ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(СПбГУ)

Образовательная программа бакалавриата «Прикладная математика, фундаментальная информатика и программирование», блок элективных дисциплин «Современное программирование»

Отчёт по

«Научно-исследовательской работе Off-policy off-line ranking evaluation»

Выполнил студент 3 курса бакалавриата
(группа 18.Б10-мкн)
Эмдин Григорий Дмитриевич

Научный руководитель:
Куликов Александр Сергеевич,
д.ф.-м.н, профессор факультета математики и компьютерных наук

Санкт-Петербург
2021

Содержание

Введение

В данном проекте затрагиваются методы оффлайн оценивания. На практике часто приходится этим заниматься, когда нет ресурсов для онлайн оценивания, например, заказчик не может себе позволить провести AB - тесты, так как есть риск потерять большое количество клиентов. Тогда на помощь приходят оффлайн метрики. Все, что им нужно - эта история старых онлайн оцениваний и некоторые дополнительные условия на них.

Давайте более строго сформулируем задачу. Предположим, что у нас есть датасет $(x_i, a_i, r_i)_{i=1}^n$ (контекст, действие и награда - соответственно), где и $x_i \sim D,\ r_i \sim D(\cdot|x_i, a_i)$ a_i выбираются в соответствии с некоторой фиксированной логирующей стратегией μ. Также у нас есть новая стратегия π, которую мы хотим оценить. То есть найти $\frac{1}{T}\sum_{i=1}^{T} E_{x\sim D}(E[r_i|x_i, \pi(x_i)])$ И наша задача заключается в том, чтобы построить какие-либо оценки на эту величину и понять, какие условия необходимо наложить на логирующую стратегию.

Данные метрики активно применяются в задаче отображения контекстной рекламы для пользователя. Контекстом в данном случае является страничка пользователя, действие – отображаемая реклама и награда – взаимодействие пользователя с этой рекламой (чаще всего для простоты рассматривается бинарная награда клик/неклик пользователя по рекламе). Другими словами, мы решаем задачу многорукого бандита.

Основные результаты практики

В проекте я изучил существующие метрики для решения поставленной задачи и проверил ряд тестирований для проверки теоретических результатов.

Для начала я рассмотрел метрику Inverse Propensity Scoring (IPS). Ее идея заключается в следующем. Предположим, мы хотим найти $m = E(f(X)) = \int_D f(x)p(x)dx$.

$$m = \int_D f(x)p(x)dx = \int_D \frac{f(x)p(x)}{q(x)}q(x)dx = E_q(\frac{f(X)p(X)}{q(X)})$$

То есть мы смогли пересчитать матожидание одной функции, через матожидание другой. А это и есть то, что мы хотим сделать в случае с различными стратегиями. В этой метрике нам важна несмещенность матожидания. Доказательство этого факта приведено в статье [1], раздел 7.2.1. Также там есть ряд оценок на дисперсию IPS.

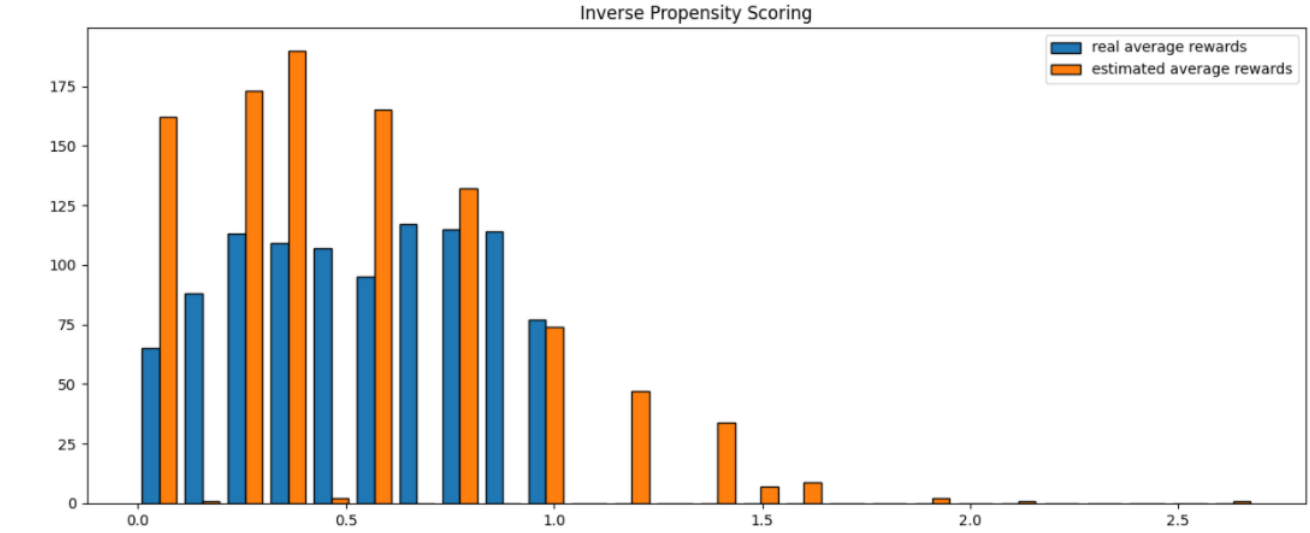
В этой же статье в разделе 7.2.2. рассказывается про Doubly Robust Estimators. Идея заключается в том, чтобы выучить какую-нибудь модель предсказывать какую награду получит конкретное действие в конкретном контексте. Тогда можно будет прогнать стратегию, которую мы хотим оценить, по набору контекстов, посмотреть, что она скажет, а потом по полученным парам (x, a) предсказать награду нашей моделью. Проблема заключается в том, что такая предсказывалка будет иметь смещение, так что приходится навешивать дополнительные веса. Метод Doubly Robust говорит какие веса лучше добавлять.

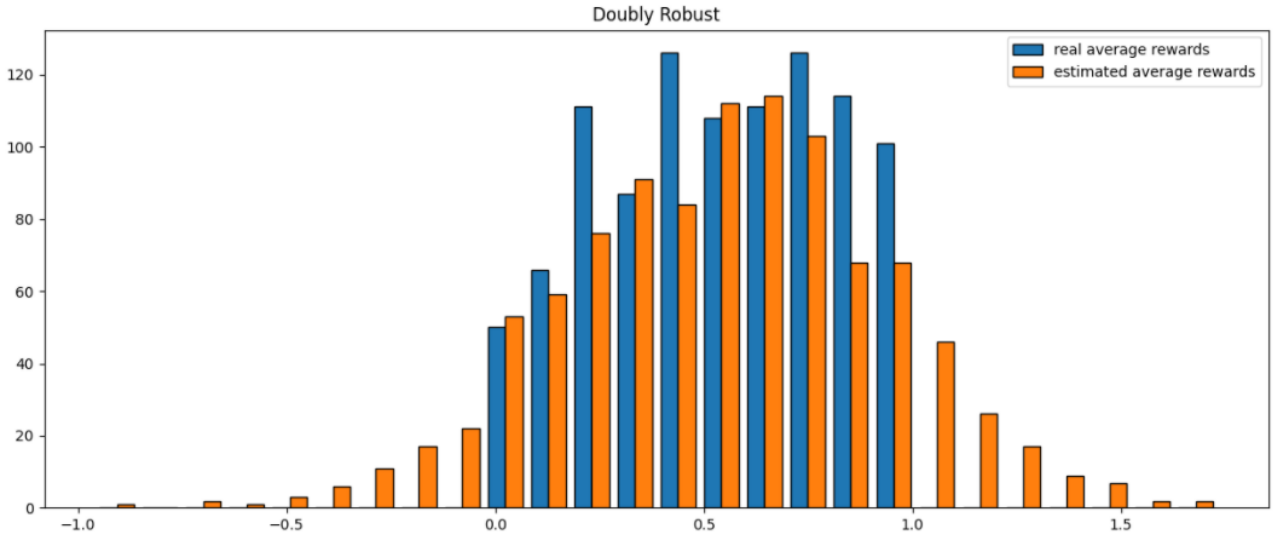
В предыдущих методах мы пользовались тем, что логирующая стратегия у нас рандомизированная и мы знаем вероятности, с которыми она выбирает то или иное действие. Но на практике мы не всегда можем узнать эти вероятности, поэтому их тоже приходится оценивать. Для этого используется метод exploration scavenging. Этот метод подробно описывается в [2]. Также в этой статье накладываются ограничения на логирующую стратегию, а именно там показаны оценки на то, как часто логирующая стратегия должна выбирать каждое действие.

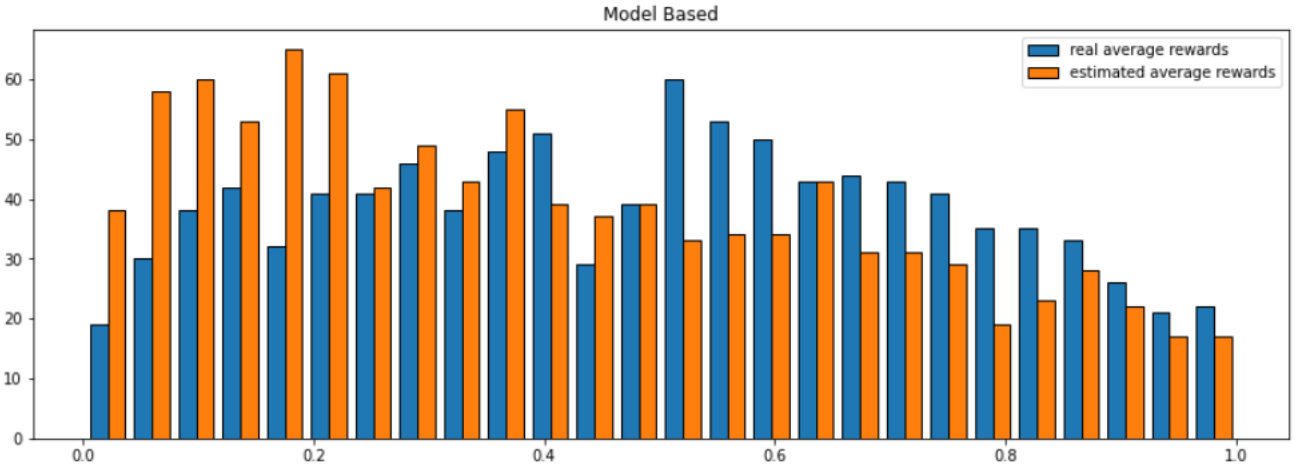
Также в случае, когда нам неизвестны вероятности у логирующей стратегии применяется replay метод для контекстуальных бандитов (многорукие бандиты с контекстом). В статье [3] про него подробно рассказывается. В отличие от предыдущих методов, в которых надо было симулировать онлайн процесс, этот метод использует только логирующие данные и легко адаптируется к различным приложениям. В статье [4] рассказано про модификацию этого метода, это один из лучших результатов, существующих на данный момент.

Для проведения экспериментов я взял датасет из открытого соревнования KDD Cup 2012. В этих данных представлены пользователи и рекламы, которые им были показаны в каких-то конкретных контекстах. Подробнее про то, как я симулирую данные можно прочитать в разделе 5.2.2 статьи [5]. В качестве логирующей стратегии я взял рандомизированную (она выбирает каждое действие с вероятностью $\frac{1}{N}$). Оценивал я алгоритм, решающий задачу многорукого контекстного бандита, взятый из библиотеки Vowpal Wabbit.

Ниже приведены графики распределения реальных наград и предсказанных разными оценщиками.







Код, с помощью которого полученные эти результаты есть на моем гитхабе.

Заключение

В этом проекте я изучил существующие методы оффлайн оценивания алгоритмов. Проверил, как они работают на практики на синтетических данных и на реальных. На графиках, приведенных выше, видно, что некоторые оценки получаются со смещением и с большой дисперсией. Избавиться от того и другого не получается, поэтому приходится чем-то жертвовать. Мы рассмотрели разные ситуации, в которых те или иные оценщики оказываются лучше.

Список использованных литературных источников и информационных материалов

[1] http://alekhagarwal.net/bandits_and_rl/off_policy.pdf
[2] https://dl.acm.org/doi/abs/10.1145/1390156.1390223
[3] https://arxiv.org/abs/1003.5956
[4] https://arxiv.org/ftp/arxiv/papers/1210/1210.4862.pdf
[5] https://dl.acm.org/doi/pdf/10.1145/2939672.2939878
[6] https://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf
[7] https://arxiv.org/pdf/2002.11642.pdf
[8] https://arxiv.org/pdf/1003.0120.pdf
[9] https://arxiv.org/pdf/1103.4601.pdf
[10] http://proceedings.mlr.press/v70/wang17a.html

Перечень использованного оборудования, в том числе оборудования Научного парка СПбГУ

[1] Ubuntu 20.04.2.0 LTS (Focal Fossa)
[2] PyCharm
[3] Python 3.8
[4] Vowpal Wabbit
[5] NumPy
[6] matplotlib.pyplot
[7] GitHub