AML, 15 Oct 2019

Reinforcement learning

Supervised and unsupervised learning

↓ labeled training data

↘ Look for patterns

RL - progress is based on rewards

e.g. the basis of AlphaGo

States (of the world), Actions available

States, Actions

state, action $\longrightarrow$ "reward" probabilistic

Goal: maximize rewards over a sequence of steps

Challenge: Estimate the rewards

Strategy to choose an action in a given state -

"Policy"

States, actions

Policy : In a state, what action to choose

Reward : Immediate feedback of choosing a given action at a state

Value : Long term estimated reward at a state

Model : Of the "environment"

Strength of RL is that model is optional

Situations

Game playing

Motion planning

"Feedback control" - e.g. balancing an object

Generally - we have a current estimate of rewards

- Choose best reward action "greedy"
- Choose to improve our knowledge of non-maximal rewards

Exploration vs Exploitation tradeoff

Exploration → Search for new action/reward info

Exploitation → Choose best reward known

Probabilistic setting

- Rewards are probabilistic $N(\mu, \sigma)$
- Policies may be probabilistic

$$\arg\max_a \text{ Estimated Reward}(a)$$ Exploitation

1. Multiple maxima – choose randomly

2. ε – explore

   $1-\varepsilon$ – exploit

Simplest concrete setting – only one state

One state, some actions, reward for each action, static in time

k-armed bandit problem

Each arm i has reward $R_i = N(\mu_i, \underset{=1}{\sigma_i})$

For i in $1, 2, \ldots, k$

Choose $m_i \in N(0,1)$

Set $R_i = N(m_i, 1)$

At time $t = 1, 2, \ldots$ we select action A_t

Corresponding reward is R_t

Estimates are $q_*(a) \stackrel{\text{def}}{=} \mathbb{E}[R_t | A_t = a]$

By repeatedly choosing a, we get a good estimate of its mean

$$Q_t(a) = \frac{\text{sum of rewards for } a \text{ at time} \leq t}{\# \text{ of times we choose } a \leq t}$$

Greedy strategy

Choose $A_t = \underset{a}{\operatorname{argmax}}\ Q_t(a)$

Instead - ε-greedy

Choose argmax with prob $1-\varepsilon$

Randomly choose non-max with prob. ε

[Graphs with $\varepsilon = 0, 0.01, 0.1$]

Seen action a n times

$$Q_{n+1} = \frac{1}{n} \sum_{i=1}^{n} R_i$$

$$= \frac{1}{n} \left[R_n + \sum_{i=1}^{n-1} R_i \right]$$

$$Q_{n+1} = \frac{1}{n}\left[R_n + (n-1)\underbrace{\frac{1}{n-1}\sum_{i=1}^{n-1} R_i}_{Q_n}\right]$$

$$Q_{n+1} = \frac{1}{n}\left[R_n + nQ_n - Q_n\right]$$

$$= Q_n + \frac{1}{n}\left[R_n - Q_n\right]$$

New Estimate = Old estimate + α [Diff]

decreasing with time $\frac{1}{n}$

Non stationary case - reward varies over time

$$Q_{n+1} = Q_n + \alpha [R_n - Q_n]$$

$$= \alpha R_n + (1-\alpha) \underline{Q_n}$$

$$\alpha R_{n-1} + (1-\alpha) Q_{n-1}$$

$$= \alpha R_n + \alpha(1-\alpha) R_{n-1} + (1-\alpha)^2 Q_{n-1}$$

$$\vdots$$

$$= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1-\alpha)^{n-i} R_i$$

Check that $(1-\alpha)^n + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} = 1$

$$\underbrace{(1-\alpha)^n}_{<1 \text{ — decay as } n \text{ increases}} Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

"Exponential recency-weighted average"

Non stationary – choose $\alpha_n = \alpha$ (constant)

vs $\frac{1}{n}$

Some strategies

Optimistic Initial Values

Choose large initial estimates

With high probability each early greedy choice produces a lower than expected reward

- Reduce estimate
- Forces exploration
- Converge faster

But only works if stationary

Systematic Exploration

Non greedy - choose an action uniformly

Instead

$$\underset{a}{\operatorname{argmax}} \left[\underbrace{Q_t(a)}_{\text{estimate}} + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

time

# of a so far

Actions not picked often get a higher chance of being explored