Mathematical Programming	**Lecture 13**
OR 630 Fall 2005	**October 6, 2005**
Notes by Saifon Chaturantabut	

1 Implementation (continued)

We noted last time that

$$\begin{aligned} B_+ &= B + (a_q - Be_p)e_p^T \\ &= B(I + (\bar{a}_q - e_p)e_p^T). \end{aligned}$$

Now, we want to find the inverse of B_+. Notice that $(a_q - Be_p)e_p^T$ has rank one and therefore B_+ is a rank-one modification of B. We can use the following proposition to find B_+^{-1}.

Proposition 1 *(Rank-One Modification) Let $M \in \mathbf{R}^{m\times m}$ be nonsingular and $u, v \in \mathbf{R}^m$. Then $M + uv^T$ is nonsingular if and only if $1 + v^T M^{-1}u$ is nonzero, in which case*

$$(M + uv^T)^{-1} = M^{-1} - \frac{M^{-1}uv^T M^{-1}}{1 + v^T M^{-1}u}. \tag{1}$$

Proof We have

$$\begin{pmatrix} I & -u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} M + uv^T & 0 \\ v^T & 1 \end{pmatrix} = \begin{pmatrix} M & -u \\ v^T & 1 \end{pmatrix} = \begin{pmatrix} M & 0 \\ v^T & 1 + v^T M^{-1}u \end{pmatrix}\begin{pmatrix} I & -M^{-1}u \\ 0 & 1 \end{pmatrix}, \text{ so}$$

$$\begin{pmatrix} M + uv^T & 0 \\ v^T & 1 \end{pmatrix} = \begin{pmatrix} I & u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} M & 0 \\ v^T & 1 + v^T M^{-1}u \end{pmatrix}\begin{pmatrix} I & -M^{-1}u \\ 0 & 1 \end{pmatrix}.$$

$M + uv^T$ is nonsingular exactly when the LHS above is, which holds if and only if the RHS is, which holds if and only if $1 + v^T M^{-1}u \neq 0$. Assume this holds.

Taking the inverse of both sides, we get

$$\begin{pmatrix} (M + uv^T)^{-1} & 0 \\ ?? & 1 \end{pmatrix} = \begin{pmatrix} I & M^{-1}u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} M^{-1} & 0 \\ -(1 + v^T M^{-1}u)^{-1} & (1 + v^T M^{-1}u)^{-1} \end{pmatrix}\begin{pmatrix} I & -u \\ 0 & 1 \end{pmatrix}.$$

Note that we don't care about the lower-left block matrix of the LHS, since we are only interested in finding the inverse of $M + uv^T$.
So,

$$(M + uv^T)^{-1} = \begin{pmatrix} I & M^{-1}u \end{pmatrix}\begin{pmatrix} M^{-1} \\ -(1 + v^T M^{-1}u)^{-1}v^T M^{-1} \end{pmatrix},$$

and we get what we want. $\square$

The equation (1) is often called the *"Sherman-Morrison formula,"* but there are earlier papers deriving it. There is a nice generalization of this formula, sometimes called the *"Sherman-Morrison-Woodbury formula,"* which goes as follows:

Remark 1 *(Sherman-Morrison-Woodbury formula) If $M \in \mathbf{R}^{m\times m}$ and $U, V \in \mathbf{R}^{k\times m}(k < m)$, then*

$M + UV^T = M + \sum_{j=1}^{k} u_j v_j^T$ *is nonsingular if and only if $I + V^T M^{-1} U$ is, in which case*

$$(M + UV^T)^{-1} = M^{-1} - M^{-1}U(I + V^T M^{-1} U)^{-1} V^T M^{-1}.$$

Above is the formula for the inverse of a rank-k modification of M. The proof of this is similar to the one for the rank-one modification.

Now, back to our case. By Proposition 1, the rank-one modification of B: $B_+ = B + (a_q - Be_p)e_p^T$ is nonsingular if and only if $1 + e_p^T B^{-1}(a_q - Be_p) = 1 + e_p^T(\bar{a}_q - e_p) = \bar{a}_{pq}$ is nonzero, and then

$$B_+^{-1} = B^{-1} - \frac{(\bar{a}_q - e_p)e_p^T B^{-1}}{\bar{a}_{pq}} = \left(I - \frac{(\bar{a}_q - e_p)e_p^T}{\bar{a}_{pq}}\right) B^{-1}.$$

Recall the 5-step (primal) simplex method.

Ways To Update in Step 5

1. Keep and update $\bar{A}$, $\bar{b}$, $\bar{c}$, and $\bar{\zeta}$:
   - $\bar{A} = B^{-1}A$,
   - $\bar{b} = B^{-1}b$,
   - $\bar{c} = c - A^T B^{-1} c_B$,
   - $\bar{\zeta} = c_B^T B^{-1} b$.

   This is all the information in the equations:

$$\begin{array}{rcrcl} \zeta & - & \bar{c}^T x & = & \bar{\zeta} \\ & & \bar{A}x & = & \bar{b}, \end{array} \tag{2}$$

   or all the information in the tableau:

Basic Variable	ζ	x	RHS
-	1	$-\bar{c}^T$	$\bar{\zeta}$
x_B	0	$\bar{A}$	$\bar{b}$

   To update this tableau,

(a) Update $\bar{A}$ and $\bar{b}$: Pre-multiply $\bar{A}$ and $\bar{b}$ by the elementary matrix inverse, E_p^{-1}:
   - Divide the p^{th} row by $\bar{a}_{pq}$;
   - Multiply the new p^{th} row by $-\bar{a}_{iq}$ and add to the i^{th} row, for $i = 1, \ldots, m$.

(b) Update $\bar{c}$ and $\bar{\zeta}$:
   - Add $\bar{c}_q$ times the new p^{th} row of $\bar{A}, \bar{b}$ to the old $-\bar{c}$, $\bar{\zeta}$ to get the new values.

Now, let's talk about the number of arithmetic operations.

Assumption 1

- *A is an $m \times n$ matrix and each of its columns has a fraction α of nonzeroes. (Think of $m = 10,000, n = 100,000, \alpha = 10^{-3}$.)*
- *B^{-1} and $\bar{A}$ are assumed to have density (fraction of nonzeroes) β and $\beta > \alpha$. (Think of $\beta = 10^{-2}$.)*

Then, work involved:

Step 1: *None.*
Step 2: *None.*
Step 3: *None.* (No work for comparisons.)
Step 4: βm (from computing $\frac{\bar{b}_i}{\bar{a}_{iq}}$ for $\bar{a}_{iq} > 0$).
Step 5: $2\beta m(n - m) + O(n)$ ($2\beta m(n - m)$ from (a) and $O(n)$ from (b): the factor 2 comes from multiplications and additions).

Using the above example ($m = 10,000$, $n = 100,000$, $\alpha = 10^{-3}$, $\beta = 10^{-2}$), we have about 2×10^7 operations needing to be done in each iteration.

Above is the original simplex method.

2. Keep and update B^{-1}, $\bar{y}$, $\bar{b}$, and $\bar{\zeta}$.
   Work involved:

   Step 1: Compute $\bar{c}$ to check optimality: $\bar{c}_j = c_j - a_j^T \bar{y}$ for $j \in \nu$.
   So, we use $2\alpha m(n - m)$ operations in this step.

   Step 2: *None.*

   Step 3: Compute $\bar{a}_q$ to check unboundedness: $\bar{a}_q = B^{-1} a_q$.
   So, we use $2\alpha m^2$ operations (note that a_q has only αm nonzeroes, so this is the number of columns of B^{-1} we need to combine).

   Step 4: βm (from computing $\frac{\bar{b}_i}{\bar{a}_{iq}}$ for $\bar{a}_{iq} > 0$).

Step 5: Update

$$\begin{array}{lll} B^{-1} & : & 2\beta m^2 \text{ (like updating } \bar{A}\text{, but only } m \text{ columns).} \\ \bar{b} & : & 2\beta m. \\ \bar{\zeta} & : & O(1). \\ \bar{y}_+ & : & 2m. \end{array}$$

Below shows how we obtain $2m$ operations used in updating $\bar{y}_+$.

$$\begin{aligned} \bar{y}_+ &= B_+^{-T} c_B^+ \\ &= \left(B^{-T} - \frac{B^{-T} e_p (\bar{a}_q - e_p)^T}{\bar{a}_{pq}} \right) \left(c_B + (c_q - c_B^T e_p) e_p \right) \\ &= B^{-T} c_B - \frac{(\bar{a}_q - e_p)^T c_B}{\bar{a}_{pq}} B^{-T} e_p + (c_q - c_B^T e_p) \underbrace{\left(B^{-T} e_p - \frac{(\bar{a}_q - e_p)^T e_p}{\bar{a}_{pq}} B^{-T} e_p \right)}_{\frac{B^{-T} e_p}{\bar{a}_{pq}}} \\ &= B^{-T} c_B + \left(\frac{c_q - c_B^T \bar{a}_q}{\bar{a}_{pq}} \right) B^{-T} e_p \\ &= B^{-T} c_B + \frac{\bar{c}_q}{\bar{a}_{pq}} B^{-T} e_p \\ &= \bar{y} + \frac{\bar{c}_q}{\bar{a}_{pq}} B^{-T} e_p. \end{aligned}$$

The work involved here is $2m$ (multiplications/additions).

Note: This is almost the same as updating $\bar{c}$. Indeed, if the original problem had slacks with 0 cost, their corresponding reduced costs would be : $0 - I^T \bar{y} = -\bar{y}$.

The total work is:

$$\underbrace{2\alpha mn}_{\text{Computing } \bar{c} \text{ and } \bar{a}_q} + \underbrace{2\beta m^2}_{\text{Updating } B^{-1}} + O(m).$$

From our earlier example ($m = 10,000$, $n = 100,000$, $\alpha = 10^{-3}$, $\beta = 10^{-2}$), we have approximately $2 \times 10^6 + 2 \times 10^6 = 4 \times 10^6$ operations needing to be done in each iteration.

Above is called the *"Revised Simplex Method."* This method requires the minimum amount of updates in order to perform each simplex iteration.

3. The third possibility is similar to the second one. The only difference is that it also updates $\bar{c}$. That is, this method has no work involved in **Step 1**, but it has additional

work in updating $\bar{c}$ in **Step 5**.

$$\bar{c}_+ = \bar{c} - \frac{\bar{c}_q}{\bar{a}_{pq}} A^T B^{-T} e_p.$$

So, for each j,

$$\bar{c}_{j+} = \bar{c}_j - \left(\frac{\bar{c}_q}{\bar{a}_{pq}}\right) \underbrace{\left(a_j^T (B^{-T} e_p)\right)}_{2\alpha m\text{-work}} \text{ for } j \in \nu.$$

Computing $a_j^T(B^{-T}e_p)$ involves $2\alpha m$ operations (multiplications/additions). Hence, the work involved in updating $\bar{c}$ is $2\alpha m(n-m)$ again as in the previous method.

Note that, in order to do the simplex algorithm, we do not need to update $\bar{c}_j$ for all $j \in \nu$, since we just want to choose only one negative $\bar{c}_j$ to determine the entering-basis variable. This can reduce the number of arithmetic operations.

Factorization

An efficient way to find B^{-1} is to compute it from a factorization of B. Here are two possible factorizations we can use.

1. LU Factorization: $B = LU$ where L is a lower triangular matrix and U is an upper triangular matrix.
2. QR Factorization: $B = QR$ where Q is an orthogonal matrix and R is an upper triangular matrix.

The LU factorization is better to use here, since it preserves the sparsity of B whereas the QR factorization can destroy the sparsity of B. We can update L and U in each iteration by changing only some entries of them.

2 Finite Termination in the Presence of Degeneracy

In order to have finite termination in the case of degeneracy, we need specialized pivot rules. On the course homepage, there is an example exhibiting cycling using a very natural pivot rule.

There are rules just restricting the choice of p (the index of the variable leaving the basis), but we will discuss one that restricts both p and q (the index of the variable entering the basis). (So we have to "mind our ps and qs.") This is the so-called *Least-Index Rule* (Bland, '77).

Least-index rule

At each iteration, call a nonbasic variable *eligible to enter the basis* if its reduced cost is negative. Choose the eligible variable x_q with *least* index q. Call a basis variable x_j *eligible to leave the basis* if it is the i^{th} basic variable, $\bar{a}_{iq} > 0$, and $\frac{\bar{b}_i}{\bar{a}_{iq}} = \min_h\{\frac{\bar{b}_h}{\bar{a}_{hq}} : h = 1, \ldots, m\ , \bar{a}_{hq} > 0\}$. Choose the eligible variable with *least* index j to leave the basis.

Next time we will see the proof that this rule leads to finite termination.