Randomization for exploration in bandits and RL

Tor and Csaba

PART I (STOCHASTIC MODELS)

- Finite-armed bandits
- UCB and Thompson sampling
- Bayesian regret
- Linear bandits
- MDPs

PART II (ADVERSARIAL MODELS)

- Adversarial bandits
- Exp3 as follow-the-perturbed-leader
- Combinatorial semibandits
- Follow-the-perturbed-leader for semibandits
- The mirror descent view

Bandits

- A stripped down version of RL
- **Learner** interacts with **environment** over n rounds
- Each round the learner plays **action** $A_t \in \mathcal{A}$
- Receives a reward $X_t \sim P_{A_t}$
- $\{P_a : a \in \mathcal{A}\}$ are **unknown** distributions
- Challenging exploration problem
- No planning
- Finite-armed bandits: $\mathcal{A} = \{1, \dots, k\}$

The learning objective

- Want to maximize reward: $\sum_{t=1}^{n} X_t$

The learning objective

- Want to maximize reward: $\sum_{t=1}^{n} X_t$
- The mean of P_a is μ_a and $\mu^* = \max_a \mu_a$
- Minimise (expected) regret

$$\mathfrak{R}_n = n\mu^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right]$$

- A useful normalisation

A classic algorithm: UCB

- Based on the principle of **optimism in the face of uncertainty**
- Assume P_a is Bernoulli with mean $\mu_a \in [0, 1]$
- Choose each arm once
- Subsequently

$$A_t = \operatorname{argmax}_a \hat{\mu}_a(t-1) + \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}}$$

confidence level

empirical mean of action a

number of plays of action a

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\mu_{A_t} = U_{tA_t} - U_{tA_t} + \mu_{A_t}$$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\begin{aligned}\mu_{A_t} &= U_{tA_t} - U_{tA_t} + \mu_{A_t} \\ &\geq U_{ta^*} - U_{tA_t} + \mu_{A_t}\end{aligned}$$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\begin{aligned}
\mu_{A_t} &= U_{tA_t} - U_{tA_t} + \mu_{A_t} \\
&\geq U_{ta^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \mu_{A_t}
\end{aligned}$$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\begin{aligned}
\mu_{A_t} &= U_{tA_t} - U_{tA_t} + \mu_{A_t} \\
&\geq U_{ta^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \hat{\mu}_{A_t}(t-1) - C_{tA_t}
\end{aligned}$$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\begin{aligned}
\mu_{A_t} &= U_{tA_t} - U_{tA_t} + \mu_{A_t} \\
&\geq U_{ta^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \hat{\mu}_{A_t}(t-1) - C_{tA_t} \\
&= \mu_{a^*} - 2C_{tA_t}
\end{aligned}$$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\begin{aligned}
\mu_{A_t} &= U_{tA_t} - U_{tA_t} + \mu_{A_t} \\
&\geq U_{ta^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \mu_{A_t} \\
&\geq \mu_{a^*} - U_{tA_t} + \hat{\mu}_{A_t}(t-1) - C_{tA_t} \\
&= \mu_{a^*} - 2C_{tA_t}
\end{aligned}$$

$$\boxed{\mu_{a^*} - \mu_{A_t} \leq 2C_{tA_t}}$$

- Define confidence width and UCB's by

$$C_{ta} = \sqrt{\frac{\log(1/\delta)}{2T_a(t-1)}} \qquad U_{ta} = \hat{\mu}_a(t-1) + C_{ta}$$

- With 'high probability' $|\hat{\mu}_a(t-1) - \mu_a| \leq C_{ta}$
- Algorithm plays $A_t = \operatorname{argmax}_a U_{ta}$

$$\mathfrak{R}_n \leq k + \mathbb{E}\left[\sum_{t=k+1}^{n} \mu_{a^*} - \mu_{A_t}\right] \lesssim k + \mathbb{E}\left[\sum_{t=k+1}^{n} \sqrt{\frac{2\log(1/\delta)}{T_{A_t}(t)}}\right]$$

$$= O(\sqrt{kn\log(1/\delta)})$$

$$\mu_{a^*} - \mu_{A_t} \leq 2C_{tA_t}$$

Bayesian viewpoint

- Environment is unknown
- Introduce an environment class $\mathcal{E}$ and prior Q
- Assume the true environment is sampled from measure Q over $\mathcal{E}$ at the beginning
- Reason about true environment using the posterior

$$Q(\nu \in \cdot \mid A_1, X_1, \ldots, A_t, X_t)$$

ν: true env.

Thompson sampling

- A randomised Bayesian algorithm
- Choose a prior over the environments
- Each round: sample an environment from the posterior and play the optimal action in that environment
- A general principle. Works for bandits, linear bandits, RL and more

Thompson sampling

- A randomised Bayesian algorithm
- Choose a prior over the environments
- Each round: sample an environment from the posterior and play the optimal action in that environment
- A general principle. Works for bandits, linear bandits, RL and more
- **Equivalent view** Play each action according to the posterior probability it is optimal

Bernoulli bandits

- Bernoulli bandits are characterised by their means
- $\mathcal{E} \equiv [0, 1]^k$
- Product of Beta distributions for the prior

$$q(\mu) \propto \prod_{a=1}^{k} \mu_a^{\alpha-1}(1-\mu_a)^{\beta-1}$$

Bernoulli bandits

- Bernoulli bandits are characterised by their means
- $\mathcal{E} \equiv [0, 1]^k$
- Product of Beta distributions for the prior

$$q(\mu) \propto \prod_{a=1}^{k} \mu_a^{\alpha-1}(1-\mu_a)^{\beta-1}$$

- Posterior is also Beta

$$q(\mu \mid a_1, x_1, \ldots, a_t, x_t) \propto \prod_{a=1}^{k} \mu_a^{\alpha+s_a-1}(1-\mu_a)^{\beta+f_a-1}$$

$$s_a = \sum_{s=1}^{t} \mathbb{1}(a_s = a, x_s = 1)$$
$$f_a = \sum_{s=1}^{t} \mathbb{1}(a_s = a, x_s = 0)$$

TS for Bernoulli bandits

A simple algorithm

Step 1 For each a sample

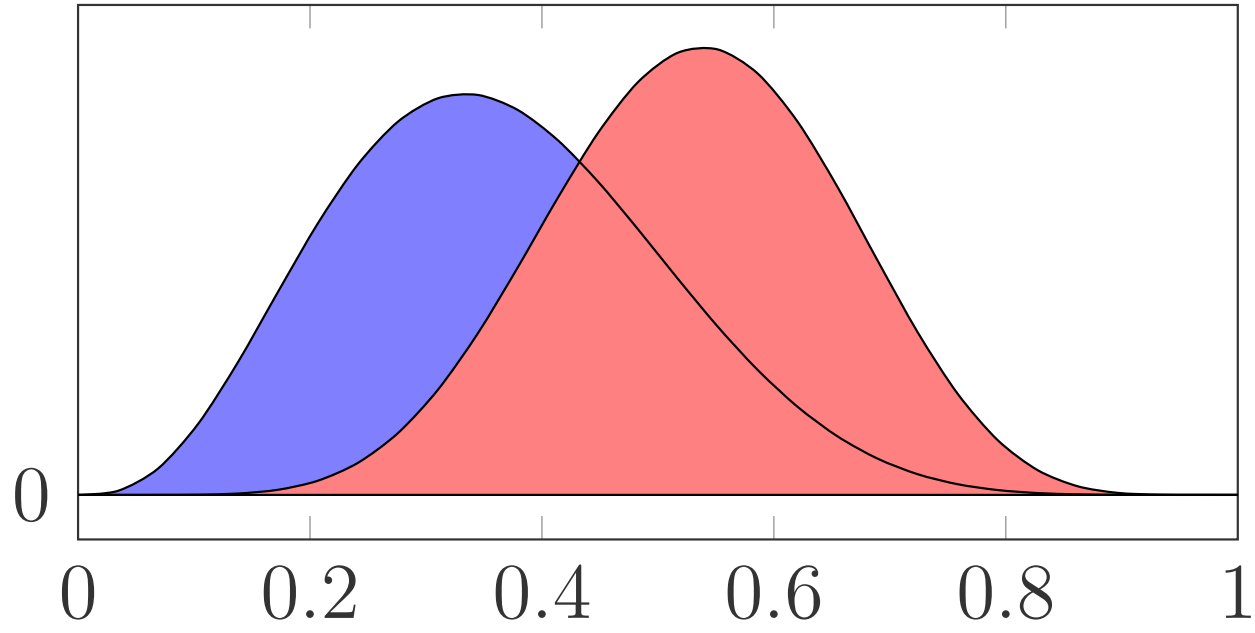
$$\tilde{\mu}_a \sim \mathrm{Beta}(\alpha + S_a(t-1), \beta + F_a(t-1))$$

$S_a(t-1)$: #wins; $F_a(t-1)$: #losses

Step 2 Play $A_t = \mathrm{argmax}_a \, \tilde{\mu}_a$

Note Mean of $\mathrm{Beta}(\alpha, \beta)$ is $\alpha/(\alpha+\beta)$

$\mathrm{Beta}(1,1)$ is uniform



	WINS	LOSSES
ARM 1	8	7
ARM 2	4	7

Plays ARM 1 with probability ~ 0.82

Analysis of TS

- Frequentist regret is delicate and prior dependent
- Bayesian regret

$$\mathfrak{BR}_n = \int_{\mathcal{E}} \mathfrak{R}_n(\nu, \pi) Q(d\nu)$$

Regret

Prior

Environment class

- Less informative
- Easier to analyse

- Optimal action is $A^* = \text{argmax}_a \, \mu_a$

- Optimal action is $A^* = \text{argmax}_a \, \mu_a$
- TS chooses A_t so that $\mathbb{P}_t \left(A_t = a \right) = \mathbb{P}_t \left(A^* = a \right)$

- Optimal action is $A^* = \operatorname{argmax}_a \mu_a$
- TS chooses A_t so that $\mathbb{P}_t(A_t = a) = \mathbb{P}_t(A^* = a)$
- Bayesian regret is $\mathfrak{BR}_n = \underset{\uparrow}{\mathbb{E}}\left[\sum_{t=1}^{n}(\mu_{A^*} - \mu_{A_t})\right]$

wrt prior, algorithm and rewards

- Optimal action is $A^* = \operatorname{argmax}_a \mu_a$
- TS chooses A_t so that $\mathbb{P}_t(A_t = a) = \mathbb{P}_t(A^* = a)$
- Bayesian regret is $\mathfrak{BR}_n = \underset{\uparrow}{\mathbb{E}}\left[\sum_{t=1}^{n}(\mu_{A^*} - \mu_{A_t})\right]$

wrt prior, algorithm and rewards

- Introducing upper confidence bounds

$$\mathbb{E}_t[\mu_{A^*} - \mu_{A_t}] = \mathbb{E}_t[\mu_{A^*} - U_{tA_t}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}]$$

- Optimal action is $A^* = \operatorname{argmax}_a \mu_a$
- TS chooses A_t so that $\mathbb{P}_t(A_t = a) = \mathbb{P}_t(A^* = a)$
- Bayesian regret is $\mathfrak{BR}_n = \underset{\text{wrt prior, algorithm and rewards}}{\mathbb{E}}\left[\sum_{t=1}^{n}(\mu_{A^*} - \mu_{A_t})\right]$
- Introducing upper confidence bounds

$$\begin{aligned}\mathbb{E}_t[\mu_{A^*} - \mu_{A_t}] &= \mathbb{E}_t[\mu_{A^*} - U_{tA_t}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}] \\ &= \mathbb{E}_t[\mu_{A^*} - U_{tA^*}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}]\end{aligned}$$

- Optimal action is $A^* = \operatorname{argmax}_a \mu_a$
- TS chooses A_t so that $\mathbb{P}_t(A_t = a) = \mathbb{P}_t(A^* = a)$
- Bayesian regret is $\mathfrak{BR}_n = \underset{\text{wrt prior, algorithm and rewards}}{\mathbb{E}} \left[\sum_{t=1}^{n} (\mu_{A^*} - \mu_{A_t}) \right]$
- Introducing upper confidence bounds

$$\begin{aligned}
\mathbb{E}_t[\mu_{A^*} - \mu_{A_t}] &= \mathbb{E}_t[\mu_{A^*} - U_{tA_t}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}] \\
&= \mathbb{E}_t[\mu_{A^*} - U_{tA^*}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}] \\
&\overset{\text{whp}}{\lesssim} \mathbb{E}_t[U_{tA_t} - \mu_{A_t}]
\end{aligned}$$

- Optimal action is $A^* = \operatorname{argmax}_a \mu_a$
- TS chooses A_t so that $\mathbb{P}_t(A_t = a) = \mathbb{P}_t(A^* = a)$
- Bayesian regret is $\mathfrak{BR}_n = \underset{\uparrow}{\mathbb{E}}\left[\sum_{t=1}^{n}(\mu_{A^*} - \mu_{A_t})\right]$

  wrt prior, algorithm and rewards
- Introducing upper confidence bounds

$$\begin{aligned}
\mathbb{E}_t[\mu_{A^*} - \mu_{A_t}] &= \mathbb{E}_t[\mu_{A^*} - U_{tA_t}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}] \\
&= \mathbb{E}_t[\mu_{A^*} - U_{tA^*}] + \mathbb{E}_t[U_{tA_t} - \mu_{A_t}] \\
&\overset{\text{whp}}{\lesssim} \mathbb{E}_t[U_{tA_t} - \mu_{A_t}] \\
&\overset{\text{whp}}{\lesssim} \mathbb{E}_t\left[\sqrt{\frac{\log(1/\delta)}{1 \vee T_{A_t}(t-1)}}\right]
\end{aligned}$$

- Continuing...

$$\begin{aligned}
\mathfrak{BR}_n &= \mathbb{E}\left[\sum_{t=1}^{n} (\mu_{A^*} - \mu_{A_t})\right] \\
&\lesssim \mathbb{E}\left[\sum_{t=1}^{n} \sqrt{\frac{\log(1/\delta)}{1 \vee T_{A_t}(t-1)}}\right] \\
&= O\left(\sqrt{nk \log(1/\delta)}\right)
\end{aligned}$$

- $\delta \approx 1/n$ as usual

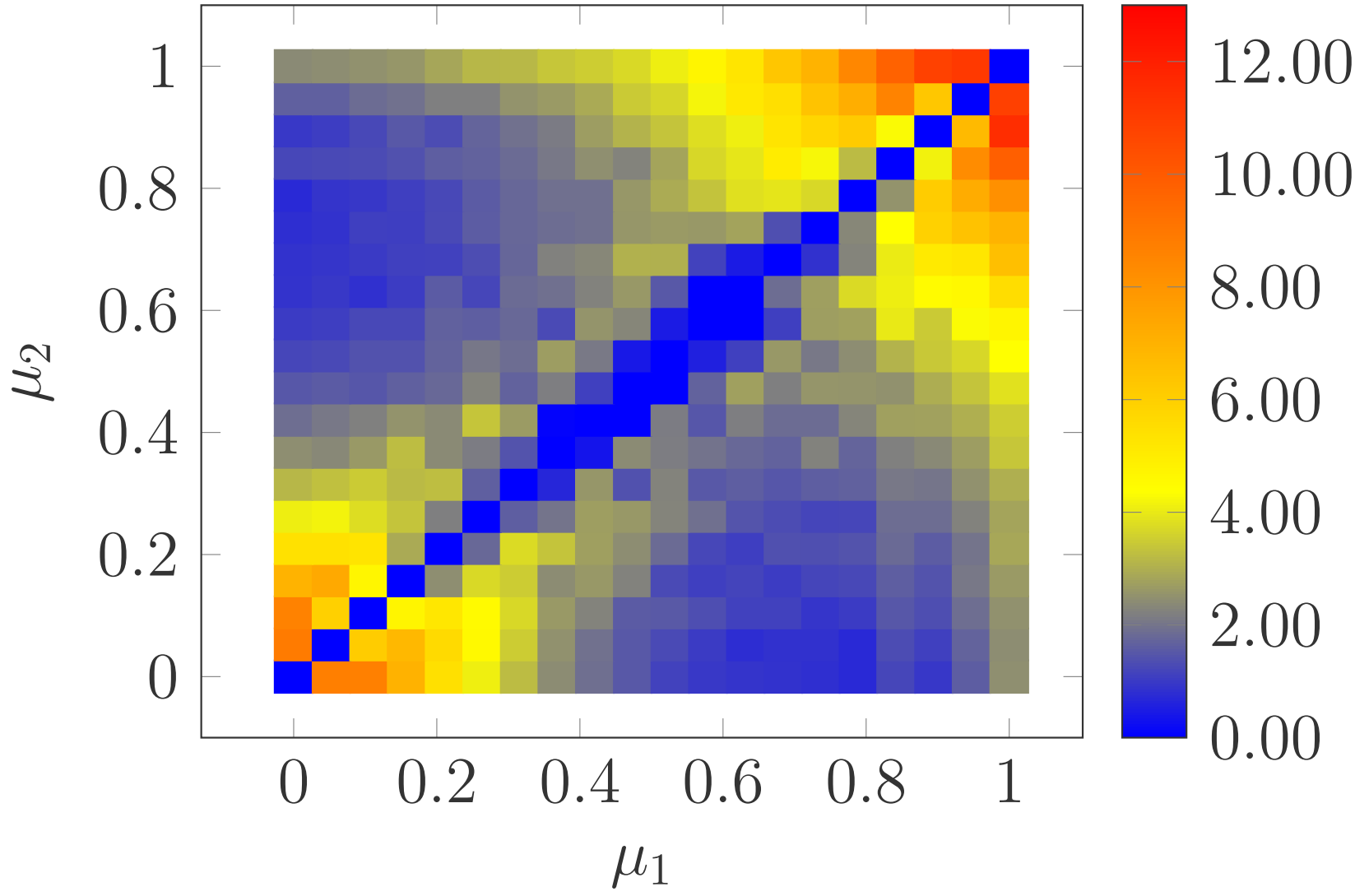
- Continuing...

$$
\begin{aligned}
\mathfrak{BR}_n &= \mathbb{E}\left[\sum_{t=1}^{n} (\mu_{A^*} - \mu_{A_t})\right] \\
&\lesssim \mathbb{E}\left[\sum_{t=1}^{n} \sqrt{\frac{\log(1/\delta)}{1 \vee T_{A_t}(t-1)}}\right] \\
&= O\left(\sqrt{nk \log(1/\delta)}\right)
\end{aligned}
$$

- $\delta \approx 1/n$ as usual
- Can be improved to $O(\sqrt{nk})$
- Optimal in the **worst case**

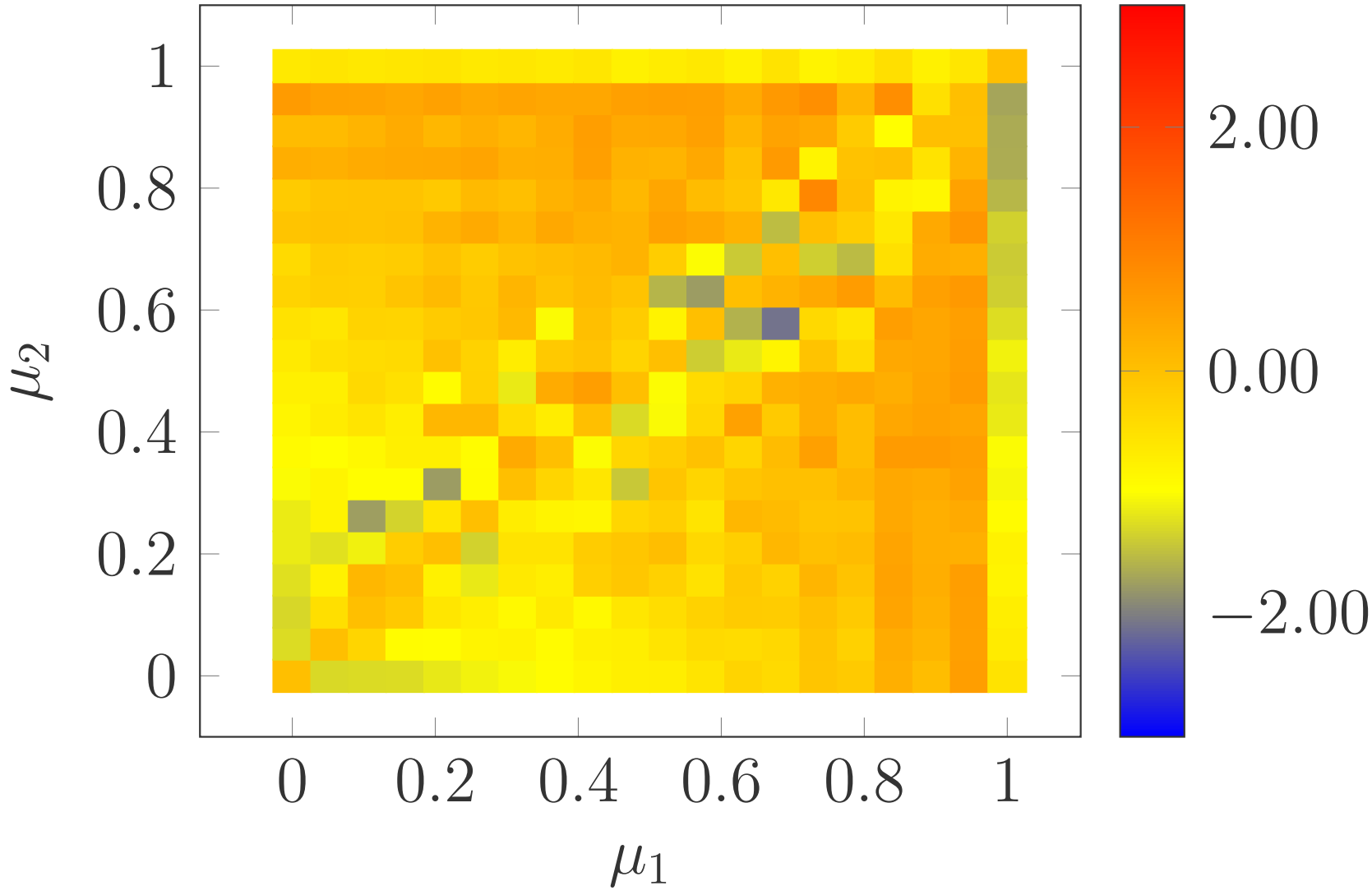
Comparison to UCB

Regret difference, $n = 1000$



Comparison to KL-UCB+

Regret difference, $n = 1000$



KL-UCB+

- UCB with tighter confidence bounds
- Redefine upper confidence bound

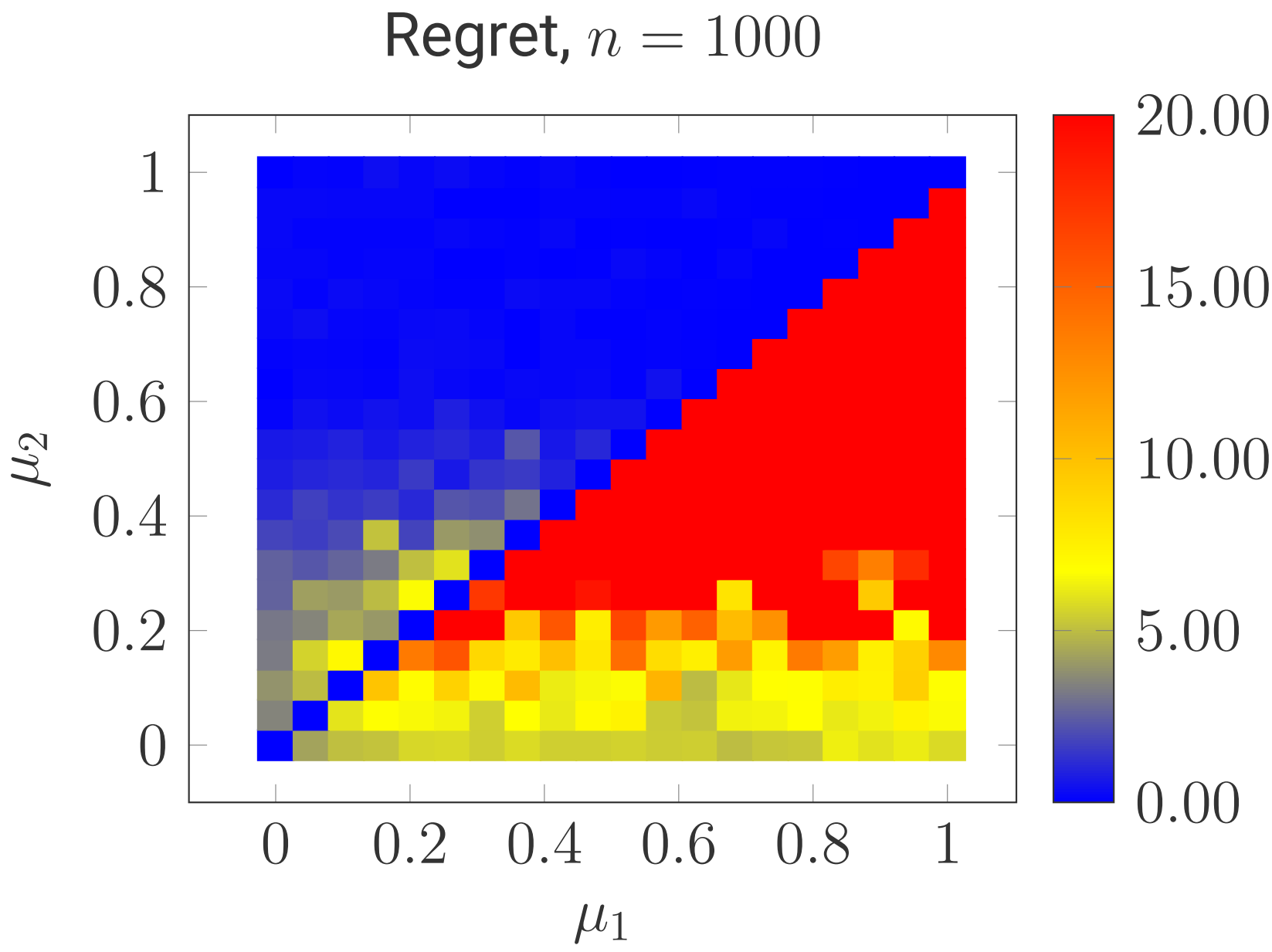
$$U_{ta} = \max \left\{ \tilde{\mu} : d(\hat{\mu}_a(t-1), \tilde{\mu}) \leq \frac{\log\left(\frac{t}{T_a(t-1)}\right)}{T_a(t-1)} \right\}$$

- $d(p, q) = p \log(p/q) + (1-p) \log((1-p)/(1-q))$
- Harder to analyze
- Deriving form of confidence bound non-trivial

Prior sensitivity

- **Standard statistics** Poorly chosen priors are quickly washed away by data
- **Reinforcement learning** Poorly chosen priors change the observed data
- Recovery can take a long time

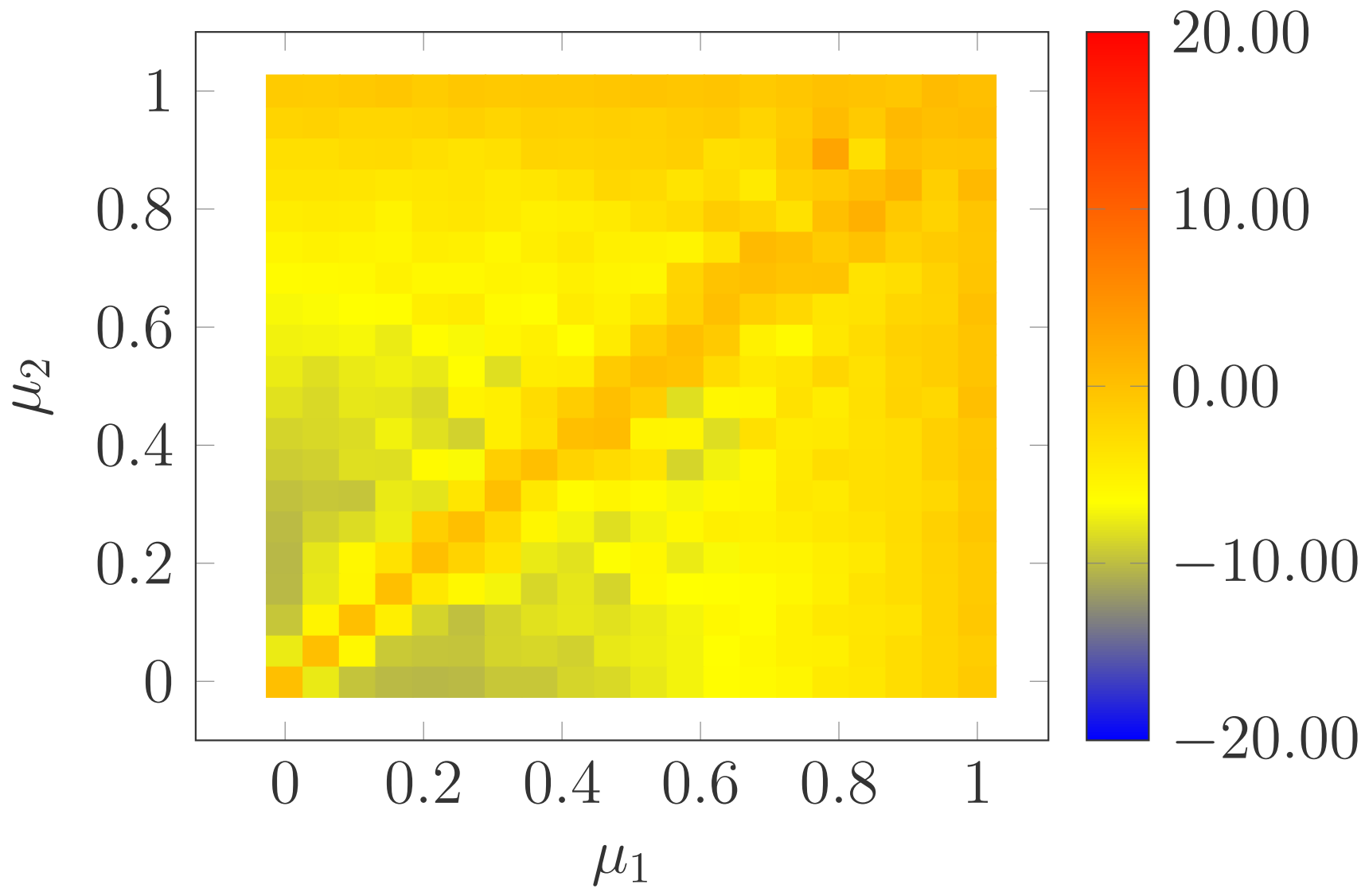
First arm $\alpha = 1, \beta = 10$ and second arm $\alpha = 1$ and $\beta = 1$



More optimistic priors improve robustness

- **Usual story** Optimism encourages exploration
- We don’t know of any theory exploring this
- Roughly though – frequentist proofs work by showing algorithm is ‘optimistic’ with large enough probability

Comparing $\alpha = 1$ and $\beta = 1$ with $\alpha = 10$ and $\beta = 1$



Frequentist results

- Thompson sampling is asymptotically optimal for any α, β

$$\lim_{n\to\infty} \frac{\mathfrak{R}_n}{\log(n)} = \sum_{a:\Delta_a>0} \frac{\Delta_a}{d(\mu_a, \mu^*)} \qquad \Delta_a = \mu^* - \mu_a$$

- Large lower-order terms for some α, β
- KL-UCB(+/*/++) are also asymptotically optimal
- **Conjecture** For Beta prior the regret of TS is

$$\max_{\nu} \mathbb{R}_n(\nu, \pi) = \Omega\left(\sqrt{nk\log(k)}\right)$$

- Suboptimal by a factor of $\sqrt{\log(k)}$

Bayesian results (cont)

- For ‘normal’ priors

$$\lim_{n\to\infty} \frac{\mathfrak{BR}_n^*(Q)}{\log(n)^2} = C_Q \qquad \mathfrak{BR}_n^* = \inf_\pi \mathfrak{BR}_n(Q,\pi)$$

- Proven by integrating frequentist bound for version of KL-UCB
- Unknown if Thompson sampling achieves this

Bayesian regret and the minimax

- Sion's minimax theorem

$$\sup_Q \inf_\pi \mathfrak{BR}_n(Q, \pi) = \inf_\pi \sup_Q \mathfrak{BR}_n(Q, \pi)$$
$$= \inf_\pi \sup_\nu \mathfrak{R}_n(\nu, \pi)$$

- Exists a Bayesian policy that is minimax optimal
- Thompson sampling need not be (near) minimax optimal

Pros and cons

- TS is simple
- **No confidence set construction**
- No tuning (except with prior)
- No tuning
- *Apparently* some increased variance
- Slightly worse than carefully tuned alternatives
- Sensitive to prior
- *Probably* not minimax optimal
- Tricky frequentist analysis

Linear bandits

- Learner chooses actions $A_t \in \mathcal{A} \subset \mathbb{R}^d$
- Reward is

unknown in $\mathbb{R}^d$ noise

$$X_t = \langle A_t, \theta \rangle + \eta_t$$

- Optimistic algorithm $A_t = \operatorname{argmax}_{a \in \mathcal{A}} \max_{\tilde{\theta} \in \mathcal{C}_{t-1}} \langle a, \tilde{\theta} \rangle$
- $\mathcal{C}_t$ is a confidence set, usually elliptical

$$\mathcal{C}_t = \left\{ \tilde{\theta} : \|\hat{\theta}_{t-1} - \tilde{\theta}\|_{G_t}^2 \leq \beta \right\}$$

least squares estimate

Gram matrix $\sum_{s=1}^{t-1} A_s A_s^\top$

Linear bandits

- Learner chooses actions $A_t \in \mathcal{A} \subset \mathbb{R}^d$
- Reward is

unknown in $\mathbb{R}^d$ noise

$$X_t = \langle A_t, \theta \rangle + \eta_t$$

- Optimistic algorithm $A_t = \operatorname{argmax}_{a \in \mathcal{A}} \max_{\tilde{\theta} \in \mathcal{C}_{t-1}} \langle a, \tilde{\theta} \rangle$
- $\mathcal{C}_t$ is a confidence set, usually elliptical

$$\mathcal{C}_t = \left\{ \tilde{\theta} : \|\hat{\theta}_{t-1} - \tilde{\theta}\|^2_{G_t} \leq \beta \right\}$$

least squares estimate Gram matrix $\sum_{s=1}^{t-1} A_s A_s^\top$

- Bilinear programming can be hard!

TS for linear bandits

- Multivariate Gaussian prior, $Q = \mathcal{N}(0, \lambda I)$
- Posterior is $\mathcal{N}(\hat{\theta}_{t-1}, V_{t-1}^{-1})$ with

$$\hat{\theta}_{t-1} = V_{t-1} \sum_{s=1}^{t-1} A_s X_s \quad \text{and} \quad V_{t-1} = \lambda I + \sum_{s=1}^{t-1} A_s A_s^\top$$

- TS samples $\tilde{\theta} \sim \mathcal{N}(\hat{\theta}_{t-1}, V_{t-1}^{-1})$
- Chooses $A_t = \operatorname{argmax}_{a \in \mathcal{A}} \langle a, \tilde{\theta} \rangle$

linear program

- Linear programming is 'easy'

Theoretic results

- Optimistic algorithm: $\mathfrak{R}_n = \tilde{O}(d\sqrt{n})$
- TS: $\mathfrak{BR}_n = \tilde{O}(d\sqrt{n})$
- The same proof idea before
- TS: $\mathfrak{R}_n = \tilde{O}(d^{3/2}\sqrt{n})$
- Proof only works when $\tilde{\theta}$ is sampled from

$$\mathcal{N}(\hat{\theta}_t, \beta V_{t-1}^{-1}) \quad \text{with } \beta \approx d$$

- Inflation of posterior is not really justified
- Empirically not required

Thompson sampling or Follow-the-perturbed-leader?

- **Follow-the-leader** Estimate θ by $\hat{\theta}_{t-1}$
- Choose $A_t = \operatorname{argmax}_{a \in \mathcal{A}} \langle a, \hat{\theta}_{t-1} \rangle$
- Does not explore enough
- **Follow-the-perturbed-leader**
- Choose $A_t = \operatorname{argmax}_{a \in \mathcal{A}} \langle a, \hat{\theta}_{t-1} + \xi_{t-1} \rangle$
- ξ_{t-1} is a carefully chosen perturbation
- TS for linear bandits, $\xi_{t-1} = \mathcal{N}(0, \beta V_{t-1}^{-1})$

Pros and cons

- Simple design
- **No confidence set construction**
- Superb empirical performance
- Easy Bayesian regret analysis
- No frequentist analysis for ‘real thing’
  (you can resolve this)
- Maybe not minimax optimal (not just logs)
  (you can resolve this too)

TS for episodic MDPs

- MDP $M = (\mathcal{S}, \mathcal{A}, P, r)$
- $\mathcal{S}$ and $\mathcal{A}$ are finite **state** and **action** spaces
- $P_a(s, s')$ is the probability of **transitioning** from state s to s' when taking action a
- $r_a(s)$ is the **reward** when taking action a in state s
- $s_\circ \in \mathcal{S}$ is an initial state
- h is the episode length

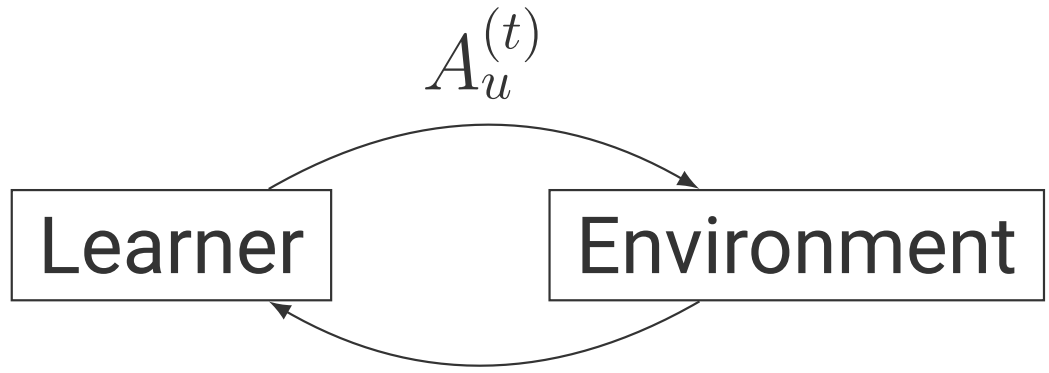
TS for episodic MDPs

- MDP $M = (\mathcal{S}, \mathcal{A}, P, r)$
- $\mathcal{S}$ and $\mathcal{A}$ are finite **state** and **action** spaces
- $P_a(s, s')$ is the probability of **transitioning** from state s to s' when taking action a
- $r_a(s)$ is the **reward** when taking action a in state s
- $s_\circ \in \mathcal{S}$ is an initial state
- h is the episode length
- P **unknown** and r **known**

Interaction protocol

n phases

Phases last h rounds



$$S_1^{(1)}, A_1^{(1)}, S_2^{(2)}, \ldots, A_h^{(1)}$$

$$\vdots$$

$$S_1^{(n)}, A_1^{(n)}, S_2^{(n)}, \ldots, A_h^{(n)}$$

Regret

- A policy is a function π from histories to actions
- Value of policy π is

$$v^{\pi} = \mathbb{E}_{\pi}\left[\sum_{u=1}^{h} r_{A_t}(S_t)\right]$$

- Optimal policy $\pi^* = \operatorname{argmax}_{\pi} v^{\pi}$
- An algorithm produces a sequence of policies $\pi_1, \ldots, \pi_n$
- Regret is $\mathfrak{R}_n = \mathbb{E}\left[\sum_{t=1}^{n} (v^* - v^{\pi_t})\right]$

Optimistic algorithm

- Build a confidence set for each $P_a(s)$
- Act optimistically

Optimistic algorithm

- Build a confidence set for each $P_a(s)$
- Act optimistically
- Often way too conservative
  (don't *really* know how to build confidence sets)

Optimistic algorithm

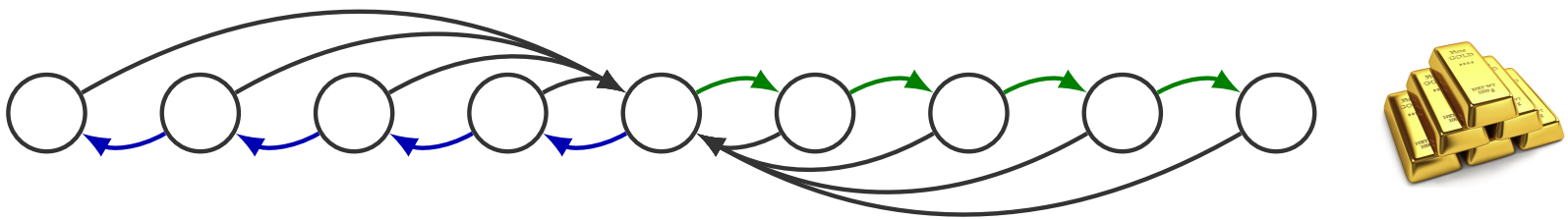
- Build a confidence set for each $P_a(s)$
- Act optimistically
- Often way too conservative
  (don't *really* know how to build confidence sets)
- Thompson sampling acts as normal, *resampling after each episode*
- Need to choose prior on space of MDPs

‘Dithering’

- Why not resample every round?

‘Dithering’

- Why not resample every round?
- Causes dithering



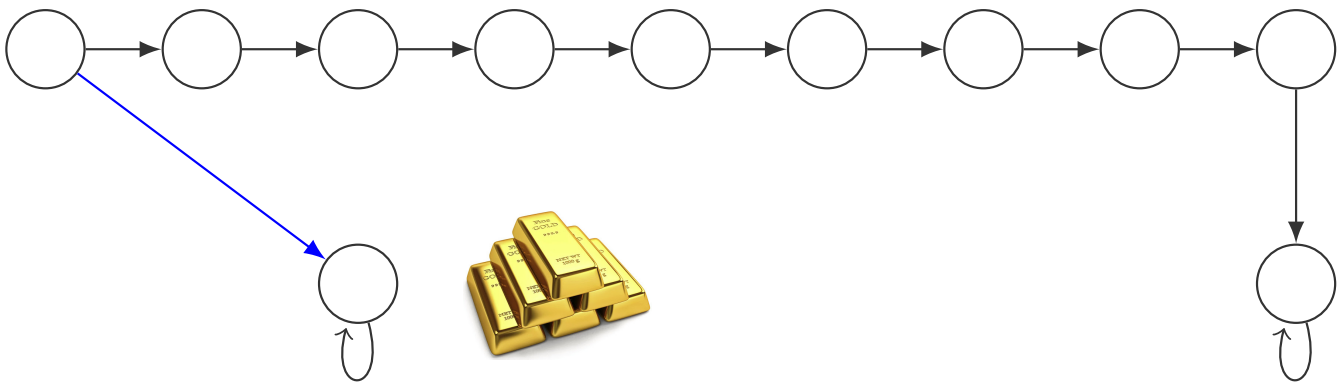
Theoretical results

- Optimistic algorithm

$$\mathfrak{R}_n = O(\sqrt{|\mathcal{S}||\mathcal{A}|n}) + \text{lower order terms}$$

- TS: $\mathfrak{BR}_n = O(\sqrt{|\mathcal{S}||\mathcal{A}|n}) + \text{lower order terms}$
- TS: $\mathfrak{R}_n = O(??)$
- Prior dependence is much more complicated

Beware the prior



Thompson sampling-esque ideas elsewhere

- Spectral bandits
- Generalized linear models
- Linear quadratic control
- **Beyond MDPs** Randomized value functions, function approximation

PART II (ADVERSARIAL MODELS)

Adversarial bandits

- Adversary secretly chooses losses $y_1, \ldots, y_n$
- $y_t \in [0, 1]^k$
- Learner chooses $A_t \in \{1, \ldots, k\}$
- Receives reward y_{tA_t}
- Regret is

$$\mathfrak{R}_n = \mathbb{E}\left[\sum_{t=1}^{n} x_{tA_t}\right] - \min_a \sum_{t=1}^{n} y_{ta}$$

wrt randomness in algorithm

The necessity of randomization

- Suppose A_t is a function of $y_{1A_1}, \ldots, y_{t-1,A_{t-1}}$
- $y_{tA_t} = 1$ and $y_{ta} = 0$ for $a \neq A_t$
- Algorithm always suffers unit loss

The necessity of randomization

- Suppose A_t is a function of $y_{1A_1}, \ldots, y_{t-1,A_{t-1}}$
- $y_{tA_t} = 1$ and $y_{ta} = 0$ for $a \neq A_t$
- Algorithm always suffers unit loss
- By pigeonhole principle,

$$\sum_{t=1}^{n} \sum_{a} y_{ta} = n \qquad \Longrightarrow \qquad \min_{a} \sum_{t=1}^{n} y_{ta} \leq \frac{n}{k}$$

The necessity of randomization

- Suppose A_t is a function of $y_{1A_1}, \ldots, y_{t-1,A_{t-1}}$
- $y_{tA_t} = 1$ and $y_{ta} = 0$ for $a \neq A_t$
- Algorithm always suffers unit loss
- By pigeonhole principle,

$$\sum_{t=1}^{n} \sum_{a} y_{ta} = n \qquad \Longrightarrow \qquad \min_{a} \sum_{t=1}^{n} y_{ta} \leq \frac{n}{k}$$

- Hence $\mathfrak{R}_n \geq \frac{n(k-1)}{k}$ **this is bad!**

How does randomization help?

- Fundamental problem in bandits
- Rewards for unplayed arms are **not observed**
- Stochastic bandits: mean can still be estimated
- No such thing in adversarial bandits
- **Magic trick** Estimate y_{ta} by $\hat{y}_{ta} = \frac{\mathbb{1}(A_t = a)}{\mathbb{P}_t(A_t = a)} y_{ta}$

How does randomization help?

- Fundamental problem in bandits
- Rewards for unplayed arms are **not observed**
- Stochastic bandits: mean can still be estimated
- No such thing in adversarial bandits
- **Magic trick** Estimate y_{ta} by $\hat{y}_{ta} = \dfrac{\mathbb{1}(A_t = a)}{\mathbb{P}_t(A_t = a)} y_{ta}$

$$\mathbb{E}_t[\hat{y}_{ta}] = \frac{\mathbb{E}_t[\mathbb{1}(A_t = a)]}{\mathbb{P}_t(A_t = a)} y_{ta} = y_{ta}$$

How does randomization help?

- Fundamental problem in bandits
- Rewards for unplayed arms are **not observed**
- Stochastic bandits: mean can still be estimated
- No such thing in adversarial bandits
- **Magic trick** Estimate y_{ta} by $\hat{y}_{ta} = \dfrac{\mathbb{1}(A_t = a)}{\mathbb{P}_t(A_t = a)} y_{ta}$

$$\mathbb{E}_t[\hat{y}_{ta}] = \frac{\mathbb{E}_t[\mathbb{1}(A_t = a)]}{\mathbb{P}_t(A_t = a)} y_{ta} = y_{ta}$$

- Only works when $\mathbb{P}_t(A_t = a) > 0$ for all a

Follow-the-perturbed-leader again

- Estimate rewards using importance-weighted estimator
- Sample perturbation $\xi_t \in \mathbb{R}^k$
- Choose $A_t = \operatorname{argmax}_a \left(\xi_{ta} - \eta \sum_{s=1}^{t-1} \hat{y}_{ta} \right)$

learning rate

- Distribution of perturbations determines exploration

Follow-the-perturbed-leader again

- **Gumbel distribution** $F(x) = 1 - \exp(-\exp(x))$
- **Gumbel trick** when ξ_{ta} is a standard Gumbel,

$$\mathbb{P}_t(A_t = a) = \frac{\exp\left(-\eta \sum_{s=1}^{t-1} \hat{y}_{ta}\right)}{\sum_b \exp\left(-\eta \sum_{s=1}^{t-1} \hat{y}_{ta}\right)}$$

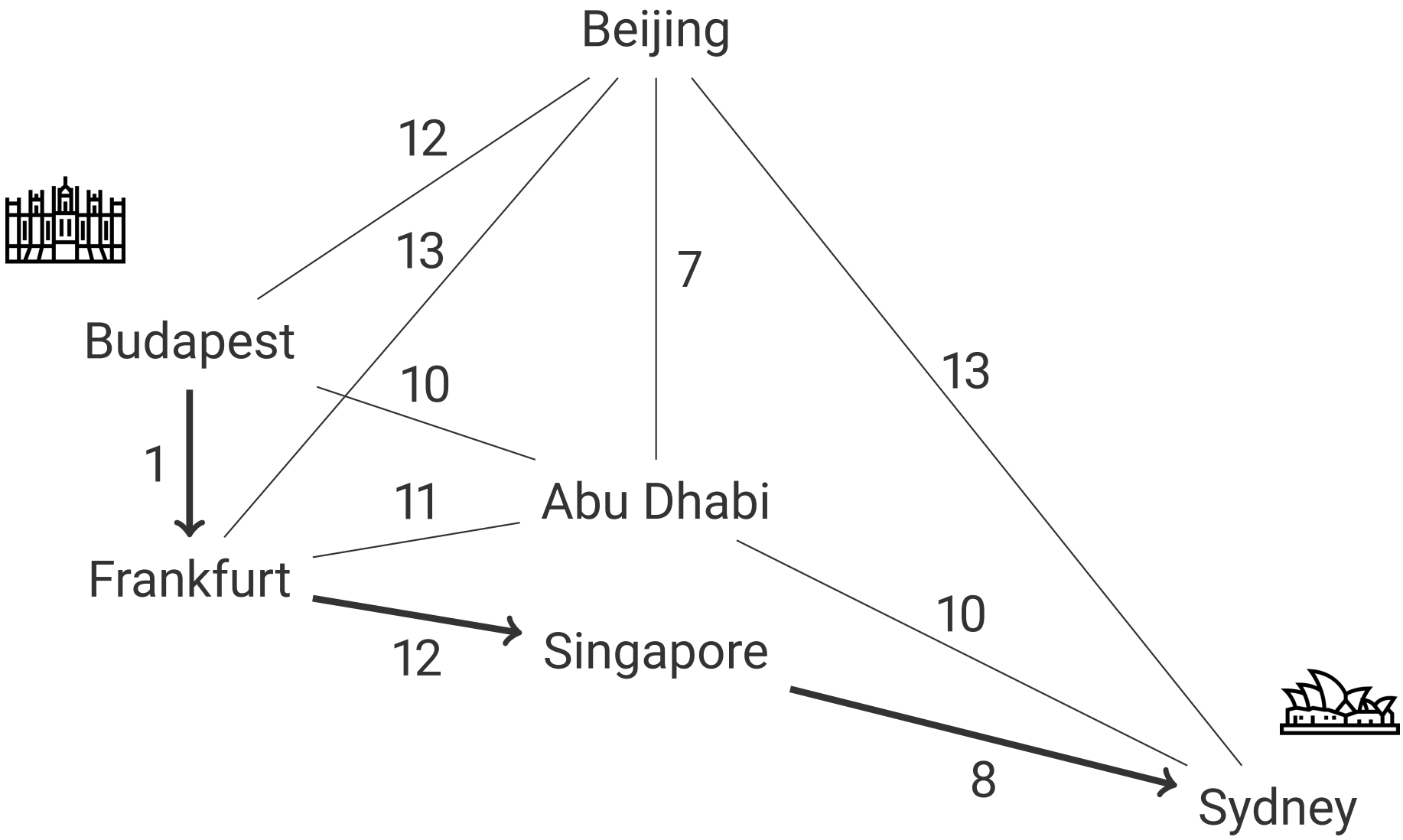
- May be familiar as **Exp3**

$$\mathfrak{R}_n \leq \sqrt{2nk \log(k)}$$

Combinatorial semibandits

- Adversary secretly chooses $y_1, \ldots, y_n$
- $y_t \in [0, 1]^d$
- Learner chooses $A_t \in \mathcal{A} \subseteq \{a \in \{0, 1\}^d : \|a\|_1 = m\}$
- Learner observes $A_{ti} y_{ti}$ for all $i \in \{1, \ldots, d\}$
- Contrast with **bandit** where you observe $\langle A_t, y_t \rangle$

Example (SHORTEST PATH PROBLEMS)



Follow-the-perturbed-leader

- Estimate y_{ti} by $\hat{y}_{ti} = \dfrac{A_{ti} y_{ti}}{\mathbb{P}_t(A_{ti} = 1)}$
- Again $\mathbb{E}_t[\hat{y}_{ti}] = y_{ti}$ provided $\mathbb{P}_t(A_{ti} = 1) > 0$
- Algorithm samples perturbation $\xi_t \in \mathbb{R}^d$
- Chooses $A_t = \operatorname{argmax}_a \left\langle a, \xi_t - \eta \sum_{s=1}^{t-1} \hat{y}_s \right\rangle$
- Computation depends on a linear program and ...

$$\mathfrak{R}_n(a) = \mathbb{E}\left[\sum_{t=1}^{n} \langle a - A_t, y_t \rangle\right]$$

Efficiency

$$A_t = \text{argmax}_a \left\langle a, \xi_t - \eta \sum_{s=1}^{t-1} \hat{y}_s \right\rangle \quad \hat{y}_{ti} = \frac{A_{ti} y_{ti}}{\mathbb{P}_t(A_{ti} = 1)}$$

- $1/\mathbb{P}_t(A_{ti} = 1) = ?$
- Do we need this? No!
- Let $X \in \{1, 2, \ldots, \}$, $X \sim \text{Geo}(\theta)$:

$$\mathbb{P}(X = k) = (1 - \theta)^{k-1} \theta^k$$

  Number of draws from $\text{Ber}(\theta)$ to get first success
- Then $\mathbb{E}[X] = 1/\theta$
- $K_{ti} \sim \text{Geo}(\mathbb{P}_t(A_{ti} = 1))$, $\hat{y}_{ti} = \min(\beta, K_{ti} A_{ti} y_{ti})$

Theoretical results

- **Laplace distribution** $p(x) = 2^{-d} \exp(-\|x\|_1)$
- Choosing η appropriately and ξ_t to be Laplace

$$\mathfrak{R}_n = \tilde{O}\left(m\sqrt{dn}\right)$$

- Exponential weights enjoys $\mathfrak{R}_n = \tilde{O}\left(\sqrt{mdn}\right)$
- Requires sampling from

$$P_{ta} = \frac{\exp\left(-\eta \sum_{s=1}^{t-1} \langle a, \hat{y}_t \rangle\right)}{\sum_b \exp\left(-\eta \sum_{s=1}^{t-1} \langle a, \hat{y}_t \rangle\right)}$$

- No obvious efficient algorithm

Analysing follow-the-perturbed-leader

A quick refresh on mirror descent

- Adversary chooses $y_1, \ldots, y_n$
- $y_t \in \mathbb{R}^d$
- Action set $\mathcal{A} \subset \mathbb{R}^d$ is convex
- Learner chooses Legendre function $F : \mathbb{R}^d \to \bar{\mathbb{R}}$ and $a_1 = \operatorname{argmin}_{a \in \mathcal{A}} F(a)$

$$a_{t+1} = \operatorname{argmin}_{a \in \mathcal{A}} \langle a, y_t \rangle + D_F(a, a_t)$$

- D_F is the **Bregman divergence**

$$D_F(a, b) = F(a) - F(b) - \langle \nabla F(b), a - b \rangle$$

Convexity, Fenchel duality

- Extended reals: $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty, -\infty\}$
- $\mathrm{dom}(f) = \{x \,:\, f(x) < +\infty\}$
- f convex if its epi-graph is convex
- f convex fn is proper if $\mathrm{dim}(f) \neq \emptyset$ and $f > -\infty$
- From now on by convex we mean proper convex
- The **Fenchel dual** of a function f is $f^*(u) = \sup_{x \in \mathrm{dom}(f)} \langle x, u \rangle - f(x)$
- Convex because the maximum of convex functions is convex; $(f^*)^* = f$ if $\mathrm{epi}(f)$ is closed.
- Also called the convex conjugate

Legendre functions

- f convex, $A = \mathrm{dom}(f)$, $C = \mathrm{int}(A)$.
- We call f Legendre if
  - f is strictly convex on C
  - C is nonempty
  - f is differentiable on C
  - $\|\nabla f(x_n)\| \to \infty$ for any sequence $(x_n)_n \subset C$, $x_n \to \partial C$, $n \to \infty$
- For f Legendre:
  - ∇f is a bijection between $\mathrm{int}(\mathrm{dom}(f))$ and $\mathrm{int}(\mathrm{dom}(f^*))$ with the inverse $(\nabla f)^{-1} = \nabla f^*$
  - $D_f(x, y) = D_{f^*}(\nabla f(y), \nabla f(x))$ for all $x, y \in \mathrm{int}(\mathrm{dom}(f))$
  - $\lim_{\alpha \to 1} \langle \nabla f((1-\alpha)x + \alpha y), y - x \rangle = \infty$, $x \in C, y \in \partial C$.

Examples

- $\mathcal{A} = \mathbb{R}^d$ and **quadratic potential** $F(x) = \frac{1}{2}\|x\|_2^2$

$$a_t = -\eta \sum_{s=1}^{t-1} y_s$$

- $\iff$ **gradient descent** with loss $\ell_t(x) = \langle x, y_t \rangle$

Examples

- $\mathcal{A} = \mathbb{R}^d$ and **quadratic potential** $F(x) = \frac{1}{2}\|x\|_2^2$

$$a_t = -\eta \sum_{s=1}^{t-1} y_s$$

- $\iff$ **gradient descent** with loss $\ell_t(x) = \langle x, y_t \rangle$

- **Simplex** action-set and **negentropy potential**

$$\mathcal{A} = \{x \geq 0 : \|x\|_1 = 1\} \quad F(x) = \sum_i x_i \log(x_i) - x_i$$

$$a_{ti} = \frac{\exp(-\eta \sum_{s=1}^{t-1} y_{si})}{\sum_j \exp(-\eta \sum_{s=1}^{t-1} y_{sj})}$$

Mirror descent regret bound

- Relative regret is $\mathfrak{R}_n(a) = \sum_{t=1}^n \langle a_t - a, y_t \rangle$
- Assume $\mathrm{dom}(F) \subset \mathrm{int}(\mathcal{A})$. Then,

$$\mathfrak{R}_n(a) \leq \frac{1}{\eta} \left(F(a) - F(a_1) + \sum_{t=1}^n D_F(a_t, a_{t+1}) \right)$$

- **Proof** Algebra with the Bregman divergence
- $\mathrm{dom}(F) \subset \mathrm{int}(\mathcal{A})$ also implies $\nabla F(a_{t+1}) = -\eta \sum_{s=1}^t y_s$, or $a_{t+1} = \nabla F^*(-\eta \sum_{s=1}^t y_s)$
- FTL=MD: $a_{t+1} = \mathrm{argmin}_{a \in \mathcal{A}} (\eta \langle a, \sum_{s=1}^t y_s \rangle + F(a))$

Mirror descent for bandits

- Replace y_t with $\hat{y}_t$
- Sample A_t from a distribution with mean $\bar{A}_t$

$$\begin{aligned}
\mathfrak{R}_n(a) &= \mathbb{E}\left[\sum_{t=1}^{n} \langle a - A_t, y_t \rangle\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{n} \langle a - \bar{A}_t, \hat{y}_t \rangle\right] \\
&\leq \mathbb{E}\left[\frac{1}{\eta}\left(F(\bar{A}_n) - F(\bar{A}_1) + \sum_{t=1}^{n} D_F(\bar{A}_t, \bar{A}_{t+1})\right)\right]
\end{aligned}$$

FTPL as mirror descent

FTPL

$$\nabla F^*(-\eta L) = a_{t+1} = \mathbb{E}[\operatorname{argmax}_a \langle a, \xi - \eta L \rangle]$$

mirror descent

- $F^* = ?$
- The **support function** of $\mathcal{K}$ is $\phi_{\mathcal{K}}(u) = \sup_{a \in \mathcal{K}} \langle a, u \rangle$
- $\nabla \phi_{\mathcal{K}}(u) = \operatorname{argmax}_{a \in \mathcal{K}} \langle a, u \rangle$ for Q smooth
- Exchanging the expectation and derivative,

$$F^*(z) = \mathbb{E}[\phi_{\mathcal{A}}(\xi + z)]$$

- Why does $\operatorname{dom}(F) \subset \operatorname{int}(\mathcal{A})$ hold? $a_{t+1} \in \operatorname{int}(\mathcal{A})$ when Q has unbounded support (e.g., Laplace)

Proof insights

$$\mathfrak{R}_n \leq \mathbb{E}\left[\frac{1}{\eta}\left(F(\bar{A}_n) - F(\bar{A}_1) + \sum_{t=1}^{n} D_F(\bar{A}_t, \bar{A}_{t+1})\right)\right].$$

- $\max_{a,b} F(a) - F(b) \leq m(1 + \log(d))$ from the properties of Laplace (non-trivial)
- With $\beta = 1/(m\eta)$, we have

$$\begin{aligned} D_F(\bar{A}_t, \bar{A}_{t+1}) &= D_{F^*}(\nabla F(\bar{A}_{t+1}), \nabla F(\bar{A}_t)) \\ &= D_{F^*}(-\eta L - \eta \hat{y}_t, -\eta L) = \frac{\eta^2}{2} \|\hat{y}_t\|^2_{\nabla^2 F^*(\xi)} \leq \frac{ed\eta}{2} \end{aligned}$$

for some $\alpha \in [0, 1]$ and $\xi = -\eta \hat{L}_{t-1} - \alpha\eta \hat{Y}_t$.

Open questions

- Can the regret bound for FTPL be improved?
- Lower bound for many perturbation noise models for $m = d$ is $d^{5/4}\sqrt{n}$ lower bound; tight?
- Hardness separation for combinatorial semibandits?

Conclusion

FTPL may not be regret-optimal,

but it often enjoys reasonable regret guarantees,

while it leads to efficient algorithms.

Questions

Bandit Algorithms

Tor & Csaba

http://banditalgs.com

References