Optimization Theory (DS2) Lecture #5 Sticking a Stake in the Heart of the Simplex Algorithm

November 7, 2016

Abstract

Today we are going to finish off the simplex algorithm, Sec 2.5 of the textbook.

Text and examples adapted from the paperback edition of A Gentle Introduction to Optimization*, B. Guerin* et al.

1 Review

1.1 Random Comment

I finally got around to looking at the Wikipedia page on the simplex algorithm in detail. In the English page, I like the explanation of the history, and I like the images, but the explanation of tableaus and pivots is not quite the way we're solving things, so I wouldn't invest too much time in trying to understand that.

The Japanese page on the simplex algorithm is actually poor. *I would* **gladly** *accept a student rewriting that Wikipedia page as an end-of-term project!* In fact, I encourage it! Any volunteers?

1.2 The Polytope

Did you try out the homework? Any comments on it?

Note that each inequality cuts the space in half, and therefore the polytope must be *convex*.

1.3 Interlude: a Possibly Useful Trick in R

The following ad hoc R code will find a solution to a set of inequalities for you. Learn how to use it if you like, but it's up to you, I'm no help here.

```
> library(Rglpk)
Loading required package: slam
```

```
Using the GLPK callable library version 4.55
> rhs <- rep(c(0, 1e-3), c(9,3))
> rhs
 [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.001 0.001
> ge <- rep(">=", 12)
> ge
 [1] ">=" ">=" ">=" ">=" ">=" ">=" ">=" ">=" ">=" ">=" ">=" ">="
> set.seed(123)
> A <- rbind(matrix(runif(27, -0.5, 0.5), nc = 3), diag(3))
>
> A
             [,1]        [,2]        [,3]
 [1,] -0.21242248 -0.04338526 -0.17207928
 [2,]  0.28830514  0.45683335  0.45450365
 [3,] -0.09102308 -0.04666584  0.38953932
 [4,]  0.38301740  0.17757064  0.19280341
 [5,]  0.44046728  0.07263340  0.14050681
 [6,] -0.45444350 -0.39707532  0.49426978
 [7,]  0.02810549  0.39982497  0.15570580
 [8,]  0.39241904 -0.25391227  0.20853047
 [9,]  0.05143501 -0.45794047  0.04406602
[10,]  1.00000000  0.00000000  0.00000000
[11,]  0.00000000  1.00000000  0.00000000
[12,]  0.00000000  0.00000000  1.00000000
> Rglpk_solve_LP(obj = numeric(3), mat = A, dir = ge, rhs = rhs)
$optimum
[1] 0

$solution
[1] 0 0 0

$status
[1] 1

$solution_dual
[1] 0 0 0

$auxiliary
$auxiliary$primal
 [1] 0 0 0 0 0 0 0 0 0 0 0 0

$auxiliary$dual
 [1] 0 0 0 0 0 0 0 0 0 0 0 0


> A2 <- rbind(cbind(numeric(9), 1, -1), diag(3))
> Rglpk_solve_LP(obj = numeric(3), mat = A2, dir = ge, rhs = rhs)
$optimum
[1] 0

$solution
[1] 0.001 0.001 0.001

$status
[1] 0

$solution_dual
[1] 0 0 0

$auxiliary
$auxiliary$primal
 [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.001 0.001

$auxiliary$dual
 [1] 0 0 0 0 0 0 0 0 0 0 0 0
```

That code is from

`http://stackoverflow.com/questions/24432059/solution-of-system-of-inequalities-i`

In order to make that work, you have to install the package `Rglpk`. On a Mac, that also involves installing the package `slam`, which involves installing the GNU FORTRAN compiler *with* the quadmath library. I also had to upgrade XQuartz, the X Windows system for Mac. Yes, it's a hassle.

Notes on installing the compiler and library at

`http://thecoatlessprofessor.com/programming/rcpp-rcpparmadillo-and-os-x-maverick`

Once everything is more or less in place, try

```
> install.packages("Rglpk", dependencies = T)
also installing the dependency slam
```

and it will crank for a while and hopefully work.

2 The Simplex Algorithm

2.1 Anzen Dai-ichi

The idea of the simplex algorithm is pretty simple: repeat a simplex operation until no more simplex operations improve the result, and you're done. But there are a couple of things we need to worry about:

1. The second simplex operation might partly undo the work of the first!

2. At each vertex, we have several possible choices for which way to go next. If we choose poorly:
    (a) we might be inefficient in finishing; or
    (b) worse, we might wind up looping forever!
3. We need to be careful to recognize when we're on an edge that extends forever (that is, the solution is unbounded).
4. Finally, of course, we need some way to recognize when we are done, and have found the (or an) optimal solution.

2.2 The Basic Algorithm

See the separate file uploaded to SFS for pseudocode for the basic algorithm, slightly reformatted from p. 70 of textbook.

3 Interlude: Simplex in R

After all of this work to understand what's going on, we can now *use* packages that do the simplex algorithm for us!

This example is from `https://stat.ethz.ch/R-manual/R-devel/library/boot/html/simplex.html`:

```
> simplex()
Error: could not find function "simplex"
> library(boot)
> simplex()
Error in simplex() : argument "a" is missing, with no default
> enj <- c(200, 6000, 3000, -200)
> fat <- c(800, 6000, 1000, 400)
> vitx <- c(50, 3, 150, 100)
> vity <- c(10, 10, 75, 100)
> vitz <- c(150, 35, 75, 5)
> simplex(a = enj, A1 = fat, b1 = 13800, A2 = rbind(vitx, vity, vitz),
+         b2 = c(600, 300, 550), maxi = TRUE)

Linear Programming Results

Call : simplex(a = enj, A1 = fat, b1 = 13800, A2 = rbind(vitx, vity,
    vitz), b2 = c(600, 300, 550), maxi = TRUE)

Maximization Problem with Objective Function Coefficients
  x1   x2   x3   x4
 200 6000 3000 -200
```

```
Optimal solution has the following values
  x1   x2   x3   x4
 0.0  0.0 13.8  0.0
The optimal value of the objective  function is 41400.
```

It begins from the inequality form, with the $=$, $\geq$ and $\leq$ constraints in separate array arguments. There are various parameters you can use to control the behavior of the algorithm, as well. Note that it actually returns a value, which you can then use in further calculations, if you wish.

I'm not going discuss the use of this function in detail; part of your homework will be to explore its use.

4 Our favorite problem, step by step

In class, I went through the following problem, step by step:

$$\max z(\vec{x}) = (2,3,0,0,0)(x_1,x_2,x_3,x_4,x_5)^\intercal \tag{1}$$

subject to

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{pmatrix} \vec{x} = \begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix} \tag{2}$$

$$x_1,x_2,x_3,x_4,x_5 \geq 0. \tag{3}$$

1. First, put it in standard equality form (SEF), by (i) turning free variables x_i into two variables x_i^+ and x_i^- and augmenting the constraints; (ii) turning inequalities into equalities by adding slack variables and augmenting the constraints; and (iii) making sure we have a maximization problem, not a minimization problem. Fortunately, our problem was given to us in SEF, so no work was involved here.

2. Pick a basis, and put the problem in canonical form for that basis. We begin with $B = \{3,4,5\}$, which is an easy call because (i) the right hand columns are already an identity matrix (step one of canonical form) and (ii) the corresponding elements of the vector $\vec{c}$ in the objective function are already zero. (The $(0,0,0)$ in the objective function.) For references purposes, let's be explicit about this in the variable names we're going to need the next time we rewrite our problem, in step 10, below.

$$\vec{c}^\intercal = (2,3,0,0,0) \tag{4}$$

$$z_a = 0 \tag{5}$$

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$\vec{b} = (6,10,4)^\intercal \tag{7}$$

3. We're also going to need a basic solution for the problem, and it's pretty easy to see by inspection that

$$\vec{x_a} = \begin{pmatrix} 0 \\ 0 \\ 6 \\ 10 \\ 4 \end{pmatrix} \tag{8}$$

is a solution. (Textbook problems are easy! Doing this is real life isn't always so easy...) Check the value of our objective function, find that $z(\vec{x_a}) = 0$. Hopefully we can do better than \$0 in profit, in the end...

4. Now pick an element from the set $N = \{1, 2\}$ to maximize; let's choose 1. Set $x_1 = t$, keep the other $x_i = 0, i \in N$, and use the variables in the $\vec{x}$ for $x_i, i \in B$.

5. Next, algebra to find the best value for t. Plug $\vec{x}^{\mathsf{T}} = (t, 0, x_3, x_4, x_5)$ into $A\vec{x} = \vec{b}$ to get

$$t \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} + 0 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix}. \tag{9}$$

(If that's not obvious, try it.) Rewriting and swapping sides, we get

$$\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix} - t \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}. \tag{10}$$

Next, using the fact that $\vec{x} \geq \vec{0}$, we know that

$$t \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix}, \tag{11}$$

so how big can we make t? That calls for the ratio test! Divide element by element, and find the minimum (ignoring the negative constraint):

$$t = \min \left\{ \frac{6}{1}, \frac{10}{2}, - \right\} = 5. \tag{12}$$

6. Plug $t = 5$ into our equation for x_3, x_4, x_5,

$$\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix} - t \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 9 \end{pmatrix}. \tag{13}$$

Plug these values and $x_1 = t = 5$ back into $\vec{x_a}$, and get

$$\vec{x_a}' = \begin{pmatrix} 5 \\ 0 \\ 1 \\ 0 \\ 9 \end{pmatrix}. \tag{14}$$

Insert back into our objective function to get the new value

$$z_a' = z(\vec{x_a}') = (2, 3, 0, 0, 0) \begin{pmatrix} 5 \\ 0 \\ 1 \\ 0 \\ 9 \end{pmatrix} = 10, \tag{15}$$

so we have a new solution with an improved value for the objective function! This completes one simplex operation with the basis $B = \{3, 4, 5\}$, but before our loop is complete we need to test whether we are done or whether we have discovered that the problem is unbounded, then pick a new basis. (All of this was just the first couple of lines of the pseudocode.)

7. *Testing for optimality:* If all of the elements of $\vec{c}$ in N are non-positive, we're done! That is, $\vec{c_N} \leq \vec{0}$, then the $\vec{x_a}$ we are holding is optimal. In this case, $\vec{c_N} = (2, 3)$, so we're not there yet.

8. *Prepping for the next iteration, testing for unboundedness:* Pick a variable to add to the basis set B: $k \in N$ s.t. $c_k > 0$; in this case, let's pick 1. Look at the correspond column of A, $A_k = A_1 = (1, 2, -1)^\intercal$. If $A_k \leq \vec{0}$, we have a proof that our objective function is unbounded, and we're done. We fail this test, so as far as we know, it's bounded. Keep on trucking!

9. Now we need to pick the element we're going to take out of the basis B. This calls for a new application of the ratio test.

$$\begin{aligned} t &= \min \left\{ \frac{b_i}{A_{i,k}} : A_{i,k} > 0 \right\} \\ &= \min \left\{ \frac{6}{1}, \frac{10}{2}, - \right\} \end{aligned} \tag{16}$$

(Note that this was the same ratio test as above, but only because we picked the same variable to add as the one we optimized above; in theory, we could have picked a different one, but it seems reasonable to make this choice.) What we're looking for this time is not the value of t itself, but which one of the elements corresponds to the minimum value of t. If there are multiple ones with the same value, any of them will do. The second element is the one that minimizes t, so $r = 2$. Let iota, ι, be the rth element in the basis $B = \{3, 4, 5\}$. $B_2 = 4$, so we will remove $\iota = 4$ from the basis B.

Set $B \leftarrow B \cup \{k\} \backslash \{\iota\} = \{3,4,5\} \cup \{1\} \backslash \{4\} = \{1,3,5\}$.

In the pseudocode, the next step is "Go to step 1," so we'll go back.

10. We need to create a new canonical formulation of the problem,

$$z(\vec{x}) = \vec{y}^\intercal \vec{b} + z_a + (\vec{c}^\intercal - \vec{y}^\intercal A)\vec{x} \tag{17}$$

such that

$$A_B^{-1} A\vec{x} = A_B^{-1}\vec{b} \tag{18}$$

$$\vec{x} \geq \vec{0}. \tag{19}$$

where

$$\vec{y} = A_B^{-\intercal} \vec{c_B}. \tag{20}$$

(See Eq. 2.19, 2.20, 2.21 in support of Proposition 2.4 in the textbook.) The A, $\vec{b}$, $\vec{c}$ and z_a are from Eqs. 4–7. To get there, use A from Eq. 6 and pick the 1, 3, and 5 columns (our new basis),

$$A_B = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}. \tag{21}$$

Use R or Octave to take the inverse of that,

$$A_B^{-1} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 1 & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix}. \tag{22}$$

Multiply to get

$$A_B^{-1} A = \begin{pmatrix} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & \frac{3}{2} & 0 & \frac{1}{2} & 1 \end{pmatrix}. \tag{23}$$

Plugging in,

$$A_B^{-1}\vec{b} = A_B^{-1} \begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ 9 \end{pmatrix}. \tag{24}$$

Oh, and we need to calculate that $\vec{y}$, right? We've been postponing that. It involves solving a set of inequalities, but there is a shortcut:

$$\vec{y} = A_B^{-\intercal} \vec{c_B} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \tag{25}$$

(From our definition of the problem in Eq 4, $c^\intercal = (2,3,0,0,0)^\intercal$, so $c_B = (2,0,0)^\intercal$.) Plug into Eq. 17 restate our new form of the problem, and we get

$$\begin{aligned}
z(\vec{x}) &= \vec{y}^\intercal \vec{b} + z_a + (\vec{c}^\intercal - \vec{y}^\intercal A)\vec{x} \\
&= (0,1,0)\begin{pmatrix} 6 \\ 10 \\ 4 \end{pmatrix} + 10 + \\
&\left((2,3,0,0,0) - (0,1,0)\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{pmatrix}\right)\vec{x} \qquad (26) \\
&= 10 + 0 + (0,2,0,-1,0)\vec{x} \qquad (27)
\end{aligned}$$

such that

$$\begin{pmatrix} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & \frac{3}{2} & 0 & \frac{1}{2} & 1 \end{pmatrix}\vec{x} = \begin{pmatrix} 5 \\ 1 \\ 9 \end{pmatrix} \qquad (28)$$

$$\vec{x} \geq \vec{0}. \qquad (29)$$

Note that z_a should be 0, not 10. This is using the original z_a in Eq. 5, not the new z_a we just calculated. I also think it's just coincidence that $\vec{y}^\intercal \vec{b}$ equals our objective function value after the first complete iteration.

This brings us to the end of line 1 of the second iteration. Your homework will be to complete this iteration.

5 Homework

See the separate file uploaded to SFS.