ЦИФРОВАЯ КУЛЬТУРА
УНИВЕРСИТЕТ ИТМО

Обучение с подкреплением

Высшая Школа Цифровой Культуры
Университет ИТМО
dc@itmo.ru

Содержание

1 Понятие обучения с подкреплением

1.1 Базовые определения и постановка задачи

Здравствуйте, уважаемые слушатели! В этой лекции мы обсудим еще одну ветку, или еще один тип машинного обучения – обучение с подкреплением. Но начнем, конечно же, с мотивировки и отличий от тех типов, что уже были изучены ранее.

Итак, давайте вспомним с чем мы имеем дело, решая, скажем, задачу обучения с учителем? У нас есть набор предикторов (или входных переменных) $X_1, X_2, ..., X_p$ и набор откликов Y (или набор правильных ответов). Нашей задачей является обобщение «имеющегося опыта» (опыта, основанного на тренировочных данных) на «все пространство» (на произвольные тестовые данные). Например, решая задачу регрессии при $p = 1$, мы имеем в руках лишь (грубо говоря) набор из n пар данных $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. Наша же цель – построить зависимость вида $Y = f(X)$, которая способна предсказать «правильный ответ» уже для любого X из возможного множества значений предиктора X_1.

Аналогично и с задачей классификации: просто вспомните, как, зная классы лишь конечного числа тренировочных объектов, в результате классификации все пространство признаков (с бесконечным числом объектов) «раскрашивается» в разные цвета, отвечающие тому или иному классу. Вот оно – обобщение на все пространство. Понятно, что аналогичные рассуждения справедливы и при рассмотрении обучения без учителя.

Однако в реальной жизни обучение происходит далеко не всегда по схеме, в которой известны «верные ответы». Скажем, разработка учеными нового лекарственного препарата, или вывод математиками очередной хитрой формулы, – это процессы, не имеющие откликов, кроме финального: работает препарат или нет, получилась формула или нет. Сам же процесс создания (будь то формулы или препарата), скорее всего, описывается примерно так: мы что-то делаем, в результате наших действий что-то происходит, этот результат мы как-то оцениваем, обдумываем, а затем продолжаем что-то делать, стараясь приблизиться к желаемому результату. Согласитесь, под такую вольно описанную логику подходит куча различных процессов, встречающихся в повседневной жизни на каждом шагу:

1. Например, а как ребенок учится вставать, научившись ползать (описание процесса может быть не очень точным)? Сначала он предпринимает множество попыток встать с колен, но, по понятным причинам, постоянно падает. Однако через какое-то время он понимает (конечно, из опыта), что когда его поддерживает рука мамы, то встать гораздо проще. Значит, нужно за что-то держаться, вставая. Стол, стул, да хотя бы

стена – все подойдет. Остается только немного покарабкаться, и вуаля – человек научился вставать.

2. А как проходит футбольный матч? Конечно, у каждой команды есть заранее продуманная тактика, учитывающая как сильные, так и слабые стороны как соперников, так и себя (кстати, наработанная опытом). Но ход игры слишком непредсказуем: почему-то именно сегодня нападающий достаточно легко может попасть к воротам соперника именно со стороны правого фланга. Видимо, защитники думают о чем-то другом. Что же, это «на руку», значит по этому флангу и имеет смысл в основном атаковать.

Конечно, можно привести еще множество различных примеров, но, скорее всего, идея ясна. Давайте теперь подойдем к некоторой формальной постановке задачи и введем определения, которые будем повсеместно использовать в дальнейшем.

Итак, говоря несколько грубо, обучение с подкреплением – это раздел машинного обучения, основывающийся на следующей парадигме: обучаемая модель не имеет детальных сведений об окружающей ее системе, однако может производить в ней (в этой системе) какие-то действия, анализируя реакцию системы на эти действия. Иными словами, анализируя подкрепления. Ясно, что как модель воздействует на систему, так и система воздействует на модель, поэтому пару система-модель разлучать нельзя. Так что следующее определение носит весьма условный характер.

Определение 1.1.1 *Модель, обучающуюся в рамках задачи обучения с подкреплением, часто называют агентом. Окружающую агента систему часто называют средой.*

Перейдем к формальному описанию процесса взаимодействия агента и среды.

1. Пусть агент и среда взаимодействуют в дискретные моменты времени $t \in \{0, 1, 2, ..., n\}$, $n \in \mathbb{N} \cup \{0, \infty\}$: количество взаимодействий может быть как конечным (если n конечно), так и бесконечным.

2. На каждом шаге агент находится в каком-то состоянии $s_t \in S$, где S – множество всех возможных состояний.

3. На основании состояния s_t, агент выбирает некоторое действие $a_t \in A(s_t)$, где $A(s_t)$ – множество действий, доступных агенту в состоянии s_t.

4. Совершив действие a_t, среда генерирует награду r_{t+1} и переводит агента в состояние s_{t+1}.

5. Шаги 3-4 повторяются, если не достигнут какой-то критерий остановки.

В дальнейшем мы подробно поясним, что означают слова «генерирует», «переводит», и опишем всю схему математически корректно. Сейчас же, чтобы закрепить общее понимание происходящего, приведем весьма условный пример диалога между средой и агентом:

1. **Среда:** Агент, ты в состоянии s_0. Тебе доступны действия с номерами $3, 4, 5$ ($A(s_0) = \{3, 4, 5\}$).
   Агент: Совершаю действие с номером 4 ($a_0 = 4$).

2. **Среда:** Ты получаешь награду в 3 единицы ($r_1 = 3$) и переходишь в состояние s_1. Теперь тебе доступны действия с номерами $1, 4, 5$ ($A(s_1) = \{1, 4, 5\}$).
   Агент: Совершаю действие с номером 5 ($a_1 = 5$).

3. **Среда:** Ты получаешь награду в -1 единицу ($r_2 = -1$) и переходишь в состояние s_2. Тебе доступны ... Ну и так далее.

Как мы видим, награды (или подкрепления) могут быть как положительными, так и отрицательными. Задача агента – максимизировать накопительную награду, опирается на следующую гипотезу (так называемую гипотезу о вознаграждении – Reward Hypothesis): «Все цели могут быть описаны путем максимизации ожидаемого совокупного вознаграждения» («All goals can be described by the maximisation of expected cumulative reward»). О том, что понимается под этими словами, мы поговорим чуть позже.

Итак, надеемся, что схема взаимодействия прозрачна и ясна. Однако остается еще одна проблема, решению которой, по большому счету, и будет посвящено все дальнейшее изложение: как выбирать действие a_t на шаге t из набора возможных действий $A(s_t)$? Так мы приходим к понятию стратегии.

Определение 1.1.2 *Набор вероятностей $\pi_t(a|s_t)$ выбора доступного действия a в момент времени t (в случае, когда агент находится в состоянии $s_t \in S$), называется стратегией агента.*

Иными словами, в случае, когда в состоянии s_t в момент времени t доступно более одного действия в наборе $A(s_t)$, выбор совершаемого действия производится вероятностным образом: согласно стратегии $\pi_t(a|s_t)$. Как же задать такую стратегию, которая максимизирует ожидаемое совокупное вознаграждение? Это и есть тот самый «вопрос на миллион», который мы и будем обсуждать в дальнейшем.

1.2 Компромисс изучение-применение

Перед тем как переходить к рассмотрению конкретных примеров и задач, обратим наше внимание на так называемую проблему (или конфликт, компромисс) между изучением и применением («exploration-exploitation»). Принцип применения основывается на следующем предложении: на каждом шаге предлагается принять наилучшее (или наиболее выгодное) решение на основе имеющейся информации. Принцип изучения меняет концепцию: собрать как можно больше информации для (возможно) более успешного применения в дальнейшем. Идею упомянутого конфликта достаточно просто понять из рисунка 1.

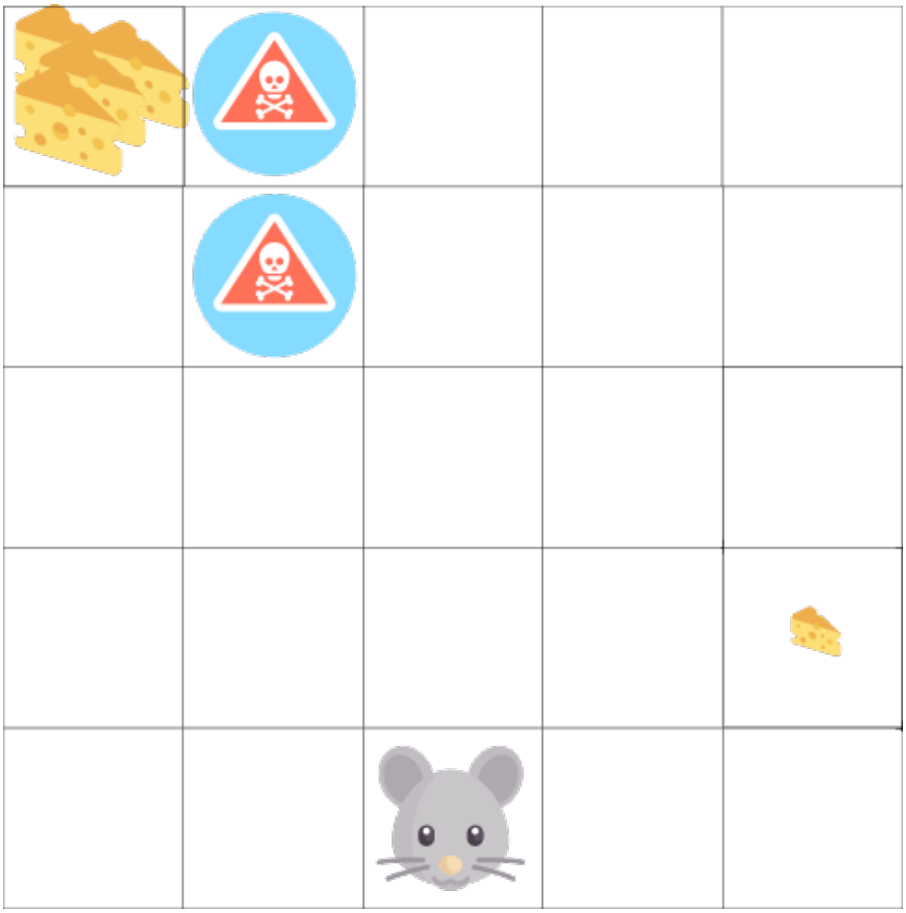


Рис. 1: Компромисс «изучение-применение»

Предположим, что за один ход мышонок (наш агент) совершает одно перемещение (действие). Игра останавливается либо когда мышонок съедает сыр, либо когда попадает в ловушку, либо когда у него заканчивается запас ходов (пусть всего их, скажем, 10). Производится некоторое конечное количество экспериментов (игр), в результате чего подсчитывается общее количество добытого сыра. Ясно, что чем больше в результате всех игр добыто сыра – тем лучше. Кроме того, не добраться до сыра (истратить все ходы) и

не попасть в ловушку (не убить агента) – результат, который намного лучше, чем попасть в ловушку (то есть убить агента).

Из рисунка очевидно, что мышонку проще (и, вероятно, быстрее, а также безопаснее) всего каждый раз добираться до маленького кусочка сыра, расположенного поблизости, – так он гарантированно будет получать хоть и маленькую, но награду в результате каждой игры. При таком подходе (который часто нарекается подходом «лучше синица в руках») внушительные запасы сыра, лежащие рядом с ловушками, обесцениваются вовсе. Описанное является ничем иным, как принципом «применения».

С другой стороны, мышонок может пожертвовать сиюминутным результатом (и, быть может, даже жизнью), чтобы исследовать окружающую среду и попытаться найти пусть и более сложную и опасную дорогу, но к более значимой награде, а значит в итоге увеличить добычу сыра (или награды) «в среднем».

Итак, тезисно компромисс между изучением и применением может быть охарактеризован следующими пунктами:

1. Необходимо получить достаточно информации об окружающей среде, чтобы решения (а значит и результаты) были наилучшими «в среднем».

2. Лучшая долгосрочная стратегия допускает (и даже обязательно включает в себя) краткосрочные потери.

Описанная проблема между изучением и применением – не что-то, созданное на пустом месте. Эта проблема регулярно возникает и в реальной жизни. Например, пусть вы решили ужинать в ресторане. Применение – это выбор проверенного ресторана, который стабильно не подводит ни качеством блюд, ни их ценой. Изучение – это выбор и опробование какого-то нового заведения: а вдруг там окажется лучше?

Ну что, надеемся, что как проблема, так и постановка задачи обучения с подкреплением ясна и прозрачна. Перейдем теперь к разбору одной из самых понятных и популярных задач – к задаче о k-руком бандите.

1.3 Постановка задачи о k-руком бандите

Как мы уже сказали, одним из самых простых примеров, иллюстрирующих идею обучения с подкреплением, является так называемая задача о k-руком бандите. Название задачи происходит из аналогии с игральным автоматом (хорошо известным, как «однорукий бандит»). В нашем же случае будем считать, что мы имеем дело с автоматом, имеющим k ручек.

Давайте для начала определимся: что в этой задаче является агентом, что средой, что стратегией и так далее. Агент – это человек, которому в каждый момент времени t в любом состоянии $s \in S$ доступно любое из k

одних и тех же действий – выбрать одну из k ручек. Конечно, каждый раз этот выбор осуществляется в соответствии с какой-то своей стратегией π_t. Выбрав действие a_t в момент времени t, агент получает заранее неизвестное вознаграждение r_{t+1}, диктуемое средой – k-руким автоматом, и переходит в следующее состояние.

Замечание 1.3.1 *Отметим отдельно, что так как возможности (или набор возможных действий $A(s)$) агента в каждом состоянии $s \in S$ одинаковы – выбор одной из k ручек, то хочется сказать, что и состояния s агента на каждом шаге одинаковы (или вообще, что состояние всего одно). Такое предположение, вообще говоря, неверно, так как в каждом состоянии $s_t \in S$ в момент времени t, агент действует согласно некоторой, вообще говоря, различной стратегии π_t.*

Теперь давайте обратимся к естественному вопросу: из каких соображений дергать за ручки? Если бы мы заранее знали ценность каждого действия, то логичной жадной (недальновидной) стратегией была бы такая: в момент времени t выбрать то действие (ту ручку), вознаграждение r_{t+1} за выбор которой будет максимальным. Мы же будущего не знаем, поэтому способны оценивать только результат наших прошлых действий: «дернул» за эту ручку – получил столько-то, за другую – столько-то, и так далее. В итоге оказывается разумным ввести следующее определение.

Определение 1.3.1 *Пусть $q_0(a) \in \mathbb{R}$ – начальная оценка ценности действия a (устанавливается исследователем).Оценкой ценности действия a в момент времени $t \geq 1$ называется величина*

$$q_t(a) = \begin{cases} q_0(a), & \text{действие } a \text{ не выбрано ни разу} \\ \dfrac{\sum\limits_{i=0}^{t-1} r_{i+1} \mathsf{I}(a_i = a)}{\sum\limits_{i=0}^{t-1} \mathsf{I}(a_i = a)}, & \text{иначе} \end{cases},$$

где $\mathsf{I}(a_i = a)$ – индикатор события $a_i = a$: он равен единице, если на i-ом шаге выбрано действие a, и нулю иначе.

Замечание 1.3.2 *Написанное выражение (в случае, если действие a выбрано хотя бы один раз) –среднее арифметическое вознаграждений, полученных при выборе действия a до момента времени t.*

Рассмотрим пример.

Пример 1.3.1 *Пусть в процессе некоторой игры агент на каждом шаге совершает одно из двух возможных действий: a или b, в соответствии с некоторой стратегией, и на каждом шаге t получает некоторую награду r_{t+1}. Данные (с использованием введенных выше обозначений) представлены в таблице.*

Шаг	***Действие***	***Награда***
$t = 0$	$a_0 = b$	$r_1 = 3$
$t = 1$	$a_1 = a$	$r_2 = 0$
$t = 2$	$a_2 = a$	$r_3 = 8$
$t = 3$	$a_3 = b$	$r_4 = 1$
$t = 4$	$a_4 = b$	$r_5 = 5$

Найдем оценки ценностей действий a и b на пятом шаге. Действие a было выбрано 2 раза, при этом были получены награды 0 и 8. Тогда оценка ценности действия a на пятом шаге может быть найдена следующим образом (как мы и говорили – как среднее арифметическое ранее полученных вознаграждений):

$$q_5(a) = \frac{0 + 8}{2} = 4.$$

Конечно, то же самое получится и при использовании «оригинального» соотношения из определения:

$$q_5(a) = \frac{\sum\limits_{i=0}^{5-1} r_{i+1}\mathsf{I}(a_i = a)}{\sum\limits_{i=0}^{5-1} \mathsf{I}(a_i = a)} = \frac{3 \cdot 0 + 0 \cdot 1 + 8 \cdot 1 + 1 \cdot 0 + 5 \cdot 0}{0 + 1 + 1 + 0 + 0} = 4.$$

Аналогичным образом найдем оценку ценности действия b на пятом шаге:

$$q_5(b) = \frac{3 + 1 + 5}{3} = 3.$$

Можно заметить, что оценка ценности действия a, посчитанная на пятом шаге, выше, чем оценка ценности действия b, однако все может измениться, если накопить большее количество информации о получаемых наградах.

Использование описанного подхода при оценке ценности влечет некоторые издержки с точки зрения вычислений и хранения информации. Нетрудно заметить, что для оценки ценности приходится хранить информацию обо всех полученных ранее вознаграждениях. Резонно задуматься, а существует ли менее затратный способ? Оказывается, существует. Он дается следующей леммой.

Лемма 1.3.1 (Рекуррентная формула оценки ценности) *Пусть действие a в течение $(t-1)$ шагов выбрано ровно $n \leq (t-1)$ раз, $q_t(a)$ – оценка его ценности на шаге t. Если оно выбрано на шаге t, и за него получено вознаграждение r_{t+1}, то*

$$q_{t+1}(a) = q_t(a) + \frac{1}{n+1}\left(r_{t+1} - q_t(a)\right),$$

иначе $q_{t+1}(a) = q_t(a)$, $q_0(a) \in \mathbb{R}$ – начальная оценка ценности.

Замечание 1.3.3 *Использование введенной выше рекуррентной формулы позволяет хранить только два параметра: текущее значение ценности действия и число активаций этого действия.*

Доказательство. Случай, когда на шаге t действие a не выбрано, очевиден – оценка ценности действия не меняется. Содержательным для рассмотрения является случай, когда на шаге t действие a все-таки выбрано. По определению,

$$q_t(a) = \frac{\sum\limits_{i=0}^{t-1} r_{i+1}\mathsf{I}(a_i = a)}{\sum\limits_{i=0}^{t-1} \mathsf{I}(a_i = a)} = \frac{\sum\limits_{i=0}^{t-1} r_{i+1}\mathsf{I}(a_i = a)}{n},$$

где последнее равенство справедливо в силу условия: действие a до шага t было выбрано n раз. Но тогда $nq_t(a)$ – это сумма вознаграждений, полученных за выбор действия a до шага $(t-1)$ включительно. Раз действие выбрано и на шаге t, то, так как за него получено вознаграждение r_{t+1} (и теперь оно выбрано $(n+1)$ раз), для его оценки ценности имеем

$$q_{t+1}(a) = \frac{1}{n+1}\left(r_{t+1} + nq_t(a)\right) = \frac{1}{n+1}\left(r_{t+1} + (n+1)q_t(a) - q_t(a)\right) =$$

$$= q_t(a) + \frac{1}{n+1}\left(r_{t+1} - q_t(a)\right).$$

□

Замечание 1.3.4 *Помимо среднего арифметического полученных вознаграждений (в частности, чтобы динамически менять стратегию), часто используют и следующее правило получения $q_{t+1}(a)$:*

$$q_{t+1}(a) = q_t(a) + \alpha_t\left(r_{t+1} - q_t(a)\right).$$

В частности, если $\alpha_t = \alpha$ – некоторая константа, то мы получаем экспоненциальное скользящее среднее:

$$q_{t+1}(a) = \alpha r_{t+1} + (1-\alpha)q_t(a).$$

Это соотношение показывает, что если $\alpha \in (0,1)$, то наиболее ценными оказываются сиюминутные награды (принцип «лучше синица в руках»), а если $\alpha > 1$ – наоборот. Сейчас мы на этом подробно останавливаться не будем.

1.4 Жадная стратегия

Итак, мы переходим к рассмотрению различных стратегий выбора действий на шаге t. Наверное, первый и самый наивный способ максимизации награды – это выбор на шаге t того (или тех действий), оценка ценностей которого (или которых) максимальна. Пусть A_t – множество действий в состоянии s_t, обладающих максимальной (и, как следствие, одинаковой) оценкой ценности. Говоря математическим языком, множество A_t – это множество

$$A_t = \operatorname*{Arg\,max}_{a \in A(s_t)} q_t(a).$$

Так как дальнейший выбор мы осуществляем только исходя из величины оценки ценности действия, а все действия из множества A_t на шаге t одинаково ценны (другие же не рассматриваются вовсе), то логично, что любое $a \in A_t$ имеет смысл выбирать равновероятно, а значит

$$\pi_t(a|s_t) = \begin{cases} \frac{1}{|A_t|}, & a \in A_t \\ 0, & a \notin A_t \end{cases},$$

где $|A_t|$ – количество элементов в множестве A_t. Таким образом, действия, обладающие максимальной оценкой ценности, выбираются с вероятностью $\frac{1}{|A_t|}$, остальные же действия игнорируются.

Определение 1.4.1 *Стратегия, введенная выше, называется жадной стратегией.*

Можно рассмотреть и полную противоположность жадной стратегии – так называемую абсолютно исследовательскую стратегию.

Определение 1.4.2 *Если в результате обучения на каждом шаге t любое действие из множества $A(s_t)$ всех доступных в состоянии s_t действий выбирается с одинаковой вероятностью, то такая стратегия называется абсолютно исследовательской.*

С точки зрения компромисса «изучение-применение» жадная стратегия – это в чистом виде применение, без какой-либо возможности изучения. Абсолютно исследовательская стратегия же скорее отвечает за постоянное изучение. Рассмотрим применение жадной стратегии на следующем примере.

Пример 1.4.1 *Владелец небольшого магазина рассматривает возможности увеличения выручки. Он прочитал, что изменение расположения товара в магазине может стимулировать покупателей к импульсивным покупкам. Например, если расположить печенье рядом с чаем, то вероятность покупки печенья увеличивается, что увеличивает выручку. Владелец решил опробовать три различные конфигурации и понаблюдать за выручкой магазина.*

Итак, определимся что есть что. Пусть «ручка» – это одна из трех выбранных конфигураций (a, b, c), t – номер дня, в который проводится эксперимент, награда – сумма потраченных покупателями денег в течение дня (выручка магазина). Пусть каждая игра состоит из 1000 *итераций (шагов) (по сути – это количество дней, в течение которых проводится эксперимент). Награды в случае выбора конфигурации a, b или c – это выборки из генеральных совокупностей ξ_1, ξ_2, ξ_3, имеющих распределения $\mathsf{N}_{2,1}$, $\mathsf{N}_{2.5,1}$ и $\mathsf{N}_{3,1}$, соответственно. Говоря формально, награды в рассматриваемом примере могут быть отрицательными, и могут трактоваться либо как ошибки в данных, либо как убытки магазина: утерянный, просроченный, разбитый товар, и так далее.*

Итак, как же поступает агент при использовании жадной стратегии? Пусть $q_0(a) = q_0(b) = q_0(c) = 0$. Так как в момент времени $t = 0$ оценки ценности всех действий (конфигураций) одинаковы, то выбор среди них производится случайным образом, а значит любая из трех конфигураций будет выбрана равновероятно. В итоге, стратегия на нулевом шаге такова:

$$\pi_0(a|s_0) = \pi_0(b|s_0) = \pi_0(c|s_0) = \frac{1}{3}.$$

Пусть, скажем, выбрана конфигурация c (то есть $a_0 = c$). Тогда оценка ценности этой и только этой конфигурации станет положительной (предполагается, что выручка магазина в первый день все же положительна), а значит, в соответствии с алгоритмом, на первой итерации будет снова выбрана конфигурация c (то есть $a_1 = c$), и так далее, а значит стратегия такова:

$$\pi_t(c|s_t) = 1, \quad t \geq 1.$$

В итоге, в нашем примере (согласно жадной стратегии) будет всегда выбираться та конфигурация, которая была выбрана (случайным образом) на нулевом шаге.

На рисунке 2 представлены результаты трех игр. Синяя кривая отвечает ситуации, когда в первой игре первым было выбрано действие $a_0 = c$ и, как мы только что отметили, оно же выбиралось на всех последующих шагах ($a_t = c$, $t \in \{0, 1, ..., 999\}$). Сам график показывает среднее значение

награды или, что в нашем случае одно и то же, оценку ценности действия c *на каждом шаге.*

Во второй игре на первом шаге выбор пал на конфигурацию a*, то есть* $a_0 = a$ *(красная кривая). График снова показывает величину средней награды при таком выборе начальной конфигурации (опять же,* $a_t = a$*,* $t \in \{0, 1, ..., 999\}$*).*

В третьей игре первым (как и последующими) было выбрано действие b*. Значения средней награды в этом случае изображены оранжевым цветом. Видно, что максимально возможное среднее значение награды достигнуто только в одном из трех случаев. Кроме того, в одном из трех случаев достигнуто минимальное среднее значение награды.*

«Игры», описанные в данном примере, на практике могут быть реализованы следующим образом: есть три магазина некоторой сети, находящиеся в одинаковых условиях. Каждый из них независимо в течение 1000 *дней использует единственную конфигурацию и оценивает среднюю выручку. Далее происходит сравнение оценок и делается вывод о том, какая конфигурация работает лучше (если такая конфигурация имеется).*

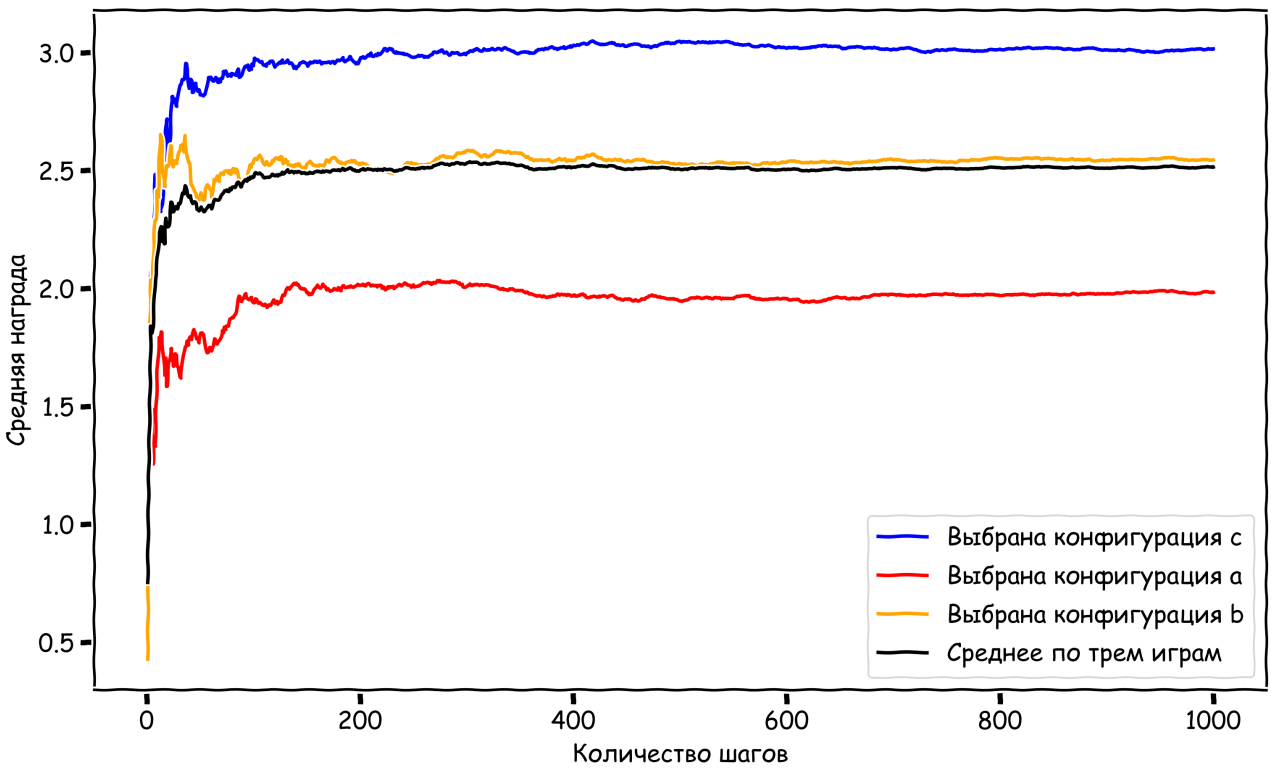


Рис. 2: Средняя выручка в зависимости от выбранной конфигурации

Понятно, что выбор лучшей начальной конфигурации в нашем примере – дело случая. Именно поэтому, с точки зрения оценки самой стратегии, нам интересно, как ведет себя алгоритм «в среднем». Для этого проводят серию

игр. На рисунке 2 черным цветом иллюстрируется среднее значение средней награды по трем играм на каждом шаге. На рисунке 3 представлены результаты применения жадной стратегии при проведении 1000 игр по 1000 итераций в каждой. Черной линией отмечено среднее значение средней награды по всем играм. В отношении примера с конфигурациями магазина, описанный подход может быть реализован следующим образом: есть 1000 магазинов некоторой сети, находящихся в одинаковых условиях. В каждом из магазинов проводится оценка одной из трех конфигураций в течение 1000 дней. Тогда при использовании жадной стратегии, «в среднем» каждый магазин будет генерировать ежедневную выручку на уровне 2.5 единиц.

Замечание 1.4.1 *В дальнейшем, для сравнения стратегий мы будем также проводить* 1000 *игр по* 1000 *итераций в каждой, а результаты усреднять. Поскольку полученные результаты не будут являться результатами применения непосредственно стратегии, а будут являться усреднением по тысячекратному применению стратегии, то будем использовать термин алгоритм (например, жадный алгоритм).*

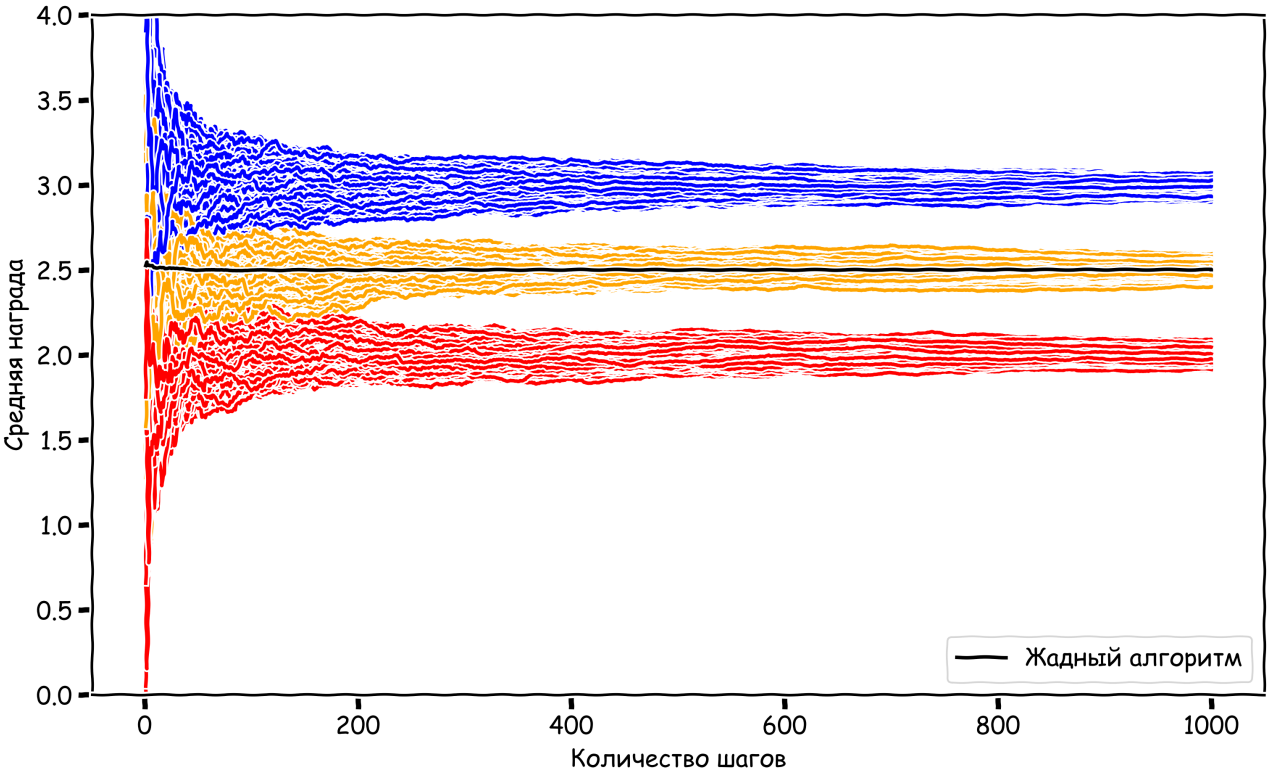


Рис. 3: Иллюстрация жадного алгоритма

Конечно, существенным недостатком жадного алгоритма в рассмотренном примере является то, что исследование среды прекращается после первого же шага, что, конечно, серьезно сказывается на результатах. Как мы

уже отмечали, на рисунке 2 видно, что «жадный» принцип в рассмотренном примере с магазинами только в одном из трех случаев приводит к максимальной возможной средней выручке. Более того, в одном из трех случаев он приводит даже к минимально возможной средней выручке. Причина этому – запрет на какое-либо исследование. Исправить эту ситуацию помогает так называемая ε-жадная стратегия.

1.5 ε-жадная стратегия

Итак, как мы уже поняли ранее, для получения более приемлемого результата скорее всего имеет смысл использовать что-то вроде смеси абсолютно исследовательской и жадной стратегий. Ведь чем лучше агент понимает, как устроена среда, тем более выгодные действия он может совершать. В то же время, исследование ради исследования – малоинтересная задача, так что «жадность» полностью отключать, конечно же, нельзя, ведь принцип: «если дают, надо брать», пока еще никто не отменял.

Определение 1.5.1 *Пусть $0 < \varepsilon < 1$. Стратегия, $\pi_t(a|s_t)$, которая в момент времени t с вероятностью $(1-\varepsilon)$ ведет себя как жадная, а с вероятностью ε как абсолютно исследовательская, называется ε-жадной.*

Таким образом, введенная стратегия – это попытка обеспечить компромисс между применением и изучением. Полезно отметить, что при $\varepsilon = 0$ или $\varepsilon = 1$ (хотя это и запрещено в определении), ε-жадная стратегия становится жадной и абсолютно исследовательской, соответственно.

Для практики бывает полезным следующее задание ε-жадной стратегии:

$$\pi_t(a|s_t) = \begin{cases} \dfrac{1-\varepsilon}{|A_t|} + \dfrac{\varepsilon}{|A(s_t)|}, & a \in A_t \\ \\ \dfrac{\varepsilon}{|A(s_t)|}, & a \notin A_t \end{cases},$$

где

$$A_t = \operatorname*{Arg\,max}_{a \in A(S_t)} q_t(a).$$

Замечание 1.5.1 *Давайте поясним, почему написанные формулы согласованы с данным выше определением. Ответим на вопрос: какова вероятность выбрать действие $a \in A_t$? Это действие выбирается с вероятностью $\frac{1}{|A_t|}$, если используется жадная стратегия, и с вероятностью $\frac{1}{|A(s_t)|}$, если используется абсолютно исследовательская стратегия. В силу того, что жадная стратегия выбирается с вероятностью $(1-\varepsilon)$, а абсолютно*

исследовательская – с вероятностью ε, получаем следующую вероятность выбрать действие $a \in A_t$:

$$\frac{1-\varepsilon}{|A_t|} + \frac{\varepsilon}{|A(s_t)|}.$$

Аналогично, действие $a \notin A_t$ может быть выбрано только в случае, когда выбрана абсолютно исследовательская стратегия, причем в этом случае оно выбирается с вероятностью $\frac{1}{|A(s_t)|}$. Так как абсолютно исследовательская стратегия выбирается с вероятностью ε, то вероятность выбрать действие $a \notin A_t$ равна

$$\frac{\varepsilon}{|A(s_t)|}.$$

Покажем, что введенное определение корректно, то есть что введенная «стратегия» является стратегией в смысле введенного ранее определения.

Лемма 1.5.1 *ε-жадная стратегия является стратегией.*

Доказательство. Достаточно доказать, что

$$\sum_{a \in A(s_t)} \pi_t(a|s_t) = 1.$$

Это и правда так, ведь

$$\sum_{a \in A(s_t)} \pi_t(a|s_t) = |A_t| \left(\frac{1-\varepsilon}{|A_t|} + \frac{\varepsilon}{|A(s_t)|} \right) + (|A(s_t)| - |A_t|) \left(\frac{\varepsilon}{|A(s_t)|} \right) = 1.$$

□

Сравнение жадного и ε-жадного алгоритмов для примера с различными конфигурациями магазина (1000 игр по 1000 итераций в каждой) представлено на рисунке 4. Отметим, что для большей информативности этот и последующие графики будут строиться, начиная с шага $t = 1$. Можно заметить, что при использовании различных значений ε, результаты несколько отличаются. Видно, например, что при значении $\varepsilon = 0.2$ (то есть в случае, когда дается больше «свободы» абсолютно исследовательской стратегии), высокие значения средней награды достигаются намного быстрее, чем в случае $\varepsilon = 0.1$ и $\varepsilon = 0.01$. Однако, когда ценности действий изучены достаточно (начиная примерно с трехсотого шага), алгоритм при $\varepsilon = 0.2$ начинает проигрывать случаю $\varepsilon = 0.1$ из-за того, что доля его исследовательских действий выше и действительно оптимальное действие (оценка ценности которого уже достаточно точна) выбирается реже. Случаю $\varepsilon = 0.01$ нужно гораздо больше времени на оценку ценностей всех действий. Потенциал этого алгоритма раскрывается в случае использования не 1000, а 10000 шагов (рисунок 5).

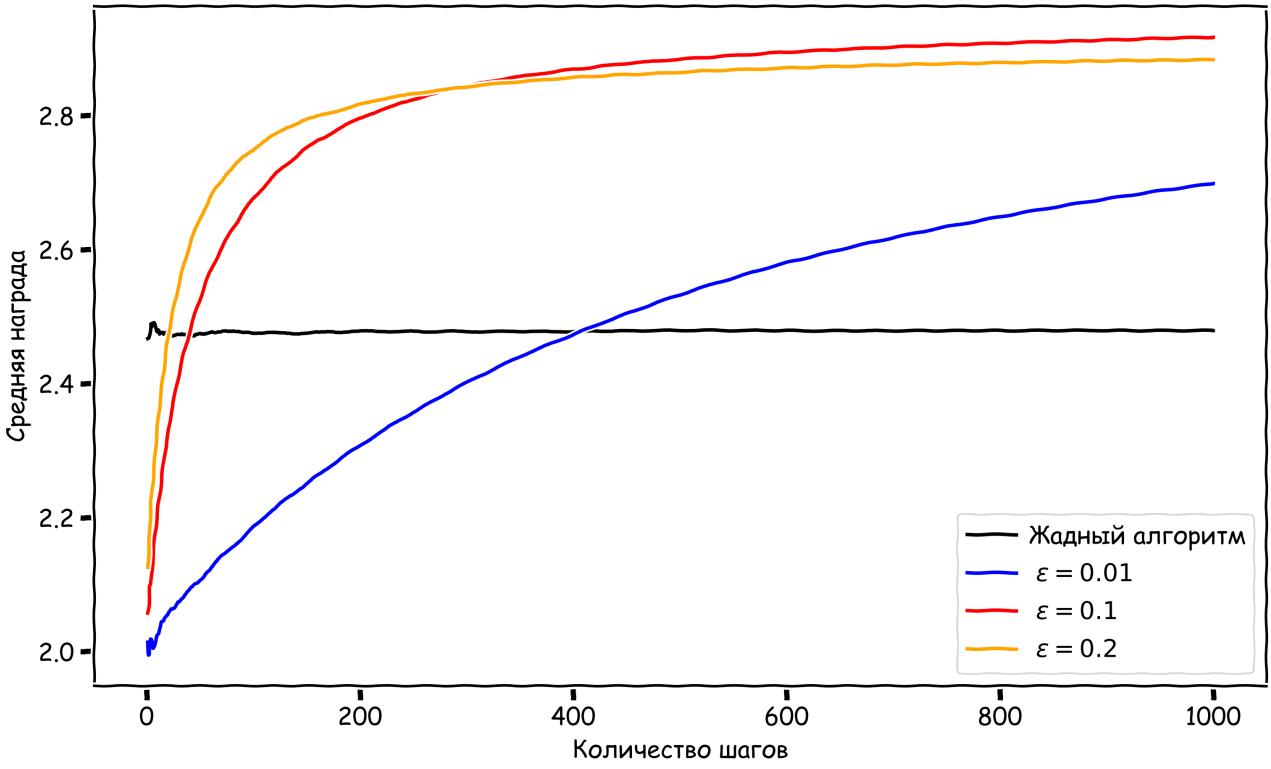


Рис. 4: Сравнение ε-жадных алгоритмов

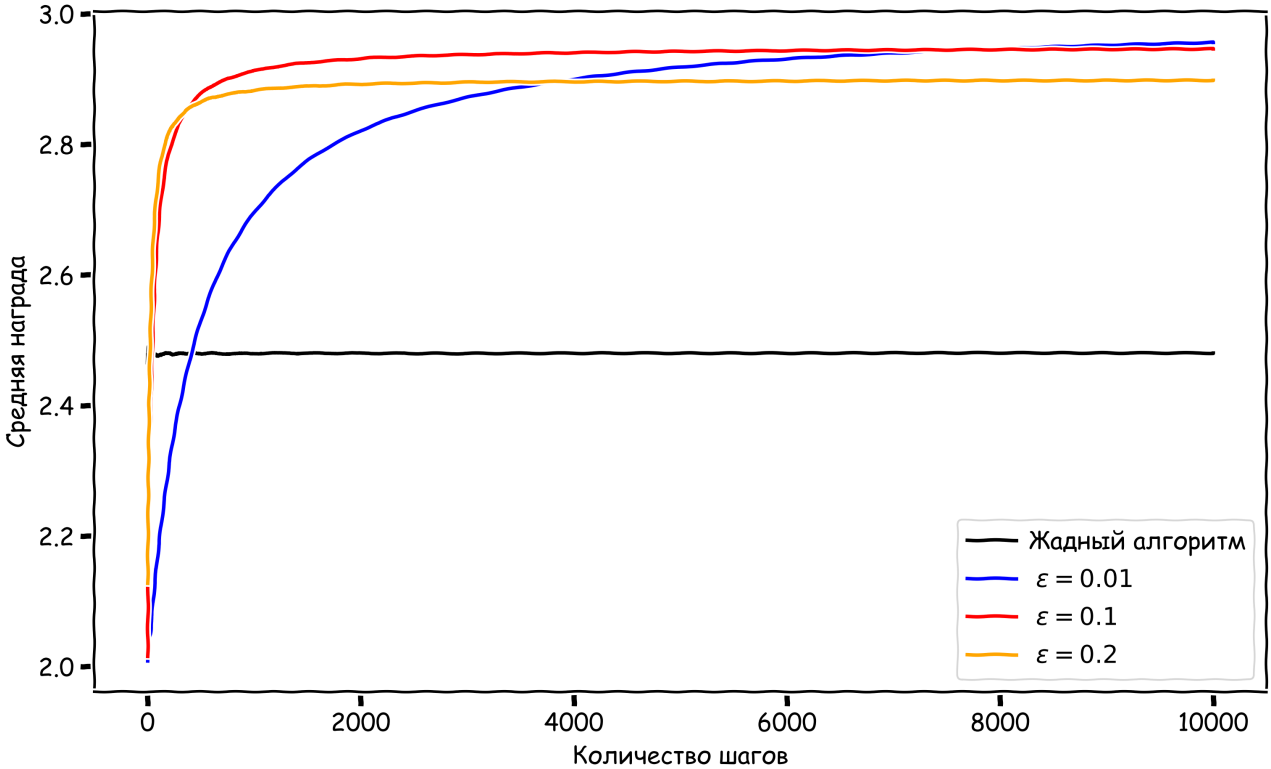


Рис. 5: Сравнение ε-жадных алгоритмов

Можно заметить, что в конечном итоге при $\varepsilon = 0.01$ значение средней награды оказывается выше конкурентов, начиная, примерно, с шага $t = 8000$.

Замечание 1.5.2 *Важно отметить, что каждый из рассмотренных ε-жадных алгоритмов работает куда лучше, чем жадный алгоритм. Более того, значения средней награды в каждом из рассмотренных случаев достаточно быстро сходятся к значениям, близким к среднему максимуму – к 3. Скачкообразное поведение некоторых графиков при малом количестве шагов обуславливается неточными сведениями об оценке ценности действий (сведения просто еще не успели накопить). Для их уточнения действия выбираются в большей степени хаотично, что и влечет скачки средней награды.*

Замечание 1.5.3 *Напоследок отметим следующее эвристическое замечание: со временем (то есть с ростом t) параметр ε имеет смысл уменьшать, ведь оценки ценностей действий становятся точнее и имеет смысл большее внимание уделить применению, нежели изучению. Например, в качестве зависимости величины ε от времени t можно рассматривать функцию вида*

$$\varepsilon(t) = \frac{1}{1 + t\beta},$$

где t – текущий момент времени, а $\beta \in (0, 1)$ – некоторый коэффициент, отвечающий за скорость уменьшения ε. В качестве β можно брать, например, $\frac{1}{k}$, где k – число рукоятей бандита. Выбор функции, конечно, отдается на откуп исследователю.

1.6 Softmax-стратегия

Использование ε-жадной стратегий имеет определенный недостаток. В случае, когда идет процесс «изучения», действия выбираются с одинаковой вероятностью, что явно не лучшим образом сказывается на конечной цели – наибольшей сумме поощрений (ведь действия с высокими и низкими оценками ценности имеют одинаковые шансы быть выбранными). Принцип, которым руководствуется softmax-стратегия, можно описать следующим образом: уменьшение потерь при исследовании на итерации t за счёт более редкого выбора действий a, которые имеют небольшую оценку ценности $q_t(a)$. Чтобы этого добиться, для каждого действия вычисляется свой «весовой коэффициент», на базе которого и происходит выбор действия. Точнее, стратегия дается следующим определением.

Определение 1.6.1 *Стратегия, при которой вероятность выбора действия $a_t \in A(s_t)$ в момент времени t равна*

$$\pi_t(a|s_t) = \frac{e^{q_t(a)/\tau}}{\sum\limits_{a \in A(s_t)} e^{q_t(a)/\tau}}, \quad \tau > 0,$$

называется softmax-стратегией.

Параметр $\tau > 0$, называемый еще и температурой, является параметром модели. При больших значениях τ стратегия становится более исследовательской, при малых – приближается к жадной. На рисунке 6 (рисунок снова строится при $t \geq 1$) представлено сравнение ε-жадного алгоритма с параметром $\varepsilon = 0.1$ и алгоритма softmax при различных значениях температуры τ. Легко понять, что конкретно в нашем случае ε-жадный алгоритм показывает

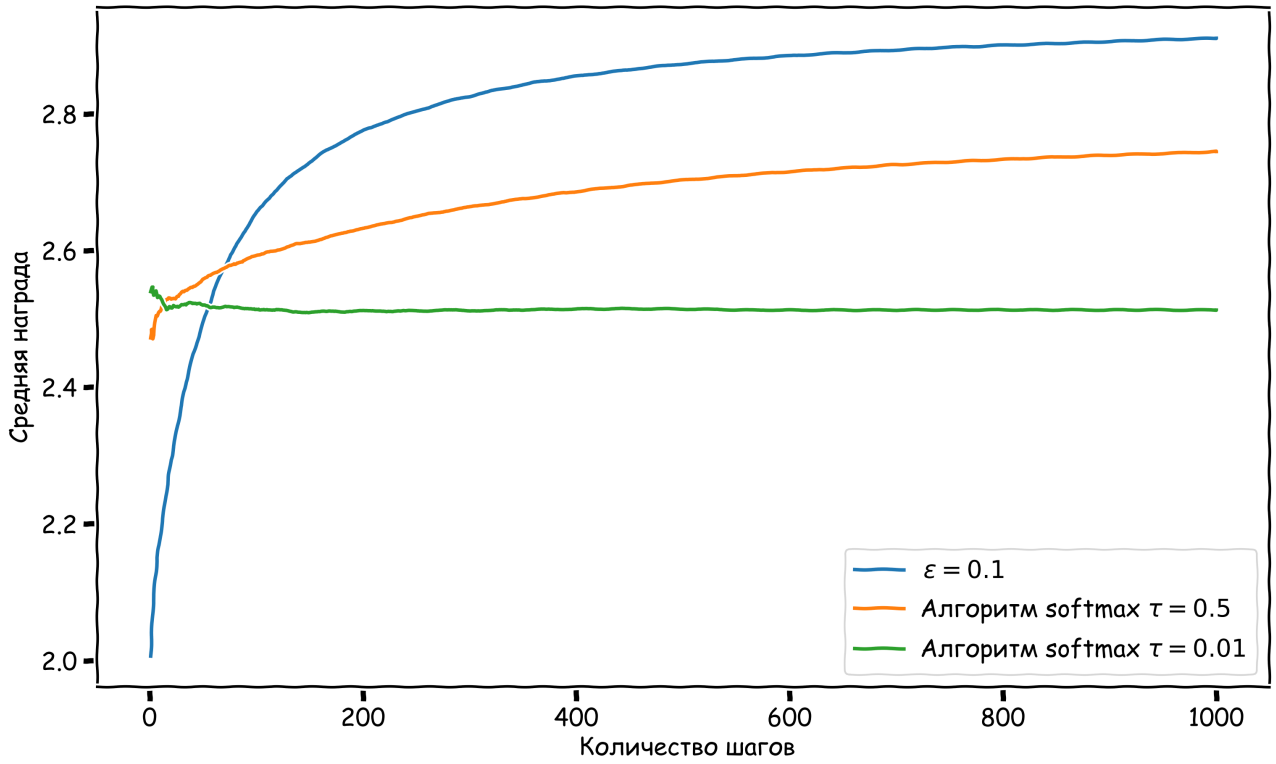


Рис. 6: Сравнение ε-жадного алгоритма и алгоритма softmax

лучший результат. При этом выбор того или иного метода обычно отдается на откуп исследователю. Заметим также, что при $\tau = 0.01$ softmax-алгоритм очень близок к жадному.

Замечание 1.6.1 *Введенное выше распределение вероятностей в физике часто называется распределением Гиббса (или Больцмана). Еще раз отметим, что при $\tau \to +\infty$ мы получаем почти одинаковые вероятности*

выбора действий $a \in A(s_t)$, а значит стратегия стремится к абсолютно исследовательской. В случае же, когда $\tau \to 0+$, стратегия стремится к жадной, так как среди всех экспонент больший вес имеет та, для которой $q_t(a)$ больше.

Замечание 1.6.2 *Как и в случае ε-жадной стратегии, параметр τ имеет смысл уменьшать со временем. Как это делать и делать ли вовсе – вопросы, которые отдаются на откуп исследователю.*

1.7 Метод UCB (Upper Confidence Bound)

Рассмотрим еще один способ нахождения компромисса при решении проблемы «изучение-применение». Этот метод получил название метода максимальной верхней оценки (Upper Confidence Bound (UCB)). Идею метода можно описать следующим образом: чем реже было выбрано какое-то действие a, тем менее точной является оценка его ценности (ведь действие мало изучено). Чтобы решить описанную проблему, кажется резонным рассмотреть в качестве новой оценки ценности действия a сумму текущей оценки ценности и «близости» этой оценки к истинному значению – правую границу некоторого доверительного интервала для истинной ценности. Опишем метод формально.

Пусть агент находится в состоянии s_t, $t \geq 0$. Пусть $N_0(a) = 0$, $N_t(a)$ – количество раз, которое было выбрано действие a до шага $(t-1)$ включительно $(t \geq 1)$.

1. Если существует $a \in A(s_t)$ такое, что $N_t(a) = 0$, тогда

$$A_t = \{a \in A(s_t) : \ N_t(a) = 0\}$$

и стратегия выбора $a_t \in A_t$ такова:

$$\pi_t(a|s_t) = \begin{cases} \frac{1}{|A_t|}, & a \in A_t \\ 0, & a \notin A_t \end{cases}.$$

2. Иначе

$$A_t = \underset{a \in A(S_t)}{\operatorname{Arg\,max}} \left(q_t(a) + \delta \sqrt{\frac{\ln t}{N_t(a)}} \right), \quad \delta > 0,$$

и стратегия выбора $a_t \in A_t$ такова:

$$\pi_t(a|s_t) = \begin{cases} \frac{1}{|A_t|}, & a \in A_t \\ 0, & a \notin A_t \end{cases}.$$

Иными словами, если действие не было выбрано ни разу, то оно и выбирается в качестве следующего (если таких действий несколько, то выбор среди них осуществляется случайным образом). Если таких действий нет, то на выбор влияет, с одной стороны, ценность действия $q_t(a)$, а с другой – степень его исследованности. Параметр $\delta > 0$ в некотором смысле отвечает за вклад последней. Подробные пояснения могут быть найдены в дополнительных материалах.

Лемма 1.7.1 *Пусть $X_1, X_2, \ldots, X_n$ – выборка из генеральной совокупности ξ, причем $X_i \in [0, 1]$,*

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Тогда справедливо неравенство Хёфдинга

$$\mathsf{P}(\overline{X} - \mathsf{E}\xi \geq \varepsilon) \leq e^{-2n\varepsilon^2}.$$

Пусть $U \geq 0$. Переписав неравенство в наших обозначениях, а также пользуясь симметричностью последнего, получим

$$\mathsf{P}(\mathsf{E}\xi - q_t(a) \geq U) \leq e^{-2N_t(a)U^2}.$$

Или иначе

$$\mathsf{P}(\mathsf{E}\xi \geq q_t(a) + U) \leq e^{-2N_t(a)U^2}.$$

Итак, мы оценили сверху вероятность того, что истинная «ценность» больше, чем сумма оценки ценности и U – числа, называемого upper confidence bound (UCB). Обозначим

$$\mathsf{p} = e^{-2N_t(a)U^2},$$

откуда

$$\ln \mathsf{p} = -2N_t(a)U^2 \Longrightarrow U = \sqrt{\frac{\ln \mathsf{p}}{-2N_t(a)}}.$$

p можно выбрать, например, как функцию от t, при этом разумно, чтобы

$$\mathsf{p}(t) \xrightarrow[t \to +\infty]{} 0.$$

Тогда, положив

$$\mathsf{p} = \frac{1}{t^{2\delta^2}},$$

получим

$$U = \delta \sqrt{\frac{\ln t}{N_t(a)}},$$

что и используется в предложенном выше алгоритме.

Замечание 1.7.1 *В условиях неравенства Хёффдинга предполагается, что все элементы выборки $X_i \in [0, 1]$, поэтому для корректности применения метода рекомендуется проводить предварительную нормировку.*

На рисунке 7 представлены результаты сравнения метода UCB с различными параметрами δ и ε-жадного алгоритма. Можно заметить, что в наблю-

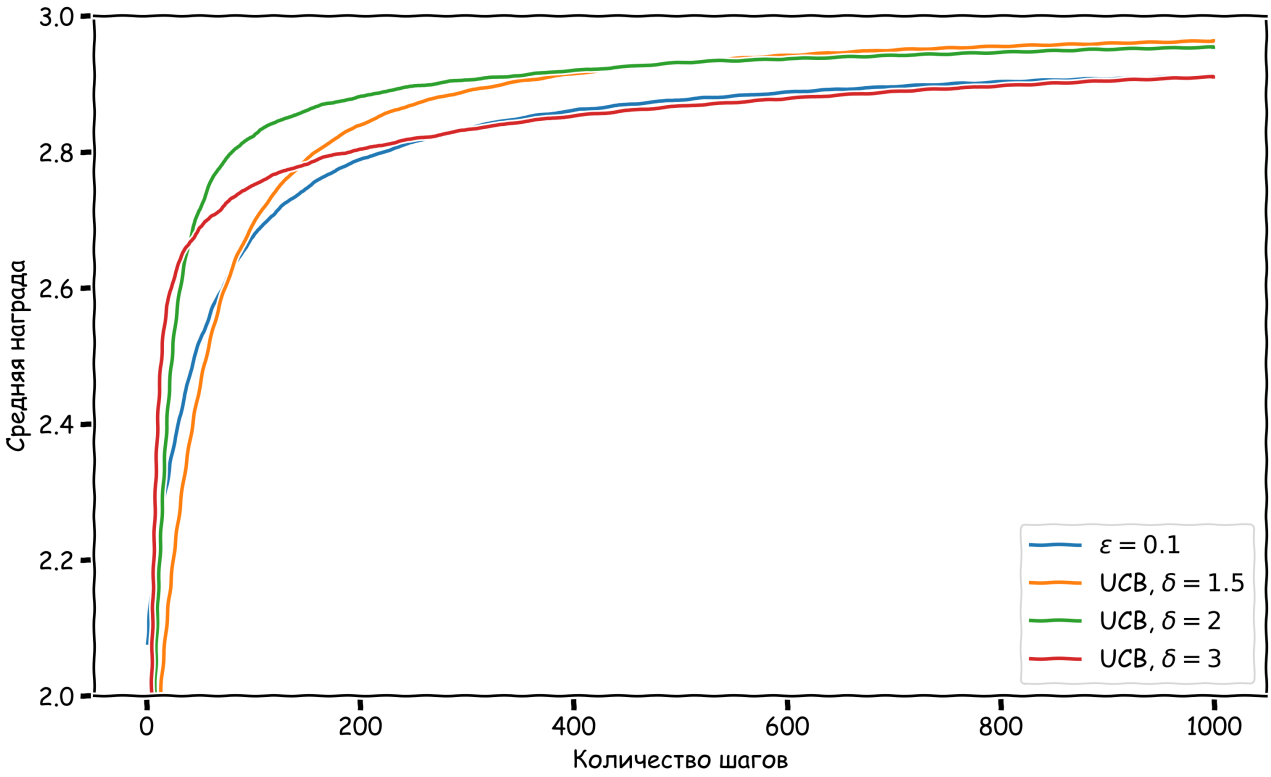


Рис. 7: Сравнение ε-жадного и UCB алгоритмов

даемых условиях чем больше δ, тем раньше алгоритм выходит на условное «плато продуктивности». В случае $\delta = 1.5$ доля исследовательских действий меньше, чем при $\delta = 2$ и $\delta = 3$, поэтому требуется больше времени на исследование, что, однако, положительно сказывается в более длительной перспективе.

Замечание 1.7.2 *Отметим также, что параметр δ опять-таки имеет смысл уменьшать со временем, для того, чтобы смещать акцент с изучения на применение и увеличивать таким образом среднюю награду.*

1.8 Оптимистичные начальные оценки

Во всех ранее рассмотренных случаях выбор того или иного действия опирался на оценку его ценности. При этом начальные оценки для всех действий предполагались равными из-за того, что агент на начальным этапе не обладает какой-либо информацией о ценностях тех или иных действий. При

использовании, например, ε-жадного алгоритма оценки ценности действий уточняются по мере выбора самих действий.

Похожая идея используется и в методе, который получил название «метод оптимистичных начальных оценок». Оптимизм, следующий из названия, заключается в том, что при инициализации каждому действию устанавливается заведомо завышенное значение ценности. Поясним сказанное на рассматриваемом нами примере.

Напомним, что значения поощрений генерировались из генеральных совокупностей, имеющих распределения $\mathsf{N}_{2,1}, \mathsf{N}_{2.5,1}, \mathsf{N}_{3,1}$. Таким образом, если в качестве начальных значений положить $q_0(a) = q_0(b) = q_0(c) = 10$ (что явно выше как 2, так и 2.5, и 3), то такие оценки можно считать даже слишком оптимистичными – они будут уменьшаться по мере обучения модели. Выбор действий будем производить при помощи жадной стратегии.

Рассмотрим несколько шагов. Оценки ценности будем заносить в соответствующую таблицу.

Шаг	$q_t(a)$	$q_t(b)$	$q_t(c)$
$t = 0$	10	10	10

Поскольку оценки ценностей всех действий одинаковы, выбирается произвольное действие, например, действие c. Пусть награда составит $r_1 = 2.8$, тогда

$$q_1(c) = q_0(c) + \frac{1}{1+1}\left(r_1 - q_0(c)\right) = 10 + \frac{(2.8 - 10)}{2} = 6.4.$$

Добавим значение в таблицу

Шаг	$q_t(a)$	$q_t(b)$	$q_t(c)$
$t = 0$	10	10	10
$t = 1$	10	10	6.4

Теперь, согласно жадному алгоритму, агент будет выбирать одно из действий a или b, так как их текущая оценка ценности выше. Пусть выбрано действие a и получена награда $r_2 = 2.2$. Тогда

$$q_2(a) = q_1(a) + \frac{1}{1+1}\left(r_2 - q_1(a)\right) = 10 + \frac{(2.2 - 10)}{2} = 6.1.$$

Занесем полученный результат в таблицу.

Шаг	$q_t(a)$	$q_t(b)$	$q_t(c)$
$t = 0$	10	10	10
$t = 1$	10	10	6.4
$t = 2$	6.1	10	6.4

Следующим, очевидно, будет выбрано действие b. Продолжая, агент будет переключаться между действиями, будучи разочарованным получаемой наградой. При этом все действия будут опробованы некоторое количество раз, что позволит построить оценки, достаточно близкие к истинным значениям.

Нетрудно заметить, что в силу «оптимизма» на первых шагах происходит изучение, а когда оценки ценности уточнены, акцент смещается на применение.

Сравнение жадного алгоритма с оптимистичными начальными оценками и ранее рассмотренных алгоритмов (1000 игр по 1000 шагов, построение при $t \geq 1$) представлено на рисунке 8.

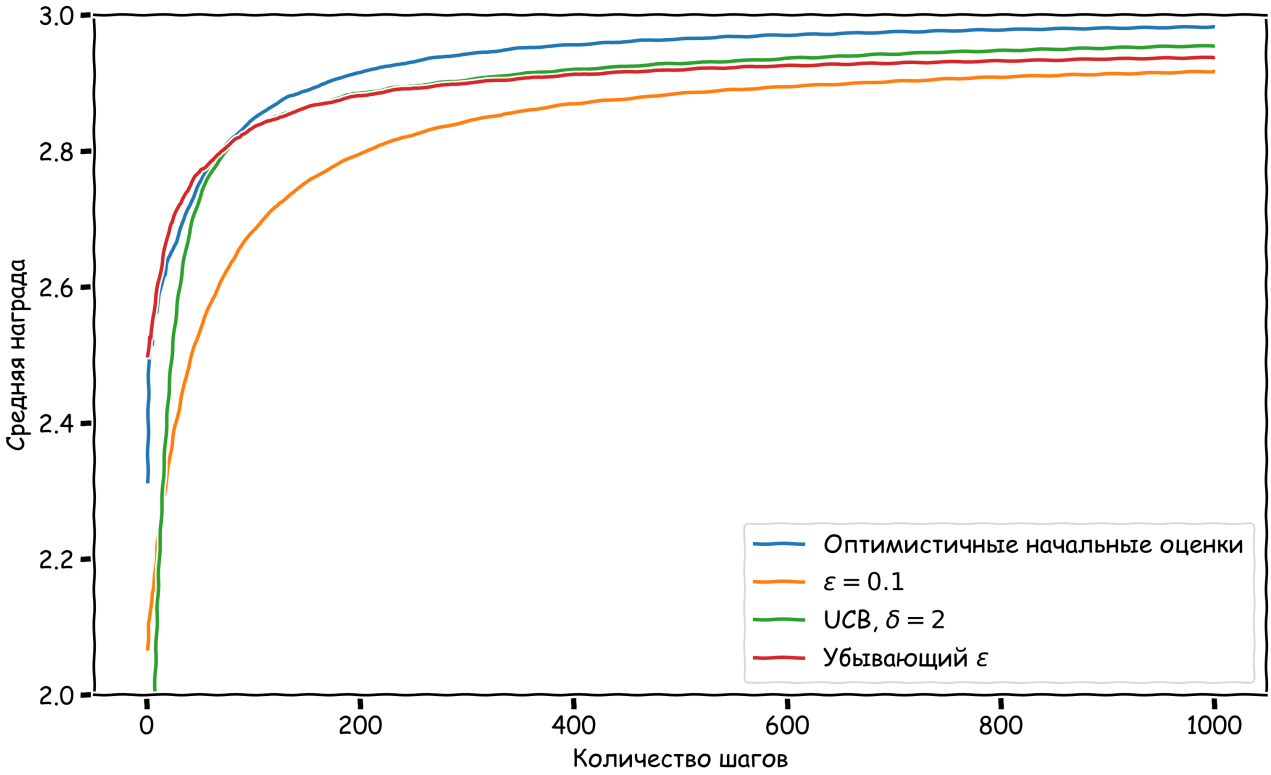


Рис. 8: Сравнение различных алгоритмов

Можно заметить, что в самом начале обучения оптимистичный алгоритм проигрывает ε-жадному с убывающим ε, так как «изучает» среду, однако по прошествии времени демонстрирует куда более высокие результаты.

2 Общая постановка задачи

2.1 Взаимосвязь агента и окружающей среды

Ранее, говоря о k-руких бандитах, мы уже описали «на пальцах» схему взаимодействия агента и среды. Строго говоря, чуть ли не самое важное в

описанном ранее – это стратегия выбора действия агентом в момент времени t. Давайте еще раз подробно обсудим весь процесс.

В каждый дискретный момент времени $t \in \{0, 1, 2, ..., n\}$, $n \in \mathbb{N} \cup \{0, \infty\}$, агент находится в некотором состоянии $s_t \in S$, где S – множество всех возможных состояний. Согласно стратегии $\pi_t(a|s_t)$, агент выбирает одно из возможных действий $a_t \in A(s_t)$, где $A(s_t)$ – множество доступных агенту действий в состоянии s_t. В случае, если на любом шаге агенту доступны все действия, будем писать $a_t \in A$, где $A = \bigcup\limits_{s \in S} A(s)$ – множество всех возможных действий. После совершения действия, агент получает награду $r_{t+1} \in \mathbb{R}$ и переходит в состояние s_{t+1}. В итоге, вся история обучения может рассматриваться как последовательность троек вида (состояние, действие, награда):

$$s_0, a_0, r_1,$$

$$s_1, a_1, r_2,$$

$$\dots$$

Замечание 2.1.1 *Отметим отдельно, что так как действия выбираются согласно стратегии (не детерминировано, а вероятностно), то как вознаграждение, так и следующее состояние агента тоже являются в некотором смысле случайными. Поэтому можно говорить, что среда генерирует как награду r_{t+1} за действие a_t, так и следующее состояние s_{t+1} агента, в зависимости от последовательности ранее выбранных действий и состояний. А значит, s_t и r_t являются случайными величинами, причем*

$$s_{t+1} \sim p(s|s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, ..., s_0, a_0),$$

$$r_{t+1} \sim p(s|s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, ..., s_0, a_0),$$

где p – какое-то вероятностное распределение.

Итого оказывается, что рассматриваемая тройка (состояние агента, стратегия выбора действия, награда за выбранное действие) на каждом последующем шаге зависит от каждого предыдущего шага. Такая постановка оказывается чрезвычайно сложной как с точки зрения теории (многомерные совместные распределения), так и с точки зрения практики: хранение огромного количества предшествующей информации. В рамках данной лекции мы упростим себе задачу и будем считать, что следующий шаг зависит только от текущего положения дел, а прошлый опыт не учитывается вовсе. Введем следующее определение.

Определение 2.1.1 *Марковским процессом принятия решений (МППР) называется четверка (S, A, R, P), где*

1. *S – множество возможных состояний агента.*

2. *A – множество доступных действий агента.*

3. *$R : S \times A \to \mathbb{R}$ – функция поощрения (или ожидаемое вознаграждение) при переходе из состояния s в состояние s' при выборе действия a.*

4. *$P : S \times A \to \Pi(S)$ – функция перехода между состояниями, $\Pi(S)$ – множество распределений вероятностей над S,*

причем вероятности последующих переходов не зависят от истории предыдущих переходов, то есть

$$s_{t+1} \sim \kappa(s_t, a_t) \in \Pi(S),$$

$$\mathsf{P}_\kappa(s_{t+1} = s' | s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, ..., s_0, a_0) = \mathsf{P}_\kappa(s_{t+1} = s' | s_t, a_t).$$

Итак, $\kappa(s_t, a_t)$ – это распределение вероятностей перехода в некоторое состояние из состояния s_t в результате совершения действия a_t, а МППР – это модель, при использовании которой у агента в каком-то смысле «отсутствует память»: вероятность перехода в состояние s' на шаге t зависит только от текущего состояния s_t и выбранного действия a_t.

Замечание 2.1.2 (!) *Прежде чем пояснить каждый пункт определения детальнее, отметим следующий важный момент. Обучение с подкреплением использует в своей основе МППР, но не описывается «вчистую» таковым. Отличие заключается в том, что агент в каждом состоянии s выбирает некоторое действие $a \in A(s)$ с вероятностью $\pi(a|s)$ (то есть действует в соответствии с некоторой стратегией). Мы будем впредь считать стратегию агента неотъемлемой частью ММПР, не обговаривая это отдельно.*

Теперь поясним каждый пункт определения детальнее. Понятия множества состояний агента и множества доступных действий нам уже известны. R – это функция, которая выдает ожидаемое вознаграждение при выборе действия $a \in A(s_t)$, находясь состоянии s_t. Грубо говоря, подавая на вход текущее состояние и выбранное действие, функция R выдает число – ожидаемое вознаграждение при таком «ходе». Функция P же выдает распределение вероятностей перехода между состояниями в случае, когда агент находится в состоянии s_t и выбирает действие a_t.

Пример 2.1.1 *На рисунке 9 приведен пример МППР.*

В рассматриваемом случае агентом является ребенок, который может находиться в одном из пяти состояний (большие кружки):

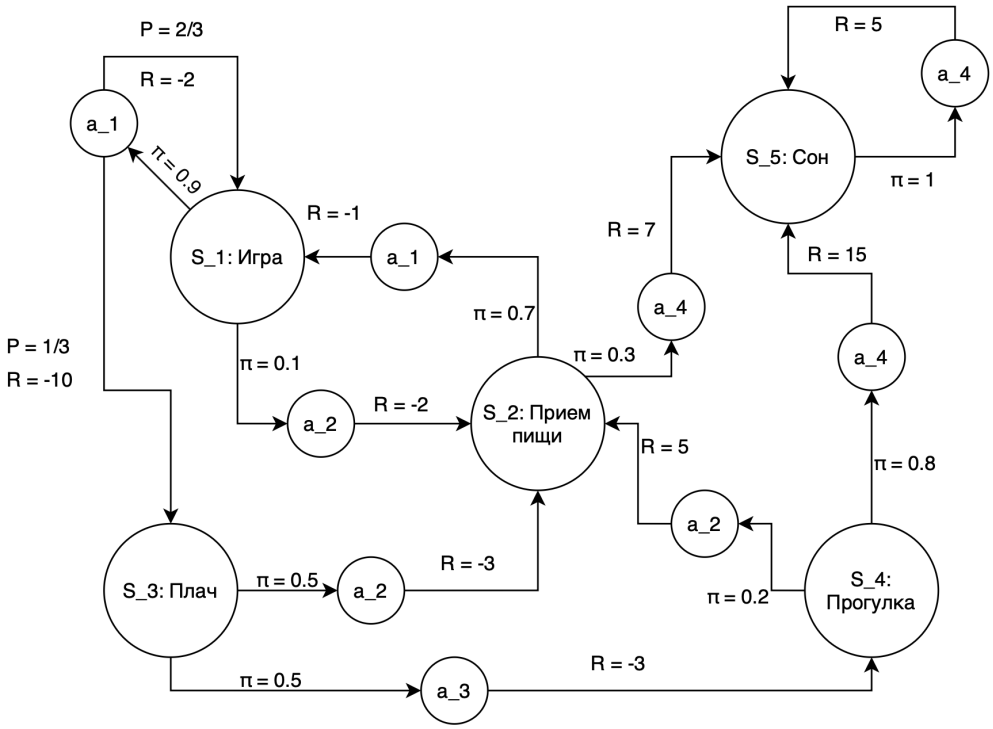


Рис. 9: МППР

s_1 : *Игра.*

s_2 : *Прием пищи.*

s_3 : *Плач.*

s_4 : *Прогулка.*

s_5 : *Сон.*

Будем считать, что всего агенту доступно лишь 4 *действия (маленькие кружки)*

a_1 : *Играть.*

a_2 : *Попросить поесть.*

a_3 : *Пойти гулять.*

a_4 : *Лечь спать.*

Итак, как же прочитать и понять информацию, заложенную на рисунке 9? Стрелки, выходящие из кругов, отвечающих состояниям, приводят нас к доступным в этом состоянии действиям (числа π *около этих стрелок диктуются стратегией агента: они показывают вероятность выбора действия, к которому идет стрелка, находясь в состоянии, из которого идет стрелка). Далее, выбрав действие, возникает распределение вероятностей над* S*: числа* P *около стрелок, выходящих из кругов, отвечающих действиям, показывают, с какой вероятностью и в какое состояние агент может перейти, а также какое вознаграждение* R *он может получить. Так, из состояния* s_1*, выбрав действие* a_1 *(с вероятностью* 0.9*), агент переходит в состояние* s_3 *с вероятностью* $1/3$*, получая вознагражс-*

дение $r = -10$, и возвращается в состояние s_1 с вероятностью $2/3$, получая при этом вознаграждение -2. Распределение вероятностей перехода из состояния s_1 при выборе действия a_1 – это и есть распределение $\kappa(s_1, a_1)$ из определения МППР. В нашем примере оно задается следующей таблицей:

	s_1	s_3
P	$2/3$	$1/3$

В то же время, выбрав в том же состоянии действие a_2 (с вероятностью 0.1), автоматически осуществляется переход (с вероятностью $P = 1$) в состояние s_2 с наградой $r = -2$, то есть распределение $\kappa(s_1, a_2)$ задается таблицей

	s_2
P	1

Ну и так далее.

В общем случае множества состояний S и действий A могут быть как конечными, так и бесконечными. Мы в нашей лекции ограничимся случаем, когда оба этих множества конечны. Такие процессы имеют даже отдельное название.

Определение 2.1.2 *Марковский процесс принятия решений, в котором множества A и S являются конечными, называется финитным.*

Отметим еще одно важное определение.

Определение 2.1.3 *Если на шаге T среда переходит в состояние, при котором взаимодействие агента и среды (игра) прекращается, то это состояние называется терминальным.*

Например, в игре «крестики-нолики» (с конечным размером поля) терминальное состояние достигается либо когда один из игроков одерживает победу, либо когда все ячейки таблицы заполнены.

Пример 2.1.2 *Немного изменим рассмотренный ранее пример – изменилось состояние s_5. Теперь это состояние оказывается терминальным – из него нет ни одного доступного действия. Родители могут отдохнуть :)*

Итак, основные определения введены. Теперь осталось понять: а что же мы будем оценивать в результате «игры»?

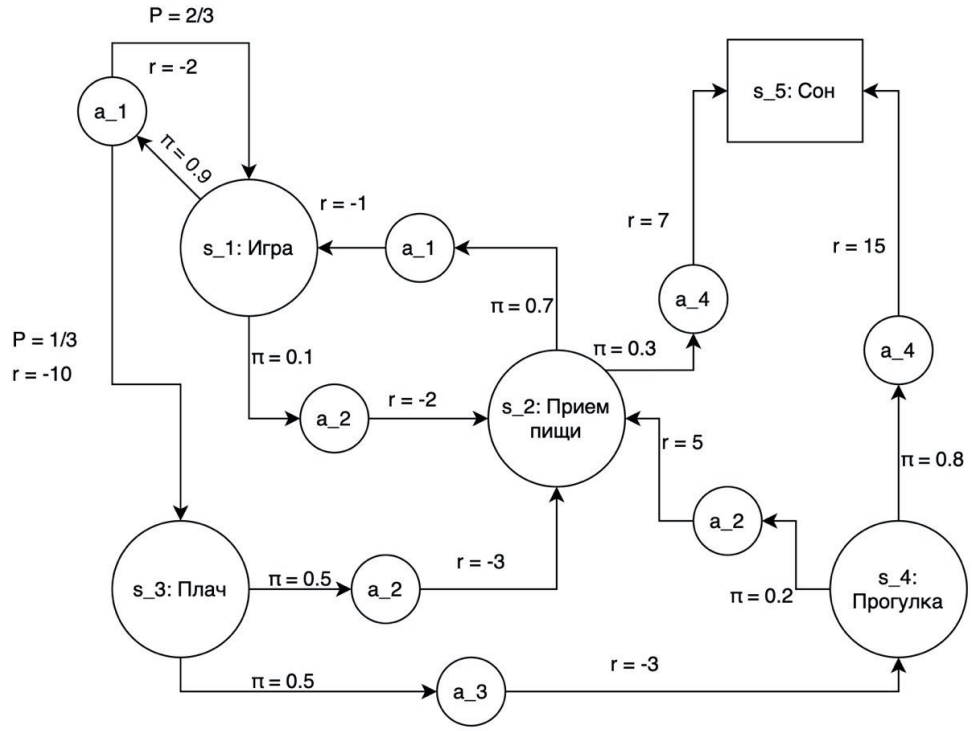


Рис. 10: МППР с терминальным состоянием

Определение 2.1.4 *Пусть мы находимся на шаге t, а взаимодействие агента и среды предполагает бесконечное число шагов. Тогда ожидаемой выгодой после шага t называется величина*

$$G_t = r_{t+1} + r_{t+2} + \ldots = \sum_{k=0}^{\infty} r_{t+k+1}$$

Если рассматривается взаимодействие, предполагающее достижение терминального состояния за конечное число шагов $t \in \{0, 1, 2, ..., T\}$, то ожидаемой выгодой называется величина

$$G_t = r_{t+1} + r_{t+2} + \ldots + r_T.$$

Наверное, введенное определение – одно из самых естественных: нет ничего проще, чем просто-напросто сложить все получаемые в дальнейшем награды (правда, пока неизвестно какие).

Замечание 2.1.3 *Отметим отдельно, что бесконечное взаимодействие агента и среды (то есть $T = +\infty$) – вовсе не выброс, а весьма частая ситуация. Кроме того, зачастую так называемое «время жизни» агента не определено. Например, выход из лабиринта может закончиться за конечное число ходов (с заранее неизвестным (!) количеством), а может не закончиться вовсе (агент заблудился).*

В то же время, максимизация суммарной награды – достаточно наивная вещь. Например, можно долго собирать «плюсики», медленно путешествуя в сторону выхода из лабиринта, хотя, казалось бы, логичнее выбраться из него как можно быстрее, то есть использовать принцип: «чем раньше прибыль, тем лучше». Тогда куда более логичным кажется использование следующего понятия.

Определение 2.1.5 *Пусть мы находимся на шаге t, а взаимодействие агента и среды предполагает бесконечное число шагов. Тогда приведенной ожидаемой выгодой называется величина*

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

где величина $\gamma \in [0, 1]$ называется коэффициентом дисконтирования.

Если рассматривается взаимодействие, предполагающее достижение терминального состояния за конечное число шагов $t \in \{0, 1, 2, ..., T\}$, то приведенной ожидаемой выгодой называется величина

$$G_t = r_{t+1} + \gamma r_{t+2} + ... + \gamma^{T-t-1} r_T.$$

Приведенную ожидаемую выгоду можно еще назвать принципом «обесценивания награды». В соответствии с описанным подходом, агент пытается выбирать действия таким образом, чтобы сумма обесценивающихся получаемых наград была наибольшей. То есть на каждом шаге агент пытается выбрать такое действие a_t, чтобы наибольшая возможная награда не обесценилась слишком сильно (последнее происходит, конечно, из-за промедления).

Можно описать это и следующим образом. Коэффициент дисконтирования γ отвечает за то, насколько важны будущие выигрыши, то есть регулирует «дальновидность» агента. Если $\gamma = 0$, то агент не обращает внимания на все последующие выигрыши и стремится максимизировать текущий. В перспективе это может занизить ожидаемую выгоду. В то же время, значения γ близкие к единице увеличивают значимость будущих вознаграждений.

Замечание 2.1.4 *Заметим, что ожидаемая выгода как в случае конечного, так и в случае бесконечного числа шагов, получается из приведенной при $\gamma = 1$.*

Справедлива следующая лемма.

Лемма 2.1.1 *Если $\gamma \in (0, 1)$ и последовательность $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$ ограничена, то приведенная выгода*

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

имеет конечное значение.

Доказательство. Так как последовательность $\{r_{t+k+1}\}_{k=0}^{\infty}$ ограничена, то $|r_{t+k+1}| \leq C$. Тогда

$$\left|\gamma^k r_{t+k+1}\right| \leq C\gamma^k.$$

Так как ряд с общим членом γ^k при $\gamma \in (0,1)$ сходится (как геометрическая прогрессия), то исходный ряд сходится абсолютно, а значит сходится. □

2.2 Функции ценности действий и состояний

Для того, чтобы агент понимал, как следует вести себя при взаимодействии со средой, то есть какой стратегии придерживаться, должен быть некоторый аппарат, описывающий состояния агента. В обучении с подкреплением применяется подход, в котором каждому состоянию присваивается некоторая численная величина – ценность этого состояния. Ценность состояния призвана охарактеризовать это состояние с точки зрения приведенной выгоды.

Определение 2.2.1 *Пусть π – стратегия агента. Функцией ценности состояния s при использовании стратегии π называется функция вида*

$$v_\pi(s) = \mathsf{E}_\pi\left(G_t | s_t = s\right) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right), \quad s \in S.$$

Функция ценности состояний определяет среднее значение приведенной выгоды, если мы начинаем в состоянии s и следуем стратегии π. Зная ценность каждого состояния, логично использовать такую стратегию, которая приведет нас к состоянию с максимальной ценностью.

Замечание 2.2.1 *Формально (в определении функции ценности) перед нами не что иное, как условное математическое ожидание – функция от состояния $s \in S$. Меняя состояние s, меняется стратегия $\pi(a|s)$, тем самым меняется и $v_\pi(s)$.*

К вопросу оценки ценности можно подойти и более педантично: оценивать приведенную выгоду, стартуя не столько из состояния s, но и выбрав действие a. Тем самым мы приходим к понятию ценности действия.

Определение 2.2.2 *Пусть π – стратегия агента. Функцией ценности действия a в состоянии s при использовании стратегии π называется функция*

$$q_\pi(s,a) = \mathsf{E}_\pi\left(G_t | s_t = s, a_t = a\right) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right).$$

Еще раз повторим, что по своей сути функция ценности действия очень похожа на функцию ценности состояния с той лишь разницей, что мы определяем приведенную выгоду, если начинаем в состоянии s, выбираем действие a и следуем стратегии π.

Вроде бы все логично, и, если бы мы знали все про среду и агента, то достаточно бы было выбирать то действие a, которое максимизирует $q(s,a)$. Конечно, это невозможно, но оказывается, что для функций ценности выполняются рекурсивные отношения.

Лемма 2.2.1 *Пусть $v_\pi(s)$ и $q_\pi(s,a)$ – функции ценности состояний и действий, соответственно. Пусть также известна вся информация о среде: известны вероятности всех переходов из состояния s в состояние s' при выполнении действия a*

$$\mathcal{P}^a_{ss'} = \mathsf{P}(s_{t+1} = s' | s_t = s, a_t = a)$$

и известны все ожидаемые премии

$$\mathcal{R}^a_{ss'} = \mathsf{E}_\pi \left(r_{t+1} | s_t = s, a_t = a, s_{t+1} = s' \right).$$

Тогда справедливы следующие соотношения

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi \left(s' \right) \right)$$

и

$$q_\pi(s,a) = \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \sum_{a' \in A(s')} \pi(a'|s') q_\pi(s',a') \right) =$$
$$= \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_{pi}(s') \right)$$

Доказательство. 1. Докажем первое равенство. По определению, используя свойство линейности математического ожидания, получим

$$v_\pi(s) = \mathsf{E}_\pi \left(G_t | s_t = s \right) = \mathsf{E}_\pi \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right) =$$

$$= \mathsf{E}_\pi \left(r_{t+1} | s_t = s \right) + \gamma \mathsf{E}_\pi \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s \right).$$

Рассмотрим пару полученных слагаемых по отдельности. Начнем с первого. Находясь в состоянии s, каждое действие выбирается с вероятностью $\pi(a|s)$,

переход в состояние s' происходит с вероятностью $\mathcal{P}^a_{ss'}$, при этом «выдается» ожидаемая премия $\mathcal{R}^a_{ss'}$. Тогда

$$\mathsf{E}_\pi\left(r_{t+1}|s_t=s\right)=\sum_{a\in A(s)}\sum_{s'\in S}\pi(a|s)\mathcal{R}^a_{ss'}\mathcal{P}^a_{ss'}.$$

Аналогично, желая заменить $s_t=s$ на $s_{t+1}=s'$, получим

$$\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_t=s\right)=\sum_{a\in A(s)}\sum_{s'\in S}\pi(a|s)\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_{t+1}=s'\right)\mathcal{P}^a_{ss'}.$$

Складывая и перегруппировывая, получим:

$$v_\pi(s)=\sum_{a\in A(s)}\pi(a|s)\sum_{s'\in S}\mathcal{P}^a_{ss'}\left(\mathcal{R}^a_{ss'}+\gamma\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_{t+1}=s'\right)\right).$$

Так как

$$v_\pi(s')=\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_{t+1}=s'\right),$$

получим

$$v_\pi(s)=\sum_{a\in A(s)}\pi(a|s)\sum_{s'\in S}\mathcal{P}^a_{ss'}\left(\mathcal{R}^a_{ss'}+\gamma v_\pi\left(s'\right)\right).$$

2. Аналогичным образом доказывается соотношение для $q_\pi(s,a)$. По определению, используя свойство линейности математического ожидания,

$$q_\pi(s,a)=\mathsf{E}_\pi\left(G_t|s_t=s,a_t=a\right)=\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t=s,a_t=a\right)=$$

$$=\mathsf{E}_\pi\left(r_{t+1}|s_t=s,a_t=a\right)+\gamma\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_t=s,a_t=a\right).$$

Так как действие a выбрано, то, во-первых,

$$\mathsf{E}_\pi\left(r_{t+1}|s_t=s,a_t=a\right)=\sum_{s'\in S}\mathcal{P}^a_{ss'}\mathcal{R}^a_{ss'},$$

а во-вторых

$$\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_t=s,a_t=a\right)=\sum_{s'\in S}\mathcal{P}^a_{ss'}\mathsf{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_{t+1}=s',a_t=a\right)=$$

$$= \sum_{s' \in S} \mathcal{P}^a_{ss'} q_\pi(s', a) = \sum_{s' \in S} \mathcal{P}^a_{ss'} \sum_{a' \in A(s')} \pi(a'|s') q_\pi(s', a').$$

В итоге,

$$q_\pi(s, a) = \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \sum_{a' \in A(s')} \pi(a'|s') q_\pi(s', a') \right) = \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v(s') \right).$$

□

Ясно, что желая получить функции ценности состояний и действий, мы приходим к системе из $|S|$ уравнений с $|S|$ неизвестными в каждом случае.

Замечание 2.2.2 *Полученные уравнения называются уравнениями Беллмана для функций ценности при использовании стратегии π. В случае функции ценности состояний уравнение Беллмана выражает зависимость ценности некоторого состояния от ценностей последующих состояний. В случае функций ценности действий – зависимость ценности пары (состояние, действие) от ценностей последующих таких пар.*

Идею (как и технику использования) уравнений можно понять из рисунка 11. Пусть нам (откуда-то) известны ценности всех состояний, а в качестве текущего состояния в данном случае рассматривается состояние s_1, тогда, используя соотношение

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi \left(s' \right) \right)$$

в случае, когда $\gamma = 0.8$, получим

$$-2.23 = 0.9 \left(\frac{1}{3} \left(-10 + 0.8 \cdot 11.45 \right) + \frac{2}{3} \left(-2 + 0.8 \cdot (-2.23) \right) \right) +$$

$$+0.1 \cdot 1 \cdot (-2 + 0.8 \cdot 6.15),$$

то есть верное равенство. Конечно, основной возникающий теперь вопрос таков: а как получить ценности состояний? Рассмотрим это на следующем примере.

2.3 Пример

Прежде мы показали как связаны ценность текущего состояния s и следующих состояния s', в которые может перейти агент. Рассмотрим на примере, как получить ценности всех состояний при помощи уравнений Беллмана, используя конкретную стратегию π. Исходные данные примера представлены

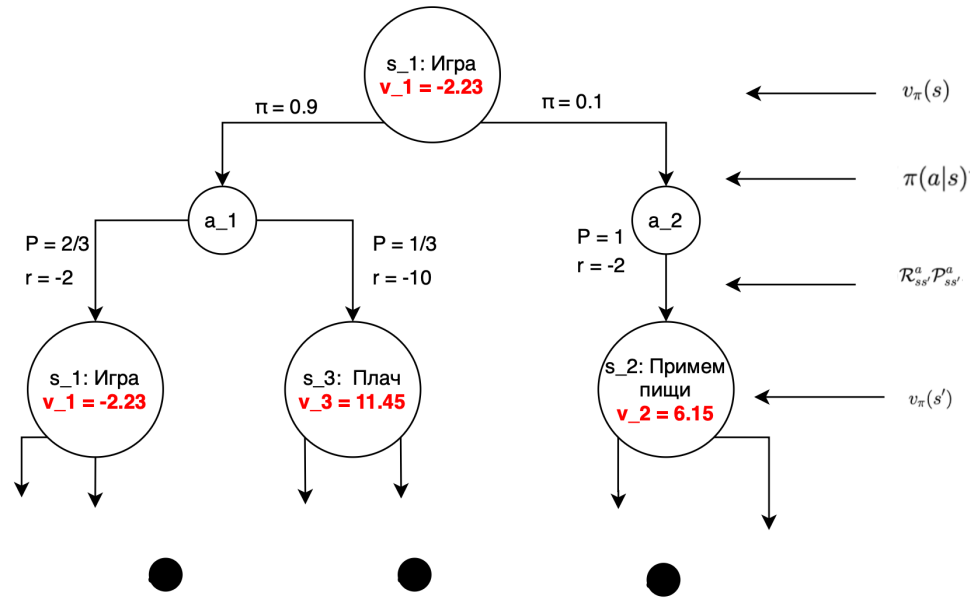


Рис. 11: МППР

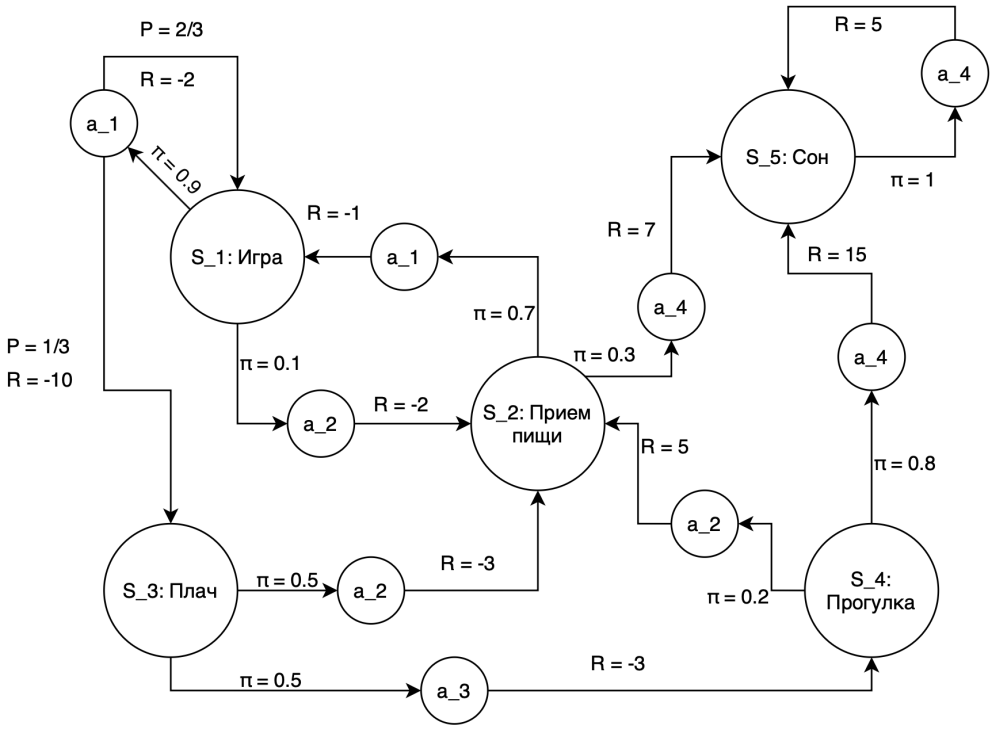


Рис. 12: Исходные данные

на уже знакомом рисунке 12. Возьмем в качестве коэффициента дисконтирования $\gamma = 0.8$ и, учитывая, что

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi \left(s' \right) \right),$$

запишем уравнение Беллмана для состояния «Игра»:

$$v_\pi(s_1) = 0.9\left(\frac{1}{3}\left(-10 + 0.8v_\pi(s_3)\right) + \frac{2}{3}\left(-2 + 0.8v_\pi(s_1)\right)\right) +$$
$$+0.1 \cdot 1 \cdot (-2 + 0.8v_\pi(s_2)).$$

Так как во всех состояниях, отличных от s_1, при совершении конкретного действия переход в следующее состояние детерминирован (осуществляется с вероятностью 1), то внутренняя сумма по s' состоит только лишь из одного слагаемого (умножаемого на 1). Учитывая это наблюдение, аналогичным образом запишем остальные уравнения:

$$v_\pi(s_2) = 0.7(-1 + 0.8v_\pi(s_1)) + 0.3(7 + 0.8v_\pi(s_5)),$$
$$v_\pi(s_3) = 0.5(-3 + 0.8v_\pi(s_2)) + 0.5(-3 + 0.8v_\pi(s_4)),$$
$$v_\pi(s_4) = 0.2(5 + 0.8v_\pi(s_2)) + 0.8(15 + 0.8v_\pi(s_5)),$$
$$v_\pi(s_5) = 5 + 0.8v_\pi(s_5).$$

Решив эту систему, получим следующие значения ценности состояний (округленные до сотых). Результаты также представлены на рисунке 13:

$$\begin{cases} v_\pi(s_1) = -2.23 \\ v_\pi(s_2) = 6.15 \\ v_\pi(s_3) = 11.45 \\ v_\pi(s_4) = 29.98 \\ v_\pi(s_5) = 25. \end{cases}$$

Замечание 2.3.1 *Очевидно, что значения ценностей состояний прямиком зависят от выбранной агентом стратегии. При изменении стратегии, вообще говоря, изменятся и ценности состояний. Например, если положить* $\pi(a_1|s_1) = 0.5, \pi(a_2|s_1) = 0.5$, *а остальные параметры оставить прежними, то получим следующие ценности состояний (округленные до сотых)*

$$\begin{cases} v_\pi(s_1) = 2.59 \\ v_\pi(s_2) = 8.85 \\ v_\pi(s_3) = 12.71 \\ v_\pi(s_4) = 30.42 \\ v_\pi(s_5) = 25. \end{cases}$$

Заметим также, что в этом случае ценности всех состояний получились не меньше, чем соответствующие ценности при использовании предыдущей стратегии. Немного позднее мы обсудим такую ситуацию подробнее.

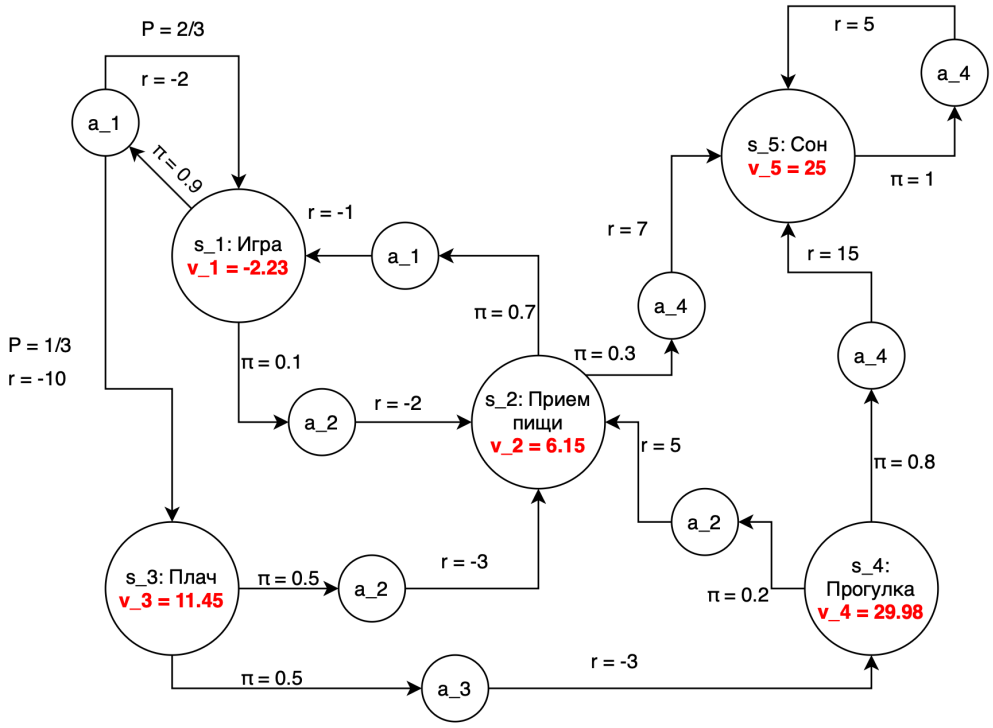


Рис. 13: Ценности состояний согласно выбранной стратегии

2.4 Оценка стратегий

В рассмотренном примере мы использовали уравнения Беллмана для нахождения ценности состояний. Для случая пяти состояний сформировать и решить систему линейных уравнений не составляет особого труда. Однако на практике количество возможных состояний агента может быть достаточно велико. Как мы уже отмечали, если количество доступных агенту состояний равно n, то возникает задача решения системы из n уравнений с n неизвестными. Стоит задуматься, а есть ли какой-то способ упростить вычисления? Оказывается есть.

Немного о матричных уравнениях и нормированных пространствах

Оказывается, решение системы уравнений Беллмана можно записать в явном виде. Перепишем систему уравнений

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi \left(s' \right) \right), \quad s \in S,$$

используя следующие матричные обозначения:

1. Пусть v_π – вектор-столбец высоты $|S|$, j-ый элемент которого равен $v_\pi(s_j)$, $j \in \{1, 2, ..., |S|\}$.

2. Пусть r_π – вектор-столбец высоты $|S|$, j-ый элемент которого равен $\mathsf{E}_\pi(r_{t+1}|s_t = s_j)$, $j \in \{1, 2, ..., |S|\}$.

3. Пусть T_π – матрица размера $|S| \times |S|$, с элементами

$$(T_\pi)_{ij} = \mathsf{P}_\pi(s_{t+1} = s_j | s_t = s_i) = \sum_{a \in A(s_i)} \pi(a|s_i)\mathcal{P}^a_{s_i s_j}.$$

В таких обозначениях система уравнений Беллмана переписывается, как

$$v_\pi = r_\pi + \gamma T_\pi v_\pi.$$

Ясно, что тогда

$$(I - \gamma T_\pi)v_\pi = r_\pi \;\Rightarrow\; v_\pi = (I - \gamma T_\pi)^{-1} r_\pi.$$

Для исследования написанного выражения, сначала напомним понятие нормированного пространства.

Определение 2.4.1 (Нормированное пространство) *Пусть X – линейное пространство. Нормой называется функция $\|\cdot\| : X \to \mathbb{R}$, удовлетворяющая следующим аксиомам:*

1. $\forall x \in X \Rightarrow \|x\| \geq 0$, *причем* $\|x\| = 0 \Leftrightarrow x = 0$.
2. $\|\lambda x\| = |\lambda|\|x\|$, $\lambda \in \mathbb{R}$, $x \in X$.
3. $\|x + y\| \leq \|x\| + \|y\|$, $x, y \in X$.

Замечание 2.4.1 *Понятно, что норма в общем случае обобщает понятие длины. Если $X = \mathbb{R}$, то можно положить $\|x\| = |x|$. Если, например, $X = \mathbb{R}^n$, то в качестве нормы $x = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$ можно рассмотреть знакомую нам длину вектора:*

$$\|x\| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}.$$

Замечание 2.4.2 *В $\mathbb{R}^n$ можно рассмотреть и другие нормы, например*

$$\|x\|_p = \sqrt[p]{|x_1|^p + |x_2|^p + ... + |x_n|^p},\ p \in \mathbb{N}.$$

Знакомая нам длина вектора – частный случай написанной нормы при $p = 2$. Мы будем пользоваться в дальнейшем нормой при $p = +\infty$:

$$\|x\|_\infty = \max_{i \in \{1,...,n\}} |x_i|.$$

Теперь перейдем к понятию нормы оператора.

Определение 2.4.2 *Пусть X – линейное нормированное пространство и $A : X \to X$ – линейный оператор. Нормой оператора A называется величина*

$$\|A\| = \sup_{x:\ \|x\|=1} \|Ax\|.$$

Итак, норма оператора в некотором смысле – это число, которое показывает максимальное значение «растяжения» вектора единичной длины под действием оператора A. Заметим, что в конечномерном пространстве норма оператора всегда конечна.

Замечание 2.4.3 *Отметим, что из определения нормы сразу следует неравенство*

$$\|Ax\| \leq \|A\|\|x\|,\ \forall x \in X.$$

Замечание 2.4.4 *Хорошо известно, что всякий линейный оператор $A : X \to X$ может быть отождествлен с его матрицей (в случае, когда в X фиксирован некоторый базис). При этом сумме и произведению операторов отвечает сумма и произведение соответствующих матриц, а обратному оператору – обратная матрица. В итоге, обратимость оператора равносильна обратимости его матрицы в некотором базисе.*

Пример 2.4.1 *Найдем норму оператора $A : \mathbb{R}^n \to \mathbb{R}^n$ с нормой $\|x\|_\infty$. Пусть в фиксированном базисе оператору сопоставлена матрица A с элементами a_{ij}, $i, j \in \{1, 2, ..., n\}$. Тогда,*

$$\|Ax\|_\infty = \max_{i\in\{1,...,n\}} \left|\sum_{j=1}^{n} a_{ij}x_j\right| \leq \max_{i\in\{1,...,n\}} \sum_{j=1}^{n} |a_{ij}|\, \|x\|_\infty.$$

Итого, при $x \neq 0$, обозначив $y = \frac{x}{\|x\|_\infty}$

$$\|Ay\| \leq \max_{i\in\{1,...,n\}} \sum_{j=1}^{n} |a_{ij}|\,,\ \|y\|_\infty = 1.$$

Если мы докажем, что написанная оценка достигается, то мы докажем, что

$$\|A\| = \max_{i\in\{1,...,n\}} \sum_{j=1}^{n} |a_{ij}|\,.$$

Пусть хотя бы один элемент матрицы не равен нулю (иначе его норма равна 0 *и равенство, очевидно, достигается). Тогда пусть*

$$i_0 = \arg\max_{i\in\{1,\dots,n\}} \sum_{j=1}^{n} |a_{ij}|$$

– какое-то i*, на котором достигается максимум. Тогда в качестве* x *возьмем вектор* $x = (\operatorname{sign} a_{i_0 1}, \operatorname{sign} a_{i_0 2}, ..., \operatorname{sign} a_{i_0 n})$*. Ясно, что* $\|x\|_\infty = 1$ *и*

$$Ax = \sum_{j=1}^{n} a_{i_0 j} \operatorname{sign} a_{i_0 j} = \sum_{j=1}^{n} |a_{i_0 j}| = \max_{i\in\{1,\dots,n\}} \sum_{j=1}^{n} |a_{ij}| .$$

Лемма 2.4.1 (Об обратимости оператора) *Для того чтобы линейный оператор* $A : X \to X$*,* X *– нормированное пространство с нормой* $\|\cdot\|$*, был обратим, необходимо и достаточно, чтобы нашлось такое* $m > 0$*, что*

$$\forall x \in X \;\Rightarrow\; \|Ax\| \geq m\|x\|.$$

Доказательство. Докажем достаточность. Написанное неравенство означает, что при $x_1 \neq 0 \;\Rightarrow Ax_1 \neq 0$. В случае линейности оператора это и означает обратимость.

Докажем необходимость. Пусть оператор обратим, тогда $y = Ax$, $x = A^{-1}y$ и

$$\|A^{-1}y\| \leq \|A^{-1}\|\|y\| \Rightarrow \|x\| \leq \|A^{-1}\|\|Ax\| \Rightarrow \|Ax\| \geq \|A^{-1}\|^{-1}\|x\|$$

и $m = \|A^{-1}\|^{-1}$. □

Лемма 2.4.2 (О существовании решения системы) *Пусть* $\gamma \in (0, 1)$*. Матрица* $(I - \gamma T_\pi)$ *обратима, а значит решение системы уравнений Беллмана существует и единственно, причем*

$$v_\pi = (I - \gamma T_\pi)^{-1} r_\pi.$$

Доказательство. Обратимость матрицы равносильна обратимости оператора, заданного в фиксированном базисе матрицей $(I - \gamma T_\pi)$. Так как $\|I\| = 1$ и $\|T_\pi\| = 1$ (так как рассматривается пространство $R^{|S|}$ с нормой $\|x\|_\infty$), то

$$\|(I - \gamma T_\pi)x\| \geq |\|Ix\| - \gamma\|T_\pi x\|| \geq (1 - \gamma)\|x\|.$$

Значит, оператор обратим, и лемма доказана. □

Введем следующее важное определение.

Определение 2.4.3 *Пусть $A : X \to X$ – линейный оператор, X – нормированное пространство. Оператор A называется сжатием, если $\exists \alpha \in (0,1)$, что*

$$\|Ax - Ay\| \leq \alpha \|x - y\|$$

По сути своей сжатие уменьшает расстояние между образами.

Определение 2.4.4 *Пусть $A : X \to X$ – линейный оператор. Тогда точка $x^* \in X$, для которой*

$$Ax^* = x^*$$

называется неподвижной точкой оператора A.

Неподвижные точки оператора – точки, которые переходят сами в себя под действием оператора.

Теорема 2.4.1 (Простейший метод сжимающих отображений) *Пусть $A : \mathbb{R}^n \to \mathbb{R}^n$ – линейный оператор, являющийся сжатием с параметром α, $\|\cdot\|$ – произвольная норма в $\mathbb{R}^n$. Тогда у оператора A существует единственная неподвижная точка x^*, причем последовательность*

$$x_{k+1} = Ax_k, \quad x_0 \in \mathbb{R}^n$$

сходится к x^* при $k \to +\infty$. Кроме того,

$$\|x^* - x_k\| \leq \frac{\alpha^k}{1-\alpha} \|x_1 - x_0\|.$$

Доказательство. Покажем, что последовательность x_k фундаментальна. Из этого будет следовать ее сходимость (вне зависимости от введенной нормы). Используя неравенство треугольника, получим

$$\|x_{k+p} - x_k\| \leq \|x_{k+p} - x_{k+p-1}\| + \|x_{k+p-1} - x_{k+p-2}\| + ... + \|x_{k+1} - x_k\|.$$

Воспользуемся определением последовательности x_k, тогда

$$\|x_{k+p} - x_k\| \leq \|Ax_{k+p-1} - Ax_{k+p-2}\| + ... + \|Ax_k - Ax_{k-1}\| \leq$$

$$\left(\alpha^{k+p-1} + ... + \alpha^k\right) \|x_1 - x_0\| = \frac{\alpha^k(1-\alpha^p)}{1-\alpha} \|x_1 - x_0\| \leq \frac{\alpha^k}{1-\alpha} \|x_1 - x_0\|,$$

что и доказывает фундаментальность. Тем самым, последовательность сходится. Пусть $p \to +\infty$, тогда

$$\|x^* - x_k\| \leq \frac{\alpha^k}{1-\alpha} \|x_1 - x_0\|.$$

Докажем единственность. Пусть x^*, y^* – две неподвижные точки. Тогда

$$\|x^* - y^*\| = \|Ax^* - Ay^*\| \leq \alpha \|x^* - y^*\|.$$

Отсюда, $\|x^* - y^*\| = 0$ и $x^* = y^*$. □

Итеративная оценка стратегий

Итак, мы готовы сформулировать теорему об итеративной оценке стратегии.

Теорема 2.4.2 (Об итеративной оценке стратегий) *Пусть $v_\pi(s)$ – функция ценности состояния, $s \in S$. Пусть так же $v_0(s) \in \mathbb{R}$ – произвольное число, если состояние s не является терминальным, и $v_0(s) = 0$ иначе. Тогда последовательность*

$$v_{k+1}(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_k \left(s' \right) \right)$$

сходится к $v_\pi(s)$ при $k \to +\infty$.

Доказательство. *Рассмотрим так называемый оператор Беллмана*

$$Bv = r_\pi + \gamma T_\pi v.$$

Покажем, что этот оператор является сжатием при рассмотрении нормы $\| \cdot \|_\infty$. Так как в этом случае $\|T_\pi\| = 1$, то

$$\|Bv - Bv'\|_\infty = \|\gamma T_\pi (v - v')\| \leq \gamma \|v - v'\|.$$

Осталось применить доказанный метод сжимающих отображений. □

Замечание 2.4.5 *Можно произвести оценку погрешности при замене истинной ценности состояния ее приближением.*

$$|v_\pi(s) - v_k(s)| \leq \frac{\gamma^k}{1 - \gamma} \max_{s \in S} |v_1(s) - v_0(s)|.$$

Получается, что для оценки ценности состояний при использовании некоторой стратегии, можно использовать следующий алгоритм:

Алгоритм 1. Итеративная оценка стратегии

Выбор оцениваемой стратегии
Инициализировать $v_0(s) = 0$ для всех $s \in S$
Инициализировать $k = 0$
Выполнять
    $\Delta \leftarrow 0$
    Для $s \in S$ **Выполнять**
        $v^* \leftarrow v_k(s)$
        $v_{k+1}(s) \leftarrow \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_k\left(s'\right)\right)$
        $\Delta \leftarrow \max\left(\Delta, |v^* - v_{k+1}(s)|\right)$
    Конец цикла
    $k \leftarrow k + 1$
Повторять пока $\Delta < \theta$ (малое положительное число)
Возвратить v_k

Применим описанный алгоритм для уже знакомого примера. Схема МППР изображена на рисунке 16.

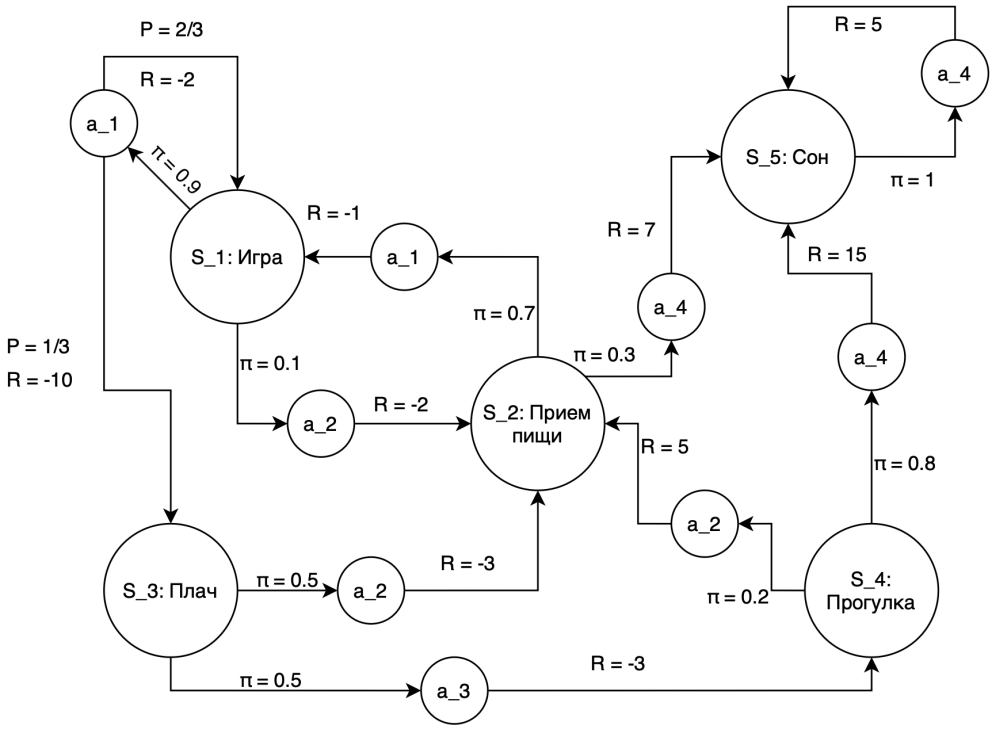


Рис. 14: Исходные данные

На первом этапе все значения ценностей состояний были приняты равными нулю. Далее, на каждом шаге значения ценностей обновлялись в соответствии с ранее полученными уравнениями Беллмана:

$$v_{k+1}(s_1) = 0.9\left(\frac{1}{3}\left(-10 + 0.8v_k(s_3)\right) + \frac{2}{3}\left(-2 + 0.8v_k(s_1)\right)\right) +$$
$$+0.1 \cdot 1 \cdot (-2 + 0.8v_k(s_2)).$$
$$v_{k+1}(s_2) = 0.7(-1 + 0.8v_k(s_1)) + 0.3(7 + 0.8v_k(s_5))$$
$$v_{k+1}(s_3) = 0.5(-3 + 0.8v_k(s_2)) + 0.5(-3 + 0.8v_k(s_4))$$
$$v_{k+1}(s_4) = 0.2(5 + 0.8v_k(s_2)) + 0.8(15 + 0.8v_k(s_5))$$
$$v_{k+1}(s_5) = 5 + 0.8v_k(s_5)$$

Результаты вычислений (округленные до сотых) представлены в таблице

Шаг	$v_k(s_1)$	$v_k(s_2)$	$v_k(s_3)$	$v_k(s_4)$	$v_k(s_5)$
0	0	0	0	0	0
1	-4.4	1.4	-3	13	5
2	-7.12	0.14	2.76	16.42	9
3	-7.14	-0.43	3.62	18.78	12.2
...	...	...	...	...	...
36	-2.24	6.14	11.45	29.98	24.99
37	-2.23	6.15	11.45	29.98	24.99
38	-2.23	6.15	11.45	29.98	24.99
39	-2.23	6.15	11.45	29.98	25

Заметим, что для получения требуемой точности (совпадение в сотых) понадобилось достаточно большое количество итераций. Погрешность составляет меньше, чем

$$13 \cdot \frac{0.8^{39}}{1 - 0.8} \approx 0.011.$$

2.5 Оптимальные стратегии

С точки зрения обучения с подкреплением, нам интересны не столько состояния, сколько стратегии, которые обеспечивают максимальную приведенную выгоду. Находя решение уравнения Беллмана для каждой стратегии, можно выбрать наилучшую или так называемую оптимальную стратегию. Что означают эти слова? Сначала введем определения.

Определение 2.5.1 *Функция*

$$v^*(s) = \max_{\pi} v_\pi(s), \quad \forall s \in S,$$

называется оптимальной функцией ценности состояний.

Определение 2.5.2 *Функция*

$$q^*(s,a) = \max_{\pi} q_\pi(s,a), \quad \forall s \in S,\ \forall a \in A(s),$$

называется оптимальной функцией ценности действий.

Итак, введенные функции показывают, какой максимальной может быть ценность состояния и (или) действия.

Замечание 2.5.1 *Покажем, что максимумы, написанные выше, в случае финитного марковского процесса принятия решений, существуют. Например, почему $v^*(s)$ – корректно определенная функция? Согласно уравнению Беллмана,*

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi\left(s'\right)\right) \leq$$

$$\leq \sum_{a \in A(s)} \pi(a|s) \max_{a \in A(s)} \left(\sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi\left(s'\right)\right) \right) =$$

$$= \max_{a \in A(s)} \left(\sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi\left(s'\right)\right) \right) = \max_{a \in A(s)} q_\pi(s,a).$$

Так как множество $A(s)$ конечно, то ясно, что стратегия, на которой достигается полученное справа выражение, такова (в случае, когда максимум достигается на одном действии):

$$\pi(a|s) = \begin{cases} 1, & a = \arg\max\limits_{a \in A(s)} q_\pi(s,a) \\ 0, & \text{иначе} \end{cases}.$$

Если максимум достигается на нескольких действиях, то можно выбирать любое из них. Итак, мы оценили нашу функция сверху и нашли стратегию, на которой эта оценка достигается. Значит, это и есть искомый максимум.

Те же самые рассуждения полезно провести в терминах стратегий.

Определение 2.5.3 *Говорят, что стратегия π не хуже, чем стратегия π', и пишут*

$$\pi \geq \pi',$$

если для всех $s \in S$

$$v_\pi(s) \geq v_{\pi'}(s).$$

Замечание 2.5.2 *Часто говорят, что введенное отношение $\geq$ индуцирует (или вводит) частичный порядок на множестве всех стратегий π. Обратите внимание, что может случиться так, что какие-то две стратегии π_1 и π_2 не сравнимы. Это значит, что найдутся $s_1, s_2 \in S$, что*

$$v_{\pi_1}(s_1) \leq v_{\pi_2}(s_1),$$

но

$$v_{\pi_1}(s_2) \geq v_{\pi_2}(s_2).$$

Пример 2.5.1 *Если в уже рассмотренном примере вместо оригинальной стратегии рисунок (15.а)) использовать стратегию π', где $\pi'(a_1|s_1) = 1, \pi'(a_2|s_1) = 0, \pi'(a_4|s_4) = 1, \pi'(a_2|s_4) = 0$ (рисунок (15.б)), то получим следующие ценности состояний:*

Стратегия	$v(s_1)$	$v(s_2)$	$v(s_3)$	$v(s_4)$	$v(s_5)$
π	-2.23	6.15	11.45	29.98	25
π'	-2.32	6.1	13.44	35	25

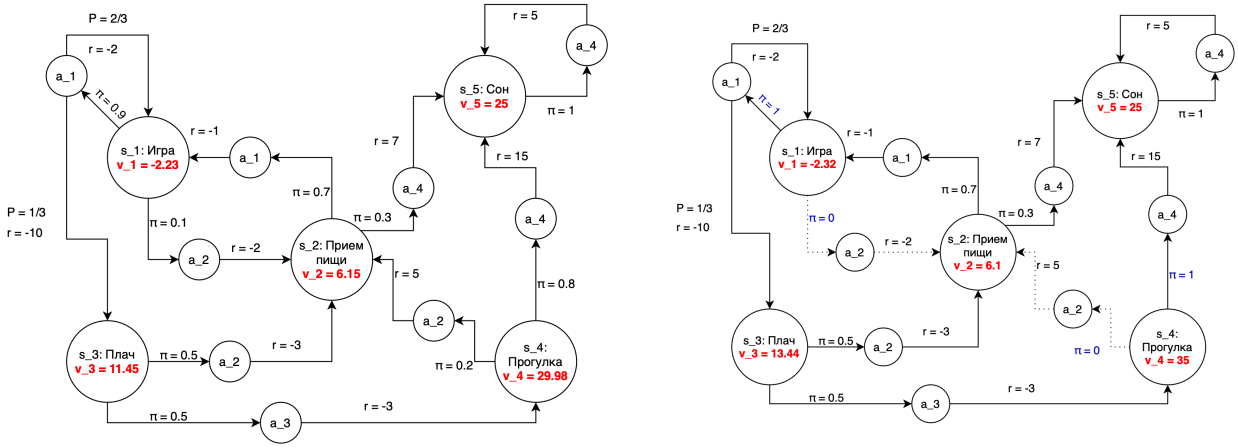


Рис. 15: Пример несравнимых стратегий.

Можно заметить, что для состояний s_1 и s_2 более высокие значения ценности получаются при использовании стратегии π, а для s_3 и s_4 ситуация обратная.

Конечно, хочется найти такую стратегию, которая не хуже всех других.

Определение 2.5.4 *Стратегия π^* называется оптимальной, если для любой стратегии π выполнено неравенство*

$$\pi^* \geq \pi.$$

Замечание 2.5.3 *Конечно, введенное определение влечет сразу много вопросов: всегда ли существует оптимальная стратегия, и будет ли оптимальная стратегия единственной? Оказывается, что в общем случае ответ на оба вопроса: «Нет». Мы вернемся к этому чуть позже.*

В то же время, имея оптимальную стратегию π^*, логично ввести в рассмотрение и функции $v_{\pi^*}(s)$ и $q_{\pi^*}(s,a)$, которые являются кандидатами на роль оптимальной функции ценности состояний и действий, соответственно. Давайте выясним их связь. Во-первых установим, что функции $v_{\pi^*}(s)$ и $q_{\pi^*}(s,a)$ в случае, когда оптимальная стратегия существует, определены корректно (на разных оптимальных стратегиях они дают одно и то же значение).

Лемма 2.5.1 *Для любых двух оптимальных стратегий π_1^* и π_2^**

$$v_{\pi_1^*} \equiv v_{\pi_2^*} \text{ и } q_{\pi_1^*} \equiv q_{\pi_2^*}.$$

Доказательство. Докажем, например, первое тождество. Так как π_1^* – оптимальная стратегия, то, согласно определению,

$$v_{\pi_1^*}(s) \geq v_\pi(s), \quad \forall s \in S,$$

в частности, когда $\pi = \pi_2^*$, откуда $v_{\pi_1^*}(s) \geq v_{\pi_2^*}(s)$. Меняя π_1^* и π_2^* местами, получаем неравенство $v_{\pi_2^*}(s) \geq v_{\pi_1^*}(s)$, что и завершает доказательство. □
Ответим теперь на вопрос: а всегда ли существует оптимальная стратегия? Оказывается, в предположении финитности марковского процесса принятия решений, да. Кроме того, справедлива следующая (достаточно ожидаемая) теорема.

Теорема 2.5.1 *Пусть марковский процесс принятия решений является финитным. Тогда*

1. *Существует оптимальная стратегия π^*.*
2. *Для любой оптимальной стратегии $v_{\pi^*} \equiv v^*$.*
3. *Для любой оптимальной стратегии $q_{\pi^*} \equiv q^*$.*

Итак, оказывается, что оптимальные функции ценности состояний и действий – это соответствующие функции ценности состояний и действий в случае, когда используется оптимальная стратегия.
Доказательство. Доказательство следует из того, что было показано ранее. На роль оптимальной стратегии π^* подходит, например, такая (если максимум достигается лишь на одном действии):

$$\pi(a|s) = \begin{cases} 1, & a = \arg\max\limits_{a \in A(s)} q_\pi(a) \\ 0, & \text{иначе} \end{cases}.$$

Если максимум достигается на нескольких действиях, то можно выбирать любое из них. Дальнейшие рассуждения очевидны. □

Аналогично тому, как были найдены уравнения Беллмана для $v_\pi(s)$ и $q_\pi(s,a)$, найдем их и для $v^*(s)$ и $q^*(s,a)$.

Лемма 2.5.2 *Пусть $v^*(s)$ и $q^*(s,a)$ – оптимальные функции ценности состояний и действий соответственно, тогда имеют место следующие соотношения.*

$$v^*(s) = \max_{a\in A(s)} \sum_{s'\in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v^*(s')\right),$$

$$q^*(s,a) = \sum_{s'\in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \max_{a'\in A(s')} q^*(s',a')\right).$$

Доказательство. Как мы выяснили, оптимальные функции ценности – это функции ценности на оптимальной стратегии π^*, то есть, если максимум достигается на одном действии, на стратегии

$$\pi(a|s) = \begin{cases} 1, & a = \arg\max\limits_{a\in A(s)} q_\pi(a) \\ 0, & \text{иначе} \end{cases}.$$

Так как в общем случае

$$v_\pi(s) = \sum_{a\in A(s)} \pi(a|s) \sum_{s'\in S} \mathcal{P}^a_{ss'}(\mathcal{P}^a_{ss'} + \gamma v_\pi(s')),$$

то для оптимальной стратегии π^*

$$v^*(s) = \max_{a\in A(s)} \sum_{s'\in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v^*(s')\right).$$

Аналогично доказывается второе соотношение. □

Замечание 2.5.4 *Важно понимать, что знание оптимальной функции ценности состояний сразу дает нам и оптимальную стратегию: нужно переходить в то состояние, чья ценность больше!*

Замечание 2.5.5 *В качестве замечания отметим, что полученные уравнения называют уравнениями оптимальности Беллмана относительно ценности состояний и действий, соответственно. В отличие от уравнений Беллмана, полученная система уже не является системой линейных алгебраических уравнений, поэтому ее решение – отдельная задача.*

Оказывается, рассмотренный ранее итерационный подход помогает и в этой ситуации.

Теорема 2.5.2 *Пусть $v^*(s)$ – оптимальная функция ценности состояния, $s \in S$. Пусть так же $v_0(s) \in \mathbb{R}$ – произвольное число, если состояние s не является терминальным, и $v_0(s) = 0$ иначе. Тогда последовательность*

$$v_{k+1}(s) = \max_{a \in A(s)} \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_k \left(s' \right) \right)$$

сходится к $v^*(s)$ при $k \to +\infty$.

Доказательство. Рассмотрим оптимальный оператор Беллмана B^*, действующий по правилу

$$B^* v = \max_{a \in A} \sum_{s' \in S} \mathcal{P}^a_{ss'} (\mathcal{R}^a_{ss'} + \gamma v(s')),$$

где

$$v = \left(v(s_1), v(s_2), \ldots, v(s_{|S|}) \right)^T.$$

Рассмотрим $B^* v$ и $B^* v'$. В силу финитности ММПР, найдутся a и a' (не обязательно различные), что

$$B^* v = \sum_{s' \in S} \mathcal{P}^a_{ss'} (R^a_{ss'} + \gamma v(s')),$$

$$B^* v' = \sum_{s' \in S} \mathcal{P}^{a'}_{ss'} (R^{a'}_{ss'} + \gamma v'(s')).$$

Кроме того, $B^* v' \geq \sum\limits_{s' \in S} \mathcal{P}^a_{ss'} (R^a_{ss'} + \gamma v'(s')$, откуда

$$B^* v - B^* v' \leq \sum_{s' \in S} \mathcal{P}^a_{ss'} (R^a_{ss'} + \gamma v(s')) - \sum_{s' \in S} \mathcal{P}^a_{ss'} (R^a_{ss'} + \gamma v'(s')) =$$

$$= \gamma \sum_{s' \in S} \mathcal{P}^a_{ss'} (v(s') - v'(s')) \leq \gamma \| v - v' \|_\infty.$$

Меняя v и v' местами, получим, что

$$|B^* v - B^* v'| \leq \gamma \| v - v' \|_\infty \ \Rightarrow \| B^* v - B^* v' \|_\infty \leq \gamma \| v - v' \|_\infty.$$

Осталось воспользоваться простейшим методом сжимающих отображений. □

Аналогично сделанному ранее, можно сформулировать алгоритм итеративного поиска оптимальных функций состояний.

Алгоритм 2. Итеративная оценка оптимальной стратегии

Выбор оцениваемой стратегии
Инициализировать $v_0(s) = 0$ для всех $s \in S$
Инициализировать $k = 0$
Выполнять
    $\Delta \leftarrow 0$
    Для $s \in S$ **Выполнять**
        $v^* \leftarrow v_k(s)$
        $v_{k+1}(s) \leftarrow \max\limits_{a \in A(s)} \sum\limits_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_k\left(s'\right)\right)$
        $\Delta \leftarrow \max\left(\Delta, |v^* - v_{k+1}(s)|\right)$
    Конец цикла
    $k \leftarrow k + 1$
Повторять пока $\Delta < \theta$ (малое положительное число)
Возвратить v_k

Замечание 2.5.6 *Аналогично тому, как было сделано ранее, можно оценить погрешность между истинным и приближенным значением оптимальной стратегии на каждом шаге алгоритма.*

Вернемся к уже знакомому примеру одного дня из жизни ребенка (рисунок 16) и выясним, какие же действия стоит предпринимать агенту, чтобы максимизировать приведенный выигрыш.

Значения ценностей состояний для случая использования оптимальной стратегии представлены на рисунке 17.

Убедимся в том, что полученные значения действительно соответствуют оптимальной стратегии, а именно проверим, что для каждого состояния выполняется равенство

$$v^*(s) = \max_{a \in A(s)} \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v^*(s')\right).$$

Для $v^*(s_1)$ имеем

$$19.6 = \max \left\{ \left(\frac{1}{3}(-10 + 0.8 \cdot 25) + \frac{2}{3}(-2 + 0.8 \cdot 19.6) \right), \right.$$

$$\left. (-2 + 0.8 \cdot 27) \right\},$$

то есть

$$19.6 = \max\{12.45, 19.6\},$$

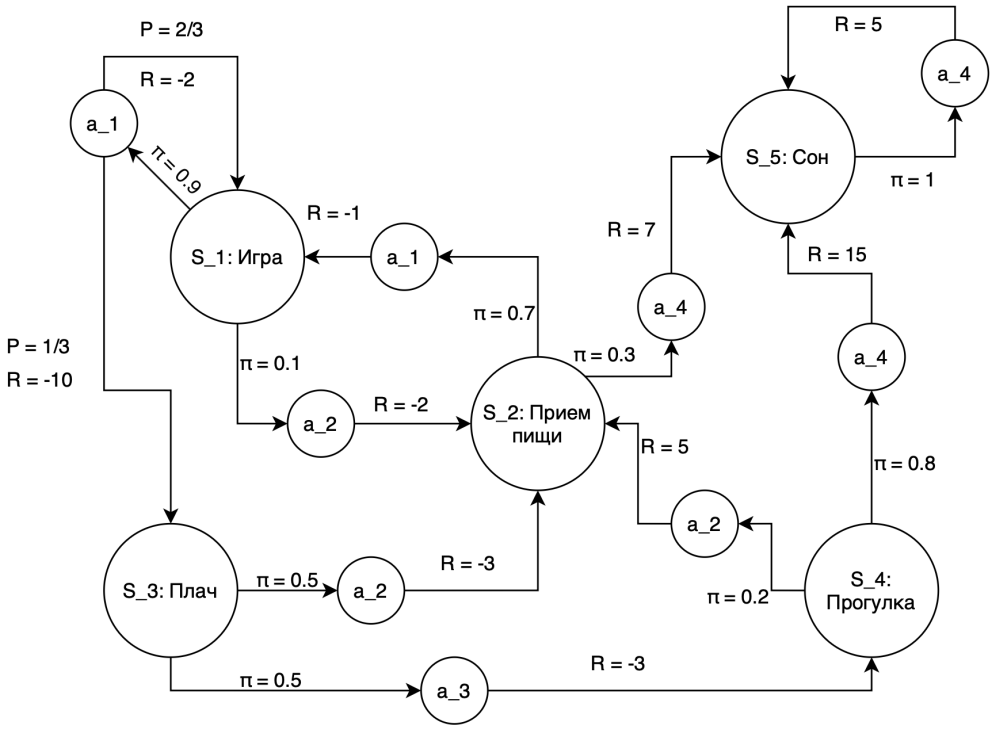


Рис. 16: Исходные данные

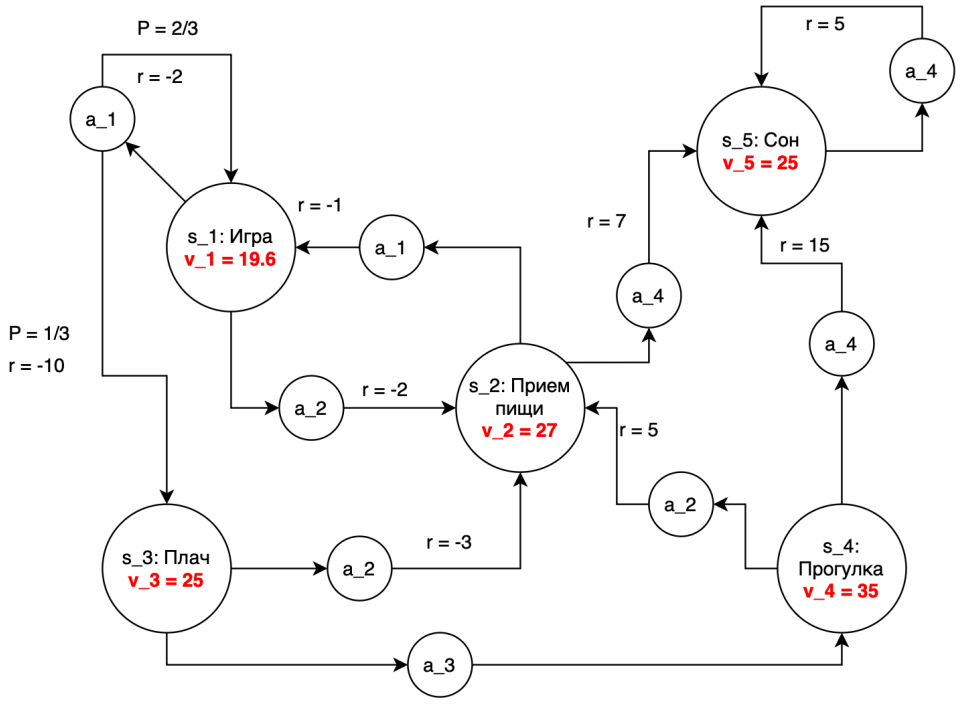


Рис. 17: Ценности состояний при использовании оптимальной стратегии

что верно. Аналогично для $v^*(s_2)$:

$$27 = \max\{(-1 + 0.8 \cdot 19.6), (7 + 0.8 \cdot 25)\} = \max\{14.68, 27\},$$

что тоже верно, и так далее.

Заметим, что зная значения оптимальной функции ценности состояний, стратегия формируется достаточно очевидным образом: нужно просто переходить в состояние с максимальным значением ценности (рисунок 18).

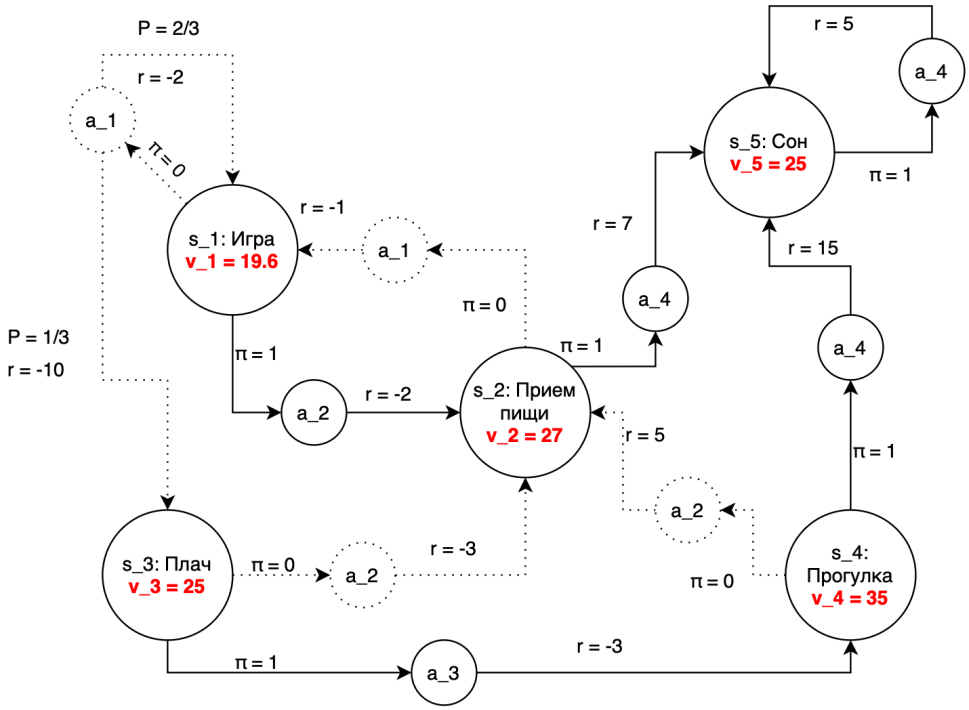


Рис. 18: Оптимальная стратегия

Смотрите, теперь наша стратегия стала полностью детерминированной: в каждый момент времени агент точно знает какое действие следует предпринять. Например, находясь в состоянии «Игра», нужно поесть и отправляться спать, а если заплакал – отправиться на прогулку, а затем уснуть.

Важным остается вопрос, а как же получить требуемые значения ценности состояний. Для получения точных значений можно решить систему уравнений оптимальности Беллмана, которая для нашего пример будет выглядеть следующим образом:

$$v^*(s_1) = \max\left\{ \left(\frac{1}{3}(-10 + 0.8v^*(s_3)) + \frac{2}{3}(-2 + 0.8v^*(s_1))\right), \right.$$

$$\left. (-2 + 0.8v^*(s_2)) \right\},$$

$$v^*(s_2) = \max\{(0.7(-1 + 0.8v^*(s_1)),\ (0.3(7 + 0.8v^*(s_5)))\}$$

$$v^*(s_3) = \max\{(0.5(-3 + 0.8v^*(s_2)),\ (0.5(-3 + 0.8v^*(s_4)))\}$$
$$v^*(s_4) = \max\{(0.2(5 + 0.8v^*(s_2)),\ (0.8(15 + 0.8v^*(s_5)))\}$$
$$v^*(s_5) = 5 + 0.8v^*(s_5)$$

Очевидно, что эта система уже не является линейной, а значит ее решение отыскать не так просто. Тут и пригодится ранее рассмотренный итерационный метод. Положим на первом шаге все значения ценностей равными нулю. Результаты вычислений (округленные до сотых) представлены в таблице

Шаг	$v_k(s_1)$	$v_k(s_2)$	$v_k(s_3)$	$v_k(s_4)$	$v_k(s_5)$
0	0	0	0	0	0
1	-2	7	-3	15	5
2	3.6	11	9	19	9
3	6.8	14.2	12.2	22.2	12.2
...	...	...	...	...	...
38	19.59	26.99	24.99	34.99	24.99
39	19.6	27	25	35	25

Отметим, что для достижения приемлемого уровня точности (совпадение в сотых) в нашем случае потребовалось порядка сорока итераций, что в данном случае вполне оправдано.

2.6 SARSA

Рассмотренные ранее методы обучения имеют достаточно серьезный недостаток. Дело в том, что они предполагают тот факт, что агент имеет представление о конфигурации среды. Иными словами, в уравнении Беллмана

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a|s) \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma v_\pi \left(s' \right) \right),$$

предполагаются известными и $\mathcal{P}^a_{ss'}$, то есть вероятности перехода из состояния s в состояние s' при совершении действия a, и все ожидаемые премии, то есть $\mathcal{R}^a_{ss'}$. В реальности это почти всегда не так. Очевидный выход – производить некоторые оценки на основе выполненных действий и полученных результатов. Идея очень похожа на идею, которая ранее была рассмотрена в разделе про многоруких бандитов. Вспомним рекуррентную формулу для нахождения ценности некоторого действия на основе экспоненциального скользящего среднего.

$$q_{t+1}(a) = q_t(a) + \alpha \left(r_{t+1} - q_t(a) \right),$$

где $\alpha \in (0, 1]$. Такой подход применим и для оценки ценности состояний. Он получил название Temporal Difference (TD). Учитывая, что

$$v_\pi(s) = \mathsf{E}_\pi\left(G_t | s_t = s\right) = \mathsf{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right) = \mathsf{E}_\pi\left(r_{t+1} + \gamma v_\pi\left(s'\right) | s_t = s\right),$$

можно получить способ оценки ценности состояния s_t:

$$v(s_t) \leftarrow v(s_t) + \alpha(r_{t+1} + \gamma v(s_{t+1}) - v(s_t))$$

В данном случае, находясь в состоянии s_t агент, следуя какой-то стратегии, выполняет действие a_t, получает награду r_{t+1} и переходит в состояние s_{t+1}. Ценность $v(s_{t+1})$ нового состояния s_{t+1} также известна, а значит можно обновить ценность состояния s_t. Иными словами, на следующем шаге обновляется значение оценки ценности для предыдущего шага.

Оценка ценности состояний хоть и важна, но не является самоцелью, так как награду приносят действия, а они выбираются в зависимости от стратегии. Поэтому при обучении нужно оптимизировать и стратегию. Для этих целей можно использовать идею TD для оценки функции действий $q(s, a)$, а именно

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} + \gamma q(s_{t+1}, a_{t+1}) - q(s_t, a_t))$$

Можно также заметить, что в процессе обучения повторяется одна и та же последовательность: состояние, действие, награда, состояние,.., то есть

$$s, a, r, s, a, r, s, a, r, \ldots,$$

поэтому такой метод и получил название $SARSA$. Зная оценку ценности каждого действия, можно использовать, например, жадную стратегию для выбора очередного действия, то есть стратегию, задаваемую следующим образом:

$$A_t = \underset{a \in A(s_t)}{\operatorname{Arg\,max}}\, q_t(a), \quad \pi(a|s_t) = \begin{cases} \frac{1}{|A_t|}, & a \in A_t \\ 0, & \text{иначе} \end{cases},$$

но все же логично уделять некоторое внимание и исследованию. Для таких целей подходит, например, ранее рассмотренная ε-жадная стратегия:

$$\pi_t(a|s_t) = \begin{cases} \dfrac{1-\varepsilon}{|A_t|} + \dfrac{\varepsilon}{|A(s_t)|}, & a \in A_t \\ \\ \dfrac{\varepsilon}{|A(s_t)|}, & a \notin A_t \end{cases},$$

Ниже приведено формальное описание алгоритма $SARSA$.

Алгоритм 3. Алгоритм SARSA

Пусть S – множество состояний, $A(s)$, $s \in S$ – множество доступных действий в состоянии s.
Инициализировать $q(s, a)$, $s \in S$, s – нетерминальное, $a \in A(s)$ произвольно
Инициализировать α и γ
Для каждой игры повторять **Выполнять**
    Инициализировать случайным образом нетерминальное состояние s_0
    Выбрать a_0 согласно стратегии $\pi_0(a|s_0)$
    $t \leftarrow 0$
    Для каждого шага игры t, пока не выполнен критерий остановки или пока s_t – не терминальное состояние, **Выполнять**
        Выполнить a_t, найти r_{t+1}, перейти в s_{t+1}
        Если s_{t+1} – терминальное, **тогда**
            $q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} - q(s_t, a_t))$
        иначе
            Выбрать a_{t+1} на основе стратегии $\pi_{t+1}(a|s_{t+1})$
            $q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} + \gamma q(s_{t+1}, a_{t+1}) - q(s_t, a_t))$
        Конец условия
        $t \leftarrow (t + 1)$
    Конец цикла
Конец цикла

Итак, согласно алгоритму, на первом этапе произвольно инициализируются ценности всех доступных действий $a \in A(s)$ для каждого нетерминального состояния $s \in S$. Далее проводится серия игр. В каждой игре начальное нетерминальное состояние агента инициализируется случайным образом. Далее, в рамках каждой игры на каждом шаге t агент, находясь в состоянии s_t, выбирает некоторое действие a_t согласно выбранной стратегии, переходит в следующее состояние s_{t+1} и получает вознаграждение r_{t+1}. Если состояние s_{t+1} – не терминальное, то согласно стратегии выбирается действие a_{t+1}, а значение $q(s_t, a_t)$ обновляется согласно формуле.

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} + \gamma q(s_{t+1}, a_{t+1}) - q(s_t, a_t)).$$

Далее происходит переход на новую итерацию игры. Если состояние s_{t+1} является терминальным, то $A(s_{t+1}) = \varnothing$ и

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} - q(s_t, a_t)).$$

Игра завершается в случае, если совершен переход в терминальное состояние или выполнен критерий остановки. Примерами выполнения критерия остановки могут служить следующие ситуации:

- достигнут заданный уровень суммы полученных наград;
- достигнут заданный уровень среднего значения полученных наград;
- достигнут заданный уровень числа потраченных шагов;
- и тому подобное.

Рассмотрим применение алгоритма $SARSA$ для уже знакомого, хотя и слегка модифицированного примера: будем считать состояние «Сон» терминальным (при переходе в него игра заканчивается). Схема представлена на рисунке 19.

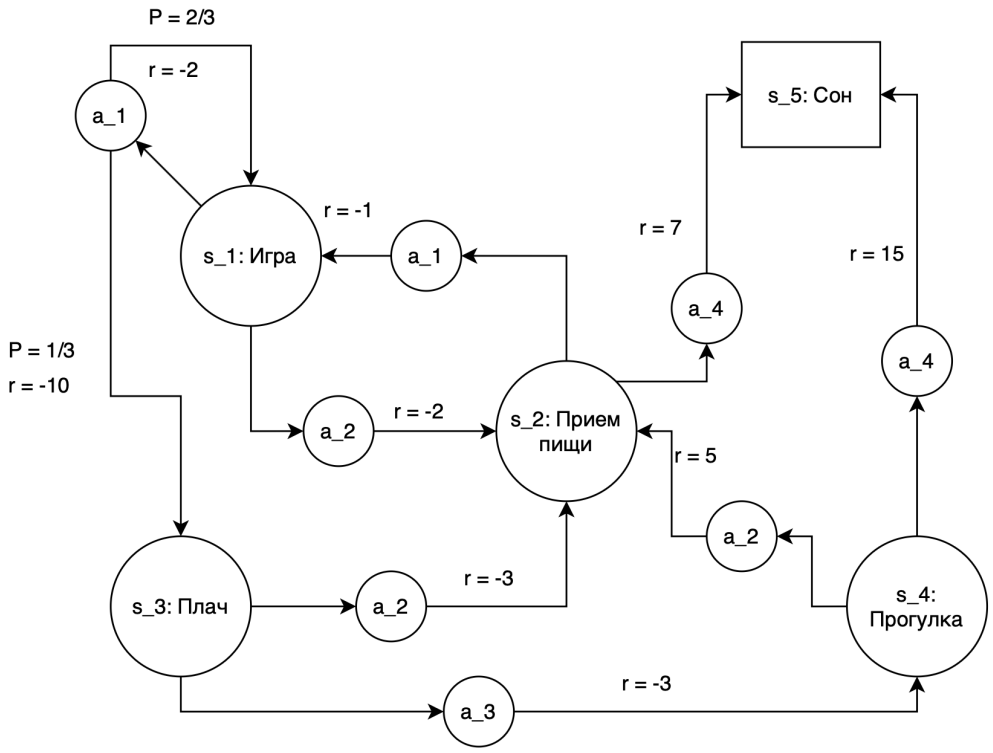


Рис. 19: Описание среды

Положим $\gamma = 0.8$, $\alpha = 0.1$. В качестве стратегии будем использовать ε-жадную стратегию со значением $\varepsilon = 0.1$. Отметим также, что теперь нам неизвестны ни вероятности переходов из состояния s в состояние s' в результате выполнения действия a, ни то, какие награды мы получим в этом случае. Все значения придется оценивать по ходу игры. Для удобства сформируем таблицу ценностей действий в зависимости от состояний и произведем начальную инициализацию (установим все возможные значения ценностей равными нулю). Обращаем внимание, что символ $\times$ означает, что действие не может быть выбрано в текущем состоянии.

	a_1	a_2	a_3	a_4
s_1	0	0	×	×
s_2	0	×	×	0
s_3	×	0	0	×
s_4	×	0	×	0
s_5	×	×	×	×

Предположим, что мы начинаем в состоянии s_1. В нем нам доступны действия a_1 и a_2, причем на данный момент $q(s_1, a_1) = q(s_1, a_2) = 0$. Согласно нашей стратегии, в данном случае произвольным образом выбирается любое из доступных действий. Пусть выбрано действие a_2. Тогда агент переходит в состояние s_2 и получает награду $r = -2$. Далее, находясь в состоянии s_2, нужно выбрать следующее действие. Нам доступны a_1 и a_4. Так как ценности одинаковые (равны нулю), выбираем любое, например a_4. Теперь можно обновить $q(s_1, a_2)$.

$$q(s_1, a_2) \leftarrow q(s_1, a_2) + \alpha(r + \gamma q(s_2, a_4) - q(s_1, a_2)) =$$
$$= 0 + 0.1 \cdot (-2 + 0.8 \cdot 0 - 0) = -0.2.$$

Занесем полученное значение в таблицу.

	a_1	a_2	a_3	a_4
s_1	0	-0.2	×	×
s_2	0	×	×	0
s_3	×	0	0	×
s_4	×	0	×	0
s_5	×	×	×	×

Продолжим выполнение алгоритма.
Текущее состояние: s_2
Выбранное действие: a_4
Следующее состояние: s_5
Награда: 7
Отметим, что так как s_5 – терминальное состояние, то игра заканчивается. Тогда

$$q(s_2, a_4) \leftarrow 0 + 0.1 \cdot (7 - 0) = 0.7.$$

	a_1	a_2	a_3	a_4
s_1	0	-0.2	×	×
s_2	0	×	×	0.7
s_3	×	0	0	×
s_4	×	0	×	0
s_5	×	×	×	×

Произведя еще несколько игр, получим следующую таблицу.

	a_1	a_2	a_3	a_4
s_1	-0.32	0.88	$\times$	$\times$
s_2	-0.02	$\times$	$\times$	6.5
s_3	$\times$	-0.09	-0.19	$\times$
s_4	$\times$	0.99	$\times$	6.14
s_5	$\times$	$\times$	$\times$	$\times$

Итак, для оценки выполненного действия мы регулярно как бы заглядываем в будущее и заранее выбираем следующее действие, что, в конечном счете, сказывается на оценке текущего действия.

Замечание 2.6.1 *Заметим, что на основе последней таблицы можно сформировать оптимальную стратегию (если выбирать действия жадно), которая совпадает с ранее найденной теоретической (рисунок 20). Находясь в состоянии s_1, надо выбирать действие a_2, так как*

$$q(s_1, a_2) = 0.88 > q(s_1, a_1) = -0.32.$$

Находясь в состоянии s_2 – выбирать действие a_4, так как

$$q(s_2, a_4) = 6.5 > q(s_2, a_1) = -0.02$$

и так далее.

Замечание 2.6.2 *В нашем примере получаемые в процессе игры награды имеют одни и те же значения вне зависимости от s, s' и a. В более общем случае это может быть не так.*

2.7 Q-обучение

Рассмотрим еще один алгоритм обучения на основе TD. Ранее нами были получены уравнения оптимальности Беллмана, в частности уравнение для оптимальной функции ценности действий

$$q^*(s, a) = \sum_{s' \in S} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \max_{a' \in A(s')} q^*(s', a') \right).$$

Идея метода Q-обучения заключается в аппроксимации оптимальной функции ценности следующим образом

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \left(r_{t+1} + \gamma \max_{a \in A(s_{t+1})} q(s_{t+1}, a) - q(s_t, a_t) \right).$$

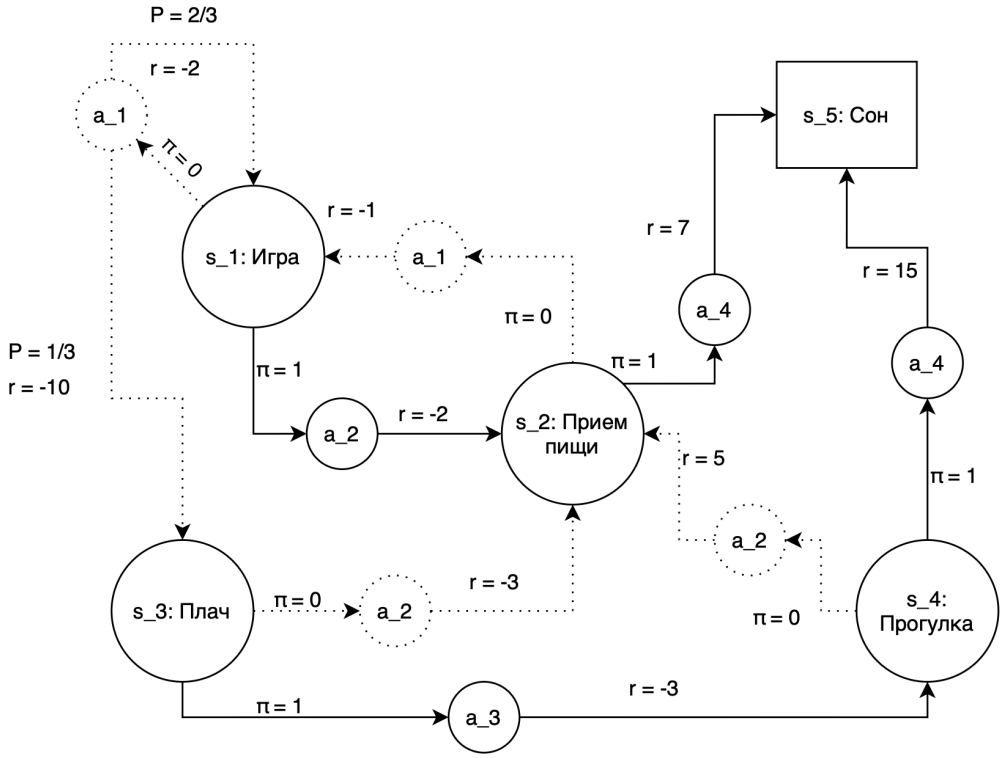


Рис. 20: Оптимальная стратегия

Стоит на секунду задуматься: а в чем отличие $SARSA$ и Q-обучения? Дело в том, что для обновления $q(s_t, a_t)$ в $SARSA$ используется значение $q(s_{t+1}, a_{t+1})$, то есть ценность действия a_{t+1}, выбранного в состоянии s_{t+1} согласно некоторой установленной стратегии $\pi_{t+1}(a|s_{t+1})$. В Q-обучении используется оптимальная стратегия, так как выбирается то действие, для которого достигается $\max\limits_{a \in A(s_{t+1})} q(s_{t+1}, a)$. В итоге при обновлении значения функции ценности действия мы опираемся не на выбранную стратегию (возможно, не оптимальную), а на оптимальную, но на следующем шаге (!). Если текущая стратегия оптимальна, то алгоритмы работают одинаково. Запишем алгоритм Q-обучения формально.

Алгоритм 4. Алгоритм Q-обучения

1: Пусть S – множество состояний, $A(s)$, $s \in S$ – множество доступных действий в состоянии s.
2: Инициализировать $q(s, a)$, $s \in S$, s – нетерминальное, $a \in A(s)$ произвольно
3: Инициализировать α и γ
4: **Для** каждой игры повторять **Выполнять**
5: Инициализировать случайным образом нетерминальное состояние s_0
6: $t \leftarrow 0$
7: **Для** каждого шага игры t, пока не выполнен критерий остановки или пока s_t – не терминальное состояние, **Выполнять**
8: Выбрать a_t согласно стратегии $\pi_t(a|s_t)$
9: Выполнить a_t, найти r_{t+1}, перейти в s_{t+1}
10: **Если** s_{t+1} – терминальное, **тогда**
11: $q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} - q(s_t, a_t))$
12: **иначе**
13: $q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max\limits_{a \in A(s_{t+1})} q(s_{t+1}, a) - q(s_t, a_t))$
14: **Конец условия**
15: $t \leftarrow (t + 1)$
16: **Конец цикла**
17: **Конец цикла**

Во-первых, как мы уже отмечали ранее, разница с $SARSA$ заключается в способе оценки ценности действия. Во-вторых отличается выбор действия: в $SARSA$ на текущей итерации находилось как текущее так и следующее действие, в то время как в Q-обучении происходит выбор только текущего действия. Если состояние s_{t+1} – не терминальное, то $q(s_t, a_t)$ обновляется согласно формуле

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a \in A(s_{t+1})} q(s_{t+1}, a) - q(s_t, a_t)).$$

Далее происходит переход на новую итерацию игры. Если состояние s_{t+1} является терминальным, то $A(s_{t+1}) = \varnothing$ и

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha(r_{t+1} - q(s_t, a_t)).$$

Игра завершается в случае, если выполнен критерий остановки, или совершен переход в терминальное состояние.

Поясним отличия алгоритма Q-обучения и алгоритма $SARSA$ на все том же примере. Пусть на каком-то этапе обучения агент окажется в состоянии s_1, выберет действие a_2, а таблица оценок будет иметь вид

	a_1	a_2	a_3	a_4
s_1	-0.32	0.88	$\times$	$\times$
s_2	-0.02	$\times$	$\times$	6.5
s_3	$\times$	-0.09	-0.19	$\times$
s_4	$\times$	0.99	$\times$	6.14
s_5	$\times$	$\times$	$\times$	$\times$

Тогда уточнение оценки $q(s_1, a_2)$ вне зависимости от используемой стратегии будет произведено следующим образом:

$$q(s_1, a_2) \leftarrow 0.88 + 0.1 \cdot (-2 + 0.8 \cdot 6.5 - 0.88) \approx 1.12,$$

так как

$$q(s_2, a_4) = \max(q(s_2, a_1), q(s_2, a_4)).$$

В алгоритме $SARSA$ это не всегда так. Например, если бы в результате использования ε-жадной стратегии в состоянии s_2 было выбрано действие не a_4, а a_1, то для обновления ценности $q(s_1, a_2)$ использовалось бы значение $q(s_2, a_1)$. В случае Q-обучения стратегия (например ε-жадная) используется только при выборе текущего действия и не участвует в обновлении оценки ценности.

2.8 Пример

Рассмотрим примеры использования рассмотренных алгоритмов. Симулируем игру, в которой агенту нужно найти выход из лабиринта, пример возможного решения представлен на рисунке 21.

В процессе игры агент помещается в точку старта, ему доступны перемещения в четырех направлениях (вверх, вправо, вниз и влево). Цель агента – достичь точки выхода из лабиринта. В результате каждого действия агент получает награду. Награды суммируются в процессе игры. Каждый шаг сопровождается наградой -0.05. Если агент переместился в уже посещенную ячейку, награда составляет -0.25. Если в результате действия агент упирается в стену, награда составляет -0.75. Выход из лабиринта сопровождается наградой $+10$. В рамках одной игры терминальное состояние достигается всегда: либо игрок находит выход и тогда игра завершается победой, либо сумма полученных наград окажется меньше заданного порогового значения. В описываемом случае пороговое значение – это

$$-\frac{1}{2} \cdot (\text{Размер лабиринта}) = -32.$$

Для обучения использовался ε-жадный алгоритм со значением $\varepsilon = 0.1$. Коэффициент дисконтирования γ равен 0.9, коэффициент $\alpha = 0.1$.

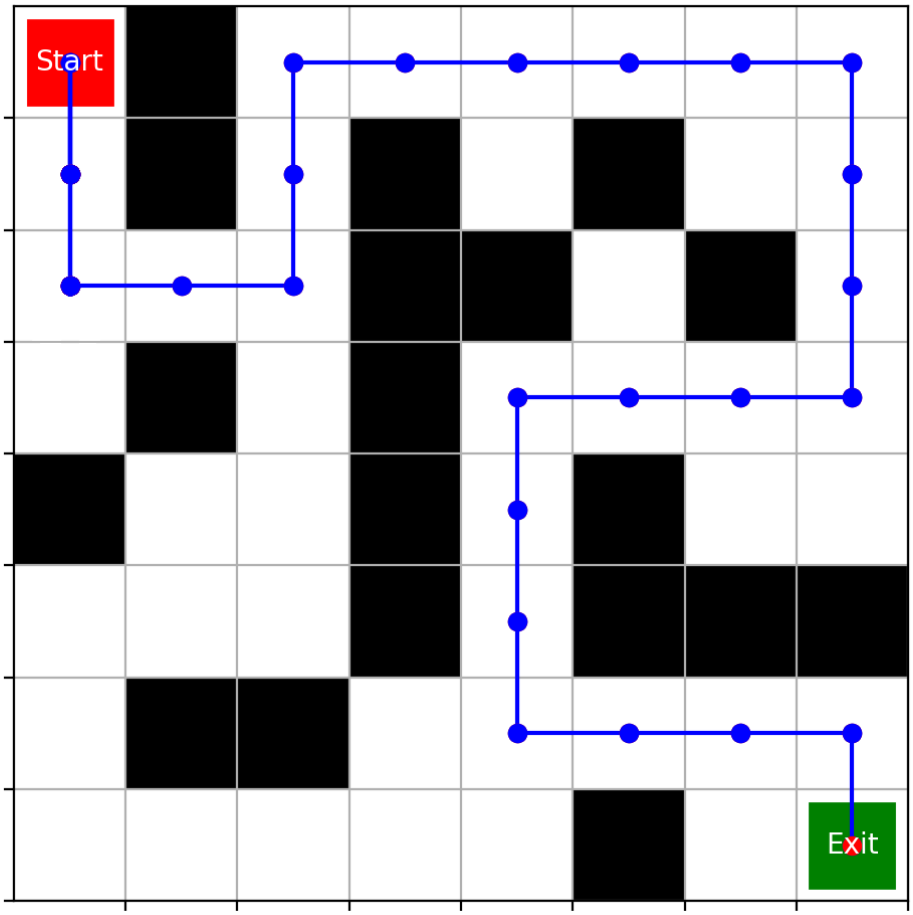
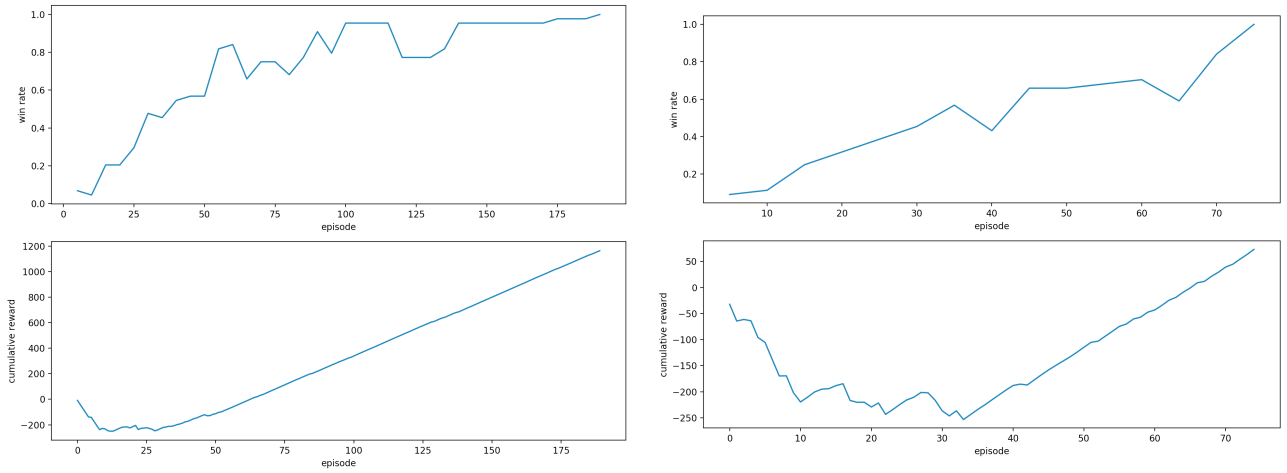


Рис. 21: Пример выхода из лабиринта.

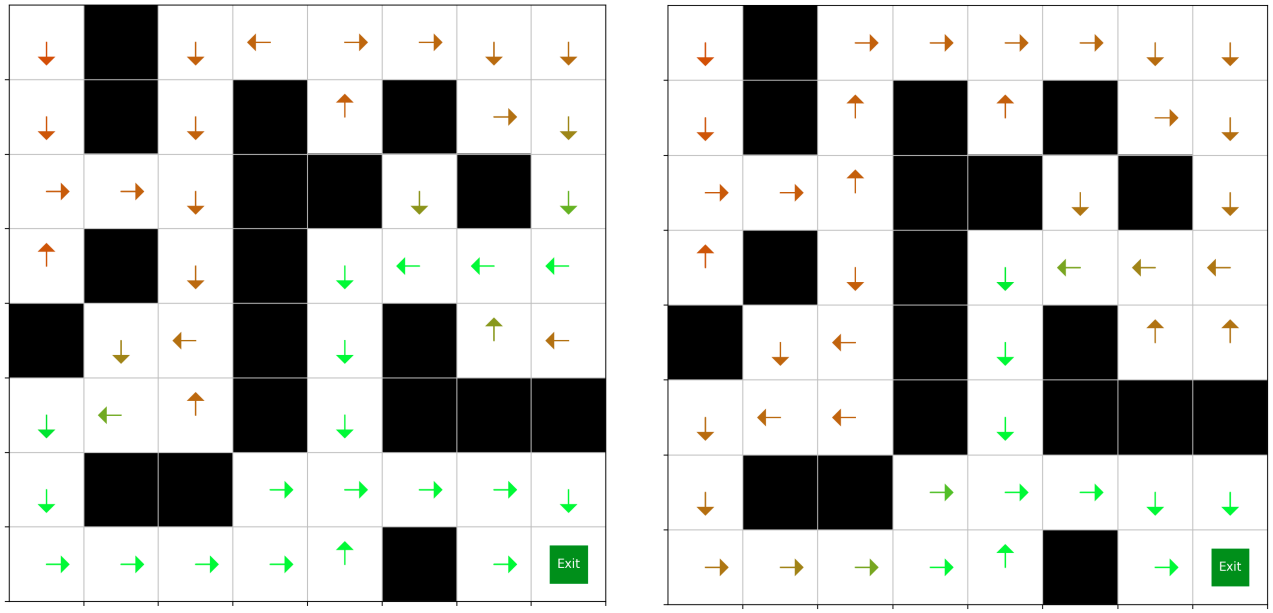
Агенту доступно 44 начальных состояния (белые клетки за исключением клетки выхода из лабиринта). Эпизодом будем называть серию из 44 игр, где каждое состояние (клетка) используется ровно один раз в качестве начального. Обучение заканчивается, если агент умеет находить путь к выходу из любого из 44 доступных состояний (не обязательно оптимальный). Иными словами, оказавшись в любом состоянии, агент точно знает какие действия нужно совершать, чтобы гарантированно выйти из лабиринта. Алгоритму Q-обучения потребовалось всего 76 эпизодов по 44 игры в каждом против 188 эпизодов у $SARSA$. Кумулятивной наградой будем называть сумму полученных наград. Соответствующие графики представлены на рисунках 22.а) и 22.б) для $SARSA$ и Q-обучения, соответственно. Для «выхода в плюс» с точки зрения кумулятивной награды каждому алгоритму понадобилось порядка 60 эпизодов. Кроме того, на рисунках представлены графики доли выигранных игр для каждого эпизода. Под долей выигранных игр понимается отношение количества побед в рамках эпизода к сорока четырем. Отметим, что алгоритм Q-обучения в данном случае продемонстрировал более плавное увеличение доли выигранных игр.



а) б)

Рис. 22: Сравнение алгоритмов обучения.

На основе полученных значений ценностей действий для всех состояний можно составить оптимальную стратегию. Полученные стратегии для алгоритмов $SARSA$ и Q-обучения представлены на рисунках 23.а) и 23.б), соответственно.



а) б)

Рис. 23: Стратегии алгоритмов $SARSA$ и Q-обучения.

Можно заметить некоторые интересные различия, связанные с использованием ε-жадных стратегий. Посмотрите на состояние $(3, 3)$ (клетка, отвечающая пересечению третьей строки и третьего столбца). Если начинать в этом состоянии, то стратегия, полученная согласно методу $SARSA$, предлагает следовать вниз, и тогда выход будет найден за 17 шагов. Стратегия,

полученная на основе Q-обучения, наоборот говорит идти вверх, но тогда выход будет найден через 21 шаг. Для этого состояния $SARSA$ оказался более эффективным. С другой стороны, у $SARSA$ можно заметить некоторую странность в ячейке $(6, 3)$ (пересечение 6 строки и 3 столбца). Различия, например, как в ячейке $(5, 8)$, роли не играют.

2.9 Резюме

В заключение отметим, что обучение с подкреплением – это, возможно, одно из самых интересных и многообещающих направлений машинного обучения. Такой тип обучения сочетает в себе как обучение с учителем, так и обучение без учителя. С одной стороны, человеческого вмешательства не требуется, агент обучается самостоятельно. С другой стороны, учителем в некотором смысле выступает среда. Еще одним существенным плюсом является то, что обучение можно производить в динамично меняющейся среде. Отметим, что в этой лекции мы рассмотрели лишь небольшую часть методов обучения с подкреплением, которых на самом деле достаточно много. При этом новые решения и идеи в этой области публикуются с завидной регулярностью.

2.10 Заключение

Дорогие друзья! На этом наш курс машинного обучения заканчивается. Надеемся, что полученные вами знания помогут вам как в решении профессиональных, так и бытовых задач. Кроме того, мы будем безгранично рады, если наш курс вдохновит кого-то из вас на дальнейшее изучение различных направлений искусственного интеллекта. Удачи и до новых встреч!