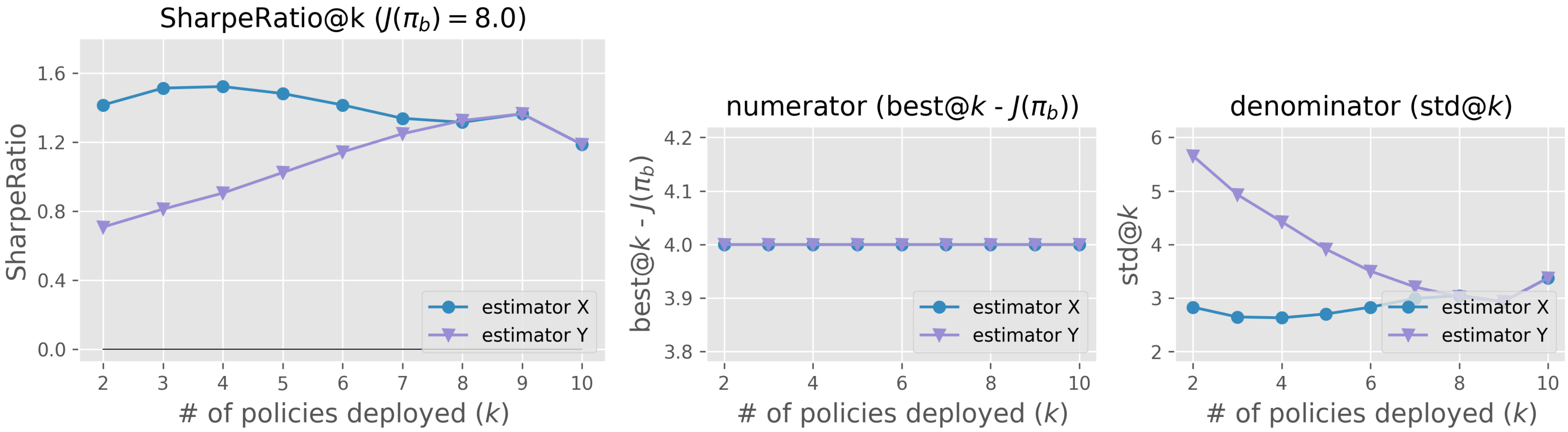
	estimator X	estimator Y
MSE	11.3	11.3
RankCorr	0.413	0.413
Regret	0.0	0.0

Top-3 policy portfolio is very different from each other.

Can SharpeRatio tell the difference in ***underestimation vs. overestimation?***

SharpeRatio enables informative assessments (1/2)

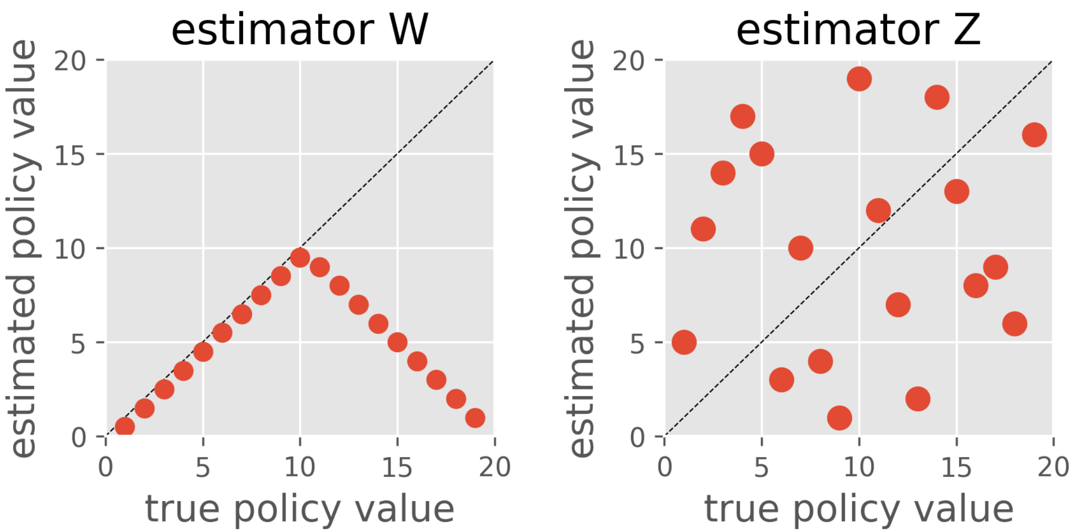
Let's compare the case where the existing metrics failed to distinguish the two.



SharpeRatio values the *safer* estimator more than the riskier estimator.

SharpeRatio enables informative assessments (2/2)

Three existing metrics reports almost the same values for the estimators W and Z.



	estimator W	estimator Z
MSE	60.1	58.6
RankCorr	0.079	0.023
Regret	9.0	9.0

estimator Z is uniform random and thus is riskier.

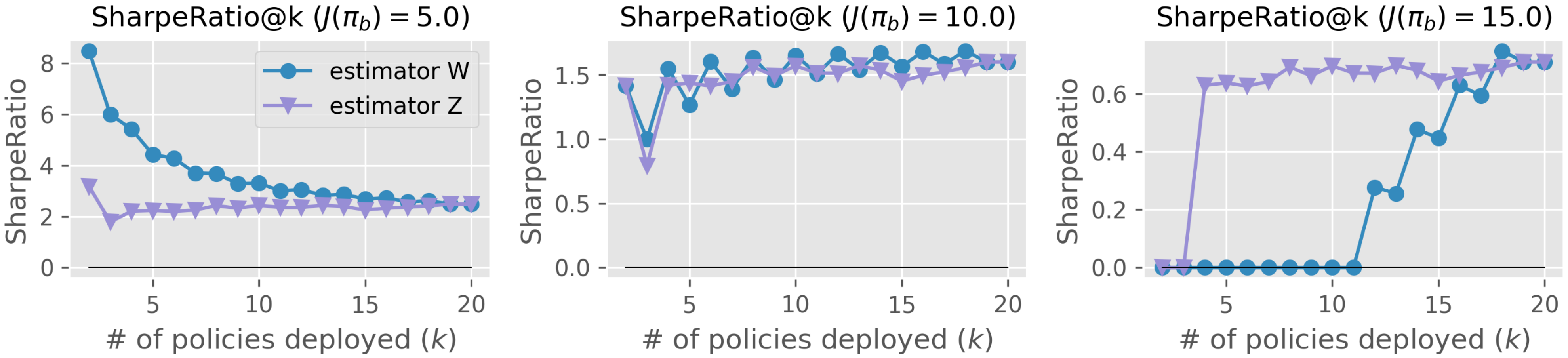
Can SharpeRatio tell the difference in ***conservative vs. high-stakes?***

SharpeRatio enables informative assessments (1/2)

Let’s compare the case where the existing metrics failed to distinguish the two.

baseline is low

baseline is high



Conservative does not deploy poor-performing policies.

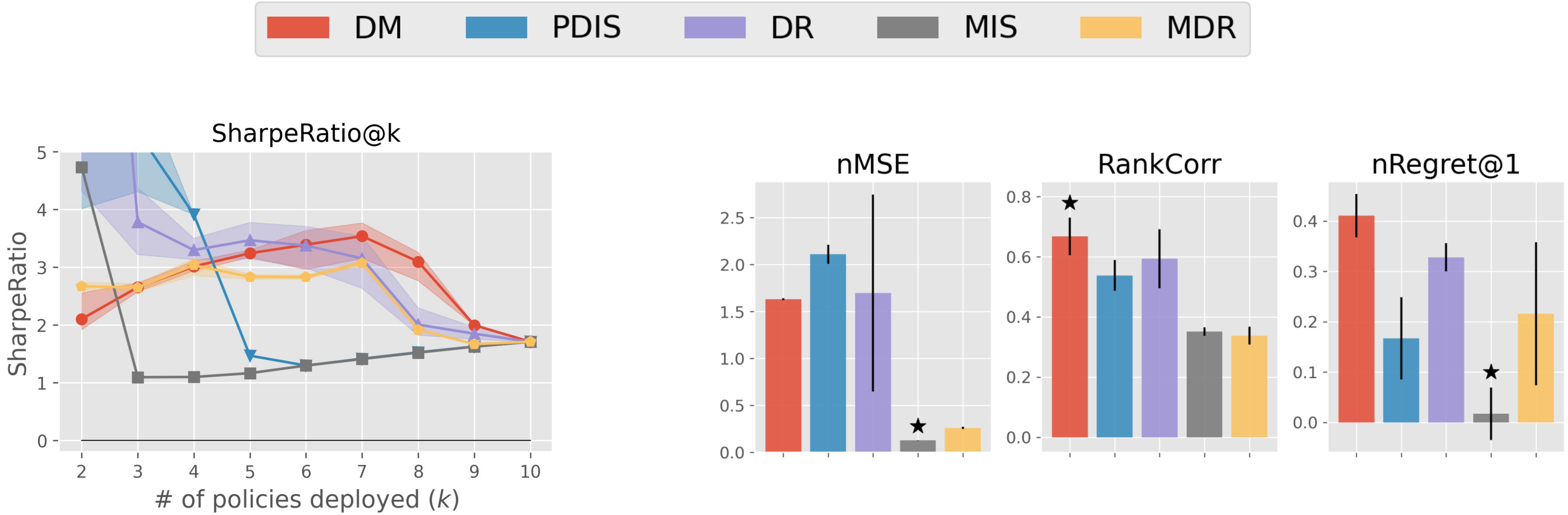
High-stakes potentially improves the baseline.

SharpeRatio identifies *efficient* estimator taking the problem instance into account.

(i.e., performance of the behavior policy)

Experiments with gym

Interestingly, SharpeRatio and existing metrics report very different results.



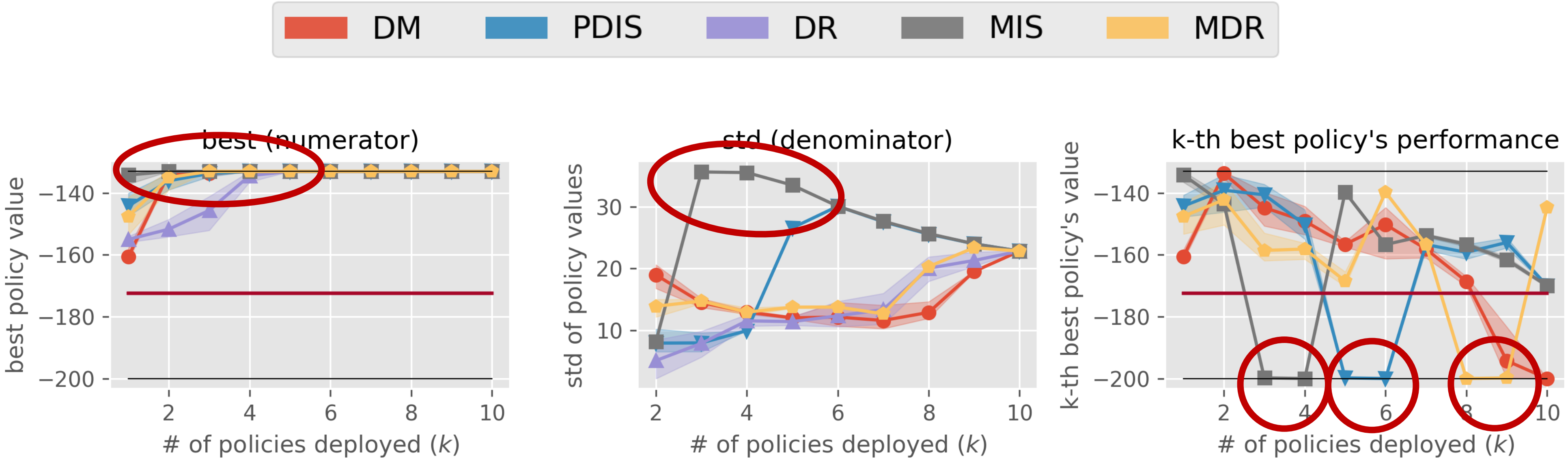
SharpeRatio values **PDIS** for k=2,..,4,
while values **DM** for k=6,..,11.

MSE and Regret values **MIS,** RankCorr evaluates **DM** highly.
RankCorr also evaluates **PDIS** higher than **MDR.**

Note: we use self-normalized variants of OPE estimators.

Experiments with gym (analysis)

SharpeRatio automatically considers the risk of deploying poor policies!



- MSE and Regret chooses **MIS**, which deploys a detrimental policy with small values of k.
- RankCorr chooses a relatively safe one (**DM**), but evaluates riskier **PDIS** higher than **MDR** for $k \geq 5$.
- SharpeRatio detects unsafe behaviors by discounting the return by the risk (std).

Summary

- OPE is often used for screening top-k policies deployed in online A/B tests.

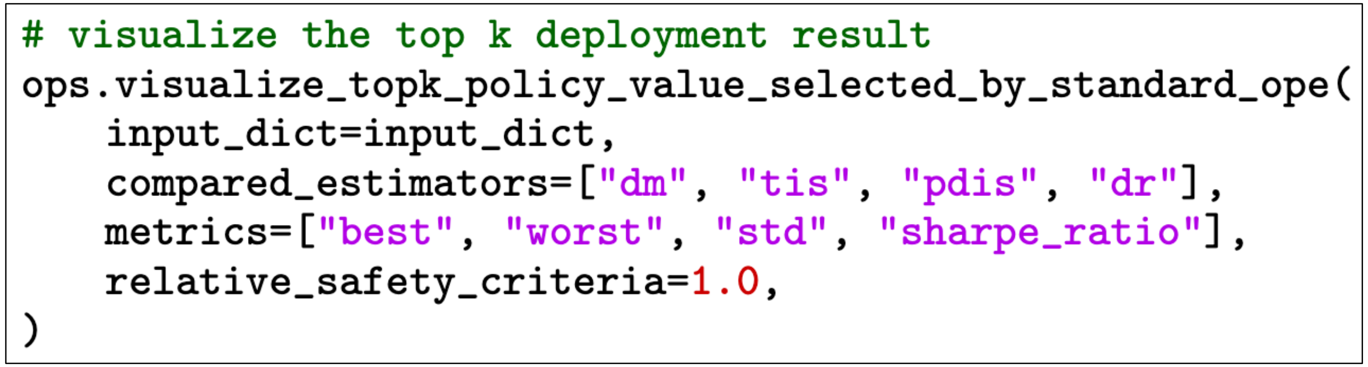
- The proposed SharpeRatio metric measures the **efficiency** of OPE estimator wrt **the risk-return tradeoff**.

- In particular, SharpeRatio can identify a **safe** OPE estimator over a risky counterpart, while also telling an **efficient** OPE estimator **taking the problem instance into account**.

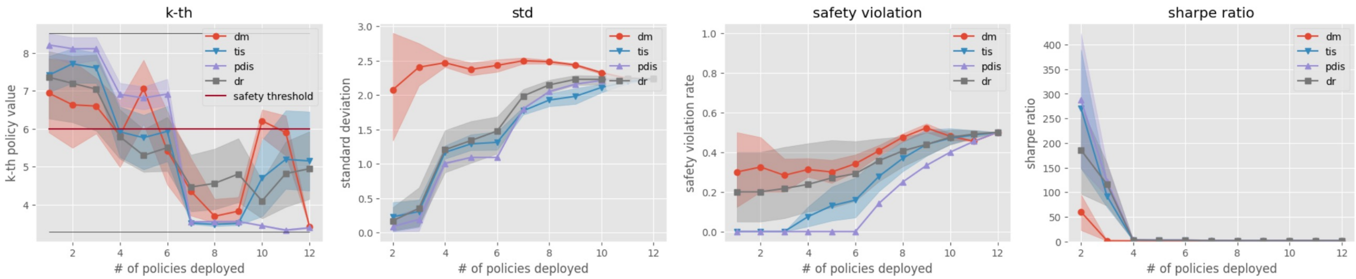
SharpeRatio is an informative assessment metric to compare OPE estimators.

SharpeRatio is available at the SCOPE-RL package!

SharpeRatio is implemented SCOPE-RL and can be used with a few lines of code.



```
# visualize the top k deployment result
ops.visualize_topk_policy_value_selected_by_standard_ope(
    input_dict=input_dict,
    compared_estimators=["dm", "tis", "pdis", "dr"],
    metrics=["best", "worst", "std", "sharpe_ratio"],
    relative_safety_criteria=1.0,
)
```



Install now!!

SCOPE-RL

GitHub

documentation

Thank you for listening!

contact: hk844@cornell.edu

Corresponding papers

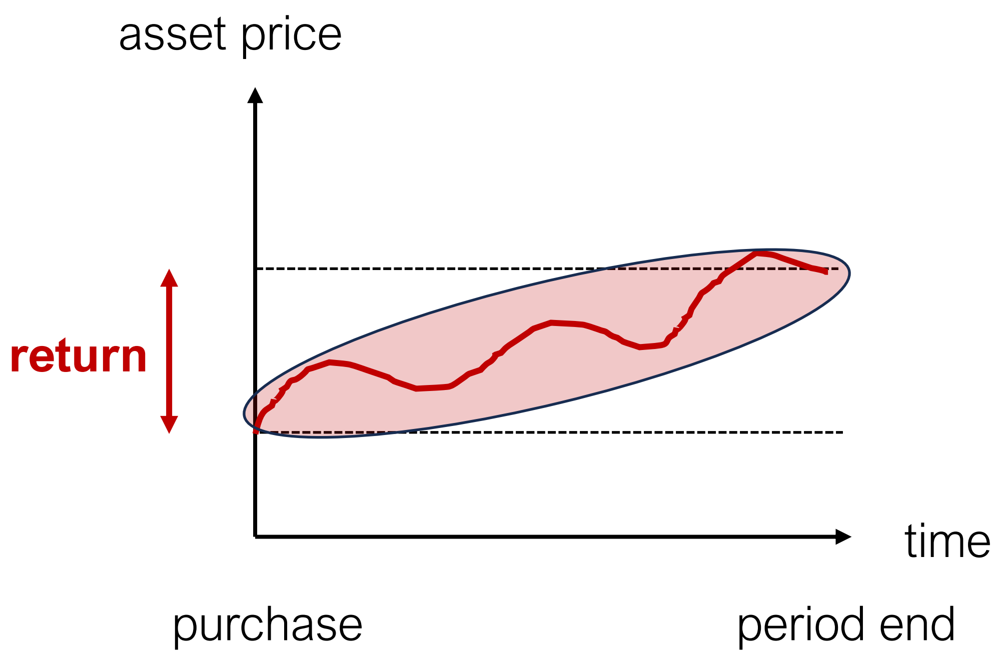
1. “Towards Assessing and Benchmarking the Risk-Return Tradeoff of Off-Policy Evaluation.” arXiv preprint, 2023. https://arxiv.org/abs/2311.18207

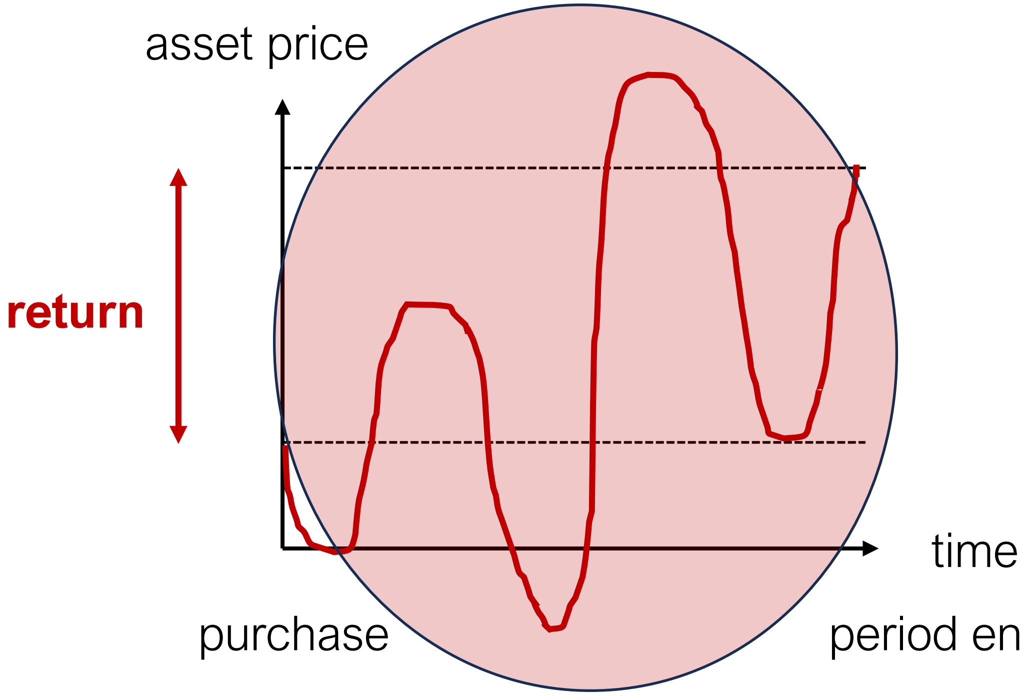
2. “SCOPE-RL: A Python Library for Offline Reinforcement Learning and Off-Policy Evaluation.” arXiv preprint, 2023. https://arxiv.org/abs/2311.18206

Appendix

Connection to the Sharpe ratio [Sharpe,98] in finance

In finance, an investment is preferable if it is low-risk and high-return.





return is not very high, but can be gained steady

return is high, but the investment is high-stakes

Connection to the Sharpe ratio [Sharpe,98] in finance

In finance, an investment is preferable if it is low-risk and high-return.

Sharpe ratio = (increase of asset price) / (deviation of asset price during the period)
= (end price – purchase price) / (std. of asset price)

To improve Sharpe ratio, we often invest on multiple assets and form a portfolio.

Connection to the Sharpe ratio [Sharpe,98] in finance

In finance, an investment is preferable if it is low-risk and high-return.

Sharpe ratio = (increase of asset price) / (deviation of asset price during the period)
= (end price – purchase price) / (std. of asset price)

To improve Sharpe ratio, we often invest on multiple assets and form a portfolio.

applying the idea

We see the top-k policies selected by an OPE estimator as its *policy portfolio*.

Connection to the Sharpe ratio [Sharpe,98] in finance

In finance, an investment is preferable if it is low-risk and high-return.

Sharpe ratio = (increase of asset price) / (deviation of asset price during the period)
= (end price – purchase price) / (std. of asset price)

SharpeRatio = (increase of policy value (pv) by A/B test) / (deviation during A/B test)
= (pv of the policy chosen by A/B test – pv of behavior policy) / (std. of pv of top-k)

We see the top-k policies selected by an OPE estimator as its *policy portfolio*.

Comparison of SharpeRatio and existing metrics

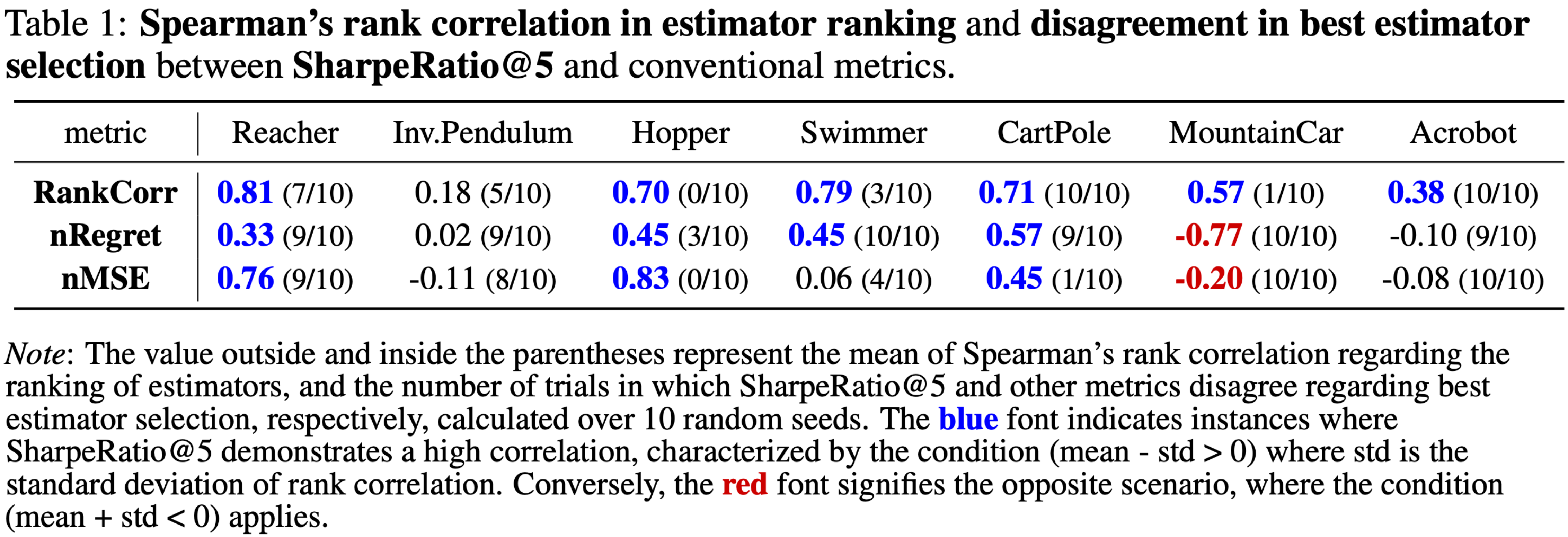


Table 1: **Spearman's rank correlation in estimator ranking** and **disagreement in best estimator selection** between **SharpeRatio@5** and conventional metrics.

metric	Reacher	Inv.Pendulum	Hopper	Swimmer	CartPole	MountainCar	Acrobot
RankCorr	**0.81** (7/10)	0.18 (5/10)	**0.70** (0/10)	**0.79** (3/10)	**0.71** (10/10)	**0.57** (1/10)	**0.38** (10/10)
nRegret	**0.33** (9/10)	0.02 (9/10)	**0.45** (3/10)	**0.45** (10/10)	**0.57** (9/10)	**-0.77** (10/10)	-0.10 (9/10)
nMSE	**0.76** (9/10)	-0.11 (8/10)	**0.83** (0/10)	0.06 (4/10)	**0.45** (1/10)	**-0.20** (10/10)	-0.08 (10/10)

Note: The value outside and inside the parentheses represent the mean of Spearman's rank correlation regarding the ranking of estimators, and the number of trials in which SharpeRatio@5 and other metrics disagree regarding best estimator selection, respectively, calculated over 10 random seeds. The **blue** font indicates instances where SharpeRatio@5 demonstrates a high correlation, characterized by the condition (mean - std > 0) where std is the standard deviation of rank correlation. Conversely, the **red** font signifies the opposite scenario, where the condition (mean + std < 0) applies.

SharpeRatio does not always align with the existing metrics.

(because SharpeRatio is the only metric taking the risk into account)

Definitions of the (normalized) baseline metrics

For MSE and Regret, we report the following normalized values.

$$\mathrm{nMSE}(\hat{J}) := \frac{\sum_{\pi \in \Pi} (\hat{J}(\pi; \mathcal{D}) - J(\pi))^2}{|\Pi| \cdot \max\{(\max_{\pi \in \Pi} J(\pi))^2, (\max_{\pi \in \Pi} J(\pi) - \min_{\pi \in \Pi} J(\pi))^2\}}$$

$$\mathrm{nRegret@}k(\hat{J}) := \frac{\max_{\pi \in \Pi} J(\pi) - \max_{\pi \in \Pi_k(\hat{J})} J(\pi)}{\max\{\max_{\pi \in \Pi} J(\pi), \max_{\pi \in \Pi} J(\pi) - \min_{\pi \in \Pi} J(\pi)\}}$$

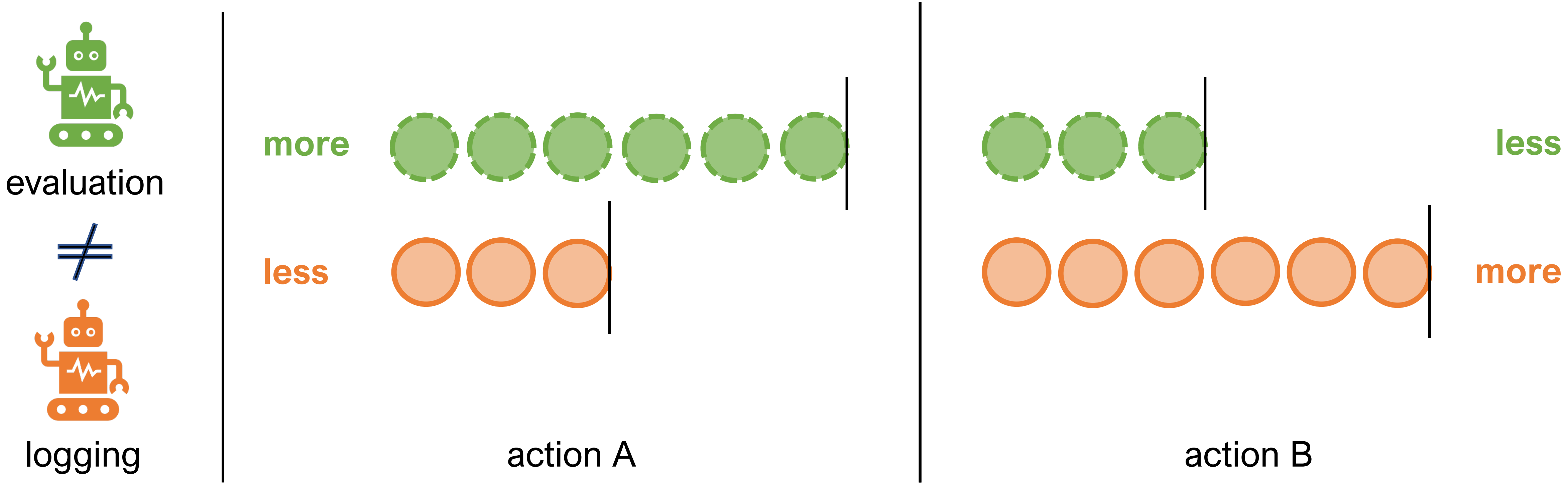
Experimental setting

- We use MountainCar from Gym-ClassicControl [Brockman+,16].
- Behavior policy is a softmax policy based on Q-function learned by DDQN [Hasselt+,16].
- Candidate policies are ε-greedy policies with various values of ε and base models trained by CQL [Kumar+,20] and BCQ [Fujimoto+,19].
- For OPE, we use FQE [Le+,19] to train $\hat{Q}$ and BestDICE [Yang+,20] to train $\hat{\rho}$.
- We also use self-normalized estimators [Kallus&Uehara,19] to alleviate the variance issue.
- We use the implementation of DDQN, CQL, BCQ, and FQE provided in d3rlpy [Seno&Imai,22].

See our paper for the details.

High-level understanding of importance sampling

The target policy chooses action A more, but the dataset contains action B more.

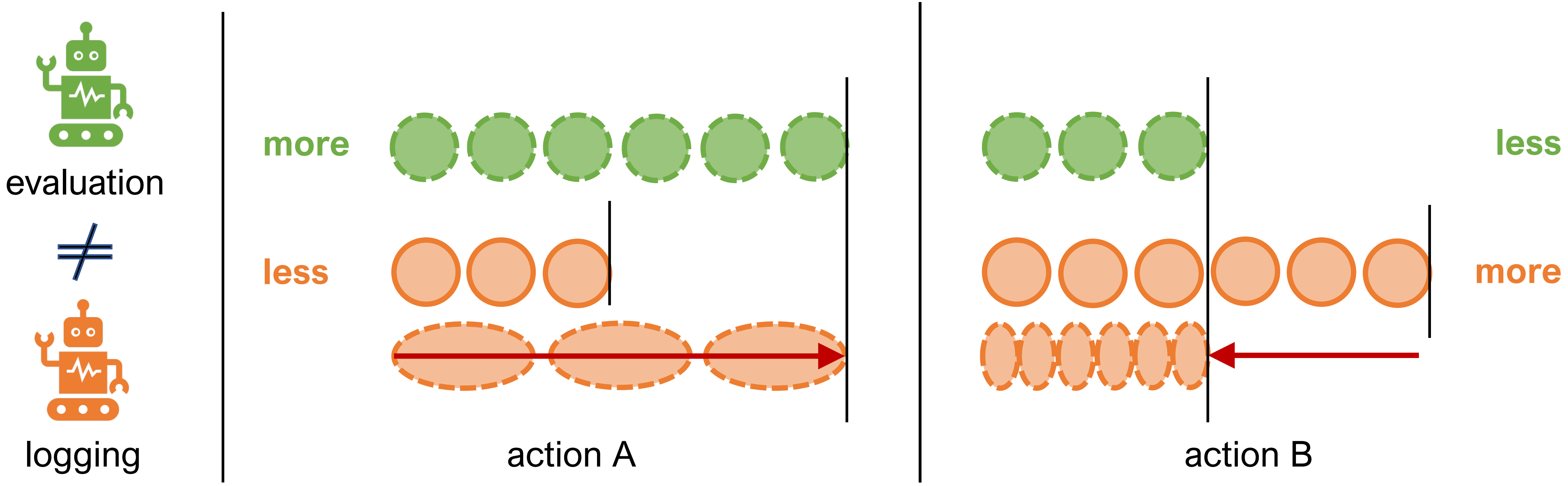


High-level understanding of importance sampling

The target policy chooses action A more, but the dataset contains action B more.

$$\prod_{t'=0}^{t} \frac{\pi(a_{t'} | s_{t'})}{\pi_b(a_{t'} | s_{t'})}$$

importance weight
virtually increases action A

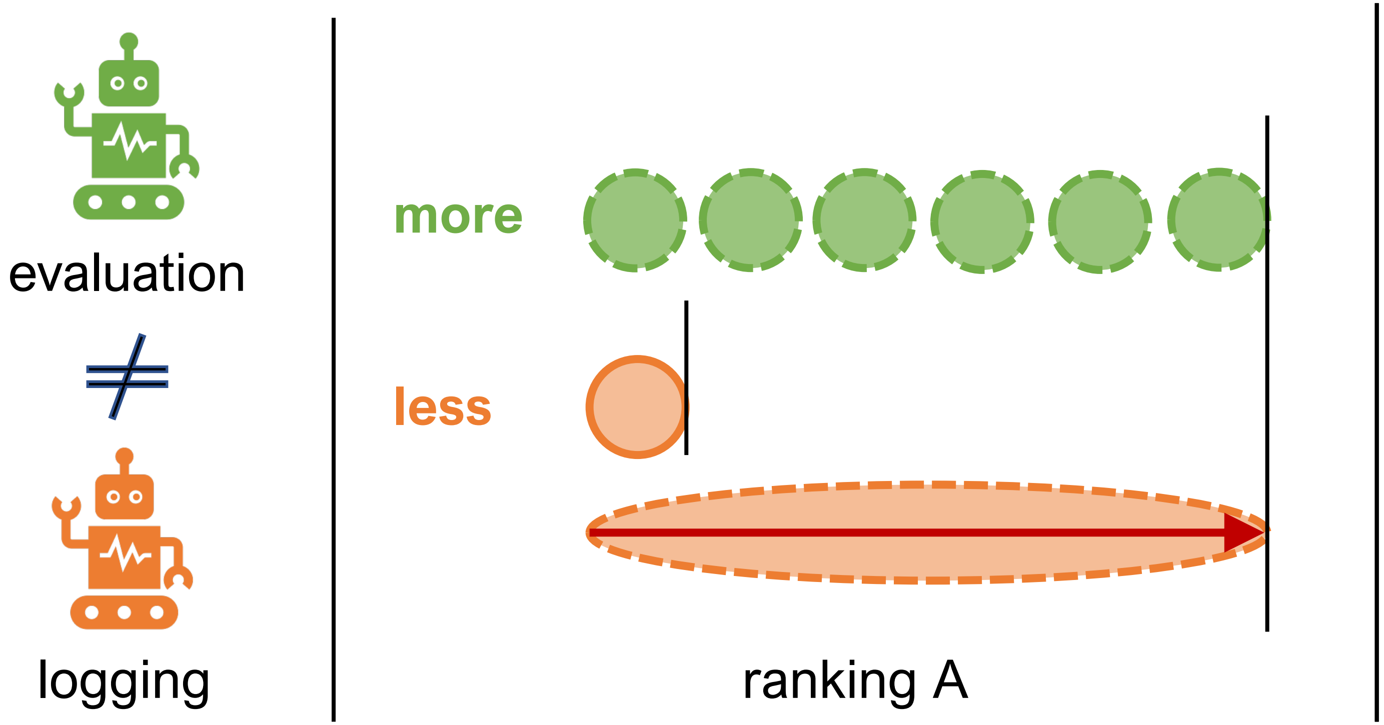


High-level understanding of importance sampling

The target policy chooses action A more, but the dataset contains action B more.

$$\prod_{t'=0}^{t} \frac{\pi(a_{t'} | s_{t'})}{\pi_b(a_{t'} | s_{t'})}$$

but can have a high variance when importance weight is large



Doubly Robust (DR) [Jiang&Li,16] [Thomas&Brunskill,16]

DR is a hydrid of DM and IPS, which apply importance sampling only on the residual.

$$\hat{J}_{\mathrm{DR}}(\pi;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{T-1}\gamma^t\left(w_{0:t}^{(i)}(r_t^{(i)} - \hat{Q}(s_t^{(i)},a_t^{(i)})) + w_{0:t-1}^{(i)}\sum_{a\in\mathcal{A}}\pi(a|s_t^{(i)})\hat{Q}(s_t^{(i)},a)\right)$$

$$w_{0:t} := \prod_{t'=0}^{t}(\pi(a_{t'}\mid s_{t'})/\pi_b(a_{t'}\mid s_{t'}))$$

(recursive form)

$$\underbrace{\hat{J}_{\mathrm{DR}}^{(i)}(T+1-t)}_{\textbf{value after timestep } t} := \gamma \underbrace{w_t^{(i)}\left(r_t^{(i)} + \hat{J}_{\mathrm{DR}}^{(i)}(T-t) - \hat{Q}(s_t,a_t)\right)}_{\textbf{importance weight is multiplied on the residual}} + \sum_{a\in\mathcal{A}}\pi(a^{(i)}|s_t)\hat{Q}(s_t^{(i)},a)$$

Doubly Robust (DR) [Jiang&Li,16] [Thomas&Brunskill,16]

DR is a hydrid of DM and IPS, which apply importance sampling only on the residual.

$$\hat{J}_{\mathrm{DR}}(\pi;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{T-1}\gamma^t\left(w_{0:t}^{(i)}(r_t^{(i)} - \hat{Q}(s_t^{(i)}, a_t^{(i)})) + w_{0:t-1}^{(i)}\sum_{a\in\mathcal{A}}\pi(a|s_t^{(i)})\hat{Q}(s_t^{(i)}, a)\right)$$

$$w_{0:t} := \prod_{t'=0}^{t}(\pi(a_{t'} \mid s_{t'})/\pi_b(a_{t'} \mid s_{t'}))$$

Pros: unbiased and often reduce variance compared to PDIS.

Cons: can still suffer from high variance when t is large.

State-action Marginal DR (SAM-DR) [Uehara+,20]

SAM-DR is a DR variant that leverages the (state-action) marginal distribution.

$$\hat{J}_{\mathrm{SAM-DR}}(\pi; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \pi(a | s_0^{(i)}) \hat{Q}(s_0^{(i)}, a) + \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \hat{\rho}(s_t^{(i)}, a_t^{(i)}) \left(r_t^{(i)} + \gamma \sum_{a \in \mathcal{A}} \pi(a | s_t^{(i)}) \hat{Q}(s_{t+1}^{(i)}, a) - \hat{Q}(s_t^{(i)}, a_t^{(i)}) \right)$$

marginal importance weight is multiplied on the residual

Pros: unbiased when $\hat{\rho}$ or $\hat{Q}$ is accurate and reduces variance compared to DR.

Cons: accurate estimation of $\hat{\rho}$ is often challenging, resulting in some bias.

Self-normalized estimators [Kallus&Uehara,19]

Self-normalized estimators alleviate variance by modifying the importance weight.

$$\hat{J}_{\mathrm{SNPDIS}}(\pi; \mathcal{D}) := \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \underline{\frac{w_{0:t}^{(i)}}{\sum_{i'=1}^{n} w_{0:t}^{(i')}}} r_t^{(i)}$$

$$\hat{J}_{\mathrm{SNDR}}(\pi; \mathcal{D}) := \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \left(\underline{\frac{w_{0:t}^{(i)}}{\sum_{i'=1}^{n} w_{0:t}^{(i')}}} (r_t^{(i)} - \hat{Q}(s_t^{(i)}, a_t^{(i)})) + \underline{\frac{w_{0:t-1}^{(i)}}{\sum_{i'=1}^{n} w_{0:t-1}^{(i')}}} \sum_{a \in \mathcal{A}} \pi(a | s_t^{(i)}) \hat{Q}(s_t^{(i)}, a) \right)$$

Self-normalized estimators are no longer unbiased, but remains consistent.

Self-normalized estimators [Kallus&Uehara,19]

Self-normalized estimators alleviate variance by modifying the importance weight.

$$\hat{J}_{\mathrm{SAM-SNIS}}(\pi; \mathcal{D}) := \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \underline{\frac{\hat{\rho}(s_t^{(i)}, a_t^{(i)})}{\sum_{i'=1}^{n} \hat{\rho}(s_t^{(i')}, a_t^{(i')})}} r_t^{(i)}$$

$$\hat{J}_{\mathrm{SAM-DR}}(\pi; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \pi(a|s_0^{(i)}) \hat{Q}(s_0^{(i)}, a)$$

$$+ \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t \underline{\frac{\hat{\rho}(s_t^{(i)}, a_t^{(i)})}{\sum_{i'=1}^{n} \hat{\rho}(s_t^{(i')}, a_t^{(i')})}} \left(r_t^{(i)} + \gamma \sum_{a \in \mathcal{A}} \pi(a|s_t^{(i)}) \hat{Q}(s_{t+1}^{(i)}, a) - \hat{Q}(s_t^{(i)}, a_t^{(i)}) \right)$$

References

References (1/4)

[Le+,19] Hoang M. Le, Cameron Voloshin, Yisong Yue. “Batch Policy Learning under Constraints.” ICML, 2019. https://arxiv.org/abs/1903.08738

[Precup+,00] Doina Precup, Richard S. Sutton, Satinder Singh. “Eligibility Traces for Off-Policy Policy Evaluation.” ICML, 2000. https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1079&context=cs_faculty_pubs

[Jiang&Li,16] Nan Jiang, Lihong Li. “Doubly Robust Off-policy Value Evaluation for Reinforcement Learning.” ICML, 2016. https://arxiv.org/abs/1511.03722

[Thomas&Brunskill,16] Philip S. Thomas, Emma Brunskill. “Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning.” ICML, 2016. https://arxiv.org/abs/1604.00923

References (2/4)

[Uehara+,20] Masatoshi Uehara, Jiawei Huang, Nan Jiang. “Minimax Weight and Q-Function Learning for Off-Policy Evaluation.” ICML, 2020. https://arxiv.org/abs/1910.12809

[Kallus&Uehara,19] Nathan Kallus, Masatoshi Uehara. “Intrinsically Efficient, Stable, and Bounded Off-Policy Evaluation for Reinforcement Learning.” NeurIPS, 2019. https://arxiv.org/abs/1906.03735

[Brockman+,16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. “OpenAI Gym.” 2016. https://arxiv.org/abs/1606.01540

[Voloshin+,21] Cameron Voloshin, Hoang M. Le, Nan Jiang, Yisong Yue. “Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning.” NeurIPS datasets&benchmarks, 2021. https://arxiv.org/abs/1911.06854

References (3/4)

[Fu+,21] Justin Fu, Mohammad Norouzi, Ofir Nachum, George Tucker, Ziyu Wang, Alexander Novikov, Mengjiao Yang, Michael R. Zhang, Yutian Chen, Aviral Kumar, Cosmin Paduraru, Sergey Levine, Tom Le Paine. “Benchmarks for Deep Off-Policy Evaluation.” ICLR, 2021. https://arxiv.org/abs/2103.16596

[Doroudi+,18] Shayan Doroudi, Philip S. Thomas, Emma Brunskill. “Importance Sampling for Fair Policy Selection.” IJCAI, 2018. https://people.cs.umass.edu/~pthomas/papers/Daroudi2017.pdf

[Kiyohara+,23] Haruka Kiyohara, Ren Kishimoto, Kosuke Kawakami, Ken Kobayashi, Kazuhide Nakata, Yuta Saito. “SCOPE-RL: A Python Library for Offline Reinforcement Learning, Off-Policy Evaluation, and Policy Selection.” 2023.

[Hasselt+,16] Hado van Hasselt, Arthur Guez, and David Silver. “Deep Reinforcement Learning with Double Q-learning.” AAAI, 2016. https://arxiv.org/abs/1509.06461

References (4/4)

[Kumar+,20] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. "Conservative Q-Learning for Offline Reinforcement Learning." NeurIPS, 2020. https://arxiv.org/abs/2006.04779

[Fujimoto+,19] Scott Fujimoto, David Meger, Doina Precup. "Off-Policy Deep Reinforcement Learning without Exploration." ICML, 2019. https://arxiv.org/abs/1812.02900

[Yang+,20] Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, Dale Schuurmans. "Off-Policy Evaluation via the Regularized Lagrangian." NeurIPS, 2020. https://arxiv.org/abs/2007.03438

[Seno&Imai,22] Takuma Seno and Michita Imai. "d3rlpy: An Offline Deep Reinforcement Learning Library." JMLR, 2022. https://arxiv.org/abs/2111.03788

[Sharpe,98] William Sharpe. "The Sharpe Ratio." Streetwise – the Best of the Journal of Portfolio Management, 1998.