Conservative Q-Learning VS Implicit Q-learning

Алгоритмы

Для выполнения тестового задания я выбрала статьи Offline Reinforcement Learning with Implicit Q-Learning и Conservative Q-Learning for Offline Reinforcement Learning. Свой выбор я обосновываю тем, что хотелось выбрать такие две статьи, которые понятным образом сравниваются между собой. В обеих статьях нет перехода к обучению в онлайне, поэтому они сравнимы между собой. Если быть честной, то в рамках моей текущей работы я уже сталкивалась с Conservative Q-Learning, поэтому имела некоторое представление об этом алгоритме заранее и поэтому выбрала именно его для воспроизведения статьи. Однако с алгоритмом Implicit Q-Learning я столкнулась впервые. Преимуществом обоих алгоритмов является простота реализации.

Алгоритм CQL интересен тем, что в отличие от классического Q-Learning в онлайне, в CQL оптимизация q-функции идет по всем данным, так как матожидается квадрат temporal difference error по всем состояниям и действиям. Это делает обучение более стабильным, но как раз требует иметь данные сразу, то есть применимо только в оффлайн постановке задачи. Также при оптимизации q-функции используется регуляризующее слагаемое для того, чтобы ограничить значения получаемой q-функции. Это слагамое-регуляризатор работает так, что это минимум значения q-функции по наихудшему апостериорному распределению действий по состояниям, то есть по максимальному значению отклонения матожидания по реальному распределнию данных и апостериорному.

Что касается IQL, то он интересен тем, что это другой подход к модификации Q-Learning, в котором неявно аппроксимируется шаг улучшения политики, рассматривая q-функцию состояния как случайную переменную со случайностью, определяемой действием, а затем берется условное математическое ожидание состояния этой случайной величины для оценки ценности лучших действий в этом состоянии.

Я взяла готовые реализации алгоритмов из библиотеки d3rlpy.

Датасеты

Библиотека, которой я пользовалась позволяет генерировать датасет из классических задач gym. Я взяла 2 вариации задачи Hopper: hopper-medium-v2, на котором лучший результат показал алгоритм IQL и hopper-medium-expert-v2, на котором наоборот результат лучше у CQL, а также взяла другую задачу walker2d-medium-v2, на которой тоже лучше IQL. Значений метрик для этиз задач я взяла из таблицы 1 из статьи Offline Reinforcement Learning with Implicit Q-Learning, потому что именно там есть сравнение этих двух алгоритмов на различных задачах. Я ожидала получить похожее поведение скоров на этих задачах.

Метрики

В качестве метрик, на которых я буду производить дальнейшее сравнение я взяла следующие.

Average TD error, так как эта метрика показывает, насколько q-функции подходят для обучающих наборов. А еще ее значение используется во время оптимизации q-функции в обоих алгоритмах, поэтому интересно на нее посмотреть.

Enviroment evaluation score, так как эта метрика вычисляет значение самой политики, поэтому на нее тоже интересно посмотреть.

Average value estimation, так как эта метрика оценивает ожидаемое значение q-функии по действию с максильным ее значением. В документации говорится, что если оценка среднего значения слишком велика, функции Q переоценивают значения действия, что может привести к сбою обучения.

И наконец я решила посмотерть на Soft Off-Policy Classification, так как эта метрика оценивает разрывы в оценке ценности действия между успешными действиями и всеми действиями.

Результаты

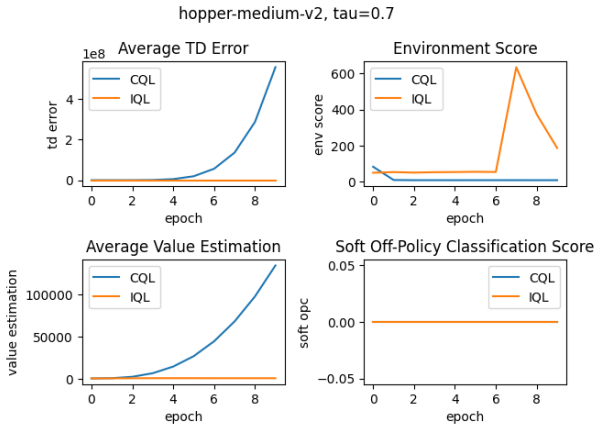
Я запускала по 1000 итераций каждый алгоритм на стандартных параметрах (10 эпох по 100 итераций).

Эффективность

Сразу могу отметить, что IQL работает гораздо эффективнее CQL. 2 минуты чистого времени исполнения у IQL на одной и той же конфигурации и параметрах против 7 минут у CQL.

Гиперпараметры

Первое, что хочется отметить, не очень успешный подбор гиперпараметров. В статье про IQL было указано, что использовалось значение $'tau = 0.7$, однако при таком значении у меня получилось переобучение, у CQL выросло значение td error до небес, что говорит о явном переобучении, а значения environment score гораздо меньше, чем на параметрах по умолчанию. Дальнейшие движения гиперпараметров не привели ник чему хорошему. Результат на картике ниже.



Метрики

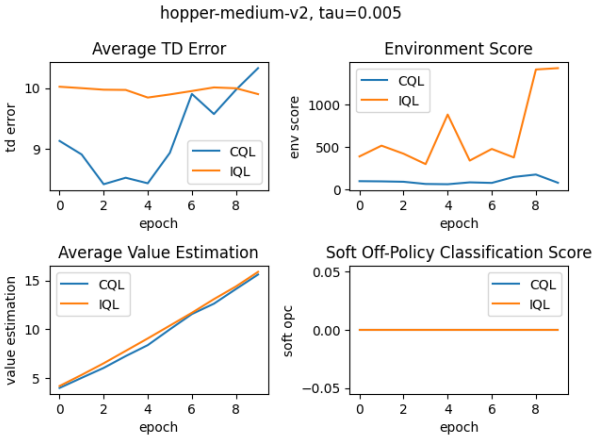
Если сравнивать метрики на стандартных гиперпараметрах, то можно сделать вывод, что так как td error растет со временем у CQL, вероятно можно сделать о том, что алгоритм переобучился, хотя значения наверное не сильно велико. Хотя у IQL видно несильную тенденцию к уменьшению td error, что сходится с предположением о том, как должен работать алгоритм.

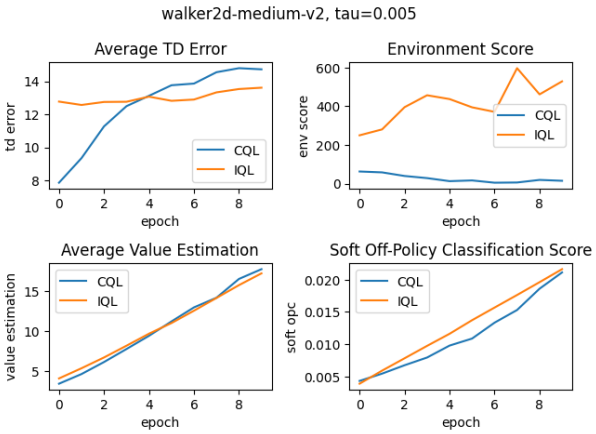
Что касается environment score, то у IQL он больше на всех трех задачах, то есть алгоритм на валидации выбивает лучший скор. Видно, что у CQL он примерно одинаковый на всех эпохах ив районе нуля, а вот у IQL хоть и нестабилен, но по значению гораздо больше.

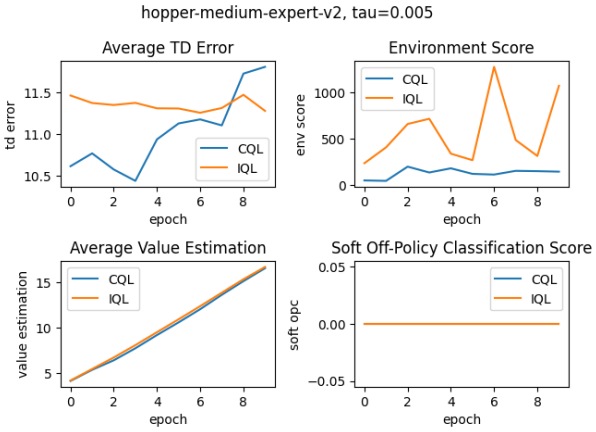
Value estimation растет на всех трех задачах и на обоих алгоритмах, это нормально, при этом мы не получаем запределно больших значений, так что на мой вгляд, все нормально, так как упрощая это среднее значение q-функции и это логично что оно растет с каждой эпохой. Наоборот при плохо подобранном гиперпараметре τ мы видели, что эта метрика может улететь в очень большие значения, это как раз знак, что обучение пошло не так.

Наконец, soft-off-policy метрика отличается от нуля только на задаче hopper-medium-expert-v2, то есть в этом случае значение q-функции на успешных действиях в среднем больше, чем на всех, это хорошо. Только странновато, что на остальных задачах мы это не получили.

В итоге, у меня не получилось получить значения близкие к представленным в статье, однако я провела анализ и сравнила эти два алгоритма и могу сделать вывод, что IQL все таки лучше справляется с поставленной задачей. Графики по которым проводился анализ представлены ниже.


hopper-medium-v2, tau=0.005
Average TD Error
td error
epoch
CQL
IQL
Environment Score
env score
Average Value Estimation
value estimation
Soft Off-Policy Classification Score
soft opc


walker2d-medium-v2, tau=0.005
Average TD Error
td error
epoch
CQL
IQL
Environment Score
env score
Average Value Estimation
value estimation
Soft Off-Policy Classification Score
soft opc


hopper-medium-expert-v2, tau=0.005
Average TD Error
td error
epoch
CQL
IQL
Environment Score
env score
Average Value Estimation
value estimation
Soft Off-Policy Classification Score
soft opc