UCB-VI and Contextual Bandits

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning

Fall 2023

Today

- Recap
- UCB-VI for tabular MDPs
- UCB-VI for linear MDPs
- Contextual bandits intro

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

$$Q_{H-1}^\star(s,a) = r(s,a)$$

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

$$Q_{H-1}^\star(s,a) = r(s,a) \qquad \pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a)$$

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H - 1$,

$$Q_{H-1}^\star(s, a) = r(s, a) \qquad \pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s, a)$$

$$V_{H-1}^\star = \max_a Q_{H-1}^\star(s, a) = Q_{H-1}^\star(s, \pi_{H-1}^\star(s))$$

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

$$Q_{H-1}^\star(s,a) = r(s,a) \qquad \pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a)$$

$$V_{H-1}^\star = \max_a Q_{H-1}^\star(s,a) = Q_{H-1}^\star(s, \pi_{H-1}^\star(s))$$

2. Assuming we have computed $V_{h+1}^\star$, $h \le H-2$, i.e., assuming we know how to perform optimally starting at $h+1$, then:

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

$$Q_{H-1}^\star(s,a) = r(s,a) \qquad \pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a)$$

$$V_{H-1}^\star = \max_a Q_{H-1}^\star(s,a) = Q_{H-1}^\star(s, \pi_{H-1}^\star(s))$$

2. Assuming we have computed $V_{h+1}^\star$, $h \leq H-2$, i.e., assuming we know how to perform optimally starting at $h+1$, then:

$$Q_h^\star(s,a) = r(s,a) + \mathbb{E}_{s' \sim P(s,a)} V_{h+1}^\star(s')$$

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

$$Q_{H-1}^\star(s,a) = r(s,a) \qquad \pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a)$$

$$V_{H-1}^\star = \max_a Q_{H-1}^\star(s,a) = Q_{H-1}^\star(s, \pi_{H-1}^\star(s))$$

2. Assuming we have computed $V_{h+1}^\star$, $h \leq H-2$, i.e., assuming we know how to perform optimally starting at $h+1$, then:

$$Q_h^\star(s,a) = r(s,a) + \mathbb{E}_{s' \sim P(s,a)} V_{h+1}^\star(s')$$

$$\pi_h^\star(s) = \arg\max_a Q_h^\star(s,a),$$

Recall: Value Iteration (VI)

VI = DP is a backwards in time approach for computing the optimal policy:

$$\pi^\star = \{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\}$$

1. Start at $H-1$,

$$Q_{H-1}^\star(s,a) = r(s,a) \qquad \pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a)$$

$$V_{H-1}^\star = \max_a Q_{H-1}^\star(s,a) = Q_{H-1}^\star(s, \pi_{H-1}^\star(s))$$

2. Assuming we have computed $V_{h+1}^\star$, $h \le H-2$, i.e., assuming we know how to perform optimally starting at $h+1$, then:

$$Q_h^\star(s,a) = r(s,a) + \mathbb{E}_{s' \sim P(s,a)} V_{h+1}^\star(s')$$

$$\pi_h^\star(s) = \arg\max_a Q_h^\star(s,a), \qquad V_h^\star = \max_a Q_h^\star(s,a)$$

Recall: UCB

For $t = 0, \ldots, T-1$:

Choose the arm with the highest upper confidence bound, i.e.,

$$a_t = \arg \max_{k \in \{1,\ldots,K\}} \hat{\mu}_t^{(k)} + \sqrt{\ln(2TK/\delta)/2N_t^{(k)}}$$

Recall: UCB

For $t = 0,\ldots,T-1$:

Choose the arm with the highest upper confidence bound, i.e.,

$$a_t = \arg\max_{k\in\{1,\ldots,K\}} \hat{\mu}_t^{(k)} + \sqrt{\ln(2TK/\delta)/2N_t^{(k)}}$$

High-level summary: estimate action quality, add exploration bonus, then argmax

UCBVI: Tabular optimism in the face of uncertainty

Assume reward function $r_h(s, a)$ known

Inside iteration n :

UCBVI: Tabular optimism in the face of uncertainty

Assume reward function $r_h(s, a)$ known

Inside iteration n :

Use all previous data to estimate transitions $\hat{P}_1^n, \ldots, \hat{P}_{H-1}^n$

UCBVI: Tabular optimism in the face of uncertainty

Assume reward function $r_h(s, a)$ known

Inside iteration n :

Use all previous data to estimate transitions $\hat{P}_1^n, \ldots, \hat{P}_{H-1}^n$

Design reward bonus $b_h^n(s, a), \forall s, a, h$

UCBVI: Tabular optimism in the face of uncertainty

Assume reward function $r_h(s,a)$ known

Inside iteration n :

Use all previous data to estimate transitions $\hat{P}_1^n, \ldots, \hat{P}_{H-1}^n$

Design reward bonus $b_h^n(s,a), \forall s, a, h$

Optimistic planning with learned model: $\pi^n = \text{VI}\left(\{\hat{P}_h^n, r_h + b_h^n\}_{h=1}^{H-1}\right)$

UCBVI: Tabular optimism in the face of uncertainty

Assume reward function $r_h(s, a)$ known

Inside iteration n :

Use all previous data to estimate transitions $\hat{P}_1^n, \ldots, \hat{P}_{H-1}^n$

Design reward bonus $b_h^n(s, a), \forall s, a, h$

Optimistic planning with learned model: $\pi^n = \text{VI}\left(\{\hat{P}_h^n, r_h + b_h^n\}_{h=1}^{H-1} \right)$

Collect a new trajectory by executing π^n in the true system $\{P_h\}_{h=0}^{H-1}$ starting from s_0

Model Estimation

Let us consider the **very beginning** of episode n:

$$\mathscr{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h$$

Model Estimation

Let us consider the **very beginning** of episode n:

$$\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h$$

Let's also maintain some statistics using these datasets:

Model Estimation

Let us consider the **very beginning** of episode n:

$$\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h$$

Let's also maintain some statistics using these datasets:

$$N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s,a,h, \quad N_h^n(s,a,s') = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i, s_{h+1}^i) = (s,a,s')\}, \forall s,a,h$$

Model Estimation

Let us consider the **very beginning** of episode n:

$$\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h$$

Let's also maintain some statistics using these datasets:

$$N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h, \quad N_h^n(s,a,s') = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i, s_{h+1}^i) = (s,a,s')\}, \forall s, a, h$$

Estimate model $\hat{P}_h^n(s' \mid s, a), \forall s, a, s', h$:

$$\hat{P}_h^n(s' \mid s, a) = \frac{N_h^n(s,a,s')}{N_h^n(s,a)}$$

Today

- Recap
- UCB-VI for tabular MDPs
- UCB-VI for linear MDPs
- Contextual bandits intro

Reward Bonus Design and Value Iteration

Recall: $\mathscr{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Reward Bonus Design and Value Iteration

Recall: $\mathscr{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = H\sqrt{\frac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Reward Bonus Design and Value Iteration

Recall: $\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\frac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Reward Bonus Design and Value Iteration

Recall: $\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\dfrac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Value Iteration (aka DP) at episode n using $\{\hat{P}_h^n\}_h$ and $\{r_h + b_h^n\}_h$

Reward Bonus Design and Value Iteration

Recall: $\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\dfrac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Value Iteration (aka DP) at episode n using $\{\hat{P}_h^n\}_h$ and $\{r_h + b_h^n\}_h$

$\hat{V}_H^n(s) = 0, \forall s$

Reward Bonus Design and Value Iteration

Recall: $\mathscr{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\dfrac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Value Iteration (aka DP) at episode n using $\{\hat{P}_h^n\}_h$ and $\{r_h + b_h^n\}_h$

$$\hat{V}_H^n(s) = 0, \forall s \qquad \hat{Q}_h^n(s,a) = \min\left\{ r_h(s,a) + b_h^n(s,a) + \mathbb{E}_{s' \sim \hat{P}_h^n(\cdot|s,a)}\left[\hat{V}_{h+1}^n(s')\right], \quad H \right\}, \forall s, a$$

Reward Bonus Design and Value Iteration

Recall: $\mathscr{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\frac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Value Iteration (aka DP) at episode n using $\{\hat{P}_h^n\}_h$ and $\{r_h + b_h^n\}_h$

$$\hat{V}_H^n(s) = 0, \forall s \qquad \hat{Q}_h^n(s,a) = \min\left\{ r_h(s,a) + b_h^n(s,a) + \mathbb{E}_{s' \sim \hat{P}_h^n(\cdot|s,a)}\left[\hat{V}_{h+1}^n(s')\right], \quad H \right\}, \forall s, a$$

$$\hat{V}_h^n(s) = \max_a \hat{Q}_h^n(s,a), \quad \pi_h^n(s) = \arg\max_a \hat{Q}_h^n(s,a), \forall s$$

Reward Bonus Design and Value Iteration

Recall: $\mathscr{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\frac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Value Iteration (aka DP) at episode n using $\{\hat{P}_h^n\}_h$ and $\{r_h + b_h^n\}_h$

$$\hat{V}_H^n(s) = 0, \forall s \qquad \hat{Q}_h^n(s,a) = \min\left\{ r_h(s,a) + b_h^n(s,a) + \mathbb{E}_{s' \sim \hat{P}_h^n(\cdot|s,a)}\left[\hat{V}_{h+1}^n(s')\right], \quad H \right\}, \forall s, a$$

$$\hat{V}_h^n(s) = \max_a \hat{Q}_h^n(s,a), \quad \pi_h^n(s) = \arg\max_a \hat{Q}_h^n(s,a), \forall s$$

$$\left\| \hat{V}_h^n \right\|_\infty \leq H, \forall h, n$$

Reward Bonus Design and Value Iteration

Recall: $\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=1}^{n-1}, \forall h, \quad N_h^n(s,a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}, \forall s, a, h,$

Define: $b_h^n(s,a) = cH\sqrt{\frac{\log\left(|S||A|HN/\delta\right)}{N_h^n(s,a)}}$

Encourage to explore new state-actions

Value Iteration (aka DP) at episode n using $\{\hat{P}_h^n\}_h$ and $\{r_h + b_h^n\}_h$

$$\hat{V}_H^n(s) = 0, \forall s \qquad \hat{Q}_h^n(s,a) = \min\left\{ r_h(s,a) + b_h^n(s,a) + \mathbb{E}_{s' \sim \hat{P}_h^n(\cdot|s,a)}\left[\hat{V}_{h+1}^n(s')\right], \quad H \right\}, \forall s, a$$

$$\hat{V}_h^n(s) = \max_a \hat{Q}_h^n(s,a), \quad \pi_h^n(s) = \arg\max_a \hat{Q}_h^n(s,a), \forall s$$

$$\left\| \hat{V}_h^n \right\|_\infty \leq H, \forall h, n$$

$b_h^n(s,a)$ specifically chosen so that $V_h^\star(s) \leq \hat{V}_h^n(s)$ with high probability

UCBVI: Put All Together

For $n = 1 \to N$:

1. Set $N_h^n(s, a) = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i) = (s, a)\}, \forall s, a, h$

2. Set $N_h^n(s, a, s') = \sum_{i=1}^{n-1} \mathbf{1}\{(s_h^i, a_h^i, s_{h+1}^i) = (s, a, s')\}, \forall s, a, a', h$

3. Estimate $\hat{P}^n$: $\hat{P}_h^n(s' \mid s, a) = \dfrac{N_h^n(s, a, s')}{N_h^n(s, a)}, \forall s, a, s', h$

4. Plan: $\pi^n =$ VI $\left(\{\hat{P}_h^n, r_h + b_h^n\}_h \right)$, with $b_h^n(s, a) = cH\sqrt{\dfrac{\log(|S||A|HN/\delta)}{N_h^n(s, a)}}$

5. Execute π^n : $\{s_0^n, a_0^n, r_0^n, \ldots, s_{H-1}^n, a_{H-1}^n, r_{H-1}^n, s_H^n\}$

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

1. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is small?

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

1. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is small?

Then π^n is close to $\pi^\star$, i.e., we are doing exploitation

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

1. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is small?

Then π^n is close to $\pi^\star$, i.e., we are doing exploitation

2. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is large?

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

1. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is small?

Then π^n is close to $\pi^\star$, i.e., we are doing exploitation

2. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is large?

Some $b_h^n(s,a)$ must be large (or some $\hat{P}_h^n(\cdot \mid s,a)$ estimation errors must be large, but with high probability any $\hat{P}_h^n(\cdot \mid s,a)$ with high error must have small $N_h^n(s,a)$ and hence high $b_h^n(s,a)$)

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

1. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is small?

Then π^n is close to $\pi^\star$, i.e., we are doing exploitation

2. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is large?

Some $b_h^n(s, a)$ must be large (or some $\hat{P}_h^n(\cdot \mid s, a)$ estimation errors must be large, but with high probability any $\hat{P}_h^n(\cdot \mid s, a)$ with high error must have small $N_h^n(s, a)$ and hence high $b_h^n(s, a)$)

Large $b_h^n(s, a)$ means π^n is being encouraged to do (s, a), since it will apparently have very high reward, i.e., exploration

High-level Idea: Exploration Exploitation Tradeoff

Upper bound per-episode regret: $V_0^\star(s_0) - V_0^{\pi^n}(s_0) \leq \hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ by construction of b_h^n

1. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is small?

Then π^n is close to $\pi^\star$, i.e., we are doing exploitation

2. What if $\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0)$ is large?

Some $b_h^n(s,a)$ must be large (or some $\hat{P}_h^n(\cdot \mid s,a)$ estimation errors must be large, but with high probability any $\hat{P}_h^n(\cdot \mid s,a)$ with high error must have small $N_h^n(s,a)$ and hence high $b_h^n(s,a)$)

Large $b_h^n(s,a)$ means π^n is being encouraged to do (s,a), since it will apparently have very high reward, i.e., exploration

$$\mathbb{E}\left[\text{Regret}_N\right] := \mathbb{E}\left[\sum_{n=1}^{N}\left(V^\star - V^{\pi^n}\right)\right] \leq \widetilde{O}\left(H^2\sqrt{SAN}\right)$$

Today

- ✓ Recap
- ✓ UCB-VI for tabular MDPs
- UCB-VI for linear MDPs
- Contextual bandits intro

Linear MDP Definition

Finite horizon time-dependent episodic MDP $\mathscr{M} = \{S, A, H, \{r\}_h, \{P\}_h, s_0\}$

S & A could be large or even continuous, hence poly($|S|, |A|$) is not acceptable

Linear MDP Definition

Finite horizon time-dependent episodic MDP $\mathcal{M} = \{S, A, H, \{r\}_h, \{P\}_h, s_0\}$

S & A could be large or even continuous, hence poly$(|S|, |A|)$ is not acceptable

$$P_h(s' | s, a) = \mu_h^\star(s') \cdot \phi(s, a), \quad \mu_h^\star : S \mapsto \mathbb{R}^d, \quad \phi : S \times A \mapsto \mathbb{R}^d$$

Linear MDP Definition

Finite horizon time-dependent episodic MDP $\mathscr{M} = \{S, A, H, \{r\}_h, \{P\}_h, s_0\}$

S & A could be large or even continuous, hence poly$(|S|, |A|)$ is not acceptable

$$P_h(s'|s,a) = \mu_h^\star(s') \cdot \phi(s,a), \quad \mu_h^\star : S \mapsto \mathbb{R}^d, \quad \phi : S \times A \mapsto \mathbb{R}^d$$

$$r(s,a) = \theta_h^\star \cdot \phi(s,a), \quad \theta_h^\star \in \mathbb{R}^d$$

Linear MDP Definition

Finite horizon time-dependent episodic MDP $\mathscr{M} = \{S, A, H, \{r\}_h, \{P\}_h, s_0\}$

S & A could be large or even continuous, hence poly$(|S|, |A|)$ is not acceptable

$$P_h(s'|s,a) = \mu_h^\star(s') \cdot \phi(s,a), \quad \mu_h^\star : S \mapsto \mathbb{R}^d, \quad \phi : S \times A \mapsto \mathbb{R}^d$$

$$r(s,a) = \theta_h^\star \cdot \phi(s,a), \quad \theta_h^\star \in \mathbb{R}^d$$

Feature map ϕ is known to the learner!

(We assume reward is known, i.e., $\theta^\star$ is known)

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \;\; \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s,a) = \mu_h^\star \phi(s,a), \quad \mu_h^\star \in \mathbb{R}^{|S|\times d}, \ \ \phi(s,a) \in \mathbb{R}^d$$

$$r_h(s,a) = (\theta_h^\star)^\top \phi(s,a), \quad \theta_h^\star \in \mathbb{R}^d$$

$V_H^\star(s) = 0, \forall s,$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \ \ \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

$V_H^\star(s) = 0, \forall s,$

$Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \;\; \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

$V_H^\star(s) = 0, \forall s,$

$$Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')$$

$$= \theta_h^\star \cdot \phi(s, a) + \left(\mu_h^\star \phi(s, a)\right)^\top V_{h+1}^\star$$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \;\; \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

$$V_H^\star(s) = 0, \forall s,$$

$$Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')$$

$$= \theta_h^\star \cdot \phi(s, a) + \left(\mu_h^\star \phi(s, a)\right)^\top V_{h+1}^\star$$

$$= \phi(s, a)^\top \left(\theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\star\right)$$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \;\; \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

$$V_H^\star(s) = 0, \forall s,$$

$$Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')$$

$$= \theta_h^\star \cdot \phi(s, a) + \left(\mu_h^\star \phi(s, a)\right)^\top V_{h+1}^\star$$

$$= \phi(s, a)^\top \left(\theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\star\right)$$

$$= \phi(s, a)^\top w_h$$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \ \ \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

$$V_H^\star(s) = 0, \forall s,$$

$$Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')$$

$$= \theta_h^\star \cdot \phi(s, a) + \left(\mu_h^\star \phi(s, a)\right)^\top V_{h+1}^\star$$

$$= \phi(s, a)^\top \left(\theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\star\right)$$

$$= \phi(s, a)^\top w_h$$

$$V_h^\star(s) = \max_a \phi(s, a)^\top w_h, \quad \pi_h^\star(s) = \arg\max_a \phi(s, a)^\top w_h$$

Planning in Linear MDP: Value Iteration

$$P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a), \quad \mu_h^\star \in \mathbb{R}^{|S| \times d}, \;\; \phi(s, a) \in \mathbb{R}^d$$

$$r_h(s, a) = (\theta_h^\star)^\top \phi(s, a), \quad \theta_h^\star \in \mathbb{R}^d$$

$$V_H^\star(s) = 0, \forall s,$$

$$Q_h^\star(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')$$

$$= \theta_h^\star \cdot \phi(s, a) + \left(\mu_h^\star \phi(s, a)\right)^\top V_{h+1}^\star$$

$$= \phi(s, a)^\top \left(\theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\star\right)$$

$$= \phi(s, a)^\top w_h$$

$$V_h^\star(s) = \max_a \phi(s, a)^\top w_h, \quad \pi_h^\star(s) = \arg\max_a \phi(s, a)^\top w_h$$

Indeed we can show that $Q_h^\pi(\cdot, \cdot)$

Is linear with respect to ϕ as well, for any π, h

UCBVI in Linear MDPs

At the beginning of iteration n:

UCBVI in Linear MDPs

At the beginning of iteration n:

1. Learn transition model $\{\hat{P}_h^n\}_{h=0}^{H-1}$ from all previous data $\{s_h^i, a_h^i, s_{h+1}^i\}_{i=0}^{n-1}$

UCBVI in Linear MDPs

At the beginning of iteration n:

1. Learn transition model $\{\hat{P}_h^n\}_{h=0}^{H-1}$ from all previous data $\{s_h^i, a_h^i, s_{h+1}^i\}_{i=0}^{n-1}$

2. Design reward bonus $b_h^n(s, a), \forall s, a$

UCBVI in Linear MDPs

At the beginning of iteration n:

1. Learn transition model $\{\hat{P}_h^n\}_{h=0}^{H-1}$ from all previous data $\{s_h^i, a_h^i, s_{h+1}^i\}_{i=0}^{n-1}$

2. Design reward bonus $b_h^n(s, a), \forall s, a$

3. Plan: $\pi^{n+1} = \text{VI}\left(\{\hat{P}^n\}_h, \{r_h + b_h^n\}\right)$

How to estimate $\{\hat{P}_h^n\}_{h=0}^{H-1}$?

How to estimate $\{\hat{P}_h^n\}_{h=0}^{H-1}$?

Denote $\delta(s) \in \mathbb{R}^{|S|}$ with zero everywhere except the entry corresponding to s

How to estimate $\{\hat{P}_h^n\}_{h=0}^{H-1}$?

Denote $\delta(s) \in \mathbb{R}^{|S|}$ with zero everywhere except the entry corresponding to s

Given s, a, note that $\mathbb{E}_{s' \sim P_h(\cdot|s,a)}\left[\delta(s')\right] = P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a)$

How to estimate $\{\hat{P}^n_h\}_{h=0}^{H-1}$?

Denote $\delta(s) \in \mathbb{R}^{|S|}$ with zero everywhere except the entry corresponding to s

Given s, a, note that $\mathbb{E}_{s' \sim P_h(\cdot|s,a)}\left[\delta(s')\right] = P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a)$

Penalized Linear Regression:

$$\min_{\mu} \sum_{i=1}^{n-1} \|\mu\phi(s_h^i, a_h^i) - \delta(s_{h+1}^i)\|_2^2 + \lambda\|\mu\|_F^2$$

How to estimate $\{\hat{P}_h^n\}_{h=0}^{H-1}$?

Denote $\delta(s) \in \mathbb{R}^{|S|}$ with zero everywhere except the entry corresponding to s

Given s, a, note that $\mathbb{E}_{s' \sim P_h(\cdot|s,a)} \left[\delta(s')\right] = P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a)$

$Y \sim X$

Penalized Linear Regression:

$(X^\top X)^{-1} X^\top Y$

$$\min_{\mu} \sum_{i=1}^{n-1} \|\mu\phi(s_h^i, a_h^i) - \delta(s_{h+1}^i)\|_2^2 + \lambda\|\mu\|_F^2$$

$$A_h^n = \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$$

$$\hat{\mu}_h^n = (A_h^n)^{-1} \sum_{i=1}^{n-1} \delta(s_{h+1}^i)\phi(s_h^i, a_h^i)^\top$$

How to estimate $\{\hat{P}_h^n\}_{h=0}^{H-1}$?

Denote $\delta(s) \in \mathbb{R}^{|S|}$ with zero everywhere except the entry corresponding to s

Given s, a, note that $\mathbb{E}_{s' \sim P_h(\cdot|s,a)}\left[\delta(s')\right] = P_h(\cdot \mid s, a) = \mu_h^\star \phi(s, a)$

Penalized Linear Regression:

$$\min_{\mu} \sum_{i=1}^{n-1} \|\mu\phi(s_h^i, a_h^i) - \delta(s_{h+1}^i)\|_2^2 + \lambda\|\mu\|_F^2$$

$$A_h^n = \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I \qquad \widehat{\mu}_h^n = (A_h^n)^{-1} \sum_{i=1}^{n-1} \delta(s_{h+1}^i)\phi(s_h^i, a_h^i)^\top$$

$$\hat{P}_h^n(\cdot \mid s, a) = \hat{\mu}_h^n \phi(s, a)$$

How to choose $b_h^n(s,a)$?

Chebyshev-like approach, similar to in linUCB (will cover next lecture):

$$b_h^n(s,a) = \beta\sqrt{\phi(s,a)^\top (A_h^n)^{-1}\phi(s,a)}, \quad \beta = \widetilde{O}(dH)$$

linUCB-VI: Put All Together

For $n = 1 \rightarrow N:$

1. Set $A_h^n = \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$

2. Set $\widehat{\mu}_h^n = (A_h^n)^{-1} \sum_{i=1}^{n-1} \delta(s_{h+1}^i)\phi(s_h^i, a_h^i)^\top$

3. Estimate $\hat{P}^n : \hat{P}_h^n(\cdot \mid s, a) = \widehat{\mu}_h^n \phi(s, a)$

4. Plan: $\pi^n = \text{VI}\left(\{\hat{P}_h^n, r_h + b_h^n\}_h\right)$, with $b_h^n(s, a) = cdH\sqrt{\phi(s, a)^\top (A_h^n)^{-1} \phi(s, a)}$

5. Execute $\pi^n : \{s_0^n, a_0^n, r_0^n, \ldots, s_{H-1}^n, a_{H-1}^n, r_{H-1}^n, s_H^n\}$

linUCB-VI: Put All Together

For $n = 1 \rightarrow N$:

1. Set $A_h^n = \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$

2. Set $\widehat{\mu}_h^n = (A_h^n)^{-1} \sum_{i=1}^{n-1} \delta(s_{h+1}^i)\phi(s_h^i, a_h^i)^\top$

3. Estimate $\hat{P}^n$: $\hat{P}_h^n(\cdot \mid s, a) = \widehat{\mu}_h^n \phi(s, a)$

4. Plan: $\pi^n = \text{VI}\left(\{\hat{P}_h^n, r_h + b_h^n\}_h\right)$, with $b_h^n(s, a) = cdH\sqrt{\phi(s, a)^\top (A_h^n)^{-1} \phi(s, a)}$

5. Execute π^n : $\{s_0^n, a_0^n, r_0^n, \ldots, s_{H-1}^n, a_{H-1}^n, r_{H-1}^n, s_H^n\}$

$$\mathbb{E}\left[\text{Regret}_N\right] := \mathbb{E}\left[\sum_{n=1}^{N} \left(V^\star - V^{\pi^n}\right)\right] \leq \widetilde{O}\left(H^2 d^{1.5} \sqrt{N}\right)$$

linUCB-VI: Put All Together

For $n = 1 \to N$:

1. Set $A_h^n = \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$

2. Set $\widehat{\mu}_h^n = (A_h^n)^{-1} \sum_{i=1}^{n-1} \delta(s_{h+1}^i)\phi(s_h^i, a_h^i)^\top$

3. Estimate $\hat{P}^n$: $\hat{P}_h^n(\cdot \mid s, a) = \widehat{\mu}_h^n \phi(s, a)$

4. Plan: $\pi^n = \text{VI}\left(\{\hat{P}_h^n, r_h + b_h^n\}_h\right)$, with $b_h^n(s, a) = cdH\sqrt{\phi(s, a)^\top (A_h^n)^{-1} \phi(s, a)}$

5. Execute π^n : $\{s_0^n, a_0^n, r_0^n, \ldots, s_{H-1}^n, a_{H-1}^n, r_{H-1}^n, s_H^n\}$

$$\mathbb{E}\left[\text{Regret}_N\right] := \mathbb{E}\left[\sum_{n=1}^{N} \left(V^\star - V^{\pi^n}\right)\right] \leq \widetilde{O}\left(H^2 d^{1.5} \sqrt{N}\right)$$

No S, A dependence!

Today

- ✓ Recap
- ✓ UCB-VI for tabular MDPs
- ✓ UCB-VI for linear MDPs
- Contextual bandits intro

Beyond simple bandits

Beyond simple bandits

In a bandit, we are presented with the same decision at every time

Beyond simple bandits

In a bandit, we are presented with the same decision at every time

In practice, often decisions are not the same every time

Beyond simple bandits

In a bandit, we are presented with the same decision at every time

In practice, often decisions are not the same every time

E.g., in online advertising there may not be a single best ad to show all users on all websites:

Beyond simple bandits

In a bandit, we are presented with the same decision at every time

In practice, often decisions are not the same every time

E.g., in online advertising there may not be a single best ad to show all users on all websites:

- maybe some types of users prefer one ad while others prefer another, or

Beyond simple bandits

In a bandit, we are presented with the same decision at every time

In practice, often decisions are not the same every time

E.g., in online advertising there may not be a single best ad to show all users on all websites:

- maybe some types of users prefer one ad while others prefer another, or
- maybe one type of ad works better on certain websites while another works better on other websites

Beyond simple bandits

In a bandit, we are presented with the same decision at every time

In practice, often decisions are not the same every time

E.g., in online advertising there may not be a single best ad to show all users on all websites:

- maybe some types of users prefer one ad while others prefer another, or
- maybe one type of ad works better on certain websites while another works better on other websites

Which user comes in next is random, but we have some context to tell situations apart and hence learn different optimal actions

Contextual bandit environment

Contextual bandit environment

Context at time t encoded into a variable x_t that we see before choosing our action

Contextual bandit environment

Context at time t encoded into a variable x_t that we see before choosing our action

x_t is drawn i.i.d. at each time point from a distribution ν_x on sample space $\mathcal{X}$

Contextual bandit environment

Context at time t encoded into a variable x_t that we see before choosing our action

x_t is drawn i.i.d. at each time point from a distribution ν_x on sample space $\mathcal{X}$

x_t then affects the reward distributions of each arm, i.e., if we choose arm k, we get a reward that is drawn from a distribution that depends on x_t, namely, $\nu^{(k)}(x_t)$

Contextual bandit environment

Context at time t encoded into a variable x_t that we see before choosing our action

x_t is drawn i.i.d. at each time point from a distribution ν_x on sample space $\mathcal{X}$

x_t then affects the reward distributions of each arm, i.e., if we choose arm k, we get a reward that is drawn from a distribution that depends on x_t, namely, $\nu^{(k)}(x_t)$

Accordingly, we should also choose our action a_t in a way that depends on x_t, i.e., our action should be chosen by a function of x_t (a policy), namely, $\pi_t(x_t)$

Contextual bandit environment

Context at time t encoded into a variable x_t that we see before choosing our action

x_t is drawn i.i.d. at each time point from a distribution ν_x on sample space $\mathscr{X}$

x_t then affects the reward distributions of each arm, i.e., if we choose arm k, we get a reward that is drawn from a distribution that depends on x_t, namely, $\nu^{(k)}(x_t)$

Accordingly, we should also choose our action a_t in a way that depends on x_t, i.e., our action should be chosen by a function of x_t (a policy), namely, $\pi_t(x_t)$

If we knew everything about the environment, we'd want to use the optimal policy

$$\pi^\star(x_t) := \arg\max_{k \in \{1,\ldots,K\}} \mu^{(k)}(x_t), \qquad \text{where } \mu^{(k)}(x) := \mathbb{E}_{r \sim \nu^{(k)}(x)}[r]$$

Contextual bandit environment

Context at time t encoded into a variable x_t that we see before choosing our action

x_t is drawn i.i.d. at each time point from a distribution ν_x on sample space $\mathcal{X}$

x_t then affects the reward distributions of each arm, i.e., if we choose arm k, we get a reward that is drawn from a distribution that depends on x_t, namely, $\nu^{(k)}(x_t)$

Accordingly, we should also choose our action a_t in a way that depends on x_t, i.e., our action should be chosen by a function of x_t (a policy), namely, $\pi_t(x_t)$

If we knew everything about the environment, we'd want to use the optimal policy

$$\pi^\star(x_t) := \arg\max_{k\in\{1,\ldots,K\}} \mu^{(k)}(x_t), \qquad \text{where } \mu^{(k)}(x) := \mathbb{E}_{r\sim\nu^{(k)}(x)}[r]$$

$\pi^\star$ is the policy we compare to in computing regret

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \to T - 1$

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \to T - 1$

1. Learner sees context $x_t \sim \nu_x$

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \rightarrow T - 1$

1. Learner sees context $x_t \sim \nu_x$ Independent of any previous data

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \rightarrow T - 1$

1. Learner sees context $x_t \sim \nu_x$ Independent of any previous data
2. Learner pulls arm $a_t = \pi_t(x_t) \in \{1, \ldots, K\}$

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \rightarrow T - 1$

1. Learner sees context $x_t \sim \nu_x$ — Independent of any previous data
2. Learner pulls arm $a_t = \pi_t(x_t) \in \{1, \ldots, K\}$ — π_t policy learned from all data seen so far

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \to T - 1$

1. Learner sees context $x_t \sim \nu_x$ — Independent of any previous data
2. Learner pulls arm $a_t = \pi_t(x_t) \in \{1,\ldots,K\}$ — π_t policy learned from all data seen so far
3. Learner observes reward $r_t \sim \nu^{(a_t)}(x_t)$ from arm a_t in context x_t

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \rightarrow T - 1$

1. Learner sees context $x_t \sim \nu_x$ — Independent of any previous data
2. Learner pulls arm $a_t = \pi_t(x_t) \in \{1, \ldots, K\}$ — π_t policy learned from all data seen so far
3. Learner observes reward $r_t \sim \nu^{(a_t)}(x_t)$ from arm a_t in context x_t

Note that if the context distribution ν_x always returns the same value (e.g., 0), then the contextual bandit reduces to the original multi-armed bandit

Contextual bandit environment (cont'd)

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \rightarrow T - 1$

1. Learner sees context $x_t \sim \nu_x$ — Independent of any previous data
2. Learner pulls arm $a_t = \pi_t(x_t) \in \{1, \ldots, K\}$ — π_t policy learned from all data seen so far
3. Learner observes reward $r_t \sim \nu^{(a_t)}(x_t)$ from arm a_t in context x_t

Note that if the context distribution ν_x always returns the same value (e.g., 0), then the contextual bandit reduces to the original multi-armed bandit

π_t might seem unfamiliar since we haven't talked about a policy in bandits before, but actually we've always had it, it's just that without context, we didn't need a name or notation for it because it was so simple!

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Now what about contextual versions?

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Now what about contextual versions?

Thompson sampling with contexts is conceptually identical!

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Now what about contextual versions?

Thompson sampling with contexts is conceptually identical!

Still start from a prior on $\{\nu^{(k)}(x)\}_{k\in\{1,\ldots,K\},x\in\mathcal{X}}$,

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Now what about contextual versions?

Thompson sampling with contexts is conceptually identical!

Still start from a prior on $\{\nu^{(k)}(x)\}_{k\in\{1,\ldots,K\},x\in\mathcal{X}}$,

but now this is $K|\mathcal{X}|$ (usually $\gg K$) distributions, so need more complicated prior

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \text{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Now what about contextual versions?

Thompson sampling with contexts is conceptually identical!

Still start from a prior on $\{\nu^{(k)}(x)\}_{k\in\{1,\ldots,K\},x\in\mathcal{X}}$,

but now this is $K|\mathcal{X}|$ (usually $\gg K$) distributions, so need more complicated prior

Still can update distribution on $\{\nu^{(k)}(x)\}_{k\in\{1,\ldots,K\},x\in\mathcal{X}}$ after each reward $r_t \sim \nu^{(a_t)}(x_t)$

Contextual bandit algorithms

What was π_t for UCB? (π_t has no argument because there was no context)

$$\pi_t = \arg\max_k \mathsf{UCB}_t^{(k)}$$

For Thompson sampling?

π_t was a *randomized* policy that sampled from the posterior distribution of $k^\star$

Now what about contextual versions?

Thompson sampling with contexts is conceptually identical!

Still start from a prior on $\{\nu^{(k)}(x)\}_{k\in\{1,\ldots,K\},x\in\mathcal{X}}$,

but now this is $K|\mathcal{X}|$ (usually $\gg K$) distributions, so need more complicated prior

Still can update distribution on $\{\nu^{(k)}(x)\}_{k\in\{1,\ldots,K\},x\in\mathcal{X}}$ after each reward $r_t \sim \nu^{(a_t)}(x_t)$

Still know posterior over $k^\star(x_t)$ that can draw from to choose a_t; this is $\pi_t(x_t)$

UCB for contextual bandits

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

Solution: share information across contexts x_t, i.e., don't treat $\nu^{(k)}(x)$ and $\nu^{(k)}(x')$ as completely different distributions which have nothing to do with one another

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

Solution: share information across contexts x_t, i.e., don't treat $\nu^{(k)}(x)$ and $\nu^{(k)}(x')$ as completely different distributions which have nothing to do with one another

Example: showing an ad on a NYT article on politics vs a NYT article on sports:

UCB for contextual bandits

UCB algorithm also conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

Solution: share information across contexts x_t, i.e., don't treat $\nu^{(k)}(x)$ and $\nu^{(k)}(x')$ as completely different distributions which have nothing to do with one another

Example: showing an ad on a NYT article on politics vs a NYT article on sports: Not *identical* readership, but still both on NYT, so probably still *similar* readership!

Modeling in contextual bandits

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4$ $\Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4 \Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

With linear model there are just 2 parameters: the two entries of $\theta_k \in \mathbb{R}^2$

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4$ $\Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

With linear model there are just 2 parameters: the two entries of $\theta_k \in \mathbb{R}^2$

Lower dimension makes learning easier, but model could be wrong/biased

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4 \Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

With linear model there are just 2 parameters: the two entries of $\theta_k \in \mathbb{R}^2$

Lower dimension makes learning easier, but model could be wrong/biased

Choosing the best model, fitting it, and quantifying uncertainty are really questions of supervised learning

Today

- ✓ Recap
- ✓ UCB-VI for tabular MDPs
- ✓ UCB-VI for linear MDPs
- ✓ Contextual bandits intro

Summary:

UCBVI algorithm applies UCB idea to MDPs to achieve exploration/exploitation trade-off

Attendance:
bit.ly/3RcTC9T

lin UCBVI

Feedback:
bit.ly/3RHtlxy