Введение в обучение с подкреплением

Тема 9: Непрерывные среды и приближение функциями

Лектор: Кривошеин А.В.

Проблемы табличных методов

Ранее мы рассматривали конечные МППР. Функции ценности состояний и действий могли быть заданы таблицами, а соответствующие методы называют **табличными методами.** При сравнительно небольшом числе состояний\действий эти методы показывают хороший результат.

Табличные методы не подойдут для тех случаев, когда пространство состояний огромно и их обработка выходит за рамки вычислительных возможностей.
Кроме того, в реальных задачах часто пространство состояний и действий **непрерывно**.

Базовая идея решения задач в этих ситуациях опирается на следующее соображение:

похожие действия в похожих состояниях могут приводить к похожему доходу.

Агенту теперь требуется **обобщать посещённые состояния**, выделяя из них наиболее важные признаки. Дальнейшее обучение и формирование стратегии выбора наилучших действий производится на основании наблюдаемых признаков.

Задача обобщения признаков состояний — это задача теории приближений: надо сложные пространства состояний уметь приближать с помощью некоторой простой модели. Существует огромное количество различных подходов, решающих эту задачу.

Работа с непрерывными средами

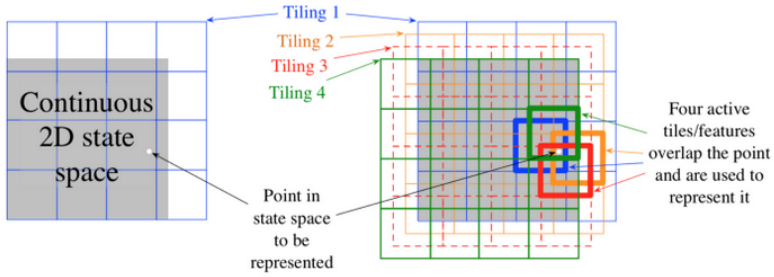
Простейший способ обобщения состояний в непрерывных средах или в средами с большим числом состояний — это дискретизация пространства состояний. Тем самым, решение задачи может быть получено с помощью табличных методов.

Дискретизацию можно осуществлять различными способами. Рассмотрим два.

1. **Равномерная дискретизация.** Пусть состояние среды описывается набором параметров, которые меняются в рамках некоторых интервалов. Для каждого из параметров можно разбить этот интервал на несколько равных по длине подинтервалов. Тогда дискретное состояние — это набор индексов тех подинтервалов, в которые попадают значения параметров состояния.

2. **Плиточное кодирование.** Рассмотрим этот способ на примере. Пусть множество состояний — это некоторое ограниченное подмножество $\mathbb{R}^2$. Накроем это множество сеткой непересекающихся квадратов. Назовём это плиточным покрытием множества состояний. Сформируем несколько таких покрытий, в простейшем случае новые покрытия получаются как сдвиги исходного покрытия с некоторым шагом по каждой из координат.

Пусть есть M покрытий, в каждом покрытии плитки индексируются от 1 до N. Тогда каждое состояние описывается M индексами, которые соответствуют плиткам, в которые это состояние попало в каждом из покрытий.



Такого типа плиточное кодирование по сути объединяет соседние состояния в одно (англ. state aggregation).

Методы на основе приближений

Более подходящий подход для работы с непрерывными средами основан на методах теории приближений.

Базовое предположение: ценности состояний/действий можно представить в виде функции специального вида с некоторым набором параметров (весов):

$v_\pi(s) \approx \hat{v}(s;\theta), \quad q_\pi(s,a) \approx \hat{q}(s,a;\theta), \quad$ где θ вектор с параметрами.

Обычно фиксируется класс приближающих функций и цель обучающего алгоритма в поиске параметров θ, при которых истинные ценности $q_\pi(s,a)$ приближаются достаточно хорошо моделью $\hat{q}(s,a;\theta)$ в соответствии с некоторым критерием.

Поиск конкретных параметров происходит по некоторому алгоритму, обычно итеративному, который зависит от вида приближающей функции. Среди основных классов приближающих функций отметим два:

- **линейное приближение**, то есть функции $\hat{v}(s;\theta)$ или $\hat{q}(s,a;\theta)$ представляют собой линейные функции от признаков состояний, а вектор θ тогда является вектором весов этих признаков;

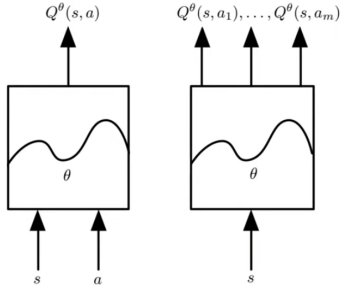
- **нелинейное приближение на основе Искусственных Нейронных Сетей (ИНС)**, то есть значения $\hat{v}(s;\theta)$ или $\hat{q}(s,a;\theta)$ вычисляются ИНС, набор параметров θ — это веса ИНС.

Методы на основе приближений

При приближении функции ценности действий $q_\pi(s, a)$ используется два базовых подхода к формированию значения приближающей функции или модели:

1. на вход функции $\hat{q}(s, a; \theta)$ подаётся пара состояние-действие и вычисляется её значение;

2. если набор действий дискретен, то на вход функции $\hat{q}(s, a; \theta)$ можно подавать только состояние и вычислять значения функции $\hat{q}(s, a; \theta)$ для всего набора действий возможных в этом состоянии, то есть результатом вычисления будет вектор из $|\mathcal{A}|$ значений $\hat{q}(s, a; \theta)$.



Отметим **отличие в обновлении ценностей от табличных методов**.

Число параметров приближающей функции меньше числа пар состояние-действие. При обновлении параметров модели для улучшения оценки ценности одного состояния, изменятся оценки и для многих других состояний. Действительно, оценки ценностей не обновляются теперь непосредственно, как для табличных методов. Обновляются параметры θ, которые в свою очередь влияют на значения ценностей $\hat{v}(s; \theta)$ или $\hat{q}(s, a; \theta)$. В этом смысле происходит обобщение полученного опыта.

Методы на основе приближений

Итак, требуется настраивать набор параметров, чтобы в ходе обучения уменьшать разницу между истинными значениями Q-функции и текущей оценкой, то есть это **задача обучения с учителем**. Но есть ряд важных особенностей.

В классической задаче обучения с учителем есть набор обучающих данных (признаки и правильные ответы) и по ним надо построить их модель.

В задачах обучения с подкреплением данные меняются. При обучении мы не знаем истинных значений ценностей, мы можем получить лишь их оценки, либо по методу МК, либо по методу TD. Именно эти оценки и будут теми целевыми значениями, к которым мы хотим приблизиться при обновлении параметров.

Более того, в TD методах целевое значение формируется с использованием текущих оценок ценностей. Новая цель обновления формируется с использованием текущих оценок, выдаваемые нашей обучаемой моделью.

Иными словами, **обучаемая модель влияет на формирование целей обновления для своего обучения.**
То есть целевые значения в различные моменты времени для одного состояния могут быть различны. Эта особенность известна как **нестационарность целей** (англ. nonstationary targets).

Методы на основе приближений

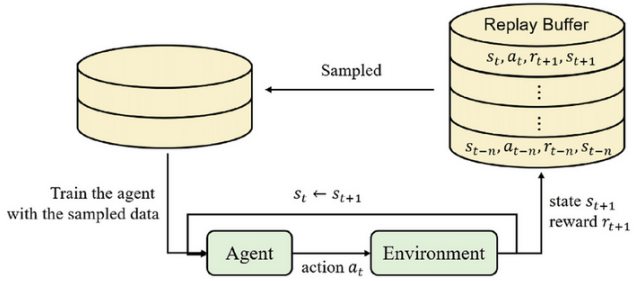
Кроме того, работа алгоритмов обучения с учителем предполагает, что данные, поступающие для обучения, не коррелируют друг с другом.

В обучении с подкреплением это не так, стратегия поведения агента напрямую влияет на генерацию данных. И обучающие примеры, последовательно поступающие агенту, сильно коррелируют друг с другом, текущие вознаграждения и состояния зависят от предыдущих состояний и действий в прошлом по траектории.

Нестационарность целей и **скоррелированность данных** для обучения усложняют вопросы теоретического обоснования сходимости методов, основанных на приближении функций, в задачах обучения с подкреплением.

Буфер памяти

При использовании методов обучения с разделённой стратегией можно проводить обучение на ранее полученном опыте. В этом случае проблема корреляции обучающих данных решается с помощью **буфера памяти**. В реальных задачах опыт может накапливаться медленно, поэтому использование сгенерированного опыта может значительно ускорить обучение.



Буфер хранит последние k шагов взаимодействия вида (s, a, r, s'). При заполнении памяти самые старые записи удаляются и заполняются новыми.

Буфер позволяет производить выборку обучающих примеров случайным образом. Особенно это важно, когда Q-функция параметрически моделируется некоторой нелинейной функцией и обучение будет более устойчивым, если обучающие примеры предъявлять пакетами (англ. batch).

Оценка стратегии

Рассмотрим сначала задачу **оценки стратегии** на основе методов параметрического приближения, то есть оценки истинных ценностей $v_\pi(s)$ с помощью модели $\hat{v}(s,\ \theta)$. Требуется настроить параметры θ так, что разность $(v_\pi(s) - \hat{v}(s,\ \theta))^2$, усреднённая по всем генерируемым по стратегии π траекториям, минимальна:

$J(\theta) = \mathbb{E}_\pi[v_\pi(s) - \hat{v}(s,\ \theta)]^2 \ \rightarrow \ \min$ для каждого состояния s.

Однако, всех данных сразу у нас нет. Поэтому для сформированной траектории параметры сдвигаются так, чтобы минимизировать величину

$\frac{1}{2}\left(v_\pi(S_t) - \hat{v}(S_t,\ \theta)\right)^2$ для текущего состояния S_t.

Шаг обновления параметров по методу градиентного спуска имеет вид:

$\theta_{k+1} = \theta_k + \alpha\left(v_\pi(S_t) - \hat{v}(S_t,\ \theta_k)\right)\ \mathrm{grad}_\theta\, \hat{v}(S_t,\ \theta_k)$.

Но и **точная цель обновления** $v_\pi(S_t)$ нам **не доступна**.

По методам МК $v_\pi(S_t)$ оценивается доходом G_t при старте из состояния s и следовании стратегии π.

По методам TD целью обновления является оценка $R_{t+1} + \gamma\, \hat{v}(S_{t+1},\ \theta)$. Тогда шаги обновления имеют вид

$\theta_{k+1} = \theta_k + \alpha\left(G_t - \hat{v}(S_t,\ \theta_k)\right)\ \mathrm{grad}_\theta\, \hat{v}(S_t,\ \theta_k)$

$\theta_{k+1} = \theta_k + \alpha\left(R_{t+1} + \gamma\, \hat{v}(S_{t+1},\ \theta_k) - \hat{v}(S_t,\ \theta_k)\right)\ \mathrm{grad}_\theta\, \hat{v}(S_t,\ \theta_k)$

Аналогичные формулы можно выписать и для оценки ценности действий $\hat{q}(S_t,\ A_t,\ \theta)$.

Оценка стратегии

Рассмотрим случай линейной параметрической модели приближения. Пусть состояние записывается в виде некоторого вектора признаков $f(s) = (f_1(s),\ \ldots,\ f_N(s))$, а оценку ценности состояний s будем высчитывать по правилу

$\hat{v}(s, \theta) = f(s)^T \theta = \sum_i f_i(s)\, \theta_i$, θ вектор с параметрами.

Тогда $\text{grad}_\theta\, \hat{v}(s, \theta) = f(s)$.

В итоге правило обновления имеет вид

$\theta^{(k+1)} = \theta^{(k)} + \alpha \left(v_\pi(S_t) - \hat{v}\left(S_t, \theta^{(k)}\right)\right) f(S_t)$.

Вместо $v_\pi(S_t)$ для методов МК и TD надо подставить значения G_t или $R_{t+1} + \gamma\, \hat{v}(S_{t+1},\ \theta)$, соответственно.

Оценка стратегии

Отметим особенность приведённой выше формулы для обновления оценки по методу TD. Цель обновления $R_{t+1} + \gamma\, \hat{v}(S_{t+1}, \theta)$ зависит от текущих значений параметров. Вернёмся к функции ошибки

$$J(\theta) = \mathbb{E}_\pi[v_\pi(S_t) - \hat{v}(S_t, \theta)]^2$$

и формально подставим туда оценку

$$J(\theta) = \mathbb{E}_\pi[R_{t+1} + \gamma\, \hat{v}(S_{t+1}, \theta) - \hat{v}(S_t, \theta)]^2.$$

Тогда, градиент функции J по параметрам на самом деле должен иметь два слагаемых, одно с градиентом от $\hat{v}(S_t, \theta)$, а другое с градиентом от $\hat{v}(S_{t+1}, \theta)$. Такой формально верный подход на практике обычно ухудшает скорость сходимости.

Кроме того, желательно, чтобы цели обновления были бы константами. Тогда формально не верный градиентный спуск более схож с задачей обучения с учителем со стационарными целями. В силу этого описанный выше метод градиентного спуска ещё называют **полуградиентным**.

О сходимости

Как сказано выше, сходимость алгоритмов при использовании методов теории приближений не гарантируется, как это было для табличных методов. Но на практике сходимость, как правило, есть, если нужным образом подобрать гиперпараметры. Есть три основных причины, которые усложняют вопрос о сходимости полуградиентных методов.

1. **Использование методов приближения** для пространства состояний: обновление параметров влияет на обновление ценностей всех связанных по обновляемому параметру состояний.
2. **Бутстреппинг**: формирование целевого значения с помощью оценок ценностей состояний, в методах TD целевое значение — это $R_{t+1} + \gamma\, \hat{v}(S_{t+1},\, \theta)$.
3. **Использование методов с разделённой стратегией**: агент формирует траектории по поведенческой стратегии, но обучается при этом оптимальной стратегии.

Сходимость можно обосновать, если хотя бы одна из трёх причин отсутствует. Но ни от одной нельзя отказаться.

Методы приближения функций на основе ИНС сделали методы обучения с подкреплением применимыми в реальных сложных задачах. Бутстреппинг делает процесс обучения более эффективным, с точки зрения мгновенного усвоения полученного опыта. Методы с разделённой стратегией можно заменить на методы с единой стратегией, но тогда пропадает возможность использование полученного ранее опыта для обучения. Кроме того, методы обучения с разделённой стратегией делают процесс обучения более похожим на процесс обучения человека: опыт, полученный при действии под некоторой стратегии, можно использовать для обучения оптимальным действиям.

На практике, обычно можно добиться сходимости алгоритмов при мониторинге обучения и аккуратной настройке гиперпараметров.

О сходимости

Сводная таблица теоретических результатов о сходимости методов оценки стратегии представлена ниже

Table 5-2. *Convergence of Prediction/Estimation Algorithms*

Policy Type	Algorithm	Table Lookup	Linear	Nonlinear
On-policy	MC	Y	Y	Y
	TD(0)	Y	Y	N
	TD(λ)	Y	Y	N
	Gradient TD	Y	Y	Y
Off-policy	MC	Y	Y	Y
	TD(0)	Y	N	N
	TD(λ)	Y	N	N
	Gradient TD	Y	Y	Y

Поиск оптимальной стратегии

Сформулируем также **метод поиска оптимальной стратегии** при использовании приближений. Базовый подход для поиска оптимальных стратегии — это ОИС. В этом случае надо сразу приближать значения функции ценности действий $q_\pi(s, a)$

$\hat{q}(s, a; \theta) \approx q_\pi(s, a)$, цель в минимизации $J(\theta) = \mathbb{E}_\pi[q_\pi(s, a) - \hat{q}(s, a; \theta)]^2$ для всех пар (s, a).

Однако, всех данных у нас нет, в каждый момент времени производится минимизация величины: $\frac{1}{2}\left(q_\pi(S_t, A_t) - \hat{q}(S_t, A_t; \theta)\right)^2$.

Минимизация производится по методу градиентного спуска:

$\theta_{k+1} = \theta_k - \alpha \operatorname{grad}_\theta J(\theta_k)$, где

$$\operatorname{grad}_\theta J(\theta) = -\left(q_\pi(S_t, A_t) - \hat{q}(S_t, A_t; \theta)\right) \operatorname{grad}_\theta \hat{q}(S_t, A_t; \theta).$$

В простейшем случае $\hat{q}(s, a; \theta)$ линейно зависит от параметров. Если $f(s, a)$ — это вектор признаков состояния-действия, то линейная зависимость имеет вид

$\hat{q}(s, a; \theta) = f(s, a)^T \theta$, а $\operatorname{grad}_\theta \hat{q}(s, a; \theta) = f(s, a)$.

Поиск оптимальной стратегии

Как и выше, истинное значение $q_\pi(s,\ a)$ нам не известно и заменяется оценками по методу МК или TD:

$\theta_{k+1} = \theta_k + \alpha \left(G_t - \hat{q}(S_t, A_t; \theta_k)\right) \text{grad}_\theta\, \hat{q}(S_t, A_t; \theta_k)$.

$\theta_{k+1} = \theta_k + \alpha \left(R_{t+1} + \gamma\, \hat{q}(S_{t+1}, A_{t+1}; \theta_k) - \hat{q}(S_t, A_t; \theta_k)\right) \text{grad}_\theta\, \hat{q}(S_t, A_t; \theta_k)$.

В рамках метода Q-обучения целевым значением будет $R_{t+1} + \gamma \max_a \hat{q}(S_{t+1}, a, \theta_k)$. Можно также использовать целевое значение по методу ExpectedSARSA или n-шаговый SARSA.

Шаг обновления параметров — это шаг **оценивания** в ОИС, в котором улучшаются оценки Q-функции для заданной стратегии, чтобы стать ближе к истинным. После этого шага **улучшается** стратегия на основе ε-жадного выбора действий относительно текущих оценок Q-функции.

О сходимости

Сводная таблица сходимости алгоритмов поиска оптимальной стратегии.

Table 5-3. *Convergence of Control Algorithms*

Algorithm	Table Lookup	Linear	Nonlinear
MC control	Y	(Y)	N
On-policy TD (SARSA)	Y	(Y)	N
Off-policy Q-learning	Y	N	N
Gradient Q-learning	Y	Y	N

(Y): fluctuates around the near-optimal value function. A guarantee of convergence is off in all nonlinear cases.