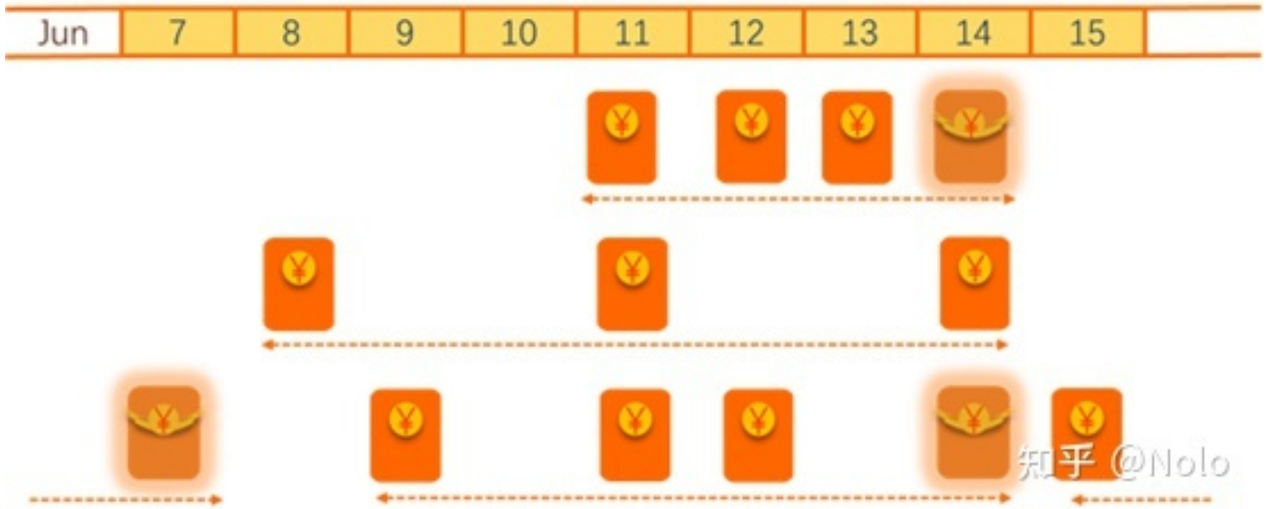
BCORLE(λ): An Offline Reinforcement Learning and Evaluation Framework for Coupons Allocation in E-commerce Market

Li Taishan, Yang Hongdi, Zhao Yibo, Zhao Yida

Introduction

Background

- E-commerce platforms provide users with coupons (red envelope!) to attract users to take action of clicks or conversions.
- Target problem: decide an appropriate value for each coupon to
  - Maximize the users' retention.
  - While limiting the cost not to exceed a fixed budget.



Background

- **Current Strategies**: mainly based on the uplift model. There are two currently adopted in Taobao Deals:
  - Logistics regression + Linear programming (LR+LP)
  - Gradient boost decision tree + Linear programming (GBDT+LP)
- Drawback: oversight, only targeting in maximizing users' retention in a single day.
- Solution: **RL-based policy**, which is inherently good at the sequential decision problems.
- Online or offline: offline RL methods are used to avoid potential financial risks in the learning process.

Problem Formulation

Coupon allocation as CMDP

- CMDP: constrained Markov decision process
  - S: state space
  - A: action space
  - P: transition probability
  - R: reward function
  - C: cost function
  - γ and γ ′: discount factors for reward and cost respectively
  - B: total budget of coupons allocation

$$max_{\pi \in \Pi} J(\pi) = \mathbb{E}_{\tau \sim \pi, \mu} \left[\sum_{t=0}^{T} \gamma^t r_t \right]$$

$$s.t. \quad C(\pi) = \mathbb{E}_{\tau \sim \pi, \mu} \left[\sum_{t=0}^{T} \gamma'^t c_t \right] \leq b,$$

Lagrangian Dual Problem

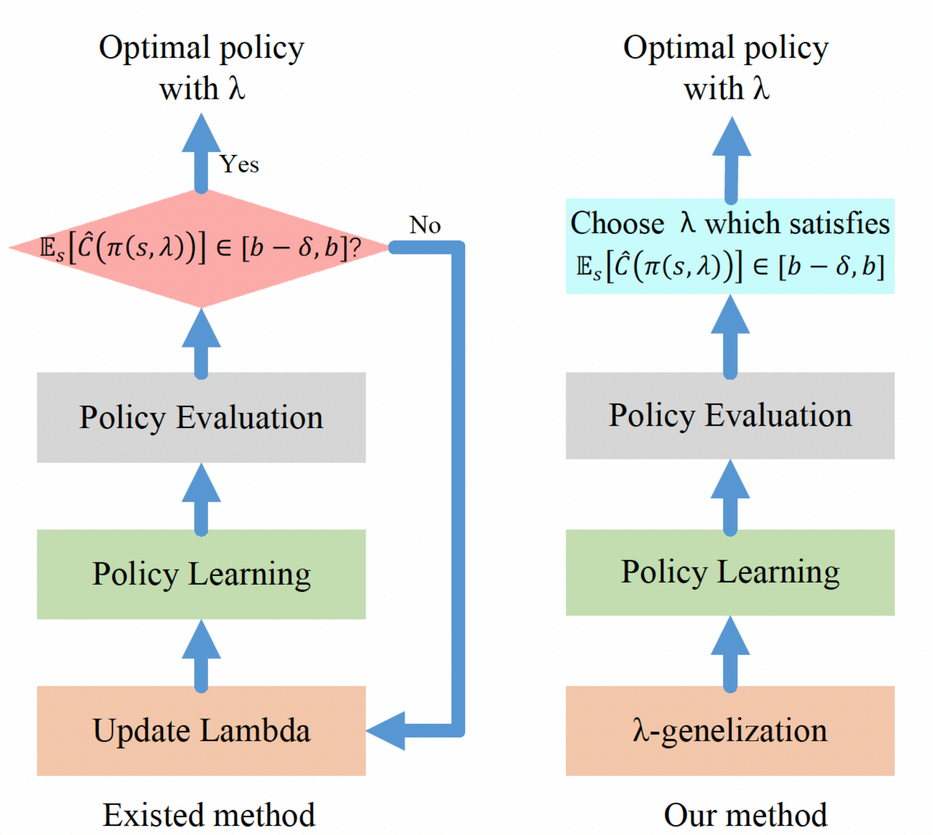
- They resort to the Lagrangian Multiplier method to solve the problem:

$$\max_{\pi \in \Pi} L(\pi, \lambda) = \quad \mathbb{E}_{\tau \sim \pi, \mu} \left[\sum_{t=0}^{T} \gamma^t r_t \right] - \lambda \left(\mathbb{E}_{\tau \sim \pi, \mu} \left[\sum_{t=0}^{T} \gamma'^t c_t \right] - b \right)$$

$$s.t. \quad \lambda \geq 0$$

λ-Generalization

- Learn policy for different λ values simultaneously.
- Less computation overhear compared with the existed methods.



Offline Policy Learning Method R-BCQ

- Combine two popular methods, BCQ and REM.
- Address mismatch problem between learned policy and the training dataset.
- Avoid the Q-value estimation bias problem.
- Policy is chosen as:

$$\pi(s,\lambda) = \underset{a|G(a|s,\lambda;\omega)/\max_{a'} G(a'|s,\lambda;\omega)>\beta}{\arg\max} \sum_i \alpha_i Q_i(s,a,\lambda).$$

- Loss of policy learning network is:

$$L(\theta) = \mathbb{E}_{s,a,r^\lambda,s',\lambda\sim D'} \left[l_k \left(r^\lambda + \gamma \max_{a'\in A} \sum_i \alpha_i Q'_i(s',a',\lambda) - \sum_i \alpha_i Q_i(s,a,\lambda) \right) \right]$$

Policy Evaluation: REME

- A model-free evaluation method.
- Use the evaluate policy π rather than the max over all actions to compute the target value:

$$L(\theta) = \mathbb{E}_{s,a,r,s',\lambda \sim D'}$$
$$\left[l_k \left(r + \gamma \sum_i \alpha_i \hat{Q}_i(s', \pi(s', \lambda), \lambda) - \sum_i \alpha_i \hat{Q}_i(s, a, \lambda) \right) \right]$$

Experiments

Experiment setting

- The paper's experiment is both conducted on a simulation platform and Taobao deals and we decide to reproduce the results in the simulation environment.
- Simulation environment setting:
  - 10000 users, 30 days time span, and the coupons ranging from 0.1 Yuan to 2.1 Yuan with the interval size of 0.1 Yuan.
  - The logging probability of each user: uniformly distributed with $P_{c_i} \geq P_{c_j}$ where c_i means the value of coupon she received the day before.
- Evaluation criterion: expect larger AvgLogins and ROI and smaller AvgCost.

Environment Setup

- Python environment: mainly based on TensorFlow and other 18 libraries.
- Unfortunately, the provided requirement is totally wrong and we found that the codes live in an inherently conflicting environment where at least several dependency conflicts exist.
- We ignored some dependency problems which won’t affect our code.
- We deliberately test different versions of potentially conflicting libraries and their installation orders.
- We made a new readme.md and three new requirementsX.txt among which the number order must be obeyed.

Code Modification

- The original experiment used the codes from offlinerl and some other existing works, but neither the provided offlinerl in the author’s code nor the new version offlinerl meets the requirement. We merge the API s from different versions to get our own offlinerl.
- The code uses APIs only accessible in PAI platform of Alibaba, which we need to obviate.
- Many bugs in the code, we solved no less than 10 of them, but failed to solve all of them.
- We mainly modified the code in offlinerl, utils, train.py and evaluation.py.

Result

- We generated the simulated user data by generator.py and trained by different methods.
- We failed to evaluate by IS, DM, and DR as there are too many bugs and misleading comments mentioned above.
- We got similar results as the original paper and we found not only did R-BCQ have better performance but also cost less time for training.
-

Method	
R-BCQ	0.359h
REM	0.492h
BCQ	1.933h

Comments

Innovations

- We are impressed by the problem formulation: It transforms the coupon allocation problem with a limited budget into a **Lagrangian problem**. As there are no ways to optimize the objective directly, it resorts to **reinforcement learning**.

- It proposed several methods to improve the offline RL of the task:

  - λ-Generalization: learn policies for different λ values, which avoids re-learning policies when we change λ or the budget.

  - Offline policy learning method R-BCQ: takes the advantage of both BCQ and REM.

  - Policy evaluation method REME: model-free.

Code Implementation

- We are also impressed by the messy and ambiguous code.
- We learn from the code there are several bad habits to avoid:
  - The duplicated lengthy code segment.
  - pinyin style naming of variables.
  - using the same name for different variables.
  - misleading comments.
- We have summarized the bugs and dependency conflicts and sent an email to the author.
- Hopefully, we can get a new version of the code and reproduce the whole experiment successfully.

Further Improvement

- Compared to the two existing methods (LR+LP and GBDP+LP), reinforcement learning is less explainable, making it hard to detect and interpret the changes in the policies when suffering attacks.
- In the simulation platform, the author assumes no variation among different users. In real scenarios, users vary in gender, age, and location which will be related to their retention probabilities.
- Since RL methods generally don’t have good properties of generalization and some small changes in the environment may lead to great loss. The setting of the simulation platform should be more complicated.
- There might be overfitting problems or other factors leading to the error rising when learning steps are large.