CS218 Design and analysis of algorithms **Jan-Apr 2024**

Exercises: Linear Programming, Approximation, Randomized

1. Prove that the following algorithm gives 2-approximation for minimum size vertex cover. That is, the set S output by the algorithm is a vertex cover and its size is at most twice of the optimal vertex cover.

$S \leftarrow$ empty set.

While the graph is non-empty

  choose an edge (u, v) and put both its endpoints in S

  delete u and v and all their incident edges

  delete isolated vertices

Observe that the edges chosen during the algorithm form a matching in the given graph. Prove that this is an 1/2-approximation for maximum matching. That is, the matching obtained has size at least half of the maximum size matching.

2. **Maximum weight matching:** Given a graph with edge weights, the goal is to find a matching (set of disjoint edges) with maximum total weight. Write an integer linear program for the maximum weight matching problem. Now, remove the integer constraint, that is, variables are allowed to take any real value. We get a linear program. Find an example (a graph with edge weights), where the optimal value of the linear program is higher than the weight of the maximum weight matching. Interestingly, if the graph is bipartite then the two values are always equal.

3. Recall the greedy algorithm for minimum makespan problem. Prove that if we go over the job in decreasing order of processing times, then the greedy algorithm gives a 3/2-approximate solution.

   Do you think your analysis is tight? Do you see an example, where the solution obtained is indeed 3/2 times the optimal?

4. (Line fitting.) Given n points $p_1, p_2, \dots, p_n \mathbb{R}^d$, with labels $\ell_1, \ell_2, \dots, \ell_n \in \mathbb{R}$, we want to compute a linear function that best fits with the points and labels. More precisely, find a function $h(x) = a_1x_1 + a_2x_2 + \cdots + a_dx_d + b$ so that we minimize the error function $E(h)$ defined as

$$E(h) = \max_{1 \leq j \leq n} \{|h(p_j) - \ell_j|\}.$$

   Write a linear program for this.

5. (Curve fitting.) Given n points $p_1, p_2, \dots, p_n \mathbb{R}^d$, with labels $\ell_1, \ell_2, \dots, \ell_n \in \mathbb{R}$, we want to compute a quadratic function that best fits with the points and labels. More precisely, find a function $h(x) = \sum_{1 \leq i \leq j \leq d} a_{i,j} x_i x_j$ so that we minimize the error function $E(h)$ defined as

$$E(h) = \max_{1 \leq j \leq n} \{|h(p_j) - \ell_j|\}.$$

   Write a linear program for finding h.

6. (Classification.) Given n points $p_1, p_2, \dots, p_n \mathbb{R}^d$, which are labeled either positive or negative, we want to compute a linear function that best fits with the points and labels. More precisely, find a function $h(x) = a_1x_1 + a_2x_2 + \cdots + a_dx_d + b$ so that we minimize the hing loss $L(h)$ defined as follows.

For a point p_j, if it's label is positive then we expect $h(p_j)$ to be (significantly) more than zero. Let's say we expect $h(p_j)$ to be at least 1. If that is not true then we consider the difference from 1 as the loss. Define loss with respect to a positively labeled point p_j as

$$L(h, p_j) \begin{cases} = 1 - h(p_j) & \text{if } h(p_j) < 1 \\ = 0 & \text{otherwise.} \end{cases}$$

Similarly, define loss with respect to a negatively labeled point p_k as

$$L(h, p_k) \begin{cases} = h(p_k) + 1 & \text{if } h(p_j) > -1 \\ = 0 & \text{otherwise.} \end{cases}$$

Finally we define the hinge loss $L(h)$ over all points as $\sum_j L(h, p_j)$. Write a linear program to find h that minimizes $L(h)$.