Lecture 9

Uri Feige

June 12, 2002, + June 19

1 Some more on LP duality

We will show interesting consequences that can be derived (rather easily) using LP duality.

1.1 The minimax theorem

A simple example of a two-person zero-sum game is "stone, paper, scissors" (which I suppose is the translation of "even niyar umisparaim"). In such games, there are two players. Each player has a set of possible moves, and both players move simultaneously (without prior knowledge of the other player's move). A payoff matrix (known to both players) specifies the payoff to the column player as a function of the two moves. The payoff to the row player is $-A$. (The gain of one player is equal to the loss of the other).

What is the best move in such a game? In general, there is no best move. The payoff that results from a move changes based on the move of the other player. If one player knows the move of the other player, then its easy for him to play the game optimaly. The question is how to play the game when one does not know the move of the other player. Clearly, keeping your own move secret from the other player is important. The "mathematical" way of generating secrets is via randomness, and this calls for a randomized strategy, or in game theory terminology, a *mixed* strategy. What should this strategy optimize? The conservative view is to seek a strategy that carries the least amount of risk. Namely, whatever the move of the other player is, the expected payoff is maximized. This is a maxmin approach: maximize (over your choice of moves) the minimum (over all possible moves of the other player) expected payoff.

Finding the optimal mixed strategy can be formulated as a linear program. Let A be a payoff matrix for the column player whose columns specify moves for the maximizing player and rows specify moves for the minimizing player. The payoff matrix to the row player is then $-A$. Let x be a vector of probabilities for the possible moves of the maximizing player. Then we have the following LP:

maximize v (the maxmin payoff)
subject to
$1^T x = 1$ (the probabilities sum up to 1),
$Ax \geq v$ (for every move of the row player, the expected payoff is at least v),
$x \geq 0$ (probabilities are nonnegative).

For the minimizing player, let y be the vector of probabilities for his possible moves. Then we have:

maximize u (the maxmin payoff)
subject to
$1^T y = 1$ (the probabilities sum up to 1),
$-A^T y \geq u$ (for every move of the column player, the expected payoff to the row player is at least u),
$y \geq 0$ (probabilities are nonnegative).

The famous minimax theorem of Von Neumann says that the values of the optimal conservative mixed strategies for the two players are equal (to the negative of each other). The mixed strategies are in equilibrium. Each player can tell the other player his own mixed strategy (which is not the same as revealing your move in advance), and still, no (conservative) player will have an incentive to change his own mixed strategy.

The proof of the minimax theorem is a simple consequence of linear programming duality. (Historically, the minimax theorem was proved before LP duality.) To see this, observe that v and u are variables. Change the main constraint for the column player to $-Ax + v \leq 0$. For the row player, change the objective function to **minimize** $-u$, and the main constraint to $-A^T y - u \geq 0$.

The complementary slackness conditions say that a move has probability 0 unless it is optimal against the strategy of the other player.

Another point worth noticing is that the support of the mixed strategy of each player need not be larger than the number of moves available to the other player (as the optimum is attained at a vertex).

1.2 Yao's Lemma

A randomized algorithm can be viewed as a distribution over deterministic algorithms. The success probability (or expected running time) of a randomized algorithm is its success probability over a worst case input. Yao's lemma (an immediate application of the minimax theorem) says that there is a worst case distribution over inputs such that the average success probability of the best deterministic algorithm over this distribution is equal to the success probability of the best randomized algorithm over the worst case input. This principle is often used in order to prove lower bounds for randomized algorithms. One exhibits a distribution on which no deterministic algorithm does well.

1.3 Vertex cover

We shall use LP duality in a natural way to develop a polynomial time algorithm for vertex cover on bipartite graphs. (Note that we also developed such an algorithm last semester.)

The vertex cover polytope is the convex hull of all vertex covers. If we could characterize it by a polynomial number of linear constraints (or even an exponential number with a separation oracle), then we could solve vertex cover.

Vertex cover can be formulated as:
minimize $\sum_{i=1}^{n} v_i$ *subject to*:
$v_i + v_j \geq 1$, for each edge (v_i, v_j),
$v_i \in \{0, 1\}$, for each vertex v_i.
NP-hardness of vertex cover implies that integer programming is NP-hard.

We relax the integer program to a linear program by replacing the $v_i \in \{0,1\}$ constraints by nonnegativity constraints $v_i \geq 0$. The gap between the value of the optimal solution for the integer program and its relaxed fractional program is known as the *integrality gap*.

The fact that it is arbitrarily close to 2 can be seen on a clique. The fact that it is no worse than 2 can be seen by rounding fractional values of 1/2 and above to 1.

The dual of the linear program is as follows:

maximize $\sum_{\{(i,j)\in E\}} y_{ij}$ *subject to*:

$\sum_{\{j|(i,j)\in E} y_{ij} \leq 1$, for every vertex i,

$y_{ij} \geq 0$, for each edge (i,j).

This can be viewed as a relaxation of the matching integer program. In general, it does not define the matching polytope. For example, on a triangle a fractional solution may give each edge a value of 1/2, giving value higher than the maximum matching.

Let us analyse the structure of solutions to the dual LP. By adding all constraints, we see that the value of the optimal solution is at most $n/2$ (because each edge is counted twice). A basic feasible solution is supported by at most n edges.

Proposition 1 *The LP has a half integral optimal solution.*

Proof: Consider a basic feasible solution. Consider an arbitrary connected component with respect to the positive edges in this bfs. If it is k vertices, then it must have at most k nonzero edges. If it has a vertex of degree 1, include its edge in the solution, and remove all edges connected to its endpoint. The value of the solution does not decrease. If all vertices are of degree 2, then the connected component is a cycle. Give each edge value 1/2. □

Corollary 2 *In a bipartite graph, the above matching LP has an optimal integer solution.*

Proof: All cycles are of even length. Replace by alternating 0 and 1. □

Hence to solve the dual LP in case of a bipartite graph, one can just run a combinatorial algorithm for maximum matching. Does the primal LP also have an optimal solution that is integer in bipartite graphs? If yes, then we can find the value of the minimum vertex cover just by running a matching algorithm.

Definition 1 *A matrix is totally unimodular if each square submatrix of it has a determinant that is either 0,1, or -1.*

Lemma 3 *If A is totally unimodular and A and b are integral, then all vertices of the polytope are integer. It particular, the optimal solution (with respect to any linear objective function) is integer.*

Proof: Consider the constraints that are tight in the optimal solution, and let x_B be the basic variables tha correspond to them. Then $x_B = B^{-1}b$. By Cramer's rule and total unimodularity of A, x_B is integer. ($x_N = 0$.) □

The matrix of constraints considered in this section is the vertex-edge incidence matrix.

Proposition 4 *The vertex-edge incidence matrix in bipartite graphs is totally unimodular.*

Proof: Consider a square submatrix. If it has a 0 column or row, then its determinant is 0. So we can assume that each row and column contain at least one 1 entry. If there is a row (or column) with exactly one 1 entry, remove this row and the column on which the 1 lies. The determinant changes only by a multiplicative factor of ± 1. Eventually, either the determinant is seen to be ± 1, or every column has exactly two 1 entries, implying that also every row has exactly two 1 entries. So the matrix represents a collection of disjoint cycles. Each cycle must be of even length. Sum up the rows corresponding to one side of the bipartization, and subtract the rows corresponding to the other side. One gets the 0 vector, implying that the rows are linearly dependent and the submatrix has determinant 0. $\square$

It follows that the vertex cover LP has optimal integer solutions on bipartite graphs. Let us derive such a solution from a maximum matching. By the equality of the value of the objective functions, we should have one vertex per matched edge.

By complementary slackness, an unmatched vertex cannot be in the vertex cover. Hence the other endpoint of every edge incident with it must be. By the other complementary slackness condition, if there is a matched edge touching a vertex in the vertex cover, its other end point cannot be in the vertex cover. Continuing this way, eventually we remain with a subgraph in which all vertices are matched. In it, take one side of the bipartization.

1.4 The max-flow min-cut theorem

The vertex-arc incidence matrix of a directed graph has vertices as its rows, arcs as its columns, and for arc $e = (i, j)$, its column has $+1$ in row i, -1 in row j, and 0 elsewhere.

Proposition 5 *The vertex-arc incidence matrix of a digraph is totally unimodular.*

The proof is left as homework.

In the max flow problem, there is a directed graph with nonnegative capacities b_{ij} on its edges. One has to send a maximum flow from vertex 1 to vertex n, obeying capacity constraints and preservation of flow. Namely, one seeks the solution to the following linear program:

maximize x_{n1}
subject to:
$\sum_j x_{ij} - x_{ji} = 0$ for all $1 \leq i \leq n$,
$x_{ij} \leq b_{ij}$ for all i, j. Here x_{n1} is not constrained.
$x_{ij} \geq 0$ for all i, j.

Proposition 6 *If all capacities are integer, then the maximum flow is integer.*

Proof: We can add nonnegative slack variables to obtain constraints $x_{ij} + z_{ij} = b_{ij}$. The resulting constrain matrix is totally unimodular. $\square$

In the min-cost flow problem, one has to route f units of flow from v_1 to v_n. Every edge has a cost c_{ij} per sending one unit of flow through it. The above LP can be modified by adding the constraint $x_{n1} = f$, and changing the objective function to **minimize** $\sum c_{ij} x_{ij}$. Again we find that the minimum cost flow is integer (though its cost need not be integer, if the c_{ij} are noninteger).

The dual of the max-flow LP can be seen to be as follows. We associate a variable y_i with every vertex and y_{ij} with every arc except for $(n, 1)$. One gets:

minimize $\sum b_{ij} y_{ij}$
subject to
$y_n - y_1 \geq 1$
$y_{ij} + y_i - y_j \geq 0$
$y_{ij} \geq 0$.

Again, total unimodularity applies, and the optimal solution is integer. We may assume without loss of generality that $y_1 = 0$ (shifting weights of all vertices by the same amount does not change the y_{ij} values, and the value of the objective function remains unchanged), and that $y_n = 1$ (multiplying all variables by some $0 < \delta < 1$ cannot increase the value of the objective function). We may further change any y variable with value greater than 1 to 1, because no y_{ij} needs to increase by this. Similarly, we may change any y variable with value smaller than 0 to 0. Finally, no y_{ij} needs to have a value greater than 1. It follows that the optimal solution is in fact a 0/1 solution. It can readily be seen to be of value equal to the minimum directed cut between v_1 and v_n. Hence we have just established the well known theorem that the maximum flow is equal to the minimum cut.