Задача обучения с подкреплением

Взаимодействие компонентов в задаче обучения с подкреплением

Агент – тот, кто действует и обучается.
Среда – то, с чем взаимодействует агент.
Полное описание среды – задача.

Задача обучения с подкреплением

- Взаимодействие осуществляется в дискретные моменты времени t=0,1,2,3,…
- В каждый момент агент имеет доступ к состоянию среды $s_t \in S$ и выбирает действие $a_t \in A(s)$, используя стратегию π_t ($\pi_t(s,a)$ – вероятность выбрать a, находясь в состоянии s).
- В следующий момент агент получает подкрепление $r_{t+1} \in \Re$ и оказывается в новом состоянии s_{t+1}.
- Задача агента – максимизировать подкрепление за длительный период времени.

Граница агент-среда

- Внутренняя часть агента – то, над чем он имеет абсолютную власть.
- Остальное – среда.

Пример: управление биореактором

- Биореактор – чан с питательными веществами и бактериями, которые производят полезные химические вещества. Управляется перемешиванием и температурой.
- Действия – задающий сигнал температуры и скорости перемешивания, которые передаются исполнительным механизмам.
- Состояния – показания температурных и других датчиков, с задержкой и отфильтрованные; описание ингредиентов.
- Подкрепление - измеренные объемы полезного вещества, произведенного в реакторе за определённый интервал времени.

Пример: робот-манипулятор

- Имеется манипулятор, который должен перекладывать объекты, совершая плавные движения.
- Состояние – последние данные о положении и скорости узлов манипулятора.
- Действия – уровни напряжений, подаваемых на моторы.
- Подкрепление
  - +1 за успешно перемещённый объект
  - Небольшое отрицательное значение за каждое «дёргание».

Пример: робот-уборщик

- Робот должен собирать пустые банки в офисе.
  - Имеет сенсоры для обнаружения банок
  - Имеет манипуляторы для сбора банок
  - Работает от батарейки

- Рассматриваем задачу высокоуровневого выбора следующего действия:
  - Активно искать банки в течение некоторого периода времени,
  - Оставаться на месте и ждать, пока кто-нибудь принесет банку,
  - Отправляться на базу для перезарядки.

Пример: робот-уборщик

- Действия – 3, как описано ранее
- Состояние – уровень заряда батареи
- Подкрепление
  - 0 если ничего не происходит
  - >0 если робот собрал банку
  - <<0 если села батарея

Подкрепления и цели

- Агент добивается максимизации суммы подкреплений за продолжительный период.
- С помощью подкрепления мы сообщаем агенту, что мы от него хотим.
- Нельзя использовать подкрепление, чтобы навязать, как решать задачу.
- Вычисление подкрепления находится снаружи агента.

Возвраты

Цель агента – максимизировать ожидаемый возврат R_t.

- Для эпизодических задач

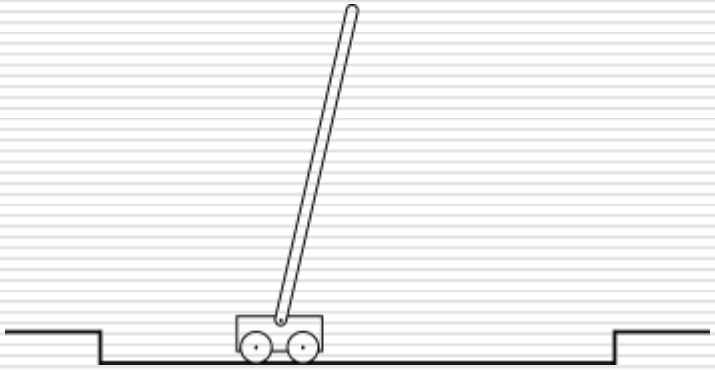
$$R_t = r_{t+1} + r_{t+2} + \cdots + r_T,$$

где T - последний шаг.

- Для непрерывных задач

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, 0 \leq \gamma < 1$$

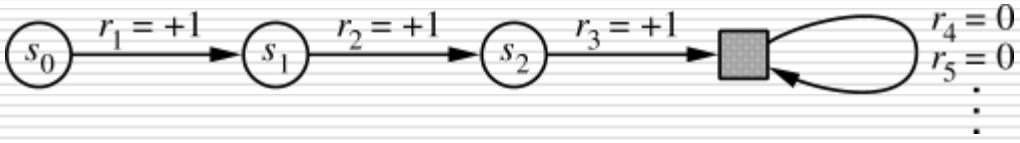
Пример – перевёрнутый маятник



- Эпизодическая задача – каждый эпизод – попытка удержать маятник. Вознаграждение +1 за каждый момент, пока маятник на весу.
- Непрерывная задача. Подкрепление -1 при каждом падении, 0 в остальные моменты – максимизируем время до падения.

Общая форма для эпизодических и непрерывных задач

- Наш агент проходит через последовательность эпизодов, поэтому, строго говоря, мы должны рассматривать последовательности состояний в моменты времени t эпизода i: $s_{t,i}$. Для простоты мы будем обычно вместо $s_{t,i}$ писать s_t.
- Для эпизодических задач введём поглощающее состояние:



- Тогда мы можем ввести возврат как $R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$,

где T может быть равно ∞ или $\gamma = 1$, но не одновременно.

Марковский сигнал

- Состояние - это та информация, которая доступна агенту.
  - Данные измерений на настоящий момент
  - Память о предыдущих событиях
- Состояние, которое сохраняет всю существенную информацию, называется Марковским.

Марковский сигнал

В общем случае динамика среды определяется (1):

$$Pr\left\{s_{t+1}=s', r_{t+1}=r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\right\},$$

В случае, если динамика среды определяется только текущим состоянием, имеем (2):

$$Pr\left\{s_{t+1}=s', r_{t+1}=r \mid s_t, a_t\right\},$$

Сигнал состояния является Марковским тогда и только тогда, когда (1) эквивалентно (2) для всех *s′*,*r* и историй $s_t, a_t, r_t, \ldots, r_1, s_0, a_0$

Пример: состояние в задаче перевернутого маятника

- Марковский сигнал для идеализированной задачи:
  - Точные положения и скорости тележки
  - Точный угол и скорость маятника
- В реальной задаче
  - Задержки и шумы в сенсорах
  - Изгибы механизмов
  - Температура и температурные расширения
  - Аэродинамика

Пример: Покер

- Сигнал состояния:
  - Разный для каждого игрока
  - Свои карты
  - Ставки и число замен, которые сделали другие игроки.
  - Кто любит блефовать, а кто играет консервативно?
  - Как можно трактовать выражения лиц?
  - Как влияет на игроков предыдущий выигрыш (проигрыш)?
  - …

Марковский процесс принятия решений

Задача обучения с подкреплением, удовлетворяющая свойству Марковости, называется Марковский процесс принятия решений.

- Переходные вероятности

$$\mathcal{P}^a_{ss'} = Pr\left\{s_{t+1} = s' \mid s_t = s, a_t = a\right\}.$$

- Ожидаемые подкрепления

$$\mathcal{R}^a_{ss'} = E\left\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\right\}.$$

Пример: робот-уборщик

- Состояния: уровень заряда $high/low$.
- Возможные действия:

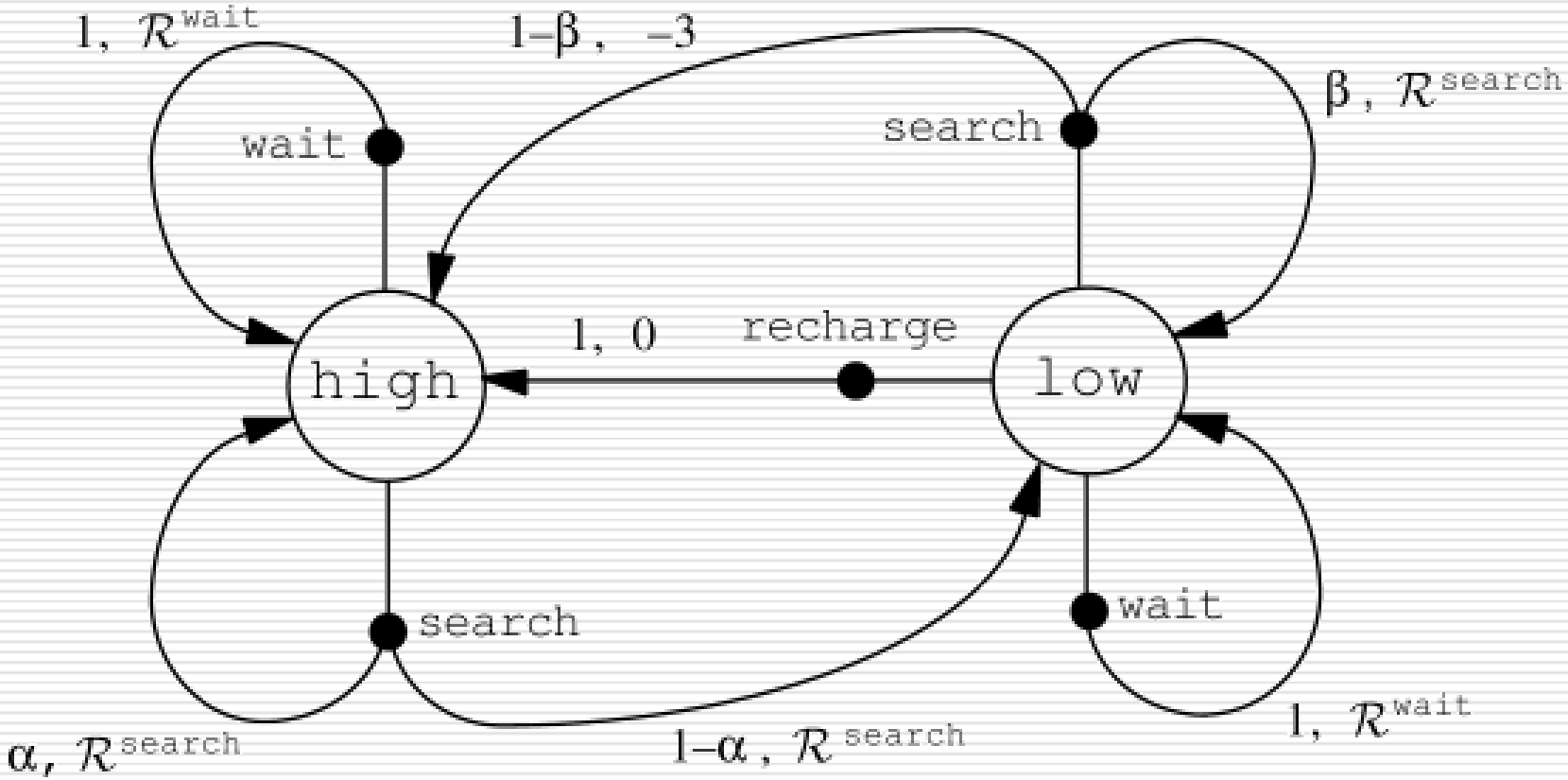
  $A(high) = \{search, wait\}$;

  $A(low) = \{search, wait, recahrge\}$.
- Переходы между состояниями:
  - при поиске в состоянии $high$ с вероятностью α остаёмся в этом состоянии;
  - при поиске в состоянии low с вероятностью β остаёмся в этом состоянии.
- Подкрепление:
  - -3 если батарея села;
  - 1 за каждую найденную банку, вероятность найти банку:
    - в режиме поиска - R^{search};
    - в режиме ожидания - R^{wait}.

Пример: робот-уборщик

$s = s_t$	$s' = s_{t+1}$	$a = a_t$	$P^a_{ss'}$	$\mathcal{R}^a_{ss'}$
High	High	Search	α	$\mathcal{R}^{search}$
High	Low	Search	$1-\alpha$	$\mathcal{R}^{search}$
Low	High	Search	$1-\beta$	-3
Low	Low	Search	β	$\mathcal{R}^{search}$
High	High	Wait	1	$\mathcal{R}^{wait}$
High	Low	Wait	0	$\mathcal{R}^{wait}$
Low	High	Wait	0	$\mathcal{R}^{wait}$
Low	Low	Wait	1	$\mathcal{R}^{wait}$
Low	High	Recharge	1	0
Low	Low	Recharge	0	0

Пример: робот-уборщик



Функции ценности

Ценность состояния *s* для стратегии π есть ожидаемое подкрепление, которое агент получит, начиная в *s* и действуя далее согласно π:

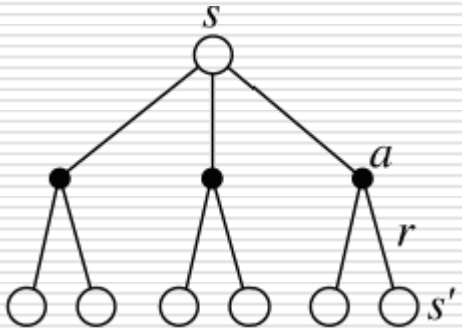
$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \;\middle|\; s_t = s\right\}$$

Аналогично ценность действия a:

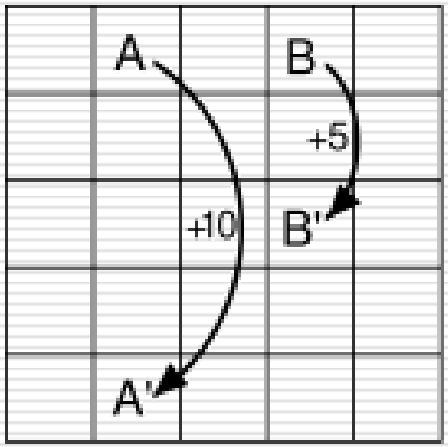
$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \;\middle|\; s_t = s, a_t = a\right\}.$$

Уравнения Беллмана

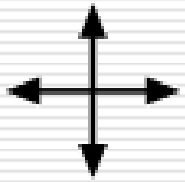
$$
\begin{aligned}
V^{\pi}(s) &= E_{\pi}\{R_t | s_t = s\} \\
&= E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s \right\} \\
&= E_{\pi}\left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, s_t = s \right\} \\
&= \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, s_{t+1} = s' \right\} \right] \\
&= \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^{\pi}(s') \right],
\end{aligned}
$$



Функция ценности состояний - пример



$\gamma = 0.9$



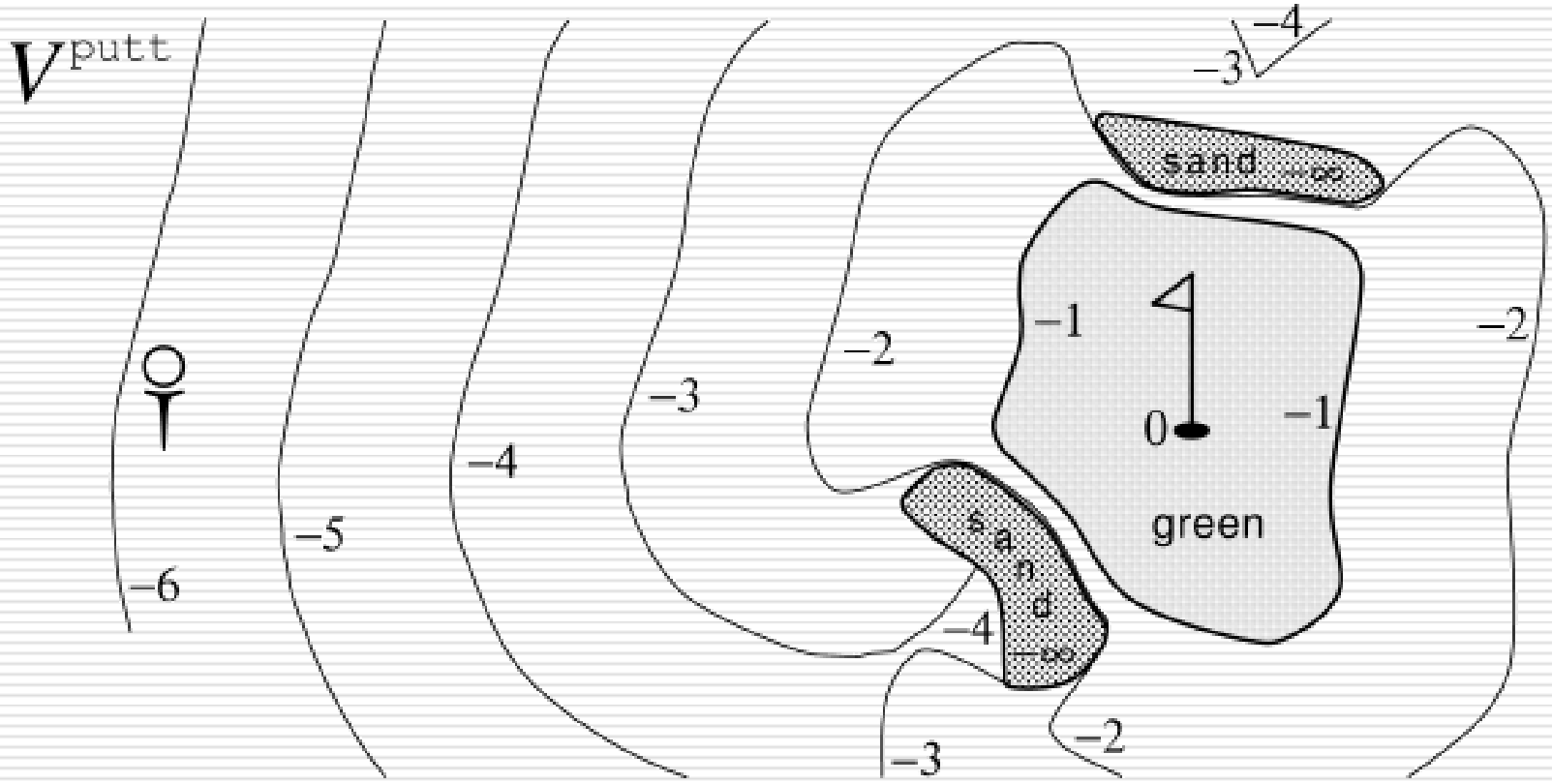
Actions

$V^{Random}(s)$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Если пытаемся сойти с поля, то остаёмся на месте и получаем подкрепление r=-1.

Функция ценности действий - пример



Оптимальные функции ценности

Скажем, что стратегия π лучше или равна, чем стратегия π', если её ожидаемый возврат больше или равен возврату при действиях по стратегии π'. Или $\pi \geq \pi' \Leftrightarrow V^{\pi}(s) \geq V^{\pi'}(s)$.

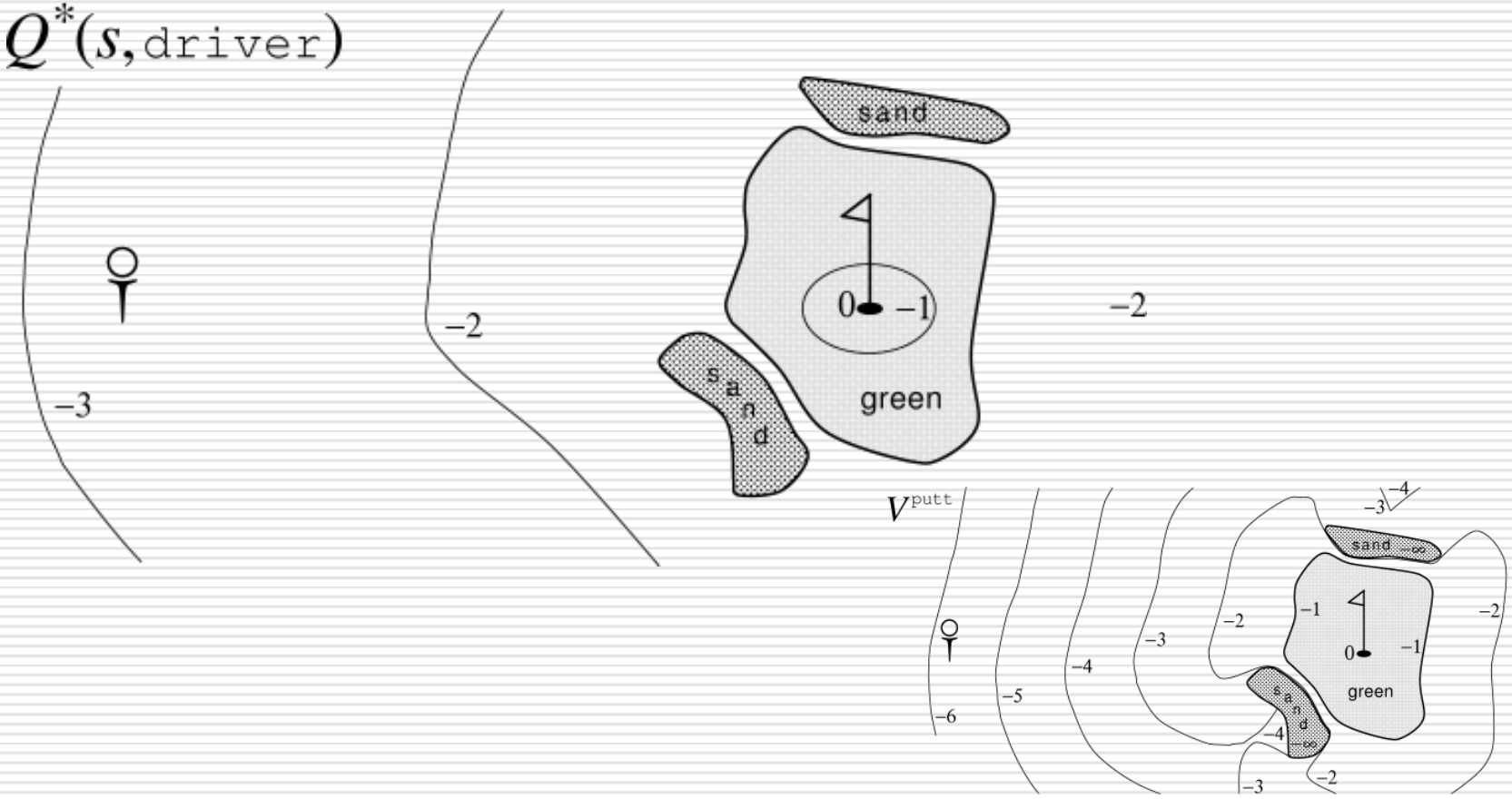
Существует оптимальная (самая лучшая) стратегия π^*.

Тогда оптимальные функции ценности есть:

$$V^*(s) = \max_{\pi} V^{\pi}(s),\ Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a),$$

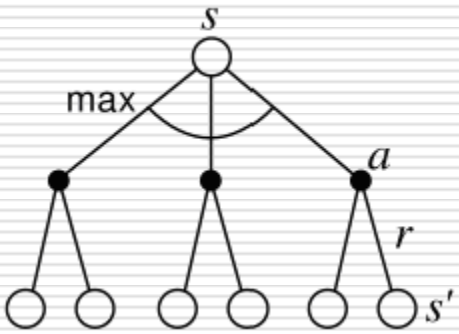
$$Q^*(s,a) = E\left\{r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a\right\}.$$

Оптимальная функция ценности - пример

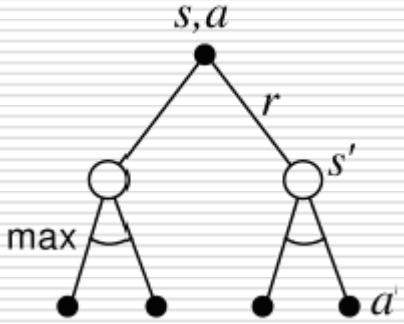


Условия оптимальности Беллмана

$$
\begin{aligned}
V^*(s) &= \max_{a \in \mathcal{A}(s)} Q^{\pi^*}(s,a) = \max_a E_{\pi^*}\left\{ R_t \;\middle|\; s_t = s, a_t = a \right\} \\
&= \max_a E_{\pi^*}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \;\middle|\; s_t = s, a_t = a \right\} \\
&= \max_a E_{\pi^*}\left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \;\middle|\; s_t = s, a_t = a \right\} \\
&= \max_a E\left\{ r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a \right\} \\
&= \max_{a \in \mathcal{A}(s)} \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^*(s') \right].
\end{aligned}
$$



$$
\begin{aligned}
Q^*(s,a) &= E\left\{ r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \;\middle|\; s_t = s, a_t = a \right\} \\
&= \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q^*(s', a') \right].
\end{aligned}
$$



Оптимальные стратегии

- Если мы знаем $V^*(s)$:

  Выбираем действие, которое приводит к следующему состоянию s' с максимальным значением $V^*(s')$.

- Если мы знаем $Q^*(s,a)$:

  Выбираем действие a^*, для которого $Q^*(s,a)$ максимально:

  $$a^* = \text{argmax}_a Q^*(s,a).$$

- Оптимальной является жадная по отношению к V^*/Q^* стратегия.

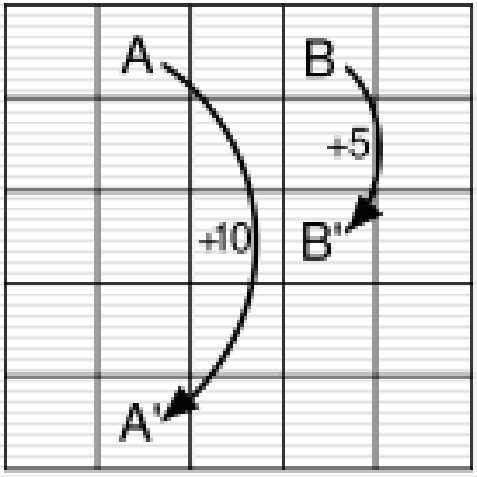
Пример: Условия оптимальности Беллмана для задачи робота-уборщика

- У нас 2 состояния, соответственно 2 уравнения:

$$
\begin{aligned}
V^*(\mathtt{h}) &= \max \left\{ \begin{array}{l} \mathcal{P}^{\mathtt{s}}_{\mathtt{hh}}[\mathcal{R}^{\mathtt{s}}_{\mathtt{hh}} + \gamma V^*(\mathtt{h})] + \mathcal{P}^{\mathtt{s}}_{\mathtt{hl}}[\mathcal{R}^{\mathtt{s}}_{\mathtt{hl}} + \gamma V^*(\mathtt{l})], \\ \mathcal{P}^{\mathtt{w}}_{\mathtt{hh}}[\mathcal{R}^{\mathtt{w}}_{\mathtt{hh}} + \gamma V^*(\mathtt{h})] + \mathcal{P}^{\mathtt{w}}_{\mathtt{hl}}[\mathcal{R}^{\mathtt{w}}_{\mathtt{hl}} + \gamma V^*(\mathtt{l})] \end{array} \right\} \\
&= \max \left\{ \begin{array}{l} \alpha[\mathcal{R}^{\mathtt{s}} + \gamma V^*(\mathtt{h})] + (1-\alpha)[\mathcal{R}^{\mathtt{s}} + \gamma V^*(\mathtt{l})], \\ 1[\mathcal{R}^{\mathtt{w}} + \gamma V^*(\mathtt{h})] + 0[\mathcal{R}^{\mathtt{w}} + \gamma V^*(\mathtt{l})] \end{array} \right\} \\
&= \max \left\{ \begin{array}{l} \mathcal{R}^{\mathtt{s}} + \gamma[\alpha V^*(\mathtt{h}) + (1-\alpha) V^*(\mathtt{l})], \\ \mathcal{R}^{\mathtt{w}} + \gamma V^*(\mathtt{h}) \end{array} \right\}.
\end{aligned}
$$

$$
V^*(\mathtt{l}) = \max \left\{ \begin{array}{l} \beta\mathcal{R}^{\mathtt{s}} - 3(1-\beta) + \gamma[(1-\beta) V^*(\mathtt{h}) + \beta V^*(\mathtt{l})] \\ \mathcal{R}^{\mathtt{w}} + \gamma V^*(\mathtt{l}), \\ \gamma V^*(\mathtt{h}) \end{array} \right\}.
$$

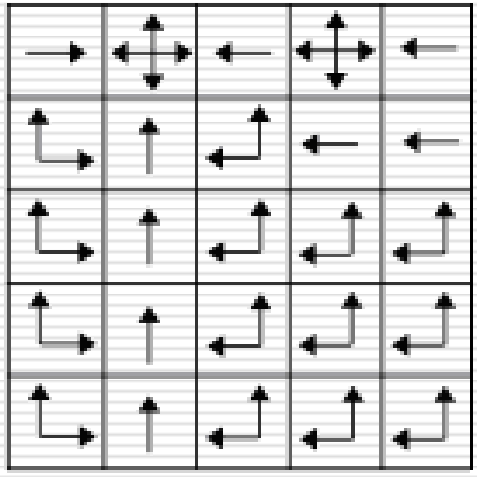
Оптимальные функция ценности и стратегия - пример



V^*

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

π^*



Когда мы можем решить уравнения Беллмана

1. Нам точно известна динамика среды
2. У нас достаточно ресурсов для решения соответствующей системы уравнений
3. Марковский процесс принятия решений

Оптимальность и приближение

- Для многих практических задач оптимальная стратегия не может быть вычислена точно
  - Нет точной модели
  - Недостаточно производительности
  - Недостаточно памяти
    - Табличное представление функций для небольших задач
    - Использование аппроксимации при большом числе состояний