Linear Regression 1

Ted Tinker, 3223468

April 6, 2017

Part One

In this situation, using the variables $\{x_1, \ldots, x_5\}$ to represent the servings of carrots, potatoes, bread, cheese, and peanut butter purchased (respectively), we seek to minimize $.14x_1 + .12x_2 + .2x_3 + .75x_4 + .15x_5$, with variables constrained by $23x_1 + 171x_2 + 65x_3 + 112x_4 + 188x_5 \geq 2000$, $.1x_1 + .2x_2 + 9.3x_4 + 16x_5 \geq 50$, $.6x_1 + 3.7x_2 + 2.2x_3 + 7x_4 + 7.7x_5 \geq 100$, and $6x_1 + 30x_2 + 13x_3 + 2x_5 \geq 250$, with $x_i \geq 0$ for all i (as we cannot sell food back).

To convert this to a more standard form, we rephrase the minimizing problem as a maximizing problem and reorient the inequalities:

$$\text{Maximize } -.14x_1 - .12x_2 - .2x_3 - .75x_4 - .15x_5, \text{ constrained by}$$

$$-23x_1 - 171x_2 - 65x_3 - 112x_4 - 188x_5 \leq -2000,$$

$$-.1x_1 - .2x_2 - 9.3x_4 - 16x_5 \leq -50,$$

$$-.6x_1 - 3.7x_2 - 2.2x_3 - 7x_4 - 7.7x_5 \leq -100, \text{ and}$$

$$-6x_1 - 30x_2 - 13x_3 - 2x_5 \leq -250, \text{ with } x_i \geq 0 \text{ for all } i.$$

This linear programming solver website (http://comnuan.com/cmnn03/cmnn03004/) states that the optimal decision is to buy 7.7147 servings of baked potatoes and eat them with 9.2800 servings of peanut butter, for a cost of \$2.304. Tallying the nutrition facts for this purchase, we find that it just barely satisfies each requirement.

Part Two

A. This calls for minimizing $cost = T+P+N+S+G+L+B$, where each variable is the salary of the employee whose name begins with that letter (or, in a standard format, maximizing $-cost = -T-P-N-S-G-L-B$). We don't pay employees negative salaries, so each variable should be greater than or equal to 0. The other constrictions can be abstracted as follows (in standard form):

$$-T \leq -20,000$$

$$T-P, T-N, \text{ and } T-S \leq -5,000$$

$$T+P-G \leq 0$$

$$G-L \leq -200$$

$$2T+2P-N-S \leq 0$$

$$P-B \leq 0$$

$$S-B \leq 0$$

$$-B-P \leq -60,000$$

$$L-B-T \leq 0$$

On Comnuan.com again, I found that Tom should be paid \$20,000, Peter \$25,000, Nina \$65,000, Samir \$25,000, Gary \$45,000, Linda \$45,200, and Bob \$35,000, at a total cost of \$260,200.

B. Use the above constraints, but optimize not the total cost, but a new variable H. Make new constraints $T-H \leq 0$, $P-H \leq 0$, and so on, so that H minimized is the maximum salary.

In this schema, Tom should be paid \$20,000, Peter \$25,000, Nina \$45,200, Samir \$44,800, Gary \$45,000, Linda \$45,200, and Bob \$45,200, at a total cost of \$270,400, but with a maximum salary of \$45,200.

Part Three

Define $x_1, \ldots, x_6$ to be the number of liters of Chemical X sent to FRESHAIR after the first, ..., sixth hour. We want to minimize $cost = 30x_1 + 40x_2 + 35x_3 + 45x_4 + 38x_5 + 50x_6$ (or, in standard format, maximize $-cost = -30x_1 - 40x_2 - 35x_3 - 45x_4 - 38x_5 - 50x_6$). Each $x_i \geq 0$, as CRUD isn't buying any Chemical X from FRESHAIR. We cannot ship more Chemical X than we have at any time, so

$$x_1 \leq 300$$

$$x_1 + x_2 \leq 540$$

$$x_1 + x_2 + x_3 \leq 1,140$$

$$x_1 + x_2 + x_2 + x_4 \leq 1,340$$

$$x_1 + x_2 + x_2 + x_4 + x_5 \leq 1,640$$

Because of the 1,000 liter limit,

$$-x_1 - x_2 - x_3 \leq -140$$

$$-x_1 - x_2 - x_3 - x_4 \leq -340$$

$$-x_1 - x_2 - x_3 - x_4 - x_5 \leq -640$$

$$-x_1 - x_2 - x_3 - x_4 - x_5 - x_6 \leq -2,540$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2,540$$

(The last two constraints ensure that at the end of the day, no Chemical X remains.) Solving this, we find that we should ship Chemical X at the following rates:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \text{300 liters} \\ \text{0 liters} \\ \text{840 liters} \\ \text{0 liters} \\ \text{500 liters} \\ \text{900 liters} \end{bmatrix}$$

for a total cost of \$94,300.

Part Four

A. We want to minimize the cost (or maximize the inverse cost), which is equal to

$$\sum_{i \in I, j \in J} c_{i,j} x_{i,j}.$$

That is, the total cost is the price to ship a unit from factory $i \in I = \{0, \ldots, p\}$ to store $j \in J = \{0, \ldots, q\}$ times the number of units shipped from factory i to store j, for all combinations of i and j. To model the problem, the constraints should be

$$x_{i,j} \geq 0 \quad \forall i, j \in I, J \text{ (because no units are shipped backwards from store to factory)}$$

$$\sum_{j \in J} x_{i,j} = s_i \quad \forall i \in I \text{ (because factory } i \text{ makes } s_i \text{ units)}$$

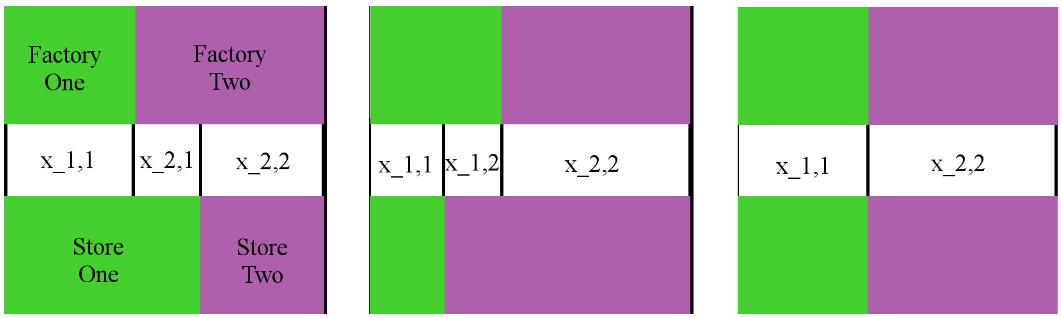
$$\sum_{i \in I} x_{i,j} = t_j \quad \forall j \in J \text{ (because store } j \text{ receives } t_j \text{ units)}$$

B. If $\sum_{i \in I} s_i < \sum_{j \in J} t_j$, then the number of units produced is smaller than the number of units requested. Hence, for at least one value of $j \in J$, $\sum_{i \in I} x_{i,j} < t_j$, contradicting the third constriction; there are not enough units to fulfil all orders.

Similarly, if $\sum_{i \in I} s_i > \sum_{j \in J} t_j$, then too many units are produced. After fulfilling all orders, we erroneously ship the remaining $\sum_{i \in I} s_i - \sum_{j \in J} t_j$ units (as the problem states "assume that every unit is shipped to a store"), so for at least one value of $j \in J$, $\sum_{i \in I} x_{i,j} > t_j$, contradicting the third constriction. Then, if the two sums are unequal, no points can be feasible solutions to the Linear Program.

If the two summations are equal, then we may declare shipping orders to satisfy all constrictions; this may not be an optimal solution, but it does demonstrate that the feasible region must be non-empty. Begin with the simple case in which there are two factories and two stores. Declare $x_{1,1} = min(s_1, t_1)$; then, either factory 1 has shipped all of its units (in the case $s_1 < t_1$), or store 1 has received all of its requested units (in the case $s_1 > t_1$), or both (in the case $s_1 = t_1$).

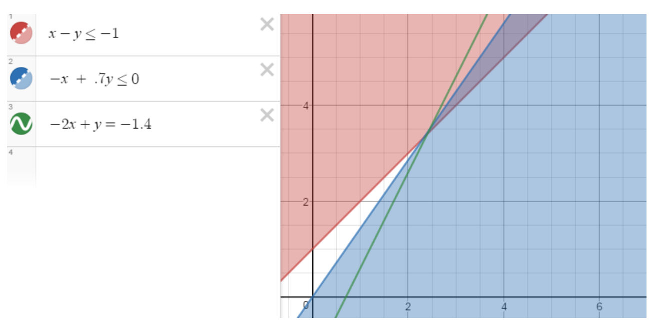
In the first case, define $x_{2,1} = t_1 - s_1$, $x_{2,2} = t_2$. We know that factory 2 has precisely $x_{2,1} + x_{2,2}$ units available because of the equality $\sum_{i \in I} s_i = \sum_{j \in J} t_i$. The equality also ensures that in the second case, we may define $x_{1,2} = s_1 - t_1$, and then $x_{2,2} = t_2 - x_{1,2}$. Of course, in the third case, we declare $x_{2,2} = t_2$.



Now, for p factories and q stores, we may rephrase the problem into the 'two factory, two store' state by conjoining factories 2 through p into one factory and stores 2 through q into one store. This resolves either factory 1's output, store 1's input, or both, and the remaining factories and stores can be iteratively dealt with in the same manner. The equality $\sum_{i \in I} s_i = \sum_{j \in J} t_i$ assures that there are valid solutions in the feasible area for us to find.

Part Five

A. When I experiment with the feasible area graphically, I found this graph enlightening:



For a feasible area to be non-empty, the lines $x_1 - x_2 = -1$ and $-x_1 + tx_2 = 0$ must intersect in the upper-right quadrant or on its boundary, to satisfy $x_1, x_2 \geq 0$. This occurs when $0 \leq t < 1$. However, if t makes $-x_1 + tx_2 = 0$ steeper than the objective function, then the feasible area will be unbounded in the sense that we may find arbitrarily high values of the objective function. Therefore, the range of solutions is $T = [.5, 1)$.

B. The two lines intersect at $\left(\frac{-t}{t-1}, \frac{-1}{t-1}\right)$. The values of t which place this point the upper-right quadrant, or on its border, are $[0, 1)$. Part D further removes the interval $[0, .5)$ from consideration, as these values cause the feasible region to be unbounded, leaving the solutions $t \in T = [.5, 1)$.