Cours 5 – Le hasard s'en mêle...

Motivation : jeux finis à somme nulle :

Considérons un jeu fini, $S_1 = \{1, \dots, m\}$ (ie J_1 dispose de m stratégies numérotées $1, \dots, m$), $S_2 = \{1, \dots, n\}$, de fonction de gain

$$g(i,j) = g_1(i,j) = -g_2(i,j)$$
$$= a_{ij} \in \mathbb{R}, \quad i = 1, \dots, m, \quad j = 1, \dots, n ;$$

Le tableau du jeu sous forme normale fait donc intervenir la matrice

$$A = (a_{ij})_{\substack{i=1,m \\ j=1,n}}, \quad m \times n :$$

J_1 \ J_2	$\cdots$	j	$\cdots$
$\vdots$	$\ddots$	$\vdots$	$\ddots$
i	$\cdots$	a_{ij}	$\cdots$
$\vdots$	$\ddots$	$\vdots$	$\ddots$

On sait d'après le cours précédent qu'on a un équilibre ssi :

$$\alpha = \max_{i=1,m} \min_{j=1,n} a_{ij} = \min_{j=1,n} \max_{i=1,m} a_{ij} = \beta$$

En général, $\alpha \leq \beta$, et on a deux cas :
(sachant qu'il existe toujours $\bar{i} \in \{1, \dots, m\}$ et $\bar{j} \in \{1, \dots, n\}$ tq :
$$\begin{cases} \alpha = \max_i \min_j a_{ij} = \min_j a_{\bar{i}j} \\ \beta = \min_j \max_i a_{ij} = \max_i a_{i\bar{j}} \end{cases})$$

i) $\alpha = \beta$: un équilibre existe, $\alpha = a_{\bar{i}\bar{j}} = \beta$, et aucune des deux joueuses n'a intérêt à jouer autrement que $\bar{i}$ pour J_1, $\bar{j}$ pour J_2, même en connaissant la stratégie de l'autre ! C'est l'exemple du dilemme des tradeuses du cours précédent.

ii) $\alpha < \beta$: c'est par exemple le cas pour le jeu suivant :

$$A = \begin{bmatrix} 5 & 6 & 7 \\ 4 & 15 & 8 \\ 12 & 3 & 10 \end{bmatrix}$$

On voit que $\bar{i} = 1$ (maximise le pire gain, c'est ici le seul), et que $\bar{j} = 3$ (minimise la pire perte, c'est aussi la seule). On a

$$\alpha = 5 < a_{\bar{i}\bar{j}} = a_{13} = 7 < \beta = 10$$

de sorte qu'en jouant $\bar{i}$ et $\bar{j}$, respectivement, J_1 et J_2 font toutes les deux mieux que ce qu'elles pouvaient espérer. Mais la tricherie devient désormais payante car, si J_1 sait par exemple que J_2 a joué $\bar{j} = 3$, elle a tout intérêt à jouer non pas $\bar{i} = 1$, mais $i = 3$ (gain de 10). Au contraire si J_2 anticipe que J_1 va jouer $i = 3$ (pour la raison précédente), elle a intérêt à jouer $j = 2$ (perte de 3)...

En général donc, dans le cas ii)
on a :

$$\alpha \le a_{\bar i \bar j} \le \beta$$

et l'une des deux inégalités au moins est stricte ; au moins l'une des deux joueuses fera strictement mieux que ce qu'elle espérait.

L'idée de Von Neumann, pour retrouver une situation d'équilibre consiste à jouer un grand nombre de parties en adoptant un pt de vue probabiliste :

- J_1 va jouer la stratégie $i=1$ avec la proba y_1, la stratégie $i=2$ avec la proba y_2...
- de même J_2 jouera la stratégie no. $j \in \{1, \dots, m\}$ avec la proba x_j.

On parle donc de "stratégie mixte", et on voit que cela revient à agrandir S_1 (resp. S_2) en $\Sigma_m := \{ y \in \mathbb{R}^m \mid y \geq 0, y_1 + \dots + y_m = 1 \}$

(resp. $\Sigma_n := \{ x \in \mathbb{R}^n \mid x \geq 0, x_1 + \dots + x_n = 1 \}$).

Choisir une stratégie mixte $y = (y_1, \dots, y_m) \in \Sigma_m$ revient donc à fixer avec quelle proba y_1 J_1 jouera $i=1$, etc. Les stratégies précédentes

dans Σ_1 (resp. Σ_2) où "stratégies pures" consistent donc à toujours jouer une stratégie $i \in \{1, \dots, m\}$ (resp. $j \in \{1, \dots, n\}$), ie à la jouer avec la proba $y_i = 1$: $y = (0, \dots, 0, 1, 0, \dots, 0)$ ($i^{\text{ème}}$)
(resp. $x_j = 1$, et $x = (0, \dots, 0, 1, 0, \dots, 0)$ ($j^{\text{ème}}$)).

Reste, pour mettre ce nouveau jeu (qui n'est plus fini ! cf. card Σ_m = card $\Sigma_n = \infty$!) sous forme normale à définir la fonction de gain (on reste à somme nulle), ce que l'on fait en considérant les gains moyens (cf. Exo poker, TD3, pour des considérations analogues), en espérance :

soient donc $(y, x) \in \Sigma_m \times \Sigma_n$, on pose

$$g(y,x) := E\left(g(s^1, s^2)\right)$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} g(i,j) \cdot \underbrace{P(s^1 = i) \cdot P(s^2 = j)}_{\text{indépendance } s^1 \text{ et } s^2}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} \cdot y_i \cdot x_j$$

$$= (A \cdot x \mid y)$$

stratégie mixte J_1 (pour y)

stratégie mixte J_2 (pour x)

où $(\cdot \mid \cdot)$ est le produit scalaire sur $\mathbb{R}^m$ ($(y \mid z) = {}^t y \cdot z = \sum_{i=1}^{m} y_i \cdot z_i$). Dans ce calcul d'espérance, on a considéré s^1 et s^2 (stratégies de J_1 et J_2, resp.) comme des v.a. (= variables aléatoires) discrètes à valeurs dans

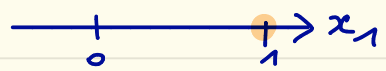
$S_1 = \{1,\dots,m\}$ et $S_2 = \{1,\dots,n\}$, resp., de densités respectives $y = (y_1,\dots,y_m)$ et $x = (x_1,\dots,x_n)$. (C'est des probas, on a bien $\sum_i y_i = \sum_j x_j = 1$.)

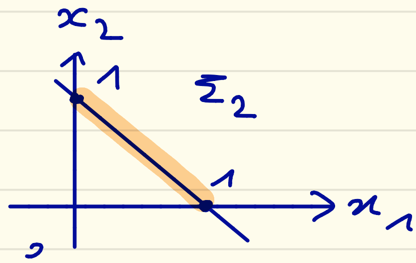
<u>Remarque</u> : soit $k \geq 1$, l'ensemble

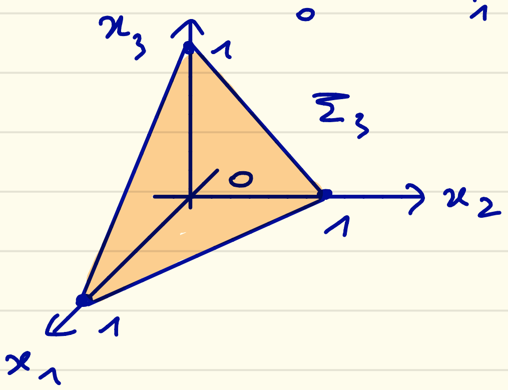
$$\Sigma_k := \{x \in \mathbb{R}^k \mid x_1 \geq 0, \dots, x_k \geq 0, x_1 + \dots + x_k = 1\}$$

s'appelle le simplexe unité de $\mathbb{R}^k$. C'est un ensemble compact (= fermé borné) et convexe.

<u>Ex.</u> $\Sigma_1 = \{1\}$







Les sommets (= $(1,0,0)$, $(0,1,0)$ et $(0,0,1)$ si $k=3$) de Σ_k correspondent aux stratégies pures (1, 2 ou 3, si $k=3$).

D'après le cours précédent, trouver un équilibre pour ce nouveau jeu (à somme nulle) dont les stratégies sont désormais étendues à $\Sigma_m \times \Sigma_n$, est équivalent à trouver un pt selle, c'est à dire un couple de stratégies $(\bar{y}, \bar{x}) \in \Sigma_m \times \Sigma_n$ tq :

$$\sup_{y \in \Sigma_m} \inf_{x \in \Sigma_n} (Ax \mid y) = \underset{\substack{\| \\ g(\bar{y}, \bar{x})}}{(A\bar{x} \mid \bar{y})} = \inf_{x \in \Sigma_n} \sup_{y \in \Sigma_m} (Ax \mid y)$$

<u>Th. (Von Neumann)</u> : un jeu fini admet un équilibre en stratégies mixtes. Autrement dit, soit A une matrice $m \times n$, il existe un point selle $(\bar y, \bar x) \in \Sigma_m \times \Sigma_n$ tq

$$\sup_{y \in \Sigma_m} \inf_{x \in \Sigma_n} (Ax|y) = \underbrace{(A\bar x|\bar y)}_{v} = \inf_{x \in \Sigma_n} \sup_{y \in \Sigma_m} (Ax|y).$$

De plus, si la valeur v du jeu est > 0, les pts-selles $(\bar y, \bar x) \in \Sigma_m \times \Sigma_n$ sont caractérisés par le fait qu'ils sont solutions des problèmes d'optimisation suivants :

i) $\bar x$ solution de
$$\begin{cases} (1_n | x) \longrightarrow \max \\ Ax \leq 1_m \\ x \geq 0 \end{cases}$$

$1_n = (1, \dots, 1) \in \mathbb{R}^n$
$1_m = (1, \dots, 1) \in \mathbb{R}^m$

ii) $\bar y$ solution de
$$\begin{cases} (1_m | y) \longrightarrow \min \\ {}^tA y \geq 1_n \\ y \geq 0 \end{cases}$$

<u>Remarque</u> : les problèmes i) et ii) sont des problèmes de programmation linéaire (LP = "Linear Programming") qui forment une classe de problèmes d'optimisation pour lesquels on dispose d'algorithmes de résolution efficaces, y compris en grande taille (m et n grands). Les éléments de preuve qui suivent utilisent ce point de vue.

dém.: soit A une matrice $m \times n$; quitte à remplacer A par $A + \|A\|_\infty + 1$ (ie à faire $a_{ij} \leftarrow a_{ij} + \max_{i,j} |a_{ij}| + 1$) on est sûr que tous les coeffs sont > 0 de sorte que $0 < \alpha \le \beta$.
Comme translater les valeurs des gains d'une même cte ne change pas les éventuelles stratégies à l'équilibre, on supposera donc qu'on a $0 < \alpha \le \beta$. Soit alors $t > 0$, on a :

$$t \geq \beta = \inf_{x \in \Sigma_n} \sup_{y \in \Sigma_m} (Ax|y)$$

$$= \min_{x \in \Sigma_n} \max_{y \in \Sigma_m} (Ax|y) \quad (\text{cf. compacité de } \Sigma_n \text{ et } \Sigma_m \dots)$$

$\Leftrightarrow (\exists x \in \Sigma_n) : \underbrace{\max_{y \in \Sigma_m} (Ax|y)}_{= \text{ la plus grande coordonnée du vecteur } Ax \in \mathbb{R}^m \text{ (évident !)}} \leq t$

$\Leftrightarrow (\exists x \in \Sigma_n) : Ax \leq t.1_m$

$\Leftrightarrow (\exists x \geq 0) : (1_n|x) = 1$ et $Ax \leq t.1_m$

$\Leftrightarrow (\exists x \geq 0) : Ax \leq 1_m$ et $(1_n|x) = 1/t$

$\Leftrightarrow 1/t \leq \max\{(1_n|x)$ avec $x \geq 0$ et $Ax \leq 1.m\}$

Donc, $\frac{1}{t} \leq \frac{1}{\beta} \Leftrightarrow \frac{1}{t} \leq \max\left\{(1_n|x) \,\middle|\, \begin{matrix} Ax \leq 1_m \\ x \geq 0 \end{matrix}\right\}$

ie $\frac{1}{\beta} = \max\left\{(1_n|x) \,\middle|\, \begin{matrix} Ax \leq 1_m \\ x \geq 0 \end{matrix}\right\}$.

Or, le lagrangien du problème d'optimisation

$$\text{i)} \begin{cases} (1_n|x) \longrightarrow \max \\ Ax \leq 1_m \\ x \geq 0 \end{cases} \Leftrightarrow \begin{cases} (-1_n|x) \to \min \\ (Ax - 1_m \leq 0) \\ -x \leq 0 \end{cases}$$

Avec contraintes inégalités s'écrit

$$L(x,y,\mu) = -(1_n|x) + (\mu|-x) + (y\,|\,Ax - 1_m).$$

Le problème i) est équivalent à

$$\operatorname*{Inf}_{x\in\mathbb{R}^n}\ \operatorname*{Sup}_{\substack{y\geq 0\\ \mu\geq 0}} L(x,y,\mu) \quad (\text{évident}),$$

lui-même équivalent par dualité (parce que le problème est linéaire donc convexe...) à

$$\operatorname*{Sup}_{\substack{y\geq 0\\ \mu\geq 0}}\ \underbrace{\operatorname*{Inf}_{x\in\mathbb{R}^n} L(x,y,\mu)}$$

$= -(1_m|y)$ si $\nabla_x L = 0$,

ie si $-1_n - \mu + {}^tAy = 0$

$$(\Leftrightarrow) \quad \operatorname*{Sup}_{\substack{y\geq 0\\ \mu\geq 0\\ {}^tAy = 1_n + \mu}} -(1_m|y)$$

$$(\Leftrightarrow) \quad \begin{cases} (1_m|y) \longrightarrow \min \\ {}^tAy \geq 1_n \\ y \geq 0 \end{cases} : \text{ii}).$$

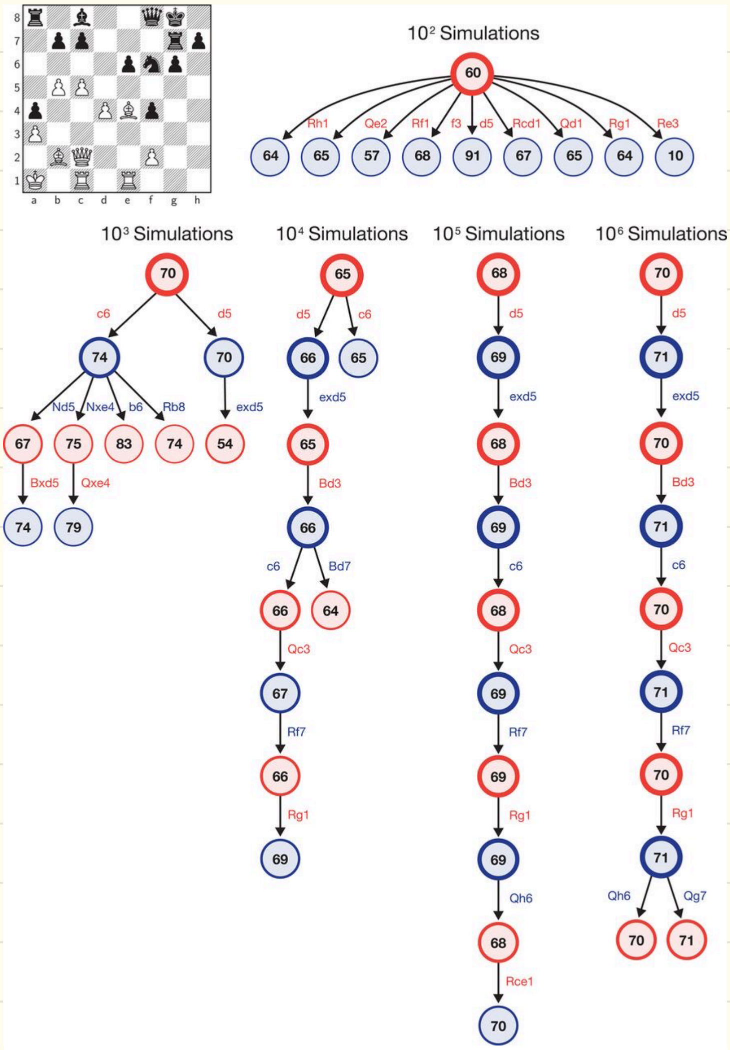
On montre de même que pour i) que la valeur de ii) est $1/\alpha$,

d'où $\frac{1}{\alpha} = \frac{1}{\beta}$, d'où l'existence de pts-selles et leur caractérisation à l'aide de i) et ii). □

Pour aller plus loin :

les jeux comme les échecs ou le go possèdent un nombre d'états trop grand pour qu'on leur applique directement ce type d'analyse. On utilise plutôt des méthodes de Machine Learning (apprentissage par renforcement, notamment) basées sur le même type d'outils mathématiques (optimisation, statistique, graphes...)

Le programme AlphaZero développé par DeepMind est actuellement capable de battre n'importe quel joueur humain aux échecs ou au go.



A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

David Silver[1,2,*,†], Thomas Hubert[1,*], Julian Schrittwieser[1,*], Ioannis Antonoglou[1], Matthew Lai[1], Arthur Guez[1], M...
+ See all authors and affiliations

Science 07 Dec 2018:
Vol. 362, Issue 6419, pp. 1140-1144
DOI: 10.1126/science.aar6404
