Core limitation 2: non-existence

For various game settings, core may be empty.

What can be expected in such games?

- Stable sub coalitions may form
- Or a weaker notion of equilibrium may be proposed

Refinements of the core

ϵ-core: idea similar in spirit to ϵ-Nash equilibrium.

Defn: A payoff vector x is in the ϵ-core of a coalitional game (N, v) if

$$\sum_{i \in S} x_i \geq v(S) - \epsilon, \quad \forall S \subseteq N, S \neq \phi.$$

One interpretation/motivation: There is a cost to move from the grand coalition, which is denoted by ϵ. If the value is not decreasing ~~beyond~~ more than that cost, it is possibly not meaningful to deviate from the grand coalition.

Mathematically, no reason why $\epsilon \geq 0$. If $\epsilon < 0$, the condition of ϵ-core is giving some "bonus" for forming a coalition. The allocation not only ~~gives~~ is coalitionally rational, it awards something that is strictly better by a constant margin.

The ϵ-core allocation is more stable than a core allocation.

But for a given ϵ, the ϵ-core may still be empty. We can continue relaxing the ϵ-core until some non-empty ϵ-core is found.

Least core

Defn: A payoff vector x is in the least core of (N, v) if x is a solution to the following linear program

LP-1
$$\begin{aligned} &\min \ \epsilon \\ &\text{s.t.} \quad \sum_{i \in S} x_i \geq v(S) - \epsilon \qquad \forall S \subset N. \\ &\qquad\quad x(N) = v(N) \end{aligned}$$

- objective is non positive iff the core of the game is non empty
- For sufficiently large ϵ, the constraints can be always satisfied
- When the core is non-empty, least core does not contain all core allocations - rather gives the least opportunity for every coalition to deviate - is a core refinement.

Still is a set-solution.

The solution of the LP may return a ~~solutions~~ vector x and ϵ s.t. not all inequalities are tight.

Exercise: construct examples of multiple solutions of least core.

Strengthening the least core

Idea: make the slack inequalities tight

Formally, say ϵ_1 is the optimal value of the LP-1. Now we optimize
[Also let $\mathcal{S}_1$ be the set of coalitions for which the inequalities are tight]

$$\text{minimize} \quad \epsilon$$

$$\text{s.t.} \quad \sum_{i \in S} x_i = v(S) - \epsilon_1 \qquad \forall S \in \mathcal{S}_1$$

$$\sum_{i \in S} x_i \geq v(S) - \epsilon \qquad \forall S \in 2^N \setminus \mathcal{S}_1$$

$$x(N) = v(N)$$

This makes few more inequalities tight.
There could still be some slack inequalities.
We sequentially repeat this procedure until all inequalities become tight.
Since there are finite number of inequalities, this always converge to a unique payoff vector, known as "nucleolus."

Defn: An allocation x is in the nucleolus of a coalitional game (N, v) if it is the solution of a series of LPs

$$(LP_1) \qquad \min \epsilon$$

$$\text{s.t.} \quad \sum_{i \in S} x_i \geq v(S) - \epsilon \qquad \forall S \subseteq N,$$

$$x(N) = v(N)$$

$$(LP_2) \qquad \min \epsilon.$$

$$\text{s.t.} \quad \sum_{i \in S} x_i = v(S) - \epsilon_1 \qquad \forall S \in \mathcal{S}_1$$

$$\sum_{i \in S} x_i \geq v(S) - \epsilon \qquad \forall S \in 2^N \setminus \mathcal{S}_1$$

$$x(N) = v(N)$$

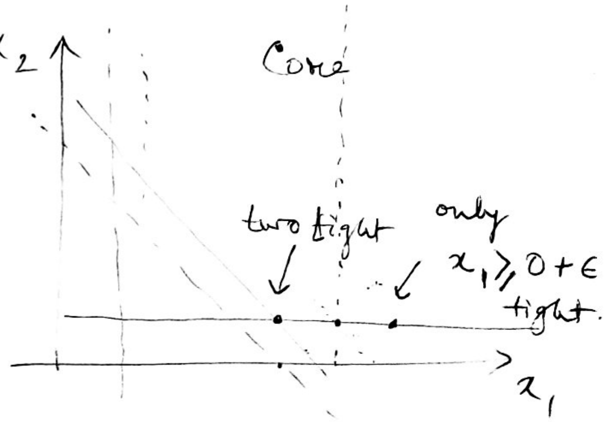
$$(LP_k) : \quad \min \epsilon$$
$$\text{s.t.} \quad \sum_{i \in S} x_i = v(S) - \epsilon_1 \quad \forall S \in \mathcal{S}_1$$
$$\vdots$$
$$\sum_{i \in S} x_i = v(S) - \epsilon_{k-1} \quad \forall S \in \mathcal{S}_{k-1} \setminus \mathcal{S}_{k-2}$$
$$\sum_{i \in S} x_i \geq v(S) - \epsilon \quad \forall S \in 2^N \setminus \mathcal{S}_{k-1}$$
$$x(N) = v(N)$$

- Needs at most n iterations (argue over the dimensions of the variables space. [Exercise]

Intuition: $v(1) = 0, \ v(2) = 1, \ v(1,2) = 3$

core is non-empty
$\epsilon < 0$, more negative is better

if $v(2) = 4$, core empty.



Theorem: For any game (N, v), nucleolus exists and is unique.

Proof: existence: The series of LPs can be solved and reach some assignment of x's s.t. all inequalities are met with equalities. In every round, at least one inequality will become tight, and it will converge (property of LP) and therefore a solution always exists.

Uniqueness: Earlier LPs influence the latter LPs only via the values. Therefore, the set of LPs will always lead to the same set of solutions $(\epsilon_1, \epsilon_2, \ldots)$ [In particular $\epsilon_1 > \epsilon_2 > \ldots$]. After all the iterations are over, we are left with 2^n equations over n variables. It has rank of ~~at most~~ n – if a solution exists, it must be unique.

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\
\begin{array}{c} S_1 \\ \\ \text{coalitions} \\ \\ S_{2^n} \end{array} & \left[\begin{array}{c} \\ \\ \text{Association} \\ \text{matrix} \\ \\ \end{array} \right] \\
 & \text{players}
\end{array}
x = \begin{bmatrix} v(S_1) - \epsilon_j \\ \vdots \\ \vdots \\ v(S_{2^n}) - \epsilon_k \end{bmatrix}
$$

columns are linearly indept.

we have made the RHS live in the space spanned by the columns of association matrix. Therefore, there exists a unique linear combination of those columns to yield the RHS [fact from linear algebra]. $\square$

An alternate definition

Nucleolus is also defined wrt excesses

Defn: The excess of a coalition S in (N, v) wrt payoff vector x is denoted as

$$e(S, x, v) = v(S) - \sum_{i \in S} x_i$$

Obs: if core is non-empty, then $\exists\, x$ s.t. all excesses are non-positive.

Given a coalitional game (N, v) and a payoff vector x compute all excesses except coalitions N and ϕ

This $2^n - 2$ dimensional vector is the "raw excess" vector.

The vector is sorted in decreasing order - sorted excess vector denoted $\theta(x, v)$

Given two payoff vectors x and y, we say excesses due to x are <u>lexicographically smaller</u> than those due to y, written $x \preceq_L y$ if for the smallest index i where $\theta(x,v)$ and $\theta(y,v)$ differ, $\theta_i(x,v) < \theta_i(y,v)$.

This is a valid binary relation, which is reflexive, transitive, complete, but not symmetric.

<u>Defn</u>: (Nucleolus, alternative defn)

Given a coalitional game (N, v), the nucleolus is the payoff vector x such that for all other payoff vectors y, $y \succeq_L x$, i.e. x lexicographically minimizes the excesses of all coalitions.

Compact representation of coalitional games

- Have seen several solution concepts – how to compute them in practice.
- The representation of the game is important in answering that question – straightforward representation will take enormous space.
  - gives a feeling that even bruteforce methods are also "good". Hence a compact representation is important.

Example: <u>Weighted Graph Games</u>

Game is defined by an undirected graph with edge weights and the value of a collection of nodes is the sum of the edge weights that run between the nodes.

Examples: Cities that are connected via high-speed toll highways – how to share the revenue among them.

<u>important</u>: easy to represent the values just $\binom{n}{2}$ numbers for the edge weights.

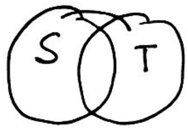
Defn: (WGG) Let (V, W) denote an undirected weighted graph, V = set of vertices, $W \in \mathbb{R}^{|V| \times |V|}$ is the set of edge weights, W symmetric, weight between i and j denoted by $w(i,j)$. The coalitional game WGG is a game (N, v) s.t.

1. $N = V$
2. $v(S) = \sum_{i,j \in S} w(i,j)$. $\forall S \subseteq N$.

Prop : If all weights are non-negative, WGG is convex.

$$v(S) + v(T) \overset{?}{\leq} v(S \cup T) + v(S \cap T)$$

$\downarrow$



edges from $S \setminus T$ and $T \setminus S$ are counted as extra.

Shapley value :

Thm: The SV of the game induced by a WGG (V, W) is

$$Sh_i(N, v) = \frac{1}{2} \sum_{j \neq i} w(i,j).$$

Proof: $$Sh_i(N, v) = \frac{1}{n!} \sum_{\pi \in \Pi(N)} w_i^{\pi}$$

$$w_i^{\pi} = v(P_i(\pi) \cup \{i\}) - v(P_i(\pi))$$

$$v(P_i(\pi)) = \sum_{j,k \in P_i(\pi)} w(j,k)$$

$$v(P_i(\pi) \cup \{i\}) = \sum_{j,k \in P_i(\pi)} w(j,k) + \sum_{k \in P_i(\pi)} w(i,k)$$

$$w_i^{\pi} = \sum_{k \in P_i(\pi)} w(i,k)$$

sum over all possible permutations, how many times should a specific k appear before i ? $\frac{n!}{2}$

$$\sum_{\pi \in \Pi(N)} w_i^{\pi} = \frac{n!}{2} \sum_{j \neq i} w(i,j)$$ $\square$

<u>Obs</u>: We can compute Shapley value in $O(n^2)$ time

Answering questions regarding the core of WGG is more complex.

<u>Cut</u>: is a set of edges that divide the nodes of a graph into two parts

$(S, V \setminus S)$ — edges between them form a cut.

Weight of a cut is the sum of its weights

$$\sum_{i \in S, j \in N \setminus S} w(i,j)$$

<u>Thm</u>: The Shapley value is in the core of a WGG iff there is no negative cut in the weighted graph.

<u>Proof</u>: From the previous result, if $\exists$ a cut $S, N \setminus S$ s.t. the weight is negative

then $$\sum_{i \in S} Sh_i = \frac{1}{2} \sum_{i \in S} \sum_{j \neq i} w(i,j)$$

$$= \frac{1}{2} \sum_{i \in S} \left[\sum_{j \neq i, j \in S} w(i,j) + \sum_{j \in N \setminus S} w(i,j) \right]$$

$$= \frac{1}{2} \cdot 2 \sum_{(i,j) \in S^2} w(i,j) + \frac{1}{2} \sum_{i \in S, j \in N \setminus S} w(i,j)$$

$$= v(S) + \frac{1}{2} \text{ weight of cut}$$

$$< v(S) \qquad \text{iff weight of cut is negative.}$$

<u>Thm</u>: The core is non-empty iff there is no negative cut in the weighted graph.

<u>Pf</u>: ($\Leftarrow$) is obvious from previous claim.

($\Rightarrow$) Suppose $\exists$ a negative cut, then we show core is empty. Say $(S, N\setminus S)$ has a negative cut

$$\left.\begin{aligned}\sum_{i\in S} Sh_i - v(S) &= \frac{1}{2}\sum_{i\in S, j\in N\setminus S} w(i,j) \quad \text{--- (1)}\\ \sum_{i\in N\setminus S} Sh_i - v(N\setminus S) &= \frac{1}{2}\sum_{i\in N\setminus S\, j\in S} w(i,j) \quad \text{--- (2)}\end{aligned}\right\}\text{same}$$

Pick any efficient payoff vector x.

$$v(N) = \sum_{i\in S} x_i + \sum_{i\in N\setminus S} x_i = \sum_{i\in S} Sh_i + \sum_{i\in N\setminus S} Sh_i \quad \text{--- (3)}$$

$$\left(\sum_{i\in S} x_i - v(S)\right) + \left(\sum_{i\in N\setminus S} x_i - v(N\setminus S)\right) = \sum_{i\in S, j\in N\setminus S} w(i,j) < 0$$

at least one of them is negative, hence core empty □

However, general WGG are harder to analyze.

<u>Thm</u>: Testing nonemptiness of core for general WGGs is NP-complete.

uses reduction from MAXCUT.