Cooperative Multi-Agent Bandits with Heavy Tails

Abhimanyu Dubey and Alex Pentland

Media Lab and Institute for Data Systems and Society (IDSS)
Massachusetts Institute of Technology

dubeya@mit.edu

ICML 2020

Multi-Armed Bandits

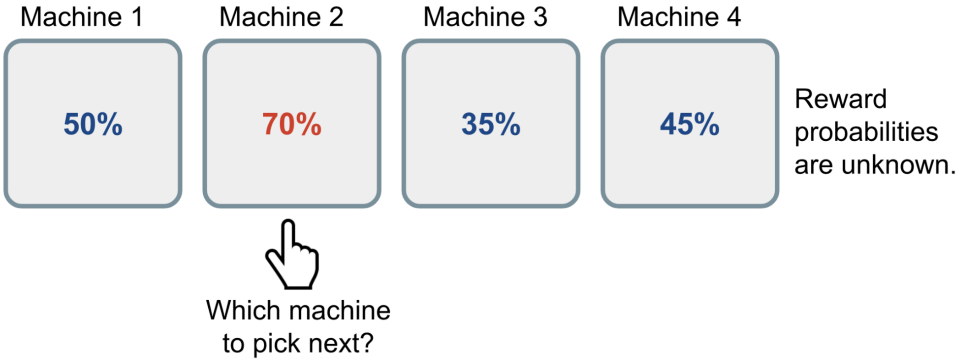

Figure: Multi-armed bandit (courtesy `lilianweng.github.io`).

Cooperative Bandits

- Distributed learning is an increasingly popular paradigm in ML: multiple parties collaborate to train a stronger joint model by sharing data.
- An alternative is to let data remain in a distributed setup, and have one ML algorithm (agent) for each data center, i.e., federated learning.
- Each agent can communicate with other agents to (securely) share relevant information, e.g., over a network.
- The group of all agents therefore collectively *cooperate* to solve their own learning problems.

Summary of Contributions

- In many application areas, observations are heavy tailed, e.g., in internet traffic analysis and supply chain networks.
- Current cooperative bandit algorithms operate largely by *distributed consensus*, that averages opinions held by agents.
- Consensus protocols are inherently not robust to heavy-tailed reward distributions, and have inefficient communication complexity.
- **Summary**: In this paper, we propose algorithms for the heavy-tailed cooperative bandit that uses an alternative decentralized communication protocol, resulting in efficient and robust multi-agent bandit learning.

Stochastic Multi-Armed Bandits

- K actions ("arms") that return rewards r_k sampled i.i.d. from K different distributions, each with mean μ_k.
- The problem proceeds in rounds; at each round t, the agent chooses action a_t, and obtains a randomly drawn reward $r(t)$, such that $\mathbb{E}[r(t)] = \mu_{a_t}$.
- The goal is to minimize *regret* (for $\mu^* = \arg\max_{k \in [K]} \mu_k$),

$$R(T) = \underbrace{T \cdot \mu^*}_{\substack{\text{best possible} \\ \text{avg. reward}}} - \underbrace{\sum_{k \in [K]} \mu_k \mathbb{E}[n_k(T)]}_{\substack{\text{obtained reward} \\ \text{(in expectation)}}} = \underbrace{\sum_{k \in [K]} (\mu^* - \mu_k) \mathbb{E}[n_k(T)]}_{\substack{\text{expected "loss" from} \\ \text{picking suboptimal arms}}}$$

Cooperative Multi-Armed Bandits

- M agents are each faced with the same K-armed bandit problem.
- Agents are connected by a (connected, undirected) graph $\mathcal{G}$.
- The agents must cooperate to collectively minimize the *group regret*:

$$R_{\mathcal{G}}(T) = \sum_{m \in \mathcal{G}} R_m(T)$$

The Upper Confidence Bound (UCB) Algorithm

- "Optimism in the face of uncertainty" strategy – i.e. to be optimistic about an arm when we are uncertain of its utility.
- For each arm, we compute

$$Q_k(t) = \underbrace{\frac{\sum_{i=1}^{n_k(t-1)} r_k^i}{n_k(t-1)}}_{\text{empirical mean}} + \underbrace{\sqrt{\frac{2\ln(t-1)}{n_k(t-1)}}}_{\text{UCB}(t)}.$$

- Choose arm with largest $Q_k(t)$.

Heavy-Tailed Distributions

- A random variable X is *light-tailed* if it admits a finite moment generating function, i.e. there exists $u_0 > 0$ such that $\forall |u| \leq u_0$,

$$M_X(u) \triangleq \mathbb{E}[\exp(uX)] < \infty.$$

  Otherwise X is heavy-tailed.
- When rewards are sub-Gaussian, the empirical mean and variance are the obvious estimators for the first 2 moments.
  - They are asymptotically optimal estimators (rate of concentration).
  - They can be computed in $O(1)$ time for streaming settings.
- In case of heavy-tailed rewards we require robust estimators to obtain optimal regret.

Robust Estimators and the Running Consensus

- Distributed consensus works by slowly "averaging" opinions between neighboring agents. This subsequent averaging causes information to diffuse throughout the network.
- Robust mean estimators, however, are fundamentally incompatible with naive averaging, and cannot be updated in $O(1)$ time.
  - Trimmed mean and Catoni's estimators require $O(T)$ consensus algorithms.
  - Median-of-means estimator requires $O(\log T)$ consensus algorithms.

Message Passing Protocol

- Instead of a consensus, each agent communicates its actions and rewards in the form of a tuple (a_t, r_t, d), where $d \leq \gamma$ is the life of the message (i.e., it is dropped after it has been forward γ times).
- For any time t, each agent
  - Gathers all messages $M(t)$ from its neighbors and discards stale messages.
  - Chooses an arm following any algorithm and obtains a reward.
  - Adds the action-reward tuple (a_t, r_t, γ) to $M(t)$.
  - Sends each message in $M(t)$ to all its neighbors.
- Since we are working with individual rewards, all robust estimators can be applied to this protocol.

Robust Message-Passing UCB

For any time t, each agent m

- Gathers all messages $M(t)$ from its neighbors and discards all messages with $d = 0$.
- Filters all unseen messages by arm k and adds new rewards to corresponding sets $\mathcal{S}_m^k(t)$.
- Computes the mean $\hat{\mu}_k^m(t)$ for each arm k from $\mathcal{S}_m^k(t)$ using any robust mean estimator.
- Chooses arm that maximizes $\hat{\mu}_k^m(t) + \text{UCB}_k^m(t)$, and obtains reward r_t.
- Adds the action-reward tuple (a_t, r_t, γ) to $M(t)$.
- Sends each message in $M(t)$ to all its neighbors.

Lower Bounds

Lower Bound for Cooperative Setting

Under suitable assumptions, for any $\Delta \in (0, 1/4)$ and $\varepsilon \in (0, 1]$, there exist $K \geq 2$ heavy-tailed distributions such that any consistent algorithm obtains regret of order $\Omega\left(K\Delta^{-1/\varepsilon} \ln T\right)$ when run on a connected graph $\mathcal{G}$.

- This is a generalization of the lower bound for multiple arm pulls to account for delayed feedback over connected graphs.
- Existing optimality rates are in comparison to a single agent pulling MT arms sequentially, which we demonstrate to be inaccurate with upper bounds that match the above lower bound.

Regret Guarantees

Regret of Robust MP-UCB

Robust MP-UCB obtains regret $O\left(\alpha\left(\mathcal{G}_\gamma\right)\left(\sum_{k=1}^{K}(2\Delta_k)^{-1/\varepsilon}\right)\log T\right)$.

- Robust MP-UCB is near-optimal in its dependence on T, K and Δ_k.
- The communication overhead is the independence number $\alpha\left(\mathcal{G}_\gamma\right)$.
  - $\mathcal{G}_\gamma$ has edge (i,j) if there exists a path of length $\leq \gamma$ between i and j in $\mathcal{G}$.
  - If $\gamma = \text{diam}(\mathcal{G})$, $\alpha\left(\mathcal{G}_\gamma\right) = 1$, matching the lower bound (up to constants).
  - If $\gamma = 0$, $\alpha\left(\mathcal{G}_\gamma\right) = M$, i.e. no communication, each agent acts in isolation.
  - $\alpha\left(\mathcal{G}_\gamma\right)$ is monotonically decreasing in γ, and hence γ can be used to strike a compromise between communication and performance.

Additional Optimizations

In certain settings, we can improve the performance of the algorithm further:

- **Cheap Communication**: When messages can be $O(M)$, we can achieve optimal regret $O\left(\left(\sum_{k=1}^{K}(2\Delta_k)^{-1/\varepsilon}\right)\log T\right)$ regardless of γ or $\mathcal{G}$.
- **Streaming Trimmed Mean**: We propose an efficient algorithm to calculate a streaming trimmed mean in $O(\log T)$ time, instead of $O(T)$.
- **Costly Communication**: With sub-Gaussian arms, our algorithm obtains $O\left(\left(\sum_{k=1}^{K}(\Delta_k)^{-1}\right)\log^{3/2} T\right)$ regret with $O(\log T)$ communication.

Applications and Future Work

Applications of multi-agent cooperative multi-armed bandits include

- Inferring preferences where users are connected via a social network.
- Policy estimation in multi-agent systems – robotics, distributed sensors, etc.

Future work includes:

- Generalization to non-identical bandit problems.
- Time-varying network analysis.
- Private communication protocols.

Thank You!
Paper ID: 282