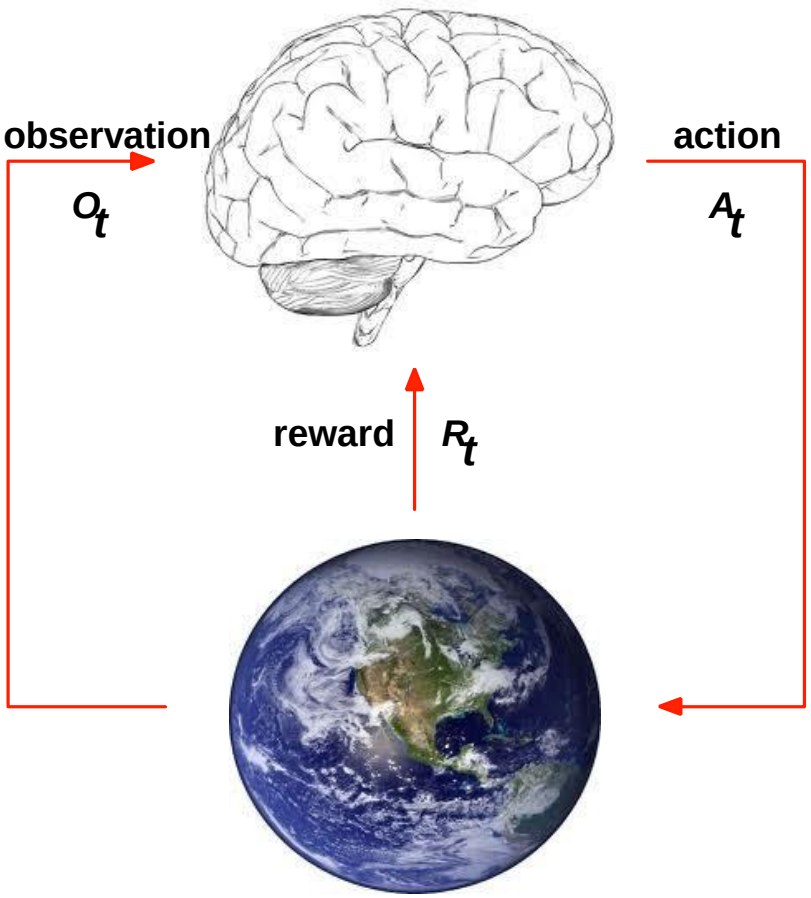
RL: Recap

План

Взаимодействие среды и агента



- В каждый момент времени t агент:
  - Выполняет действие a_t
  - Получает наблюдение o_t
  - Получает скалярное вознаграждение r_t
- В каждый момент времени t среда:
  - Реагирует на действие a_t
  - Выдает следующее наблюдение o_{t+1}
  - Выдает скалярное вознаграждение r_{t+1}
- Переход к шагу t+1

Вознаграждение (reward). Суммарное вознаграждение (отдача, return)

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

- Дисконтирующий множитель $\gamma \in [0, 1]$ - оценка значений будущих вознаграждений
- Значение получаемых вознаграждений после *k + 1* шагов - $\gamma^k r$
- Моментальное вознаграждение важнее отложенных будущих:
  - $\gamma \sim 0$ – близорукий агент
  - $\gamma \sim 1$ – дальнозоркий агент

Строение RL агента

- Агент RL может включать один или несколько из этих компонентов:
  - Стратегия (policy): функция поведения агента
  - Функция полезности (value function): оценка насколько хорошо каждое состояние и/или действие
  - Модель (model): представление агента о среды

Стратегия

- Стратегия (политика) - это функция поведения агента. Обычно это отображения состояния в действие
- Она представляет собой отображение из состояния в действие, например
  - ✓ Детерминированная политика: $a = \pi(s)$
  - ✓ Стохастическая политика: $\pi(a|s) = P[a_t = a|s_t = s]$

Функция полезности

- Функция полезности - это предсказание будущего вознаграждения и используется для оценки состояния
- Используется для оценки на сколько состояние является ценным и, следовательно, для выбора между действиями, например

$$V^{\pi} = E_{\pi}[r_t + \Upsilon r_{t+1} + \Upsilon^2 r_{t+2} + \cdots | s_t = s] = E_{\pi}[G_t | s_t = s]$$

- Функция полезности действия

$$Q^{\pi} = E_{\pi}[r_t + \Upsilon r_{t+1} + \Upsilon^2 r_{t+2} + \cdots | s_t = s, a_t = a] = E_{\pi}[G_t | s_t = s, a_t = a]$$

Модель

- Модель предсказывает, что произойдет в среде в следующий момент времени

Примеры:

- Модель переходов P предсказывает следующее состояние

$$P_{ss'}^{a} = P\left[s_{t+1} = s' | s_t = s, a_t = a\right]$$

- Модель вознаграждений R предсказывает следующее (мгновенное) вознаграждение:

$$R_s^a = E\left[r_t | s_t = s, a_t = a\right]$$

Типизация RL агентов

- Оценивающие функцию полезности (value base)
  - Стратегия не представлена явно
  - Функция полезности
- Оценивающие стратегию (policy base)
  - Стратегия
  - Функция полезности не вычисляется явно
- Актор-критик (actor-critic)
  - Стратегия
  - Функция полезности

Типизация RL агентов

- Безмодельные (model free)
  - Стратегия и/или функция полезности,
  - Нет модели
- Основанные на модели (model based):
  - Стратегия и/или функция полезности
  - Строят модель

Марковский процесс принятия решений

- Марковский процесс принятия решения (МППР, MDP) моделирует взаимодействие агента и среды
- Предполагается что среда полностью наблюдаема (fully observable)
- Текущее состояние полностью характеризует весь процесс взаимодействия
- Почти все задачи RL могут быть сведены к задаче с MDP
  - Оптимальное управление — MDP непрерывным множеством состояний и действий
  - Частично наблюдаемы среды могут быть сведены к MDP
  - Игровые автоматы — пример MDP с одним состоянием

Марковское свойство

Будущее не зависит от прошлого и определяется только настоящим

Определение

- *Состояние s_t называется марковским если и только если*

$$P[s_{t+1}|s_t] = P[s_{t+1}|s_1, s_2, \ldots, s_t]$$

- Текущее состояние содержит всю информация из истории взаимодействия
- Если есть текущее состояние, история далее может не учитываться
- Состояние содержит достаточно статистики для определения будущего

Матрица переходов

❑ Вероятность перехода для марковского состояния s в следующее состояние s’ определяется как:

$$P_{ss'} = P[s_{t+1} = s' | s_t = s]$$

Матрица переходов P определяет вероятности переходов между всеми возможными состояниями

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \cdots & \cdots & \cdots \\ P_{n1} & .. & P_{nn} \end{bmatrix}$$

Каждая строчка матрицы в сумме дает 1

Марковский процесс (принятия решений)

- Марковский процесс (Markov process, Markov chain) - это случайный процесс без памяти, т.е. Последовательность случайных состояний s1, s2, ..., обладающая свойством Маркова.

Определение

Марковский процесс (или цепь Маркова) - это пара <*S,P*>, где

- S - (конечное) множество состояний
- P - матрица вероятностей перехода между состояниями,

$$P_{ss'} = P[s_{t+1} = s' | s_t = s]$$

Марковский процесс вознаграждения

- Марковский процесс вознаграждения - это марковская цепь с дополнительными значениями вознаграждений за переходы

Определение

Марковский процесс вознаграждений - это тройка <*S,P, R, γ*>, где

- S - (конечное) множество состояний
- P - матрица вероятностей перехода между состояниями,

$$P_{ss'} = P[s_{t+1} = s' | s_t = s]$$

- R – функция вознаграждения $R_s = E[r_{t+1} | s_t = s]$
- $\gamma \in [0,1]$ – дисконтирующий множитель

Марковский процесс принятия решений

Марковский процесс принятия решений – это марковский процесс вознаграждения для действий. Он описывает взаимодействие со средой с марковским состояниями

❑ Определение

Марковский процесс принятия решений – это кортеж <*S,A,P,R,ɣ*>, *где*

- *S – конечное множество состояний*
- *A – конечное множество действий*
- *P – матрица перехода*

$$P_{ss'}^{a} = P[s_{t+1} = s' | s_t = s, a_t = a]$$

- R – функция вознаграждения $R_s^a = E[r_{t+1} | s_t = s, a_t = a]$
- $\gamma \in [0, 1]$ - дисконтирующий множитель

Уравнение Беллмана для MRP

Функцию полезности V(s) можно представить в виде суммы двух слагаемых:

- немедленное вознаграждение r_{t+1}
- Дисконтированное значение следующего состояния $\gamma V(s_{t+1})$:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = r_{t+1} + \gamma(r_{t+2} + \gamma r_{t+3} + \cdots) = r_{t+1} + \gamma G_{t+1}$$

$$V(s) = E[G_t | s_t = s] = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s] =$$
$$E[r_{t+1} + \gamma(r_{t+2} + \gamma r_{t+3} + \cdots) | s_t = s] = E[r_{t+1} + \gamma G_{t+1} | s_t = s] = E[r_{t+1} + \gamma V(s_{t+1}) | s_t = s]$$

Уравнение Беллмана в матричной форме

$$V = R + \gamma PV$$

Где V – это вектор колонка с одной компонентой на состояние

$$\begin{pmatrix} V(1) \\ \dots \\ V(n) \end{pmatrix} = \begin{pmatrix} R(1) \\ \dots \\ R(n) \end{pmatrix} + \gamma \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \dots & \dots & \dots \\ P_{n1} & \dots & P_{nn} \end{pmatrix} \begin{pmatrix} V(1) \\ \dots \\ V(n) \end{pmatrix}$$

$$V(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} V(s')$$

Стратегия

❑ Определение

Стратегией π будем называть вероятностное распределение на множестве действий действиям при текущем состоянии s:

$$\pi(a|s) = P[a_t = a|s_t = s]$$

- *Стратегия полностью определяет поведение агента*
- *MDP стратегии зависят от текущего состояния (не от истории).*
- *Стратегии стационарны (не зависят от времени)*

$$a_t \sim = \pi(\cdot\,|s_t), \forall t > 0$$

Стратегия

- Пусть дан MDP $\mathrm{M} = \langle S, A, P, R, \gamma \rangle$ и стратегия π.
- *Последовательность состояний s1,s2,... является марковским процессом* $\langle S, P^{\pi} \rangle$
- *Последовательность состояний и вознаграждений s1,r2,s2,... представляет собой марковский процесс вознаграждения* $\langle S, P^{\pi}, R^{\pi}, \gamma \rangle$

$$P_{ss'}^{\pi} = \sum_{a \in A} \pi\ (a|s) P_{ss'}^{a},$$

$$R_{ss'}^{\pi} = \sum_{a \in A} \pi\ (a|s) R_{ss'}^{a},$$

Стратегия

- Определение
  - *Функция полезности состояний MDP $V^{\pi}(s)$ - это математическое ожидание отдачи, начиная с состояния s, при выполнении политики π:*

$$V^{\pi}(s) = E_{\pi}[G_t | s_t = s]$$

- Определение
  - *Функция полезности действия MDP $Q^{\pi}(s, a)$ - это математическое ожидание отдачи, начиная с состояния s, выбранного действия a, при выполнении стратегии π*

$$Q^{\pi}(s, a) = E_{\pi}[G_t | s_t = s, a_t = a]$$

Оптимальная функция полезности

- *Определение*

Оптимальная функция полезности состояний $V^*(s)$ - это максимальное значение функции полезности по всем стратегиям

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

Оптимальная функции полезности действия $Q^*(s,a)$ - это максимальное значение функции полезности действия по всем стратегиям

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

- *Оптимальные стратегии характеризуют лучшее поведение в MDP*
- *Говорят, что задача MDP решена, когда найдена оптимальная функция полезности*

Оптимальная стратегия

Определим частичный порядок на множестве стратегий:

$$\pi \geq \pi' \; if \; V^{\pi}(s) \geq V^{\pi'}(s), \forall s$$

- *Теорема*

Для любого MDP

- *Существует оптимальная стратегия,* π_* *которая лучше или равна всем другим стратегиям* $\pi_* \geq \pi, \forall \pi$
- *Все оптимальные стратегии доставляют оптимум функции (ценности) состояния - действия* $V^{\pi^*}(s) \geq V^{\pi}(s)$ $Q^{\pi^*}(s,a) \geq Q^{\pi}(s,a)$

Поиск оптимальной стратегии

Оптимальная стратегия π_* *может быть найдена максимизацией функцией полезности действий* $Q^{\pi^*}(s,a)$ *:*

$$\pi^*(a|s) = \begin{cases} 1, \text{если } a = arg\max\limits_{a \in A} Q^*(s,a) \\ 0, \text{иначе} \end{cases}$$

- *Для любого MDP существует оптимальная детерменированная стратегия*
- *Если известна* $Q^*(s,a)$*, то мы получаем одновременно и оптимальную стратегию*

Уравнение Беллмана для MDP

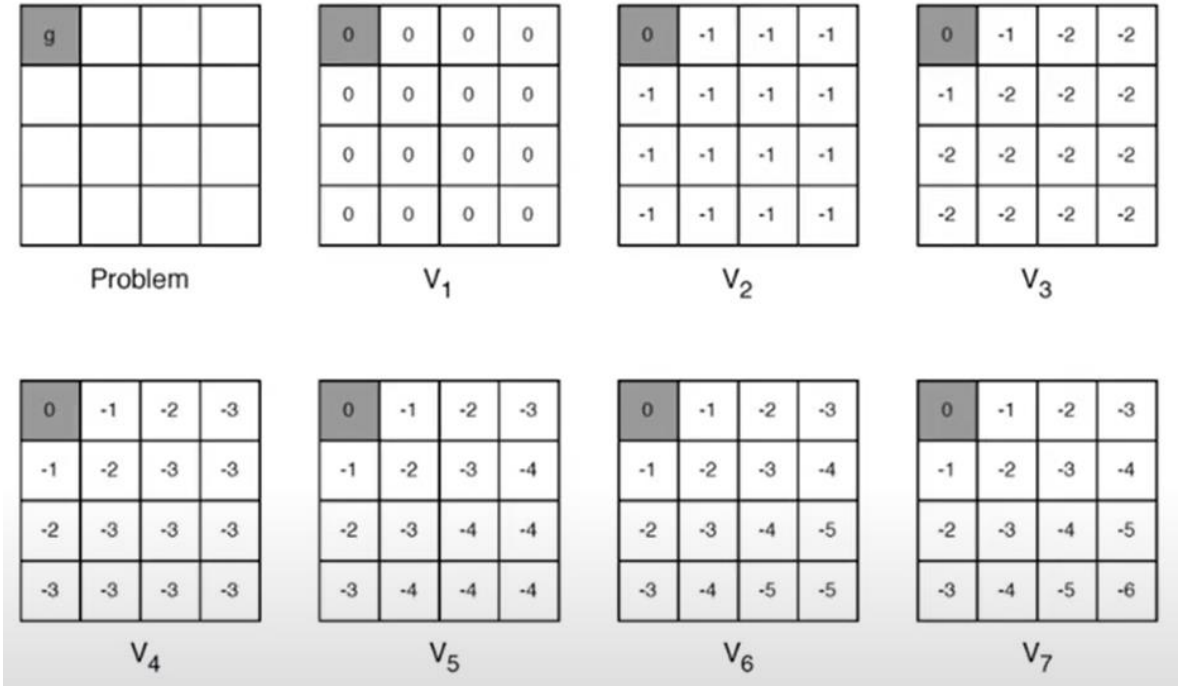
- *Функция полезности состояния может быть разложена на немедленное вознаграждение плюс дисконтированная стоимость следующего состояния*

$$V^{\pi}(s) = E_{\boldsymbol{\pi}}[r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s]$$

- *Аналогично можно разложить функцию полезности действия,*

$$Q^{\pi}(s, a) = E_{\boldsymbol{\pi}}[r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) | s_t = s, a_t = a]$$

Итерационные алгоритмы



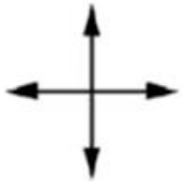
Итерационная оценка стратегии

- Задача: оценить текущую стратегию π
- Решение: итеративное применение уравнения Беллмана:

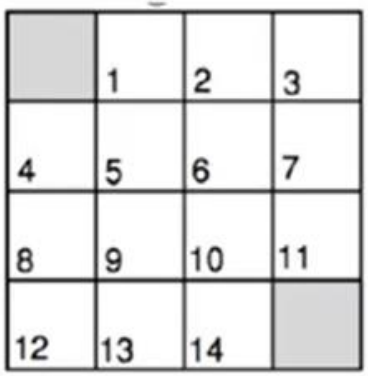
$$V^1 \to V^2 \to V^3 \to \cdots \to V^n, \qquad V(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} V(s')$$

- Использование синхронных шагов:
  - Для каждой итерации *k + 1:*
  - Для каждого состояния $a \in A$
  - Обновить $V^{k+1}(s)$ по $V^k(s')$, где ***s'*** – следующее состояние после *s*
- Можно использовать асинхронные шаги
- Сходится к истинным значениям V^π

Пример: клеточный мир



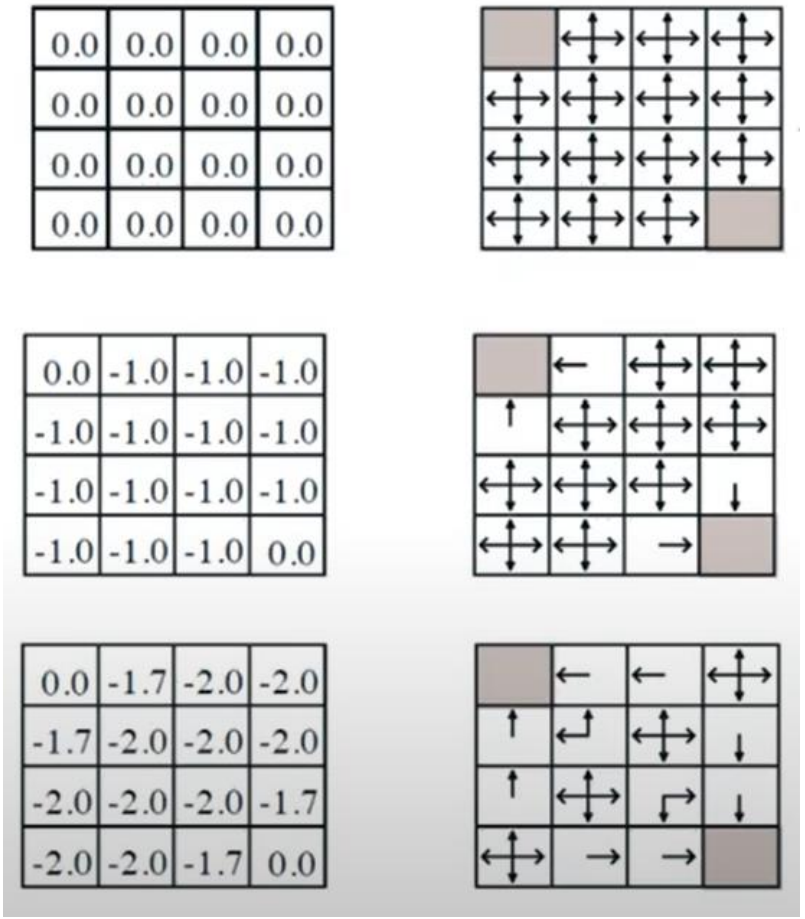
actions



$r = -1$
on all transitions

Пример: клеточный мир

$$V^{k+1} = R^{\pi} + \Upsilon P^{\pi} V^{k}$$



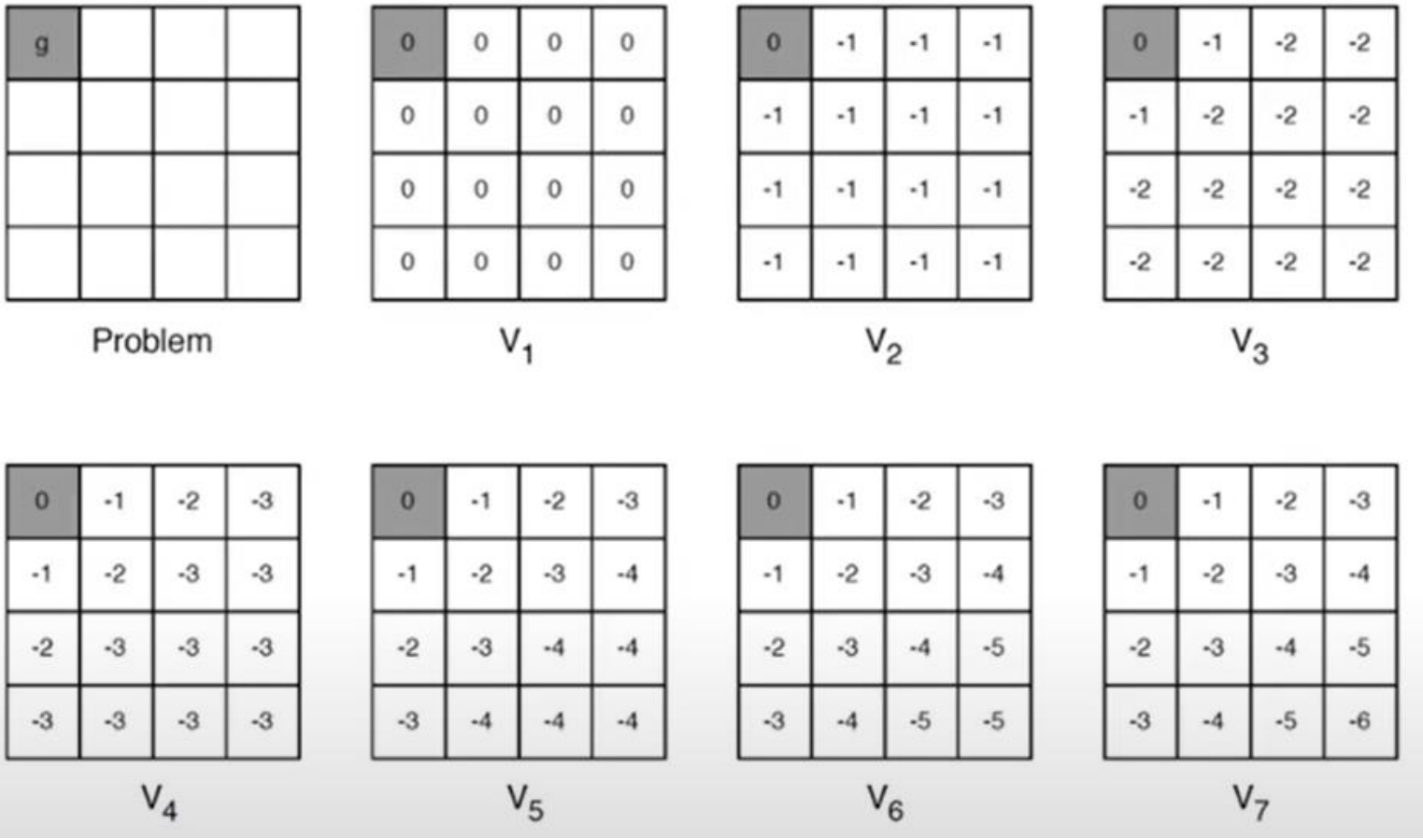
Детерменированные итерации по полезностям

- Пусти мы знаем решение для подзадачи $V^*(s')$,
- Тогда мы можем найти решение за один шаг

$$V^*(s) \leftarrow \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V^*(s') \right)$$

- Идея итераций по ценностям – применять эти обновления рекурсивно
- Интуиция: начать с конечных вознаграждений и двигаться назад

Пример: кратчайший путь



Базовые алгоритмы

- Монте Карло
- Временные различия
- Q обучение
- SARSA

Среднее приращение (МК)

- Средние значения могут быть вычислены последовательно:

$$\mu_k = \frac{1}{k}\sum_{i=1}^{k} x_i = \frac{1}{k}\left(x_k + \sum_{i=1}^{k-1} x_i\right) =$$

$$= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) \Rightarrow$$

$$\Rightarrow \mu_k = \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)$$

Монте Карло для приращений

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

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

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

- Для нестационарных задач м.б. полезно отслеживать текущее среднее:

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

MC и TD

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

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

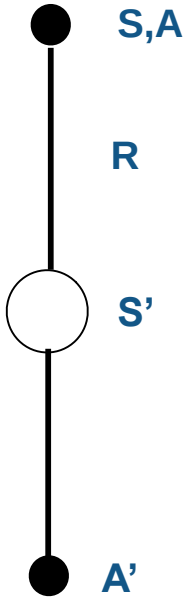
- Подход TD:
  - Обновляем на основе ожидаемой отдачи $r_t + \gamma V(s_{t+1})$

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

  - $r_t + \gamma V(s_{t+1})$ называется TD показателем
  - $\delta_t = r_t + \gamma V(s_{t+1}) - V(s\_t)$ называется TD ошибкой

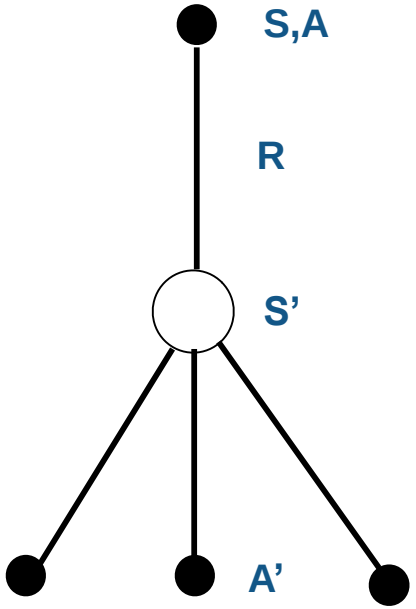
- TD приближает значения на основе предыдущего приближения

Обновление функции полезности по SARSA



$$\mathrm{Q}(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a))$$

Алгоритм управления с Q обучением (SARSAMAX)



$$\mathrm{Q}(s,a\,) \leftarrow Q(s,a\,) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a\,))$$