System thinking
Lecture 14. Reinforcement learning

Oleksii Ignatenko

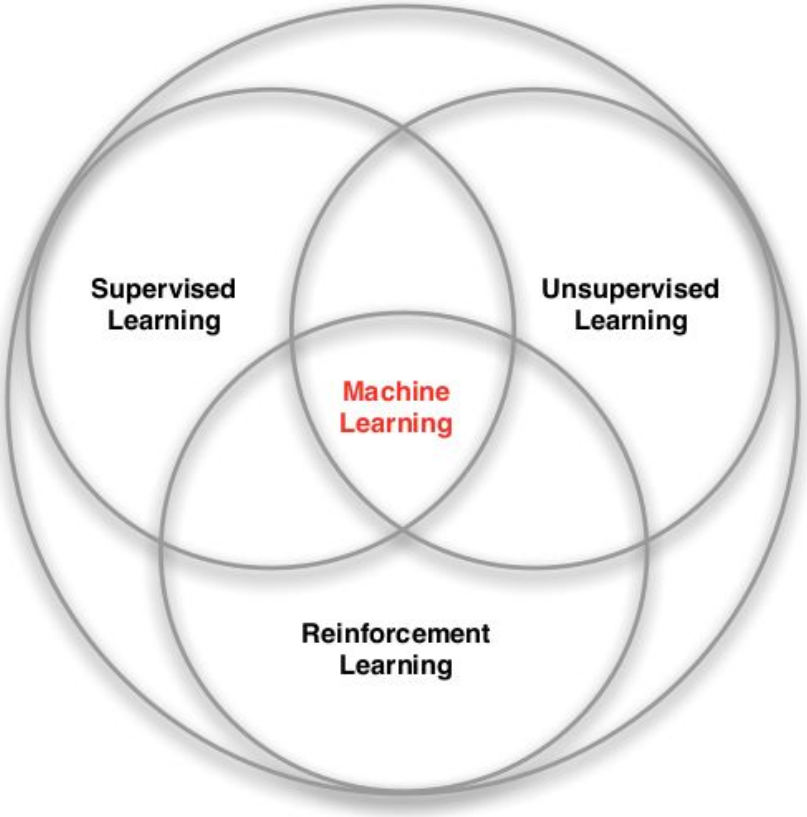
Plan

1. Reinforcement learning idea
2. Q-learning
3. MDP and idea of value function
4. Value, policy based methods
5. VIA and its implementation
6. Last Lab task

Навчання з підкріпленням

Alice … went on “Would you please tell me, please, which way I ought to go from here?” “That depends a good deal on where you want to get to,” said the Cat. “I don’t much care where -” said Alice. “Then it doesn’t matter which way you go,” said the Cat.
Lewis Carroll (1832-1898) Alice’s Adventures in Wonderland, 1865

Машинне навчання. Навчання з підкріпленням



В чому особливість:

1. Немає розмічених даних, лише виграш.
2. Зворотній зв’язок затримується.
3. Час важливий для рішень.
4. Дії агента впливають на його наступні виміри

Виграш

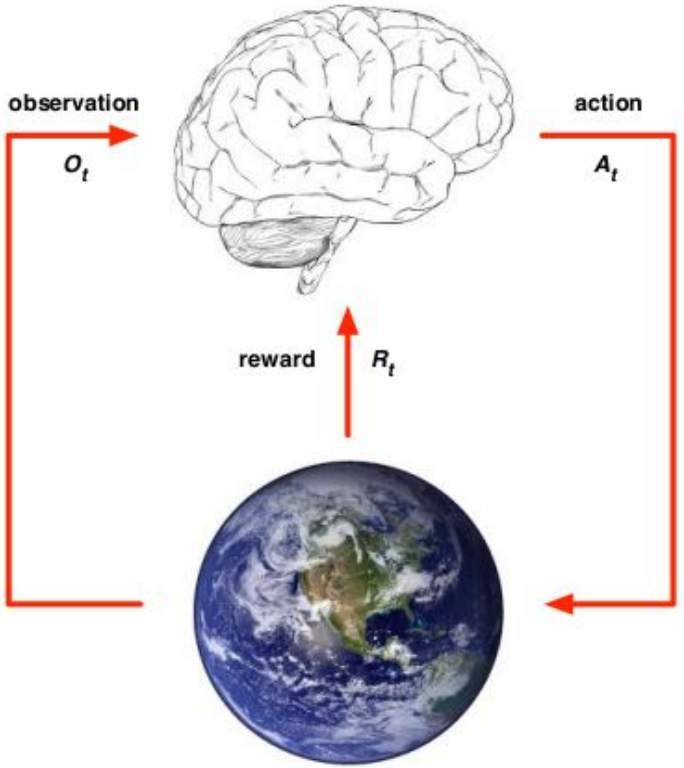
Виграш R(t) це салярний фідбек, який показує наскільки успішно поводить себе агент на кроці t.

Завдання агента - максимізувати сумарний виграш.

Гіпотеза Навчання з підкріпленням.

Будь-яку мету можна описати як максимізацію очікуваного сумарного виграшу.

Загальна модель агент-середовище



На кожному кроці агент виконує дію, отримує спостереження і виграш.

Середовище. Отримує дію, генерує спостереження і винагороду.

Історія

Історія - це послідовність спостережень, дій і виграшів.

$$H_t = O_1, R_1, A_1, ..., A_{t-1}, O_t, R_t$$

Стан - це інформація, достатня для того, щоб зрозуміти, що станеться далі.

Формально, стан - це функція історії:

$$S_t = f(H_t)$$

Властивість Маркова

Інформаційний стан (або стан Маркова) містить всю корисну інформацію з історії

Стан є Марківським тоді і тільки тоді, коли

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

Або іншими словами, якщо майбутнє не залежить від минулого при заданому теперішньому

$$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$

Компоненти агента, який навчається

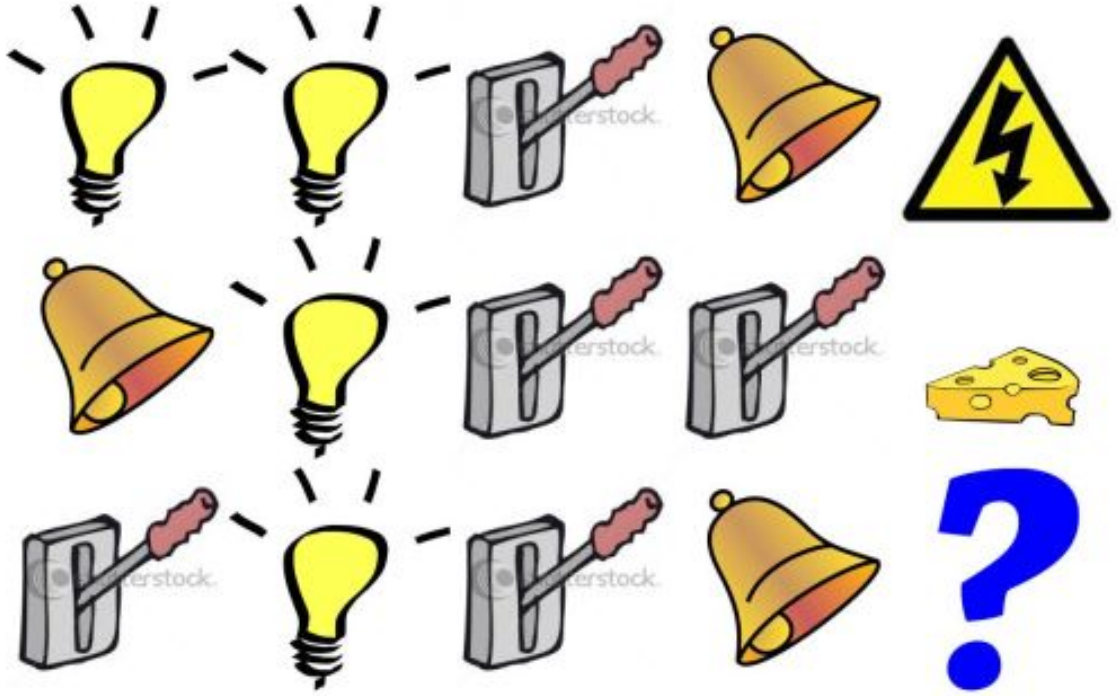
Агент може мати один або більше компонентів:

- Policy: функція поведінки агента

- Value function: оцінка успішності для кожного стану та/або дії

- Model: представлення середовища


WILL PRESS
LEVER
FOR
FOOD


?

Політика

Політика - це відображення простору станів у простір дій, тобто:

Детерміністична політика $a = \pi(s)$

Стохастична політика $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

Функція ціни

Після визначення функцій ціни потрібно визначити як агенти будуть їх використовувати. Як правило, агенти отримують інформацію про поточний стан та мають певні можливості змінювати його, вибираючи власні дії. Іноді, звичайно ці дії не досконалі - сенсори можуть помилятися, а механізми не завжди працювати як очікується.

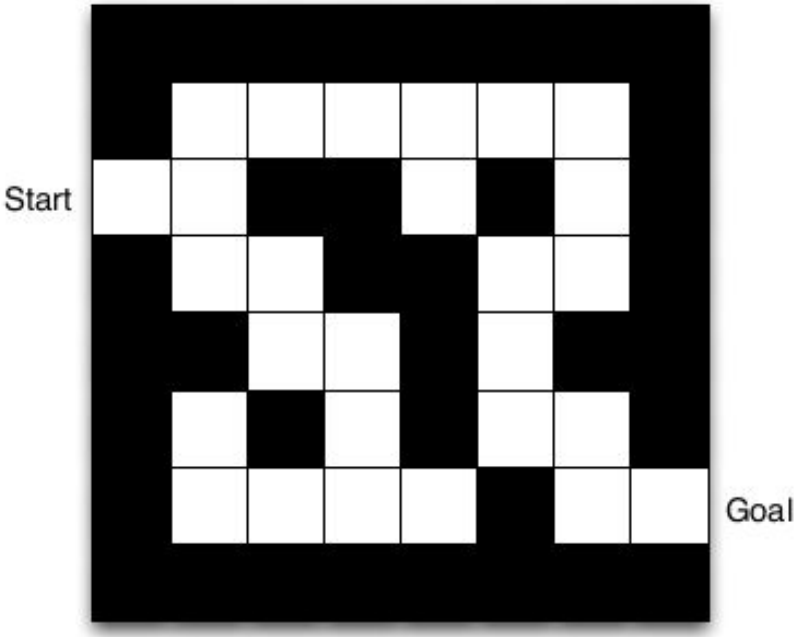
Функція ціни

Функція ціни, це передбачення майбутньої винагороди

Використовується для оцінки хорошості - поганості стану і, таким чином, для вибору між діями

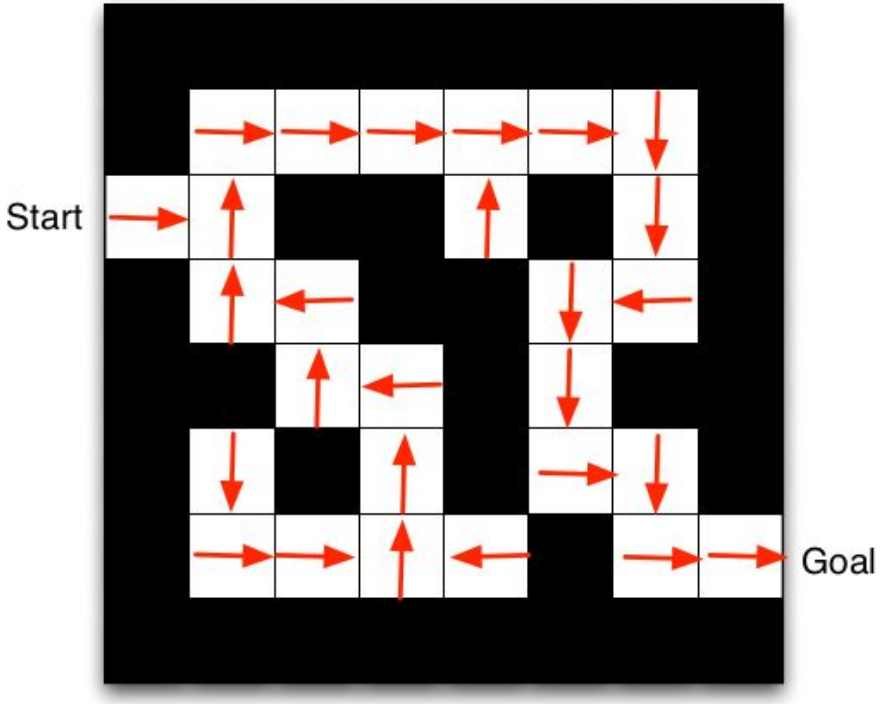
$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s \right]$$

Приклад.

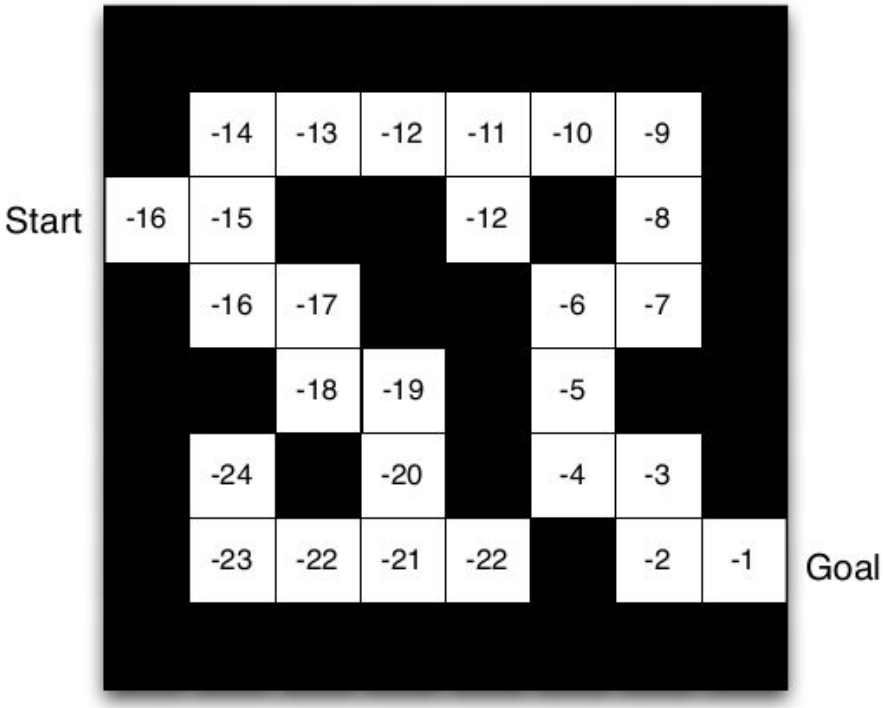


- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

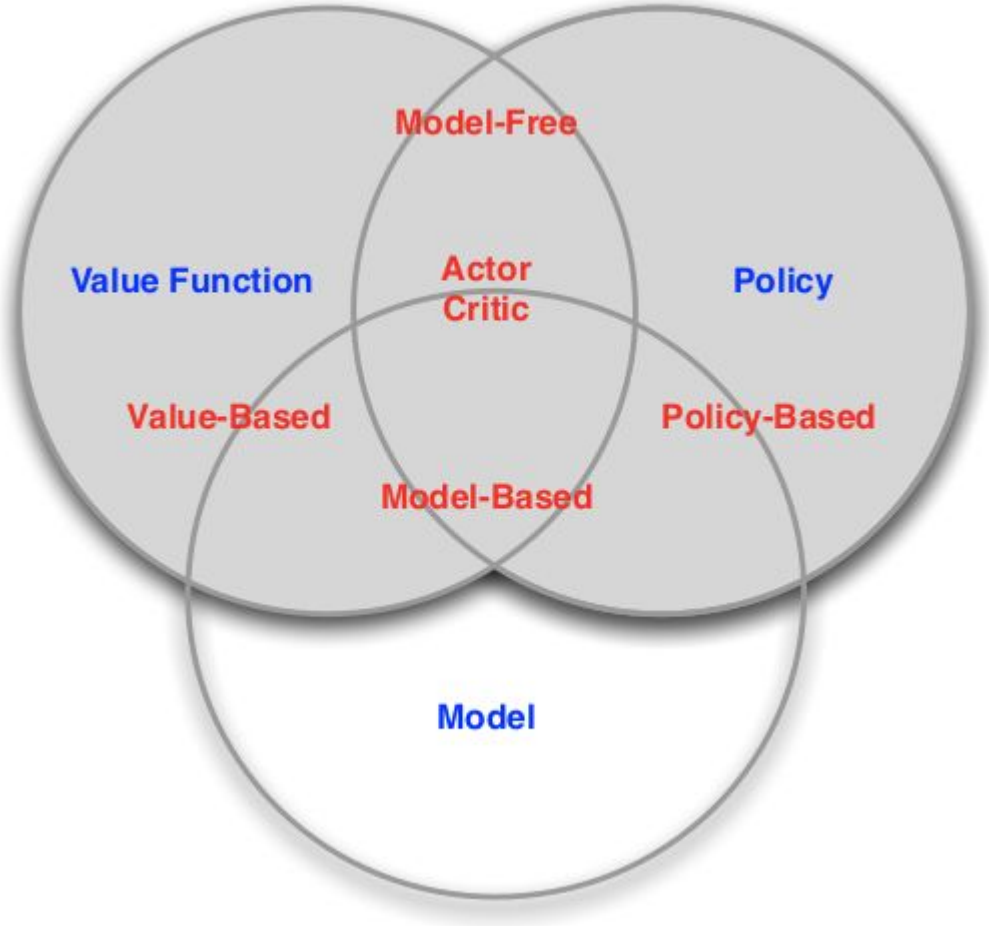
Політика



Функція ціни



Типи агентів



Дві фундаментальні проблеми

Навчання з підкріпленням:

Середовище на початку не відоме.

Агенти взаємодіють з середовищем

Агенти покращують свою політику

Дві фундаментальні проблеми

Планування

Модель середовища відома

Агенти виконують обчислення з моделі, без реальної взаємодії

Агенти покращують свою політику

Q навчання

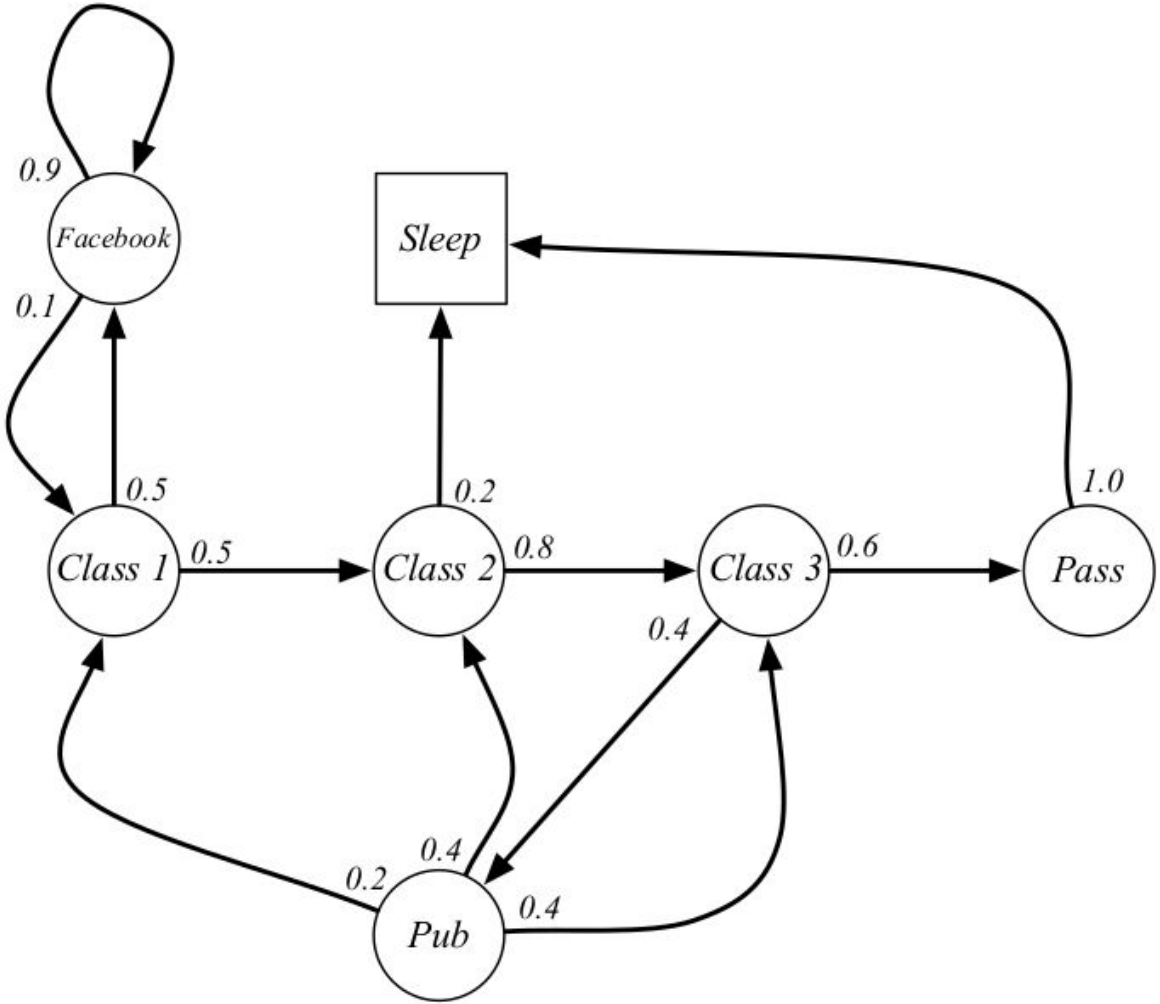
$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Big(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal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

МППР

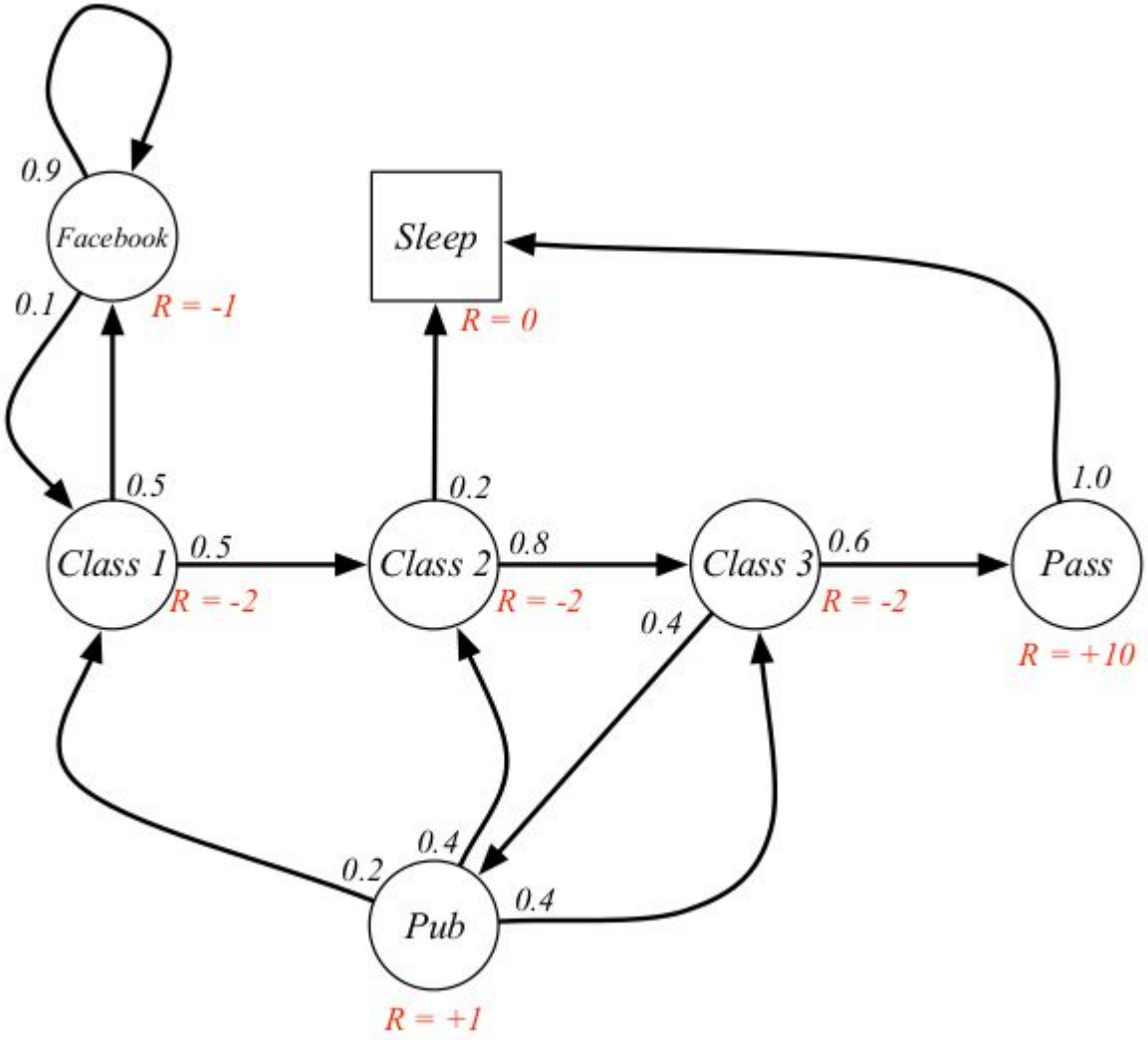
В реальних задачах агенти існують у середовищі, стан якого змінюється внаслідок дій агента або зовнішних подій. Тобто агент сприймає стан світу, виконує дію, сприймає новий змінений стан. Формально ця ідея виражається Марковським процесом прийняття рішень (МППР).

Визначення. (Марковського процесу прийняття рішень). МППР визначається початковим станом $s_1 \in S$, перехідною функцією $T(s, a, s_0)$ і функцією виграшу $r : S \to R$.

Перехідна функція $T(s, a, s_0)$ дорівнює ймовірності переходу агента зі стану s в стан s_0 при здійсненні дії a. Для чисто детерміністичного світу ця функція буде для кожного фіксованого s приймати значення 1 для одного s_0 і 0 для всіх інших пар.


0.9
Facebook
0.1
Sleep
0.5
0.2
1.0
Class 1
0.5
Class 2
0.8
Class 3
0.6
Pass
0.4
0.4
0.2
Pub
0.4

Марківський процес з виграшами



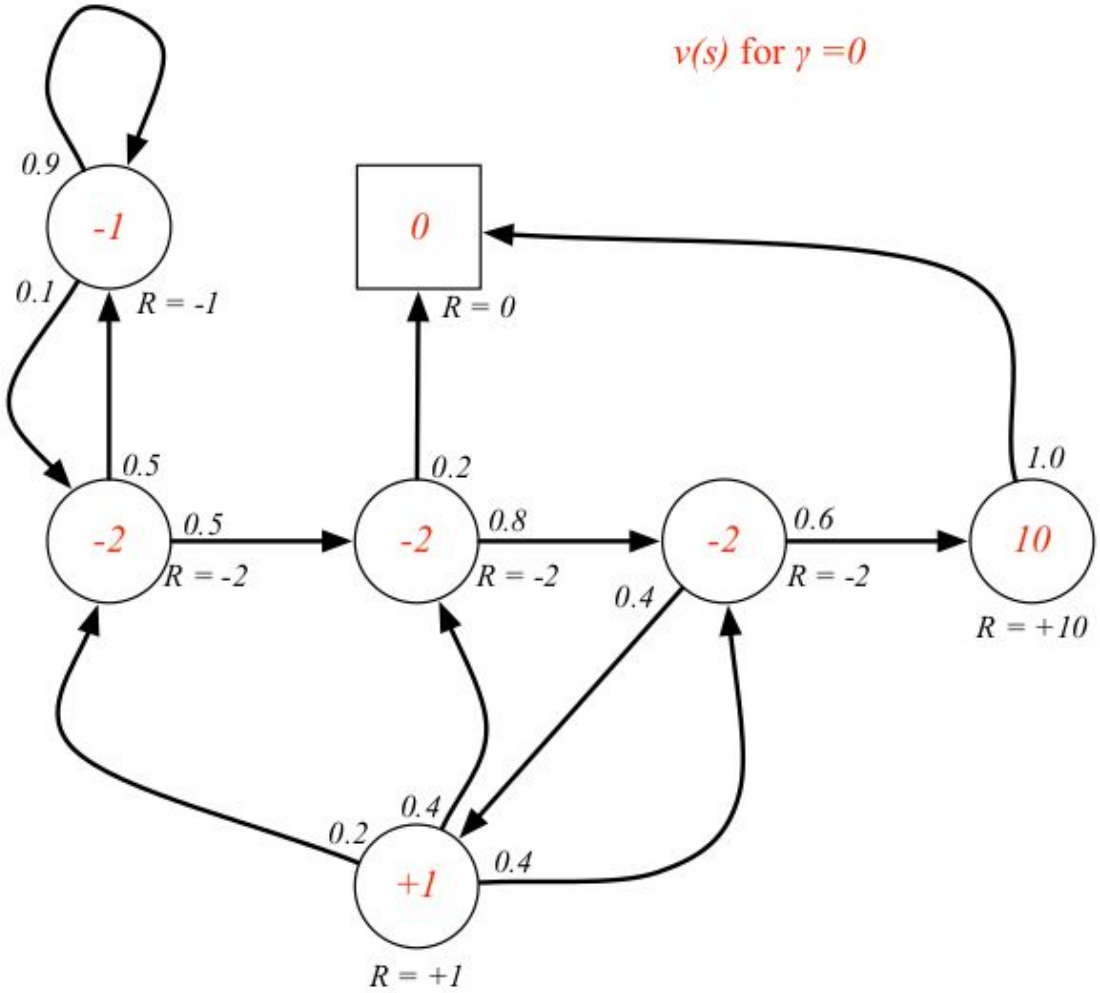
Типи винагороди

- повна винагорода: $V = \sum_{i=1}^{\infty} r(s_i)$. Працює для випадків коли сума точно скінчена, інакше важко порівнювати виграші різних політик. (наприклад ненульова ймовірність переходу у термінальний стан гарантує скінченність)

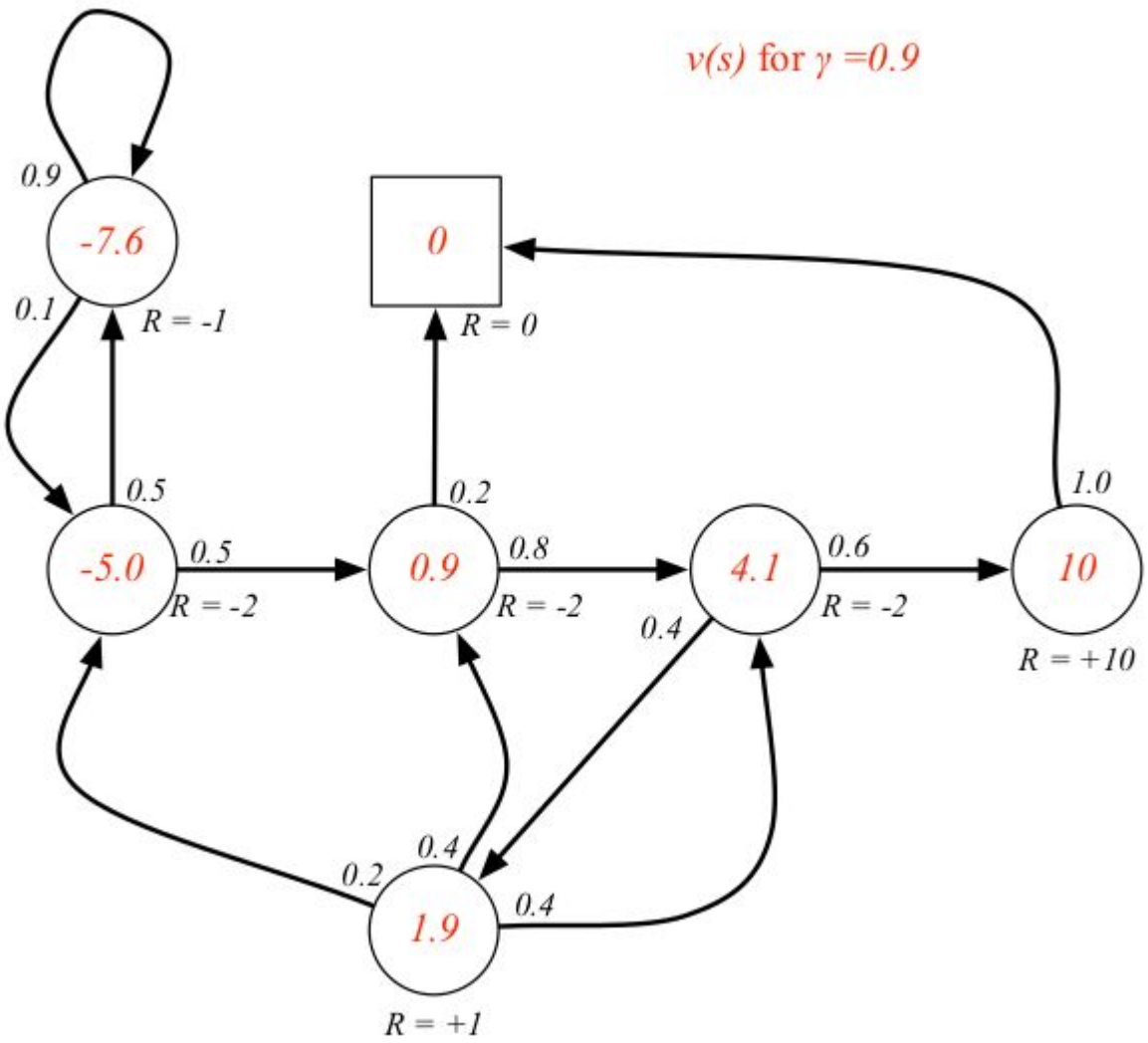
- середня винагорода: $V = \lim_{n\to\infty}(\frac{r(s_1)+...+r(s_n)}{n})$. В цьому випадку виграші усереднені протягом часу. Тут важлива скінченність виграшів. Якщо виграші скінченні, то ліміт нульовий, отже всі дії агента будуть мати нульову корисність. Однак, якщо додати нескінченну кількість дій, то єдине. що буде важливим - останній стан агента. Тобто неважливо скільки поганих виграшів він отримав на шляху до виграшної ситуації.

- дисконтована винагорода: $V = \gamma^0 r(s_1) + \gamma^1 r(s_2) + \gamma^2 r(s_3) + \ldots$. При використанні цього підходу визначається дисконтуюча змінна $\gamma \in [0, 1]$, яка зменшує майбутній очікуваний виграш. Якщо $\gamma = 1$, то даний виграш співпадає з повною винагородою, Якщо $\gamma = 0$, то агент ігнорує майбутні винагороди (жадібна стратегія). Для всіх інших значень агент враховує майбутні виграші тою чи іншою мірою. Гарантується також, що функція

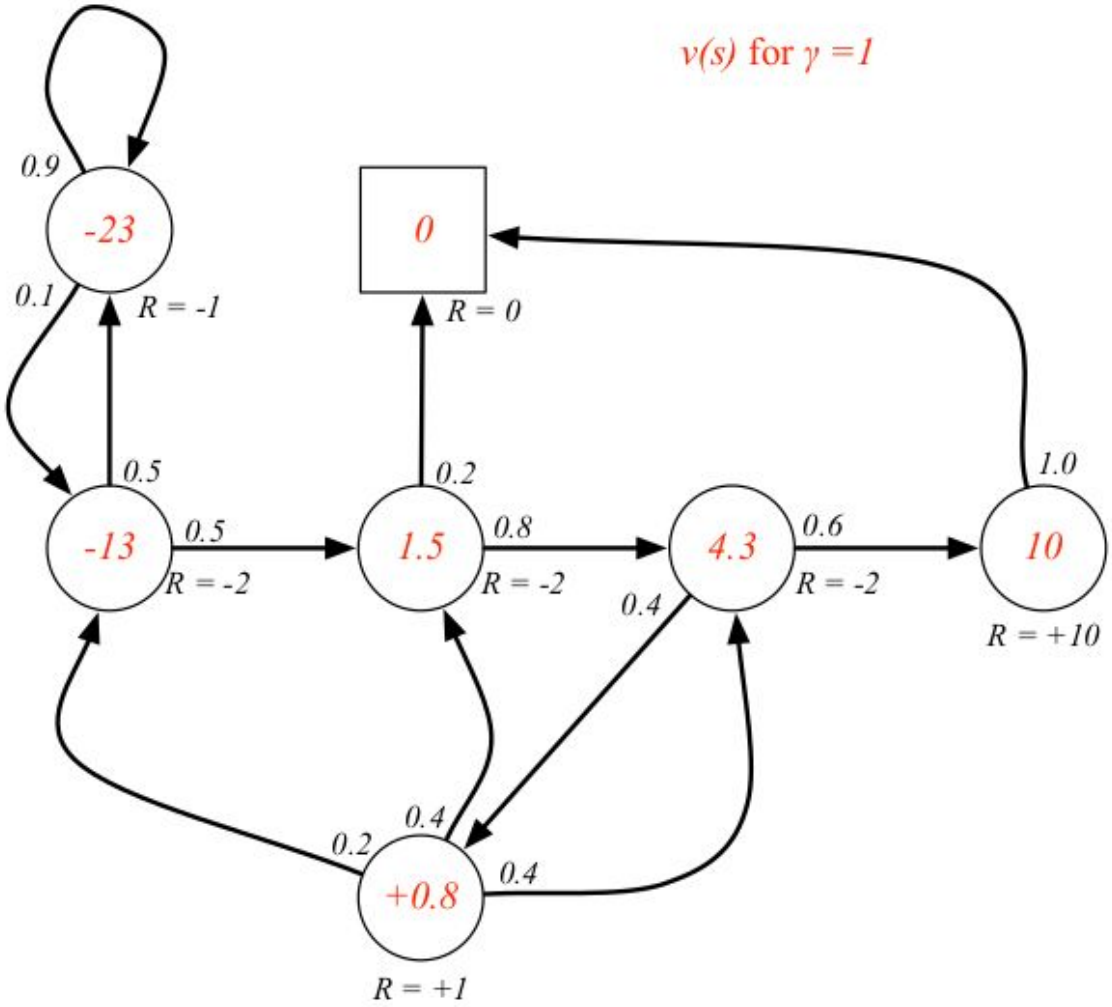
Функція ціни для прикладу



Функція ціни з дисконтом



Функція ціни з дисконтом

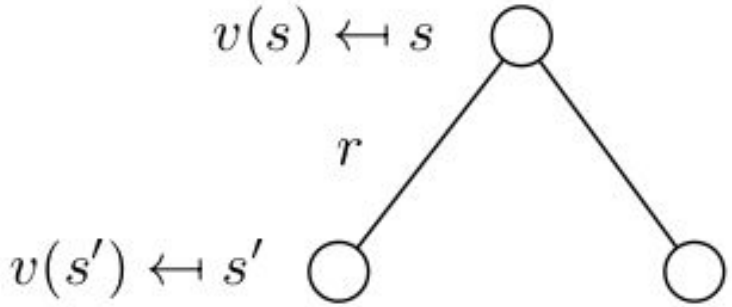


Функція ціни - ключова для визначення політики

Функція ціни кожного стану можна розбити на безпосередній виграш плюс дисконтований виграш від наступного стану.

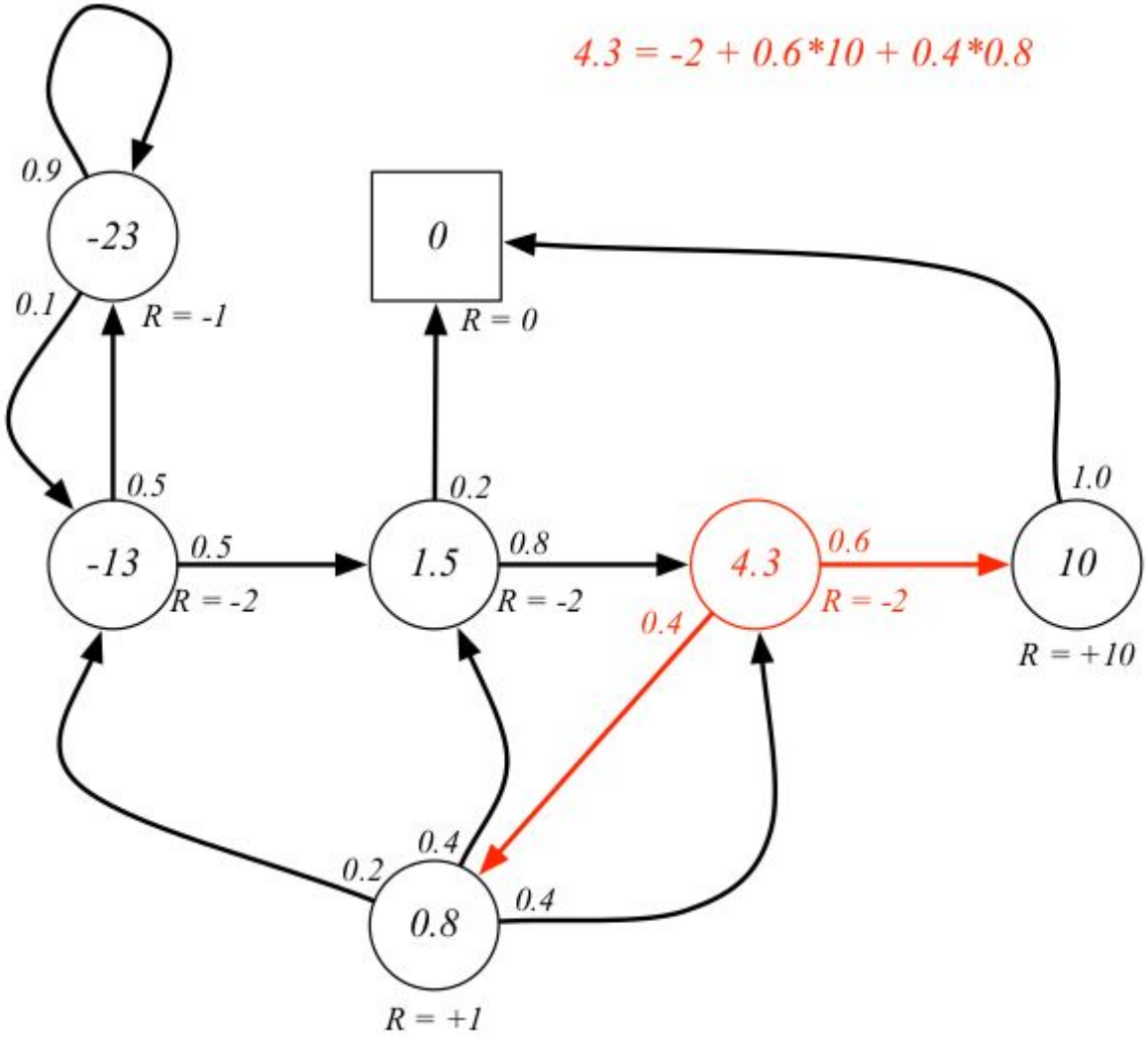
$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right]$$

$$v(s) = \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]$$



$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

Приклад.


4.3 = -2 + 0.6*10 + 0.4*0.8
0.9
-23
0.1
R = -1
0
R = 0
0.5
0.2
1.0
-13
0.5
1.5
0.8
4.3
0.6
10
R = -2
R = -2
0.4
R = +10
0.4
0.2
0.4
0.8
R = +1

Рівняння Белмана можна виписати явно

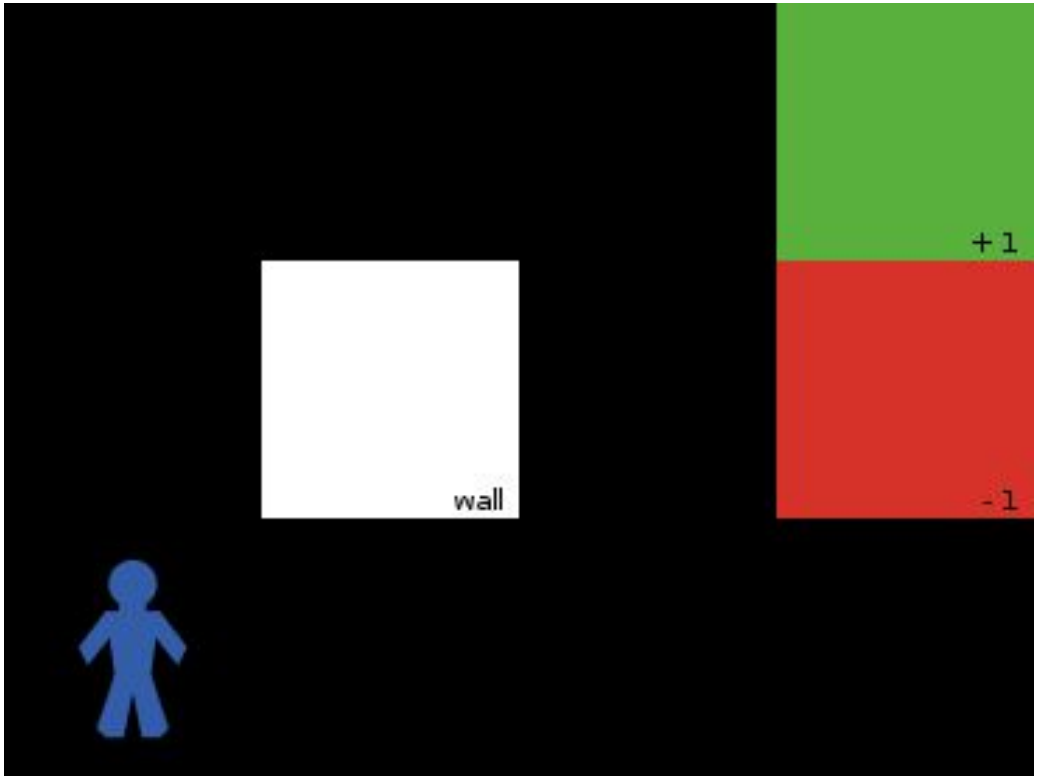
$$
\begin{aligned}
v &= \mathcal{R} + \gamma \mathcal{P} v \\
(I - \gamma \mathcal{P})\, v &= \mathcal{R} \\
v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R}
\end{aligned}
$$

Прямий розв’язок можливий лише для невеликих МППР

Ітеративні методи:

Динамічне програмування, Пошук Монте-Карло, TD-навчання

Простий лабіринт



Приклад обчислення функції ціни

Алгоритм Белмана

VALUE-ITERATION(T, r, γ, ϵ)

do
  $u \leftarrow u'$
  $\delta \leftarrow 0$
  for $s \in S$
    do $u'(s) \leftarrow r(s) + \gamma \max_a \sum_{s'} T(s, a, s')u(s')$
      if $|u'(s) - u(s)| > \delta$
        then $\delta \leftarrow |u'(s) - u(s)|$
  until $\delta < \epsilon(1 - \gamma)/\gamma$
return u

Рівняння Белмана

$$u(s) = r(s) + \gamma \max_a \sum_{s'} T(s, a, s') u(s').$$

$$u^{t+1}(s) \leftarrow r(s) + \gamma \max_a \sum_{s'} T(s, a, s') u^t(s').$$