ADVANCED ALGORITHM DESIGN
Homework 2

October 27, 2024

Problem 1

Solution

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

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

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 relaxation, 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 $\mathbb{S}^2$ in the following sense: if $t \in \mathbb{R}$ is such that $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in \mathbb{S}^2$ 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:

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

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

Let's say the optimal solution of this SDP is attained at X^*, take a Cholesky factorization $X^* = V^\top V$ where $V \in \mathbb{R}^{r\times n}$ and $r = \operatorname{rank} V$. Let the columns of V be $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n \in \mathbb{R}^r$. In the rounding step, we choose random vectors $\boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_3 \in \mathbb{R}^r$ such that each $\boldsymbol{R}_{i,j} \sim \mathcal{N}(0,1)$ (for $1 \leq j \leq r$) is chosen independently. These will give us the partitions, by rounding each $\boldsymbol{y}_i$ to the component nearest among $\boldsymbol{R}_j$. More precisely, we partition $V = V_1 \sqcup V_2 \sqcup V_3$ as follows:

$$\begin{aligned} V_1 &:= \left\{ i \in V \ \middle|\ \boldsymbol{y}_i^\top \boldsymbol{R}_1 \geq \boldsymbol{y}_i^\top \boldsymbol{R}_2, \boldsymbol{y}_i^\top \boldsymbol{R}_1 \geq \boldsymbol{y}_i^\top \boldsymbol{R}_3 \right\} \\ V_2 &:= \left\{ i \in V \ \middle|\ \boldsymbol{y}_i^\top \boldsymbol{R}_2 \geq \boldsymbol{y}_i^\top \boldsymbol{R}_1, \boldsymbol{y}_i^\top \boldsymbol{R}_2 \geq \boldsymbol{y}_i^\top \boldsymbol{R}_3 \right\} \\ V_3 &:= \left\{ i \in V \ \middle|\ \boldsymbol{y}_i^\top \boldsymbol{R}_3 \geq \boldsymbol{y}_i^\top \boldsymbol{R}_2, \boldsymbol{y}_i^\top \boldsymbol{R}_3 \geq \boldsymbol{y}_i^\top \boldsymbol{R}_1 \right\} \end{aligned}$$

while breaking ties at random. In fact assign $\boldsymbol{x}_i := \boldsymbol{a}_j$ if $i \in V_j$.

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 $\mathbb{E}\left[f_R\right] = \displaystyle\sum_{\{i,j\}\in E} \mathbb{P}\left[\boldsymbol{x}_i \neq \boldsymbol{x}_j\right]$.

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 $\mathcal{K}$:

- `OPTIMIZE`$(\mathcal{K})$: given a vector $c \in \mathbb{R}^n$, output $\arg\max_{x \in \mathcal{K}} c^\top x$;
- `SEPARATE`$(\mathcal{K})$: given a point $x \in \mathbb{R}^n$, output either $x \in \mathcal{K}$ 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 `OPTIMIZE`$(\mathcal{K})$, then there is a polynomial time algorithm for `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 `SEPARATE`$(\mathcal{K})$.

$$\begin{aligned} \max_{w \in \mathbb{R}^n, T \in \mathbb{R}} \quad & w^\top x - T \\ \text{s.t.} \quad & w^\top y \leq T \ \forall \ y \in \mathcal{K} \\ & -1 \leq T \leq 1 \end{aligned} \tag{3}$$

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

Solution

(a) Suppose the value of this LP is > 0 and is attained at $(\overline{w}, \overline{T})$. Then for any $y \in \mathcal{K}$, $\overline{w}^\top y - \overline{T} \leq 0$. This means that $x \notin \mathcal{K}$.
Suppose $x \notin \mathcal{K}$. Then there is a vector $w \in \mathbb{R}^n$ such that $w^\top x > 0$ and $w^\top y \leq 0 \ \forall \ y \in \mathcal{K}$. This $(w, T = 0)$ is feasible to 3 with objective > 0. Thus its optimal value is > 0.

Therefore $x \in \mathcal{K}$ iff the optimal value of 3 is ≤ 0. If ≤ 0 with optimal $w = \overline{w}$, a separating hyperplane is $\overline{w}$ because of what is discussed above.

(b)

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).

Solution

Problem 5

Solution

Problem 6

Solution