Lecture 2

October 18, 2021

1 Linear programming

We want to solve linear programming problem in standard form:

$$\operatorname{argmin}_{x\in S}\langle c, x\rangle$$

where $S = \{x : Ax = b, x \geq 0\}$. To simplify presentation we will assume that rows of A are linearly independent. This does not reduce generality. Namely, we can choose maximal linearly independent set T of rows of A. Such set forms a basis of space spanned by rows of A, so we can express all other rows as linear combinations of rows from T. Consequently, we can compute Ax knowing only coordinates corresponding to rows in T, so equations corresponding to rows not in T are either redundant (if computed value is the same as right hand side) or the whole system has no solutions. If system $Ax = b$ has no solutions we can stop. Otherwise removing from A redundant rows we get system with linearly independent rows.

1.1 Recall idea of simplex algorithm

Let $J \subset \{1, \ldots, k\}$ and $\operatorname{card}(J) = l = \operatorname{rank}(A)$. We say that solution to $Ax = b$ is basic solution corresponding to J if $x_j \neq 0$ implies $j \in J$ and columns A_j of A for $j \in J$ are linearly independent (so they form basis of image of A).

Lemma 1.1 *If linear programming problem has optimal solution, then it also has basic optimal solution. If linear programming problem has feasible solution, then it also has basic feasible solution.*

Second idea: for given J basic coordinates are uniquely determined by non-basic coordinates. This allows rewriting goal function so that new function is equal to old on S and basic coordinates of new c are zero.

1.2 Simplex algorithm

Of course $l \leq k$. For convenience we may renumber coordinates so that $J = \{1, \ldots, l\}$. Then we can write A in block form $A = [B, N]$ where B is formed from first l columns of A and N from the other. Similarly we can write arbitrary

$x \in \mathbb{R}^k$ as (x_B, x_N) where x_B gives basic coordinates of x (that is coordinates with numbers from J) and x_N nonbasic. Since rows of A are linearly independent, so A has l rows. Also B has l rows. Since B has l linearly independent columns B is invertible.

Now

$$Ax = Bx_B + Nx_N$$

so equation $Ax = b$ becomes

$$Bx_B + Nx_N = b.$$

We multiply this by B^{-1} and rearrange to get

$$x_B = B^{-1}b - B^{-1}Nx_N.$$

Writing $c = (c_B, c_N)$ we can rewrite goal function $\langle c, x\rangle$ as

$$\langle c, x\rangle = \langle c_B, x_B\rangle + \langle c_N, x_N\rangle = \langle c_B, B^{-1}b - B^{-1}Nx_N\rangle + \langle c_N, x_N\rangle$$
$$= h + \langle r, x\rangle.$$

where $r = (0, c_N - (B^{-1}N)^T c_B)$ and $h = (c_B, B^{-1}b)$.

Lemma 1.2 *If $r \geq 0$, then basic solution corresponding to J is optimal.*

Proof: In basic solution the x_N part is zero. In any other feasible solution y we have $y_N \geq 0$, so

$$\langle c, y\rangle - \langle c, x\rangle = (h + \langle r, y_N\rangle) - (h + \langle r, x_N\rangle) = \langle r, y_N\rangle \geq 0.$$

□

So, if $r \geq 0$, then we can stop computation since we found optimal solution.

Assume that i-th coordinate of r is negative. Then $\langle r, e_i\rangle < 0$ where e_i is i-th element of standard basis of $\mathbb{R}^k$. We want to replace $x_N = 0$ by $\alpha(e_i)_N$, that is we need to replace $x_B = B^{-1}b$ by

$$B^{-1}b - \alpha B^{-1}N(e_i)_N = x_B - \alpha B^{-1}Ae_i.$$

That is we replace x by $y = x - (\alpha B^{-1}Ae_i, 0) + \alpha e_i$, so for positive α we have

$$\langle c, y\rangle - \langle c, x\rangle = \langle r, y\rangle = \langle r, \alpha e_i\rangle < 0$$

so value of goal function is smaller. So we need to find maximal α such that y is feasible, that is $y \geq 0$. By construction nonbasic coordinates of y are nonnegative so we need only to look at basic coordinates, that is we need

$$x_B - \alpha B^{-1}Ae_i \geq 0.$$

Or simpler

$$x_B \geq \alpha B^{-1}Ae_i.$$

If all coordinates of x_B are positive, we just take minimal quotient of coordinates of x_B by corresponding coordinate of $B^{-1}Ae_i$ (we skip coordinates that would give division by zero or negative number). If all coordinates of $B^{-1}Ae_i$ are zero or negative, then goal function is unbounded from below.

Note that with the choice of α as above one of basic coordinates of y, say j will be zero. So we can remove j from basic set J and add i instead, that is replace J by $J - \{j\} \cup \{i\}$.

Also, columns of A with numbers from new J will be linearly independent, so y will be basic feasible solution corresponding to new J.

Our assumption above that $J = \{1, \ldots, l\}$ was only to simplify notation. We can still form matrices B from columns of A in J and N from columns not in J for arbitrary J. Other calculations above work similarly.

So given feasible basic set J we get the following algorithm

1. Compute B^{-1}, $B^{-1}N$, x and r corresponding to J
2. Find i such that i-th coordinate of r is negative
3. If no such i then return x as optimal solution
4. Find maximal α such that $x_B \geq \alpha B^{-1}Ae_i$, if α can be arbitrarily large return informing that goal function is unbounded from below
5. If only possible α is zero we have degeneracy, handle this.
6. Otherwise compute $x_B - \alpha B^{-1}Ae_i$ and find zero coordinate j of it.
7. Replace J by $J - \{j\} \cup \{i\}$ and goto 1.

1.3 Simplex algorithm, improvement

In practice there is no need to use original matrix A, we can replace it by $B^{-1}A$ and replace b by $B^{-1}b$. Also, we can replace c by r. After that once we compute new set J we can incrementally update $B^{-1}A$, $B^{-1}b$ and r. More precisely, we join $B^{-1}b$ to $B^{-1}A$ and add r as an extra row. Since in each step we add only a single new element to J at each step current B has only one nontrivial column and is very easy to invert. Moreover, inverse has similar form, so we can reduce matrix multiplication to multiplication by vector and addition.

1.4 Simplex algorithm, example

Consider the following linear programming problem:

$$\begin{array}{rl} \text{minimize} & 5x_1 + 3x_2 + 4x_3 + 2x_4 + x_5 \\ \text{subject to} & 4x_1 - x_2 + 2x_3 - 3x_4 = 12 \\ & -2x_1 + 3x_2 + 2x_4 + 3x_5 = 9 \\ & \text{and } x \geq 0. \end{array}$$

We have

$$[A, b] = \begin{pmatrix} 4 & -1 & 2 & -3 & 0 & 12 \\ -2 & 3 & 0 & 2 & 3 & 9 \end{pmatrix}$$

Choose initial basic set $J = \{1, 2\}$. Then we have

$$B = \begin{pmatrix} 4 & -1 \\ -2 & 3 \end{pmatrix}, \qquad B^{-1} = \frac{1}{10} \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}.$$

so $B^{-1}b$ is $(4.5, 6) \geq 0$ so $x = (4.5, 6, 0, 0, 0)$ is a basic solution corresponding to basic set J. Next,

$$B^{-1}[A, b] = \begin{pmatrix} 1 & 0 & 0.6 & -0.7 & 0.3 & 4.5 \\ 0 & 1 & 0.4 & 0.2 & 1.2 & 6 \end{pmatrix}.$$

Finally we get $c_B = (5, 3)$, $c_N = (4, 2, 1)$ and

$$(B^{-1}N)^T = \begin{pmatrix} 0.6 & 0.4 \\ -0.7 & 0.2 \\ 0.3 & 1.2 \end{pmatrix}$$

(the last matrix can be read from $B^{-1}[A, b]$).

Consequently, nonbasic part of r is

$$c_N - (B^{-1}N)^T c_B = (-0.2, 4.9, -4.1)$$

We can collect what we computed up to now in table $\tilde{A}$ below:

$$\begin{pmatrix} 1 & 0 & 0.6 & -0.7 & 0.3 & 4.5 \\ 0 & 1 & 0.4 & 0.2 & 1.2 & 6 \\ 0 & 0 & -0.2 & 4.9 & -4.1 & \end{pmatrix}.$$

Since most negative element of r has index 5 we chose $i = 5$. Comparing column number 5 above, that is $(0.3, 1, 2)$ with column number 6 that is $B^{-1}b = (4.5, 6)$ we see that maximal possible α is 5, so $y = (3, 0, 0, 0, 5)$, that is as j in the step 6 of simplex algorithm we have to take 2.

Next, we put $J_2 = \{1, 5\}$ and go back to step 1 of second iteration of simplex algorithm.

Now, we have

$$B_2 = \begin{pmatrix} 1 & 0.3 \\ 0 & 1.2 \end{pmatrix}.$$

This matrix has special form: it differs from identity matrix only in second column, so we can immediately write down its inverse:

$$B_2^{-1} = \begin{pmatrix} 1 & -\frac{3}{12} \\ 0 & \frac{10}{12} \end{pmatrix}.$$

Since the inverse differs from identity matrix only in second column, we can simplify computation of $(B_2)^{-1}$ time two first rows of $\tilde{A}$ and get

$$\begin{pmatrix} 1 & -\frac{1}{4} & \frac{1}{2} & -\frac{3}{4} & 0 & 3 \\ 0 & \frac{5}{6} & \frac{1}{3} & \frac{1}{6} & 1 & 5 \end{pmatrix}$$

Note that columns 2, 3 and 4 of matrix above give matrix $B_2^{-1}N_2$. Basic part r_B of vector r from previous step is $(0, -4.1)$ so computing $(B_2^{-1}N_2)^T r_B$ just multiplies second row of $B_2^{-1}N_2$ by -4.1, so we get $(-\frac{41}{12}, -\frac{41}{30}, -\frac{41}{60})$, so $r_2 = (0, \frac{41}{12}, \frac{7}{6}, \frac{67}{12}, 0)$. Since all coordinates of r_2 are nonnegative, this is optimal solution.

If it were not the last step, then we could collect what we computed up to now in table $\tilde{A}_2$ below:

$$\begin{pmatrix} 1 & -\frac{1}{4} & \frac{1}{2} & -\frac{3}{4} & 0 & 3 \\ 0 & \frac{5}{6} & \frac{1}{3} & \frac{1}{6} & 1 & 5 \\ 0 & \frac{41}{12} & \frac{7}{6} & \frac{67}{12} & 0 & \end{pmatrix}.$$

and proceed to step 4 of second iteration.

1.5 Simplex algorithm, feasible point

In the previous example there were happy coincidence that basic set $J = \{1, 2\}$ gave us feasible solution. In general finding such set is almost as difficult as solving linear programming problem. To have obvious basic feasible solution we can add two extra variables to the problem, extending A by diagonal matrix having entries form $\{1, -1\}$. In the previous example we would get

$$[A, b] = \begin{pmatrix} 4 & -1 & 2 & -3 & 0 & 1 & 0 & 12 \\ -2 & 3 & 0 & 2 & 3 & 0 & 1 & 9 \end{pmatrix}$$

(since both 12 and 9 are positive we take positive sign in added matrix). Then $x = (0, 0, 0, 0, 0, 12, 9)$ is a basic feasible solution corresponding to basic set $\{6, 7\}$.

To eliminate added coordinates we can change goal function. In two phase method we use auxiliary goal vector $c_a = (0, 0, 0, 0, 0, 1, 1)$, that is entries corresponding to original coordinates are 0, while entries corresponding to added coordinates are 1. If original problem have feasible solution, then extending it with zeros we get feasible solution of new problem giving value 0 for goal function. Since new goal function is nonnegative this is optimal solution. On the other hand, if optimal value of new goal function is zero, it means that added variables in optimal solution have value 0, so by dropping added coordinates we get feasible solution of original problem.

1.6 Simplex algorithm, termination

Assuming nondegeneracy at each step of simplex algorithm values of goal function will be strictly decreasing, so all basic feasible points produced by simplex algorithm will be distinct. There is only finitely many basic feasible points, less then 2^k (namely at most on basic feasible point for each l-element subset of $[1, \ldots, k]$). So after finite number of steps simplex algorithm must terminate at optimal solution.

1.7 Simplex algorithm, complexity

Exponential estimate for number of steps is quite bad. However, there are examples where in exact arithmetic simplex algorithm really performs exponential number of steps, so in general this estimate can not be improved.

Fortunately, random linear programming problem with high probability requires number of steps linear in dimension of the problem.

In actual machine computations usually rounding error causes behaviour similar to random problems.

1.8 Simplex algorithm, degeneracy

In case of degeneration parameter α equals 0, so x is unchanged. However, since one of basic coordinates of x is 0 we can remove it from basic set and add a new one. This causes recomputation of various quantities like in normal step of simplex algorithm, in particular we will get new value of r.

When implemented naively, such approach may led to cycles. However, for careful choice of coordinates one can show that eventually either we will get $r \geq 0$ or we will get positive α (so decrease value of goal function).

1.9 Simplex algorithm, convexity

Linear programming problem is an example of convex optimization problem.

Definition. Closed line segment joining x and y is set of points of form

$$tx + (1 - t)y$$

where $t \in [0, 1]$.

Definition. Set S is a convex set when for all $x_1, x_2 \in S$ closed line segment joining x_1 and x_2 is contained in S.

Equivalently set S is a convex set when for all $x_1, x_2 \in S$ and $t \in [0, 1]$ also $tx + (1 - t)y \in S$.

Lemma 1.3 *Solution set S of system of linear equations and inequalities is a convex set.*

Proof: Consider equations $Ax = a$ and inequalities $Bx \geq b$. Assume $x_1, x_2 \in S$, that is $Ax_1 = a$, $Ax_2 = a$, $Bx_1 \geq b$, $Bx_2 \geq b$. Let $t \in [0, 1]$, then also $(1 - t) \in [0, 1]$. Put $y = tx_1 + (1 - t)x_2$. We have

$$Ay = tAx_1 + (1 - t)Ax_2 = ta + (1 - t)a = a$$

and

$$By = tBx_1 + (1 - t)bx_2 \geq tb + (1 - t)b = b$$

so $y \in S$. □

1.10 Simplex algorithm, geometric view

By previous lemma feasible set of a linear programming problem is a convex set. This set may be slightly more general than convex polyhedron, it is so called convex polyhedral set.

For linear programming problem is standard form basic feasible solutions are exactly vertices of feasible set.

Simplex algorithm moves from given vertex to one of connected vertices (with lower value of goal function).

1.11 Simplex algorithm, more general version

In practice converting linear programming problem to standard form may significantly increase size of the problem. So it is preferable to treat directly problems in nonstandard forms. Form previous discussion it follows that we should look at vertices of feasible set. Vertices are points where several inequality constraints turns into equalities such that obtained set of equalities (both original equalities and equalities obtained from constraints) is of full rank and has unique solution. So natural generalization replaces basic feasible points by points where some subset of inequalities is replaced by equations in such a way that resulting system is of full rank.

2 Further reading 1

Simplex method is described in chapter 3 of book: David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, Springer 2008.

In subchapter 5.2 of book above there is example showing exponential number of steps for a specific variant of simplex method.

3 Convex sets

Recall that set S is convex if and only if for each two points $x, y \in S$ line segment joining x and y is contained in S.

Example: Closed halfspace $\{x : (a, x) \leq b\}$, open halfspace $\{x : (a, x) < b\}$, interval $[a, b]$, ...

Example: One point set $\{a\}$ where a is a fixed element of $\mathbb{R}^n$.

Lemma 3.1 *Let Δ be an arbitrary family of convex subsets of $\mathbb{R}^n$. Then intersection*

$$K = \bigcap_{S \in \Delta} S$$

is a convex set.

Proof: Let $x, y \in K$, $t \in [0, 1]$, $S \in \Delta$, $z = tx + (1 - t)y$. Since S is convex $z \in S$. Since S is arbitrary for all $S \in \Delta$ we have $z \in S$. Hence $z \in K$. Since t

is arbitrary, for all $t \in [0,1]$ we have $tx + (1-t)y \in K$, so line segment joining x and y is contained in K, so K is convex. $\square$

Example: intersection of halfspaces is convex. In particular, triangle is convex.

4 Further reading 2

Stephen Boyd, Lieven Vandenberghe, Convex Optimization, Chapters 2 and 3.