Homework 4
Due: Friday Dec 3, 11:59pm EST on Gradescope

Exercises (Please Solve but Do Not Submit)

1. A Covering Problem. Given a set of n items, each having a size $s_i > 0$ and a cost $c_i \geq 0$, you want to pick a min-cost subset of them so that the total size is at least 1. (You will see this in Problem #1 below as well.) Show that the greedy algorithm, which picks items in increasing order of c_i/s_i does *not* give a constant factor approximation.

Suppose you guess the cost C of the most expensive item in the optimal solution, throw away all items more expensive than C, and now run the greedy algorithm. Show that your cost is now at most $2C$, and hence you have a 2-approximation.

2. Another Covering Problem. Given a set system $(U, \mathcal{F})$ with n elements and m sets, now you are given a coverage requirement r_e for each $e \in U$, and a cost c_S for each $S \in \mathcal{F}$. You want to pick the collection $\mathcal{F}' \subseteq \mathcal{F}$ of smallest total cost such that each element e belongs to r_e different sets. (Each set can be picked at most once into $\mathcal{F}'$.) Write an LP for this problem. Show a randomized rounding algorithm that achieves an $O(\log n)$-approximation.

Problems

Please solve any four out of the five problems.

1. Round or Separate! Given a set of n items, each having a size $s_i > 0$ and a cost $c_i \geq 0$, you want to pick a subset of them of minimum total cost, so that the total size is at least 1.

(a) (Do not submit.) You write the natural LP relaxation for this problem:

$$\min\{\sum_i c_i x_i \mid \sum_i s_i x_i \geq 1, x_i \in [0,1]\}.$$

Show this LP has a large integrality gap: i.e., give an instance (with a *single item*) where the LP value is much smaller than the cost of the optimal integer solution. (Hint: what if $s_i \gg 1$?)

(b) Now suppose you truncate the size of each item as follows: $s_i \leftarrow \min(s_i, 1)$. Show that the LP relaxation above still has an unbounded integrality gap. (Hint: now your example instance can have *two items*, both of size at most 1.)

(c) To fix this problem, you write a stronger LP relaxation. For a set $A \subseteq [n]$ of items, let $S(A) := \sum_{i \in A} s_i$. Define the requirement with respect to A to be $D(A) = \max(1 - S(A), 0)$, and the marginal size of item i with respect to A to be $s_i(A) := \min(s_i, D(A))$. Argue that the constraints

$$\sum_{i \notin A} s_i(A)\, x_i \geq D(A) \qquad \forall A \subseteq [n]$$

are valid constraints—namely that any (integer) solution to the original covering problem satisfies these constraints.

(d) Show that, for the two-item example you constructed above, this LP has cost at least half of the optimal integer solution.

(e) So we add in all these constraints to our LP. Since there are exponentially many such constraints, and separating for them is not easy, here's a way out. Given any fractional solution $x \in [0,1]^n$ for this larger LP, we focus on one constraint (that depends on x). If it is not satisfied, we have found a violated constraint (and use this as a separation oracle for Ellipsoid). OTOH, if this constraint is satisfied, we show how to get an integer solution from this constraint itself!

Given x, define $A := \{i \in [n] \mid x_i \geq 1/2\}$. Consider the constraint $\sum_{i \notin A} s_i(A)\, x_i \geq D(A)$. Suppose this constraint is satisfied (fractionally). Define $X_i = 1$ for all $i \in A$. For each item i in $[n] \setminus A$ independently let $X_i = 1$ with probability $2x_i$.

Show that this solution X is a feasible integer solution with constant probability.

2. Pack Up Your Items in Your Old Kit Bag. Recall the *max-value knapsack* problem we saw in class. We are given items with size $s_i > 0$ and *value* $v_i \geq 0$. We want to pick items whose total size is *at most* S, and to *maximize* the sum of values of items in this set. (This is a packing problem, unlike the covering problem above.)

(a) Suppose all the values are integers. Give a dynamic-programming algorithm with running time $O(nV^*)$, where V^* is the value of the *optimal solution*. So, this would be a better algorithm if the sizes are much larger than the values.

Note: your algorithm should work even if V^* is not known in advance. You may want to first assume you are given V^* up front and then afterwards figure out how to remove that requirement.

(b) Now given an instance I of knapsack and some real $k \geq 1$, define new values $v'_i := k \cdot \left\lfloor \frac{v_i}{k} \right\rfloor$, but retain the old sizes. This gives a new instance I'. Since item sizes and S remain the same, clearly the feasible solutions to I and I' are the same, albeit with different values.

For any feasible solution, let its value in I be V, and its value in I' be V'. Show that $V \geq V' \geq V - nk$.

(c) Use part (a) to show that I' can be solved in at most $O(\frac{n^2 v_{\max}}{k})$ time.

(d) Given any knapsack instance I and a value $\varepsilon \in (0,1)$, show that setting $k := \frac{\varepsilon\, v_{\max}}{n}$ gives an algorithm that returns a feasible solution to I, has value least $(1-\varepsilon)$ times the optimal value of I, and runs in time $O(\frac{n^3}{\varepsilon})$.

In other words, if you wanted to find a solution whose value is within 99% of the optimum value, use this algorithm with $\varepsilon = 0.01$.

3. Constructive Caratheodory. for a graph $G = (V, E)$ with $E = \{e_1, \ldots, e_m$, let the characteristic vector χ_T for a spanning tree T denote an m-bit vector where the i^{th} bit of χ_T is 1 exactly when $e_i \in T$. The spanning tree polytope K is the convex hull of $\{\chi_T \mid T \text{ is a spanning tree of } G\}$. Given a point $x \in K$ as input, our goal is to find scalars $\{\lambda_T\}$ such that $x = \sum_T \lambda_T \chi_T$, where $\lambda_T \geq 0$ and $\sum_T \lambda_T = 1$.

(a) Write this problem as a linear program P with variables λ_T. How many variables does this have? How many constraints, apart from the non-negativity constraints? Use this to infer that x can be written as a convex combination of at most m spanning trees. (This statement is pretty much Carathéodory's Theorem.)

(b) Write down the linear programming dual D of your LP. How many constraints does it have? Show that you can solve this LP optimally in polynomial time using the Ellipsoid method.

(c) Consider the program D' obtained by just taking the subset of polynomially many constraints returned by the separation oracle for the Ellipsoid algorithm. Argue that the optimal objective function value of D' must be the same as that of D.

(d) Consider the dual P' of D'. Observe that any solution to P' also gives a solution to P. Show that solving P' explicitly gives a way to write x as a polynomially-sized convex combination of spanning trees.

All that the above argument uses about the spanning tree polytope is that we can solve the min-cost spanning tree problem in polynomial time. This can be extended to give constructive Carathéodory theorems for other integer polytopes for which we can efficiently find optimal vertex solutions.

4. Sum-of-Squares for Set Cover In this problem, you will design a $m^{O(n^{\varepsilon})}$ time algorithm that takes input m sets $S_1, S_2, \ldots, S_m \subseteq [n]$ with non-negative costs $c_1, c_2, \ldots, c_m$ as input and finds a set cover of cost $((1-\varepsilon)\ln n + O(1))OPT$ where OPT is minimum possible cost of any set cover.

You are allowed to use the following fact that we proved in the lectures without proof: Let $0 \leq x_1, x_2, \ldots, x_m \leq 1$ satisfy $\sum_{S: i \in S} x_i \geq 1$. Then, the greedy algorithm for rounding such x produces a solution of cost at most $H(k)\sum_{j=1}^{m} c_j x_j$ where k is the maximum size of any S_i such that $x_i > 0$ and $H(k) \leq \ln(k) + 1$.

We will analyze the following algorithm. Assume first that the number OPT is known to us. For $t = 2n^{\varepsilon} + 2$, find a pseudo-distribution $\tilde{\zeta}$ on $\{0,1\}^n$ that satisfies the constraints:

$$K = \{\sum_{i: j \in S_i} x_i \geq 1 \forall j \in [n]; \sum_{i=1}^{m} c_i x_i \leq OPT\}$$

In the following parts, you will come up with a rounding algorithm that takes input the first t pseudo-moments of $\tilde{\zeta}$ and outputs a set cover.

(a) (2 points) For any i such that $\tilde{\mathbb{E}}_{\tilde{\zeta}}[x_i] > 0$, define a new pseudo-distribution $\tilde{\zeta}'$ as follows: for any monomial x_S of size $\leq t-2$, let $\tilde{\mathbb{E}}_{\tilde{\zeta}'}[x_S] = \frac{\tilde{\mathbb{E}}_{\tilde{\zeta}}[x_S x_i^2]}{\tilde{\mathbb{E}}_{\tilde{\zeta}}[x_i^2]}$.

Prove that $\tilde{\zeta}'$ is a pseudo-distribution of degree $\geq t-2$ that satisfies the constraints in K and that $\tilde{\mathbb{E}}_{\tilde{\zeta}'}[x_i] = 1$. We will say that $\tilde{\zeta}'$ is the *conditioning of* $\tilde{\zeta}$ on $x_i = 1$.

(b) (2 points) **Rounding:** Our rounding repeatedly modifies the pseudo-distribution $\tilde{\zeta}$. Set $\tilde{\zeta}^{(0)} = \tilde{\zeta}$. For $j = 0, 1, 2, \ldots, (t-2)/2$, find set S_j that covers maximum possible number of elements in $[n] \setminus (S_1 \cup \cdots \cup S_{j-1})$ such that $\tilde{\mathbb{E}}_{\tilde{\zeta}^{(j)}}[x_j] > 0$ and set $\tilde{\zeta}^{(j+1)}$ to be the conditioning of $\tilde{\zeta}^{(j)}$ on $x_j = 1$. If no such S_j exists, stop and output all sets S_k such that $\tilde{\mathbb{E}}_{\tilde{\zeta}^{(a)}}[x_k] > 0$.

Using part (1) above, prove that $\tilde{\zeta}^{((t-2)/2))}$ is a pseudo-distribution of degree 2 satisfying all the constraints in K and $\tilde{\mathbb{E}}_{\tilde{\zeta}^{(t/2-2)}}[x_j] = 1$ for every set S_j chosen in the iterations above.

(c) (3 points) Let $U = [n] \setminus (S_1 \cup S_2 \cup \ldots S_{(t-2)/2})$ be the set of all uncovered elements and let $S_1 \cap U, S_2 \cap U, \ldots, S_m \cap U$ be the *residual set system*. Prove that for every S_r such that $\tilde{\mathbb{E}}_{\tilde{\zeta}^{t-2}}[x_r] > 0$, $|S_r \cap U| \leq n^{1-\varepsilon}$. (Hint: prove that if not, then every set chosen in the iterative conditioning step above covers $\geq n^{\varepsilon}$ new elements. How many such steps can there be?)

(d) (2 points) Consider the rounding algorithm that selects all the $\leq t/2$ sets obtained in the iterative rounding step and rounds the residual set system by the greedy algorithm. Prove that this algorithm constructs a set cover of cost $\leq ((1-\varepsilon)\ln n + O(1))OPT$.

(e) (1 points) Describe how to modify the algorithm above if OPT is not known and all costs $c_1, c_2, \ldots, c_m$ are integers in $[0, 2^n]$.

5. Sum-of-Squares and Squared Triangle Inequality Recall the squared triangle inequality that we used in the analysis of the ARV algorithm for computing the (uniform) sparsest cuts in graphs. In this problem, you will show that every degree 4 pseudo-distribution over the hypercube satisfies the squared triangle inequality. As a consequence, you will "break" the integrality gap of the cycle graph for Max-Cut.

(a) (5 points) Prove that if $\tilde{\zeta}$ is a pseudo-distribution on $\{-1,1\}^n$ of degree at least 6, then, for any i, j, k, $\tilde{\mathbb{E}}_{\tilde{\zeta}}[(x_i - x_j)^2] \leq \tilde{\mathbb{E}}_{\tilde{\zeta}}[(x_i - x_k)^2] + \tilde{\mathbb{E}}_{\tilde{\zeta}}[(x_k - x_j)^2]$. (Hint: Consider the polynomial $(x_i - x_k)^2 + (x_k - x_j)^2 - (x_i - x_j)^2$. Observe that this polynomial is non-negative over ± 1 variables. Use the local distribution property of pseudo-distributions proved in the class.)

(b) (Bonus) Show that the conclusion of part (1) holds even if $\tilde{\zeta}$ is a pseudo-distribution of degree 4 on $\{-1,1\}^n$.

(c) (5 points) For any odd n, consider the graph C_n – the n-cycle on n vertices. Let $f(x) = \frac{1}{4n}\sum_{\{i,j\}\in C_n}(x_i - x_j)^2$ be the polynomial that computes the normalized cut size in C_n. Prove that for any pseudo-distribution $\tilde{\zeta}$ of degree 6 (4, if you proved the bonus), it holds that $\tilde{\mathbb{E}}_{\tilde{\zeta}}[f(x)] \leq 1 - \frac{1}{n}$. Conclude that the relaxation: max $\tilde{\mathbb{E}}_{\tilde{\zeta}}[f(x)]$ over all pseudo-distributions $\tilde{\zeta}$ on $\{-1,1\}^n$ of degree ≥ 6 has no integrality gap on C_n.