CS170–Fall 2022 — Homework 8 Solutions

Deming Chen cdm@pku.edu.cn

Last Modified: December 28, 2022

Collaborators: NONE

2 LP Meets Linear Regression

Consider a linear program with $(n+2)$ variables: $a, b, z_1, z_2, \cdots, z_n$. The objective function is

$$\min\ z_1 + z_2 + \cdots + z_n. \tag{1}$$

The constraints are:

$$\begin{aligned} z_i &\geqslant y_i - (a + bx_i), i = 1, 2, \cdots, n; \\ z_i &\geqslant (a + bx_i) - y_i, i = 1, 2, \cdots, n. \end{aligned} \tag{2}$$

We now prove the linear program above solves the linear regression problem. For arbitrary a and b, let $\tilde{z}_i = |y_i - (a + bx_i)|, i = 1, 2, \cdots, n$. Apparently, these $\tilde{z}_i$ and a, b satisfy constrains (2). So, $\sum_{i=1}^{n} \tilde{z}_i = \sum_{i=1}^{n} |y_i - (a + bx_i)|$ implies that $\min \sum_{i=1}^{n} z_i \leqslant \sum_{i=1}^{n} |y_i - (a + bx_i)|$. Since a, b are arbitrarily selected,

$$\min \sum_{i=1}^{n} z_i \leqslant \min \sum_{i=1}^{n} |y_i - (a + bx_i)|. \tag{3}$$

On the other side, for arbitrary $a, b, z_1, z_2, \cdots, z_n$ which satisfy (2), $\sum_{i=1}^{n} z_i \geqslant \sum_{i=1}^{n} |y_i - (a + bx_i)|$. So, $\sum_{i=1}^{n} z_i \geqslant \min \sum_{i=1}^{n} |y_i - (a + bx_i)|$. Since $a, b, z_1, z_2, \cdots, z_n$ are arbitrarily selected,

$$\min \sum_{i=1}^{n} z_i \geqslant \min \sum_{i=1}^{n} |y_i - (a + bx_i)|. \tag{4}$$

In conclusion,

$$\min \sum_{i=1}^{n} z_i = \min \sum_{i=1}^{n} |y_i - (a + bx_i)|. \tag{5}$$

3 Flow vs LP

(a) Define a graph of $(m+n+2)$ vertices $G = (V, E)$, where $V = \{s, t, u_1, u_2, \cdots, u_m, v_1, v_2, \cdots, v_n\}$. $u_1, u_2, \cdots, u_m$ represents m suppliers, and $v_1, v_2, \cdots, v_n$ represents n purchasers. There are edges from s to all u_is and edges from all v_is to t. $(u_i, v_j) \in E$ if and only if supplier i is within 1000 miles of purchaser j. The capacity of edges between u_i and v_j is ∞. The capacity of the edge from s to u_i equals $s[i]$, and the capacity of the edge from v_j to t equals $b[j]$:

$$c_{su_i} = s[i],\ i = 1, 2, \cdots, m$$
$$c_{v_jt} = b[j],\ j = 1, 2, \cdots, n.$$

The flow on edge (u_i, v_j) represents the amount of product supplier i sells to purchaser j. Since the flow entering u_i is less than $c_{su_i} = s[i]$, so the total amount of product supplier i sells is less than $s[i]$. Similarly, the flow leaving v_j is less than $c_{v_jt} = b[i]$, so the total amount of products purchase j buys is less than $b[j]$. Therefore, the output from the network flow algorithm gives a valid solution to this problem.

(b) If if supplier i is within 1000 miles of purchaser j, we introduce a variable x_{ij}, which represents the amount of products supplier i sells to purchaser j. So, our objective is

$$\max \sum_{(i,j)\in L} x_{ij}. \tag{6}$$

The constrains are

$$\begin{aligned} &\sum_j x_{ij} \leqslant s[i],\ \ i = 1, 2, \cdots, m; \\ &\sum_i x_{ij} \leqslant b[j],\ \ j = 1, 2, \cdots, n; \\ &x_{ij} \geqslant 0,\ \ (i, j) \in L. \end{aligned} \tag{7}$$

4 A Cohort of Secret Agents

Without loss of generality, suppose no agents starts at a point in T. Construct a graph $\tilde{G} = (\tilde{V}, \tilde{E})$. Let's denote $V \backslash T$ by $\{v_1, v_2, \cdots, v_m\}$. Then $\tilde{V} = \{s, t, u_1, u_2, \cdots, u_m, w_1, w_2, \cdots, w_m\}$. For each i, $(u_i, w_i) \in \tilde{E}$. $(s, u_i) \in \tilde{E}$ if and only if some agents starts at v_i. $(w_i, t) \in \tilde{E}$ if and only if for some vertex $x \in T$, $(v_i, x) \in E$. So, $t \in \tilde{V}$ represents all points in T. For each $i \neq j$, $(w_i, u_j) \in \tilde{E}$ if and only if $(v_i, v_j) \in E$.

There is a one-to-one mapping between a path in G from a vertex where some agent starts to a vertex in T and a path in $\tilde{G}$ from s to t. If a path passes through a vertex $v_i \in V \backslash T$, then its corresponding path in $\tilde{G}$ passes through edge (u_i, w_i).

Now, let's set edges' capacities: $c(s, \cdot) = 1$, $c(u_i, w_i) = c$, and other edges' capacities are infinity. If the max flow from s to t in $\tilde{G}$ has size k, then their corresponding paths in G is the solution to the original problem.

5 Applications of Max-Flow Min-Cut

(a) Suppose G has a L-perfect matching. For $X \subset L$, consider vertices in R which match vertices in X in that L-perfect matching. There are exactly $|x|$ such vertices, and they are all connected to some vertex in X. So, X is connected to at least $|X|$ vertices in R.

We prove the other direction by induction on $|L|$. The base case is when $|L| = 1$. The only vertex in L must be connected to some vertex in R, since G has a L-perfect matching.

Now, suppose $|L| = k > 1$. Let's take G as a directed graph, where all edges directed from a vertex in L to a vertex in R. Besides we add two vertices s, t and some edges: from s to all vertices in L, and from all vertices in R to t. The capacity on edges from a vertex in L to a vertex in R is infinity, and the capacity on other edges is 1. For $Y \subset L$, let $\Gamma(Y)$ denote vertices in R connected to some vertex in Y, i.e. $\Gamma(Y) = \{v \in R | \exists u \in Y \text{ s.t.} (u, v) \in E\}$.

Consider the maximum flow f_m from source s to sink t. Since each edge has an integer capacity, we can pick f_m such that the flow on each edge is an integer. If the size of f_m is $|L| = k$, then all edges directed from s is saturated. The flow on an edge between L and R is either 0 or 1 (k of such edges have 1 unit of flow and others have 0 unit of flow). Those edges with 1 unit of flow share no common vertex, since edges directed from s and directed to t can handle up to 1 unit of flow. This forms an L-perfect matching. If the size of f_m is less than $|L|$, then $\{s\}$ and $L \cup R \cup \{t\}$ is not a minimum cut of G. Consider the minimum cut $V = S \cup T$, where $s \in S$ and $t \in T$. Let $X = S \cap L \neq \varnothing$. Since edges between L and R have infinity capacities, $\Gamma(X) \subset S$. Since $|\Gamma(L)| \geqslant |L|$, $X \neq L$, otherwise the minimum cut is larger than $|\Gamma(L)|$. So, $X \cup \Gamma(X) \subset S$, and $(L \backslash X) \cup \Gamma(L \backslash X) \subset T$. Since $S \cap T = \varnothing$, $\Gamma(X) \cap \Gamma(L \backslash X) = \varnothing$. We can apply the induction hypothesis on the subgraph $(X, \Gamma(X))$ and $(L \backslash X, \Gamma(L \backslash X))$. There is a matching with size $|X|$ between X and $\Gamma(X)$, and a matching with size $|L \backslash X|$ between $L \backslash X$ and $\Gamma(L \backslash X)$. So, combining these two matches we get an L-perfect matching.

(b) Assign each edge a unit capacity so that G forms a network flow.

First, we claim that for any integer flow of non-zero size, we can find a path from s to t such that every edge in the path has non-zero flow. Since the capacity of each edge is 1, the flow on it is either 0 or 1. So, except for s and t, the number of edges directed to any vertex with unit flow equals the number of edges directed from that vertex with unit flow. Consider the subgraph composed of edges with unit flow. Then, s and t are connect, otherwise some vertex other than s or t has odd degree. So, there is a path from s to t in the subgraph.

Now, we show that the maximum number of edge-disjoint s-t paths equals the size of the maximum flow from s to t. Consider a maximum flow f where flow on each edge is an integer. By the proposition we proved above, this flow can be decomposed to $\texttt{size}(f)$ disjoint paths, which implies that the maximum number of edge-disjoint s-t paths is no less than $\texttt{size}(f)$. For any k edge-disjoint paths, set flow on each edge as 1 if it belongs to any of these paths. Obviously, this forms a s-t network flow of size k. So, the maximum number of edge-disjoint s-t paths is no larger than the size of the maximum flow from s to t. Therefore, these two numbers are equal.

At last, we prove that there is a one-to-one mapping from a set of edges whose removal disconnects t from s to a cut of this network. By the definition of cut, removing edges from s part to t part disconnects t from s. Besides, for a one-to-one mapping from a set of edges whose removal disconnects t from s, let S be the set of vertices which can be reached from s. Then S and $V \backslash S$ form a cut, and E' is just the set of edges from S to $V \backslash S$.

By the max-flow min-cut theorem, as a result, the maximum number of edge-disjoint s-t paths equals the minimum number of edges whose removal disconnects t from s.