Optimization I	**Summer 2011** **Lecture 6**

Simplex Method Implementation

Last time, we gave the following description of an iteration of the simplex method:

1. We start with a basis $A_{B(1)}, \ldots, A_{B(m)}$ and an associated bfs x.
2. We compute $\bar{c}_j = c_j - c_B^T \mathbf{B}^{-1} A_j$ for every non-basic variable. If $\bar{c}_j \geq 0$ for all j, we conclude x is optimal, and the algorithm terminates.
3. Otherwise, choose j such that $\bar{c}_j < 0$. Variable x_j enters the basis.
4. Let $u = \mathbf{B}^{-1} A_j$. If $u_i \leq 0$ for all i, then the optimal value is $-\infty$, and the algorithm terminates.
5. Otherwise, set $\theta^* = \min_{i:u_i>0} x_{B(i)}/u_i$, and let ℓ be such that $\theta^* = x_{B(\ell)}/u_\ell$. Variable $x_{B(\ell)}$ leaves the basis.
6. The new basis is obtained by replacing $A_{B(\ell)}$ by A_j, and the new bfs by setting $x_j^{new} = \theta^*$ and $x_{B(i)}^{new} = x_{B(i)} - \theta^* u_i$ for $i = 1, \ldots, m$.

We saw last time that, if we have an initial bfs, and there is no degeneracy, then the Simplex Method finds an optimal solution or correctly reports that the optimal value is $-\infty$ in finite time.

We will see how to prevent cycling in the case of degeneracy and how to find an initial basic feasible solution. Before we deal with these questions, we'll see a more convenient way of doing a simplex iteration.

1 Implementation of a Simplex Iteration

1.1 Revised Simplex Method

Note that in subsequent iterations, the matrix $\mathbf{B}$ only changes in one column. Hence, it seems that it should be possible to update $\mathbf{B}^{-1}$ without having to start from scratch. This is called the revised simplex method.

- Recall how to compute a matrix inverse of an $m \times m$ matrix $\mathbf{B}$ using Gauss-Jordan elimination: We append the columns of the identity matrix to obtain $[\mathbf{B}|I_m]$ and perform elementary row operations (which is the same as premultiplying by some $m \times m$ matrix Q), so we get $Q[\mathbf{B}|I_m] = [Q\mathbf{B}|Q]$. The row operations are chosen, so that the first m columns give the identity, i.e., $Q\mathbf{B} = I_m$. So $Q = \mathbf{B}^{-1}$ which we can read from the last m columns.

  Example: $\mathbf{B} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$.

$$\left[\begin{array}{ccc|ccc} 3 & 1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right] \xrightarrow{\text{subtract } 1/3R1 \text{ to } R2, \text{ divide } R1 \text{ by } 3} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & -\frac{4}{3} & 0 & -\frac{1}{3} & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{\text{add } 1/4R2 \text{ to } R1, \text{ add } 3/4R2 \text{ to } R3, \text{ divide } R2 \text{ by } -\frac{3}{4}} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 1 & 0 & \frac{1}{4} & -\frac{3}{4} & 0 \\ 0 & 0 & 1 & -\frac{1}{4} & \frac{3}{4} & 1 \end{array}\right].$$

  So $\mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & -\frac{3}{4} & 0 \\ -\frac{1}{4} & \frac{3}{4} & 1 \end{bmatrix}$.

- Now, if we update $\mathbf{B}$ in the Simplex Method, we just change one column. Suppose we replace column 2 with the column $\begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix}$, and suppose we do the exact same operations as we did previously. Then only the second column of the 3×6-matrix above changes. We also know what the second column looks like after the last step of the Gauss-Jordan elimination we did above: it will be equal to $\mathbf{B}^{-1} \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} =$ $\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & -\frac{3}{4} & 0 \\ -\frac{1}{4} & \frac{3}{4} & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ -\frac{1}{2} \\ \frac{3}{2} \end{bmatrix}$.

  So we can reuse our previous computations and find the new inverse as follows:

$$\left[\begin{array}{ccc|ccc} 1 & \frac{3}{2} & 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{4} & -\frac{3}{4} & 0 \\ 0 & \frac{3}{2} & 1 & -\frac{1}{4} & \frac{3}{4} & 1 \end{array}\right] \xrightarrow{\text{add } 3R2 \text{ to } R1, \text{ add } 3R2 \text{ to } R3, \text{ divide } R2 \text{ by } -\frac{1}{2}} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & -2 & 0 \\ 0 & 1 & 0 & -\frac{1}{2} & \frac{3}{2} & 0 \\ 0 & 0 & 1 & -\frac{1}{2} & -\frac{3}{2} & 1 \end{array}\right].$$

  So $\mathbf{B}_{new}^{-1} = \begin{bmatrix} 1 & -2 & 0 \\ -\frac{1}{2} & \frac{3}{2} & 0 \\ -\frac{1}{2} & -\frac{3}{2} & 1 \end{bmatrix}$.

- Note that for an $m \times m$ matrix, we thus only have to do $O(m^2)$ operations to update the inverse. Note that we can omit column 1 and 3 from the 3×6 matrix above (since they don't change anyway).

  In general, when we change to a new basis, we replace the ℓ-th column of $\mathbf{B}$, $A_{B(\ell)}$, by A_j. This changes the ℓ-th column of the outcome of the Gauss-Jordan elimination computation that computes $\mathbf{B}^{-1}$ (i.e., $\left[\begin{array}{c|c} I_m & \mathbf{B}^{-1} \end{array}\right]$) to $\mathbf{B}^{-1}A_j = u$, so we get

$$\left[\begin{array}{ccccccc|c} e_1 & \dots & e_{\ell_1} & u & e_{\ell+1} & \dots & e_m & \mathbf{B}^{-1} \end{array}\right].$$

  We need to do elementary row operations to change this to $\left[\begin{array}{c|c} I_m & \mathbf{B}_{new}^{-1} \end{array}\right]$ (where we note that columns $1, \dots, \ell-1, \ell+1, \dots, m$ will not change, so we could just delete them and only do the elementary row operations to change $\left[\begin{array}{c|c} u & \mathbf{B}^{-1} \end{array}\right]$ into $\left[\begin{array}{c|c} e_\ell & \mathbf{B}_{new}^{-1} \end{array}\right]$).

1.2 Tableau Implementation

A second way to implement the Simplex Method is the Tableau Implementation. Rather than just updating $\mathbf{B}^{-1}$ in a smart way, the tableau allows us to update all information we need in a handy way.

A tableau looks as follows:

$$\begin{array}{|c|c|} \hline -c_B^T\mathbf{B}^{-1}b & c^T - c_B^T\mathbf{B}^{-1}A \\ \hline \mathbf{B}^{-1}b & \mathbf{B}^{-1}A \\ \hline \end{array} \quad \overset{\text{associated solution } x_B = \mathbf{B}^{-1}b}{\Rightarrow} \quad \begin{array}{|c|c|} \hline -c_B^T x_B & \bar{c}_1 \dots \bar{c}_n \\ \hline x_{B(1)} & \\ \vdots & \mathbf{B}^{-1}A \\ x_{B(m)} & \\ \hline \end{array}$$

The tableau has rows labeled 0 to m, and columns labeled 0 to n.

The j-th column of the tableau looks like this: $\begin{array}{|c|} \hline \bar{c}_j \\ \hline \mathbf{B}^{-1}A_j \\ \hline \end{array}$. If $j = B(i)$, then this is equal to $\begin{array}{|c|} \hline 0 \\ \hline e_i \\ \hline \end{array}$.

Given the tableau, we can very conveniently do a simplex iteration:

- $\bar{c}_j$ for all j is given in row 0, so we can just pick a column j with $\bar{c}_j < 0$. We call this the *pivot column.*
- Column j in the tableau also contains $\mathbf{B}^{-1}A_j$ which is the vector u we need.
- Let ℓ be the index of the row that minimizes $x_{B(i)}/u_i$ over all rows with $u_i > 0$ (this is called the *minimum ratio test*). We call row ℓ the *pivot row*, and (ℓ, j) the *pivot element.*
- Update the tableau: add/subtract multiples of the ℓ-th row to the other rows, so that the j-th column has a 1 in position (ℓ, j) and 0 everywhere else.

  To see this is correct, recall that we already argued that doing elementary row operations is the same as premultiplying by some matrix Q. First consider the rows $1, \dots, n$ of the tableau. Note that the matrix Q satisfies $Q\mathbf{B}^{-1}A_{B(i)} = e_i$ for all $i \neq \ell$ (since the columns for the variables that remain basic stay the same), and $Q\mathbf{B}^{-1}A_j = e_\ell$. So $Q\mathbf{B}^{-1}\mathbf{B}_{new} = I_m$, or $Q\mathbf{B}^{-1} = \mathbf{B}_{new}^{-1}$. Then $Q \left[\begin{array}{c|c} \mathbf{B}^{-1}b & \mathbf{B}^{-1}A \end{array} \right] = \left[\begin{array}{c|c} \mathbf{B}_{new}^{-1}b & \mathbf{B}_{new}^{-1}A \end{array} \right]$ which is exactly what we need.

  For row 0, note that it is originally equal to

$$\left[\begin{array}{c|c} -c_B^T\mathbf{B}^{-1}b & c^T - c_B^T\mathbf{B}^{-1}A \end{array} \right] = \left[\begin{array}{c|c} 0 & c^T \end{array} \right] - c_B^T\mathbf{B}^{-1} \left[\begin{array}{c|c} b & A \end{array} \right].$$

  Then we add some multiple γ of row ℓ, which is itself equal to $\mathbf{b}_i^{-1,T} \left[\begin{array}{c|c} b & A \end{array} \right]$, where $\mathbf{b}_i^{-1,T}$ is the i-th row of $\mathbf{B}^{-1}$. So, the new row 0 is

$$\left[\begin{array}{c|c} 0 & c^T \end{array} \right] - (c_B^T\mathbf{B}^{-1} - \gamma\mathbf{b}_i^{-1,T}) \left[\begin{array}{c|c} b & A \end{array} \right]. \tag{1}$$

  Let $p^T = c_B^T\mathbf{B}^{-1} - \gamma\mathbf{b}_i^{-1,T}$. We want to show that $p^T = c_{B_{new}}^T\mathbf{B}_{new}^{-1}$:

  We chose γ so that the reduced cost of j becomes 0, and note that we also maintain that the reduced cost of the other variables that remain basic in the new basis is 0. Since the reduced cost of these variables in the new tableau is $c_{B_{new}}^T - p^T \left[A_{B_{new}} \right]$ (we get this from (1) by removing the non-basic variables and the zero-th entry), we thus get that $p^T = c_{B_{new}}^T A_{B_{new}}^{-1} = c_{B_{new}}^T \mathbf{B}_{new}^{-1}$.

1.3 An Example of the Tableau Implementation (with Degeneracy)

Example:

$$\begin{array}{llllllll}
\min & -2x_1 & -x_2 & & & & \\
\text{s.t.} & 3x_1 & +x_2 & +x_3 & & & = 6 \\
 & x_1 & -x_2 & & +x_4 & & = 2 \\
 & & x_2 & & & +x_5 & = 3 \\
 & x_1, & x_2, & x_3, & x_4, & x_5 \geq 0 &
\end{array}$$

We first choose x_3, x_4, x_5 as the basic variables. Then $\mathbf{B}$ is the identity, so $\mathbf{B}^{-1}A = A$, $\mathbf{B}^{-1}b = b$. Also, $c_B^T = [0, 0, 0]$, so $c_B^T x_B = 0$ and $\bar{c}_j = c_j - c_B^T\mathbf{B}^{-1}A_j = c_j$.

Our first tableau is therefore:

0	-2	-1	0	0	0
6	3	1	1	0	0
2	1	-1	0	1	0
3	0	1	0	0	1

We choose x_1 as the entering variable. We have a choice for the pivot row (or leaving variable), since $\frac{6}{3} = \frac{2}{1}$. We choose the second row, which means x_4 leaves the basis. We add/subtract copies of the second row to each other row, so that the first column has 0's everywhere except in the second row.

4	0	-3	0	2	0
0	0	4	1	-3	0
2	1	-1	0	1	0
3	0	1	0	0	1

Note that this solution is degenerate.

In the next iteration, x_2 is our only choice for the entering variable, and the first row is the pivot row (so x_3 is the leaving variable).

4	0	0	3/4	-1/4	0
0	0	1	1/4	-3/4	0
2	1	0	1/4	1/4	0
3	0	0	-1/4	3/4	1

Note that this is the same solution $(x_1, x_2, x_3, x_4, x_5) = (2, 0, 0, 0, 3)$. Now, we let x_4 enter the basis. The pivot row is the last row.

5	0	0	2/3	0	1/3
3	0	1	0	0	1
1	1	0	1/3	0	-1/3
4	0	0	-1/3	1	4/3

We have found an optimal solution, $x(_1, x_2, x_3, x_4, x_5) = (1, 3, 0, 4, 0)$ with objective value -5.

Remarks:

- Time per operation is $O(mn)$.
- Note that we had no choice but to do a degenerate pivot in this example, but we did not encounter cycling – after one degenerate pivot, the next pivot (which was our only choice) led to a different solution. See http://glossary.computing.society.informs.org/notes/cycling.pdf for an example where cycling does occur (note: the representation in this paper has our row 0 as its bottom row, and our column 0 as its rightmost column).