CPSC 500

Fundamentals of Algorithm Design and Analysis

Nov.18, 2013

Lecturer: Prof. Will Evans

Scriber: Andrey Novitskiy

Linear Programming Duality

1 Overview

In this lecture, basic concepts of Linear Programming Duality were covered. Using previously reviewed profit maximization problem (recall chocolate example) as well as new examples of different two-players zero sum games, we showed how each LP problem can be redefined into a new one, which leads to the same result.

2 Chocolate example

Recall from the previous lectures, we had two types of chocolate with different prices (x_1, x_2) . We wanted to maximize our profit function, having set of constraints on how we can sell it. As a result, we had following LP problem :

$$
\begin{aligned}
&\max(x_1 + 6x_2) \\
&x_1 \leq 200 \\
&x_2 \leq 300 \\
&x_1 + x_2 \leq 400 \\
&x_1, x_2 \geq 0
\end{aligned}
$$

Now we want to create minimization problem, having our LP. To achieve that, let's create new variables (multipliers) y_1, y_2, y_3 and multiply each constraint by corresponding variable:

$$
\begin{array}{ll}
y_1 & x_1 \le 200 \\
y_2 & x_2 \le 300 \\
y_3 & x_1 + x_2 \le 400
\end{array}
$$

After multiplying and adding all three we have:

$$(y_1 + y_2)x_1 + (y_2 + y_3)x_2 \le 200y_1 + 300y_2 + 400y_3$$

This is a **bound** for our problem. We want to find such y_1, y_2, y_3, that give the best upper bound subject to previous constraints, therefore we need to solve new LP :

$$
\begin{aligned}
&\min 200y_1 + 300y_2 + 400y_3 \\
&y_1 + y_2 \ge 1 \\
&y_2 + y_3 \ge 6 \\
&y_1, y_2, y_3 \ge 0
\end{aligned}
$$

This is Dual LP, which corresponds to Primal LP (Chocolate problem). If we find feasible solutions for both primal and dual, they both give optimum to our problem. In this example, we have:

Primal : $(x_1, x_2) = (100, 300)$

Dual : $(y_1, y_2, y_3) = (0, 5, 1)$

Both give optimal value of 1900.

In general, both primal and dual LP can be written in matrix form:

Primal LP:

$$
\begin{aligned}
&\max c^T x \\
&Ax \le b \\
&x \ge 0
\end{aligned}
$$

Dual LP:

$$\min y^T b$$
$$y^T A \le c^T$$
$$y \ge 0$$

or

Primal LP:

$$\max c_1 x_1 + \ldots + c_n x_n$$
$$a_{i1} x_1 + \ldots + a_{in} \le b_i \quad for \; i \in I$$
$$a_{i1} x_1 + \ldots + a_{in} = b_i \quad for \; i \in E$$
$$x_j \ge 0 \quad for \;\; j \in N$$

Dual LP:

$$\min b_1 y_1 + \ldots + b_m y_m$$
$$a_{1j} y_1 + \ldots + a_{mj} \ge c_j \quad for \;\; j \in N$$
$$a_{1j} y_1 + \ldots + a_{mj} = c_j \quad for \;\; j \notin N$$
$$y_i \ge 0 \quad for \;\; i \in I$$

In general case, we have **Duality Theorem**:

If a linear program has a bounded optimum, then so does it's dual, and the two optimum values coincide.

We can see it in our example.

3 Zero-sum Games

A good example of this is Rock-Paper-Scissors. Assuming we have to players: Row and Column, each of two can make a move from the set $\{r, p, s\}$. The result of each move can be represented in payoff matrix G. If Row make *i*-th move and Column *j*-th, then Column pays to Row the amount in *(i,j)* position:

G		Column		
		r	p	s
Row	r	0	-1	1
	p	1	0	-1
	s	-1	1	0

Let's assume players do their moves repeatedly. The strategy of each player can be represented as a set of probabilities to make certain move: $x=(x_1,x_2,x_3)$ for Row and $y=(y_1,y_2,y_3)$ .for Column. Since on every round the probability of row making *i*-th move and column making *j*-th move is $x_i y_j$, the expected payoff is:

$$\sum_{i,j} G_{i,j}\cdot x_i y_j$$

Row wants to maximize it and Column tries to minimize it. Let's assume that Row plays completely random strategy: x = (1/3, 1/3, 1/3). Then the average payoff is:

$$\sum_{i,j} G_{i,j}\cdot x_i y_j=\sum_{i,j} G_{i,j}\cdot\frac{1}{3}y_j=\sum_j y_j\left(\sum_i \frac{1}{3}G_{i,j}\right)=\sum_j y_j\cdot 0=0\,.$$

Therefore, playing completely random strategy, Row forces expected payoff of 0, no matter what Column does, what means that Column cannot hope for negative payoff. Symmetrically, if Column plays randomly, Row cannot hope for positive payoff. Here we showed what we know intuitively: the best strategy in this game is being completely random.

4 Nonsymmetric game

In the previous example, we got such results mainly because of payoff matrix's structure – sum of elements in each row or column there is zero. What if we have a game with more complex payoff matrix?

Let's play a game, where two moves are allowed for each player:

G		Col	
		a	*b*
Row	*a*	*3*	*-1*
	b	*-2*	*1*

Assuming Column knows Row's strategy $x = (x_1, x_2)$, there is always an optimal strategy for Column: either move *a* with payoff $3x_1 - 2x_2$ or *b* with $-x_1 + x_2$. Therefore, Row knows that Column's best response will achieve $z = \min\{3x_1 - 2x_2, -x_1 + x_2\}$ payoff and Row wants to pick x_1 and x_2 to maximize *z*. Since for *fixed* x_1 and x_2 $z = \min\{3x_1 - 2x_2, -x_1 + x_2\}$ equivalent to:

$$\begin{aligned}
&\max z \\
&z \le 3x_1 - 2x_2 \\
&z \le -x_1 + x_2
\end{aligned}$$

we can write LP for Row's $x = (x_1, x_2)$ strategy:

$$\begin{aligned}
&\max z \\
&-3x_1 + 2x_2 + z \ge 0 \\
&x_1 - x_2 + z \le 0 \\
&x_1 + x_2 = 1 \\
&x_1, x_2 \ge 0
\end{aligned}$$

The same can be done for Column's (y_1, y_2):

$$\begin{aligned}
&\min w \\
&-3y_1 + y_2 + w \ge 0 \\
&2y_1 - y_2 + w \ge 0 \\
&y_1 + y_2 = 1 \\
&y_1, y_2 \ge 0
\end{aligned}$$

This two LP's are dual to each other and have the same optimum *V*. Therefore, if Row plays (x_1, x_2), this will lead to V no matter what Column does and vice versa, if Column's playing (y_1, y_2) yields V.

This example can be generalized to arbitrary games and yields min-max theorem:

$$\max_x \min_y \sum_{i,j} G_{i,j} x_i y_j = \min_y \max_x \sum_{i,j} G_{i,j} x_i y_j ,$$

that shows the existence of mixed strategies that are optimal for both players and achieve the same value.

5 Summary

In this lecture we've covered the Duality of Linear Programming problems. We showed how, at first sight, problems far from linear programming, like Zero-Sum Games, have LP duality in it. The next topic is NP-complete problems.

6 References

Wikipedia http://en.wikipedia.org/wiki/Linear_programming

S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, p. 220 - 226