Linear Programming

The Simplex Algorithm

Nick Morris

04/04/2016

README

Code

The code is in the simplx_HW 3.R file provided. I suggest opening this file in notepad ++, go to Language → R → R and this will highlight key words and comments to make reading the code easier. Open up R and then go back to notepad++, highlight the entire script, and then copy and paste it into the "R Console" window in R. This will create the function "simplx()" in R.

Strings, Vectors, & Matrices

The function simplx() requires you to enter a string, 4 vectors, and a matrix as inputs. The way to write a string in R is with quotes (ie. "min", "max", etc). There are two R commands to know, to create the vectors and matrix. The command to create a vector is "`c(e1, e2, …, en)`" Below is a vector with 7 elements. All row vectors and column vectors should be entered this way, and the code will adjust them into row and column vectors.

```
c(1, 2, 3, 4, 7, 44, 88)
```

The commands to create a matrix are "`rbind(r1, r2, … , rn)`" and "`c(e1, e2, …, en)`". Below is a [3 x 7] matrix. The command rbind() binds vectors by rows, so the first vector below: `c(1, 2, 3, 4, 7, 44, 88)` is the first row vector in the matrix and the third vector below: `c(1, 2, 3, 4, 5, 6, 7)` is the third and last row vector in the matrix.

```
rbind(c(1, 2, 3, 4, 7, 44, 88), c(4, 6, 2, 8, 4, 9, 2), c(1, 2, 3, 4, 5, 6, 7))
```

An Example

Figure 1 below is a maximization problem in standard form.

$$
\begin{array}{l}
Max \quad Z = 60x_1 + 30x_2 + 20x_3 \\
s.t. \\
\qquad 8x_1 + 6x_2 + x_3 + s_1 = 48 \\
\qquad 4x_1 + 2x_2 + 1.5x_3 + s_2 = 20 \\
\qquad 2x_1 + 1.5x_2 + 0.5x_3 \quad s_3 = 8 \\
\qquad x_2 \quad s_4 = 5 \\
\qquad x_1, x_2, x_3, s_1, s_2, s_3, s_4 \geq 0
\end{array}
$$

Figure 1: Linear Program in Standard Form

The inputs for simplx() would be built the following way:

```
# vector of objective function coefficients
C = c(60, 30, 20, 0, 0, 0, 0)
```

```
# matrix of constraint coefficients
A = rbind(c(8, 6, 1, 1, 0, 0, 0), c(4, 2, 1.5, 0, 1, 0, 0), c(2, 1.5, 0.5, 0, 0, 1, 0),
c(0, 1, 0, 0, 0, 0, 1))

# vector of right hand side scalars
b = c(48, 20, 8, 5)

# vector of a basic feasible solution
x = c(0, 0, 0, 48, 20, 8, 5)

# vector of 1's and 0's indicating a variable is or isn't in the basis respectively
basis = c(0, 0, 0, 1, 1, 1, 1)
```

This problem would be solved with simplx() in the following way:

```
simplx(type = "max", C = C, A = A, b = b, x = x, basis = basis)

$iterations
[1] 2

$objective
[1] 280

$solution
  x1 x2 x3 x4 x5 x6 x7
1  2  0  8 24  0  0  5

$table
  it obj step    d1 d2 d3 d4   d5 d6 d7 x1 x2 x3 x4 x5 x6 x7
1  0   0    0  0.00  0  0  0  0.0  0  0  0  0  0 48 20  8  5
2  1 240    4  1.00  0  0 -8 -4.0 -2  0  4  0  0 16  4  0  5
3  2 280    8 -0.25  0  1  1 -0.5  0  0  2  0  8 24  0  0  5
```

The output of the solution above shows that the problem required two simplex iterations to be solved and the maximum value of the objective function while respecting the constraints is 280. The solution is given where {x1 x2 x3} correspond to {x1 x2 x3} as seen in Figure 1, whereas {x4 x5 x6 x7} in the solution correspond to {s1 s2 s3 s4} as seen in Figure 1. The table in the solution output shows simplex iterations across rows. The columns include the iteration identity "it", the value of the objective function "obj" achieved from that iteration, the value of the step size "step" used to achieve the "obj" value, the direction vector which contains columns "d1" to "d7" used to achieve the "obj" value, and the solution vector which contains columns "x1" to "x7" used to achieve the "obj" value.