Homework 1

CSE 599m: Interactive Learning
Instructor: Kevin Jamieson

Problems 1-3 are due by the end of Week 6 to Gradescope. You only need to complete exercise 4 *or* 5 (not both) and this must be turned in by the end of Week 10. That is, you can turn in the whole assignment by week 6, or just turn in 1-3 by week 6 and do the experimental problems later and turn them in week 10.

Part I: Mirror Descent and Follow the Regularized Leader. Due end of Week 6

1. Problem 26.4 of [SzepesvariLattimore].

2. Problem 28.4 of [SzepesvariLattimore]. For part b, consider both FTRL and mirror descent. State the explicit update rules for both algorithms, and comment on when they are different. No need to actually "implement" them experimentally unless you are uncertain about how they will behave.

3. Problem 28.15 of [SzepesvariLattimore], excluding part h (just read that part).

Part II: Experiments. Due end of Week 10

Choose **ONE** of the following exercises $\{4, 5\}$ to implement. You may leave one of them blank on your homework and receive full credit.

4. Implement the following algorithms: FTRL with Tsalis-INF (Exercise 28.15), EXP3 (Section 11.3), EXP3-IX (Section 12.1), Follow-the-Leader[1], UCB (Section 7.1), and Thompson Sampling (Section 36.1, use Beta(1,1) as prior distributions). Note that we're looking at *losses* so **lower is better**–modify the algorithms as necessary. Also take note that losses are bounded in $[0, 1]$. Include plots and comment on the following:

- Run all algorithms on a standard stochastic bandit with $P_i = \text{Bernoulli}(\mu_i)$ in $\{0, 1\}$, $n = 10$ arms, $\mu_i = 1/4$ for $i = 1$, $\mu_i = 3/4$ for $i \neq 1$. Use time horizon $T = 10000$.
- Run all algorithms on a standard stochastic bandit with $P_i = \text{Bernoulli}(\mu_i)$ in $\{0, 1\}$, $n = 10$ arms, $\mu_i = 1/4 + \frac{1}{2}\sqrt{(i-1)/(n-1)}$. Use time horizon $T = 10000$.
- With $T = 1000$, $n = 10$, find a set of losses such that UCB suffers roughly linear regret while EXP3 and FTRL both suffer sub-linear regret. The losses should be chosen in an oblivious fashion—they may rely on knowledge of the algorithms but cannot adapt to actions played by the algorithm (i.e. the losses should be chosen in advance).
- With $T = 1000$, $n = 10$, find an example where FTL suffers linear regret while FTRL suffers sub-linear regret. As above the losses should be chosen in an oblivious fashion.

5. Compare the performance of EXP3(γ) applied to linear bandits (Section 27.1) to the algorithms designed for the stochastic setting: LinUCB (Section 19.2) and Action Elimination (Section 22.0). Fix $\theta^* = \mathbf{e}_1$ and draw arms from a Gaussian distribution. Recall that the EXP3(γ) algorithm is as follows:

```
Input: $T$, $n$ arms, $\eta > 0$, $\gamma \in [0, 1]$, exploration distribution $\lambda$
Init: $p_1 = (1/n, \ldots, 1/n)$, Adversary chooses $\{y_t\}_{t=1}^T \in [-h, h]^n$
for $t = 1, 2, ..., T$:
    • Player draws $I_t \sim (1-\gamma)p_t + \gamma\lambda$
    • Player suffers loss $y_{t,I_t}$
    • Player computes $\hat{y}_{t,i}$ satisfying $\mathbb{E}[\hat{y}_{t,i}|p_t] = y_{t,i}$
    • Update $\tilde{p}_{t+1,i} = \exp(-\eta \sum_{s=1}^t \hat{y}_{s,i})$, $p_{t+1,i} = \tilde{p}_{t+1,i}/\|\tilde{p}_{t+1}\|_1$
```

[1] Start by pulling each arm once. At each following time, compute the empirical means so far and pull the arm with the best empirical mean so far.

Implement this algorithm with both the original action set, and the action set augmented to map stochastic bandits to the adversarial setting (see Section 29.2 of [SzepesvariLattimore]). If $\mathcal{X}$ is our original action set and $x \in \mathcal{X}$ an arm, the augmented action set is defined as:

$$\mathcal{X}_{aug} = \{[x, 1] \; : \; x \in \mathcal{X}\}$$

Note that the theoretical regret bounds we proved for EXP3(γ) only hold in the stochastic setting when we use the augmented action set.

Experiment with the choices of d and n. Can you find values of d and n for which EXP3(γ) with augmented actions significantly outperforms LinUCB? Provide plots comparing the performance of each of these algorithms in settings of n and d where:

- LinUCB outperforms Action Elimination and EXP3(γ) with augmented actions.
- EXP3(γ) with augmented actions outperforms both, and LinUCB outperforms Action Elimination.
- EXP3(γ) with augmented actions outperforms both, and Action Elimination outperforms LinUCB

(Hint: how do the regret bounds we proved for each of these algorithms scale with d and n? You may need to use a different learning rate for EXP3(γ) than the theory motivates to attain this performance. All of these objective should be attainable with values of d and n below 200.). Also note that while we proved the correctness of EXP3(γ) for rewards in $[-1, 1]$ is is straightforward to show that it holds for Gaussian as well.