Лекция 2: Монте-Карло, временные различия, Q-обучение

Алексей Скрынник
Артем Латышев

План лекции

1. Метод Монте-Карло
2. Методы временных различий
3. Q-обучение

Метод Монте-Карло

Монте-Карло подход к обучению с подкреплением

1. Будем оценивать стратегию напрямую по эпизодам взаимодействия со средой.
2. Используем безмодельный подход (model-free): модель переходов МППР и функция вознаграждения не известны.
3. Будем проводить обучение по полным эпизодам.
4. Подход Монте-Карло (МК) использует максимально простую идею: полезность равна средней отдаче.

Монте-Карло оценка стратегии

- Цель: построить V^π по эпизодам взаимодействия по стратегии π:

$$s_1, a_1, r_1, \dots s_k \sim \pi$$

- *Отдача* (return) – это суммарное вознаграждение:

$$R_t = r_{t+1} + \gamma r_{t+2} + \dots + \gamma^{T-1} r_T$$

- Функция полезности – это матожидание отдачи:

$$V^\pi(s) = \mathbb{E}_\pi[R_t | s_t = s]$$

- Монте-Карло оценка стратегии использует *эмпирическое среднее* отдачи вместо матожидания

Монте-Карло оценка стратегии с первым посещением

Оцениваем состояние s:

- для **первого** по времени посещения состояния s в эпизоде:
- $N(s) \leftarrow N(s) + 1$,
- $S(s) \leftarrow S(s) + R_t$

Полезность оцениваем как среднюю отдачу: $V(s) = S(s)/N(s)$

По закону больших чисел:

$$V(s) \xrightarrow{N(s)\to\infty} V^{\pi}(s)$$

Несмещенная, состоятельная оценка с высокой дисперсией

Монте-Карло оценка стратегии с каждым посещением

Оцениваем состояние s:

- для **каждого** по времени посещения состояния s в эпизоде:
- $N(s) \leftarrow N(s) + 1$,
- $S(s) \leftarrow S(s) + R_t$

Полезность оцениваем как среднюю отдачу: $V(s) = S(s)/N(s)$

По закону больших чисел:

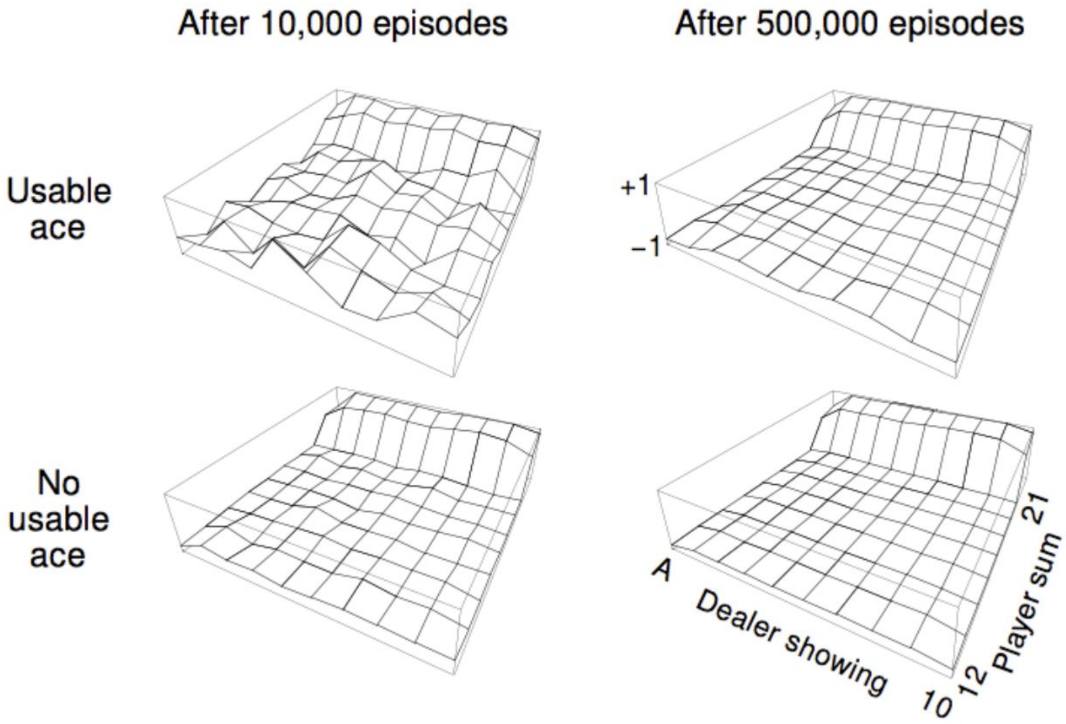
$$V(s) \xrightarrow{N(s) \to \infty} V^{\pi}(s)$$

Смещенная, состоятельная оценка с более низкой дисперсией.

Пример: блек-джек

- **200 состояний:**
  - Текущая сумма (12-21)
  - Дилер показывает карту (максимально - 10 очков)
  - Есть ли особая комбинация (да/нет)
- **Действия:**
  - **Stick:** Не получать карты (завершить игру)
  - **Twist:** Взять новую карту (без замены)
- **Вознаграждения за Stick:**
  - +1, если сумма карт больше, чем у дилера
  - 0, если сумма карт такая же, как у дилера
  - -1, если сумма карт меньше, чем у дилера
- **Вознаграждения за Twist:**
  - -1, если сумма карт больше 21
  - 0 иначе
- **Переходы:**
  - Автоматически Twist, если сумма карт меньше 12

Функция полезности после обучение MC



Среднее приращений

Средние последовательности могут быть вычислены последовательно (incremental mean):

$$
\begin{aligned}
\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j = \\
&= \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) = \\
&= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) = \\
&= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)
\end{aligned}
$$

Монте-Карло для приращений

- Обновим $V(s)$ с приращением (incrementally) для эпизода $s_1, a_1, r_1, \ldots, s_T$
- Для каждого состояния s_t с отдачей R_t:

$$N(s_t) \leftarrow N(s_t) + 1,$$

$$V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(R_t - V(s_t))$$

- В нестационарных задачах может быть полезно отслеживать текущее среднее:

$$V(s_t) \leftarrow V(s_t) + \alpha(R_t - V(s_t))$$

Метод временных различий

Обучение на основе временных различий

- **Подход:**
  - Используем безмодельный подход (model-free): модель переходов МППР и функция вознаграждения не известны
- **Рассмотрение неполных эпизодов:**
  - Использование бутстрепа (bootstrapping) для получения информации о оставшихся будущих шагах
- **Метод временных различий (Temporal-Difference, TD):**
  - Идея метода: приближать значение полезности на основе предыдущего приближения

Монте-Карло и временные различия

- Цель: построить V^{π} интерактивно (online) по эпизодам взаимодействия по стратегии π
- Монте-Карло с каждым посещением для приращений: обновляем $V(s_t)$ на основе текущей отдачи R_t

$$V(s_t) \leftarrow V(s_t) + \alpha(R_t - V(s_t))$$

- Самый простой подход временных различий: TD(0):
  - обновляем на основе *ожидаемой* отдачи $r_{t+1} + \gamma V(s_{t+1})$

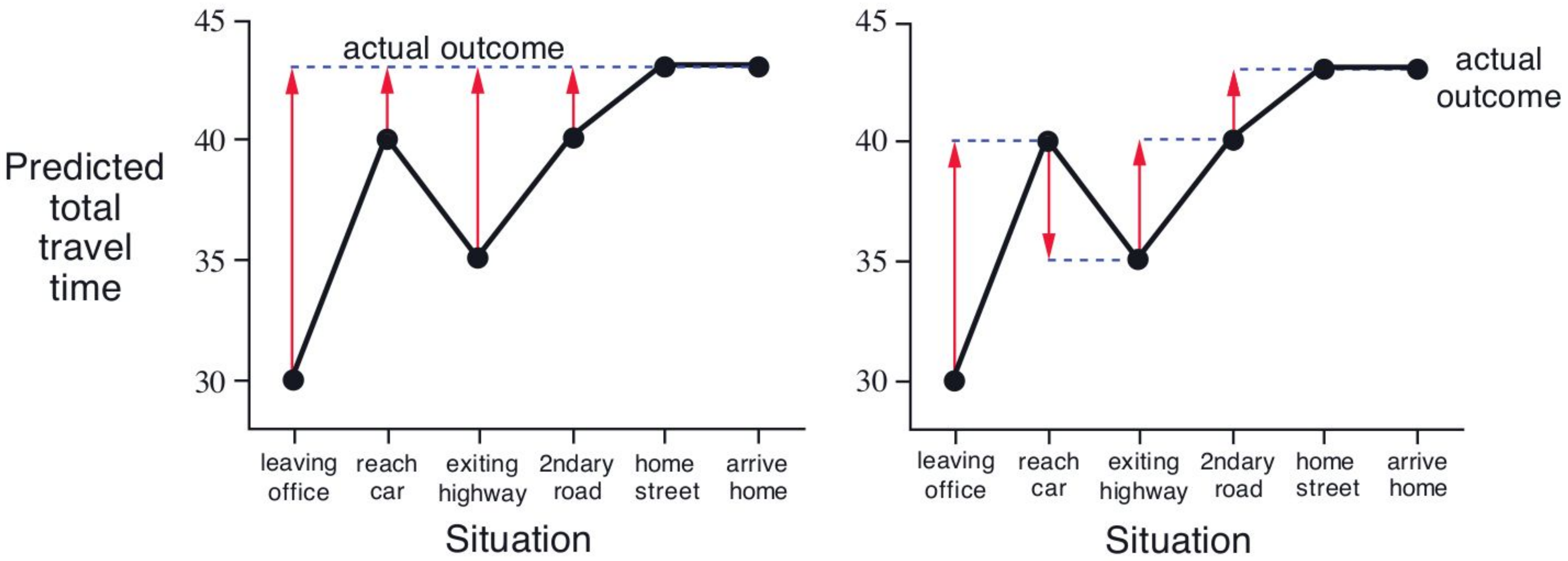
$$V(s_t) \leftarrow V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)),$$

  - $r_{t+1} + \gamma V(s_{t+1})$ называется *TD показателем*,
  - $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ называется *TD ошибкой*

Пример: поездка домой

Состояние	Истекшее время (мин)	Ожидаемое оставшееся время (мин)	Предсказываемое общее время (мин)
Выйти из офиса	0	30	30
Начать поездку, дождь	5	35	40
Съехать с шоссе	20	15	35
Ехать за грузовиком	30	10	40
Домашняя улица	40	3	43
Зайти домой	43	0	43

Пример: Монте-Карло и временные различия



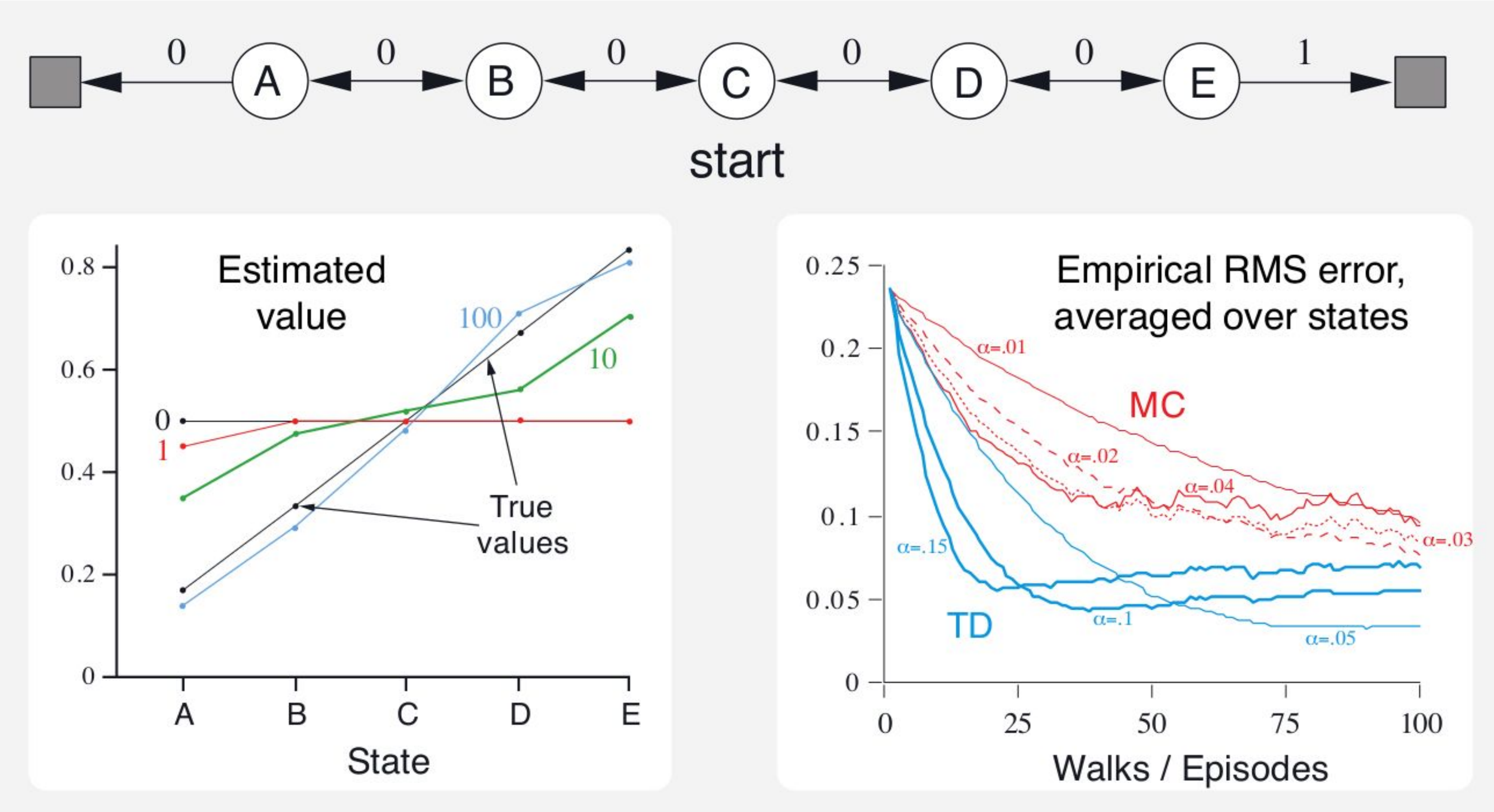
Смещенность и дисперсия

- Отдача $R_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_T$ является несмещенной оценкой для $V^\pi(s)$
- Истинный TD показатель $r_{t+1} + \gamma V^\pi(s_{t+1})$ является несмещенной оценкой для $V^\pi(s)$
- TD показатель $r_{t+1} + \gamma V(s_{t+1})$ является смещенной оценкой для $V^\pi(s)$
- TD показатель имеет меньшую дисперсию, чем отдача:
  - отдача зависит от *большого* количества случайных действий, переходов, вознаграждений,
  - показатель зависит только от *одного* случайного действия, перехода и вознаграждения

Преимущества и недостатки

- МК обладает высокой дисперсией и нулевым смещением:
  - ▸ может обучаться только на полных эпизодах,
  - ▸ МК работает только в эпизодических окружениях (с терминальными состояниями),
  - ✓ хорошие показатели сходимости (даже без аппроксимации),
  - ✓ не очень сильно зависит от начального приближения,
  - ✓ очень прост для понимания и использования,
- ВР имеет низкую дисперсию, ненулевое смещение:
  - ✓ может обучаться интерактивно на каждом шаге,
  - ✓ работает и для бесконечных (без терминального состояния) окружений
  - ✓ обычно более эффективен, чем МК,
  - ▸ TD(0) сходится к $V^{\pi}(s)$ (но не всегда при использовании аппроксимации),
  - ▸ более чувствителен к начальному приближению

Пример: МК vs. ВР



Пакетные МК и ВР

- MC и TD сходятся к $V(s) \to V^{\pi}(s)$, если опыт $\to \infty$
- А если мы применим пакетный (batch) подход для конечного опыта?

$$\begin{gathered} s_1^1, a_1^1, r_1^1, \ldots, s_{T_1}^1 \\ \vdots \\ s_1^K, a_1^K, r_1^K, \ldots, s_{T_K}^K \end{gathered}$$

- Например, многократно выбирать эпизод $k \in [1, K]$
- Применять MC или TD(0) к эпизоду k

Пример: AB

Два состояния A,B; нет дисконтирования, 8 эпизодов опыта:

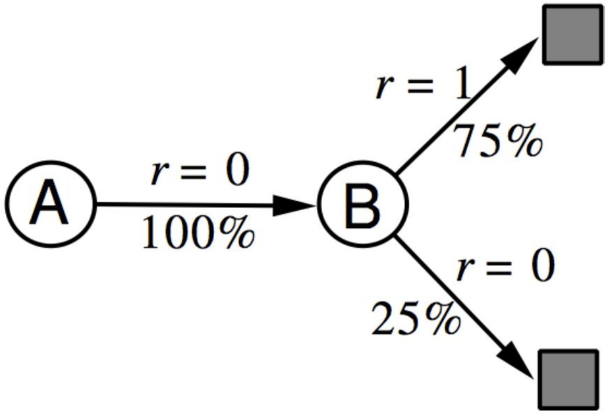
A, 0, B, 0
B, 1
B, 1
B, 1
B, 1
B, 1
B, 1
B, 0

Какие значения $V(A)$ и $V(B)$?

Пример: AB

Два состояния A,B; нет дисконтирования, 8 эпизодов опыта:

A, 0, B, 0
B, 1
B, 1
B, 1
B, 1
B, 1
B, 1
B, 0



Какие значения $V(A)$ и $V(B)$?

Эквивалентность

- МК сходится к решению с минимальной среднеквадратичной ошибкой
  - Наилучшее приближение к наблюдаемой отдаче:

$$\sum_{k=1}^{K}\sum_{t=1}^{T_k}(R_t^k - V(s_t^k))^2$$

  - Для AB примера $V(A) = 0$
- TD(0) сходится к решению максимально правдоподобной марковской модели
  - Решение для МППР $\langle S, A, \hat{\mathcal{P}}, \hat{\mathcal{R}}, \gamma \rangle$, который лучше всего удовлетворяет данным

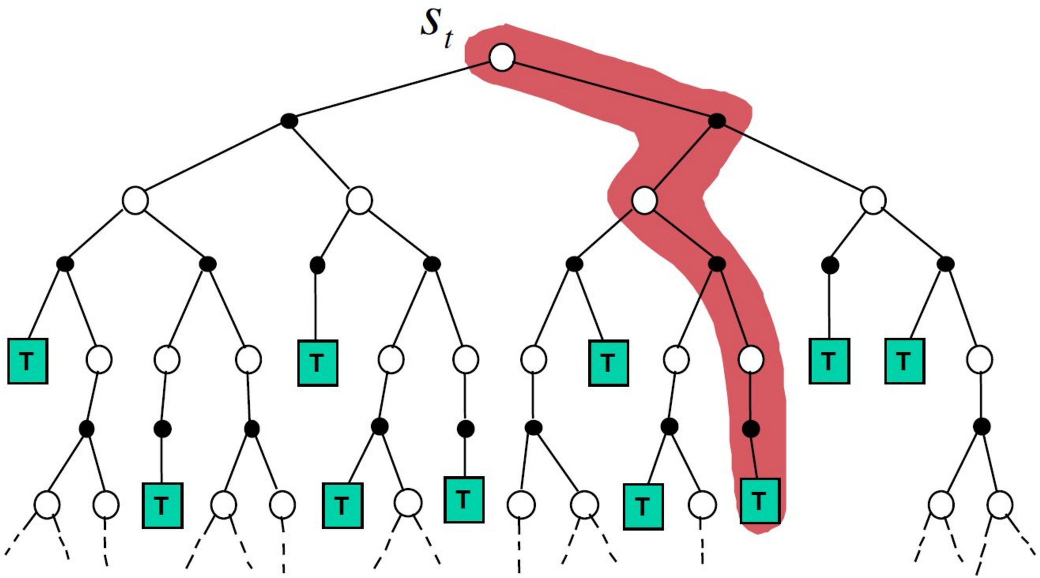
$$\hat{\mathcal{P}}_{ss'}^a = \frac{1}{N(s,a)}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbf{1}(s_t^k, a_t^k, s_{t+1}^k = s, a, s'),$$

$$\hat{\mathcal{R}}_s^a = \frac{1}{N(s,a)}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbf{1}(s_t^k, a_t^k = s, a) r_t^k$$

  - Для AB примера $V(A) = 0.75$

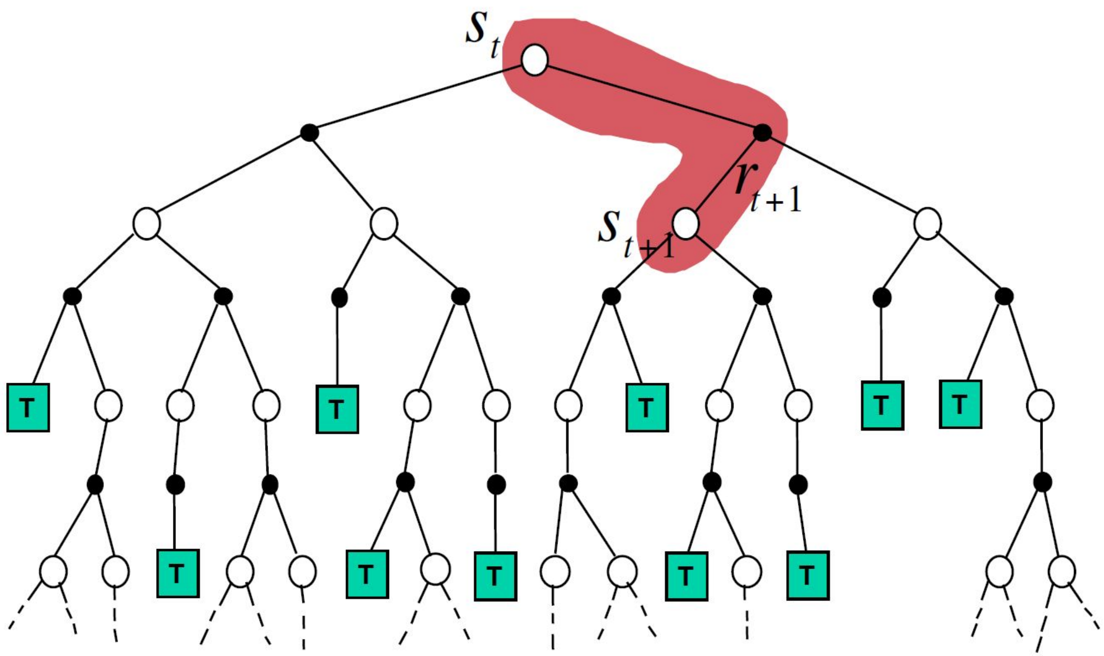
Монте-Карло обновление

$$V(s_t) \leftarrow V(s_t) + \alpha(R_t - V(s_t))$$



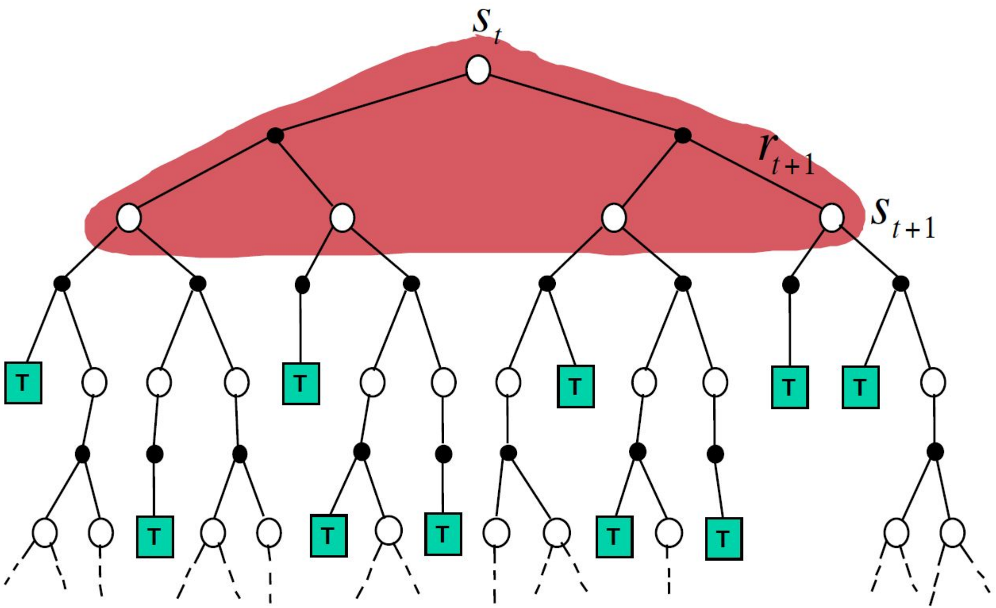
Обновление временных различий

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

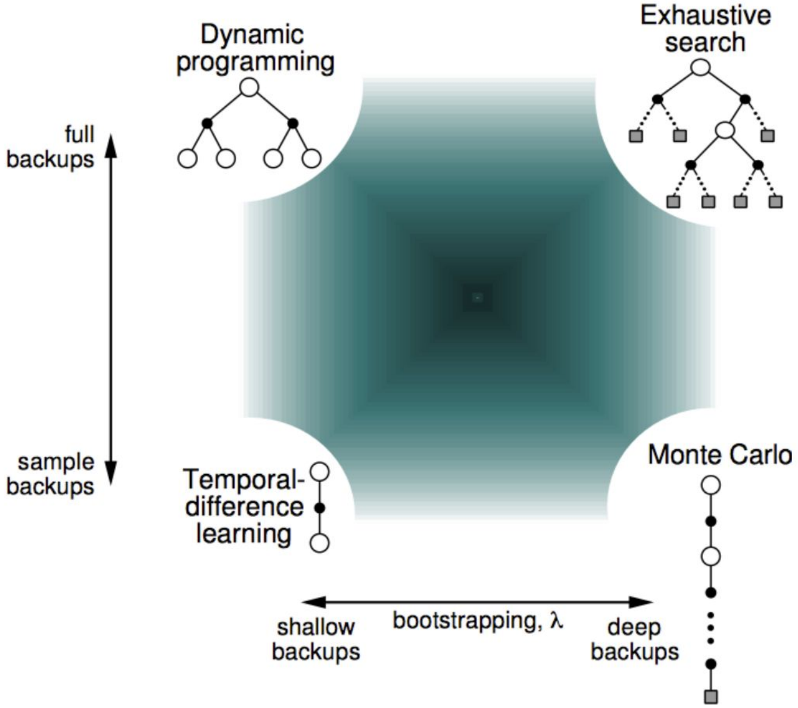


Обновление динамического программирования

$$V(s_t) \leftarrow \mathbb{E}_\pi[r_{t+1} + \gamma V(s_{t+1})]$$



Обобщенный подход к обучению с подкреплением

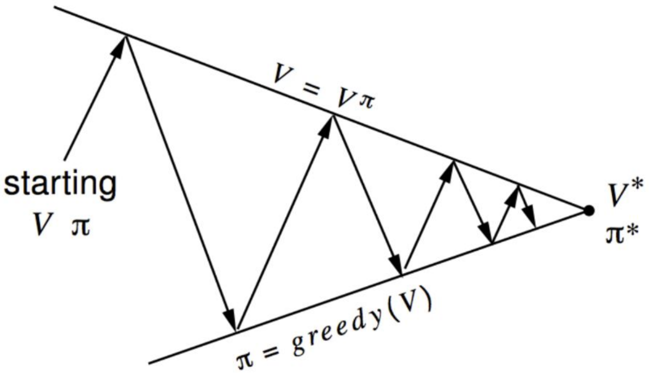


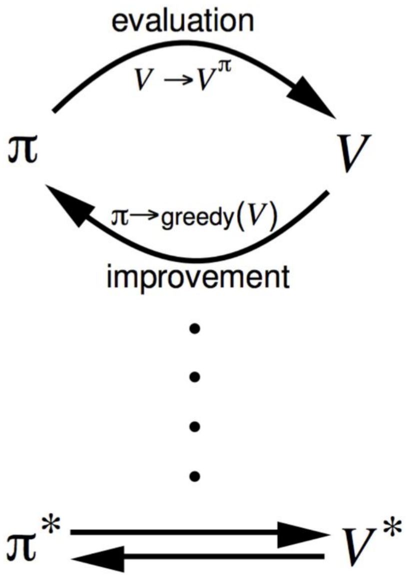
Q-обучение

Безмодельное предсказание по актуальному и отложенному опыту

- **Безмодельное предсказание (model-free prediction)**: оценка функции полезности по неизвестному МППР
- **Безмодельное управление (model-free control)**: оптимизация функции полезности по неизвестному МППР
- Обучение **по актуальному опыту** (on-policy):
  - “обучение по ходу дела”,
  - обучение стратегии π по опыту, полученном на основе π
- Обучение **по отложенному опыту** (off-policy):
  - “обучение на чужих ошибках”,
  - обучение стратегии π по опыту, полученному на основе μ

Обобщенные итерации по стратегиям





Оценка стратегии – вычисление V^π

Итеративная оценка стратегии

Улучшение стратегии – генерация $\pi' \geq \pi$

Жадное обновление стратегии

Использование функции полезности действий

- Жадное улучшение стратегии по $V(s)$ требует знания модели МППР:
$$\pi'(s) = \arg\max_{a \in A} \left(\mathcal{R}_s^a + \mathcal{P}_{ss'}^a V(s') \right)$$
- Жадное обновление стратегии по $Q(s, a)$ не требует знания модели:
$$\pi'(s) = \arg\max_{a \in A} Q(s, a)$$

Пример жадного выбора действий:

"Behind one door is tenure - behind the other is flipping burgers at McDonald's."

- Перед вами две двери
- Вы открываете левую дверь и получаете вознаграждение 0 $V(left) = 0$
- Вы открываете правую дверь и получаете вознаграждение +1 $V(right) = 1$
- Вы открываете правую дверь и получаете вознаграждение +2 $V(right) = 2$
- Вы открываете правую дверь и получаете вознаграждение +2 $V(right) = 2$
- $\vdots$
- Вы уверены, что выбирали лучшую дверь?

Исследование

- Простейшая идея, обеспечивающее постоянное исследование среды
- Все действия выбираются с ненулевой вероятностью
- С вероятностью $\epsilon - 1$выбираем действие жадно
- С вероятностью ϵ выбираем действие случайно

$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon, & \text{если } a^* = \arg\max_{a \in A} Q(s, a), \\ \epsilon/m, & \text{иначе} \end{cases}$$

Q-обновление

$$Q^{new}(s_t, a_t) \leftarrow (1 - \underbrace{\alpha}_{\text{learning rate}}) \cdot \underbrace{Q(s_t, a_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \underbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}_{\text{new value (temporal difference target)}}$$

Алгоритм

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until S is terminal

Спасибо за внимание!