COMS 4995-2: Advanced Algorithms (Spring'20) March 3, 2020

Lecture 13: Linear Programming, Duality

Instructor: *Alex Andoni* Scribes: *Jiaqi Liu, Jianan Yao*

1 System of Linear Equations

Given a matrix $A \in R^{m \times n}$, $b \in R^m$, we are trying to find a solution $x \in R^n$ such that

$$Ax = b$$

We first consider the case where $n = m$ and $det(A) \neq 0$. If $n = m$ and $det(A) \neq 0$, we have already known that

$$x_i = \frac{det(\text{matrix composed entries of } A, b)}{det(A)}$$

via Gaussian Elimination.
Note: More precisely, considering Cramer's Rule,

$$x_i = \frac{det(A_i)}{det(A)}, i = 1, ..., n$$

where A_i is the matrix formed by replacing the i-th column of A by the column vector b.

Claim 1. *Given $n \times n$ square matrix A and vector b of length n, where each element of A and b is an integer with $\leq b$ bits, the solution x to $Ax = b$ is describable with polynomial number of bits.*

Proof. Recall how the determinant of A is calculated. $det(A)$ has the form

$$det(A) = \sum_{\tau \in S_n} sgn(\tau) \cdot [\text{product of } n \text{ elements of } A]$$

Each element of A is at most 2^b, so the product of n such elements is at most 2^{nb}, and the summation over $n!$ such products is at most $n! \cdot 2^{bn}$. (When using Leibniz formula or Laplace's formula to compute the determinant of a $x \times n$ matrix, the number of required operations is of order $O(n!)$.) So

$$\{\# \text{ of bits to represent } det(A)\} \leq \log(det(A)) \leq O(n \log n + bn)$$

So solution x is describable with $O(n \log n + bn)$ bits. ☐

Now we consider a more general case when $n \neq m$ or $det(A) = 0$. We define $col(A)$ as the set of columns of A and $span(col(A))$ as the vector space spanned by $col(A)$, that is,

$$span(col(A)) = \{x | x \in R^m, \exists \alpha_1, ..., \alpha_n, x = \sum_{i=1}^{n} \alpha_i A_i\}$$

where A_i is the i-th column of A.
Then we select a maximal set of columns of A that are linear independent, denoted by S. Any other column in A should be in span(S). We view $Ax = b$ as

$$[A_1\ A_2\ ...\ A_n]\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = b$$

Then Ax is just a linear combination of the columns of A, so

$$Ax = b \text{ has solution(s)} \iff b \in span(col(A)) \iff b \in span(S)$$

Let $\bar{S}$ = completion of S to a basis in R^m (we know $|S| \leq m$), then we solve the equations

$$[S_1\ S_2\ ...\ S_{|S|}\ S_{|S|+1}\ ...\ S_m]\begin{bmatrix} x' \\ y \end{bmatrix} = b$$

where $x' \in R^{|S|}$, $y \in R^{m-|S|}$.
The set of the first $|S|$ columns $\{S_1, S_2, ..., S_{|S|}\}$ is equivalent to S, or in other words, they are columns of A. The remaining $m - |S|$ columns $S_{|S|+1}, ..., S_m$ are from $\bar{S}$ and added to make the basis of R^m.
If $Ax = b$ has solution(s), then we expect to see $y = 0$ because b is in $span(S)$.

2 Proof of No Solution

What if the system of linear equations has no solution? It is easy to show the system has a solution by providing a witness. Similarly, we want to find a "nice" witness to no solution. The following claim tells us that proving no solution is equivalent to solving another linear system of equations, which gives a sense of duality.

Claim 2. *$Ax = b$ has no solution $\iff \exists y \in R^m$, s.t. $y^T A = 0$ and $y^T b \neq 0$.*

Proof. We first prove $\Longleftarrow$ part by contradiction.
Suppose there exists some y such that $y^T A = 0$ and $y^T b \neq 0$ and $Ax = b$ has a solution. Then for any x,

$$Ax = b \implies y^T Ax = y^T b$$

but we also have

$$y^T Ax = (y^T A)x = 0x = 0$$
$$y^T b \neq 0$$

that is a contradiction to $Ax = b$.
Then we prove $\implies$ part. Since $Ax = b$ has no solution, we know $b \notin span(col(A))$. Let $proj_A b$ be the projection of vector b on space $span(col(A))$. We construct $y = b - proj_A b$. Then we have $y \perp span(col(A))$, so $y^T A = 0$.

On the other hand, $y^T b = y^T(y + proj_A b) = ||y||^2 \neq 0$ (which can be rescaled so $||y||^2 = 1$), so we find a valid y. $\square$

Such a y can be found by solving the following system of equations.

$$\begin{bmatrix} A^T \\ b^T \end{bmatrix} y = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in R^{n+1}$$

3 Basics on Linear Programming

Standard formulation of linear programming:

$$\min c^T x \quad s.t. \quad Ax = b \text{ and } x \geq 0$$

Definition 3. *An inequality is tight iff it takes "$=$".*

Definition 4. *The feasible set $F = \{x | Ax = b, x \geq 0\}$*

Definition 5. *$x \in F$ is a basic feasible solution iff it is not a convex combination of points in F. More formally, a point p is a basic feasible solution iff*

$$\exists y^1, ..., y^{n+1} \in F, \alpha_1, ..., \alpha_{n+1} \in R, \ s.t. \ p = \sum_{i=1}^{n+1} \alpha_i y^i \ \text{and} \ \sum_{i=1}^{n+1} \alpha_i = 1 \ \text{and} \ y^i \neq p, \forall i \in [n+1]$$

An example is shown in Figure 1.

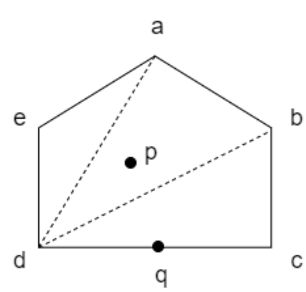

Figure 1: p is a convex combination of a,b, and d. q is a convex combination of c and d. e is not a combination of any other points, so it can be a basic feasible solution (if it is indeed a solution).

Claim 6. *A basic feasible solution is a vertex of polytope F.*

Claim 7. *If the Linear Programming problem is feasible and bounded (i.e., not infinity), then there exists an optimal solution which is a basic feasible solution.*

Proof. Take x^* that is an optimal solution to LP. If x^* is not a basic feasible solution, then x^* is not a vertex. In particular, we know that the number of linearly independent equations and tight inequalities is at most $n-1$. Let C be the subspace spanned by these linearly independent equations and tight inequalities. There is at least 1 dimension of freedom.

$$\implies \exists \bar{d} \neq 0, \forall \alpha, x^* + \alpha \bar{d} \in C$$

Further suppose $\exists \epsilon \in R - \{0\}, x = x^* + \epsilon \bar{d} \in F$. We rewrite our optimization objective

$$c^T x = c^T(x^* + \epsilon \bar{d}) = c^T x^* + \epsilon c^T \bar{d}$$

Since x^* is the optimal solution, $c^T d$ must be zero. So the $c^T x$ remains unchanged if we change x on the direction of $\bar{d}$. We can push x such that one more inequality in $x >= 0$ becomes tight. This procedure is repeated until we reach an optimal solution x^*, where there are n linearly independent equations and tight inequalities. $\square$

4 Linear Programming Algorithms

We introduce our first algorithm for Linear Programming. Each time we consider n linear independent equations and tight inequalities. Then we solve for x and compute $c^T x$. Finally, we choose x that is feasible and minimizes $c^T x$.
Time complexity = # of choices of linearly independent equations/tight inequalities

$$= \binom{n+m}{n} \times poly(n)$$

where n and m are the number of equality and inequality constraints. This is not a polynomial time algorithm. To Be Continued.