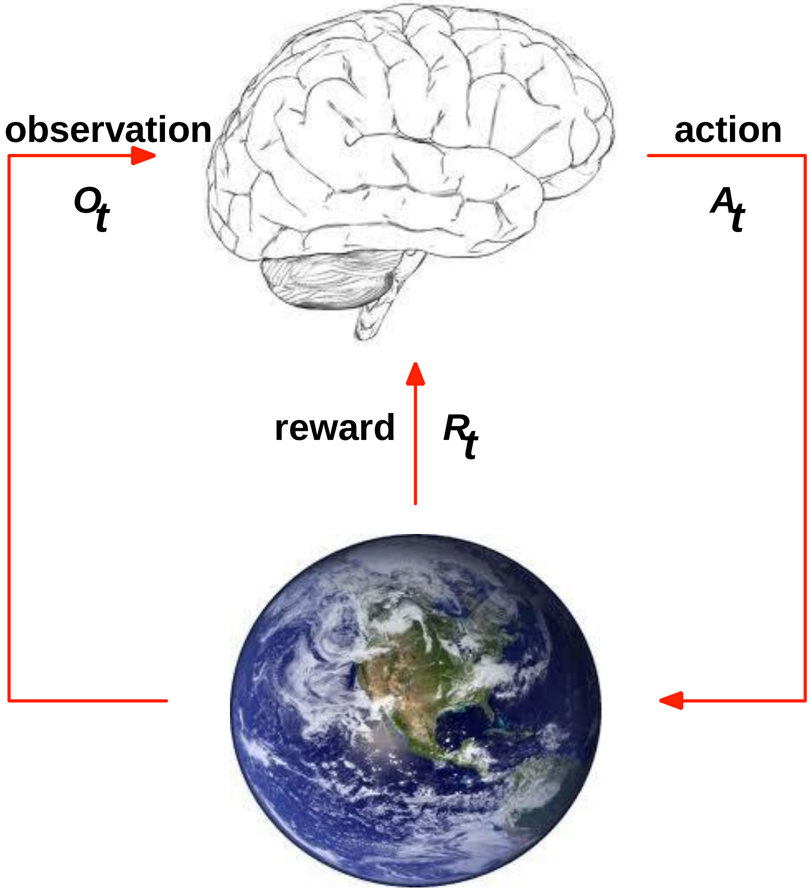
RL: Value Approximation

План

- Аппроксимация функции полезности
- Инкрементальные алгоритмы
- Double Q-learning
- Dueling Network

Взаимодействие среды и агента. MDP



Пространство большой размерности в RL

- ❑ Нарды 10^{20}
- ❑ Go: 10^{170}
- ❑ Автономный автомобиль: непрерывное множество

Как можно использовать безмодельные методы (например Q learning)?

Аппроксимация функции полезности

- Функция полезности – это таблица (lookup table):
  - Для *V(s)* таблица состояние – оценка стоимости
  - Для Q(s,a) – таблица, где паре <состояние, действие> ставится в соответствии оценка стоимости
- Проблемы для больших MDP:
  - Много состояний, чтобы хранить в памяти
  - Медленное обучение
- Решение для больших MDP:
  - Оценка полезности с помощью функции аппроксимации:

$$\hat{V}(s,w) \approx V^{\pi}(s), \quad \hat{Q}(s,a,w) \approx Q^{\pi}(s,a)$$

  - Обобщение наблюдаемых состояний на ненаблюдаемые
  - Обновление параметров w с использование MC или TD

Аппроксиматоры

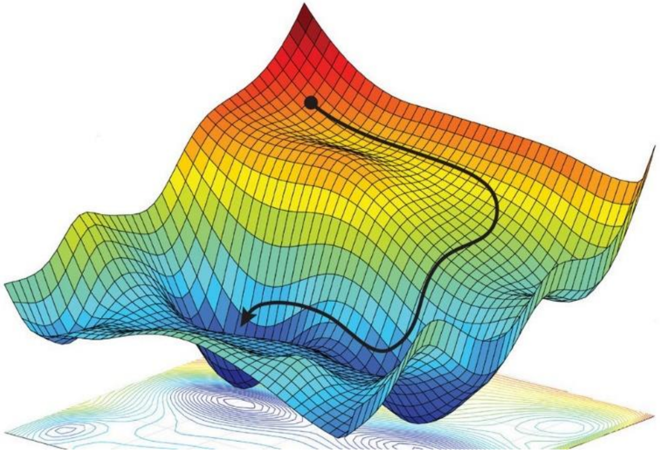
На классе **дифференцируемых** аппроксиматоров можно выделить:

- **Линейные (линейная комбинация признаков)**
- **Нейронные сети**
- Деревья принятия решений
- Ближайшие соседи
- Фурье/вейвлет разложения
- ….

Градиентный спуск

- Пусть *J*(***w***) – дифференцируемая функция вектора параметров ***w***
- Градиент функции *J*(***w***)

$$\nabla_w J(w) = \begin{pmatrix} \frac{\partial J(w)}{\partial w_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial J(w)}{\partial w_n} \end{pmatrix}$$



- Ищем локальный минимум функции J(w)
- Обновляем w в направлении антиградиента

$$\Delta w = -\frac{1}{2}\alpha \nabla_w J(w)$$

Стохастический градиентный спуск

- Предполагаем, что мы знаем истинные значения $V^{\pi}(s)$, как найти вектор параметров, минимизируя СКО?

$$J(w) = E_{\pi}\left[\left(V^{\pi}(s) - \hat{V}(s,w)\right)^2\right]$$

- Градиентный спуск позволяет найти локальный минимум:

$$\Delta w = \frac{1}{2}\alpha \nabla_w J(w) = \alpha E_{\pi}\left[\left(V^{\pi}(s) - \hat{V}(s,w)\right)\nabla_w \hat{V}(s,w)\right]$$

- Стохастический градиентный спуск производит спуск по подвыборке

$$\Delta w = \alpha\left(V^{\pi}(s) - \hat{V}(s,w)\right)\nabla_w \hat{V}(s,w)$$

Вектор признаков

- Состояние как вектор признаков:

$$x(s) = \begin{pmatrix} x_1(s) \\ \vdots \\ x_n(s) \end{pmatrix}$$

- Примеры:
  - Положение фигуры на шахматной доске
  - Расстояние автомобиля до ключевых точек (автономное вождение)
  - Цена ценной бумаги на бирже
  - Матрица пикселей в atari

Линейная аппроксимация

- Функция полезности как линейная комбинация признаков

$$\hat{V}(s,w) = x(s)^T w = \sum_{i=1}^{n} x_i(s) w_i$$

- Целевая функция (в задаче регрессии с L2 нормой)

$$J(w) = E_\pi[(V^\pi(s) - x(s)^T w)^2]$$

- Градиентный спуск сходится к глобальному оптимуму
- Правил обновления

$$\nabla_w \hat{V}(s,w) = x(s)$$

$$\Delta w = \alpha(V^\pi(s) - x(s)^T w)x(s)$$

- Вопрос: что делать, если истинные значения полезности неизвестны?

Инкрементальные алгоритмы предсказания

- Предыдущие выводы были справедливы если мы знаем истинные значения функции полезности
- В действительности мы не знаем функцию полезности
- На практике заменяют учителя, оценкой функции полезности $V^{\pi}(s)$:
  - Для МК оценка – это отдача R_t :

$$\Delta w = \alpha\left(R_t - \hat{V}(s_t, w)\right)\nabla_w \hat{V}(s_t, w)$$

  - Для TD(0) оценка – это показатель $r_{t+1} + \gamma\hat{V}(s_{t+1}, w)$:

$$\Delta w = \alpha\left(r_{t+1} + \gamma\hat{V}(s_{t+1}, w) - \hat{V}(s_t, w)\right)\nabla_w \hat{V}(s_t, w)$$

МК с аппроксимацией функции полезности

- Отдача R_t несмещенная, зашумленная подвыборка истинного значения $V^{\pi}(s)$
- Можем применить обучение с учителем для “обучающей выборки”:

$$\langle s_1, R_1 \rangle, \langle s_2, R_2 \rangle, \dots, \langle s_\tau R_\tau \rangle$$

- Пример: использование линейной оценки МК:

$$\Delta w = \alpha \left(R_t - \hat{V}(s_t, w) \right) \nabla_w \hat{V}(s_t, w) = \alpha \left(R_t - \hat{V}(s_t, w) \right) x(s)$$

- МК оценка сходится к локальному оптимуму
- Это верно и при использовании нелинейных аппроксиматоров

TD обучение с аппроксимацией функции полезности

- TD показатель $r_{t+1} + \gamma \hat{V}(s_{t+1}, w)$ - смещенная подвыборка истинного значения $V^{\pi}(s)$
- Можем применить обучение с учителем для “обучающей выборки”:

$$\langle s_1, r_2 + \gamma \hat{V}(s_2, w) \rangle, \langle s_2, r_3 + \gamma \hat{V}(s_3, w) \rangle, \dots, \langle s_\tau, r_{\tau+1} + \gamma \hat{V}(s_{\tau+1}, w) \rangle$$

- Пример: использование линейного TD(0):

$$\Delta w = \alpha \left(r_{t+1} + \gamma \hat{V}(s_{t+1}, w) - \hat{V}(s_t, w) \right) \nabla_w \hat{V}(s_t, w) = \alpha \delta x(s)$$

- Линейная TD(0) оценка сходится к локальному оптимуму

Аппроксимация функции полезности действия

- Аппроксимация функции полезности действия:

$$\hat{Q}(s,a,w) \approx Q^{\pi}(s,a)$$

- Минимизация СКО:

$$J(w) = E_{\pi}\left[\left(Q^{\pi}(s,a) - \hat{Q}(s,a,w)\right)^2\right]$$

- Использование стохастического градиентного спуска для поиска локального минимума

$$-\frac{1}{2}\nabla_w J(w) = \left(Q^{\pi}(s,a) - \hat{Q}(s,a,w)\right)\nabla_w \hat{Q}(s,a,w)$$

$$\Delta w = \alpha\left(Q^{\pi}(s,a) - \hat{Q}(s,a,w)\right)\nabla_w \hat{Q}(s,a,w)$$

Линейная аппроксимация функции полезности действия

- Представим состояние и действие в виде вектора признаков:

$$x(s) = \begin{pmatrix} x_1(s,a) \\ \vdots \\ x_n(s,a) \end{pmatrix}$$

- Функция полезности как линейная комбинация признаков:

$$\hat{Q}(s,a,w) = x(s,a)^T w = \sum_{i=1}^{n} x_i(s,a) w_i$$

- Правил обновления

$$\nabla_w \hat{Q}(s,a,w) = x(s,a)$$

$$\Delta w = \alpha(Q^\pi(s,a) - x(s,a)^T w) x(s,a)$$

Инкрементальный алгоритм управления

- Как и в предсказании, меняем оценку для $Q^{\pi}(s,a)$

  - Для МК оценка – это отдача R_t:

$$\Delta w = \alpha \left(R_t - \hat{Q}(s_t, a_t, w) \right) \nabla_w \hat{Q}(s_t, a_t, w)$$

- Для TD(0) оценка – это показатель $r_{t+1} + \gamma \hat{Q}(s_{t+1}, a_{t+1}, w)$:

$$\Delta w = \alpha \left(r_{t+1} + \gamma \hat{Q}(s_{t+1}, a_{t+1}, w) - \hat{Q}(s_t, a_t, w) \right) \nabla_w \hat{Q}(s_t, a_t, w)$$

Линейное предсказание LSM в пакетном методе

- В точке минимума матожидание обновления должно быть равно 0:

$$E_D[\Delta w] = 0$$

$$\alpha \sum_{t=1}^{\tau} x(s_t)(V_t^\pi - x(s_t)^T w) = 0$$

$$\sum_{t=1}^{\tau} x(s_t) V_t^\pi = \sum_{t=1}^{\tau} x(s_t) x(s_t)^T w$$

$$w = \left(\sum_{t=1}^{\tau} x(s_t) x(s_t)^T \right)^{-1} \sum_{t=1}^{\tau} x(s_t) V_t^\pi$$

- Для N признаков сложность прямого решения равна $O(N^3)$
- Можно найти решение итерационными методами за $O(N^2)$

Алгоритмы линейного предсказания в LSM

- Истинные значения V_t^π не известны
- На практике, данные обучения будут давать зашумленную и/или смещенную выборку V_t^π
  - МК в качестве оценки V_t^π использует R_t
  - Метод TD в качестве оценки V_t^π использует $r_{t+1} + \gamma\hat{V}(s_{t+1}, w)$
- В каждом случае возможно прямое решение для фиксированной точки

Алгоритмы линейного предсказания в LSM

- LSMC

$$0 = \sum_{t=1}^{\tau} \alpha \left(R_t - \hat{V}(s_t, w) \right) x(s_t)$$

$$w = \left(\sum_{t=1}^{\tau} x(s_t) x(s_t)^T \right)^{-1} \sum_{t=1}^{\tau} x(s_t) R_t$$

- LSTD

$$0 = \sum_{t=1}^{\tau} \alpha \left(r_{t+1} + \gamma \hat{V}(s_{t+1}, w) - \hat{V}(s_t, w) \right) x(s_t)$$

$$w = \left(\sum_{t=1}^{\tau} x(s_t) \big(x(s_t) - \gamma x(s_{t+1}) \big)^T \right)^{-1} \sum_{t=1}^{\tau} x(s_t) r_{t+1}$$

Q обучение LSM

- Рассмотрим следующее обновление линейным Q обучением:

$$\delta = r_{t+1} + \gamma \hat{Q}(s_{t+1}, \pi(s_{t+1}, w\ ) - \hat{Q}(s_t, a_t, w)$$

$$\Delta w = \alpha \delta x(s_t, a_t)$$

- LSTDQ алгоритм: решение ищем по нулевому обновлению

$$0 = \sum_{t=1}^{\tau} \alpha \left(r_{t+1} + \gamma \hat{Q}(s_{t+1}, \pi(s_{t+1}, w\ ) - \hat{Q}(s_t, a_t, w) \right) x(s_t, a_t)$$

$$w = \left(\sum_{t=1}^{\tau} x(s_t, a_t) \big(x(s_t, a_t) - \gamma x(s_{t+1}, \pi(s_{t+1})) \big)^T \right)^{-1} \sum_{t=1}^{\tau} x(s_t, a_t) r_{t+1}$$