ADVANCED ALGORITHM DESIGN

Homework 2

October 27, 2024

Problem 1

Recall the max-flow problem from undergraduate algorithms: for a directed graph $G(V, E)$ with non-negative capacities c_e for every $e \in E$ and two special vertices s (source, with no incoming edges) and t (sink, with no outgoing edges), a flow in G is an assignment $f : E \to R_{\geq 0}$ such that $f(e) \leq c_e$ for every edge e, and for every vertex $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f((u, v)) = \sum_{(v,w)\in E} f((v, w))$. The task is to find a maximum flow f, that is, a flow such that $\sum_{(s,u)\in E} f((s, u))$ is maximized.

(a) Show that the following LP is a valid formulation for computing the value of the maximum flow in G. There is a variable $f((u, v))$ for all $(u, v) \in E$.

$$\begin{aligned} \max_{(f((i,j)))\in\mathbb{R}^E} & \sum_u f((u,t)) \\ \text{s.t. } & f((u,v)) \leq c_{(u,v)} \ \forall\ (u,v) \in E \\ & \sum_{(u,v)\in E} f((u,v)) = \sum_{(v,w)\in E} f((v,w)) \ \forall\ v \in V \setminus \{s,t\} \\ & f(e) \geq 0 \ \forall\ e \in E. \end{aligned} \tag{1}$$

(b) Write the dual for the LP 1. Show that this dual LP computes the minimum fractional $s-t$ cut in G. A fractional cut places each node v at some point y_v on the unit interval $[0, 1]$, with s placed at 0 and t placed at 1. The value of the fractional cut is $\sum_{e=(u,v)\in E(G)} c_e \cdot \max\{0, y_v - y_u\}$ (where c_e is the weight of edge e). Observe that if instead each $y_v \in 0, 1$, that this is simply an $s-t$ cut. Use strong LP duality to conclude the fractional max-flow min-cut theorem. That is, if the max-flow is C, there exists a fractional $s - t$ cut of value C, and no fractional $s - t$ cut of value $< C$.

(c) Devise a rounding scheme that takes as input a fractional min-cut of value C and outputs a true (deterministic) min-cut of value C. (Hint: try correlated randomized rounding — choose a threshold $c \in [0, 1]$ uniformly at random and set $S = \{v \mid y_v \leq c\}$. What can you say about the probability that a given edge is in this (randomized) cut?) Conclude the max-flow min-cut theorem.

Solution

(a) For a valid flow f on G (that is, f is feasible to 1), we denote by $T(v) = T_f(v)$ to be the total flow *through* a vertex $v \in V$. More concretely, $T(v) = \sum_{i \to v} f((i,v)) - \sum_{v \to j} f((v,j))$. So $T(s) = -\sum_{s \to j} f((s,j))$ and $T(t) = \sum_{i \to t} f((i,t))$. f is constrained to be conserved at each vertex $v \in V \setminus \{s,t\}$, so $T(v) = 0 \ \forall \ v \in V \setminus \{s,t\}$. This means $\sum_{v \in V} T(v) = T(s) + T(t)$. However, every outgoing edge e for some vertex is an incoming edge for some other vertex. That is to say, every edge in the above summation occurs positively and negatively exactly once each. So $\sum_{v \in V} T(v) = 0$. This establishes $\sum_v f((s,v)) = \sum_u f((u,t))$ for any valid flow f. This justifies the objective of 1. The constraints simply come from the problem description, namely respectively, that the flow value in each edge is at most the capacity, the conservation law holds at every vertex except s, t, and the flow is non-negative on every edge.

(b) The dual problem will have a dual variable per constraint and will be formed as a minimization problem. Let the dual variables be $x_{(i,j)}$ for each $(i,j) \in E$, and y_i for each $i \in V \setminus \{s,t\}$. The objective is to minimize $\sum_{e \in E} c_e x_e$ with the constraints that $x_e \geq 0 \ \forall \ e \in E, 1 \leq x_{(u,t)} - (-y_u) \ \forall \ (u,t) \in E$ where $u \neq s$, $0 \leq x_{(s,u)} - y_u \ \forall \ (s,u) \in E$ where $u \neq t$, $1 \leq x_{(s,t)}$ if $(s,t) \in E$, $y_v - y_u \leq x_{(u,v)} \ \forall \ (u,v) \in E, v \neq t, u \neq s$.

We justify this as follows. Firstly it is not hard to see that (using the same analysis as done in class) that the following is a primal dual pair (without any duality gap):

$$\begin{aligned} \max_{\boldsymbol{f}} \ & \boldsymbol{a}^\top \boldsymbol{f} \\ \text{s.t. } & A\boldsymbol{f} = 0 \\ & \boldsymbol{f} \geq 0 \\ & \boldsymbol{f} \leq \boldsymbol{c} \end{aligned} \qquad\qquad \begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \ & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t. } & \boldsymbol{c} \leq \boldsymbol{x} - A^\top \boldsymbol{y} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

The A matrix has columns indexed by E and rows indexed by $V \setminus \{s,t\}$, with entries $0, \pm 1$. To explain the entries of A, we just illustrate it for one row, namely the vertex given by v - there is a 1 at column (u,v) for each such valid edge, -1 at column (v,w) for each such valid edge, and 0 elsewhere. So $A\boldsymbol{f} = 0$ forces the conservation-of-flow constraint. The other two constraints, namely $\boldsymbol{c} \geq \boldsymbol{f} \geq 0$ are self explanatory.

The dual is obtained by pattern matching with the above. There are dual variables x_e for each edge capacity constraint and y_v ($v \neq s,t$) for each vertex flow-conservation constraint. The vector $\boldsymbol{a}$ has entry 1 exactly for the edges going to t, and 0 elsewhere. So the corresponding dual constraints have a scalar 1 on one side of the inequality for edges going into t, and 0 for others. Now for each column (u,v) (edge-labeled) of A we have to satisfy the constraint that looks like $c_{(u,v)} \leq x_{(u,v)} - y_v + y_u$. It looks a little different if $u = s$ or $v = t$ and changes into the aforementioned constraints. Combining all of these, the required dual program is

$$
\begin{aligned}
\min_{(x_{(i,j)})\in\mathbb{R}^E,(y_v)\in\mathbb{R}^{V\smallsetminus\{s,t\}}} & \sum_{(i,j)\in E} c_{(i,j)}x_{(i,j)} \\
\text{s.t. } & x_e \geq 0 \; \forall \; e\in E \\
& 1 \leq x_{(u,t)} + y_u \; \forall \; (u,t)\in E, u\neq s \\
& 0 \leq x_{(s,u)} - y_u \; \forall \; (s,u)\in E, u\neq t \\
& 1 \leq x_{(s,t)} \text{ if } (s,t)\in E \\
& y_v - y_u \leq x_{(u,v)} \; \forall \; (u,v)\in E, v\neq t, u\neq s.
\end{aligned} \tag{2}
$$

The last four constraints can be written in one line as $x_{(u,v)} + y_u - y_v \geq 0 \; \forall \; (u,v)\in E$ with the understanding that $y_s = 0, y_t = 1$. This combined with the non-negativity of $\boldsymbol{x}$ is same as saying $x_{e=(u,v)} \geq \max\{0, y_v - y_u\} \;\; \forall \;\, e \in E$. So our modified program is $\min_{\boldsymbol{x},\boldsymbol{y}} \sum_{e\in E} c_e x_e$ subject to $x_{e=(u,v)} \geq \max\{0, y_v - y_u\} \;\; \forall \;\, e \in E$. Since $c \geq 0$, the minimum value occurs when each x_e is minimum, so the $\boldsymbol{x}$ can be eliminated to get an equivalent LP that $\min_{\boldsymbol{y}} \sum_{e=(u,v)\in E} c_e \max\{0, y_v - y_u\}$ subject to $y_s = 0, y_t = 1$. Now note that if any (u,t) is such that $y_u > 1$, then forcefully setting $f_u = 1$ does not change the objective value. Extend this to all vertices one by one to conclude that there is an optimal solution with $y_u \leq 1 \; \forall \; u \in V$. Same argument, starting with (forward) neighbors of s will give that there is an optimal solution with $y_u \geq 0 \; \forall \; u \in V$. Thus our equivalent LP is

$$
\begin{aligned}
\min_{\boldsymbol{y}\in\mathbb{R}^V} & \sum_{(u,v)\in E} c_{(u,v)} \max\{0, y_v - y_u\} \\
\text{s.t. } & 1 \geq y_u \geq 0 \; \forall \; u \in V \\
& y_s = 0, y_t = 1.
\end{aligned} \tag{3}
$$

This was exactly what was asked, to minimize the given cut value by assigning each vertex a real number in $[0,1]$.

(c) We randomly choose a $c \in (0,1)$ and set $S \coloneqq \{v \in V \mid y_v \leq c\}$ and $T \coloneqq \overline{S} = V \smallsetminus S$. This gives an $s-t$ cut (S_c, T_c). Say $(y_v)_{v\in V}$ determines a fractional mincut of value $C = \sum_{(u,v)\in E} c_{(u,v)} \max\{0, y_v - y_u\}$. We round it using our randomized rounding scheme. Then its cut value is $\chi_c = \sum_{e\in E} \mathbf{1}[e \text{ is cut}] \cdot c_e$. Note that $\chi_c \geq C$ for any $c \in (0,1)$ because any rounding determined by c gives a valid cut which must be at least the optimal, namely C. In expectation $\mathbb{E}[\chi_c] = \sum_{e\in E} c_e \cdot \mathbb{P}[e \text{ is cut}]$. However $\mathbb{P}[(u,v) \text{ is cut}] = \mathbb{P}[y_u \leq c < y_v] \leq y_v - y_u$. But probabilities are ≥ 0, so actually $\mathbb{P}[(u,v) \text{ is cut}] \leq \max\{0, y_v - y_u\}$. This means $\mathbb{E}[\chi_c] \leq \sum_{(u,v)\in E} c_{(u,v)} \max\{0, y_v - y_u\} = C$. We have established that the expectation of the non-negative random variable $\chi_c - C$ is 0, which means $\chi_c - C \equiv 0$. In other words, $\chi_c = C \; \forall \; c \in (0,1)$.

Therefore our rounding procedure is to take $c = \frac{1}{2}$ and $S = \left\{v \in V \;\middle|\; y_v \leq \frac{1}{2}\right\}, T = V \smallsetminus S$.

Problem 2

The maximum cut problem asks us to cluster the nodes of a graph $G = (V, E)$ into two disjoint sets X, Y so as to maximize the number of edges between these sets:

$$\max_{X,Y} \sum_{(i,j)\in E} \mathbf{1}\left[(i \in X, j \notin X) \vee (i \in Y, j \notin Y)\right].$$

Consider instead clustering the nodes into three disjoint sets X, Y, Z. Our goal is to maximize the number of edges between different sets:

$$\max_{X,Y,Z} \sum_{(i,j)\in E} \mathbf{1}\left[(i \in X, j \notin X) \vee (i \in Y, j \notin Y)(i \in Z, j \notin Z)\right].$$

Design an algorithm based on SDP relaxation that solves this problem with approximation ration greater then 0.7.

Solution

(We assume undirected graph G just to write the notation $\{i, j\}$) For the problem with two partitions, we had modeled the problem with having variables $x_v \in \{\pm 1\}$ for each vertex $v \in V$. For the corresponding problem with three partitions we will restrict each such variable to be a $2-$vector among $\boldsymbol{a}_1 := (1, 0), \boldsymbol{a}_2 := \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right), \boldsymbol{a}_3 := \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$. It is easy to verify that $\boldsymbol{a}_1^\top \boldsymbol{a}_2 = \boldsymbol{a}_2^\top \boldsymbol{a}_3 = \boldsymbol{a}_3^\top \boldsymbol{a}_1 = -\frac{1}{2}$. The three vertices $\boldsymbol{a}_{1,2,3}$ stand for the three partitions X, Y, Z. Any edge $(u, v) \in E$ that gets assigned different classes of vertices, say $\boldsymbol{x}_u = \boldsymbol{a}_1, \boldsymbol{x}_v = \boldsymbol{a}_2$, contributes exactly $1 = \frac{2}{3}\left(1 - \boldsymbol{a}_1^\top \boldsymbol{a}_2\right)$ to the cut value. If they are in the same class then $\boldsymbol{x}_u = \boldsymbol{x}_v$ and $\boldsymbol{x}_u^\top \boldsymbol{x}_v = 1$ giving a contribution of 0 from the expression $\frac{2}{3}\left(1 - \boldsymbol{x}_u^\top \boldsymbol{x}_v\right)$.

Let's make things formal now. Let $G = (V = [n], E)$ be the given graph. Introduce variables $\boldsymbol{x}_v \in \mathbb{R}^2$, one for each $v \in V$, and constrain them $\boldsymbol{x}_v \in \{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$ where $\boldsymbol{a}_i$ are as in the above paragraph. Given the above discussion, our problem is modeled as follows

$$\begin{aligned} f^* := \max_{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^2} \quad & \frac{2}{3} \sum_{(i,j)\in E} \left(1 - \boldsymbol{x}_i^\top \boldsymbol{x}_j\right) \\ \text{s.t.} \quad & \boldsymbol{x}_i \in \{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\} \ \forall\, i \in V \end{aligned} \tag{4}$$

We are essentially interested in $\displaystyle\min_{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^2} \frac{2}{3} \sum_{(i,j)\in E} \boldsymbol{x}_i^\top \boldsymbol{x}_j$ s.t. $\boldsymbol{x}_i \in \{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\} \ \forall\, i \in V$.

To get an SDP <u>relaxation</u>, we relax our constraints to $\|\boldsymbol{x}_i\|_2 = 1 \ \forall\, i \in V$ and $\boldsymbol{x}_i^\top \boldsymbol{x}_j \geq -\frac{1}{2}$. The last constraint gives the best angle separation among 3 vectors on $\mathbb{S}^1$ in the following sense: if $t \in \mathbb{R}$ is such that $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in \mathbb{S}^{14}$ satisfy $\boldsymbol{v}_i^\top \boldsymbol{v}_j \leq t \ (\forall\, i \neq j)$ then $0 \leq \|\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3\|_2^2 = 3 + 2 \cdot 3 \cdot t \implies t \geq -1/2$. So we design an SDP with the rank-2 matrix $\begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{bmatrix}_{n\times 2} \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n \end{bmatrix}_{2\times n} \succeq 0$ in mind:

$$
\begin{aligned}
\frac{2m}{3} - f_S = \min_{X \in S^{n\times n}} \quad & \mathrm{Tr}\left[\frac{2}{3} Q X\right] \\
\text{s.t.} \quad & X_{ii} = 1 \ \forall\, i \in V \\
& X_{ij} \geq -\frac{1}{2} \ \forall\, i \neq j \in V \\
& X \succeq 0
\end{aligned} \tag{5}
$$

where Q is a matrix whose $(i,j)^{\text{th}}$ entry is $\frac{1}{2}$ if $\{i,j\} \in E$ and 0 otherwise, $S^{n\times n}$ denotes the space of all real symmetric $n \times n$ matrices, and f_S is the optimal value obtained from SDP relaxation.

Let's say the this SDP attains its optimal solution at X^*. Take a Cholesky factorization $X^* = V^\top V$ where $V \in \mathbb{R}^{r\times n}$ and $r = \operatorname{rank} V$. Say V has columns $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n \in \mathbb{S}^{r-1}$ (unit vectors because diagonal of X is 1). So $\frac{2m}{3} - f_S = \frac{2}{3} \sum_{\{i,j\}\in E} \boldsymbol{y}_i^\top \boldsymbol{y}_j \implies f_S = \frac{2}{3} \sum_{\{i,j\}\in E} \left(1 - \boldsymbol{y}_i^\top \boldsymbol{y}_j\right) = \frac{2}{3} \sum_{\{i,j\}\in E} (1 - \cos\theta_{ij})$ where $\theta_{ij} \in [0, 2\pi)$ is the angle between $\boldsymbol{y}_i, \boldsymbol{y}_j$. We relaxed a maximization problem so $f^* \leq f_S$. Also note that $\cos\theta_{ij} = X_{ij} \geq -\frac{1}{2}$ so $\theta_{ij} \in \left[0, \frac{2\pi}{3}\right] \cup \left[\frac{4\pi}{3}, 2\pi\right]$.

In the rounding step, we choose two random vector $\boldsymbol{R}, \boldsymbol{W} \in \mathbb{S}^{r-1}$ uniformly and take inner products $s_i = \overline{\langle \boldsymbol{y}_i, \boldsymbol{R}\rangle}, t_i = \langle \boldsymbol{y}_i, \boldsymbol{W}\rangle$. Then we devise the following scheme: if $s_i > 0$ take $\boldsymbol{x}_i = \boldsymbol{a}_1$, if $s_i \leq 0, t_i > 0$, take $\boldsymbol{x}_i = \boldsymbol{a}_2$, otherwise $s_i \leq 0, t_i \leq 0$ and take $\boldsymbol{x}_i = \boldsymbol{a}_3$.

Let f_R denote the cut value produced by the above-mentioned randomized rounding. So $f_R = \sum_{\{i,j\}\in E} \mathbf{1}\left[\boldsymbol{x}_i \neq \boldsymbol{x}_j\right]$. We are interested in $\mathbb{E}\left[f_R\right] = \sum_{\{i,j\}\in E} \mathbb{P}\left[\boldsymbol{x}_i \neq \boldsymbol{x}_j\right]$ because we eventually want to bound $\frac{\mathbb{E}[f_R]}{f^*}$ which is already $\geq \frac{\mathbb{E}[f_R]}{f_S} = \frac{\sum_{\{i,j\}\in E} \mathbb{P}[\boldsymbol{x}_i \neq \boldsymbol{x}_j]}{\frac{2}{3}\sum_{\{i,j\}\in E}(1-\cos\theta_{ij})}$.

Now let's bound. Focus on one quantity $\mathbb{P}\left[\boldsymbol{x}_1 \neq \boldsymbol{x}_2\right] = 1 - \mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{x}_2\right] = 1 - \mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_1, \boldsymbol{x}_2 = \boldsymbol{a}_1\right] - \mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_2, \boldsymbol{x}_2 = \boldsymbol{a}_2\right] - \mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_3, \boldsymbol{x}_2 = \boldsymbol{a}_3\right]$. We are only using inner products, so the spherical symmetry of $\boldsymbol{R}, \boldsymbol{W}$ implies that it is enough to take them as uniformly random points on the intersection of $\mathbb{S}^{r-1}$ and the hyperplane spanned by $\boldsymbol{x}_1, \boldsymbol{x}_2$ (because it's essentially the projection). This intersection looks like $\mathbb{S}^1$. Therefore think of $\boldsymbol{R}, \boldsymbol{W}$ as independent $\alpha, \beta \sim \text{Unif}((-\pi, \pi])$ where we think of all angles modulo 2π. So $-\frac{\pi}{2}$ would mean $\frac{3\pi}{2}$. We can also WLOG take $\boldsymbol{x}_1 = (1,0), \boldsymbol{x}_2 = (\cos\theta, \sin\theta)$ with $\theta \in (-\pi, \pi]$. Now the feasible α, β for $\mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_1, \boldsymbol{x}_2 = \boldsymbol{a}_1\right]$ are $\alpha \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right] \cap \left[\frac{-\pi}{2} + \theta, \frac{\pi}{2} + \theta\right] = \begin{cases} \left[\frac{-\pi}{2} + \theta, \frac{\pi}{2}\right] & \text{if } \theta \geq 0 \\ \left[\frac{-\pi}{2}, \frac{\pi}{2} + \theta\right] & \text{otherwise} \end{cases}$ and $\beta \in (-\pi, \pi]$. So $\mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_1, \boldsymbol{x}_2 = \boldsymbol{a}_1\right] = \frac{\pi - |\theta|}{2\pi}$. Similar analysis shows that $\mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_2, \boldsymbol{x}_2 = \boldsymbol{a}_2\right] = \mathbb{P}\left[\boldsymbol{x}_1 = \boldsymbol{a}_3, \boldsymbol{x}_2 = \boldsymbol{a}_3\right] = \left(\frac{\pi - |\theta|}{2\pi}\right)^2$. Therefore $\mathbb{P}\left[\boldsymbol{x}_1 \neq \boldsymbol{x}_2\right] = 1 - \frac{\pi - |\theta|}{2\pi} - \frac{(\pi - |\theta|)^2}{2\pi^2}$. WolframAlpha shows that $\dfrac{1 - \frac{\pi - |\theta|}{2\pi} - \frac{(\pi - |\theta|)^2}{2\pi^2}}{\frac{2}{3}(1 - \cos\theta)} \geq \dfrac{7}{9} \simeq 0.77$ for $\theta \in \left[0, \frac{2\pi}{3}\right] \cup \left[-\frac{2\pi}{3}, 0\right]$. We conclude by the previous paragraph that $\mathbb{E}\left[f_R\right] \geq 0.77 f^*$.

Problem 3

The Ellipsoid algorithm we saw in the lecture solves convex programs assuming a separation oracle. Here, we want to show the opposite. To be more specific, consider the following two tasks regarding a convex body $\mathcal{K}$:

- $\texttt{OPTIMIZE}(\mathcal{K})$: given a vector $\boldsymbol{c} \in \mathbb{R}^n$, output $\arg\max_{\boldsymbol{x}\in\mathcal{K}} \boldsymbol{c}^\top \boldsymbol{x}$;
- $\texttt{SEPARATE}(\mathcal{K})$: given a point $\boldsymbol{x} \in \mathbb{R}^n$, output either $\boldsymbol{x} \in \mathcal{K}$ or a separating hyperplane.

We are going to show that if for a specific convex body $\mathcal{K}$, there is a polynomial time algorithm for $\texttt{OPTIMIZE}(\mathcal{K})$, then there is a polynomial time algorithm for $\texttt{SEPARATE}(\mathcal{K})$.

(a) Suppose for a given x, we can solve the following LP with infinitely many constraints (finding the optimal w and T). Show that we can use such an algorithm to solve $\texttt{SEPARATE}(\mathcal{K})$.

$$\begin{aligned} \max_{\boldsymbol{w}\in\mathbb{R}^n, T\in\mathbb{R}} \quad & \boldsymbol{w}^\top \boldsymbol{x} - T \\ \text{s.t.} \quad & \boldsymbol{w}^\top \boldsymbol{y} \le T \ \forall\, \boldsymbol{y} \in \mathcal{K} \\ & -1 \le T \le 1 \end{aligned} \tag{6}$$

(b) Design a polytime separation oracle for the above LP using $\texttt{OPTIMIZE}(\mathcal{K})$, and conclude.

Solution

(a) Suppose the value of this LP is > 0 and is attained at $(\overline{\boldsymbol{w}}, \overline{T})$. Then for any $\boldsymbol{y} \in \mathcal{K}$, $\overline{\boldsymbol{w}}^\top \boldsymbol{y} - \overline{T} \le 0$. This means that $\boldsymbol{x} \notin \mathcal{K}$.
Suppose $\boldsymbol{x} \notin \mathcal{K}$. Then there is a vector $\boldsymbol{w} \in \mathbb{R}^n$ such that $\boldsymbol{w}^\top \boldsymbol{x} > 0$ and $\boldsymbol{w}^\top \boldsymbol{y} \le 0 \ \forall\, \boldsymbol{y} \in \mathcal{K}$. This $(\boldsymbol{w}, T = 0)$ is feasible to 6 with objective > 0. Thus its optimal value is > 0.
Therefore $\boldsymbol{x} \in \mathcal{K}$ iff the optimal value of 6 is ≤ 0. If ≤ 0 with optimal $\boldsymbol{w} = \overline{\boldsymbol{w}}$, a separating hyperplane is $\overline{\boldsymbol{w}}$ because of what is discussed above.

(b) The feasible set of 6 is $\mathcal{S} = \left\{ (\boldsymbol{w}, T) \in \mathbb{R}^{n+1} \;\middle|\; -1 \le T \le 1, \boldsymbol{w}^\top \boldsymbol{y} \le T \ \forall\, \boldsymbol{y} \in \mathcal{K} \right\}$. We want to design a polytime separation oracle for $\mathcal{S}$, that is, given any $(\boldsymbol{a}, s) \in \mathbb{R}^n \times \mathbb{R}$, the oracle will either say $(\boldsymbol{a}, s) \in \mathcal{S}$ or will give a separating hyperplane with normal $(\boldsymbol{v}, t) \in \mathbb{R}^n \times \mathbb{R}$ such that $\langle (\boldsymbol{v}, t), (\boldsymbol{w}, T) \rangle \le 0 \ \forall\, (\boldsymbol{w}, T) \in \mathcal{S}$ and $\langle (\boldsymbol{v}, t), (\boldsymbol{a}, s) \rangle > 0$.
Let $(\boldsymbol{a}, s)$ be an input to the oracle. We have access to $\texttt{OPTIMIZE}(\mathcal{K})$, so we can determine $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}\in\mathcal{K}} \boldsymbol{a}^\top \boldsymbol{y}$. If $\boldsymbol{a}^\top \boldsymbol{y}^* \le s$ then $\boldsymbol{a}^\top \boldsymbol{y} \le \boldsymbol{a}^\top \boldsymbol{y}^* \le s$ by definition of $\boldsymbol{y}^*$ whence $(\boldsymbol{a}, s) \in \mathcal{S}$. Otherwise, $\boldsymbol{a}^\top \boldsymbol{y}^* > s$ and $(\boldsymbol{y}^*, -1)$ determines a hyperplane that separates $(\boldsymbol{a}, s)$ from $\mathcal{S}$. Indeed if $(\boldsymbol{w}, T) \in \mathcal{S}$ then $\langle (\boldsymbol{y}^*, -1), (\boldsymbol{w}, T) \rangle = \boldsymbol{w}^\top \boldsymbol{y}^* - T \le 0$ by definition of $\mathcal{S}$ and since $\boldsymbol{y}^* \in \mathcal{K}$, and $\langle (\boldsymbol{y}^*, -1), (\boldsymbol{a}, s) \rangle = \boldsymbol{a}^\top \boldsymbol{y}^* - s > 0$ by our hypothesis on the value $\boldsymbol{a}^\top \boldsymbol{y}^*$. This oracle is polytime because vector inner product operation requires linear (in n) many operations and $\texttt{OPTIMIZE}(\mathcal{K})$ is a polytime oracle, so the overall number of steps is polynomial in n.
Once we have a separation oracle for the feasible set of 6, we use the ellipsoid algorithm to solve the LP. Since all oracles are polytime and the ellipsoid algorithm is polytime, the LP can be solved in polynomial time. We earlier showed that if the LP can be solved in polynomial time, then we can solve $\texttt{SEPARATE}(\mathcal{K})$. This shows how $\texttt{OPTIMIZE}(\mathcal{K})$ is used to solve $\texttt{SEPARATE}(\mathcal{K})$ in polynomial time.

Problem 4

Describe separation oracles for the following convex sets. Your oracles should run in linear time, assuming that the given oracles run in linear time (so you can make a constant number of black-box calls to the given oracles).

(a) The ℓ_1 ball, $\{\boldsymbol{x} : \|\boldsymbol{x}\|_1 \leq 1\}$. Recall that $\|\boldsymbol{x}\|_1 = \sum\limits_{i=1}^{n} |\boldsymbol{x}_i|$.

(b) Any convex set A that we have a projection oracle for. i.e. we have an oracle to compute $\arg\min\limits_{\boldsymbol{x}\in A} \|\boldsymbol{x} - \boldsymbol{y}\|_2$ for any $\boldsymbol{y}$.

(c) The $\varepsilon-$neighborhood E of any convex set A:

$$E = \{\boldsymbol{x} \ \mid \ \exists\, \boldsymbol{y} \in A \text{ with } \|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq \varepsilon\}$$

given a projection oracle for A.

Solution

(a) Let the input point be $\boldsymbol{a} \in \mathbb{R}^n$. First we compute $\|\boldsymbol{a}\|_1 = \sum_{i=1}^n \boldsymbol{a}_i$ in $\mathcal{O}(n)$ time.
If $\|\boldsymbol{a}\|_1 \leq 1$, we report that $\boldsymbol{a}$ is in the ℓ_1 ball.
If not, that is $\|\boldsymbol{a}\|_1 > 1$, then the hyperplane $\mathcal{H} := \left\{\boldsymbol{x} \in \mathbb{R}^n \ \middle|\ \mathbf{1}^\top \boldsymbol{x} = \frac{\|\boldsymbol{a}\|_1 + 1}{2}\right\}$ where $\mathbf{1} = \mathbf{1}_n \in \mathbb{R}^n$ is the vector comprising of all one's. Indeed $\mathbf{1}^\top \boldsymbol{a} = \|\boldsymbol{a}\|_1 = \frac{\|\boldsymbol{a}\|_1}{2} + \frac{\|\boldsymbol{a}\|_1}{2} > \frac{\|\boldsymbol{a}\|_1+1}{2}$, and if $\boldsymbol{x}$ is in the ℓ_1 ball then $\mathbf{1}^\top \boldsymbol{x} = \|\boldsymbol{x}\|_1 \leq \frac{1+1}{2} < \frac{\|\boldsymbol{a}\|_1+1}{2}$. Again, this hyperplane is computed in $\mathcal{O}(n)$ steps. Overall it requires at most $2\mathcal{O}(n) = \mathcal{O}(n)$ steps.

(b) Let's start with a fundamental lemma about projections onto convex sets A.

Lemma 1
Denote by $\pi(\boldsymbol{x}) := \arg\min\limits_{\boldsymbol{y}\in A} \|\boldsymbol{y} - \boldsymbol{x}\|_2$ which is a point in the closure $\overline{A}$ of A. Then $(\boldsymbol{x} - \pi(\boldsymbol{x}))^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \leq 0 \ \forall\, \boldsymbol{z} \in A$.

Proof. A is convex $\implies$ $\overline{A}$ is convex. Fix points $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in A$. Take the variable point $\boldsymbol{z}_t := (1-t)\pi(\boldsymbol{x}) + t\boldsymbol{z}$ for $t \in (0,1)$. For brevity we take $s = 1-t$. Each $\boldsymbol{z}_t \in \overline{A}$ as $\overline{A}$ is convex and both $\pi(\boldsymbol{x}), \boldsymbol{z} \in \overline{A}$. For each t and by construction of $\pi(\boldsymbol{x})$,

$$
\begin{aligned}
& \|\boldsymbol{z}_t - \boldsymbol{x}\|^2 \geq \|\pi(\boldsymbol{x}) - \boldsymbol{x}\|^2 \\
\implies & \|\boldsymbol{z}_t\|^2 - 2\boldsymbol{x}^\top \boldsymbol{z}_t \geq \|\pi(\boldsymbol{x})\|^2 - 2\boldsymbol{x}^\top \pi(\boldsymbol{x}) \\
\implies & \|\boldsymbol{z}_t\|^2 - \|\pi(\boldsymbol{x})\|^2 \geq 2\boldsymbol{x}^\top (\boldsymbol{z}_t - \pi(\boldsymbol{x})) \\
\implies & \left[s^2 \|\pi(\boldsymbol{x})\|^2 + t^2 \|\boldsymbol{z}\|^2 + 2st\boldsymbol{z}^\top \pi(\boldsymbol{x})\right] - \|\pi(\boldsymbol{x})\|^2 \geq 2t\boldsymbol{x}^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \\
\implies & t^2 \|\boldsymbol{z}\|^2 + 2st\boldsymbol{z}^\top \pi(\boldsymbol{x}) + (s-1)(s+1) \|\pi(\boldsymbol{x})\|^2 \geq 2t\boldsymbol{x}^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \\
\implies & t^2 \|\boldsymbol{z}\|^2 + 2st\boldsymbol{z}^\top \pi(\boldsymbol{x}) - t(s+1) \|\pi(\boldsymbol{x})\|^2 \geq 2t\boldsymbol{x}^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \\
\implies & t \|\boldsymbol{z}\|^2 + 2(1-t)\boldsymbol{z}^\top \pi(\boldsymbol{x}) - (2-t) \|\pi(\boldsymbol{x})\|^2 \geq 2\boldsymbol{x}^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \qquad [\because t > 0] \\
\overset{t \to 0}{\implies} & 2\boldsymbol{z}^\top \pi(\boldsymbol{x}) - 2 \|\pi(\boldsymbol{x})\|^2 \geq 2\boldsymbol{x}^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \\
\implies & \pi(\boldsymbol{x})^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \geq \boldsymbol{x}^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \\
\implies & (\boldsymbol{x} - \pi(\boldsymbol{x}))^\top (\boldsymbol{z} - \pi(\boldsymbol{x})) \leq 0
\end{aligned}
$$

■

Now assume A is closed so that the argmin actually exists in A. Let $\boldsymbol{a}$ be the given input point. We compute $\boldsymbol{y}^* = \arg\min_{\boldsymbol{y} \in A} \|\boldsymbol{y} - \boldsymbol{a}\|_2$ in $\mathcal{O}(n)$ steps.
If $\boldsymbol{y}^* = \boldsymbol{a}$ then $\boldsymbol{a} \in A$ and our algorithm returns this.
If not, take $\boldsymbol{p} := \frac{\boldsymbol{a}+\boldsymbol{y}^*}{2}$ and the hyperplane $\mathcal{H} := \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; (\boldsymbol{a} - \boldsymbol{y}^*)^\top (\boldsymbol{p} - \boldsymbol{x}) = 0\right\}$. This hyperplane passes through $\boldsymbol{p}$ and is perpendicular to $\boldsymbol{a} - \boldsymbol{y}^*$. So $(\boldsymbol{a} - \boldsymbol{y}^*)^\top (\boldsymbol{p} - \boldsymbol{a}) = -(\boldsymbol{a} - \boldsymbol{y}^*)^\top \left(\frac{\boldsymbol{a}-\boldsymbol{y}^*}{2}\right) < 0$ because $\boldsymbol{a} \neq \boldsymbol{y}^*$. And if $\boldsymbol{x} \in A$ then $(\boldsymbol{a} - \boldsymbol{y}^*)^\top (\boldsymbol{p} - \boldsymbol{x}) = (\boldsymbol{a} - \boldsymbol{y}^*)^\top (\boldsymbol{p} - \boldsymbol{y}^*) + (\boldsymbol{a} - \boldsymbol{y}^*)^\top (\boldsymbol{y}^* - \boldsymbol{x}) \overset{\text{Lemma 1}}{\geq} (\boldsymbol{a} - \boldsymbol{y}^*)^\top (\boldsymbol{p} - \boldsymbol{y}^*) = (\boldsymbol{a} - \boldsymbol{y}^*)^\top \left(\frac{\boldsymbol{a}-\boldsymbol{y}^*}{2}\right) > 0$. Computing this hyperplane takes $\mathcal{O}(n)$ steps (for vector multiplication). Since we made only one call to the given blackbox, the overall number of steps is $\mathcal{O}(n)$.

(c) We denote $\pi(\boldsymbol{x}) = \arg\min_{\boldsymbol{y} \in A} \|\boldsymbol{y} - \boldsymbol{x}\|$.

Claim 2
E is convex.

Proof. Take two points $\boldsymbol{x}, \boldsymbol{y} \in E$ and $t \in [0,1]$ and let $s = 1 - t$. We will show $t\boldsymbol{x} + s\boldsymbol{y} \in E$. There are points $a, b \in A$ such that $\|\boldsymbol{a} - \boldsymbol{x}\| \leq \varepsilon, \|\boldsymbol{b} - \boldsymbol{y}\| \leq \varepsilon$. By convexity of A, $t\boldsymbol{a} + s\boldsymbol{b} \in A$. Clearly $\|(t\boldsymbol{a} + s\boldsymbol{b}) - (t\boldsymbol{x} + s\boldsymbol{y})\| = \|t(\boldsymbol{a} - \boldsymbol{x}) + s(\boldsymbol{b} - \boldsymbol{y})\| \leq t\|\boldsymbol{a} - \boldsymbol{x}\| + s\|\boldsymbol{b} - \boldsymbol{y}\| \leq t\varepsilon + s\varepsilon = \varepsilon$. ■

Claim 3
$\boldsymbol{x} \in E$ iff $\|\boldsymbol{x} - \pi(\boldsymbol{x})\| \leq \varepsilon$.

Proof. Say $\boldsymbol{x} \in E$. Then $\exists \boldsymbol{y} \in A$ such that $\|\boldsymbol{x} - \pi(\boldsymbol{x})\| \overset{\text{definition of } \pi(\boldsymbol{x})}{\leq} \|\boldsymbol{x} - \boldsymbol{y}\| \leq \varepsilon$.

Say $\boldsymbol{x} \notin E$. Then $\forall \boldsymbol{y} \in A, \|\boldsymbol{x} - \boldsymbol{y}\| > \varepsilon$. So de definition of $\pi(\boldsymbol{x})$, $\|\boldsymbol{x} - \pi(\boldsymbol{x})\| \geq \varepsilon$. But since $\pi(\boldsymbol{x}) \in A$, the equality can't occur as otherwise $\boldsymbol{x}$ would be in E, whence $\|\boldsymbol{x} - \pi(\boldsymbol{x})\| > \varepsilon$. ■

Claim 4
If $\boldsymbol{x} \notin E$ then $\pi'(\boldsymbol{x}) \coloneqq \pi(\boldsymbol{x}) + \varepsilon \frac{\boldsymbol{x}-\pi(\boldsymbol{x})}{\|\boldsymbol{x}-\pi(\boldsymbol{x})\|} \in E$ satisfies $\pi'(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}\in E} \|\boldsymbol{x}-\boldsymbol{y}\|$.

Proof. The given point $\pi'(\boldsymbol{x}) \in E$ because $\|\pi'(\boldsymbol{x}) - \pi(\boldsymbol{x})\| = \varepsilon$ and $\pi(\boldsymbol{x}) \in A$.
If $\boldsymbol{q} \in E$ then $\|\boldsymbol{x}-\boldsymbol{q}\| \overset{\Delta-\text{ineq}}{\geq} \|\boldsymbol{x}-\pi(\boldsymbol{q})\| - \|\pi(\boldsymbol{q})-\boldsymbol{q}\|] \overset{\text{Claim 3}}{\geq} \|\boldsymbol{x}-\pi(\boldsymbol{q})\| - \varepsilon \geq \|\boldsymbol{x}-\pi(\boldsymbol{x})\| - \varepsilon = \|\boldsymbol{x}-\pi(\boldsymbol{x})\| \left(1 - \frac{\varepsilon}{\|\boldsymbol{x}-\pi(\boldsymbol{x})\|}\right) = \left\|\boldsymbol{x}-\pi(\boldsymbol{x}) - \frac{\varepsilon(\boldsymbol{x}-\pi(\boldsymbol{x}))}{\|\boldsymbol{x}-\pi(\boldsymbol{x})\|}\right\| = \|\boldsymbol{x}-\pi'(\boldsymbol{x})\|$. ■

Again assume A is closed. Note that as a byproduct of the proof of claim 4, we showed that E is closed because if $\boldsymbol{x} \in \partial E$ then $\|\boldsymbol{x}-\pi(\boldsymbol{x})\| = \varepsilon$. Our oracle will work as follows for a given $\boldsymbol{a}$. Since we have an oracle to compute $\pi(\boldsymbol{a})$, we can also find $\pi'(\boldsymbol{a}) = \arg\min_{\boldsymbol{y}\in E} \|\boldsymbol{a}-\boldsymbol{y}\|$ in $\mathcal{O}(n)$ time by claim 4. This exactly reduces to the previous problem, replacing A by E.

Problem 5

In class we designed a $3/4-$approximation for MAX-2SAT using LP rounding. The MAX-SAT problem is similar except for the fact that the clauses can contain any number of literals. Formally, the input consists of n boolean variables $x_1, x_2, \cdots, x_n$ (each may be either 0 (false) or 1 (true)), m clauses $C_1, \cdots, C_m$ (each of which consists of disjunction (an "or") of some number variables or their negations) and a non-negative weight w_i for each clause. The objective is to find an assignment of 1 or 0 to x_i's that maximize the total weight of satisfied clauses. As we saw in the class, a clause is satisfied if one of its non-negated variable is set to 1, or one of the negated variable is set to 0. You can assume that no literal is repeated in a clause and at most one of x_i or $\neg x_i$ appears in any clause.

(a) Generalize the LP relaxation for MAX-2SAT seen in the class to obtain a LP relaxation of the MAX-SAT problem.

(b) Use the standard randomized rounding algorithm (the same one we used in class for MAX-2SAT) on the LP-relaxation you designed in part (1) to give a $(1 - 1/e)$ approximation algorithm for MAX-SAT. Recall that clauses can be of any length.
(Hint: there is a clean way to resolve "the math" without excessive calculations).

(c) A naive algorithm for MAX-SAT problem is to set each variable to true with probability $1/2$ (without writing any LP). It is easy to see that this unbiased randomized algorithm of MAX-SAT achieves $1/2$-approximation in expectation. Show the algorithm that returns the best of two solutions given by the randomized rounding of the LP and the simple unbiased randomized algorithm is a $3/4-$approximation algorithm of MAX-SAT. (Hint: it may help to realize that in fact randomly selecting one of these two algorithms to run also gives a $3/4-$approximation in expectation).

(d) Using the previous part (and in particular, the hint) for intuition, design a direct rounding scheme of your LP relaxation to get a $3/4-$approximation (that is, design a function f which assigns a literal x_i to be true independently with probability $f(z)$ when the corresponding variable z_i in your LP relaxation is equal to z). (Hint: here, it may get messy to fully resolve the calculations. You will get full credit if you state the correct rounding scheme and clearly state the necessary inequalities for the proof. You should also attempt to show that the inequalities hold for your own benefit, but not for full credit).

Solution

(a) MAX-SAT is the following integer program where we have n variables $x_1, \cdots, x_n$ and m clauses (modelled by the values $z_1, \cdots, z_m$) where the i^{th} clause has t_i many literals:

$$\begin{aligned} \max_{\boldsymbol{z}\in\mathbb{R}^m, \boldsymbol{x}\in\mathbb{R}^n} \; & w_1 z_1 + \cdots + w_m z_m \\ \text{s.t. } & x_1, \cdots, x_n \in \{0, 1\} \\ & z_k \leq y_k^{(1)} + \cdots + y_k^{(t_k)} \; \forall\, 1 \leq k \leq m \\ & z_k \leq 1 \; \forall\, 1 \leq k \leq m \end{aligned} \tag{7}$$

where $y_j^{(i)}$ are either the literal x corresponding to a variable or their negation $1-x$ counting contribution towards the j^{th} clause. **We shall assume that in each clause, the literals come from distinct terms** since repeated literals can be combined into one, and presence of a variable and its negation gives 0 overall contribution. This integer program can be relaxed to the following LP:

$$\begin{aligned} \max_{\boldsymbol{z}\in\mathbb{R}^m,\boldsymbol{x}\in\mathbb{R}^n} & w_1 z_1 + \cdots + w_m z_m \\ \text{s.t. } & x_1, \cdots, x_n \in [0,1] \\ & z_k \le y_k^{(1)} + \cdots + y_k^{(t_k)} \ \forall\, 1 \le k \le m \\ & z_k \le 1 \ \forall\, 1 \le k \le m. \end{aligned} \tag{8}$$

(b) Say we solve LP 8 to get the optimal optimal solution $\overline{z}_1, \cdots, \overline{z}_m, \overline{x}_1, \cdots, \overline{x}_n$ with optimal value $\overline{Z} = \sum_{i=1}^{m} w_i \overline{z}_i$. They satisfy $\overline{z}_k \le \min\left\{1, \sum_{j=1}^{t_k} \overline{y}_k^{(j)}\right\}$, where $\overline{y}_k^{(j)}$ are determined by $\overline{x}_i$'s.

(To see why such an optimal solution exists we first observe that adding the constraint $z_k \ge 0$ does not change anything because if $z_{k_0} < 0$ for some k_0 then force-setting $z_{k_0} = 0$ only increases the objective. This makes the feasible set compact. We recall the theorem from analysis that any continuous function attains a maxima over a compact set, and conclude the existence of a solution.)

Now we design the following randomized rounding. For each $1 \le i \le n$, take $\tilde{x}_i$ to be 1 with probability $\overline{x}_i$, and 0 with probability $1 - \overline{x}_i$ (so toss an $\tilde{x}_i$−coin). The $\tilde{y}_k^{(i)}$ are determined by the values of the $\tilde{x}_j$, and so $\tilde{z}_k = \min\left\{1, \tilde{y}_k^{(1)} + \cdots + \tilde{y}_k^{(t_k)}\right\} = \begin{cases} 0 & \text{if } \tilde{y}_k^{(1)} = \cdots = \tilde{y}_k^{(t_k)} = 0 \\ 1 & \text{otherwise} \end{cases}$ to maximize the total value of the objective. The random variable denoting the objective is $\tilde{Z} = \sum_{i=1}^{m} w_i \tilde{z}_i$ and has expectation $\sum_{i=1}^{m} w_i \mathbb{E}\left[\tilde{z}_i\right]$. This lets us analyze term-by-term.

Let's only focus on one term $\tilde{z}$ (with the corresponding $\overline{z}$), say $\tilde{z} = \tilde{z}_1 = \sum_{j=1}^{t_1} \tilde{y}_1^{(j)}$. For short we refer to them as only $\tilde{y}^{(j)}$ and say (without loss of generality) $y^{(j)}$ comes only from the variable x_j. Further, for $1 \le j \le t_1$, let $\alpha_j \coloneqq \mathbb{P}\left[\tilde{y}^{(j)} = 1\right] = \overline{y}^{(j)} = \begin{cases} \overline{x}_{p_j} & \text{if } y^{(j)} = x_j \\ 1 - \overline{x}_{p_j} & \text{if } y^{(j)} = \neg x_j \end{cases}$.
Now $\mathbb{E}\left[\tilde{z}\right] = \mathbb{P}\left[\tilde{z} = 1\right] = 1 - \mathbb{P}\left[\tilde{z} = 0\right] = 1 - \prod_{j=1}^{t_1} \mathbb{P}\left[y^{(j)} = 0\right] = 1 - \prod_{j=1}^{t_1} (1 - \alpha_j)$. Denote $S \coloneqq \sum_{j=1}^{t_1} \alpha_j \ge \overline{z}$. Combining,

$$\prod_{j=1}^{t_1} (1-\alpha_j) \overset{\text{AM-GM}}{\le} \left(\frac{1}{t_1} \sum_{j=1}^{t_1} (1-\alpha_j) \right)^{t_1} = \left(\frac{t_1 - S}{t_1} \right)^{t_1} = \left(1 - \frac{S}{t_1} \right)^{t_1} \overset{S \ge \overline{z}}{\le} \left(1 - \frac{\overline{z}}{t_1} \right)^{t_1}.$$

The last inequality is possible because $S = \sum_{j=1}^{t_1} \alpha_j \le t_1 \implies 1 - \frac{S}{t_1} \ge 0$.

Note that $\frac{\mathrm{d}^2}{\mathrm{d}\,z^2}\left(1-\frac{z}{t_1}\right)^{t_1}\bigg|_{z=\overline{z}} = \frac{t_1-1}{t_1}\left(1-\frac{\overline{z}}{t_1}\right)^{t_1-2} \geq 0$ for $t_1 \geq 1$, hence convex. It follows that $\left(1-\frac{\overline{z}}{t_1}\right)^{t_1} \leq (1-\overline{z})\left(1-\frac{0}{t_1}\right)^{t_1} + \overline{z}\left(1-\frac{1}{t_1}\right)^{t_1} \implies 1-\left(1-\frac{\overline{z}}{t_1}\right)^{t_1} \geq \overline{z}\left(1-\left(1-\frac{1}{t_1}\right)^{t_1}\right)$.

Thus $\dfrac{\mathbb{E}\left[\tilde{z}\right]}{\overline{z}} = \dfrac{1-\prod_{j=1}^{t_1}(1-\alpha_j)}{\overline{z}} \geq \dfrac{1}{\overline{z}}\left(1-\left(1-\dfrac{\overline{z}}{t_1}\right)^{t_1}\right) \geq 1-\left(1-\dfrac{1}{t_1}\right)^{t_1} \geq 1-\dfrac{1}{e}$.

By linearity of expectation, $\mathbb{E}\left[\tilde{Z}\right] = \sum_{i=1}^{m} w_i \mathbb{E}\left[\tilde{z}_i\right] \geq \sum_{i=1}^{m}\left(1-\frac{1}{e}\right) w_i \overline{z}_i = \left(1-\frac{1}{e}\right)\overline{Z}$.

(c) Let's use the same notation as above, namely $\tilde{x}_i, \tilde{y}_i, \tilde{z}_i$. We now take $\tilde{x}_i$ to be $0/1$ by tossing a fair coin. Looking at the exact same analysis as above, we have all $\alpha_i = \frac{1}{2}$ which makes $\mathbb{E}\left[\tilde{z}_i\right] = 1-2^{-t_i}$ which is $\geq \frac{1}{2}$ because $t_i \geq 1$. We conclude by linearity of expectation that this gives a $\frac{1}{2}-$approximation.

The key point in these two analyses is that for the $LP-$based rounding, we have an approximation ratio $\geq \sum_{i=1}^{m}\left[1-\left(1-\frac{1}{t_i}\right)^{t_i}\right]$ per clause, where t_i is the number of terms in clause i, and the unbiased rounding gives a ratio $\geq \sum_{i=1}^{m} 1-\frac{1}{2^{t_i}}$ per clause. So if our algorithm was to take the better of the two, namely $\max\left\{\tilde{Z}_{LP}, \tilde{Z}_{1/2}\right\} \geq \frac{\tilde{Z}_{LP}+\tilde{Z}_{1/2}}{2}$, where $\tilde{Z}_{LP}, \tilde{Z}_{1/2}$ are the final values obtained by using the LP-rounding and unbiased rounding respectively, then we get the approximation $\dfrac{\tilde{Z}_{LP}+\tilde{Z}_{\frac{1}{2}}}{2} \geq \displaystyle\sum_{i=1}^{m} \dfrac{\left(1-\left(1-\frac{1}{t_i}\right)^{t_i}\right)+\left(1-2^{-t_i}\right)}{2} w_i \overline{z}_i$. We note that

- $t = 1,2 \implies \dfrac{\left(1-\left(1-\frac{1}{t}\right)^{t}\right)+\left(1-2^{-t}\right)}{2} = \dfrac{3}{4}$.
- $t \geq 3 \implies \dfrac{\left(1-\left(1-\frac{1}{t}\right)^{t}\right)+\left(1-2^{-t}\right)}{2} \geq \dfrac{1}{2}\left(1-\dfrac{1}{e}+1-\dfrac{1}{8}\right) \geq \dfrac{1}{2}\left(1-\dfrac{3}{8}+1-\dfrac{1}{8}\right) = \dfrac{3}{4}$ where we used the fact that $e \geq \frac{8}{3}$.

We thus obtained that $\dfrac{\max\left\{\tilde{Z}_{LP}, \tilde{Z}_{\frac{1}{2}}\right\}}{2\overline{Z}} \geq \dfrac{3}{4}$.

(d) We are looking for a function f which satisfies that $1-\prod_{i=1}^{t}(1-\alpha_i) \geq \frac{3}{4}\min\left\{1, \sum_{i=1}^{t}\overline{y}_i\right\}$ where α_i is $f(\overline{x_i})$ or $1-f(\overline{x}_i)$ depending on whether the corresponding $y^{(i)} = x_i$ or $y^{(i)} = \neg x_i$ respectively, and the value of $\overline{y}_i = \neg\overline{x}_i$ is interpreted as $1-\overline{x}_i$ in case $y_i = \neg x_i$. For simplicity look at a clause with no negated variable and we want to find f such that $1-\prod_{i=1}^{t}(1-f(x_i)) \geq \frac{3}{4}\min\left\{1, \sum_{i=1}^{t} x_i\right\}$, as long as $f(x) \leq 1-f(1-x)$ so that we can replace the corresponding terms in the product.

We ideally want to turn product into sum so want to take $f(x) = 1-\alpha^x$ so that $\prod(1-f(x_i)) = \alpha^{\sum x_i}$ and can use that to compare with $\frac{3}{4}\sum x_i$. In other words we want to find a constant α such that $1-\alpha^x \geq \frac{3}{4}x$ whenever $x \in [0,1]$.

Lemma 5
$4^{-x} \leq 1 - \min\left\{\frac{3}{4}, \frac{3x}{4}\right\} \forall x \geq 0$

Proof. If $x \geq 1$ then $4^{-x} \leq \frac{1}{4} = 1 - \frac{3}{4} = 1 - \min\left\{\frac{3}{4}, \frac{3x}{4}\right\}$.
Assume $x \in [0,1]$. We want to show $4^{-x} \leq 1 - \frac{3x}{4}$. The second derivative of $h(x) := 1 - \frac{3x}{4} - 4^{-x}$ is $-(\ln(4))^2 \cdot 4^{-x} < 0$ hence concave. Note $h(0) = h(1) = 0$ which means $x \in [0,1] \implies h(x) \geq xh(0) + (1-x)h(1) = 0$. ∎

For a final verification, let's check that $1 - 4^{-x} \leq 4^{x-1}$ for $x \in [0,1]$. Graphing in GeoGebra indicates that $1 - 4^{-x} - 4^{x-1} \leq 0$ which can be easily checked with calculus.

So our rounding scheme will be to let x_i be 1 with probability $f(\overline{x}_i) = 1 - 4^{-\overline{x}_i}$ and 0 with probability $4^{-\overline{x}_i}$.