

## Lecture 11: Simplex Algorithm: Determining a neighbour of greater cost

Lecturer: *Sundar Vishwanathan*  
COMPUTER SCIENCE & ENGINEERING

Scribe: *Rajhans Samdani*  
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

### 1 Simplex Algorithm

First have a relook at the simplex algorithm :

1. Start at an extreme point.
2. Move to a “neighbour” of greater cost if one exists and repeat the same procedure with it. If no such neighbour exists then exit with this point as the optimum point.

### 2 The notion of directions

Lets consider an example

EXAMPLE 1 Consider two half-planes in 2-d space given by

$$3x + 4y = 12$$

$$7x + 3y = 6$$

We have to find out their point of intersection and the direction of the vectors  $v_1$  and  $v_2$  “along” the lines. It turns out in this case that if the coordinates of the point of intersection are  $(x, y)$  then we can find out  $x$  and  $y$  by solving the equation

$$\begin{bmatrix} -7 & 3 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \end{bmatrix}$$

Also the direction of potential neighbours can be determined to be the vectors  $\begin{bmatrix} 4 \\ -3 \end{bmatrix}$  and  $\begin{bmatrix} -3 \\ -7 \end{bmatrix}$ , respectively. If we write  $v_1$  and  $v_2$  as columns of a matrix  $B$ ,  $\begin{bmatrix} -3 & -7 \\ 4 & -3 \end{bmatrix}$  then we observe that  $AB$  is a diagonal matrix with negative entries on the diagonal.  $\square$

We generalize this to  $n$  dimensions. Consider the usual linear optimization problem with the constraint  $Ax \leq b$  and the cost function  $c^T x$ . Consider any of its extreme points  $x_o$ . The point  $x_o$  satisfies the equation

$$A'x_o = b' \tag{1}$$

where  $A'$  is the matrix formed by certain  $n$  linearly independent rows of the matrix  $A$  and  $b'$  is a vector having corresponding coefficients from  $b$ . Now, from  $x_o$  there are  $n$  lines or directions coming out which “connect” it to its neighbours. We wish to find out the vectors corresponding to these directions.

Let the matrix  $A'$  be written as

$$\begin{bmatrix} A'_1 \\ A'_2 \\ \vdots \\ A'_n \end{bmatrix}$$

where  $A'_i$  is the  $i^{\text{th}}$  row. Now consider a point  $x_i$  on any line  $l_i$  coming out of  $x_o$ . Clearly this line is determined by the intersection of certain  $n - 1$  rows out of the  $n$  linearly independent rows of  $A'$ . Let  $x_i$  be such that

$$A'_j x_j = b'_j \quad \forall j, 1 \leq j \leq n, j \neq i \quad (2)$$

$$A'_j x_j < b'_j \quad j = i \quad (3)$$

The direction of the vector along the line  $l_i$  from the point  $x_o$  towards  $x_i$  is  $x_i - x_o$ . Now we construct a matrix, having as columns all such vectors for  $i = 1, 2, \dots, n$ ,

$$Z = [x_1 - x_o, \quad x_2 - x_o, \quad \dots, \quad x_n - x_o]$$

Now consider

$$A'Z = [A'x_1 - A'x_o, \quad A'x_2 - A'x_o, \quad \dots, \quad A'x_n - A'x_o] \quad (4)$$

Given the properties of  $x_i$ , we know that

$$(A'x_i)^j = (A'x_o)^j \quad j \neq i \quad (5)$$

$$(A'x_i)^j < (A'x_o)^j \quad j = i \quad (6)$$

So it is clear that the  $i^{\text{th}}$  row of the matrix  $A'Z$  has a negative entry at the  $i^{\text{th}}$  position and zeroes at all other positions which means that matrix  $A'Z$  is a diagonal matrix with negative entries on the diagonal. Clearly enough we can multiply each column of the matrix  $Z$  with a suitable **negative** constant such that

$$A'Z = I, \quad (7)$$

where  $I$  is an identity matrix. And now since we only care about the sign of a direction vector and not its magnitude (which basically means we can ignore the magnitude of the constants multiplied with each column of  $Z$  to get  $A'Z = I$ ) so we can say the following about the directions coming out of the point  $x_o$

**THEOREM 1** *The direction vectors are the columns of the negative of the inverse of matrix  $A'$ .*

### 3 Finding neighbouring points with higher cost

One of the steps in the simplex algorithm says “move to the neighbour with larger cost”. Once we know the direction towards each neighbour, it is very easy to find out which of the neighbours have higher cost. This is because of the fact that if one of the neighbours  $x_i$  has higher cost than the current extreme point  $x_o$ , then every point on the line segment joining  $x_i$  and  $x_o$  has cost higher than  $x_o$ . Lets prove this.

**PROOF:** Let  $x'$  be any arbitrary point on the line segment joining  $x_o$  and  $x_i$  such that  $x' \neq x_o$ . Now, we know that any point on the line segment joining  $x_o$  and  $x_i$  can be written as a convex combination of these two. So,

$$x' = \lambda x_i + (1 - \lambda)x_o \quad (8)$$

, where  $\lambda > 0$ . So we get,

$$c^T x' = \lambda c^T x_i + (1 - \lambda)c^T x_o \quad (9)$$

or,

$$c^T x' = \lambda(c^T x_i - c^T x_o) + c^T x_o \quad (10)$$

And since  $c^T x_i > c^T x_o$  so clearly  $c^T x' > c^T x_o$

□

So choose any point on the line segment joining  $x_i$  and  $x_o$  given by a column of  $(-A')^{-1}$  and see if it has higher cost than  $x_o$ . If it has, then the neighbour  $x_i$  will also have higher cost than  $x_o$  otherwise not.

After having found out whether a particular direction fetches higher cost or not, now we move on to actually finding that neighbouring point. Given the direction vector  $v_i$  from  $x_o$  to  $x_i$ , any point  $x'$  on the line segment joining these two can be written as

$$x' = x_o + \epsilon v_i \quad \epsilon \geq 0 \quad (11)$$

where  $\epsilon$  is suitably chosen so that the point  $x'$  doesn't shoot past  $x_i$ . This can be expressed in the constraint

$$A(x_o + \epsilon v_i) \leq b \quad (12)$$

We know that the point  $x' = x_o + \epsilon v_i$  satisfies

$$A'_j x' = b'_j \quad \forall j \neq i, \quad 1 \leq j \leq n \quad (13)$$

So we keep increasing the value of  $\epsilon$  until  $x'$  starts satisfying

$$A_k x' = b_k \quad (14)$$

where  $A_k$  is linearly independent of all rows of  $A'$ . This  $x'$  will be the required neighbour. Hence,

$$\epsilon = \min_t \frac{b_t - A_t x_o}{A_t v_i}, \quad (15)$$

where  $t$  ranges over all those rows of  $A$  which do not belong to  $A'$ .

The neighbouring point can also be found in another way.

1. We take  $n - 1$  linearly independent rows of  $A'$  which decide one particular direction.
2. Then we add to them any row from the matrix  $A$  such that this row is linearly independent of all the  $n$  rows of the matrix  $A'$  and then we find out the point determined by this new set of rows.
3. If this point is feasible, that is it satisfies  $Ax \leq b$  then this is the required neighbouring point, else we repeat this process with other rows of the matrix  $A$  until we find the neighbouring point.