CS 170 HW 9

Due on 2019-04-01, at 11:59 pm

1 Study Group

List the names and SIDs of the members in your study group.

2 Modeling: Tricks of the Trade

One of the most important problems in the field of *statistics* is the *linear regression problem.* Roughly speaking, this problem involves fitting a straight line to statistical data represented by points – $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ – on a graph. Denoting the line by $y = a + bx$, the objective is to choose the constants a and b to provide the "best" fit according to some criterion. The criterion usually used is the *method of least squares*, but there are other interesting criteria where linear programming can be used to solve for the optimal values of a and b. For each of the following criteria, formulate the linear programming model for this problem:

1. Minimize the sum of the absolute deviations of the data from the line; that is,

$$\min \sum_{i=1}^{n} |y_i - (a + bx_i)|$$

2. Minimize the maximum absolute deviation of the data from the line; that is,

$$\min \max_{i=1 \ldots n} |y_i - (a + bx_i)|$$

3 Zero Sum Games

Alice and Bob are playing a zero-sum game whose payoff matrix is shown below. The ij^{th} entry of the matrix shows the payoff that Alice receieves if she plays strategy i and Bob plays strategy j. Alice is the row player and is trying to maximize her payoff.

Alice \Bob	A	B
1	4	1
2	2	5

Now we will write a linear program to find a strategy that maximizes Alice's payoff. We consider Alice's strategy to be a probabilistic one. The variables of the linear program are x_1, x_2 are the probabilities of Alice picking strategy 1 and 2 respectively (So obviously $x_1 + x_2 = 1$) and p is the value of Alice's payoff.

(a) Write the linear program for maximizng Alice's payoff. (Hint : You should think of setting up the constraints of the program such that it finds the best worst case strategy. This would depend on the strategy Bob plays assuming he knows what Alice's (probabilistic) strategy is.)

(b) Eliminate x_2 from the linear program and write it in terms of p and x_1 alone.

(c) Draw the feasible region of the above linear program in p and x_1. You are encouraged to use a plotting software for this.

(d) What is the optimal solution and what is the value of the game?

4 Repairing a Flow

In a particular network $G = (V, E)$ whose edges have integer capacities c_e, we have already found a maximum flow f from node s to node t where f_e is an integer for every edge. However, we now find out that one of the capacity values we used was wrong: for edge (u, v) we used c_{uv} whereas it should have been $c_{uv} - 1$. This is unfortunate because the flow f uses that particular edge at full capacity: $f_{uv} = c_{uv}$. We could redo the flow computation from scratch, but there's a faster way.

Describe an algorithm to repair the max-flow in $O(|V| + |E|)$ time. Also give a proof of correctness and runtime justification.

5 Generalized Max Flow

Consider the following generalization of the maximum flow problem.

You are given a directed network $G = (V, E)$ where edge e has capacity c_e. Instead of a single (s, t) pair, you are given multiple pairs $(s_1, t_1), ..., (s_k, t_k)$, where the s_i are sources of G and t_i are sinks of G. You are also given k (positive) demands $d_1, ..., d_k$. The goal is to find k flows $f^{(1)}, ..., f^{(k)}$ with the following properties:

(a) $f^{(i)}$ is a valid flow from s_i to t_i.

(b) For each edge e, the total flow $f_e^{(1)} + f_e^{(2)} + ... + f_e^{(k)}$ does not exceed the capacity c_e.

(c) The size of each flow $f^{(i)}$ is at least the demand d_i.

(d) The size of the *total* flow (the sum of the flows) is as large as possible.

Write a linear problem using the variables $f_e^{(i)}$ whose optimal solution is exactly the solution to this problem. For each constraint as well as the objective in your linear program briefly explain why it is correct. (Note: Since linear programs can be solved in polynomial time, this implies a polynomial-time algorithm for the problem)

6 Reductions Among Flows

Show how to reduce the following variants of Max-Flow to the regular Max-Flow problem, i.e. do the following steps for each variant: Given a directed graph G and the additional variant constraints, show how to construct a directed graph G' such that

(1) If F is a flow in G satisfying the additional constraints, there is a flow F' in G' of the same size,

(2) If F' is a flow in G', then there is a flow F in G satisfying the additional constraints with the same size.

Prove that properties (1) and (2) hold for your graph G'.

(a) **Max-Flow with Vertex Capacities:** In addition to edge capacities, every vertex $v \in G$ has a capacity c_v, and the flow must satisfy $\forall v : \sum_{u:(u,v)\in E} f_{uv} \leq c_v$.

(b) **Max-Flow with Multiple Sources:** There are multiple source nodes $s_1, \ldots, s_k$, and the goal is to maximize the total flow coming out of all of these sources.

7 A Flowy Metric

Consider an undirected graph G with capacities $c_e \geq 0$ on all edges. G has the property that any cut in G has capacity at least 1. For example, a graph with a capacity of 1 on all edges is connected if and only if all cuts have capacity at least 1. However, c_e can be an arbitrary nonnegative number in general.

1. Show that for any two vertices $s, t \in G$, the max flow from s to t is at least 1.

2. Define the *length* of a flow f to be $\texttt{length}(f) = \sum_{e\in G} |f_e|$. Define the *flow distance* $d_{\texttt{flow}}(s,t)$ to be the minimum length of any $s-t$ flow f that sends one unit of flow from s to t and satisfies all capacities; i.e. $|f_e| \leq c_e$ for all edges e.

   Show that if $c_e = 1$ for all edges e in G, then $d_{\texttt{flow}}(s,t)$ is the length of the shortest path in G from s to t.

   (*Hint*: Let $d(s,t)$ be the length of the shortest path from s to t. A good place to start might be to first try to show $d_{\texttt{flow}}(s,t) \leq d(s,t)$. Then try to show $d_{\texttt{flow}}(s,t) \geq d(s,t)$)

3. **(Optional)** The shortest path satisfies the *triangle inequality*, that is for three vertices, s, t, and u in G, if $d(x,y)$ is the length of the shortest path from x to y, then $d(s,t) \leq d(s,u) + d(u,t)$. Show that the triangle inequality also holds for the flow distance. That is; show that for any three vertices $s, t, u \in G$

   $$d_{\texttt{flow}}(s,t) \leq d_{\texttt{flow}}(s,u) + d_{\texttt{flow}}(u,t)$$

   even when the capacities are arbitrary nonnegative numbers.