1 Overview

In our previous lecture we saw the application of the strong duality theorem to game theory, and then saw how that is applied to learning theory where we showed that weak learning implies strong learning. Today we'll present the simplex method for solving linear programs. We will start with discussing basic solutions and then show how this applies to the simplex algorithm.

2 Basic Feasible Solutions

Definition 1. *We say that a constraint* $\boldsymbol{ax} \leq \boldsymbol{b}$ *is* ***active*** *(or* ***binding****) at point* $\bar{\boldsymbol{x}}$ *if* $\boldsymbol{a\bar{x}} = \boldsymbol{b}$*.*

Definition 2. *A solution in* $P = \{\boldsymbol{x} : A\boldsymbol{x} \leq b\}$ *is called* ***basic feasible*** *if it has* n *linearly independent active constraints.*

Definition 3. *A solution in* $P = \{\boldsymbol{x} : A\boldsymbol{x} \leq b\}$ *is called* ***degenerate*** *if it has more than* n *linearly independent active constraints.*

Example: Degeneracy does not imply redundancy. Consider the pyramid in $\mathbb{R}^3$. Any feasible solution in the pyramid only has 3 linearly independent active constraints, but we need at least 4 constraints to represent the pyramid.

2.1 Basic solutions in standard form

We say that an LP is in standard form if we express it as:

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

Let us assume that A is a $m \times n$ matrix. Any linear program can be written in the standard form with $m \leq n$. Without loss of generality we can assume that $rank(A) = m$ (if $rank(A) < m$, then the system has redundant constraints that can be identified and removed). Pick a set of indices $B \subseteq [n]$ that correspond to m linearly independent columns of the matrix A. Now, we can think of the matrix A as the concatenation of two matrices A_B and A_N where A_B is the $m \times m$ matrix whose columns are indexed by the indices in B, and A_N is the $m \times (n-m)$ matrix whose columns are indexed by the indices in $[n] \setminus B$. Similarly we can think of $\mathbf{x}$ as $[\mathbf{x}_B, \mathbf{x}_N]$ in a natural manner.

$$
\begin{aligned}
A &= [A_B \mid A_N], \\
\mathbf{x} &= [\mathbf{x}_B \mid \mathbf{x}_N].
\end{aligned}
$$

Remark 4. *For any basic feasible solution $\boldsymbol{x}$, we have a set $B \subseteq [n]$ of m indices that correspond to a linearly independent set of columns of A such that:*

1. $\boldsymbol{x}_N = 0$
2. $\boldsymbol{x}_B = A_B^{-1}\boldsymbol{b}$.

In addition, for any set $B \subseteq [n]$ of m indices that correspond to a linearly independent set of columns, if $\boldsymbol{x}_B = A_B^{-1}\boldsymbol{b} \geq 0$ then $(\boldsymbol{x}_B, \boldsymbol{x}_N)$ is basic feasible.

We'll conclude this discussion with an important theorem which will be used in the simplex method.

Theorem 5. *Let $P = \{\boldsymbol{x} \in \mathbb{R}^n \ : A\boldsymbol{x} \leq b \}$ then $\boldsymbol{x}$ is an extreme point of P if and only if $\boldsymbol{x}$ is a basic feasible solution of P.*

The proof follows the same principles as the proofs for extreme points and is left as an exercise in your next problem set.

3 The Simplex Algorithm

From the above discussion, it is clear that in order to find an optimal solution, it is sufficient to search over the basic feasible solutions to find the optimal one. The Simplex Algorithm, given by Dantzig, does this search in an organized fashion.

Algorithm 1 Simplex

1: Let $(\mathbf{x}_B, \mathbf{x}_N)$ be a basic feasible solution.
2: $\bar{\mathbf{c}}^T \leftarrow \mathbf{c}^T - \mathbf{c}_B^T A_B^{-1} A$
3: $\mathbf{x}_B \leftarrow \bar{\mathbf{b}} := A_B^{-1}\mathbf{b}$
4: **if** $\bar{\mathbf{c}} \geq 0$ **then**
5:     STOP and return $(\mathbf{x}_B, \mathbf{x}_N)$ as optimal solution.
6: **end if**
7: Select $j \in N$ such that $\bar{\mathbf{c}}_j < 0$
8: $d_j \leftarrow A_B^{-1} A_j$
9: **if** $d_j \leq 0$ **then**
10:     STOP; return "LP is unbounded!! YOLO".
11: **end if**
12: $k \leftarrow \text{argmin}_{\{i \in B : d_{ji} > 0\}} (\bar{\mathbf{b}}_i / d_{ji})$
13: $B \leftarrow (B \setminus \{k\}) \cup \{j\}, N \leftarrow (N \setminus \{j\}) \cup \{k\}$
14: go to step 1.

Example. Consider the following linear program.

$$\begin{aligned} \min\ & -x_1 - 3x_2 \\ \text{s.t. } & 2x_1 + 3x_2 \leq 6 \\ & -x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Let us first write this in standard form:

$$\begin{aligned} \min\ & -x_1 - 3x_2 \\ \text{s.t. } & 2x_1 + 3x_2 - x_3 = 6 \\ & -x_1 + x_2 - x_4 = 1 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Say we start with $B = \{3, 4\}$ and $N = \{1, 2\}$. It should be clear that the resulting solution $(x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1)$ is a basic feasible solution (Verify that the corresponding columns of A are linearly independent). Then,

$$A_B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 2 & 3 \\ -1 & 1 \end{bmatrix}.$$

Iteration 1.

$$\begin{aligned} \bar{c}_N^T &= c_N^T - c_B^T A_B^{-1} A_N \\ &= (-1, -3) - (0, 0) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ -1 & 1 \end{bmatrix} = (-1, -3) \end{aligned}$$

Next,

$$x_B = b = A_B^{-1} b$$

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \end{bmatrix}$$

Say we select $j = 2$ to enter the basis.

$$d_2 = A_B^{-1} A_2$$

$$\begin{bmatrix} d_{23} \\ d_{24} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Thus,

$$k = argmin\{b_3/d_{23}, b_4/d_{24}\} = argmin\{6/3, 1/1\} = 4$$

leaves the basis. So, the next $B = \{3, 2\}$ and $N = \{1, 4\}$ and

$$A_B = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 2 & 0 \\ -1 & 1 \end{bmatrix}.$$

Iteration 2.

$$\begin{aligned}\bar{c}_N^T &= c_N^T - c_B^T A_B^{-1} A_N \\ &= (-1,0) - (0,-3)\begin{bmatrix}1 & -3\\ 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0\\ -1 & 1\end{bmatrix} = (-4,3)\end{aligned}$$

Next,

$$x_B = b = A_B^{-1} b$$
$$\begin{bmatrix}x_3\\ x_2\end{bmatrix} = \begin{bmatrix}b_3\\ b_2\end{bmatrix} = \begin{bmatrix}1 & -3\\ 0 & 1\end{bmatrix}\begin{bmatrix}6\\ 1\end{bmatrix} = \begin{bmatrix}3\\ 1\end{bmatrix}$$

We select $j = 1$ to enter the basis.

$$d_1 = A_B^{-1} A_1$$
$$\begin{bmatrix}d_{13}\\ d_{12}\end{bmatrix} = \begin{bmatrix}1 & -3\\ 0 & 1\end{bmatrix}\begin{bmatrix}2\\ -1\end{bmatrix} = \begin{bmatrix}5\\ -1\end{bmatrix}.$$

Thus,

$$k = argmin\{b_3/d_{13}\} = 3$$

leaves the basis. So, the next $B = \{1,2\}$ and $N = \{3,4\}$ and

$$A_B = \begin{bmatrix}2 & 3\\ -1 & 1\end{bmatrix}, \quad A_N = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

Iteration 3.

$$\begin{aligned}\bar{c}_N^T &= c_N^T - c_B^T A_B^{-1} A_N \\ &= (0,0) - (-1,-3)\begin{bmatrix}1/5 & -3/5\\ 1/5 & 2/5\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = (4/5,3/5)\end{aligned}$$

and hence the optimal solution is

$$x_B = b = A_B^{-1} b$$
$$\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}1/5 & -3/5\\ 1/5 & 2/5\end{bmatrix}\begin{bmatrix}6\\ 1\end{bmatrix} = \begin{bmatrix}3/5\\ 8/5\end{bmatrix}$$

Proposition 6. *If $\bar{\mathbf{c}}^T = \mathbf{c}^T - \mathbf{c}_B^T A_B^{-1} A \geq 0$ then the solution is optimal.*

Proof. Consider the dual of the problem:

$$\begin{aligned}&\max \mathbf{y}^T \mathbf{b}\\ &\text{s.t. } \mathbf{y}^T A \leq \mathbf{c}^T\end{aligned} \tag{1}$$

which in the language of basic and nonbasic solutions we can write as:

$$\begin{aligned}&\max \mathbf{y}^T \mathbf{b}\\ &\text{s.t. } \mathbf{y}^T A_B \leq \mathbf{c}_B^T\\ &\quad\quad \mathbf{y}^T A_N \leq \mathbf{c}_N^T\end{aligned}$$

Observe that $\mathbf{c}^T - \mathbf{c}_B^T A_B^{-1} A \geq 0$ implies that $\mathbf{y}^T = \mathbf{c}_B^T A_B^{-1}$ is a feasible solution to the dual problem. The value of this solution in the dual objective is:

$$\mathbf{c}_B^T A_B^{-1} \mathbf{b}.$$

Now observe that for a feasible solution $\mathbf{x}$ we have that $A\mathbf{x} = \mathbf{b}$, which in the language of basic feasible solutions, is:

$$A\mathbf{x} = A_B \mathbf{x}_B + A_N \mathbf{x}_N = A_B \mathbf{x}_B$$

since $\mathbf{x}_N = 0$. So a feasible solution respects $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ and therefore the value in the primal objective for such a solution is:

$$\mathbf{c}_B^T A_B^{-1} \mathbf{b}.$$

And therefore when $\mathbf{c}^T - \mathbf{c}_B^T A_B^{-1} A \geq 0$ we have that the primal and dual objectives have the same value. From the strong duality theorem this implies that the solution must be optimal. □

Remarks. What is the running time of the Simplex algorithm? Is it even finite? What if the algorithm repeatedly cycles over the same set B of basis indices? Observe that the algorithm is well-defined only after we specify the tie-breaking rules to be used for selecting the indices that leave and enter the basis (Steps 7 and 12). One way to break ties is by picking the least indices among the possible choices. This rule, due to Robert Bland, provably avoids cycling thereby ensuring that the algorithm is finite. It remains open to design variations of the tie-breaking rules so that the total number of iterations performed by the Simplex algorithm is polynomial in the number of variables.