Data Poisoning Attacks on Stochastic Bandits

Fang Liu

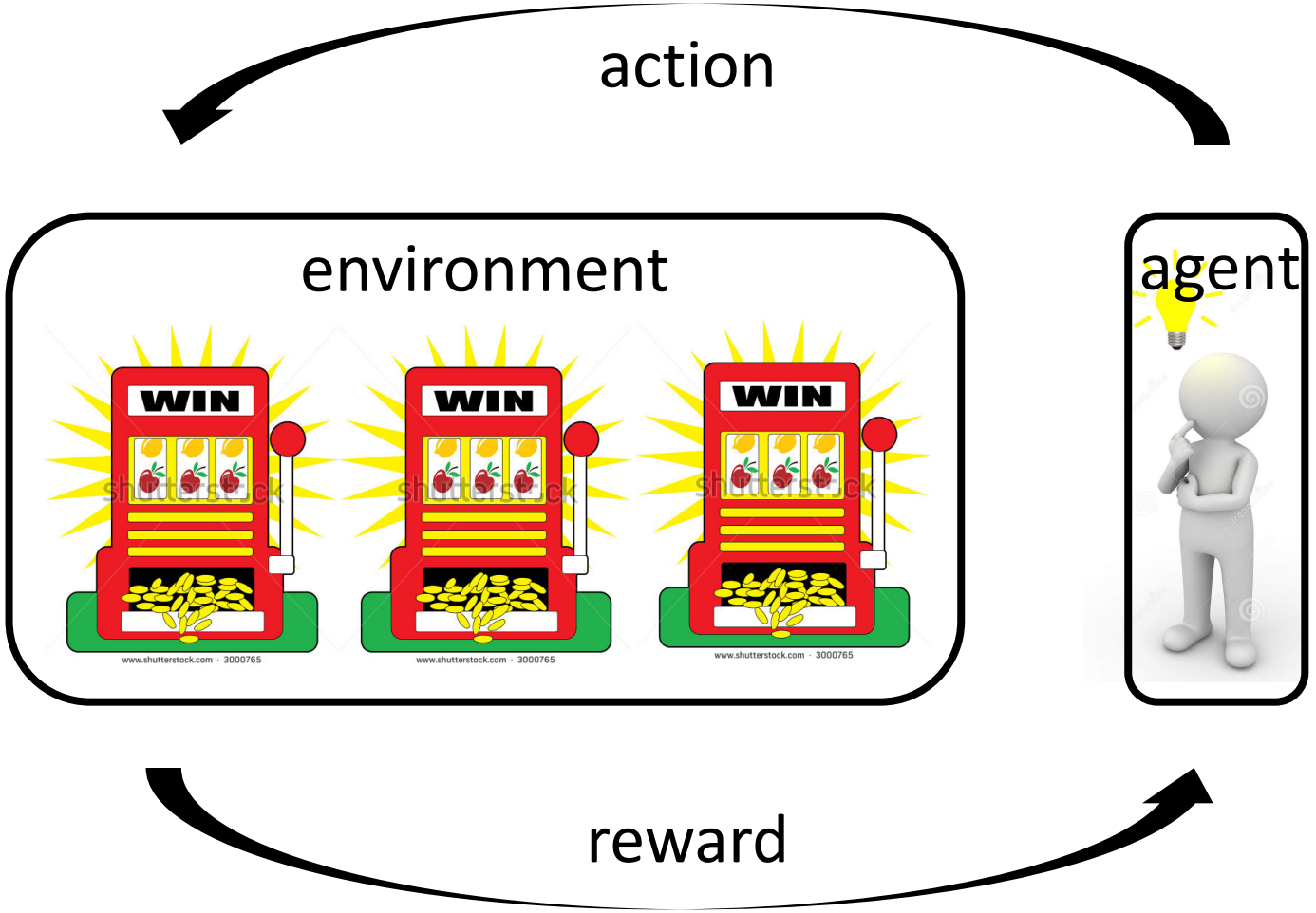
THE OHIO STATE UNIVERSITY

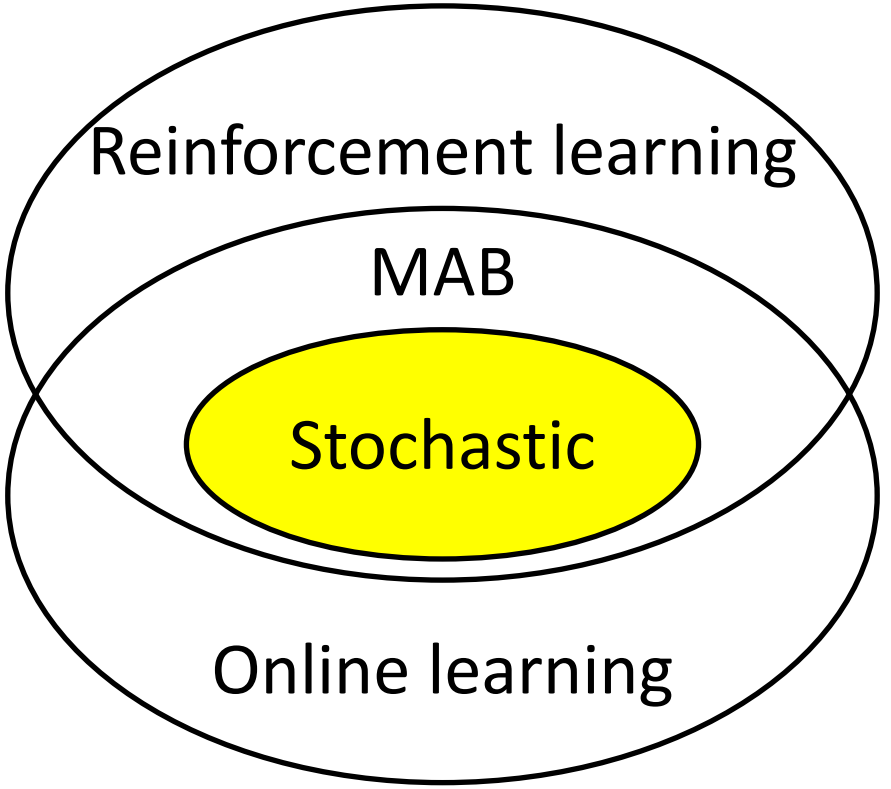
Outline

- Background
  - ❑What are bandits?
  - ❑Why study bandits?
  - ❑Who cares?
  - ❑FAST Signals?
- Data poisoning attacks
  - ❑Motivations
  - ❑Offline model
  - ❑Online model
  - ❑Simulation results
- Conclusion

What are bandits?

- Repeated game between an agent and an environment





What are bandits?

Time	1	2	3	4	5	6	7	8	9	10
Left	$1	$0			$1					
Right			$1	$0						

Which arm would you play next?

What are bandits?

- Model
  - At each (discrete) time t, the agent plays action A_t from a set of K actions
  - The agent receives reward $Y_{A_t,t}$, drawn from unknown distribution A_t
- Performance measure
  - Regret(loss) $$R(T) = \mathbb{E}\left[\max_{i\in[K]} \sum_{t=1}^{T} Y_{i,t} - \sum_{t=1}^{T} Y_{A_t,t}\right]$$
  - Minimize regret = maximize total reward
- Regret lower bounds
  - Problem-dependent: $$\Omega\left(\sum_i \frac{\mu^* - \mu_i}{KL(\mu_a, \mu^*)} \log T\right)$$ where μ_i is expected reward
  - Problem-independent: $$\Omega\left(\sqrt{KT}\right)$$
- Popular algorithms
  - Upper Confidence Bounds (UCB), Thompson Sampling, epsilon-greedy

Why study bandits?

- Many applications
  - Clinical trials
  - Recommendation systems
  - Ad placement
  - A/B test
  - A component of game-playing algorithms (MCTS), e.g. AlphaGo
  - Resource allocation
- A way of isolating one important part of reinforcement learning

  Exploration vs Exploitation
- Rich and beautiful in math ☺

Who cares?

- DeepMind
  - Csaba Szepesvari (co-inventor of UCT)
- Google Research
  - Lihong Li (co-inventor of LinUCB)
- Microsoft Research
  - Sebastien Bubeck
- Adobe Research
  - Branislav Kveton/Zheng Wen

Inspired MCTS for DRL

Yahoo! Front Page

Online Convex Opt.

Ad placement

Facebook?

FAST Signals?

Maybe not

FAST Signals?

Wait

FAST Signals?

Free Lunch Wednesday Problem

- Go FLW weekly
- New to restaurants nearby
- Voting is not fair
- Deploy an algorithm to decide
- Quality of food and service is noisy (factors, e.g., menu, waiters, chef, ingredients)

- T = 52 x (years)
- For each week t, alg. chooses a restaurant $A_t \in \{1, ..., K\}$
- Team feedback aggregates a reward r_t
- r_t is i.i.d. sampled from an unknown distribution A_t

Bandits can maximize the total “happiness” over time under uncertainty.

Data poisoning attacks

- Algorithms intro later
- A new topic in the bandit community
  - The results are quite fundamental and cute in some sense
  - TL,DR: The bandits are vulnerable under certain attack models.

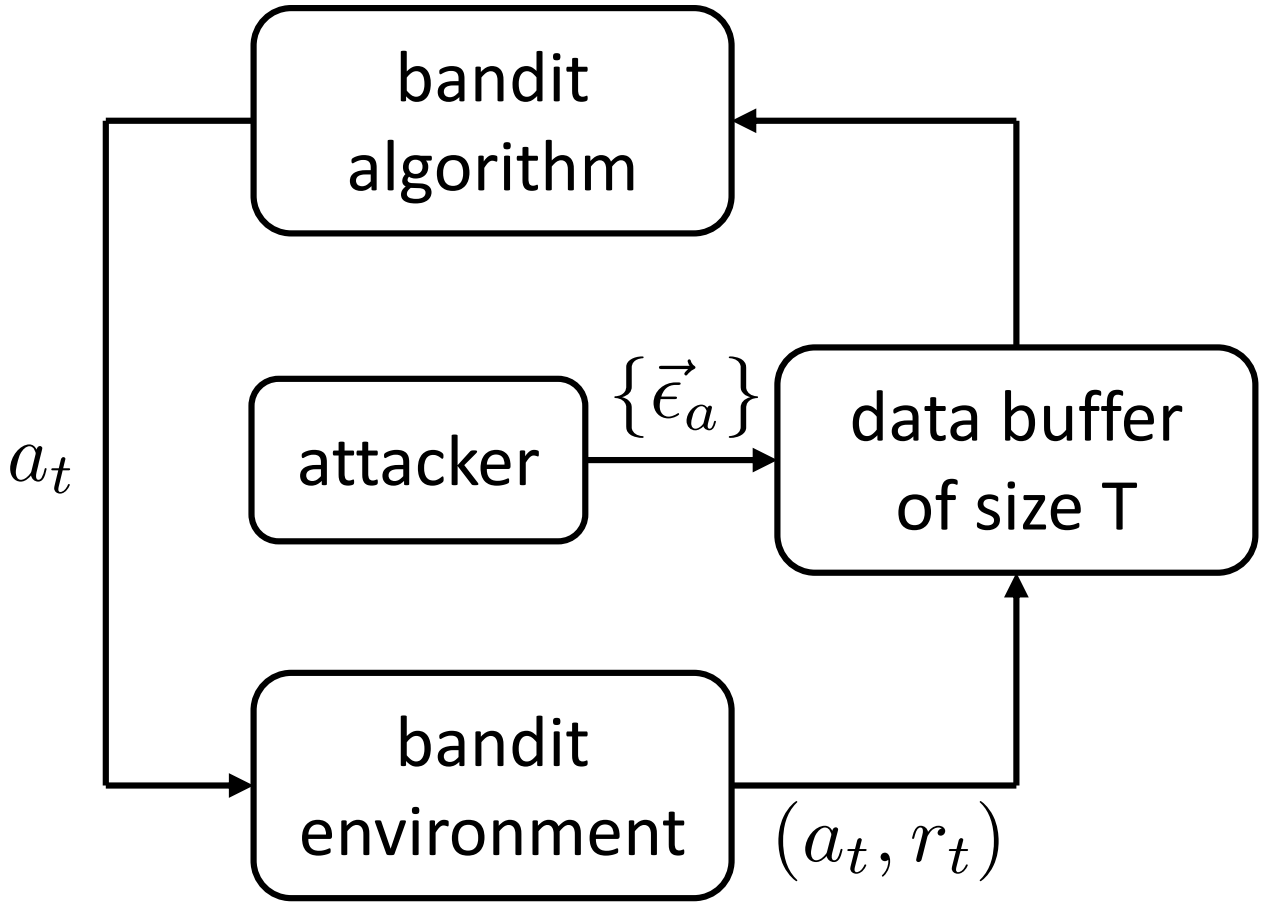
Motivations

- Adversarial learning well studied in deep learning
- How robust are RL? Open
  - If under stealthy attack, hard to detect (nature of exploration-exploitation)
  - Bandit is a good start point
  - Bandit is deployed in some real-world systems

Offline model

- Distributed system
- Algorithm updates in batches
  - Yahoo! Front Page (daily)
- Attacker manipulates the rewards r_t by adding ϵ_t
- Target arm a^*, sub-optimal
- Goal: bandit plays a^* with high prob. $1-\delta$ at T+1
- Cost:

$$C(T)^2 = \sum_{t=1}^{T} \epsilon_t^2 = \sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2.$$



Offline model: epsilon greedy algorithm

$$a_t = \begin{cases} \text{draw uniformly over } \mathcal{A}, & \text{w.p. } \alpha_t \\ \arg\max_{a \in \mathcal{A}} \tilde{\mu}_a(t-1), & \text{otherwise} \end{cases}.$$

- Optimal para: $\alpha_t = \Theta(1/t)$
- Post-attack empirical mean: $\tilde{\mu}_a(t)$
- Attack error tolerance: $\delta = \frac{K-1}{K}\alpha_{T+1}$
- Quadratic program with linear constraints

$$P_1 : \min_{\vec{\epsilon}_a : a \in \mathcal{A}} \quad \sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2$$

$$s.t. \quad \tilde{\mu}_{a^*}(T) \geq \tilde{\mu}_a(T) + \xi, \quad \forall a \neq a^*$$

Offline model: UCB algorithm

$$a_t = \arg\max_{a \in \mathcal{A}} u_a(t) := \tilde{\mu}_a(t-1) + 3\sigma\sqrt{\frac{\log t}{N_a(t-1)}}.$$

- Attack error tolerance: $\delta = 0$
- Conditional "deterministic" algorithm
- Quadratic program with linear constraints

$$P_2 : \min_{\vec{\epsilon}_a : a \in \mathcal{A}} \quad \sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2$$

$$s.t. \quad u_{a^*}(T+1) \geq u_a(T+1) + \xi, \quad \forall a \neq a^*$$

Offline model: Thompson Sampling

$$a_t = \arg\max_{a \in \mathcal{A}} \theta_a(t) \sim \mathcal{N}(\tilde{\mu}_a(t-1)/\sigma^2, \sigma^2/N_a(t-1))$$

- Bayesian algorithm: prior-posterior, prob. matching
- Quadratic program with convex constraints

$$P_3 : \min_{\vec{\epsilon}_a : a \in \mathcal{A}} \quad \sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2$$

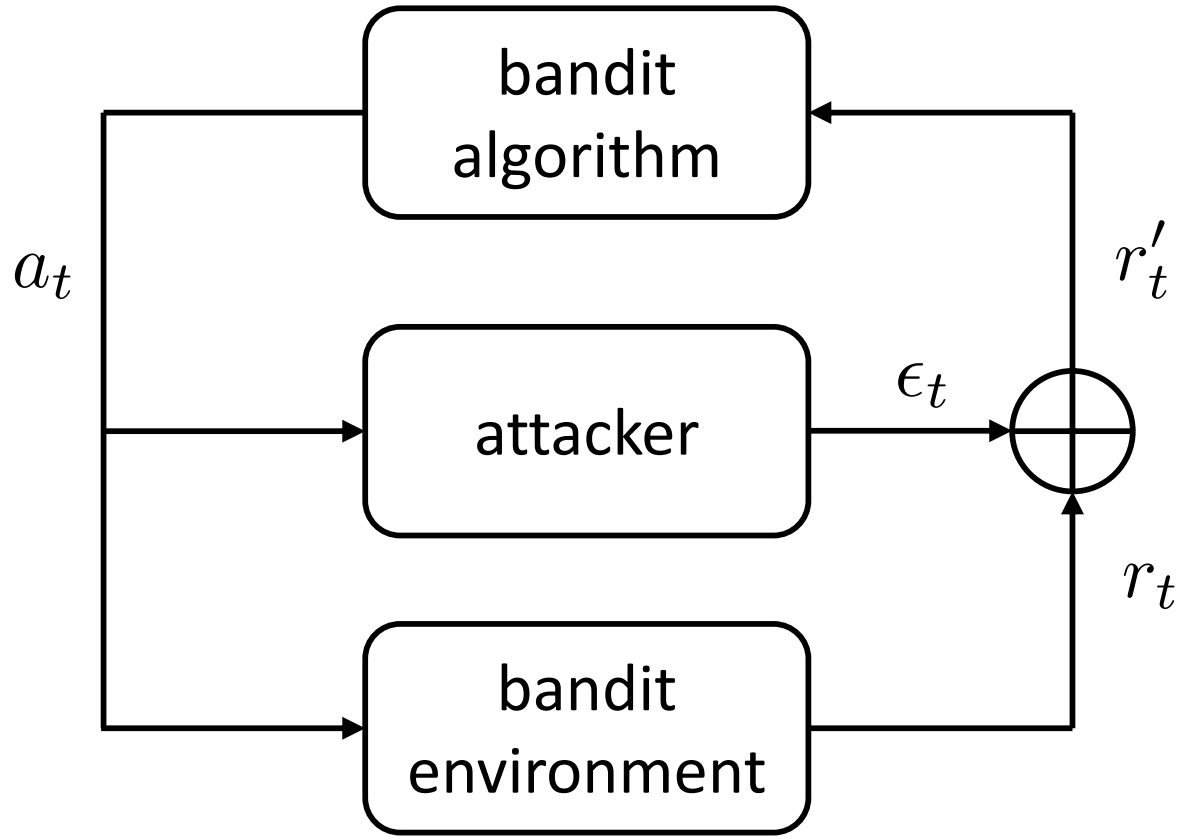
$$s.t. \quad \sum_{a \neq a^*} \Phi\left(\frac{\tilde{\mu}_a(T) - \tilde{\mu}_{a^*}(T)}{\sigma^3 \sqrt{1/m_a + 1/m_{a^*}}} \right) \leq \delta$$

$$\tilde{\mu}_a(T) - \tilde{\mu}_{a^*}(T) \leq 0, \quad \forall a \neq a^*$$

Online model

- Algorithm updates online
- Attacker manipulates the rewards r_t by adding ϵ_t
- Target arm a^*, sub-optimal
- Goal: bandit plays a^* in $\Theta(T)$ with high prob. $1-\delta$
- Cost:

$$C(T) = \sum_{t=1}^{T} |\epsilon_t|$$



Online model: Oracle attacks

$$\epsilon_t = -\mathrm{I}\{a_t \neq a^*\}[\mu_{a_t} - \mu_{a^*} + \xi]^+$$

- Attack against any bandit algorithm
- Not practical: unknown expectations

Proposition 1. Assume that the bandit algorithm achieves an $O(\log T)$ regret bound. Then the oracle attack with $\xi > 0$ succeeds, i.e., $\mathbb{E}[N_{a^*}(T)] = T - o(T)$. Furthermore, the expected attack cost is $O(\sum_{i \neq a^*} [\mu_i - \mu_{a^*} + \xi]^+ \log T)$.

Online model: Oracle constant attacks

$$\epsilon_t = -\mathrm{I}\{a_t \neq a^*\} C_{a_t}$$

- Attack against any bandit algorithm
- Sufficient and necessary conditions
- Not efficient: wild guess

Proposition 2. Assume that the bandit algorithm achieves an $O(\log T)$ regret bound. Then the constant attack with $\{C_a\}_{a \neq a^*}$ succeeds if and only if $C_a > [\mu_a - \mu_{a^*}]^+$, $\forall a \neq a^*$. If the attack succeeds, then the expected attack cost is $O(\sum_{a \neq a^*} C_a \log T)$.

Adaptive attacks by constant Estimation (ACE)

$$\epsilon_t = -\mathrm{I}\{a_t \neq a^*\}[\hat{\mu}_{a_t}(t) - \hat{\mu}_{a^*}(t) + \beta(N_{a_t}(t)) + \beta(N_{a^*}(t))]^+$$

- where $\beta(n) = \sqrt{\frac{2\sigma^2}{n} \log \frac{\pi^2 K n^2}{3\delta}}$ is decreasing in n
- Pre-attack empirical mean: $\hat{\mu}_a(t)$
- Attack against any bandit algorithm
- Adaptive and efficient: estimation
- How: concentration inequality + union bound

Lemma 1. For $\delta \in (0,1)$, $\mathbb{P}(E) > 1 - \delta$, where

$$E = \{\forall a \in \mathcal{A}, \forall t : |\hat{\mu}_a(t) - \mu_a| < \beta(N_a(t))\}.$$

Online model: ACE attacks

$$\epsilon_t = -\mathrm{I}\{a_t \neq a^*\}[\hat{\mu}_{a_t}(t) - \hat{\mu}_{a^*}(t) + \beta(N_{a_t}(t)) + \beta(N_{a^*}(t))]^+$$

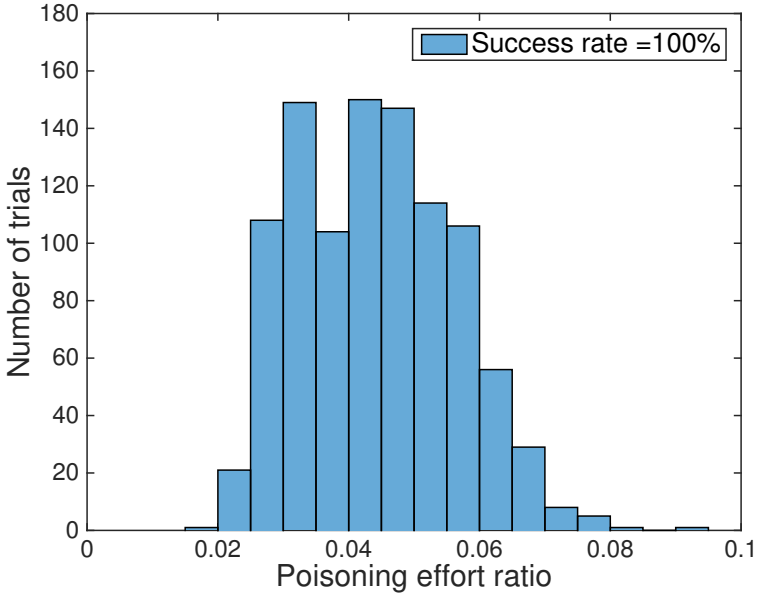
- Tight to oracle attack (with some additive constant)

Theorem 1. Given any $\delta \in (0, 0.5)$, assume that the bandit algorithm achieves an $O(\log T)$ regret bound with probability at least $1 - \delta$. With probability at least $1 - 2\delta$, the ACE attacker forces the bandit algorithm to play the target arm a^* in $N_{a^*}(T)$ times, such that $N_{a^*}(T) = T - o(T)$, using the accumulated attack cost

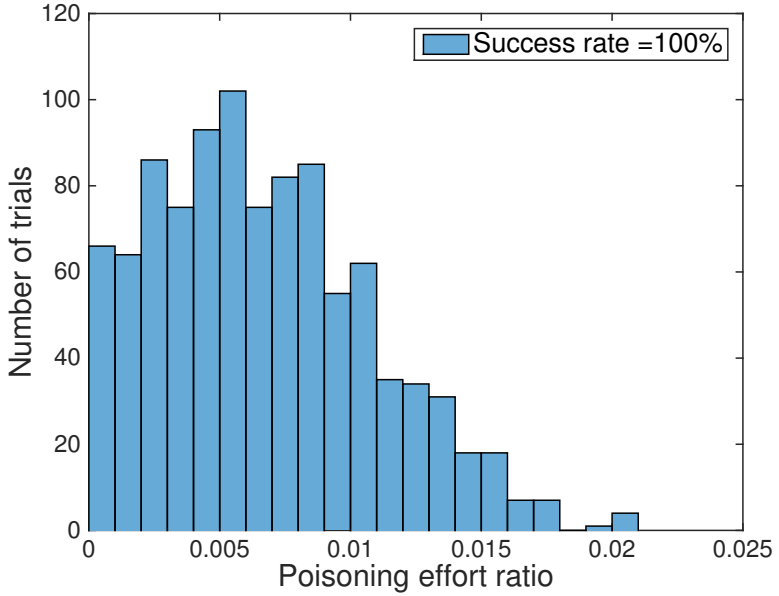
$$\sum_{t=1}^{T} |\epsilon_t| \leq O\left(\sum_{a \neq a^*} \left([\mu_a - \mu_{a^*}]^+ + 4\beta(1)\right) \log T\right).$$

Simulation results: offline model

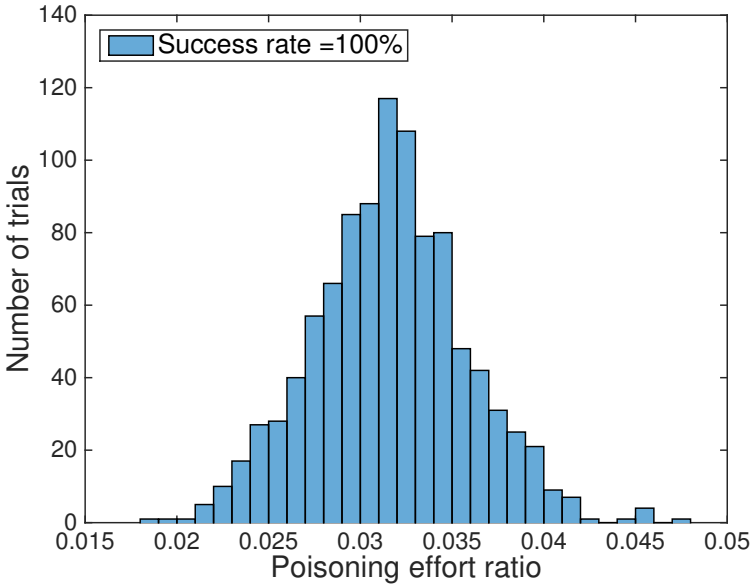
- Gaussian distributions with random drawn expectations
- Parameters: $K = 5, \sigma = 0.1, T = 1000, \delta = 0.05$
- Poisoning effort ratio: $$\frac{||\vec{\epsilon}||_2}{||\vec{r}||_2} = \sqrt{\frac{\sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2}{\sum_{a \in \mathcal{A}} ||\vec{r}_a||_2^2}}$$



Epsilon-greedy



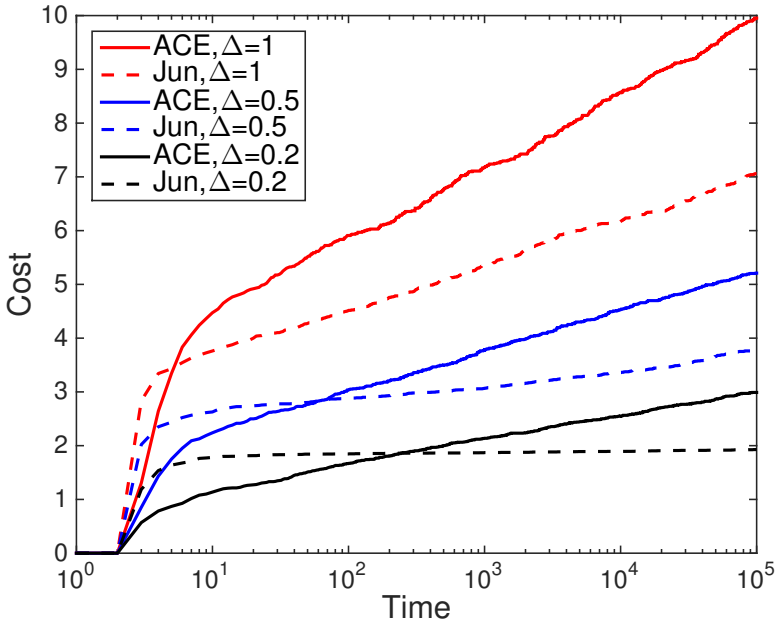
UCB



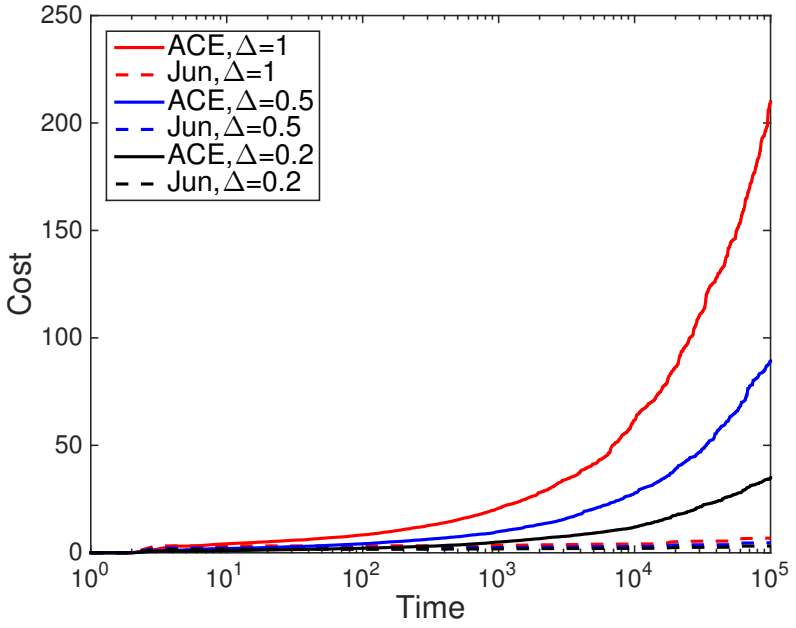
Thompson Sampling

Simulation results: online model

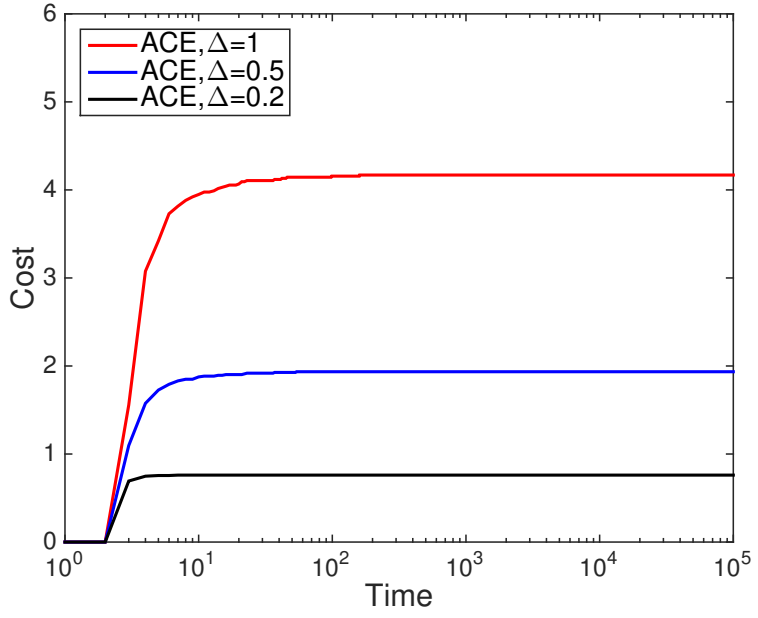
- Gaussian distributions with random drawn expectations
- Parameters: $K = 2, \sigma = 0.1, T = 10^5, \delta = 0.05$
- 3 cases: $\mu_1 = \Delta, \mu_2 = 0$
- Jun's attack is optimized if the bandit algo. is known (esp. deterministic).



Epsilon-greedy



UCB



Thompson Sampling

Recall

Free Lunch Wednesday Problem

- Suppose target rest. 1
- If A_t is not 1, attack by degrading the quality of food or service “slightly”
- How: order weird food, give extremely negative feedback, etc.
- Outcome: go to rest. 1 also every time

- T = 52 x (years)
- For each week t, alg. chooses a restaurant $A_t \in \{1, ..., K\}$
- Team feedback aggregates a reward r_t
- r_t is i.i.d. sampled from an unknown distribution A_t

Conclusions

- Negative results: bandits are vulnerable!
  - Algorithm-specific attacks on 3 popular bandits in offline model
  - Adaptive attacks on any bandit in online model
- Any hope to build a robust world?
- Crack the model
  - Encrypt decision
  - Replicate reward records
- Detect by distribution outlier detection