HOMEWORK 4
Due: Thursday, October 20.

Ground rules: *same as for Homework 1.*

Solve Problems 1–3, and two out of 4–7

---

1. Johnny Minimaximilius. We'll now give a proof of the minimax theorem. For a positive integer k, let $\Delta_k = \{x \in [0,1]^k \mid \sum_i x_i = 1\}$. Now the theorem says that for any matrix $A \in R^{m\times n}$,

$$\max_{x\in\Delta_m} \min_{y\in\Delta_n} x^\top Ay = \min_{y\in\Delta_n} \max_{x\in\Delta_m} x^\top Ay$$

In this problem we will prove this theorem using basic duality.

(a) Show that the minimax theorem is equivalent to saying:

$$\max_{x\in\Delta_m} \min_{j} x^\top Ae_j = \min_{y\in\Delta_n} \max_{i} (e_i)^\top Ay$$

where e_i is the i^{th} standard basis (column) vector.

(b) Show the "easy" direction of the minimax theorem:

$$\max_{x\in\Delta_m} \min_{y\in\Delta_n} x^\top Ay \le \min_{y\in\Delta_n} \max_{x\in\Delta_m} x^\top Ay$$

(c) For any fixed setting of $x \in \Delta_m$, define $c = A^\top x$. Consider the LP $\min\{c^\top y \mid \sum_j y_j = 1, y \ge 0\}$, and let its optimal value be L_x. Write its LP dual. Hence, write an LP whose value equals $\max_{x\in\Delta_m} \min_{y\in\Delta_n} x^\top Ay$. *(Make sure you write a linear program — you have no constraints with products of variables, etc.)*

(d) Similarly, write an LP whose value equals $\min_{y\in\Delta_n} \max_{x\in\Delta_m} x^\top Ay$.

(e) Use the above parts to infer the minimax theorem.

2. Matroids and the Greedy Algorithm. Given a finite universe U of elements, a collection of subsets $\mathcal{I} \subseteq 2^U$ is subset-closed if $B \in \mathcal{I}$ and $A \subseteq B \subseteq U$ implies $A \in \mathcal{I}$: the set system $(U, \mathcal{I})$ is called an *independence system* if $\mathcal{I}$ is subset closed and $\emptyset \in \mathcal{I}$. (In that case we call a set A *independent* if $A \in \mathcal{I}$.) An independence system is called a *matroid* if it satisfies the additional property (*) that if $A, B \in \mathcal{I}$ and $|A| < |B|$ there is an element $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$.

Some more notation: a maximal independent set is called a *base*. A cycle (or circuit) is a minimal *dependent* set. A *cut* is a minimal set that intersects every base. The rank of a matroid is the cardinality of any maximal (and hence *maximum*) independent set in the matroid.

(a) (Don't hand in) Show that the following set systems are matroids: (a) $\mathcal{I}$ is the collection of all subsets of cardinality at most r, (b) $U = U_1 \uplus U_2 \uplus \ldots \uplus U_\ell$ and $\mathcal{I} = \{A \subseteq U \mid \forall i \in [\ell] |A \cap U_i| \leq r_i$, (c) U is the set of edges of some graph G and $\mathcal{I}$ contains all acyclic subsets of edges, (d) $U = \mathbb{F}^r$ for some field $\mathbb{F}$ and $\mathcal{I}$ contains every set of r-bit vectors that are linearly independent over the field $\mathbb{F}$. What are the ranks of these matroids?

(b) (Don't hand in) Verify that the following independence system does not form a matroid in general by giving a counterexample: the elements are the edges of an undirected bipartite graph $G = (V, E)$ and the independent sets are matchings in G (i.e., subsets of edges such that no two edges share a common endpoint).

(c) (Don't hand in) Show that the cardinality of any two bases in a matroid is the same.

(d) Given $M = (U, \mathcal{I})$, define the *dual* matroid M^* to have the same set of elements; the independent sets $\mathcal{I}^*$ are the complements of bases in M, as well as all subsets of these bases. Show that M^* is indeed a matroid.

(e) Given distinct element weights $w : U \to \mathbb{R}$, the min-weight basis problem seeks to find a basis B of minimum weight $w(B) = \sum_{e \in B} w(e)$. Consider the following two coloring rules:

- (red rule/cycle rule) for any cycle C in the matroid that does not have any red elements, color the maximum weight element in this cycle red, and
- (blue rule/cut rule) for any cut X in the matroid that does not have any blue elements, color the minimum weight element in the cut blue.

Show that repeatedly applying cut rule (blue rule) and the cycle rule (red rule) to any matroid in an arbitrary order will color all the elements, and the resulting blue elements form the min-weight basis.

(f) Consider the natural generalization of Kruskal's algorithm: let $S \leftarrow \emptyset$, sort elements by weight and consider them in non-decreasing order, when considering an element e add it to the S if $S \cup \{e\}$ is independent. Show that this algorithm solves the min-weight basis problem. (You may assume an oracle that answers membership queries for $\mathcal{I}$ in constant time.)

(g) Show that for an independence system $(U, \mathcal{I})$, if the greedy algorithm correctly solves the min-weight basis problem for any setting of weights, then the independence system is a matroid. Hence the greedy algorithm characterizes matroids.

(h) (Don't hand in) Given $M = (U, \mathcal{I})$ and a set $S \subseteq U$, define the *restriction* matroid $M[S] = (S, \mathcal{I}[S])$, where $\mathcal{I}[S] = \{A \cap S \mid A \in \mathcal{I}\}$. Show that $M[S]$ is indeed a matroid. Let $r(S)$ denote the rank of $M[S]$; since $M[U] = M$, $r(U)$ is just the rank of the original matroid.

(i) Consider the following LP with variables $\{x_e\}_{e \in U}$:

$$\min\{w^\top x \mid x(S) \leq r(S) \ (\forall S \subset U), \quad x(U) = r(U), \quad x \geq 0\}.$$

Give an alternate proof of the fact that the greedy algorithm outputs an optimal min-weight basis, by giving a feasible solution to its dual whose value equals the greedy solution weight.

3. Perturbing to get started with Ellipsoid. *(Here we use the notation $\langle z \rangle = \text{size}(z)$ for the bit-representation size, defined in Homework 1. You may use any results from Homework 1.)* Suppose you are given a polyhedron $P \subseteq \mathbb{R}^n$ specified as $\{Ax = b, x \geq 0\}$. Give a poly$(\langle A \rangle, \langle b \rangle)$-time algorithm which outputs a polyhedron $P' \subseteq \mathbb{R}^n$ specified as $\{A'x \leq b'\}$, along with a rational $r > 0$, such that:

- If P is empty then P' is empty.
- If P' is not empty then P' contains a ball of radius r.

4. Martin, Laci, and Lex. Assume the following theorem, which we basically proved in class:

> **Theorem.** Assume the Ellipsoid Algorithm is given (i) a poly-time "weak separation oracle" for the convex set $K \subseteq \mathbb{R}^n$, (ii) radius $R \in \mathbb{Q}$ such that $K \subseteq B(0, R)$, and (iii) an error parameter $0 < \epsilon \in \mathbb{Q}$. Then in poly$(n, \langle R \rangle, \langle \epsilon \rangle)$ time, it either outputs a point $s \in K^{+\epsilon}$ or an ellipsoid $E \supseteq K$ with vol$(E) \leq \epsilon$.

Here a "weak separation oracle" for K takes as input a point $y \in \mathbb{Q}^n$, and an error parameter $0 < \delta \in \mathbb{Q}$, and either asserts $y \in K^{+\delta}$ or finds a vector $a \in \mathbb{Q}^n$ with $\|a\|_\infty = 1$ such that $a^\top x \leq a^\top y + \delta$ for all $x \in K^{-\delta}$.

Give a "weak optimization algorithm" which, given the same inputs as the Ellipsoid Algorithm as well as a vector $c \in \mathbb{Q}^n$, runs in time poly$(n, \langle R \rangle, \langle \epsilon \rangle, \langle c \rangle)$ and either correctly asserts that $K^{-\epsilon}$ is empty, or else finds a point $y \in K^{+\epsilon}$ such that $c^\top x \leq c^\top y + \epsilon$ for all $x \in K^{-\epsilon}$.

5. The Highest Compact. In Lecture #7 we saw an LP formulation for the min-cost arborescence problem: $\min\{c^\top x \mid x(\partial v) = 1 \;\; (\forall x \neq r), x(\partial S) \geq 1 \;\; (\forall S \neq \emptyset, r \notin S), x \geq 0\}$. This LP has an exponential number of constraints.

(a) Note that the exponential number of constraints above model the fact that the minimum cut (using the x_e values as the edge capacities) between each non-root vertex v and the root r has capacity at least 1. Use this fact to write an LP with only *poly*(n) variables and constraints whose optimal value equals the optimal value of the LP above.

(b) (Extra Credit.) Either show your new "compact" LP is integral, or show how you can use this new LP to find an integer solution to the min-cost arborescence problem.

6. Violation implies Separation.

(a) Let $K \subset \mathbb{R}^n$ be a closed convex set with $B(0, r) \subseteq K \subseteq B(0, R)$, $0 < r < R$. Define

$$K^* = \{y \in \mathbb{R}^n : y^\top x \leq 1 \; \forall x \in K\}.$$

Show that K^* is convex and that $B(0, 1/R) \subseteq K^* \subseteq B(0, 1/r)$.

(b) Show that $K^{**} = K$. (You may use the fact that for any $x \notin K$, there is a hyperplane strictly separating x from K.)

The "strong violation problem" for a convex set K_1 is: Given $c \in \mathbb{Q}^n$ and $\gamma \in \mathbb{Q}$, assert that $c^\top x \leq \gamma$ for all $x \in K$ or else find $x \in K$ such that $c^\top x > \gamma$. The "strong validity problem" for a convex set K_2 is: Given $c \in \mathbb{Q}^n$ and $\gamma \in \mathbb{Q}$, decide whether or not $c^\top x \leq \gamma$ for all $x \in K_2$. The "strong membership problem" for a convex set K_3 is: Given $y \in \mathbb{Q}^n$, decide whether $y \in K_3$.

(c) Assume a poly-time strong validity algorithm for K. Show a poly-time strong membership algorithm for K^*.

(d) Assume a poly-time strong violation algorithm for K. Show a poly-time strong separation algorithm for K^*.

(e) *Suppose* that one could get a poly-time strong violation algorithm for K from a poly-time membership algorithm for K, whenever one is given r and R with the promise that $B(0, r) \subseteq K \subseteq B(0, R)$.[1] Then show that there is a poly-time reduction from strong validity to strong separation for such K.

7. Where is it? In Problem 3, suppose we are only given a "weak membership oracle" for K (which, given y and δ, asserts either $y \in K^{+\delta}$ or $y \notin K^{-\delta}$). In class we mentioned (without proof) that one can still solve the weak optimization problem in polynomial time if one is also given a "center" $s_0 \in \mathbb{Q}^n$ and a radius $0 < r \in \mathbb{Q}$ with the promise that $B(s_0, r) \subseteq K$. Show that if you're just given r but not s_0, polynomial-time weak optimization is impossible.

[1] Such a reduction does not exist. However, as mentioned in class, there *is* a poly-time reduction from *weak* optimization (and hence *weak* violation) to *weak* membership. Everything in this problem actually works in the "weak" case. But we're trying to keep things simple for you here.