КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2023 Т. 15 № 2 С. 329–353
DOI: 10.20537/2076-7633-2023-15-2-329-353

Ки&М

МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ

УДК: 519.8

Обзор выпуклой оптимизации марковских процессов принятия решений

В. Д. Руденко[1,2,a], Н. Е. Юдин[1,3,b], А. А. Васин[4,c]

[1]Московский физико-технический институт (национальный исследовательский университет),
Россия, 141701, Московская обл., г. Долгопрудный, Институтский пер., 9
[2]Национальный исследовательский университет «Высшая школа экономики»,
Россия, 109028, г. Москва, Покровский бульвар, д. 11
[3]Федеральный исследовательский центр «Информатика и управление» Российской академии наук,
Россия, 119333, г. Москва, ул. Вавилова, д. 44, корп. 2
[4]Московский государственный университет имени М. В. Ломоносова,
Россия, 119991, г. Москва, Ленинские горы, д. 1, стр. 52

E-mail: [a] Rudenko.VD@phystech.edu, [b] Iudin.NE@phystech.edu, [c] Vasin@cs.msu.su

Получено 19.02.2023.
Принято к публикации 23.02.2023.

В данной статье проведен обзор как исторических достижений, так и современных результатов в области марковских процессов принятия решений (*Markov Decision Process*, MDP) и выпуклой оптимизации. Данный обзор является первой попыткой освещения на русском языке области обучения с подкреплением в контексте выпуклой оптимизации. Рассматриваются фундаментальное уравнение Беллмана и построенные на его основе критерии оптимальности политики — стратегии, принимающие решение по известному состоянию среды на данный момент. Также рассмотрены основные итеративные алгоритмы оптимизации политики, построенные на решении уравнений Беллмана. Важным разделом данной статьи стало рассмотрение альтернативы к подходу Q-обучения — метода прямой максимизации средней награды агента для избранной стратегии от взаимодействия со средой. Таким образом, решение данной задачи выпуклой оптимизации представимо в виде задачи линейного программирования. В работе демонстрируется, как аппарат выпуклой оптимизации применяется для решения задачи обучения с подкреплением (*Reinforcement Learning*, RL). В частности, показано, как понятие сильной двойственности позволяет естественно модифицировать постановку задачи RL, показывая эквивалентность между максимизацией награды агента и поиском его оптимальной стратегии. В работе также рассматривается вопрос сложности оптимизации MDP относительно количества троек «состояние–действие–награда», получаемых в результате взаимодействия со средой. Представлены оптимальные границы сложности решения MDP в случае эргодического процесса с бесконечным горизонтом, а также в случае нестационарного процесса с конечным горизонтом, который можно перезапускать несколько раз подряд или сразу запускать параллельно в нескольких потоках. Также в обзоре рассмотрены последние результаты по уменьшению зазора нижней и верхней оценки сложности оптимизации MDP с усредненным вознаграждением (*Averaged MDP*, AMDP). В заключение рассматриваются веществennозначная параметризация политики агента и класс градиентных методов оптимизации через максимизацию Q-функции ценности. В частности, представлен специальный класс MDP с ограничениями на ценность политики (*Constrained Markov Decision Process*, CMDP), для которых предложен общий прямодвойственный подход к оптимизации, обладающий сильной двойственностью.

Ключевые слова: MDP, выпуклая оптимизация, Q-обучение, линейное программирование, методы градиента политики

Авторы выражают глубокую благодарность Даниилу Тяпкину за продуктивные дискуссии и ценные замечания. При подготовке работы использовалась презентация к обзорной лекции А. В. Гасникова на конференции MOTOR 2022. Исследование выполнено за счет гранта Российского научного фонда (проект № 21-71-30005), https://rscf.ru/project/21-71-30005/.

COMPUTER RESEARCH AND MODELING
2023 VOL. 15 NO. 2 P. 329–353
DOI: 10.20537/2076-7633-2023-15-2-329-353

Ки&М

MATHEMATICAL MODELING AND NUMERICAL SIMULATION

UDC: 519.8

Survey of convex optimization of Markov decision processes

V. D. Rudenko[1,2,a], N. E. Yudin[1,3,b], A. A. Vasin[4,c]

[1]Moscow Institute of Physics and Technology,
9 Institutskiy per., Dolgoprudny, Moscow region, 141701, Russia
[2]National Research University Higher School of Economics,
11 Pokrovsky Bulvar, Moscow, 109028, Russia
[3]Federal Research Center "Informatics and Control" of Russian Academy of Sciences,
44/2 Vavilova st., Moscow, 119333, Russia
[4]Lomonosov Moscow State University,
1/52 Leninskiye Gory, Moscow, 119991, Russia

E-mail: [a] Rudenko.VD@phystech.edu, [b] Iudin.NE@phystech.edu, [c] Vasin@cs.msu.su

Received 19.02.2023.
Accepted for publication 23.02.2023.

This article reviews both historical achievements and modern results in the field of *Markov Decision Process* (MDP) and convex optimization. This review is the first attempt to cover the field of reinforcement learning in Russian in the context of convex optimization. The fundamental Bellman equation and the criteria of optimality of policy — strategies based on it, which make decisions based on the known state of the environment at the moment, are considered. The main iterative algorithms of policy optimization based on the solution of the Bellman equations are also considered. An important section of this article was the consideration of an alternative to the Q-learning approach — the method of direct maximization of the agent's average reward for the chosen strategy from interaction with the environment. Thus, the solution of this convex optimization problem can be represented as a linear programming problem. The paper demonstrates how the convex optimization apparatus is used to solve the problem of *Reinforcement Learning* (RL). In particular, it is shown how the concept of strong duality allows us to naturally modify the formulation of the RL problem, showing the equivalence between maximizing the agent's reward and finding his optimal strategy. The paper also discusses the complexity of MDP optimization with respect to the number of state–action–reward triples obtained as a result of interaction with the environment. The optimal limits of the MDP solution complexity are presented in the case of an ergodic process with an infinite horizon, as well as in the case of a non-stationary process with a finite horizon, which can be restarted several times in a row or immediately run in parallel in several threads. The review also reviews the latest results on reducing the gap between the lower and upper estimates of the complexity of MDP optimization with average remuneration (*Averaged MDP*, AMDP). In conclusion, the real-valued parametrization of agent policy and a class of gradient optimization methods through maximizing the Q-function of value are considered. In particular, a special class of MDPs with restrictions on the value of policy (*Constrained Markov Decision Process*, CMDP) is presented, for which a general direct-dual approach to optimization with strong duality is proposed.

Keywords: MDP, convex optimization, Q-learning, linear programming, policy gradient methods

Citation: *Computer Research and Modeling*, 2023, vol. 15, no. 2, pp. 329–353 (Russian).

Authors thank Daniil Tiapkin for insightful thoughts and discussions. This work is partially based on a presentation for the review lecture by A. V. Gasnikov at the MOTOR 2022 conference. The research was supported by Russian Science Foundation (project No. 21-71-30005), https://rscf.ru/en/project/21-71-30005/.

Введение

Задачи обучения с подкреплением (*Reinforcement Learning*, RL) направлены на поиск почти оптимальной политики, определяющей стратегию выбора действий агента в зависимости от состояния, в котором он находится, для оптимизации долгосрочного процесса принятия решений в окружающей среде, максимизируя сумму кумулятивной награды. Таким образом, RL применяется в огромном множестве приложений, начиная от робототехники, медицины, рекомендательных систем и продолжая игровым искусственным интеллектом. По своему обычаю в задачах RL процесс принятия решений является последовательным с зависимостью текущих решений агента от совершенных ранее, при этом взаимодействие агента с окружающей средой происходит исключительно через процесс симуляции взаимодействия: наблюдение текущего состояния, принятие решения по текущему состоянию, переход в новое состояние и оценка полезности совершенного действия агентом. В это же время динамика, описывающая траектории агента, является неизвестной. Окружающая среда и лежащая в ее основе динамика обычно представляются как марковский процесс принятия решений (*Markov Decision Process*, MDP). Это приводит к рекуррентным уравнениям Беллмана, которые характеризуют функцию оптимального значения и политику поведения с точки зрения динамического программирования (*Dynamic Programming*, DP) проблемы RL [Bellman, 1966]. Наиболее эффективные подходы к решению задач RL базируются на принципах DP, описывая аппроксимацию стационарной точки, возникающей при раскрытии рекуррентной зависимости. Среди известных алгоритмов решения уравнений Беллмана можно обозначить temporal-difference (TD), включая алгоритм SARSA [Sutton, 1995], алгоритм Q-обучения (Q-learning) [Watkins, 1989] и его более современные варианты, базирующиеся на достижениях глубокого обучения [Mnih et al., 2015; Wang et al., 2016; Van Hasselt, Guez, Silver, 2016]. Несмотря на богатые аппроксимационные возможности TD-алгоритмов, их обучение (настройка параметров по данным симуляции) является нестабильной процедурой, которая даже может расходиться [Sutton, Barto, 1998].

Поставив перед собой цель снизить неустойчивость получаемой политики агента, естественно будет взглянуть на другую парадигму решения задач RL, основанную на задачах линейного программирования (*Linear Programming*, LP) [Manne, 1960; Denardo, 1970]. Целый ряд задач RL, в том числе оценивание полезности политики и оптимизация данной политики, сводится к соответствующим задачам линейного программирования с линейными ограничениями. Более того, с помощью принципа двойственности в выпуклой оптимизации возможно трансформировать исходную задачу в более удобную задачу LP для решения методами стохастической оптимизации. И стоит обратить внимание на то, что, хоть применение LP в задачах RL не так и ново, оно в последнее время вызвало заслуженный интерес в сообществе специалистов по обучению с подкреплением, во многом из-за большого потенциала как самого направления выпуклой оптимизации, так и из-за очевидной выгоды от обхода типичных проблем, возникающих при решении задач RL через принципы DP [Chen, Wang, 2016; Wang, 2017b; Serrano, Neu, 2020; Kamoutsi, Banjac, Lygeros, 2021; Zhang et al., 2022; Neu, Okolo, 2022].

В задаче RL для каждой модели агента интересуют количество и длина траекторий, описывающих его динамику, необходимых для нахождения почти оптимальной политики. Каждая такая траектория может быть представлена в терминах статистики как отдельная выборка, то есть возникает вопрос о сложности выборки. Для различных MDP существуют свои оценки сложности выборки, некоторые интересные результаты недавних статей будут представлены ниже.

Основная цель статьи состоит в обзоре наиболее важных результатов в области настройки MDP и применения для этого теории выпуклой оптимизации. С этой целью в первой части вводятся основная терминология на русском языке, понятие марковского процесса принятия решений, а также уравнения оптимальности Беллмана для функции значений и Q-функции. Во

второй части статьи приводятся алгоритмы итерации значений и политики. Третья часть посвящена Q-обучению и различным модификациям этого алгоритма с рассмотрением как ставших уже «классическими» результатов [Watkins, 1989], так и недавних [Jin et al., 2018]. Альтернативный подход — линейное программирование — рассматривается в четвертой части и является наиболее значимой частью статьи, рассматривающей MDP задачу в ее линейном представлении, показывая красоту математики. Далее следует часть про оценки на MDP, в которой приводятся наиболее интересные и важные результаты последних лет, а также в явном виде формулируется открытая проблема разрыва верхней и нижней оценок для AMDP (Averaged MDP), марковского процесса с усредненной наградой. В завершение представлены градиентные методы оптимизации политики — подход, аппроксимирующий оптимальную политику, сэмплируя[1] траектории агента из марковского процесса как из генеративной модели, а также свежие интересные результаты.

Марковский процесс принятия решений (MDP)

В обучении с подкреплением взаимодействия между агентом и окружающей средой часто описываются марковским процессом принятия решений (MDP). Различают дисконтированный марковский процесс принятия решений (*Discounted Markov Decision Process*, DMDP), MDP с усредненным вознаграждением (*Averaged Markov Decision Process*, AMDP), а также некоторые другие, описанные ниже. Марковский процесс принятия решений представляет собой систему, которая со временем ($t = 0, 1, 2, \ldots$) претерпевает случайные изменения и обозначается кортежем $M = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$ со следующими объектами.

(i) $\mathcal{S}$ — пространство состояний, $S := |\mathcal{S}|$ — количество уникальных состояний.

(ii) $\mathcal{A}$ — пространство действий, $A := |\mathcal{A}|$ — количество уникальных действий.

(iii) $p(s, a; s')$ — вероятность перехода из состояния $s \in \mathcal{S}$ в момент времени t с определенным действием $a \in \mathcal{A}$ в состояние $s' \in \mathcal{S}$ в момент $(t + 1)$ $\Big($при этом $\sum\limits_{s' \in \mathcal{S}} p(s, a; s') = 1$, $p(s, a; s') \equiv \mathsf{P}(s' \mid s, a)\Big)$. Функция вероятности $p(\cdot)$ вместе с функцией вероятности $\mathsf{P}(a|s)$ задают вероятности перехода для марковского ядра, оно же ядро MDP.

(iv) Функция награды $r_\xi(s, a)\colon \mathbf{\Omega} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ ($\mathbb{E}_\xi[r_\xi(s, a)] = R(s, a)$, где $\mathbb{E}[\cdot]$ — математическое ожидание). В зависимости от постановки задачи функция награды может зависеть от следующего за состоянием s состояния s':

$$r_\xi(s, a; s')\colon \mathbf{\Omega} \times \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1], \quad \mathbb{E}_\xi[r_\xi(s, a; s')] = R(s, a; s').$$

Стоит отметить, что мы предполагаем в общем случае стохастическую природу функции награды в зависимости от случайной величины $\xi \in \mathbf{\Omega}$. При детерминированном вычислении награды относительно фиксированных (s, a) или (s, a, s') мы просто опускаем обозначение ξ в $r_\xi(\cdot)$ и математическое ожидание по нему. В текущей работе используется предположение об ограниченности награды за каждое действие, поэтому без ограничений общности использованы приведенные выше определения функции $r(\cdot)$. В работе используются детерминированные относительно своих аргументов награды, если не оговорено иное.

[1] Сэмплирование — метод выбора подмножества наблюдаемых величин из данного множества с целью выделения неких свойств исходного множества. Сэмпл — элемент подмножества наблюдаемых величин из исходного множества. В данном случае сэмплирование траекторий означает их генерацию через взаимодействие агента со средой.

(v) $\gamma \in (0, 1]$ — коэффициент дисконтирования для DMDP, для AMDP $\gamma = 1$; но, просто положив $\gamma = 1$, из DMDP не сделать AMDP, понадобится еще усреднение суммарной награды агента за взаимодействие с MDP по времени.

Нередко рассматривается более общая форма $M = (\mathcal{S}, \mathcal{A}, p, r, \mu_0, \gamma)$, в которой μ_0 — вероятностное распределение начального состояния $s_0 \sim \mu_0$, при явном отсутствии μ_0 происходит обуславливание всех вычислений на $s_0 \in \mathcal{S}$. Тут нужно сделать важный комментарий по поводу природы пространств $\mathcal{S}$ и $\mathcal{A}$. Выше определения даны для пространств $\mathcal{S}$ и $\mathcal{A}$ с дискретной структурой. В случае $s' \in \mathcal{S}$ из областей непрерывности носителя функция $p(s, a; s')$ будет уже представлять собой плотность вероятности или, в общем случае, смесь из непрерывного распределения и дискретного распределения. Здесь и далее приводятся результаты для дискретных $\mathcal{S}$ и $\mathcal{A}$ с конечными мощностями, однако они могут быть обобщены на непрерывный случай заменой суммы по переменной в области ее непрерывности на соответствующий интеграл по области. При этом придется вероятность объекта в области непрерывности заменить на плотность вероятности этого объекта, а равенство вероятности данного объекта 1 означает замену функции плотности на дельта-функцию Дирака с центром в точке, совпадающей с этим же объектом. Более того, некоторые непрерывные области действий и состояний могут быть достаточно точно аппроксимированы (всюду) плотным множеством точек, то есть можно заменить фактический MDP на процесс с $S < \infty$ и $A < \infty$. Такой марковский процесс принятия решений с конечными мощностями множеств $\mathcal{S}$ и $\mathcal{A}$ называется табличным, или табулярным.

Политику агента обозначим через символ π и присвоим ему отображения, задающие вероятностную меру на пространстве действий:

$\pi(a|s) \equiv \mathsf{P}(a|s)$ в общем случае, $\widehat{a} \sim \pi(\cdot|s)$;

$\widehat{a} := \pi(s)$ в случае вырожденного распределения: $\mathsf{P}(\widehat{a}|s) = 1$ или $p(a|s) = \delta(a - \widehat{a})$.

Введенное распределение позволяет явно записать марковское ядро перехода между состояниями $s \mapsto s'$, оно же — ядро MDP, его также корректно называть марковским ядром, обусловленным политикой π:

$$\mathsf{P}^{\pi}(s'|s) := \sum_{a \in \mathcal{A}} \pi(a|s) p(s, a; s').$$

В процессах с конечным количеством состояний марковское ядро можно задать с помощью матрицы:

$$P^{\pi} = \left(\sum_{a \in \mathcal{A}} \pi(a|s) p(s, a; s') \right)_{s' \in \mathcal{S},\, s \in \mathcal{S}}, \quad s' \text{ соответствует строке, } s \text{ соответствует столбцу.}$$

Политика $\pi(\cdot)$ определяет стратегию принятия решений, в которой агент выбирает действия адаптивно на основе истории наблюдений; точнее, политика представляет собой (возможно, рандомизированное) отображение траектории ($\tau_t = (s_0, a_0, r_0, \ldots, s_t, a_t, r_t)$) на действие. Важно отметить, что при переходе в новое состояние агент продолжает работать с той же $\pi(\cdot)$. Правдоподобие траектории τ_{H-1}, полученной с помощью политики π, определяется следующим образом:

$$\mathsf{P}(\tau_{H-1}|\pi) = \mu_0(s_0) \prod_{t=0}^{H-2} (\pi(a_t|s_t) p(s_t, a_t; s_{t+1})) \pi(a_{H-1}|s_{H-1}).$$

Здесь и далее будем рассматривать DMDP, если не оговорено иное. Для фиксированной политики и начального состояния $s_0 = s$ определяется V-функция значений (ценности) $V^{\pi}\colon \mathcal{S} \to \mathbb{R}$ как дисконтированная сумма будущих вознаграждений:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \,\middle|\, \pi,\ s_0 = s \right],$$

где s_t — состояние системы в момент времени t, $a(s_t)$ — выбор действия в соответствии с политикой $\pi(\cdot)$. Здесь приведено определение функции ценности для марковского процесса с бесконечным горизонтом, этот момент не является принципиальным в нашем анализе, так как при замене ∞ на $H \in \mathbb{N}$ изменятся в основном мажоранты функций, определенных через дисконтированную кумулятивную награду за эпизод длины H, $t = \overline{0, H-1}$:

$$R_t^{H-1} := \sum_{j=t}^{H-1} \gamma^{j-t} r(s_j, a(s_j)) \quad \text{и} \quad R_t := R_t^{\infty} := \sum_{j=t}^{\infty} \gamma^{j-t} r(s_j, a(s_j)) \quad \text{для } H = \infty.$$

При переходе к AMDP ($\gamma = 1$) наиболее естественным аналогом кумулятивной награды является среднее арифметическое наград по времени:

$$R_t^{H-1} = \frac{1}{H-t} \sum_{j=t}^{H-1} r(s_j, a(s_j)) \quad \text{и} \quad R_t = R_t^{\infty} = \lim_{H\to\infty} \frac{1}{H-t} \sum_{j=t}^{H-1} r(s_j, a(s_j)) \quad \text{для } H = \infty.$$

Вернемся к DMDP. В нашем случае мажоранта на $V^{\pi}(\cdot)$ такая:

$$r(s, a) \in [0, 1]: \quad 0 \leqslant V^{\pi}(s) \leqslant \frac{1}{1-\gamma} \quad \forall s \in \mathcal{S},\ a \in \mathcal{A}.$$

В текущей работе рассматриваются награды со значениями из отрезка [0, 1], при наличии предположения об ограниченности награды сам отрезок не принципиален, так как оригинальный масштаб наград можно всегда биективно привести к отрезку [0, 1]. Для любых $\pi(a|s)$, s выполняется следующее условие согласованности между s и любыми последующими состояниями — уравнение Беллмана для функции значений в силу однородности марковского процесса:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \,\middle|\, \pi,\ s_0 = s\right] = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s, a; s') \left[R(s, a) + \gamma V^{\pi}(s')\right]. \tag{1}$$

Схожим образом задается Q-функция ценности $Q^{\pi}\colon \mathcal{S}\times\mathcal{A} \to \mathbb{R}$:

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t)) \,\middle|\, \pi,\ s_0 = s,\ a_0 = a\right].$$

Данная функция обладает той же мажорантой, что и $V^{\pi}(\cdot)$:

$$r(s, a) \in [0, 1]: \quad 0 \leqslant Q^{\pi}(s, a) \leqslant \frac{1}{1-\gamma} \quad \forall s \in \mathcal{S},\ a \in \mathcal{A}.$$

Цель задачи обучения с подкреплением (RL) — поиск политики, позволяющей получить максимальное кумулятивное вознаграждение в долгосрочной перспективе. В большинстве практических случаев задача RL формулируется как задача оптимизации следующего формата:

$$\pi^* \in \operatorname*{argmax}_{\pi\in\widehat{\Pi}} \left\{\mathbb{E}_{\mathsf{P}(\tau_{H-1}|\pi)}\left[R_0^{H-1}\right] = \mathbb{E}_{s\sim\mu_0}\left[V^{\pi}(s)\right]\right\};$$

$$\widehat{\Pi} := \left\{\pi \,\middle|\, \pi(a|s) \geqslant 0,\ \sum_{\widehat{a}\in\mathcal{A}} \pi(\widehat{a}|s) = 1,\ a \in \mathcal{A},\ s \in \mathcal{S}\right\}.$$

Ниже утверждение задает подкласс оптимальных политик, в рамках которого достаточно производить поиск интересующей π.

Теорема 1. *Пусть* Π *— набор всех нестационарных и рандомизированных политик.* $V^\pi(s)$, $Q^\pi(s, a)$ *зажаты между* 0 *и* $\frac{1}{1-\gamma}$, *следовательно, существуют конечные*

$$V^*(s) := \sup_{\pi\in\Pi} V^\pi(s), \quad Q^*(s, a) := \sup_{\pi\in\Pi} Q^\pi(s, a);$$

$\exists\pi$ *— стационарная, детерминированная, такая, что* $\forall s \in \mathcal{S}$, $a \in \mathcal{A}$

$$V^\pi(s) = V^*(s), \quad Q^\pi(s, a) = Q^*(s, a),$$

а значит, π *— оптимальная политика.*

В утверждении сверху мы можем легко заменить операцию sup на операцию max:

$$V^*(s) = \max_{\pi\in\Pi} V^\pi(s),$$

где V^* — оптимальная функция значений. Введем обозначение класса всех отображений, описывающих детерминированные политики в данном процессе:

$$\mathbb{A} = \{a(\cdot) \mid a\colon \mathcal{S} \mapsto \mathcal{A}\}.$$

Если воспользоваться принципом динамического программирования, то удается вывести уравнение оптимальности Беллмана на V-функцию ценности:

$$V^*(s) = \max_{a(\cdot)\in\mathbb{A}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a(s_t))\right] = \max_{a(\cdot)\in\mathbb{A}} \mathbb{E}\left[r(s, a(s)) + \gamma \sum_{t=0}^{\infty} \gamma^t r(s_{t+1}, a(s_{t+1}))\right] =$$

$$= \max_{a\in\mathcal{A}} \left(R(s, a) + \gamma\mathbb{E}\left[V^*(s')\right]\right) = \max_{a\in\mathcal{A}} \left(R(s, a) + \gamma \sum_{s'\in\mathcal{S}} p(s, a; s')V^*(s')\right). \quad (2)$$

Соответственно, мы можем провести аналогичные рассуждения для Q-функции ценности:

$$Q^*(s, a) = \max_{\pi\in\Pi} Q^\pi(s, a);$$

$$Q^*(s, a) = \mathbb{E}\left[r(s, a(s)) + \gamma V^*(s') \mid s_0 = s,\ a_0 = a\right].$$

Теорема ниже позволяет вывести критерий оптимальности относительно Q-функции ценности с приложением к дискретным пространствам $\mathcal{S}\times\mathcal{A}$, в котором функцию $Q^\pi(\cdot)$ при фиксированной политике можно однозначно определить как элемент евклидового пространства $\mathbb{R}^{|\mathcal{S}|\,|\mathcal{A}|}$.

Теорема 2. *Вектор* $Q \in \mathbb{R}^{|\mathcal{S}|\,|\mathcal{A}|}$ *представляет собой оптимальную функцию ценности* Q^*, *если и только если он удовлетворяет уравнениям оптимальности Беллмана*:

$$Q(s, a) = \mathbb{E}\left[r(s, a(s)) + \gamma \max_{a'\in\mathcal{A}} Q(s_{t+1}, a') \;\middle|\; s_0 = s,\ a_0 = a\right] =$$

$$= \sum_{s'\in\mathcal{S}} p(s, a; s')\left[R(s, a) + \gamma \max_{a'\in\mathcal{A}} Q(s', a')\right] \quad \forall s \in \mathcal{S},\ a \in \mathcal{A}. \quad (3)$$

Кроме того, детерминированная политика, определенная как

$$\pi(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} Q^*(s, a),$$

есть оптимальная политика.

Из утверждения выше можно получить аналогичный критерий оптимальности относительно V-функции ценности, заменив уравнение Беллмана для Q-функции на уравнение Беллмана для V-функции и выразив оптимальную Q-функцию через оптимальную V-функцию для выражения оптимальной политики.

Алгоритмы итеративной оптимизации ценности и политики

В предыдущем разделе мы рассмотрели критерии оптимальности, с помощью которых можно определить, какие из функций $V^\pi(\cdot)$ и $Q^\pi(\cdot)$ соответствуют оптимальным политикам π и как введенные критерии позволяют по функциям $V^\pi(\cdot)$ и $Q^\pi(\cdot)$ вычислить оптимальную политику. В данном разделе будут рассмотрены базовые алгоритмы поиска асимптотически оптимальных функций $V^*(\cdot)$ и $Q^*(\cdot)$. Рассматриваемый класс алгоритмов еще называют алгоритмами итерации ценности и политики. Данный класс включает в себя набор алгоритмов, которые используются для вычисления оптимальных политик на MDP. Уравнение (1) можно решать итеративным способом. Начальное распределение V_0 выбирается произвольно, а каждая последовательная итерация реализуется согласно уравнению Беллмана:

$$V_{k+1}(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s,\, a;\, s')[R(s,\, a) + \gamma V_k(s')],$$

где $V_k = V^\pi$ — фиксированная точка. Для получения каждого последующего приближения, V_{k+1} из V_k, при итеративной оценке политики применяется та же операция к каждому состоянию s, и ее называют ожидаемым обновлением, а данный алгоритм — итерации функции ценности.

Для определенной функции значения V^π произвольной детерминированной политики π и некоторых состояний s надо понимать, следует ли изменять политику, чтобы детерминированно выбирать действие $a \neq \pi(s)$. Один из способов: рассмотреть возможность выбора a, s и последующего следования существующей политике π согласно критерию (3). Если в терминах V-функций оно больше $V^\pi(s)$ (то есть если лучше выбрать a для s согласно (3) и затем следовать π, чем следовать π все время), тогда можно ожидать, что будет еще лучше выбирать a каждый раз, когда встречается s, и что новая политика на самом деле будет более эффективной. Ниже сформулирована теорема, указывающая на корректность описанной процедуры выбора улучшения политики π.

Теорема 3 (об улучшении политики). *Пусть π и π' — любая пара детерминированных политик, таких, что*

$$\forall s \in \mathcal{S} \quad Q^\pi(s, \pi'(s)) \geqslant V^\pi(s). \tag{4}$$

Тогда π' должна быть не хуже, чем π, то есть ценность не хуже $\forall s \in \mathcal{S}$:

$$V^{\pi'}(s) \geqslant V^\pi(s). \tag{5}$$

Более того, если в каком-либо состоянии существует строгое (4)*, то и* (5) *должно быть строгим.*

Ранее изменение политики оценивалось в одном состоянии s, учитывая π и $V^\pi(s)$. Однако можно рассмотреть изменения во всех s, выбирая в каждом состоянии действие, которое выглядит наилучшим образом в соответствии с $Q^\pi(s, a)$; таким образом, получается π' — новая «жадная» политика:

$$\pi'(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} Q^\pi(s,\, a) = \operatorname*{argmax}_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} p(s,\, a;\, s')\,[R(s,\, a) + \gamma V^\pi(s')], \tag{6}$$

где argmax обозначает множество значений a, при котором выражение максимизируется. «Жадная» политика выбирает лучшее действие в краткосрочной перспективе согласно V^π и удовлетворяет (4), что делает ее не хуже исходной. Если новая политика π' так же хороша, но не лучше π, то $V^\pi = V^{\pi'}$ и из (6) для $\forall s \in \mathcal{S}$ получается уравнение оптимальности Беллмана (2), следовательно, $V^{\pi'} = V^*$, π, π' — оптимальные политики.

Описанную ранее процедуру поиска оптимальной политики можно представить как последовательность монотонно улучшающихся политик и функций ценности:

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \ldots \xrightarrow{I} \pi_* \xrightarrow{E} V^*,$$

где $\xrightarrow{E}$ обозначает оценку политики, $\xrightarrow{I}$ — улучшение политики, то есть получен алгоритм итеративной оптимизации политики.

Q-обучение

Несложно заметить, что если

$$V^*(s) = \max_{a \in \mathcal{A}} Q(s, a),$$

то Q-функция должна удовлетворять *Q-уравнению*:

$$Q(s, a) = \sum_{s' \in \mathcal{S}} p(s, a; s') \Big(r(s, a; s') + \gamma \max_{a' \in \mathcal{A}} Q(s', a') \Big),$$

в текущем случае рассматривается уравнение Беллмана для более общего процесса MDP, в котором награда зависит уже от (s, a, s'), то есть добавилось еще и следующее за s состояние s'. С такой зависимостью наград удобнее рассматривать траекторию τ_{H-1} политики π как набор четверок (s, a, r, s') с уже несколько другим правдоподобием:

$$\tau_{H-1} = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, \ldots, a_{H-1}, r_{H-1}, s_H);$$
$$\mathsf{P}(\tau_{H-1}|\pi) = \mu_0(s_0) \prod_{t=0}^{H-1} \pi(a_t|s_t) p(s_t, a_t; s_{t+1}).$$

Данное уравнение Беллмана может быть решено методом простых итераций; если смотреть на $Q = \{Q(s, a)\}_{s \in \mathcal{S},\, a \in \mathcal{A}}$ как на вектор, то можно записать в операторном виде $Q = F(Q)$ (метод простых итераций будет иметь вид $Q_{t+1} = F(Q_t)$), где, по определению, оператор в правой части F является сжимающим с коэффициентом γ в норме Чебышёва:

$$\max_{s \in \mathcal{S},\, a \in \mathcal{A}} |F(\widetilde{Q}(s, a)) - F(Q(s, a))| \leqslant \gamma \max_{s \in \mathcal{S},\, a \in \mathcal{A}} |\widetilde{Q}(s, a) - Q(s, a)|, \quad \widetilde{Q} \text{ и } Q \text{ — произвольные.}$$

Основная идея Q-обучения заключается в замене невычислимой правой части в уравнении $Q_{t+1} = F(Q_t)$ на ее вычислимую несмещенную оценку:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha_t(s, a) \Big(r(s, a; s'(s, a)) + \gamma \max_{a' \in \mathcal{A}} Q_t(s'(s, a), a') - Q_t(s, a) \Big), \tag{7}$$

где $s'(s, a)$ — положение процесса на шаге $t + 1$, если на шаге t процесс был в состоянии s и было выбрано действие a, то параметр $0 < \alpha_t(s, a) \leqslant 1$, иначе $\alpha_t(s, a) = 0$. Правая часть (7) следует из перехода в $s'(s, a)$, $\{Q_t(s, a)\}_{s \in \mathcal{S},\, a \in \mathcal{A}}$ известно с прошлой итерации (можно посчитать $\max\limits_{a' \in \mathcal{A}} Q_t(s'(s, a), a')$). Вознаграждение $r(s, a; s'(s, a))$ получается при переходе из состояния s при действии a в $s'(s, a)$, в ненаблюдаемых случаях $\alpha_t(s, a) = 0$, то есть значение r не интересно. Подход Q-обучения, в отличие от ранее рассмотренных, полагается на сэмплирование непосредственно траекторий из MDP, что освобождает от знания всего пространства $\mathcal{S} \times \mathcal{A}$ в каждый конечный момент времени, при этом описанный процесс вычисления Q_{t+1} все также реализован через сжимающее отображение, гарантирующее асимптотическую сходимость к оптимальной политике, что сформулировано в теореме ниже.

Теорема 4. *Если при стратегии* $a(s)$ *с вероятностью* 1 *каждая пара* (s, a) *будет неограниченное число раз встречаться на бесконечном горизонте наблюдения, то при*

$$\sum_{t=0}^{\infty} \alpha_t(s, a) = \infty, \quad \sum_{t=0}^{\infty} \alpha_t^2(s, a) < \infty$$

следует сжимаемость (7):

$$\lim_{t\to\infty} Q_t(s, a) = Q(s, a), \quad V^*(s) = \max_{a\in\mathcal{A}} Q(s, a).$$

Таким образом, после достаточно большого числа шагов, даже в отсутствие какой-либо информации об управляемом марковском процессе, можно определить оптимальную стратегию $a(s) \in \operatorname{argmax}_{a\in\mathcal{A}} Q(s, a)$.

Алгоритм 1 отражает псевдокод шага процедуры Q-обучения, функция Q – learning вызывается после каждого перехода MDP в новое состояние.

Algorithm 1. Функция должна вызываться после каждого перехода.

function Q – learning(s_t, a, r, s_{t+1}, Q).

Input: s_t — последнее состояние, a — последнее действие, r — полученное немедленное вознаграждение, s_{t+1} — следующее состояние, Q — массив, хранящий текущую оценку функции значения действия.

1: $\delta \leftarrow r + \gamma \cdot \max_{a'\in\mathcal{A}} Q(s_{t+1}, a') - Q(s_t, a)$;

2: $Q(s_t, a) \leftarrow Q(s_t, a) + \alpha_t(s_t, a) \cdot \delta$;

return Q.

Если последовательность $(Q_t;\ t \geqslant 0)$ сходится к Q^*, где Q^* — оптимальная функция, тогда используются соответствующие локальные скорости обучения [Tsitsiklis, 1994; Jaakkola, Jordan, Singh, 1994]. Скорость сходимости Q-обучения была изучена [Szepesvari, 1997] в асимптотической постановке и в постановке с конечной выборкой [Even-Dar, Mansour, Bartlett, 2003].

Для Q-обучения характерно, что значения оптимального действия могут быть выражены в виде ожиданий, это позволяет оценивать значения действий поэтапно. Существуют многоступенчатые версии Q-learning [Sutton, Barto, 1998].

Плюсы Q-learning — простота, возможность использовать произвольную стратегию для генерации выборки обучающих данных при условии, что в пределе все пары «состояние–действие» обновляются конечное число раз, то есть подход Q-обучения является off-policy. Для замкнутого цикла обычно используемые стратегии заключаются в выборке действий, следующих схеме «ε-жадного выбора действия» или схеме Больцмана (в последнем случае вероятность выбора действия a в момент времени t выбирается пропорционально $e^{\beta Q_t(s_t, a)}$, $\beta > 0$). Стоит также заметить, что Q-обучение является одним из основных подходов при настройке MDP в так называемой парадигме model-free, то есть в процессе оптимизации происходит настройка политики $\pi(a|s)$, однако явного обучения самой динамики MDP, ядра марковского процесса не происходит.

Q-обучение с аппроксимацией функции

Если в предыдущем подразделе использовалась форма Q-функции, преимущественно определяемой табличными данными, то в текущем подразделе уже рассматривается аппроксимация Q-функции в заданном параметрическом семействе. Расширение Q-обучения для θ-параметрической аппроксимации функций $(Q_\theta;\ \theta \in \mathbb{R}^d)$ выражается через следующее соотношение:

$$\theta_{t+1} = \theta_t + \alpha_t(s, a)\left\{r(s, a;\ s'(s, a)) + \gamma \max_{a'\in\mathcal{A}} Q_{\theta_t}(s'(s, a), a') - Q_{\theta_t}(s, a)\right\} \nabla_\theta Q_{\theta_t}(s_t, a_t).$$

В алгоритме 2 продемонстрирован псевдокод, соответствующий случаю, когда используется метод линейной аппроксимации функций:

$$Q_\theta = \theta^\top \varphi, \quad \varphi\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d.$$

Хоть подобная аппроксимация и кажется простой, однако даже в таком классе θ-параметризованных Q-функций возможно аппроксимировать достаточно сложные зависимости через определение отображения φ, которое по смыслу выглядит как признаковое описание прецедента $(s, a) \in \mathcal{S} \times \mathcal{A}$ в евклидовом пространстве $\mathbb{R}^d$. Данная идея имеет тесную связь с ядровыми функциями, используемыми в машинном обучении и в статистической теории обучения для решения основных задач классификации, регрессии, кластеризации и некоторых других с помощью настройки обобщенных линейных моделей [Schölkopf, 2000; Grunewalder et al., 2012].

Algorithm 2. Функция должна вызываться после каждого перехода.

function Q – learningLinFApp(s_t, a, r, s_{t+1}, θ).

Input: s_t — последнее состояние, a — последнее действие, r — полученное немедленное вознаграждение, s_{t+1} — следующее состояние, $\theta \in \mathbb{R}^d$ — вектор параметра.

1: $\delta \leftarrow r + \gamma \cdot \max\limits_{a' \in \mathcal{A}} \theta^\top \varphi(s_{t+1}, a') - \theta^\top \varphi(s_t, a)$;

2: $\theta \leftarrow \theta + \alpha_t(s_t, a) \cdot \delta \cdot \varphi(s_t, a)$;

return θ.

Изначально результат сходимости был получен при сильных ограничениях распределения выборки [Melo et al., 2008]. Несколько позже с помощью жадного градиентного алгоритма Q-learning удалось снять сильные ограничения на распределение выборки, получив гарантию сходимости вне зависимости от распределения выборки [Maei et al., 2010]. Поскольку целевая функция, используемая при выводе, является невыпуклой, алгоритм может застрять в локальных минимумах даже с аппроксимацией линейной функции.

Q-обучение на основе интерполяции

Существует также другая разновидность аппроксимаций Q-learning, основанная на интерполяции (IBQ-learning) [Szepesvari, Smart, 2004]. IBQ одновременно обновляет все компоненты вектора параметров, тем самым уменьшая дисперсию обновлений. IBQ-обучение можно рассматривать как обобщение Q-обучения, используемого с агрегацией состояний и действий, на интерполяторы [Tsitsiklis, 1994].

Идея в том, чтобы рассматривать каждый компонент θ_i вектора параметров как оценку значения некоторой «репрезентативной» пары «состояние–действие»,

$$(s_i, a_i) \in \mathcal{S} \times \mathcal{A} \quad (i = 1, \ldots, d),$$

что делает Q_θ интерполятором, у которого параметр $\theta \in \mathbb{R}^d$ выбирается следующим образом:

$$Q_\theta(s_i, a_i) = \theta_i \quad \forall i = 1, \ldots, d. \tag{8}$$

Вместе с тем вводятся функции подобия на парах «состояние–действие» $l_i\colon \mathcal{S} \times \mathcal{A} \to [0, \infty)$, причем $l_i(\cdot)$ могут быть как зафиксированными, так и адаптивно настраиваемыми вместе с θ_i. Основное правило обновления параметров настраиваемой Q-функции в IBQ-learning следующее:

$$\theta_{t+1,i} = \theta_{t,i} + \alpha_{t,i}(s_t, a_t)\left\{r(s_t, a_t;\, s'(s_t, a_t)) + \gamma \max_{a' \in \mathcal{A}} Q_{\theta_t}(s'(s_t, a_t), a') - Q_{\theta_t}(s, a)\right\} l_i(s_t, a_t).$$

Каждый компонент обновляется в зависимости от того, насколько хорошо он прогнозирует общую будущую награду и насколько связана с ним пара «состояние–действие», похожая на

только что посещенную пару. Если сходство невелико, то влияние ошибки также будет небольшим. Алгоритм использует локальные последовательности размера шага $\left(\alpha_{t,i},\ t \geqslant 0\right)$.

Алгоритм сходится почти наверное [Szepesvari, Smart, 2004], если:

(i) Q_θ удовлетворяет вышеуказанному свойству интерполяции (8) и отображение $\theta \mapsto Q_\theta$ обладает свойством нерасширения:

$$\|Q_\theta - Q_{\theta'}\|_\infty \leqslant \|\theta - \theta'\|_\infty \quad \forall \theta,\ \theta' \in \mathbb{R}^d;$$

(ii) последовательности локального размера шага $\left(\alpha_{t,i};\ t \geqslant 0\right)$ выбраны соответствующим образом с асимптотикой как в условии теоремы 4;

(iii) каждая пара (s, a) из пространства состояний–действий $\mathcal{S} \times \mathcal{A}$ встречается с ненулевой вероятностью.

Так как отображение $\theta \mapsto Q_\theta$ не является расширением, алгоритм реализует инкрементную приближенную версию итерации значений, при этом базовым оператором является сжатие. Идея использования нерасширений впервые появилась в исследовании итерации с фиксированным значением [Gordon, 1995; Tsitsiklis, 1994].

Подобранная Q-итерация

В отличие от предыдущего подхода с *IBQ*-learning рассматривается более общий вид аппроксимации Q-функции ценности в параметрическом семействе с параметром θ. В данном случае алгоритм Q-обучения реализует оценку ценности политики относительно Q-уравнения, осуществляя процесс, похожий на метод простых итераций: $Q_{t+1} = F(Q_t)$. Учитывая текущее приближение Q_t, формируется приближение Монте-Карло к действию ядра MDP $p(\cdot)$ и оптимальной политики $\pi^*(\cdot)$ на Q_t в выбранных парах «состояние–действие», а затем выполняется регрессия по результирующим точкам. Алгоритм 3 показывает псевдокод этого метода. В алгоритме 3 отображение PREDICT оценивает оптимальную Q-функцию, оператор APPEND формирует обучающую выборку для настройки параметров θ, отображение REGRESS представляет собой регрессор, то есть процедуру решения задачи регрессии на выборке G для настройки аппроксимации оптимальной Q-функции PREDICT. Алгоритм 3 является инкрементальным и может быть реализован в онлайн-режиме с последовательным поступлением объектов из выборки D, представляющей собой траекторию MDP. Сама выборка D, описывающая взаимодействие со средой, в принципе, может быть результатом как случайного блуждания, случайного взаимодействия со средой, так и результатом работы другой политики, например результатом прошлого запуска

Algorithm 3. Функция должна вызываться до тех пор, пока не будет выполнен некоторый критерий сходимости.

function FittedQ(D, θ).

Input: $D = ((s_i, a_i, r_i, s_{i+1});\ i = 1, \ldots, n)$ — список переходов, θ — параметры регрессора.

1: $G \leftarrow []$;
2: **for** $i = 1 \to n$ **do**
3: $\quad R \leftarrow r_i + \max\limits_{a' \in \mathcal{A}} \text{PREDICT}((s_{i+1}, a'), \theta)$;
4: $G \leftarrow \text{APPEND}(G, \langle (s_i, a_i), R \rangle)$;
5: **end for**
6: $\theta \leftarrow \text{REGRESS}(G)$;

return θ.

процедуры FittedQ. Стоит заметить, что алгоритм 3 аппроксимирует Q-функцию оптимальной политики для MDP с $\gamma = 1$ и может быть легко обобщен на случай $\gamma \in (0, 1)$ добавлением множителя γ перед операцией $\max_{a' \in \mathcal{A}}(\cdot)$.

Подобранная Q-итерация может расходиться, если не используется специальный регрессор [Baird, 1995; Boyan, Moore, 1995; Tsitsiklis, Van Roy, 1996]. Существует несколько модификаций данного подхода: с использованием усреднения ядра MDP [Ormoneit, Sen, 2002], с использованием регрессоров на основе дерева [Ernst, Geurts, Wehenkel, 2005], также были получены хорошие эмпирические результаты с использованием нейронных сетей [Riedmiller, 2005].

Линейное программирование

Существуют альтернативные подходы к поиску оптимальной политики для известного MDP. Рассмотрим случай, в котором полностью известен MDP $M = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$; p, r, γ задаются рациональными числами.

Итерационные алгоритмы, описанные ранее, строго говоря, не являются алгоритмами полиномиального времени, поскольку они полиномиально зависят от $\frac{1}{1-\gamma}$. Ограничение на рациональность уже позволяет предложить алгоритм решения задачи RL с полилогарифмической сложностью, если не с полиномиальной, так как любое рациональное значение $1 - \gamma$ может быть задано только с помощью $O\left(\log \frac{1}{1-\gamma}\right)$ разрядов системы счисления с основанием, равным основанию логарифма. Использование линейного программирования (LP) обеспечивает алгоритм полиномиального времени, когда даны знания о MDP, у которого было бы время вычисления, зависящее от длины описания MDP M, а параметры заданы в виде рациональных чисел.

Для AMDP LP-задача вводится следующим образом:

$$V^* = \max_{a(\cdot) \in \mathbb{A}} \lim_{H \to \infty} \frac{1}{H} \mathbb{E}\left[\sum_{t=0}^{H-1} r(s_t, a_t(s_t))\right],$$

где H — эпизодическое ограничение, то есть максимальная длина эпизода. В случае эпизодов конечной длины предел опускается и используется максимальное значение H. Для политики $\pi(a|s)$ генерируется стационарное распределение:

$$\nu_\pi(s') = \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p(s, a; s') \pi(a|s) \nu_\pi(s), \quad s' \in \mathcal{S},$$

которое соответствует своему вектору из вероятностей $\nu_\pi = (\nu_\pi(s))_{s \in \mathcal{S}}$. И если MDP равномерно эргодично (все состояния марковской цепи из одного класса), то

$$V^\pi = \lim_{H \to \infty} \frac{1}{H} \left[\sum_{t=0}^{H-1} r(s_t, a_t(s_t))\right] = \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} R(s, a) \pi(a|s) \nu_\pi(s).$$

Вводится распределение действий по состояниям — $\mu(s, a) = \nu_\pi(s)\pi(a|s)$, следовательно, можно переписать задачу поиска оптимальной политики в AMDP как задачу LP со смыслом оценки ценности политики по распределению μ:

$$\max_{\mu \in \Delta^{\mathcal{S} \times \mathcal{A}}} \left[V(\mu) = \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} R(s, a) \mu(s, a) = \langle R, \mu \rangle : \sum_{b \in \mathcal{A}} \mu(s', b) = \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p(s, a; s') \mu(s, a), \ s' \in \mathcal{S} \right];$$

$$\Delta^{\mathcal{S} \times \mathcal{A}} = \left\{ \mu : \mu(s, a) \geqslant 0, \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mu(s, a) = 1 \right\}, \quad \pi_\mu(a|s) = \frac{\mu(s, a)}{\sum\limits_{b \in \mathcal{A}} \mu(s, b)}.$$

Данную задачу можно напрямую переписать в матричной форме ($\widehat{I}$ — нестандартная единичная матрица, P — матрица вероятностей перехода):

$$\max_{\mu \in \Delta^{\mathcal{S}\times\mathcal{A}}} \langle R, \mu\rangle;$$
$$s.t.\ (\widehat{I} - P)\mu = 0.$$

Единичная матрица $\widehat{I}$ имеет нестандартный формат: это прямоугольная матрица размером $S \times (SA)$, на каждой строке $s \in \mathcal{S}$ только элементы, соответствующие паре (s, a), $a \in \mathcal{A}$, равняются единице, остальные элементы данной строки равняются нулю, то есть на каждой строке $\widehat{I}$ ровно A единиц. У матрицы P размером $S \times (SA)$ в каждом столбце $(s, a) \in \mathcal{S} \times \mathcal{A}$ записано распределение $\mathsf{P}(\cdot|s, a)$. Здесь и далее отношения равенства и неравенства применяются поэлементно к скалярам, матрицам и векторам. Для этой задачи LP напрямую строится двойственная задача, с условием, что $\mu \geqslant 0$, которая имеет смысл оценки ценности оптимальной политики через V-функцию:

$$\min_{\overline{V}\in\mathbb{R},\, V\in\mathbb{R}^{|\mathcal{S}|}} \overline{V};$$
$$s.t.\ R - \overline{V}\cdot 1 - (\widehat{I} - P)^\top V \leqslant 0.$$

Таким образом, имеет место уравнение оптимальности Беллмана со средним вознаграждением:

$$V(s) = \max_{a\in\mathcal{A}} \left(R(s, a) - V^* + \sum_{s'\in\mathcal{S}} p(s, a; s')V(s') \right),$$

полученное из ограничений вида неравенства:

$$\widehat{I}^\top V \geqslant R - \overline{V} + P^\top V.$$

Здесь можно положить $\overline{V} = V^*$, так как решение задачи LP находится на границе допустимой области, заданной аффинными ограничениями.

Для DMDP задача LP на уравнение (2) записывается в следующем виде (q — распределение начального состояния μ_0 в виде вектора):

$$\min_{V\in\mathbb{R}^{|\mathcal{S}|}} \langle q, V\rangle;$$
$$s.t.\ R - (\widehat{I} - \gamma P)^\top V \leqslant 0,$$

и ей соответствует такая двойственная задача:

$$\max_{\mu \in \Delta^{\mathcal{S}\times\mathcal{A}}} \langle R, \mu\rangle;$$
$$s.t.\ (\widehat{I} - \gamma P)\mu = q.$$

Главная особенность введенных задач LP состоит в том, что их $O(|\mathcal{S}| \cdot |\mathcal{A}|)$ ограничений однозначно задают искомую переменную вне зависимости от оптимизируемого функционала, то есть ограничений слишком много, чтобы задача оптимизации непосредственно имела смысл, однако структура LP-проблем позволяет избавиться от части ограничений, и один из способов состоит в введении рандомизации по ограничениям, то есть на каждой итерации решения задачи LP рассматривается лишь сэмпл аффинных ограничений. Для DMDP в прямой задаче снизить количество аффинных ограничений позволяет следующее соответствие:

$$R - (\widehat{I} - \gamma P)^\top V \leqslant 0 \quad \Longleftrightarrow \quad \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left[R(s, a) - \left\langle \widehat{I}_{(s,a)} - \gamma P_{(s,a)}, V \right\rangle \right] \leqslant 0.$$

Здесь $\widehat{I}_{(s,a)}$ и $P_{(s,a)}$ являются столбцами $(s, a) \in \mathcal{S} \times \mathcal{A}$ соответствующих матриц $\widehat{I}$ и P. Так как множество ограничений полностью задает V, то достаточно итерироваться каким-либо методом стохастической оптимизации, минимизируя левую часть неравенства из ограничения в прямой задаче, например, используя стохастический градиентный спуск. Несмещенный стохастический градиент $R(s, a) - \left\langle \widehat{I}_{(s,a)} - \gamma P_{(s,a)}, V \right\rangle$ можно представить как $\widehat{I}_{(s,a)} - \gamma e_{P(s,a)}$, где $e_{P(s,a)}$ — вектор со всеми нулями и одной 1 на случайно выбранной позиции с распределением $P_{(s,a)} = p(s, a; \cdot)$. Данный подход применим аналогичным образом и к задаче LP в случае процесса AMDP, но уже к двойственной постановке.

Существует также постановка задачи LP для ограниченного DMDP (Constrained Markov Decision Process, CMDP):

$$\max_{\mu \in \Delta^{S \times \mathcal{A}}} \langle R, \mu \rangle;$$
$$s.t.\ (\widehat{I} - \gamma P)\mu = q, \quad D\mu \geqslant c.$$

По сравнению с предыдущими задачами линейного программирования вводится дополнительно аффинное ограничение вида неравенства: $D\mu \geqslant c$. Наиболее естественная интуиция подобного ограничения состоит в следующем: каждая строка матрицы D представляет собой функцию $Q^{\widehat{\pi}}(\cdot)$ для заданной заранее политики $\widehat{\pi}$, то есть настраиваемая политика π должна быть скоррелирована с каждой такой $\widehat{\pi}$ не меньше, чем на уровне значений элементов вектора c, соответствующих обозначенным строкам матрицы D. Постановка с ограничением $D\mu \geqslant c$ также может быть использована для настройки политики π в стиле off-policy, то есть мы не только максимизируем награду по текущей политике π, но еще и заставляем политику π быть похожей на экспертную политику $\widehat{\pi}$ в смысле установления корреляции между π и $\widehat{\pi}$ не ниже заданного уровня в соответствующем элементе вектора c. Заметим, что вместо обозначенного ранее распределения $\mu(s, a) = \nu_\pi(s)\pi(a|s)$ может быть полезно рассмотреть

$$\mu(s, a) := \mu^\pi(s, a) = \mathbb{E}_{s_0 \sim \mu_0}\left[\sum_{t=0}^{\infty} \gamma^t \mathsf{P}(s_t = s,\ a_t = a \mid s_0)\right]$$

или даже масштабированную сумму сверху в виде корректно определенной вероятностной меры:

$$\mu(s, a) := \widetilde{\mu}^\pi(s, a) = (1 - \gamma)\mu^\pi(s, a) = \mathbb{E}_{s_0 \sim \mu_0}\left[(1 - \gamma)\sum_{t=0}^{\infty} \gamma^t \mathsf{P}(s_t = s,\ a_t = a \mid s_0)\right].$$

В обоих случаях получается одна и та же политика:

$$\pi(a|s) = \frac{\mu^\pi(s, a)}{\sum\limits_{b \in \mathcal{A}} \mu^\pi(s, b)} = \frac{\widetilde{\mu}^\pi(s, a)}{\sum\limits_{b \in \mathcal{A}} \widetilde{\mu}^\pi(s, b)}.$$

Оценки на MDP

Марковское ядро MDP представляет собой генерирующую модель, принимает в качестве входных данных пару (s, a) и возвращает s', вдобавок вычисляется награда $r(s, a)$ или $r(s, a, s')$ ($r_\xi(s, a)$ или $r_\xi(s, a, s')$, если награды являются стохастическими). В данном случае можно утверждать, что траектория MDP представляет собой выборку из троек $(s, a, r(s, a))$ или четверок $(s, a, r(s, a, s'), s')$ в зависимости от удобства рассмотрения траекторий и постановки задачи. Их основной проблемой является зависимость от размера пространства состояний $S = |\mathcal{S}|$, который может быть чрезвычайно большим. Здесь и далее при обозначении асимптотики $\widetilde{O}(\cdot)$

для оценки сверху волнистая черта над символом означает сокрытие зависимости от полилогарифмических факторов[1]. Символ $\Omega(\cdot)$ означает оценку снизу[2]. В текущем разделе оценивается точность $\varepsilon > 0$ решения задачи RL по невязке на V-функцию ценности:

$$V^*(s) - V^{\pi}(s) \leqslant \varepsilon \quad \forall s \sim \mu_0.$$

Таблица 1. Оценки на различные MDP, под логарифмом скрыта зависимость от вероятности $\delta \in (0, 1)$, так как приведенные оценки на количество сэмплов выполнены с вероятностью $(1 - \delta)$

Тип	Фактор	Фактор количества пар (s, a)	Точность оценки V-функции	Число сэмплов $(s, a, r(s, a))$
DMDP	$\frac{1}{1-\gamma}$	SA	$\epsilon \simeq (1-\gamma)\varepsilon$	$\widetilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right)$ [Li et al., 2020]
HMDP	H	HSA	$\epsilon \simeq \varepsilon$	$\widetilde{O}\left(\frac{H^2SA}{\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right)$ [Tiapkin et al., 2022]
AMDP	t_{mix}	SA	$\epsilon \simeq \varepsilon$	$\widetilde{O}\left(\frac{t_{\text{mix}}^2 SA}{\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right)$ [Wang, 2017a] $\Omega\left(\frac{t_{\text{mix}} SA}{\varepsilon^2}\right)$ [Jin, Sidford, 2021]

Оценки на DMDP

Для оценки модели с точностью ϵ нужны $\widetilde{O}\left(\frac{SA}{\epsilon^2}\right)$ сэмплов, при этом можно поставить $\epsilon \simeq$ $\simeq (1-\gamma)\varepsilon$ (так как $V^* \sim (1-\gamma)^{-1}$) и получить из оценки сходимости при полной модели $O\left(\frac{1}{1-\gamma}\log\frac{1}{\varepsilon}\right)$ требуемый размер выборки (траектории) в случае табличного DMDP:

$$\widetilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right) \quad \text{[Wainwright, 2019].}$$

В классе методов на основе настройки Q-функции с применением уравнения Беллмана для решения задачи RL в DMDP была установлена оптимальность оценки размера выборки из таблицы 1 [Li et al., 2020].

Оценки на HMDP

Текущий подраздел посвящен эпизодическим марковским процессам принятия решений (H-episodic MDP, HMDP), то есть процессам с конечным горизонтом ($H < \infty$), что обязательно указывает на наличие терминального или терминальных состояний, из которых ни при каких действиях не получится выйти. Наиболее классическим примером таковых сред являются игровые среды, в которых терминальным состоянием является результат игры, а награда соответствует количественной оценке выигрыша или проигрыша. Нередко в таких средах награда начисляется только в терминальных состояниях, что на практике часто реализуется как нулевая награда во всех состояниях, кроме терминальных. Причем каждый эпизод может отличаться от других по длине, но все они не превосходят по длине общей мажоранты H. Для настройки оптимальной политики одного эпизода может не хватить в случае конечного горизонта, особенно если в марковском процессе принятия решений существует несколько терминальных состояний, то есть понадобится запускать эпизод как минимум два раза. Поэтому в данном подразделе обозначим через $T \in \mathbb{N}$ — количество эпизодов длины, не превышающей H, в анализе часто без

[1] $f(x) = \widetilde{O}(g(x)) \iff \exists C > 0$, P: $f(x) \leqslant C \cdot g(x) \cdot \mathrm{P}(x)\ \forall x \in \mathrm{dom}(f) \cap \mathrm{dom}(g) \cap \mathbb{R}_{++}^{\dim(x)}$, $\mathrm{P}(x)$ — полином от логарифмов элементов x.

[2] $f(x) = \Omega(g(x)) \iff \exists C > 0$: $f(x) \geqslant C \cdot g(x)\ \forall x \in \mathrm{dom}(f) \cap \mathrm{dom}(g)$.

ограничения общности рассматривают все эпизоды равной длины H, на практике данное допущение можно симулировать сэмплированием фиктивных состояний с нулевой наградой для любого действия после достижения терминального состояния, если фактическая длина эпизода оказалась меньше H. Такой подход во многом допустим вследствие использования минимаксных оценок сложности. Нетрудно догадаться, что в подобных ситуациях мы теряем стационарность у марковского процесса. В текущем подразделе в задаче настройки политики π рассматривается функция потерь следующего вида:

$$T \cdot (V^*(s) - V^\pi(s)) \quad \forall s \sim \mu_0.$$

В работах [Azar, Osband, Munos, 2017; Dann, Lattimore, Brunskill, 2017; Zanette, Brunskill, 2019] для эпизодических и в [Jaksch, Ortner, Auer, 2010; Fruit et al., 2018; Talebi, Maillard, 2018] для неэпизодических (то есть взаимодействуем со средой в рамках одного большого эпизода) MDP реализуется принцип оптимизма в условиях неопределенности введением бонусов к наградам. Добавляя их, можно построить верхние доверительные границы (Upper Confidence Bound, UCB) на основе неравенств концентрации меры Хёфдинга – Чернова или Бернштейна – Фридмана для оптимального значения Q-функции и действовать «жадно» по отношению к ним. В отличие от классических постановок MDP с данной таблицей переходов $s \mapsto s'$ разработанный алгоритм Upper Confidence Bound Value Iteration (UCBVI) [Azar, Osband, Munos, 2017] улучшает не только политику π, но еще и описание динамики среды, а именно частотно оценивает $\mathsf{P}(s'|s, a)$, то есть это полноценный подход к настройке MDP на основе обучения динамики среды $p(s, a; s')$ с трактовкой среды как черного ящика, из которого достаточно сэмплировать четверки (s, a, r, s') для попеременной оценки $Q^\pi(s, a)$ и $p(s, a; s')$.

Для такого подхода в [Azar, Osband, Munos, 2017] была доказана оценка функции потерь порядка $\widetilde{O}\left(\sqrt{H^3SAT}\right)$ — верхняя граница совпадает в первом порядке до полилогарифмических членов с известной нижней границей из [Domingues et al., 2021; Jin et al., 2018]. Недостатки метода:

(i) алгоритмы с бонусами, вычисляющие оптимальную границу на функцию потерь, часто плохо работают на практике, даже для простых MDP [Osband, Russo, Van Roy, 2013; Osband, Van Roy, 2017];

(ii) понятие графа, используемое в бонусах, нелегко обобщить на среды за пределами табличной постановки или на простые линейно-параметризованные среды, даже если некоторые решения существуют [Bellemare et al., 2016; Tang et al., 2017; Burda et al., 2019].

Возможно также использование принципа оптимизма в алгоритме настройки HMDP посредством введения шума. В работах [Osband, Russo, Van Roy, 2013; Osband et al., 2016b; Osband, Van Roy, 2017; Agrawal, Jia, 2017] предложена генерация выборки из апостериорного распределения для обучения с подкреплением с помощью алгоритма вида Posterior Sampling Reinforcement Learning, PSRL [Osband, Russo, Van Roy, 2013], являющаяся адаптацией алгоритма генерации выборки Томпсона [Thompson, 1933] для многоруких бандитов. Используя байесовский подход, PSRL хранит апостериорное распределение параметров MDP и в каждом эпизоде выбирает новый параметр из этого распределения, действуя «жадно» по отношению к нему. Несмотря на хорошую эмпирическую производительность по сравнению с алгоритмами на основе бонусов [Osband, Russo, Van Roy, 2013; Osband, Van Roy, 2017], неизвестно, может ли PSRL независимо от задачи достичь нижней границы. Лучшая оценка на функцию потерь имеет порядок $\widetilde{O}\left(H^2S\sqrt{AT}\right)$ [Agrawal, Jia, 2017; Qian et al., 2020].

Еще один достойный внимания подход к решению HMDP базируется на инкрементальной[1] Q-функции через минимизацию суммы функции потерь, задействованной в Q-обучении,

[1] Имеется в виду онлайн-обучение с итеративным увеличением размера выборки.

и регуляризатора, представляющего собой квадрат расстояния текущей аппроксимации Q-функции до случайного сэмпла из пространства допустимых Q-функций на основе гауссовского шума. Данные сэмплы в среднем выдают нулевое значение. С ростом номера итерации дисперсия генератора Q-функций стремится к нулю. Как и в предыдущих подходах, в самом подходе аппроксимируется и оптимальная политика, и ядро марковского процесса принятия решений. Соответствующий данному подходу алгоритм настройки HMDP называется Randomized Least Squares Value Iteration. RLSVI [Russo, 2019] показывает хорошую производительность на практике и может быть обобщен на задачи за пределами табличной постановки [Osband et al., 2019]. Существует обобщение RLSVI на среды для задач глубокого обучения с подкреплением [Osband et al., 2016a; Osband et al., 2018; Osband et al., 2019]. В частности, объединили RLSVI с Deep Q-Network, DQN [Mnih et al., 2015], заменив добавление гауссовского шума в RLSVI на генерацию выборки последующих вознаграждений с помощью бутстрапа [Efron, 1992]. Изначально было доказано, что оценка на функцию потерь для оригинальной версии RLSVI имеет порядок $\widetilde{O}\left(H^2 S^{3/2} \sqrt{AT}\right)$ [Russo, 2019], позже оценку улучшили до $\widetilde{O}\left(\sqrt{H^3 SAT}\right)$ [Xiong et al., 2021].

Но лучше всех на данный момент с решением задачи настройки HMDP в теории справляется алгоритм BayesUCBVI [Tiapkin et al., 2022], достигнув нижней оценки на порядок функции потерь. Алгоритм не полагается на бонусы, а использует квантили апостериорных значений Q-функции в качестве UCB на оптимальные значения Q-функции. Основные перечисленные в текущем подразделе алгоритмы решения HMDP и их оценки на невязку по V-функции приведены в таблице 2.

Таблица 2. Верхняя граница на оценку функции потерь агента $T \cdot (V^*(s) - V^{\pi}(s))$, $\forall s \sim \mu_0$ для эпизодического нестационарного табличного марковского процесса принятия решений

Алгоритм	Верхняя граница
UCBVI	$\widetilde{O}\left(\sqrt{H^3 SAT}\right)$ [Azar, Osband, Munos, 2017]
PSRL	$\widetilde{O}\left(H^2 S \sqrt{AT}\right)$ [Agrawal, Jia, 2017]
RLSVI	$\widetilde{O}\left(\sqrt{H^3 SAT}\right)$ [Xiong et al., 2021]
BayesUCBVI	$\widetilde{O}\left(\sqrt{H^3 SAT}\right)$ [Tiapkin et al., 2022]
Нижняя оценка	$\Omega\left(\sqrt{H^3 SAT}\right)$ [Tiapkin et al., 2022]

Оценки на AMDP

В AMDP роль фактора $(1-\gamma)^{-1}$ выполняет t_{mix}, являющееся наихудшим (для наихудшей π) временем смешивания для матрицы вероятности перехода:

$$t_{\text{mix}} := \max_{\pi \in \Pi} \left\{ \operatorname*{argmin}_{t \geqslant 1} \left\{ \max_{q \in \Delta^S} \left\{ \left\| (P^{\pi})^t \, q - \nu_{\pi} \right\|_1 \right\} \leqslant \frac{1}{2} \right\} \right\}.$$

Для оценки оптимального AMDP с точностью $\epsilon \simeq \varepsilon$ нужно как минимум $\Omega\left(\frac{t_{\text{mix}} SA}{\epsilon^2}\right)$ сэмплов, что значительно меньше, чем количественная оценка сэмплов у лучших существующих алгоритмов: $\widetilde{O}\left(\frac{t_{\text{mix}}^2 SA}{\varepsilon^2} \log\left(\frac{1}{\varepsilon}\right)\right)$, то есть поиск оптимального алгоритма относительно количества сэмплов является открытой проблемой и одной из наиболее интересных проблем, имеющих отношение к MDP.

Задача настройки AMDP представима как задача линейного программирования (LP). Например, для работы с неизвестными переходными ядрами марковского процесса текущие исследования были сосредоточены на использовании общих стохастических прямодвойственных

методов оптимизации седловой точки [Wang, 2017a; Jin, Sidford, 2020]. Несмотря на элегантность стохастического подхода LP, существуют значительные ограничения:

(i) из-за высокой размерности пространства состояний размер LP может быть огромным;

(ii) в MDP и RL обычно на практике требуется включить определенные нелинейные целевые функции и/или ограничения;

(iii) в текущих стохастических подходах LP для RL необходимо иметь доступ к генерирующей модели для моделирования вероятности перехода в следующее состояние для любой пары «состояние–действие».

Некоторые классические методы динамического программирования, особенно итеративная оптимизация ценности, были адаптированы для настройки процессов AMDP с неизвестными переходными ядрами [Abounadi, Bertsekas, Borkar, 2001; Gosavi, 2004; Wan, Naik, Sutton, 2021; Zhang, Zhang, Maguluri, 2021]. Однако полноценный переход к оценке генерирующей модели ухудшил оценку на общий размер траектории по сравнению с решением LP-задачи [Wang, 2017a; Jin, Sidford, 2020]. Недавняя работа [Jin, Sidford, 2021] устанавливает метод сокращения, который решает AMDP путем решения связанного DMDP. При использовании метода улучшается зависимость от параметров задачи (например, времени перемешивания), однако зависимость от показателя точности является субоптимальной: $O\left(\frac{1}{\varepsilon^3}\right)$. В другой недавней работе [Pesquerel, Maillard, 2022] предлагается оптимизировать эргодический AMDP как композицию моделей «многоруких бандитов», получая асимптотически оптимальную оценку на количество взаимодействий со средой относительно заданной динамики среды — $(p(s, a; \cdot), \mu_0(\cdot))$, а не в смысле min / max, то есть предлагается алгоритм, оценка сложности которого в худшем случае находится между нижней и верхней границами, указанными для AMDP в таблице 1. Также стоит заметить, что в работе [Pesquerel, Maillard, 2022] используется предположение о легких хвостах на распределение награды.

Методы градиента политики (Policy Gradient Methods)

В предыдущих разделах были рассмотрены в основном методы, производящие при обучении MDP настройку динамики среды и настройку самой политики агента, то есть рассматривался преимущественно подход с обучением динамики среды $p(s, a; s')$. В текущем разделе будет рассмотрен подход без обучения динамики среды $p(s, a; s')$, заключающийся в том, что теперь взаимодействие со средой идет как с «черным» ящиком, стохастическим оракулом нулевого порядка, из которого лишь можно генерировать траектории MDP, а обучается в итоге только политика MDP $\pi(a|s)$. Основной проблемой подхода на основе генерирующих моделей среды является зависимость от размера выборки $\mathcal{S}$, который может быть чрезвычайно большим. Наиболее популярным решением в подобных случаях стала параметризация политики $\pi_\theta(a|s)$, $\theta \in \mathbb{R}^{d'}$, обычно параметризация выбирается гладкой. При этом настройка агента происходит путем максимизации средней кумулятивной награды $J(\theta)$. Средняя кумулятивная награда может быть представлена несколькими способами, наиболее удобный из них для теоретического анализа записан ниже:

$$J(\theta) := \sum_{s\in\mathcal{S}} \nu_{\pi_\theta}(s) \sum_{a\in\mathcal{A}} \pi_\theta(a|s) Q^{\pi_\theta}(s, a).$$

Возможный алгоритм настройки π_θ будет, скорее всего, построен на основе стохастического градиентного подъема:

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)}, \quad \alpha > 0,$$

где $\widehat{\nabla J(\theta_t)} \in \mathbb{R}^{d'}$ — это стохастическая оценка, математическое ожидание которой аппроксимирует $\nabla J(\theta_t)$ относительно θ_t. $J(\theta)$ зависит от a под влиянием π, и $\nu_{\pi_\theta}(s)$ — распределение состояния s относительно π, которое зависит от окружающей среды и, как правило, неизвестно. Тем не менее для оценки $\nabla J(\theta_t)$ известно соотношение, сформулированное в виде теоремы о градиенте политики (Policy Gradient Theorem) [Sutton, Barto, 1998]:

$$\nabla J(\theta) = \sum_{s\in\mathcal{S}} \nu_{\pi_\theta}(s) \sum_{a\in\mathcal{A}} \pi_\theta(a|s)\left(Q^{\pi_\theta}(s,\, a)\nabla_\theta \ln \pi_\theta(a|s)\right) = \mathbb{E}_{\nu_{\pi_\theta},\pi_\theta}\left[Q^{\pi_\theta}(s,\, a)\nabla_\theta \ln \pi_\theta(a|s)\right]. \tag{9}$$

Для градиентных методов оптимизации политики в CMDP с большим количеством ограничений **M** общее количество итераций градиентного метода оптимизации политики равно $\widetilde{\mathcal{O}}\left((1-\gamma)^{-1}\varepsilon^{-1}\right)$, и на каждой итерации требуется $\simeq (1-\gamma)^2\varepsilon$ — приблизительное значение Q-функции (относительно $\pi_\theta(a|s)$ в (9)). Интересный результат для CMDP показан в [Li et al., 2021] — прямодвойственный подход с интеграцией трех компонент: оптимизатора политики с помощью регуляризации энтропии, l_2-регуляризации двойственной переменной и двойственного оптимизатора ускоренного градиентного спуска Нестерова. Введенный алгоритм AR-CPO [Li et al., 2021] сходится к глобальному оптимуму с количеством итераций $\widetilde{\mathcal{O}}\left(\frac{1}{\epsilon}\right)$. Целью CMDP является решение следующей задачи ограниченной оптимизации:

$$\begin{gathered}\max_{\pi\in\Pi} V_0^\pi(\mu_0);\\ s.t.\ V_i^\pi(\mu_0) \geqslant c_i, \quad i = 1,\, \ldots,\, m,\end{gathered} \tag{10}$$

где $V^\pi(\mu_0) = \mathbb{E}_{s_0\sim\mu_0}\left[V^\pi(s_0)\right]$. Популярным методом решения проблемы CMDP в уравнении (10) является прямодвойственный подход — решение минимаксной задачи над построенной функцией Лагранжа $\mathcal{L}(\pi,\, \lambda)$:

$$\min_{\lambda\in\mathbb{R}_+^m}\max_{\pi\in\Pi}\left\{\mathcal{L}(\pi,\, \lambda) := V_0^\pi(\mu_0) + \sum_{i=1}^m \lambda_i\left(V_i^\pi(\mu_0) - c_i\right) = V_0^\pi(\mu_0) + \left\langle\lambda,\, V^\pi(\mu_0) - c\right\rangle\right\},$$

где $V^\pi(s) = [V^{\pi_1}(s),\, \ldots,\, V^{\pi_m}(s)]^\top$, $\lambda = [\lambda_1,\, \ldots,\, \lambda_m]^\top$ обозначает вектор двойственной переменной, $c = [c_1,\, \ldots,\, c_m]^\top$ — вектор ограничений. Внутри алгоритма AR-CPO решается минимаксная задача над $(\upsilon,\, \rho)$ (регуляризированным лагранжианом) с помощью ускоренного двойственного спуска:

$$\min_{\lambda\in\mathbb{R}_+^m}\max_{\pi\in\Pi}\left\{\mathcal{L}_{\upsilon,\rho}(\pi,\, \lambda) := \mathcal{L}(\pi,\, \lambda) + \upsilon\mathcal{H}(\pi) + \frac{\rho}{2}\|\lambda\|_2^2\right\},$$

где $\mathcal{H}(\pi) = -\mathbb{E}\left[\sum\limits_{t=0}^{\infty}\gamma^t \log(\pi(a_t|s_t)) \;\middle|\; s_0 = s,\ a_t \sim \pi(\cdot|s_t)\right]$ — дисконтированная энтропия политики π и $\upsilon,\, \rho \geqslant 0$ — регуляризационные константы. Анализ сходимости AR-CPO основан на результате, состоящем в том, что обучение с ограниченным подкреплением имеет нулевой зазор двойственности [Paternain et al., 2019]:

Теорема 5 (о сильной двойственности). *Если r_i ограничено для $\forall i = 1,\, \ldots,\, m$ и выполняется условие Слейтера* (*теорема* 6) *для* (10)*, тогда имеет место сильная двойственность*[1] *для* (10).

Теорема 6. *Если $\exists\eta \in \mathbb{R}_+$ и хотя бы одна $\pi_\eta \in \Pi$ такая, что $\forall i = 1,\, \ldots,\, m$: $V_i^{\pi_\eta} \geqslant c_i + \eta$, то имеет место выполнение условия Слейтера.*

[1] Задача оптимизации называется сильно двойственной, если оптимальное значение ее оптимизируемого функционала совпадает с оптимальным значением оптимизируемого функционала задачи, двойственной к исходной.

На основе [Li et al., 2021] было получено дальнейшее развитие прямодвойственного подхода, в котором предлагается двойственный подход с интеграцией двух компонент: оптимизатора политики с помощью регуляризации энтропии и двойственной оптимизации по методу Вайды [Gladin et al., 2022]. Использование метода режущей плоскости Вайды в алгоритме для задачи выпуклой оптимизации со сложностью $\widetilde{O}\left(m \log \frac{m}{\epsilon}\right)$ делает его хорошим выбором для формул с малой или умеренной размерностью, таких как двойственная задача:

$$\min_{\lambda \in \mathbb{R}_+^m} \left\{ d_\upsilon(\lambda) := \max_{\pi \in \Pi} \mathcal{L}_\upsilon(\pi, \lambda) \right\}, \quad \upsilon > 0, \tag{11}$$

где υ — скалярный параметр, используемый для масштабирования $\mathcal{H}(\pi)$. Так как метод может быть использован с неточным субградиентом и не накапливает ошибку, то он подходит для задачи (11).

Заключение

В работе рассмотрен общий подход к решению задач, наиболее часто возникающих в обучении с подкреплением (RL) и имеющих математическое описание в виде различных марковских процессов принятия решений (MDP). Каждая описанная в текущей работе проблема естественно предстает в виде задачи математической оптимизации. Более того, в общем виде задачу поиска оптимальной политики по принятию решения в интересуемом процессе и задачу оценки полезности такой политики можно свести не просто к задаче выпуклой оптимизации, а именно к задаче линейного программирования. В рассмотренных MDP возможность сведения поиска оптимальной политики к задаче линейного программирования не зависит от природы функции награды $r(\cdot)$ и продолжительности взаимодействия с марковским процессом. Рассмотренные подходы к Q-обучению расширяют границы применения аппарата выпуклого анализа и выпуклой оптимизации для решения прикладных задач в RL.

В связи с наличием тесной связи между вероятностной теорией задач RL и выпуклой оптимизацией ожидается возникновение задач и их решений, вдохновленных как теорией вероятностей, статистикой, статистической теорией обучения, теорией RL, так и оптимизацией, в том числе и выпуклой, основанной на принципах сильной двойственности и динамического программирования. В недавно установленных взаимосвязях между RL и оптимизацией как таковой, в том числе и в данной работе, заложен большой потенциал для возникновения новых научных связей и интересных работ. Например, в еще не рассмотренных здесь постановках задачи RL (в случайных процессах с непрерывным временем, в случайных процессах принятия решений с несколькими взаимодействующими агентами $a(\cdot)$ в рамках одного процесса, в обучении с подкреплением, обусловленном функциональными ограничениями, описывающими допустимый или даже безопасный случайный процесс) формулировка проблемы зависит от конечного приложения. Вместе с тем большой интерес вызывают вопросы, связанные с формулировкой конкретной задачи RL как задачи стохастической аппроксимации или интерполяции в пространстве функций заданного класса как в выпуклом случае, так и в невыпуклом.

Список литературы (References)

Abounadi J., Bertsekas D. P., Borkar V. S. Learning algorithms for Markov decision processes with average cost // SIAM Journal on Control and Optimization. — 2001. — Vol. 40, No. 3. — P. 681–698.

Agarwal A., Kakade S., Yang L. F. Model-based reinforcement learning with a generative model is minimax optimal // Conference on Learning Theory. — PMLR, 2020. — P. 67–83.

Agrawal S., Jia R. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds // Advances In Neural Information Processing Systems. — 2017. — Vol. 30.

Azar M. G., Osband I., Munos R. Minimax regret bounds for reinforcement learning // International Conference On Machine Learning. — PMLR, 2017. — P. 263–272.

Baird L. C. Residual algorithms: Reinforcement learning with function approximation // Machine Learning Proceedings. — Morgan Kaufmann, 1995. — P. 30–37.

Bellemare M., Srinivasan S., Ostrovski G., Schaul T., Saxton D., Munos R. Unifying count-based exploration and intrinsic motivation // Advances in neural information processing systems. — 2016. — Vol. 29.

Bellman R. Dynamic programming // Science. — 1966. — Vol. 153, No. 3731. — P. 34–37.

Bertsekas D. P. Reinforcement learning and optimal control. — Athena Scientific, 2019.

Boyan J. A., Moore A. W. Generalization in reinforcement learning: Safely approximating the value function // Advances in neural information processing systems. — 1994. — Vol. 7.

Burda Y., Edwards H., Storkey A., Klimov O. Exploration by random network distillation // 7th International Conference on Learning Representations. — New Orleans, LA, USA, May 6–9, 2019.

Cen S., Cheng C., Chen Y., Wei Y., Chi Y. Fast global convergence of natural policy gradient methods with entropy regularization // Operations Research. — INFORMS, 2022. — Vol. 70, No. 4. — P. 2563–2578.

Chen Y., Wang M. Stochastic primal-dual methods and sample complexity of reinforcement learning // arXiv preprint arXiv:1612.02516. — 2016.

Dann C., Lattimore T., Brunskill E. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning // Advances in Neural Information Processing Systems. — 2017. — Vol. 30.

Denardo E. V. On linear programming in a Markov decision problem // Management Science. — 1970. — Vol. 16, No. 5. — P. 281–288.

Domingues O., Ménard P., Kaufmann E., Valko M. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited // Algorithmic Learning Theory. — PMLR, 2021. — P. 578–598.

Efron B. Bootstrap methods: another look at the jackknife. — Springer New York, 1992. — P. 569–593.

Ernst D., Geurts P., Wehenkel L. Tree-based batch mode reinforcement learning // Journal of Machine Learning Research. — 2005. — Vol. 6. — P. 503–556.

Even-Dar E., Mansour Y., Bartlett P. Learning Rates for Q-learning // Journal of machine learning Research. — 2003. — Vol. 5, No. 1.

Fruit R., Pirotta M., Lazaric A., Ortner R. Efficient bias-span-constrained exploration-exploitation in reinforcement learning // International Conference On Machine Learning. — PMLR, 2018. — P. 1578–1586.

Gladin E., Lavrik-Karmazin M., Zainullina K., Rudenko V., Gasnikov A., Takáč M. Algorithm for constrained Markov decision process with linear convergence // arXiv preprint arXiv:2206.01666. — 2022.

Gordon G. J. Stable function approximation in dynamic programming // Machine learning proceedings. — Morgan Kaufmann, 1995. — P. 261–268.

Gosavi A. Reinforcement learning for long-run average cost // European journal of operational research. — 2004. — Vol. 155, No. 3. — P. 654–674.

Grunewalder S., Lever G., Baldassarre L., Pontil M., Gretton A. Modelling transition dynamics in MDPs with RKHS embeddings // arXiv preprint arXiv:1206.4655. — 2012.

Jaakkola T., Jordan M., Singh S. On the convergence of stochastic iterative dynamic programming algorithms // Neural Computation. — 1994. — Vol. 6, No. 6. — P. 1185–1201.

Jaksch T., Ortner R., Auer P. Near-optimal regret bounds for reinforcement learning // Journal of Machine Learning Research. — Vol. 99. — P. 1563–1600.

Jin C., Allen-Zhu Z., Bubeck S., Jordan M. I. Is Q-learning provably efficient? // Advances in neural information processing systems. — 2018. — Vol. 31.

Jin Y., Sidford A. Efficiently solving MDPs with stochastic mirror descent // International Conference on Machine Learning. — PMLR, 2020.

Jin Y., Sidford A. Towards tight bounds on the sample complexity of average-reward MDPs // International Conference on Machine Learning. — PMLR, 2021.

Kamoutsi A., Banjac G., Lygeros J. Efficient performance bounds for primal-dual reinforcement learning from demonstrations // International Conference on Machine Learning. — 2021. — P. 5257–5268.

Li G., Wei Y., Chi Y., Gu Y., Chen Y. Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction // Advances in neural information processing systems. — 2020. — Vol. 33. — P. 7031–7043.

Li T., Guan Z., Zou S., Xu T., Liang Y., Lan G. Faster algorithm and sharper analysis for constrained Markov decision process // arXiv preprint arXiv:2110.10351. — 2021.

Liu T., Zhou R., Kalathil D., Kumar P. R., Tian C. Fast global convergence of policy optimization for constrained MDPs // arXiv preprint arXiv:2111.00552. — 2021.

Maei H., Szepesvari C., Bhatnagar S., Sutton R. Toward off-policy learning control with function approximation // ICML. — 2010. — Vol. 10. — P. 719–726.

Manne A. S. Linear programming and sequential decisions // Management Science. — 1960. — Vol. 6, No. 3. — P. 259–267.

Melo F. S., Meyn S. P., Ribeiro M. I. An analysis of reinforcement learning with function approximation // Proceedings of the 25th international conference on Machine learning. — 2008. — P. 664–671.

Mnih V., Kavukcuoglu K., Silver D., Rusu A. A., Veness J., Bellemare M. G., Graves A., Riedmiller M., Fidjeland A. K., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wierstra D., Legg S., Hassabis D. Human-level control through deep reinforcement learning // Nature. — 2015. — Vol. 518, No. 7540. — P. 529–533.

Neu G., Okolo N. Efficient global planning in large MDPs via stochastic primal–dual optimization // arXiv preprint arXiv:2210.12057. — 2022.

Ormoneit D., Sen S. Kernel-based reinforcement learning // Machine Learning. — 2002. — Vol. 49. — P. 161–178.

Osband I., Aslanides J., Cassirer A. Randomized prior functions for deep reinforcement learning // Advances In Neural Information Processing Systems. — 2018. — Vol. 31.

Osband I., Blundell C., Pritzel A., Van Roy B. Deep exploration via bootstrapped DQN // Advances in Neural Information Processing Systems. — 2016. — Vol. 29.

Osband I., Roy B., Russo D., Wen Z. Deep exploration via randomized value functions // Journal of Machine Learning Research. — 2019. — Vol. 20, No. 124. — P. 1–62.

Osband I., Roy B., Wen Z. Generalization and exploration via randomized value functions // Proceedings of the 33rd International Conference on Machine Learning. — PMLR, 2016. — P. 2377–2386.

Osband I., Russo D., Van Roy B. (More) Efficient reinforcement learning via posterior sampling // Advances in Neural Information Processing Systems. — 2013. — Vol. 26.

Osband I., Van Roy B. Why is posterior sampling better than optimism for reinforcement learning? // Proceedings of the 34th International Conference on Machine Learning. — PMLR, 2017. — P. 2701–2710.

Paternain S., Chamon L., Calvo-Fullana M., Ribeiro A. Constrained reinforcement learning has zero duality gap // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.

Pesquerel F., Maillard O. IMED-RL: Regret optimal learning of ergodic Markov decision processes // NeurIPS 2022-Thirty-sixth Conference on Neural Information Processing Systems. — 2022. — Vol. 35.

Puterman M. L. Markov decision processes discrete stochastic dynamic programming. — University of British Columbia, 1994.

Qian J., Fruit R., Pirotta M., Lazaric A. Concentration inequalities for multinoulli random variables // arXiv preprint arXiv:2001.11595. — 2020.

Riedmiller M. Neural fitted Q iteration-first experiences with a data efficient neural reinforcement learning method // Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3–7, 2005. Proceedings 16. — Springer. — P. 317–328.

Russo D. Worst-case regret bounds for exploration via randomized value functions // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.

Schölkopf B. The kernel trick for distances // Advances in neural information processing systems. — 2000. — Vol. 13.

Serrano J. B., Neu G. Faster saddle-point optimization for solving large-scale Markov decision processes // Learning for Dynamics and Control. — 2020. — P. 413–423.

Sidford A., Wang M., Wu X., Yang L., Ye Y. Near-optimal time and sample complexities for solving Markov decision processes with a generative model // Advances in Neural Information Processing Systems. — 2018. — Vol. 31.

Sidford A., Wang M., Wu X., Ye Y. Variance reduced value iteration and faster algorithms for solving Markov decision processes // Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms. — Society for Industrial and Applied Mathematics, 2018. — P. 770–787.

Singh S. P., Jaakkola T., Littman M. L., Szepesvari C. Convergence results for single-step on-policy reinforcement-learning algorithms // Machine Learning. — 2000. — Vol. 38, No. 3. — P. 287–308.

Sutton R. S. Generalization in reinforcement learning: Successful examples using sparse coarse coding // Advances in neural information processing systems. — 1995. — Vol. 8.

Sutton R. S., Barto A. G. Reinforcement learning: An introduction. — Bradford Book, MIT Press, 1998.

Szepesvari C. The asymptotic convergence-rate of Q-learning // Advances in Neural Information Processing Systems. — 1997. — Vol. 10.

Szepesvari C. Static and dynamic aspects of optimal sequential decision making // Unpublished Ph. D. dissertation, Bolyai Institute of Mathematics, "József Attila" University, Szeged, Hungary. — 1998.

Szepesvari C., Smart W. D. Interpolation-based Q-learning // Proceedings of the twenty-first international conference on Machine learning. — 2004.

Talebi M. S., Maillard O. Variance-aware regret bounds for undiscounted reinforcement learning in MDPs // Algorithmic Learning Theory. — PMLR, 2018. — P. 770–805.

Tang H., Houthooft R., Foote D., Stooke A., Xi Chen O., Duan Y., Schulman J., DeTurck F., Abbeel P. Exploration: a study of count-based exploration for deep reinforcement learning // Advances in Neural Information Processing Systems. — 2017. — Vol. 30.

Thompson W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples // Biometrika. — Oxford University Press, 1933. — Vol. 25, No. 3–4. — P. 285–294.

Tiapkin D., Belomestny D., Moulines É., Naumov A., Samsonov S., Tang Y., Valko M., Ménard P. From Dirichlet to Rubin: Optimistic exploration in RL without bonuses // International Conference on Machine Learning. — PMLR, 2022. — P. 21380–21431.

Tiapkin D., Gasnikov A. Primal-dual stochastic mirror descent for MDPs // International Conference on Artificial Intelligence and Statistics. — PMLR, 2022. — P. 9723–9740.

Tsitsiklis J. N. Asynchronous stochastic approximation and Q-learning // Machine Learning. — 1994. — Vol. 16, No. 3. — P. 185–202.

Tsitsiklis J. N., Van Roy B. Feature-based methods for large scale dynamic programming // Machine Learning. — 1996. — Vol. 22. — P. 59–94.

Van Hasselt H., Guez A., Silver D. Deep reinforcement learning with double q-learning // Proceedings of the AAAI conference on artificial intelligence. — 2016. — Vol. 30, No. 1.

Wainwright M. J. Variance-reduced Q-learning is minimax optimal // arXiv preprint arXiv:1906.04697. — 2019.

Wan Y., Naik A., Sutton R. S. Learning and planning in average-reward Markov decision processes // International Conference on Machine Learning. — PMLR, 2021. — P. 10653–10662.

Wang M. Primal-dual π-learning: Sample complexity and sublinear run time for ergodic Markov decision problems // arXiv preprint arXiv:1710.06100. — 2017.

Wang M. Randomized linear programming solves the discounted Markov decision problem in nearly-linear (sometimes sublinear) running time // arXiv preprint arXiv:1704.01869. — 2017.

Wang Z., Schaul T., Hessel M., Hasselt H., Lanctot M., Freitas N. Dueling network architectures for deep reinforcement learning // International conference on machine learning. — 2016. — P. 1995–2003.

Watkins C. J. C. H. Learning from delayed rewards. — King's College, Cambridge United Kingdom, 1989.

Xiong Z., Shen R., Cui Q., Du S. S. Near-optimal randomized exploration for tabular MDP // arXiv preprint arXiv:2102.09703. — 2021.

Ying D., Ding Y., Lavaei J. A dual approach to constrained Markov decision processes with entropy regularization // International Conference on Artificial Intelligence and Statistics. — PMLR, 2022. — P. 1887–1909.

Zanette A., Brunskill E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds // Proceedings of the 36th International Conference on Machine Learning, (ICML). — PMLR, 2019. — P. 7304–7312.

Zhan W., Cen S., Huang B., Chen Y., Lee J. D., Chi Y. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence // arXiv preprint arXiv:2105.11066. — 2021.

Zhang S., Zhang Z., Maguluri S. T. Finite sample analysis of average-reward TD learning and Q-learning // Advances in Neural Information Processing Systems. — 2021. — Vol. 34.

Zhang T., Ren T., Yang M., Gonzalez J., Schuurmans D., Dai B. Making linear mdps practical via contrastive representation learning // International Conference on Machine Learning. — PMLR, 2022. — P. 26447–26466.

Zhang Z., Ji X. Regret minimization for reinforcement learning by evaluating the optimal bias function // arXiv preprint arXiv:1906.05110. — 2019.

Zhang Z., Zhou Y., Ji X. Almost optimal model-free reinforcement learning via reference-advantage decomposition // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 15198–15207.