10-725: Optimization **Fall 2012**

Lecture 14: October 11

Lecturer: Geoff Gordon/Ryan Tibshirani *Scribes: Zitao Liu*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

This lecture's notes illustrate some uses of various LaTeX macros. Take a look at this and imitate.

14.1 Previous class review

Notations: For the Linear Programming(LP) problems, we have m constraints, and n variables.

$\mathbf{A} : R^{m\times n}, \mathbf{b} : R^m, \mathbf{c} : R^n, \mathbf{x} : R^n$

14.1.1 Inequality form of LP

$$[\min \mathit{or} \max]\ \mathbf{c}^T\mathbf{x} \text{ s.t. } \mathbf{Ax} \leq \mathbf{b}$$

where $m \geq n$

14.1.2 Standard form of LP

$$[\min \mathit{or} \max]\ \mathbf{c}^T\mathbf{x} \text{ s.t. } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0$$

where $m \leq n$

14.2 Nonsingular standard form of LP

We follow the same notation used in 14.1.

$$[\min \mathit{or} \max]\ \mathbf{c}^T\mathbf{x} \text{ s.t. } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0$$

where $m \leq n$ and $\mathbf{A}$ has full row rank.

Change LP to its nonsingular form: keep dropping rows until $\mathbf{A}$ becomes a full row rank matrix.

14.3 Solving LPs

Optimum of LPs is always achieved at a corner/vertex(= a 0-face).

14.3.1 Naive algorithm

Intuition: enumerate all possible bases.

Algorithm 1 Naive algorithm solving LPs

```
Transform LP to its nonsingular standard form
// Iterate through all subsets of $m$ variables out of $n$ variables
repeat
  // Check the subset for full rank and feasibility
  isFullRank = checkFullRank()
  isFeasible = checkFeasibility()
  if isFullRank && isFeasible then
    // Compute objective function's value
    objVal = computeObjValue();
    // for max objective function, flip the direction if we have min objective function.
    if objVal > bestObjVal then
      bestObjVal = objVal;
    end if
  end if
until explore all the $\binom{n}{m}$ subsets.
```

14.3.2 Simplex algorithm

Intuition: reduce the search space of basis in Native algorithm. Maybe the neighbors of good bases are also good.

Neighbors: Two bases are called neighbors if they share *m-1* variables.

Algorithm 2 Simplex algorithm solving LPs

```
Transform LP to its nonsingular standard form
Start from a feasible basis and its tableau
repeat
  Pick a non-basis variable whose coefficient in negative in objective function.
  Pivot it into basis and get neighboring basis
until all non-basis variables have positive coefficients in objective function
```

Example:

$$
\begin{aligned}
&\quad \max\ z = 2x + 3y \\
&s.t \\
&\quad x + y \le 4 \\
&\quad 2x + 5y \le 12 \\
&\quad x + 2y \le 5 \\
&\quad x \le 4 \\
&\quad x, y \ge 0
\end{aligned}
$$

Solving it using simplex algorithm:

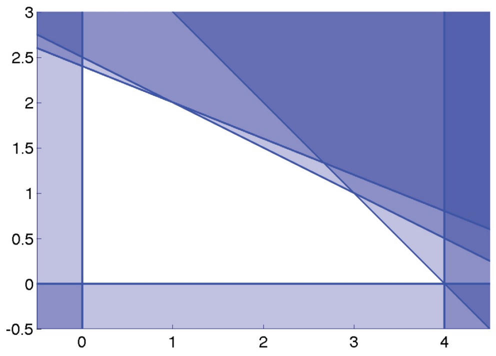

Figure 14.1: Graphical representation of the example LP problem.

1. Transform it into a nonsingular standard form.

$$
\begin{aligned}
&\max\ z = 2x + 3y \\
s.t \\
&x + y + s = 4 \\
&2x + 5y + t = 12 \\
&x + 2y + u = 5 \\
&x + v = 4 \\
&x, y, s, t, u, v \geq 0
\end{aligned}
$$

2. Start from a feasible basis and its tableau. See Table 14.1.

Table 14.1: Simplex Tableau for basis $< s, t, u, v >$.

x	y	s	t	u	v	z	RHS
1	1	1	0	0	0	0	4
2	5	0	1	0	0	0	12
1	2	0	0	1	0	0	5
1	0	0	0	0	1	0	4
-2	-3	0	0	0	0	1	0

3. Change to a feasible neighbor basis and its tableau. Move y in and t out. See Table 14.2.

4. Change to a feasible neighbor basis and its tableau. Move x in and u out. See Table 14.3.

5. Change to a feasible neighbor basis and its tableau. Move t in and s out. See Table 14.4.

6. All the non-basis variables have the positive coefficients in the objective function, which means we have achieved the optimum values. So the $z_{max} = 9$.

Table 14.2: Simplex Tableau for basis $< s, y, u, v >$.

x	y	s	t	u	v	z	RHS
0.4	1	0	0.2	0	0	0	2.4
0.6	0	1	-0.2	0	0	0	1.6
0.2	0	0	-0.4	1	0	0	0.2
1	0	0	0	0	1	0	4
-0.8	0	0	0.6	0	0	1	7.2

Table 14.3: Simplex Tableau for basis $< s, y, x, v >$.

x	y	s	t	u	v	z	RHS
1	0	0	-2	5	0	0	1
0	1	0	1	-2	0	0	2
0	0	1	1	-3	0	0	1
0	0	0	2	-5	1	0	3
0	0	0	-1	4	0	1	8

Table 14.4: Simplex Tableau for basis $< t, y, x, v >$.

x	y	s	t	u	v	z	RHS
1	0	2	0	-1	0	0	3
0	1	-1	0	1	0	0	1
0	0	1	1	-3	0	0	1
0	0	-2	0	1	1	0	1
0	0	1	0	1	0	1	9

14.4 Degeneracy

14.4.1 What is degeneracy?

We all know that simplex algorithm starts at one feasible corner point and moves to an adjacent corner point by increasing the value of a non-basic variable with a negative value in the z-row. In the ordinary case, every time we change the basis, the objective value will be improved, but in some cases, the objective function value does not increase at all, which means the objective value and solution does not change, but there is an exiting variable. We call this situation is "degeneracy".

The degeneracy will happen when on of the RHS coefficients in the tableau is zero. Basically, we can say there are two types of degeneracies:

1. **Ordinary degeneracy**: stay at the same corner(exiting variable was already 0)
2. **Dual degeneracy**: move to another corner with the same objective value(coefficient of entering variable in objective was 0)

14.4.2 Degeneracy examples

The following is a degeneracy example in 2D.

$$\begin{aligned}
&\max_{x,y} x + y \\
s.t.& \\
&x + y + u = 4 \\
&2x + 5y + v = 12 \\
&x + 2y + w = 16/3 \\
&x, y, u, v, w \geq 0
\end{aligned}$$

And the graphical illustration is in Figure 14.2. As we can see, there are 3 lines cross the same point and if we write down the tableau and we can see that even we change the basis according to the simplex algorithm, we can not increase the objective value.

Here is another example about degeneracy in 3D. Its graphical illustration is in Figure 14.3. If we look at the red circled vertex, we can see that there are 5 faces cross at that red point in our 3D setting. So there are $\binom{5}{3}$ different basis we can choose for that vertex, which yields a degeneracy case.

14.4.3 Solutions for degeneracy

Degeneracy is bad since it may lead our simplex algorithm to an infinite cycle. There are two directions to solve this problem:

1. Use an anti-cycling rules to break the cycle.
2. Perturbation approach(add tiny random numbers to the objective function and RHS).

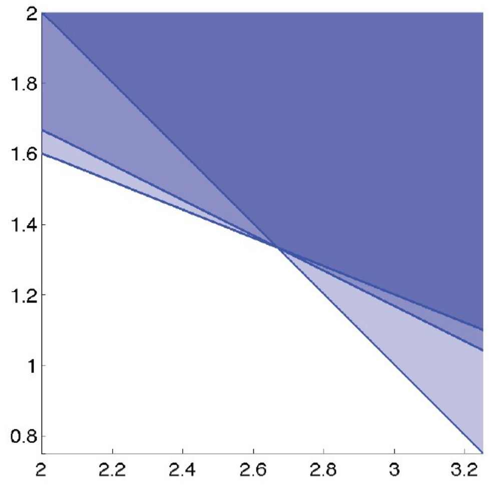

Figure 14.2: A degeneracy example in 2D.

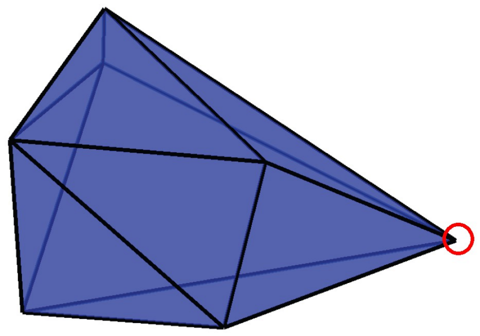

Figure 14.3: A degeneracy example in 3D.