MAT 3007 – Optimization

Complexity and Duality Theory

Lecture 07

June 22th

Andre Milzarek

iDDA / CUHK-SZ

Repetition

Recap: Simplex Tableau

The simplex tableau provides a practical and efficient way to implement the simplex method.

$c^\top - c_B^\top A_B^{-1} A$	$-c_B^\top A_B^{-1} b$
$A_B^{-1} A$	$A_B^{-1} b$

- Each step in the simplex method can be carried out as some manipulation. on the numbers in the simplex tableau.
- The upper left part is the reduced costs, and the upper right corner is the negative of the objective value.
- The lower part is a transformation of the constraints. In particular, the right hand side column shows the current basic variables.

Recap: Simplex Tableau

Starting from a canonical form, each iteration in the simple tableau operation consists of:

1. Determine optimality: look at the top row (the reduced costs):
   - If the costs are nonnegative, then the BFS is already optimal.
   - Otherwise choose the smallest index j such that $\bar{c}_j < 0$ as the incoming basis (pivot column).
2. Perform the minimal ratio test (between the current basic variables and the incoming column – positive entries only).
3. Choose the row that attains the smallest ratio to be the pivot row and determine the outgoing basis (smallest index rule).
4. Update the tableau:
   - Divide each entry in the pivot row by the pivot element.
   - Add proper multiples of the new pivot row to each other rows, such that all other elements in the pivot column become zeros (including the top row).

Recap: Simplex Tableau

Comments:

- If there is no positive entry in the pivot column, then the problem is unbounded.
- The update of the tableau should always include the top row and also the right hand side b.
- One can still proceed normally even if there is a degenerate solution in the process. As long as the smallest index rule (Bland's rule) is used, it is guaranteed that we will stop at an optimal solution in a finite number of steps.

Two-Phase Method in Simplex Tableau

For the simplex tableau, when there is no obvious initial basic feasible solution, we can use the two-phase method.

Phase I:

- The auxiliary problem is not in a canonical form: Although the constraint contains an identity matrix (which is the initial basis), the corresponding objective coefficients are not 0's!
- We need to compute the top row (reduced costs):
  - For basic part, the reduced costs are 0.
  - For nonbasic part, $\bar{c}_j = c_j - c_B^\top A_B^{-1} A_j = -\mathbb{1}^\top A_j$, so the jth reduced cost is the negative of the sum of that column.
- This also applies to the initial objective value, which equals the negative of the sum of the right hand side vector.

Two-Phase Method in Simplex Tableau

Suppose we have finished the Phase I simplex tableau operations, resulting in an optimal solution $(x^*, 0)$ with optimal value 0.

To carry out Phase II in the simplex tableau:

1. Drop all the columns associated with the auxiliary variables.
2. Change the top row to the reduced costs corresponding to the original problem by using the reduced cost formula:

$$\bar{c}_j = c_j - c_B^\top A_B^{-1} A_j$$

   and compute the current (negative) objective value.
3. Continue the simplex tableau to solve the original problem.

If the optimal solution of the auxiliary problem is degenerate, then replace the basic indices of included auxiliary variables with indices of original variables to form a basis.

Example

$$\begin{array}{lrrrrcr}
\text{minimize} & x_1 & +x_2 & +x_3 & & & \\
\text{subject to} & x_1 & +2x_2 & +3x_3 & & = & 3 \\
& & -4x_2 & -9x_3 & & = & -5 \\
& & & +3x_3 & +x_4 & = & 1 \\
& x_1, & x_2, & x_3, & x_4 & \geq & 0
\end{array}$$

First, make b positive and construct the auxiliary problem:

$$\begin{array}{lrrrrrrrcr}
\text{minimize} & & & & & x_5 & +x_6 & +x_7 & & \\
\text{subject to} & x_1 & +2x_2 & +3x_3 & & +x_5 & & & = & 3 \\
& & 4x_2 & +9x_3 & & & +x_6 & & = & 5 \\
& & & +3x_3 & +x_4 & & & +x_7 & = & 1 \\
& x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7 & \geq & 0
\end{array}$$

Example Continued

Construct the initial tableau for the auxiliary problem

B	-1	-6	-15	-1	0	0	0	-9
5	1	2	3	0	1	0	0	3
6	0	4	9	0	0	1	0	5
7	0	0	3	1	0	0	1	1

Carry out the simplex method (Step 1):

B	0	-4	-12	-1	1	0	0	-6
1	1	2	3	0	1	0	0	3
6	0	4	9	0	0	1	0	5
7	0	0	3	1	0	0	1	1

Example Continued

Step 2:

B								
	0	0	-3	-1	1	1	0	-1
1	1	0	-3/2	0	1	-1/2	0	1/2
2	0	1	9/4	0	0	1/4	0	5/4
7	0	0	3	1	0	0	1	1

Step 3:

B								
	0	0	0	0	1	1	1	0
1	1	0	0	1/2	1	-1/2	1/2	1
2	0	1	0	-3/4	0	1/4	-3/4	1/2
3	0	0	1	1/3	0	0	1/3	1/3

This is optimal for the auxiliary problem. $x = (1, 1/2, 1/3, 0)$ is a BFS for the original problem ($B = \{1, 2, 3\}$).

Example Continued

B								
B	0	0	0	0	1	1	1	0
1	1	0	0	1/2	1	-1/2	1/2	1
2	0	1	0	-3/4	0	1/4	-3/4	1/2
3	0	0	1	1/3	0	0	1/3	1/3

We drop all the columns for auxiliary variables. Then we recompute the reduced cost for the original problem for $B = \{1, 2, 3\}$:

$$\bar{c}^\top = c^\top - c_B^\top A_B^{-1} A = (0, 0, 0, -1/12).$$

We also need to compute the current objective value: 11/6.

Now the simplex tableau becomes:

B	0	0	0	-1/12	-11/6
1	1	0	0	1/2	1
2	0	1	0	-3/4	1/2
3	0	0	1	1/3	1/3

Example Continued

Then we continue from the new simplex tableau:

B					
	0	0	0	-1/12	-11/6
1	1	0	0	1/2	1
2	0	1	0	-3/4	1/2
3	0	0	1	1/3	1/3

The next pivot:

B	0	0	1/4	0	-7/4
1	1	0	-3/2	0	1/2
2	0	1	9/4	0	5/4
4	0	0	3	1	1

This is optimal. The optimal solution is $x = (1/2, 5/4, 0, 1)$. The optimal value is $7/4$.

Simplex Method Summary

We have completed our discussions of the simplex method.

- The idea of simplex method (search among the BFS and neighbors) and its justifications.
- The algebraic procedures.
- The simplex tableau.
- Other issues about the simplex method: initialization, degeneracy, etc.

Complexity Theory

How Efficient is the Simplex Method?

We have learned that the simplex method terminates within a finite number of iterations.

$\rightsquigarrow$ But finite could still be large ...

- We know that the number of iterations should not exceed the number of BFS, which is bounded by $C(n, m)$. But this is a large number!

To answer this question, we discuss some basics about complexity theory.

Complexity Theory

Complexity

We count the number of arithmetic operations (addition: $+$, subtraction: $-$, multiplication: $\cdot$, division: $\div$, comparisons, etc.) that an algorithm needs to solve a problem. This is called the complexity of the algorithm.

When we analyze the complexity of an algorithm, we consider the worst-case instance for that algorithm:

- For a specific instance, an algorithm may finish much faster than its complexity suggests.

Examples:

- Find the largest element among n numbers has complexity n.
- Add two $m \times n$ matrices has complexity mn.
- Multiply two $n \times n$ matrices in a naive way has complexity n^3.

Polynomial-Time Algorithms

The main watershed in complexity is whether an algorithm has a polynomial complexity or not:

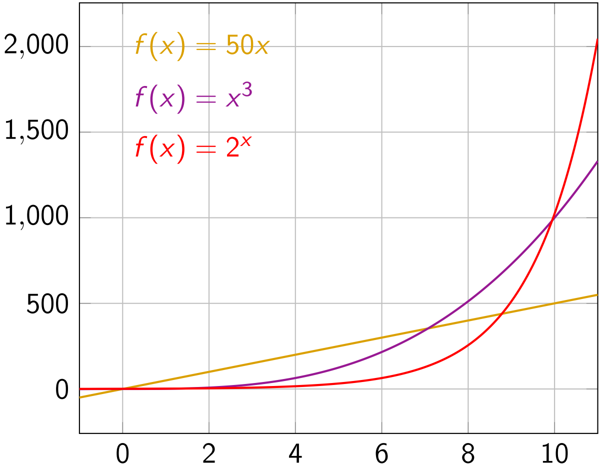
- We call an algorithm a polynomial-time algorithm if the number of arithmetic operations required to solve any instance of the problem is bounded by a polynomial of the input size.
- A polynomial-time algorithm is usually deemed to be a practical algorithm.
- Otherwise, it is usually deemed impractical.

We introduce a notation which is useful when studying complexity:

Definition: Landau Notation

We write $g(n) \sim O(f(n))$ if and only if there exists nonnegative constants c_1 and c_2 such that $|g(n)| \leq c_2 f(n)$ for all $n \geq c_2$.

Polynomial vs Non-Polynomial



- Polynomial vs Exponential Function.

Polynomial-Time Algorithms

Here are some known polynomial algorithms:

- Gaussian elimination for matrix inversion: $O(n^3)$.
- Fast method for matrix inversion: $\sim O(n^{2.373})$.
- Naive method for sorting n numbers: $O(n^2)$.
- Merge sort for sorting n numbers: $O(n \log n)$.

Here are some non-polynomial algorithms:

- Enumeration method for traveling salesman problem: $O(n!)$.
- Dynamic programming for TSP: $O(2^n n^2)$.

Complexity of a Problem

Definition: P

For a problem, if there exists a polynomial-time algorithm, then we say it is a polynomial-time solvable problem or – short – in P.

There exists a class of problems that researchers haven't found any polynomial-time algorithm (e.g., the traveling salesman problem):

- This class is called NP-hard problems.
- In fact, if one can show that one NP-hard problem has a polynomial-time algorithm, then a large subclass of them (NP-complete) will have polynomial-time algorithms.

Million Dollar Problem:

- Is there a polynomial-time algorithm for NP-hard problems (NP=P)?

Complexity of the Simplex Method

We consider a standard LP with n variables and m constraints.

What is the complexity of the simplex method?

- We can configure the simplex method by selecting different rules for the incoming/outgoing basis indices. Each will result in a different algorithm and may have different complexity.

Current Result: None of the investigated rules will result in a polynomial-time algorithm.

- In other words, for each proposed configuration, there is an instance such that the simplex method with the specified configuration requires exponentially many iterations to find the optimal solution.
- It remains an open question whether there is a configuration such that the simplex method is a polynomial-time algorithm.

Complexity of Simplex Method

For all the known simplex variants, problem instances have been found such that the algorithm will run with exponential iterations in terms of n and m to solve that problem (Klee and Minty Example).

- Very bad news from theoretical point of view!

However ...

- This is only the worst-case performance!
- The practical performance of the simplex method is quite good. Typically, it needs a small multiple of m iterations.
- One can prove that on average, the simplex method stops within a polynomial number of iterations.
- It is still one of the most widely-used algorithms for LPs.

Complexity of Linear Problems

Although the simplex method may not be a polynomial-time alg., this does not mean that LPs are not polynomial-time solvable.

- The simplex method is just one possible alg. for solving LPs.

Whether LP is in P was a major problem in the 20th century. It was finally solved by Soviet Union mathematician Khachiyan in 1979, who showed a first polynomial-time algorithm for LPs. His method is called the ellipsoid method (see Chapter 8 in Bertsimas's book).

- Although the ellipsoid method is a polynomial-time algorithm, it is extremely slow in practice. Therefore, it is more of a theoretical contribution than a practical one.

Can we do better both theoretically and practically?

The Answer is: Yes!

Later in the course, we will introduce another algorithm for LPs called the interior point method.

- Before we get to that, we need to first study an important theoretical framework for LPs: Duality theory.

Duality Theory for LPs

Duality Theory: Motivation

We consider the standard LP (m constraints, n variables):

$$\begin{aligned} \text{minimize}_x \quad & c^\top x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

We can write it as:

$$\begin{aligned} \text{minimize}_x \quad & c^\top x + \max_{y \in \mathbb{R}^m} y^\top (b - Ax) \\ \text{subject to} \quad & x \geq 0. \end{aligned}$$

Why?

- If $Ax \neq b$, then we can find y such that $y^\top(Ax - b) = \infty$, which can not be optimal. Hence, we implicitly enforce the constraint $Ax = b$.

Continue ...

$$\min_{x \geq 0} \quad \max_{y} \quad c^\top x + y^\top (b - Ax)$$

⇝ Now we assume that we can exchange max and min (we will justify it later). Then the problem becomes:

$$\max_{y} \quad b^\top y + \min_{x \geq 0} \; x^\top (c - A^\top y)$$

We claim that this is equivalent to:

$$\begin{aligned} \text{maximize}_y \quad & b^\top y \\ \text{subject to} \quad & A^\top y \leq c. \end{aligned}$$

Why?

$$\min_{x \geq 0} x^\top (c - A^\top y) = \begin{cases} 0 & \text{if } A^\top y \leq c, \\ -\infty & \text{if } A^\top y \not\leq c. \end{cases}$$

Two Linear Programs ...

$$\begin{aligned} \text{minimize}_x \quad & c^\top x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

and

$$\begin{aligned} \text{maximize}_y \quad & b^\top y \\ \text{subject to} \quad & A^\top y \leq c \end{aligned}$$

We call them dual to each other. If we call one the primal problem, then the other one is called the dual problem.

- ▶ We call y the dual variables (to the first LP).
- ⇝ By our derivation, they should have the same optimal value (provided the exchange of min and max is valid).

Dual Problem

We have derived our first pair of dual problems.

- Duality theory is very important for linear optimization (and for more general optimization problems as well).
- The dual problem carries much useful information for the primal problem ($\rightsquigarrow$ later).
- The dual problem can also be used to solve the primal problem.

Different Duality Forms

We have derived the dual problem of the standard form. What if we want to find the dual problem of:

$$\begin{aligned} \text{minimize} \quad & c^\top x \\ \text{subject to} \quad & Ax \geq b \end{aligned} \tag{1}$$

Of course, one can first transform (1) to the standard form and then derive the dual. But we can directly apply our previous arguments.

The primal problem can be written as:

$$\min_x \quad c^\top x + \max_{y \geq 0} \; y^\top (b - Ax)$$

Why?

$$\max_{y \geq 0} \; y^\top (b - Ax) = \begin{cases} 0 & \text{if } Ax \geq b, \\ \infty & \text{if } Ax \ngeq b. \end{cases}$$

Continue ...

Assume we can exchange the order of max and min, then:

$$\max_{y \geq 0} \quad b^\top y + \min_x \; x^\top (c - A^\top y).$$

This is further equivalent to:

$$\begin{array}{rl} \text{maximize}_y & b^\top y \\ \text{subject to} & A^\top y = c \\ & y \geq 0 \end{array}$$

This is the dual problem of (1).

General Dual Problems

Primal			Dual		
min	$c^\top x$		max	$b^\top y$	
s.t.	$a_i^\top x \geq b_i,$	$i \in M_1,$	s.t.	$y_i \geq 0,$	$i \in M_1$
	$a_i^\top x \leq b_i,$	$i \in M_2,$		$y_i \leq 0,$	$i \in M_2$
	$a_i^\top x = b_i,$	$i \in M_3,$		y_i free,	$i \in M_3$
	$x_j \geq 0,$	$j \in N_1,$		$A_j^\top y \leq c_j,$	$j \in N_1$
	$x_j \leq 0,$	$j \in N_2,$		$A_j^\top y \geq c_j,$	$j \in N_2$
	x_j free,	$j \in N_3,$		$A_j^\top y = c_j,$	$j \in N_3$

- $a_i^\top$ is the ith row of A, A_j is the jth column of A.
- Each primal constraint corresponds to a dual variable.
- Each primal variable corresponds to a dual constraint.
- Equality constraints always correspond to free variables.

An Example

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & x_1 + x_2 \geq 5 \\ & x_1 - x_2 \leq 3 \\ & x_1 \geq 0,\ x_2 \in \mathbb{R} \text{ (free)} \end{aligned}$$

Constructing the dual problem:

- $\rightsquigarrow$ Associate constraint 1 with dual variable y_1.
- $\rightsquigarrow$ Associate constraint 2 with dual variable y_2.

$$\begin{aligned} \text{maximize} \quad & 5y_1 + 3y_2 \\ \text{subject to} \quad & y_1 + y_2 \leq 1 \\ & y_1 - y_2 = 2 \\ & y_1 \geq 0,\ y_2 \leq 0 \end{aligned}$$

To Memorize: Table & Signs

Primal	minimize	maximize	Dual
constraints	$\geq b_i$	≥ 0	variables
	$\leq b_i$	≤ 0	
	$= b_i$	free	
variables	≥ 0	$\leq c_j$	constraints
	≤ 0	$\geq c_j$	
	free	$= c_j$	

Example

Consider the following problem:

$$\begin{array}{lccccc} \text{minimize} & x_1 & +2x_2 & +3x_3 & \\ \text{subject to} & -x_1 & +3x_2 & & = 5 \\ & 2x_1 & -x_2 & +3x_3 & \geq 6 \\ & & & x_3 & \leq 4 \\ & x_1 \geq 0 & x_2 \leq 0 & x_3 \text{ free} & \end{array}$$

The dual is:

$$\begin{array}{lccccc} \text{maximize} & 5y_1 & +6y_2 & +4y_3 & \\ \text{subject to} & -y_1 & +2y_2 & & \leq 1 \\ & 3y_1 & -y_2 & & \geq 2 \\ & & +3y_2 & +y_3 & = 3 \\ & y_1 \text{ free} & y_2 \geq 0 & y_3 \leq 0 & \end{array}$$

One More Example

Recall the support vector machine problem. The primal problem is:

$$\begin{array}{lll} \text{minimize}_{w,b,t} & \sum_{i=1}^m t_i & \\ \text{subject to} & y_i(x_i^\top w + b) + t_i \geq 1, & \forall\ i = 1, \ldots, m \\ & t_i \geq 0 & \forall\ i = 1, \ldots, m \end{array}$$

with variables $w \in \mathbb{R}^n$, $b \in \mathbb{R}$, and $t \in \mathbb{R}^m$.

⇝ Associate u_i to each of the constraints.

- The dual problems has $n + m + 1$ constraints.
- Defining the data matrix $X = \begin{pmatrix} x_1 & x_2 & \ldots & x_m \end{pmatrix} \in \mathbb{R}^{n \times m}$, the primal constraints can be written compactly:

$$\begin{pmatrix} \mathrm{diag}(y) X^\top & y & I \end{pmatrix} \begin{pmatrix} w \\ b \\ t \end{pmatrix} \geq \mathbb{1}.$$

Example Continued

The dual constraint becomes

$$\begin{pmatrix} X\mathrm{diag}(y) \\ y^\top \\ I \end{pmatrix} u = \begin{pmatrix} X\mathrm{diag}(y)u \\ y^\top u \\ u \end{pmatrix} \begin{array}{llll} \Longrightarrow & = 0 & (w \text{ free}) \\ \Longrightarrow & = 0 & (b \text{ free}) \\ \Longrightarrow & \le \mathbb{1} & (t \ge 0) \end{array}$$

with additional constraints $u \ge 0$.

The dual problem is

$$\begin{array}{rl} \text{maximize}_u & \sum_{i=1}^m u_i \\ \text{subject to} & X\mathrm{diag}(y)u = 0 \\ & y^\top u = 0 \\ & 0 \le u_i \le 1, \qquad \forall\ i = 1, \ldots, m \end{array}$$

Example: Summary

SVM - Primal Problem:

$$\begin{aligned} \text{minimize}_{w,b,t} \quad & \mathbb{1}^\top t \\ \text{subject to} \quad & \mathrm{diag}(y)X^\top w + yb + t \geq \mathbb{1} \\ & t \geq 0 \end{aligned}$$

SVM - Dual Problem:

$$\begin{aligned} \text{maximize}_{u} \quad & \mathbb{1}^\top u \\ \text{subject to} \quad & X\mathrm{diag}(y)u = 0 \\ & y^\top u = 0 \\ & u \geq 0,\ u \leq \mathbb{1} \end{aligned}$$

- X is the data matrix, y are the labels (± 1).

Questions?