College of Computer & Information Science — Fall 2007
Northeastern University — 14 September 2007
CS G399: Algorithmic Power Tools I — Scribe: Eric Robinson

Lecture Outline:

- Linear Programming: Vertex Definitions
- Linear Programming: Optimal at a Vertex
- LP: Duality
- LP: Duality: Weak Duality
- LP: Duality: Set Cover

This lecture gives an introduction to the geometry of linear programming. We discuss three equivalent definitions of a vertex of a polytope, and then show that there exists an optimal solution to any bounded feasible LP lies at a vertex of the underlying polytope. We then introduce the important concept of LP duality and present an alternative analysis of the greedy set cover algorithm using the idea of dual fitting. References for the material covered in this lecture are [Goe94, Kar91, Vaz03].

1 Linear Programming

Linear Programming was described in the previous class. Recall the standard form:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

1.1 Vertex Definitions

As was discussed the previous class, three definitions of a vertex, x, for a convex polytope, P, can be used:

- $\neg\exists y \neq 0$ s.t. $x + y, x - y \in P$
- $\neg\exists y, z \in P, \alpha \in (0, 1)$ s.t. $\alpha y + (1 - \alpha)z = x$
- $\exists n$ linearly independant equations that are tight at x

The proof that definitions 1 and 2 are equivalent shall be left to the reader. Here we will prove that claims 1 and 3 are equivalent.

1.1.1 1 implies 3

Suppose $\neg 3$. Let A' be the submatrix corresponding to the tighe inequalities, let b' be the corresponding right hand sides. This leads to the inequality $A'x = b'$, where there are remaining equations A'' and right hand sides, $b - b''$. The rank of A' is strictly less than n. This implies that $\exists y \neq 0$ s.t. $A'y = 0$.

Consider $x + \lambda y$, $A'(x + \lambda y) = A'x + \lambda A'y = b'.Find\lambda \geq 0$ s.t. $x + \lambda y, x - \lambda \in P$. Now consider any one of the non-tight inequalities, j. $A_j x > b_j$. Apply λ, getting $A_j(x + \lambda y) = A_j x + \lambda A_j y$. It is known that $A_j x > b_j$. Because $A_j x = b_j$ is not tight, $\lambda A_j y$ must be feasible in one direction for a short distance, and the other direction infinitely. We shall select a distance λ_j equal to that short distance. Finally, we shall select $\lambda = \min_j |\lambda_j|$. Because the associated constraints are not tight, λ cannot be zero. This directly contradicts 1.

1.1.2 3 implies 1

Let A' denote the submatrix for tight inequalities, as was done above. This implies the rank of $A' \geq n$. Suppose $neg1$. This implies $\exists y \neq 0$ s.t. $x + y, x - y \in P$. Consider the fact that $A'(x + y) \geq b', A'(x - y) \geq b'$. This implies that $A'y = 0$, which implies that the rank of A' is less than n. This contradicts the assertion made earlier.

1.2 Optimality at a Vertex

The claim was made in the previous class that an optimal solution must occur at a vertex. More specifically, if the linear program is feasible and bounded, then $\exists v \in Vertices(P)$ s.t. $\forall x \in P, C^T v \leq C^T x$. This statement, along with fundamentally stating that an optimal solution must occur at a vertex, also shows the decidability of solving for the system, as one can simply check all the vertices. A proof follows.

1.2.1 Proof

The proof shall proceed by picking a non-vertex point and showing that there exists as good or better of a solution with at least one less non-tight equation. Hence progress is made towards a vertex by applying this process iteratively. To find a vertex that is optimal this iteration will have to be done at most $n + m$ times.

Suppose $x \in P$ and $x \not\subset Vertices(P)$. This implies $\exists y \neq 0$ s.t. $x + y, x - y \in P$. This in turn implies $(x + y, x - y \geq 0) \rightarrow (x_i = 0 \rightarrow y_i = 0)$. Consider $v = x + \lambda y$. $C^T v = C^T x + \lambda C^T y$. Assume without loss of generality that $C^T y \leq 0$. If this was not the case, the other direction could simply have been selected $(x - y)$.

Now suppose that A' and b' give the set of tight variables, as before. We know that $A'(x + y) \geq b'$. This implies $A'y = 0$, and that $A'(x + \lambda y) = b'$. Once again, we will find the λ_j values for the non-tight equations. There are two possibilities:

- $\exists \lambda_j \geq 0$: In this case, we selecte the smallest such λ_j value, λ. The point $x + \lambda y$ now makes one additional inequality tight while satisfying the property that it keeps the previously tight inequalities tight. And we have made progress towards a solution.

- $\forall_j \lambda_j < 0$: Here, either $C^T y < 0$, in which case the solution is unbounded, or $C^T y < 0$ in which case we use the go to the $\exists \lambda_j \geq 0$ case, as our cost function will remain the same no matter which direction we move in.

1.3 Duality

One can view any minimization linear program as a maximization. Consider the following linear system:

$$\begin{array}{lll} \min & 3x_1 + 2x_2 + 8x_3 & \\ \text{s.t.} & x_1 - x_2 + 2x_3 & \geq & 5 \\ & x_1 + 2x_2 + 4x_3 & \geq & 10 \\ & x_1, x_2, x_3 & \geq & 0 \end{array}$$

Where Z^* is OPT, we know $Z^* = 3x_1^* + 2x_2^* + 8x_3^*$, for some $x_1^*, x_2^*, x_3^* \in P$. By adding two of the inequalities, we arrive at $2x_1 + x_2 + x_3 \geq 15$. Since $x_1^*, x_2^*, x_3^* \geq 0$, we know that $Z^* \geq 15$. But we aren't limited to addition, multiplication is another way the equations can be combined. So how is this new formulation bouned? This is done by using the dual formulation, D of the minimization, which for this problem is:

$$\begin{array}{llll} \max & 5y_1 + 10y_2 & & \\ \text{s.t.} & y_1 + y_2 & \leq & 3 \\ & -y_1 + 2y_2 & \leq & 2 \\ & 2y_1 + 4y_2 & \leq & 8 \\ & y_1, y_2 & \geq & 0 \end{array}$$

The theory of LP duality (sometimes referred to as the Strong Duality Theorem) says that if the primal LP P is bounded and feasible, then the value of the primal LP equals the value of the dual LP.

1.3.1 Weak Duality

Weak duality makes only the claim that the value of the primal LP is at least the value of the dual LP. Consider the primal P and its dual D:

$$\begin{array}{llll|llll} P & & & & D & & & \\ \min & c^T x & & & \max & b^T y & & \\ \text{s.t.} & Ax & \geq & b & \text{s.t.} & A^T y & \leq & c \\ & x & \geq & 0 & & y & \geq & 0 \end{array}$$

Suppose that x^* is an optimal solution to P and y^* is an optimal solution to D. We need only show that $c^T x^* \geq b^T y^*$.

$$\begin{aligned} c^T x^* &\geq (A^T y^*)^T x^* \\ &= {y^*}^T A x^* \\ b^T y^* &\leq {x^*}^T A^T y^* \\ &= ({y^*}^T A x^*)^T \end{aligned}$$

Noting that the last equation of each of these comparison are identical (since the transpose of a scalar is the scalar itself) leads to the desired conclusion.

1.3.2 Set cover analysis using dual fitting

The set cover linear programming problem, P, can be recast into its dual framework, D:

$$\begin{array}{llll|llll} P & & & & D & & & \\ \min & c_s x_s & & & \max & \sum_{e \in U} y_e & & \\ \text{s.t.} & \sum_{S \ni e} x_s & \geq 1 & \forall e \in U & \text{s.t.} & \sum_{e \in s} y_e & \leq c_s & \forall s \in S \\ & x_s & \geq 0 & \forall s \in S & & y_e & \geq 0 & \forall e \in U \end{array}$$

Analysis We seek to construct a dual-feasible solution with a cost close to the integer solution. We shall look at y_e, or $price(e)$, the cost of an element when it is first covered. This corresponds to $\frac{cost(s)}{\text{number of uncovered elements covered}}$ when e is first covered. The cost of the greedy solution is $\sum_{e \in U} y_e$.

The problem with this is that the constraints for the dual are not covered because the greedy solution adds in an additional cost for previously covered elements. So, we will uniformly decrease the y_e's, $y_e = \frac{price(e)}{\alpha}$. The claim is that y is dual feasible if $\alpha = H_n$, where n is the number of elements in U.

To prove this, we need to show $\sum_{e \in s} y_e \leq c_s$. If we considered the elements ordered again, and pick any set $s \in S$, we'll see that a portion of the elements have been covered, and a portion have not. This means that:

$$\begin{aligned} y_{e1} &\leq \frac{c_s}{k} \times \frac{1}{H_n} \\ y_{e2} &\leq \frac{c_s}{k-1} \times \frac{1}{H_n} \\ \vdots &\ \vdots\ \vdots \end{aligned}$$

Or,

$$\begin{aligned} \sum_{e \in S} y_e &\leq \frac{c_s}{H_n} \times \left(\frac{1}{k} + \frac{1}{k-1} + \cdots + 1\right) \\ &= \frac{c_s}{H_n} \times H_k \\ &\leq c_s \end{aligned}$$

So, from this, we can show that the greedy solution is an H_n approximation:

$$\begin{aligned} \text{Cost of greedy algorithm} &= H_n \times cost(y) \\ &\leq H_n \times OPT(D) && \text{proof above} \\ &\leq H_n \times OPT(P) && \text{weak duality} \\ &\leq H_n \times OPT && \text{fractional to integral} \end{aligned}$$

References

[Goe94] M. Goemans. Introduction to Linear Programming. Lecture notes, available from http://www-math.mit.edu/~goemans/, October 1994.

[Kar91] H. Karloff. *Linear Programming*. Birkhäuser, Boston, MA, 1991.

[Vaz03] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2003.