Network Friendly Recommendations Project part II

Kalamarakis Theodoros: 2018030022
Toganidis Nikos: 2018030085

September 11, 2023

Environment Overview

- **States**: There are K states in total, where state i represents the user watching video i
- **Action** The actions are the recommendation batch of N videos.
- **Cost and Rewards** If a video is cached, its cost is 0. If it is not cached, the cost is 1. $\rightarrow Reward_i = 1 - 2 \cdot Cost_i$
- **Parameters Selection**
  - $\gamma = 1 - q$
  - $\epsilon = \frac{1}{t^{1/3}}(\#num\ of\ states \cdot \log t)^{1/3}$ where t is the number of episodes
  - $\alpha = 0.01$

SlateQ vs. Tabular Q-Learning

- In tabular Q-Learning:

$$
\begin{aligned}
Q^t(s, A) = {} & Q^{t-1}(s, A) \\
& + \alpha \left(R(s, A, s') + \max_{A'} \left(\gamma Q^{t-1}(s', A') \right) - Q^{t-1}(s, A) \right)
\end{aligned}
$$

SlateQ vs. Tabular Q-Learning

- In tabular Q-Learning:

$$\begin{aligned}Q^t(s,A) &= Q^{t-1}(s,A) \\ &\quad + \alpha\left(R(s,A,s') + \max_{A'}\left(\gamma Q^{t-1}(s',A')\right) - Q^{t-1}(s,A)\right)\end{aligned}$$

- Dimension of $Q(s,A)$: $K \times \binom{K}{N}$. Time-consuming for large K and N.

SlateQ vs. Tabular Q-Learning

- In tabular Q-Learning:

$$Q^t(s, A) = Q^{t-1}(s, A) \\ + \alpha \left(R(s, A, s') + \max_{A'} \left(\gamma Q^{t-1}(s', A') \right) - Q^{t-1}(s, A) \right)$$

- Dimension of $Q(s, A)$: $K \times \binom{K}{N}$. Time-consuming for large K and N.
- SlateQ introduces $\bar{Q}(s, i)$ which quantifies the value of being in state s and choosing item i.

SlateQ vs. Tabular Q-Learning

- In tabular Q-Learning:

$$Q^t(s,A) = Q^{t-1}(s,A) + \alpha\left(R(s,A,s') + \max_{A'}\left(\gamma Q^{t-1}(s',A')\right) - Q^{t-1}(s,A)\right)$$

- Dimension of $Q(s,A)$: $K \times \binom{K}{N}$. Time-consuming for large K and N.
- SlateQ introduces $\bar{Q}(s,i)$ which quantifies the value of being in state s and choosing item i.
- Definition with Bellman equation

$$\bar{Q}(s,i) = R(s,i) + \gamma \sum_{s'} P(s'|s,i)V(s')$$

SlateQ vs. Tabular Q-Learning

- Update using:

$$\bar{Q}^{t+1}(s,i) = \bar{Q}^{t}(s,i) + \alpha\left(R(s,i) + \max_{A'}\left(\gamma Q^{t}(i,A')\right) - \bar{Q}^{t}(s,i)\right)$$

SlateQ vs. Tabular Q-Learning

- Update using:

$$\bar{Q}^{t+1}(s,i) = \bar{Q}^{t}(s,i) + \alpha\left(R(s,i) + \max_{A'}\left(\gamma Q^{t}(i,A')\right) - \bar{Q}^{t}(s,i)\right)$$

- It can be proven that:

$$Q(s,A) = \sum_{i \in A} P(i|s,A)\bar{Q}(s,i)$$

SlateQ vs. Tabular Q-Learning

- Update using:

$$\bar{Q}^{t+1}(s,i) = \bar{Q}^{t}(s,i) + \alpha\left(R(s,i) + \max_{A'}\left(\gamma Q^{t}(i,A')\right) - \bar{Q}^{t}(s,i)\right)$$

- It can be proven that:

$$Q(s,A) = \sum_{i \in A} P(i|s,A)\bar{Q}(s,i)$$

- Thus , the update rule becomes:

$$\bar{Q}^{t+1}(s,i) = \bar{Q}^{t}(s,i) + \alpha\left(R(s,i) + \max_{A'}\left(\gamma \sum_{j \in A'} P^{t}(j|i,A')\bar{Q}^{t}(i,j)\right) - \bar{Q}^{t}(s,i)\right)$$

Maximization Problem in SlateQ

- Maximization problem:

$$\max_{A} \sum_{i \in A} P(i|s, A)\bar{Q}(s, i) = \max_{A} \sum_{i \in A} \frac{1}{N}\bar{Q}(s, i)$$

Maximization Problem in SlateQ

- Maximization problem:

$$\max_A \sum_{i \in A} P(i|s, A)\bar{Q}(s, i) = \max_A \sum_{i \in A} \frac{1}{N}\bar{Q}(s, i)$$

- define the vector $\mathbf{x} \rightarrow$ if $i \in A$: $x_i = 1$ else: $x_i = 0$

Maximization Problem in SlateQ

- Maximization problem:

$$\max_{A} \sum_{i \in A} P(i|s, A)\bar{Q}(s, i) = \max_{A} \sum_{i \in A} \frac{1}{N}\bar{Q}(s, i)$$

- define the vector $\mathbf{x} \rightarrow$ if $i \in A$: $x_i = 1$ else: $x_i = 0$
- We can rewrite maximization problem as:

$$\max_{\mathbf{x}} \sum_{i} x_i \frac{1}{N}\bar{Q}(s, i)$$

Maximization Problem in SlateQ

- $$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \sum_i x_i \frac{1}{N} \bar{Q}(s,i) \\ \text{subject to} \quad & x_i \in \{0,1\} \\ & \sum_i x_i = N, \end{aligned}$$
- To transform this into a linear optimization problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \sum_i x_i \frac{1}{N} \bar{Q}(s,i) \\ \text{subject to} \quad & x_i \in [0,1] \\ & \sum_i x_i = N, \end{aligned}$$

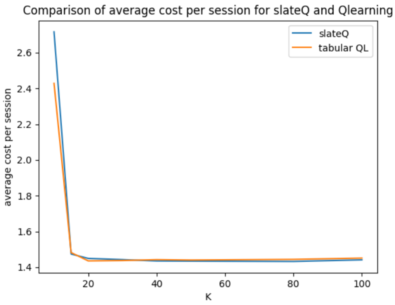
Evaluation Description and Optimal Average Cost

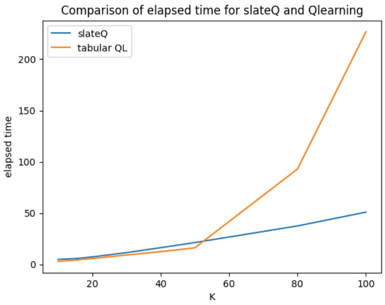
- Evaluate algorithm's effectiveness using a simulation function.
- Calculate average cost per session using the derived policy.
- For fixed number of cached items $C = 0.2K$, the optimal average cost, $E(S)$, is:

$$E[S] = 0.8 + 0.8(\frac{1}{q} - 1)(1 - \alpha)$$

- With $\alpha = 0.8$ and $q = 0.2$, we get $E[S] = 1.44$.
- Optimal policy should yield this average cost.

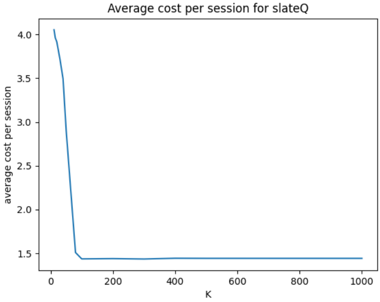
Simulation Results and Analysis

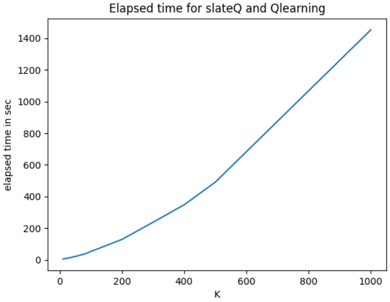




- Average cost for SlateQ aligns with Q-Learning and converges to 1.44.
- SlateQ time increases linearly with K, Q-Learning escalates exponentially.

Simulation Results and Analysis





- With a large library, the algorithm identifies optimal policy, cost remains 1.44.
- Elapsed time exhibits a near-linear increase. Algorithm operates as expected.