

Instructions:

- There are 5 main questions with a maximum score of 65 points. There is also a 6th (challenging) question that is worth 25 bonus points. The total time allotted is 3 hours.
- No cellphones or electronic aids are allowed. You may use notes made on one A4 size sheet of paper for reference.

1. *Fenchel dual and Bregman divergence* (10 points)

Let $F : (0, +\infty)^d \rightarrow \mathbb{R}$, $F(x) := \sum_{i=1}^d x_i \log x_i - \sum_{i=1}^d x_i$. This is called the generalized negative entropy function.

- Find the Bregman divergence D_F induced by F . (2 points)
- Find the Fenchel dual F^* of F . Mention its domain clearly. (4 points)
- Find the Bregman divergence D_{F^*} induced by F^* . (2 points)
- Show the following ‘three-point inequality’ for the Bregman divergence $D_R(x, y)$ induced by a differentiable convex function $R : \mathbb{R}^d \rightarrow \mathbb{R}$: (2 points)

$$\forall u, v, w \in \mathbb{R}^d : D_R(u, v) + D_R(v, w) = D_R(u, w) + \langle u - v, \nabla R(w) - \nabla R(v) \rangle.$$

2. *Online vs. Classical Optimization* (10 points)

Suppose you have a good online convex optimization algorithm \mathcal{A} with the following property. For any number of rounds $T \geq 1$ and any sequence of T loss functions from \mathcal{L} : a family of convex functions defined over a convex set \mathcal{K} , the algorithm’s regret is at most $f(T)$ in T rounds, with respect to any single point of \mathcal{K} . Assume that $f(T) = o(T)$ is a known sublinear function.

Now consider the standard (or ‘batch’) convex optimization problem: you want to find a minimum of the convex function $f \in \mathcal{L}$ over \mathcal{K} to within an accuracy $\epsilon > 0$. In other words, you must output $x \in \mathcal{K}$ satisfying $f(x) \leq \min_{y \in \mathcal{K}} f(y) + \epsilon$. How you would accomplish this using the algorithm \mathcal{A} ?

[Hint: Run \mathcal{A} with a particular sequence of functions and a particular number of rounds, and output a single point in the end. Jensen’s inequality may be useful.]

3. *Stochastic Bandit Algorithms* (3 × 6 points; +2 points for getting all = 20 points)

Consider the general iid¹ stochastic bandit with N arms and all arms’ rewards being Bernoulli-distributed. If μ_i denotes the expected reward of the i th arm, then the regret of a bandit algorithm, that plays an arm $I_t \in [N]$ at each time $1 \leq t \leq T$ and observes only the random reward from the chosen arm is defined to be $R(T) := T \cdot \max_i \mu_i - \sum_{t=1}^T \mathbb{E}[\mu_{I_t}]$.

Explain briefly which of the following algorithms will/will not always achieve sublinear regret with time horizon T ($R(T)$ is sublinear $\Leftrightarrow \lim_{T \rightarrow \infty} \frac{R(T)}{T} = 0$).

- Initialize $s_i := 0 \forall i \in [N]$. At each time $t \leq T$, play $I_t := \arg \max_i s_i$ (break ties in any fixed manner), get (stochastic) reward R_t and update $s_{I_t} \leftarrow s_{I_t} + R_t$.
- Play all arms exactly once. For each arm i , initialize s_i to be its observed reward and $n_i := 1$. At each time $t \leq T$, play $I_t := \arg \max_i s_i/n_i$ (break ties in any fixed manner), get (stochastic) reward R_t and update $s_{I_t} \leftarrow s_{I_t} + R_t$, $n_{I_t} \leftarrow n_{I_t} + 1$.
- Play all arms exactly once. For each arm i , initialize s_i to be its observed reward and $n_i := 1$. At each time $t \leq T$, play $I_t := \arg \max_i \left(s_i/n_i + \sqrt{2 \log t/n_i} \right)$ (break ties in any fixed manner), get (stochastic) reward R_t and update $s_{I_t} \leftarrow s_{I_t} + R_t$, $n_{I_t} \leftarrow n_{I_t} + 1$.

¹independent and identically distributed

- (d) For each arm $i \in [N]$, initialize $u_i = 1, v_i = 1$. At each time $t \leq T$, sample independent random variables $\theta_i(t) \sim \text{Beta}(u_i, v_i)$, and play $I_t := \arg \max_i \theta_i(t)$ (break ties in any fixed manner). Get (stochastic) reward R_t and update $u_{I_t} \leftarrow u_{I_t} + R_t, v_{I_t} \leftarrow v_{I_t} + (1 - R_t)$.
- (e) For each arm $i \in [N]$, initialize $u_i = 1, v_i = 1$. At each time $t \leq T$, for each arm i let $\theta_i(t) = u_i/(u_i + v_i)$, i.e., the expected value of the $\text{Beta}(u_i, v_i)$ distribution. Play $I_t := \arg \max_i \theta_i(t)$ (break ties in any fixed manner). Get (stochastic) reward R_t and update $u_{I_t} \leftarrow u_{I_t} + R_t, v_{I_t} \leftarrow v_{I_t} + (1 - R_t)$.
- (f) For each arm $i \in [N]$, initialize $p_i := 1/N$. At each time $t \leq T$, play a random arm I_t drawn independently at that time according to the probability distribution (p_1, \dots, p_N) . Get (stochastic reward) R_t , let $\tilde{p}_i := p_i \exp\left(\frac{R_t \mathbb{1}[I_t=i]}{p_i \sqrt{T}}\right)$ for each $i \in [N]$, and update $p_i \leftarrow \tilde{p}_i / \sum_{j \in [N]} \tilde{p}_j$ for all arms i . (Note: $\mathbb{1}[A]$ is the indicator random variable of event A .)

4. ‘Worst-case’ UCB regret (10 points)

Consider the same Bernoulli stochastic N -armed bandit setting as in Question 3. We proved in class the following bound for the Upper Confidence Bound (UCB) algorithm:

$$\mathbb{E}[T_i(T)] \leq \frac{8 \log T}{\Delta_i^2} + \frac{\pi^2}{3}, \quad (1)$$

for any suboptimal arm i (i.e., $\mu_i < \max_j \mu_j$), with $T_i(T)$ denoting the total number of times that arm i was chosen up to time T , and $\Delta_i := \max_j \mu_j - \mu_i$ denoting its gap from an optimal arm. This directly gave us the regret bound $R(T) := \sum_{i=1}^N \Delta_i \mathbb{E}[T_i(T)] \leq 8 \sum_{i: \Delta_i > 0} \frac{\log T}{\Delta_i} + \frac{N\pi^2}{3}$. The bound, however, is of no use when Δ_i is very small as the regret can never be more than T .

Show instead that the following *gap-independent* (or problem-independent, or ‘worst-case’) regret bound for UCB holds.

$$R(T) \leq \sqrt{NT \left(8 \log T + \frac{\pi^2}{3}\right)} \quad \forall N, T.$$

[Hint: Bound the regret using a well-known inequality, use the bound (1), and note that the $T_i(T)$ sum to T across all i .]

5. *Sequential probability estimation* (15 points)

Suppose you are observing an arbitrary stream of bits y_1, y_2, \dots with $y_i \in \{0, 1\}$, generated from some source (e.g., a digital voice signal). The following occurs at each round $t \geq 1$: You are asked to guess a probability distribution $\hat{p}_t \equiv (\hat{p}_t(0), \hat{p}_t(1)) \in \{(p, 1-p) : 0 \leq p \leq 1\}$ for the next bit y_t . Following your guess, y_t is revealed and you suffer a loss of $\log \frac{1}{\hat{p}_t(y_t)}$.

Consider competing in this game with the class of all ‘constant experts’. A constant expert is a rule parameterized by $p \in [0, 1]$ that always guesses the probability distribution $(p, 1-p)$ (this is the analog of Constantly Rebalancing Portfolios in finance).

Write down the natural version of the Exponential Weights prediction algorithm with uniform initial weights and learning rate $\eta = 1$ [Note: There are infinitely many experts!]. Can you express its prediction at each time t in the *simplest* possible form²? You may use the identity

$$\int_0^1 q^{n_1} (1-q)^{n_2} dq = \frac{1}{(n_1 + n_2 + 1) \binom{n_1 + n_2}{n_1}},$$

for any integers $n_1, n_2 \geq 0$, and where $\binom{a}{b}$ is the standard binomial coefficient $\frac{(a+b)!}{a!b!}$.

6. **★ Bonus question - Sequential probability estimation** (25 points)

With regard to the previous question. show that the regret of the Exponential Weights algorithm you wrote down (with $\eta = 1$) in T rounds with respect to all the constant experts, for any sequence of bits y_1, \dots, y_T , is no more than $\log(1 + T)$.

²implementable using finitely many arithmetic operations