CS 6515/4540 (Fall 2025)

Exam 3

Name: _____

GTID: _____

GT Username: _____

INSTRUCTIONS

1. When the exam starts, write your name (or GT ID or GT Username) on each page.
2. Bringing two pages of notes is permitted.
3. No books, calculators, or other electronic devices permitted.
4. Make your answers clear and concise.
5. ONLY write on the **FRONT** of each sheet. Backs will not be scanned.
6. You may refer to standard algorithms from class without giving full pseudocode.

I am aware of and in accordance with the Academic Honor Code of Georgia Tech and the Georgia Tech Code of Conduct. I will use no person's help on this test. Also, I have read all the instructions on this page.

Signature:

---

1 Short Answer Questions (36 points)

1. True/False: We can use the gradient descent algorithm to solve linear programs in polynomial time. Briefly explain.

   False. As discussed in class, gradient descent computes ϵ-approximation in poly$(1/\epsilon)$ time, whereas we need poly$(\log(1/\epsilon))$ time algo to solve LPs in time polynomial in input size.

2. Provide an example of a continuous function $f : \mathbb{R} \to \mathbb{R}$ that fails to be β-smooth for any constant β.

   Example: $f(x) = x^3$ on $\mathbb{R}$.
   We have $f'(x) = 3x^2$ and $f''(x) = 6x$, which is unbounded as $x \to \infty$ or $x \to -\infty$. For β-smoothness we need $|f''(x)| \leq \beta$ for all x. Since $|6x|$ is unbounded, no finite β works.

3. Consider a zero-sum game between player A (row player) and player B (column player) with A having actions $\{A_1, A_2\}$ and B having actions $\{B_1, B_2\}$. The entries in this table correspond to the rewards of player A (row player):

	B_1	B_2
A_1	3	-2
A_2	4	-1

   If player A has to commit a strategy before player B (i.e., B will choose their strategy/action knowing A's strategy), can player A guarantee an expected reward of at least 0?

   (Hint: No calculations are needed to answer this.)

   No, since the column player will always play B_2, which results in negative reward for the row player.

4. If the max-coverage problem can be solved in polynomial time, then the set cover problem can be solved in polynomial time?

   Yes, since we can solve max-coverage for every size $k \in \{1, \ldots, n\}$ and select the smallest k that covers all the elements.

5. Consider the n dimensional convex body $K = [-1, 1]^n$, i.e., all n coordinates are between -1 and -1. What is the diameter of K?

   $2\sqrt{n}$. This is because the furthest two points are $(-1.-1.\ldots, -1)$ and $(1.1.\ldots, 1)$, which are at distance $\sqrt{4n}$.

6. True/False: For the experts problem, there exists a deterministic algorithm that obtains average regret $O(\frac{\sqrt{n}}{\sqrt{T}})$. Briefly explain (you may assume anything discussed in class).

   No. We saw in class that deterministic algorithms have $\Omega(1)$-average regret as the adversary can always make the algorithm pay cost 1 while the best expert pays average cost $\leq 1/2$.

2 Convex Optimization (20 points)

Consider an α-strongly convex and β-smooth function $f : \mathbb{R}^d \to \mathbb{R}$ with a global minimum at some $x^* \in \mathbb{R}^d$. We run gradient descent with no constraints (no projection) on f beginning from some point x_0, with step size $\eta = \frac{1}{\beta}$. Let $x_k \in \mathbb{R}^d$ be the iterate after k steps:

$$x_{k+1} = x_k - \frac{1}{\beta}\nabla f(x_k).$$

1. (8 points) Using the definition of β-smoothness and step size $\eta = 1/\beta$, show that

$$\frac{1}{2\beta}\|\nabla f(x_k)\|^2 \;\leq\; f(x_k) - f(x_{k+1}).$$

This is exactly the descent lemma proved in lecture (for step size $\eta = 1/\beta$): using β-smoothness, we have

$$f(x_{k+1}) \;\leq\; f(x_k) + \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2}\|x_{k+1} - x_k\|^2.$$

Plugging in $x_{k+1} - x_k = -\frac{1}{\beta}\nabla f(x_k)$ gives

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{\beta}\|\nabla f(x_k)\|^2 + \frac{\beta}{2}\left\|\frac{1}{\beta}\nabla f(x_k)\right\|^2 = f(x_k) - \frac{1}{2\beta}\|\nabla f(x_k)\|^2,$$

which rearranges to the desired inequality.

2. (6 points) For any α-strongly convex function f with global minimum at x^*, it can be shown that

$$\|\nabla f(x_k)\|^2 \;\geq\; 2\alpha(f(x_k) - f(x^*)).$$

Assuming this and part 1, prove that

$$f(x_{k+1}) - f(x^*) \;\leq\; \left(1 - \frac{\alpha}{\beta}\right)(f(x_k) - f(x^*)).$$

Part 1 gives

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{1}{2\beta}\|\nabla f(x_k)\|^2 \;\geq\; \frac{\alpha}{\beta}(f(x_k) - f(x^*)).$$

Rewriting,

$$f(x_{k+1}) - f(x^*) \;\leq\; (f(x_k) - f(x^*)) - \frac{\alpha}{\beta}(f(x_k) - f(x^*)) = \left(1 - \frac{\alpha}{\beta}\right)(f(x_k) - f(x^*)).$$

3. (6 points) Using the results above show that for any $k \geq 0$

$$f(x_k) - f(x^*) \leq \left(1 - \frac{\alpha}{\beta}\right)^k (f(x_0) - f(x^*)),$$

i.e., gradient descent on α-strongly convex and β-smooth functions converges exponentially fast.

From the previous part,

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\alpha}{\beta}\right) (f(x_k) - f(x^*)).$$

Iterating this inequality:

$$f(x_k) - f(x^*) \leq \left(1 - \frac{\alpha}{\beta}\right)^k (f(x_0) - f(x^*)),$$

which shows exponential (linear) convergence.

3 Online Learning (19 points)

Consider online learning on the real line $K = [-1, 1]$. On each round $t = 1, \ldots, T$:

- The learner chooses $x_t \in [-1, 1]$.
- The adversary reveals a convex loss function

$$f_t(x) = (x - a_t)^2 \quad \text{where } a_t \in [-1, 1].$$

- The learner is revealed function $f_t(x)$ and suffers loss $f_t(x_t)$.

Suppose the learner runs Online Gradient Descent with step size η.

1. (6 points) Compute $\nabla f_t(x_t)$ in terms of x_t and a_t.

$\nabla f_t(x_t) = 2(x - a_t)$

2. (7 points) Write the explicit update rule for x_{t+1} before projection onto $[-1, 1]$.

$x_{t+1} = x_t - \eta \cdot 2(x - a_t)$

3. (6 points) What is the largest constant α for which $f_t(x)$ is α-strongly convex?

We know the second derivative of f_t equals 2. Hence, this is $\alpha = 2$ strongly convex.

4 Complexity (25 points)

In the *subset-query* problem, we are given n query-sets $A_1, \ldots A_n \subseteq \{e_1, \ldots, e_d\}$ and n database-sets $D_1, \ldots D_n \subseteq \{e_1, \ldots, e_d\}$, i.e., all query-sets and database-sets are subsets of d elements $\{e_1, \ldots, e_d\}$. The goal is to determine whether there exist indices i, j such that $A_i \subseteq D_j$.

1. (7 points) Give a simple algorithm for subset-query problem with running time $O(n^2 d)$.

   We just try every pair of query-set i and database-set j. There are n^2 such pairs and for each pair we spend at most $O(d)$ time to check whether $A_i \subseteq D_j$.

Next, we will reduce subset-query problem to the orthogonal vectors problem:

- Define n vectors $\{a_1, \ldots, a_n\}$ in $\{0,1\}^d$ where for $k \in \{1, \ldots, d\}$ we have $a_i(k) = 1$ if query-set A_i contains e_k and $a_i(k) = 0$ otherwise.
- Define n vectors $\{b_1, \ldots, b_n\}$ in $\{0,1\}^d$ where for $k \in \{1, \ldots, d\}$ we have $b_j(k) = 0$ if database-set D_j contains e_k and $b_j(k) = 1$ otherwise.

2. (6 points) Prove that if vectors a_i and b_j are orthogonal then $A_i \subseteq D_j$.

   Since the dot product is zero and every entry is non-negative, we have for every coordinate k that $a_i(k) \cdot b_j(k) = 0$. Thus, for every coordinate k, either $a_i(k) = 0$ (i.e., $e_k \notin A_i$) or $b_j(k) = 0$ (i.e., $e_k \in B_j$), and either-ways $A_i \subseteq D_j$.

3. (6 points) Prove that if vectors a_i and b_j are not orthogonal then $A_i \not\subseteq D_j$, i.e., there exists an element in A_i that is not in D_j.

Since the dot product is non-zero and every entry is non-negative, we have some coordinate k such that $a_i(k) \cdot b_j(k) = 1$. Thus, $a_i(k) = 1$ (i.e., $e_k \in A_i$) and $b_j(k) = 1$ (i.e., $e_k \notin B_j$), which means $A_i \not\subseteq D_j$.

4. (6 pts) Since we reduced subset-query to orthogonal vectors, which problem is at least as hard as the other? Briefly justify.

It means orthogonal vectors is at least as hard as subset-query. This is because the reduction implies that if we have a fast algorithm for orthogonal vectors then there is a fast algorithm for subset-query, which means orthogonal vectors is the harder problem.

END OF EXAM