CS 224: Advanced Algorithms Fall 2014

Lecture 15 — October 21, 2014

Prof. Jelani Nelson *Scribe: Matt Rauen*

1 Overview

In the last lecture we covered nearest neighbor search with guest lecturer Piotr Indyk.

In this lecture we cover solving linear programs using the simplex algorithm.

1. Simplex
2. Strong duality/complementary slackness
3. Ellipsoid

2 Standard form

In a general linear program, we want to minimize $c^T x$ such that some set of inequalities are satisfied. We would like to convert this into standard form:

$\min c^T x$ such that $Ax = b$ and $x \geq 0$

$x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, n \geq m$

If we have a maximization problem, we can just flip the sign of c^T and switch it to a minimization problem, since minimizing $-c^T x$ is equivalent to maximizing $c^T x$.

2.1 Getting to standard form

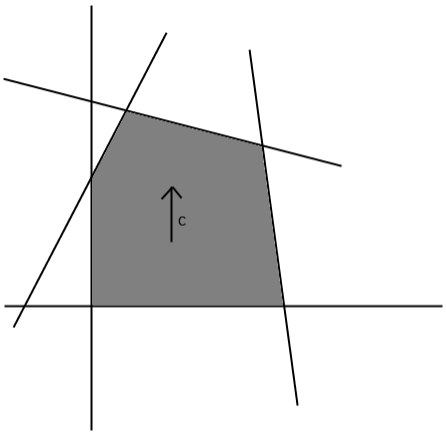
In order to convert a linear program to standard form, we observe the following two facts:

Fact 1. $\langle a_i, x \rangle = b_i \iff \langle a_i, x \rangle \geq b_i$ *and* $-\langle a_i, x \rangle \geq -b_i$

Fact 2. $\langle a_i, x \rangle \leq b_i \iff -\langle a_i, x \rangle \geq -b_i$

Convert all constraints in this way so that we are left with $\geq$ symbols. Then for each constraint i, we define a slack variable $s_i \geq 0$ to represent the difference between the sides of the inequality. Replace $\langle a_i, x \rangle \geq b_i$ with $\langle a_i, x \rangle - s_i = b_i$. Then, to account for the constraint that $x \geq 0$, we can write $x_i = x_i^+ - x_i^-$ where $x_i^+ \geq 0$ and $x_i^- \geq 0$. Note that the number of variables in the new LP is at least the number of constraints, since we are adding a slack variable for each constraint.

2.2 LPs geometrically



The constraints restrict the region of valid points, and c indicates how the objective function changes with x.

3 Simplex Algorithm

Simplex Algorithm reproduced here [1].

Main Idea: Start at some vertex, greedily move to adjacent vertices that improve the objective, and we stop when we are at the optimal vertex.

Let $P = \{x : Ax = b, x \geq 0\}$.

Definition: We say that $x \in P$ is a vertex if $\nexists y \neq 0$ such that $x - y \in P$ and $x + y \in P$.

We claim that finding an x that satisfies those constraints (that is, lies in P) is equivalent to solving the linear program. This is because if we have a guess r for the outcome of the linear program, we can add in the constraint $c^T x \leq r$. Checking if this new LP is satisfiable allows us to bound the minimum, and we can continue via binary search to solve the LP.

This will terminate since the optimal x is a rational number of bounded precision, and our binary search gives you at least a bit of information each time it runs. Here we are operating under the assumptions that the optimal solution is a vertex and that we can obtain the vertices by solving a linear system on A, which is what allows us to bound the precision.

Because of this, finding a starting vertex is difficult, so we will run the simplex algorithm on a different LP in order to find one. This other usage of the simplex algorithm has an easy starting vertex, so we only have to run the simplex algorithm twice, once to find the starting vertex, and once to actually solve our LP.

Starting Vertex LP: Minimize t such that $Ax = (1 - t)b$, $x \geq 0$, and $0 \leq t \leq 1$.

This is not in standard form, since we have the constraint $t \leq 1$, but we can rewrite this via a slack variable s_t. The constraint becomes $s_t + t = 1$ for some $s_t \geq 0$. We have a feasible point given by $x = 0$, $t = 1$, $s_t = 0$.

Question: Why is this a vertex?

If $x-y, x+y \in P$, then y cannot have any mass on x, since if y or $-y$ is negative on any coordinate corresponding to x, we violate $x \geq 0$. It also cannot have any mass on t or s_t since the constraints $t \leq 1$ and $s_t \geq 0$ are already tight.

Definitions:

- x is <u>feasible</u> if $x \in P$
- An LP is <u>feasible</u> if $\exists$ a feasible x
- An LP is <u>infeasible</u> if $\nexists$ a feasible x
- An LP is <u>bounded</u> if $\min c^T x$ over $x \in P$ is $> -\infty$

Claim: If an LP is bounded, then $\forall$ feasible x, $\exists$ vertex $x' \in P$ such that $c^T x' \leq c^T x$.

Proof. If x is a vertex, set $x' = x$. Else, $\exists y \neq 0$ such that $x + y \in P$ and $x - y \in P$, which implies that $A(x+y) = A(x-y) = b$, so $Ay = 0$. Without loss of generality, $c^T y \leq 0$. If this isn't the case, we simply replace y with $-y$. Also, if $c^T y = 0$, then we can assume that there is an index j such that $y_j < 0$ (if not, we can just flip the sign of y, as before).

Case 1: $\exists j$ such that $y_j < 0$.

Look at $x + ty$ for $t > 0$. If $x_i = 0$, then $y_i = 0$. If this were not true, then on this coordinate, one of $x + y$ and $x - y$ would violate the constraint that $x \geq 0$. As t increases, $(x + ty)_j$ goes to 0 for all j such that $y_j < 0$.

There is some $t^* > 0$ such that $x + t^* y \in P$. Pick t^* as large as possible so that $x + t^* y \in P$.

This case can happen at most n times, since every time it happens, our new x has at least one more coordinate that is 0. From then on, that coordinate must be fixed so that we don't violate the constraint that $x \geq 0$.

Question: Why does a $t^* > 0$ exist?

If $y_j < 0$ then $x_j > 0$. This implies that $t \cdot y_j + x_j > 0$ if $t < -\frac{x_j}{y_j}$. Choose t^* to be the minimum such upper bound over j such that $y_j < 0$.

Case 2: $\forall j$, $y_j \geq 0$.

Now $A(x + ty) = Ax + t \cdot Ay = b$, and $\forall t > 0$, $x + ty \geq 0$, but $c^T(x + ty) = c^T x + t \cdot c^T y < 0$ (since otherwise we would be in Case 1). Then $t \to \infty$ demonstrates that the LP is unbounded, which contradicts the assumption that we are in a bounded LP. □

Corollary: If an LP is finite, then $\exists$ a vertex in P achieving OPT.

Corollary: If $P \neq \emptyset$, then it has at least one vertex.

The second follows from the fact that we can choose any objective function, and then there must be a vertex achieving OPT, so in particular there must be a vertex.

Definition: Given $x \in P$, define A_x as the submatrix of A corresponding to $\{j : x_j > 0\} = B_x$.

Claim: x is a vertex $\iff$ columns of A_x are linearly independent.

Note that the column rank and row rank are both m. All redundant rows either make the LP "infeasible" or are automatically satisfied and can be removed. If we have fewer columns than m in A_x, we can pad with more columns to get a square $m \times m$ matrix.

Proof. We prove the contrapositive of both directions.

(i) x is not a vertex implies that the columns of A_x are linearly dependent.

If x is not a vertex, then $\exists y \neq 0$ such that $x + y, x - y \in P$, which implies that $A_y y = 0$, and so $x_i = 0 \implies y_i = 0$. Thus, A_y has dependent columns, and so A_x has dependent columns because A_y is a submatrix of A_x.

(ii) The columns of A_x being linearly dependent implies that x is not a vertex.

If the columns of A are not linearly independent, then there is some $y \neq 0$ such that $A_x y = 0$. We can extend y to n dimensions by making all other coordinates equal to 0. This then implies $\exists y \neq 0$ such that $Ay = 0$. For some $t > 0$, we have that $x + ty, x - ty \in P$ (since y only has mass in places where x has mass, so we can choose sufficiently small t as we did in our selection of t^*). Thus, x is not a vertex. □

4 Implementation

For each vertex $x \in P$, we have some basis $B = \{j : x_j > 0\}$, and $|B| \leq m$. We have the following fact from linear algebra:

Fact 3. $Cz,\ C = (C_1, C_2, \ldots, C_m) \implies Cz = \sum_{i=1}^{m} z_i C_i$.

Then $Ax = b$ implies that $\sum_i x_i A_i = \sum_{i \in B} x_i A_i = A_B x_B$. If $|B| = m$, then A_B is $m \times m$ and has full rank, so it is invertible. Thus, $A_B x_B = b \implies x_B = A_B^{-1} b$. For every vertex v, we then extend its base to have size m while still being linearly independent.

Then we can rewrite the LP as $\min c_B^T x_B + c_N^T x_N$ where $N = [n] \backslash B$ such that $A_B x_B + A_N x_N = b$ and $x_B, x_N \geq 0$. Then for all $x \in P$, we have that $x_B + A_B^{-1} A_N x_N = A_B^{-1} b$. In particular, $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$, and therefore

$$\begin{aligned} c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T A_B^{-1} b - c_B^T A_B^{-1} A_N x_N + c_N^T x_N \\ &= (c_N - A_N^T (A_B^{-1})^T c_B)^T x_N + c_B^T A_B^{-1} b \\ &= \tilde{c}_N^T x_n + c_B^T A_B^{-1} b. \end{aligned}$$

Simplex Algorithm:

1. Maintain current vertex x and its base B.
2. Rewrite LP as above in terms of x_B, x_n, etc.

3. If $\tilde{c}_N \geq 0$, then HALT and return x.

4. If $\exists j \in [n]$ such that $\tilde{c}_j < 0$, then increment x_j as long as we can so that the feasibility of x is maintained.

Annoying Detail: If we start changing a coordinate in x_N, a coordinate in x_B might change, so we change only until some coordinate in x_B hits zero. That is, we keep incrementing x_j as long as $x_B \geq 0$. It might not be possible to increment x_j at all, so we put j into the basis, and evict a coordinate that is already zero. This means that we have reached a new vertex, as a new constraint became tight. Note that it is possible that the "pivoting rule" that we use to determine which basis element to evict contains a cycle.

There is a pivoting rule which provably has no infinite loops. It is known as Bland's rule, and is proved in [2]. No implementation of the simplex algorithm will run in polynomial time, but they tend to be reasonably fast in practice.

References

[1] George Dantzig. Maximization of a linear function of variables subject to linear inequalities. In *The basic George B. Dantzig.* ed. by Richard W. Cottle. 24–32, Stanford: Stanford University Press, 2003.

[2] Robert Bland. New Finite Pivoting Rules for the Simplex Method. *Mathematics of Operations Research*, 2(2):103–107, 1977.