Initialization, Anti-Cycling, Number of Iterations

Today, we are going to try to wrap up all the remaining loose ends of the Simplex Method. We will see how to find an initial bfs (but some details are left for the homework) and we will see how to prevent cycling (and thus how we can ensure that the Simplex Method terminates in finite time). Finally, we will summarize previous discussions of some other pivot rules and we will discuss the number of iterations and complexity of the simplex method.

1 Initialization

Suppose we want to solve

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

We know how to find the optimal solution once we have a basic feasible solution to start with. But how to find this initial bfs?

First, we can make sure that the right hand side vector b has only non-negative entries (otherwise, if $b_i < 0$, we can just replace $a_i^T x = b_i$ by $-a_i^T x = -b_i$). We then first solve the following problem:

$$\begin{aligned} \min \quad & [1, \ldots, 1]y \\ \text{s.t.} \quad & Ax + I_m y = b \\ & x, y \geq 0 \end{aligned}$$

The variables $y_1, \ldots, y_m$ are called the *artificial variables*. For this problem, the initial bfs is easily found – just take the solution $x = \vec{0}, y = b$. This is feasible and the corresponding basis is I_m.

Now, we can use the simplex method to find an optimal solution for the problem of minimizing $\sum_i y_i$. This is called Phase I. We either find a solution such that $\sum_i y_i = 0$, or the optimal solution has $\sum_i y_i > 0$. In the latter case, there exists no x such that $x \geq 0, Ax = b$, so the problem we really want to solve is infeasible. In the first case, we found a solution x to $Ax = b, x \geq 0$. But is it a *basic* feasible solution and what is the associated basis?

In the homework, you will see that this is not necessarily the case, but you will see how to turn x into a bfs.

Once we have a bfs for $Ax = b, x \geq 0$, we can continue with the Simplex Method to minimize the real objective $c^T x$. This is referred to as Phase II of the Simplex method.

2 Anti-Cycling Rules

We say that the Simplex Method *cycles* if we encounter a sequence of bases $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(r)}$ where $\mathbf{B}^{(r)} = \mathbf{B}^{(1)}$; in other words, the r-th basis is exactly the same as the first one (and, therefore, we keep cycling forever). It must be the case that each of the pivots in this sequence is degenerate, i.e., $\theta^* = 0$ in each of the Simplex iterations in this sequence – otherwise, we know the objective value decreases by $\bar{c}_j \theta^* < 0$, and hence it is not possible to come back to the same basis.

We will now see a simple rule that can make sure we do not cycle.

Definition 1 (Bland's least index rule).
- *If there is a choice of entering variables (i.e., there are several j such that $\bar{c}_j < 0$), choose the variable with the smallest index, (i.e., we choose index $j^* = \min\{j : \bar{c}_j < 0\}$).*
- *If there is a choice of leaving variables (i.e., there are several ℓ such that $x_{B(\ell)}/u_\ell = \min_{i:u_i>0} x_{B(i)}/u_i$), choose the variable with the smallest index (i.e., choose index $j^* = \min\{j : j = B(\ell)$ for some $\ell = 1, \ldots, m, u_\ell > 0$, and $x_{B(\ell)}/u_\ell = \min_{i:u_i>0} x_{B(i)}/u_i\})$.*

Theorem 1. *If the entering and leaving variables are chosen according to Bland's rule, then the Simplex Method does not cycle.*

Proof. Suppose the Simplex Method does cycle. Then there is a sequence of bases $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(r)}$ where $\mathbf{B}^{(r)} = \mathbf{B}^{(1)}$. Note that the corresponding basic feasible solution is the same for all these bases.

We remove all columns and rows from the tableau that never contain the pivot element in any of the iterations of the cycle. Note that this corresponds to looking at an instance with fewer variables and fewer constraints, but that - if the Simplex Method cycles on the original instance - the Simplex Method will also cycle on this instance. Also, note that all variables in the remaining instance have $x_j = 0$ in the current basic feasible solution.

Let x_t be the variable with the highest index. There must be some iteration in the cycle when x_t enters the basis and when x_t leaves the basis.

- Consider the iteration when x_t enters the basis. Let $B(1), \ldots, B(m)$ be the indices of the basic variables in this iteration, and let A_B be the basis matrix. Let $\bar{c}$ be the reduced cost in this iteration, i.e., $\bar{c}^T = c^T - c_B^T A_B^{-1} A$. Note that $\bar{c}_j \geq 0$ for all j except for $j = t$ (since for basic variables, $\bar{c}_j = 0$ and for all non-basic variables, we must have $\bar{c}_j \geq 0$ because otherwise we would have chosen a different entering variable.)
- Consider the iteration when x_t leaves the basis. Let $D(1), \ldots, D(m)$ be the indices of the basic variables in this iteration, and let A_D be the basis matrix. Let x_s be the variable that enters the basis in this iteration, and let x_t be the ℓ-th basic variable, i.e., $D(\ell) = t$.

  The pivot column of the tableau is $\begin{bmatrix} c_s - c_D^T A_D^{-1} A_s \\ A_D^{-1} A_s \end{bmatrix}$. Let $u = A_D^{-1} A_s$.

  Now, since $x_{D(i)} = 0$ for all i, any basic variable $x_{D(i)}$ with $u_i > 0$ can be chosen to leave the basis. Since we choose $x_t = x_{B(\ell)}$ (the variable with the highest index), it must be the case that $u_i \leq 0$ for all $i \neq \ell$.

  Define
  $$d_j = \begin{cases} -u_i & \text{if } j = D(i) \\ 1 & \text{if } j = s \\ 0 & \text{otherwise} \end{cases}$$
  Then $d_j \geq 0$ for all j except $j = t$.

  Finally, we note that $c_s - c_D^T A_D^{-1} A_s < 0$, since the Simplex Method chose x_s to enter the basis.

We consider the product $\bar{c}^T d$. Since $\bar{c}_j \geq 0$ for all j except $j = t$ and $d_j \geq 0$ for all j except t, and $\bar{c}_t < 0, d_t < 0$, we have $\bar{c}^T d > 0$.

On the other hand, note that $Ad = A_D d_D + A_s = -A_D u + A_s = -A_D A_D^{-1} A_s + A_s = 0$. Therefore,
$$\begin{aligned} \bar{c}^T d &= (c^T - c_B^T A_B^{-1} A) d \\ &= c^T d \qquad\qquad \text{since } Ad = 0 \\ &= c_D^T d_D + c_s \\ &= -c_D^T u + c_s \\ &= -c_D^T A_D^{-1} A_s + c_s. \end{aligned}$$
But we know this last quantity, $c_s - c_D^T A_D^{-1} A_s$ must be less than zero, since it is the reduced cost of x_s when x_s entered the basis. So, we have a contradiction. □

3 Some Other Pivot Rules

Bland's rule is a pivot rule which guarantees that the number of iterations is finite, but in practice, other pivot rules may lead to a faster algorithm. Here are a few common rules, and their advantages and disadvantages.

- Choose the column for which $\bar{c}_j < 0$ is most negative.

  Disadvantage: it may be the case that θ^* - the amount by which we can move in the direction d which has $d_j = 1, d_{j'} = 0$ for $j' \neq j, B(1), \ldots, B(m)$ and $d_B = -u = -\mathbf{B}^{-1}A_j$ - is tiny so we do not actually make much progress.

- Choose the column for which $\theta^* \bar{c}_j$ is most negative.

  Disadvantage: we need to compute d and then θ^* for each column, which is computationally expensive.

  Advantage: the number of iterations tends to be smaller in practice.

- Choose an arbitrary column for which $\bar{c}_j < 0$ (say, the first one we encounter).

  Advantage: we don't have to compute all $\bar{c}_j$'s. Note also that this is the same as Bland's rule, so we know cycling does not occur.

  Disadvantage: we may make only a small amount of progress in an iteration.

- Choose the column for which $\bar{c}_j/\|d\|$ is most negative (steepest edge).

  Disadvantage: we need to compute d for each column, which is computationally expensive (but we do not need to know θ^*, which we needed for the second pivot rule).

  Advantage: in practice, the number of iterations tends to be smaller.

4 Number of Iterations

We have seen that an iteration of the Simplex method takes polynomial time (for example, the full tableau implementation takes $O(mn)$ time). To determine the complexity of the Simplex method, we need to bound the number of iterations.

We know that if we use Bland's rule, the Simplex method considers each basis at most once. Hence, the number of iterations is at most the number of bases, $\binom{n}{m}$. In practice, $O(m)$ iterations seem to suffice for most problems.

For most known pivot rules, examples are known where the number of iterations is $2^n - 1$. The idea behind these examples is due to Klee and Minty (1970): start with the n-dimensional unit hypercube (so the constraints are $0 \leq x_j \leq 1$ for $j = 1, \ldots, n$). You can show that there is a Hamiltonian path, i.e. a path that visits all vertices exactly once. The idea is now to perturb the constraints and choose the costs so that the simplex algorithm visits all vertices according to the Hamiltonian path, and thus makes $2^n - 1$ pivots.

No pivot rule is known for which the number of iterations is polynomial. Hence, although algorithms based on the Simplex method (with a "good" pivot rule and other tricks to speed up computation) work well in practice, they are not polynomial time algorithms.

In fact, the existence of a pivot rule which would guarantee that the number of iterations is related to the *diameter* of a polyhedron: Let's define the distance $d_P(x, y)$ between two bfs x and y of a polyhdron P as the minimum number of non-degenerate pivots needed to go from x to y. Then, the diameter $D(P)$ of a polyhedron P is defined as $\max_{x,y:\text{ bfs of } P} d_P(x, y)$.

Let $\mathcal{P}_{n,m}$ be the set of all **bounded**[1] polyhedra $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ for which $A \in \mathbb{R}^{m \times n}$, and define

$$\Delta(n, m) = \max_{P \in \mathcal{P}_{n,m}} D(P).$$

[1] In lecture, I forgot to say that P should be bounded

If $\Delta(n, m)$ is not bounded by a polynomial in n, m, then clearly, no pivot rule can exist that is guaranteed to give at most a polynomial number of iterations[2] If there exists a pivot rule which leads to a polynomial number of iterations, then this implies that $\Delta(n, m)$ is bounded by some polynomial in n, m. A famous conjecture that was open for a long time (since 1957) is the Hirsch conjecture:

Conjecture 1 (Hirsch conjecture)**.**

$$\Delta(n, m) \leq m - n.$$

Last year, Francisco Santos gave an example that shows that the **Hirsch conjecture is not true**.

Note however, that it may still be true that $\Delta(m, n)$ is bounded by a polynomial in n, m. The best upperbound currently known on $\Delta(m, n)$ is $m^{(1+\log n)}$.

[2]In lecture, I said something which is not right: instead of "if not A then not B" (as is written here), I said "B then A". Sorry for my mistake!