Reinforcement Learning

Pranabendu Misra
based on slides by Madhavan Mukund.

Advanced Machine Learning 2022

An alternative approach to learning

- Supervised learning — use labelled examples to learn a classifier

An alternative approach to learning

- Supervised learning — use labelled examples to learn a classifier
- Unsupervised learning — search for patterns, structure in data

An alternative approach to learning

- Supervised learning — use labelled examples to learn a classifier
- Unsupervised learning — search for patterns, structure in data
- Reinforcement learning — learning through interaction
  - Choose actions in an uncertain environment
  - Actions change state, yield rewards
  - Learn optimal strategies to maximize long term rewards

An alternative approach to learning

- Supervised learning — use labelled examples to learn a classifier
- Unsupervised learning — search for patterns, structure in data
- Reinforcement learning — learning through interaction
  - Choose actions in an uncertain environment
  - Actions change state, yield rewards
  - Learn optimal strategies to maximize long term rewards
- Examples
  - Playing games — AlphaGo, reward is result of the game
  - Motion planning — robot searching for an optimal path with obstacles
  - Feedback control — balancing an object

The components

- Policy What action to take in the current state
  - “Strategy”, can be probabilistic

The components

- Policy What action to take in the current state
  - “Strategy”, can be probabilistic
- Reward In response to taking an action
  - Short-term outcome, may be negative or positive

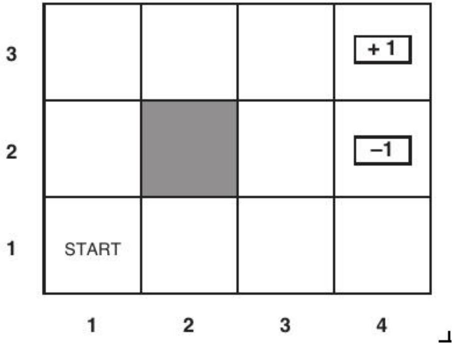
The components

- Policy What action to take in the current state
  - "Strategy", can be probabilistic
- Reward In response to taking an action
  - Short-term outcome, may be negative or positive
- Value Accumulation of rewards over future actions
  - Long-term outcome, goal is to maximize value

The components

- Policy What action to take in the current state
  - "Strategy", can be probabilistic
- Reward In response to taking an action
  - Short-term outcome, may be negative or positive
- Value Accumulation of rewards over future actions
  - Long-term outcome, goal is to maximize value
- Environment Model How the environment will behave
  - Given a state and action, what is the next state, reward?
  - Probabilistic, in general
  - Use models for *planning*
  - Can also use RL without models, trial-and-error learners

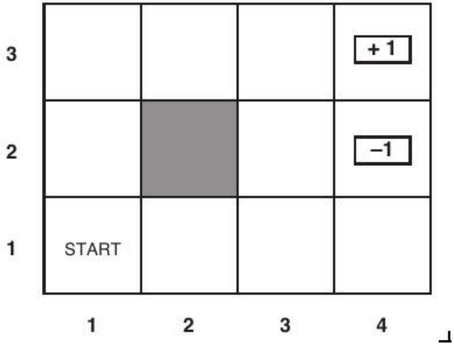
- Rewards are attached to states
  - Two terminal states with rewards $+1$, -1
  - All other states have reward -0.04
  - Move till you reach a terminal state
  - Maximize the sum of the rewards seen



Moving in a $4x3$ grid
from the START state
to one of the two terminal state.

Motion planning example

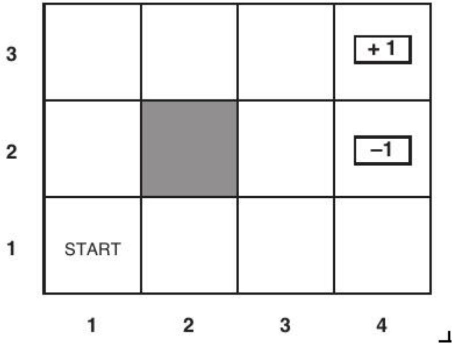
- Rewards are attached to states
  - Two terminal states with rewards $+1$, -1
  - All other states have reward -0.04
  - Move till you reach a terminal state
  - Maximize the sum of the rewards seen
- Policy — which direction to move from a given square in the grid

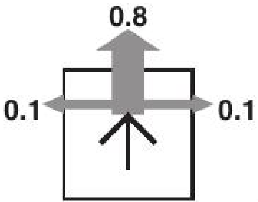


Moving in a $4x3$ grid
from the START state
to one of the two terminal state.

Motion planning example

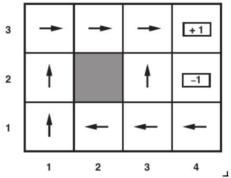
- Rewards are attached to states
  - Two terminal states with rewards $+1$, -1
  - All other states have reward -0.04
  - Move till you reach a terminal state
  - Maximize the sum of the rewards seen
- Policy — which direction to move from a given square in the grid
- Outcome of action is non-deterministic. Unintended things could happen.
  - With probability 0.8, go in intended direction
  - With probability 0.2, deflect at right angles
  - Collision with boundary keeps you stationary





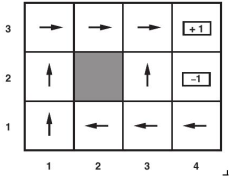
Motion planning example

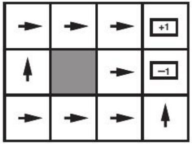
- Optimal policy learned by repeatedly moving on the board
  - From bottom right, conservatively follow the long route around the obstacle to avoid -1



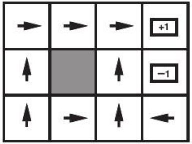
Motion planning example

- Optimal policy learned by repeatedly moving on the board
  - From bottom right, conservatively follow the long route around the obstacle to avoid -1
- $R(s)$: reward for non-final states s
  - If $R(s) < -1.6284$, terminate as fast as possible
  - If $-0.4278 < R(s) < -0.0850$, risk going past -1 to reach $+1$ quickly
  - If $-0.0221 < R(s) < 0$, take no risks, avoid -1 at all cost
  - $R(s) = 0$ is shown above
  - If $R(s) > 0$ avoid terminating

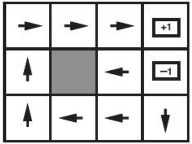




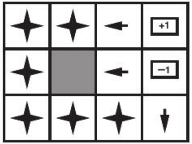
$R(s) < -1.6284$



$-0.4278 < R(s) < -0.0850$



$-0.0221 < R(s) < 0$



$R(s) > 0$

Exploration vs exploitation

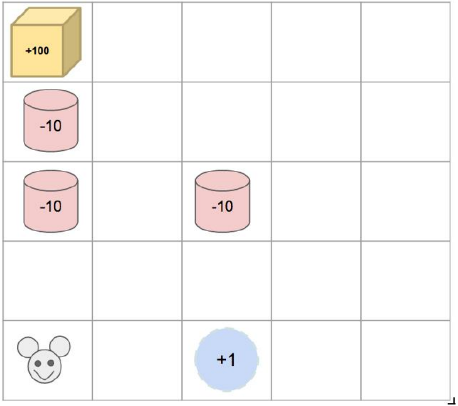
- Policy evolves by experience

Exploration vs exploitation

- Policy evolves by experience
- Greedy strategy is to always choose best known option

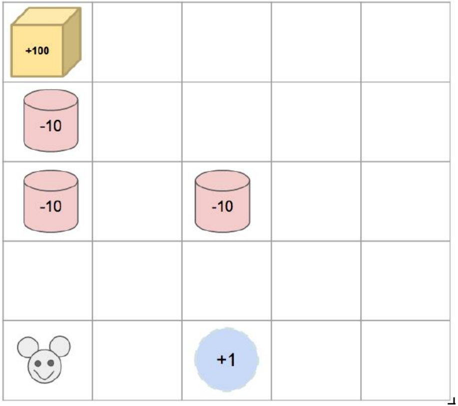
Exploration vs exploitation

- Policy evolves by experience
- Greedy strategy is to always choose best known option
- Using this we may get stuck in a local optimum
  - Greedy strategy only allows the mouse to discover water with reward $+1$
  - Mouse never discovers a path to cheese with $+100$ because of negative rewards en route



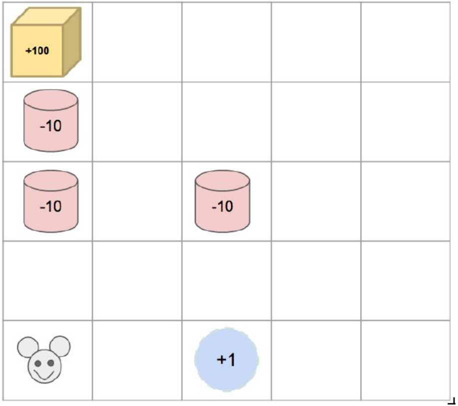
Exploration vs exploitation

- Policy evolves by experience
- Greedy strategy is to always choose best known option
- Using this we may get stuck in a local optimum
  - Greedy strategy only allows the mouse to discover water with reward $+1$
  - Mouse never discovers a path to cheese with $+100$ because of negative rewards en route
- How to balance exploitation (greedy) vs exploration?



Exploration vs exploitation

- Policy evolves by experience
- Greedy strategy is to always choose best known option
- Using this we may get stuck in a local optimum
  - Greedy strategy only allows the mouse to discover water with reward $+1$
  - Mouse never discovers a path to cheese with $+100$ because of negative rewards en route
- How to balance exploitation (greedy) vs exploration?
- Formalize these ideas using Markov Decision Processes



Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution

Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution
- k-armed bandit
  - Each arm has a different reward probability
  - Goal is to maximize total reward over a sequence of plays

Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution
- k-armed bandit
  - Each arm has a different reward probability
  - Goal is to maximize total reward over a sequence of plays
- Action corresponds to choosing the arm

Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution
- k-armed bandit
  - Each arm has a different reward probability
  - Goal is to maximize total reward over a sequence of plays
- Action corresponds to choosing the arm
  - For each action a, $q_*(a)$ is expected reward if we choose a

Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution
- k-armed bandit
  - Each arm has a different reward probability
  - Goal is to maximize total reward over a sequence of plays
- Action corresponds to choosing the arm
  - For each action a, $q_*(a)$ is expected reward if we choose a
  - A_t is action chosen at time t, with reward R_t

Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution
- k-armed bandit
  - Each arm has a different reward probability
  - Goal is to maximize total reward over a sequence of plays
- Action corresponds to choosing the arm
  - For each action a, $q_*(a)$ is expected reward if we choose a
  - A_t is action chosen at time t, with reward R_t
  - If we knew $q_*(a)$ we would always choose $A_t = \arg\max_a q_*(a)$

Bandits

- One-armed bandit — slang for a slot machine in a casino
  - Put in a coin and pull a lever (the arm)
  - With high probability, lose your coin (the bandit steals your money)
  - With low probability, get varying reward, rewards follow some probability distribution
- k-armed bandit
  - Each arm has a different reward probability
  - Goal is to maximize total reward over a sequence of plays
- Action corresponds to choosing the arm
  - For each action a, $q_*(a)$ is expected reward if we choose a
  - A_t is action chosen at time t, with reward R_t
  - If we knew $q_*(a)$ we would always choose $A_t = \arg\max_a q_*(a)$
  - Assume $q_*(a)$ is unknown — build an estimate $Q_t(a)$ of $q_*(a)$ at time t

Exploration and exploitation

- Build $Q_t(a)$, estimate of $q_*(a)$ at time t, from past observations (sample average)

$$\frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$$

Exploration and exploitation

- Build $Q_t(a)$, estimate of $q_*(a)$ at time t, from past observations (sample average)

$$\frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$$

- **Greedy policy** chooses $\arg\max_a Q_t(a)$
- But it fails to explore other options.
  How will we learn about all actions?

Exploration and exploitation

- Build $Q_t(a)$, estimate of $q_*(a)$ at time t, from past observations (sample average)

$$\frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$$

- **Greedy policy** chooses $\arg\max_a Q_t(a)$
- But it fails to explore other options.
  How will we learn about all actions?
- ε-greedy policy
  - With small probability ε, choose a random action (uniform distribution)
  - With probability $1 - \varepsilon$, follow greedy

Exploration and exploitation

- Build $Q_t(a)$, estimate of $q_*(a)$ at time t, from past observations (sample average)

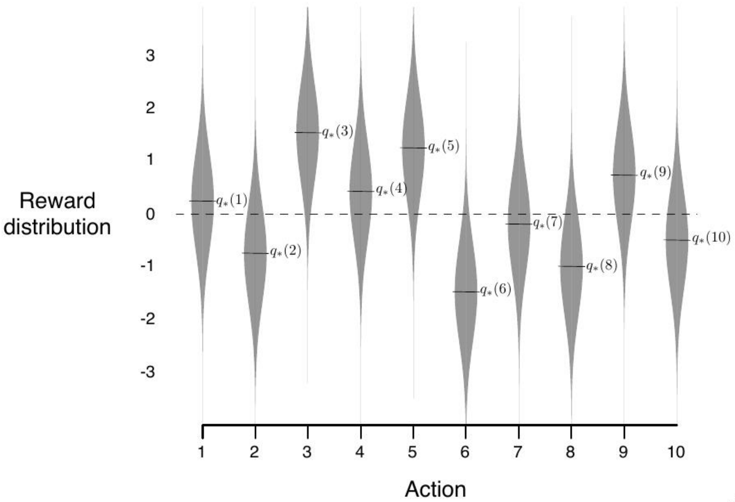
$$\frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$$

- **Greedy policy** chooses $\arg\max_a Q_t(a)$
- But it fails to explore other options.
  How will we learn about all actions?
- ε-greedy policy
  - With small probability ε, choose a random action (uniform distribution)
  - With probability $1 - \varepsilon$, follow greedy
- ε-greedy is a simple way to balance exploitation with exploration
  - Theoretically, explores all actions infinitely often
  - Practical effectiveness depends

Exploration and exploitation

10 bandit experiment

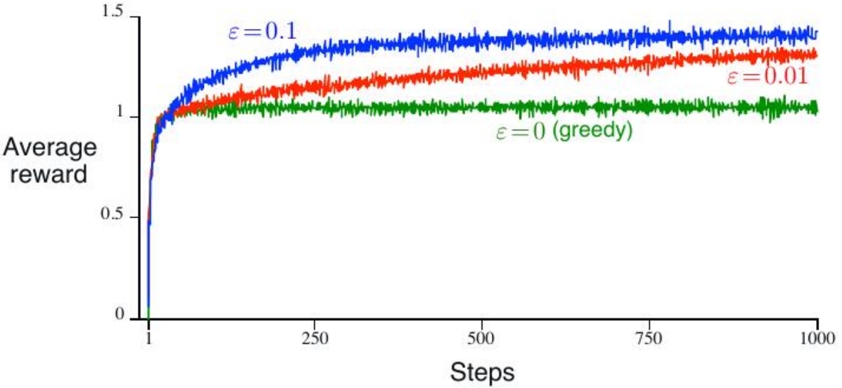
- Each bandit's reward follows Gaussian distribution
- Same variance, mean is chosen randomly



Exploration and exploitation

Performance of ε-greedy strategies

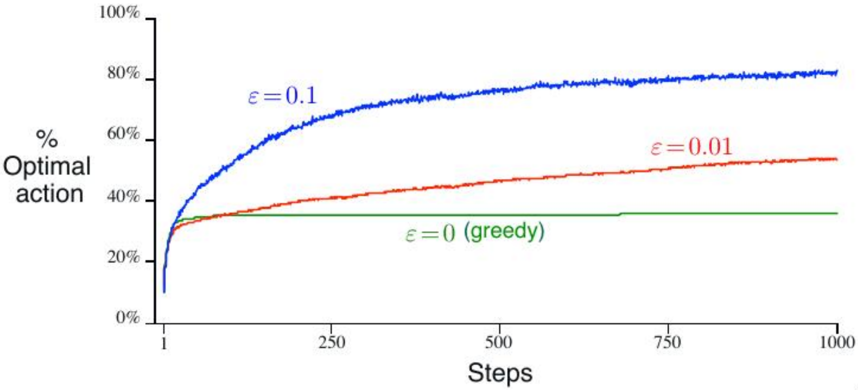
- Pure greedy strategy is sub-optimal
- Initial "learning rate" is more or less equal



Exploration and exploitation

Discovery of optimal actions

- Pure greedy strategy discovers optimal action only 1/3 of the time



Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \dfrac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $Q_{n+1} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} R_i$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \dfrac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $Q_{n+1} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} R_i \;\; = \dfrac{1}{n}\left(R_n + \displaystyle\sum_{i=1}^{n-1} R_i\right)$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \dfrac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $Q_{n+1} = \dfrac{1}{n}\sum_{i=1}^{n} R_i = \dfrac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) = \dfrac{1}{n}\left(R_n + (n-1)\dfrac{1}{n-1}\sum_{i=1}^{n-1} R_i\right)$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \dfrac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) = \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right)$$
  $$= \frac{1}{n}(R_n + (n-1)Q_n)$$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \dfrac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) = \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right)$$
$$= \frac{1}{n}(R_n + (n-1)Q_n) = \frac{1}{n}(R_n + nQ_n - Q_n)$$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) = \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right)$$
  $$= \frac{1}{n}(R_n + (n-1)Q_n) = \frac{1}{n}(R_n + nQ_n - Q_n) = Q_n + \frac{1}{n}[R_n - Q_n]$$

Sidenote: Efficient Incremental calculation of Estimated Rewards

- Focus on a single action a. Sample average is $\frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$
- R_i — reward when a is selected for ith time
- Q_n — estimate of action value after a has been selected $n-1$ times
- $Q_n = \dfrac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$
- $$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) = \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right)$$
  $$= \frac{1}{n}\left(R_n + (n-1)Q_n\right) = \frac{1}{n}\left(R_n + nQ_n - Q_n\right) = Q_n + \frac{1}{n}\left[R_n - Q_n\right]$$
- We will see this pattern often:

  NewEstimate = OldEstimate + Step [Target - OldEstimate]

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$
- $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$
- $Q_{n+1} = Q_n + \alpha[R_n - Q_n] \; = \alpha R_n + (1 - \alpha)Q_n$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$
- $Q_{n+1} = Q_n + \alpha[R_n - Q_n] \; = \alpha R_n + (1 - \alpha) Q_n$
  $= \alpha R_n + (1 - \alpha)[\alpha R_{n-1} + (1 - \alpha) Q_{n-1}]$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$
- $$\begin{aligned} Q_{n+1} &= Q_n + \alpha[R_n - Q_n] \;\; = \alpha R_n + (1-\alpha)Q_n \\ &= \alpha R_n + (1-\alpha)[\alpha R_{n-1} + (1-\alpha)Q_{n-1}] \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + (1-\alpha)^2 Q_{n-1} \end{aligned}$$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$
- $$\begin{aligned} Q_{n+1} &= Q_n + \alpha[R_n - Q_n] \; = \alpha R_n + (1-\alpha)Q_n \\ &= \alpha R_n + (1-\alpha)[\alpha R_{n-1} + (1-\alpha)Q_{n-1}] \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + (1-\alpha)^2 Q_{n-1} \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + \alpha(1-\alpha)^2 R_{n-2} + \cdots + \alpha(1-\alpha)^{n-1}R_1 + (1-\alpha)^n Q_1 \end{aligned}$$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time

- Assume the distribution changes gradually.
  This means recent rewards are more important.

- Exponentially decaying weighted average of rewards

- Use a constant step $\alpha \in (0,1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$

- $$\begin{aligned} Q_{n+1} &= Q_n + \alpha[R_n - Q_n] \;= \alpha R_n + (1-\alpha)Q_n \\ &= \alpha R_n + (1-\alpha)[\alpha R_{n-1} + (1-\alpha)Q_{n-1}] \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + (1-\alpha)^2 Q_{n-1} \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + \alpha(1-\alpha)^2 R_{n-2} + \cdots + \alpha(1-\alpha)^{n-1}R_1 + (1-\alpha)^n Q_1 \\ &= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i \end{aligned}$$

Stationary vs non-stationary Rewards

- *Non-stationary Rewards*: The Reward probability distribution change over time
- Assume the distribution changes gradually.
  This means recent rewards are more important.
- Exponentially decaying weighted average of rewards
- Use a constant step $\alpha \in (0, 1]$ — $Q_{n+1} = Q_n + \alpha[R_n - Q_n]$
- $$\begin{aligned} Q_{n+1} &= Q_n + \alpha[R_n - Q_n] \;\; = \alpha R_n + (1-\alpha)Q_n \\ &= \alpha R_n + (1-\alpha)[\alpha R_{n-1} + (1-\alpha)Q_{n-1}] \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + (1-\alpha)^2 Q_{n-1} \\ &= \alpha R_n + \alpha(1-\alpha)R_{n-1} + \alpha(1-\alpha)^2 R_{n-2} + \cdots + \alpha(1-\alpha)^{n-1}R_1 + (1-\alpha)^n Q_1 \\ &= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i \end{aligned}$$
- Initial value Q_1 affects the calculation — different heuristics possible

Summary

- k-armed bandit is the simplest interesting situation to analyze
- ε-greedy strategy balances exploration and exploitation
- Incremental update rule for estimates

  $$\text{NewEstimate} = \text{OldEstimate} + \text{Step}\ [\text{Target - OldEstimate}]$$

- Exponentially decaying weighted average when rewards change over time (non-stationary)