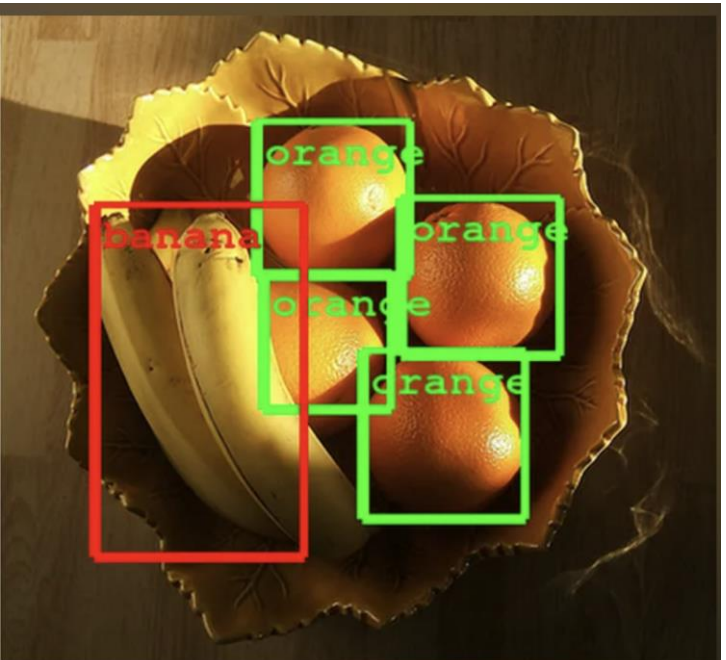
Reinforcement Learning

Sorokin Dmitriy

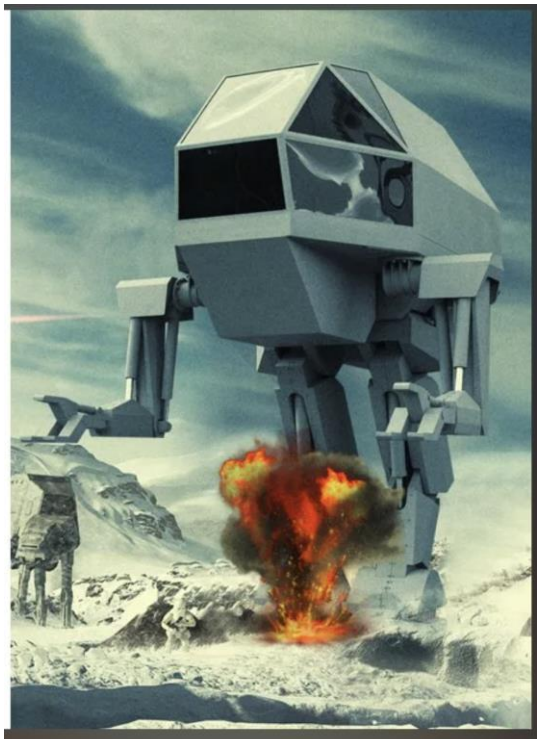
Проблема:

Классические задачи
Обучения с учителем

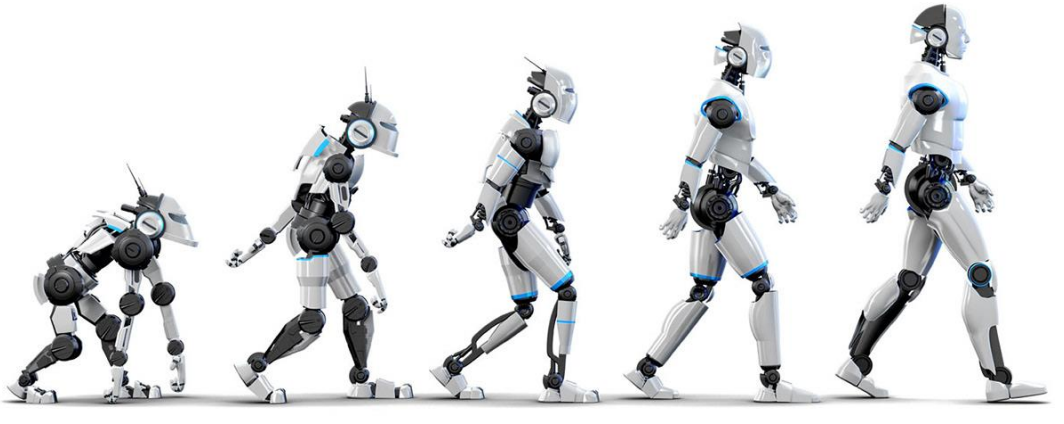


VS

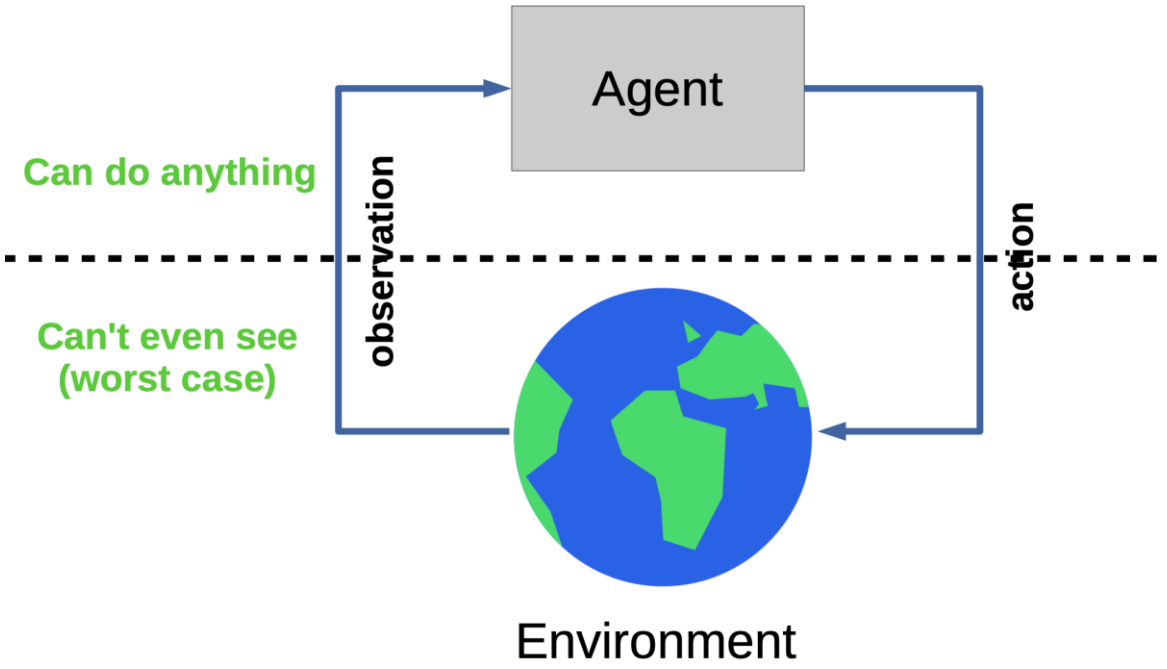
Ходячий робот?



Идея RL:



Постановка задачи



Формализация и MDP

Определения:

Состояние среды: $s \in S$

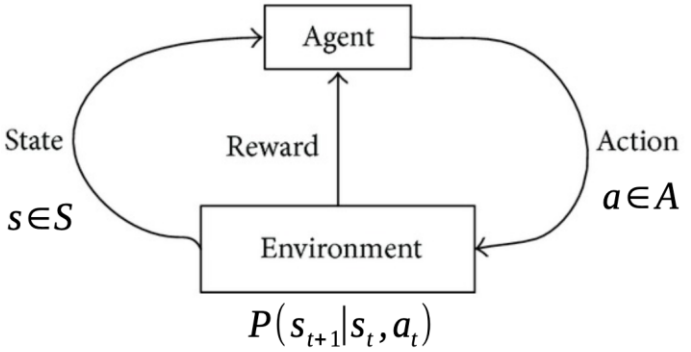
Действия агента: $a \in A$

Политика агента: $\pi(a|s) = P(take\ action\ a \mid state\ s)$

Вероятности перехода: $P(s_{t+1}|s_t, a_t)$

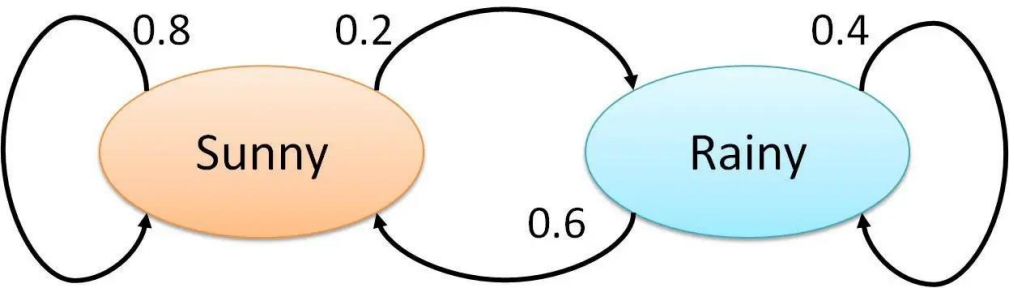
Награда за действия: $r \in R$

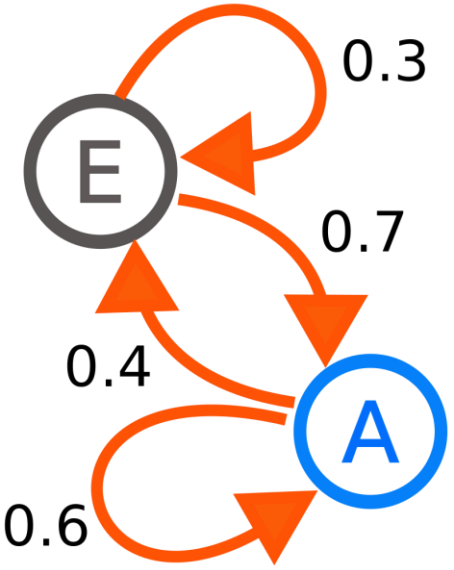
Markov Decision Process:



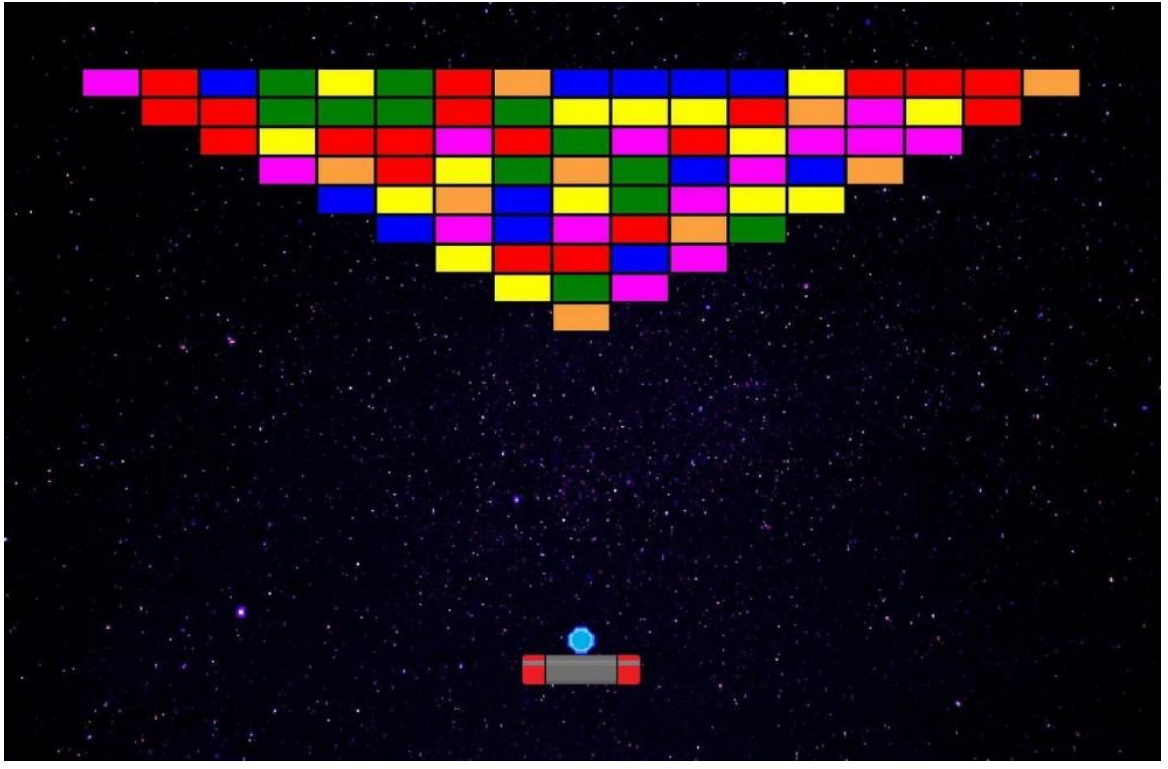
Markov assumption

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}) = P(s_{t+1}|s_t, a_t)$$





Markov assumption



Поговорим о награде

$$R = \sum_t r_t$$

$$R_t = \sum_{t'=t}^{n} r_{t'}$$

$$R_t = \sum_{k=0}^{\infty} y^k r_{t'+k}$$

$$\pi(a|s) = argmax E_\pi(R)$$

Кросс энтропийный метод

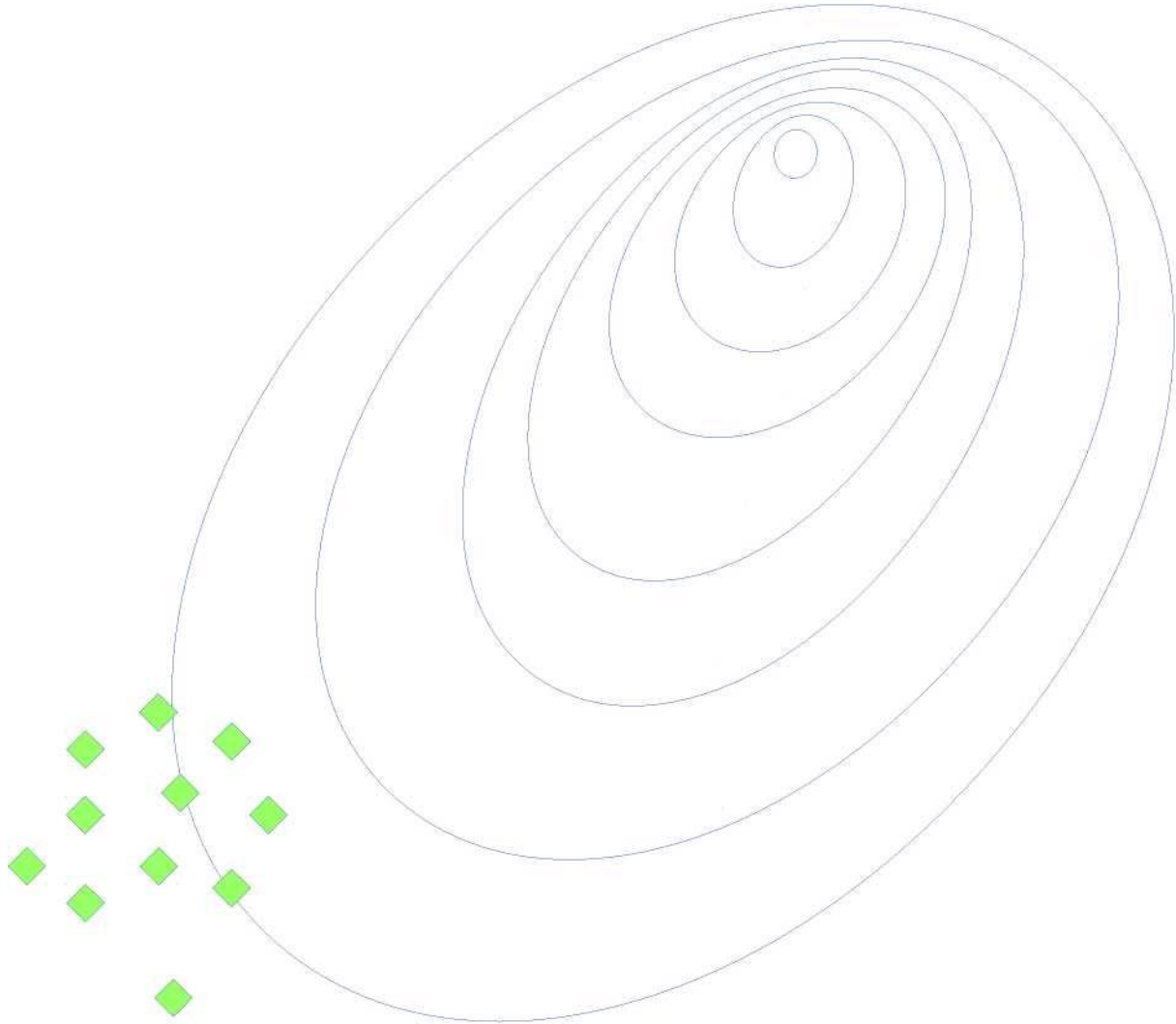
Шаги:

1) Запускаем наш алгоритм n раз

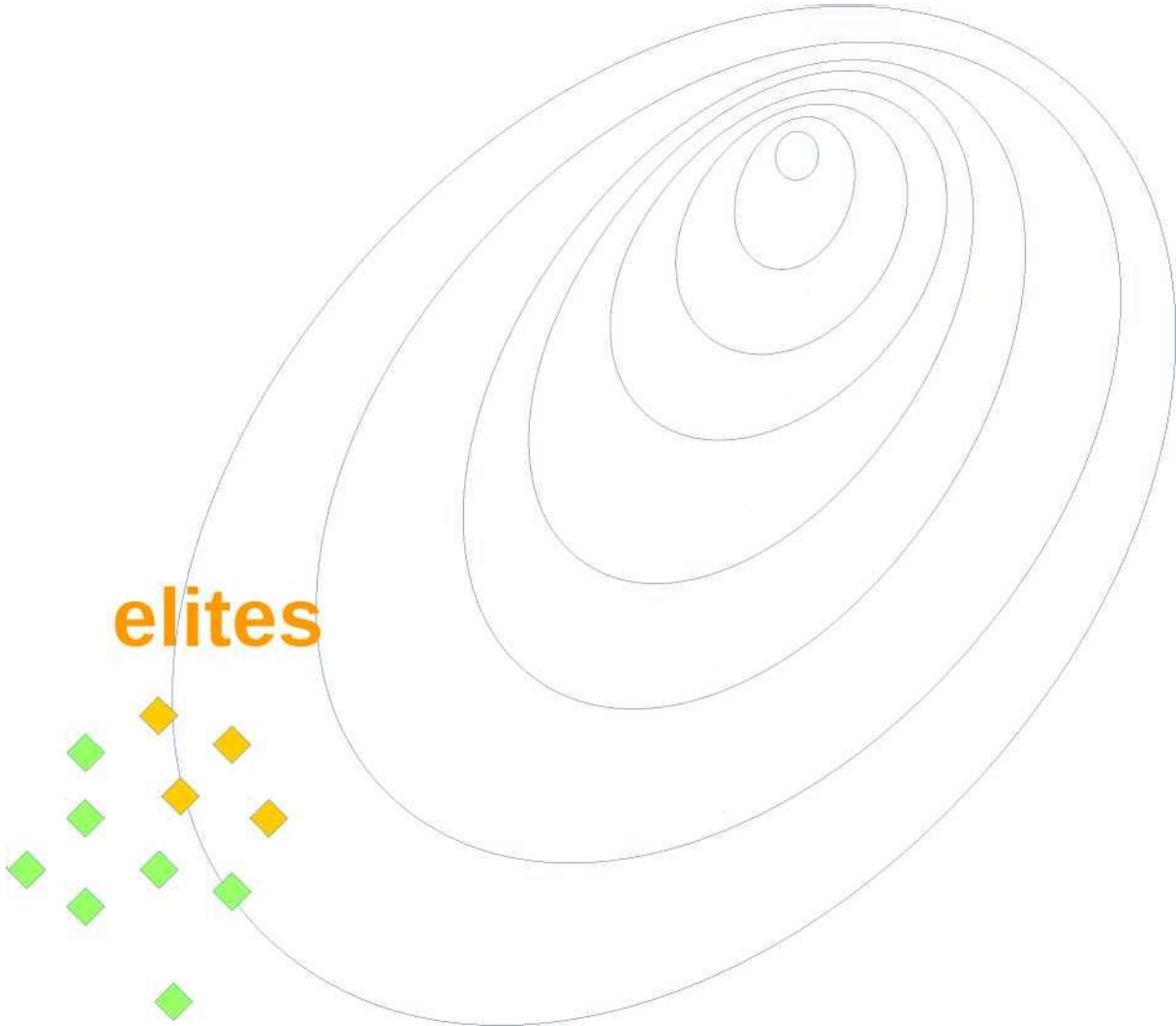
2) Выбираем m лучших сессий по награде

3) Изменяем политику чтобы сделать действия лучших сессий более вероятными

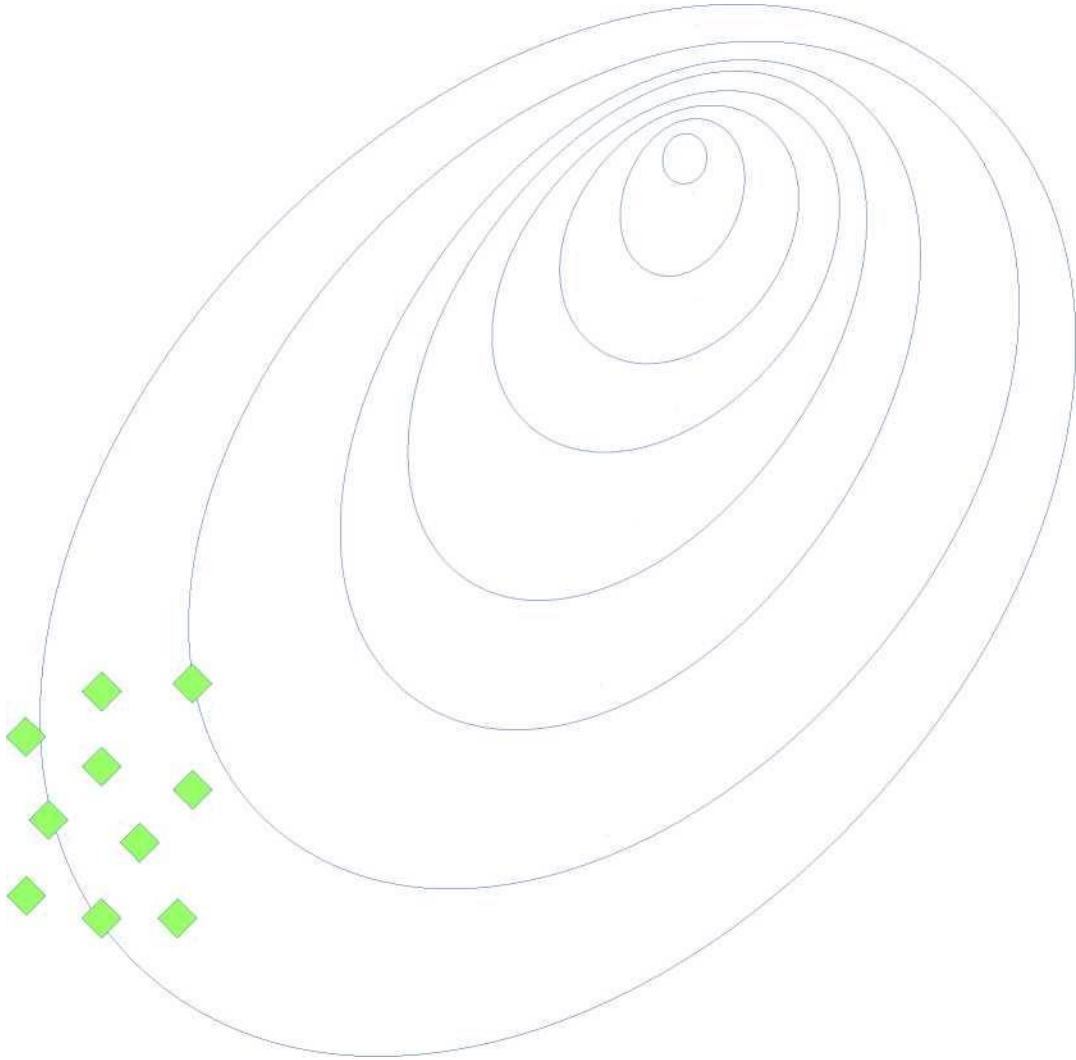
Пример:



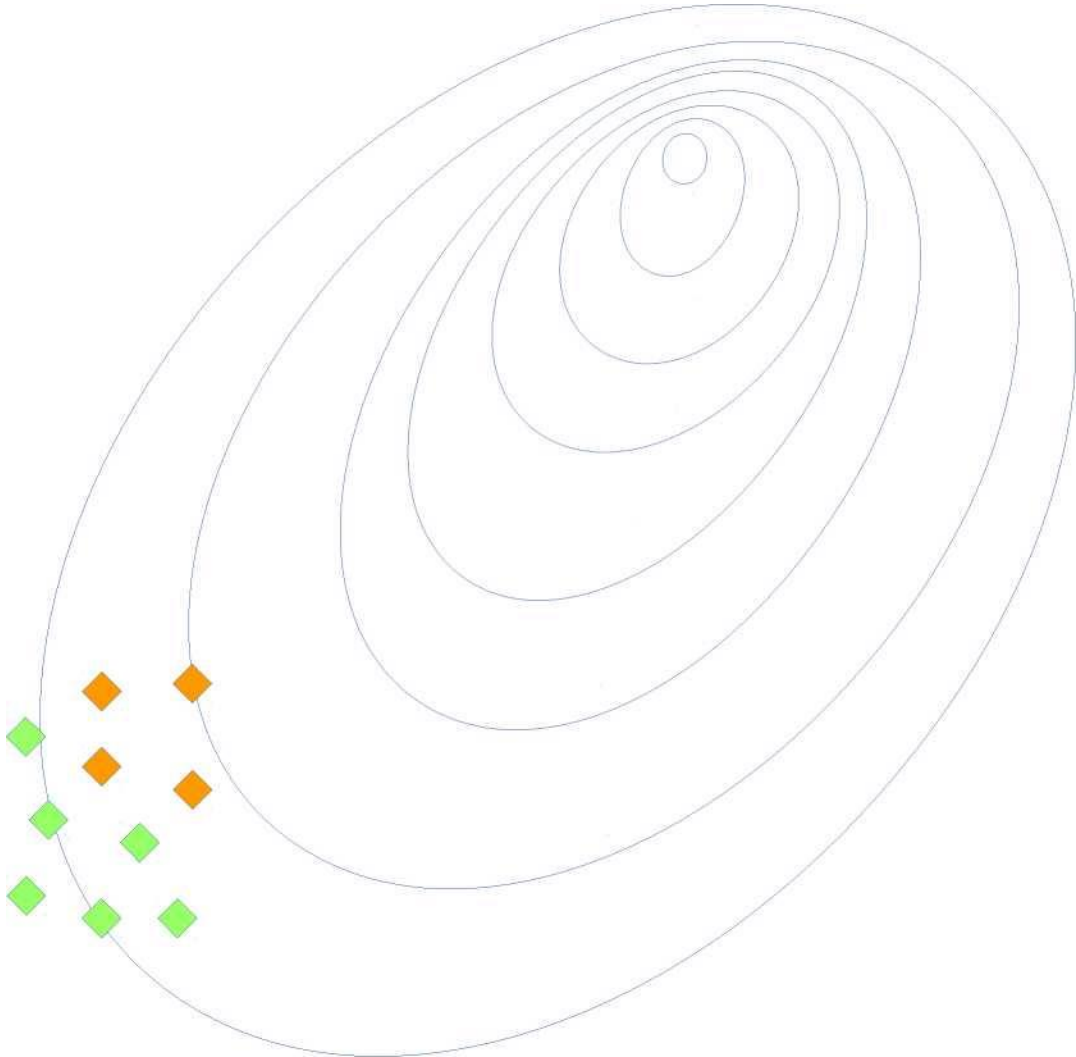
Пример:



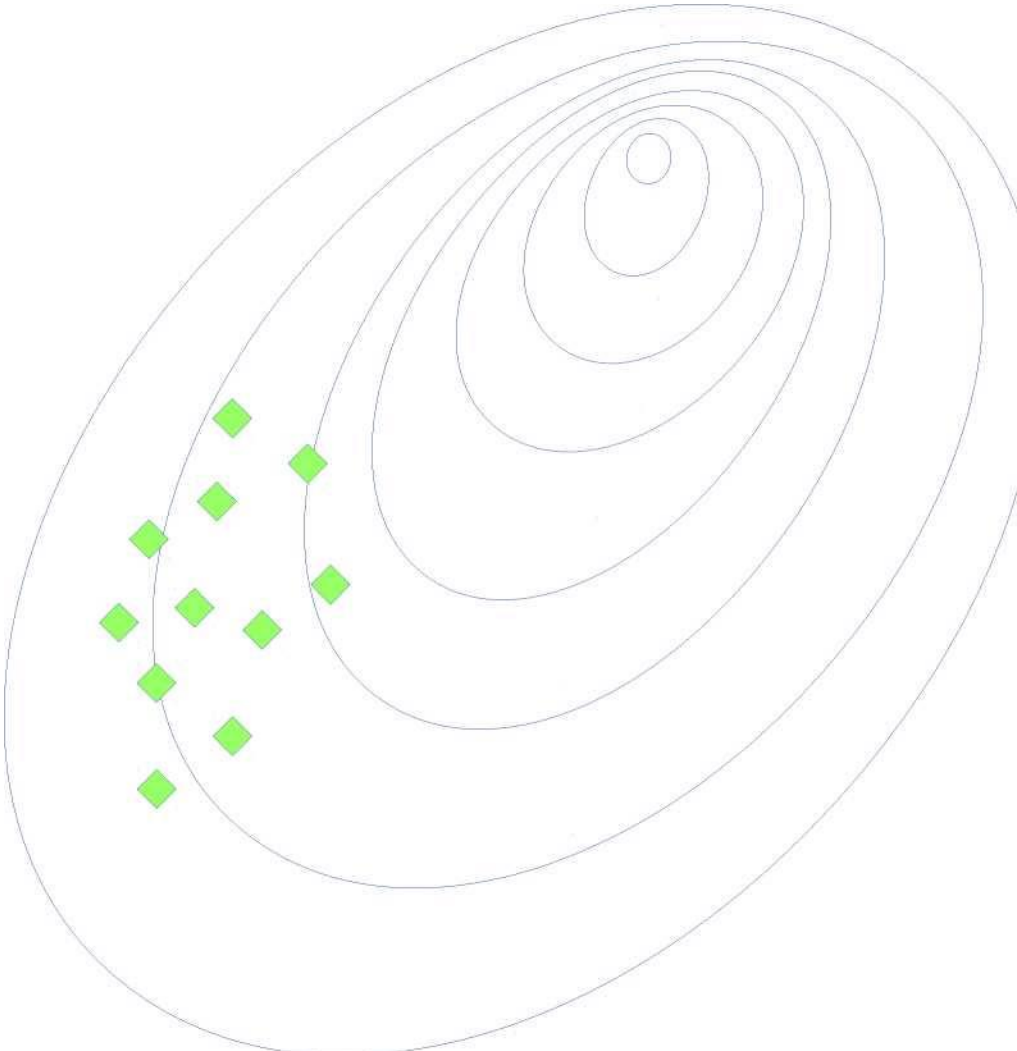
Пример:



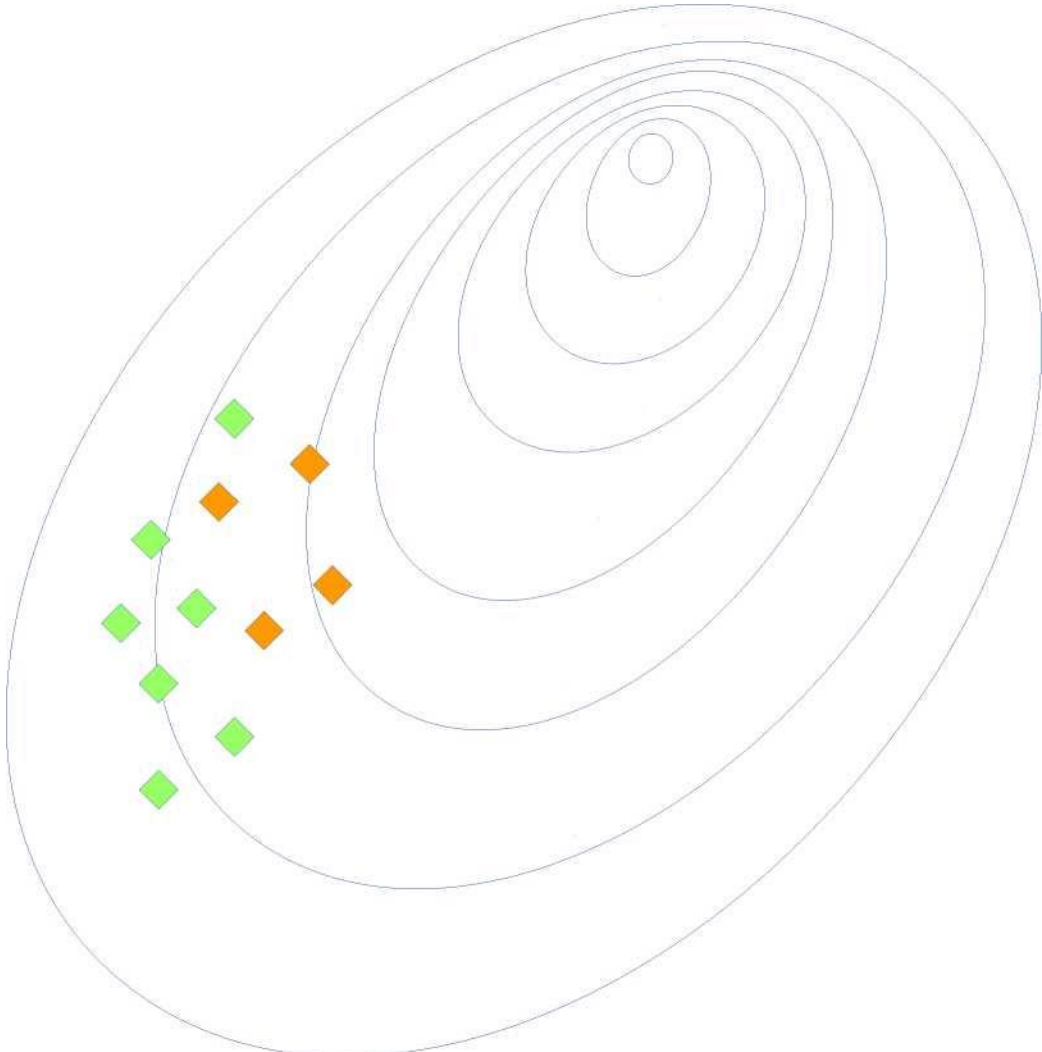
Пример:



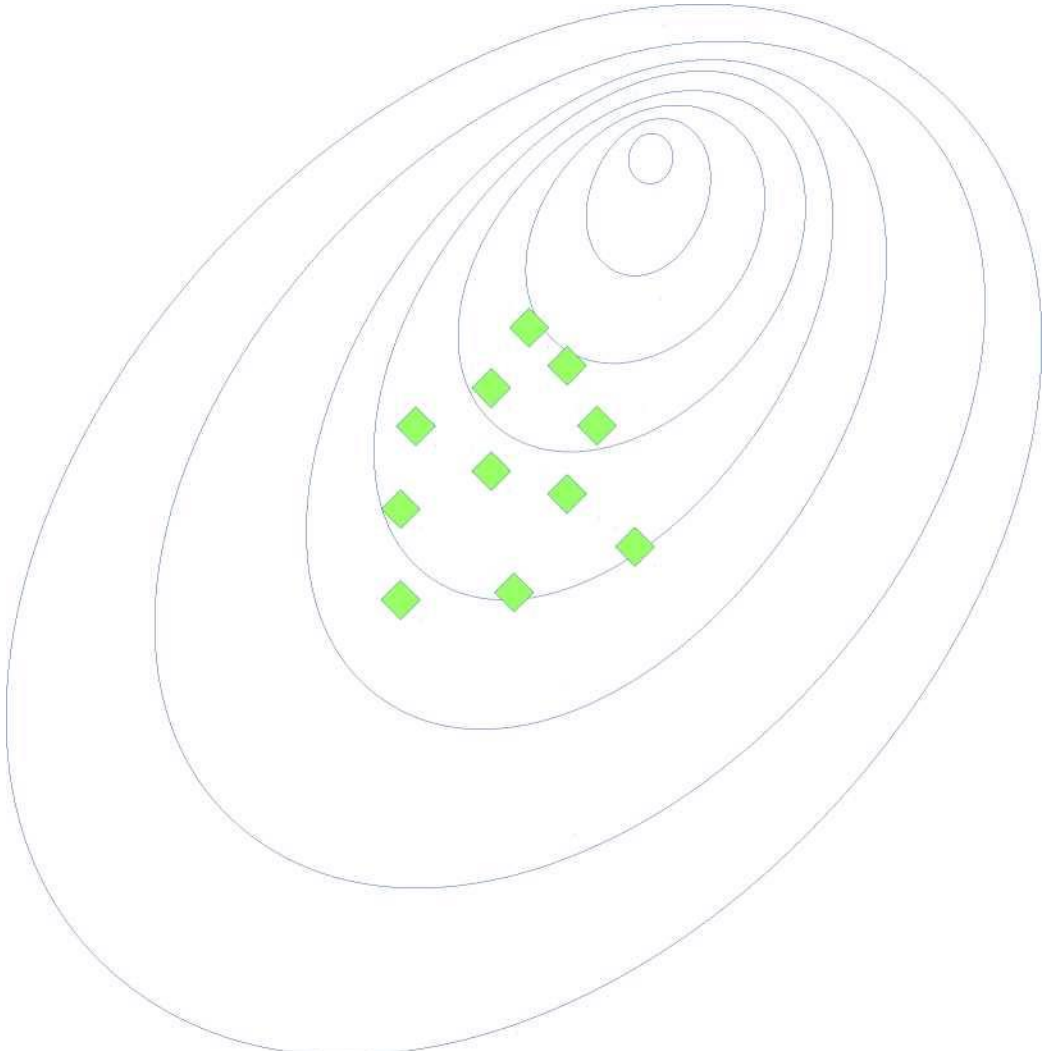
Пример:



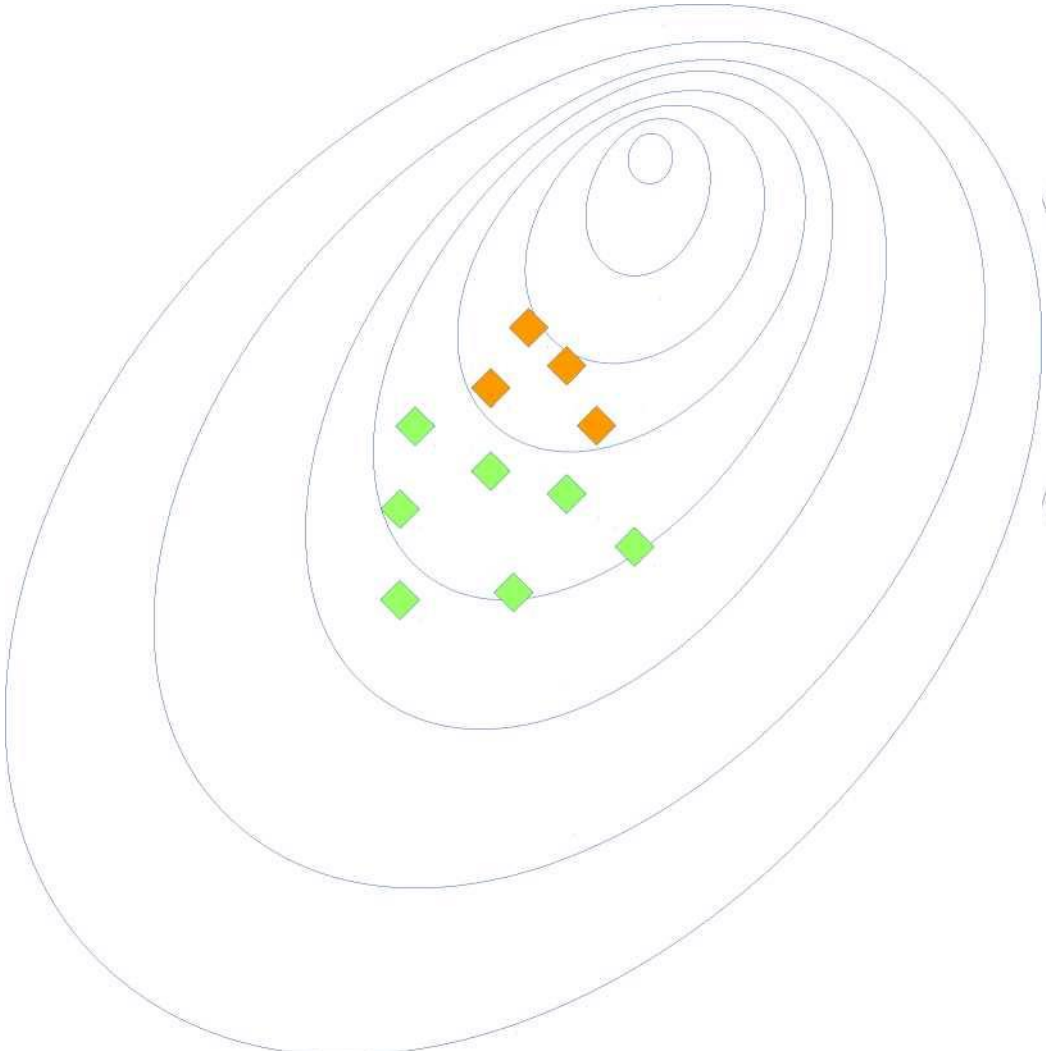
Пример:



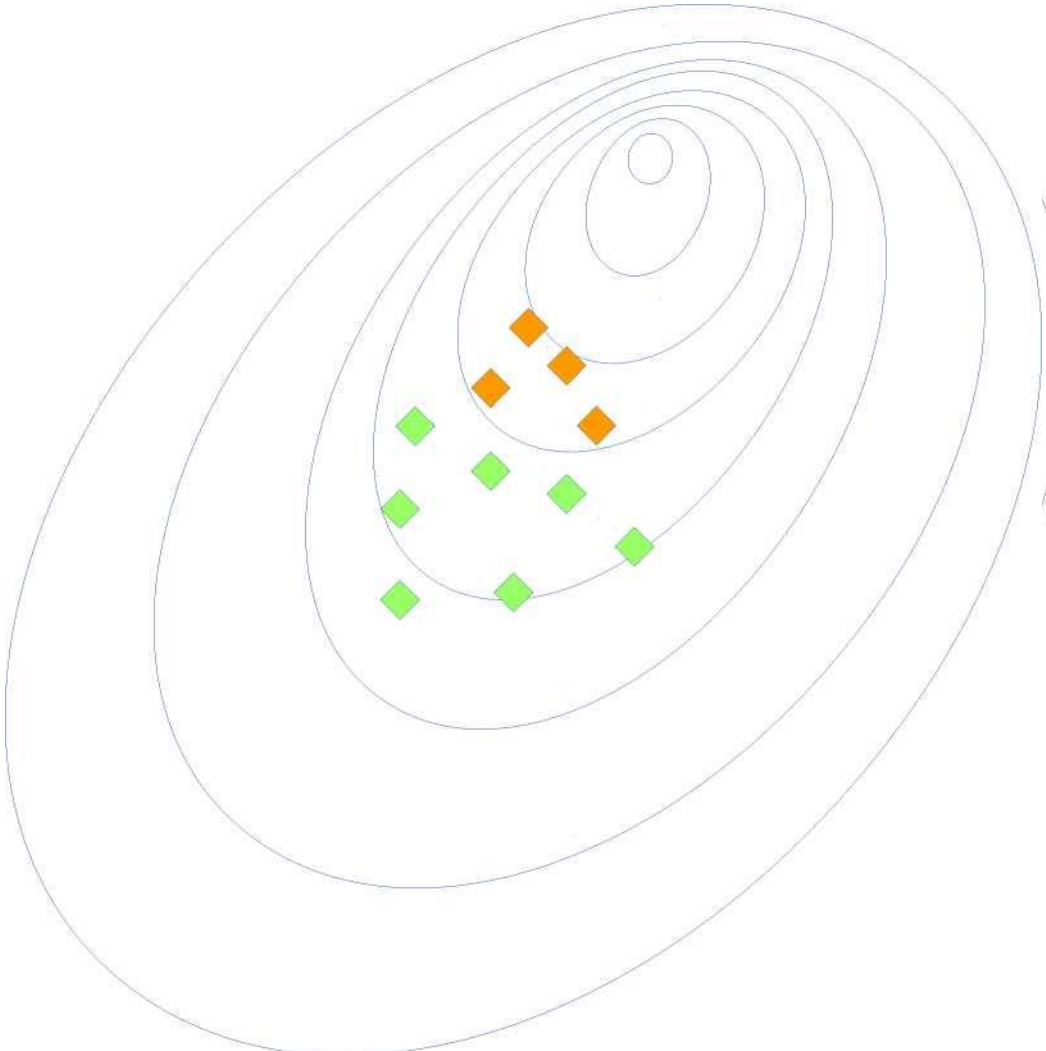
Пример:



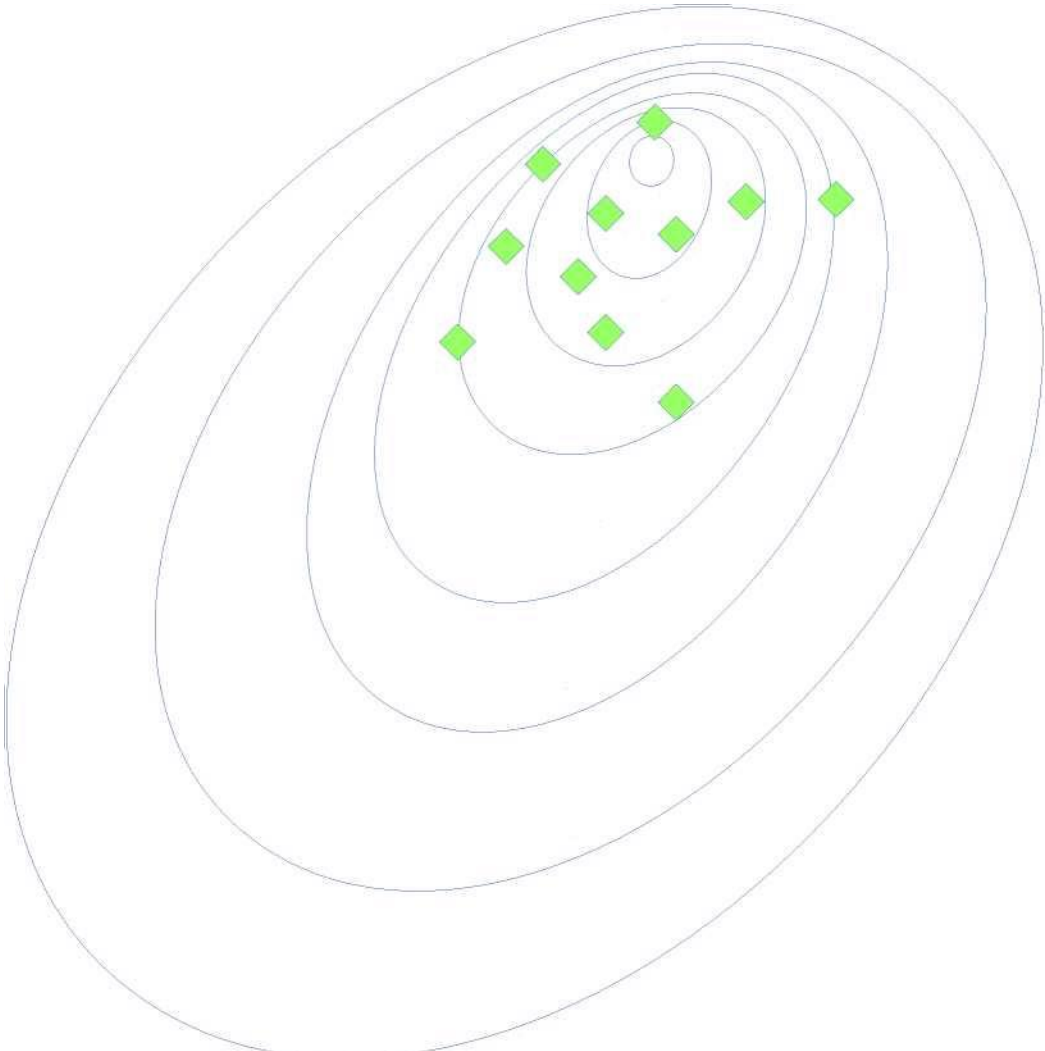
Пример:



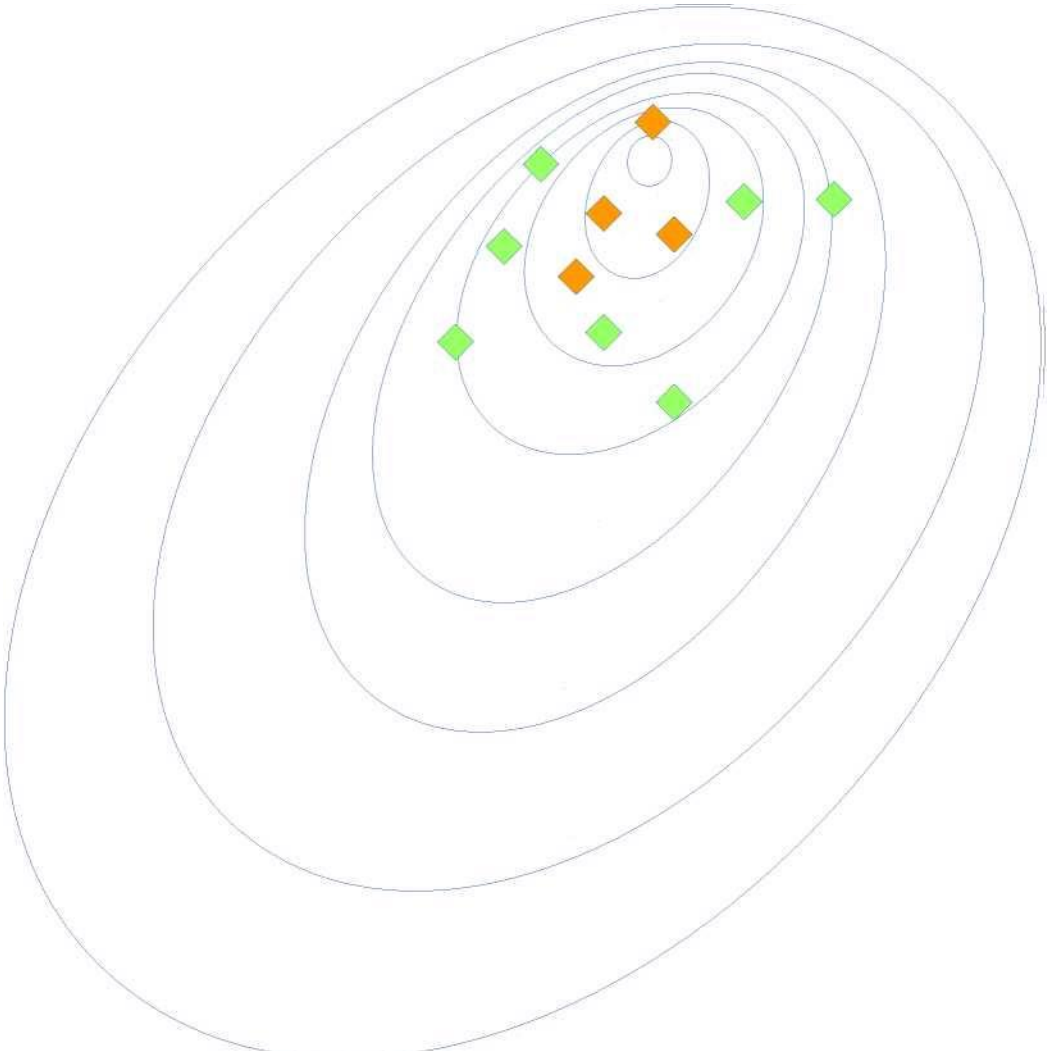
Пример:



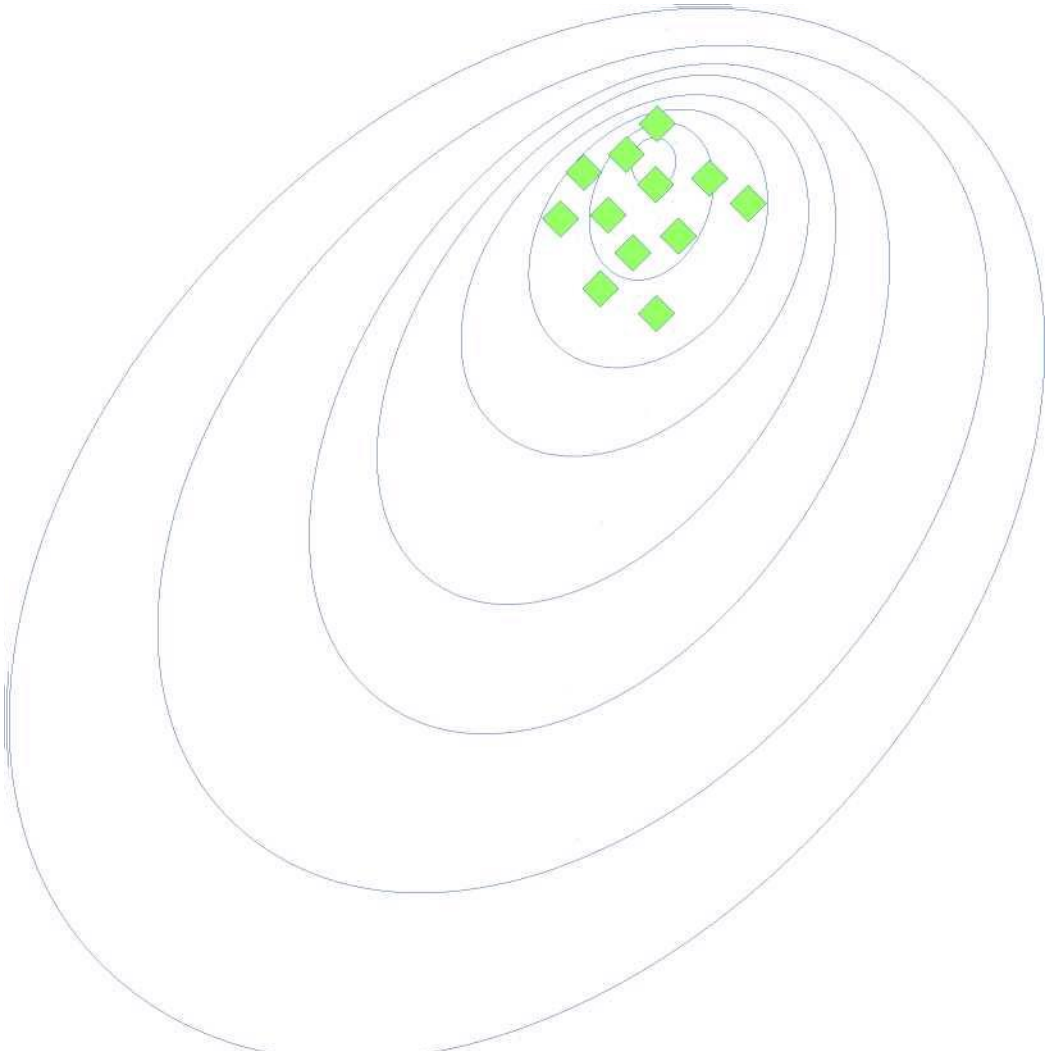
Пример:



Пример:



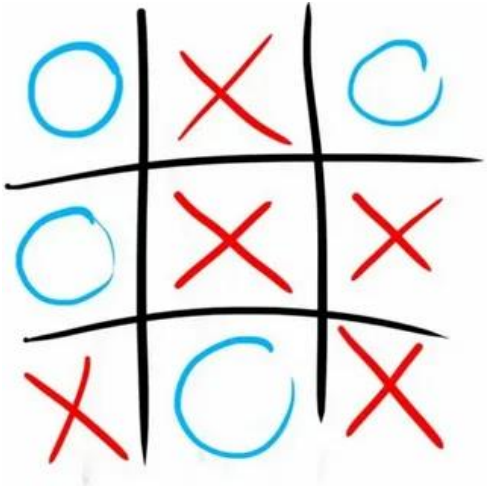
Пример:



Табличный кросс энтропийный метод

Политика: $\pi(a|s) = A_{s,a},\ A \in R^{n \times k}$

Переход: $\pi(a|s) = \frac{\sum\limits_{s_t, a_t \in Elite} [s_t = s][a_t = a]}{\sum\limits_{s_t, a_t \in Elite} [s_t = s]}$



Табличный метод и реальность:

Метод кросс энтропии с апроксимацией нейросетями
Алгоритм

1) Инициализируем модель случайными весами (W_0)
2) Запускаем N сессий
3) Выбираем m лучших(elite)
4) $W_{i+1} = W_i + \alpha\Delta[\sum_{s_i,a_i \in Elite} \log \pi_{w_i}(a_i|s_i)]$

v и q функции

$$v_\pi(s) = E_\pi[R_t|s_t = s] = E_{a\sim\pi}E_{s'\sim p(*|s,a)}[r(s,a) + \gamma v_\pi(s')]$$

$$q_\pi(s,a) = E_\pi[R_t|s_t = s, a_t = a] = E_{s'}[r(s,a) + \gamma v_\pi(s')]$$

$$v_*(s) = \max_a E_{s'}[r + \gamma v_*(s')]$$

$$q_*(s,a) = E_{s'}[r + \max_a \gamma q_*(s',a')]]$$

Общий алгоритм улучшения:

1) Инициализируем π
2) v = evaluate_policy(π)
3) new π = improve_policy(v)
4) Если политика не изменилось - сошлись к оптимальной
5) Вернутся к 2 шагу

Policy iteration

```
def evaluate_policy(pi):
    for s in ALL_STATES:
        v(s) = v_function(s)

def improve_policy(pi):
    for s in ALL_STATES:
        policy(s) = argmax v(s)
```

Value iteration

```
def evaluate_policy(pi):
    for s in ONE_STATE:
        v(s) = v_function(s)

def improve_policy(pi):
    for s in ONE_STATE:
        policy(s) = argmax v(s)
```

Supervised vs Reinforcement

Обучение с учителем (supervised)

- Учимся приближать заранее размеченный ответ
- Нужна разметка данных
- Модель не изменяет входные данные/среду

Обучение с подкреплением (reinforcement)

- Подбираем оптимальную стратегию путем проб и ошибок
- Нужен фидбэк для действий агента
- Агент влияет на среду

Используемые материалы:

Курс ШАДа по RL: https://github.com/yandexdataschool/Practical_RL

Richard Sutton: RL book http://incompleteideas.net/book/RLbook2020.pdf

Спасибо за внимание