Отчет о проделанной работе за последнее время

9 марта 2017 г.

1 Что было начато до этого

1.1 Safe reinforcement learning

В последнем отчете я описывал, что прочитал очень занимательную статью о безопасном обучении с подкреплением и поделился планами о том, что хочу сделать алгоритм, который бы обучал агента действовать в среде таким образом, чтобы со стопроцентной вероятностью агент не получал нагрды меньше какого-нибудь порога.

Идея была в том, чтобы можно было сначала обучить агнета действовать оптималь в безопасном месте (например в виртуальной среде), а потом уже в ходе применения в реальных условиях быть уверенным, что не произойдет критически плохой ситуации. Примером(хоть и немного гипертрофированным) может быть обучение марсохода двигаться по некоторой поверхности. Его можно обучить в безопасном ангаре на Земле или в виртуальной среде и в ходе этого обучнения он может переворочиваться без проблем - его можно просто перевернуть обратно или перезапустить виртуальный симулятор. Но когда он обучился и уже полетел на Марс - мы не можем позволить ему перевернуться никак, потому что перевернуть обратно его некому.

Аналогичным примером может быть некоторый манипулятор на заводе (назовем его для простоты роборукой). Там тоже может быть применена подобная идея, мы обучаем эту роборуку в вуртуальной среде и потом хотим быть уверены, что в ходе реальной работы на заводе она не испортит дорогостоющую деталь, над которой она роботает и не убъет рабочего.

Об безопасном обучении с покреплением я задумался именно думая о манипуляторе на заводе, но, несмотря на то, что пример с марсоходом менее применимый, я буду в примерах использовать его, потому что он нагляднее.

2 Что я пытался сделать, но провалился

Очень хотелось сделать алгоритм, который давал бы стопроцентную гарантию того, что не произоейдет критической ситуации.

Я подумал, что мы можем пожертвовать ради этого оптимальностью и получать последовательность действий чуть похуже, но зато абсолютно безопасную.

Для описания алгоритма я буду использовать пример, когда нужно просто пройти из случайной точки некоторого начального множества A в точку

Б и не попасть в яму. В этом случае состоянием у нас будет некоторый вектор, состоящий из:

- координаты в пространстве
- скорость
- ускорение
- угол относительно поверхности
- любые другие механические показатели

Наградой будет -1 за кадый момент времени, большая положительная награда за попадение в точку Б и большая отрицательная награда за попадение в яму.

Действие тут будет просто шаг в какую-то сторону. В этом случае последовательность действий можно называть траекторией, тк каждый шаг приводит к передвижению в пространстве.

Сразу оговоримся, что если мы используем явно модель среды (иными словами, в примере с путешествия из множества A в точку Б, если нам известно к чему приводит каждый наш шаг и количество состояний у нас не очень большое), то решать задачу безопасного обучения с подкреплением не имеет смысла, так как мы можем непосредственно посчитать для каждого состояния что будет, после того, что мы сделаем некоторое действие и можем явно проверить не произойдет ли критической ситуации.

А вот если у нас достаточно большое количество состояний (например, потому что размерность вектора, описывающего состояние большая), то мы вынуждены применять model free методы обучения с подкреплением, и тут мы уже не можем проверить что будет после следующего шага непосредственно.

Сначала я подумал, что нужно сделать алгоритм состоящий приблизительно из следующих шагов:

- Сначала мы методами классического обучениния с подкреплением находим некоторое решение. Т.е. каждому состоянию ставим в соответсвие некторое действие, которое считаем оптимальным. Теперь у нас есть представление что делать в каждом конкретном состоянии, но нет никакого представления о том, насколько такое движение безопасно.
- Выбираем одну или несколько точек из множетсва A и в нашем симуляторе проходим весь путь до точки Б и непосредтсвенно убеждаемся, что трагедии не произошло. Теперь у нас есть одна какая-то безопасная траектория, но для остальных точек все еще нет безопасного решения.
- Делим A как-нибудь на много частей и назначем каждую часть A за новое A'.
- Теперь для всех точек из новых множеств A' определяем целью (точкой Б) точки, которые лежат в безопасной траектории, полученной ренее. Теперь у нас есть много задач аналогичных предыдущей.

- Дробим исходную задачу таким образом до достижения наперед заданного уровня дробления.
- Решаем каждую подзадачу, так как описано в первых двух пунктах.
- Состовив воедино решения всех задач мы получим траекторию, которая будет решать исходную задачу.

Полученное в итоге решение будет безопасным, так как безопасность этой состовной траектории мы проверяли непосредственно при решении каждой из задач.

Этот алгоритм я хотел реализовать и показать на Ломоносове, пока не понял, что он не имеет никакого практического применения. Дело в том, что по сути тут предлагается обойти все пространтво, пусть и с каким-то шагом и если это пространство большое, то это невозможно. А если пространство маленькое, то, как я говорил раншьше, нам не нужен этот алгоритм вовсе.

Я много времени потратил, чтобы продумать как это все сдеалать и был немало разочарован, когда понял, что это никогда не заработает.

2.1 Более оптимистичный вариант

Потерпев такой провал, я все же хотел понять как бы я решал задачу отправки лунохода на Марс и как бы гарантировал его безопасность там. После некоторого времени сложных изысканий в машинном обучении, я понял, что решение в действительности было бы очень простым: были бы выбраны опасные положения, скажем, те, когда аппарат сильно накреняется и их бы просто не допускали при управлении марсоходом.

Это очень постое и безопасное решение, однако оно очень неэффективно, потому что не понятно как выбирать границу допустимого угла наклона аппарата, чтобы с одной стороны все было безопасно, с другой, чтобы это ограничение не мешало марсоходу ходить по сложной и бугристой поверхности Марса.

Я понял, что если в задачу, в той формулировке, в которой я ее решал до этого добавить что-то вроде этого самого угла наклона аппарата как непрерывную меру риска - то получится более реальная для решения задача, причем не менее общая с точки зрения применения!

То есть, теперь у нас агент знает не только состояние и возможные варианты движений, но также и некоторое число(в предыдущем примере этот самый угол, в общем - не обязательно), которое показывает насколько агент близок к рискованой ситуации, может также быть некоторый порог этой переменной, который говорит о том, что произошел крах (например, угол достиг 90 градусов и марсоход перевернулся).

Важно отметить, что, несмотря на то, что у нас есть эта мера риска, и пусть даже у нас есть некоторый порог краха, мы не можем действовать свободно, находясь даже вдали от этого порога. Дело в том, что в общем случае, нам не известна динамика среды и если мы будем неаккуратно действовать вдали от этого порога, то мы сделать что-нибудь необратимое, что приведет нас к тому, что мы неотвратимо будем двигаться в сторону порога и будет уже невозможно выбрать такое управление, чтобы избежать пересечение этого порога.

Однако, у нас есть представление, о том, что действия, который направлены на уменьшение этой рисковой переменной безопасны. И это уже очень много для обчения агента.

Я поискал в литературе по безопасному обучению с подкреплением похожие методы и ничего такого не нашел. Во всяком случае в том, достаточно полном обзоре, которые я описывал в прошлом отчете.

2.2 Выводы

В общем, мне кажется, что в этой формулировке можно описать много прикладных задач, при этом похожих исследований я не видел.

Однако, несмотря на то, что я, кажется, напал на след очень интересной и незанятой темы, мне нужно обсудить с Вами как к ней подстуриться.