THE OHIO STATE UNIVERSITY

An Online Learning Approach to Networking Problems

Fang Liu[1]

Joint work with Yin Sun[2], Sinong Wang[1], Zizhan Zheng[3], Joohyun Lee[4], Swapna Buccapatnam[5], Atilla Eryilmaz[1] and Ness Shroff[1]

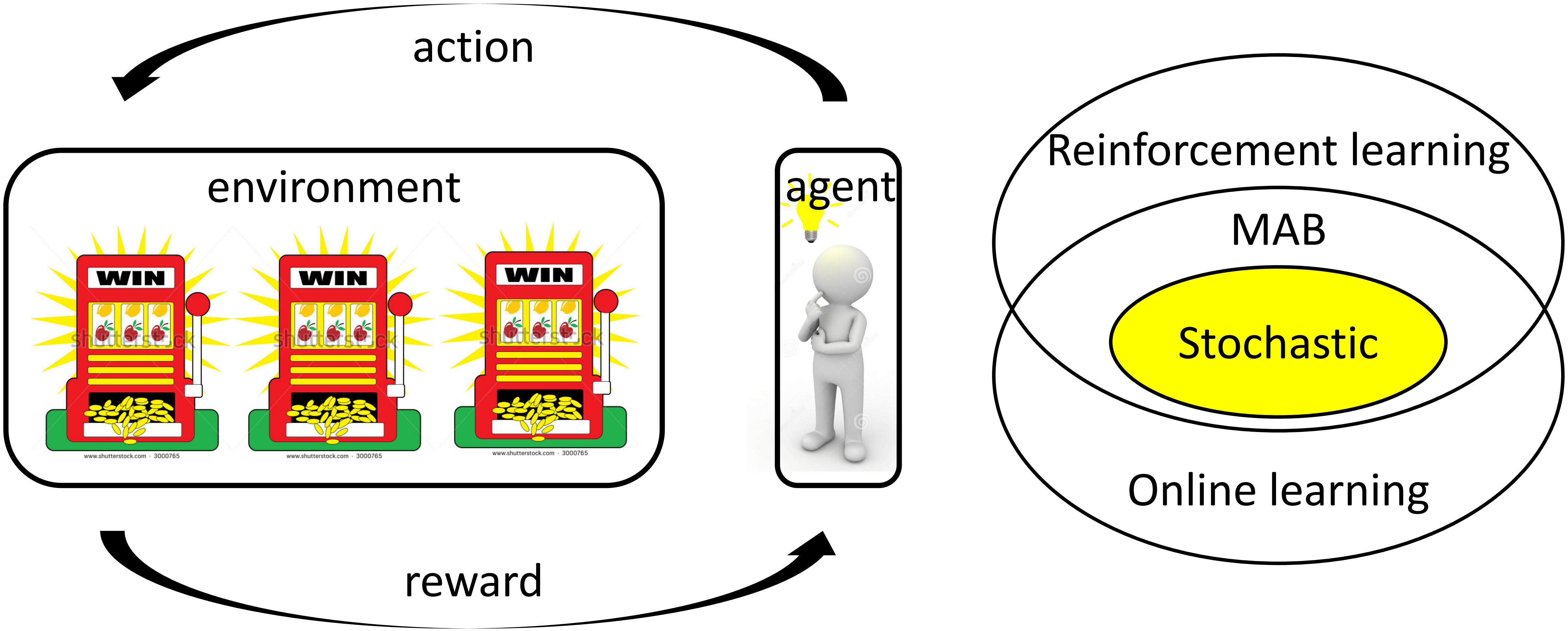
[1]The Ohio State University, [2]Auburn University, [3]Tulane University, [4]Hanyang University, [5]AT&T Labs Research

Outline

- Multi-Armed Bandits Framework
  - ❑Stochastic Bandits At a Glance
- Motivations/Applications to Networking Problems
  - ❑Stochastic Routing Problem
  - ❑Real-time Control Problem
  - ❑Edge Computing
  - ❑Task Scheduling Problem
- Variants of Bandits
  - ❑Graphical Bandits
  - ❑Boosting Bandits
  - ❑Non-stationary Bandits
  - ❑Parameterized Clustering Bandits
- Conclusion

Multi-Armed Bandits Framework

- Repeated game between an agent and an environment

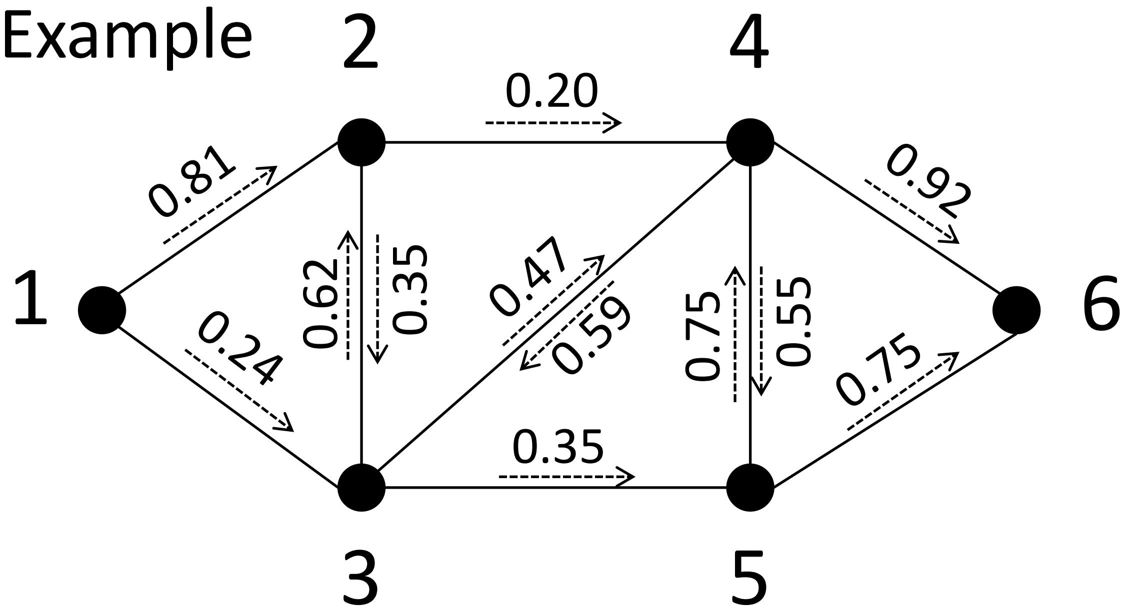


Stochastic Bandits At a Glance

- 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

Applications to Networking Problems

- Stochastic Routing Problem
  - Action => routing path
  - Observation => random delay (link delay or end-to-end delay)
  - Reward => minus delay (or 1/delay, etc)
  - Statistics of delay is uknown
- Example



Playing one action (partially) observes the outcome if playing others

Reduce dependence on K

Graphical Bandits

Applications to Networking Problems

- Real-time Control Problem
  - Action => control
  - Reward => train the learner in reinforcement learning way
  - Real-time
- Example
  - Network function virtualization
    - Want no delay due to control at each node
  - Security monitors with tracking ability
    - Want no tracking failure due to slow decision
  - Physical layer channel selection
    - Want to select within coherence time

Time-sensitive applications require the algorithm to respond quickly

Complexity vs Optimality

Boosting Bandits

Applications to Networking Problems

- Edge Computing
  - Make decisions on devices in the fog
  - Learning user pattern
- Example
  - Smartphone application management
    - Want to close background applications
    - Save energy without painful cold start
  - Update for perishable mobile content
    - Want to pull the latest content
    - Keep data fresh without draining energy
  - IoT services
    - Want to suggest services actively
    - Understand the master

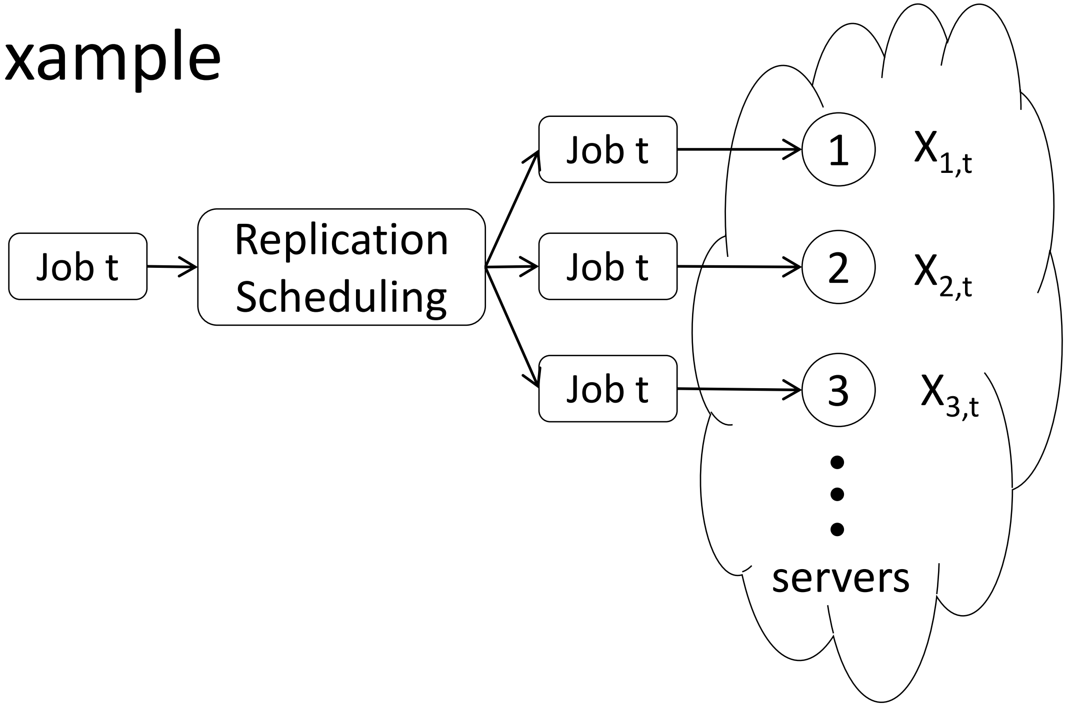
User preference or pattern may change over time

Adaptive to changing env.

Non-stationary Bandits

Applications to Networking Problems

- Task Scheduling Problem
  - Make replications to be robust to straggling servers
  - Action => replication number
  - Reward => (minus) minimum service time
  - Servers with unknown service time distribution
- Example



The outcome of playing one action implies some information about others

Handle correlations

Parameterized Clustering Bandits

Graphical Bandits

- What is graphical bandits?
  - A graph G over the actions, possibly known (or unknown) to the agent
  - An arc (i,j) means playing action i also observes one outcome of action j
- Graph theory review
  - Clique cover number $\chi(G)$
  - Independence number $\beta_0(G)$
  - Domination number $\gamma(G)$
- Recap of stochastic bandits
  - Curse of dimensionality $O(K \log T)$ or $O(\sqrt{KT})$
- Why graphical bandits?
  - Reduce dependence on K to graph numbers

Graphical Bandits

- Literature review
  - Proposed in adversarial bandits by Shie Mannor et. al. [MS2011] $\beta_0(G)$
  - UCB-N, introduced to stochastic bandits by S. Caron et. al. [CKLB2012] $\chi(G)$
  - UCB-LP, epsilon-greedy-LP, improved by Swapna et. al. [BES2014] $\gamma(G)$
  - Generalized to bi-partite graph by Swapna et. al. [1] $\gamma(G)$
  - Without graph information, studied by Cohen et. al. [CHK2016] $\beta_0(G)$
  - TS-N, evaluated by Tossou et. al. [TDD2017] $\chi(G)$
  - IDS-N, proposed by Liu et. al. [2] $\chi(G)$
  - TS-N, improved analysis for TS-N and IDS-N by Liu et. al. [3] $\beta_0(G)$

[1] Swapna Buccapatnam, Fang Liu, Atilla Eryilmaz and Ness Shroff, "Reward maximization under uncertainty: Leveraging side-observations on networks", accepted by JMLR.

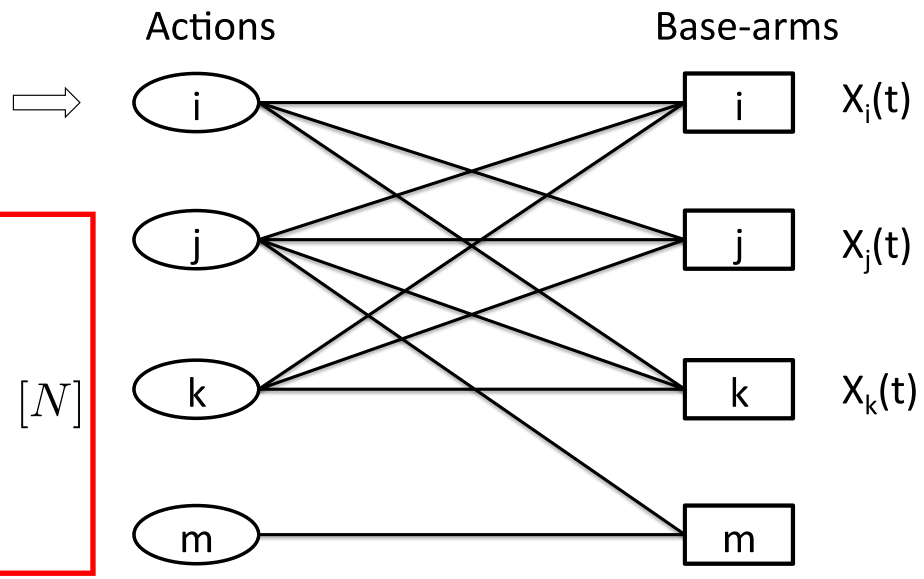
[2] Fang Liu, Swapna Buccapatnam and Ness Shroff, "Information directed sampling for stochastic bandits with graph feedback", AAAI 2018.

[3] Fang Liu, Zizhan Zheng and Ness Shroff, "Analysis of Thompson Sampling for Graphical Bandits Without the Graphs", UAI 2018.

Graphical Bandits

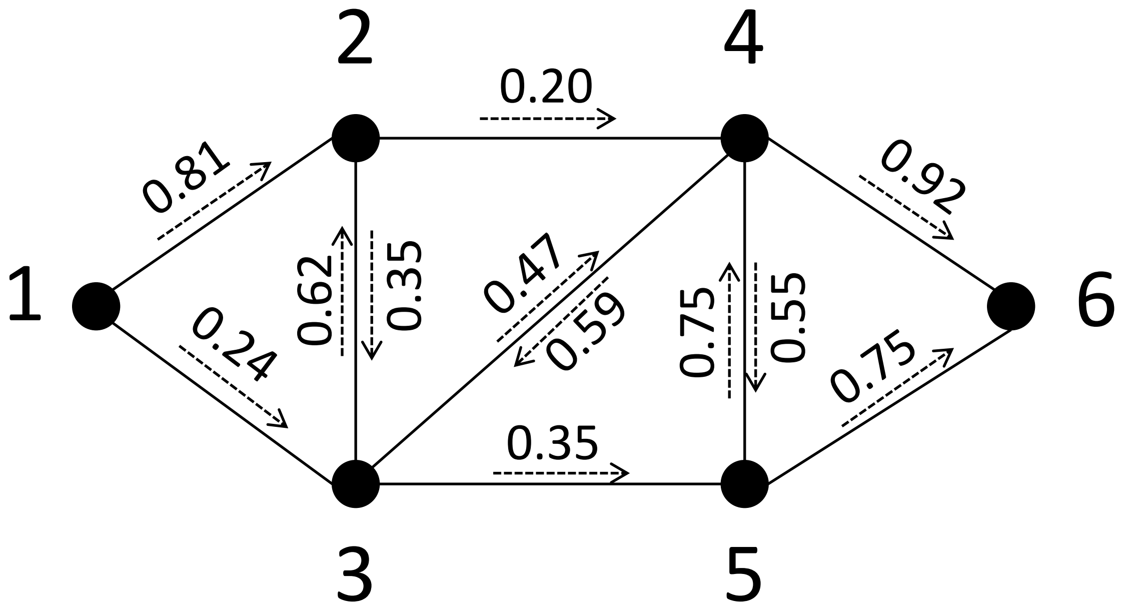
- Time-invariant bipartite graph setting
  - Known graph structure (otherwise, play each action once)
  - Action base-arm bipartite graph: model stochastic routing problem
  - Action => routing path
  - Base-arm => link
- UCB-LP/epsilon-greedy-LP algorithms
  - Dominating set (hitting set)
  - Explore on dominating set
  - LP relaxation of dominating set
  - Regret $O(\gamma(G)\log T)$

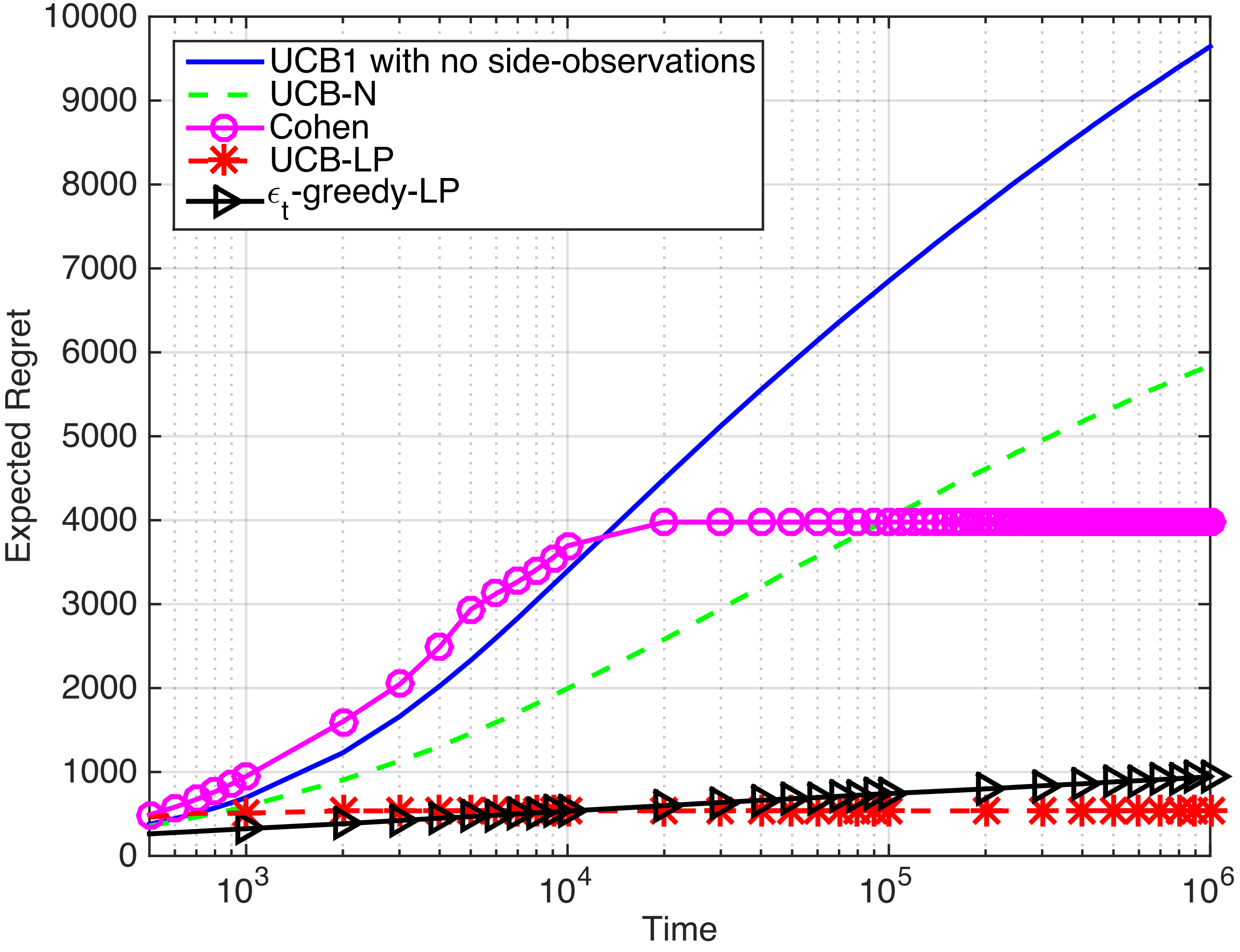
$$\begin{aligned} \min \quad & \sum_{j\in[K]} z_j \\ s.t. \quad & \sum_{j\in S_i} z_j \geq 1, \forall i \in [N] \\ & z_j \geq 0, \forall j \in [K] \end{aligned}$$



Graphical Bandits

- Numerical Results
  - Stochastic routing example
  - Reduce at least 75% regret of the state of the art



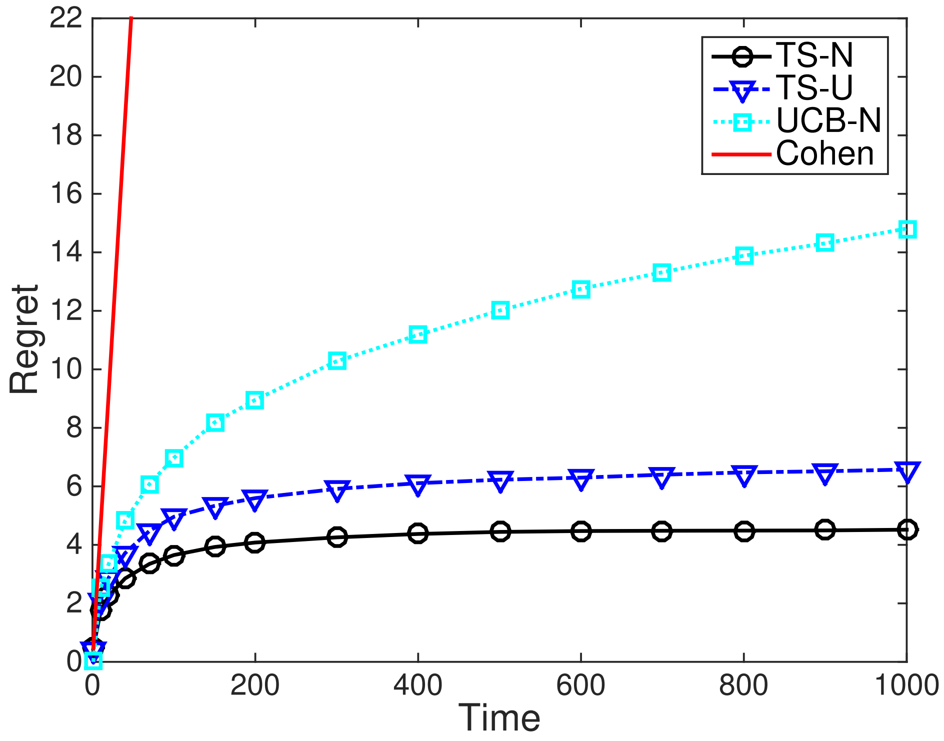


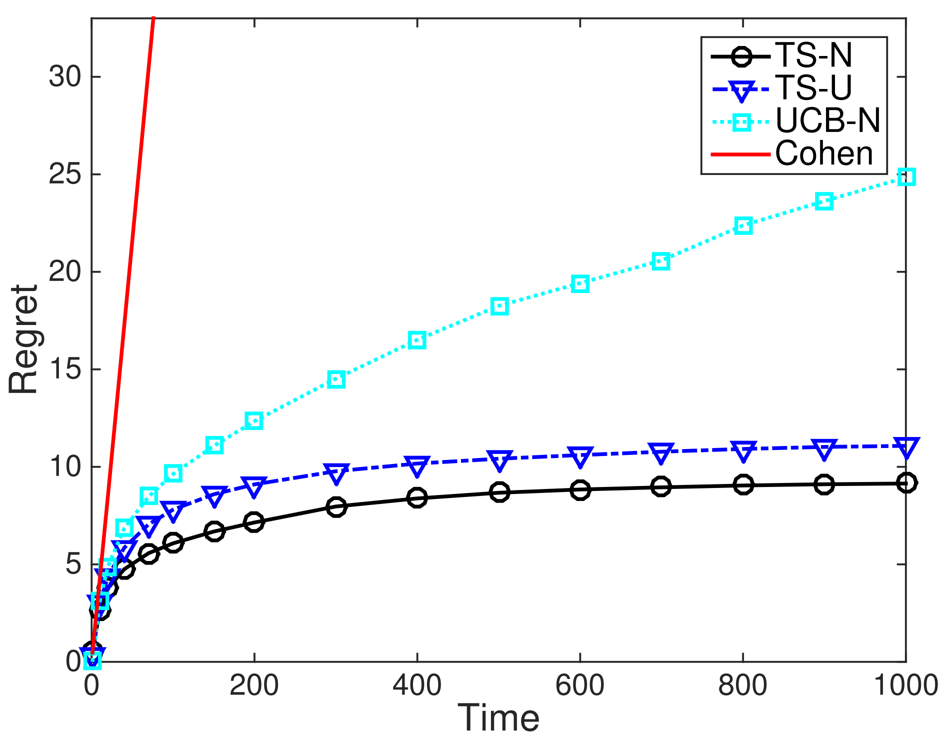
Graphical Bandits

- Time-variant graph setting
  - Unknown graph structure
  - Worst case: graph is generated by opponent. Never able to learn the graph
  - However, free side observations improves the learning performance.
- TS-N algorithm
  - Update posterior with all observations
  - Sampling $\pi_t = \alpha_t$ where α_t is the posterior over actions that is optimal
  - Problem-independent regret $O(\sqrt{\beta_0(G)T\log K})$ if graph is undirected
- TS-U algorithm
  - Sampling $\pi_t = (1-\epsilon_t)\alpha_t + \epsilon_t \frac{1}{K}$
  - Mixing with uniform distribution allows exploring the graph
  - Problem-independent regret $\tilde{O}(\sqrt{\beta_0(G)T\log K})$ if graph is directed

Graphical Bandits

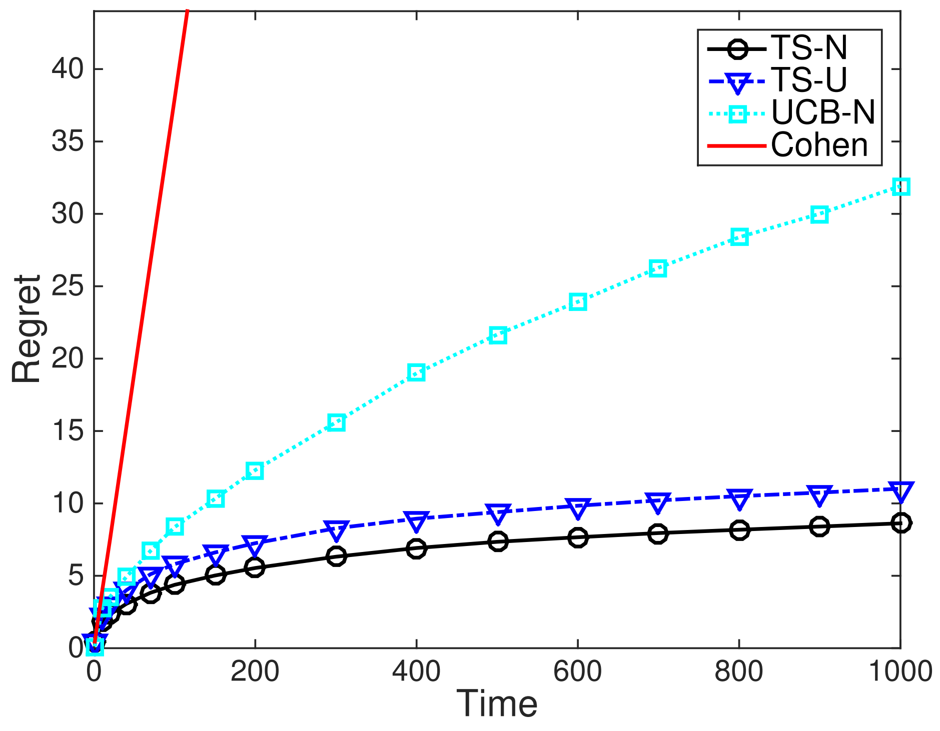
- Numerical Results
  - Bernoulli case in undirected graphs
  - Time-invariant case (left) and time-variant case (right)

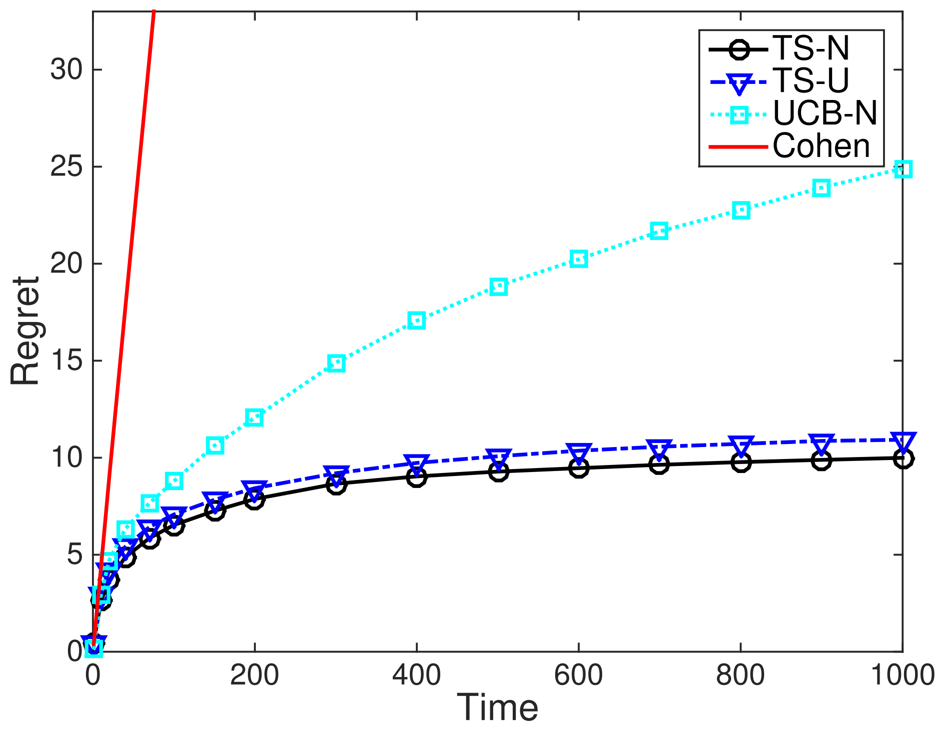




Graphical Bandits

- Numerical Results
  - Bernoulli case in directed graphs
  - Time-invariant case (left) and time-variant case (right)



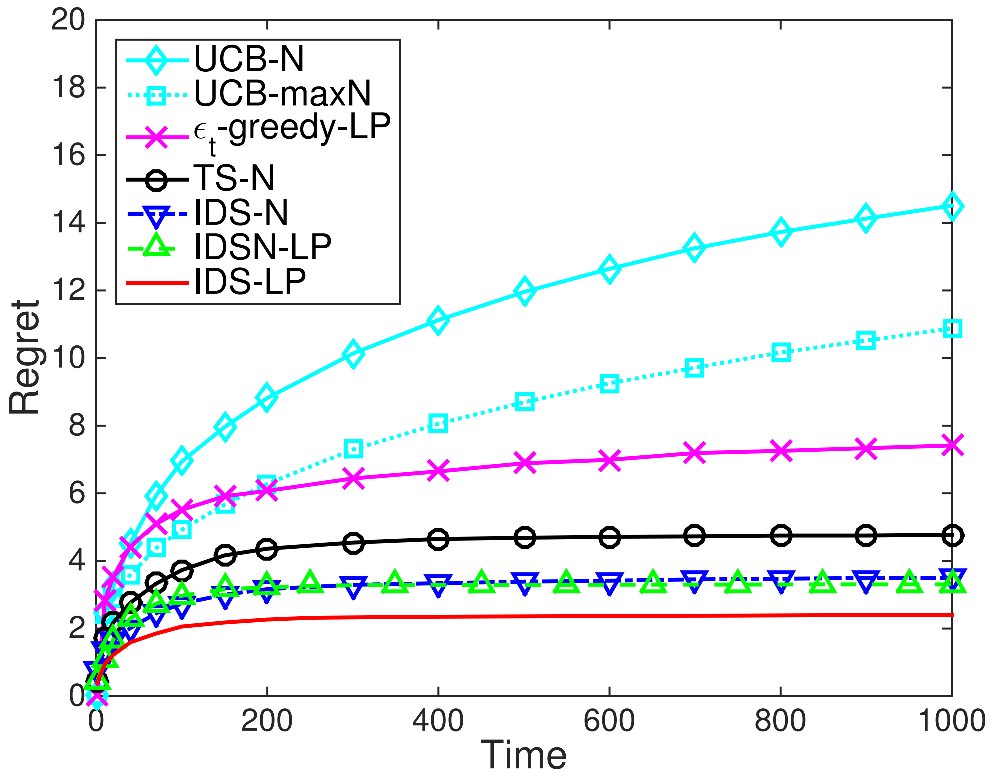


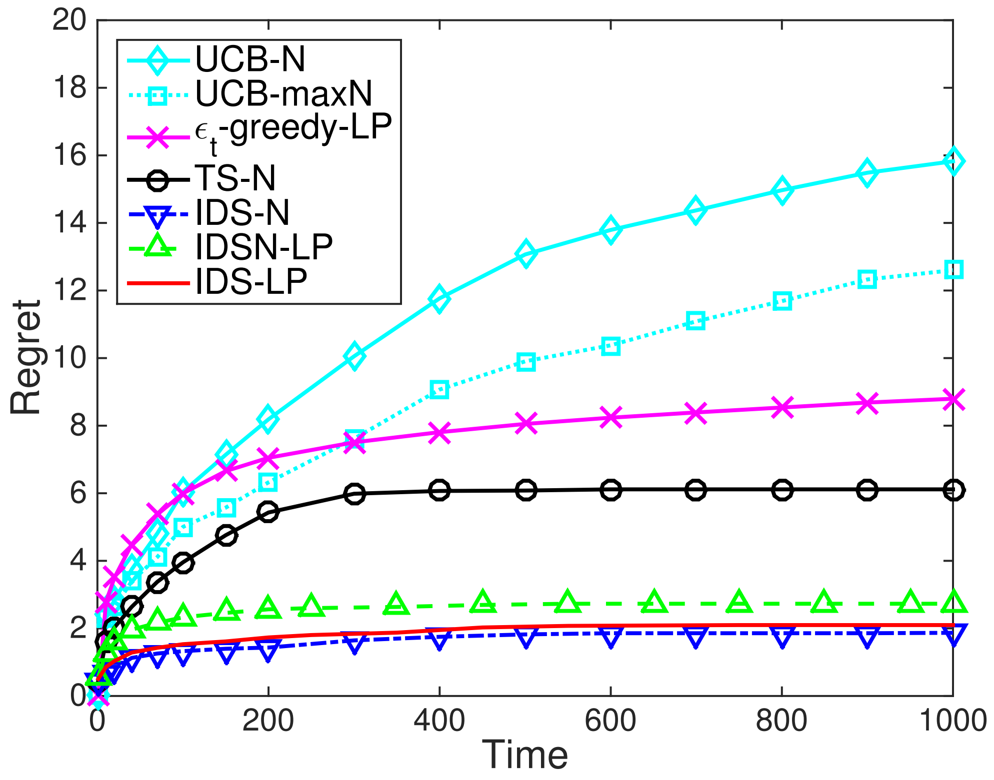
Graphical Bandits

- Time-variant graph setting (cont.)
  - What if the graph is known each time?
- Information Directed Sampling
  - Update posterior with all observations
  - Sampling actions according to $\arg\min_{\pi_t} \frac{(\pi_t^T \Delta_t)^2}{\pi_t^T G_t h_t}$, that min. information ratio
  - where G_t is graph information, Δ_t is expected regret, h_t is information gain.
  - IDS-N Enjoys same regret bound as TS-N, and better empirical performance.
  - Can be generalized to (Erdos-Renyi) random graph feedback
  - Relax the optimization problem => variants of IDS
  - However, more computation cost than TS-N.

Graphical Bandits

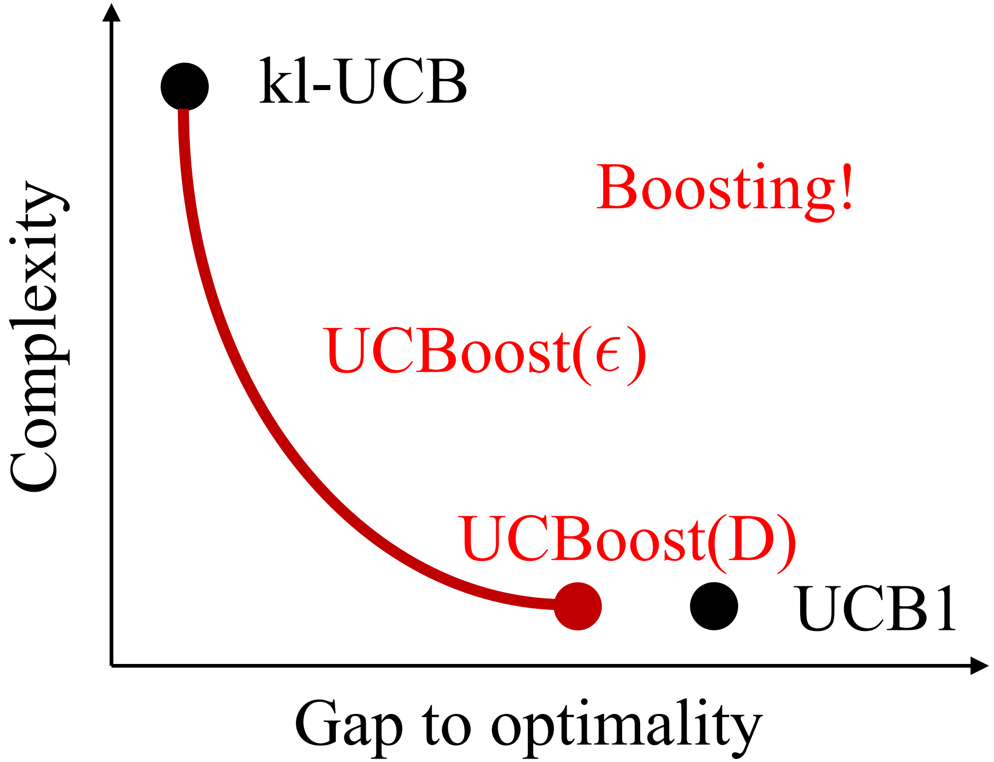
- Numerical Results
  - Bernoulli case
  - Time-invariant case (left) and time-variant case (right)





Boosting Bandits

- Complexity vs Optimality Dilemma
  - Optimal algorithms involve optimization problems: kl-UCB
  - Simple algorithms are far from being optimal: UCB1
- UCBoost algorithms [4]
  - Ensemble a set of "weak" but closed-form UCB-type algorithms
  - Offer trade-off between complexity and optimality with guarantees



	kl-UCB	UCBoost(ϵ)	UCBoost(D)	UCB1
Regret/log(T)	$O\left(\sum_a \frac{\mu^*-\mu_a}{d_{kl}(\mu_a,\mu^*)}\right)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{d_{kl}(\mu_a,\mu^*)-\epsilon}\right)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{d_{kl}(\mu_a,\mu^*)-1/e}\right)$	$O\left(\sum_a \frac{\mu^*-\mu_a}{2(\mu^*-\mu_a)^2}\right)$
Complexity	unbounded	$O(\log(1/\epsilon))$	$O(1)$	O(1)

[4] Fang Liu, Sinong Wang, Swapna Buccapatnam and Ness Shroff, "UCBoost: A Boosting Approach to Tame Complexity and Optimality for Stochastic Bandits", in IJCAI 2018.

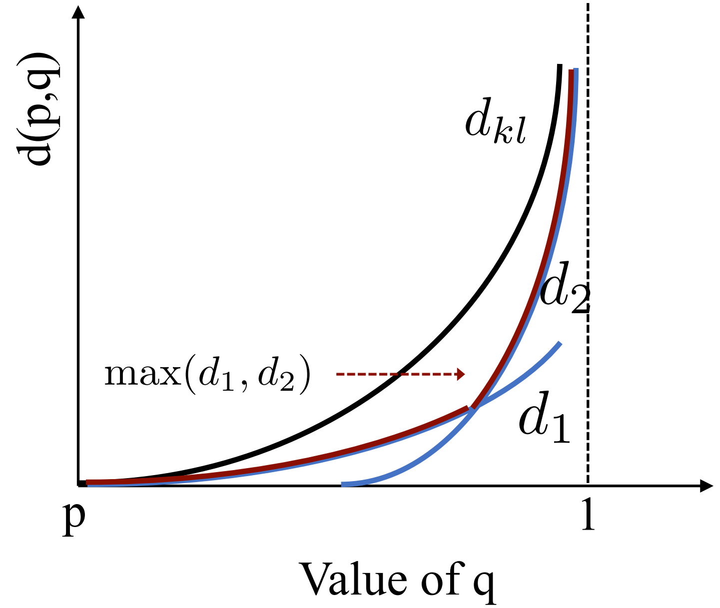
Boosting Bandits

- Understanding UCBoost
  - UCB kernel is a distance function d, associated with $$P(d): \max_{q \in \Theta} \quad q \qquad s.t. \quad d(p,q) \leq \delta$$
    - kl-UCB: $d_{kl}(p,q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$
    - UCB1: $d_{sq}(p,q) = 2(p-q)^2$
  - UCBoost ensembles a set D of distance functions (UCB-types algorithms) by taking the minimum
  - For each d in D, P(d) has closed-form solutions.
- UCBoost(D)
  - Ensemble a fixed (finite) set of distance functions
- UCBoost(ϵ)
  - Ensemble an infinite set of step functions + one distance function
  - Bisection search

Boosting Bandits

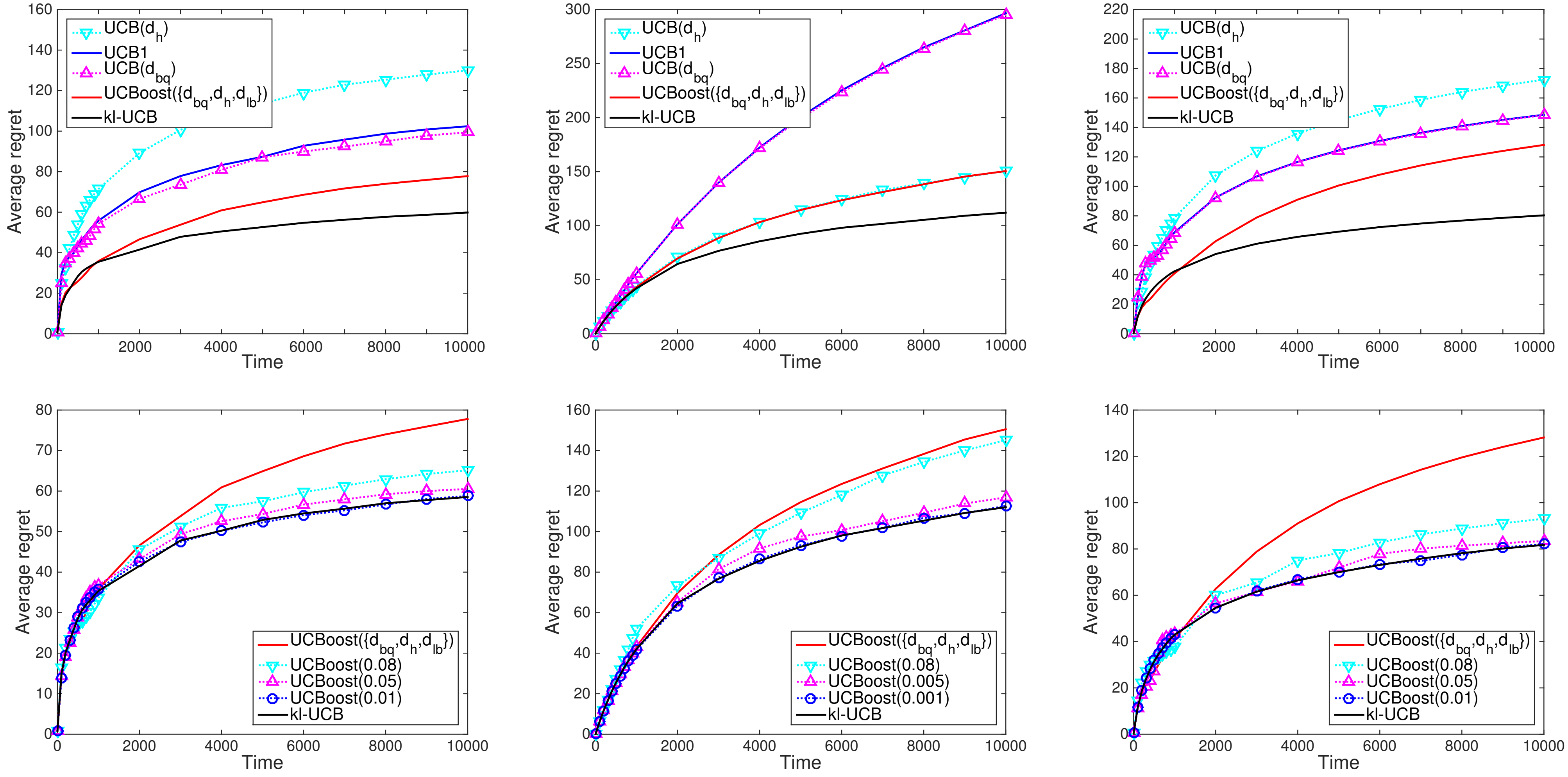
- Why taking the minimum?
- Philosophy of voting
  - Majority vote? No!
  - If the ordering is known, follow the leader.
  - UCBoost takes the minimum, thus the tightest upper confidence bound.
- Geometric view of UCBoost
  - Kernel of UCBoost is $\max_{d \in D} d$
  - Taking the minimum = solving $P\left(\max_{d \in D} d\right)$
  - The closer to KL divergence, the better regret

	UCB1	UCB2	UCB3	UCBoost
	0.9	0.8	0.6	0.6
	0.8	0.75	0.7	0.7
	0.2	0.2	0.3	0.2
decision	1	1	2	2



Boosting Bandits

- Numerical Results



(a) Bernoulli scenario 1 (b) Bernoulli scenario 2 (c) Beta scenario

Boosting Bandits

- Computational Costs per arm per round

Scenario	kl-UCB	UCBoost(ϵ) $\epsilon = 0.01(0.001)$	UCBoost(ϵ) $\epsilon = 0.05(0.005)$	UCBoost(ϵ) $\epsilon = 0.08$	UCBoost($\{d_{bq}, d_h, d_{lb}\}$)	UCB1
Bernoulli 1	$933\mu s$	$7.67\mu s$	$6.67\mu s$	$5.78\mu s$	$1.67\mu s$	$0.31\mu s$
Bernoulli 2	$986\mu s$	$8.76\mu s$	$7.96\mu s$	$6.27\mu s$	$1.60\mu s$	$0.30\mu s$
Beta	$907\mu s$	$8.33\mu s$	$6.89\mu s$	$5.89\mu s$	$2.01\mu s$	$0.33\mu s$

  - 1% computation cost of kl-UCB to achieve competitive regret
  - UCBoost(D) outperforms UCB1

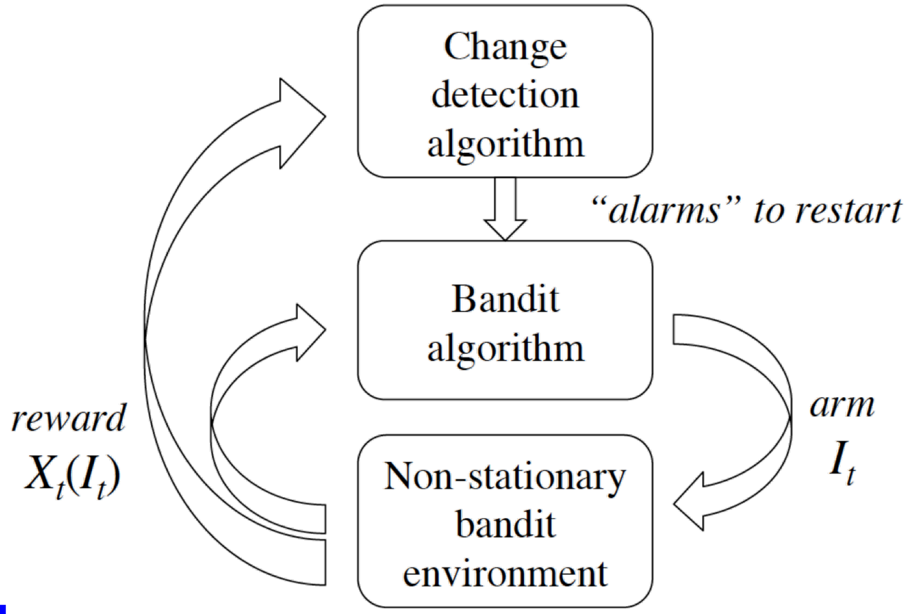
Non-stationary Bandits

- What is non-stationary bandits?
  - The distributions associated with actions may change over time
  - Unknown change points
  - Model varying user preference
- Existing recipes in stochastic domain
  - Discounting: D-UCB [GM2011]
  - Sliding window: SW-UCB [GM2011]
  - Passively adaptive
- Change-detection based framework [5]
  - CD-UCB: UCB with any CD algorithm
  - CUSUM-UCB: Cumulative Sum as CD
  - Actively adaptive

[5] Fang Liu, Joohyun Lee and Ness Shroff, "A Change-Detection based Framework for Piecewise-stationary Multi-Armed Bandit Problem", in AAAI 2018.

Non-stationary Bandits

- Change-detection based framework
  - CD-UCB: develop a general UCB algorithm with any CD element
  - CUSUM-UCB: develop a modified Cumulative Sum as CD element
  - CUSUM-UCB enjoys the best known regret bound

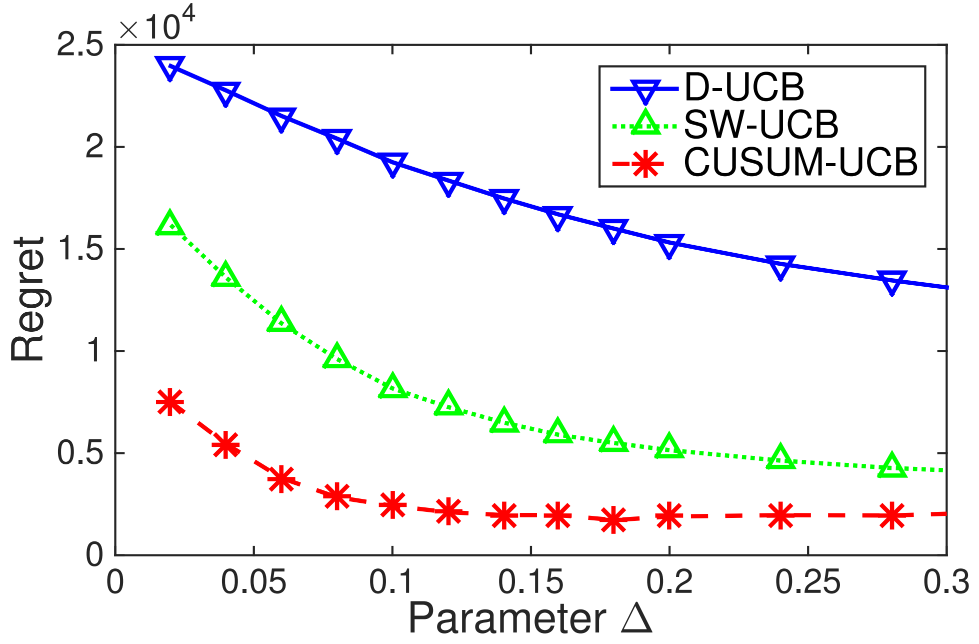


γ_T = number of changes up to time T

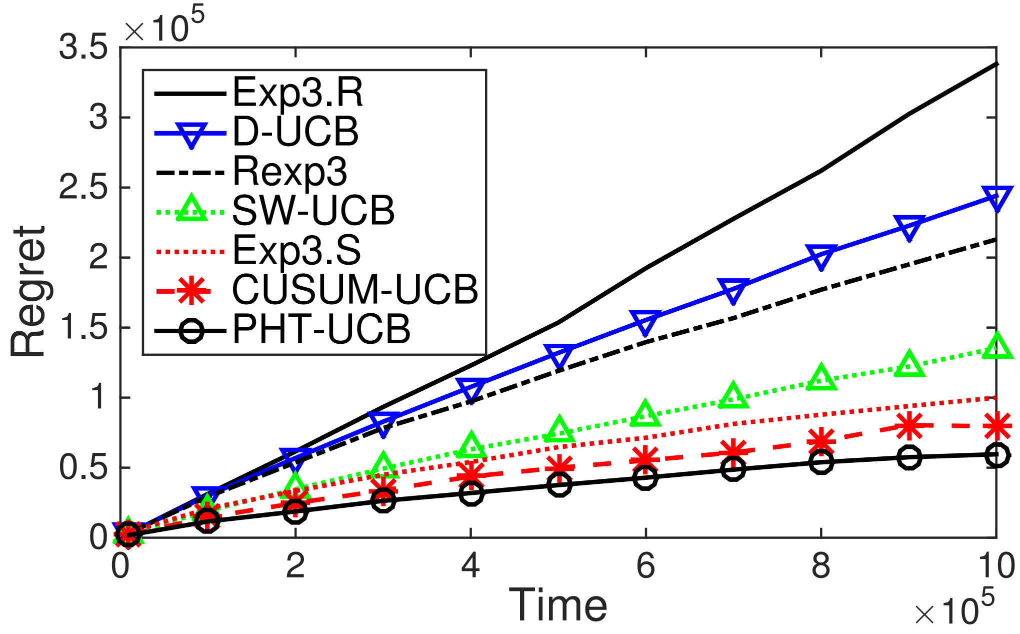
	Passively adaptive			*Actively adaptive*		
Policy	D-UCB (Kocsis and Szepesvári 2006)	SW-UCB (Garivier and Moulines 2008)	Rexp3 (Besbes, Gur, and Zeevi 2014)	Adapt-EvE (Hartland et al. 2007)	CUSUM-UCB	*lower bound* (Garivier and Moulines 2008)
Regret	$O(\sqrt{T\gamma_T}\log T)$	$O(\sqrt{T\gamma_T}\log T)$	$O(V_T^{1/3}T^{2/3})$	Unknown	$O(\sqrt{T\gamma_T \log \frac{T}{\gamma_T}})$	$\Omega(\sqrt{T})$

Non-stationary Bandits

- Numerical Results
  - Flipping environment: 2 Bernoulli arms, $\mu_t(1) = 0.5$, $\mu_t(2) = \begin{cases} 0.5 - \Delta, & \frac{T}{3} \leq t \leq \frac{2T}{3} \\ 0.8, & \text{otherwise} \end{cases}$.
  - Switching environment: $\mu_t(i) = \begin{cases} \mu_{t-1}(i), & \text{with probability } 1 - \beta(t) \\ \mu \sim U[0,1], & \text{with probability } \beta(t) \end{cases}$



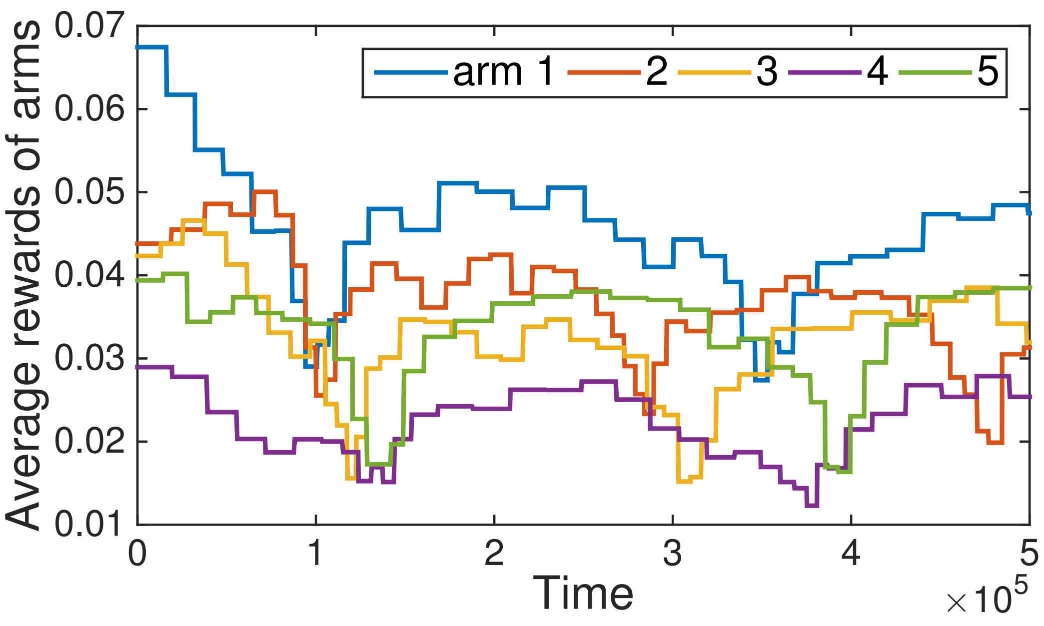
Flipping environment



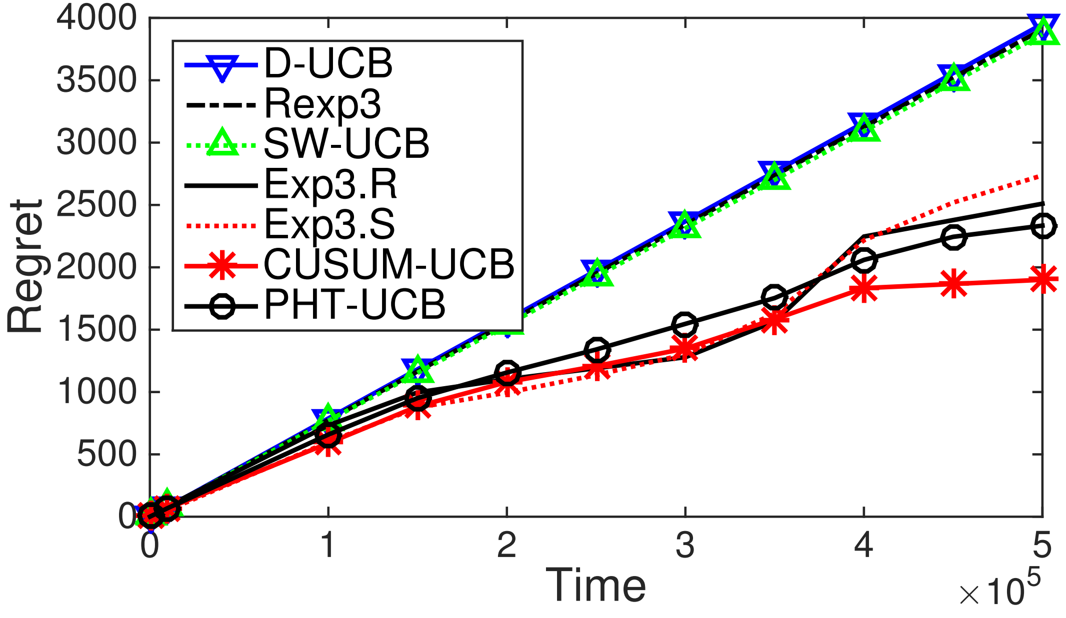
Switching environment

Non-stationary Bandits

- Numerical Results
  - Yahoo! Front Page dataset



Yahoo! ground truth



Yahoo! regret result

Parameterized Clustering Bandits

- Paper in preparation
- General idea:
  - Model correlations by clusters of actions
  - Goal 1: show lower bound result depends on number of clusters
  - Design algorithm that can aggregate the observations in each cluster
  - This involves joint maximum likelihood estimation
  - Goal 2: show upper bound result depends on number of clusters
- Why interesting?
  - # of clusters << # of actions
  - Task scheduling problem: regret depends on “types” of servers

Reference

[MS2011] Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In NIPS, pages 684–692, 2011.

[CKLB2012] S. Caron, B. Kveton, M. Lelarge, and S. Bhagat. Leveraging side observations in stochastic bandits. In UAI, pages 142–151. AUAI Press, 2012.

[BES2014] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B. Shroff. Stochastic bandits with side observations on networks. SIGMETRICS Perform. Eval. Rev., 42(1):289–300, June 2014.

[CHK2016] Alon Cohen, Tamir Hazan, and Tomer Koren. Online learning with feedback graphs without the graphs. ICML 2016.

[TDD2017] Aristide Tossou, Christos Dimitrakakis, and Devdatt Dubhashi. Thompson sampling for stochastic bandits with graph feedback. In AAAI Conference on Artificial Intelligence, 2017.

[GM2008] Garivier, A., and Moulines, E. On upper-confidence bound policies for switching bandit problems. ALT 2011.