CS7800: Advanced Algorithms. Fall 2016
Homework 4

Instructor: Jonathan Ullman, TA: Mehraneh Liaee

Due Friday, Nov 18 at 11:59pm
(Email to m.liaee2050+CS7800@gmail.com)

- You must type your solutions using LaTeX. Please submit both the source and PDF files using the naming conventions lastname_hw3.tex and lastname_hw3.pdf.
- **Please put your name somewhere on the first page of your submission.**
- Strive for clarity and conciseness in your solutions, emphasizing the main ideas over low-level details. I recommend looking at the introduction in Jeff Erickson's textbook for advice on writing up solutions to algorithms problems.
- Do not share written solutions, and remember to cite all collaborators and sources of ideas. Sharing written solutions, and getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Linear Programming

Problem 1 (Linear Models via Linear Programming, 25 points).

In this problem you will see how to express various interesting problems in machine learning as linear programs.

(a) As we saw during our discussion of the segmented least-squares problem, given n points in the plane $(x_1, y_1), \ldots, (x_n, y_n)$, there is a closed-form expression for the line $y = ax + b$ that minimizes the L_2^2-error: $\sum_{i=1}^{n}(ax_i + b - y_i)^2$. However, there are other natural notions of error that we'd like to minimize, that do not have closed-form solutions, but can still be found via linear programming.

(a) Write a linear program whose solution is a line $y = ax + b$ that minimizes the L_1-error:

$$\sum_{i=1}^{n} |ax_i + b - y_i|.$$

(b) Write a linear program whose solution is a line $y = ax + b$ that minimzes the L_∞-error:

$$\max_{i=1,\ldots,n} |ax_i + b - y_i|.$$

(b) Another common problem in machine learning is to classify data using a *linear threshold function* $y = \text{sign}(a \cdot x)$, where a is a unit vector without loss of generality. Given n points $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$, we want to find a vector $a \in \mathbb{R}^d$ such that

$$\forall\, i = 1, \ldots, n \quad y_i = \text{sign}(a \cdot x_i).$$

Write a linear program that can be used to either find such a linear threshold function or determine that none exists.

In all of these problems where you are asked to write an LP, you should give a clear (but potentially informal) explanation of why your LP solves the given problem.

Problem 2 (Subcovers via Linear Programming, 25 points).

Recall the minimum cover problem: Given a collection of half-open intervals $I = \{[a_1, b_1), \ldots, [a_n, b_n)\}$, where all the endpoints are distinct, find a minimum-size subset of intervals $C \subseteq I$ that contains the unit interval $[0, 1]$.

(a) Show that for every input I, there is a finite set of points T such that for every $C \subseteq I$, C covers $[0, 1]$ if and only if C covers T. Try to make T as small as possible.

(b) Formulate the minimum subcover problem as a linear program. What are the decision variables, the objective, and the constraints? For this part, do not worry about whether or not your LP has an integral solution.

(c) Write down the dual of your LP.

(d) Give an interpretation of the dual LP. What problem is it solving?[1]

(e) Prove that both the LP and its dual have *integral* optimal solutions.[2]

[1] I recommend looking at the linear programming material in the Erickson book to see examples of problems whose natural linear programming formulations are dual.

[2] Hint: You will likely find it helpful to use the greedy algorithm from Homework 1.

NP-Completeness

Problem 3 (Maximum Saturated Flow, 25 points)**.**

We discussed at great length the fact that maximum flow problem can be solved in polynomial time. However, consider a slight modification of the maximum flow problem where we additionally require that every edge carries either no flow or is saturated. That is, for every edge e, either $f(e) = 0$ or $f(e) = c(e)$. We call such a flow *saturated*.

(a) Formulate the maximum saturated flow problem as a decision problem MaxSaturatedFlow.

(b) Prove that MaxSaturatedFlow is in *NP*.

(c) Prove that MaxSaturatedFlow is *NP*-complete.

Problem 4 (Generalized Interval Scheduling, 25 points)**.**

One of the first algorithms we saw was a greedy algorithm that solves the *interval scheduling problem*: given n intervals $(s_1, f_1), \ldots, (s_n, f_n)$, find a maximum-size subset of the intervals such that no two of the intervals in your subset overlap. We will consider the following *generalized interval scheduling problem*: given n *unions-of-intervals* $S_1, \ldots, S_n$, where

$$S_i = (s_{i1}, f_{i1}) \cup (s_{i2}, f_{i2}) \cup \cdots \cup (s_{im}, f_{im}),$$

find a maximum-size subset of the unions-of-intervals such that no two of the unions-of-intervals in your set overlap. To be clear, your set consists of entire unions-of-intervals S_i, and you cannot pick and choose from the intervals within S_i. Observe that the input size is $2nm$ numbers.

(a) Formulate this problem as a decision problem GeneralizedIntervalScheduling.

(b) Prove that GeneralizedIntervalScheduling is in *NP*.

(c) Prove that GeneralizedIntervalScheduling is *NP*-complete.