Contextual Combinatorial Cascading Bandit Experiment

1 Preliminary

1.1 Synthetic dataset

Let $\mathcal{S} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ be the unit ball of $\mathbb{R}^d$. Let $E = \{1, ..., L\}$ be the set of all base arms. We randomly choose θ_* with $\|\theta_*\|_2 = 1$ and randomly assign a $\mu_i \in \mathcal{S}$ to i for any $i \in E$. At each time t, we choose $b_{t,i} \in \mathcal{S}$ randomly for any base arm i. Also we fix a constant h to balance weights of μ_i and disturbance $b_{t,i}$. Let $x_{t,i} = \frac{\mu_i + hb_{t,i}}{\|\mu_i + h \cdot b_{t,i}\|_2}$ be the context of base arm i at time t. And the weight for base arm i at time t is $w_t(i) = \theta_*^\top x_{t,i} + \epsilon_{t,i}$ where $\epsilon_{t,i} \sim N(\mu_i, \sigma_i)$ for fixed σ_i.

1.2 MovieLens

Let L be the number of all movies and let M be the number of all users. The MovieLens dataset is a big matrix $A \in \mathbb{R}^{M \times L}$ where $A(i,j) \in \{0, 1\}$ denotes whether user i has watched movie j or not. We split A to be $H + F$ by putting entry-1 of A to H and F with probability $\sim \mathrm{Ber}(p)$ for some fixed p. We can regard H as know information about history 'What users have watched' and regard F as future criterion. We use H to derive feature vectors of both users and movies by SVD decomposition $H = USV^\top$ where $U = (u_1; ...; u_M)$ and $V = (v_1; ...; v_L)$. At every time t, use $x_{t,i} = u_i v_j^\top$ as the context information of base arm i and randomly choose a user I_t. And use $w_t(j) = F(I_t, j)$ as the weight of base arm j.

Notice that for this case, fixed number of base arms, it might have problem if we use (u_{I_t}, v_j) as context information. Since to find the best arm, it is equivalent to find the best one with highest weights sum, so is equivalent to the best one with highest $\theta_v^\top x$.

The measurement for MovieLens is accuracy because we don't know the true θ_*.

1.3 Routing

Let $G = (V, E = \{e_1, ..., e_L\})$ be the topology representation of an ISP network. G is symmetric by it's definition. Considering the scinario where a package is sent from its source node to its destination, it's returned back to the source after trying to bypass an edge with high latency. And the routing is failed if so, and the source will receive the routing history till the failing edge. The agent is then motivated to assign an routing path, which can be recognized as an simple path in G from the source node to the destination node, so as to avoid edges with high latency. Assume we have some sort of tell about the dynamics of the network conditions between the hosts, each being encoded in a d-dimentional vector, the network routing problem is to find the routing path least likely to involve an edge with high latency. Denote $x_{i,t}$ be the vector associated with edge e_i, at time t, assume the corresponding latency is drawn from an exponential distribution with mean $1 - \theta_*^\top x_{t,i}$ independently, and define the latency is high iff it's greater than a constant tolerance value τ. We formulate the network routing problem as an contextual combinatorial cascading problem.

Let $E = \{e_1, ..., e_L\}$ be the set of arms, and $x_{i,t}$ be the context associated with arm e_i at time t. In order to send a package from u_t to v_t, the agent have to choose an superarm from

$$S = \{A = (e_{k_1}, ..., e_{k_n}) : e_{k_j} \in E, e_{k_1}, ..., e_{k_n} \text{is a simple path of } G \text{ from } u_t \text{ to } v_t\},$$

Algorithm	**Cumulative Reward** $\sum_{t=1}^{T} r_i$
Li	4342.73
Qin	4323.26
Monkey	1765.41
Kveton	1787.81
Perfect Play	N/A

Table 1: **Cumulative reward w.r.t different baselines, under Movielens setting.**

with the expected payoffs (opt out discount)

$$E[r_A|\theta] = \Pi_{1\leq i\leq n(A)}(1 - \exp(-\tau/(1-\theta^\top x_{k(A)_i,t}))),$$

where $n(A)$ and $k(A)_i$ are the amount of arms in superarm A, and the index of the i-th arm, separately. The agent finds A_t by running shortest path algorithm on G with weight $\hat{\theta}^\top x_{k_i,t}$ assigned to e_i, which yields

$$A_t = \arg\min_{A\in S} \sum_{1\leq i\leq n(A)} \hat{\theta}^\top x_{k(A)_i,t}.$$

For $A \in S$, denote $\hat{\mu}(A)_i = 1 - \hat{\theta}^\top x_{k(A)_i,t}$, we have then

$$\begin{aligned}
E[r_{A_t}|\hat{\theta}]/E[r_A|\hat{\theta}] &= \frac{\Pi_{1\leq i\leq n(A_t)}(1-e^{-\tau/\hat{\mu}(A_t)_i})}{\Pi_{1\leq i\leq n(A)}(1-e^{-\tau/\hat{\mu}(A)_i})} \\
&\geq \min_{0\leq\sigma\leq n(A),\sum_i \hat{\mu}(A_t)_i=\sum_i \hat{\mu}(A)_i=\sigma} \frac{\Pi_{1\leq i\leq n(A_t)}(1-e^{-\tau/\hat{\mu}(A_t)_i})}{\Pi_{1\leq i\leq n(A)}(1-e^{-\tau/\hat{\mu}(A)_i})} \\
&\geq \min_{0\leq\sigma\leq n(A)} \frac{(1-e^{-\tau})^\sigma}{(1-e^{-\tau n(A)/\sigma})^{n(A)}} \\
&\geq \min_{0\leq\sigma\leq n(A)} \frac{1-e^{-\tau/\sigma}}{(1-e^{-\tau n(A)/\sigma})^{n(A)}} \\
&\geq 1-e^{-\tau/n(A)}/(1-e^{-\tau})^{n(A)} \\
&\geq (1-e^{-\tau/|V|})/(1-e^{-\tau})^{|V|}.
\end{aligned}$$

Let $\alpha(\tau, G) = (1-e^{-\tau/|V|})/(1-e^{-\tau})^{|V|}$, the above inequality shows that we can realize the $\alpha(\tau, G)$-approximation oracle using shortest path algorithm. After the agent chooses the shortest path as the superarm, the reward and the first ever edge with high latency, if any, is feedbacked.

2 Results

Example results, with $T = 10000$ and $T = 30000$ seperately, are listed on Table 1 and Table 2.

Algorithm	**Cumulative Reward** $\sum_{t=1}^{T} r_i$
Li	7676.54
Qin	7165.70
Monkey	5428.52
Kveton	5468.97
Perfect Play	N/A

Table 2: **Cumulative reward w.r.t different baselines, under ISP routing setting.**