Homework 2

CSE 541: Interactive Learning, Autumn 2026
Instructor: Kevin Jamieson

Due Sunday, November 22, 2026, 11:59 PM Pacific (Gradescope)

Instructions. Submit a single typeset PDF to Gradescope (LaTeX strongly recommended; no photos or scans). Put the code for Problems 2 and 6 at the end of the respective problem, after the figures; every plot needs axis labels, a title, and a legend. You may discuss with classmates, but write up your own solutions and code and list your collaborators. Read the course AI policy before you start; on proof questions use AI only minimally, and whatever you use it for, say so in your submission. Total: 100 points.

Notation. $[n] = \{1, \dots, n\}$, $\triangle_n$ is the probability simplex, $\|x\|_A^2 = x^\top A x$, and log is the natural logarithm. A *Gaussian bandit instance* has n arms with means $\theta^* \in \mathbb{R}^n$; pulling arm i returns $X \sim \mathcal{N}(\theta_i^*, 1)$. Arm 1 is best, $\Delta_i = \theta_1^* - \theta_i^*$, $T_i(t)$ is the number of pulls of arm i in the first t rounds, $T_i = T_i(T)$, and the regret is $R_T = \mathbb{E}\big[\sum_{t=1}^T \theta_1^* - \theta_{I_t}^*\big] = \sum_{i=1}^n \Delta_i \, \mathbb{E}[T_i]$. Starter code: `hw2_starter.py` on the course website; use it so that your plots are comparable to ours.

Problem 1: The regret of UCB, and why optimism (15 points)

Consider UCB as stated in lecture: pull each arm once, then for $t = n+1, \dots, T$ play

$$I_t = \arg\max_{i \in [n]} \; \hat{\theta}_i(t-1) + \sqrt{\frac{2\log(2nT^2)}{T_i(t-1)}},$$

where $\hat{\theta}_i(t)$ is the empirical mean of the first $T_i(t)$ pulls of arm i. In lecture we sketched the analysis; here you fill in every step.

(a) (3 points) Let $\hat{\theta}_{i,s}$ denote the empirical mean of the first s pulls of arm i and define the event

$$E \;=\; \bigcap_{i \in [n]} \bigcap_{s \le T} \Big\{ |\hat{\theta}_{i,s} - \theta_i^*| \le \sqrt{\tfrac{2\log(2nT^2)}{s}} \Big\}.$$

Show that $\mathbb{P}(E) \ge 1 - 1/T$.

(b) (4 points) Show that on E, for every $i \neq 1$: $\quad T_i \;\le\; 1 + \dfrac{8\log(2nT^2)}{\Delta_i^2}$.

(c) (3 points) Show that $\mathbb{E}[T_i] \le \dfrac{8\log(2nT^2)}{\Delta_i^2} + 2$, and conclude that when $n \le T$,

$$R_T \;\le\; \sum_{i=2}^{n} \Big(\frac{24\log(2T)}{\Delta_i} + 2\Delta_i \Big).$$

(d) (3 points) The bound in (c) is useless when some Δ_i is tiny. Show that for $n \le T$ there is a universal constant c with $R_T \le c\sqrt{nT \log T}$ for every instance with $\max_i \Delta_i \le 1$. (Hint: for each arm, $\Delta_i \mathbb{E}[T_i] \le \min\{\Delta_i \mathbb{E}[T_i],\ (c)\}$, the worst gap is $\Delta_i \approx \sqrt{\log T / \mathbb{E}[T_i]}$, and $\sum_i \mathbb{E}[T_i] = T$. This is the same argument as for elimination.)

(e) (2 points) Why optimism? Consider the *greedy* algorithm: pull each arm once, then always play $\arg\max_i \hat{\theta}_i(t-1)$ (zero confidence width). With a two-armed Gaussian instance, explain in a few sentences, with a short calculation but no formal proof, why greedy has regret growing linearly in T with constant probability, and what the confidence width in UCB prevents.

Problem 2: UCB, Thompson sampling, explore-then-commit, and EXP3 in practice (15 points)

Implement the four algorithms in `hw2_starter.py`:

- **UCB** exactly as in Problem 1.
- **Thompson sampling (TS)** with prior $\theta_i^* \sim \mathcal{N}(0,1)$ and likelihood $X \sim \mathcal{N}(\theta_i^*, 1)$: at each round sample $\tilde{\theta}_i$ from each arm's posterior, play $\arg\max_i \tilde{\theta}_i$, and update that arm's posterior. (After s pulls of arm i with sum S, the posterior is $\mathcal{N}(S/(s+1),\ 1/(s+1))$.)
- **Explore-then-commit (ETC)** with parameter m: play the arms round-robin until each has m pulls, then commit to the empirical best for the remaining rounds.
- **EXP3** from lecture with uniform exploration, $\eta = \sqrt{\log(n)/(3nT)}$ and $\gamma = \eta n$. EXP3 needs bounded losses: feed it $\ell_t = \mathrm{clip}(-X_t/4, -1, 1)$, which is almost never clipped on these instances and preserves the ordering of the arms.

The starter code averages pseudo-regret $\sum_{s \le t} \Delta_{I_s}$ over 5 seeds and plots all algorithms, with ETC for $m \in \{10, 100, 1000\}$, on two instances:

(a) (6 points) $n = 10$, $\theta_1^* = 0.1$ and $\theta_i^* = 0$ for $i > 1$; (b) (6 points) $n = 40$, $\theta_1^* = 1$ and $\theta_i^* = 1 - 1/\sqrt{i-1}$ for $i > 1$. Use $T = 10^5$ (or larger, if your machine allows). Include both plots.

(c) (3 points) Discuss: which curves look like $\log T$ and which like $\sqrt{T}$, and why? Why does the best m for ETC differ between the two instances? Where does EXP3 sit relative to UCB and TS, and is that what the theory ($\sqrt{nT \log n}$ versus $\sum_i \log T/\Delta_i$) predicts? Which algorithm would you deploy if you did not know whether the environment was stochastic?

Problem 3: Adaptive designs break the fixed-design confidence interval (10 points)

In lecture we showed that for a *fixed* design $x_1, \dots, x_T \in \mathbb{R}^d$, observations $y_t = \langle x_t, \theta^* \rangle + \eta_t$ with $\eta_t \sim \mathcal{N}(0,1)$ i.i.d., and $V_T = \sum_t x_t x_t^\top$ invertible, the least-squares estimate satisfies $\langle z, \hat{\theta}_T - \theta^* \rangle \sim \mathcal{N}(0, \|z\|^2_{V_T^{-1}})$ for every fixed z, so $\mathbb{E}\big[\langle z, \hat{\theta}_T - \theta^* \rangle^2 / \|z\|^2_{V_T^{-1}}\big] = 1$. This problem (Exercise 20.2 of Lattimore–Szepesvári) shows that when the x_t are chosen *adaptively*, as every bandit algorithm does, that is false by a factor of d, which is why the confidence bounds for sequential designs pay for d.

Let $\theta^* = 0$, so that $\hat{\theta}_T = V_T^{-1} S_T$ with $S_T = \sum_{t=1}^{T} \eta_t x_t$. Let $T \geq 2d$ and let $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^d$. Exhibit a sequence $x_1, \ldots, x_T$ in which x_t may depend on $(x_1, \eta_1, \ldots, x_{t-1}, \eta_{t-1})$, with V_T invertible almost surely, such that

$$\mathbb{E}\left[\frac{\langle \mathbf{1}, \hat{\theta}_T \rangle^2}{\|\mathbf{1}\|^2_{V_T^{-1}}}\right] \;\geq\; c\, d$$

for a universal constant $c > 0$. *Hint:* take $x_1, \ldots, x_d = e_1, \ldots, e_d$, look at the signs of $\eta_1, \ldots, \eta_d$, and then spend the remaining pulls on basis vectors chosen to push $\langle \mathbf{1}, \hat{\theta}_T \rangle$ away from 0. Only basis vectors are needed, so V_T is diagonal and everything is explicit.

Problem 4: Pure exploration in linear bandits is not about pulling good arms (10 points)

Pure-exploration linear bandits: a finite set $\mathcal{X} \subset \mathbb{R}^d$, $\delta \in (0, 1)$; at each round the player chooses $x_t \in \mathcal{X}$ and observes $y_t = \langle \theta^*, x_t \rangle + \eta_t$, $\eta_t \sim \mathcal{N}(0, 1)$; the player may stop at any time and must output a single $\hat{x} \in \mathcal{X}$. Assume a unique best arm $x^\star$ and write $\Delta_x = \langle x^\star - x, \theta^* \rangle$. The following is **given**: there is an algorithm that outputs $\hat{x} = x^\star$ with probability $\geq 1 - \delta$ after a number of pulls that scales like $\rho^\star \log(1/\delta)$ up to constants and log factors, where

$$\rho^\star \;=\; \inf_{\lambda \in \triangle_{\mathcal{X}}} \max_{x \neq x^\star} \frac{\|x^\star - x\|^2_{A(\lambda)^{-1}}}{\Delta_x^2}, \qquad A(\lambda) = \sum_{x \in \mathcal{X}} \lambda_x x x^\top,$$

and no algorithm can do better than $\rho^\star \log(1/\delta)$ up to constants (Fiez, Jain, Jamieson, Ratliff 2019; notes §11.5). The intuition: to certify that $x^\star$ beats x you must estimate $\langle x^\star - x, \theta^* \rangle$ to accuracy Δ_x, the variance of that estimate under design λ with τ pulls is $\|x^\star - x\|^2_{A(\lambda)^{-1}} / \tau$, and $\rho^\star$ is the best achievable ratio.

The instance (Soare, Lazaric & Munos 2014). Let $d \geq 2$, $0 < \alpha < \pi/4$ small, and

$$\mathcal{X} = \{e_1, e_2, \ldots, e_d,\; x'\}, \qquad x' = (\cos\alpha, \sin\alpha, 0, \ldots, 0), \qquad \theta^* = e_1.$$

(a) (2 points) Compute $x^\star$ and every gap Δ_x. Which arm is the hard one to rule out, and which *direction* $y = x^\star - x$ has to be estimated accurately to do so? Note that the hard arm and the direction you need are very different vectors.

(b) (4 points) Show that $\rho^\star \asymp \alpha^{-2}$ (constants do not matter, but be explicit about both directions). *Upper bound:* exhibit a design λ and compute $\max_x \|x^\star - x\|^2_{A(\lambda)^{-1}} / \Delta_x^2$ for it; a design supported on two arms suffices. *Lower bound:* for any design and any unit vector u, $\|y\|^2_{A(\lambda)^{-1}} \geq \langle y, u \rangle^2 / \|u\|^2_{A(\lambda)}$ (Cauchy–Schwarz applied to $\langle A^{-1/2} y, A^{1/2} u \rangle$); choose u well.

(c) (2 points) Now suppose an algorithm only ever pulls arms from $\{e_1, x'\}$, the two arms that are close to optimal. (After a few pulls, any reasonable algorithm knows $e_2, \ldots, e_d$ are terrible; elimination drops them, and an optimistic algorithm never plays them again.) Show that for every design supported on $\{e_1, x'\}$, $\|x^\star - x'\|^2_{A(\lambda)^{-1}} = 1/\lambda_{e_1} + 1/\lambda_{x'}$, and conclude that such an algorithm needs $\gtrsim \alpha^{-4} \log(1/\delta)$ pulls to identify $x^\star$.

(d) (2 points) Think about LinUCB on this instance: at each round it plays $\arg\max_{x \in \mathcal{X}} \langle x, \hat{\theta}_t \rangle + \beta_t \|x\|_{V_t^{-1}}$. Describe, informally but concretely, which arms it plays after the first few rounds and why, and what its sample complexity for identifying $x^\star$ therefore is. What does it fail to understand about this instance, and what would a good pure-exploration algorithm do instead?

Problem 5: Non-parametric (Lipschitz) bandits (15 points)

Let $\mathcal{F}_{\text{Lip}}$ be the set of functions $f : [0,1] \to [0,1]$ with $|f(x)-f(y)| \le L|x-y|$ for all x, y, for a known $L > 0$. At each round t the player chooses $x_t \in [0,1]$ and observes $y_t \in [0,1]$ with $\mathbb{E}[y_t] = f^*(x_t)$ for an unknown $f^* \in \mathcal{F}_{\text{Lip}}$. Define $R_T = \mathbb{E}\big[\sum_{t=1}^{T} f^*(x^\star) - f^*(x_t)\big]$ where $x^\star = \arg\max_{x\in[0,1]} f^*(x)$.

(a) (8 points) Propose an algorithm, which may use the horizon T, with $R_T \le O(T^{2/3})$ regret (constants and log factors may be ignored, but say how they depend on L), and prove the bound using a regret guarantee from lecture as a black box.

(b) (7 points) Argue that $T^{2/3}$ is minimax optimal: build an explicit family of hard instances (a picture of the functions is encouraged), explain in the style of the lower-bound lectures why no algorithm can have regret $o(T^{2/3})$ on all of them, with the key calculation, but no formal proof.

Problem 6: Experiments (35 points)

Use `hw2_starter.py`; it contains the data-generating code, the plotting code, and the function signatures below.

6.1 G-optimal design (10 points)

Fix $x_1, \dots, x_n \in \mathbb{R}^d$ and for $\lambda \in \triangle_n$ let $A(\lambda) = \sum_i \lambda_i x_i x_i^\top$, $f(\lambda) = \max_i \|x_i\|^2_{A(\lambda)^{-1}}$ (the G criterion) and $g(\lambda) = -\log\det A(\lambda)$ (the D criterion). Kiefer–Wolfowitz says $\min_\lambda f(\lambda) = d$ and that the minimizers of f and g coincide. Implement `g_optimal(X, N)` using **one** of the following strategies and **one** $h \in \{f, g\}$, with N iterations:

1. **Greedy.** For $i = 1, \dots, 2d$ pick I_i uniformly at random from $[n]$. For $t = 2d+1, \dots, N$ pick $I_t = \arg\min_{k\in[n]} h\big(x_k x_k^\top + \sum_{j<t} x_{I_j} x_{I_j}^\top\big)$. Return $\hat\lambda = \frac{1}{N}\sum_{t\le N} e_{I_t}$.

2. **Frank–Wolfe.** Initialize $\lambda^{(2d)}$ as the empirical distribution of $2d$ uniformly random indices. For $k = 2d+1, \dots, N$ let $I_k = \arg\min_{j\in[n]} \partial h(\lambda)/\partial\lambda_j\big|_{\lambda=\lambda^{(k-1)}}$ and $\lambda^{(k)} = (1-\eta_k)\lambda^{(k-1)} + \eta_k e_{I_k}$ with $\eta_k = 2/(k+1)$. Return $\hat\lambda = \lambda^{(N)}$. You will need the partial derivatives of h; recall $\partial_{\lambda_i} A(\lambda)^{-1} = -A(\lambda)^{-1} x_i x_i^\top A(\lambda)^{-1}$ and $\nabla_A \log\det A = A^{-\top}$.

State which strategy and which h you chose and write out the update you implemented. Then run the provided experiment: $d = 10$, $N = 1000$, $n \in \{10 + 2^i\}_{i=1}^{10}$, points drawn either as standard Gaussians or as heavy-tailed Student-t with 2 degrees of freedom. The plot shows $f(\hat\lambda)$ for your design and $f(\lambda)$ for the uniform allocation $\lambda = \mathbf{1}/n$ as n grows. Comment: why does the uniform allocation get worse as n grows while yours does not, why does the tail of the distribution matter for one and not the other, and how close to the Kiefer–Wolfowitz value d did you get in $N = 1000$ iterations?

6.2 Experimental design for function estimation (10 points)

Let $f : [0,1] \to \mathbb{R}$ be unknown and fix locations $X = \{x_i\}_{i=1}^n$. We want an estimate $\hat f$ that is uniformly good over X, i.e. $\max_{x\in X} \mathbb{E}[(\hat f(x) - f(x))^2]$ small, from $\tau = 1000$ noisy measurements $Y = f(x) + \eta$, $\eta \sim \mathcal{N}(0,1)$, at locations in X (repeats allowed) that must all be chosen before any observation is revealed. This problem emphasizes that the linear setting of the course encodes very rich function classes once you choose the features.

Linear functions. Suppose first that $f(x) = \langle \phi(x), \theta^* \rangle$ for a known feature map $\phi : X \to \mathbb{R}^d$ and unknown θ^*. Propose a strategy for choosing the τ measurement locations and the estimator $\hat{f}$. What can you guarantee about $\max_{x \in X} \mathbb{E}[(\hat{f}(x) - f(x))^2]$? About the random quantity $\max_{x \in X} (\hat{f}(x) - f(x))^2$?

Sobolev functions. Now let f be any absolutely continuous function on $[0, 1]$ with $\int_0^1 |f'(x)|^2 dx < \infty$ (the first-order Sobolev space). Remarkably this is the reproducing kernel Hilbert space of the kernel $k(x, x') = 1 + \min\{x, x'\}$ (Wainwright, *High-dimensional statistics*, Ch. 12). What matters for us: if $K_{ij} = k(x_i, x_j)$ has eigendecomposition $K = \sum_{j=1}^{n} e_j v_j v_j^\top$ with $e_1 \geq e_2 \geq \dots$, then with $\phi(x_i) := (v_{1,i}\sqrt{e_1}, \dots, v_{d,i}\sqrt{e_d}) \in \mathbb{R}^d$ we have $f(x_i) \approx \langle \phi(x_i), \theta \rangle$ for some $\theta \in \mathbb{R}^d$, with an error that vanishes as $d \to n$ and is already negligible for $d = 30$ because the e_j decay quickly. The starter code builds X (50 grid points plus 250 points clustered around 0.25), the features $\Phi \in \mathbb{R}^{300 \times 30}$, and the target $f(x) = -x^2 + x\cos(8x) + \sin(15x)$, and provides `sample_and_estimate(Phi, f_star, lam, tau, rng)`, which draws τ locations i.i.d. from a design λ and returns the least-squares fit on all of X together with $X^\top X$. Use your `g_optimal` from 6.1 to produce the three plots in `problem6_2`:

(a) the CDFs over $x \in [0, 1]$ of the uniform allocation and of the G-optimal allocation; comment on the shapes and how they relate to where the x_i are;

(b) f, the G-optimal estimate, and the uniform-allocation estimate;

(c) the absolute errors of the two estimates, the line $\sqrt{d/\tau}$, and the two confidence widths $\|\phi(x)\|_{(X^\top X)^{-1}}$. Comment on how these five curves relate.

6.3 Linear bandits (15 points)

Implement the three linear bandit algorithms below with $X \leftarrow \{\phi(x_i)\}_{i=1}^{300} \subset \mathbb{R}^{30}$ from 6.2 and rewards $y_t = f(x_t) + \eta_t$, $\eta_t \sim \mathcal{N}(0, 1)$. For $T = 40{,}000$, plot the pseudo-regret $R_t = \sum_{s \leq t} \big(\max_x f(x) - f(x_s) \big)$ against t for all three on one set of axes (the starter code also runs elimination with β_k replaced by the constant 2), and comment: which algorithm would you recommend to minimize regret, does the ranking match what the regret bounds suggest, what happens when β_k is replaced by 2 and why, and what is it about elimination, rather than the size of β_k, that makes it slow here?

Elimination (input T; $\tau = 100$, $\delta = 1/T$, $\gamma = 1$, $U = 1$ as a bound on $\|\theta^*\|_2$, $V_0 = \gamma I$, $S_0 = 0$, $\hat{X}_0 = X$): for $k = 1, \dots, \lfloor T/\tau \rfloor$: compute a G-optimal design $\lambda^{(k)}$ over $\hat{X}_k$; draw $x_{(k-1)\tau+1}, \dots, x_{k\tau} \sim \lambda^{(k)}$, pull them and observe y_t; $V_k = V_{k-1} + \sum_t x_t x_t^\top$, $S_k = S_{k-1} + \sum_t x_t y_t$, $\theta_k = V_k^{-1} S_k$; $\beta_k = \sqrt{\gamma}\, U + \sqrt{2\log(1/\delta) + \log(|V_k|/|V_0|)}$; $\hat{x}_k = \arg\max_{x \in \hat{X}_k} \langle x, \theta_k \rangle$; $\hat{X}_{k+1} = \hat{X}_k \setminus \{x : \langle \hat{x}_k - x, \theta_k \rangle > \beta_k \|\hat{x}_k - x\|_{V_k^{-1}}\}$ (strict, so that $\hat{x}_k$ itself survives).

LinUCB (input T; $\delta = 1/T$, $\gamma = 1$, $U = 1$, $V_0 = \gamma I$, $S_0 = 0$): for $t = 0, \dots, T-1$: $\beta_t = \sqrt{\gamma}\, U + \sqrt{2\log(1/\delta) + \log(|V_t|/|V_0|)}$, $\theta_t = V_t^{-1} S_t$, $x_t = \arg\max_{x \in X} \langle x, \theta_t \rangle + \beta_t \|x\|_{V_t^{-1}}$; pull x_t, observe y_t, $V_{t+1} = V_t + x_t x_t^\top$, $S_{t+1} = S_t + x_t y_t$.

Linear Thompson sampling (input T; $\gamma = 1$, $V_0 = \gamma I$, $S_0 = 0$): for $t = 0, \dots, T-1$: $\theta_t = V_t^{-1} S_t$, $\tilde{\theta}_t \sim \mathcal{N}(\theta_t, V_t^{-1})$, $x_t = \arg\max_{x \in X} \langle x, \tilde{\theta}_t \rangle$; pull, observe, update as in LinUCB.