Led 3 (wed, 25 Jan): Lemke-Howson Algorithm (1964)

Wednesday, January 25, 2017 11:24 AM

*Recap :

NE in a Game $\Rightarrow$ SNE in symm. Game

Symmetric Game : (players are indistinguishable)

$S_1 = S_2 = S \Rightarrow \Delta_1 = \Delta_2 = \Delta$; $B = A^T \Rightarrow A = B^T$

Symmetric NE : NE $(\bar{x}, \bar{y})$ s.t. $\bar{x} = \bar{y}$

$\Rightarrow$ NE $(\bar{x}, \bar{x})$ iff

$$\boxed{\forall i \in S: \quad x_i > 0 \Rightarrow (Ax)_i = \max_k (Ax)_k}$$

$\rightarrow$ Due to scale invariance without loss of generality we can assume that $A > 0$. (See lec 2 notes).

★ Linear Complementarity Problem (LCP) Formulation.

$\rightarrow$ Define : $V_x = \max_k (Ax)_k$

- x is SNE iff

$\forall i : \quad x_i > 0 \quad (Ax)_i \leq V_x \quad ; \quad \sum x_i = 1$ (*)

$\therefore$ x is SNE iff

$$\left.\begin{array}{l} \forall i \in S: \; x_i \geq 0, \quad (Ax)_i \leq V_x \;\; \& \;\; \sum_i x_i = 1 \\ \forall i \in S: \; x_i = 0 \text{ or } (Ax)_i = V_x \end{array}\right\} (*)$$

$\rightarrow$ Since $A > 0$, we have $V_x > 0$ at any $x \in \Delta$.

Define: $z_i = \dfrac{x_i}{V_x}$ then, equivalent system is

$$\left.\begin{array}{l} \forall i \in S: \; z_i \geq 0, \quad (Az)_i \leq 1 \\ \forall i \in S: \; z_i = 0 \text{ or } (Az)_i = 1 \end{array}\right\} \rightarrow (\#)$$

Claim 0: If $z \neq \bar{0}$ is a solⁿ of (#) then x s.t. $x_i = \dfrac{z_i}{\sum_i z_i}$, $\forall i \in S$ & $V_x = \dfrac{1}{\sum_i z_i}$ is a solⁿ of $(*)$.

Proof: Clearly $\bar{x}$ is well defined because $\sum_i z_i > 0$. The rest follows by definition of $(*)$ & (#).

Due to the above claim, finding SNE of game (A, A^T) reduces to finding Sol^n of (#) that is non zero.

Goal: Find $\bar{Z} \neq \bar{0}$ that satisfies (#).

☆ Lemke-Howson (LH) Algorithm.

(Polytope) $P : \{ z \mid z_i \geq 0 \,,\ (Az)_i \leq 1 \quad \forall i \}$

Complementary conditions: $z_i = 0$ or $(Az)_i = 1 \quad \forall i$

Call $z_i \geq 0$ by name $i_n \to$ (non-negative) &

$(Az)_i \leq 1$ by " $i_p \to$ (payoff)

☆ Definitions:

→ We say an "in-equality is tight" at a point if it holds with equality there

i.e. at $\bar{z} = 0$, inequality $z_1 \geq 0$ is tight

but " $(Az)_1 \leq 1$ is not.

→ $T_z = \{$ set of tight inequalities at $z\}$ for $z \in P$. i.e. for $\bar{z}=0$ $T_z = \{i_n \mid i \in S\}$.

→ We say "label $i \in S$ is present" at any point $z \in P$ if either $i_p \in T_z$ or $i_n \in T_z$.

→ Fully - Labeled (FL): $z \in P$ is called fully-labeled if $\forall i \in S$, label i is present at z.

→ 1-Almost Full Labeled (FL_1): $\forall z \in P$ where all but label 1 have to be present, i.e. $\forall i \in S, i \neq 1$, label i is present at z.

Next 3 claims follows from definitions.

Claim 1: $FL \subset FL_1 \subset P$

$\therefore$ $z \in FL \iff z$ satisfies #

Claim 2: $z \in FL \iff z$ satisfies (#)

Claim 3: $z \in \overline{O}$

From Claim 2, we can rewrite our goal as.

Goal: Find a non-zero fully labeled point.
I.e., find $z \in FL$, $z \neq 0$.

Lemke-Howson algorithm starts at $z = 0$ and follows a path of 1-almost fully labeled points. To understand why such a path exists we need to know some facts about polytopes first.

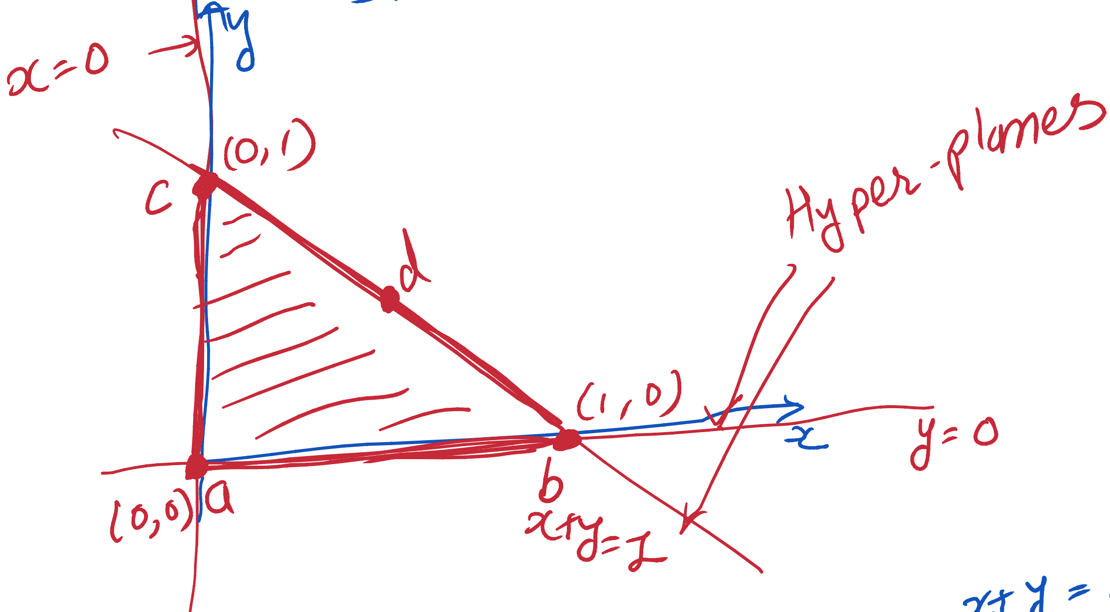
☆ Some Properties of Polytopes:

A set of linear inequalities defines a polytope (or called polyhedra if unbounded). System

$$Ax \leq b \quad \text{where } A \text{ is } m \times n$$

$Ax \le b$ where A is $m \times n$ defines polytope in n-dimensions with m inequalities. For example

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ defines}$$



$T_a = \{ x=0 , y=0 \}$ $\quad$ $T_c = \{ x=0, x+y=1 \}$

$T_b = \{ y=0, x+y=1 \}$ $\quad$ $T_d = \{ x+y=1 \}$

Assume Non-degeneracy: Any set of at most n inequalities are linearly independent.

Lemma 1: If, x, s.t. $Ax \le b$ then

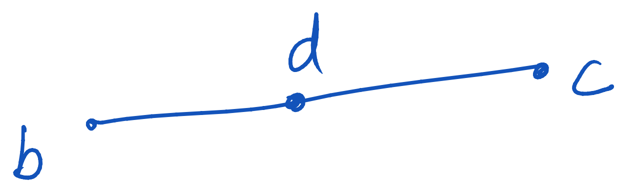
Lemma 1: If x, s.t. $Ax \leq b$ ...

⓪ $|T_x| \leq n$.

① $|T_x| = n$ then x is a vertex.

② $|T_x| = n-1$ then x is on an edge.

Above, we have

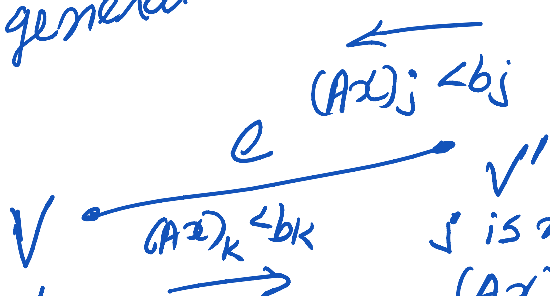


$|T_b| = |T_c| = 2$

$|T_d| = 1$

$T_d = T_b \cap T_c$

☆ In general,



k is tight

$(Ax)_k = b_k$

$(Ax)_j < b_j$

j is new tight

$(Ax)_j = b_j$

$(Ax)_k < b_k$

Lemma 2: Let vertices v & v' are adjacent and connected by edge e. If v' can be reached from v by relaxing inequality k, namely $(Ax)_k = b_k$, and at the new tight inequality is j namely inequality k, [illegible]

v' the new tight inequality is j, namely $(Ax)_j = b_j$ then

① $T_{v'} = \{T_v \setminus \{k\}\} \cup \{j\}$

② for any $d \in e$, $d \neq v, v'$

$$T_d = T_v \setminus \{k\}$$
$$= T_{v'} \setminus \{j\}$$
$$= T_v \cap T_{v'}$$

③ Relaxing $(Ax)_j = b_j$ at v' leads back to v.

✩ Back to the LH Algorithm.

Note that in polytope P is defined on n variables & $2n$ constraints. Therefore it is in n-dimensions.

$\therefore$ n inequalities are tight at it's vertices (Lemma 1).

→ Duplicate Label: We say that label $i \in S$ is duplicate at $z \in P$ if $i_p \in z$ & $i_n \in z$.

Lemma 3: If $z \in FL$ then z is a vertex of polytope P and has no duplicate label.

Proof: $z \in FL \Rightarrow i_p \in T_z$ or $i \notin T_z$, $\forall i \in S$

$\therefore |S| = n \Rightarrow |T_z| \geq n$

If there is a duplicate label at z then $|T_z| > n$, which contradicts Lemma 1.

$\therefore$ no duplicate label at z

$\Rightarrow |T_z| = n \Rightarrow z$ is a vertex (Lemma 1).

Next, we will try to understand structure of FL_1.

Lemma 5: Let $z \in FL_1$ then

① z is either a vertex or on an

① z is either a vertex or on an edge

② If z is a vertex and is not fully-labeled ($z \notin FL$) then there is exactly one duplicate label at z.

Number of edges adjacent to z in FL_1 is two.

③ If $z \in FL$, then it has one adjacent edge in FL_1.

Proof:

Part ①: By definition of FL_1,

$\forall i \in S, i \neq 1$ either $i_p \in T_z$ or $i_n \in T_z$

$\Rightarrow |T_z| \geq n-1 \Rightarrow z$ on a vertex or on an edge of $\mathcal{P}$

($\because$ Lemma 1)

Part ②: vertex $z \in FL_1$ & $z \notin FL$.

To the contrary, if no $k \in S$, $k \neq 1$ is duplicate

then $\forall k \in S, k \neq 1$ exactly one of

i & i in T_z, and

then $\forall k \in S, k \neq 1$ ...

i_p & i_n in T_z, and

none of 1_p & 1_n in T_z

$\Rightarrow |T_z| = n-1 \Rightarrow$ Contradiction to z being a vertex.

Suppose $k \neq 1$ is duplicate

If more than 1 duplicate label then by similar argument we have

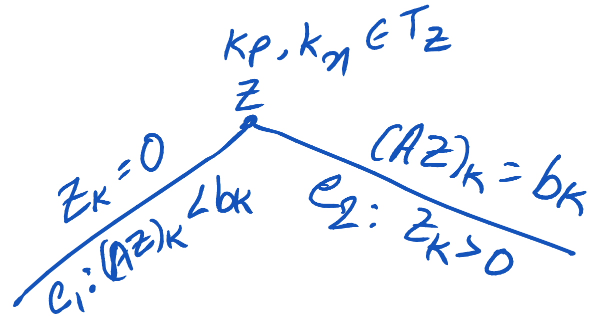
$|T_z| > n \Rightarrow$ contradiction to Lemma 1.

$\therefore$ Exactly one duplicate label at z.

Let it be $k \in S$, $k \neq 1$.

Then $k_p \in T_z \Rightarrow (Az)_k = b_k$

$k_n \in T_z \Rightarrow z_k = 0$



Suppose, Relaxing $(Az)_k = b_k$ gives edge e_1 &

... e_2.

Suppose, Relaxing $(Az)_k = b_k$ gives edge e_1.
relaxing $z_k = 0$ " " e_2.

Then on e_1, $z_k = 0$ still holds ($\because$ Lemma 2)
on e_2, $(Az)_k = b_k$ " "

$\therefore$ $k_n \in T_{e_1}$, $k_p \in T_{e_2}$

$\Rightarrow e_1, e_2 \in FL_1$.

For any other $i \neq k$ $i \neq 1$ we have exactly one of
① $i_p \in T_z \Rightarrow (Az)_i = b_i$
② $i_n \in T_z \Rightarrow z_i = 0$
Say ① is the case, then
e: edge obtained by relaxing $(Az)_i = b_i$ at z

then label i is not present on $e \Rightarrow e \notin FL_1$

This proves that exactly two edges of FL_1 are adjacent to vertex $z \in FL_1 \setminus FL$.

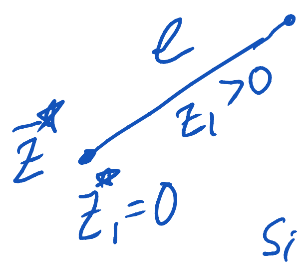
Part ③: Proof is similar to that of part ②.
$z^* \in FL \Rightarrow$ no duplicate label at z^* ($\because$ Lemma 3)
... $i_n \in T_{z^*}$

$z^* \in FL \Rightarrow$ no duplicate label ...

$\Rightarrow$ exactly one of i_p or $i_n \in T_{z^*}$

$\Rightarrow$ exactly one of $(Az^*)_i = 1$ or $z^*_i = 0$ hold.

$$T_e = T_z \setminus \{1_n\}$$



$\Rightarrow e \in FL_1$

Similarly, if $(Az^*)_1 = 1$ then relaxing it will give an edge in FL_1

Relaxing any other equality from T_{z^*} will loose a label from $\{2, \ldots, n\} \Rightarrow$ the corresponding edge can not be in FL_1. ∎

Lemma 4 & 5 implies that

Theorem 1: FL_1 consists of paths & cycles on 1-skeleton (edges & vertices) of polytope $\mathcal{P}$. Endpoints of the paths are exactly the points in FL.

Proof: Part 1 of Lemma 5 $\Rightarrow$

$FL_1 \subset$ 1-skeleton of $\mathcal{P}$.

$FL_1 \subset$ 1-skeleton of $\mathcal{J}$.

Part ② Lemma 5 $\Rightarrow$ vertex $v \in FL_1 \setminus FL$

$\Rightarrow$ # edges of FL_1 incident on v is two

$\Rightarrow$ deg(v) in $FL_1 = 2$.

Part ③ Lemma 5 $\Rightarrow$ Vertex $v \in FL$

$\Rightarrow$ # edges of FL_1 incident on v is one.

$\Rightarrow$ deg(v) in $FL_1 = 1$

Thus, FL_1 is a graph where degree of vertices is 1 or 2

$\Rightarrow$ FL_1 is a set of paths & cycles.

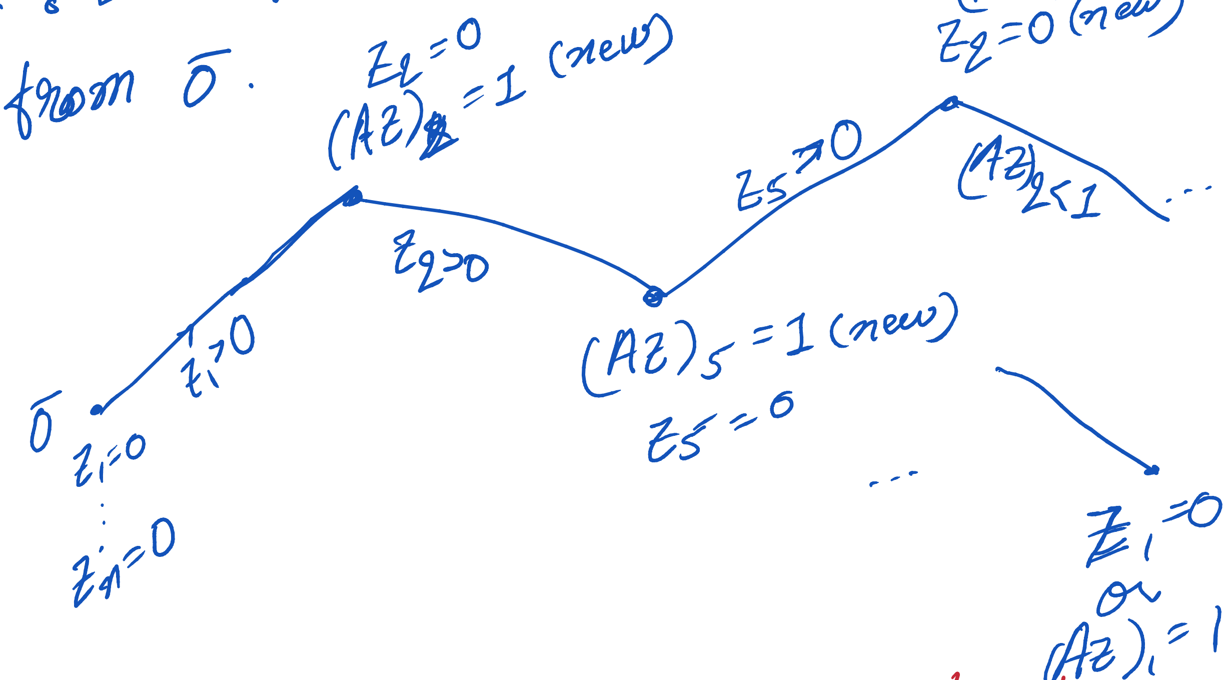
Part ③ Lemma 5 $\Rightarrow$ vertex $v \in FL$ iff deg(v) in FL_1 is 1

$\Rightarrow$ FL = end-points of paths of FL_1.

Theorem 1 implies that $\bar{0} \in FL$ is end-point of one of the paths in FL_1 which has to ... point in $FL \Rightarrow$ At a of one of the paths ...

end at a non-zero point in FL $\Rightarrow$ At a sol^n of our problem.

$\therefore$ LH Algorithm traverses this path starting from $\bar{0}$.



Thumb Rule: Leave duplicate label

Complimentary Pivot Algorithm

(0) Initialize: $\bar{z}=0$

(1) $z' \leftarrow$ relax $z_1=0$

(2) $z \leftarrow z'$; $k \leftarrow$ label of new tight inequality

(3) If $k=1$ then O/P $\bar{z}$. STOP.

(4) If $(Az)_k=1$ is the new tight ineq. then

(3) If ...

(4) If $(AZ)_K = 1$ is the new tight ...

$z' \leftarrow$ relax $z_K = 0$

Else $z' \leftarrow$ relax $(AZ)_K = 1$

Go To (2)

Lemke-Howson Algorithm

Theorem 2: LH Algorithm terminates with a solution of (#)

Proof: Clearly Algorithm starts and ends at a fully-labeled vertex & in between maintains all labels except 1 $\Rightarrow$ It follows the path in FL1 starting at $\bar{0}$. ($\because$ Theorem 1)

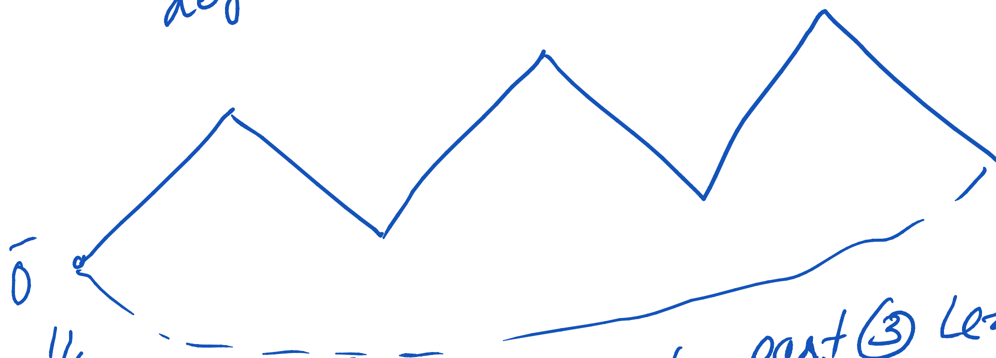
Since $A > 0$, P is bounded

$\Rightarrow$ The path has to terminate.

$\Rightarrow$ The algorithm has to terminate

$\Downarrow$

$\deg(v) = 3 \Rightarrow$ contradiction to part ② Lemma 5.



$\Downarrow$

$\deg(\bar{0}) = 2 \Rightarrow$ contradiction to part ③ Lemma 5.

$\Rightarrow$ Path ends at a vertex $v \neq \bar{0}$, $v \in FL$

($\because$ Theorem 1)

$\Rightarrow$ v is a solⁿ of (#).

Lemma 6: A non-degenerate symmetric Game (A, A^T) has odd number of SNE.

pf: Claim 0 $\Rightarrow$ SNE of (A, A^T) = solⁿ (#)

Claim 2 $\Rightarrow$ solⁿ (#) $= FL \setminus \{0\}$

Theorem 1 $\Rightarrow$ FL = End points of paths in FL_I

End points of paths in FL_I is odd

$\Rightarrow |FL \setminus \{0\}|$ is odd.

$\Rightarrow |FL \setminus \{0\}|$ is odd.

☆ Complexity class PPAD

Polynomial Parity Argument for Directed Graphs.

☆ Definition: Given $G = (V, E)$ where $V = \{0, \ldots, 2^n - 1\}$. Let bit(v) be bit represent of $v \in V$. Let S & P be Boolean circuit & directed edge $(u,v) \in E$ iff $S(u) = v$ & $P(v) = u$. Suppose $P(S(\bar{0})) = \bar{0}$, but $S(P(\bar{0})) \neq \bar{0}$.

Then every vertex in this graph G has both in-degree & out-degree at most 1. And $\bar{0}$ has in-degree zero & out-degree 1

$\Rightarrow G$ consists of directed paths & cycles and one of the path starts at $\bar{0}$.

end-point

and one ..

Goal: Find a non-zero end-point of any path.

It always exists -- just traverse the path starting at $\bar{0}$.

But this could be of exponential length.

Lemma 6: PPAD $\subset$ NP (TFNP)

Pf: Given $v \in P$ it is easy to check

if ① $v \neq 0$

② $S(P(v)) \neq v$ or $P(S(v)) \neq v$.

$\Downarrow$ in-deg$(v) = 0$ $\Downarrow$ out-deg$(v) = 0$

v is a sol^n. ∎