COMS 4995-3: Advanced Algorithms Mar 29, 2017

Lecture 18 – Linear Programming, Duality

Instructor: *Alex Andoni* Scribes: *Peilin Zhong, Ruiqi Zhong*

1 Last Class

In the last class, we began to introduce linear programming. We now discuss about Linear system of equalities.

1.1 Linear system of equalities

Given $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ if A is a square matrix i.e., $m = n$ and $det(A) \neq 0 \Leftrightarrow$ We can use Gaussian elimination to find a unique $x^* \in \mathbb{R}^n$ such that $Ax^* = b$.

2 Equivalency of "No Solution" and "Exists Solution"

In this scribe, all the inequalities between vectors mean pairwise inequalities between entries.

Definition 1. $Span(col(A)) = \{\sum_{i=1}^{n} \alpha_i A_i, \alpha_i \in \mathbb{R}, A_i \textit{ is the } i^{th} \textit{ column of } A\}$

Now we extract the maximum number of linearly independent columns of A, and let S be the set of indexes of these column vectors in A. Then we have $span(col(A)) = span(col(A_S))$. Suppose that $\exists x\ s.t. Ax = b$, then $b \in span(col(A)) = span(col(A_S))$. Let $C = \begin{bmatrix} A_S & B \end{bmatrix}$ where B is a set of $m - |S|$ linearly independent vectors outside $span(col(A))$. Now we solve for $C \begin{bmatrix} x_S \\ y \end{bmatrix} = b$, and we must find the solution to be $x_i = \begin{cases} x_{S_i} & i \in S \\ 0 & otherwise \end{cases}$

2.1 What if there is no solution?

Claim 2. *There is no solution* $\Leftrightarrow \exists y\ s.t.\ y^\top A = 0, y^\top b \neq 0$

Proof. "$\Leftarrow$" direction:
By contradiction, if $\exists x$ s.t. $Ax = b$, then $0 \neq y^\top b = y^\top Ax = 0^\top x = 0$
Intuition: $y^\top$ is a linear combination of $col(A)$ is 0, but b disagrees. □

Proof. "$\Rightarrow$" direction: there is such a y being a "certificate of no solution"
no solution $\Rightarrow b \notin span(col(A))$
Let $proj_A(b)$ be a projection of b on $span(col(A))$, let $y = b - proj_A(b)$. Then we have that $b^\top y \neq 0, A^\top y = 0$. Also we $\exists y$ s.t. $b^\top y = 1$ (by normalization) □

2.2 How to find y quickly?

Solve the system: $\begin{bmatrix} A^\top \\ b^\top \end{bmatrix} y = \begin{bmatrix} 0_n \\ 1 \end{bmatrix}$

Note also that Gaussian elimination can also give a certificate of "no solution".

3 Back to the Linear Programming

We have a standard form of linear programming:

$$\begin{aligned} & \min c^\top x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

Let $P = \{x \mid Ax = b, x \geq 0\}$. If $P \neq \emptyset$ and P is bounded, then we call P as a polytope.

3.1 Basic Feasible Solution and Vertex

Definition 3. *An inequality (equality) constraint is tight if the equality holds. Point $x \in \mathbb{R}^n$ is basic if it is a solution to n linearly independent tight constraints. A basic feasible solution (bfs, for short) x is both feasible and basic.*

Definition 4. *A point $x \in P$ is a vertex if and only if it is not a convex combination of other points in P, namely:*

$$\nexists y^1, \cdots, y^{n+1} \in P \textit{ and } \alpha_1, \cdots, \alpha_{n+1} \geq 0 \textit{ s.t. } x = \sum_{i=1}^{n+1} \alpha_i y^i, \sum_{i=1}^{n+1} \alpha_i = 1, \forall i \in [n+1], y^i \neq x$$

Claim 5. *A bfs is equivalent to a vertex.*

Claim 6. *If an LP is feasible and bounded $\Rightarrow$ A bfs is an optimal solution.*

Proof. Consider a solution x^* that is not basic but optimal. $\Rightarrow$ it satisfies at most $n-1$ linearly independent tight constraints. (Let's name the set of the such tight constraints T.) Tight constraints T defines a linear subspace $\Rightarrow$ contains a line, let $\vec{d}$ be the direction of the line.

$$\exists \epsilon > 0 \text{ s.t. both } x^* \pm \epsilon \vec{d}$$

are feasible. Since x^* is optimal, $x^* \pm \epsilon \vec{d}$ are feasible $\Rightarrow \vec{d}^\top c = 0$ (Otherwise one of $c^\top(x \pm \epsilon \vec{d}) > c^\top x^*$) $\Rightarrow$ We can change x^* in the direction of $\vec{d}$ s.t. one of its coordinate decreases. How much can we decrease? Until something else becomes tight, which is a coordinate becoming 0. In particular, $x_i = 0$. Then we have added one constraints, $x_i \geq 0$. Therefore, we can repeatedly add in linearly independent constraints, until the point is basic. □

A naive algorithm Now we have a first algorithm for Linear Programming:

1. We can brutally try all bfs $\Rightarrow$ iterate through all the $\binom{n+m}{n}$ subsets of constraints. We let the set of constraints be T.

2. Solve as if constrains in T are tight.
3. Check whether the solution is feasible.
4. Choose the optimal bfs.

3.2 Duality

- It is easy to show that optimal solution $\leq v^*$ (just give the right $x^* \in P$ & $c^\top x^\top \leq v^*$)
- How to show the optimal solution $\geq v^*$?

Suppose v^* is the optimal solution, and suppose (just suppose for now) we can find a $y \in \mathbb{R}^m, y^\top A = c^\top \Rightarrow y^\top Ax = c^\top x = y^\top b \Rightarrow y^\top b = c^\top x \ \forall x$. However, this is too good to be true. Let's be less ambitious:

If we can find a $y^\top A \leq c^\top$, then we have $y^\top Ax \leq y^\top b$ (since $x \geq 0 \Rightarrow (y^\top A)x \leq c^\top x \Rightarrow y^\top b \leq c^\top x$)

How to find a best lower bound based on $b^\top y$ (the same as $y^\top b$.) Now we have another Linear Programming problem, namely: we have an unknown $y \in \mathbb{R}^m$, and we want to maximize $c^\top y$ given the constraint $y^\top A \leq c^\top$. This linear programming problem we name it **Dual Program** and the original problem we name is **Primal Program**.

4 Next Class...

Let w^* = optimal solution of Dual Program = $\max_{Ay \leq c} b^\top y$. We have already proved the weak duality: $w^* \leq v^*$. Next class we will prove strong duality, which is $w^* = v^*$ if both solutions are feasible. Also, we will demonstrate the fact that Dual(Dual) = Primal.