Lemke-Howson Algorithm

- let $m = |A_1|$, $n = |A_2|$, WOLOG $A_1 = \{1, \dots, m\}$, $A_2 = \{m+1 \dots m+n\}$
- let A be matrix of payoffs for player 1, B for player 2
- player 1 picks rows ⟺ mixed strat $x$ is a nonnegative row vector with $m$ entries adding to 1
- similarly player 2's mixed ~~strat~~ is a $n$-element column vec

$$u_1(x,y) = x^T A y \qquad u_2(x,y) = x^T B y$$

---

- a "polyhedron" is a region defined by linear inequalities

$$P_1 = \{x \in \mathbb{R}^M \mid \forall i \in M \ \ x_i \geq 0 \ \& \ \forall j \in N \ \ x^T \cdot \underset{\text{}}{\overset{j^{th}\text{ column}}{B_j}} \leq 1\}$$

$$P_2 = \{y \in \mathbb{R}^N \mid \forall j \in N \ \ y_j \geq 0 \ \& \ \forall i \in M \ \ \underset{i^{th}\text{ row}}{A_i} \, y \leq 1\}$$

Assumption 1. A, B nonnegative, A has no zero columns, B has no zero rows.

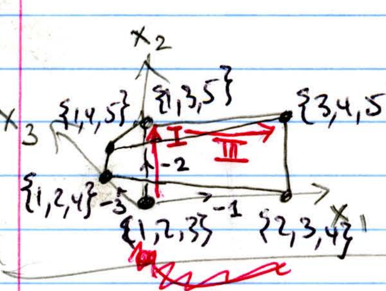
- to satisfy, add constant to all entries of A / B; NE unchanged
- e.g. $A = \begin{matrix} -1 & 1 & -2 \\ -2 & -2 & 0 \\ 0 & -1 & -1 \end{matrix} \xrightarrow{\text{add } 2} A = \begin{matrix} 1 & 3 & 0 \\ 0 & 0 & 2 \\ 2 & 1 & 1 \end{matrix}$

- using As. 1, $P_1$ is $m$-dimensional, $P_2$ is $n$-dimensional, $P_1$ & $P_2$ bounded; called "polytopes".

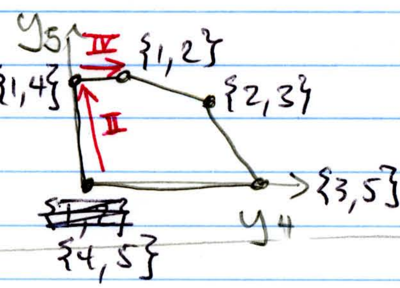
---

- e.g. $A = \begin{matrix} 0 & 6 \\ 2 & 5 \\ 3 & 3 \end{matrix} \qquad B = \begin{matrix} 1 & 0 \\ 0 & 2 \\ 4 & 3 \end{matrix}$

$P_1$




$P_2$




• = vertex of polytope

maxminimization no good for nonzerosum games (L2P):
(roughly) maxmin: indiff b/w opp's strats
NE: indiff b/w own strats

(c.f. 364.2 of some homework?)

Lemke-Howson Algorithm.

Q. - is there any way we can use a linear programming approach for non-zero-sum, two-player games, to find an NE

Hint: - recall Support Char., in two-player case:

$(p^*, q^*)$ is a NE $\Leftrightarrow$ $\text{supp}\, p \subseteq B_1(q)$
& $\text{supp}\, q \subseteq B_2(p)$.

- idea: guess supp p, then "$\text{supp}\, p \subseteq B_1(q)$" becomes an LP
similarly for q.

- Algorithm to find all NEs:

For all $S \subseteq A_1$, $S \neq \emptyset$
for all $T \subseteq A_2$, $T \neq \emptyset$
let $\text{feas}_1 = \{ p \mid \exists x:$
$p_i > 0 \quad \forall p_i \in S$
LP! → $p_i = 0 \quad \forall p_i \in (A_1 \setminus S)$
$\sum_i p_i = 1$

x: best value for player 2 — linear in $p_i$'s, x:
$u_2(p, a_2) = x \quad \forall a_2 \in T$
$u_2(p, a_2) \leq x \quad \forall a_2 \in (A_2 \setminus T) \}$

let $\text{feas}_2 = \{ \dots \}$
output $\{ (p,q) \mid p \in \text{feas}_1, q \in \text{feas}_2 \}$.

(c.f. Dickhaut & Kaplan 1991)
- using ">" easy to get around.

- issue: nice enough algorithm but doesn't prove that it outputs anything

- today study "Lemke-Howson" algorithm for general 2-player games.
  - from 1964 paper
  - looks like simplex algorithm
  - can take exponentially many steps ☹ (Savani - von Stengel '04)
  → but unlike LPs no polytime algorithm known
  - PRO: constructive proof that every 2p strategic game has a (m)NE.
  - finds only one NE (not good for exhausting all)

## Labels

- $x \in P_1$ has label $i \in M$ if $x_i = 0$ } faces of $P_1$
- $x \in P_1$ has label $j \in N$ if $x^T B_j = 1$
- $y \in P_2$ has label $i \in M$ if $A^i y = 1$ } faces of $P_2$
- $y \in P_2$ has label $j \in N$ if $y_j = 0$

Note that $x, y$ may not add to 1, hence not exactly mixed strats; define

$$\text{nrml}(v) = v / (\textstyle\sum_i v_i).$$

so, e.g. if $x \overset{\in P_1}{\neq} 0$ then $\text{nrml}(x)$ is a mixed strat

Proposition. Let $x \in P_1$, $y \in P_2$, $x \neq 0$, $y \neq 0$. Then $x$ & $y$ together have all labels iff $(\text{nrml}(x), \text{nrml}(y))$ is a NE. All NE arise in this way. $\boxtimes$

Pf. Sketch/ Use Support Characterization; e.g.,

$x$ has label $i \Rightarrow x_i = 0$

$x$ doesn't have label $i \Rightarrow y$ has label $i$

$\Rightarrow i$ is a best response by player 1 to $\text{nrml}(y)$ $\boxtimes$

key step

A d-dimensional polytope is simple if exactly d faces meet at each vertex.

Assumption 2. $P_1$ & $P_2$ are simple.

Holds in our example; we'll explain general strategy later. (If As. 2 doesn't hold game is "degenerate".)

Proposition. If As. 2 holds,

- every vertex of $P_1$ (resp. $P_2$) gets exactly $m$ (resp. $n$) labels
- if $v$ has label set $I_+$, we can pick any $l \in I_+$ to delete from the label set, and there exists a new label $l'$ so that a vertex $v'$ is adjacent to $v$ with label set $I_+ \setminus \{l\} \cup \{l'\}$.

"pivoting"

continue example.

## Lemke-Howson algorithm
- starts at ($\vec{x}=\vec{0}$, $y=\vec{0}$) where all labels present
- $x$ and $y$ are always vertices of $P_1$ and $P_2$
- "pivots" in $x$ & $y$ until all labels present again.

## High-level description

$x=0$, $y=0$
pick any label $k_0$ of $x$; $k=k_0$
repeat
  pivot $x$ to new vertex in $P_1$ by removing $k$; $k'$ is added
  stop if $k'=k_0$
  pivot $y$ to new vertex in $P_2$ by removing $k'$; $k''$ is added
  stop if $k''=k_0$
  $k=k''$

output (nrml($x$), nrml($y$))

Why does it work? (sketch)
→ "configuration": pair $(x,y)$ with all labels but $k_0$
→ $(x,y)$ & $(x,y')$ "adjacent" if $y$ & $y'$ related by pivot
  $(x,y)$ & $(x',y)$ " " $x$ & $x'$ " "
→ hence we get a graph.

Configurations having all labels, same as NE $\cup \{(\vec{0},\vec{0})\}$, have degree 1. All (other) configurations have degree 2.

∴ Configuration graph has path & cycle components

Vertex $(\vec{0},\vec{0})$ has degree 1 ∴ endpoint of a path. L-H walks to other endpoint which is an NE. $\boxtimes$

Show walk in example. ($k_0=2$).

## Tableau Method.

$B^T x + s = 1$

$Ay + r = 1$

- add slack variables (vectors) $r, s$ to $P_1$ & $P_2$

$\Rightarrow$ two linear systems ~~$B^T x + r = 1$, $Ay + s = 1$~~.

- initial tableaux, solve for $r, s$ as basis

$$\underline{r = 1 - \cancel{B^T x}\ Ay} \qquad \underline{s = 1 - \cancel{Ay}}\ B^T x$$

$$\text{basic}\begin{cases} r_1 = 1 \phantom{- 2y_4} - 6y_5 \\ r_2 = 1 - 2y_4 - 5y_5 \\ r_3 = 1 - \underbrace{3y_4 - 3y_5}_{\text{nonbasic } (=0)} \end{cases} \qquad \text{basic}\begin{cases} s_4 = 1 - x_1 \phantom{- 2x_2} - 4x_3 \\ s_5 = 1 \phantom{- x_1} \underbrace{- 2x_2 - 3x_3}_{\text{nonbasic } (=0)} \end{cases} \quad (*)$$

- pivots are same as simplex pivots!
- "drop label" $\Leftrightarrow$ "add to basis" (backwards!)
- "label $k$" $\Leftrightarrow$ complementary pair of variables
  $\{x_i, r_i\} : i \in N$ $\qquad$ $\{y_j, s_j\} : j \in N$
- stop when in each complementary pair at least one is zero/nonbasic, i.e. when $x \cdot r = 0$ and $y \cdot s = 0$.

E.g./ we had $k_0 = 2$. corresponds to $x_2$ entering basis.
- min-ratio rule determines leaving variable, $s_5$
  (informally, if $x_2 \uparrow$, which basic variable hits zero first?)

$$s_4 = 1 - x_1 - 4x_3$$

$\therefore$ (*) changes to

$$x_2 = \frac{1}{2} \qquad - \frac{3}{2}x_3 - \frac{1}{2}s_5$$

- L-H's main loop is "after a variable leaves, its complement enters". e.g. $s_5$ left, now $y_5$ enters.

- min-ratio rule $\Rightarrow r_1$ leaves, get

$$y_5 = 1/6 - 1/6\, r_1$$
$$r_2 = 1/6 + 5/6\, r_1 - 2y_4$$
$$r_3 = 1/2 + 1/2\, r_1 - 3y_4$$

Since $r_1$ left, $x_1$ enters.
Min-ratio rule $\Rightarrow$ $s_4$ leaves, get

$$x_1 = 1 - 4x_3 - s_4$$
$$x_2 = \tfrac{1}{2} - \tfrac{3}{2}x_3 - \tfrac{1}{2}s_5$$

now $y_4$ enters, min-ratio rule $\Rightarrow$ $r_2$ leaves
(note: same pair as initial).

$$y_4 = 1/12 + 5/12\, r_1 - 1/2\, r_2$$
$$y_5 = 1/6 - 1/6\, r_1$$
$$r_3 = 1/4 - 3/4\, r_1 + 3/2\, r_2$$

STOP! In every complementary pair, one variable is basic.
Set nonbasic to 0 $\Rightarrow$ $x_3 = s_4 = s_5 = 0$
basic are $x_1 = 1$, $x_2 = 1/2$ $y_4 = 1/12$ $y_5 = 1/6$ $r_3 = 1/4$.

covered 10/9

$x = (1, 1/2, 0)$ $\quad$ $y = (1/12, 1/6)$
$\text{nrml}(x) = (2/3, 1/3, 0)$ $\quad$ $\text{nrml}(y) = (1/3, 2/3)$

~~IF THERE IS TIME, VERIFY~~

Remark: $u_1(x,y) = 4 = \left(\sum_i y_i\right)^{-1}$, $u_2(x,y) = 2/3 = \left(\sum_i x_i\right)^{-1}$ } true in general - look at proof.

How did nondegeneracy help?
Proposition. If the game is nondegenerate, there is always a unique winner of the min-ratio rule. ☒

Although there is a "lexicographic method" under which a similar prop. holds for degenerate games,

- for general games, if you break ties "badly", cycling through the same bases is possible
  - → to fix: if you repeat a basis, break tie in different way
  - → we'll set up problems so cycling doesn't occur