Enfoque Computacional para Juegos de Suma Cero

A continuación exploraremos el **enfoque computacional** para **juegos de suma cero** desde la óptica **min-max** o, equivalentemente, **max-min**. Nos enfocaremos especialmente en:

1. **Formulación del problema** en juegos de suma cero.
2. **Idea intuitiva** del *algoritmo minimax* (y por qué es relevante).
3. **Método de resolución** y **pseudocódigo** (principalmente a través de un enfoque de **Programación Lineal**).
4. **Complejidad computacional** (tanto para casos particulares como de forma general).
5. **Ejemplo** concreto de cómo se aplica este método.

---

1. Fundamentos: Juegos de Suma Cero y Minimax

1.1. Juego de Suma Cero

- Un **juego de suma cero** (en forma normal) con dos jugadores (llamados a menudo *Jugador Fila* y *Jugador Columna*) se describe por una **matriz de pagos** A de tamaño $m \times n$.
- Cuando el *Jugador Fila* (1) elige una fila i y el *Jugador Columna* (2) elige una columna j, el pago de Jugador 1 es A_{ij}.
- El pago de Jugador 2 es $-A_{ij}$. Así, la suma de ambos pagos es cero.
- Buscamos **estrategias mixtas** para cada jugador (distribuciones de probabilidad sobre sus acciones puras) que alcancen un **equilibrio de Nash**. Para juegos de suma cero, esto coincide con la solución **minimax** (o **maximin**).

1.2. Teorema Minimax

Von Neumann demostró que en los juegos de suma cero, el **valor de maximin** coincide con el **valor de minimax**, y la estrategia que lo logra es el **equilibrio de Nash**.

PROF

- Sea $\mathbf{x}$ la distribución de probabilidad del Jugador Fila sobre sus m filas ($\mathbf{x} \ge 0$, $\sum_i x_i=1$).
- Sea $\mathbf{y}$ la distribución de probabilidad del Jugador Columna sobre sus n columnas ($\mathbf{y}\ge 0$, $\sum_j y_j=1$).
- La utilidad esperada de Jugador 1 (Fila), jugando $\mathbf{x}$ contra $\mathbf{y}$, es $\mathbf{x}^\top A \mathbf{y}$.
- El Jugador 1 quiere **maximizar** este valor; el Jugador 2 quiere **minimizarlo** (equivalente a maximizar $-\mathbf{x}^\top A \mathbf{y}$).

El **Teorema del minimax** indica:
$
\max_{\mathbf{x}\ge 0, \sum x_i=1}
\min_{\mathbf{y}\ge 0, \sum y_j=1} \mathbf{x}^\top A \mathbf{y}
;=;
\min_{\mathbf{y}\ge 0, \sum y_j=1}

$$
\max_{\mathbf{x}\ge 0, \sum x_i=1} \mathbf{x}^\top A \mathbf{y}
\;=\; v,
$$
donde v es el **valor del juego**.

---

2. Idea Intuitiva del Algoritmo Minimax

Motivación:

- El Jugador Fila (1) mezcla sus acciones para **maximizar** su ganancia **asegurada** (independientemente de la estrategia del adversario).
- El Jugador Columna (2) mezcla sus acciones para **minimizar** la ganancia del primero.
- En el equilibrio, ninguno puede mejorar cambiando unilateralmente su mezcla: es la solución que iguala **maximin** y **minimax**.

Idea breve:

- Piense en Jugador 1 intentando "garantizar" que su ganancia sea al menos v.
- Para lograrlo, necesita una estrategia $\mathbf{x}$ tal que **para todas** las posibles columnas $\mathbf{y}$ de Jugador 2, la utilidad sea $\geq v$.
- Jugador 2, por su parte, quiere lograr que la ganancia sea $\leq v$.
- Resolveremos estas condiciones mediante **programación lineal**.

---

3. Formulación y Resolución por Programación Lineal

Existen varias formas de escribir el **problema primal** (para Jugador 1) y su **dual** (para Jugador 2). Aquí damos la formulación típica:

3.1. Modelo en PL (Programación Lineal) para Jugador Fila

Para un **juego de suma cero** con matriz de pagos A (dimensiones $m \times n$):

Objetivo: Encuentra $\mathbf{x} = (x_1,\dots,x_m)$ y un escalar v tales que:

$$
\max_{\mathbf{x},\,v} \; v
$$
sujeto a las restricciones:

1. $\sum_{i=1}^m x_i = 1$,
2. $x_i \ge 0 \quad \forall i,$
3. $\mathbf{x}^\top A \mathbf{e}_j \ge v \quad$ para cada columna j.
   - ($\mathbf{e}_j$ *es el vector canónico que significa "columna j pura"; en la práctica esto se traduce a* $\sum_i x_i A_{ij} \ge v$).

Interpretación:

- La tercera restricción garantiza que, contra cada **columna pura** del oponente, la utilidad sea al menos v. Por ende, si el oponente mezcla, la utilidad seguirá siendo $\geq v$.
- El objetivo es maximizar v.

3.2. Equivalencia con resolución mediante una variable de escalado

En muchos textos se introduce una variable "u" para eliminar la necesidad de v negativo, etc. Por ejemplo, a menudo se fuerza la matriz A a tener entradas no negativas sumándole un offset. Pero, a nivel conceptual, la formulación anterior ya da la idea.

3.3. Dual del Problema (para Jugador Columna)

El **dual** se interpreta como el problema de Jugador 2 que desea **minimizar** la utilidad de Jugador 1. El resultado de resolver el primal o el dual es el mismo valor v. Jugador 2 obtiene una mezcla $\mathbf{y}$ con la cual la utilidad no puede superar v.

---

4. Pseudocódigo del Algoritmo

Para resolver el **problema minmax** computacionalmente, típicamente se recurre a un **resolver de PL** (Programación Lineal). Sin embargo, a nivel de pseudocódigo, podemos dar una visión simplificada:


```
Entrada:
    - Matriz A (dimensión m x n) con valores (pueden ser positivos y/o
negativos).

Objetivo:
    - Hallar distribución x (x_i >= 0, sum x_i = 1)
    - Hallar valor v (escalar)
    - Tal que para cada columna j: sum_i( x_i * A[i,j] ) >= v
    - Maximizando v

Algoritmo (Bosquejo):

1. Normalizar (opcional):
   - Sea alpha = -min(A); (el valor más negativo de la matriz)
   - Construir A' = A + alpha (sumar alpha a cada entrada de A para
volverla no negativa).
     Esto a veces simplifica el tratamiento, pero no es obligatorio.

2. Crear variables y restricciones en un solver de PL:
   - Variables: x_1, x_2, ..., x_m (>=0), y la variable v.
   - Restricciones:
       R1: x_1 + x_2 + ... + x_m = 1
       R2: Para cada j en {1..n}:
             sum_i( x_i * A[i, j] ) >= v
       (Si se hizo normalización, adaptar la restricción para A'.)
   - Función Objetivo: Max v

3. Ejecutar un método de PL (por ejemplo, método símplex o interior-
```

```
point) sobre dichas restricciones.

4. Obtener solución:
   - (x_1^*, ..., x_m^*)  y valor v^*
   - Ese v^* es el "valor del juego" para el Jugador Fila.
   - (x_1^*,...,x_m^*) es la mezcla óptima del Jugador Fila.

5. (Opcional) Resolver el dual para obtener la mezcla y del Jugador
Columna:
   - Mín v
   - sujeta a: sum_j( y_j ) = 1, y_j >= 0, y "sum_j( y_j * A[i,j] ) <= v
para cada i"
   - O recuperar la mezcla dual con el método que devuelva los
multiplicadores duales.

Salida:
   - La estrategia mixta óptima de Jugador Fila (x^*)
   - La estrategia mixta óptima de Jugador Columna (y^*)
   - Valor del juego v^*
```

Nota: En la práctica, basta con usar **cualquier** solver de PL estándar (por ejemplo, *simplex*, *interior point*, *branch & bound* en caso de variables binarias, etc.) y plantear el problema con las restricciones indicadas.

---

5. Complejidad Computacional

5.1. Resolución de PL en general

- La **programación lineal** puede resolverse en **tiempo polinomial** en la *tamaño* de la entrada (con métodos de punto interior o el método del elipsoide).
- El método símplex, en el peor caso, puede tener complejidad exponencial, pero en la práctica suele ser muy eficiente.
- Para una matriz de **dimensión $m \times n$**, los **números** en la matriz influyen en la *longitud* de la descripción. Si la matriz tiene coeficientes que pueden representarse en L bits, las técnicas de punto interior usualmente resuelven en **$\mathrm{O}(\mathrm{poly}(m+n,,L))$**.

PROF

5.2. Casos específicos

1. **Juegos pequeños (m, n muy reducidos)**:

    - Podemos resolverlos directamente con la fórmula de PL y un solver genérico en un tiempo muy manejable.

2. **Juegos grandes (m, n grandes)**:

    - La dimensión de la matriz puede hacer que un método de PL con $\mathrm{O}(mn)$ restricciones (o variables) sea costoso, pero *sigue siendo polinomial* en $(m+n)$.
    - En problemas con m, n en el orden de millones, se requieren métodos más especializados (algoritmos de aproximación, descomposición, etc.).

3. **Juegos con estructura especial** (p.ej. "juegos matriciales esparsos" o "juegos en grafos"):
    - A veces podemos explotar la estructura para acelerar la resolución (por ejemplo, usando *column generation*, *row generation*, etc.).

5.3. Interpretación general

- **Polinomial** en el sentido de la **Teoría de la Complejidad**: existen algoritmos (elipsoide, interior point) que garantizan que el número de pasos es polinómico con respecto al número de bits necesarios para describir la instancia.
- En **casos prácticos**, el método símplex suele bastar: su **rendimiento medio** es muy bueno, a pesar de su peor caso exponencial.

Ejemplo Completo: Resolución de un Juego de Suma Cero

1. Definición del Juego

Consideremos un juego de suma cero entre dos jugadores: el Jugador Fila y el Jugador Columna. El juego está definido por la siguiente matriz de pagos A:

$$
A = \begin{pmatrix}
4 & 0 & 2 \\
2 & 3 & -1
\end{pmatrix}
$$

Esta matriz representa los pagos para el Jugador Fila. Dado que es un juego de suma cero, los pagos para el Jugador Columna son exactamente $-A$.

Interpretación del juego

- El **Jugador Fila** tiene 2 estrategias puras: F_1 y F_2
- El **Jugador Columna** tiene 3 estrategias puras: C_1, C_2 y C_3
- Cada entrada A_{ij} indica cuánto gana el Jugador Fila cuando elige la estrategia i y el Jugador Columna elige la estrategia j
- Por ejemplo, si Jugador Fila elige F_1 y Jugador Columna elige C_3, entonces el Jugador Fila gana 2 unidades (y el Jugador Columna pierde 2 unidades)

Forma normal completa

	C_1	**C_2**	**C_3**
F_1	(4, -4)	(0, 0)	(2, -2)
F_2	(2, -2)	(3, -3)	(-1, 1)

Donde cada celda contiene (pago al Jugador Fila, pago al Jugador Columna).

2. Enfoque de Resolución: Programación Lineal

Para resolver este juego, buscaremos la estrategia mixta óptima para cada jugador y el valor del juego. Recordemos que una estrategia mixta es una distribución de probabilidad sobre las estrategias puras.

2.1 Formulación para el Jugador Fila

El Jugador Fila busca una estrategia mixta $\mathbf{x} = (x_1, x_2)$ que maximice su ganancia mínima.

Variables:

- $x_1, x_2 \geq 0$ con $x_1 + x_2 = 1$ (probabilidades de usar F_1 y F_2)
- v = valor del juego (a maximizar)

Restricciones (una por cada columna):

- Columna C_1: $4x_1 + 2x_2 \geq v$
- Columna C_2: $0x_1 + 3x_2 \geq v$
- Columna C_3: $2x_1 - 1x_2 \geq v$

Función objetivo:
$\max ; v$

2.2 Resolución Analítica

Empezamos con la restricción $x_1 + x_2 = 1$, o equivalentemente, $x_2 = 1 - x_1$.

Reescribamos las restricciones:

- (1) $4x_1 + 2x_2 \geq v$, o equivalentemente, $4x_1 + 2(1-x_1) = 4x_1 + 2 - 2x_1 = 2x_1 + 2 \geq v$
- (2) $0x_1 + 3x_2 \geq v$, o equivalentemente, $3(1-x_1) = 3 - 3x_1 \geq v$
- (3) $2x_1 - 1x_2 \geq v$, o equivalentemente, $2x_1 - (1-x_1) = 2x_1 - 1 + x_1 = 3x_1 - 1 \geq v$

Estrategia de Resolución

En el equilibrio, el valor v estará definido por al menos dos de estas restricciones activas (igualdades). Analizaremos todas las posibles combinaciones de restricciones activas.

Caso 1: Restricciones (1) y (2) activas

Igualamos:

- De (1): $v = 2x_1 + 2$
- De (2): $v = 3 - 3x_1$

Resolviendo: $2x_1 + 2 = 3 - 3x_1$, lo que nos da $5x_1 = 1$, por lo tanto $x_1 = 0.2$

Con $x_1 = 0.2$, calculamos:

- $x_2 = 1 - x_1 = 1 - 0.2 = 0.8$
- $v = 2(0.2) + 2 = 0.4 + 2 = 2.4$

Verificamos la restricción (3):
$3x_1 - 1 = 3(0.2) - 1 = 0.6 - 1 = -0.4$

Como $-0.4 < 2.4$, la restricción (3) no se cumple. Por lo tanto, este caso no es viable.

Caso 2: Restricciones (1) y (3) activas

Igualamos:

- De (1): $v = 2x_1 + 2$
- De (3): $v = 3x_1 - 1$

Resolviendo: $2x_1 + 2 = 3x_1 - 1$, lo que nos da $x_1 = 3$

Este valor no es admisible porque x_1 debe estar entre 0 y 1 (es una probabilidad).

Caso 3: Restricciones (2) y (3) activas

Igualamos:

- De (2): $v = 3 - 3x_1$
- De (3): $v = 3x_1 - 1$

Resolviendo: $3 - 3x_1 = 3x_1 - 1$, lo que nos da $3 + 1 = 3x_1 + 3x_1$, por lo tanto $4 = 6x_1$, es decir, $x_1 = \frac{2}{3}$

Con $x_1 = \frac{2}{3}$, calculamos:

- $x_2 = 1 - x_1 = 1 - \frac{2}{3} = \frac{1}{3}$
- $v = 3 - 3(\frac{2}{3}) = 3 - 2 = 1$

Verificamos la restricción (1):

$2x_1 + 2 = 2(\frac{2}{3}) + 2 = \frac{4}{3} + 2 = \frac{4}{3} + \frac{6}{3} = \frac{10}{3} \approx 3.33$

Como $3.33 > 1$, la restricción (1) se cumple con holgura. Por lo tanto, este caso es viable.

Validación final

Verifiquemos que la estrategia $\mathbf{x} = (\frac{2}{3}, \frac{1}{3})$ con valor $v = 1$ satisface todas las restricciones:

- Para C_1: $4(\frac{2}{3}) + 2(\frac{1}{3}) = \frac{8}{3} + \frac{2}{3} = \frac{10}{3} \approx 3.33 > 1$ ✓
- Para C_2: $0(\frac{2}{3}) + 3(\frac{1}{3}) = 0 + 1 = 1 \geq 1$ ✓
- Para C_3: $2(\frac{2}{3}) - 1(\frac{1}{3}) = \frac{4}{3} - \frac{1}{3} = 1 \geq 1$ ✓

Las restricciones (2) y (3) se cumplen con igualdad, como esperábamos, mientras que la restricción (1) se cumple con holgura.

3. Problema Dual: Estrategia del Jugador Columna

El problema dual corresponde a la estrategia óptima del Jugador Columna.

Variables:

- $y_1, y_2, y_3 \geq 0$ con $y_1 + y_2 + y_3 = 1$ (probabilidades de usar C_1, C_2 y C_3)
- w = valor del juego (a minimizar)

Restricciones (una por cada fila):

- Fila F_1: $4y_1 + 0y_2 + 2y_3 \leq w$
- Fila F_2: $2y_1 + 3y_2 - 1y_3 \leq w$

Función objetivo:
$\min ; w$

3.1 Resolución del Problema Dual

Por la teoría de la dualidad en programación lineal y el teorema minimax, sabemos que $v = w$. Además, las restricciones activas en el problema primal corresponden a las variables positivas en la solución dual.

Dado que las restricciones (2) y (3) estaban activas en el problema primal, esperamos que $y_2 > 0$ y $y_3 > 0$, mientras que $y_1 = 0$.

Verificamos mediante las ecuaciones:

- Para F_1: $4y_1 + 0y_2 + 2y_3 = w = 1$
- Para F_2: $2y_1 + 3y_2 - 1y_3 = w = 1$

Con $y_1 = 0$, obtenemos:

- De la primera ecuación: $2y_3 = 1$, por lo tanto $y_3 = \frac{1}{2}$
- De la segunda ecuación: $3y_2 - 1y_3 = 1$, lo que nos da $3y_2 = 1 + y_3 = 1 + \frac{1}{2} = \frac{3}{2}$, por lo tanto $y_2 = \frac{1}{2}$

Verificamos que $y_1 + y_2 + y_3 = 0 + \frac{1}{2} + \frac{1}{2} = 1$ ✓

Por lo tanto, la estrategia óptima del Jugador Columna es $\mathbf{y} = (0, \frac{1}{2}, \frac{1}{2})$.

4. Interpretación de los Resultados

Estrategias Óptimas:

- **Jugador Fila**: $\mathbf{x}^* = (\frac{2}{3}, \frac{1}{3})$
    - Debe jugar la estrategia F_1 con probabilidad $\frac{2}{3}$ y la estrategia F_2 con probabilidad $\frac{1}{3}$
- **Jugador Columna**: $\mathbf{y}^* = (0, \frac{1}{2}, \frac{1}{2})$

    - Debe jugar la estrategia C_2 con probabilidad $\frac{1}{2}$ y la estrategia C_3 con probabilidad $\frac{1}{2}$
    - Nunca debe jugar la estrategia C_1

Valor del Juego:

- $v^* = 1$
    - Este valor positivo indica que el juego favorece al Jugador Fila
    - En promedio, el Jugador Fila ganará 1 unidad por partida si ambos jugadores usan sus estrategias óptimas
    - El Jugador Columna perderá en promedio 1 unidad por partida

Verificación Cruzada:

Podemos calcular el valor esperado del juego cuando ambos jugadores emplean sus estrategias óptimas:

$\mathbf{x}^{T} A \mathbf{y}^ = [\frac{2}{3}, \frac{1}{3}] \begin{pmatrix} 4 & 0 & 2 \ 2 & 3 & -1 \end{pmatrix} [0, \frac{1}{2}, \frac{1}{2}]^T$

$= [\frac{2}{3}, \frac{1}{3}] \begin{pmatrix} 0 & 1 \ 1.5 & -0.5 \end{pmatrix}$

$= [\frac{2}{3} \cdot 0 + \frac{1}{3} \cdot 1.5, \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot (-0.5)]$

$= [0.5, \frac{2}{3} - \frac{1}{6}] = [0.5, 0.5]$

$= 0.5 + 0.5 = 1$

Confirmamos que el valor esperado del juego es efectivamente 1, como habíamos calculado.

5. Conclusiones

- Las **estrategias mixtas** son fundamentales en los juegos de suma cero cuando no existe un **equilibrio en estrategias puras**.

- La **programación lineal** proporciona un método efectivo para encontrar las estrategias óptimas y el valor del juego.

- Las **restricciones activas** (que se cumplen con igualdad) nos indican qué estrategias puras del oponente son igualmente óptimas en el equilibrio.

- El **valor del juego** nos indica qué jugador tiene ventaja: si es positivo, favorece al Jugador Fila; si es negativo, favorece al Jugador Columna.

- La **dualidad** en programación lineal nos permite resolver tanto para la estrategia del Jugador Fila como para la del Jugador Columna.