**Fall 2017 Qualifier Exam:**
**OPTIMIZATION**

**September 18, 2017**

## GENERAL INSTRUCTIONS:

1. Answer each question in a separate book.

2. Indicate on the cover of *each* book the area of the exam, your code number, and the question answered in that book. On *one* of your books list the numbers of *all* the questions answered. *Do not write your name on any answer book.*

3. Return all answer books in the folder provided. Additional answer books are available if needed.

## SPECIFIC INSTRUCTIONS:

Answer all 4 questions.

## POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tries to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced a problem has been stated incorrectly, mention this to the proctor. If necessary, the proctor can contact a representative of the area to resolve problems during the *first hour* of the exam. In any case, you should indicate your interpretation of the problem in your written answer. Your interpretation should be such that the problem is nontrivial.

1. **Linear Programming**

   Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ and $Q = \{x \in \mathbb{R}^n : Cx \leq d\}$ be two non-empty polyhedra.

   (a) Write a linear programming formulation that solves the problem:
   $$\min\{\|x - y\|_1 : x \in P, y \in Q\}$$
   where $\|z\|_1 = \sum_{i=1}^{n} |z_i|$ is the 1-norm.

   (b) Write the dual of the formulation you wrote in part (a).

   (c) Justify that both the primal and dual problems have an optimal solution (you may use the strong duality theorem).

   (d) Using the above primal/dual pair of linear programs, show that if $P \cap Q = \emptyset$, then there exists a vector $p \in \mathbb{R}^n$ such that $p^\top x < p^\top y$ for all $x \in P$ and $y \in Q$. [Hint: the vector $p$ can be defined using an optimal dual solution. ]

---

2. **Modeling**

   It is common knowledge that words/objects/entities have color associations. For example, *anger* is often associated with the color red. These associations are not one-to-one mappings, e.g. *strawberry* is also associated with the color red. The associations are not unique either; *apple* can be associated with red or green, and if we're talking about the company Apple Inc., the associations will be different still!

   You are given a bar graph where each bar represents a different entity and your task is to choose colors to use for each of the bars. For example, the graph might look like the one below:
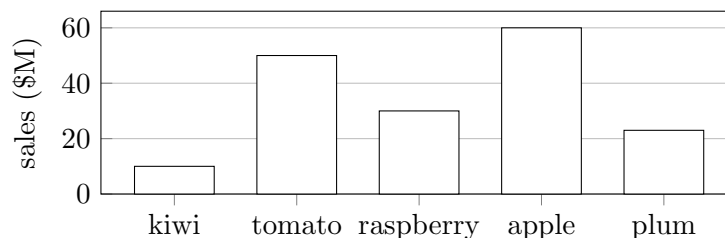


   Your task is to choose colors for the bars in the graph so that each chosen color has a strong association with the category it represents. Suppose the labels for the bars in the graph are $\{b_1, \ldots, b_m\}$ and the colors at your disposal are $\{c_1, \ldots, c_n\}$. You have access to a dataset of color-category association strengths. The data is in the form of a table:

| Category \ Color | $c_1$ | $c_2$ | $\cdots$ | $c_n$ |
|---|---|---|---|---|
| $b_1$ | $a_{11}$ | $a_{12}$ | $\ldots$ | $a_{1n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $b_m$ | $a_{m1}$ | $a_{m2}$ | $\ldots$ | $a_{mn}$ |

So $a_{ij}$ is association strength between category $b_i$ and color $c_j$. We'll assume all the data are normalized so that $0 \le a_{ij} \le 1$ and $\sum_j a_{ij} = 1$. In other words, you can think of the $i^{\text{th}}$ row of the table as a distribution over colors for the category $b_i$. We'll assume $n \ge m$, so there are more colors than categories.

(a) Suppose we would like to assign the colors to the categories in a way that maximizes the total association strength of all pairs. For example, if we associate $b_1$ with $c_6$ and $b_2$ with $c_1$, then the total association strength is $a_{16} + a_{21}$. Note: you cannot assign the same color to two different categories. Formulate this optimization problem as a linear program that depends on the data $\{a_{ij}\}$. Be sure to explain why your model is correct and describe the variables, constraints, and objective function.

(b) The approach of minimizing total association strength doesn't work as well when several categories all have similar color association profiles. Instead, we'll look for a way to assign colors to categories such that the chosen pair has a high association strength and the non-chosen pairs have a low association strength. To this effect, we modify the objective such that, if color $c_j$ is associated with category $b_i$, this contributes $h_{ij}$ to the objective, where

$$h_{ij} = a_{ij} - \tau \max_{k \ne i} a_{kj}$$

Here, $\tau \ge 0$ is a parameter and the max is taken over all $k \in \{1, \ldots, i-1, i+1, \ldots, m\}$. The net effect of using such an objective is that $b_i$ should be strongly associated with $c_j$ and at the same time, $c_j$ should not be strongly associated with any of the other $b'_k s$ for $k \ne i$. How should you modify your linear program to account for this new objective?

(c) Picking a different $\tau$ in the formula for $h_{ij}$ generally leads to a different optimal assignment of colors to categories. Prove that when $m = 2$ and $n = 2$ (two categories and two colors), all values of $\tau \ge 0$ lead to the same solution.

---

3. **Integer Optimization**

Given an undirected graph $G = (V, E)$ and a weight function $w : E \to \mathbb{R}$ on the edges, a *matching* $M \subseteq E$ is a subset of pairwise disjoint edges of $G$ (i.e., every node of $G$ is contained in at most one edge of $M$). The weight $w(M)$ of a matching $M$ is defined as the sum of the weights of the edges in $M$, namely

$$w(M) = \sum_{e \in M} w_e.$$

In this setting the *maximum weight matching problem* asks to find a matching in the graph with maximum weight.

(a) Explain how the maximum weight matching problem can be solved in polynomial time if $G$ is bipartite.

The *greedy algorithm* for the maximum weight matching problem proceeds as follows:

- Set $M := \emptyset$.
- Set $A := E$.
- While $A \neq \emptyset$, do:
  - Let $e$ be an edge in $A$ with highest weight.
  - Add $e$ to $M$.
  - Remove from $A$ all edges adjacent to $e$.
- Return $M$.

(b) Show an example in which the greedy algorithm does not find a matching with maximum weight.

(c) We now consider a restricted version of the maximum weight matching problem in which the weights of all edges are 1, hence a maximum matching $M$ is simply a matching with maximum cardinality $|M|$. Notice that the greedy algorithm in this case chooses an arbitrary edge from $A$ in every iteration.

Let $OPT$ be the cardinality of the optimal solution and let $M_g$ be the output of the greedy algorithm. Show that $\frac{|M_g|}{OPT} \geq 0.5$ (In other words, the matching that the greedy algorithm finds is at least half the size of an optimum one).

(Hint: Consider the relationship between the edges in a greedy matching and those in an optimal matching.)

(d) For every $n \in \mathbb{Z}_+$ give a graph with at least $n$ vertices for which the greedy algorithm could possibly yield a matching with $\frac{|M_g|}{OPT} = 0.5$.

---

4. **Nonlinear Optimization**

(a) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function such that $f(x) \geq 0$ for all $x \in \mathbb{R}^n$. Define the function $g(x) = \frac{1}{2}f(x)^2$, and consider the following two problems:

$$\min_x \ f(x), \tag{F}$$

$$\min_x \ g(x). \tag{G}$$

Verify that the first order necessary conditions for these two problems are equivalent, that is, $x^*$ satisfies first-order necessary conditions for (F) if and only if $x^*$ satisfies first-order necessary conditions for (G). (Hint: Consider the case of $f(x^*) = 0$ carefully.)

(b) Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has Lipschitz continuous gradient, that is, there is $L > 0$ such that

$$\|\nabla f(y) - \nabla f(z)\|_2 \le L\|y - z\|_2 \quad \text{for all } y, z \in \mathbb{R}^n.$$

Suppose in addition that $f(x) \ge \bar{f}$ for all $x$, and for some $\bar{f} > -\infty$. Consider the following short-step steepest descent method:

$$x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k), \quad k = 0, 1, 2, \ldots.$$

Show that the following three inequalities hold:

$$f(x^{k+1}) \le f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|_2^2, \quad k = 0, 1, 2, \ldots,$$

$$\sum_{k=0}^{T-1} \|\nabla f(x^k)\|_2^2 \le 2L\left[f(x^0) - \bar{f}\right], \quad \text{for all } T \ge 1,$$

$$\min_{k=0,1,\ldots,T-1} \|\nabla f(x^k)\|_2 \le \sqrt{\frac{2L[f(x^0) - \bar{f}]}{T}}, \quad \text{for all } T \ge 1.$$

Cite explicitly any theorems you use in proving these results. (Hint: Prove these three inequalities in sequence, using each one to prove the next in the sequence.)